國立交通大學

電控工程研究所

博士論文

多重感知整合式輔助安全車載資通訊系統

An Assistant Safety Telematics System with Integrated Multiple Sensors

研究生：沈子貴

指導教授：林進燈　教授

中華民國九十九年十一月

多重感知整合式輔助安全車載資通訊系統

An Assistant Safety Telematics System with Integrated Multiple Sensors

研 究 生：沈子貴　　Student：Tzu-Kuei Shen

指導教授：林進燈 教授　　Advisor：Prof. Chin-Teng Lin

國立交通大學

電控工程研究所

博士論文

A Dissertation

Submitted to Institute of Electrical Control Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Doctor

in

Electrical Control Engineering

November 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年十一月

多重感知整合式輔助安全車載資通訊系統

學生：沈子貴　　　　　　　　　　　　　　指導教授：林進燈 教授

國立交通大學電控工程研究所 博士班

摘要

近年來隨著都市人口和車輛數目不斷的增加，都市的交通問題越來越嚴重，過多的車輛造成了交通壅塞的情況，交通意外事故的發生也更加的頻繁，這些事故造成了人民生命與財產的損失，讓每個人平均花在醫療上的費用大幅的增加，造成了龐大的社會成本與負擔，並降低國家的整體經濟競爭能力。因此，各個先進國家投入了相當多的人力在研究智慧型運輸系統(Intelligent Transportation Systems, ITS)，ITS 的主要目的是利用先進科技於車輛及道路設施上，協助駕駛對車輛之控制，以減少事故，增進行車安全並達到提高用路效率與節能減碳的目標。本論文為一完整多重感知整合式輔助安全車載資通訊系統，其中包含車輛安全輔助系統與路口安全監控兩部份。車輛安全輔助系統利用單顆魚眼鏡頭攝影機對車輛進行周邊障礙物偵測與倒車輔助駕駛。周邊障礙物偵測系統是透過反映射模型將影像轉換至世界座標平面上，再根據此影像維度中具立體資訊的障礙物相對移動特性，進而定位出車輛周遭障礙物的位置與距離。倒車輔助駕駛系統是

以電腦視覺為基礎，自動估測倒車時的相對移動量，於畫面上估測出未來的行進軌跡。路口監控的部份，主要為一事件監控系統，開發一影像嵌入式系統平台用於收集路口相關資訊，並由路側單元接收與彙整，再經由DSRC與車上機進行資訊交流，整合車上與附近路口資訊給駕駛者參考，建立起完整的輔助安全車載資通訊系統。

An Assistant Safety Telematics System with Integrated Multiple Sensors

Student：Tzu-Kuei Shen Advisors：Prof. Chin-Teng Lin

Institute of Electrical Control Engineering
National Chiao Tung University

ABSTRACT

In recent years, the urban population and vehicles increase continuingly. The problems of city traffic are more and more serious, such as too many vehicles cause traffic jams and accidents frequently. These accidents do not only make people lost their life and fortune, but also waste lots of medical resources. Upper disastrous influences make enormous social costs, besides they also debase whole national economic competitiveness. For dealing with these problems, Intelligent Transportation System (ITS) becomes an important policy in each country. The main objective of ITS is to develop high-end technology on the electrical equipment in vehicles and traffic applications. Drivers can reduce the probability of traffic accidents and improve self-driving safety via controlling high-end assist driving technology and then achieve the goals of increasingly efficiency in road freight and energy saving and carbon reduction. This dissertation presents a whole integrated multi-sensors telematics safety system that can be divided into safety vehicle assistant system and intersection video surveillance system two parts. The first part contains obstacle detection system and parking assistant system. The obstacle detection system one is transferred image coordinate into world coordinate by fisheye lens inverse perspective mapping modal (FLIPM) and follows the property of moving obstacle to position candidates' location. The parking assistant system is based on computer vision algorithm via motion vector, and estimates the curve in the path of vehicle. The other segment is intelligent intersection surveillance system. Our concern is to consider a whole intersection events monitor system. It collects the traffic data from local intersection by embedded platform and then arranges these data for road side unit (RSU) to communicate with on board unit (OBU) in vehicles via DSRC protocol to set up an assistant safety telematics system.

Acknowledgement

在新竹交大博士求學生活的這五年裡，由衷的感謝指導教授 林進燈老師給予我學術與待人處事上的許多啟發。在研究的過程中提供了完善的設備與幫助，同時也藉由許多合作機會，使我能將理論與實務結合，進而實現許多創新概念的新應用，並提升我的研究與實作能力。同時，亦感謝亦口試委員 楊谷洋老師、 張志永老師、 林文傑老師、 鄭銘揚老師、 許佳興老師與 蒲鶴章博士在百忙之中抽空蒞臨指導，給予我許多建議使本論文更趨完善。

這五年多的博士生活裡，首要感謝 蒲鶴章博士給予我生活上、產業合作和專業領域的許多建言，以及 范剛維博士於研究過程與程式撰寫時的許多幫助，從你們兩人的身上我學到了許多做研發與看人、事、物的方法與準則。此外，要感謝建霆與琳達同學這五年多來的攜手共進，不論是課業上、研究上或是生活上你們都是最佳的夥伴。另外，還要感謝東霖、勝智與肇廷學弟，實驗室有你們的協助，才可以完成許多的合作專案，讓實驗室能有如今的成果。另外，也要感謝許多的學弟妹的陪伴，在這五年裡大家一同學習、研究與討論的過程中，讓彼此都更進一步，讓我再這些充實的日子裡留下美好的回憶。

我要感謝我的父母、兩位可愛的妹妹與我的岳父岳母，你們的關心與幫忙才能讓我在博士的求學生涯裡心無旁騖，可以專心致力於學業，並順利完成論文。另外，要特別感謝老婆宗綾，感謝妳這麼長時間的相陪與與支持，妳的支持是我突破瓶頸與樂於接受新事物挑戰的原動力。最後，要感謝女兒羿彤，妳的出生讓我更有動力去完成博士學位。最後，僅以此論文獻給所有關心我的人。

Table of Contents

Page

Chinese Abstract i

English Abstract ii

Acknowledgement v

Table of Contents vi

Figures viii

Tables x

1. Introduction 1
   - 1.1 Background and Motivation 1
   - 1.2 Objective and Methods 2
   - 1.3 Organization 2
2. Related Work 4
   - 2.1 Telematics 4
   - 2.2 Inverse Perspective Mapping (IPM) 10
   - 2.3 Obstacle Detection 12
   - 2.4 Object Tracking 14
3. Structure of Safety Assistant Driving Telematics System 16
   - 3.1 System Overview 16
   - 3.2 On Board Vision-Based Detection System 19
     - 3.2.1 Embedded Platform Description 19
     - 3.2.2 Sinffer Equipment 22
   - 3.3 Intelligent Surveillance System 25
     - 3.3.1 Improved Intelligent Surveillance System 25
     - 3.3.2 Vision-based incident detector 26
4. Vision-Based Intelligent Technique 28
   - 4.1 Fisheye lens inverse perspective mapping (FLIPM) 28
     - 4.1.1 The Modified Normal Lens IPM Method 29
     - 4.1.2 Fisheye Lens Inverse Perspective Mapping (FLIPM) 33
   - 4.2 Obstacle Detection with single Camera 37
     - 4.2.1 The Pre-Process 37
     - 4.2.2 Profile image 37
     - 4.2.3 The temporal FLIPM difference image 39
     - 4.2.4 Road Detection 42
     - 4.2.5 Ground Movement Estimation 48
     - 4.2.6 Obstacle Feature Searching algorithm 54
     - 4.2.7 Histogram Post-processing 58
     - 4.2.8 Object Tracking and Information Extraction 59
   - 4.3 Dynamic Distance Gauge (DDG) 66
     - 4.3.1 System architecture of the DDG 66
5. Experiment Results 70
   - 5.1 Obstacle Detection Experiment 70
     - 5.1.1 Comparisons about the Normal Lens IPM Method 72
     - 5.1.2 The Experimental Configurations 73
     - 5.1.3 Results in Various Environments 73
     - 5.1.4 Accuracy Evaluation of Obstacle Distance 82
   - 5.2 Dynamic Distance Gauge (DDG) 83

5.3 Entire System Experiment ... 85
5.3.1 Environment description ... 85
5.3.2 Protocol Test Results ... 86
5.3.3 Intersection Test Results ... 88
5.4 Discussions ... 91
6. Conclusions ... 92
Bibliography ... 94

Figures

Figure 1: Harmonized 5.9 GHz DSRC BAND PLAN 5
Figure 2: DSRC Performance envelops 6
Figure 3: The structure of SAE J2735 7
Figure 4: A package processing step........ 7
Figure 5: Structure of Safety Driving Assistant Telematics System........ 16
Figure 6: Structure of Telematics service platforms 17
Figure 7: Current application on telematics system 18
Figure 8: Vehicle Sensors 19
Figure 9: Structure of Intelligent Image Processing Embedded Platform........ 19
Figure 10: Structure of Embedded Platform PCB Design........ 20
Figure 11 : S100 WAVE Box 22
Figure 12: S100 detail Spec........ 22
Figure 13: Sirit Sniffer Card (PCMCIA Interface) 23
Figure 14: OBU Service and Navigation Structure 23
Figure 15: The flowchart of transmitting message 24
Figure 16: Traditional intelligent surveillance system 25
Figure 17: Improved intelligent surveillance system 25
Figure 18: Fisheye lens inverse perspective mapping structure 28
Figure 19 : The vertical line projection of Eq. (4.1)........ 30
Figure 20: The projected result of Eq. (4.5) 30
Figure 21: The figures and expected results 31
Figure 22: The geometrical relations of the image and world coordinate system for deriving our equations. 32
Figure 23 : The original and adjusted scope........ 34
Figure 24: Illustrations for distortion images 35
Figure 25: The flowchart of image pre-processing 37
Figure 26: The results in the profile searching process 38
Figure 27: Illustrations for the temporal FLIPM difference image 39
Figure 28: Results of edge detection and its corresponding optical flow........ 41
Figure 29: Results of corner detection and its corresponding optical flow 42
Figure 30: Block diagram of feature point extraction 42
Figure 31: A color ball i in the L*a*b* color model whose center is at (L_m, *a_m, *b_m) and with radius λ_{max} 44
Figure 32: Sampling area and color ball with a weight which represents the similarity to current road color. 45
Figure 33: Pixel matched with first B weight color balls which are the most represent standard color. 46
Figure 34: Results of road detection........ 47
Figure 35: Results of feature point extraction. The upper image is result of road detection, and lower image is position of feature points........ 48
Figure 36: Flowchart of ground movement estimation 48
Figure 37: Difference between optical flow of original image and those of bird's view image when vehicle is moving straight. 49
Figure 38: Difference between optical flow of original image and those of bird's view image when vehicle is turning. 50
Figure 39: Differences between optical flows of obstacle and those of planar object. 50
Figure 40: Histogram distribution of optical flow in world coordinate system 51
Figure 41: Two-dimensional coordinate plane 52

Figure 42: Procedure of the compensated image building 53
Figure 43: Chart of temporal coherence 54
Figure 44: The results in the feature searching procedure by using profile images 55
Figure 45: The results of the feature searching procedure using temporal difference FLIPM images 56
Figure 46: The flowchart of feature searching 57
Figure 47: The block diagram of histogram post-processing 58
Figure 48 **:** Illustrative figures of the trapezoid histogram distributions. 58
Figure 49: The processes of histogram post-processing. x axis means the polar histogram's angle and y axis means the accumulation on each angle. 59
Figure 50 : The block diagram of object tracking and information extraction.........................59
Figure 51: Flowchart of Obstacle Detection 60
Figure 52: Results of image difference between current image and compensated image........61
Figure 53: Flowchart of obstacle localization 61
Figure 54: the obstacle candidate image and the corresponding vertical-orientated histogram 62
Figure 55: Procedure of creating vertical-orientated histogram 63
Figure 56: Procedure of obstacle verification 64
Figure 57: Transformation between image coordinate and world coordinate 64
Figure 58: Scale measure between world coordinate and real length 65
Figure 59: Block diagram of distance measurement 65
Figure 60: System architecture of the proposed DDG algorithm 66
Figure 61: Allocation of motion detection areas. 67
Figure 62: The scheme of block matching 67
Figure 63: The results of the normal lens IPM equations 72
Figure 64: The set-up location of camera. 73
Figure 65: The results of FLIPM and obstacle detection in different scenes. 76
Figure 66: Results of obstacle tracking in Scenery 1 77
Figure 67: Results of obstacle tracking in Scenery 3 77
Figure 68: Results of obstacle warning in the lateral direction 78
Figure 69: Results of obstacle warning in the rear direction..78
Figure 70: Results of obstacle warning with moving objects..79
Figure 71: The results in different environments with heavy noise. 81
Figure 72: Lane marking for distance measurement 82
Figure 73: Testing results of DDG in various environments..83
Figure 74: Test Street Location..85
Figure 75 Experiment structure for testing transmission rate..86
Figure 76: WSM Captured Message 86
Figure 77: WSM header 87
Figure 78 : The locations of camera on setting intersection..88
Figure 79: The location of fix camera on intersection..88
Figure 80: Simulating a pedestrian crossing intersection..89
Figure 81: Examples of erroneous detected results in our system 91

Tables

Table 1: Content sets of SAE J2735 8
Table 2: VSC-A Basic Safety Message 9
Table 3: Features of BlackFin 561 DSP 20
Table 4: SK34B Electrical Characteristics 21
Table 5: SAE J2735 WSM Message Protocol 23
Table 6: The specifications of our working platform 70
Table 7: The runtime in each processing step 70
Table 8: Comparisons of different obstacle algorithms 71
Table 9: Experimental result of distance measurement 83
Table 10: Parameters of Testing 86
Table 11: Static test (600 packages per test) 87
Table 12: Active test (600 packages per test, 30 Km/hr) 87
Table 13: The minimum distance of stop view and overtake view 89
Table 14 : Table of reaction time via OBU 90

1. Introduction

1.1 Background and Motivation

In recent years, there has been a dramatic proliferation of vehicle and population in the world. This city phenomenon has caused many traffic and environmental protection related problems, such as collision, traffic jam, traffic offence, exhaust emission and so on. In order to solve these problems, many advanced countries have started to develop Intelligent Transportation System (ITS). Up to now, ITS can be divided into automotive electronics technology and traffic control system two research fields. Driver’s fatigue, drowsiness, inattention, and distraction are reported a major causal factor in many traffic accidents. Due to the drivers lost their attention, they had markedly reduced the perception, recognition and vehicle control abilities. Since, related studies had become a major interest research topic in automotive safety engineering. Previously, vehicle detect obstacle via radar sensor, but its response speed is not immediately. Collision happening occurs if the warning signal of radar sensor is presented with a little delay time. Nowadays, mounting a fisheye lens camera on the bumper is a general method to let driver to see reverse direction scene. However, fisheye lens scene has a serious distortion formation of image. This effect makes driver to estimate distance and control the direction of vehicle difficultly. In the other field, microwave, ground loop and radar are now three kinds of popular methods for traffic control applications. All of them do not reconstruct traffic events specifically. On the roads, camera and digital video recorder (DVR) are used to monitor road condition and catch traffic offense. Management costs too much to employ people attending these surveillance systems. Since, a part of researches interesting in vision based traffic control and analysis start to aim the objectives of ITS, such as vehicle detector, event detector, self-adaptive traffic light control and so on. When we can get so much useful traffic related information from these sensors, we may begin integrating all of them to assist drivers keep danger away. Therefore, the concept of telematics is to exchanging information by any kind of detector sensor on the vehicle and road intersection. It a whole system focuses on communicating data, for instance, traffic condition, driver’s spirit state and navigation information etc.. Summarily, combing the advantage of vision-based sensor and telematics technology can develop a whole safety driver assistant system.

1.2 Objective and Methods

This thesis presents a whole safety driving assistant telematics system including obstacle detection with fisheye lens camera on vehicle and road condition surveillance system. We develop a vision-based obstacle detection system by utilizing our proposed fisheye lens inverse perspective mapping (FLIPM) method. The new mapping equations are derived to transform an images captured by a fisheye lens camera into an undistorted remapped ones under practical circumstances. In the obstacle detection, we make use of the features of vertical edges on objects from remapped images to indicate the relative positions of obstacles. In order to obtain a suitable feature, adaptive road recognizing is the first step to extract obvious useless compensation points and mitigate interference by shadow and illumination changing. Our obstacle detection can export a warning signal on the screen within a limited distance from nearby vehicles while the detected obstacles are even with the quasi-vertical edges.

Road condition surveillance system contains intelligent detector and communication module. We improve traditional surveillance system by integrating intelligent detectors and developing road side unit (RSU) and on board unit (OBU) to exchange traffic information. We set up multiple cameras on the intersection to capture video streams from different directions. First, we make use of adaptive Gaussian Mixture Model (GMM) to form basic background. We will track foreground objects' location and predict their future path for estimating collision occurrence. According foreground objects' moving speed and appearance, we classify pedestrian and vehicle to record the behavior of this intersection. We also establish a world coordinate system to map each camera view field. Upper information will be collected by RSU and transfer them to OBU for driver to remind possible danger. Our messages follow WAVE/DSRC SAE J2735 protocol and the effective transmission distance is over 100 meters long. This intelligent system also contains back-end storage equipment and control human interface with DVR and content management system.

1.3 Organization

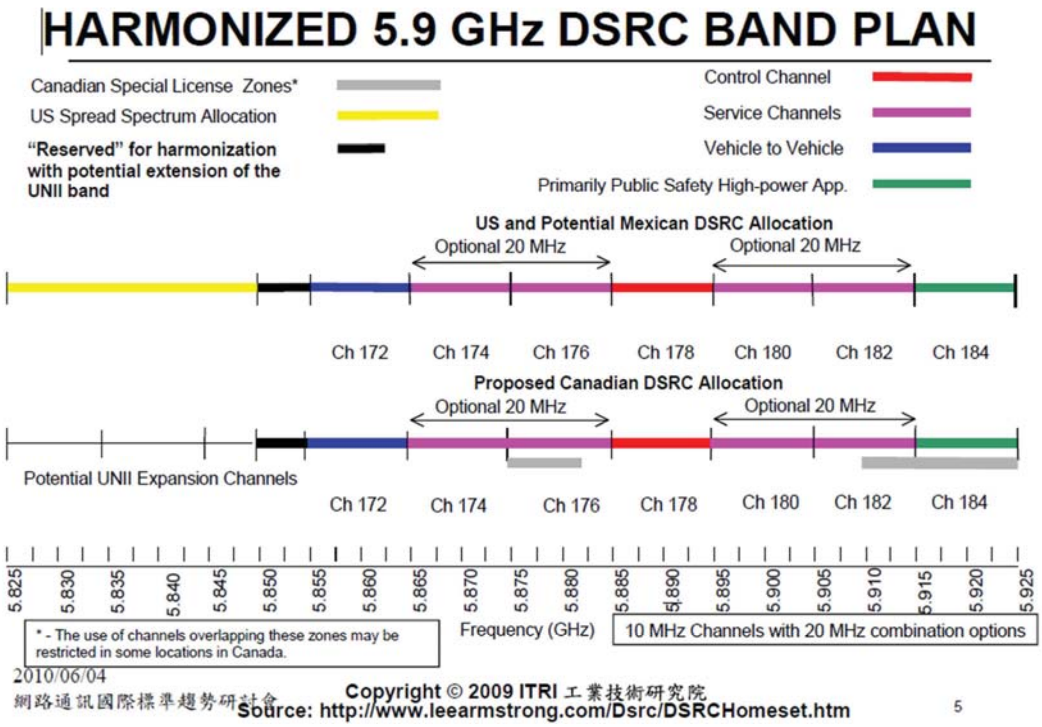
This thesis is organized as follows: Chapter II gives an overview of related work about this research realm and discusses the traditional methods. Chapter III introduces our proposed system divided into three parts. First, we will show a new design of intelligent surveillance system. On the front-end, we present the structure of OBU and show the hardware configuration on a sport utility vehicle. On the back-end, RSU collects and process information captured from multiple intelligent sensors. Chapter IV presents technologies used in each intelligent sensor. The sensors on vehicle contain FLIPM, obstacle detection with single camera and dynamic distance gauge (DDG) three techniques, and on intersection contain collision estimation, moving objects classification and coordinate correction. Chapter V shows experimental results of the implementation of our proposed algorithm and the successful working on actual environment. Finally, we made a conclusion of this study in Chapter VI.

2. Related Work

Our safety assistant driving system has four related researches. The first is telematics fundamental structure including WAVE/DSRC international protocol. Next, we will aim at inverse perspective mapping methods and indicate unsuited spot. Besides, correcting distortion image captured with fisheye lens is also an interesting study. There are two kinds of methods to deal with this problem. Third is the most important subject in safety assistant driving system, and we will confer deeply past methods with different sensors. Finally, we restrict the scope of tracking algorithm without recognition and classifying method. In this chapter, we will review some researches related to these main modules.

2.1 Telematics

Telematics includes two important compositions, one is telecommunications and the other one is informatics. Since Global Positioning System technology is applied to navigate on vehicle driving, the first telematics system would be produced. Nowadays, there are twelve kinds of popular contents for telematics applications, including telematics education, vehicle tracking, trailer tracking, cold store freight logistics, fleet management, satellite navigation, mobile data and mobile television, wireless vehicle safety communications, emergency warning system for vehicles, intelligent vehicle technology, car clubs and auto insurance etc. [1]. In our telematics system, we follow the international standard protocol called Wireless Access in the Vehicular Environment (WAVE)/ Dedicated Short Range Communication (DSRC). It follows IEEE 802.11p and IEEE 1609 international communication standard protocol, and suitable to exchanging information in short distance for safety assistant driving. The communication module is a multi-channels structure, and its licensed band is 5.8 to 5.9GHz. There are seven channels with 10MHz Bandwidth, and the harmonized 5.9 GHz DSRC Band PLAN as shown in Figure 1.


HARMONIZED 5.9 GHz DSRC BAND PLAN
Canadian Special License Zones*
US Spread Spectrum Allocation
"Reserved" for harmonization with potential extension of the UNII band
Control Channel
Service Channels
Vehicle to Vehicle
Primarily Public Safety High-power App.
US and Potential Mexican DSRC Allocation
Optional 20 MHz
Optional 20 MHz
Ch 172
Ch 174
Ch 176
Ch 178
Ch 180
Ch 182
Ch 184
Proposed Canadian DSRC Allocation
Optional 20 MHz
Optional 20 MHz
Potential UNII Expansion Channels
Ch 172
Ch 174
Ch 176
Ch 178
Ch 180
Ch 182
Ch 184
5.825
5.830
5.835
5.840
5.845
5.850
5.855
5.860
5.865
5.870
5.875
5.880
5.885
5.890
5.895
5.900
5.905
5.910
5.915
5.920
5.925
* - The use of channels overlapping these zones may be restricted in some locations in Canada.
Frequency (GHz)
10 MHz Channels with 20 MHz combination options
2010/06/04
網路通訊國際標準趨勢研討會
5

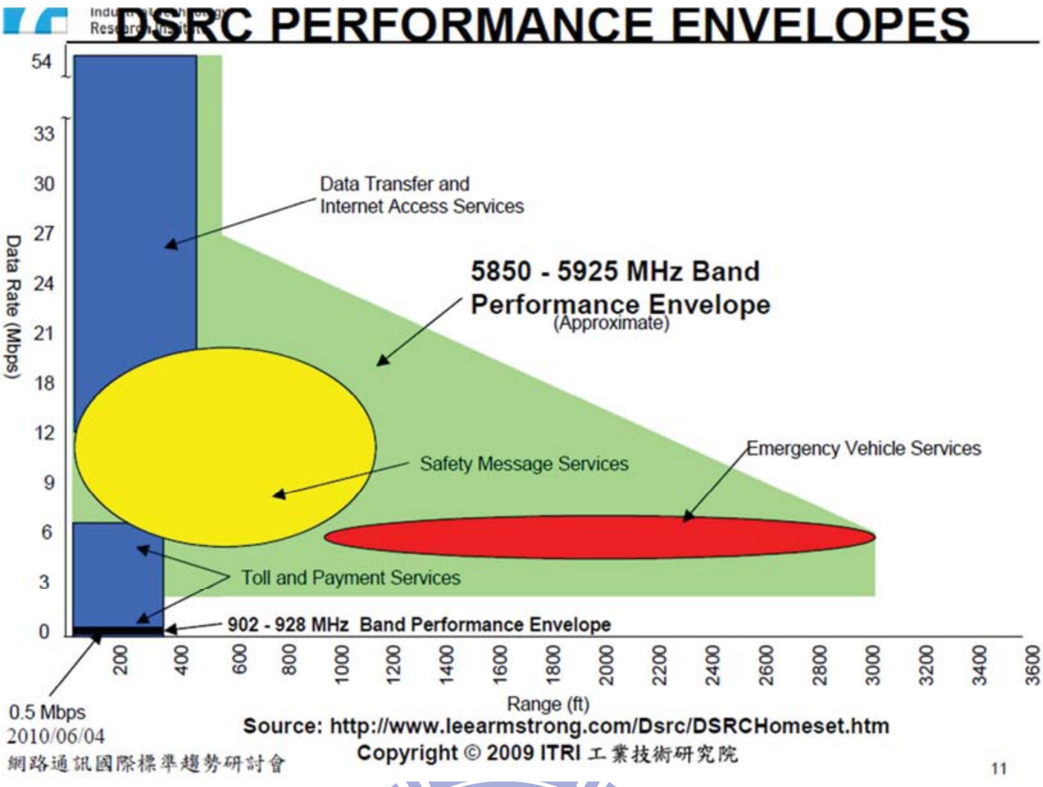
Copyright © 2009 ITRI 工業技術研究院

Source: http://www.leearmstrong.com/Dsrc/DSRCHomeset.htm

Figure 1: Harmonized 5.9 GHz DSRC BAND PLAN

Source: Industrial Technology Research Institute

Standard of IEEE 802.11p is based on CALM [2] and ASTM [3] E2213-03. The former is an European system to provide a standardized set of air interface protocols and parameters for medium and long range, high speed ITS communication using one or more of several media, with multipoint and networking protocols within each media, and upper layer protocols to enable transfer between media. Its communication modes are on V2I, I2I and V2V. The other one follows International standard organization with US playing a major role. E2213-03: Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems - 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications. It purposed to provide wireless communications over short distances between information sources and transactions stations on the roadside and mobile radio units, between mobile units, and between portable units and mobile units. DSRC performance envelopes are shown in Figure 2.


DSRC PERFORMANCE ENVELOPES

Source: http://www.leearmstrong.com/Dsrc/DSRCHomeset.htm

Copyright © 2009 ITRI 工業技術研究院

2010/06/04

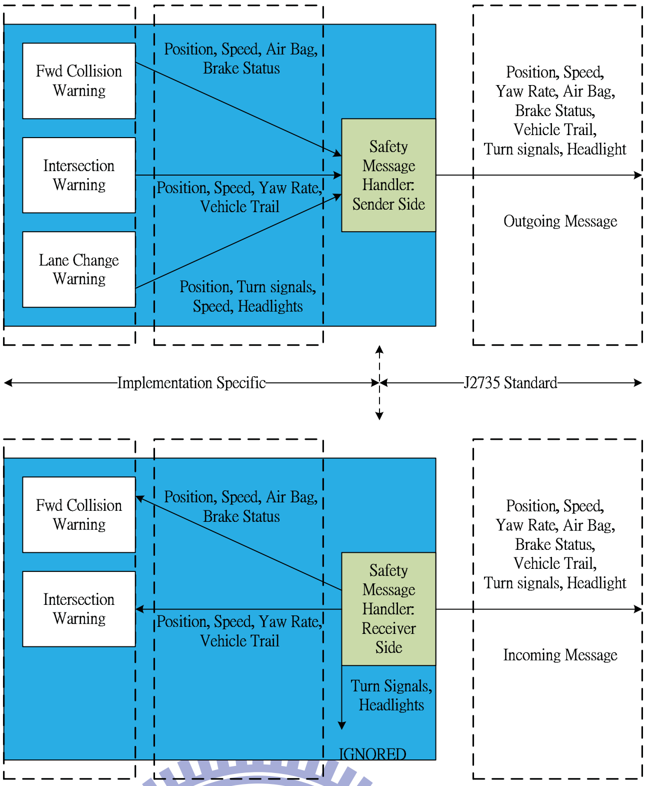
網路通訊國際標準趨勢研討會

Figure 2: DSRC Performance envelops

Source: Industrial Technology Research Institute

IEEE 1609 (Full use) contains five services including 1609.1, 1609.2, 1609.3, 1609.4 and 1609.11. 1609.1 is remote management service and 1609.2 is security services for applications and management messages. 1609.3 is networking services, and 1609.4 is multi-channel operation. Finally, 1609.11 is over-the-air data exchange protocol for ITS.

DSRC SAE J2735 standard protocol defines Message Sets, Data Frames and basic Data elements, and exchange data with ASN.1 (Abstract Syntax Notation one) DER (Distinguish Encoding Rules). It is a popular communication method to encode and decode around the world, and easy to combine with other WAVE/DSRC equipment. Here we only discuss two kinds of related messages protocol to represent SAE J2735. One is Basic Safety Message (BSM), and the other one is Emergency vehicle Alert (EVA). DSRC SAE J2735 is based on Wave Short Message Protocol (WSMP) that contains many related safety driving message sets. In general, the structure of Safety Message Handler (SMH) is shown in Figure 3. The function of SMH is to transfer and receive data from upper and down layers. Supporting safety application message protocol with decode and encode capability can auto choose suitable coding rules for different applications either be receiving or transmitting mode. It also can filter some useless information and abandon them to only keep defined kinds of data.

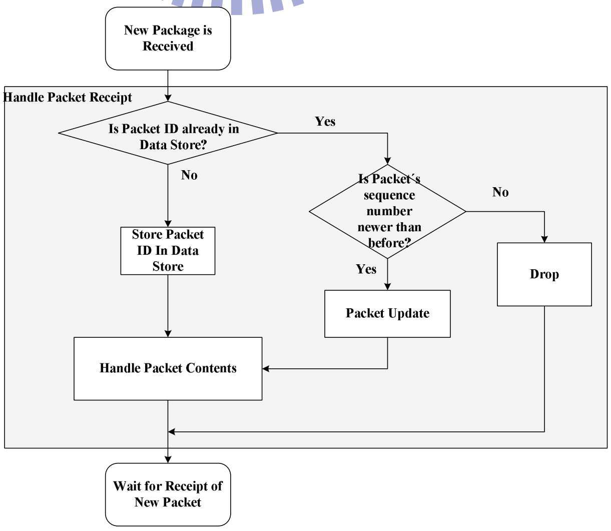


Fwd Collision Warning
Intersection Warning
Lane Change Warning
Position, Speed, Air Bag, Brake Status
Position, Speed, Yaw Rate, Vehicle Trail
Position, Turn signals, Speed, Headlights
Safety Message Handler: Sender Side
Position, Speed, Yaw Rate, Air Bag, Brake Status, Vehicle Trail, Turn signals, Headlight
Outgoing Message
Implementation Specific
J2735 Standard
Fwd Collision Warning
Intersection Warning
Position, Speed, Air Bag, Brake Status
Position, Speed, Yaw Rate, Vehicle Trail
Safety Message Handler: Receiver Side
Turn Signals, Headlights
IGNORED
Position, Speed, Yaw Rate, Air Bag, Brake Status, Vehicle Trail, Turn signals, Headlight
Incoming Message

Figure 3: The structure of SAE J2735

Source: DSRC SAE J2735 Rev 31.

There is an important issue on correctly processing and maintain all kinds of safety information from multiple vehicles. In Figure 4, we show a package processing step, each package has a unique identifier to make the differentiation with other sources.



New Package is Received
Handle Packet Receipt
Is Packet ID already in Data Store?
Yes
No
Is Packet's sequence number newer than before?
No
Yes
Store Packet ID In Data Store
Drop
Packet Update
Handle Packet Contents
Wait for Receipt of New Packet

Figure 4: A package processing step

Table 1 shows a list of SAE J2735 defined messages. All messages have their own purpose and fitting protocol.

Table 1: Content sets of SAE J2735

Contents	
1	MSG_Al_a_Carte (ACM)
2	MSG_BasicSafetyMessage
3	MSG_CommonSafetyRequest (CSR)
4	MSG_EmergencyVehicleAlert (EVA)
5	MSG_IntersectionCollisionAvoidance (ICA)
6	MSG_MapData (MAP)
7	MSG_NEMA_Corrections (NEMA)
8	MSG_ProbeDataManagement (PDM)
9	MSG_ProbeVehicleData (PVD)
10	MSG_RoadSideAlbert (RSA)
11	MSG_RTCM_Corrections (RTCM)
12	MSG_SignalPhaseAndTiming Message (SPAT)
13	MSG_SignalRequestMessage (SRM)
14	MSG_SignalStatusMessage (SSM)
15	MSG_TravelerInformation Message (TIM)
16	MSG_BasicSafetymessage_Verbose (BSV)

Basic safety driving information contains location, speed, and direction of moving or static vehicle. It is composed with two different kinds of elements. One contains upper necessary information of vehicle, and the other one depends on user selection. In Table 2, it is the VSC-A Basic Safety Message composing structure.

Table 2: VSC-A Basic Safety Message

VSC-A Basic Safety Message	
Part I	● Message Sequence Number ● Temporary ID ● Time ● Position Latitude, Longitude, Elevation, Accuracy ● Vehicle Speed, Heading, Steering Wheel Angle ● Vehicle Accelerations, Yaw Rate ● Brake Status ● Vehicle Length, Width
Part II	Vehicle Events Object
	Vehicle Path History Object
	Vehicle Path Prediction Object
	Vehicle Relative Positioning RTCM 1002 Data Object

In the newest version of SAE J2735 Rev. 31, it already defines 16 message protocols, 71 data structures and 147 basic data units. All of them are applied on V2V and V2R message formats in application layer. The objective is to standardize the message format on RSU and OBU with different hardware platform.

2.2 Inverse Perspective Mapping (IPM)

In general, the objective of camera calibration is to extract the intrinsic and extrinsic information of the camera and the extracted information could be used to reconstruct the 3D world coordinate. Nevertheless, the performance of camera calibration would depend on the perspective effect, lens distortion, and the number of cameras. An alternative method, namely inverse perspective mapping (IPM), was proposed to reconstruct the 3D world coordinates by using a single camera only. Broggi et al. [4, 5] utilized the IPM method and stereo cameras to detect obstacles in front of the vehicle, and implemented the parallel processor for image checking and analysis (PAPRICA) system Single Instruction Multiple Data (SIMD) computer architecture, to construct their obstacle and lane detection system, called GOLD (Generic Obstacle and Lane Detection) [5]. The GOLD implemented in the ARGO (derived from Argo and Argus, a research group from Italy) experimental vehicle made automatic driving possible. Ji [6] utilized IPM to get the 3D information of the front vehicle, and Cerri and Grisleri [7] presented the stabilized sub-pixel precision IPM image and the time correlation to estimate the possible driving space on highways. Muad et al. [8] used IPM to implement lane tracking and gave discussions of the factors which might have the influences on IPM. Tan et al. [9] combined IPM and the optical flow to detect obstacles for the lateral blind spot of the vehicle. Jiang et al. [10] proposed the fast IPM algorithm and used it to detect lanes and obstacles. Nieto et al. [11] introduced how to stabilize IPM images by using vanish point estimation. However in their approaches based on IPM, the planar objects such as lane markings were eliminated and the prominent objects like quasi-triangle pairs were reserved.

Our algorithm used the IPM’s property; therefore, the polar histograms derived from the IPM images could help to obtain the information of images in 1-D distributions. For separating from non-planar obstacles, we also constructed a novel method to detect and localize obstacles. With the intrinsic and extrinsic parameters from camera calibration, the obstacle detection system could establish a transformation table for mapping the coordinates of real-road surfaces into the distorted image coordinates. The objective of IPM method was to remove the perspective effects caused by cameras, and the higher performance of IPM methods made it possible to achieve better image processing results. Since IPM methods have been proven to be more efficient and applicable to real traffic conditions, we would focus on developing an accurate IPM algorithm for both normal lens and fisheye lens by improving the previous IPM methods. Our obstacle detection system aimed at detecting obstacles with either vertical or quasi-vertical edges. In fact, the obstacles with the significant height in vertical or quasi-vertical edges could be mapped to the radial lines of the transformed bird-view images. As a result, we could deal with the transformed images to extract the profile of edges and obtain the polar histogram for post-processing.

2.3 Obstacle Detection

The performance of those detection methods would obviously depend on the height, width, distance, and shape of an obstacle. There have been some other methods proposed for obstacle detection. Lai [12] used both of vision and the ultrasonic sensors on the mobile robot to detect the wall in the indoor environment. For the pedestrian detection, Curio et al. [13] used the contour, local entropy, and binocular vision to detect pedestrians. Bertozzi et al. [14] utilized stereo infrared cameras and three steps including warm area detection, edge-based detection, and v-disparity computation to detect pedestrians and used the morphological and thermal characteristics of heads to validate the presence of pedestrians. Though infrared cameras could perform well in either daytime or nighttime, the applications would be still restricted because of the higher prices of those cameras. There have existed many kinds of features such as symmetry, color, shadow, corner, Vertical/horizontal edges, texture, and vehicle light for vehicle detection [15]. Kyo et al. [16] used edges to detect possible vehicles and further validated the vehicles by the characteristics of symmetry, shadow, and differences in the gray-level average intensity, and Denasi and Quaglia [17] used pattern matching to detect and validate vehicles. These methods would usually fail if the obstacles did not match the defined models. For the general obstacle detection task, the optical flow-based and stereo-based methods have been most popular in recent researches. The optical flow based methods would detect obstacles by analyzing the differences between the expected and real velocity fields. Krueger et al. [18] combined the optical flow with odometry data to detect obstacles, but the optical flow-based methods would have the higher computational complexity and might fail if the relative velocity between obstacles and the detector was too small. For the stereo-based methods, Forster and Tozzi [19] utilized disparities of obstacles to detect obstacles and used a Kalman filter to track obstacles. However, stereo methods are highly dependent on the accuracy of identification of correspondences in the two images. In other words, searching the pairs of homogeneous points was much tougher for stereo-based methods. In recent years, there were two important subjects, including improving the accuracy of compensation estimation and obstacle detection. After an IPM image was acquired, a serious problem on resolution between the original and remapped images might be caused. Therefore, how to get an appropriate compensation result would be difficult, especially in our fish-eye lens approach. In Yang et al. [20], the compensation estimation was gained by the recursive method in trials and errors. Firstly, he chose randomly two pixels with a predefined distance to compare the optical flow values until gaining twenty pairs, and then used the median pair to be the value of compensation estimation. However, the

IPM remapped images may cause a serious problem for computing the optical flow values in case of the worse resolution. Furthermore, even if the recursion method was used to avoid choosing non-planar pixels, it was still probably to get similar or non-planar points when the values of optical flows were very close. In our approach, we adopted the edge features and images with time difference to improve the above problems in both static and dynamic environments. For dynamic environments, since the non-planar edge features may change more vibrantly than planar edge features, the values of compensation estimation can be easily determined by the compensated image with the minimum number of candidate pixels of obstacles. To improve stability and robustness of our system, we considered both the time interval and the earlier k frames to average and update the latest compensation estimation. For obstacle detection, in Ma et al. [21] approach, he adopted the pedestrian features and symmetrical property to search the possible positions of obstacles in the region of interest. Although the performance of their system was acceptable, the results would be not stable and robust with the detection rate in 58% ～92%. That was because the pedestrians' feet steps might be influenced by lane markings, shadows of trees, and any other planar noises.

2.4 Object Tracking

Besides foreground segmentation, objects tracking is another key module of surveillance systems. The purpose of tracking module is to track moving objects from one frame to another in an image sequences. And, a tracking algorithm needs to match the observed objects to the corresponding objects detected previously. Useful mathematical tools for objects tracking include the Kalman filter, the condensation algorithm, the dynamic Bayesian network, the geodesic method, etc. Hu et al. [22] presented there are four major categories of tracking algorithms: region-based tracking algorithms, active-contour-based tracking algorithms, feature-based tracking algorithms, and model-based tracking algorithms. Firstly, region-based tracking algorithms [23] were dependent on the variation of the image regions corresponding to the moving objects. The motion regions were usually detected by subtracting the background from the current image. Secondly, active contour-based tracking algorithms represented the outline of moving objects as contours. These algorithms had been successfully applied to vehicle tracking [24]. Thirdly, feature-based tracking algorithms performed the recognition and tracking of objects by extracting elements, clustering them into higher level features, and then matching the features between images. The global features used in feature-based algorithms include centroids, perimeters, areas, some orders of quadratures, and colors [25], etc. Fourthly, model-based tracking algorithms localized and recognized vehicles by matching a projected model to the image data. Tan et al. [26] proposed a generalized Hough transformation algorithm based on single characteristic line segment matching an estimated vehicle pose.

Besides, much research presented tracking algorithms with different categories integrated together for better tracking performance. McKenna et al. [27] proposed a tracking algorithm at three levels of abstraction: regions, people, and groups in indoor and outdoor environments. Each region has a bounding box and regions can merge and split. A human is composed of one or more regions under the condition of geometric constraints, and a human group consists of one or more people grouped together. Cucchiara et al. [28] presented a multilevel tracking scheme for monitoring traffic. The low-level consists of image processing while the high-level tracking is implemented as knowledge-based forward chaining production system. Veeraraghavan et al. [29] used a multilevel tracking approach with Kalman filter for tracking vehicles and pedestrians at intersections. The approach combined low-level image-based blob tracking with high-level Kalman filtering for position and shape estimation. An intermediate occlusion-reasoning module served the purpose of detecting occlusions and filtering relevant measurements. Chen et al. [30] proposed a learning-based automatic framework to support the multimedia data indexing and querying of spatio-temporal relationships of vehicle objects. The relationships were captured via unsupervised image/video segmentation method and object tracking algorithm, and modeled using a multimedia augmented transition network (MATN) model and multimedia input strings. Useful information was indexed and stored into a multimedia database for further information retrieval and query. Kumar et al. [31] presented a tracking algorithm combined Kalman filter-based motion and shape tracking with a pattern matching algorithm. Zhou et al. [32] presented an approach that incorporates appearance adaptive models in a particle filter to realize robust visual tracking and recognition algorithms. Nguyen et al. [33] proposed a method for object tracking in image sequences using template matching. To update the template, appearance features are smoothed temporally by robust Kalman filters, one to each pixel.

In regard to the cameras of surveillance systems, there are fixed cameras, active cameras and multiple cameras used for capturing the surveillance video. Kang et al. [34] presented an approach for continuous tracking of moving objects observed by multiple, heterogeneous cameras and the approach processed video streams from stationary and Pan-Tilt-Zoom cameras. Besides, much research used fixed cameras for the convenience of system construction and combining with the traditional surveillance system.

In this dissertation, we combined region-based and feature-based tracking methods and used plentiful features as effective inputs of tracking analysis. This proposed algorithm can do a good job to handle multi-objects with occlusion events or split events.

3. Structure of Safety Assistant Driving Telematics System

In this chapter, we will present our system structure and the details of hardware platform in each part. The system structure is composed of two sub-systems: On Board Vision-Based Detection System and Intersection Intelligent Surveillance System. In section 3.1, we use a diagram of the global system to show two sub-systems and their functions. In section 3.2, we present the structure on the vehicle with vision-based sensors and how they work. In section 3.3, we present the RSU structure with communication module in intelligent surveillance system with embedded platform.

3.1 System Overview

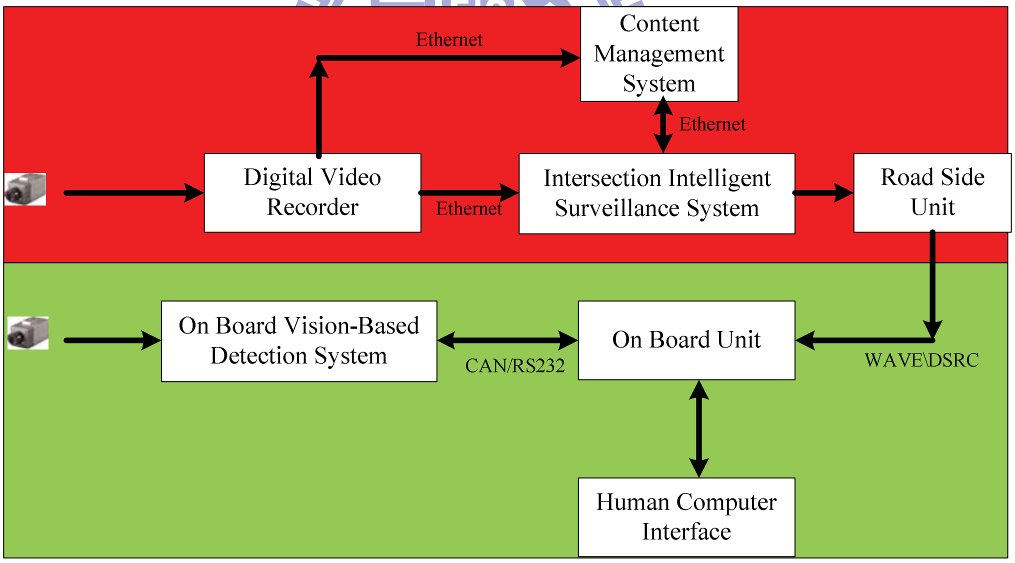


Figure 5: Structure of Safety Driving Assistant Telematics System

In Figure 5, there is an entire structure of our proposed system. The upper part is set on the intersection. Digital Video Recorder (DVR) responds to capture video from each camera on the different location. Intersection Intelligent Surveillance System analyzes and recognizes the content of video stream in four intelligent functions, moving object recognition, moving object tracking, collision prediction and coordinates calibration. Content Management System (CMS) also connects to DVR to receive video stream from DVR and configure parameters on Intelligent Surveillance System. In the end, Road Side Unit (RSU) gathers information together and encodes message, follows SAE J2735 protocol, to broadcast the intersection condition.

The lower part of Figure 5 is established on a vehicle. The on Board Vision-Based Detection System contains three intelligent functions, side collision warning (SCW), parking assistant system (PAS) and Obstacle Detection. SCW and PAS both are already developed on an embedded platform. Each of them captures the composite NTSC/PAL analog video signal from cameras, and exports the detection results via serial communication or CAN Bus to On Board Unit (OBU). Besides, the embedded platform also exports video stream with detection result on the screen to caution drivers. Obstacle Detection is a PC-Based Program, and it processes the video from reserve, right and left side of vehicle. Upper three directions' outcomes will be rearranged their priorities with the detection results of intersection condition. After all, driver can obtain the entire driving related information and makes a decision when an incident is occurring.

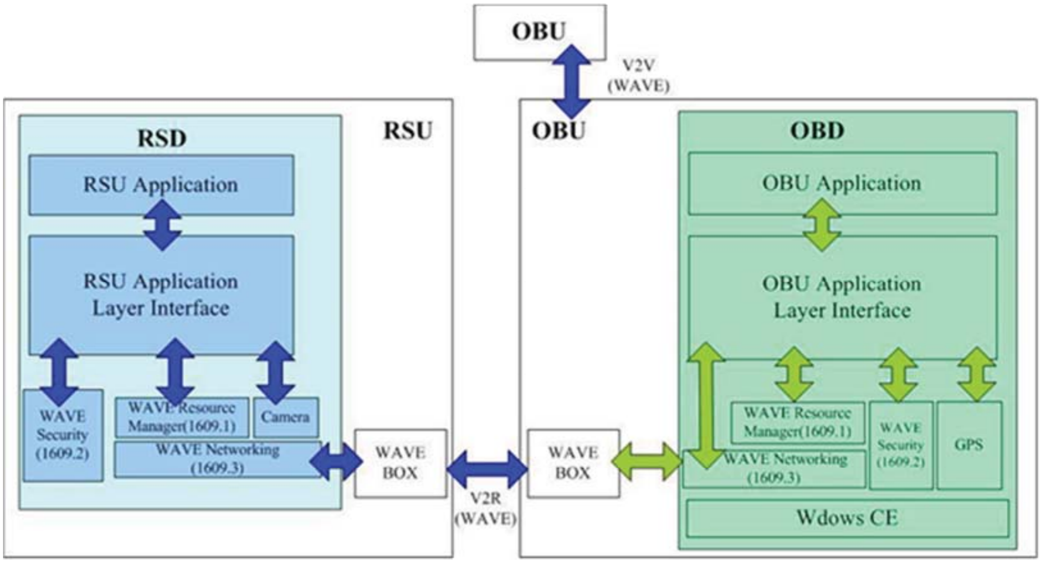


Figure 6: Structure of Telematics service platforms

Source: Institute for Information Industry

Figure 6 is the structure of OBU and RSU, they are two important facilities. The objective is to design a transmitting protocol for broadcasting local intersection condition to drivers. By WAVE/DSRC protocol, we can realize V2R and V2V applications via IEEE 1609.3 and IEEE 1609.4. Besides, we also follow DSRC SAE J2735 standard to deal with related safety information, and package them by ASN.1 encoding rules.

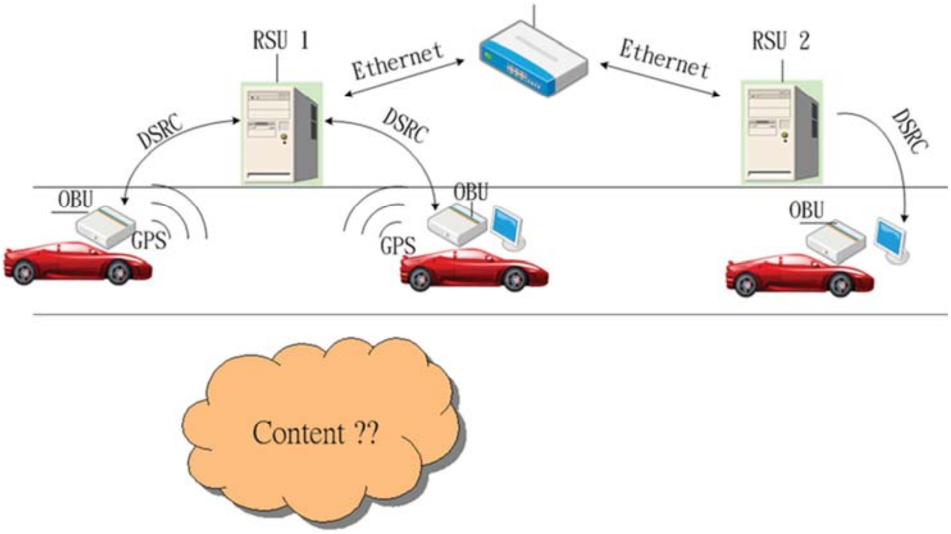


Figure 7: Current application on telematics system

Although ITS is a popular research, high accuracy traffic application detectors are still rare. The first useful traffic related information for driver is GPS. Because of the property of GPS signal, it is more accurate in country than in city. Besides, there are several kinds of useful information on the road that were already defined. In recent years, vision-based traffic detectors such as vehicle detector (VD), Image incident detection (IID) and self-adaptive traffic sign are developed for ITS applications. In our proposed safety driving assistant telematics system, we add vision-based intelligent detectors to provide safety detector results from the sensors on vehicle and intersection.

3.2 On Board Vision-Based Detection System

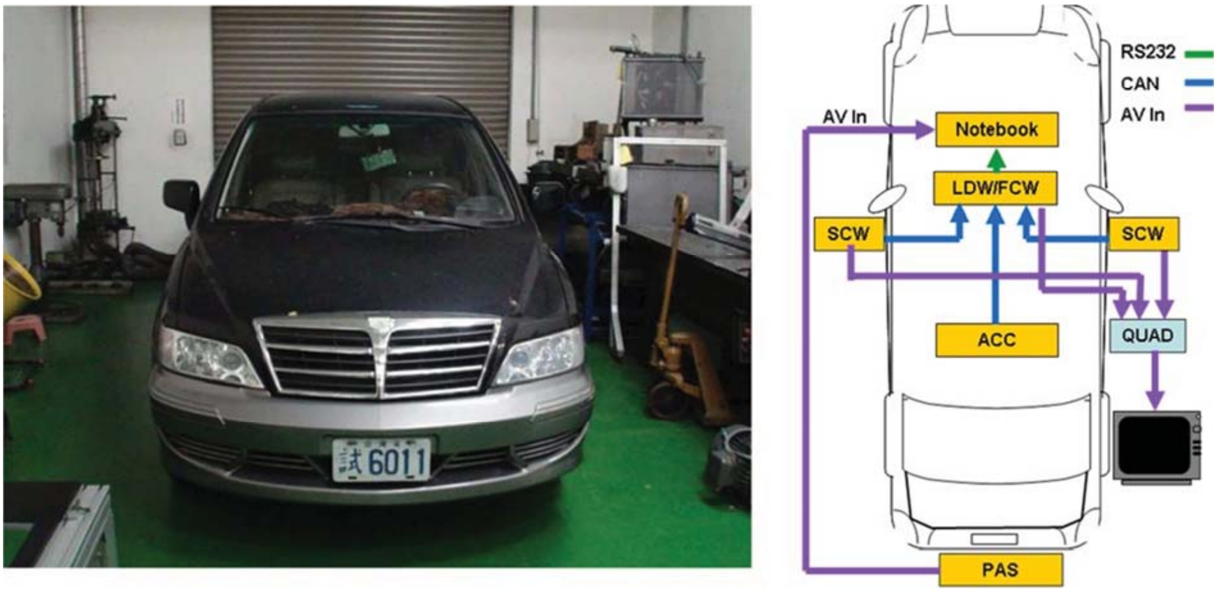


Figure 8: Vehicle Sensors

From Figure 8, it shows the sensors location and the assistant driving signal flow. There are three vision-based detectors, including side collision warning (SCW), parking assistant system (PAS), Obstacle detection. SCW and PAS are already integrated on embedded platform with CAN Bus protocol, and obstacle detector can work on laptop 10 fps. There is a WAVE Box on the vehicle to receive the information of intersection condition. WAVE Box communicating with notebook via RS232 protocol.

3.2.1 Embedded Platform Description

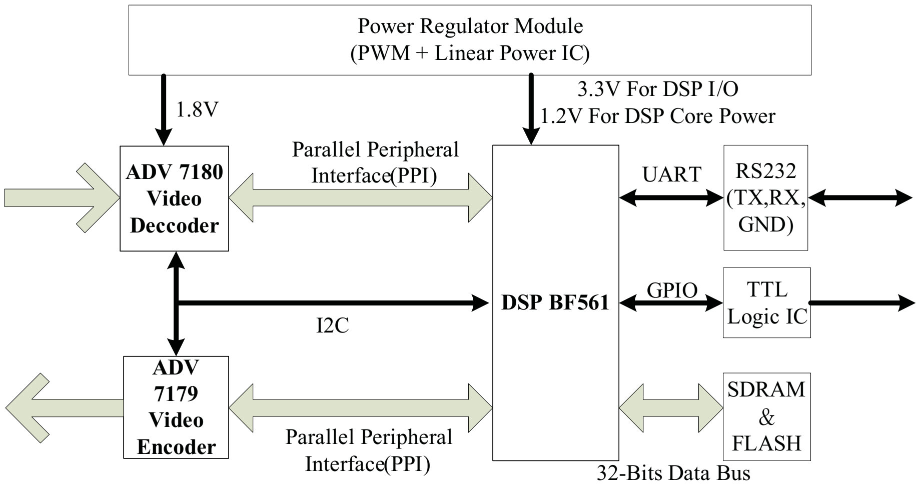


Figure 9: Structure of Intelligent Image Processing Embedded Platform

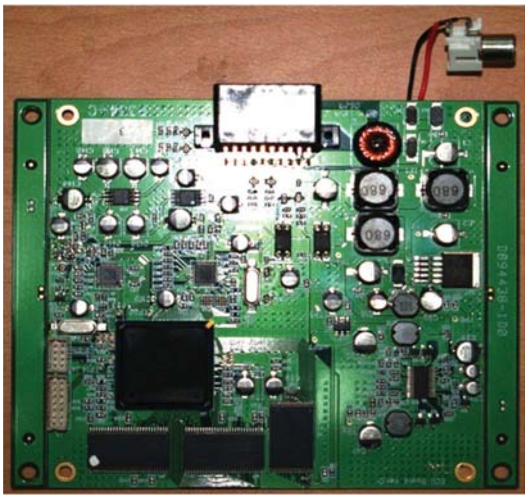


Figure 10: Structure of Embedded Platform PCB Design

In Figure 10, there is an embedded platform for SCW, PAS and intersection vision-based instant detector applications. It contains an ADI Blackfin BF561 Duo-Core DSP, Power Regulator, Video Decoder, Video Encoder, Memory, and I/O Modules. Detail features of Blackfin 561 are shown in Table 3.

Table 3: Features of BlackFin 561 DSP

Features	Peripherals
Dual symmetric 600MHz high performance Blackfin cores 328K bytes of on-chip memory	Two parallel input/output peripheral interface units supporting ITU-656 video gluesless interface to analog front end ADCs
Each Blackfin core includes: Two 16-bit MACs, two 40-bit ALUs, four 8-bit video ALUs, 40-bits shiffer	Two dual channel, full duplex synchronous serial ports supporting eight stereo I^2S channels
RISC-like register and instruction model for ease of programming and compiler-friendly support	Dual 16-channel DMA controllers and one internal memory DMA controller
Advanced debug, trace, and performance monitoring 0.8V-1.2V core V_{DD} 3.3V and 2.5V tolerant I/O	12 general-purpose 32-bit timer/counters, with PWM capability
256-ball mini-BGA and 297-ball PBGA package options	SPI-compatible port
	UART with support for IrDA®
	Dual watchdog timers
	48 programmable flags
	On-chip phase-locked loop capable of 1× to 63 × frequency multiplication

Source: ADI Blackfin 561 Datasheet

In this embedded platform, we consider the reference design from ADI BF561 EZ-KIT Lite manual. Due to our application, we change decoder chip into ADV 7180 for simple video stream decode steps, and redesign the power regulator module for 12V input. DSP controls video decoder and encoder by I2C protocol, and gets digital video data via 8 bits parallel peripheral Interface. In DSP inside, we can program the DMA register to control the DSP PPI catching data by itself automatically without wasting too much computing power on this kind of I/O data flow.

Power regulator module in this design, we choose the 2-channel PWM power regulator chip LM2642. At first, when 7V-24V (regular is 12V) is sent in the board, we string a schottky rectifiers and radial leaded PTC fuse to protect circuit. Table 4 is the SK34B's electrical characteristics, it limits the basic current forward and flow rate. FRX-090-60 is a fuse, its function is to avoid short connect occurrence. Besides protect circuit, we also consider the stability. We use DLW5BSN351SQ2 and two inductances to keep input voltage before entering LM2642 more smoothly. This embedded system total needs four kinds of voltages, including 5V, 3.3V, 1.8V and 1.2V. LM2642 outputs DSP needed two kinds of main power 3.3V and 1.2V. 3.3V supports DSP'I/O, RS232, RS485, TTL logic IC and 1.8V regulator's input voltage. 1.2V only supports DSP core power. 5V and 1.8V are both regulated by linear power regulator IC LM7805 and RT9166-18PX.

Table 4: SK34B Electrical Characteristics

Electrical Characteristics			
Average forward current	I_F(AV)	3.0A	T_J=120° C
Maximum surge current	I FSM	100A	8.3ms half-sine
Max repetitive reverse current	I_R	2A	F=1KHz
Max peak forward voltage (SK32B-SK34B)	V_{FM}	.50V	I_F=3A, T_J=25° C
Max peak reverse current	I_{RM}	.5mA	V_{RRM}, T_J=25° C
Typical junction capacitance	C_J	250pF	V_R=5.0V, T_J=25° C

As a result of our applications, end-user needs to see the exporting scream with intelligent detection result. For exporting a stable video output, we add a video OP after ADV 7179. Furthermore, we design a low-pass filter to reduce the noise influence.

3.2.2 Sinffer Equipment

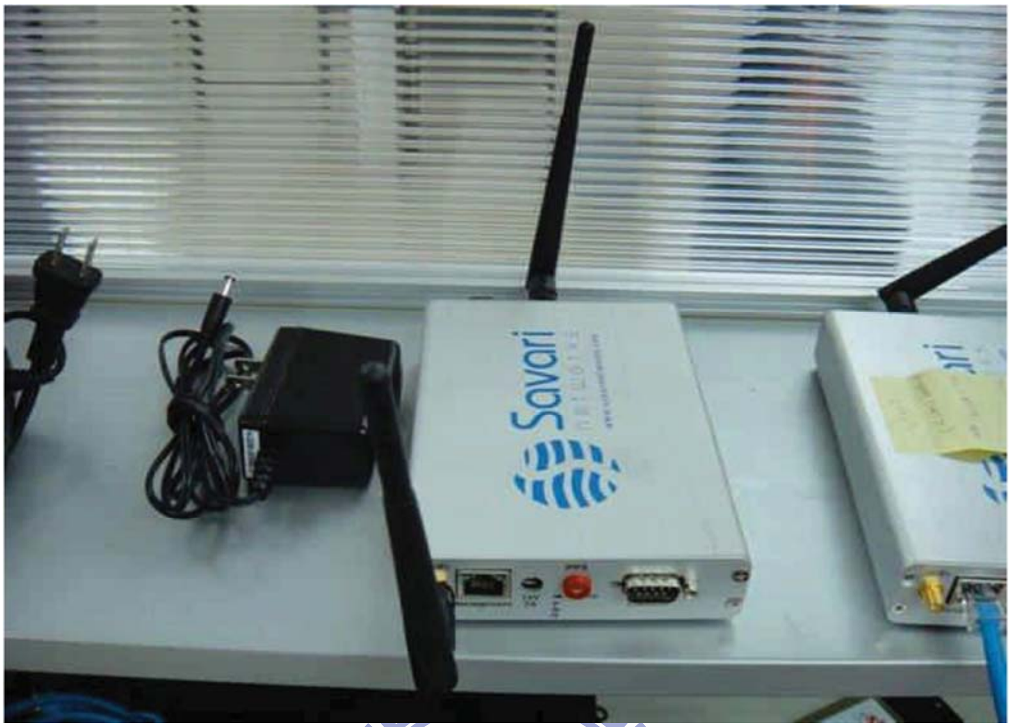
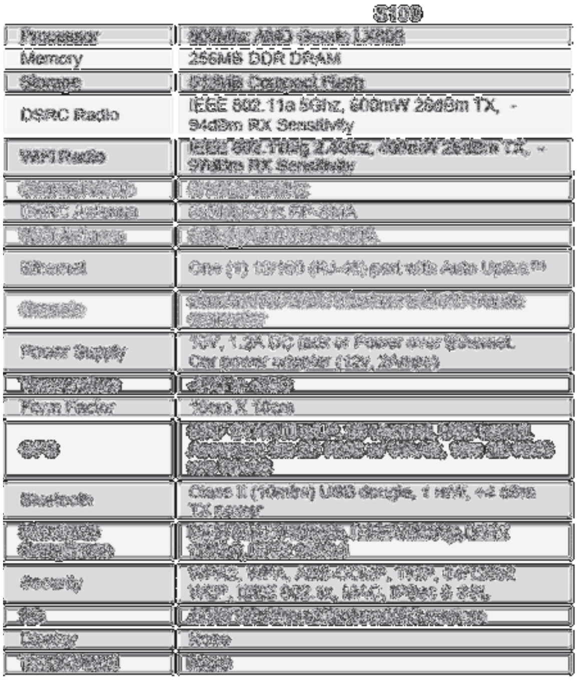


Figure 11 : S100 WAVE Box

For verifying the contents fits IEEE protocol, we choose Savari's S100 WAVE Box and Sirit's WSM Sniffer Card. The detail spec is shown in Figure 12.



	S100
Processor	500Mhz AMD Geode LX800
Memory	256MB DDR DRAM
Storage	512MB Compact Flash
DSRC Radio	IEEE 802.11a 5Ghz, 600mW 28dBm TX, -94dBm RX Sensitivity
WiFi Radio	IEEE 802.11b/g 2.4Ghz, 400mW 26dBm TX, -97dBm RX Sensitivity
[illegible]	[illegible]
DSRC Antenna	[illegible] RP-SMA
WiFi Antenna	[illegible]
Ethernet	One (1) 10/100 (RJ-45) port with Auto Uplink™
Console	[illegible]
Power Supply	[illegible], 1.2A DC jack or Power over Ethernet. Car power adapter (12V, 2Amps)
Temperature	[illegible]
Form Factor	15cm X 10cm
GPS	[illegible]
Bluetooth	Class II (10mtrs) USB dongle, 1 mW, +4 dBm TX power
[illegible]	[illegible]
Security	WPA2, WPA, AES-CCMP, TKIP, 64/128bit WEP, IEEE 802.1x, MAC, IPSec & SSL
OS	[illegible]
Display	None
[illegible]	[illegible]

Figure 12: S100 detail Spec

Figure 13: Sirit Sniffer Card (PCMCIA Interface)

Table 5: SAE J2735 WSM Message Protocol

1	1	1	1	1	4	2	Variable
WSM Version	Security Type	Channel Number	Data Rate	TxPwr_level	Provider Service Identifier	WSM Length	WSM DAta

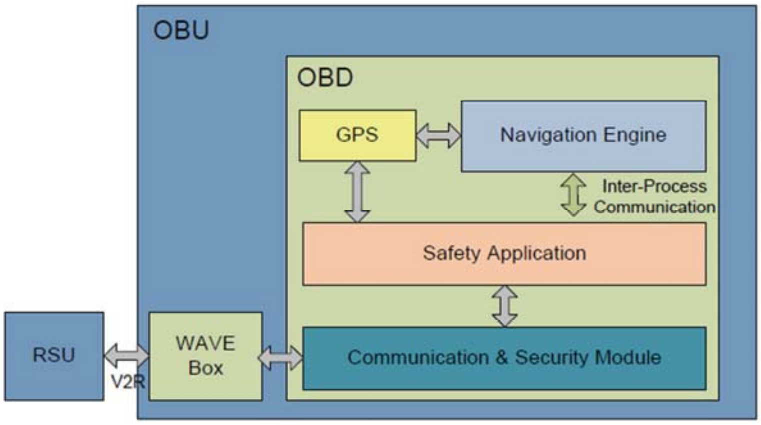


Figure 14: OBU Service and Navigation Structure

Figure 14 is the structure of OBU for Service and Navigation. RSU and OBU have a communication bridge by DSRC protocol via their WAVE device. For fitting protocol, when it transmits message, Safety Message handler will provide up layer related message for choosing encoding rule. So message can be distributed different decoder, for example BER, DER and XER. For the same reason, when receiving a message either from OBU or RSU in communication layer, first step needs to do is to judgment what kind of coding rule, and then use the suitable decoder to gain the message.

As mentioned previously, in our propose system, WAVE device supports DSRC ability for each safety applications. The working method is to develop a WSMP-IP converter on the WAVE device. When Safety application needs to send the message, it only sends UDP package to WSMP-IP converter to repackage into WSM format, and then transmits the repackage message via IEEE 1609 protocol stack. The flowchart in navigation application is shown in Figure 15.

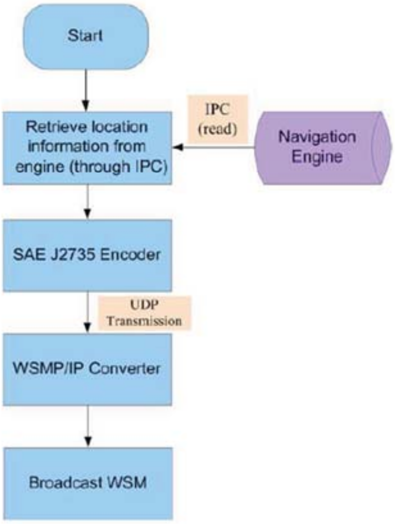


Figure 15: The flowchart of transmitting message

If OBU works successfully, it will receive traffic condition from many RSU on different intersections and detection result of the vehicle sensors. As a result of too much information, driver should get a sorted event list and suggestion

3.3 Intelligent Surveillance System

In this section, we propose a surveillance system with intelligent functions on an embedded platform. In traditional surveillance system, each Video Stream Input peer needs to connect to CMS for updating the current video stream as shown in Figure 16.

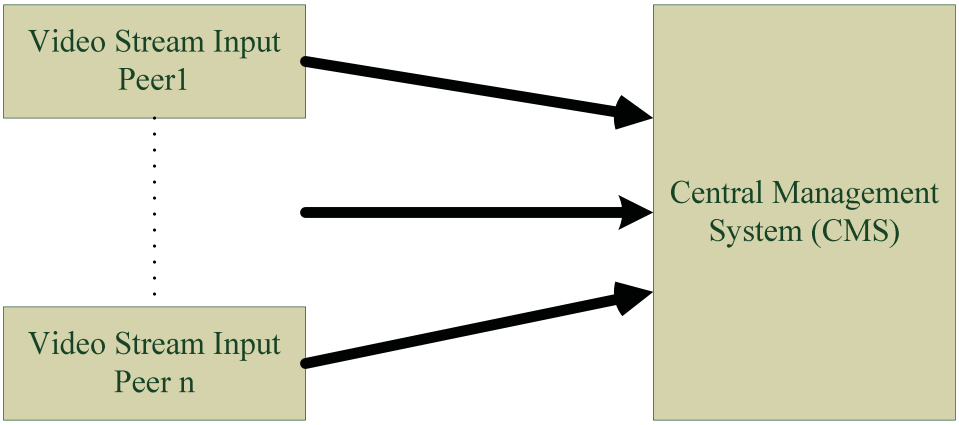


Figure 16: Traditional intelligent surveillance system

3.3.1 Improved Intelligent Surveillance System

In traditional structure, it needs a lot of manpower to pay attention on monitor, although the false alarm rate is less. Besides, this kind of design lacks adaptability for detectors to connect. Since, we propose an entire system with vision-based incident process platform and CMS. The vision-based incident process platform contains image incident detector and image incident collector. The former analyzes the image and then classifies incidents. It marks a sign on the video stream to distinguish the incident types. The image incident collector contains two parts. One is image storage server, and the other one is incident information storage server. The incident information storage server becomes a communication bridge between image storage server and incident detector, and also produces a trigger signal when some incident occurs.

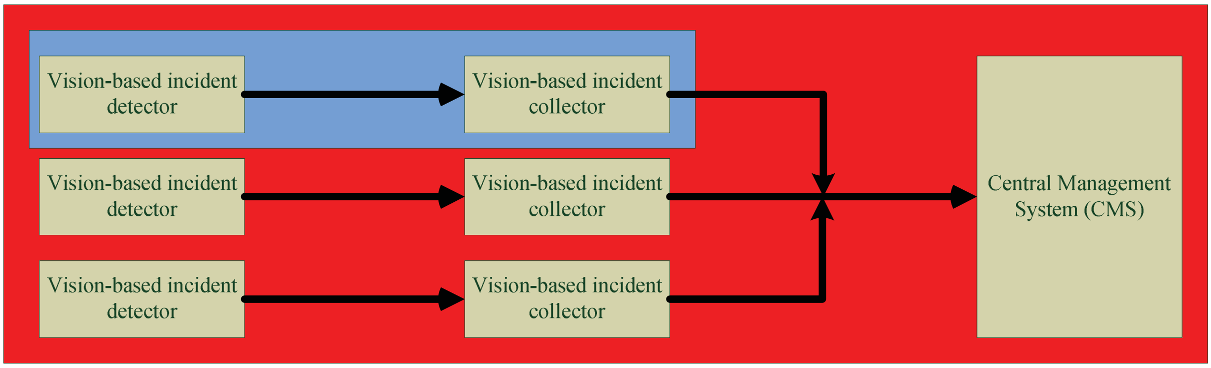

Vision-based incident detector
Vision-based incident collector
Vision-based incident detector
Vision-based incident collector
Vision-based incident detector
Vision-based incident collector
Central Management System (CMS)

Figure 17: Improved intelligent surveillance system

In Figure 17, red region means our proposed intelligent surveillance system. The blue region is vision-based incident process platform with incident detector and collector. Hence, we design RSU as a vision-based incident detector. After analyzing and classifying incident in video stream, it integrates all the information and uploads the results with correct content protocol to telematics transmitter via Ethernet. And then the WAVE Box will broadcast information in WAVED\DSRC channel to remind drivers to take care intersection condition.

3.3.2 Vision-based incident detector

In this section, we will discuss the basic functions on the vision-based incident detector. Due to the intersection traffic applications, we developed collision prediction and object tracking two algorithms. Since we utilize the fix camera, adaptive GMM background update method is suitable to extract foreground objects. Although GMM method needs heavy computing power, we modified the update period by the video frame rate per line. It means the cycle of update rate is about 16 seconds when the GMM background update method is running in single line update mode. Intersection traffic also have a serious problem of such kind of background method is the static object. When a vehicle stops before stop line to wait the traffic light, traditional GMM method make the wrong background image easily. Hence, our adaptive GMM Background algorithm have changeable update rate to adjust the parameters.

On our embedded platform, we capture composite NTSC/PAL video stream signal, and then ADV7180 decodes the analog signal to YUV 4:2:2 digital data into DSP Blackfin 561. In SDRAM, we allocate four image buffers for different processes. First frame is input frame, and it stores the data via DMA. Second frame is overlay frame, it overlays the lines or patterns on the image. Third frame is output frame, and it is prepared to export to ADV7179. The last frame is preprocessing frame, and it copies the data from input frame for image processing.

For our applications, tracking algorithm is the most important process. We extract the foreground and make a connected component image I_c. We accumulates I_c in each iteration to make accumulation image I_a. If pixel in I_c is foreground pixel, the value of I_a will add 1 until to threshold Th_a, otherwise I_a will decrease 1 until 0. Th_a is equal to 50 in our applications, and the value also can be estimated by the height of camera. The tracking index is connected component's label. From different time images, we first refer the objects' locations and ranges. If the location is close enough and range has overlap, it marks as a candidate object. The second step is to refer I_a image, from this image, we have the pixel's variance in past 50 frames. Therefore, we can filter some error foreground pixels and stable the tracking sequence. Finally, we package the whole information, and transfer the package via UART to PC client. At the same time, we also display the intersection incident and objects' location and trail on the scream.

4. Vision-Based Intelligent Technique

In this chapter we present four vision-based algorithms. In section 4.1 we will present a whole FLIPM method. It considers an optical path with mathematical calculating. In section 4.2 we will promote obstacle detection with single camera. Using the mapping image captured from the result of section 4.1, we analyze the characteristic to attend our objective. Finally, in section 4.3 we realize a vision-based dynamic distance gauge system for parking assistant system.

4.1 Fisheye lens inverse perspective mapping (FLIPM)

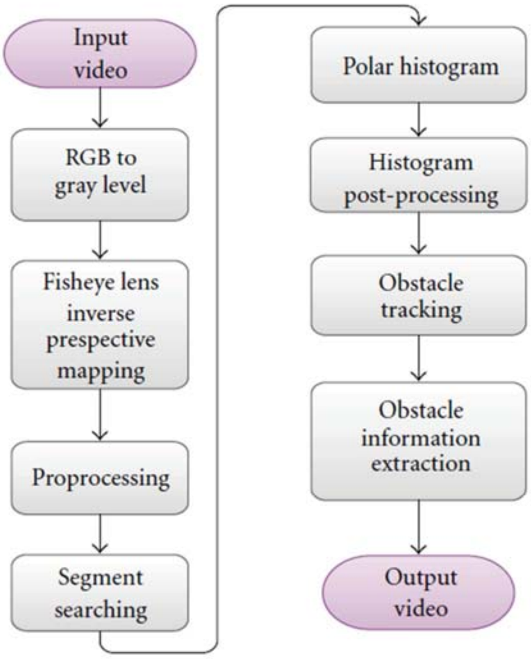


Figure 18: Fisheye lens inverse perspective mapping structure

Our overall systematic structure is illustrated in Figure 18. The obstacle detection is performed after obtaining the bird-view images of road surfaces captured by the camera mounted on the lateral side of the vehicle. The edge profile of road surfaces in bird-view images or temporal FLIPM difference image should be acquired, and then the segment searching algorithm will use the edge profile to get the feature radial lines which indicate the obstacles. After searching the feature elements, the polar histogram which represents the direction and size of obstacles will be computed. The histogram post-processing will also be used to filter out some noises and obstacles with shorter height. We still have to identify the detected obstacles and extract the relative information of the obstacle after the obstacle tracing process. After all the processes, we can obtain the final results in the output videos.

4.1.1 The Modified Normal Lens IPM Method

To find more practical applications and set up the appropriate mapping equations in our system, we modify the previous approaches proposed by Broggi et al. [2] and make the obstacle detection system more complete. Let u and v represent the image coordinate system and X,Y,Z be the world coordinate system where (X,Y, 0) indicates the road surface. L, D, H are the coordinates of the camera in the world coordinate system while θ and γ are the camera's tilt and pan angles, respectively. α,β are the horizontal and vertical aperture angles. m and n indicate the height and width of an image. O is the optic axis vector, and η_x,η_y are the vectors representing the optic axis vector O projected on the road surface and its perpendicular vector.

$$\begin{aligned} \mathrm{X} &= \mathrm{H} * \cot(\theta - \beta + \mathrm{u}\frac{2\beta}{\mathrm{m}-1}) * \cos(\gamma - \alpha + \mathrm{v}\frac{2\alpha}{\mathrm{n}-1}) + \mathrm{L} \\ \mathrm{Y} &= \mathrm{H} * \cot(\theta - \beta + \mathrm{u}\frac{2\beta}{\mathrm{m}-1}) * \sin(\gamma - \alpha + \mathrm{v}\frac{2\alpha}{\mathrm{n}-1}) + \mathrm{D} \end{aligned} \tag{4.1}$$

From Eq. (4.1), the vertical straight line in the image coordinate system can be represented by the set of pixels whose v coordinate value is constant. If we assume that $\mathrm{Kv} = \gamma - \alpha + \mathrm{v}\frac{2\alpha}{\mathrm{n}-1}$ is constant, then Eq. (4.1) will be simplified to Eq. (4.2).

$$\begin{aligned} \mathrm{X} &= \mathrm{H} * \cot(\theta - \beta + \mathrm{u}\frac{2\beta}{\mathrm{m}-1}) * \cos(\mathrm{Kv}) + \mathrm{L} \\ \mathrm{Y} &= \mathrm{H} * \cot(\theta - \beta + \mathrm{u}\frac{2\beta}{\mathrm{m}-1}) * \sin(\mathrm{Kv}) + \mathrm{D} \end{aligned} \tag{4.2}$$

After simple calculations, we can obtain Eq. (4.3) from Eq. (4.2), which is shown in Figure 19.

$$\mathrm{X} - \mathrm{L} = (\mathrm{Y} - \mathrm{D}) * \cot(\mathrm{Kv}) \tag{4.3}$$

The equation Eq. (4.3) means that a vertical straight line in the image which represents the vertical edge of obstacles or other planar markings in the world coordinate system will be projected into a straight line whose prolongation will pass the vertical projection point of the camera on the world surface.

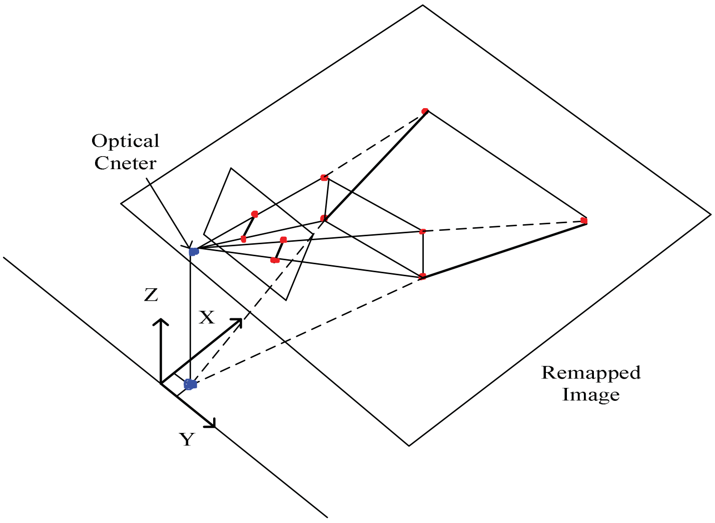


Figure 19 : The vertical line projection of Eq. (4.1)

Similarly, the horizontal straight line in the image coordinate system can be represented by the set of pixels whose u coordinate value is a constant. If we assume $\mathrm{Ku} = \theta - \beta + \mathrm{u}\frac{2\beta}{\mathrm{m}-1}$ is constant, then Eq. (4.1) will be also simplified to Eq. (4.4).

$$\begin{aligned} \mathrm{X} &= \mathrm{H} * \cot(\mathrm{Ku}) * \cos(\gamma - \alpha + \mathrm{v}\frac{2\alpha}{\mathrm{n}-1}) + \mathrm{L} = \mathrm{K} * \cos(\gamma - \alpha + \mathrm{v}\frac{2\alpha}{\mathrm{n}-1}) + \mathrm{L} \\ \mathrm{Y} &= \mathrm{H} * \cot(\mathrm{Ku}) * \sin(\gamma - \alpha + \mathrm{v}\frac{2\alpha}{\mathrm{n}-1}) + \mathrm{D} = \mathrm{K} * \sin(\gamma - \alpha + \mathrm{v}\frac{2\alpha}{\mathrm{n}-1}) + \mathrm{D} \end{aligned} \tag{4.4}$$

Thus, we can derive Eq. (4.5) from Eq. (4.4), which is shown in Figure 20.

$$(\mathrm{X} - \mathrm{L})^2 + (\mathrm{Y} - \mathrm{D})^2 = \mathrm{K}^2 \tag{4.5}$$

The equation Eq. (4.5) means that a horizontal straight line in the image will be projected to an arc on the world surface.

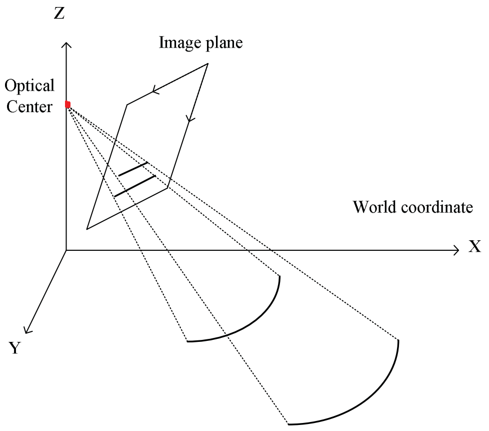


Figure 20: The projected result of Eq. (4.5)

In order to modify the original IPM model, we propose a new pair of transformation equations for two expected results. First, a vertical straight line in the image will still be projected to a straight line whose prolongation will pass the vertical projection point of the camera on the world surface. Second, a horizontal straight line in the image will be projected to a straight line instead of an arc on the world surface. The results can be verified by the similar triangle theorem. With some prior knowledge such as the assumptions on flat roads, intrinsic and extrinsic parameters, we will be able to reconstruct a 2D image without the perspective effect. The illustrated figures and expected results are shown in Figure 21

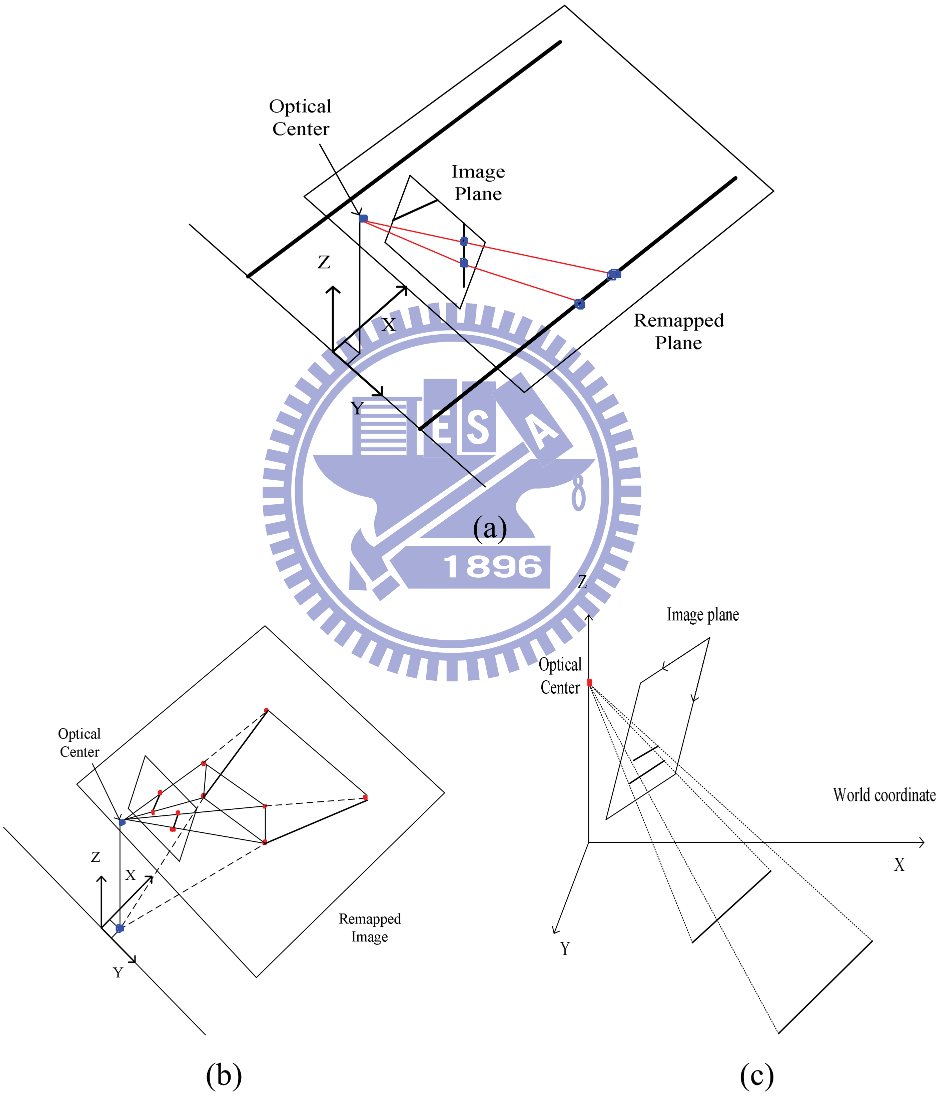


Figure 21: The figures and expected results

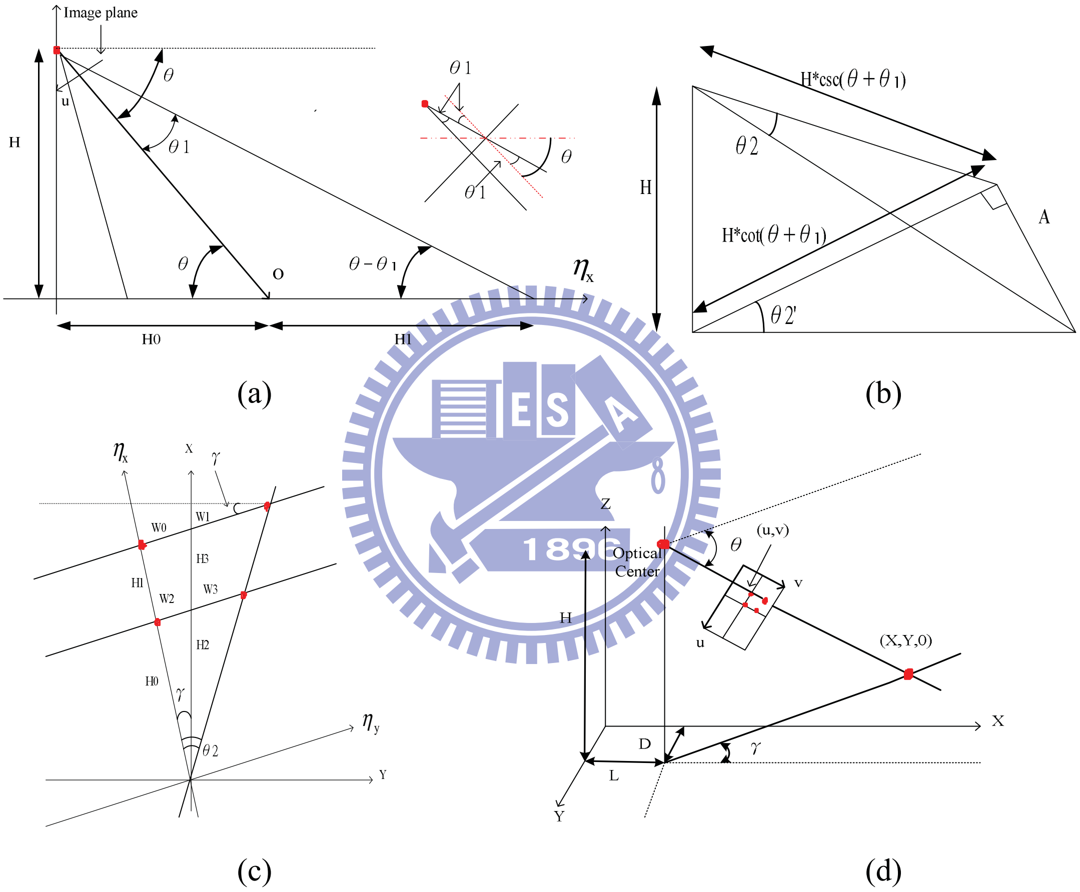
(a) perspective effect removing (b) a vertical straight line in the image will be projected to a straight line whose prolongation will pass the vertical projection point of the camera on the world surface (c) a horizontal straight line in the image will be projected to a straight line instead of an arc on the world surface.

By referring to the notations, the diagrams of relationship between the image coordinate system and the world coordinate system are shown in Figure 22. We will derive a new pair of transformation equations by simple mathematical computations in triangular functions. From Figure 22(a) and (b), we can obtain (6) and (7).

$$\rightarrow \theta_1 = \mathrm{u}\frac{2\beta}{\mathrm{m}-1} - \beta$$

$$\rightarrow \mathrm{H}_0 = \mathrm{H} * \cot(\theta)$$

$$\rightarrow \mathrm{H}_0 + \mathrm{H}_1 = \mathrm{H} * \cot(\theta + \theta_1) \qquad (4.6)$$



(a)

(b)

(c)

(d)

Figure 22: The geometrical relations of the image and world coordinate system for deriving our equations.

$$\rightarrow \theta_2 = \mathrm{v}\frac{2\alpha}{\mathrm{n}-1} - \alpha$$

$$\rightarrow \tan(\theta_2^{'}) = \tan(\theta_2) * \sec(\theta + \theta_1) \qquad (4.7)$$

Figure 22(c) describes how the points in the first quadrant of the image coordinate system will be projected onto the road surface. If the world coordinate of camera is (0, 0, H), we will finally obtain Eq. (4.8) by the geometrical descriptions in Figure 22(c) (d) and the length of each segment listed below.

$$\rightarrow \mathrm{H}_0 = \mathrm{H} * \cot(\theta)$$

$$\rightarrow \mathrm{H}_0 + \mathrm{H}_1 = \mathrm{H} * \cot(\theta + \theta_1)$$

$$\rightarrow \mathrm{H}_2 = \mathrm{H}_o * \sec(\gamma)$$

$$\rightarrow \mathrm{H}_2 + \mathrm{H}_3 = (\mathrm{H}_0 + \mathrm{H}_1) * \sec(\gamma)$$

$$\rightarrow \mathrm{W}_0 = (\mathrm{H}_0 + \mathrm{H}_1) * \tan(\gamma)$$

$$\rightarrow \mathrm{W}_0 + \mathrm{W}_1 = (\mathrm{H}_0 + \mathrm{H}_1) * \tan(\theta_2) * \sec(\theta + \theta_1)$$

$$\rightarrow \mathrm{W}_2 = \mathrm{H}_0 * \tan(\gamma)$$

$$\rightarrow \mathrm{W}_2 + \mathrm{W}_3 = \mathrm{H}_0 * \tan(\theta_2) * \sec(\theta + \theta_1)$$

$$\Rightarrow \mathrm{X} = \mathrm{H}_2 + \mathrm{H}_3 + \mathrm{W}_1 * \sin(\gamma) = \mathrm{H} * \cot(\theta + \theta_1) * \left[\cos(\gamma) + sec(\theta + \theta_1) * \tan(\theta_2) * \sin(\gamma)\right]$$

$$\Rightarrow \mathrm{Y} = \mathrm{W}_1 * \cos(\gamma) = \mathrm{H} * \cot(\theta + \theta_1)\left[- \sin(\gamma) + sec(\theta + \theta_1) * \tan(\theta_2) * \cos(\gamma)\right] \qquad (4.8)$$

Now, we have obtained the forward transformation equations, and the backward transformation equations shown in Eq. (4.9) can also be obtained easily by some mathematical computations in inverse functions.

$$\Rightarrow \theta_1 = \cot^{-1}\left(\frac{\mathrm{X} * \cos(\gamma) - \mathrm{Y} * \sin(\gamma)}{\mathrm{H}}\right) - \theta \;\; ; \;\; \theta_2 = \tan^{-1}\left(\frac{\mathrm{X} * \sin(\gamma) + \mathrm{Y} * \cos(\gamma)}{\mathrm{H} * \csc(\theta + \theta_1)}\right)$$

$$\Rightarrow \mathrm{u} = \frac{\mathrm{m} - 1}{2\beta} * \left(\cot^{-1}\left(\frac{\mathrm{X} * \cos(\gamma) - \mathrm{Y} * \sin(\gamma)}{\mathrm{H}}\right) - \theta + \beta\right)$$

$$\Rightarrow \mathrm{v} = \frac{\mathrm{n} - 1}{2\alpha} * \left(\tan^{-1}\left(\frac{\mathrm{X} * \sin(\gamma) + \mathrm{Y} * \cos(\gamma)}{\mathrm{H} * \csc(\theta + \theta_1)}\right) + \alpha\right) \qquad (4.9)$$

4.1.2 Fisheye Lens Inverse Perspective Mapping (FLIPM)

Forster et al. [19] proposed a camera spherical projection model to implement the endoscope image formation process and utilized the warping transformation equations to correct the radial distortion. The warping transformation equation pairs and its inverse pairs are shown in Eq. (4.10) and Eq. (4.11). The coordinate (X,Y,Z) is the position of point in the 3D world coordinate system, (u_1,v_1) is the coordinate in the un-distorted image, and (u,v) is the coordinate in the distorted one.

$$\mathrm{X} = \frac{\mathrm{f} * \mathrm{u}}{\sqrt{\mathrm{R}^2 - \mathrm{u}^2 - \mathrm{v}^2}} \;\; ; \;\; \mathrm{Y} = \frac{\mathrm{f} * \mathrm{v}}{\sqrt{\mathrm{R}^2 - \mathrm{u}^2 - \mathrm{v}^2}} \qquad (4.10)$$

$$u = \frac{R * X}{\sqrt{f^2 + X^2 + Y^2}} \;\; ; \;\; v = \frac{R * Y}{\sqrt{f^2 + X^2 + Y^2}} \tag{4.11}$$

Where f is the focal length of camera, and R is the radius of the sphere. We modify and redefine that model for our applications in this dissertation. We regard the X1-Y1 plane as an undistorted image plane and the u-v plane as the distorted one, thus we can derive the modified equations in Eq. (4.12) and Eq. (4.13).

$$\begin{cases} u_1 = \dfrac{f * u}{\sqrt{R^2 - u^2 - v^2}} = k_1 * \sqrt{\dfrac{u^2 + v^2 * \sin^2(\theta_1)}{\cos^2(\theta_1)}} \\ v_1 = \dfrac{f * v}{\sqrt{R^2 - u^2 - v^2}} = k_1 * \sqrt{\dfrac{v^2 + u^2 * \sin^2(\theta_2)}{\cos^2(\theta_2)}} \end{cases} \tag{4.12}$$

$$\begin{cases} u = \dfrac{R * u_1}{\sqrt{f^2 + {u_1}^2 + {v_1}^2}} = \dfrac{u_1}{k_1\sqrt{1 + (\tan^2(\theta_1) + \tan^2(\theta_2))}} \\ v = \dfrac{R * v_1}{\sqrt{f^2 + {u_1}^2 + {v_1}^2}} = \dfrac{v_1}{k_1\sqrt{1 + (\tan^2(\theta_1) + \tan^2(\theta_2))}} \end{cases} \tag{4.13}$$

Where $k_1 = \dfrac{f}{R}$, and $\theta_1 = \sin^{-1}(\dfrac{u}{\sqrt{R^2 - v^2}}) = \tan^{-1}(\dfrac{u_1}{f})$ and $\theta_2 = \sin^{-1}(\dfrac{v}{\sqrt{R^2 - u^2}}) = \tan^{-1}(\dfrac{v_1}{f})$ are the angles between the lines connected the horizontal or vertical direction projection point of an image point with the optical center on the optical axis. The equation Eq. (4.14) instead of Eq. (4.13) will be used through this dissertation since Eq. (4.13) may produce many non-pixel-values of the image. We also can obtain the distorted or un-distorted images no matter if the focal length is known or not by tuning the parameter k$_1$.

1. The Complete Fisheye Lens Inverse Perspective Mapping

A fisheye lens inverse perspective mapping (FLIPM) algorithm consists of two parts, the forward and backward mapping algorithm. The objective of the forward mapping algorithm is to search the dimensions or ranges of remapped images, which can be illustrated in Figure 23.

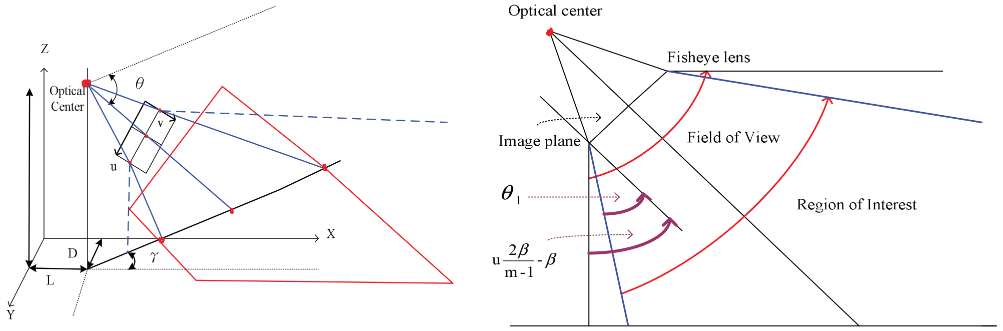


Figure 23 : The original and adjusted scope

The dimensions of scopes are only related to the view-ranges of a camera, that is to say, either the use of the normal lens or fisheye lens with fixed tile and pan angle will determine the factors of influences. In order to reduce the computational loadings in use of the tangent and secant triangular functions, we restrict the scope of a camera by narrowing down its view-range. Without loss of generality, we still keep the broadest range of scopes and minimize discarding far and fringe information. Furthermore, we narrow down the view-angles by using Snell's Law as shown in Eq. (4.14) where IR simulates the index of refraction and controls the scopes of resultant ranges. The range of IR is between 1.3~1.7 for glass-based lens.

$$\begin{aligned} \sin(\mathrm{u}\frac{2\beta}{\mathrm{m}-1}-\beta) &= \mathrm{IR}*\sin(\theta_1) \\ \sin(\mathrm{v}\frac{2\alpha}{\mathrm{n}-1}-\alpha) &= \mathrm{IR}*\sin(\theta_2) \end{aligned} \tag{4.14}$$

The angles θ_1 and θ_2 can be substituted into Eq. (4.9) to compute the extreme values about the coordinate values of X, Y, and in this way we will obtain the dimension of the remapped image. The backward mapping algorithm is different from the forward one because a plus of the radial distortion correction step should make it more rational. We firstly consider the ideas of the backward mapping algorithm by Figure 24.

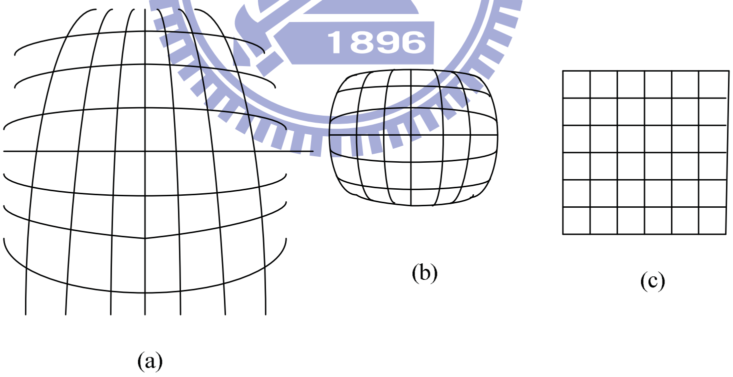


Figure 24: Illustrations for distortion images

(a) the real scene image, (b) the distorted image, and (c) the desired image

Since the images captured by the fisheye cameras which can be shown in Figure 24(a) have the perspective effects and distortions, we have to remove those undesired effects to acquire the available images just like Figure 24(b) in pursuit of Figure 24(c) where the perspective effect and distortion have been completely removed. Thus so, we can derive the backward mapping algorithm by modifying Eq. (4.9) as Eq. (4.15). We also complete the distortion correction process by using Eq. (4.13) and the derived formulas of angles in Eq. (4.15). By tuning the parameter of IR and k_1, we will easily obtain the undistorted and perspective effect removed images.

$$\theta_1 = \cot^{-1}(\frac{X * \cos(\gamma) - Y * \sin(\gamma)}{H}) - \theta$$

$$\theta_2 = \tan^{-1}(\frac{X * \sin(\gamma) + Y * \cos(\gamma)}{H * \csc(\theta + \theta_1)}) \qquad (4.15)$$

4.2 Obstacle Detection with single Camera

In this section we develop an obstacle detection algorithm by using both spatial and temporal information of the FLIPM method. We use a single fisheye camera mounted on the lateral side of the vehicle to detect obstacles. The definitions of obstacles in this dissertation are the objects with the height shorter than a threshold and with non quasi-vertical edges. The straight line in the vertical direction in the images represents the vertical edges of obstacles in the world coordinate system, and will be projected to a straight line whose prolongation will pass the vertical projection point of the camera on the world surface. To illustrate our systematic mechanism more clearly, we will introduce the obstacle detection algorithm in the following parts, including some image pre-processing steps, feature selection, histogram analysis, object tracking, and information extraction.

4.2.1 The Pre-Process

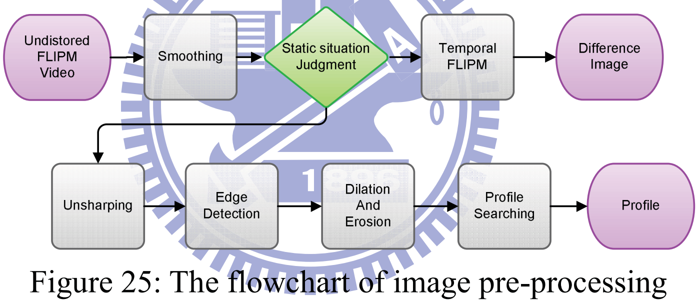


Figure 25: The flowchart of image pre-processing

We have to simplify the image patterns for our following procedures by some image preprocessing techniques shown in Figure 25. At first, the remapped image will be smoothed by mean filter to reduce the noises resulted from FLIPM transformation. Our developed equations in FLIPM have the advantages of IPM in removing the information of height and can help to detect the obstacles on the surface of roads. We also propose two different strategies toward feature extraction. We use the profile image which will be introduced next to extract the feature series when the detected objects and our cameras are relatively motionless, otherwise we acquire the features by the obstacle-sensitive temporal FLIPM difference image which will be clarified in Section 4.3.

4.2.2 Profile image

The obvious edges of obstacles will be essential for extracting the profile images. We hence enhance the edges by the unsharp mask at this time to make up for over-blurred images, and detect edges by simple Sobel operations. The binary images can be obtained by thresholding after edge detecting of the remapped image, and we have to use the morphological operations on dilation and erosion to get the useful edges for our processes. As for extraction of the feature segments, we remodel the thinning algorithm introduced in [20] in thinning the binary edges in order to meet our real-time needs in the applications of ITS. We turn to use the center pixel of a mask to extract the exterior profile of a pattern without checking the conditions of patterns iteratively. Figure 26 shows the processed results of our profile image searching.

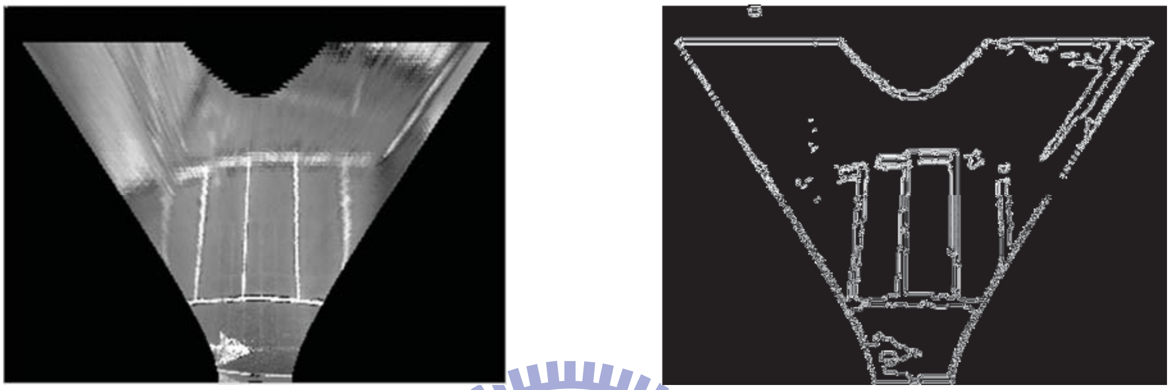
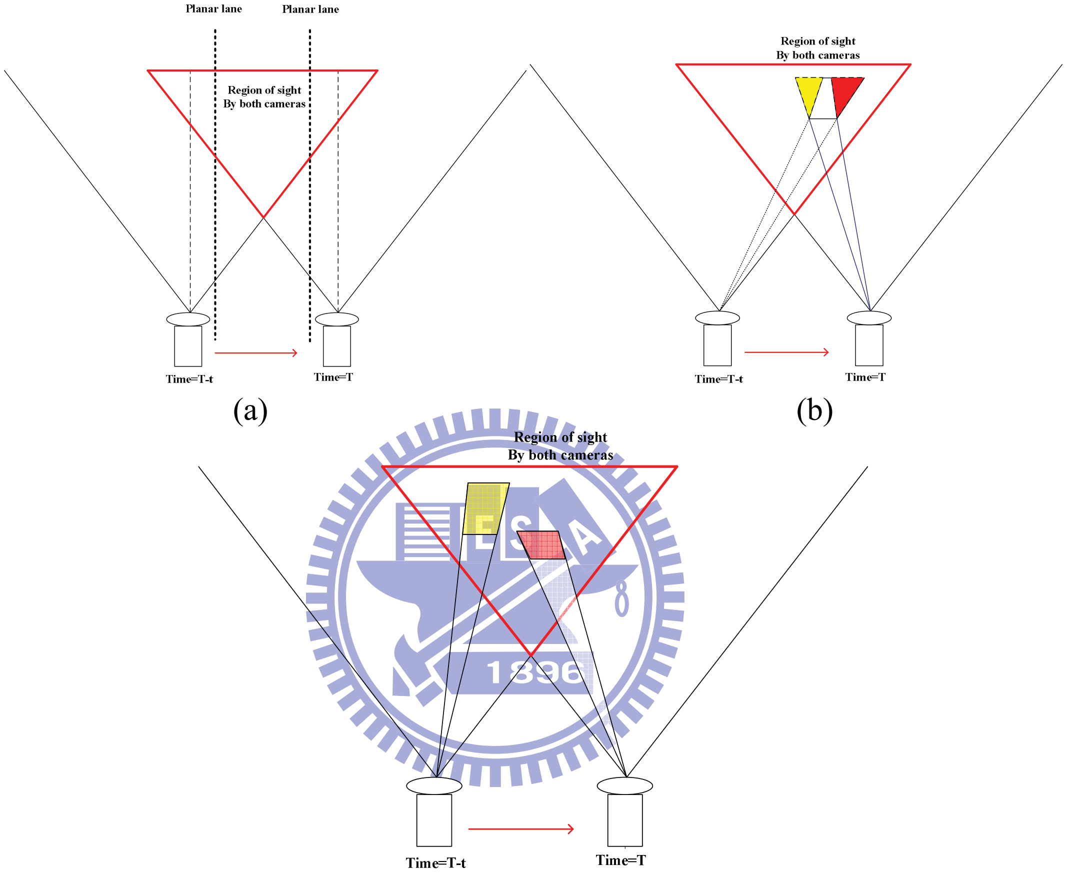


Figure 26: The results in the profile searching process
(a) the remapped image (b) the profile image.

4.2.3 The temporal FLIPM difference image

The objective of temporal FLIPM difference process is to simulate the stereo vision of captured images. The stereo IPM can keep the non-plane objects and remove the plane objects such as lane-markings, shadows by comparing the differences between the left and right remapped image, which will be illustrated in Figure 27.



(a)

(b)

(c)

Figure 27: Illustrations for the temporal FLIPM difference image

(a) the planar object patterns and (b) non-planar object patterns (c) Moving non-planar object patterns

In Figure 27(a), it presents a result of planar object patterns. The planar pattern in IPM image has a shift movement in different time; hence we can easily remove the planar pattern via accurate enough shift movement to compensate the different images. Figure 27(b) shows a result of a static non-planar object on the IPM image. By the projection effect, the non-planar point also can be projected to the ground with the farer distance. Therefore the difference of t-T and t frame can extracts the non-planar object. The same result also appears in the moving object case such as Figure 27(c).

According to the stereo IPM method [21], two cameras should be used to acquire the sufficient information of overlapped regions. Since this thesis focuses on using a single camera, we take advantage of time difference to simulate the effects of stereo cameras. As Figure 27 shows, we have to address on two important issues, selections of time interval and the shift displacement of the remapped image.

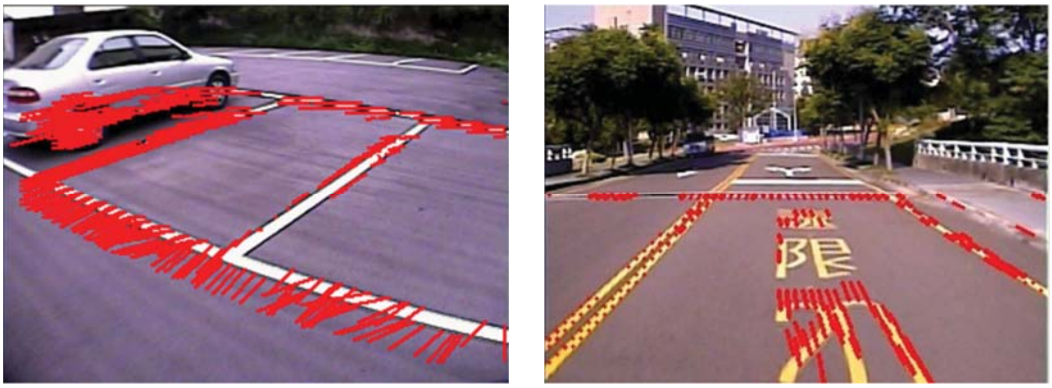
1. Feature Analysis

There are many kinds of local features that one can track. Features which are used to estimate ground movement based on its motion, that is to find these features from one frame in a subsequent frame of the video stream. Obviously, if we pick a point on a large blank wall then it won't be easy to find that same point in the next frame of a video. If all points on the wall are identical or even very similar, then we won't have much luck tracking that point in subsequent frames. On the other hand, if we choose a point that is unique then we have a pretty good chance of finding that point again. In practice, the point or feature we select should be unique, or nearly unique, and should be parameterizable in such a way that it can be compared to other points in another image. Therefore, we might be tempted to look for points that have some significant change within neighboring local area that is the good features which have a strong derivative in spatial domain. Another characteristic of features is about the position of the image. Due to the objective of the following procedure is to estimate the ground movement information, features lie on the ground region is useful for the following ground movement estimation algorithm. According to above analysis, a good feature to track should have two characteristics. First, a feature should have strong derivative which could assist us to track them and obtain a precise motion. Then, the position of feature should be restricted on the road region (non-obstacle region). The features which we will use them to estimate the ground movement information should conform the above two characteristic, these features will be suitable for estimating ground movement information.

To consider the first characteristic – strong derivative, a point to which a strong derivative is associated may be on an edge of some kind. Then considering this property of edge we employ the Sobel operator to find out the edge of image. The points which be extracted by edge detection are used to be feature points, which then calculate optical flow for all of these feature points. These edge points and its optical flow of image is shown in Figure 28. But a problem is arising as depicted in Figure 28(b), it could look like all of the other points along the same edge. An ambiguous optical flow will happen when the edge points parallel to the direction of motion. It turns out that strong derivative of a single direction which is not enough. However, if strong derivatives are observed in two orthogonal directions then we can hope that this point is more likely to be unique. For this reason, many track-able features in the image that are corners. Intuitively, corners are the points that contain enough information to be picked out from one frame to the next. We examined by the most commonly used definition of a corner was provided by Harris[25]. This definition relies on the matrix of the second-order derivatives ($\partial^2 x$, $\partial^2 y$, $\partial x \partial y$) of the image intensities. We can think of the second-order derivatives of images, taken at all points in the image, as forming new second-derivative images or, when combined together, a new *Hessian* image. This terminology comes from the Hessian matrix around a point, which is defined in two dimensions by:

$$H(p) = \begin{bmatrix} \dfrac{\partial^2 I}{\partial x^2} & \dfrac{\partial^2 \mathrm{I}}{\partial x \partial y} \\ \dfrac{\partial^2 I}{\partial y \partial x} & \dfrac{\partial^2 I}{\partial y^2} \end{bmatrix} \tag{4.16}$$

By using Harris corner definition, the result of corner detection and its corresponding optical flow is shown in Figure 29. It is obvious to observe that motion of corner is relatively accurate than edge point. Although corners have more precise optical flow, another problem is arising that is position of corner almost lie on obstacle region such as vehicle component. The Figure 29(b) shows the result of corner detection in a common driving condition, we can see that the feature points are nearly located on obstacle regions. These feature points are not suitable to use to estimate the ground movement.



(a) (b)

Figure 28: Results of edge detection and its corresponding optical flow

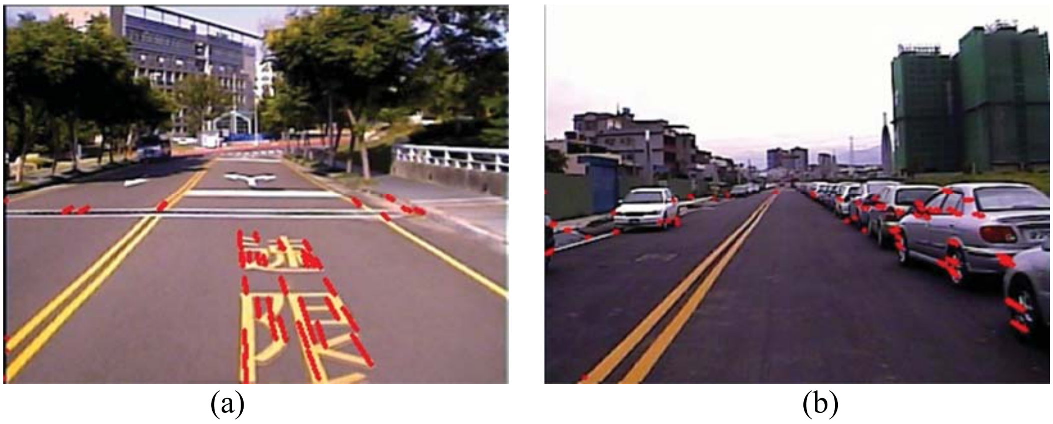


Figure 29: Results of corner detection and its corresponding optical flow

By considering above analysis of features, we proposed a feature point extraction method employ road detection procedure to assist in getting ground features. The flowchart of proposed feature point extraction is shown in Figure 30. The objective is to distinguish the major road region and non-road region, and utilize the result of the road detection, that is to extract the boundary of major road and some good features within road region. By integrating road detection, the more useful ground features could be extracted and could improve results of ground movement effectively. The next chapter will introduce the detail of road detection and describe what feature point will be selected.

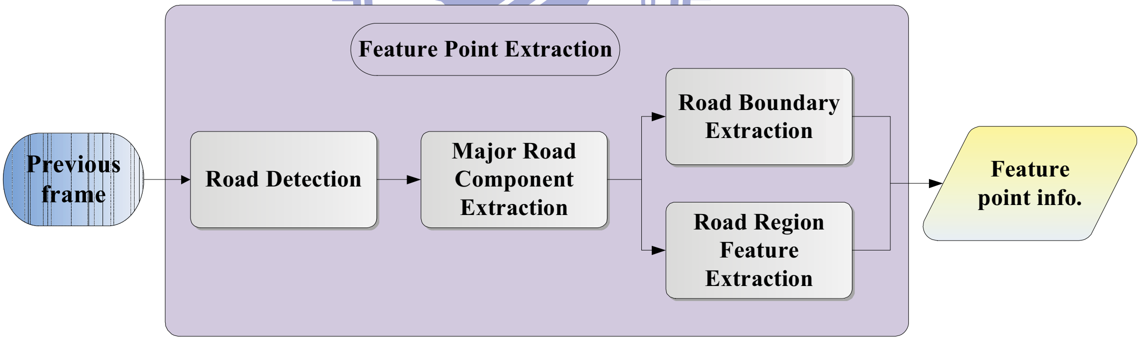


Figure 30: Block diagram of feature point extraction

4.2.4 Road Detection

The proposed feature point extraction technique is integrating a road detection procedure which is used an on-line color model that we can train an adaptive color model to fit road color. The main objective of road detection is to discriminate the road and non-road region roughly, because the result is used to support feature extraction not used to extract obstacle regions. However, we adopt an on-line learning model that allows continuously update during driving, through the training method that can enhance plasticity and ensure the feature is on the road region.

Due to the color appearance in the driving environment, we have to select the color features and using these color features to build a color model of the road. Therefore, we have to choose a color space which has uniform, little correlation, concentrated properties in order to increase the accuracy of the model. In computer color vision, all visible colors are represented by vectors in a three-dimensional color space. Among all the common color spaces, RGB color space is the most common color feature selected because it is the initial format of the captured image without any distortion. However, the RGB color feature is high correlative, and the similar colors spread extensively in the color space. As a result, it is difficult to evaluate the similarity of two color from their 1-norm or Euclidean distance in the color space.

The other standard color space HSV is supposed to be closer to the way of human color perception. Both HSV and L*a*b* resist to the interference of illumination variation such as the shadow when modeling the road area. However, the performance of HSV model is not as good as L*a*b* model because the road color cause the HSV model not uniform that lead to the HSV color model not as uniform as the L*a*b* color model. There are many reasons attribute this result. Firstly, HSV is very sensitive and unstable when lightness is low. Furthermore, the Hue is computed by dividing (I_{max} - I_{min}) in which I_{max} = max(R,G,B), I_{min} = min(R,G,B), therefore when a pixel has a similar value of Red, Green and Blue components, the Hue of the pixel may be undetermined. Unfortunately, most of the road surface is in similar gray colors with very close R, G, and B values. If using HSV color space to build road color model, the sensitive variation and fluctuation of Hue will generate inconsistent road colors and decrease the accuracy and effectiveness. L*a*b* color space is based on data-driven human perception research that assumes the human visual system owing to its uniform, little correlation, concentrate characteristics are ideally developed for processing natural scenes and is popular for color-processed rendering. L*a*b* color space also possesses these characteristics to satisfy our requirement. It maps similar colors to the reference color with about the same differences by Euclidean distances measure and demonstrates more concentrated color distribution than others. Then considering the advantaged properties of L*a*b* for general road environment, the L*a*b* color space for road detection is adopted.

The RGB-L*a*b* conversion is described as follow equations:

1. RGB-XYZ conversion:

$$\begin{aligned} X &= 0.431 \cdot R + 0.342 \cdot G + 0.178 \cdot B \\ Y &= 0.222 \cdot R + 0.707 \cdot G + 0.071 \cdot B \\ Z &= 0.020 \cdot R + 0.130 \cdot G + 0.939 \cdot B \end{aligned} \tag{4.17}$$

2. Cube-root transformation:

$$\begin{cases} L^* = 166 \cdot \left(\dfrac{Y}{Y_n}\right)^{\frac{1}{3}} - 16 & if \ \dfrac{Y}{Y_n} > 0.008856 \\ L^* = 903.3 \cdot \left(\dfrac{Y}{Y_n}\right)^{\frac{1}{3}} & if \ \dfrac{Y}{Y_n} \leq 0.008856 \end{cases}$$

$$a^* = 500 \cdot [f(\frac{X}{X_n}) - f(\frac{Y}{Y_n})]$$

$$b^* = 200 \cdot [f(\frac{Y}{Y_n}) - f(\frac{Z}{Z_n})]$$

$$where \ \mathrm{X}_n, Y_n, Z_n, are \ \mathrm{XYZ} \ tristimulus \ values \ of \ reference \ white \ point$$

$$X_n = 95.05, Y_n = 100, Z_n = 108.88$$

$$f(x) = \begin{cases} t^{\frac{1}{3}} & \dfrac{Y}{Y_n} > 0.008856 \\ 7.787t + 16/116 & \dfrac{Y}{Y_n} \leq 0.008856 \end{cases} \tag{4.18}$$

By modeling and updating of the L*a*b* color model, the built road color model can be used to extract the road region. The L*a*b* model is constituted of K color balls, and each color ball m_i is formed by a center on $(L_{m_i}, {}^*a_{m_i}, {}^*b_{m_i})$ and a fixed radius $\lambda_{\max} = 5$ as seen in Figure 31. In order to train a color model, we set a fixed area of the lower part of the image and assume pixels in the area are the road samples. For each of these pixels in the beginning 30 frames are used to initialize the color model, and updating the model every ten frames to increase processing speed but still maintain high accurate performance.

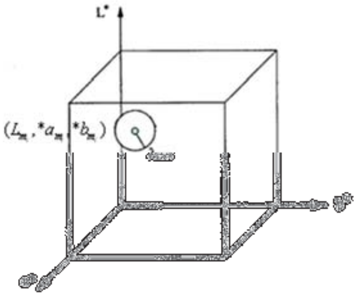


Figure 31: A color ball i in the L*a*b* color model whose center is at (L_m, *a_m, *b_m) and with radius $\lambda_{\max}$

The sampling area is used to be modeled by a group of K weighted color balls. We denote the weight and the counter of the $\boldsymbol{m_{i\,th}}$ color ball at a time instant $\boldsymbol{t}$ by $W_{m_i,t}$ and $Counter_{m_i,t}$, and the weight of each color ball represents the stability of the color. The color ball which more on-line samples belonged to over time accumulated a bigger weight value shown in Figure 32. Adopting the weight module increases robustness of the model.

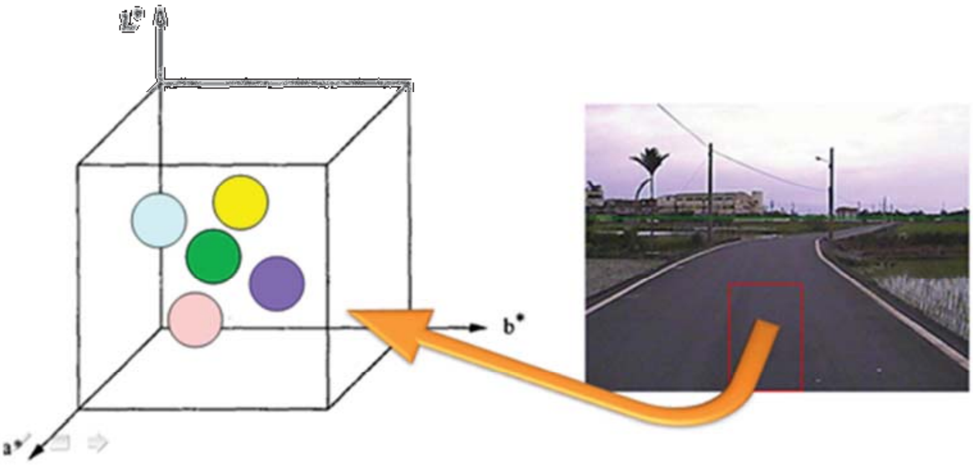


Figure 32: Sampling area and color ball with a weight which represents the similarity to current road color.

The weight of each color ball is updated by its counter when the new sample is coming which is called one iteration. Therefore the counter would be initialized to zero at the beginning of iteration. The counter of each color ball records the number of pixels added from the on-line samples in the iteration. The first thing to do is that which color ball is chosen to be added. We measure the similarity between new pixel x_t and the existing K color balls using a Euclidean distance measure Eq. (4.19). The maximum value of K is 50 which represents each on-lined model contains 50 color balls at most.

$$Similarity(x,m_i)=sqrt\left[(L_{m_i}-L_x)^2+(a_{m_i}-a_x)^2+(b_{m_i}-b_x)^2\right]<=\lambda_{\max} \qquad (4.19)$$

If a new pixel x_t was covered by any of the color ball in the model, one will be added to the counter of best matching color at this iteration as the Eq. (4.20). After entire new sample pixels at this iteration undertake the matching procedures mentioned above, the weights of every color ball are updated according to their current counter and their weight at last iteration. The updating method is as follows:

$$m(x_i)=arg_{m_i}min(Similarity(x_i,m_i)<=\lambda_{max}) \qquad (4.20)$$

$$W_{m_i,t+1}=\alpha_w\cdot W_t+(1-\alpha_w)\cdot counter_{m_i}/\left|N_{sample}\right|$$

$$\alpha_w\in[0,1],\quad \left|N_{sample}\right|$$

, where α_w is user-defined learning rate, N_{sample} is the sampling area

Then using the weight to decide which color ball of the model most adapt and resemble current road. The color balls are sorted in a decreasing order according to their weights. As a result, the most probable road color features are at the top of the list. The first B color balls are selected to be enabled as standard color for road detection, and these color balls with a higher weight has more importance in detection step. Road detection is achieved via comparison of the new pixel x_t with the existing B standard color balls selected at the previous instant of time shown in Figure 33. If no match is found, the pixel x_t is considered as non-road. On the contrary, the pixel x_t is detected as road.

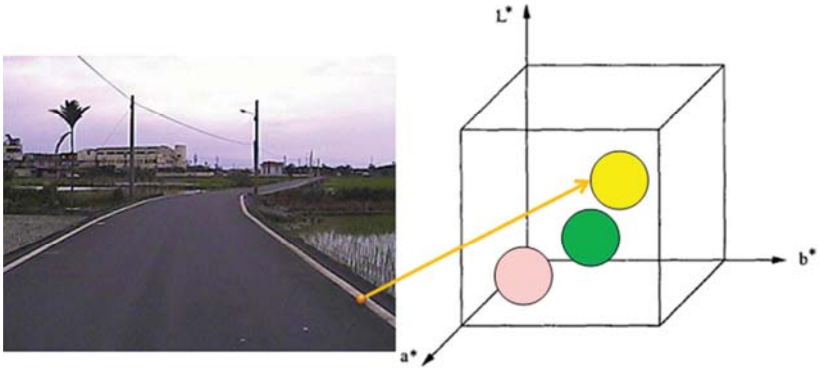
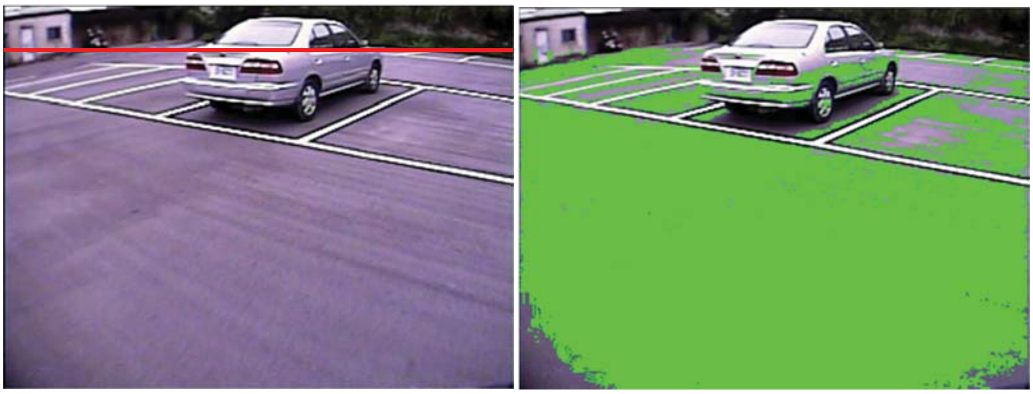


Figure 33: Pixel matched with first B weight color balls which are the most represent standard color.

Figure 34 shows some results of on-line L*a*b* road detection, the green areas are determined as road else are the non-road regions. However, we will not undertake road detection to all of the image because of the following procedure is processed in the world coordinate, that is the range of road detection is restricted to the horizon which is caused by the geometrical characteristic of IPM, and could be the different position in the image due to camera set up environment (camera height, tilt angle and so on).



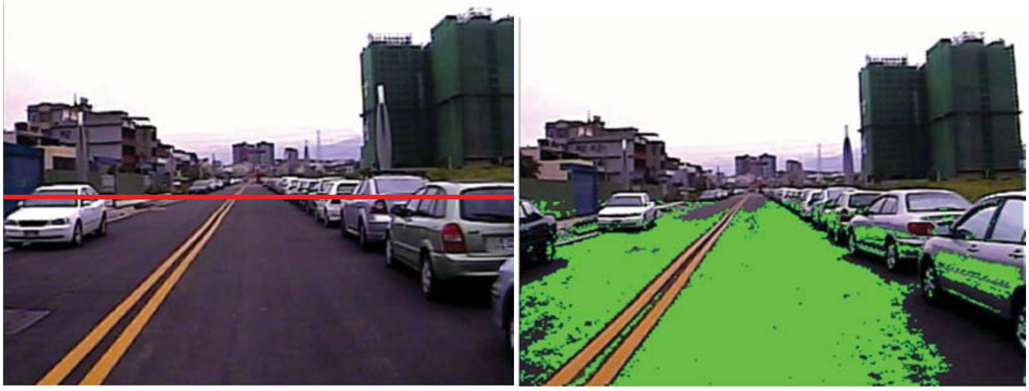


Figure 34: Results of road detection

The objective of road detection is to distinguish the major road and non-road region, and the result will be used to extract feature point. As above mentioned, we consider the two characteristics that are strong derivative and ground feature, the road boundary and strong gradient points are selected to be feature points.

Therefore, the first step of feature point extraction is to extract the major road region. When result of road detection is obtained, the dilation and erosion procedure is used to merge the neighboring region and to reduce the fragmentation then the connected component is executed to separate the road regions to several components. After that we will find out the maximum component of all components and assumed that to be the major road region. Then the boundary of major road region is extracted to be feature points, because the border of road and non-road should be the road feature and have strong derivative. Besides, we analyze the gradient distribution of major road area, and the more strong gradient points will be extracted to be feature points because of their strong derivative and position. Then the feature points are collected completely by these road boundary and high gradient features. The Figure 35 is shown some results of feature point extraction. By employing road detection to support feature point extraction, the more useful ground features can be extracted.

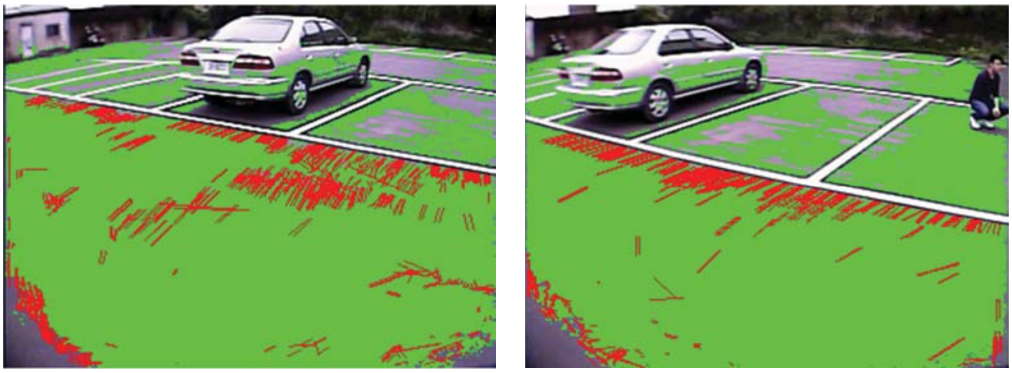


Figure 35: Results of feature point extraction. The upper image is result of road detection, and lower image is position of feature points.

4.2.5 Ground Movement Estimation

In this section we will introduce the proposed ground movement estimation procedure. Ground movement information is estimated from optical flow in the world coordinate system. By analyzing the principal distribution of optical flow can let us get the most representative ground movement, which will be used to compensate for the previous frame and difference with current frame. In addition, ground movement will be verified via temporal coherence. The flowchart of ground movement estimation is illustrated in Figure 36.

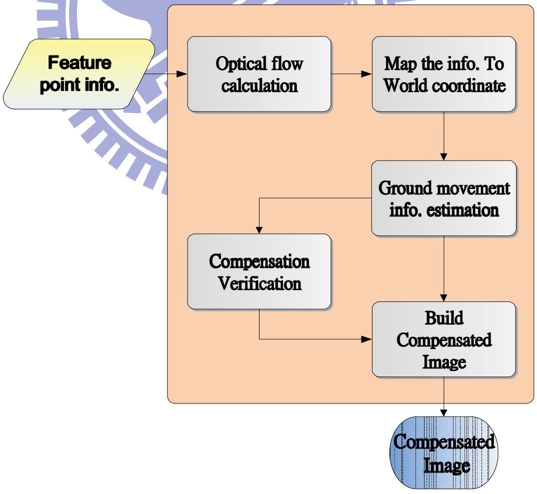
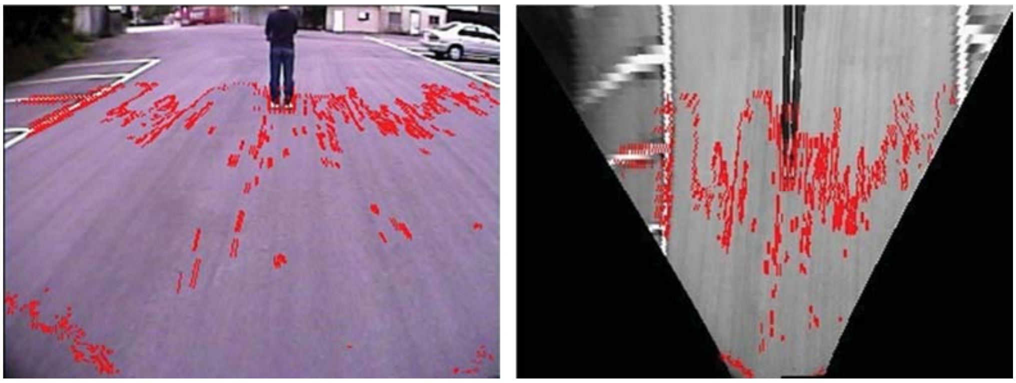

Feature point info.
Optical flow calculation
Map the info. To World coordinate
Ground movement info. estimation
Compensation Verification
Build Compensated Image
Compensated Image

Figure 36: Flowchart of ground movement estimation

1. Ground Motion Estimation

By feature point extraction procedure as described in upper section, the useful features for ground movement estimation are obtained. Then these feature points will be used to estimate ground motion. Therefore, when the feature points are acquired the main tasks of ground movement estimation procedure are high accuracy optical computation and ground movement information estimation. The first step is to calculate the optical flow for all of these feature points. The pyramidal Lucas and Kanade algorithm introduced, which copes efficiently with large movements, is used to calculate the optical flow for these feature points in the original image. As a result, Figure 37(a) and Figure 38(a) shows the feature points and its corresponding optical flow in the original image. Due to the perspective effect, the directions and lengths of the optical flows on the road in the original image are not the same when vehicle is moving straight shown in Figure 37(a). The case in which the vehicle is turning is shown in Figure 38. In this case the complicated optical flow distribution appeared in the original image. However, the inconsistent optical flow of road in the original image let us encounter a difficulty to estimate the ground movement. Therefore, we will take advantage of the IPM to remove the perspective effect. The optical flow information of an original image is mapped into world coordinate. The objective of inverse perspective mapping (IPM) is to remove the perspective effect by transforming the image coordinate to world coordinate, and scale the world coordinate that can obtain a bird's view image. Therefore, the world coordinate information is same as bird's view image that both of them are perspective removal. In our research, the IPM is used to remove the perspective effect and transform the image coordinate information to world coordinate and the ground movement procedure is processed in the world coordinate the bird's view image is used to display and examine some results, which we will not process on it.

The difference between the optical flow of the original image and that of the bird's view image is shown in Figure 37 and Figure 38. When a vehicle is moving straight, the optical flows in the bird's view image have the same direction and length independent of the locations of the optical flows. Similarly when vehicle is turning, the optical flow distribution in the bird's view image draws concentric circle following the movement of the ground but roughly have a similar magnitude.



(a) (b)

Figure 37: Difference between optical flow of original image and those of bird's view image when vehicle is moving straight.
(a) Optical flows of original image,(b) Optical flows of bird's view image

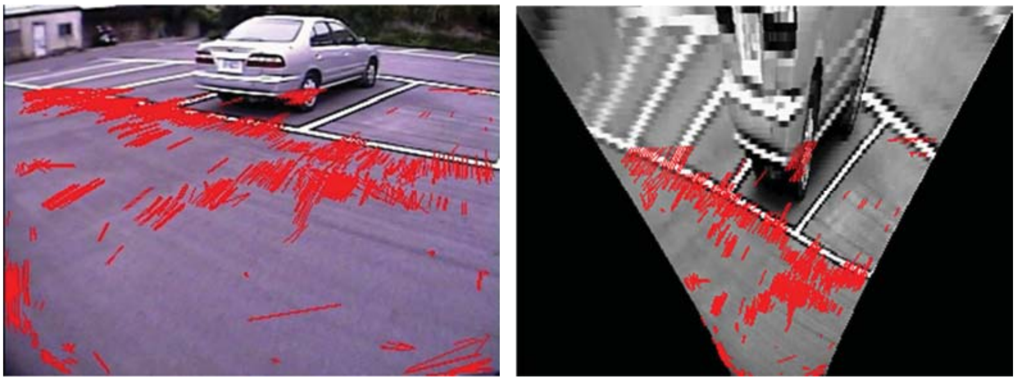


Figure 38: Difference between optical flow of original image and those of bird's view image when vehicle is turning.
(a) optical flows of original image, (b) optical flows of bird's view image

The kernel concept of ground compensation based detection algorithm is adopting the following characteristics. The optical flow distribution of ground region is approximately consistent in the bird's view image. On the contrary, a vertical straight line in the image which represented the vertical edge of obstacle in the world coordinate system is projected to a straight line whose prolongation will pass the camera vertical projection point on the world surface. Therefore, the optical flow distribution of obstacle regions are different drastic to the ground region. Then we can estimate the ground movement and used build a compensated image which assume the image is all planar object (ground). Therefore, the planar region will be eliminated but the obstacle regions will not. Figure 39 shows the difference between optical flow of obstacle and those of planar object.

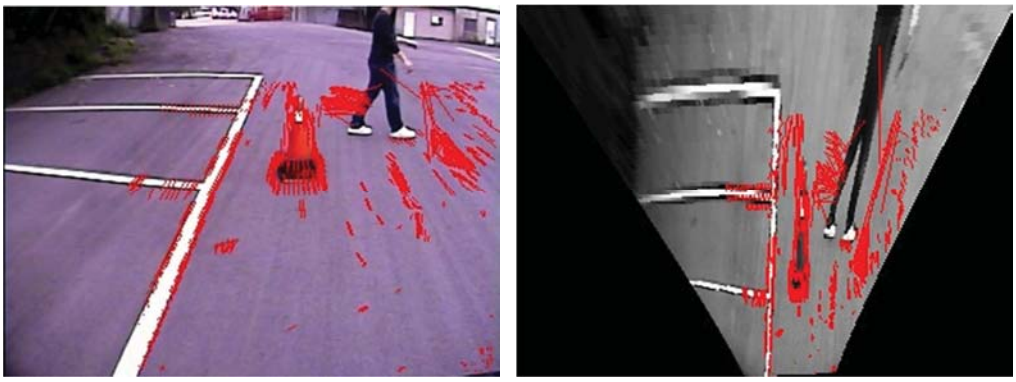


Figure 39: Differences between optical flows of obstacle and those of planar object.

Thanks to the mapping between original image and world coordinate system, ground movement information can be estimated based on the optical flow of feature points in the world coordinate system. The feature points which we obtained via integrating road detection are with a characteristic that most of these features will lie on ground region. Besides, the specific optical flow distribution of ground region in the world coordinate system, that is optical flow of them will approximately consistent. Due to the above characteristics, we would like to find out the principal distribution of optical flow that can let us get the most representative ground motion. We analyzing the principal distribution of optical flow by calculating the histogram of optical flow according to its direction and magnitude, and the peak of the histogram is considered to be ground motion. As shown in Figure 40, that is the histogram distribution of corresponding optical flow. By utilizing the principal motion, that can let us avoid some errors such as ambiguous optical flow or the non-peak value of optical flow is possibly causing by obstacle feature point.

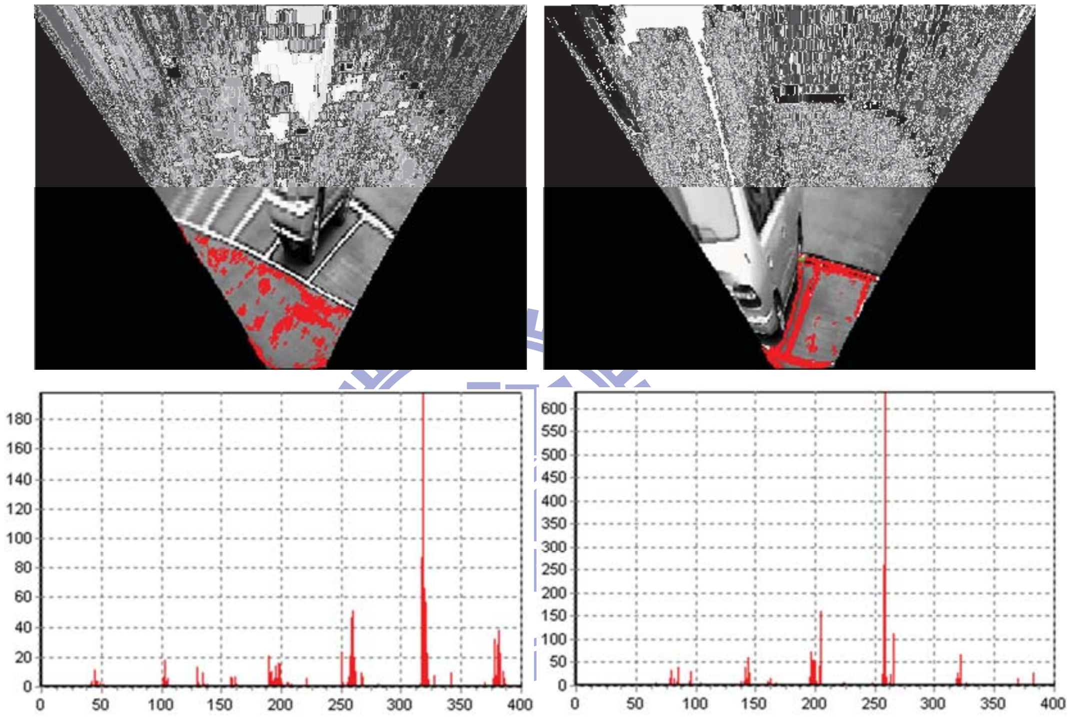


Figure 40: Histogram distribution of optical flow in world coordinate system

2. Compensated Image Building

The specific optical flow distribution in the world coordinate system shows that the movement of ground in the world coordinate system can be described as a translation or rotation of a two-dimensional coordinate plane. The ground movement information is estimated by using optical flow in world coordinate system, and the optical flow is ground motion which is estimated by analyzing the principal distribution of optical flow of feature points, that is utilized the procedure as described in last section. Then we can acquire ground motion in the world coordinate system which is used to estimate ground movement information.

In the world coordinate system converted from consecutive images captured at difference location O1 and O2 as depicted in Figure 41, if we represent the world coordinates of a ground feature point P as $(x_1, y_1)^T$ and $(x_2, y_2)^T$ before and after vehicle movement respectively, the optical flow of the feature point P can be written as Eq. (4.21).

$$\begin{pmatrix} f_x \\ f_y \end{pmatrix} = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} - \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \tag{4.21}$$

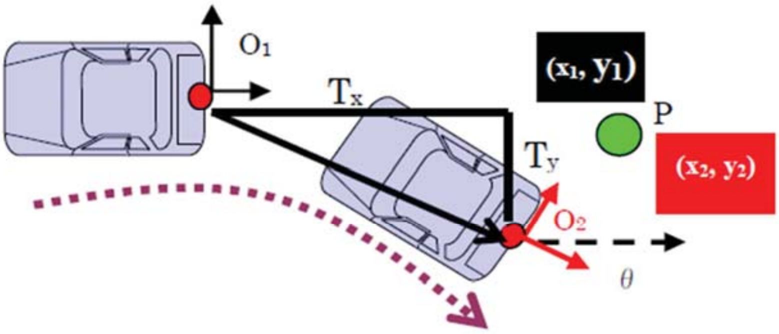


Figure 41: Two-dimensional coordinate plane

If we represent the ground movement information between consecutive world coordinate system by a two-dimensional coordinate plane as shown in Figure 41, the relationship of corresponding points between consecutive world coordinate system can be described with the Eq. (4.22), where Θ is the rotation component of vehicle movement, and $(T_x, T_y)^T$ are the translation components of ground movement. Then ground movement information including rotation and translation component. That is we can utilize Eq. (4.22) to find out the compensated coordinate of point $(x_1,y_1)^T$ but Θ and $(T_x, T_y)^T$ should be calculated in advance. By substituting Eq. (4.21) into Eq. (4.22), then the relationship between Θ and $(T_x, T_y)^T$ can be derived as Eq. (4.23).

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \begin{pmatrix} T_x \\ T_y \end{pmatrix} \tag{4.22}$$

$$\theta \begin{pmatrix} y_1 \\ -x_1 \end{pmatrix} - \begin{pmatrix} T_x \\ T_y \end{pmatrix} + \begin{pmatrix} f_x \\ f_y \end{pmatrix} = 0 \tag{4.23}$$

Where Θ and $(T_x, T_y)^T$ are unknown parameters, $(f_x,f_y)^T$ and $(y_1,-x_1)^T$ can be obtained during the process of optical flow calculation. Therefore, when obtaining the ground motion we can use these ground point information including world coordinate and magnitude of optical flow to calculate the ground movement information, that is unknown parameters Θ and $(T_x, T_y)^T$ can be acquired.

By undertaking above procedure, the ground movement information can be obtained and used to interpolate image, then compensated image is generated. Therefore, thanks to the mapping between image coordinate and world coordinate, we will compensate previous frame image to a new image by projecting the image coordinates to the ground plane and compensating the ground movement on the ground plane then back projecting to the image plane again. Figure 42 illustrates the procedure of the compensated image building. For each pixel of the previous frame will be transformed to ground plane which is world coordinate by IPM forward mapping, then for each world coordinate is compensated to new world coordinate by using ground movement information and Eq. (4.22), the coordinates are back transforming to image coordinate by IPM backward mapping, a new image is interpolated in the image plane. Therefore, the new built image is assuming formed by planar object because of compensated using ground movement. Due to these assume the non-planar object (obstacle) can be extracted by comparing current image and compensated image.

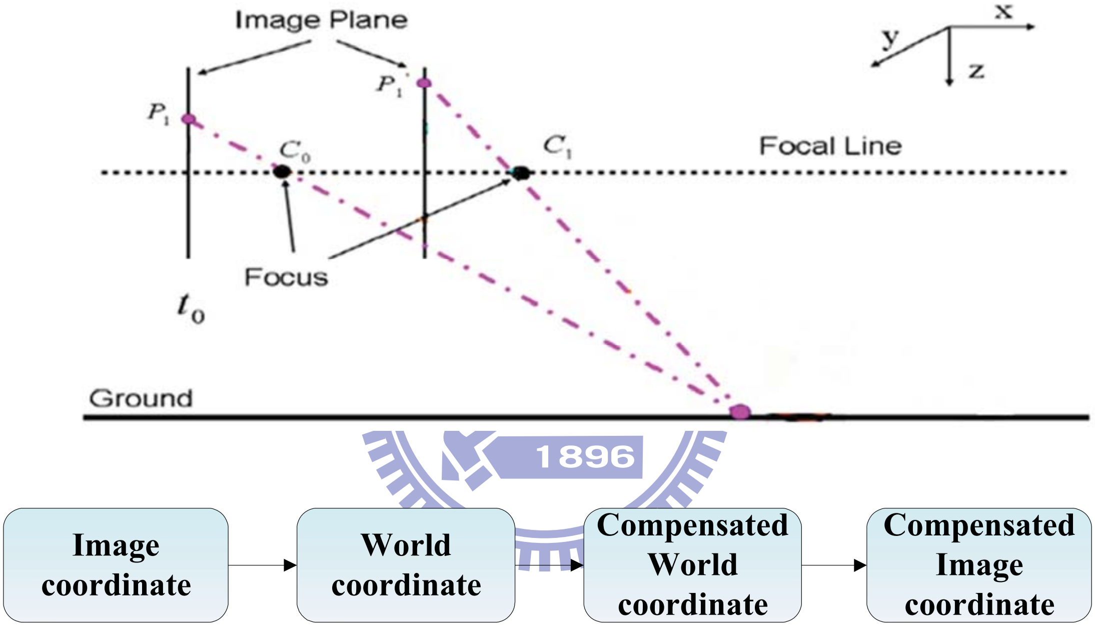


Figure 42: Procedure of the compensated image building

3. Compensation Verification

The ground movement information is used to compensate the image and a new image is built. By considering the temporal coherence, that is to consider the condition that the scene will not change a lot during the few frames. For these reason, when the ground movement information is obtained, the compensated image will be built and used to compare with current frame image then we can acquire an obstacle candidate image which indicate the pixel is obstacle or not. The detail of how to generate obstacle candidate image will be introduced in latter section. That is number of obstacle candidate should be stable because of little varied scene during few frames. Therefore, number of obstacle candidate is anomalous relative to the neighboring frames would indicate the compensation information is not correct.

Observing the recent frames can assist us to verify and ensure the ground movement information is correct. Because the correctness of ground movement is greatly determining the results of following detection procedure, the verification is essential and worth to undertake. As shown in Figure 43, number of obstacle candidate in previous ten frames will be record, and the mean of these is calculated. The Eq. (4.24) is used to check correctness of ground movement. If the equation is conformed, that is indicating anomalous amount of obstacle candidate and the ground movement information in the current frame possibly erroneous. Then the previous compensation information will be utilized to compensate.

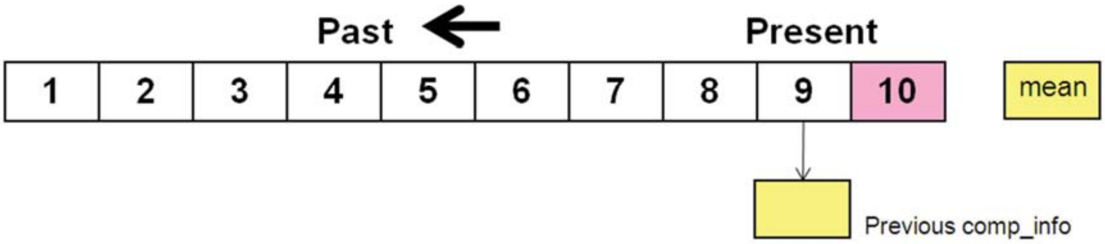


Figure 43: Chart of temporal coherence

Quantitative criterion:

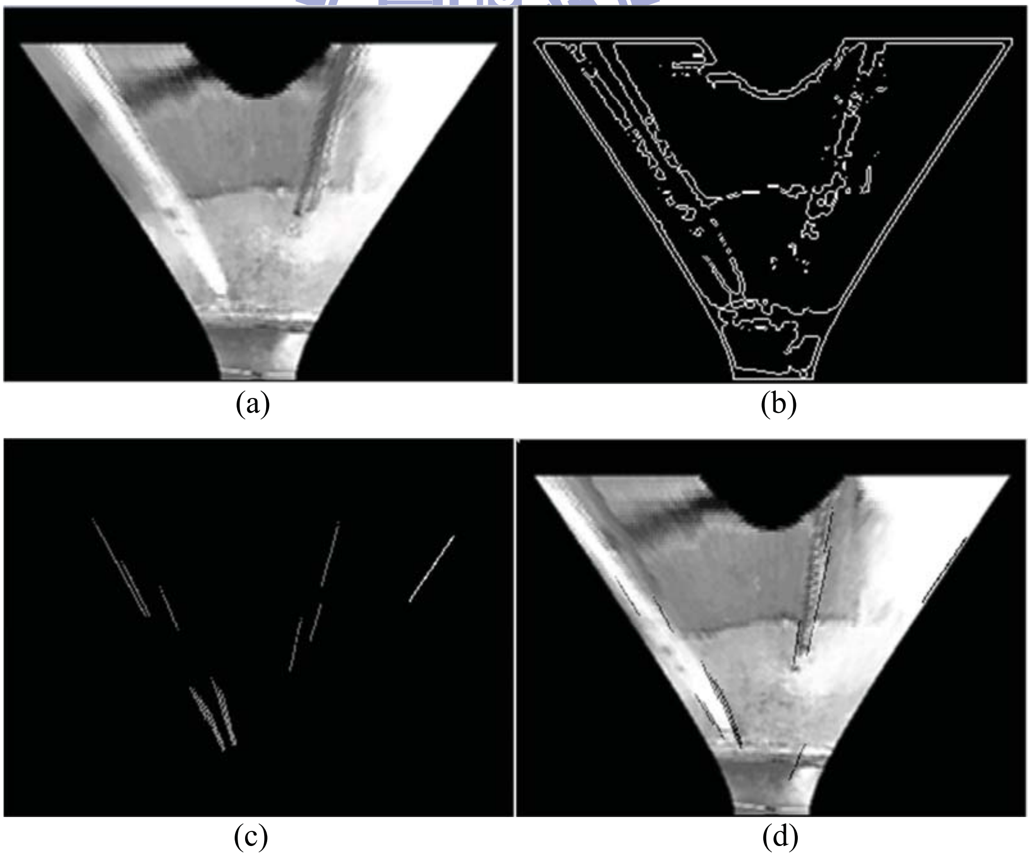
$$\begin{aligned} &if\ \ obs\_num\ > High\_Thd * mean \\ &\qquad\qquad\quad < Low\_Thd * mean \\ \\ &\Rightarrow comp\_info = previous\ comp\_info \end{aligned} \tag{4.24}$$

,where High_Thd is user-defined threshold about 1.3, Low_Thd is about 0.8

By using temporal coherence, the average amount of obstacle candidate in recent frames is calculated and could be validate the compensation information.

4.2.6 Obstacle Feature Searching algorithm

As what we have mentioned previously, we only prefer to search the features extracted from the objects with quasi-vertical edges in the remapped image. Based on the observation that those qualified features will always pass through the vertical projection point of cameras, we propose a feature searching algorithm and use polar histogram to accurately detect obstacles even for the noisy images. Our searching algorithm begins with the vertical projection point of a camera in the remapped image (denoted as CP). After that, we scan the acquired profile or temporal difference image angle by angle from the center to outmost border of a circle in the defined radius which can be determined by the information of remapped images. We then use a voting method in the mask searching and adjust the searching space flexibly according to the intensities and relative distances between vehicles and obstacles. The voting threshold is fixed and can be determined by the half of the total elements in the mask. We can keep the major features, for the Gaussian weighting values in each 5x5 mask indicates the important regions in this mask. For each angle, a feature segment will be taken only if its corresponding percentage is higher than some specific value in order to reduce the errors caused by statistics. The next searching point at the same angle must be close enough to the last searched segment so as to concentrate on the points close to CP. After the searching process at each angle, the number of points at each angle will be used to produce a polar histogram in our system. Some results in the feature searching procedure are shown in Figure 44 and Figure 45, and the complete flowchart of our feature searching algorithm is shown in Figure 46. As Figure 46 shows, our algorithm can deal with two kinds of problems. One is that our systematic design can effectively improve the accuracy rate and reduce the possibility of lost pixels by using the features which are defined in each angle line by the results of mask searching to indicate the distance between objects and the camera. The other is that our proposed method can discriminate the meaningful pixels from others by the presented model for checking the length of searched feature segments. Therefore, the flaws of the polar histogram can be made up and our obstacle detection will make a great progress in performance.


(a) (b)

(c) (d)

Figure 44: The results in the feature searching procedure by using profile images

(a) the sharpened remapped image, (b) the profile image, (c) the scanned feature segments, and

(d) the scanned feature segments superposed on the sharpened remapped image.

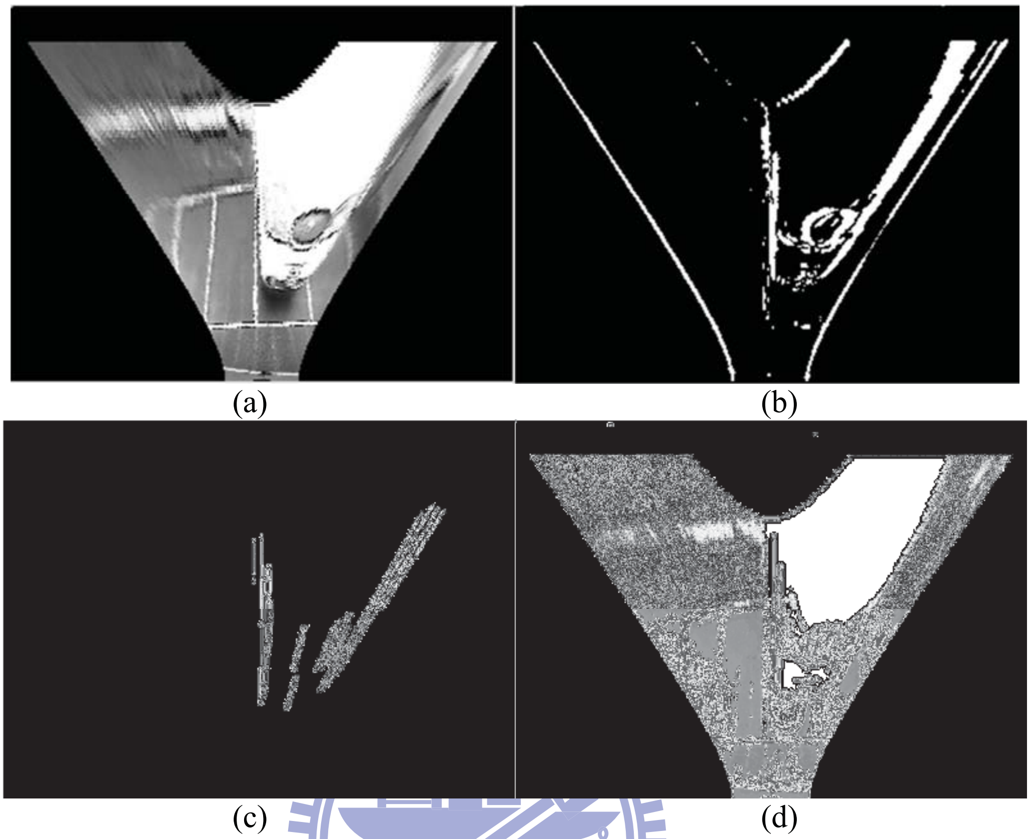


Figure 45: The results of the feature searching procedure using temporal difference FLIPM images

(a) the remapped image, (b) the temporal difference FLIPM image, (c) the scanned feature segments, (d) the scanned feature segments superposed on the remapped image.

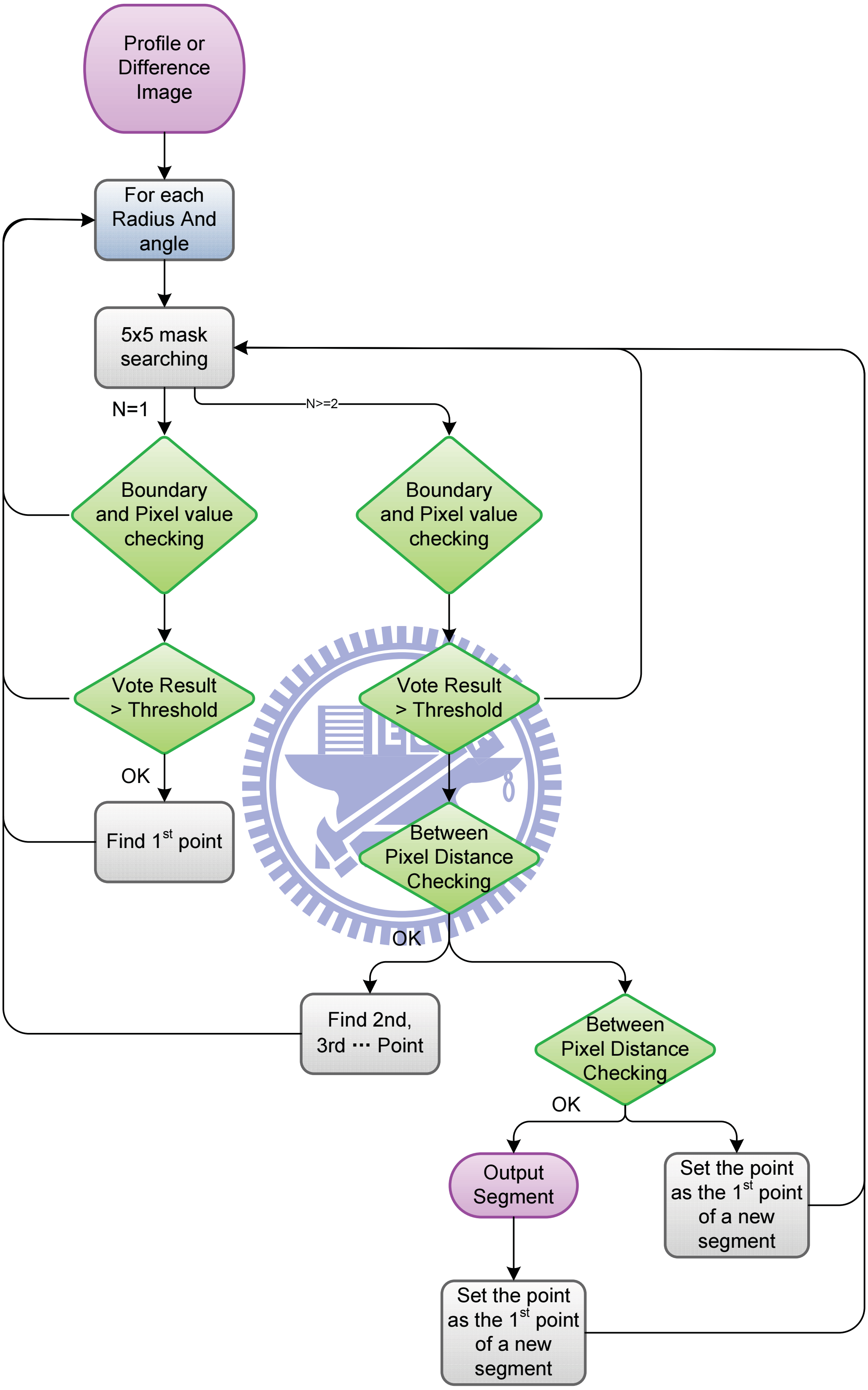

Profile or Difference Image
For each Radius And angle
5x5 mask searching
N=1
N>=2
Boundary and Pixel value checking
Boundary and Pixel value checking
Vote Result > Threshold
Vote Result > Threshold
OK
Find 1st point
Between Pixel Distance Checking
OK
Find 2nd, 3rd ··· Point
Between Pixel Distance Checking
OK
Output Segment
Set the point as the 1st point of a new segment
Set the point as the 1st point of a new segment

Figure 46: The flowchart of feature searching

4.2.7 Histogram Post-processing

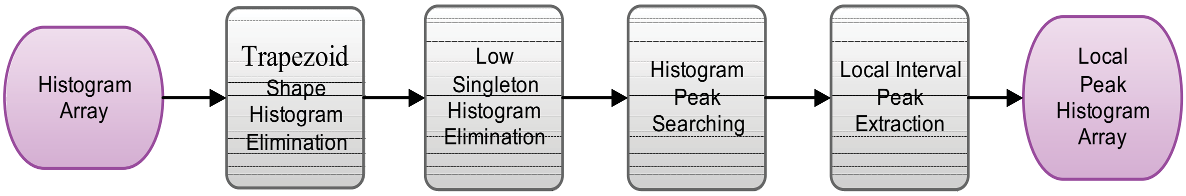
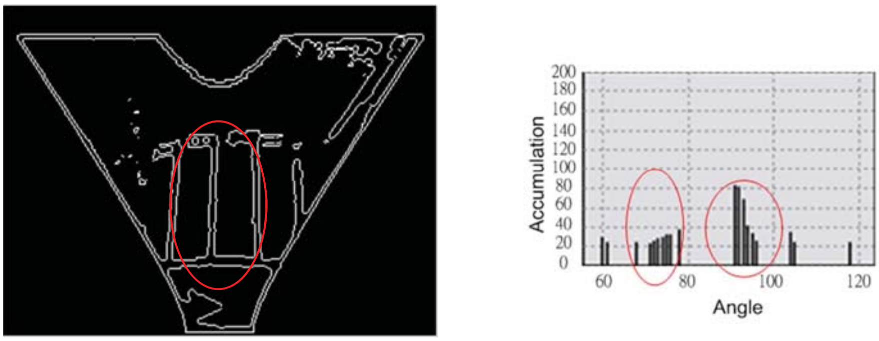

Histogram Array
Trapezoid Shape Histogram Elimination
Low Singleton Histogram Elimination
Histogram Peak Searching
Local Interval Peak Extraction
Local Peak Histogram Array

Figure 47: The block diagram of histogram post-processing

Figure 47 shows the processes in the histogram post-processing. The post-processing on histograms is necessary since we have to consider some important problems such as how to obtain the peak values of the histogram, how to reduce the influences of light, and how to find the best way in statistics for our applications. After we obtain the polar histogram of feature segments, we still need to find our desired histograms to remove the segments of planar objects and noises. Our procedures in histogram post-processing try to reduce the undesired information which may be produced in the polar histogram. For instance, the line segments belonging to planar objects will still be extracted in the polar histogram step. By observing the polar histogram, we can discover that the trapezoid histogram represents the planar objects. We can thus remove those clusters of bins in the histogram and we do not process the over-small histograms (the columns in the histogram are few) to avoid disturbances. After eliminating planar objects and noise, we will search the position of local maximum which represents the segment position of non-planar object in the polar histogram. Also, we only pick a peak column at an angular interval to prevent from detecting too many obstacles at the same time. Some results in the histogram post-processing procedure will be show in Figure 48 and Figure 49. As Figure 48 shows, the regions in red circles (Figure 48 (b)) are corresponding to the lane markings, as shown in Figure 48 (a). In Figure 48 (b), x-axis and y-axis represent the angles of polar histograms and the accumulation on each angle, respectively.



(a) (b)

Figure 48 : Illustrative figures of the trapezoid histogram distributions.

(a) The figure of lane markings, (b) The trapezoid histograms.

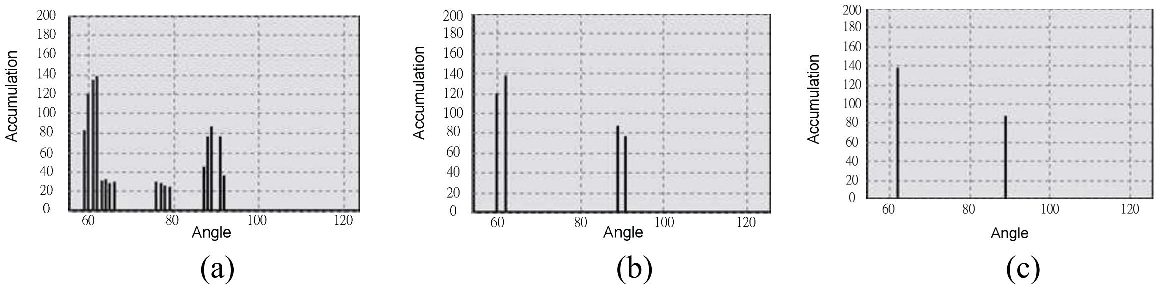


Figure 49: The processes of histogram post-processing. x axis means the polar histogram's angle and y axis means the accumulation on each angle.

(a) The polar histogram of Figure 45. (b) The histogram of (a) after the trapezoid histogram elimination, low singleton histogram elimination and the peak searching procedure. (c) Local peak histogram.

4.2.8 Object Tracking and Information Extraction

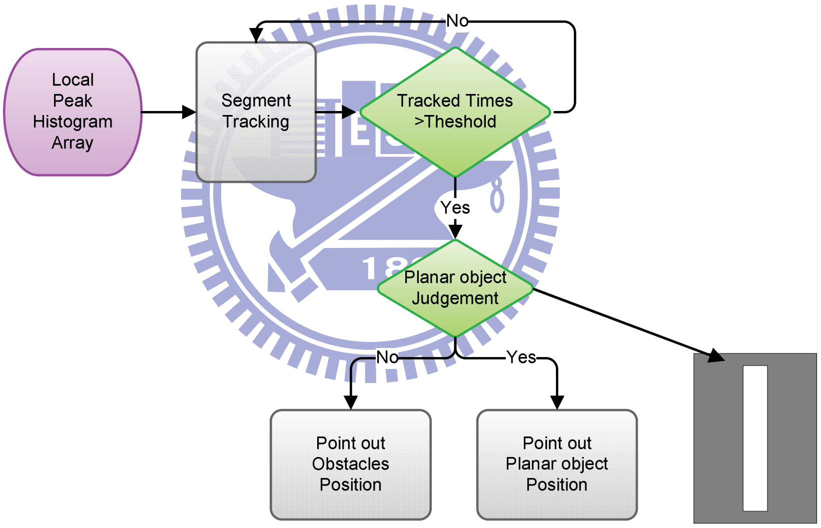

Local Peak Histogram Array
Segment Tracking
Tracked Times >Theshold
No
Yes
Planar object Judgement
No
Yes
Point out Obstacles Position
Point out Planar object Position

Figure 50 : The block diagram of object tracking and information extraction

Our tracking procedure is used to confirm the detected objects. We choose the displacement and variation of angles in the extracted feature segment as the judgment conditions in the tracking process. If the feature segment has been extracted, we would judge again whether this feature segment belongs to the planar object by a pattern matching approach. Our tracking process and the pattern for representing the planar object are shown in Figure 50. We can finally confirm that the detected feature segment is an obstacle and also obtain the position of feature segments or other useful information.

1. Obstacle Candidate Image

Due to the property of compensated image as described in the previous, that is the compensated image is interpolated with ground movement information so that the planar object such as road will be correspond to the same position with current image and the non-planar object will not. Therefore, these characteristic is utilized to detect the obstacle region via image difference as depicted in Figure 51 the image difference block. By image compare between current frame image and compensated image, the planar region will be eliminated and the obstacle region will be marked.

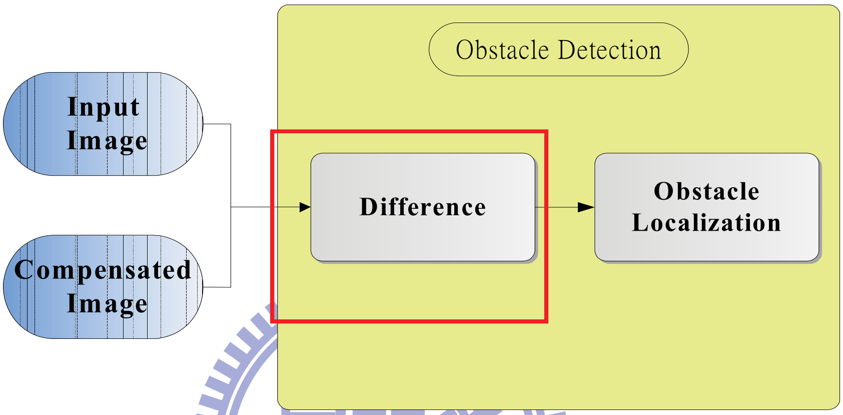


Figure 51: Flowchart of Obstacle Detection

Here, it can be determined whether a point is on the ground plane or not by comparing the gray values of the corresponding pixels on two image frames Eq. (4.25), where △ is the threshold for the maximum disparity of gray value of two corresponding points. In this way, we can detect obstacles above the ground plane. When image difference is completed, the obstacle candidate image is obtained which is used to indicate every pixel is belonging to obstacle or not. Figure 52 shows some results of image difference between current image and compensated image, the pixels which are marked as blue are determined as obstacle.

$$\left|cur\_image(P) - comp\_image(P)\right| = \begin{cases} \geq \Delta \quad , & \rightarrow \text{obstacle} \\ < \Delta \quad , & \rightarrow \text{ground} \end{cases} \tag{4.25}$$

,where cur_image is current frame image, and comp_image is compensated image

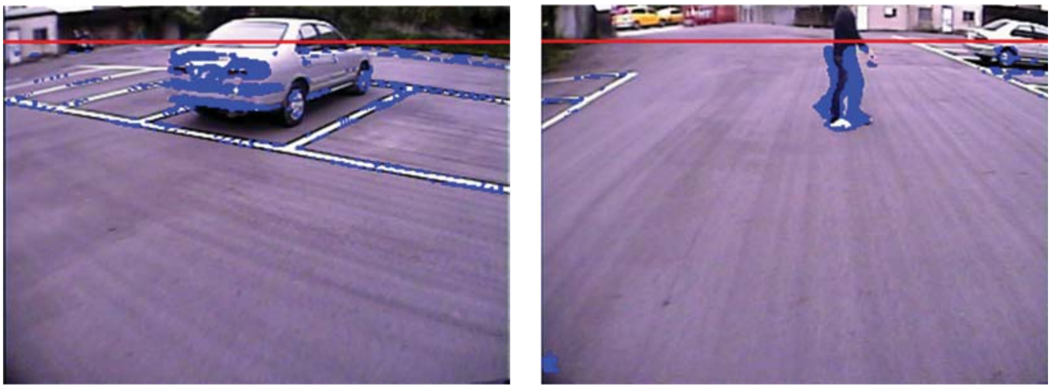


Figure 52: Results of image difference between current image and compensated image

2. Obstacle Localization

In this section, the objective of the obstacle localization procedure is to locate the position of each target objects. The obstacle candidate image is used to extract obstacle region and locate the position of obstacle. Figure 53 illustrates the flow of obstacle localization, obstacle candidate image is integrating vertically and for each obstacle we will locate the closest position to ego-vehicle. Owing to the distance between each obstacle and the ego-vehicle is the most important task for driver when backing up. Therefore, in our research the main purpose is to look for the closest point to the camera not to bind the entire obstacle region.

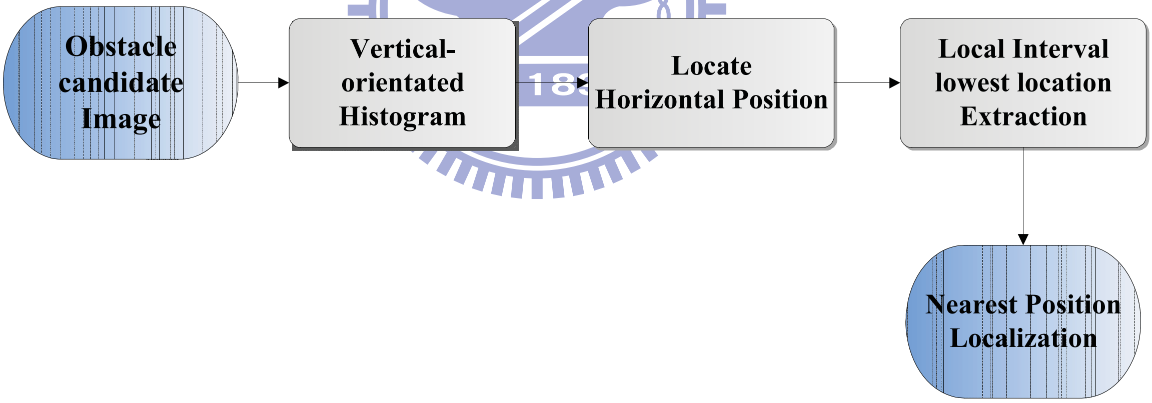

Obstacle candidate Image
Vertical-orientated Histogram
Locate Horizontal Position
Local Interval lowest location Extraction
Nearest Position Localization

Figure 53: Flowchart of obstacle localization

The first step of the obstacle localization is to calculate the vertical-orientated histogram. The vertical-orientated histogram of binary detection result(obstacle candidate image) is created to ensure horizontal positions of obstacle by extracting the bins which exceeding threshold. The idea is based on the fact that obstacle have more vertically oriented edges compared with their background. By integrating the detection result vertically, the detection on vertically oriented objects is amplified. The vertical-orientated histogram P is created with Eq. (4.26).

$$P_x = \sum_y I_{xy} \quad \forall x \tag{4.26}$$

Here, x is the horizontal coordinate of the image and y is the vertical coordinate of the image. Since the obstacle candidate image is binarized, the pixel intensity I either has a value 0 or 1. Figure 54 illustrates the obstacle candidate image and the corresponding vertical-orientated histogram. Then by using a threshold to indicate which horizontal position have significant accumulation that means obstacle could lie on these x coordinates.

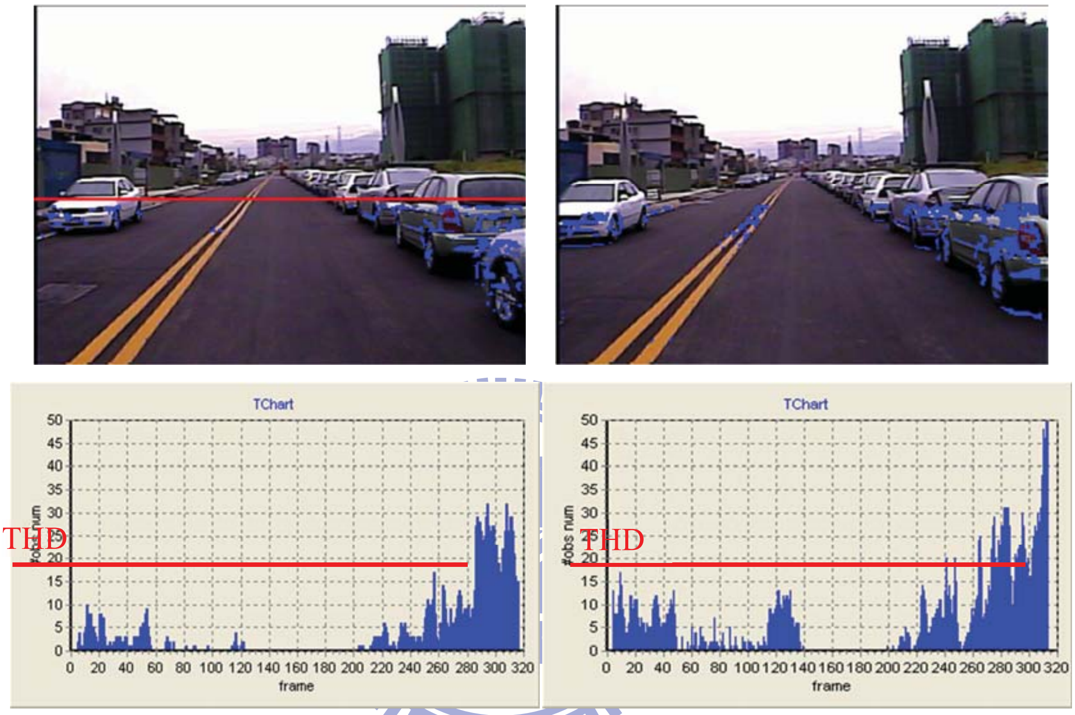


Figure 54: the obstacle candidate image and the corresponding vertical-orientated histogram

The detail of creating vertical-oriented histogram is depicted in Figure 55, first the image will be divided vertically to several bins, for each bin is to scan from bottom to top and to accumulate the number of obstacle candidate. If the interval between two candidates is smaller than a threshold, we consider that these still belong to the same object so that accumulate to the histogram in these bin; otherwise, clean up the accumulation. Besides, creating the vertical-oriented histogram we will save the lowest position of each bin simultaneously. Finally, by extracting lowest position of local interval, the nearest position to the ego-vehicle of each obstacle region can be obtained and the results can be refined. That is for all of bins of histogram which is over the threshold, in every local area that will reserve single position which is the lowest coordinate. For every obstacle region, the closest position to our ego-vehicle will be located and marked to alert the driver.

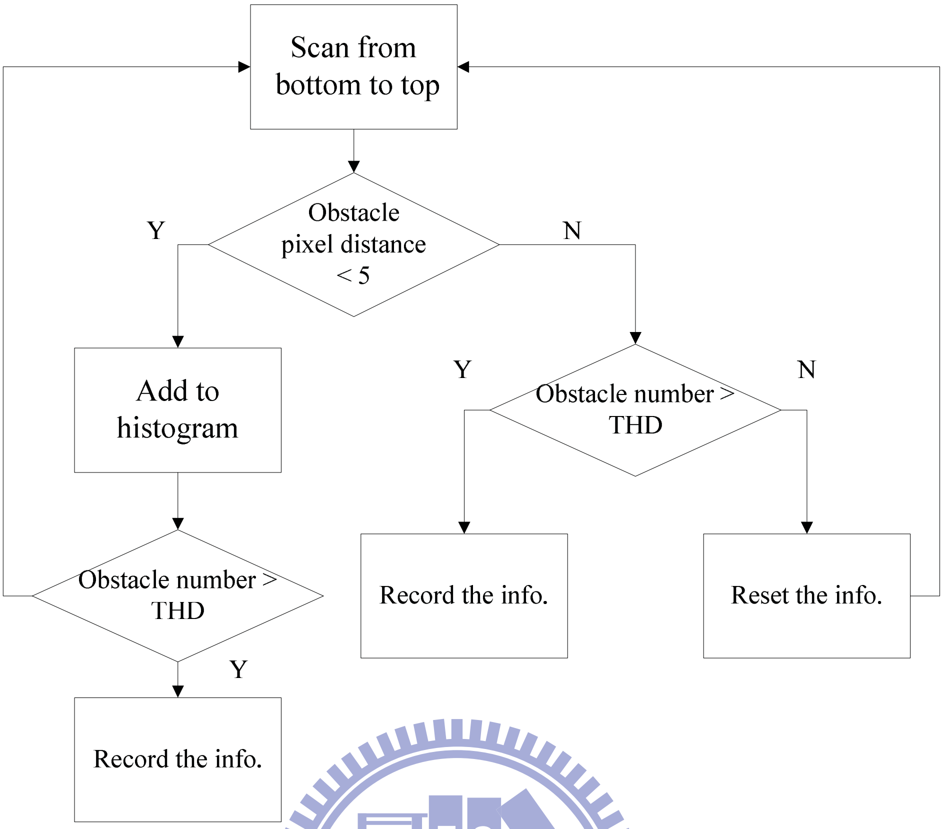


Figure 55: Procedure of creating vertical-orientated histogram

3. Obstacle Verification

To decrease amount of false alarming which would cause by erroneous ground movement, the false alarming have occurred commonly on road marking or shadows. Therefore, knowledge of road detection is used to verify results of obstacle detection. In other words, for all of potential target locations are verified using some prior knowledge about the road information. For all of potential target positions, verification is exceeded by checking road ratio of fixed bounding box. If road ratio of a bounding box is over than a threshold, that the position could be the road region, then we will filter out the result. On the contrary, road ratio of a bounding box is smaller than a threshold then the position is not considered to be a ground region. Therefore, we consider that the region is part of obstacle and reserve the result. Figure 56 is illustrated the idea of obstacle verification. By final checking of road ratio, some error such as road marking would be removed and the result of obstacle detection is more robust.

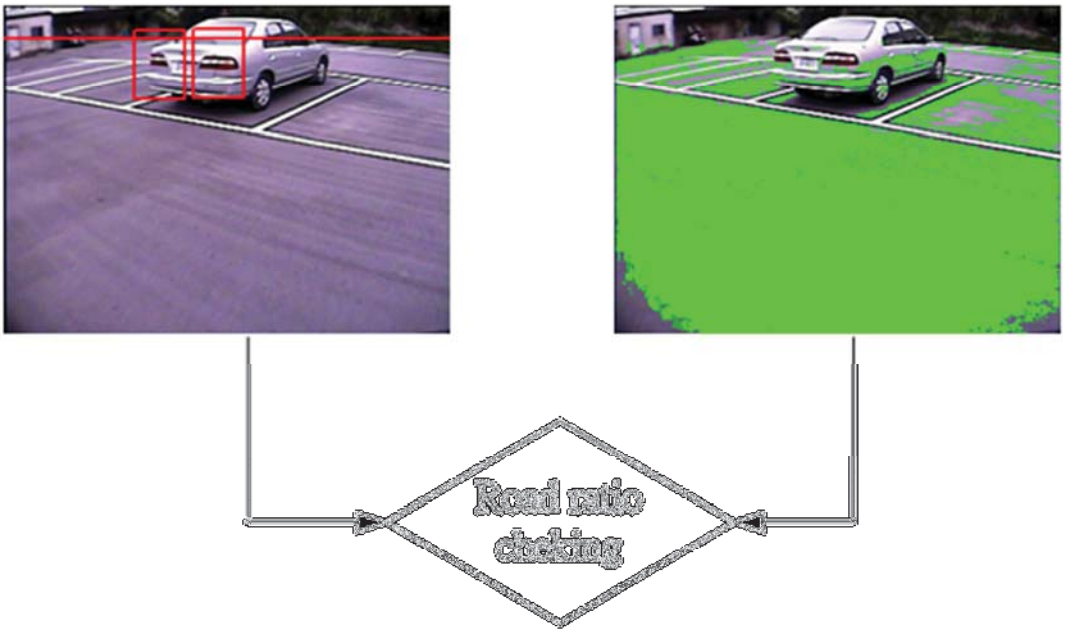


Figure 56: Procedure of obstacle verification

$$\textit{if} \quad road\_ratio > 60\% \quad => \quad \textit{filter out the result}$$

4. Distance Measurement

For backing up maneuver safety, the position of objects in rear view which are driver concerning with during parking period. Among of all obstacle information the distance between target and ego-vehicle is most essential. By acquiring the distance, we can realize objects are near or far away from our ego-vehicle. Then some backing-up crashes could be avoided. The distance which we want to know is the closest position of obstacle to our vehicle because of the nearest collision would occur here. Therefore, the distance between the lowest position of each obstacle which is located by the obstacle localization procedure and our vehicle will be estimated by our procedure.

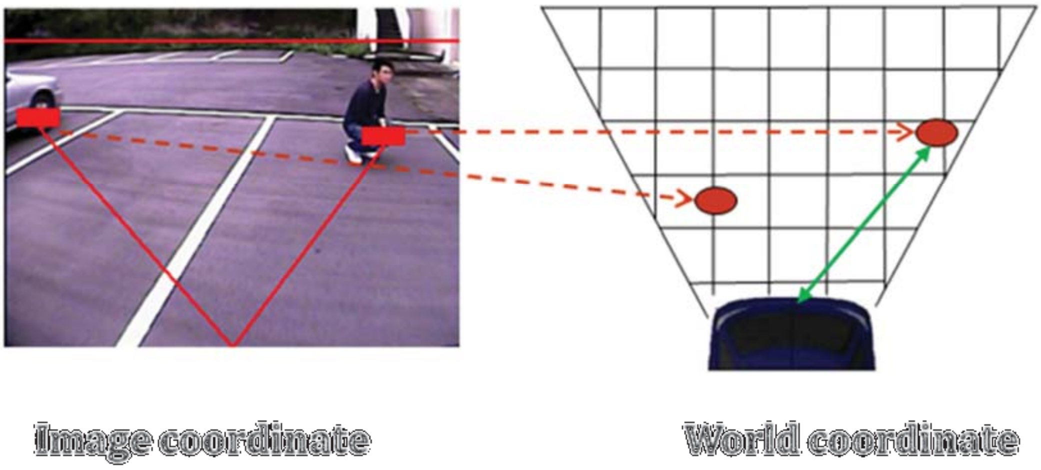


Figure 57: Transformation between image coordinate and world coordinate

As shown in Figure 57, all the obstacle positions can be transformed from the image coordinate to world coordinate by IPM calibration procedure. Therefore, the first step of distance measurement is to obtain the world coordinate of each target. Then the world coordinate will be transformed to real distance by scaling. In order to estimate the scale between real length and world coordinate, the horizontal and vertical reference line on the ground should be measured as depicted in Figure 58. By calculating the proportion between reference line in real scene and in the world coordinate, the scale of horizontal and vertical will be acquired to undertake the mapping between world coordinate and real length. The Figure 59 is illustrated the block diagram of distance measurement, that transforms the image coordinate of target to world coordinate first, then mapping the world coordinate to real distance. The distance how far away our vehicle is obtained.

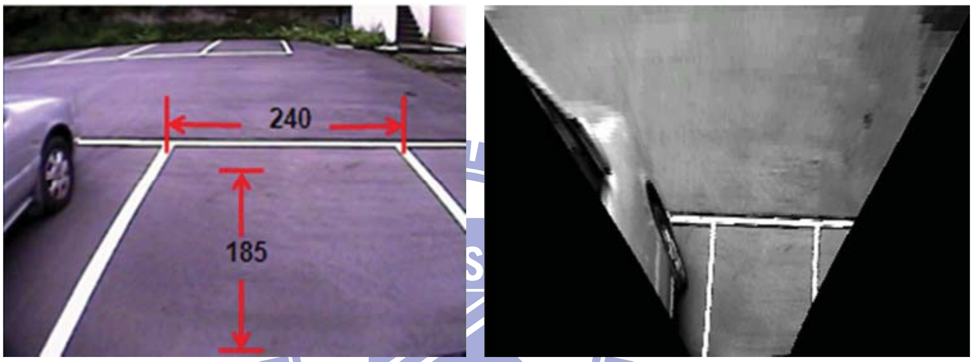


Figure 58: Scale measure between world coordinate and real length

Image coordinate
World coordinate
Distance

Figure 59: Block diagram of distance measurement

4.3 Dynamic Distance Gauge (DDG)

This work proposes a novel, real-time digital parking assist system to dynamically gauge the motion direction and distance from image sequences captured by rear view camera of a car and produce assist-cues in the monitor. This technique applies image processing methods to extract motion vectors from image sequence and use a weight-based motion combination formula to evaluate the turning degree of a car. This work introduces a novel application in the field of image processing. Most previous parking assist systems require physical information from the steering wheel of a car; however, the proposed system requires only camera information. Hence, the proposed system can be applied simply to any vehicle with a backward camera. It has been implemented and tested on both PC-based and real-time embedded systems.

4.3.1 System architecture of the DDG

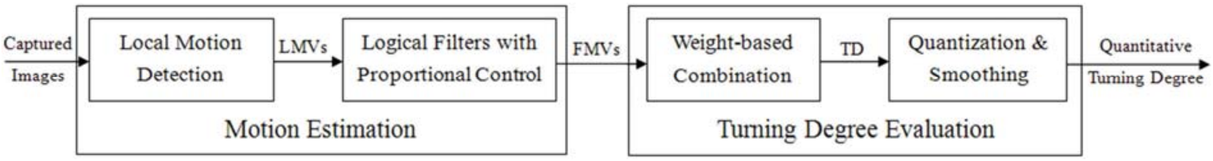


Figure 60: System architecture of the proposed DDG algorithm

As shown in the Figure 60, the system architecture of the proposed algorithm consists of two processing units: the motion estimation unit and turning degree evaluation. In the motion estimation unit, local motion vectors (LMVs) are detected by the block matching method with three pre-defined detection areas which are suitable for further evaluation of turning degree. Logical filters with a proportional factor are applied to filter out sudden noises caused by moving objects and generate filtered motion vectors (FMVs). Following the motion estimation, a weight-based combination formula is proposed to combine FMVs to estimate the turning degree (TD). Furthermore, some post-processing methods are employed to quantize and smooth the evaluated TD for diver assistance systems to plot driver cues in the output images.

1. Motion Estimation

The motion estimation unit presented in Figure 59 contains motion detectors and logical filters for pre-defined detection areas. Three detection areas are selected as shown in the Figure 61. Area1 is located in the top-center of the image, and it is the primary area for turing evaluation. The horizontal motion of area1 defines the direction of turing in the following turning degree evaluation. Area2 and area3 are located in the top-left and top-right respectively.

Motion vectors evaluated from area2 and area3 are used to calculate the difference of side-velocities of a car and evaluate the turning degree. LMVs corresponding to every detection areas are obtained by block matching (BM) algorithm.

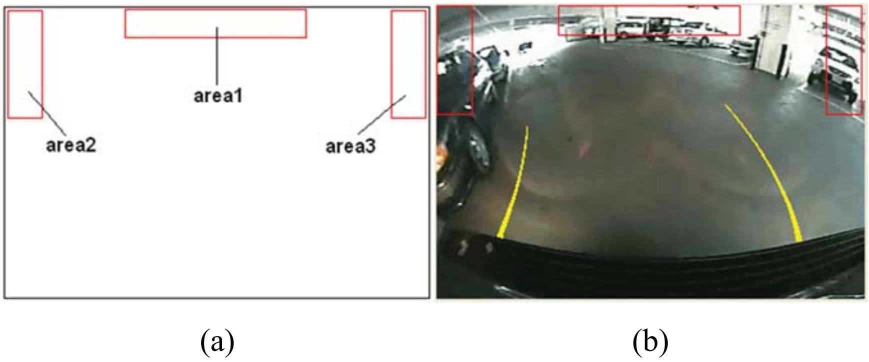


Figure 61: Allocation of motion detection areas.

(a) The defined location of area1, area2 and area3. (b) An illustration of detection areas in a realistic image.

2. Block Matching (BM) Algorithm

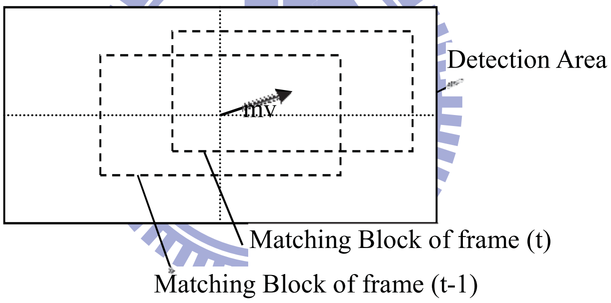


Figure 62: The scheme of block matching

Figure 62 shows the scheme of block matching. A matching block is used to find the local motion vector in the detection area. The BM algorithm is quite simple that it firstly calculates the sum of absolute difference (SAD) between matching blocks of two fames for every possible motion vector (mv), and then searches for the motion vector corresponding to the minimum of SADs. BM algorithm can be achieved as follows;

$$SAD_i(p,q) = \sum_{X,Y \in A_i} \left| I(t-1,X,Y) - I(t,X+p,Y+q) \right|$$

$$lmv_i = (x_i, y_i), \quad for\ SAD_i(x_i, y_i) = \min_{p,q} SAD_i(p,q) \tag{4.27}$$

where *I*(*t-1, X, Y*) is the intensity of the pixel (*X, Y*) at frame $t-1$, A_i is the i-th detection area in the image. *SAD*$SAD_i(p, q)$ is the SAD measure for a shift (*p*, *q*) between the pixels in area *i* at frame $t-1$ and the relative shifting points at frame *t*. Since $\min_{p,q} SAD_i(p,q)$ is the minimum SAD value in area *i*, the shift vector lmv_i that produces the minimum SAD value for area *i* represents the LMV of the detection area.

3. Logical Filter with a Proportional factor

After evaluating LMVs, logical filters with proportional control are proposed to filter out sudden noises caused by moving objects and generate filtered motion vectors (FMVs). The logical filter is implemented as,

$$If\ |lmv_i(t) - lmv_i(t-1)| < TH$$

$$fmv_i(t) = lmv_i(t)$$

$$else$$

$$fmv_i(t) = k_p \cdot fmv_i(t-1),\quad where\ 0 < k_p < 1 \tag{4.28}$$

In (Eq. 4.28), a proper threshold, denoted as *TH*, is defined by pre-experiments to present an unusual change of LMVs. If the quantity of the difference of lmv_i within a sampling time is below the given threshold, the filtered motion vector of area i at frame *t*, say $fmv_i(t)$, is set to $lmv_i(t)$; otherwise, $fmv_i(t)$ is set to $fmv_i(t\text{-}1)$ multiplied by a proportional factor k_p.

4. Tuning degree Evaluation

In order to calculate the turning degree (TD) of a car based on filtered motion vectors generated from the motion estimation unit, a weight-based combination formula is proposed as follows;

$$TD = w_1 \cdot x_1 + w_2 \cdot (y_3 - y_2) + w_3 \cdot (x_2 + x_3) \tag{4.29}$$

where (x_i, y_i) for i=1, 2, 3 are the filtered motion vectors with respect to area *i*, and $0 < w_1, w_2, w_3 < 1$ are weight factors associated with different kinds of motion features. The turning degree (TD) is a combination of various motion features. Motion features in (Eq. 4.29) can be divided into three categories: the center-horizontal motion, side-vertical motions and side-horizontal motions. The center-horizontal motion, which is presented by the motion parameter x_1, dominates the turning direction and degree. The difference between side-vertical motions characterized by $(y_3 - y_2)$ and summation of side-horizontal motions characterized by $(x_2 + x_3)$ can also characterize the turning degree. A large quantity of $(y_3 - y_2)$ or $(x_2 + x_3)$ refers to a big turning. In this work, the weights are experimentally tuned and set to be robust to various environments. Moreover, the weights of last two motion features, say w_2 and w_3, are set smaller than w_1, because motion vectors detected from area2 and area3 are sometimes influenced by the depth of field in side areas.

Some simple post-processing methods are proposed for DDG system to enhance the robustness of turning degree evaluation and produce a quantitative value for practical usage. Post-processing methods applied in DDG are achieved as following procedures;

$$QTD(t) = (1-\alpha) \cdot QTD(t-1) + \alpha \cdot Q(TD(t)) \text{ , } \quad 0<\alpha<1 \qquad (4.30)$$

$$Q(TD(t)) = N \cdot TD(t)/max|TD(t)| \qquad (4.31)$$

where $QTD(t)$ is the quantized turning degree, N is a numerical number used to scale the quantity of turning degree. The value of parameter N is set proportionally to the resolution of the output image. The parameter α in (Eq. 4.31) is set closely to 1 to preserve most sensitivity of turning degree.

5. Experiment Results

5.1 Obstacle Detection Experiment

We arranged the information of the working platform and listed in Table 6 shown below. To show our experimental results more clearly, we categorized the experiments according to the proposed process and approach which have been introduced in the previous sections as follows.

Table 6: The specifications of our working platform

CPU	Intel Core Duo T2050 1.6GHz
Memory	1GB DDR2 RAM
Programming Tool	Borland C++ Builder 6.0
Operation System	Microsoft Windows XP
Video Resolution	640x480
Camera Frame Rate	30fps

Table 7: The runtime in each processing step

Processing Function	Runtime / frame (ms)
FLIPM (with the IPM remapped position table)	2.793
Input image, transfer gray-level image and setting dynamic array memory	31.98
Pre-Processing	8.847
Segment searching and Polar Histogram	6.515
Histogram and Post-Processing	0.253
Obstacle tracking and extraction	0.347
Display	13.725
Total	64.46

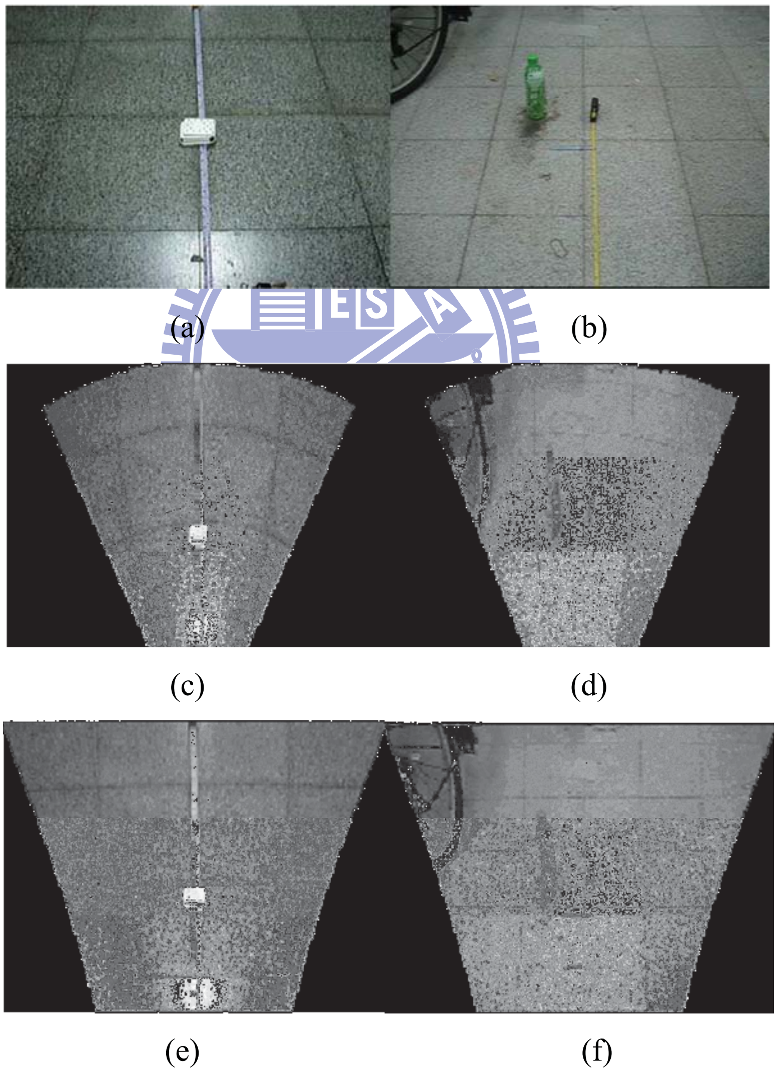
Table 8: Comparisons of different obstacle algorithms

Compared Methods	Runtime (ms)	Sensor Type	Moving compensation	Field of View
Our approach	64.46 (CPU 1.6GHz)	Fisheye Camera	YES	125∘
M. Bertozzi and A. Broggi [2]	100 (FPGA based)	Stereo Camera	NO	28∘
Wen-Liang Ji [3]	66.7(CPU 3GHz)	Single Camera	NO	34∘
P. Cerri and P. Grisleri [4]	950 (CPU 2.8GHz	Single Camera	YES	Normal Lens
S. Kyo et al. [14]	65 (multipile IMAP-VISION board)	Single Camera	NO	Normal Lens
Changhui Yang et al.[17]	50 (CPU 3.6GHz)	Single Camera	YES	Normal Lens
Guanglin Ma et al [18] (pedestrian candidate detection module)	16 (CPU 3.6GHz)	Single Camera	NO	48.8∘

Table 7 showed the runtime in each processing step defined in Figure 18. As Table 7 demonstrated, our system processed 15 frames per second. We tested the complete system by two parts, Input and Display stages. Therefore, the performance of the whole system could be improved easily by upgrading the video I/O equipment and optimizing the FLIPM Kernel functions in regions of interest. Table 8 exhibited the performance of different obstacle algorithms. We gave the compared results in four parts, including the runtime, types of sensors, moving compensation, and field of views. Although our approach included image I/O routine processes on common development platform, it still had the better performance than the others. On the detection module, we adopted the polar histogram to simplify the analytic step. It had two benefits where one was to reduce the complexity and accelerate the processing speeds, and the other was to improve the detection rate and accuracy of obstacle detection. In Table 8, [2], [17] and our approach considered the polar histogram, however, our system had the better detection rate than the others and might not be easily influenced by the planar markings, shadows, and other noises.

5.1.1 Comparisons about the Normal Lens IPM Method

In section 4, we proposed a modified forward and backward normal lens IPM equation pairs. The experimental results of our proposed approach and the most popular one by Broggi's equations were given in Figure 63. From Figure 63, the captured images by the normal lens camera were transformed to the bird-view images by using our equations. In Figure 63(b) and (c), the perspective effect was eliminated by both of Broggi's and our equations. Nevertheless, the horizontal line in the original image would be transformed to an arc by Broggi's equations as shown in Figure 63(b). With our modified equations, the horizontal straight line in the original image could be transformed to a horizontal straight line in the bird-view image as shown in Figure 63(c).


(a) (b)

(c) (d)

(e) (f)

Figure 63: The results of the normal lens IPM equations

(a) the original captured image (b) the bird-view image using Broggi's equations (c) the bird-view image using our equations

5.1.2 The Experimental Configurations

For the experiments in obstacle detection, we mounted a fisheye lens camera at the center between two side doors with the appropriate height as shown in Figure 64. To avoid disorders of frames, we would only detect the objects whose heights are more than a threshold and whose edges are quasi-vertical. The objects such as sidewalks, small balls and so on were excluded in our detection system. The experimental environments would also be constrained to the brighter backgrounds and the speed of vehicles should be under a reasonable limit so that the objects between frames would not change too drastically.

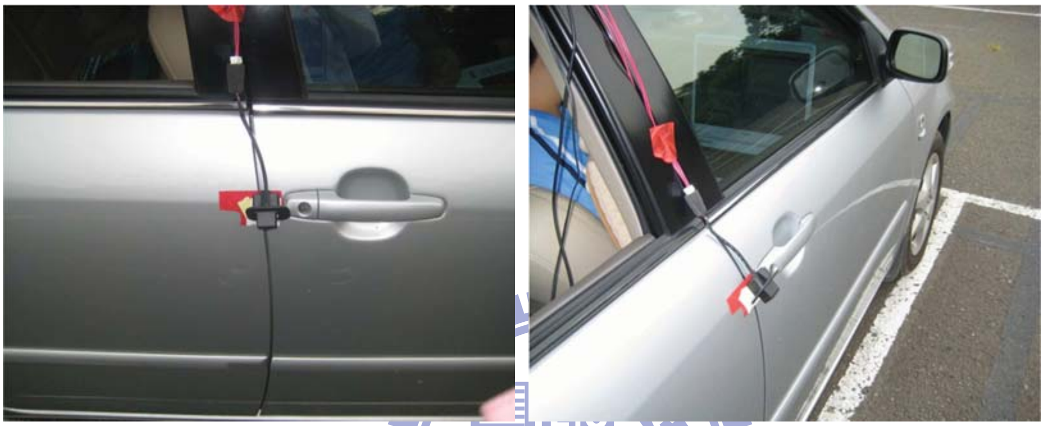
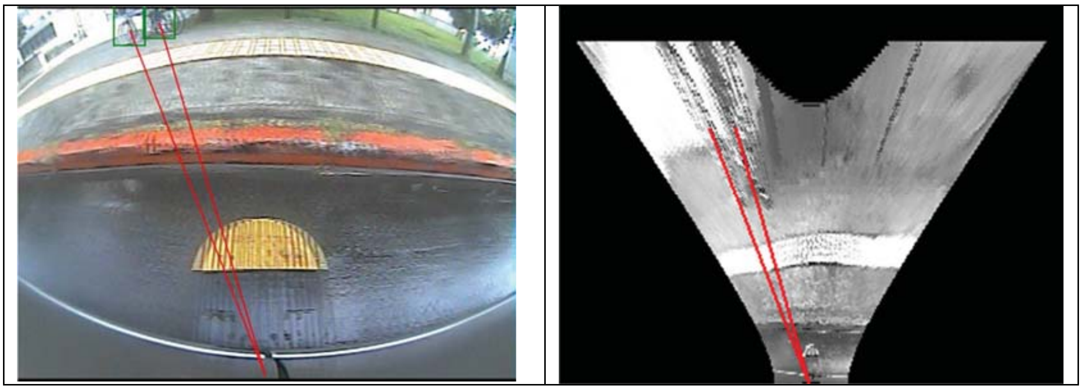
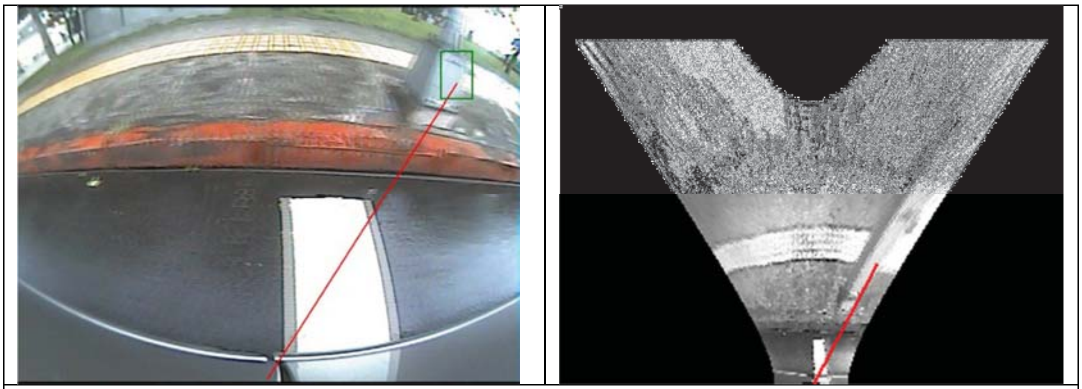


Figure 64: The set-up location of camera.

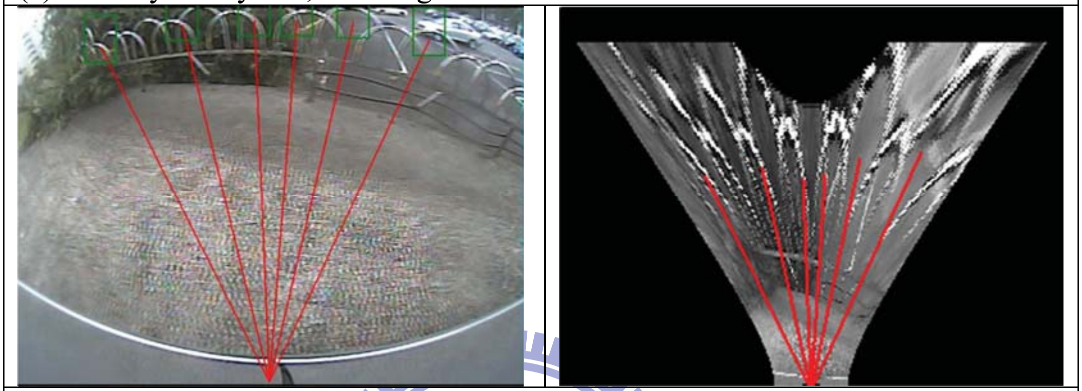
5.1.3 Results in Various Environments

As Figure 65 showed, we could accurately detect various kinds of obstacles with quasi-vertical edges by using our FLIPM methods and obstacle detection algorithm.

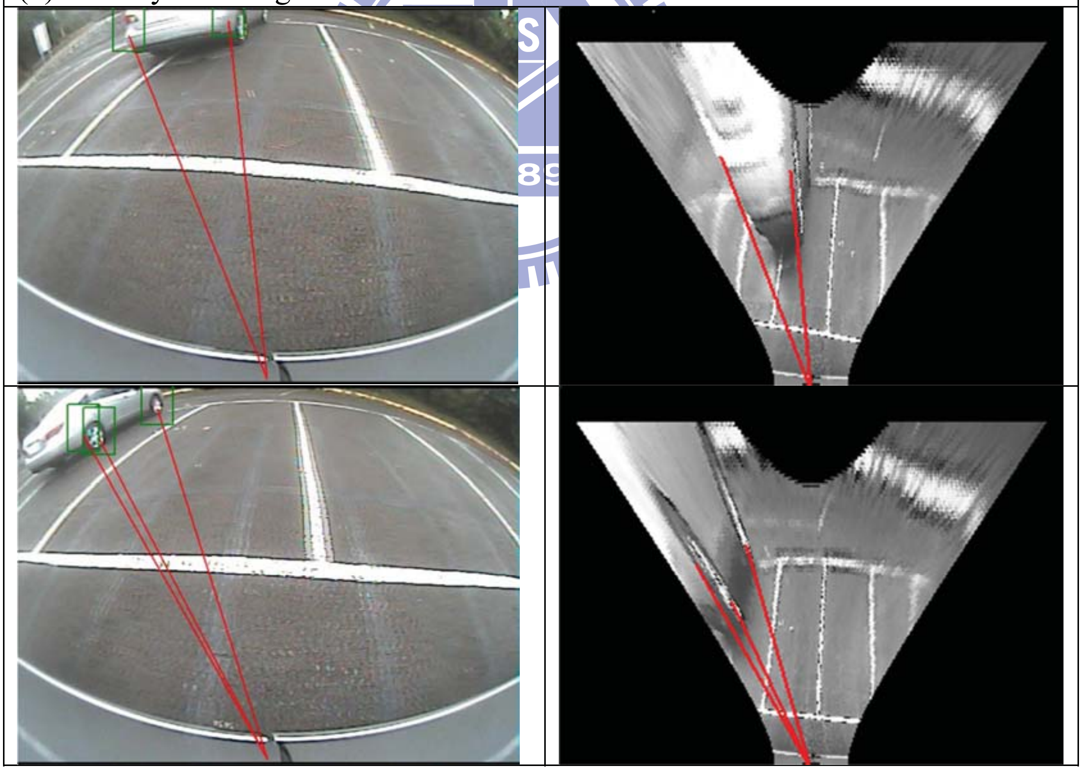


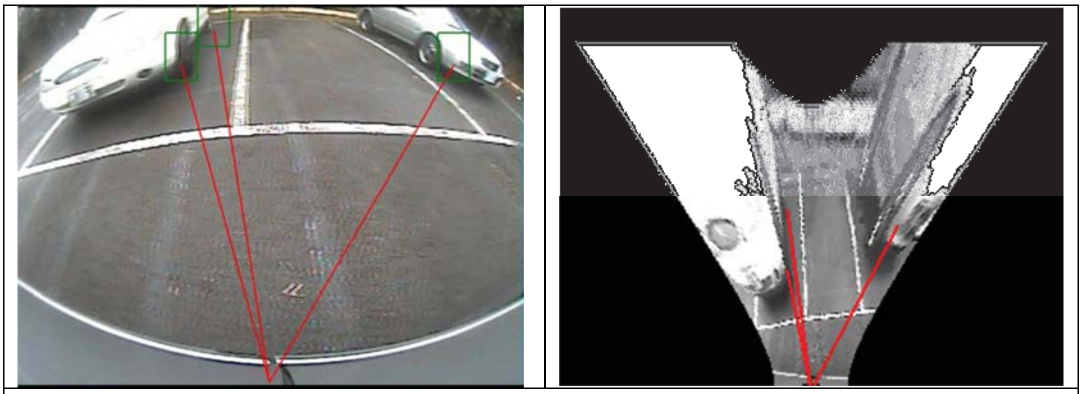


(a) Scenery1: bicyclist, street light.

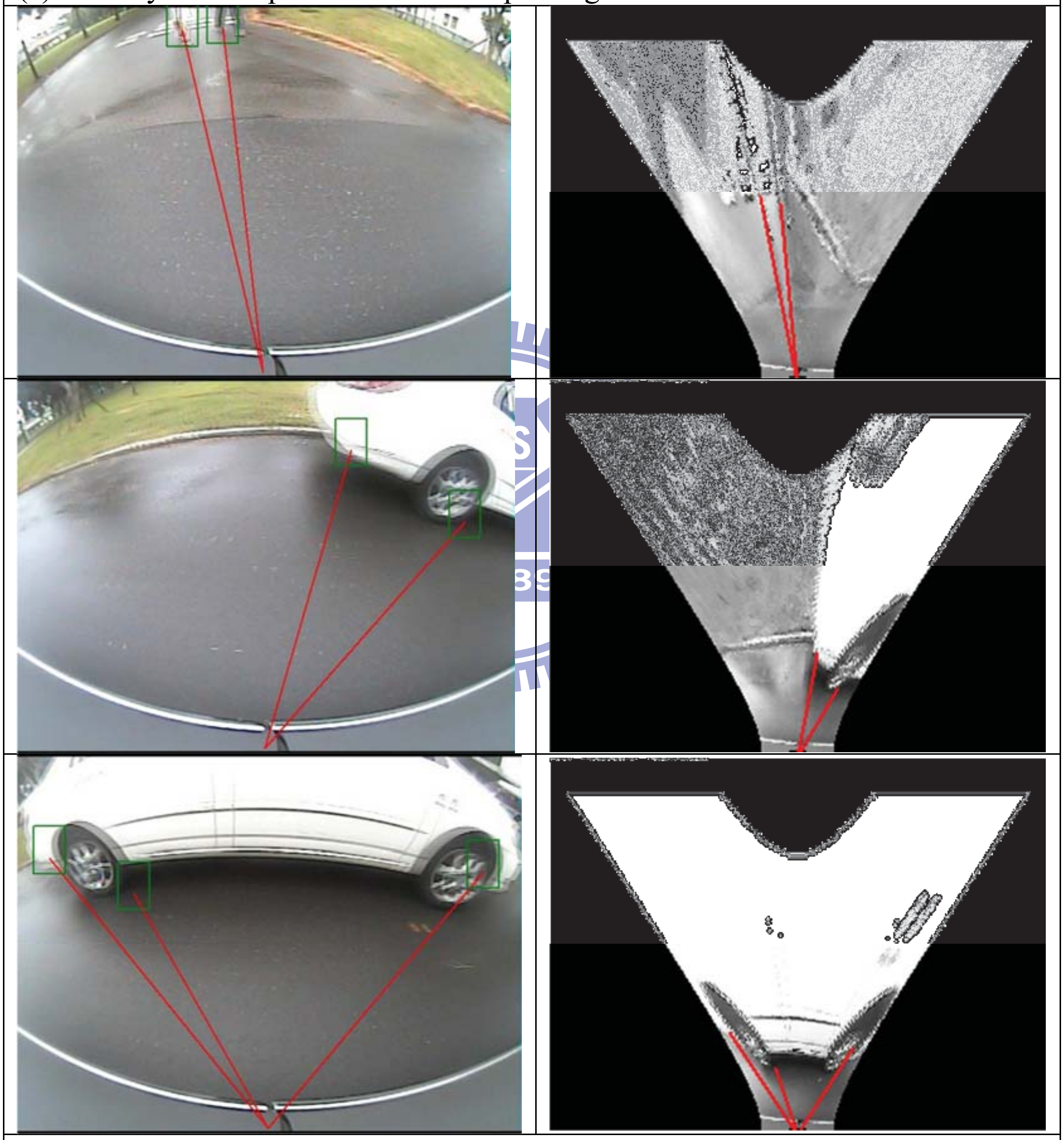


(b) Scenery 2: railings.

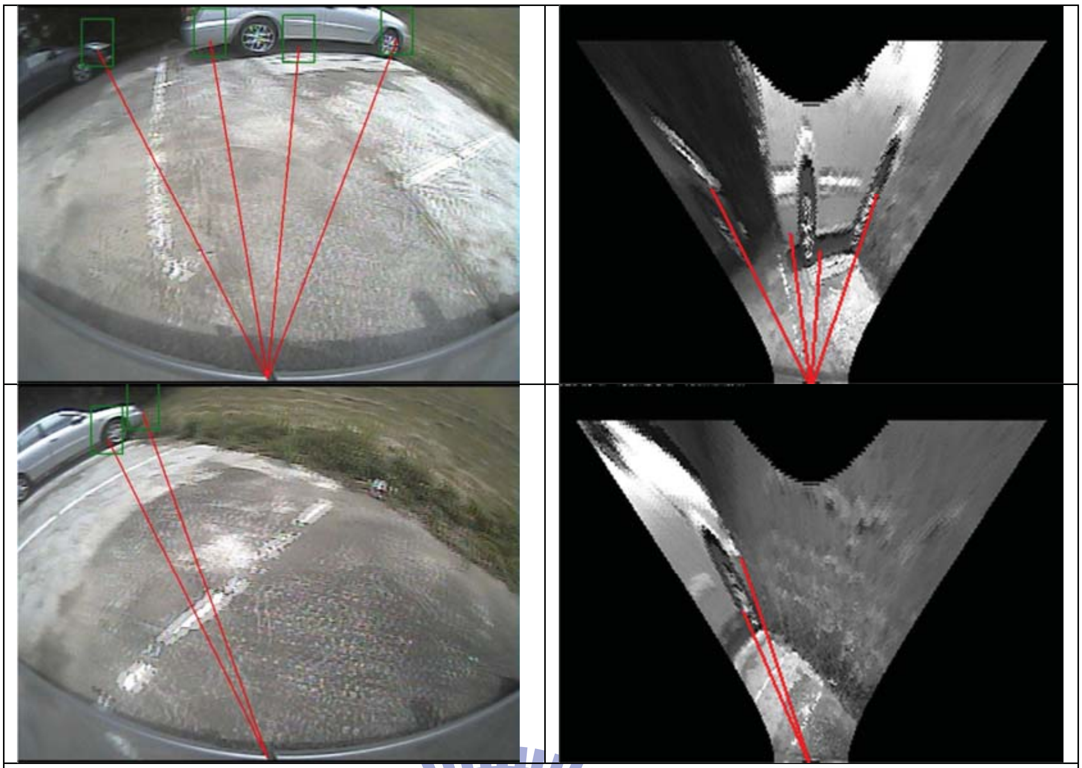




(c) Scenery 3: multiple vehicles in the parking area.



(d) Scenery 4: pedestrians, nearby vehicles

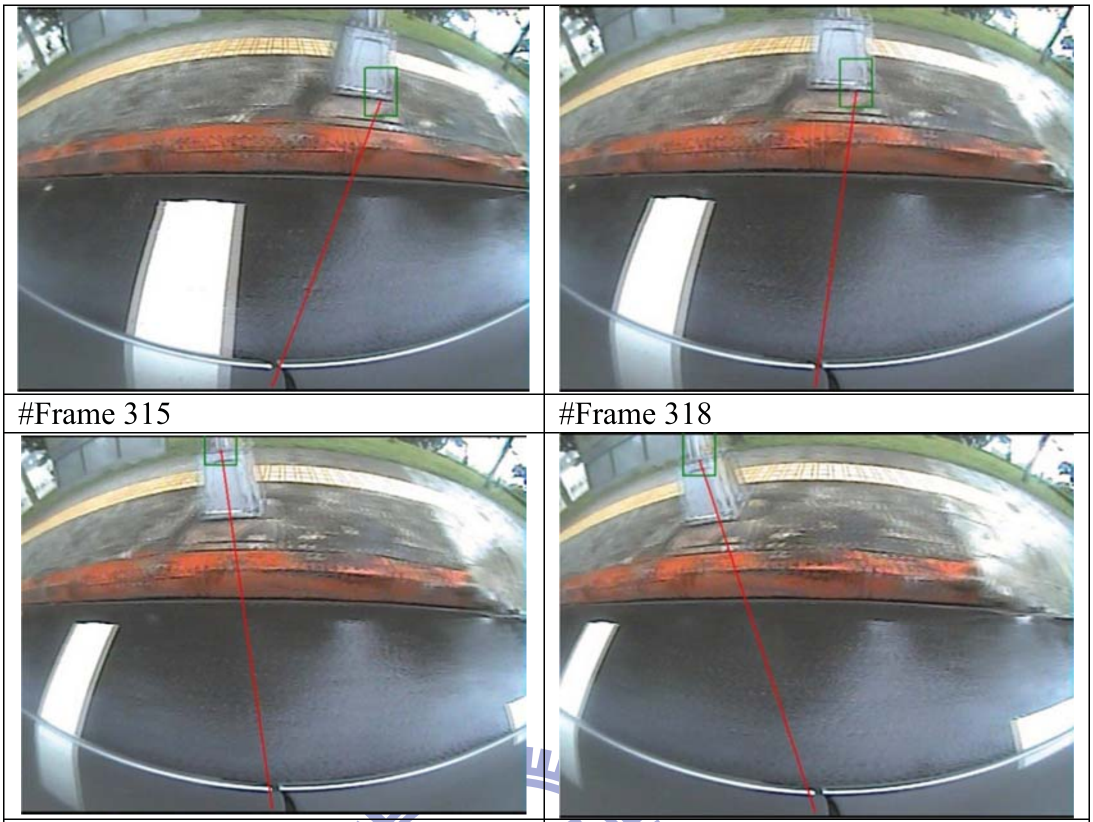


(e) Scenery 5: multiple vehicles.

Figure 65: The results of FLIPM and obstacle detection in different scenes.

(a) Scenery1: bicyclist, street light. (b) Scenery 2: railings. (c) Scenery 3: multiple vehicles in the parking area. (d) Scenery 4: pedestrians, nearby vehicles. (e) Scenery 5: multiple vehicles.

Our tracking process was carried out by iteratively checking the displacement and angular shift in the image, and we also demonstrated the results of the tracking process in successive frames as given in Figure 65 and Figure 66. According to the FLIPM method, the 3D world coordinate value could be estimated from the remapped image. In other words, when we detected the obstacle in the remapped image, we could also estimate the position information. We hence set up an obstacle warning system on the lateral and rear of the vehicle to give a warning signal when the detected obstacles were over close to our vehicle. We showed the results of the obstacle warning system In Figure 67, Figure 68 and Figure 69 where the rectangles in the upper images and the lines in the below images indicated "the position of obstacles" and "the distance and direction between the vertical projection point of camera and the detected obstacles, respectively.



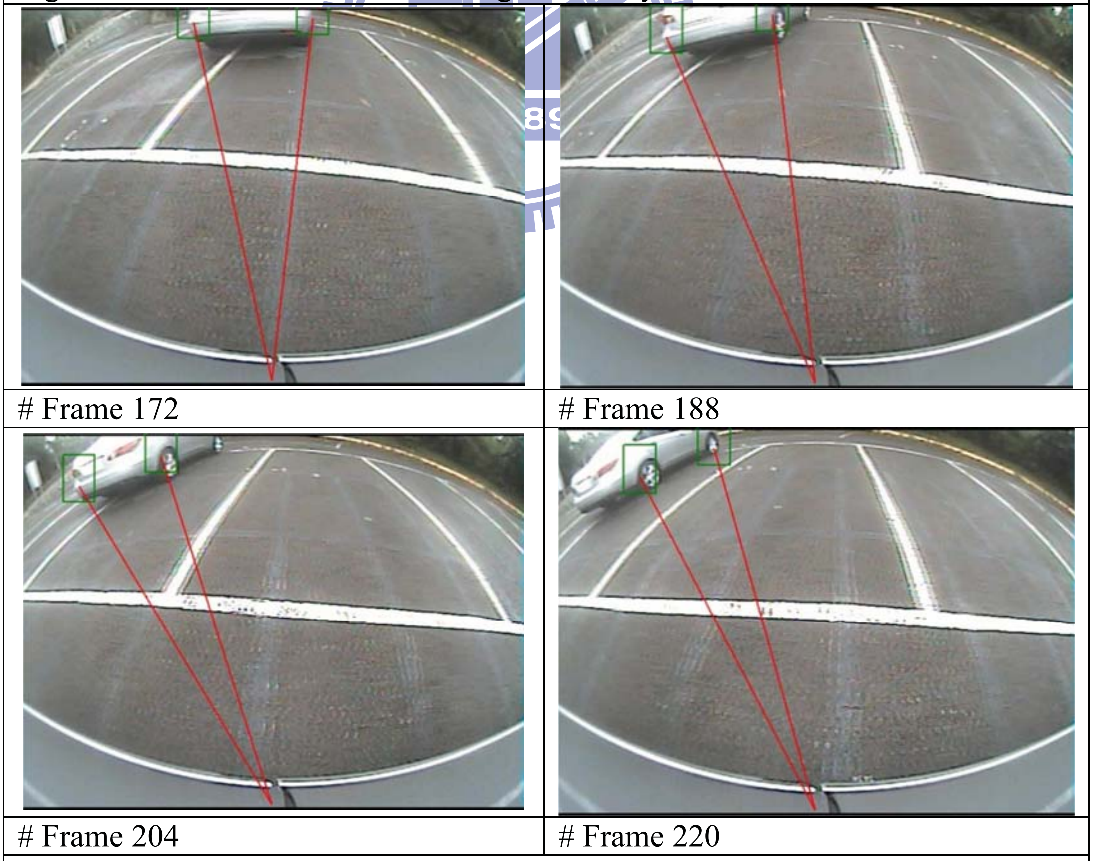
#Frame 315

#Frame 318

# Frame 322

Frame 326

Figure 66: Results of obstacle tracking in Scenery 1



# Frame 172

Frame 188

# Frame 204

Frame 220

Figure 67: Results of obstacle tracking in Scenery 3

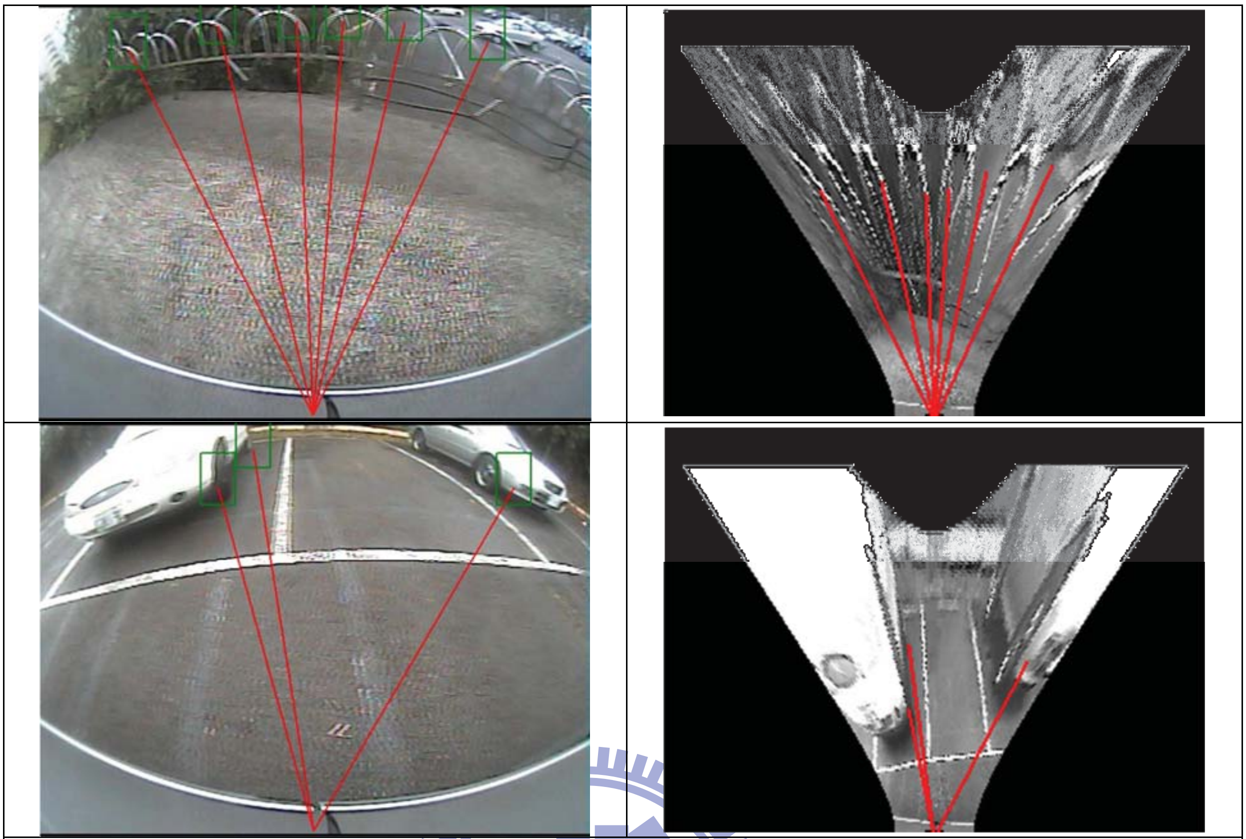


Figure 68:　Results of obstacle warning in the lateral direction

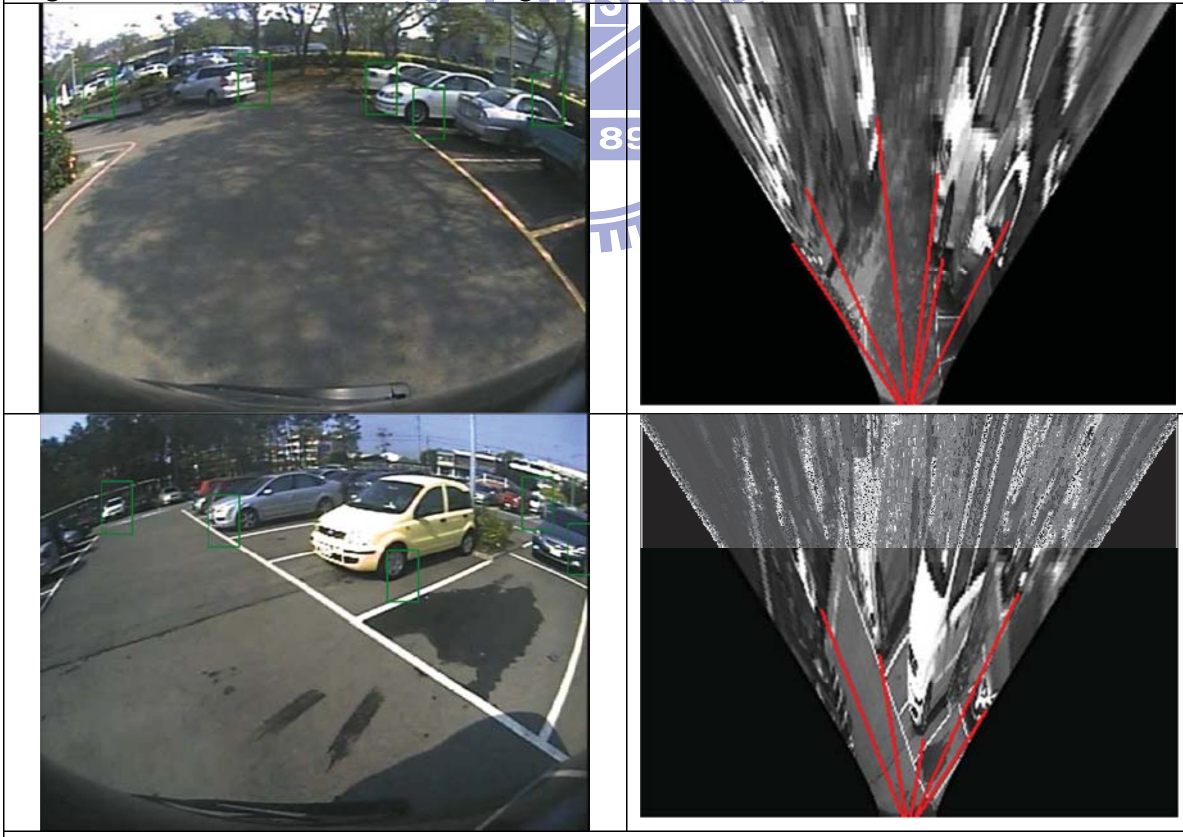


Figure 69:　Results of obstacle warning in the rear direction.

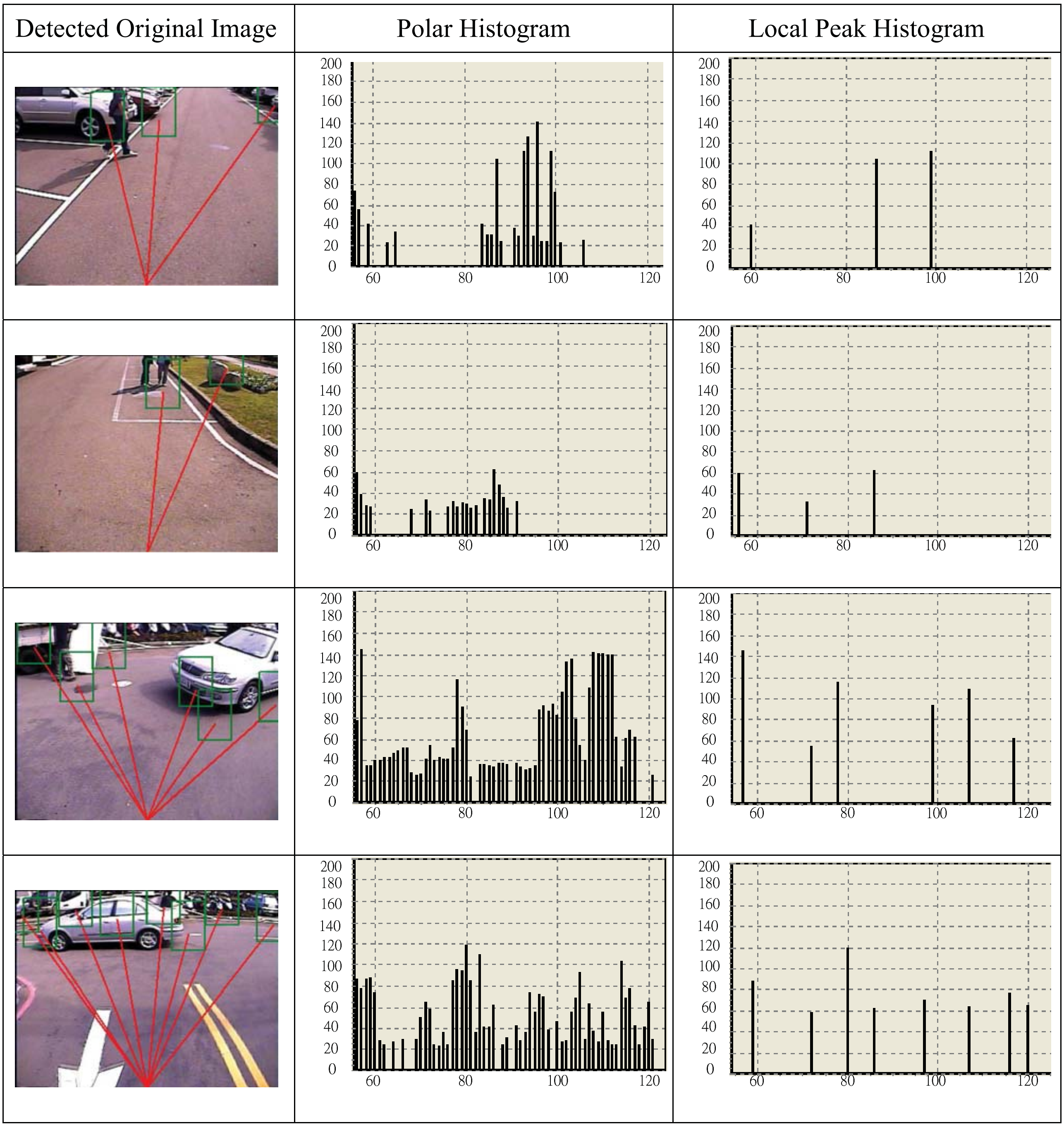
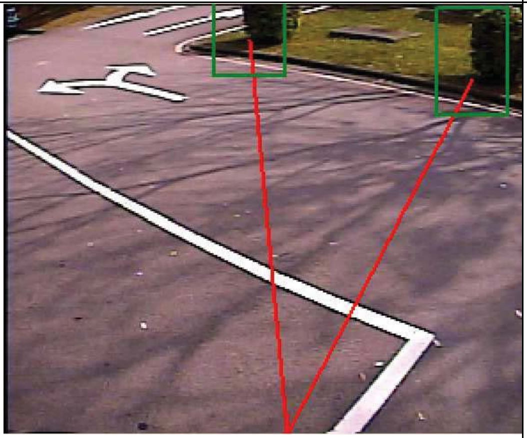
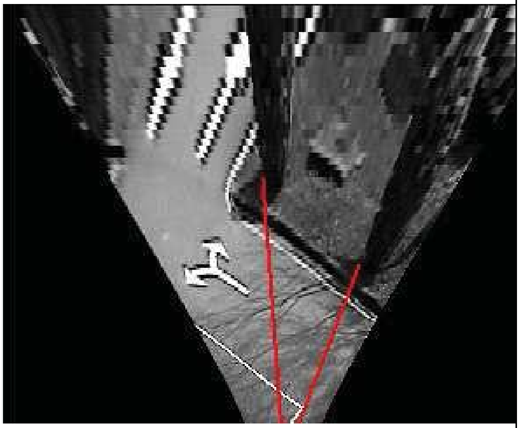
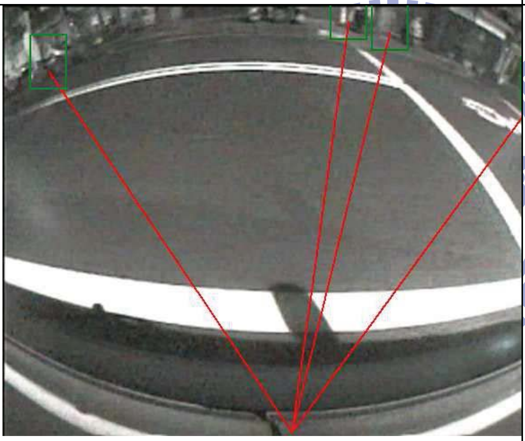
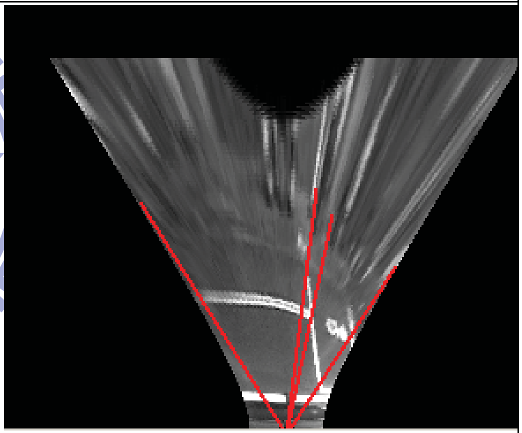
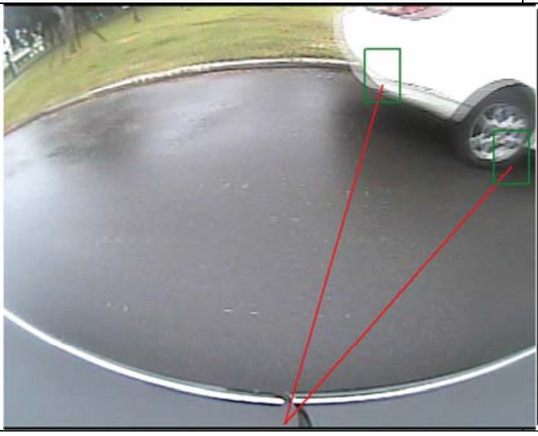
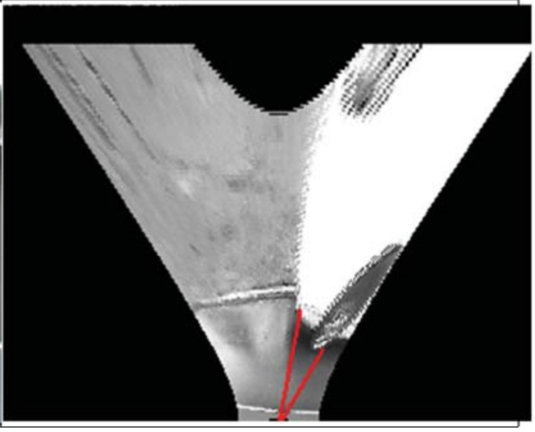


Figure 70: Results of obstacle warning with moving objects.

Upper two rows are the results with moving humans. Bottom two rows are the results with a moving vehicle turning into the corner.

In Figure 70, there were results of obstacle warning with moving objects. We had two simulation situations. One was that a pedestrian was walking form a parking region to another side while the demonstrated vehicle was leaving the parking region. The other situation contained a corner where a coming vehicle, the static obstacle and clear planar markings existed. Here we presented two issues in the moving obstacle detection and compensation estimation with rotation angles. We could get the results from distributions of the right polar histograms, localize each non-planar obstacle by dominant peaks, and filter the lane markings by trapezoid distributions in the original polar histograms.

Environment	Detected Original Image	FLIPM image
Daytime with Shadows		
Nighttime with Shadows		
(C) Raining Daytime on wet ground with light reflection		

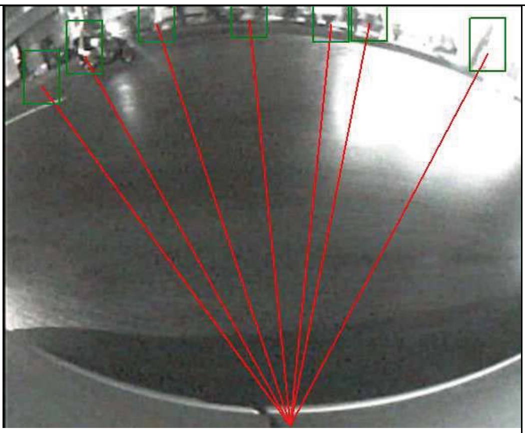
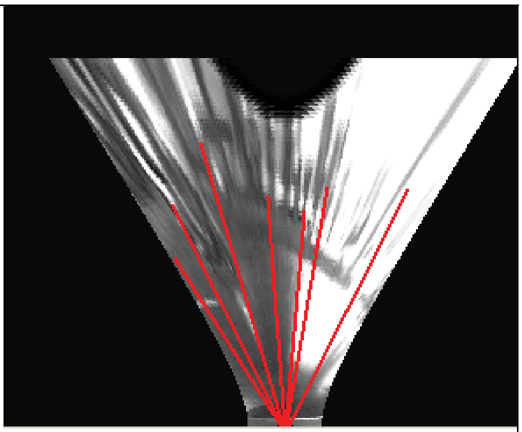
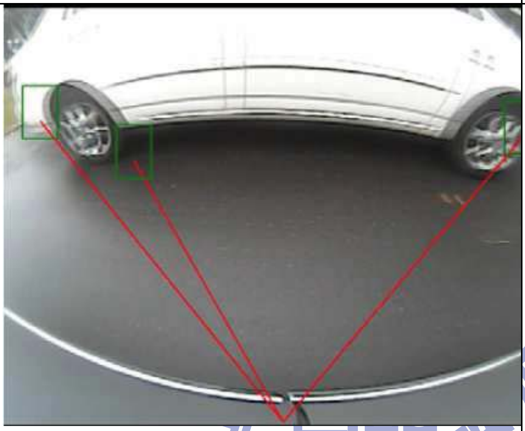
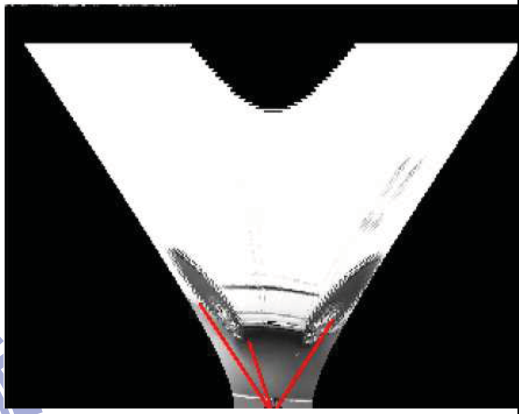
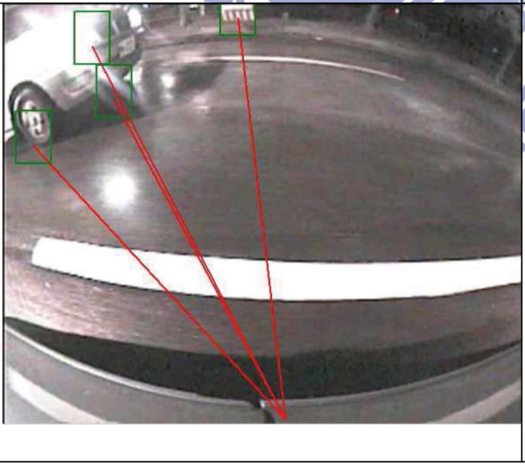
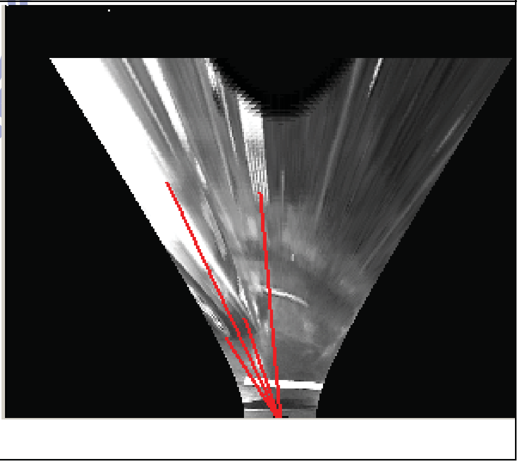
(D) Night on wet ground with light reflection		
(E) Daytime with other vehicles		
(F) Nighttime with other vehicles		

Figure 71: The results in different environments with heavy noise.

(A) In the daytime with many shadow effect. (B) In the nighttime with self shadow projected by several different direction road lights. (C) In the raining daytime on wet ground with light reflection. (D) In the raining night time on wet ground with strong light reflection. (E) Daytime with other vehicles. (F) Nighttime with other vehicles.

In Figure 71, there existed some simulated environments with heavy noises, such as shadows, light reflection, and light refraction from wet roads. In case (A) where there existed many shadows of trees on the grounds, we could obtain the remapping image as shown in the right one by the FLIPM transformation. By our approach, the shadows could be filtered out by the polar histogram if its accumulations on each angle were small and its shapes were in trapezoid. For case (B) where there were two road lights at the front and rear of our vehicle in the nighttime, we would find two different shadows on the ground. One was not clearly recognized from far light projection and the other was obvious due to near light projection. Our proposed method, however, could take advantage of FLIPM effects to remove the shadowing effect no matter how serious the illuminating conditions might be. Our compensation estimation could shift the new frame to the adaptive position to gain the minimum candidate pixels of obstacles. For case (C) and case (D), we demonstrated our results to be reliable and satisfactory in the raining conditions in the daytime and nighttime. In the nighttime, noises from light reflection were much more serious than those in the daytime, and our approach could successfully avoid misrecognizing them as obstacles in the fixed illumination condition. As for case (E) and case (F), we showed the experimental results in the common situations which simulated the roads in the daytime and nighttime. As a result of advantages of the fisheye camera in a wider angle of view, our obstacle detection algorithm could detect the range of the field of view up to 125 degrees which was much wider than other algorithms by common lens.

5.1.4 Accuracy Evaluation of Obstacle Distance

For evaluation of distance measurement, we compare length of real line with the length which estimated by the proposed procedure to obtain the distance measurement error. By testing 590 data such as illustrated in Figure 72 the land marking, the error of distance measurement within short and long range respectively is shown in Table 9. Due to the geometric characteristic of calibration procedure, the calibration error in the far range will be enlarged. Therefore, the average distance measure error of near range is about 0.16m but of far range is about 0.68m. The experimental results demonstrate that estimation for distance of target position is accurately.

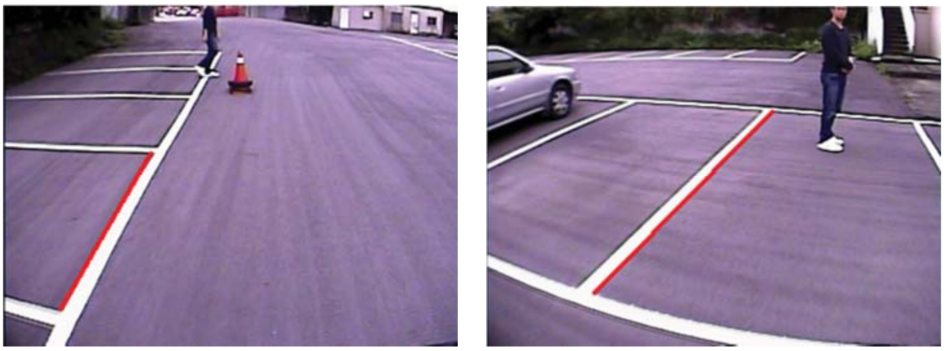
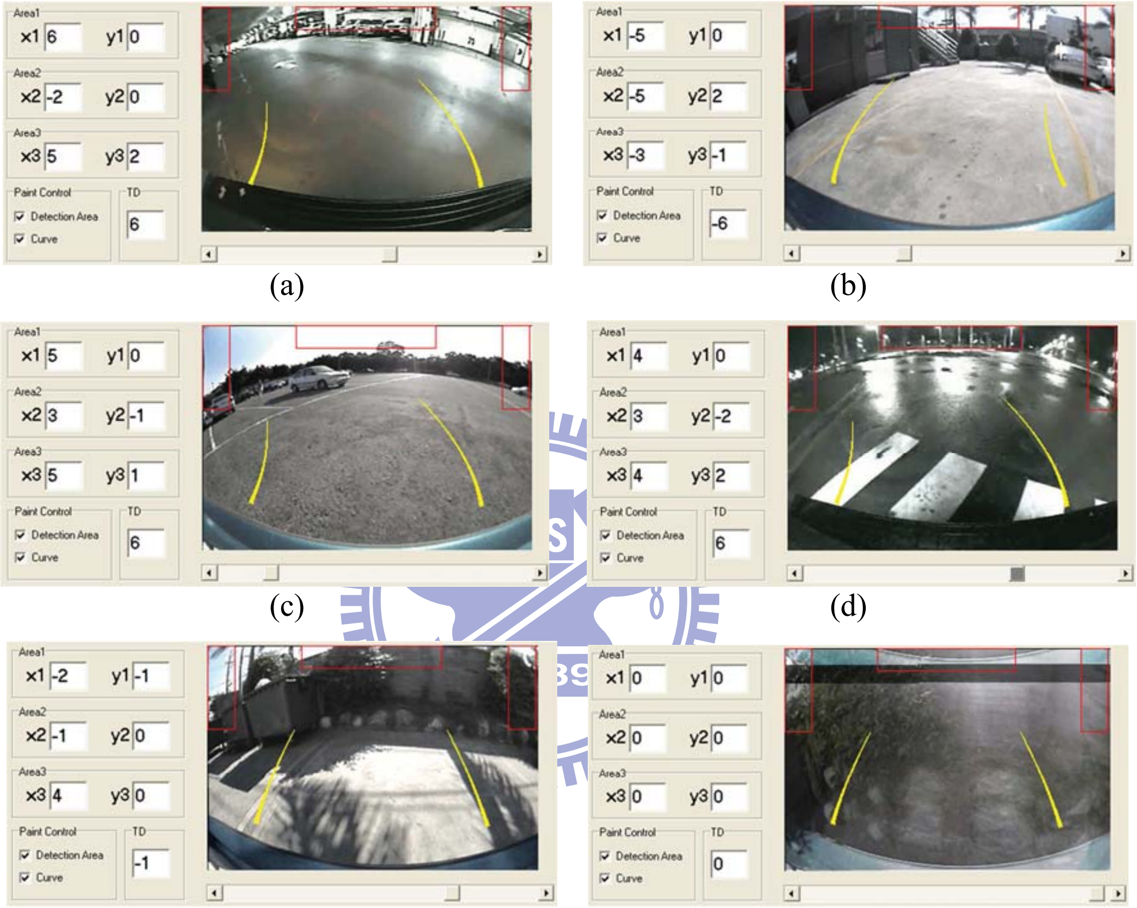


Figure 72: Lane marking for distance measurement

Table 9: Experimental result of distance measurement

	Near range (3~5m)	Far range (5~8m)
Average Distance Error (m)	0.16 m	0.68 m

5.2 Dynamic Distance Gauge (DDG)



Area1
x1 6 y1 0
Area2
x2 -2 y2 0
Area3
x3 5 y3 2
Paint Control
☑ Detection Area
☑ Curve
TD
6

(a)

Area1
x1 -5 y1 0
Area2
x2 -5 y2 2
Area3
x3 -3 y3 -1
Paint Control
☑ Detection Area
☑ Curve
TD
-6

(b)

Area1
x1 5 y1 0
Area2
x2 3 y2 -1
Area3
x3 5 y3 1
Paint Control
☑ Detection Area
☑ Curve
TD
6

(c)

Area1
x1 4 y1 0
Area2
x2 3 y2 -2
Area3
x3 4 y3 2
Paint Control
☑ Detection Area
☑ Curve
TD
6

(d)

Area1
x1 -2 y1 -1
Area2
x2 -1 y2 0
Area3
x3 4 y3 0
Paint Control
☑ Detection Area
☑ Curve
TD
-1

(e)

Area1
x1 0 y1 0
Area2
x2 0 y2 0
Area3
x3 0 y3 0
Paint Control
☑ Detection Area
☑ Curve
TD
0

(f)

Figure 73: Testing results of DDG in various environments.

(a) Left tuning at an indoor place. (b) Right turning at an outdoor place. (c) Testing at an outdoor place with low texture in the background. (d) Testing at night-time. (e) Slightly left turning at outdoor place with swinging trees. (f) Stop turning at outdoor place with signal disturbance.

The proposed system in this work has been implemented and tested on both PC-based and real-time embedded platforms. Figure 73 demonstrates the testing results of the proposed DDG system in various conditions. In Figure 73, the car is turning left at an indoor place, and the value of turning degree (TD) is positive. Moreover, motion features in Eq. (4.29) including x_1, $(y_3 - y_2)$ and $(x_2 + x_3)$ are all positive. Figure 73 shows a right turning at an outdoor place. In this case, the values of TD and motion features defined in Eq. (4.29) are all negative. Figure 73(c) and (d) shows that the proposed system can perform well at low texture condition and night time. Figure 73(e) and (d) demonstrates the robustness of the proposed system. The testing result in Figure 73(e) shows the DDG system is robust to the disturbance of trees' swinging; because the sudden motions caused by the movement of leaves can be filtered by the logical filter. Figure 73(f) shows the testing result with signal disturbance caused by an improper hardware connection, and TD is hold at zero when there is no turning. Experimental results in this work show that the proposed DDG algorithm can actually evaluate the turning degree of a car and robustly perform at various conditions.

5.3 Entire System Experiment

This experiment includes three entries. First stage tests the intelligent function on a RSU. RSU captures the video stream from cameras setting on the intersection with difference directions. Then, analyzing multi-channel video streaming can estimate the collision rate of moving objects. The next entry is WAVE/DSRC package transmission test. Driving two or more vehicle with different interval, we test the efficiency and accuracy. The Last test is to receive messages on the vehicle, and verify its content and protocol fit in with WSM and SAE J2576 telematics protocol.

5.3.1 Environment description

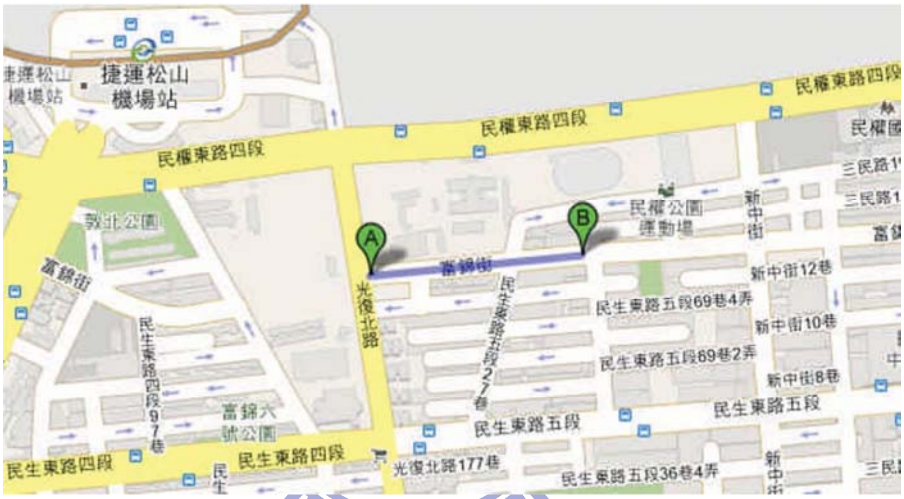


Figure 74: Test Street Location

We put a wave box in the A Car and put another in the B Car. The longest distance between these two cars are 100 meters long, and the shortest distance is 20 meters long. During each test, we average the success rate by transmitting 10 packages. All test structure is present in Figure 74.

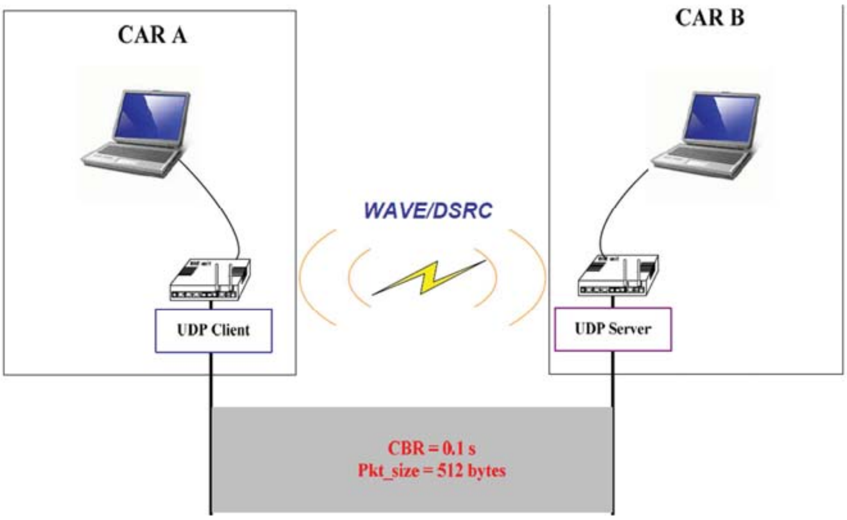
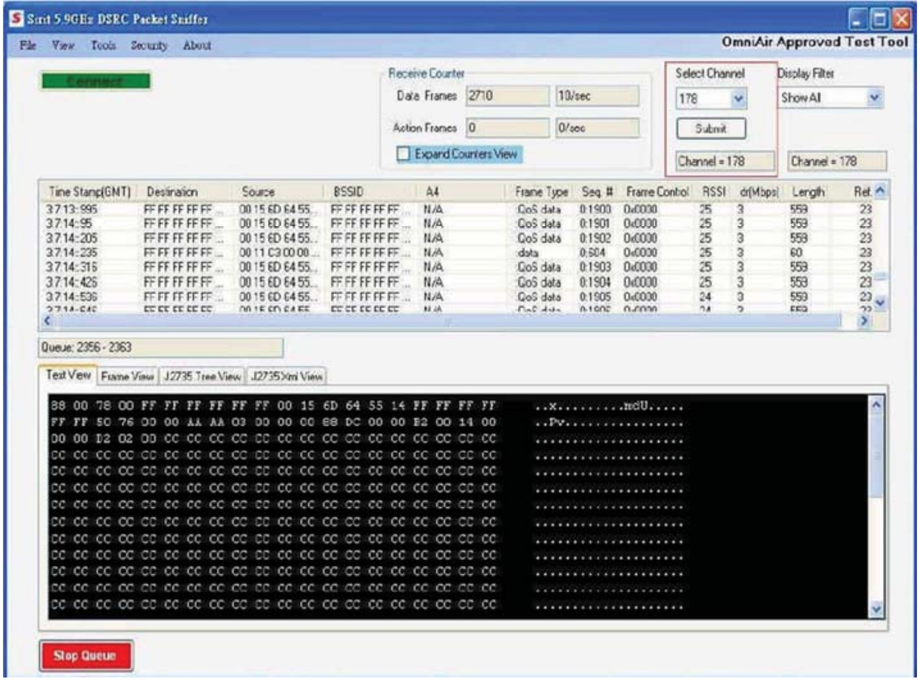


Figure 75 Experiment structure for testing transmission rate

Table 10: Parameters of Testing

SAVARI S200	
Tx Power (dbm)	10
Channel	178
MAX RF Distance (m)	250
Input parameter	
# of UDP Packets	600
Packet Size (bytes)	512
CBR (HZ)	10

5.3.2 Protocol Test Results



Receive Counter

Data Frames	2710	10/sec
Action Frames	0	0/sec

Select Channel: 178

Display Filter: Show All

Channel = 178

Channel = 178

Time Stamp(GMT)	Destination	Source	BSSID	A4	Frame Type	Seq. #	Frame Control	RSSI	dr(Mbps)	Length	Ref.
3:7:13::995	FF FF FF FF FF ...	00 15 6D 64 55...	FF FF FF FF FF ...	N/A	QoS data	0:1900	0x0000	25	3	559	23
3:7:14::95	FF FF FF FF FF ...	00 15 6D 64 55...	FF FF FF FF FF ...	N/A	QoS data	0:1901	0x0000	25	3	559	23
3:7:14::205	FF FF FF FF FF ...	00 15 6D 64 55...	FF FF FF FF FF ...	N/A	QoS data	0:1902	0x0000	25	3	559	23
3:7:14::235	FF FF FF FF FF ...	00 11 C3 00 00 ...	FF FF FF FF FF ...	N/A	data	0:604	0x0000	25	3	60	23
3:7:14::316	FF FF FF FF FF ...	00 15 6D 64 55...	FF FF FF FF FF ...	N/A	QoS data	0:1903	0x0000	25	3	559	23
3:7:14::426	FF FF FF FF FF ...	00 15 6D 64 55...	FF FF FF FF FF ...	N/A	QoS data	0:1904	0x0000	25	3	559	23
3:7:14::536	FF FF FF FF FF ...	00 15 6D 64 55...	FF FF FF FF FF ...	N/A	QoS data	0:1905	0x0000	24	3	559	23

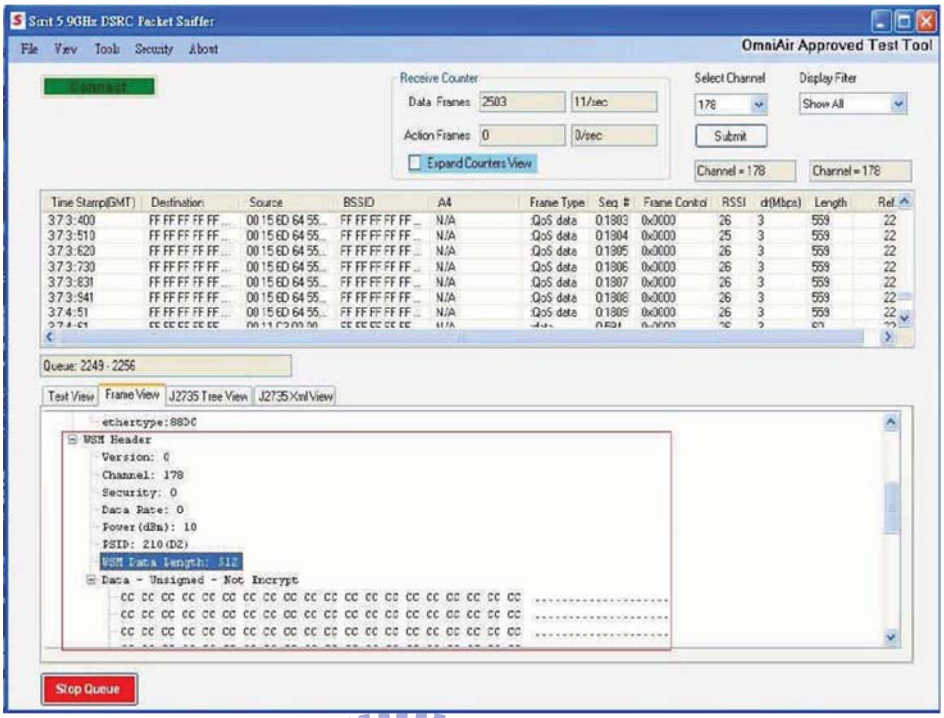
Queue: 2356 - 2363

```
88 00 78 00 FF FF FF FF FF FF 00 15 6D 64 55 14 FF FF FF FF     ..x.........mdU.....
FF FF 50 76 00 00 AA AA 03 00 00 00 88 DC 00 00 B2 00 14 00     ..Pv................
00 00 D2 02 00 CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC     ....................
CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC     ....................
CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC     ....................
CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC     ....................
CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC     ....................
CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC     ....................
CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC     ....................
CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC     ....................
CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC     ....................
CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC     ....................
CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC CC     ....................
```

Figure 76: WSM Captured Message

As shown in Figure 76, the red block is the WSM message received by Sniffer Card in

Channel 178.


Sirit 5.9GHz DSRC Packet Sniffer
File View Tools Security About
OmniAir Approved Test Tool
Connect
Submit
Test View
Frame View
J2735 Tree View
J2735XmlView
Stop Queue

Receive Counter
Data Frames 2503 11/sec
Action Frames 0 0/sec
□ Expand Counters View

Select Channel 178
Display Filter Show All
Channel = 178 Channel = 178

Time Stamp(GMT)	Destination	Source	BSSID	A4	Frame Type	Seq #	Frame Control	RSSI	dr(Mbps)	Length	Ref.
3:7:3:400	FF FF FF FF FF ...	00 15 6D 64 55...	FF FF FF FF FF ...	N/A	QoS data	0:1803	0x0000	26	3	559	22
3:7:3:510	FF FF FF FF FF ...	00 15 6D 64 55...	FF FF FF FF FF ...	N/A	QoS data	0:1804	0x0000	25	3	559	22
3:7:3:620	FF FF FF FF FF ...	00 15 6D 64 55...	FF FF FF FF FF ...	N/A	QoS data	0:1805	0x0000	26	3	559	22
3:7:3:730	FF FF FF FF FF ...	00 15 6D 64 55...	FF FF FF FF FF ...	N/A	QoS data	0:1806	0x0000	26	3	559	22
3:7:3:831	FF FF FF FF FF ...	00 15 6D 64 55...	FF FF FF FF FF ...	N/A	QoS data	0:1807	0x0000	26	3	559	22
3:7:3:941	FF FF FF FF FF ...	00 15 6D 64 55...	FF FF FF FF FF ...	N/A	QoS data	0:1808	0x0000	26	3	559	22
3:7:4:51	FF FF FF FF FF ...	00 15 6D 64 55...	FF FF FF FF FF ...	N/A	QoS data	0:1809	0x0000	26	3	559	22

Queue: 2249 - 2256

```
  ethertype:88DC
⊟ WSM Header
    Version: 0
    Channel: 178
    Security: 0
    Data Rate: 0
    Power(dBm): 10
    PSID: 210(D2)
    WSM Data Length: 512
  ⊟ Data - Unsigned - Not Encrypt
      cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc   ....................
      cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc   ....................
      cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc   ....................
```

Figure 77: WSM header

The red block in Figure 77 is WSM header. It conforms the protocol with WSM Version : 0、Security Type : 0、Channel Number : 178、Data Rate ：0、TxPwr_Level ：10、Provider Service Identifier ：210、WSM Length ：512、WSM Data ：cc cc …

Table 11: Static test (600 packages per test)

Distance	No1.	No2.	No3.	No4.	No5.	No6.	No7.	No8.	No9.	No10	Avg
20 m	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
40 m	100%	99.33 %	100%	100%	100%	100%	100%	100%	100%	100%	99.93 %
60 m	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
80 m	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
100 m	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

Table 12: Active test (600 packages per test, 30 Km/hr)

Distance	No1.	No2.	No3.	No4.	No5.	No6.	No7.	No8.	No9.	No10	Avg
10m~ 100m	98.5% (591)	96.3% (578)	96.5% (579)	95.1% (571)	99.1% (595)	97.3% (584)	96.3% (578)	95.8% (575)	97.1% (583)	98.8% (593)	97.08 %

From Table 11and Table 12, the active test is more meaningful. Car A and Car B moves in 30 Km/hr speed and the average success rate is around 97%.

5.3.3 Intersection Test Results

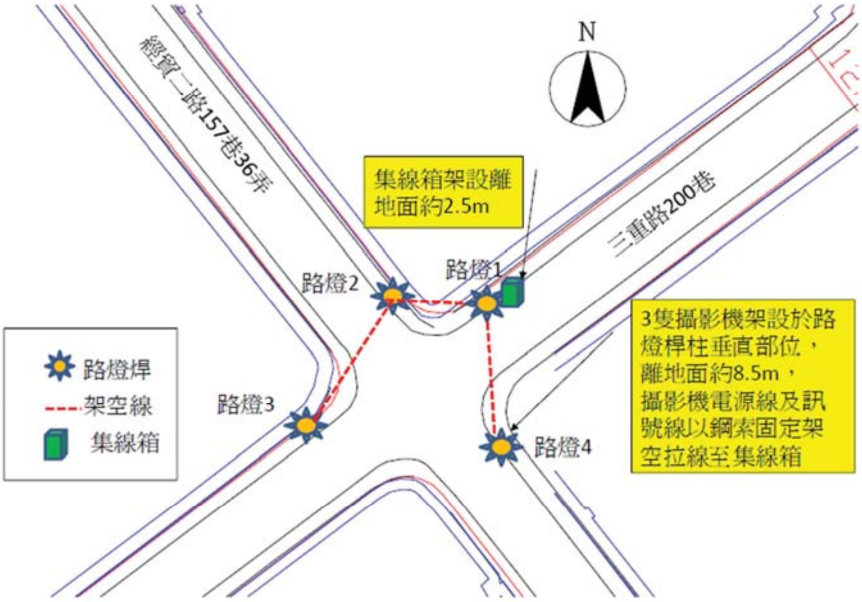

經貿二路157巷36弄
N
集線箱架設離地面約2.5m
三重路200巷
路燈2
路燈1
3隻攝影機架設於路燈桿柱垂直部位，離地面約8.5m，攝影機電源線及訊號線以鋼索固定架空拉線至集線箱
路燈焊
架空線
集線箱
路燈3
路燈4

Figure 78 : The locations of camera on setting intersection

From Figure 78, we set four cameras forward different directions, and the height of camera’s location is 8.5 meters long.

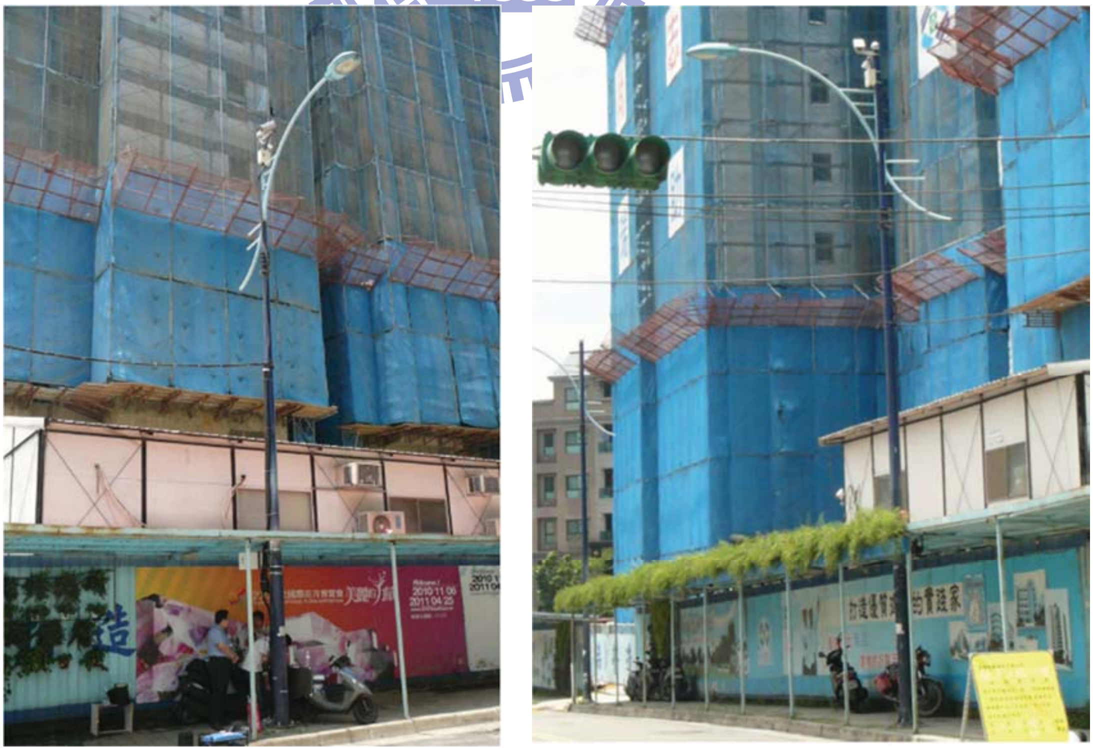


Figure 79: The location of fix camera on intersection

According the principle of government, view distance can be divided into stop view distance, contingency view distance and overtake view distance.

1. Stop view distance means when driver sees an obstacle on the road, and then starts to brake until vehicle stop distance.
2. Contingency view distance means when driver finds an unexpected road condition, and then still can change lane, speed and direction.
3. Overtake view distance means when driver overtakes front car on the reverse lane, and then accomplish the active before vehicle coming.

Table 13: The minimum distance of stop view and overtake view

Speed (Km/ Hr)	Stop view distance (meters)		Overtake view distance (meters)	
	minimum	Suggest value	minimum	Suggest value
60	70	85	290	410
50	55	65	240	340
40	40	45	200	280
30	30	30	160	220
25	25	25	140	195

We set the speed of test vehicle with receiving wave box at 40 Km/hr, and distance of view is over 50 meters long. Setting a RSU on the road side, and the minimum distance between RSU and Camera is 85 meters long. First using 3G Ethernet calibrates machine clock. Second, we simulate pedestrian crossing intersection and collision events, and then measure the whole transmission time.

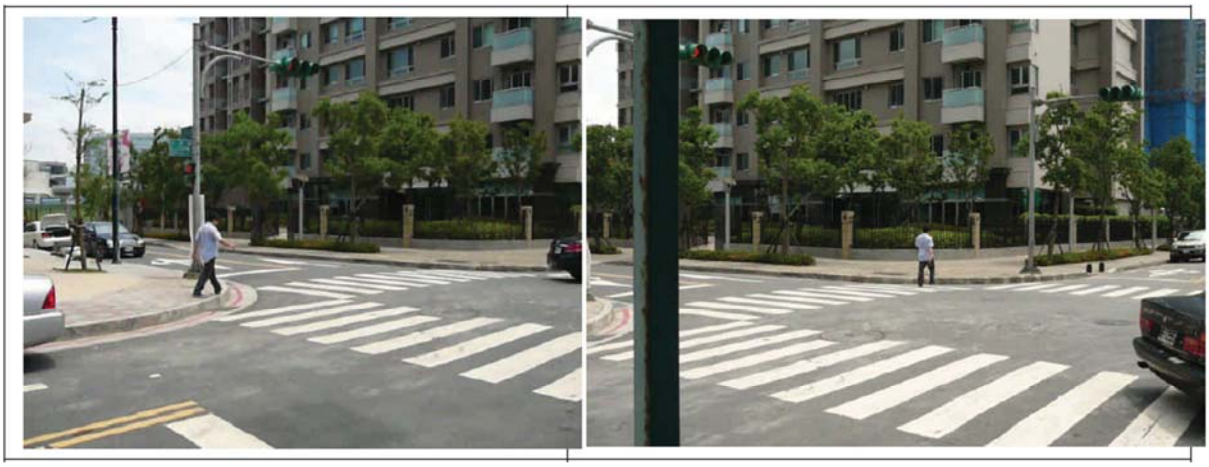


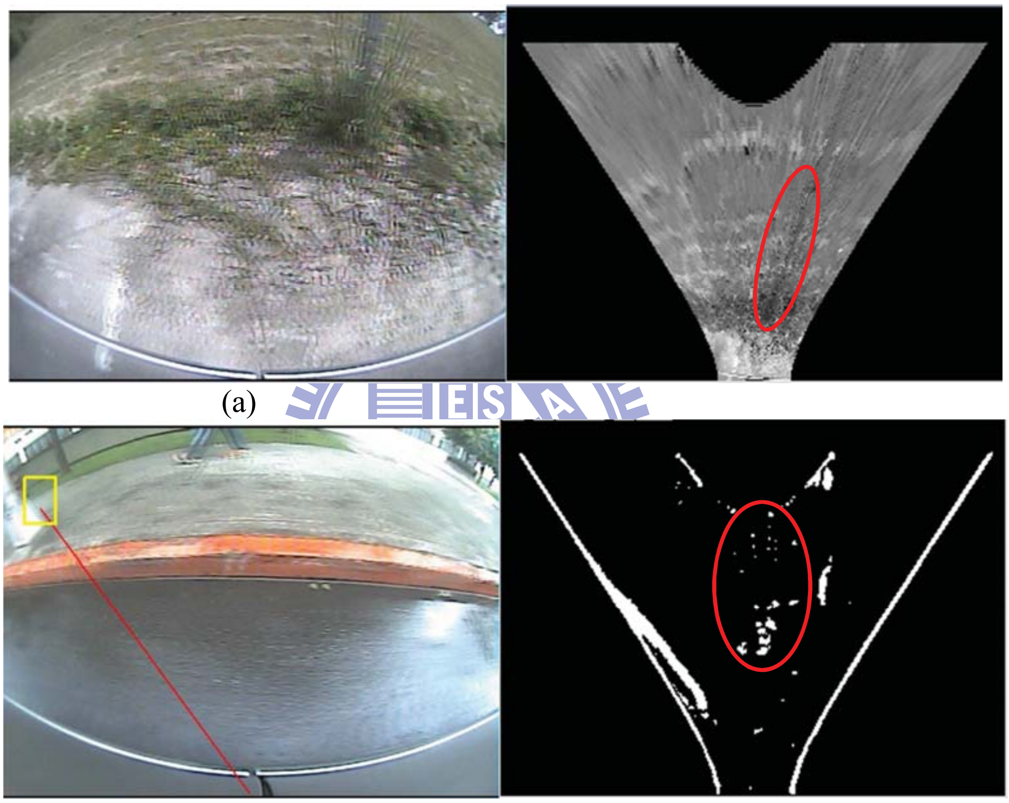
Figure 80: Simulating a pedestrian crossing intersection

Table 14 : Table of reaction time via OBU

Test No.	Reaction Time (Sec)
1	0.359
2	0.468
3	0.500
4	0.531
5	0.515
6	0.515
7	0.672
8	0.391
9	0.907
10	0.188

5.4 Discussions

Although the performance of our obstacle detection system based on FLIPM method was quite satisfactory, there have existed some disturbance factors as shown below. In Figure 81(a), the street light in the remapped image were too unapparent to be detected because its texture was similar to that of the grassland behind it. In addition, the completeness of obstacle shape in the profile or temporal FLIPM difference image would be critical for the following obstacle detection process. Figure 81(b) showed the broken shape of obstacles in the temporal FLIPM difference image, which could lead to the erroneous detected results.



(a)

(b)

Figure 81: Examples of erroneous detected results in our system

6. Conclusions

In this dissertation, we propose a complete and novel structure for the safety assistant telematics system containing safety vehicle assistant system and intersection surveillance system. In intersection surveillance system, we accomplish an image processing embedded platform, DSRC transferring and receiving functions, and an advanced surveillance system. The embedded platform captures the composite NTSC/PAL video stream, and processes intelligent functions via DSP dual core, and finally export the composite video stream with intelligent incident detection results. The DSRC protocol contents are verified by the sniffer card manufacture from active duty products. The advanced intelligent surveillance system contains incident detector and collector. Detector is a front-end equipment with our vision-based intelligent algorithm, therefore the adjustment of the detector can provide more related information to pedestrians and drivers to take attention intersection traffic. In the vehicle assistant system, the brand-new structure includes three major parts, FLIPM algorithm, obstacle detection and dynamic distance gauge. With our modified normal lens IPM equations, the vertical/horizontal straight lines in the original image will be projected to a straight line whose prolongation will pass the vertical/horizontal projection point of camera in the remapped image. The resultant phenomenon has two advantages in removing planar objects and detecting obstacles. One is to give more information in predicting the compensation quantification between difference frames, which helps us to remove the planar objects such like shadow, water, lane line and so on. The other one is to reinforce the feature points of obstacles, which efficiently reduces the computational loading in searching obstacles. Besides, we consider the fisheye lens distortion effect and provide a high efficient and all-direction feature searching method on polar histogram for both of the static and dynamic environments. We use the polar histogram to find the position and length of feature segments by referring to the edge and temporal difference images. We also present the histogram post-processing to exclude the planar lane markings and noises. In dynamic distance gauge system, we develop a vision-based curve estimation method with any vehicle machine information. We do not only accomplish the algorithm, but also develop is on the embedded platform. Finally, all the experimental results of our proposed system show the satisfactory performance and provide the accurate detection rate. Our experiment is also verified by the national supervision company. In the future, our advanced intelligent surveillance system can apply in ITS applications, such as vehicle detection, incident detection, traffic light management, e-police, presence rate estimation and so on. Beside, obstacle detection system can be integrated into the driving assistance and safety system, including vehicle collision-free development, warning system, and lane departure warning system. Furthermore, we will work on different shapes of obstacles for those without quasi-vertical edges and speed up our detection system for more real-time applications. We will also keep develop more efficiency safety assistant researches by following the DSRC newest information.

Bibliography

[1] http://en.wikipedia.org/wiki/Telematics

[2] http://www.calm.hu

[3] http://www.astm.org

[4] M. Bertozzi, A. Broggi, and A. Fascioli, “Stereo inverse perspective mapping: theory and applications,” Image and Vision Computing, vol. 16, no. 8, pp. 585–590, 1998.

[5] M. Bertozzi and A. Broggi, “GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection,” IEEE Transactions on Image Processing, vol. 7, no. 1, pp. 62–81, 1998.

[6] W.-L. Ji, “A CCD-based intelligent driver assistance system-based on lane and vehicle tracking”, Ph.D. thesis, National Cheng Kung University, Tainan, Taiwan, 2005.

[7] P. Cerri and P. Grisleri, “Free space detection on highways using time correlation between stabilized sub-pixel precision IPM images,” in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA ’05), pp. 2223–2228, Barcelona ,Spain, April 2005.

[8] A. M. Muad et al., “Implementation of inverse perspective mapping algorithm for the development of an automatic lane tracking system,” in Proceedings of the IEEE Region 10 Conference on Analog and Digital Techniques in Electrical Engineering (TENCON ’04), vol. 1, pp. 207–210, Chiang Mai, Thailand, November 2004.

[9] S. Tan et al., “Inverse perspective mapping and optic flow: a calibration method and a quantitative analysis,” Image and Vision Computing, vol. 24, no. 2, pp. 153–165, 2006.

[10] G. Y. Jiang et al., “Lane and obstacle detection based on fast inverse perspective mapping algorithm,” in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pp. 2969–2974, Nashville, Tenn, USA, October 2000.

[11] M. Nieto et al., “Stabilization of inverse perspective mapping images based on robust vanishing point estimation,” in Proceedings of the IEEE Intelligent Vehicles Symposium, pp. 315–320, Istanbul, Turkey, June 2007.

[12] J.-H. Lai, Development of an exploration system for a vision guided mobile robot in an unknown indoor environment, M.S. thesis, St. John’s University, 2006.

[13] C. Curio, J. Edelbrunner, T. Kalinke, C. Tzomakas, andW. Von Seelen, “Walking pedestrian recognition,” IEEE Transactions on Intelligent Transportation Systems, vol. 1, no. 3, pp. 155–162, 2000.

[14] M. Bertozzi et al., “Stereo Vision-based approaches for Pedestrian Detection,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’05), vol. 2005, p. 16, San Diego, Calif, USA, 2005.

[15] Z. Sun, G. Bebis, and R. Miller, “On-road vehicle detection using optical sensors: a review,” in Proceedings of the 7^{th} International IEEE Conference on Intelligent Transportation Systems (ITSC ’04), pp. 585–590, October 2004.

[16] S. Kyo et al., “Robust vehicle detecting and tracking system for wet weather conditions using the IMAP-VISION image processing board,” in Proceedings of the IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems, pp. 423–428, Tokyo, Japan, October 1999.

[17] S. Denasi and G. Quaglia, “Obstacle detection using a deformable model of vehicles,” in Proceedings of the IEEE Intelligent Vehicles Symposium (IV ’01), pp. 145–150, Tokyo, Japan, 2001.

[18] W. Krueger, W. Enkelmann, and S. Roessle, “Real-time estimation and tracking of optical flow vectors for obstacle detection,” in Proceedings of the Intelligent Vehicles Symposium, pp. 304–309, Detroit, Mich, USA, September 1995.

[19] C. H. Q. Forster and C. Tozzi, “Towards 3D reconstruction of endoscope images using shape fromshading,” in Proceedings of the 13th Brazilian Symposium on Computer Graphics and Image Processing, pp. 90–96, 2000.

[20] C. Yang, H. Hongo, and S. Tanimoto, “A new approach for in-vehicle camera obstacle detection by ground movement compensation,” in Processing of the 11th IEEE Conference on Intelligent Transportation Systems (ITSC ’08), pp. 151–156, Beijing, China, October 2008.

[21] G. Ma et al., “Pedestrian detection using a single-monochrome camera,” IET Intelligent Transport Systems, vol. 3, no. 1, pp. 42–56, 2009.

[22] W. Hu et al., “A survey on visual surveillance of object motion and behaviors,” IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews, vol. 34, no. 3, pp. 334 – 352, Aug. 2004.

[23] S. Gupte et al., “Detection and classification of vehicles,” IEEE Transactions on Intelligent Transportation Systems, vol. 3, no. 1, pp. 37–47, Mar. 2002.

[24] D. Koller et al., “Towards robust automatic traffic scene analysis in real-time,” in Proc. of the 12th IAPR International Conference on Pattern Recognition, vol. 1, 1994, pp. 126 – 131.

[25] A. Chachich et al., “Traffic sensor using a color vision method,” in Proc. of SPIE:

Transportation Sensors and Controls: Collision Avoidance, Traffic Management, and ITS, vol. 2902, pp. 156–165, 1996.

[26] T. N. Tan, G. D. Sullivan, and K. D. Baker, “Model-based localization and recognition of road vehicles,” International Journal of Computer Vision, vol. 27, no. 1, pp. 5–25, 1998.

[27] S. McKenna et al., “Tracking groups of people,” Computer Vision and Image Understanding, vol. 80, no. 1, pp. 42–56, 2000.

[28] R. Cucchiara, P. Mello, and M. Piccaidi, “Image analysis and rule-based reasoning for a traffic monitoring system,” IEEE Transactions on Intelligent Transportation Systems, vol. 1, no. 2, pp. 119–130, June 2000.

[29] H. Veeraraghavan, O. Masoud, and N. P. Papanikolopoulos, “Computer vision algorithms for intersection monitoring,” IEEE Transactions on Intelligent Transportation Systems, vol. 4, no. 2, pp. 78 - 89, June 2003.

[30] S. C. Chen et al., “Learning-based spatio-temporal vehicle tracking and indexing for transportation multimedia database systems,” IEEE Transactions on Intelligent Transportation Systems, vol. 4, no. 3, pp. 154 – 167, Sep. 2003.

[31] P. Kumar et al., “Co-operative multi-target tracking and classification,” in Proc. of European Conference on Computer Vision, May 2004, pp. 376–389.

[32] S. K. Zhou, R. Chellappa, and B. Moghaddam, “Visual tracking and recognition using appearance-adaptive models in particle filters,” IEEE Transactions on Image Processing, vol. 13, no. 11, pp. 1491 – 1506, Nov. 2004.

[33] H. T. Nguyen and A. W. M. Smeulders, “Fast occluded object tracking by a robust appearance filter,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 8, pp. 1099 – 1104, Aug. 2004.

[34] J. Kang, I. Cohen, and G. Medioni, “Continuous tracking within and across camera streams,” in Proc. of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, June 2003, pp. I-267 - I-272.