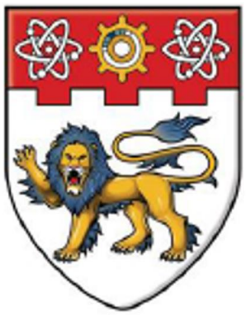


NANYANG TECHNOLOGICAL UNIVERSITY

Vision based Scene Understanding for Collision Avoidance on Roadway

WU MEIQING

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

2016

Vision based Scene Understanding for Collision Avoidance on Roadway

WU MEIQING

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2016

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor Prof. Thambipillai Srikanthan for his valuable and professional guidance, constructive suggestions and unceasing support during the whole period of my Ph.D. study. I am also very thankful for his guidance in presentation skill and sharing of positive life attitudes during weekly meetings. It is a great honor for me to be one of Prof. Thambipillai Srikanthan's Ph.D. students and obtain his personal guidance.

My heartfelt thanks also goes to Prof. Lam Siew Kei for his continuous guidance, valuable advices and unending patience. Prof. Lam Siew Kei has given up a lot of his spare time for the scientific discussion and refinement of my thesis. His insightful inputs have been of great contribution in bringing this thesis to this level of completion.

I would like to express my appreciation to all the team members that I have worked with at Hardware and Embedded Systems Lab (HESL). Special thanks goes to Nirmala Ramakrishnan and Zhou Chengju for the time spent in close technical discussion and the suggestions on improving the research papers.

My thanks also goes to Mr. Chua Ngee Tat for the technical and logistic support throughout the course of my Ph.D. study.

Last but not the least, I would like to thank my family and my friend Gu Xiaoxia for their continuous encouragement and spiritual support.

Abstract

Collision Avoidance Systems (CASs) are attracting a lot of attention as one of the most preferred solutions for advanced driver assistance and autonomous driving. However, scene understanding, which is an essential functionality in CASs, remains a major challenge mainly due to the need for real-time understanding of highly dynamic and complex environment. In this research, a number of robust and low complexity vision based scene understanding techniques for collision avoidance on roadway have been proposed.

It has been well recognized in the literature that road surface detection in a dynamic environment is both challenging and computationally intensive. An efficient non-parametric road surface detection algorithm that exploits the depth cue is proposed to overcome the limitations of existing road surface detection methods. Unlike existing methods that attempt to fit the road surface into rigid models, the proposed method results in low computational complexity, mainly due to the reliance on four intrinsic road scene attributes observed under stereo geometry. It has been demonstrated that the proposed method is capable of detecting both planar and non-planar road surfaces. Extensive experimental results using three challenging benchmarks (i.e. enpeda, KITTI stereo/flow, and Daimler) show that the proposed road surface detection algorithm outperforms the baseline algorithms both in terms of detection accuracy (up to 23.12%) and runtime performance (up to 95.00%).

Next, robust and low complexity algorithm for computing the ego-vehicle's motion state is proposed. The proposed method estimates the ego-motion of the vehicle by first employing a novel pruning technique to reduce the computational complexity of the corner feature detection process without compromising on the quality of the extracted corner features. A robust and compute-efficient KLT tracker is proposed to facilitate the generation of the feature correspondences. Finally, an early RANSAC termination condition is introduced to the Gaussian-Newton optimization scheme to achieve rapid convergence of the motion estimation process. Evaluations based on the KITTI odometry benchmark show that the proposed visual odometry method outperforms the baseline algorithms both in terms of

accuracy (up to 48.36%) and runtime performance. In addition, the proposed algorithm is placed among the top 15% when evaluated using the well-known KITTI odometry platform.

Methods for robust and low complexity stereo-vision based obstacle detection and tracking are proposed. Unlike the works that focus only on the detection of vehicles or pedestrians, the proposed obstacle detection method relies on u-v disparity space to detect all obstacles in the scene. A Space of Interest (SOI) is defined to greatly reduce the search space of obstacles prior to employing adaptive hysteresis thresholding and connected component labeling techniques to segment SOI into sets of obstacles. Method for tracking obstacles across frames is also proposed by constructing a distinctive object appearance model. A number of strategies to further increase the distinctiveness and reduce the computational complexity for constructing the object model are also adopted. Finally, an online multi-object tracking framework is proposed by integrating the obstacle detection and data association modules in a robust way. Evaluations using the KITTI tracking benchmark confirm that the proposed obstacle detection and tracking method outperforms the baseline algorithm in terms of tracking accuracy by up to 51.78%. In addition, compared to the baseline algorithm that achieves about 0.23 frame per second (fps), the proposed method lends well for real-time performance with 20 fps.

Finally, an efficient and robust risk assessment framework is proposed by integrating the obstacle detection and tracking, and visual odometry methods proposed in this thesis. The Extended Kalman Filter is customized to enhance the robustness of the predicted trajectory of the obstacles for assessing the collision risk. The robustness of collision prediction has been enhanced by accommodating positioning uncertainty. Evaluations based on the KITTI tracking dataset demonstrate that the proposed method are capable of robust and efficient assessment of the collision risk in diverse traffic scenarios.

The proposed vision based scene understanding techniques in this research have paved the way towards realizing a real-time capable collision avoidance system that is both affordable and dependable.

Table of Contents

List of Figures

List of Tables

List of Listings

Nomenclature

General Notation

Scalar	regular lower case e.g. u, v, d, x, y, z
Vector	bold lower case e.g. $\mathbf{p}, \mathbf{q}, \mu, \nu$
Matrix	bold upper case e.g. $\mathbf{M}, \mathbf{T}, \mathbf{A}, \mathbf{R}$
Numbers	blackboard upper case e.g. $\mathbb{R}$
Funcitons	fraktur lower case e.g. $\mathfrak{h}, \mathfrak{g}, \mathfrak{f}$
Set	caligraphic upper case e.g. $\mathscr{T}, \mathscr{D}, \mathscr{L}$

Symbols

(u_0, v_0)	camera principal point
Δ	difference in value
θ	camera pitch angle
$baseline$	length of the baseline of the stereo camera rig
d	disparity value
$focal$	camera focal length
i, j, k, n	subscript or superscript index
I_x, I_y	image gradient in horizontal and vertical direction respectively
t_i	i^{th} time step
u, v	pixel coordinate in the camera coordinate system
x, y, z	point coordinate in the world coordinate system
α, β	color histogram

μ, ν	velocity vector
$\mathbf{e}$	6DOF camera motion parameters
$\mathbf{m}, \mathbf{o}$	point in the camera coordinate system
$\mathbf{p}, \mathbf{q}$	point in the world coordinate system
$\mathbf{u}_k$	control vector at k^{th} time step
$\mathbf{w}_k, \mathbf{a}_k$	noise vector at k^{th} time step
$\mathbf{x}_k$	system state at k^{th} time step
$\mathbf{tr}, \mathbf{t}$	camera translation vector
$\mathbf{F}_k$	state transition matrix at k^{th} time step
$\mathbf{H}_k$	observation matrix at k^{th} time step
$\mathbf{J}_{\mathscr{L}}$	Jacobian matrix
$\mathbf{M}, \mathbf{T}, \mathbf{C}, \mathbf{A}$	camera motion transformation matrix
$\mathbf{Q}_k, \mathbf{B}_k$	covariance matrix at k^{th} time step
$\mathbf{RO}, \mathbf{R}$	camera rotation matrix
η	camera projection function
$\mathfrak{f}$	state transition function
$\mathfrak{g}$	camera triangulation function
$\mathfrak{h}$	measurement function
$\mathfrak{l}$	residual function
$\emptyset$	empty set
$\mathscr{D}$	set of detected obstacles
$\mathscr{L}$	set of residual functions
$\mathscr{T}$	set of object tracks

Acronyms / Abbreviations

ADAS	Advanced Driver Assistance System
BilSub	Bilateral Filter based Background Subtraction
BRIEF	Binary Robust Independent Elementary Features

BT	Birchfield and Tomasi's Method
CA	Constant Acceleration
CAS	Collision Avoidance System
CCS	Camera Coordinate System
CSS	Color Self-Similarity
CTRA	Constant Turn Rate and Acceleration
CTRV	Constant Turn Rate and Velocity
CV	Constant Velocity
DARPA	Defense Advanced Research Projects Agency
DEM	Digital Elevation Map
DOF	Degree of Freedom
DPM	Deformable Part Model
EKF	Extended Kalman Filter
ERTC	Early RANSAC Termination Condition
FAST	Features from Accelerated Segment Test
TF	Track Fragmentations
FN	False Negative
FP	False Positive
FREAK	Fast Retina Keypoint
GNO	Gaussian-Newton Optimization Method
GPS	Global Positioning System
GUI	Generation of U-Dispairty Image
GVI	Generation of V-Dispairty Image
HOG	Histogram of Oriented Gradients
IDS	Identity Switches
IMU	Inertial Measuring Units
KITTI	Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago
KLT	Kanade-Lucas-Tomasi feature tracker
LBP	Local Binary Pattern
LOG	Laplacian of Gaussian

MFI	Multi-Frame Integration Visual Odometry Method
MI	Mutual Information
ML	Mostly-Lost
MOTA	Multiple Object Tracking Accuracy
MOTP	Multiple Object Tracking Precision
MT	Mostly-Tracked
NCC	Normalized Cross Correlation
NHTSA	National Highway Traffic Safety Administration
NTSB	National Transportation Safety Board
OpenCV	Open Source Computer Vision Library
OpenMP	Open Multi-Processing API
ORG-KLT	Conventional KLT based Visual Odometry Method
RANSAC	Random Sample Consensus method
SAD	Sum of Absolute Difference
SGM	Semi-Global Matching
SIFT	Scale-Invariant Feature Transform
SMC	Smooth Motion Constraint
SOI	Space of Interest
SSD	Sum of Squared Difference
SURF	Speeded Up Robust Features
TN	True Negative
TP	True Positive
TTC	Time To Collision
TTX	Time To Intersection
VISO2-S	Stereo Scan Visual Odometry Method
WCS	World Coordinate System
WHO	World Health Organization
ZNCC	Zero-mean Normalized Cross Correlation
ZSAD	Zero-mean Sum of Absolute Difference
ZSSD	Zero-mean Sum of Squared Difference

CHAPTER 1

INTRODUCTION

1.1 Motivation

Vehicles have unquestionably improved the quality of people's lives worldwide. However, they also raise serious issues on road safety. According to the global survey on road safety from 182 countries, which is conducted by the World Health Organization (WHO) in 2013 [1], the number of road traffic deaths worldwide remains unacceptably high at 1.24 million per year. In addition, between 20 and 50 million people sustain non-fatal crash injuries on the roads annually. Figure 1.1 illustrates the proportion of road traffic deaths among types of road user within various WHO regions. Globally, 31% of deaths involve car occupants while vulnerable road users including pedestrians (22%), cyclists (5%), motorcyclists (23%) account for almost half of the world's road traffic deaths. The remaining 19% involve unspecified road users. Based on this trend, road traffic fatalities are predicted to rise to the fifth leading cause of death by 2030 [1].

Road traffic crashes can take an enormous toll on individuals and even on national economies [1]. Although measures to ensure road safety have been taken by each country through road infrastructures such as traffic lights or regulating the use of seat belts and installation of airbags, road injuries have not reduced to an acceptable rate. Progress in ensuring road safety must therefore be intensified and accelerated [1]. Due to this reason, there is a high

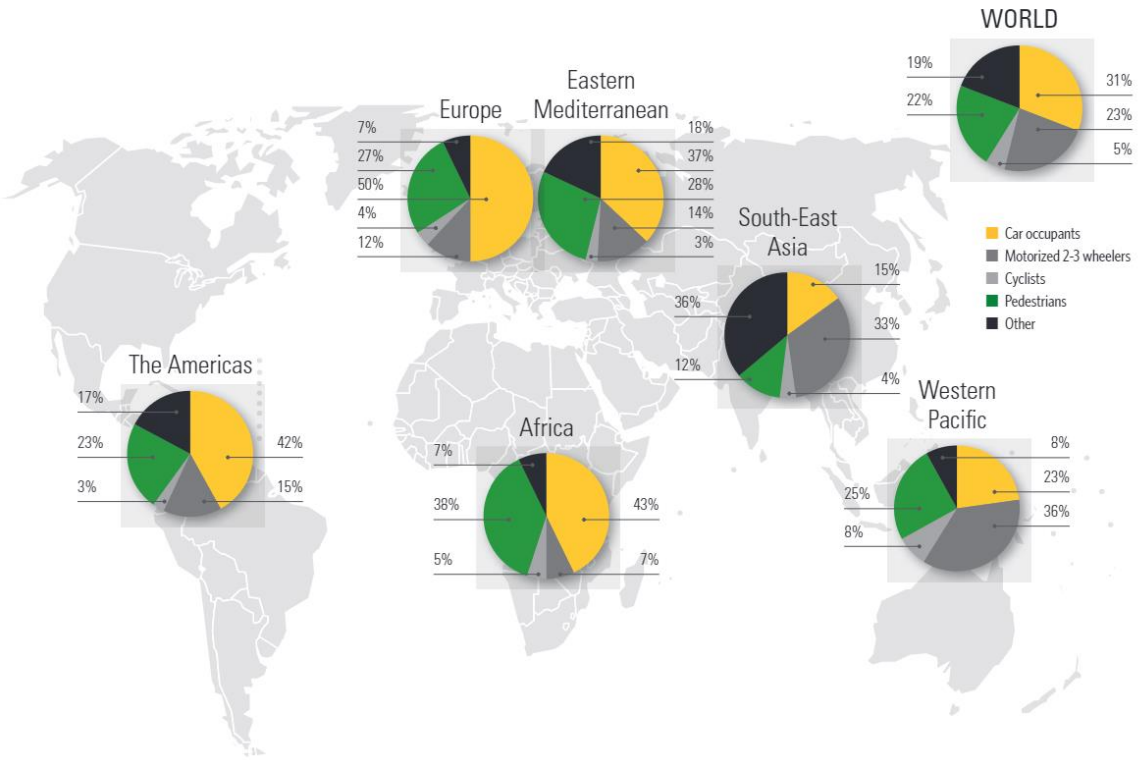


Fig. 1.1 Road traffic death by type of road users and WHO regions: Globally, 31% of deaths involve car occupants while vulnerable road users including pedestrians (22%), cyclists (5%), motorcyclists (23%) account for almost half of the world's road traffic deaths. The remaining 19% involve unspecified road users. Based on this trend, road traffic fatalities are predicted to rise to the fifth leading cause of death by 2030. Figure from [1].

expectation for technologies to improve the road safety and protect vulnerable road users from accidents.

In recent years, due to the rapid progress of modern computer technologies, active intelligent systems for improving road safety have become possible and are increasingly receiving lots of attention from governments, automotive industry and academic community. Instead of taking post-crash measures, such systems are expected to have the ability to predict the imminent collision as early as possible and warn the driver or take pre-crash protective measures to either mitigate the collision or avoid it completely. Such intelligent systems are called Collision Avoidance Systems (CASs) [2]. CASs has been a very active and challenging research field for decades. The National Transportation Safety Board (NTSB) in United States released a recent report on the use of forward collision avoidance systems to prevent and mitigate rear-end crashes in 2015 [3]. The investigation from NTSB found that collision

avoidance technologies show clear benefits to reduce rear-end crash fatalities. Up to 93.7% of crashes might have been prevented, had the vehicles been equipped with the CASs [3].

In spite that significant progress has been achieved, collision avoidance system however remains an unsolved problem [3, 4]. Existing CASs which are deployed on commercial series vehicles usually perform well only in low speed scenarios and respond only to the vehicles driving ahead. The functionality of existing system is limited and needs to be enhanced in order to cover more scenarios [5].

1.2 Scope and Objectives

This PhD research aims at developing vision based scene understanding techniques to assist drivers in avoiding collision on roadway. The main challenge of this research is to develop robust techniques that will enable the collision avoidance system to be deployed in complex and challenging urban traffic environment. At the same time, the proposed solutions must also be computationally efficient so that they are well-suited for realization in embedded systems with limited computing resources.

In particular, the following robust and computationally efficient methods that form the core functional blocks of a vision based collision avoidance system will be devised in this PhD study:

- An efficient road surface detection algorithm that can robustly detect both planar and non-planar roads in realistic dynamic road scenarios.
- A framework for robust and runtime-efficient visual odometry method that determines the motion state for the ego-vehicle[1].
- A robust and low complexity obstacle detection and tracking method that detects and tracks the obstacles to obtain their locations and motion states.
- A method for efficient risk assessment to assess the collision risk between the ego-vehicle and each of the obstacles present in the scene.

[1] Ego-vehicle refers to the vehicle where the collision avoidance system is installed. The motion state of the ego-vehicle is therefore termed as ego-motion. Visual odometry refers to vision based ego-motion estimation.

1.3 Summary of Contributions

The contribution of this research work are summarized as follows:

- A thorough literature survey of existing works for collision avoidance on the road is conducted. The limitations of existing work are identified, and these limitations will form the motivation for devising the proposed techniques in this thesis.
- An efficient non-parametric road surface detection algorithm that exploits the depth cue is proposed. The proposed method has been shown to reliably detect both planar and non-planar road surfaces efficiently. In particular, the proposed method overcomes the limitations of existing parametric methods, which are unable to cope with cases where the road profile doesn't fit the pre-defined model, or when the constantly varying road profiles cannot be modeled mathematically. This contribution has been published in [6].
- A low complexity pruning technique to accelerate the Shi-Tomasi and Harris corner detectors is proposed, which forms an essential part in the visual odometry framework. This contribution has been published in [7, 8].
- A framework for estimating the ego-motion of vehicle that integrates runtime-efficient strategies with robust techniques at various core stages in visual odometry is proposed. The proposed visual odometry algorithm achieves a high ranking in the well-known KITTI odometry evaluation platform by producing accurate ego-motion estimation in notably lesser amount of time. A paper based on this contribution has been submitted to IEEE Transactions on Intelligent Transportation Systems and is currently under the second round of revision [9].
- A robust and low complexity stereo-vision based obstacle detection and tracking method is developed. It has been demonstrated that the proposed algorithm is able to robustly detect and track the obstacles in the presence of drastic scale change, obstacle occlusion and inconsistent illumination. The contributions have been published in [10–12].
- A novel and efficient risk assessment module is devised that takes into account both of the tracked obstacles and ego-vehicle's motion state. In addition, the Extended Kalman Filter is customized to enhance the robustness of the predicted trajectory. The

robustness of collision prediction has been enhanced by accommodating positioning uncertainty. The proposed risk assessment strategy is extensively evaluated using diverse challenging realistic traffic scenarios.

- All the functional blocks, namely, road surface detection (Chapter 3), visual odometry (Chapter 4),obstacle detection (Chapter 5), obstacle tracking (Chapter 5), and risk assessment (Chapter 6) are integrated to provide a holistic vision based scene understanding solution for collision avoidance on roadway.

1.4 Organization of Thesis

This thesis is structured as follows:

Chapter 2 presents a comprehensive review of the state-of-art in Collision Avoidance Systems (CASs). The existing work on different aspects of sensing technologies and functional blocks of CASs are reviewed to identify the open challenges in this research area.

Chapter 3 addresses the problem of detecting planar and non-planar road surface in highly dynamic road scenarios. Inspired by four intrinsic road attributes observed under stereo geometry, simple but efficient non-parametric depth based road surface detection algorithm is introduced and thoroughly evaluated using the well-known enpedia, KITTI, and Daimler datasets.

Chapter 4 addresses the problem of estimating the motion of ego-vehicle during its traversal. A framework for estimating the ego-motion of vehicle that integrates runtime-efficient strategies with robust techniques at various core stages are presented. The proposed algorithm is extensively evaluated using the well-known KITTI odometry dataset.

Chapter 5 focuses on tackling the problem of detecting and tracking obstacles in realistic challenging traffic scenarios. Techniques that are able to enhance the robustness of the proposed techniques in the presence of drastic scale change, obstacle occlusion and inconsistent illumination are devised. The proposed algorithm incorporates low complexity strategies so that it is well suited for real-time[2] realization. The proposed obstacle detection and tracking are thoroughly evaluated using the well-known KITTI tracking dataset.

[2]real-time means that the processing can be performed at least at the same rate that the image frames are captured.

Chapter 6 addresses the problem of assessing the collision risk in the environment. A robust and efficient risk assessment method is proposed which relies on the results of obstacle detection, tracking and visual odometry. The proposed technique is extensively evaluated on challenging scenarios.

Chapter 7 concludes the research works presented in this thesis and discusses the future research direction in this area.

1.5 Publication List

The following lists the papers that have been published in international conferences and journals by the author of the thesis.

Journals:

[J1] **Meiqing Wu**, Siew-Kei Lam, and Thambipillai Srikanthan. "Nonparametric Technique Based High-Speed Road Surface Detection." *IEEE Transactions on Intelligent Transportation Systems* 16.2 (2015): 874-884.

[J2] Nirmala Ramakrishnan, **Meiqing Wu**, Siew-Kei Lam, and Thambipillai Srikanthan, "Enhanced Low-Complexity Pruning for Corner Detection", *Journal of Real-Time Image Processing* (2014):1-17.

[J3] **Meiqing Wu**, Siew-Kei Lam, Thambipillai Srikanthan, "A Framework for Fast and Robust Visual Odometry." *IEEE Transactions on Intelligent Transportation Systems*, 2016 (Under Revision).

Conferences:

[C1] **Meiqing Wu**, Chengju Zhou, Thambipillai Srikanthan, "Robust and Low Complexity Obstacle Detection and Tracking." *IEEE Conference on Intelligent Transportation Systems(ITSC)*, Brazil (Rio de Janeiro), 2016 (Accepted).

[C2] **Meiqing Wu**, Siew-Kei Lam, and Thambipillai Srikanthan, "Stereo based ROIs Generation for Detecting Pedestrians in Close Proximity", *IEEE Conference on Intelligent Transportation Systems(ITSC)*, China (Qingdao), 2014.

[C3] **Meiqing Wu**, Siew-Kei Lam, Thambipillai Srikanthan and Tushar Shah, "Vision-based Pedestrian Tracking System using Color and Motion Cue", *International Symposium on Integrated Circuits (ISIC)*, Singapore, 2014.

[C4] Nirmala Ramakrishnan, **Meiqing Wu**, Siew-Kei Lam, Thambipillai Srikanthan, "Mask-based Non-Maximal Suppression with Iterative Pruning for Low Complexity Corner Detection", *International Symposium on Integrated Circuits (ISIC)*, Singapore, 2014.

[C5] Nirmala Ramakrishnan, **Meiqing Wu**, Siew-Kei Lam, Thambipillai Srikanthan, "Automated Thresholding for Low Complexity Corner Detection", *NASA/ESA Conference on Adaptive Hardware and Systems*, UK (Leicester), 2014.

[C6] Gaurav Mishra, Yan Lin Aung, **Meiqing Wu**, Siew-Kei Lam and Thambipillai Srikanthan, "Real-Time Image Resizing Hardware Accelerator for Object Detection Algorithms", *4th International Symposium on Electronic System Design (ISED)*, Singapore, 2013.

[C7] **Meiqing Wu**, Nirmala Ramakrishnan, Siew-Kei Lam, Thambipillai Srikanthan, "Low-complexity pruning for accelerating corner detection." *The IEEE International Symposium on Circuits and Systems (ISCAS)*, Korea (Seoul), 2012.

CHAPTER 2

LITERATURE REVIEW

This chapter presents a comprehensive review of state-of-art Collision Avoidance Systems (CASs). As CAS is an important component in Advanced Driver Assistance Systems (ADASs) [13–17] and Autonomous Driving [18–21], the role of CASs in these applications will first be reviewed. The architecture of a CAS can be typically decomposed into three parts, i.e. environment perception, risk assessment and actuation [4, 19]. Existing CASs can be grouped into different categories in terms of the types of working sensors used for environment perception. Based on this categorization, a survey of existing CASs from the commercial and academic sector that have been deployed in vehicles will then be provided, which reveals that the camera sensor, whether used stand-alone or as part of sensor fusion, is widely acknowledged as the major sensor option for CASs. Finally, a detailed review of research efforts in each of the functional blocks of vision based CASs is presented and their limitations are summarized. These limitations form the motivation of the research undertaken in this thesis.

2.1 Collision Avoidance for Advanced Driver Assistance Systems and Autonomous Driving

2.1.1 Advanced Driver Assistance System

Safety enhancement has always been one of the main pursuits in the automotive industry. Passive safety features such as seat belts and airbags focus on reducing the effect of damage in case of an accident. On the other hand, active safety systems are increasingly being used to help keep a car under control with the aim of preventing the occurrence of accidents [16]. Active safety systems such as antilock braking system, traction control system and electronic stability program have become commonplace in most cars today. Although such systems have contributed to accident prevention, they are still far from meeting the safety expectations of drivers today [16, 22]. Recently, thanks to the huge progress of the sensor and computer technologies, attention has been shifted to more advanced systems that can assist the driver for decision making in the driving process to avoid accidents. These systems are referred to as Advanced Driver Assistance Systems (ADASs) [14, 23–25].

ADASs can be divided into a number of sub-systems based on their functionality [16]. These sub-systems include lane departure warning system, lane change assistance system, adaptive cruise control, traffic sign recognition system, driver monitoring system, pedestrian protection system, collision avoidance system (CAS), etc. In particular, CAS refers to the system that can monitor the driving environment, predict imminent collision, warn the driver or take automatic action to either mitigate the collision or avoid it completely [2]. The authors in [13] have conducted a classification of these ADAS sub-systems based on the type and amount of impact each sub-system has on the road safety and traffic efficiency. It is evident from Table 2.1 that CAS plays an important role in ADASs in that it has high impact on both the road safety and traffic efficiency.

2.1.2 Autonomous Driving

Autonomous driving refers to the ability of the vehicle to sense the environment and navigate without the human input [26–28]. There is a tight connection between ADASs and Autonomous Driving. ADASs form the critical first step in a slow transition to autonomous driving [29], while autonomous driving can be considered as one step beyond driver assistance [30]. In 2013, The U.S. Department of Transportation's National Highway Traffic

Table 2.1 Classification of ADAS sub-systems based on road safety impact and traffic efficiency impact [13].

		Road Safety Impact	
		High	**Low**
Traffic Efficiency Impact	**High**	Adaptive cruise control, Lane change & merge, Collision avoidance, Vision enhancement	Automated transactions, Platooning, Real-time traffic & traveller information
	Low	Automatic stop and go, Speed control, Obstacle and pedestrian detection, Intersection collision warning, Integrated navigation, electronic mirror, Driver identification, Hands-free & remote control, Driver vigilance monitoring, Driver health monitoring, Road and lane departure	Navigation routing, Parking & reversing aid, Tachograph, Alerting systems, Vehicle diagnostics,

Safety Administration (NHTSA) defined vehicle automation as having five levels [31], which is shown in Table 2.2. According to the definition in Table 2.2, CAS is a level 2 automation system and serves as the basis for self-driving automation.

2.1.3 Collision Avoidance System

CAS, also known as pre-crash system, aims at identifying potential collision and reducing the corresponding damage on the road [3]. It has been pointed out that the ability to initiate emergency braking half a second ahead of time has the potential to avoid 60% of collisions. This figure can be increased to 90% if a warning one second ahead of time is provided [32].

As depicted in Figure 2.1, a collision avoidance system is typically composed of three stages, i.e. environment perception, risk assessment and actuation [4, 19]. Environment perception relates to the sensing and analysis of the environment information based on a particular type of sensor or fusion of different types of sensors. The core tasks for this stage lie in estimating the motion state, i.e. position and velocity, of ego-vehicle and obstacles present in the scene. The risk assessment module will assess the collision risk in the environment and determine when and how collisions can be avoided. Finally, the actuation module will interpret the command from the previous stage and take the corresponding intervention action. The first two stages, i.e. environment perception and risk assessment, are together referred to as scene understanding in this thesis.

Table 2.2 Five levels of vehicle automation defined by NHTSA [31].

No-Automation (**Level 0**)	The driver is in complete and sole control of the primary vehicle controls – brake, steering, throttle, and motive power – at all times.
Function-Specific Automation (**Level 1**)	Automation at this level involves one or more specific control functions. Examples include electronic stability control or pre-charged brakes, where the vehicle automatically assists with braking to enable the driver to regain control of the vehicle or stop faster than possible by acting alone.
Combined Function Automation (**Level 2**)	This level involves automation of at least two primary control functions designed to work in unison to relieve the driver of control of those functions. An example of combined functions enabling a Level 2 system is adaptive cruise control in combination with lane centering.
Limited Self-Driving Automation (**Level 3**)	Vehicles at this level of automation enable the driver to cede full control of all safety-critical functions under certain traffic or environmental conditions and in those conditions to rely heavily on the vehicle to monitor for changes in those conditions requiring transition back to driver control. The driver is expected to be available for occasional control, but with sufficiently comfortable transition time. The Google car is an example of limited self-driving automation.
Full Self-Driving Automation (**Level 4**)	The vehicle is designed to perform all safety-critical driving functions and monitor roadway conditions for an entire trip. Such a design anticipates that the driver will provide destination or navigation input, but is not expected to be available for control at any time during the trip. This includes both occupied and unoccupied vehicles.

2.2 Existing Collision Avoidance Systems

The development of CASs has achieved significant progress over the last decade. Many automotive manufacturers offer some form of CASs in their vehicles today. A number of research labs from the academic community have also deployed CASs in their prototype vehicles. In this section, existing available CASs are categorized with respect to the working sensors used for environment perception and a comprehensive review of commercial and academic efforts that have resulted in successful deployment of these CASs in vehicles are presented.

2.2.1 Radar based CASs

Radar is a type of active sensor that emits radio waves to the environment and observes their reflection from the objects in the environment [34]. The 'time-of-flight' property allows radar

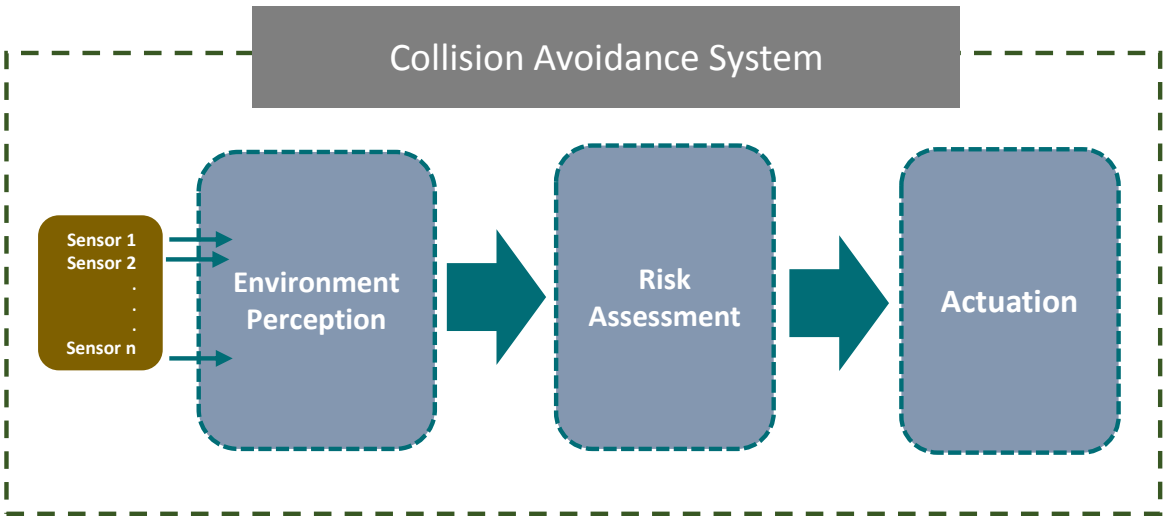


Fig. 2.1 Architecture of collision avoidance system: a collision avoidance system can be typically decomposed into three stages, i.e. environment perception, risk assessment and actuation. Environment perception relates to the sensing and analysis of the environment situation using some sensors. The risk assessment module will assess the collision risk present in the environment and determine when and how collision can be avoided. And the last module, i.e. actuation, will interpret the command from the previous stage and take the corresponding intervention action.

to directly obtain distance and speed information of the observed objects, which enables rapid detection of the objects ahead. Once an imminent crash is detected, corresponding measures are taken.

Commercial radar-based CASs include the following. Toyota developed their first forward collision warning system and launched it in the redesigned Japanese domestic Harrier model in 2003. They later deployed their CAS on the Lexus Ls 430, making it the first CAS offered in American [35]. When Toyota's "Pre-Collison system" determines a collision is unavoidable, the front seatbelts are automatically tightened and the brakes are prepped to assist the driver to avoid or mitigate the collision. In 2003, Honda introduced its CAS with autonomous braking capability in the Inspire and later in Acura models [36]. Mercedes-Benz introduced their first brake assist system "BAS PLUS" on the redesigned W221 S-Class model in 2005 [37]. Audi introduced their CAS called "Braking guard" on Audi Q7 in 2006 [38].

Although radar can quickly obtain the distance information, it has several limitations [34, 39, 40]. Firstly, the cost of radar is not cheap. For example, the Bosch middle range radar costs about 2490 Euros [41]. In addition, the field of view for radar is small. In order for

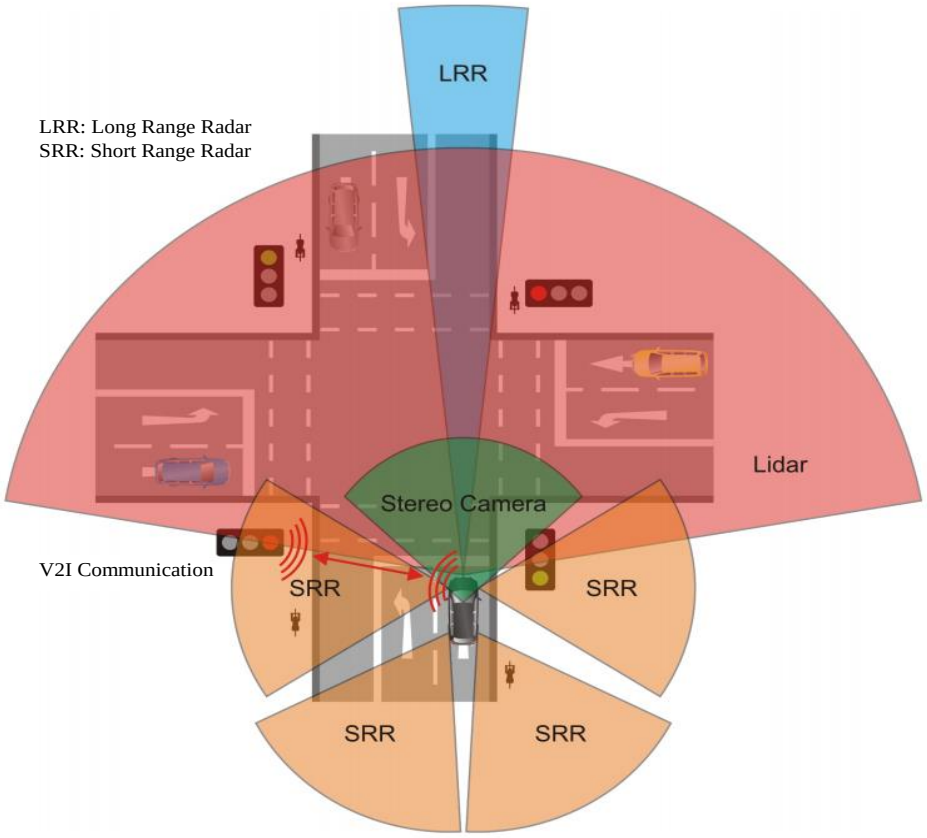


Fig. 2.2 Sensors deployed on the prototype vehicle. Figure from [33].

radar to work effectively, it is often necessary to use several radars to cover a wider field of view. Hence, systems using multiple radars are a costly option. In addition, the angular resolution, i.e. spatial resolution[1], of radar is low, which makes it difficult to interpret the corresponding output signal. Finally, radar cannot distinguish signals reflected from vehicles and other objects. Therefore, the above mentioned CASs are restricted to target vehicles moving in the front and in the same direction as the ego-vehicle [4]. These systems which use radar sensor are denoted as the first generation systems.

Both the academic community and the automotive industry are now competing to develop the next generation CASs, which will enrich the existing CASs with more general scene understanding ability [4, 42, 43]. In order to improve the capabilities of the vehicles to perceive environment, other sensor types or fusion of different sensors have been adopted.

[1] spatial resolution is a term that refers to the number of pixels utilized in construction of a digital image.

2.2.2 Fusion of Radar and Camera based CASs

In 2006, in cooperation with mobileye, Volvo introduced their “Collision Warning with Auto Brake" in the 2007 S80 by fusing radar and camera sensor [44]. In the same year, Toyota also featured in their Lexus LS and LS hybrid a further advanced version of pre-collision system. This system fuses four stereo cameras and a more sensitive radar and is able to detect non-vehicle objects like pedestrians and animals for the first time [45]. In 2009, Toyota further improved their pre-collision system in the redesigned Crown by adding a front-side millimeter-wave radar to monitor side collisions at intersections [46, 47]. In 2010, Audi introduced the ‘Pre-sense’ autonomous emergency braking system which adopted twin radar and monocular camera as the working sensors [48]. In 2013, BMW introduced ‘Driving Assistant Plus’ in most of their vehicle models, which detects potential collision by fusing front–facing camera and front radar sensors [49]. This system doesn’t just monitor the vehicles driving ahead but also pedestrians who are approaching the vehicle. In 2013, Mercedes enhanced their pre-safe system in the W222 S-class series by adding pedestrian detection capabilities by fusing stereo camera and radar sensors [50]. In 2014, Volkswagen also introduced pedestrian detection capabilities based on sensor fusion of camera and radar [51].

2.2.3 Fusion of Lidar and Camera based CASs

Instead of relying on the fusion of camera and radar, some other systems utilize lidar, which is another type of active sensors. Unlike radar which emits radio waves, lidar emits laser. Its main advantage is direct and fast generation of distance measurement of a vehicle’s 360 degree surrounding by relying on spinning lasers with a transceiver rate of more than a million times a second [34].

Mazda deployed their “Smart City Brake Support System” with lidar to detect vehicle and other front obstacles in 2013. In the same year, all Volvo’s automobiles came available with lidar sensor to monitor the road situation [52].

Besides the traditional automotive manufacturers, some other companies also show high interest in intelligent vehicles. The most famous one is Google’s driverless car [53, 54]. Google has equipped different types of cars, e.g. Toyota Prius, Audi TT and Lexus RX450h, with the intelligent engine which fuses many sensors and cost up to $150,000 including a $70,000 Velodyne 64-beam Lidar system [54].

The academic community has also undertaken a lot of research in lidar or fusion of lidar and camera based systems. In the DARPA Urban Challenge [55, 56] which was initiated by the American Defense Advanced Research Projects Agency (DARPA) in 2007, many research teams from universities built vehicles to navigate intelligently to avoid other vehicles on the 96 kilometers urban area course at the site of the now-closed George Air Force Base in Victorville, California. In this challenge, six teams successfully completed the task with CMU's Tartan Racing team, with vehicle "Boss", winning first place. The second and third place winner are Stanford Racing team with their vehicle "Junior" and the team Victor Tango from Virginia Tech with their vehicle "Odin". While the Urban Challenge endeavor came closer to urban traffic situations, the streets were wider than usual, the field of view was unobstructed and only a very limited number of traffic participants were present [27]. In addition, all the teams relied heavily on the lidar and manually annotated maps of sub-meter precision for localization and collision avoidance [27]. In 2010, the research group from VisLab lead by Alberto Broggi from University of Parma tested their autonomous vehicle on a long, intercontinental trip from Parma, Italy to Shanghai, China [57]. CAS is an important module in their autonomous vehicle, which also relies on the fusion of a monocular camera and a lidar to provide obstacle information [58].

Although the cost of lidar has reduced recently, it is still very expensive with the lowest price about $8000 for the 16-beam configuration [59]. In addition, both radar and lidar emit signals to the environment, which lead to environment pollution. Signal interference will become a critical problem in realistic environment when a large number of active sensors are deployed in many moving vehicles and emit signals to the environment simultaneously [34]. The SAVE-U project [60] has pointed out that although radar works well in simple test tracks, they become unreliable at 10-15m in realistic scenarios due to signal interference caused by the reflections from other objects.

2.2.4 Camera only based CASs

Recently, based on Mobileye's EyeQ2 chip and sensing solution, BMW introduced a camera-only active safety system which includes city collision mitigation in their BMW i3 [61].

In contrast to active sensors like lidar and radar, camera is a passive sensor which only receives signal provided by natural energy sources and therefore will not lead to environment pollution. In addition, visual cameras have the following advantages [34, 39, 62]. Firstly, they are very cheap. For example, the camera sensor adopted by Mobileye costs just about

$14 [63, 64]. The images obtained are rich in texture and color cues and have high spatial resolution. All these properties facilitate the extraction of discriminative visual features to describe objects' appearance, depth and motion information. It is worth noting that the 3D structure of the environment can also be reconstructed using stereo cameras [65].

Table 2.3 summarizes the advantages and disadvantages of radar, lidar and camera used for environment perception in CASs. It is evident from Table 2.3 that visual cameras play a key role in CASs regardless of whether it is used stand-alone or fused with other sensors for enhancing performance. In the following section, a detailed review of all the functional blocks in vision based collision avoidance system is presented.

2.3 Vision based Collision Avoidance System

As stated in Section 2.1.3, a CAS typically consists of three parts: environment perception, risk assessment and actuation. In vision based CAS, cameras are used for perceiving the environment to retrieve the knowledge of the obstacles' location and their movement patterns relative to the ego-vehicle. Three core tasks are therefore involved in vision-based environment perception: visual odometry (i.e. vision based ego-motion estimation), obstacle detection and obstacle tracking. In order to achieve good performance for obstacle detection, it is essential to have the knowledge of road surface in advanced [65, 66]. Also, stereo matching is relied upon to reconstruct the 3D structure of the environment using stereo cameras. Therefore, the research in vision based CASs have mainly focused on the following modules: stereo matching, road surface detection, obstacle detection and tracking, visual odometry, collision risk assessment and actuation. Similar to [23], road surface detection and stereo matching are classified as preprocessing steps.

A detailed literature survey on the various modules in vision based CASs will be discussed in the following sub-sections.

2.3.1 Preprocessing

2.3.1.1 Stereo Matching

Table 2.3 Comparison between different sensors for collision avoidance systems.

	Sensor Type	DR	FOV	Resolution		Cost	Further Comment	Application		
				Spatial	Range			Single	Fusion-1	Fusion-2
Passive Sensor	Camera	S/M	M/L	M/H	M	S	Sensitive to illumination condition	BMW(2013)		
Active Sensor	Radar	S-L	S/M	S	H	M	Works in dark, rain, fog	Toyota(2003); Honda(2003); Mercedes-Benz(2005); Audi(2006)	Volvo(2006); Toyota(2006-2009); Audi(2010); BMW(2013); Mercedes-Benz(2013); Volkswagen(2014);	Mazda(2012); Volvo(2013); Google; VisLab
	Lidar	M/L	S/L	M	H	H	Works in dark			

*L: Large; S: Small; M: Medium; H: High;

*DR: Detection Range;

*FOV: Field of View;

*$Single$: Sensor is used stand-alone; $Fusion\text{-}1$: Fusion of Radar and Camera; $Fusion\text{-}2$: Fusion of Lidar and Camera.

Stereo matching is the process of recovering the 3D structure of the scene from a binocular camera setup. This process is plagued with lots of challenges arising from radiometric distortion, perspective distortion, occlusions, inconsistent depth value along object boundaries, low or repetitive textures and so on [67]. These problems pose a great deal of difficulty in designing stereo matching algorithms for real-time implementation. According to the survey papers [67–70], the main steps of stereo matching consist of choosing an appropriate matching cost and designing a strategy to utilize the matching cost. In addition, based on the strategy that utilizes the matching cost, stereo matching algorithms can be divided into three categories: local methods, global methods and semi-global methods.

A. Matching Cost

Matching cost serves as the criterion for measuring the similarity of image regions in two stereo images. Ideally, pixels that correspond to the same scene point in the two images should have the same intensity or color values, which is termed as radiometrically similar in the literature [71]. However, due to reasons such as different camera settings, vignetting[2], image noise, non-Lambertian surfaces[3], etc. radiometric differences often occur in reality. It is therefore very important to choose a proper matching cost that is robust to radiometric variations. The work in [72] groups the existing matching costs into three major categories: parametric costs, non-parametric costs and Mutual Information (MI). Parametric costs include Sum of Absolute or Squared Difference (SAD/SSD), Normalized Cross Correlation (NCC), their corresponding zero-mean versions ZSAD, ZSSD and ZNCC, BT [73], Laplacian of Gaussian (LOG) [74], Bilateral Filter based Background Subtraction (BilSub) [75] and so on. The Rank and Census costs proposed in [76] are non-parametric costs. H. Hirschmuller and D. Scharstein conducted a comprehensive comparison of all the matching cost described above on images without radiometric difference, with simulated and real radiometric differences respectively in [72]. The results show that the performance of matching cost can depend on the underlying stereo matching algorithm. In addition, the authors observed that BilSub performs consistently very well for images with low radiometric differences; MI is slightly better as a pixel-wise matching cost is used for some special cases and for images with strong image noise; and Census cost gives the best and most robust overall performance on all test sets.

[2]Vignetting is a term used in photography and optics to represent a phenomenon that the image's brightness or saturation at the periphery is reduced compared to the image center. Vignetting is often an unintended and undesired effect caused by camera settings or lens limitations.

[3]Lambertian surface is a term representing diffusely reflecting surface. The observed brightness of a Lambertian surface is the same regardless the view angle of the observer.

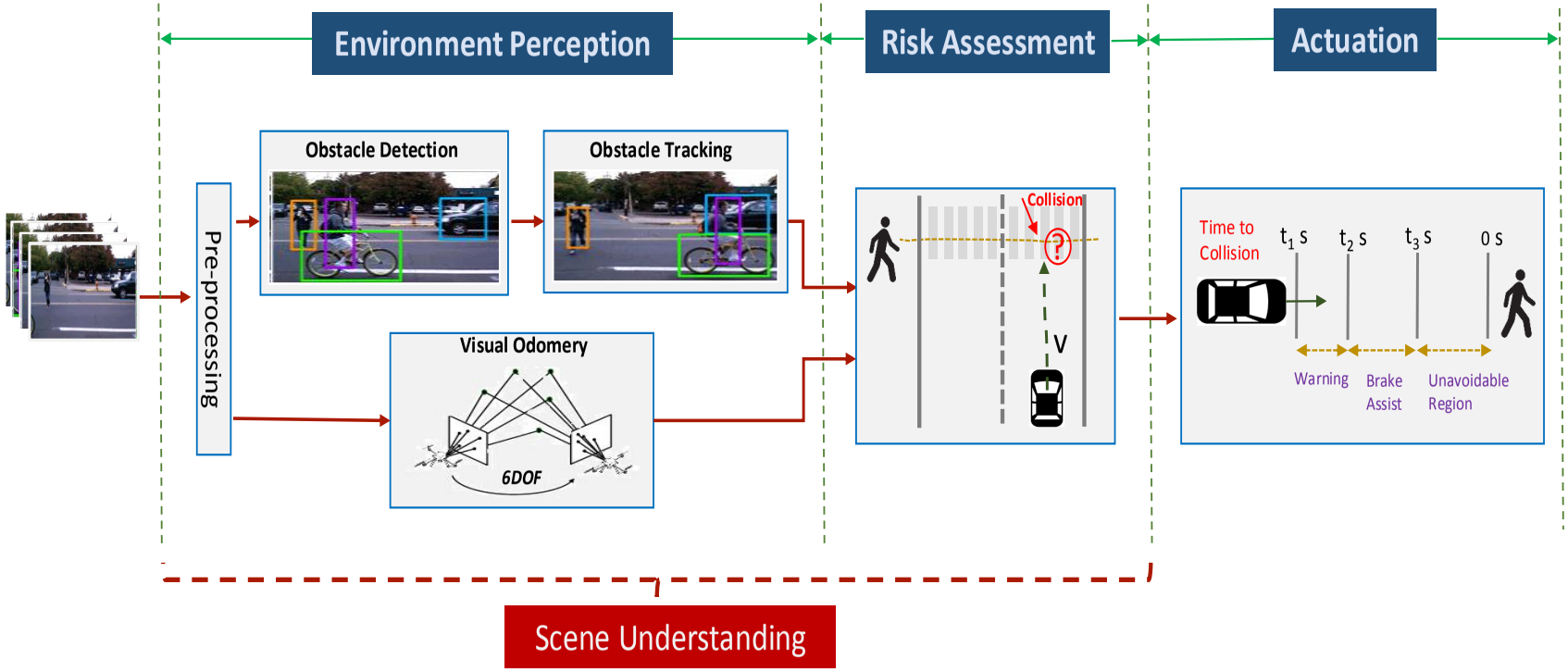


Fig. 2.3 Vision based collision avoidance system: cameras are used for environment perception to retrieve the knowledge of obstacles' location and their movement patterns relative to the ego-vehicle. Three core tasks are therefore involved in vision-based environment perception: visual odometry (i.e. vision based ego-motion estimation), obstacle detection and obstacle tracking. In order to achieve good performance for obstacle detection, it is essential to have the knowledge of road surface in advanced. Also, stereo matching is relied upon to reconstruct the 3D structure of the environment using stereo cameras. Therefore, the research in vision based CASs have mainly focused on the following modules: stereo matching, road surface detection, obstacle detection and tracking, visual odometry, collision risk assessment and actuation. Similar to [23], road surface detection and stereo matching are classified as preprocessing steps. The term "Scene Understanding" refers to the first two stages, i.e. environment perception and risk assessment.

B. Stereo Matching Framework

Once the matching cost is determined, the next step is to utilize the matching cost to establish correspondence between the two images. Based on the strategy that utilizes the matching cost, stereo matching algorithms can be divided into three categories: local methods [67, 74, 77–85], global methods [86–94] and semi-global methods [95]. For local methods, the computation of the disparity value for a given point depends only on intensity values within a finite neighborhood window, usually making implicit smoothness assumptions by aggregating support. Global methods formulate the stereo matching as an optimization problem and find the final disparity map as the minimum solution of a global cost function. In general, local methods achieve real-time performance at the price of reduced matching accuracy. On the other hand, global methods provide higher matching accuracy but suffers from slow execution time. In order to achieve high stereo matching quality that meet the real-time requirement, Hirschmuller originally proposed the semi-global matching (SGM) framework [95]. The SGM method is based on the main framework of local stereo matching methods. However, cost aggregation is formulated as a global cost function and approximated by path-wise optimizations from all directions. The final disparity map is obtained using winner-takes-all scheme. In the works [95–97], the Mutual Information is adopted as the matching cost while the works in [98, 99] utilized the Census cost. SGM and its variants have been receiving increasing attention in the research community as they are capable of obtaining competitive results at a relatively higher speed.

All of the stereo matching methods aforementioned match each stereo pair independently and do not utilize any temporal relationship between the stereo sequences. Recently, researchers have started to exploit the temporal consistence between the successive frames to achieve better stereo matching results [100–104].

Stereo matching is able to recover the 3D scene structure and help to reduce the whole system's complexity. However, stereo matching itself is a computationally intensive task and can become the system’s bottleneck. According to [67], global methods in general produce better matching results. However, they are more time consuming which make them unsuitable for real-time applications like ADASs applications. Local methods are relatively faster but lead to poorer matching results. Hence, the challenge in devising an efficient stereo matching algorithm often lies in finding a good balance between accuracy and runtime performance. The SGM method and its variants are henceforth proposed to resolve this problem. Due to its good balance between accuracy and runtime performance, the semi-global matching methods have become increasingly popular in automotive applications

that require real-time implementation [105]. Recently, Andreas Geiger *et al.* have provided the first platform to evaluate the existing stereo matching algorithms with a challenging dataset captured in realistic traffic environment [106]. It can be observed from this evaluation platform that even when a large error (3 pixel) is tolerated, the best quality achieved at the time this thesis is written still has a 3.43% error with a long computation time of 265 seconds per stereo pair when executed on a platform with >8 cores @3.0 Ghz CPU. The fastest algorithm's computation time is 0.1 second per stereo pair on a platform with 1 core @2.5 Ghz CPU. However, the quality achieved is far from satisfactory (25.27% error rate with 58.54% reconstructed density only). For the aforementioned SGM algorithm [95], the percentage of erroneous pixels in total is 10.86% with the computation time 1.1 seconds on a platform with 1 core @2.5 GhZ CPU. Therefore, in spite that significant progress has been achieved in this area, there is still much room for improvement. In particular, there is an urgent need for a low complexity stereo matching algorithm that will not compromise heavily on the required accuracy for ADASs applications.

2.3.1.2 Road Surface Detection

The task of detecting road surface is the first key step towards the realization of automotive related applications [107, 108]. There are two main benefits of road detection. Firstly, road detection obtains the free space that the vehicle is allowed to proceed. Secondly, road surface detection is an important part of scene geometrical structure exploitation. Obtaining the road surface in advance can greatly reduce the search space for detecting obstacles and help to increase the detection accuracy by rejecting the false positives at an early stage [65, 109].

Road surface detection in realistic road scenarios faces a myriad of challenges due to the following reasons. The road surface can be planar or non-planar (e.g. up-hills, down-hills, and undulating hills). The shape of the roads can also vary (e.g. straight or curved). In addition, the presence of crowded objects, cluttered background, varying illumination condition, dappled shadows make road surface detection extremely challenging. In order to deal with such high variability in road scenes, many cues such as color, texture, and depth[4] are often exploited for detecting the road surfaces.

A. Monocular Vision based

Road color and texture are the two important cues exploited for monocular vision based road surface detection. Thorpe *et al.* [108] classifies the image points as ‘road’ or ‘non-road’

[4]depth refers to the distance in the Z direction relative to the ego-vehicle.

using multiclass adaptive color classification. The work in [110] estimates the road shape by extracting lane markings. However, lane markings are not present in many realistic road scenarios, hence the method in [110] is only applicable to specific road scenes. The approach in [111] detects road surfaces by combining the results of road boundary estimation based on intensity image and road-area segmentation based on color image. The authors in [112, 113] estimate the vanishing points of roads by exploiting the texture cue first and then localizing the road boundary using the color cue. The work in [114] proposes a road detection algorithm by combining low-level, contextual and temporal cues in a Bayesian framework. However, this approach imposes a high computational cost.

B. Stereo Vision based

There is also a large body of work that exploits the depth cue for road surface detection. By restricting the problem to planar roads, the work in [115] observed that the corresponding longitudinal road profile[5] can be projected as a diagonal straight line in the v-disparity image. Curve fitting techniques e.g. Hough transform [115], Radon Transform [116], linear regression scheme [117] and so on are then adopted to extract the straight line. Instead of working in v-disparity space, the method proposed in [118] works in the Euclidean space and detects the road surface by fitting the 3D road data points into a plane using RANSAC based least-squares approach.

Unfortunately, the stereo vision based techniques mentioned above restrict the problem to handle only planar roads, hence limiting their applicability in many real world scenarios. Road surfaces are often highly unstructured due to up-hills, down-hills, undulating hills, road speed bumps, etc. Recently, researchers are increasingly shifting their efforts to deal with the non-planar road geometry. In addition to considering the planar road, Labayrade *et al.* also propose a method to model the non-planar road surface as a succession of parts of oblique planes [115]. The corresponding longitudinal road profile is formulated as a piecewise linear curve. Based on this model, a global road profile is extracted in [119], and a classification and propagation operation is performed to refine the road profile. Beside the piecewise planar model, quadratic model [120] or clothoid model [121] are also utilized to approximate the road surface. However, all of the aforementioned techniques allow for road slope changes in only one direction [66]. This motivates the work in [66] to represent the road surface as a general parametric B-spline curve. However, determining the surface parameters is an extremely challenging task.

[5] longitudinal road profile refers to the curve resulted from the projection of the 3D road surface onto the Y-Z plane.

Depth based scene geometry exploitation is beneficial to many ADASs applications and has recently received much attention. The existing stereo vision based road surface detection methods attempt to fit the road surface into rigid models (e.g. planar, clothoid or B-Spline), thereby restricting to road surfaces that match specific models. Therefore, these approaches are not robust to deal with realistic environments where the road is highly dynamic and can be of any shape. In addition, the curve fitting strategies employed in the stereo vision based road surface detection techniques incur high-computational complexity making them unsuitable for in-vehicle deployments. There is a need to devise more robust and less complex road detection techniques for practical collision avoidance systems.

2.3.2 Obstacle Detection

From the perspective of ego-vehicle, obstacles in the scene refer to the objects that impede the advancement of the vehicle on the road. These include not only traffic participants like vehicle, bicyclist, motorcyclist and pedestrians but also other road infrastructures like traffic lights, sign posts, barriers, trees and so on. In general, obstacle detection methods can be divided into two categories: monocular vision based and stereo vision based.

2.3.2.1 Monocular Vision based

For monocular vision based methods, due to issues like significant intra-class variance of appearance[6], occlusion, etc., researchers often resort to a general object recognition framework, where an implicit representation of object is learned from training samples [23, 122–126]. Such object detection system can be typically divided into two stages: Hypothesis generation and hypothesis verification.

Hypothesis generation aims at identifying all of the potential locations of obstacles using sliding window scheme [127–129] or based on some cues [130–132]. The challenge in this stage is to generate as few candidates as possible without omitting any true positives in the scene [23]. In addition, with less number of candidates generated, the computational burden of the subsequent classification stage will be reduced.

Hypothesis verification extracts some distinctive features like Haar wavelets [128, 133–142], HOG [127, 131, 143–148], LBP [149, 150], CSS [151], DPM [152–154], shadow [148, 155, 156], symmetry [157, 158], and then classifies the hypothesis into specific objects

[6]intra-class variance of appearance refers to the appearance difference within the same object type while inter-class variance refers to the difference between the different object types

using a trained classifier like support vector machine [127, 143, 149, 150, 152, 154, 159–161], neural network [162–166], adaboost [131, 134, 144, 167–174] and so on.

There exist many kinds of obstacles on the road. Due to the high inter-and intra-class variation of appearance, it is very hard to find a feature pattern that is distinctive for all the different obstacles. Existing monocular based methods mainly focus on vehicle and pedestrian detection only, and hence they are unable to meet the full requirements of CASs.

2.3.2.2 Stereo Vision based

The main benefit of stereo vision over monocular vision is that the former can recover the depth information which is helpful for determining the scene geometry and filtering out many irrelevant regions. As such, stereo based systems have been regarded as the primary choice for obstacle detection [23, 65].

Generally, in order to increase the detection performance, stereo vision based road surface detection is utilized to remove the irrelevant image regions first [65, 66]. Recently, the authors in [65] divide stereo vision based obstacle detection methods into four categories.

The first category is probabilistic occupancy grid map based methods. Probabilistic occupancy grid map [175] represents the scene's structure as a two dimensional lattice. Each cell in the lattice corresponds to a certain area in the scene and maintains the occupancy status which can be a binary value or a probability that the cell is occupied by obstacle. Based on the corresponding mathematical space, the occupancy grid map can be further divided into three variants [176]: Cartesian grid, U-disparity grid, and polar grid. Recently, Franke's team has proposed a well-known work called Stixel World [177–183]. Stixel World is built based on the polar occupancy grid map and each stixel in Stixel World models a certain part of an upright oriented object together with its distance and height. The second category is digital elevation map (DEM) based methods. DEM stores the highest height of the 3-D points contained within the cell. In [120, 184], the authors detect the obstacles based on the fusion of DEM and a density map. The third category employs scene flow segmentation schemes that focus on recovering the motion of the scene and tessellate the moving objects from static ones by fusion of the depth and motion information [185–190]. The last category is a generic solution. Full 3D scene reconstruction is build. Clustering based on geometric and texture information is undertaken to segment the scene into a set of obstacles [191–197].

Among the four categories for the existing stereo vision based obstacle detection methods, the occupancy grid maps based method is most widely adopted due to its efficiency. However,

robust segmentation of occupancy grid maps or DEM without dividing single objects into several parts or merging different objects into one is still an unsolved issue. On the contrary, scene flow schemes and general geometry based clusters are able to provide a full 3D representation of the scene. However, they are very time-consuming. In addition, the performance of such systems heavily depends on the stereo matching quality. For example, the system may perform poorly when the output of stereo matching is noisy. The stereo matching algorithm may also become the system's new bottleneck since it is very time-consuming.

2.3.3 Object Tracking

Obstacle tracking is one of the most essential modules for intelligent vehicle's scene understanding. The aim of obstacle tracking is to generate the trajectory of the obstacle by locating its position in every frame. Through tracking, obstacle's behavior can be understood.

Vision based object tracking is challenging due to a number of factors [198]. For example, the urban driving environment is highly dynamic and cluttered. Objects can be static or subjected to regular or abrupt motion. Their appearances can also vary from time to time due to partial occlusion, illumination change and so on. In addition, the tracking algorithm must have low computational complexity as real-time response of vehicle to the environment is essential for avoiding collisions.

Significant progress has been achieved in object tracking during the last decades. Generally, a specific tracker is characterized by several aspects: object localization, appearance model for object association and filtering [199]. A popular taxonomy is made according to object appearance model, which divides the existing works into three main categories [198, 200, 201]: 1) point based tracker, 2) contour based tracker, and 3) kernel based tracker.

2.3.3.1 Point based Tracker

In this category, objects are modeled as a set of points. In [202, 203], the driving scene is sampled with 2000 3D world points. In [204], the obstacles in the world are represented as a set of particles and projected onto a 2D grid map. In general, the performance of point based tracker is tightly related to the chosen number of feature points [204]. Small number of points may not be able to accurately model the scene while larger number of feature points requires huge computation power. Therefore, finding a suitable tradeoff between accuracy and speed is crucial.

2.3.3.2 Contour based Tracker

Contour based tracking tightly depends on the performance of the chosen shape detector, and therefore only shows vitality in dedicated domain for certain objects [200]. In the context of urban traffic, the entire scene is highly cluttered and uncontrolled. Also, there exist many kinds of obstacles and the shape of obstacles may evolve at different time due to motion or change of view angle. Hence, contour based method is not suitable for complex urban traffic scenarios.

2.3.3.3 Kernel based Tracker

In this category, a color histogram is generally built for each object and correspondence is made by comparing the similarity between the color histograms for different objects. Kernel based tracking has been widely studied in the literature and has demonstrated promising results in many areas. Notable works in this area include [199, 201, 205]. Instead of a brute force search, the work in [199] adopts the mean-shift method to locate objects in next frame. Although mean-shift scheme reduces the search space compared to the exhaustive search, it is still compute intensive. The authors in [201] utilizes the concept of fragments to reduce the influence from background and increases the tolerance of the algorithm to occlusion. However, it requires an exhaustive search. Wu *et al.* focuses on increasing the distinctiveness of the appearance model by mitigating the influence from background based on a complex motion model [205].

Despite much progress in object tracking that has been achieved in recent decades, it is still a largely unsolved problem as objects' appearances are easily affected by many factors such as inconsistent illumination, partial occlusion, shape deformation and change of view angle. Designing a distinctive object representation model to make object tracking simple but accurate and robust remains a very challenging problem [200, 206, 207].

2.3.4 Visual Odometry

Vehicle on the road are generally subjected to six degrees of freedom, that is, vehicle might translate and rotate in three directions, i.e. X-Y-Z, in real-world. The knowledge of ego-vehicle's motion state relative to the road serves as the foundation for assessing the risk of collision in Advanced Driver Assistance Systems (ADASs) and autonomous driving. Conventional means of obtaining the motion state of the ego-vehicle rely on Inertial Measuring Units (IMUs) or Global Positioning System (GPS). However, IMUs cannot

provide all the necessary information like the pitch angle and the roll rate [203], while GPS cannot be relied upon to obtain the vehicle's ego-motion in GPS-denied environment e.g. under bridges or urban jungles [208]. As such, vision based methods for ego-motion estimation are becoming increasingly popular as they overcome the drawbacks of IMUs and GPS. In addition, camera-based systems offer other advantages e.g. ease of maintenance and integration into other functionality modules, and reduced cost [209]. Ego-motion estimation that relies solely on vision-based sensing is referred to as visual odometry [210].

Visual odometry in the context of external traffic environments faces huge challenges. Firstly, unlike the indoor environment, the external traffic scene is totally uncontrolled. The scene can be cluttered and contains a lot of moving objects. The scene can also be subjected to inconsistent illumination. As such, visual odometry algorithms must be able to work robustly under such challenging situations. Secondly, computing systems in vehicles are embedded systems with restricted computational resources. The proposed algorithm should therefore be of low computational complexity to allow for in-vehicle deployment.

According to the review in [211, 212], the core stages of visual odometry are feature correspondences setup by feature extraction and tracking, and motion model estimation by solving a mathematical optimization problem based on the set of correspondences. In addition, in order to deal with the noisy correspondences, there is a need for robust estimation.

2.3.4.1 Feature Correspondence Extraction

In the field of visual odometry, point features rather than edge features are preferred since they can be accurately positioned [212]. Point features consist of corner and blob. Popular corner feature detectors are Harris [213], KLT [214], FAST [215], or even much simpler form – maxima and minima of Sobel filter response [216], etc. Popular blob feature descriptors include SIFT [217], SURF [218, 219], FREAK [220], BRIEF [221, 222], etc. Once features are extracted, they will be tracked or matched to find correspondence across frames. The former like Harris and KLT aims to detect features in the first frame and track them in the second frame using local search techniques. The latter like SIFT, SURF aim to detect features independently in both of the images and match them based on a certain similarity measure.

The extraction of reliable feature correspondences across frames in the realistic environments plays a deterministic role in the success of visual odometry [223]. In addition, as illustrated in [216, 223], the computational hot spots of visual odometry lies in the feature detection and tracking.

2.3.4.2 Motion Estimation Model

Once the feature correspondences have been identified, depending on the dimensions of the features, the motion estimation model can be divided into two categories [224]: monocular vision based [209, 210, 225–244] and stereo vision based [210, 216, 223, 245–254]. In the first case, since the features are encoded in 2D, a relative scale factor needs to be determined using methods like trifocal tensor. At least 5 feature pairs are required to obtain the solution, which is found by determining the transformation that minimizes the re-projection error of the triangulated points in each image [210]. For stereo vision, the 3D scene structure can be directly reconstructed through triangulation of the stereo rig. The minimal-case solution involves 3 non-collinear correspondences [210]. When both of the feature correspondences are specified in Euclidean space, that is 3D-to-3D, the solution is found by finding the alignment transformation that minimizes the distances between the correspondences [224]. Instead of minimizing the residuals in Euclidean space (i.e. 3D-3D), a better solution is to work in the image space (i.e. 3D-2D) [254]. In this case, the solution is determined by minimizing the re-projection error.

2.3.4.3 Robust Estimation

Apart from the two main steps discussed above, there are many other issues that must be taken into consideration in order to increase the accuracy of visual odometry. For example, the identified set of feature correspondences are usually contaminated by outliers (noisy or erroneous feature correspondence). Robust estimation methods like M-estimations [255], RANSAC [256, 257] are strategies that have been utilized to increase the accuracy of model estimation in the presence of outliers. In addition, visual odometry is a dead-reckoning algorithm and is prone to error accumulation over time [223]. Therefore, the estimated camera pose can easily drift from the real path. To deal with this problem, some works combine other sensor data from GPS [258] or IMU [259–261] to improve the positioning accuracy. Another popular solution is the bundle adjustment algorithm [259, 262] that imposes geometrical constraints over multiple frames. However, this approach is time consuming. Recently, Badino *et al.* proposes a technique to reduce the motion drift by introducing an augmented feature set that contains the accumulated information of tracked features over all frames [223].

Existing solutions fail to achieve a good balance of high accuracy and low computational complexity [26]. For example, the work in [216] is able to achieve a real-time performance, but its accuracy is far from satisfactory. On the contrary, the work in [223] achieves high estimation accuracy, but are very time consuming.

2.3.5 Risk Assessment

The ultimate goal of CASs is to take necessary measures in a potential collision. Next to the perception of how the scene evolves, it is inevitable for CAS to assess the collision risk between the ego-vehicle and its surrounding obstacles in the near future so that ego-vehicle can react correctly.

Risk assessment is generally decomposed into two steps [263]: First, the future trajectories for obstacles are predicted based on some rules. Second, collision between ego-vehicle and each obstacle is predicted and a risk is derived based on the overall chance of collision.

2.3.5.1 Trajectory Prediction

To predict object's motion, a mathematical model that describes how the situation evolves must be built. Kinematic models is a widely used model to describe an object's motion state in the literature [263]. A detailed survey of kinematic models for vehicle can be found in [264]. The trajectories of vehicle can be straight or curvilinear. The velocity of vehicles can be of constant or varying. Based on this, models like Constant Velocity (CV), Constant acceleration (CA), Constant Turn Rate and Velocity (CTRV) and Constant Turn Rate and Acceleration (CTRA) and so on are proposed. CV and CA are the simplest models, which assume straight trajectories for objects [202, 203, 265–269]. On the other hand, the works in [42, 270–273] use the CTRV or CTRA model to take into account the change of object's curvilinear motion.

Measurements from sensor alone are in generally contaminated by statistical noise and other inaccuracies. Data filtering technology is generally utilized to increase the accuracy of predicted motion states. Kalman filter [274] is a standard technique used to estimate the state of a linear system in the presence of Gaussian noise distribution. The new states of variables are first predicted using the motion model and then corrected using current observations. More precise results about the state estimation can be achieved compared to those based on measurement alone. A lot of works have adopted Kalman filter for object motion estimation [202, 203, 275–279]. Extensions of linear Kalman filter like Extended Kalman filter [42, 270, 271, 280–283] and the Unscented Kalman Filter [137, 284–286] and the more general case like Particle Filter [184, 204, 287–289] are designed to work on nonlinear systems.

2.3.5.2 Collision Prediction

The assessment of collision risk between each pair of ego-vehicle and one surrounding obstacle must be undertaken based on a proper metric. Several risk metrics have been proposed in the literature [265, 290–295]. Among these, Time-To-Collision (TTC) corresponds to the remaining time before the collision takes place and is the most popular risk indicator [265, 269, 272, 291, 296, 297]. The smaller TTC is, the higher the collision risk is. Based on TTC, collision risk can be quantified and proper prevention action can be designed accordingly.

Given the future trajectories on both entities, collision can be estimated by solving the equations of the motion models corresponding to ego-vehicle and obstacle and locating the intersection point between two trajectories. If both the ego-vehicle and obstacle arrive at the intersection point at the same time step, a collision is detected [267, 269] and TTC is derived accordingly. Since vehicle or other obstacles have a certain volume, they cannot be simply treated as a point as assumed in physics. Taking into account the size and other factors like positioning uncertainty, a more common solution is to represent the vehicles as polygons [290, 291], circles [298] or ellipses [265] and a collision is predicted by setting a condition on the "overlap between the shapes of the two vehicles" [263]. In general, uncertainties come from the perception process and from the unknown behavior of the traffic participants for the prediction time interval [17]. When taking into account these uncertainties, the more advanced systems compute the occurrence of a predicted collision in a probabilistic manner and the TTC values becomes a probability distribution [5, 17, 21, 299, 300].

Risk assessment for the CASs in the complex urban environment faces a lot of challenges, especially in the presence of great uncertainties associated with the observed sensor data and motion models. Although some work have been proposed for limited scene scenarios, the performance achieved is still far from satisfactory. In addition, existing risk assessment is mainly conducted for vehicle-to-vehicle [5, 290] and pedestrian-to-vehicle [19, 301]. Obstacles on the road, however do not only include vehicles and pedestrians, but also bicyclist, traffic light and so on. Reliably estimating the risk between vehicle and each possible obstacle in the complex urban environments still remains an unsolved problem.

2.3.6 Actuation

When a collision is expected to occur, intervention measures must be in place to avoid the collision. In general, there exist several strategies to intervene the occurrence of a collision.

Such strategies include warning and autonomous intervention like braking and steering or a combination of warning and autonomous intervention.

2.3.6.1 Warning

The warning mode issues an alert about an imminent collision to the driver and it is the responsibility of the driver to react accordingly [4, 32, 302–306]. The type of warning modalities will vary between vehicles. Examples include visual, auditory and/or haptic warning. Designing a friendly human-machine interface is of great importance since it directly affects the driver's acceptance of such system. It has been consistently reported that multi-modal signals rather than a single sensory cue will lead to a faster response time of driver to a specific event. In addition, a lesson learnt over last decade is that the perceivable threat must be present so that driver agree that a threat does exist and need to respond [4].

2.3.6.2 Autonomous Intervention

If the driver can react appropriately and timely to a given warning, they are likely to handle the situation well. However, in some complex and abrupt situations, there may be insufficient time to warn the driver or the driver is not able to react in time to the warning. In such cases, autonomous intervention support e.g. braking [278, 281, 293-296] and steering [4, 19, 307, 308] is highly beneficial. Steering intervention typically requires that the ego-vehicle is not steered into a secondary object.

One concern in autonomous intervention lies in choosing between braking and steering. The authors in [309] point out that collision avoidance by braking is appropriate at low vehicle speeds (e.g. below 50 km/h) while collision avoidance by steering is appropriate at higher vehicle speeds. More recent work [307] is able to automatically choose between braking and steering in arbitrary traffic situation.

The main principle in designing a successful intervention strategy is to warn and/or intervene as early as possible in order to maximize the safety margins without disturbing the driver with unnecessary intervention [4]. Otherwise, false alarms will dramatically reduce the drivers' acceptance rate of such intervention systems.

Achieving the above goal faces great challenges. The biggest one lies in how to ensure the robustness of the intervention strategy in the presence of inaccurate perception and threat assessment results. Secondly, early intervention necessitates the accurate prediction of the evolving scene over a not short time, which significantly increases the algorithmic complexity.

2.4 Summary

A comprehensive literature review of the existing works related to CASs is presented in this chapter. In this section, the key observations that are made from the literature survey are summarized.

A broad review of existing CASs in terms of sensor deployment has been presented. Systems that are based on lidar or radar are very costly, and hence such systems are currently only found in premium vehicles or in prototype vehicles in research labs. On the other hand, camera offers a number of merits such as low cost and they can provide images that are rich in texture and color cues. In addition, they can be easily installed and maintained. The 3D structure of the environment can also be reconstructed using stereo camera. Hence, camera has been regarded as the major sensor option for CASs regardless of whether they are used stand-alone or fused with other types of sensors.

CASs typically consists of three parts: environment perception, risk assessment and actuation. In particular, vision based environment perception can be further divided into functional blocks like stereo matching, road surface detection, obstacle detection and tracking, and visual odometry. A detailed survey of each of the functional blocks is presented in Section 2.3 with the following key findings.

First, stereo matching is able to recover the 3D scene structure and help to reduce the whole system's complexity. However stereo matching itself is a computationally intensive task, and can become the system's new bottleneck. Although significant progress has been achieved in this area, urgent need for a low complexity stereo matching algorithm that will not compromise heavily on the required accuracy for ADASs applications still exists.

Secondly, existing road surface detection methods attempt to fit the road surface into rigid models (e.g. planar, clothoid or B-Spline), thereby restricting to road surfaces that match specific models. These approaches are hence not robust to deal with realistic environments where the road is highly dynamic and can be of any shape. In addition, the curve fitting strategies employed in such techniques incur high-computational complexity making them unsuitable for in-vehicle deployments.

Thirdly, there exist many kinds of obstacles on the road. Due to the high inter-and intra-class variation of appearance, it is difficult to find a feature pattern that is distinctive for all of the different obstacles. Existing monocular based methods mainly focus on vehicle and

pedestrian detection only, and hence they cannot sufficiently serve the CASs. On the other hand, stereo based systems have been regarded as the primary choice for obstacle detection. Existing stereo based obstacle detection methods can be divided into four categories. Among these four categories, the occupancy grid maps based methods are most widely adopted due to their efficiency. However, robust segmentation of occupancy grid maps or digital elevation maps without dividing single objects into several parts or merging different objects into one is still an unsolved issue. On the contrary, scene flow schemes and general geometry based clusters are able to provide a full 3D representation of the scene. However, they are very time-consuming.

Fourthly, an effective object appearance model is crucial for the success of a visual tracker. Despite significant progress in object tracking in recent decades, designing a distinctive object representation model to make object tracking simple but accurate and robust is still an unresolved problem as objects' appearances are easily affected by factors like inconsistent illumination, partial occlusion, shape deformation, change of view angle and so forth.

Fifthly, visual odometry is an essential module in automotive applications. Existing methods reported in the literature faces a lot of challenges in realistic urban driving environment and is therefore prone to motion drift. The extraction of reliable feature correspondences across frames in realistic environment plays a deterministic role in the success of visual odometry. In addition, the computational hot spots of visual odometry lies in feature detection and tracking. Existing solutions fall short of achieving a good balance between high accuracy and low computational complexity.

Finally, risk assessment for CASs in complex urban environment face a lot of challenges, especially in the presence of great uncertainties associated with the sensor data and motion models. Although previous work have been proposed to handle limited scene scenarios, the performance achieved is still far from satisfactory. Reliably estimating the risk between vehicle and each possible obstacle in the complex urban environment still remains an unsolved problem.

It is evident from the literature survey that even though there is a large body of research works in the area of vision based CAS, we are still far from realizing a deployable and affordable system that can work in realistic scenarios. This is due to the fact that existing approaches usually have very high computational complexity that do not lend themselves well for low-cost realization on embedded systems. In the following chapters of this thesis,

robust and computationally efficient techniques to address the key challenges in vision-based CASs will be proposed.

As the road surface detection has been shown to lead to significant increase in the performance of obstacle detection and tracking, an efficient non-parametric road surface detection algorithm that exploits the depth cue is proposed in the next chapter.

CHAPTER 3

NONPARAMETRIC TECHNIQUE BASED HIGH-SPEED ROAD SURFACE DETECTION

Road refers to the drivable area that allows vehicle to proceed without encountering obstructions that would prevent the onward traversal. As discussed in Section 2.3.1.2, the knowledge about the position of the road is of great help to other automotive tasks like obstacle detection. A comprehensive review of existing road detection algorithms in the literature has been presented in Section 2.3.1.2. It is well recognized that detecting road surface in a realistic environment is a challenging problem that is also computationally intensive.

In this chapter, a nonparametric road surface detection algorithm that relies on the depth cue only is proposed. Unlike existing methods, the proposed road surface detection algorithm does not fit the road surface into fixed mathematical models. Instead, it relies on simple but effective strategies that are based on four special attributes present in realistic road conditions. By formulating these observations, the proposed algorithm is able to work with highly dynamic road scenarios at low computational complexity. The contributions in this chapter has been published in [6].

This chapter is organized as follows: The stereo camera geometry and related concepts are introduced in Section 3.1. The proposed road surface detection algorithm is presented in Section 3.2. In Section 3.3, experimental results to demonstrate that the proposed algorithm outperforms the existing well-known techniques both in terms of detection accuracy and runtime performance in various complex road scenarios are presented. Finally, Section 3.4 summarizes the contributions made in this chapter.

3.1 Stereo Geometry Model

In this section, the mathematical principles residing in the stereo camera geometry are first explained in detailed. The description of three important concepts pertaining to the stereo geometry, i.e. the disparity map, u-disparity image and v-disparity image, will then be presented. The contents presented in this section serves as the mathematical foundation for Section 3.2.

3.1.1 Stereo Camera Geometry

A stereo camera rig is set up as in [115]. As shown in Figure 3.1, the left and right cameras are positioned at the same plane and at the same height above the road. There exist three coordinate systems, i.e. the world coordinate system WCS and the left and right camera coordinate systems CCS_l and CCS_r, in the stereo camera rig. The principle points of the left camera and the right camera, i.e. the intersection point of the optical axis of the camera and the camera plane, are $\mathbf{o}_l$ and $\mathbf{o}_r$ respectively. A scene point $\mathbf{p} = (x, y, z)^T$ in WCS will be projected onto an image point (u_l, v) in CCS_l in the left image and another image point (u_r, v) in CCS_r in the right image. The epipolar line, i.e. the intersection line resulting from the intersection of the plane constructed by these three points and the camera plane, will be parallel to the baseline of the stereo rig, i.e. the connection line between $\mathbf{o}_l$ and $\mathbf{o}_r$.

Assume θ is the cameras' pitch angle, which is the angle between the optical axis of the camera and the horizontal X-Z plane; *height* is the height of the cameras above the ground; *baseline* is the length of the stereo baseline, i.e. the distance between $\mathbf{o}_l$ and $\mathbf{o}_r$; $focal$ is the focal length measured in pixel; (u_0, v_0) is the image coordinate for $\mathbf{o}_l$ in the left camera system or $\mathbf{o}_r$ in the right camera system. Then the coordinate for the projected image point (u_l, v) in CCS_l or (u_r, v) in CCS_r is given in Eq. 3.1 and Eq. 3.2:

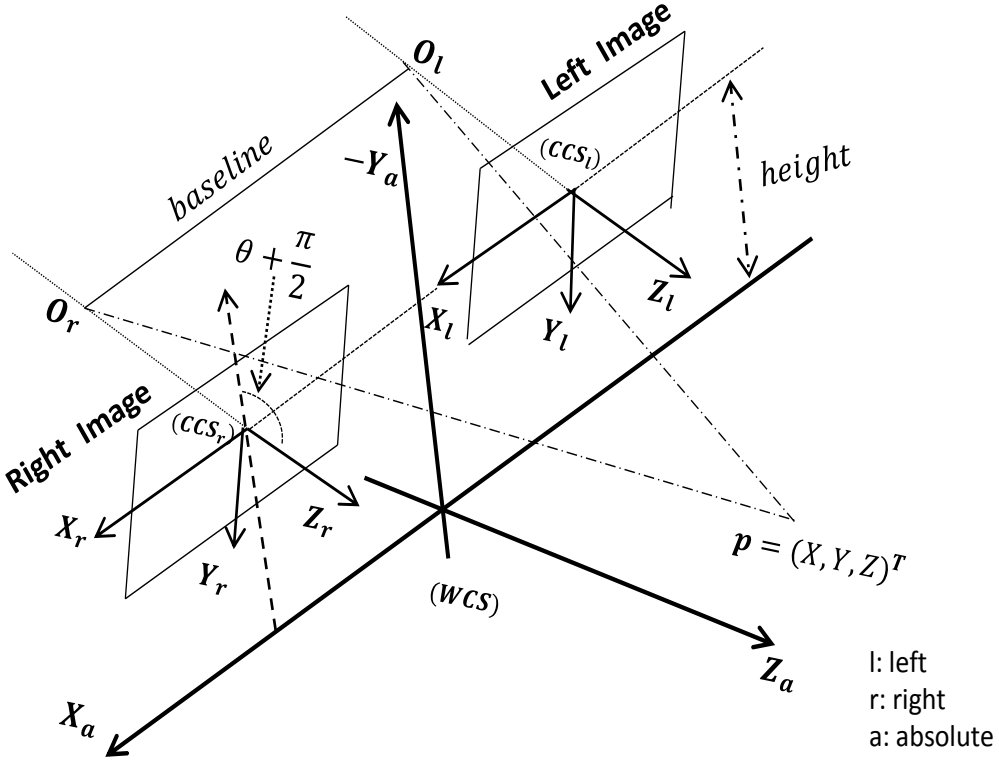


Fig. 3.1 A stereo camera rig: the left and right cameras are positioned at the same plane and at the same height above the road. There exist three coordinate systems, i.e. the world coordinate system WCS and the left and right camera coordinate systems CCS_l and CCS_r, in the stereo camera rig. The principle points of the left camera and the right camera, i.e. the intersection point of the optical axis of the camera and the camera plane, are $\mathbf{o}_l$ and $\mathbf{o}_r$ respectively. A scene point $\mathbf{p} = (x, y, z)^T$ in WCS will be projected onto an image point (u_l, v) in CCS_l in the left image and another image point (u_r, v) in CCS_r in the right image. The epipolar line, i.e. the intersection line resulting from the intersection of the plane constructed by these three points and the camera plane, will be parallel to the baseline of the stereo rig, i.e. the connection line between $\mathbf{o}_l$ and $\mathbf{o}_r$. Figure from [115].

$$u_i = u_0 + \frac{focal * x - \varepsilon_i * \frac{baseline}{2} * focal}{(y + height)\sin\theta + z\cos\theta} \tag{3.1}$$

$$v = v_0 - focal * \tan\theta + \frac{focal * (y + height)}{\cos\theta[(y + height)\sin\theta + z\cos\theta]} \tag{3.2}$$

where the subscript i indicates left or right, $\varepsilon_l = -1$ and $\varepsilon_r = 1$.

The difference between u_l and u_r is referred to as the disparity value d of the point **p**. That is, $d = u_l - u_r$ as given in Eq. 3.3:

$$d = u_l - u_r = \frac{focal * baseline}{(y + height)\sin\theta + z\cos\theta} \tag{3.3}$$

The relationship between d, v and y is then given in Eq. 3.4:

$$d = \frac{baseline * (v\cos\theta - v_0\cos\theta + focal\sin\theta)}{y + height} \tag{3.4}$$

Note that when θ is small enough, $\sin\theta \approx 0$, and the term $(y + height)\sin\theta$ can be ignored. A small θ is assumed in this thesis. In addition, the assumption that the cameras are installed such that the roll angle is negligible is made as in [121, 310, 311].

3.1.2 U-V Disparity Images

A map containing the disparity value for all the points is called the disparity map. From Eq. 3.3, it can be observed that the disparity value d is inversely proportional to the depth value z. This implies that the disparity map encodes the scene's geometrical structure from the view angle of the ego vehicle. For example, for the original left image in Figure 3.2(a), the corresponding disparity map is in Figure 3.2(b), which reflects the 3D structure of the scene. In addition, by transforming the disparity map in certain way, projections of the scene from different view angles can be obtained, which therefore educes the concepts of the u-disparity image and v-disparity image.

The concepts of the u-disparity image and v-disparity image are first proposed in [115]. By accumulating the points with the same disparity in the scan-line of the disparity map, the v-disparity image is obtained and provides a side-view projection of the 3-D scene, i.e. projection onto the Y-Z plane. U-disparity image on the other hand, accumulates the points with the same disparity in a column wise manner and therefore provides a bird's-eye view projection of the scene, i.e. projection onto the X-Z plane. The height of the v-disparity image is equal to the height of the disparity map while the width of the u-disparity image equals to the width of the disparity map. In addition, both of the width of the v-disparity image and the height of the u-disparity image corresponds to the maximum disparity value in the disparity map, which corresponds to the distance in z direction the scene has spanned.

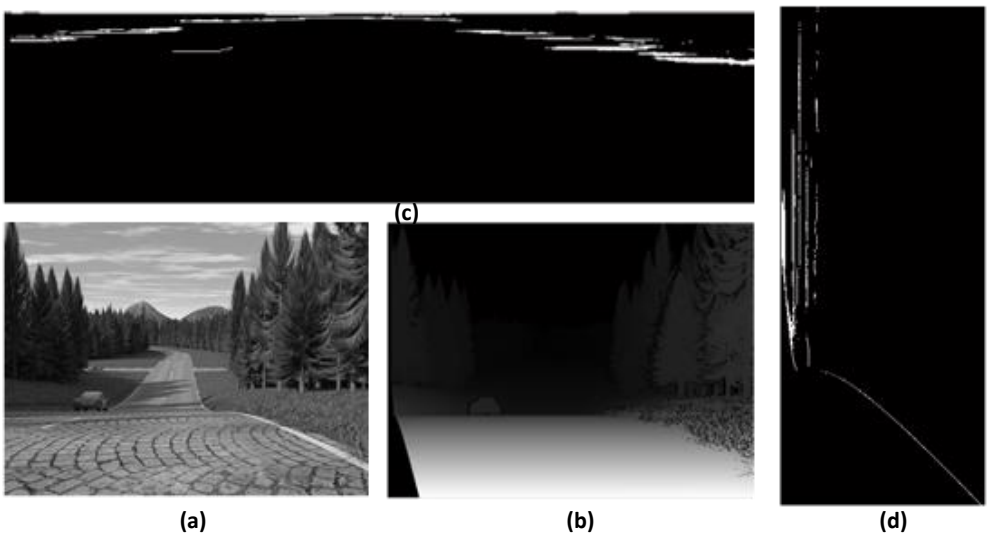


Fig. 3.2 Illustration of u-v disparity images: (a) original image; (b) disparity map; (c) the corresponding u-disparity image; (d) the corresponding v-disparity image. Detailed explanations of the concepts about disparity map, u-disparity image and v-disparity image can be found in Section 3.1.2 and Section 3.2.1. Note both of the u-disparity image and v-disparity image are enlarged for better visualization.

The corresponding pseudo codes to generate the u-disparity image and v-disparity image are given in Listing 3.1 and Listing 3.2 respectively.

For the original image in Figure 3.2(a) and the corresponding disparity map in Figure 3.2(b), the corresponding u-disparity image and the v-disparity image are illustrated in Figure 3.2(c) and (d) respectively.

Listing 3.1 Generation of U-Disparity Image (*GUI*)

Input: Disparity map $disMap$;
Output: U-disparity Image $udisImg$;

```
1: maxdisp ← maximum disparity value in disMap;
   /* create a matrix with size(maxdisp, disMap.width) and initialize it with value 0 */
2: udisImg = zeros(maxdisp, disMap.width);
3: for j = 0 : disMap.height − 1 do
4:    for i = 0 : disMap.width − 1 do
5:       tmpd = disMap[j][i];
6:       udisImg[tmpd][i] = udisImg[tmpd][i] + 1;
7:    end for
8: end for
```

Listing 3.2 Generation of V-Disparity Image (*GVI*)

Input: Disparity map $disMap$;
Output: V-disparity Image $udisImg$;

```
maxdisp ← maximum disparity value in disMap;
vdisImg = zeros(disMap.height, maxdisp);
for j = 0 : disMap.height − 1 do
    for i = 0 : disMap.width − 1 do
        tmpd = disMap[j][i];
        vdisImg[j][tmpd] = vdisImg[j][tmpd] + 1;
    end for
end for
```

3.2 Proposed Algorithm

Existing methods have mainly concentrated on fitting the shape of road surface into a specific model. However, it is very difficult to find a single model that can accommodate all road scenarios since the environment is highly dynamic. As such, a novel road surface detection algorithm that is based on the road scene's intrinsic attributes under a stereo geometry is developed in this section.

In this section, four important road scene attributes observed under the stereo geometry will first be discussed. The way how these attributes are combined and formulated into an efficient road surface detection algorithm will then be showed.

3.2.1 Road Scene Attributes under Stereo Geometry

The example of a road scene depicted in Figure 3.3 shows a challenging road scenario, i.e. non-planar road with undulating hill and dynamic obstacles on the road. When the road scene is mapped to the stereo coordinate system as described in Section 3.1, several intrinsic road scene attributes in the u-v disparity images are identified, which will enable us to distinguish the road surface from the obstacles. The following describes these observed attributes:

1. A longitudinal road line is defined as a set of road points which have the same x value but varying z values. According to Eq. 3.1 & 3.3, a longitudinal road line will fall into distributed regions in the u-disparity image. On the other hand, up-right obstacle points with the same x and z values will converge onto the same position in the u-disparity image, therefore producing peak regions, i.e. pixels with high values in the u-disparity image. Figure 3.2(c) clearly illustrates this concept.

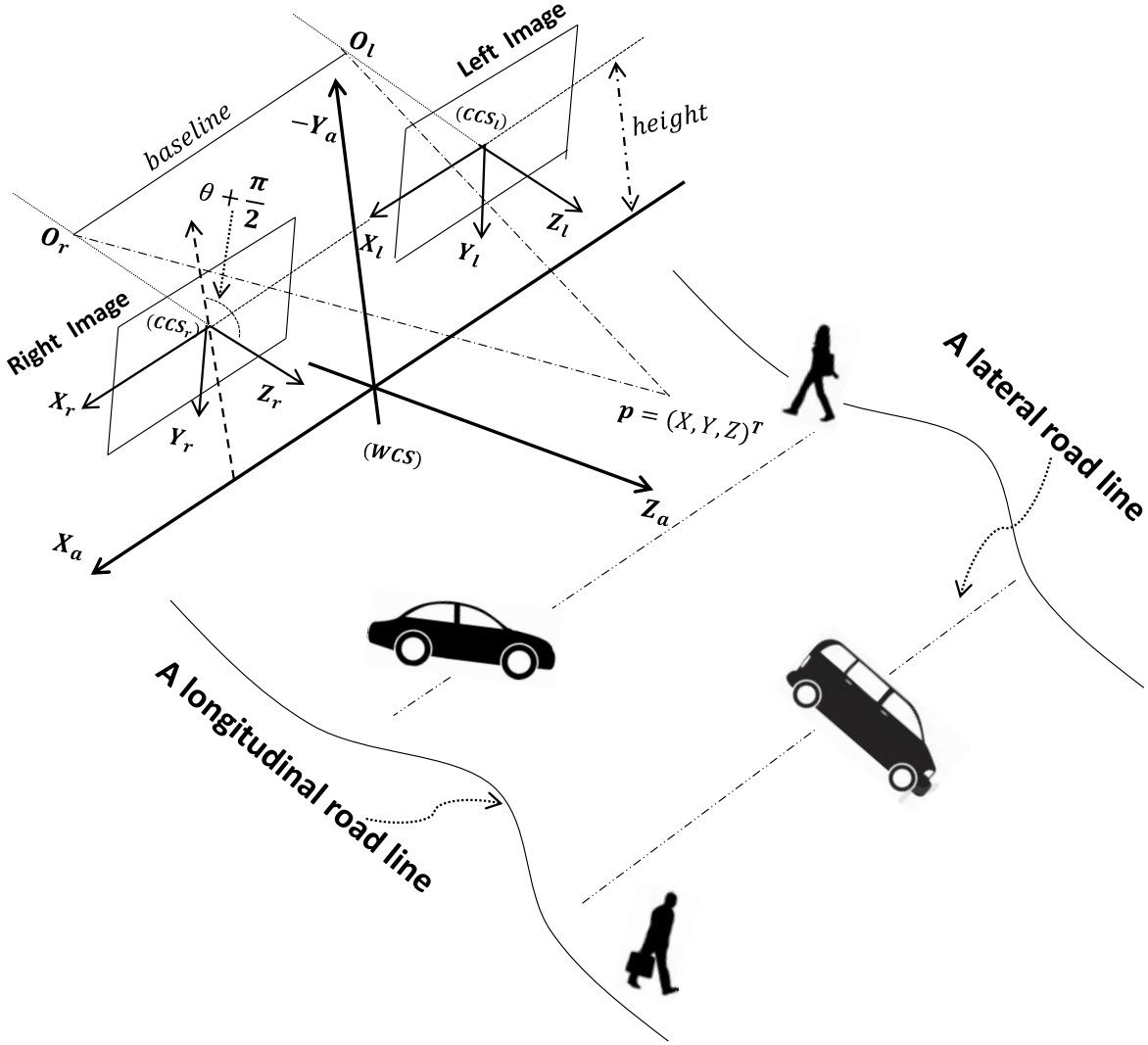


Fig. 3.3 An example of a road scenario with undulating hill under the stereo model.

2. A lateral road line is defined as a set of road points which have the same z value but varying x values. In most cases, the points along the same lateral road line have the same value of y or values of y that are very close to each other. According to Eq. 3.2 & 3.3, the lateral road line will be projected onto the same v and associated with the same d, i.e. they will converge onto the same position in the v-disparity image. When a vehicle is moving forward, the road surface will occupy a major part of the scene, especially in the vicinity of the vehicle. This implies that the point with maximum intensity value for each row in the v-disparity image is very likely to be the projection of the points of the corresponding lateral road line. However, this property will be violated when there are lots of obstacles (especially large obstacles) on the road. In

order to increase the confidence that the peak regions in the v-disparity image will correspond to the lateral road lines, the obstacles can be first removed from the disparity map prior to generating the v-disparity image.

3. According to Eq. 3.2 & 3.3, larger values of z will lead to smaller values of v and d. This explains the following phenomenon: due to the perspective projection effect of the camera imaging process, road points that are farther away from the camera will be projected onto the higher part of the captured image. In addition, they will have smaller disparity values.

4. According to Eq. 3.4, when two points are projected onto the same row v, the point with larger value of y will have smaller d. This implies that when a road point and an obstacle point are projected onto the same row v, the z value of the obstacle point is smaller than that of the road point. In addition, the disparity value corresponding to the obstacle point is larger than the disparity value for the road point.

3.2.2 Road Surface Detection

The observed road scene attributes serve as the mathematical foundations of the proposed algorithm presented in this section. Figure 3.4 shows the top-level block diagram of the proposed algorithm. Taking the disparity map as the inputs, the proposed method consists of four stages: 1) Crude obstacles removal; 2) Longitudinal road profile extraction; 3) Determination of the horizon line; 4) Road surface extraction. The input disparity map required by the proposed algorithm can be dense or semi-dense.

3.2.2.1 Crude Obstacle Removal

This step serves as a pre-processing step to facilitate road profile extraction that will be discussed in Section 3.2.2.2. The u-disparity image is resorted to for fast obstacle removal. As explained in Section 3.2.1, obstacles, in particular large obstacles correspond to peak regions in the u-disparity image. In order to identify these peaks in the u-disparity image, a thresholding operation is applied with a threshold denoted as *threshold_largeobj*. If the intensity value of a point in the u-disparity image is higher than *threshold_largeobj*, the point is labeled as obstacle point using label *LOBJ_MARKER*. A new disparity map is then generated after removing the obstacles from the original disparity map.

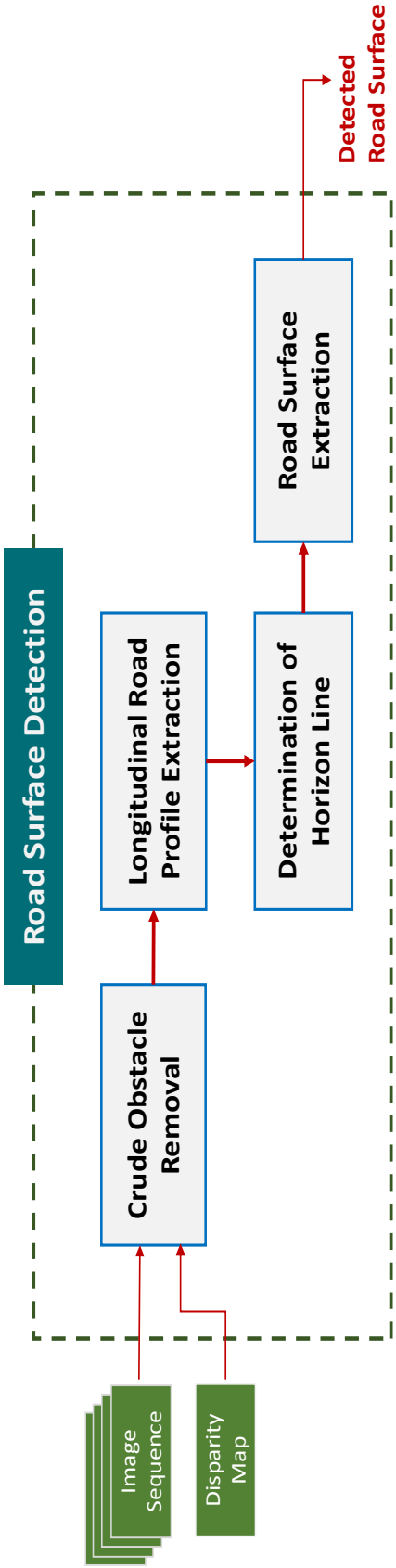


Fig. 3.4 Top-level block diagram of the proposed road surface detection method: Taking the image sequence and the corresponding disparity map as the inputs, the proposed road surface detection methods consists of four stages: 1) Crude obstacles removal; 2) Longitudinal road profile extraction; 3) Determination of the horizon line; 4) Road surface extraction. The input disparity map required by the proposed algorithm can be dense or semi-dense.

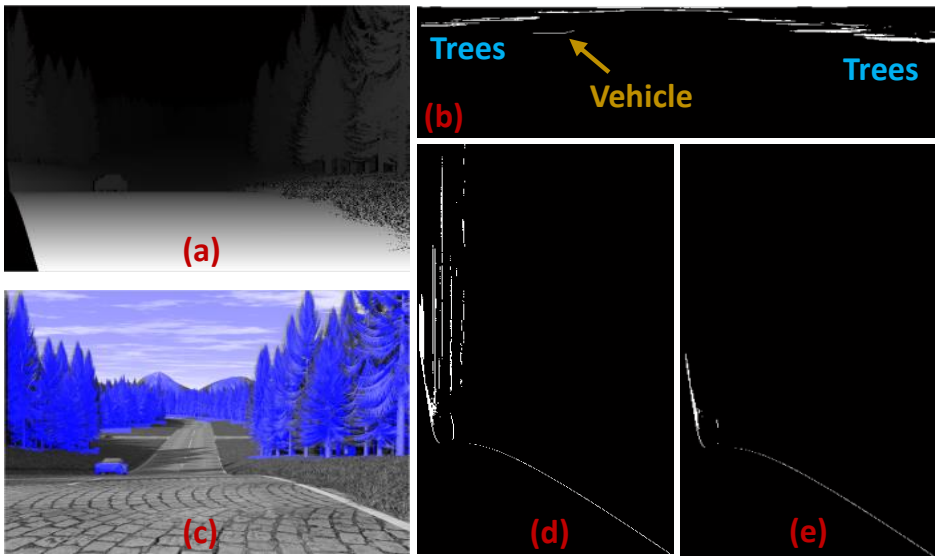


Fig. 3.5 Crude obstacle removal: (a) shows the original disparity map and the corresponding u-disparity image and v-disparity image are shown in (b) and (d). The obstacles in (c), highlighted in blue are identified in the crude obstacle removal step, and the new v-disparity image (after removing obstacles) is shown in (e).

It is noteworthy that this pre-processing stage is not intended at finding all the obstacle pixels present in the disparity map, but rather as a means to reduce the difficulty of extracting the road profile. It can be observed in Figure 3.5(c) that the crude obstacle removal step removes the sky and trees, but retains the short grass. This is however sufficient for increasing the accuracy of road profile extraction which will be discussed in the following section. The pseudo code for the proposed crude obstacle removal method is given in Listing 3.3.

3.2.2.2 Longitudinal Road Profile Extraction

In this step, the longitudinal road profile is extracted based on the v-disparity image, which is generated from the new disparity map after removing the large obstacles. As described in Section 3.2.1, the points with maximum intensity value for each row in the v-disparity image are very likely to correspond to the projections of the lateral road lines after removing the obstacles. These points constitute the initial road profile. Figure 3.6(c) shows an example of the initial road profile for a given disparity map.

In the experiments, it is observed that the majority of the initial road profile points do correspond to the actual road profile, especially in the vicinity of the vehicle. Nevertheless, there can be exceptions due to two reasons. The first reason is the higher portion of the initial road profile may not correspond to the physical lateral road lines. For example in Figure

Listing 3.3 Crude Obstacle Removal

Input: Disparity map $disMap$;
threshold $threshold\_largeobj$;
Large object label $LOBJ\_MARKER$.
Output: A new disparity map without large objects $disMap\_without\_largeobj$.

```
1: udisImg = GUI(disMap); // using algorithm listed in Listing 3.1;
2: disMap_without_largeobj = disMap;
3: for j = 0 : disMap.height − 1 do
4:    for i = 0 : disMap.width − 1 do
5:       current_d = disMap[j][i];
6:       current_u_vote = udisImg[d][i];
7:       if current_u_vote > threshold_largeobj then
8:          disMap_without_largeobj[j][i] = LOBJ_MARKER;
9:       end if
10:   end for
11: end for
```

3.6(c), the green rectangle corresponds to the higher region of the captured image which does not contain the road. Another reason is due to the highly unstructured road scenes or the noisy and erroneous input disparity map. The initial road profile therefore needs to be refined.

According to the third observation in Section 3.2.1, as the road regions extend away from the camera, they will be projected onto the higher portion of the image and will be associated with smaller disparity values. Mathematically, this can be formulated as shown in Eq. 3.5

$$d_{roadprofile(v)} \leq d_{roadprofile(v+1)} \tag{3.5}$$

where $d_{roadprofile(v)}$ represents the disparity value of the road profile point in row v. Eq. 3.5 is therefore used as a criterion to examine the validity of the initial road profile. This examination process is carried out from the bottom row to the highest row in the v-disparity image. If Eq. 3.5 is violated for some row, the next point with the largest intensity that conforms to Eq. 3.5 is determined as the new road profile point for that row.

There are two important issues that need to be addressed during the refinement of the road profile. Firstly, Eq. 3.5 implies that the refinement of road profile for current row depends on its previous row. It is therefore very important to determine a suitable initial road profile point to begin the refinement process. As can be observed in the blue inset of Figure 3.6(c),

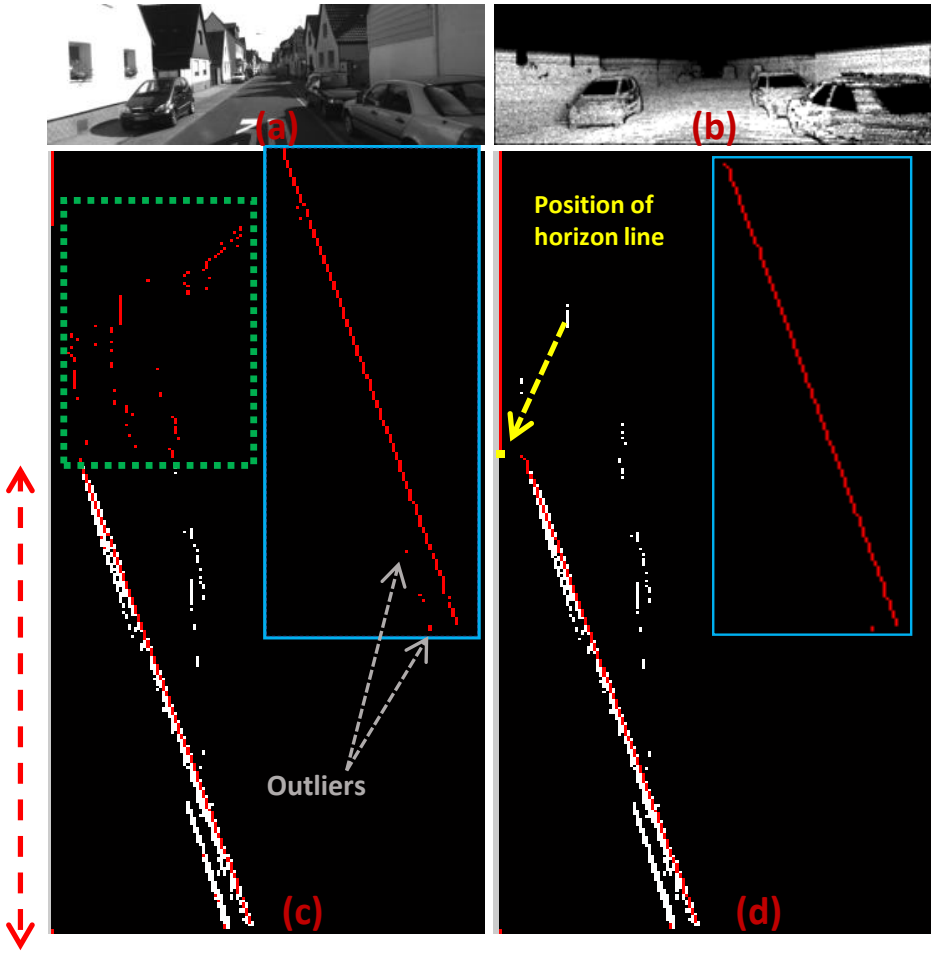


Fig. 3.6 Road profile extraction: (a) shows a captured image and its corresponding disparity map with enhanced contrast is shown in (b). (c) shows the corresponding v-disparity image where the curve highlighted in red is the initial road profile (see blue inset for enhanced visualization of the range indicated by the red vertical dotted line), (d) the curve highlighted in red in the v-disparity image is the refined road profile. The row highlighted by yellow dot is determined as the horizon line. The part of the refined road profile under the horizon line is determined as the final road profile (see blue inset for enhanced visualization). The v-disparity image has been enlarged for better visualization.

it is possible that the initial road profile point for the first row is not an actual road profile point when the disparity values for most of the pixels in the first row of the disparity map are uncertain or erroneous. At this time, it is an outlier. If the refinement process is begun from this row, the extraction of the following road profile points will be affected. From the experiments, it can be observed that a good road profile point to start the refinement process is one that possesses the largest disparity value among the first five rows. The row corresponding to that road profile point is treated as the starting row.

Secondly, Eq. 3.5 is able to distinguish the outliers from the initial road profile that appear on the right side of the actual road profile. However, in a few cases, outliers may appear on the left side of the actual road profile due to noisy input. As illustrated in Figure 3.6(c), the outliers can be easily visually identified as they are located far from the previous rows. However, in the subsequent rows, the initial road profile points get closer again. If this situation is not corrected, Eq. 3.5 will cause the extracted road profile to deviate from the actual road profile in the subsequent rows during the refinement process. In order to resolve this issue, during the refinement of the initial road profile, if the current row's initial road profile point conforms to Eq. 3.5, the distance $\Delta d = d_{roadprofile(v+1)} - d_{roadprofile(v)}$ is checked. If Δd is larger than some threshold value denoted as *threshold_close*, then the first road profile point in the subsequent rows within a limited range whose disparity value is larger than $d_{roadprofile(v)}$ is found. The corresponding row is denoted as v'. If $d_{roadprofile(v')} < d_{roadprofile(v+1)}$, the initial road profile point for row v is then classified as an outlier. Then the point located in the range $[d_{roadprofile(v+1-threshold\_close)}, d_{roadprofile(v+1)}]$ that has the largest intensity in the corresponding v-disparity image will be selected as the new road profile point.

3.2.2.3 Determination of the Horizon Line

According to the third observation in Section 3.2.1, the further away the road is, the smaller the values v and d will become. Therefore, if the extracted road profile stops extending towards the left for a predefined number (*threshold_largeobj*) of rows in the v-disparity image, this means that the road will cease to be visible in the road scene at that particular row. This row is the horizon line as illustrated in Figure 3.6(d). The part of extracted road profile under the horizon line is kept as the final road profile. The pseudo code to extract the road profile is given in Listing 3.4.

3.2.2.4 Road Surface Extraction

Once the longitudinal road profile is extracted, according to the fourth observation in Section 3.2.1, it should be straightforward to extract the road surface as follows. For the image portion under the horizon line, the image points whose disparity values are smaller than or equal to the corresponding road profile are classified as road points, otherwise, they are regarded as obstacle points. Regions with invalid disparity values [1] are labeled according to their neighbors. This is the methodology adopted in [115]. However, the disparity map is often noisy. Some points in the same lateral road lines may not have the same disparity value in the given disparity map. For example, some road points may have larger disparity values than the majority of the road points for that scan-line. Hence, this approach will result in a lot of small blobs that are wrongly labeled.

A strategy to extract the road surface in a more controlled fashion is proposed:

1. For each row v in the disparity map, the image point whose disparity value is smaller than or equal to $d_{roadprofile(v)}$ is classified as road point. In addition, the image point whose disparity value is larger than $d_{roadprofile(v)}$ by certain degree denoted as *threshold_variance* and is not labeled as obstacle point (based on the outcome of crude obstacles removal as discussed in Section 3.2.2.1), is also classified as road point.

2. The input disparity map may contain invalid regions whose disparity values are uncertain. Hence, it is necessary to perform interpolation for the invalid regions. The interpolation is performed in a scan-line manner. For each continuous invalid span, its left and right neighboring spans are checked to see if they are of the same label. If the left and right neighboring spans have the same label, then the invalid scan-line is assigned the same label as its neighbors. Otherwise, the invalid scan-line is assigned the same label as its larger neighbor.

3. At this time, there may still be some wrongly classified regions consisting of thin horizontal blobs within the correctly classified regions. Post-processing must then be undertaken to filter out these wrongly classified blob regions. The filtering process is carried out by checking its neighbors in a column-wise manner. For each column in the disparity map, small vertical spans consisting of points with the same label are identified. If the length of the span is smaller than *threshold_largeobj* and also notably

[1] The disparity maps generated by existing stereo matching algorithms usually contains some regions whose disparity values are not determined.

Listing 3.4 Road Profile Extraction

Input: Disparity map without large object $disMap\_without\_largeobj$;
thresholds $threshold\_close$ and $threshold\_largeobj$;
Output: Road profile $roadProfile$;
horizon line $horizon\_line$.

$vdisImg = GVI(disMap\_without\_largeobj)$;
$r = vdisImg.height$;
/*Extract the Inital Road Profile*/
for $j = 0 : r-1$ **do**
  $first\_maximum\_idx = \arg\max_{i} vdisImg[j][i]$;
  $roadProfile[j] = first\_maximum\_idx$;
end for
/*Determine the Starting Row*/
$first\_robut\_row\_id = \arg\max_{j\in[r-1,r-6]} roadProfile[j]$;
/*Refinement of the Initial Road Profile*/
for $j = first\_robust\_row\_id - 1 : 1$ **do**
  if $roadProfile[j] \leq roadProfile[j+1]$ **then**
    if $roadProfile[j+1] - roadProfile[j] > threshold\_close$ **then**
      $first\_max\_subj = \arg\max_{subj} roadProfile[subj] > roadProfile[j]$;
      if $roadProfile[first\_max\_subj] < roadProfile[j+1]$ **then**
        $idx = roadProfile[j+1]$;
        $another\_idx = \arg\max_{i\in[idx-threshold\_close,idx]} vdisImg[j][i]$;
        $roadProfile[j] = another\_idx$;
      end if
    end if
  else
    $idx = roadProfile[j+1]$;
    $newid = \arg\max_{i\in[1,idx]} vdisImg[j][i]$;
    $roadProfile[j] = newid$;
  end if
end for
$horizon\_line \leftarrow$ the row where road profile stops extends towards left

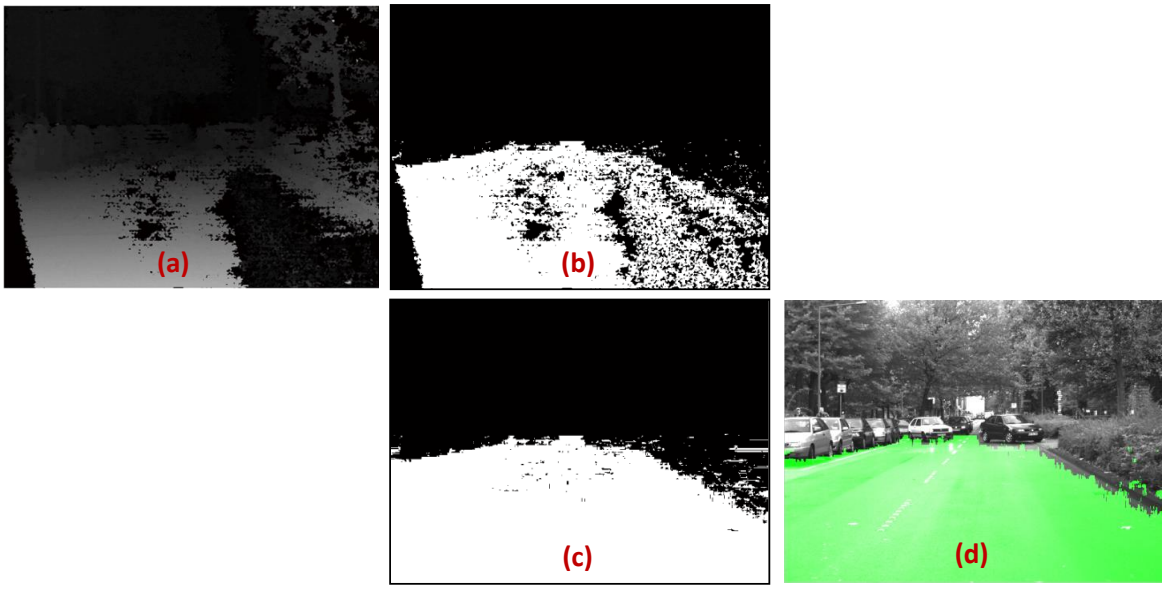


Fig. 3.7 Road surface extraction: (a) shows the noisy input disparity map with enhanced contrast for better visualization; (b), (c) are the intermediate results during the extraction process; and (d) is the final detected road surface for the proposed method.

smaller than the length of its neighboring spans, the points of the vertical span will be re-assigned a new label. That is, if they are labeled as obstacle, they will be re-labeled as road and vice versa. The whole process of road surface extraction is illustrated in Figure 3.7 and the corresponding pseudo code is given in Listing 3.5.

3.3 Experimental Evaluation

In this section, the proposed road surface detection algorithm will be evaluated on three challenging benchmarks and compared with two well-known baseline algorithms both in terms of accuracy and runtime performance.

3.3.1 Experimental Setup

3.3.1.1 Benchmarks

Three large-scale challenging datasets are chosen to evaluate the proposed algorithm:

- The enpeda dataset [312] is a sequence of 394 640*480 synthetic stereo frames containing both planar and non-planar road scenarios. The ground truth disparity map is provided with sub-pixel accuracy for every pixel except for the occlusion part.

Listing 3.5 Road Surface Extraction

Input: Disparity map $disMap$;
Disparity map without large obj $disMap\_without\_largeobj$;
Road profile $roadProfile$;
Horizon line $horizon\_line$;
Threshold $threshold\_variance$.
Output: Road map $roadMap$. // 0: obstacle; 1: road;

```
1:  roadMap = zeros(disMap.height, disMap.width);
    /*Classification*/
2:  for j = horizon_line + 1 : disMap.height − 1 do
3:     separatrix = roadProfile[j] + 1;
4:     for i = 0 : disMap.width − 1 do
5:        current_d = disMap[j][i];
6:        if current_d == INVALID then
7:           roadMap[j][i] = 2; // invalid disparity value
8:        else
9:           if current_d ≤ separatrix then
10:             roadMap[j][i] = 1;
11:          else
12:             if disMap_without_largeobj[j][i]! = LOBJ_MARKER and current_d ≤
                separatrix + threshold_variance then
13:                roadMap[j][i] = 1;
14:             end if
15:          end if
16:       end if
17:    end for
18: end for
    /*Interpolation for Spans with Invalid Disparity Value*/
19: for j = horizon_line + 1 : disMap.height − 1 do
20:    -Divide row j into continuous spans with different label;
21:    -Label the spans with invalid disparity values using its largest neighboring span's label;
22: end for
    /*Post-processing*/
23: for i = 0 : disMap.width − 1 do
24:    -Divide column i into spans with different label;
25:    -Re-assign the small spans with its neighboring span's label;
26: end for
```

(a) (b)

Fig. 3.8 Samples of three datasets for road surface detection: (a) left images of the stereo pairs from the enpeda dataset (first row), the KITTI dataset (second row), and the Daimler dataset (third row) and (b) their corresponding disparity maps for the three datasets used in the experiments. The contrast is enhanced for better visualization. The disparity maps serve as the inputs.

- The KITTI stereo/flow dataset [26] contains 194 1240*376 stereo pairs and the corresponding semi-dense (approximately 50%) ground truth disparity maps. This dataset mainly focuses on planar road scenarios. However, it covers quite an amount of different road contexts.
- The Daimler dataset [313] is a large scale sequence of 21,790 stereo frames captured in busy urban environment with planar and non-planar roads. As no ground truth disparity maps are provided for this dataset, the OpenCV [2] implementation of the Semi-Global Matching algorithm [95] has been chosen to generate the disparity maps. The default parameter settings in OpenCV are used. The obtained disparity maps are noisy and contain large invalid and erroneous regions.

The three chosen datasets constitute a comprehensive test bed which encompasses highly dynamic road situations. Some samples of the three benchmarks and the corresponding disparity maps are shown in Figure 3.8.

3.3.1.2 Baseline Algorithms

The works in [115] and [120] have been chosen as the baseline algorithms. The first baseline algorithm, i.e. [115], is chosen as it is a representative work of the planar road assumption,

[2]OpenCV is a open source computer vision library: http://opencv.org/.

which is the most widely used model to date. In addition, to the best of our knowledge, the work in [115] has presented the lowest computational complexity among all the reported stereo based road surface detection methods in the literature. The second baseline algorithm, i.e. [120], is one of the most recent works in the literature at the time the proposed road surface detection work is developed. In the following, the work in [115] is denoted as *Baseline_A* and the work in [120] is denoted as *Baseline_B*.

3.3.1.3 Implementation Details

Three parameters are required for the proposed algorithm, namely, *threshold_largeobj*, *threshold_close* and *threshold_variance* as introduced in Section 3.2.2. In this thesis, *threshold_largeobj* is set to 12 for all three datasets. Since sub-pixel accuracy is enabled and the ground truth disparity value is provided for every pixel in the enpeda dataset, *threshold_close* is set to 2 and *threshold_variance* is set to 1 for this dataset. For the other two datasets, *threshold_close* is set to 1 and *threshold_variance* is set to 2.

The work in *Baseline_A* formulates the longitudinal road profile as a slanted straight line for planar road and a piecewise linear curve for non-planar road in the v-disparity image. Hough Transform is then utilized to extract the lines. The binary image input to the Hough Transform is obtained by thresholding the v-disparity image with the value 25 in this experiment. For the planar road case, the longest line is extracted and treated as the road profile. Whereas, for the non-planar road case, the 10 highest Hough transform values are extracted. The upper or lower envelope of these 10 lines, depending on the accumulative grey value score, corresponds to the final road profile. For a fair comparison, all of these parameters are obtained after careful tuning. In addition, as explained in Section 3.2.2.4, the road surface extraction module in *Baseline_A* is highly susceptible to noisy input. Again, in order to ensure a fair comparison, the proposed road surface extraction step has been applied to *Baseline_A* in order to increase the its robustness. Since the KITTI dataset encompasses planar road scenarios, the planar road assumption for *Baseline_A* is enabled on the KITTI benchmark. For the other two benchmarks, the non-planar road assumption for *Baseline_A* is enabled. For *Baseline_B*, the parameter settings suggested by the authors in [120] have been adopted for the experiments.

The proposed algorithm and both of the baseline algorithms have been implemented on a PC platform Acer Veriton S670Gan, where the processor is Core 2 Duo E7600 3.06 GHz with 2 GB memory. All the codes are developed in C++ in the Visual Studio 2012 running in Windows 7.

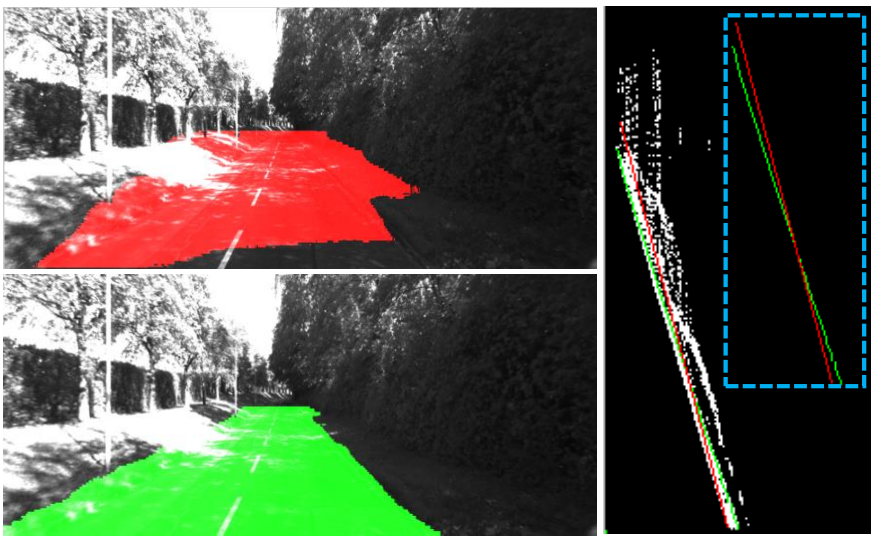


Fig. 3.9 Comparison of *Baseline_A* and the proposed method in a planar road scenario. The top-left image shows the result of *Baseline_A*, and the bottom left image shows the result of the proposed algorithm. The right image shows the corresponding extracted road profiles in the v-disparity image for *Baseline_A* (red) and proposed algorithm (green). Blue inset highlights the two road profiles for enhanced visualization. Note the v-disparity image has been enlarged for better visualization.

3.3.2 Accuracy Evaluation

The main problem in *Baseline_A* stems from the fact that it heavily relies on data fitting techniques, but it does not employ any countermeasure to deal with cases where the inputs to data fitting techniques are noisy. Not only roads but also obstacles will be projected as lines in the v-disparity image. Hence, line extraction techniques like Hough Transform will end up extracting multiple lines and this makes it difficult to identify the line (planar) or a family of lines (non-planar) that corresponds to the actual road profile. The advantage of the proposed algorithm over *Baseline_A* is clearly illustrated in Figure 3.9 and 3.10. In Figure 3.9, the two algorithms are compared for a planar road scenario. The effectiveness of using the Hough Transform to extract the road profile is clearly impeded by the projection of the bushes along the side of road. In Figure 3.10, the two algorithms are compared using a non-planar road scenario with an undulating hill. Since *Baseline_A* assumes that the road profile curvature is of constant sign, which is not true in this case, *Baseline_A* fails. It is noteworthy that the road profile extracted in Figure 3.10 by the proposed method is not a regular curve that can be modeled mathematically. This clearly demonstrates that the proposed algorithm is not restricted to specific road scenarios but is able to accurately extract varying road profile shapes. In addition, it is evident from Figure 3.10 that the proposed method is able to detect

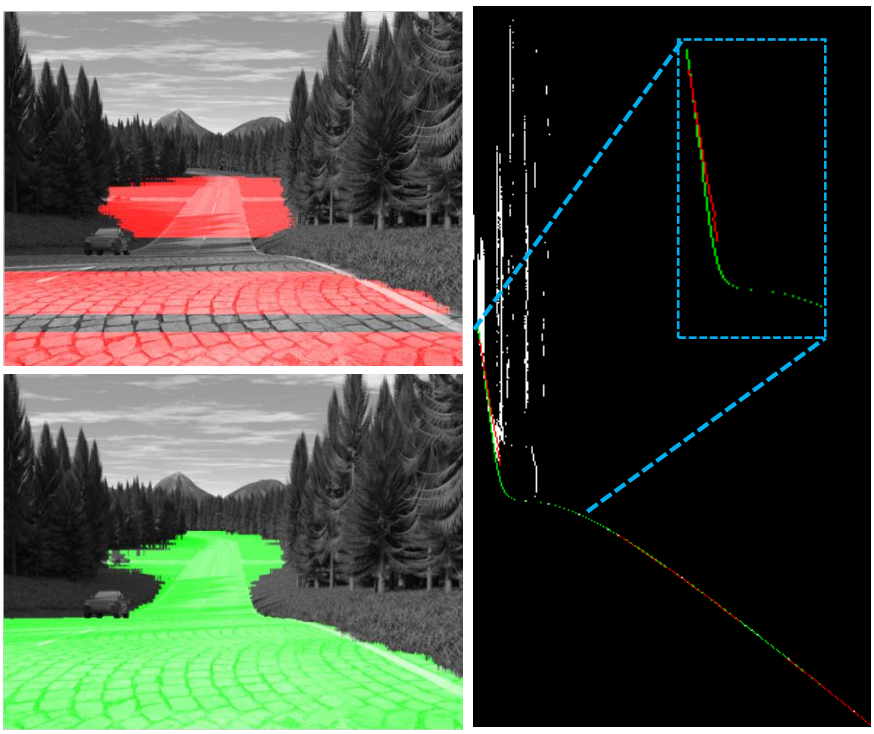
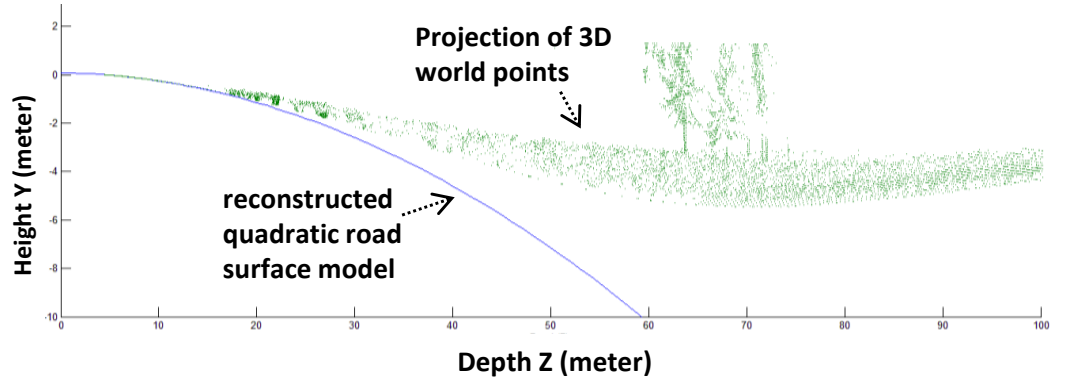


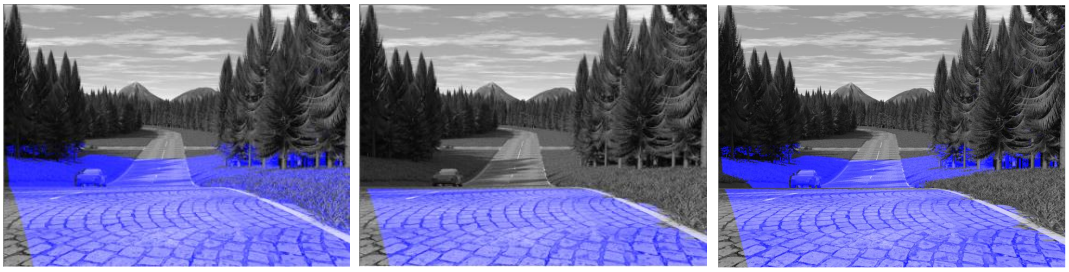
Fig. 3.10 Comparison of *Baseline_A* and the proposed method in a non-planar road scenario: The top-left image shows the result of *Baseline_A*, and the bottom left image shows the result of the proposed algorithm. The right image shows the corresponding extracted road profiles in the v-disparity image for *Baseline_A* (red) and the proposed algorithm (green). Blue inset highlights a portion of the two road profiles for better visualization. Note that two vehicles are present in the image. The proposed algorithm is also able to correctly distinguish the vehicle that is further away whereas *Baseline_A* fails. The v-disparity image has been enlarged for better visualization.

road at a large distance and correctly distinguish the vehicle that is very far away as non-road object.

Instead of working in the u-v-disparity space, *Baseline_B* works in the 3D digital elevation map (DEM) space. To distinguish road from non-road entities including obstacle and traffic isles, two classifiers are adopted. The density-based classifier marks DEM cells as road or obstacles based on the density of the reconstructed 3D points. For the road surface based classifier, the road surface is modeled such that quadratic variations of the height y with the horizontal displacements x and depths z are allowed. Then a combination of RANSAC, region growing and least-squares fitting are employed to compute the quadratic road surface. Based on the computed road surface model, road and non-road entities are discriminated. Fusion and error filtering is finally performed on the results of the two classifiers.



(a)



(b) (c) (d)

Fig. 3.11 *Baseline_B* is evaluated in a non-planar road scenario: (a) the quadratic road model reconstructed by *Baseline_B* only dovetails the realistic road surface in the vicinity of the vehicle. (b) Classification result of the density based classifier from *Baseline_B*. (c) Classification result of the road surface based classifier from *Baseline_B*. (d) Final classification result after fusing and filtering of (b) and (c) for *Baseline_B*.

Compared to the planar road model, quadratic road model is more capable in some situations where the road surface presents quadratic curvature. However, due to its restricted parameterization, quadratic road model can only model slope changes in one direction. Hence, it will fail if the road is undulating. Figure 3.11 presents the same road scenario as the one in Figure 3.10 where the road surface is in an undulating shape. As shown in Figure 3.11(a), the quadratic road model reconstructed by *Baseline_B* only dovetails the realistic road surface in the vicinity of the vehicle. From the range of about 25 meter, the reconstructed road model begins to fail.

Three-dimensional world points are reconstructed from the disparity space and reconstruction noise will be further introduced to the input data. As mentioned by the authors of *Baseline_B*, the reconstructed points' height uncertainty increases with depth and a 3D road point at a depth of 30 meter can have a height uncertainty of up to 17cm. This makes the road surface based classifier reliable only within a certain range. Due to this limitation, in the fusion and

Fig. 3.12 Examples of the detection results of the proposed algorithm for scenarios where the vicinity of the vehicle is filled with crowded objects.

error filtering step, the work in *Baseline_B* relies only on the results of the density based classification for points at depth greater than 30m.

The working principle for density based classification is that different scene patches have different density values in the DEM. A double thresholding technique is designed by taking into account only the uphill road surface. Although the density based classifier works fine with the planar and uphill road surface, it may fail for the case of downhill road surface. An obstacle standing on the downhill road may be misclassified as road depending on its height. As can be seen in Figure 3.11(b), the vehicle on the downhill part of the road is wrongly classified by the density based classifier. Figure 3.11(c) shows the result of the road surface based classifier. The final detection result is presented in Figure 3.11(d), which is the fusion of Figure 3.11(b) and Figure 3.11(c). It can be observed that the algorithm proposed in *Baseline_B* eventually wrongly classifies the vehicle as the road due to the intrinsic limitation of each classifier.

Crowded road scenario is a kind of situation that needs to be given high attention as they are usually encountered in daily traffic. Since the proposed algorithm is well designed to remove the large obstacle at the first step, it is able to deal with this situation well. Some detection results from the proposed algorithm in crowded road scenarios are shown in Figure 3.12.

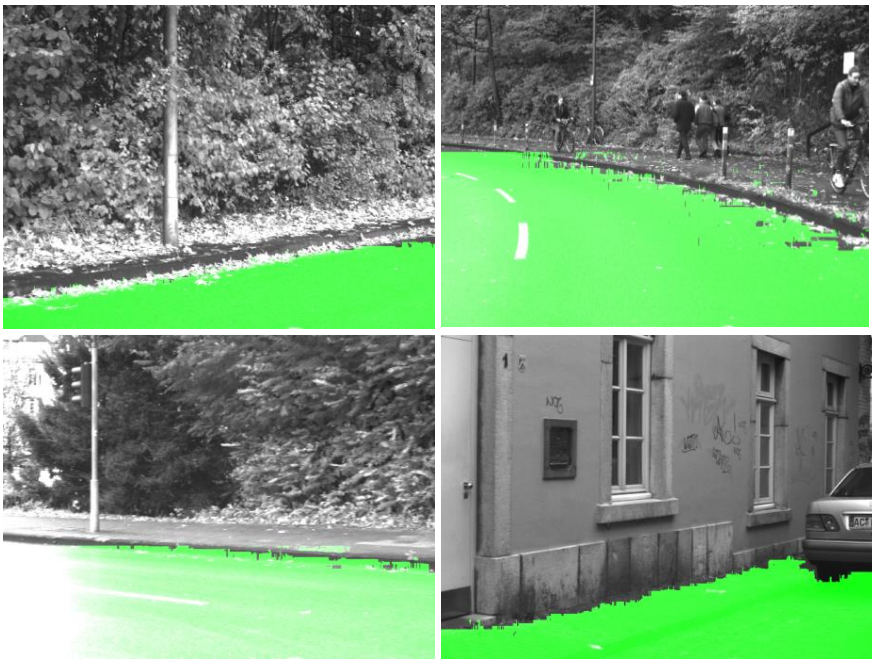


Fig. 3.13 Examples of the detection results of the proposed algorithm for scenarios where vehicle is turning. The corresponding yaw rate for the vehicle is 0.5256, -0.2144,-0.5625 and 0.3946. The unit of the yaw rate is degree/second.

Another special scenario is one where the vehicle is turning. The yaw rate, i.e. the angular velocity of the yaw rotation, depicts vehicle's turning degree. Although the yaw rate will not impact directly on the proposed method's computations since the proposed algorithm does not utilize the temporal information, the high yaw rate may lead to changes in the roll angle. This may violate the assumption that is made in section 3.1.1 that the cameras are installed such that the roll angle is negligible. Two factors help to lessen vehicle's turning effect for the proposed algorithm. First, the proposed algorithm is working on rectified images. The turning effect can be compensated by the image rectification algorithm to a large degree. Second, the proposed algorithm extracts the road surface in a controlled manner. The parameter *threshold_variance* helps to further reduce the turning effect. The Daimler dataset contains many cases where the vehicle is turning. Note that the dataset has been rectified. Some examples of the detection results from the proposed algorithm for scenarios where the ego-vehicle is turning at a high yaw rate are presented in Figure 3.13. The evaluations show that the proposed algorithm also works fine in these scenarios.

Finally, both qualitative and quantitative evaluation between the proposed and the two baseline algorithms are conducted. Note that the datasets used in this evaluation contain a lot of invalid regions whose disparity values are not available. For the image points

within these regions, the corresponding 3D coordinates are not reconstructed and therefore cannot be classified. To ensure a fair comparison, the interpolation technique for the invalid regions proposed in Section 3.2.2.4 has been also applied to *Baseline_B*. Besides Figure 3.9, 3.10, 3.11, more qualitative evaluation results are presented in Figure 3.14 and Figure 3.15 respectively. Note that the testing samples in Figure 3.14 and Figure 3.15 includes different scenarios such as planar road scenarios, up-hill, down-hill and undulating hill non-planar road scenarios, the scenario where the vicinity of the vehicle is filled with crowded objects and the scenario where the vehicle is turning. As explained earlier, *Baseline_A* is frail in these complex scenarios. *Baseline_B* works well with the KITTI dataset since the KITTI dataset mainly contains planar or up-hill road scenarios. In addition, the disparity maps used in this dataset are ground truth, hence the reconstructed 3D points present high accuracy. It is noteworthy that the traffic isles in this dataset are also correctly discriminated from the road by *Baseline_B* as shown in Figure 3.14. Since the enpeda dataset contains many frames where the road surface is undulating, *Baseline_B* can only detect the road surface correctly in the vicinity of the vehicle for this dataset. The Daimler dataset encompasses all the dynamic road shape including planar, up-hill, down-hill, and undulating hill road scenarios. In addition, the corresponding disparity maps are quite noisy and contain large invalid and erroneous regions. The data noisy will be further amplified during the process of 3D reconstruction from disparity space. In order to distinguish the road surface from traffic isles, the classification threshold for road in *Baseline_B* is set conservatively. These three factors contribute to large amount of false detection in *Baseline_B*. The visual comparison in Figure 3.14 and Figure 3.15 clearly demonstrates the superiority of the proposed algorithm over the baseline algorithms in various challenging road scenarios.

For the quantitative evaluation, the evaluation framework in [114] is adopted. Manual road labeling is performed on all the left images in the enpeda, KITTI datasets, and a subset of the Daimler dataset containing 1613 frames to generate the ground truth. The Daimler dataset consists of a 27-minute sequence of video. The frames within the 10^{th} and 20^{th} minute period of the video sequence have been chosen for manual labeling. The 10^{th} minute of the Daimler sequence depicts a situation where the vehicle is moving on a planar road with crowded obstacles in the vicinity of the vehicle. The 20^{th} minute Daimler sequence is a situation which encompasses all the dynamic road situations like planar, up-hill, down-hill and undulating hill road scenarios. In addition, many cases where the vehicle is turning are present in this sub-sequence. Based on the ground truth and the detection results, each pixel in the test samples is labeled as one of the four cases: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) and *False Negative* (FN) according to Table 3.1. Then four metrics are

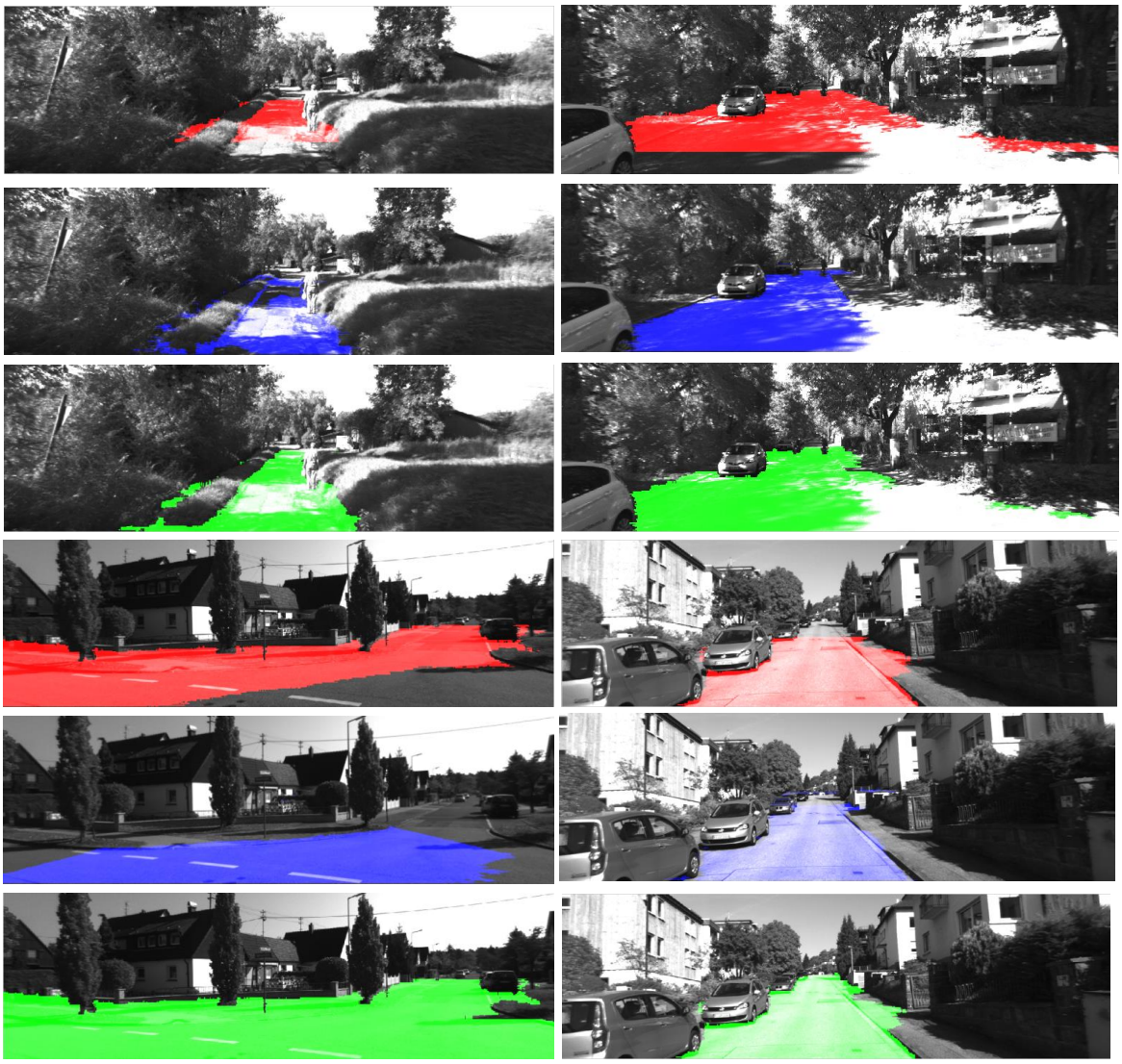


Fig. 3.14 More comparisons between *Baseline_A* (red), *Baseline_B* (blue) and the proposed algorithm (green) for KITTI dataset.

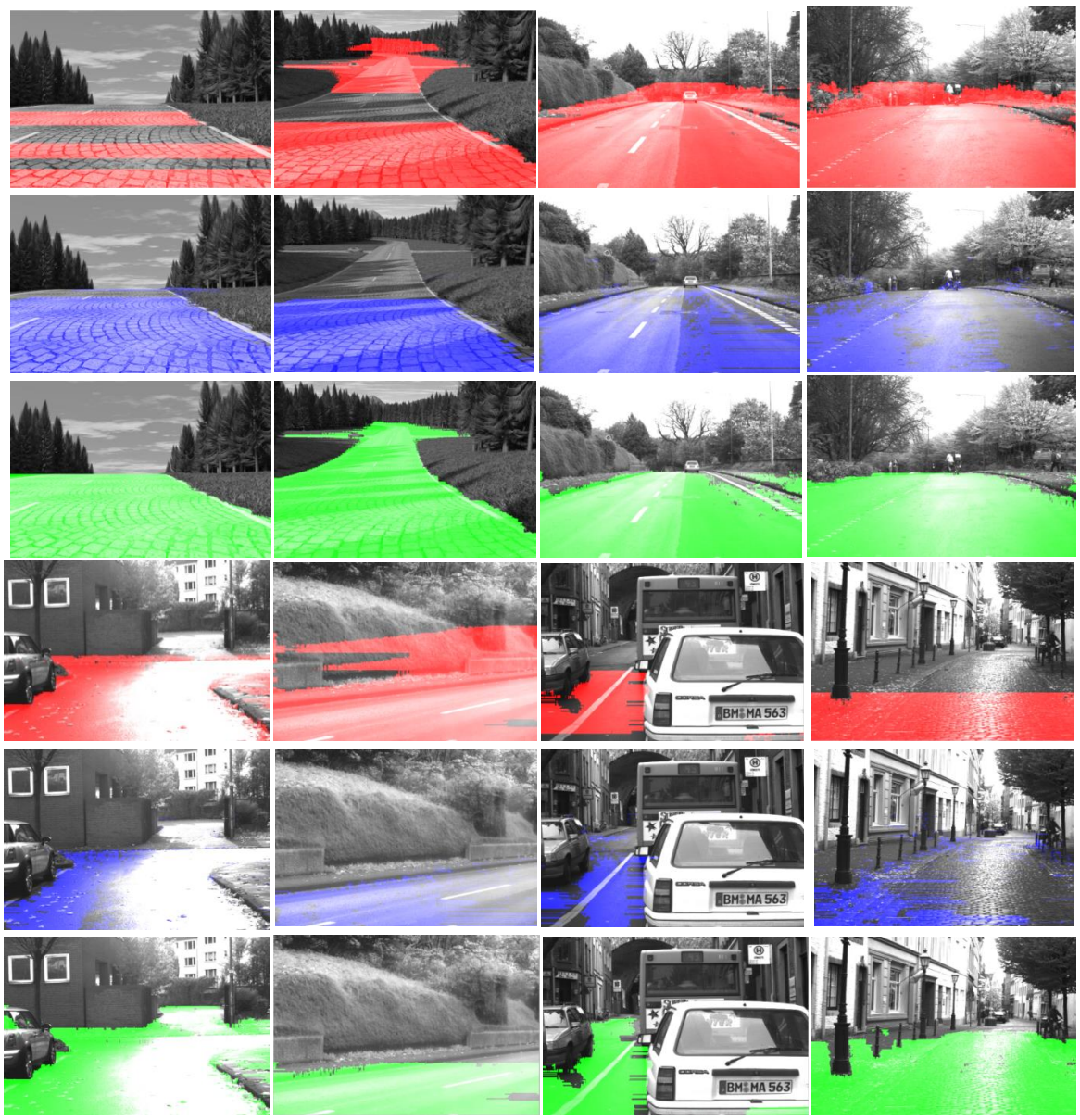


Fig. 3.15 More comparisons between *Baseline_A* (red), *Baseline_B* (blue) and proposed algorithm (green) for enpeda and Daimler datasets. The images within the first two columns in the higher part are from the enpeda dataset while the others are from the Daimler dataset.

Table 3.1 Contingency table

		Ground Truth	
		Non-Road	**Road**
Estimated Result	**Non-Road**	TN	FN
	Road	FP	TP

adopted to describe the detection performance: *Quality*, *Detection Rate*, *Detection Accuracy* and *Effectiveness* as formulated in Table 3.2. Each of the metric provides a different insight. For the detailed interpretation of these metrics, please refer to [114].

The quantitative evaluation results have been summarized in Table 3.3. From Table 3.3, it can be observed that the proposed algorithm outperforms *Baseline_A* for all the metrics on all the test cases. The proposed algorithm also achieves better results than *Baseline_B* for all the metrics except the detection rate on the enpeda and KITTI datasets. The proposed algorithm obtains a slightly lower detection rate than *Baseline_B* on these two datasets due to the following reasons: for the enpeda dataset, the proposed method misclassifies the grass field at a distance of more than 150 meter as road surface. While for the KITTI dataset, some of the traffic isles whose height is quite close to the road surface are misclassified by the proposed algorithm as road surface. The quantitative evaluation results are consistent with the qualitatively results.

It is noteworthy that although the input disparity maps for the KITTI dataset only achieve semi-dense density (approximately 50%) and the input disparity maps for the Daimler dataset are noisy and contain large invalid and erroneous regions, the proposed algorithm can still achieve high detection performance. This clearly demonstrates that the proposed algorithm is robust to noisy input. For the enpeda dataset, the major contributing factor that prevents the detection performance of the proposed algorithm from achieving 100% performance lies in the existence of the grass field present from the 71^{th} frame to the 200^{th} frame in the dataset. An example of this is illustrated in Figure 3.10. As can be seen, the grass field extends far away from the camera (more than 150 meter) and the difference between the heights of the grass field and the road surface is small. Therefore, the observed disparity values for the grass field and the disparity values for the road surface nearby are almost the same. Due to this reason, the proposed algorithm classifies these grass fields as the road surface.

Table 3.2 Four pixel-wise metrics for road surface detection accuracy evaluation

Pixel-wise Metric	**Definition**
Quality	$\frac{TP}{TP+FP+FN}$
Detection Rate	$\frac{TP}{TP+FP}$
Detection Accuracy	$\frac{TP}{TP+FN}$
Effectiveness	$\frac{2PR}{P+R}$

3.3.3 Runtime Performance Evaluation

The proposed algorithm does not require complex calculations, that is, only integer addition and comparison operations are needed. The run-time bottleneck of *Baseline_A* mainly lies in the Hough Transform employed to extract the lines in the v-disparity image.

Baseline_B presents a much higher computational complexity. Dense 3D points need to be reconstructed. Canny edge detector and RANSAC based plane fitting are employed for initial surface fitting. The biggest bottleneck for *Baseline_B* lies in the uncertainty model-driven surface growing period. During this period, the quadratic surface is recomputed, in a least-squares fashion, about N times per frame on average. The value of N depends on the road surface types, the quality and number of the input points and the selection of cells for initial surface fitting. N can be up to 200 as reported in *Baseline_B*. For our experiments conducted on the three adopted datasets, it is observed that N can be up to 300. Many intensive computations with double precision data type are required.

In order to make a fair comparison, the highly optimized implementation of the Hough Transform in OpenCV for the implementation of *Baseline_A* and the Canny edge detector for *Baseline_B* have been employed. In addition, for our implementation of *Baseline_B*, the expected road density maps are computed offline for each dataset and not included into the running time measurement. During the stage of region growing, the matrix used to compute the quadratic surface model defined by Equation (5) in *Baseline_B* is updated using only the newly added DEM cells for each iteration. Compared to the experiment setup in *Baseline_B*, the image size is larger in our experiments. In addition, the 3D points are reconstructed in software instead of using specific hardware.

Table 3.3 Detection accuracy comparison between the baseline and the proposed road surface detection algorithms.

		Quality(%)	**Detection Rate**(%)	**Detection Accuracy**(%)	**Effectiveness**(%)
enpeda	*Baseline_A*	64.17	93.66	67.08	78.17
	Baseline_B	81.33	98.70	82.21	89.71
	Proposed	88.34	95.62	92.07	93.81
	Improvement_A	24.17	1.96	24.99	15.64
	Improvement_B	7.01	-3.08	9.86	4.10
KITTI	*Baseline_A*	79.25	84.92	92.23	88.42
	Baseline_B	78.24	93.15	83.02	87.79
	Proposed	82.09	86.32	94.36	90.16
	Improvement_A	2.84	1.40	2.13	1.74
	Improvement_B	3.85	-6.83	11.34	2.37
Daimler	*Baseline_A*	80.72	90.40	88.29	89.33
	Baseline_B	63.22	89.32	68.39	77.46
	Proposed	85.16	92.47	91.51	91.99
	Improvement_A	4.44	2.07	3.22	2.66
	Improvement_B	21.94	3.15	23.12	14.53

The results in Table 3.4 show that the proposed algorithm easily achieves real-time performance (less than 0.015s per frame). Therefore, the proposed algorithm is highly suitable for in-vehicle deployments which often have limited computation resources. In addition, the proposed algorithm is about 35% faster than *Baseline_A* and about 94.13% faster than *Baseline_B* on average. Note the computation time does not include the time of generating the disparity map for both the baseline algorithms and the proposed algorithm.

3.4 Summary

Road surface detection is usually required as an initial step in many applications (e.g. autonomous navigation, object detection and tracking) to provide the geometrical constraint for the subsequent step. In this chapter, a simple but efficient non-parametric depth based road surface detection algorithm that is inspired by four intrinsic road attributes observed under stereo geometry is proposed. Unlike existing methods that rely on rigid mathematical models, the proposed non-parametric algorithm has been shown to be capable of tackling highly dynamic road scenarios. This has paved the way for overcoming the limitations of existing parametric methods that cannot cope with cases where the road profile doesn't

Table 3.4 Runtime performance comparison between the baseline and proposed road surface detection approaches.

		enpeda	**KITTI**	**Daimler**
Baseline_A (Seconds)	Total	4.423	2.640	236.168
	Per Frame	0.011	0.014	0.011
Baseline_B (Seconds)	Total	41.429	41.121	2187.59
	Per Frame	0.105	0.212	0.100
Proposed (Seconds)	Total	2.749	2.054	131.09
	Per Frame	0.007	0.011	0.006
Speedup_A(%)		37.85	22.20	44.50
Speedup_B(%)		93.36	95.00	94.01

fit the pre-defined model or when the constantly varying road profiles cannot be modeled mathematically. Extensive experimental results using three challenging datasets (i.e. enpeda, KITTI, and Daimler) show that the proposed algorithm outperforms the baseline algorithms both in terms of detection accuracy and runtime performance. The data set used to evaluate the proposed technique includes different scenarios such as various planar road scenarios, up-hill, down-hill and undulating hill non-planar road scenarios, the scenarios where the vicinity of the vehicle is filled with crowded objects and scenarios involving turning vehicles. The experimental results show that the proposed algorithm achieves both high detection and runtime performance, and hence it is well suited for deployment in a wide range of applications involving collision avoidance..

The knowledge of the ego-vehicle's motion state relative to the road serves as the foundation for the risk reasoning in collision avoidance systems. In the next chapter, a robust and efficient visual odometry technique is proposed.

CHAPTER 4

ROBUST AND LOW-COMPLEXITY VISUAL ODOMETRY

Knowledge of the ego-vehicle's motion state is essential for assessing the collision risk between the ego-vehicle and the obstacles present in the driving environment. A comprehensive review of existing visual odometry methods has been conducted in Section 2.3.4. The existing solutions fail to achieve a good balance between high accuracy and low computational complexity [26].

In this chapter, a framework for robust and runtime-efficient visual odometry is proposed. The proposed framework integrates runtime-efficient strategies with robust techniques at each of the core stages in visual odometry. Specifically, a pruning method is employed to reduce the computational complexity of KLT feature detection without compromising on the quality of detected features. Ego-motion prior is leveraged on to determine a better initial position for KLT tracking process to increase the chance for correct convergence, which significantly increases the proposed technique's robustness in challenging environments. The robustness of the proposed technique is further enhanced by adopting an automatic tracking failure detection scheme during feature tracking. In addition, an adaptive and small integration window for each feature is set during tracking based on its distance from the ego-vehicle. This significantly reduces the computational complexity. Finally, an early

RANSAC termination condition is applied in the Gaussian-Newton based motion estimation process to further increase the algorithmic robustness and reduce the algorithmic computation time. Experimental results based on the KITTI dataset show that the proposed technique outperforms state-of-the-art visual odometry methods by producing more accurate ego-motion estimation in notably shorter amount of time. Part of the works presented in this chapter have been published in [7, 8]. In addition, a paper based on the work in this chapter has been submitted to IEEE Transactions on Intelligent Transportation Systems and is currently under the second round of revision [9].

This chapter is organized as follows: Section 4.1 formulates visual odometry as a mathematical minimization problem. The proposed visual odometry algorithm is presented in Section 4.2. A comprehensive evaluation of the proposed visual odometry method with existing state-of-the-art methods using the well-known KITTI odometry dataset is presented in Section 4.3 and Section 4.4 summarizes this chapter.

4.1 Problem Formulation

A camera installed on a moving vehicle is subjected to six degrees of freedom (DOF). That is, it can be translated in three perpendicular *X-Y-Z* axes, denoted as (tx, ty, tz) (in meter), and rotated about the three axes, denoted as (rx, ry, rz) (in radian). The goal of visual odometry is to obtain the value of $\boldsymbol{e} = (tx, ty, tz, rx, ry, rz)^T$ at each discrete time instance.

As shown in Eq. 4.1, the motion of a camera installed on the moving vehicle from the previous frame I_{n-1} to current frame I_n can be represented by the matrix $\boldsymbol{M}_n \in \mathbb{R}^{4*4}$ [314]:

$$\boldsymbol{M}_n = \begin{bmatrix} \boldsymbol{RO}_n & \boldsymbol{tr}_n \\ \boldsymbol{0} & \boldsymbol{1} \end{bmatrix} \tag{4.1}$$

$$\boldsymbol{RO}_n = \begin{bmatrix} cy*cz & -cy*sz & sy \\ sx*sy*cz+cx*sz & -sx*sy*sz+cx*cz & -sx*cy \\ -cx*sy*cz+sx*sz & cx*sy*sz+sx*cz & cx*cy \end{bmatrix} \tag{4.2}$$

$$\boldsymbol{tr}_n = [tx \quad ty \quad tz]^T \tag{4.3}$$

$$\begin{aligned} sx &= sin(rx); cx = cos(rx) \\ sy &= sin(ry); cy = cos(ry) \\ sz &= sin(rz); cz = cos(rz) \end{aligned} \tag{4.4}$$

As shown in Eq. 4.5, the camera pose $\boldsymbol{C}_n$ from the time of initialization can be obtained by concatenating all the transformations $\{\boldsymbol{M}_i | i = 1, \cdots, n\}$ [253]. $\boldsymbol{T}_n$ in Eq. 4.6 represents the scene motion, which is the inverse of the camera motion.

$$\boldsymbol{C}_n = \prod_{i=1}^{n} \boldsymbol{M}_i \tag{4.5}$$

$$\boldsymbol{T}_n = \boldsymbol{M}_n^{-1} = \begin{bmatrix} \boldsymbol{R}_n & \boldsymbol{t}_n \\ \boldsymbol{0} & \boldsymbol{1} \end{bmatrix} \tag{4.6}$$

Assume that the set of static points (features) in Euclidean space observed in frame I_{n-1} are $\{\boldsymbol{p}_{n-1}^i = (x_{n-1}^i, y_{n-1}^i, z_{n-1}^i)^T | i = 1, 2, \ldots, k\}$ and their correspondences in frame I_n are $\{\boldsymbol{p}_n^i = (x_n^i, y_n^i, z_n^i)^T | i = 1, 2, \ldots, k\}$. Then the relationship between a pair of correspondences $\boldsymbol{p}_{n-1}^i$ and $\boldsymbol{p}_n^i$ through $\boldsymbol{T}_n$ is shown in Eq. 4.7:

$$\boldsymbol{p}_n^i = \boldsymbol{R}_n * \boldsymbol{p}_{n-1}^i + \boldsymbol{t}_n \tag{4.7}$$

Therefore the ego-motion parameters $\boldsymbol{e}_n = (tx, ty, tz, rx, ry, rz)^T$ can be found by minimizing the residual function in Eq. 4.8, where w_i is the weighting factor that denotes the contribution of point i to the least square solution:

$$E = \arg\min_{\{\boldsymbol{e}_n\}} \sum_{i=1}^{k} w_i \| \boldsymbol{p}_n^i - \boldsymbol{R}_n * \boldsymbol{p}_{n-1}^i - \boldsymbol{t}_n \|^2 \tag{4.8}$$

Eq. 4.8 calculates the residual in Euclidean space. However, as discussed in [254, 315], stereo triangulation error can be highly anisotropic and correlated. As such, a recommended

approach is to compute the residual in image space, where the noise level is similar for all components of the measurement vector:

$$E = \arg\min_{\{\boldsymbol{e}_n\}} \sum_{i=1}^{k} w_i \|\boldsymbol{m}_n^i - \eta(\boldsymbol{R}_n * \mathfrak{g}(\boldsymbol{m}_{n-1}^i) + \boldsymbol{t}_n)\|^2 \tag{4.9}$$

Where $\boldsymbol{m}_n^i = (u_n^i, v_n^i, d_n^i)^T$ is the projection of $\boldsymbol{p}_n^i$ in the image frame I_n. $\mathfrak{g}$ is the triangulation equation, while $\eta = \mathfrak{g}^{-1}$ is the projection function. $baseline$ and $focal$ are the corresponding baseline and focus length, (u_0, v_0) is the principal point as discussed in Section 3.1.1.

$$\boldsymbol{p}_n^i = \mathfrak{g}(\boldsymbol{m}_n^i) = \begin{cases} x_n^i = (u_n^i - u_0) * baseline / d_n^i \\ y_n^i = (v_n^i - v_0) * baseline / d_n^i \\ z_n^i = focal * baseline / d_n^i \end{cases} \tag{4.10}$$

$$\boldsymbol{m}_n^i = \eta(\boldsymbol{p}_n^i) = \begin{cases} u_n^i = focal * x_n^i / z_n^i + u_0 \\ v_n^i = focal * y_n^i / z_n^i + v_0 \\ d_n^i = focal * baseline / z_n^i \end{cases} \tag{4.11}$$

The aim of feature detection is to identify a set of points $\{\boldsymbol{m}_{n-1}\}$ in frame I_{n-1}, while the aim of feature tracking is to identify $\{\boldsymbol{m}_n\}$, which are the correspondences of $\{\boldsymbol{m}_{n-1}\}$ in frame I_n. Motion estimation computes the ego-motion parameters $\boldsymbol{e}_n = (tx, ty, tz, rx, ry, rz)^T$ by solving Eq. 4.9.

4.2 Proposed Algorithm

As highlighted in [223], the extraction of reliable feature correspondences that correspond to the static scene plays an essential role in the success of visual odometry. The KLT feature tracker [214], which consists of corner feature detection and tracking, is a widely accepted method for feature correspondence extraction [42, 43, 181, 202, 253]. Although KLT has been shown to be one of the best feature tracker, direct adoption of KLT can lead to inaccurate tracking results in highly complex urban environments as will be shown in the following discussion and experimental results. In addition, KLT is time consuming [316]. On the other hand, the extracted feature correspondences may come from self-moving objects. Direct motion estimation based on feature correspondences that are contaminated with self-moving or inaccurately tracked features will lead to inaccurate results.

The proposed technique aims to overcome the limitations of existing solutions by integrating strategies to achieve robust visual odometry at low computational complexity at various core stages of the ego-motion estimation framework. Figure 4.1 shows the top-level block diagram of the proposed visual odometry framework. Taking the image sequence from the stereo rig and the corresponding disparity map as the input, the proposed visual odometry framework consists of the following two stages: 1) Feature correspondences setup, and 2) Motion estimation.

The first stage incorporates techniques for robust and low-complexity feature correspondences setup. This stage further consists of the following two steps: *(i) Low-complexity corner detection with pruning, and (ii) Robust and low-complexity feature tracking using improved KLT tracker.* For each time step n, corner detection is applied on the previous left image I_{n-1} to extract a set of corner features $\{\boldsymbol{m}_{n-1}\}$ using a pruning technique. The computational complexity of corner detection is significantly reduced due to the pruning process without compromising on the quality of the extracted corner features. Next, for each feature $\boldsymbol{m}^i_{n-1}$ in $\{\boldsymbol{m}_{n-1}\}$, its correspondence $\boldsymbol{m}^i_n$ in current left image I_n is identified using an improved KLT tracker. Smooth motion constraint is utilized to determine a better starting point for KLT tracking process, which leads to fast and accurate convergence during the tracking. In addition, an adaptive window technique, which is based on the distance of the feature from the ego-vehicle and the smooth motion constraint, is employed to track each feature. This significantly reduces the runtime complexity. Finally, an automatic tracking failure detection scheme is adopted during feature tracking to further increase the robustness of the method.

The second stage incorporates techniques for robust and fast motion estimation. Given the set of feature correspondences $\{\boldsymbol{m}_{n-1}\}$ and $\{\boldsymbol{m}_n\}$, the motion parameters $\boldsymbol{e}_n = (tx, ty, tz, rx, ry, rz)^T$ are computed by solving the function formulated in Eq. 4.9 using Gaussian-Newton method. In order to increase the robustness and also decrease the computation time, RANSAC with an early termination condition is enabled to remove the outliers that do not exhibit coherent movement.

In the following sub-sections, detailed descriptions of each stage for the proposed framework are provided.

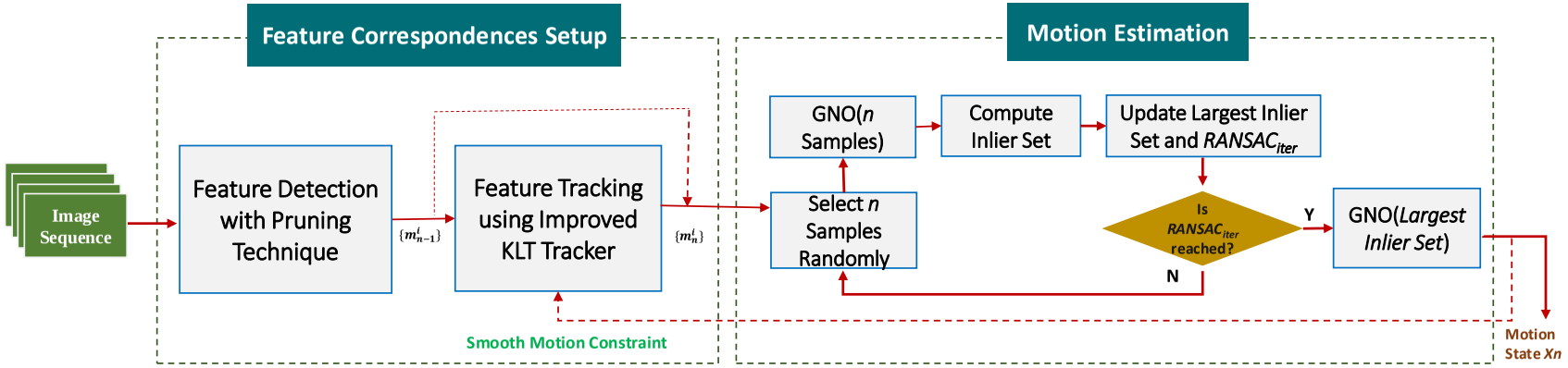


Fig. 4.1 Top-level block diagram of the proposed visual odometry framework: The proposed visual odometry framework consists of the following two stages: 1) Feature correspondences setup, and 2) Motion estimation. The first stage incorporates techniques for robust and low-complexity feature correspondences setup. This stage further consists of the following two steps: (i) Low-complexity corner detection with pruning, and (ii) Robust and low-complexity feature tracking using improved KLT tracker. The second stage incorporates techniques for robust and fast motion estimation. Given the set of feature correspondences $\{\boldsymbol{m}_{n-1}\}$ and $\{\boldsymbol{m}_n\}$, the motion parameters $\boldsymbol{e}_n = (tx, ty, tz, rx, ry, rz)^T$ are computed by solving the function formulated in Eq. 4.9 using Gaussian-Newton method. In order to increase the robustness and also decrease the computation time, RANSAC with an early termination condition is enabled to remove the outliers that do not exhibit coherent movement.

4.2.1 Low Complexity Corner Detection with Pruning

In order to detect corners, KLT computes a corner response λ_2 for each pixel:

$$\lambda_2 = \frac{(a+c) - \sqrt{(a-c)^2 + 4b^2}}{2} \tag{4.12}$$

λ_2 corresponds to the minimum eigen-value of the matrix $\boldsymbol{D}$, which approximates a local auto-correlation function:

$$\boldsymbol{D} = \begin{bmatrix} \sum wI_x^2 & \sum wI_xI_y \\ \sum wI_xI_y & \sum wI_y^2 \end{bmatrix} = \begin{bmatrix} a & b \\ b & c \end{bmatrix} \tag{4.13}$$

Where I_x and I_y are the horizontal and vertical gradients respectively, and w is the weight function, which can be a simple box window or Gaussian window. The eigen-values λ_1 and λ_2 of $\boldsymbol{D}$ (where $\lambda_1 >= \lambda_2$) represent the two dominant directions of intensity change.

A threshold is applied on the corner response λ_2 to remove the obvious non-corners. The rest of the pixels are then ranked in descending order of their corner response and the pixels with the highest corner response are selected as corners after applying non-maximal suppression.

In order to identify good features, KLT computes the complex corner measure λ_2 for each pixel and chooses the ones with high λ_2 value. However, the obvious non-corners, i.e. the smooth and low curvature regions, constitute a large majority of the image in most cases. This incurs a lot of computational redundancies when the complex corner measure for the obvious non-corners are computed. As such, a pruning method to select only the most relevant features for tracking is employed. The pruning method is explained as follows.

Expanding Eq. 4.12, we obtain:

$$\begin{aligned} \lambda_2 &= \frac{(a+c) - \sqrt{(a-c)^2 + 4b^2}}{2} \\ &= \frac{(a+c) - \sqrt{(a+c)^2 - 4(ac-b^2)}}{2} \end{aligned} \tag{4.14}$$

It can be observed that λ_2 is heavily influenced by the term $(ac-b^2)$ as the two $(a+c)$ terms get cancelled out. Hence the goal of identifying pixels with large λ_2 can be simplified as one that identifies pixels with large $(ac-b^2)$. In addition, in order to maximize $(ac-b^2)$, the

first term ac should be large. In other words, pixels that have small ac values are less likely to be good features.

Based on the analysis above, when applying an appropriate threshold to discard pixels with low ac values, the remaining pixels contain the final good KLT corners. Figure 4.2(b) shows the corner candidates selected by applying threshold = 0.05 * max(ac).

In addition, the I_x^2 and I_y^2 terms in the a and c value can be approximated with the absolute values of I_x and I_y respectively as follows:

$$a' = \sum |I_x|; \quad c' = \sum |I_y| \tag{4.15}$$

This eliminates the multiplication operations involved in the squared gradients. As such, pixels that have high $a'c'$ values will also have high ac values and therefore are highly likely to be good KLT corners. Figure 4.2(c) shows that the $a'c'$ map does not lose the distinctive corner regions. In addition, as shown in Figure 4.3, when the threshold for $a'c'$ map is reduced, distinctive corner and edge features are released before texture and flat regions. Hence, $a'c'$ can be used as a corner indicator measure as it can effectively distinguish the corner regions from the non-corner regions.

Therefore, as illustrated in Listing 4.1, instead of computing the complex corner measure as formulated in Eq. 4.12 for every pixel, a much simpler corner candidate indicator $a'c'$ as formulated in Eq. 4.15 is utilized as a pruning measure to remove the non-corner regions quickly and generate a small set of corner candidates. The final KLT corner measure is computed only on the small set of corner candidates and corners with highest measure value are chosen. By doing this, the computational complexity for corner detection is reduced and the quality of the extracted features is not compromised. Readers are referred to [8] for a detailed discussion.

We would like to point out that compared to the Sobel edge key points which are calculated based on the sobel response over only a single point, the pruning metric $a'c'$ is the product of the value of a' and c' that are calculated based on the sum of sobel response over their respective neighborhood window. A point that has a high sobel response may not necessarily have a large $a'c'$ value and vice versa. The pruning metric $a'c'$ has a tighter association with the conventional KLT corner measure and it ensures that corners are detected in the same order as the conventional KLT corner detection method, that is, corners with higher KLT corner quality are detected earlier than those that are of lower corner quality. As such the

Fig. 4.2 Illustration of corner detection using different metrics: (a) Original image; (b) Corner candidates selected using ac at $threshold$=0.05*max(ac); (c) Corner candidates selected using $a'c'$ at $threshold$=0.05*max($a'c'$); (d) Corner regions with $\lambda_2 > 0.05$*max(λ_2).

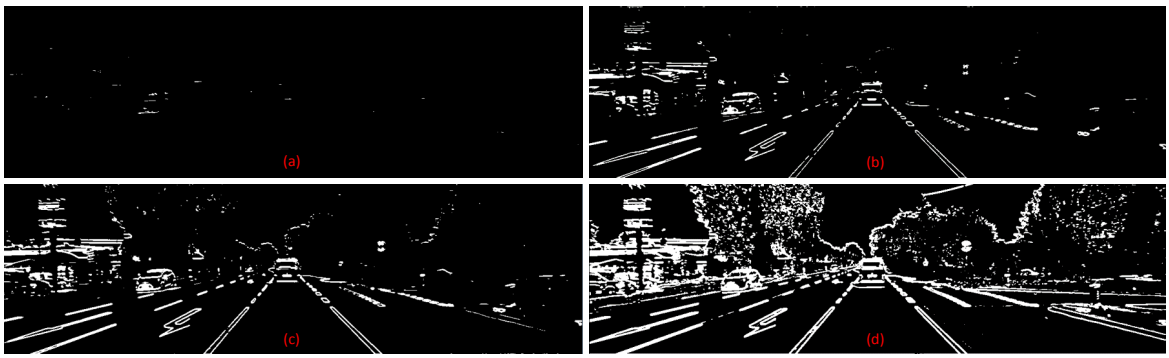


Fig. 4.3 $a'c'$ map at various thresholds: (a) 0.5; (b) 0.1; (c) 0.05; (d) 0.01.

proposed pruning technique enables rapid corner detection without losing distinctive KLT corners.

4.2.2 Feature Tracking using Improved KLT Tracker

Mathematically, the KLT tracking process is formulated as a least square problem to minimize a residual function over an integration window as defined in Eq. 4.16:

$$E = \arg \min_{\{\Delta u, \Delta v\}} \sum_{u=u_x-r}^{u_x+r} \sum_{v=v_y-r}^{v_y+r} \left(I_{n-1}(u,v) - I_n(u+\Delta u, v+\Delta v)\right)^2 \tag{4.16}$$

Where $I_{n-1}(u_x, v_y)$ and $I_n(u_x+\Delta u, v_y+\Delta v)$ are the correspondence located in image I_{n-1} and I_n respectively. $(\Delta u, \Delta v)^T$ is the optical flow for feature $I_{n-1}(u_x, v_y)$. r is the radius of the integration window.

Listing 4.1 Pruning based Corner Detector

Input: Image I, feature quality threshold t
Output: A set of corners $\{\boldsymbol{m}_i\}$ in image I
/* Pruning */
Compute horizontal and vertical gradient image I_x, I_y;
Compute $|I_x|$, $|I_y|$ for each pixel in I;
Compute $a'c' = \sum |I_x| * |I_y|$ for each pixel in I;
Threshold $a'c'$ map with *threshold* = t * max($a'c'$) to obtain corner candidate set $\mathscr{O}$;
/* Corner Response Function */
for each pixel in $\mathscr{O}$ **do**
    Compute $a = \sum I_x^2$, $c = \sum I_y^2$, $b = \sum I_x I_y$;
    Compute corner response λ_2;
end for
Threshold $\mathscr{O}$ with *threshold* = t * max(λ_2) to obtain candidate set $\mathscr{P}$;
Sort $\mathscr{P}$ in descending order of λ_2;
Apply non-maximal suppression to obtain $\{\boldsymbol{m}_i\}$.

The way KLT solves Eq. 4.16 using an iterative Newton Raphson method consists of a sequence of search operations that try to find a image patch with size $[2r+1, 2r+1]$ in image I_n such that there is minimum intensity difference between it and the image patch in image I_{n-1} of size $[2r+1, 2r+1]$ with feature $I_{n-1}(u_x, v_y)$ in the center. As observed in [317], starting the search process in the position (u_x, v_y) in image I_n, a small integration window size is preferred to increase the accuracy by avoiding smoothing out the image details. However, the integration window must also be sufficiently large to cater to the displacement of feature that undergoes large motion to increase robustness. In order to obtain a good tradeoff between local accuracy and robustness when choosing the integration window size, the pyramidal implementation of KLT has been introduced in [317]. However, this approach is time-consuming as tracking needs to be performed at different levels of the pyramid. In addition, it will be showed in the following that the pyramidal implementation of KLT can still lead to inaccurate results in highly complex urban driving environment.

In order to evaluate the accuracy of the conventional KLT in complex urban driving environment, an experiment is conducted based on the KITTI's stereo/flow benchmark [26], which provides 194 training images with ground truth flow fields and disparity maps. For each consecutive pair of images, up to 500 good features are extracted and tracked using OpenCV's implementation of KLT. The computed optical flows are compared with the ground truth. Figure 4.4 illustrates the distribution of the estimated flow error for all features in the order of the descending corner measure for one of the image pairs. It can be observed that the

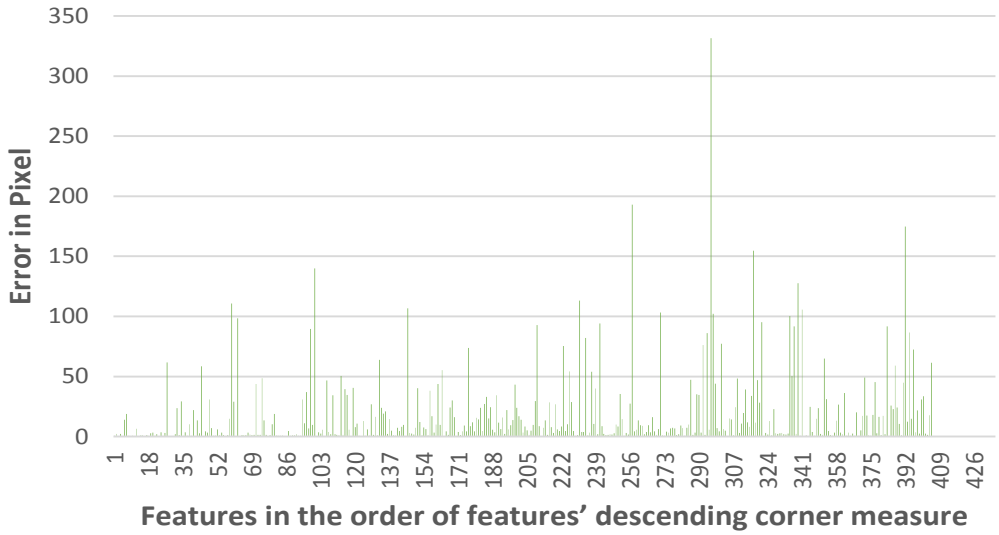


Fig. 4.4 Error distribution of optical flows estimated using conventional KLT algorithm.

conventional KLT results in high tracking error even for the features with high quality. The tracking error becomes more prominent when the feature quality decreases. This indicates that the conventional KLT for feature tracking is highly susceptible to noise.

4.2.2.1 Smooth Motion Constraint

In the conventional KLT tracker, the optimization process as indicated in Eq. 4.16 starts the search of a feature in the current frame at its same position in the previous frame. This can easily lead to KLT tracking failure if the starting point is too far from the convergence region. Such cases are common in scenarios where ego-mtion is large or when the features are not distinctive enough from their surroundings. Such a scenario will be explained with the help of Figure 4.5. Assume that feature *A* has been detected in the previous frame. Its ground truth correspondence in the current image is *A1* but the conventional KLT tracker results in *A2*. The reason that the KLT tracker fails in detecting the correct correspondence is due to the fact that it starts the search for *A*'s correspondence at *A3* (the same position as *A* in the previous image). Since *A3* is closer to *A2* and the local patches around *A2* and *A* are largely similar, the KLT algorithm converges to *A2* and terminates the search. This demonstrates that the initial position for the correspondence search significantly affects the accuracy of KLT.

The authors in [316] also observed the importance of setting a proper starting point for KLT tracker. To ensure that the starting point falls as close as possible to the convergence point, the work in [316] relies on inertial sensor that is attached to the camera. However, this requires additional effort for sensor calibration and synchronization. Unlike [316], the

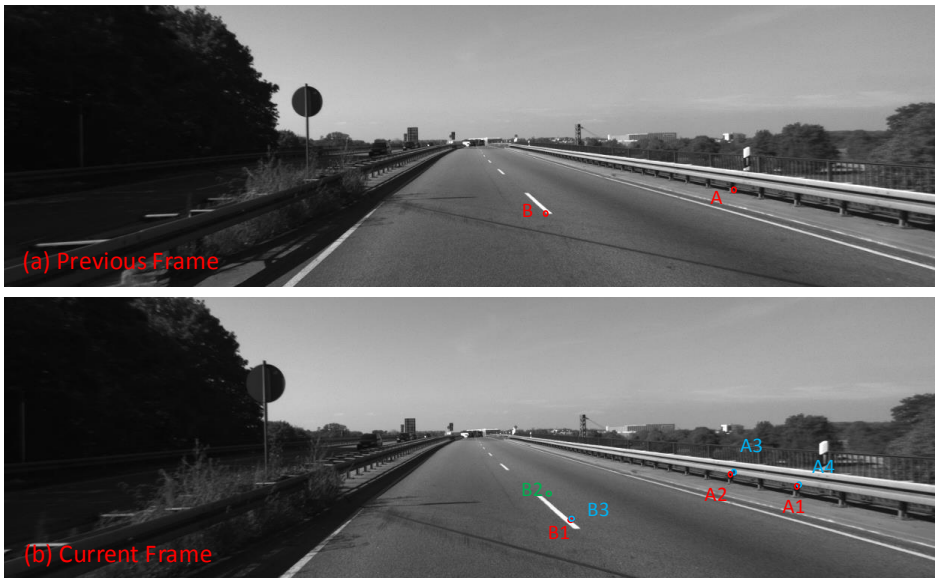


Fig. 4.5 An example of road scenario: Features that are detected in previous frame (a) are tracked in current frame (b).

proposed method determines a better starting point for KLT with the aid of the ego-motion estimated in previous step. In general, when the frame rate is high enough, a smooth motion pattern is presented between consecutive frames [253]. That is, the motion at time n is highly likely to be similar to the immediate previous motion at time *n-1*. Such phenomenon is referred to as Smooth Motion Constraint (SMC) [253].

Let $\mathbf{M}_{n-1}$ denotes the motion estimated from frame I_{n-2} to frame I_{n-1}. When frame I_n is available, by projecting the features detected in frame I_{n-1} to frame I_n using the previous motion $\mathbf{M}_{n-1}$, the projected location in frame I_n is highly likely to reside in the convergence region of KLT and therefore serves as a good starting point for KLT tracking process. We will describe this phenomenon again with the help of Figure 4.5. By transforming A with the previously estimated motion and projecting it onto the current image, the new position locates in *A4*, which is much closer to the ground truth. Using *A4* as the starting point, KLT is able to adapt to the motion in the current frame and finally correctly converge to *A1*.

There exist works that utilize SMC for ego-motion computation in a different manner from the proposed method. For example, the work in [253] utilizes SMC to remove outliers that exhibit incoherent movement. The work in [223] utilizes SMC to generate an additional set of augmented features that try to complement the original features. The work in [318] utilizes camera motion to define a specific search region in the image for normalized cross-correlation based feature matching. In order to avoid the danger that the predicted search region misses

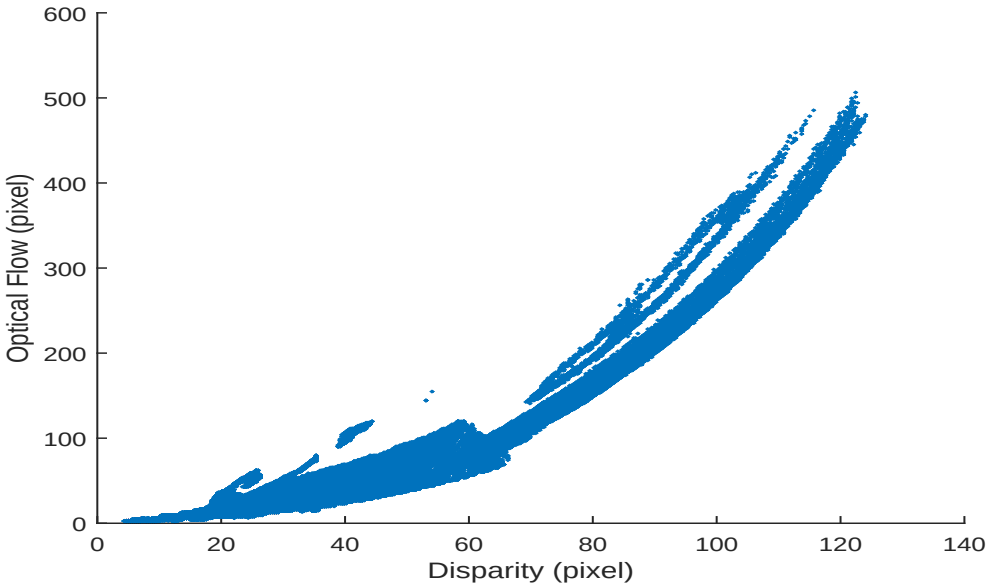


Fig. 4.6 Relationship between disparity and optical flow: pixels at a near distance from the camera are prone to large motion and pixels at a far distance are prone to small motion.

the target, a computationally complex two-step projection operation together with uncertainty calculation is performed in [318]. Moreover, an additional step is required for re-localization in the presence of high uncertainty. Unlike existing works, the proposed method utilizes SMC to increase the chance of correct convergence for KLT tracking process by determining a better starting point.

4.2.2.2 Adaptive Integration Window Technique

The size of the integration window for KLT tracker will affect not only the tracking accuracy but also the computational complexity. The conventional KLT tracker employs uniform window size and pyramid levels for all features. This easily violates the fact that a small window size is preferred to avoid smoothing out the details contained in the images while a large integration window is required to handle large motions.

In order to determine a suitable window size for KLT feature tracking, the relationship between the optical flow and the corresponding disparity field has been analyzed using the KITTI's flow/stereo benchmark. It can be observed from Figure 4.6 that pixels at a near distance from the camera are prone to large motion and pixels at a far distance are prone to small motion. Inspired by this idea, an adaptive window size for the KLT can be employed based on the disparity information. For features in the near region, a larger window size or more pyramid levels are used. For features in the far region, a small window size or lesser

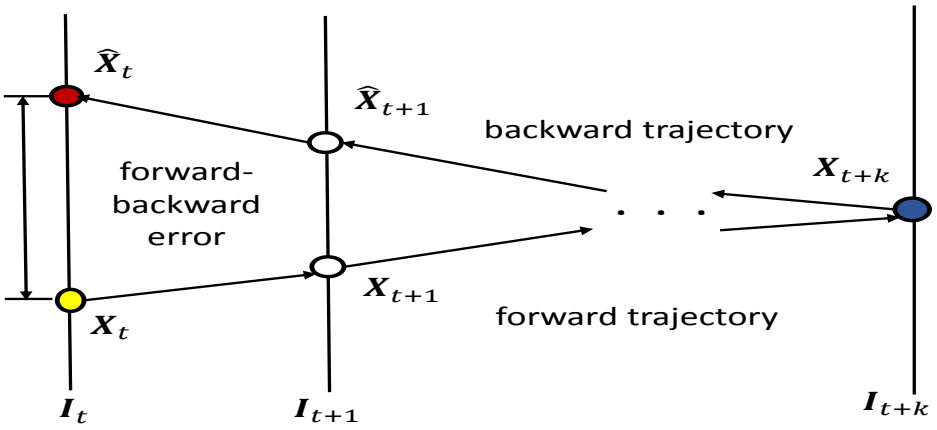


Fig. 4.7 Automatic tracking failure detection scheme. Figure from [319].

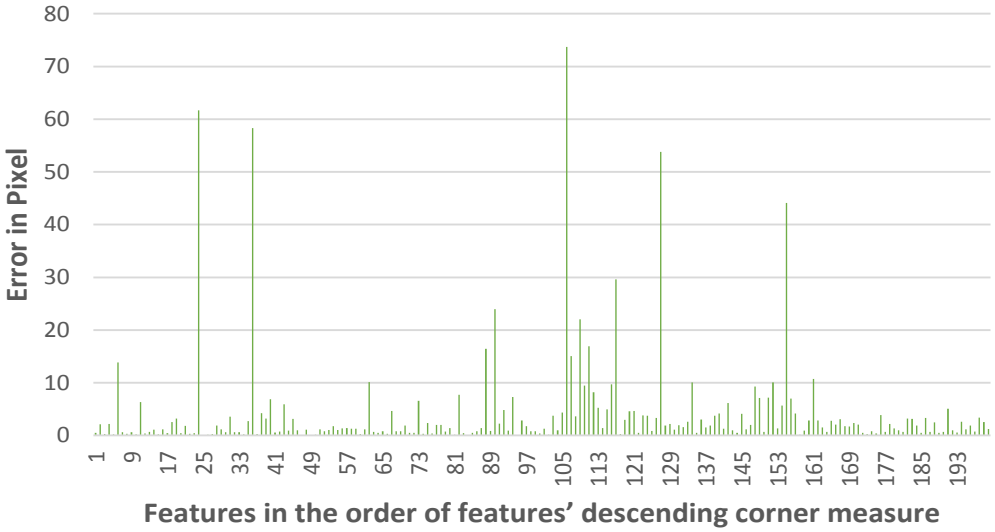


Fig. 4.8 Error distribution of optical flows estimated using KLT with automatic tracking failure detection.

pyramid levels are employed. This scheme helps to avoid the deployment of unnecessary large window size or pyramid levels for the features that undergo small motion, which will therefore improve the accuracy of the KLT tracker and also the compute efficiency.

In addition, large KLT window size or more pyramid levels which are typically required for features undergoing large motion can be further avoided by utilizing SMC. As shown in Figure 4.5, given the feature *B* detected in the previous frame, its ground truth correspondence is *B1* in the current frame. The initial search position employed by the conventional KLT is at *B2*, which is far from *B1*. As such, a large window size is needed to track feature *B* correctly. However, following the strategy proposed in previous sub-section to identify the initial point with the aid of SMC, the initial position to track feature *B* by the proposed method is set

at *B3*. It can be observed that *B3* is close to *B1* and the required window size and pyramid levels can therefore be set to a smaller value.

4.2.2.3 Automatic Tracking Failure Detection Scheme

As pointed out in [253], tracking error is unavoidable. In order to identify such tracking failures, an automatic tracking failure detection scheme that is presented in [319] is adopted. As illustrated in Figure 4.7, the basic idea is to check the forward-backward error during tracking. That is, forward and backward tracking is performed and the discrepancy between the starting point of the forward trajectory and the end-point of the backward trajectory is computed. If the forward-backward error is larger than some threshold, the corresponding feature pair is regarded as wrong setup and is therefore rejected. Figure 4.8 shows the new distribution of estimated flow error after applying the automatic tracking failure detection scheme. It can be observed that the majority of features with high tracking error in Figure 4.4 has been removed.

Based on the discussion above, it is proposed to improve the conventional KLT tracker as follows: 1) improve the tracking robustness by determining a better starting point for KLT tracking process with the aid of SMC; 2) improve tracking accuracy and efficiency by setting the integration window adaptively; 3) further improve the tracking robustness by enabling the automatic tracking failure detection scheme. The improved KLT feature tracking method is outlined in Listing4.2.

4.2.3 Gaussian-Newton based Motion Estimation with Early RANSAC Termination Condition

Given the set of feature correspondences, motion estimation computes the six motion parameters by solving the nonlinear least square problem as defined in Eq. 4.9. The Gaussian-Newton optimization algorithm [320] is chosen to solve this residual function as it avoids computing the second derivatives.

Starting with an initial estimate $\boldsymbol{e}^0$, the Gaussian-Newton algorithm iteratively converges to a local minimum through Eq. 4.17, where $\mathscr{L} = \{\mathfrak{l}\}$ is the set of residual functions and $\boldsymbol{J}_{\mathscr{L}} \in \mathbb{R}^{2k*6}$ represents the Jacobian matrix.

$$\boldsymbol{e}^{s+1} = \boldsymbol{e}^s - (\boldsymbol{J}_{\mathscr{L}}^T * \mathbf{J}_{\mathscr{L}})^{-1} * \boldsymbol{J}_{\mathscr{L}}^T * \mathscr{L}(\boldsymbol{e}^s) \tag{4.17}$$

Listing 4.2 Improved KLT Tracker

Input: Consecutive images I_{n-1} and I_n;
Detected feature set $\{\boldsymbol{m}^i_{n-1}\}$ in frame I_{n-1};
Previous ego-motion $\boldsymbol{M}_{n-1}$;
Disparity maps $disMap_{n-1}$ and $disMap_n$;
Calibrated camera parameters.

Output: Tracked feature correspondences $\{\boldsymbol{m}^i_n\}$ in I_n.

/* Forward Tracking */

for each feature $\boldsymbol{m}^i_{n-1}$ in frame I_{n-1} **do**
- Get its disparity value d^i_{n-1};
- Set $\boldsymbol{q}^i_n$ as its initial estimate in frame I_n:
  $\boldsymbol{p}^i_{n-1} = \mathfrak{g}(\boldsymbol{m}^i_{n-1})$;
  $\boldsymbol{p}^i_n = (\boldsymbol{M}_{n-1})^{-1} * \boldsymbol{p}^i_{n-1}$;
  $\boldsymbol{q}^i_n = \eta(\boldsymbol{p}^i_n)$;
- Perform KLT pyramid setup:
  Level = 1 (2 levels only);
  if $d^i_{n-1} < 10$ **then**
    integration window size r = 3;
  else if $d^i_{n-1} < 20$ **then**
    integration window size r = 5;
  else
    integration window size r = 7;
  end if
- Generate the new position of $\boldsymbol{m}^i_n$ by applying the KLT tracker.

end for

/* Backward Tracking */

for each feature $\boldsymbol{m}^i_n$ in frame I_n **do**
- Get its disparity value d^i_n;
- Set $\boldsymbol{q}^i_{n-1}$ as its initial estimate in frame I_{n-1}:
  $\boldsymbol{p}^i_n = \mathfrak{g}(\boldsymbol{m}^i_n)$;
  $\boldsymbol{p}^i_{n-1} = \boldsymbol{M}_{n-1} * \boldsymbol{p}^i_n$;
  $\boldsymbol{q}^i_{n-1} = \eta(\boldsymbol{p}^i_{n-1})$;
- Perform KLT pyramid setup:
  Level = 1 (2 levels only);
  if $d^i_n < 10$ **then**
    integration window size r = 3;
  else if $d^i_n < 20$ **then**
    integration window size r = 5;
  else
    integration window size r = 7;
  end if
- Generate the new position of $\boldsymbol{o}^i_{n-1}$ by applying the KLT tracker.

end for

Reject feature correspondences where $dist(\boldsymbol{m}^i_{n-1}, \boldsymbol{o}^i_{n-1}) > 1$ pixel.

$$\mathfrak{l}_i = \boldsymbol{w}_i(\boldsymbol{m}_n^i - \eta(\boldsymbol{R}_n * \mathfrak{g}(\boldsymbol{m}_{n-1}^i) + \boldsymbol{t}_n)), i = 1,2,\ldots,k \tag{4.18}$$

$$(\boldsymbol{J}_{\mathscr{L}})_{ij} = \frac{\partial \mathfrak{l}_i(\boldsymbol{e}^s)}{\partial \boldsymbol{e}_j}, i = 1,2,\ldots,2k; j = 1,2,\ldots,6 \tag{4.19}$$

Eq. 4.17 is repeatedly computed until the residual Δ in Eq. 4.20 is smaller than some predefined threshold.

$$\Delta = |\boldsymbol{e}^{s+1} - \boldsymbol{e}^s| = |(\boldsymbol{J}_{\mathscr{L}}^T * \boldsymbol{J}_{\mathscr{L}})^{-1} * \boldsymbol{J}_{\mathscr{L}}^T * \mathscr{L}(\boldsymbol{e}^s)| \tag{4.20}$$

The feature correspondences are usually contaminated with outliers. This is typically exhibited in feature correspondences extracted from moving objects such as pedestrians or vehicles. In addition, some feature points will be wrongly tracked. All of these noisy correspondences contribute to outliers and should be eliminated from the motion computation in order to increase the robustness of the estimated motion. In order to ensure robust estimation, the RANSAC algorithm [257] is adopted to identify outliers. The basic idea of RANSAC is to compute a fitting model from a set of samples selected randomly and check the number of points that are in consensus with the current estimated fitting model, i.e. inliers. This process is iteratively repeated until the maximum number of iterations has elapsed. Finally, the final model parameters are estimated using the largest set of inliers.

Instead of setting the maximum number of iterations manually, it has been pointed out in [257] that the number of iterations for RANSAC needed to achieve a desired accuracy requirement can be theoretically derived as shown below:

$$RANSAC_{iter} = \frac{\log(1-p)}{\log(1-ratio^n)} \tag{4.21}$$

Where n is the number of minimum points needed for estimating a model, $ratio$ is the percentage of inliers in the data points, p is the requested probability of success. Due to the formulation equation adopted in Eq. 4.9, at least three points are needed for estimating a model, therefore $n = 3$. It can be observed from Eq. 4.21 that $RANSAC_{iter}$ is dynamically determined based on the number of inliers found in current iteration. This means that once a set of inliers that are large enough are identified, there is no need to continue repeating

the RANSAC sampling operation anymore. We refer to this phenomenon as *Early RANSAC Termination Condition (ERTC)*. The proposed method has employed strategies to increase the accuracy of the extracted feature correspondences in previous sections. With the accurate feature correspondences provided from Stage 1, the Gaussian-Newton optimization with *ERTC* enabled is able to converge faster.

Based on the above discussion, given the set of correspondences $\{\boldsymbol{m}_{n-1}^i\}$ and $\{\boldsymbol{m}_n^i\}$, the method proposed for motion estimation is given in Listing 4.3 and Listing 4.4.

Listing 4.3 Gaussian-Newton Optimization Method (GNO)

Input: Feature correspondences $\{\boldsymbol{p}_i\}$;
Successful probability p;
Maximum Gaussian Newton iteration GN_{max};
Residual threshold t_{res};
Output: $\boldsymbol{e} = (t_x, t_y, t_z, r_x, r_y, r_z)^T$.
/* *Initialization* */
$\boldsymbol{e}^0 = 0$;
$s = 0$;
/* *Start Gaussian Newton Minimization Circle* */
while not converged and $s < GN_{max}$ **do**
  s = s + 1;
  Calculate $\boldsymbol{J}_{\mathscr{L}}$ at $\boldsymbol{e}^{s-1}$;
  $\boldsymbol{e}^s = \boldsymbol{e}^{s-1} - (\boldsymbol{J}_{\mathscr{L}}^T * \boldsymbol{J}_{\mathscr{L}})^{-1} * \boldsymbol{J}_{\mathscr{L}}^T * \mathscr{L}(\boldsymbol{e}^{s-1})$;
  $\Delta = \boldsymbol{e}^s - \boldsymbol{e}^{s-1}$;
  if $|\Delta| < t_{res}$ **then**
    Successfully converged;
  end if
end while
$\boldsymbol{e} = \boldsymbol{e}^s$.

4.3 Experimental Evaluation

In this section, the proposed method will be thoroughly evaluated using a large scale benchmark. The description of the way to setup experiment including the evaluation benchmark, the evaluation criteria and the baseline algorithms is presented first, which is followed by the evaluation of the proposed visual odometry algorithm in terms of accuracy and computation time by comparing it with the state-of-art baseline algorithms.

Listing 4.4 Motion Estimation

Input: Feature correspondences $\{\boldsymbol{m}_{n-1}^i\}$ and $\{\boldsymbol{m}_n^i\}$;
Successful probability p;
Output: $\boldsymbol{e} = (t_x, t_y, t_z, r_x, r_y, r_z)^T$
/* *Initialization* */
Transform features from 2D to 3D via triangulation:
$\boldsymbol{p}_{n-1}^i = \mathfrak{q}(\boldsymbol{m}_{n-1}^i)$ for each $i = 1, 2, \ldots, k$;
largest inlier set $\mathscr{A} = \emptyset$;
$RANSAC_{iter} = 50$;
$trial_{count} = 0$;
/* *RANSAC Iterative Refinement* */
while $trial_{count} < RANSAC_{iter}$ **do**
  $\{\boldsymbol{p}_i\} \leftarrow$ 3 correspondences selected randomly;
  $\boldsymbol{e} = GNO(\{\boldsymbol{p}_i\})$;
  Calculate current inlier set in_{curr} based on $\boldsymbol{e}$;
  if $in_{curr}.size > \mathscr{A}.size$ **then**
    $\mathscr{A} = in_{curr}$;
  end if
  Update $RANSAC_{iter}$ based on Eq. 4.21;
  $(trial_{count})$++;
end while
$\boldsymbol{e}$ = GNO($\mathscr{A}$)

4.3.1 Experimental Setup

4.3.1.1 Benchmarks

Experiments are conducted based on the widely known KITTI odometry evaluation platform [26]. The KITTI's odometry benchmark consists of 22 stereo *1344*391* sequences, where the first 11 sequences *(00-10)* are provided with ground truth trajectories for training and the remaining 11 sequences do not have ground truth. These 22 sequences were collected from stereo cameras installed in a vehicle that was driven around Karlsruhe, Germany. This benchmark covers a variety of road scenarios and provides a very challenging test-bed. Some samples of the benchmark are illustrated in Figure 4.9.

4.3.1.2 Evaluation Criteria

The evaluation criteria suggested by KITTI [26] is adopted, that is, **translational** and **rotational errors** for all possible subsequences of length $(100, ..., 800)$ meters. Translational errors are measured in percentage while rotational errors are measured in degrees per meter.

Fig. 4.9 Some samples of the KITTI odometry benchmark.

The average of these errors are used to compare the performance of various approaches in the KITTI evaluation platform. The implementation of these evaluation metrics provided in the KITTI odometry website [1] are adopted.

4.3.1.3 Baseline Algorithms

Many works have been submitted to the KITTI platform for evaluation. It can be observed that the existing solutions in KITTI evaluation platform fail to achieve a good balance of high accuracy and low computational complexity [26]. For example, [223] outperforms all other visual odometry methods in terms of translational and rotational accuracy till 2015, but it is time consuming. On the contrary, the work in [216] is able to achieve a real-time performance, but suffers from low accuracy. In the following, we will denote the work from [223] as *MFI* and the work from [216] as *VISO2-S*.

MFI: In order to reduce the motion drift caused by accumulation of feature tracking errors from frame to frame, *MFI* uses the whole history of the tracked feature points to compute the ego-motion. In their technique, the key idea is to integrate the features measured and tracked over all past frames into a single and improved estimate. An augmented feature set, obtained by the sample mean of all previous measured features which are transformed into the current frame, is added to the optimization formula. The importance of each feature is weighted in terms of their life age.

[1] http://www.cvlibs.net/datasets/kitti/eval_odometry.php

VISO2-S: In order to reduce the computational complexity, *VISO2-S* adopts a much simpler feature detector and descriptor. Feature locations are found by extracting the maximum or minimum Sobel filter response. In addition, instead of using the compute-intensive rotation and scale invariant feature descriptors like SURF or SIFT, features are described by concatenating the response over a sparse set of 16 locations within the 11*11 block and matched based on the sum of absolute differences (SAD) dissimilarity metric. Finally, the inputs to the visual odometry algorithm are features matched between four images, namely the left and right images of two consecutive frames. Given these 'circular' feature correspondences, the camera motion is computed by minimizing the sum of re-projection errors using the Gaussian-Newton optimization.

ORG-KLT: The proposed work is capable of rapidly extracting a set of accurate feature correspondences by improving the KLT feature tracker. In order to illustrate this improvement, the proposed algorithm will also be compared to the visual odometry algorithm that uses conventional KLT with automatic tracking failure detection ability and the motion estimation method proposed in Section 4.2.3. This baseline algorithm is denoted as *ORG-KLT*.

The differences between the baseline algorithms and the proposed algorithm have been highlighted in Table 4.1.

4.3.1.4 Implementation Details

For the proposed and *ORG-KLT* algorithm, up to 500 features are extracted for each frame and tracked in the consecutive frame. The required disparity information for features are provided by the OpenCV implementation of the Semi-Global Matching algorithm [95]. The default parameter settings in OpenCV are adopted and the multi-thresholding programming functionality inside OpenCV are not enabled.

Both of the proposed algorithm and *ORG-KLT* are implemented on a PC platform Hp Z420 Workstation, where the processor is Intel(R) Xeon(R) CPU E5-1650 v2 3.50 GHz with 16GB memory. All the codes are developed in C++ in the Visual Studio 2012 running in Windows 7. It is noteworthy that unlike the implementation of *MFI*, Not any code optimization technique like multi-threshold programming or GPU programming are employed for the proposed algorithm. For *MFI* and *VISO2-S*, the experimental figures reported in their papers are directly used.

4.3.2 Accuracy Evaluation

Table 4.1 Highlight of the algorithmic differences between the proposed visual odometry algorithm and the baseline algorithms.

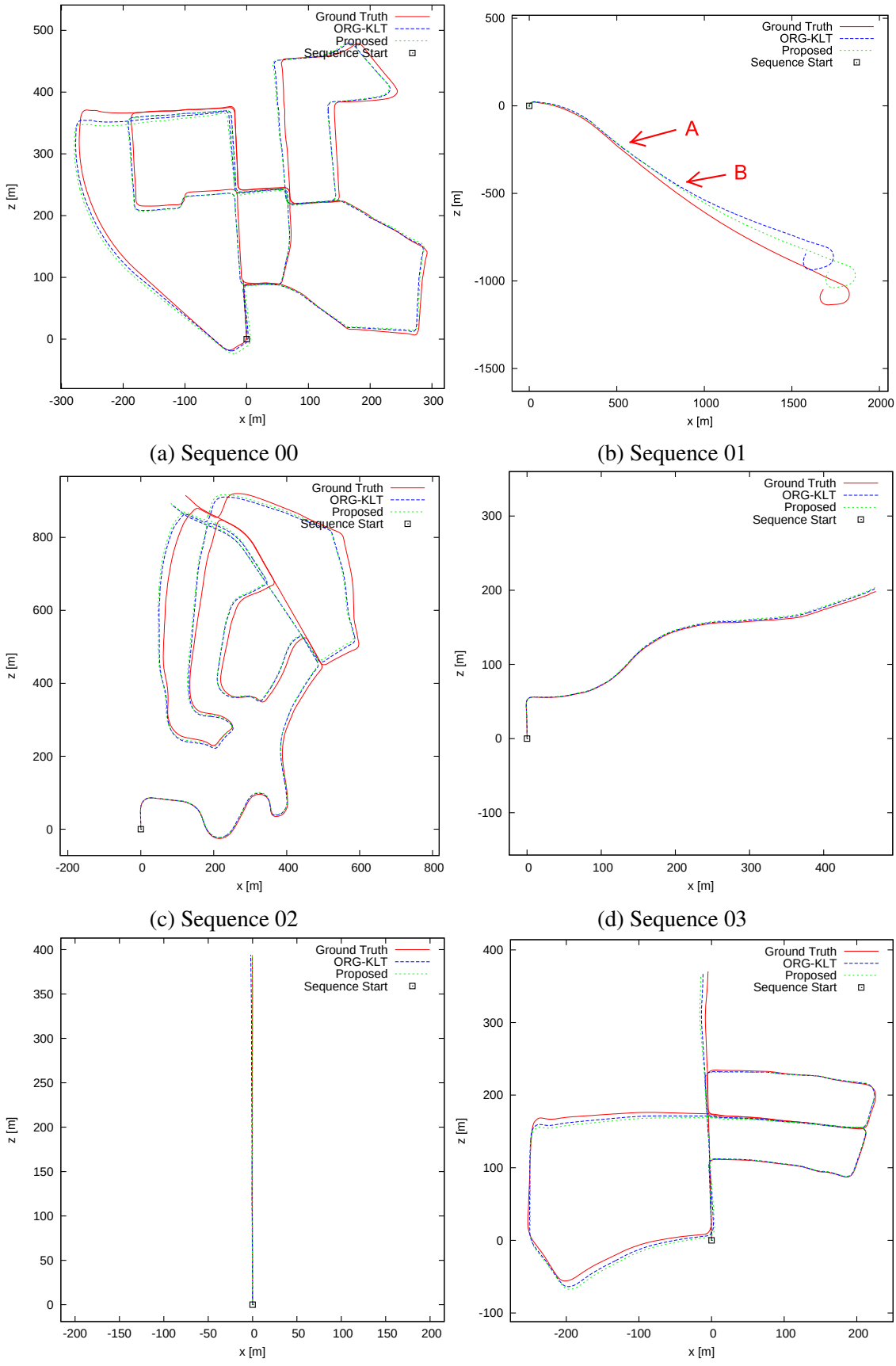
	Feature Correspondence Setup		**Motion Estimation**	
	Feature Detection	**Feature Tracking**	**Optimization Scheme**	**Robust Estimation**
MFI	Harris+FREAK	Feature matching by brute-force combinatorial search	Newton method based minimization of re-projection residual in left image space	Iteratively reject outliers whose re-projection residual is larger than some threshold
VISO2-S	Minima and maxima of blob and corner filter responses + concatenation of sobel filter responses based descriptor	SAD dissimilarity metric based two-passes circle feature matching	Gaussian-Newton method based minimization of re-projection residual in both of left and right image space + Kalman Filter Refinement	RANSAC
ORG-KLT	Conventional KLT corner detector	conventional KLT tracker with automatic tracking failure detection ability	Gaussian-Newton method based minimization of re-projection residual in left image space	RANSAC with early termination condition
Proposed	KLT corner detector with pruning	Improved KLT tracker	Gaussian-Newton method based minimization of re-projection residual in left image space	RANSAC with early termination condition

First, an extensive quantitative evaluation between *ORG-KLT* and the proposed algorithm is conducted based on the 11 training sequences. Figure 4.10 and Figure 4.11 show the ground-truth and estimated vehicle's trajectories from *ORG-KLT* and the proposed algorithm for these 11 training sequences. This provides an intuitive way to visualize the evaluation results. It can be observed that the estimated trajectories from the proposed algorithm are closer to the ground truth than the ones from *ORG-KLT*.

In particular, interesting phenomenons can be observed from Figure 4.10(b) corresponding to *Sequence 01*, where there exists a lot of challenging scenarios in the mid trajectory segments as discussed in Section IV.B. Firstly, it can be observed that the reconstructed paths from both of the *ORG-KLT* and the proposed algorithm deviate from the ground truth at Position *A* and persist for certain amount of time. At Position *B*, the reconstructed path from *ORG-KLT* deviates again. This means that the proposed method is more robust than *ORG-KLT* in dealing with challenging environment. Secondly, although the proposed method fails at position *A*, the reconstructed path from the proposed algorithm shows the same shape as the ground-truth at the end. This means that the proposed algorithm is able to automatically recover from wrong previous motion estimation when the scene is not challenging. The reason that the proposed algorithm is capable of recovering from wrong motion estimation in scenarios where the scene is not challenging is due to the fact that the utilization of SMC in the proposed method aims to increase the chance that the starting search point for KLT falls within the convergence region, thereby enabling the KLT tracker to more likely converge to the true global minimum. However, as pointed out by [316], KLT tracking process is tolerant to an initial parameter error as long as the initial point falls within the convergence region. Therefore, if the initial position guided by wrong motion still falls in the convergence region, KLT is still able to converge correctly and the proposed method is therefore able to recover from wrong previous motion estimation.

In addition, the average translational and rotational errors relative to the ground truth for both of the proposed algorithm and *ORG-KLT* for the 11 training sequences are presented in Figure 4.12. On average, the translation and rotation errors for *ORG-KLT* and the proposed algorithm for the 11 training sequences are (1.3974%, 0.0061[deg/m]) and (0.9768%, 0.0056[deg/m]). Therefore, the proposed algorithm is 30% better than *ORG-KLT*.

The results of the proposed method for the 11 test sequences have also been submitted to the KITTI odometry evaluation platform. The average translation and rotation errors relative to the ground truth for proposed algorithm, *MFI* and *VISO2-S* for the 11 test sequences are presented in Figure 4.13. In addition, the corresponding estimated trajectories from



(a) Sequence 00

(b) Sequence 01

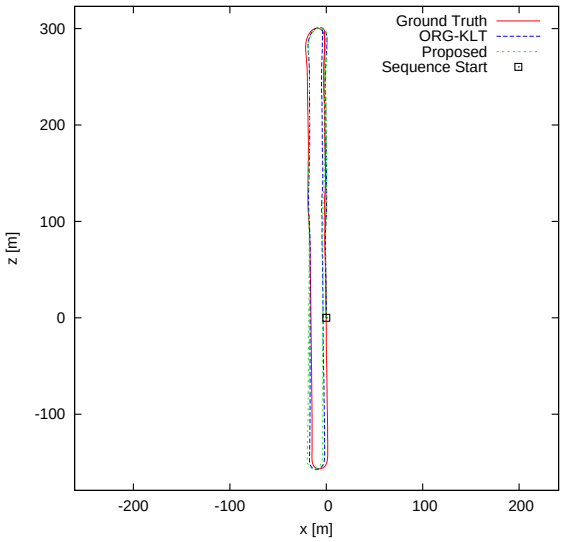
(c) Sequence 02

(d) Sequence 03

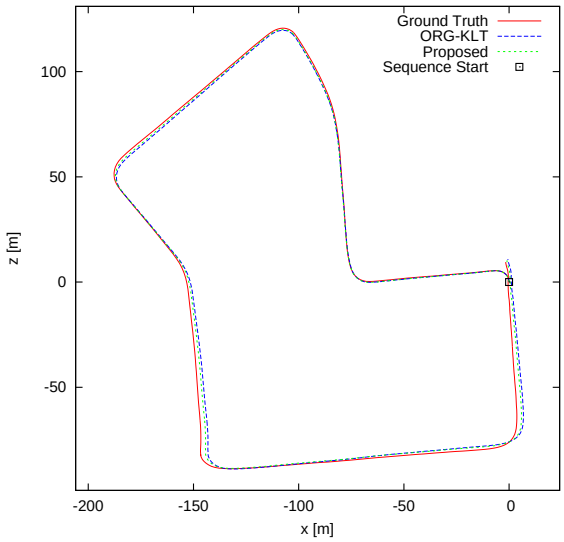
(e) Sequence 04

(f) Sequence 05

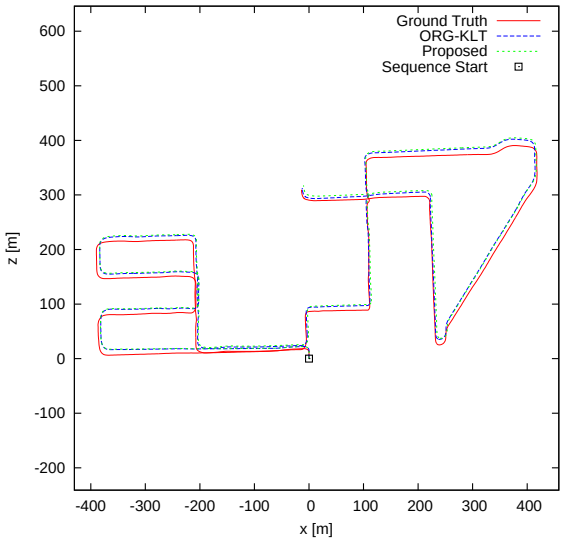
Fig. 4.10 Reconstruction of paths from *ORG-KLT* and *proposed* algorithm for Sequences 00-05.



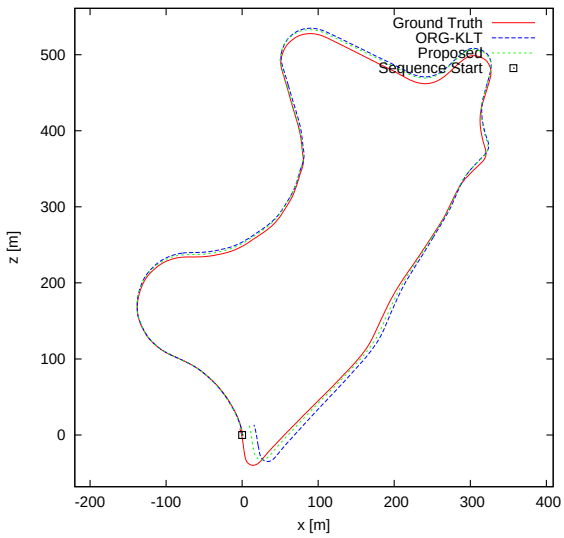
(a) Sequence 06



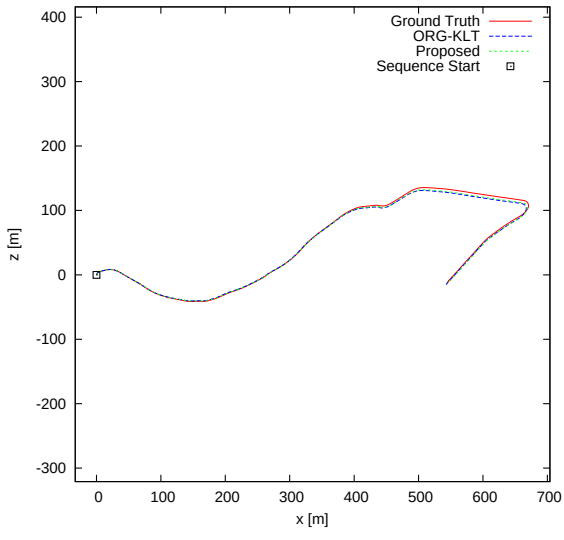
(b) Sequence 07



(c) Sequence 08



(d) Sequence 09



(e) Sequence 10

Fig. 4.11 Reconstruction of paths from *ORG-KLT* and *proposed* algorithm for Sequences 06-10.

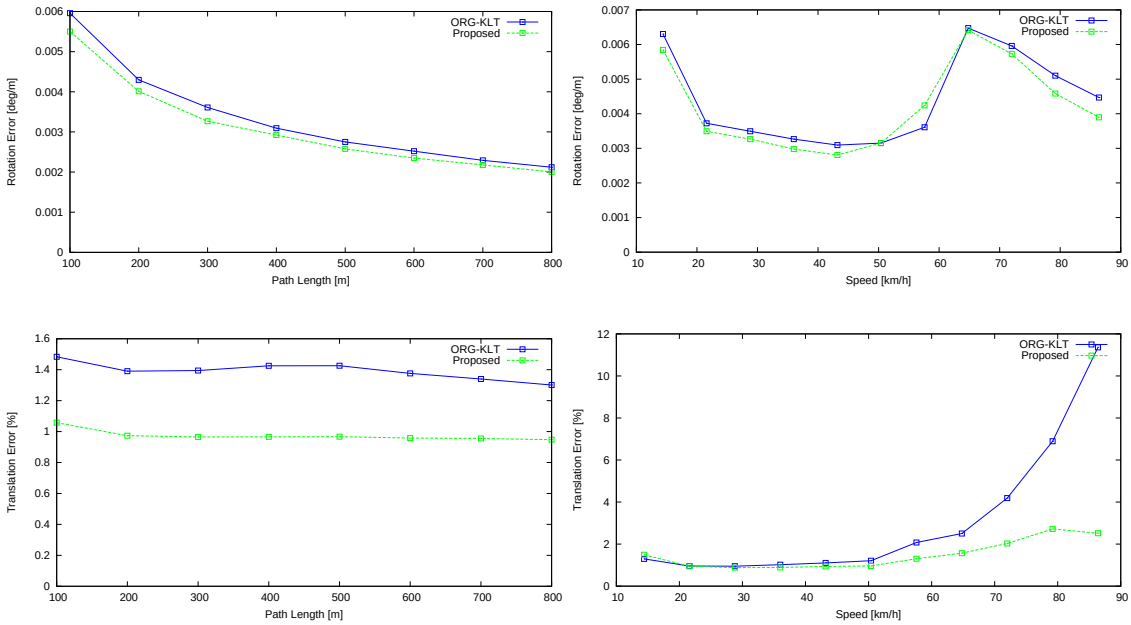


Fig. 4.12 Average translational and rotational error for *ORG-KLT* and proposed algorithm over sequences 00-10. The proposed algorithm is 30% better than *ORG-KLT*.

proposed algorithm, *MFI* and *VISO2-S* over sequences 11-15 (the website only provide the computed trajectories relative to the ground-truth for sequences 11-15) are depicted in Figure 4.14. On average, the translational and rotational errors for *MFI, VISO2-S* and the proposed algorithm for the 11 testing sequences are (1.30%, 0.0030[deg/m]), (2.44%, 0.0114[deg/m]) and (1.26%, 0.0038[deg/m]) respectively. It can be observed that the proposed algorithm performs approximately 3% better than *MFI* and 48% better than *VISO2-S*. At the time of submission into the KITTI odometry evaluation platform [2], the proposed method was ranked in the 8^{th} place for the visual odometry category in terms of accuracy, while *MFI* and *VISO2-S* were ranked in the 10^{th} and 38^{th} places respectively. As illustrated in Table 4.2, the proposed method is currently[3] ranked in 15^{th} place in terms of accuracy, while *MFI* and *VISO2-S* are ranked in the 16^{th} and 38^{th} places respectively.

[2]http://www.cvlibs.net/datasets/kitti/eval_odometry.php
[3]The time this thesis is completed.

Table 4.2 The proposed visual odometry method (FRVO) ranks in the 15^{th} place for the visual odometry categories on the KITTI odometry platform at the time this thesis is completed. Note that the KITTI website ranks all the methods including laser points (la) based and camera (st) based solutions together. The methods that are ranker higher than the proposed method include 3 laser based solutions.

	Method	**Setting**	**Translation**	**Rotation**	**Runtime**	**Environment**
1	V-LOAM	la	0.68 %	0.0016 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)
2	LOAM	la	0.78 %	0.0021 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)
3	HypER	st	0.88 %	0.0027 [deg/m]	0.25 s	2 cores @ 2.0 Ghz (C/C++)
4	SOFT	st	0.88 %	0.0022 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)
5	RotRocc	st	0.88 %	0.0025 [deg/m]	0.3 s	2 cores @ 2.0 Ghz (C/C++)
6	GVO	st	0.90 %	0.0027 [deg/m]	0.1 s	1 core @ 2.5 Ghz (C/C++)
7	svo2	st	0.94 %	0.0021 [deg/m]	0.2 s	1 core @ 2.5 Ghz (C/C++)
8	ROCC	st	0.98 %	0.0028 [deg/m]	0.3 s	2 cores @ 2.0 Ghz (C/C++)
9	cv4xv1-sc	st	1.09 %	0.0029 [deg/m]	0.145 s	GPU @ 3.5 Ghz (C/C++)
10	JDO	st	1.12 %	0.0030 [deg/m]	0.06 s	1 core @ >3.5 Ghz (C/C++)
11	DEMO	la	1.14 %	0.0049 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)
⋮	⋮	⋮	⋮	⋮	⋮	⋮
18	**FRVO**	**st**	**1.26 %**	**0.0038 [deg/m]**	**0.03 s**	**1 core @ 3.5 Ghz (C/C++)**
19	MFI	st	1.30 %	0.0030 [deg/m]	0.1 s	1 core @ 2.2 Ghz (C/C++)
⋮	⋮	⋮	⋮	⋮	⋮	⋮
41	VISO2-S	st	2.44 %	0.0114 [deg/m]	0.05 s	1 core @ 2.5 Ghz (C/C++)
⋮	⋮	⋮	⋮	⋮	⋮	⋮
59	LPF-II		6.13 %	0.0096 [deg/m]	5 s	1 core @ 2.5 Ghz (C/C++)
60	IO	st	6.55 %	0.0315 [deg/m]	5 s	1 core @ 2.5 Ghz (Matlab)
61	VISO2-M + GP		7.46 %	0.0245 [deg/m]	0.15 s	1 core @ 2.5 Ghz (C/C++)
62	RMVO		8.33 %	0.0233 [deg/m]	0.01 s	1 core @ 3.5 Ghz (C/C++)
63	VISO2-M		11.94 %	0.0234 [deg/m]	0.1 s	1 core @ 2.5 Ghz (C/C++)
64	1SCTE		15.08 %	0.0171 [deg/m]	0.02 s	1 core @ 2.5 Ghz (Matlab)
65	OABA		20.95 %	0.0135 [deg/m]	0.5 s	1 core @ 3.5 Ghz (C/C++)

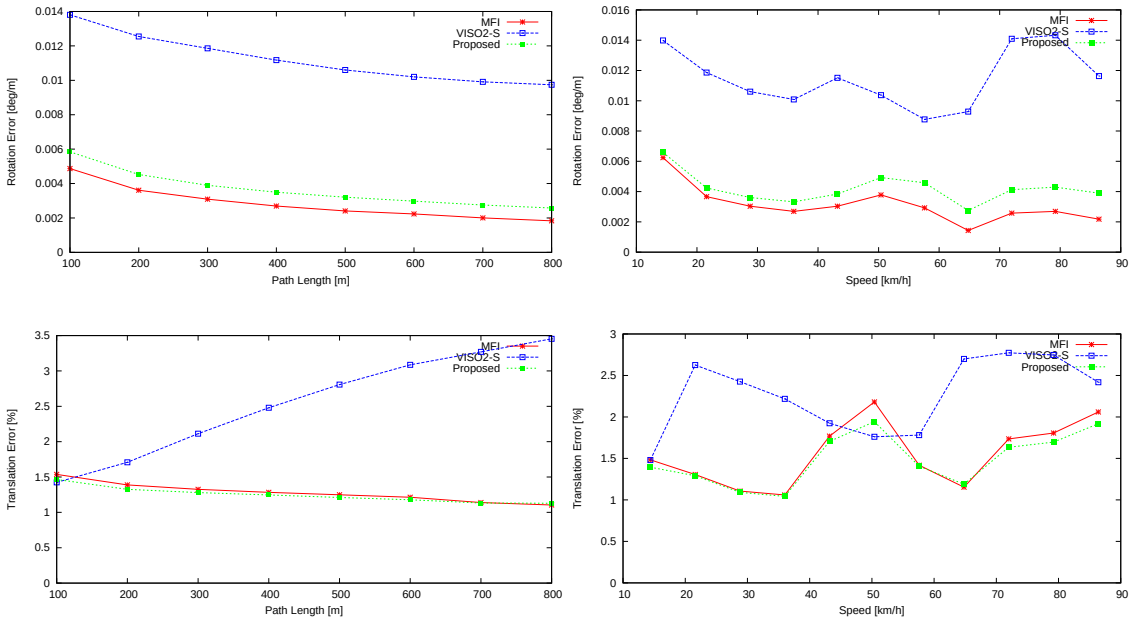
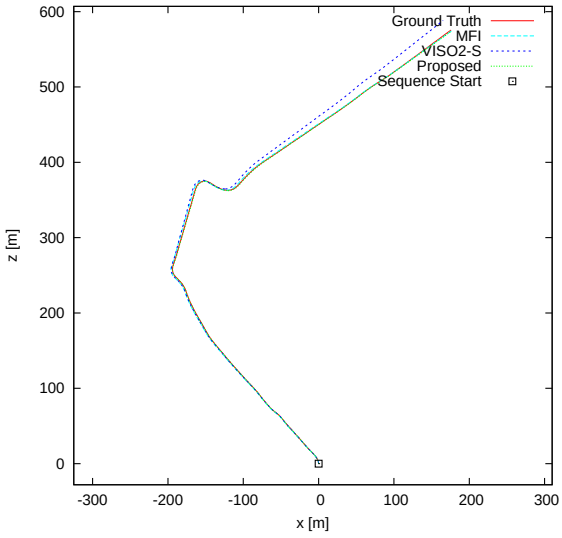


Fig. 4.13 Average translational and rotational error for *MFI*, *VISO2-S* and *proposed* algorithm over Sequences 11-21. The proposed algorithm performs 3% better than *MFI* and 48% better than *VISO2-S*.

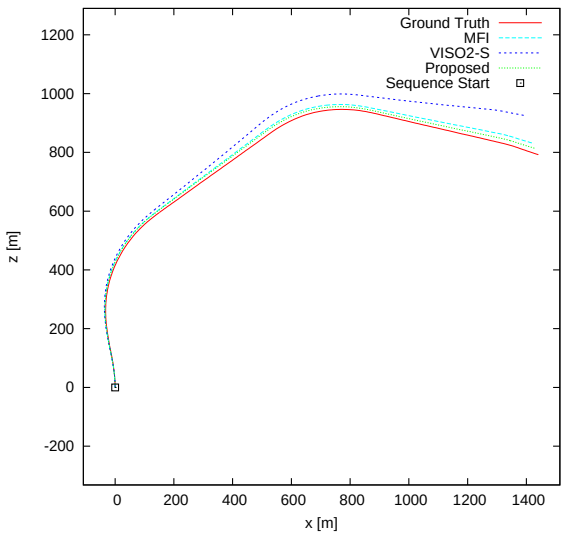
4.3.3 Runtime Performance Evaluation

Table 4.3 shows the computation time for the proposed algorithm and all the three baseline algorithms. In the current implementation, the dense disparity map is directly provided for the proposed algorithm and *ORG-KLT*. For a fair comparison, the computation time for disparity computation is not included for all the four algorithms. The proposed algorithm is 28% faster than *ORG-KLT*. It can be observed that both of the two stages in the proposed method contribute to runtime performance gain. This is due to the low computational complexity strategies adopted in the feature correspondence setup. In addition, since robust techniques during the KLT tracking process and the RANSAC with early termination rule are employed, the set of accurate feature correspondences allows the Gaussian-Newton process to converge faster.

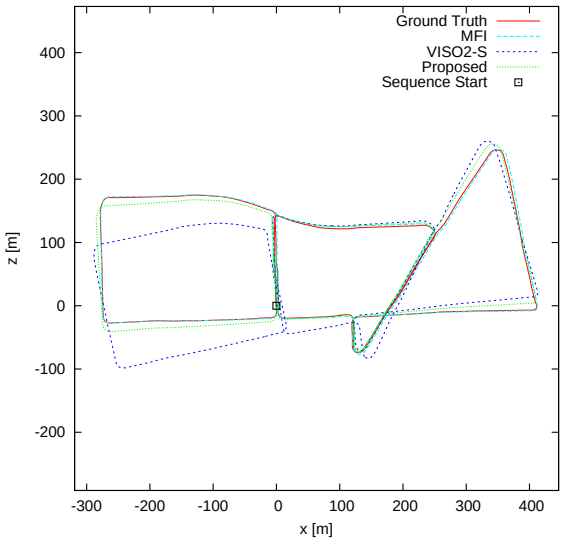
MFI is able to reduce pose error compared to their earlier work [253, 254], however this is achieved at the expense of huge computational complexity. Up to 4,096 features are tracked between consecutive frames. The key-points are matched between consecutive frames by brute-force combinatorial search. The computation time reported by *MFI* is only possible after they enable the multi-thresholding programming technology OpenMP and intense



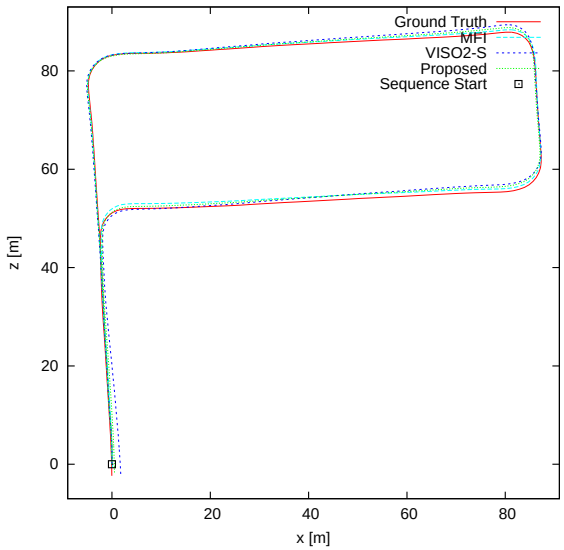
(a) Sequence 11



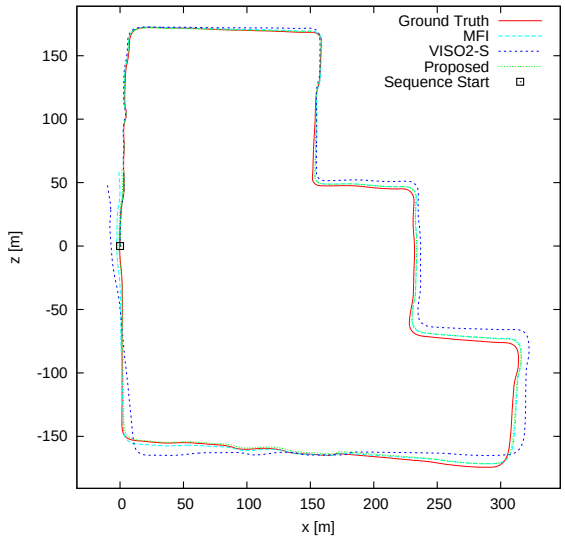
(b) Sequence 12



(c) Sequence 13



(d) Sequence 14



(e) Sequence 15

Fig. 4.14 Reconstruction of paths from *MFI*, *VISO2-S* and *proposed* algorithm for Sequences 11-15.

code optimization using the Intel Performance Primitives library on 2.7 GHz CPU with 4 cores. Finally, *VISO2-S* achieves a short computation time at the price of reduced estimation accuracy.

The proposed method also exhibits lower computational complexity when compared to other methods that are recently submitted to the KITTI evaluation platform. For example, the computation time for the top three visual odometry methods in the KITTI platform (at the time this paper is submitted) are between 0.1 to 0.3 seconds/frame on 2.0Ghz or 2.5 Ghz platforms with dual cores. The computation time for the proposed algorithm is 0.03 seconds/frame on a 3.5GHz platform (one core) without utilizing any code optimization technique (e.g. multi-threshold programming or GPU programming). We are confident that the computation time of the proposed method will further reduce if such code optimization techniques are enabled.

Table 4.3 Runtime performance comparison between the proposed visual odometry algorithm and the baseline algorithms.

	Correspondence Setup	**Motion Estimation**	**Total**	**Platform**
MFI	37.1 ms	8.8 ms	45.9 ms	2.7GHz(4 Cores)
VISO2-S	36.6 ms	4.3 ms	40.9 ms	2.5GHz(1 Core)
ORG-KLT	40.8 ms	1.1 ms	41.9 ms	3.5GHz(1 Core)
Proposed	29.5 ms	0.8 ms	30.3 ms	3.5GHz(1 Core)

4.4 Summary

It has been shown that the proposed method for estimating the ego-motion of vehicle overcomes the limitations of existing solutions by integrating runtime-efficient strategies with robust techniques at various core stages in visual odometry. A novel pruning technique is adopted to notably reduce the computational complexity of detecting corner features without compromising on the quality of the extracted corner features. A robust and compute-efficient KLT tracker is proposed to facilitate the generation of the feature correspondences in a robust and runtime efficient way. The accuracy of extracted feature correspondences is improved by leveraging on ego-motion prior to determine a better initial point for fast and accurate feature convergence during tracking and incorporating an automatic tracking failure detection scheme to exclude the feature correspondences with large tracking error. In addition, the computational complexity of the conventional KLT has been improved by setting the

integration window size adaptively. With the accurate feature correspondences provided, Gaussian-Newton optimization scheme supported by an early RANSAC termination condition is shown to converge faster in the motion estimation process. The above contributions are integrated into a framework for fast and robust visual odometry. The experimental results based on a widely used evaluation platform clearly demonstrate the advantages of the proposed framework over existing state-of-the-art solutions for robust and runtime-efficient visual odometry.

Obstacle detection and tracking is one of the most essential modules in CASs. In the next chapter, a robust and low complexity stereo-vision based obstacle detection and tracking method is proposed.

CHAPTER 5

LOW-COMPLEXITY TECHNIQUES FOR ROBUST OBSTACLE DETECTION AND TRACKING

Obstacle detection and tracking are essential modules for collision avoidance systems. The behaviors of obstacles can be understood only after they are detected and tracked. Vision based obstacle detection and tracking is challenging due to a number of factors [198]. A detailed literature review on existing works for obstacle detection and tracking has been conducted in Section 2.3.2 and Section 2.3.3 respectively. Despite the tremendous progress in recent decades, object detection and tracking remains an unsolved problem [200, 206, 207].

In this chapter, a robust and low complexity stereo vision based obstacle detection and tracking method is proposed. Low complexity techniques are employed to detect obstacles in the u-v-disparity image space. In addition, effective strategies are proposed to construct a distinctive object appearance model for data association efficiently. Finally, an online multi-object tracking framework is proposed by integrating the obstacle detection and data association modules in a robust way. Extensive experimental results on the well-known KITTI tracking dataset demonstrate that the proposed method is able to detect and track various obstacles robustly and efficiently in diverse challenging scenarios. The works presented in this chapter have been published in [10–12].

This chapter is organized as follows: Section 5.1 introduces some mathematical principles which will be relied upon in this chapter. The proposed robust and low complexity stereo vision based obstacle detection and tracking method is presented in Section 5.2. A comprehensive evaluation of the proposed method with the state-of-art baseline algorithm based on the KITTI tracking benchmark is conducted in Section 5.3. Finally, Section 5.4 summarizes this chapter.

5.1 Mathematical Principles

In this section, some mathematical principles that serve as the the mathematical foundations of the proposed algorithm in Section 5.2 are presented.

A detailed introduction to the stereo geometry has been introduced in Section 3.1.1. When the pitch angle in the stereo rig in Section 3.1.1 is assumed to be zero and the origin of the Euclidean space is at the center of the left image, the relationship between a point (x,y,z) in the world coordinate system and its projection (u,v,d) in the image coordinate system can be simplified as follows:

$$d = \frac{focal * baseline}{z} \tag{5.1}$$

$$u = \frac{focal * x}{z} + u_0 \tag{5.2}$$

$$v = \frac{focal * y}{z} + v_0 \tag{5.3}$$

Where $baseline$ is the stereo baseline; $focal$ is the focal length measured in pixel; (u_0, v_0) is the camera's principal point.

From Eq. 5.1, it can be observed that the disparity d is inversely proportional to the depth z. Hence, the distance interval in z direction corresponding to a minor change in d is shown in Eq. 5.4

$$\Delta z_{d,d+1} = \frac{focal * baseline}{d} - \frac{focal * baseline}{d+1} = \frac{focal * baseline}{d(d+1)} \tag{5.4}$$

Assuming another disparity value $d' = \kappa * d$, then

$$\Delta z_{d',d'+1} = \frac{focal * baseline}{d'} - \frac{focal * baseline}{d'+1} = \frac{focal * baseline}{\kappa d * (\kappa d + 1)} \approx \frac{1}{\kappa^2} \Delta z_(d, d+1) \tag{5.5}$$

Eq. 5.5 indicates that a minor change in d correspond to a minor change in z in the near region and a large change in depth z in the far region.

In order to cover the same distance interval as $\Delta z_{(d,d+1)}$ from d', that is, in order to obtain Eq. 5.6:

$$\Delta z_{(d,d+1)} = \Delta z_{(d',d'+\zeta)} \tag{5.6}$$

ζ should be as shown in Eq. 5.7:

$$\zeta = \frac{\kappa^2 d}{d+1-\kappa} \tag{5.7}$$

In addition, it can be found from Eq. 5.2 that a line segment spanning from the point (x_1, y, z) to the point (x_2, y, z) with length $\Delta x = x_2 - x_1$ in the world coordinate system will be projected into a line segment with the length Δu_d in image space:

$$\Delta u_d = \frac{\Delta x * d}{baseline} \tag{5.8}$$

Also, a line segment spanning from the point (x, y_1, z) to the point (x, y_2, z) with height $\Delta y = y_2 - y_1$ in the world coordinate system will be projected into a line segment with the height Δv_d in image space according to Eq. 5.3:

$$\Delta v_d = \frac{\Delta y * d}{baseline} \tag{5.9}$$

5.2 Proposed Algorithm

In this section, the proposed obstacle detection and tracking method is presented. The proposed obstacle detection method relies on the u-v-disparity space to detect all the obstacles

in the scene. A Space of Interest (SOI) is defined to greatly reduce the search space of obstacles prior to employing the adaptive connected component labeling techniques to segment SOI into sets of obstacles based on the u-disparity image. To associate obstacles across frames, a color histogram based appearance model is constructed for each obstacle. In order to incorporate robustness of the model to inconsistent illumination, L*a*b* color space is utilized. Moreover, pixels belonging to the background are excluded based on the depth information when constructing the histogram, which further increases the distinctiveness of the appearance model. A chessboard pattern based sparse sampling technique is also adopted to significantly reduce the number of operations and memory accesses for constructing the histogram. The similarity measure to associate obstacles across frames takes into account color, motion and spatial distance between obstacles. Finally, an online multi-object tracking framework is proposed by integrating the obstacle detection and data association modules in a robust way. The corresponding top-level block diagram for the proposed obstacle detection and tracking method is depicted in Figure 5.1.

5.2.1 Obstacle Detection

Unlike the works which focus on detecting vehicle [27, 321, 322] or pedestrian [323, 324] only, the proposed method detects all the obstacles in the scene. This is achieved with the help of the u-v-disparity image space. A detailed discussion of the concepts related to the u-disparity image and v-disparity image has been presented in Section 3.1.2. From the analysis presented in Section 3.1 and Section 5.1, it can be found that the problem of detecting obstacle is mathematically equivalent to locating the peak regions in the u-disparity image.

As shown in Figure 5.1, the proposed obstacle detection method consists of three steps: 1) Generation of Space of Interest (SOI); 2) Segmentation of SOI; 3) Determination of the final bounding box for obstacles.

5.2.1.1 SOI Generation

In collision avoidance systems, only objects that are in close proximity to the ego-vehicle are of concern. This is due to the fact that it is impossible for the ego-vehicle to collide with the far objects e.g. those in the sky or high objects like the higher part of the trees which hangs over the road. In addition, road surface is another important scene element, which is a region that is exclusive to the obstacles. The aim of this step is to remove all of the irrelevant image regions based on the knowledge of the geometrical structure of the scene. This enables the

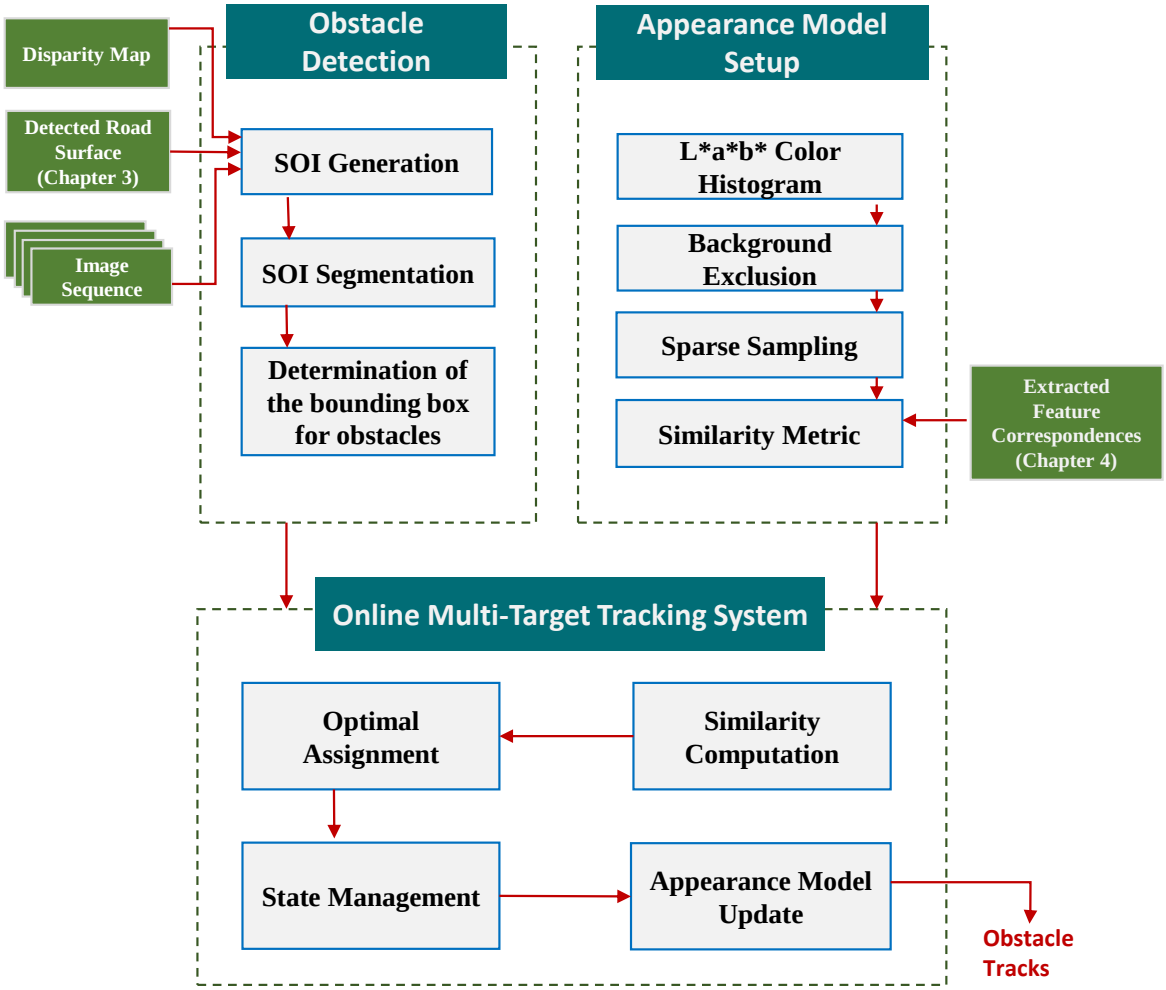


Fig. 5.1 Top-level block diagram of the proposed obstacle detection and tracking method: The proposed obstacle detection method relies on the u-v-disparity space to detect all the obstacles in the scene. A Space of Interest (SOI) is defined to greatly reduce the search space of obstacles prior to employing the adaptive connected component labeling techniques to segment SOI into sets of obstacles based on the u-disparity image. To associate obstacles across frames, a color histogram based appearance model is constructed for each obstacle. In order to incorporate robustness of the model to inconsistent illumination, L*a*b* color space is utilized. Moreover, pixels belonging to the background are excluded based on the depth information when constructing the histogram, which further increases the distinctiveness of the appearance model. A chessboard pattern based sparse sampling technique is also adopted to significantly reduce the number of operations and memory accesses for constructing the histogram. The similarity measure to associate obstacles across frames takes into account color, motion and spatial distance between obstacles. Finally, an online multi-object tracking framework is proposed by integrating the obstacle detection and data association modules in a robust way.

search space for obstacles to be greatly reduced and some false positives can be rejected at an early stage. The remaining image regions are referred to as Space of Interest (SOI), where each obstacle is an individual object within the SOI.

In the proposed method, SOI is generated as follows: Firstly, pixels that correspond to road surface are removed. The road surface detection method that is proposed in Chapter 3 is utilized here. Details of the road surface detection method can be found in Chapter 3. Secondly, scene points that are too far or above certain height are excluded. That is, scene point (u,v,d) whose position does not meet one of following requirement is removed.

$$|x| = |\frac{(u-u_0)*baseline}{d}| \leq threshold\_SOI\_x \tag{5.10}$$

$$y = \frac{(v_0-v)*baseline}{d} \leq threshold\_SOI\_y \tag{5.11}$$

$$z = \frac{baseline*focal}{d} \leq threshold\_SOI\_z \tag{5.12}$$

where $baseline$ is the baseline length of the stereo camera rig, $focal$ is the focal length of the camera, $(thresold\_SOI\_x, thresold\_SOI\_y, thresold\_SOI\_z)$ define the 3D boundaries of SOI in the world coordinate system.

After removing the irrelevant regions, the remaining region in the image is the SOI as illustrated in Figure 5.2 (c). It is evident that the SOI significantly reduces the search space for obstacle detection. In addition, the noise disturbance outside this region is removed, which can lead to higher obstacle detection accuracy as discussed in the next section.

5.2.1.2 SOI Segmentation

Segmentation of SOI into set of obstacles is performed on the u-disparity image. The value for each pixel (u,d) in the u-disparity image represents the number of pixels that are located in column u of the original image where the corresponding disparity value is d. Hence, up-right obstacle points with the same x and z values will converge onto the same position in the u-disparity image, thereby producing peak regions in the u-disparity image. The SOI segmentation step in this section aims to detect these peak regions in the u-disparity image and partition them into a set of individual clusters.

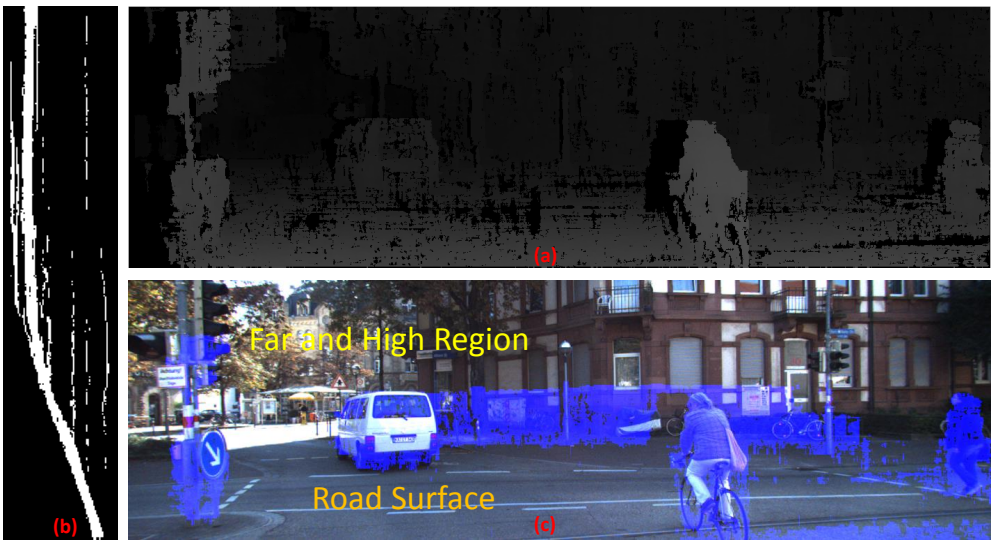


Fig. 5.2 SOI generation: (a) a disparity map; (b) the corresponding v-disparity image; (c) space of interest highlighted in blue, which is obtained by removing the pixels corresponding to the road surface and far and high regions;

Accurate segmentation of the peak regions in the u-disparity image is a challenging task. As shown in Figure 5.3(a), a lot of noise exist in the original u-disparity image. Figure 5.3(b) shows that the noise can be removed when the image is thresholded with a large threshold. This however will lead to the problem where a single object is also likely to be partitioned into multiple parts. On the other hand, when a small threshold is used, noise cannot be easily removed and multiple objects are likely to be spatially connected due to the existence of noise as shown in Figure 5.3(c).

In order to solve this problem, the hysteresis thresholding technique in [325] is applied to the u-disparity image to remove noise and locate the peak regions. Two thresholds are maintained, which are denoted as *height_threshold_high* and *height_threshold_low* respectively. *height_threshold_low* is set to be in certain percentage of *height_threshold_high*. If the pixel value in u-disparity image is higher than *height_threshold_high*, it is marked as *strong u-peak*. If the pixel value is between *height_threshold_high* and *height_threshold_low*, it is marked as *weak u-peak*. The pixels marked as *strong u-peak* correspond to true objects, and pixels in the u-disparity image whose value is lower than *height_threshold_low* are suppressed as noise. Special treatment is needed for the pixels that are marked as *weak u-peak*, as they can either correspond to true objects or noise. The *weak u-peaks* that correspond to noise need to be removed. It was observed that typically, the *weak u-peaks* corresponding valid objects will usually be connected to the *strong u-peak*. Hence, as long as there is a

strong u-peak falling into its 8-connected neighborhood, the *weak u-peak* is re-labeled as *strong u-peak*. The remaining *weak u-peaks* are then suppressed as noise.

From the analysis in Section 5.1, it can be inferred that the setting of the two thresholds *height_threshold_high* and *height_threshold_low* cannot be uniform across the whole u-disparity image. Instead, they are set adaptively as presented in Eq. 5.13 and Eq. 5.14.

$$height\_threshold\_high = \lceil \frac{min\_height * u\_d}{baseline} \rceil \tag{5.13}$$

$$height\_threshold\_low = height\_threshold\_high * ratio \tag{5.14}$$

Where *min_height* represents the minimum height for an object that will be focused, *u_d* represents the disparity value which corresponds to a specific row index in the u-disparity image. Both *min_height* and *ratio* are predefined values. It is worthy to note that the computation of *height_threshold_high* and *height_threshold_low* can be computed in advanced and stored in a small look-up table.

Since the disparity value is not linear in Euclidean space, in the regions that are very near to the ego-vehicle, a small change in z will be projected into a region covering a large change in d. Therefore, one problem with the above thresholding operation is that, for certain object that is near, a lot of weak u-peaks rather than strong u-peaks are generated and suppressed as noise. At this time, this object is easily segmented into several parts. Such phenomenon is illustrated in Figure 5.3(d) and Figure 5.3(e). The white regions in Figure 5.3(d) represent the *strong u-peak* regions identified using the hysteresis thresholding technique. Without applying any compensation, the thresholding operation divides the near bicyclist into two parts. In order to deal with this problem, a box filter is applied into the near region in the u-disparity image before the hysteresis thresholding operation. The white regions in Figure 5.3(f) represent the final *strong u-peak* regions identified by applying box filter based compensation and using the hysteresis thresholding technique. As can be observed, the bicyclist is no longer to be divided into two parts.

Once all the *strong u-peak* regions in the u-disparity image are identified, an adaptive connected component labeling technique is utilized to group these peak regions into set of clusters. Two contributions are made here. Firstly, unlike the classical connected component labeling algorithm [326], which processes at the pixel level, the proposed approach processes based on

u-span. *u-span* refers to a continuous interval of peak regions in each row of the u-disparity image. Processing at the *u-span* level can lead to significant reduction in the clustering complexity. A *u-span* whose left-most position, right-most position and associated disparity value are u_left, u_right, and u_d respectively is denoted as $u\text{-}span[u\_left, u\_right, u\_d]$. Secondly, an adaptive connectivity is employed for different *u-span*s when performing the connected component labeling. This is to account for the problem caused by the fact that a minor change in d will result in a minor change in z for near regions but a major change in z for far regions. The proposed method overcomes the problem with an adaptive strategy that allows the connectivity to be changed based on disparity values.

Let $Rect[r\_left, r\_right, r\_top, r\_bottom]$ denote a rectangular region in the u-disparity image whose left-most, right-most, top-most and bottom-most position are r_left, r_right, r_top and r_bottom respectively. Then the connectivity, i.e. examination neighborhood region, of a $u\text{-}span[u\_left, u\_right, u\_d]$ is defined as $Rect[u\_left - u\_neighborhood, u\_right + u\_neighborhood, u\_d + 1, u\_d + d\_neighborhood]$. $d\_neighborhood$ and $u\_neighborhood$ are determined adaptively based on Eq. 5.15 $\sim$ Eq. 5.17, which are derived based on the analysis in Section 5.1:

$$\kappa = \frac{u\_d}{d\_reference} \tag{5.15}$$

$$d\_neighborhood = \lceil \frac{\kappa^2 * d\_reference}{d\_reference + 1 - \kappa} \rceil \tag{5.16}$$

$$u\_neighborhood = \lceil \frac{min\_\Delta x * u\_d}{baseline} \rceil \tag{5.17}$$

Where the depth distance defined by $\Delta z_{(d\_reference, d\_reference+1)}$ is the minimum distance set to separate two objects in the z direction. $min\_\Delta x$ is the minimum distance set to separate two objects in the x direction. It is noteworthy that the computation of $d\_neighborhood$ and $u\_heighborhood$ can be computed in advanced and stored in a small look-up table. For a given *u-span* ψ, any *u-span* $\xi \neq \psi$ which falls in the examination neighborhood region of ψ is assumed to be connected to ψ.

Each cluster obtained from the adaptive connected component labeling technique corresponds to one obstacle in the scene. Figure 5.3(f) shows the clustering results by applying the adaptive connected component labeling technique onto the final strong u-peak regions, which are

generated by applying the box filter for non-linearity compensation and using the proposed hysteresis thresholding technique.

5.2.1.3 Determination of the Bounding Box for Obstacles

Each cluster obtained from the earlier step corresponds to one obstacle in the scene. The bounding box of the obstacle is determined as follows: The left and right boundaries of obstacle are determined by the left-most and right-most positions of the cluster in the u-disparity image. A typical way to determine the top and bottom boundaries of the bounding box is to rely on the vertical line in the v-disparity image. This operation however can lead to inaccuracies since the height of a vertical line in the v-disparity image corresponds to the highest object at that distance (and not necessary the object of interest). When there are two objects with different heights at the same distance from the vehicle, the height of the shorter object will be wrongly determined since its vertical line in v-disparity image is occluded by the vertical line of the higher object. Instead, the top and bottom boundaries of the bounding box are determined by scanning from the row corresponding to the road in the disparity map, and identifying the regions whose disparity value is in the range of disparity values covered by the corresponding cluster. Figure 5.3(g) illustrates the final detection result. The pseudo code for the proposed obstacle detection method is presented in Listing 5.1.

5.2.2 Appearance Model Setup

Appearance model refers to the representation of object based on specific features. A good model should be able to accurately distinguish the object from its surroundings, i.e. the model should exhibit high similarity for the same object and low similarity for different objects. The appearance of the obstacle in the urban scene is easily affected by many factors like inconsistent illumination, partial occlusion, scale and view point change and so on. Hence, the design of a good appearance model needs to take into account these factors.

In this work, a color histogram is constructed to represent each obstacle. The color histogram is utilized due to its simplicity and its high tolerance to scale and view angle change and partial occlusion. In order to reduce the sensitivity of the model to illumination change, the *L*a*b** color space is employed. The distinctiveness of the model is also enhanced by excluding the pixels corresponding to the background while constructing the histogram. The histogram intersection based similarity measure is enhanced by taking into consideration the motion and spatial distance information between obstacles. Finally, to further reduce the computational complexity of the proposed algorithm, a chess-board pattern based sampling technique is

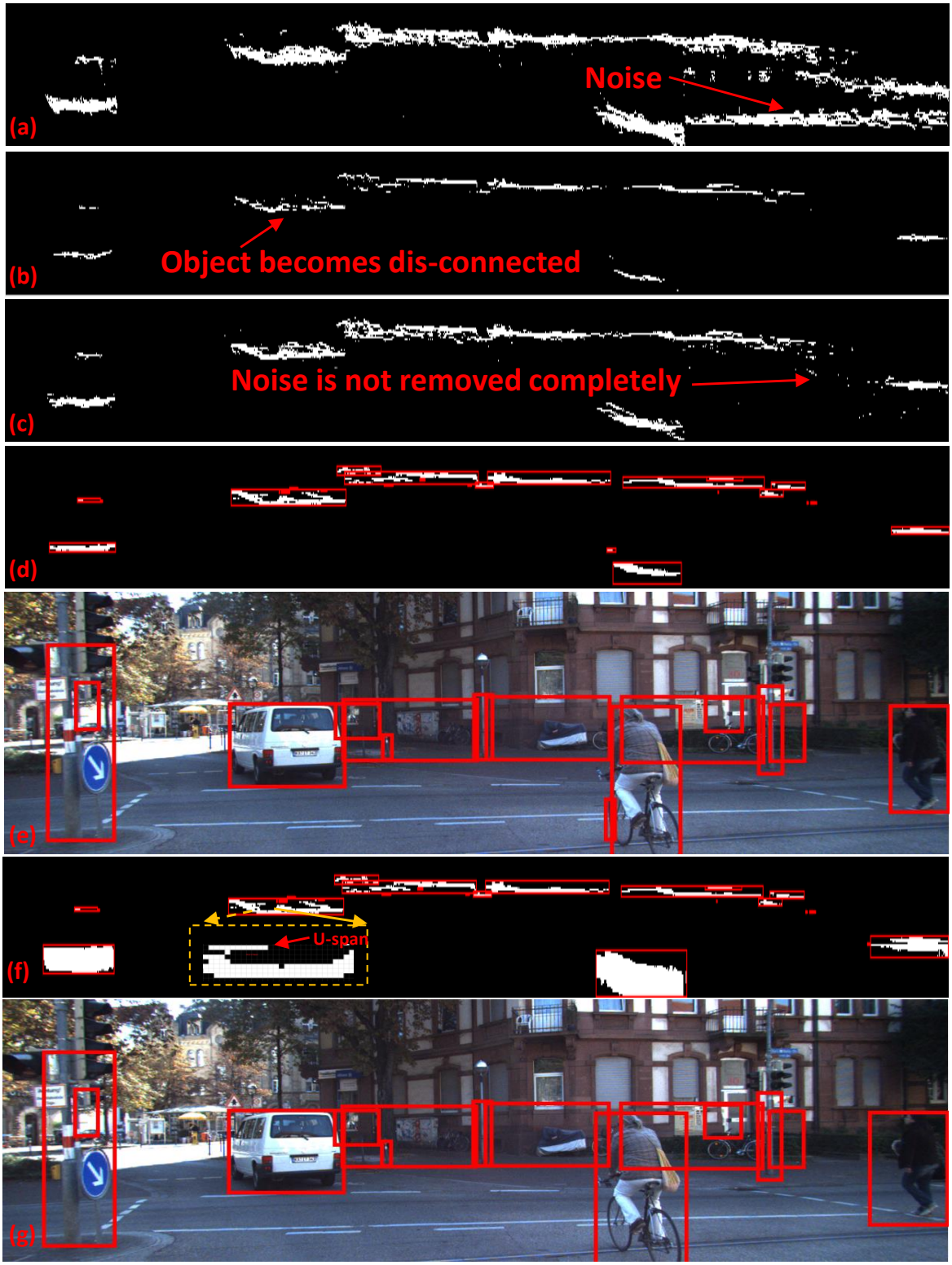


Fig. 5.3 Segmentation of SOI for the example given in Figure 5.2:(a) original u-disparity; (b) with a large threshold, single object is divided into parts; (c) with a small threshold, noise is not removed completely; (d) without applying any compensation, the thresholding operation divides the near bicyclist into two parts and the corresponding obstacle detection results are shown in (e); (f) adaptive connected component labeling technique based segmentation of the final strong u-peak regions into clusters, which are generated by applying box filter for compensation and using the proposed hysteresis thresholding technique; yellow inset is an enhanced visualization of a u-span; (g) obstacles are detected finally.

adopted when generating histogram. The techniques contributed to the establishment of the object appearance model can be over-viewed in Figure 5.1 and will be described in detail in the following sub-sections.

Listing 5.1 Obstacle Detection

Input: Disparity map $disMap$;
Road map $roadMap$;
Thresholds $threshold\_SOI\_x, threshold\_SOI\_y, threshold\_SOI\_z$;
Thresholds min_height,$min\_\Delta x$, $d\_reference$;
Ratio $ratio$.
Output: A set of obstacles $\{obstacle\}$;
/* SOI Generation*/
1: $SOI = disMap$;
2: **for** each point (u,v) in $disMap$ **do**
3: **if** $roadMap(v,u) == 1$ or $Eq.5.10 \sim Eq.5.12$ are not satisfied **then**
4: $SOI(v,u) = 0$;
5: **end if**
6: **end for**
/*SOI Segmentation*/
7: $udisImg\_soi = GUI(SOI)$;
8: Apply box filter to the bottom part of $udisImg\_soi$;
9: Applying hysteresis thresholding technique to $udisImg\_soi$;
10: Generate a set of $\{u\_span\}$ by clustering the continuous *strong u_peak* in the same row in $udisImg\_soi$;
11: **for** each $u\_span[u\_left, u\_right, u\_d]$ **do**
12: Compute the connectivity $Rect[u\_left - u\_neigh, u\_right + u\_neigh, u\_d + 1, u\_d + d\_neigh]$;
13: **end for**
14: Group $\{u_span\}$s into set of clusters $\{u\_cluster\}$ by applying the adaptive connected component labeling technique;
/*Determination of the Bounding Box for Obsacles*/
15: **for** each $u\_cluster \in \{u\_cluster\}$ **do**
16: create a new object $obstacle$;
17: determine its bounding box for $obstacle$;
18: **end for**

5.2.2.1 Utilizing L*a*b* Color Space

There are many ways to encode color. Compared to the popular *RGB* color model, *L*a*b** color (with dimension L^* for lightness, a^* and b^* for the color-opponent dimensions) is closer to human visual perception [327]. In particular, the L^* component closely matches human perception of brightness. As shown in Figure 5.4(a), two patches with the same size

and texture but subjected to different illumination are sampled from the back of the bicyclist. The histogram of the *RGB* and *L*a*b** color components for each patch is depicted in Figure 5.4(b). Figure 5.4(b) clearly illustrates that when illumination changes, the corresponding histogram change drastically for all of the three components of *RGB* color. On the other hand, the histograms for a^* and b^* component are stable, and only L^* component is affected. This means that the *RGB* color space is sensitive to illumination change while the a^* and b^* components of *L*a*b** are insensitive to illumination change. This motivates us to construct the color histogram in *L*a*b** color space. The number of bins for L^* component is only half of those required for a^* and b^* components. By doing this, the interference from illumination change is notably mitigated and robustness to illumination change are therefore increased.

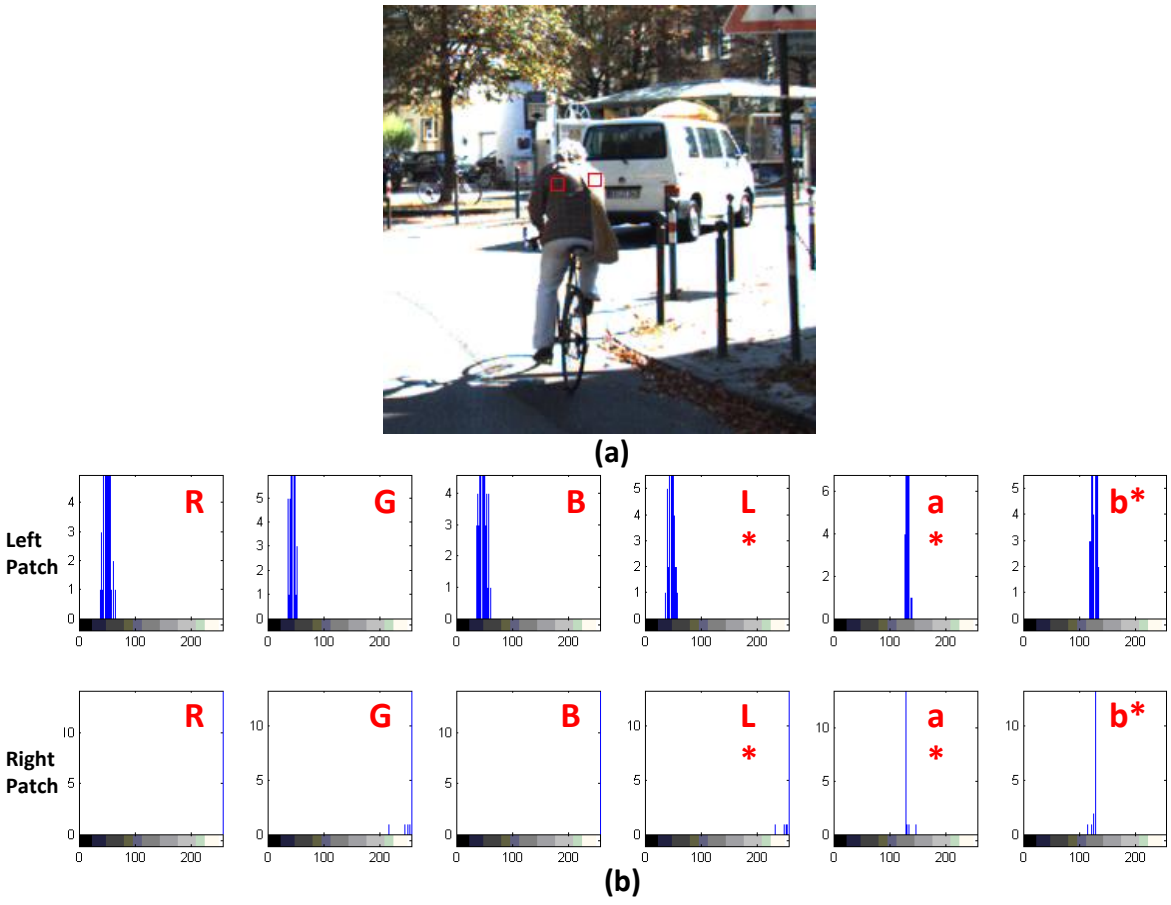


Fig. 5.4 Color histogram distributions for the same object in RGB and L*a*b* color spaces: (a) Two patches with same size and texture but subjected to different illumination are sampled from the back of the bicyclist. (b) Top row shows the histograms for R, G, B, L*, a*, b* color component corresponding to the left patch. Bottom row shows the histograms for R, G, B, L*, a*, b* color component corresponding to the right patch.

5.2.2.2 Excluding Background Information

When building the histogram for a kernel based model, the interference from the background is another big concern. The inclusion of background pixels will result in inconsistency in the histograms of the same object when the background varies across frames.

In order to overcome this problem, the depth information is exploited to exclude the background pixels when constructing histograms. This is possible as generally, obstacles and background are associated with different depth values. In addition, the corresponding depth range of the obstacles are made available during obstacle detection as discussed in section 5.2.1. Therefore, as illustrated in Figure 5.5(c), instead of considering all the pixels inside the bounding box, only pixels within the bounding box whose depth are in the range of the corresponding obstacle will contribute to the generation of color histogram for the obstacle. This strategy can effectively enhance the distinctiveness of the obstacle's appearance model.

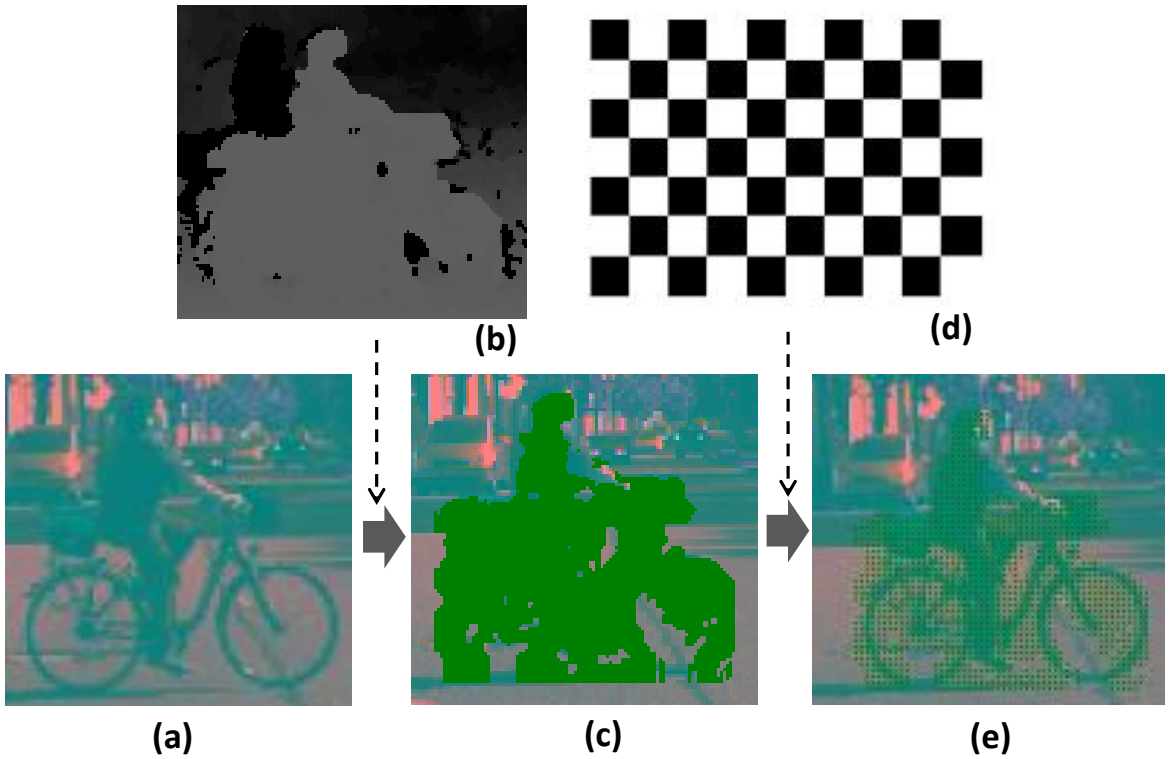


Fig. 5.5 Illustration of appearance model setup: (a) a L*a*b patch corresponding to one obstacle; (b) the corresponding disparity map; (c) background pixels are excluded; (d) a chessboard pattern; (e) only the pixels that do not belong to background and are not masked by the chessboard pattern will contribute to the final histogram construction.

5.2.2.3 Sparse Sampling

The stereo vision community [328–330] in general accepts the notion that the sparse correlation window configuration presents low computational complexity but is enough to provide good correlation accuracy. To test the effect of a sparse correlation window, the author in [331] studied its effect on SAD, one of the most common correlation methods. In the experiment, various sparse window configurations as illustrated in Figure 5.6 were tested. Table 5.1 shows the correlation accuracy for the full window and all the sparse windows configuration. It can be observed that the accuracy for sparse window with 50% cover is only 0.42% lower than the full window for the window size 13*13. When the sparseness increases, the accuracy decreases. The author has made further changes to the window size and observed the resulting effects. As illustrated in Figure 5.7, sparse sampling on a small window size leads to detrimental results. This is due to the reason that insufficient pixels are sampled for computing the similarity measure. However, when the window size increases to 21, the difference between the correlation accuracy of the sparse window and the full window can be neglected.

Inspired by this idea, a chessboard pattern based sampling technique when constructing the histogram is constructed. The idea of sparse sampling technique is illustrated in Figure 5.5(d) and (e). Only pixels that do not belong to background and are not masked by the chessboard pattern will contribute to the final histogram construction. The sparse window with 50% cover reduces the number of pixels in the window by half, resulting in approximately 50% less computations.

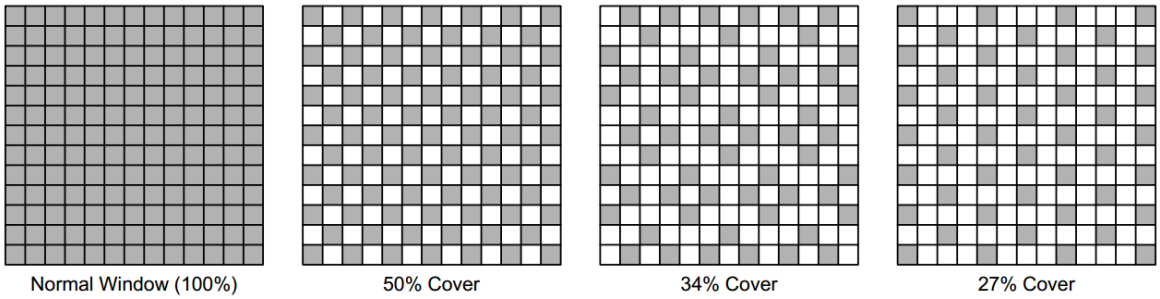


Fig. 5.6 Sparse 13*13 correlation windows of various densities. Only the darkened pixels are included in the similarity measure computation. Figure from [331].

5.2.2.4 Similarity Measure for Data Association

The histogram intersection distance is utilized to measure the similarity between two given normalized *L*a*b** histograms. Color histogram based data association works well for most

Table 5.1 Average correlation accuracy for sparse 13*13 SAD. Figure from [331].

Window	Average Accuracy	Degradation
Full(100%)	85.72%	-
50% Cover	85.36%	0.42%
34% Cover	84.82%	1.05%
27% Cover	84.44%	1.50%

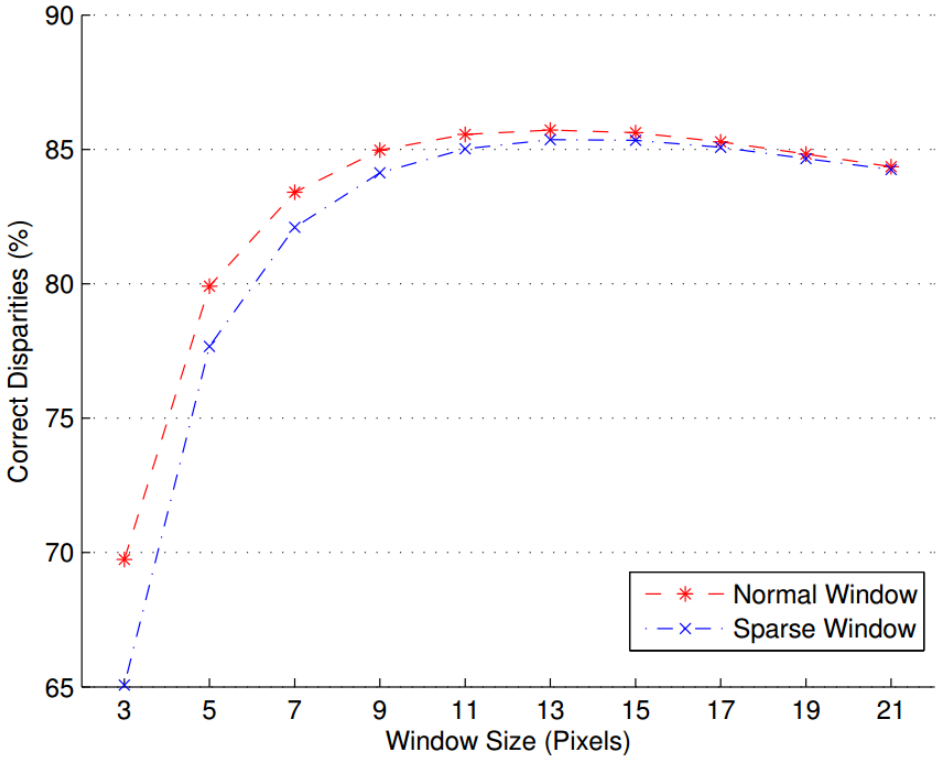


Fig. 5.7 Correlation accuracy comparison between normal (full) window and sparse window with 50% cover. Figure from [331].

cases. However, it has limitation in certain cases. Two different objects are likely to share the same color distribution, which may result in identical color histograms for the two objects. On the other hand, the same object may associate different color histogram in continuous frames due to the varying lighting conditions. Therefore, the color cue based object similarity measure will be enhanced by fusing the motion and spatial distance cues as discussed in the following.

In Chapter 4, a set of sparse feature points are extracted and tracked to estimate the ego-motion. As illustrated in Figure 5.8, the optical flows resulting from the set of tracked feature points encode the motion of the obstacles and can help in distinguishing the obstacle from still objects (such as trees, lamp posts, etc.) and other moving objects with different moving direction and speed. Similarity measure that takes into account the encoded motion information will be then more distinctive.

In addition, the similarity between two obstacles is further weighted by their spatial distance. Obstacles in the scene can be still or in small or large motion. However, there is an upper bound to the speed of obstacles due to their physical limitation. Therefore, a weighting factor based on the distance from the camera is introduced to adjust the final similarity score. If the distance between two obstacles is larger than τ , they cannot be the same obstacle in two consecutive frames and the similarity measure between them is therefore directly set to 0.

The final similarity measure for two obstacles is shown in Eq. 5.20:

$$similarity_1(\alpha,\beta) = \sum_{i=1}^{n} min(\alpha_i,\beta_i) \tag{5.18}$$

$$similarity_2(\alpha,\beta) = similarity_1(\alpha,\beta) * (1 + MatchedPoints * 0.5) \tag{5.19}$$

$$similarity(\alpha,\beta) = \begin{cases} 1.5 * similarity_2(\alpha,\beta), & dist < 0.5 * \tau \\ similarity_2(\alpha,\beta), & 0.5 * \tau < dist \leq \tau \\ 0, & dist > \tau \end{cases} \tag{5.20}$$

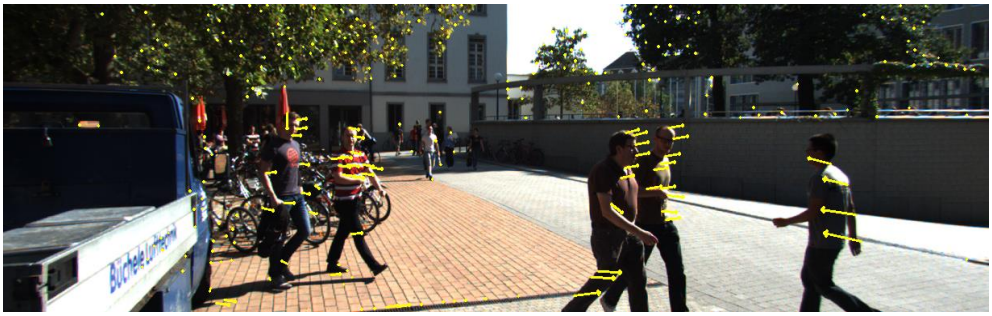
where α and β are two normalized $L*a*b*$ color histograms for the two obstacles. $similarity_1(\alpha,\beta)$ is the histogram intersection distance between α and β. *dist* refers to the distance between the two obstacles. τ is a predefined threshold. $MatchedPoints$ repre-

sents the number of tracked feature points falling into the regions corresponding to the two obstacles.

The pseudo code for the proposed method to built the appearance model for obstacle is presented in Listing 5.2.

(a)

(b)



(c)

Fig. 5.8 Point feature correspondences encode motion information: A set of feature points in (a) and their correspondences in (b), which is an intermediate result of the modified KLT method in Chapter 4. (c) the corresponding optical flows encode the motion information for the obstacles.

Listing 5.2 Appearance Model Setup

Input: Obstacle *obstacle*;
Original color image *srcImg*;
Number of bins for the color histogram $bin1, bin2, bin3$.
Output: Appearance model *hist* for *obstacle*;

```
1: srcImg_Lab = RGB2Lab(srcImg);
2: hist = zeros(bin1, bin2, bin3);
3: for j = obstacle.top : 2 : obstacle.bottom do
4:    for i = obstacle.left : 2 : obstacle.right do
5:       if disMap[j][i] <= obstacle.dmax and disMap[j][i] >= obstacle.dmin then
6:          L_component = get_L_component(srcImg_Lab[j][i]);
7:          a*_component = get_a*_component(srcImg_Lab[j][i]);
8:          b*_component = get_b*_component(srcImg_Lab[j][i]);
9:          binid_l = L_component/bin1;
10:         binid_a* = a*_component/bin2;
11:         binid_b* = b*_component/bin3;
12:         hist(binid_l, binid_a*, binid_b*) = hist(binid_l, binid_a*, binid_b*) + 1;
13:      end if
14:   end for
15: end for
16: apply L1 normalization to hist.
```

5.2.3 Online Multi-Object Tracking Framework

Given a set of tracks $\mathscr{T} = \{tr_i\}$ identified from earlier frames and a set of detected obstacles $\mathscr{D} = \{de_j\}$ in current frame, the whole tracking framework as shown in Figure 5.1 is presented as follows.

Step 1 - Similarity Computation: Compute the similarity matrix $\mathbf{S} = \{similarity_{ij}\}$ between the tracks $\mathscr{T}$ and detections $\mathscr{D}$ using the metric defined in Eq. 5.20. $similarity_{ij}$ refers to the similarity value between track tr_i and detection de_j.

Step 2 - Tracks Assignment: Assign the detections $\mathscr{D}$ to the tracks $\mathscr{T}$ by solving a bipartite matching problem with the Hungarian method [332].

Step 3 - State Management: The total states for tracks can be: *stable, new, lost*. Through the maintenance of these three states, the context of the scene is well understood.

After *Step 2*, there are three types of assignment: tracks $\mathscr{T}_1 = \{tr_i^1\}$ are assigned with detections $\mathscr{D}_1 = \{de_j^1\}$, unassigned tracks $\mathscr{T}_2 = \{tr_i^2\}$, and unassigned detections $\mathscr{D}_2 = \{de_j^2\}$.

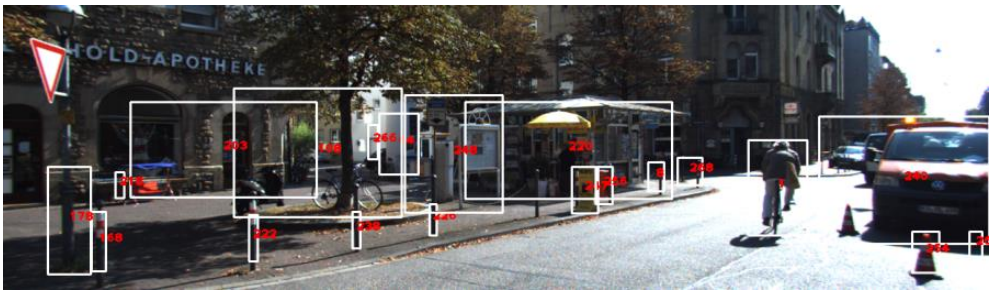
1) For each track tr_i^1 in $\mathscr{T}_1$, its state is updated as *stable*.

2) For each track tr_i^2 in $\mathscr{T}_2$, the following two cases are checked in the order listed: tr_i^2 is merged with other track; and tr_i^2 is lost.

There are several reasons as to why a track is unable to find its correspondence in current frame. Firstly, the track can be merged with other track in the current frame due to close proximity or the inaccuracy in obstacle detection. Secondly, the corresponding object physically disappears in the current frame. Measures should be taken to differentiate these two cases. An example for the first case is given in Fig.5.9. Fig.5.9(a) shows the detection and tracking results for Frame 0085, where the bicyclist with id 1 and the vehicle with id 240 are separated and treated as individual tracks. When the new frame 0086 comes, the road surface is texture-less and the corresponding disparity map as shown in Fig.5.9(b) is inaccurate. This causes the bicyclist and the vehicle to be detected as one entity as shown in Fig.5.9(c). As illustrated in Fig.5.9(d), if no countermeasure is taken, the bicyclist with track id 1 gets unassigned and the vehicle with track id 240 will be updated wrongly with an new object as a result of merging the bicyclist and vehicle.

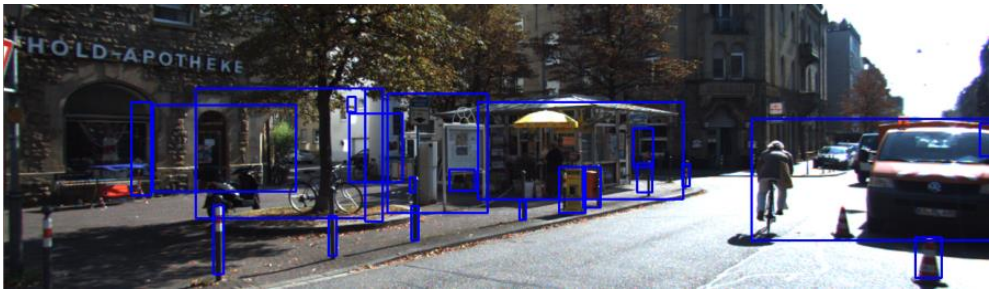
In order to check whether tr_i^2 falls under the first case mentioned above and to perform the correction if it happens, the corresponding detection de_j where the similarity value between tr_i^2 and de_j is highest is found. The corresponding track that de_j is assigned to is denoted as tr_m^1. If $similarity(tr_i^2, de_j)$ and $similarity(tr_m^1, de_j)$ are similar and the objects corresponding to tr_i^2 and tr_m^1 are in close proximity, they are deemed to have merged. At this time, tr_i^2 is assigned with a detection de_{new}, which is the predicted position of tr_i^2 in current frame. de_j is corrected by excluding the part that corresponds to de_{new}. tr_i^2 is updated with state *stable* and added to $\mathscr{T}_1$. de_{new} is added to $\mathscr{D}_1$. As illustrated in Fig.5.9(e), with the proposed correction strategy, the bicyclist with track id 1 and the vehicle with track id 240 are correctly tracked. If the first case doesn't happen, tr_i^2 belongs to the second case where the corresponding obstacle disappears in the current frame. For this case, the state of tr_i^2 is labeled as *lost*.

3) For each detection de_j^2 in $\mathscr{D}_2$, create a track with state *new*. Although it is beyond the scope of this thesis, it is worth pointing out that in the case where the object's type needs to be identified, an object classification process can be deployed at the time it is newly detected. By doing this, object classification process needs to be conducted only once for the entire life-span for the object in the scene. This leads to significant computational savings compared to the strategy where object needs to be classified in every frame.

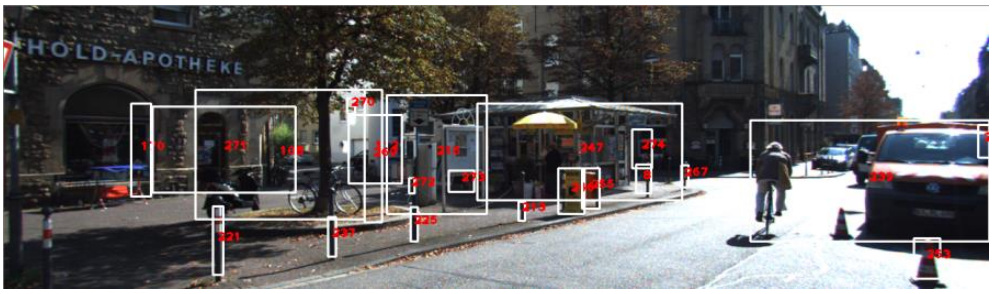


(a)

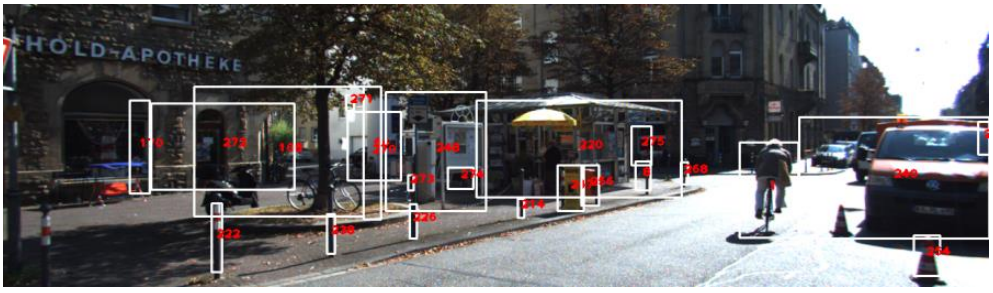
(b)



(c)



(d)



(e)

Fig. 5.9 The tracking process helps to correct the inaccuracy made in the obstacle detection stage. For the detailed description of this figure, please refer to the text in Section 5.2.3

Step 4 - Appearance Model Update: Update the appearance model for the tracks with *stable* and *new* state. Delete the tracks that have been lost for *threshold_n* frames.

Since the proposed multi-object tracking method takes into account only the information of previous frames for inference at any time instance during tracking, it is an online tracking framework [333]. This is contrary to the offline tracking framework, which processes image sequences in a batch mode, that is, image frames from the future time steps are also utilized to solve the data association problem in the current frame. The pseudo code for the proposed online multi-object tracking framework is presented in Listing 5.3.

5.3 Experimental Evaluation

5.3.1 Experimental Setup

5.3.1.1 Benchmark

The well-known KITTI tracking dataset [26] has been chosen to evaluate the proposed obstacle detection and tracking algorithm. The KITTI tracking dataset consists of 21 training sequences and 29 test sequences, which cover various challenging road scenarios. Obstacles on the road include vehicle, pedestrian, bicyclist, traffic light, barrier, etc. Inconsistent illumination is very evident in this dataset. Objects are frequently occluded by others. Also, the scale and pose of the obstacles change drastically across frames. Some samples of the KITTI tracking dataset are illustrated in Figure 5.10.

Fig. 5.10 Some samples of the KITTI tracking dataset.

Listing 5.3 Online Multi-Object Tracking System

Input: A set of tracks $\mathscr{T} = \{tr_i\}$ identified from earlier frames;
A set of obstacles $\mathscr{D} = \{de_i\}$ detected in current frame;
Thresholds $threshold\_c, threshold\_d, threshold\_n$.

Output: A set of tracks $\mathscr{T} = \{tr_i\}$ updated in current frame.

1: Compute similarity matrix $\mathbf{S} = \{similarity_{ij}\}$;
2: $[\mathscr{T}_1, \mathscr{T}_2, \mathscr{D}_1, \mathscr{D}_2] = Hungarian(\mathbf{S})$;
3: **for** each $tr_i^2 \in \mathscr{T}_2$ **do**
4: find de_j and tr_m^1;
5: **if** $|similarity(tr_i^2, de_j) - similarity(tr_m^1, de_j)| < threshold\_c$ && $distance(tr_i^2, tr_m^1) < threshold\_d$ **then**
6: de_{new} = predict tr_i^2 in current frame;
7: $de_j = de_j - de_{new}$;
8: tr_i^2.state = $stable$;
9: $tr_i^2.hist = de_{new}.hist$;
10: $tr_i^2.life = tr_i^2.life + 1$;
11: $tr_i^2.lostlife = 0$;
12: **else**
13: tr_i^2.state = $lost$;
14: $tr_i^2.lostlife = tr_i^2.lostlife + 1$;
15: **if** $tr_i^2.lostlife > threshold\_n$ **then**
16: delete $tr_i^2 from \mathscr{T}_2$;
17: **end if**
18: **end if**
19: **end for**
20: **for** each $tr_i^1 \in \mathscr{T}_1$ and $de_i^1 \in \mathscr{D}_1$ **do**
21: $tr_i^1.state = stable$;
22: $tr_i^1.life = tr_i^1.life + 1$;
23: $tr_i^1.lostlife = 0$;
24: $tr_i^1.hist = de_i^1.hist$;
25: **end for**
26: **for** each $de_j^2 \in \mathscr{D}_2$ **do**
27: create a new track tr_{new};
28: create a new track $tr_{new}.hist = de_j^2.hist$;
29: $tr_{new}.state = new$;
30: $tr_{new}.life = 1$;
31: $tr_{new}.lostlife = 0$;
32: **end for**

5.3.1.2 Baseline Algorithm

Andreas Geiger *et al.* has proposed an object tracker in his recent work [27, 321, 322]. This object tracker follows the track-by-detection framework and therefore can also be divided into two parts: object detection and objects association across frames.

In order to detect objects in an image, Andreas Geiger adopts the well-known object detector named Deformable Parts Model (DPM) [152]. DPM detects object based on mixtures of multi-scale star-structured deformable part models. Each of the part-based models enriches the Dalal-Triggs HOG model [127] by defining a "root" filter plus a set of part filters and associated deformation models. The part filters capture features at twice the spatial resolution relative to the features captured by the root filter. DPM has been reported to be able to tackle the intra-category diversity problem in object detection and achieve good detection results in the challenging PASCAL object detection dataset [152].

Once the objects are detected in every frame, association of the objects across frames is conducted directly in the image domain rather than in the 3D domain. All the detected objects in the current frame are associated with the existing tracklets, i.e. object tracks, by solving the bipartite matching problem using the Hungarian algorithm [332]. The similarity between objects is computed by fusing both geometry and appearance cues of the object. That is, the normalized cross-correlation (NCC) based appearance similarity is weighted by the bounding box intersection over union score. In order to compensate for the problems like imperfections of the object detector or occlusion and increase the object association accuracy across frames, a second stage trackelet-to-tracklet association is conducted. The Hungarian algorithm is employed again to assign one tracklet to another. Each entry of the association matrix refers to a pair of tracklets within the whole sequence. Bounding boxes of each tracklet are extrapolated linearly to predict the bounding boxes of the other tracklet and returns the mean of the normalized prediction errors with respect to the bounding box location, width and height. Object appearances similarity via the normalized cross-correlation score is computed over all possible combinations of object detections.

5.3.1.3 Implementation Details

The proposed algorithm is implemented on a PC platform Hp Z420 Workstation, where the processor is Intel(R) Xeon(R) CPU E5-1650 v2 3.50 GHz with 16GB memory. All the codes are developed in C++ in the Visual Studio 2012 running in Windows 7. For the baseline algorithm, the code released by the authors in their website[1] is directly used.

[1]http://www.cvlibs.net/software/trackbydet/

5.3.2 Accuracy Evaluation

A. Obstacle Detection

Before detecting specific objects, the DPM object detector needs to be trained to obtain the corresponding knowledge for the object in advanced. However, in reality, the driving environment is very complex. Obstacles in the scene include not only the common objects like vehicles, pedestrians, bicyclists but also some unexpected objects like barriers. As illustrated in Figure 5.11, the baseline algorithm only detects the vehicles and is therefore unable to meet the requirements of collision avoidance. On the contrary, the proposed algorithm detects the objects based on the geometrical topology of the scene and hence is able to detect not only vehicles but also pedestrian, traffic sign, traffic light, traffic barrier and tree simultaneously. Additional detection results from the proposed obstacle detection algorithm in diverse traffic scenarios are provided in Figure 5.12.

Although it is not within the scope of this thesis, it is worth pointing out that another benefit of the proposed obstacle detection algorithm is that the results of obstacle detection can then be used as the initial object hypotheses that are fed into object classification operation in the applications where the object class is needed. By doing this, the number of candidates that needs to be classified is significantly reduced. This is in contrast to the baseline algorithm, which generates the initial object hypotheses by shifting the detection window over the whole image at various locations and scales in order to detect the cars in an image.

B. Frame Association

In order to exclude the effect of object detection and evaluate the accuracy performance for the object appearance model for both of the proposed and baseline algorithms, the proposed and baseline tracking algorithms are fed with the same inputs. The KITTI tracking dataset [26] provides the ground truth for objects with class 'Car', 'Van', 'Truck', 'Pedestrian', 'Person_sitting','Cyclist','Tram','Misc' or 'DontCare'. Therefore, the object appearance models from the proposed and baseline tracking algorithm will be evaluated with these ground truth detections in the following.

The appearance model adopted in the baseline algorithm is based on the geometric distance weighted NCC similarity metric. However, NCC similarity metric is sensitive to a lot of factors like illumination change, occlusion, scale change, background noise, etc. In addition, the geometric distance is just based on the simple u-v 2D image space, which doesn't reflect

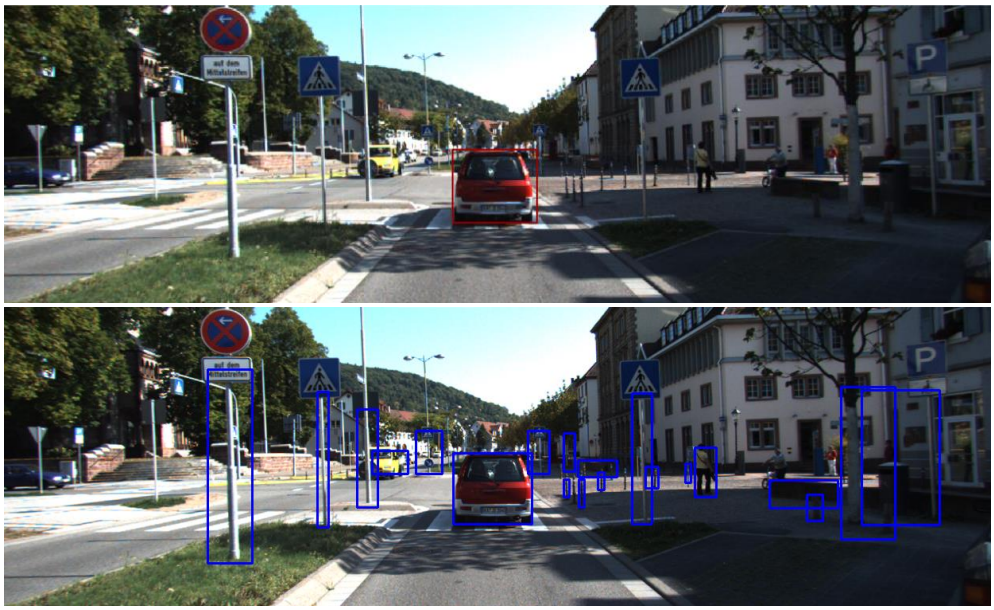


Fig. 5.11 Qualitative comparison between the baseline and proposed object detection methods: (a) The baseline algorithm only detects the vehicles. (b)The proposed algorithm is able to detect not only vehicle but also pedestrian, traffic sign, traffic light, traffic barrier, tree simultaneously.

the true distance in 3D Euclidean space correctly. This easily leads to problems like track fragmentation and track switching. One example is illustrated in Figure 5.13. Figure 5.13 (a1) and (a2) are the tracked results from the baseline algorithm with only one association stage. It can be seen that both the van and bicyclist are tracked wrongly. Figure 5.13 (b1) and (b2) are the tracked results from the baseline algorithm with two association stages. After the correction of the second stage, both van and bicyclist are tracked correctly. Although a second stage tracklet-to-tracklet association helps to reduce the tracking error in some degree, it still fails in some other cases since the similarity measure adopted in this stage is still based on the normalized cross-correlation similarity measure. This is illustrated in Figure 5.14. Figure 5.14 (a1) and (a2) and Figure 5.14 (b1) and (b2) correspond to the tracking results from the baseline algorithm with only one association stage and two association stages respectively. It can be observed that the car is tracked wrongly in either of the cases.

On the other hand, thanks to the proposed distinctive object appearance model, the proposed obstacle tracking method is able to track the objects correctly as illustrated in Figure 5.13 and Figure 5.14.

Fig. 5.12 Additional detection results using the proposed obstacle detection method in diverse traffic scenarios. It can be observed that the proposed method is capable of robustly detecting various types of obstacles in the scene.

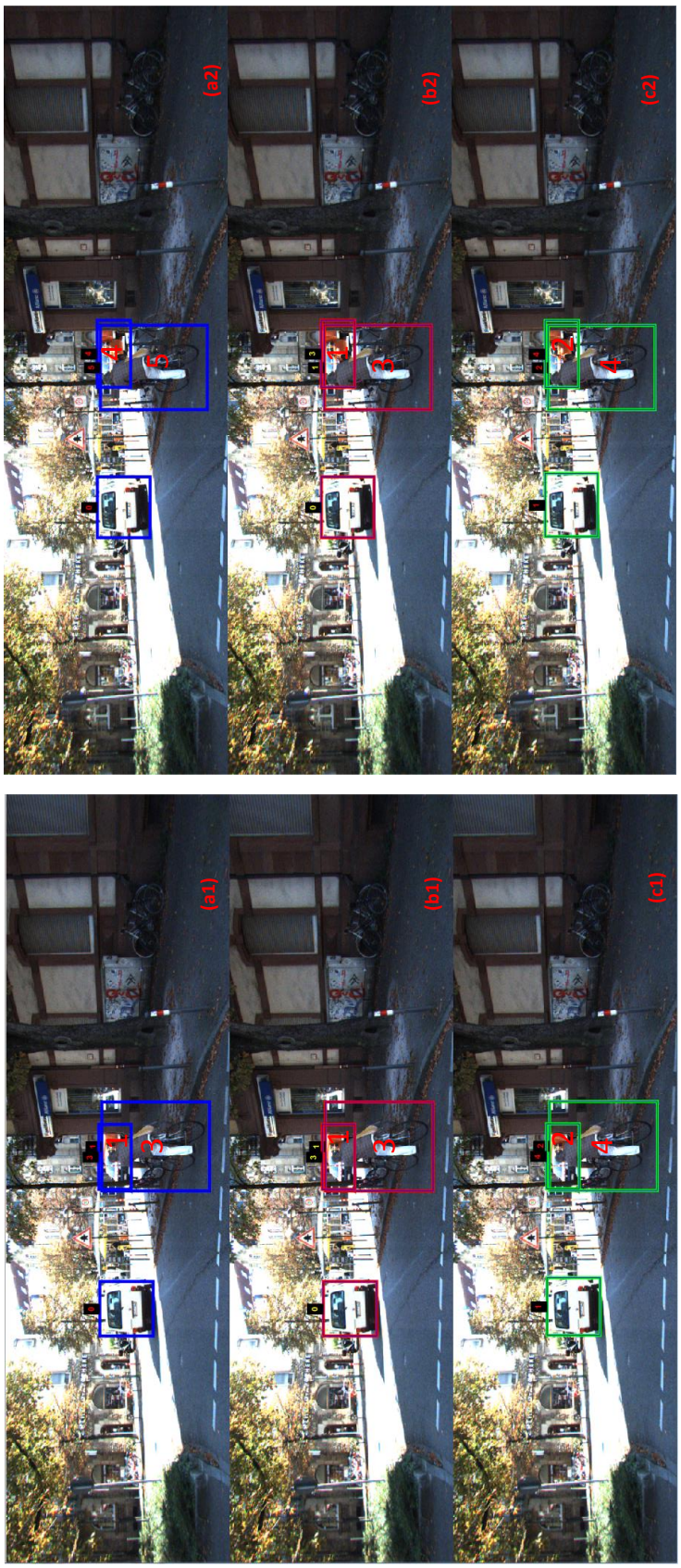


Fig. 5.13 Qualitative comparison between the baseline and proposed object tracking methods in scenario I: the left column corresponds to the previous frame and the right column corresponds to the current frame; (a1) and (a2) are the tracked results from the baseline algorithm with only one association stage. Both van and bicyclist are tracked wrongly. (b1) and (b2) are the tracked results from the baseline algorithm with two association stages. After the correction of the second stage, both van and bicyclist are tracked correctly. (c1) and (c2) are the tracked results from the proposed algorithm. Both van and bicyclist are tracked correctly by the proposed algorithm.

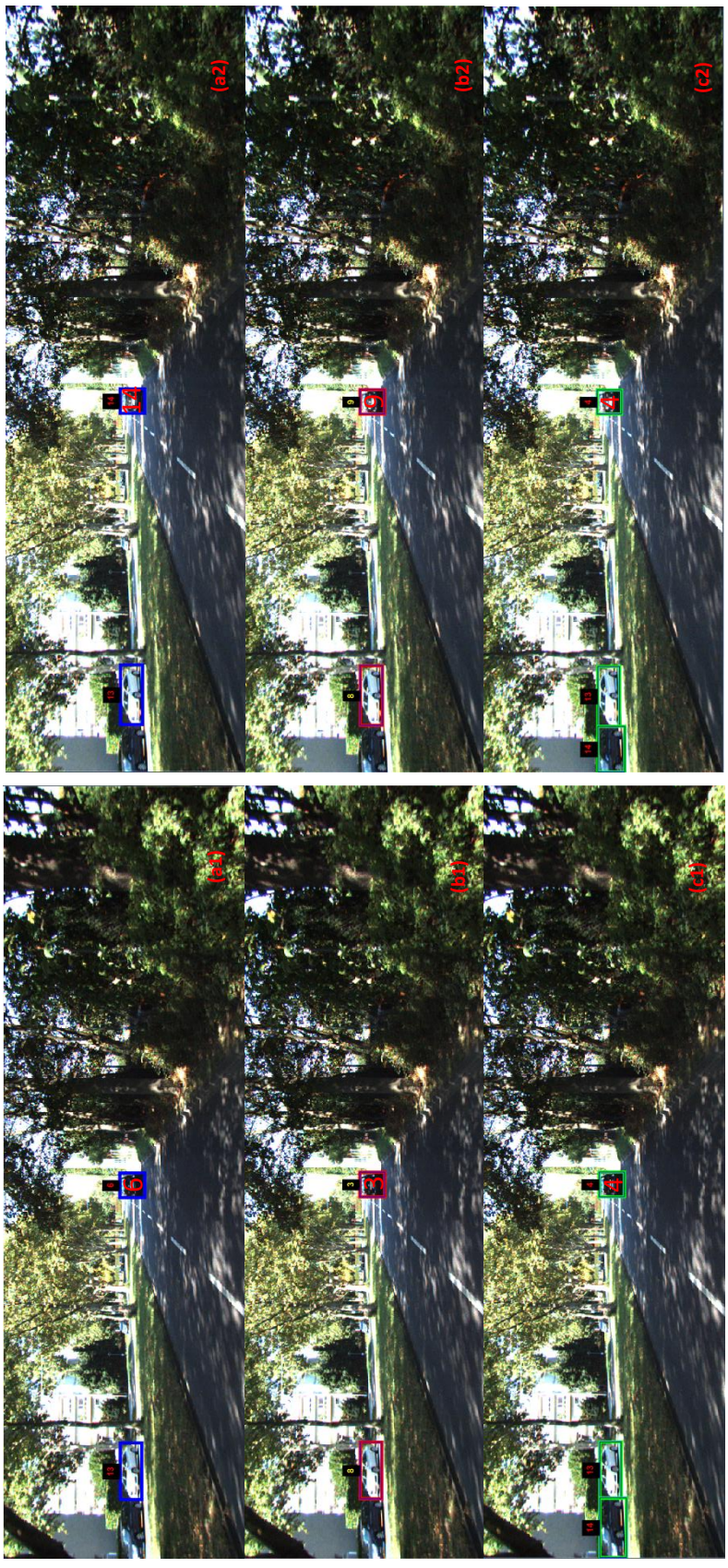


Fig. 5.14 Qualitative comparison between the baseline and proposed object tracking methods in scenario II: the left column corresponds to the previous frame and the right column corresponds to the current frame; (a1) and (a2) are the tracked results from the baseline algorithm with only one association stage. The car is tracked wrongly. (b1) and (b2) are the tracked results from the baseline algorithm with two association stages. Even though the second stage tracklet-to-tracklet association is enabled, the car is still tracked wrongly. (c1) and (c2) are the tracked results from the proposed algorithm. The car is tracked correctly by the proposed algorithm.

Table 5.2 Quantitative evaluation results in tracking accuracy.

Method	MOTA(%)	MOTP(%)	MT(%)	ML(%)	FM	IDS
Baseline	88.10	94.62	65.74	4.56	808	512
Proposed	95.11	98.24	99.78	0	736	731

Finally, an extensively quantitative evaluation between the proposed and baseline frame association algorithms is conducted. The popular CLEARMOT metrics, i.e. *Multiple Object Tracking Accuracy* (MOTA) and *Multiple Object Tracking Precision* (MOTP), proposed in [334] and additional metrics like *Mostly-Tracked* (MT), *Mostly-Lost* (ML), *Track Fragmentations* (FM), and *Identity Switches* (IDS) proposed in [335] are adopted. Different metrics provide different insights. For the detailed interpretation of these metrics, please refer to [334, 335]. The implementation of these evaluation metrics provided in the KITTI website [2] and their default parameter settings are adopted. The corresponding results are shown in Table 5.2. It is evident that the proposed tracking algorithm significantly outperforms the baseline algorithm.

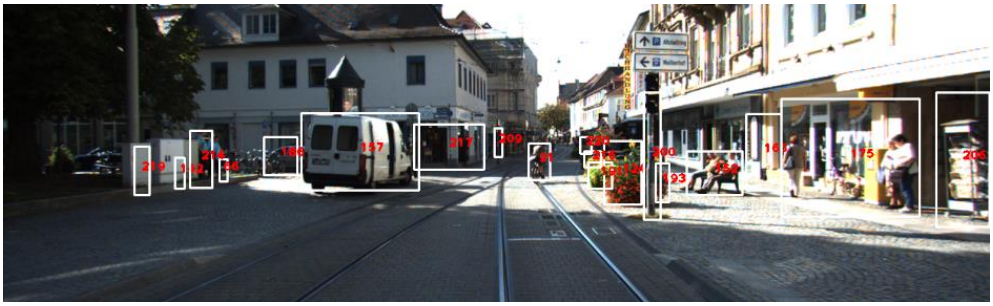
C. Online Multi-Obstacles Tracking System

Finally, a comprehensive qualitative evaluation of the proposed online multi-object tracking system is conducted. In the dataset used for the evaluation, obstacles on the road include vehicle, pedestrian, bicyclist, traffic light, barrier, etc. Inconsistent illumination is very evident in this dataset. In addition, objects are frequently occluded by others. Also, the scale and pose of the obstacles changed drastically across frames. In this section, the effectiveness of the proposed algorithm in overcoming these problems is demonstrated.

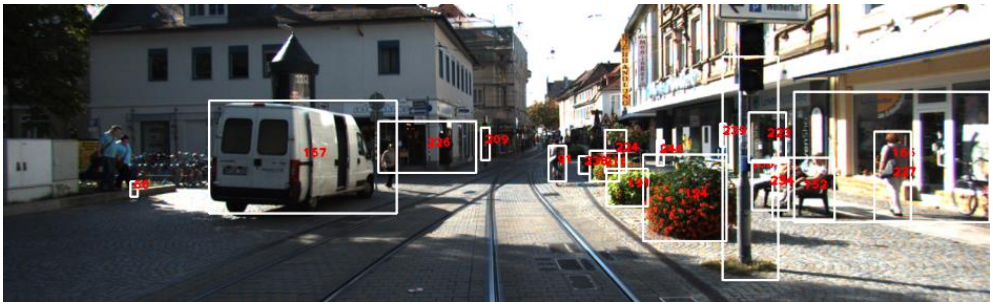
Figure 5.15 shows a busy road scenario. It is evident that not only common obstacles like pedestrian, vehicle, bicyclist but also unexpected ones like traffic light and flowerbed are simultaneously tracked. The obstacles can be standing still or subjected to motion. In Figure 5.16, a train appears in the scene. The scale of the train varies drastically over the frames. However, the proposed algorithm is still able to robustly track it. The dataset also contains scenarios where obstacles are occluded by others. For example, in Figure 5.17, the man in grey shirt with id 41 walks towards a group of two people, gets merged and occluded by them, and finally appears again. The man is correctly tracked by the proposed method throughout the entire course. The proposed algorithm is also insensitive to inconsistent illumination. Figure 5.18 illustrates a scenario where the illumination changes abruptly.

[2]http://www.cvlibs.net/datasets/kitti/eval_tracking.php

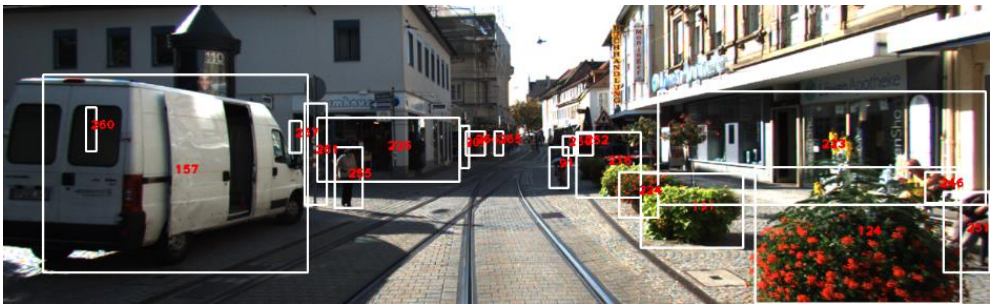
Although subjected to different illumination conditions, the cyclist is continuously tracked over frames. Therefore, the experimental results confirm that the proposed algorithm is capable of tracking obstacles in challenging conditions.



(a) Frame 0102



(b) Frame 0110

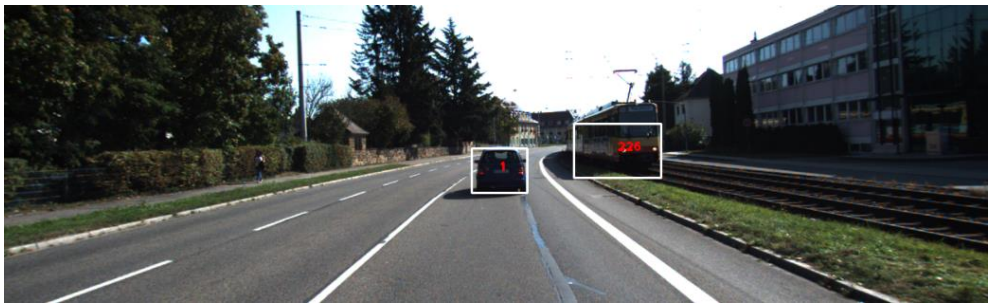


(c) Frame 0118

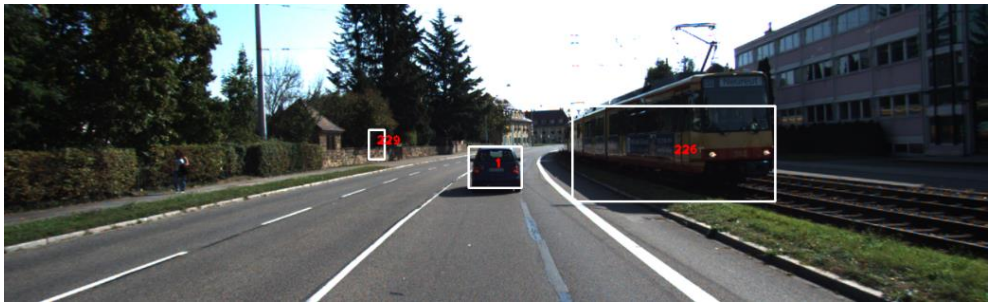
Fig. 5.15 Tracking results from the proposed tracking algorithm in busy road scenario: Not only common obstacles like pedestrian, vehicle and bicyclist but also unexpected ones like traffic light and flowerbed are simultaneously tracked. The obstacles can be standing still or subjected to motion.

5.3.3 Runtime Performance Evaluation

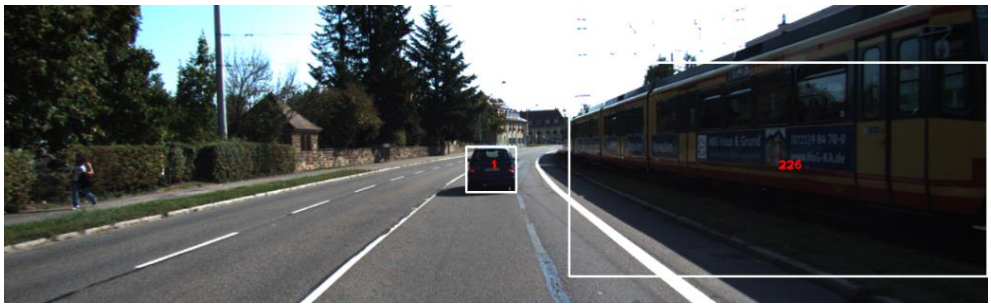
The author of the baseline algorithm has reported the computation time of their object tracking system in [27] and it is 4.34 second per frame on average. It is worth pointing out that only vehicle is of concern in their current system, and the computation time will further



(a) Frame 0210

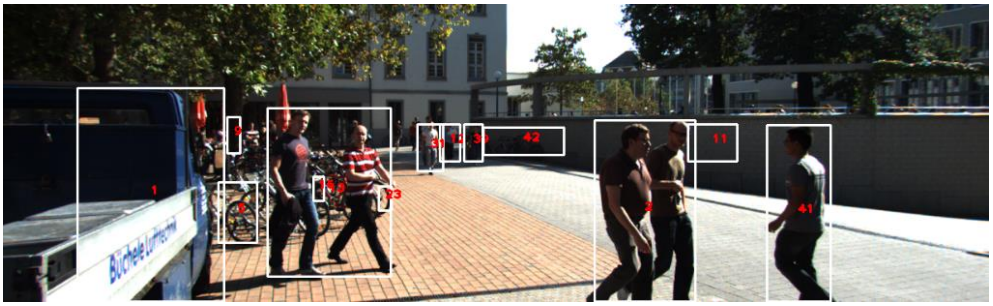


(b) Frame 0214

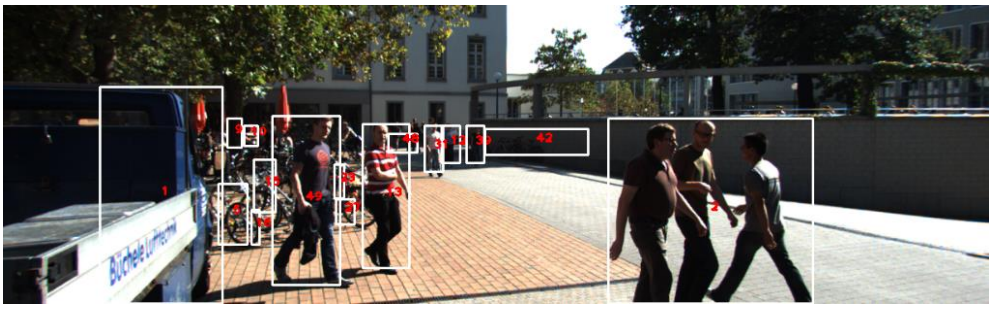


(c) Frame 0217

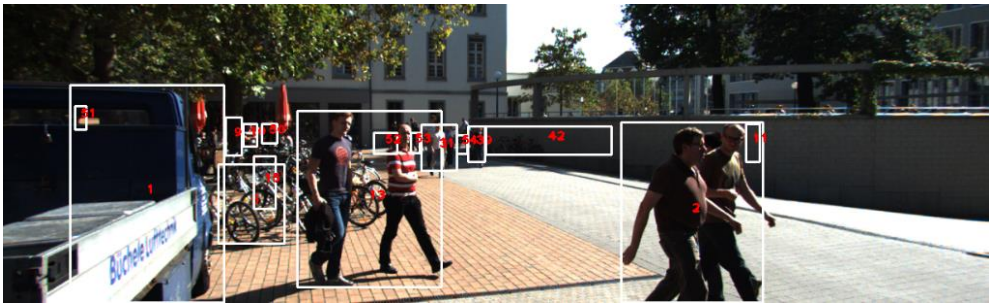
Fig. 5.16 Tracking results from the proposed tracking algorithm in scenario with large object scale change: The scale of the train varies drastically over the frames. But the train is tracked robustly.



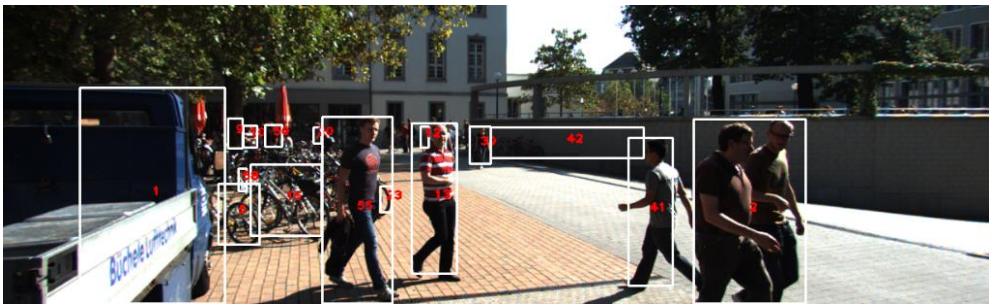
(a) Frame 0024



(b) Frame 0026

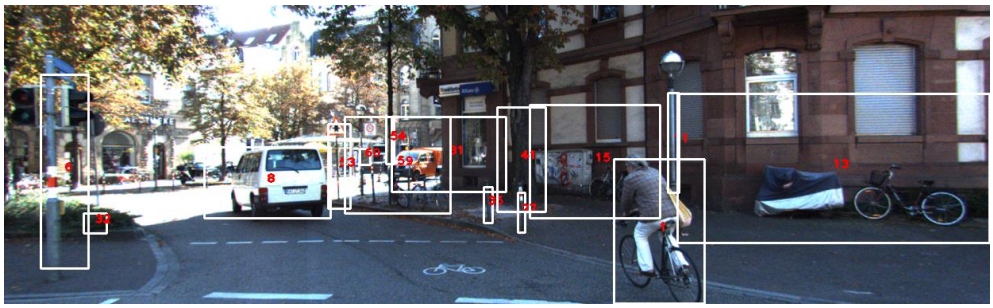


(c) Frame 0028

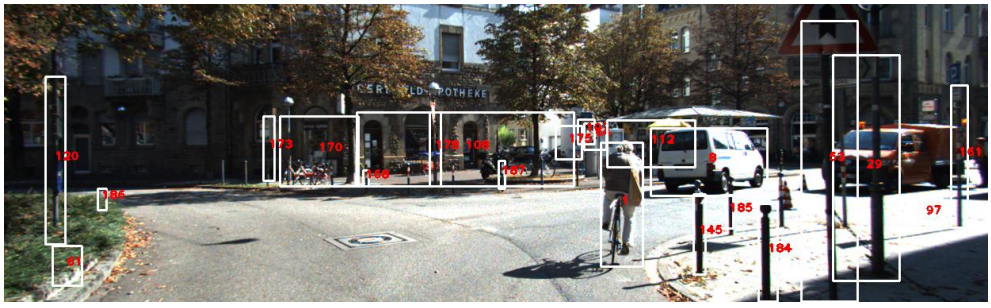


(d) Frame 0031

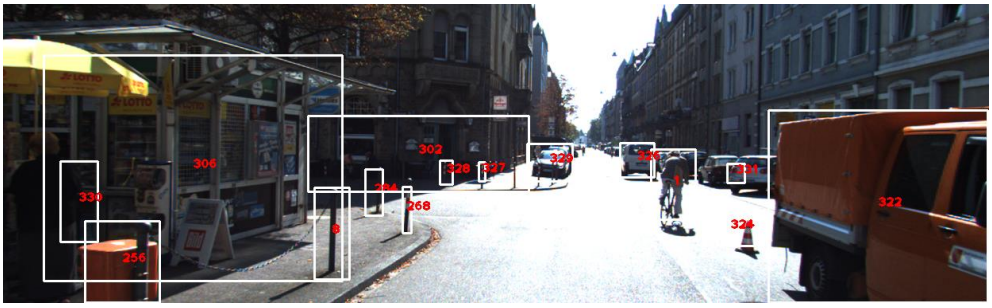
Fig. 5.17 Tracking results from the proposed tracking algorithm in the presence of occlusion: the man in grey shirt with id 41 walks towards others, gets merged and occluded by them, and finally appears again. The proposed method correctly tracks the man throughout.



(a) Frame 0019



(b) Frame 0059



(c) Frame 0106

Fig. 5.18 Tracking results from the proposed tracking algorithm in inconsistent illumination scenario: Although subjected to different illumination conditions, the cyclist is continuously tracked over frames.

Table 5.3 Runtime performance comparison between the baseline and proposed object detection and tracking algorithms.

Method	Detection (Second/Frame)	Tracking (Second/Frame)	Total (Second/Frame)	Platform
Baseline	3.88	0.46	4.34	CPU@2.67GHZ
Proposed	0.046	0.003	0.049	CPU@3.5GHZ

increase nonlinearly when the concerned object types extend to others e.g. pedestrian and bicyclists. This is due to the two reasons as follows. Firstly, for the detection part, a complex and time-consuming strategy is used to extract features for classification. In addition, it employs the sliding-window scheme where filters are applied at all positions and scales of an image and a large number of candidates are tested. Furthermore, this detection process needs to be conducted for every frame. Therefore, the DPM algorithm adopted in the baseline algorithm is unable to fulfill real-time requirements. Secondly, during the tracking process, in order to alleviate the effect of semi-occlusion and variance in object's appearance, a second stage trackelet-to-tracklet association is conducted. This is also a very time-consuming task that is required in the baseline algorithm. The baseline algorithm therefore has a very high computational complexity.

On the other hand, given the input color image and the corresponding disparity map, the proposed algorithm yields low computational complexity due to the following strategies. Firstly, obstacles are detected in the u-v-d image space. Space of Interest (SOI) is generated to reduce the search space of obstacle detection. The algorithm adopted to detect the road surface is lightweight [6]. Secondly, the generation of histogram is fast due to the sparse sampling technique. Therefore, the proposed algorithm only needs 0.046 second/frame for detection and 0.003 second/frame for tracking. The reported obstacle detection time 0.046 second /frame includes 0.0069 second for road surface detection (Chapter 3) and 0.0276 second for visual odometry (Chapter 4).

5.4 Summary

A robust and low complexity obstacle detection and tracking algorithm is proposed in this chapter. Using the widely-known and challenging benchmark, it has been demonstrated that the proposed obstacle tracking algorithm is capable of detecting and tracking obstacles in the presence of drastic scale change, partial occlusion and inconsistent illumination.

The proposed algorithm not only simultaneously detects and tracks common obstacles like vehicles, pedestrians, bicyclists but also unexpected ones like traffic lights, sign posts, barriers, and trees and so on. The obstacles can be standing still or in motion.

The robustness of the proposed algorithm is achieved by detecting obstacles in an optimized search space with adaptive hysteresis thresholding technique and adaptive connected component labeling technique and constructing a distinctive object appearance model utilizing several optimization strategies (i.e., L*a*b color histogram, background removal, motion and distance enhanced histogram similarity metric) for object association. In addition, In addition, the proposed obstacle detection and data association modules are integrated to form an online multi-object tracking framework in a robust way.

The proposed algorithm also yields low computational complexity as it incorporates strategies for fast obstacle detection in the u-v disparity space and the employment of chessboard pattern based sparse sampling technique. The proposed method has been shown to lend well for real-time realization with 20 fps.

In the following chapter, an efficient collision risk assessment module that relies on the tracked obstacles obtained from the work in this chapter and the estimated ego-motion state from the work in the previous chapter will be presented.

CHAPTER 6

RISK ASSESSMENT

Collision risk is defined as the likelihood and severity of damage that a vehicle of interest may suffer in the future [263]. The ultimate goal of collision avoidance system is to allow the driver to react in advance so that necessary measures can be taken in order to avoid collision. Hence, once obstacles are detected and tracked, it is necessary to assess the risk of collision between the ego-vehicle and each obstacle.

There are many traffic participants present in the traffic environment. These include not only vehicles and pedestrians but also traffic lights, sign posts, trees, etc. All of these traffic participants are referred to as obstacles. The obstacles can be moving in any direction or remaining still. On the other hand, the ego-vehicle is in the state of motion most of the time, but it may be stationary occasionally. Therefore, in order to assess the risk of the environment, it is necessary to build mathematical models which are able to predict how the scene evolves in the near future [263].

A literature review of existing works about risk assessment has been conducted in Section 2.5. It has been found that reliable collision risk assessment between the ego-vehicle and each obstacle in a complex urban environment still remains an unsolved problem.

This chapter is structured as follows: Section 6.1 proposes an Extended Kalman Filter (EKF) based motion trajectory prediction model for obstacles and Section 6.2 proposes an efficient

technique for collision prediction. This is followed by the description of the strategy for deriving the collision risk indicator in Section 6.3. A comprehensive evaluation of the proposed risk assessment method using the KITTI tracking dataset is conducted in Section 6.4. Section 6.5 summarizes this chapter.

In this chapter, a risk assessment method is proposed. This chapter is structured as follows: Section 6.1 presents the proposed risk assessment method. A comprehensive evaluation of the proposed risk assessment method using the KITTI tracking dataset is conducted in Section 6.2. Section 6.3 summarizes this chapter.

6.1 Proposed Algorithm

The proposed risk assessment method relies on the motion state of the ego-vehicle obtained from visual odometry module (Chapter 4), and obstacle tracks obtained from obstacle detection and tracking module (Chapter 5). As illustrated in Figure 6.1, the proposed risk assessment framework consists of three main stages: 1) Extended Kalman Filter (EKF) based trajectory prediction which estimates the near future trajectories for all the obstacles present in the traffic scene. 2) Collision prediction which evaluates collisions between the ego-vehicle and each obstacle based on their predicted trajectories; 3) Risk quantification which derives a risk indicator to describe the likelihood of the risk.

6.1.1 Trajectory Prediction

6.1.1.1 Kinematic Motion Model

A trajectory is a spatial-temporal representation of the displacement of the object. Based on the kinematic model, the evolution of obstacle's trajectory is controlled by the parameters of the movement such as obstacle's position and velocity. The same assumptions as in [202, 203], i.e. objects will maintain the same moving profile in the near future time period, are made in this thesis. That is, within a limited time range, it can be assumed that the velocity of object is constant.

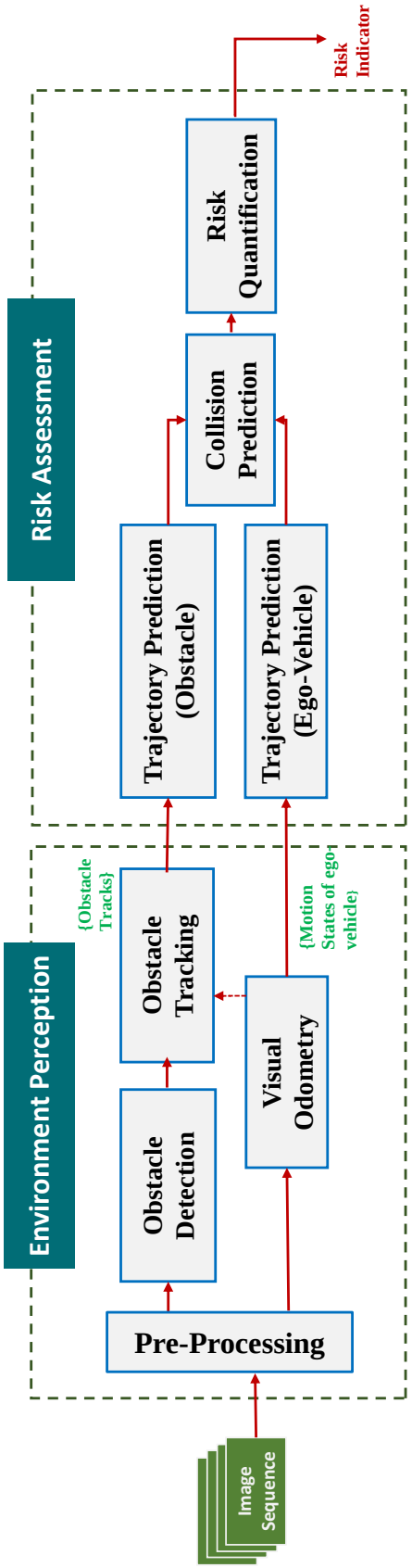


Fig. 6.1 Top-level block diagram of the proposed risk assessment algorithm: By integrating the motion state of ego vehicle obtained from visual odometry module (Chapter 4), and obstacle tracks obtained from obstacle detection and tracking module (Chapter 5), the proposed risk assessment framework composed of three main stages: 1) Extended Kalman Filter (EKF) based trajectory prediction: estimate the near future trajectories for all the obstacles present in the traffic scene. 2) Collision prediction: evaluate collisions between ego-vehicle and each obstacle based on their predicted trajectories; 3) Risk quantification: derive a risk indicator to describe the likelihood of the risk.

For collision avoidance on the road, only the earthbound movement is of concern. This means that the 3-dimentional *X-Y-Z* world movement can be reduced into 2-dimentional *X-Z* movement. In a world coordinate system where the origin is fixed and the *X-axis* and *Z-axis* point to the left and forward respectively, assume that an object is located in position $\mathbf{p}_i = (p_x^i, p_z^i)$ with speed $v_i = (v_x^i, v_z^i)$ at time t_i. The position, velocity of the object as function of time can then be formulated as shown in Eq. 6.1 – Eq. 6.2 respectively:

$$v_j = v_i \tag{6.1}$$

$$\mathbf{p}_j = \mathbf{p}_i + v_i * \Delta t \tag{6.2}$$

Where Δt represents the time elapsed from time t_i to t_j.

However, in reality, the scene is observed from the viewing angle of the camera installed on the ego-vehicle. This means that the origin of the coordinate is moving along with the ego-vehicle. All the above variables therefore need to be compensated by the ego-motion. Assume that the observed sub-chain of scene motion (the inverse of ego-motion) $\mathbf{A}_{i,j}$ from time t_i to time t_j is expressed as shown in Eq.6.3:

$$\mathbf{A}_{i,j} = \prod_{k=j}^{i+1} * \mathbf{T}_k = \mathbf{T}_j * \mathbf{T}_{j-1} * \therefore * \mathbf{T}_{i+1} = \begin{bmatrix} \mathbf{R}_{i,j} & \mathbf{t}_{i,j} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{6.3}$$

Where $\mathbf{T}_k$ is the scene motion from frame I_{k-1} to frame I_k as formulated in Eq. (4.6) in Chapter 4. Then the new position and velocity of the object as function of time are given as Eq. 6.4 – Eq. 6.5 respectively:

$$v_j = \mathbf{R}_{i,j} * v_i \tag{6.4}$$

$$\mathbf{p}_j = \mathbf{R}_{i,j} * (\mathbf{p}_i + v_i * \Delta t) + \mathbf{t}_{i,j} \tag{6.5}$$

[1] In actual matrix operation, since $\mathbf{R}_{i,j} \in \mathbb{R}^{3*3}$ and $\mathbf{t}_{i,j} \in \mathbb{R}^{3*1}$, $v_i = (v_x^i, v_z^i)$ needs to be expanded into a form of $v_i = (v_x^i, v_y, v_z^i)$, where v_y is set with a constant value.

[2] In actual matrix operation, due to the same reason, v_i needs to be expanded like above, and $\mathbf{p}_i = (p_x^i, p_z^i)$ also needs to be expanded into a form of $\mathbf{p}_i = (p_x^i, p_y, p_z^i)$, where p_y is set with a constant value.

6.1.1.2 Extended Kalman Filter based Motion Model

The problem with the above formulation is that it does not take into account noisy or erroneous measurement of object's motion parameters that are obtained from sensors or certain algorithms. Sometimes the measurement can even be lost. For example, this can occur during the stereo matching process, which is an important step used to reconstruct the 3D world. The fidelity of the disparity map generated by existing available stereo matching techniques is limited and hence, the estimated position of the obstacle will be easily contaminated with noise. In order to deal with such problem and increase the corresponding accuracy of objects' motion parameters in realistic environment, Extended Kalman Filter (EKF) is used for the stabilization of noisy measurements.

Kalman Filter is a data filtering algorithm that produces more accurate estimations of the internal state of a linear dynamic system from the noisy series of measurements observed over time. When the system variables to be estimated and (or) the measurement relationship to the system variables is non-linear, the Kalman Filter is extended and improved into a generalization form, which is referred to as Extended Kalman Filter [336].

The main principal of the Extended Kalman Filter will be explained in the following. Let $\mathbf{x}_k$ represent the system states at time t_k. Assume $\mathbf{x}_k$ is evolved from the state $\mathbf{x}_{k-1}$ at time t_{k-1} according to Eq. 6.6:

$$\mathbf{x}_k = \mathfrak{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k \tag{6.6}$$

Where $\mathfrak{f}$ is a differentiable function describing the relationship between $\mathbf{x}_k$ and $\mathbf{x}_{k-1}$. $\mathbf{u}_k$ is the control vector. $\mathbf{w}_k$ is the process noise which is assumed to be zero mean multivariate normal distribution with covariance $\mathbf{Q}_k$:

$$\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k) \tag{6.7}$$

In addition, an observation $\mathbf{z}_k$ at time t_k is made, which has the relationship with the system variable $\mathbf{x}_k$ according to Eq. 6.8:

$$\mathbf{z}_k = \mathfrak{h}(\mathbf{x}_k) + \mathbf{a}_k \tag{6.8}$$

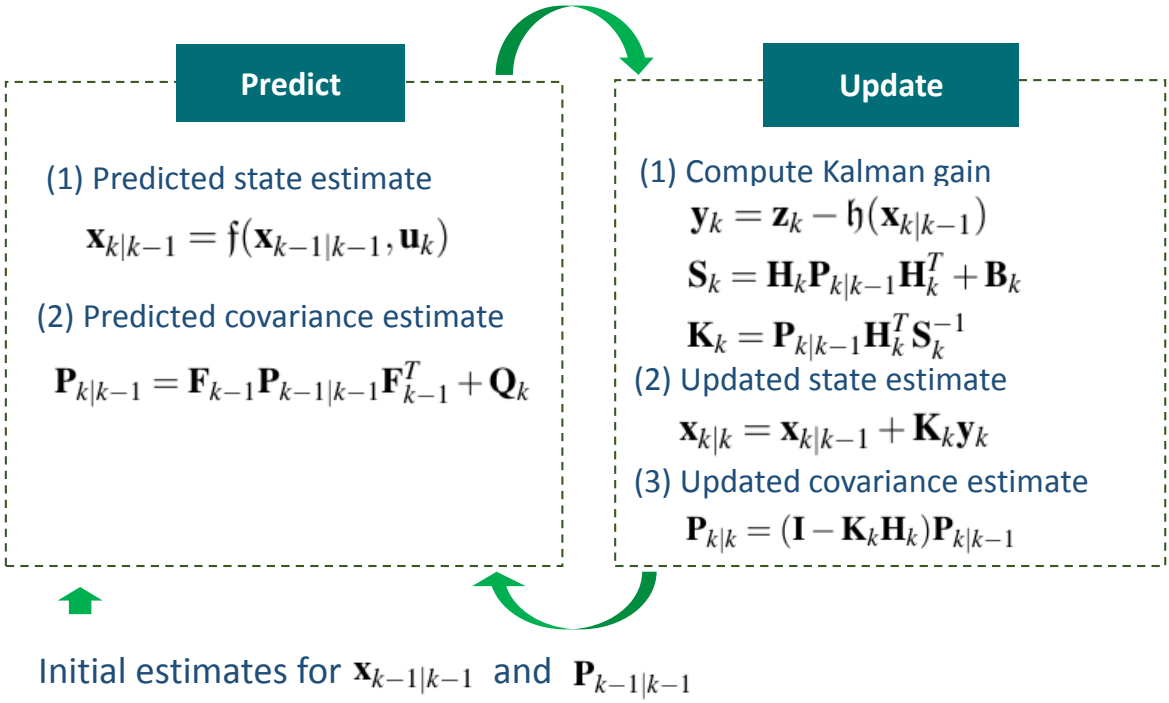


Fig. 6.2 Overview of Extended Kalman Filter: estimation of the system variables through the Extended Kalman Filter can be decomposed into two stages: "Predict" and "Update". The "Predict" step utilizes the state estimate at time t_{k-1} to produce *a priori* estimate of the state $\mathbf{x}_{k|k-1}$ at time t_k. In the "Update" step, by combing the measurement observed at time t_k, the *a priori* prediction is refined into the *a posteriori* state estimate $\mathbf{x}_{k|k}$. $\mathbf{F}_{k-1}$ is the state transition matrix and $\mathbf{H}_k$ is the observation matrix. Figure from [336].

Where $\mathfrak{h}$ is a differentiable function that describes the relationship between measurement and system variables. $\mathbf{a}_k$ is the observation noise which is assumed to be zero mean Gaussian white noise with covariance $\mathbf{B}_k$.

$$\mathbf{a}_k \sim \mathcal{N}(0, \mathbf{B}_k) \tag{6.9}$$

Based on the above formulation, estimation of the system variables through the Extended Kalman Filter can be decomposed into two stages: "Predict" and "Update". As illustrated in Figure 6.2, the "Predict" step utilizes the state estimate from the previous time step t_{k-1} to produce *a priori* estimate of the state $\mathbf{x}_{k|k-1}$ at the current time step t_k. In the "Update" step, by combing the measurement observed at current time step t_k, the *a priori* prediction is refined into the *a posteriori* state estimate $\mathbf{x}_{k|k}$ at t_k.

The Jacobian matrix $\mathbf{F}_{k-1}$ appearing in Figure 6.2 is called the state transition matrix as illustrated in Eq. 6.10:

$$\mathbf{F}_{k-1} = \frac{\partial \mathfrak{f}}{\partial \mathbf{x}}|_{\mathbf{x}_{k-1|k-1},\mathbf{u}_k} \tag{6.10}$$

The Jacobian matrix $\mathbf{H}_k$ appearing in Figure 6.2 is called the observation matrix as illustrated in Eq. 6.11:

$$\mathbf{H}_k = \frac{\partial \mathfrak{h}}{\partial \mathbf{x}}|_{\mathbf{x}_{k|k-1}} \tag{6.11}$$

By integrating the Extended Kalman Filter into the kinetic motion model, the proposed Extended Kalman Filter based motion model will be presented in the following.

Combining position (p_x, p_z) and velocity (v_x, v_z) in the state vector, the system variables at time t_k are denoted as:

$$\mathbf{x}_k = (p_x^k, p_z^k, v_x^k, v_z^k)^T \tag{6.12}$$

The process noise $\mathbf{w}_k$ is assumed to be zero mean multivariate normal distribution with covariance $\mathbf{Q}_k$.

The state transition matrix $\mathbf{F}_k$ is then formulated as in Eq. 6.13:

$$\mathbf{F}_k = \begin{bmatrix} \mathbf{R}_{k-1,k} & \mathbf{R}_{k-1,k} * \Delta t \\ \mathbf{0} & \mathbf{R}_{k-1,k} \end{bmatrix} \tag{6.13}$$

And the control vector $\mathbf{u}_k$ is shown in Eq. 6.14:

$$\mathbf{u}_k = \begin{bmatrix} \mathbf{t}_{k-1,k} \\ \mathbf{0} \end{bmatrix} \tag{6.14}$$

where $\mathbf{R}_{k-1,k}$ in Eq. 6.13 and $\mathbf{t}_{k-1,k}$ in Eq. 6.14 are the corresponding scene rotation matrix and translation vector from time t_{k-1} to time t_k as discussed in Eq. 6.3.

In addition, at time t_k, a measurement of the image coordinates u, v, and the corresponding disparity value d is observed. Following the camera model introduced in Section 3.1 and

Section 5.1, the relationship between the state variables p_x^k and p_z^k and the measurements u_k and d_k are formulated as shown in Eq. 6.15:

$$\begin{bmatrix} u_k \\ d_k \end{bmatrix} = \begin{bmatrix} \frac{focal * p_x^k}{p_z^k} + u_0 \\ \frac{baseline * focal}{p_z^k} \end{bmatrix} \tag{6.15}$$

Where $baseline$, $focal$ and (u_0, v_0) are the baseline, focus length and principal point for the camera system respectively.

The observation model $\mathbf{H}_k$ is therefore formulated as shown in Eq. 6.16:

$$\mathbf{H}_k = \begin{bmatrix} \frac{focal}{p_z^k} & -\frac{k * p_x^k}{(p_z^k)^2} & 0 & 0 \\ 0 & -\frac{focal * baseline}{(p_z^k)^2} & 0 & 0 \end{bmatrix} \tag{6.16}$$

The noise term $\mathbf{a}_k$ is assumed to be Gaussian white noise with covariance matrix $\mathbf{B}_k$.

Each time a new obstacle is detected, a corresponding Extended Kalman Filter is created for the obstacle. The system variables $\mathbf{x}$ is initialized with $\mathbf{x}_0 = (p_x^0, p_z^0, 0, 0)^T$, where (p_x^0, p_z^0) corresponds to the position of the detected obstacle. When the obstacle is tracked in the new frame, its state is refined using the Extended Kalman Filter.

The benefit of state estimation using the Extended Kalman Filter is clearly illustrated in Figure 6.3. It can be observed that with Extended Kalman Filter, the estimated velocity in Figure 6.3(a) and (b) for the pedestrian highlighted in Figure 6.3(c) correctly converges to the value of v_x = 1.75 meter/frame and v_z = 0 meter /frame. The pseudo code for the proposed Extended Kalman Filter based trajectory estimation method is presnted in Listing 6.1.

6.1.2 Collision Prediction

A collision between two entities means that the two corresponding trajectories evolving from their current places will arrive at the same position at the same time instant in the future. Such phenomenon is called route contention [269].

Figure 6.4 depicts a general trajectory intersection scenario. Two objects at (x_1, z_1) and (x_2, z_2) are moving at speeds v_1 and v_2 respectively. Then the expected path intersection

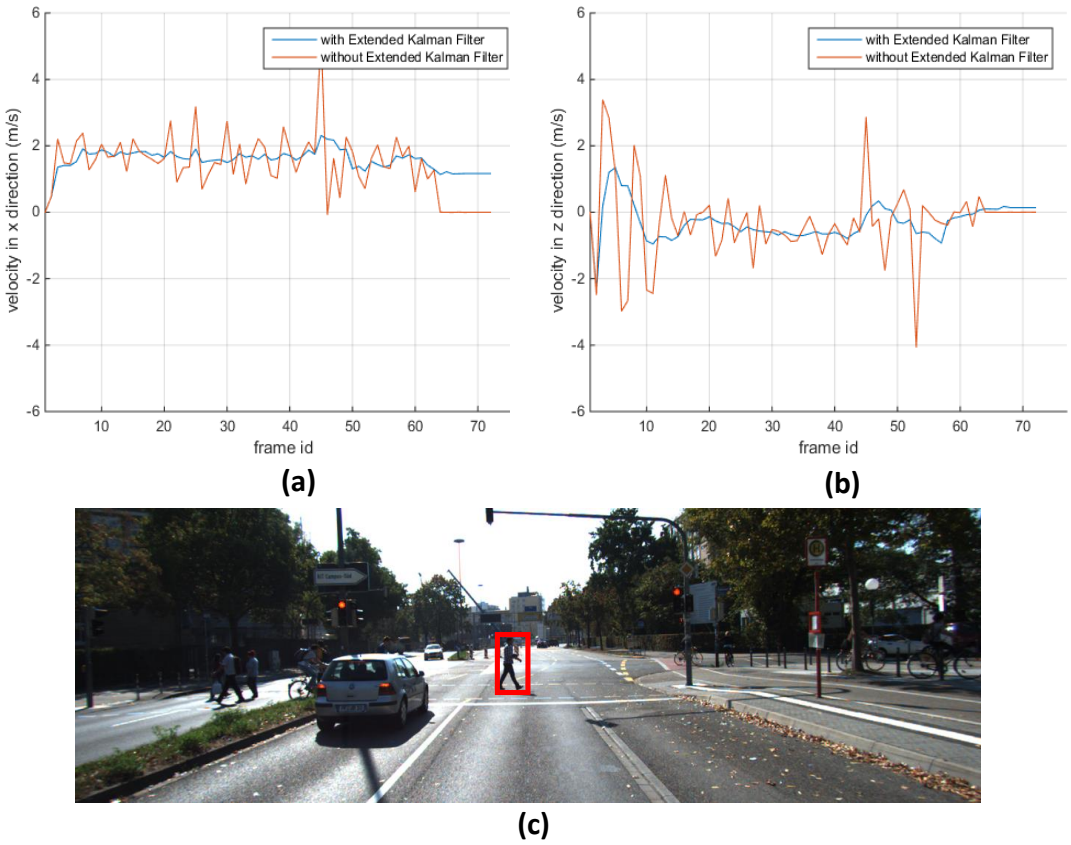


Fig. 6.3 Velocity estimated with Extended Kalman Filter (blue plot) and without Extended Kalman Filter (orange plot). With Extended Kalman Filter, the estimated velocity in (a) and (b) for the pedestrian highlighted in (c) correctly converges to the value of v_x = 1.75 meter/frame and v_z = 0 meter /frame.

Listing 6.1 Extended Kalman Filter (EKF) based Trajectory Estimation

Input: A set of tracks $\mathscr{T} = \{tr_i\}$ identified in current frame;
Output: A updated set of tracks $\mathscr{T} = \{tr_i\}$ with updated motion model.

1: **for** each track $tr_i \in \mathscr{T}$ **do**
2: **if** $tr_i.state = new$ **then**
3: $tr_i.\mathbf{x} = EKF\_Initialization(p_x^0, p_z^0, 0, 0)^T$;
4: **else**
5: $EKF\_Prediction(tr_i.\mathbf{x})$;
6: $EKF\_Update(t_i.\mathbf{x}, t_i.\mathbf{z})$;
7: **end if**
8: **end for**

(x_+, z_+) of the two objects trajectories are computed as shown in Eq. 6.17 and Eq. 6.18 respectively.

$$x_+ = \frac{(z_2 - z_1) - (x_2 \tan\varphi_2 - x_1 \tan\varphi_1)}{\tan\varphi_1 - \tan\varphi_2} \tag{6.17}$$

$$z_+ = \frac{(x_2 - x_1) - (z_2 \cot\varphi_2 - z_1 \cot\varphi_1)}{\cot\varphi_1 - \cot\varphi_2} \tag{6.18}$$

Based on the motion model, the expected time-to-intersection (TTX) for each object is computed as illustrated in Eq. 6.19 and Eq. 6.20:

$$TTX_1 = \frac{|\vec{r}_+ - \vec{r}_1|}{|\vec{v}_1|} sign((\vec{r}_+ - \vec{r}_1) \cdot \vec{v}_1) \tag{6.19}$$

$$TTX_2 = \frac{|\vec{r}_+ - \vec{r}_2|}{|\vec{v}_2|} sign((\vec{r}_+ - \vec{r}_2) \cdot \vec{v}_2) \tag{6.20}$$

Where $\vec{v}_1$ and $\vec{v}_2$ are the velocities of the two objects respectively, $\vec{r}_n$ is the vector representation of coordinate (x_n, z_n), and $sign()$ is a sign function.

When $TTX_1 = TTX_2$, a route contention is identified. The time-to-collision (TTC) is then determined as shown in Eq. 6.21:

$$TTC = \begin{cases} TTX_i, & \text{if there is a route contention} \\ undefined, & \text{otherwise} \end{cases} \tag{6.21}$$

The above route contention model from [269] is based on the assumption that the vehicles or objects are abstract points. This is however not true in reality. Instead, objects including both of the ego-vehicle and obstacles are of different sizes. In addition, the localization uncertainty increase due to the sensor sampling and objects' unexpected behaviour. To overcome these problems, both the ego vehicle and the the detected obstacles are represented using circles as illustrated in Figure 6.5. In addition, the radius of the circle is increased linearly as a function of the estimated travelled distance by the ego-vehicle or obstacle. At this time, as shown in Figure 6.6, a potential collision is detected when the circle corresponding to the ego-vehicle intersects at least one circle corresponding to the dynamic objects at the same

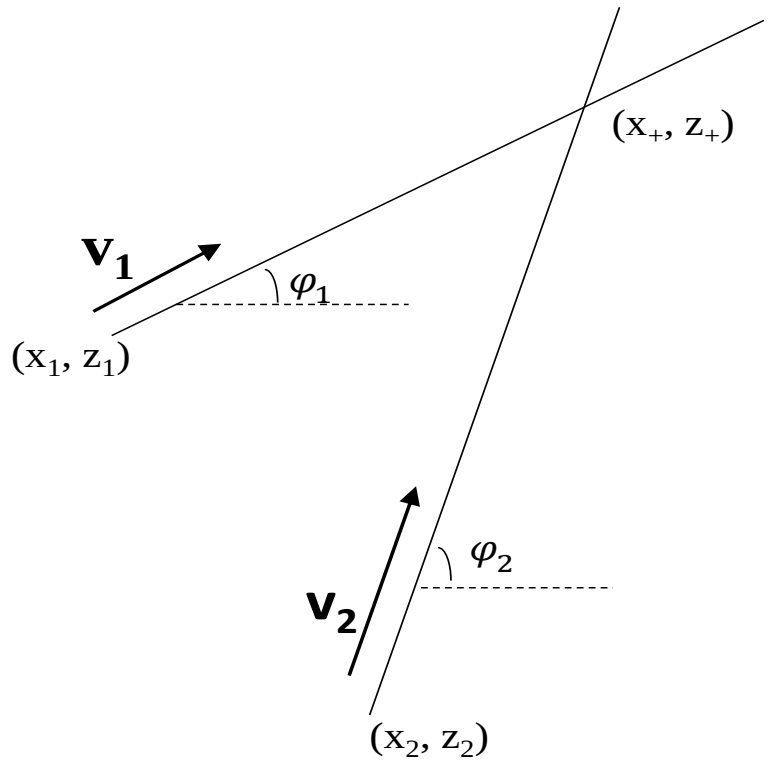


Fig. 6.4 Illustration of two-objects route contention. Figure from [269].

time. An assumption is made here wherein objects maintain constant speed until the moment of the collision.

Assume that the position and velocity of the obstacle at time t_i are $\mathbf{p}_i = (p_x^i, p_z^i)$ and $v_i = (v_x^i, v_z^i)$ respectively. And the position and velocity of the ego-vehicle at time t_i are $\mathbf{q}_i = (q_x^i, q_z^i)$ and $\mu_i = (\mu_x^i, \mu_z^i)$ respectively. In addition, the radius for the circles corresponding to the ego-vehicle and obstacle at time t_i are RA_i and ra_i. After some time period Δt has elapsed, the new position $\mathbf{p}_j = (p_x^j, p_z^j)$ and radius RA_j for the circle corresponding to the obstacle and the position $\mathbf{q}_j = (q_x^j, q_z^j)$ and radius ra_j for the circle corresponding to the ego-vehicle are as shown in Eq. 6.22 – Eq. 6.27 respectively:

$$p_x^j = p_x^i + v_x^i * \Delta t \tag{6.22}$$

$$p_z^j = p_z^i + v_z^i * \Delta t \tag{6.23}$$

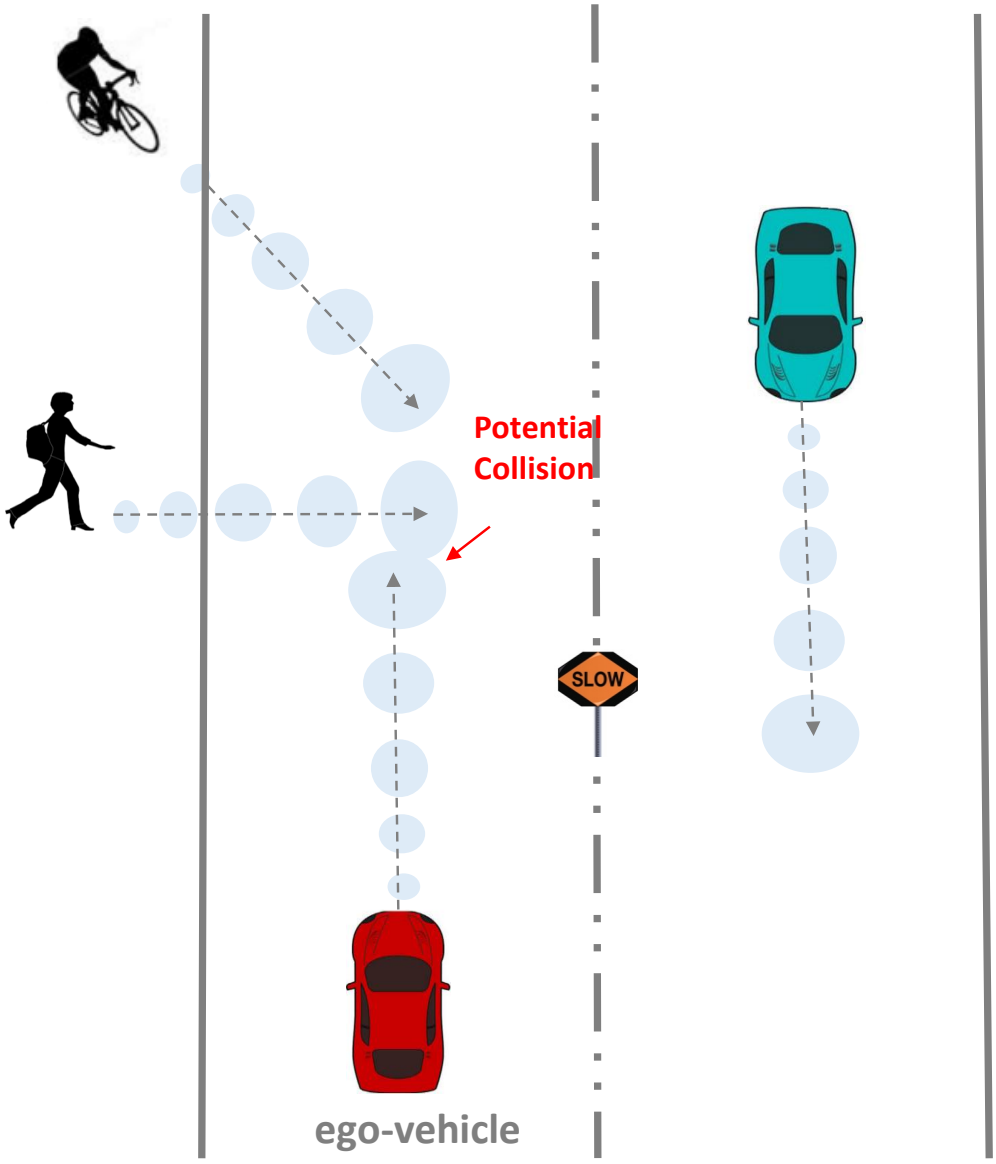


Fig. 6.5 Example of a potential collision. Both the ego vehicle and the detected obstacles are represented using circles. In addition, the radius of the circle is increased linearly as a function of the estimated travelled distance by the ego-vehicle or the obstacle.

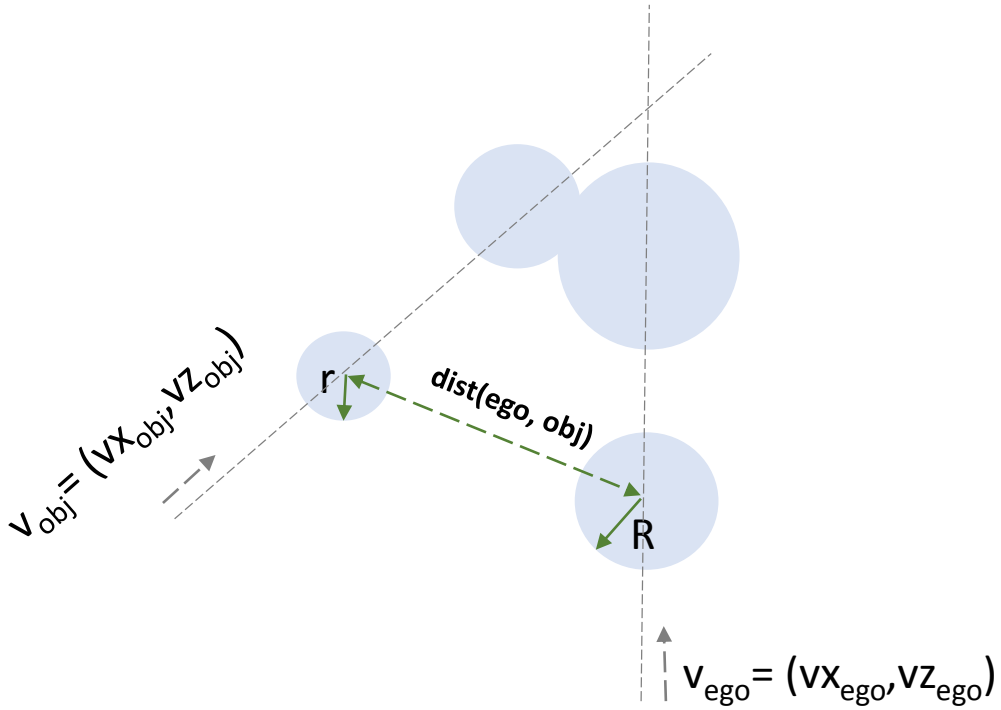


Fig. 6.6 Illustration of collision prediction.

$$q_x^j = q_x^i + \mu_x^i * \Delta t \tag{6.24}$$

$$q_z^j = q_z^i + \mu_z^i * \Delta t \tag{6.25}$$

$$RA_j = RA_i + ratio * |v_i| * \Delta t \tag{6.26}$$

$$ra_j = ra_i + ratio * |\mu_i| * \Delta t \tag{6.27}$$

where $ratio$ is a parameter controlling the increase of the circle's size. $|v_i|$ and $|\mu_i|$ are specified as in Eq. 6.28 and Eq. 6.29:

$$|v_i| = \sqrt{(v_x^i)^2 + (v_z^i)^2} \tag{6.28}$$

$$|\mu_i| = \sqrt{(\mu_x^i)^2 + (\mu_z^i)^2} \tag{6.29}$$

The distance between the two circle centres are then formulated as

$$distance(ego, obj) = \sqrt{(p_x^j - q_x^j)^2 + (p_z^j - q_z^j)^2} = \sqrt{a * \Delta t^2 + b * \Delta t + c} \tag{6.30}$$

The two circles will intersect and a collision is detected when the following Eq. 6.31 holds:

$$distance(ego, obj) < RA_j + ra_j \tag{6.31}$$

Combing Eq. 6.22 - Eq. 6.30, Eq. 6.31 can be transformed into a form like Eq. 6.32:

$$\alpha * \Delta t^2 + \beta * \Delta t + \gamma < 0 \tag{6.32}$$

$$\alpha = (v_x^0 - \mu_x^0)^2 + (v_z^0 - \mu_z^0)^2 - ratio^2 * (|\mu_0| + |v_0|)^2 \tag{6.33}$$

$$\beta = 2(v_x^0 - \mu_x^0)(p_x^0 - q_x^0) + 2(v_z^0 - \mu_z^0)(p_z^0 - q_z^0) - 2 * ratio * (|\mu_0| + |v_0|)(RA_0 + ra_0) \tag{6.34}$$

$$\gamma = (p_x^0 - q_x^0)^2 + (p_z^0 - q_z^0)^2 - (RA_0 + ra_0)^2 \tag{6.35}$$

Depending on the value of α, Eq. 6.32 can be a linear or quadratic inequality problem which takes Δt as the variable.

Taking into account that Δt must be non-negative, if the solution for Eq. 6.32 exists, the solution can be expressed as a for shown in Eq. 6.36:

$$\Delta t \in (t_1, t_2), \text{where } t_1 \geq 0 \tag{6.36}$$

Once the solution for Eq. 6.32 exists, a collision is detected between the ego-vehicle and the corresponding obstacle. At this time, the value of TTC is determined as the time that remains before the first impact occurs, that is, $TTC = t_1$. Otherwise, if there is no solution for Eq. 6.32, TTC is set to an infinitely large value, which means that it is impossible for the ego-vehicle to collide with the obstacle.

$$TTC = \begin{cases} t_1, & \text{solution for Eq.6.32 exists;} \\ \infty, & \text{otherwise.} \end{cases} \tag{6.37}$$

6.1.3 Risk Quantification

TTC is a good metric describing the severity degree of the detected collision [265, 269, 272, 291, 296, 297]. Small TTC means that there is a risk of collision in near future and the ego-vehicle is in a safety-critical situation. Large TTC means that although a collision is predicted, the ego-vehicle is still safe. An indefinitely large TTC means no collision is detected between the ego-vehicle and the corresponding obstacle.

The value of TTC can therefore be used to derive a collision risk indicator. In the context of collision avoidance, when a vehicle comes to a full stop, the driver needs at least 1 second to react and then the vehicle also needs another 1 second to respond [298, 337]. Therefore, the same risk quantification method as in [298] is adopted. As shown in Figure 6.7, when TTC is smaller than 2 seconds, the risk is highest and equals to one. When TTC falls between 2 to 5 seconds, the risk decreased linearly. When TTC is larger than 5, the situation is safe and the collision risk equals 0.

Based on the derived TTC and the quantification of the collision risk, suitable intervention strategy like warning or braking can then be deployed accordingly. The pseudo code for the proposed collision prediction and risk quantification method is presented in Listing 6.1.

6.2 Experimental Evaluation

Based on the motion state of the ego-vehicle obtained from visual odometry module (Chapter 4), and the perceived obstacle tracks obtained from object detection and tracking module (Chapter 5), the proposed framework in this chapter aims at providing an assessment of the potential collision risk between the ego-vehicle and obstacles that are present in the

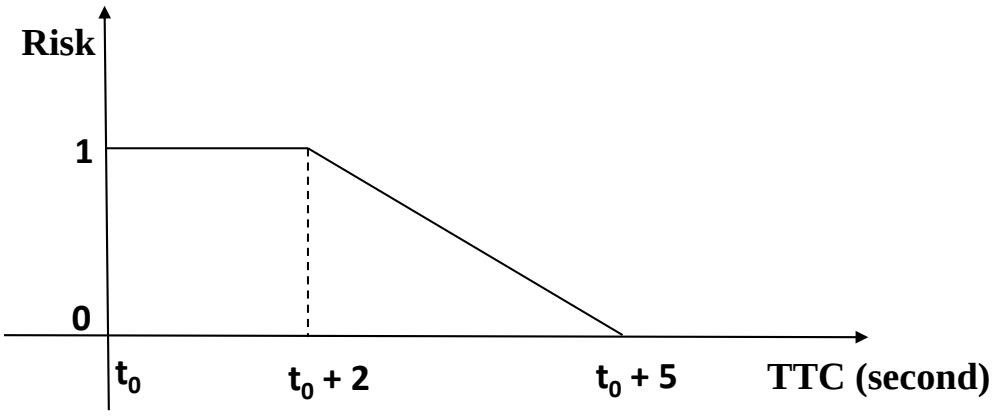


Fig. 6.7 Relationship between TTC and risk indicator. When TTC is smaller than 2 seconds, the risk is highest and equals to one. When TTC falls between 2 to 5 seconds, the risk decreased linearly. When TTC is larger than 5, the situation is safe and the collision risk equals 0. Figure from [298].

Listing 6.2 Collision Prediction and Risk Quantification

Input: A set of tracks $\mathscr{T} = \{tr_i\}$;
Output: Times to Collision $\{TTC_i\}$ and Risk indicators $\{risk\_indicator_i\}$.

for each track $tr_i \in \mathscr{T}$ **do**
  Solve Eq. 6.32;
  if the solution $(t1, t2)$ of Eq. 6.32 exists **then**
    $TTC_i = t_1$;
  else
    $TTC_i = \infty$;
  end if
  if $TTC_i \leq 2$ **then**
    $risk\_indicator_i = 1$;
  else
    if $TTC_i \leq 5$ **then**
      $risk\_indicator_i = \dfrac{(55 - 11 * TTC_i)}{3}$;
    else
      $risk\_indicator_i = 0$;
    end if
  end if
end for

surrounding environment. In this section, a comprehensive evaluation of the proposed risk assessment framework will be provided.

6.2.1 Benchmarks

A comprehensive evaluation of the proposed risk assessment method is conducted on the well-known KITTI tracking dataset [26], which has been introduced in Chapter 5. As mentioned in [202], the most probable practical traffic scenarios are: stationary objects, vehicles driving in the same direction as the ego-vehicle with small relative speed, oncoming traffic, traffic from left, and traffic from right. The ego-vehicle itself can be stationary or moving. hence, in the following, the evaluation results for these representative scenarios described above will be presented. Note the frame rate for the KITTI tracking benchmark is about 10 frames/second.

6.2.2 Accuracy Evaluation

In the figures that follow, different symbols are relied on to express different meanings. In the upper part of each figure, each detected obstacle is denoted using white bounding box. The red number appearing in the middle of the white bounding box represents the corresponding tracked id. Obstacles that are associated with the same tracked id across frames means that they correspond to the same object. Green line with arrow indicates the predicted position for the corresponding obstacle in 0.5 second. Once collision risk is predicted for certain obstacle, it is highlighted using red bounding box. For the bottom plot, the trajectories estimated for the ego-vehicle and the obstacles across frames are denoted using red star and blue dot respectively.

A. Scenario I

Figure 6.8 depicts a scenario where the ego-vehicle stops at the intersection and is waiting for the traffic light to turn green. The bottom part of Figure 6.8 shows a projection of the 3D scene, which evolves from frame 25 to frame 50 and lasts for 2.5 seconds, into the 2D X-Z plane. The red stars represent the recorded trajectory for the ego-vehicle. As can be seen, the red stars do not scatter across time. This means that the ego-vehicle is stationary and its velocity is zero. At the same time, the three obstacles (traffic light) have been detected at the left side and front of the ego-vehicle with distances of 6.87 meter, 8.91 meter and 31.98 meter respectively. They are tracked across frames. Their behaviors are analyzed using the method proposed in the current chapter and it is found that their velocities are zero. The green line with arrow is used to indicate the predicted position of the corresponding obstacle in 0.5

second. For the three detected obstacles (traffic light), the starting point and ending point of the green line overlaps. This means that the obstacles are stationary and will stay at the same position for 0.5 seconds. Based on the motion states of both ego-vehicle and obstacles, the collision prediction module proposed in this chapter determines that no collision will take place and therefore the collision risk is 0.

At frame 39, a vehicle appears in the scene. At soon as the vehicles appear in the frame, it is detected and tracked. Its behavior is analyzed and it is found that its velocity in x direction is almost zero and the velocity in z direction is about 7.65 meter/second. This means that the vehicle is just overtaking the ego-vehicle and moving forward. The predicted position shows that its distance from the ego-vehicle will progressively increase. Therefore, the collision risk in the environment is 0.

B. Scenario II

Figure 6.9 depicts a scenario where the ego-vehicle is moving forward on the road. From the proposed visual odometry method presented in Chapter 4, it is found that the ego-vehicle is moving at a constant speed of 14.19 m/s. An obstacle (vehicle) is detected at a distance of 20.59 meters in the front of the ego-vehicle and it is tracked across frames. The trajectory prediction module presented in this chapter determines that the velocity of the detected vehicle is almost equal to the ego-vehicle and it is on the same course as ego-vehicle. The risk assessment module therefore determines that the risk between ego-vehicle and the vehicle ahead is zero.

At frame 81, two vehicles are detected at the left side of the ego-vehicle. They are tracked and it is determined that they move in the opposite direction and the courses are in parallel to the ego-vehicle. Therefore, they are safe. At the same time, the static obstacles such as traffic sign and trees are also correctly perceived and analyzed. No collision risk exist between the ego-vehicle and these static obstacles.

C. Scenario III

Figure 6.10 depicts a scenario where the ego-vehicle is moving forward and a cyclist ahead of the ego-vehicle tries to move across the road. It can be seen from Figure 6.10 that the ego-vehicle moves forward at a speed of 7.13 meter/second. The cyclist is detected and tracked across frames and his trajectory is analyzed and predicted. The green arrow indicates that the cyclist intends to move across the road. However, since the distance between the cyclist and the ego-vehicle is still larger than 30 meters, the risk assessment module proposed

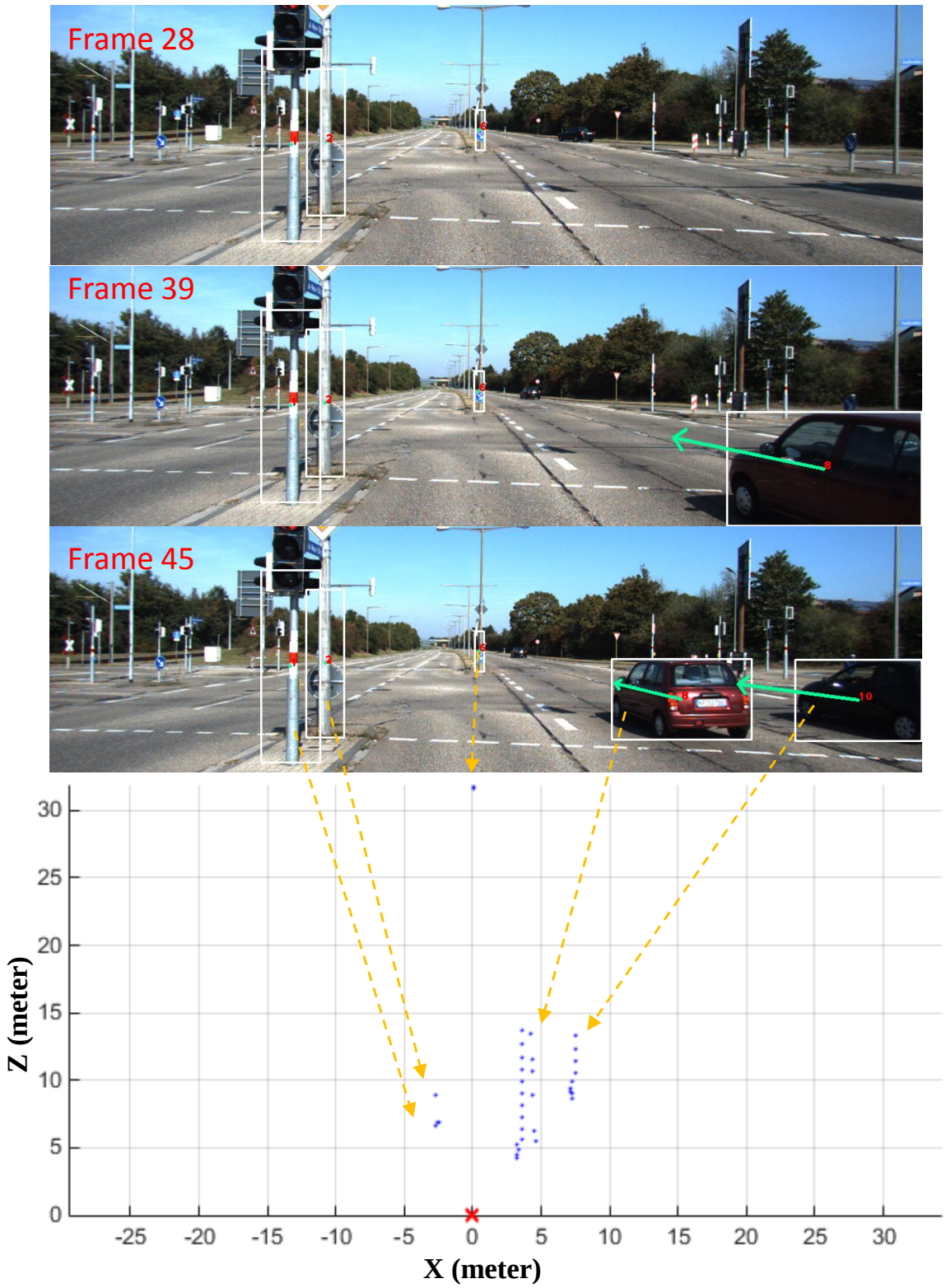


Fig. 6.8 Scenario I: the ego-vehicle is stopping at the intersection waiting for the traffic light. The scene understanding results of the proposed risk assessment algorithm for this scenario is discussed in Section 6.2.2-A.

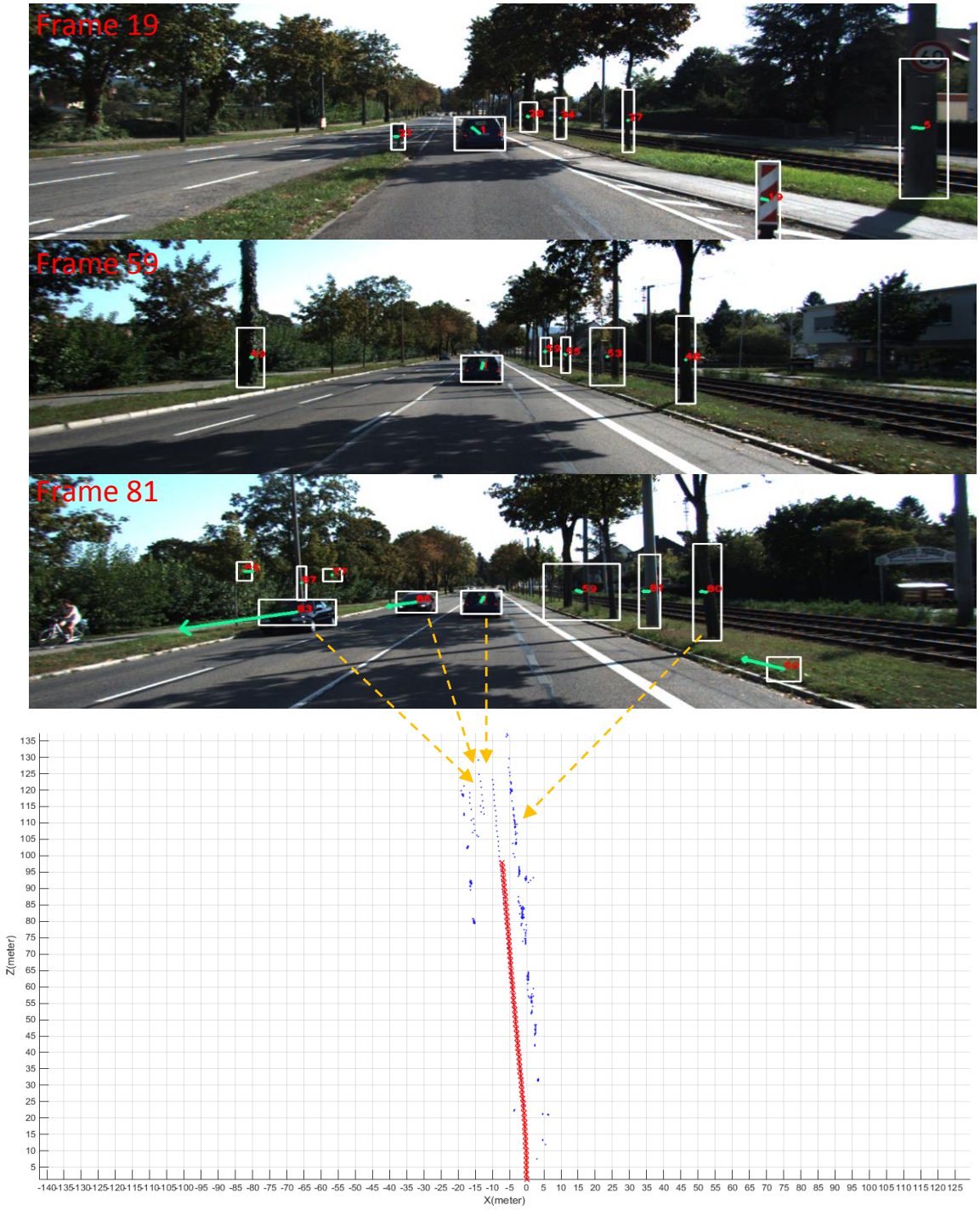


Fig. 6.9 Scenario II: the ego-vehicle is moving forward on the road. The scene understanding results of the proposed risk assessment algorithm is discussed in Section 6.2.2-B.

in this chapter determines that by the time the bicyclist arrives at the center of the road, he will be within a safe distance from the ego-vehicle. The ego-vehicle can therefore maintain the same speed and direction of motion.

D. Scenario IV

Figure 6.11 depicts a scenario at an intersection where the ego-vehicle is moving towards the intersection. One obstacle (vehicle) is identified on the left-front side and two obstacles (traffic light) are identified on the right-front side. Their speed is determined as zero. At the same time, an obstacle (pedestrian) is detected and is determined to be moving left-wards with a speed of about 1.76 meter /second in the X direction and 0 meter /second in the Z direction. Since the obstacle (pedestrian) is far, the risk assessment module proposed in this chapter determines that no collision is going to happen between ego-vehicle and the pedestrians. At frame 72, a new obstacle (pedestrian) is detected and he is also found to be moving left at a speed of about 1.81 meter /second in the X direction only. The current speed for ego-vehicle estimated from the visual odometry module indicates that the ego-vehicle only moves in the Z direction at a speed of 2.85 meter /second. Assuming that both the ego-vehicle and the pedestrian keep their motion state in the following time period, the risk assessment module proposed in this chapter determines that they are going to collide in 2.3 seconds. This pedestrian is therefore highlighted in red bounding box indicating a collision risk is predicted for it. At the same time, another obstacle (cyclist) is also entering into the zone with a larger velocity than the pedestrian. Based on the extracted motion state, a collision risk between the cyclist and ego-vehicle is zero. The ego-vehicle continues to decrease its speed and finally stops at some position. At this time, the collision risk between the ego-vehicle and other obstacles are found to be zero.

E.Some Failure Cases

The accurate execution of the risk assessment is easily affected by the results of the obstacle detection and tracking. If errors happen in either stage of the obstacle detection or tracking, the risk assessment will become inaccurate. Figure 6.12 and Figure 6.13 shows some typical failure cases under these scenarios. In Figure 6.12, two black vehicles which are close to each other are located on the right-hand side of the road. Since their appearance are highly similar, the obstacle tracking module wrongly associate them as a single object (i.e., obstacle tracklet with id 747). At this time, the position for the tracklet with id 747 changes. Hence, it is detected to be in motion and the green arrow shows its predicted position in 0.5 second. However, tracklet with id 747 actually corresponds to two stationary vehicles in the continuous frames.

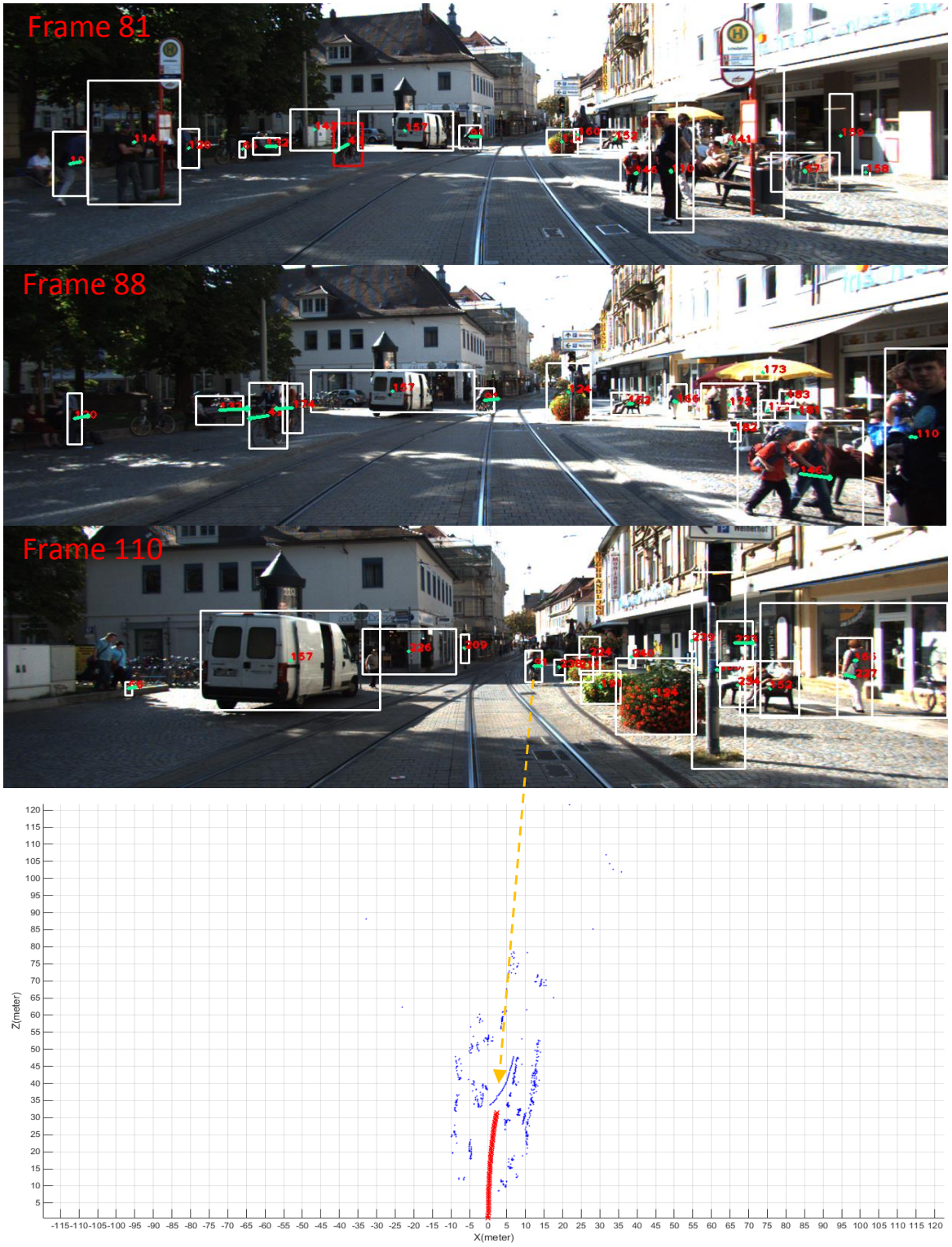


Fig. 6.10 Scenario III: ego-vehicle is moving forward and a cyclist ahead of the ego-vehicle tries to move across the road. The scene understanding results of the proposed risk assessment algorithm is discussed in Section 6.2.2-C.

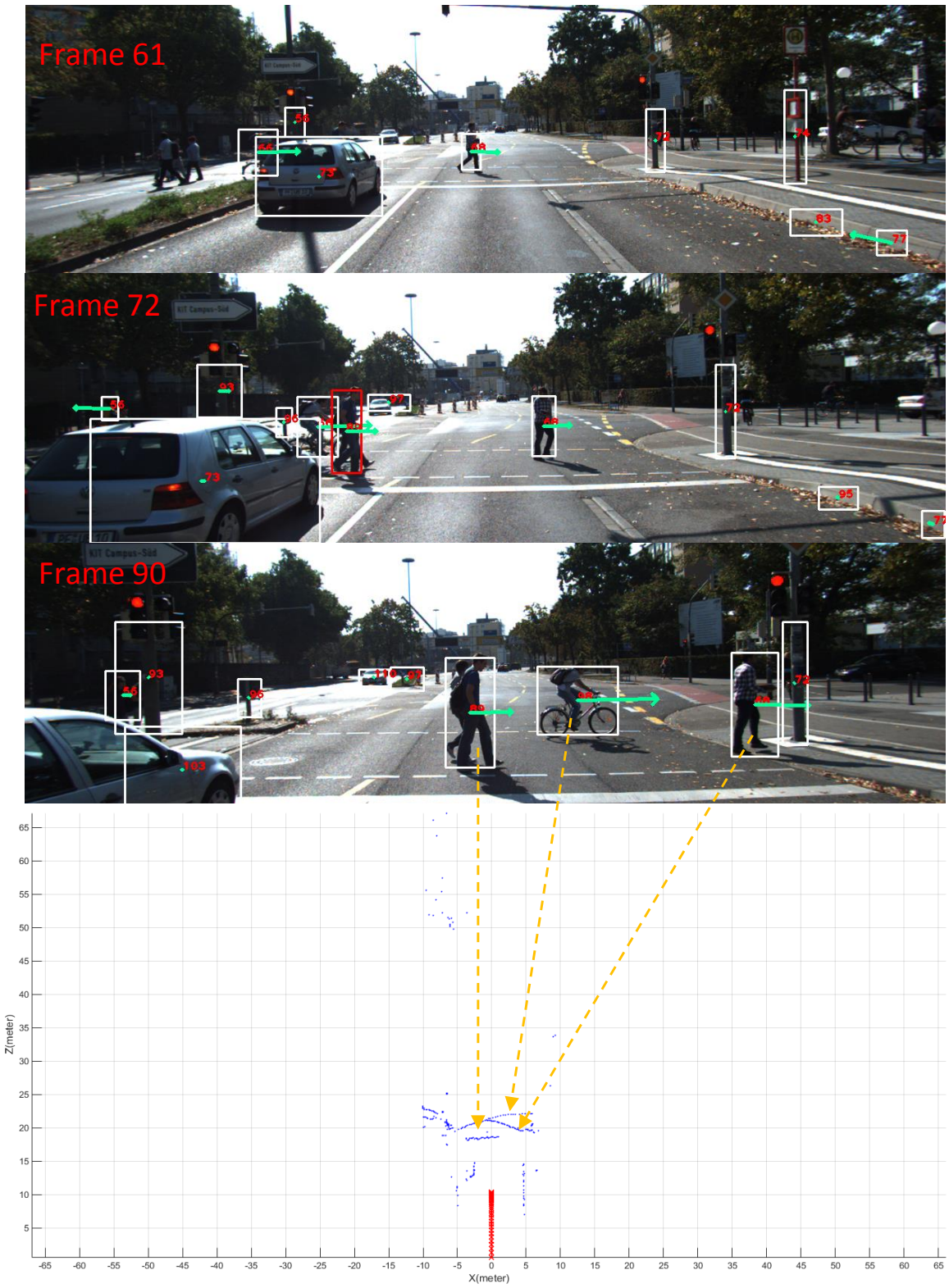


Fig. 6.11 Scenario IV: the ego-vehicle is moving closer to the intersection. The scene understanding results of the proposed risk assessment algorithm is discussed in Section 6.2.2-D.

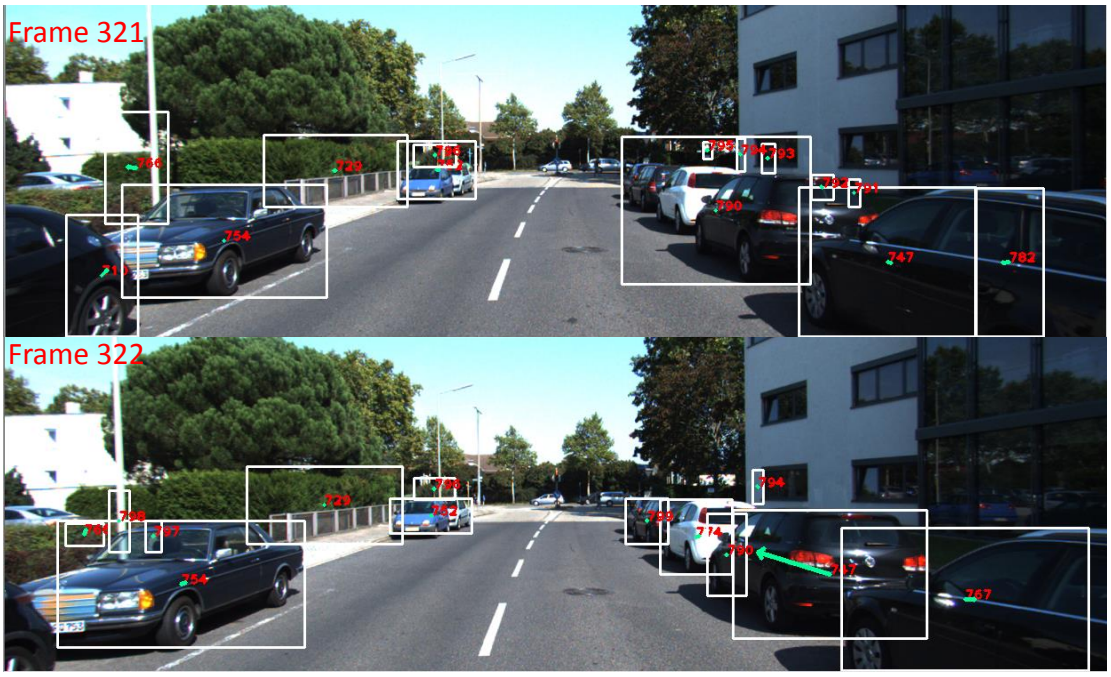


Fig. 6.12 Typical failure case I: two black vehicles locates on the right-hand side of the road and are close to each other. Since their appearance are highly similar, obstacle tracking module wrongly associate them as one object (i.e., obstacle tracklet with id 747). At this time, the position for the tracklet with id 747 changes. Hence, motion is detected for it and the green arrow shows its predicted position in 0.5 second. However, tracklet with id 747 actually correspond to two stationary vehicles at different time instant.

In Figure 6.13, some bushes appear on the right side of the road and are detected with tracklet id 23 and 34. Since the corresponding disparity map for this part is not accurate, the detected range for them are not accurate. Hence, they are determined to be in motion although they are stationary in reality.

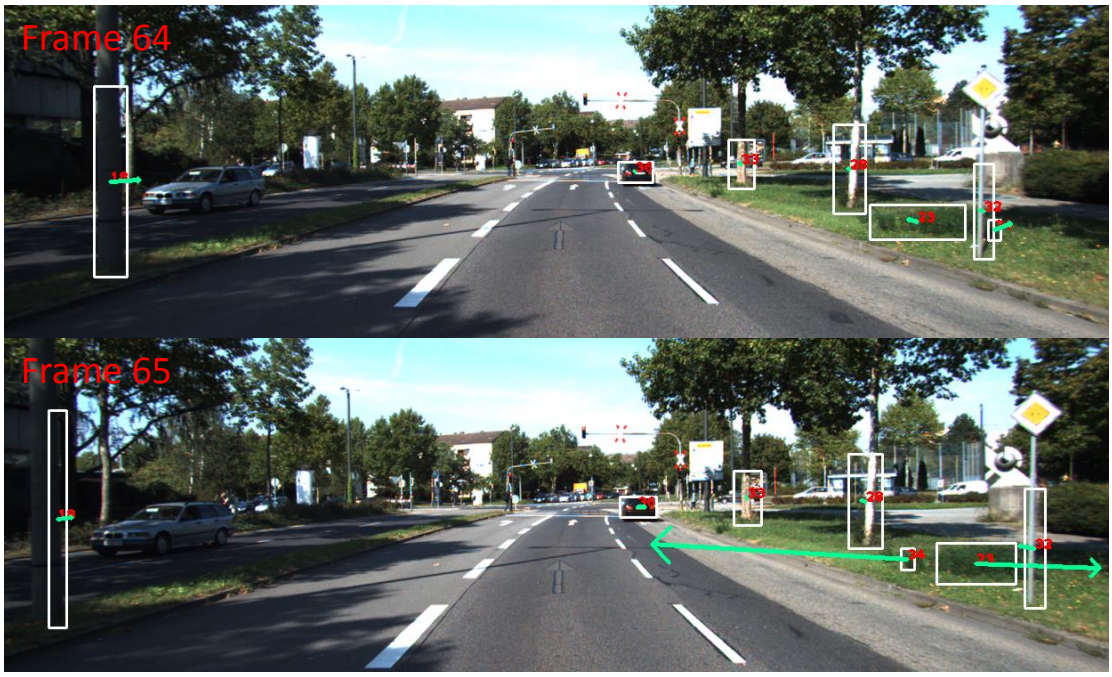


Fig. 6.13 Typical failure case II: some bushes appear on the right side of the road and are detected with tracklet id 23 and 34. Since the corresponding disparity map for this part is not accurate, the detected range for them are not accurate. Hence, motion is detected for them, even though they are in fact stationary.

In this thesis, only straight motion model is considered. This will lead to some inaccurate assessment of the environment in some scenarios. As shown in Figure 6.14, by making the assumption that the ego-vehicle will keep its motion state including direction unchanged in near future, collision risk for obstacles with id 49 and 50 is predicted at frame 232 since they are located in the predicted course of the ego-vehicle. However, the ego-vehicle is in fact making a left-turn and obstacles with id 49 and 50 should not pose any risk of collisions.

6.2.3 Runtime Performance Evaluation

In this section, the evaluation of the proposed algorithm's runtime performance is presented. Currently, the risk assessment module and other functional blocks (i.e., road surface detection, obstacle detection and tracking, visual odometry) has been integrated into a complete vision

Fig. 6.14 Typical failure case III: Taking the assumption that ego-vehicle will keep its motion state unchanged in near future, collision risk for obstacles with id 49 and 50 is predicted at frame 232 since they are located in the predicted course of the ego-vehicle at that moment. However, the ego-vehicle is in fact making a left-turn and obstacles with id 49 and 50 should not pose any collision risks.

Table 6.1 Runtime performance evaluation for the proposed risk assessment algorithm on a platform with 3.5GHz CPU. In addition, the computational time for other stages of the overall scene understanding system has also been listed. On average, the computational time of the proposed scene understanding system is about 0.77 second per frame.

Stage	**Second/Frame**
Loading Images	0.06725
Stereo Matching (OpenCV Implementation)	0.65023
Road Surface Detection (Chapter 3)	0.00692
Visual Odometry (Chapter 4)	0.02755
Obstacle Detection (Chapter 5)	0.01156
Obstacle Tracking (Chapter 5)	0.00304
Risk Assessment (Chapter 6)	0.00005
Total	0.76660

based scene understanding system for collision avoidance on roadway. The whole system is implemented on a PC platform Hp Z420 Workstation, where the processor is Intel(R) Xeon(R) CPU E5-1650 v2 3.50 GHz with 16GB memory. All the codes are developed in C++ in the Visual Studio 2012 running in Windows 7. As illustrated in Table 6.1, only 0.05 millisecond is required for the risk assessment part. In addition, the computational time for other stages of the whole collision avoidance system is also shown. It can be observed that the whole system runs at 0.77 second per frame on average. The biggest bottleneck lies in stereo matching, which we currently utilize the default OpenCV implementation of Semi-Global Matching (SGM) algorithm. Excluding the time for loading images and stereo matching, the other modules only needs about 0.049 second/frame. One of our future work is therefore to propose a robust stereo matching algorithm with low computational complexity. It is worth pointing out that the current implementation doesn't employ any code optimization techniques e.g. instruction- or thread-level parallelism or hardware accelerators on FPGA platform. By employing these optimization techniques, we expect that the run-time of the proposed algorithms can be further reduced.

6.3 Summary

In this chapter, all the computational blocks, namely, road surface detection (Chapter 3), visual odometry (Chapter 4), obstacle detection (Chapter 5), and obstacle tracking (Chapter 5), are integrated into a framework for risk assessment. The proposed risk assessment method is composed of three stages: trajectory prediction, collision detection and risk quantification.

In addition, Extended Kalman Filter is utilized to enhance the robustness of the predicted trajectory. Positioning uncertainty is taken into account in the process of collision detection. Extensive evaluation on diverse, challenging and realistic traffic scenarios enable us to fully validate the proposed risk assessment strategy.

In the next chapter, conclusions are drawn for all the research work presented in this thesis, and recommendation for the future work is discussed.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

A number of robust and computationally efficient vision based scene understanding techniques for collision avoidance on roadway have been proposed in this thesis. This is made possible by addressing the challenges of designing robust and low complexity algorithms for the key functional modules, namely, road surface detection, visual odometry, obstacle detection and tracking, and risk assessment.

An efficient non-parametric high-speed road surface detection algorithm that exploits the depth cue only has been proposed by formulating four special attributes that are observed in realistic road conditions. The proposed method is nonparametric and has paved the way for overcoming the limitations of existing parametric methods that cannot cope with cases where the road profile doesn't fit the pre-defined model or when the constantly varying road profiles cannot be modeled mathematically. In addition, the proposed algorithm is capable of accurately detecting both planar and non-planar road surfaces in various challenging scenarios with low computational complexity. Extensive experimental results using three challenging benchmarks (i.e. enpeda, KITTI, and Daimler) show that the proposed algorithm outperforms the baseline algorithms both in terms of detection accuracy (up to 23.12%) and runtime performance (up to 95.00%). The notable improvement to the runtime is mainly

due to the fact that the nonparametric technique exempts the proposed method from the computational intensive curve fitting techniques.

It has been shown that the proposed method for estimating the ego-motion of vehicle overcomes the limitations of existing solutions by integrating runtime-efficient strategies with robust techniques at various core stages in visual odometry. A novel pruning technique is adopted to notably reduce the computational complexity of detecting corner features without compromising on the quality of the extracted corner features. A robust and compute-efficient KLT tracker is proposed to facilitate the generation of the feature correspondences in a robust and runtime efficient way. The accuracy of extracted feature correspondences is improved by leveraging on egomotion prior to determine a better initial point for fast and accurate feature convergence during tracking and incorporating an automatic tracking failure detection scheme to exclude the feature correspondences with large tracking error. In addition, the computational complexity of the conventional KLT has been improved by setting the integration window size adaptively. With the accurate feature correspondences provided, Gaussian-Newton optimization scheme supported by an early RANSAC termination condition is shown to converge faster in the motion estimation process. The above contributions are integrated into a framework for fast and robust visual odometry. Extensive evaluation based on the KITTI odometry benchmark shows that the proposed visual odometry method outperforms the baseline algorithms both in terms of accuracy (up to 48.36%) and runtime performance. In addition, the proposed algorithm is placed among the top 15% when evaluated using the well-known KITTI odometry platform.

The proposed stereo-vision based obstacle detection and tracking method is shown to be both robust and of low complexity. Unlike the works that focus on detecting vehicle or pedestrian only, the proposed obstacle detection method relies on u-v disparity space to detect all obstacles in the scene. The practical nature of collision avoidance systems where only the objects that are not far from the ego-vehicle are of concern is leveraged on in order to reduce the computational complexity. A Space of Interest (SOI) is defined to remove irrelevant image regions to greatly reduce the search space of obstacle and reject some false positives at an early stage. Segmentation of SOI into set of obstacles relies on adaptive hysteresis thresholding technique to filter noise and locate the peak regions, and adaptive connected component labeling technique to group the peak regions into set of clusters. Unlike the classical connected component labeling algorithm which processes at the pixel level, the proposed approach processes based on u-span to significantly reduce the clustering complexity.

In order to track obstacle across frames, a distinctive object appearance model is constructed. Color histogram is employed due to its simplicity and its high tolerance to scale and view angle change and partial occlusion. In addition, the L*a*b* color space has ensured that the sensitivity of the model to illumination change can be reduced. The distinctiveness of the object model is further enhanced by excluding the pixels that belong to the background. Histogram intersection distance is utilized to measure the similarity between two given objects and this process is further improved by incorporating optical flow based motion information (derived from the visual odometry module) and distance information. A strategy to reduce the computational complexity for constructing the object model is also proposed by employing a chess-board pattern based sampling technique in the process of generating the histogram, which has resulted in approximately 50% less computations. Finally, the proposed obstacle detection and data association modules are integrated to form an online multi-object tracking framework in a robust way.

The integrated online tracking system has been shown to improve detection when obstacles are in close proximity. Using a widely-known and challenging benchmark, the proposed obstacle tracking algorithm has been demonstrated to be capable of not only tracking common moving or stationary obstacles like vehicles, pedestrians, bicyclists but also unexpected ones like traffic lights, sign posts, barriers, trees etc. simultaneously. In addition, the proposed algorithm has been shown to successfully detect and track obstacles in the presence of notable scale change, partial occlusion and inconsistent illumination. Evaluations using the KITTI tracking benchmark confirm that the proposed obstacle detection and tracking method outperforms the baseline algorithm in terms of tracking accuracy by up to 51.78%. In addition, compared to the baseline algorithm that achieves about 0.23 frame per second (fps), the proposed method lends well for real-time performance with 20 fps.

The proposed risk assessment module has been devised by customizing the Extended Kalman Filter to enhance the robustness of the predicted trajectories of each obstacle in the scene. The robustness of collision prediction has been enhanced by accommodating positioning uncertainty. Risk assessment evaluations based on the KITTI tracking dataset demonstrate that the proposed method is capable of robustly and efficiently assessing the collision risk in diverse traffic scenarios.

Finally, the integrated framework consisting of all the functional blocks, namely, road surface detection (Chapter 3), visual odometry (Chapter 4), obstacle detection (Chapter 5), obstacle tracking (Chapter 5), risk assessment (Chapter 6) provides for a holistic vision based scene understanding solution for real-time collision avoidance on roadway.

7.2 Future Work

The following lists the future research directions of this thesis:

- A nonparametric road surface detection algorithm with low computational complexity has been proposed which relies only on the disparity map as the input. Noting that the performance of the proposed method can be sensitive to the accuracy of the disparity map inputs, a possible research direction could be to explore suitable fusion of image inputs in order to improve the robustness of the proposed road surface detection method. One possibility is to incorporate color or intensity of images to provide additional valuable cues for complementing the depth inputs.

- The proposed visual odometry method in Chapter 4 requires a large number of features (up to 500) in order to guarantee that sufficient inlier features are generated for accurate ego-motion estimation. While it has been shown that the proposed technique is of low computational complexity compared to other existing methods, it will be of interest to explore ways to reduce the number of features in an attempt to further improve the runtime efficiency. One possible direction could be to consider the spatial distribution of the features during the feature selection process.

- In Chapter 5, the trajectories of the obstacles are modelled for risk assessment without knowledge of the obstacle type (e.g. vehicle/pedestrian). Noting that the object type can lead to more accurate modelling of the obstacles' behavior for enhancing the accuracy of the risk assessment process, it will be of interest to introduce object recognition to facilitate this. Such an object recognition method must cope with the extreme challenges in the realistic uncontrolled environment and be of low complexity to facilitate real-time computations.

- The proposed techniques in this thesis relies on one existing stereo matching algorithm called 'Semi-global matching'. While stereo vision is increasingly adopted in many automotive applications, existing stereo matching algorithms often fail to find a perfect balance between accuracy and speed. Hence, designing a robust and efficient stereo matching algorithm that is suitable for in-vehicle deployment is a worthy future research direction.

- The proposed vision based scene understanding techniques for collision avoidance in this thesis has been shown to produce high quality results on datasets with image

resolution 640*480 or 1240*376. It would be interesting to investigate techniques that can accommodate to images with lower resolution in order to further reduce the computational cost while still providing for acceptable quality of results.

- The techniques proposed in this thesis rely on visual camera for environment perception. Although visual camera are cheap and have been shown to be reliable in normal day time, it poses severe problems in extreme weather scenarios like night time, or heavy rain, fog and snow weather condition. With the continuous progress in electronic technologies, it is worth investigating solutions that are based on hybrid sensing technologies, e.g. fusion of camera with infrared camera or radar. The proposed techniques can be extended to leverage on the fusion of sensor data in order to cope with all challenging road scenarios and weather conditions.

- The novel motion prediction framework proposed for risk assessment in Chapter 6 has been shown to achieve good results in challenging traffic scenarios. The proposed model predicts the motion state of the ego-vehicle or obstacles vehicles by only relying on the laws of physics. While this allows for efficient computation of the collision risk, it is limited to only short-term collision prediction. The predicted motion of a vehicle should take into account other factors such as driver's status and intention, the road topology, traffic rules, etc. As such, there is scope for future work to design more advanced prediction models for computing the collision risk in a probabilistic manner that takes into account these factors. This will enhance the capability of computing the risk assessment under various traffic uncertainties.

- So far, all the proposed techniques are implemented in C++ and validated on an Intel 3.5 GHz machine. It will be of interest to port these techniques to a hybrid embedded computing platform in order to meet the real-time performance at low cost. This will necessitate constraint-aware hardware-software partitioning to accelerate the compute intensive modules by exploiting the inherent parallelism.

BIBLIOGRAPHY

[1] W. H. Organization, "Global Status Report on Road Safety 2013." http://www.who.int/violence_injury_prevention/road_safety_status/2013/en/, 2013.

[2] I. I. for Highway Safety Highway Loss Data InstituteI (IIHSHLDI), "Crash Avoidance Technologies Overview." http://www.iihs.org/iihs/topics/t/crash-avoidance-technologies/topicoverview.

[3] N. T. S. Board, "The Use of Forward Collision Avoidance Systems to Prevent and Mitigate Rear-End Crashes." http://www.ntsb.gov/safety/safety-studies/Pages/SIR1501.aspx, 2015.

[4] M. L. Aust, L. Jakobsson, M. Lindman, and E. Coelingh, "Collision avoidance systems-advancements and efficiency," tech. rep., SAE Technical Paper, 2015.

[5] A. Houénou, P. Bonnifait, and V. Cherfaoui, "Risk assessment for collision avoidance systems," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 386–391, IEEE, 2014.

[6] M. Wu, S.-K. Lam, and T. Srikanthan, "Nonparametric technique based high-speed road surface detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 874–884, 2015.

[7] N. Ramakrishnan, M. Wu, S.-K. Lam, and T. Srikanthan, "Enhanced low-complexity pruning for corner detection," *Journal of Real-Time Image Processing*, pp. 1–17, 2014.

[8] M. Wu, N. Ramakrishnan, S.-K. Lam, and T. Srikanthan, "Low-complexity pruning for accelerating corner detection," in *2012 IEEE International Symposium on Circuits and Systems*, pp. 1684–1687, IEEE, 2012.

[9] M. Wu, S.-K. Lam, and T. Srikanthan, "A framework for fast and robust visual odometry," *IEEE Transactions on Intelligent Transportation Systems*, 2016. Under Revision.

[10] M. Wu, S.-K. Lam, and T. Srikanthan, "Stereo based rois generation for detecting pedestrians in close proximity," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 1929–1934, IEEE, 2014.

[11] M. Wu, S.-K. Lam, T. Srikanthan, and T. Shah, "Vision-based pedestrian tracking system using color and motion cue," in *2014 International Symposium on Integrated Circuits (ISIC)*, pp. 372–375, IEEE, 2014.

[12] M. Wu, C. Zhou, and T. Srikanthan, "Robust and low complexity obstacle detection and tracking," in *19th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2016. Accepted.

[13] J. Golias, G. Yannis, and C. Antoniou, "Classification of driver-assistance systems according to their impact on road safety and traffic efficiency," *Transport reviews*, vol. 22, no. 2, pp. 179–196, 2002.

[14] A. Broggi, A. Zelinsky, M. Parent, and C. E. Thorpe, *Intelligent Vehicles*, pp. 1175–1198. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[15] M. Lu, K. Wevers, and R. Van Der Heijden, "Technical feasibility of advanced driver assistance systems (adas) for road traffic safety," *Transportation Planning and Technology*, vol. 28, no. 3, pp. 167–187, 2005.

[16] S. Graeme, B. Alex, L. Gaby, and P. Stewart, "Advanced Driver Assistance Systems Report," 2011.

[17] A. Berthelot, A. Tamke, T. Dang, and G. Breuel, "Stochastic situation assessment in advanced driver assistance system for complex multi-objects traffic situations," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1180–1185, IEEE, 2012.

[18] M. Durali, G. A. Javid, and A. Kasaiezadeh, "Collision avoidance maneuver for an autonomous vehicle," in *9th IEEE International Workshop on Advanced Motion Control, 2006.*, pp. 249–254, IEEE, 2006.

[19] D. F. Llorca, V. Milanés, I. P. Alonso, M. Gavilán, I. G. Daza, J. Pérez, and M. Á. Sotelo, "Autonomous pedestrian collision avoidance using a fuzzy steering controller," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 390–401, 2011.

[20] K. D. Kusano and H. C. Gabler, "Safety benefits of forward collision warning, brake assist, and autonomous braking systems in rear-end collisions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1546–1555, 2012.

[21] M. Althoff, O. Stursberg, and M. Buss, "Model-based probabilistic collision detection in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 2, pp. 299–310, 2009.

[22] D. G. Gomez, *A global approach to vision-based pedestrian detection for advanced driver assistance systems*. PhD thesis, Universitat Autònoma de Barcelona, 2009.

[23] D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 7, pp. 1239–1258, 2010.

[24] A. Lindgren, F. Chen, P. W. Jordan, and H. Zhang, "Requirements for the design of advanced driver assistance systems-the differences between swedish and chinese drivers," *International Journal of Design*, vol. 2, no. 2, 2008.

[25] B. Matthew, "Global market review of driver assistance systems - forecasts to 2017: 2010 edition: Chapter 2 The market." http://search.proquest.com/docview/213131187?accountid=12665, 2010.

[26] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 3354–3361, IEEE, 2012.

[27] A. Geiger, *Probabilistic Models for 3D Urban Scene Understanding from Movable platforms*. PhD thesis, Karlsruhe Institute of Technology, 2013.

[28] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, *et al.*, "Making bertha drive—an autonomous journey on a historic route," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014.

[29] T. I. T. S. of America (ITSA), "Advanced Driver Assistance and Autonomous Vehicles - Challenges and Opportunities." http://www.itsa.org/knowledgecenter/technology-assessment/driver-assistance-and-autonomous-vehicles, 2014.

[30] S. R. Kumar, *Embedded Computing Techniques For Vision-based Lane Change Decision Aid Systems*. PhD thesis, Nanyang Technological University, 2013.

[31] N. H. T. S. A. (NHTSA), "U.S. Department of Transportation Releases Policy on Automated Vehicle Development." http://www.nhtsa.gov/About+NHTSA/Press+Releases/U.S.+Department+of+Transportation+Releases+Policy+on+Automated+Vehicle+Development, 2013.

[32] A. Vahidi and A. Eskandarian, "Research advances in intelligent collision avoidance and adaptive cruise control," *IEEE transactions on intelligent transportation systems*, vol. 4, no. 3, pp. 143–153, 2003.

[33] Q. Baig, O. Aycard, T. D. Vu, and T. Fraichard, "Fusion between laser and stereo vision data for moving objects tracking in intersection like scenario," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 362–367, IEEE, 2011.

[34] A. Discant, A. Rogozan, C. Rusu, and A. Bensrhair, "Sensors for obstacle detection-a survey," in *2007 30th International Spring Seminar on Electronics Technology (ISSE)*, pp. 100–105, IEEE, 2007.

[35] W. H. Organization, "Denso's New Pre-Collision System Available on Lexus LS 430." http://www.atzonline.com/Aktuell/Nachrichten/1/2141/Denso-s-New-Pre-Collision-System-Available-on-Lexus-LS-430.html, 2004.

[36] "Honda Develops World's First 'Collision Mitigation Brake System' (CMS) for Predicting Rear-end Collisions and Controlling Brake Operations." http://world.honda.com/news/2003/4030520.html, 2003.

[37] ""BAS plus" -Mercedes-Benz's first forward warning collision system." http://techcenter.mercedes-benz.com/en_SG/bas_plus/detail.html.

[38] "Safety Based on Radar Technology – Audi Braking Guard." http://www.fourtitude.com/news/publish/Audi_News/article_4002.shtml, 2008.

[39] T. Gandhi and M. M. Trivedi, "Pedestrian protection systems: Issues, survey, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 413–430, 2007.

[40] D. M. Gavrila, "Sensor-based pedestrian protection," *IEEE Intelligent Systems*, vol. 16, no. 6, pp. 77–81, 2001.

[41] "Bosch Middle Range Radar Sensor." http://www.systemplus.fr/reverse-costing-reports/bosch-mid-range-radar-mrr-sensor/.

[42] A. Barth and U. Franke, "Estimating the driving state of oncoming vehicles from a moving platform using stereo vision," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 4, pp. 560–571, 2009.

[43] M. Muffert, D. Pfeiffer, and U. Franke, "A stereo-vision based object tracking approach at roundabouts," *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 2, pp. 22–32, 2013.

[44] "New Collision Warning with Auto Brake helps prevent rear-end collisions." https://www.media.volvocars.com/global/en-gb/media/pressreleases/12129, 2007.

[45] "Toyota Strengthens Efforts to Develop Safe Vehicles." http://www.toyota.co.jp/en/news/06/0825.html, 2006.

[46] "Toyota Launches Redesigned Crown Majesta in Japan." http://www.motor1.com/news/14665/toyota-launches-redesigned-crown-majesta-in-japan, 2009.

[47] "Toyota Adds to Pre-crash Safety Technologies." http://www.toyota.co.jp/en/news/09/0226.html, 2009.

[48] "Extensive safety in the new Audi A8." http://www.bosch-presse.de/presseforum/details.htm?txtID=4570&locale=en, 2010.

[49] "Driving Assistant Plus." http://www.bmw.com/com/en/newvehicles/7series/sedan/2012/showroom/driver_assistance/assistance.html#t=l, 2012.

[50] "Extended PRE-SAFE protection: Prevention is better than cure." http://media.daimler.com/dcmedia/0-921-1549267-1-1549456-1-0-0-1549717-0-0-11702-854934-0-1-0-0-0-0-0.html, 2013.

[51] "Volkswagen Passat(B8)." https://en.wikipedia.org/wiki/Volkswagen_Passat_(B8).

[52] "Seeking the truth with autonomous cars." http://www.volvocars.com/intl/about/our-innovation-brands/intellisafe/intellisafe-autopilot/news/seeking-the-truth-with-autonomous-cars, 2015.

[53] "Google Self-Driving Car Project." https://www.google.com/selfdrivingcar/.

[54] "Google Self-driving Car." https://en.wikipedia.org/wiki/Google_self-driving_car.

[55] "DARPA Grand Challenge." https://en.wikipedia.org/wiki/DARPA_Grand_Challenge.

[56] "DARPA Urban Challenge." http://archive.darpa.mil/grandchallenge/.

[57] A. Broggi, P. Medici, P. Zani, A. Coati, and M. Panciroli, "Autonomous vehicles control in the vislab intercontinental autonomous challenge," *Annual Reviews in Control*, vol. 36, no. 1, pp. 161–171, 2012.

[58] "VisLab BRAiVE." http://www.braive.vislab.it/index.php.

[59] "Velodyne announces 7999 dollar puck lidar sensor." http://www.sparpointgroup.com/news/vol12no37-velodyne-announces-puck-lidar-sensor.

[60] P. Marchal, M. Dehesa, D. Gavrila, M. Meinecke, N. Skellern, and R. Viciguerra, "Save-u. final report," *Information Society Technology Programme of the EU, Tech. Rep*, 2005.

[61] "BMW launch 2013." http://www.mobileye.com/markets/oem/oem-launches/bmw/bmw-launch-2013/, 2013.

[62] L. Hamilton, L. Humm, M. Daniels, and H. Yen, "The role of vision sensors in future intelligent vehicles," tech. rep., SAE Technical Paper, 2001.

[63] "Mobileye Digital CMOS Camera." http://www.mobileye.com/technology/development-evaluation-platforms/cameras/.

[64] "Datasheet for Micron's MT9V022 Image Sensor." http://www.datasheet-pdf.com/PDF/MT9V022-Datasheet-Micron-500017.

[65] N. Bernini, M. Bertozzi, L. Castangia, M. Patander, and M. Sabbatelli, "Real-time obstacle detection using stereo vision for autonomous ground vehicles: A survey," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 873–878, IEEE, 2014.

[66] A. Wedel, H. Badino, C. Rabe, H. Loose, U. Franke, and D. Cremers, "B-spline modeling of road surfaces with an application to free-space estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 4, pp. 572–583, 2009.

[67] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[68] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," *IEEE Transactions on Pattern analysis and machine intelligence*, vol. 25, no. 8, pp. 993–1008, 2003.

[69] U. R. Dhond and J. K. Aggarwal, "Structure from stereo-a review," *IEEE transactions on systems, man, and cybernetics*, vol. 19, no. 6, pp. 1489–1510, 1989.

[70] M. Bleyer and C. Breiteneder, "Stereo matching—state-of-the-art and research challenges," in *Advanced Topics in Computer Vision*, pp. 143–179, Springer, 2013.

[71] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2007.

[72] H. Hirschmuller and D. Scharstein, "Evaluation of stereo matching costs on images with radiometric differences," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 9, pp. 1582–1599, 2009.

[73] S. Birchfield and C. Tomasi, "A pixel dissimilarity measure that is insensitive to image sampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 4, pp. 401–406, 1998.

[74] H. Hirschmüller, P. R. Innocent, and J. Garibaldi, "Real-time correlation-based stereo vision with reduced border errors," *International Journal of Computer Vision*, vol. 47, no. 1-3, pp. 229–246, 2002.

[75] A. Ansar, A. Castano, and L. Matthies, "Enhanced real-time stereo using bilateral filtering," in *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, pp. 455–462, IEEE, 2004.

[76] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *European conference on computer vision*, pp. 151–158, Springer, 1994.

[77] A. F. Bobick and S. S. Intille, "Large occlusion stereo," *International Journal of Computer Vision*, vol. 33, no. 3, pp. 181–200, 1999.

[78] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: Theory and experiment," *IEEE transactions on Pattern analysis and Machine Intelligence*, vol. 16, no. 9, pp. 920–932, 1994.

[79] O. Veksler, "Fast variable window for stereo correspondence using integral images," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1, pp. I–556, IEEE, 2003.

[80] M. Gerrits and P. Bekaert, "Local stereo matching with segmentation-based outlier rejection," in *The 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*, pp. 66–66, IEEE, 2006.

[81] K.-J. Yoon and I. S. Kweon, "Adaptive support-weight approach for correspondence search.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 650–656, 2006.

[82] C. Richardt, D. Orr, I. Davies, A. Criminisi, and N. A. Dodgson, "Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid," in *European Conference on Computer Vision*, pp. 510–523, Springer, 2010.

[83] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 504–511, 2013.

[84] K. He, J. Sun, and X. Tang, "Guided image filtering," in *European conference on computer vision*, pp. 1–14, Springer, 2010.

[85] D. Min, J. Lu, and M. N. Do, "Joint histogram-based cost aggregation for stereo matching," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 10, pp. 2539–2545, 2013.

[86] Y. Ohta and T. Kanade, "Stereo by intra-and inter-scanline search using dynamic programming," *IEEE Transactions on pattern analysis and machine intelligence*, no. 2, pp. 139–154, 1985.

[87] S. Birchfield and C. Tomasi, "Depth discontinuities by pixel-to-pixel stereo," *International Journal of Computer Vision*, vol. 35, no. 3, pp. 269–293, 1999.

[88] J. Sun, N.-N. Zheng, and H.-Y. Shum, "Stereo matching using belief propagation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 25, no. 7, pp. 787–800, 2003.

[89] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3, pp. 15–18, IEEE, 2006.

[90] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér, "Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 492–504, 2009.

[91] S. Roy and I. J. Cox, "A maximum-flow formulation of the n-camera stereo correspondence problem," in *Computer Vision, 1998. Sixth International Conference on*, pp. 492–499, IEEE, 1998.

[92] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[93] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2, pp. 508–515, IEEE, 2001.

[94] L. Hong and G. Chen, "Segment-based stereo matching using graph cuts," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, pp. I–74, IEEE, 2004.

[95] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 328–341, 2008.

[96] S. K. Gehrig and U. Franke, "Improving stereo sub-pixel accuracy for long range stereo," in *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–7, IEEE, 2007.

[97] S. K. Gehrig, H. Badino, and U. Franke, "Improving sub-pixel accuracy for long range stereo," *Computer Vision and Image Understanding*, vol. 116, no. 1, pp. 16–24, 2012.

[98] S. K. Gehrig and C. Rabe, "Real-time semi-global matching on the cpu," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pp. 85–92, IEEE, 2010.

[99] M. Humenberger, T. Engelke, and W. Kubinger, "A census-based stereo vision algorithm using modified semi-global matching and plane fitting to improve matching quality," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pp. 77–84, IEEE, 2010.

[100] M. El Ansari, S. Mousset, and A. Bensrhair, "Temporal consistent real-time stereo for intelligent vehicles," *Pattern Recognition Letters*, vol. 31, no. 11, pp. 1226–1238, 2010.

[101] M. Gong, "Enforcing temporal consistency in real-time stereo estimation," in *European Conference on Computer Vision*, pp. 564–577, Springer, 2006.

[102] J. Davis, R. Ramamoorthi, and S. Rusinkiewicz, "Spacetime stereo: A unifying framework for depth from triangulation," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2, pp. II–359, IEEE, 2003.

[103] L. Zhang, B. Curless, and S. M. Seitz, "Spacetime stereo: Shape recovery for dynamic scenes," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2, pp. II–367, IEEE, 2003.

[104] H. Tao, H. S. Sawhney, and R. Kumar, "Dynamic depth recovery from multiple synchronized video streams," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. I–118, IEEE, 2001.

[105] S. K. Gehrig, F. Eberli, and T. Meyer, "A real-time low-power stereo vision engine using semi-global matching," in *International Conference on Computer Vision Systems*, pp. 134–143, Springer, 2009.

[106] "KITTI Vision Benchmarks." http://www.cvlibs.net/datasets/kitti/index.php.

[107] A. B. Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," *Machine vision and applications*, vol. 25, no. 3, pp. 727–745, 2014.

[108] C. Thorpe, M. H. Hebert, T. Kanade, and S. A. Shafer, "Vision and navigation for the carnegie-mellon navlab," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 3, pp. 362–373, 1988.

[109] A. Wedel, U. Franke, H. Badino, and D. Cremers, "B-spline modeling of road surfaces for freespace estimation," in *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 828–833, IEEE, 2008.

[110] B. Southall and C. J. Taylor, "Stochastic road shape estimation," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1, pp. 205–212, IEEE, 2001.

[111] Y. He, H. Wang, and B. Zhang, "Color-based road detection in urban traffic scenes," *IEEE Transactions on intelligent transportation systems*, vol. 5, no. 4, pp. 309–318, 2004.

[112] C. Rasmussen, "Grouping dominant orientations for ill-structured road following," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, pp. I–470, IEEE, 2004.

[113] H. Kong, J.-Y. Audibert, and J. Ponce, "Vanishing point detection for road detection," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 96–103, IEEE, 2009.

[114] J. M. Alvarez, T. Gevers, and A. M. Lopez, "3d scene priors for road detection," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 57–64, IEEE, 2010.

[115] R. Labayrade, D. Aubert, and J.-P. Tarel, "Real time obstacle detection in stereovision on non flat road geometry through" v-disparity" representation," in *Intelligent Vehicle Symposium, 2002. IEEE*, vol. 2, pp. 646–651, IEEE, 2002.

[116] T. Dang and C. Hoffmann, "Fast object hypotheses generation using 3d position and 3d motion," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops*, pp. 56–56, IEEE, 2005.

[117] S. J. Krotosky and M. M. Trivedi, "On color-, infrared-, and multimodal-stereo approaches to pedestrian detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 4, pp. 619–629, 2007.

[118] A. D. Sappa, F. Dornaika, D. Ponsa, D. Gerónimo, and A. López, "An efficient approach to onboard stereo vision system pose estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 476–490, 2008.

[119] N. Soquet, D. Aubert, and N. Hautiere, "Road segmentation supervised by an extended v-disparity algorithm for autonomous navigation," in *2007 IEEE Intelligent Vehicles Symposium*, pp. 160–165, IEEE, 2007.

[120] 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, 2010.

[121] S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, T. Graf, and R. Schmidt, "High accuracy stereovision approach for obstacle detection on non-planar roads," *Proc. IEEE INES*, pp. 211–216, 2004.

[122] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: Survey and experiments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 12, pp. 2179–2195, 2009.

[123] A. Mukhtar, L. Xia, and T. B. Tang, "Vehicle detection techniques for collision avoidance systems: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2318–2338, 2015.

[124] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1773–1795, 2013.

[125] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 5, pp. 694–711, 2006.

[126] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[127] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893, IEEE, 2005.

[128] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *International Journal of Computer Vision*, vol. 38, no. 1, pp. 15–33, 2000.

[129] A. Shashua, Y. Gdalyahu, and G. Hayun, "Pedestrian detection for driving assistance systems: Single-frame classification and system level performance," in *Intelligent Vehicles Symposium, 2004 IEEE*, pp. 1–6, IEEE, 2004.

[130] P. Sudowe and B. Leibe, "Efficient use of geometric constraints for sliding-window object detection in video," in *International Conference on Computer Vision Systems*, pp. 11–20, Springer, 2011.

[131] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 1491–1498, IEEE, 2006.

[132] W. Zhang, G. Zelinsky, and D. Samaras, "Real-time accurate object detection using multiple resolutions," in *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8, IEEE, 2007.

[133] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE transactions on pattern analysis and machine intelligence*, vol. 23, no. 4, pp. 349–361, 2001.

[134] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," 2009.

[135] W. Liu, X. Wen, B. Duan, H. Yuan, and N. Wang, "Rear vehicle detection and tracking for lane change assist," in *2007 IEEE Intelligent Vehicles Symposium*, pp. 252–257, IEEE, 2007.

[136] Z. Sun, G. Bebis, and R. Miller, "Monocular precrash vehicle detection: features and classifiers," *IEEE transactions on image processing*, vol. 15, no. 7, pp. 2019–2034, 2006.

[137] D. Ponsa, A. López, J. Serrat, F. Lumbreras, and T. Graf, "Multiple vehicle 3d tracking using an unscented kalman," in *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.*, pp. 1108–1113, IEEE, 2005.

[138] G. Y. Song, K. Y. Lee, and J. W. Lee, "Vehicle detection by edge-based candidate generation and appearance-based classification," in *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 428–433, IEEE, 2008.

[139] D. Withopf and B. Jahne, "Learning algorithm for real-time vehicle tracking," in *2006 IEEE Intelligent Transportation Systems Conference*, pp. 516–521, IEEE, 2006.

[140] A. Haselhoff, S. Schauland, and A. Kummert, "A signal theoretic approach to measure the influence of image resolution for appearance-based vehicle detection," in *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 822–827, IEEE, 2008.

[141] A. Haselhoff and A. Kummert, "A vehicle detection system based on haar and triangle features," in *Intelligent Vehicles Symposium, 2009 IEEE*, pp. 261–266, IEEE, 2009.

[142] S. Sivaraman and M. M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 267–276, 2010.

[143] X. Wang, T. X. Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 32–39, IEEE, 2009.

[144] P. Dollár, S. Belongie, and P. Perona, "The fastest pedestrian detector in the west.," in *BMVC*, vol. 2, p. 7, Citeseer, 2010.

[145] A. C. Cosma, R. Brehar, and S. Nedevschi, "Part-based pedestrian detection using hog features and vertical symmetry," in *Intelligent Computer Communication and Processing (ICCP), 2012 IEEE International Conference On*, pp. 229–236, IEEE, 2012.

[146] P. Geismann and G. Schneider, "A two-staged approach to vision-based pedestrian recognition using haar and hog features," in *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 554–559, IEEE, 2008.

[147] S. Sivaraman and M. M. Trivedi, "Active learning for on-road vehicle detection: a comparative study," *Machine vision and applications*, vol. 25, no. 3, pp. 599–611, 2014.

[148] M. Cheon, W. Lee, C. Yoon, and M. Park, "Vision-based vehicle detection system with consideration of the detecting location," *IEEE transactions on intelligent transportation systems*, vol. 13, no. 3, pp. 1243–1252, 2012.

[149] Y. Mu, S. Yan, Y. Liu, T. Huang, and B. Zhou, "Discriminative local binary patterns for human detection in personal album," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[150] J. Wu, C. Geyer, and J. M. Rehg, "Real-time human detection using contour cues," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 860–867, IEEE, 2011.

[151] S. Walk, N. Majer, K. Schindler, and B. Schiele, "New features and insights for pedestrian detection," in *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pp. 1030–1037, IEEE, 2010.

[152] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[153] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, "Cascade object detection with deformable part models," in *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pp. 2241–2248, IEEE, 2010.

[154] P. F. Felzenszwalb, R. B. 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, 2010.

[155] C. Tzomakas and W. von Seelen, "Vehicle detection in traffic scenes using shadows," in *IR-INI, INSTITUT FUR NUEROINFORMATIK, RUHR-UNIVERSITAT*, Citeseer, 1998.

[156] M. B. Van Leeuwen and F. C. Groen, "Vehicle detection with a mobile camera: spotting midrange, distant, and passing cars," *IEEE robotics & automation magazine*, vol. 12, no. 1, pp. 37–43, 2005.

[157] M. Bertozzi, A. Broggi, R. Chapuis, F. Chausse, A. Fascioli, and A. Tibaldi, "Shape-based pedestrian detection and localization," in *Procs. IEEE Intl. Conf. on Intelligent Transportation Systems 2003*, pp. 328–333, 2003.

[158] S. S. Teoh and T. Bräunl, "Symmetry-based monocular vehicle detection system," *Machine Vision and Applications*, vol. 23, no. 5, pp. 831–842, 2012.

[159] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[160] A. Bar-Hillel, D. Levi, E. Krupka, and C. Goldberg, "Part-based feature synthesis for human detection," in *European Conference on Computer Vision*, pp. 127–142, Springer, 2010.

[161] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[162] L. Zhao and C. E. Thorpe, "Stereo-and neural network-based pedestrian detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 3, pp. 148–154, 2000.

[163] S. Munder and D. M. Gavrila, "An experimental study on pedestrian classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 11, pp. 1863–1868, 2006.

[164] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

[165] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1995.

[166] M. Szarvas, A. Yoshizawa, M. Yamamoto, and J. Ogata, "Pedestrian detection with convolutional neural networks," in *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, pp. 224–229, IEEE, 2005.

[167] P. Sabzmeydani and G. Mori, "Detecting pedestrians by learning shapelet features," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2007.

[168] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *International Journal of Computer Vision*, vol. 63, no. 2, pp. 153–161, 2005.

[169] C. Wojek and B. Schiele, "A performance evaluation of single and multi-feature people detection," in *Joint Pattern Recognition Symposium*, pp. 82–91, Springer, 2008.

[170] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *European conference on computational learning theory*, pp. 23–37, Springer, 1995.

[171] M. Mählisch, M. Oberländer, O. Löhlein, D. Gavrila, and W. Ritter, "A multiple detector approach to low-resolution fir pedestrian recognition," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV2005), Las Vegas, NV, USA*, 2005.

[172] L. Zhang, B. Wu, and R. Nevatia, "Pedestrian detection in infrared images based on local shape features," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2007.

[173] Z. Lin and L. S. Davis, "A pose-invariant descriptor for human detection and segmentation," in *European Conference on Computer Vision*, pp. 423–436, Springer, 2008.

[174] P. Dollár, Z. Tu, H. Tao, and S. Belongie, "Feature mining for image classification," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2007.

[175] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[176] H. Badino, U. Franke, and R. Mester, "Free space computation using stochastic occupancy grids and dynamic programming," in *Workshop on Dynamical Vision, ICCV, Rio de Janeiro, Brazil*, vol. 20, 2007.

[177] H. Badino, U. Franke, and D. Pfeiffer, "The stixel world-a compact medium level representation of the 3d-world," in *Joint Pattern Recognition Symposium*, pp. 51–60, Springer, 2009.

[178] D. Pfeiffer and U. Franke, "Efficient representation of traffic scenes by means of dynamic stixels," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 217–224, IEEE, 2010.

[179] D. Pfeiffer and U. Franke, "Modeling dynamic 3d environments by means of the stixel world," *IEEE Intelligent Transportation Systems Magazine*, vol. 3, no. 3, pp. 24–36, 2011.

[180] D. Pfeiffer, F. Erbs, and U. Franke, "Pixels, stixels, and objects," in *European Conference on Computer Vision*, pp. 1–10, Springer, 2012.

[181] F. Erbs, B. Schwarz, and U. Franke, "From stixels to objects—a conditional random field based approach," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pp. 586–591, IEEE, 2013.

[182] M. Muffert, N. Schneider, and U. Franke, "Stix-fusion: a probabilistic stixel integration technique," in *Computer and Robot Vision (CRV), 2014 Canadian Conference on*, pp. 16–23, IEEE, 2014.

[183] T. Scharwächter, M. Enzweiler, U. Franke, and S. Roth, "Stixmantics: A medium-level model for real-time semantic scene understanding," in *European Conference on Computer Vision*, pp. 533–548, Springer, 2014.

[184] R. Danescu and S. Nedevschi, "A particle-based solution for modeling and tracking dynamic digital elevation maps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, pp. 1002–1015, 2014.

[185] C. D. Pantilie, S. Bota, I. Haller, and S. Nedevschi, "Real-time obstacle detection using dense stereo vision and dense optical flow," in *Intelligent Computer Communication and Processing (ICCP), 2010 IEEE International Conference on*, pp. 191–196, IEEE, 2010.

[186] S. Bota and S. Nedevschi, "Tracking multiple objects in urban traffic environments using dense stereo and optical flow," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 791–796, IEEE, 2011.

[187] W. Kruger, W. Enkelmann, and S. Rossle, "Real-time estimation and tracking of optical flow vectors for obstacle detection," in *Intelligent Vehicles' 95 Symposium., Proceedings of the*, pp. 304–309, IEEE, 1995.

[188] M. Nishigaki and Y. Aloimonos, "Moving obstacle detection using cameras for driver assistance system," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 805–812, IEEE, 2010.

[189] P. Lenz, J. Ziegler, A. Geiger, and M. Roser, "Sparse scene flow segmentation for moving object detection in urban environments," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 926–932, IEEE, 2011.

[190] A. Wedel, A. Meißner, C. Rabe, U. Franke, and D. Cremers, "Detection and segmentation of independently moving objects from dense scene flow," in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 14–27, Springer, 2009.

[191] A. Talukder, R. Manduchi, A. Rankin, and L. Matthies, "Fast and reliable obstacle detection and segmentation for cross-country navigation," in *Intelligent Vehicle Symposium, 2002. IEEE*, vol. 2, pp. 610–618, IEEE, 2002.

[192] A. Broggi, M. Buzzoni, M. Felisa, and P. Zani, "Stereo obstacle detection in challenging environments: the viac experience," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1599–1604, IEEE, 2011.

[193] M. Bertozzi, L. Bombini, A. Broggi, M. Buzzoni, E. Cardarelli, S. Cattani, P. Cerri, A. Coati, S. Debattisti, A. Falzoni, *et al.*, "Viac: An out of ordinary experiment," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 175–180, IEEE, 2011.

[194] A. Broggi, P. Cerri, S. Debattisti, M. C. Laghi, P. Medici, M. Panciroli, and A. Prioletti, "Proud-public road urban driverless test: Architecture and results," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 648–654, IEEE, 2014.

[195] H. P. Moravec, "Robot spatial perceptionby stereoscopic vision and 3d evidence grids," *Perception*, 1996.

[196] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems," in *Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, vol. 2, 2010.

[197] A. Broggi, S. Cattani, M. Patander, M. Sabbatelli, and P. Zani, "A full-3d voxel-based dynamic obstacle detection for urban scenario using stereo vision," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pp. 71–76, IEEE, 2013.

[198] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm computing surveys (CSUR)*, vol. 38, no. 4, p. 13, 2006.

[199] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[200] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2014.

[201] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1, pp. 798–805, IEEE, 2006.

[202] U. Franke, C. Rabe, H. Badino, and S. Gehrig, "6d-vision: Fusion of stereo and motion for robust environment perception," in *Joint Pattern Recognition Symposium*, pp. 216–223, Springer, 2005.

[203] C. Rabe, U. Franke, and S. Gehrig, "Fast detection of moving objects in complex scenarios," in *2007 IEEE Intelligent Vehicles Symposium*, pp. 398–403, IEEE, 2007.

[204] R. Danescu, C. Pantilie, F. Oniga, and S. Nedevschi, "Particle grid tracking system stereovision based obstacle perception in driving environments," *IEEE Intelligent Transportation Systems Magazine*, vol. 4, no. 1, pp. 6–20, 2012.

[205] Z. Wu, J. Zhang, and M. Betke, "Online motion agreement tracking.," in *BMVC*, 2013.

[206] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel, "A survey of appearance models in visual object tracking," *ACM transactions on Intelligent Systems and Technology (TIST)*, vol. 4, no. 4, p. 58, 2013.

[207] S. Salti, A. Cavallaro, and L. Di Stefano, "Adaptive appearance modeling for video tracking: Survey and evaluation," *IEEE Transactions on Image Processing*, vol. 21, no. 10, pp. 4334–4348, 2012.

[208] Y. Watanabe, P. Fabiani, and G. Le Besnerais, "Simultaneous visual target tracking and navigation in a gps-denied environment," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pp. 1–6, IEEE, 2009.

[209] D. Scaramuzza, "1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints," *International journal of computer vision*, vol. 95, no. 1, pp. 74–85, 2011.

[210] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, pp. I–652, IEEE, 2004.

[211] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part i: The first 30 years and fundamentals," *IEEE Robotics and Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[212] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part ii: Matching, robustness, optimization, and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012.

[213] C. Harris and M. Stephens, "A combined corner and edge detector.," in *Alvey vision conference*, vol. 15, p. 50, Citeseer, 1988.

[214] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pp. 593–600, IEEE, 1994.

[215] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision*, pp. 430–443, Springer, 2006.

[216] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 963–968, IEEE, 2011.

[217] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[218] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*, pp. 404–417, Springer, 2006.

[219] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[220] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on*, pp. 510–517, Ieee, 2012.

[221] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European conference on computer vision*, pp. 778–792, Springer, 2010.

[222] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "Brief: Computing a local binary descriptor very fast," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1281–1298, 2012.

[223] H. Badino, A. Yamamoto, and T. Kanade, "Visual odometry by multi-frame feature integration," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 222–229, 2013.

[224] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[225] A. R. Bruss and B. K. Horn, "Passive navigation," *Computer Vision, Graphics, and Image Processing*, vol. 21, no. 1, pp. 3–20, 1983.

[226] S. Carlsson, "Recursive estimation of ego-motion and scene structure from a moving platform," in *Proceedings of the 7th Scandinavian Conference on Image Analysis, Aalborg*, pp. 958–965, 1991.

[227] G. P. Stein, O. Mano, and A. Shashua, "A robust method for computing vehicle ego-motion," in *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, pp. 362–368, IEEE, 2000.

[228] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

[229] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, 2006.

[230] P. Corke, D. Strelow, and S. Singh, "Omnidirectional visual odometry for a planetary rover," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 4, pp. 4007–4012, IEEE, 2004.

[231] R. Goecke, A. Asthana, N. Pettersson, and L. Petersson, "Visual vehicle egomotion estimation using the fourier-mellin transform," in *2007 IEEE Intelligent Vehicles Symposium*, pp. 450–455, IEEE, 2007.

[232] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis, "Monocular visual odometry in urban environments using an omnidirectional camera," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2531–2538, IEEE, 2008.

[233] M. J. Milford and G. F. Wyeth, "Single camera vision-only slam on a suburban road network," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 3684–3689, IEEE, 2008.

[234] D. Scaramuzza and R. Siegwart, "Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles," *IEEE transactions on robotics*, vol. 24, no. 5, pp. 1015–1026, 2008.

[235] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Real time localization and 3d reconstruction," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1, pp. 363–370, IEEE, 2006.

[236] D. Scaramuzza, F. Fraundorfer, and R. Siegwart, "Real-time monocular visual odometry for on-road vehicles with 1-point ransac," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 4293–4299, IEEE, 2009.

[237] A. Pretto, E. Menegatti, and E. Pagello, "Omnidirectional dense large-scale mapping and navigation based on meaningful triangulation," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3289–3296, IEEE, 2011.

[238] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[239] M. J. Milford, G. F. Wyeth, and D. Prasser, "Ratslam: a hippocampal model for simultaneous localization and mapping," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 1, pp. 403–408, IEEE, 2004.

[240] B. Liang and N. Pears, "Visual navigation using planar homographies," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 1, pp. 205–210, IEEE, 2002.

[241] Q. Ke and T. Kanade, "Transforming camera geometry to a virtual downward-looking camera: Robust ego-motion estimation and ground-layer detection," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1, pp. I–390, IEEE, 2003.

[242] H. Wang, K. Yuan, W. Zou, and Q. Zhou, "Visual odometry based on locally planar ground assumption," in *2005 IEEE International Conference on Information Acquisition*, pp. 6–pp, IEEE, 2005.

[243] J. J. Guerrero, R. Martinez-Cantin, and C. Sagüés, "Visual map-less navigation based on homographies," *Journal of Robotic Systems*, vol. 22, no. 10, pp. 569–581, 2005.

[244] D. Scaramuzza, F. Fraundorfer, M. Pollefeys, and R. Siegwart, "Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 1413–1419, IEEE, 2009.

[245] L.-P. Morency and T. Darrell, "Stereo tracking using icp and normal flow constraint," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 4, pp. 367–372, IEEE, 2002.

[246] D. Demirdjian and R. Horaud, "Motion–egomotion discrimination and motion segmentation from image-pair streams," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 53–68, 2000.

[247] A. Mallet, S. Lacroix, and L. Gallo, "Position estimation in outdoor environments using pixel tracking and stereovision," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 4, pp. 3519–3524, IEEE, 2000.

[248] R. Mandelbaum, G. Salgian, and H. Sawhney, "Correlation-based estimation of ego-motion and structure from motion and stereo," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1, pp. 544–550, IEEE, 1999.

[249] L. Matthies and S. Shafer, "Error modeling in stereo navigation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 3, pp. 239–248, 1987.

[250] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone, "Rover navigation using stereo ego-motion," *Robotics and Autonomous Systems*, vol. 43, no. 4, pp. 215–229, 2003.

[251] W. van der Mark, D. Fontijne, L. Dorst, and F. C. Groen, "Vehicle ego-motion estimation with geometric algebra," in *Intelligent Vehicle Symposium, 2002. IEEE*, vol. 1, pp. 58–63, IEEE, 2002.

[252] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme.," in *Intelligent Vehicles Symposium*, pp. 486–492, 2010.

[253] H. Badino, "A robust approach for ego-motion estimation using a mobile stereo platform," in *Complex Motion*, pp. 198–208, Springer, 2007.

[254] H. Badino and T. Kanade, "A head-wearable short-baseline stereo system for the simultaneous estimation of structure and motion.," in *MVA*, pp. 185–189, 2011.

[255] P. H. Torr and D. W. Murray, "The development and comparison of robust methods for estimating the fundamental matrix," *International journal of computer vision*, vol. 24, no. 3, pp. 271–300, 1997.

[256] D. Nistér, "Preemptive ransac for live structure and motion estimation," *Machine Vision and Applications*, vol. 16, no. 5, pp. 321–329, 2005.

[257] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[258] M. Pollefeys, D. Nistér, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, *et al.*, "Detailed real-time urban 3d reconstruction from video," *International Journal of Computer Vision*, vol. 78, no. 2-3, pp. 143–167, 2008.

[259] J.-P. Tardif, M. George, M. Laverne, A. Kelly, and A. Stentz, "A new approach to vision-aided inertial navigation," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 4161–4168, IEEE, 2010.

[260] K. Konolige, M. Agrawal, and J. Sola, "Large-scale visual odometry for rough terrain," in *Robotics research*, pp. 201–212, Springer, 2010.

[261] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *The International Journal of Robotics Research*, vol. 30, no. 4, pp. 407–430, 2011.

[262] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *International workshop on vision algorithms*, pp. 298–372, Springer, 1999.

[263] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *Robomech Journal*, vol. 1, no. 1, p. 1, 2014.

[264] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *Information Fusion, 2008 11th International Conference on*, pp. 1–6, IEEE, 2008.

[265] S. Ammoun and F. Nashashibi, "Real time trajectory prediction for collision risk estimation between vehicles," in *Intelligent Computer Communication and Processing, 2009. ICCP 2009. IEEE 5th International Conference on*, pp. 417–422, IEEE, 2009.

[266] N. Kaempchen, K. Weiss, M. Schaefer, and K. C. Dietmayer, "Imm object tracking for high dynamic driving maneuvers," in *Intelligent Vehicles Symposium, 2004 IEEE*, pp. 825–830, IEEE, 2004.

[267] J. Hillenbrand, A. M. Spieker, and K. Kroschel, "A multilevel collision mitigation approach—its situation assessment, decision making, and performance tradeoffs," *IEEE Transactions on intelligent transportation systems*, vol. 7, no. 4, pp. 528–540, 2006.

[268] A. Polychronopoulos, M. Tsogas, A. J. Amditis, and L. Andreone, "Sensor fusion for predicting vehicles' path for collision avoidance systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 549–562, 2007.

[269] R. Miller and Q. Huang, "An adaptive peer-to-peer collision warning system," in *Vehicular technology conference, 2002. VTC Spring 2002. IEEE 55th*, vol. 1, pp. 317–321, IEEE, 2002.

[270] A. Barth and U. Franke, "Where will the oncoming vehicle be the next second?," in *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 1068–1073, IEEE, 2008.

[271] A. Barth and U. Franke, "Tracking oncoming and turning vehicles at intersections," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pp. 861–868, IEEE, 2010.

[272] T. Batz, K. Watson, and J. Beyerer, "Recognition of dangerous situations within a cooperative group of vehicles," in *Intelligent Vehicles Symposium, 2009 IEEE*, pp. 907–912, IEEE, 2009.

[273] P. Lytrivis, G. Thomaidis, and A. Amditis, "Cooperative path prediction in vehicular environments," in *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pp. 803–808, IEEE, 2008.

[274] G. Welch and G. Bishop, "An introduction to the kalman filter," tech. rep., Chapel Hill, NC, USA, 1995.

[275] S. Chen, "Kalman filter for robot vision: a survey," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4409–4420, 2012.

[276] M. Bertozzi, A. Broggi, A. Fascioli, A. Tibaldi, R. Chapuis, and F. Chausse, "Pedestrian localization and tracking system with kalman filtering," in *Intelligent Vehicles Symposium, 2004 IEEE*, pp. 584–589, IEEE, 2004.

[277] D. M. Gavrila and S. Munder, "Multi-cue pedestrian detection and tracking from a moving vehicle," *International journal of computer vision*, vol. 73, no. 1, pp. 41–59, 2007.

[278] D. M. Gavrila, J. Giebel, and S. Munder, "Vision-based pedestrian detection: The protector system," in *Intelligent Vehicles Symposium, 2004 IEEE*, pp. 13–18, IEEE, 2004.

[279] G. Grubb, A. Zelinsky, L. Nilsson, and M. Rilbe, "3d vision sensing for improved pedestrian safety," in *Intelligent Vehicles Symposium, 2004 IEEE*, pp. 19–24, IEEE, 2004.

[280] A. E. Nordsjo, "A constrained extended kalman filter for target tracking," in *Radar Conference, 2004. Proceedings of the IEEE*, pp. 123–127, IEEE, 2004.

[281] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *AeroSense'97*, pp. 182–193, International Society for Optics and Photonics, 1997.

[282] J. Lou, H. Yang, W. M. Hu, and T. Tan, "Visual vehicle tracking using an improved ekf," in *Proc. Asian Conf. Computer Vision*, pp. 296–301, 2002.

[283] N. Schneider and D. M. Gavrila, "Pedestrian path prediction with recursive bayesian filters: A comparative study," in *German Conference on Pattern Recognition*, pp. 174–183, Springer, 2013.

[284] M. Meuter, U. Iurgel, S.-B. Park, and A. Kummert, "The unscented kalman filter for pedestrian tracking from a moving host," in *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 37–42, IEEE, 2008.

[285] C. Liu, P. Shui, and S. Li, "Unscented extended kalman filter for target tracking," *Journal of Systems Engineering and Electronics*, vol. 22, no. 2, pp. 188–192, 2011.

[286] J. J. Laviola, "A comparison of unscented and extended kalman filtering for estimating quaternion motion," in *American Control Conference, 2003. Proceedings of the 2003*, vol. 3, pp. 2435–2440, IEEE, 2003.

[287] R. Danescu, F. Oniga, and S. Nedevschi, "Modeling and tracking the driving environment with a particle-based occupancy grid," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1331–1342, 2011.

[288] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 425–437, 2002.

[289] A. Vatavu, R. Danescu, and S. Nedevschi, "Tracking multiple objects in traffic scenarios using free-form obstacle delimiters and particle filters," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pp. 1346–1351, IEEE, 2013.

[290] N. Kaempchen, B. Schiele, and K. Dietmayer, "Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 4, pp. 678–687, 2009.

[291] M. Brannstrom, E. Coelingh, and J. Sjoberg, "Model-based threat assessment for avoiding arbitrary vehicle collisions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 658–669, 2010.

[292] J. Hillenbrand, A. Spieker, and K. Kroschel, "Efficient decision making for a multilevel collision mitigation system," in *2006 IEEE Intelligent Vehicles Symposium*, pp. 460–465, IEEE, 2006.

[293] R. Horst, "Time-to-collision as a cue for decision-making in braking," *VISION IN VEHICLES–III*, 1991.

[294] C.-Y. Chan, "Defining safety performance measures of driver-assistance systems for intersection left-turn conflicts," in *2006 IEEE Intelligent Vehicles Symposium*, pp. 25–30, IEEE, 2006.

[295] F. Seeliger, G. Weidl, D. Petrich, F. Naujoks, G. Breuel, A. Neukum, and K. Dietmayer, "Advisory warnings based on cooperative perception," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 246–252, IEEE, 2014.

[296] A. Eidehall and L. Petersson, "Statistical threat assessment for general road scenes using monte carlo sampling," *IEEE Transactions on intelligent transportation systems*, vol. 9, no. 1, pp. 137–147, 2008.

[297] G. S. Aoude, B. D. Luders, K. K. Lee, D. S. Levine, and J. P. How, "Threat assessment design for driver assistance system at intersections," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pp. 1855–1862, IEEE, 2010.

[298] O. Aycard, Q. Baig, S. Bota, F. Nashashibi, S. Nedevschi, C. Pantilie, M. Parent, P. Resende, and T.-D. Vu, "Intersection safety using lidar and stereo vision sensors," in *IV'2011-IEEE Intelligent Vehicles Symposium*, pp. 863–869, 2011.

[299] N. Saunier, T. Sayed, and C. Lim, "Probabilistic collision prediction for vision-based automated road safety analysis," in *2007 IEEE Intelligent Transportation Systems Conference*, pp. 872–878, IEEE, 2007.

[300] G. R. de Campos, A. H. Runarsson, F. Granum, P. Falcone, and K. Alenljung, "Collision avoidance at intersections: A probabilistic threat-assessment and decision-making system for safety interventions," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 649–654, IEEE, 2014.

[301] Z. Chen, C. Wu, N. Lyu, G. Liu, and Y. He, "Pedestrian-vehicular collision avoidance based on vision system," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 11–15, IEEE, 2014.

[302] E. Raphael, R. Kiefer, P. Reisman, and G. Hayon, "Development of a camera-based forward collision alert system," *SAE International Journal of Passenger Cars-Mechanical Systems*, vol. 4, no. 2011-01-0579, pp. 467–478, 2011.

[303] B.-C. Chen, C.-W. Shih, and Y. Lin, "Design of forward collision warning system using estimated relative acceleration and velocity vector," tech. rep., SAE Technical Paper, 2014.

[304] T. Hsu and G. Nusholtz, "Kinematic fcw system modeling and application for fcw warning strategy evaluation," tech. rep., SAE Technical Paper, 2011.

[305] D. J. LeBlanc, R. J. Kiefer, R. K. Deering, M. A. Shulman, M. D. Palmer, and J. Salinger, "Forward collision warning: Preliminary requirements for crash alert timing," tech. rep., SAE Technical Paper, 2001.

[306] E. Coelingh, L. Jakobsson, H. Lind, and M. Lindman, "Collision warning with auto brake: a real-life safety perspective," *Innovations for Safety: Opportunities and Challenges*, 2007.

[307] M. Brännström, E. Coelingh, and J. Sjöberg, "Decision-making on when to brake and when to steer to avoid a collision," *International Journal of Vehicle Safety 1*, vol. 7, no. 1, pp. 87–106, 2014.

[308] A. Khanafer, D. Balzer, and R. Isermann, "A rule-based collision avoidance system–scene interpretation, strategy selection, path planning and system intervention," *SAE International Journal of Passenger Cars-Mechanical Systems*, vol. 2, no. 2009-01-0156, pp. 389–397, 2009.

[309] S. A. Kanarachos, "A new method for computing optimal obstacle avoidance steering manoeuvres of vehicles," *International Journal of Vehicle Autonomous Systems*, vol. 7, no. 1-2, pp. 73–95, 2009.

[310] C. G. Keller, M. Enzweiler, M. Rohrbach, D. F. Llorca, C. Schnorr, and D. M. Gavrila, "The benefits of dense stereo for pedestrian detection," *IEEE transactions on intelligent transportation systems*, vol. 12, no. 4, pp. 1096–1106, 2011.

[311] A. Broggi, C. Caraffi, P. P. Porta, and P. Zani, "The single frame stereo vision system for reliable obstacle detection used during the 2005 darpa grand challenge on terramax," in *2006 IEEE Intelligent Transportation Systems Conference*, pp. 745–752, IEEE, 2006.

[312] T. Vaudrey, C. Rabe, R. Klette, and J. Milburn, "Differences between stereo and motion behaviour on synthetic and real-world stereo sequences," in *2008 23rd International Conference Image and Vision Computing New Zealand*, pp. 1–6, IEEE, 2008.

[313] C. G. Keller, M. Enzweiler, and D. M. Gavrila, "A new benchmark for stereo-based pedestrian detection," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 691–696, IEEE, 2011.

[314] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[315] G. Sibley, L. Matthies, and G. Sukhatme, "Bias reduction and filter convergence for long range stereo," in *Robotics Research*, pp. 285–294, Springer, 2007.

[316] M. Hwangbo, J.-S. Kim, and T. Kanade, "Gyro-aided feature tracking for a moving camera: fusion, auto-calibration and gpu implementation," *The International Journal of Robotics Research*, vol. 30, no. 14, pp. 1755–1774, 2011.

[317] J.-Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel Corporation*, vol. 5, no. 1-10, p. 4, 2001.

[318] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[319] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-backward error: Automatic detection of tracking failures," in *Pattern recognition (ICPR), 2010 20th international conference on*, pp. 2756–2759, IEEE, 2010.

[320] A. Björck, *Numerical methods for least squares problems*. Siam, 1996.

[321] H. Zhang, A. Geiger, and R. Urtasun, "Understanding high-level semantics by modeling traffic patterns," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3056–3063, 2013.

[322] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3d traffic scene understanding from movable platforms," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 5, pp. 1012–1025, 2014.

[323] C. G. Keller and D. M. Gavrila, "Will the pedestrian cross? a study on pedestrian path prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 494–506, 2014.

[324] H. Cho, P. E. Rybski, A. Bar-Hillel, and W. Zhang, "Real-time pedestrian detection with deformable part models," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pp. 1035–1042, IEEE, 2012.

[325] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[326] L. He, Y. Chao, and K. Suzuki, "A run-based two-scan labeling algorithm," *IEEE Transactions on Image Processing*, vol. 17, no. 5, pp. 749–756, 2008.

[327] R. S. Hunter, "Accuracy, precision, and stability of new photoelectric color-difference meter," in *Journal of the Optical Society of America*, vol. 38, pp. 1094–1094, AMER INST PHYSICS CIRCULATION FULFILLMENT DIV, 500 SUNNYSIDE BLVD, WOODBURY, NY 11797-2999, 1948.

[328] C. Zinner, M. Humenberger, K. Ambrosch, and W. Kubinger, "An optimized software-based implementation of a census-based stereo matching algorithm," in *International Symposium on Visual Computing*, pp. 216–227, Springer, 2008.

[329] N. Y.-C. Chang, T.-H. Tsai, B.-H. Hsu, Y.-C. Chen, and T.-S. Chang, "Algorithm and architecture of disparity estimation with mini-census adaptive support weight," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 6, pp. 792–805, 2010.

[330] W. S. Fife and J. K. Archibald, "Improved census transforms for resource-optimized stereo vision," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 1, pp. 60–73, 2013.

[331] W. S. Fife, *Improved Stereo Vision Methods for FPGA-Based Computing Platforms*. PhD thesis, Brigham Young University-Provo, 2011.

[332] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[333] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2411–2418, 2013.

[334] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, pp. 1–10, 2008.

[335] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: Hybridboosted multi-target tracker for crowded scene," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2953–2960, IEEE, 2009.

[336] P. S. Maybeck, "The kalman filter: An introduction to concepts," in *Autonomous robot vehicles*, pp. 194–204, Springer, 1990.

[337] Y. Zhang, E. K. Antonsson, and K. Grote, "A new threat assessment measure for collision avoidance systems," in *2006 IEEE Intelligent Transportation Systems Conference*, pp. 968–975, IEEE, 2006.