國 立 交 通 大 學

電機與控制工程學系

博 士 論 文

交 通 參 數 估 測 系 統 之
攝影機參數校正與影像追蹤

Camera Calibration and Image Tracking for Traffic Parameter Estimation

研 究 生：戴任詔

指導教授：宋開泰 博士

中 華 民 國 九 十 五 年 七 月

交 通 參 數 估 測 系 統 之
攝影機參數校正與影像追蹤

Camera Calibration and Image Tracking for Traffic Parameter Estimation

研 究 生：戴任詔　　Student: Jen-Chao Tai

指導教授：宋開泰 博士　　Advisor: Dr. Kai-Tai Song

國 立 交 通 大 學

電 機 與 控 制 工 程 學 系

博 士 論 文

A Dissertation

Submitted to Department of Electrical and Control Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Electrical and Control Engineering

July 2006

Hsinchu, Taiwan, Republic of China

中 華 民 國 九 十 五 年 七 月

交通參數估測系統之攝影機參數校正與影像追蹤

學生:戴任詔　　指導教授:宋開泰 博士

國立交通大學電機與控制工程學系

摘要

本論文主要探討運用於交通參數估測系統之先進影像處理技術，包括動態攝影機參數校正、背景影像建立、影子移除、車輛偵測追蹤與交通參數量測。進行交通參數估時，首先必須準確地校正攝影機參數，方能利用二維影像畫面來精確地求出物體於空間之三維物體位置，本論文設計一套可自動校正 Pan-Tilt-Zoom 攝影機參數之方法，發展出利用一組平行的車道標線與其在影像畫面中之幾何位置，產生焦距方程式來求焦距，再計算出其他攝影機參數值。在車輛偵測與追蹤前處理階段，一般利用背景影像移除方法將移動中車輛由影像畫面中分離出來，本論文提出基於群組直方圖 (Group-Based Histogram)方法，可快速建立良好之背景影像，此方法對於感測雜訊與慢速移動車輛具強健性。進行交通監測影像分析時，車輛的影子會造成車輛影像嚴重變形，甚至與其他車輛影像產生重疊，嚴重影響車輛偵測與追蹤的準確性。本論文利用影子色彩特性與統計方法，提出一個色彩空間比值模型，此模型可迅速偵測影像中之影子像素，配合兩種幾何分析方法再提昇影子偵測的準確性。以這個比值模型所設計的動態車輛偵測方法比現有方法更有效率。交通參數估測時，必須對多車道中大小不一之車輛進行追蹤，本論

文發展一套自動輪廓初始化方法，利用特殊設計之偵測窗來偵測進入影像畫面中多車道內任意位置且大小不同之車輛，並依據車輛大小與位置所產生車輛之初始追蹤輪廓，再利用卡曼濾波器進行追蹤，分析追蹤結果可得到車流與車速之交通參數。另外，針對T字路口轉彎率量測，本論文設計一套可即時判定車輛移動方向的光流偵測技術，並結合偵測窗來量測路口之車輛轉彎率。本論文所發展之方法理論，均已利用實際道路交通影像進行驗證，量測所得之交通參數，如平均車速、車流量、車流密度與轉彎率，誤差值在5%內，顯示本論文所提出的方法確實能正確且快速完成交通參數估測。

Camera Calibration and Image Tracking for Traffic Parameter Estimation

Student: Jen-Chao Tai Advisor: Dr. Kai-Tai Song

Department of Electrical and Control Engineering

National Chiao Tung University

ABSTRACT

The objective of this thesis is to study advanced image processing methodologies for estimating traffic parameters with functional accuracy. The developed methodologies consist of camera calibration, single Gaussian background modeling and foreground segmentation, shadow suppression, vehicle detection and tracking, and optical-flow-based turn ratio measurement. The accuracy of estimating vehicle speed depends not only on image tracking but also on the accuracy of camera calibration. A novel algorithm has been proposed for automatic calibration of a pan-tilt-zoom camera overlooking a traffic scene. A focal length equation has been derived for camera calibration based on parallel lane markings. Subsequently, the pan and tilt angles of the camera can be obtained using the estimated focal length. To locate the parallel lane markings, we develop an image processing procedure. In the preprocessing step of vehicle detection and tracking algorithm, foreground segmentation can be accomplished by using background removal. The quality of background generation affects the performance of foreground segmentation. Thus, a group-based histogram algorithm has been designed and implemented for the estimation of a single Gaussian model of a

background pixel in real-time. The method is effective and efficient for building the Gaussian background model from traffic image sequences. It is robust against sensing noise and slow-moving objects. However, shadows of moving objects often cause serious errors in image analysis due to the misclassification of shadows as moving objects. A shadow-region-based statistical nonparametric method has been developed to construct a RGB ratio model for shadow detection of all pixels in an image frame. This method of shadow model generation is more effective than existing methods. Additionally, two types of spatial analysis have been employed to enhance the shadow suppression performance.

An automatic contour initialization procedure has been developed for image tracking of multiple vehicles based on an active contour and image measurement approach. The method has the capability to detect moving vehicles of various sizes and generate their initial contours for image tracking in a multi-lane road. The proposed method is not constrained by lane boundaries. The automatic contour initialization and tracking scheme has been tested for traffic monitoring. Additionally, this paper proposes a method for automatically estimating the vehicle turn ratio at an intersection by using techniques of detection window and optical flow measurement. Practical experimental studies using actual video clips are carried out to evaluate the performance of the proposed method. Experimental results show that the proposed scheme is very successful in estimating traffic conditions such as traffic flow rate, vehicle speeds, traffic density, and turn ratio.

誌謝

衷心感謝我的指導教授宋開泰博士，多年來，在專業上或處理事情上，真摯誠意且不厭其煩的給我指導及修正，使我受益良多，順利完成博士學業。感謝論文口試委員—交通大學 莊仁輝教授、中央研究院 廖弘源教授、暨南國際大學 王文俊院長、清華大學 彭明輝教授與交通大學 林進燈院長，給我的建議與指引，強化本篇論文的嚴整性與可讀性。感謝明新科技大學的支持，歷屆校長、各級長官、機械系主任、同事們的鼓勵與行政支援，讓我能盡全力進行博士班研究課業。

感謝 ISCI 實驗室的學弟們多年來提供的協助，尤其是參與本論文初期研究的學弟，如舜璋、清波、辰竹，也感謝博士班學弟奇諡與孟儒對本論文的建議與討論，另外特別感謝十項全能的嘉豪，在軟體程式應用發展上各項協助。感謝教育部，由『下世代人性化智慧型運輸系統』卓越計畫提供的硬體設備與經費，讓許多實驗得以順利完成，並感謝林進燈院長的資訊媒體實驗室鶴璋學長出借 PTZ 攝影機，讓我順利完成動態攝影機參數校正實驗。也感謝謝明新科大專題學生，特別是建榮與文淇，在炎熱夏日中，進行道路交通影像拍攝，以供實驗進行。

感謝父母與岳母的鼓勵與栽培，在父母重病其間，哥哥與姊姊的扶持提攜，公園教會兄姐的代禱，讓我能夠成長，完成學業。感謝內人 美生體諒我就讀研究所博士班，課業繁重，協助我照料家庭，免除後顧之憂，並將全家一切需要放在禱告中，向全能的上帝祈求賜福，使我能潛心向學，最值得我感謝。也謝謝體貼人意的女兒涵妮與涵芸的加油與鼓勵，給予我無比的勇氣，但在你們成長的歲月，為人父的我，因為課業壓力沒有好好陪伴成長。希望在未來人生旅途上，全家都能在　神保守下，和睦相愛。

最後，感謝全能的上帝，在我人生各個階段的參與、保守與帶領，希望未來能行於主道並宣揚福音，將一切榮耀歸給上帝。

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Image-based traffic monitoring has become an active research area in recent years for the development of intelligent transportation systems (ITS). Under the framework of advanced transportation management and information systems (ATMIS), services such as traveler information, route guidance, traffic control, congestion monitoring, incident detection, and system evaluation across complex transportation networks have been extensively studied to enhance traveling safety and efficiency [1]-[3]. For these advanced applications, various types of traffic information need to be collected online and distributed in real time. Hence, automatic traffic monitoring and surveillance have become increasingly important for road usage and management.

Various sensor systems have been applied to estimate traffic parameters. Currently, magnetic loop detectors are the most widely-used sensors, but they are difficult to install and maintain. It is widely recognized that image-based systems are flexible and versatile for advanced traffic monitoring and surveillance applications. Compared with loop detectors, image-based traffic monitoring systems (ITMS) provide more flexible solutions for estimating traffic parameters [4]-[8]. In ITMS, it is important to segment and track various moving vehicles from image sequences. Thanks to image tracking techniques, the pixel coordinates of each moving vehicle can be recorded in each time frame. Using calibrated camera parameters,

one can transform the pixel coordinates of moving vehicles into their world coordinates [9]. Accordingly, useful traffic parameters—including vehicle speeds, vehicle travel directions, traffic flow, *etc.*—can be obtained from image measurement. These quantitative traffic parameters are useful for traffic control and management [6]. However, it is still a challenge to obtain traffic parameters in real-time for ITMS. Great uncertainties exist in traffic imagery for extracting useful information. Robust tracking of moving vehicle requires advanced image processing techniques to handle external uncertainties coming from shadows, background generation, and dynamic camera calibration. Further, urgent attention is needed to simultaneously detect motorcycles and cars in urban area of Taiwan and many Asia cities. This study aims to investigate methodologies for tangible real-time on-line traffic monitoring systems.

1.2 Literature Survey

1.2.1 Dynamic Camera Calibration

For an ITMS, the basic function is to automatically extract real-time traffic parameters using image processing techniques [10]-[13]. Traffic parameters, such as vehicle speed, are often obtained using image tracking techniques. The accuracy of traffic parameter estimation is affected by camera parameters, moving vehicle segmentation, and the tracking algorithms. An ITMS works only if the cameras are calibrated properly, and its accuracy is very sensitive to the calibration results. Moreover, in order to obtain flexibility of views and range of observation, an increasing number of ITMS systems rely on moveable cameras with adjustable pan/tilt and zoom settings. Proper calibration of camera parameters for pan-tilt-zoom (PTZ) cameras plays an important role in image-based traffic applications.

Most calibration methods in traffic monitoring and surveillance [14]-[17] utilize known information in a scene to estimate camera parameters, including tilt angle, pan angle, and

focal length of the camera. In [18] and [19], sets of parallel lines of a hexagon are employed to estimate the camera parameters. Results from these presentations demonstrate that parallel lines can be employed to adequately determine camera parameters. Effective algorithms have been developed for estimating camera parameters using parallel lanes in a traffic scene [20]-[22]. Bas and Crisman [20] used the height and the tilt of the camera along with a pair of parallel lines in a traffic scene to calibrate the camera. However, their approach requires special manual operations to measure the tilt of the camera. In [21] and [22], multiple parallel lanes and a special perpendicular line were used to calibrate the camera parameters. The drawback of their design is that a perpendicular line seldom appears in a traffic scene. Moreover, the lane markings need to be manually assigned in all the methods mentioned above, which is impractical for a traffic monitoring system using PTZ cameras, where manual operation should be avoided. Therefore, it will be necessary for the ITMS to possess the capacity of PTZ cameras to be dynamically calibrated.

In their recent presentation on dynamic calibration [23], Schoepflin and Dailey employed the trajectories as well as the bottom edges of vehicles to obtain two sets of parallel lines for PTZ camera calibration. The calibration procedure can be automated using the presented approach. However, the accuracy is considerably sensitive to the trajectories of the vehicles in traffic imagery. Furthermore, to obtain reliable tracks with high quality, the system takes a longer time to capture a larger number of image frames for recording the recognizable tracks of vehicles. It becomes very time-consuming and cannot meet the real-time requirement of ITMS. For practical applications, a method to speed up the process and obtain stable results is urgently required.

In this thesis, a novel focal length equation will be derived to estimate the PTZ camera parameters. The derivation requires only a single set of parallel lane markings, the lane width, and the camera height. Compared with existing approaches, the proposed method has the

advantage of requiring neither the camera tilt information nor multiple sets of parallel lines. Furthermore, an image processing algorithm is also proposed to automatically locate the edges of lane markings. Using the lane-marking edges and the derived focal length equation, one can estimate the focal length, the tilt and pan angles of a PTZ camera.

1.2.2 Background Modeling and Foreground Segmentation

Many approaches to moving vehicle segmentation have been studied for image-based traffic monitoring [6]-[8]. Background removal is a powerful tool for extracting foreground objects from image frames. The performance of background removal is affected by the quality of background estimation. A fixed camera for monitoring a scene is a common practice of surveillance applications. One of the key steps in analyzing the scene is to detect moving objects. In general, a common assumption is that the imagery of a background scene without any moving objects can be described well by a statistical model. If one obtains the statistical model of the background imagery, moving objects can be detected by checking the captured pixel intensity with its corresponding background model. Thus background estimation is an important step in image-based surveillance applications and traffic monitoring and enforcement applications that involve motion segmentation utilizing a near-stationary background [24]-[26].

A simple method for obtaining a background image is averaging sampled images over a span of time. The main drawback of this approach is that the foreground detection in the next step is sensitive to the noise induced by *ghosts* or *trails* of the averaged image. Zheng *et al.* [27] employed all incoming intensities of a pixel (including background and foreground object) to construct the Gaussian background model. However, in their method the foreground-object intensity degrades the quality of background estimation. To obtain a better Gaussian background model, Kumar *et al.* [28] utilized a method to monitor the intensities

from several frames without any foreground object for a few seconds. The drawback of their approach is that it can be difficult to find enough foreground-free frames to build a reliable intensity distribution of the background image.

In recent years, Gaussian mixture model (GMM)-based approaches to obtaining reliable background images have gained increasing attention for image-based motion detection [29]-[30]. GMM–based methods feature effective background estimation under environmental variations through a mixture of Gaussians for each pixel in an image frame. However, this approach has an important shortcoming when applied to ITMS. In urban traffic, vehicles stop occasionally at intersections because of traffic lights or control signals. Such kinds of transient stops increase the weight of non-background Gaussians and seriously degrade the background estimation quality of a traffic image sequence.

For an image sequence captured by a static camera, pixel values may have complex distributions, and the intensity of a background pixel will dominate the largest Gaussian. The intensity of a background pixel can be found if it is most frequently recorded at this pixel position. Thus, several histogram approaches have been studied to estimate the single Gaussian background model [31]-[32]. In histogram approaches, the intensity with the maximum frequency in the histogram is treated as background intensity, because each intensity frequency in the histogram is proportional to its appearance probability. The mean of the Gaussian background model can be determined by searching for the maximum frequency in the intensity histogram. Accordingly, the standard deviation can be computed using the background-intensity region (centered at the mean). Intensities of the transient stop foreground will not be recognized as the background intensity because its frequency is smaller than the background pixel. However, often more than one intensity level will have the same maximum frequencies in the histogram. Thus, it is difficult to determine the mean intensity of a Gaussian background model from the histogram. To increase the robustness against

intensity variation, the system needs to process many more image frames to establish a reliable intensity distribution. However, this will cost much more computation time and degrade the real-time performance.

Although the conventional histogram approach is robust to the transient stops of moving foreground objects, the estimation is still less accurate than GMM. In this thesis, we propose a novel group-based histogram (GBH) approach for estimating the Gaussian background model. The GBH effectively exploits an average filter to smooth the frequency curve of a conventional histogram for a more accurate estimation of the Gaussian mean. Accordingly, the standard deviation can be estimated by using the estimated mean and the histogram. One can easily and efficiently estimate the single Gaussian model constructed by background intensities from image sequences during a fixed span of time.

1.2.3 Shadow Suppression for Image Analysis

Shadows of moving objects often cause serious errors in image analysis due to the misclassification of shadows or moving objects. Shadows occur when a light source is partially or totally occluded by an object in the scene. In traffic imagery, shadows attached to their respective moving vehicles introduce distortions and cause problems in image segmentation. Therefore, cast-shadow suppression is an essential prerequisite for image-based traffic monitoring. In order to increase the accuracy of image analysis, it is desirable to develop a method to separate moving vehicles from their shadows.

Many algorithms have been proposed to detect and remove moving cast-shadows in traffic imagery [33]-[40]. Shadow suppression can be classified into two main categories: shape-based approaches and spectrum-based approaches. Shape-based methodologies employ a priori geometric information of the scenes, the objects, and the light-source location to solve the shadow detection problem. Hsieh *et al.* employed the lane features and line-based

algorithm to separate all unwanted shadows from the moving vehicles [33]. Yoneyama *et al.* designed 2D joint-vehicle/shadow models to represent the objects and their attached shadows in order to separate the shadows from the objects [34]. Under specific conditions, such as when vehicle shapes and illumination directions are known, these models can accurately detect shadows. However, they are difficult to implement, since the knowledge of scene conditions, object classes, and illumination conditions are not readily available in practical applications.

On the other hand, spectrum-based techniques identify shadows by exploiting spectral information of lit regions and shadow regions to detect shadows [35]-[36]. Compared with the shape-based approaches, color information inference is more explicit, because the spectral relationship between lit regions and object-shadow regions is only affected by illumination, not by object shapes and light source directions. Horprasert *et al.* proposed a computational RGB color model to detect shadows [37]. Brightness distortions and chromaticity distortions are defined and normalized to classify each pixel. The detection accuracy of the algorithm is sensitive to the background model; however, it is often difficult to obtain enough foreground-free frames to build a reliable background model in traffic scenes. Cucchiara *et al.* used the Hue-Saturation-Value (HSV) color information to extract the shadow pixels from previously extracted moving foreground pixels [38]. The threshold operation is performed to find the shadow pixels. The empirically-determined thresholds dominate the detection accuracy. The shortcoming of their method is that it is not an analytical method. Salvador *et al.* employed invariant color features to detect shadows [39]. Invariant color features describe the color configuration of each pixel discounting shading, shadows, and highlights. These features are invariant to a change in the illumination conditions, and therefore, are powerful indexes for detecting shadows. However, the computational load is rather high. Bevilacqua proposed a gray-ratio-based algorithm to effectively detect shadows with an empirical ratio

threshold [40]. The gray-ratio-based analysis considerably shortens the computation time, but often misclassifies objects as shadows because pixels of different colors sometimes have the same gray value.

We propose in this work a new, analytically-solved shadow model for effectively detecting shadows. Current color-constancy-based shadow suppression methods examine each pixel of the image to build the shadow models. Analyzing the spectrum properties of each pixel covered by shadows in an image sequence is strenuous work. Furthermore, it is difficult to obtain enough shadow properties of some pixels for probability analysis from image sequences. In traffic imagery, the region of interest is the roadway in the background image and light sources like the sun and the sky are practically fixed. The road can be treated as a Lambertian surface, since the sun direction does not change over a few frames and the light source from the sky is pretty uniform. Thus, it is unnecessary to model each pixel respectively for shadow detection because the shadow property of each road pixel is similar under the same Lambertian condition [41]. Nevertheless, the color values of each pixel are different from each other in color space, so it is impossible to construct a unique model for each pixel merely by using color space. Investigating of other shadow properties to resolve the unique model is crucial for cast-shadow detection.

In this thesis, a shadow-region-based statistical nonparametric approach is developed to construct a unique model for shadow detection of all pixels in an image frame. To effectively and analytically build a model for shadow detection, the model is established not by examining each pixel in image sequences but through using a shadow region in a single image frame. In our design, a color ratio between lit pixels and shadow pixels is utilized as an index to establish the unique model for different shadow pixels. We will show later that the ratio can be considered constant, under the Lambertian condition, in all image sequences. The model generation procedure requires much less effort and hence computational load compared with

current available methods. For further improving the performance of shadow suppression, a post-processing stage of spatial analysis is added to verify the actual shadow pixels. The shapes and boundary information of the detected shadow region are used to verify the actual shadow pixels.

1.2.4 Vehicle Detection and Tracking

The information obtained from image measurement allows precise vehicle tracking and classification. Thanks to image processing techniques, useful traffic parameters including vehicle speeds, vehicle travel direction, traffic flow, *etc.* can be obtained. The study of image tracking for traffic monitoring has gained increasing attention in recent years [42]-[43]. However, in practical applications, multiple vehicles on a multi-lane road need to be segmented and tracked simultaneously. A timely tracking initialization procedure plays an important role in traffic image tracking.

Many powerful tools for real-time image tracking have been proposed for traffic parameter estimation. Pece and Worrall [44] proposed an expectation-maximization (EM) contour algorithm to track vehicles. Their method used cluster analysis of image difference to accomplish the initialization. Lim *et al.* [45] employed a feature-based algorithm to obtain the state of the vehicle. Loop detectors were used to initialize image tracking for vehicle speed estimation in their design. Masoud *et al.* [11] employed sets of blobs and rectangular patches to track vehicles. Their method established the correspondence among blobs and tracked vehicles for tracking initialization. Kamijo *et al.* [46] employed a spatio-temporal Markov random field model to obtain the state of each pixel for tracking purposes. They employed a slit at each entrance to examine entering vehicles and initiate a tracking process. Hsu *et al.* [47] employed a detection zone and entropy concept to monitor similar-sized cars. Lai *et al.* [48] employed the idea of virtual loop and direction-based motion estimation to classify and track

vehicles, assigning virtual loops to each lane for tracking. In the previous work of our group [49], an initialization method exploiting a detection line was proposed based on the concept of contour growing. When a vehicle hits the detection line, a grabbing mode is activated to generate a contour to cover the part of the vehicle that has entered the detection region. The algorithms mentioned above all achieve reliable vehicle tracking in a road scene. However, one drawback of these methods is that they simply use fixed-size models for all vehicles. Their methods do not have the capacity to detect and track vehicles of various sizes simultaneously. Thus, it is desirable to develop a tracking algorithm for general traffic imagery that can handle multi-sized vehicles, including cars and motorcycles. Moreover, the previous methods cannot handle vehicles that travel across lane boundaries in the initialization stage of image tracking. Therefore, urgent attention is needed both to develop initialization and tracking algorithms for detecting vehicles of various sizes and to generate initial tracking for all moving vehicles in any position of a multi-lane road. In this thesis, we address both problems by proposing an image-based traffic monitoring system that automatically detects and tracks multiple different-sized vehicles that travel in any portion of a multi-lane road.

1.2.5 Turn Ratio Measurement

The information regarding traffic flow from an individual origin to a destination is required for traffic planning and control [50]-[51]. Traffic flow data are usually organized in a matrix termed origin/destination matrix (OD matrix) in the study of traffic management systems [52]. Each element of the matrix represents the traffic flow from a specific origin to a specific destination referred to a time unit. In traffic networks, the dynamic OD matrix characterizes the time-variant traffic demand mode between each pair of origin and destination. It is an essential input for dynamic traffic assignment models and is also useful for online identification and control of traffic systems. Thus, many researchers have studied

dynamic OD matrix estimation and prediction [53]-[54]. In traffic modeling, vehicle turn ratios as well as traffic flow data at intersections are employed to estimate OD matrices. A traffic monitoring system must detect all passing vehicles and provide vehicle counts to the ATMIS for further analysis. Therefore, at an intersection with signals, the traffic surveillance system needs to classify the motion of vehicles, such as moving straight ahead, or turning left or right [51], [54]-[55]. The turn ratio of passing vehicles, computed from the count of vehicles traveling straight and turning, is an important component in dynamic OD estimation.

However, current techniques are not suitable for providing useful information to dynamic OD estimation [56]-[57]. Most researchers still employ manual counting to estimate the turn ratio for their studies of traffic control and prediction. To our best knowledge, there is not yet a solution for image-based turn ratio measurement. It is desirable to automate the measurement of this important parameter, not only because the execution difficulties can be reduced, but also because reliability and real-time performance can be increased.

This thesis proposes an image processing procedure for segmenting and counting passing vehicles that travel in prescribed driving directions from an image sequence. From the detected vehicle count of each driving direction at a road intersection, the corresponding turn ratio can be automatically measured. Specially designed detection windows are provided to count the passing vehicles at an intersection. Furthermore, vehicles with different driving directions are separated by exploiting an optical flow measurement. Many researchers have devoted their efforts to algorithms of optical flow estimation [58]-[59]. However, most current algorithms require many iterations to estimate the optical flow field before a stable solution can be found. In traffic monitoring applications, computation time needs to be minimized for real-time performance [60]-[62]. This thesis proposed a novel algorithm for motion vector estimation based on a real-time optical flow computation to assist the detection window in vehicle counting. As opposed to the conventional approaches to estimating optical

flow, this work will focus on the real-time determination of vehicle motion detection.

1.3 Research Objectives and Organization of the Thesis

The objective of this thesis is to develop advanced image processing methodologies for estimating traffic parameters with satisfactory accuracy. The developed methodologies consist of camera calibration, single Gaussian background modeling and foreground segmentation, shadow suppression, vehicle detection and tracking, and optical-flow-based turn ratio measurement. For traffic monitoring, the background image of a traffic scene is first generated in real-time from image sequences. Accordingly, the moving vehicle will be segmented by background removal. Next, shadow suppression method will be studied to improve the accuracy of foreground segmentation. Vehicle detection algorithm will be developed to detect each moving vehicle and initiates a tracking process. Analyzing the detection and tracking result allow us to estimate the useful traffic parameters, such as traffic flow, vehicle speed and traffic density. The accuracy of estimating vehicle speed depends on not only the vehicle tracking results and but also on camera calibration. Camera calibration is an important step in traffic image analysis. Moreover, a new procedure will be designed to measure vehicle turn ratio at an intersection. Optical flow method is used to check where the vehicle comes from and to enable the detection process for the moving vehicle that moving in a specific direction.

The material in the thesis is organized according to the methodologies used in the traffic parameter estimation. A simplified overview is shown in Fig. 1-1. In Chapter 2, we present a novel calibration method for a PTZ camera overlooking a traffic scene. A background estimation method for an image-based traffic monitoring system using a single Gaussian scheme is presented in Chapter 3. In Chapter 4, a shadow-region-based statistical nonparametric method has been developed to detect the shadow pixels in traffic image frames.

In Chapter 5, we present a design of automatic contour initialization and image tracking of multiple vehicles. Furthermore, a method is proposed to automatically estimate the turn ratio at an intersection by using techniques of detection window as well as optical flow calculation. Chapter 6 shows the experiment of traffic parameter estimation. Finally, conclusions and recommendations for further research are provided in Chapter 7.

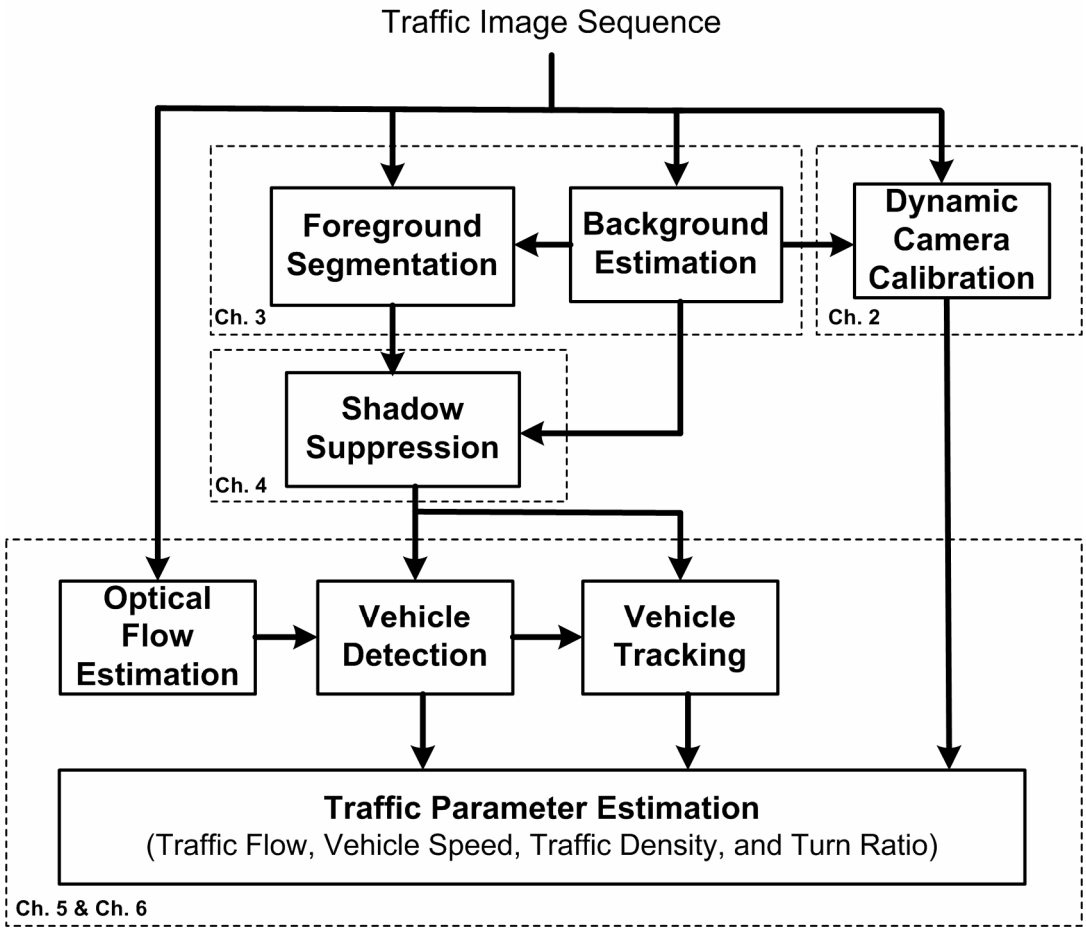


Fig. 1-1. Structure of the thesis.

Chapter 2

Dynamic Camera Calibration

2.1 Introduction

Current camera calibration techniques for traffic monitoring use more than one set of parallel lines and other known information, such as the tilt and height of the camera, to obtain the camera parameters [20]-[22]. It is easy to select or prepare enough sets of parallel lines for camera calibration in general applications. However, there is frequently only one set of parallel lines on a road. To obtain the camera parameters with reduced known information, it is necessary to generate a second line perpendicular to the existing lines for extracting the camera parameters in a traffic scene. Wang used a set of parallel lanes and a special line perpendicular to the lanes to calibrate the camera [15]. The drawback is that it is impractical to establish an unnecessary line on a road. Schoepflin and Dailey employed the bottom edges of vehicles to obtain the other sets of parallel lines for camera calibration [23]. However, the trajectories of moving vehicles degrade the extracted parallel lines. In this chapter, we present a method that uses a set of parallel lines in combination with the known height of the installed camera to calibrate the camera.

The rest of this chapter is organized as follows. Section 2.2 presents the derivation of camera calibration equations for the focal length, the pan angle, as well as the tilt angle. Section 2.3 describes image processing algorithms for lane-marking detection. Synthetic sensitivity analysis and experimental results of camera parameter estimation are presented in Section 2.4. Section 2.5 summarizes the contribution of this design. The detailed derivation of

focal length equation is presented in Appendix A. Appendix B describes the conversion between pixel coordinates and world coordinates.

2.2 Focal Length Equation

The objective of camera calibration is to determine all the required parameters for estimating the world coordinates from the pixel coordinates (u, v) of a given point in an image frame. In the following presentation, it is assumed that the change of camera height and intrinsic parameters, except for the focal length, are negligible as the camera view changes. These parameters can be considered as fixed in image-based traffic applications and calibrated only once during PTZ camera installation. A method for computing the changeable camera parameters, including the focal length, the pan and tilt angles, will be presented below.

Fig. 2-1 illustrates three coordinate systems utilized in the derivation: the world coordinate system (X, Y, Z), the camera coordinate system (X_c, Y_c, Z_c), and the camera-shift coordinate system (U, V, W). Fig. 2-1(a) depicts the top view of the ground plane in the world coordinate system, lines L_1, L_2 and L_3 represent parallel lane markings, and point O is the

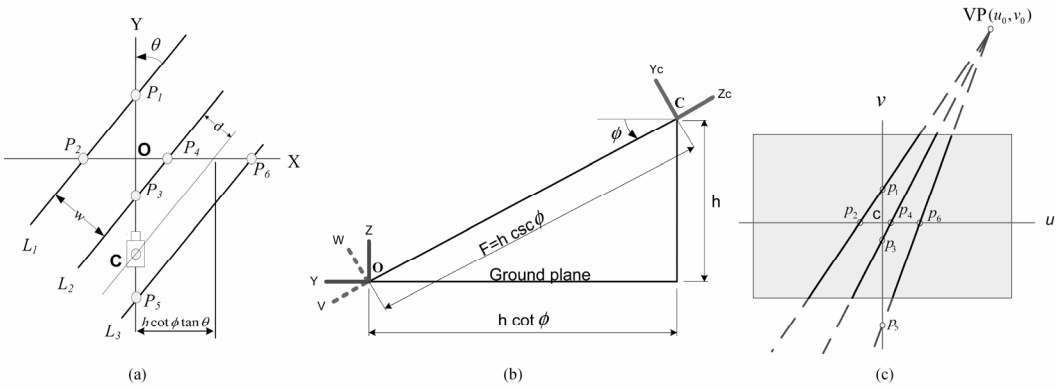


Fig. 2-1. Coordinate systems used in the PTZ camera calibration. (a) Top view of road map on world coordinate system. (b) Side view of camera setup and its coordinate systems used in calibration. (c) Road schematics used in the pixel-based coordinate system.

origin of the world coordinate system on the road plane. The pan angle θ is defined by the angle between *Y*-axis and lane markings, f is the focal length, and w is the width between parallel lanes. The symbol d denotes a shift distance, which is a perpendicular distance between the projection of the principle point of camera and L_3. Fig. 2-1(b) depicts the side view of the road scene, which is used to describe the geometric relationship between the ground plane and the camera; the direction of vector $\overrightarrow{CO}$ is perpendicular to the image plane. In Fig. 2-1(b), ϕ is the tilt angle of the camera, h is the installed camera height, and *F* is the length of vector $\overrightarrow{CO}$. In this chapter, the counterclockwise rotation is positive in expressing the sign of angles. The camera-shift coordinate system can be obtained by rotating the world coordinate system at an angle ϕ around the *X*-axis. The relationship between the camera-shift coordinate frame and the world coordinate frame is given by:

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \tag{2.1}$$

By shifting the camera-shift coordinate frame from point *O* to point *C* along the vector $\overrightarrow{OC}$ and inversing the *V*-axis of the camera shift coordinate frame, one can obtain the camera coordinate system. The camera coordinates of any point on the road plane (where *Z* equals to 0) can be expressed as a function of world coordinates via a coordinate transformation between the camera-shift coordinate frame and the world coordinate frame:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} U \\ W \\ -V-F \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \sin\phi \\ 0 & -\cos\phi \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix}. \tag{2.2}$$

As given by the pin-hole camera model [22], any point in camera coordinates has a perspective projection on the image plane. The relationship between pixel coordinates and camera coordinates can be written as:

$$u = -f\frac{X_c}{Z_c} = -f\frac{X}{-Y\cos\phi - F}, \tag{2.3}$$

$$v = -f\frac{Y_c}{Z_c} = -f\frac{Y\sin\phi}{-Y\cos\phi - F}. \tag{2.4}$$

The pixel coordinate system is shown in Fig. 2-1(c); the rectangular region represents the sensing area of the image sensor. Solid lines represent the lane markings that can be observed by the camera. Dashed lines denote the lane markings that are out of the field of view of the camera and cannot be observed. The parallel lines in Fig. 2-1(a) are projected onto a set of lines in Fig. 2-1(c) that intersect at a point known as vanishing point *(VP)*. The vanishing point lies at a position where *Y* coordinate of (*X*, *Y*, *Z*) approaches infinite. The coordinate (u_0, v_0) of *VP* is given by

$$u_0 = \lim_{Y->\infty} u = \lim_{Y->\infty}(-f\frac{X}{-Y\cos\phi - F}) = \lim_{Y->\infty}(-f\frac{Y\tan\theta}{-Y\cos\phi - F}) = f\tan\theta\sec\phi, \tag{2.5}$$

$$v_0 = \lim_{Y->\infty} v = \lim_{Y->\infty}(-f\frac{Y\sin\phi}{-Y\cos\phi - F}) = f\tan\phi. \tag{2.6}$$

In this study, we propose to use parallel lane markings to establish a geometric relationship between a road plane and its camera view. As shown in Fig. 2-1(a), L_1, L_2 and L_3 intersect *X*-axis and *Y*-axis at six points; these points are denoted by P_1–P_6, respectively. From the perspective model, the corresponding coordinates of these points in image plane can be obtained. Through geometric analysis, deriving the focal length equation expressed below is a straightforward process:

$$am^2 + bm + c = 0, \tag{2.7}$$

where m is f^2 and Table 2-1 summarizes the variables used in (2.7). The detailed derivation is presented in Appendix A.

Table 2-1 List of Variables for Focal Length Equation

Variables	Meaning and Description
m	f^2
a	$B-\beta_2^{\,2}$
b	$B(u_0^{\,2}+v_0^{\,2})-\beta_1^{\,2}-v_0^{\,2}\beta_2^{\,2}$
c	$-v_0^{\,2}\beta_1^{\,2}$
B	$\beta_1^{\,2}\left(\frac{qw}{hu_4}\right)$
β_1	$\frac{\alpha v_1 v_3 - v_1 v_3}{\alpha v_1 - \alpha v_3}$
β_2	$\beta_2 = \frac{\alpha v_3 - v_1}{\alpha v_1 - \alpha v_3}$
q	$q = \frac{1}{1-\alpha}$
α	$\frac{u_2}{u_4}$
u_2	u coordinate of point p_2
u_4	u coordinate of point p_4
v_1	v coordinate of point p_1
v_3	v coordinate of point p_3

The solution f^2 of (2.7) must be positive. Accordingly, the focal length f is

$$f = \sqrt{m}. \tag{2.8}$$

Using (2.6), the tilt angle is given by

$$\phi = \tan^{-1}\frac{v_0}{f}. \tag{2.9}$$

From (2.5), the pan angle is expressed as

$$\theta = \tan^{-1}\frac{u_0}{f\sec\phi}. \tag{2.10}$$

If (2.7) has two positive roots, then the meaningful solution will be the one that satisfies (A.24). Using the camera parameters, one can transform the pixel coordinates into their corresponding world coordinates (*X,Y,0*) [63], the detailed procedure is described in Appendix B.

2.3 Detection of Parallel lane Markings

In this section, an image processing procedure is proposed to automatically detect parallel lane markings in road imagery. The complete procedure consists of background segmentation, edge extraction, erosion, dilation, labeling, and lane marking analysis. Fig. 2-2 shows the functional block diagram of the image processing procedure.

2.3.1 Edge Detection

From the extracted background image, the edges of lane markings can be obtained adopting an intensity gradient method [64]. The detected edges of traffic lane markings are depicted in Fig. 2- 3(a). To verify the detection performance, we examine the background image together with the detected edges, as shown in Fig. 2- 3(b). Note that two edges appear on both sides of lane markings. In this design, only the right edges are selected for further calculation. A filter is designed to remove all adjacent edge pixels except those of the right-most edge:

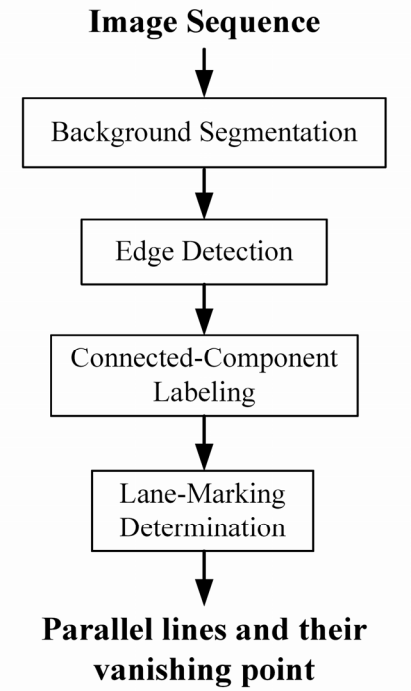


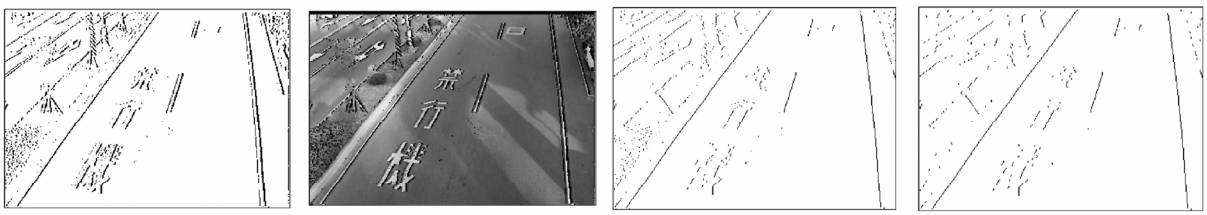
Fig. 2-2. System architecture of image-based lane-marking determination.

$$P(u,v)=0 \quad \text{if} \quad \sum_{j=u+1}^{u+5} P(j,v) \geq 1, \tag{2.11}$$

where $P(u,v)$ is the binary value at (u,v) in the edge map. Fig. 2-3(c) depicts the filtered result. The left edges of lane markings are removed as expected. An erosion operation is then employed to remove *salt-and-pepper* noise and shrink the detected edge [64]. Next, a dilation operation is applied to reconnect discontinuous features, which belong to a same object [65]**.** Fig. 2-3(d) shows the final result of edge detection. It is clear that the salt-and- pepper noise is removed and the extracted edges are ready for lane-marking analysis.

2.3.2 Connected-Component Labeling and Lane-Marking Determination

As depicted in Fig. 2-3(d), lane-marking segments are longer than the features generated by other object such as trees, bushes, guideposts, etc. Using a connected-component labeling operation [66], one can classify and label the pixels that are linked together. Fig. 2-4(a) shows the labeling result of the binary image of Fig. 2-3(d). The count (length) of connected pixels can be used to determine whether the connected pixels are features of a lane marking or not. Only those with larger count are preserved; the rest will be removed. The result of this operation is illustrated in Fig. 2-4(b). On a multi-lane road, the lane markings of the road edges are normally indicated by solid lines, while the lane divider lines are marked by broken lines. Based on this premise, labeled segments that have the first and the second largest number are considered as the sides of a multi-lane road. They are termed as *sidelines*. Each



(a) (b) (c) (d)

Fig. 2-3. Edge maps of background image. (a) Edge map. (b) Background image and its associated edge map. (c) Right-side edge map. (d) Denoised edge map.

sideline will then be represented by a linear polynomial equation:

$$y = \lambda x + \rho, \tag{2.12}$$

where λ and ρ are real numbers. One can use a least-square approximation to obtain λ and ρ. Accordingly, the intersection of sidelines can be computed. It gives us the vanishing point of parallel lane markings.

Other segments are similarly processed to obtain their first-degree polynomial equations, as plotted in Fig. 2-4(c). Next, these lines are checked whether they parallel the sidelines in the real world. If a straight line is parallel with the sidelines, then the intersection of the line with the sidelines needs to be located within a vanishing-point region (Vr):

$$Vr = \{(u,v) : \left|(u-u_0)^2 + (v-v_0)^2\right| \le 13\}, \tag{2.13}$$

where (u_0, v_0) is the vanishing point. Most lines, which are not parallel with the sidelines, will be removed by this intersection discrimination. As shown in Fig. 2-4(d), only those lines satisfying (2.13) are reserved.

In order to correctly locate all the lane markings on the road, the disconnected segments, which belong to a broken lane-divider line, must be merged into a line for obtaining a correct least-square linear representation. A criterion has been developed to find those lines which are near to each other. Fig. 2-5(a) shows the result after merging such lines in of Fig. 2-4(d). As

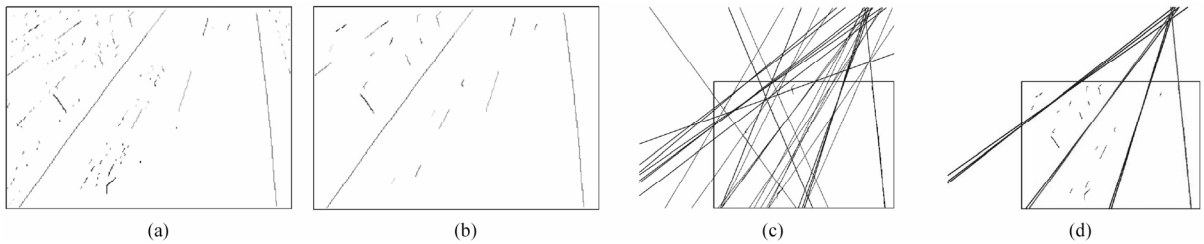


Fig. 2-4. Linear approximation of lane markings. (a) Labeled feature map. (b) Labeled segment with larger count are kept. (c) Linear approximations map. (d) The lines which intersect the sidelines and locate within a vanishing-point region are reserved.

shown in Fig. 2-5(a), although with reduced line numbers, extra lines still might exist in the image. Only the lane-divider lines that lie inside the sidelines need to be kept; others must be removed as well. Exploiting the assumption that each traffic lane is of the same width on the road, one can apply a virtual horizontal line to intersect each candidate line to obtain its position information in the image plane. As shown in Fig. 2-5(b), circles are used to represent the positions of the candidate lines and star symbols are used to represent the position of sidelines that have already been found. As shown in Fig. 2-5(b), the line with a circle lies in the center of two stars will be the lane-divider line. Fig. 2-5(c) shows the detected lane markings and their vanishing point. Fig. 2-5(d) illustrates the background image together with the detected lane markings. This lane-marking detection algorithm is computationally efficient compared with popular Hough transform approaches. A video clip of the image processing steps for finding parallel lane markings can be found at http://isci.cn.nctu.edu.tw/video/JCTai/Lane_detection.mpg.

2.4 Experimental Results

To demonstrate the performance of the proposed calibration algorithm, we first use synthetic traffic data to carry out a sensitivity analysis and then validate the calibration results using actual traffic images.

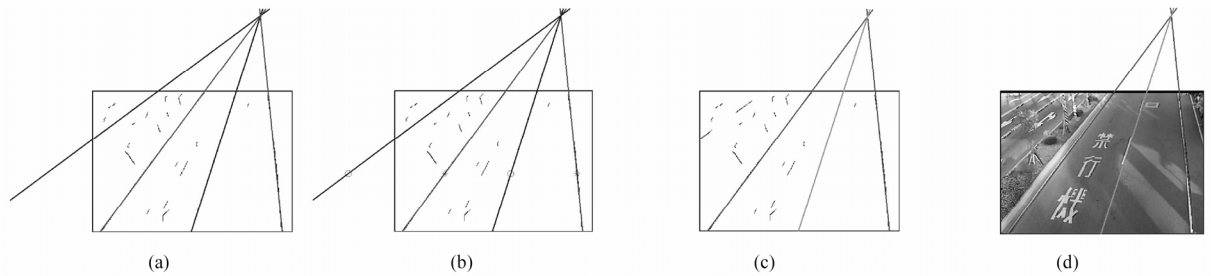


Fig. 2-5. Parallel-lane markings and their vanishing point. (a) Parallel-line map. (b) Location map of parallel lines. (c) Parallel-lane markings map. (d) Background image and its associated parallel-lane markings.

2.4.1 Sensitivity Analysis

In actual applications, there might be intrinsic or extrinsic errors of camera calibration that cause measurement error. For example, the principle point might vary with zooming [10], making pixel coordinates incorrect. Radial distortion also affects the accuracy of parallel lane detection. Tilt or pan operations will change the height of the image sensor and cause errors in the focal length estimation. In order to assure the robustness of the proposed calibration method, we present a sensitivity analysis on the focal length estimation using synthetic data containing intrinsic and extrinsic errors.

Fig. 2-6(a) illustrates the synthetic traffic scene with two parallel lanes. The circle in the figure represents the camera. The camera view of Fig. 2-6(a) is constructed according to (2.3) and (2.4); the result is shown in the rectangle region of Fig. 2-6(b). Three lines of image view intersect one another at the vanishing point, which is denoted by a cross in Fig. 2-6(b). The camera parameters–such as tilt angle, pan angle, and focal length–are calculated from the synthetic data using the calibration method described in Section 2.2. In the simulation, the tilt angle is changed from 30 to 60 degrees and the pan angle is changed from -20 to 20 degrees. These angles reflect most situations in actual ITMS applications. To emulate the effect caused by radial distortion or incorrect principle-point position, the vanishing point is shifted two pixels upward, to the right, and diagonally. New parallel lines are established in accordance with the new vanishing point and the intersections of the original lines and the u-axis. Using (2.7), focal length is calculated in accordance with these new parallel lines. In the simulation, the height of the camera is set to 7.05m and the focal length is set to 430 pixels. The maximum error rates calculated for the condition of translational error in, respectively, horizontal, vertical, and diagonal directions are presented in simulations 1, 2, and 3 of Table 2-2. The absolute error rates of focal length estimation are within 4.1%. Furthermore, the result reveals that the focal length estimation is more sensitive to vertical than horizontal

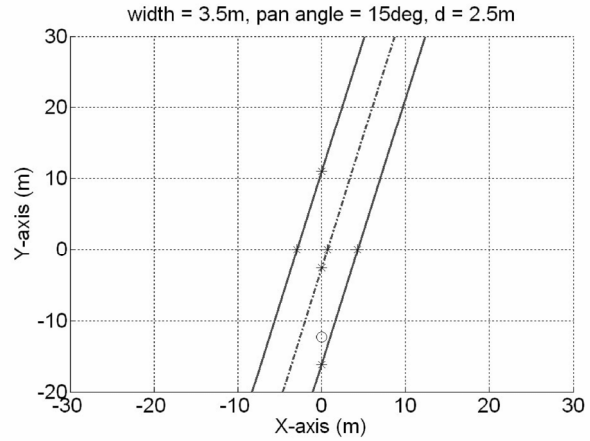
translational error.

It is observed from the sensitivity analysis that the tilt angle estimation is also more sensitive to vertical translational error. The absolute error rates of the tilt angle are within 4.1%. Because the estimated parameters will be employed to estimate the position of vehicle in ITMS, the error rates of the position in the image frame are calculated accordingly. The absolute error rates of vertical position are within 0.7% at pixel coordinate (150, 120). To investigate the effect of inaccurate camera height, we introduced a height error of -0.02m into the simulation. Traffic view is generated according to the true height (7.05m); the focal length is then estimated using the inaccurate height data. Simulation 4 of Table 2-2 shows

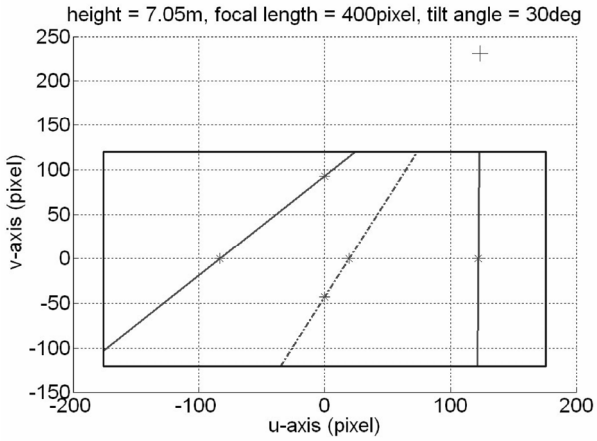
Table 2-2 Maximum Error Rates of Focal Length, Tilt Angle and Vertical Position under Different Simulated Error

Simulation index	Focal length (pixel)	Height (m)	Simulated error			Maximum error rates *1		
			du (pixel)	dv (pixel)	dh (m)	Focal length	Tilt angle	Vertical position*2
1	430	7.05	2	0	0	0.8%	0.7%	0.4%
2	430	7.05	0	2	0	3.3%	3.5%	0.4%
3	430	7.05	2	2	0	4.1%	4.1%	0.7%
4	430	7.05	0	0	-0.02	1.7%	1.4%	1.1%
5	430	7.05	2	2	-0.02	5.6%	5.5%	1.7%
6	330	7.05	2	2	-0.02	6.6%	6.7%	2.3%
7	530	7.05	2	2	-0.02	4.9%	4.7%	1.3%
8	730	7.05	2	2	-0.02	4.0%	3.8%	0.8%
9	430	6.05	2	2	-0.02	5.8%	5.7%	1.9%
10	430	8.05	2	2	-0.02	5.4%	5.3%	1.6%
11	430	10.05	2	2	-0.02	5.1%	5.1%	1.4%

*1 : In each simulation, the tilt angle is changed from 30 to 60 deg and the pan angle is changed from -20 to 20 deg.
*2 : measurement at pixel coordinate (150,120)



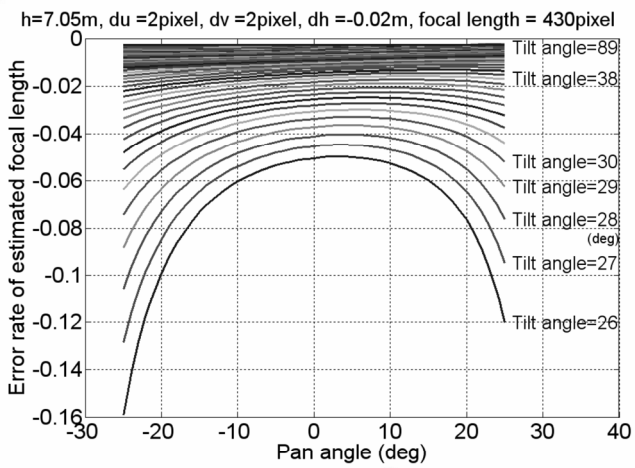
(a)



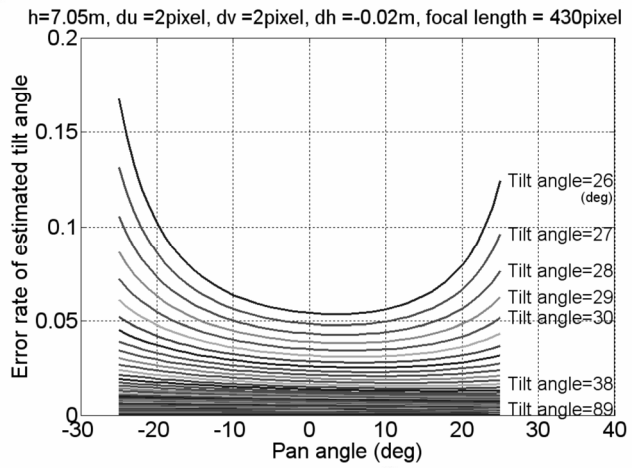
(b)

Fig. 2-6. Synthetic traffic scene for simulation. (a) Top view of a road scene. (b) The road view in image plane.

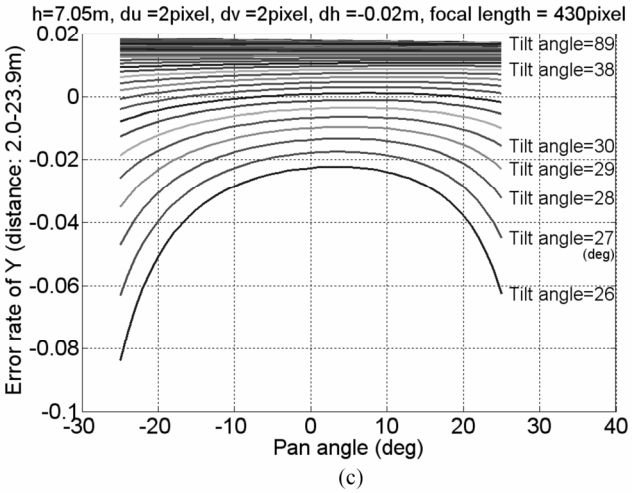
that three kinds of error rates are all within 1.7%. Detailed error-rate results are presented in Fig. 2-7 to examine estimation of focal length, tilt angle, and position, respectively. It is observed that for a tilt angle less than 30 degrees or an absolute pan angle greater than 20 degrees, the error rates will become unacceptable. This phenomenon is mainly caused by the fact that in the image plane the parallel lanes and their vanishing point will deviate more seriously due to radial distortion and incorrect principle-point position. In addition, diagonal error and height error are also simulated. The error rates are all within 5.6%, as shown in simulation 5 of Table 2-2.



(a)



(b)



(c)

Fig. 2-7. Sensitivity analysis of translation and height errors.

greater than 20 degrees, the error rates will become unacceptable. This phenomenon is mainly caused by the fact that in the image plane the parallel lanes and their vanishing point will deviate more seriously due to radial distortion and incorrect principle-point position. In

addition, diagonal error and height error are also simulated. The error rates are all within 5.6%, as shown in simulation 5 of Table 2-2.

As for the case of translational error and height error, simulations 6, 7, and 8 of Table 2-2 show the simulation results with a focal length of 330, 530, and 730 pixels, respectively. The absolute error rates of focal length and tilt angle are within 6.7% and the absolute error rates of vertical translational position are within 2.3%. The results reveal that the larger the focal length, the less the error rate.

Finally, for cases with translational error and height error, simulations 9, 10, and 11 of Table 2-2 show the simulation results of a height of 6.05m, 8.05m, and 10.05m, respectively. The absolute error rates of the focal length and the tilt angle are within 5.8%. The absolute error rates of the vertical position are within 2%. The results reveal that the higher the camera, the less the error rates. Furthermore, the effect of height change is not obvious in the test. From the synthetic analysis, the errors introduced by extrinsic and intrinsic errors are within 6.7% (focal length = 330 pixels). These error rates of position measurement are acceptable for traffic monitoring.

2.4.2 Experiments with Actual Imagery

The proposed algorithm has been tested with image sequences recorded from a main road near National Chiao Tung University. The camera used in the experiments is a SONY EVI-D31 digital camera. The image sequences were captured with a resolution of 352*240 pixels. For traffic monitoring, the camera was installed at a height of 7.03m and the width between parallel lane markings is 3.52m. In the experiments, the background image was first segmented and then used for the lane-marking detection. The vanishing point and the camera parameters–such as the focal length, tile angle, and pan angle–were calculated using (2.7), (2.9), and (2.10), respectively. To validate the estimated parameters, twelve sample features were assigned in a traffic scene for distance measurement, as shown in Fig. 2-8. To

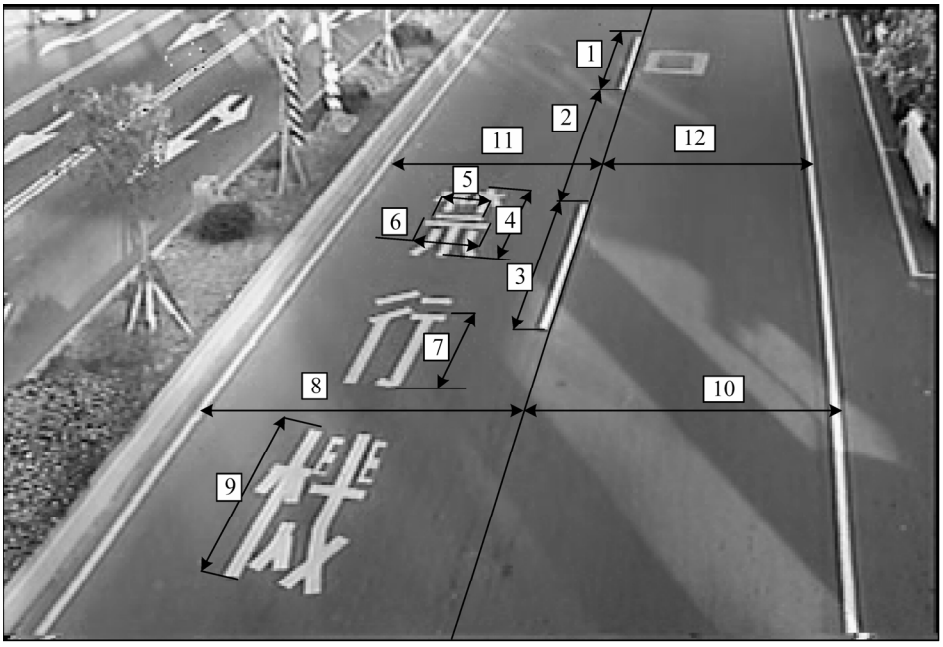


Fig. 2-8. Sample features selected for image measurement in road imagery.

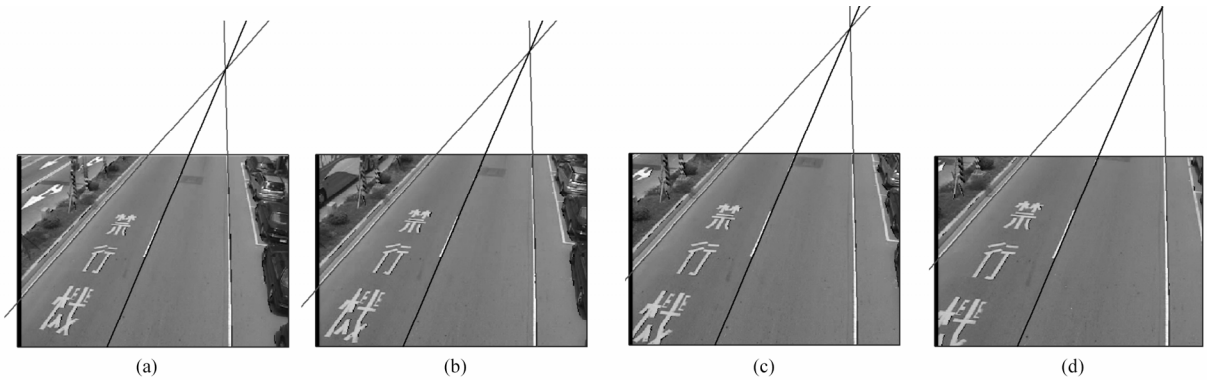


Fig. 2-9. Traffic images captured under different zoom settings. (a) Image of zoom setting A. (b) Image of zoom setting B. (c) Image of zoom setting C. (d) Image of zoom setting D.

validate the estimated parameters, twelve sample features were measured manually and compared with the estimated distances for evaluation. The estimated distances were computed based on the calibrated camera parameters.

In the first experiment, traffic images with different zoom settings (A, B, C and D) were captured to demonstrate the robustness to the radial distortion, as shown in Fig. 2-9. The principle point of the camera is practically fixed for these zoom settings. The experiment results are listed in Table 2-3. The focal lengths are estimated to be 452.91, 496.70, 542.54, and 592.36 pixels, respectively, for the four different zoom settings. Using the focal length, angle are 27.45 and 0.33 degree, respectively. The estimated mean and standard deviation of

Table 2-3 Calibration Results under Different Zoom Settings

Zoom settings		A	B	C	D
Estimated focal length (pixel)		452.91	496.70	542.54	592.36
Estimated tilt angle (deg)		27.19	27.21	27.51	27.89
Estimated pan angle (deg)		8.01	7.92	8.02	8.09
Sample feature index	Ground truth (m)	Residual error rate	Residual error rate	Residual error rate	Residual error rate
1	4	-0.96%	0.82%	*	*
2	5.8	-2.69%	-2.56%	-1.07%	-1.29%
3	4	-4.09%	-4.85%	-3.47%	-0.73%
4	2.5	-4.84%	-3.97%	-4.98%	-0.15%
5	0.75	0.72%	1.37%	-3.25%	2.07%
6	0.99	-2.84%	-3.07%	-2.90%	-1.32%
7	1.9	1.49%	-0.05%	1.78%	3.45%
8	3.52	-0.84%	-1.59%	-1.09%	-1.25%
9	2.42	-3.10%	-4.59%	*	*
10	3.52	-1.26%	-0.94%	-2.59%	-0.05%
11	3.52	-1.75%	-2.26%	-1.28%	-1.85%
12	3.52	-1.42%	-1.47%	-1.11%	-1.39%
Abs. mean		2.17%	2.34%	2.39%	1.29%
Abs. standard deviation		1.29%	1.49%	1.20%	1.00%

* : Feature sample is not measurable in the scene image.

the pan angle are 8.01 and 0.07 degree. These experimental results show that error rates of absolute mean and standard deviation are within 2.39% and 1.49%. The proposed calibration algorithm gives satisfactory accuracy and is robust against zoom changes.

To evaluate the robustness of lane detection with respect to variation in environmental illumination, we took images at various hours of a sunny day. Six sets of image sequences are presented to show different illumination conditions, as shown in Fig. 2-10. The intensity

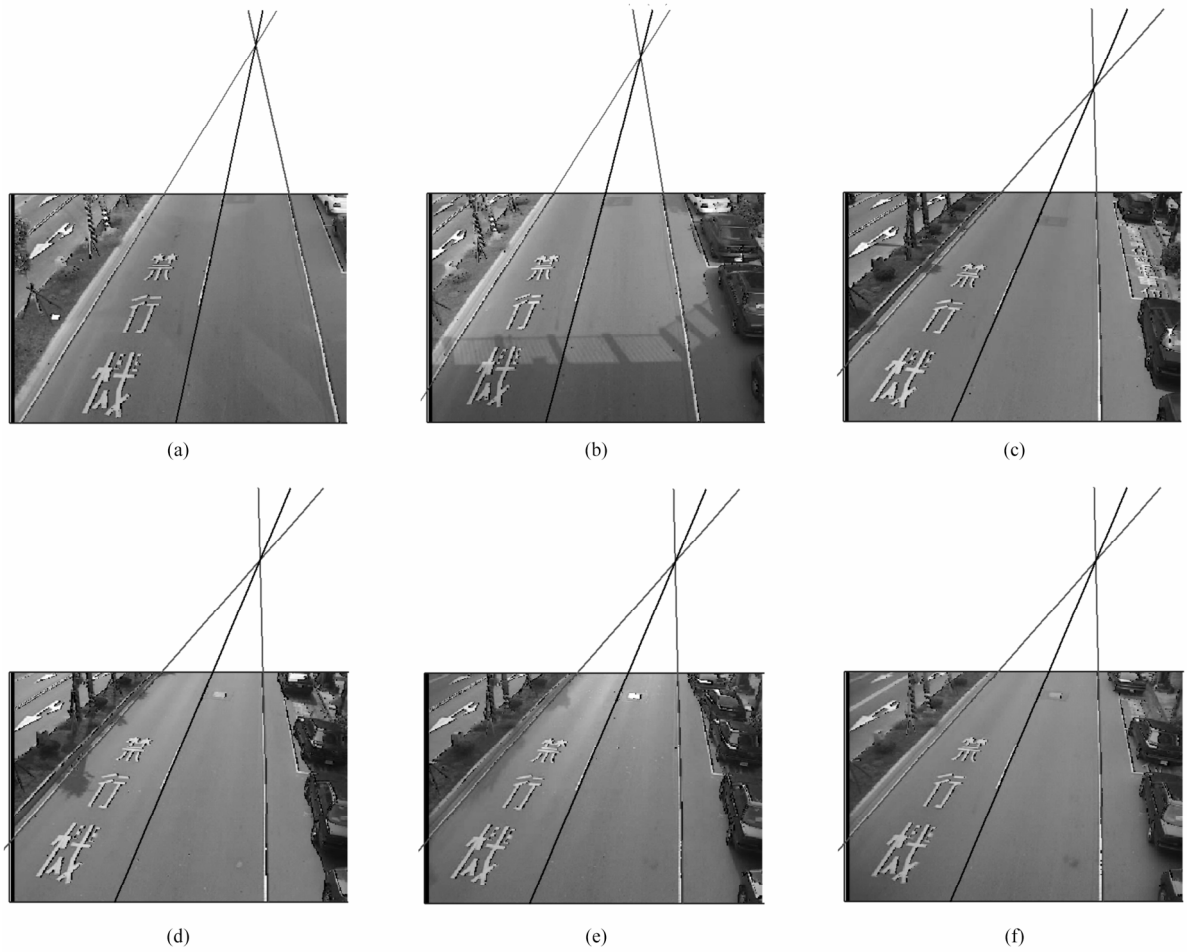


Fig. 2-10. Traffic images captured under different illumination conditions. (a) Image with weak shadow. (b) Image with strong shadow. (c) Image under bright illumination. (d) Image under soft illumination. (e) Image captured at sunset. (f) Image under darker illumination.

values of lane markings vary in these image frames, but the lane markings always have higher intensities than their adjacent region. The gradient can be successfully used to detect the edge of lane markings, as discussed in Section 2.3. The results reveal that the lane detection method performs satisfactorily under different lighting conditions. This robustness partly results from the fact that the SONY PTZ camera has auto exposure and backlight compensation functions to ensure that the subject remains bright even in harsh backlight conditions.

Finally, the algorithm is evaluated with a fixed zoom under different pose settings (A, B, C and D) of the PTZ camera, as shown in Fig. 2-11. Table 2-4 shows the experimental results of the estimation of feature sizes as depicted in Fig. 2-8. The mean and standard deviation of

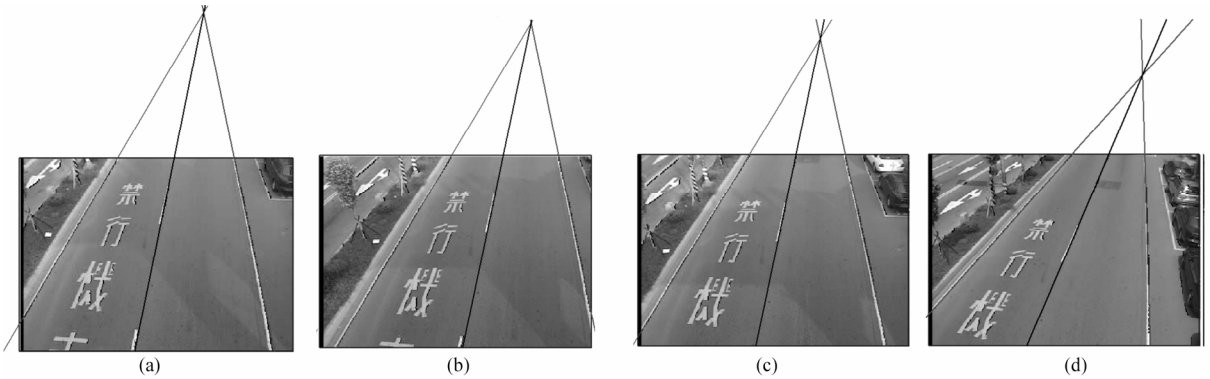


Fig. 2-11. Traffic images captured under different camera pose settings. (a) Image of pose setting A. (b) Image of pose setting B. (c) Image of pose setting C. (d) Image of pose setting D.

estimated focal lengths are 417.08 and 9.56 pixels, respectively. The mean and the standard deviation of absolute error rates among these measurements are within 2.32% and 1.58%.

The experimental results of different zoom and view settings show that the maximum calibration error of distance measurement is within 5%, which is comparable to the results achieved by [20] and [21]. However, our method offers improved autonomy and efficiency. A video clip of image processing sequence for traffic parameter estimation can be found at http://isci.cn.nctu.edu.tw/video/JCTai/Speed_detection.mpg.

2.5 Summary

`A novel algorithm has been proposed for automatic calibration of a PTZ camera overlooking a traffic scene. The proposed approach requires no manual operation to select positions of special features. It automatically uses a set of parallel lane markings and the lane width to compute the camera parameters, namely, focal length, tilt angle, and pan angle. Image processing procedures have been developed for automatically finding parallel lane markings. Subsequently, the pan and tilt angles of the camera can be obtained by using the estimated focal length. To locate the parallel lane markings, an image processing procedure has been developed. Synthetic data and actual traffic imagery have been employed to validate accuracy and robustness of the propose method.

Table 2-4 Calibration Results under Different Camera Pose Settings

Pose settings		A	B	C	D
Estimated focal length (pixel)		421.82	413.27	427.50	405.74
Estimated tilt angle (deg)		35.63	34.06	31.68	30.86
Estimated pan angle (deg)		7.95	12.18	7.36	14.38
Sample feature index	Ground truth (m)	Residual error rate	Residual error rate	Residual error rate	Residual error rate
1	4	*	*	*	-1.68%
2	5.8	*	4.94%	4.33%	4.94%
3	4	0.00%	-0.98%	-2.38%	2.15%
4	2.5	-1.75%	0.38%	2.94%	-1.31%
5	0.75	-3.01%	1.61%	-3.99%	-0.25%
6	0.99	-2.99%	-0.92%	-4.85%	-3.40%
7	1.9	4.35%	2.93%	4.57%	3.23%
8	3.52	-0.78%	-3.11%	-1.20%	-2.55%
9	2.42	-1.45%	1.48%	-0.52%	2.77%
10	3.52	-1.59%	-2.76%	-0.54%	-2.84%
11	3.52	-0.22%	-3.00%	-1.90%	-1.64%
12	3.52	-1.10%	-4.10%	-1.34%	-2.88%
Abs. mean		1.68%	2.04%	2.32%	2.18%
Abs. standard deviation		1.27%	1.24%	1.58%	1.08%

* : Feature sample is not measurable in the scene image.

Chapter 3

Background Generation and Foreground Segmentation

3.1 Introduction

An image is considered a background image if there are no moving objects in the image. Hence, one can obtain a background image by merely capturing an image with no moving objects. However, in a heavily trafficked road, it is difficult to find a traffic scene with no moving vehicles. Researchers have devoted themselves to developing methods for extracting the background image from traffic image sequences. Most methods exploit the concept of probability for background segmentation; examples are the GMM-based method and the histogram method. However, several directions deserve further study for improving the quality of background segmentation. For instance, the GMM-based method is sensitive to slow moving object, which will be misrecognized as a background object. In contrast, the histogram is sensitive to the sensing noise. Sensing noise degrades the quality of background segmentation. As an alternative to these two methods, a background estimation method using a single Gaussian scheme is presented in this chapter. A group-based histogram (GBH) algorithm is proposed to build the Gaussian background model of each pixel from traffic image sequences. The method is efficient and uninterrupted by both slow moving vehicles and sensor noise. Accordingly, moving vehicles can be segmented by background removal.

The rest of this chapter is organized as follows. Section 3.2 describes the GBH

algorithm for estimating the single Gaussian background model. Experimental results of the proposed method and several interesting examples of traffic parameter estimation are presented in Section 3.3. Section 3.4 concludes the contributions of this work.

3.2 Group-Based Histogram

In traffic imagery, the intensity of the background scene is the most frequently recorded intensity at its pixel position. The background intensity can therefore be determined by analyzing the intensity histogram. However, sensing variation and noise from image acquisition devices may result in erroneous estimation and cause a foreground object to have the maximum intensity frequency in the histogram. To solve this problem, we propose the GBH to estimate the single Gaussian model of static background. In the first stage, the group-based frequency of each intensity level is generated from the incoming intensity of an image sequence. The intensity level that has the maximum group-based frequency is treated as the mean of the single Gaussian background model. The standard deviation of the Gaussian model can be computed by using the estimated mean and the histogram. The detail procedure is explained in the following paragraphs.

3.2.1 Single Gaussian Background Modeling and GBH

In GBH, a cumulative frequency is generated from the frequency of each individual intensity level as well as those of its neighboring levels in the histogram. In other words, the frequency of each intensity level and the frequency of its neighboring levels are summed to form the group-based histogram. First note that the frequency or probability of a conventional histogram is updated by using a single intensity, while the probability of GMM is constructed from a group of intensities. Thus, the GMM is more suited than a simple histogram for representing intensity distribution of the background image. GBH possesses similar merits to GMM because it takes the variation of pixel intensity into account. This operation effectively handles the problem of sensor noises. Further, the GBH algorithm gives a reliable Gaussian

background model by using several image frames.

The GBH can be generated by using an average filter of width of $2w+1$ where w is the half-width of an average window. The output $n^*_{u,v}(l)$ of the average filter at level l can be expressed as

$$n^*_{u,v}(l) = \sum_{r=-w}^{w} n_{u,v}(l+r); \qquad 0 \le l+r \le (L-1), \tag{3.1}$$

where $n_{u,v}(l+r)$ is the count of pixel with intensity $l+r$ at location (u,v), and L is the number of intensity levels. In GBH, the maximum probability density $p^*_{u,v}$ of a pixel at location (u,v) over the recorded image frames can be expressed as

$$p^*_{u,v} = \frac{\max_{0 \le l \le L-1} \{n^*_{u,v}(l)\}}{N^*}, \tag{3.2}$$

where N^* is the total frequency of the GBH. It is clear that if the window width of the filter is smaller than a certain value, the maximum peak of the GBH will be located at a position closer to the center of a Gaussian model than the original histogram. This is because the filter smooths the histogram curve. Thus, the intensity that has the maximum frequency in the GBH can be treated as the mean intensity $\mu_{u,v}$ of the background model:

$$\mu_{u,v} = \arg\max_l \{n^*_{u,v}(l)\}. \tag{3.3}$$

A smaller window width can save computation time for building up the GBH, while a larger width can produce a smoother GBH. For further discussion on the determination of window width, an example given below employs 13 Gaussians that were generated by using a Gaussian random number generator. The means of all Gaussians were chosen to be 205 and the standard deviations varied from 3 to 15. The histogram and the GBH generated with different widths were used to estimate the mean of each Gaussian; the error rates are shown in

Table 3-1. The error rates that fall within ±2% are highlighted in the table. The results show that the estimation of the proposed method is superior to that of a conventional histogram. One can conclude from the simulation results that a larger window width of an average filter will be needed for high-accuracy performance as the standard deviation increases.

Keeping the error rate of mean estimation within ±2%, and using the simulation result, the width can be determined as follows:

$$w = \begin{cases} 3, & 3 \le \sigma_i \le 7 \\ 5, & 8 \le \sigma_i \le 10 \\ 7, & 11 \le \sigma_i \le 15 \end{cases}. \tag{3.4}$$

where σ_i is the standard deviation of the original Gaussian.

In the following derivation of the Gaussian model, $C(i)$ is used for recording the GBH frequency of intensity i. The mean intensity $\mu_{u,v}$ can be obtained from the maximum-value counter. The system does not process all counters when a new intensity l is captured, because the new intensity only increases the adjacent counters of counter l. The proposed

Table 3-1 Estimation Error Rate of Gaussian Mean using Histogram and GBH.

Standard deviation	3	4	5	6	7	8	9	10	11	12	13	14	15
	-1.5%	-1.5%	-2.0%	-2.4%	-2.4%	-2.9%	-3.4%	-2.9%	-2.9%	-4.4%	-2.9%	-4.9%	-4.4%
width w	Estimation result of the proposed GBH												
1	0.5%	1.0%	0.0%	-0.5%	-0.5%	2.0%	-3.4%	-2.4%	-2.9%	-1.5%	-3.4%	0.5%	1.0%
2	0.5%	0.0%	0.0%	-1.5%	-1.0%	1.0%	-2.4%	-2.0%	-2.4%	-2.0%	-3.9%	1.0%	1.5%
3	0.0%	0.0%	-0.5%	-1.0%	0.5%	-1.5%	-2.4%	2.0%	2.0%	-2.4%	-1.5%	1.5%	-2.9%
4	0.5%	0.5%	0.0%	-0.5%	-0.5%	0.5%	-1.5%	-1.0%	-1.5%	-2.4%	-1.0%	2.4%	-2.4%
5	0.5%	0.5%	-0.5%	0.0%	0.0%	-0.5%	-1.0%	-0.5%	-0.5%	-2.0%	-1.5%	0.5%	0.5%
6	0.5%	0.0%	-0.5%	-0.5%	-0.5%	-1.0%	-0.5%	0.0%	-0.5%	-1.5%	-1.5%	0.0%	-1.5%
7	0.5%	0.5%	-0.5%	0.5%	0.0%	-0.5%	0.0%	0.0%	0.0%	-0.5%	-1.5%	1.0%	-1.0%

algorithm for obtaining the mean of Gaussian model is summarized below.

step 1. Record the current intensity l of a pixel.

step 2. Iterate, for $i = l - w, \cdots, \min(l + w, 255)$:

$C(i) = C(i) + 1$;

If $C(i) \geq C_\mu$, then

$\mu_{u,v} = i$;

$C_\mu = C(i)$;

endif

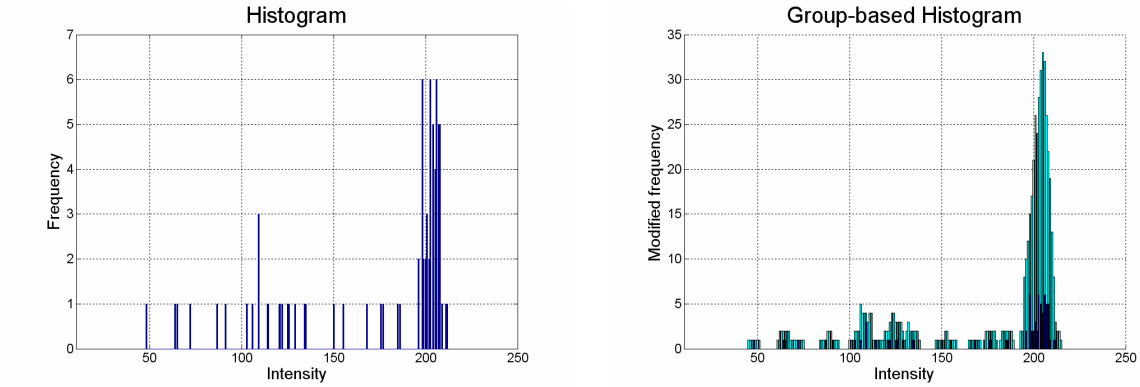
where μ is the estimated mean intensity of a background pixel and C_μ is the GBH frequency of intensity μ.

After the center of the background model is found, the variance can be computed as follows:

$$\sigma_{u,v}^{\ 2} = \frac{1}{\sum_{x=\mu_{u,v}-3\sigma'}^{\mu_{u,v}+3\sigma'} n_{u,v}(x)} \times \sum_{x=\mu_{u,v}-3\sigma'}^{\mu_{u,v}+3\sigma'} (x - \mu_{u,v})^2 n_{u,v}(x). \tag{3.5}$$

where σ' is the maximum standard deviation of the Gaussian background models (the value can be experimentally obtained by analyzing the background models from image sequences offline).

Fig. 3-1(a) shows an example of intensity histogram of a pixel in a traffic image sequence. It is clear from the figure that the background intensity level is distributed in the range from 195 to 215. Further analysis of the sampled data in the background-intensity region was performed by using MINITAB Statistical Software (release 13.32 for WINDOWS; Minitab, Inc, State College, PA). The result shows that the data can be modeled as a Gaussian [67] with a mean and standard deviation of 203.65 and 3.88, respectively. However, one



(a) (b)

Fig. 3-1. Statistical analysis of pixel intensity. (a) Histogram. (b) Group-based histogram of Fig. 3-1(a).

cannot determine the center of the background model from the histogram because three intensities have the same maximum frequency. On the contrary, the background intensity can be easily found in the group-based histogram (with width $w = 3$), as shown in Fig. 3-1(b). The mean and the standard deviation of the estimated Gaussian model of Fig. 3-1(b) are 205 and 4, respectively. The error rate of the mean and the standard deviation is 0.67% and 3.17%. The results confirm that the derived probability density function satisfactorily fits the background intensities.

Note that the proposed GBH only uses addition and comparison to estimate the mean of background pixels; on the contrary, other methods such as GMM involve much complex computation, such as multiplication, sorting, and division operations. Because estimation of the mean of background model is less time-consuming than the estimation of standard deviation, the mean intensity, the GBH, and the histogram are updated at every sample interval. The standard deviation is updated every 30 (or more) frames. To reduce the computation burden, the estimation of standard deviation is completed step-by-step in each sample interval. In traffic monitoring applications, the GBH, the histogram, and the background model are renewed every 15 minute to cope with illumination variations. The

computational load of the proposed method is significantly lower than the methods presented in [24] and [29]. Thus, it is more suitable for the real-time requirement of visual tracking of vehicles.

3.2.2 Foreground Segmentation

In this paragraph, we present the method for detecting moving foreground objects based on the background estimation results. The intensities of each pixel obtained from an image sequence generate a background Gaussian as well as many foreground distributions, as shown in Fig. 3-1(a). The historical intensities of each pixel can be divided into two groups: the static background and the moving foreground. If a foreground pixel appears, its intensity is then located outside the background Gaussian. To cope with the variation of the background intensity, a pixel will be classified as foreground if its current intensity is located outside $\pm 3\sigma_{u,v}$ from the mean $\mu_{u,v}$ of the Gaussian background model:

$$F(u,v)=\begin{cases}1 \ (moving\ object) & if\ |I_{u,v}-\mu_{u,v}|>3\sigma_{u,v}\\ 0 & otherwise\end{cases}, \tag{3.6}$$

where $I_{u,v}$ are the current intensity at location (u,v), $(\mu_{u,v},\sigma_{u,v})$ represents the estimated Gaussian background model at location (u,v). An erosion operation is further employed to remove salt-and-pepper noise [64]. A traffic surveillance system then performs image measurement to monitor moving objects from the binary image.

3.3 Experimental Results

Several experiments have been carried out to validate the performance of the proposed method. A pixel level experiment is presented in Section 3.3.1 to demonstrate the effectiveness and robustness of the GBH approach. In Section 3.3.2, background images generated by using GMM and GBH are presented for performance comparison. Section 3.3.3

presents interesting experimental results of traffic parameter estimation.

3.3.1 Pixel Level Experiments

Fig. 3-2 shows the experimental results of background estimation of a pixel in a traffic image sequence as shown in Fig 3-2(b). The intensity was recorded over a fixed span of time (196 frames in 6.5 seconds). For comparison, a GMM with three Gaussian models was first constructed to model the pixel [29]. If a Gaussian matches the current pixel intensity x_t at time t, the mean c_t of Gaussian is updated by

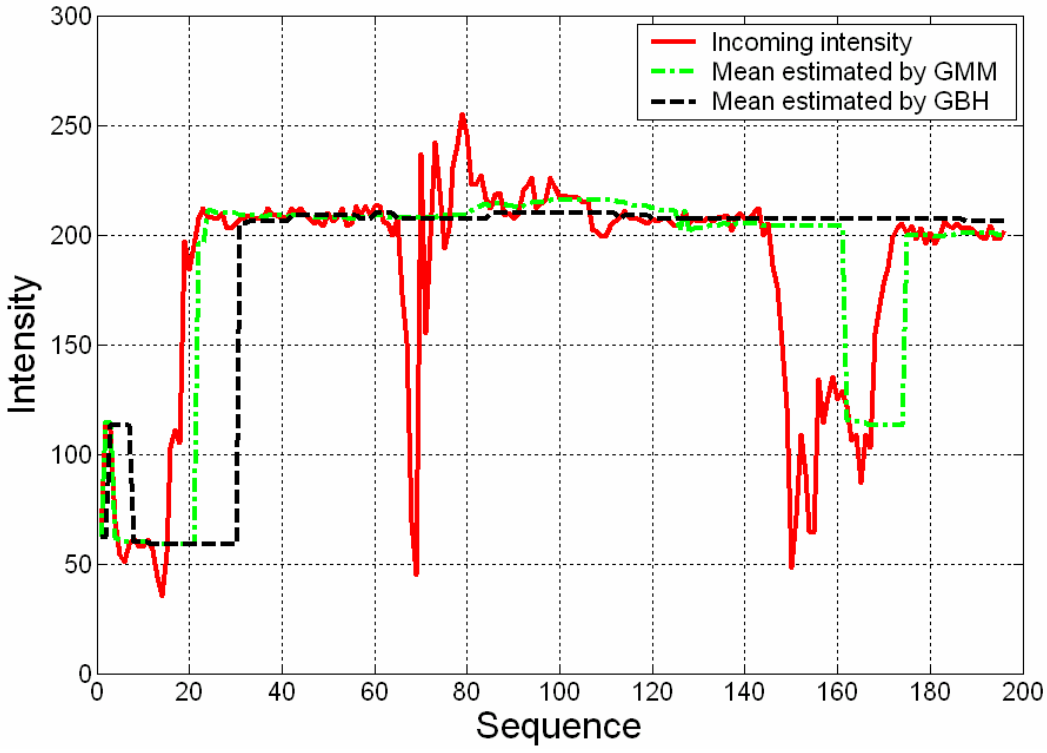
$$c_t = c_{t-1} + \beta(x_t - c_{t-1}), \tag{3.7}$$

where β is the learning rate. All weights are updated by

$$\omega_t^* = \omega_{t-1} + \beta(M_t - \omega_{t-1}), \tag{3.8}$$

where M_t is equal to 1 for a matched Gaussian; otherwise, M_t is set to 0. All the weights are represented by $\omega_t = \frac{\omega_t^*}{S}$ for normalization, where S is $\sum \omega_t^*$ and the result ω_t is equivalent to the probability of intensity based on past values. The normalized clusters are sorted in an order of decreasing weight. In this experiment, the Gaussian model that has the maximum weight is classified as the background Gaussian. The experimental result is shown in Fig. 3-2(a). In the figure, the solid line indicates the experimental data and the dashed line represents the estimation result of the GMM. In the experiment, the learning rate was selected as 0.1. One can see that the estimated mean matches with the background intensity well. However, transient appearances of slow-moving objects easily change the weight and magnitude order of GMM background estimation. For example, from sequence 150 to 162, a transient slow-moving foreground object quickly results in a matched Gaussian model and intensity 110 is erroneously chosen as the background intensity for its maximum weight. This type of erroneous estimation is caused by the fact that in GMM the foreground object that

(a)



(b)

Fig. 3-2. (a) Traffic scene with a cross at the middle left showing the position of the sample pixel. (b) Background estimation of Fig. 3-2(a) using GMM-based approach and the GBH-based approach.

moves slowly or stops for a short period of time will cause its intensity cluster to obtain the maximum weight, even though its intensities do not have maximum probability in the historical data.

The proposed GBH method was then applied to the same data for finding the background intensity of the pixel. The dashed line of Fig. 3-2(a) represents the estimation result using the GBH method; the gray level of background pixel was maintained near 205 as expected. The GBH-based method succeeds to estimate an accurate mean of background model from image sequence. The proposed method is robust to the transient appearances of slow-moving objects.

3.3.2 Background Estimation of Traffic Imagery

To examine the performance of the proposed method, we utilized twelve image frames of the traffic scene to construct the background image. In the experiments, we first convert *RGB* color information in image frames to *Y* luminance [68]:

$$Y = 0.299R + 0.587G + 0.114B. \tag{3.9}$$

Fig. 3-3 shows the original image sequence. There are four rows in the figure arranged in order from left to right. The image in the upper-left corner is the first of the sequence. For comparison, both the GMM and the proposed GBH methods were employed to determine the background image of the traffic scene.

Fig. 3-4 depicts the background estimation results using GMM. A reliable background image is obtained as depicted in the figure. However, the transient appearance of slow moving vehicles degrades the quality of the background image constructed by GMM. For example, the squared area in the lower-right region is well constructed in the fifth image; however, the background image of the portion is unreliable in the later image frames, as depicted in the

Fig. 3-3. Image sequence for background image generation.

figure. This phenomenon is caused by a vehicle that slowly passed this region, as shown in Fig. 3-3. One can use a smaller learning rate to solve this problem; however, with a small learning rate, a reliable background image can only be obtained with much more computation.

Fig. 3-5 shows the estimation results using the proposed GBH method. The estimation quality of the GBH is practically the same as the one from the GMM, as shown in Fig. 3-5. As expected, the moving foreground vehicles almost disappear beginning with the fifth background image. Furthermore, the quality of the final background image is better than that from the GMM, especially in the lower-right region, as marked in Fig. 3-5.

The processing time of this experiment is listed in Table 3-2 for comparison. The computation time of the proposed GBH algorithm is considerably smaller than that of GMM (with a reduction rate of 35%). From these results, it is clear that the performance of GBH is more suitable than the GMM for image-based traffic surveillance applications.

Fig. 3-4. Background images constructed by GMM method.

Fig. 3-5. Background images constructed by the proposed method.

Table 3-2 CPU Time of the Tested Background Estimation Algorithms.

Algorithm	GBH method	Gaussian mixture model
Time(sec)	0.037	0.057
Specification: image size= 352x240, CPU type= AMD Athlon XP2000+, RAM=512MB.		

3.3.3 Application to Traffic Flow Estimation

For investigating traffic congestion problems, the background segmentation module has been integrated into an ITMS for extracting traffic parameters. Our research team have established a real-time web video streaming system to monitor the traffic in Hsinchu Science Park [69]. The system provides an H.263 video stream to an ITMS. H.263 is suitable for digital video transmission over networks with a high compression and decompression ratio [70]. However, due to image compression and decompression, the degraded image is neither stable nor consistent for image processing of image-based applications. This phenomenon introduces extra challenges to image processing design.

In this study, video stream provided by the imaging system is employed to measure the traffic flow at a multi-lane entrance of the Science Park. The video stream in CIF format (352 x 288 pixels) is transmitted to the computer in the lab at a rate of 7 frames per second through ADSL. The system architecture of the real-time traffic monitoring system is shown in Fig. 3-6. The image system consists of three parts: image processing module, vehicle detection module, and traffic parameter extraction module. The image processing module adopts an ActiveX component to decompress the actual image from the video stream and converts the image into gray-level format. Real-life images are employed to construct a background image. Based on

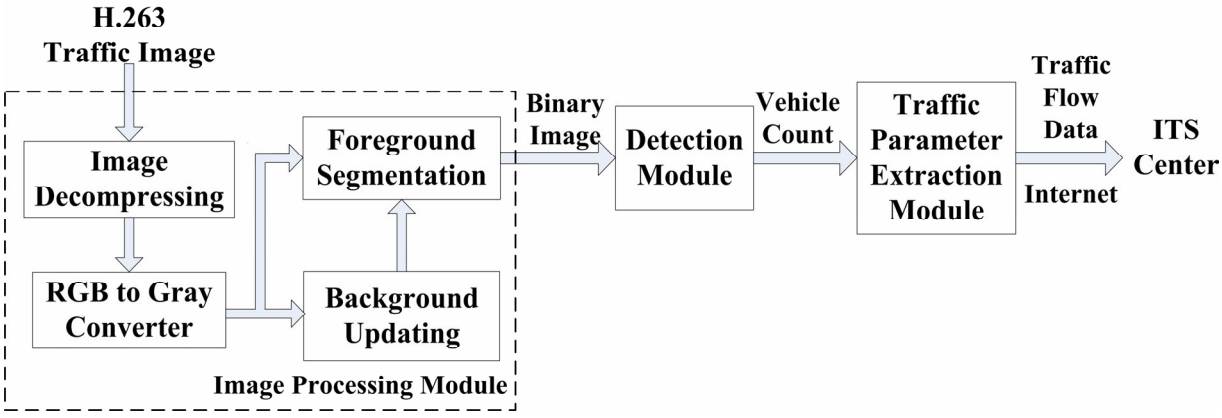


Fig. 3-6. Block diagram of the image-based traffic parameter extraction system.

the constructed background image, a binary image of the moving vehicle is obtained through foreground segmentation. The detection module employed a detection window that behaves like a loop detector to count the number of vehicles in a multi-lane road [71]. The detection window is able to check if there is a vehicle entering or leaving the window from the binary image. It simultaneously detects multiple vehicles from the binary image. Traffic parameter extraction module calculates the traffic flow data and provides information to the ITS center. Fig. 3-7 illustrates a display of the system results. The upper left part of the figure shows a real-time image and the lower right part shows the background image created from the image sequence. The upper right part depicts the binary image of moving vehicles as well as the processed results of the detection window. The detected vehicles count and the traffic flow data are displayed on the lower left portion of the figure. The estimated traffic flow is calculated through a measure of vehicle count using the equation below:

$$traffic\ flow = \frac{N_{car}}{t_{dur}}, \tag{3.10}$$

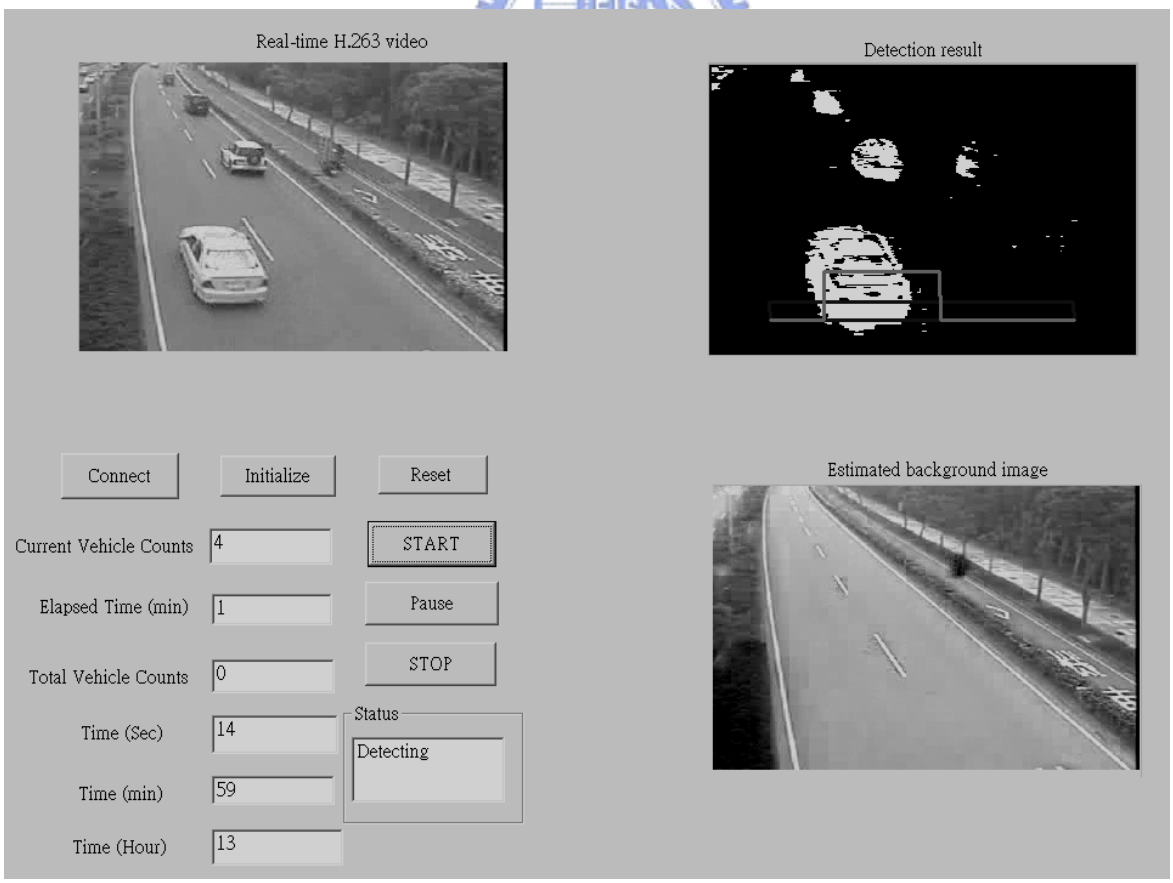


Fig. 3-7. The display of traffic flow estimation.

where N_{car} is the detected count and t_{dur} is the time duration. A video clip of experimental results can be found at: http://isci.cn.nctu.edu.tw/video/JCTai/Traffic_flow.wmv.

3.4 Summary

An algorithm of a group-based histogram is proposed to build the Gaussian background model of each pixel from traffic image sequences. This algorithm features improved robustness against transient-stop of foreground objects and sensing noise. Furthermore, the method features low computational load, and thus, meets the real-time requirements in many practical applications. The proposed method can be extended to construct a color background image to further increase the robustness during intensity analysis.

Chapter 4

Cast-Shadow Detection in Traffic Image

4.1 Introduction

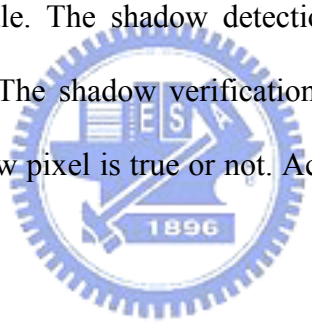
Cast-shadow suppression is an essential prerequisite for image-based traffic monitoring. Shadows of moving objects often cause serious errors in image analysis due to misclassification of shadows or moving objects. It is necessary to design a shadow suppression method to improve the accuracy of image analysis. Shadow detection can be divided into two types: shape-based approaches [33]-[34] and spectrum-based approaches [37]-[39]. In the shape-based method, sophisticated models are constructed to identify the shadow according to the object and its surrounding illumination conditions. The accuracy of shadow detection depends on the knowledge of environment conditions. Moreover, it can not meet the real-time requirement because the computation load of the spatial analysis is heavy. in contrast, spectrum-based approaches exploit color space information to find the shadow. The model is built by strenuously analyzing the shadows in image frames. Generation of the model is difficult and inefficient. Additionally, conversion between different color spaces requires much computation time and degrades the real-time performance. Based on the Lambertian assumption, RGB ratios between lit pixels and shadowed pixels can be treated as a constant in image sequences. This information leads us to the development of a RGB ratio model to detect shadow pixels in traffic imagery. The proposed approach does not require many image sequences to construct the model. Instead, the model can be easily built using a

shadow region in a single image frame. To increase the accuracy of shadow detection, two types of spatial analysis are proposed to verify the actual shadow pixels.

The following sections will focus on how the ITMS with shadow suppression is developed. Section 4.2 focuses on shadow detection. A comparison with existing methods is presented in Section 4.3. Section 4.4 gives our concluding remarks.

4.2 Cast-Shadow Detection in Traffic Image

The block diagram of the proposed shadow suppression method is presented in Fig. 5-1. The complete system consists of four modules: a background estimation module (Chapter 3), a background removal module (Chapter 3), a shadow detection module, and a shadow verification post-procession module. The shadow detection module uses an RGB spectral ratio model to identify shadows. The shadow verification module employs spatial analysis schemes to check whether a shadow pixel is true or not. Accordingly, moving objects and the cast-shadows can be separated.



4.2.1 Spectral Ratio Shadow Detection

A traffic scene is illuminated by a faraway point source (the sun) and a diffused source (the sky). The cast shadow is caused by sunlight occlusion in the scene. The distance between objects and their cast shadows is negligible in traffic scenes, compared with the distance between the light source and the objects. Thus, this type of the cast shadow is mostly an umbra or a strong shadow [35]. Shadow regions are darker than the background and their color spectrum also differs from that of the background. Since the RGB components of each pixel of the roadway differ from one another, it is impossible to use a unique model to detect the shadow of each pixel by merely using RGB color space. Therefore, color space conversion or normalization is employed to find the model. To resolve the generation of a

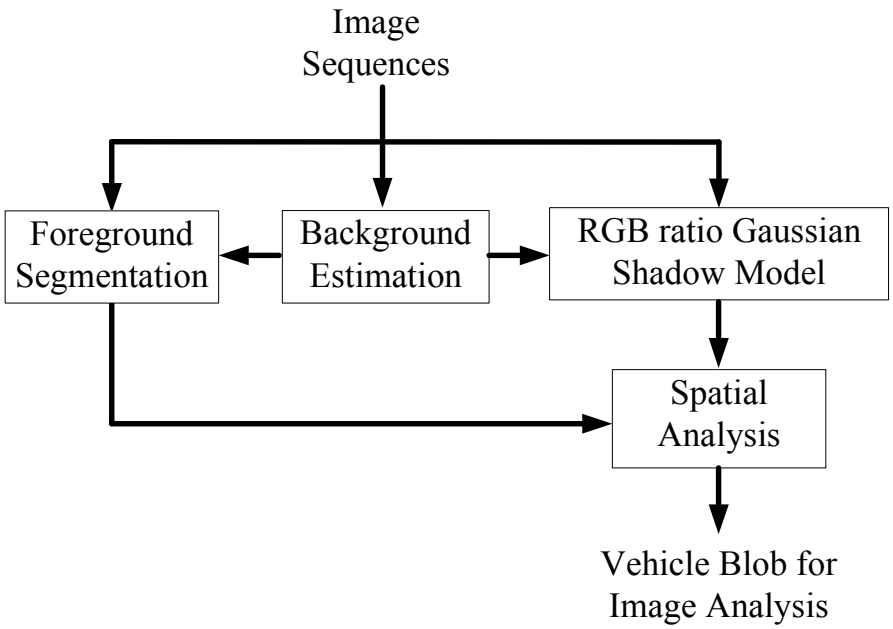


Fig. 4-1. The block diagram of the proposed shadow detection method.

shadow model, we hypothesize that the RGB ratio between lit regions and shadow regions is constant for all pixels of a traffic scene in image sequences. This hypothesis facilitates the construction of a unique model for shadow detection in an image frame. The model-built procedure can be effectively achieved by using a shadow region in the image, not from image sequences. Detailed reasoning of the hypothesis is presented in Appendix C.

According to the hypothesis, in each RGB channel, the color ratio between a lit pixel and a shadow pixel can be treated as a constant for each pixel in the traffic imagery. As a result, a pixel is classified as a shadow pixel if its RGB component satisfies

$$R_{shadow} = \alpha R_{lit} \quad \text{and} \quad G_{shadow} = \beta G_{lit} \quad \text{and} \quad B_{shadow} = \gamma B_{lit} \quad , \tag{4.1}$$

where $(R_{lit}, G_{lit}, B_{lit})$ is the RGB value of a lit pixel, $(R_{shadow}, G_{shadow}, B_{shadow})$ is the RGB value of a shadow pixel and α, β, γ is less than 1.

Based on (4.1), a shadow-region-based statistical nonparametric approach is developed to construct the ratio model for shadow detection of all pixels in the image frame. Gaussian models are exploited to represent the constant RGB-color ratios between a lit pixel and a

shadow pixel in this approach. The unique ratio model can be found by analyzing shadow samples taken from the shadow region in an image frame. To cope with the variation of the ratio, we selected the Gaussian distribution inside ±1.5σ (88.6%.) as a threshold. Thus, a shadow pixel can be determined as follows:

$$S(x,y)=\begin{cases}1 & if \left|\frac{I_R(x,y)}{\mu_R(x,y)}-r_R\right|<1.5\sigma_{rR} \ and \ \left|\frac{I_G(x,y)}{\mu_G(x,y)}-r_G\right|<1.5\sigma_{rG} \ and \ \left|\frac{I_B(x,y)}{\mu_B(x,y)}-r_B\right|<1.5\sigma_{rB}\\ 0 & otherwise\end{cases} \quad (4.2)$$

where $I_R(x,y)$, $I_G(x,y)$ and $I_B(x,y)$ are the input RGB value; $\mu_R(x,y)$, $\mu_G(x,y)$ and $\mu_B(x,y)$ denote the background RGB value at coordinate (x,y); r_R, r_G and r_B are the RGB mean ratio value of pixels when shadowed in the image; σ_{rR}, σ_{rG} and σ_{rB} represent the RGB standard deviation ratio of pixels when shadowed in the background image.

Figures 4-2(a)-(d) illustrate an example of the RGB Gaussian shadow models. In Fig. 4-2(a), 100 samples of shadow pixels in the image frame are selected to build the Gaussian

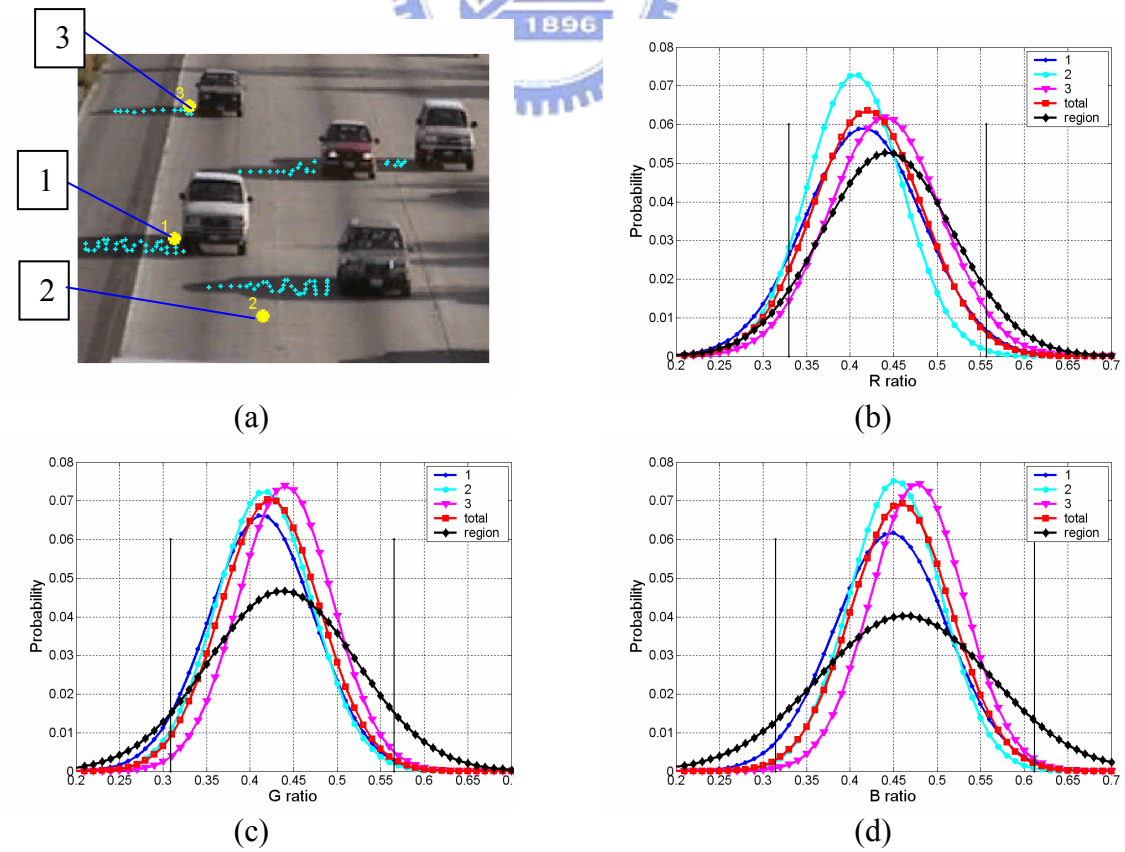


Fig. 4-2. The Gaussian models of RGB ratio of recorded samples and shadow-region data.

RGB ratio model, as depicted by small white dots in the shadow region of the figure. To validate the Gaussian model found from the shadow region, we tested shadow data at three points (indicated by large white dots in Fig. 4-2(a)). Furthermore, all the recorded data are also combined and used to verify the hypothesis of constant color ratio. The Gaussian RGB ratio models of recorded data (labeled by 1, 2, 3 and total) and shadow-region data (labeled by region) are depicted in Figs. 4-2(b)-(d), respectively. The mean and the standard deviation of the RGB ratio of each sample are presented in Table 4-1. In our design, a point is regarded as a shadow point if its RGB ratio satisfies (4.2). The results computed from the four groups of samples (labeled by 1, 2, 3 and total) demonstrate that the accuracy is higher than 86%, as shown in Table 4-1. It demonstrates the effectiveness of the hypothesis and therefore, one can use the shadow-region-based RGB ratio model to determine the shadow in image sequences. Figure 4-3 illustrates an example of shadow detection. By using (4.2) and the Gaussian

Table 4-1 The Gaussian Models Of RGB Ratio of Recorded Data and Shadow-Region Data

Item	Mean	Standard deviation	Accuracy
R ratio of region	0.443	0.076	86.64%
R ratio of 1	0.416	0.067	88.16%
R ratio of 2	0.406	0.055	91.58%
R ratio of 3	0.440	0.064	92.16%
R ratio of total samples	0.420	0.063	91.15%
G ratio of region	0.437	0.085	86.64%
G ratio of 1	0.413	0.060	96.39%
G ratio of 2	0.416	0.055	96.70%
G ratio of 3	0.440	0.055	97.65%
G ratio of total samples	0.423	0.057	97.25%
B ratio of region	0.463	0.099	86.64%
B ratio of 1	0.447	0.065	97.47%
B ratio of 2	0.452	0.053	99.41%
B ratio of 3	0.477	0.054	99.30%
B ratio of total samples	0.459	0.058	99.02%

model, one can detect the shadow, as shown in Fig. 4-3(a). Figure 4-3(b) shows the detected moving region. The result of statistical nonparametric shadow suppression is shown in Fig. 4-3(c). One observes in Fig. 4-3(c) that some shadow pixels are not recognized as expected. This is mainly due to uncertainties in the image sensing. We handle this insufficiency and improve the performance of shadow suppression by adding a spatial analysis post-processing step.

4.2.2 Spatial Analysis for Shadow Verification

Two types of shadow detection error may occur, namely, shadow detection failure and object detection failure. The first type of error occurs if a shadow pixel has ratios outside the detection range of the shadow model, causing the shadow to not be recognized. As shown in Fig. 4-3(c), some shadow pixels are not recognized. The second type of failure detection occurs if the ratios of an object pixel lie inside the detection range of the shadow model. This occurs especially when the ratios are higher than the mean of Gaussian ratio model and still inside the detection range; there are almost no shadow pixels in this region, as shown in

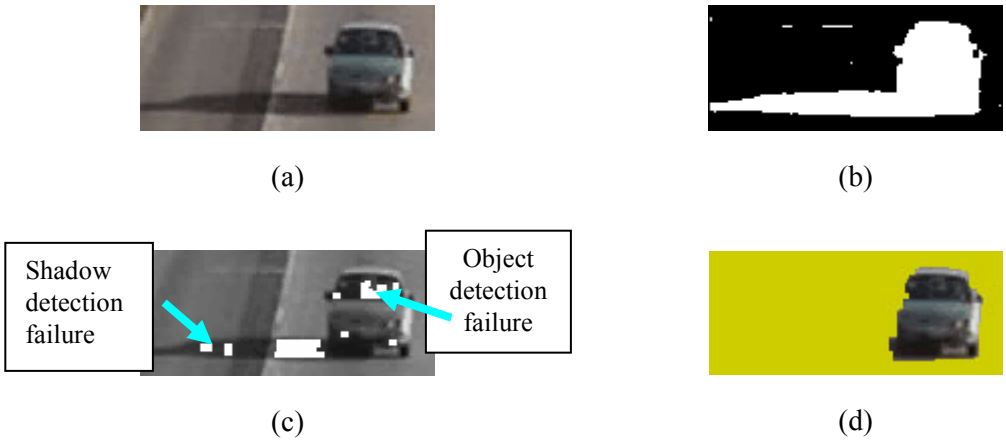


Fig. 4-3. Explanation of shadow suppression steps. (a) Original image. (b) Moving object segmentation result of background removal. (c) Shadow segmentation result of spectral ratio shadow detection. Detected shadow is indicated by white area. (d) Segmentation result of shadow suppression after spatial analysis.

sample plots of Figs. 4-2(b)-(d). For instance, in Fig. 4-3(c), partial pixels of a vehicle are misclassified as shadow pixels. To improve the accuracy of shadow detection, a post-processing of spatial analyses is added for shadow confirmation. The spatial analyses are performed to confirm the true shadows and the true objects according to their geometric properties.

4.2.3 Size Discrimination of Moving Object Candidates

In the process of shadow detection, shadow candidates sometimes break actual shadow into small isolated blobs. Generally the sizes of these small shadow blobs are smaller than the detected moving vehicles in image sequences. These small blobs will not be considered moving object candidates. Thus, one can discriminate the small blobs of shadow from the big blobs of moving objects by using the size information. In our design, all blobs of moving object candidates are grouped into different regions using a connected components labeling algorithm. The regions that have small sizes are recognized as shadow regions.

4.2.4 Border Discrimination of Moving Cast Shadow Candidates

The moving blobs segmented by background removal consist of shadow pixels and object pixels. In practice, the true shadow pixels cluster in the fringes of the blobs. If a part of the detected vehicle is misclassified as a shadow, most of boundaries of this region will be located inside the candidate foreground, as shown in Figs. 4-3(b)-(c). If the shadow candidate is a true shadow, more than a half of the boundary is adjacent to the boundary of foreground candidates. Thus, one can use the boundary information of a shadow-candidate region to confirm whether the shadow is a true shadow or not [40]. In our design, the boundaries of the foreground candidate are segmented by Sobel edge detection. Next, each distinct candidate shadow region is determined by using a connected components labeling algorithm. Sobel edge detection is also used to find the edge of each distinct shadow region [64]. The number

N_f of boundary shadow pixels that are adjacent to the boundary of a foreground-candidate region and the number N_s of all boundary shadow pixels are computed. The shadow is considered a true shadow if the ratio $\frac{N_f}{N_s}$ is greater than 50%. The confirmation result of spatial analysis is shown in Fig. 4-3(d). It indicates that the accuracy of shadow detection is greatly improved in comparison to the original.

4.3 Comparison Results

For traffic monitoring and surveillance applications, shadow suppression prevents misclassification and erroneous counting of moving vehicles. The goal of shadow suppression is to minimize the false negatives (FN_S , the shadow pixels misclassified as background/foreground) and the *false positives* (FP_S , the background/foreground pixels misclassified as shadow pixels). In order to systematically evaluate the performance of the proposed method, we adopted two metrics, namely the shadow detection rate η and the object detection rate ξ [36] for quantitative comparison:

$$\eta = \frac{TP_S}{TP_S + FN_S}, \tag{4.3}$$

$$\xi = \frac{TP_F}{TP_F + FN_F}. \tag{4.4}$$

where TP_S (resp. TP_F) is the number of shadow (resp. foreground) pixels correctly identified, and FN_S (resp. FN_F) is the number of shadow (resp. foreground) pixels falsely identified. A comparison with existing methods has been carried out to validate the performance of the proposed algorithm. A statistical nonparametric (SNP) approach [72] and a deterministic nonmodel-based (DNM) approach [38] were selected for comparison. The SNP

approach treated object colors as a reflectance model from the Lambertian hypothesis. The work used the normalized distortion of the brightness α_i' and the distortion of the chrominance CD_i', computed from the difference between the background color of a pixel and its value in the current image, to classify a pixel in four categories:

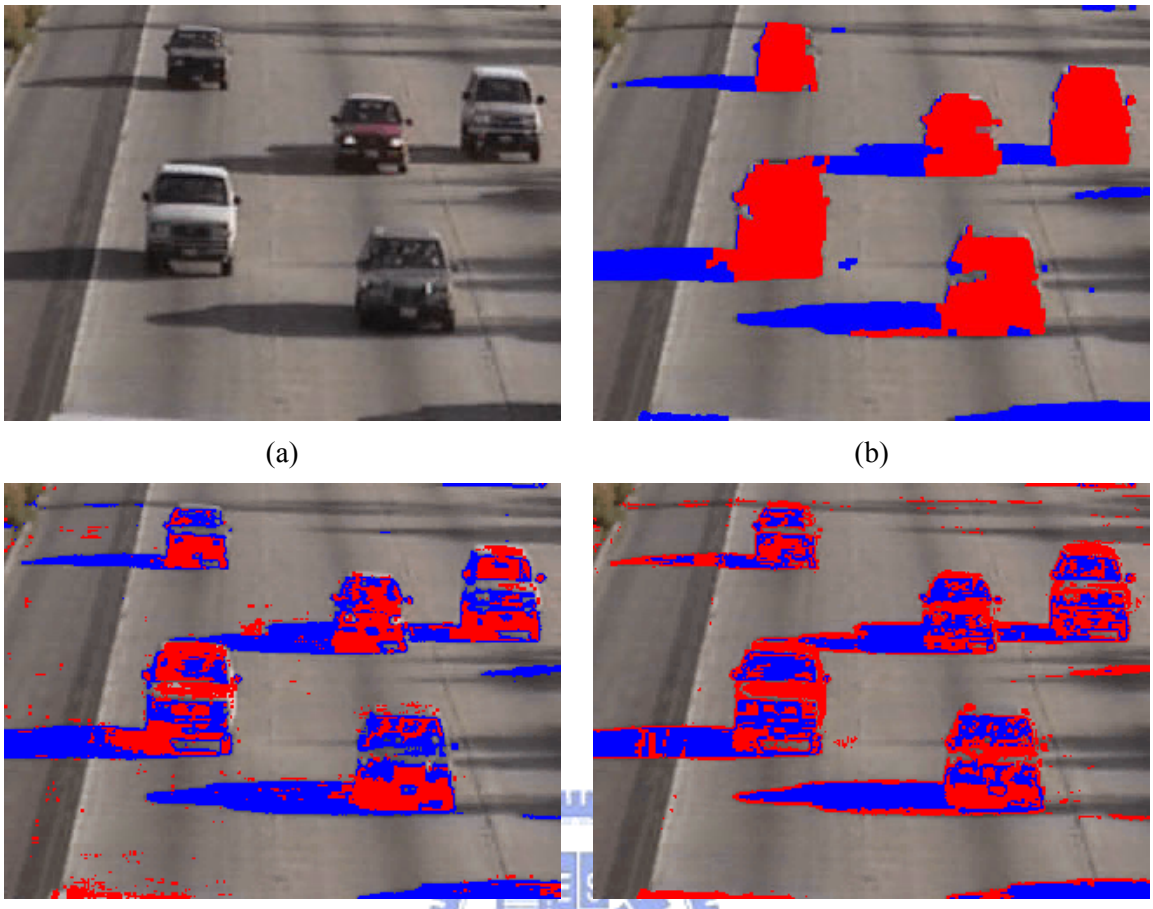
$$C(i) = \begin{cases} \text{Foreground :} & CD_i' > \tau_{CD} \text{ or } \alpha_i' < \tau_{\alpha lo} \text{ else} \\ \text{Background:} & \alpha_i' < \tau_{\alpha 1} \text{ and } \alpha_i' > \tau_{\alpha 2} \text{ else} \\ \text{Shadowed background:} & \alpha_i' < 0 \text{ else} \\ \text{Highlighted background:} & \text{Otherwise} \end{cases} \quad (4.5)$$

The DNM approach works in the HSV color space. Shadow detection is determined according to the following equation:

$$SP(x,y) = \begin{cases} 1 & \alpha \le \frac{I_k^V(x,y)}{B_k^V(x,y)} \le \beta \wedge I_k^S(x,y) - B_k^S(x,y) \le \tau_S \wedge D_H(x,y) \le \tau_H \\ 0 & otherwise, \end{cases}, \quad (4.6)$$

where $D_H(x,y) = \min(\left|I_k^H - B_k^H\right|,\ 360 - \left|I_k^H - B_k^H\right|)$, $I_k(x,y)$ and $B_k(x,y)$ are the pixel values at (x,y) coordinate in the input image (frame k) and in the background model (computed at frame k) respectively.

To achieve an objective comparison, the model for the background image is updated in advance for all algorithms in the test. Figure 4-4 shows the test results of different methods (Proposed, DNM [38] and SNP [72]) from a benchmark sequences Highway-I [36]. First, we check the computation time of each algorithm for efficiency comparison, as shown in Table 4-2. The proposed algorithm requires the least computation time. It is reduced to 3.6%-21% of the other two methods because it merely utilizes division operation to obtain the shadow



(a) (b)

(c) (d)

Fig. 4-4. Comparison results of shadow suppression method (red pixels represent the moving vehicle; the blue pixels represent the attached shadow). (a) Original image. (b) The proposed method. (c) The SNP method. (d) The DNM method.

Table 4-2 Computation Time for Shadow Spectral Analysis

Algorithm	**Proposed**	**SNP**	**DNM**
Time (msec)	**3.4**	**93**	**15.6**
Specification:	**Image size= 352x240,** **CPU type= Intel Pentium 4 2.4GHz,** **RAM=448MB.**		

information. The SNP algorithm takes the longest time because of its complex normalization, which consists of square, square root, addition, and division operations. Second, we examine the shadow detection rate and the object detection rate of each algorithm. The ground truth for each frame is necessary for calculating the quantitative metrics of (4.3) and (4.4). We manually and accurately classified the pixels into foreground, background, and shadow

regions in the image sequences. Figures 4-5 and 4-6 show the comparison results of shadow detection rate and object detection rate between the proposed method, SNP, and DNM. The mean values of accuracy corresponding to the plots in Figs. 4-5 and 4-6 are reported in Table 4-3. The results evaluated by Prati *et al.* [36] (by analyzing tens of frames for each video sequence representative of different situations) are also listed in Table 4-3. The experimental results demonstrate that the proposed method generally provides more reliable object detection results compared with other state-of-the-art methods. The results obtained from algorithms with or without spatial analysis are also listed in the table for checking the effectiveness of spatial analysis. One observes that as expected, spatial analysis improves the performance of shadow detection. For instance, the shadow detection rate of the DNM algorithm is increased from 57% to 80%. The merit of spatial analysis is that it can combine with other existing shadow suppression methods for further improvement of performance. The proposed method outperforms the other two methods in shadow suppression and moving object detection because it uses the ratio model, which is constructed from only a single image frame. The video clip of the experimental results can be found at: http://isci.cn.nctu.edu.tw/video/JCTai/shadow1.wmv.

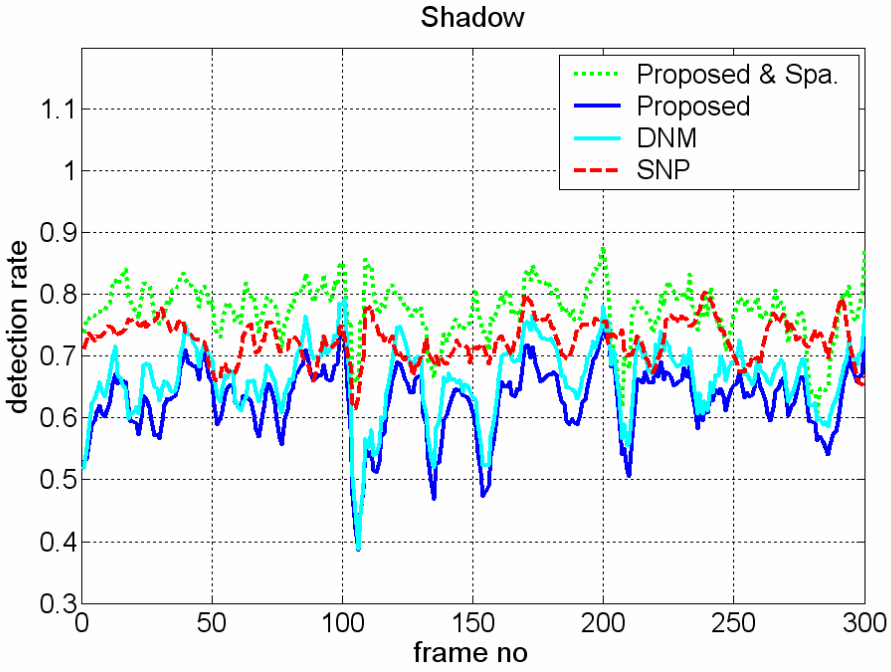


Fig. 4-5. Comparison result of shadow detection rate between the proposed method, DNM, and SNP.

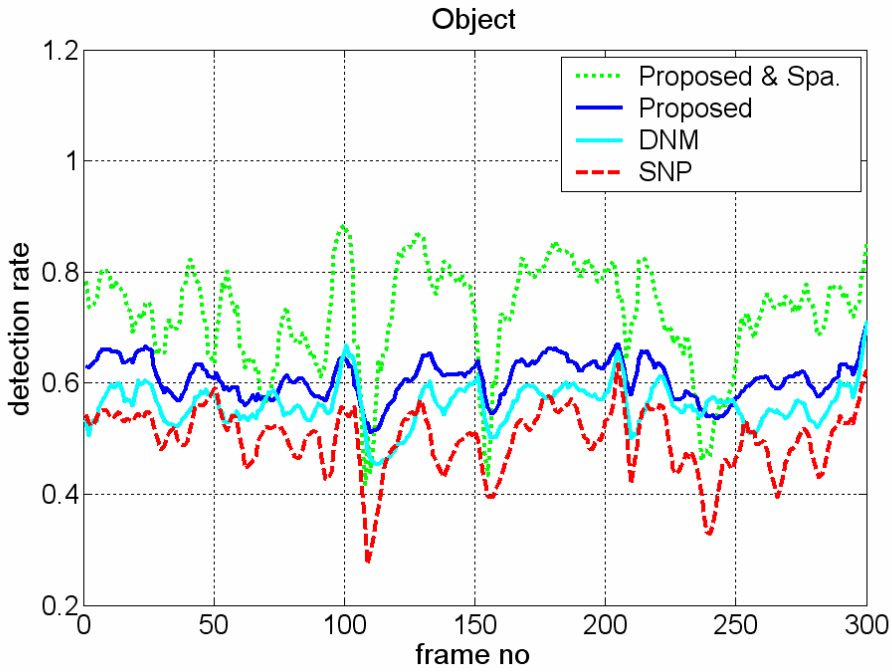


Fig. 4-6. Comparison result of object detection rate between the proposed method, DNM, and SNP.

Table 4-3 The Accuracy of Detection Results.

Method	Shadow detection rate[*1]	Object detection rate[*1]	Shadow detection rate[*2]	Object detection rate[*2]
Proposed &Spa.	77.5%	72.2%	76.86%	80.52%
Proposed	62.9%	60.7%	71.97%	70.14%
SNP	72.6%	49.9%	74.48% (*81.59%*[*3])	59.39% (*63.76*[*4])
DNM	66.4%	56.07%	67.94% (*69.72%*[*3])	68.57%, (*76.93%*[*4])

*1: The mean accuracy of detection results obtained by analyzing 300 frames for each video sequence.

*2: The mean accuracy of detection results obtained by analyzing tens of frames for each video sequence representative of different situations.

*3: Results from [36].

*4: Results from [36], the false positives belonging to the background were not considered in the computation of the object detection rate.

4.4 Summary

In this chapter, a shadow-region-based statistical nonparametric method has been developed to construct a ratio model for shadow detection of all pixels in an image frame.

Based on the Lambertian assumption, RGB ratios between lit pixels and shadowed pixels can be treated as a constant in image sequences. This assumption leads us to the development of a novel ratio model for detecting shadow pixels in traffic imagery. The proposed approach does not require many image sequences to construct the model. Instead, the model can be easily built using a shadow region in a single image frame. To increase the accuracy of shadow detection, two types of spatial analysis are proposed to verify the actual shadow pixels.

Chapter 5

Vehicle Detection and Tracking for Traffic Parameter Estimation

5.1 Introduction

Recently, image tracking has become an important technique for traffic monitoring and surveillance applications. Many algorithms based on image tracking have been developed for real-time traffic parameter estimation. After the moving vehicles have been segmented, the process of traffic parameter estimation consists of three main stages: vehicle detection, vehicle tracking, and traffic parameter estimation. Most existing algorithms used a special-design region lying in each lane to detect and track entering vehicles. The main drawback of these methods is that only similarly sized vehicles passing through a particular region of the road can be detected and tracked. If the size of a vehicle differs from the predefined one or if the vehicle does not pass through the particular region, it will not be detected and tracked by these existing algorithms.

In this chapter, we propose an image-based traffic monitoring system that automatically detects and tracks multiple different-sized vehicles that travel in any portion of a multi-lane road. We adopted active contour models to represent vehicle contours in an image frame. A method based on image measurement is developed to predict initial positions and sizes of vehicles for image contour generation. This method features simultaneous detection of

multiple vehicles that travel in any portion of the road. Kalman filtering techniques are then applied for active-contour-based image tracking of various vehicles. Analyzing the detection and tracking results allows us to estimate traffic parameters such as traffic flow rate, vehicle speed, and traffic density. Moreover, by using an optical flow method to obtain the traveling direction of vehicles, the detection method can also be used for estimating the vehicle turn ratio at an intersection.

The rest of this chapter is organized as follows. Section 5.2 gives an overview of the system for traffic parameter estimation. An image measurement algorithm for active contour representation will be described. Section 5.3 presents the proposed image tracking method and its application in ITMS. Section 5.4 describes the turn ratio estimation based on optical flow measurement. Experimental results of traffic parameter estimation using the proposed method will be presented in Section 5.5. Section 5.6 summarizes the contribution of this work.

5.2 System Overview of Image Tracking

Figure 5-1 shows the system architecture of the proposed contour initialization and tracking system. This image tracking system consists of three parts: the image processing module, the contour initialization module and the vehicle tracking module. The image processing module captures the traffic scene, segments the background image, removes the shadow, and extracts the binary image of moving vehicles from the image sequence. The contour initialization module uses a detection window to generate an initial vehicle contour for tracking operation. Once initialized, the vehicle contour will be tracked and updated iteratively. The active contour model is exploited to represent vehicle contours in this design. The vehicle tracking module employs a dynamic model to predict the vehicle contour from its historical states. The contour of targeted vehicle is iteratively obtained by using image

measurement and Kalman filtering.

5.2.1 Active Contour Model

Active contour modeling is a powerful tool for model-based image segmentation [6] [73]-[74]. Based on the active contour concept, an image measurement method for obtaining the best-fit curve of a vehicle contour for image tracking is presented below. In this work, *B*-spline functions are adopted to represent vehicle contours in image frames. The vehicle contour $(x(s), y(s))$ is represented using N_B *B*-spline functions:

$$x(s) = \sum_{n=0}^{N_B-1} B_n(s) q_n^x = B(s)^T Q^x \quad \text{for} \quad 0 \le s \le N_B \tag{5.1}$$

$$y(s) = \sum_{n=0}^{N_B-1} B_n(s) q_n^y = B(s)^T Q^y \quad \text{for} \quad 0 \le s \le N_B, \tag{5.2}$$

where $B(s) = (B_0(s), \ B_1(s), \ \ldots \ , \ B_{N_B-1}(s))^T$,

$$B_0(s) = \begin{cases} s^2/2 & \text{if} \quad 0 \le s < 1 \\ \frac{3}{4} - (s - \frac{3}{2})^2 & \text{if} \quad 1 \le s < 2 \\ (s-3)^2/2 & \text{if} \quad 2 \le s < 3 \\ 0 & \text{if} \quad \text{otherwise} \end{cases},$$

$$B_n(s) = B_0(s-n),$$

$$Q^x = \begin{pmatrix} q_0^x \\ q_1^x \\ \vdots \\ q_{N_B-1}^x \end{pmatrix} \text{ and } Q^y = \begin{pmatrix} q_0^y \\ q_1^y \\ \vdots \\ q_{N_B-1}^y \end{pmatrix}.$$

The contour $r(s)$ is represented by a vector $Q(=\begin{pmatrix} Q^x \\ Q^y \end{pmatrix})$ and Q contains the *X*-*Y* coordinates of the control points of the B-spline curve, so that

$$r(s)=\begin{pmatrix} x(s) \\ y(s) \end{pmatrix}=I_2 \otimes B(s)^T Q=\begin{pmatrix} B(s)^T & 0' \\ 0' & B(s)^T \end{pmatrix} Q=U(s)Q\,, \tag{5.3}$$

where I_2 denotes a 2x2 identity matrix, $\otimes$ is the Kronecker product denotation of two matrices and $0'$ is $(0,0,0,0,0,0,0,0)^T$. In this design, a control point vector Q containing 8 control points is used to represent the vehicle contour, as shown in Fig. 5-2. The control points are indicated by circles in the figure.

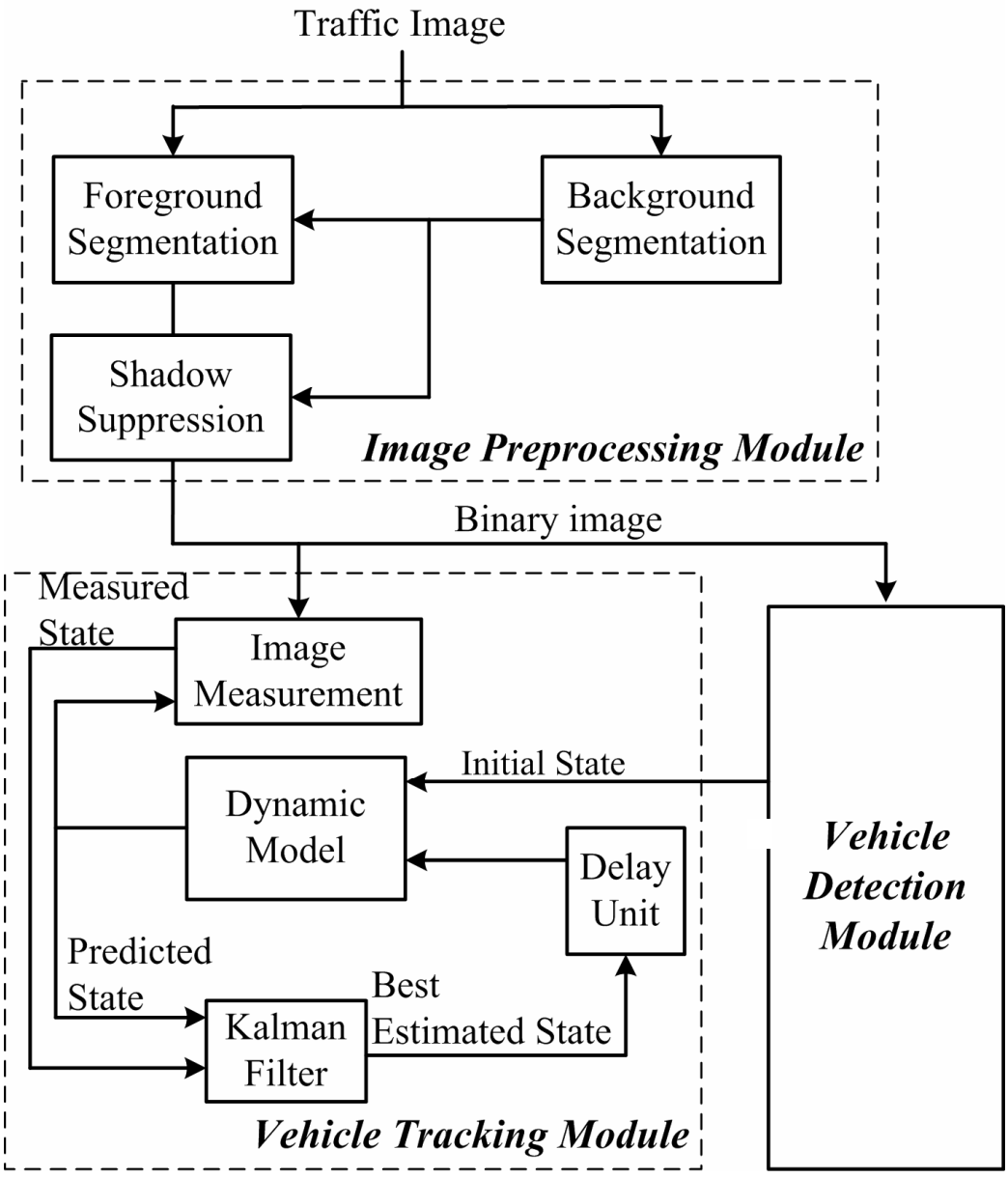


Fig. 5-1. Vehicle tracking system architecture.

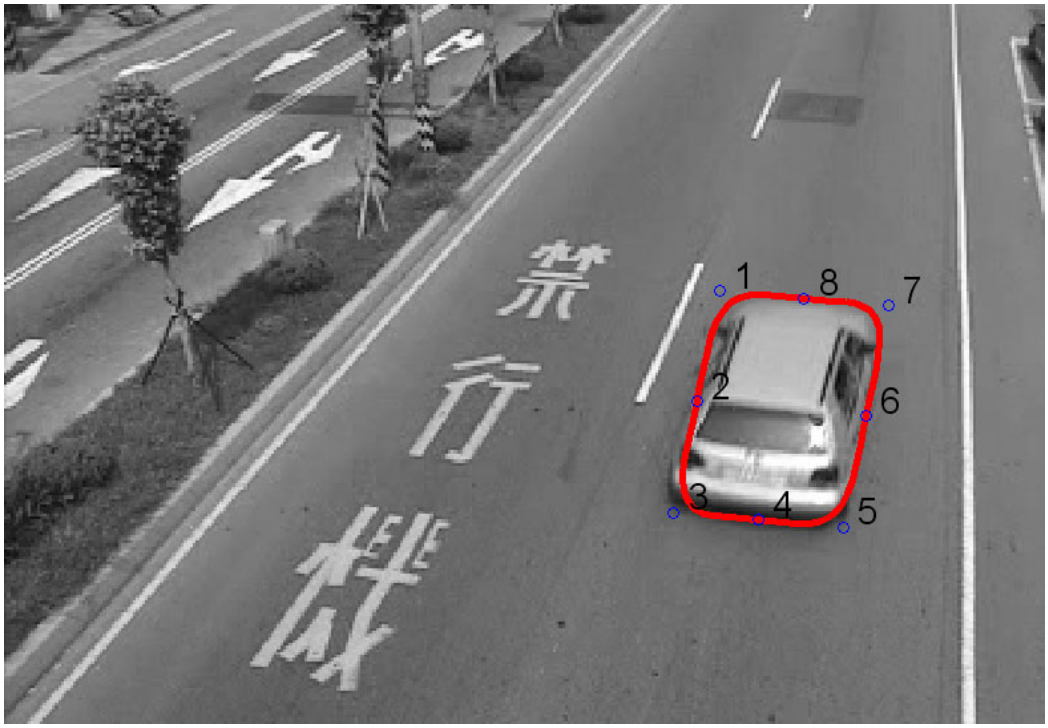


Fig. 5-2. Active contour of a vehicle.

5.2.2 Shape Space Transformation

In traffic image sequences, the contour of a moving vehicle will vary due to projective effects of camera's view angle. Within a reasonable view angle, it can be assumed that the variation of vehicle contours is linear in traffic imagery. The vehicle contour can then be described by a shape-space planar affine transformation in the image plane. The boundary curve $r(s)$ of each vehicle is expressed using a template curve $r_0(s)$ [75]:

$$r(s) = u_t + Mr_0(s), \tag{5.4}$$

where $u_t = (u_x, u_y)^T$ is a two-dimensional translation vector and M is a 2×2 affine-matrix comprising one rotation and three deformation (horizontal, vertical, and diagonal) elements. Subtracting $r_0(s)$ from (5.4), one obtains:

$$r(s) - r_0(s) = u_t + (M - I)U(s)Q_0 = U(s)\begin{pmatrix} 1' & 0' & Q_0^x & 0' & 0' & Q_0^y \\ 0' & 1' & 0' & Q_0^y & Q_0^x & 0' \end{pmatrix} X = U(s)WX, \tag{5.5}$$

where $1'$ is $(1,1,1,1,1,1,1,1)^T$, Q_0^x, Q_0^y are X-Y coordinates of the control points of the template curve Q_0 and shape-space vector is

$$X = \begin{pmatrix} u_x & u_y & M_{11}-1 & M_{12} & M_{21} & M_{22}-1 \end{pmatrix}^T .$$

Subtracting $r_0(s)$ from (5.3), (5.5) can be rewritten as

$$r(s) - r_0(s) = U(s)(Q - Q_0) . \tag{5.6}$$

Comparing (5.5) and (5.6), one obtains a linear transformation:

$$Q = WX + Q_0 . \tag{5.7}$$

Using (5.7), one can transform a vehicle contour to a shape-space vector *X*. This simplifies the post-processing of contour tracking in the image plane and the vehicle contour will be restricted to vary steadily by the shape-space vector.

5.2.3 Image Measurement

An image measurement procedure is responsible for obtaining the best-fit curve of the vehicle outline in an image according to a predicted vehicle contour generated from a predicted shape-space vector $\widetilde{X}$, a template curve Q_0, and its shape matrix W. The binary image of a traveling vehicle is segmented from traffic imagery exploited a background removal operation (see Chapter 3). The contour feature $r_f(s)$ is obtained by applying one-dimensional (1-D) image processing along the normal direction of a predicted curve $\bar{r}(s)$ [76]. Curve-fitting method of the detected features is employed to obtain the best-fit curve of the vehicle contour. In carrying out the curve fitting of contour features, one has to increase the tolerance for image disturbance and eliminate possible interference from features of other objects in the background. A contour shape-space vector $\widetilde{X}$ and a regularization constant α are used to stand for the relative effect of the shape in the curve fitting and

meet the criteria mentioned above.

Introducing the concept of information matrix S_i and information weight sum Z_i, the algorithm for finding the best fitting curve can be summarized as follows:

1) Select N regularly equal-spaced samples s_i, $i = 1,2,3....N$ and $s_1 = h$, $s_{i+1} = s_i + h$, $s_N = Nh = N_B$.

2) For each i, find the position of $r_f(s_i)$ by applying one-dimensional (1-D) image processing along the normal line passing through $\bar{r}(s)$ ($\bar{r}(s)$ is the contour of $\tilde{X}$) at $s = s_i$.

3) Initialize

$$Z_0 = 0, \quad S_0 = 0.$$

Iterate, for $i = 1,2,3....N$

$$\nu_i = (r_f(s_i) - \bar{r}(s_i)) \cdot \bar{n}(s_i) \tag{5.8}$$

$$h(s_i)^T = \bar{n}(s_i)^T U(s_i) W \tag{5.9}$$

$$S_i = S_{i-1} + \frac{1}{\sigma_i^2} h(s_i) h(s_i)^T \tag{5.10}$$

$$Z_i = Z_{i-1} + \frac{1}{\sigma_i^2} h(s_i) \nu_i \tag{5.11}$$

where $\bar{n}(s_i)$ is the normal unit vector of curve $\bar{r}(s)$ at $s = s_i$ and $\sigma_i^2 = N_B$.

4) The aggregated observation vector is

$Z = Z_N$ with the associated statistical information $S = S_N$

5) The best-fit curve is expressed as a shape-space vector [76]

$$\bar{X} = \tilde{X} + (\bar{S} + S)^{-1} Z, \tag{5.12}$$

where $\bar{S} = \alpha W^T (\frac{1}{N_B} \int_0^{N_B} (I_2 \otimes B(s)^T)^T (I_2 \otimes B(s)^T) ds) W$.

5.3 Image Tracking and Traffic Parameter Estimation

In our design, the image-based traffic monitoring procedure includes four steps: foreground segmentation, contour initialization, vehicle tracking, and traffic parameter extraction. The first step segments moving vehicles from the image sequence. The contour initialization step detects the moving vehicle and generates an initial contour for tracking. In the vehicle tracking step, targeted vehicles are tracked using a specially designed Kalman filter. The final step extracts the traffic parameters by using the tracking results.

5.3.1 Initial Contour Generation

To track multiple vehicles that are of various sizes and that might travel in any portion of a multi-lane road, we propose a contour initialization algorithm to generate initial contours for image tracking by using a special-designed detection window. The concept of the detection window is shown in Fig. 5-3. Depending on current vehicle imagery, there can be multiple detection regions and initialization regions in the detection window, as shown in Fig. 5-3. In the beginning, the entire detection window is the detection region. The system works to check whether any vehicle enters the detection region. When a vehicle is detected, the related detection region will change into an initialization region. The rest of the detection region remains unchanged. If the detected vehicle leaves the initialization region, this region will be released and become a detection region again. The detection region and the initialization region are automatically adjusted according to the current traffic imagery.

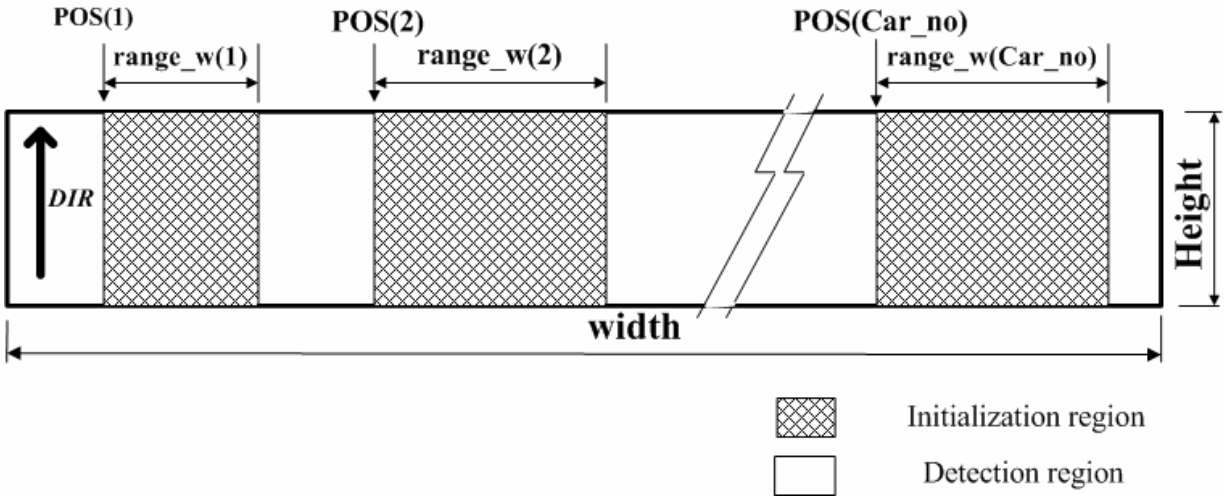


Fig. 5-3. A detection window consists of initialization regions and detection regions.

5.3.1.1 Image Processing in Detection Region and Initialization Region

To facilitate vehicle detection, the detection region is divided into several 1-pixel-width sub-regions. A sub-region is termed as an occupied sub-region if a foreground object exists in this sub-region. When the front part of vehicle enters the detection window, a cluster of occupied sub-regions will appear. The width of the vehicle image can be found from the front part of the vehicle image. As the related detection region contains enough occupied sub-regions, it will change into an initialization region, as mentioned in the previous paragraph. Fig. 5-4 shows a test example of a car and a motorcycle in the detection window. Both the vehicle and the motorcycle are detected; two initialization regions are automatically generated in this case.

5.3.1.2 Initial Contour Generation

The final stage of contour initialization is the generation of an initial contour as the vehicle leaves the detection window. Fig. 5-5 depicts the concept of initial contour generation. First, an estimated contour is automatically generated by using the geometric information of the initialization region. The width of the estimated contour is the width w of initialization region. The length L_b of the estimated contour is assigned to $\Re w$, $\Re$ is an empirical ratio

of the length to the width of vehicle in the captured image frame. The location of the estimated contour is assigned at the exit of the initialization region, as shown in Fig. 5-5. It is clear that the estimated contour is generated via simple geometric relationships in the initialization region; it may not match the actual situation of the vehicle image. Next, the estimated contour will be refined by using image measurement, as described below.

The dimension and location of the estimated contour are adjusted by using the actual vehicle image. The length L_b and the location of the estimated contour are possibly different from the actual ones and need to be corrected. The width w is previously estimated by the detection window when the vehicle entered the detection window; it is not necessary to adjust the width of estimated contour. We project the captured binary image into two one-dimensional arrays and use the projection to measure the occupancy of a vehicle image. The projection values are the sum of vehicle pixels along the vertical and horizontal

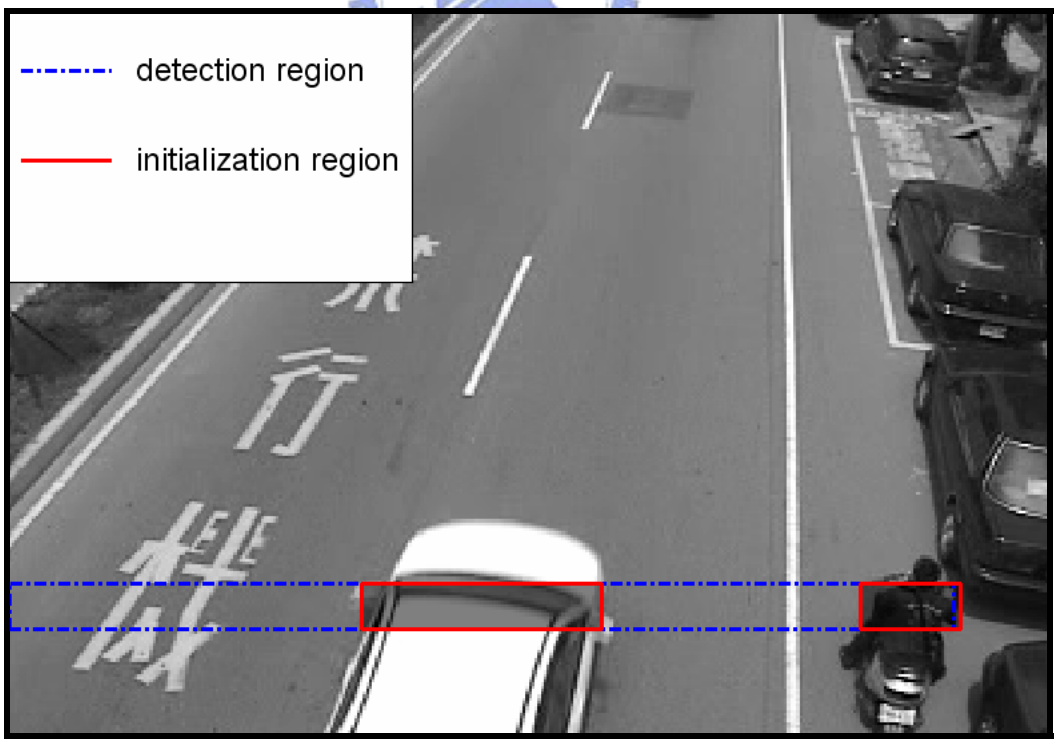


Fig. 5-4. The detection of a car and a motorcycle.

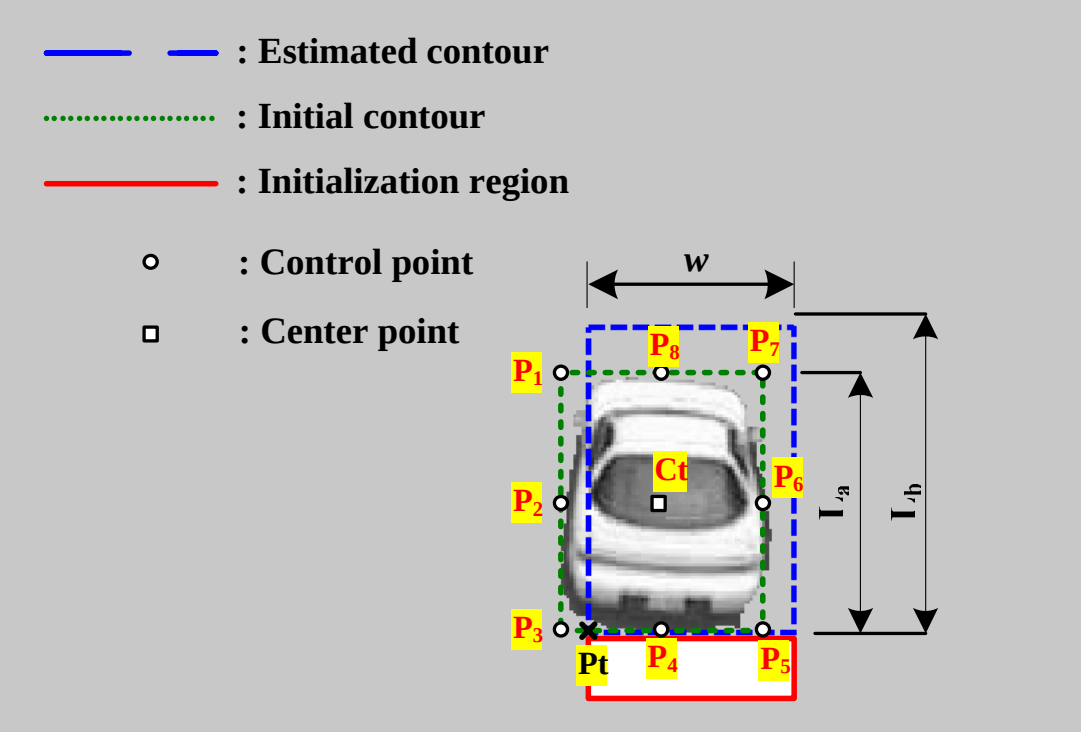


Fig. 5-5. Generation of an initial contour.

directions, respectively. Fig. 5-6 illustrates an example. In this case, the vertical projection reveals that the corner point *Pt* of initial contour should shift to the point P_3, while the horizontal projection reveals that the length L_b should be adjusted to L_{a1}, as shown in Fig. 5-5. The control points and center point of the initial contour is then generated using w, L_a, P_3, and initialization region. It is employed to calculate the template Q_0 and the shape-space vector X for image tracking:

$$Q_0^x = \begin{pmatrix} u_1 - u_c \\ u_2 - u_c \\ \vdots \\ u_8 - u_c \end{pmatrix}, \quad Q_0^y = \begin{pmatrix} v_1 - v_c \\ v_2 - v_c \\ \vdots \\ v_8 - v_c \end{pmatrix} \quad \text{and} \quad X = \begin{pmatrix} u_c \\ v_c \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

where $(u_1, v_1), (u_2, v_2), \ldots, (u_8, v_8)$ and (u_c, v_c) are pixel-based coordinates of points P_1, P_2, …, P_8 and C_t.

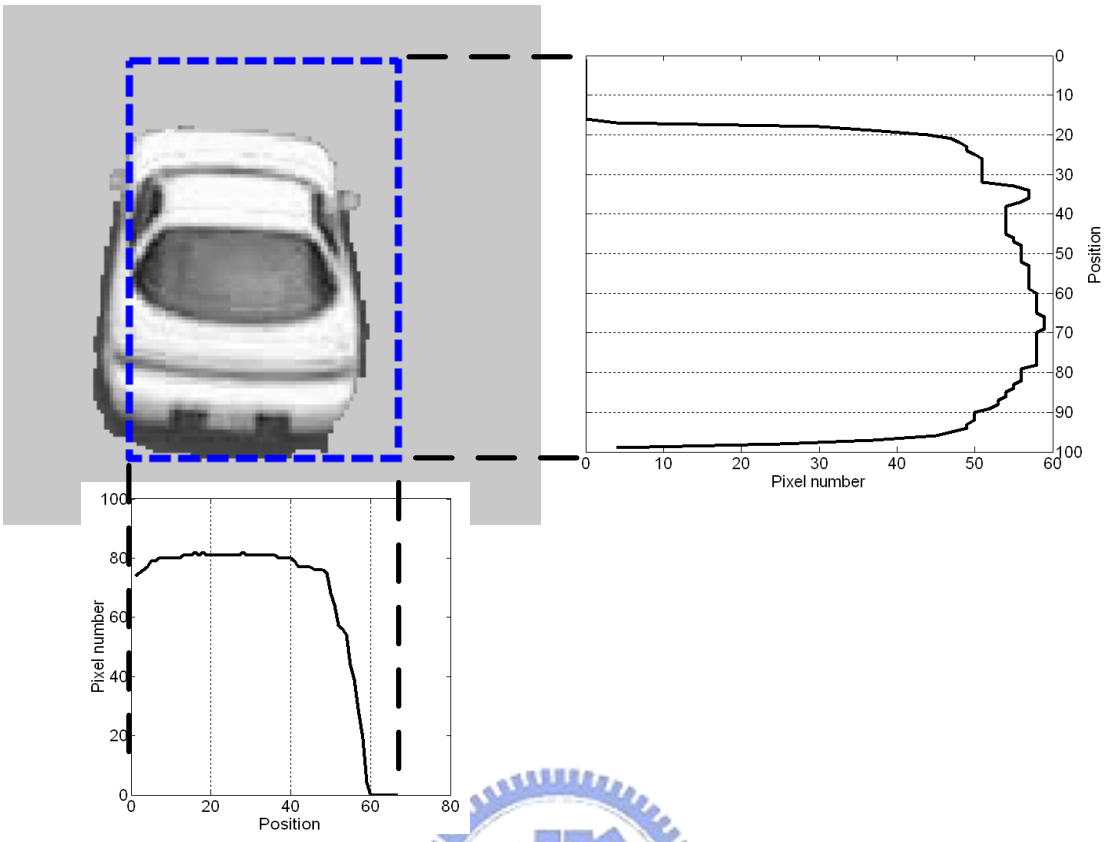


Fig. 5-6. The projection profile of an estimated contour.

5.3.2 Kalman Filtering for Tracking

The vehicle contour is represented by a shape-space vector X with six elements. The first two elements of X are position coordinates of the template curve and the remaining elements are shape-scaling elements, as described in Section 5.2.2. The vehicle tracking module employs two dynamic models to predict the horizontal and vertical positions from their historical position states. As for the shape-scaling elements, because the change of vehicle contour is very small within two consecutive frames, it is not necessary to employ complex dynamic models to predict the shape-scaling elements. The predicted states of these elements are simply assigned using the previously measured state $\overline{X}$ obtained from the image measurement, as described in Section 5.2.3. The predicted states are provided to the information fusion stage for tracking a vehicle.

Below is the design of the dynamic position model for contour prediction. The state of horizontal or vertical position can be governed by:

$$x_k = x_{k-1} + (x_{k-1} - x_{k-2}) + \varepsilon_k, \tag{5.13}$$

where ε_k is the system noise. Let $X_k = \begin{pmatrix} x_k \\ x_{k-1} \end{pmatrix}$, then dynamical model is

$$X_k = AX_{k-1} + \Gamma\varepsilon_k, \tag{5.14}$$

where $A = \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix}$ and $\Gamma = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. The observation model is

$$v_k = CX_k + \eta_k, \tag{5.15}$$

where $C = [1 \quad 0]$ and η_k is a observation noise. Let $\{\varepsilon_k\}$ and $\{\eta_k\}$ be sequences of zero-mean Gaussian white noise such that $Var(\varepsilon_k) = Q_k$ and $Var(\eta_k) = R_k$ are positive definite matrices and $E(\varepsilon_k \eta_l) = 0$ for all k and l. Equations (5.16) and (5.17) are the state space description of linear stochastic system. A Kalman filter is designed to combine the information from the predicted states and the best-fit states obtained from (5.12) [77]. Fig. 5-7 shows the block diagram of the image tracking system. The tracking procedure over one time-step is summarized as follows:

1) Predict the state $\hat{X}_{k,k-1}$ according to the previous state $\hat{X}_{k-1,k-1}$:

$$\hat{X}_{k,k-1} = A\hat{X}_{k-1,k-1}. \tag{5.16}$$

2) Determine the error covariance ahead,

$$P_{k,k-1} = AP_{k-1,k-1}A^T + \Gamma Q \Gamma^T. \tag{5.17}$$

3) Compute the Kalman gain,

$$K = P_{k,k-1}C^T (CP_{k,k-1}C^T + R)^{-1}. \tag{5.18}$$

4) Use a Kalman filter to obtain the information fusion:

$$\hat{X}_{k,k} = \hat{X}_{k,k-1} + K(v_k - C\hat{X}_{k,k-1}), \tag{5.19}$$

where v_k is the measured state.

5) Update the error covariance

$$P_{k,k} = (I - KC)P_{k,k-1}; \tag{5.20}$$

then go to step 1 for the next iteration.

5.3.3 Traffic Parameter Estimation

As vehicles in an image sequence are successfully tracked, traffic parameters such as traffic flow rate, vehicle speeds, and traffic density can be obtained through simple calculation. The traffic flow rate can be calculated by the ratio of detected vehicle numbers to the elapsed time. Traffic density D (car/km) is calculated as follows:

$$D = \frac{q}{s}, \tag{5.21}$$

where q is the flow rate (car/hr) and s is the average travel speed (km/hr).

Vehicle speed can be obtained from two recorded positions of vehicle image and the elapsed time between these two positions. The center of the bottom edge of a vehicle image was taken as the reference point of the vehicle's position. This point can be easily obtained from the control vectors of vehicle contour.

From the tracking result of a tracked vehicle, the reference point P_a (tracking operation is initialized) at time t_a and P_b (as the vehicle attains a predefined region, tracking operation terminates) at time t_b are recorded. Using (B.6) and (B.7), positions P_a and P_b are transformed to world coordinates to calculate the traveling distance L between P_a and P_b. The vehicle speed vel can be calculated by

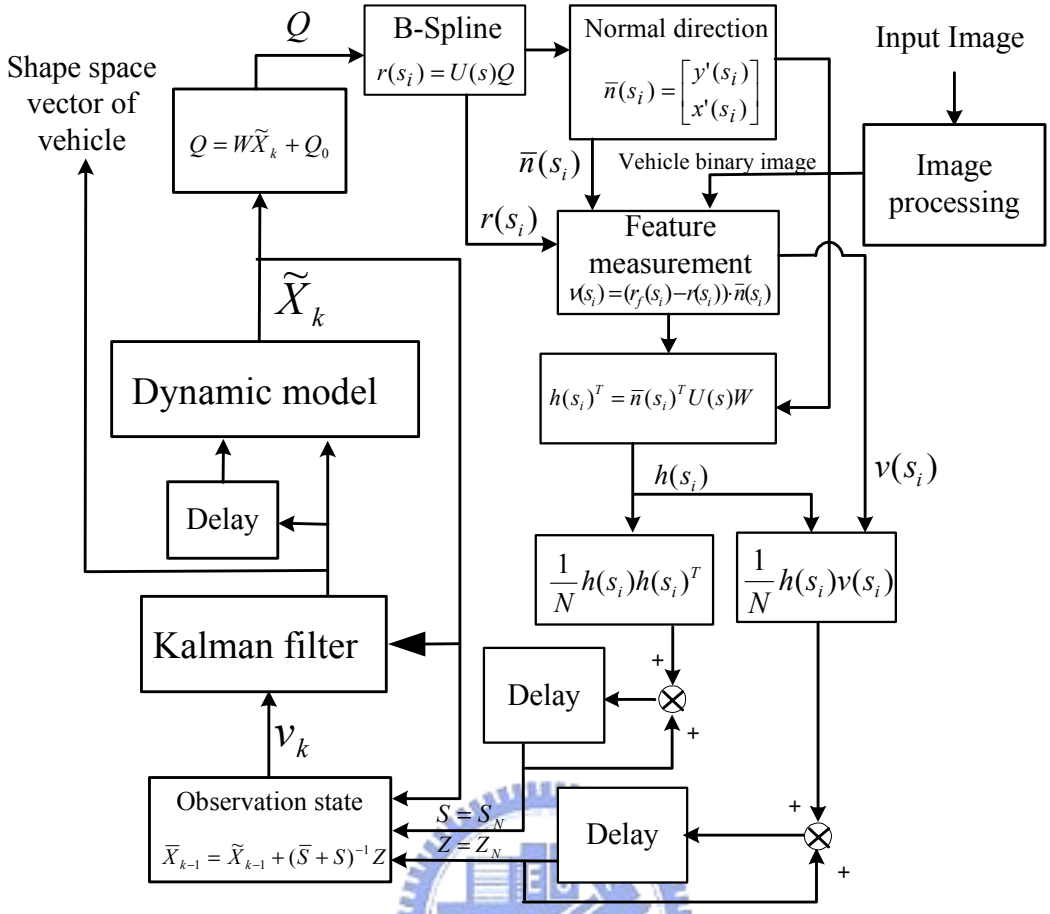


Fig. 5-7. Block diagram of the vehicle tracking system.

$$vel = \frac{L}{(t_b - t_a)}. \tag{5.22}$$

5.4 Turn Ratio Measurement

For estimating the vehicle turn ratio, two detection windows are set up to detect oncoming vehicles turning left and moving straight, respectively. The driving direction of vehicles that travel in the left-turn detection window, for instance, consists of three different types of vehicles – moving right, moving straight ahead, and moving left, as depicted in Fig. 5-8. A rectangular area that covers the left-turn detection window is placed in the lower right region as shown in the figure. An oncoming vehicle that is turning left travels to the right in the rectangular area of Fig. 5-8(a). Two vehicles moving straight pass through the rectangular area of Fig. 5-8(b). Their driving direction is straight ahead. In Fig. 5-8(c), two vehicles

traveling straight ahead in the cross-lane direction travel to the left in the rectangular area. Only the oncoming vehicles turning left must be monitored in this case. As noted above, the motion vector estimation module is designed to measure and classify the driving direction of vehicles in the detection window based on an optical flow measurement. The method of motion vector estimation will be described in Section 5.4.1.

The counting of vehicles by using image processing procedures described in the previous section facilitates the calculation of turn ratio. In this design, the turn ratio at an intersection is measured automatically. For instance, left-turn ratio is defined as

$$\text{turn ratio} = \frac{N_{left}}{N_{total}}, \tag{5.23}$$

where N_{total} is the total count of detected vehicles in a time period and N_{left} is the count of detected left-turn oncoming vehicles in the time period. In Fig. 5-9, two detection windows are set up to detect and count the passing vehicles at a T-shape intersection. The detection window needs to be properly set in the situation that the bottom of each vehicle is recognizable to each other in the detection window. Otherwise, the detection window cannot work with functionary accuracy. For instance, if one set a detection window near the stop line marking in the figure, the detection window incorrectly detects the vehicle because the bottom of vehicle is not recognizable. It can be seen that N_{left} is measured in the lower-right window. Note that vehicles in this detection window involve three types of motion – oncoming vehicles turning left, ongoing vehicles moving straight, and vehicles of the cross-lane moving straight. Only the oncoming vehicles turning left need to be monitored for the left-turn ratio calculation. It is clear that the motion direction of different types of vehicles is considerably different. The motion direction reveals where the moving vehicles come from. In other words, information about the vehicle motion direction indicates the type of vehicles at an intersection. Furthermore, vehicles passing through the left detection window involve two types of

motion – oncoming vehicles moving straight, and vehicles of the cross-lane turning left. The driving direction of vehicles in the left-turn window is also used to determine whether the system detect the vehicles that pass through the left detection window.

(a)

(b)

(c)

Fig. 5-8. Three types of vehicle driving directions: (a) Moving right. (b) Moving straight ahead. (c) Moving left.

5.4.1 Corner-Based Optical Flow Estimation

The vehicle motion field can be computed by identifying corresponding pairs of points in two successive image frames taken at time t and $t+\Delta t$. The points must be sufficiently distinct so that they can be identified and located in both images. Corner features are selected as the identified points. In this design, the Harris corner detector is adopted to extract the corner points from traffic imagery because of their superior repeatability, robustness to viewpoint changes, and resistance to illumination variation [78].

Once a set of interesting points $\{P_i\}$ is identified in the image frame I_1 taken at time t, the corresponding points must be identified in the successive image frame I_2 taken at time $t+\Delta t$. As shown in Fig. 5-10, given a sample point P at *(Tx,Ty)* in image frame I_1, we take a square block centered at P in I_1 and find the best-correlated block in I_2 under the assumption that the amount of movement is limited. If the best match is found at *(Hx,Hy)*, then the point *(Hx,Hy)* will be the destination of point P. The vector from *(Tx,Ty)* to *(Hx,Hy)* is the motion vector of point P.

In the searching process described above, it is assumed that pixels belonging to the block are displaced with the same amount. However, this assumption sometimes fails to hold in a traffic scene. The vehicle image gradually changes due to the viewpoint of camera and vehicle motion behavior. It is difficult to obtain the same blocks of a moving object in two successive frames. Still, a certain similarity exists between these two blocks. The more the corresponding pixels that have similar intensities, the more correspondence exists between these two blocks. We use a minimum absolute difference of individual pixels as an error criterion for examining the correspondence between two blocks [79]. For two $N \times N$ blocks A and B, the matching distance is defined by

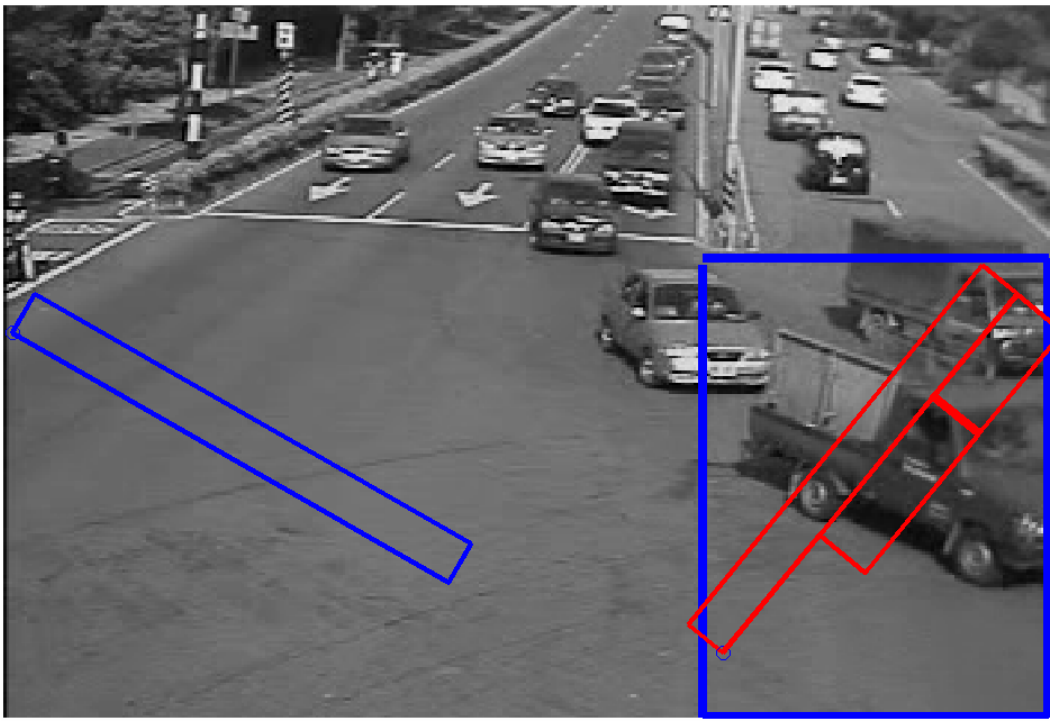


Fig. 5-9. Detection windows.

$$D = \sum_{1 \le r \le N} \sum_{1 \le c \le N} d(r,c), \tag{5.24}$$

where $d(r,c)$ is 1 if $|A(r,c) - B(r,c)| < T$, otherwise, 0; T is a threshold value. The matching distance is used as the similarity index of two blocks. Two blocks that produce the largest matching distance are the best match.

For a $N \times N$ block with maximum absolute displacement represented by [$d_r^{\max}$ $d_c^{\max}$], a full search calls for evaluation of the matching criterion at $(2d_r^{\max}+1)\times(2d_c^{\max}+1)$ blocks. Searching for the best-match block might fail to fulfill the real-time requirement. In order to speed up the search process in this design, the search operation is executed only on blocks containing a detected corner. As shown in Fig. 5-10, there are two corner points in the search region of image I_2. Thus, the matching algorithm performs evaluation of the matching check only at $2N^2$ blocks. In practical realization, the absolute displacement was selected as

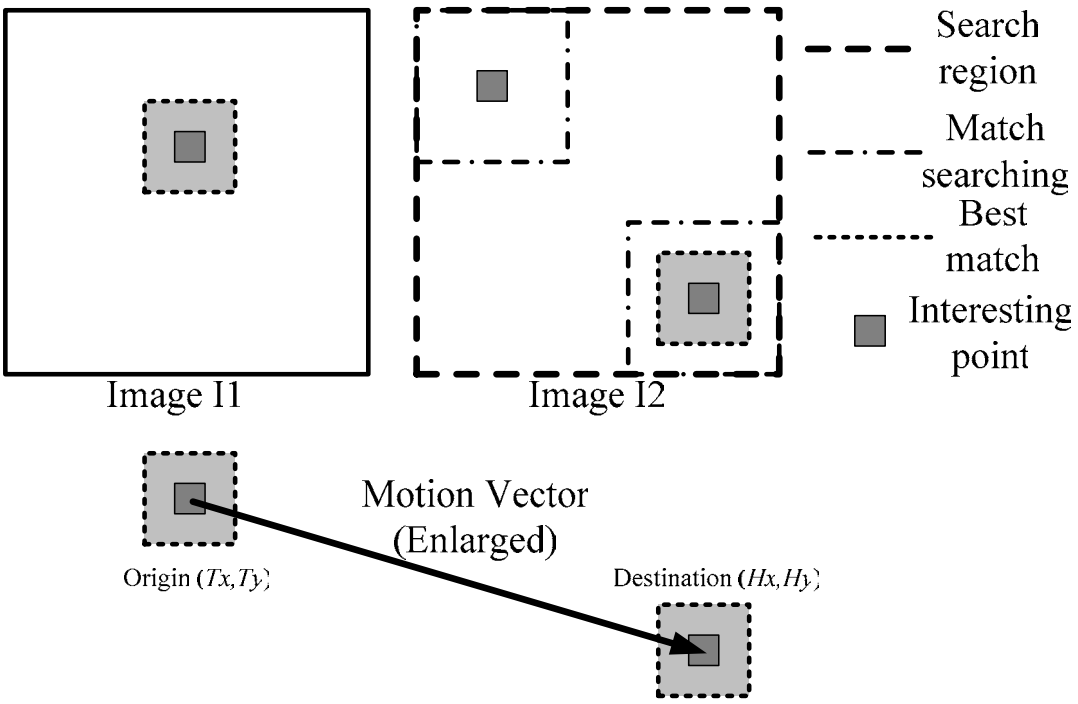


Fig. 5-10. Motion vector detection.

$d_r^{\max} = d_c^{\max} = 5$ and $N = 3$, the evaluation execution cycles are reduced from 121 to 18 times.

The design procedure of the average direction measurement of moving vehicles using the proposed optical flow method is summarized as follows:

1) Use a Harris corner detector to extract the corner points from the current image frame. Each extracted corner point and its neighboring pixels are selected as check points.

2) Record the positions of all corner points as well as the check points for the correlation operation.

3) Choose a 3×3 region centered at one corner point of the preceding image frame.

4) Get a $N \times N$ region centered at the corresponding corner point of step 3 from the current image frame. The sub-region centered at the check point is assigned as check sub-region in the selected region.

5) Measure the matching distance between the 3×3 region of step 3 and all check sub-regions of step 3.

6) If the maximum matching distance of step 5 is greater than a specific threshold, the center of the check sub-region with which the matching distance is the maximum is set as a new position of the corresponding corner point of step 3 in the current image frame. Thus, the motion vector of the corner point in the preceding image frame is obtained.

7) Repeat steps 3-6 to estimate all motion vectors of corner points in the preceding image frame.

Figure 5-11 depicts a test result of the optical flow estimation. In the figure, small arrows indicate the detected motion direction of interesting points of vehicles. An average motion vector is obtained by taking into account motion vectors of all interesting points. Some errors can be observed in the figure. These errors are mainly caused by imperfect corner selection and match searching. However, such estimation errors can be averaged out in most cases and the vehicle moving direction can still be obtained for motion-type discrimination with high accuracy. In Fig. 5-11, the direction of the big arrow indicates the average motion vector of

Fig. 5-11. Result of optical flow estimation.

vehicles in the specific region. It accurately reflects the motion type of vehicles in this region. According to the information of the estimated direction, the system can identify where the vehicles come from and determine the turn ratio accordingly.

5.5 Summary

An automatic contour initialization procedure has been developed for image tracking of multiple vehicles based on active contour and image measurement approach. A method is proposed to detect moving vehicles of various sizes and generate their initial contours for image tracking in a multi-lane road. The proposed method is not constrained by the lane boundary. The automatic contour initialization and tracking scheme have been proposed for traffic monitoring. Moreover, in company with motion vector estimation, the special designed window can be used to measure the vehicle turn ratio in real-time. An algorithm for estimating motion vector based on corner correlation has been designed to meet the real-time requirement.

Chapter 6

Experimental Results

6.1 Introduction

In this chapter, five experimental results of traffic parameter estimation are presented and evaluated. First three experiments under a rare shadow condition evaluate the algorithms of vehicle detection and tracking. First, cars and motorcycles are simultaneously detected and tracked by the proposed method. Second, vehicles are detected and tracked for traffic parameter estimation. Thirdly, a stand-alone traffic surveillance system has been implemented to estimate the vehicle turn ratio at an intersection. The rest experiments are concerned with not only the performance of tracking algorithm but also the capability of shadow suppression. One test evaluates the shadow suppression and the tracking performance of the proposed algorithm, and the other verifies the proposed method for turn ratio estimation. The test video images under different shadow conditions were recorded in advance at an expressway as well as an intersection near National Chiao Tung University. In both cases shadows attached to their respective moving vehicles will degrade the performance of traffic monitoring if the ITMS has no shadow suppression function. Nevertheless, the ITMS should be able to detect and track the moving vehicles even when the shadows appear in the traffic scene. In the experiments, we not only validate the feasibility of the shadow suppression algorithm but also estimate traffic parameters such as traffic flow, traffic density, vehicle speeds, and vehicle turn ratios with functional accuracy.

The rest of this chapter is organized as follows. Four experiments under a rate shadow

condition are performed to examine the performance of the proposed method for traffic parameter estimation in Section 6.2. Practical experimental results of traffic parameter estimation under shadow condition are presented in Section 6.3. Section 6.4 shows the experiment of turn ratio estimation under shadow condition. Section 6.5 gives some concluding remarks.

6.2 Traffic Parameter Estimation

6.2.1 Vehicle Tracking

Practical experiments of traffic parameter extraction have been conducted to evaluate the tracking performance of the proposed method by using two video clips of the traffic scene. The frame rate adopted in both experiments is 15 frame/s. The pixel resolution of each test frame is 352x240 pixels. Fig. 6-1 illustrates an example of image tracking of cars and motorcycles. In Fig. 6-1(a), two motorcycles are tracked and another motorcycle is detected in the detection window. Fig. 6-1(b) shows that the motorcycle leaves the detection window and an initial contour is generated accordingly in the initialization region. In Fig. 6-1(c), two motorcycles are tracked and another motorcycle has not yet entered the detection window. A motorcycle is detected in the detection window in Fig. 6-1(d). The motorcycle leaves the detection window and an initial contour is generated accordingly in Fig. 6-1(e). Another motorcycle is detected in Fig. 6-1(f). Two motorcycles are simultaneously detected in Fig. 6-1(g). One motorcycle leaves the detection window and an initial contour is generated in Fig. 6-1(h). A car is detected in Fig. 6-1(i). The car leaves the detection window and an initial contour is generated for tracking in Fig. 6-1(j). The car is tracked while another car is detected in Fig. 6-1(k). The car is tracked as expected and the other two cars are detected simultaneously in Fig. 6-1(l). The experimental results demonstrate that the proposed contour initialization procedure successfully provides initial contours for vehicle contour tracking. Moreover, this experiment indicates that cars and motorcycles are detected and tracked

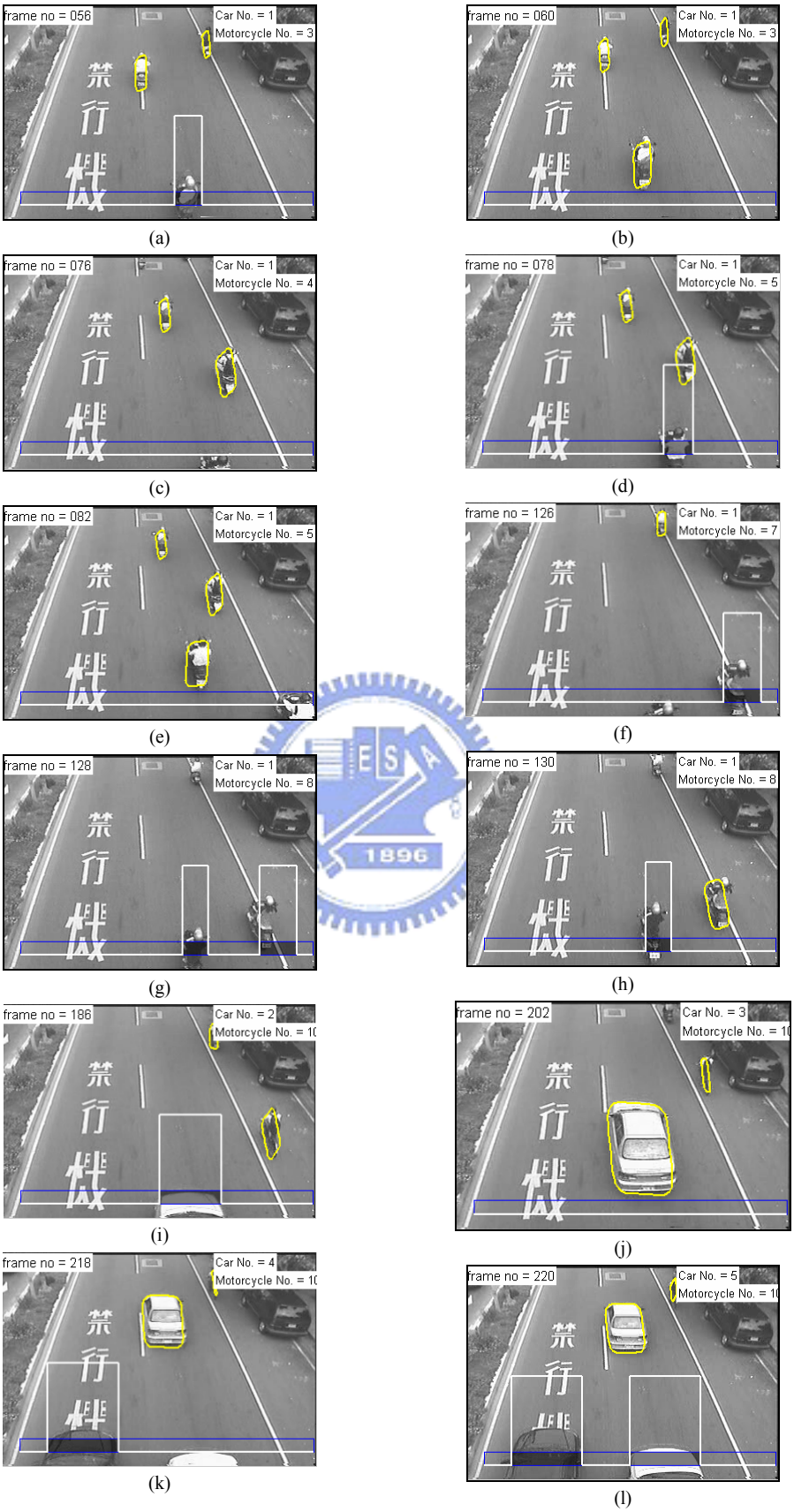


Fig. 6-1. Experimental results of image tracking of cars and motorcycles.

simultaneously in any portion of a multi-lane road by the proposed method. A video clip of experimental results can be found at: http://isci.cn.nctu.edu.tw/video/JCTai/tracking.avi.

6.2.2 Vehicle Tracking and Traffic Parameter Estimation

The second experiment was conducted using a traffic scene recorded from a surveillance camera installed at the main gate of the Hsinchu Science Park, where a traffic monitoring and control system was implemented. Figure 6-2 shows the experimental results of vehicle tracking for traffic parameters estimation. In Fig. 6-2(a), a car is detected in the detection window. The first detected car leaves the detection window and an initial contour is generated accordingly in Fig. 6-2(b). In Fig. 6-2(c), two cars are tracked and another detected car passes through the detection window. Figure 6-2(d) shows that the detection window detects a vehicle in the left lane. Two cars simultaneously cross the detection window and are detected in Fig. 6-2(e). Cars leave the detection window and are tracked in Fig. 6-2(f). It is interesting to note that two tracked cars are close to each other and still being tracked precisely in Fig. 6-2(g). In Fig. 6-2(h), a car traveling between two lanes is detected in the detection window.

In this example, useful traffic parameters are estimated using the proposed method. In a time duration of 20 minutes, a total of 335 vehicles were detected (the ground truth was 334 vehicles). The accuracy of vehicle number estimation is very high, thanks to the detection window for handling vehicles that travel across lane boundaries. Table 6-1 shows the experimental results of traffic parameter estimation. In the table, ground truth, which was manually measured from image sequences, is also presented for comparison. The estimated parameters are as follows: the average speed was 42.87 km/hr (the ground truth was42.88 km/hr), the traffic flow rate was 1002 car/hr (the ground truth was 1005 car/hr) and the density was 23.37 car/km (the ground truth was 23.44 car/km). Because the initial contour

generated by the proposed method is highly suited for active contour-based image tracking, the error of average speed estimation is within 5%. A video clip of experimental results can be found at: http://isci.cn.nctu.edu.tw/video/JCTai/Tracking2.mpg

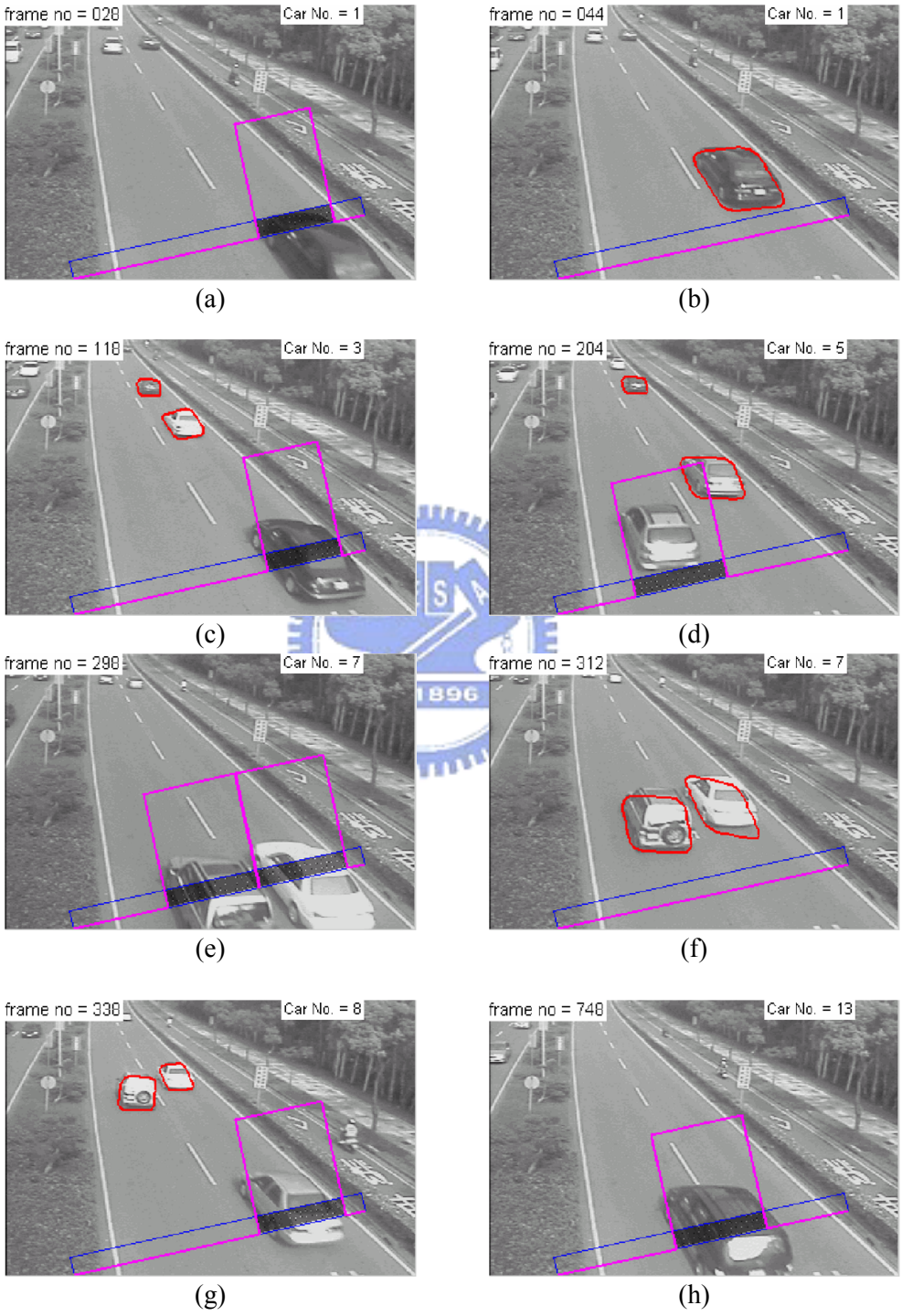


Fig. 6-2. Image tracking results of cars.

Table 6-1 Experimental Results of Traffic Parameter Estimation

Time index (min)	Vehicle number (car)		Traffic flow rate (car/hr)		Average speed (km/hr)			Traffic density (car/km)	
	Ground truth	Estimated	Ground truth	Estimated	Ground truth	Estimated	error	Ground truth	Estimated
1	13	14	780	840	36.9	35.6	3.51%	21.1	23.6
2	38	38	2280	2280	44.0	42.2	4.00%	51.9	54.0
3	6	6	360	360	53.3	51.1	4.17%	6.8	7.0
4	2	2	120	120	43.1	42.9	0.47%	2.8	2.8
5	38	37	2280	2220	42.5	41.7	2.03%	53.6	53.3
6	12	12	720	720	43.5	43.2	0.82%	16.5	16.7
7	15	15	900	900	41.2	41.8	-1.47%	21.9	21.6
8	21	21	1260	1260	43.0	43.8	-1.84%	29.3	28.8
9	5	5	300	300	49.8	50.8	-2.16%	6.0	5.9
10	23	23	1380	1380	36.9	35.6	3.51%	37.4	38.8
11	4	4	240	240	45.6	46.0	-0.80%	5.3	5.2
12	23	23	1380	1380	38.7	38.4	0.77%	35.7	36.0
13	26	27	1560	1620	40.2	40.2	0.01%	38.8	40.3
14	5	5	300	300	48.8	49.6	-1.67%	6.2	6.1
15	30	30	1800	1800	41.6	42.1	-1.28%	43.3	42.7
16	22	22	1320	1320	46.2	47.0	-1.59%	28.6	28.1
17	1	1	60	60	44.6	44.5	0.24%	1.3	1.3
18	27	28	1620	1680	39.3	39.9	-1.48%	41.2	42.1
19	12	12	720	720	45.7	46.8	-2.27%	15.7	15.4
20	11	10	660	600	32.8	34.4	-4.69%	20.1	17.5

6.2.3 Turn Ratio Estimation

6.2.3.1 System Architecture

A stand-alone traffic surveillance system has been installed at a main intersection in the Hsinchu Science Park, where the traffic condition is very often congested. A CCD camera is used to monitor the T-shape signal-controlled intersection. The turn ratio as well as the traffic flow parameters of the intersection are measured. This onsite traffic monitoring system uses a USB 2.0 image grabber to acquire the traffic scene. The monitoring system employs two detection windows to detect oncoming vehicles, as shown in Fig. 6-3. One detection window

detects the vehicles moving straight and the other detects the vehicles turning left. The measured vehicle count is automatically uploaded to a SQL server every minute. The collected data are used for real-time traffic parameter extraction. Fig. 6-4 depicts the display of the turn ratio measurement system. The upper-left region of the display is the real-time image; the upper-right region is the measurement result. The lower-right region is the background image; the lower-left region shows the vehicle count and the elapsed time.

For the T-shape signal-controlled intersection, the traffic control signals consist of three phases: one for vehicles moving straight, the other for oncoming vehicles turning left, and another for vehicles in the cross lanes. In this experiment, only oncoming vehicles need to be monitored in the traffic imagery. As shown in Figs. 6-5-7, the motion vector of lower-right region is measured to identify where the vehicles come from. It is clear in these figures that the developed ITMS successfully estimates three types of motion vectors. A video clip of experimental results can be found at: http://isci.cn.nctu.edu.tw/video/JCTai/Turn_ratio.mpg.

6.2.3.2 Turn Ratio Measurement

From a seven-minute video, useful traffic parameters are extracted using the developed ITMS in real time. Results of traffic flow estimation are shown in Fig. 6-8. The upper half of Fig. 6-8 presents the counting result of vehicles moving straight. In the figure, the actual number of vehicles, which were manually counted from image sequences, is also presented for comparison. In this test, 57 vehicles moving straight were recorded passing through the detection window. The system detected 59 vehicles with an error of 3.5%. The lower half of Fig. 6-8 presents the count of vehicles turning left. There were 53 vehicles recorded passing through the detection window, and the system detected 57 vehicles with an error of 7.5%. The results of turn ratio measurement are depicted in Fig. 6-9. The upper half of Fig. 6-9 shows

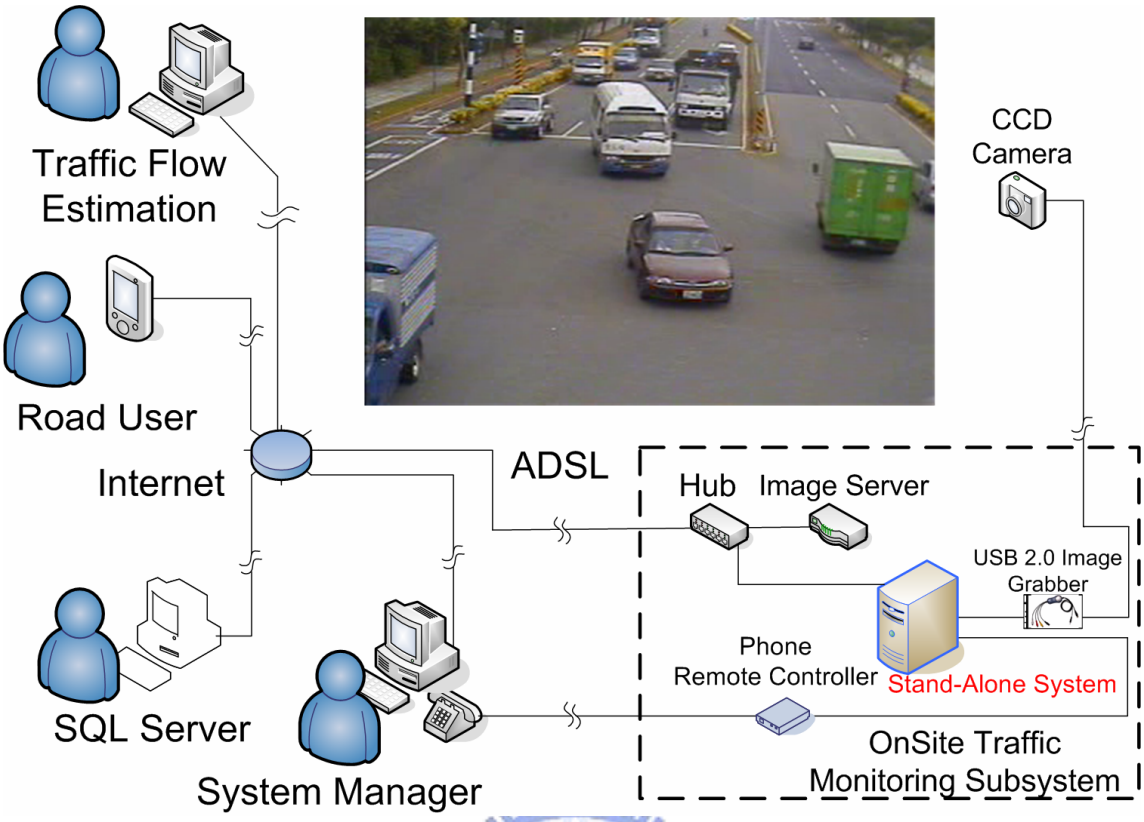


Fig. 6-3. System architecture of turn ratio measurement.

Fig. 6-4. The display of traffic monitoring.

Fig. 6-5. Motion estimation results of ongoing vehicles vehicles moving straight ahead.

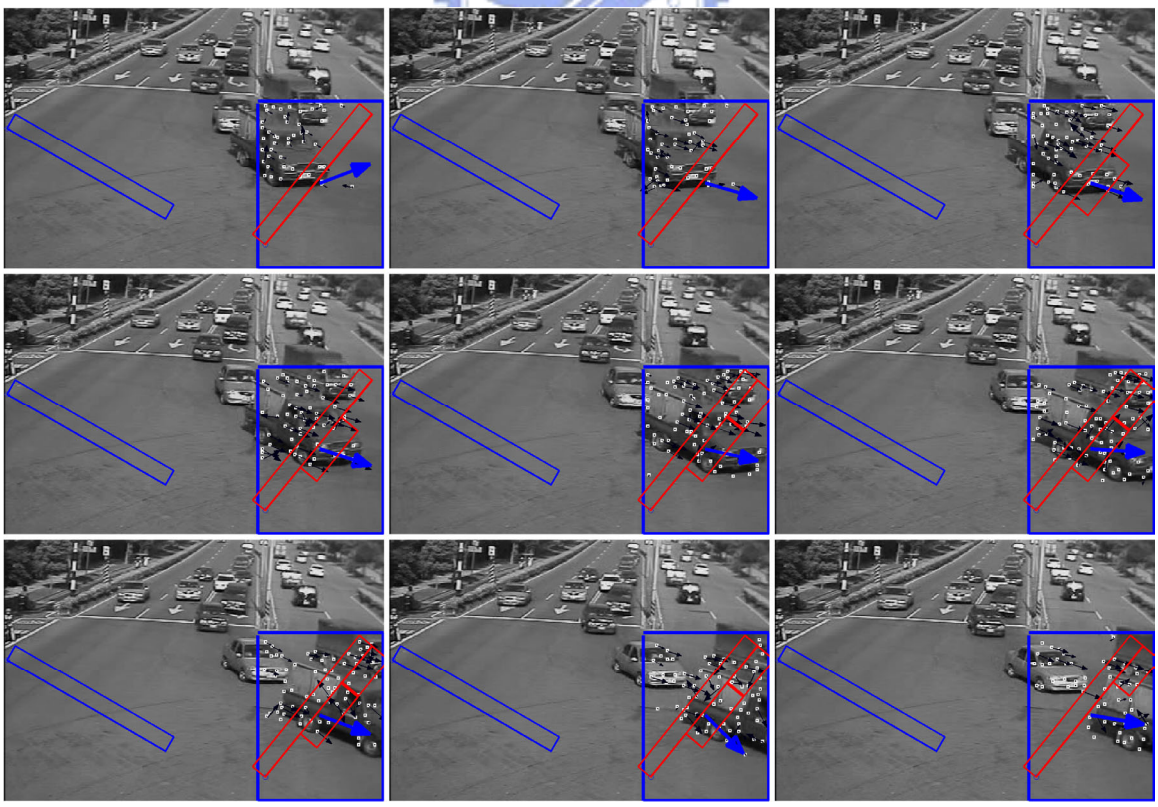


Fig. 6-6. Motion estimation results of oncoming vehicles turning left.

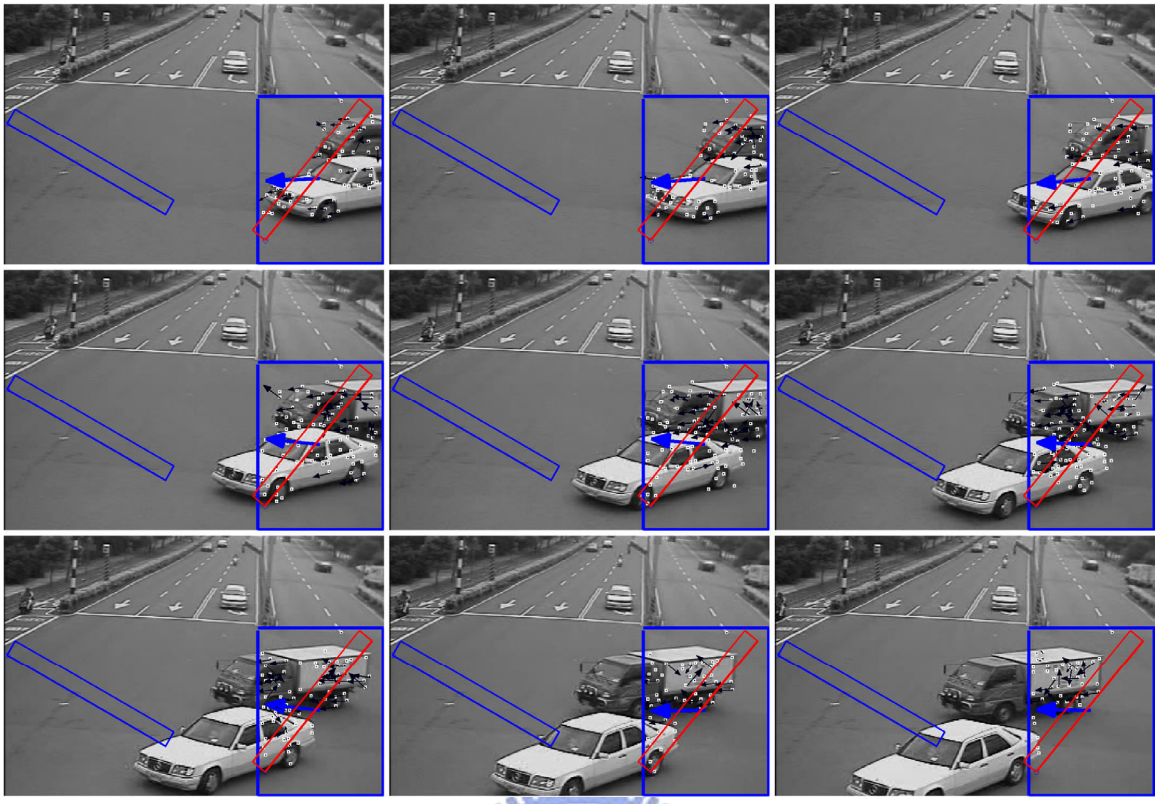


Fig. 6-7. Motion estimation results of vehicles of the cross-lane moving straight ahead.

both the measured and the actual turn ratios. The error of the turn ratio measurement is also shown in the lower half of Fig. 6-9. In this setting, the actual turn ratio is 48.18% and the measured is 49.14%. The average error is 2%. The measurement error is mainly caused by the imperfect installation site of the camera. The same motion that appears in different regions has different displacements in the image frame. Since the installation site of the camera is too close to the monitored vehicles in the test, the motion of vehicles that are near the camera will have a larger displacement than the motion of vehicles further away in the image frame. Consequently, in the matching process, the displacement of the nearer vehicles will probably exceed the prescribed searching range and cause errors in the matching process. This problem will be improved if a better site can be selected for the camera in use.

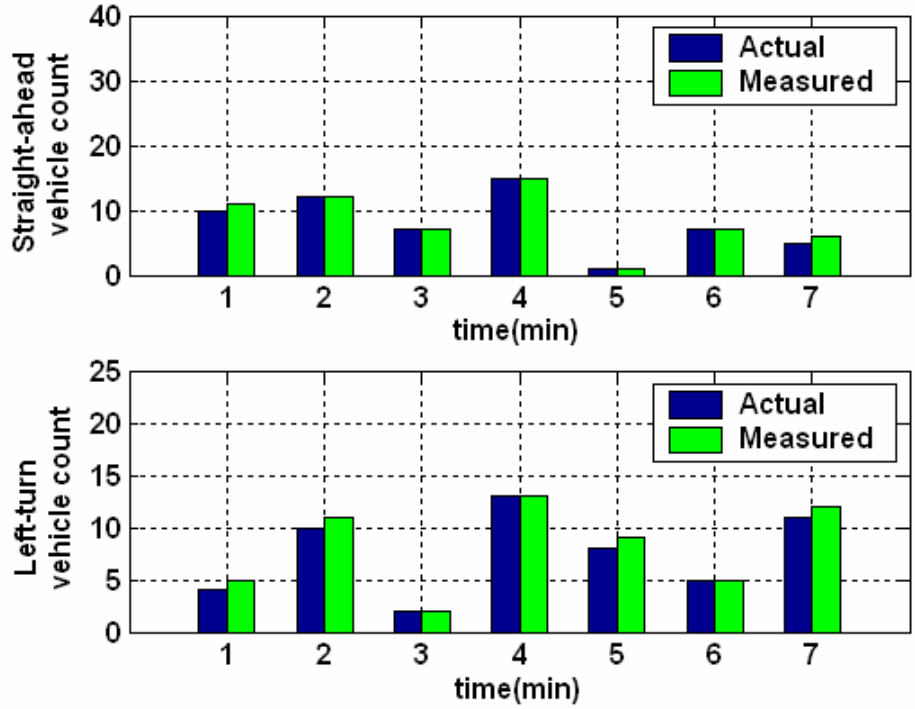


Fig. 6-8. Experimental results of traffic flow measurement.

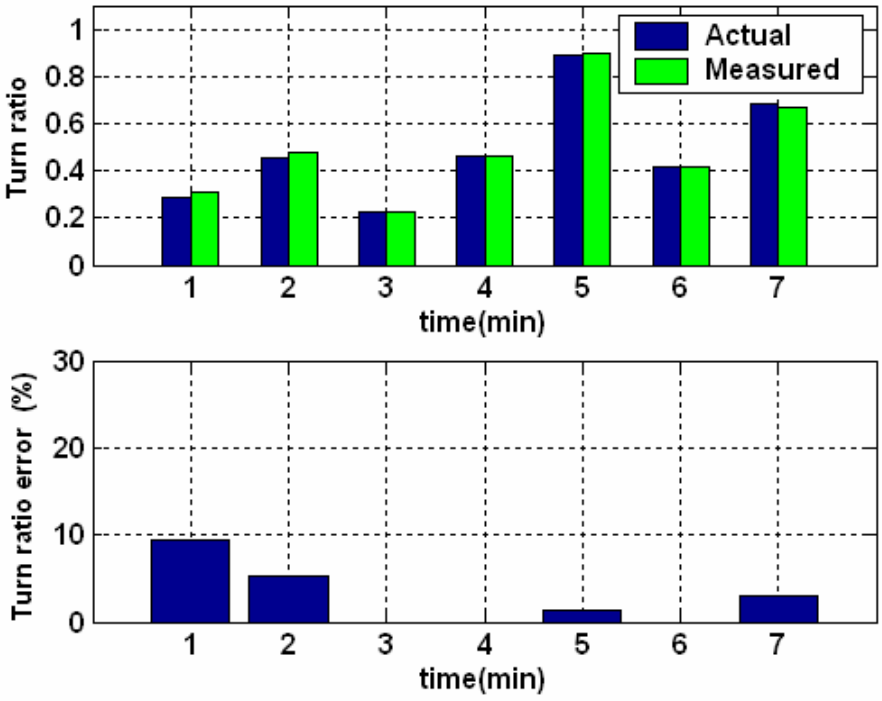


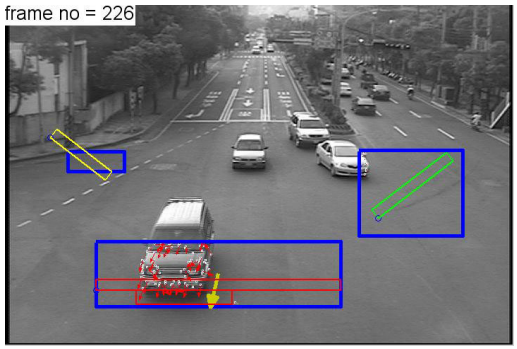
Fig. 6-9. Experimental results of turn ratio measurement.

6.2.3.3 Turn Ratio Estimation at A Cross Intersection

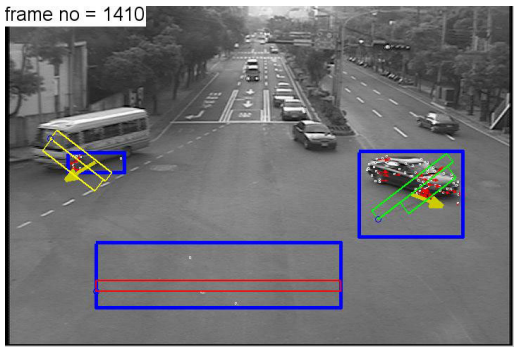
In this experiment, the turn ratios of oncoming vehicles at a cross intersection are estimated. The monitoring system employs three detection windows to detect oncoming vehicles, as shown in Figs. 6.10(a)-(d). One detection window detects the vehicles moving straight, another detects the vehicles turning left and the other detects the vehicles turning right. Excluding the oncoming vehicles, ongoing vehicles moving straight and vehicles of the cross-lane moving straight also pass through these detection windows. However, only the oncoming vehicles should be detected for turn ratio estimation. It is necessary to use the developed optical flow method to estimate the driving direction for enabling the vehicle detection. The motion information can be used to determine where the vehicles that pass through the detection window come from. Three specific regions that cover the detection windows are set for the motion estimation, as shown in Figs. 6.10(a)-(d). Accordingly the oncoming vehicles can be discriminated and detected for turn ratio estimation.

Figures 6.10(a)-(d) depict a test result of vehicle detection and motion vector estimation. In the figures, the small arrows indicate the detected motion direction of interesting points of the vehicles. An average motion vector is obtained by averaging all estimated motion vectors of all interesting points in the specific regions. Figure 6.10(a) shows that an ongoing vehicle moving straight is detected by using the detection window and the driving direction of the vehicle is also estimated correctly. As shown in Fig. 6.10(b), a vehicle moving right and a vehicle moving left pass through their respective detection window. The driving directions are estimated successfully and enable the detection windows to detect the vehicle. Figures 6.10(c)-(d) shows that the motion information is used to disable the detection windows when vehicles of the cross-lane moving straight pass through the detection windows.

In a green-light period, 24 oncoming vehicles pass through the detection windows and

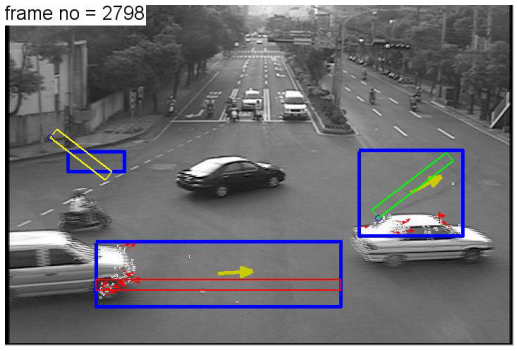


(a)



(b)

(c)



(d)

Fig. 6-10. Experimental results of turn ratio estimation at a cross intersection.

the system detects 25 vehicles with an error of 6.25%. There are 7 vehicles moving left that pass through the detection window and the vehicles are successfully detected by the system. The optical flow estimation provides the motion information to the system for the vehicle detection. The vehicles moving right are also processed in the same way. The actual left turn ratio is 28% and the actual right turn ratio is 4%, both errors are 4%. Partial results can be found at: http://isci.cn.nctu.edu.tw/video/JCTai/Tr_Jctai.wmv.

6.3 Vehicle Tracking and Traffic Parameter Estimation with Shadow Suppression

Figure 6-11 shows the system architecture of the proposed traffic parameter estimation system with shadow suppression. The system consists of five parts: the foreground segmentation module, shadow suppression module, vehicle detection module, vehicle tracking module, and traffic parameter estimation module. The foreground segmentation module constructs the background image for foreground segmentation from image sequences, as described in Chapter 3. The shadow suppression module separates the cast shadow from the moving regions for improving the quality of foreground segmentation, as presented in Chapter 4. The vehicle detection module detects the moving vehicles for vehicle counting and tracking initialization, as described in Chapter 5. The vehicle tracking module tracks the moving vehicle for speed estimation. Finally the traffic parameter estimation obtains useful traffic parameters by analyzing detection and tracking result, as shown in Section 5.3.3.

The frame rate adopted in the experiment is 15 frames per second. The pixel resolution of each test frame is 352x240 pixels. Figure 6-12 illustrates an example of image tracking of cars and trucks on an intercity expressway. In the scene, the shadow strength is medium, and the shadow size is large. The ITMS will fail to detect and track the moving vehicles if three is no shadow suppression tool in use. Figs. 6-12(b)-(l) shows that the proposed method successfully separates the shadow from the moving region. In Fig. 6-12(b), two cars are

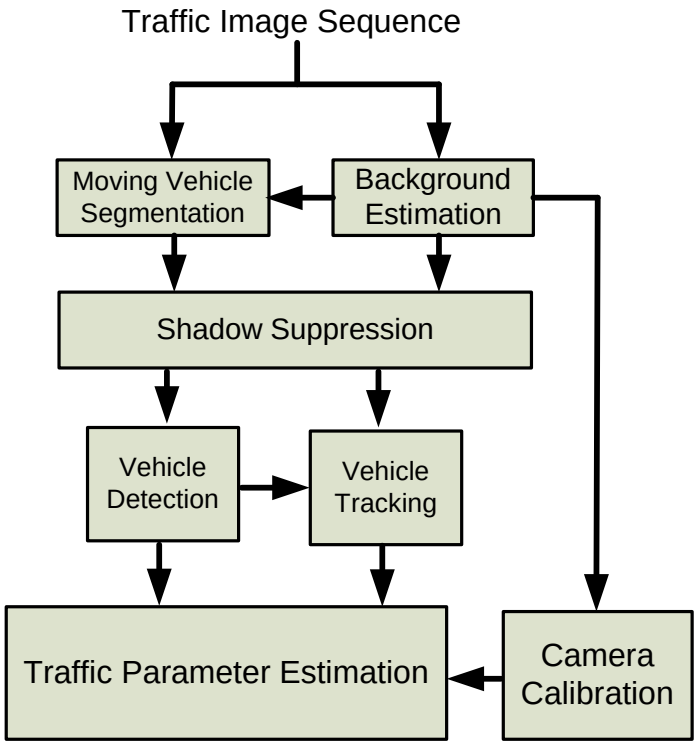


Fig. 6-11. Block diagram of the traffic parameter estimation system.

detected in the detection window, and one car is tracked. One car leaves the detection window and an active contour is initiated for tracking in Fig. 6-12(d). Three cars are tracked in Fig. 6-11(f). One car is tracked, and a truck is detected in the detection window, as shown in Fig. 6-12(h). The truck leaves the detection window and an initial contour is generated for tracking in Fig. 6-12(j). In Fig. 6-12(l), the truck is tracked and a car is detected in the detection window. The experimental results demonstrate that the traffic monitoring system successfully tracks the moving vehicles with shadow suppression. Partial results can be found at: http://isci.cn.nctu.edu.tw/video/JCTai/shadow_track.wmv.

In this example, useful traffic parameters are estimated. In a time duration of 23 seconds, a total of 16 vehicles were detected (the ground truth is 18 vehicles). The accuracy of vehicle number estimation and tracking are satisfactory, thanks to the shadow suppression and the detection window for handling vehicles that travel across lane boundaries. Table 6-2 shows the experimental results of the estimation of vehicle speed. In the table, the ground truth,

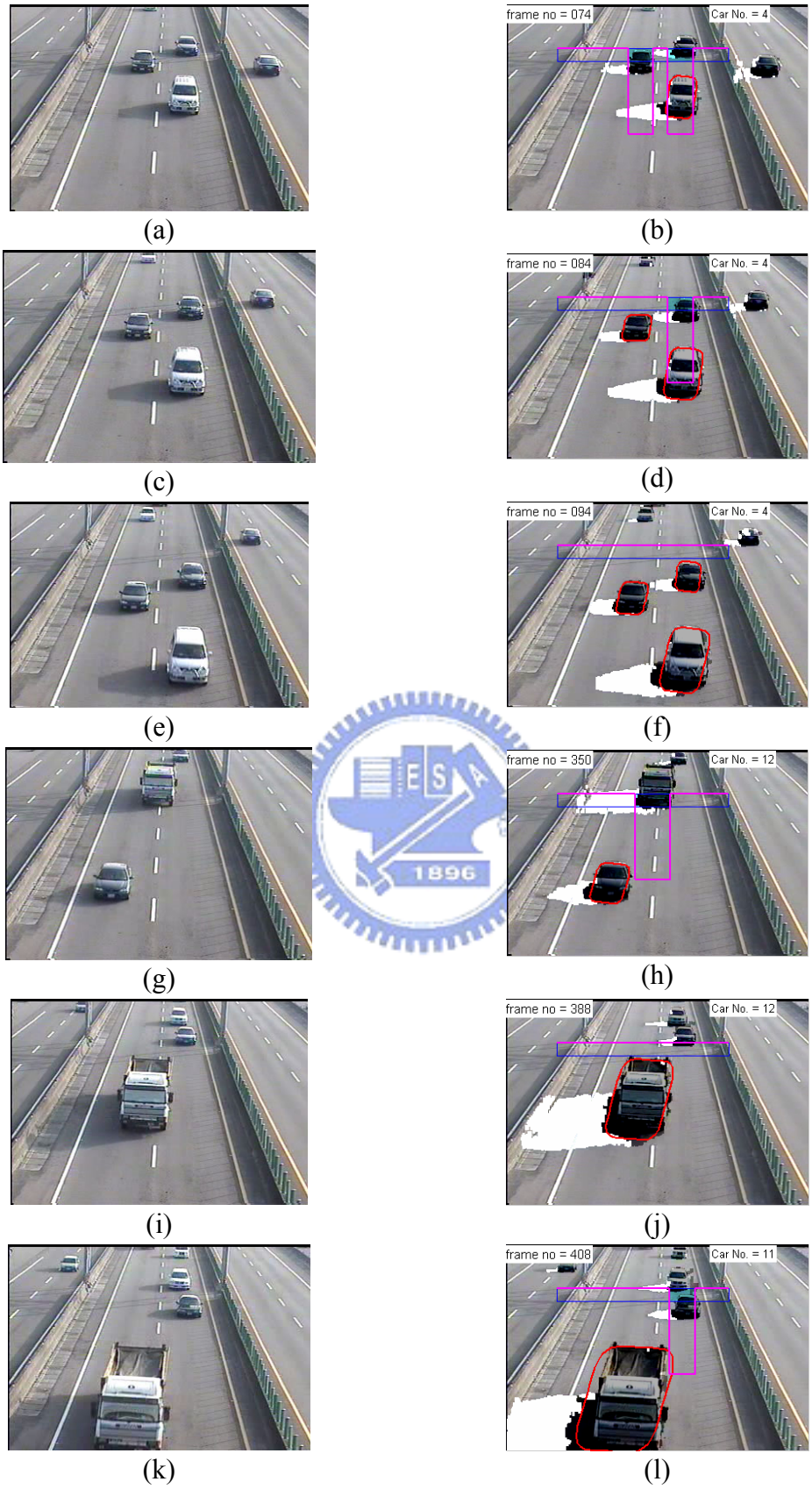


Fig. 6-12. Experimental results of vehicle tracking with shadow suppression in an expressway. (Detected shadows are indicated in white area and detected vehicles are indicated in darker area.)

Table 6-2 Experimental Results of Vehicle Speed Estimation

Vehicle Index	Estimated speed (km/hr)	Ground truth (km/hr)	Error %
1	84.1	80.6	4.3%
2	86.0	84.9	1.4%
3	86.4	85.7	0.8%
4	84.9	86.8	-2.2%
5	86.8	86.4	0.4%
6	101.3	103.5	-2.2%
7	82.9	87.6	-5.3%
8	100.6	95.7	5.1%
9	73.1	73.1	0.0%
10	75.0	76.8	-2.3%
11	77.0	77.3	-0.4%
12	79.3	77.6	2.2%
13	84.1	82.9	1.4%
14	54.9	54.0	1.7%
15	66.0	66.6	-0.9%
16	64.8	64.5	0.5%
Average	80.4	80.2	0.3%

which was manually measured from image sequences, is also presented for comparison. Because the initial contour generated by the proposed method is well-suited for the active-contour-based image tracking with shadow suppression; the error of average speed estimation is within 5%. The traffic parameters are estimated as follow: the average speed is 80.4 km/hr (the ground truth is 80.2 km/hr), the traffic flow rate is 2504.3 car/hr and the density is 31.3 car/km.

6.4 Turn Ratio Estimation with Shadow Suppression

Figure 6-13 shows the system architecture of the proposed turn ratio estimation system. The system consists of five parts: the foreground segmentation module, shadow suppression module, vehicle detection module, optical flow detection module, and vehicle turn ratio module. The first three preceding modules are the same as described in Fig.6-12. The direction of movement detected by the optical flow detection module is used to determine whether the detection window detects the moving vehicle, as presented in Section 5.4. The turn ratio module computes the turn ratio by using the vehicle count of each vehicle detection module, as described in Section 5.4.

In this experiment, the left turn ratio of oncoming vehicles is estimated at a T-shape intersection as shown in Fig. 6-14. Figures 6-14(a)-(c) depict a test result of shadow suppression and the motion vector estimation. The image sequences are captured during

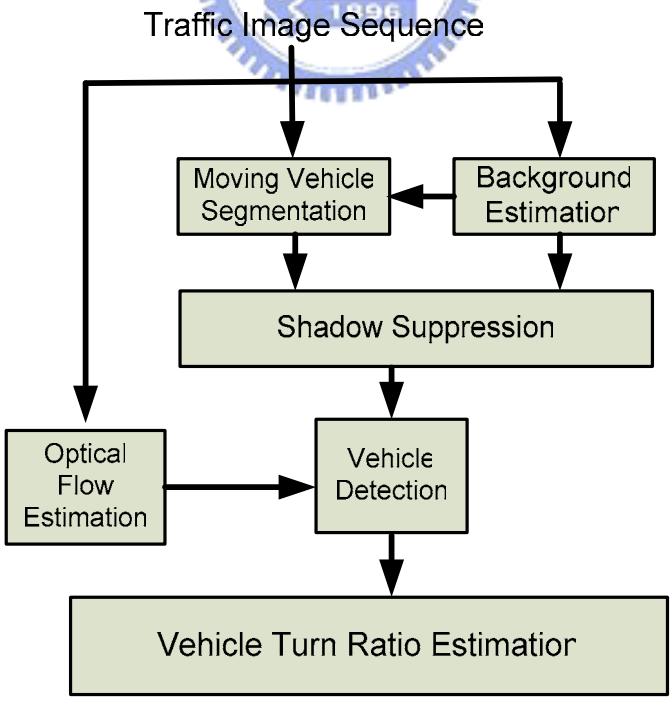


Fig. 6-13. Block diagram of the vehicle turn ratio estimation system.

(a)

(b)

(c)

Fig. 6-14. Experimental results of turn ratio estimation with shadow suppression at an intersection. Three types of driving direction of vehicles are distinguished: (a) Moving right. (b) Moving left. (c) Moving straight ahead.

sunset. In the scene, the shadow strength is large, and the shadow size is medium. The actual shadow has been successfully separated from the moving vehicle by the proposed method. In the figures, the small arrows indicate the detected motion direction of interesting points of the vehicles. An average motion vector is obtained by considering the motion vectors of all interesting points in the specific region. The direction of the big arrow shows the average motion vector of vehicles in the specific region. Figure 6-14(a) shows two oncoming vehicles

passing the specific region. One vehicle moves left and passes through the detection window in Fig. 6-14(b). Figure 6-14(c) shows an ongoing vehicle passes through the specific region. The detected motion accurately reflects the motion type of vehicles in this region and the shadow has been suppressed in traffic monitoring. According to the information of the estimated direction, the system can identify where the vehicles come from and determine the turn ratio accordingly. In this test, the actual turn ratio is 7% and the calculated value is 7.2%. The averaged error is about 3.7%. Partial results can be found at: http://isci.cn.nctu.edu.tw/video/JCTai/shadow_ratio.wmv.

6.5 Summary

This chapter evaluates the proposed GBH, traffic parameter estimation, and shadow detection approaches through experimental results. Five experiments have been carried out to validate the performance of the proposed scheme. In the first three experiments, the vehicles are detected and tracked for traffic parameter estimation under a rare shadow condition. In the rest experiments, the proposed method of shadow suppression successfully removes the overlapping shadows attached to moving vehicles for traffic image analysis. The moving vehicles are successfully segmented using our proposed method, showing that our tracking algorithm and turn ratio estimation work satisfactorily under shadow condition. Traffic parameters are obtained with high accuracy.

Chapter 7

Conclusion and Future Work

7.1 Dissertation Summary

To measure reliable traffic parameters, a series of designs and implementations of image processing are proposed in this thesis. The extracted traffic parameters are average speed, traffic flow, traffic density, and turn ratio. The image processing methodologies consist of camera calibration, single Gaussian background modeling, vehicle detection and tracking, optical-flow-based turn ratio measurement, and shadow suppression.

We propose a novel algorithm to automatically calibrate a PTZ camera that overlooks a traffic scene. The geometric information of parallel lane markings is used to derive the focal length equation for camera calibration. The computed focal length is further used to obtain the pan and tilt angles of camera. Furthermore, an image processing procedure has been developed to locate the parallel lane markings. The simulation results reveal that the error rates of position estimation are within 2.3% under the presence of reasonable translational and height errors. In actual experiments, twelve feature samples in a road scene were selected for distance measurement. The maximum error of the distance measurement is within 5% and the absolute mean of the error is below 2.39%.

To estimate a single Gaussian model of background pixel in real time, a group-based histogram algorithm has been designed and implemented. The method is highly effective for building the Gaussian background model from traffic image sequences. It is robust to errors

caused by sensing noise and slow-moving objects. The computational load of the GBH is 35% less than the popular GMM approaches, and is thus more suitable for the real-time requirement of surveillance applications. The performance of the proposed algorithm has been evaluated and successfully applied to an ITMS.

An automatic contour initialization procedure has been developed for image tracking of multiple vehicles based on an active contour and image measurement approach. First, a detection-window-based method is proposed to detect moving vehicles of various sizes and generate their initial contours for image tracking in a multi-lane road. The proposed method is not constrained by the lane boundary. Kalman filtering techniques are then applied for active contour-based image tracking of various vehicles. The automatic contour initialization and tracking scheme have been tested for traffic monitoring. Experimental results reveal that the proposed method successfully tracks motorcycles as well as cars on an urban multi-lane road. Traffic parameters are successfully extracted by using the developed method. The error of average vehicle speed estimation is within 5%.

A method is proposed to automatically estimate the turn ratio at an intersection by using techniques of optical flow calculation and detection window. An algorithm for estimating motion vector based on corner correlation has been designed to meet the real-time requirement. The automatic turn ratio measurement has been tested for traffic monitoring in Hsinchu Science Park, Taiwan. Experimental results show that the proposed method successfully detects vehicles and estimate the motion vectors of moving vehicles at an intersection. Turn ratio is measured with a satisfactory accuracy.

To increase the performance of moving vehicle segmentation, we propose a shadow-region-based statistical nonparametric method to construct a ratio model for shadow pixel detection. The generation of this shadow model is more effective than existing methods.

Two types of spatial analysis have been further designed to enhance the performance of shadow suppression. Practical experimental results show that the shadow detection method outperforms existing methods both in the shadow suppression rate and the computation time.

The proposed scheme has been successfully applied to an ITMS. Traffic parameter estimation with shadow suppression works satisfactorily. Traffic parameters such as traffic flow, turn ratios, traffic densities, and vehicle speeds have been obtained with acceptable accuracy. The absolute error rate is less than 5.3% for vehicle speed estimation and within 3.7% for the turn ratio estimation.

7.2 Future Directions

Some directions for future study are recommended below:

(1) In the future, practical applications of PTZ cameras will be further studied. For instance, most image-based traffic surveillance methods adopt a virtual window to detect vehicles [71]. If the view of a PTZ camera is changed, then the position and size of the window must be adjusted manually. Using the dynamic calibration procedure developed in this thesis, the detection window can be arranged automatically. On the other hand, the effects of lens distortion and having a non-fixed principal point need to be handled in order to increase the accuracy of PTZ camera calibration.

(2) Several directions on vehicle detection and tracking deserve further study in the future. On one hand, heavy occlusion of vehicles influences the accuracy of image measurement. Methods need to be developed to distinguish individual vehicles. Color information of individual tracked cars can be very useful for solving this problem [80]. On another hand, in order to increase the accuracy of foreground segmentation, we will focus on the study of new methods to select adaptive thresholds for handling the change of environment

illumination on the road. To achieve a dependable performance, a Neuro-Fuzzy classifier can be desirable for an ITMS.

(3) For future shadow-detection studies, more emphasis will be directed to increase the robustness of the shadow detection under various illumination conditions. First, the Gaussian ratio model built under a specific illumination condition might fail under a considerably different illumination. In traffic monitoring applications, it will be beneficial to build a database of ratio models for different illumination conditions. Additionally, shadow pixels that lie nearmoving vehicles or overlap other vehicles might be misclassified as moving-vehicle pixels. The pixels of the same shadow region may have similar color information to the traffic imagery. The color distribution can be used to find uniform sub-regions and hence the uniform sub-regions can be used to verify the actual shadow region [81].

Appendix A

Derivation of Focal Length Equation

In this appendix, the focal length equation will be derived by using only two parallel lines in the image. As shown in Fig. 2-1(a), lines L_1 and L_2 are two parallel lines, L_1 and L_2 intersect X-axis and Y-axis at P_1, P_2, P_3 and P_4; the corresponding coordinates of these points in image plane are expressed as follows:

$$u_2 = \frac{fX_2}{F} = -\frac{fY_1 \tan\theta}{h\csc\phi} = -\frac{f\alpha Y_3 \tan\theta}{h\csc\phi}, \tag{A.1}$$

$$u_4 = -\frac{fX_4}{F} = -\frac{fY_3 \tan\theta}{h\csc\phi}, \tag{A.2}$$

$$v_1 = \frac{fY_1 \sin\phi}{Y_1 \cos\phi + h\csc\phi}, \tag{A.3}$$

$$v_3 = \frac{fY_3 \sin\phi}{Y_3 \cos\phi + h\csc\phi}, \tag{A.4}$$

where X_2 is the X coordinate of P_2, X_4 is the X coordinate of P_4, Y_1 is the Y coordinate of P_1 and Y_3 is the Y coordinate of P_3. Dividing (A1) by (A.2), we obtain $\alpha = \frac{Y_1}{Y_3} = \frac{u_2}{u_4}$. Writing r as $fY_3 \sin\phi$, s as $Y_3 \cos\phi$ and t as $h\csc\phi$, (A.3) and (A.4) can be rewritten as

$$v_1 = \frac{\alpha r}{\alpha s + t}, \tag{A.5}$$

$$v_3 = \frac{r}{s + t}. \tag{A.6}$$

Applying trigonometric function properties, one can easily find the relationship between r, s and t. Computing $r^2 + f^2 s^2$ and $r^2 t^2$, one can obtain

$$r^2 + f^2 s^2 = f^2 Y_3^2, \tag{A.7}$$

$$r^2 t^2 = (f Y_3 h)^2. \tag{A.8}$$

Dividing (A.7) by (A.8), one has

$$\frac{r^2 + f^2 s^2}{r^2 t^2} = \frac{1}{h^2}. \tag{A.9}$$

Solving for r, s in terms of t from (A5) and (A6), one has

$$r = \frac{\alpha v_1 v_3 - v_1 v_3}{\alpha v_1 - \alpha v_3} t = \beta_1 t, \tag{A.10}$$

$$s = \frac{\alpha v_3 - v_1}{\alpha v_1 - \alpha v_3} t = \beta_2 t, \tag{A.11}$$

Substituting (A.10) and (A.11) into (A.9) and using $t = h\csc\phi$, (A.9) can be rewritten as

$$\frac{\beta_1^2 + f^2 \beta_2^2}{\beta_1^2 (h\csc\phi)^2} = \frac{1}{h^2}. \tag{A.12}$$

Rearranging (A.12), we obtain

$$\beta_1^2 + f^2 \beta_2^2 = \beta_1^2 \csc^2 \phi. \tag{A.13}$$

Next, the relationship between $\csc\phi$ and $\sec\theta$ is derived. In Fig. 2-1, the world coordinates of the camera is $(0, -h\cot\phi, h)$ and the X-coordinates of P_2 and P_4 are expressed below:

$$X_4 = h\cot\phi\tan\theta - d\sec\theta, \tag{A.14}$$

$$X_2 = X_4 - w\sec\theta. \tag{A.15}$$

The u coordinates of P_2 and P_4 in an image frame are expressed as follows:

$$u_2 = \frac{fX_2}{F} = \frac{f(X_4 - w\sec\theta)}{F}, \tag{A.16}$$

$$u_4 = \frac{fX_4}{F}, \tag{A.17}$$

Let $X_4 = qw\sec\theta$; then, $X_2 = (q-1)w\sec\theta$, (A.16) and (A.17) can be rewritten as

$$u_2 = \frac{f(q-1)w\sec\theta}{F}, \tag{A.18}$$

$$u_4 = \frac{fqw\sec\theta}{F}. \tag{A.19}$$

From (A18) and (A19), q can be expressed in terms of u_1 and u_2:

$$q = \frac{1}{1-\frac{u_2}{u_4}} = \frac{1}{1-\alpha}. \tag{A.20}$$

Substituting $F = h\csc\phi$ into (A.19), we obtain the relationship between $\csc\phi$ and $\sec\theta$:

$$\csc\phi = \frac{qw}{hu_4} f\sec\theta. \tag{A.21}$$

Substituting (A21) into (A13), we obtain

$$\beta_1^2 + f^2\beta_2^2 = \beta_1^2\left(\frac{qw}{hu_4}\right)^2 f^2\sec^2\theta. \tag{A.22}$$

Using the vanishing point constraints and trigonometric function properties, one can proceed to derive equations that will determine the equation containing $\sec^2\theta$ and, finally,

the focal length equation. Squaring (2.5) and using $\sec^2\phi = 1 + \tan^2\phi$, we have

$$u_0^{\,2} = f^2 \tan^2\theta + f^2 \tan^2\theta \tan^2\phi . \tag{A.23}$$

Since $v_0 = f\tan\phi$, (A.23) can be rewritten as follows:

$$u_0^{\,2} = (f^2 + v_0^{\,2})\tan^2\theta . \tag{A.24}$$

Using $\tan^2\theta = (\sec^2\theta - 1)$, (A.24) yields

$$u_0^{\,2} + f^2 + v_0^{\,2} = (f^2 + v_0^{\,2})\sec^2\theta . \tag{A.25}$$

The equation containing $\sec^2\theta$ is found. Dividing (A.25) by (A.22), we get

$$\frac{u_0^{\,2} + f^2 + v_0^{\,2}}{\beta_1^2 + f^2\beta_2^2} = \frac{f^2 + v_0^{\,2}}{\beta_1^2 (\frac{qw}{hu_4})^2 f^2} . \tag{A.26}$$

Rearranging (A.26), we arrive at:

$$am^2 + bm + c = 0 . \tag{A.27}$$

where m is f^2 and other variables are listed Table 2-1. This governing equation is presented in Section 2.2 as the focal length equation.

Next, let us discuss how camera parameters affect the sign of coefficient a. For simplicity, the sign of at^2 is discussed instead of the sign of a

$$at^2 = Bt^2 - \beta_2^{\,2} t^2 = Y_3^{\,2}(\cos^2\theta - \cos^2\phi) . \tag{A.28}$$

(A.28) reveals that the magnitude of the tile angle and pan angle affect the sign of coefficient a; the details are listed below:

- $a > 0$, if $|\phi| > |\theta|$,

- $a = 0$, if $|\phi| = |\theta|$ or $Y_3 = 0$,
- $a < 0$, if $|\phi| < |\theta|$.

It is clear that the difference between the absolute value of tilt angle and pan angle determines the sign of coefficient a. When $a = 0$, the focal length equation becomes linear and the focal length can be easily estimated. This completes the derivation of the focal length equation.

When the vanishing point is far from the image center or disappears (for instance, when the tilt angle equals 90 deg) in the image frame, (A.27) cannot be used to find the focal length. Instead the focal length can be easily obtained by the perspective projection equation:

$$f = h\frac{w_p}{w}, \tag{A29}$$

where w_p is the width between parallel lanes in the image frame.

Appendix B

Conversion between Pixel Coordinates and World Coordinates

In this appendix, we derive the transformation between pixel coordinates and world coordinates. We will explain how focal length and tilt angle are used to obtain the world coordinates of a feature in the ground plane.

A pixel coordinate (u,v) is expressed as a function of world coordinate (X,Y,Z)

$$u = -f\frac{X_c}{Z_c} = -f\frac{X}{-Y\cos\phi - F}, \tag{B.1}$$

$$v = -f\frac{Y_c}{Z_c} = -f\frac{Y\sin\phi}{-Y\cos\phi - F}. \tag{B.2}$$

(B.2) is rewritten as

$$vY\cos\phi + vh\csc\phi = fY\sin\phi. \tag{B.3}$$

Dividing (B.3) by $\cos\phi$, one can have

$$vY + \frac{vh\csc\phi}{\cos\phi} = fY\frac{\sin\phi}{\cos\phi}. \tag{B.4}$$

Using $v_0 = f\tan\phi$, (B.4) is rewritten as

$$vY + \frac{vh}{\sin\phi\cos\phi} = Yv_0. \tag{B.5}$$

The solution of (B.5) is

$$Y = \frac{h}{f}\frac{v}{\sin^2\phi}\frac{v_0}{v_0 - v}. \tag{B.6}$$

Substituting (B.6) into (B.1), it is easy to obtain

$$X = \frac{h}{f}\frac{u}{\sin\phi}\left(\frac{v}{v_0 - v} + 1\right) = \frac{h}{f}\frac{u}{\sin\phi}\frac{v_0}{v_0 - v}. \tag{B.7}$$

From (B.6) and (B.7), one can transform the pixel coordinates into their world coordinates.

Appendix C

RGB Color Ratio Model of Shadow Pixels

In an outdoor daytime environment, there are two light sources, namely, a light point source and a diffused extended light source. In the following derivation, the road is assumed as to be Lambertian, with a constant reflectance in a traffic scene. The radiance L_{lit} of the light reflected at a given point on a surface in the scene is formulated as [82]

$$L_{lit}(\lambda,i,e,g) = L_s(\lambda,i,e,g) + L_b(\lambda,i,e,g) + L_a(\lambda), \quad \text{(C.1)}$$

where $L_s(\lambda,i,e,g)$, $L_b(\lambda,i,e,g)$, $L_a(\lambda)$ are the surface, body and ambient reflection terms, respectively; λ is the wavelength; i is the angle of incidence between the illumination direction and the surface normal at a considered point; e is the reflection angle between the surface normal and the viewing direction; and g is the phase angle illumination direction and the viewing direction. When sunlight occlusion creates shadows, the radiance L_{shadow} of the reflected light becomes

$$L_{shadow}(\lambda,i,e,g) = L'_a(\lambda), \quad \text{(C.2)}$$

where $L'_a(\lambda)$ is the ambient reflection term in presence of the occluding object. To simplify the analysis, the design assumes that the ambient light coming from the sky is not influenced by the presence of the occluding objects, that is, $L'_a(\lambda) = L_a(\lambda)$.

The model is derived based on an RGB mode. The color components of the reflected intensity reaching the RGB sensors at a point (x,y) in the 2D image plane can be expressed

as

$$C_i = \int_{\Lambda} E(\lambda, x, y) S_{C_i}(\lambda) d\lambda , \tag{C.3}$$

where $C_i \in \{R, G, B\}$ are the sensors responses, $E(\lambda, x, y)$ is the image irradiance at (x, y) point and $S_{C_i}(\lambda)$ is the spectral sensitivities of the RGB sensors of a color camera, respectively, and Λ is determined by $S_{C_i}(\lambda)$, which is non-zero over a bounded interval of wavelengths λ. We assume that the scene radiance and the image irradiance are the same because the situation is considered a Lambertian scene under uniform illumination [83]. For a point in direct light, the sensor measurements are

$$C_i(x, y)_{lit} = \int_{\Lambda} [L_s(\lambda, i, e, g) + L_b(\lambda, i, e, g) + L_a(\lambda)] S_{C_i}(\lambda) d\lambda , \tag{C.4}$$

When a point is in the shadow, the sensor measurements are

$$C_i(x, y)_{shadow} = \int_{\Lambda} L_a(\lambda) S_{C_i}(\lambda) d\lambda . \tag{C.5}$$

Since the reflection terms $L_s(\lambda, i, e, g)$, $L_b(\lambda, i, e, g)$ and $L_a(\lambda)$ of a road surface are similar in a traffic scene, for each object point of the road surface, the RGB measurement ratio between the lit and the shadow condition is approximately constant:

$$\frac{C_i(x, y)_{shadow}}{C_i(x, y)_{lit}} = const. , \tag{C.6}$$

that is, $R_{shadow} = \alpha R_{lit}$, $G_{shadow} = \beta G_{lit}$ and $B_{shadow} = \gamma B_{lit}$ where α, β, γ is less than 1.

Bibliography

[1] B. McQueen and J. McQueen, *Intelligent Transportation Systems Architectures*. Norwood, MA: Artech House, 1999, pp. 19-49.

[2] K. Hayashi and M. Sugimoto, "Signal control system (MODERATO) in JAPAN," in *Proc. IEEE Int. Conf. Intell. Transport. Syst.,* Tokyo, Japan, 1999, pp. 988 – 992.

[3] G. K. H. Pang, K. Takabashi, T. Yokota and H. Takenaga, "Adaptive route selection for dynamic route guidance system based on fuzzy-neural approaches," *IEEE Trans. Veh. Technol.*, vol. 48, no. 6, pp. 2028 – 2041, Nov. 1999.

[4] V. Kastrinaki, M. Zervakis and K. Kalaitzakis, "A survey of video processing techniques for traffic applications," *Image and Vision Computing,* vol. 21, no. 4, pp. 359-381, Dec. 2003.

[5] R. Cucchiara, M. Piccardi and Paola Mello, "Image analysis and rule-based reasoning for a traffic monitoring system," *IEEE Trans. Intell. Transport. Syst.,* vol. 1, no. 2, pp. 119-130, Jun. 2000.

[6] N. J. Ferrier, S. M. Rowe and A. Blake, "Real-time traffic monitoring," in *Proc. Second IEEE Workshop on Applications of Computer Vision*, Sarasota, Florida, 1994, pp. 81–88.

[7] D. Koller, K. Daniilidis and H. H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," *International Journal of Computer Vision*, vol. 10, no.3, pp. 257-281, Jun. 1993.

[8] H. Veeraraghavan, O. Masoud and N. Papanikolopoulos, "Computer vision algorithms for intersection monitoring," *IEEE Trans. Intell. Transport. Syst.,* vol. 4, no. 2, pp.78-89, Jun. 2003.

[9] K. T. Song and J. C. Tai, "Dynamic calibration of pan-tilt-zoom cameras" *IEEE Trans. Syst., Man, Cybern. B*, vol. 36, no. 5, in press.

[10] D. Beymer, P.F. McLauchlan, B. Coifman and J. Malik, "A real-time computer vision system for measuring traffic parameters," in *Proc. IEEE Comput. Vis. and Pattern Recogn.*, San Juan, Puerto Rico, 1997, pp. 495-501.

[11] O. Masoud, N. P. Papanikolopoulos and E. Kwon, "The use of computer vision in monitoring weaving sections," *IEEE Trans. Intell. Transport. Syst.,* vol. 2, no. 1, pp. 18–25, Mar. 2001.

[12] J. C. Tai, S. T. Tseng, C. P. Lin and K. T. Song, "Real-time image tracking for automatic traffic monitoring and enforcement applications," *Image and Vision Computing Journal,* vol. 22, no. 6, pp. 485–501, Jun. 2004.

[13] Z. Zhu, G. Xu, B. Yang, D. Shi and X. Lin, "VISATRAM: a real-time vision system for automatic traffic monitoring," *Image and Vision Computing Journal,* vol. 18, no. 10. pp. 485–501, July 2000.

[14] A. M. Sabatini, V. Genovese and E. S. Maini, "Toward low-cost vision-based 2D localisation systems for applications in rehabilitation robotics," in *Proc. IEEE Int. Conf. Intell. Robots and Syst.*, Lausanne, Switzerland, 2002, pp. 1355–1360.

[15] F. Y. Wang, "A simple and analytical procedure for calibrating extrinsic camera parameters," *IEEE Trans. Robot. Automat.*, vol. 20, no. 1, pp. 121–124, Feb. 2004.

[16] S. Ying and G. W. Boon "Camera self-calibration from video sequences with changing focal length," in *Proc. IEEE Int. Conf. Image Processing,* Chicago, Illinois, 1998, vol. 2, pp. 176–180.

[17] E. Izquierdo, "Efficient and accurate image based camera registration," *IEEE Trans. Multimedia,* vol. 5, no. 3, pp. 293–302, Sept. 2003.

[18] L. L. Wang and W. H. Tsai, "Camera calibration by vanishing lines for 3-D computer vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 4, pp. 370–376, Apr. 1991.

[19] T. Echigo, "A camera calibration technique using three sets of parallel lines," *Machine Vision and Applications,* vol. 3, no. 3, pp. 159–167, 1990.

[20] E. K. Bas and J. D. Crisman, "An easy to install camera calibration for traffic monitoring," in *Proc. IEEE Conf. Intell. Transport. Syst.,* Boston, Massachusetts, 1997, pp. 362–366.

[21] C. Zhaoxue and S. Pengfei, "Efficient method for camera calibration in traffic scenes," *IEE Electron. Lett.*, vol. 40, no. 6, pp. 368–369, Mar. 2004.

[22] A. H. S. Lai and N. H. C. Yung, "Lane detection by orientation and length discrimination," *IEEE Trans. Syst., Man, Cybern. Part B*, vol. 30, no. 4, pp. 539–548, Aug. 2000.

[23] T. N. Schoepflin and D. J. Dailey, "Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation," *IEEE Trans. Intell. Transport. Syst.,* vol. 4, no. 2, pp. 90–98, Jun. 2003.

[24] N. Li, J. Bu and C. Chen, "Real-time video object segmentation using HSV space," in *Proc. IEEE Conference on Image Processing*, Rochester, New York, 2002, pp. 85-88.

[25] R. Cucchiara, M. Piccardi and P. Mello, "Image analysis and rule-based reasoning for a traffic monitoring system," *IEEE Trans. on Intelligent Transportation Systems,* vol. 1, no. 2, pp.119-130, 2000.

[26] C. Eveland, K. Konolige, and R. Bolles, "Background modeling for segmentation of video-rate stereo sequences," in *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Santa Barbara, 1998, pp. 266-271.

[27] J. B. Zheng, D. D. Feng, W. C. Siu, Y. N. Zhang, X. Y. Wang and R. C. Zhao, "The accurate extraction and tracking of moving objects for video surveillance," in *Proc. International Conference on Machine Learning and Cybernetics,* Beijing, China, 2002, pp. 1909-1913.

[28] P. Kumar, K. Sengupta and A. Lee, "A comparative study of different color spaces

for foreground and shadow detection for traffic monitoring system," in *Proc. IEEE 5th International Conference on Intelligent Transportation Systems,* Singapore, 2002, pp. 100-105.

[29] C. Stauffer and W.E.L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 747-757, 2000.

[30] D. Butler, S. Sridharan and V.M. Bove, "Real-time adaptive background segmentation," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing,* Hong Kong, 2003, pp. 349-352.

[31] R. A. Johnson and G. K. Bhatacharyya, *Statistics: principles and methods*, John Wiley & Sons, New York, 2001.

[32] P. Kumar, S. Ranganath, W. Huang and K. Sengupta, "Framework for real-time behavior interpretation from traffic video," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 1, pp. 43-53, 2005.

[33] J. W. Hsieh, S. H. Yu, Y. S. Chen and W. F. Hu, "A shadow elimination method for vehicle analysis," in *Proc. IEEE Int. Conf. on Pattern Recognition* Cambridge, UK, 2004, pp. 372–375.

[34] A. Yoneyama, C. H. Yeh and C. C. J. Kuo, "Moving cast shadow elimination for robust vehicle extraction based on 2D joint vehicle/shadow models," in *Proc. IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*, Miami, Florida, 2003, pp. 21–22.

[35] S. Nadimi and B. Bhanu, "Physical models for moving shadow and object detection in video," *IEEE Trans. Pattern Anal. Mach. Intel.,* vol. 26, no. 8, pp. 1079-1087, 2004.

[36] A. Prati, I. Mikic, M. M. Trivedi and R. Cucchiara, "Detecting moving shadows: algorithms and evaluation," *IEEE Trans. Pattern Anal. Mach. Intell*., vol. 25, no. 7, pp. 918-923, Jul. 2003.

[37] T. Thongkamwitoon, S. Aramvith and T. H. Chalidabhongse, "An adaptive real-time background subtraction and moving shadows detection," in *Proc. IEEE Int. Conf. on Multimedia and Expo*, Taipei, 2004, pp. 1459–1462.

[38] R. Cucchiara, C. Grana, M. Piccardi and A. Prati, "Detecting moving objects, ghosts and shadows in video streams", *Trans. Pattern Anal. Mach. Intel.,* vol. 25, no. 10, pp. 1337-1342, Oct. 2003.

[39] E. Salvador, A. Cavallaro and T. Ebrahimi, "Cast shadow segmentation using invariant colour features," *Computer Vision and Image Understanding,* vol. 95, no. 2, pp. 238–259, 2004.

[40] A. Bevilacqua, "Effective shadow detection in traffic monitoring applications," *Journal of WSCG*, vol. 11, no. 1, pp. 57–64, 2003.

[41] Y. Sato and K. Ikeuchi, "Reflectance analysis under solar illumination," in *Proc. IEEE Workshop on Physics-Based Modeling and Computer Vision '95*, Cambridge, Massachusetts, 1995, pp. 180-187.

[42] E. E. Hilbert, C. Carl, W. Gross, G. R. Hanson, M. J. Olasaby and A. R. Johnson, "Wide area detection system - conceptual design study," *Report No. FHWA-RD-77-86, Federal Highway Administration*, Washington, D.C., USA, 1978.

[43] D. Koller, K. Daniilidis and H. H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," *Int. J. Comput. Vis.* vol. 10, pp. 257–281, 1993.

[44] A. E. C. Pece and A. D. Worrall, "Tracking with the EM contour algorithm," in *Proc. Eur. Conf. Computer Vision*, Copenhagen, 2002, pp. 28–31.

[45] D. W. Lim, S. H. Choi and J. S. Jun, "Automated detection of all kinds of violations at street intersection using real time individual vehicle tracking," in *IEEE Southwest Symp. Image Anal.Interpretation*, Santa Fe, New Mexico, 2002, pp. 126–129.

[46] S. Kamijo, Y. Matsushita, K. Ikeuchi and M. Sakauchi, "Traffic monitoring and accident detection at intersections," *IEEE Trans. Intell. Transport. Syst.* vol. 1, pp. 108–118, 2000.

[47] W. L. Hsu, H. Y. M. Liao, B. S. Jeng and K. C. Fan, "Real-time traffic parameter extraction using entropy," *IEE Proc. Vis. Image Signal Process,* vol. 151, no. 3, pp. 194–202, 2004.

[48] A. H. S. Lai and N. H. C. Yung, "Vehicle-type identification through automated virtual loop assignment and block-based direction-biased motion estimation," *IEEE Trans. Intell. Transport. Syst.,* vol. 1, no. 2, pp. 86-97, Jun. 2000.

[49] S. T. Tseng and K. T. Song, "Real-time image tracking for traffic monitoring," in *Proc. IEEE Int. Conf. Intell. Transport. Syst.*, Singapore, 2002, pp. 1–6.

[50] M. V. D. Berg, B. D. Schutter, A. Hegyi and J. Hellendoorn, "Model predictive control for mixed urban and freeway networks," in *Proc. 83rd Annual Meeting Transport. Research Board*, Washington, D.C., 2004, pp. 1-19.

[51] H. C. Liu and M. Kuwahara, "A study on real-time signal control for an oversaturated network," in *Proc. 7th World Congress Intell. Transport. Syst.*, Torino, 2000.

[52] R. Camus, G. Fenu, G. Longo, F. Pampanin and T. Parisini, "Identification of freeway-traffic dynamic models: a real case study," in *Proc. IEEE American Control Conf.*, Denver, 2003, pp. 4579–4584.

[53] V. D. Zijpp, "Dynamic origin-destination matrix estimation on motorway networks," Ph. D. dissertation, Dept. Transport. Planning Traffic Eng., Delft Univ. Technology, GA Delft, Netherlands, 1996.

[54] Y. Asakura, "Origin-destination matrices estimation model using automatic vehicle identification data and its application to the Han-Shin expressway network," *Transport. Research*, vol. 27, no. 4, pp. 419–438, Jan 2000.

[55] C. Oh, S. G. Ritchie, J, Oh and R. Jayakrishnan, “Real-time origin-destination (OD) estimation via anonymous vehicle tracking,” in *Proc. IEEE Int. Conf. Trans. Intell. Transport. Syst.*, Singapore, 2002, pp. 582–586.

[56] A. G. Hobeika and C. K. Kim, “Traffic flow prediction systems based on upstream traffic,” in *Proc. IEEE-IEE Vehicle Navigation and Information Syst. Conf.*, Yokohama, Japan, 1994, pp. 345 -350.

[57] S. Chen, Z. P. Sun and B. Bridge, “Automatic traffic monitoring by intelligent sound detection,” in *Proc. IEEE Conf. Intell. Transport. Syst.*, Boston, MA, 1997, pp. 171–176.

[58] S. S. Beauchemin and J. L. Barron, “The computation of optical flow,” *ACM Computing Surveys*, vol. 27, no. 3, pp. 433–467, Sep. 1995.

[59] B. K. Horn and B. G. Schunck, “Determining optical flow,” *Artificial Intelligence,* vol. 17, no. 1, pp. 185–203, Aug. 1981.

[60] D. J. Fleet and K. Langley, “Recursive filters for optical flow,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 1, pp. 61–67, Jan. 1995.

[61] G. Tziritas, “Recursive and/or iterative estimation of the two-dimensional velocity field and reconstruction of three-dimensional motion,” *Signal Processing*, vol. 16, no. 1, pp. 53–72, 1989.

[62] M. Elad and A. Feuer, “Recursive optical flow estimation - adaptive filtering approach,” *J. Visual Comm. and Image Representation*, vol. 9, no. 2, pp. 119–138, Jun. 1998.

[63] Y. U. Yim and S. Y. Oh “Three-feature based automatic lane detection algorithm (TFALDA) for autonomous driving,” *IEEE Trans. Intell. Transport. Syst.,* vol. 4, no. 4, pp. 219–225, Dec. 2003.

[64] R. C. Jain, R. Kasturi and B.G. Schunck, *Machine Vision*. McGraw-Hill, New York, 1995.

[65] S. T. Bow, *Pattern Recognition and Image Preprocessing*. New York: Marcel Dekker, 2002.

[66] A. C. Bovik, S. J. Aggarwal, F. Merchant, N. H. Kim and K. R. Diller, “Automatic area and volumn measurement from digital biomedical images,” In Image analysis methods and applications: methods and applications, D. P. Hader (Editor), CRC Press, Boca Raton, FL., 2001, pp. 23-64.

[67] J. T. McClave, Terry Sincich and William Mendenhall, *Statistics (8th Edition)*, Prentice Hall, New Jersy, 1999.

[68] L. G. Shapiro and G. C. Stockman, *Computer vision*, Prentice Hall, New Jersy, 2001.

[69] See http://www.itstaiwan.org/Home_English.asp, last visited July 14, 2006.

[70] See http://www.cmlab.csie.ntu.edu.tw/cml/dsp/training/coding/h263/h263.html, last visited July 14, 2006.

[71] J. C. Tai and K.T. Song, "Automatic contour initialization for image tracking of multi-lane vehicles and motorcycles," in *Proc. IEEE Conf. Intell. Transport. Syst.*, Shanghai, China, 2003, pp. 808–813.

[72] I. Haritaoglu, D. Harwood and L. S. Davis, "W4: Real-time surveillance of people and their activities," *Trans. Pattern Anal. Mach. Intel.,* vol. 22, pp. 809-830, Aug. 2000.

[73] M. Baumberg and D.C. Hogg, "An efficient method for contour tracking using active shape models," in *IEEE Motion Non-rigid Articulated Objects Workshop*, Austin, Texas, 1994, pp. 194–199.

[74] A. Koschan, S. K. Kang, J. K. Paik, B. R. Abidi and M.A. Abidi, "Video object tracking based on extended active shape models with color information," in *Proc. Eur. Conf .Color Graphics Imaging Vision*, Poitiers, France, 2002, pp. 126–131.

[75] G. Iannizzotto and L. Vita, "Real-time object tracking with movels and affine transformations," in *Int. Conf. Image Processing*, 2000, pp. 316–318.

[76] A. Blake and M. Isard, *Active contours*, Springer Press, London, England, 1998.

[77] S. M. Bozic, *Digital and Kalman filtering*, Edward Arnold, London, England, 1994.

[78] C. J. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vision Conf.,* Manchester, 1988, pp. 147–151.

[79] A. Singh, *Optic Flow Computation: A Unified Perspective*. Los Alamitos, CA: IEEE Computer Society Press, 1991, pp. 33–36.

[80] W. Hu, X. Xiao, D. Xie, T. Tan and S. J. Maybank, "Traffic accident prediction using 3-D model-based vehicle tracking," *IEEE Trans. Veh. Technol.* vol. 53, no. 3, pp. 677–694, 2004.

[81] D. Comaniciu and P. Meer. "Mean shift: A robust approach toward feature space analysis," *Trans. Pattern Anal. Mach. Intel.,* vol. 24, no. 5, pp. 603–619, 2002.

[82] E. Salvador, "Shadow segmentation and tracking in real-world conditions," Ph.D. Thesis. Signal Processing Institute, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2004.

[83] P. Favaro and S. Soatto, "A variational approach to scene reconstruction and image segmentation from motion blur cues," in *Proc. IEEE Intl. Conf. Comp. Vis. and Patt. Recog.,* Washington, DC, USA, 2004, pp. 631-637.

Vita

姓名: 戴任詔

性別: 男

生日: 中華民國 51 年 4 月 22 日

籍貫: 台灣省台中縣

論文題目: 中文: 交通參數估測系統之攝影機參數校正與影像追蹤

英文: Camera Calibration and Image Tracking for Traffic Parameter Estimation

學歷:

1. 民國 74 年 6 月　　國立清華大學動力機械工程學系畢業
2. 民國 81 年 6 月　　國立交通大學控制工程研究所畢業
3. 民國 88 年 9 月　　國立交通大學電機及控制工程研究所博士班

經歷:

1. 76 年 6 月～76 年 11 月　　羽田機械 助理工程師
2. 76 年 12 月～79 年 3 月　　三陽工業 二級助理研究員
3. 81 年 9 月～迄今　　明新科技大學 機械工程系講師

Publication List

Journal paper

[1] K. T. Song and J. C. Tai, "Dynamic Calibration of Pan-Tilt-Zoom Cameras," *IEEE Trans. Systems, Man, and Cybernetics, Part B.* vol. 36, no. 5, in press. [4 點].

[2] J. C. Tai, S. T. Tseng, C. P. Lin, and K. T. Song, "Real-Time Image Tracking for Automatic Traffic Monitoring and Enforcement Applications," Image and Vision Computing, vol. 22, no. 6, pp. 485–501, 2004 [1.2 點].

[3] J. C. Tai and K. T. Song, "On the Parametric Approach to the MEMS Mask Design," (in Chinese) *Journal of Technology*, vol. 17, no. 1, pp. 73–82, 2002. [0 點].

[4] K. T. Song and J. C. Tai, "Real-Time Background Estimation of Traffic Imagery Using Group-Based Histogram," revised, *Journal of Information Science and Engineering.*

[5] K. T. Song and J. C. Tai, "Image-Based Traffic Monitoring with Shadow Suppression," revised, *Proceedings of the IEEE*.

[6] K. T. Song and J. C. Tai, "Automatic Contour Initialization and Multi-Vehicle Tracking for Vision-Based Traffic Monitoring," submitted to *International Journal of Imaging Systems and Technology*.

Conference paper

[1] K. T. Song and J. C. Tai, "Image-Based Turn Ratio Measurement at Road Intersection," in *Proc. of The IEEE International Conference on Image Processing*, Genova, Italy, 2005, pp. I-1077–1080.

[2] J. C. Tai and K. T. Song, "Background Segmentation and its Application to Traffic Monitoring Using Modified Histogram," *in Proc. of 2004 IEEE International Conference on Networking, Sensing & Control*, Taipei, Taiwan, 2004, pp. 13–18.

[3] J. C. Tai and K. T. Song, "Automatic contour initialization for image tracking of multi-lane vehicles and motorcycles," in *Proc. of the IEEE 6th International Conference on Intelligent Transportation Systems,* Shanghai, 2003, pp.808–813.

[4] C. P. Lin, J. C. Tai and K. T. Song, "Traffic monitoring based on real-time image tracking," *Proc. of. 2003 IEEE International Conference on Robotics and Automation*, Taipei, 2003, pp.2091–2096.

[5] J. C. Tai and K. T. Song, "Design of a Novel Electrostatic Linear Stepping Micromotor, " in *Proc. of the 6th International Conference on Mechatronics Technology*, Kitakyushu, Japan, 2002, pp.411–416.