Ph.D. DISSERTATION

Vision-based Distance Measurement and Localization for Automated Driving

자율주행을 위한 카메라 기반 거리 측정 및 측위

유 인 섭

AUGUST 2017

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
GRADUATE SCHOOL
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

Vision-based Distance Measurement and Localization for Automated Driving

자율주행을 위한 카메라 기반 거리 측정 및 측위

유 인 섭

AUGUST 2017

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
GRADUATE SCHOOL
SEOUL NATIONAL UNIVERSITY

Vision-based Distance Measurement and Localization for Automated Driving

자율주행을 위한 카메라 기반 거리 측정 및 측위

지도교수 서 승 우
이 논문을 공학박사 학위논문으로 제출함

2017년 8월

서울대학교 대학원

전기 컴퓨터 공학부

유 인 섭

유인섭의 공학박사 학위 논문을 인준함

2017년 8월

위 원 장: 조 남 익
부위원장: 서 승 우
위 원: 김 창 수
위 원: 김 현 진
위 원: 김 성 우

Abstract

Automated driving vehicles or advanced driver assistance systems (ADAS) have continued to be an important research topic in transportation area. They can promise to reduce road accidents and eliminate traffic congestions. Automated driving vehicles are composed of two parts. On-board sensors are used to observe the environments and then, the captured sensor data are processed to interpret the environments and to make appropriate driving decisions. Some sensors have already been widely used in existing driver-assistance systems, e.g., camera systems are used in lane-keeping systems to recognize lanes on roads; radars (Radio Detection And Ranging) are used in adaptive cruise systems to measure the distance to a vehicle ahead such that a safe distance can be guaranteed; LIDAR (Light Detection And Ranging) sensors are used in the autonomous emergency braking system to detect other vehicles or pedestrians in the vehicle path to avoid collision; accelerometers are used to measure vehicle speed changes, which are especially useful for air-bags; wheel encoder sensors are used to measure wheel rotations in a vehicle anti-lock brake system and GPS sensors are embedded on vehicles to provide the global positions of the vehicle for path navigation.

In this dissertation, we cover three important application for automated driving vehicles by using camera sensors in vehicular environments. Firstly, precise and robust distance measurement is one of the most important requirements for driving assistance systems and automated driving systems. We propose a new method for providing accurate distance measurements through a frequency-domain analysis based on a stereo camera by exploiting key information obtained from the analysis of captured images. Secondly, precise and robust localization is another important requirement for safe automated driving. We propose a method for robust localization in diverse driving situations that measures the vehicle positions using a camera with respect to a given map for vision based navigation. The proposed method includes technology for removing dynamic objects and preserving features in vehicular environments using a background model accumulated from previous frames and we improve image quality using illuminant invariance characteristics of the log-chromaticity. We also propose a vehicle localization method using structure tensor and mutual information theory. Finally, we propose a novel algorithm for estimating the drivable collision-free space for autonomous navigation of on-road vehicles. In contrast to previous approaches that use stereo cameras or LIDAR, we solve this problem using a sensor fusion of cameras and LIDAR.

keywords: Automated driving, distance measurement, Free space detection, image processing, vehicle localization.

student number: 2011-20885

Contents

Abstract **i**

Contents **iii**

List of Tables **vi**

List of Figures **vii**

1 Introduction **1**

1.1 Background and Motivations 1

1.2 Contributions and Outline of the Dissertation 3

1.2.1 Accurate Object Distance Estimation based on Frequency-Domain Analysis with a Stereo Camera 3

1.2.2 Visual Map Matching based on Structural Tensor and Mutual Information using 3D High Resolution Digital Map 3

1.2.3 Free Space Computation using a Sensor Fusion of LIDAR and RGB camera in Vehicular Environment 4

2 Accurate Object Distance Estimation based on Frequency-Domain Analysis with a Stereo Camera **5**

2.1 Introduction 5

2.2 Related Works 7

2.3 Algrorithm Description 10

2.3.1 Overall Procedure 10

2.3.2 Preliminaries 12

2.3.3 Pre-processing 12

2.4 Frequency-domain Analysis 15

2.4.1 Procedure 15

2.4.2 Contour-based Cost Computation 20

2.5 Cost Optimization and Distance Estimation 21

2.5.1 Disparity Optimization 21

2.5.2 Post-processing and Distance Estimation 23

2.6 Experimental Results 24

2.6.1 Test Environment 24

2.6.2 Experiment on KITTI Dataset 25

2.6.3 Performance Evaluation and Analysis 28

2.7 Conclusion 32

3 Visual Map Matching Based on Structural Tensor and Mutual Information using 3D High Resolution Digital Map 33

3.1 Introduction 33

3.2 Related Work 35

3.3 Methodology 37

3.3.1 Sensor Calibration 37

3.3.2 Digital Map Generation and Synthetic View Conversion . . . 39

3.3.3 Dynamic Object Removal 41
3.3.4 Illuminant Invariance 43
3.3.5 Visual Map Matching using Structure Tensor and Mutual Information 43
3.4 Experiments and Result 49
3.4.1 Methodology 49
3.4.2 Quantitative Results 53
3.5 Conclusions and Future Works 54

4 Free Space Computation using a Sensor Fusion of LIDAR and RGB Camera in Vehicular Environments 55
4.1 Introduction 55
4.2 Methodology 57
4.2.1 Dense Depth Map Generation 57
4.2.2 Color Distribution Entropy 58
4.2.3 Edge Extraction 60
4.2.4 Temporal Smoothness 61
4.2.5 Spatial Smoothness 62
4.3 Experiment and Evaluation 63
4.3.1 Evaluated Methods 63
4.3.2 Experiment on KITTI Dataset 64
4.4 Conclusion 68

5 Conclusion 70

Abstract (In Korean) 87

List of Tables

2.1 Experimental results on the KITTI 2015. The setting column provides insight into how the disparity map is computed: F indicates the use of optical flow. The error column reports the percentage of misclassifed pixels and the runtime column measures the time, in seconds, required to process one pair of images. 28

2.2 Distance accuracy comparison according to images. 31

3.1 Precision and recall for removing dynamic objects. 52

3.2 The localization errors. 53

4.1 Comparison of evaluation on KITTI Dataset. 65

4.2 Distance accuracy comparison on KITTI Dataset. 65

List of Figures

2.1 Processing flow of the proposed algorithm. 11

2.2 Stereo disparity and image coordinates of point. 13

2.3 Comparison of an original and the same image with the sky removed (shown in black). Based on the image with the sky removed, the region of interest is reduce when generating the disparity map. 16

2.4 Comparison of an original and the same image with the sky removed (shown in black). Based on the image with the sky removed, the region of interest is reduce when generating the disparity map. 17

2.5 Original image and the result image through the high-pass Gaussian filter. The pixel value gets brighter around the boundary of the object. 22

2.6 Original image and the result image through the high-pass Gaussian filter. The pixel value gets brighter around the boundary of the object. 22

2.7 SNU autonomous vehicle. The vehicle is equipped with a stereo sensor (indoor), LIDAR (outdoor). 26

2.8 Experimental situations. We tested the algorithm in various environments (tunnel, urban area, highway). 27

2.9 The experimental results 31

3.1 Proposed system architecture. 37

3.2 Original image and extrinsic calibration showing a perspective view with camera sensor data projected into the 3D LIDAR points. 40

3.3 The example of the map data. 41

3.4 Example of synthetic view. 42

3.5 Example of removing dynamic objects. Vehicles are removed while preserving road markers, such as lane markers and speed bumps. . . . 44

3.6 Original and structure tensor images in pseudo color. 46

3.7 Autonomous vehicle equipped with a 3D LIDAR for generating map. monocular camera is mounted on the location of rear-view mirror in the vehicle. 50

3.8 (a) Digital map generation result and (b) aerial image of the area with the vehicle trajectory (red). 51

3.9 Experimental results. First row: original images, Second row: structural tensor images in pseudo color, Third row: synthetic view generated from digital map, Fourth row: the result of removing dynamic object. 52

4.1 Processing flow of the proposed algorithm. 57

4.2 Original image, dense depth images, LIDAR frame. 59

4.3 Examples of original image and depth images. 60

4.4 Original image and LIDAR frame. 66

4.5 Example of experimental result. 66

4.6 Example of experimental result. 67

4.7 Example of experimental result. 67

4.8 Ground truth, false positive, false negative and true positive. 67

Chapter 1

Introduction

1.1 Background and Motivations

The research of automated driving vehicles is to be able to perform navigation without any user interaction. Although this problem has been of interest for several decades, the recent successes of automated driving vehicles have demonstrated that this dream might actually happen in the near future [1]. Especially, a camera sensor can provide plenty of information around vehicular environment. In computer vision, research has focused on a wide variety of problems such as stereo [2][3], scene understanding [4], image-based localization [5][6], and pedestrian detection [7].

A common assumption has been that if we are able to develop robust and accurate solutions to these problems, we should be able to solve autonomous navigation relying mainly on visual information. In this dissertation, three important problems for automated driving vehicles are proposed by using the camera sensor in vehicular environment.

In driving situations, determining the distance to objects in real time is important for various active safety applications, and many of them, such as emergency braking and smart cruise-control systems, rely on object detection and accurate distance measurement to objects. However, it is important to select suitable algorithms with illumination-invariant characteristics and which can be applied in real time with high accuracy levels.

A common approach to automated driving vehicles is to use detailed prior maps that are annotated with precise lane locations, traffic signs, and other metadata that govern the rules of the road. These maps are generated offline, which allows the use of complex algorithms that are not necessarily real-time to be used by the operating automated driving vehicle. The use of prior maps allows researchers to turn some of the difficult perception tasks into a localization problem. Localization in the robotics community is a mature research area that yields a bounded problem given the well structured environment an automobile operates in. localization robustness is critical as it is a subsystem that cannot fail or the online autonomous platform would no longer be able to operate.

Drivable free space detection is challenging due to the drastic change of road scenes, illumination, weather condition and the clutter of background. Since each modal of sensor has its weakness, multi-modal sensor fusion can be a straightforward solution to fill the gap.

This thesis focuses on extending the state-of-the-art to increase robustness of image processing techniques for automated driving vehicles. We propose a fast and efficient computer vision algorithm that can calculate the distance about object and estimate the vehicle's position. Additionally, we propose a sensor fusion algorithm that is able to detect the drivable free space.

1.2 Contributions and Outline of the Dissertation

1.2.1 Accurate Object Distance Estimation based on Frequency-Domain Analysis with a Stereo Camera

Precise and robust distance measurement is one of the most important requirements for driving assistance systems and automated driving systems. In this study, we propose a new method for providing accurate distance measurements through frequency-domain analysis based on a stereo camera by exploiting key information obtained from the analysis of captured images. Moreover, the proposed method was extensively tested and evaluated on a real urban road, highway and tunnel. Based on these results, we show that the proposed method provides more precise distance information in real time compared with conventional algorithms. By applying the methodology to measure the distances of various objects, it can be demonstrated that the algorithm offers an improvement of up to 10 percent.

1.2.2 Visual Map Matching based on Structural Tensor and Mutual Information using 3D High Resolution Digital Map

Precise and robust localization is one of the most important requirements for safe automated driving. We propose a method for robust localization in diverse driving situations that measures the vehicles position using a camera with respect to a given map for vision based navigation. The proposed method includes technology for removing dynamic objects and preserving features in vehicular environments using a background model accumulated from previous frames and we improve image quality using illuminant invariance characteristics of the log chromaticity. We also propose a vehicle localization method using structure tensor and mutual information theory.

The proposed system achieves decimeter order accuracy for visual localization without requiring high precision GPS. The technology is sufficiently robust to be used in diverse weather and road conditions. We evaluated the proposed method using a campus dataset and challenging scenarios, and showed outstanding results for vehicle localization.

1.2.3 Free Space Computation using a Sensor Fusion of LIDAR and RGB camera in Vehicular Environment

In this paper we propose a novel algorithm for determining the drivable free space for automated driving vehicles. In contrast to previous approaches that use cameras or LIDAR, we show a method to solve this problem using a sensor fusion method of LIDAR and a monocular camera. We focus on distance information which we can get from 3D LIDAR point cloud and generate dense depth map. Specifically, given a pair of RGB image and sparse depth map projected from LIDAR point cloud, we generate dense depth map. Furthermore, we compute the drivable free space using visual features from dense depth map. Our algorithm exploits several image and geometric features based on edges, color, temporal and spatial information to estimate the drivable free space. We show promising results on the challenging KITTI dataset.

Chapter 2

Accurate Object Distance Estimation based on Frequency-Domain Analysis with a Stereo Camera

2.1 Introduction

In recent years, a driver assistance system (DAS) technology has been actively developed to maximize both the safety and convenience of driving, and is eventually expected to facilitate autonomous driving. It utilizes various sensor technologies, such as cameras, radar, laser scanners and global positioning system (GPS), and it is anticipated that efficient technologies to retrieve valuable information from these sensors will be crucial in future autonomous driving systems [10].

In driving situations, determining the distance to objects in real time is important for various active safety applications, and many of them, such as emergency braking and smart cruise-control systems, rely on object detection and accurate distance measurements to objects. Thus far, distance measurements to objects are typically carried out using light detection and ranging (LIDAR) [11] and radar sensors but, because these sensors are expensive compared with camera sensors, many suppliers are investigating the development of camera-based distance measurement systems.

There is a method using the structure from motion (SFM) to calculate distances using a monocular camera [12]. In SFM, three-dimensional (3D) measurements can be achieved by acquiring images of objects while camera moves from one viewpoint to another. However, it has a problem that the absolute scale of objects must be computed in other ways, including by measuring the motion baseline or the size of an element in a scene, or by using other sensors like inertial measurement unit and GPS [13]. Nevertheless, a stereo camera is the only image-based sensor able to provide comprehensive 2D/3D information and many researchers and automotive companies have developed solutions for complete 3D environmental and object information using it [14]. A stereo camera can calculate distances to objects using parameters such as the baseline, focal length and disparity values (the difference between two images in pixel units) between the corresponding pixels in the left and right images. After determining all of the horizontal differences between corresponding pixels, a disparity map of the images can be obtained.

Stereo vision technology has initially been assessed in indoor situations [15][16] and extended to outdoor [17][18] for DAS in vehicles to obtain distance information, with a focus on providing the imagery necessary to compute fully dense depth maps. Many stereo-matching algorithms have been proposed in the stereo vision community and evaluated using online stereo camera datasets from Middlebury stereo evaluations [2] and the KITTI [19] vision benchmark. Although the quality of a disparity map determines the accuracy of the distance to objects, the procedure used to generate an accurate disparity map is quite time consuming. For accurate results on high-resolution images, computation can be prohibitively expensive with many of the top-ranked methods on the KITTI stereo benchmark, requiring multiple seconds or minutes of computation time per video frame [20][21]. In the vehicular environment, the position or colour of the light source usually changes. Moreover, matching between two images with a different intensity distribution is a more challenging problem due to lighting condition. Thus, to apply stereo-matching algorithms to these conditions, it is important to select suitable algorithms with illumination-invariant characteristics and which can be applied in real time with high accuracy levels.

In this chapter, we propose an accurate distance measurement algorithm for objects using a stereo camera. The features of the proposed algorithm can be summarised as follows:

1. Providing an accurate distance measurement: it provides more accurate distance measurement results than conventional algorithms while maintaining computation speed.

2. Providing a robust distance measurement in various situations: it performs well in diverse vehicular environments such as highways, urban and tunnel areas under various illuminations conditions.

The rest of this chapter is organised as follows. Related research studies are presented in Section 2.2. In Section 2.3, we describe the proposed algorithm and the frequency-domain analysis in Section 2.4. In Section 2.5, we present the cost optimization and distance estimation. The experimental results are presented in Section 2.6, and Section 2.7 contains conclusions concerning this study.

2.2 Related Works

Over the past few decades, researchers have developed algorithms for object distance estimations based on stereo-matching methods which compute 3D information instantaneously. The work in [2] presents a complete taxonomy of the approaches used for stereo disparity estimation. Stereo-matching methods can be divided into either local or global methods.

Local methods compute disparity values based on the local information around certain pixel positions. However, the computations used for the local method are simple but less accurate than those for the global method. The sum of absolute differences (SAD) and the sum of squared differences (SSD) are the most commonly used local parametric methods. In contrast to these, non-parametric approaches rely on the relative ordering of pixel values. An example of the latter is the census transform [23], which produces a bit string for the support window based on intensity comparisons. An improved version of the transform was used to determine stereo correspondence under varying illumination conditions [24].

Global methods consider an image in terms of cost values and use an optimisation process to determine disparities. These methods typically provide with highly accurate distance information, but, due to high computational complexity, they are not widely adopted in real-time vehicular applications. In [25], the authors addressed the pixel matching problem by trying to reduce the effect of varying colour between pixel pairs using histogram specification, and Ogale and Aloimonos [26] proposed a contrast invariant stereo-matching method on multiple spatial frequency channels. Efficient global optimisation techniques include graph cuts [27], belief propagation [28] and cooperative optimisation [29].

The semi-global matching (SGM) stereo algorithm, which contains features of both the local and global methods, is fast and efficient, minimising the disadvantages of global matching and ensures good accuracy at the same time. Furthermore, it shows excellent performance by employing a global optimisation process combined with various local optimisation results and is particularly useful for matching regions in two images which have high texture components. Disparities are determined by minimising a cost function which computes the absolute difference between the grey levels at a pixel position in two separate images. Cost aggregation based on a 1D path traversal simplifies the optimisation and ensures the constraints with respect to the explicit direction of the path. The final cost of each pixel and the resulting disparity are obtained by summing up the costs of the paths in all directions, and the final step finds the disparity that minimises the cost of each pixel [3].

In one study, Hermann and Klette [30] introduced iterative SGM (iSGM) as a new cost integration concept of SGM. In iSGM, the accumulated cost is evaluated iteratively to support a rapid analysis of the spatial disparity information cost. Spangenberg *et al.* [31] proposed an extended version of SGM based on the census transform, which is advantageous for outdoor scenes because it strengthens the smoothness constraints of SGM but must handle a large computational load; unfortunately, this hinders its application to vehicular environments. The researchers also proposed methods which improve the efficiency of SGM without special and additional hardware [32].

Several techniques, implemented in GPU and field programmable gate arrays (FPGAs), are able to generate disparity information from stereo videos in real time. For automotive and mobile applications, GPUs offer far higher computational throughput with the same power consumption than an equivalent CPU [33]. The method is able to generate disparity information from low-resolution video at a rate of 10 frames per second (fps) [34], and the approach which describes the FPGA implementation architecture of a SGM method is able to generate disparity maps from VGA images at a rate of 30 fps [35]. The paper presents a hardware-oriented disparity estimation algorithm that uses iterative refinement [36].

Despite good performance in a general situation, SGM-based methods have one crucial drawback of temporary memory requirement that depends on the number of pixels and the disparity range. Moreover, these methods are sensitive to the illumination condition in that they cannot handle local illumination differences due to lighting condition changes.

2.3 Algrorithm Description

2.3.1 Overall Procedure

In this section, the proposed distance-estimation algorithm for use in vehicular environments is described (Figure 2.1 illustrates the overall procedure of the proposed method). We initially detect the vanishing point and reduce the region of interest in the preprocessing stage, then process the given images in the frequency domain, and check the degree to which each frequency component value is related in the disparity maps, and finally, we calculate the distance of the target object. For reliable distance estimations in a vehicular environment, it is necessary to minimise the dependency on good illumination condition and the occlusion effect. The proposed algorithm uses a normalized cross-correlation (NCC) method to compensate for the difference in illumination between two images, and it reduces the occlusion/discontinuity effect using a high-pass filter to emphasise the object area. In the following section, each of these processes is explained in detail. Before we explain the details of our algorithm, we start by describing the basic concepts of stereo matching.

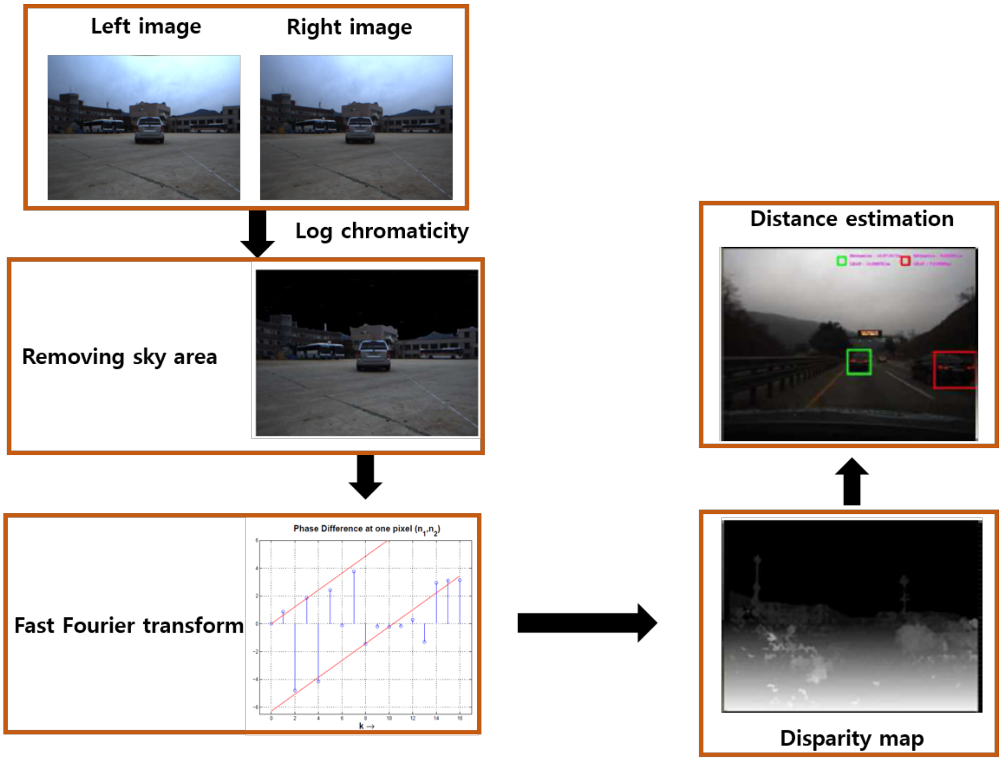


Figure 2.1: Processing flow of the proposed algorithm.

2.3.2 Preliminaries

In a stereo camera, distance information for objects is calculated using the baseline (the distance between the two monocular cameras), the focal length and the disparity between the pixels. If all distortion types are fixed through calibration and rectification, the two image planes of the camera are perfectly coplanar and the two optical axes are located in parallel, where the focal length and the principal point of the two cameras are identical. In addition, a specific row of an image corresponds to the same row of the other image via epipolar constraints. Under these assumptions, the disparity of the pixels can be determined using the horizontal difference between the corresponding points, as expressed in (2.1)–(2.3).

$$x_l = f\frac{X}{Z}, \tag{2.1}$$

$$x_r = f\frac{X - T_x}{Z}. \tag{2.2}$$

With equations, disparity d is determined by

$$d = x_l - x_r = f\frac{T_x}{Z}, \tag{2.3}$$

where f represents the focal length, T_x is the baseline and d denotes the disparity. Figure 2.2 illustrates the relationship of the disparity and depth with image coordinates of point using stereo camera.

2.3.3 Pre-processing

In the pre-processing stage, we reduce the search space by detecting the vanishing point and diminishing the region of interest. The vehicle travels towards the vanishing point in a driving situation. In other words, there is a close relationship between the

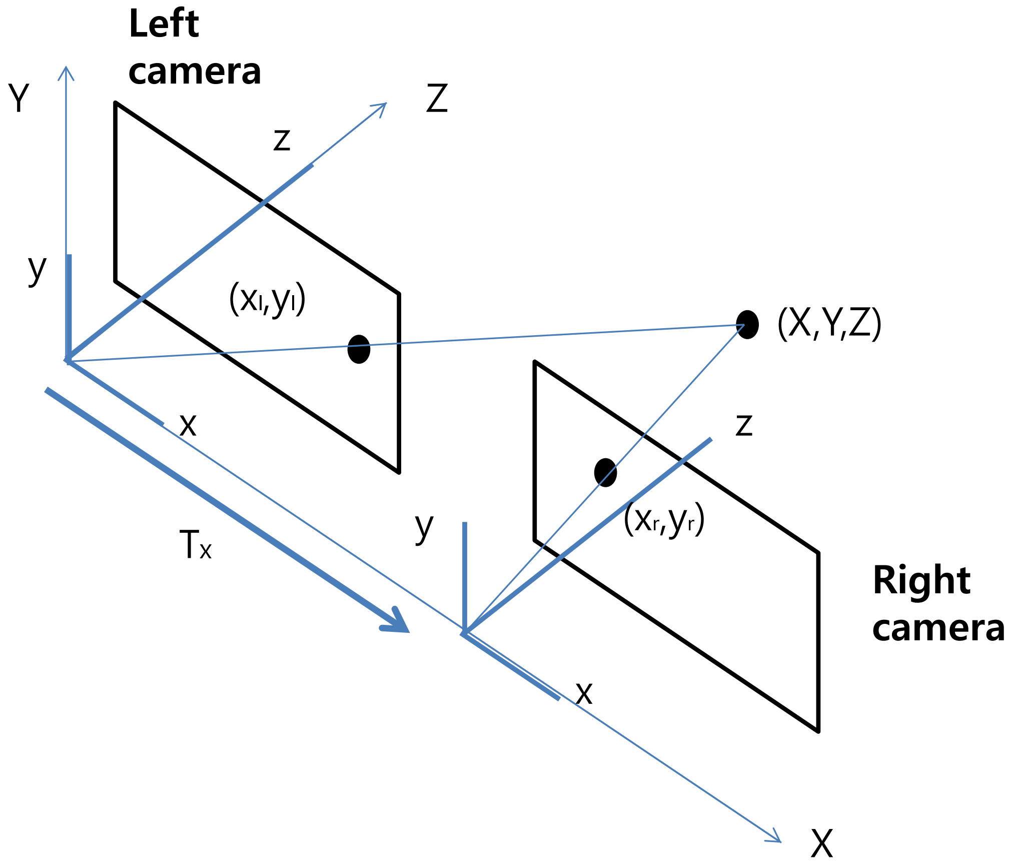


Figure 2.2: Stereo disparity and image coordinates of point.

vanishing point of the image and the distance to the object from the standpoint of the camera equipped in the ego-vehicle.

If the scene is simply modelled with a planar road and vertical obstacles, it has been shown that the v-disparity image is powerful for finding the relationship [30]. The vanishing point can be estimated using the orientation of the lane markers and curbs, and, for its calculation, we use the Canny edge detection algorithm [31] to find the edge line of the lane and to find the intersection point of the left and right lanes of the ego-vehicle.

In addition, we reduce the search space to enhance the computation speed when calculating the disparity of the pixels in a high-resolution image; for example, the upper section of the image may consist of a sky area, which is not meaningful for calculating distances in a vehicle environment and so can be discounted. To reduce the search space, we utilise the illuminant invariance characteristics of the log chromaticity colour space to detect the sky area and, once the chromaticity is known, its illumination effect can be successfully eliminated. The log chromaticity space can be converted from the RGB colour space as follows:

$$C_k = R_k / \sqrt[3]{\prod_{i=1}^{3} R_i}, \tag{2.4}$$

where C_k is the chromaticity, R_k is the RGB color value and k is one of the colour channels.

If we randomly choose a few seed pixels in the upper part of the image, we can find similar neighbour pixels belonging to the sky area using

$$I(x_s, y_s) - ld_{r,g,b} < I(x, y) < I(x_s, y_s) + ud_{r,g,b,}, \tag{2.5}$$

where $I(x, y)$ is the colour value of the pixel point (x, y), x_s and y_s are seed pixels and $ld/ud_{r,g,b}$ refers to the lower/upper differences in the colour value. Figure 2.3 and

2.4 illustrate the results of the pre-processing stage. After we apply the illumination-invariant characteristic of the chromaticity, the region of interest is reduced to nearly half of the original image size [39].

2.4 Frequency-domain Analysis

2.4.1 Procedure

In this section, we outline the frequency-domain analysis procedure, which is the core of the proposed algorithm. In the frequency domain, the image is represented by a combination of basis functions via a Fourier transform where the frequency component indicates the degree of change in brightness in the image (a high-frequency component means that brightness changes frequently occur). For example, when we analyse the image in the frequency domain in the vehicle environment, the road area is composed of low-frequency components, and the area including the foreground has multiple areas of high frequency.

$$F_n = \sum_{k=0}^{N-1} f_k e^{-j2\pi nk/N}, \tag{2.6}$$

$$f_k = \frac{1}{N}\sum_{n=0}^{N-1} F_n e^{-j2\pi nk/N}. \tag{2.7}$$

In these equations, n represents the frequency, k denotes the pixel index, f_k represents the intensity of the pixels and N represents the number of pixels used when calculating the Fourier transform. The equations show the discrete Fourier transform and the inverse discrete Fourier transform in a 1D signal. The Fourier transform $F(u,v)$ has a complex number value, including the real part $R(u,v)$ and the imaginary part $I(u,v)$. If we express this in the form of an exponential term, $F(u,v)$ is the Fourier spectrum and $\phi(u,v)$ is the phase angle using Euler's equation. Equation (2.8) is used

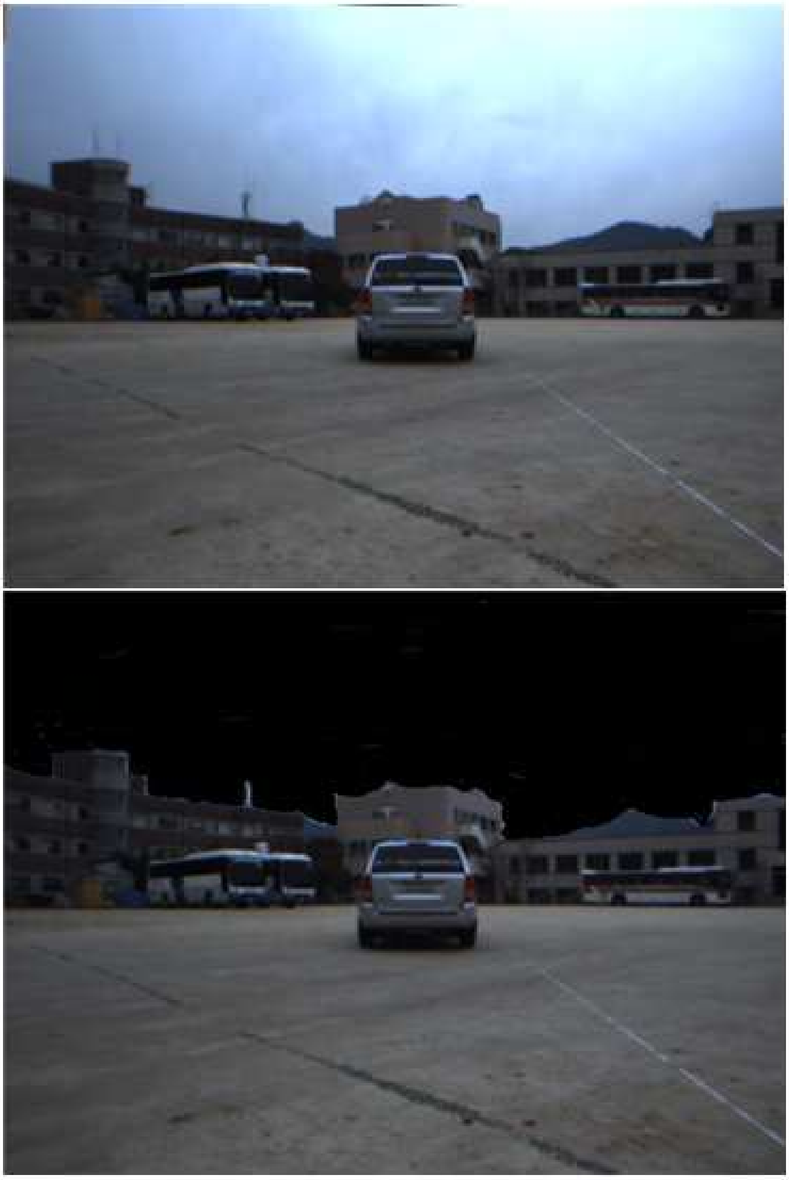


Figure 2.3: Comparison of an original and the same image with the sky removed (shown in black). Based on the image with the sky removed, the region of interest is reduce when generating the disparity map.

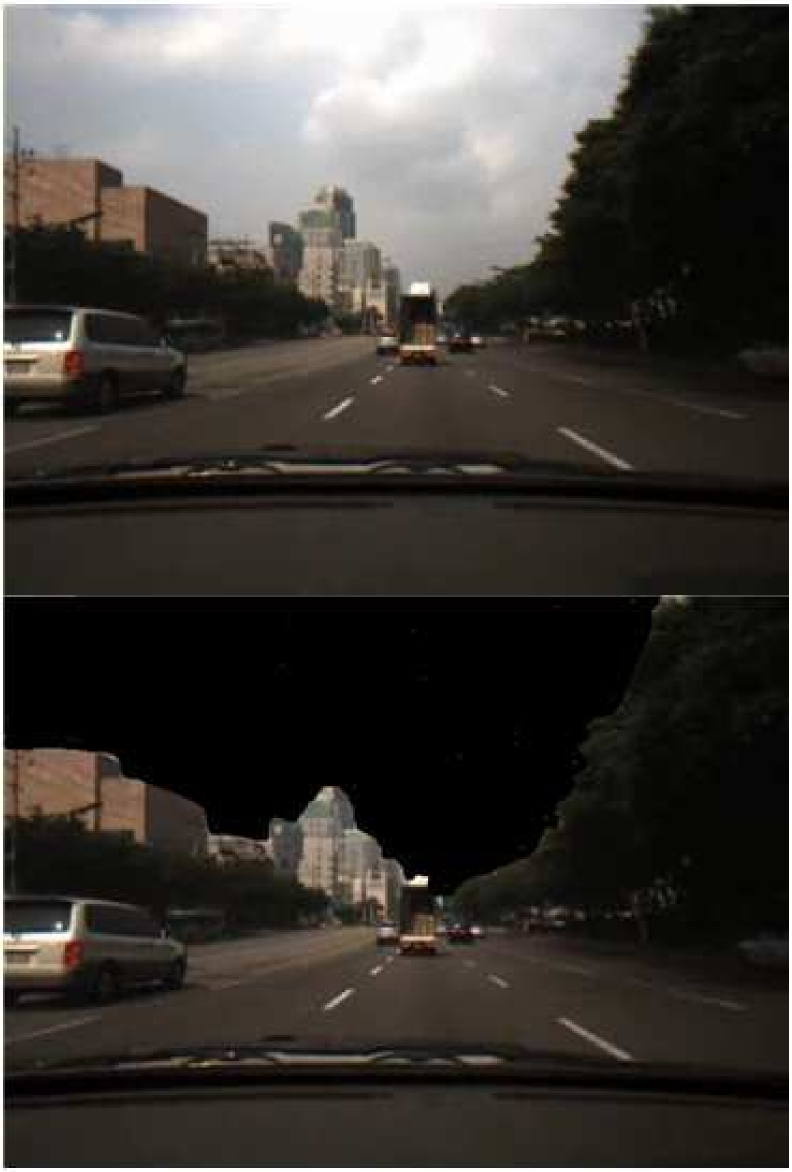


Figure 2.4: Comparison of an original and the same image with the sky removed (shown in black). Based on the image with the sky removed, the region of interest is reduce when generating the disparity map.

to convert the form of the exponential term from the discrete Fourier transform

$$\begin{aligned} f(n-d) \longleftrightarrow & F_n(u,v)exp(-j\phi_d(u,v)) \\ & = |F_n(u,v)|exp(j\phi_n(u,v))exp(-j\phi_d(u,v)), \end{aligned} \tag{2.8}$$

where $F_n(u,v)$ represents the 2D Fourier transform including the magnitude and the phase component. The translation and shift properties of the Fourier transform are expressed as

$$\begin{aligned} F_n(u,v) = & \sum_{k=0}^{N-1} f_k e^{-j2\pi nk/N} \\ = & |F_n(u,v)|exp(j\phi_n(u,v)), \end{aligned} \tag{2.9}$$

where d represents the disparity. The disparity value can be obtained directly from the frequency domain using the phase value of the Fourier transform. By utilizing the characteristics of the Fourier transform as described above, the process of calculating the disparity information proceeds as follows. As there is a horizontal disparity between the left and right images in the calibrated camera set, we can apply the translation and shift properties of the Fourier transform to the stereo-matching algorithm

$$I_L(n) = I_R(n-d), \tag{2.10}$$

$$\begin{aligned} & |I_L(K)|exp(j\phi_L(K)) \\ & = |I_R(K)|exp(j\phi_R(K))exp(-j\phi_d(K)). \end{aligned} \tag{2.11}$$

In this equation, I_L and I_R , respectively, represent the intensity values of the left and right images. Furthermore, (2.10) is the Fourier transform of (2.11). In fact, the intensity values of two images are unlikely to match at a high probability in an outdoor environment. Therefore, we consider the magnitude cost to compensate for the error of the magnitude spectra as

$$M(K) = \frac{|I_R(K)|}{|I_L(K)|}, \tag{2.12}$$

$$exp(j\phi_d(K)) = M(k)exp(j(\phi_R(K) - \phi_L(K))), \tag{2.13}$$

where $M(K)$ denotes the ratio between the left and right spectra. $M(K)$ does not affect the calculation of the imaginary part of (2.13), and so we consider $M(K)$ as the weighted factor with which to calculate the disparity value

$$\theta_L(k) = \arg\max_{\phi_L(k)} |I_L(K)|, \tag{2.14}$$

$$\theta_R(k) = \arg\max_{\phi_R(k)} |I_R(K)|. \tag{2.15}$$

Equations (2.14) and (2.15) compute the phase value which has the spectrum with the maximum magnitude. The original intensity values of the two images are not identical, but the intensity distribution is the same if the surrounding pixels are similar. In order to measure the correspondence more precisely, several maximum magnitude spectra are needed

$$\begin{aligned} \phi_d(K) = M(k)(\phi_R(K) - \phi_L(K)) \\ +(1 - M(k))(\theta_R(k) - \theta_L(k)), \end{aligned} \tag{2.16}$$

$$d(K) = \phi_d(K) \times \frac{N}{2\pi k}, k = 0, 1, ..., N - 1. \tag{2.17}$$

If the magnitude spectra $I_L(K)$ and $I_R(K)$ are nearly identical, the phase term has a strong influence. Equation (2.17) shows that the disparity can be calculated by means of a frequency-domain analysis. Moreover, the advantage of this method is that the disparity can be calculated directly from the phase value.

The discrete Fourier transform takes $O(N^2)$ times to calculate N frequencies, and a complex number of multiplications should be performed N times in order to calculate each frequency in an N-point 1D discrete Fourier transform. We implement the fast Fourier transform method to accelerate the calculation of the discrete Fourier transform of the high-resolution image (the fast Fourier transform rapidly computes such transformations by factorising the matrix into a product of sparse factors); if we use this method, the computation time can be reduced to $O(NlogN)$ [41].

2.4.2 Contour-based Cost Computation

In general, the method of stereo matching defines the cost to compute the disparity of the corresponding points in the two images. The SAD and SSD are representative methods which can be used to measure the degree of similarity between two images. If the cost value of the SAD or SSD with regard to two patches is low, they can be viewed as similar by considering a threshold value. Therefore, the similarity threshold value and the size of the patch are important factors in determining the performance of a patch-based stereo-matching algorithm. However, when using the stereo-matching algorithm in an outdoor environment with a stereo camera, it is difficult to discriminate the same part of the patch between the left and right images resulting from computation with SSD or SAD when there is a difference in brightness between the two cameras.

To compensate for this problem, we use the NCC method whereby a normalisation process is performed to make the average brightness level zero and the standard deviation level one by calculating the brightness for each of the patches. This places more emphasis on the brightness difference in a pixel as compared with the overall patch or image. Aligning the average brightness at zero has a significant effect when there is a large difference between the average brightness of two patches. Furthermore, giving the standard deviation of the brightness a value of one compensates for the effect of a difference in contrast between two patches even if they have the same average brightness value [42].

The cost of the NCC method is expressed as

$$Cost_{NCC} = \frac{1}{N}\sum_{x,y}\frac{(I_1(x,y) - m_1)(I_2(x,y) - m_2)}{\sigma_1\sigma_2}, \tag{2.18}$$

where N denotes the number of the pixels, m represents the average brightness, σ is the standard deviation and I represents the intensity of the pixel. In a vehicle environment with high-resolution images, a considerable amount of computation time is required to apply the NCC method to all of the image pixels. In this paper, we apply the NCC method to areas where the brightness changes frequently, such as corner points or the outlines of objects, using a high-pass filter in the frequency-domain area. If we restore signals having a high frequency in the image, the boundaries of the objects will be emphasised. Next, we apply the NCC method to the emphasised area using a high-pass filter and, in our case, a high-pass Gaussian filter was used (Figure 2.5 and 2.6 illustrate the resulting image after using this technique). The area filtered through the high-pass Gaussian filter is calculated by

$$H(u,v) = 1 - exp(-D(u,v)^2/2D_0^2), \tag{2.19}$$

where D_0 represents the radius and $D(u,v)$ denotes the frequency area. If we apply a large value of D_0, the area is increased relative to the edge region.

2.5 Cost Optimization and Distance Estimation

2.5.1 Disparity Optimization

The disparity map is generated based on the content described above. We enhance the calculation speed using the fast Fourier transform and the high-pass filter in the frequency domain to reduce the region of interest and for optimisation, we use the simple winner-takes-all disparity selection strategy. We improve the accuracy using

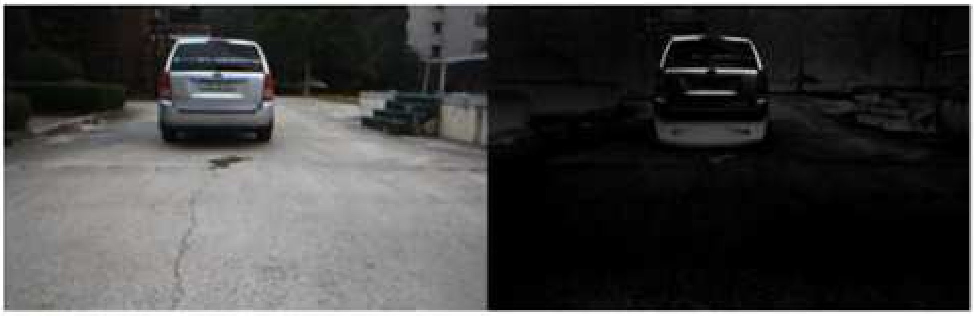


Figure 2.5: Original image and the result image through the high-pass Gaussian filter. The pixel value gets brighter around the boundary of the object.

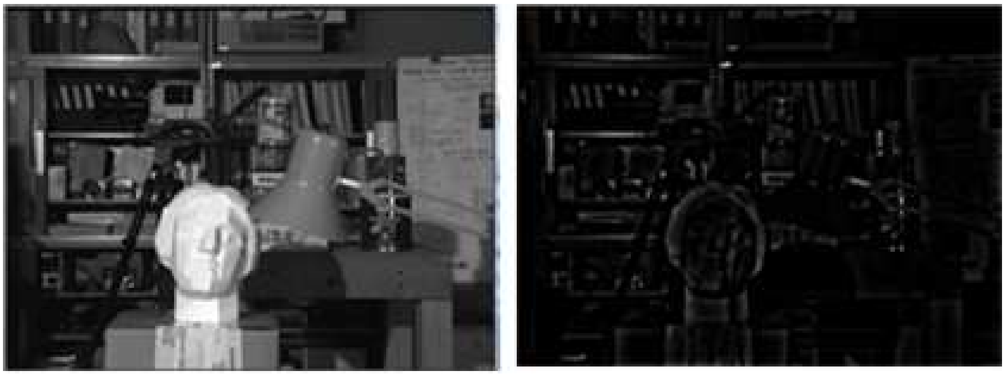


Figure 2.6: Original image and the result image through the high-pass Gaussian filter. The pixel value gets brighter around the boundary of the object.

the NCC method, which reduces the effects of the difference in illumination between the left and right cameras

$$min_d D_{pixel}(u, v, d) = \alpha D_{NCC}(u, v) + (1 - \alpha) D_{phase}(u, v). \tag{2.20}$$

In this equation α is a weight factor determining whether the pixel is located on the boundary of the object or not. If the pixel is located on the boundary of the object, the value of α is close to one

2.5.2 Post-processing and Distance Estimation

In this paper, we apply a bilateral filter [43] using the standard deviation of a Gaussian function, with the weighted average of the pixel values and distances from the kernel centre to remove erroneous pixels. In addition, the filter does not weaken the sharpness of the edges, which is an advantage of a Gaussian filter.

The bilateral filter is given by

$$BF[I]_p = \frac{1}{W_p} \sum_q G_{\sigma_s}(||p - q||) G_{\sigma_r}(|I_p - I_q|) I_q, \tag{2.21}$$

where W_p represent the normalisation factor, G_{σ_s} is the space weight and G_{σ_r} represents the range weight.

Finally, we estimate the distance to the target object using the baseline, the focal length and the disparity value pertaining to the pixels, as explained in the previous section.

2.6 Experimental Results

2.6.1 Test Environment

In this section, we present experimental results from a vehicular environment as described in Section 2.3 to 2.5. Our C++ implementation requires 10 fps to estimate the target object distance on a standard PC and so, in order to evaluate the performance of the proposed algorithms, it was tested on a PC with an Intel Core i7-4770 CPU at 3.40 GHz with 8.00 GB of RAM. We tested video clips taken with a stereo camera at a resolution of 640 by 480 at 48 fps and collected datasets on a highway during the daytime (each dataset consisted of 10,000 frames). In order to evaluate the performance of the proposed algorithm, we obtained data from LIDAR as the ground truth (LIDAR is a type of active sensor that detects the location of objects by emitting light at a known frequency and measuring the return-trip time of flight). We used an outdoor LIDAR model which measures distances up to 65 m and has a field of view of 190 degrees. For LIDAR, the error is 2.5 cm for a range of 1–10 m, 3.5 cm for a range of 10–20 m and 5 cm for a range of 20–30 m. We adopted the theoretical distance measurement error from the datasheets provided by the manufacturer. Fig. 2.7 shows the SNU autonomous vehicle equipped with LIDAR and a stereo camera. Next, all of the recorded data from both sources were synchronised. To merge all datasets including LIDAR and camera, we used the CAN bus as the master software synchronisation bus for exchanging timestamps [44].

We also evaluate our algorithm on KITTI visual benchmark suite (particularly the object subset) and public Middlebury dataset. The KITTI dataset is an open-source dataset created by Dr. Andreas Geiger *et al.* for outdoor autonomous driving application. The object sub-dataset in KITTI is captured by Velodyne HDL-64E laser scanner and two highly-calibrated binocular cameras. Each image captured by camera is registered to a set of 3D laser point cloud, which means that ,for each image, we can get one corresponding depth map by projecting the 3D laser point cloud to the image plane. The KITTI object dataset is preliminarily created for outdoor objects detection, like cars, trucks, cyclist. Therefore, we concentrate on this dataset for its capability to estimate the distance of these objects in the depth map. Due to the lack of ground truth dataset, we only provide visual experiment result. For quantitative evaluation, we further test our algorithm on Middlebury dataset. The Middlebury dataset is captured in indoor environment, it contains six different types of scenes of daily life: moebius, dolls, laundry, books, art and reindeer.

The proposed method was compared with the SGM-based method [3], which is a conventional algorithm used in outdoor environments. To investigate the performance over the distance range, we experimented with various scenarios, including a highway, an urban area and a tunnel, as shown in Figure 2.8.

2.6.2 Experiment on KITTI Dataset

KITTI object dataset is a great dataset to test our algorithm because it encompasses all kinds of street view in automated driving vehicle. The objects in this datasets span to various categories, locations, orientations. We show some results in Fig. 2.9. The ground truth disparities for the test set are withheld and an online leaderboard is provided where researchers can evaluate their method on the test set. Submissions are allowed once every three days. Error is measured as the percentage of pixels where the true disparity and the predicted disparity differ by more than three pixels. Translated into distance, this means that, for example, the error tolerance is 3 centimeters for objects 2 meters from the camera and 80 centimeters for objects 10 meters from

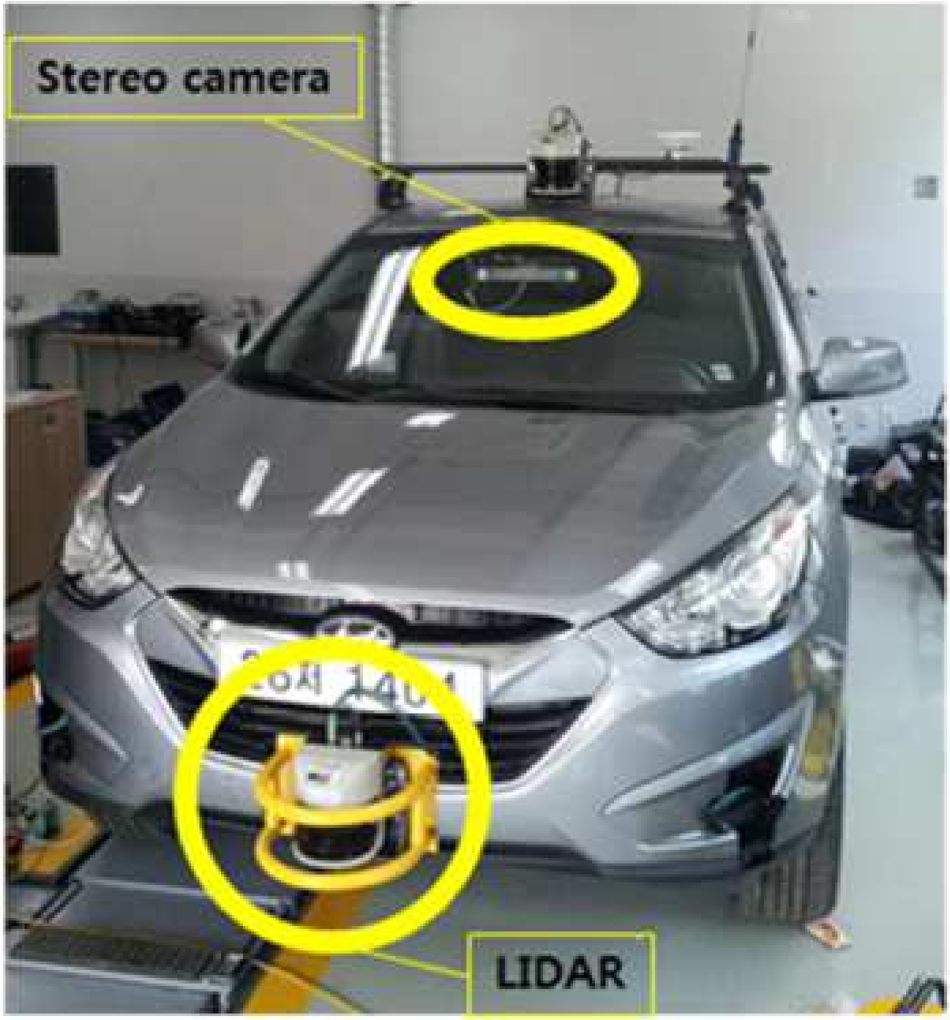
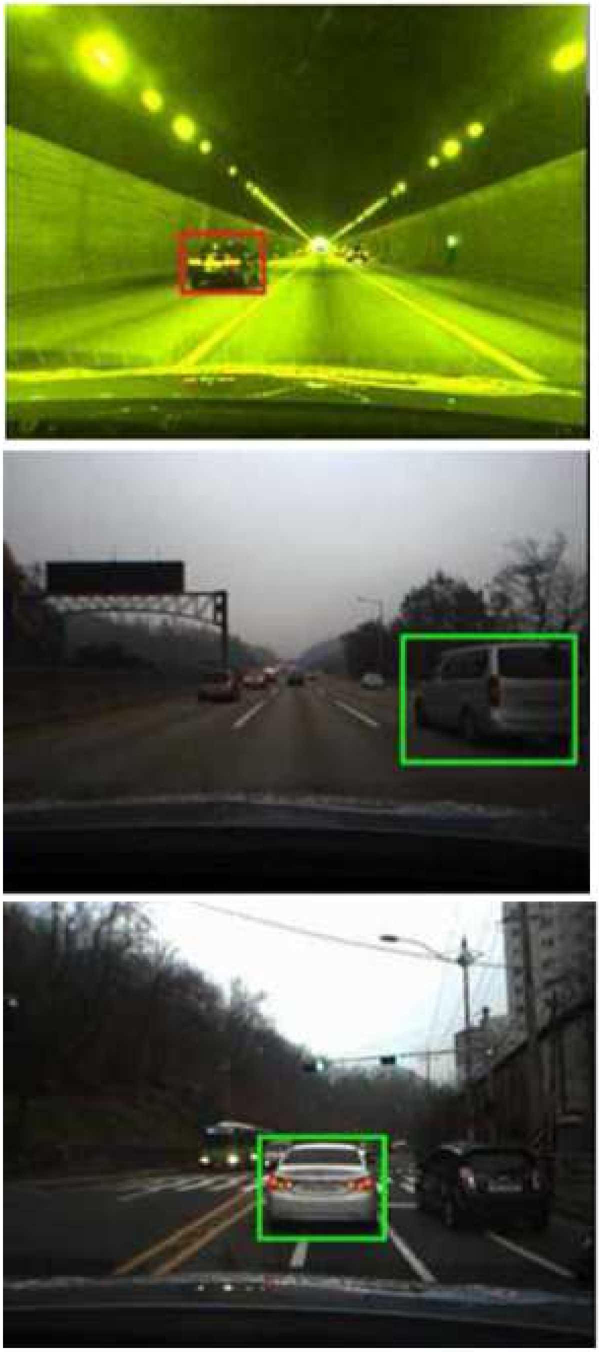


Figure 2.7: SNU autonomous vehicle. The vehicle is equipped with a stereo sensor (indoor), LIDAR (outdoor).

Figure 2.8: Experimental situations. We tested the algorithm in various environments (tunnel, urban area, highway).



the camera.

Two KITTI stereo data sets exist: KITTI 2012 and, the newer, KITTI 2015. For the task of computing stereo they are nearly identical, with the newer data set improving some aspects of the optical flow task. The 2012 data set contains 194 training and 195 testing images, while the 2015 data set contains 200 training and 200 testing images. There is a subtle but important difference introduced in the newer data set: vehicles in motion are densely labeled and car glass is included in the evaluation. This emphasizes the method's performance on reflective surfaces.

Method	Author	Setting	Error
SGM+C+NL	Hirschmuller(2008); Sun *et al.*(2014)	F	5.79
SGM+LDOF	Hirschmuller;Brox and Malikk(2011)	F	6.24
SGM+SF	Hirschmuller(2008);Hornacek *et al.*(2014)	F	6.84
OCV-SGBM	Hirschmuller		10.86
PROPOSED	Yoo		5.31

Table 2.1: Experimental results on the KITTI 2015. The setting column provides insight into how the disparity map is computed: F indicates the use of optical flow. The error column reports the percentage of misclassifed pixels and the runtime column measures the time, in seconds, required to process one pair of images.

2.6.3 Performance Evaluation and Analysis

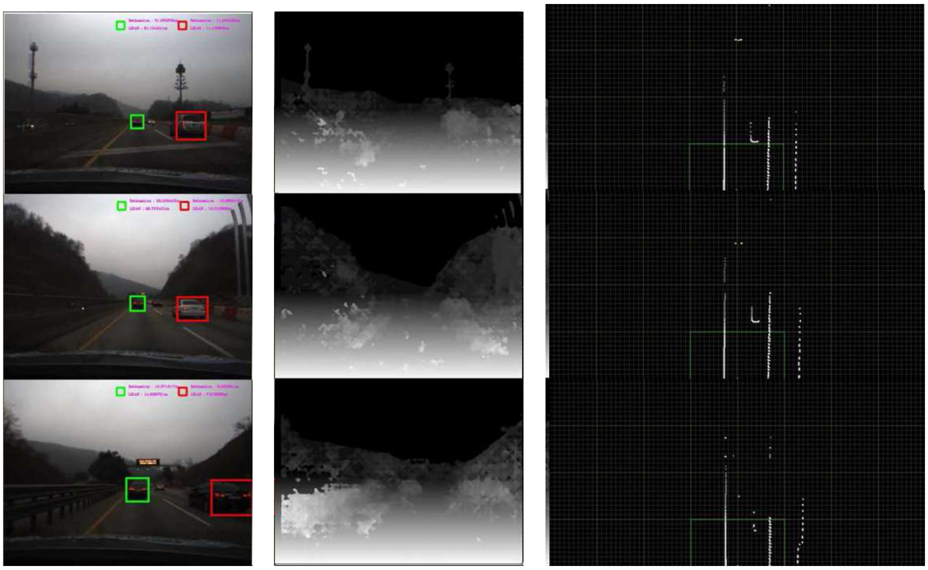
In the following paragraphs, we present the results of several experiments. For estimating a distance to target, we considered a vehicle in the view of the stereo camera. Fig. 2.8 shows the various situations: a tunnel, an urban area and a highway. The highway scenario had a good illuminative environment under constant brightness conditions, the tunnel scenario included a tunnel entrance and the exit had mid-bright conditions with illumination variance, and the urban area has illumination variance because the target vehicle was often shaded by buildings or trees. The distance accuracy of each method is listed in Table 2.2.

When used for measuring the distance to a target object, neither algorithm produce errors within 20 m compared with the LIDAR system. However, the proposed algorithm provided more accurate distance measurements than the SGM-based algorithm when the distance exceeded 20 m; errors occurred when using the latter algorithm because the colour information was insufficient for objects in the image over this distance. The SGM-based algorithm performs scanline optimisation (cost aggregation) in different aggregation directions, thus each pixel is influenced only by pixels located on eight horizontal, vertical or diagonal lines. Since the distance is inversely proportional to the disparity and the number of pixel is limited, it is hard to compute the distance of objects when they are more than 20 m away. The SGM-based algorithm requires only two parameters which are the matching penalties used for every path whereas the proposed algorithm compensates for the error by using phase-based cost computation.

Figure 2.9(a) show the target vehicle with green and red box. The vehicle detection algorithm is not applied in this paper but we draw the rectangular in the image for focusing on measuring the distance using the disparity map shown as Figure 2.9(b). Figure 2.9(c) shows the 2D Local map which shows the front environment of the ego-vehicle using laser scanner.

The performance of the proposed algorithm was well maintained with small errors in various scenarios when the distance exceeded 20 m because it uses contour information and NCC-based cost computation and so provides robust distance measurements in various situations. However, the SGM-based algorithm showed a larger distance error in the tunnel and urban scenarios than the highway scenario. The SGM-based algorithm uses hierarchical mutual information to find the correspondence by utilising the joint histogram of intensities between two input images, but local illumination variations due to brightness changes cannot be handled by mutual information based cost computation. The computational complexity of the proposed algorithm takes on average 25 fps at a resolution of 640 x 480.

The proposed algorithm calculated distance nicely in most cases. However, we did observe that errors appeared at the tunnel entrance probably because the lighting conditions momentarily became a dark-bright pattern when entering the tunnel. In addition, due to reflection and irradiation coming from the tunnel lamps, we observed a performance degradation of 9.7 percent. When we compared our and LIDAR results, the error occurred when the intensity distribution was different between the left and right images. In the urban case, there was a lot of street furniture such as streetlights and tree trunks, and these items generated shadow areas and a different intensity distribution. Colour consistency can enhance the performance of stereo matching, while accurate disparity maps can improve the colour consistency or constancy. Our method performed well with illumination changes to a large extent, although some limitations exist when applied in severe shadow regions. In future work, this limitation could be further resolved by incorporating techniques like intrinsic image decomposition methods [45].



(a) The original image (b) The disparity map (c) LIDAR data(Ground truth)

Figure 2.9: The experimental results. The original image(a) with target vehicle presented by green and red box. Distance measurement results are written in upper side of the image. the Disparity map(b) and the LIDAR data(c) is generated in each frame. All data was synchronized.

Distance	Built-in camera	SGM	SGM+C+NL	SGM+LDOF	SGM+SF	Yoo
5-10 m	0.9	0.8	0.7.	0.5	0.4	0.7
10-15 m	1.1	1.0	0.8	0.7	0.5	0.9
15-20 m	1.5	1.4	0.9	1.5	1.0	1.1
20-25 m	1.9	1.8	1.3	1.5	1.5	1.4
25-30 m	2.5	2.2	1.9	2.1	1.8	1.7
30-35 m	4.1	3.5	2.5	2.6	2.3	1.9
35-40 m	7.0	5.9	3.4	3.2	3.3	2.1

Table 2.2: Distance accuracy comparison according to images.

2.7 Conclusion

In this paper, we proposed a new method for measuring distances based on frequency-domain analysis using a stereo camera. By analysing the frequency-domain information, the depth value of an object can be obtained accurately in less time than the existing comparable methods. Experimental results show that the proposed method significantly improves the accuracy of distance measurement. The advantage of the proposed method is that the disparity value can be calculated directly through the phase of the frequency domain. When conducting our experiments in various scenarios, the proposed algorithm performed 10 percent better on average than the conventional method. Our method was also shown to be suitable for use in real-time applications such as autonomous vehicles. In future work, we hope to improve the operation speed while maintaining the accuracy. We intend to develop object detection and tracking algorithms, such as a Kalman filter [46] and a particle filter [47], to integrate the system.

Chapter 3

Visual Map Matching Based on Structural Tensor and Mutual Information using 3D High Resolution Digital Map

3.1 Introduction

In recent years, there has been huge interest in automated vehicles with driver assistance systems. Many vehicle companies and research groups have been actively involved in maximizing safety and convenience of automated vehicles. Vehicle localization is the fundamental application that determines a vehicle's position and is essential for vehicle control. Global Positioning System (GPS) based navigation devices have gained popularity, with many vehicles now factory equipped with GPS systems and a variety of portable devices also commercially available. However, such devices have difficulty localizing in situations where the GPS signal is unavailable or insufficiently accurate. Using digital maps with accuracy in the range of a few cm, it is possible to develop sensor based localization without relying on GPS.

Fusing visual information with a digital map to improve vehicle global localization accuracy and overcome GPS problems has attracted much research and development attention in recent years. Sensors for vehicle localization without GPS are categorized into cameras and light detection and ranging (LIDAR) systems. Three dimensional (3D) LIDAR is the most technically appropriate method to acquire centimeter level road geometry, since it can provide accurate 3D information about roads. However, 3D LIDAR is expensive to set up in a vehicle. On the other hand, camera sensors provide highly reliable color information about the ambient environment, and are price competitive and light enough to be set up in a typical passenger vehicle.

The goal of this paper is to estimate the vehicle position in a previously scanned environment using in-vehicle camera images. This problem has been recently reported by researchers in perspective vision, exploiting normalized mutual information (NMI) [9] or point features [48]. Vision based approaches depending on visual features are often sensitive to illumination changes and environmental modifications, and the vision processing technique(s) employed to provide accurate position information are crucial. This paper separates foreground and background objects, preserving local features and removing dynamic objects. We represent the vehicle localization system using structure tensor and mutual information methods based on a high-precision 3D digital map. The objective is to optimize a similarity function between the camera image and an image from the reference map. The proposed method achieves decimeter level accuracy for vehicle localization.

Related research works are presented in Section 3.2. Section 3.3 describes the proposed algorithm for digital map generation, calibration, dynamic object removal, improving illumination, and visual localization using structural tensors and mutual information. Experimental results are presented in Section 3.4, and our conclusions and closing remarks are in Section 3.5.

3.2 Related Work

The problem of map localization has been traditionally approached using Monte Carlo methods [49]. Given a map of the environment, the algorithm estimates vehicle position and orientation as it moves and senses the environment. In the DARPA Grand Challenge (2005) and Urban Challenge (2007), vehicle localization using Monte Carlo methods was used in the GPS denial areas. Pose estimation comprises a 2D position (x,y) and yaw angle for 2D map localization.

In computer vision, place recognition methods attempt to localize [50] an image, given a large database of georeferenced images. If the initial position is known, visual odometry [51] provides relative motion estimates that determine the position and orientation of a vehicle by analyzing associated camera images. However, error is compounded when the vehicle operates on non-smooth surfaces and location becomes increasingly unreliable as these errors accumulate and compound over time.

Simultaneous localization and mapping (SLAM) has been intensively studied in the robotics community. SLAM provides localization with map building using reflectivity information obtained with vision sensors, LIDAR, and an estimator, e.g. extended Kalman (EKF) or Rao-Blackwellized particle (RBPF) filters.

Localization in road network maps has also been approached using map matching techniques [52], which can efficiently localize a query map within a larger map region.

A standard scheme to visual SLAM consists of first extracting a sufficiently large set of features and robustly matching them between successive images. These features are then input to the joint process of estimating the camera pose and scene structure [53]. The majority of visual SLAM methods fall into this class independent of the applied filtering technique, e.g. EKF-SLAM [54] and FastSLAM 2.0 [55]. However, these features frequently vary with time of day and weather conditions, and cannot be used without an intricate observability model.

Many studies have considered map assisted visual localization, largely focused on matching photometric characteristics of the environment either by comparing image feature descriptors, e.g. SIFT [56] or SURF [57], or directly operating on image intensity values. However, one of the main issues in visual localization is that the environmental photometric appearance changes substantially over time, particularly across seasons [6].

Mapping with different sensor modalities can be a useful addition. In contrast to methods based on matching photometry, approaches for camera localization in geometric maps built from LIDAR data are less present in the literature. Wolcott *et al.* [9] and Foster *et al.* [59] have been leaders in this approach. Wolcott *et al.* [9] proposed a method to localize an autonomous vehicle in urban environments. Using LIDAR intensity values, they render a synthetic view of the mapped ground plane and match it against the camera image by maximizing normalized mutual information. While this approach provides the 3 degrees of freedom (DoF) pose, Pascoe *et al.* [60] proposed a system that estimates the full 6 DoF camera pose. Their appearance prior map combines geometric and photometric data to render a view that is then matched against the live image by minimizing the normalized information distance. Both approaches perform matching in 2D space, which require expensive image rendering supported by GPU hardware. Furthermore, they did not consider illumination effects or dynamic objects, which degrade practical localization performance.

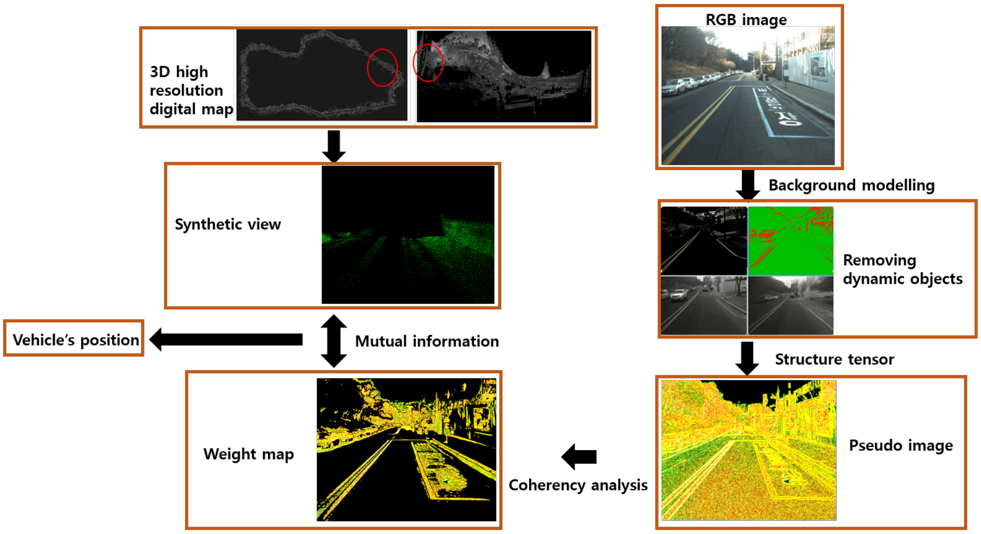


Figure 3.1: Proposed system architecture.

3.3 Methodology

In this section, the proposed methodology is described.(Figure 3.1 illustrates the overall procedure of the proposed method). We initially generate a high precision 3D digital map, then perform an extrinsic calibration between LIDAR and monocular cameras. We remove dynamic objects and apply an illuminant invariance algorithm to improve accuracy, then apply the proposed visual localization using a structure tensor and mutual information algorithm. Each of these processes is explained in detail in the following sections.

3.3.1 Sensor Calibration

Given the use of the LIDAR map data and a monocular camera, it is crucial to perform extrinsic calibration between these to perform a localization. Pinhole model is the mostly used model to represent a camera projection process. Three coordinate systems are considerd(world system, camera frame, image frame). A camera has intrinsic and extrinsic parameters. The intrinsic parameters are related to its intrinsic characteristics, including focal length, position of the principle point on image plane, image pixel size, scaling factors of row and column pixels, skew factor, and lens distortion. The extrinsic parameters are related to its position and orientation with respect to a fixed world system.

For a pinhole camera, the relationship between a 3D point, X_i, and its homogeneous image projection, $\tilde{x}_i$, is

$$\tilde{x}_i = K[R|t]\tilde{X}_i, \tag{3.1}$$

where the extrinsic parameters, R and t, are the orthonormal rotation matrix and translation vector, respectively, relating the laser and camera coordinate systems; and K is the camera intrinsic matrix. Generally, R is parametrized by the Euler angles. The rotation matrix is an orientation matrix of a camera in the world system is related to its rotation from the world frame to the camera frame. The rotation matrix R in 3-dimensional space can be decomposed into three rotation matrices : R_X with angle α around X_C axis, R_Y with angle β around Y_C axis, and R_Z with angle γ around Z_C axis. They are respectively written as :

$$R_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}, \tag{3.2}$$

$$R_Y = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}, \tag{3.3}$$

$$R_Z = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.4}$$

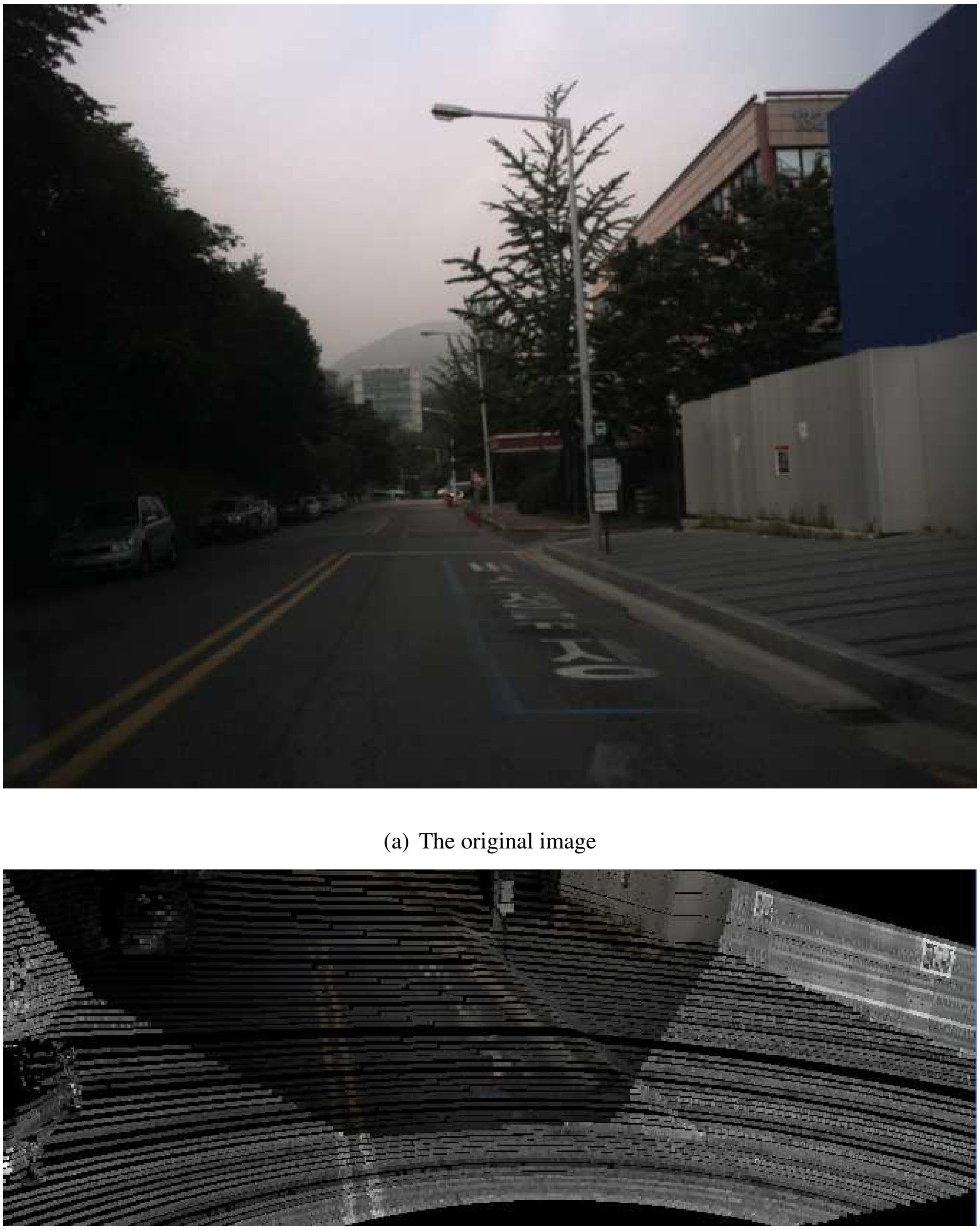
Then, the full rotation matrix R is given by the product of these three matrices.

We maximize the mutual information by registering reflectance values obtained from a Velodyne 64 beam LIDAR to camera pixel intensities. We also consider the laser reflectivity of a 3D point and the corresponding grayscale of the image pixel, using the marginal and joint probability distribution for LIDAR reflectivity and grayscale intensity, respectively, calculating the rotation and translation matrix between the LIDAR and camera, as shown in Fig 3.2.

The synthetic view includes the 3D information and laser reflectivity values. The views were generated by varying longitudinal and lateral translation around the image.

3.3.2 Digital Map Generation and Synthetic View Conversion

For accurate control and localization of an automated vehicle, a high precision 3D map is fundamental. We acquire 3D point cloud data utilizing a 3D mobile laser scanning sensor. During the data acquisition, we collect the 6D vehicle pose data and 3D point cloud data. The vehicle pose data includes the accurate 3D global position and 3D attitude with respect to the trajectory of the vehicle. For a high-precision vehicle positioning system, we use a RTK-GPS and a high-precision INS. For making a high-precision map, we apply a ground extraction to each frame to remove unnecessary points. The ground points for every frame are accumulated on a global coordinate system using 6D vehicle pose data synchronized with LIDAR. The 3D point cloud data is represented on a vehicle coordinate system, and a rigid body transformation is defined by a transformation matrix. The 3D point cloud is accumualted on the global coordi-



(a) The original image

(b) The perspective view of input image

Figure 3.2: Original image and extrinsic calibration showing a perspective view with camera sensor data projected into the 3D LIDAR points.

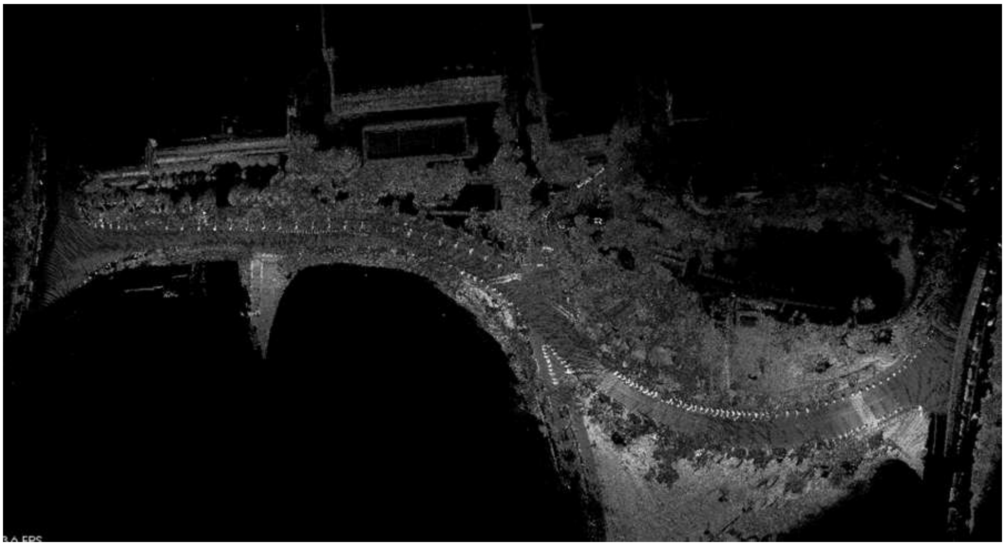


Figure 3.3: The example of the map data.

nate system by the rigid body transformation and dense point cloud data representing the ground region is obtained[61]. Figure 3.3 shows the example of the map data.

After generating the digital map, we create a synthetic view, as shown in Figure 3.4 for calculating the matching cost comparing with the camera image. We generate the candidate images by varying longitudinal and lateral translation around the optimally aligned image.

3.3.3 Dynamic Object Removal

When we use the high-precision 3D digital map for localization, it is essential to remove dynamic objects, because the base map does not include these, such as vehicles, pedestrians, etc. As more obstacles overtake the image, the algorithm can be distracted and lead to erroneous registrations. To remove the object, we first generate the background model to compare with input image data. We consider the accumulation factor to separate background and foreground objects.

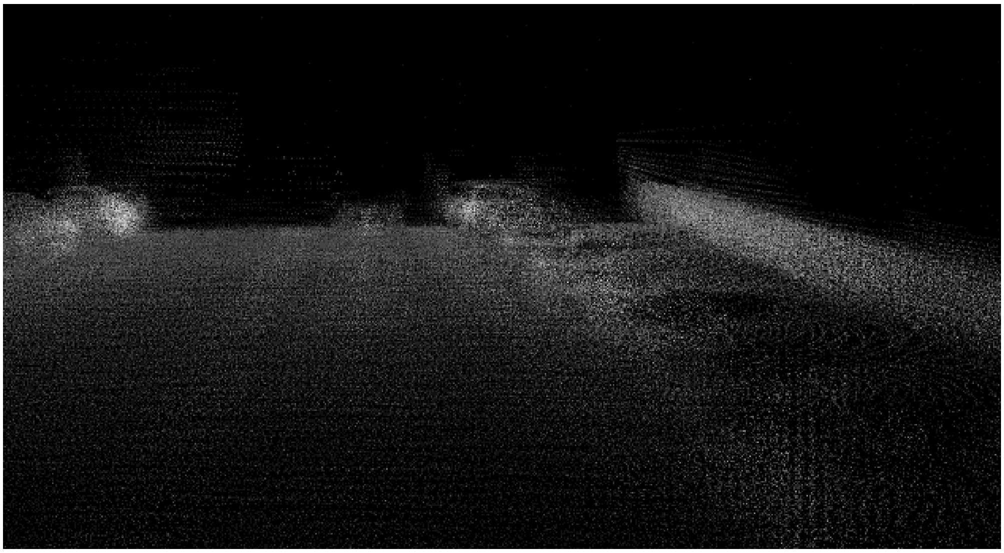


Figure 3.4: Example of synthetic view.

$$\mu_i^t = \frac{\alpha_i^{t-1}}{\alpha_i^{t-1}+1}\tilde{\mu} + \frac{1}{\tilde{\alpha}_i^{t-1}+1}M_i^t, \tag{3.5}$$

$$\sigma_i^t = \frac{\tilde{\alpha}_i^{t-1}}{\tilde{\alpha}_i^{t-1}+1}\tilde{\sigma}_i^{t-1} + \frac{1}{\tilde{\alpha}_i^{t-1}+1}V_i^t, \tag{3.6}$$

$$\alpha_i^t = \tilde{\alpha}_i^{t-1} + 1, \tag{3.7}$$

where $|G_i|$ is the number of pixels, I_j^t is pixel intensity, α_i^t is an accumulator factor, μ_i^t is the mean pixel intensity, σ_i^t is the pixel variance,

$$M_i^t = \frac{1}{|G_i|}\sum_{j \in G_i} I_j^t, \tag{3.8}$$

and

$$V_i^t = \max_{j \in G_i}(\mu_i^t - I_j^t)^2. \tag{3.9}$$

After obtaining the background model for time t, we remove dynamic objects that have a large difference from the input image, then apply the saliency map model [64] to extract the features. The saliency map considers intensity, orientation, and color to compute a unique quality for each pixel. Applying this algorithm, we can preserve lane, road markers, tree trunks, poles, etc., which are key information for comparing with a prior map. Figure 3.5 shows an example result removing dynamic objects.

3.3.4 Illuminant Invariance

The camera is sensitive to illumination conditions, and provides relatively inaccurate localization. We used a chromaticity based algorithm to reduce the effect of illumination. Illuminant invariance is based on the observation that log chromaticity of the color space is largely unaffected by illuminant variations. Therefore, once the chromaticity of an object is known, illumination effects, such as shadowing, can be eliminated successfully in[65, 78]. Log chromaticity can be calculated from RGB color space,

$$c_k = R_k / \sqrt[3]{\prod\nolimits_{i=1}^{3} R_i}, \tag{3.10}$$

where c_k is chromaticity, R_k is RGB color value, and k is one of the color channels.

3.3.5 Visual Map Matching using Structure Tensor and Mutual Information

Structure tensors are the matrix representation of partial derivative information. They are typically used to represent gradient or edge information. They also provide a powerful description of local patterns, in contract to directional derivative, through coherence,

$$\mathrm{S_{pq}(x)} = \int_{-\infty}^{\infty} w(x-x')(\frac{\partial g(x)}{\partial x_p{}'}\frac{\partial g(x')}{\partial x'_q})d^2x' \tag{3.11}$$

, where w is a window function; and S is the structure tensor, a symmetric 2x2 matrix,

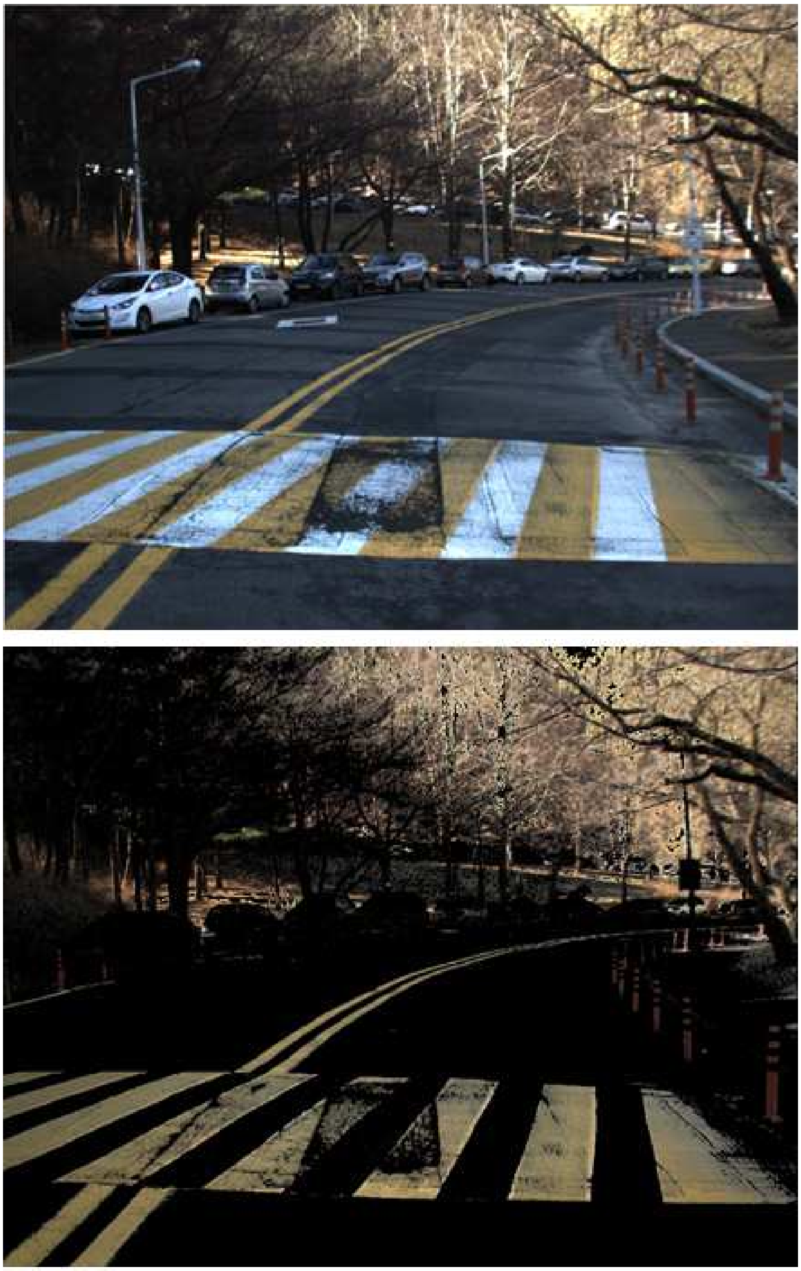


Figure 3.5: Example of removing dynamic objects. Vehicles are removed while preserving road markers, such as lane markers and speed bumps.

$$S = \begin{bmatrix} S_{11} & S_{21} \\ S_{12} & S_{22} \end{bmatrix}. \tag{3.12}$$

Several useful quantities can be calculated from the structure tensor,

$$I = \frac{S_{11} + S_{22}}{(S_{11} + S_{22}) + \sigma_{img}}, \tag{3.13}$$

$$\phi = \frac{1}{2} \arctan(\frac{2S_{12}}{S_{11} + S_{22}}), \tag{3.14}$$

and

$$C_c(S) = \frac{\sqrt{(S_{22} - S_{11})^2 + 4S_{12}^2}}{S_{11} + S_{22}}, \tag{3.15}$$

where I is the intensity, a measure of the magnitude of the gradients; $(S_{11} + S_{22})$ is the mean square magnitude of the gradient; σ_{img} is the standard deviation of the gray value in the input image; ϕ denotes the angle of orientation of the structure tensor; and C_c is the coherency, which reduces the local structure to a local orientation. Figure 4.8 shows pseudo color images, where ϕ codes the color, C_c the saturation, and the intensity of each pixel of the image in HSV representation. Using gradient information of local neighbourhoods has the disadvantage that one cannot distinguish between homogeneous areas (i.e. constant gray values) and uncorrelated noise (i.e. isotropic orientation of the gradients in the respected neighbourhood). The coherency value varies between zero and one. It reduces the local structure to a local orientation. In case of an ideal orientation the value is one, in case of an isotropic gray value structure it is zero.

Suppose we have two images, a reference image, X, converted from the 3D digital map, and an input image, Y. Both images are represented by grayscale intensity values. Mutual information of the two images X and Y is

$$\mathrm{M(X, Y) = H(X) + H(Y) - H(X, Y)}, \tag{3.16}$$

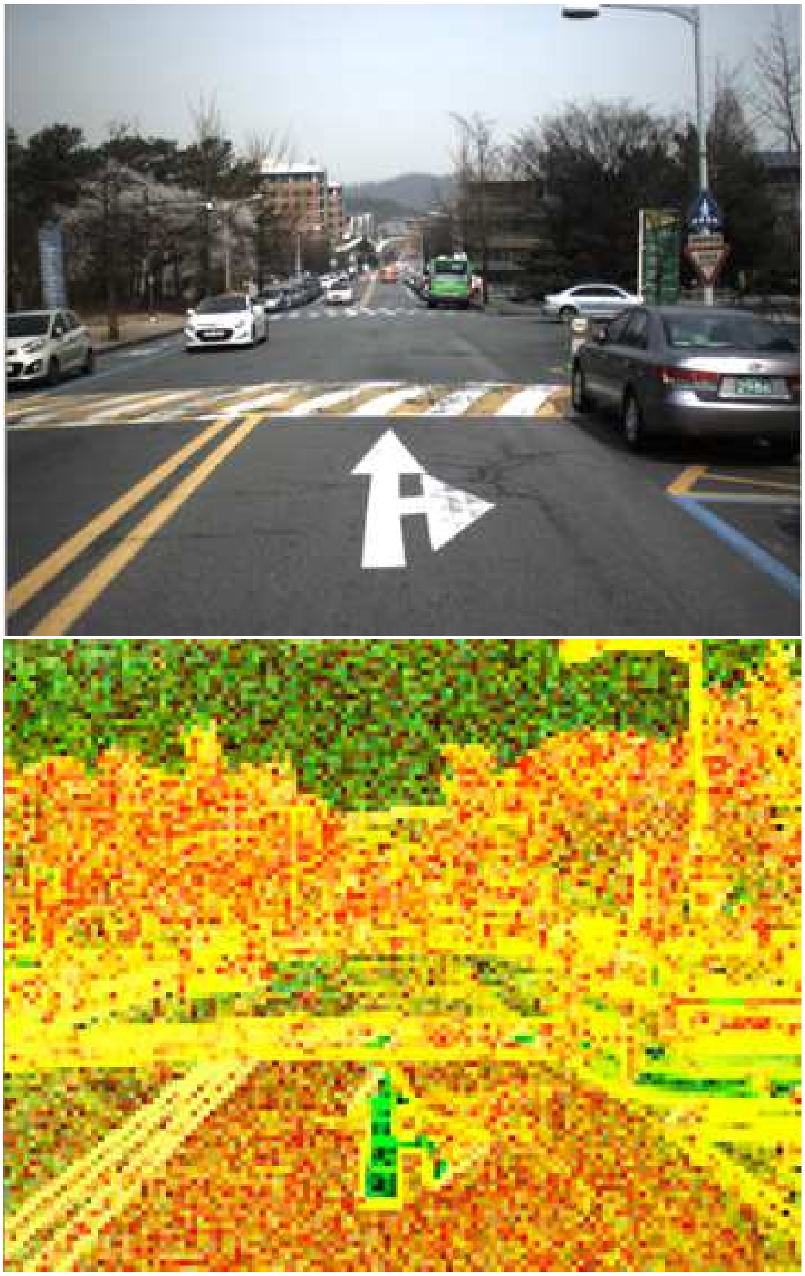


Figure 3.6: Original and structure tensor images in pseudo color.

where $H(X)$ is the entropy of the reference image, $H(Y)$ is the entropy of the input image and $H(X, Y)$ is the joint entropy of the two images. The marginal and joint entropies are defined as

$$\mathrm{H(X)} = \sum_{x} -P_x(x) \log P_x(x), \tag{3.17}$$

$$\mathrm{H(Y)} = \sum_{y} -P_y(y) \log P_y(y), \tag{3.18}$$

$$\mathrm{H(X, Y)} = \sum_{x,y} -P_{x,y}(x, y) \log P_{X,Y}(x, y), \tag{3.19}$$

$$P_X(x) = \sum_{y} P_{X,Y}(x, y), \tag{3.20}$$

$$P_Y(y) = \sum_{x} P_{X,Y}(x, y). \tag{3.21}$$

Mutual information measures the distance between joint distribution $P_{(X;Y)}(x, y)$ and the completely independent distributions, $P_X(x)$, $P_Y(y)$. The two images are considered registered when $M(X, Y)$ is maximal. However, mutual information is sensitive to the amount of overlap between the images and, to overcome this, the more robust normalized mutual information (NMI) measure was proposed [9],

$$M(X, Y) = \frac{H(X) + H(Y)}{H(X, Y)}. \tag{3.22}$$

The joint and marginal distributions can be estimated using joint and marginal histograms, $H(X, Y)$, $H(X)$, and $H(Y)$, where the joint histogram, $H(X, Y)$, for images X and Y over their region of overlap can be estimated by counting the number of times the intensity pair (x, y) occurs in corresponding pixel pairs. Mutual information yields values from 0 to ∞. The higher the mutual information is, the more information is shared between X and Y. However, high values of mutual information might be unintuitive and hard to interpret due to its unbounded range of values. Normalized Mutual

Information measures try to bring the possible values to bounded range. Specifically, case of a maximum value is useful due to ease of comparison with commonly used correlation coefficients. Normalizing the joint histogram provides an estimate of the joint probability distribution,

$$P_{X,Y}(x, y) = \frac{H(x, y)}{\sum_{x,y} H(x, y)}. \tag{3.23}$$

Thus, estimation of MI and NMI similarity measures only requires calculation of the joint histogram.

Estimation of $h(x, y)$ for images X and Y is usually achieved by calculating the number of times the intensity pair (x, y) occurs, i.e., all pixels contribute similarly. In contrast, we give weight $w(s)$ to each pixel based on its structure tensor value. Coherency varies between 0 and 1. If $C_c(s)$ = 1, the pixel is included in the line landmarks, otherwise, the pixel is included in the road surface. Thus, each pixel contributes to the calculation of the joint histogram according to its weight. This is similar to calculating a similarity measure using irregular sampling. The weight, $w(s)$, given to each pixel, s, in the reference image was chosen to be an exponential function of $I(s)$, $\phi(s)$, and $C_c(s)$ (from eqs. (10) to (12), respectively),

$$\mathrm{w(s) = exp(-q\phi(s)/C_c(s)I(s))}, \tag{3.24}$$

where q is a positive constant. The constant k is used to control the slope of the exponential function and can be set to any convenient fixed value proportional to the image resolution. For q = 0, the equation reduces to the conventional mutual information method.

The weight function allows us to calculate the joint histogram from pixels including different landmarks. Rather than calculating the matching cost of whole pixel, it is effective to calculate meaningful pixel values including important features. Therefore, $H(X), H(Y)$ and $H(X, Y)$ become

$$\mathrm{H(X)} = \sum_{x} -w(s)P_x(x)\log[w(s)P_x(x)], \tag{3.25}$$

$$\mathrm{H(Y)} = \sum_{y} -w(s)P_y(y)\log[w(s)P_y(y)], \tag{3.26}$$

and

$$\mathrm{H(X,Y)} = \sum_{x,y} -w(s)P_{x,y}(x,y)\log[w(s)P_{X,Y}(x,y)]. \tag{3.27}$$

In summary, we can find the values of x, y, and yaw by optimizing

$$(\widehat{x}_k, \widehat{y}_k, \widehat{\theta}_k) = \underset{(x_k, y_k, \theta_k)}{\arg\max} \frac{H(X) + H(Y)}{H(X,Y)}. \tag{3.28}$$

3.4 Experiments and Result

3.4.1 Methodology

We present results from real data collected using a 3D laser scanner (Velodyne HDL-64E) on the roof and monocular camera mounted at the location of the rear-view mirror within the vehicle. To evaluation the proposed algorithm, we used an automatically created high-precision 3D digital map. The test vehicle was equipped with RTK-GPS (OXTS RT3002) and high-precision Inertial Navigation System (INS), as shown in Fig. 3.7, which provides ground truth for the position measurements. The proposed localization system was evaluated on campus roads, comprising approximately 3.7 km, as shown in Fig. 3.8. All experiments were performed using a PC with 3.40 GHz i7-4770 CPU. We determined the absolute position errors, including lateral and longitudinal error, by differencing ground truth and measured positions using the proposed algorithm.

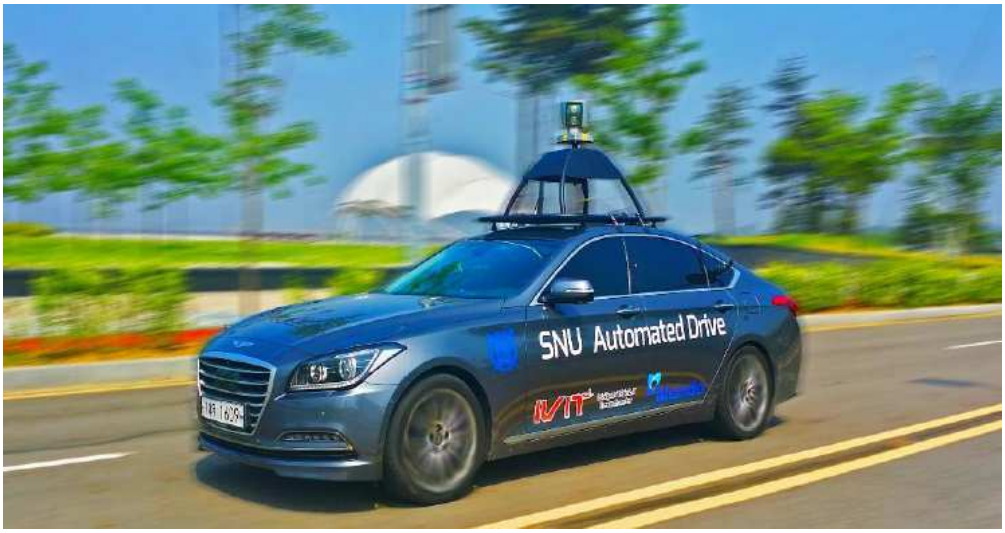


Figure 3.7: Autonomous vehicle equipped with a 3D LIDAR for generating map. monocular camera is mounted on the location of rear-view mirror in the vehicle.

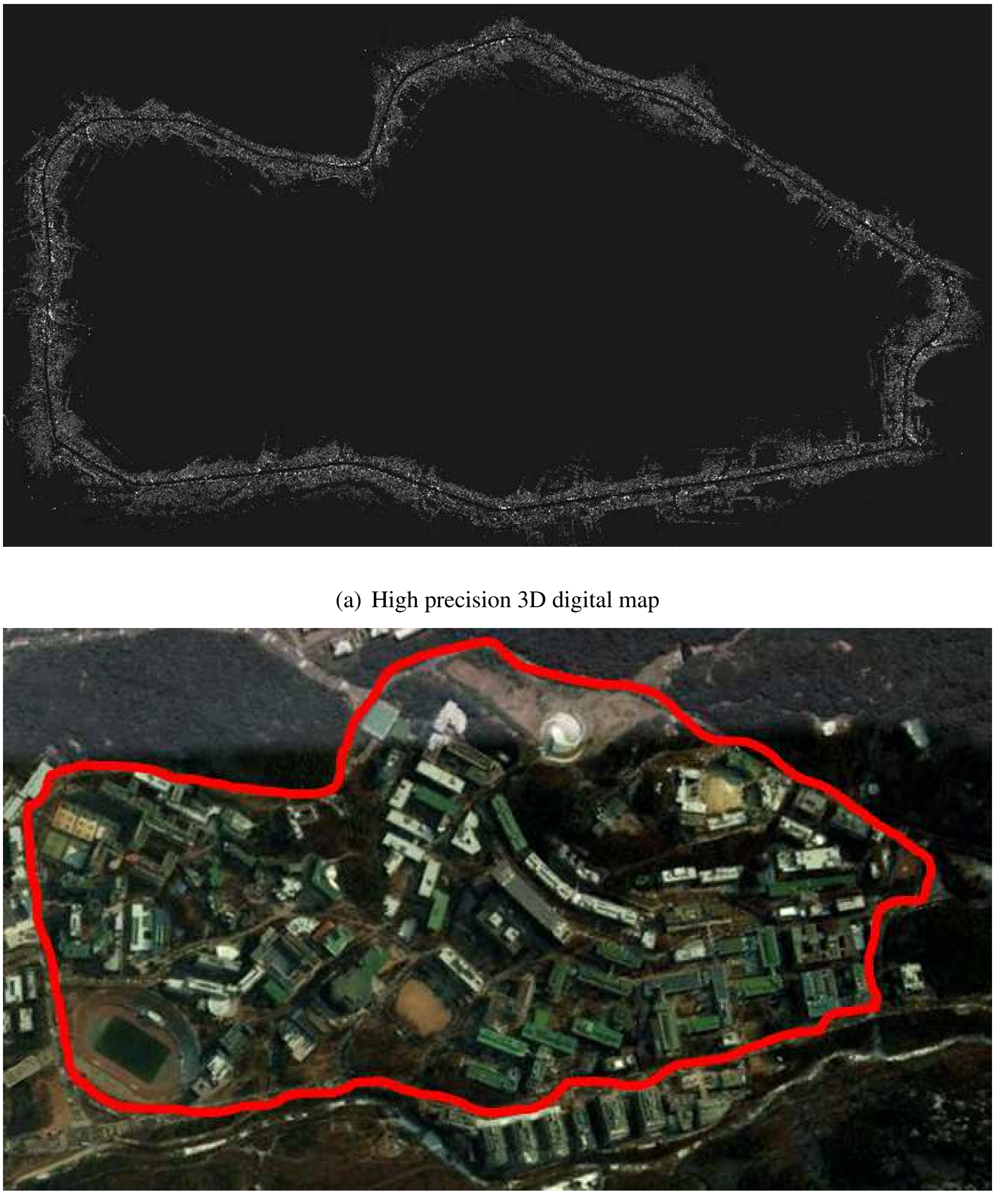
We also measured the performance of removing dynamic objects using

$$precision = \frac{TP}{(TP + FP)}, \tag{3.29}$$

and

$$recall = \frac{TP}{(TP + FN)}, \tag{3.30}$$

where TP, FP, and FN denote true positive, false positive, and false negative, respectively. These were computed object-wise in each frame and frames that did not include the actual object or detected object were skipped for accurate performance evaluation. Successful detection means more than 50 percent of objects were detected for accurate performance evaluations.



(a) High precision 3D digital map

(b) Aerial image

Figure 3.8: (a) Digital map generation result and (b) aerial image of the area with the vehicle trajectory (red).

Methods	Detection rate
Precision	0.824
Recall	0.811

Table 3.1: Precision and recall for removing dynamic objects.

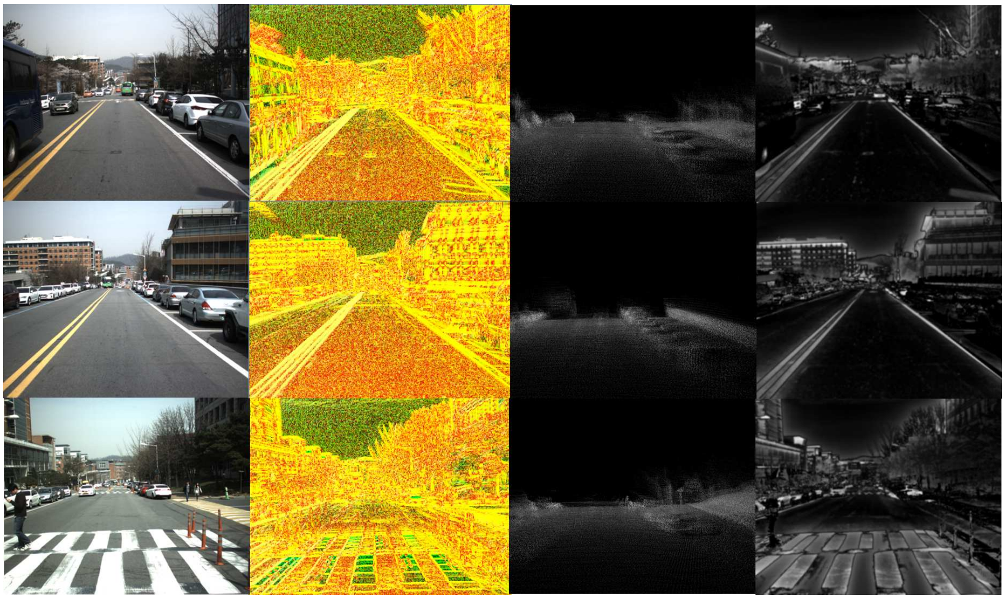


Figure 3.9: Experimental results. First row: original images, Second row: structural tensor images in pseudo color, Third row: synthetic view generated from digital map, Fourth row: the result of removing dynamic object.

Position	RMS Error
Absolute position	20.19cm
Lateral position	7.53cm
Longitudinal position	18.73cm
Yaw angle	0.354 deg.

Table 3.2: The localization errors.

3.4.2 Quantitative Results

Figure 3.8 shows the digital map describing the campus, with the vehicle trajectory in red. We verify the effectiveness of removing dynamic objects with experiments on the campus dataset. To create ground truth images, we used graphical software to label dynamic objects manually, focusing on preserving features while removing dynamic objects such as vehicles and pedestrians. Table 3.1 shows the precision and recall for removing dynamic objects using the proposed method compared to the ground truth images. Recall is also referred to as the true positive rate or sensitivity, and precision is also referred to as positive predictive value.

The vehicle traveled a loop of the campus. We assume the initialization position was given using GPS and calculated the position errors from ground truth, achieving root mean square (RMS) errors of 20.19 cm. The campus was located in mountainous terrain and had many speed bumps in the route chosen. Therefore, the experimental result had maximum RMS error of 40 cm. In future work, we propose to develop a stabilization algorithm for motion compensation.

Finally, Fig. 3.9 shows several examples of the proposed method, summarizing results for removing dynamic objects, digital map conversion, and structural tensor images in psuedo color.

3.5 Conclusions and Future Works

We proposed a visual localization system based on a high-precision 3D digital map. Localization is the core application for automated driving. The proposed system comprises extrinsic calibration, dynamic object removal, illumination invariant transformation, and visual localization. We propose a visual localization algorithm using structure tensors and mutual information. Experiments were conducted to demonstrate the system estimates for vehicle location on the high precision map. Although the achieved accuracies are not yet sufficient for autonomous driving, the proposed system enables robust localization in areas without GPS, and a good basis for constitutive visual mapping methods.

Further improvements could be realized by reducing processing time using GPU acceleration, and the proposed approach should be extended to urban areas for improved reliability and accuracy.

Chapter 4

Free Space Computation using a Sensor Fusion of LIDAR and RGB Camera in Vehicular Environments

4.1 Introduction

Drivable free space is a fundamental research topic in automated driving vehicles. In addition, visual perception and sensor fusion have made clear that they will play a key role. To achieve accurate and reliable performance, various algorithms based on different kinds of sensors have been developed. For automated driving vehicles, reliable and accurate free space detection is a prerequisite. As there are many different kinds of roads, such as highways, urban roads and country roads, with different features, the approaches to detect them are different. Besides obstacle avoidance, the drivable free space can also facilitate the path planning and decision making, especially in such a situation where lane markers are invisible or not present.

The problem has been investigated for many years and a large variety of approaches can be found in the literature based on a monocular camera. Among the algorithms that perform best on the KITTI road benchmark data set [85], the majority work only on single camera image and several make use of deep neural networks [71]. Despite achieving outstanding results, image-based approaches are greatly affected by lighting environment. As a result, their outcomes are expected to decrease significantly in the time of night or whenever presented with light conditions that deviate from those seen during training.

Stereo vision based approaches first get dense disparity map through stereo matching. Then the image including depth information can be used to detect the road. However, it is a contradiction between the computational cost and recover accuracy. In addition, stereo camera can also be badly affected by the light like a monocular camera.

High definition 3D LIDARs use the accurate 3D information to analyze the structure of the field and take the ground area without obstacles as road. This type of method uses only the sparse 3D information while the color and texture information are not enough to distinguish the non-road areas that have little differences in height.

Since each modal of sensor has its weakness, multi-modal sensor fusion can be a simple solution to fill the gap. This chpater will provide a method to detect the reliable free space fusing the information of LIDAR and monocular image. Specifically, given a pair of RGB image and sparse depth map projected from LIDAR point cloud, we generate dense depth map. Furthermore, we compute the drivable free space using visual features from dense depth map. We tested our algorithm on the KITTI-Road dataset[85] and the results show that the proposed method reaches the state-of-the-art.

The chapter is organized as follows: In Section 4.2, we describe the proposed algorithm . The experimental results are presented in Section 4.3, and Section 4.4 contains conclusions concerning this study.

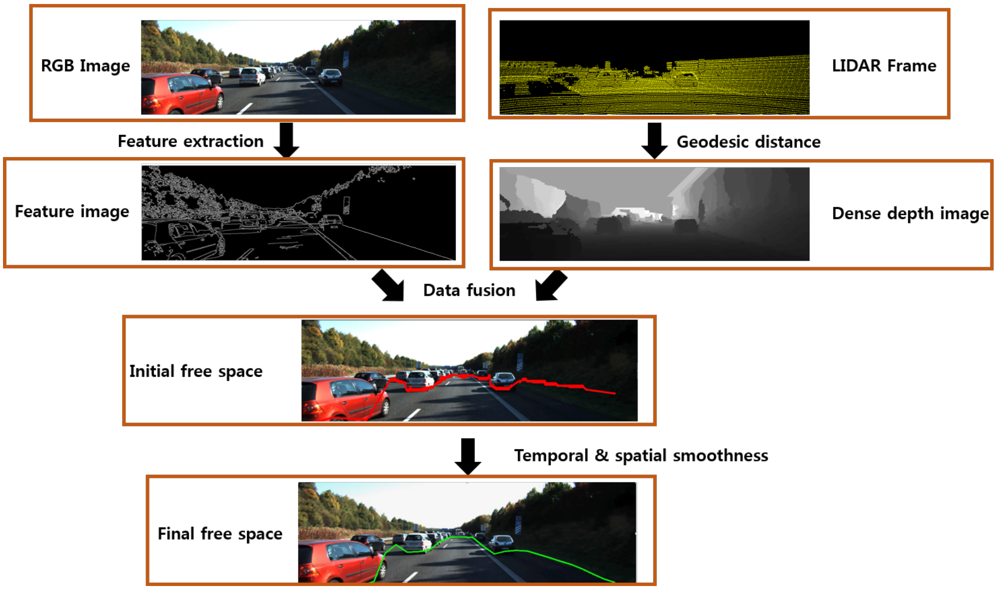


Figure 4.1: Processing flow of the proposed algorithm.

4.2 Methodology

This section describes overall procedure of the proposed algorithm. We first generate the dense depth map using LIDAR data and introduce how to extract feature from image. Figure 4.1 shows a whole framework of this chapter.

4.2.1 Dense Depth Map Generation

A point cloud is a set of data points in some coordinate systems. The point cloud represents the set of points that that the device has measured. Unfortunately, such point clouds tend to be sparse. By using the LIDAR and the camera, we achieve a best approximation of a pixel's depth value based on the values at surrounding pixels. An iterative computer vision technique in which the intensity value, I_c, of each pixel p is updated according to the intensity value of its neighbors $q0$, $q1$, $q2$, and $q3$:

$$I_c^n(p) = I_c^{n-1}(p)(1 - \lambda \sum_{j=0}^{3} c(p, q_j)^{n-1}) + \lambda \sum_{j=0}^{3} c(p, q_j)^{n-1} I_c{}^{n-1}(q_j), \tag{4.1}$$

where 0<λ<0.25 controls the influence of neighboring pixels and n is the iteration number. Different functions can be used to implement the diffusion coefficient $c(p, q)$. One possibility is the exponential function of the negative euclidean distance Δc_{pq} in the sparse depth map from LIDAR data:

$$c(p, q) = \exp(-\frac{\Delta c_{pq}}{r_c}). \tag{4.2}$$

Figure 4.2 shows a example of a dense depth map. The image is a sample image in KITTI dataset; RGB image, sparse depth map and dense depth map result. It produces some blurry boundaries in object but we smooth the area using the image data. Some example results using KITTI dataset are depicted in figure 4.3.

4.2.2 Color Distribution Entropy

We develop an color distribution entropy using a monocular camera to model the equation for each area to be free space or not. We use a area prior to enforce pixels that are always free space to be so. Our purpose is to detect the free space that the line exists on the boundary. Towards this, we derive an equation that finds the entropy of the color distribution in blocks around the labels:

$$E_a(y_i = k) = (w_1 T[L(z) = 0] + w_2 T[L(z) > 0]) \times H(i, k) \sum_{j=k}^{h} H(i, j). \tag{4.3}$$

Here the entropy $H(i, j)$ is computed in terms of the color distribution of free space pixels in a block centered at pixel location (i, j). w_1 and w_2 are penalty considering the height of the LIDAR data. To extract the color distribution of free space pixel, we

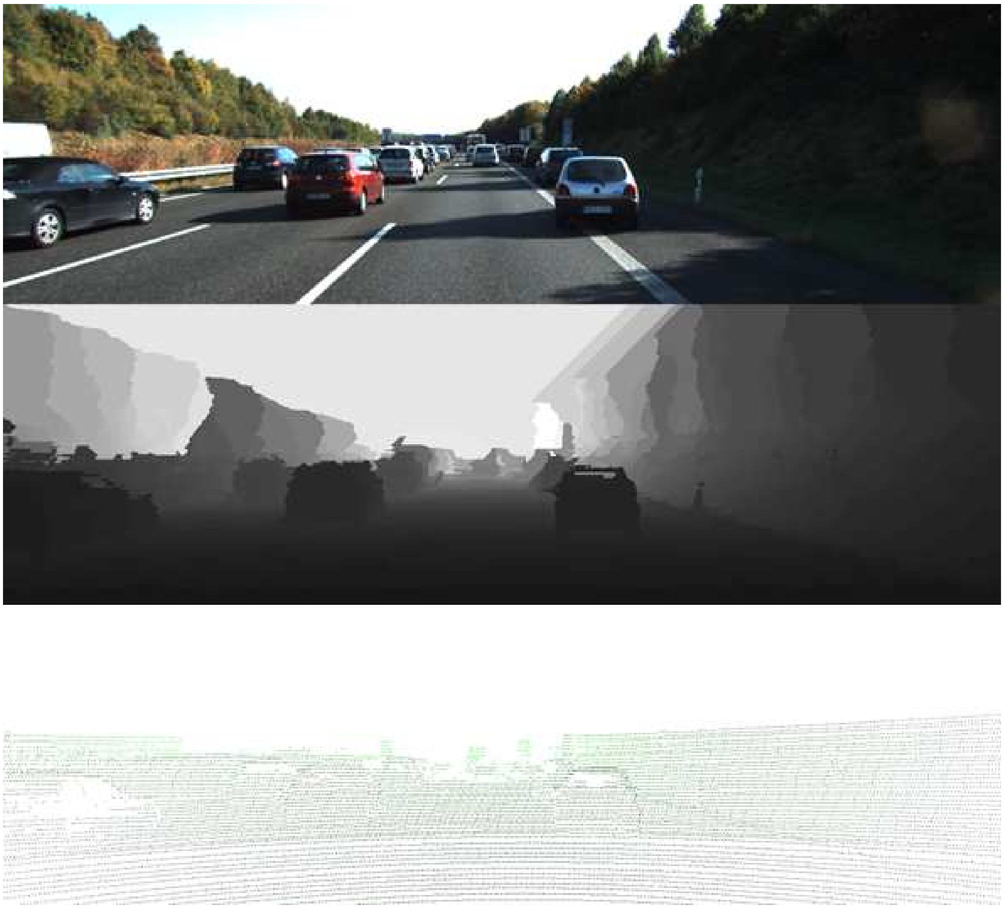


Figure 4.2: Original image, dense depth images, LIDAR frame.

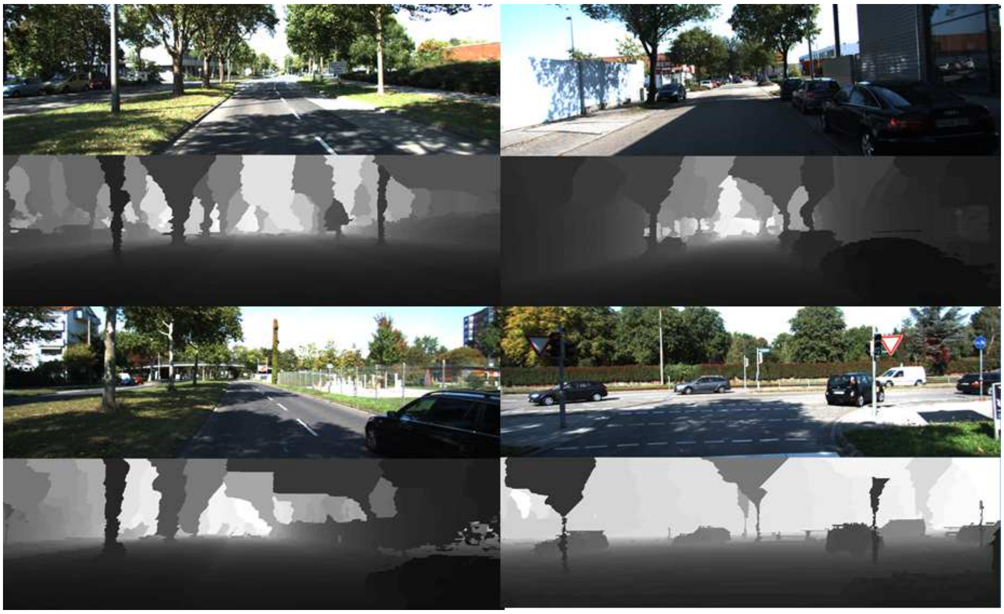


Figure 4.3: Examples of original image and depth images.

consider the data near the ground. If the pixel is belong to the object, the weight factor is small. The entropy $H(i,j)$ should be higher near the boundary pixels. Since we are finding a line that passes through the boundary between the closest set of obstacles and the road, we use a cumulative sum that prefers pixels that are closer to the obstacle and the ones with non-zero $H(i,k)$ values.

4.2.3 Edge Extraction

Edge detection is a process of locating an edge of an image. Detection of edges in an image is a important step towards understanding image features. Edges consist of meaningful features and contain significant information. It significantly reduces the image size and filters out information that may be regarded as less relevant, thus preserving the improtant structural properties of an image. Since edges often occur at image locations representing object boundaries, edge detection is extensively used in boundary detection when images are divided into areas corresponding to different objects. Considering this, we define an equation which accumulates edge evidence as

$$E_{edge}(y_i = k) = e(i,k) \sum_{j=k}^{h} e(i,j), \tag{4.4}$$

where $e(i,j) = 1$ if there is an edge at the pixel (i,j), and 0 otherwise. The edges are obtained by simply using the Canny edge detector with default parameters. Note that more complicated edge detectors could be realized.

4.2.4 Temporal Smoothness

In a vehicular environment, a camera is ported in front of a vehicle and the road is observed by this camera set-up. Temporal smoothness between consecutive frames by cumulating temporal data increases robustness and accuracy of the estimation. As the first step, the percentage of histogram change is calculated between previous and current frames as follows.

$$\Delta^t = \frac{1}{t} \sum_{i=1:w} |Hist^t(i) - Hist^{t-1}(i)|, \tag{4.5}$$

where N is the total number of frames, $Hist^t$ is the histogram of the frame at time instant t. The rate of change is utilized as a weighting function to model in temporal transfer of data relating weights. Hence, weights in the current frame are weighted by the values in the previous frame as follows:

$$\mu_t(x) = (1 - \Delta^t)\mu_t(x) + \Delta^t \mu_{t-1}(x), \tag{4.6}$$

where μ_t is the vector involving weights in four fundamental directions for time instant t. The update formula in (4.11) enforces utilization of weights in the current frame, as long as significant scene change is not observed. On the other hand, when there is significant scene change, which is not an expected case, weights of the previous frame are utilized. Once the weights are calculated, histogram of the current frame is updated by the change factor for the analysis of the next frame as follows:

$$Hist^t(i) = (1 - \Delta^t)Hist^t(i) + \Delta^t Hist^{t-1}(i), \tag{4.7}$$

The histogram update in (4.12) provides robustness against multiple inconsistent consecutive frames. Therefore, data from the last reliable frame is transferred to the frames involving severe flares, reflections and sudden large occlusions as soon as a consistent frame is encountered with similar histogram characteristics.

The other temporal modification is provided by enforcing smoothness of intensity values along pixels with low intensity change between consecutive frames. For this purpose, temporal weights are calculated for each pixel as

$$\mu^t(x) = \exp(-|I^t(x) - I^{t-1}(x)|/\sigma), \tag{4.8}$$

where I^t is the intensity image for time instant t, σ is a scaling factor (set as 16). Temporal weight relates the change of the corresponding pixel in time, which is utilized to enforce intensity values of the previous frame to the estimation of current intensity value.

4.2.5 Spatial Smoothness

Spatial smoothing means that data points are averaged with their neighbours. This has the effect of a low pass filter meaning that high frequencies of the signal are removed from the data while enhancing low frequencies. We employ the spatial smoothing shown as below.

$$E_{spatial} = \exp(-\alpha(y_i - y_j)^2), \tag{4.9}$$

where α is constants.

4.3 Experiment and Evaluation

4.3.1 Evaluated Methods

We evaluate our algorithm on KITTI visual benchmark suite[19]. The KITTI dataset is an opensource dataset created by Dr. Andreas Geiger *et al.* for outdoor autonomous driving application. The object sub-dataset in KITTI is captured by Velodyne HDL-64E laser scanner and two highly-calibrated binocular cameras. Each image captured by camera is registered to a set of 3D laser point cloud, which means that, for each image, we can get one corresponding sparse depth map by projecting the 3D laser point cloud to the image plane. We use the training set from the road challenge in KITTI. The road and lane estimation benchmark consists of 289 training and 290 test images. It contains three different categories of road scenes: UU(Urban Unmarked), UM (Urban Marked), and UMM (Urban Multiple Marked Lanes). In this chapter, we only deal with free space detection, the lane information is not considered here. The dataset offer groundtruth for training data and online evaluation of testing data is offered on the website. For evaluation, the dataset offered pixel-based evaluation and behaviour-based evaluation. A set of metrics including precision (PRE), recall (REC), maximum F1-measure (MaxF) ,average precision (AP), false positive rate (FPR) and false negative rate (FNR) are used for evaluation.

$$Precision = \frac{TP}{TP + FP}, \tag{4.10}$$

$$Recall = \frac{TP}{TP + FN}, \tag{4.11}$$

$$F - measure = (1 + \beta^2)\frac{PrecisionRecall}{\beta^2 Precision + Recall}, \tag{4.12}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}. \tag{4.13}$$

Furthermore, in order to provide insights into the performance over the full recall range, the average precision (AP) as defined in [76] is computed for different recall values r:

$$AP = \frac{1}{11} \sum_{r \in 0,0.1,\ldots 1} \max_{r':r'>r} precision(r'). \tag{4.14}$$

The experiments were tested on a standard PC with 8GB of RAM and a dual-core Intel Core i7-4770 CPU clocked at 3.4 GHz. The algorithm was implemented with C++ under Windows8. As we take each pixel as a random variable, for images from KITTI-Road dataset(with resolution of about 1240x375), the average time consuming is about 0.4 seconds.

4.3.2 Experiment on KITTI Dataset

We compared our algorithm with the recently developed ones, including, RES3D-Velo[103], FusedCRF[104], and HybridCRF[105]. The evaluation on KITTI datasets and the average results are shown in table 4.1 and 4.2. From the results showed in the tables, we can see that our algorithm get fine results using fusion method on KITTI dataset. But the results is less competitive than those on deep learning-based method, that may be due to the images contain little high rising non-road objects and the 3D information of LIDAR are less helpful. And in average, the proposed algorithm gets the best MaxF. In Figure 4.5 to 4.7, the result of the proposed algorithm are shown. We generate the dense depth map from the LIDAR frame and detect the free space using the fusion of the camera data and LIDAR frame. We compare the result from the ground truth. The Table 4.1 show the MaxF, AP, PRE, REC, FPR and FNR results. Table 4.2 shows the result based on the distance.

Method	MaxF	AP	PRE	REC	FPR	FNR
FUSEDCRF	89.55%	80%	84.87%	94.78%	7.7%	5.22%
HYBRIDCRF	90.99%	85.26%	90.65%	91.33%	4.29%	8.67%
RES3D-VELO	83.81%	93.95%	78.56%	89.8%	11.16%	10.2%
AUTHOR'S	91.36%	84.92%	90.18%	93.28%	5.45%	6.72%

Table 4.1: Comparison of evaluation on KITTI Dataset.

Distance(m)	MaxF	PRE	REC	FPR	FNR
0-20	92.36	90.67	94.11	5.19	6.92
20-25	92.37	90.78	94.02	4.8	6.93
25-30	92.34	90.75	93.99	4.5	6.95
30-35	92.13	90.54	93.78	4.37	7.15
35-40	91.9	90.36	93.5	4.33	7.42
AVERAGE	92.26	90.65	93.88	4.45	6.97

Table 4.2: Distance accuracy comparison on KITTI Dataset.

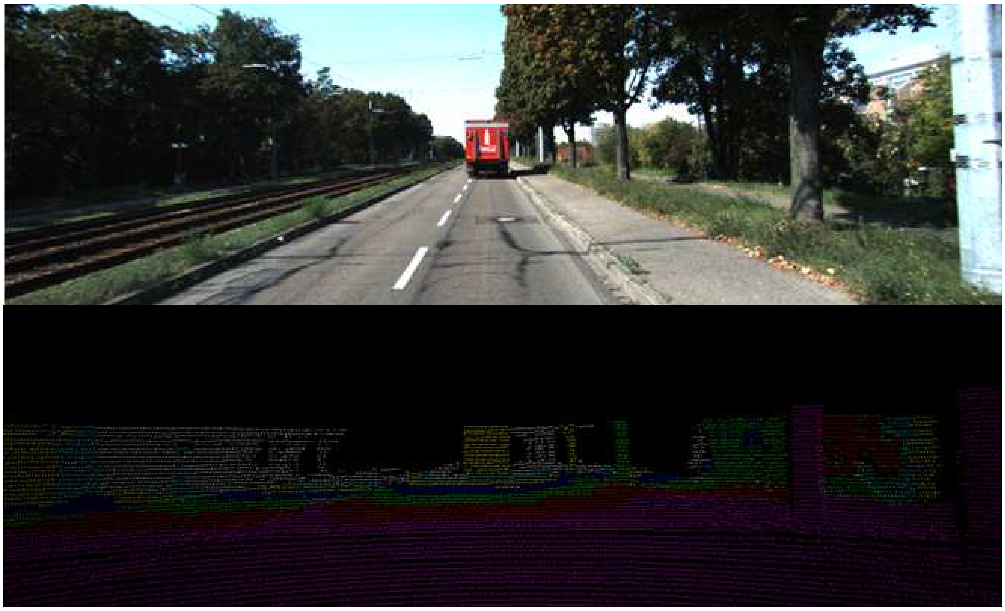


Figure 4.4: Original image and LIDAR frame.

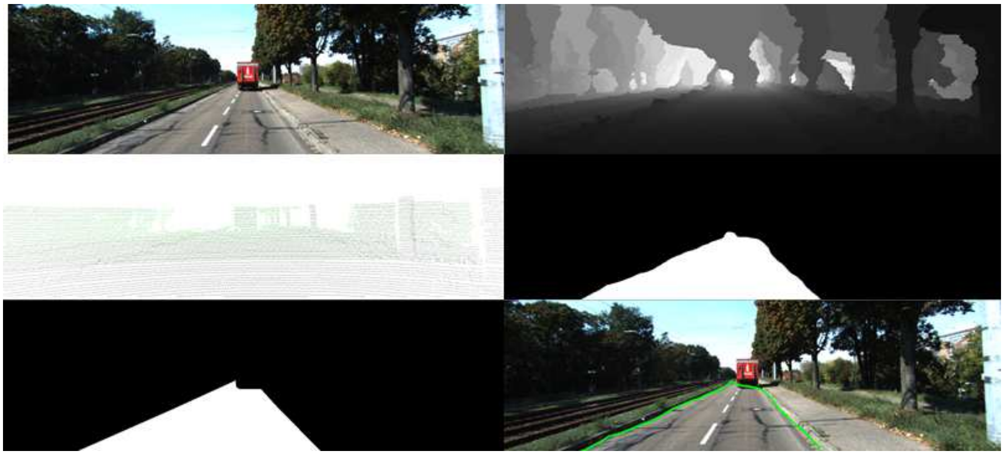


Figure 4.5: Example of experimental result.

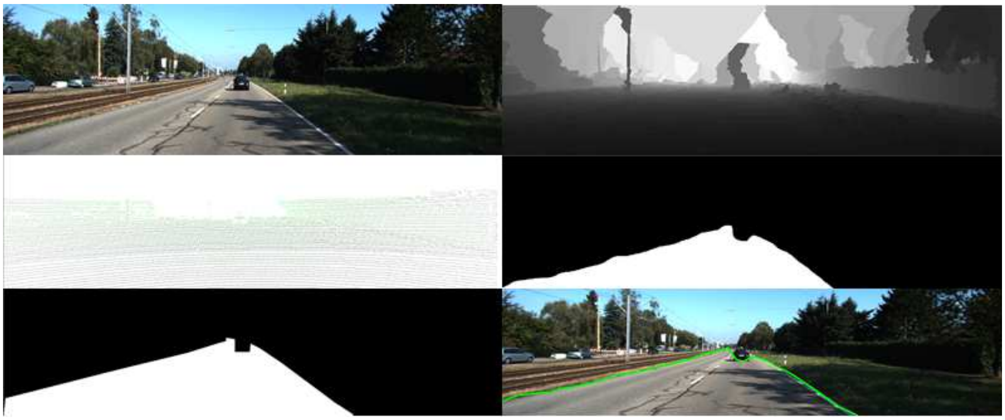


Figure 4.6: Example of experimental result.

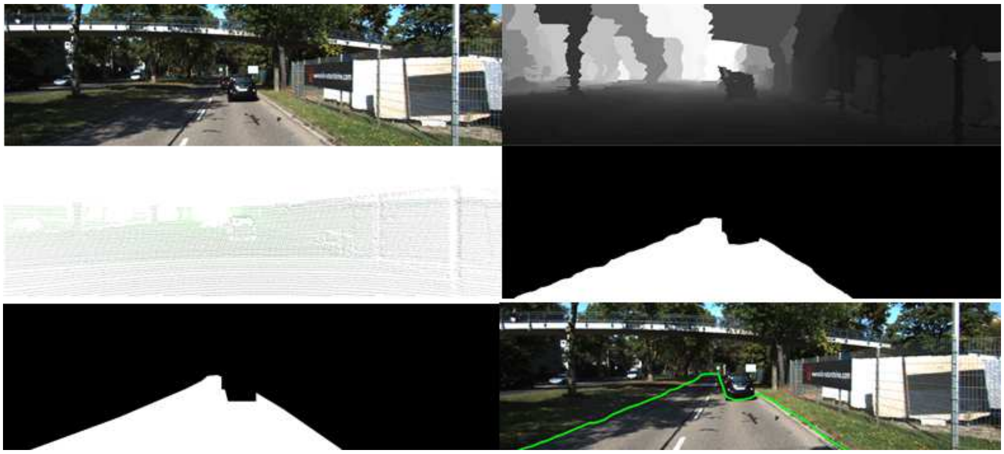


Figure 4.7: Example of experimental result.

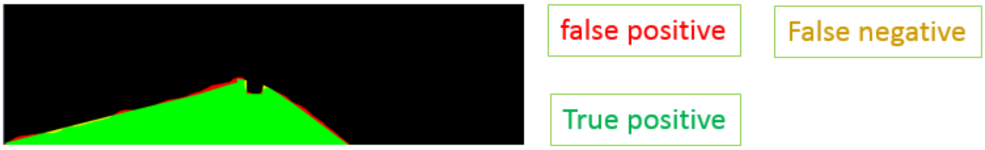


Figure 4.8: Ground truth, false positive, false negative and true positive.

4.4 Conclusion

In this chapter, we have proposed free space detection method by combing RGB image and LIDAR. We generate the dense depth map from the LIDAR data and extract the feature using the camera data. Also, we propose a method using a temporal and spatial smoothing technique. Particularly, the approach was tested on KITTI datasets involving road scenes. We designed our algorithm using simple and light-weight techniques for real-time computation. In future work, we will involve the initial depth value into geodesic distance computation and implement the algorithm on GPU for real-time application.

Chapter 5

Conclusion

In this dissertation, we focused on several issues for the computer vision applications of the automated driving vehicles. In chapter 2, we proposed a method for measuring the distances based on frequency-domain analysis using a stereo camera. By analyzing the frequency-domain information, the depth value of an object can be obtained accurately in less time than the existing comparable methods. Experimental results show that the proposed method significantly improves the accuracy of distance measurement. In chapter 3, we proposed a visual localization system based on a high-precision 3D digital map. The proposed system comprises extrinsic calibration, dynamic object removal, illumination invariant transformation, and visual localization. We propose a visual localization algorithm using structure tensors and mutual information. Experiments were conducted to demonstrate the system estimates for vehicle location on the high precision map. Although the achieved accuracies are not yet sufficient for autonomous driving, the proposed system enables robust localization in areas without GPS, and a good basis for constitutive visual mapping methods. In chpater 4, we proposed a method for drivable free space detection using fusion of a LIDAR and a camera. We generate the dense depth map from LIDAR and extract features from a camera.

A number of open problems should be solved to develop the automated driving vehicles that can be driven on roads. Although some parts of the studies remain unresolved and need further attention in the future, we believe that the theoretical analysis and outstanding results in this dissertation will definitely provide useful guidelines to improve the automated driving vehicles. We hope many follow-up studies will be carried out and many valuable research results will be produced.

Bibliography

[1] M. Buehler, K. Iagnemma, and S. Singh. The DARPA Urban Challenge: Autonomous Vehicles in City Traffic. Springer, 2009.

[2] Scharstein, D., Hirschmüller, H., Kitajima, Y.: 'High-resolution stereo datasets with subpixel-accurate ground truth'. German Conf., Pattern Recognition, September 2014, pp. 31–42

[3] Hirschmuller, H.: 'Stereo processing by semiglobal matching and mutual information', IEEE Trans. Pattern Anal. Mach. Intell., 2008, 30, (2), pp. 328–341

[4] L. Ladicky, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. Torr. Joint optimisation for object class segmentation and dense stereo reconstruction. In BMVC, 2010.

[5] R. Benenson, R. Timofte, and L. Gool. Stixels estimation without depth map computation. In ICCV, 2011.

[6] M. Brubaker, A. Geiger, and R. Urtasun. Lost! leveraging the crowd for probabilistic visual self-localization. In CVPR, 2013.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR, 2005.

[8] W. P. Een, N. Jindapetch, L. Kuburat, and N. Suvanvorn, “A Study of the Edge Detection for Road Lane,” Electrical Engineering/ Electronics, Computer, Telecommunications and Information Technology (ECTI), pp. 995-998, 2012.

[9] Wolcott, Ryan W., and Ryan M. Eustice. ”Visual localization within lidar maps for automated urban driving.” 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2014. 1, 2, 5

[10] Wei, J., Snider, J.M., Kim, J., *et al.*: ‘Towards a viable autonomous driving research platform’. IEEE Intelligent Vehicles Symp. (IV), June 2013, pp. 763–770

[11] Velodyne Lidar: ‘Velodyne HDL-64E product description’. Available at http://velodynelidar.com/hdl-64e.html, accessed December 2016

[12] Davison, A.J.: ‘Real-time simultaneous localisation and mapping with a single camera’. Proc. Int. Conf. Computer Vision, October 2003, pp. 1403–1410

[13] Pollefeys, M., Nistér, D., Frahm, J.M., *et al.*: ‘Detailed real-time urban 3d reconstruction from video’, Int. J. Comput. Vis., 2008, 78, (2–3), pp. 143–167

[14] Stellet, J.E., Schumacher, J., Lange, O., *et al.*: ‘Statistical modelling of object detection in stereo vision-based driver assistance’. Intelligent Autonomous Systems 13, 2016, pp. 749–761

[15] Henry, P., Krainin, M., Herbst, E., *et al.*: ‘RGB-D mapping: using depth cameras for dense 3D modeling of indoor environments’. Experimental Robotics Springer, Berlin, Heideberg, pp. 477–491

[16] Newcombe, R.A., Izadi, S., Hilliges, O., *et al.*: ‘KinectFusion: real-time dense surface mapping and tracking’. Int. Conf. Mixed and Augmented Reality (ISMAR), October 2013, pp. 127–136

[17] Urmson, C., Anhalt, J., Bagnell, D., *et al.*: ‘Autonomous driving in urban environments: boss and the urban challenge’, J. Field Robot., 2008, 25, (8),pp. 425–466

[18] Wolcott, R.W., Eustice, R.M.: ‘Fast LIDAR localization using multiresolution Gaussian mixture maps’. IEEE Int. Conf. Robotics and Automation (ICRA), May 2015, pp. 2414–2821

[19] Geiger, A., Lenz, P., Urtasun, R.: ‘Are we ready for autonomous driving? The KITTI vision benchmark suite’. IEEE Computer Vision and Pattern Recognition, June 2012, pp. 3354–3361

[20] Guney, F., Geiger, A.: ‘Displets: resolving stereo ambiguities using object knowledge’. Proc. IEEE Conf. Computer Vision and Pattern Recognition, June 2015, pp. 4165–4175

[21] Zbontar, J., LeCun, Y.: ‘Stereo matching by training a convolutional neural network to compare image patches’, J. Mach. Learn. Res., 2016, 17, pp. 1–32

[22] Scharstein, D., Szeliski, R.: ‘A taxonomy and evaluation of dense two-frame stereo correspondence algorithms’, Int. J. Comput. Vis., 2002, 47, (1–3), pp.7–42

[23] Zabih, R., Woodfill, J.: ‘Non-parametric local transforms for computing visual correspondence’. European Conf. Computer Vision, May 1994, pp. 151–158

[24] Luan, X., Yu, F., Zhou, H., *et al.*: ‘Illumination-robust area-based stereo matching with improved census transform’. Int. Conf. Measurement, Information and Control (MIC), May 2012, pp. 194–197

[25] Baydoun, M., Al-Alaoui, M.A.: ‘Enhancing stereo matching with varying illumination through histogram information and normalized cross correlation’. Int. Conf. Systems, Signals and Image Processing (IWSSIP), July 2013, pp.5–9

[26] Ogale, A.S., Aloimonos, Y.: ‘Robust contrast invariant stereo correspondence’. Proc. IEEE Int. Conf. Robotics and Automation, April 2005, pp. 819–824

[27] Kolmogorov, V., Zabih, R.: ‘Computing visual correspondence with occlusions using graph cuts’. Proc. IEEE Int. Conf. Computer Vision, 2001, pp. 508–515

[28] Klaus, A., Sormann, M., Karner, K.: ‘Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure’. IEEE Int. Conf. Pattern Recognition (ICPR), August 2006, pp. 15–18

[29] Wang, Z.F., Zheng, Z.G.: ‘A region based stereo matching algorithm using cooperative optimization’. IEEE Int. Conf. Computer Vision and Pattern Recognition, June 2008, pp. 1–8

[30] Hermann, S., Klette, R.: ‘Iterative semi-global matching for robust driver assistance systems’. Asian Conf. Computer Vision, November 2012, pp. 465–478

[31] Spangenberg, R., Langner, T., Rojas, R.: ‘Weighted semi-global matching and center-symmetric census transform for robust driver assistance’. Int. Conf. Computer Analysis of Images and Patterns, August 2013, pp. 34–41

[32] Spangenberg, R., Langner, T., Adfeldt, S., *et al.*: 'Large scale semi-global matching on the cpu'. IEEE Intelligent Vehicles Symp. (IV), June 2014, pp.195–201

[33] Huang, S., Xiao, S., Feng, W.C.: 'On the energy efficiency of graphics processing units for scientific computing'. IEEE Int. Symp. on Parallel and Distributed Processing, May 2009, pp. 1–8

[34] Mei, X., Sun, X., Zhou, M., *et al.*: 'On building an accurate stereo matching system on graphics hardware'. IEEE Int. Conf. Computer Vision Workshops (ICCV Workshops), November 2011, pp. 467–474

[35] Banz, C., Hesselbarth, S., Flatt, H., *et al.*: 'Real-time stereo vision system using semi-global matching disparity estimation: architecture and FPGAimplementation'. IEEE Int. Conf. Embedded Computer Systems (SAMOS), July 2010, pp. 93–101

[36] Akin, A., Baz, I., Schmid, A., , *et al.*: 'Dynamically adaptive real-time disparity estimation hardware using iterative refinement', Integr. VLSI J., 2014, 47, (3), pp. 365–376

[37] Labayrade, R., Aubert, D., Tarel, J.P.: 'Real time obstacle detection in stereovision on non flat road geometry through 'v-disparity' representation'. IEEE Intelligent Vehicles Symp. (IV), June 2002, pp. 646–651

[38] Canny, J.: 'A computational approach to edge detection', IEEE Trans. Pattern Anal. Mach. Intell., 1986, 6, pp. 679–698

[39] Kang, S.N., Yoo, I.S., Shin, M., *et al.*: 'Accurate inter-vehicle distance measurement based on monocular camera and line laser', IEICE Electron. Express, 2014, 1, (9), pp. 1–7

[40] Ahlvers, U., Zoelzer, U., Rechmeier, S.: ‘FFT-based disparity estimation for stereo image coding’, Int. Conf. Image Processing, September 2003, pp. 761–764

[41] Van Loan, C.: ‘Computational frameworks for the fast Fourier transform’ (SIAM, 1992)

[42] Heo, Y.S., Lee, K.M., Lee, S.U.: ‘Joint depth map and color consistency estimation for stereo images with different illuminations and cameras’, IEEE Trans. Pattern Anal. Mach. Intell., 2013, 35, (5), pp. 1094–1106

[43] Yang, Q., Tan, K.H., Ahuja, N.: ‘Real-time O(1) bilateral filtering’. IEEE Int. Conf. Computer Vision and Pattern Recognition, June 2009, pp. 557–564

[44] Schneider, S., Himmelsbach, M., Luettel, T., *et al.*: ‘Fusing vision and lidarsynchronization, correction and occlusion reasoning’. IEEE Intelligent Vehicles Symp. (IV), June 2010, pp. 388–393

[45] Shen, L., Tan, P., Lin, S.: ‘Intrinsic image decomposition with non-local texture cues’. IEEE Int. Conf. Computer Vision and Pattern Recognition, June 2008, pp. 1–7

[46] Julier, S.J., Uhlmann, J.K.: ‘New extension of the Kalman filter to nonlinear systems’. AeroSense’97, July 1997, pp. 182–739

[47] Nummiaro, K., Koller-Meier, E., Van Gool, L.: ‘An adaptive color-based particle filter’, Image Vis. Comput., 2003, 21, (1), pp. 99–110

[48] Jaramillo, Carlos, *et al.* ”6-DoF pose localization in 3D point-cloud dense maps using a monocular camera.” Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on. IEEE, 2013. 1

[49] Fox, Dieter, *et al.* ”Monte carlo localization: Efficient position estimation for mobile robots.” AAAI/IAAI 1999 (1999): 343-349. 1

[50] Zhang, Wei, and Jana Kosecka. ”Image based localization in urban environments.” 3D Data Processing, Visualization, and Transmission, Third International Symposium on. IEEE, 2006. 2

[51] Nistr, David, Oleg Naroditsky, and James Bergen. ”Visual odometry.” Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. Vol. 1. IEEE, 2004.2

[52] Raymond, Rudy, *et al.* ”Map matching with hidden Markov model on sampled road network.” Pattern Recognition (ICPR), 2012 21st International Conferenceon. IEEE, 2012. 2

[53] Silveira, Geraldo, Ezio Malis, and Patrick Rives. ”An efficient direct approach to visual SLAM.” IEEE transactions on robotics 24.5 (2008): 969-979. 2

[54] Smith, Randall C., and Peter Cheeseman. ”On the representation and estimation of spatial uncertainty.” The international journal of Robotics Research 5.4 (1986): 56-68. 2

[55] Michael, M., *et al.* ”Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges.” Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI). 2003. 2

[56] Lowe, David G. ”Distinctive image features from scale-invariant keypoints.” International journal of computer vision 60.2 (2004): 91-110. 2

[57] Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. ”Surf: Speeded up robust features.” European conference on computer vision. Springer Berlin Heidelberg,2006. 2

[58] Brubaker, Marcus A., Andreas Geiger, and Raquel Urtasun. ”Map-based probabilistic visual selflocalization.” IEEE transactions on pattern analysis and machine intelligence 38.4 (2016): 652-665. 2

[59] Forster, Christian, Matia Pizzoli, and Davide Scaramuzza. ”Air-ground localization and map augmentation using monocular dense reconstruction.” 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2013. 2

[60] Pascoe, Geoffrey, William Maddern, and Paul Newman. ”Direct Visual Localisation and Calibration for Road Vehicles in Changing City Environments.” Proceedings of the IEEE International Conference on Computer Vision Workshops. 2015. 2

[61] Gwon, Gi-Poong, *et al.* ”Generation of a Precise and Efficient Lane-Level Road Map for Intelligent Vehicle Systems.” (2015). 2

[62] Pandey, Gaurav, *et al.* ”Automatic Targetless Extrinsic Calibration of a 3D Lidar and Camera by Maximizing Mutual Information.” AAAI. 2012. 3

[63] Moo Yi, Kwang, *et al.* ”Detection of moving objects with non-stationary cameras in 5.8 ms: Bringing motion detection to your mobile device.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2013. 3

[64] Itti, Laurent, Christof Koch, and Ernst Niebur. ”A model of saliency-based visual attention for rapid scene analysis.” IEEE Transactions on pattern analysis and machine intelligence 20.11 (1998): 1254-1259. 3

[65] Berens, Jeff, and Graham D. Finlayson. ”Logopponent chromaticity coding of colour space.” Pattern Recognition, 2000. Proceedings. 15th International Conference on. Vol. 1. IEEE, 2000. 4

[66] Finlayson, Graham D., *et al.* ”On the removal of shadows from images.” IEEE transactions on pattern analysis and machine intelligence 28.1 (2006): 59-68. 4

[67] Alvarez, Jos M. lvarez, and Antonio M. Lopez. ”Road detection based on illuminant invariance.” IEEE Transactions on Intelligent Transportation Systems 12.1 (2011): 184-193. 4

[68] Mattern, Norman, and Gerd Wanielik. ”Vehicle localization in urban environments using feature maps and aerial images.” 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, 2011. 4

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

[70] J. Fritsch, T. Kuehnl, and A. Geiger, “A new performance measure and evaluation benchmark for road detection algorithms,” in International Conference on Intelligent Transportation Systems (ITSC), 2013.

[71] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[72] R. Mohan, “Deep deconvolutional networks for scene parsing,” arXiv preprint arXiv:1411.4101, 2014.

[73] L. Ankit, K. Mehmet, S. Luis, and M. Hebert, “Map-supervised road detection,” in IEEE Intelligent Vehicles Symposium Proceedings, 2016.

[74] J. Fritsch, T. Kuehnl, and A. Geiger, “A new performance measure and evaluation benchmark for road detection algorithms,” in International Conference on Intelligent Transportation Systems (ITSC), 2013.

[75] M. A. Sotelo, F. J. Rodriguez, and L. Magdalena, “Virtuous: Visionbased road transportation for unmanned operation on urban-like scenarios,” Intelligent Transportation Systems, IEEE Transactions on, vol. 5, no. 2, pp. 69–83, 2004

[76] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (VOC) challenge,” Int. J. of Computer Vision, vol. 88, no. 2, pp. 303–338, June 2010.

[77] C. Tan, T. Hong, T. Chang, and M. Shneier, “Color model-based realtime learning for road following,” in Intelligent Transportation Systems Conference, 2006. ITSC ’06. IEEE, Sept 2006, pp. 939–944.

[78] J. Alvarez and A. M. Lopez, “Road detection based on illuminant invariance,” Intelligent Transportation Systems, IEEE Transactions on, vol. 12, no. 1, pp. 184–193, 2011

[79] B. Wang, V. Fremont, and S. Rodriguez, “Color-based road detection and its evaluation on the kitti road benchmark,” in Intelligent Vehicles Symposium Proceedings, 2014 IEEE, June 2014, pp. 31–36.

[80] J. M. Alvarez, M. Salzmann, and N. Barnes, “Learning appearance models for road detection,” in Intelligent Vehicles Symposium (IEEE IV). IEEE, 2013.

[81] J. M. Alvarez, T. Gevers, Y. LeCun, and A. M. Lopez, “Road scene segmentation from a single image,” in European Conference on Computer Vision, ECCV, 2012.

[82] C. Rasmussen, “Grouping dominant orientations for ill-structured road following,” in Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, vol. 1. IEEE, 2004, pp. I–470.

[83] H. Kong, J.-Y. Audibert, and J. Ponce, “Vanishing point detection for road detection,” in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 96–103.

[84] T. Kuhnl, F. Kummert, and J. Fritsch, “Monocular road segmentation using slow feature analysis,” in Intelligent Vehicles Symposium (IV), 2011 IEEE, June 2011, pp. 800–806.

[85] J. Fritsch, T. Kuehnl, and F. Kummert, “Monocular road terrain detection by combining visual and spatial information,” Transactions on Intelligent Transportation Systems, vol. 15, no. 4, pp. 1586–1596, 2014.

[86] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. R. Bradski, “Self-supervised monocular road detection in desert terrain,” in Robot. Sci. Syst. Conf. (RSS), 2006.

[87] Y. Alon, A. Ferencz, and A. Shashua, “Off-road path following using region classification and geometric projection constraints,” in Proceedings of the 2006 IEEE

Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1, ser. CVPR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 689–696.

[88] P. Y. Shinzato, V. G. Jr, F. S. Osorio, and D. F. Wolf, “Fast visual road recognition and horizon detection using multiple artificial neural networks,” in Intelligent Vehicles Symposium Proceedings, 2012 IEEE, June 2012, pp. 1090–1095.

[89] J. Shotton, J. Winn, C. Rother, and A. Criminisi, “Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation,” in Computer Vision–ECCV 2006. Springer Berlin Heidelberg, 2006, pp. 1–15.

[90] J. M. Alvarez, Y. LeCun, T. Gevers, and A. M. Lopez, “Semantic road segmentation via multi-scale ensembles of learned features,” in Computer Vision–ECCV 2012. Workshops and Demonstrations. Springer Berlin Heidelberg, 2012, pp. 586–595.

[91] C. Guo, S. Mita, and D. McAllester, “Robust road detection and tracking in challenging scenarios based on markov random fields with unsupervised learning,” vol. 13, no. 3, pp. 1338–1354, Sep 2012.

[92] Z. He, T. Wu, Z. Xiao, and H. He, “Robust road detection from a single image using road shape prior,” in Image Processing (ICIP), 2013 20th IEEE International Conference on, Sept 2013, pp. 2757–2761.

[93] R. Labayrade, D. Aubert, and J.-P. Tarel, “Real time obstacle detection in stereovision on non flat road geometry through ”v-disparity” representation,” in Intelligent Vehicle Symposium, 2002. IEEE, vol. 2, June 2002, pp. 646–651.

[94] H. Badino, U. Franke, and R. Mester, “Free space computation using stochastic occupancy grids and dynamic programming,” in Workshop on Dynamical Vision, ICCV, Rio de Janeiro, Brazil, vol. 20, 2007.

[95] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, *et al.*, “Stanley: The robot that won the darpa grand challenge,” in The 2005 DARPA Grand Challenge. Springer, 2007, pp. 1–43.

[96] F. Moosmann, O. Pink, and C. Stiller, “Segmentation of 3d lidar data in non-flat urban environments using a local convexity criterion,” in Intelligent Vehicles Symposium, 2009 IEEE. IEEE, 2009, pp. 215– 220.

[97] T. Chen, B. Dai, R. Wang, and D. Liu, “Gaussian-process-based realtime ground segmentation for autonomous land vehicles,” Journal of Intelligent and Robotic Systems (JINT), vol. 76, pp. 563–582, Sep 2013.

[98] G. B. Vitor, D. A. Lima, A. C. Victorino, and J. V. Ferreira, “A 2d/3d vision based approach applied to road detection in urban environments,” in Intelligent Vehicles Symposium (IV), 2013 IEEE, 2013, pp. 952–957.

[99] P. Y. Shinzato, D. F. Wolf, and C. Stiller, “Road terrain detection: Avoiding common obstacle detection assumptions using sensor fusion,” in Intelligent Vehicles Symposium Proceedings, 2014 IEEE, June 2014, pp. 687–692.

[100] X. Hu, S. A. R. F., and A. Gepperth, “A multi-modal system for road detection and segmentation,” in Intelligent Vehicles Symposium Proceedings, 2014 IEEE, June 2014, pp. 1365–1370.

[101] Teichmann, Marvin, *et al.* ”MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving.” arXiv preprint arXiv:1612.07695 (2016).

[102] Mohan, Rahul. ”Deep deconvolutional networks for scene parsing.” arXiv preprint arXiv:1411.4101 (2014).

[103] Caltagirone, Luca, *et al.* ”Fast lidar-based road detection using convolutional neural networks.” arXiv preprint arXiv:1703.03613 (2017).

[104] Xiao, Liang, *et al.* ”Crf based road detection with multi-sensor fusion.” Intelligent Vehicles Symposium (IV), 2015 IEEE. IEEE, 2015.

[105] L. Xiao, R. Wang and B. Dai, ”Hybrid Conditional Random Field based Camera-LIDAR Fusion for Road Detection”, Journal of Information Sciences, 2016.

초 록

기계 및 전자 기술의 발달로 운전자의 개입 없이 주변 환경을 인식하고 주행 상황을 판단해 차량을 제어하는 자율 주행 자동차가 현실로 다가오기 시작했다. 본 논문집은 카메라 센서를 활용하여 자율 주행 차량에 필요한 영상 처리 기법에 대한 연구결과를 다루었다. 첫 번째 문제는, 선행 차량에 대한 거리 측정 방법에 대한 연구이다. 스테레오 카메라를 활용하여 주파수 영역에서의 분석을 통해 정확한 거리를 추정하는 방법을 고안하여 문제를 해결하였다. 두 번째 문제는, 고정밀 3차원 정밀 지도 상에서 단안 카메라를 활용하여 위치를 추정하는 방법에 대한 연구이다. 움직이는 물체를 제거하고 다양한 조명 상황에서 강건하게 위치를 파악할 수 있도록 상호 정보량을 활용하여 문제를 해결하였다. 마지막 문제는, 주행 가능 영역을 탐지하는 방법에 대한 연구이다. 카메라와 라이다 센서 융합을 통해 특징 추출 및 변위 지도를 생성하여 문제를 해결하였다.

주요어: 거리 탐지, 차량 측위, 주행 가능 영역 탐지, 자율 주행, 영상 처리

학번: 2011-20885