



Road scene perception based on fisheye camera, LIDAR and GPS data combination

Yong Fang

► To cite this version:

Yong Fang. Road scene perception based on fisheye camera, LIDAR and GPS data combination. Artificial Intelligence [cs.AI]. Université de Technologie de Belfort-Montbéliard, 2015. English. NNT : 2015BELF0265 . tel-01625515

HAL Id: tel-01625515

<https://theses.hal.science/tel-01625515>

Submitted on 27 Oct 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.


SPIM

Thèse de Doctorat



école doctorale sciences pour l'ingénieur et microtechniques
UNIVERSITÉ DE TECHNOLOGIE BELFORT-MONTBÉLIARD

Road scene perception based on fisheye camera, LIDAR and
GPS data combination

 YONG FANG

SPIM

Thèse de Doctorat



école doctorale sciences pour l'ingénieur et microtechniques
UNIVERSITÉ DE TECHNOLOGIE BELFORT-MONTBÉLIARD

N° X | X | X |

THÈSE présentée par

YONG FANG

pour obtenir le

Grade de Docteur de

l'Université de Technologie de Belfort-Montbéliard

Spécialité : **Informatique**

Road scene perception based on fisheye camera, LIDAR and GPS data
combination

Unité de Recherche :

Institut de Recherche sur les Transports, l'Énergie et la Société (IRTES)

Soutenue publiquement le 24 September 2015 devant le Jury composé de :

JEAN-CHARLES NOYER	Rapporteur	Professeur à Université du Littoral Côte d'Opale
VINCENT FREMONT	Rapporteur	Maître de Conférences HDR à Université de Technologie de Compiègne
MAAN EL BADAQUI EL NAJJAR	Examineur	Professeur à Université de Lille
CYRIL MEURIE	Examineur	Charge de recherche à l'IFSTTAR
CINDY CAPPELLE	Co-encadrante	Maître de Conférences à Université de Technologie de Belfort-Montbéliard
YASSINE RUICHEK	Directeur de these	Professeur à Université de Technologie de Belfort-Montbéliard

Contents

1	Introduction	5
1.1	Background	5
1.2	Problem Statements and Objectives	7
1.3	Thesis Organization	8
2	Extrinsic calibration between fisheye camera and laser range finder	11
2.1	Introduction	11
2.2	State of the Art	12
2.3	Fisheye Camera Modeling	14
2.3.1	Scaramuzza’s representation for fisheye camera	14
2.3.2	Kannala’s representation for fisheye camera	17
2.3.3	Mei’s representation for fisheye camera	19
2.4	Proposed Method for Extrinsic Calibration Between Fisheye Camera and LRF	22
2.4.1	Known points estimation	23
2.4.2	Normal vector of laser plane estimation	24
2.4.3	Geometrical constraints construction	29
2.5	Experiments	31
2.5.1	Simulation tests	31
2.5.2	Real data experiments	34
2.5.3	Application in ICP algorithm	38
2.6	Conclusion and Future Works	41

3	Road Detection Based on Fisheye Camera and Laser Range Finder	43
3.1	Introduction	43
3.2	State of the Art	44
3.3	Framework Overview	46
3.4	Coarse Road Detection Based on Illumination Invariant Image	48
3.4.1	1 dimensional illumination invariant images	48
3.4.2	Classification based on illumination invariant image histogram	52
3.5	Road Detection Refinement Using LRF Measurements	54
3.5.1	Coherence checking between coarse road image and LRF mea- surements	54
3.5.2	Refined road detection procedure	55
3.6	Experiment	62
3.6.1	Setup	62
3.6.2	Experimental results	63
3.7	Conclusion and Future Works	66
4	Multisensor Based Obstacles Detection in Challenging Scenes	67
4.1	Introduction	67
4.2	State of the Art	68
4.3	Overview of the Proposed Algorithm	70
4.4	Possible Region of Obstacle Presence Extraction	71
4.4.1	Inverse perspective mapping	72
4.4.2	Central line marker detection	74
4.4.3	Road shape model computation	77
4.4.4	Road model mapping	82
4.4.5	Potential obstacle detection	82
4.5	LRF based Obstacles Confirmation	83
4.6	Experiments	85
4.7	Conclusion and Future Works	90

5	Objects tracking based on small-region growth	93
5.1	Introduction	93
5.2	State of the Art	94
5.3	Overview of the Proposed Algorithm	96
5.4	Framework overview	96
5.5	Initialization	97
5.6	Frame Information Buffer	99
5.7	Object Representation	99
5.8	Estimation of Small Region	100
5.9	Growth Strategy	101
5.10	Experiment	104
5.10.1	Test results in real scenarios	104
5.10.2	Comparison results	108
5.11	Conclusion and Future Works	109
6	Conclusion and Future Works	113
6.1	Conclusion	113
6.2	Future Works	114
A	Transformation from WGS84 to Extended Lambert II	123
B	Abstract	139

Chapter 1

Introduction

1.1 Background

The first idea of intelligent vehicle was born in 1960s [1]. However, the level of technique at that time didn't allow people to make that dream come true. But, during the past two decades, intelligent vehicle has achieved great growth with the improvement of sensor techniques. In 1990s, Bundeswehr University Munich tested a vehicle with a 1758 km trip from Munich to Copenhagen in Denmark and back. In 95% of the trip, the vehicle was in a autonomously running state. In the early of this century, the research in intelligent vehicle is gradually developed into the test of performance in realistic scenarios. The ideal of advanced driving assistance systems (ADAS) is proposed by many researchers. It means that intelligent vehicle is still controlled by the driver, but there exists a monitoring system that detect possible dangerous situations to provide warning to the driver or take in charge the vehicle in emergency case. In 2003, the Defense Advanced Research Projects Agency (DARPA) launched a race named by Grand Challenge for autonomous vehicles. All the participants are demanded to autonomously run more than 200 km in unstructured environments. This challenge attracted many top-level research institutes (See Fig.1-1). Recently, several important achievements have taken in the intelligent vehicle community. In August of 2012, google company announced that its self-driving cars had completed over 300,000 miles with no accidents. And, in the same year, the states of Nevada, Florida and



(a) Carnegie Mellon University ¹

(b) The Stanford Racing Team ²

Figure 1-1: Example of participants in 2003 Grand Challenge

California in USA passed the law permitting driverless cars. In 2014, google released the new version self-driving of its cars. Fig.1-2 shows the new prototype of google's driverless cars.



Figure 1-2: Google new prototype driverless car ³

¹http://archive.darpa.mil/grandchallenge/images/Team_Pics/TartanRacing_3.jpg

²<http://archive.darpa.mil/grandchallenge/Teams/stanfordracing.html>

³http://www.dotnews.com/focus/connected-cars-the-latest-motoring-innovations#close_subscription

1.2 Problem Statements and Objectives

Sensing the environment around a vehicle is a crucial ability for intelligent vehicle application. The most important work is to understand road scene. The commonly used sensors to scan the environment are 2D/3D Laser Ranger Finders (LRFs), RADARs, cameras (monocular, stereo vision, fisheye and omnidirectional) and GPS. Both LRFs and RADARs are active sensors. Compared to optical sensors, they have the advantages of long distance detection ability, wide sensing range, and robust performance in evening, and in foggy or rainy day. However, they usually lack of high spatial resolution. Cameras are passive sensors. Cameras can provide high spatial resolution information and visual feature. In certain road scene such as road signs and traffic lights, cameras can work merely. GPS can be used for the drawing map and giving location information. To obtain robust and good performance, as described in [2], sensor fusion is commonly used by researcher for intelligent vehicle. In this thesis, we focus on the usage of LRF, camera (fisheye) and GPS.

Our research is part of the project CPER "Intelligence du Véhicule Terrestre", conducted within IRTES-SET Lab of UTBM. It aims to develop a multisensors system to robustly and precisely analyze and represent the road scene. The followings are the main objectives and contributions in our works:

- 1.The first objective is to fuse sensors data. Our perception system is composed of a LRF and a fisheye camera. Determining the position relation between between LRF and camera is an important work for further research. Extrinsic parameters calibration between LRF and fisheye camera refers to calculate the rigid transformation between their coordinate systems. In this work, a new calibration method is proposed. Meanwhile, three known fisheye model are tested and evaluated.

- 2.The second objective is to detect road in outdoor scenarios. Perceiving the presence of road is a crucial ability for intelligent vehicle. This task is mainly conducted by camera sensor because the equipped LRF is 2D. In this thesis, a road detection method based on illuminance invariant space and HSI space is proposed. This method

mainly addresses over-saturated or under saturated issues that occur in cloudy days.

3.The third objective is to detect obstacles in front of running vehicle. Security issues for intelligent vehicle always can attract the attention from most of people. Detecting robustly obstacles around the vehicle can effectively prevent us from collisions. We propose a new method based on LRF, GPS and camera for obstacle detection in the thesis. Our contribution consist in using features obtained from map to overcome the difficulty related to visual features which is invalid in motion blur case.

4.The final objective is to track objects. Tracking objects can give us prior knowledge regarding objects movements. This knowledge is a favor to prejudge the motion of objects in following time. We combine image space and LRF space to conduct object tracking. Our contribution consists in using weak visual features to perform tracking without losing objects in a long-term test.

1.3 Thesis Organization

The rest of the thesis is divided into five chapters:

Chapter2: In this chapter, we present a method for extrinsic calibration between fisheye camera and LRF. It is mainly based on the LRF scan plane and several LRF measurements. The method is tested and evaluated by simulated data and tested by real data.

Chapter3: In this chapter, a camera and LRF based road detection approach is presented. It conducts a preliminary road detection in illuminance invariant image and then refines the results in HSI space.

Chapter4: In this chapter, we proposed a method based on GPS, LRF and camera for obstacle detection. It mainly copes with the problems caused by motion blur which stems from object and camera movements. The method is evaluated by the ground truth data.

Chapter5: In this chapter, we propose an approach for object tracking. It is mainly composed of two steps: the extraction of small region in image and the small

region growth in LRF space.

Chapter6: In this chapter, conclusions and some research perspectives for this thesis are presented.

Chapter 2

Extrinsic calibration between fisheye camera and laser range finder

2.1 Introduction

This chapter deals with the issue of extrinsic calibration between fisheye camera and laser range finder. The aim of extrinsic calibration is to determine the rigid transformation between LRF and fisheye camera. In the field of intelligent vehicle, the road environment is often perceived by various sensors, such as video cameras, laser rang finder. However, each sensor has some weaknesses. In a complex traffic environment, using only a single sensor to perceive and analyze the environment could limit the accuracy and robustness. Recently, LRF and camera mounted together on a car have become very common for perceiving the environment. On the one hand, LRF measure distances between itself and the detected objects with high accuracy within wide-area view but with low resolution. On the other hand, camera provides visual information around itself. These visual information provide many important clues for applications such as object recognition, obstacle detection. However, a conventional camera suffers from narrow field of view. Multiple conventional cameras can be employed to expand the perception area, but it will make extra expense and cause some real-time processing problems when the vehicle is traveling with high speed. Compared with conventional camera, camera with fisheye lens is then an

attractive choice in many applications, as it provides a large field of view (FOV) with a single sensor. Due to these reasons, fisheye camera and 2D LRF are combined in our research. The used experimental platform is shown in Fig.2-1. Before performing road scene understanding, the rigid transformation between LRF and fisheye camera has to be known.

The rest of this chapter is organized as follows. Section 2.2 introduces the state of the art. Section 2.3 describes three different kinds of fisheye camera models. Section 2.4 presents LRF plane based extrinsic calibration between LRF and fisheye camera. Section 2.5 deals with results and evaluation of the simulation tests and real data based experiments. Finally, a conclusion ends the chapter.



Figure 2-1: The experimental platform. The fisheye camera is on the top bracket and the LRF is fixed in front of the vehicle

2.2 State of the Art

Most of state of the art works about calibration between LRF and camera fall into two categories depending on the type of LRF: 2D LRF or 3D LRF. Paper [3] suggests a new method for calibration between 3D LRF and a stereo camera. The considered 3D LRF is built by moving a 2D LRF along one of its axes. By rotating the 2D scanner around its radial axes, it is possible to obtain the spherical coordinates of the points measured. The method firstly computes the three transformation matrices

from LRF reference frame to stereo camera reference frame when the LRF is placed in three different specified orientations around its rotation axis. Then, based on the obtained three transformation matrices, the rigid transformation matrix between the LRF and the center of rotation of LRF is determined. Finally, the transformation matrix between LRF in any arbitrary angle and the stereo-camera is computed. This method needs to put pattern into several different poses. In paper [4], the authors propose a self calibration method between a 3D LRF and an omnidirectional camera. Based on the 3D LRF measurements, a depth image named bearing angle (BA) image is constructed. Several corresponding points between BA image and intensity image are selected manually. With these points, the extrinsic parameters between the two sensors frames are estimated. However, the accuracy of this method is depending on the resolution of the 3D LRF. High resolution of 3D LRF often means a high cost. In paper [5], a visible LRF is used to estimate transformation between both LRF and camera. However, in many applications, laser spots emitted by LRF are often invisible for camera.

Paper [6] introduces a method based on distance constraints from camera to 2D LRF system. A chessboard is placed in the common field of view (FOV) of the two sensors. A set of geometrical constraints on rigid transformation between LRF and camera can be defined by changing the chessboard location in the common FOV. Based on this set of geometrical constraints, the extrinsic parameters can be determined. This method needs to change the chessboard location and is affected by the orientation between the chessboard and LRF. In paper [7], the authors provide an approach based on 2D LRF with both visible and invisible trace. This method is an extended version of the approach in paper [6]. It also needs to put a chessboard placed in the FOV of the two sensors. The constraint conditions are not based on the laser points directly but the straight lines consisting of them. In paper [8], the authors present an approach for the calibration between a fisheye camera and laser range finder. The main idea is to justly use regular lens to get extrinsic parameters and then to replace it with fisheye lens. In paper [9], the authors propose a minimal approach to determine the extrinsic parameters between a 2D laser scanner and a

camera, using only six measurements of a planar calibration board.

The work in this chapter also focuses on extrinsic calibration between 2D LRF and fisheye camera. There are two contributions in this research. The first one is the evaluation and comparison between three different models that have been proposed to fisheye camera modeling. The second one is the proposition of a novel approach of extrinsic calibration between the two sensors. The proposed method requires LRF and camera to observe a chessboard moved in their common field of view. By analyzing successive LRF measurements, a set of points located in the laser beams plane can be detected. These detected points are then used to estimate the equation of the plane of the laser beams in the camera coordinate system. Finally, two geometrical constraints based on the equation of this plane and this set of points are constructed to estimate the extrinsic parameters between the fisheye camera and the LRF. The performance of the approach is evaluated through experiments on both simulated and real data.

2.3 Fisheye Camera Modeling

Camera with fisheye lens provides a wide angle vision, but cause great distortions in the image. It makes then the conventional camera model invalid. Up to now, there is no unified projection model for fisheye camera. In this chapter, three models proposed by Scaramuzza [10], Kannala [11] and Mei [12], are studied.

2.3.1 Scaramuzza's representation for fisheye camera

The fisheye model proposed in paper [10] is illustrated in Fig.2-2. It consists of fisheye lens surface, image plane and sensor plane. The sensor plane is a hypothetical plane orthogonal to the mirror axis, with the origin lying on the plane-axis intersection. In practice, the sensor plane corresponds to the camera CCD plane, where the pixels are stated in physical size. The image plane corresponds to the image, where the pixels are expressed in pixel coordinates. The image point M corresponding to the scene point P_0 is produced by three steps. Firstly, P_0 is mapped onto fisheye lens surface

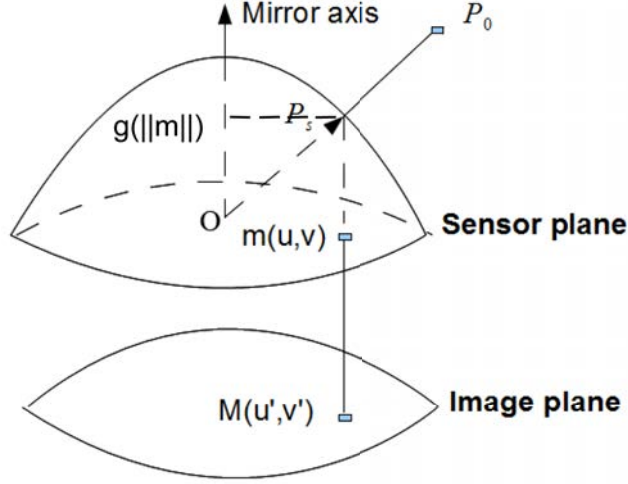


Figure 2-2: Fisheye model

as point P_s . Then, P_s is mapped onto sensor plane as the point m . Finally, the point m is mapped onto image plane as the point M . From the mapping procedure, we can see that each point in image plane has one unique corresponding point on fisheye lens surface. In other words, any point in image can be represented by a unique vector from the center of sensor plane to fisheye lens surface. Let \mathbf{P}_s denote (see Fig.2-2) the vector corresponding to the image point M , so we have:

$$\mathbf{P}_s = f(m) = \begin{bmatrix} m \\ g(||m||) \end{bmatrix} = \begin{bmatrix} u \\ v \\ g(||m||) \end{bmatrix} \quad (2.1)$$

where $g(||m||)$ represents the corresponding point of M on fisheye lens surface, $||m||$ is the module of point m on sensor plane. In paper [10], the authors consider that the function g represent the curve line of fisheye lens surface and obeys the following polynomial form:

$$g(||m||) = a_0||m||^1 + a_1||m||^2 + \cdots + a_N||m||^N \quad (2.2)$$

where $a_i, i = 0, 1, 2, \cdots, N$ are coefficients estimated by calibration.

Equation 2.1 refers to the sensor plane. It needs to be transformed to image plane.

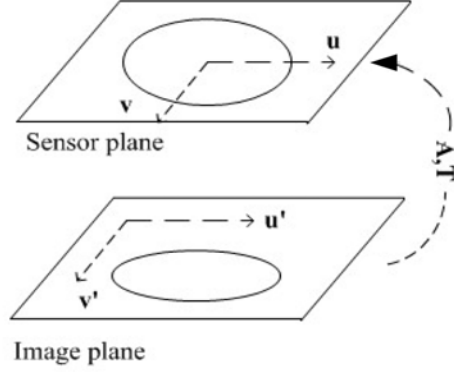


Figure 2-3: Affine transformation between the sensor plane and the image plane

The image plane and the sensor plane are linked by an affine transformation shown in Fig.2-3. Given $m = [u, v]$ and $M = [u', v']$, the affine transformation between the two points is as follows:

$$\begin{aligned}
 m = \begin{bmatrix} u \\ v \end{bmatrix} &= \mathbf{A} \begin{bmatrix} u' \\ v' \end{bmatrix} + \mathbf{T} \\
 &= \mathbf{A}M + \mathbf{T}
 \end{aligned} \tag{2.3}$$

where $\mathbf{A} \in \mathbf{R}^{2 \times 2}$ and $\mathbf{T} \in \mathbf{R}^2$. \mathbf{A} is a stretch matrix and \mathbf{T} is a translation vector. Based on $g(\|m\|)$, \mathbf{A} and \mathbf{T} , the complete model of fish-eye camera is represented as follows:

$$\begin{aligned}
 \lambda \cdot \mathbf{P}\mathbf{s} = \lambda \cdot f(m) &= \lambda \cdot f(\mathbf{A}M + \mathbf{T}) \\
 &= \lambda \cdot \begin{bmatrix} \mathbf{A}M + \mathbf{T} \\ g(\|\mathbf{A}M + \mathbf{T}\|) \end{bmatrix}, \lambda > 0
 \end{aligned} \tag{2.4}$$

Fisheye camera calibration corresponds to the estimation of A , T and $a_i, i = 0, 1, 2, \dots, N$. Although these intrinsic parameters are not determined directly, they can be estimated with the help of a calibration pattern. There are three steps for estimating these intrinsic parameters. Firstly, assuming $A = I$ (unit matrix) and $T = 0$, the extrinsic parameters between the calibration pattern and the camera are estimated. Secondly, the coefficients of the function g are calculated by incorporating several observations

of the calibration pattern. Finally, non-linear refinement is used for all parameters to obtain more accurate results. For more details, the reader can refer to paper [10].

2.3.2 Kannala's representation for fisheye camera

In paper [11], the authors propose to divide the full fisheye camera projection model into two parts: the radially symmetric part and the asymmetric part.

Radially symmetric part

The radially symmetric part of fisheye camera projection model is illustrated in Fig.2-4. P is a scene point, p is the image of P in fisheye image, p' is the image of P in pinhole

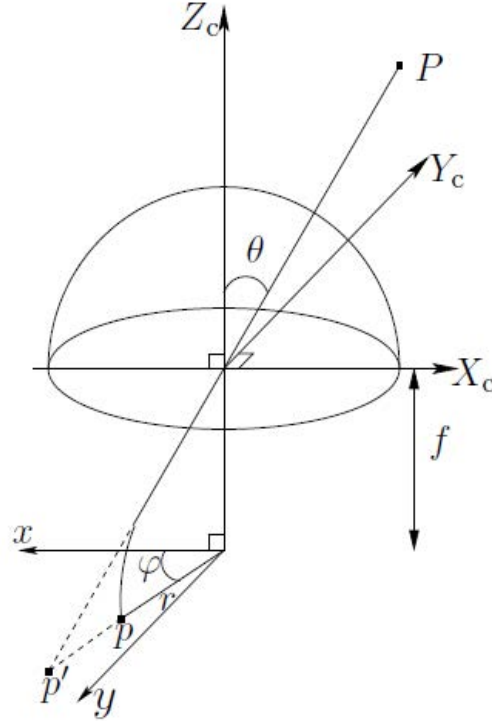


Figure 2-4: Fisheye camera projection model in paper [11]. The image of the point P is p whereas it would be p' by a pinhole camera

image, θ is the angle between principal axis and the incoming ray, r is the distance between the fisheye image point p and the principal point and f is the focal length. As illustrated in Fig.2-4, the radially symmetric part of fisheye camera projection model

is different from the pinhole camera model. In Generally, it is designed to obey one of the following projections:

$$\begin{aligned}
r &= 2f \tan(\theta/2) && (\textit{stereographic projection}) \\
r &= f\theta && (\textit{equidistance projection}) \\
r &= 2f \sin(\theta/2) && (\textit{equisolid angle projection}) \\
r &= f \sin(\theta) && (\textit{orthogon alangle projection}) \quad (2.5)
\end{aligned}$$

To facilitate the calibration work, the authors of paper [11] propose to use a general model suitable for different types of lens. The proposed general form is:

$$r(\theta) = k_1\theta + k_2\theta^3 + k_3\theta^5 + k_4\theta^7 + k_5\theta^9 + \dots \quad (2.6)$$

where $k_1, k_2, k_3, k_4, k_5, \dots$ are the model parameters. For computation consideration, the authors assume that the first five terms have enough degrees of freedom for good approximation of different projection curves. Thus, the radially symmetric part of the projection model is defined by the five parameters k_1, k_2, k_3, k_4 , and k_5 , that should be estimated.

Asymmetric part

In practice, real lens may deviate from precise radial symmetry and therefore an asymmetric part is added to obtain a full projection model. For wide application, the authors propose a flexible mathematical distortion model to represent the asymmetric part. This flexible distortion model consists of two terms. One term acts in the radial direction and has the following form:

$$\Delta_r(\theta, \varphi) = (l_1\theta + l_2\theta^3 + l_3\theta^5)(i_1\cos\varphi + i_2\sin\varphi + i_3\cos 2\varphi + i_4\sin 2\varphi) \quad (2.7)$$

and the other term is concerned with the tangential direction:

$$\Delta_t(\theta, \varphi) = (m_1\theta + m_2\theta^3 + m_3\theta^5)(j_1\cos\varphi + j_2\sin\varphi + j_3\cos 2\varphi + j_4\sin 2\varphi) \quad (2.8)$$

Full model

The full fisheye camera projection model is made up of radially symmetric part and asymmetric part. By combining 2.8, 2.7 and 2.6, an image point $\mathbf{X}_d = (x_d, y_d)$ in millimeter coordinate system of an image point can be represented as follows:

$$\mathbf{X}_d = r(\theta)\mathbf{e}_r(\varphi) + \Delta_r(\theta, \varphi)\mathbf{e}_r(\varphi) + \Delta_t(\theta, \varphi)\mathbf{e}_\varphi(\varphi) \quad (2.9)$$

where \mathbf{e}_r and \mathbf{e}_φ are the unit vectors in radial and tangential directions. To get the full projection model, millimeter coordinate system is needed for transformation into image pixel coordinate system. Let (u, v) be the pixel coordinate, (u_0, v_0) is the principal point, and n_u and n_v represent the number of pixels unit distance in horizontal and vertical directions respectively. Finally, the full model is written as follows:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{bmatrix} n_u & 0 \\ 0 & n_v \end{bmatrix} \begin{pmatrix} x_d \\ y_d \end{pmatrix} + \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} \quad (2.10)$$

This full model is defined by 23 parameters. To facilitate computation, the asymmetric part is often ignored. In this case, the full model has only 9 parameters and is denoted by p_9 in the following. To solve these parameters, several observation points on a calibration pattern are used. For more details, reader can refer to paper [11].

2.3.3 Mei's representation for fisheye camera

In paper [12], the authors use an unified projection model representing omnidirectional camera to approximate fisheye camera. It is based on the model proposed by Geyer and Barreto in papers [13] and [14], and is illustrated in Fig.2-5. This model contains three parts: a unit sphere, a sensor plane and an image plane. Let F_m represents the coordinate system centered at C_m (centre of the sphere) and F_p is the coordinate system centered at C_p .

A world point $\chi_{F_m} = (X, Y, Z)$ is projected onto the image plane \mathbf{p} using the

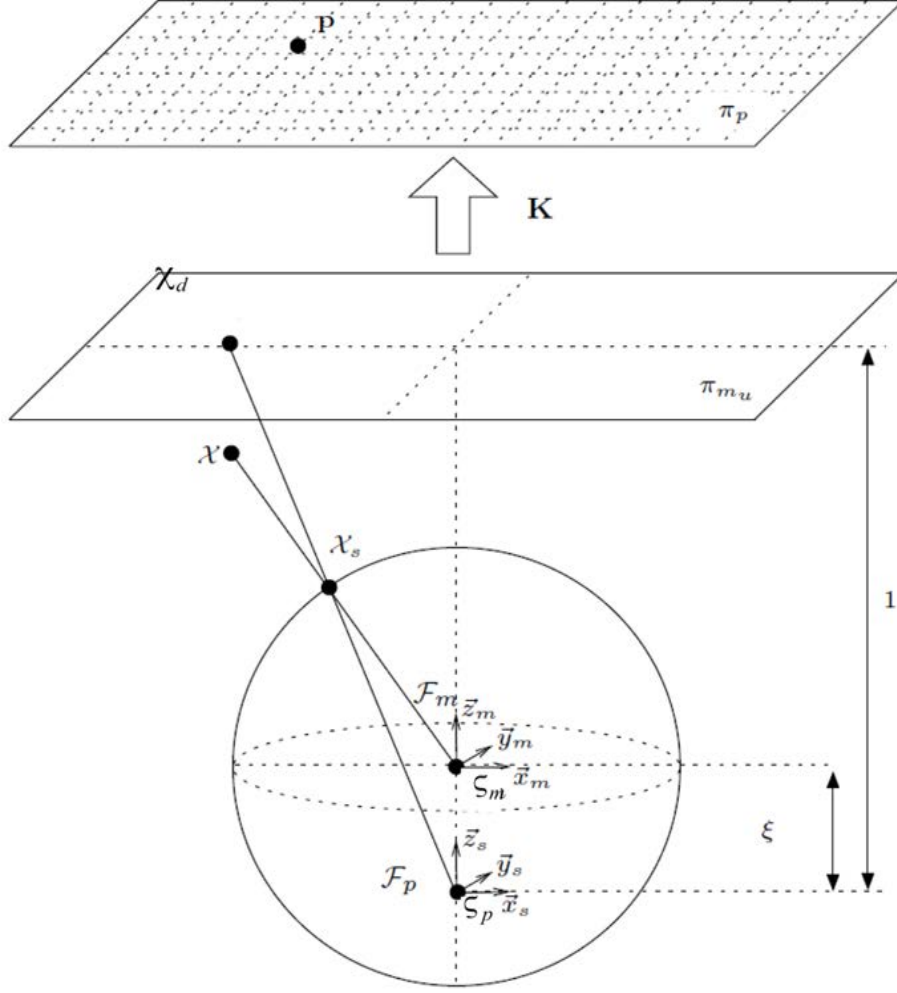


Figure 2-5: Unified projection model in paper [12]

following steps:

1. The world point χ is projected onto the unit sphere to obtain the point χ_s ,
 $(\chi)_{F_m} \longrightarrow (\chi_s)_{F_m} = \frac{\chi}{\|\chi\|} = (X_s, Y_s, Z_s)$
2. χ_{sF_m} is then converted to F_p system $\mathbf{C_p} = (0, 0, \xi)$, $(\chi_s)_{F_m} \longrightarrow (\chi_s)_{F_p} = (X_s, Y_s, Z_s + \xi)$. ξ is a parameter depending on the mirror type located at C_m
3. Then, the point (χ_s) is projected onto the sensor plane π_{mu} point χ_d , $(\chi_s)_{F_p} \longrightarrow (\chi_d)_{F_p} = (\frac{X_s}{Z_s + \xi}, \frac{Y_s}{Z_s + \xi}, 1)$

4. The final projection involves the camera intrinsic matrix \mathbf{K}

$$\chi_{\mathbf{d}} \longrightarrow \mathbf{p} = \mathbf{K}\chi_{\mathbf{d}} = \begin{bmatrix} a_x & a_x\alpha & u_0 \\ 0 & a_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \chi_{\mathbf{d}} \quad (2.11)$$

where (u_0, v_0) the principal point, α the skew, a_x and a_y the focal length of the camera in terms of pixel dimensions in the x and y direction respectively.

Approximation of fisheye lens

The great distinction between fisheye image and classic image is that there is great distortion along radial direction in fisheye image. In paper [15], the authors show that the great radial distortion in image can be approximated as a division model. In paper [16], the fisheye image can be considered as a conventional image but with great radial distortion. In other words, the point in fisheye image can be obtained by following two step. Firstly, a 3D point is mapped onto image plane (pinhole model). Then, a division model function is applied to this projected point to solve the distorted image point. Let $\mathbf{P}_{\mathbf{u}} = [x_u, y_u]$ be a point before distortion and $\mathbf{P}_{\mathbf{d}} = [x_d, y_d]$ the corresponding point after distortion. With $p_u = \sqrt{x_u^2 + y_u^2}$ and $p_d = \sqrt{x_d^2 + y_d^2}$, the division model is expressed as follows:

$$p_u = k_1 \frac{p_d}{1 - k_2 p_d^2}; \quad (2.12)$$

where k_1 and k_2 are two scalar parameters depending on the fisheye lens type. Given the focal length $f = 1$, the projected point χ of a word point on sensor plane under pinhole model can be written as:

$$\chi = (x_\chi, y_\chi, 1) = (X/Z, Y/Z, 1) \quad (2.13)$$

where $(x, y, 1)$ are the homogeneous coordinates of the word point on the sensor plane. In C.Mei's representation, with $\xi = 1$, the projected point χ_d of this word point is:

$$\chi_d = (x_{\chi_d}, y_{\chi_d}, 1) = \left(\frac{X_s}{Z_s + 1}, \frac{Y_s}{Z_s + 1}, 1 \right) = \left(\frac{X}{Z + \|\chi\|}, \frac{Y}{Z + \|\chi\|}, 1 \right) \quad (2.14)$$

By algebraic manipulation, we obtain the following relation from equations 2.13 and 2.14:

$$p_\chi = \frac{2p_{\chi_d}}{1 - p_{\chi_d}^2} \quad (2.15)$$

where $p_\chi = \sqrt{x_\chi^2 + y_\chi^2}$ and $p_{\chi_d} = \sqrt{x_{\chi_d}^2 + y_{\chi_d}^2}$. Equation 2.15 has the same form as equation 2.12. (The influence of the mentioned three models on the calibration will be discussed in experiment section of this chapter.)

2.4 Proposed Method for Extrinsic Calibration Between Fisheye Camera and LRF

The objective of extrinsic calibration between fisheye camera and LRF is to determine the geometric transformation between the two sensor systems.

As illustrated in Fig.2-6, given a point P , this transformation is expressed by:

$$P_c = R * P_L + T \quad (2.16)$$

where P_c represents the coordinates of P in the camera system, P_L represents the coordinates of P in the LRF system, R and T are the rotation matrix and translation vector between the camera system and LRF system respectively. The elements of R and T have to be estimated by calibration.

The framework of the proposed approach is shown in Fig.2-7. Firstly, several known points (defined in the following section) are determined (section 2.4.1). Then, the normal vector of laser plane is estimated (section 2.4.2). Two geometrical

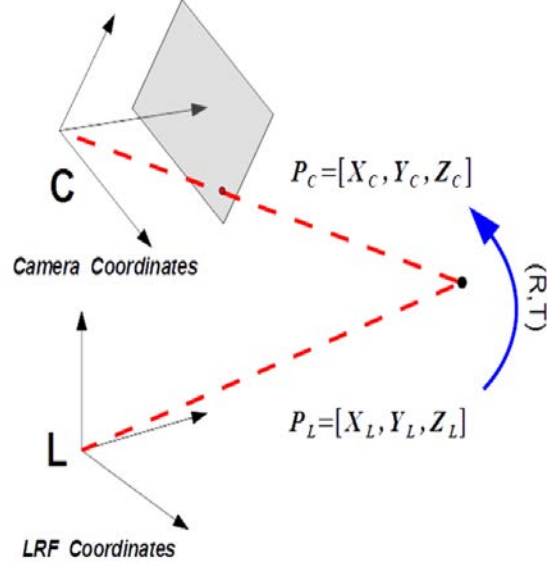


Figure 2-6: Extrinsic calibration between the two sensors

constraints are constructed based on this computed normal vector, and the extrinsic parameters between the two sensors are estimated using the Levenberg-Marquardt algorithm. (section 2.4.3)

2.4.1 Known points estimation

A known point in our method is a point for which its coordinates in LRF coordinate systems and camera coordinate systems are determined. In paper [3], a laser pointer is used to determine the known points for a system of a stereo camera and LRF. Spurring by this idea, a method based on the corner point of a chessboard is proposed to find out known points for our system composed of a fisheye camera and LRF. The procedure is illustrated in Fig.2-8. A chessboard is moved down slowly from a position above the plane of the laser beams in the field of view of the LRF. During the way down, the LRF measurements are checked continuously. When the chessboard is above the laser beams plane, the LRF measurements remain almost constant and can be considered as the background measurements, as we consider that there is no other moving object in the scene. When the chessboard intersects with the laser beams

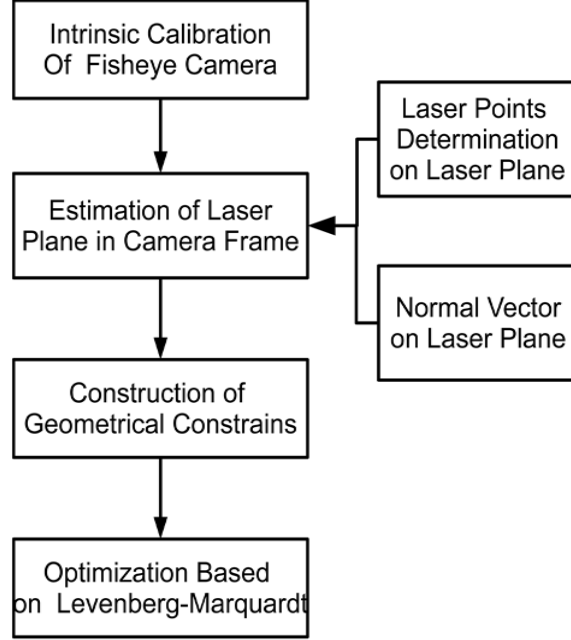


Figure 2-7: Framework of the proposed extrinsic calibration approach

plane, the LRF measurements become different from the background measurements. So when the first difference between the LRF measurements and the background measurements is detected during the way down, the corner point (red point in Fig.2-8) of the chessboard is believed to be intersecting with the laser beams plane, that is, being on the laser beams plane. The corner point at this position can then be considered as a known point. Its coordinates in LRF system can be obtained directly from LRF measurements and the coordinate in camera system can be calculated using the calibration toolbox introduced in [10]. This procedure is repeated until several known points are obtained.

2.4.2 Normal vector of laser plane estimation

It is known that three points which don't stand on a line can define a plane uniquely in space. To find out the coordinates of normal vector of laser plane in camera coordinate system, it is then necessary to know the coordinates of three points at least on laser plane in camera coordinate system. Fortunately, based on the method described in

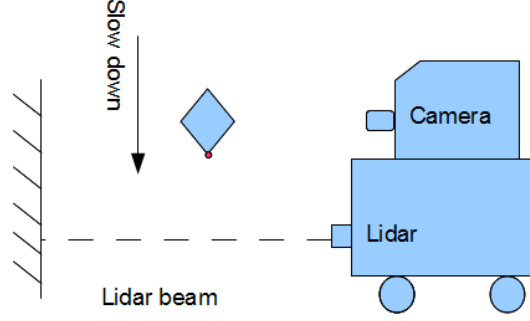


Figure 2-8: Corresponding points determination methodology

the previous step, the obtained known points are the proper ones for estimating the coordinate of normal vector of laser plane in camera coordinate system. However, there may exist several outliers in obtained known points. These outliers may cause error estimations when using linear regression method to calculate the normal vector of laser plane. So, the first job to us is to remove the outliers before solving laser plane.

Random sample and consensus

RANSAC (Random Sample and Consensus) algorithm was first introduced by [17]. It's a method to estimate parameter for a specific model with several outliers in input data. In paper [18], a datum is considered to be an outlier if it will not fit the "true" model instantiated by the "true" set of parameters within some error threshold that defines the maximum deviation attributable to the effect of noise. Generally, despite many improvement versions, the RANSAC algorithm is essentially composed of two steps that are repeated in an iterative fashion:

- 1) Hypothesize: First minimal sample sets (MSSs) are randomly selected from the input dataset and the model parameters are computed using only the elements of the MSSs.

- 2) Test: In the second step, RANSAC checks which elements of the entire dataset are consistent with the model instantiated with the parameters estimated in the first step. The set of such elements is called consensus set (CS). RANSAC terminates

when the probability of finding a better ranked CS drops below a certain threshold.

In our method, RANSAC is used to eliminate the outliers from known points set. Let denote $\mathbf{x}^i = (x_1^i, x_2^i, x_3^i)^T, (i = 1, 2, \dots, N)$ the i -th point set on plane. The used mathematical model is as follows:

$$\begin{cases} \tau_1 x_1^1 + \tau_2 x_2^1 + \tau_3 x_3^1 + \tau_4 = 0 \\ \vdots \\ \tau_1 x_1^N + \tau_2 x_2^N + \tau_3 x_3^N + \tau_4 = 0 \end{cases} \quad (2.17)$$

where $\tau_1, \tau_2, \tau_3, \tau_4$ are the model parameters. To group these equations in matrix form, we can get:

$$\begin{bmatrix} (\mathbf{x}^1)^T & 1 \\ \vdots & \vdots \\ (\mathbf{x}^N)^T & 1 \end{bmatrix} \tau = \mathbf{X}\tau = 0 \quad (2.18)$$

So the estimation of the parameters vector which instantiates the plane can be transformed as:

$$\tau^* = \underset{\|\tau=1\|}{\operatorname{argmin}} \|\mathbf{X}\tau\|^2 \quad (2.19)$$

For each data point \mathbf{x} , with above equation, the fitting algebraic error is defined as follows:

$$e_{\mathbf{x}} = \frac{([\mathbf{x}^T 1]\tau^*)^2}{\|\tau_{1:3}^*\|} \quad (2.20)$$

After obtaining of least three "true" known points, a least-square regression based approach is used to estimate the normal vector of laser plane in camera coordinate systems.

Multiple linear regression

Regression analysis is one of useful statistical tools for analyzing multifactor data. Its broad appeal and usefulness results from the conceptually logical process of using an equation to reveal the inherent relationship between variables. Similarly, it also can be regarded as an interesting theory due to elegant underlying mathematics and a well developed statistical theory. Successful use of regression requires an appreciation of both the theory and the practical problems that typically arise when the technique is employed with real-world data [19].

The simplest regression model is unary linear model, that is, a model with a single regressor x that has a relationship with a response y that is a straight line. Multiple linear regression which involves more than one regressor variable is the extension of the unary linear model. Plane normal vector estimation belongs to this case. However, the linear regression is not robust to noise. This is the reason why we have to use RANSAC algorithm to remove outliers firstly. In camera coordinate system, the plane equation can be expressed as follows:

$$N_x(X_i - X_0) + N_y(Y_i - Y_0) + N_z(Z_i - Z_0) = 0 \quad (2.21)$$

where $N_v = [N_x, N_y, N_z]$ is the plane normal vector, $P_i = [X_i, Y_i, Z_i]^T$ is any point on the plane and $P_0 = [X_0, Y_0, Z_0]^T$ a fixed point on the plane. In our case, the fixed point is regarded as the centroid point $P_C = [\bar{X}, \bar{Y}, \bar{Z}]^T$ of the known points set. For convenience, the equation 2.21 is rewritten as:

$$(P_i - P_C)N_v = 0 \quad (2.22)$$

The normal vector estimation problem can be converted the following minimization problem:

$$\operatorname{argmin} \sum_{i=1}^N d_i \quad (d_i = \|N_v(P_i - P_C)\|^2) \quad (2.23)$$

Under normalization constraints of N_v ($N_x^2 + N_y^2 + N_z^2 = 1$), to use lagrange multipliers, the above equation is transformed to:

$$d = \sum_{i=1}^N d_i - \lambda_v (N_x^2 + N_y^2 + N_z^2 - 1) \quad (2.24)$$

$$(2.25)$$

Given $\Delta X_i = X_i - \bar{X}$, $\Delta Y_i = Y_i - \bar{Y}$, $\Delta Z_i = Z_i - \bar{Z}$, we can get the partial derivative of d :

$$\begin{aligned} \frac{\partial d}{\partial N_x} &= 2 \sum_{i=1}^N (N_x \Delta X_i + N_y \Delta Y_i + N_z \Delta Z_i) \Delta X_i - 2\lambda_v N_x = 0 \\ \frac{\partial d}{\partial N_y} &= 2 \sum_{i=1}^N (N_x \Delta X_i + N_y \Delta Y_i + N_z \Delta Z_i) \Delta Y_i - 2\lambda_v N_y = 0 \\ \frac{\partial d}{\partial N_z} &= 2 \sum_{i=1}^N (N_x \Delta X_i + N_y \Delta Y_i + N_z \Delta Z_i) \Delta Z_i - 2\lambda_v N_z = 0 \end{aligned} \quad (2.26)$$

By grouping these equations into matrix form, we have:

$$\begin{bmatrix} \sum \Delta X_i \Delta X_i & \sum \Delta X_i \Delta Y_i & \sum \Delta X_i \Delta Z_i \\ \sum \Delta X_i \Delta Y_i & \sum \Delta Y_i \Delta Y_i & \sum \Delta Y_i \Delta Z_i \\ \sum \Delta X_i \Delta Z_i & \sum \Delta Y_i \Delta Z_i & \sum \Delta Z_i \Delta Z_i \end{bmatrix} \begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix} = \lambda_v \begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix} \quad (2.27)$$

Given the assumption $\|N_v\| = 1$, that is the inner product $(N_v, N_v) = 1$. Let

$$E = \begin{bmatrix} \sum \Delta X_i \Delta X_i & \sum \Delta X_i \Delta Y_i & \sum \Delta X_i \Delta Z_i \\ \sum \Delta X_i \Delta Y_i & \sum \Delta Y_i \Delta Y_i & \sum \Delta Y_i \Delta Z_i \\ \sum \Delta X_i \Delta Z_i & \sum \Delta Y_i \Delta Z_i & \sum \Delta Z_i \Delta Z_i \end{bmatrix} \quad (2.28)$$

We have:

$$EN_v = \lambda_v N_v$$

$$\Rightarrow (E - \lambda_v)N_v = 0 \quad (2.29)$$

$$(2.30)$$

N_v is the eigenvector corresponding to the minimum eigenvalue of the following equation:

$$|(E - \lambda_v)I| = 0 \quad (2.31)$$

where I is identity matrix.

2.4.3 Geometrical constraints construction

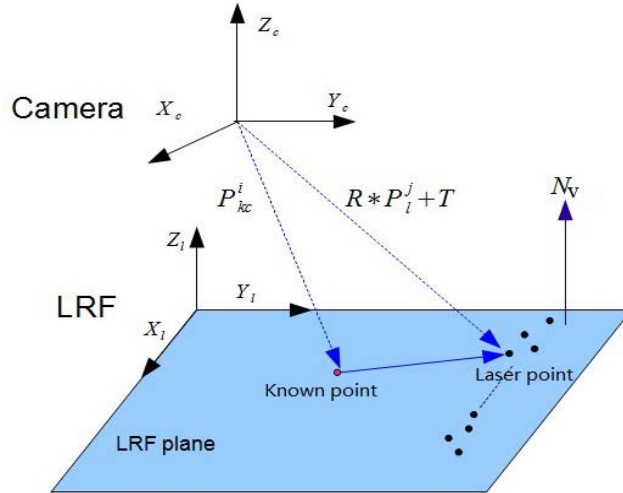


Figure 2-9: First constraint. The laser beams plane is colored by blue. In camera coordinate system, any vector on this plane is perpendicular to the vector N_v

The geometrical constraints reveal the LRF measurements relationship equations in 3D space. Based on previously obtained the known points set and normal vector, two geometrical constraints can be constructed. For one single-planar LRF, it is known that all laser beams lay on one plane. This characteristic is used to build

the first constraint condition. Let S_l denote the set of LRF measurements, the first constraint is:

$$N_v(R * P_l^j + T - P_{kc}^i) = 0 \quad (P_l^j \in S_l) \quad (2.32)$$

where P_l^j is the coordinates of the j-th laser point of S_l in LRF coordinate system, P_{kc}^i is the coordinates of the i-th known point in fisheye camera system. The geometrical interpretation is illustrated in Fig.2-9. In camera coordinate system, the vectors belonging to the laser beams plane are perpendicular to N_v .

The second constraint is based on the known points. From the previous statements, the coordinates of the "known points" are known in the two sensor coordinate systems. Let P_{kl}^i denote the coordinates of the i-th known point in LRF coordinate system. By taking P_{kl}^i and P_{kc}^i into equation 2.16, the second constraint is given as follows:

$$R * P_{kl}^i + T - P_{kc}^i = 0 \quad (2.33)$$

To summarize the above description, the two geometrical constraints are:

$$\begin{cases} N_v(R * P_l^j + T - P_{kc}^i) = 0 \\ R * P_{kl}^i + T - P_{kc}^i = 0 \end{cases} \quad (2.34)$$

To solve this equation, the following equations are proposed:

$$Min \begin{cases} W_1 \sum_{i=1}^N \sum_{j=1}^{M_p} N_v(R * P_l^j + T - P_{kc}^i) = 0 \\ W_2 \sum_{i=1}^{N_p} R * P_{kl}^i + T - P_{kc}^i = 0 \end{cases} \quad (2.35)$$

where M_p and N_p are the number of laser points and known points respectively. Levenberg-Marquardt [20] algorithm is applied to find the optimal solution to equation 2.35. However, in practice, the solution of R and T may not make the two equations to reach their minimum values simultaneously. For example, if M_p is far greater than N_p , the minimization process will mainly focus on the first equation. To

make a balance between the two constraints, two parameters W_1 and W_2 are used to adjust the influence of M_p and N_p on the minimization process.

2.5 Experiments

This section describes results obtained with simulated and real data. The program is implemented in Matlab. With real data experiment, it is hard to give performance estimation due to lack of ground-truth data. Therefore, simulated data are used firstly to estimate the performance of our method.

2.5.1 Simulation tests

The relative position and orientation of the LRF and fisheye camera are randomly set to $[10; 300; 200](mm)$ and $[75^\circ, -5^\circ, 5^\circ]$ respectively. The coordinates of the known points are calculated based on relative pose of the camera with respect to the LRF.

Influence of the known points number in ideal case (without noise)

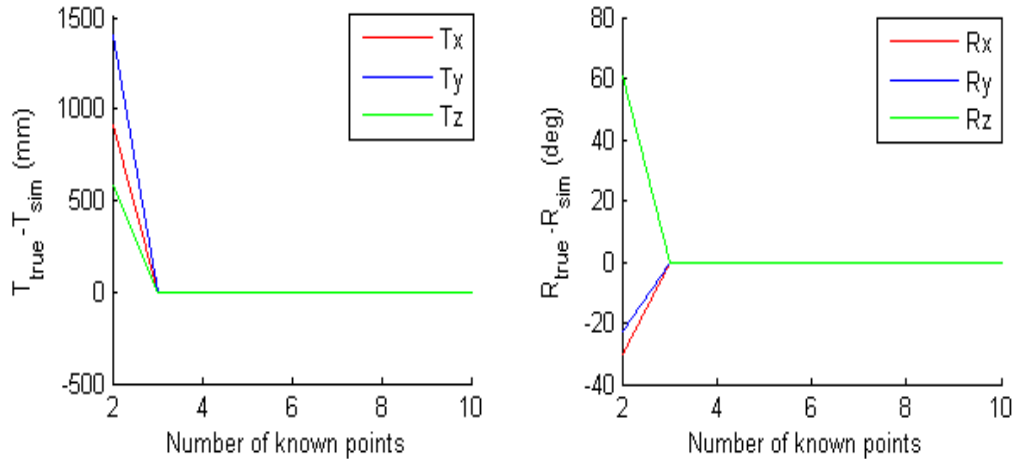


Figure 2-10: Performance w.r.t the number of known points in ideal case

This simulation evaluates the influence of the number of known points in ideal case (without noise). The number of known points varies from 2 to 10. For each

experiment, 100 independent trials have been carried out. The results are shown in Fig.2-10. When the number of known points increases to 3, the rotation matrix and translation matrix errors drop rapidly. With more than 3 known points, the error approximates zero and almost remains the same. From this simulation, we can conclude that at least 3 points are needed for applying the proposed approach.

Influence of known points number in noisy case

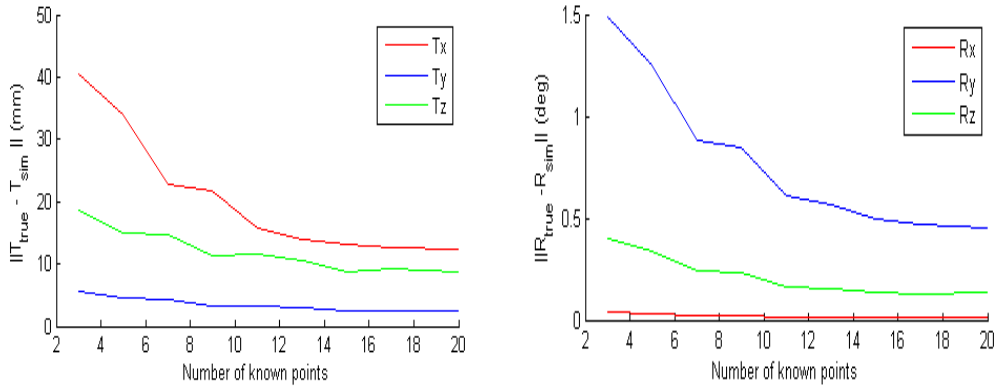


Figure 2-11: Performance w.r.t the number of known points in noisy case

This simulation aims at evaluating the performance of the proposed method with respect to the number of known points in noisy case. Gaussian noise with zero mean and 5cm standard deviation is added to all laser points (including known points and unknown points). The number of known points varies from 3 to 20. For each number, 150 independent random trials are carried out. The estimated R and T results are compared with ground truth. The average errors are shown in Fig.2-11. With the number of known points increasing, both T and R errors curve decline.

Comparison of calibration results

This experiment evaluates how the noise on laser data affects the performance and compares these results with the result obtained using the approach proposed in paper [7]. Gaussian noise with zero mean and standard deviation (from 1 cm to 10 cm) is added to all laser points. For each noise level, 150 independent random trials are

R	T(mm)
$0; -\pi/4; -\pi/36$	$0; 120; -1200$
$\pi/30; -3 * \pi/4; \pi/36$	$0; 120; -1200$
$\pi/18; -\pi/3; \pi/25$	$500; 170; -1500$
$-\pi/18; -5 * \pi/6; \pi/30$	$500; 170; -1500$
$\pi/25; -8 * \pi/9; \pi/36$	$-300; 210; -600$
$\pi/20; -\pi/12; 0$	$-300; 210; -600$
$\pi/15; -3 * \pi/4; -\pi/32$	$350; 255; -900$
$0; -2 * \pi/9; \pi/30$	$350; 255; -900$

Table 2.1: The configuration of the 8 poses used for applying the approach proposed in paper [7]

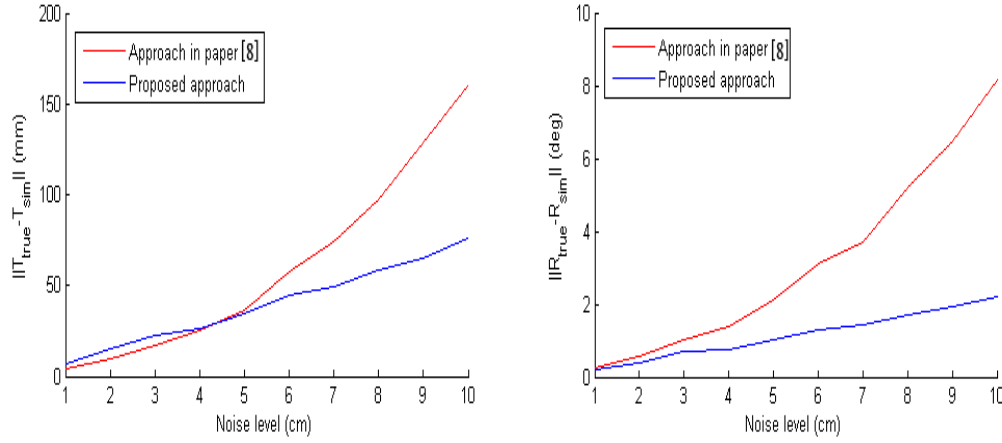


Figure 2-12: Comparison of calibration results under different noise levels

carried out. For our approach, 8 known points are used. For the approach proposed in [7], 8 poses are used to get stable outputs, whose parameters are shown in Table2.1. The reason

The average calibration error is shown in Fig.2-12. The proposed approach is better than the approach in paper [7] with respect to the estimation of the rotation matrix. For the translation matrix, the two approaches get almost the same results for noise level from 1cm to 5cm. With noise level over 5 cm, the proposed method performs better.

2.5.2 Real data experiments

In real experiments, the used fisheye lens is a Fujinon FE185C057HA1 which provides up to 185 degrees wide angle. The used camera is a pixellink PL-B742 with 1.3 megapixels (1280x1024). The used mono-layer LRF is a LMS221 with 1 degree resolution, 180 degrees field of view and up to 80m measurement range. The used computer is a normal laptop with intel core i5. All devices are mounted on the front of the vehicle (as already illustrated in fig.2-1).

Analysis of fisheye model choice

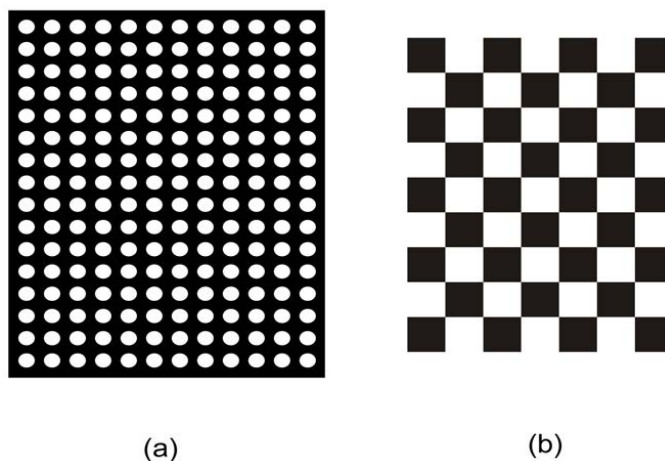


Figure 2-13: Two patterns used in convergence rate test

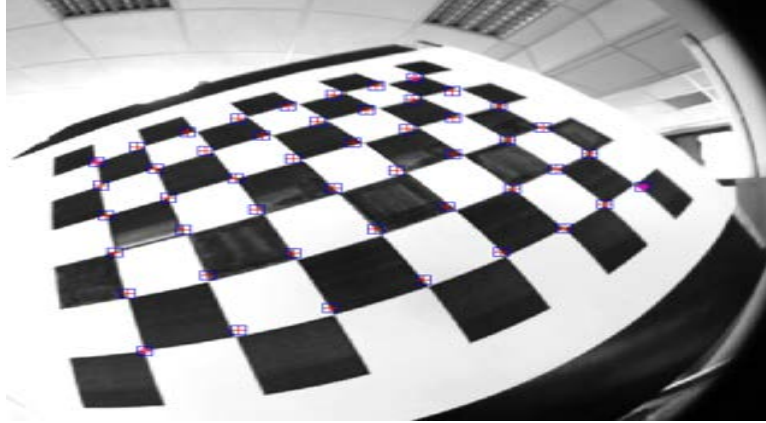
In this section, we carry out three trials to determine which model presented in section 2.3 should be adopted in our approach. The selective criterion is based on the calibration performance of the three models described in section 2.3. For the first trial, all three models are compared. The used patterns are shown in Fig.2-13. The

Model	Kannala [11]	Scaramuzza [10]	Mei [12]
Convergence rate	3–4 hours	2–3 minute	2–3 minute

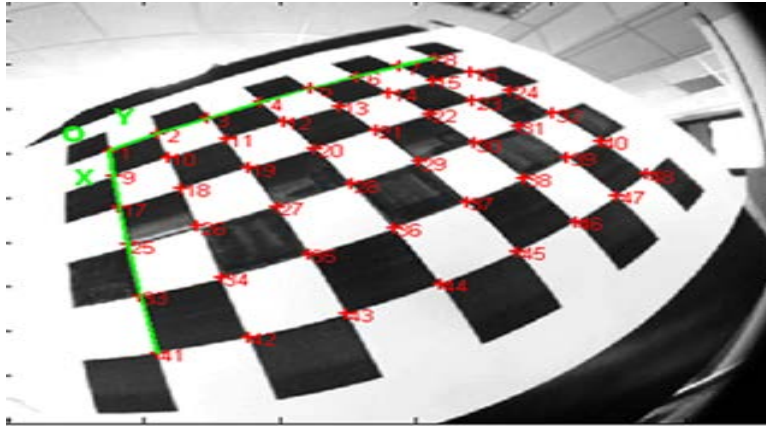
Table 2.2: The convergence time of three methods in solution optimization procedure

pattern (See Fig2-13(a)) proposed in paper [11] is combined with the 9 parameters

model (M_{P9}) and the pattern (See Fig2-13(b)) is adopted by method proposed in papers [10][12]. The results of the convergence time rate of intrinsic and extrinsic parameters optimization procedure are shown in Table 2.2. The computation cost of the method proposed by Kannala is more expensive than the other two methods.



(a) Approach in paper [12]



(b) Approach in paper [10]

Figure 2-14: Point reprojection using the calibration results of the approaches proposed in [12] and [10]

For the second trial, Scaramuzza's approach and Mei's method are compared with respect to the ability of the extraction of chessboard corners for fisheye camera calibration. For the two approaches, the reprojection errors of corner point positions are used as key parameters to evaluate the approaches performance. Therefore, the ability to extract corner points is regarded as an important indicator. The experimental results are shown in Fig.2-14. As illustrated in Fig.2-14, the extraction of chessboard

corners are better for the approach proposed by Scaramuzza (Fig.2-14.(b)) than for approach proposed by Mei (Fig.2-14.(a)). In great distortion case, all corner points are extracted correctly for Scaramuzza’s approach.

For the last trial, the three approaches are compared with respect to the average reprojection errors. The results are shown in Table 2.3. As illustrated in Table 2.3, the performance of the Scaramuzza’s approach is better than the performance of the other two method.

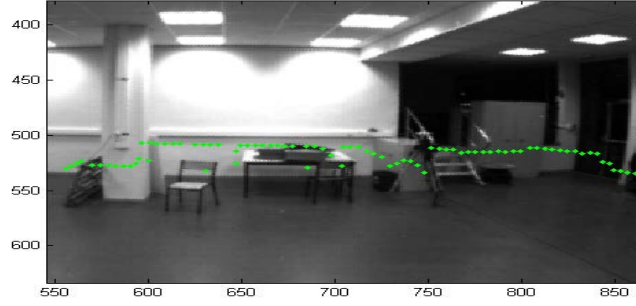
Approach	Average reprojection errors (unit:pixel)
Scaramuzza’s approach	0.55
C.Mei’s approach	0.66
Kannala’s approach	1.2

Table 2.3: Average reprojection errors

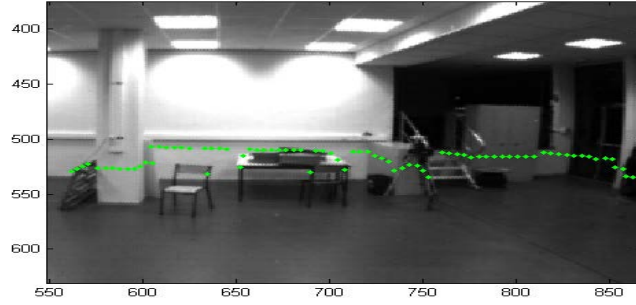
Based on these experiments, we find that the model proposed by Mei haven’t a good ability to deal very well with corner extraction in great distortion case. The computation cost regarding intrinsic and extrinsic parameters optimization for the method proposed by Kannala is too expensive. The method proposed by Scaramuzza is a good option according to the test results with respect to convergence time, corner extraction and reprojection errors. Therefore, the model proposed by Scaramuzza is chosen in our approach.

Extrinsic calibration between LRF and fisheye camera

In this experiment, real data are used to evaluate the performance of the proposed approach. As explained in section 2.4, the method requires to determine the coordinates of at least three points (known points) on the laser beams plane in camera coordinate system. In order to locate these points conveniently, two rectangle boards are used. Firstly, one is fixed in the view of the LRF and camera. And, another one is drew close to the fixed board until there is only one laser beam to pass through the gap between the two boards. Finally, the chessboard is slowly moved down along with the gap.



(a) Laser points reprojection into fisheye image using 3 known points



(b) Laser points reprojection into fisheye image using 8 known points

Figure 2-15: Calibration results for real data considering 3 known points(a) or 8 known points (b)

The performance of the proposed approach under different number of known points is evaluated. 3 known points and 8 known points are used respectively. The view (a) and view (b) in Fig.2-15 are the zoom parts of LIDAR reprojection into fisheye image using 3 known points and 8 known points respectively. We find that the location of laser points in view (b) is more reasonable than in view (a), especially near the edges of desk and chair.

Comparison of calibration results

In this experiment, we compare our approach with the method proposed in paper [7]. 8 known points and 8 poses are used respectively. The calibration results are shown in Table 2.4. From these results, we can see that the two methods give almost

the same results. According to LMS211 technique manual, the average measurement error of LRF is about $2cm$. From the simulation results, within this range, there are no big differences between the two outcomes.

Approach	R			T(m)
Approach proposed in this paper	0.034	− 0.135	− 0.990	1.413
	0.999	0.019	0.032	0.309
	0.014	− 0.991	0.135	−0.929
Approach proposed in paper [7]	0.031	− 0.140	− 0.990	1.402
	0.999	0.025	0.028	0.314
	0.020	− 0.990	0.140	−0.921

Table 2.4: Calibration results obtained by the two approaches

2.5.3 Application in ICP algorithm

In order to illustrate the interest of extrinsic calibration between LRF and fisheye camera, we use the calibration result to integrate both LRF measurements and color information in ICP algorithm (Iterative Closet Points). ICP algorithm was introduced in the early 1900s [21] and was further developed by various researchers. The most cited version is the one proposed in [22]. Since it was born, ICP algorithm is widely used in many research areas, especially in geometric alignment of 3D models.

Basic ICP algorithm

Given two roughly aligned shaped represented by point clouds, the ICP algorithm will implement the following tasks:

- 1) Generate temporary correspondences from the two clouds of points.
- 2) Estimate the relative rigid body transformation between the two clouds.

The first step is the key factor to the final estimation for rigid body transformation. In order to get a relatively good correspondence for a point, the ICP algorithm iteratively perform the following steps:

- ◊ Matching: the nearest neighbor of each data point in the points clouds is found.
- ◊ Minimization: the error metric for whole data set is minimized.

◇ Transformation: data points are transformed using minimization result.

The algorithm is terminated based on the number of iterations or the relative change in the error metric.

Closet point with color information

Generally, ICP algorithm converges very quickly, however several problems may occur:

1) Local minima: Instead of the global minimum, the algorithm may converge towards one of multiple local minima in the error metric

2) Noise and outliers: Outliers and noise may play a great side effect on the minimization process of the error metric, which leads to faulty results.

3) Partial overlap: the point clouds may lose partial information regarding the object due to the partial overlap.

To solve these problems, many variants have been introduced based on the basic ICP concept. In a review paper [23], the authors classified the proposed variants of the algorithm as affecting one of the six subtasks:

- ◇ Selection (Choosing subsets of input point sets).
- ◇ Matching.
- ◇ Weighting (correspondences).
- ◇ Outlier removal.
- ◇ Error metric.
- ◇ Minimization.

Our work belongs to matching part. In the original ICP algorithm, the euclidean distance was used as the criterion to determine the closet point. Let S_v and S'_v denote the two points sets observed by a sensor from two different viewpoints. The problem to find the closet point of the point d_i in S_v can be converted to the following question:

$$\arg \min_{d_i \in S_v, d'_i \in S'_v} \|d_i - d'_i\| \quad (2.36)$$

where d'_i is a point in S'_v . Euclidean distance is useful in many cases. However, it may

become invalid if the partial overlap case happens. To handle this situation, many additional features are proposed. The color information is one of them. To obtain the corresponding color value, the extrinsic parameters between vision sensor and range sensor are needed. As done in paper [24], the modified version of euclidean distance is:

$$\arg \min_{d_i \in S_v, d'_i \in S'_v} \|d_i - d'_i\| + \sqrt{\alpha_1(a_1 - a_2)^2 + \alpha_2(b_1 - b_2)^2 + \alpha_3(c_1 - c_2)^2} \quad (2.37)$$

where $\alpha_1, \alpha_2, \alpha_3$ are weight coefficients and a_x, b_x, c_x ($x=1,2,3$) are respectively R,G and B values in RGB model.

Real data experiment

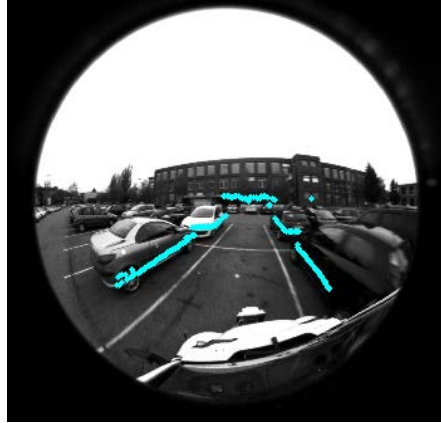
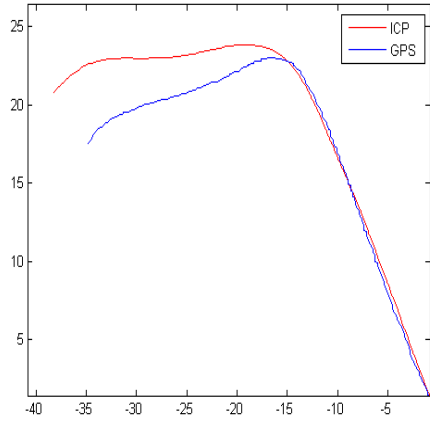
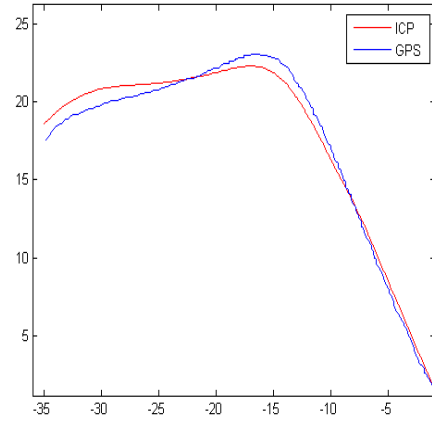


Figure 2-16: Projection of LRF measurements in the fisheye image

Our experiment is conducted in a parking. The environment and projection of LRF measurements in the fisheye image are shown in Fig.2-16. Fig.2-17 shows the result of the estimation of motion trajectory. The red and blue curves are the results obtained by the ICP algorithm and GPS respectively. GPS results are used as ground truth data to evaluate the ICP performance. As illustrated in Fig.2-17(a), only with LRF measurements, we can see that localization error happens while the vehicle turns toward left in the range $[-10, -15]$. Nevertheless, in Fig.2-17(b), with LRF measurements and color information, this error is more controlled.



(a) With LRF measurements only



(b) With LRF measurements and color information obtained from fisheye image

Figure 2-17: Estimation of motion trajectory under two different case

2.6 Conclusion and Future Works

In this chapter, we firstly give a brief review about the state of the art of extrinsic calibration between LRF and camera. Extrinsic calibration is used to determine the rigid transformation between LRF and fisheye camera. It can help researcher to overcome a single vision sensor drawbacks for perceiving the environment. Three sorts of fisheye model are introduced. Unlike classic vision camera, fisheye camera provides a wide angle vision but delivers images with great distortions. It makes the traditional pinhole model invalid. However, a unified fisheye model doesn't exist. To find out a relatively good fisheye model is important to the extrinsic calibration work. After that, we introduce our extrinsic calibration method in detail. Generally speaking, it is based on the laser scanning plane equation and some known points on it. In the experiment section, we used simulated and real data to show the effectiveness of our method. At last, an interesting application of extrinsic calibration is presented.

The drawback of our method is that it takes some time to determine the known points. As future works, to improve the accuracy and to save user time, the following works could be attempted:

- 1) Infrared detection card (IR card) could be used to save the time required to

locate the laser beams. IR card emits clearly visible light when illuminated by laser diode. This allows to locate easily.

2) Some special plane information could be considered to improve the accuracy. In the proposed method, only laser beam plane is employed to construct the geometrical constraints. However, in fact, some other special planes can be explored to develop the geometrical constraints (e.g., the plane formed by the view point and the LRF scanning line on the pattern).

Chapter 3

Road Detection Based on Fisheye Camera and Laser Range Finder

3.1 Introduction

For autonomous vehicles and Advanced Driver Assistance Systems (ADAS), an important task is to keep the vehicle traveling in a safe region and prevent collisions. To meet that requirement, the vehicle has to perceive the structure of the environment around itself. The free road surface ahead of the vehicle has then to be detected. In addition, a robust effective road detection system also plays an important role in higher other tasks such as vehicle and pedestrian detection (see chapter 4). The derived free road space can indeed provide a significant contextual information to reduce the region-of-interest for searching targets (cars, pedestrians,...), which contributes to reach a reasonable computational cost and to remove false detections.

In our work, we aim at performing road detection using a monocular camera with fisheye lens and a 2D LRF. An example of expected result is illustrated in Fig.3-1. Compared to classic lens, fisheye lens has greater FOV providing more information about the scene. But the disadvantage is the great distortion appearing in the images. Therefore, we propose to use the color space as feature space. In paper [25], the authors prove that the log-chromaticity based illumination invariant grayscale image is more suitable than HSI (as done in paper [26]) for road detection. Howev-

er, in our research, we notice that using only illumination invariant image can cause over saturation problem or under saturation problem in some cases such as cloudy situation. So, in this chapter, a novel approach combining log-chromaticity space (as in paper [25]), HSI space [27] and LRF information is proposed. It firstly derives the coarse road binary image by histogram based classification of the illumination invariant grayscale image. Then, a validation step is applied to check the coarse road binary image. Finally, a refined process based on HSI space is carried out.

The rest of the chapter is organized as follows: Section 3.2 introduces the state of the art. Section 3.3 presents the general framework overview. Section 3.4 describes the coarse road detection based on log-chromaticity space. Section 3.5 introduces how HSI color feature and LRF are used to refine the coarse road detection results. Section 3.6 shows real data experimental results and compares results of the proposed approach against illuminant-invariance based algorithm [25]. Conclusions are given in section 3.7.

3.2 State of the Art

Road detection has been widely studied for past several years and many approaches have been proposed. According to the used equipments, methods can be categorized into three types: approach based on LRF only, approach based on camera only, approach based on both LRF and camera.

In papers [28] and [29], the authors proposed approaches based on 3D LRF data. The road information is segmented from points cloud. The advantage of LRF is that it can provide reliable range measurements that are not likely affected by the illumination. In certain cases, LRF-based methods can perform very well. However, a limitation of these methods is that LRF can't offer visual information, for example, traffic signals and object appearance. Yet, in many applications, such as object recognition and tracking, visual information is crucial for autonomous vehicle. Besides that, the cost of 3D LRF sensor is still very high.

Compared to LRF, camera can offer substantive visual information in favor to

the recognition of on-road objects and traffic signals. Moreover, such passive sensor is not affected by the interference problem between the same type of devices. Generally, road detection based on vision is a challenging work for autonomous vehicle in outdoor scenario due to the background changement with vehicle traveling and the presence of many moving objects on the road whose movement is hard to predict. Furthermore, the structure of road is not fixed and the materials, illumination and weather conditions have effects on the road appearances. Therefore, a variety of vision-based approaches have been developed by researchers.

Generally, camera based approach is divided into monocular based and stereo based. In paper [30], stereo camera is used for urban scene reconstruction. Firstly, the depthmap of stereo camera is utilized to estimate a set of piecewise planes in image. These piecewise planes belonging to the same one are then linked as one complete planar. These different completed planes are labelled as different classes. Although depthmap is a useful tool for estimating the vertical plane, it can't work well for horizontal plane. In general, the road plane is a horizontal plane. In paper [31], authors propose to use V-disparity map to detect the road area in the image. The V-disparity map is the image which counts the number of consistent points with same disparity value along vertical direction in disparity image. In V-disparity map, the road surface, in ideal case, is an oblique line from upper left to right down, and the obstacles, in general, are vertical lines. These lines in V-disparity map are then detected by hough transformation. However, for hough transformation, it is hard to decide the proper number of the lines.

In paper [32], Conditional Random Fields (CRF) based monocular classification is used to segment multiple scene objects in the field of view (FOV) of a single camera. Using conventional CRF based methods for segmentation generally makes the assumption that all pixels in any small segmentation belong to the same object. However, the pixels on the boundary of an object can be shared by multiple object classes. The authors propose a high order potential function as soft constraint to deal with the problem. Nevertheless, the accuracy of detection is still not very high and the problem formulated as an energy minimization task is a NP-hard problem. In

paper [33], a mixture of Gaussians in RGB color space and a Gaussian distribution are used to model the road. The pixels can be classified by the property of corresponding gaussian model. But the drawback of this sort of method is hard to decide the proper number of Gaussians. In paper [34], the texture orientation of pixels and a soft voting scheme are employed to seek the vanishing point of road. Firstly, several vanishing points are picked out as candidates. For each candidate, it is estimated by a soft scheme voting strategy based on a local region defined by the authors. The voting strategy is based on texture orientation of the pixels in the local region and ratio coefficient decided by the diagonal length of the image and the distance between the pixels in the region and vanishing point. Each candidate has a confidence as estimation results. The candidate with the highest confidence is considered as the real vanishing point. And, finally, this vanishing point will work with the texture orientation to determine the road region in the image. However, this approach is not suitable for urban case because the boundary of road is often covered by parked or moving car. In paper [35], Structure-From-Motion (SFM) is used to estimate a map-based road boundary model.

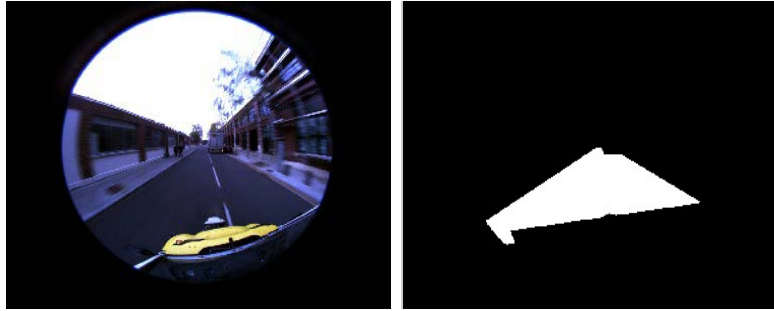


Figure 3-1: The left image is original fisheye image, the right image is the road image which we aim to obtain.

3.3 Framework Overview

The general framework diagram of the proposed approach is shown in Fig.3-2. The input is a fisheye camera image. In a fisheye image, the middle part is the context of

the captured traffic scene, and the remaining part is useless black area (see Fig. 3-3). The useless area, having side effect on solving illumination invariant grayscale image, is firstly removed.

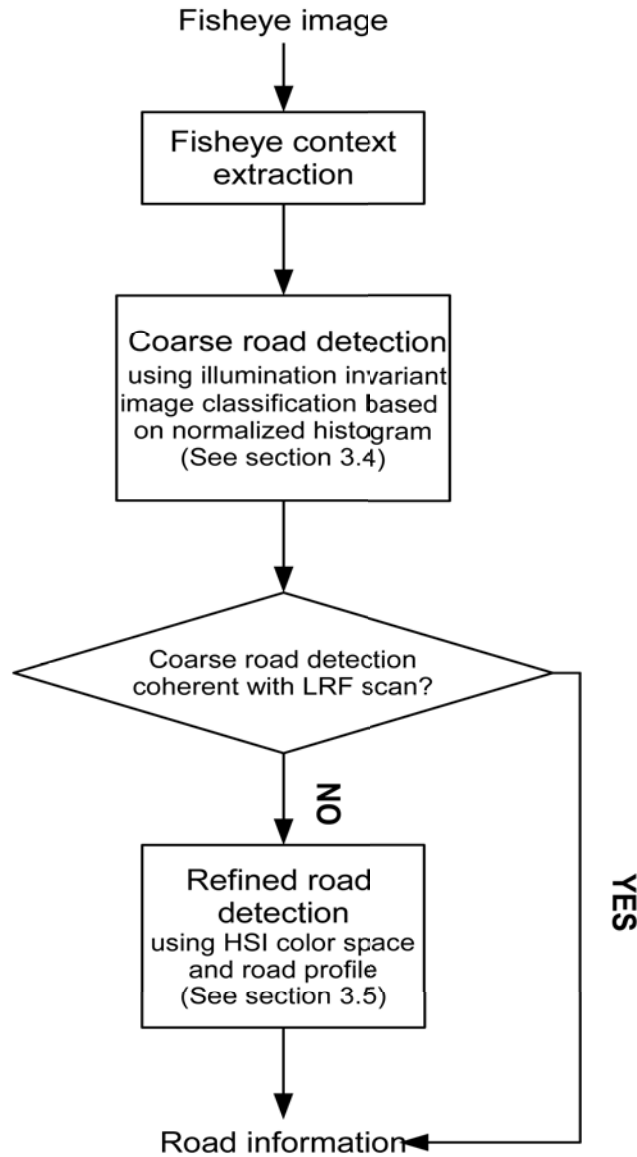


Figure 3-2: Road detection framework diagram

After context extraction, the coarse road detection algorithm is carried out. It is based on the illumination invariant grayscale image. The illumination invariant grayscale image is computed from the mapping of the image from RGB space to

log-chromaticity space where the illumination invariant angle is obtained. Then, a classification step based on the histogram of this image (as in paper [36]) is implemented. If a pixel is classified as road, its value is set to 1. Otherwise, its value is set to 0. The scattered road pieces are connected to form road binary image by connected-component and fill-in hole algorithm. However, there may exist error classification due to over saturation or under saturation problem. A validation step based on LRF data is used to check and correct these errors using a refined procedure based on HSI color space information.

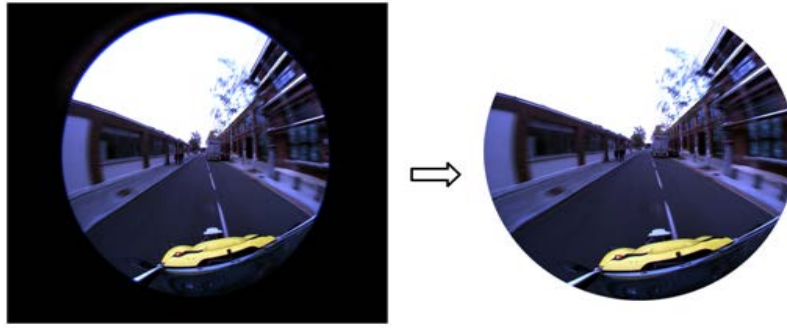


Figure 3-3: The context of the captured traffic scene is extracted from the original fisheye image.

3.4 Coarse Road Detection Based on Illumination Invariant Image

Coarse road detection step is based on log-chromaticity space in paper [37]. It is introduced by Finlayson in paper [38] and used to find the illumination invariant feature.

3.4.1 1 dimensional illumination invariant images

The 1 dimensional illumination invariant image is based on the Lambertian model introduced in paper [39]. In this model, RGB color of a pixel is represented by a spectral power distribution and surface reflectance function. Let $E(\zeta, x, y)$ denote

the spectral power distribution, $S(\zeta, x, y)$ the surface reflectance function, the RGB color formed at a pixel (x, y) can be described as follows:

$$\rho(x, y)_k = \int E(\zeta_k, x, y) S(\zeta, x, y) Q_k(\zeta), \quad k = R, G, B \quad (3.1)$$

where $Q_k(\zeta)$ denotes the spectral sensitivity of the k channel camera sensor, ζ the wavelength of light. The integral usually occupies the range of the visible light. Based on the assumption that the camera sensor obeys to Dirac delta functions ($Q(\zeta) = q_k \delta(\zeta - \zeta_k)$), Equation 3.1 can be simplified as:

$$\rho_k = E(\zeta_k, x, y) S(\zeta_k, x, y) q_k \quad (3.2)$$

If the illumination follows the Planckian's laws described in paper [40], a light with a spectral power distribution can be parameterised by its colour temperature T :

$$E(\zeta) = F c_1 \zeta^{-5} e^{-\frac{T\zeta}{c_2}} \quad (3.3)$$

where c_1 and c_2 are constants, and F is a variable adjusting the overall intensity of the light. Taking equation 3.3 into 3.2, it becomes:

$$\rho_k = F c_1 \zeta_k^{-5} e^{-\frac{T\zeta_k}{c_2}} S(\zeta_k) q_k, \quad k = R, G, B \quad (3.4)$$

By dividing ρ_R and ρ_B by ρ_G , then we have the log-chromaticity coordinate as follows:

$$\omega_j = \log \frac{\rho_j}{\rho_G} = \log \frac{s_j}{s_G} + \frac{1}{T} (e_j - e_G), \quad j = R, B \quad (3.5)$$

where $s_k = c_1 \zeta_k^{-5} S(\zeta_k) q_k$ depends on the surface and camera. $e_k = -c_2 / \zeta_k$ depends on camera. With temperature T change, for a given surface, ω will move along a straight line in log-chromaticity space. The direction of this line is determined by $(e_k - e_G)$ which depends on camera, but is independent of the surface and illumination. It means that the surface color under different illumination lies on a straight line in the log-chromaticity space and all those lines are parallel each other for different surface

colors with slope $(e_k - e_G)$ (See Fig.3-4). The 1-d illumination invariant image can be determined by projecting log-chromaticity of pixels into the direction orthogonal to the vector $(e_k - e_G)$.

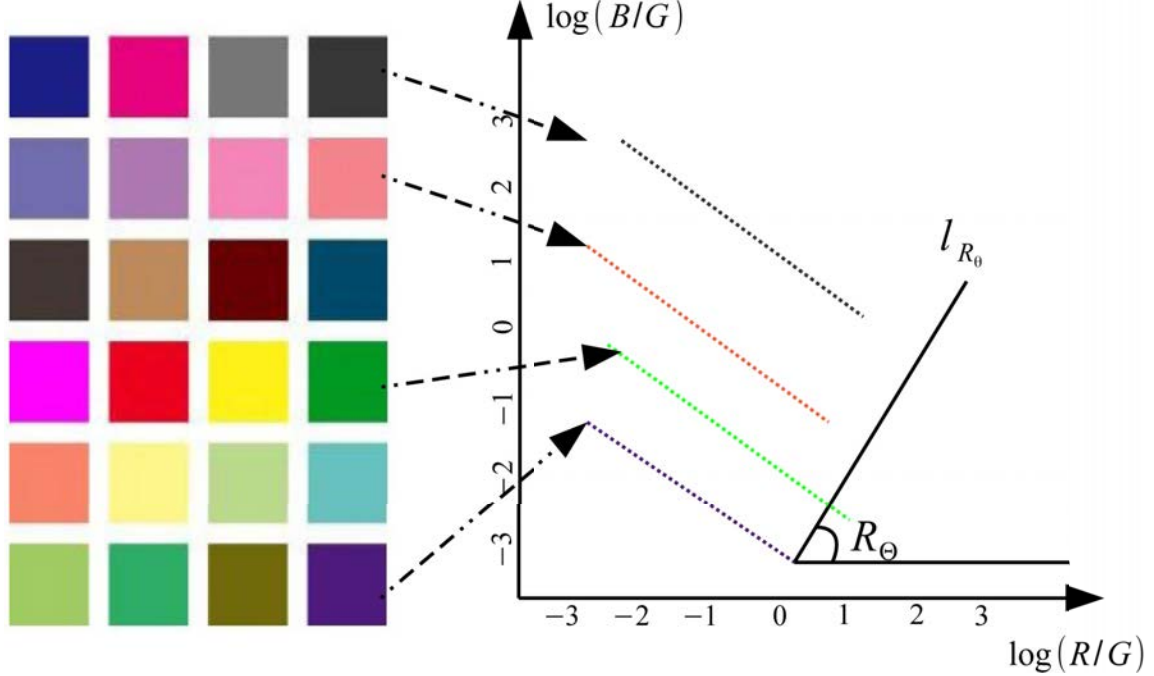


Figure 3-4: Left image is a color checker, right graph represents the color in color checker under different illumination maps to log-chromaticity space. A set of different color surfaces under different illuminations form several parallel lines. l_{R_θ} is a line perpendicular to these parallel lines. The projection of log-chromaticity of pixels into l_{R_θ} form a 1-d illumination invariant image.

Illumination invariant direction estimation

According to the above description, it is known that the direction $(e_k - e_G)$ is the key point for deriving the 1 dimensional illumination invariant images. In log-chromaticity space, the direction of the line perpendicular to parallel lines formed by different color surfaces under different illuminations is defined as illumination invariant direction D_i . This direction can be expressed angle (R_θ) of its slope (See Fig.3-4) and can be determined by its entropy energy. Let I_{gi} denote the grayscale image derived from projecting the log-chromaticity coordinate of a given color image onto a line l_{R_θ} with slope R_θ . The entropy energy E_R of I_{gi} based on its histogram H_R is defined as

follows:

$$E_R = - \sum_{i=1}^{N_b} H_R(i) \log(H_R(i)) \quad (3.6)$$

where N_b is the number of bins of the histogram H_R . The entropy value depends on the value of R_θ . If R_θ is consistent with D_i , the log-chromaticity values of pixels will scatter in same bin of H_R . Thus, it leads to a low value of the entropy energy E_R . Conversely, if R_θ deviates from D_i , a high value of E_R will be expected. Hence, the illumination invariant direction can be determined as long as the minima of E_R is solved. In paper [41], the authors propose to determine R_θ based on a single image content. This approach is not robust for many different images. Effectively, the

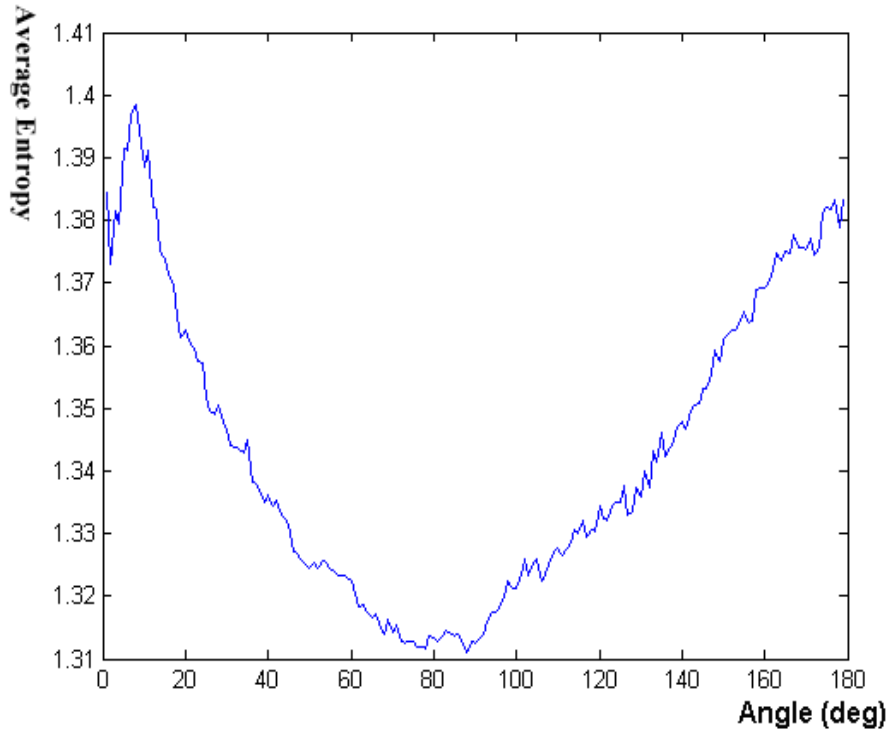


Figure 3-5: Average entropy minimization for different values of the angle R_θ

quality of one single image content can be affected by many factors and it is not

sufficient for estimating D_i . In paper [25], a method based on a set of images content is proposed and is proved to be more robust than the approach proposed in paper [41]. In our research, we adopt this method. Firstly, a set of color images under different illuminations is chosen, and each of them is mapped into log-chromaticity space. And then, the points of the image in this space are projected onto the line with R_θ initialized to 0 to form a new grayscale image. For the new grayscale image, Chebyshev's theorem proposed in paper [42] is then applied to reject outlier points. Among the remaining points, the middle 90% points are picked out to computer the histogram of the image. The bin width of the histogram is fixed by Scott's rule [43]. The Scott's rule is defined as :

$$Bin_{width} = 3.5N_I^{-1/3}std(I_i) \quad (3.7)$$

where *std* stands for stand deviation, and N_I is the number of pixels of image I_i . After obtaining the histogram, the entropy energy E_R of the image can be calculated using equation 3.6. However, the derived energy is only for $R_\theta = 0$. To get full results, R_θ is varied from 0 to 360 by a fixed step, and the entropy is also estimated again for each value of R_θ . Finally, the average entropy energy of the set of images for each angle is obtained (See Fig.3-5). The angle corresponding to the minima average entropy is the expected illumination invariant direction. Fig.3-6 shows the transformation from RGB image to illumination invariant grayscale image.

3.4.2 Classification based on illumination invariant image histogram

After the illumination invariant grayscale image computation, a classification based on normalized histogram of road model is applied. The aim is to classify the illumination invariant grayscale image into two classes "road" and "non-road". As in paper [44], the road model is a fixed small field in front of vehicle in each image (that is reasonable

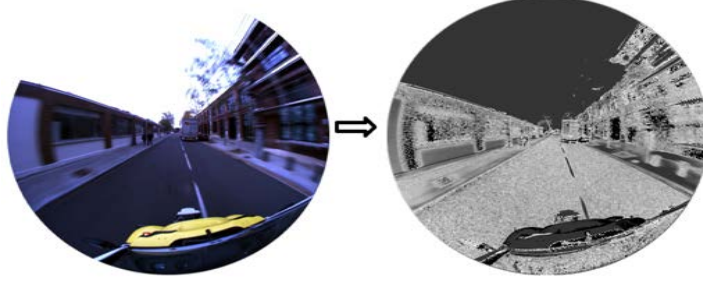


Figure 3-6: The context of the captured traffic scene in RGB space is converted to illumination invariant grayscale image

in most of cases). Let S_r denote the fixed small region (represented as a blue rectangle in Fig.3-7), G_{rmin} and G_{rmax} are respectively the minimum and maximum gray level of illumination invariant image in S_r , G_i the gray level of i -th pixel in illumination

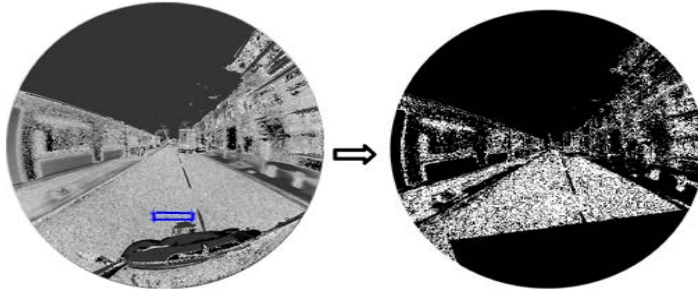


Figure 3-7: The illumination invariant grayscale image is classified as road and not road. For convenience, the front part of the vehicle is covered by a dark rectangle.

invariant image and λ_i the probability of the i -th pixel provided by the normalized histogram of illumination invariant grayscale image. If the two following conditions are satisfied, the pixel in illumination invariant grayscale image is identified as road:

$$\begin{cases} G_{rmin} < G_i < G_{rmax} (i = 1, 2, \dots, N) \\ \lambda_i > \lambda_f (\lambda_f > 0) \end{cases} \quad (3.8)$$

where λ_f is a threshold which is set to 0.25 (empirical value) and N is the number of pixels in the image. The result of classification is illustrated in Fig.3-7. The white

pixels are labeled as "road" and the black pixels are "non-road".

However, as shown in fig.3-7, many scattered pieces of road pixels are presented in the derived result. To form a more complete road image, further processing is needed. Firstly, a connected-component algorithm is applied to the binary image obtained by the previous classification algorithm to form a connected groups. Then, flood-fill operation based on morphology is used to fill holes in each connected group. In fact, flood-fill operation brings the intensity values of dark areas that are surrounded by lighter areas up to the same intensity level as surrounding pixels. These connected groups, which are not connected to the predefined road region S_r , are removed. Finally, a relatively more complete road binary image is formed as illustrated in Fig.3-8.

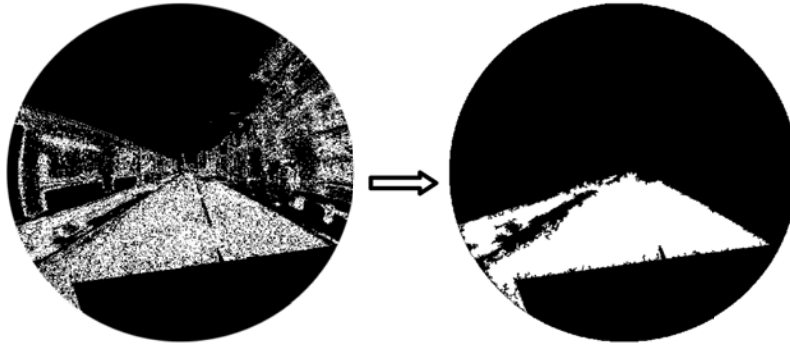


Figure 3-8: Scattered pieces of road pixels are connected to form the so called road binary image.

3.5 Road Detection Refinement Using LRF Measurements

3.5.1 Coherence checking between coarse road image and LRF measurements

However, some pixels are possibly falsely classified as road or non-road in I_{dr} . To detect and correct such errors, a checking procedure based on LRF measurements and

amount of road pixels is proposed. For the convenience of statement, the binary road image obtained from above describe step is called as I_{dr} . A 2D-LRF is mounted on the front of the vehicle (see Fig.3-15) and there exist an upwards pitch angle for laser scanning plane. In this configuration, roads often lay below the laser scanning plane. Given the extrinsic parameters between the LRF and the fisheye camera known, the laser scanning plane corresponds to a line in image. In this paper, this line is called as dividing line. So if the pixels above dividing line in the image I_{dr} are labeled as "road", the validation step should consider that an error occurs. Obviously, merely this condition is not sufficient if we consider the pixels below dividing line are falsely classified. To address this issue, the information of the amount of road pixels are used. Suppose I_{dr_n} is the current road image, and $I_{dr_{n-1}}$ is the previous one. By observation, we find that the amount of road pixels is a useful tracking information for checking if there exists errors in the image I_{dr} . Because it remains relatively stable between two consecutive frames if there is no error occurs due to over or under saturation. So, if there exists dramatic change in the amount of road pixels between two consecutive frames, it have a high probability that the errors have happened. In summary about the above two cases, the conflict checking condition is:

$$\begin{cases} P_{up_n} \in P_{r_n} \\ M_{F_n} > (1 + \beta)M_{F_{n-1}} \\ M_{F_n} < (1 - \beta)M_{F_{n-1}} \end{cases} \quad (3.9)$$

where P_{up_n} denotes a pixel above the dividing line in the image I_{dr_n} , P_{r_n} represents the set of road pixels in the image I_{dr_n} , M_{F_n} and $M_{F_{n-1}}$ are the amounts of road pixels in the image $I_{dr_{n-1}}$ and I_{dr_n} respectively. β is a threshold set manually. In experiments, it is set to 0.1 (experience value).

3.5.2 Refined road detection procedure

The framework of the refining procedure is shown in Fig.3-9. It firstly adopts two consecutive frames $(I_{dr_n}, I_{dr_{n-1}})$ as inputs to find out two fields: discrepancy field

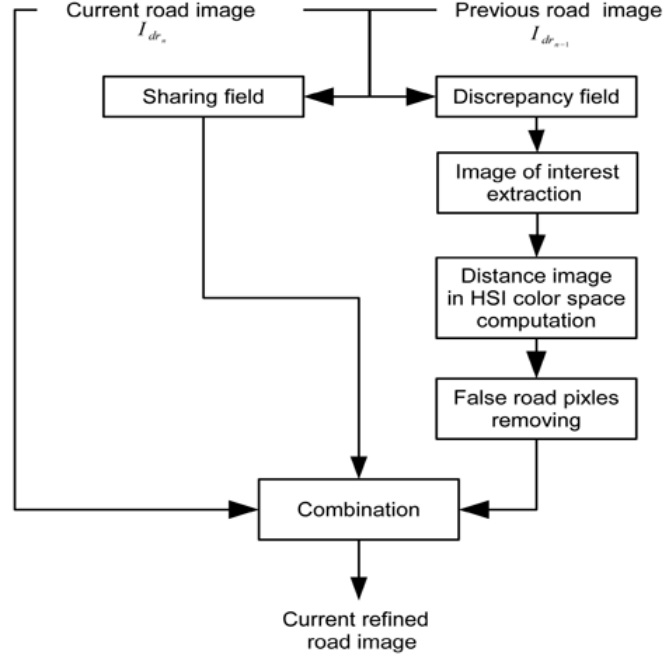


Figure 3-9: Framework of refined procedure

and sharing field. The sharing field is the field that contains the overlapping road area of two consecutive coarse road image. Similarly, discrepancy field is the one that contains the different road area between the image I_{dr_n} and the image $I_{dr_{n-1}}$. In our observation, false road pixels are usually within the discrepancy field in the image I_{dr_n} . So we mainly focus to correct the errors in discrepancy field in the image I_{dr_n} . Firstly, in current fisheye image, the content in discrepancy field and in region S_r (already defined in detection step) are extracted as region of interest (ROI) (See Fig.3-10). Then, the ROI is converted to HSI space to solve a distance image where a threshold value is calculated to rule out the falsely classified pixels. At last, the outcome after false road pixels removing procedure, the content in sharing field in the image I_{dr_n} and the image I_{dr_n} are combined to form a refined road binary image.

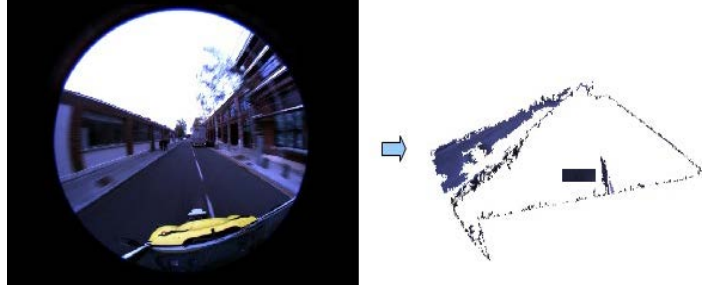


Figure 3-10: ROI extraction

Distance image computation

The distance image is based on Euclidean metric adopted in HSI space. Compared with RGB model, HSI model is compatible with the vision psychology of human eyes [45]. It is defined as :

$$\begin{aligned}
 I &= \frac{R + G + B}{3} \\
 S &= 1 - \frac{3}{R + G + B} \times \min(R, G, B) \\
 H &= \arccos\left(\frac{0.5 \times [(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}}\right)
 \end{aligned} \tag{3.10}$$

Fig.3-11 shows an example of the content of a RGB image in the H,S and I channels. As illustrated in this figure, it is known that: I (intensity) component has the most useful information, S (saturation) component has a few available information, H (Hue) component hardly provides useful information. Hue component represents color properties and only the content with great color change can be discriminated in this channel. This characteristic in H channel also can explain why color processing method can't perform more better than grayscale processing approach. In paper [26], the authors analyze the role of the histogram of the two components S and I in image segmentation procedure. They consider that intensity values alone (without position

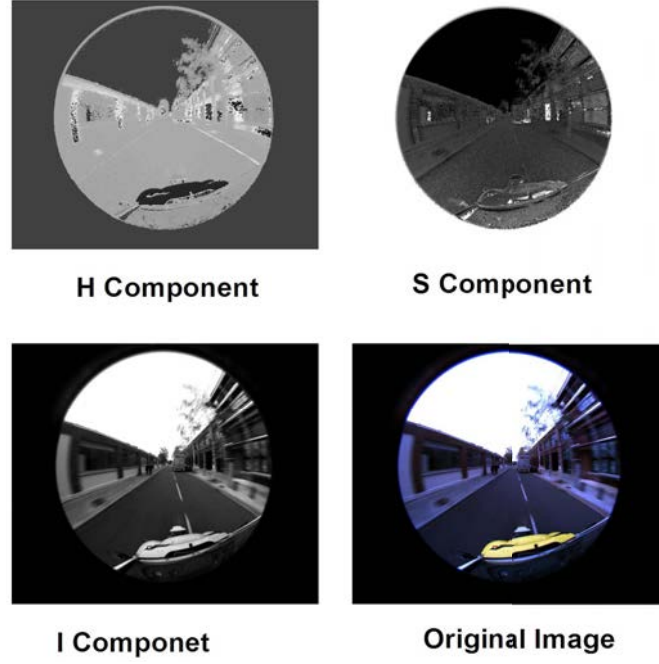


Figure 3-11: The content of the image in H,S,I channels.

information) hardly get better segmentation results for road detection because there exist great difference between road pixels and lane marker pixels. Only with S component, the road boundary is hard to be detected. So, the feasibility of detecting the road in SI plane is studied. To analyze the histogram of SI plane, the authors find that there always exist two major peaks corresponding to the road and the sky. In high contrast image, the two major peaks are far in SI plane. Otherwise, in low contrast image, the two major peaks are close. And the rest of small peaks correspond to other objects in the image (See Fig.3-12). According to the authors view, a pixel can be well classified as 'road' if the average S and I values of road are known. Thus, choosing the road reference area is one of the key aspect when classifying the pixels.

In our research, we also adopt this research line and choose road area S_r (already defined in coarse road detection step) as reference road area. Based on the chosen reference road area and relevant components of HSI model, distance image of ROI in HSI space is built. In HSI space, a distance image indicates the difference between the pixel and the average value of the reference road area. Let ROI_{HSI} represent the

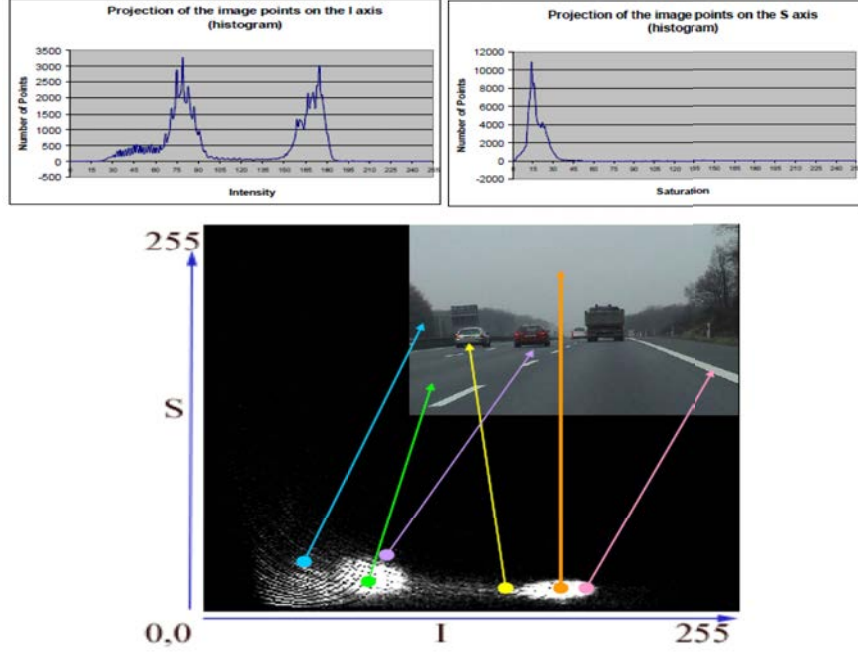


Figure 3-12: The histograms for S,I and SI plane shown in [26]

ROI image in HSI space and (I_{SI}, I_{SS}) denote the average value of the region S_r in I and S channels respectively. The distance image is defined as follows:

$$d_{SI} = \sqrt{\frac{(I_I - I_{SI})^2 + (I_S - I_{SS})^2}{d_f}} \quad (3.11)$$

where I_I and I_S are the pixel values in I and S channels respectively, the denominator d_f is a dynamical factor based on the content of ROI_{HSI} . In HSI space, ROI_{HSI} is divided into two areas: inside area and outside area. The inside area is S_r and the outside area is the rest part of ROI. Factor d_f represents the difference between inside area and outside area. Let (I_{SO}, I_{IO}) denote the average values of outside area in S and I channels. The factor d_f is defined as follows:

$$d_f = (I_{SI} - I_{IO})^2 + (I_{SS} - I_{SO})^2 \quad (3.12)$$

False road pixels removing in distance image

In the distance image, a deciding threshold is calculated to rule out the false road pixels. Generally, this threshold is associated with the prior knowledge of road. In our case, we have LRF measurements and image information. Combining the two types of information can make the threshold more robust. Let L_d denote dividing line (already define in conflict checking) in fisheye image, L_b the estimation of road boundary. In our case, L_b has two options. One depends on the road boundary of previous image. Generally, there is no drastic change of road boundary between two consecutive frames. The road boundary of previous image can be used to approximate the road boundary of the current frame. Another option is to estimate road boundary by LRF measurements. In practice, we predefine a line parallel to L_d and below it, d_λ the distance between the two lines in vertical direction. This predefined line can be treated as an estimation of road boundary. The way to choose L_b obey the following principle: L_b is the road boundary of previous image if the position of road boundary of previous image is blow L_d in vertical direction; Otherwise, L_b is the predefined line. In our experiments, d_λ is set to 30. In the distance image, one road pixel is considered as false road pixel if it is above L_d in vertical direction. If one pixel is located in the field between L_d and L_b and its value is less than a predetermined upper limit threshold T_2 , it is reserved as road. Similarly, if one pixel is below L_b and its value is less than a predetermined upper limit threshold T_1 , it is also reserved as road. The thresholds T_1 and T_2 are based on the average value and standard deviation of S_r area in the distance image, and they are defined as follows:

$$\begin{cases} T1 = M_{sr} + \sqrt{f_d} * std_{sr} \\ T2 = M_{sr} - \sqrt{f_d} * std_{sr} \end{cases} \quad (3.13)$$

where M_{sr} is the average value of S_r area in the distance image and std_{sr} the standard deviation. Fig.3-13 shows the result of removing false road pixels in the distance image. For the convenience of statement, the outcome of this procedure is called as I_f .

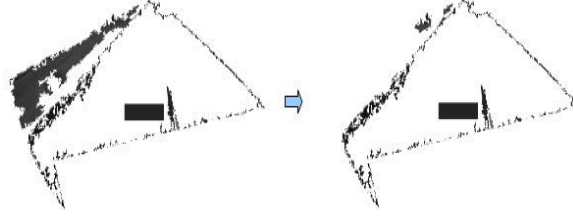


Figure 3-13: Process of removing the false road parts in distance image

Combination

Through above classification algorithm, most of incorrect road pixels can be ruled out. However, there still exist some incorrect parts in the image I_f , like Fig.3-13 upper-left part. But fortunately, some key points are discarded. Key point is the intersection point of non road part and road part in the image I_{dr} . If the key points are classified to road, the non road part will be connected with road part by the connect-component algorithm. As long as these key points are abandoned, it will be easy to remove the non road parts. Firstly, the content in sharing field in the image I_{dr} is added to the image I_f to form a improvement road image I_{ir} . And then the connect-component algorithm is applied to the image I_{ir} . In the image I_{ir} , the pixels which are not connected to the road region S_r are treated as errors road pixels and are discarded. So far, a refined road image I_{rr} is obtained. However, the above refined approach often take the distant road pixels in ROI as error road pixels. Although the amount of the distant road pixel is few, we still try to get them back. Let I_{rr} be divided into several equal intervals (See.Fig3-14) according to the height of road area, and F_{int} represent the first interval. h_1 is the row which corresponds to the lower limit of road area in F_{int} , V_1 and V_2 are the columns which correspond to the left and right limits of road area in F_{int} respectively. The three lines h_1 , V_1 and V_2 can form a closed field (red field in Fig.3-14). The road pixels in this closed field in the image I_{dr} are picked out to add to the image I_{rr} to fill the miss distant road information.

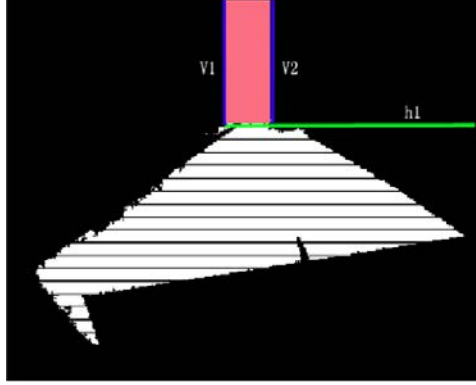


Figure 3-14: Image is divided into equal intervals

3.6 Experiment

3.6.1 Setup



Figure 3-15: The configuration of the used experimental platform

The layout of LRF and fisheye camera is shown in Fig.3-15. The fisheye camera is put on the top of vehicle, and LRF is at bottom.

3.6.2 Experimental results

The experimental data are composed of 336 images from four different sequences. To reduce the computational time, all images are down sampled to 640×512 pixels resolution. The ground truth is labelled manually.

In first experiment, the algorithm is tested in HSI and RGB space. For each frame, we record the accuracy of detection result. The accuracy is defined as follows:

$$Accuracy = \frac{B_t + R_t}{M_I}. \quad (3.14)$$

where B_t and R_t are the amount of correct background pixels and correct road pixels respectively, M_I is total pixel number. The average accuracy for each sequence is resumed in Table3.1. We can see that the proposed approach performs better in HSI than in RGB.

	Proposed approach	
Sequence (number of images)	RGB(Average Accuracy)	HSI(Average Accuracy)
1 (56)	0.9715	0.9908
2 (128)	0.9371	0.9469
3 (111)	0.9640	0.9688
4 (41)	0.9543	0.9766

Table 3.1: Comparison of the proposed method performance in RGB nd HSI color space

In second experiment, the proposed algorithm is evaluated in detail. For quantitative evaluation, three indicators (as in paper [46]) are calculated: 1) Accuracy; 2) Type I error rate; 3) Type II error rate. The accuracy is defined in the same way in first experiment. The type I error evaluates the cases: when road pixel is falsely classified as road pixels. Let B_s denote the amount of background pixel, B_e the amount of error background pixel. The type I error is defined as:

$$Type\ I\ error = \frac{B_e}{B_s}. \quad (3.15)$$

The type II error evaluates the cases: when background pixel is falsely classified as road pixel. Let R_s denote the amount of road pixel, R_e the amount of error road pixel. The type II error is defined as:

$$Type\ II\ error = \frac{R_e}{R_s}. \quad (3.16)$$

The average of these three indicators on the 336 images are resumed in Table 3.2. N_{image} denotes the number of images refined in the sequence. As shown, we that the amount of refined image in sequence 2 is more than the others. That is because there exist many buildings around the road. It makes the work to distinguish the sidewalk from road get harder.

	Propose approach			
Sequence (number of images)	Acc	Type I	Type II	N_{image}
1 (56)	0.9908	0.0057	0.0228	28
2 (128)	0.9469	0.0122	0.1583	87
3 (111)	0.9688	0.0252	0.0500	53
4 (41)	0.9766	0.0180	0.0466	23

Table 3.2: Performance of road detection considering the proposed approach

In last experiment, the proposed algorithm is compared with the approach proposed in [25] based only on illumination invariant. To compare the results, the same indicators are adopted. The results are shown in Table 3.3. We notice that the most significant improvement is obtained for the sequence 2, for which the percentage of refined images is the greater. Finally, it is to notice that the proposed approach outperforms the only illumination-invariant based algorithm for each indicator.

Fig.3-16 shows some experimental results obtained by the proposed approach. As shown, we can see that the proposed algorithm can detect road pixels well. Nevertheless, the first and third images illustrate that some patterns (as white line in road centre) on the road can affect the performance of the proposed algorithm.

Fig.3-17 compares some experimental results obtained by the proposed approach

	Approach proposed in [25]		
Sequence (number of images)	Acc	Type I	Type II
1 (56)	0.9582	0.0080	0.1383
2 (128)	0.8930	0.0227	0.3347
3 (111)	0.9283	0.0188	0.2148
4 (41)	0.9437	0.0180	0.0466

Table 3.3: The performance of road detection considering the method proposed in paper [25]

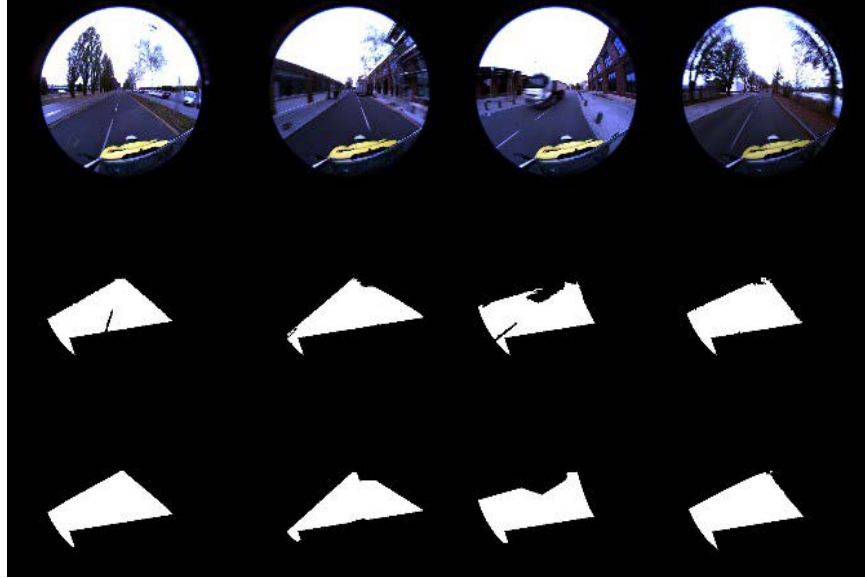


Figure 3-16: Experimental results. (Top row) Original image; (Middle row) Detection result; (Last Row) Ground truth.

and the method proposed in paper [25]. The lost road part in the second image of middle row (method in paper [25]) is filled as it can be seen in the second image of the last row (our method). The redundant road part in the first image of middle row is removed with our method. All above results prove that the combination of various information of image can permit to improve road detection.

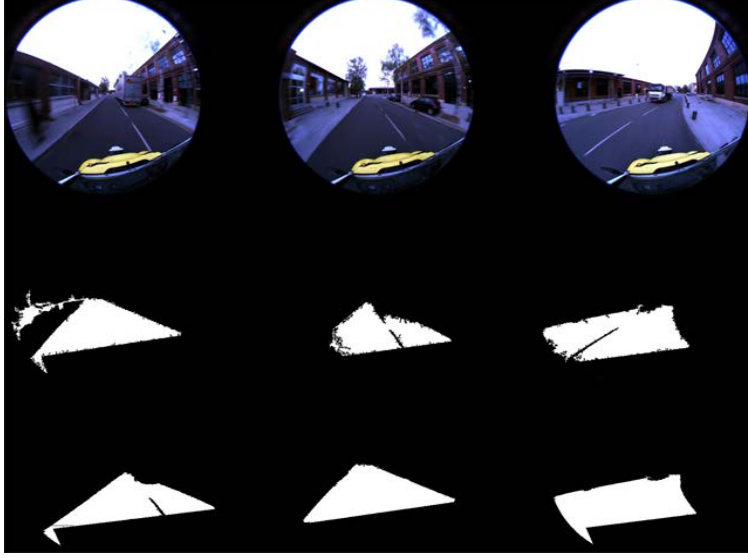


Figure 3-17: Experimental results. The images of the first row are original images, the second row are results obtained using the approach proposed in paper [25], the third are the results obtained by the proposed method in this paper

3.7 Conclusion and Future Works

In this chapter, we presented an efficient algorithm for road detection in outdoor scenarios. The proposed method combines HSI color information, illumination invariant image and LRF measurements to extract road area from the fisheye image. It firstly conducts preliminary road detection in illumination invariant image. A coherence checking based on LRF measurements and the amount of road pixels is then applied to derive the road image. Unqualified images are finally refined in HSI color space. Compared to the road detection method only based on illumination invariant image, the experiment results have shown that the proposed approach can permit to achieve some improvements for road detection. For future work, we can attempt the combination of different sorts of color space. We constated that illumination invariant may also lose the color or intensity information of objects when it reduces the effect of shadow. Searching for a proper color descriptor should bring some improvement.

Chapter 4

Multisensor Based Obstacles Detection in Challenging Scenes

4.1 Introduction

Obstacles detection is a broad research field and plays an important role in industry. For autonomous vehicles, it aims at making the passenger to stay in a safe situation. To achieve that goal, the vehicle has to perceive the obstacles around itself. In outdoor scenarios, obstacles detection is a tough work because the background is varying with the traveling of the vehicle and the appearance of obstacles is not predictable. For the past decade, many researchers put great efforts into settling these problems in the intelligent vehicle research community. Some researchers pay their attentions to the obstacles detection in daylight time, and others focus on night obstacles detection. In this chapter, multi sensors (fisheye camera, 2D LRF and GPS receiver) based obstacle detection method is proposed to handle a challenging case like in Fig.4-1. In this case, the obstacles to detect have serious motion blur problem.

The rest of this chapter is organized as follows: Section 4.2 introduces the state of the art. Section 4.3 presents the framework overview of the proposed approach. Section 4.4 describes in details how to extract potential obstacle areas. Section 4.5 shows how to locate the real obstacle position from these potential obstacles areas. Section 4.6 gives the real data experimental results. Conclusions are presented in

4.2 State of the Art

According to the used equipments, most of the proposed approaches for obstacle detection can be divided into three types: passive sensors based, active sensors based and both of them based.

Generally, camera based approaches are in common use for obstacles detection. Among these approaches, some of them employ a priori knowledge of obstacles such as color [47], vertical and horizontal edges [48], texture [49], and symmetry of objects [50] to separate the obstacles from the background. These methods often have the benefit to be simple and efficient, but they are not robust to illuminance and weather changes. Some methods are based on the estimation of ego-motion [51][52][53]. In paper [52], the authors propose a method based on the motion trace of feature points to detect and track target vehicles ahead with the same running direction as the observer vehicle. Low level features such as corner, intensity and horizontal line are extracted firstly. Then, these features in each frame are projected vertically to form a 1-D profile. All consecutive 1D profiles along the time axis are linked to generate features motion trace image. In this image, a motion model based on hidden Markov model(HMM) is used to separate the background and target vehicle. This method is restricted to detect vehicles ahead with the same running direction as the observer vehicle. Besides, a proper motion probability model of target vehicle in HMM is hard to decide and the threshold for the separation of the line of background and target vehicle in 1-D profile is based on a experiential value. In paper [53], a method based on optical flow residual is used to detect the obstacle rear to the vehicle. It firstly find feature correspondence in consecutive frames. Then, these features are transformed to bird eye view (BEV). In the BEV image, they are classified into ground/non-ground plane, and the features belonging to ground plane are used to estimate the ground plane ego-motion. Finally, based on the estimation of ground ego-motion, a residual motion map with respect to the ground is calculated. Using this map, the

moving obstacles around vehicle can be identified. Generally, road plane has very few texture features and is difficult to extract many saliency features. If feature points are not sufficient, it is hard to estimate precisely the entire road ego-motion. Other researchers utilize stereo vision [54] [55] [56] to estimate the position of obstacles. In paper [55], the authors construct a V-disparity image based on stereo image pair to detect obstacles on the road. The drawback of this method is that it assumes road is dominant along the image rows and it can be sensitive to roll angle changes. In paper [56], the authors suggest to detect obstacles based on the density of digital elevation map. But this method can't discriminate the sidewalk around free road area. In practice, it is a dangerous case.

In paper [57], a LRF is utilized for obstacles detection task. LRF is independent to illumination change. It is an active sensor which can provide reliable and high accuracy range measurements. The geometrical figures consisting of LRF measurements are employed to represent vehicles, and predict their location using an extended Kalman filter (EKF). However, in fact, it's hard to describe all kind of objects around the vehicle merely using a few simple geometrical shapes. In paper [58], obstacles are detected using an occupancy grid map which is transformed into regularly spaced grid of cells. Nevertheless, the resolution of obstacles depends on the grid map. The higher resolution grid map is, the more memory it takes.

In paper [59], both LRF and camera are used to detect and recognize objects in front of a vehicle. The LRF measurements provide the regions of interest (ROI), and a classifier based on support vector machine (SVM) is applied to recognize the content in ROI. Our research is belonging to this line.

Nevertheless, all of the methods described above don't consider the motion blur case. If motion blur emerges in image, many salient features (corner, SURF, FAST) will become invalid, and put the relevant methods in failure. Most of motion blur stems from objects movement in a scene during a long time exposure. In weak lighting scene, the long time exposure of camera often occurs. It means motion blur has a high probability to appear in this case.

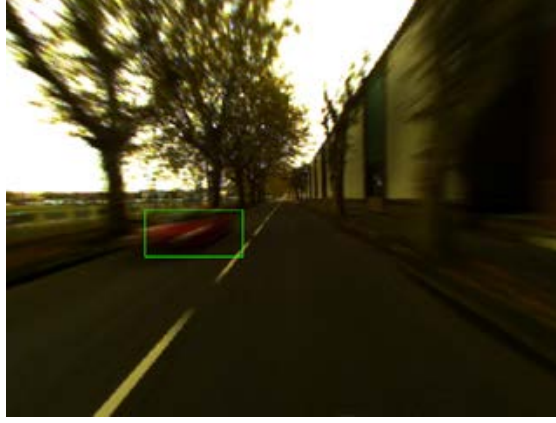


Figure 4-1: The challenging case: motion blur effect for the red vehicle in the image

4.3 Overview of the Proposed Algorithm

The obstacle detection algorithm proposed in this thesis is shown in Fig.4-2. It firstly copes with the fisheye image. There exist two parallel flows for the fisheye image. In the right one, the road detection method described in chapter 3 is firstly applied. To facilitate the work, the distortion in road detection result is then removed. After that, a geometrical transformation named inverse perspective mapping (IPM) is applied to the undistorted road image to remove the perspective effect and to form a new image I_r . In the left flow, distortion in the fisheye image is firstly removed. And then, the same geometrical transformation as the right flow is applied to the undistorted image to form a new image I_o . In the image I_o , the central lane marker is then detected. The derived central lane marker is used as base line to map the road model into I_o . Road model is built using Geographic Information System (GIS) information obtained from GPS and Openstreet map. This road model can determine the road boundary in the image I_o . The derived road boundary is then mapped into the image I_r to extract several possible regions of obstacle presence. In the last step, the real obstacle field is determined from these candidate regions using LRF measurements.

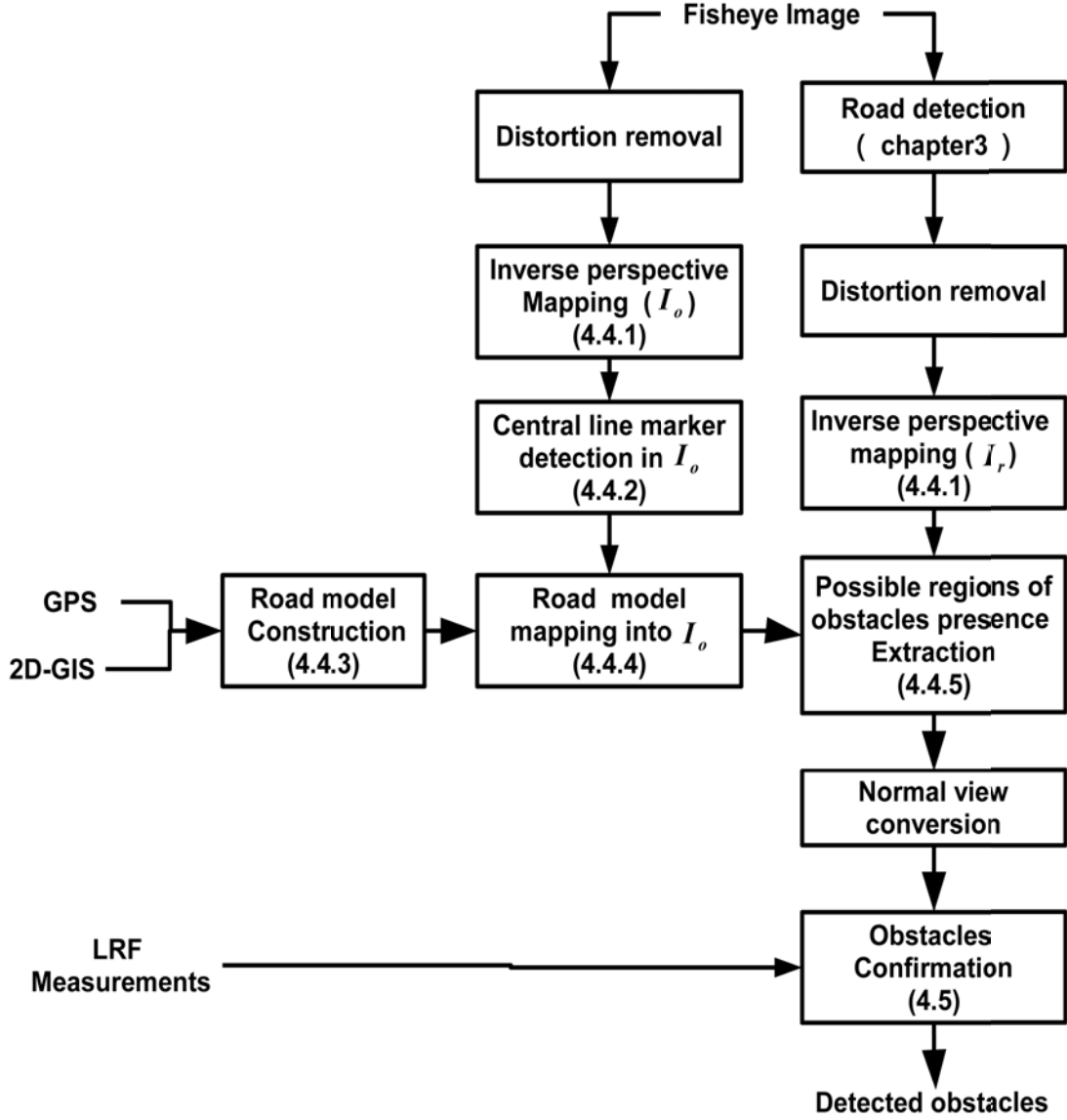


Figure 4-2: Processing flow chart for obstacle detection

4.4 Possible Region of Obstacle Presence Extraction

As illustrated in Fig.4-2, there are several procedures before the possible obstacle region extraction step. We introduce some important steps in the following sections.

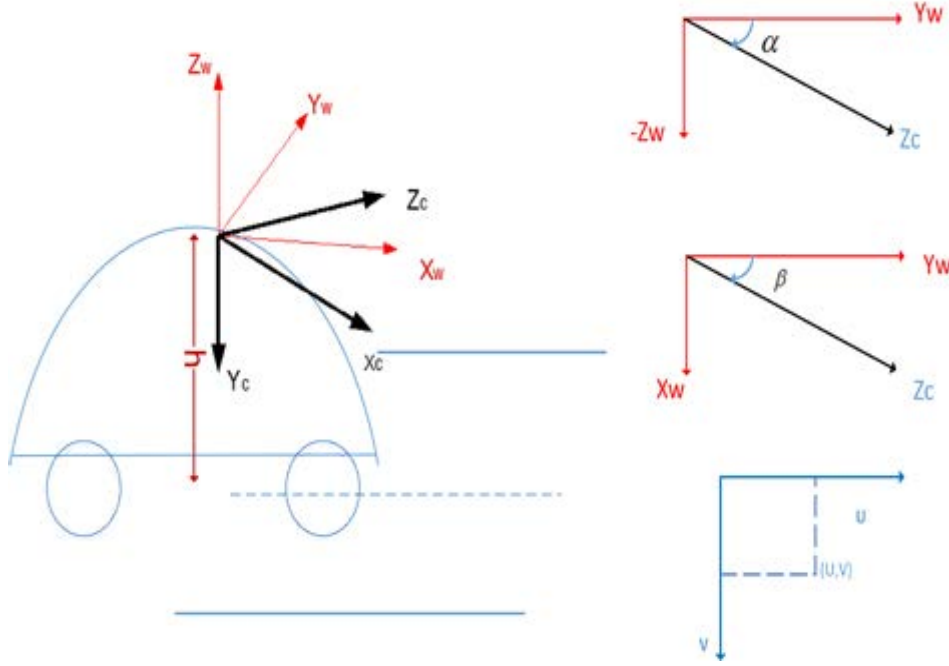


Figure 4-3: IPM coordinate system

4.4.1 Inverse perspective mapping

Inverse Perspective Mapping (IPM) is often used in many literatures to remove the perspective effect. In low level, the perspective effect associates different meanings to different image pixels. But for the entire image, it gives rise to the geometrical distortion of an object in the image. IPM technique resamples each pixel in original image and maps it in a different position to create a new image (IPM image). As done in [60][61], given an assumption of a flat road, this transformation can be expressed through the camera intrinsic and extrinsic parameters. Let $C_w(X_w, Y_w, Z_w)$ denote the world frame, $C_c(X_c, Y_c, Z_c)$ the camera frame and $C_i(u, v)$ the image plane. The relationship between C_w and C_c are shown in Fig.4-3. The origin points of C_w and C_c overlap. $X_w Y_w$ plane is parallel to ground plane. of Camera frame X_c lays on $X_w Y_w$ plane, and optical axis Z_c is allowed for α and β the pitch and yaw angle but no roll. h is the height between viewpoint and ground plane. Any point $P_i(u_x, v_y, 1, 1)$ in image plane, it can be mapped to the ground plane point P_g by the following

transformation:

$$P_g = h \begin{bmatrix} \frac{-\cos\beta}{f_u} & \frac{\sin\alpha\sin\beta}{f_v} & \frac{p_u\cos\beta}{f_u} - \frac{p_v\sin\alpha\sin\beta}{f_v} - \cos\alpha\sin\beta & 0 \\ \frac{\sin\beta}{f_u} & \frac{\sin\alpha\cos\alpha}{f_v} & -\frac{p_u\sin\beta}{f_u} - \frac{p_v\sin\alpha\cos\beta}{f_v} - \cos\alpha\cos\beta & 0 \\ 0 & \frac{\cos\alpha}{f_v} & -\frac{p_v\cos\alpha}{f_v} + \sin\alpha & 0 \\ 0 & -\frac{\cos\alpha}{hf_v} & \frac{p_v\cos\alpha}{hf_v} - \frac{\sin\alpha}{h} & 0 \end{bmatrix} \begin{bmatrix} u_x \\ v_y \\ 1 \\ 1 \end{bmatrix} \quad (4.1)$$

where f_u, f_v are the horizontal and vertical focal lengths, p_u, p_v are the optical center coordinates. Similarly, to convert any point $P_g(g_x, g_y, -h, 1)$ in IPM image to normal view image, the following transformation can be applied:

$$P_i = \begin{bmatrix} f_u\cos\beta + p_u\cos\alpha\sin\beta & p_u\cos\alpha\cos\beta - \sin\beta f_u & -p_u\sin\alpha & 0 \\ \sin\beta(p_v\cos\alpha - f_v\sin\alpha) & \cos\beta(p_v\cos\alpha - f_v\sin\alpha) & -f_v\cos\alpha - p_v\sin\alpha & 0 \\ \cos\alpha\sin\beta & \cos\alpha\cos\beta & -\sin\alpha & 0 \\ \cos\alpha\sin\beta & \cos\alpha\cos\beta & -\sin\alpha & 0 \end{bmatrix} \begin{bmatrix} g_x \\ g_y \\ -h \\ 1 \end{bmatrix} \quad (4.2)$$

Fig.4-4 shows an IPM example. The left image is the undistorted image of an original fisheye image, and the right image is the corresponding IPM image. As illustrated in Fig.4-4(b), the lane width is fixed and appears horizontal.

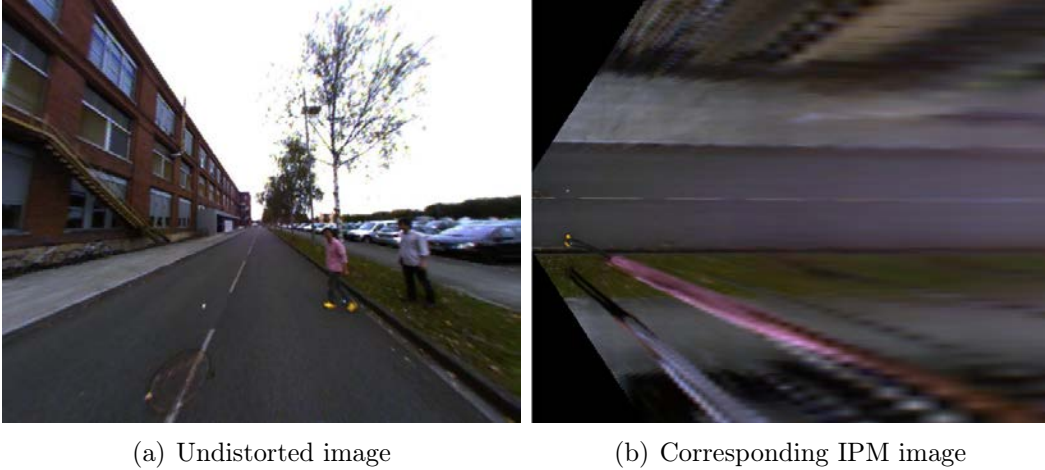


Figure 4-4: IPM example: In IPM image, the borders of road appears parallel.

4.4.2 Central line marker detection

The lane central marker is used as a reference line in later road model mapping section. In paper [62], the authors propose to map the road shape model to driver's view. However, it involves a transformation in four different kinds of coordinate systems. Such transformation often brings many errors, and meanwhile, it is hard to determine all of transformational parameters. To avoid this problem, we propose to map road shape model to the image I_o (IPM image after distortion removal). However, this mapping requires some marks on the road. Road boundary and lane central marker are the two choices. In our method, the lane central marker is the favorable one because it is not often covered by other objects in the traveling journey. Given two-lane straight street, the lane marker detection flow chart is shown in Fig.4-5.

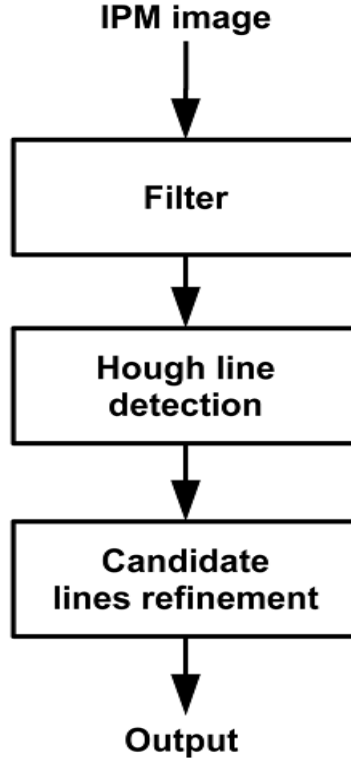


Figure 4-5: Lane marker detection flow chart

Filter

The image I_o is firstly sent to a filter that aims at getting a high response to the lane markers and removing irrelevant pixels. Two separable Gaussian kernels as in paper [61] are used to compose the filter. One for vertical direction, and another for in horizontal direction. The horizontal direction is a smoothing Gaussian kernel with expected width of lane segment. The filter for vertical direction is a second derivative of Gaussian kernel with the height of the lanes. One example of the performance of the proposed filter is shown in Fig.4-6. As seen, the lane marker have higher response

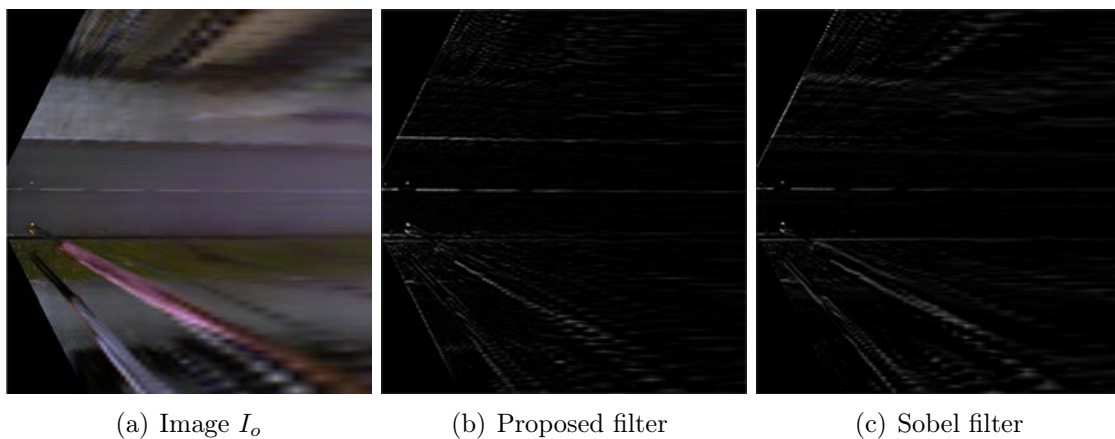


Figure 4-6: Filter performance

in Fig.4-6(a) than in Fig.4-6(b). After that, the remaining pixels are sorted in descent order by their responses. The top $T_{thr}\%$ (empirical value:5%) pixel are binarized to 1, and the rest of pixels is set to 0. The new obtained image is defined as I_f .

Line detection

This step is referred to detect the lines in I_f . The standard Hough line detection technique is applied. For Hough transformation, a line is expressed in the corresponding Polar system. It means a point (x, y) in Cartesian coordinate system corresponds to a pair (r, θ) in Polar coordinate system. So, for any point in image, a sinusoid curve corresponds. A line can be detected by finding the number of intersections between these sinusoid curves. High the number of intersections is, high the number of points

the line has. A minimum number of intersections is set to rule out the unqualified line.

Candidate lines refinement



Figure 4-7: The red area in the image is the predefined fix area. The yellow line expresses the center of this area

The proposed approach requires to detect the central line markers. In the image I_o , the road often occupies the nearby area of the center of the image. According to this property, two conditions are exploited to search the central line marker. The first one is a fixed predefined area nearby the center of the image I_o which is used to screen the candidate lines. If the position of the detected line is within this area, it is reserved. Otherwise, it is discarded. For the reserved lines, we just keep the lines nearest to the center of the defined area. If there is only one line to fit above condition, this line will be treated as the central line marker candidate. If there are more than one line are possible, the average line between these lines is calculated as central line marker candidate.

The second condition is a line tracking model. In practice, some parts of a road may haven't obvious markers or have other similar figures (zebra stripes). It brings the difficulties to detect central line marker. To make the approach get more robust, a tracking model is explored. In fact, the tracking model is a cache which conserves the central line marker positions of the latest five frames. Given l_{ave} the average position of central line marker positions of the latest five frames, a candidate is accepted if it

fits the the following condition:

$$0.9 * l_{ave} < l_c < 1.1 * l_{ave} \quad (4.3)$$

where l_c is the position of the central line marker candidate. Otherwise, l_{ave} is considered as the central line marker. However, if five successive central line marker candidates are outside this range, the cache will be cleaned and updated.

4.4.3 Road shape model computation

Road shape model reflects the geometry of the road such as left turn, right run, or cross-like junction. This contextual information is a very useful clue in many applications. To acquire ahead road shape model, driving direction information is necessary. To obtain this information effectively, it needs to transform GPS coordinate from 3D world geodetic system to 2D local surface system.

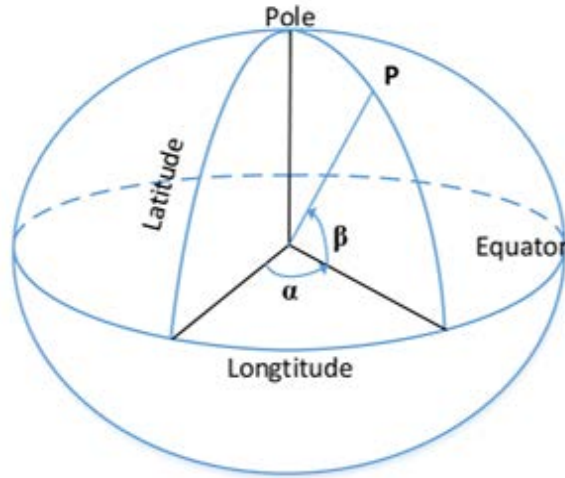


Figure 4-8: WGS84 coordinate system

Transformation of GPS coordinates

GPS is a fully operational optimum positioning system and provides accurate, continuous position and velocity information to the users equipped with GPS receiver. It

uses world geodetic system (WGS84) as reference coordinate system. As illustrated in

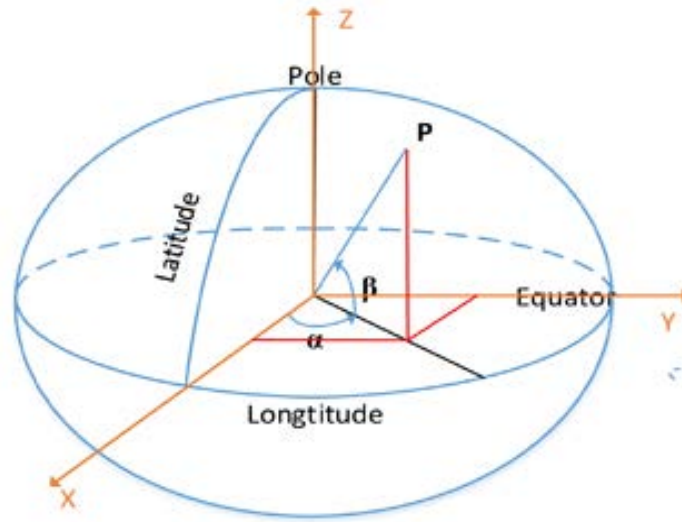


Figure 4-9: ECEF coordinate system

Fig.4-8, WGS84 is based on an ellipsoid with minor radius at poles and major radius at equator. Its datum is described by the longitude, latitude and elevation. The data

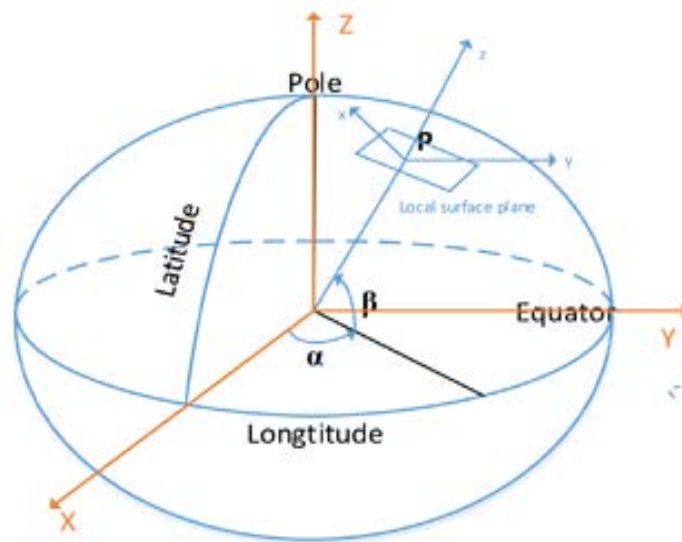


Figure 4-10: Local surface plane

in this form is not convenient for our use. The data in local 2D surface form is our desired target. To achieve this goal, the original GPS data are preprocessed by the

following steps:

1. Transformation from WGS84 to a 3D cartesian coordinate (X, Y, Z) in an earth centered and earth-fixed (ECEF) system as shown in Fig.4-9.

2. Converting the 3D cartesian coordinate into a local 2D surface plane (See Fig.4-10) by extended Lambert II projection model.

The details for these coordinate conversion algorithms are given in appendix A.

Openstreetmap

GIS is the most important factor for building the road shape model. It is extracted from OpenstreetMap database. OpenstreetMap is a project to build a free geographic database of the world. It gathers all sorts of information including building, road, waterways and so on. There are three primitive type data combined with free form tagging scheme in OpenstreetMap to describe geographic feature: nodes, ways and relations. The first two types play an important role in the proposed approach and are introduced in details.

Nodes are fundamental elements. They are points in space and the only primitive to have position information [63]. The other two primitives are based on the nodes. Each node has its latitude and longitude stored in decimal format up to 7 decimal places. This guarantees the latitude and longitude resolution can reach round $1cm$ at the equator or $0.6cm$ at Greenwich. This accuracy is enough for most applications. Fig.4-11 shows several nodes in OpenstreetMaps.

Ways are lists of nodes arranged in order. They can be used to describe something with linear features, such as roads and paths. In the map, a way must have at least two nodes and a maximum of 2000 nodes. The ordering of these nodes in a way is preserved, so the way often has direction. The direction is from the first node to the last one. If the first and last nodes of a way are the same, the way forms a closed area. For a more complex shape than a simple polygon, this can be done using several ways and a relation to link them. Additionally, for a road, the way is usually placed down the center line of the physical feature.



Figure 4-11: OpenstreetMap. (a) The map under editor view. The red circle points are nodes. (b) The map in normal view

Road shape model construction

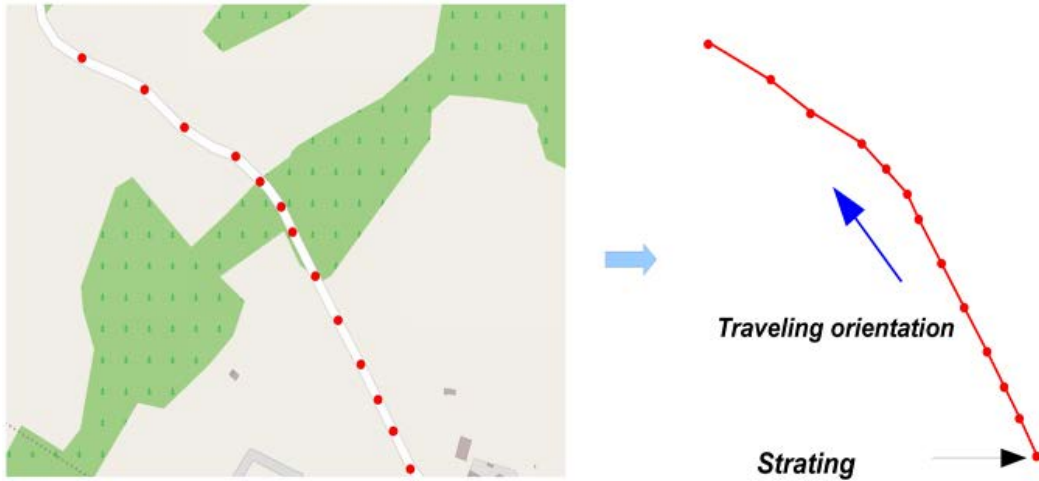


Figure 4-12: Road skeleton extraction

The road shape model contains the road skeleton model and road delimiter model. The road skeleton model is a link which concatenates the road nodes within a map database one by one (See Fig.4-12). However, great gap between two adjacent road nodes often appears in map database. Before concatenating the road nodes, it needs to operate a linear interpolation. Firstly, the GPS data are used to pick out which road

the target vehicle is traveling within the map and to identify traveling direction on this road. Then, the corresponding road nodes are extracted from the map database (OpenStreetMap [64]). Let R_c denote the road where the vehicle is traveling, $R_i (i = 1, 2, \dots, N)$ the i -th road node of R_c in the map database. The linear interpolation operation is implemented between two adjacent road nodes if the following condition is satisfied.

$$d_{R_i R_{i+1}} < m_\lambda \quad (i = 1, 2, \dots, N - 1) \quad (4.4)$$

where $d_{R_i R_{i+1}}$ is the distance between R_i and R_{i+1} nodes, m_λ is a threshold set manually. In our approach, it is set to 2 (unit meter). After that, the closest node to the vehicle location is chosen from the road nodes and interpolation nodes and is used as the starting point for the road skeleton construction. From this point, the interpolation nodes and road nodes are concatenated one by one along the traveling orientation to constitute the skeleton of the road ahead the vehicle (See Fig.4-12).

The road delimiter model consists of a set of lines piecewise parallel to the road skeleton. The width of these piecewise parallel lines depends on the road width information estimated according to the road attributes in the database and additional countries national road legislation. The road model is shown in Fig.4-13.

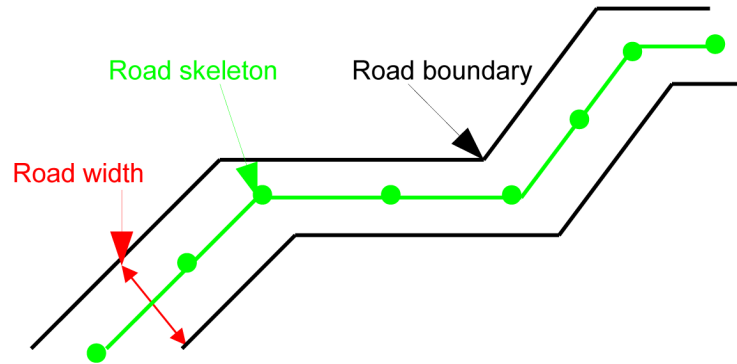


Figure 4-13: Road model

4.4.4 Road model mapping

Road model mapping means to map road geometry into the image I_o . Based on the derived central lane marker, the mapping procedure is actually an aligning procedure that it just needs to rotate the road skeleton for an angle. The above mapping process is illustrated in Fig.4-14.

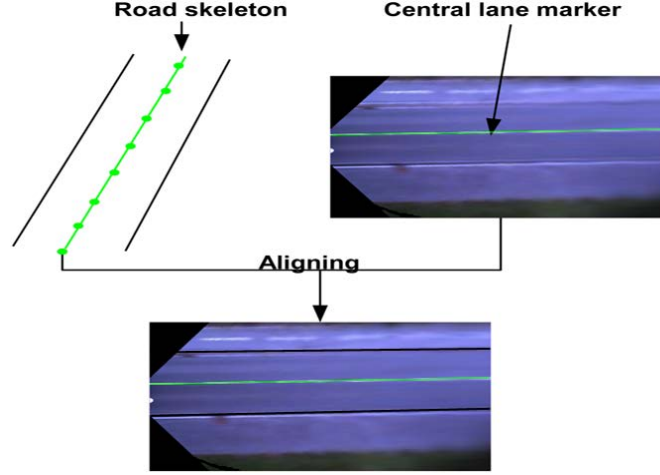


Figure 4-14: Road model mapping

4.4.5 Potential obstacle detection

Potential obstacle detection is based on the image I_o image with previously described road geometry and the image I_r obtained by applying IPM transformation to the undistorted road detection results. Because I_r is actually the road detection result of I_o , they share the same road geometrical information. So, the road boundary in I_r can be found out. The road pixels within the boundary are discarded, and the non-road pixels are conserved (Fig.4-15(a)). The conserved non-road pixels are then grouped by using a connect-component algorithm. Polar histogram (Fig.4-15(b)), similarly to paper [65], is used to filter these groups by the following two conditions:

$$\begin{cases} G_p > K_{gp} \\ G_g > K_{gg} \end{cases} \quad (4.5)$$

where G_p is the peak number of a group in polar histogram, G_g is the pixels number of the group, K_{gp} and K_{gg} are rough thresholds to eliminate the small groups. In our application, these two parameters are set to 25 and 75 (empirical value). A group is conserved if the above two conditions are satisfied. As a result, the remaining group fields represent the possible regions of obstacles presence.

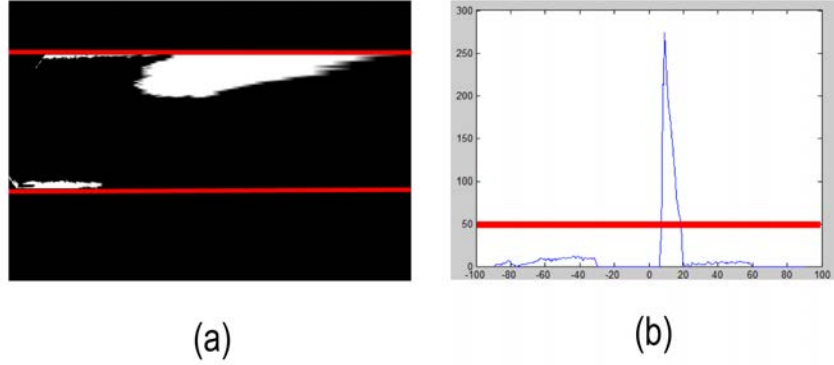


Figure 4-15: (a) Red lines in the image I_r shows the boundary of the road. Non-road pixels within the road boundary are conserved (b) Corresponding polar histogram

4.5 LRF based Obstacles Confirmation

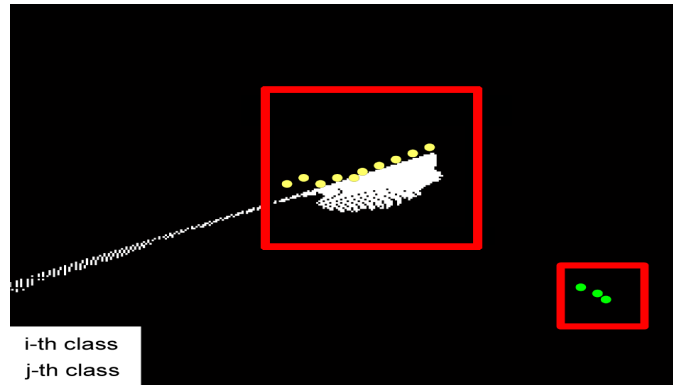


Figure 4-16: The searching windows in normal view image

Obstacle confirmation refers to cut the "tail" pixels which don't belong to ob-

stacles (ie. belonging to the road)(See Fig.4-16). LRF measurements are used to implement this task. Firstly, we suppose that the extrinsic parameters between the fisheye camera and LRF are known and the IPM image containing the derived potential area of obstacles is converted to normal view. Let denote I_{nv} this normal view image and $L_i(i = 1, 2, \dots, K)$ the corresponding reprojection pixel of the i -th LRF measurement. In our work, we find that the 8-neighbor of L_i often belongs to a same object. To extend the useful information, the 8-neighbor are treated as the correspondence, and they share the LRF measurement of L_i . The correspondences lying beyond the boundary of the road are ruled out by using the acquired road geometry information. For the remaining correspondences, a cluster algorithm is applied to cluster. Let denote L_j the j -th correspondences, L_{j+1} the adjacent correspondences of $j - th$ correspondences. If the following two conditions are satisfied, the two adjacent correspondences are put in the same class:

$$\begin{cases} |\theta_{L_{j+1}} - \theta_{L_j}| < \theta_L \\ |M_{L_{j+1}} - M_{L_j}| < M_L \end{cases} \quad (4.6)$$

where $M_{L_{j+1}}$ and M_{L_j} are the LRF distance of the two adjacent correspondences respectively, $\theta_{L_{j+1}}$ and θ_{L_j} are the corresponding LRF angle of them respectively, θ_L and M_L are empirical thresholds. In experiment, θ_L and M_L are set to 4 and 0.5 meter respectively. As a result, these different correspondences gather into several different classes. For each class, a rectangle filter window (See Fig4-16) is assigned to each of them. The center of this window is located at the centroid of this class. The width and height of the window are based on the minimum bounding box of the class. They are determined by following equations:

$$\begin{cases} W_{wi} = 1.2 * W_{ci}; \\ H_{wi} = 1.2 * W_{ci} + H_{ci}; \end{cases} \quad (4.7)$$

where W_{wi} and H_{wi} are the width and height of the window of the i -th class respectively, W_{ci} and H_{ci} are the width and height of the minimum bounding box of i -th

class. A pixel in the normal view image I_{nv} belongs to an obstacle if it is within the corresponding filter window. Otherwise, the pixel is discarded. The minimum bounding box which contains remaining pixels and the corresponding class are the final output.

4.6 Experiments

The experimental data consist of a road database obtained from openstreetmap and two sequences (sequence1: 231 frames, sequence2: 171 frames) with the corresponding LRF measurements and GPS data captured by our experimental vehicle. The ground truth is labelled manually. For the sequence 1, the results obtained using our method are compared with the results obtained with two other methods proposed in [66] and [67]. Public codes for the two methods are available on the internet. Some results of these three approaches are shown in Fig.4-17. We can notice that the appearance of the vehicle in Fig.4-17(a) is blur, especially for the front part of the vehicle. Meanwhile, we can see that all the outcomes of the method in paper [66] are incorrect and the method proposed in paper [67] detect nothing. The reason that make the two methods invalid is the loss of visual feature. Fig.4-17(d) shows that, to an extent, our method can overcome this issue and give more reasonable results.

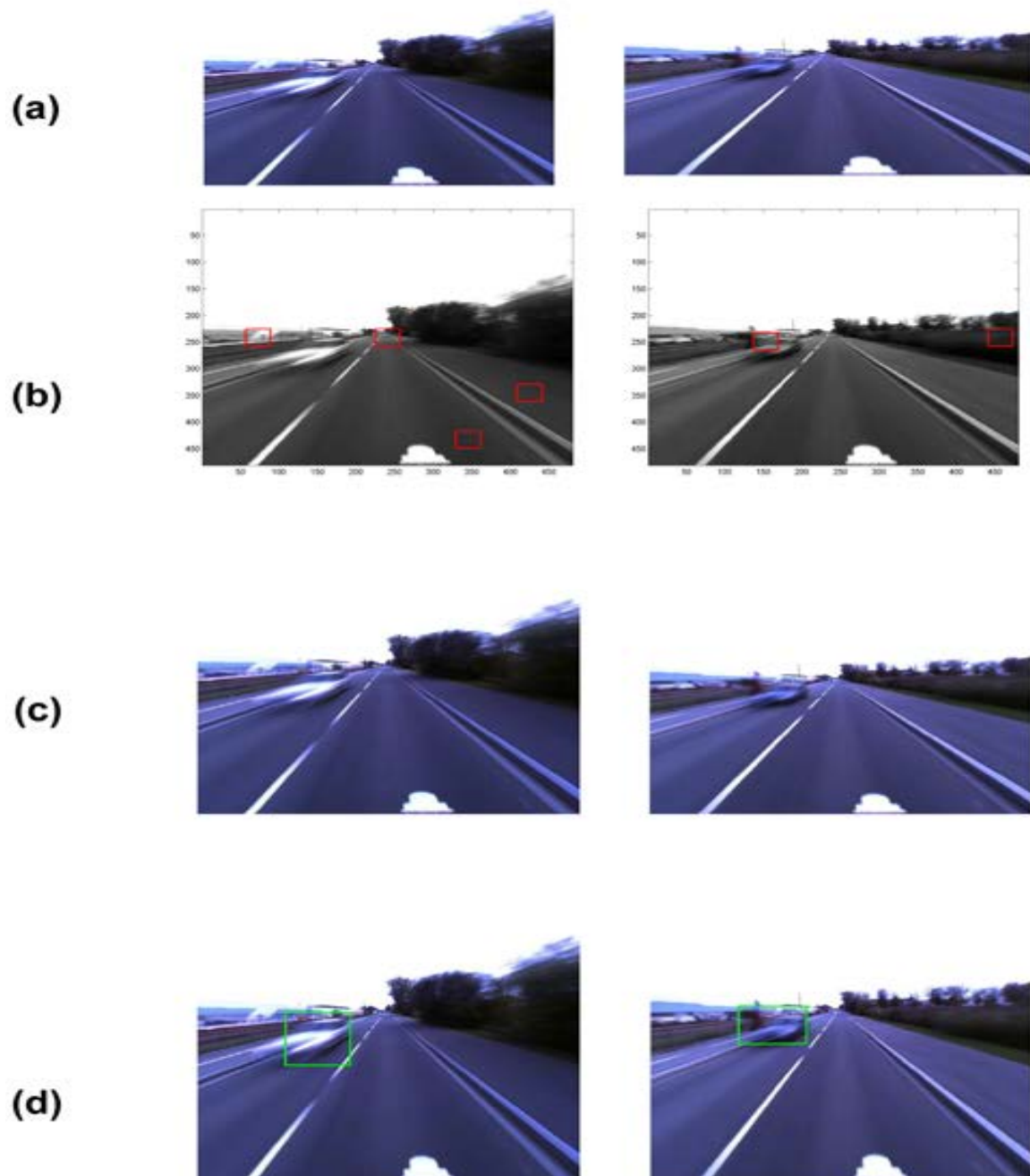


Figure 4-17: (a) Original image (b) Results obtained with the approach of paper [66] (c) Results obtained with the approach of paper [67]. (d) Results obtained with our approach.

For the sequence 2, the results obtained using our method are compared with the results obtained with the method proposed in [66]. Some results of the two approaches are shown in Fig.4-18. As illustrated in Fig.4-18(a), the appearance of the pedestrian is not very clear, especially for the leg. Meanwhile, in Fig.4-18(b), similar with the experiment results in sequence 1, the method proposed in paper [67] still can't detect the object. From the Fig.4-18(c), we can see that our method can output more reasonable results. It is noticeable that, in Fig.4-18(c), the right detection result lose partial leg information of the pedestrian. That is because, in road detection procedure, the lost leg part are treated as road. We also evaluate the influence of LRF measurements based confirmation on the performance of the proposed approach. The results are shown in Table 4.1. For sequence 1, one can notice that the LRF measurements based confirmation improve the correct rate greatly. However, the hit rate declines slightly due to the shift of LRF measurements of obstacles. For sequence 2, we can see that the two criteria are higher with LRF measurements based confirmation case than without LRF measurements based confirmation case. The reason which makes hit rate better is that the super correspondence can provide additional object information when the motion blur happens at leg of pedestrians and super correspondence lay on the leg. In other words, without LRF measurements based confirmation, the leg part in the image is often treated as road making the hit rate worse.

Condition	Hit Rate	Correct Rate
Without LRF measurements based confirmation 1	0.8076	0.5297
Without LRF measurements based confirmation 2	0.6076	0.4297
With LRF measurements based confirmation sequence 1	0.7255	0.8716
With LRF measurements based confirmation sequence 2	0.7518	0.7762

Table 4.1: The performance of the proposed approach with and without LRF measurements. $Hit\ Rate = \frac{The\ number\ of\ the\ correct\ detected\ obstacles}{The\ total\ number\ of\ the\ ground\ truth\ obstacles}$ $Correct\ Rate = \frac{The\ number\ of\ the\ correct\ detected\ obstacles}{The\ total\ number\ of\ the\ detected\ obstacles}$

Fig.4-19 shows some examples of such failure case. As illustrated in the first row of Fig.4-19, the pedestrians around the road boundary, the proposed method can't

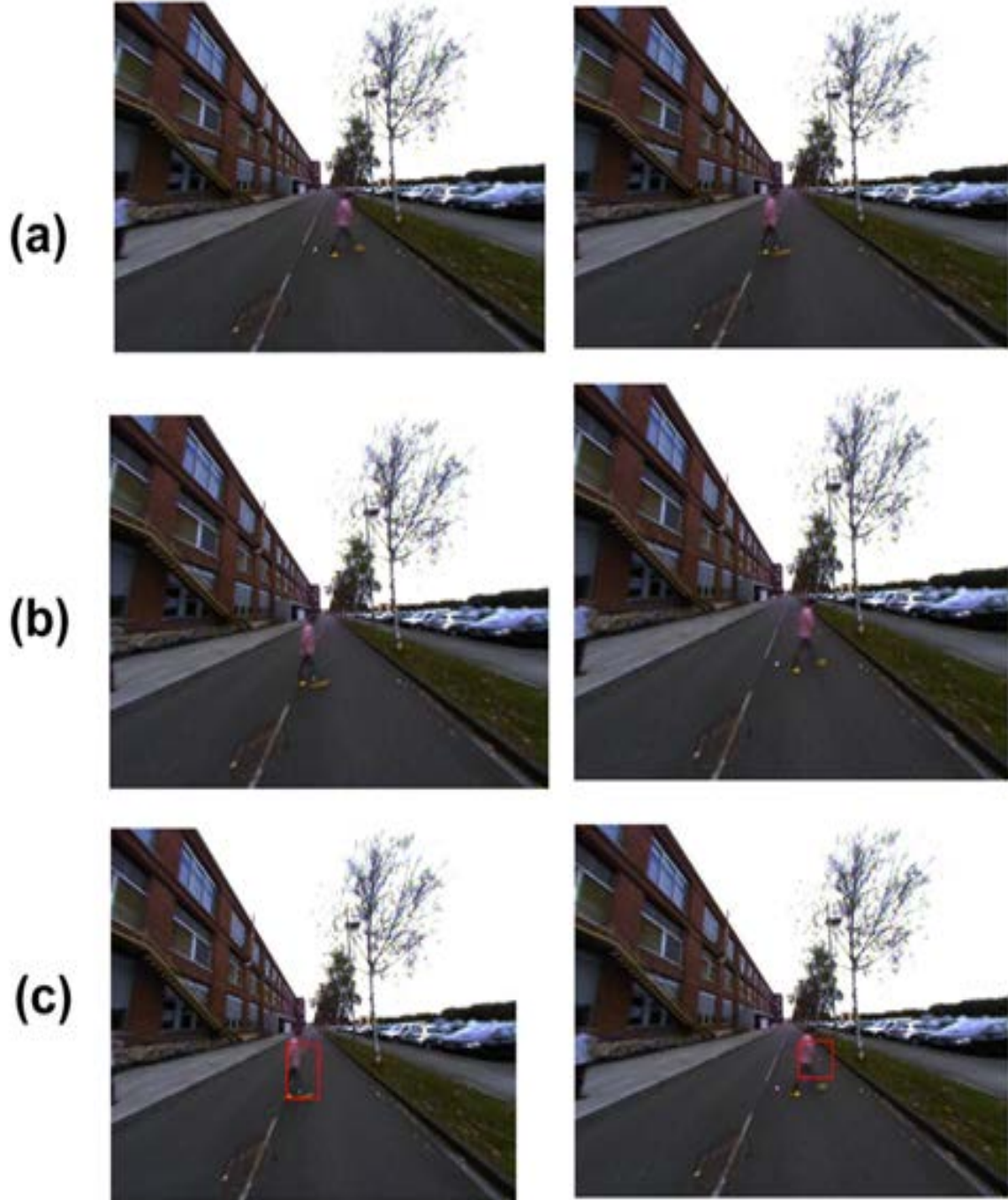


Figure 4-18: (a) Original image (b)Results obtained with the approach of paper [66] (c)Results obtained with our approach.

detect the part out of the boundary. From the second row in Fig.4-19, we notice that, for the multi obstacles detection, the algorithm can't separate them when the two objects overlap. As shown in the last row of Fig.4-19, if we lose the LRF measurements

on obstacles, the proposed method becomes invalid.

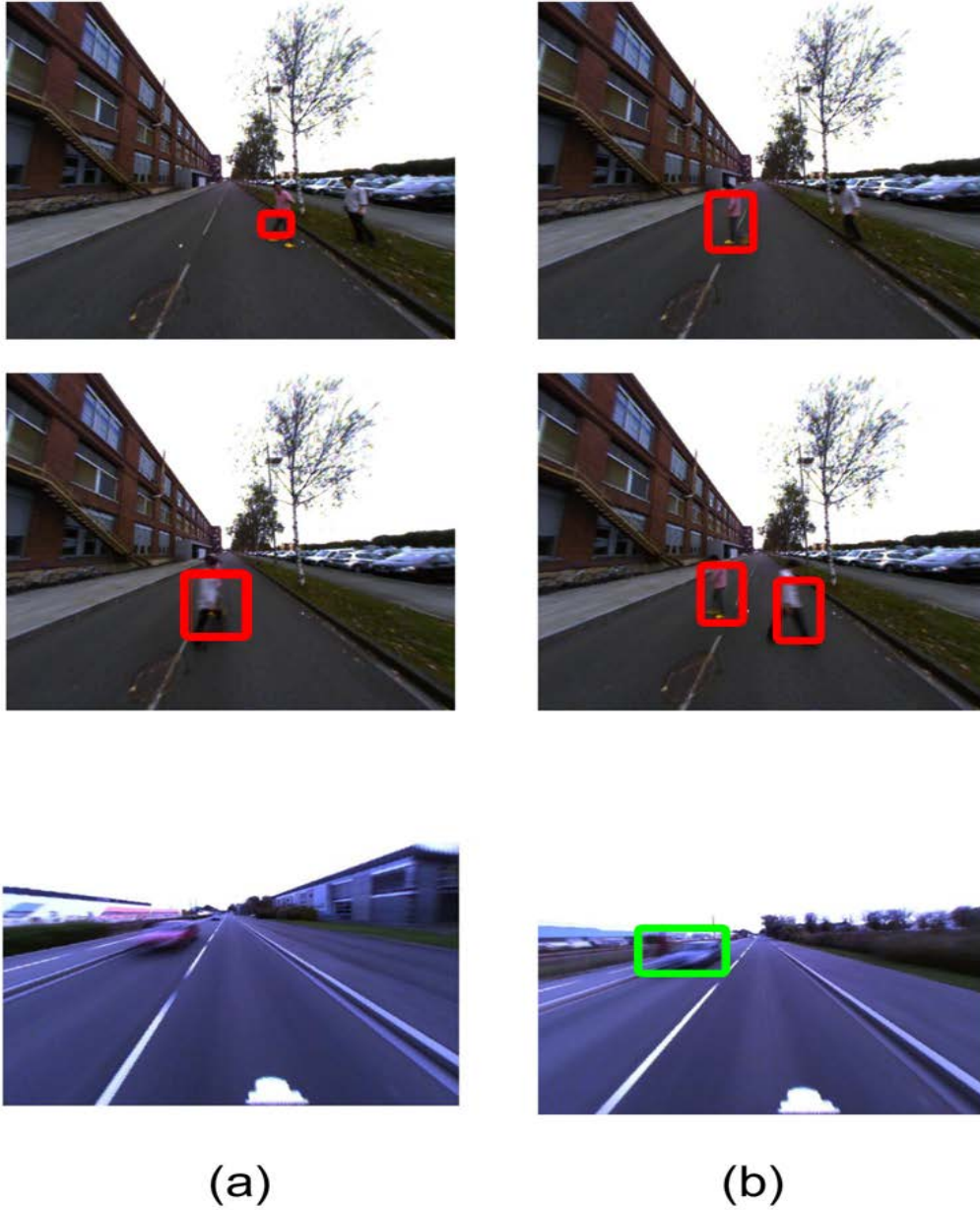


Figure 4-19: (a) error case (b)correct case

4.7 Conclusion and Future Works

In this paper, we presented a novel method to detect obstacles with motion blur case. The obstacle visual features are usually useless. This is a big challenge for many object detection methods. To handle this issue, GIS, road information and LRF measurements are combined with image information. The GIS and road information are applied to extract possible regions of obstacle presence and the LRF measurements are used to filter these regions.

In the possible regions of obstacle presence extraction procedure, the IPM technique is firstly used to remove the perspective effect to the undistorted road image and undistorted fisheye image. Then, a road model ahead of the vehicle is constructed by using GPS data and Openstreet map. After that, this model is mapped into IPM image by aligning the road skeleton with the central line marker in that image. The road boundary can be determined during the mapping procedure. Potential obstacles area can be found out using the road information and relative road boundary information.

In obstacle confirmation step, LRF measurements are used. LRF measurements can provide the width information of target obstacles which can help us to eliminate the disturbance of "tail" pixels such as shown in Fig.4-16. The LRF measurements are firstly screened by the road boundary. And , the remaining data are then grouped into several classes. In each class, a filter window based on the bounding boxes of this class is built. This window is then applied to filter the "non-obstacle" pixels.

For future works, the following aspects could be expected to get improvements:

- 1.Currently, all the road nodes in the vehicle running road are gathered manually. For future works, we would like to develop a method to extract these nodes automatically.

- 2.For the LRF measurements classification, the geometrical shape which is made up of LRF measurements on the objects can be exploited to help grouping these measurements. For most of the cars, their geometrical shapes consisting of LRF measurements on them are similar. This property is useful to us to classify the LRF

measurements.

3. For roads without lane marker, it is hard to map the road model shape into the image. To deal with this case, it is a considerable choice to detect the road boundary.

Chapter 5

Objects tracking based on small-region growth

5.1 Introduction

In this chapter we will present a novel object tracking algorithm. Because of the constant fluctuation of the outside world, it often brings many difficulties for outdoor object tracking. One of problems is hard to get high quality images. For low quality images, many visual based methods may become invalid due to the missing of features information of tracked objects. To address this sort of issue, a method based on LRF measurements and raw pixel information is proposed. It makes use of raw pixel information to locate interesting small regions firstly. And then these small regions are mapped to LRF space. In this space, each small region will growth to a complete region which covers most part of the corresponding target through some growth decision factors. The proposed method is tested in two long-term video which contains 1000 frames. The rest of the chapter is organized as follows: Section 5.2 introduces the state of the art. Section 5.3 presents the framework of the proposed method. Section 5.4 describes in details the initialization step for the method in details. Section 5.5 introduces the frame buffer information function. Section 5.6 presents the way to encode the tracked objects. Section 5.7 describes how to locate interesting small regions. Section 5.8 introduces how to expand an interesting small region to become

a complete region representing a tracked object. Real date experiments are shown in Section 5.9. Finally, a conclusion is presented to end this chapter.

5.2 State of the Art

Although there are many methods regarding objects tracking, most of them can be categorized by their appearance descriptors. In this chapter, we group them into four classes: raw pixel descriptor based, local feature descriptor based, statistical descriptor based and online self-learning descriptor based.

Raw pixel descriptor reflects the basic statical property of object appearance such as color distribution, pixel displacements, correlation information of object pixels and so on. In paper [68], target objects are encoded by their color or intensity values in image. In paper [69], local binary pattern (LBP) technique is used to represent object appearance. However, this sort of approaches is not robust enough due to the accumulation errors during the tracking process. To address this problem, other information are explored. In paper [70][71], optical flow is used to represent targets. Although optical flow information can help to locate moving object in the image, it doesn't work very well in the presence of motion blur and illumination fluctuation. Other researchers [72][73] propose covariance representation based on affine-invariant metric or log-euclidian metric to capture the targets in image. This sort of approaches is computational efficient and robust to illumination changes and occlusion due to seizing the intrinsic correlation properties of tracking object. Nevertheless, because of using pixel-wise statistics, it is also easily affected by image noise. In papers[74][75], histogram of target object is adopted by researcher. Color histogram based on HSV [74] is applied in object representation. In paper [75], a rg-histogram based on normalized RGB color mode is proposed to represent aim object. However, the methods will gradually drift from targets during the tracking procedure. To improve the robustness of color or intensity histogram based methods, many researchers suggest to embody other information into it. In paper [76], a color-spatial joint model is used to describe the color distribution in spatial space. In paper [77], tracked region is

divided into several patches. By the spatial layout information of these patches and their intensity histograms, the final tracking location can be determined. In paper [78], texture information is integrated into color histogram. In paper [79], gradient orientation histogram [80] and color histogram are utilized to determine the tracking position.

Local feature descriptor means to represent targets based on several local interesting points or their combination. In paper [81], a SIFT together with color invariants based descriptor is applied in object tracking. In paper [82], 3D SIFT based on bag of words descriptor is proposed. In paper [83], an extended SURF visual features are proposed for object tracking. In paper [84], corner features are considered for the construction of appearance model. Recently, learning based methods are often proposed in several literatures. Different sort of visual features are packed to train a set of weaker learners. These weaker learning classifiers are then sent to construct a composed classifier which is applied in object tracking. In paper [85], Harr-like features [86] are used to construct weak classifiers. Generally, local feature descriptor is robust to shape deformation and rotation. However, feature point detection is often disturbed by image noise and background changes. For feature set based weak learners, the computational cost is expensive.

Statistical descriptor use different kinds of statistical models to fit tracked objects. These models could be Gaussian model, kernel model and template in subspace. In paper [87], gaussian model is introduced into object tracking field. The authors use a set of gaussian models to define a density function to approximate target appearance. Although the gaussian model based method can get reliable results, it is hard to determine the number of gaussian function. In paper [88], a kernel-based model is introduced to represent target object. The authors use spatially isotropic kernel to regularize target color histogram and mean shift based on Bhattacharyya metric to locate target position. In paper [89], edge information is integrated into kernel function. Other researchers focus on subspace for object tracking. In paper [90], target object is considered as linear combination of several basis templates in subspace. In paper [91], the authors propose a tracking method based on sparse representation.

Online self-learning descriptor is based on online self learning classifier. It firstly trains the classifier by several previous frame samples, after which it estimates the target position in image. Using the new estimated sample in current frame, the classifier is updated. The method in paper [92] is based on this technique. However, this method often suffers from drift problem due to accumulation errors in the tracking procedure. In paper [93], the authors use multi instances to address this problem. The conventional self-learning usually only uses positive samples. Multi-instances based self-learning proposed by the authors also adopt negative samples around the positive ones. This method can help classifier discriminating non-targets around a target.

All above mentioned methods are based on a relatively high quality image. However, in our case, motion blur of tracked objects may emerge in image. Only visual based method may not lead to a robust result. To cope with this problem, a LRF space and image based method is proposed in our work.

5.3 Overview of the Proposed Algorithm

5.4 Framework overview

The framework of the proposed tracking algorithm is shown in Fig.5-1. It takes previous tracked target's LRF measurements and previous image with annotated targets by tracking boxes as starting point. For the first frame, target is annotated with a tracking box manually, and its LRF measurements are picked out by the same way. In the initialization step, the image content in the tracking box is regularized into standard size and the tracked target's LRF measurements are used to calculate key information which are sent to frame information buffer. After that, in object representation step, the standard size image content is encoded by a method based on weighted RGB values. By this sort of code, in the current undistorted fisheye image, a list of patches is launched to determine an interesting small region that partially contains the target object. And then, a growth strategy based on current

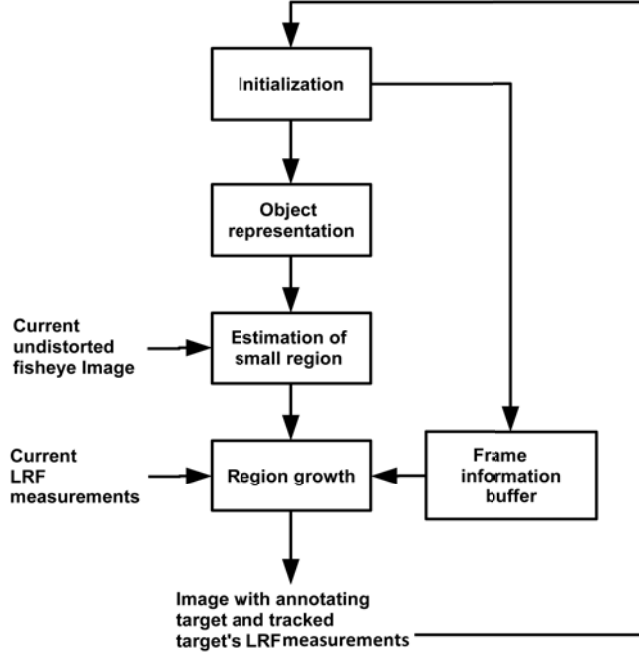


Figure 5-1: Processing flow chart for object tracking

LRF measurements together with frame buffer information are applied to help the small region to grow to cover the most part of the target. The area spanned by the final region and the LRF measurements in this region are considered as the final outputs.

5.5 Initialization

The input for the initialization step is the previous image with annotated tracked target by a tracking box and the tracked object's LRF measurements. The target in the tracking box in the previous image is regularized into 24x24 size (Fig.5-2). The tracked object's LRF measurements are used to obtain object information. Explicitly, object information contains: the positions of the projected points of the tracked object's LRF measurements in the image, the distances of the tracked object's LRF, the angles of the tracked object's LRF. The distances and angles of the tracked object's LRF are used to calculate the average distance, average angle, the maximum distance residual and the maximum degree residual. For statement convenience, we

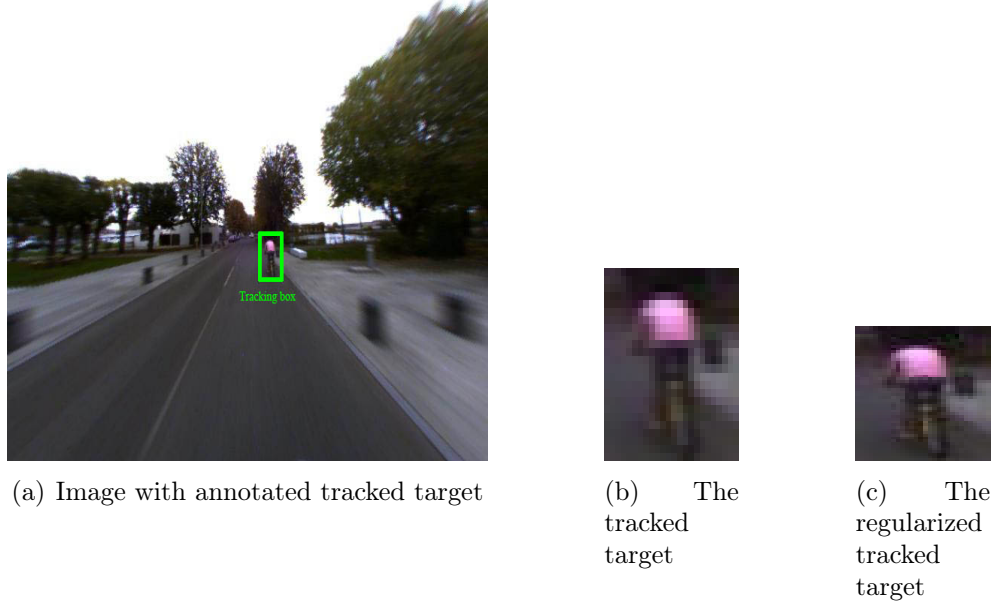


Figure 5-2: Regularize procedure

use A_{di} , A_{an} , M_{di} and M_{an} to denote the average distance, average angle, maximum distance residual and maximum angle residual respectively. The positions of the projected points of LRF measurements in the image are combined with the position of the tracking box in the image to compute low part proportion (R_{lpp}). Actually, R_{lpp} is a ratio of the distance between the average row of the projected points of the tracked object's LRF in the image and the row of the bottom of the tracking box in the image to the height of the tracking box. Let $P_r^i(P_{rx}^i, P_{ry}^i)$ denote the i -th projected point of the tracked object's LRF measurements in image, $P_{br}(P_{brx}, P_{bry})$ the lower right point of the tracking box in image, $P_{bl}(P_{blx}, P_{bly})$ the upper left point of the tracking box in image. R_{lpp} is defined as:

$$R_{lpp} = \frac{P_{bry} - \frac{\sum_{i=1}^N P_{ry}^i}{N}}{P_{bry} - P_{bly}} \quad (5.1)$$

where N is the amount of the tracked object's LRF measurements. The obtained A_{di} , A_{an} , M_{di} , M_{an} and R_{lpp} are packed as key information to send to frame information buffer.

5.6 Frame Information Buffer

The frame information buffer stores key information of three previous frames. These key information in the buffer are used to calculate three types of decision factors: distance, angle and low part proportion. The distance decision factor is a range decided by the average distance and average maximum distance residual of the three previous frames. Let D_{di} denote the distance decision range factor, F_{adi} and F_{mdi} the average distance and average maximum distance residual of the three pervious frames, k the current frame, then we have:

$$D_{di} = [F_{adi} - F_{mdi}, F_{adi} + F_{mdi}] \quad (5.2)$$

$$\begin{cases} F_{adi} = \frac{A_{di}^{k-3} + A_{di}^{k-2} + A_{di}^{k-1}}{3} \\ F_{mdi} = \frac{M_{di}^{k-3} + M_{di}^{k-2} + M_{di}^{k-1}}{3} \end{cases}$$

where A_{di}^k and M_{di}^k are the average distance and maximum distance residual in the frame k . The angle decision factor (D_{an}) is also a range and can be calculated by the same way. The LPP decision factor, which is also range, is determined by the following expression:

$$D_{lpp} = [0.95 * F_{lpp}, 1.05 * F_{lpp}] \quad (5.3)$$

$$with \quad F_{lpp} = \frac{R_{lpp}^{k-3} + R_{lpp}^{k-2} + R_{lpp}^{k-1}}{3}$$

where R_{lpp}^k is the LPP in the frame k . The above decision factors are used in region growth in the following step.

5.7 Object Representation

Taking into account motion blur case, we tend to adopt raw pixel information based descriptor to encode tracked objects. The frequently used method is to encode object using color or intensity value directly. As edge visual features of objects are not very clear in motion blur case, we mainly focus to encode their central field. The proposed

object descriptor is divided into two directions: vertical and horizontal directions. As the encoding way in both directions is the same, in the following we will only introduce how to encode the object in the vertical direction. The regularized tracked target patch obtained from initialization is encoded by its weight color. Let denote m and n the standard size patch height and width respectively, D_{ki} the k -th row code in i (R,G,B) channel. The way to model the regularized tracked target patch is as follows:

$$D_{ki} = P_k C_{ki} = P_k \sum_j^n I_{ijk} \quad (k = 1, 2, \dots, m) \quad (5.4)$$

where I_{ij} ($j = 1, 2, \dots, n$) is the j -th column intensity in i (R,G,B) channel, C_{ki} denotes the sum of the k -th row's intensity in i (R,G,B) channel and P_k is the probability density of the k -th row. In practice, taking the computation cost and resolution of tracked object, m and n are set 24. To promote the central field of image, 1-D Gaussian model is used to estimate P_k . The above procedure is shown in Fig.5-3

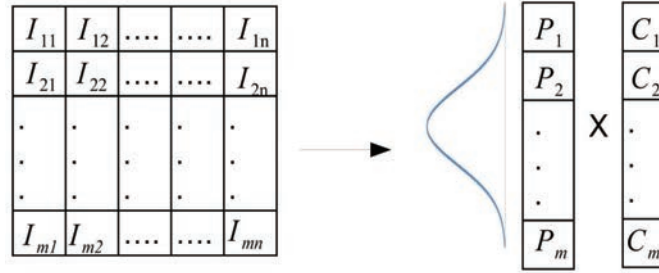


Figure 5-3: Encoding procedure in one color channel

5.8 Estimation of Small Region

Interesting small region (ISR) is a field that partially covers a target object. To determine ISR, we firstly define a searching area as a rectangular field, which has the same center of the tracking box in the previous frame, its width and height are three times larger than the tracking box's width and height respectively. An example of

searching area is shown in Fig.5-4. In the searching area, a group of scanning grids

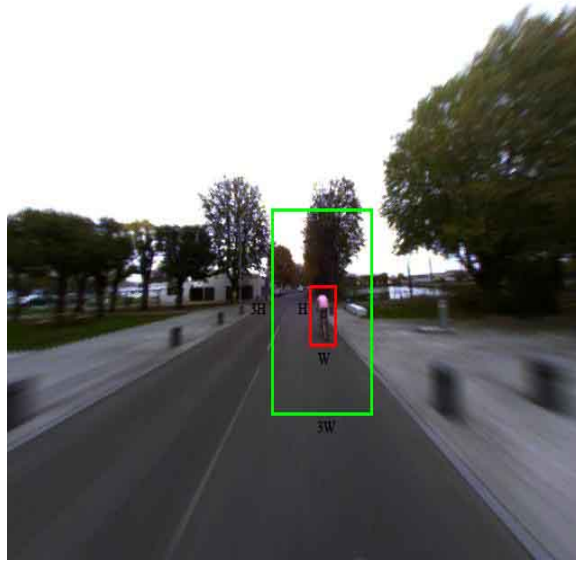


Figure 5-4: Example of searching area: the red rectangle is the tracking box, the green rectangle is the searching area

[93] is set up, as shown in Fig.5-5. The content in each scanning grid is an image patch. The patches in these scanning grids are used to constitute a list. In the list, all the patches are sorted according to a cost function. In our case, for each patch, this function is the sum of squared difference (SSD) of the code of the patch and the code of the content in the tracking box in the previous frame. The method to encode these patches is the approach described in object representation step. The top seven patches in the list are picked out and their common field corresponds to the required ISR.

5.9 Growth Strategy

ISR is the field merely containing a small part of the tracked object. To get a relative complete field of the tracked object, this region is expanded. Generally, there exist three region growth strategies: single direction, bidirectional and mixture way. For single direction and mixture way strategies, the key point is to know where is the brink of the object. However, in practice, it is hard to find out this information. For

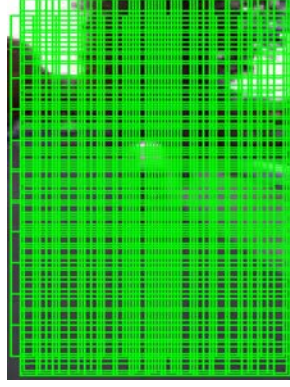


Figure 5-5: Group of scanning grids

bidirectional strategy, we just need to catch the partial location information of the tracked object in the image and know where it should stop. It is more convenient and easy to find out these two information than to determine the brink. In the proposed approach, the bidirectional growth strategy is adopted. This strategy is performed in the horizontal and vertical directions. We will firstly introduce the horizontal direction based growth, and then the vertical direction based one.

As described above, to perform bidirectional strategy we need to know the partial location information of the tracked object in the image and where it should stop. The first information is derived from the previous section (estimation of small region). The second information can be obtained in LRF space, which is defined as a 2D reprojection space of LRF measurements and it completely overlaps with the image. Each element in the space is characterized by three attributes: distance, position and angle. Fig.5-6 shows an example of LRF space. The growth of ISR in the horizontal direction is implemented in LRF space. Let's define the region corresponding to ISR in LRF space as LISR. It means that the growth of LISR is equivalent to the growth of ISR. Taking the middle point in LISR as the center, the horizontal brink of LISR is firstly expanded until the point (in LRF space), which is not within the range D_{di} . Then, a shrink operation is applied to the horizontal brink if the angle of the points in the expanded LISR is beyond the range D_{an} . Through the expansion and shrink procedures, the target object LRF measurements and its horizontal brink can

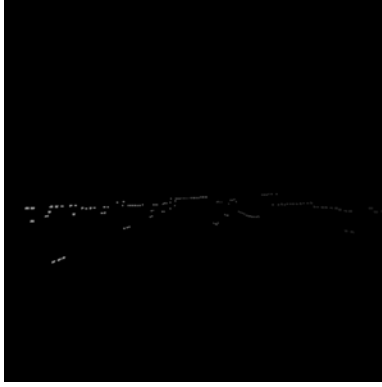


Figure 5-6: LRF space. The intensity denotes the degree of corresponding measurement.

be determined.

Because of the lack of the upper and lower limits of the tracked object in LRF space, it is hard to use the same way to expand the LISR in vertical direction. To address this problem, an alternative way is proposed. We firstly expand the LISR in vertical direction to a predefined height, which is calculated by multiplying the width of LISR and the ratio of the tracking box's height to its width in the first frame. However, this may lead to drifting gradually from the target during the tracking procedure (Fig.5-7). To tackle this issue, the position of spanned region has to be

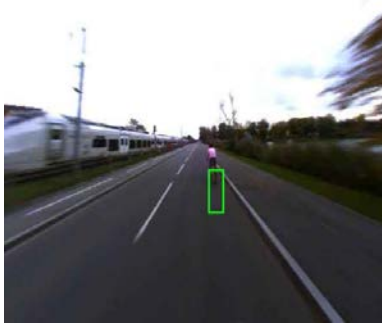


Figure 5-7: Tracking box drifting from the target

translated properly in vertical direction. In practice, we can note that the scanning position from LRF gradually moves from top to bottom when an object gets close to the LRF because there exists an upwards pitch angle between LRF and ground plane.

According to this principle, if R_{lpp} of that region is higher than the upper limit of the range D_{lpp} and the average distance in that region is less than F_{adi} , that region is translated upwards until its LPP fall into D_{lpp} . In other words, when the vehicle is approaching to tracked objects, their R_{lpp} should get small. Conversely, if R_{lpp} of that region is smaller than the lower limit of the range D_{lpp} and the average distance in that region is more than F_{adi} , that region is translated downwards until its LPP fall into D_{lpp} . The above conditions can be summarized as follows:

$$\begin{cases} D_{as} < F_{adi} : \downarrow (R_{lpp} > 1.05F_{lpp}) \\ D_{as} < F_{adi} : \uparrow (R_{lpp} > 0.95F_{lpp}) \end{cases} \quad (5.5)$$

where D_{as} denotes average distance of spanned region.

5.10 Experiment

5.10.1 Test results in real scenarios

In the first experiment, the proposed algorithm is tested in two long term videos captured by our experimental car moving in outdoor scene. There are totally 1000 frames in each video and several different cases (normal, presence of motion blur, background change, scale and appearance changes, over exposure and shadow). A man riding a bike is our target. The proposed method is firstly tested on the first video.

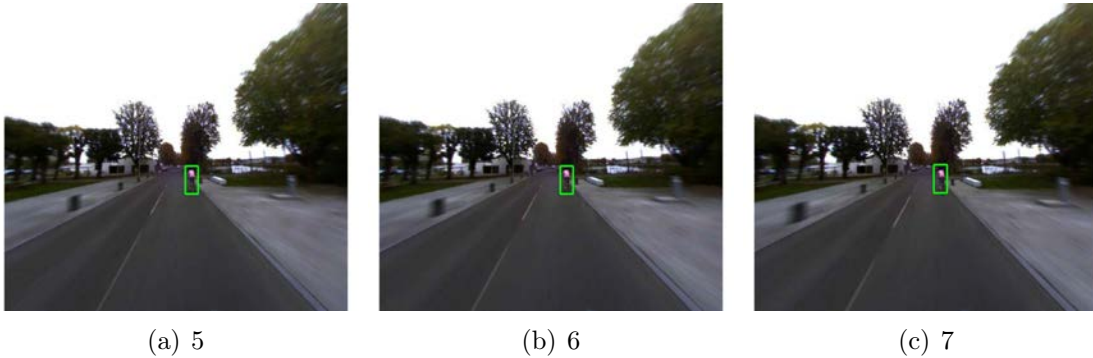


Figure 5-8: Test results in normal case

Fig.5-8 shows several test results in normal case. The number denote the index of frame. The proposed method can track the object accurately. Fig.5-9 shows

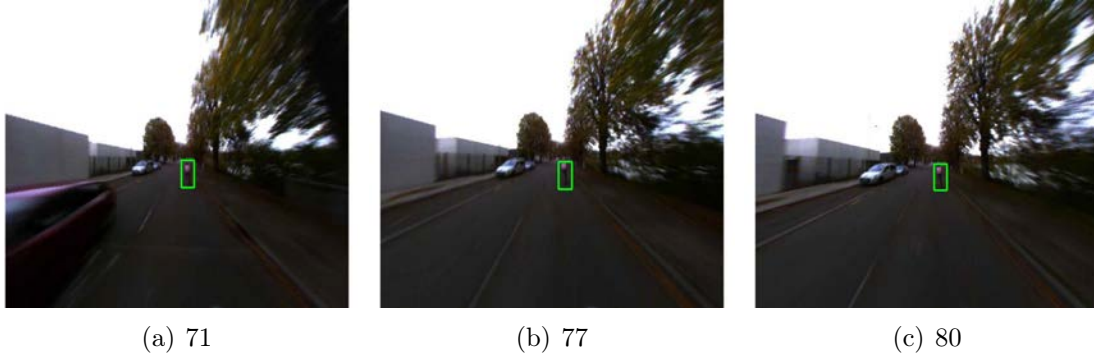


Figure 5-9: Test results in motion blur case

several test results in motion blur case. Compared with the normal case, we can note that the presence of motion blur in the image. This phenomenon is caused by the shaking of the car due to non flat ground. From frame 77 and 78, we can see that the vibration makes the tracking box a bit shift from target. Fig.5-10 shows test

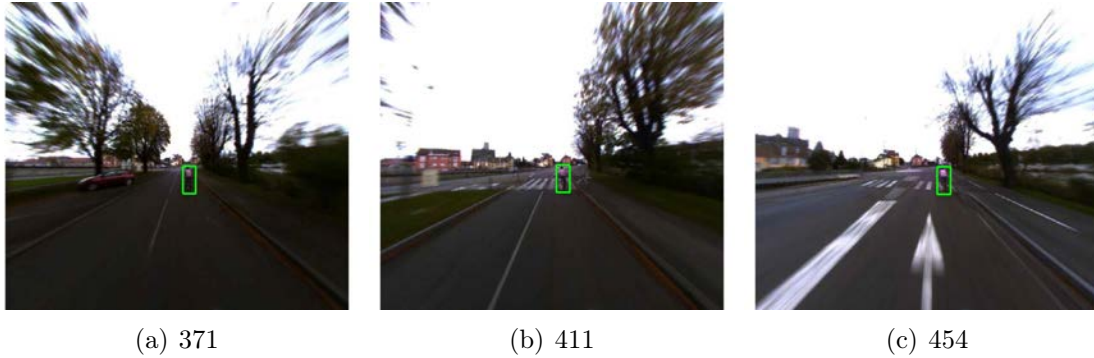


Figure 5-10: Test results in background change case

results in background change case. In frame 371, the background around the target is the road. In frame 411, the pedestrian crossing the road occupies the most part of the scene background. From the results, we can see that the background change don't affect the robustness of the proposed method. Fig.5-11 shows test results in appearance and scale changes case. In this test, we track two objects simultaneously. If we focus on the vehicle, in frame 245, only its left side appears in the image and

its size is small. In frame 259, the front part of the vehicle occupies the most of the appearance and its size is larger than the one in frame 245. In frame 270, the size of the front part and left side part are almost the same and the scale of the object in this frame changed significantly when compared with the one in frame 245. Meanwhile, we note the presence of motion blur in these frame.

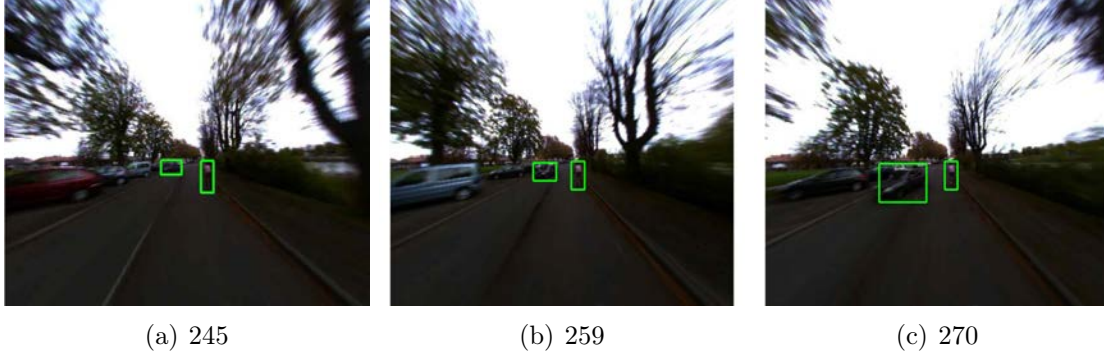


Figure 5-11: Test results in the appearance and scale changes case.

The following experiment results are obtained from the processing of the second video. Fig.5-12 show the test results in a normal case. Similar with the results in video 1, the proposed method can work normally. Fig.5-13 shows test results in illuminance change case. In frame 94, the target object is in shadow. In frame 117, the target is exposed to the light. From these frame, we can see that the intensity of tracked object varies greatly. Fig.5-14 shows test results in motion blur, appearance change and scale change case. From the 606 and 706 frames, it is noticeable that great appearance and scale change has taken place. From frame 706 and 712 frame, appearance change and motion blur occur simultaneously. This is a great challenge to some object tracking methods. In this challenge case, the proposed method also can work normally. Fig.5-15 shows test results in over exposure case. From frame 835 and 845, most part of the back of tracked object lose its original color and turn into white. In this case, the performance of some vision based tracking approaches is not very well duo to the loss of intensity.

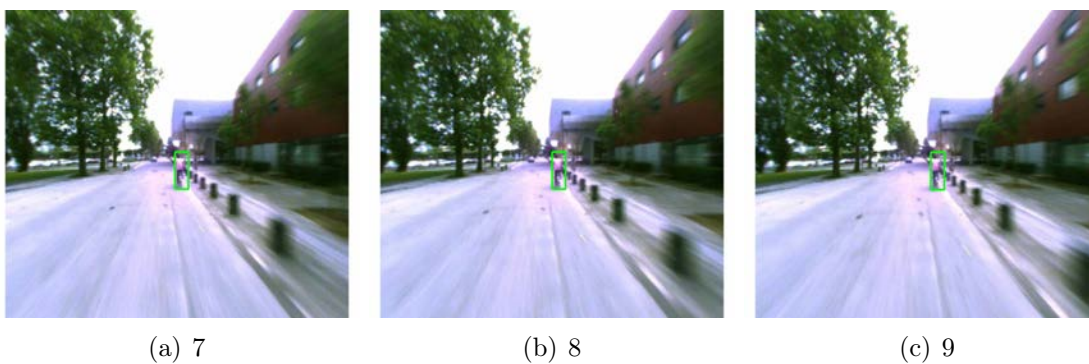


Figure 5-12: Test results in normal case

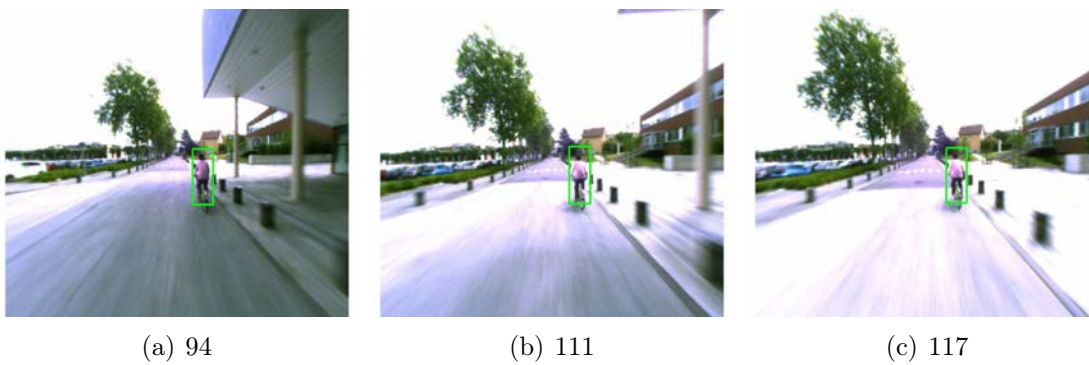


Figure 5-13: Test results from shadow scene to light scene

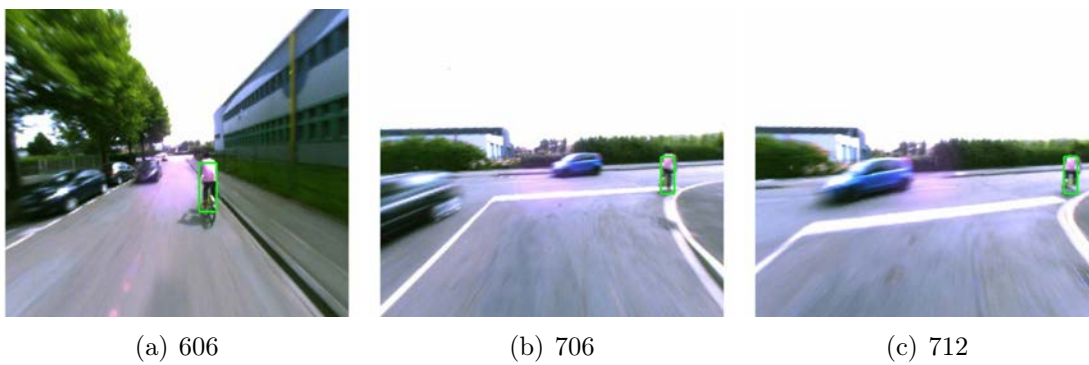


Figure 5-14: Test results in motion blur, appearance change and scale change case

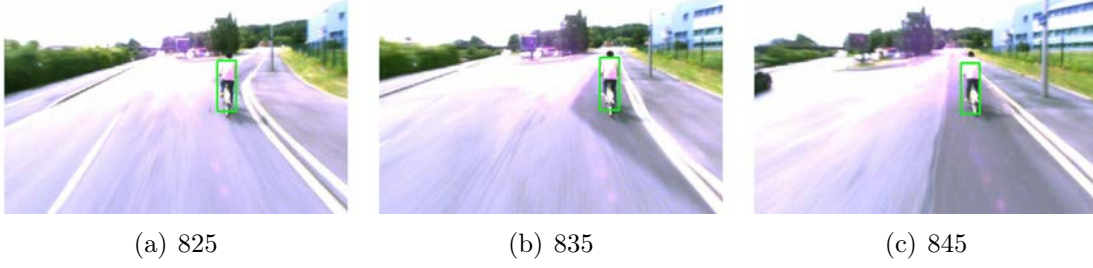


Figure 5-15: Test results in over exposure case

5.10.2 Comparison results

In the second experiment, the proposed method is compared with the method presented in paper [93]. To get ground truth, we manually annotate the tracked objects with a minimum bounding box. The criteria indicator is the overlapping rate (OR). Let S_b denote the area of minimum bounding box in the image, S_t the area of the tracking box in the image. OR is defined as follows:

$$OR = \frac{S_b \cap S_t}{Max(S_b, S_t)}. \quad (5.6)$$

The comparison results are shown in Fig.5-16 and Fig.5-17. The red line represents the results obtained from the proposed approach. The blue one is the results derived from the approach presented in paper [93]. In Fig.5-16, the method proposed in paper [93] missed the target from frame 100 to frame 600. After frame 600, it gets back the target. The reason is that the method in [93] has a big buffer which stores the descriptors of the tracked object in the past frames. This buffer can help the method to get back the target when it is lost. However, in certain cases, it also does harm to track the object without other judgment conditions because it may reserve the inaccurate object descriptor. In Fig.5-17, from frame 1 to frame 600, the proposed method performs a little better than the approach in [93] overall. From frame 630 to frame 670, the outcomes of the proposed approach get worse than the method proposed in paper [93]. The reason is that the performance is affected by

LRF measurements. In deed, if the LRF measurements get worse, the outcome of our approach will go bad. After frame 700, our method outperforms the approach proposed in paper [93]. The reason that the performance of the approach proposed in paper [93] decrease rapidly is that the variation of the appearance of target object and motion blur occur simultaneously from frame 705 to frame 715.

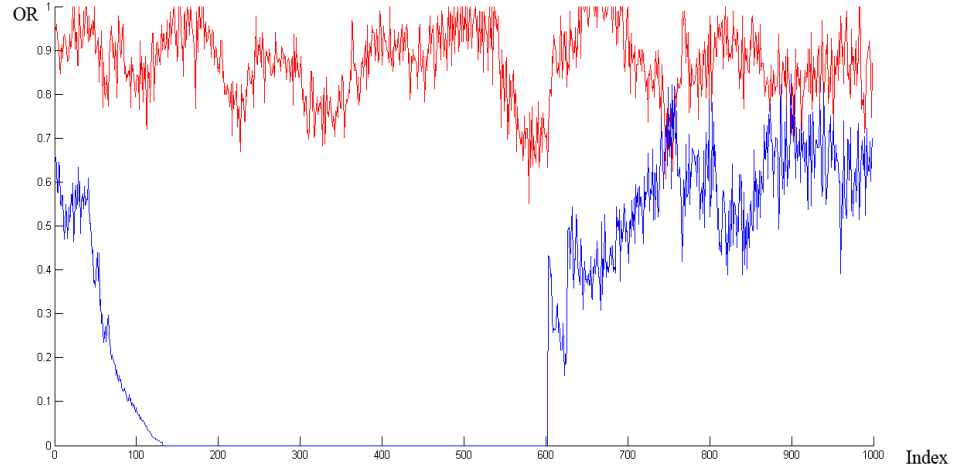


Figure 5-16: Comparison results between the proposed method and the approach presented in paper[93] for first video

Fig.5-18 and Fig.5-19 show some examples of visual comparison results. Above is the examples from the method in paper [93]. Below is the results of the proposed method. From Fig.5-18, after vibration of vehicle, the method proposed in paper [93] gradually lost the target. The same things also happens in Fig.5-19. The presence of motion blur does very harm to the approach proposed in paper [93]. To an extent, the proposed approach can handle this issue.

5.11 Conclusion and Future Works

This chapter presented a method for tracking objects. It establishes a list of samples in a searching area to determine the interesting small region. This region is mapped to LRF space. In this space, a bidirectional growth strategy is applied to the small

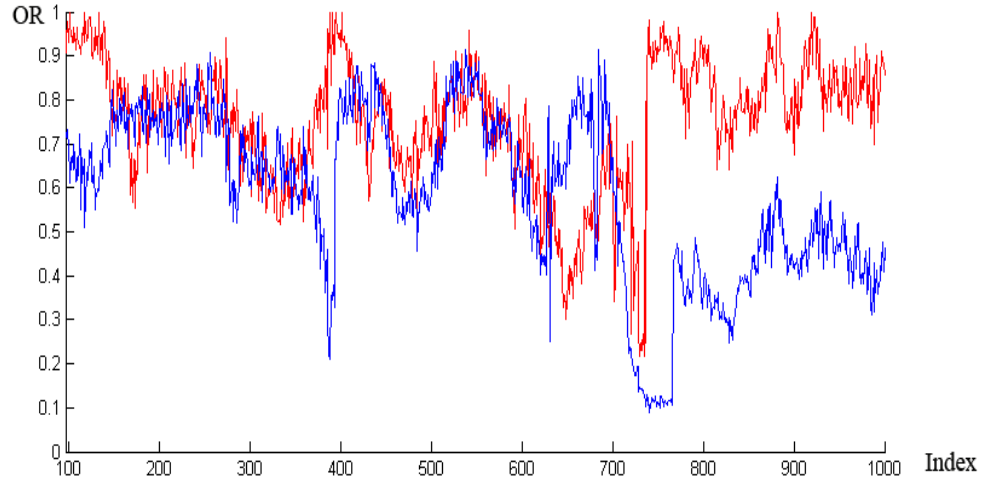


Figure 5-17: Comparison results between the proposed method and the approach presented in paper[93] in second video

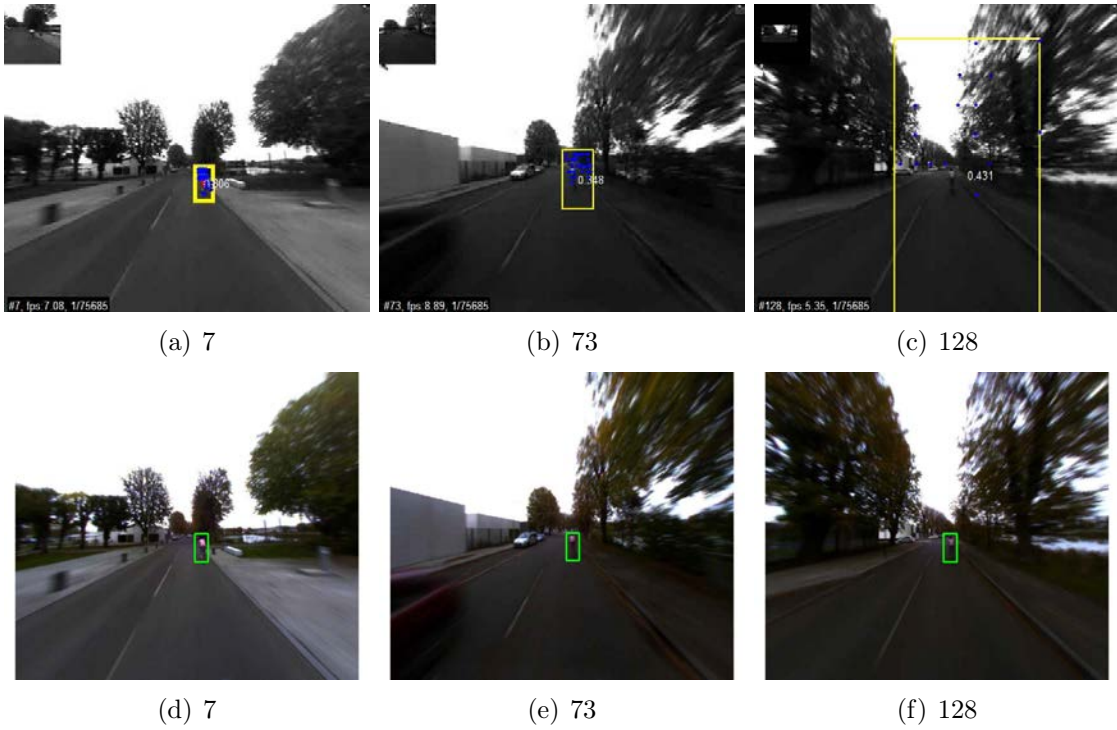


Figure 5-18: Examples of comparison experiment in first video

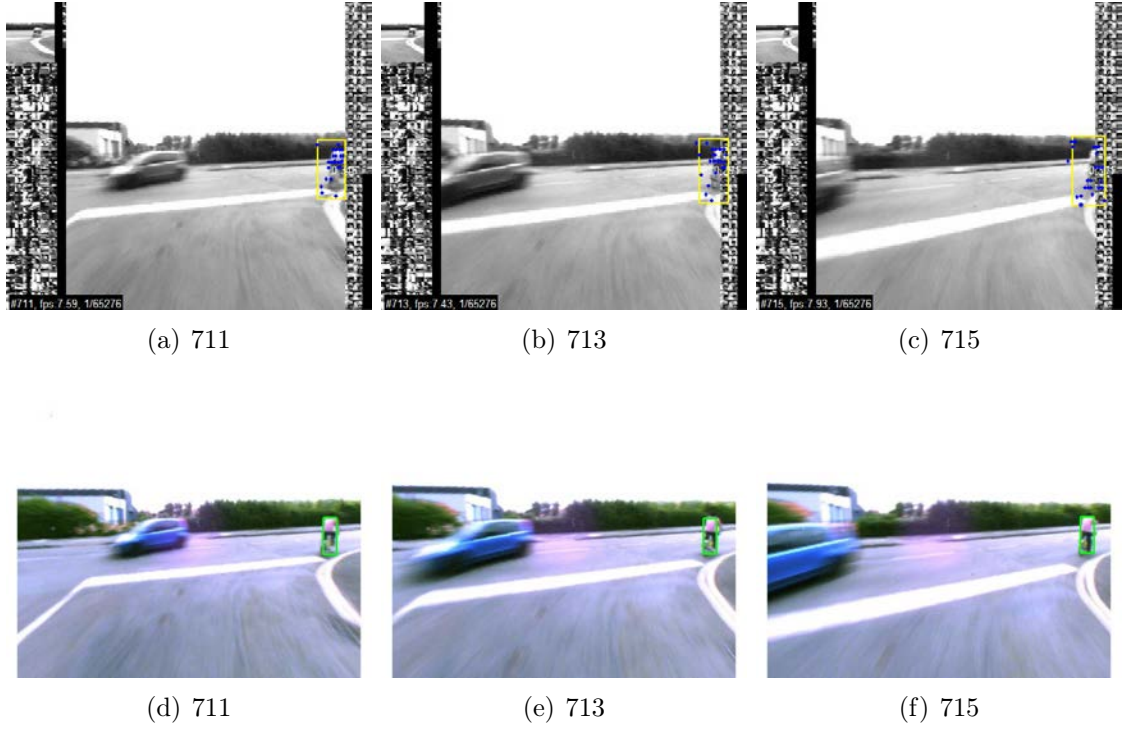


Figure 5-19: Examples of visual comparison results (second video)

region. The growth strategy is controlled by three decision factors which are calculated by previous frame information. A long-term real data test shows the efficiency of the proposed method even in presence of motion blur, background and scale changes. However, the occlusion problem is not tackled by the proposed method. Considering the low quality of the images, it is a challenge to find a proper descriptor to deal with the problem. This work can be treated as a direction of future research. Besides that, the case of objects leaving and reappearings is not tackled in our method.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

The issues addressed in this thesis refer to several basic aspects of road scene understanding in perception systems of intelligent vehicle. A complete and precise description of road scene state will pave a smooth way for intelligent vehicle applications. The following paragraphs provide a general conclusion for the methods proposed in the thesis.

In the first, a method regarding extrinsic calibration between fisheye camera and Laser Ranger Finder is presented in chapter2. The extrinsic calibration is based on the intrinsic parameters estimation of fisheye camera. Three indicators are proposed to compare three type models of fisheye camera introduced in literature. Based on the selected fisheye model, a chessboard pattern is applied to locate several points on the LRF scanning plane. The scanning plane is then estimated by these points. Based on the obtained scanning plane and several points on this plane, two geometrical constraints are built. Simulation and real experiments show the effectiveness of the proposed method. Finally, we applied it for motion trajectory estimation.

In the second, fisheye camera and LRF based road detection method is introduced in chapter3. It firstly uses histogram based classification method to implements a preliminary road detection in illuminance invariant image. The derived result is then checked by a coherence principal based on the LRF measurements and the amount of

road pixels. The unqualified road detection result is refined by a distance image based classification method in HSI space. The real experiment results show the effectiveness of the proposed approach in reducing the "over saturated" or "under saturated" errors caused by the method merely based on illuminance invariant image in cloudy day.

In the third, a LRF, GPS and fisheye camera based obstacle detection approach is proposed in chapter4. Firstly, a distortion removal operation is applied to both the fisheye image and the road image (chapter3). Then, the perspective effect is removed from the undistorted fisheye image and the undistorted road image by inverse perspective mapping algorithm. After that, the center lane marker in undistorted fisheye image is detected. Meanwhile, a road model is built by the GIS information obtained by GPS data and Openstreetmap. This model is mapped into the undistorted fisheye image by aligning the road skeleton to the center lane marker, in order to determine the complete road boundary information. The potential obstacle regions can be found out by combining the road boundary information and road image without perspective effect. Finally, LRF measurements are used to pick out the real obstacle regions from these candidate regions. The real experiments show the effectiveness of the proposed method in coping with obstacle detection in presence of motion blur.

In the fourth, we present a method for object tracking in chapter 5. It encodes target objects by weighted colors. Based on the derived code, a list of samples in searching area is sorted to determine the interesting small region. This small region is expanded in LRF space according to three decision factors obtained from the position of tracking box and the LRF measurements of target objects in previous images. The long term real experiment results show the effectiveness of the proposed tracking algorithm.

6.2 Future Works

In the author's point of view, some recommendations are given to improve the work.

For extrinsic calibration between camera and LRF, different sensors could be attempted such as 3D LRF and stereo vision. Meanwhile, extrinsic calibration between

the combination of several kinds of cameras (fisheye, stereo, omnidirectional) and LRF is also an interesting work. The property of straight line in fisheye image is another important research direction.

For road detection, a robust color descriptor for road deserves researching. Besides that, GIS features in map could be an available information to improve the results. Combination of several cameras is also a considerable option. Segmentation technique based on condition random field (CRF) also deserves attempting.

For obstacle detection, optical flow with fisheye camera should be interesting to improve detection results. To build more kinds of road model, it can make the proposed method more generalized.

For object tracking, the mixture of gaussian models can be used to represent target objects to deal with the occlusion problem. Multi-instances based self-learning techniques could be attempted for shortly object disappearance.

List of Figures

1-1	Example of participants in 2003 Grand Challenge	6
1-2	Google new prototype driverless car ¹	6
2-1	The experimental platform. The fisheye camera is on the top bracket and the LRF is fixed in front of the vehicle	12
2-2	Fisheye model	15
2-3	Affine transformation between the sensor plane and the image plane .	16
2-4	Fisheye camera projection model in paper [11]. The image of the point P is p whereas it would be p' by a pinhole camera	17
2-5	Unified projection model in paper [12]	20
2-6	Extrinsic calibration between the two sensors	23
2-7	Framework of the proposed extrinsic calibration approach	24
2-8	Corresponding points determination methodology	25
2-9	First constraint. The laser beams plane is colored by blue. In camera coordinate system, any vector on this plane is perpendicular to the vector N_v	29
2-10	Performance w.r.t the number of known points in ideal case	31
2-11	Performance w.r.t the number of known points in noisy case	32
2-12	Comparison of calibration results under different noise levels	33
2-13	Two patterns used in convergence rate test	34
2-14	Point reprojection using the calibration results of the approaches pro- posed in [12] and [10]	35

2-15	Calibration results for real data considering 3 known points(a) or 8 known points (b)	37
2-16	Projection of LRF measurements in the fisheye image	40
2-17	Estimation of motion trajectory under two different case	41
3-1	The left image is original fisheye image, the right image is the road image which we aim to obtain.	46
3-2	Road detection framework diagram	47
3-3	The context of the captured traffic scene is extracted from the original fisheye image.	48
3-4	Left image is a color checker, right graph represents the color in color checker under different illumination maps to log-chromaticity space. A set of different color surfaces under different illuminations form several parallel lines. l_{R_θ} is a line perpendicular to these parallel lines. The projection of log-chromaticity of pixels into l_{R_θ} form a 1-d illumination invariant image.	50
3-5	Average entropy minimization for different values of the angle R_θ . . .	51
3-6	The context of the captured traffic scene in RGB space is converted to illumination invariant grayscale image	53
3-7	The illumination invariant grayscale image is classified as road and not road. For convenience, the front part of the vehicle is covered by a dark rectangle.	53
3-8	Scattered pieces of road pixels are connected to form the so called road binary image.	54
3-9	Framework of refined procedure	56
3-10	ROI extraction	57
3-11	The content of the image in H,S,I channels.	58
3-12	The histograms for S,I and SI plane shown in [26]	59
3-13	Process of removing the false road parts in distance image	61
3-14	Image is divided into equal intervals	62

3-15	The configuration of the used experimental platform	62
3-16	Experimental results. (Top row) Original image; (Middle row) Detection result; (Last Row) Ground truth.	65
3-17	Experimental results. The images of the first row are original images, the second row are results obtained using the approach proposed in paper [25], the third are the results obtained by the proposed method in this paper	66
4-1	The challenging case: motion blur effect for the red vehicle in the image	70
4-2	Processing flow chart for obstacle detection	71
4-3	IPM coordinate system	72
4-4	IPM example: In IPM image, the borders of road appears parallel. . .	73
4-5	Lane marker detection flow chart	74
4-6	Filter performance	75
4-7	The red area in the image is the predefined fix area. The yellow line expresses the center of this area	76
4-8	WGS84 coordinate system	77
4-9	ECEF coordinate system	78
4-10	Local surface plane	78
4-11	OpenstreetMap. (a) The map under editor view. The red circle points are nodes. (b) The map in normal view	80
4-12	Road skeleton extraction	80
4-13	Road model	81
4-14	Road model mapping	82
4-15	(a) Red lines in the image I_r shows the boundary of the road. Non-road pixels within the road boundary are conserved (b) Corresponding polar histogram	83
4-16	The searching windows in normal view image	83

4-17	(a) Original image (b) Results obtained with the approach of paper [66] (c) Results obtained with the approach of paper [67]. (d) Results obtained with our approach.	86
4-18	(a) Original image (b)Results obtained with the approach of paper [66] (c)Results obtained with our approach.	88
4-19	(a) error case (b)correct case	89
5-1	Processing flow chart for object tracking	97
5-2	Regularize procedure	98
5-3	Encoding procedure in one color channel	100
5-4	Example of searching area: the red rectangle is the tracking box, the green rectangle is the searching area	101
5-5	Group of scanning grids	102
5-6	LRF space. The intensity denotes the degree of corresponding measurement.	103
5-7	Tracking box drifting from the target	103
5-8	Test results in normal case	104
5-9	Test results in motion blur case	105
5-10	Test results in background change case	105
5-11	Test results in the appearance and scale changes case.	106
5-12	Test results in normal case	107
5-13	Test results from shadow scene to light scene	107
5-14	Test results in motion blur, appearance change and scale change case	107
5-15	Test results in over exposure case	108
5-16	Comparison results between the proposed method and the approach presented in paper[93] for first video	109
5-17	Comparison results between the proposed method and the approach presented in paper[93] in second video	110
5-18	Examples of comparison experiment in first video	110
5-19	Examples of visual comparison results (second video)	111

List of Tables

2.1	The configuration of the 8 poses used for applying the approach proposed in paper [7]	33
2.2	The convergence time of three methods in solution optimization procedure	34
2.3	Average reprojection errors	36
2.4	Calibration results obtained by the two approaches	38
3.1	Comparison of the proposed method performance in RGB nd HSI color space	63
3.2	Performance of road detection considering the proposed approach . .	64
3.3	The performance of road detection considering the method proposed in paper [25]	65
4.1	The performance of the proposed approach with and without LRF measurements. $Hit Rate = \frac{The\ number\ of\ the\ correct\ detected\ obstacles}{The\ total\ number\ of\ the\ ground\ truth\ obstacles}$ $Correct Rate = \frac{The\ number\ of\ the\ correct\ detected\ obstacles}{The\ total\ number\ of\ the\ detected\ obstacles}$	87

Appendix A

Transformation from WGS84 to Extended Lambert II

Given $(\alpha_{g0}, \beta_{g0})$ the latitude and longitude from GPS receiver, the coordinate (x^l, y^l) in extended Lamber II system can be calculated by the following steps:

1) Transform $(\alpha_{g0}, \beta_{g0})$ from decimal degree format to degree/radians format (α_g, β_g) .

2) Convert (α_g, β_g) to cartesian coordinate (x_w, y_w, z_w)

Given:

$$\left\{ \begin{array}{l} a_0 = 6378137 \\ b_0 = 6356752.314 \\ e_0 = \frac{a_0^2 - b_0^2}{a_0^2} \\ N = \frac{a_0}{\sqrt{1 - e_0 \sin^2(\alpha_g)}} \end{array} \right. \quad (\text{A.1})$$

we have the following equations:

$$\left\{ \begin{array}{l} x_w = N \cos \alpha_g \cos \beta_g \\ y_w = N \cos \alpha_g \sin \beta_g \\ z_w = N(1 - e_0) \sin \alpha_g \end{array} \right. \quad (\text{A.2})$$

3) Translate (x_w, y_w, z_w) to Cartesian coordinate NTF (Nouvelle Triangulation de la

$$\text{France})(x_n, y_n, z_n) \quad \begin{cases} x_n = x_w + 168 \\ y_n = y_w + 60 \\ z_n = z_w - 320 \end{cases} \quad (\text{A.3})$$

4) Calculate geographic coordinate NTF (α_n, β_n) from (x_n, y_n, z_n) .

Given:

$$\begin{cases} e_n = \frac{a_n^2 - b_n^2}{a_n^2} \\ \lambda_0 = \text{atan}(d_n z_n (1 - \frac{a_n e_n}{\sqrt{x_n^2 + y_n^2 + z_n^2}})) \end{cases} \quad (\text{A.4})$$

$$\lambda_1 = \text{atan}(\frac{d_n z_n}{1 - \frac{a_n e_n \cos(\lambda_0)}{\sqrt{(x_n^2 + y_n^2)(1 - e_n \sin^2(\lambda_0))}}}) \quad (\text{A.5})$$

where $a_n = 6378249.2$, $b_n = 6356515$ and $d_n = \frac{1}{\sqrt{x_n^2 + y_n^2}}$.

(α_n, β_n) can be derived by the following algorithm:

Algorithm for calculating (α_n, β_n)

While $|\lambda_1 - \lambda_0| > e^{-10}$, **do**

$\lambda_0 = \lambda_1$;

Calculate λ_1 with Eq.A.5.

End

Then, $\alpha_n = \lambda_1$ and $\beta_n = \text{atan}(\frac{y_n}{x_n})$

5) Calculate (x^l, y^l) through (α_n, β_n)

Given:

$$\left\{ \begin{array}{l} s = 0.7289686274 \\ n = 11745793.39 \\ x_f = 600000 \\ y_f = 8199695.768 \\ \sigma_0 = 0.04079234433198 \\ L = \log(\tan(\frac{\pi}{4} + \frac{\alpha_n}{2}))(\frac{1 - \sqrt{e_n} \sin \alpha_n}{1 + \sqrt{e_n} \sin \alpha_n})^{\sqrt{e_n}/2} \end{array} \right. \quad (\text{A.6})$$

(x^l, y^l) will be derived by the following equations:

$$\left\{ \begin{array}{l} x^l = x_f + ne^{-sL} \sin(s(\beta_n - \sigma_0)) \\ y^l = y_f - ne^{-sL} \cos(s(\beta_n - \sigma_0)) \end{array} \right. \quad (\text{A.7})$$

Bibliography

- [1] A. Broggi, A. Zelinsky, M. Parent, and C. E. Thorpe, “Intelligent vehicles,” in *Springer Handbook of Robotics*. Springer, 2008, pp. 1175–1198.
- [2] A. Eskandarian, *Handbook of intelligent vehicles*. Springer, 2012.
- [3] H. Aliakbarpour, P. Nuez, J. Prado, K. Khoshhal, and J. Dias, “An efficient algorithm for extrinsic calibration between a 3D laser range finder and a stereo camera for surveillance,” *International Conference on Advanced Robotics*, pp. 1–6, Jun. 2009.
- [4] D. Scaramuzza, A. Harati, and R. Siegwart, “Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes,” *International Conference on Intelligent Robots and Systems*, pp. 4164 –4169, Nov. 2007.
- [5] V. Niola, C. Rossi, S. Savino, and S. Strano, “A method for the calibration of a 3-d laser scanner,” *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 2, pp. 479–484, 2011.
- [6] Q. Zhang and R. Pless, “Extrinsic calibration of a camera and laser range finder (improves camera calibration),” *International Conference on Intelligent Robots and Systems*, pp. 2301 –2306 vol.3, Oct. 2004.
- [7] C. Mei and P. Rives, “Calibration between a central catadioptric camera and a laser range finder for robotic applications,” *International Conference on Robotics and Automation*, pp. 532 –537, 2006.

- [8] X. Brim and F. Goulette, “Modeling and calibration of coupled fish-eye CCD camera and laser range scanner for outdoor environment reconstruction,” *Sixth International Conference on 3-D Digital Imaging and Modeling*, pp. 320–327, Aug 2007.
- [9] O. Naroditsky, A. Patterson, and K. Daniilidis, “Automatic alignment of a camera with a line scan LIDAR system,” *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3429–3434, 2011.
- [10] D. Scaramuzza, A. Martinelli, and R. Siegwart, “A toolbox for easily calibrating omnidirectional cameras,” *International Conference on Intelligent Robots and Systems*, pp. 5695–5701, Oct. 2006.
- [11] J. Kannala and S. Brandt, “A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1335–1340 vol.8, Aug. 2006.
- [12] C. Mei and P. Rives, “Single view point omnidirectional camera calibration from planar grids,” *International Conference on Robotics and Automation*, pp. 3945–3950, Apr. 2007.
- [13] C. Geyer and K. Daniilidis, “A unifying theory for central panoramic systems and practical implications,” in *ECCV 2000*. Springer, 2000, pp. 445–461.
- [14] J. P. Barreto and H. Araujo, “Issues on the geometry of central catadioptric image formation,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 2. IEEE, 2001, pp. II–422.
- [15] A. Fitzgibbon, “Simultaneous linear estimation of multiple view geometry and lens distortion,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. I–125–I–132 vol.1.

- [16] X. Ying and Z. Hu, “Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model,” in *Computer Vision - ECCV 2004*, ser. Lecture Notes in Computer Science, T. Pajdla and J. Matas, Eds. Springer Berlin Heidelberg, 2004, vol. 3021, pp. 442–455.
- [17] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: <http://doi.acm.org/10.1145/358669.358692>
- [18] M. Zuliani, C. S. Kenney, and B. S. Manjunath, “The multiransac algorithm and its application to detect planar homographies,” in *IEEE International Conference on Image Processing*, Sep 2005.
- [19] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*. John Wiley & Sons, 2012, vol. 821.
- [20] A. Ranganathan, “The levenberg-marquardt algorithm,” *Tutorial on LM Algorithm*, pp. 1–5, 2004.
- [21] Y. Chen and G. Medioni, “Object modelling by registration of multiple range images,” *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.
- [22] P. J. Besl and N. D. McKay, “Method for registration of 3-d shapes,” in *Robotics-DL tentative*. International Society for Optics and Photonics, 1992, pp. 586–606.
- [23] S. Rusinkiewicz and M. Levoy, “Efficient variants of the icp algorithm,” in *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*. IEEE, 2001, pp. 145–152.
- [24] J. Joung, K. H. An, J. W. Kang, M.-J. Chung, and W. Yu, “3d environment reconstruction using modified color ICP algorithm by fusion of a camera and a 3d laser range finder,” in *International Conference on Intelligent Robots and Systems*, 2009, pp. 3082–3088.

- [25] J. Alvarez and A. López, “Road detection based on illuminant invariance,” *IEEE Transactions on Intelligent Transportation Systems.*, vol. 12, no. 1, 2011.
- [26] C. Rotaru, T. Graf, and J. Zhang, “Color image segmentation in HSI space for automotive applications,” *Journal of Real-Time Image Processing*, vol. 3, no. 4, pp. 311–322, 2008.
- [27] H.-C. Chen, W.-J. Chien, and S.-J. Wang, “Contrast-based color image segmentation,” *Signal Processing Letters*, vol. 11, no. 7, pp. 641–644, 2004.
- [28] N. Vandapel, D. Huber, A. Kapuria, and M. Hebert, “Natural terrain classification using 3-d ladar data,” in *IEEE International Conference on Robotics and Automation*, vol. 5, pp. 5117–5122, 2004.
- [29] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel, “On the segmentation of 3D LIDAR point clouds,” in *IEEE International Conference on Robotics and Automation*, pp. 2798–2805, 2011.
- [30] D. Gallup, J.-M. Frahm, and M. Pollefeys, “Piecewise planar and non-planar stereo for urban scene reconstruction,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1418–1425, 2010.
- [31] N. Soquet, D. Aubert, and N. Hautiere, “Road segmentation supervised by an extended v-disparity algorithm for autonomous navigation,” in *IEEE Intelligent Vehicles Symposium*, pp. 160–165, 2007.
- [32] P. Kohli, L. Ladick, and P. H. S. Torr, “Robust higher order potentials for enforcing label consistency,” *International Journal of Computer Vision*, vol. 82, no. 3, pp. 302–324, May 2009.
- [33] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. R. Bradski, “Self-supervised Monocular Road Detection in Desert Terrain,” in *Robotics: Science and Systems*, 2006.

- [34] H. Kong, J.-Y. Audibert, and J. Ponce, “Vanishing point detection for road detection,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 96–103, 2009.
- [35] M. Darms, M. Komar, and S. Lueke, “Map based road boundary estimation,” in *IEEE Intelligent Vehicles Symposium*, pp. 609–614, 2010.
- [36] M. Jones and J. Rehg, “Statistical color models with application to skin detection,” in *Computer Vision and Pattern Recognition*, vol. 1, pp. 280 Vol. 1, 1999.
- [37] B. Kim, J. Son, and K. Sohn, “Illumination invariant road detection based on learning method,” in *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 1009–1014.
- [38] G. Finlayson, S. Hordley, C. Lu, and M. Drew, “On the removal of shadows from images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 59–68, 2006.
- [39] B. Horn, *Robot Vision*. MIT Press, Jan. 1986.
- [40] G. Wyszecki, *Color science: concepts and methods, quantitative data and formulae*. New York; Chichester: Wiley, 2001.
- [41] G. Finlayson, M. Drew, and C. Lu, “Intrinsic images by entropy minimization,” in *Computer Vision - ECCV 2004*, ser. Lecture Notes in Computer Science, T. Pajdla and J. Matas, Eds. Springer Berlin Heidelberg, 2004, vol. 3023, pp. 582–595.
- [42] B. Amidan, T. Ferryman, and S. Cooley, “Data outlier detection using the chebyshev theorem,” in *2005 IEEE Aerospace Conference*, Mar. 2005, pp. 3814–3819.
- [43] D. W. Scott, “On optimal and data-based histograms,” *Biometrika*, vol. 66, no. 3, p. 605, Dec. 1979.

- [44] C. Tan, T. Hong, T. Chang, and M. Shneier, "Color model-based real-time learning for road following," in *IEEE Intelligent Transportation Systems Conference*, pp. 939–944, 2006.
- [45] C. Zhang and P. Wang, "A new method of color image segmentation based on intensity and hue clustering," in *15th International Conference on Pattern Recognition, 2000. Proceedings*, vol. 3, 2000, pp. 613–616 vol.3.
- [46] A. Zymnis, S. Boyd, and E. Candes, "Compressed sensing with quantized measurements," *IEEE Signal Processing Letters*, vol. 17, no. 2, pp. 149–152, 2010.
- [47] D. Guo, T. Fraichard, M. Xie, and C. Laugier, "Color modeling by spherical influence field in sensing driving environment," *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 249–254, 2000.
- [48] T. Bucher, C. Curio, J. Edelbrunner, C. Igel, D. Kastrup, I. Leefken, G. Lorenz, A. Steinhage, and W. Von Seelen, "Image processing and behavior planning for intelligent vehicles," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 1, pp. 62–75, 2003.
- [49] T. Kalinke, C. Tzomakas, and W. V. Seelen, "A texture-based object detection and an adaptive model-based classification," *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 341–346, 1998.
- [50] T. Zielke, M. Brauckmann, and W. Vonseelen, "Intensity and edge-based symmetry detection with an application to car-following," *CVGIP: Image Understanding*, vol. 58, no. 2, pp. 177–190, 1993.
- [51] W. Kruger, W. Enkelmann, and S. Rossle, "Real-time estimation and tracking of optical flow vectors for obstacle detection," *Proceedings of the IEEE the Intelligent Vehicles Symposium*, pp. 304–309, 1995.
- [52] A. Jazayeri, H. Cai, J. Y. Zheng, and M. Tuceryan, "Vehicle detection and tracking in car video based on motion model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 583–595, jun 2011.

- [53] J. Molineros, S. Cheng, Y. Owechko, D. Levi, and W. Zhang, "Monocular rear-view obstacle detection using residual flow," in *ECCV 2012. Workshops and Demonstrations*, 2012, vol. 7584, pp. 504–514.
- [54] 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.
- [55] R. Labayrade, D. Aubert, and J. P. Tarel, "Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation," *Proceedings of the IEEE Intelligent Vehicle Symposium*, vol. 2, pp. 646–651, jun 2002.
- [56] F. Oniga and S. Nedevschi, "Processing dense stereo data using elevation maps: Road surface, traffic isle, and obstacle detection," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 3, pp. 1172–1182, Mar. 2010.
- [57] K. Rebai, A. Benabderrahmane, O. Azouaoui, and N. Ouadah, "Moving obstacles detection and tracking with laser range finder," *International Conference on Advanced Robotics*, pp. 1–6, jun 2009.
- [58] T. Weiss, B. Schiele, and K. Dietmayer, "Robust driving path detection in urban and highway scenarios using a laser scanner and online occupancy grids," *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 184–189, jun 2007.
- [59] G. Monteiro, C. Premevida, P. Peixoto, and U. Nunes, "Tracking and classification of dynamic obstacles using laser range finder and vision," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006.
- [60] M. Bertozzi and A. Broggi, "Gold: a parallel real-time stereo vision system for generic obstacle and lane detection," *Image Processing, IEEE Transactions on*, vol. 7, no. 1, pp. 62–81, Jan 1998.

- [61] M. Aly, “Real time detection of lane markers in urban streets,” in *Intelligent Vehicles Symposium, 2008 IEEE*, June 2008, pp. 7–12.
- [62] J. Alvarez, F. Lumbreras, T. Gevers, and A. Lopez, “Geographic information for vision-based road detection,” *IEEE Intelligent Vehicles Symposium*, pp. 621–626, jun 2010.
- [63] J. Bennett, *OpenStreetMap*. Packt Publishing Ltd, 2010.
- [64] M. Haklay and P. Weber, “OpenStreetMap: user-generated street maps,” *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.
- [65] C. Yang, H. Hongo, and S. Tanimoto, “A new approach for in-vehicle camera obstacle detection by ground movement compensation,” *Conference on Intelligent Transportation Systems (ITSC)*, pp. 151–156, 2008.
- [66] P. Negri, X. Clady, S. M. Hanif, and L. Prevost, “A cascade of boosted generative and discriminative classifiers for vehicle detection,” *EURASIP J. Adv. Signal Process*, 2008. [Online]. Available: <http://dx.doi.org/10.1155/2008/782432>
- [67] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, sep 2010.
- [68] G. Silveira and E. Malis, “Real-time visual tracking under arbitrary illumination changes,” in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, June 2007, pp. 1–6.
- [69] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, Jul 2002.
- [70] Y. Wu and J. Fan, “Contextual flow,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 33–40.

- [71] B. D. Lucas, T. Kanade *et al.*, “An iterative image registration technique with an application to stereo vision.” in *IJCAI*, vol. 81, 1981, pp. 674–679.
- [72] O. Tuzel, F. Porikli, and P. Meer, “Pedestrian detection via classification on riemannian manifolds,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 10, pp. 1713–1727, 2008.
- [73] I. Austvoll and B. Kwolek, “Region covariance matrix-based object tracking with occlusions handling,” in *Computer Vision and Graphics*. Springer, 2010, pp. 201–208.
- [74] J. Van De Weijer, T. Gevers, and A. D. Bagdanov, “Boosting color saliency in image feature detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 1, pp. 150–156, 2006.
- [75] T. Gevers, J. Van De Weijer, and H. Stokman, “Color feature detection,” *Color image processing: methods and applications*, vol. 9, pp. 203–226, 2006.
- [76] S. T. Birchfield and S. Rangarajan, “Spatiograms versus histograms for region-based tracking,” in *Computer Vision and Pattern Recognition, CVPR*, vol. 2. IEEE, 2005, pp. 1158–1163.
- [77] S. S. Nejhum, J. Ho, and M.-H. Yang, “Online visual tracking with histograms and articulating blocks,” *Computer Vision and Image Understanding*, vol. 114, no. 8, pp. 901–914, 2010.
- [78] J. Ning, L. Zhang, D. Zhang, and C. Wu, “Robust object tracking using joint color-texture histogram,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 07, pp. 1245–1263, 2009.
- [79] J. Wang and Y. Yagi, “Integrating color and shape-texture features for adaptive real-time object tracking,” *IEEE Transactions on Image Processing*, vol. 17, no. 2, pp. 235–240, 2008.

- [80] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.
- [81] A. E. Abdel-Hakim and A. A. Farag, “Csift: A sift descriptor with color invariant characteristics,” in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2. IEEE, 2006, pp. 1978–1983.
- [82] P. Scovanner, S. Ali, and M. Shah, “A 3-dimensional sift descriptor and its application to action recognition,” in *Proceedings of the 15th international conference on Multimedia*. ACM, 2007, pp. 357–360.
- [83] G. Willems, T. Tuytelaars, and L. Van Gool, “An efficient dense and scale-invariant spatio-temporal interest point detector,” in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 650–663.
- [84] Z. Kim, “Real time object tracking based on dynamic feature grouping with background subtraction,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.
- [85] B. Babenko, M.-H. Yang, and S. Belongie, “Visual tracking with online multiple instance learning,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 983–990.
- [86] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–511.
- [87] T. Yu and Y. Wu, “Differential tracking based on spatial-appearance model (sam),” in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006, pp. 720–727.

- [88] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 5, pp. 564–577, 2003.
- [89] I. Leichter, M. Lindenbaum, and E. Rivlin, “Tracking by affine kernel transformations using color and boundary cues,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 1, pp. 164–171, 2009.
- [90] L. Wen, Z. Cai, Z. Lei, D. Yi, and S. Z. Li, “Online spatio-temporal structural context learning for visual tracking,” in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 716–729.
- [91] X. Mei and H. Ling, “Robust visual tracking using ℓ_1 minimization,” in *Computer Vision, International Conference on*. IEEE, 2009, pp. 1436–1443.
- [92] H. Grabner, C. Leistner, and H. Bischof, “Semi-supervised on-line boosting for robust tracking,” in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 234–247.
- [93] Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-learning-detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1409–1422, 2012.

Appendix B

Abstract

Road scene understanding is one of key research topics of intelligent vehicles. This thesis focuses on detection and tracking of obstacles by multisensors data fusion and analysis. The considered system is composed of a lidar, a fisheye camera and a global positioning system (GPS). Several steps of the perception scheme are studied: extrinsic calibration between fisheye camera and lidar, road detection and obstacles detection and tracking.

Firstly, a new method for extrinsic calibration between fisheye camera and lidar is proposed. For intrinsic modeling of the fisheye camera, three models of the literature are studied and compared. For extrinsic calibration between the two sensors, the normal to the lidar plane is firstly estimated based on the determination of n known z points. The extrinsic parameters are then computed using a least square approach based on geometrical constraints, the lidar plane normal and the lidar measurements.

The second part of this thesis is dedicated to road detection exploiting both fisheye camera and lidar data. The road is firstly coarse detected considering the illumination invariant image. Then the normalised histogram based classification is validated using the lidar data. The road segmentation is finally refined exploiting two successive road detection results and distance map computed in HSI color space.

The third step focuses on obstacles detection, especially in case of motion blur. The proposed method combines previously detected road, map, GPS and lidar information. Regions of interest are extracted from previously road detection. Then

road central lines are extracted from the image and matched with road shape model extracted from 2D η SIG map. Lidar measurements are used to validated the results.

The final step is object tracking still using fisheye camera and lidar. The proposed method is based on previously detected obstacles and a region growth approach. All the methods proposed in this thesis are tested, evaluated and compared to stateof η the η art approaches using real data acquired with the IRTES η SET laboratory experimental platform.

Keywords: fisheye camera, laser ranger finder, GPS, extrinsic calibrationn, road detection, obstacle detection, object tracking.

Résumé :

La perception de scènes routières est un domaine de recherche très actif. Cette thèse se focalise sur la détection et le suivi d'objets par fusion de données d'un système multi-capteurs composé d'un télémètre laser, une caméra fisheye et un système de positionnement global (GPS). Plusieurs étapes de la chaîne de perception sont étudiées : le calibrage extrinsèque du couple caméra fisheye/télémètre laser, la détection de la route et enfin la détection et le suivi d'obstacles sur la route.

Afin de traiter les informations géométriques du télémètre laser et de la caméra fisheye dans un repère commun, une nouvelle approche de calibrage extrinsèque entre les deux capteurs est proposée. La caméra fisheye est d'abord calibrée intrinsèquement. Pour cela, trois modèles de la littérature sont étudiés et comparés. Ensuite, pour le calibrage extrinsèque entre les capteurs, la normale au plan du télémètre laser est estimée par une approche de RANSAC couplée à une régression linéaire à partir de points connus dans le repère des deux capteurs. Enfin une méthode des moindres carrés basée sur des contraintes géométriques entre les points connus, la normale au plan et les données du télémètre laser permet de calculer les paramètres extrinsèques. La méthode proposée est testée et évaluée en simulation et sur des données réelles.

On s'intéresse ensuite à la détection de la route à partir des données issues de la caméra fisheye et du télémètre laser. La détection de la route est initialisée à partir du calcul de l'image invariante aux conditions d'illumination basée sur l'espace log-chromatique. Un seuillage sur l'histogramme normalisé est appliqué pour classifier les pixels de la route. Ensuite, la cohérence de la détection de la route est vérifiée en utilisant les mesures du télémètre laser. La segmentation de la route est enfin affinée en exploitant deux détections de la route successives. Pour cela, une carte de distance est calculée dans l'espace couleur HSI (Hue, Saturation, Intensity). La méthode est expérimentée sur des données réelles.

Une méthode de détection d'obstacles basée sur les données de la caméra fisheye, du télémètre laser, d'un GPS et d'une cartographie routière est ensuite proposée. On s'intéresse notamment aux objets mobiles apparaissant flous dans l'image fisheye. Les régions d'intérêts de l'image sont extraites à partir de la méthode de détection de la route proposée précédemment. Puis, la détection dans l'image du marquage de la ligne centrale de la route est mise en correspondance avec un modèle de route reconstruit à partir des données GPS et cartographiques. Pour cela, la transformation IPM (Inverse Perspective Mapping) est appliquée à l'image. Les régions contenant potentiellement des obstacles sont alors extraites puis confirmées à l'aide du télémètre laser. L'approche est testée sur des données réelles et comparée à deux méthodes de la littérature.

Enfin, la dernière problématique étudiée est le suivi temporel des obstacles détectés à l'aide de l'utilisation conjointe des données de la caméra fisheye et du télémètre laser. Pour cela, les résultats de détection d'obstacles précédemment obtenus sont exploités ainsi qu'une approche de croissance de région. La méthode proposée est également testée sur des données réelles.

Mots-clés : caméra fish-eye, télémètre laser, GPS, calibrage, détection de route, détection d'obstacle, suivi d'objet.

The logo for the SPIM (École doctorale SPIM) features the letters 'SPIM' in a large, white, sans-serif font. To the left of the letters is a blue horizontal bar.

■ École doctorale SPIM - Université de Technologie Belfort-Montbéliard

F - 90010 Belfort Cedex ■ tél. +33 (0)3 84 58 31 39

■ ed-spim@univ-fcomte.fr ■ www.ed-spim.univ-fcomte.fr

