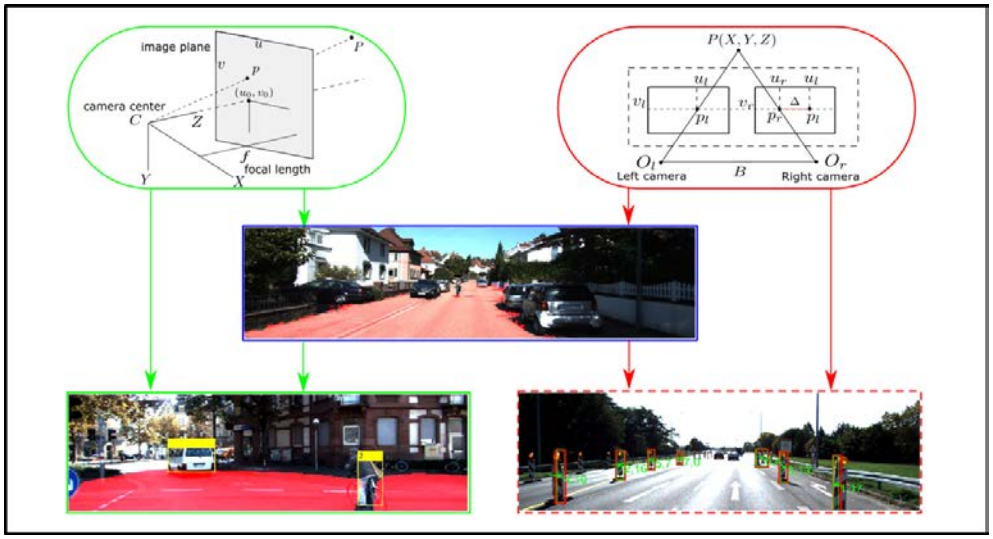
Geometrical and contextual scene analysis for object detection and tracking in intelligent vehicles

Bihao Wang

Par **Bihao WANG**

Geometrical and contextual scene analysis for object detection and tracking in intelligent vehicles

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 08 juillet 2015
Spécialité : Information and Systems Technologies

D2197

Université de Technologie de Compiègne

Thesis

Submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Area of specialization: Information and Systems Technologies

by

Bihao WANG

Geometrical and Contextual Scene Analysis for Object Detection and Tracking in Intelligent Vehicles

Heudiasyc Laboratory, UMR UTC/CNRS 7253

Defended on the 8th July

Thesis Committee:

Rapporteurs:		
	Samia Bouchafa-Bruneau	Prof. Université d'Evry Val d'Essonne, IBISC
	Michel Devy	DR CNRS, LAAS-CNRS
Examinateurs:		
	Pascal Vasseur	Prof. Université de ROUEN, EA LITIS
	Dominique Gruyer	DR IFSTTAR, LIVIC
	Véronique Berge Cherfaoui	MCF-HDR, UTC, HEUDIASYC
Directeurs de these:		
	Vincent Frémont	MCF-HDR, UTC, HEUDIASYC
	Sergio Alberto Rodriguez Florez	MCF, Université Paris-Saclay, IEF

Université de Technologie de Compiègne

Thèse

pour obtenir le grade de

Docteur

Spécialité: Technologie de l'information et des systèmes

par

Bihao WANG

Analyse de Scène Contextuelle et Géométrique pour la Détection et le Suivi d'Objets dans les Véhicules Intelligents

Laboratoire Heudiasyc, Unité Mixte de Recherche UTC/CNRS 7253

Soutenue le 8 Juillet

Devant le jury constitué de :

Rapporteurs:		
	Samia Bouchafa-Bruneau	Prof. Université d'Evry Val d'Essonne, IBISC
	Michel Devy	DR CNRS, LAAS-CNRS
Examinateurs:		
	Pascal Vasseur	Prof. Université de ROUEN, EA LITIS
	Dominique Gruyer	DR IFSTTAR, LIVIC
	Véronique Berge Cherfaoui	MCF-HDR, UTC, HEUDIASYC
Directeurs de these:		
	Vincent Frémont	MCF-HDR, UTC, HEUDIASYC
	Sergio Alberto Rodriguez Florez	MCF, Université Paris-Saclay, IEF

Acknowledgments

I sincerely wish to extend my heartfelt thanks to my supervisors Mr. Vincent Frémont and Mr. Sergio Alberto Rodriguez Florez who guided me into this fascinating field of computer vision. They also show me the significance and practical interest of this work that contribute to change our daily life . The whole thesis is a big challenge for me, accompanying with a lot of difficulties especially in the last year. But as an old saying goes: no pains no gains. I have learned how to analyze and solve the problem in a scientific and efficient way from my supervisors, . I am also grateful to China Scholar Censorship who provided me a financial support so that I can be fully devoted in my studies and research. Thanks to my friends, they taught me how to handle pressure and keep optimistic. Thanks to Madame Catherine, her support is precious and indispensable for me to get through my hardest period in the process of achieving my dreams while maintaining a healthy mental condition. I feel lucky that the administration jury from UTC accepted my application five years ago so that I could have the opportunity to live and study in this beautiful place. I learned painting and sculpture in art studio organized by CROUS, I enjoyed my sport life thanks to sport UTC and SNC. There are so many beautiful memories here, no matter where I am going to be, Compiègne will always be a place that I remember and miss.

Last but not the least, I would like to thank my parents for supporting me all these years. Their endless love is the fountain of my courage to pursue my dream.

Abstract

For autonomous or semi-autonomous intelligent vehicles, perception constitutes the first fundamental task to be performed before decision and action/control. Through the analysis of video, Lidar and radar data, it provides a specific representation of the environment and of its state, by extracting key properties from sensor data with time integration of sensor information. Compared to other perception modalities such as GPS, inertial or range sensors (Lidar, radar, ultrasonic), the cameras offer the greatest amount of information. Thanks to their versatility, cameras allow intelligent systems to achieve both high-level contextual and low-level geometrical information about the observed scene, and this is at high speed and low cost. Furthermore, the passive sensing technology of cameras enables low energy consumption and facilitates small-size system integration. The use of cameras is however, not trivial and poses a number of theoretical issues related to how this sensor perceives its environment.

In this thesis, we propose a vision-only system for moving object detection. Indeed, within natural and constrained environments observed by an intelligent vehicle, moving objects represent high risk collision obstacles, and have to be handled robustly. We approach the problem of detecting moving objects by first extracting the local context using a color-based road segmentation. After transforming the color image into illuminant invariant image, shadows as well as their negative influence on the detection process can be removed. Hence, according to the feature automatically selected on the road, a region of interest (ROI), where the moving objects can appear with a high collision risk, is extracted. Within this area, the moving pixels are then identified using a plane+parallax approach. To this end, the potential moving and parallax pixels are detected using a background subtraction method; then three different geometrical constraints: the epipolar constraint, the structural consistency constraint and the trifocal tensor are applied to such potential pixels to filter out parallax ones. Likelihood equations are also introduced to combine the constraints in a complementary and effective way. When stereo vision is available, the road segmentation and on-road obstacles detection can be refined by means of the disparity map with geometrical cues. Moreover, in this case, a robust tracking algorithm combining image and depth information has been proposed. If one of the two cameras fails, the system can therefore come back

to a monocular operation mode, which is an important feature for perception system reliability and integrity.

The different proposed algorithms have been tested on public images dataset with an evaluation against state-of-the-art approaches and ground-truth data. The obtained results are promising and show that the proposed methods are effective and robust on the different traffic scenarios and can achieve reliable detections in ambiguous situations.

Résumé

Pour les véhicules intelligents autonomes ou semi-autonomes, la perception constitue la première tâche fondamentale à accomplir avant la décision et l'action. Grâce à l'analyse des données vidéo, Lidar et radar, elle fournit une représentation spécifique de l'environnement et de son état, à travers l'extraction de propriétés clés issues des données des capteurs. Comparé à d'autres modalités de perception telles que le GPS, les capteurs inertiels ou les capteur de distance (Lidar, radar, ultrasons), les caméras offrent la plus grande quantité d'informations. Grâce à leur polyvalence, les caméras permettent aux systèmes intelligents d'extraire à la fois des informations contextuelles de haut niveau et de reconstruire des informations géométriques de la scène observée et ce, à haute vitesse et à faible coût. De plus, la technologie de détection passive des caméras permet une faible consommation d'énergie et facilite leur miniaturisation. L'utilisation des caméras n'est toutefois pas triviale et pose un certain nombre de questions théoriques liées à la façon dont ce capteur perçoit son environnement.

Dans cette thèse, nous proposons un système de détection d'objets mobiles basé seulement sur l'analyse d'images. En effet, dans les environnements observées par un véhicule intelligent, les objets en mouvement représentent des obstacles avec un risque de collision élevé, et ils doivent être détectés de manière fiable et robuste. Nous abordons le problème de la détection d'objets mobiles à partir de l'extraction du contexte local reposant sur une segmentation de la route. Après transformation de l'image couleur en une image invariante à l'illumination, les ombres peuvent alors être supprimées réduisant ainsi leur influence négative sur la détection d'obstacles. Ainsi, à partir d'une sélection automatique de pixels appartenant à la route, une région d'intérêt où les objets en mouvement peuvent apparaître avec un risque de collision élevé, est extraite. Dans cette zone, les pixels appartenant à des objets mobiles sont ensuite identifiés à l'aide d'une approche plan+parallaxe. À cette fin, les pixels potentiellement mobiles et liés à l'effet de parallaxe sont détectés par une méthode de soustraction du fond de l'image; puis trois contraintes géométriques différentes: la contrainte épipolaire, la contrainte de cohérence structurelle et le tenseur trifocal, sont appliquées à ces pixels pour filtrer ceux issus de l'effet de parallaxe. Des équations de vraisemblance sont aussi proposées afin de combiner les différents contraintes d'une manière

complémentaire et efficace. Lorsque la stéréovision est disponible, la segmentation de la route et la détection d'obstacles peuvent être affinées en utilisant une segmentation spécifique de la carte de disparité. De plus, dans ce cas, un algorithme de suivi robuste combinant les information de l'image et la profondeur des pixels a été proposé. Ainsi, si l'une des deux caméras ne fonctionne plus, le système peut donc revenir dans un mode de fonctionnement monoculaire, ce qui constitue une propriété importante pour la fiabilité et l'intégrité du système de perception.

Les différents algorithmes proposés ont été testés sur des bases de données d'images publiques en réalisant une évaluation par rapport aux approches de l'état de l'art et en se comparant à des données de vérité terrain. Les résultats obtenus sont prometteurs et montrent que les méthodes proposées sont efficaces et robustes pour différents scénarios routiers et les détections s'avèrent fiables notamment dans des situations ambiguës.

Contents

List of Symbols

Photometrical Quantities

I	Photometrical matrix representation of an image
I	Overall Intensity of the image
R_i	RGB channel values of an color image, $i = R, G, B$
λ	Wave length of the light
$\mathrm{E}(\lambda)$	Spectral power distribution
$\mathrm{S}(\lambda)$	Surface spectral reflectance function
$\mathrm{Q}_i(\lambda)$	Sensitivity of the camera for each channel, $i = R, G, B$
T	Light temperature
ϱ	Lambertian shading
c_i	Chromaticity

Mathematical Notations

e	Euler's number
η	Entropy of axis-calibration
$\mathcal{P}$	Probability density
$\mathcal{L}$	Likelihood
$\mathcal{N}$	Normal distribution
$\mathcal{H}$	Normalized Histogram
χ^2	Chi-square distribution
$\mu,\ \sigma$	Mean and standard deviation
$c\nu$	Coefficinet of variation
∇	Gradient operator
$\delta(\cdot)$	Dirac delta function
$\delta_{i,j}$	Kronecker delta
$\mathbf{I}$	Identity matrix
$\mathbf{A},\ \mathbf{B}$	Matrices
$\mathbf{a},\ \mathbf{b}$	Vectors

$a,\ b,\ c$	Scalar value
$\times$	Cross-product operator
$[\cdot]_\times$	The skew-symmetric matrix form

Computer Vision Notations

$\theta,\ \phi,\ \beta,\ \psi$	Angle variables
$\boldsymbol{p}$, $\boldsymbol{x}$	A generic point in the image plane
$\boldsymbol{P}$	A generic point in the world space
$\boldsymbol{l}$	A generic line in the image plane
$\boldsymbol{L}$	A generic line in the world space
$\mathit{\Pi}$	A generic plane in the world space
$(u,\ v)$	Pixel image coordinates
$(X,\ Y,\ Z)$	The world coordinates
f	The focal length of the camera
B	Baseline of two camera centers
$\boldsymbol{O}$	Origin of image coordinates
Δ	Disparity value
I_Δ	Disparity map
$I_\Delta(u, v)$	Pixel value of the disparity map I_Δ at the coordinates (u, v)
i_p	Intensity of a pixel in U-V-disparity map
$\mathbf{K}$	The intrinsic parameters matrix
$\mathbf{P}$	The projective camera matrix
$\mathbf{H}$	The homography matrix
$\mathbf{F}$	The fundamental matrix
$\mathbf{G}$	The structure consistency matrix
$\mathbf{T}$	Tensor matrix
$\mathcal{T}$	Trifocal tensor
$\mathbf{R},\ \mathbf{t}$	Rotation matrix and translation vector
Tr	Camera motion matrix $[\mathbf{R} \mid \mathbf{t}]$
κ	The projective depth
$\widetilde{\boldsymbol{P}}$	The projective structure
$\boldsymbol{e}$	The epipole
$\{\boldsymbol{p}_1, \boldsymbol{p}_2\}$	Corresponding points set
$\{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3\}$	Triplets set
d	Geometric distance
r	Residual of geometric constraints

Road Detection

I_R	Road surface detection result from illuminant invariant image
I_G	Ground plane detection result from disparity map
I_{final}	The free road surface detection result
I_{ROI}	The driving space where the obstacles need to be detected

Object Detection and Tracking

M	Binary map of moving object detection result
$\mathbf{L}$	Connected region in binary image
O	Detected obsacle
k	Discrete frame index
τ	Discrete index of targets
T	Time interval between successive frames
T_{ass}	Time period during which unassociated trackers are preserved
$w,\ h$	Width and height of obstacles
$N_{obs},\ N_S$	Number of obstacles, Number of particles
$Z(k)$	Observation or measurement
$S(k)$	Condensation state vector of the tracker
$s^i(k)$	Particles/samples
$s^i(k+1 \mid k)$	Predicted or expected state of each particle/sample
$\pi^i(k)$	The normalized weight for each sample
$\boldsymbol{f}(\cdot)$	State evolution function of dynamic model
$W(k)$	The noise vector
(x, y)	The position of obsatcle's centroid in image plane
G_O	The gate value for obstacle O
d_{TT}	The target-to-track distance
c	The regularization coefficients vector of noise $W(k)$

Acronyms

IV	Intelligent Vehicles
ADAS	Advanced Driver Assistance Systems
AE	Average Error
AP	Average Precision
BEV	Bird Eye View
BL	Base Line
BM	Binary Map
CM	Confidence Map
CNN	Convolutional Neutral Network
CV	Coefficients of Variation
DLT	Direct Linear Transformation
FN	False Negative
FP	False Positive
FOV	Filed Of View
GNN	Global Nearest Neighbor
IPM	Inverse Perspective Mapping
LMedS	Least Median of Squares
MCMC	Markov Chain Monte Carlo
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimation
MSE	Mean Square Error
MTT	Multiple Target Tracking
NN	Nearest Neighbor
RANSAC	Random Sample Consensus
ROI	Region Of Interest
SfM	Structure from Motion
SLAM	Simultaneous Localization, Mapping
SLAMMOT	Simultaneous Localization, Mapping, and Moving Object Tracking
SPRAY	Spatial Ray

SVD	Singular Value Decomposition
TN	True Negtive
TP	True Positive
UM	Urban Marked
UMM	Urban Multiple Marked
UU	Urban Unmarked

List of Figures

Introduction

Contents

0.1 Context

In recent years, with the fast development of intelligent vehicle technologies, like Advanced Driving Assistance Systems (ADAS) and autonomous vehicles, traffic safety has become the essential concern of many related researches. According to the global status report on road safety 2013 of World Health Organization, nearly 1.24 million of people are killed on the world's roads and another 20 to 50 millions of people are injured or disabled in traffic accidents every year. Projections indicate that these numbers will rise by about 65% over the next 20 years unless there is an effective prevention. Given the fact that approximately 90% of all traffic accidents are caused (sole cause or contributing factor) by human errors [JT77], the intelligent vehicle technologies are welcomed. A common exception is that they offer the greatest potential for reducing the number or severity of road accidents. In fact, it is estimated that safety technologies could reduce fatalities and injuries by 40% [ROA03]. Generally speaking, the intelligent vehicle technologies can be classified in two categories: ADAS and autonomous vehicles.

Advanced Driving Assistance System (ADAS) helps to avoid accidents by assisting the driver in directing his attention to relevant information, and by providing prior knowledge on the next traffic situation. In addition, ADAS is able to increase traffic efficiency and comfort during the transportation. Many ADAS products have already been introduced in commercial market, such as adaptive cruise control (ACC), Collision avoidance systems, and traffic warning system of dangerous situation. Fig. 1 shows an example of an ADAS system from Mobileye company: when pedestrians appear in front of the vehicle, this system detects them instantly and alerts the driver. Similar warning

functions are also integrated for safety distance detection and blind spot monitor.

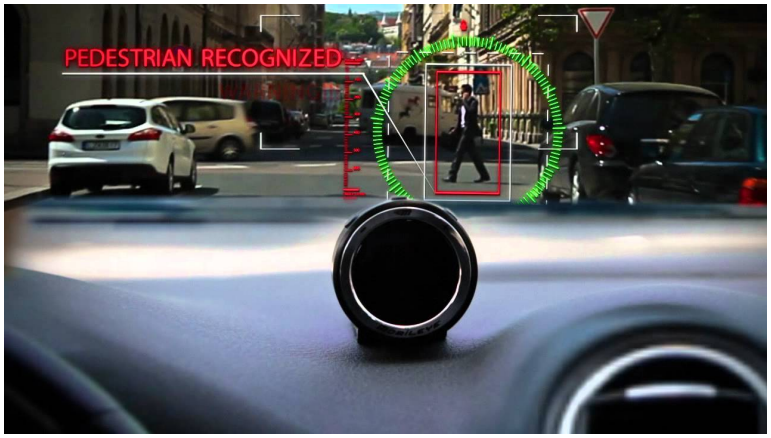
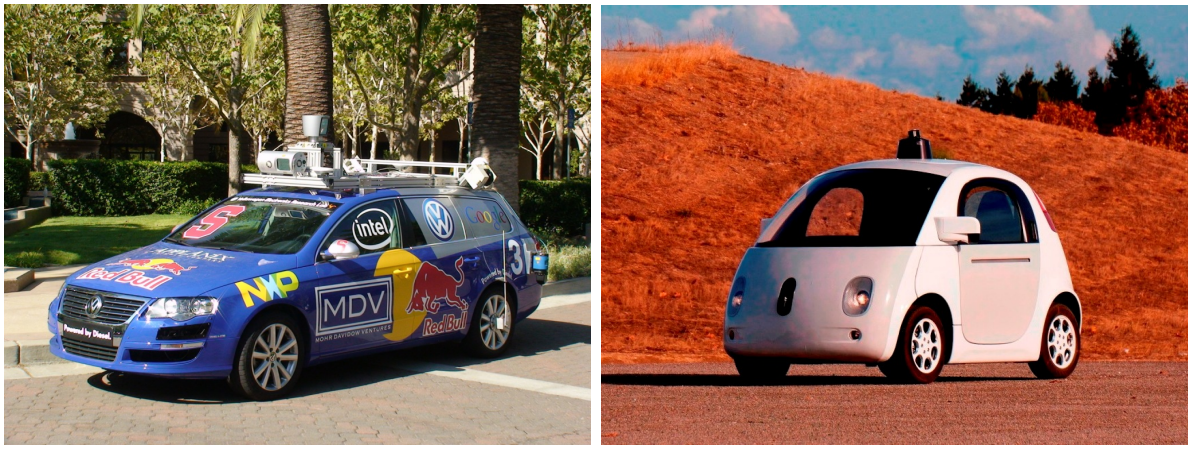


Figure 1 – *Example of ADAS system from Mobileye company*

On the other hand, fully autonomous (also called self-driving, driver-less) vehicles are developed. They are capable of sensing their environment and navigating in it without human intervention. It means that the autonomous vehicles shall be able to make the decisions involving path planning and collision avoidance based on the perception analysis of the environment dynamics. Therefore, a reliable scene understanding system becomes a strict requirement for the autonomous vehicles. From 1980s, the autonomous vehicle research first started in university research centers. Since then, numerous research institutions and major companies have developed working prototypes of autonomous vehicles. During 2003 to 2007, the U.S. Defense Advanced Research Projects Agency (DARPA) held three "Grand Challenges" that remarkably accelerated advancements in autonomous vehicles technology. Major companies also investigate in developing and/or testing driver-less cars, including Audi, BMW, Ford, Volkswagen and Google. In 2014, Google has published their latest assembled autonomous car "Prototype", after a year of improvement, "Prototype" is going to run on public roadways in California.

In driving scenery, traffic safety and efficiency do not only depend on the initiative behavior of host driver or autonomous vehicle, but also rely on the traffic condition and their interaction with the other traffic participants. Therefore, understanding the surrounding environment is the key to ADAS and autonomous driving vehicles. Both of the two technologies employed multiple perception sensors such as GPS, inertial measurement unit (IMU), range sensors (radar [JCL14], Lidar [MCB10, MRD+12], ultrasonic [MAG+02]) and cameras [BBA14, BBA10, MGD12]. For example, in the

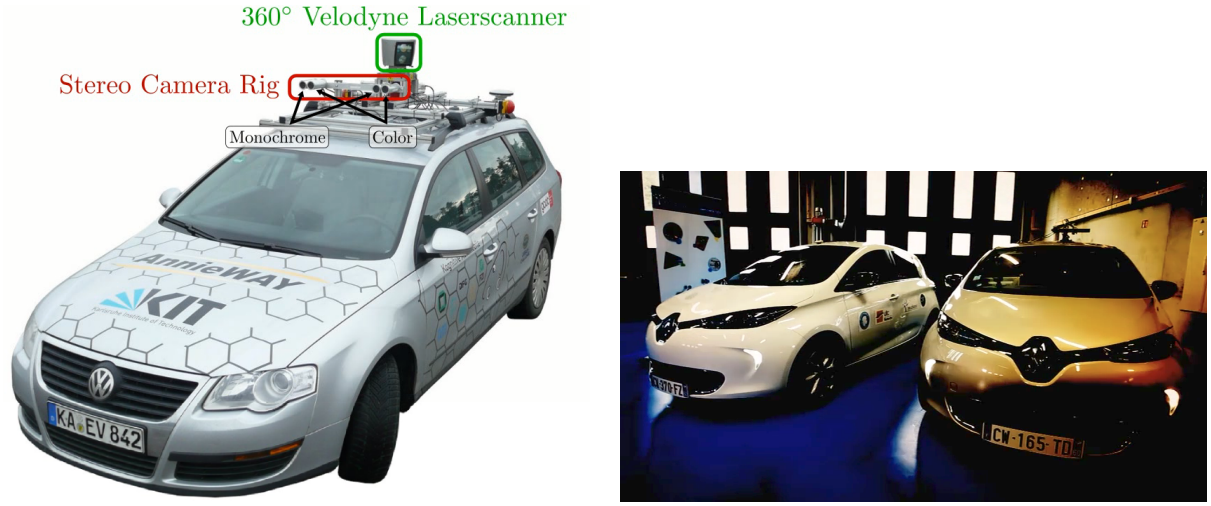


(a) *DARPA Urban Challenge: "Junior" from Stanford*

(b) *"Prototype" from Google*

Figure 2 – *Autonomous vehicles*

intelligent vehicle test platform of KITTI [GLU12], they equipped classical car with two color and two grayscale cameras for object segmentation, a Velodyne laser scanner for accurate 3D information of the environment and a GPS/IMU unit with Real Time Kinematic (RTK) correction signals for localization. The same kind of equipped vehicle exists in the Heudiasyc laboratory. Fig.3 shows these intelligent vehicles.



(a) *Fully equipped KITTI vehicle. Figure from [GLU12]*

(b) *Fully equipped intelligent vehicles of Heudiasyc laboratory.*

Figure 3 – *Examples of intelligent vehicles from Germany and France*

Among the perception sensors, the cameras offer a great amount of information. This advantage allow the vision based systems to perform multiple functions. The color and texture information are employed to segment drivable road area and to detect other

traffic participants as well as obstacles. The geometrical information from multiple cameras can be used to locate host vehicle and to measure its motion. Especially compared to Lidar, cameras show incomparable strength in certain aspects:

- They enable low energy consumption and easy to be employed
- They provide rich information in high dynamics at a low price.

Because of above reasons, vision-based intelligent vehicle techniques have drawn a lot of attentions in last decades. In [MGD12], the authors proposed an active vision-based method for moving objects detection and tracking from a mobile robot. They introduce the motion grid to guide the detection of moving features. An Extended Kalman Filter is then applied to track their motions. Image information can also assist in contextual scene analysis, like road detections that are proposed in [BMVF08], land marking detection proposed by [RGP13]. Besides, cameras are also often combined with other sensors to get a better understanding of the scene. For example, in [PDH+14] a framework for robot localization is proposed using both 2D (camera) and 3D (Lidar) information. Similarly, the authors in [FFBC14] propose a multi-modal system for object detection and localization which integrated stereo camera Lidar and CAN-bus sensors.

0.2 Objective and Challenges

Scene understanding is the foundation of high level functions (such as: path planing and speed control) in intelligent vehicle techniques. However, extracting useful information from camera perception is not trivial and poses a number of theoretical issues. In this thesis, we propose to build a reliable vision-based perception system that can provide a reliable scene analysis to improve the traffic safety. The complete system is represented in Fig. 4. Instead of a solving a single problem, we designed the whole system from a long-term vision. It should be able to cope with the main challenges under variant conditions.

Free road surface and driving space extraction

Using images, only the information within the driving space is useful for navigation and obstacle detection. The first essential question is to define the driving space where all the traffic participants appear. In our approach, we define the driving space as the

road area (which includes the on-road obstacles and free road surface). Without prior-knowledge, it is hard to define the complete driving space from the image sequences. But the free road surface can be directly obtained from images. Usually, the free road surface is also considered as navigable space for intelligent vehicles. In our research, the driving space is then approximated by a convex area of the free road surface in perspective view. However, caused by the illuminant conditions and weather conditions, the free road surface does not always present an homogenous texture [BB94]. Especially, shadows are the most impactive factor since they appear frequently and may lead to false negatives detections. Thus, free road surface detection in varying illumination conditions becomes a hard issue, which should be treated carefully before further processing. To cope with this issue, we introduced illuminant invariant image for free road surface detection. In our approach, shadows are removed by transforming the color images into illuminant invariant images. A confidence interval is then applied as a classifier that works on the intrinsic feature of the pixels in such an image.

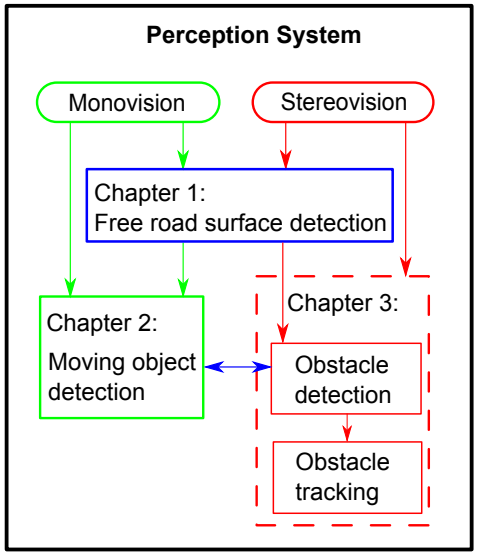


Figure 4 – *Complete perception system for geometrical and contextual scene analysis*

Moving object detection

In driving space, moving objects are traffic participants with a high collision risk. The traffic safety depends on the interaction between the host vehicle with the moving objects. Thus, moving objects must be detected separately in order to analyze and to predict their behavior. Especially, when the sensor resources are limited (i.e. monocular camera), motion-based moving object detection can greatly improve the system efficiency compared to the semantic object detection methods [FMP+13, JDSW12]. There is a variety of methods to detect moving objects, such as background subtraction

[Hei00], spectral clustering [NJW+02] and geometric constraints [KKS09]. However, separating the independent object motions from host vehicle motion (camera motion) remains to be a great challenge. The objects may appear for a short time period; the background can changes rapidly; strong parallax and occlusions constantly exist in the scene... All these situations lead to the difficulties in moving object detection in dynamic environments. The authors in [BBA14] proposed a dynamic objects detection method using stereo vision. In this work, visual odometry is employed to identify the dynamics of objects. In our perception system, we employed multiple geometric constraints to detect moving objects in monovision.

On-road obstacle detection and tracking

A tracking-by-detection system can assist to understand and to predict the object's behavior in traffic scenes. The difficulty of this problem highly depends on how the object is detected and tracked. To avoid traffic collisions, obstacles standing in the driving space should be detected regardless of their shape and motion modality. From this consideration, we choose U-V disparity images in stereo vision as the method to detect the presence of on-road obstacles.

In tracking stage, the scale of obstacle in the image plane is inversely related to its distance to the camera (i.e. projective distortion). This problem makes the tracking in the image space, a tough issue. Especially for some ADAS that need to indicate the location of metric obstacle. Therefore, robust, accurate and high performance approach is required to handle these difficulties. In this thesis, we propose a dynamic particle filter which integrate the depth information to cope with the scale variation caused by projective distortion.

0.3 Contributions and Organization of the Thesis

The perception system we proposed includes three sub-systems: free road surface detection, moving object detection, on-road obstacle detection and tracking. It is intended to work with both monocular camera and stereo rig. Each of the perception sub-systems are connected to assist each other (see Fig.4).

The contributions and the organization of this thesis can be summarized in three main points:

- Chapter 1 presents an illuminant invariant road detection method that can work robustly from color images in the presence of shadows. In mono-vision, a process called axis-calibration is applied to find the parameter that can transform the color image into an illuminant invariant image. After this transformation, the

pixels on the same surface present a homogenous consistency [G.D09]. Shadows as well as their negative influence on the detection process can be removed. Thus, the free road space can be extracted given a confidence interval on the distribution of the illuminant invariant value from the seed pixels. Using stereo vision, the detection result can be refined by a combination with the ground plane extraction from disparity maps. Furthermore, the representation of the detection results in the confidence map is investigated to handle the difficulties in complex road conditions.

- Chapter 2 focus on the monocular moving object detection by geometric constraints. Within the Region of Interest (ROI) generated from free road area, the moving pixels are identified using a plane+parallax approach. First, the potential moving and parallax pixels are detected using a background subtraction method; then three different geometrical constraints: the epipolar constraint, the structural consistency constraint and the trifocal tensor are applied to such potential pixels to filter out parallax ones. The residual distribution modals are introduced carefully to construct moving pixels likelihood equations for each constraint. Finally, a combination of the likelihoods that are measured on different constraints is proposed in a complementary and effective way. Besides, visual odometry is applied to detect the camera motion state, different strategies are then employed accordingly.

- Chapter 3 describes a robust on-road detection and tracking algorithm when stereo vision is available. The obstacles in the driving space can be detected by means of U-V disparity images. Their location in the image plane is then refined by sub-image of disparity map with the help of the geometrical cues. A modified particle filter that combines image and depth information has been proposed for obstacle tracking. If one of the two cameras fails, the system can therefore come back to a monocular operation mode, which is an important feature for the reliability and integrity of perception system.

- Finally, we make concludes the complete perception system proposed in this thesis. The strength and limitations of our work and future perspectives are also discussed.

The different proposed algorithms have been tested on public dataset with an evaluation against state-of-the-art approaches and ground-truth data. The obtained results are promising and show that the proposed methods are effective and robust on the different traffic scenarios and can achieve reliable detections in ambiguous situations.

Chapter 1

Free Road Surface Detection from Illuminant Invariant Image

Contents

1.1 Introduction

Road detection is one of the key issues for intelligent vehicle perception system and traffic scene understanding. It provides the main region of interest that contains crucial information like the navigable space and the obstacles. Autonomous vehicle and advanced driving assistance system both rely on the analysis of the perceived traffic environment to avoid collisions. Generally speaking, the traffic environment is composed of the road texture (including lane markings) and the motion of other traffic participants. With the restriction of a road area, efforts of scene analysis such as obstacle detection, and tracking are targeted to improve the traffic efficiency. It can also help to reduce the processing load. From these considerations, road detection is a necessary component of the intelligent vehicle perception system.

In last decades, many approaches have been developed to obtain a better understanding of the traffic environment. These approaches involve different kinds of sensors such as Lidar, radar or camera. Among the vision based approaches, there are road-like appearance detection using homography estimation [GMM09], road detection in omnidirectional Images with optical flow [YMOI08] and active contours based road detection [MLIM10]. These approaches are evaluated on different datasets with different measurements. Even the results presentations are different: perspective mapping in [WF13], occupancy grid in [PYSL10] and Bird-Eye-View in [CG05]. Fortunately, [FKG13] has introduced an open-access dataset and benchmark which is called KITTI-ROAD for road area detection. They also provide a web interface to evaluate road detection approaches in the 2D Bird's Eye View (BEV) space. Many recent approaches have published their evaluated results on this benchmark.

In this chapter, we propose a vision based fast road detection approach that is able to handle variant illumination conditions. This approach does not rely on any training processing or any other prior knowledge. Hence, it can be used in dynamic driving conditions regardless the road shape. Along with the proposed approach, we construct a free road surface detection system that can function with both monocular camera and stereo rig. Constrained to monovision, it uses the illuminant invariance theory of color images to extract intrinsic feature of road surface, and then classify the pixels using confidence interval. With stereo-vision, a disparity map based road profile extraction is employed to improve the detection results. At last, to improve the flexibility and reliability of the results, confidence models are introduced to build a confidence map of the detection results.

The free road surface detection system is illustrated in Fig. 1.1. From the consideration of the complete perception system, the binary road detection result can be directly used for moving object detection and tracking. The confidence models are noted in dashed lines.

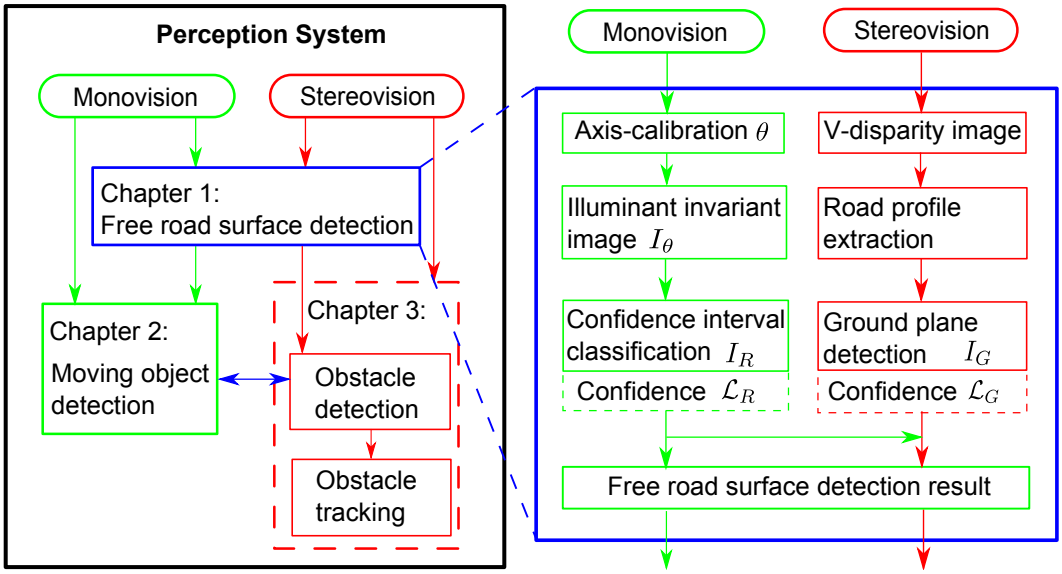


Figure 1.1 – *Sub-system of free road surface detection*

In this chapter, geometric and photometric camera models are first introduced in Section 1.2. Then related works of road detection are discussed in Section 1.3. The theory of illuminant invariant image is presented (Section 1.4) followed by the confidence interval-based road detection algorithm in mono-vision (Section 1.5.1). Complementary algorithms using stereo vision are presented in Section 1.5.2. The confidence models are designed afterwards to improve the previous detection results. Finally the approach is evaluated on the KITTI-ROAD dataset. The results proved that our approach is fast, robust and can be applied for real-time embedded systems.

1.2 Visual sensors models

In the proposed visual perception system, we perform the scene understanding purely on the information from image sequences. Therefore, we need to introduce first both the geometric and photometric camera models.

1.2.1 Geometric camera models

1.2.1.1 Pinhole camera model

In the pinhole camera model (see Fig. 1.2), a 3D point $\boldsymbol{P} = (X, Y, Z, 1)^T$ in the camera coordinate frame is projected on the image at a 2D point $\boldsymbol{p} = (u, v, 1)^T$. The location of point $\boldsymbol{p}$ is given by the projective transformation defined in Eq. 1.1.

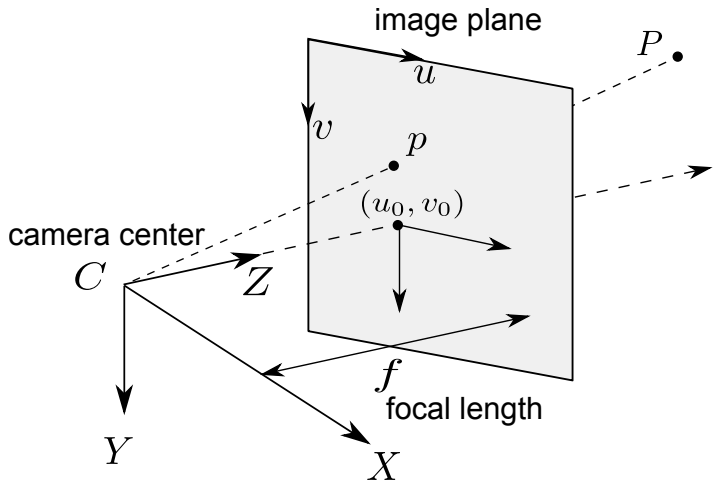


Figure 1.2 – *Pinhole camera model using perspective projection*

$$\lambda \boldsymbol{p} = \begin{bmatrix} f_x & s & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \boldsymbol{P} \tag{1.1}$$

where, u_0, v_0 are the coordinates of the principal point, f_x, f_y are the focal length for each axis, λ is the scale factor linked to the homogenous coordinates, and s is the skew factor. For most cameras $s = 0$.

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.2}$$

where, $\mathbf{K}$ is called the camera intrinsic parameter matrix. We can then set the projection matrix as:

$$\mathbf{P} = \mathbf{K}[\mathbf{I} \mid \mathbf{0}] \tag{1.3}$$

where, $\mathbf{I}$ is a 3×3 identity matrix and $\mathbf{0}$ is a 3×1 zero vector.

If the camera center is not located at the origin of the world coordinate frame, the projective matrix is written as:

$$\mathbf{P} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}] \tag{1.4}$$

where, $[\mathbf{R} \mid \mathbf{t}]$ is the euclidean transformation from world coordinate to camera coordinate frame, $\mathbf{R}$ is the rotation matrix and $\mathbf{t}$ is the translation vector.

In reality, the images taken from camera show lens distortions. Let a point $\boldsymbol{p}_d = (u_d, v_d)^T$ be a distorted point image, and $\boldsymbol{p} = (u, v)^T$ be its corresponding point in ideal coordinates. Then, the real image coordinates are distorted from ideal coordinates

following Eq. 1.5.

$$\boldsymbol{p}_d = \boldsymbol{p} + \delta\boldsymbol{p} \tag{1.5}$$

where,

$$\boldsymbol{p}_0 = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} \tag{1.6}$$

$$\delta\boldsymbol{p} = \begin{pmatrix} \delta u \\ \delta v \end{pmatrix} = \begin{pmatrix} \delta u^{(r)} & \delta u^{(t)} \\ \delta v^{(r)} & \delta v^{(t)} \end{pmatrix} \tag{1.7}$$

$\delta\boldsymbol{p}$ is the approximated lens distortions composed by the radial distortions:

$$\begin{pmatrix} \delta u^{(r)} \\ \delta v^{(r)} \end{pmatrix} = \begin{pmatrix} (u-u_0)(k_1 r_d^2 + k_2 r_d^4 + k_3 r_d^6) \\ (v-v_0)(k_1 r_d^2 + k_2 r_d^4 + k_3 r_d^6) \end{pmatrix} \tag{1.8}$$

and the tangential distortions:

$$\begin{pmatrix} \delta u^{(t)} \\ \delta v^{(t)} \end{pmatrix} = \begin{pmatrix} 2k_4(u-u_0)(v-v_0) + k_5(r_d^2 + 2(u-u_0)^2) \\ k_4(r_d^2 + 2(v-v_0)^2) + 2k_5(u-u_0)(v-v_0) \end{pmatrix} \tag{1.9}$$

where,

$$r^2 = (u-u_0)^2 + (v-v_0)^2$$

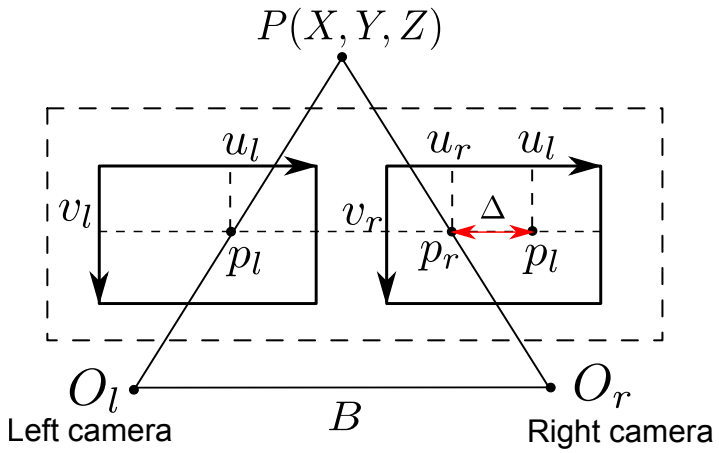
$k_1,\ k_2,\ k_3$ are the radial distortion coefficients; $k_4,\ k_5$ are the tangential distortion coefficients.

In this thesis, we consider that the ideal perspective projection model (Eq. 1.1) is applied in the system. Therefore, the images we used for experiments are undistorted and rectified by camera calibration [Zha00].

1.2.1.2 Stereo vision model

With two images of the same scene captured from slightly different viewpoints, a disparity map $I\Delta$ could be computed [TV98]. It refers to the displacement of the relative features or pixels between a pair of calibrated stereo images.

In the camera coordinate frame, the position of a point in the stereo image planes is given by its coordinates (u_l, v_l) in the left image and (u_r, v_r) in the right image separately (see Fig. 1.3a). For a calibrated stereo rig, the same point in the world coordinate captured by two cameras follows Eq. 1.10 in the image coordinates, and the disparity Δ is defined by the displacement along the u-axis as in Eq. 1.11:



(a)

Figure 1.3 – *Disparity illustration in image coordinates of stereo rig:* $\boldsymbol{P}(X, Y, Z)$ *is a 3D point observed by stereo cameras.* O_l, O_r *are the optical centers of left camera and right camera respectively.* B *is the baseline.* (u_l, v_l) *and* (u_r, v_r) *are the image coordinates for left frame and right frame.* $\boldsymbol{p}_l$ *and* $\boldsymbol{p}_r$ *are the projections of point* $\boldsymbol{P}$ *in the stereo images. They have the same v-axis position. Their displacement along the u-axis is so-called disparity* Δ*.*

$$v_l = v_r = v \tag{1.10}$$

$$\text{disparity}: \ \Delta = u_l - u_r \tag{1.11}$$

In many recent work [GHRDKP14, KB12], the disparity map is commonly used to extract 3D information of the scene, because the disparity value of a pixel in the disparity map is inversely proportional to the depth of the corresponding 3D point $\boldsymbol{P}(X, Y, Z)$:

$$\Delta = f \cdot \frac{B}{Z} \tag{1.12}$$

where, B is baseline distance in meters.

The U-V-disparity images are built by accumulating the pixels of same disparity in $I\Delta$ along the u, v axis separately [Pri03]. For example, the intensity of each pixels in the V-disparity image $I_v\Delta$ is calculated according to Eq. 1.13.

$$I_v\Delta(v_i, \Delta_i) = \sum_{p \in I\Delta} \delta_{v,v_i}\delta_{\Delta,\Delta_i} \tag{1.13}$$

where, $\delta_{i,j}$ denotes the Kronecker delta.

The points that stand at the same distance to the stereo rig have the same disparity value. Thus, U-V-disparity images can help to understand the structure of the scene. However, they present different characteristics which we will introduce in the following

part of the thesis.

1.2.2 Photometric camera model

In color imaging, the primary colors of the visual spectrum are clustered into three channels: red, green and blue (RGB). RGB color at a pixel is resulting from an integral over the visible wavelength. Under the assumptions of Lambertian reflectance, approximately Planckian lighting, the value of each RGB channel is represented by:

$$\mathrm{R}_i = \varrho \int \mathrm{E}(\lambda)\mathrm{S}(\lambda)\mathrm{Q}_i(\lambda)d\lambda, \quad i = R, G, B \tag{1.14}$$

where, ϱ is a Lambertian shading; $\mathrm{E}(\lambda)$ is the spectral power distribution; $\mathrm{S}(\lambda)$ is the surface spectral reflectance function; and $\mathrm{Q}_i(\lambda)$ is the sensitivity of the camera. The RGB color is usually used as appearance information for scene understanding; and in our case, for road detection to get contextual information about the observed scene.

1.3 Related works

The flexibility of vision systems provides a variety of information like colors, shapes, and depth at low cost with reduced power consumption. For this reason, several vision-based road detection approaches have been proposed recently. For example, in [YGyY07], the road detection is performed by boosting image features. The authors of [KAP10] propose to detect the road using a new approach of vanishing points detection combined with texture orientations extraction. Its novelty lies in the introduction of a soft-voting scheme for finding the vanishing point of a single image. Then it extracts two main rays towards the vanishing point in the image as the border of road. This work relies on the assumption of vanishing points and two borders, so it easily fails when the structure of road is complex such as traffic intersection, uphill road, etc. The same problem exists in the approach of [GG12]. In this work, the road extraction is based on road borders extraction using texture classification. Shadows or other illuminant changes on the road can present different textures which are hard to be characterized. Some other approaches such as [KKF12] proposes a spatial ray feature which can distinct a lane of road without requiring an explicit model. However, the result is sensitive to initial settings. If the road is considered as the dominant plane, methods like Inverse Perspective Mapping [CG05] are proposed to remove the perspective effect by transforming the image into another view with homogeneously distributed information. Moreover, IPM allows to obtain a bird's eye view of the scene. It requires the knowledge of the camera pose relative to the ground plane. The

same assumption is also used in homography estimation with road-like appearance detection [GMM09], omnidirectional images with optical flow [YMOI08] or stereo vision [BMVF08]. However, such assumptions do not always stand caused by the shape of road and the vibration of cameras.

Generally, the main drawback of vision systems is their sensibility to illumination conditions such as shadows, back-lighting and low rising sun conditions. Especially, shadows are most impactive since they appear randomly and may lead to spurious detection. Hence, road detection in varying illumination conditions becomes a tough issue, that should be handled with care before further processing.

To solve this problem, [FDC04] presented a method from the view of "Invariant Image". Shannon's entropy is used to find and to distinguish the intrinsic quality of surface's specular properties. This method has been first introduced for road extraction by [AL11]. In this work, the road detection approach uses the Illuminant Invariance Theory on color images to classify road pixels. A model-based classifier is built to extract the drivable road area after that the intrinsic features are obtained from RGB images. It realizes a simple and efficient separation of road and non-road area from RGB images. But some drawbacks still remain. For example, it did not avoid the impact of skylight on the axis-calibration result, and the classification threshold value of the classifier relies on prior manually segmented ground-truth. From this consideration, we proposed relevant modifications to improve the performances of this algorithm. Furthermore, we extended our method with stereo vision for 3D road plane extraction. This work has been published in [WF13]. However, in real traffic scenario, especially with unstructured road, a simple binary classifier is limited since ambiguities often happen in the real driving scenes. To handle this issue, we proposed an algorithm that provides a confidence map of the detection result inspired by [WF13]. There are two main parts in the algorithm: pre-detection from illumination intrinsic image and plane extraction from the V-disparity map segmentation. The idea is to build an on-road confidence degree for each pixel after these two main procedures, and to calculate a confidence map by fusing the two confidence degrees. The objective is to show that the confidence map should be more flexible than a simple binary map in complex environments.

Both the results of the binary map and the confidence map are evaluated on the KITTI-ROAD dataset [FKG13]. The evaluation results support our hypotheses in the way that the confidence map is more adaptive to ambiguous situation, while binary map outperforms in regular road scenes configuration. Comparisons are also made with the other algorithms published on the KITTI-ROAD benchmark website.

1.4 Illuminant Invariant Image

Vision systems are often used in the robotic field to perceive the environment. However, vision-based systems are sensible to illumination conditions such as shadows, back-lighting and low rising sun conditions. According to [FDC04], shadows can be removed by extracting an essential quality of different surfaces. Thus, an image that is invariant to effects of illumination can be obtained.

1.4.1 Shadow removal

Narrow band cameras capture only a very small part of the spectrum within the range of each channel. Thus the RGB value of the image is distinctively represented by different wavelengths. For convenience, we can assume that camera sensitivity is exactly a Dirac delta function:

$$\mathrm{Q}_i(\lambda) = \mathrm{q}_i \delta(\lambda - \lambda_i) \tag{1.15}$$

where, λ_i is the wavelength of each channel that can be captured by the camera. For each color camera, there exit Dirac delta functions to simulate the channels. Only the difference lies in the parameters.

As a result, the Eq. 1.14 can be written as:

$$\mathrm{R}_i = \varrho \mathrm{E}(\lambda_i)\mathrm{S}(\lambda_i)\mathrm{q}_i, \quad i = R, G, B \tag{1.16}$$

Supposing that lighting can be approximated by Planck's law, with Wien's approximation [WS82], we get:

$$\mathrm{E}(\lambda, \mathrm{T}) \simeq \mathrm{I}\ k_1 \lambda^{-5} e^{-\frac{k_2}{\mathrm{T}\lambda}} \tag{1.17}$$

$$\mathrm{R}_i = \varrho \mathrm{I}\ k_1 \lambda_i^{-5} e^{-\frac{k_2}{\mathrm{T}\lambda}} \mathrm{S}(\lambda_i)\mathrm{q}_i, \quad i = R, G, B \tag{1.18}$$

where, k_1, k_2 are constants resulting from the calculation of constant factors in Wien's approximation, the temperature T characterizes the lighting color and I gives the overall light intensity.

By calculating the ratio between the color channels, we can effectively remove the effect of Lambertian shading ϱ and illumination I from Eq. 1.18 which are defined as:

$$\mathrm{c}_i = \mathrm{R}_i / \mathrm{R}_3 \tag{1.19}$$

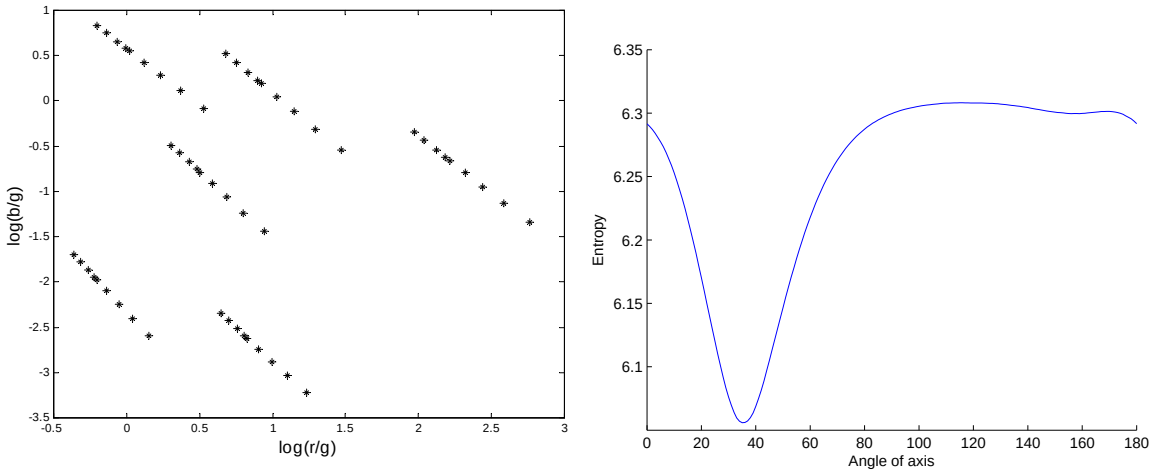
R_3 represents one channel picked from R,G,B, and R_i represent the other two channels. In this way, a 2-vector of chromaticities is constructed to be independent with respect to the illumination intensity. After the division, chromaticity c_i can be considered as a term composed with a constant factor which is apart from spectral power distribution $E(\lambda_i)$ and the surface spectral reflectance function $S(\lambda_i)$. Taking the logarithm of the chromaticities, the value of the log-chromaticities vectors are now linearly correlated:

$$\rho_i \equiv \log(c_i) = \log(s_i/s_3) + (e_i - e_3)/T, \quad i = 1, 2 \tag{1.20}$$

With $s_k \equiv k_1 \lambda_i^{-5} S(\lambda_i) q_i$ and $e_i \equiv -k_2/\lambda_i$. Thus, ρ_1 and ρ_2 form a log-chromaticity space (see Fig. 1.4a). In this space, the pixels on the same surfaces under different illuminations are on a straight line. The lines l_i that represent different chromaticities are almost parallel. Their directions are only determined by the vector $\mathbf{e} \equiv (e_i - e_3)$, which corresponds to the spectral power distribution $E(\lambda_i)$. Their displacements $\log(s_i/s_3)$ are only related to the surface spectral reflectance function $S(\lambda_i)$.

Hence, an intrinsic grayscale image I_θ with suppressed shadows, can be formed by projecting the lines l_i into the direction $\mathbf{e}^\perp$ which is orthogonal to the vector $\mathbf{e} \equiv (e_k - e_p)$. This is their common orthogonal axis which makes an angle θ with the horizontal axis. Therefore, I_θ is lighting independent and is also shadow-free:

$$I_\theta = \boldsymbol{\rho}^T \mathbf{e}^\perp, \boldsymbol{\rho} = (\rho_1, \rho_2); \ \mathbf{e}^\perp = (\cos\theta, \sin\theta) \tag{1.21}$$



(a) **(b)**

Figure 1.4 – *(a) Example of chromaticities for 6 surfaces under 9 lights in log-chromacity space [FDC04] (b) Entropy plot for different axis.*

1.4.2 Axis-calibration

In Section 1.4.1, a substantial representation of the illuminant invariant feature can be represented as a projection angle θ in the log-chromaticity space. In this section we will discuss about the way to find the correct projecting direction.

In [FDC04], Finlayson et al. proposed a method to self-calibrate the camera with entropy minimization from a single image. In this calibration method, Shannon's definition of entropy is used on the image histogram. The choice of the bin width of the histogram that we use in our approach is presented in Appendix A. Finally, the entropy η is calculated as Eq. 1.22.

$$\eta(\theta) = -\sum \mathcal{P}_j(I_\theta) \log \mathcal{P}_j(I_\theta). \tag{1.22}$$

where $\mathcal{P}_j$ is the empirical probability for each bin in the histogram of I_θ. The axis that generates a grayscale image with minimum entropy is the correct angle for distinguishing different surfaces. This process is called "axis-calibration".

$$\theta = \arg \min_{\theta \in [0,\pi]} \eta(\theta)$$

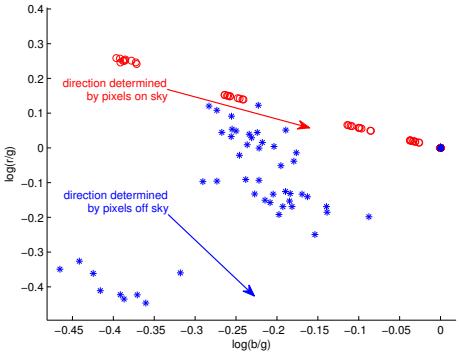
By projecting the log chromaticities of pixels to this angle θ, the image can be transformed into an illuminant invariant image of grayscale. It is possible to recover the shadow-free color image furthermore, but for road detection, the grayscale image is sufficient.

1.5 Road detection

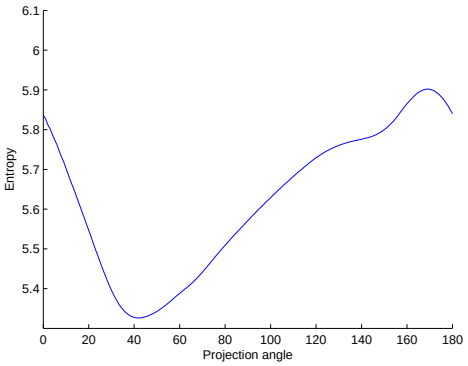
Among the popular vision-based researches, road detection not only provide a straightforward information for drivable area but also helps to obtain a precise obstacle detection and a road profile estimation. In this section, we first present a monocular approach for drivable road detection in variant illumination conditions by extracting its specular intrinsic feature from a color image. A sky removal function is added in order to eliminate the negative effects of sky light on axis-calibration result (See Section 1.4.2). Then, a confidence interval helps the pixel classification to speed up the detection processing and release the approach from dependence on the training processing. If the vehicle is equipped with stereo rig, a disparity based extension helps to obtain a 3D road profile extraction and can improve the detection results when the

assumption of planar road is not established for stereo vision based road profile extraction, we improve the algorithm by constructing a pixel-level confidence map. Such a strategy copes better with ambiguous environments, compared to a simple binary map. Evaluations and comparisons of both, binary map and confidence map, have been done using the KITTI-ROAD benchmark.

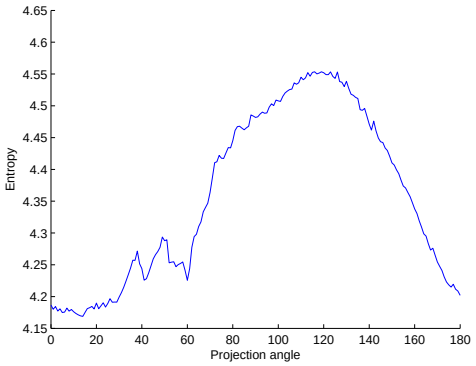
(a) *Samples selected both from and other areas*



(b) *Sample pixels transformed in the log-chromaticity space*



(c) *Entropy plot with sky removal*



(d) *Entropy plot with only pixels from sky area*

Figure 1.5 – *Influence of sky pixels in axis-calibration. In (b) red circles are sky pixels and blue stars are the other pixels; arrows are directions determined by these pixels.*

1.5.1 Monocular vision road detection

1.5.1.1 Sky removal for axis-calibration

Axis-calibration results show great variations, especially when the sky takes more than 30% of the image area. Fig. 1.5 compares the entropy plots calculated with the pixels from non-sky area and only-sky area. Especially, Fig. 1.5b shows that the pixels from sky do not respect the axis-calibration theory and the line formed by them is not

parallel to the others in the log-chromaticity space. In fact, the varied appearance of the sky can be explained by the Rayleigh scattering function:

$$I_{scat}(\lambda) = \text{const} \cdot I_{inc}(\lambda)/\lambda^4 \tag{1.23}$$

where, I_{scat} represents the intensity of the scattered light, I_{inc} is the intensity of the incident light and λ represents the wavelength. Clean air scatters blue light more than red wavelengths, and so the midday sky appears blue. At sunrise and sunset, the distance that light goes through from sun to camera is longer, blue light is almost scattered away so they can not reach to the camera. Meanwhile, the red light is preserved in $I(\lambda)_{scat}$ and makes the sky red. On contrary, for trees and roads, their colors captured by the camera are caused by the reflection of black-body radiation. This difference tells why axis-calibration theory is not suitable for the sky area.

Hence, sky removal becomes necessary for axis-calibration. Usually, if the rotation and translation from camera to ground plane is known, it is possible to determine the horizon line in the images directly. Under this condition, we can cut off the above part for sky removal. Otherwise, if there is a lack of horizon information, an adaptive horizon finding is applied with monocular camera as in [NVFZ11]. The authors propose an algorithm that analyzes the highest 60% area of the image that is divided into 10 parts by empirical horizon line; for each line, an Ostu threshold [S$^+$04] that minimizes the inter-class variance of sky and ground scene is calculated, and the most effective value is expected horizontal line to segment the sky. The Otsu threshold gives satisfactory results when the numbers of pixels in each class is close to each other.

To simplify the procedure and to save time, we propose that once the horizon line has been determined at the beginning of the sequence, it can be directly applied to the next few images on a finite time horizon. We assume that the horizontal line varies slightly within a short time period. Even if sky area is partially mis-classified into road area, in practice, this mis-classified sky area is relatively small and only takes a small proportion of the segmented road scene. Hence, their influence in axis-calibration can be neglected. When extended to the stereo vision, the horizontal line that separate sky area and road area can be easily obtained from V-disparity image (see Section 1.5.2).

For the stereo vision application, we use the analysis of V-disparity map to determine the infinite horizontal line this processing is fast and simple, thus it is applied with each frame.

1.5.1.2 Improved log-chromaticity space

After an evaluation of the previous approach, the axis-calibration results still show an unstable variance as written in Tab. 1.1. This is because in a scene, one of the channels

could take the most importance and the other are rarely appeared. For this reason, we introduced a new log-chromaticity space built using the geometric mean to offer an equal processing for the transformations of each channel [FDC04].

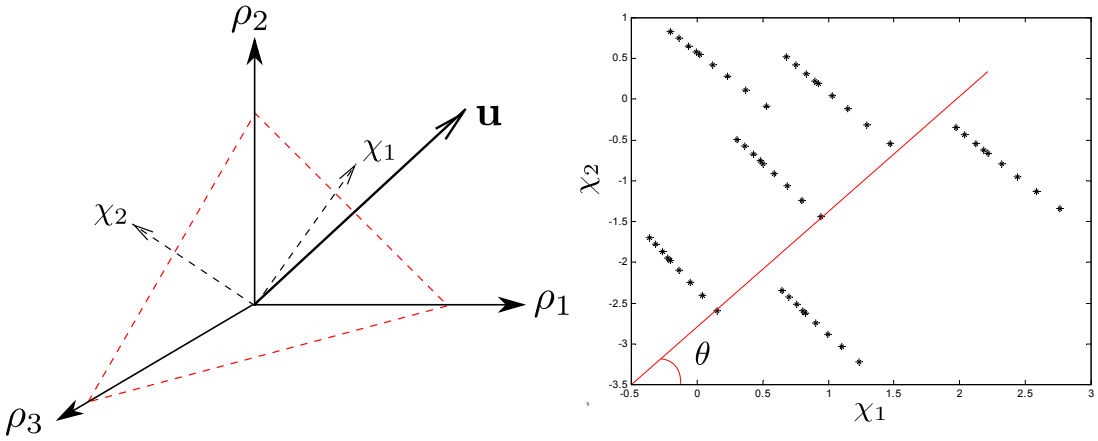


Figure 1.6 – *Log-chromaticity space transformation[FDC04]*

From Eq. 1.19, σ and q_i can be removed through a division by any of the other color channels. However, the choice of the denominator is still a tough issue. If the denominator appears rarely in the whole image (e.g. channel red in Dataset1), then the variance of the division would be quite important [G.D09]. In reality, for long-term driving, the background tonal is changing continuously even dramatically, e.g. an urban scene with colorful buildings along the road. In order to avoid favoring one particular channel, the R, G, B factors could be transformed to division by their geometric mean, i.e

$$C_{ref} = \sqrt[3]{R \cdot G \cdot B} \tag{1.24}$$

Thus, the definition of the chromaticity becomes[1]:

$$c_i = \mathrm{R}_i / C_{ref} \tag{1.25}$$

and the log version remains: $\rho_i = \log(c_i)$. Hence we get a 3-dimension space from the log-chronomaticity of RGB channels. One can notice that in the log space, the color space $\boldsymbol{\rho}$ is orthogonal to vector $\mathbf{u} = 1/\sqrt{3}(1,1,1)^T$ as shown in Fig. 1.6.
Therefore, the transformation from the geometric mean division based 3D space to the 2D log space can be written as (details can be found in [G.D09]):

$$\boldsymbol{\chi} \equiv \mathbf{U}\boldsymbol{\rho} \tag{1.26}$$

[1]the value 0 in R,G,B channels should be carefully preprocessed. In our experiments, all 0 values have been replaced by a small value as 10^{-4} during the algorithm, since the smallest nonzero value of the channels is around 10^{-3} .

where, $\boldsymbol{\chi}$ is a 2×1 vector, $\mathbf{U} = [\mathbf{v}_1, \mathbf{v}_2]^T$, with $\mathbf{v}_1 = (1/\sqrt{2}, -1/\sqrt{2}, 0)^T$ and $\mathbf{v}_2 = (-1/\sqrt{6}, -1/\sqrt{6}, 2/\sqrt{6})^T$ obtained from [G.D09].

Points are still organized on the parallel lines in the new log space [G.D01]. Then the new shadow free gray image is obtained by:

$$I_\theta = \boldsymbol{\chi}_1 \cos\theta + \boldsymbol{\chi}_2 \sin\theta \tag{1.27}$$

The calibrated angle is defined off-line, and the result can be directly used for real-time driving road detection [AL11]. The algorithm can be summarized as follow:

Algorithm 1.1 Axis-calibration algorithm

Input: - Color images in the same sequence I

Output: correct projection axis θ

1: ▶ Determination of the horizon line, cut off above area I_S
2: ▶ Form a 2D log-chromaticity representation $\chi(I_R)$ for the rest of the image I_R.
3: **for** $\theta' = 1°$ **to** $180°$ **do**
4: ▶ Form gray-scale image $I_{\theta'}$ by projecting $\chi(I_R)$ to axis θ': $I_{\theta'} = \chi(I_R) \cdot \overrightarrow{\theta'}$
5: ▶ Forms the histogram of $I_{\theta'}$ with a careful bin-choice and outliers exclusion.
6: ▶ Calculate entropy by Eq. 1.22
7: **end for**
8: ▶ Correct axis θ equals to the anglewhich lead to the minimum Entropy:

$$\theta = \arg\min_{\theta'} \eta(\theta')$$

We compared the result of axis-calibration using normal log-chronomacity space and using geometric mean division based log-chronomacity space in Tab 1.1. In this experiment, 10 frames are randomly selected from two different datasets (See Section.1.6.1). From the comparison, we can see that the axis-calibration in the log-space constructed using geometric mean division outperforms the axis-calibration in original log-space.

Dataset	Dataset1		Dataset2	
Measure (log-chroma)	Robust mean	Standard deviation	Robust mean	Standard deviation
Normal	56.3°	17.19°	49.67°	3.88°
Geometric mean	43.4°	16.09°	34.33°	2.17°

Table 1.1 – *Comparison of normal and geometric mean chromaticities*

1.5.1.3 Confidence interval classification

According to the illuminant invariant property of I_θ, road pixels are expected to be similar regardless of the illumination variety. Therefore, free-road surface and non-road surface could be separated by a model-based classifier as follow [AL11]:

$$\begin{cases} \boldsymbol{p} \text{ is road}, & \text{if } \mathcal{P}(I_\theta(\boldsymbol{p}) \mid \text{road}) \geq \varepsilon \\ \boldsymbol{p} \text{ is background}, & \text{otherwise} \end{cases} \tag{1.28}$$

where $\mathcal{P}(I_\theta(\boldsymbol{p})|\text{road})$ represents the probability $\mathcal{P}$ of pixel $\boldsymbol{p}$ being on the road according to its illuminant-invariant gray scale value $I_\theta(\boldsymbol{p})$. ε is a predefined threshold on this measure. $\mathcal{P}(I_\theta(\boldsymbol{p})|road)$ is obtained from a normalized histogram composed of the selected pixels on the road.

As in [AL11], ε is determined by the measurement of detection effectiveness $F-$measure. The highest effectiveness $F-$measure value illustrates the desired ε. However, this calculation needs manually segmented ground-truth mask as criterion. Thus, the road-extraction results generated by this method fit more appropriately for the drivable road area.

In practical applications, fast road detection should be adaptive to all kinds of environment. For this reason, it is necessary to sever the dependency of the prior knowledge about road's ground truth. Based on this consideration, we introduced confidence interval to determine the threshold ε for the model-based classifier which separates the pixels into road or non-road class.

Notice that, since I_θ has eliminated the influence of shadows, the histogram composed by pixels on road surface is expected to be uni-modal with low dispersion and skewness. Therefore, the normalized histogram follows the empirical form of a normal distribution for a random variable, i.e. $I_\theta(\text{road}) \sim \mathcal{N}(\mu, \sigma^2)$.

Under the assumption that the bottom area of a driving scene image indicates the safe driving distance, as written in [AL11], a road surface model could be built with a subset of pixels dispersed in this area (which is assumed as road surface area). The road's illuminant invariant grayscale value can be subtracted from normalized histogram, i.e., $\mathcal{H}(I_\theta(\boldsymbol{p}))$, where $\boldsymbol{p}$ represents the selected pixels. In our work, 9 patches with a size of 10×10 pixels at the bottom of images have been devoted to modal construction. With this model, it is easy to calculate statistic parameters μ and σ and to stimulate the distribution of $I_\theta(\text{road})$. Empirically, we believe that the middle 95% data in the histogram, represents road's illuminant invariance feature; therefore, we defined confidence level $1 - \alpha = 0.75$ to calculate the confidence interval $[\varepsilon_1 = \mu - 0.6745\frac{\sigma}{\sqrt{n}}, \varepsilon_2 = \mu + 0.6745\frac{\sigma}{\sqrt{n}}]$ of $\mathcal{H}(I_\theta(\boldsymbol{p}))$.

In the whole image, pixels whose grayscale values fall outside this interval would be

regarded as background or obstacles (e.g. vehicles, trees, buildings, etc.). Therefore the classifier could be redefined as:

$$\begin{cases} I_R = 1 \text{ Road}, & \text{if } \varepsilon_1 \leq I(\boldsymbol{p}) \leq \varepsilon_2 \\ I_R = 0 \text{ non Road}, & \text{otherwise} \end{cases} \tag{1.29}$$

Eq. 1.29 is a classifier, and provides binary images of road detection results. Because the thresholds are based on a confidence level, some pixels can be mis-classified. Holes filling and 'majority' morphological operations can cope with false negative errors. As a result, confidence interval calculation helps the pixels classification to speed up the detection processing. For the false positives, we need fusion information to refine them. This is how stereo vision works in the proposed algorithm (see Section 1.5.2).

Notably, when the vehicle stops right behind the front vehicle, the assumption of bottom road may not stand. For videos or continuous image sequences, a tracking process is recommended to detect such a situation. However, for the dataset composed of discrete frame from different sequences like in the KITTI-ROAD benchmark, it is still a tough issue to be discussed. A possible way to solve this problem is to use grouped disparity regions to decide if the bottom area is on a quasi vertical plane or quasi-horizontal plane which helps to instruct the sample selection.

1.5.2 Refinement with stereo vision

When driving in a complex environment, especially in urban areas, artificial constructions are all along the road, and can show a similar intrinsic feature with the road surface. Besides, the road surface itself also presents some color variations (e.g., worn out asphalt and non-uniformly wet road or the lanes), which may lead to dispersion and noise of the road gray scale. There could exist deviation from confidence interval based on monocular road detection results. Thus, the detection performance may have some wobbles in performance. Hence, we introduced plane extraction based on stereo vision to limit the range of road area. Conversely the detected area can help to build a clear V-disparity line for the 3D reconstruction of the road profile. As to the false negative pixels that are excluded using confidence interval, holes filling filters would be useful to fix them. In this section, the former method presented in Section 1.5.1 is employed with stereo vision to decrease the false positive detection and to obtain precise 3D road parameters.

In past few years, research efforts have been made to use stereo vision in intelligent vehicles applications such as pedestrian detection [LSH+12] and road extraction [NSH07]. A well-known approach in the intelligent vehicle community is the V-disparity approach [HU05]. With two images of the same scene captured from slightly different viewpoints,

a disparity map $I\Delta$ could be computed and then it is possible to recover the depth of an object. $I_v\Delta$ is the so called V-disparity image built by accumulating the pixels of the same disparity in $I\Delta$ along the v axis [Pri03]. The grayscale of each pixels in the V-disparity image is calculated according to Eq. 1.13:

For different horizontal offsets, their disparities are different. Thus, a flat plane extended to the far distance is projected as a piece-wise curve in the V-disparity image [HLPA06]. The disparity values on this curve is linearly related to the v-axis.

$$\Delta = av + b \tag{1.30}$$

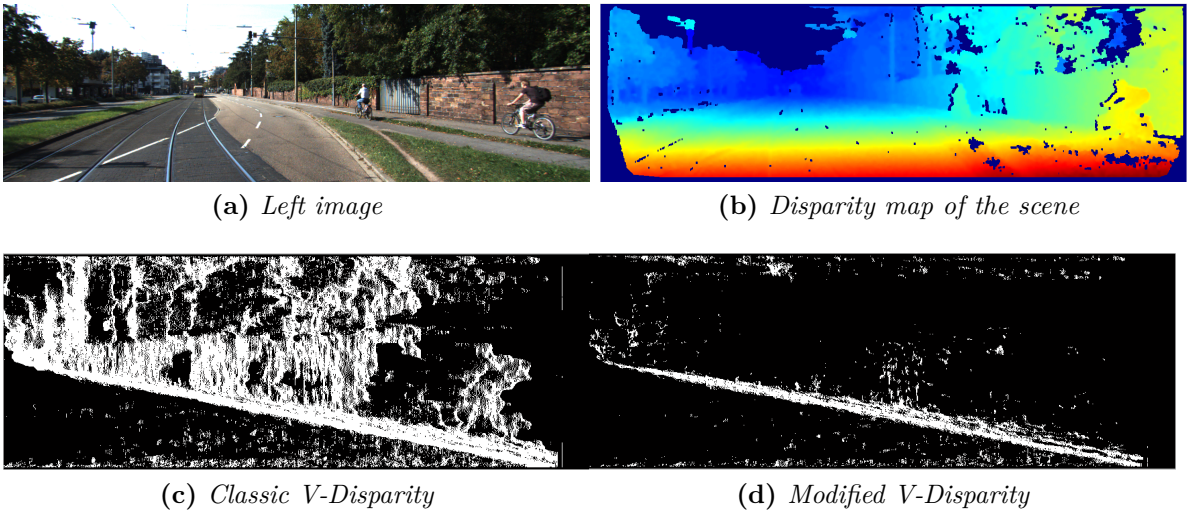
The road is modeled as a plane so that it can be represented by straight a slope line in V-disparity image. To be noted that in many cases the road is not flat, thus the road profile can not be modeled by a straight line. In the KITTI dataset we are using, the roads are mostly flat, so in application we keep the algorithm simple. For complex situation the solution can be found in [LAT02]. The intersection of this line with v-axis is where the road ends, and this value of v-axis can be used to segment sky with the road area. In this way, the V-disparity approach helps to estimate the longitudinal profile of the road.

As mentioned previously, all patches on disparity map will be accumulated to compute the V-disparity image. However, especially in urban scene, it is hard to definite the main line that represents a road area (See Fig. 1.7) with buildings and plants along the road. Because all the vertical objects will be represented as vertical lines stands the road profile. To solve this problem, in our algorithm, only the pixels that are classified as road surface in the binary image I_R will be accumulated to the V-disparity map. The limitation of region of interest (ROI) will greatly reduce the run-time consumption. In this work, the ROI is where $I_R = 1$. Reminding that I_R gives an pre-detection of drivable road area, most of the obstacles (e.g. vehicles and pedestrians) will not be calculated for the V-disparity map. Hence, a regular sloping line as a representation of the drivable area can be achieved as shown in Fig. 1.7d.

According to Eq. 1.30 extracted by Hough Transform [DH72], the image of the ground plane I_G could be constructed by:

$$\begin{cases} I_G = 1 & \text{if } \Delta_{\boldsymbol{p}} \in [\Delta_v \pm \varepsilon_v] \\ I_G = 0 & \text{otherwise} \end{cases} \tag{1.31}$$

Here, we define a dynamic variance $\varepsilon_v = c \cdot v$. The variable c is a positive parameter that indicates that the closer the layout is, the greater the variance becomes. It is determined by preserving the pixels in a 0.75 leveled confidence interval. The variance



(a) *Left image* **(b)** *Disparity map of the scene*

(c) *Classic V-Disparity* **(d)** *Modified V-Disparity*

Figure 1.7 – *Comparison of V-disparity images from the entire disparity map and the ground area of disparity map*

of the disparity on each line is calculated during the accumulation and is used to obtain the proper factor c. Once this parameter is fixed, it can be directly used to most of the driving scenes in the same sequence. Finally, the intersection of I_G and I_R represents a verified road detection result, i.e. $I_{final} = I_R \cap I_G$.

The algorithm can be summarize as follows:

Algorithm 1.2 Stereo vision approach for road extraction

Input: - Stereo color images I_l, I_r
- Primary detection result I_R from illuminant invariant theory

Output: final detection result I_{final}

1: ▶ Compute I_Δ
2: ▶ Compute $I_v\Delta$ with only valid pixels on I_R:

$$I_v\Delta(v_i, \Delta_i) = \sum_{p \in I\Delta} \delta_{v,v_i}\delta_{\Delta,\Delta_i} \mid I_R(p) = 1$$

3: ▶ Extract line's function in $I_v\Delta$ with Hough transform;
4: ▶ Calculate disparity Δ_v for each horizontal offset v
5: ▶ Reconstruct the ground plane area I_G
6: ▶ Verify the final detection result:$I_{final} = I_R \cap I_G$

However, the V-disparity images are not always as ideal as expected. Sidewalks, or the deformation of the road edge (usually depression) represent a bunch of lines slightly different from the road profile in V-disparity space. Fortunately, they only take a small portion of the ROI. The three upper images in Fig. 1.8 shows a comparison of the V-disparity images before and after preserving pixels with high intensity values. As a result, the line indicating the road profile becomes finer and more precise .

Figure 1.8 – *Examples of the different V-disparity images. First row and second row show the V-disparity images from a planar road and a non-planar road respectively. Right to the original images, the first column of the V-disparity images are obtained from the entire disparity map after sky removal; the second column of the V-disparity images are obtained from the disparities on road area*

Fig. 1.9 shows the final result of the algorithm combining the illumination invariant image, confidence interval and the V-disparity image. This result is represented as a binary map, which means that there is no need for further training or threshold determination on this detection result.

However, in real traffic scenario, especially with unstructured road, a simple binary classifier presents some limitations. Ambiguities often happen in the real driving scene. To handle this issue, we proposed an algorithm which provides a confidence map of the detection result [WF13]. There are two main parts in the algorithm: pre-detection from illumination intrinsic image and plane extraction from the V-disparity image segmentation. The idea is to distribute a confidence degree for each pixel after these two main procedures, and to calculate a confidence map by fusing the two confidence degrees. The objective is to show that the confidence map should be more flexible than a simple binary map specially in complex environments.

1.5.3 Road detection using confidence map

The binary map detection result requires a strict precision of each parameter in the algorithm. However, in some other cases, even the same road might be composed of different materials with different surface textures. Thus, we not only need to be able to separate the surfaces different from road but also need to be tolerant to different

Figure 1.9 – *Binary map detection results on the KITTI-ROAD dataset. Each line shows two images from different categories (see Section 1.6.1) separately.*

textures on the same road. This would be a cruel request for binary map detection. To solve this problem a confidence map is built to provide a more flexible and still reliable road detection result. A confidence map might be much more practical in unstructured roads and high variability in scene layout and in strong changes of illumination conditions.

1.5.3.1 Confidence distribution from pre-detection

As in Section 1.5.1, a pre-detection binary image I_R is obtained from the intrinsic image I_θ. As mentioned above, there exist surfaces with similar intrinsic appearance to the road surface, and also the road surface itself might show different textures caused by materials or extreme illumination conditions due to over-saturation. The materials problem will lead to false positive detection, and the illumination difference will cause false negative detection. To deal with these two conflicting situations in a robust algorithm, a confidence degree is assigned to each pixel of I_R with a 3-by-3 matrix composed of 1. For every pixel, the confidence degree is distributed by the sum of its on-road neighbors in the 3-by-3 operator, and then normalized by the total number of neighbors:

$$\mathcal{L}_R(v,u) = \sum_{i=v-1}^{v+1} \sum_{j=u-1}^{u+1} I_R(i,j)/9 \tag{1.32}$$

Since I_R is a binary image, only the valid pixels (where, $\mathcal{L}_R = 1$) after pre-detection will be accumulated to the confidence distribution. To be noted that the I_R using for confidence map construction is pure pre-detection result without morphological operation. The more valid pixels around, the more likely they are on the road surface.

Thus, the algorithm takes the spatial constraint into consideration. As to the false positive pre-detected pixels, they are commonly spread sparsely. After filtered by the operator, they are distributed only with a small confidence degree. On the other hand, for the false negative pre-detected pixels, they can gain some confidence degrees thanks to their correctly pre-detected neighbors. Thus, Eq. 1.32 successfully transforms a binary result into confidence map. An example is showed in Fig. 1.10 the red region has a higher confidence than the green region. The right side of this figure presents how the confidence degree of a pixel is distributed according to his neighbors.

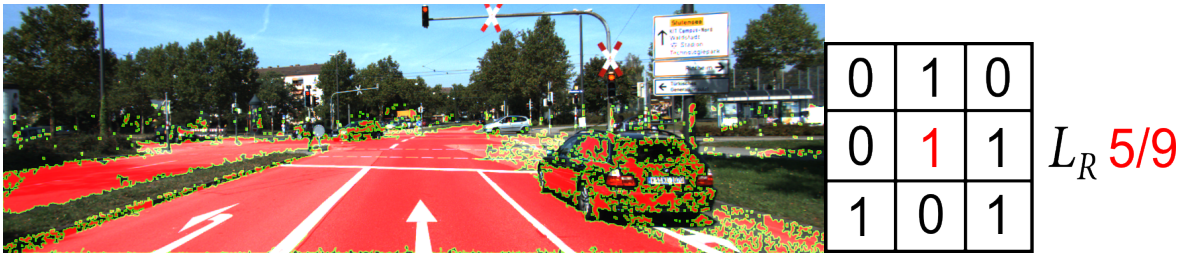


Figure 1.10 – *Example of confidence distribution of pre-detection result, the confidence degree reduces from red color to green color*

1.5.3.2 Confidence distribution from plane extraction

In our approach, we do not use prior knowledge about the camera pose. Hence, the scenery captured by the vehicle camera might be non-horizontal caused by the yaw angle of vehicle while running. For example, the left side is lower/higher than the right side, as shown in the first picture of Fig. 1.8. In this case, the disparity plane values along the u-axis are not centered around a specific value Δ_v, but differ in a broader range, as shown in the disparity map in Fig. 1.11. The disparity values of plane pixels stands on the same row v of the image are more likely following a uniform distribution. In this case, a dominant disparity value does not exist for this layer.

When there is an obstacle along the road standing on the lower side of ground plane, the disparity value of the obstacle might give ambiguity in the V-disparity accumulation. As shown in Fig. 1.8, the upper line highlights an image taken by non-horizontal cameras. Right next to the image, the V-disparity image extracted from pre-detection result is illustrated. A bunch of lines spread almost uniformly in a broad range. After the refinement step for preserving dominant disparity values along the v-axis, two lines appear causing an ambiguous situation.

Hence, a simple binary classifier is not enough to handle this problem. Firstly, the range of disparity values on the plane $[\Delta_v \pm \varepsilon_v]$ is widen, so it is possible to have mis-detected obstacles as road surface, as long as their disparity value falls into this

wide range. The disparity value corresponding to the v-axis Δ_v might even deviate by a wrong Hough line extraction. However, building a complex binary classifier is time consuming, and probably need to verify multiple cues like road's topological and morphological characters. For example, the continuity of the pixels disparity value on the same v-axis might need to be considered.

Another way is to build a confidence distribution for plane extraction, which measures the deviation of the disparity values to its expectation Δ_v. Since the dominant value extraction is not reliable, another criterion need to be proposed. The median factor is a proper candidate for the new criterion Δ_v. It is because road is a sloping plane, then disparity on it on the same v-axis should follow a uniform disparity. Since the v-disparity map is built by accumulating mono-vision based detection result which is mostly composed of road pixels (see Section 1.5.1), the median value will fall on the ground plane even if there exist bias and noise. To reduce the influence of false positive pre-detection, and to speed up the algorithm, only the biggest connected component in I_R is preserved as a new Region-of-Interest I_{new_ROI}.

In the binary detection algorithm, a range of disparity values along v-axis is calculated as $[\Delta_v \pm \varepsilon_v]$, but here: $\Delta_v = \text{median}(\Delta(\boldsymbol{p}_v))$, where $\Delta(\boldsymbol{p}_v)$ is the disparity value of the pixels on the v-th row of the image and $\boldsymbol{p}_v \in I_{new\_ROI}$. The advantage of choosing median value is because in the range of pre-detection result, the median value of disparity of each row surely stands on the road surface and this definition can adapt the algorithm with non-flat road. For pixels are detected as road surface in I_R, if their disparity value fall out of this range, they will be distributed with a confidence degree depending on their disparity difference to Δ_v. Big deviation from median disparity value leads to lower confidence degree.

$$\mathcal{L}'_G(v,u) = (1 - \mid \Delta_{I_R}(v,u) - \Delta_v \mid / \Delta_v) \tag{1.33}$$

With Eq. 1.33, every positive pixel in I_R has a value up to 1; then we add a unit step function to eliminate those negative confidence degree. Thus, we get a confidence map of ground plane $\mathcal{L}_G$ within the range of [0,1].

$$\mathcal{L}_G = \frac{1}{2}\mathcal{L}'_G \cdot (1 + \text{sgn}(\mathcal{L}'_G)) \tag{1.34}$$

1.5.3.3 Confidence map generation

With the two confidence maps $\mathcal{L}_R$ and $\mathcal{L}_G$, a principal confidence map of road detection could be generated based on the following fusion function:

$$\mathcal{L}(v,u) = \mathcal{L}_R(v,u) \cdot \mathcal{L}_G(v,u)$$

Every potential road surface pixel has been distributed with a confidence value. After an evaluation on the training set of KITTI-ROAD benchmark, the best threshold for the confidence value will be found. Fig. 1.11 shows a comparison between the detection performance of binary map and confidence map on the non-horizontal image (see Fig. 1.8). In Fig. 1.11, the binary map detection result is deviated due to the ambiguity of the road profile line in v-disparity map. On contrary, every pixel that is likely to be on the road is presented with a confidence value in the map. Then a proper threshold of classification can be decided by Precision-Recall curve (PR-curve) analysis on training dataset. Therefore, the confidence map is more reliable in complex situations. However, it needs training processing to get a binary detection result. One can notice that the confidence map itself can be used as basement for further applications.

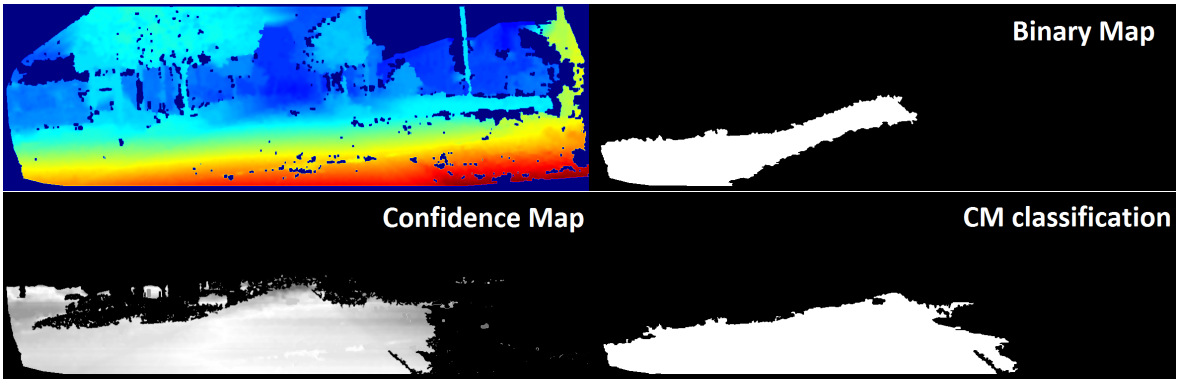


Figure 1.11 – *Detection results of non-horizontal image. Up left: Disparity map of the non-horizontal image presented in Fig.1.8. Up right: Binary map generate by original algorithm (Section 1.5.2), which, directly represents the road detection result. Bottom left: Confidence map generated by improved algorithm (Section 1.5.3.2). Bottom right: Road detection result by applying a proper threshold on confidence map (CM).*

1.6 Experimental results

1.6.1 Dataset and Processing Platform

The following datasets have been used for test and evaluation:

- Dataset1: Road images database from Computer Vision Center (CVC) of University Autònoma of Barcelona[AL11].

- Dataset2: KITTI_ROAD benchmark (Karlsruhe Institute of Technology and Toyota Technological Institute) Vision Benchmark Suite[FKG13].

The dataset from CVC laboratory consists of two continuous monocular sequences acquired on the same scenario at different daytime and under different weather conditions. The day sequence (noon with sunny-shadows) includes 854 frames, and the after-rain (morning) sequence has 841 frames. The resolution of the images is 640×480 pixels. They used a Bumblebee camera that works with fixed calibration parameters. In this dataset, the ground truth of road segmentation is given in perspective view.

The KITTI-ROAD dataset provides 289 annotated training images and 290 annotated test images with a broad spectrum of urban road scenes at a minimum spatial distance of 20m. These images are clustered into three categories:

UU - urban unmarked road;

UM - urban marked two direction road;

UMM - urban multiple marked lanes' road.

The resolution of the images is 375×1242 pixels. There are manually generated ground truth in the perspective view for the training set. We can also evaluate the results from testing set with the KITTI evaluation server. However, the evaluation method proposed by KITTI_ROAD is performed in Bird-Eye-View (BEV). The transformation parameters and function from perspective view to BEV are available on their website[2].

Many researchers have published their evaluation results on this website for comparison, including our stereo vision based detection result.

In our experiment, the processing platform is a standard PC with Windows 7 Enterprise OS, with CPU of 2.66 GHz. The computation environment is MATLAB R2012a.

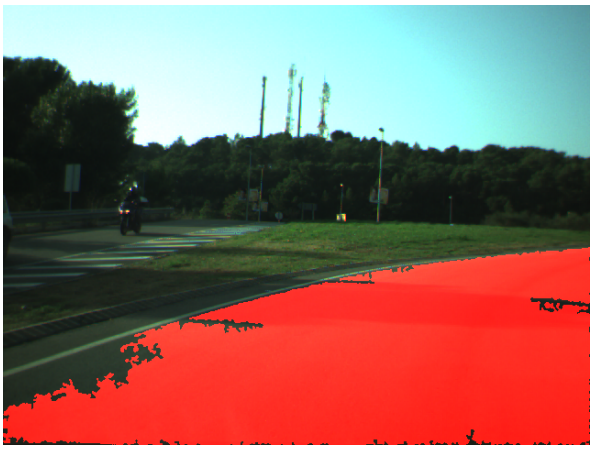
1.6.2 Experimental validation of sky removal

Fig. 1.12a presents an original RGB image from Dataset1, on its right is displayed the axis-calibration result with sky removal. Bottom images in Fig. 1.12 show the comparison between the whole image axis-calibration and sky-removed axis-calibration. In the result of the whole image axis-calibration, the sky pixels become uniform, but shadows remain on the road. After the modification of sky removal, the projection axis of 36° is much closer to the correct axis and shadows are greatly attenuated. As a conclusion, the sky color does not follow the rules in Section 1.4, and sky removal can obviously help to correct the final result.

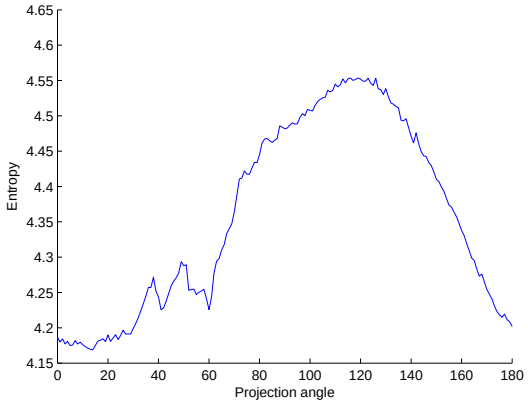
Undeniable, during the test of Dataset1, some results are not satisfying caused by some extremely illuminant conditions like over-saturation in some part of the images. Especially when the driving direction is back-lighted, it is hard to capture real colors

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

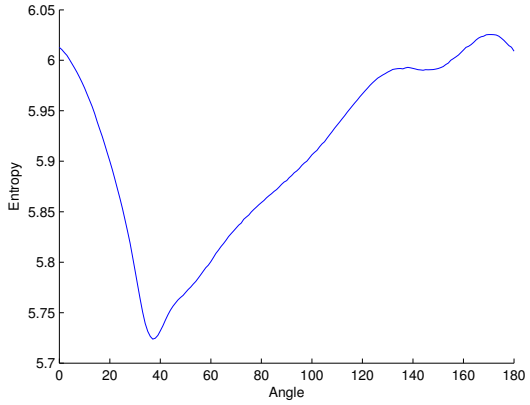
(a) *Original color image*



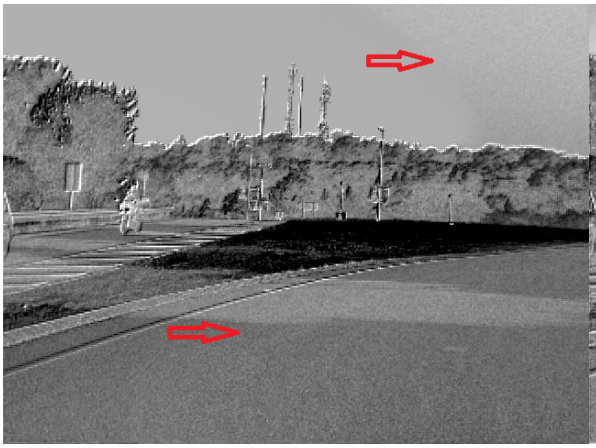
(b) *Road detection result with sky-removal*



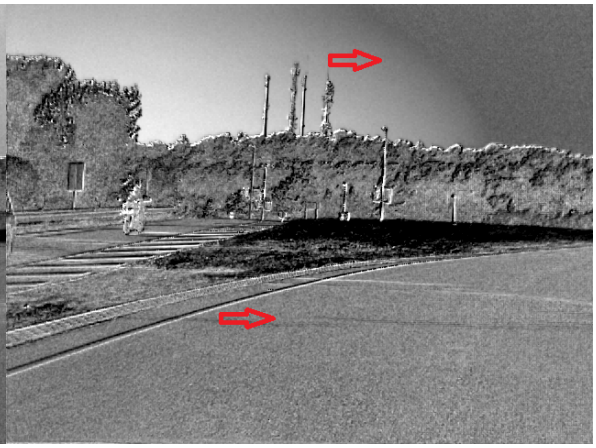
(c) *Axis-calibration with sky region*



(d) *Entropy plot without sky factor*



(e) *Gray image by whole image axis-calibration*



(f) *grayscale image get by sky removal axis-calibration*

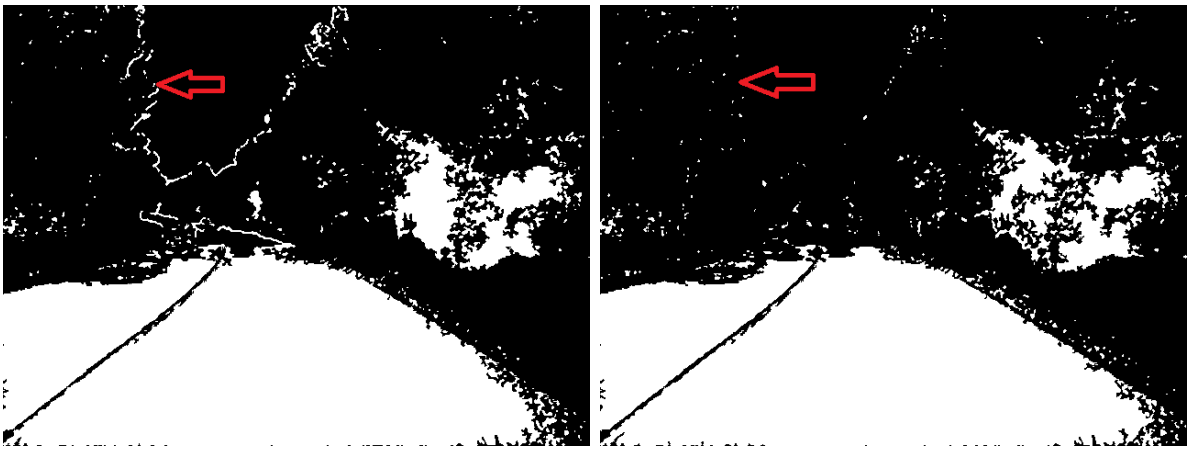
Figure 1.12 – *Comparison the gray images got from original algorithm and sky removed algorithm on Dataset1*

of the scene. Additionally, some parts with low illumination may lead to a mixture of self-shadow and cast-shadow which is more complex to separate from their optical features.

1.6.3 Geometric axis calibration

Tab. 1.1 compares normal axis-calibration result with only sky removal and the result with modifications of both sky removal and geometric mean division for chromaticities. From this comparison, we can see that the result for geometric mean in 2D space is more stable with a smaller deviation. Fig. 1.13 gives an qualitative proof of the improvement. We can see that the result for geometric mean in 2D space (Fig. 1.13b) is finer. So the geometric mean division helps to reduce the axis-calibration variance and to improve the detection precision.

Additionally, we see that the variance of Dataset1 is much greater than Dataset2, that's because the images from Dataset1 contain some special weather and illumination conditions such as cloudy day and over-saturation of the scene.



(a) **(b)**

Figure 1.13 – *(a) Primary detection result with simply sky removal (b) Primary detection result with both geometric mean transform and sky removal*

1.6.4 Monocular road detection using confidence intervals

For free road surface detection evaluation, we use the Dataset 2 which consists of sequences of stereo images of driving scene. KITTI benchmark also provides the evaluation method for a public comparison.

Road detection evaluation measurements

The detection results are evaluated by $F-$measure, average precision, accuracy, and other standard measures like: precision, recall and false positive/negative rate [FKG13]. These measurements is defined as:

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$F-\text{measure} = (1+\beta^2)\frac{\text{Precision.Recall}}{\text{Precision}+\text{Recall}}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$$

where, TP, FP, TN, FN stands for true positive, false positive, true negative and false negative respectively. The parameter β in the $F-$measure is set to 1. For methods that output confidence maps (in contrast to binary road classification), the classification threshold ε is chosen to maximize the $F-$measure, yielding $F_{\max}$:

$$F_{\max} = \arg\min_{\varepsilon} F-\text{measure}$$

Furthermore, in order to provide insights into the performance over the full recall range, the average precision (AP), as defined in [EVGW$^+$10], is computed for different recall values r:

$$\text{AP} = \frac{1}{11}\sum_{r\in 0,0.1,\ldots 1}\max_{\tilde{r}:\tilde{r}>r}\text{Precision}(\tilde{r})$$

Monovision based road detection

In the KITTI-ROAD dataset, the frames are selected from different sequences. It means that there is no clue and indications that can be obtained from previous frames. Hence a reliable detection result for single image is necessary. However, it leads to a problem for usual axis-calibration. The intrinsic angles are different from frame to frame according to possible changes in the camera parameters. In order to provide a reliable result on KITTI-ROAD benchmark, the ground truth of the whole training dataset is used to calculate the axis angle θ, which is approximately equal to 33°.

The monovision road detection result (MonoBM) is evaluated on the testing dataset in BEV. Its performance is listed in Tab 1.2 with comparison to our extended stereo vision method (BM) and other methods like Spatial ray classification (SPRAY [KKF12]); Convolutional neutral network (CNN [AGLL12]) and BaseLine (BL [FKG13]) approaches. In order to provide a lower bound for the performance any road detection algorithm should achieve, authors of [FKG13] extract baselines by averaging all ground truth road maps from the training set. This results in confidence maps indicating for each perspective/BEV location the confidence for being road area or ego-lane. All these methods have been introduced in [FKG13]. According the comparison, our monocular based detection algorithm performs better than CNN method and BL method. Especially it outperforms the other methods on the measurement of recall score.

An example of monocular detection result is presented in Fig. 1.14. As we can see, there exist false alarms in the area of plants or human-build structures with similar texture of the road. On contrary, the false negative is rare, and this leads to a high recall score as showed in Tab. 1.2. In fact, the false alarms can be reduced by taking a narrow confidence interval on the measure of invariant feature of road. However, without the help of prior knowledge it is difficult to find the most suitable interval. Besides, this interval changes in different environment. Hence we propose to introduce a stereo vision method to reduce the false alarms on the objects above the ground plane.

Figure 1.14 – *Monocular road detection result*

As we can see from Tab. , the false positive rate of monovision based road detection is the highest among the methods listed. This is because that we chose a high confidence level for road detection from illuminant invariant image. It means the algorithm shall

detect more potential road pixels. It also leads to a low rate of false negative rate. Since the free road detection is further served for the complete road structure construction, we prefer a lower false negative rate rather than a lower positive rate.

Table 1.2 – *Results [%] of pixel-based road area evaluation on testing dataset.*

URBAN - BEV space						
	F_{max}	AP	Prec.	Rec.	FPR	FNR
SPRAY	86.33	90.88	86.75	85.91	7.55	14.09
BM	82.32	68.95	76.15	**89.56**	16.15	**10.44**
MonoBM	79.45	66.16	69.22	**93.23**	22.83	**6.77**
CNN	78.92	79.14	76.25	81.79	14.67	18.21
BL	75.61	79.72	68.93	83.73	21.73	16.27

1.6.5 Stereo-Vision Based Road Extraction

The results of both binary map and confidence map using stereo vision are tested on the KITTI-ROAD benchmark [FKG13].

The binary map detection (BM) is evaluated on testing dataset in Bird Eye View (BEV) space (see Fig. 1.15). Its performance is listed in Tab. 1.2.

The confidence map (CM) is evaluated on training dataset as a reference to binary map and baseline algorithm.

The run-time for the binary map algorithm (BM) is about 2s per frame. A complete confidence map generation algorithm (CM) takes about 4s per frame. To speed up the algorithm, we add a processing of maximum connected area preservation in the calculation of ROI for plane extraction.

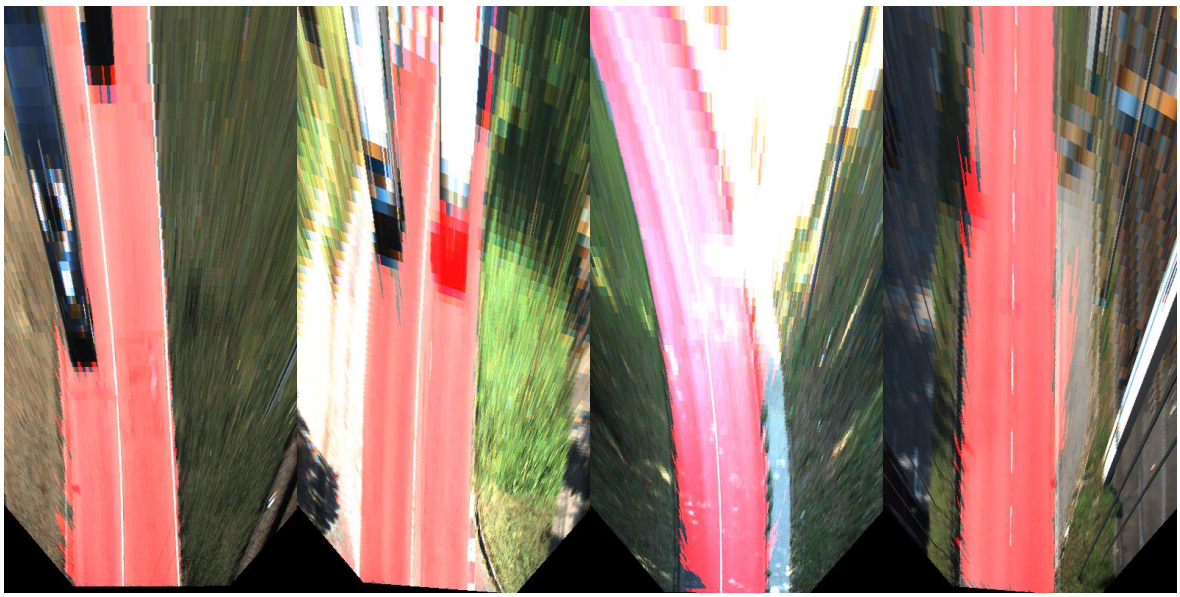


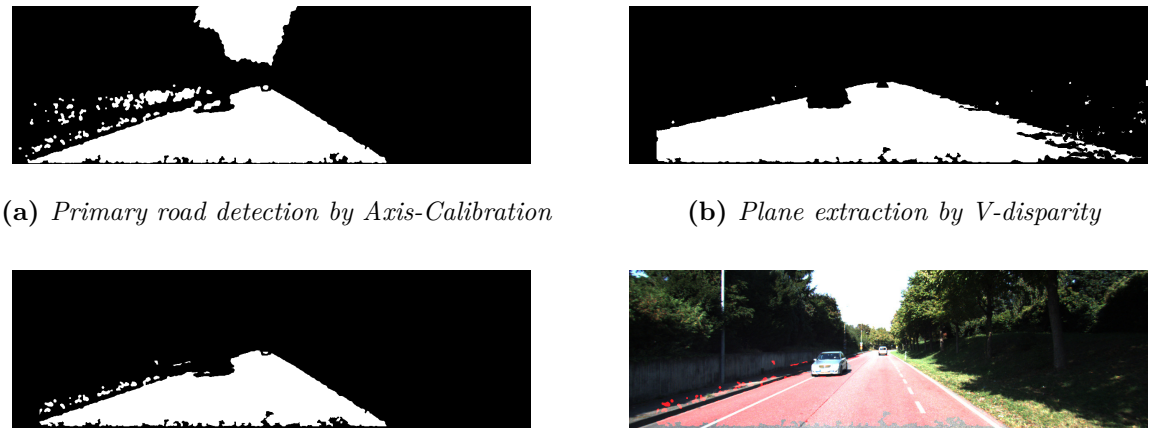
Figure 1.15 – *Detection results transformed in Bird Eye View (BEV) space.*

1.6.5.1 Binary map evaluation

Fig. 1.16 presents the detection results that combine the illuminant invariant image and stereo vision: the first three images present the results of primary road detection I_R by Algo. 1.1, plane extraction I_G and final detection I_{final} separately. The bottom image presents a comparison of the final result (red region) with the original RGB image.

In the comparison presented in Tab 1.2, stereo vision based binary map detection (BM) performs the best in the measurement of recall and false negative rate. As a proof, with the extension of stereo vision, the road detection performance is indeed improved compared to the MonoBM. However, in some special situations, false positive detections are still unavoidable by stereo vision based BM algorithm. For example, when sidewalk shows similar intrinsic features with the road surface, it has a strong probability of being detected as road area.

In general, binary map detection method provides a relative high value on $F-$measure among the compared algorithms in BEV space. The strength of binary map is its independence from prior knowledge of ground truth. This makes it a portable algorithm that can be used for dynamic environments analysis.



(a) *Primary road detection by Axis-Calibration*

(b) *Plane extraction by V-disparity*

(c) *Final road detection result by fusion processing*

(d) *Final Road detection result in original image*

Figure 1.16 – *Road detection with stereo vision*

1.6.5.2 Confidence map evaluation

Considering that the KITTI-ROAD dataset is composed of discrete frames from different sequences, (in different weather condition, with different camera parameters),

a higher confidence level is assigned during the pre-detection step for confidence distribution. Thus, more potential road surface pixels are taken into consideration for confidence map generation.

The evaluation result in perspective view of confidence map detection is listed in Tab. 1.3 with a comparison of binary map results. It is interesting to see that the average precision of the binary map generation algorithm is quite low, which can be hardly compared to the other algorithms. Actually, this measurement, average precision (AP), is a description of Precision-Recall curve with different thresholds to classify the confidence maps. While the binary map directly provides a definite Precision and Recall value, this measurement might not be suitable for it. Under this consideration, we develop the confidence map based on original algorithm so as to evaluate our approach on this measure. According to Tab. 1.3, the confidence map greatly improved the average accuracy of the detection compare to binary map. Therefore, the performance of confidence map still has the potential to provide reliable results in complex environments, as illustrated in Fig. 1.11. Besides, in the subset of UMM, confidence map even outperforms the baseline. If the confidence map algorithm is improved furthermore, a good performance is promising, and the evaluation in BEV space can be proceeded later. However, we did not proceed further investigation because the confidence map detection relies on the analysis of PR-curve on training set.

Table 1.3 – *Results [%] of pixel-based road area evaluation on training set.*

UM perspective space							
	F_{max}	AP	Acc	Prec.	Rec.	FPR	FNR
BL	89.27	92.18	96.53	88.93	90.17	2.26	9.83
BM	85.67	72.21	94.89	77.83	**95.26**	5.18	**4.74**
CM	81.69	80.46	94.09	81.06	82.33	6.67	17.67
UMM perspective space							
	F_{max}	AP	Acc	Prec.	Rec.	FPR	FNR
BM	**88.76**	81.29	**94.55**	**87.04**	**90.55**	**4.20**	**9.45**
CM	**85.28**	82.08	**92.99**	**85.09**	85.46	**4.67**	14.54
BL	82.81	89.21	91.23	77.54	88.86	8.02	11.14
UU Perspective space							
	F_{max}	AP	Acc	Prec.	Rec.	FPR	FNR
BL	80.79	86.13	94.70	79.00	82.67	3.42	17.33
BM	80.50	62.53	94.19	73.44	**89.07**	5.02	**10.93**
CM	75.88	71.48	93.18	72.53	79.55	4.69	20.45

1.6.6 Conclusion and further discussions

In this chapter, we introduced an improved method for fast road detection including shadow and sky removals, confidence intervals application and stereo vision-based detection. The experimental results show that our method provides more stable and more precise results for drivable road detection at a reduced computational cost. The main advantages of our method are: 1. It is simple and can be suitable for real-time and on-line computations; 2. It is independent from prior knowledge of road conditions and temporal constraints. 3. Integration of stereo vision not only improved detection precision but also can provide a reliable platform for obstacle detection with binocular information.

We also constructed a confidence distribution function with our former algorithm; the results of the original algorithm using a binary map and the improved one based on a confidence map are evaluated on the KITTI-ROAD benchmark. The experimental results show that the binary map provides a high value on the $F-$measure compare to the other algorithms (second place, only behind the SPRAY algorithm). Nevertheless, when drive in complex environments, the detection performance using the binary map falls sharply. As an improved approach, the confidence map performs better in these situations, such as non flat road surface and saturated images.

Chapter 2

Monovision based Moving Object Detection

Contents

2.1 Introduction

For intelligent vehicle techniques, the most crucial task is to detect moving objects because they represent the most dangerous participants in traffic scenes. Detecting and monitoring their behavior can greatly help the drivers and autonomous driving systems to get accurate information of the scene in order to make proper decisions

and reactions. For this consideration, moving objects detection is an essential issue of dynamic scene understanding.

Several major works of moving object detection and segmentation [OB12, VRS14, NKKJ12, NHLM13, YMKC07] have been done during the last decades. Among them, monovision-based moving object detection from a mobile platform (such as robots, intelligent vehicles) has always been one of the most challenging subjects, because of the complexity of motion models and limited information for processing.

The approaches can be mainly structured into three categories: Multi-body factorization methods [YP06, RTVM08, EV09] , Graph/Layer based segmentation method [MC08, SM00, AS09] and Geometric constraints based detection [DRSS12, KKS09, YMKC07]. Each category meets different requirements for specific applications. For example, factorization methods usually incorporate affine camera model and use subspace constraints to segment the different motions. These methods provide impressive results in complex motion models. However, most of them [EV09] are based on prior assumptions and are restricted to shorter video sequences. Graph/Layer based motion segmentation methods are usually integrated with spectral clustering [NJW+02]. It can handle spatial and temporal information at the same time. But its main drawback is that it fails in the cases of complex environment, like significant occlusion phenomena or false segmentation caused by complex scene appearance (such as: extreme illumination conditions or similar texture mixed together). Geometric constraints are more effective for scene reconstruction related moving object detections like multi-body SfM (Structure from Motion) [OSVG10, HCB+13] or SLAMMOT (Simultaneous Localization, Mapping and Moving Object Tracking) [KKJ10]. But they have certain limitations when facing degenerated motion cases.

To cope with more complex situation, hybrid approaches have been proposed in recent years. For example, in [YP06], a factorization method is integrated with spectral clustering. Some other approaches, such as [PB06], proposes an incremental approach in which the detection criterion is obtained from accumulated information on time series. In this work, the framework is not limited by a certain amount of frames (ex: n-views motion detection), features are detected and tracked through the image sequence. Each time a new frame is captured, the segmentation is performed by feature grouping according to accumulated evidence over time. The movement is accumulated over time, and when it reach to a predetermined threshold, the features of this motion model will be grouped and segmented. This is to avoid the missing detection of slowly moving objects which may be hard to detected during a short period. A similar idea of evolved approach is presented in [DRSS12] for epipolar constraints based motion segmentation.

In our research, the moving object detection is specifically focused on monocular camera based perception, which, may be further integrated into a visual SLAMMOT (Simulta-

neous Localization, Mapping, and Moving Object Tracking) application[MRM+09]. In this context, the moving object detection should facilitate the future tracking processing. From this consideration, the methods of multi-body model factorization become toilsome when handling multiple complex motion models. While Graph/Layer based motion segmentation is more useful in object recognition rather than in mobile object tracking. Under such circumstances, geometric based motion detection is more practical and efficient for further development (ex: Visual SLAM). Like in [NKKJ12], geometric constraints like Flow Vector Bound constraint [KKS09] are combined with graph-based clustering to segment motions recursively. Another inspiring work called plane plus parallax approach [YMKC07] has draw many attentions recent years. In this approach, a valid homography is firstly estimated to register the static points on a 3D plane. The off-plane 3D points, also noted as parallax; together with the real moving points are detected as potential moving points by background subtraction algorithm. Then, filters composed by two and three-view geometric constraints are applied to segment moving pixels from parallax. The only concern of this approach is a prior assumption of a dominant plane in the scene and the requirement of small baseline camera motion.

In this chapter, a modified approach based on plane plus parallax method is proposed. It can be applied to more cluttering scenarios and grand camera motion (i.e. 15km/h velocity of a camera with frequency of 10Hz). Both two-views and three-views geometric constraints are applied in our approach. During the detection stage, the fundamental matrix, a recently proposed structure consistency matrix and and trifocal tensor are estimated and applied on the potential moving pixels. A dense detection result with the shape of objects is certainly desired, it helps get more precise information of the object moving model. It relies on dense optical flow estimation between frames. However, dense optical flow estimation and dense pixel classification are computational expensive. In [YMKC07], the authors use motion compensation to subtract the moving pixels and parallax pixels. Then they apply the geometric constraint to segment moving pixels from parallax ones. This is a good strategy, but its drawback is that it can be only applied in the situation that camera moves within a small baseline, and the background must be approximated by a plane. In driving situation these conditions are not always met. Thus, we need to add other processes to breakout the limitations in this method [YMKC07].

In this chapter, we propose a new combination of geometric constraints for moving object detection to cope with different degeneration situations. Differently from [YMKC07], a camera motion detection mechanism is integrated in our approach by visual odometry. Different strategies are applied according to the camera motion state. The background subtraction by homography compensation is mainly used while the camera is nearly static. When the camera is moving, the criterion values in geometric

constraint based segmentation functions are updated every time a new frame is captured by the camera. Our proposed approach can be adapted to dynamic scenarios with multiple objects moving in and out of the scene. Besides, we draw a important correction of a projective depth calculation equation which is broadly used in SfM [YMKC07, LKSV07]. The experiments are conducted on the KITTI benchmark dataset [GLU12] to evaluate the performance of our method in different traffic scenarios. The experimental results show that our approach can handle many challenging environments. The schema of moving object detection system presented in chapter is illustrated in Fig. 2.1.

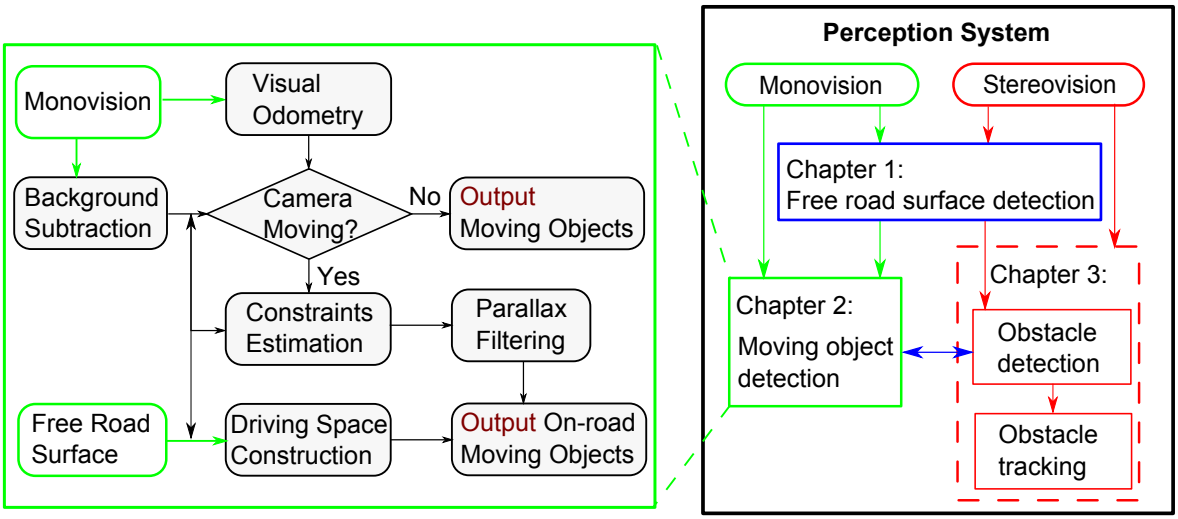


Figure 2.1 – *Subsystem of monocular moving object detection*

2.2 Multi-view Geometric Constraints

The homography based background subtraction is one of the earliest methods for motion segmentation. The main limitation of this method is that the estimation of homography and image registration relies a dominant plane in the scene. In driving scenario, the environment can not be simply described as a major plane in world coordinates. The 3D points that do not lie on the homography plane are detected as independent regions after background subtraction and they are called parallax in the scene. To isolate moving objects from parallax regions, additional geometric constraints need to be employed.

In this section, we will present the main geometries that are applied in our system. With two-views perception, homography transform and epipolar geometry are briefly introduced. Extended to three-view geometry, we will introduce a recently proposed structure consistency constraint and trifocal tensor transfer. All these geometries can be used for moving point detection.

2.2.1 Homography Transform

The concept of homography had been introduced to understand, explain and study visual perspective, specifically, the appearance difference of plane objects viewed from different points of view. It has been used in many practical applications, such as image rectification, image registration, etc.

Let us consider two different views of a planar scene in 3D space (assuming a pinhole camera model). Their projective images are related by a homography transformation through the planar surface. The homography transform is represented by a 3×3 non-singular matrix with 8 degrees of freedom. As showed in Fig. 2.2, $\boldsymbol{P}$ is an arbitrary 3D point in the planar scene, it is projected as points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ in two images separately. Let the image plane of the first view be the reference plane, the projective point $\boldsymbol{x}_2$ from the second view can be transferred to projective point $\boldsymbol{x}_1$ in the first view by:

$$\boldsymbol{x}_1 \sim \mathbf{H}_{12}\boldsymbol{x}_2 \tag{2.1}$$

where, point $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are represented in homogeneous image coordinates $(u, v, 1)$, and $\sim$ means "equals up to a scale factor". The transformation matrix $\mathbf{H}_{12}$ is called homography matrix from second view to first view. Conversely, if the second view is reference view, the points in first view can also be transferred to their corresponding points in second view by: $\boldsymbol{x}_2 \sim \mathbf{H}_{21}\boldsymbol{x}_1$, where we can get $\mathbf{H}_{21} = \mathbf{H}_{12}^{-1}$.

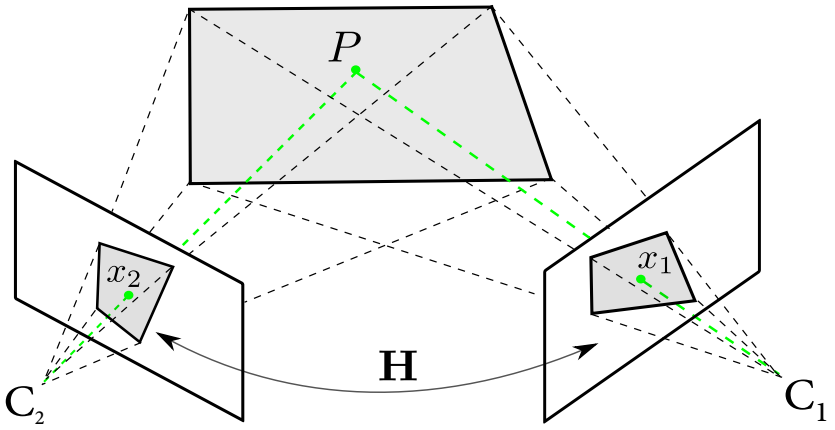


Figure 2.2 – *Homography is defined by a plane in 3D space observed in both two views. The projection of in-plane points can be transferred from one view to another.*

To be noticed that images of a rigid scene taken from pinhole cameras are related by a homography if and only if :

1. The optical center does a pure rotation motion or possibly change of camera settings.

2. The viewed scene is planar.

The homgraphy is determined by the relative motion between two views, as well as determined by the scene plane parameters. When the two views are fixed, the homography relationship is independent of the scene structure. One dominant plane can generate exactly one homography matrix that maps its in-plane points from one view to another.

With the images taken from two different views, the homography transformation can be directly estimated from a set of feature points in correspondence by the Direct Linear Transformation(DLT) algorithm [HZ04]. The linear equations used for solving homography entries can be generated by cross-producing $\boldsymbol{x}_1$ with itself:

$$\mathbf{H}_{12}\boldsymbol{x}_2 \times \boldsymbol{x}_1 = 0 \tag{2.2}$$

Since each pair of corresponding points $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ provides two independent linear equations, the eight unknowns in homography are solved from four pairs of pixels which are not collinear. However, in the presence of outliers in the corresponding points, a robust estimation scheme is needed to find the correct homography from a set of noisy points. The Random Sample Consensus (RANSAC) algorithm is a common choice, which finds a solution with the largest inlier support[FB81]. Then Levenberg-Marquardt (LM) optimization [Mar63] is applied to find the homography that optimizes a Maximum likelihood (ML) function with the inliers. The complete automatic estimation method is described in [HZ04].

Many graph-based motion detection methods estimate 2D homography for foreground and background segmentation. Here, the homography is used to to find the optimal transformation matrix that maps the scene from one view to another (which is also called the reference view) and registered with a reference frame. When the background can be approximated by a planar surface in 3D coordinates, then its projection from multiple views can be transformed to and registered in the reference view. Thus, a background model is generated from the aligned part in the registered image. This processing is called motion compensation which is mostly applied in background subtraction with slight camera motion (See Section2.2.1). The pixels on moving objects and parallax pixels do not satisfy the homography transformation; hence they are segmented as residuals from the registered background.

2.2.2 Epipolar Geometry

Let a 3D point $\boldsymbol{P}$ in world coordinate observed from two views, its projections in the two image planes are denoted by point $\boldsymbol{x}_1$ and point $\boldsymbol{x}_2$. If not collinear, point $\boldsymbol{P}$ and

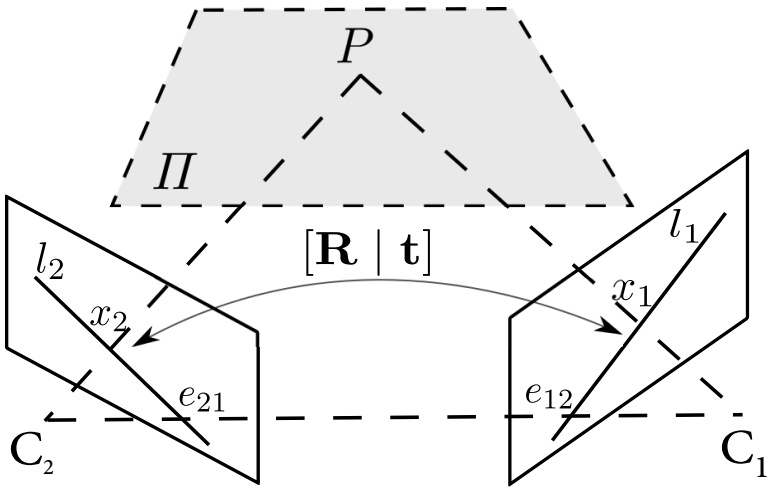


Figure 2.3 – *Epipolar Geometry between two views. The relative pose of camera between two views are enclosed in the fundamental matrix which can transfer a point from one view to an epipolar line in another view*

the camera centers $\boldsymbol{C}_1$, $\boldsymbol{C}_2$ together define an epipolar plane $\mathit{\Pi}$. This plane introduces a homography denoted as $\mathbf{H}_{\mathit{\Pi}}$, which maps point $\boldsymbol{x}_1$ in the first view to point $\boldsymbol{x}_2$ in the second view. The line connecting the two camera centers is denoted baseline, its intersections with image planes are called epipoles. The epipole which indicates the projection of second camera center to the first view is denoted as $\boldsymbol{e}_{12}$. Conversely, the epipole in second view is denoted as $\boldsymbol{e}_{21}$. The intersections of epipolar plane $\mathit{\Pi}$ with the image planes are called epipolar lines, denoted as $\boldsymbol{l}_1$ in the first view and $\boldsymbol{l}_2$ in the second view.

Given the point $\boldsymbol{x}_2$, the epipolar line $\boldsymbol{l}_2$ passing through $\boldsymbol{x}_2$ and epipole $\boldsymbol{e}_{21}$ can be written as:

$$\boldsymbol{l}_2 = \boldsymbol{e}_{21} \times \boldsymbol{x}_2 = [\boldsymbol{e}_{21}]_{\times}\boldsymbol{x}_2 \tag{2.3}$$

where, the operator $[\cdot]_{\times}$ denotes the skew-symmetric matrix form. Since $\boldsymbol{x}_2 = \mathbf{H}_{\mathit{\Pi}}\boldsymbol{x}_1$, we have:

$$\boldsymbol{l}_2 = [\boldsymbol{e}_{21}]_{\times}\mathbf{H}_{\mathit{\Pi}}\boldsymbol{x}_1 = \mathbf{F}_{12}\boldsymbol{x}_1 \tag{2.4}$$

The fundamental matrix $\mathbf{F}_{12}$ defines a mapping which transfers the point $\boldsymbol{x}_2$ to its corresponding epipolar line $\boldsymbol{l}_2$ in the first view.

If $\boldsymbol{P}$ is static, as mentioned in the first paragraph of this section, its projecting points in two different views, i.e. $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, lie right on the epipolar lines. Therefore, we have $\boldsymbol{x}_2^T\boldsymbol{l}_2 = 0$. Replacing $\boldsymbol{l}_2$ with Eq. 2.4 in the former equation, the epipolar constraint is represented as follows:

$$\boldsymbol{x}_2^T \mathbf{F}_{21} \boldsymbol{x}_1 = 0 \tag{2.5}$$

The fundamental matrix $\mathbf{F}_{21}$ is a 3×3 matrix of rank 2 that represents the geometric relationship of two views. It only depends on the cameras' intrinsic calibration matrices and the relative pose $[\mathbf{R} \mid \mathbf{t}]$ of the two camera centers[1]. Eq. 2.6 shows the calculation of fundamental matrix knowing the relative pose of second camera with respect to the first one:

$$\mathbf{F}_{21} = \mathbf{K}_2^{-T} [\mathbf{t}]_\times \mathbf{R} \mathbf{K}_1^{-1} \tag{2.6}$$

where, $\mathbf{R}$ is rotation matrix, $\mathbf{t}$ is translation vector, $\mathbf{K}_1$, $\mathbf{K}_2$ are the intrinsic calibration matrices of the two cameras.

With only the information of images from two views, the fundamental matrix can also be estimated from a set of corresponding feature points from the images. In fact, each pair of corresponding points can build a linear equation in the form of Eq. 2.5. At least 8 pairs of corresponding points can solve the entries of fundamental matrix up to a scale. In our application, the Algorithm 11.4 in [HZ04] is applied. It uses a RANSAC robust estimator and Levenberg-Marquardt non-linear optimization. In real applications, the estimation of fundamental matrix is not only used to detect motion models but also is used to reconstruct 3D camera matrices and structure of 3D space. If the intrinsic matrix of camera $\mathbf{K}$ is known, it is possible to recover the camera motion in 3D coordinates. This property is also used in our system (visual odometry that serves in Section 2.2.4).

2.2.3 Structure Consistency Constraint

In general cases, 3D scenery is not staged in a single 3D plane. The off plane points are not exactly fixed, but their position can be expressed by a residual parallax with respect to the plane.

Assuming that a plane Π in 3D space is captured by two views, according to Section 2.2.1, it introduces a homography matrix $\mathbf{H}_{12}$ that can transfer all the in-plane points from the second view to the first view. For a general point $\boldsymbol{P}$ in 3D space, its projections in the two views are denoted by points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Let point $\boldsymbol{P}'$, be the intersection of the projection ray $\boldsymbol{C}_2\boldsymbol{P}$ with the plane Π. The projection of point $\boldsymbol{P}'$ in the first view can be obtained by the homgraphy transform introduced by plane Π as follows:

$$\boldsymbol{x}_1' \sim \mathbf{H}_{12} \boldsymbol{x}_2.$$

[1]It could also be one camera taking two pictures from different views.

The second camera center $\boldsymbol{C}_2$, in-plane point $\boldsymbol{P}'$, off-plane point $\boldsymbol{P}$ are collinear, according to the invariant properties of projective transform, their projections in the first view, i.e. epipole $\boldsymbol{e}_{12}$, point $\boldsymbol{x}_1'$ and point $\boldsymbol{x}_1$ remain collinear. Hence, the point $\boldsymbol{x}_1$ can be represented by:

$$\boldsymbol{x}_1 \sim \mathbf{H}_{12}\boldsymbol{x}_2 + \kappa_{12}\boldsymbol{e}_{12} \tag{2.7}$$

where, the scalar κ_{12} is specifically defined as the projective depth relative to the reference plane Π [HZ04]. It is usually calculated by Eq. 2.8 as proposed in [FCC+03]:

$$\kappa_{12} = \frac{(\mathbf{H}_{12}\boldsymbol{x}_2 \times \boldsymbol{x}_1)^T(\boldsymbol{x}_1 \times \boldsymbol{e}_{12})}{\| \boldsymbol{x}_1 \times \boldsymbol{e}_{12} \|^2} \tag{2.8}$$

This equation is derived from Eq. 2.7 by cross-multiplying both sides of the equation with $\boldsymbol{x}_1$. If $\kappa_{12} = 0$, it means the point $\boldsymbol{P}$ is on the plane Π. Otherwise, the sign of κ_{12} indicates in which direction point $\boldsymbol{P}$ stands to the reference plane Π.

With the projective depth, the 3D points $\boldsymbol{P}$ can be represented by a projective structure constructed from 2 views:

$$\widetilde{\boldsymbol{P}}_{12} = (\boldsymbol{x}_1; \kappa_{12}) = [u_1, \upsilon_1, 1, \kappa_{12}]^T \tag{2.9}$$

If a third view is introduced, then the 3D point $\boldsymbol{P}$ can also be represented by the projective structure between the second view and the third view: $\widetilde{\boldsymbol{P}}_{23} = (\boldsymbol{x}_2; \kappa_{23}) = [u_2, \upsilon_2, 1, \kappa_{23}]^T$, where, κ_{23} is the projective depth to the new reference plane which connects the second view and the third view. There exists a relationship which links the two projective structures in form as the fundamental matrix. This relationship is called structure consistency constraint and is given in Eq. 2.10. (See [YMKC07] for the proof):

$$\widetilde{\boldsymbol{P}}_{23}^T \mathbf{G} \widetilde{\boldsymbol{P}}_{12}^T = 0 \tag{2.10}$$

where, $\mathbf{G}$ is a 4×4 matrix representing a bilinear constraint for 3D projective structures of the same point.

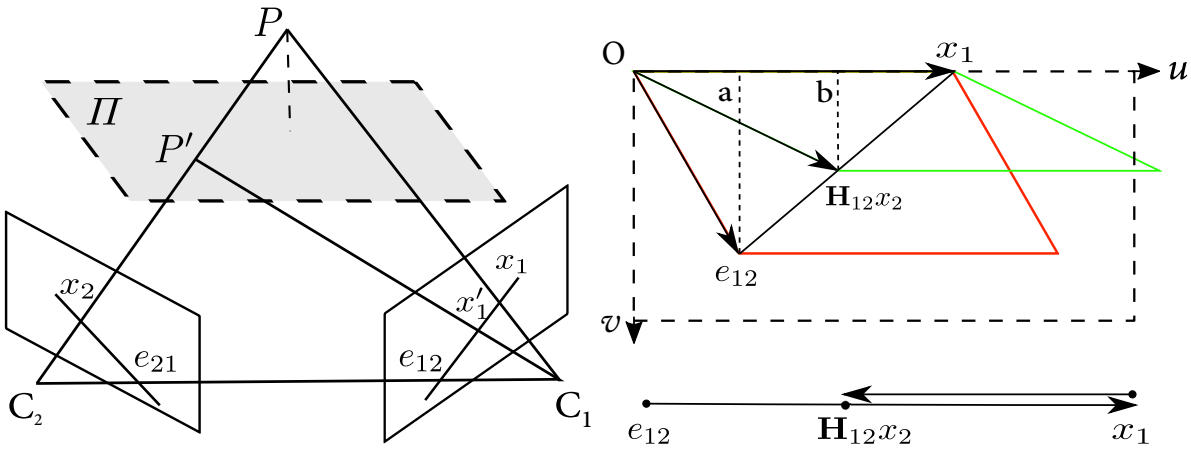
The matrix $\mathbf{G}$ encapsulates the normal vectors of two reference planes, the camera's relative orientation, and two unknown scale factors. It directly relates the pair of projective structures from views (1,2) and views (2,3) without knowing the camera configuration and the plane position. Furthermore, it can be extended to four views: the structure consistency of the 3D point $\boldsymbol{P}$ is still valid between the projective structures from view (1,2) and view (3,4). As long as the two pairs of views share the same scene, there exists a bilinear relationship $\mathbf{G}$ that connects these projective structures: $\widetilde{\boldsymbol{P}}_{34}^T \mathbf{G} \widetilde{\boldsymbol{P}}_{12}^T = 0$.

Just like the other constraints, singular value decomposition (SVD) [HZ04] is used to estimate matrix $\mathbf{G}$ by solving the Eq. 2.10 subject to $\| \mathbf{G} \| = 1$. Since the matrix $\mathbf{G}$ has 16 entries, it requires at least 15 pairs of projective structures $\widetilde{\boldsymbol{P}}_{12} \leftrightarrow \widetilde{\boldsymbol{P}}_{23}$ to find a linear solution. Generally, a robust estimation such as RANSAC should be applied with a set of corresponding projective structures to avoid the influence of image noise. However, it is hard to define a meaningful threshold for RANSAC scheme[2] with Eq. 2.10. Thus, the authors of [YMKC07] propose using Least Median of Squares (LMedS) [Rou84] estimation with LM refinement as an alternative to RANSAC estimation.

The advantage of the structure consistency constraint is its ability to detect the degenerated motions that the epipolar constraint cannot detect. For example, the point which moves along the baseline between the second view and the third view. It is more reliable in small camera motion since it relies on the estimation of homography, while epipolar constraint provides more significant results with wide camera motion.

Proposed Calculation of the Projective Depth

To construct the projective structures $\widetilde{\boldsymbol{P}}_{12}$ and $\widetilde{\boldsymbol{P}}_{23}$ for the estimation of the matrix $\mathbf{G}$ through three views, the projective depth κ_{12} and κ_{23} must be calculated for each image triplets. Eq. 2.8 has been broadly used for the parallax based works such as structure from motion [LKSV07]. However, this equation can not be applied to certain points in the image planes.



(a) *Off-plane point* $\boldsymbol{P}$ *is observed in two views, the relationship between its projections can be decomposed to planar part and parallax part.* **(b)** *The plane plus parallax composition figured in first view*

Figure 2.4 – *Plane+Parallax indication graph*

[2]The selection of threshold of RANSAC scheme for different constraint is listed in [HZ04]

As we all know, the scale of the cross product of two vectors is the area of the parallelogram with these two vectors as sides. As shown in Fig. 2.4b, the cross products of $\boldsymbol{x}_1$ with $\boldsymbol{e}_{12}$ and $\mathbf{H}_{12}\boldsymbol{x}_2$ in Eq. 2.8 can be represented by the two parallelograms (red one and green one respectively). But, when the two vectors for cross production are collinear , their cross production is zero regardless of their scale. In this case, the Eq. 2.8 is invalid when the point $\boldsymbol{x}_1$ lie on the line defined by the origin of the image coordinate $\boldsymbol{O}$ and the epipole $\boldsymbol{e}_{12}$. Otherwise, the denominator of Eq. 2.8 would be 0. Fig. 2.5 shows the result of moving point detection using Eq. 2.8. Pixels on the line which pass though the image origin and the epipole are wrongly detected as moving pixels.

To cope with this situation, we propose a new method to calculate the projective depth:

$$\kappa_{12} = \cos\theta \cdot \frac{\| \mathbf{H}_{12}\boldsymbol{x}_2 - \boldsymbol{x}_1 \|}{\| \boldsymbol{x}_1 - \boldsymbol{e}_{12} \|} \tag{2.11}$$

and

$$\cos\theta = \frac{(\mathbf{H}_{12}\boldsymbol{x}_2 - \boldsymbol{x}_1) \cdot (\boldsymbol{x}_1 - \boldsymbol{e}_{12})}{\| \mathbf{H}_{12}\boldsymbol{x}_2 - \boldsymbol{x}_1 \| \| \boldsymbol{x}_1 - \boldsymbol{e}_{12} \|} \tag{2.12}$$

where, $\cos\theta$ is the sign of κ_{12}.

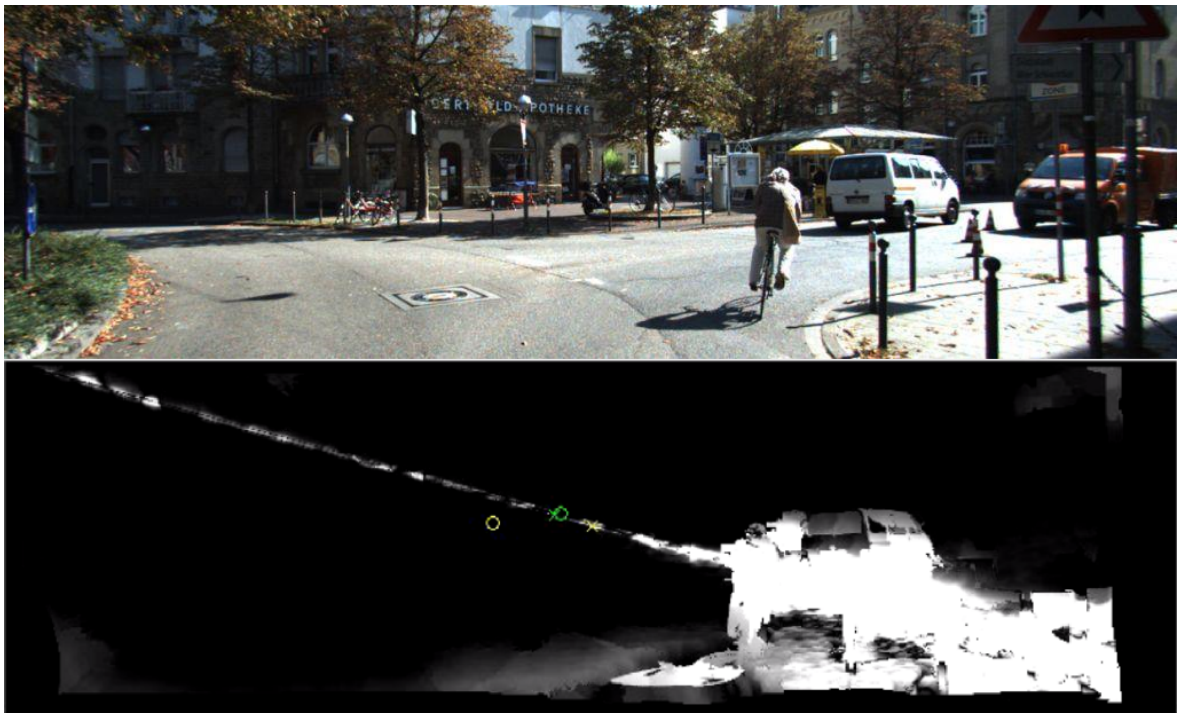


Figure 2.5 – *Example of unstable detection result caused by unmodified projective depth calculation. Top: original image. Bottom: moving pixels detected by Eq.2.8*

Proof: both the numerator and the denominator of Eq. 2.8 multiply with $(\boldsymbol{x}_1 \times \boldsymbol{e}_{12})$, we can get:

$$\kappa_{12} = \frac{(\mathbf{H}_{12}\boldsymbol{x}_2 \times \boldsymbol{x}_1)^T(\boldsymbol{x}_1 \times \boldsymbol{e}_{12})}{\parallel \boldsymbol{x}_1 \times \boldsymbol{e}_{12} \parallel^2} \cdot \frac{(\boldsymbol{x}_1 \times \boldsymbol{e}_{12})}{(\boldsymbol{x}_1 \times \boldsymbol{e}_{12})} = \cos\theta \cdot \frac{\parallel (\mathbf{H}_{12}\boldsymbol{x}_2 \times \boldsymbol{x}_1)^T \parallel}{\parallel (\boldsymbol{x}_1 \times \boldsymbol{e}_{12}) \parallel}$$

According to the property of the cross product, for any $\boldsymbol{x}_1 \neq \boldsymbol{e}_{12}$, the parameter κ_{12} can be considered as the signed area proportion of parallelograms sided by $(\overrightarrow{\boldsymbol{O}\boldsymbol{e}_{12}}, \overrightarrow{\boldsymbol{O}\boldsymbol{x}_1})$ and $(\overrightarrow{\boldsymbol{O}\mathbf{H}_{12}\boldsymbol{x}_2}, \overrightarrow{\boldsymbol{O}\boldsymbol{x}_1})$. Besides, the area of parallelogram can also be calculated by the product of its base and height:

$$A = d \cdot h \tag{2.13}$$

As in Fig. 2.4b, the height of parallelogram sided by $(\overrightarrow{\boldsymbol{O}\boldsymbol{e}_{12}}, \overrightarrow{\boldsymbol{O}\boldsymbol{x}_1})$ is denoted by a; the height of parallelogram sided by $(\overrightarrow{\boldsymbol{O}\mathbf{H}_{12}\boldsymbol{x}_2}, \overrightarrow{\boldsymbol{O}\boldsymbol{x}_1})$ is denoted by b. While the base of the two parallelograms are the same: $d = \parallel \overrightarrow{\boldsymbol{O}\boldsymbol{x}_1} \parallel$. Hence, the scale of κ_{12} is simplified as the proportion of parallelogram's heights. Using similar triangle rules, we can get:

$$\mid \kappa_{12} \mid = \frac{b}{a} = \frac{\parallel \mathbf{H}_{12}\boldsymbol{x}_2 - \boldsymbol{x}_1 \parallel}{\parallel \boldsymbol{x}_1 - \boldsymbol{e}_{12} \parallel}$$

The sign of κ_{12} indicates the direction of point $\boldsymbol{P}$ to plane Π. Projected into the second view, the sign is defined by the direction of point $\boldsymbol{x}_1$ to point $\mathbf{H}_{12}\boldsymbol{x}_2$. Considering that the points $\boldsymbol{e}_{12},\ \boldsymbol{x}_1,\ \mathbf{H}_{12}\boldsymbol{x}_2$ are collinear, the direction of the vector $(\mathbf{H}_{12}\boldsymbol{x}_2 - \boldsymbol{x}_1)$ can be represented by its intersection angle θ with vector $(\boldsymbol{x}_1 - \boldsymbol{e}_{12})$. If the two vectors are in the same direction, κ_{12} is positive, and therefore, $\theta = 0$ and $\cos\theta = 1$. On the contrary, if the two vectors are in opposite directions, κ_{12} is negative, at this moment, $\theta = 180,\ \cos\theta = -1$. As a consequence, $\cos\theta$ can be used to indicate the sign of the projective depth. It can be calculated directly from the two vectors by Eq. 2.12.

Comparing to the original method [FCC+03] to calculate κ, our proposed can be used for all the points in image plane except for the epipole. Therefore, we can replace Eq. 2.8 by Eq. 2.11 to improve the precision of the approach.

2.2.4 Trifocal Tensor

The trifocal tensor is constituted by a set of three 3×3 matrices (tensors) $\{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3\}$. It describes the projective relations of corresponding triplets and lines in three views. It is composed of 27 elements with 18 degrees of freedom. As the fundamental matrix $\mathbf{F}$ is determined by relative motion between two views (Eq. 2.6), trifocal tensor encapsulates the relative geometry of three views.

Assume the camera matrices of three views are canonical matrices: $\mathbf{P}_1 = [\mathbf{I} \mid \mathbf{0}]$ and $\mathbf{P}_2 = [\mathbf{A} \mid \mathbf{a}_4]$, $\mathbf{P}_3 = [\mathbf{B} \mid \mathbf{b}_4]$, where $\mathbf{P}_2$ and $\mathbf{P}_3$ are the projective matrices with respect

to the first frame. $\mathbf{A}$ and $\mathbf{B}$ are 3×3 matrices that are the infinite homographies from the first to the second and the third cameras respectively. The 3×1 vectors $\mathbf{a}_4$ and $\mathbf{b}_4$ are the epipoles in second view and the third view respectively, arising from the first camera. The tensors can be constructed as:

$$\mathbf{T}_i = \mathbf{a}_i \mathbf{b}_4^T - \mathbf{a}_4 \mathbf{b}_i^T \tag{2.14}$$

where, the vectors $\mathbf{a}_i$ and $\mathbf{b}_i$ are the ith columns of the camera matrix $\mathbf{P}_2$ and $\mathbf{P}_3$ for $i = 1, ..., 3$. This definition ensures a three view geometry relationship as follows:

$$\boldsymbol{l}_1^T = \boldsymbol{l}_2{}^T [\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3] \boldsymbol{l}_3 \tag{2.15}$$

serves

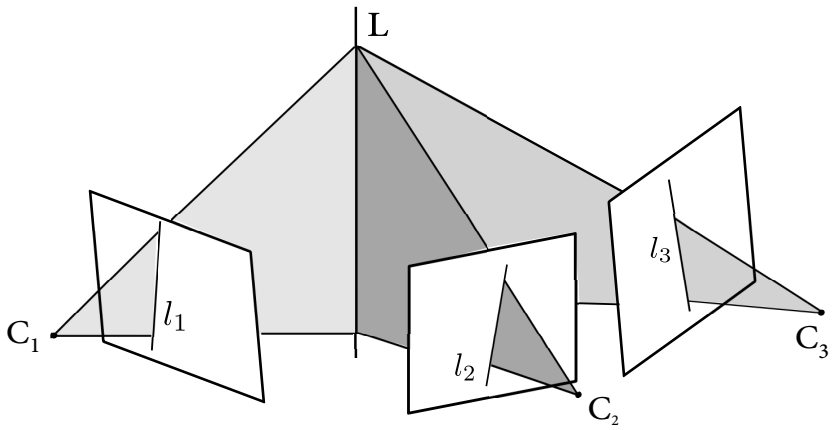


Figure 2.6 – *Back-projection of the lines from three views defines a intersection line of 3D planes in space [HZ04]*

As illustrated in Fig. 2.6, $\boldsymbol{l}_1 \leftrightarrow \boldsymbol{l}_2 \leftrightarrow \boldsymbol{l}_3$ are the projected triplets in three views of a line $\boldsymbol{L}$ in the 3D space. This equation tells that a line projected in first view can be calculated through tensors $[\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3]$ with knowing its corresponding lines in the other two views. This set of matrices is called trifocal tensor.

$$\mathcal{T} = [\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3] \tag{2.16}$$

Eq. 2.16 is the matrix notation of trifocal tensor. More commonly, since $\mathcal{T}$ has three indices, the Einstein notation (Appendix 1 in [HZ04]) is specially introduced for trifocal tensor:

$$\mathcal{T}_i^{jk} = \mathbf{a}_i^j \mathbf{b}_4^k - \mathbf{a}_4^j \mathbf{b}_j^k \tag{2.17}$$

To be noted, in our research, the matrix notation is enough for further application. Hence, in the rest of this chapter we remain using matrix notation.

Trifocal tensor can transfer a pair of points from the first view and the second view into the third view [HZ04]. Therefore, the geometric constraint becomes the comparison between transferred point location and observed point location in the third view.

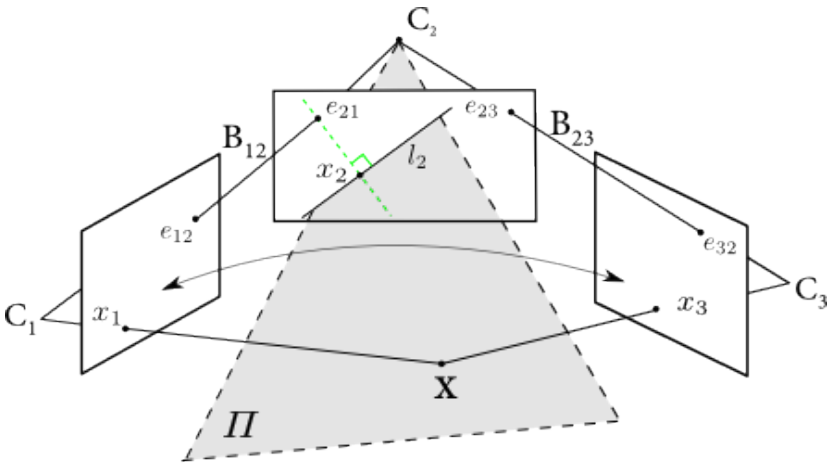


Figure 2.7 – *Point transfer through the trifocal tensor: from the two views to the third view. Figure modified from [HZ04]*

The point transfer using the trifocal tensor is based on tensor's point-line-point relationship (see Appendix C). A homography transform between the first view and the third view is encoded in the trifocal tensor by back-projecting a line in the second view to the 3D space.

As shown in Fig. 2.7, a pair of corresponding points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ in the first and the second views is given. Firstly, we define a line $\boldsymbol{l}_2$ which pass through point $\boldsymbol{x}_2$ in the second view. Together with the camera center $\boldsymbol{C}_2$, the line $\boldsymbol{l}_2$ defines a plane Π which intersects the back-projection ray of $\boldsymbol{C}_1\boldsymbol{x}_1$ at a 3D point $\boldsymbol{X}$. This point $\boldsymbol{X}$ is projected in the third view as point $\boldsymbol{x}_3$ which is exactly the corresponding point of point $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Thus, the plane Π induces the homography $\mathbf{H}_{31}$ which can transfer point $\boldsymbol{x}_1$ from the first view to the third view.

$$\mathbf{H}_{31}(\boldsymbol{l}_2) = [\mathbf{T}_1^T, \mathbf{T}_2^T, \mathbf{T}_3^T]\boldsymbol{l}_2 \tag{2.18}$$

and

$$\boldsymbol{x}_3 = \mathbf{H}_{31}(\boldsymbol{l}_2) \cdot \boldsymbol{x}_1 \tag{2.19}$$

To avoid the degenerated configuration, the line $\boldsymbol{l}_2$ in the second view is chosen to be perpendicular to the epipole line of $\boldsymbol{x}_2$ (Fig. 2.7). The method is summarized as following steps in Algo. 2.1

Algorithm 2.1 Point-line-point transfer

Input: Corresponding points $\boldsymbol{x}_1, \boldsymbol{x}_2$ in first view and second view respectively
Trifocal tensor of the three views $\mathcal{T} = [\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3]$

1: ▶ Estimate fundamental matrix $\mathbf{F}_{21}$ from $\mathcal{T}$ (See Appendix C)
2: ▶ Calculate the epipole line of $\boldsymbol{x}_1$ in the second view by $\boldsymbol{l}_{e_{21}} = \mathbf{F}_{21} \cdot \boldsymbol{x}_1$
3: ▶ Get the perpendicular line $\boldsymbol{l}_2$ to $\boldsymbol{l}_{e_{21}}$ by:

$$\boldsymbol{l}_2 = [\boldsymbol{l}^{(2)}_{e_{21}}, -\boldsymbol{l}^{(1)}_{e_{21}}, -\boldsymbol{x}^{(1)}_2 \boldsymbol{l}^{(2)}_{e_{21}} + \boldsymbol{x}^{(2)}_2 \boldsymbol{l}^{(1)}_{e_{21}}]^T$$

with $\boldsymbol{l}_{e_{21}} = [\boldsymbol{l}^{(1)}_{e_{21}},\ \boldsymbol{l}^{(2)}_{e_{21}},\ \boldsymbol{l}^{(3)}_{e_{21}}]^T$, $\boldsymbol{x}_2 = [\boldsymbol{x}^{(1)}_2,\ \boldsymbol{x}^{(2)}_2,\ 1]^T$

4: ▶ Get homography $\mathbf{H}_{31}$ by Eq. 2.18
5: ▶ Estimate projective point in the third view $\boldsymbol{x}_3 = \mathbf{H}_{31} \cdot \boldsymbol{x}_1$

The strength of using trifocal tensor based point transfer is that it has less degenerated configurations than fundamental matrix based point transfer in three views (See Section 15.3 in [HZ04]).

2.3 System Design and Realization

In the work of Yuan.C et al. [YMKC07], the parallax pixels and moving ones are first isolated from the background by means of a homography registration. Such homography is constructed through a certain number of successive frames. Then the other geometric constraints, like the epipolar constraint, are used to filter the moving objects from the parallax pixels. They also propose a novel three-view constraint, called the "structure consistency constraint" as an alternative to the trilinear constraint. Pixels which are consistent with the constraints are parallax, the others that show inconsistencies are detected as moving pixels. Nevertheless, their approach can only be applied to image sequences with a small inter-frame baseline motion and scenes should contain a dominant plane. However, either of these conditions can be filled under the circumstance of moving object detection in traffic scene. The sequences are used in the experiment in [YMKC07] are either from hand-hold camera (small baseline) or from an unmanned aerial vehicle (UAV) mounted camera (the observed scene can be approximated by a dominant plane, for example, the ground plane).

In our research, we focus on developing a system that can be applied for ADAS. Based on this, the KITTI dataset that aims to develop a challenging benchmark for autonomous driving related topics, is an appropriate choice for our experiment. In this dataset, the vehicle mounted cameras recorded traffic scenes from urban road, rural area and high-ways. Examples in Fig. 2.8 illustrate the challenges in driving scenarios: in urban area, strong parallax are almost everywhere; while on the high-way, the small baseline of the camera motion is almost impossible. These constraints make the moving objects detection problem more difficult.

(a) *Urban traffic scene*

(b) *High way*

Figure 2.8 – *Examples of traffic scene in the KITTI dataset*

In order to breakout the limitation of plane plus parallax based methods, and to build moving object detection system for intelligent vehicles, modifications must be introduced based on plan plus parallax methods. In our moving object detection system, a background subtraction approach based on homography registration is first applied to preserve potential moving pixels[3] in a residual image. At the same time, the camera motion is estimated by means of a monocular visual odometry algorithm. If the camera is static, the moving objects in the scene are exactly the result of background subtraction. On contrary, if the camera is moving, geometric constraints as fundamental matrix $\mathbf{F}$ and the trilinear structure consistency matrix $\mathbf{G}$ need to be estimated for potential moving pixels classification. The trifocal tensor is added so as to enhance the moving pixels classification. Before applying the geometric constraints on the residual image, the road detection results detailed in Chapter 1 are used to define a driving space (i.e. ROI) in order to reduce the computing time. The corresponding points of potential moving pixels in multiple views are obtained through dense optical flow estimation referred in [Liu09]. Likelihood is assigned to each constraint based on detection results. Finally, a suitable fusion function is defined to combine information from the different constraints to segment the parallax from potential moving ones. This processing is called as parallax filtering. After rolling out the moving pixels outside driving space, on-road moving objects are then detected using a blob analysis approach. Parameters are updated with every new frame acquired by the camera. The details of the system are illustrated in Fig. 2.1.

[3]Because there are parallax pixels included, so the residual pixels after background subtraction can only be treated as potential moving pixels

2.3.1 Background subtraction approach

The key to background subtraction is to build a background model to distinguish between moving pixels generated from moving objects and the ones induced by the camera motion. By comparing the current frame with this reference background in the scene, foreground moving objects can be detected by a simple pixel value subtraction operation. Considering that the main system is built upon multiple view geometry, the background model is therefore constructed by homography based image registration. As mentioned in Section 2.2.1, static points on a 3D plane can be transformed from the first view to the second view through a homography transform. For the same scene, frames that are taken from different views can be wrapped and aligned into a single reference image. The pixels showing consistency on intensity are considered as background in this reference image. Those who do not locate at the same position in reference image are either parallax pixels or moving ones.

In this section, we follow the motion compensation method proposed in [YMKC07]. In this work, the images are registered within a sliding temporal window $W_{regis}(t_0) = [t_0 - \Delta t, t_0 + \Delta t,]$ to the reference frame t_0, where Δt is the half temporal window size. The inter-frame motion caused by the moving camera is compensated by wrapping all the frames within the window $t \in W_{regis}(t_0)$ to the reference frame t_0 by homography: $\boldsymbol{p}_{t_0} \sim \mathbf{H}_{t_0,t}\boldsymbol{p}_t$. After the motion compensation, the pixel intensities of the aligned image at frame t_0 are used to represent the background.

Potential moving objects and parallax regions are detected by subtracting the estimated background from the original frame. The pixels with intensity differences larger than a given threshold ς are noted as residual pixels (Parallax+Moving objects). The proposed procedure is summarized in Algo. 2.2.

It is important to note that the homography transform can only be established with a planar scene and smooth inter-frame motion assumption. However, the driving scenarios are always challenging and these conditions may not be valid in many situations. As illustrated in Fig. 2.9, there exist many parallax pixels that take the majority of the whole residual pixels. These parallax pixels are mostly related to hard-to-track points, for example: trees, leaves, etc. Landmarks on the ground may also be detected as residual pixels. This is because the homography plane Π for the image transformation is usually chosen automatically as a virtual plane with a small angle from the camera plane during the estimation. Thus is not an ideal choice for vehicle mounted camera. This is because the ground plane does not occupy the majority area in driving scenery (there is sky area, building faces, near obstacles, etc). Besides, the road surface is an area with lack of features, where it is hard to have points defining the ground plane as the homography plane. Therefore, the landmarks that do not lie on the homography plane Π are detected as parallax pixels after the background subtraction. To handle

Algorithm 2.2 Background subtraction algorithm

Input: - Successive frames in gray scale $I_{t_0-\Delta t}, ..., I_{t_0}, ..., I_{t_0+\Delta t}$

Output: Residual image at frame t_0: I_{res,t_0}

1: ▶ Extract feature points between two successive frames, e.g. I_t, I_{t-1}

2: ▶ Estimate the homomgraphy between the two successive frames, e.g. $\mathbf{H}_{t,t-1}$

3: **for** $t = t_0 - \Delta t$ **do** $t_0 + \Delta t$

4: ▶ warp images I_t to time t_0 with $\mathbf{H}_{t_0,t}$:

$$I_{t_0,t} = \mathbf{H}_{t_0,t} \cdot I_t$$

where,

$$\mathbf{H}_{t_0,t} = \begin{cases} \mathbf{H}_{t_0,t_0-1}\mathbf{H}_{t_0-1,t_0-2}\cdots\mathbf{H}_{t+2,t+1}\mathbf{H}_{t+1,t} & \text{if } t \le t_0 \\ (\mathbf{H}_{t,t-1}\mathbf{H}_{t-1,t-2}\cdots\mathbf{H}_{t_0+2,t_0+1}\mathbf{H}_{t_0+1,t_0})^{-1} & \text{if } t > t_0 \end{cases}$$

5: **end for**

6: ▶ Background Pixels Estimation

$$I_{bg,t_0} = \frac{1}{2\Delta t + 1}\sum_{t_0-\Delta t}^{t_0+\Delta t}(I_{t_0,t})$$

7: ▶ Residual image by background subtraction $I_{res,t_0} = \text{sgn}(\text{abs}(I_{bg,t_0} - I_{t_0}) - \varsigma)$;
where, ς is a threshold to set the difference between background and original frame t_0

this problem, the parallax pixels are filtered out from the residual pixels by applying robust outlier detection methods with respect to the other geometric constraints (see Section. 2.3.4).

The aim of applying homography-based background subtraction is to narrow down the range of detecting moving pixels and to increase the computation efficiency. Another important reason, as presented in Section.2.3.5, is that background subtraction can be used as the moving object detection function when the camera is in stationary state while the other constraints can not be applied.

In our experiment, the temporal half window size Δt is set to 2, which means that 5 successive images together can generate a background model. The image intensity difference threshold ς is set at a low value, for example, 40 intensity difference with a intensity range of 0~255. Normally, the threshold needs to be adjusted to different scene configurations in order to include all the possible motion pixels and enough parallax pixels as well. If the threshold is set too high, then the motion regions may not be fully detected because the intensity difference itself is not an identical feature, and foreground objects may share similar similar intensity with the background. Thus, we suggest choosing a small threshold. Another technique can be used, is to sharpen the image within the temporal window. Assuming that the illumination of successive frames within a short time period are stable, sharpen the image will not change the in-

tensity distribution. On the other hand, it enlarges the difference between the different surfaces. Hence, it is helpful in the background subtraction result improvement.

Figure 2.9 – *Background subtraction while camera is moving*

2.3.2 Driving Space Generation

When the camera performs a big motion or moves in a narrow street with many buildings along the road, background subtraction may not performs well. Especially some strong parallax pixels along the road cannot be precisely estimated with dense optical flow. Thus, the corresponding points in multiple views are be correctly related. Geometric constraints based detection will wrongly take these pixels as moving ones. Hence, these false alarms must be removed from the detection result.

Since our system focus on extracting key information of the dynamic scene for intelligent vehicles, traffic area would be the region of interest where most important information comes from. In this section, the method for free road detection presented in Chapter 1 is integrated to generate a ROI. However, the free road surface only contains part of the driving space. Those places that are occupied by the other participants are not included. Therefore, a complete driving space still needs to be generated from this partial road surface. To this end, a convex hull operation provides the smallest convex area that contains a given subset of pixels. For instance, free road surface is the subset of the complete traffic area. The convex hull applied on I_R can circumscribe the holes and the depressions caused by on-road obstacles. Generally, such a convex area which contains the free road surface can be considered as the desired traffic road area i.e. the ROI. Obstacle detection can be focused on this approximated road traffic

area. Even if the convex hull may not exactly follow the shape of the road, in most of the cases, it is sufficient to provide a satisfying ROI to apply geometric constraints.

Figure 2.10 – *Example of driving space construction*

2.3.3 Estimation of multi-view geometric constraints

The proposed geometric constraints for moving object detection need to be computed with at least three views: two views for the epipolar constraint, three views for structure consistency constraint and trifocal tensor. All these constraints can be estimated from corresponding triplets. Correspondent triplets are extracted from three key frames with a temporal interval ϖ between the successive key frames. In our system the temporal interval between key frames for triplets extraction is defined as: $\varpi = 2$. It means that for each three frames$\{t, t+\varpi, t+2\varpi\}$, they together compose a three views geometry. For simplicity, these frames denoted as $1 :\mapsto t, 2 :\mapsto t+\varpi, 3 :\mapsto t+2\varpi$. In the remaining part of this section, this notation will be used to indicate the three frames in geometric constraint equations.

To find the matched feature points through three frames, we use the Libvisio2 toolbox from [GZS11]. This toolbox detects the feature points with Sobel filter and blob detector. The features points are tracked through the three views by and are noted as a triplet set $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3\}$. The subscript indicates which view is the point lies in. A bucketing strategy is used during the feature extraction to make sure that the feature points are well spread all over the image. This toolbox is also used to estimated camera motion using the proposed visual odometry algorithm. The camera motion is not only used to detect the camera state (static or moving), but is also used to estimated the trifocal tensor.

Epipolar constraint estimation

The fundamental matrix $\mathbf{F}_{13}$ between each 2 discrete frames I_t, $I_{t+2\varpi}$ from the triplets set $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3\}$ is estimated. Only the points in the first view $\boldsymbol{p}_1$ and the third view

$\boldsymbol{p}_3$ are used for the estimation. This is because fundamental matrix estimation is more accurate with wider baseline, compared to the other constraints. Wide baseline indicates significant camera motion, fundamental matrix which encloses the motion of the camera thus can be properly estimated in the presence of image noise.

For the implementation, a RANSAC robust estimation with the normalized 8-point algorithm[FB81] is firstly applied to find an initial estimation of fundamental matrix $\hat{\mathbf{F}}_{13}$, and its inlier triplets $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3\}^{\mathbf{F}_{13}}$ accordingly. Then the Levenberg-Marquardt algorithm is used to refine the fundamental matrix from all corresponding inliers by minimizing the re-projection error of the corresponding points:

$$\mathbf{F}_{13} = \arg\min \sum \mathrm{d}(\boldsymbol{p}_1, \hat{\boldsymbol{p}}_1)^2 + \mathrm{d}(\boldsymbol{p}_3, \hat{\boldsymbol{p}}_3)^2 \tag{2.20}$$

where, $\boldsymbol{p}_1$ and $\boldsymbol{p}_3$ are the corresponding points from the triplet set $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3\}$, $\hat{\boldsymbol{p}}_1$ and $\hat{\boldsymbol{p}}_3$ are estimated corresponding points that satisfy $\hat{\boldsymbol{p}}_1^T \mathbf{F}_{13} \hat{\boldsymbol{p}}_3 = 0$. d stands for the geometric distance. The complete fundamental matrix estimation algorithm is introduced in [HZ04].

Structure consistency constraint estimation

Before estimating the structure consistency matrix $\mathbf{G}$, new structures $\widetilde{\boldsymbol{P}}_{12} = [u_1, \upsilon_1, 1, \kappa_{12}]^T$ and $\widetilde{\boldsymbol{P}}_{23} = [u_2, \upsilon_2, 1, \kappa_{23}]^T$ in Eq. 2.10 must be constructed with triplet set $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3\}$. The homogenous coordinates of the points are $\boldsymbol{p}_1 = [u_1, \upsilon_1, 1]^T$ and $\boldsymbol{p}_2 = [u_2, \upsilon_2, 1]^T$. The key to construct new structures is to estimate the relative structure depth κ_{12} and κ_{23}.

Eq.2.11 has give a modified method of calculating relative structure depth. First, homograph $\mathbf{H}_{12}$ and $\mathbf{H}_{23}$ between each two successive views are estimated from correspondent triplet set $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3\}$. The inliers from the estimation are preserved as $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3\}^{\mathbf{H}_{12}}$and $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3\}^{\mathbf{H}_{23}}$ respectively.

From these homographies, the epipole $\boldsymbol{e}_{12}$ and $\boldsymbol{e}_{23}$ can be extracted as the intersection of the lines $(\mathbf{H}_{12}\boldsymbol{p}_2^{\sim\mathbf{H}}) \times \boldsymbol{p}_1^{\sim\mathbf{H}}$ in first view and $(\mathbf{H}_{23}\boldsymbol{p}_3^{\sim\mathbf{H}}) \times \boldsymbol{p}_2^{\sim\mathbf{H}}$. The set $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3\}^{\sim\mathbf{H}}$ is composed of the outlier triplets after homography estimation. Here, we do not estimate the epipoles from the epipolar relationship between corresponding points. This is because the epipoles in Eq. 2.11 must be compatible with the homography. If the epipoles are estimated from fundamental matrix, their positions are most likely to be deviate from the ones estimated from homography. It is possible to solve the compatibility problem by refining the estimation of related fundamental matrix and homography, however, it is a big cost for our system.

In the presence of image noise and erroneous matches, using the whole triplet set will introduce unnecessary errors and will influence the accuracy of the estimated $\mathbf{G}$ matrix.

From this consideration, the matrix $\mathbf{G}$ is estimated from a triplet set that is composed of the inliers from epipolar geometry estimation and homography estimation. This specialized set of triplets is noted as $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3\}^{\mathbf{G}}$:

$$\{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3\}^{\mathbf{G}} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3\}^{\mathbf{F}} \cup \{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3\}^{\mathbf{H}_{12}} \cup \{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3\}^{\mathbf{H}_{23}} \tag{2.21}$$

The relative structure depth κ_{12} and κ_{23} are then calculated only for the triplets in this set. On one hand, this processing promises a more reliable estimation, on the other hand, it reduces the calculation load for constructing the new structures, i.e. $\widetilde{\boldsymbol{P}}_{12}$ and $\widetilde{\boldsymbol{P}}_{23}$.

After this preliminary process, matrix $\mathbf{G}$ can be estimated by solving the equation $\widetilde{\boldsymbol{P}}_{23}^{T}\mathbf{G}\widetilde{\boldsymbol{P}}_{12} = 0$, subject to $\| \mathbf{G} \| = 1$. The RANSAC estimation needs a threshold on the measure of residual error to define the inliers. In our case, this threshold is hard to be selected, because the residual error of structure consistency constraint is not a geometric distance. In [YMKC07], they employ the LMedS estimator, which is followed by a LM non-linear refinement [HZ04]. Since the matrix $\mathbf{G}$ is a 4×4 matrix with 16 entries, 15 equations are sufficient for a linear solution. The LMedS estimator randomly selects 15 pairs of projective structures to compute the $\mathbf{G}$ matrix, and the squared residual errors from this estimation over the whole set of projective structures. After a large number of iterations, the $\mathbf{G}$ matrix, which minimizes the median residual error, is considered as the correct solution. Any points with their smaller errors than the median are classified as inlier points, whereas the rest are outliers. An implicit assumption made by the LMedS estimator is that the outlier points take up less than 50 percent of the whole set of points such that the median error reaches its minimum when the correct solution from inlier points is obtained.

In our method, we increase the percentage of inliers from median to a 70 percent. This will ensure the optimization solution is obtained from a majority of the pixels. Thus, the estimation shall be more accurate and fit the most part of the background.

Before solving $\mathbf{G}$, we perform a data normalization to pairs of projective structures, such that the pixel coordinates and projective depth values are normalized to $[-1, 1]$. This normalization step helps reduce numerical errors and increases the robustness of the estimation [HZ04].

The modified LMedS algorithm provides an initial estimation, then a non-linear refinement algorithm is applied as described in [YMKC07]. As in Section.2.3.3, LM algorithm is used here as well, the cost function for non-linear refinement is defined by mean square error (MSE):

$$\frac{1}{n}\sum \mid \widetilde{\boldsymbol{P}}_{23}^{T}\mathbf{G}\widetilde{\boldsymbol{P}}_{12}^{T} \mid^2$$

After non-linear refinement, the $\mathbf{G}$ itself need to be normalized to ensure that $\| \mathbf{G} \| = 1$.

Trifocal tensor estimation

Unlike the other constraints, estimating the trifocal tensor from triplets is a non-trivial task. It requires accurate point correspondences and large camera motion. Therefore, we choose to construct the trifocal tensor directly from its definition. If the camera matrices are known, with mono-visual odometry by Libvisio2 toolbox [GZS11], we can estimate the camera motion $[\mathbf{R} \mid \mathbf{t}]$ between each pair of successive images. Knowing the camera motion and the camera intrinsic parameters, it is possible to construct the canonical projective matrices of three views from the camera. Therefore, trifocal tensor from these three views can be built from the camera matrices according to Eq. 2.14. For that purpose, we applied Algo. 2.3.

Algorithm 2.3 trifocal tensor construction

Input: Set of triplets from three views $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3\}$
Output: Trifocal tensor of the three views $\mathcal{T}$

1: ▶ Get the camera motion $Tr_{12} = [\mathbf{R}_{12} \mid \mathbf{t}_{12}]$ by $\{\boldsymbol{p}_1, \boldsymbol{p}_2\}$ and $Tr_{23} = [\mathbf{R}_{23} \mid \mathbf{t}_{23}]$ by $\{\boldsymbol{p}_2, \boldsymbol{p}_3\}$
2: ▶ Set the perspective matrix from 3-views as:
$\mathbf{P}_1 = \mathbf{K}[\mathbf{I} \mid \mathbf{0}]$, $\mathbf{P}_2 = \mathbf{K}[\mathbf{R}_{12} \mid \mathbf{t}_{12}]$, $\mathbf{P}_{23} = \mathbf{K} \cdot Tr_{12} \cdot Tr_{23}$
3: ▶ Define canonical matrix transform as:
$\mathbf{H} = [\mathbf{P}_1; 0\ 0\ 0\ 1]^{-1}$
4: ▶ Transform the perspective matrices into canonical matrices by:
$\mathbf{P}'_i = \mathbf{P}_i \cdot \mathbf{H};\ i = 1, 2, 3$
5: ▶ Note the canonical matrices as:
$\mathbf{P}_1 = [\mathbf{I} \mid \mathbf{0}]$ and $\mathbf{P}_2 = [\mathbf{A} \mid \mathbf{a}_4]$, $\mathbf{P}_3 = [\mathbf{B} \mid \mathbf{b}_4]$
6: ▶ Construct trifocal tensor $\mathcal{T} = [\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3]$ by Eq. 2.14:

2.3.4 Moving Object Detection

Before filtering the parallax pixels, dense pixel correspondences [Liu09] are established between and restricted to the residual pixels in frames $\{t, t + \varpi, t + 2\varpi\}$. After background subtraction, the number of residual pixels is smaller than the number of image pixels and therefore helps to reduce the computation load. However, the optical flow computation on KITTI dataset is still a challenge because the images in this dataset exhibit more realistic imaging conditions, with cast shadows, glare, specular reflections, changes in camera gain, etc. In our experiment, we choose the code proposed in [Liu09] to calculate the optical flow. This work is a good trade off between accuracy and efficiency.

Applying the matrices and tensors estimated in Section 2.3.3 to the residual pixels after background subtraction, we can get three different residuals related to the proposed

geometric constraints. The residuals of each pixel indicate its bias with respect to the constraints. From statistic analysis, we can get the likelihoods of a pixel being mobile according to its residual values on the measurement of each constraint. Finally, we combine the likelihoods from three constraints with different weight to calculate the possibility of the pixel belonging to a mobile category.

In the following, we will introduce how the residual values are calculated for each triplet of residual pixels.

2.3.4.1 Error model of epipolar constraint

In this section, a re-projective pixel-to-line distance is defined to measure how much the pixel pair deviates from the epipolar lines:

$$r_{epi} = (\mid \boldsymbol{l}'_1 \cdot \boldsymbol{p}_1 \mid + \mid \boldsymbol{l}_2 \cdot \boldsymbol{p}'_2 \mid)/2 \tag{2.22}$$

If the epipolar constraint in Eq. 2.5 is established, the points should lie on their corresponding epipolar lines. $\mid \boldsymbol{l}'_i \cdot \boldsymbol{p}_i \mid$ are the perpendicular distances from residual pixels $\boldsymbol{p}_i$ in ith view to its relative epipolar line $\boldsymbol{l}_i$ respectively. Ideally, if the point is static, r_{epi} should be equal to 0, however, because of the image noise, it is more like a positive value close to 0, thus r_{epi} is called the residual of epipolar constraint. Eq. 2.22 can be written in the quadratic form as in Eq. 2.5. If we assume the noise of points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ follow the normal distribution, for the static points, their re-projective distance from two views should follow a Chi-squared $\sigma^2\chi^2$ distribution as show in Fig. 2.11. If r_{epi} is a value bigger than the value of majority inliers, then it is more likely the pixels is mobile in the scene. Since the residual is calculated by the distance from point to line, the co-dimension of the Eq. 2.22 is 1, thus it can be modeled by $\sigma^2\chi^2$ distribution with rank 1 [HZ04]. Fig. 2.11 shows the histogram of residual values measure on the inliers obtained from constraint estimation.

2.3.4.2 Error model for the structure consistency constraint

Because there is no geometric meaning of the projective structure built for the consistency constraint, the error function employed to measure the consistency of a pair of projective structures is defined as:

$$r_G = \mid \widetilde{\boldsymbol{P}}_{23}^T \mathbf{G} \widetilde{\boldsymbol{P}}_{12} \mid \tag{2.23}$$

If $r_G \to 0$, then the two projective structures are consistent with the $\mathbf{G}$ matrix and the corresponding points are static. After normalization, we can assume the noise

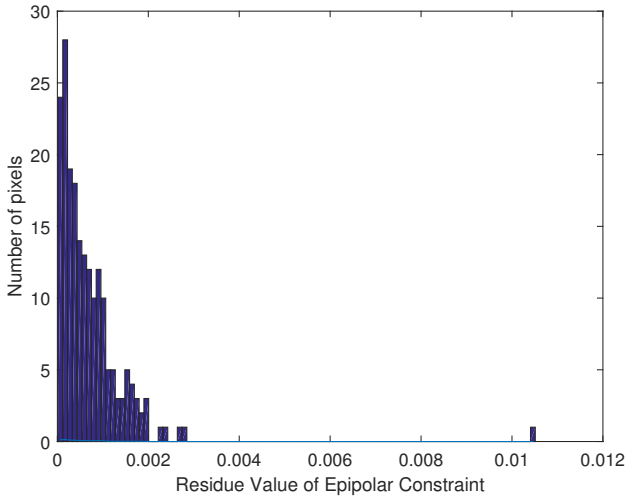


Figure 2.11 – *The histogram of r_{epi} on the inliers follows the χ^2 distribution of rank 1*

of each element in projective structures have the same normal distribution deviation. Just as for the epipolar constraint, the residual of structure consistency constraint as well follows a $\sigma^2\chi^2$ distribution because it is computed from a quadratic form (see Fig.2.12). Intuitively, if we consider $\widetilde{\boldsymbol{P}}_{23}$ as a 3D point, then $\mathbf{G}\widetilde{\boldsymbol{P}}_{12}$ creates a 3D plane in which $\widetilde{\boldsymbol{P}}_{23}$ should lie. Therefore, r_{epi} can be simulated by $\sigma^2\chi^2$ distribution with rank 1 [HZ04].

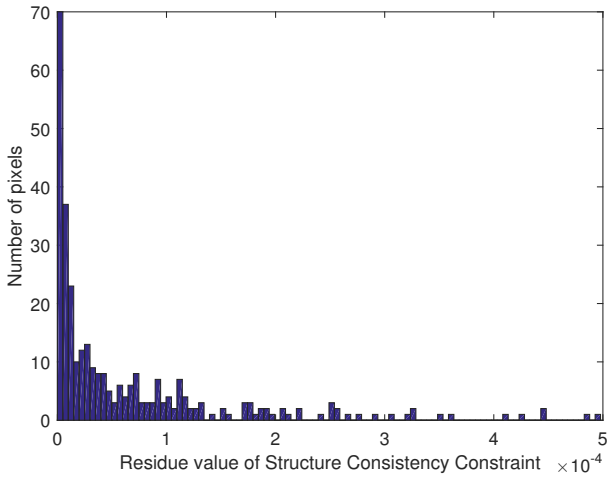


Figure 2.12 – *The histogram of r_G on the inliers follows the χ^2 distribution of rank 1*

2.3.4.3 Error model of trifocal tensor based point transfer

Through the transform of point-line-point transform, the projective position of a 3D static point in the third view can be estimated by trifocal tensor and its projective

position in the first two views. The estimated point $\boldsymbol{x}'_3$ should be the same as the point $\boldsymbol{x}_3$ in correspondent triplet. From this consideration, the condition $\| \boldsymbol{x}'_3 - \boldsymbol{x}_3 \| = 0$ is the constraint that should be satisfied by static points. The distance between these two points are considered as residuals that can be used to measure if the point is static or not. As show in Fig. 2.13, the histogram of residuals from trifocal tensor based point transfer on inliers follows the chi-square distribution as well. But in this equation, it calculates the distance between points, and each point has 2 degree of freedom, thus the co-dimension is 2. The residuals on static points should follow a $\sigma^2\chi^2$ distribution of rank 2.

$$r_T = \| \boldsymbol{x}_3 - \boldsymbol{x}'_3 \| \tag{2.24}$$

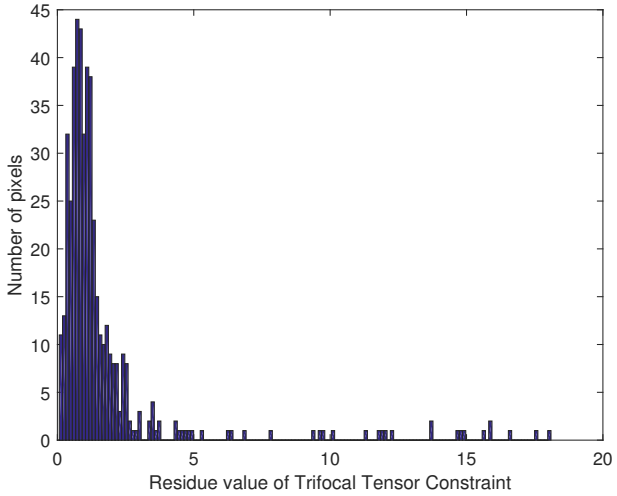


Figure 2.13 – *The histogram of r_T on the inliers follows χ^2 distribution of rank 2*

2.3.4.4 Likelihood function definition for the constraints

If we take a confidence interval of 95%, the pixels whose residual falls in this interval are considered as static. A pixel will have three different criteria according to its residual value on each constraint. According to [HZ04], for χ^2 distribution with rank-1, the residual of static point will have 95% of possibility that falls in $3.84\sigma^2$, where σ is the standard deviation of residual values from the inliers. As to the χ^2 of rank-2, the bound is $5.99\sigma^2$. On the other hand, for pixels that their residual value fall out of the 95% interval, the larger the residual values are, the more likely those pixels are mobile. Based on these analysis, we can build likelihood functions that define how much likely a pixel being mobile.

For the epipolar constraint and structure consistency constraint, the likelihood of a pixel being mobile is defined as:

$$\mathcal{L}_F(\boldsymbol{p}) = \begin{cases} 1 - e^{-\lambda_F r_F} & r_{epi} > 3.84\sigma_F^2 \\ 0 & r_{epi} \leqslant 3.84\sigma_F^2 \end{cases} \tag{2.25}$$

$$\mathcal{L}_G(\boldsymbol{p}) = \begin{cases} 1 - e^{-\lambda_G r_G} & r_G > 3.84\sigma_G^2 \\ 0 & r_G \leqslant 3.84\sigma_G^2 \end{cases} \tag{2.26}$$

As to trifocal tensor constraint, the likelihood of a pixel being mobile is defined as:

$$\mathcal{L}_T(\boldsymbol{p}) = \begin{cases} 1 - e^{-\lambda_T r_T} & r_T > 5.99\sigma_T^2 \\ 0 & r_T \leqslant 5.99\sigma_T^2 \end{cases} \tag{2.27}$$

σ_F, σ_G and σ_T are the standard deviations of residuals from the inliers of each constraint, it can be estimated by maximum-likelihood estimation (MLE). $\lambda_F, \lambda_G, \lambda_T$ are the balance coefficients assigned to ensure the three likelihood functions have a similar distribution. The residuals r_{epi}, r_G, r_T from different constraints are at different scales, without the balance coefficients, the likelihoods will have big differences between each other. This is illustrated by comparing the horizontal coordinates in Fig. 2.11, Fig. 2.12 and Fig. 2.13.

2.3.4.5 Likelihoods fusion

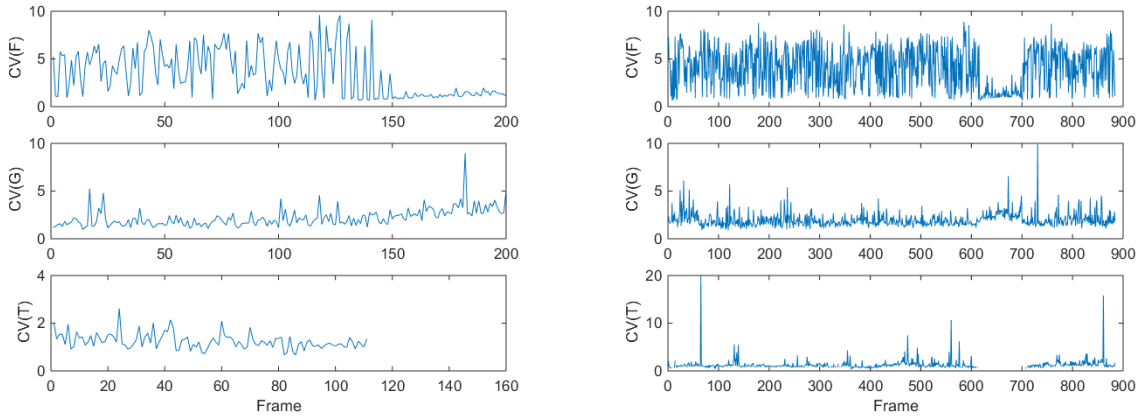
The combined likelihood of pixel $\boldsymbol{p}$ being mobile is is defined by a fusion equation:

$$\mathcal{L}(M \mid \boldsymbol{p}) = \sum w_i \cdot \mathcal{L}_i(\boldsymbol{p}) \tag{2.28}$$

where, $\mathcal{L}_i(\boldsymbol{p})$ is the likelihood distribution to each pixel $\boldsymbol{p}$ with each constraint. i indicate which constraint is applied: epipolar constraint, structure consistency constraint or trifocal tensor point transfer. Eq. 2.28 indicates how the constraints are combined together to provide a final detection result with all three constraints. w_i is the weight which is assigned to each constraint's likelihood, $\sum w_i = 1$. It can be obtained by analyzing the statistic characters of the threes constraints. During the experience, the standard coefficients of variation (abbreviation: CV, notation: $c\nu$) of each constraint's estimation residuals are analyzed and compared as a cue for the weight assignment. The coefficients of variation is calculated by Eq. 2.29. It is a standardized measure of dispersion of a probability distribution or a frequency distribution. The smaller CV value is, the more reliable the constraint is. It is defined as the ratio of the standard deviation σ to the mean μ .

$$c\nu = \frac{\sigma}{\mu} \tag{2.29}$$

The actual value of coefficients of variation is independent of the unit in which the measurement has been taken. Because of this characteristic, the coefficients of variation is usually used for comparison between data sets with different units or widely different means. In the experiments, the mean of residuals by different constraints are usually different: epipolar constraint is about 10^{-3} scale, while the mean of residuals by structure consistency constraint is about 10^{-5} or even smaller. After dividing the standard variance by mean, the ratio scale on the two different residual sets have been removed.



(a) *KITTI Sequence 2011_0926_0011* **(b)** *KITTI Sequence 2011_0929_0071*

Figure 2.14 – *The coefficients of variation (CV) comparison among three different constraints*

Fig. 2.14 shows that for different dataset, the performance of the three geometric constraints varies. In most case, structure consistency constraint is more reliable than epipolar constraint because of its smaller CV value. And the trifocal tensor commonly performs less promising than the other two constraints. But that is not for all the sequences, as illustrated in Fig. 2.14b, trifocal tensor based point transfer maintains a small CV value except for few frames. Because of the dynamics, the weights distributed to each constraint are assigned according to the recent comparison of CV values, the smaller CV value, the bigger weight will be assigned to. In this thesis, we recommend the the inverse proportion rule:

$$w_F : w_G : w_T = \frac{1}{c\nu_F} : \frac{1}{c\nu_G} : \frac{1}{c\nu_T}, \text{ with } \sum w_i = 1$$

For each frame, the weights are decided by the current CV values ratios between the constraints. We also call it self-adaptive weight distribution.

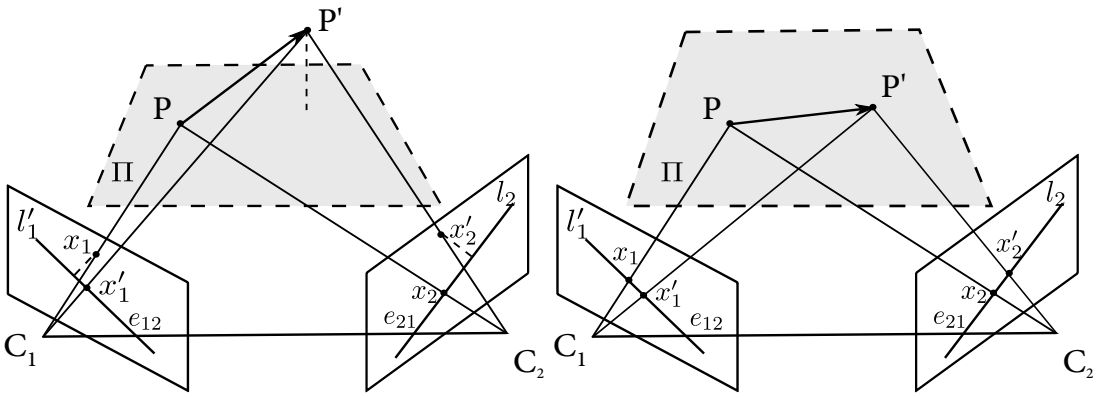
2.3.4.6 Bounding box generation for moving object detection

After classifying all the residual pixels from background subtraction, bounding box can be generated by blob analysis function in MATLAB. It provides the position and size of the grouped regions. With these information, it is possible to decide the bottom position of the object, if more than half of the object bottom is in the ROI, the object is considered as standing on the road. The other detected moving objects that considered to be outside the ROI are eliminated from the on-road moving object detection result.

2.3.4.7 Degenerated configurations

Even though, we used three different constraints for moving object detection, misdetection can still exist in special circumstances. They are called degenerated configurations, where the moving points cannot be detected by the geometric constraints.

For the epipolar constraint, a degenerated configuration happens when the camera follows the objects moving in the same direction (as shown in Fig.2.15). In order to solve this degeneracy, multi-view constraints are introduced, they are used to detect moving points across three views. But they also have different degenerated configurations.



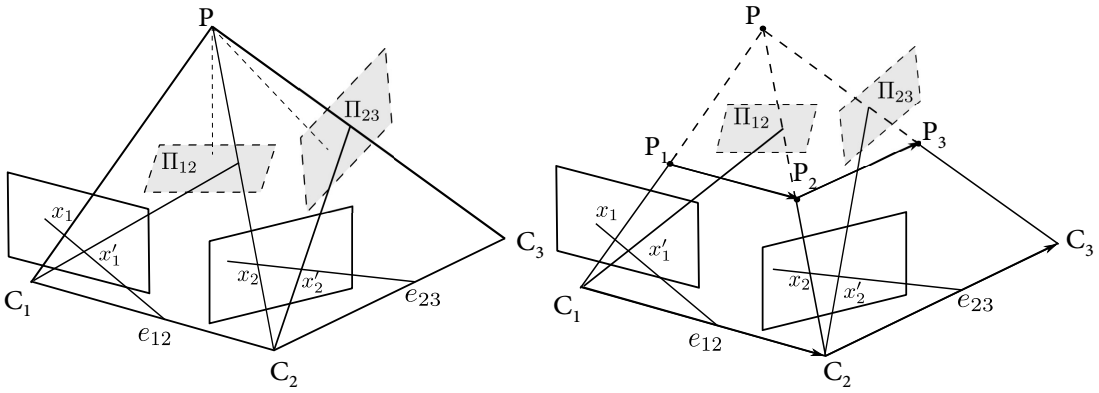
(a) *Non-degeneration: 3D point moves off the epipolar plane. The projection of 3D point after movement deviate from epipolar line in the second view*

(b) *Degeneration configuration: 3D point af-moves in the same epipolar plane, projection of 3D point after movement lies in the epipolar line in the second view*

Figure 2.15 – *Degeneration configuration of epipolar constraint*

For structure consistency, if the object moves in the same direction with the camera and keeps at a constant proportional velocity to the camera's speed, the structure

consistency constraint cannot determine whether the pixel triplet comes from a static point or a moving one. Fortunately, this subset happens rarely in urban traffic scene. Fig. 2.16 shows an example of a camera tracking a moving point across three views. The static point $\boldsymbol{P}$ has the same projections with the moving point $\{\boldsymbol{P}_1 \rightarrow \boldsymbol{P}_2 \rightarrow \boldsymbol{P}_3\}$ in the three view. This leads to the ambiguity for the structure consistency constraint based detection.



(a) *Static 3D Point Structure Consistency with respect to two reference planes. The twoture Consistency. When the 3D points moves in "plan+parallax" projective structures defines athe proportional velocity of the camera motion, point in space*

(b) *Ambiguity of mobile 3D Point in Structo The two "plan+parallax" projective structures are the same with a static 3D point in space.*

Figure 2.16 – *Degeneration configuration of structure consistency constraint*

Unlike point epipolar transfer, trifocal tensor based transfer does not fail for 3D point lying on or close to the trifocal plane which is defined by the centers of cameras. But it can not transfer the point whose 3D location lie on the baseline B_{12} in Fig. 2.4a. However, such points are extremely rare in the scene.

2.3.5 Stop-go-stop Adaptation Design

The advantage of our method is that it can be adapted to the vehicle state while it changes from stop to moving or from moving to stop. From monovision-based visual odometry, the camera motion Tr is estimated, when $Tr = 0$, it means that the camera stays static and the detection result is only based on the background subtraction. While the vehicle is moving, the detection result is defined by Eq. 2.30. If we consider a virtual residual distribution that is related to the combined likelihood in Eq. 2.28, an common choice is treating it as a natural distribution with $\mu = 0$. Generally, when the sample value is bigger than 3σ of the normal distribution, it is classified as outlier.

After checking the stablility, the likelihood value relation to 3σ is 0.6321. Hence, if the combination of $\mathcal{L}(M \mid \boldsymbol{p})$ is greater than this value, the pixel is detected as mobile.

$$M(\boldsymbol{p}) = \begin{cases} 1, & \mathcal{L}_{t|t-1}(M \mid \boldsymbol{p}) \geq 0.6321 \\ 0, & \text{otherwise} \end{cases} \tag{2.30}$$

$M(\boldsymbol{p})$ is the state of a pixel being mobile or not, state 1 means the pixel belongs to a moving object, 0 means the pixel is static parallax. All the residual pixels are classified into these two categories. After the classification, a blob analysis is applied to extract a bounding box from the moving objects. As mentioned before, there are false alarms caused by optical flow estimation. Therefore, all the objects need to be verified on the traffic area, if not, they will be rejected from detection results. In the end, the complete moving object detection algorithm can be summarized as:

Algorithm 2.4 Moving object detection

Input: Image sequence
Output: Detected on-road moving objects

1: ▶ Get potential moving pixels $\{\boldsymbol{p}_{res}\}$ by simple background subtraction
2: ▶ Camera motion Tr estimation
3: **if** $Tr = 0$ **then**
4: ▶ $M(\boldsymbol{p}) = 1,\ \boldsymbol{p} \in \{\boldsymbol{p}_{res}\}$
5: **else**
6: ▶ Estimate geometric constraints $\mathbf{F}, \mathbf{G}, \mathcal{T}$
7: ▶ Apply constraints with dense optical flow on $\{\boldsymbol{p}_{res}\}$.
8: ▶ likelihood calculation by Algo.2.28
9: ▶ Classification by Eq.2.30
10: **end if**
11: ▶ Blob analysis with output in the form of boundingbox
12: ▶ Traffic area generation.
13: ▶ Confirm the blobs which locate in the traffic area.

2.4 Experimental Results

The on-road moving object detection system is tested on KITTI datasets [GLSU13]. We selected three different sequences of urban scene for the experiments:

- Dataset 1: KITTI raw data, 2011_09_26_drive_0005 with a minivan and a cyclist continuously appearing in the sequence.
- Dataset 2: KITTI raw data, 2011_09_29_drive_0071 with narrow street passing through a commercial center, with many pedestrians and other traffic participants moving in different directions.

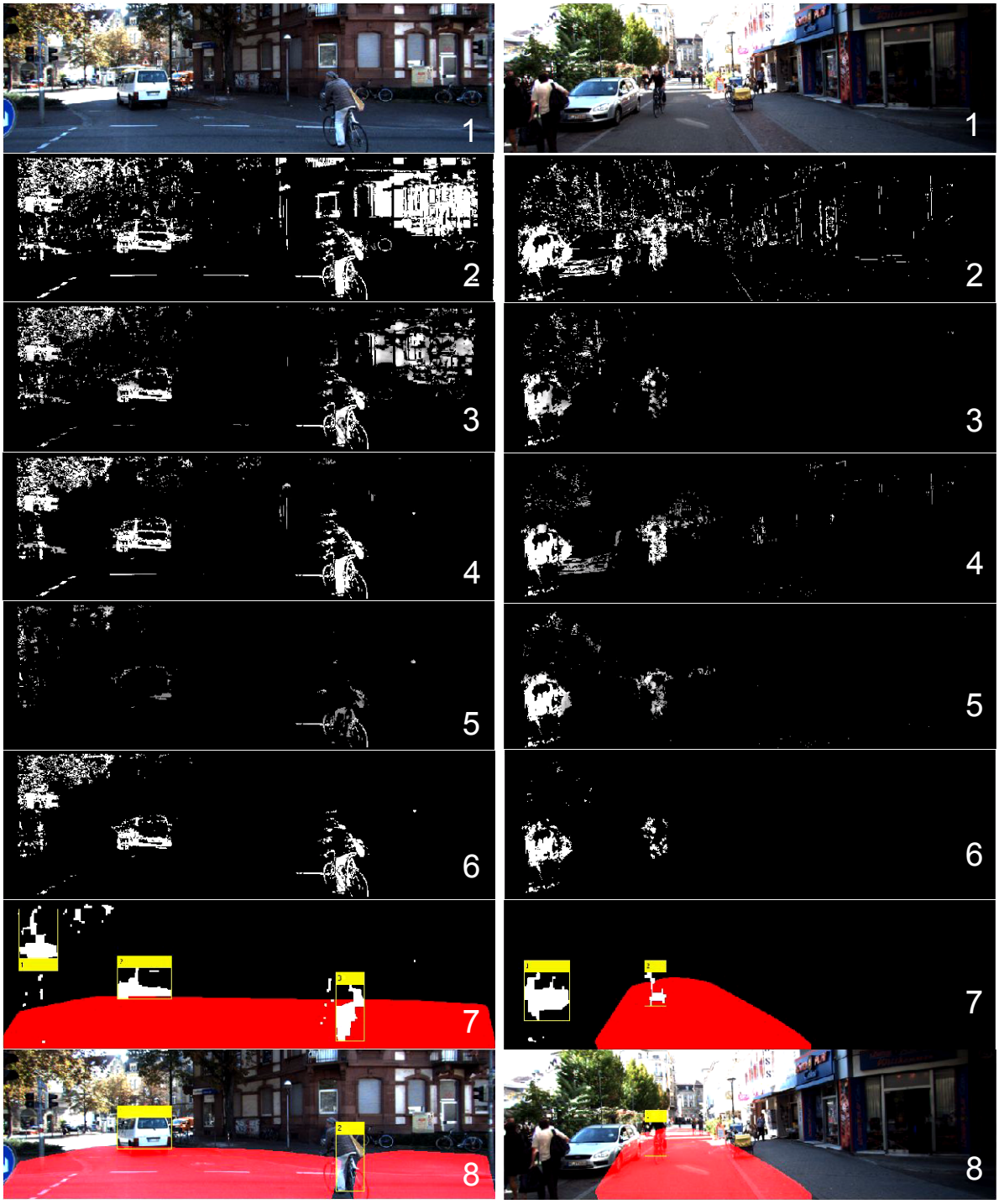
– Dataset 3: KITTI raw data, 2011_09_26_drive_0011 with host vehicle moving in structured urban road and then stopped at a intersection while the other vehicles passing by in orthogonal direction.

The algorithm is implemented on a standard PC with Windows 7 Enterprise OS, Intel CPU of 2.66 GHz. The development environment is MATLAB R2013b. The disparity maps are obtained from the LIBELAS toolkit [GZS11]. We choose to use LIBELAS because this is the toolkit developed by the founder of KITTI dataset. This toolkit is designed to perform with this dataset. Since the images of KITTI dataset have larger width compared to the other methods, LIBELAS fits the dataset best and the parameters in this toolkit can be adjusted easily by the other users. Optical flow estimation is implemented with the code of [Liu09], because this method is relatively fast among the methods that provide the same level of precision. The average error of the optical flow estimation using this method is about 3~5 pixels for KITTI dataset. This result is obtained by evaluating the estimation result on KITTI-flow benchmark.

General result

Fig.2.17 shows the detection result on the first two datasets. From top to bottom, each row in the figure presents: 1 - the original image; 2 - residual image after background subtraction; 3 - confidence map of epipolar constraint based detection; 4 - confidence map of structure consistency constraint based detection; 5 - confidence map of trifocal tensor constraint based detection; 6 - combined likelihood based detection result; 7 - traffic area construction and blob analysis; 8 - final detection result of on-road moving object detection. The comparison between the residual image of background subtraction (number 2) and the detection result of combined likelihood from geometric constraints (number 6) proves that the geometric constraints we applied, can effectively eliminate the majority of parallax in the scene.

From the result of Dataset 1, we can see that there is a false alarm after combination of the constraints. This false alarm which is located on the left side of the image is caused by the plants and the parallax in front of it. This is a common example, most of the false alarms after geometric constraint were found around trees or the occluded parallax. This is because in this region, the dense optical flow cannot be correctly estimated. Fortunately, this region mostly appears along the road or outside the road. Hence, apply the ROI of driving space can greatly reduce such kind of false alarms. However, every method has its strength and weakness. The monovision-based road detection indeed provides a promising driving space. But there still exist many false positives/negatives as demonstrated in Chapter 1. That will lead to negative impact on the moving object detection. For example, in Fig. 3.13b, on the left side



(a) *Example of moving detection in Dataset 1* **(b)** *Example of moving detection in Dataset 2*

Figure 2.17 – *Examples of on-road moving object detection in two datasets: first row to the end are: 1- original image; 2- residual image after background subtraction; 3- confidence map of epipolar constraint; 4- confidence map of structure consistency constraint; 5- confidence map of trifocal tensor constraint; 6- combined likelihood based detection result; 7- traffic area construction and blob analysis; 8- final detection result of on-road moving object detection.*

of the road there are two pedestrians walking away. They are very well detected by the geometric constraints, but according to the ROI, they are not in the traffic area, thus they are eliminated from the final detection result. This is because the left part of the road surface are completely covered by the pedestrians. Then it will not be detected as traffic area. This situation is hard to avoid in cluttered scenery. Yet we can introduce tracking or evolution algorithm to predict the presence of moving object in this situation. In this chapter, we mainly focus on the presentation of detection system. Object tracking in the 2D images will be introduced in the next chapter.

It is important to notice that this algorithm can only detect the objects within a certain distance. When the object is too far, its residuals after background subtraction only occupies a small area in the image, in this case it is more likely to be filtered out during the detection processing (ex: the processing of blob analysis). Since we are using monocular camera, it is hard to measure the distance of moving objects. Besides, it also depends on the size of objects, for example the vehicles can be detected in a further distance than the pedestrians.

Figure 2.18 – *Example of detection result of Dataset 1 and Dataset 3, red area are the ROIs, yellow boxes are the detection result, green boxes are the ground truth*

	detection rate	mis-detection	false alarms	redundant detection
Dataset 1	50.80%	49.20%	29.84%	3.66%
Dataset 3	74.64%	25.36%	6.69%	28.03%

Table 2.1 – *General evaluation of the moving object detection by monovision*

Evaluation

In order to further analyze the result of out method, we evaluated our detection results on Dataset 1 and Dataset 3 that include two completely different object motion models: object moving in the same direction with host vehicle and object moves in orthogonal direction to the host vehicle as shown in Fig. 2.18. (We didn't use Dataset 2, because for the scene of commercial center street, road surface is hard to be defined just by the appearance and plane structure. Besides, all the man-built structures and objects are clustered in the scene, the influence factors are too complex to be used for analyzing the performance of the detection method.)

As we can see from the Tab. 2.1, the detection rate of Dataset 1 is less than Dataset 3. The reason is when the objects are moving in the same direction with host vehicle, there comes configuration of degeneration. In this situation, the geometric constraints can not tell the moving pixels from static background. And the false alarm rate is higher in Dataset 1 since the scene is clustered and there are more parallax than the structured urban road. Meanwhile the redundant detections that are caused by the default of background subtraction appear more in Dataset 2. When the object is big and in homogenous appearance, background subtraction can not extract the object as a whole, just as the bus shown in Fig. 2.18.

Furthermore, we analyzed the influence of distance between the object and the host vehicle to the detection result using our method so to define the range of detection. The result is shown in Tab. 2.2: from both datasets, we can get the detection range of our method is 35m; for the object that moves in the same direction with host vehicle (as in dataset 1), the further it stands the harder it can be detected due to the perspective distortion and the configuration of degeneration. For the object that moves in the orthogonal direction (as in dataset 3), the detection rate is almost the same within the 35m distance. To be noted that the detection rate of Dataset 3 within 10m is abnormally low, this is caused by the limitation of ROI. As shown in Fig. 2.18, in Dataset 3 when the object moves close to the host vehicle, the area where it stands is not included in the ROI, thus the detection is eliminated from final result. Hence, the detection rate for Dataset 3 within 10m distance does not indicate the efficiency of geometric constraints based detection method.

detection rate	<10m	10m~15m	15m~20m	20m~25m	25m~30m	30m~35m	>35m
Dataset 1	77.53%	41.38%	42.47%	10%	11.1%	9.1%	0%
Dataset 3	21.41%*	46.15%	71.43%	74.19%	73.81%	91.43%	<5%

Table 2.2 – *Evaluation of the moving object detection by monovision based on distance factor*

Difficulties

After the homography based background subtraction, there may still exist landmarks that are detected as parallax. The reason is as mentioned in Section 2.3.1: the road surface was not chosen to be the homography plane automatically. Since the subtraction is performed on the intensity value, missing detections and incomplete detections appear when the object has a similar intensity with its surroundings. As shown in Fig. 2.19. The first left pedestrian is not detected by background subtraction, and the pedestrian in front of Pizza Hut is partially detected. To solve this problem more texture information and even new techniques are needed for moving object detection, however, it could cost more computing time.

Figure 2.19 – *Example of missing detections during the background subtraction*

Another difficulty lies in the dense optical flow estimation. Many researchers have evaluated their algorithms on KITTI-flow benchmark. Fig. 2.20 presents the work of [YMU14] which performs the best on this benchmark using monovision. Nevertheless, its optical flow estimation still contains large errors in the parallax regions and the regions lack of feature. It leads to the false alarms in these regions when the geometric constraints are employed. Even introduction of ROI can eliminated such false alarms

outside the driving space, but within the driving space, we need more stable point tracking methods.

In the simulation, the detection result with trifocal tensor performs much better than the method in [YMKC07] especially in the configuration of degeneration when object moves in the exactly same direction with host vehicle. However, in the experiments using the image sequences of real traffic scene, the improvement using trifocal tensor is not significant. For example, in Dataset 1, the mis-detection is reduced by 3 frames for the cyclist using our method. But in percentage rate it is hardly noticeable. Because the other influence factors such as dense pixel tracking and estimation of constraints, they all contribute to the final detection result. It remains a difficulty to isolate each factor for evaluation.

Figure 2.20 – *Example of optical flow evaluation on KITTI dataset.The top image is the original image for optical flow estimation, the bottom one is the error map scales linearly between 0 (black) and >=5 (white) pixels error. Red in the error map denotes all occluded pixels, falling outside the image boundaries. The method of this example is presented in [YMU14]*

2.5 Conclusion

In this chapter, we have presented a complete system for on-road moving objects detection based on monocular vision. It integrates multiple geometric constraints to detect the moving pixels in an estimated driving space. All the components together improved the efficiency and flexibility of the system: efficiency because it is concentrated on detecting the traffic participants, flexibility because the system can change its detection strategy according to the motion state of the camera/vehicle.

We also analyzed the strength and limitations of each constraint. Especially, for the structure consistency constraint, we correct the formula for calculating projective depth. This simple correction may help to improve the reliability of the approach. For each constraint, we defined a likelihood function of a pixels. Furthermore, we introduced the coefficients of variation as criteria to decide the importance of each constraints in the process of fusion of likelihoods. Another contribution of this work, is the use of visual odometry to detect the camera state, different strategies of moving objects detection are employed while the camera is static and moving.

As indicated in Section 2.4, there are still many challenges that remain to be solved. The discussion about our future research facing these challenges will be presented in the end of this thesis.

Chapter 3

Stereo Vision based Obstacle Detection and Tracking

Contents

3.1 Problem Statement

The purpose of the visual detection and tracking system is to locate objects in the frame of a video sequence and to observe their dynamic behavior. A reliable detection and tracking of on-road obstacles allows an intelligent vehicle perception system to predict the motion of other participants in surrounding area and to avoid potential

collisions. Sensors, such as radar, Lidar and stereo vision have been employed for this purpose. Sometimes, they were used together to assist each other [PLR+06, ESLVG10, MCTM05].

Vision-based systems may not be able to compete with range sensors on the geometric accuracy at present, but their advantage lies in the rich appearance information that they get from the scene. During the object detection and tracking process, this information can be used to classify objects in different categories or motion models, and to associate them with the correct tracks. Based on this unique advantage, the output of object detectors can be directly used as the observation for target tracking in the appearance-based object detection and tracking approaches [BRL+09, GLW+14, JCZT11].

However, detecting and tracking obstacles on the road is difficult for the following reasons:

1. There exists a variety of possible obstacle types, ex: car, pedestrian, traffic cones, etc.
2. The motion state (i.e. moving or static) of obstacles is unknown and it could change at any time.
3. The number of obstacles is time-varying with obstacles entering/leaving the scene
4. Noisy observation from images can lead to tracking ambiguities.
5. Tracking in the 2D image plane has the projective distortion problem (far objects appear smaller than the near ones)

In this chapter, we propose a stereo vision-based on-road obstacle detection and tracking system to better cope with these difficulties.

Region-of-Interest Analysis

In our approach, the stereo-vision based obstacle detection is applied on a region of interest (ROI) defined by the driving space where all the traffic participants stand. The authors of [BB94], use an edge detector and watershed algorithm to select the ROI for obstacle detection and tracking. As mentioned in Chapter 1, texture-based method appears less stable when the road appearance changes due to the illuminant conditions. While in [SO07], the authors extract the road area under the assumption that the road surface is planar. They estimate a homography matrix of the road plane from the two views of a stereo-rig, then they warp the right frame to the left one, and all the aligned parts after homography transform are subtracted as road area.

Unfortunately, an accurate homography estimation of road plane is hard to satisfy because of the lack of features on road surface. And it relies on a prior definition of plane region. In our approach, the fast road surface detection method described in Chapter 1 is applied to obtain the ROI of the driving space. This method is reliable under different illuminant conditions and do not rely on any assumption of the road profile. As mentioned in Chapter 2, the convex area of the detected free road surface can be considered as a complete driving space.

Projective Distortion

In the pinhole camera model (See Chapter 1), the scale of an obstacle's size and its displacement is inversely related to the depth and directly related to the disparity value. Our obstacle tracking algorithm is performed in the image plane with a moving platform. Therefore, the projective scale variation of obstacles must be handled carefully.

During the detection stage, the scale of obstacles can be used as a constraint to eliminate false alarms. While for the tracking stage, the projective distortion is the key to ensure correct track-to-target associations and reliable tracking trajectories. The discussions and application of projective distortion are presented in details in this chapter.

Obstacle Detection

In recent years, many vision-based approaches [SO07, JCZT11, KB12] have been developed to deal with these difficulties. In the semantic object detection method [FMP+13, JDSW12], only trained categories of objects can be detected or recognized. Unfortunately, these categories can not represent all types of on-road obstacles which may include: artificial structures like traffic cones, animals that are passing through the country road, and other unpredictable obstacles. On the other hand, motion-based detection methods [Hei00] fail in detecting static obstacles. In Chapter2, we presented a monovision based system of moving objects detection. This system can be extended to scene reconstruction from multiple views, thus to find and to locate the static obstacles. However, it drives the problem into a more complex level with high computational consumption. Instead, if the stereo rig is available, stereo vision [LAT02, LSH+12] based methods can effectively detect obstacles regardless their appearance or their motion model. They are only related to the disparity value of each obstacle. In addition, the depth information referred by disparity values can further contributes to object tracking.

In our stereo vision based approach, a method of connected region extraction from the U-disparity image is developed to locate the 2D position of obstacles within the ROI (i.e. the driving space). To get a more accurate detection result, the size and depth of the obstacle (represented by disparity value) are refined from a sub-image of disparity

map where the obstacle is located. In addition, to improve the reliability of detection results, additional criteria are integrated in the system, such as: an adaptive height limitation of the obstacles in different distances and a combination of small closely detected regions in U-disparity image.

Multiple-Target Tracking

After obtaining the position, size and depth of the obstacles on the road, they become the targets that need to be tracked continuously during their presence in the frames. The sequence of states that a target follows during its present period is called a track [ORS04]. The task of multiple-target tracking (MTT) is to link the corresponding detections across frames to form the object trajectories. Each trajectory is represented by a sequence of track states. Among the entire object tracking techniques, particle filters [OTDF$^+$04, NTMM12, BRL$^+$09] have been widely used to solve multiple time-varying obstacles tracking problems. Their strength lies in their ability to represent non-Gaussian distributions by a large number of samples which enclose and maintain target properties. In our approach, a modified particle filter is applied to solve MTT problem specifically for on-road obstacles in 2D image plane. The information (position, height, width, depth) of the obstacles received from detection stage is the observation/measurement in the particle filtering. The tracks of these obstacles are generated and updated by multiple filters simultaneously. During the processing, associating the tracks to their corresponding target observation becomes the key to a successful tracking system. In our approach, target-to-track association is carried out following a global nearest neighbor (GNN) criterion. Since only 2.5D position (here 2.5D means (u, v, Δ)) and size of the obstacles in images are known from the detection stage for tracking. It is then necessary to define an association criterion which is well adapted to the proposed method. To cope with projective distortion, a dynamic noise generation function and self-adaptive gating for data association are defined in the filter. Considering the number of obstacles varies over time, for each iteration of the filtering, multiple hypotheses are made to either create a track for new objects, or to delete a track of an object which has already left the scene.

The main system is structured into two parts: on-road obstacles detection based on U-V disparity image analysis; and the modified particle filter tracking of multiple targets. Fig. 3.1 shows the outline of the proposed system. The strength of this approach are: (1) It proposes a reliable detection and tracking system which can be directly applied to different driving scenarios. (2) It is capable of detecting any kind of obstacles on the road, regardless of their shapes and poses. (3) It presents a modified particle filter

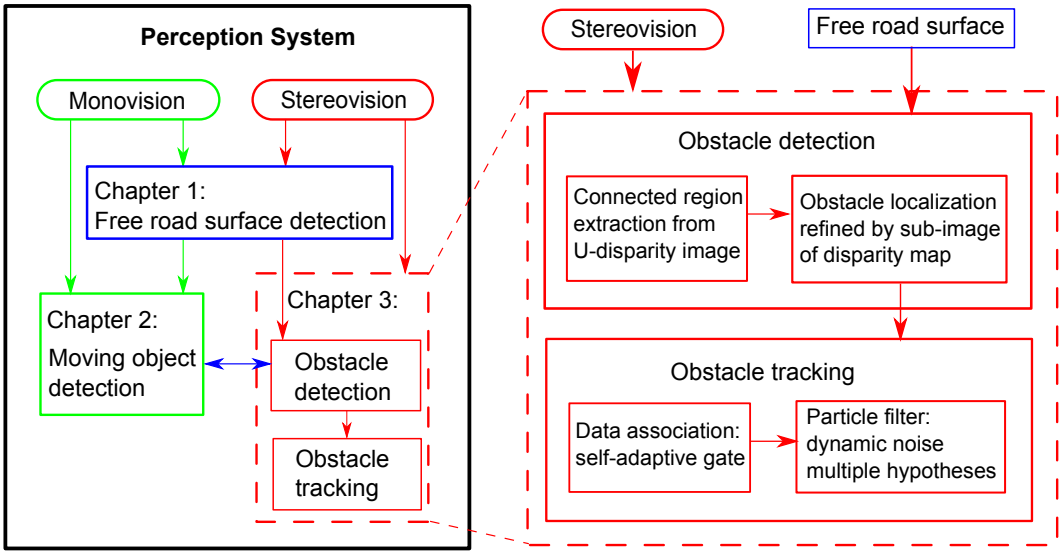


Figure 3.1 – *Outline of On-road Obstacle detection and tracking subsystem*

for visual tracking, which can tolerate the dynamics of obstacles in the images caused by the projective distortion.

Since the ROI generation has already been discussed in Chapter 1 and Chapter 2, in this chapter we will mainly focus on stereo-vision based obstacle detection (Section 3.2) and particle filtering for multiple-target tracking (see Section 3.3). The detection part starts with a primary obstacle localization within the U-disparity image in Section 3.2.1. Then it is followed by Section 3.2.2 that presents refinement detection method with sub-image of disparity map. Additional object detection criteria are briefly discussed in Section 3.2.3. For the tracking stage, a brief methodology of particle filter is first introduced in Section 3.3.1, and a dynamic particle filter which can cope with projective distortion is proposed in Section 3.3.2. The target-track association problem is discussed in Section 3.3.3.

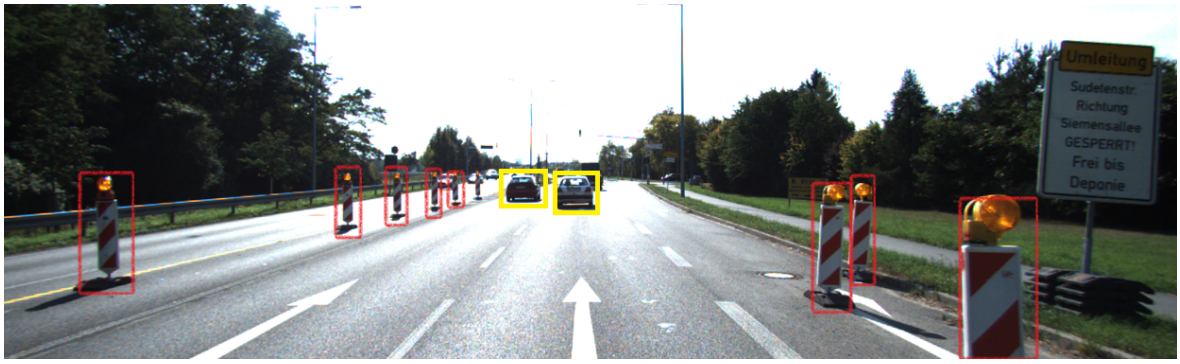
Experimental results and analysis on the KITTI dataset are presented in Section 3.4. Finally the chapter ends with conclusions and perspectives.

3.2 Stereovision-based On-road Obstacle Detection

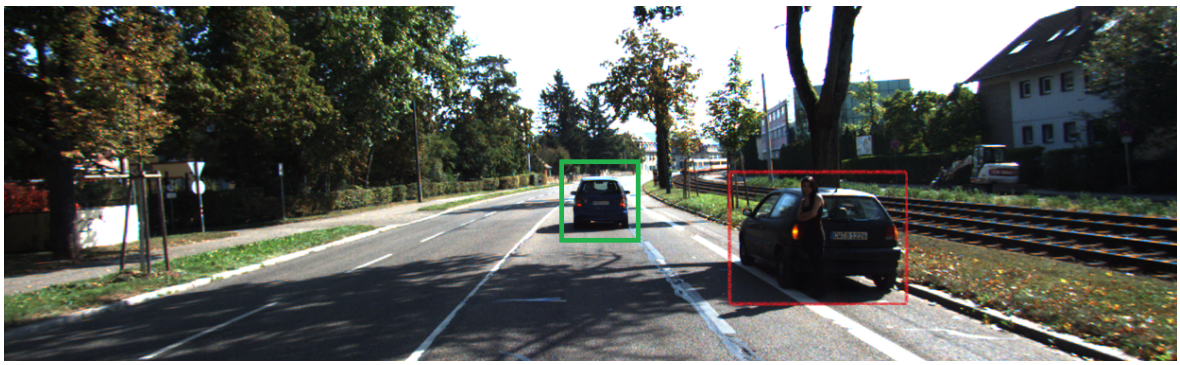
Among the recent approaches, semantic based obstacle detection [FMP+13, RLSA11] focused on the detection of specific types of objects, like pedestrians and vehicles in the context of traffic scenes. Mostly, the semantic information of a certain category of obstacles are learned from a training dataset. It requires a prior training work of possible obstacles on the road. However, in dynamic driving scenes, it is impossible to conclude all the categories of objects that may appear on the road. Some other vision-based methods extract the moving objects either by background subtraction [ZNW08]

either by motion model construction [JCZT11, JT12]. These methods are designed to detect only moving objects, other static obstacles such as traffic signs are ignored during the detection. These missing detection cases are illustrated in Fig. 3.2.

In this Section we will introduce a stereo vision based on road obstacle detection algorithm. As presented in Chapter 2, free road area detection based on illuminant invariant image is employed at an early stage. It is followed by a convex hull construction for generating a Region of Interest (ROI) which includes the main driving space. Within this ROI, a U-disparity image is built to characterize on-road obstacles. In our approach, connected region extraction is applied for obstacle detection instead of standard Hough Transform. Besides, additional object detection criteria, such as obstacle's size verification and combination of redundant detections, are embedded in the system to improve its accuracy. Experimental results, presented in Section 3.4.2, show that the system is effective and reliable when applied on different traffic video sequences from the KITTI benchmark.



(a) *Example of semantic based method detection*



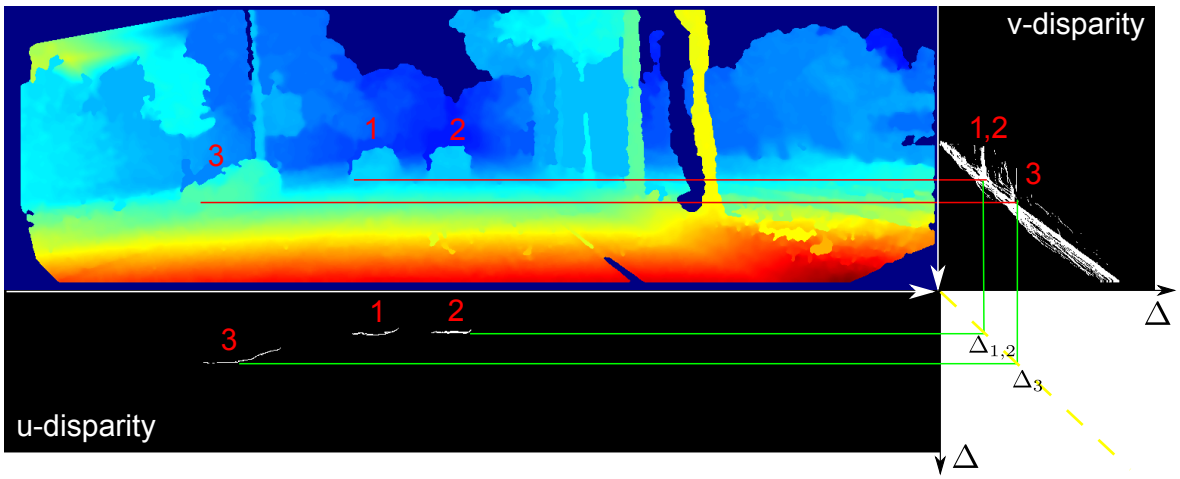
(b) *Example of motion based deteciton method*

Figure 3.2 – *Illustration of missing detection cases in semantic based detection method and motion based method. The yellow bounding boxes indicate the detection result of semantic based detection method; the green bounding box indicates the motion based detection result; the red bounding boxes in both figures are the missed detections.*

3.2.1 Obstacle localization with U-disparity image analysis

As presented in Chapter 1, the disparity map $I\Delta$ can be extracted from calibrated stereo images [TV98]. And the U-V-disparity images can be built by accumulating the pixels of same disparity in $I\Delta$ along the u, v axis separately [Pri03]. For example, the intensity value of each point $I_u\Delta(u_i, \Delta_i)$ in the U-disparity image represent the number of coherent points with disparity Δ_i along current u_i axis (or u_i axis) of stereo images. The points on an obstacle facing to the camera are approximately at the same distance to the stereo rig. They present a homogenous disparity value on the obstacle's front face. Meanwhile, the disparity value of road reduces and extends to far distance. Thus, in U-V disparity images, on-road obstacles are represented by high intensity regions apart from road profile. An example is shown in Fig. 3.3.

(a) *Original image from left view*



(b) *Disparity map and U-V-disparity image accumulated from ROI*

Figure 3.3 – *Characteristic of U-V Disparity image . This figure shows two ambiguous situations in U-V disparity image. In first situation, obstacle 1 and obstacle 2 stand at the same distance to the stereo-rig. Their representation lines in in V-disparity image are overlapped; but in U-disparity image they are represented separately by two horizontal lines. The second situation shows that when an obstacle (number 3) is passing by the stereo-rig, both its front face and side face are observed by the stereo cameras. In V-disparity image, the obstacle 3 is represented by two different lines. The vertical one refers to its front face, and the oblique one refers to its side face. In U-disparity image, obstacle 3 is represented by a connected region which is composed of an horizontal part and an oblique part.*

In several approaches [HLPA06, NSH07, LAT02], V-disparity are more used than U-disparity, since it reveals the road profile that it is the main coherent area in the 3D traffic scene. Nevertheless, for a precise obstacle detection, U-disparity image provides more accurate and useful information of the obstacles rather than the V-disparity. The V-disparity image $I_v\Delta$ is usually used to estimate the longitudinal profile of the road (See Chapter 1). It can also be used to detect the presence of obstacles by using a line extraction algorithm [LAT02] since the obstacles are represented by the vertical lines on the road profile. However, in complex scenarios, detecting lines in the V-disparity image can lead to ambiguities. For example, as illustrated in Fig. 3.3, the obstacles (obstacle 1 and obstacle 2 in the figure) at the same distance to the camera are represented by a single vertical line in the V-disparity image. In this situation, it is impossible to detect the two obstacle separately from the V-disparity image. Another problem comes when an obstacle is getting close to the camera and is about to pass by the camera. During this short time period, the side face of the obstacle is also observed by the camera, an example is shown with obstacle 3 in Fig. 3.3. Unfortunately, the side face of obstacles usually extends through different distance layers. They either cannot provide enough accumulation in each layer of the V-disparity space; either are accumulated into a curve due to the non-homogeneous disparity value of each layer. On contrary, the U-disparity image, which we note as $I_u\Delta$, preserves more information of the scene: the objects width, their relative positions and the depth information. All those observed surfaces of obstacles are projected as straight lines, and the lines indicate different obstacles are distinctively spread in the U-disparity image. These properties of U-V-disparity images are illustrated in Fig. 3.3. Especially, in cases of successive road side vertical obstacles: such as a fence or a wall along the road, they will be projected as slope lines as a barrier boundary for the road area. In our approach, the U-disparity image is used for primary obstacles detection, while the V-disparity image is employed to assist road surface extraction step.

3.2.1.1 Traffic Area construction

In many stereo vision-based obstacle detection approaches [KB12, HU05, LLKK11], all pixels on disparity images are used to detect the obstacles. As shown in Fig. 3.4, the obstacles outside the driving space are presented in the detection result as well. Nevertheless, the aim of our vision based detection and tracking system is to analyze traffic information and to help the driver with decision making. A reliable detection of ROI can not only eliminate irrelevant object outside the driving space, but also reduce the cost of computational resource, i.e. time and memory space. From this consideration, a free road surface detection combined with a convex hull operation is proposed hereafter to approximate the driving space/road area.

Figure 3.4 – *Example of obstacle detection using all the patches in the disparity map [KB12]*

In this approach, the free road area I_{final} is firstly extracted based on the method introduced in Chapter 1. However, free road surface is only partial of a complete road area. Convex hull algorithm provides the smallest convex area that contains a given subset. For instance, free road surface is the subset of the complete traffic area. The convex hull, applied on I_{final} can mend the holes and the depressions caused by on-road obstacles. Therefore, this convex area that contains the free road surface can be considered as the desired driving space i.e. the ROI which is denoted by I_{ROI}. Even if the convex hull may not exactly follows the shape of the road, in most cases, it is sufficient to provide a satisfying ROI for further detections. In some cases, Fig. 3.5b for example, when the obstacle stands on a corner of the road area, the convex hull function cannot recover this corner of road area. Thus, the obstacle can not be detected because it is "regarded" as being outside of the road. The tracking process detailed in Section 3.3 will help to cope with the default by predicting its presence from previous observation.

(a) *Example of recovered road surface from drivable area*

(b) *Recovered traffic area by taking convex hull operation on free road surface*

Figure 3.5 – *Free road surface and recovered road area where all the traffic participants stand. Top image is free road surface detected by the Algo. 1.2 in Chapter 1*

3.2.1.2 Connected-region extraction in U-disparity

In the ROI, the disparity value of free traffic area is supposed to continuously decrease as the distance increases along the direction pointing to vanishing point. On the other hand, the obstacles on the road surface present homogenous disparity value. As a result, the obstacles are accumulated to a higher intensity region in U-disparity image.

In order to extract these high intensity pixels, U-disparity image $I_u\Delta$ needs to be classified into binary image: pixels with intensity value higher than a certain threshold ε is set to 1, and others are set to 0. The value of ε depends on the camera calibration parameters, for different image sequences, the value of ε needs to be set accordingly. Generally speaking ε is the minimum possible height of an obstacle in the image. After using Eq. 3.1, high intensity regions are preserved in the U-disparity image $I_u\Delta$ and the other pixels are assigned to 0 (i.e. background).

$$I_u\Delta(\boldsymbol{p}) = \mathrm{sgn}(I_u\Delta(\boldsymbol{p}) - \varepsilon) \tag{3.1}$$

where, $\boldsymbol{p}$ refers to the pixels in the U-disparity image $I_u\Delta$.

In many approaches, the Hough Transform for line extraction is applied to get the obstacle information [HLPA06, HU05] since obstacles are usually represented as horizontal lines in the U-disparity image $I_u\Delta$. However, when an obstacle is passing near the camera side, both the front face and side face of the obstacle are observed. The obstacle is then represented by two connected lines: a quasi-horizontal one for front face and an oblique one for its side face. This situation is illustrated in Fig. 3.3, for the obstacle 3. In this case, single line detection is not able to detect these two lines as a whole. To deal with this problem, a connected-component extraction algorithm is employed in our approach to replace the classical Hough line detection.

The advantage of connected-component extraction is that it has no constraint on the form of obstacle representations in U-disparity image $I_u\Delta$. It could be two connected lines like demonstrated in Fig. 3.3, or even a curve if the obstacle is of complex shape. As long as the object shows a homogeneous or continuous disparity value, its accumulation in U-disparity image $I_u\Delta$ should be a connected component with a high intensity. Thus, even in complex situations as mentioned before, the obstacles can be extracted completely. In addition, the application of connected-component extraction is much easier than the Hough transform.

After applying Eq. 3.1, noisy pixels are removed from $I_u\Delta$ by erosion and clean morphological operations. Each connected-region $\mathbf{L}$ being preserved in $I_u\Delta$ indicates a potential obstacle O_L. It provides the following information of corresponding obstacle: Left bound u_l and right bound u_r of O_L on the u-axis of the image; and its disparity value $\widetilde{\Delta}_O$:

$$u_l = \min\{u_L\}$$

$$u_r = \max\{u_L\}$$

$$\widetilde{\Delta}_O = \min\{\Delta_L\}$$

where, $\{u_L\}$ and $\{\Delta_L\}$ are the set of pixel positions along u-axis and Δ-axis in the connected-region L.

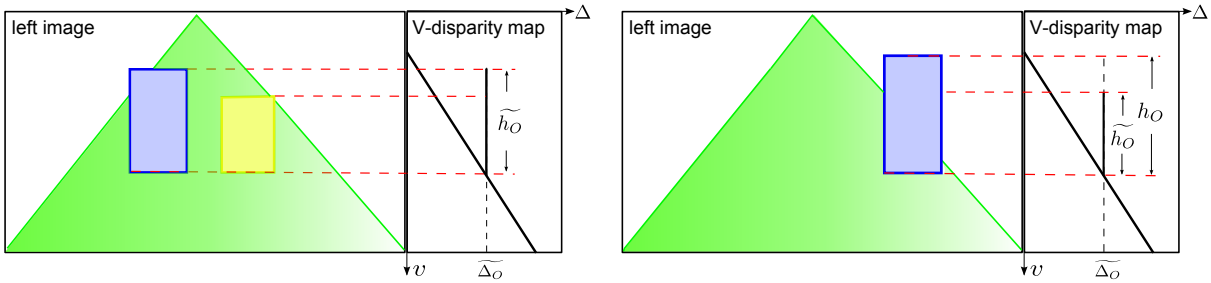
The other information of obstacle O_L like the potential height $\widetilde{h}_O$ and its potential bottom position on the v-axis $\widetilde{v}_b$ can be extracted from the V-disparity image $I_v\Delta$. As presented before, obstacle detection from the V-disparity image is not reliable, hence these two parameters are furthermore refined by sub-image of disparity map (see Section 3.2.2). As a conclusion, after this processing, we get the approximate position $[u_l, u_r, \widetilde{v}_b, \widetilde{h}_O]$ and disparity value $\widetilde{\Delta}_O$ of potential obstacle O_L.

3.2.2 Refinement with sub-image of disparity map

The U-disparity image $I_u\Delta$ provides the location of obstacles on the road surface, but the height of obstacles are still observed from V-disparity image $I_v\Delta$:

$$\widetilde{h}_O = \max(v_L) - \min(v_L)$$

where, $\{v_L\}$ are the set of v-axis position of pixels whose disparity value equals to $\widetilde{\Delta}_O$. However, the height information is not reliable especially in two ambiguous situations. The first one is that different obstacles stand at the same distance to the camera. As demonstrated in Fig. 3.3, the representations of the two obstacles at the same distance in V-disparity image $I_v\Delta$ overlap in a single vertical line. In this situation, it is impossible to tell the height of the shorter obstacle (illustration is shown in Fig. 3.6). Another ambiguous situation appears when a obstacle stands close to the road border, top of the obstacle surpasses the road area. Hence, the pixels of obstacle which stand out side the road area are not accumulated in V-disparity image. In this case, the height $\widetilde{h}_O$ from primary detection is not the real height of the obstacle.



(a) *Two obstacles at the same distance* **(b)** *The obstacle close to the road side*

Figure 3.6 – *Illustration of height ambiguities from primary detection. The green area is the road surface. (a) For two obstacles at the same distance: the real height of obstacle in yellow is impossible to be retrieved from the V-disparity image (b) For obstacle standing by the side of road: in perspective image, the top of obstacle is not in the convex of road area, thus the estimated height is in fact smaller than its real height*

In fact, because of the noise, the location of each obstacle extracted from U-V-disparity images is not accurate as well. In order to refine the location of a potential obstacle O_L and acquire its real height h_O, a sub-image of disparity map $I_O\Delta$ is extracted from the complete disparity map $I\Delta$ according to the obstacle's primary location $[u_l, u_r, \widetilde{v}_b, \widetilde{h}_O]$ in the image. To avoid the second situation of ambiguity, the height of sub-image is set to two times of $\widetilde{h}_O$. Thus we can get a bounding box for each potential obstacle. As illustrated in Fig. 3.7, the black bounding boxes are where the sub-images are located. Then, a binary classifier is applied on each sub-image of the disparity map: pixels with a disparity value close to $\widetilde{\Delta}_O$ are considered belonging to the correspondent obstacle.

$$\begin{cases} I_O(\boldsymbol{p}) = 1 \text{ object}, & \text{if } I_O\Delta(\boldsymbol{p}) \in [\Delta_1, \Delta_2] \\ I_O(\boldsymbol{p}) = 0 \text{ background}, & \text{otherwise} \end{cases} \tag{3.2}$$

where, I_O is the binary image with labeled obstacles. $\Delta_{1,2} = \widetilde{\Delta}_O \pm \sigma$ where, σ is the bias of possible disparity value of the same obstacle. Obstacle's position and size information is then refined in I_O (the red bounding boxes in Fig. 3.7).

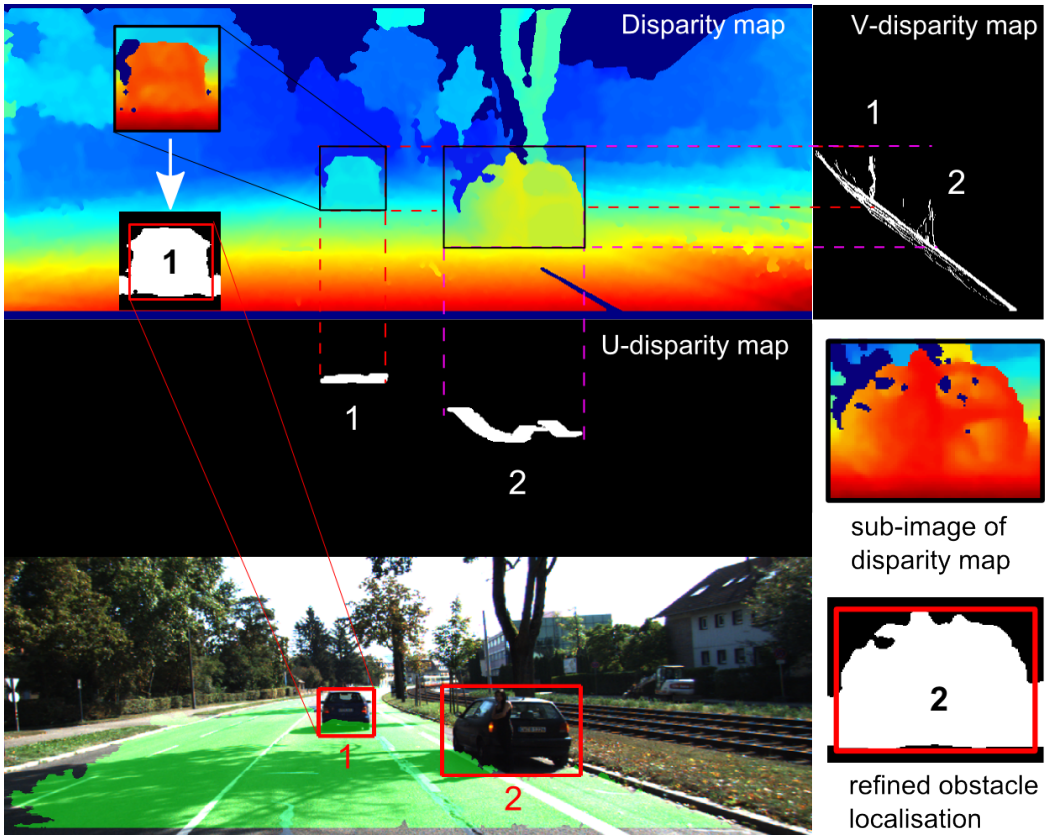


Figure 3.7 – *Sub-image of disparity map for refinement of obstacle's location. The green area in final detection result is the road surface. Black bounding boxes are the primary detected location for each obstacle based on U-V disparity image. Sub-images of disparity map $I_O\Delta$ are extracted from these locations and then classified into the binary images I_O. The red bounding box are the refined obstacle locations from I_O.*

The obstacles' location are modeled by their centroid (x_O, y_O), their width w_O, their height h_O and their disparity Δ_O. These values are therefore refined by the binary image I_O. Compare to region growing algorithms, our proposed extraction method is much faster and more effective. An illustration of the use of sub-image of disparity map is showed in Fig. 3.7.

3.2.3 Additional object detection criteria

Because the disparity map is noisy, it can lead to false alarms and fragments of an obstacle during the detection. Thus, multiple criteria are introduced to improve the detection results:

- Combination of closely stand small connected-regions: Since the U-disparity image is accumulated on discrete values from the disparity map, there could be fragments of the same obstacle using the representation of connected-regions. This will lead to redundant detections. To handle this problem, a combination operation, like bridge and dilation morphological operations are applied.

- Height limitation of potential obstacles: A small height value could be caused by deceleration strip or non-planar region of the road, but mostly it is caused by the noise in disparity map. In both cases, these "potential obstacles" are not our concern. Hence, we set a threshold ς_h: potential obstacle with a height smaller than ς_h will be eliminated from final detection result. The threshold ς_h is proportional to the disparity value of this potential obstacle $\varsigma_h \propto \Delta_O$. The closer potential obstacle stands to the camera, the higher threshold ς_h will be.

- Width limitation of potential obstacles: In some cases, man-build structures along the road like walls may show similar texture with the road surface, thus the ROI for obstacle detection is enlarged, and these man-build structures are therefore detected as on-road obstacles. To avoid such kind of false alarms, a comparison of potential obstacles' width with a threshold ς_w is necessary. Same as height limitation, this threshold is positive correlated to the disparity value of the obstacle: $\varsigma_w \propto \Delta_O$. Any potential obstacle that has a width larger than this threshold is eliminated from final detection result.

Fig. 3.8 gives an example of the height limitation criteria. It shows the difference between false alarm and a real obstacle. The left bounding box is related to a false alarm, while the right bounding box corresponds to a real on-road obstacle. As we can see, in the sub-image of disparity map which belongs to the false alarm, there is no

obvious obstacle, thus the height of this "potential obstacle" is very small. Compared to the threshold ς_h, the false alarm is eliminated from detection result.

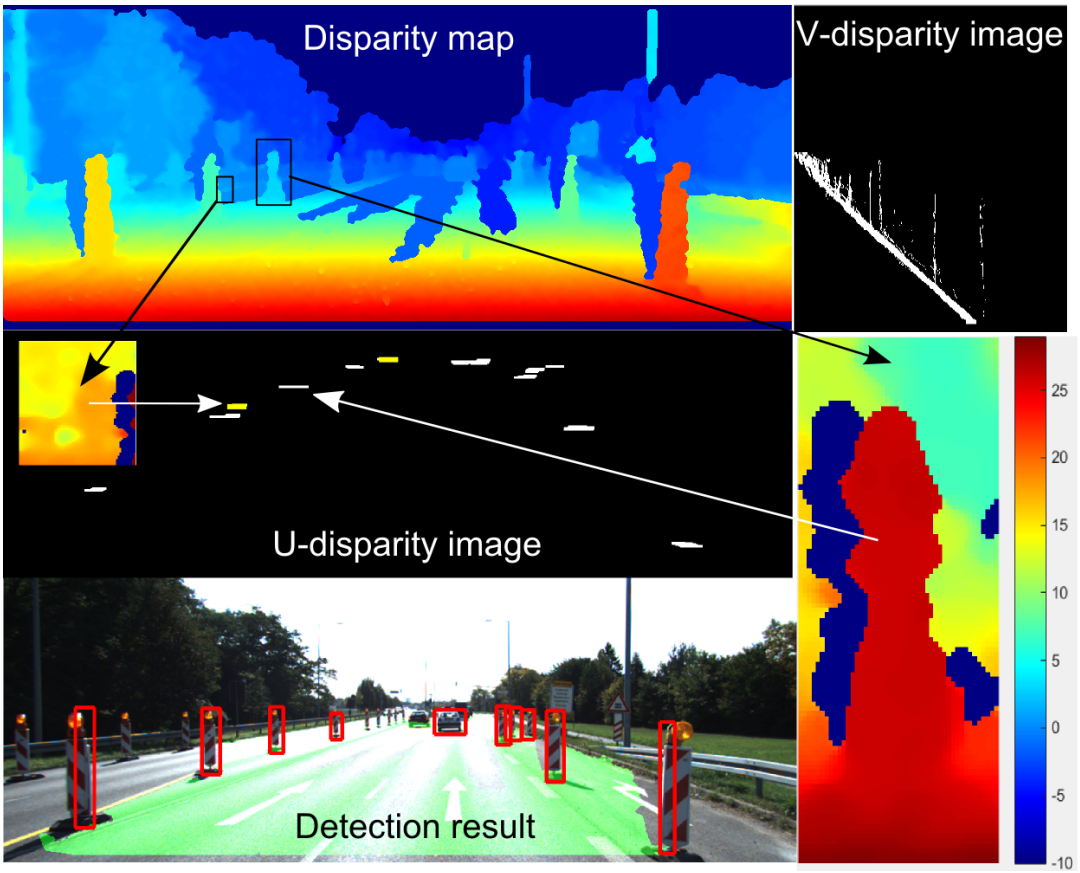


Figure 3.8 – *Example of false alarms eliminated by additional detection criteria: The yellow lines in U-disparity image are the false alarms which are eliminated by the height limitation criteria later on. The potential obstacle is extracted from corresponding sub-image of disparity map. The two sub-images of disparity map indicate a false alarm and a real obstacle separately. The deep blue in the sub-image of disparity map is where the disparity value cannot be correctly estimated.*

The complete on-road obstacles detection pipeline is summarized in Algo. 3.1.

3.3 Multiple Target Tracking using Dynamic Particle Filter

After detecting the on-road obstacles in the image, a tracking process is employed in our work to follow their trajectories and predict their future behaviors. The aim of the multiple-target tracking (MTT) is to find tracks of multiple targets from the noisy measurements. It is a challenging task because it involves a lot of unknowns: The measurement noise; the varying number of targets; the unknown association between current

Algorithm 3.1 On-road Obstacle Detection Algorithm

Input: - Stereo color images $I_l,\ I_r$
Output: -Number of detected obstacles N_{obs}
- Position, size and disparity value of obstacles $O_{1...N_{obs}} = [x_O, y_O, w_O, h_O, \Delta_O]$

for k =first frame **to** lastframe **do** ▷ Evolution of frames
  ▶ Calculate disparity map $I_\Delta \leftarrow (I_l,\ I_r)$
  ▶ Calculate free road surface I_{final} by Algo.1.2;
  ▶ Convex hull construction: $I_{ROI} \leftarrow I_{final}$;
  ▶ U-V-disparity image on ROI: $[I_u\Delta, I_v\Delta] \leftarrow (I_{ROI},\ I_\Delta)$:
  ▶ Label the N connected-regions $L_{1,...,N}$ in $I_u\Delta$:
  for $i = 1$ **to** N **do** ▷ Location extraction
    ▶ Extract primary position and disparity value:
    $[u_l, u_r, \widetilde{v}_b, \widetilde{h}_O, \widetilde{\Delta}_O] \leftarrow (L_i, I_u\Delta, I_v\Delta)$
    ▶ Generate sub-images of disparity map for each object O_i:
    $I_O\Delta \leftarrow (u_l, u_r, \widetilde{v}_b, \widetilde{h}_O)$
    ▶ Extract obstacle from sub-images of disparity map to a binary map (3.2):
    $I_O \leftarrow I_O\Delta$
    ▶ Refine obstacle position from the binary map
    $[x_O, y_O, w_O, h_O, \Delta_O] \leftarrow I_O$,
    if $h_O \geq \delta(\Delta_O)$ **then** ▷ Eliminate false alarm
      ▶ $N_{obs} \leftarrow N_{obs} + 1$;
      ▶ $O_{N_{obs}} = [x_O, y_O, w_O, h_O, \Delta_O]$
    end if
  end for
end for

targets and existing tracks. In tracking-by-detection approaches, the false alarms and missing detections also increase the difficulty of MTT task. In order to cope with these difficulties, a wide range of MTT approaches [BRL+09, ESLVG10, JDSW12, KBD05, KS09] relies on the recursive update of tracks according to the latest detections. For instance, Kalman filtering [WKT06, RAG04] is an efficient way to address multi-target tracking when the number of objects remains small [Mag04, FMP+13]. It predicts and updates the states of trackers linearly with assumption of uni-modal (Gaussian) distribution of the target state. The main advantage of Kalman filtering is that it is computationally efficient which is suitable for real-time applications. However, in reality most target states are often non-Gaussian. Particle filtering [SGPO05, MDB08, NTMM12] can overcome the limitation of Kalman filtering by representing the state probability density with a set of weighted particles. The weight represents the probability of a particle being sampled from the probability density function. This representation supports for multi-modal (non-Gaussian) distributions, therefore it is more feasible to capture and follow target trajectories. In the same spirit, [YMC07] relies on Markov chain Monte Carlo (MCMC) to recover trajectories of targets using a batch of observations. [MTC08] applies a Probability Hypothesis Density filter to track multiple objects from noisy observations. In an attempt to increase tracking reliability, some hybrid approaches have been proposed. For example, [EM08] uses a motion model and

nearest neighbor data association algorithm to build tracks out of people detected from scene captured by a calibrated camera. The tracks thus generated are then merged and split into the final trajectories using heuristics based on overlap criterion, directions and speed. However, hybrid methods do not guarantee convergence to a global optimum because of their ad-hoc strategies. To improve robustness to wrong identity assignment, research has recently focused on linking detections over a larger time window using various optimization schemes. For example, [KS09] applies graph cuts to extract trajectories from a batch of people detections obtained using homographic constraints over a window of frames. Unfortunately, the computational complexity of such an approach can be prohibitive. In a driving scenery, especially in highway, every appearance of obstacles last only for a short period, which need an instantaneous detection and tracking algorithm for each frame. If the tracking is performed in the image plane, different depths lead to different projective scale of the obstacle. This problem must be handled carefully in the recursive filtering.

In this section, we present a modified particle filter for visual tracking, because particle filtering is easy to implement and it is robust with non-Gaussian object states. The tracking is performed in the image plane of the left camera in the stereo vision system. The on-road obstacles are regarded as the targets; their detected position and size in 2D image plane are the measurement. Target-to-track association is carried out following a global nearest neighbor (GNN) criterion. To cope with the scale variations caused by projective distortion, the observed obstacle dynamics are employed to define an adaptive association gate. A dynamic noise generation function is then employed in the filter for prediction step. For each iteration of the filtering, multiple hypotheses are made to create, delete and update the existing tracks.

3.3.1 Fundamentals of particle filtering

The Condensation algorithm [IB98], as a special case of particle filtering, has been widely used for visual tracking. It provides a well-established methodology for generating samples from the required distribution without requiring assumptions about the state-space model or the state distributions. The state-space model can be non-linear and the initial state and noise distributions can take any form. At each time step, particles' weight and spatial distribution are used for state prediction. Every time a new observation is perceived, the weights of particles are re-distributed according to the distance between prediction and observation. Then a new set of new particles are re-sampled from current sample set according to the weights.

– Observation:

The observation $Z(k)$, also noted as measurement, is the detection result of target objects at frame k in the framework of tracking by detection. It can be represented

by the position of the targets, the shape of the targets, or the texture features of the targets.

– Estimation:

$$S(k) = \sum_{i=1}^{N_s} s^i(k) \cdot \pi^i(k) \tag{3.3}$$

where, $S(k)$ is the condensation state of the tracker, N_s is the number of samples (particles in a sample set). $s^i(k)$ is current sample states, where, $i = 1, \ldots, N_s$, and k represents the time step. The variable $\pi^i(k)$ is the normalized weight distributed for each sample by the posterior $P(s^i(k) \mid Z(k))$.

– Prediction:

$$s^i(k+1 \mid k) = \boldsymbol{f}(s^i(k)) + W(k) \tag{3.4}$$

where, $s^i(k)$ is the particle states of current frame k and $s^i(k+1 \mid k)$ is the predicted particle states for next frame $k+1$. $\boldsymbol{f}(\cdot)$ is the evolution function of dynamic model for . $W(k)$ is the state noise vector.

– Update and re-sampling:

When a new observation $Z(k+1)$ arrives, the update of the particle states depends on the estimation of their probability density $\mathcal{P}(s^i(k+1) \mid Z(k+1))$. According to Bayes rule, it can be calculated by Eq. 3.5.

$$\mathcal{P}(s^i(k+1) \mid Z(k+1)) \propto \mathcal{P}(Z(k) \mid s^i(k+1)) \cdot \mathcal{P}(s^i(k+1)) \tag{3.5}$$

where, $\mathcal{P}(Z(k+1) \mid s^i(k+1))$ is the likelihood of observation $Z(k+1)$ given the particle state $s^i(k+1)$. $\mathcal{P}(s^i(k+1))$ is a prior probability density of particle state $s^i(k+1)$ occurring.

In reality, the posterior distribution $\mathcal{P}(s^i(k) \mid Z(k))$ may be difficult to compute in closed form. An alternative solution is to represent $\mathcal{P}(s^i(k) \mid Z(k))$ using Monte Carlo samples: each particle is attributed with a state s^i and a weight π^i. The weight which can be considered as probability $\mathcal{P}(s^i(k))$ as illustrated in Eq.3.6. It indicates how samples will be drawn from previous sample set. High probability samples are drawn more frequently. Low probability samples are drawn less frequently. The drawn samples $s^i(k)$ thus follow a distribution that approximates $\mathcal{P}(s^i(k) \mid Z(k))$. This process is called re-sampling. To be noted, the more particles, the better the approximation.

$$\pi^i(k+1) = \frac{\mathcal{P}(s^i(k+1 \mid k) \mid Z(k+1))}{\sum_{i=1}^{N} \mathcal{P}(s^i(k+1 \mid k) \mid Z(k+1))} \tag{3.6}$$

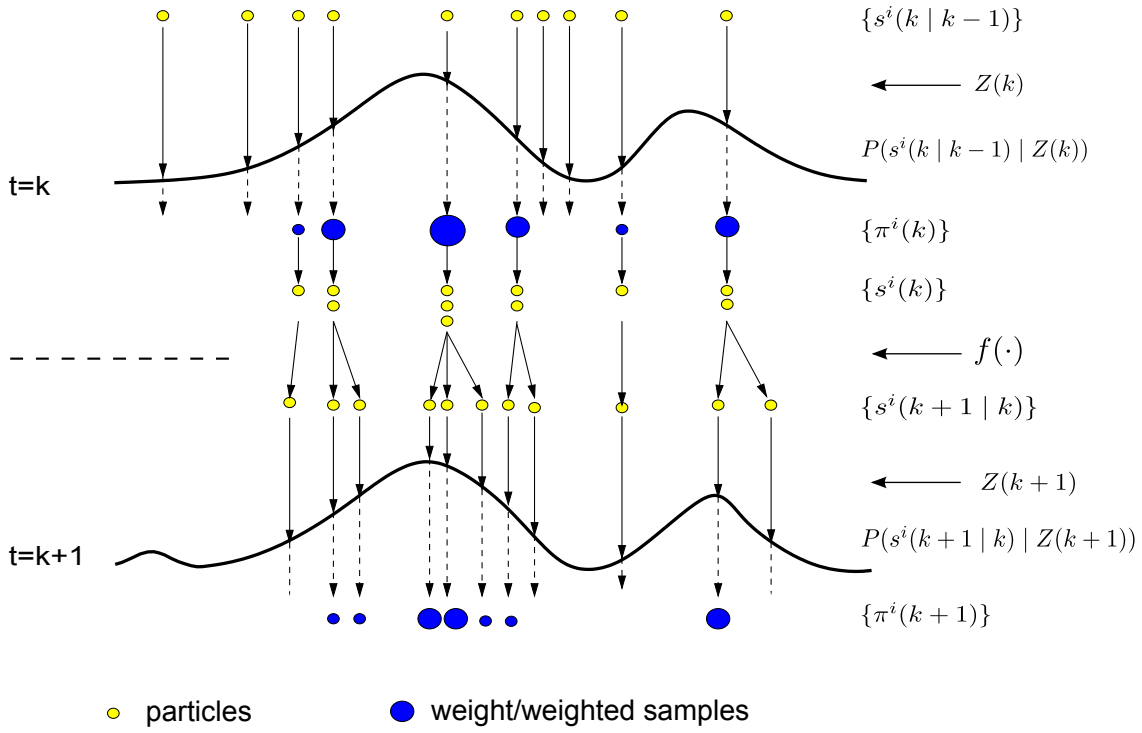


Figure 3.9 – *Illustration of condensation algorithm*

For each evolution, particles $s^i(k+1 \mid k)$ are predicted from their previous state $s^i(k)$ by Eq. 3.9. Then, the confidence density $P(s^i(k+1 \mid k) \mid Z(k+1))$ is distributed to $s^i(k+1 \mid k)$ through the comparison between particle states and their associated observations $Z(k+1)$. The normalized weight distributed for each sample is then updated from the confidence density by Eq. 3.6. Subsequently, a new set of N_s particles $s^i(k)$ are constituted from the current sample set $\{s^i(k+1|k)\}$ with probability proportional to the confidence distribution [IB98].

3.3.2 Track State and Evolution Model

In order to define a uniform state space, all the states are described in the pixel level in our approach. The depth information is then represented by its corresponding disparity value. Our approach automatically initializes a separate particle filter for each detected obstacle. The filter model is defined as follows:

– State vector:

$$S = [x, y, v_x, v_y, w, h, \Delta]^T \tag{3.7}$$

It is composed by the centroid of the object position (x, y) on the image; the velocity of the centroid v_x, v_y ; the width w; the height h and the disparity value Δ respectively.

– Observation:

$$Z = [x_O, y_O, v_{xO}, v_{yO}, w_O, h_O, \Delta_O]^T \tag{3.8}$$

The values x_O, y_O, w_O, h_O, Δ_O represent the information of detection result of Algorithm 3.1. Since the velocity v_{xO} and v_{yO} of the object cannot be measured directly, it has been initialized to a 0 value. During the detection and tracking processing, the velocity is calculated by the displacement of obstacle's centroid between two successive frames.

– Initialization:

N_s particles s^i with $i = 1 \dots N_S$ are generated during the creation of each tracker. Within a generation range, the particles are initialized by following a Gaussian distribution of each newly detected measurement.

– Evolution model:

In this approach, f is a constant velocity model. Thus, Eq. 3.4 can be written as :

$$s^i(k) = [x^i(k), y^i(k), v_x^i(k), v_y^i(k), w^i(k), h^i(k), \Delta^i(k)] \quad i = 1 \dots N_S \tag{3.9}$$

with:

$$\begin{cases} x^i(k+1 \mid k) = x^i(k) + T \cdot v_x(k) + W_x(k) \\ y^i(k+1 \mid k) = y^i(k) + T \cdot v_y(k) + W_y(k) \\ v_x^i(k+1 \mid k) = v_x^i(k) + W_{v_x}(k) \\ v_y^i(k+1 \mid k) = v_y^i(k) + W_{v_y}(k) \\ w^i(k+1 \mid k) = w^i(k) + W_w(k) \\ h^i(k+1 \mid k) = h^i(k) + W_h(k) \\ \Delta^i(k+1 \mid k) = \Delta^i(k) + W_\Delta(k) \end{cases} \tag{3.10}$$

$$W(k) = [W_x(k), W_y(k), W_{v_x}(k), W_{v_y}(k), W_w(k), W_h(k), W_\Delta(k)]^T \tag{3.11}$$

where, T is the elapsed time between two successive frames. In our approach, the evolution function (Eq. 3.10) is a linear constant velocity model. However, in 2D image coordinates, if an object is closer to the camera, it appears larger, and a lateral movement of the same velocity in world coordinates yields a larger velocity of pixel displacement in image plane. To handle the depth variation on the displacement and the scale of obstacle, a dynamic noise vector is introduced in Section 3.3.3. Thus the growth/reduce of relevant elements in state vector can be compensated by the noise.

During the test, the constant velocity model appears more suitable than constant acceleration model for this problem. Because the vibration of the host vehicle can lead to an irregular displacement of the observation, in the constant acceleration model, a small bias of the position can be magnified by acceleration and finally lead to unstable tracking result.

3.3.3 Data association

It is important to maintain the identity of multiple targets while tracking them, thus the data association between targets and tracks must be carefully handled. There data association techniques for MTT system range from the simplest Nearest Neighbor (NN) algorithm to the very complex multiple hypothesis tracker (MHT). The NN algorithm computes association distance from each track to all observed measurements. For each round of association, the minimum distance between unpaired tracks and targets is found to associate the closest track and target. This algorithm is efficient in less cluttered scenes but it can lead to local minima. The MHT method forms alternative association hypotheses by taking into account all the possible assignments. The hypotheses are propagated in the future until more data received for a decision. MHT provides promising performance, but it is difficult to implement and it requires extensive computational resources.

Based on the application context, we apply a Global Nearest Neighbor (GNN) method with specified parameter settings to associate tracks with measurement. Different from NN, GNN finds the optimal track-to-target assignment by minimizes the summed total association distance. Recently the increased computational power of the computers allows using this approach in real time implementations.

In concrete application, for M targets and N tracks, an $M \times N$ distance matrix is built. The Mahalanobis distance between each target and each track is mostly used to fill the matrix. The data association process begins with a gating test which preserves the possible target-to-track pairs in the matrix if their distance is smaller than the given association value. The assignment is then decided by global nearest neighbor algorithm. However, not all the tracks or obstacles can be associated, if the distance is larger the gate, association will not be established. In such cases, new hypotheses need to be made. This situation will be discussed in the end of this section.

3.3.3.1 Noise modeling

The projective scale problem includes both the scale of obstacle's displacement and size in image plane. From this consideration, the noise vector added in Eq. 3.10 should be linked to the change of disparity value (since the disparity value is a representation

of depth in stereo vision), i.e. $W(k) \propto \Delta_O(k)$. Therefore, the tracker should be able to keep following the observation state in 2D image coordinates. At each step, the dynamic noise update function is designed as follows:

$$W(k) = c \cdot \Delta_O(k) \cdot Z(k) \tag{3.12}$$

where, c is the regularization coefficient that needs to be adjusted given the application. The value of the coefficients for the position and size are set bigger than coefficients for speed, because the observation of the velocity depends on the relative motion of the obstacles and the movement of the camera. For a given camera equipment, these parameters can be set for once, because the scale variations lead by depth are related to the camera's intrinsic matrix. But the initial values of the noise vector $W(0)$ must be assigned with big values to provide a broader range for sample initiation. In the very beginning, the velocity of obstacles cannot be measured, thus its initial state is set to 0. A big initial noise vector allows the samples to track the obstacles that moves fast. Once the velocity can be measured, noise vector $W(k)$ can be dynamically adapted to smaller values to ensure the convergence of particle predictions.

3.3.3.2 Association criterion

Every time a target state is observed, the measurements are firstly compared to filter predictions by a statistical distance. In our approach, association distances between each observation and prediction are calculated by:

$$d_{TT} = c_1 \mid \nabla_{x,y} \mid + c_2 \mid \nabla_{w,h} \mid + c_3 \mid \nabla_{\Delta} \mid \tag{3.13}$$

where, $c_{1,2,3}$ are the normalized weights for different measurements which indicate their contribution to the distance criterion. $(\nabla_x, \nabla_y, \nabla_w, \nabla_h, \nabla_\Delta)$ is the difference between estimation and observation. Thus, the obstacle's centroid (x, y) in the image is not the only criterion that contributes to the distance calculation, it works with the width and height (w, h) and the disparity value $\Delta$are . Thus, the mis-association situations can be reduced with multiple measurements.

3.3.3.3 Self-adaptive gate

Target-to-track association is usually simplified by using of a gate. This gate is usually set to a constant value for eliminating unlikely observation-to-track pairing that has an association distance beyond the value. However, for the object tracking performed in the image plane, the measurements of obstacle at near distance contain bigger noise on both scale and displacement. If the gate is set too small, these obstacles may not be able to pass the gating test and to be correctly associated with their tracks. A constant

gate cannot cope with obstacles at different distances. A self-adaptive gate is therefore defined according to the scale and depth information of each observed obstacle:

$$G_O = a \cdot \mid (w_O(k),\ h_O(k)) \mid + b \mid \Delta_O(k) \mid \tag{3.14}$$

where, G_O is the adaptive gate for each obstacle according to their observations at time k. It is only related to the current observation's scale and depth information. This design is able to cope with to projective distortion of the obstacles in image plane and the measurement noise.

3.3.3.4 Multiple hypotheses

Obstacles and tracks are associated by global minimal distance within the gate. If there is no association established, it leads to two possible situations: non-associated obstacle or non-associated track. They are usually related to the difficulty named varying number of obstacles from observation. In the first case, it is assumed that a new obstacle is newly detected, and a new track needs to be created for this obstacle. In the second case, non-associated track will be preserved and updated for a short time period until the tracking failed up to a time delay threshold. In that case, only the tracks that have failed over a certain time period (denoted as T_{ass}) are removed. This allows the tracker to recover from instant detection failure or incorrect association. If the non-associated track is caused by a missing detection or occlusion during the observation, once the obstacle is observed again, the track could still be paired with the obstacle. In that case, the track stays continuous and mends the gap of missing detection.

The on-road obstacle tracking algorithm

The complete algorithm of the modified particle filtering for on-road obstacle tracking is summarized as in Algo. 3.2.

3.4 Experiments

The detection and tracking algorithm in this chapter are evaluated on two benchmark datasets from the KITTI benchmark suite [GLSU13]. We considered both urban road and structured highway road for the experiments. The tracklet labels of the datasets are used as "ground truth" for the comparison and evaluation.

– Dataset 1: urban road, KITTI raw data, 2011_09_26_drive_0056

– Dataset 2: high way, KITTI raw data, 2011_09_26_drive_0032

Algorithm 3.2 The MTT algorithm using Modified Particle Filter

1: ▶ Initialization: $k = 0$, generate a sample set $\{s^i(\tau, 0)\}$ for each obstacle
where, $i = 1, \ldots, N_s$, $\tau = 1, \ldots, N_{obs}(0)$.
N_s is the number of samples
$N_{obs}(0)$ is the number of detected obstacles at initial time
2: ▶ Draw samples $\{s^i(\tau, k)\}$ from Gaussian distribution around observation $Z(\tau, k)$ with initial noise range $W(0)$
3: **for** k =first frame **to** last frame **do** ▷ Frame evolution
4: **for** $\tau = 1$ **to** $N_{obs}(k)$ **do** ▷ Tracking of each obstacle
5: ▶ Compute adaptive gate $G_O(\tau, k)$ by Eq. 3.14 for data association
6: ▶ Associate tracker with obstacle by GNN algorithm [FF84]
7: ▶ Update the particles' weights of $\pi^i(\tau, k)$ by Eq. 3.6
8: ▶ Re-sampling of particles $s^i(\tau, k)$ from current sample set according to $\pi^i(\tau, k)$
9: ▶ Estimate the tracker state by Eq. 3.3
10: ▶ Predict the state of particles $s^i(\tau, k + 1)$ by Eq. 3.10
11: **if** Non-associated Obstacle **then** ▷ New obstacle
12: ▶ Generate new tracker sample set $\{s^i(\tau, k)\}$ for the obstacle ⟶Step 2
13: **end if**
14: **end for**
15: **for** Non-associated tracker **do** ▷ Obstacle left the scene
16: **if** the tracker has not been associated for a period T_{ass} **then**
17: ▶ Prune the tracker.
18: **end if**
19: **end for**
20: **end for**

The algorithm is implemented in a standard PC with Windows 7 Enterprise OS, Intel CPU of 2.66 GHz. The development environment is MATLAB R2013b. Disparity map are obtained from LIBELAS toolbox [GRU11]; it is a cross-platform (Linux, Windows) C++ library with MATLAB wrappers for computing disparity maps from rectified grayscale stereo images. The particle filer functions from OpenCV [BK08] are integrated in the code. The run-time is 3.4s per frame for on-road detection processing and 0.05s per frame for multiple obstacles tracking algorithm. The detection distance in disparity map is limited to 35m in front of the camera (i.e. stereo camera field of view). The detection and tracking algorithm are performed in the image of the left camera of the stereo vision system.

3.4.1 Evaluation method design

The 3D tracklet labels from the KITTI dataset provide the position and the motion history of the obstacles that appeared in the scene. After projection onto the image plane, the tracklet 2D position could be seen as ground truth trajectories for object detection and tracking. However, there are some considerations that need to be made

to evaluate our detection and tracking result based on the tracklet labels. First, tracklet only contains certain categories of objects, such as car, van and pedestrians, other types of obstacles are not included. For example, in Dataset 2, traffic cones are not listed in the tracklet, they still should be detected as on-road obstacles. Second, the tracklet labels also provide off road information of obstacles which are beyond the traffic area considered in our approach. Third, our stereo vision based detection distance is set up to 35m, while the tracklet reaches to 70m. Thus, the evaluation is constrained to the intersection between our detection results and the labeled tracklets. To establish a fair evaluation platform, we chose a sequence of 100 frames from Dataset 1 and Dataset 2 respectively. Then, the GNN association is applied to pair our experimental results with tracklets. The result of this processing provides the intersection of our experimental result with tracklet labels. Hence, on-road tracklets are picked out by data association. From the tracklets label list, all the obstacles that present within 35m distance to the camera are considered as ground truth. To ensure the integrity of evaluation results, some special cases like false alarms, missed detection and redundant detections are also examined and noted manually during the experiment of detection and tracking.

3.4.2 On-road obstacle detection results

Parameter settings

In the test with the KITTI dataset [GLSU13], the accumulating threshold ε to generate binary U-disparity image (see Eq. 3.1) is about 8~15 accumulated pixels for an obstacle with a height around 0.5m within the detection range of 35m in front of the camera.

Results evaluation

Dataset 1 contains 150 detections on road, 136 of them are associated with trackelet. Dataset 2 contains 363 detections on road, 48 of them are associated with tracklets. In Dataset 2, except for false alarms and redundant detections, 309 non-associated detections are traffic cones standing on the road and they are not listed in the tracklet.

The associated results are evaluated following 5 indicators stated in Table 2.2: false alarms, missed detections, redundant detections and average error (AE) of position and size. According to the experimental results, most of false alarms in Dataset 1 are caused by the obstacles standing beside the road edge, e.g. trees. In Dataset 2, there are no obstacles besides the road edge, so false alarms rarely occurred. Most of missed

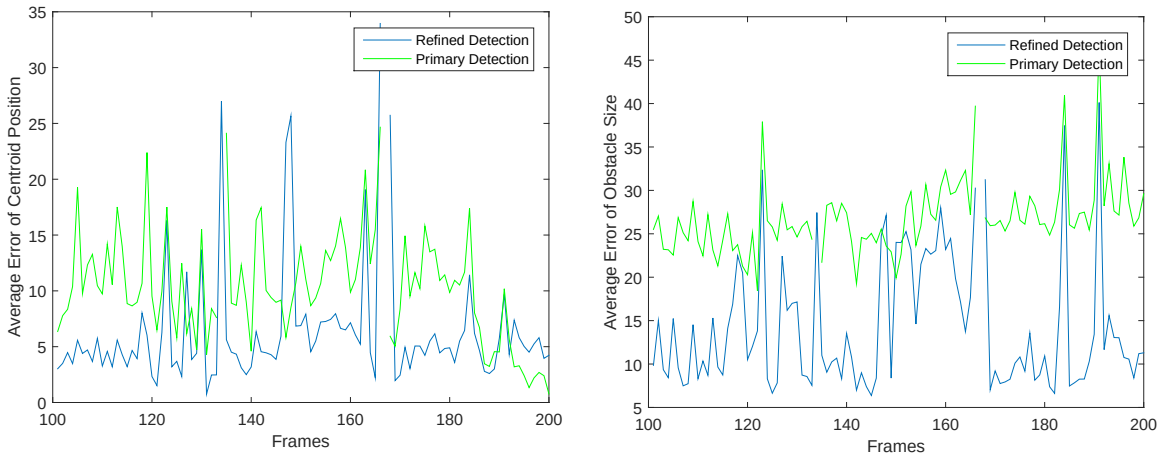
Figure 3.10 – *Obstacle detection results with stereo vision in difference scenarios: left column highway, right column urban road*

detections in our experiment appear in the two sides of the image at the bottom. They are usually induced by the errors of the disparity map. Imprecise disparity map values can also lead to redundant obstacle detections. The average error (AE) illustrates the average distance between detected obstacle centroid and labeled tracklet centroid (ground truth). In addition, the average error that measures the variation of size scale is also listed in Table 2.2. The two indicators are measured on pixel-level of the images. The smaller the AE, the better the accuracy of the track is. To be noticed, considering our method does not contain obstacle recognition processing, when two obstacles are close to each other they are more likely to be detected as one.

Measures	false alarms	missed detections	redundant detections	average error of centroid	average error of size
Dataset 1	6.0%	4.0%	3.3%	5.3 px	8.7 px
Dataset 2	1.1%	3.3%	1.3%	6.8 px	12.4 px

Table 3.1 – *Evaluation of the on-road detection results*

We also compared the detection results without using sub-image disparity for refinement. As illustrated in Fig. 3.11, the refined obstacle detection using sub-image of disparity map provides more accurate results on both the measurements of obstacle centroid position and the measurement of the size of obstacle.



(a) *The average error on centroid position* **(b)** *Average Error on obstacle's size*

Figure 3.11 – *Comparison of the detection results on Dataset 1 with/without refinement using sub-image of disparity map: green lines are primary detection result without refinement, blue lines are refined detection result*

3.4.3 Multiple target tracking results

Parameter settings

In our experiment of multiple target tracking, some parameters in the modified particle filter need to be defined.

The weights of different measurement for calculating the target-track association distance d_{TT} (see Eq. 3.13) are assigned as $c_1 = 0.5$ for the displacement of the obstacle, $c_2 = 0.3$ for the size difference between obstacle and track, $c_3 = 0.2$ for the depth difference between obstacle and track.

The regularization coefficient c for dynamic noise generation is set to 0.1. This coefficient functions together with the disparity value of the track (see Eq. 3.12). They effect on the random generation range of noise vector in evolution function. Since the range of disparity value varies from 0 to 127, a regularization coefficient can avoid over-sparse noise distribution.

Finally, we set the coefficients of self-adaptive gating in Eq. 3.14 as: $a = 0.5,\ b = 0.2$, which is basically the radius of the circumscribed circle of obstacle plus tolerance range defined as 20% percentage of the disparity value.

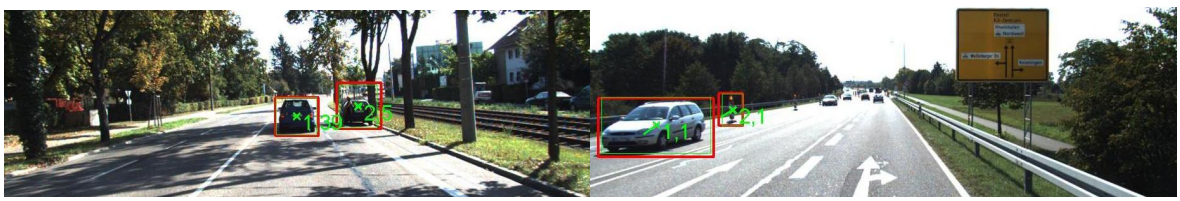
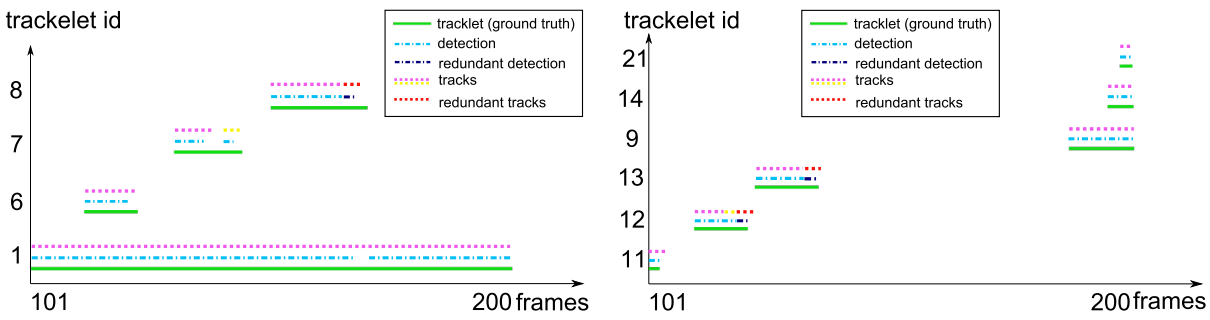


Figure 3.12 – *Examples of obstacle tracking results with stereo vision*

Result evaluation

Multiple-target tracking can not only record and predict the motion of the obstacles, but can also deal with occasionally missed detections or occlusions. As shown in Fig. 3.13a, for the obstacle with tracklet number 1, there is one frame with missed detection, while the track remains complete by filling blanks with predictions. During the tracking process, four metrics are evaluated: rate of track fragmentation, rate of overlap, the average precision of tracker's position and size at each time step. In Dataset 1, the tracking result is satisfying, with only 2.36% track fragmentation during the sequence and an average error of 5.5 pixels from the ground truth centroid. But its rate of tracks overlap is lower than Dataset 2. This is because in Dataset 1, the road is not horizontal. Thus, the disparity distribution of the road surface fluctuates, which makes the left-side vehicle (Fig. 3.13a, tracklet number 7) hardly being detected. In Dataset 2, track fragmentation happens when obstacles move closely (tracklet number 12 and tracklet number 13). They are illustrated in Fig. 3.13b, in which different colors of track stands for different tracks. Under the high speed circumstances, obstacles that move towards the camera, have a high relative speed. As projected in the image, their centroids move faster and their sizes change rapidly over time. When the tracker is lost for a certain period, it will be pruned and a new tracker will be created for the obstacle. One should notice here that, even Dataset 2 has a missed detection rate close to Dataset 1 (3.3% to 4%), its rate of track overlap outperforms Dataset 1 by about 9%. Because in Dataset 2, obstacles are detected again soon after their missed detections, thus the tracks remain continuous and complete. There also exist redundant tracks in both Datasets. They are caused by redundant detections where the disparity map is not precise enough.



(a) *Dataset 1* **(b)** *Dataset 2*

Figure 3.13 – *Consistence of the tracks related to tracklets*

Measures	rate of track fragmentation	rate of overlap	average error of centroid	average error of size
Dataset 1	2.36%	87.6%	5.5 px	7.5 px
Dataset 2	6.81%	96.3%	8.6 px	11.8 px

Table 3.2 – *Evaluation of the multiple-traget tracking results*

3.5 Conclusion

In this chapter, we proposed a reliable stereo vision based obstacle detection system that can be directly applied in various driving scenarios. It is capable of detecting all the on-road obstacles with efficiency and accuracy, regardless of their shapes, poses and motion models. The proposed sub-image of disparity map which is used for detection refinement effectively improves the obstacle detection accuracy on multiple measurements. Moreover, the modified particle filter for visual tracking shows a great tolerance for the dynamics of obstacles in images caused by projective scale problem. When facing the projective scale problem, a self-adaptive gate for data association and dynamic filter noise function have been applied to enhance the tracking performance. Experimental results indicate that our detection and tracking system is efficient and reliable. Most obstacles that appear in 35m, can be accurately detected and correctly tracked. The main contribution of this work is that our algorithm can work under dynamic circumstances. Nevertheless, the use of 2D coordinates has a certain limit for further localization and tracking of the obstacles. Therefore, the next research step will be to focus on exploring the proposed approach from a 3D point of view. Furthermore, a main advantage of vision system opposed to radar point tracking is the rich appearance information, which can simplify data association in case of close-range targets. This could be integrated in our future detection and tracking system.

Compared to the mono-vision based moving object detection system, the stereo vision based detection is faster, and is able to detect all kinds of obstacles. Together with the tracking algorithm, the obstacle detection and tracking system that is proposed in this chapter can be seen as an extension from Chapter 2.

Conclusions and Perspectives

Synthesis

Road traffic accidents are perceived as one of the major societal problems in today's world. This thesis was devoted to improve the traffic safety using intelligent vehicles technologies. A complete intelligent vehicle operation is mainly composed of three stages: perception, decision making and control. The differences between the two main functionalities lie in the last two stages: for ADAS, drivers react to the enhanced perception information; for autonomous vehicles, advanced algorithms define how the vehicles have to react to the environment. As the common stage to both ADAS and autonomous vehicles, the perception system is the foundation of all intelligent vehicles technologies. The main contribution of this thesis is that we proposed a reliable and complete vision-based perception system dedicated to dynamic scene understanding in complex environments.

First, it is able to detect the free road area under variant illumination conditions. This is an intuitive information for path planning and obstacle detection. In addition, the free road area can be used to generate a driving space which is also the region of interest of the object detection and tracking process. This algorithm is developed in three different levels based on the input resources and output requirements. The use of confidence intervals allows the algorithm to work in monovision without prior knowledge . When a stereo rig is available, the detection result is refined using disparity map analysis. The highest level fuses the information from illuminant invariant image and disparity profile in a confidence map. This result shows a strong adaptability in complex road conditions, even with several road users.

Next, a geometric constraints-based moving object detection algorithm is proposed under the condition of monocular camera perception. Within this algorithm, the visual odometry has been used to calculate the trifocal tensor from three images over time using a sliding buffer strategy. More than that, visual odometry also works as a camera motion test. It leads to different strategies. When the camera is moving, geometric constraints are applied to filter out parallax pixels from moving ones. When the camera is static, a background subtraction approach is applied to detect the moving pixels. The

detection result from the first strategy relies on a precise dense optical flow, otherwise there would remain several false alarms in the scene. One drawback of the geometric constraints, is that there exist degenerated configurations that prevent the moving objects to be detected. Fortunately, these situations happen rarely, and can be assisted by the object tracking schema.

Meanwhile, an obstacle detection and tracking algorithm has been built using stereovision. The two detection algorithms ensure the reliability and the integrity of the perception system. Monovision-based algorithm can efficiently detect the most dangerous elements in the scene along with their motion state. Stereovision-based detection provides more general information which can assist the collision avoidance in the presence of all types of obstacles. It is specially helpful by detecting the temporary traffic signs like traffic cones and warning signs. Moreover, the depth information from stereovision plays an important role in the tracking algorithm. A particle filter is used and can adjust its parameters according to the obstacle's depth information (i.e. the disparity value). Such a modification enables a reliable object tracking in the image plane.

This perception system is also easy to implement in a vehicle platform with a stereo rig. When the two cameras are both functional, the free road surface detection and obstacle detection/tracking are performed, meanwhile moving objects are detected by geometric constraints performed with left camera. If one of the cameras is not working, the free road surface detection switch to monovision mode, and only moving objects will be detected with the functional camera.

Perspectives

Extensions to current work

- **Driving space construction:** The definition of driving space in this thesis is the entire road area where all the traffic participants may appear. In our current work, we use a convex hull operation to build a quasi-complete driving space. However, as analyzed in Chapter 2, this method may lead to missed detections. From this consideration, building an evolved driving space model (e.x. road shape, edge locations, etc) over time could be an interesting research direction.

- **Moving object detection:** We are interested in applying a sparse to dense strategy for moving object detection in order to reduce the false alarms induced by optical flow errors. Especially, the dense optical flow estimation from a driving scene is a big challenge. Many researchers have been testing their method on the KITTI dataset. Fig. 2.20 presents one of their works. As we can see, in the strong

parallax regions, the estimation errors are rather important. This is because, in this area, there are no good feature to track. Instead, sparse features tracking can overcome the problem. Indeed, only the strong feature points are tracked through multiple views. After localizing the potential target, the dense detection method can be applied to refine the result. Image segmentation may also be used to improve the accuracy and the robustness of our system. It is important to notice that, since the camera motion is determined by the velocity of the vehicle, a fixed interval between key frames for triplets calculation may lead to the deviation of the detection performances. From this consideration, a dynamic temporal frames window could be used to improve the actual results.

- **On-road obstacle tracking:** In this thesis, the obstacle tracking is performed in the image plane. It is helpful for HMI (Human Machine Interface) design, like with Head-up display, to inform the driver within an ADAS functionality. But for autonomous vehicle, tracking in the image plane cannot be used as direct information for decision making and control. Therefore, we are thinking about constructing a tracking algorithm in the world coordinates frame. In this algorithm, the velocity of obstacles and their distance to the host vehicle can be directly used for higher level applications.

- **Reliability test of the system:** The moving object detection by monovision and on-road obstacle detection by stereovision are redundant functions that are designed to improve the system reliability. We would like to have the stereovision function and monovision function working at the same time. Besides, the system should be able to detect the failure of the cameras. Once one of the stereo cameras is not working, the other camera should be able to automatically function in monovision mode.

Long-term developments

We would like to integrate the full system in a real experiment platform, i.e. intelligent vehicles in the Heudiasyc laboratory, for further research.

Moreover, with all the existing components in our current system, it is possible to integrate Simultaneous Localization, Mapping, and Moving Object Tracking (SLAM-MOT) framework. The geometric constraints estimated for moving object detection can be directly used during the 3D scene reconstruction. On the other hand, visual odometry estimation and depth information from stereovision can serve to improve the localization. Especially if monovision is used, the static on-road obstacles can also be detected using machine learning approaches based on image feature aggregation. This will greatly improve the integrity of the system when it functions in monovision mode.

A last research direction to improve the integrity of the perception system, would be to integrate other sensors modalities in a data fusion framework, using visual confirmation architecture from range sensors measurements.

Appendix A

Data Processing for Road Detection

A. The choice of bin-width

To form a histogram, the bin-width (noted as BW) is a important parameter which directly related to the result: If BW is too small, it will lead to large variance. On the other hand, if BW is too large, then the histogram represent statistically large bias. Hence, the proper choice for BW should balance the bias and variance by minimizing, for example, the integrated mean squared error. According to [Sco79]: the bin width parameter is proposed as guideline for histogram construction, which assumes a Gaussian reference standard and requires small sample size and an estimate of the standard deviation. The optimal histogram bin width is derived which asymptotically minimizes the integrated mean squared error.

$$BW = 3.49\sigma N^{1/3} \tag{A.1}$$

where, N is the number of samples σ is an estimate of the standard deviation.

B. Outliers exclusion

In [Ami05] Amidan and his colleagues proposed a effective way of excluding the outliers in a mount of data depend on Chebyshev inequality. This method assumes that the data are independent measurements and that a relatively small percentage of outliers are contained in the data, but there is no assumptions about the distribution of the data. We can calculate upper and lower outlier detection limits with the Chebyshev inequality:

$$P(|X - \mu| \leq k\sigma) \geq (1 - \frac{1}{k^2}). \tag{A.2}$$

X represents the data, μ is the data mean, σ is the standard deviation of the data, and k represents the number of standard deviations from the mean.

The author proposed two method of calculation: one for data is not unimodal another is for unimodal. Considering our concrete application, we chose the method for no-unimodal, here are the steps:

1. Values for $P_{1,}P_2$ are decided; P_1 which determine the outlier percentage of the whole data during the calculation; P_2 determine the final outlier percentage to expel. As an example in this article P_2 equals to 0.1 since we could like to preserve 90% of the raw data . While P_1equals to 0.2 which is a value preferred to be larger than P_2.

2. Calculate k with equation:

$$k_1 = \frac{1}{\sqrt{p_1}} \tag{A.3}$$

3. Use Chebyshev inequality to calculate upper and lower outlier detection boundaries :

$$ODV_{1u} = \mu_1 + k_1\sigma_1 \tag{A.4}$$

$$ODV_{1l} = \mu_1 - k_1\sigma_1 \tag{A.5}$$

$\mu_1$and σ_1 are got by complete dataset.

4. All data that are more extreme than the appropriate $ODV_{1u,1l}$ are considered to be outliers.

5. With the data reserved within the $ODV_{1u,1l}$, calculate a new $\mu_2$and $\sigma 2$, and set a value P_1which is smaller thanP_2 ,which is the range that we want for further calculation.

6. Like step2. calculate a new k with

$$k_2 = \frac{1}{\sqrt{p_2}} \tag{A.6}$$

7. Calculate the upper bound and lower bound by equations:

$$ODV_{2u} = \mu_2 + k_2\sigma_2 \tag{A.7}$$

$$ODV_{2l} = \mu_2 - k_2\sigma_2 \tag{A.8}$$

$\mu_2$and σ_2 are got by reserved data-set.

4. All data (for complete data-set) that are more extreme than the appropriate ODV are considered to be outliers.

This method allows for detection of multiple outliers, not just one at a time. Chebyshev's inequality gives a bound of what percentage of the data falls outside of k standard deviations from the mean. Data values that are not within the range of the upper and lower limits would be considered data outliers. It identifies potential outlier data with a more reasonable result.

Appendix B

Inverse Perspective Mapping

Coordinates definition and camera parameters:

To get the Inverse Perspective Mapping (IPM) of the input image, we assume a flat road, and use the camera intrinsic and extrinsic parameters to perform the transformation.

- intrinsic parameters:

focal length : $f_u,\ f_v$

optical center : $c_u,\ c_v$

- extrinsic parameters:

pitch angle : α

yaw angle : β

height above ground : h

The relationship of the world coordinate with camera coordinate defines the pitch angle and yaw angle of the camera. Fig.B.1 illustrates the relationship between different coordinates. To be noticed, The world coordinates are denoted by $(X_W,\ Y_W,\ Z_W)$ which centered at the camera optical center. The camera coordinates are denoted by $(X_C,\ Y_C,\ Z_C)$.

The coordinates of image plane for transformation is defined as as $(u, v, 1)$

A. Transformation from perspective view to Bird-Eye-View

For any point in the image plane ${}^iP = (u, v, 1, 1)$ (perspective view), its projection on the road plane (Bird-Eye-View) can be found by applying the homogeneous transformation:

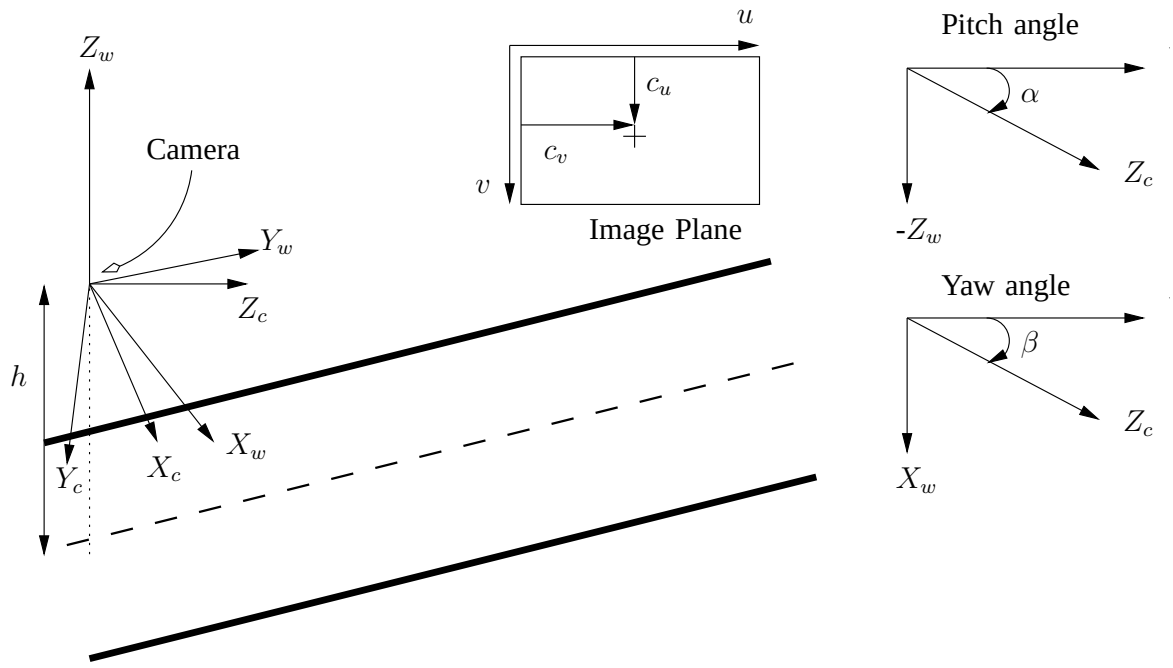


Figure B.1 – *The relationship between different coordinates, figure from [Aly08]*

$$ {}^{g}P = {}^{g}_{i}T\,{}^{i}P $$

$$ {}^{g}_{i}T = h \begin{bmatrix} -\frac{1}{f_u}c_2 & \frac{1}{f_v}s_1 s_2 & \frac{1}{f_u}c_u c_2 - \frac{1}{f_v}c_v s_1 s_2 - c_1 s_2 & 0 \\ \frac{1}{f_u}s_2 & \frac{1}{f_v}s_1 c_1 & -\frac{1}{f_u}c_u s_2 - \frac{1}{f_v}c_v s_1 c_2 - c_1 c_2 & 0 \\ 0 & \frac{1}{f_v}c_1 & -\frac{1}{f_v}c_v c_1 + s_1 & 0 \\ 0 & -\frac{1}{h f_v}c_1 & \frac{1}{h f_v}c_v c_1 - \frac{1}{h}s_1 & 0 \end{bmatrix} \tag{B.1} $$

where, $c_1 = \cos\alpha$, $c_2 = \cos\beta$, $s_1 = \sin\alpha$, $s_2 = \sin\beta$

B. Transformation from Bird-Eye-View to perspective view

Inverse transform from points on road plane ${}^{g}P = (x_g, y_g, -h, 1)$ to the image plane of perspective view

$$ {}^{i}_{g}T = \begin{bmatrix} f_u c_2 + c_u c_1 s_2 & c_u c_1 c_2 - s_2 f_u & -c_u s_1 & 0 \\ s_2\left(c_v c_1 - f_v s_1\right) & c_2\left(c_v c_1 - f_v s_1\right) & -f_v c_1 - c_v s_1 & 0 \\ c_1 s_2 & c_1 c_2 & -s_1 & 0 \\ c_1 s_2 & c_1 c_2 & -s_1 & 0 \end{bmatrix} \tag{B.2} $$

where, $c_1 = \cos\alpha$, $c_2 = \cos\beta$, $s_1 = \sin\alpha$, $s_2 = \sin\beta$

Appendix C

Geometrical Relationships in Trifocal Tensor

A. Geometry relationships of trifocal tensor in standard tensor notation

Trifocal tensor provides 4 type of relationships in the three views:

- Three corresponding points/triplets:$x^i x'^j x"^k \epsilon_{jqs}\epsilon_{krt}\mathcal{T}_i^{qr} = 0_{st}$
- Two points and a line:$x^i x'^j l"_r \epsilon_{jqs}\mathcal{T}_i^{qr} = 0_s$
- Two lines and a point:$x^i l'_q l"_r \mathcal{T}_i^{qr} = 0$
- Three lines :$l_p l'_q l"_r \epsilon^{piw}\mathcal{T}_i^{qr} = 0^w$

where,

$$\epsilon_{rst} = \begin{cases} 0 & \text{unless } r, s \text{ and } t \text{ are distinct} \\ +1 & \text{if } rst \text{ is an even permutation of 123} \\ -1 & \text{if } rst \text{ is an odd permutation of 123} \end{cases}$$

B. Extract epipolar geometry from trifocal tensor

Given the trifocal tensor written in matrix notation as $[T_1, T_2, T_3]$.The epipolar geometry can be extracted from the tensors.

- Retrieve the epipoles e_{21}, e_{31}:

Let u_i and v_i be the left and right null-vectors respectively of T_i, i.e. $u_i^T T_i = 0^T$,$T_i v_i = 0$. Then the epipoles are obtained as the null-vectors to the following 3×3 matrices:

$$e_{21}^T [u_1, u_2, u_3] = 0$$

$$e_{31}[v_1, v_2, v_3] = 0$$

– Retrieve the fundamental matrices F_{21} F_{31}:

$$F_{21} = [e_{21}]_{\times}[T_1, T_2, T_3]e_{31}$$

$$F_{31} = [e_{31}]_{\times}[T_1^T, T_2^T, T_3^T,]e_{21}$$

References

[AGLL12] Jose M. Alvarez, Theo Gevers, Yann LeCun, and Antonio M. Lopez. Road scene segmentation from a single image. In *ECCV 2012*, volume 7578 of *Lecture Notes in Computer Science*, pages 376–389. Springer Berlin Heidelberg, 2012.

[AL11] JMA Alvarez and Antonio M Lopez. Road detection based on illuminant invariance. *Intelligent Transportation Systems, IEEE Transactions on*, 12(1):184–193, 2011.

[Aly08] Mohamed Aly. Real time detection of lane markers in urban streets. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 7–12. IEEE, 2008.

[Ami05] Richland WA Ferryman T.A. ; Cooley S.K. Amidan, B.G.Battelle-Pacific Northwest Div. Data outlier detection using the chebyshev theorem. *Aerospace Conference, 2005 IEEE*, 2005.

[AS09] Alper Ayvaci and Stefano Soatto. Motion segmentation with occlusions on the superpixel graph. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 727–734. IEEE, 2009.

[BB94] Serge Beucher and Michel Bilodeau. Road segmentation and obstacle detection by a fast watershed transformation. In *Intelligent Vehicles' 94 Symposium, Proceedings of the*, pages 296–301. IEEE, 1994.

[BBA10] Adrien Bak, Samia Bouchafa, and Didier Aubert. Detection of independently moving objects through stereo vision and ego-motion extraction. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 863–870. IEEE, 2010.

[BBA14] Adrien Bak, Samia Bouchafa, and Didier Aubert. Dynamic objects detection through visual odometry and stereo-vision: a study of inaccuracy and improvement sources. *Machine vision and applications*, 25(3):681–697, 2014.

[BK08] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[BMVF08] Hernán Badino, Rudolf Mester, Tobi Vaudrey, and Uwe Franke. Stereo-based free space computation in complex traffic scenarios. In *Image Analysis and Interpretation, 2008. SSIAI 2008. IEEE Southwest Symposium on*, pages 189–192. IEEE, 2008.

[BRL+09] Michael D Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1515–1522. IEEE, 2009.

[CG05] Pietro Cerri and Paolo Grisleri. Free space detection on highways using time correlation between stabilized sub-pixel precision ipm images. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2223–2228. IEEE, 2005.

[DH72] Richard O Duda and Peter E Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.

[DRSS12] Soumyabrata Dey, Vladimir Reilly, Imran Saleemi, and Mubarak Shah. Detection of independently moving objects in non-planar scenes via multi-frame monocular epipolar constraint. In *Computer Vision–ECCV 2012*, pages 860–873. Springer, 2012.

[EM08] Ran Eshel and Yael Moses. Homography based multiple camera detection and tracking of people in a dense crowd. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[ESLVG10] Andreas Ess, Konrad Schindler, Bastian Leibe, and Luc Van Gool. Object detection and tracking for autonomous navigation in dynamic environments. *The International Journal of Robotics Research*, 29(14):1707–1725, 2010.

[EV09] Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2790–2797. IEEE, 2009.

[EVGW+10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[FB81] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[FCC+03] Andrea Fusiello, Stefano Caldrer, Sara Ceglie, Nikolaus Mattern, and Vittorio Murino. View synthesis from uncalibrated images using parallax. In *Image Analysis and Processing, 2003. Proceedings. 12th International Conference on*, pages 146–151. IEEE, 2003.

[FDC04] G.D. Finlayson, M.S. Drew, and L. Cheng. Intrinsic images by entropy minimization. In *European Conference on Computer Vision*, 2004.

[FF84] Keinosuke Fukunaga and Thomas E Flick. An optimal global nearest neighbor metric. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (3):314–318, 1984.

[FFBC14] Sergio Alberto Rodriguez Florez, Vincent Frémont, Philippe Bonnifait, and Véronique Cherfaoui. Multi-modal object detection and localization for high integrity driving assistance. *Machine vision and applications*, 25(3):583–598, 2014.

[FKG13] Jannik Fritsch, Tobias Kuehnl, and Andreas Geiger. A new performance measure and evaluation benchmark for road detection algorithms. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.

[FMP+13] Xue Fan, Shubham Mittal, Twisha Prasad, Suraj Saurabh, and Hyunchul Shin. Pedestrian detection and tracking using deformable part models and kalman filtering. *Journal of Communication and Computer*, 10:960–966, 2013.

[G.D01] M.S.Drew G.D.Finlayson. 4-sensor camera calibration for image representation invariant to shading, shadows, lighting, and specularities. *ICCV'01: International Conference on Computer Vision,IEEE*, 2001.

[G.D09] Cheng Lu G.D.Finlayson, M.S.Drew. Entropy minimization for shadow removal. *International Journal of Computer Vision*, 2009.

[GG12] Stevica Graovac and Ahmed Goma. Detection of road image borders based on texture classification. *International Journal of Advanced Robotic Systems*, 9, 2012.

[GHRDKP14] V Gonzalez-Huitron, E Ramos-Diaz, V Kravchenko, and V Ponomaryov. 2d to 3d conversion based on disparity map estimation. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 982–989. Springer, 2014.

[GLSU13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[GLU12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[GLW+14] Andreas Geiger, Martin Lauer, Christian Wojek, Christoph Stiller, and Raquel Urtasun. 3d traffic scene understanding from movable platforms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(5):1012–1025, 2014.

[GMM09] Chunzhao Guo, Seiichi Mita, and David McAllester. Drivable road region detection using homography estimation and efficient belief propagation with coordinate descent optimization. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 317–323. IEEE, 2009.

[GRU11] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *Computer Vision–ACCV 2010*, pages 25–38. Springer, 2011.

[GZS11] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV)*, 2011.

[HCB+13] Jacques Harvent, Benjamin Coudrin, Ludovic Brèthes, Jean-José Orteu, and Michel Devy. Multi-view dense 3d modelling of untextured objects from a moving projector-cameras system. *Machine vision and applications*, 24(8):1645–1659, 2013.

[Hei00] Bernd Heisele. Motion-based object detection and tracking in color image sequences. In *Fourth Asian Conference on Computer Vision*, pages 1028–1033, 2000.

[HLPA06] Nicolas Hautière, Raphaël Labayrade, Mathias Perrollaz, and Didier Aubert. Road scene analysis by stereovision: a robust and quasi-dense approach. In *Control, Automation, Robotics and Vision, 2006. ICARCV'06. 9th International Conference on*, pages 1–6. IEEE, 2006.

[HU05] Zhencheng Hu and Keiichi Uchimura. Uv-disparity: an efficient algorithm for stereovision based scene analysis. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 48–54. IEEE, 2005.

[HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[IB98] Michael Isard and Andrew Blake. Condensation-conditional density propagation for visual tracking. *International journal of computer vision*, 29(1):5–28, 1998.

[JCL14] Shyr-Long Jeng, Wei-Hua Chieng, and Hsiang-Pin Lu. Estimating speed using a side-looking single-radar vehicle detector. *Intelligent Transportation Systems, IEEE Transactions on*, 15(2):607–614, 2014.

[JCZT11] Amirali Jazayeri, Hongyuan Cai, Jiang Yu Zheng, and Mihran Tuceryan. Vehicle detection and tracking in car video based on motion model. *Intelligent Transportation Systems, IEEE Transactions on*, 12(2):583–595, 2011.

[JDSW12] Roland Jonsson, Johan Degerman, Daniel Svensson, and Johannes Wintenby. Multi-target tracking with background discrimination using phd filters. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 854–860. IEEE, 2012.

[JT77] S.T. McDonald D. Shinar R.D. Hume R.E. Mayer R.L. Stansifer N.J. Castellan J.R. Treat, N.S. Tumbas. Special analyses final report. U.S. Department of Transportation Washington, DC: Government Printing Office (Indiana University, Institute for Research in Public Safety), 1977.

[JT12] Kinjal A Joshi and Darshak G Thakore. A survey on moving object detection and tracking in video surveillance system. *IJSCE, ISSN*, pages 2231–2307, 2012.

[KAP10] Hui Kong, J-Y Audibert, and Jean Ponce. General road detection from a single image. *Image Processing, IEEE Transactions on*, 19(8):2211–2220, 2010.

[KB12] Sebastien Kramm and Abdelaziz Bensrhair. Obstacle detection using sparse stereovision and clustering techniques. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 760–765. IEEE, 2012.

[KBD05] Zia Khan, Tucker Balch, and Frank Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(11):1805–1819, 2005.

[KKF12] T. Kuehnl, F. Kummert, and J. Fritsch. Spatial ray features for real-time ego-lane extraction. In *Proc. IEEE Intelligent Transportation Systems*, 2012.

[KKJ10] Abhijit Kundu, K Madhava Krishna, and CV Jawahar. Realtime motion segmentation based multibody visual slam. In *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing*, pages 251–258. ACM, 2010.

[KKS09] Abhijit Kundu, K Madhava Krishna, and Jayanthi Sivaswamy. Moving object detection by multi-view geometric techniques from a single camera mounted robot. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 4306–4312. IEEE, 2009.

[KS09] Saad M Khan and Mubarak Shah. Tracking multiple occluding people by localizing on multiple scene planes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(3):505–519, 2009.

[LAT02] Raphael Labayrade, Didier Aubert, and J-P Tarel. Real time obstacle detection in stereovision on non flat road geometry through" v-disparity" representation. In *Intelligent Vehicle Symposium*, volume 2, pages 646–651. IEEE, 2002.

[Liu09] Ce Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Citeseer, 2009.

[LKSV07] Ting Li, Vinutha Kallem, Dheeraj Singaraju, and René Vidal. Projective factorization of multiple rigid-body motions. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–6. IEEE, 2007.

[LLKK11] Chung-Hee Lee, Young-Chul Lim, Soon Kwon, and Jonghwan Kim. Stereo vision-based obstacle detection using dense disparity map. In *2011 International Conference on Graphic and Image Processing*, pages 82853O–82853O. International Society for Optics and Photonics, 2011.

[LSH+12] DF Llorca, MA Sotelo, AM Hellín, A Orellana, M Gavilán, IG Daza, and AG Lorente. Stereo regions-of-interest selection for pedestrian protection: A survey. *Transportation research part C: emerging technologies*, 25:226–237, 2012.

[MAG+02] Luis Moreno, Jose M Armingol, Santiago Garrido, Arturo De La Escalera, and Miguel A Salichs. A genetic algorithm for mobile robot localization using ultrasonic sensors. *Journal of Intelligent and Robotic Systems*, 34(2):135–154, 2002.

[Mag04] Derek R Magee. Tracking multiple vehicles using foreground, background and motion models. *Image and vision Computing*, 22(2):143–155, 2004.

[Mar63] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial & Applied Mathematics*, 11(2):431–441, 1963.

[MC08] Fernando C Monteiro and Aurélio Campilho. Region and graph-based motion segmentation. In *Image Analysis and Recognition*, pages 609–618. Springer, 2008.

[MCB10] Julien Moras, Véronique Cherfaoui, and Philippe Bonnifait. A lidar perception scheme for intelligent vehicle navigation. In *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*, pages 1809–1814. IEEE, 2010.

[MCTM05] Roberto Manduchi, Andres Castano, Ashit Talukder, and Larry Matthies. Obstacle detection and terrain classification for autonomous off-road navigation. *Autonomous robots*, 18(1):81–102, 2005.

[MDB08] Thomas Mauthner, Michael Donoser, and Horst Bischof. Robust tracking of spatial related components. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[MGD12] David Márquez-Gámez and Michel Devy. Active visual-based detection and tracking of moving objects from clustering and classification methods. In *Advanced Concepts for Intelligent Vision Systems*, pages 361–373. Springer, 2012.

[MLIM10] Pauline Merveilleux, Ouiddad Labbani-Igbida, and El Mustapha Mouaddib. Free space detection using active contours in omnidirectional images. In *ICIP*, pages 3533–3536, 2010.

[MRD+12] Julien Moras, FSA Rodriguez, Vincent Drevelle, Gérald Dherbomez, Véronique Cherfaoui, and Philippe Bonnifait. Drivable space characterization using automotive lidar and georeferenced map information. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 778–783. IEEE, 2012.

[MRM+09] Davide Migliore, Roberto Rigamonti, Daniele Marzorati, Matteo Matteucci, and Domenico G Sorrenti. Use a single camera for simultaneous localization and mapping with mobile object tracking in dynamic environments. In *Proceedings of International workshop on Safe navigation in open and dynamic environments application to autonomous vehicles*, 2009.

[MTC08] Emilio Maggio, Murtaza Taj, and Andrea Cavallaro. Efficient multitarget visual tracking using random finite sets. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(8):1016–1027, 2008.

[NHLM13] Manjunath Narayana, Allen Hanson, and Erik Learned-Miller. Coherent motion segmentation in moving camera videos using optical flow orientations. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1577–1584. IEEE, 2013.

[NJW+02] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

[NKKJ12] Rahul Kumar Namdev, Abhijit Kundu, K Madhava Krishna, and CV Jawahar. Motion segmentation of multiple objects from a freely moving monocular camera. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4092–4099. IEEE, 2012.

[NSH07] Didier Aubert Nicolas Soquet and Nicolas Hautiere. Road segmentation supervised by an extended v-disparity algorithm for autonomous navigation. *Intelligent Vehicle Symposium, 2007. IEEE*, 2007.

[NTMM12] Hossein Tehrani Niknejad, Akihiro Takeuchi, Seiichi Mita, and David McAllester. On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation. *Intelligent Transportation Systems, IEEE Transactions on*, 13(2):748–758, 2012.

[NVFZ11] A Miranda Neto, Alessandro Correa Victorino, Isabelle Fantoni, and Douglas Eduardo Zampieri. Robust horizon finding algorithm for real-

time autonomous navigation based on monocular vision. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 532–537. IEEE, 2011.

[OB12] Peter Ochs and Thomas Brox. Higher order motion models and spectral clustering. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 614–621. IEEE, 2012.

[ORS04] Songhwai Oh, Stuart Russell, and Shankar Sastry. Markov chain monte carlo data association for general multiple-target tracking problems. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 1, pages 735–742. IEEE, 2004.

[OSVG10] Kemal E Ozden, Konrad Schindler, and Luc Van Gool. Multibody structure-from-motion in practice. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(6):1134–1141, 2010.

[OTDF+04] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J Little, and David G Lowe. A boosted particle filter: Multitarget detection and tracking. In *Computer Vision-ECCV 2004*, pages 28–39. Springer, 2004.

[PB06] Shrinivas J Pundlik and Stanley T Birchfield. Motion segmentation at any speed. In *BMVC*, pages 427–436, 2006.

[PDH+14] Danda Pani Paudel, Cédric Demonceaux, Adlane Habed, Pascal Vasseur, and In So Kweon. 2d-3d camera fusion for visual odometry in outdoor environments. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 157–162. IEEE, 2014.

[PLR+06] Mathias Perrollaz, Raphaël Labayrade, Cyril Royere, Nicolas Hautiere, and Didier Aubert. Long range obstacle detection using laser scanner and stereovision. In *Intelligent Vehicles Symposium, 2006 IEEE*, pages 182–187. IEEE, 2006.

[Pri03] David Pritchard. Cloth parameters and motion capture. Technical report, 2003.

[PYSL10] Mathias Perrollaz, J-D Yoder, Anne Spalanzani, and Christian Laugier. Using the disparity space to compute occupancy grids from stereo-vision. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2721–2726. IEEE, 2010.

[RAG04] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon. Beyond the kalman filter. *IEEE AEROSPACE AND ELECTRONIC SYSTEMS MAGAZINE*, 19(7):37–38, 2004.

[RGP13] Marc Revilloud, Dominique Gruyer, and Evangeline Pollard. An improved approach for robust road marking detection and tracking applied to multi-lane estimation. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 783–790. IEEE, 2013.

[RLSA11] Mikel Rodriguez, Ivan Laptev, Josef Sivic, and J-Y Audibert. Density-aware person detection and tracking in crowds. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2423–2430. IEEE, 2011.

[ROA03] Road safety: Impact of new technologies. OECD, 2003.

[Rou84] Peter J Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984.

[RTVM08] Shankar R Rao, Roberto Tron, René Vidal, and Yi Ma. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[S+04] Mehmet Sezgin et al. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic imaging*, 13(1):146–168, 2004.

[Sco79] David W. Scott. On optimal and data-based histograms. *Biometrika*, 1979.

[SGPO05] Kevin Smith, Daniel Gatica-Perez, and Jean-Marc Odobez. Using particles to track varying numbers of interacting people. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 962–969. IEEE, 2005.

[SM00] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[SO07] Akihito Seki and Masatoshi Okutomi. Robust obstacle detection in general road environment based on road extraction and pose estima-

tion. *Electronics and Communications in Japan (Part II: Electronics)*, 90(12):12–22, 2007.

[TV98] Emanuele Trucco and Alessandro Verri. *Introductory techniques for 3-D computer vision*, volume 201. Prentice Hall Englewood Cliffs, 1998.

[VRS14] Christoph Vogel, Stefan Roth, and Konrad Schindler. View-consistent 3d scene flow estimation over multiple frames. In *Computer Vision–ECCV 2014*, pages 263–278. Springer, 2014.

[WF13] Bihao Wang and Vincent Frémont. Fast road detection from color images. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 1209–1214. IEEE, 2013.

[WKT06] Shiuh-Ku Weng, Chung-Ming Kuo, and Shu-Kang Tu. Video object tracking using adaptive kalman filter. *Journal of Visual Communication and Image Representation*, 17(6):1190–1208, 2006.

[WS82] Gunter Wyszecki and Walter Stanley Stiles. *Color science*, volume 8. Wiley New York, 1982.

[YGyY07] Sha Yun, Zhang Guo-ying, and Yang Yong. A road detection algorithm by boosting using feature combination. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 364–368. IEEE, 2007.

[YMC07] Qian Yu, Gérard Medioni, and Isaac Cohen. Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[YMKC07] Chang Yuan, Gerard Medioni, Jinman Kang, and Isaac Cohen. Detecting motion regions in the presence of a strong parallax from a moving camera by multiview geometric constraints. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(9):1627–1641, 2007.

[YMOI08] Wataru Yoshizaki, Yoshihiko Mochizuki, Naoya Ohnishi, and Atsushi Imiya. Free space detection from catadioptric omnidirectional images for visual navigation using optical flow. In *The 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras-OMNIVIS*, 2008.

[YMU14] Koichiro Yamaguchi, David McAllester, and Raquel Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *ECCV*, 2014.

[YP06] Jingyu Yan and Marc Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *Computer Vision–ECCV 2006*, pages 94–106. Springer, 2006.

[Zha00] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.

[ZNW08] Tao Zhao, Ramakant Nevatia, and Bo Wu. Segmentation and tracking of multiple humans in crowded environments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(7):1198–1211, 2008.