공학박사학위논문

도심 도로 상황에서 어라운드뷰 모니터링 시스템 기반 위치 추정 및 모델 예측 기반 자율 주행 차량 제어

Around View Monitoring System based Vehicle Localization and Model Predictive Control for Automated Vehicle in Urban Roads

2016년 8월

서울대학교 대학원

기계항공공학부

김 동 욱

Abstract

Around View Monitoring System based Vehicle Localization and Model Predictive Control for Automated Vehicle in Urban Roads

Dongwook Kim
School of Mechanical and Aerospace Engineering
The Graduate School
Seoul National University

Recently, major corporations have announced plans to begin selling automated vehicles in a few years, and some jurisdictions have passed legislation to allow such vehicles to operate legally on public roads. Even, some leading companies release new vehicles providing an opportunity to experience partially automated driving.

Distronic Plus (Mercedes-Benz), Driving Assistant Plus (BMW), Highway Driving Assist (Hyundai Motor Company) are typical examples of partially automated driving system. With these system, a car can maintain a distance from a vehicle in front of it without the driver doing anything. It only works when the driver holds the steering wheel. When the driver takes their hands off for a certain time, the function is disabled. In spite of operating limit, they must be offering increased comfort and safety.

The main reason of limited operation is that the environment sensing system available for production cars still do not reach a satisfying level of development in terms of robustness and availability at various road conditions. Actually, many of these problem could be solved by using the current state-of-the-art sensor such as highly accurate inertial navigation systems and 3D scanning laser rangefinders However, integrating them into cars will increase the price and represent yet another barrier to adoption.

Therefore, this dissertation focused on developing a fully automated driving algorithm which is capable of automated driving in urban roads while a chosen sensor configuration is closer to current automotive serial production in terms of cost and technical maturity than in many autonomous vehicles presented earlier. Mainly two research issues are considered: a lane-level localization and a model predictive vehicle control.

The lane-level localization implies positioning the vehicle with centimeter-level accuracy with respect to a map. In order to achieve a satisfactory level of position accuracy with a low-cost GPS, a sensor fusion approach is essential for lane-level localization. The proposed sensor fusion

approach for the lane-level localization of a vehicle uses an Around View Monitoring (AVM) module and vehicle sensors. The proposed algorithm consists of three parts: lane detection, position correction, and localization filter. In order to detect lanes, a commercialized AVM module is used. Since this module can acquire an image around the vehicle, it is possible to obtain accurate position information of the lanes. With this information, vehicle position can be corrected by the iterative closest point (ICP) algorithm. This algorithm estimates the rigid transformation between the lane map and lanes obtained by AVM in real-time. The vehicle position corrected by this transformation is fused with the information of vehicle sensors based on an extended Kalman filter (EKF). For higher accuracy, the covariance of the ICP is estimated using Haralick's method.

In contrast with highway, automated vehicles are allowed a relatively short headway distance in urban roads. Accordingly, the information from the environment sensor such as radars, lidars and cameras was very limited due to neighboring vehicles. In such situations, suddenly appeared obstacles may put automated vehicles in danger. To overcome this problem, preceding vehicle's behavior information was used for automated vehicle control in this dissertation. Preceding vehicle Behaviors including a yaw motion was precisely estimated using front lidars. It makes possible to generate the trajectory for following the preceding vehicle. An optimal vehicle following problem while avoiding collision with obstacles is formulated in terms of cost minimization under constraints. To solve this minimization problem, we use model predictive control approach.

The performances of the proposed localization and control algorithm of automated vehicle are verified via computer simulations and vehicle tests. Test results show that the proposed methods can achieve centimeter-level localization accuracy and robustness of automated vehicle control system in urban roads.

Keywords: Automated driving vehicle, Vehicle localization, Lane map generation, Map-matching, Iterative closest point, Model predictive control, Target detection and tracking

Student Number: 2008-22869

Contents

List of Figures

Chapter 1

Introduction

1.1. Background and Motivation

Driver assistance systems, such as active cruise control (ACC) or lane departure warning (LDW) have become a common option in premium vehicles in recent years. With each new model, new driver assistance systems are being introduced. Distronic Plus (Mercedes-Benz), Driving Assistant Plus (BMW), Highway Driving Assist (Hyundai Motor Company) are examples of these systems. With these system, a car can maintain a distance from a vehicle in front of it without the driver doing anything. It only works when the driver holds the steering wheel. When the driver takes their hands off for a certain time, the function is disabled. It has been proven that driver assistance systems offer greater safety to the driver and the surrounding traffic [Karush 12]. The pinnacle of driver assistance systems is automated driving.

According to NHTSA automated driving system can be classified into four different levels. The levels of automation described below provide a summary of the NTSA definitions [NHTSA 13].

Level 1 – Function-specific Automation: Automation of specific control functions, such as cruise control, lane guidance and automated parallel parking. Drivers are fully engaged and responsible for overall vehicle control (hands on the steering wheel and foot on the pedal at all times).

Level 2 - Combined Function Automation: Automation of multiple and integrated control functions, such as adaptive cruise control with lane centering. Drivers are responsible for monitoring the roadway and are expected to be available for control at all times, but under certain conditions can disengaged from vehicle operation (hands off the steering wheel and foot off pedal simultaneously).

Level 3 - Limited Self-Driving Automation: Drivers can cede all safety-critical functions under certain conditions and rely on the vehicle to monitor for changes in those conditions that will require transition back to driver control. Drivers are not expected to constantly monitor the roadway.

Level 4 - Full Self-Driving Automation: Vehicles can perform all driving functions and monitor roadway conditions for an entire trip, and so may operate with occupants who cannot drive and without human occupants.

The above mentioned driver assistance systems correspond to level 2. In recent years, many projects from automobile manufacturers, automotive suppliers, universities and research institutes across the world have been developing prototype vehicles for level 3 automated driving. These research

efforts are summarized [Broggi 14] in table 1.1.

Table 1.1 Automated vehicle tests

Test	Scenario	Approach
DARPA Urban Challenge 2007 [Urmson 08]	Complex course in a staged city environment	Mission based, negotiating other moving traffic and obstacles while obeying traffic regulations
IVFC 2013 [IVFC 13]	Suburban and urban road test	Natural environment perception, 4s scale (safety, smartness, smoothness and speed)
Hyundai Autonomous Challenge 2014 [HAC 14]	Proving ground replicated urban roads	Follow a course and perform missions (Intersection, Pedestrian detection, Traffic sign recognition, Parking etc)
Braunschweig Stadtpilot-Project 2010 [BSP 10]	Urban environment of Braunschweigs inner ring road	Lane keeping, interaction with traffic, lane change maneuvers at speeds up to 60 kph
Google autonomous car test [Google 15]	Highway and city streets	Map based navigation, lane and speed keeping, lidar-based dynamic objects detection
Bertha Benz Memorial Route 2013 [Franke 13]	Overland passages, urban areas, small village	Stereo vision, self-localization, free-space analysis, object detection
VIAC 2010 [Broggi 12]	Real traffic road from Italy to China	Vehicle following (virtual towing), no digital maps are used
PROUD 2013 [PROUD 13]	Real traffic roads in free-way and urban scenarios	Waypoint following, Lane keeping, Obstacle avoidance, round-about, junctions, pedestrian crossing, traffic lights

Although there have been major improvements in the last decade, all aspects of the automated driving system, including perception, localization, decision-making and path planning algorithms, still need to be further developed in order to bridge the gap between robotics research and a

customer-ready system.

Therefore, this dissertation focused on developing a fully automated driving algorithm which is capable of automated driving in urban roads while a chosen sensor configuration is closer to current automotive serial production in terms of cost and technical maturity than in many autonomous vehicles presented earlier. Mainly two research issues are considered: a localization and a vehicle control.

1.1.1. Lane-level Localization using AVM Camera

Much research of vehicle localization has been done over the last decade. Early approaches of localization were based on Global Positioning System(GPS) and vehicle sensors [Bonnifait 01] [Rezaei 07]. In open space these localization methods reach the required accuracies in the centimeter range. In these scenarios any path can be defined in WGS84 coordinates and the current position of the autonomous system can be measured by a commercially available INS. Unfortunately, in typical traffic scenes several disturbing effects, i.e. occlusions by vegetation and buildings or multi-path effects due to reflections, render this approach as not feasible. For this reason, a vehicle localization based on high-definition map [Tao 13] [Gruyer 14] has been in the limelight recently. A digital map is built that includes sufficient information to determine the current position of the autonomous vehicle relative to this map. Thus, the digital map is used as a powerful additional

sensor to improve the performance of the vehicle localization. Typical examples of map-based localizations are Google [Levinson10] and Mercedes-Benz [Schreiber 13].

In previous work, Google builds a bird's eye image of 3D laser reflections on the road surface and Benz detect lanes and curbs using stereo camera. And they determine the current position by matching the current sensor readings with the recorded and filtered data. Both system achieve sufficient localization precision within centimeter-range for automated driving. However, 3D laser scanners and stereo cameras can only be a research solution since costs and design make them unattractive for the use in consumer vehicles.

In order to overcome this issue, an Around View Monitoring(AVM) camera was used for vehicle localization in this dissertation. AVM cameras are pervasively used in driver assistance systems in mid-class vehicles. And an AVM camera have further merit comparing 3D scanners and stereo cameras. First, the top-view image obtained by AVM enables direct calculation of the lateral offset without any model for lane tracking. The tracking models may sometimes result in inaccurate lateral offset due to road conditions, especially a road slope. Second, it is barely affected by invisible lane images caused by neighboring vehicles even in heavy traffic. There are few neighboring vehicles that invade AVM's field of view (FOV) in the same situation. Third, AVM cameras are more robust to weather and illumination conditions than front cameras, because AVM cameras are mounted toward the ground.

1.1.2. MPC based Automated Vehicle Control in Urban Roads

In developing an automated driving system, it is required to operate in a wide operating region and limit the set of permissible states and inputs. Therefore model predictive control (MPC) approach has been used widely because of its capability to handle system constraints in a systematic way and adaptability for various driving scenarios [Anderson 10, Falcone 08, Gray 13].

Due to the fact that traditional MPC generates control updates relatively slowly, MPC has been mostly relegated to the industrial process control industry, where it has seen significant success [Qin 96]. Autonomous vehicle guidance can be posed as a kinodynamic motion planning problem, which is defined as solving for an optimal trajectory between a start and an end state while avoiding obstacles and respecting a system model as well as input and state constraints [Donald 93]. In order to make MPC applicable to vehicle control systems, the speed of MPC has to be significantly increased.

The past ten years have seen the development of modifications to MPC that address computational delay. [Allgöwer 00] and [Findeisen03] extended the MPC approach as well as its stability theory to account for computational delay. In addition, a number of fast MPC strategies have been developed by [Diehl 02, Ohtsuka 04, Nagy 07]. However, it is hard to apply these

algorithms to automated vehicle control system. And these algorithms are not verified their performance in various constraint conditions.

If the constraints change dramatically comparing to previous time step, MPC requires long calculation time. Most of these case may be due to suddenly appeared obstacle in front of the vehicle. In such cases, using preceding vehicles information as initial guess of MPC can minimize the variations of constraints. And it makes calculation time of MPC shortened. In this dissertation, we use above approaches for robust and safe control of an automated vehicle.

1.2. Thesis Objectives

Mainly two research issues are considered in this dissertation: a lane-level localization and MPC based automated vehicle control.

A lane-level localization focused on positioning the vehicle within centimeter-range accuracy using low-cost sensors such as AVM camera, GPS and chassis sensors. For this, three research issues are considered: lane detection using AVM camera, lane map building and position correction based on iterative closest point(ICP).

MPC based automated vehicle control focused on calculating the maneuver inputs to ensuring the safety of an automated vehicle quickly and effectively. For this, two research issues are considered: Target tracking using lidars and MPC based automated steering control.

The performances of the proposed localization and control algorithm of automated vehicle are verified via computer simulations and vehicle tests.

This dissertation is structured as follows: the overall architecture of the proposed automated driving control algorithm is described in Section 2. Section 3 presents the lane-level localization using AVM camera. Section 4 presents MPC based automated vehicle control. The contribution of this research and introduction of future works are summarized in Section 5.

Chapter 2

Overall Architecture of an Automated Driving System in Urban Roads

2.1. Automated driving architecture

The overall architecture of our automated driving system is outlined in Fig. 2.1. The environment representation system consists of three main modules: vehicle localization, static obstacle map construction, and moving obstacle tracking.

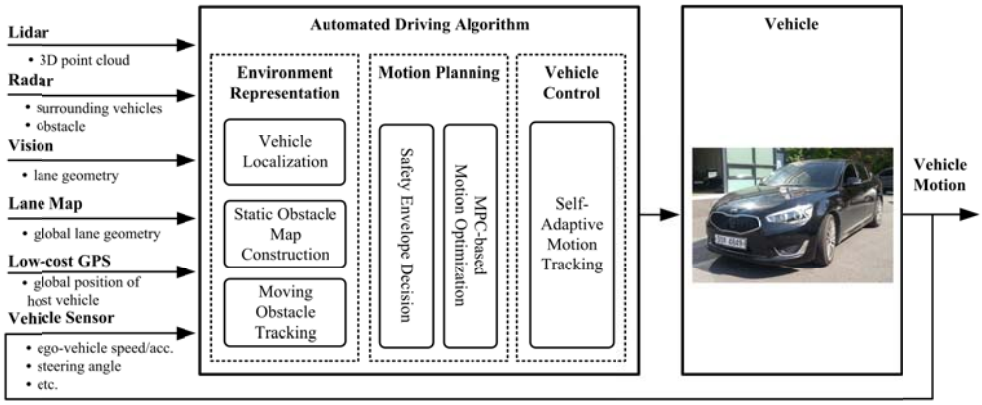


Figure 2.1 Overall architecture of automated driving system

All system modules make use of information from the various sensors. The main sensing components are radar, LiDAR, vision, low-cost GPS, and proprioceptive sensors. The selected sensor configuration is closer to current automotive serial production in terms of cost and technical maturity than in many autonomous robots presented earlier. The objective of the motion planning modules is to derive an optimal trajectory as a function of time, from the environment representation results. A safety envelope definition module determines the complete driving corridor that leads to the destination while assigning all objects to either the left or right corridor boundary. In the case of moving objects such as other traffic participants, their behaviors are anticipated in the near future. An optimal trajectory planner uses the safety envelope as a constraint and computes a trajectory such that the vehicle stays within its bounds. The vehicle control module is fed back to the position estimate of the localization module to guide the vehicle along the planned trajectory.

2.2. Test Vehicle Configuration

Our test vehicle is equipped with close-to-production sensors and a referencing system. We have a radar and two single-layer LiDARs mounted on the front bumper. For lane detection, a monocular vision system was mounted on the windshield and AVM cameras were mounted on each side of the vehicle. A low-cost GPS was also mounted for localization, as well as a RTK-GPS receiver for the mobile mapping process and ground truth. The RTK-DGPS is completely independent of the GPS input to the system. The low-cost GPS and DGPS have accuracies of about 2.5m and 0.02m CEP (Circular error probable), respectively. The actuator module contains steering, throttle and brake actuators. These systems are interfaced using the control area network (CAN) bus. The command signals are transmitted digitally. The controller consists of a computer and micro-autobox. The complete sensor, actuator and controller setup is shown in Fig. 2.2

Figure 2.2 Test vehicle configuration

To power this equipment, an additional sub-battery is installed in the trunk space, and the sub-battery has a mechanism to be charged by an alternator. Moreover, a 220 V, 2000 W inverter system is also installed. For lane-level localization, we use proprioceptive sensors (velocity, yaw-rate) and AVM cameras.

Chapter 3

Lane-Level Localization using an AVM camera

Recently, automated driving was become widely regarded as mainstream in the automotive industry since it offers increased comfort and safety. Many motor vehicle manufacturers aim to commercialize self-driving cars by 2020 and are spurring intelligent vehicle research to realize this. Among these research activities, localization has emerged recently as one of the hottest issues in the development of autonomous vehicles.

In open space high precision Inertial Navigation Systems (INS) combining an Inertial Measurement Unit (IMU) and a global navigation satellite system (GNSS) reach the required accuracies in the centimeter range. In these scenarios, these GNSS satisfy the required performance for autonomous driving. However, the sole use of differential GNSS receivers cannot guarantee the required accuracy always, particularly if GNSS outages exist. Indeed, in urban areas, high-rise buildings and monuments block often the

GPS satellites and induce multipath and GPS outages.

Therefore, a localization system must be independent of satellite systems. For this reason, there is much interest in the vehicle localization based on digital maps. These maps are built that includes sufficient information to determine the current position of the autonomous vehicle relative to this map. Thus, the digital map is used as a powerful additional sensor to improve the performance of the vehicle localization.

In previous work, Google builds a bird's eye image of 3D laser reflections on the road surface and Benz detect lanes and curbs using stereo camera. And they determine the current position by matching the current sensor readings with the recorded and filtered data. Both system achieve sufficient localization precision within centimeter-range for automated driving. However, 3D laser scanners and stereo cameras can only be a research solution since costs and design make them unattractive for the use in consumer vehicles.

In order to overcome this issue, an Around View Monitoring(AVM) camera was used for vehicle localization in this dissertation. AVM cameras are pervasively used in driver assistance systems in mid-class vehicles. And an AVM camera have further merit comparing 3D scanners and stereo cameras. This is described in more detail on section 3.4.

3.1. Related Research

Much research of map-based localization has been done over the last decade. The progress of Google's fleet of map-based autonomous vehicles [Thrun 10] and the recent purchase of Nokia HERE maps by a consortium of automotive manufacturers [Trenholm 15] illustrate the importance of prior maps for autonomous road vehicles. Key requirements for these algorithm will be a digital map, a method of map-matching and a localization filter.

The digital maps used for localization can be mainly classified according to features for map-matching. The most commonly used feature is a lane marking. Lane markings are in standard use and exist on almost all roads. Thus, many researches [Schreiber 13, Tao 13, Suganuma 11, Cui 14] use lane marking data to build their digital map. The second most used feature is a curb [Schreiber 13, Hata 14]. Curbs usually appear at the borders between streets and sidewalks. They are another important feature to determine the drivable area. Besides these two features, 3D features [Chong 14], keypoints [Zlot 14], visual features [Wang 14] and GPS shadow [Wang 13] are also considered to generate the digital map for the map-aided localization. Based on the works above, it is apparent that map with these features enhances the performance of vehicle localization. However, features other than lane markings still have problems with the standardization of the feature map. This is because lane markings have a consistent appearance and are usually painted based on rules given by the government. In contrast, other features have many exceptions that make it difficult to build a large map. To detect other features,

high-cost sensors such as a stereo camera [Schreiber 13] or a LiDAR [Hata 14, Chong 14, Zlot 14] are required. So, in our work, we only use lane markings as features for map-matching and analyze the performance of localization based on this feature.

To correct the vehicle position using a map and feature data obtained by sensors in real-time, a map-matching algorithm is needed. The most famous method of map-matching is the iterative closest point (ICP) algorithm initially proposed in [Besl 92]. Given two point clouds, the ICP algorithm estimates the rigid transformation between them. The ICP algorithm sets up correspondences between the source and the target point clouds. Then it finds a transformation that minimizes an error metric function and transforms the source point cloud based on this transformation. It iterates over these steps until the residual error between the source and the target point clouds is smaller than a certain threshold. There are many variants of ICP [Rusinkiewicz 01]. Among these variants of ICP, the point-to-plane approach is known to be very accurate and fast [Park 03], so we apply this approach to our algorithm. In spite of an enormous amount of research effort to improve the accuracy of ICP [Pomerleau 13], these variants still can result in inaccurate alignment due to the problems of noise and outliers. It is therefore essential to fuse the vehicle position derived from the map-matching algorithm with other sensors. To implement it properly, a precise estimate of ICP's covariance is needed. The covariance estimator of minimization algorithms such as ICP was initially proposed in [Haralick 98], and has come to be referred to as Haralick's method [Mallios 14]. Haralick's method is

based on the Hessian of the cost function with respect to the estimated displacement and the derivative of the Jacobian of the cost function with respect to the measurements. In [Censi 07], this method was adapted to ICP for the first time. In our work, we adjust this method to apply to our matching algorithm.

There is a body of work in the field of localization filters (non-linear filter): mono-model approaches (EKF, UKF, DD1, DD2: the first–order and second–order Divided Difference Filter [Norgaard 00, Lefebvre 04, Marin 14]), multi-model [Toledo 07, Ndjeng 11] and particle filter [Han 13]. As a result of improvements in estimation performance, the complexity and computational load of developed local- ization filters are increasing. Although estimation performance is an important consideration for localization filters, there are other practical issues related to the filter selection. One of the most important practical considerations is the computational load of the filter, especially for real-time applications. Based on both theoretical and empirical analyses of a particular application, the EKF is well known for its computational efficiency compared to other non-linear filters [Rhudy 13]. As long as the process or measurement model is not strongly nonlinear, a nonlinear filtering algorithm is less likely to be useful [Martinelli 08]. Because of this, our localization filter is designed based on an extended Kalman filter.

There are two main differences between the proposed localization approach and the previous works. The main difference is sensor configuration to detect map-matching features. Conventional researches use front cameras or lidars. These approaches have disadvantage in a heavy traffic situation because its

field of view are disturbed by other vehicles. In order to overcome this issue, an AVM camera was used in our approach. The other difference is consideration of an error caused by mismatching. In previous work, the error variances were only determined by sensors noises. By incorporating the error caused by mismatching as well as sensor noise in the proposed algorithm, the proposed localization performance is more enhanced.

3.2. Vehicle Localization Architecture

The vehicle localization is performed using three steps: macro-level, road-level and lane-level localization. These steps are depicted in Fig. 3.1 and Fig. 3.2.

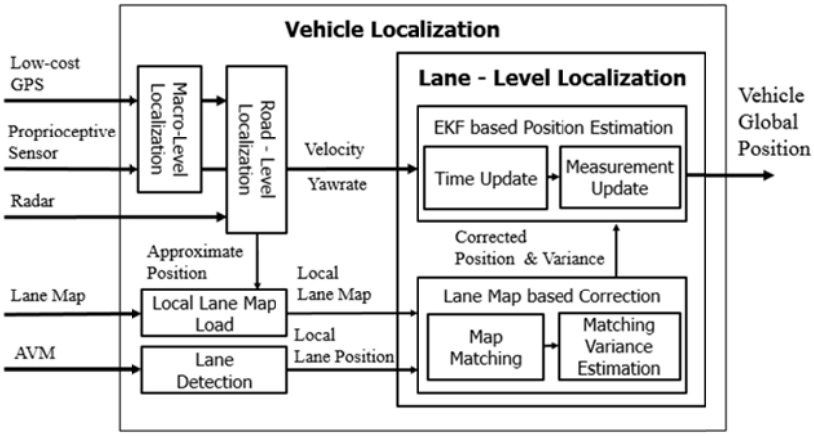


Figure 3.1 Structure of lane-level localization

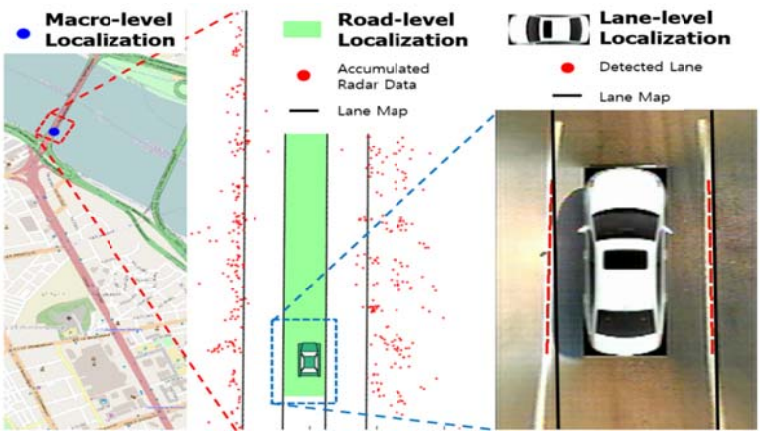


Figure 3.2 Concept of lane-level localization

Macro-level localization has already been widely used to set up car navigation systems based on low-cost GPS and proprioceptive sensors. The accuracy of these systems is on the order of 10 m [Schroedl 04]. The objective of the road-level localization is to determine the lane in which the ego-vehicle is located on a multi-lane road. This determined lane is used as the initial guess for the lane-level localization. The key idea is to recognize surrounding vehicles and road boundaries by radar under the assumption that the maximum number of lanes is known by GPS and the lane map information. The detailed description of road-level localization was presented in [Kim 14]. However, road-level accuracy of localization is not enough to control the automated vehicle on urban roads. The required precision for automated driving is within a few centimeters [Schreiber 13], i.e., lane-level localization is essential.

Three research issues are considered in lane-level localization: lane detection using AVM, position correction based on map-matching, and a localization filter. For the lane detection, we use AVM cameras enabling one to obtain a top-view image around the vehicle. There are four reasons for using an AVM module for the lane-level localization. First, the top-view image obtained by AVM enables direct calculation of the lateral offset without any model for lane tracking. The tracking models may sometimes result in inaccurate lateral offset due to road conditions, especially a road slope. Second, it is barely affected by invisible lane images caused by neighboring vehicles even in heavy traffic. As shown in Fig. 3.3, the lane markings may be occluded by other vehicles in front when using the front camera.

Figure 3.3 Comparing front camera and AVM: left (heavy traffic condition) , right (rainy condition)

However, there are few neighboring vehicles that invade AVM's FOV (Field Of View) in the same situation. Furthermore, thanks to the wide FOV, the whole straight lane marking at the side may be observed, which makes it easier and more robust to detect the lane marking [Li 08].

Third, AVM cameras are more robust to weather and illumination conditions than front cameras, because AVM cameras are mounted toward the ground. Finally, AVM cameras are pervasively used in driver assistance systems in mid-class vehicles. For these reasons, we think that an AVM camera is the most suitable sensor for lane-level localization.

However, AVM cannot be used alone due to its small field of view (FOV). Thus, map matching is required to localize the vehicle relative to the digital map. For this, the iterative closest point (ICP) algorithm is used. The ICP algorithm is widely used in spatial and geometric alignment. In this paper, we applied the point-to-plane matching method to correct the vehicle position. To solve the problem of false matching, we estimate the covariance of the ICP

algorithm and set up a validation gate in the localization filter. Finally, the localization filter is designed based on an extended Kalman filter. The corrected vehicle position obtained by map-matching is used as an observation inside a Kalman filter framework. The main contribution of this paper is the application of existing techniques to develop a low-cost system of lane-level localization and its experimental validation. Detailed explanations of each algorithm are given in the following sections.

3.3. Lidar based Lane Detection for Map Building

A field of view(FOV) of AVM camera is too narrow to build a lane map. Therefore, a 2D Lidar sensor that has wide FOV was used for lane map generation. Lane marks can be reliably extracted by their corresponding intensity value of a lidar measurement [Isogai 09]. Dark road surface reflects significantly less laser light energy than the brighter lane markers. The detailed description of lane detection is presented in following sections

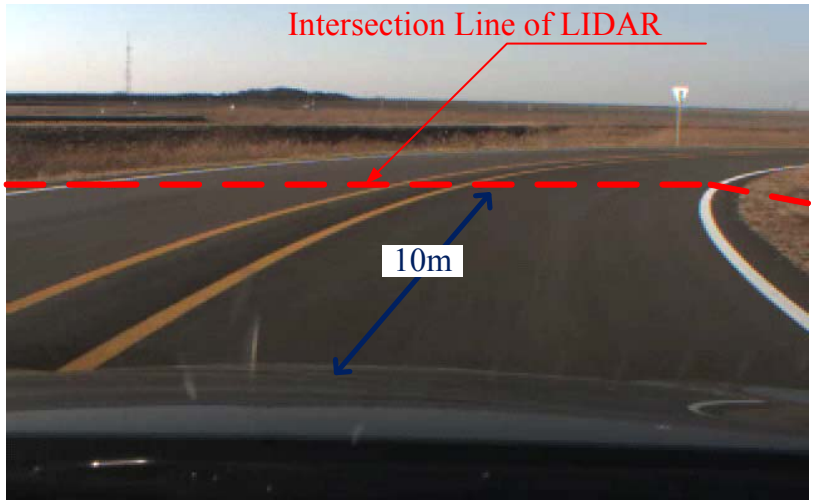
3.3.1. Sensor Configuration of a Map-building Vehicle

SICK LMS-511 are installed on the roof of the vehicle as shown in Fig.1. A layers of the lidar is facing ground and are intersecting the road surface in a distance of approximately 10m. This laser range sensors provides 180°scans (with0.25°angular resolution) up to 25 Hz scan rate. Their sensing maximum range reaches 80 m with a 50 mm error and they also can output reflectivity values.

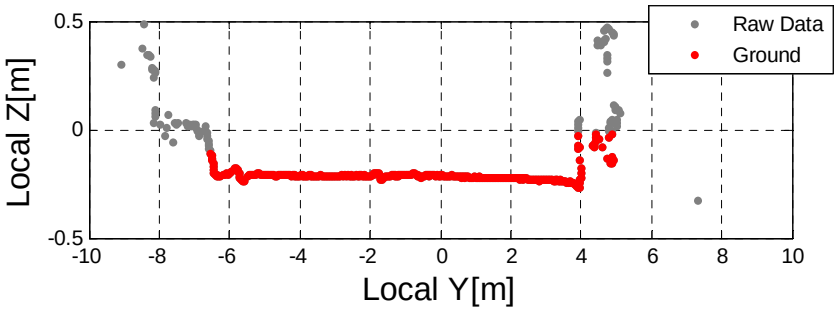
Figure 3.4 Sensor Configuration of a Map-building Vehicle

3.3.2. Road Surface Estimation

In the first step, each scan is processed individually. Looking at figures 3.5(b), where scan points are projected in the vehicle coordinates system, it can be noticed that the road is flat. So we find a polyline expression that best approximates the scan data for detecting the ground points. Using RANSAC algorithm, only two thresholds are necessary;



(a) Corresponding scenario



(b) Road surface points detected by a lidar sensor

Figure 3.5 Result of road surface estimation

the first one indicates the expected outliers rate in the points set, which is directly related to the iterations number, and the second one specifies the distance above which a point is considered as an outlier. The result of road surface estimation is presented on figures 3.5(b). Red points represent the extracted road surface point using the RANSAC algorithm. It can be seen that the road is well extracted, although some points could be wrongly classified, as they fortuitously belong to the best line.

3.3.3. Lane Marking Extraction

With a correctly extracted ground and using the reflectivity data, we can also extract lane markings trough a simple thresholding. Asphalt presents a much lower reflectivity than road markings so that threshold determination is quite easy [Dietmayer 05]. Using the reflectivity information recorded by lidars enable our system to detect lane markers in the presence of shadows, against direct sunlight and even at night. This makes the lane map much less dependent on ambient lighting than is the case for passive cameras.

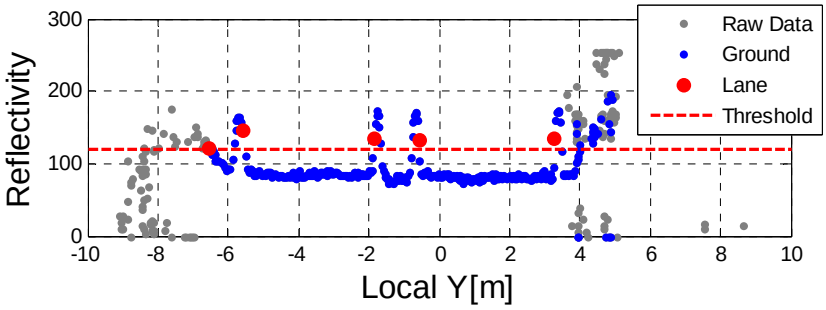


Figure 3.6 Lane marking extraction

3.3.4. Data Registration

For the absolute description of lane marking, we combined an inertial measurement unit (IMU) with a high-precision differential global positioning system (DGPS). This sensor sends the current position characterized by latitude, longitude and altitude as well the current heading. Since the sampling rate of the IMU is one order of magnitude higher than the actual scanning rate, the sended spatial resolution is sufficient without any interpolation. For the lane map building, we transform the given ego-position into UTM coordinates and, from that point on, convert all the relative positions into absolute UTM coordinates.

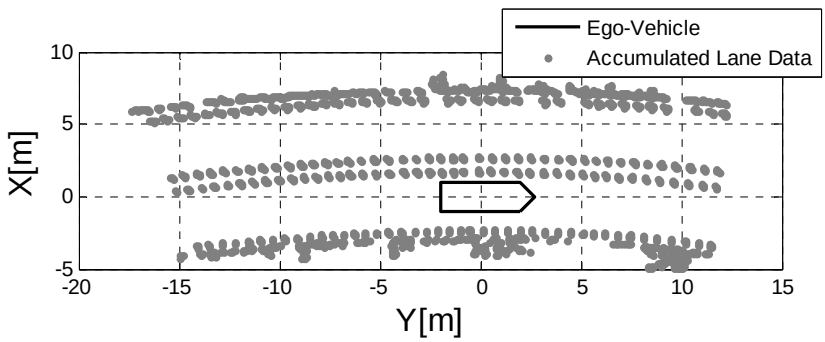


Figure 3.7 Lane Data Registration

3.4. Lane Map Building

Our localization algorithm was based on the lane map includes geo-localized lane markings. For real-time application, a limited number of parameters must represent the map line segments. However, it is very difficult to extract simplified lane data by processing the whole lane marking data at once. Lane markings have similar properties with road center-lines. This is the starting point of our algorithm. First we find a polyline expression that best approximates the road center-lines. The road center-lines are surveyed by driving as close as possible to center of road using a high-precision DGPS. Based on these polylines, we divide global lane marking data into local segment for easier lane searching. From segmented lane data, polylines of lane are approximated using Radon transform. In the following, we describe the detailed processes that are involved in the map building.

3.4.1. Road Center-line Segmentation

We need to find the simplified poly-lines of road center-lines. In other words, we need to determine the segments. The Douglas–Peucker's algorithm [Douglas 73] is used to find the shape points which divide the road center-lines into parts with different headings. This algorithm is a well-known algorithm for reducing the number of points in a curve that is approximated by a series of points. In this algorithm, the accuracy is determined by choosing suitable tolerance which is the maximal euclidean distance allowed

between the simplified segments and the origin points. If we choose a very small tolerance, the segments in a polyline can be too short, and there can be too many shape points in the polyline. For this reason, we propose to choose a tolerance of decimeter-level.

In order to further improve the accuracy of the polylines by reducing the errors effects due to the segmentation process, a least-squares algorithm is performed with every point between two adjacent shape points of the segmented polyline [Tao 13].

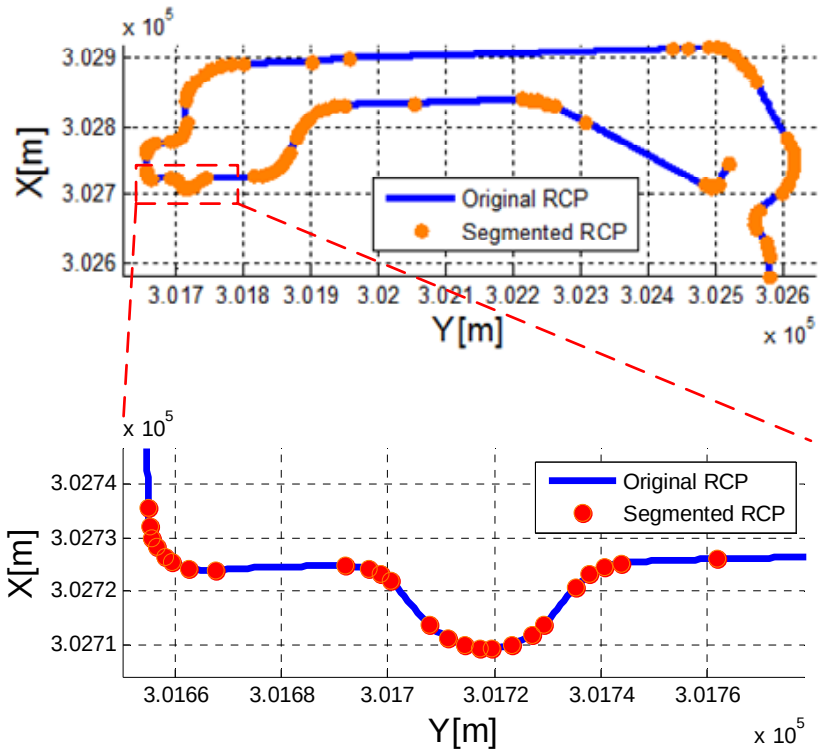


Figure 3.8 Illustration of Road Center-line Segmentation. The tolerance is 0.1m

3.4.2. Lane Approximation

For each segment determined by lateral offset distance from the road center-line as shown in figure 3.9(a), lanes are detected by applying the radon transform to segmented lane data. The Radon transform was proved to be robust against occlusions, noise and outliers. Compared to the Hough transform, the Radon transforms exhibits the advantage of a calculation time independent of the numbers of lane markers and the capability to handle gray-scale images efficiently and without thresholding [Kammel 08].

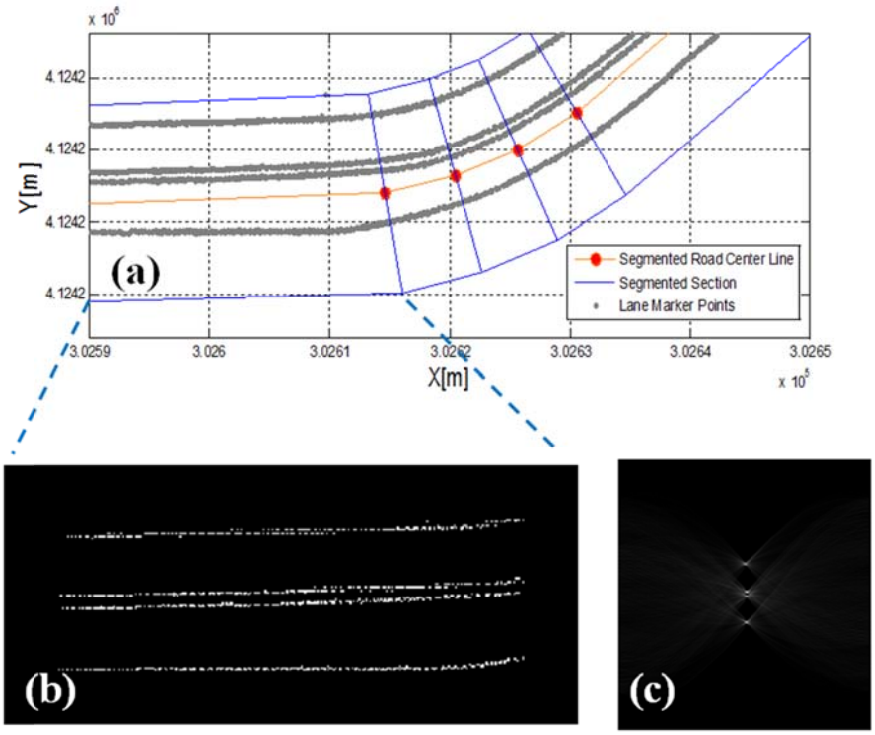


Figure 3.9 Lane Extraction (a) Global lane data, (b) Segmented lane data, (c) Radon transform of (b)

The Radon transform of segmented lane data f(x,y); denoted as g(s,; is defined as its line integral along a line inclined at an angle from the y-axis and at a distance s from the origin. Mathematically, it is written as

$$g(s,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(x\cos\theta + y\cos\theta - s)\,dx\,dy \tag{3.1}$$

where . The Radon transform is the one-dimensional projection of f(x,y) at an angle The maxima in the Radon image are considered as candidates for lane. To improve the performance of the algorithm, the number of points contributing to this possible lane determined.

3.4.3. Evaluation of Lane Maps

We built the three different type of lane maps: proving ground, local road and urban road. A proving ground is located at the Korea Automobile Testing and Research Institute (KATRI). The test track replicates a real urban environment that includes intersecting streets, pedestrian crosswalks, and traffic signals. A local road is located in an area far from downtown. This road includes signalized intersections. There was very little traffic on this road. Finally, urban road is located in the center of Seoul in Korea. This road includes signalized intersections and pedestrian crosswalks. There was heavy traffic.

Fig 3.10, 3.11 and 3.12 show lane maps of these roads. In the Fig 3.10 and 3.11, blue lines of lane maps are quite consistent with the lane markings on satellite images. It can be shown that lane maps are successfully generated.

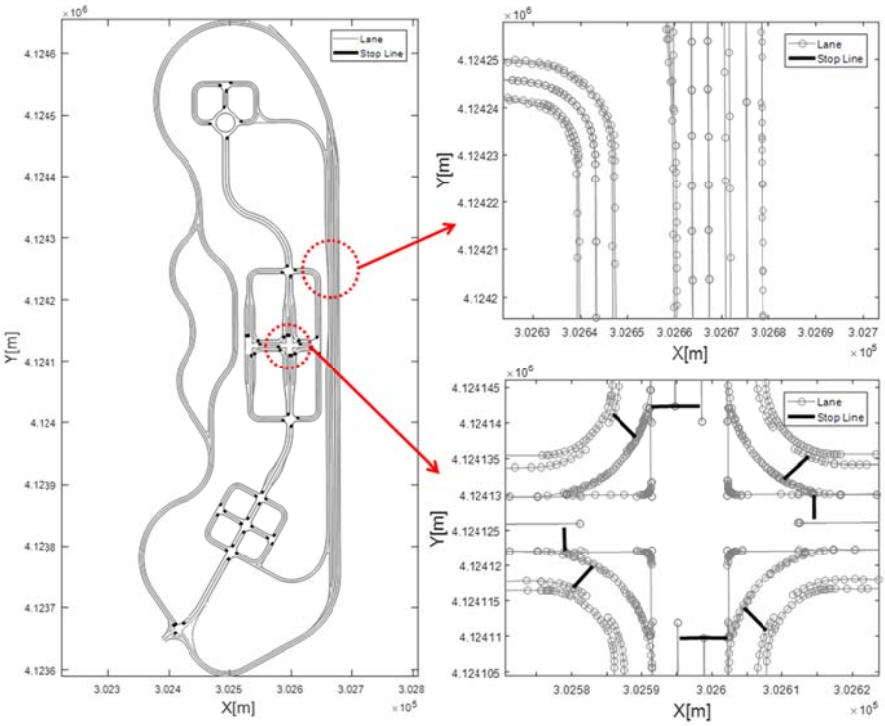


Figure 3.10 Lane map of proving ground

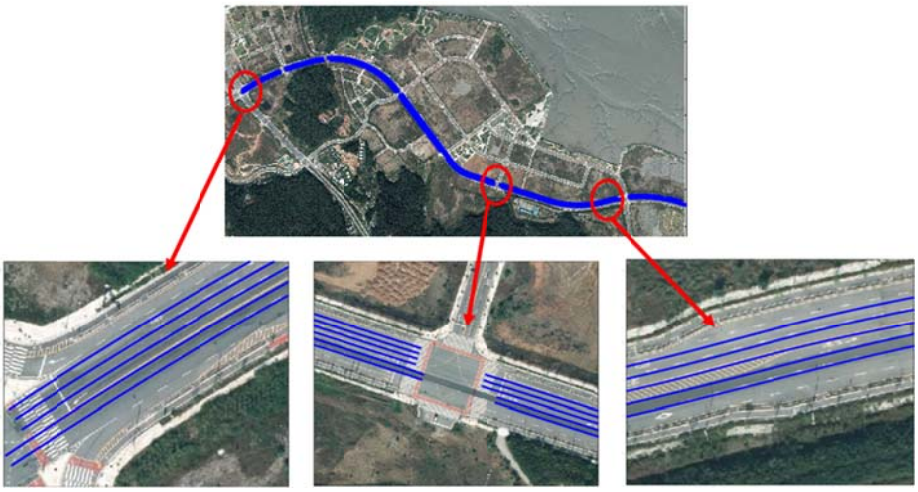


Figure 3.11 Local road lane map overlaid on satellite image

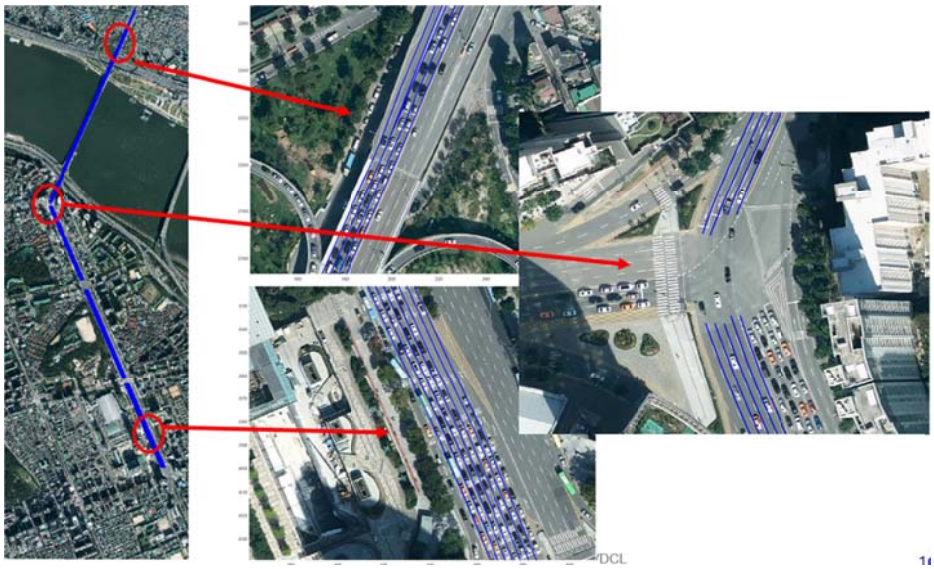


Figure 3.12 Urban road lane map overlaid on satellite image

3.5. AVM Camera based Lane Detection for Map-matching

3.5.1. Lane Detection

For detecting the lane, we use a commercial AVM module. The advantage of the AVM has already been described in Section 3.2. This module is composed of 4 fish-eye cameras mounted on each side of the vehicle as shown in Fig. 3.13. Usually, there are many steps involved the image processing of AVM cameras to obtain the top-view image around the vehicle: camera calibration, warping distortion rectification, view-point conversion, image stitching and so on [Chang 13]. However, a detailed description of these processes is skipped here, because we use a commercial AVM module that provides top-view images.

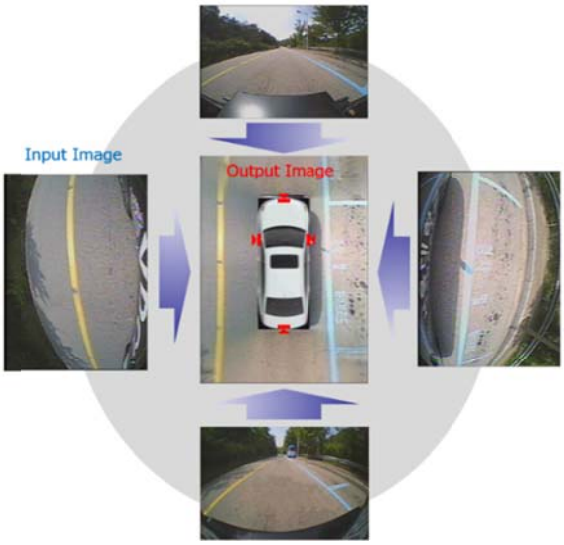


Figure 3.13 Extrinsic configuration of the four fisheye cameras and its field

of view

To improve the computational efficiency, regions of interest (ROI) were set as shown on the left of Fig. 3.14. The ROI reduce the searching area by assigning the line type (lane or stop line, vertical or horizontal line) detected in each ROI.

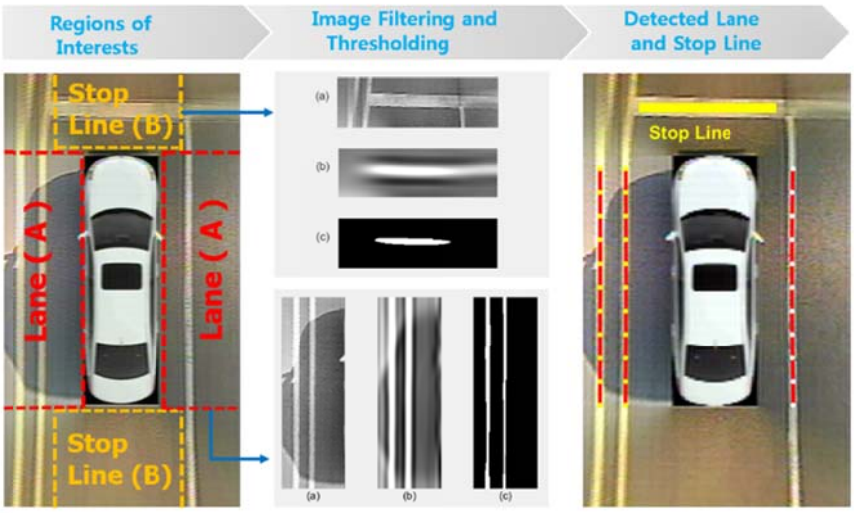


Figure 3.14 Schematic diagram of the lane and stop line detection; (a) grayscale Image (b) image after filtering (c) image after thresholding

The red and orange dashed boxes (A, B) in Fig. 3.14 indicate the ROI in order to detect lanes and stop lines. Images outside the ROI are not used due to the possibility of image distortion. Then, each region's images are filtered by a two-dimensional Gaussian kernel [Aly 08]. The vertical direction of image A and the horizontal direction of image B are smoothing Gaussians, whose $\sigma_{y,A}, \sigma_{x,B}$ are adjusted according to the required height of the lane and

stop lines to be detected.

$$f_{v,A}(y_A) = \exp\left(-\frac{y_A^{\ 2}}{2\sigma_{y,A}^{\ 2}}\right) \tag{3.2}$$

$$f_{h,\mathrm{B}}(x_B) = \exp\left(-\frac{x_B^{\ 2}}{2\sigma_{x,B}^{\ 2}}\right) \tag{3.3}$$

where f is a Gaussian function for calculating the transformation to apply to each pixel in the image. x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis. Subscript A/B and x/y represent image A, B and horizontal/vertical axis, respectively. The horizontal direction of image A and vertical direction of image B are the second derivatives of Gaussians, whose $\sigma_{x,A}, \sigma_{y,B}$ are adjusted according to the expected width of the lanes and stop lines.

$$f_{h,A}(x_A) = \frac{1}{\sigma_{x,A}^{\ 2}} \exp\left(-\frac{x_A^{\ 2}}{2\sigma_{x,A}^{\ 2}}\right)\left(1 - \frac{x_A^{\ 2}}{2\sigma_{x,A}^{\ 2}}\right) \tag{3.4}$$

$$f_{v,B}(x_B) = \frac{1}{\sigma_{y,B}^{\ 2}} \exp\left(-\frac{y_B^{\ 2}}{2\sigma_{y,B}^{\ 2}}\right)\left(1 - \frac{y_B^{\ 2}}{2\sigma_{y,B}^{\ 2}}\right) \tag{3.5}$$

Each filter A and B is adjusted for vertical (lane) and horizontal (stop line) bright lines on a dark background of a specific width. Fig. 3.14(b) show the images filtered by these Gaussian kernels. As shown in Fig. 3.14(b), areas where lanes or stop lines exist have high response. These filters can also handle quasi-vertical and quasi-horizontal lines, which produce considerable output after the thresholding process. The threshold value was determined by selecting the q% quantile value from the filtered image. The filtered image was then binarized by this threshold value. In this paper, q is set to 95%. Fig.

3.14(c) shows the result after thresholding.

Next, the pixels exceeding the threshold value are projected in the vehicle coordinate system. To find a polyline expression that best approximates the lane data, the RANSAC (Random sample consensus) [David 02] algorithm was used. Lanes and stop lines in the AVM images can be sufficiently represented by a second-order polynomial. So we use a second- order polyline for expressing the line. Two thresholds are necessary for the RANSAC algorithm; the first one indicates the expected outliers rate in the points set, which is directly related to the iterations number, and the second one specifies the distance above which a point is considered an outlier. In this research, we set the iterations number and threshold distance to determine outliers to 50 and 0.05 m, respectively. The results of lane and stop line detection are shown in Fig. 3.15. Red dashed lines represent the extracted lane marker and yellow solid lines represent the detected stop lines. It can be seen that the various lane markers are well extracted. Final outputs of the lane detection algorithm are points of the lane approximated by a second-order polynomial. These points are maintained at 10cm intervals.

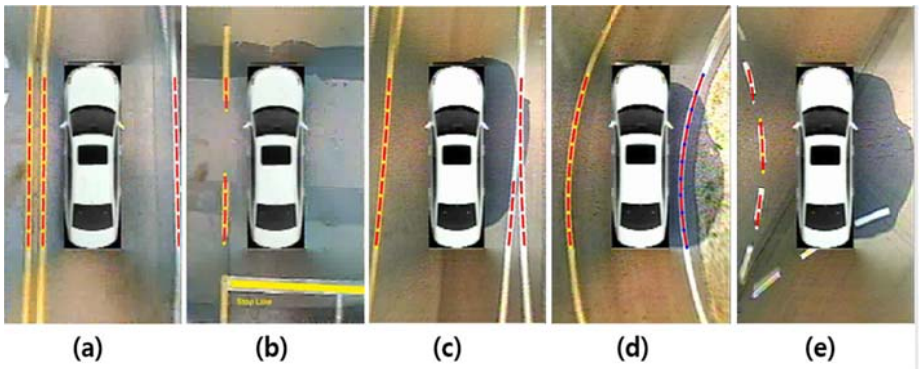


Figure 3.15 Result of various lines detection; (a) straight double and single line (b) dashed line and stop line (c) merged line (d) curved line (e) curved

and dashed line.

3.5.2. Shadow Removal in AVM Images

As shown in Fig. 3.16(d), the boundaries of the shadows have high response after filtering the grayscale image represented in Fig. 3.16(b). In this case, the proposed algorithm fails to detect the lane as depicted in Fig. 3.16(e).

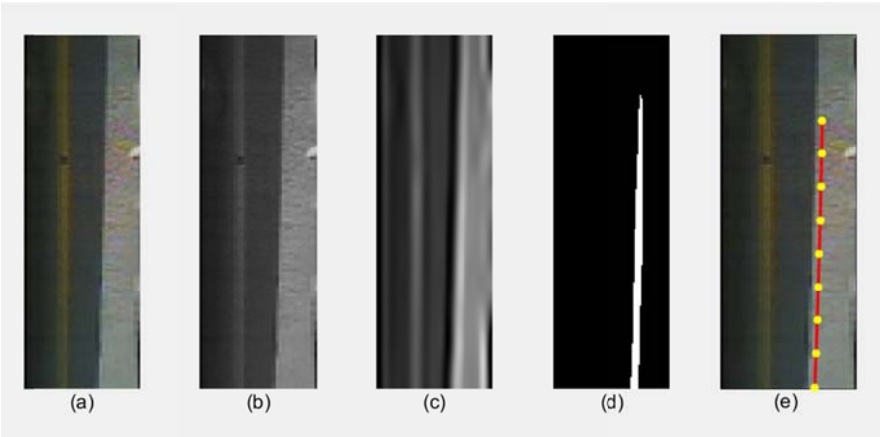


Figure 3.16 Failure of lane detection caused by the existence of shadow (a) original image (b) grayscale image (c) image after filtering (d) image after thresholding (e) failure of lane detection

Investigating the recorded image using AVM on real urban roads, there are quite a number of similar cases of missing the lane due to shadows. Thus, shadow removal in AVM images is needed for robustness of the algorithm. Many studies [Xu 06] related to this topic have already been conducted. To guarantee real-time execution, the computational load of the shadow removal algorithm is more important than the performance. Thus, we use the method proposed in [Singh 12].

This method detects the shadow using the Hue- Saturation-Value (HSV) color model. In HSV color space, the shadows have the following features.

1) low value because the direct light from the Sun is occluded by elevated objects;
2) high saturation with short blue-violet wavelength due to atmospheric Rayleigh scattering effect;
3) high hue values because shadow areas are dark.

Based on these features, shadow areas are extracted by Otsu's thresholding algorithm [Otsu 75]. After this thresholding, a binary image that represents the shadow area is obtained as shown in Fig. 3.17(b). To remove this shadow, the value of the shadow area is compensated using the buffer area around the shadow. The buffer area is estimated using morphological operations as shown in Fig. 3.17(c). If the above processes are performed correctly, we can obtain the shadow-free image represented in Fig. 3.17(d). The shadow-free image enables the lane detection algorithm to find the lane correctly as shown in Fig. 3.17(e).

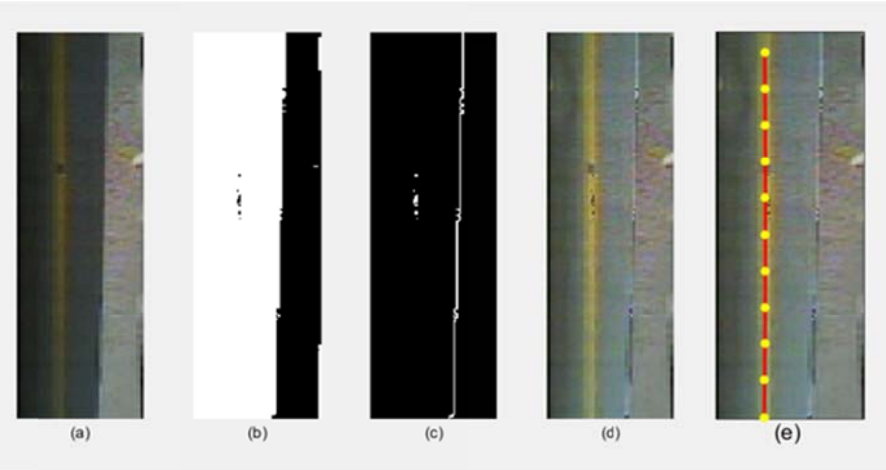


Figure 3.17 Lane detection using shadow-free Image (a) original image (b)

binary image showing shadows (c) buffer area of shadows (d) shadow free image (e) success of lane detection.

3.6. Position Correction

Position correction can be achieved by matching the digital map with the lane data obtained by AVM. The digital map used for our approach has been presented in Section 3.4. This map includes geo-localized lane markings. For real-time purposes, a limited number of parameters must represent the map line segments.

3.6.1. Map-matching based on ICP

As mentioned in Section 3.1, we use a two-dimensional point-to-plane ICP algorithm for map-matching. Point-to-plane ICP was initially introduced by Chen and Medioni [Chen, 92] and has come into widespread use as a faster and more accurate variant of standard ICP. This algorithm improves performance by using surface normal information. In common with standard ICP, the point-to-plane ICP algorithm finds the best transformation between two point clouds, iteratively repeating the following two steps until the alignment error is smaller than a set threshold.

1) compute correspondences between the two point clouds.
2) compute a transformation which minimizes the error metric function between corresponding points.

The only difference is the error metric function. Instead of minimizing

Euclidean distance between corresponding points, the point-to-plane algorithm minimizes error along the surface normal.

This error metric function can be written as

$$J = \sum_{i=1}^{N} \left\| \eta_i \cdot \left(R \cdot p_i + T - q_i \right) \right\|^2 \tag{3.6}$$

where $p_i = \left[p_{i,x}, p_{i,y} \right]^T$, $q_i = \left[q_{i,x}, q_{i,y} \right]^T$ are the N correspondences used in the iteration of ICP; and η_i is the surface normal at q_i. In our work, p_i represents the points obtained by the lane detection algorithm and q_i represents the points of the lane map with respect to ego vehicle coordinates. R, T are the rotation matrix and translation vector estimated by the ICP algorithm. Because we deal with two-dimensional ICP in this paper, R,T can be written as

$$R = \begin{bmatrix} \cos(r) & -\sin(r) \\ \sin(r) & \cos(r) \end{bmatrix}, \qquad T = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{3.7}$$

where r, t_x, t_y are the matching result (amount of angle and position correction). This result can be used to correct the vehicle position. Let (X_V, ψ) express the current vehicle position. The corrected vehicle position (X^*_V, ψ^*) can be derived [Fang 07] as:

$$\begin{aligned} X_V^* &= R_\psi \cdot T + X_V \\ \psi^* &= r + \psi \end{aligned} \tag{3.8}$$

$$where\ \ R_{\psi} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix}$$

3.6.2. Matching Covariance Estimation

Although the ICP algorithm provides a very good estimate for correcting the vehicle position, it doesn't consider its uncertainty. Calculating the covariance of ICP is essential when it has to be fused with other measurements in a stochastic localization framework. For estimating the ICP's covariance, we use Haralick's method [Haralick 98]. This method is summarized in the following proposition.

Proposition 1 : Let Z be the input/measurements and χ be the output of an algorithm A that operates on minimizing an objective function J, i.e., $\chi^* = A(\mathrm{Z}) = \arg\min_{\chi} J(\mathrm{Z}, \chi)$. Then the approximate value of the covariance of χ will be:.

$$\mathrm{cov}(\chi^*) \approx \left(\frac{\partial^2 J}{\partial \chi^2}\right)^{-1} \frac{\partial^2 J}{\partial \mathrm{Z} \partial \chi} \mathrm{cov}(\mathrm{Z}) \frac{\partial^2 J}{\partial \mathrm{Z} \partial \chi}^{T} \left(\frac{\partial^2 J}{\partial \chi^2}\right)^{-1} \tag{3.9}$$

In this paper, χ corresponds to the matching result (r, t_x, t_y) and Z corresponds to the points p_i obtained by the lane detection algorithm and the points q_i of the lane map. Thus, $\mathrm{cov}(\mathrm{Z})$ is to be set in accordance with the noise of the AVM camera and the lane map. The detailed equation for $\mathrm{cov}(\chi^*)$ can be found in [Prakhya 15].

Monte Carlo simulations were conducted to verify the performance of Haralick's method for different shapes of matching data, as shown in Fig.

3.18. The true χ is fixed and a noise-corrupted Z is created to reflect noise characteristics of an AVM and a digital map. Using the noise-corrupted Z, our map-matching algorithm estimates χ^* and the true error of matching $(\chi^* - \chi)$ is calculated.

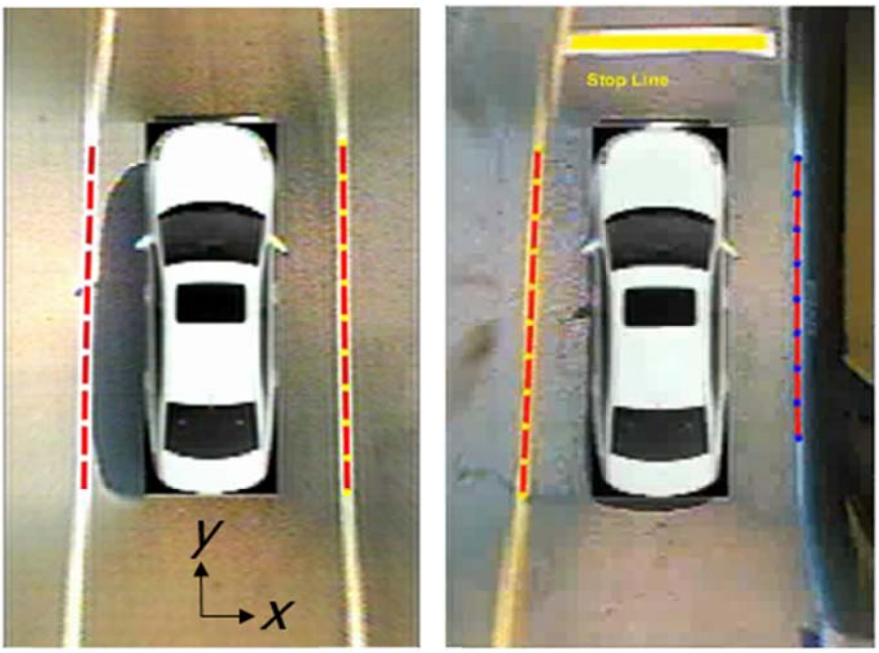
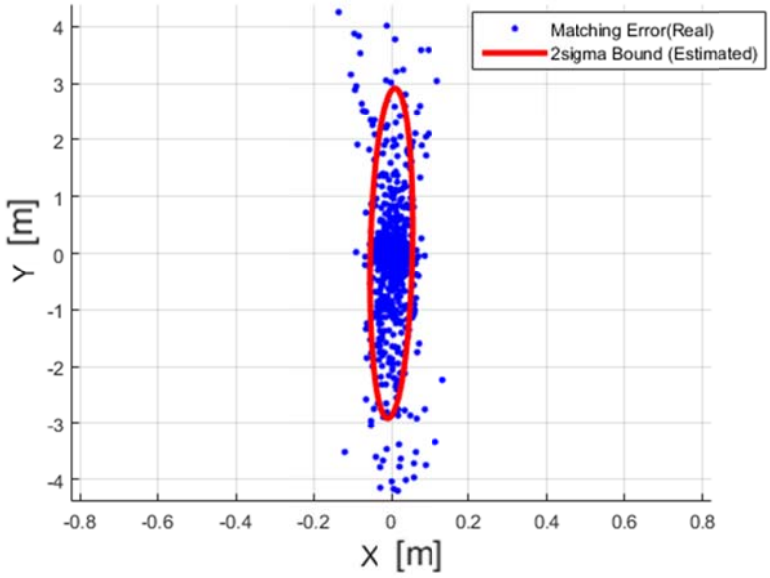
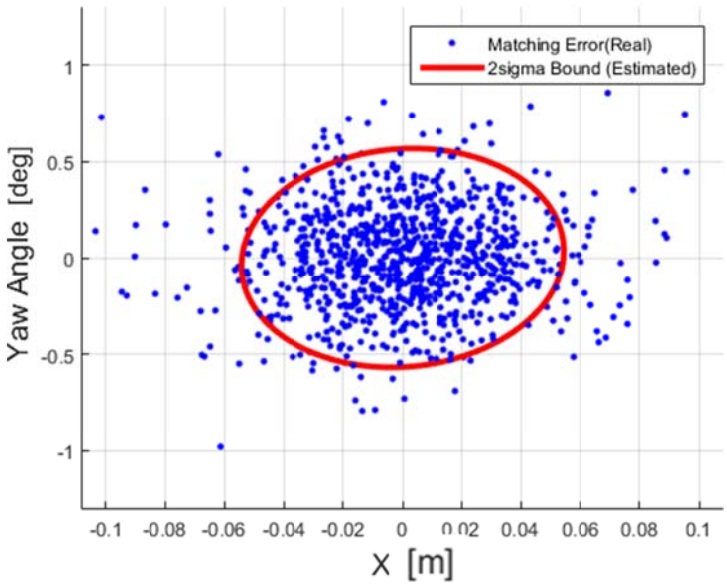


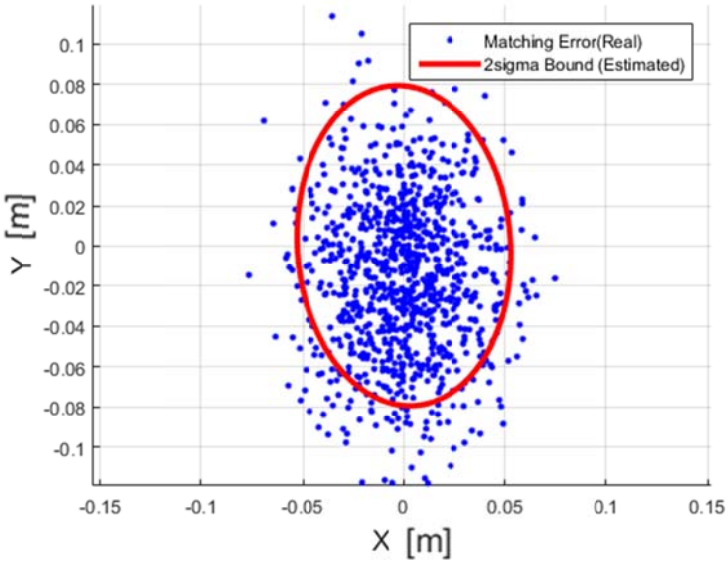
Figure 3.18 Different shape of matching data: corridor (left), U-shape (right)



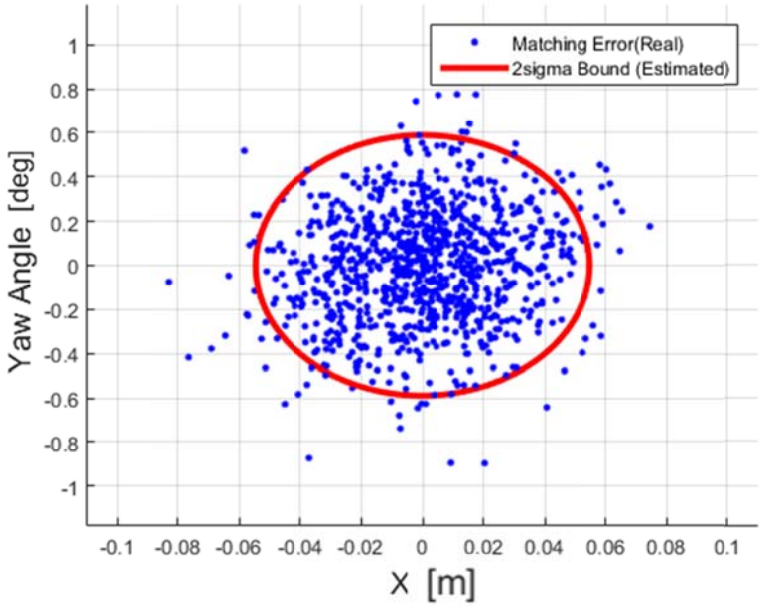
(a) X-Y error (corridor shape matching)



(b) X-Yaw error (corridor shape matching)



(c) X-Y error (U-shape matching)



(d) X-Yaw error (U shape matching)

Figure 3.19 Estimated standard deviation of matching error (red line) and

true error obtained by Monte Carlo simulation (blue dot)

True error samples obtained by repeating the above process 500 times are represented by blue dots in Fig. 3.19. Red solid lines in Fig. 3.19 show the 2-sigma bound of $\mathrm{cov}(\chi^*)$ estimated by Haralick's method. When the ICP uses the corridor shape data, it can only correct the errors in x and ψ, and not in y. For this reason, the variance in y should theoretically be infinite. The Haralick's method captures this well and shows a large uncertainty in y as shown in Fig. 3.19(a). In the case of matching U-shape data, it can fully determine translation and rotation. It can also be said that the uncertainty in x and y should be similar. This method also captures this well, as shown in Fig. 3.19(c). The standard deviations obtained by the true error and Haralick's method are summarized below in table 3.1.

TABLE 3.1 Comparing Standard Deviation of Map-Matching

Matching Shape	Corridor		U-shape	
	Haralick	True	Haralick	True
$\sigma_x(m)$	0.026	0.039	0.027	0.023
$\sigma_y(m)$	1.588	1.337	0.036	0.039
$\sigma_\psi(^\circ)$	0.296	0.271	0.294	0.258

3.7. Localization Filter

In this section, we present the key stages of the localization system. As show in Section 3.6, the result of map-matching inevitably contains some errors, due to the shape of data used for matching as well as the noise of sensors. In order to get higher accuracy, the map-matching result should be fused with proprioceptive sensor data [Madhavan 98]. The measurement of vehicle position calculated by ICP is used as an observation inside a Kalman filter framework. We also set up a validation gate in the localization filter to solve the problem of false matching.

3.7.1. Extended Kalman Filter

The ego vehicle is described by means of a point position (x, y) and orientation (ψ) in the global coordinate system, shown in Fig. 3.20

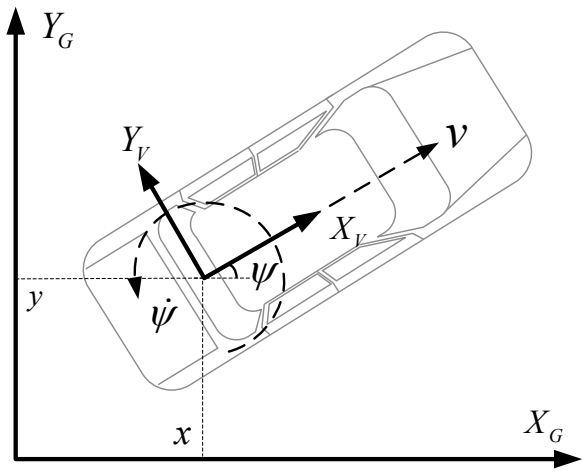


Figure 3.20 Vehicle model for prediction

The state vector is then given by

$$X = \begin{bmatrix} x & y & \psi \end{bmatrix}^T \tag{3.10}$$

The basic framework for the EKF involves estimation of the state of a discrete-time nonlinear dynamic system, shown below:

$$\begin{aligned} X_k &= f\left(X_{k-1}, \mathrm{U}_k\right) + w_k \\ Z_k &= h\left(X_{k-1}\right) + v_k \end{aligned} \tag{3.11}$$

where U_k is the known external input (velocity and yaw-rate); Z_k is the corrected vehicle position by the map-matching process. The process noise and measurement noise are given by w_k and v_k , respectively. The measurement noise is estimated by the method described in section IV.B. The process noise is associated with proprioceptive sensors. It is assumed to have zero mean and a Gaussian distribution. The specifications of the proprioceptive sensors are given in Table 3.2.

TABLE 3.2 Specification of proprioceptive sensors

Sensor	Range	Resolution	Noise (RMS)	Unit
Yaw rate	±120	0.0625	0.5	deg/s
Wheel speed	0-130	0.035	0.3	m/s

By integrating using the Euler approximation and assuming that the control signals, velocity and yaw-rate are approximately constant over the sample period, the nominal discrete process model equations can be written as:

$$\bar{X}_{k|k-1} = \begin{bmatrix} \hat{x}_{k-1|k-1} + v_k \cdot \Delta t \cdot \cos(\hat{\psi}_{k-1|k-1} + \Delta t \cdot \dot{\psi}_k) \\ \hat{y}_{k-1|k-1} + v_k \cdot \Delta t \cdot \sin(\hat{\psi}_{k-1|k-1} + \Delta t \cdot \dot{\psi}_k) \\ \hat{\psi}_{k-1|k-1} + \Delta t \cdot \dot{\psi}_k \end{bmatrix} \tag{3.12}$$

where $\hat{x}_{k-1|k-1}, \hat{y}_{k-1|k-1}, \hat{\psi}_{k-1|k-1}$ are estimated states from the previous time step and k is the time index of the discrete model. The covariance of the predicted state is described as:

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k + G_k Q G_k$$
$$where \;\; F_k = \left.\frac{\partial f}{\partial x}\right|_{\hat{X}_{k-1|k-1}, u_k}, \;\; G_k = \left.\frac{\partial f}{\partial u}\right|_{\hat{X}_{k-1|k-1}, u_k} \tag{3.13}$$

where Q describes the covariance matrix related to the proprioceptive sensor's noise.

Vehicle positions are corrected by a measurement update of the extended Kalman filter as follows:

$$\hat{X}_{k|k} = \bar{X}_{k|k-1} + K_k \cdot \left(Z_k - H \cdot \bar{X}_{k|k-1} \right)$$

$$K_k = P_{k|k-1} H^T \cdot S_k^{-1} \qquad where \;\; H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.14}$$

$$S_k = H P_{k|k-1} H^T + R_k$$

where R_k describes the covariance matrix of the measurement.

The covariance of the estimated state is described as,

$$P_{k|k} = \left(I - K_k H \right) \cdot P_{k|k-1} \tag{3.15}$$

The rate of time and measurement update is 10 Hz.

3.7.2. Validation Gate

A validation gate is set up to prevent fusing the result of false matching. The validation gate represents a threshold that is associated with the acceptability of the measurements. Only measurements inside of the validation gate are used to update the filter [SHALOM 87]. The validation gate can be obtained as follows

$$e^2 = (Z_k - H\bar{X}_{k|k-1})^T S_k^{-1} (Z_k - H\bar{X}_{k|k-1})$$

$$V_k = \left\{ Z : e^2 < g^2 \right\} \tag{3.16}$$

where g^2 is chosen as a confidence level. A confidence level is generally chosen between 1 and 3. In this paper, we set this value to 3. A three-sigma gate is commonly used, ensuring that the measurement will fall in the gate with a probability of 0.998 under the Gaussian assumption. Normalized error e^2 varies as a Chi-Squared distribution with the number of measurement

degrees of freedom

3.8. Test Results

3.8.1. Proving Ground Tests

In order to validate the proposed algorithm, tests have been carried out at the Korea Automobile Testing and Research Institute Proving ground. The test track replicates a real urban environment that includes intersecting streets, pedestrian crosswalks, and traffic signals as shown in Fig. 3.21. Since the lane-level map was generated using RTK-DGPS, the accuracy of the digital road map in an absolute coordinate system was on the order of centimeters. For real-time implementation, a lane-level localization algorithm was built with LabVIEW software installed on the computer. A test driver manually drove the test vehicle along fifteen different trajectories represented in Fig 3.22. In keeping with general urban driving conditions, the speed and acceleration of the test vehicle were restricted to 60 kph and 2 m/s^2. The total distance and driving time of the tests were about 16.1 km and 32 minutes. All data sets have in common, that ground truth data are obtained by RTK-DGPS.

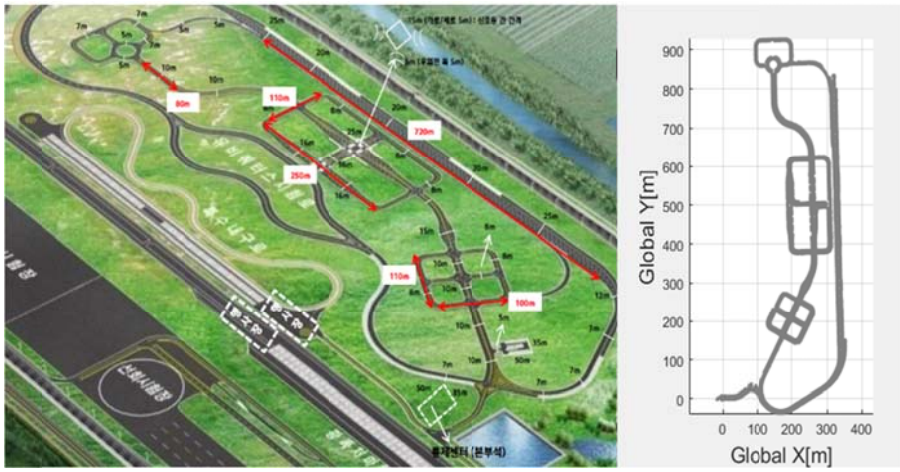


Figure 3.21 Map of a Test Site in Korea Automobile Testing and Research Institute (KATRI) Proving ground.

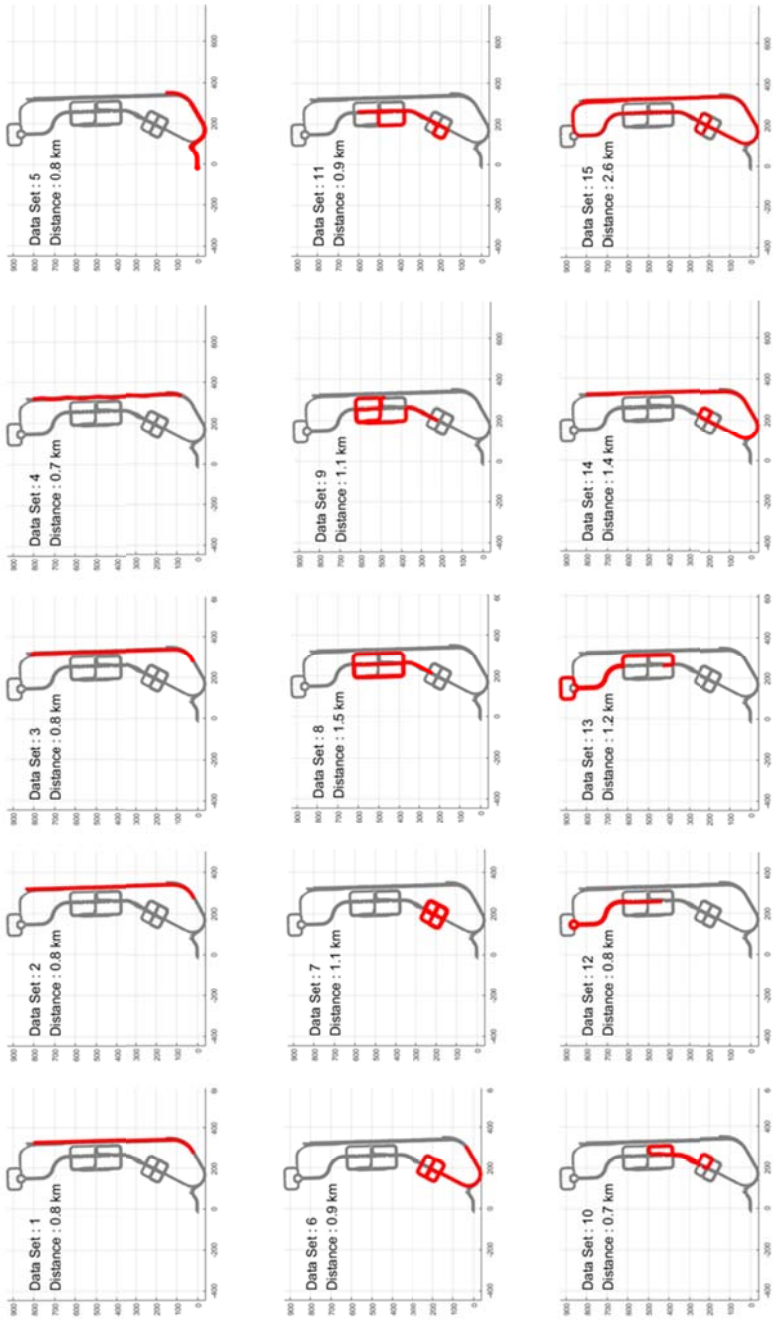


Figure 3.22 Trajectories of Data Sets for Validating the Proposed Localization Algorithm

Fig. 3.23, 3.24 and 3.25 represent the result of two typical test scenarios. The first scenario is lane keeping in the various environments such as intersection, crosswalk, roundabout and merging/ splitting roads. The second scenario is consecutive lane changing. Most previous works validated their localization algorithm only under the lane keeping condition. However, lane changes occur frequently on urban roads. Therefore, it is necessary to validate the proposed algorithm under the lane change condition. Through the results of the two tests, we can verify that the proposed localization algorithm is suitable for automated driving in urban environments. The detailed trajectory and travel time of the two tests are represented in Fig. 3.23.

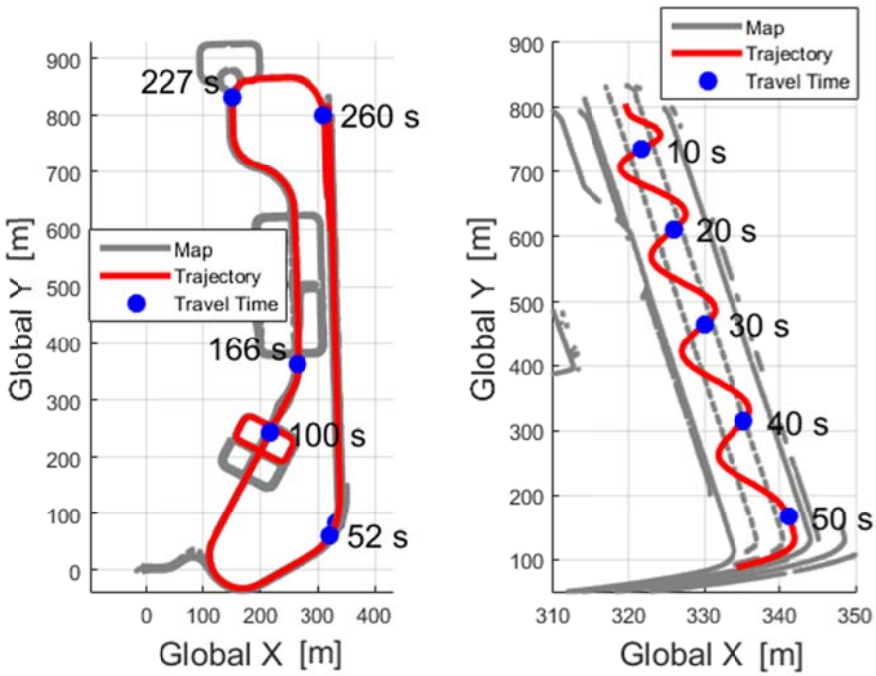
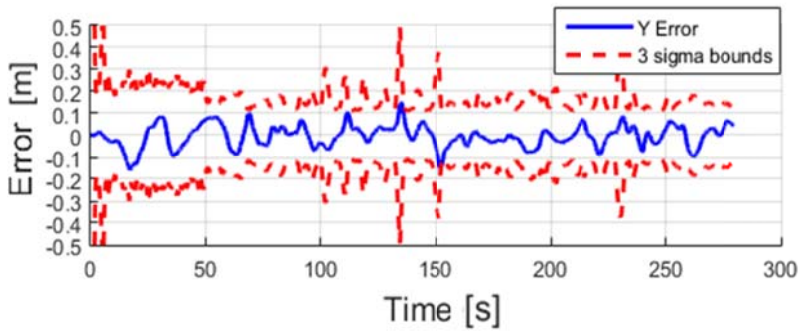
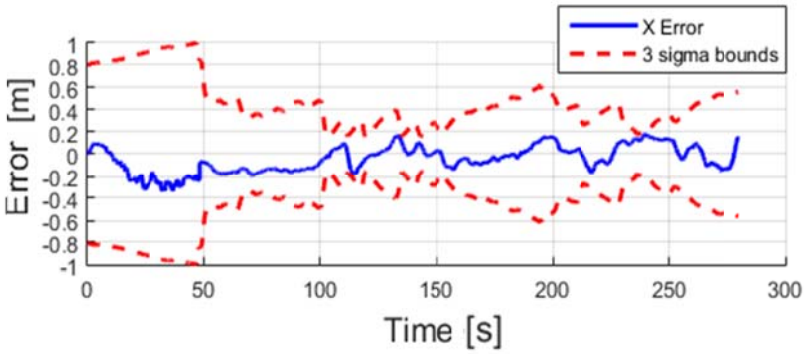


Figure 3.23 Detailed trajectory and travel time of two typical scenarios.

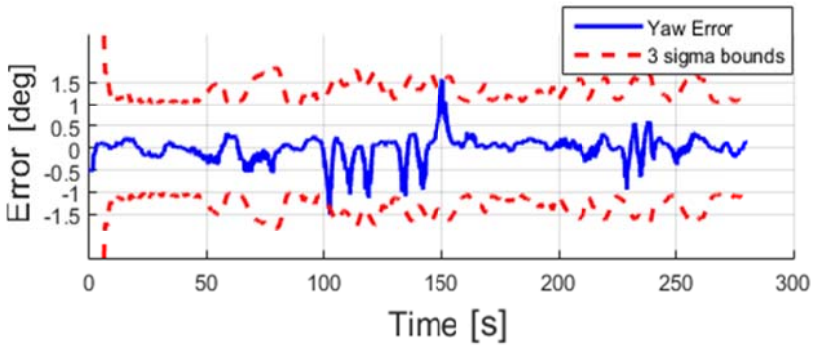
Fig. 3.24 and 3.25 (a), (b), (c) show changes of localization errors with respect to the vehicle coordinates. To show how the filter is well tuned, ±3σ bounds are plotted as red dashed lines. In both cases, the lateral error is smaller than the longitudinal error because the most frequently used shape for map-matching is a corridor shape. Fig. 3.24 and 3.25 (d) represent the history of shapes used to matching the map. As mentioned in Section 3.6.2, the covariance of map-matching in the longitudinal direction is much larger than that in the lateral direction in the case of corridor shape matching. The confidence of the lateral position is much higher than that of the longitudinal position in the overall tests.



(a) Lateral position error

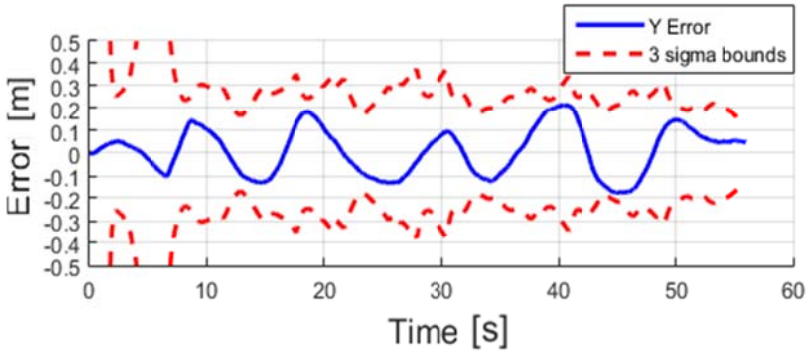


(b) Longitudinal position error

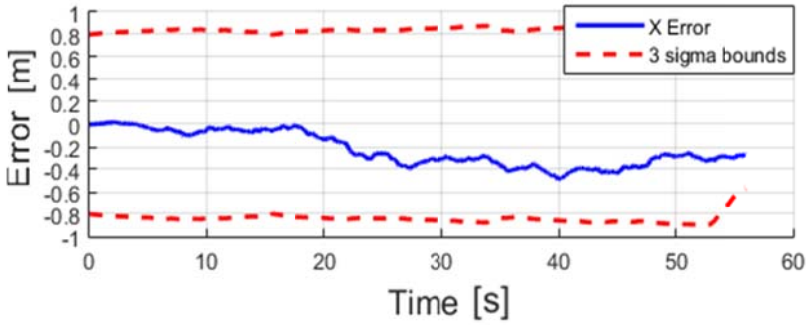


(c) Yaw angle error

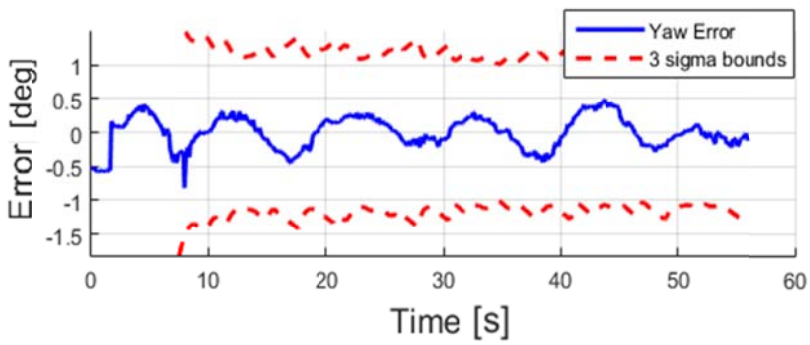
Figure 3.24 Localization result of 1st scenario



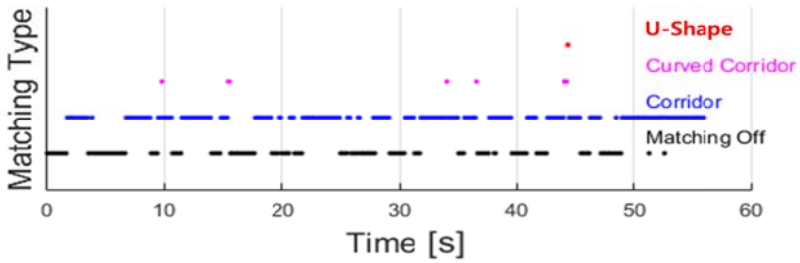
(a) Lateral position error



(b) Longitudinal position error



(c) Yaw angle error



(d) Map-matching history

Figure 3.25 Localization result of 2nd scenario

Actually, the longitudinal error is not important to control an automated vehicle for keeping the lane on small-curvature roads. However, for roads with large curvature (especially, in intersections), the longitudinal error should be decreased to the same level as that of the lateral error. In most cases, there are stop lines or merge lines that enable a U-shape matching before entering a road with large curvature. The longitudinal error and variance significantly decrease whenever a U-shape matching occurs. Thus, it can be said that the proposed algorithm has a sufficient localization performance to control an automated vehicle on the proving ground. Fig. 3.26 shows the situations where U-shape matchings occurred during the 1st Scenario.

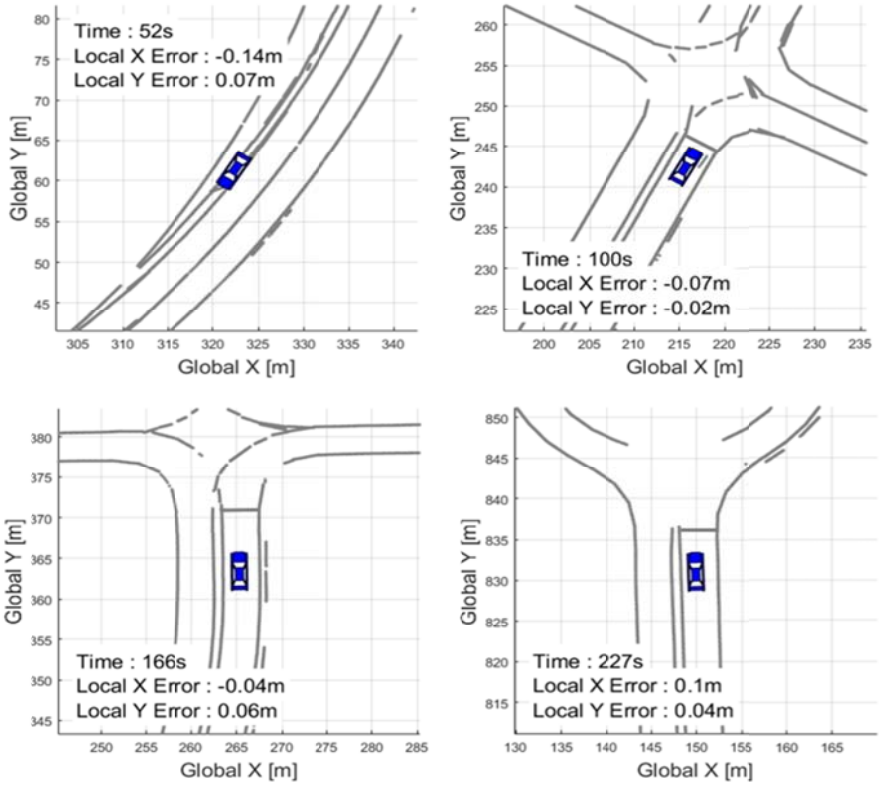


Figure 3.26 Location where U-shape matchings occurred

The lateral error of the 2nd scenario is slightly bigger than that of the 1st. This is caused by the lack of visual measurement updates. Since an AVM camera has a small field of view, lane data used for map-matching are not sufficient when the test vehicle changes lane. This is depicted in Fig. 3.27. However, in spite of consecutive lane changing, the mean errors of lateral direction is also small(less than 20 cm) enough to control the automated vehicle.

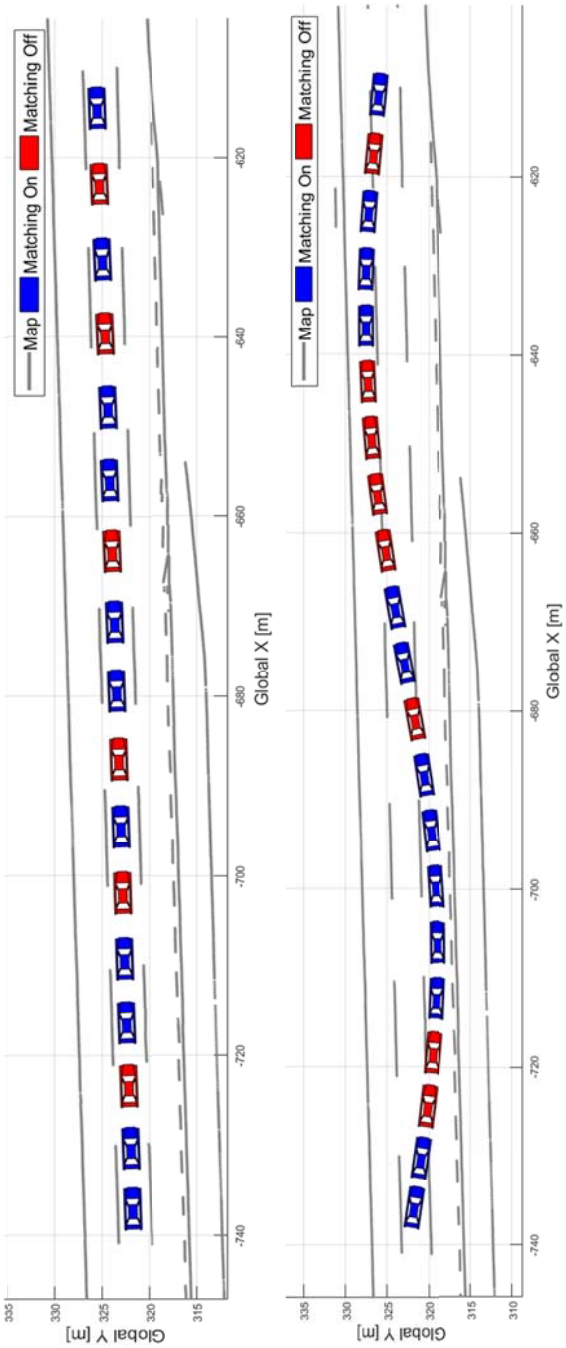
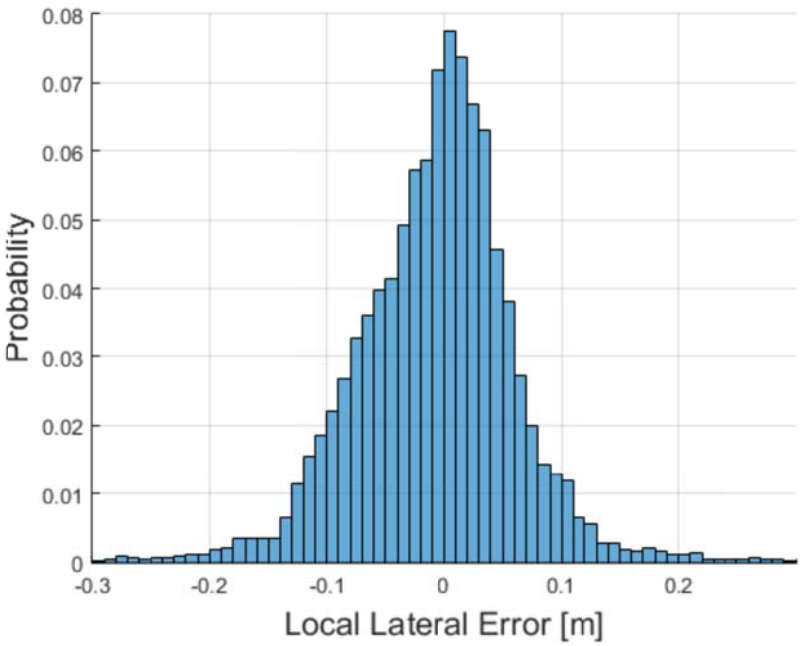
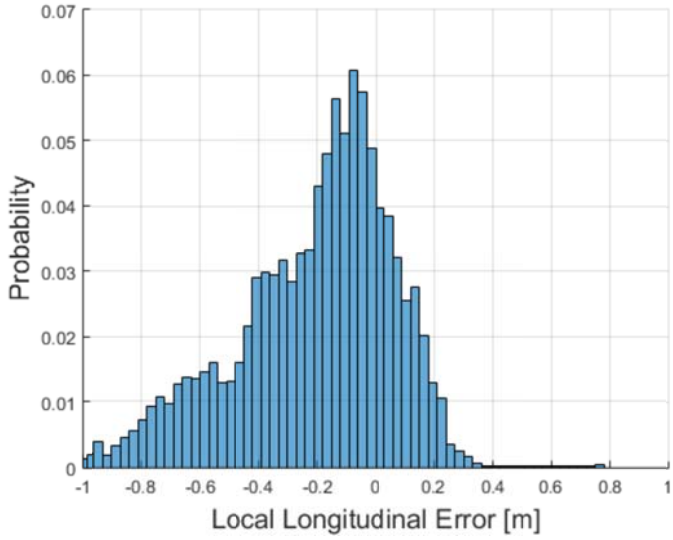


Figure 3.27 Operation of map-matching according to the driving maneuvers

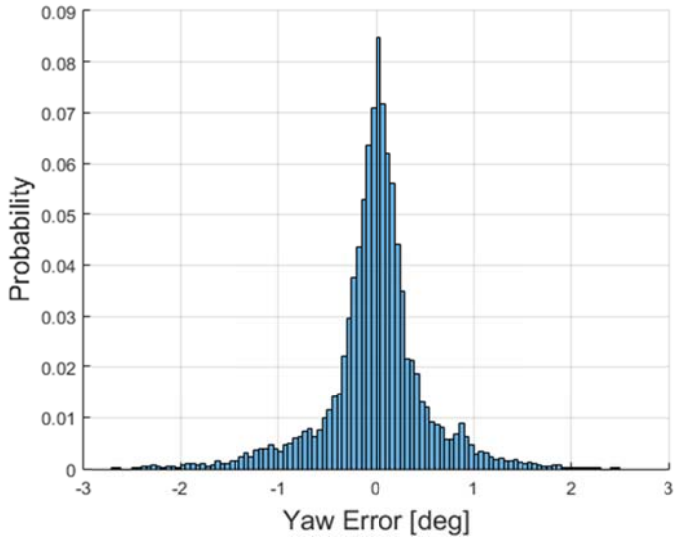
Figure 3.28 shows the histogram of the measurement residuals over all data sets. The medians of the lateral position and yaw error are respectively less than 20 cm and 1 deg. The mean error in the longitudinal direction are much larger than that in the lateral. However, as mentioned earlier, longitudinal position error decreases to the same level as that of the lateral error before entering a road with large curvature. Therefore, it does not matter to the control of the automated vehicle. From these results, it was confirmed that our localization method has applicability to automated driving in an urban environment.



(a) Lateral error distribution



(b) Longitudinal error distribution



(c) Yaw error distribution

Figure 3.28 Histogram of local longitudinal / lateral position and yaw angle error of total data

We compared our algorithm with another localization method based on front camera [Tao 13]. Table 3.3 gives performance metrics of both localization methods. Since the two results weren't obtained under the same conditions, the results cannot be compared directly. Even then, it is obvious that lateral accuracy is significantly improved by the proposed algorithm. The major factor that brings about these results is the AVM's characteristics mentioned in section 3.1. AVM images enable direct calculation of the lateral offset and make it possible to improve the lateral accuracy through map-matching.

TABLE 3.3 Error statics. (PE: Positioning Error; I: Front camera-based Localization; II: AVM-based Localization)

	Lateral PE (m)		Longitudinal PE (m)	
	I	II	I	II
mean	0.26	0.072	0.39	0.26
std. dev	0.34	0.067	0.39	0.23

3.8.2. Urban Roads Tests

Automated driving test based on real-time implementation of proposed algorithm is conducted on actual urban road to evaluate localization and driving performance. In this test case, test vehicle drove automatically itself along designated route under controlled traffic of government as depicted in Fig.3.29. In keeping with general urban driving conditions, the vehicle speed was limited to 60 km/h. The test mileage and driving time were about 3 kilometers and 5 minutes.

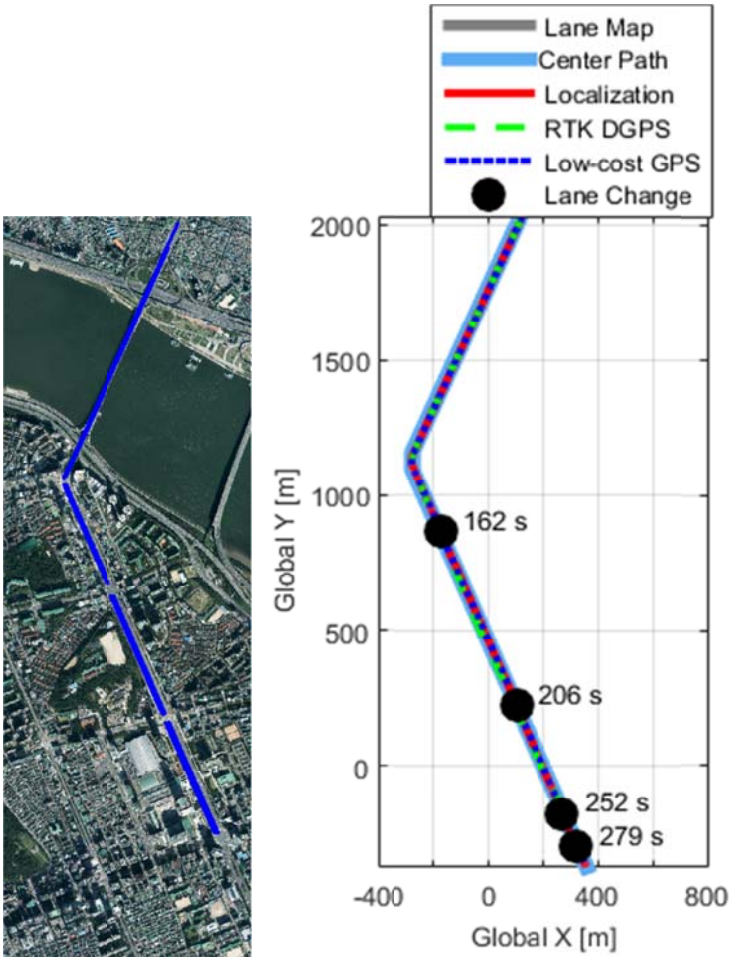
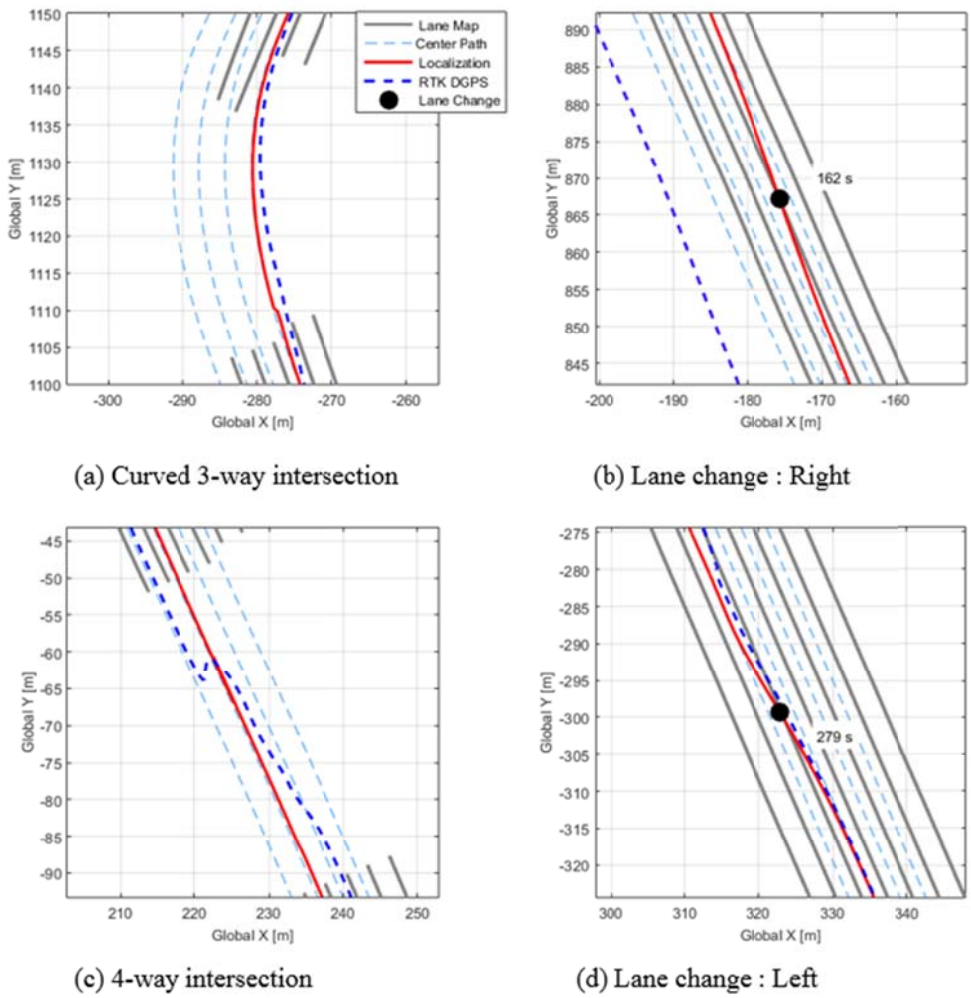


Figure 3.29 Detailed trajectory of urban road test.

Four subplots of Fig.3.30 describe some typical traffic situation in urban road. Fig.3.30-(a) shows curved 3-way intersection, Fig.3.30-(c) indicates typical 4-way intersection. The accurate previous localization and precise dead reckoning is required due to no lanes in intersection region. Fig.3.30-(b) and (d) depict right and left and right lane change situation on straight road.



(a) Curved 3-way intersection (b) Lane change : Right

(c) 4-way intersection (d) Lane change : Left

Figure 3.30 Description of issued region

Lateral position error of localization result, RTK-DGPS and commercial GPS in time domain are shown in Fig.3.31. The lateral position error obtained from GPS devices are large due to poor GPS condition such as multipath error, refraction, and so on. The lateral error of localization is maintained under accurate level, except lane change situation indicated in black circle region in Fig.3.31.

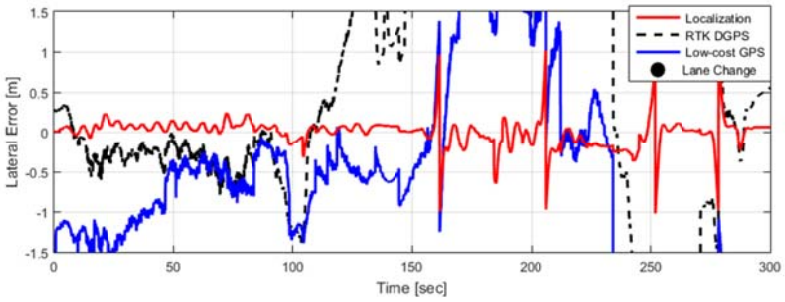


Figure 3.31 Lateral position error of actual urban automated driving

3.9. Performance Evaluation

We compared proposed vehicle localization algorithm with previous approaches based on a front camera to identify more clearly our contribution. For this purpose, we evaluated the lane detection performance of two vision sensor(AVM: Around View Monitoring Camera , FCAM: Front Camera) based method. In addition, we compared matching covariances of these sensors. As mentioned earlier, matching covariances are closely linked with shapes of sensor data. These shapes are greatly affected by each sensor's FOV. Then, we did a comparative analysis of localization performances of two approaches.

3.9.1. Lane Detection Performance Evaluation

A commercialized front camera for lane detection generally has a resolution of 640x480 pixels, as shown in Fig. 3.32.

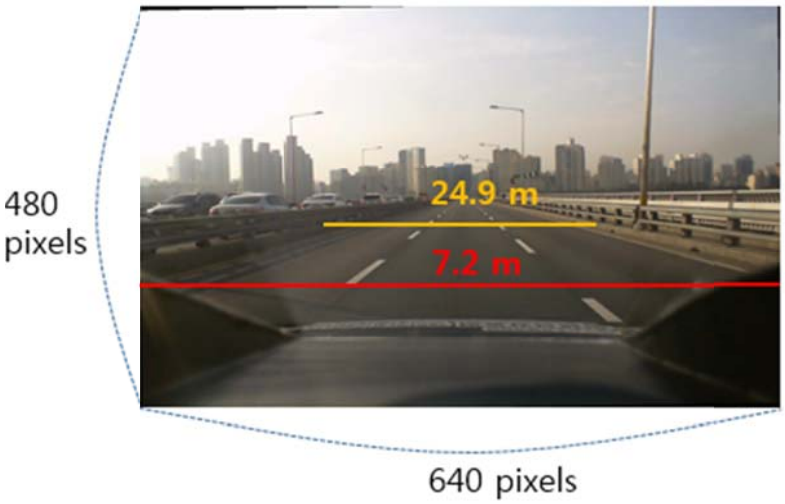


Figure 3.32 Resolutions of a front camera

To calculate a resolution (m/pixel) of FCAM, an inverse perspective mapping (IPM) image is required, as shown in the left of Fig 3.33. IMP is a geometrical transformation technique that projects each pixel of the 2D perspective view of a 3D object, and re-maps it to a new position and constructs a new image on an inverse 2D planar. Mathematically, IPM can be described as a projection from a 3D Euclidean space on a 2D planar. Using this, we described a resolution of each sensors in Table 3.4

TABLE 3.4 resolutions of two vision sensors

Sensors	**FCAM (7.2m)**	**FCAM (24.9m)**	**AVM**
Resolution	0.015(m/pixel)	0.033 (m/pixel)	0.021(m/pixel)

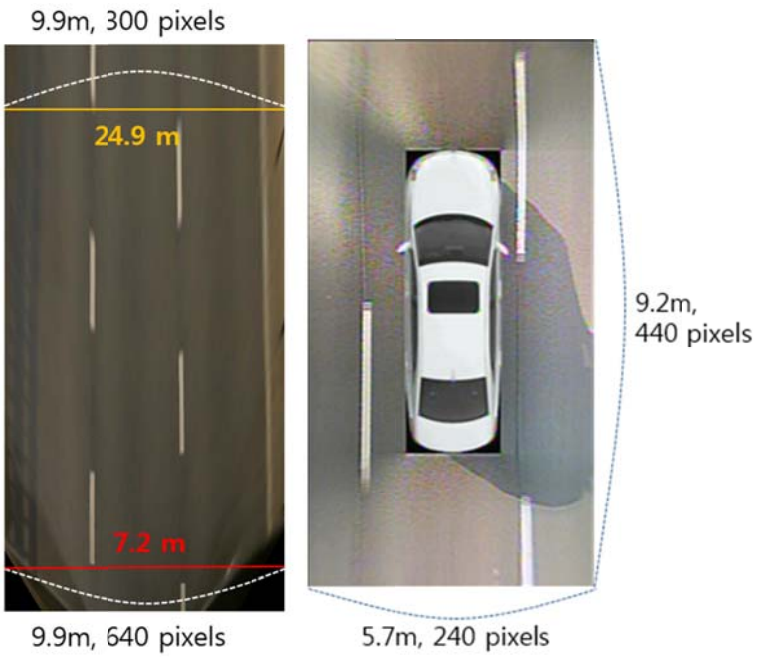


Figure 3.33 Comparing Resolutions of FCAM (Left) and AVM (Right)

As shown in table 3.4, two vision sensors have similar resolution. A lane detection performance cannot be evaluated only by a resolution of sensor. A lane detection algorithm must be considered. However, it is difficult to directly analyze a performance of these algorithms. So we conduct a simple experimental analysis based on real sensor data and a lane map. First, we transform detected lane data into a global coordinate using high accuracy GNSS as show in Fig. 3.34. Then, we can calculate the error of lane detection using a pre-built lane map.

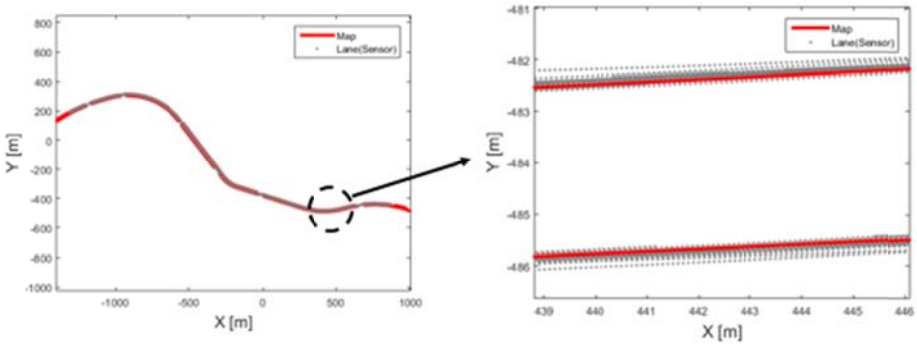


Figure 3.34 Detected lane in a global coordinate and lane maps(Ground truth)

Fig. 3.35 shows the performance of two sensors. In spite of similar resolutions, AVM camera was approximately twice accurate.

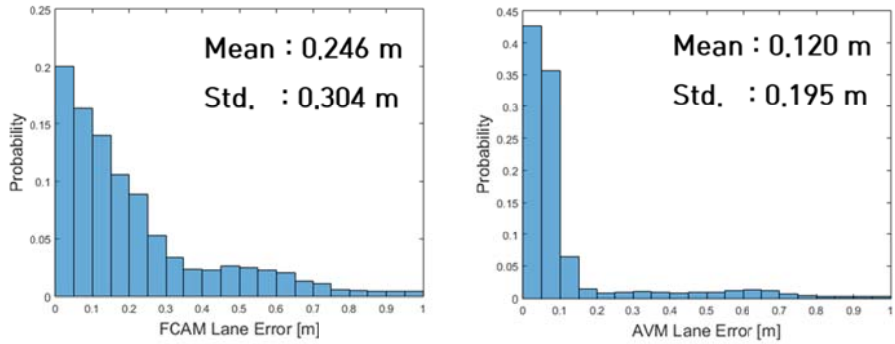


Figure 3.35 Lane detection performance of FCAM (Left) and AVM (Right)

The main reason for this is that a performance of lane detection based on FCAM is easily influenced by disturbances (road gradient, pitch motions of ego vehicle). Fig. 3.36 describes about these disturbances.

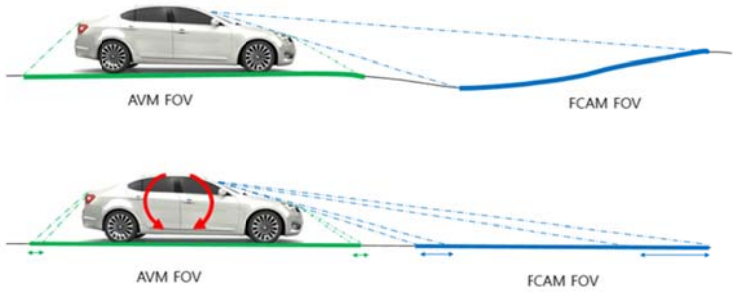


Figure 3.36 Disturbances of roads: road gradient (Top) and pitch motions of ego-vehicle (Bottom).

3.9.2. Comparing the Matching Covariance

By calculating the matching covariance of sensor's lane data, we can predict the effect of each sensor on a localization performance. For the accurate analysis of sensor effects, various curvature road must be considered. Therefore, we use virtual lane data considering each sensor's FOVs and curvatures of roads. Fig. 3.37 shows these virtual lane data.

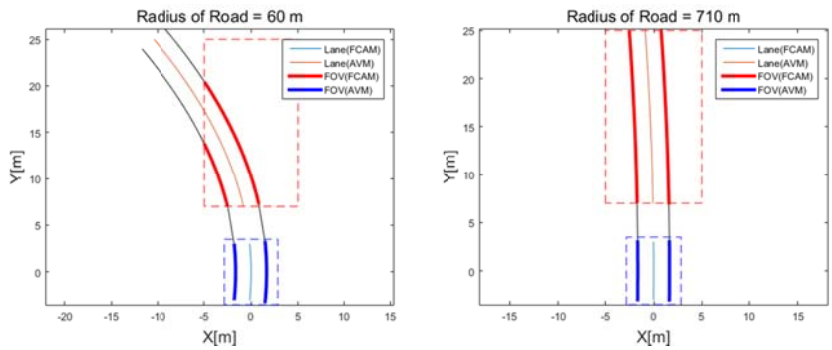
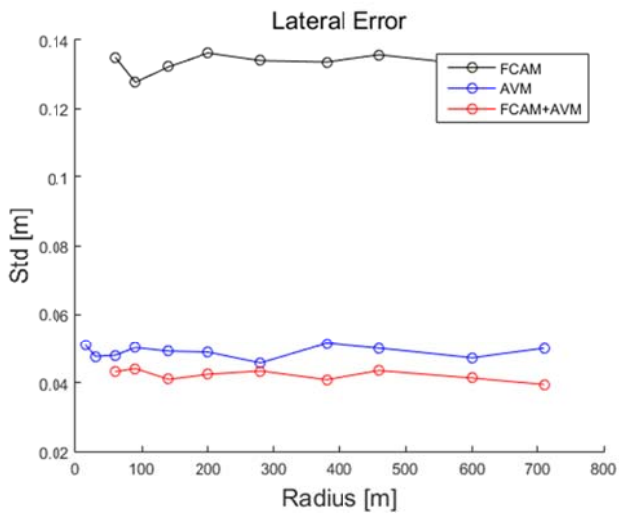
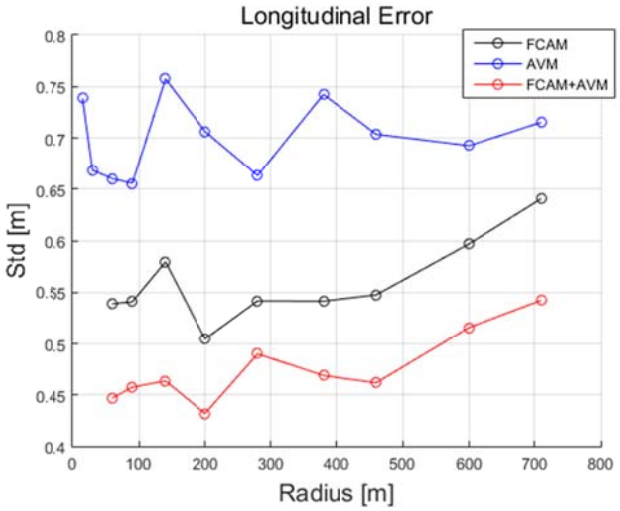


Figure 3.37 Shapes of lane data determined by a curvature of road and sensors' FOV

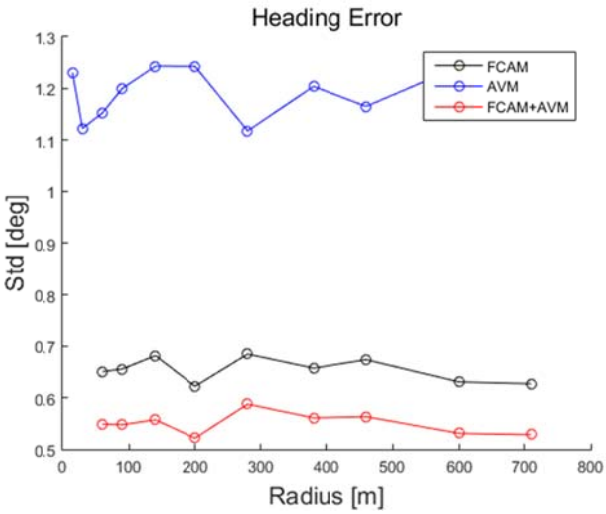
Fig. 3.38 show the standard deviations of each sensor's matching errors of with respect to radiuses of roads. Of course, the performance in case of using both sensors is the best. In addition, we can confirm that matching errors of AVM are independent to radius of roads. This is because AVM's FOV is too narrow to detect the changes of curvature.



(a) Lateral errors with respect to radiuses of roads.



(b) Longitudinal errors with respect to radiuses of roads.



(c) Standard deviations of heading errors with respect to radiuses of roads.

Figure 3.38 Standard deviations of matching errors with respect to radiuses of roads.

3.9.3. Localization Performance Evaluation

In earlier sections, we considered each sensor's effects on localization results, theoretically. In this section, we evaluate performances via vehicle tests in urban roads. The trajectory of vehicle tests depicted in Fig. 3.39. In addition, vehicle states during the test are presented in Fig.3.40. In keeping with general urban driving conditions, the vehicle speed was limited to 60 km/h. The test mileage and driving time were about 3 kilometers and 3 minutes.

Figure 3.39 Trajectory of urban road test for each sensor's performance evaluation.

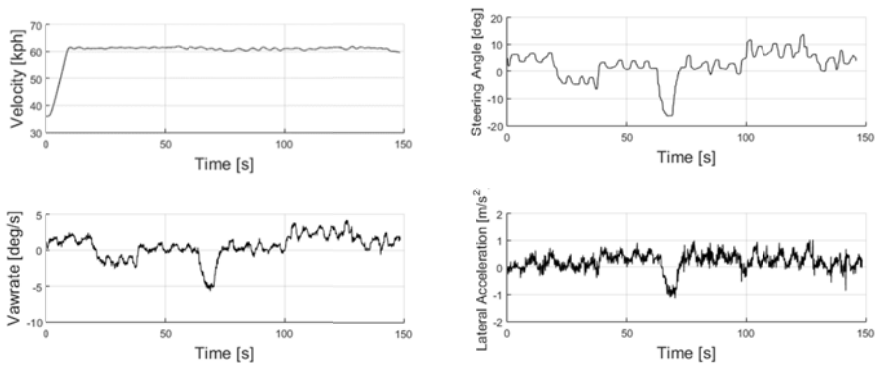
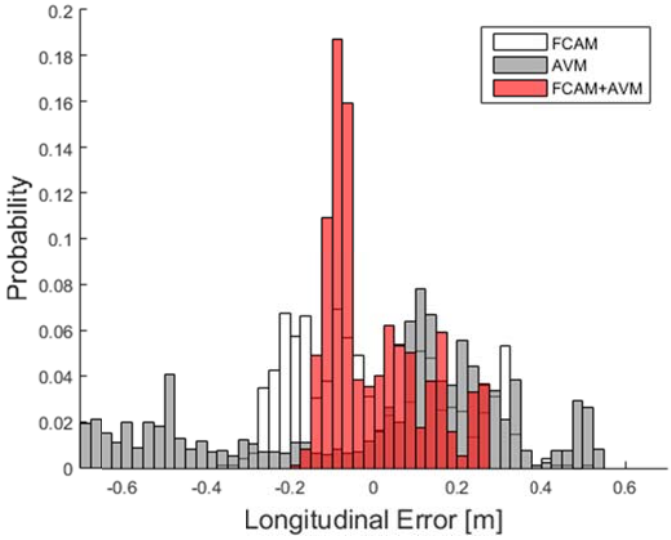
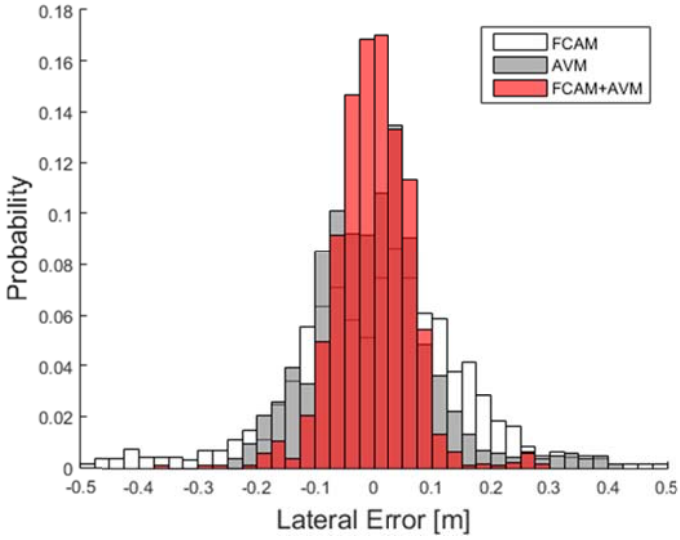


Figure 3.40 States history of a test vehicle.



(a) Longitudinal errors distribution.



(b) Lateral errors distribution.

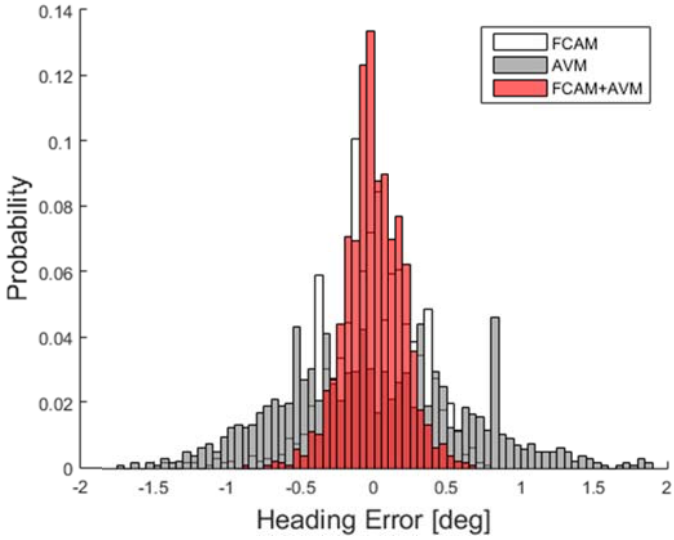


Figure 3.41 Histogram of local longitudinal / lateral position and yaw angle error of total data of each sensor system.

Fig.3.40 show a similar tendency of matching covariance. The standard deviation of the lateral position and yaw error of proposed algorithm based on AVM are respectively less than 10 cm and 0.62 deg. The error in the longitudinal direction are much larger than that in the lateral. However, as mentioned in Sec.3.8.1, longitudinal position error decreases to the same level as that of the lateral error before entering a road with large curvature. Therefore, it does not matter to the control of the automated vehicle. From these results, it was confirmed that our localization method has applicability to automated driving in an urban environment. A multi-sensor (FCAM, AVM) fusion approach are approximately 2 times more accurate than general approach. The total test results are arranged in Table. 3.5

TABLE 3.5 Localization performance evaluation.

Sensors	FCAM	AVM	FCAM+ AVM
$\sigma_x(m)$	0.18	0.34 (▼88 %)	0.11 (▲38 %)
$\sigma_y(m)$	0.17	0.10 (▲42 %)	0.06 (▲65 %)
σ_{yaw}(deg)	0.30	0.62 (▼106 %)	0.19 (▲36 %)

Chapter 4

MPC based Automated Vehicle Control in Urban Roads

The increase in the number of vehicles on the road results in a progressive saturation of the road network. So many drivers in the big city are suffering from traffic congestion. In addition to the immediate impacts on drivers (wasting time, stress, aggressiveness), congestion of the road network also has significant economic and environmental costs. In order to solve this problem, leading automotive manufacturer and part suppliers are developing systems that help driver in the monotonous situation. This kind of system typically uses radar and camera technology to keep pace with other vehicles and provide automatic steering control to stay in the current lane. Under the urban driving condition, it is often difficult to recognize the lane due to damaged paint or external environmental factors. However, in such a case, human drivers can drive their vehicles without any problem. Because they can infer the direction of travel based on the preceding vehicle's motion. In this study, we developed an automated urban driving system based on this characteristic of the driver.

4.1. Related Research

In order to perform these functions, this system usually consists of two subsystems: longitudinal control and lateral control. The longitudinal control has been studied for almost half a century and many challenges associated with it have been resolved [Swaroop 96, Moon09]. However, there are many problems yet to be solved for lateral control of vehicle following system.

Approaches used to solve the perception problem could be classified into three categories [Teck 04]: GPS [CAUGV 03, Farwell 99], active sensors [Lee 02, Chandak 02, Han 01] and passive sensor[Cowan 03, Das 01, Artus 03]. In GPS based approach, the positions of the leader vehicle are echoed to the follower vehicle to generate and adjust of the follower's path. The main advantage is that no perception sensors are needed and the system can be operated regardless of lighting condition. However, its reliability depends on the availability of GPS data. For example in built-up areas or where dense tree foliage blocks satellite signals, its confidence is very low since availability of GPS cannot be guaranteed. Active sensor(Lidar) based approach is the most stable in various road condition, however, the tracked target may escape from the single line scan sensor's field of view(FOV) when the vehicle travels on an irregular terrain. In passive sensor(Vision) based approach, the sensor can provide more information about the environment, nevertheless, the heading angle of the leader vehicle can cause errors on the tracking algorithms. And environmental conditions such as lighting pose are severe constraints in this approach.

Approaches used to control the lateral motion of vehicle could be classified into two categories: direct control and trajectory based control. Direct control approaches [Lu 04, Plamen 08] are very similar to path tracking. They formulate inter-vehicle kinematics in error coordinate and design the steering controller using adaptive control theory. In trajectory based approaches

[Gehrig 98, Takehiko 98], that use the time history of the lead vehicle, the ego-vehicle has to follow the path of the leader. Robustness and real-time performance of these methods have been validated. However, the disadvantage of methods is that the system cannot respond to obstacles.

In this dissertation, we proposed the new method to guarantee the safety of automated vehicle following system on inner-city streets. In order to achieve this, MPC has been applied to control the lateral motion of vehicle. An optimal vehicle following problem while avoiding collision with obstacles is formulated in terms of cost minimization under constraints. Information on obstacles is incorporated online in the nonlinear model-predictive framework as they are sensed within a limited sensing range. The overall problem is solved online with nonlinear programming. And, in order to build such system, we use single line lasers sensor that is one of the most robust sensor in various road condition. To solve the problem of FOV mentioned above, extended Kalman filter has been applied to our system.

4.2. Control Architecture

The proposed vehicle following system consists of a vehicle tracking part and a vehicle following part, as shown figure 4.1.

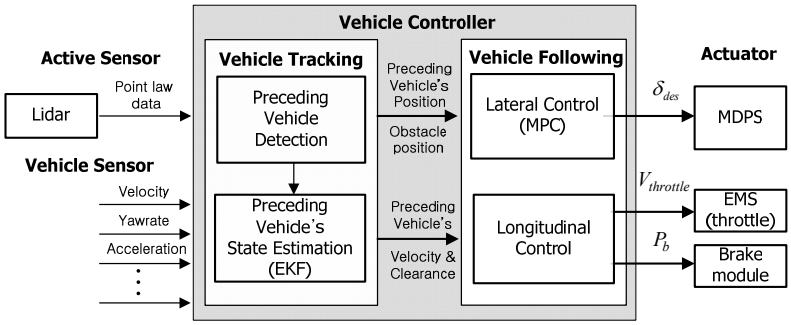


Figure 4.1 Control structure

Tracking algorithm is designed to detect the preceding vehicle from point clouds of two lidars(shown in figure 4.2) based on the geometrical configuration and estimate its states using extended Kalman filter. In a vehicle following part, lateral controller determines the optimal steering input using model predictive control method and longitudinal controller determines the throttle/brake inputs for keeping pace with preceding vehicle.

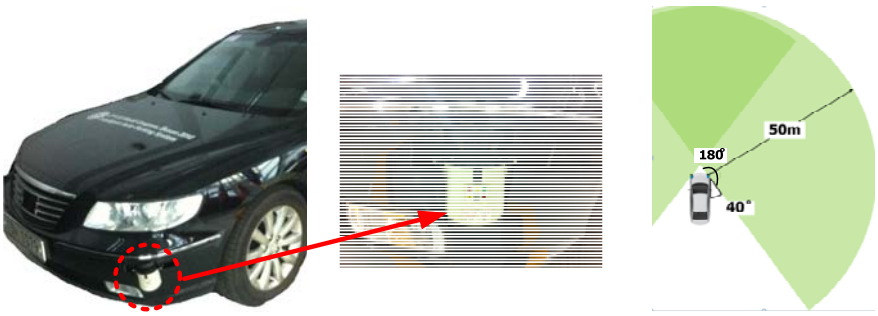


Figure 4.2 Lidar sensor configuration

4.3. Target Tracking

A vehicle tracking algorithm is consisted two parts: a target detection and a target estimation. The detection algorithm extracts the information about relative position and heading angle of preceding vehicle from point clouds. And the state estimation algorithm estimate velocity, yawrate as well as the information obtained by prior process. Furthermore, the estimator provide area where preceding vehicle is expected to exist at the next time step for detection algorithm.

4.3.1. Target Detection

To detect a preceding vehicle, point clouds acquired from two lidars mounted on each side of the front bumper were segmented based on distance between the two points adjacent. And then, vehicle detection algorithm classifies the segment that corresponds to the preceding vehicle using the information about expected path of ego-vehicle or expected area where preceding vehicle exists provided by tracking algorithm as shown figure 4.3.

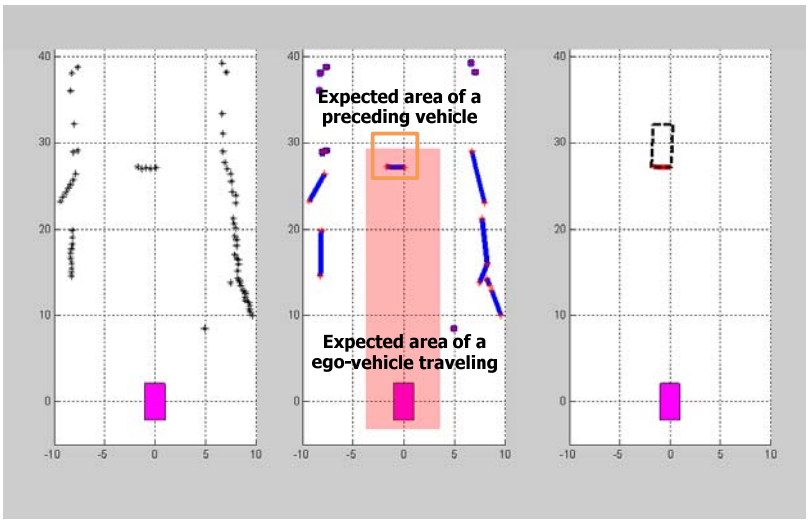


Figure 4.3 Processing flow at each scan

Next we extract information about relative position and heading angle from the segment obtained by prior process using Ramer algorithm [Nashashibi 08]. The main objective of this algorithm is to model a set of points by one or two segments as shown Fig.4.4

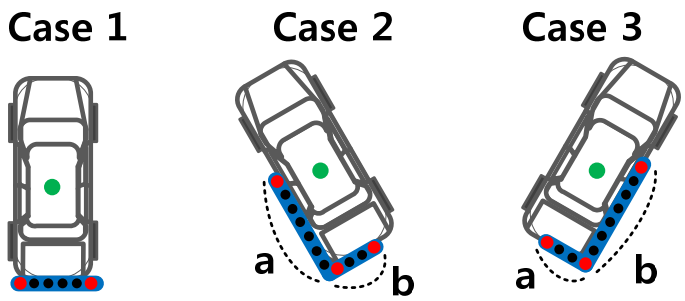


Figure 4.4 Position estimation from point clouds

4.3.2. Target Estimation

The extended Kalman filter is used to estimate preceding vehicle states such as longitudinal velocity, yaw rate, longitudinal acceleration and yaw acceleration from preceding vehicle's relative position obtained through prior process and the vehicle sensor signals such as longitudinal velocity yaw rate under the assumption of the Gaussian white noise [Kim 11]. The state vector is defined as following in order to represent preceding vehicle's planar behavior:

$$x = \begin{bmatrix} p_{x,rel} & p_{y,rel} & \theta_{rel} & v_{pre} & \gamma_{pre} & a_{pre} & \dot{\gamma}_{pre} \end{bmatrix}^T \tag{4.1}$$

where is the preceding vehicle's position relative to ego-vehicle, is its longitudinal velocity, yaw rate, longitudinal acceleration and yaw acceleration. The measurement vector is defined as following.

$$z = \begin{bmatrix} p_{x,rel} & p_{y,rel} & \psi_{rel} \end{bmatrix}^T \tag{4.2}$$

Assuming that the time derivatives of preceding vehicle's longitudinal acceleration and the yaw acceleration can be considered as the process noise, the process model and measurement model are given by following form:

$$\begin{aligned} x_{k+1} &= f\left(x_k, u_k\right) + w_k \\ z_k &= h\left(x_k\right) + v_k \end{aligned} \tag{4.3}$$

where

$$f(x) = x + \begin{bmatrix} v\cos\theta_{rel} - v + p_{y,rel}\cdot\gamma \\ v\sin\theta_{rel} - p_{x,rel}\cdot\gamma \\ \gamma_{pre} - \gamma \\ a_{pre} \\ \dot{\gamma}_{pre} \\ 0 \\ 0 \end{bmatrix} \cdot \Delta t, \quad h(x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where is the sampling time which taken as 0.1 second in this study, is ego-vehicle's longitudinal velocity and yawrate. The process noise is assumed to be a white noise with associated covariance matrix, W. The measurement noise is also assumed to be a white noise with associated covariance, V. With above process and measurement model, preceding vehicle states are recursively estimated by using the extended Kalman filter which is a sequence of time and measurement update steps as following specific equations:

Time update

$$\bar{x}[k] = f(\hat{x}[k-1])$$
$$F[k-1] = \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}[k-1]}, \quad H[k-1] = \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}[k-1]}$$
$$M[k] = F[k-1] \cdot P[k-1] \cdot F[k-1]^T + W \tag{4.4}$$

Measurement update

$$\hat{x}[k] = \bar{x}[k] + K[k] \cdot (z[k] - H \cdot \bar{x}[k])$$
$$K[k] = M[k] H^T \cdot (HM[k] H^T + V[k])^{-1}$$
$$P[k] = (I - K[k] H) \cdot M[k] \tag{4.5}$$

Abnormal measurements are often occurring due to the pitch motion of ego vehicle or the slope of the road. In the case as shown in the right two graphs of figure 5, vehicle detection algorithm can't extract information about preceding vehicle. However, it is possible to predict the preceding vehicle's state only using time update process because these symptoms occur for a while. So we estimate the preceding vehicle's state only using time update process during the abnormal measurement are occurs.

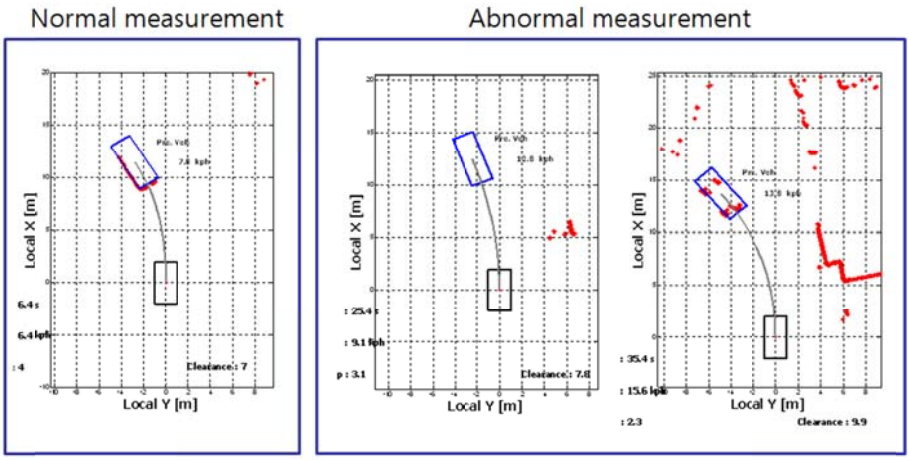


Figure 4.5 Vehicle tracking using extended kalman filter

4.4. Vehicle Control

4.4.1. Reference Trajectory Generation

A good initial guess and reference trajectory are essential for fast convergence of MPC. So before the optimizing process, we calculate feasible initial guess and reference trajectory for vehicle following based on the cubic spline. A cubic spline provides a very feasible reference trajectory in most driving conditions. Using this spline, we also calculate the initial guess of steering input sequence using curvature of the spline.

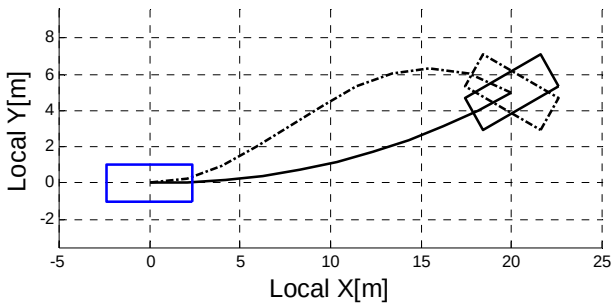


Figure 4.6 Reference path generation using cubic spline

As show the above figure, both ends of the spline is on the two points : ego-vehicle position $(x_1 \quad y_1 \quad \psi_1)$ and preceding vehicle position $(x_2 \quad y_2 \quad \psi_2)$. And each slope of the tangent line is agreements with each heading angle.

The coefficient of cubic spline is calculated using eq.4.6

$$A = X^{-1}Y = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix}^T \tag{4.6}$$

Where

$$X = \begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ 3x_1^2 & 2x_1 & 1 & 0 \\ 3x_2^2 & x_2 & 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \tan(\psi_1) \\ \tan(\psi_2) \end{bmatrix}$$

Assuming that there is no side slip, the steering angle is calculated by the following equation.

$$\delta_{des} = atan(L / r) \tag{4.7}$$

where

$$\mathrm{r} = \left| \frac{(1+\dot{y}^2)^{3/2}}{\ddot{y}} \right|, \qquad \begin{aligned} \dot{y} &= 3a_1 x_1^2 + 2a_2 x_1 + a_3 \\ \ddot{y} &= 6a_1 x_1 + 2a_2 \end{aligned}$$

and L is the wheelbase of ego vehicle.

4.4.2. Lateral Control

In order to use MPC for lateral control, it is necessary to construct a vehicle model for the path generation. In the vehicle model, the state is defined by including the longitudinal velocity, heading angle, time derivative of heading angle and longitudinal/lateral position of a vehicle, as given in equation 4.8. With this definition, the state equation is obtained as follows:

$$\xi = [\beta \quad \psi \quad \dot{\psi} \quad X \quad Y] \tag{4.8}$$

The state equation can be written as given in equation 4.9.

$$\frac{d}{dt}\begin{bmatrix}\beta\\ \psi\\ \dot{\psi}\\ X\\ Y\end{bmatrix}=\begin{bmatrix}\frac{-2\left(C_f+C_r\right)}{mv_x}\beta+\left(\frac{-2\left(l_fC_f-l_rC_r\right)}{mv_x^{\ 2}}-1\right)\dot{\psi}+\frac{2C_f}{mv_x}\delta_f\\ \dot{\psi}\\ \frac{-2\left(l_fC_f-l_rC_r\right)}{I_z}\beta+\frac{-2\left(l_f^{\ 2}C_f+l_r^{\ 2}C_r\right)}{I_zv_x}\dot{\psi}+\frac{2l_fC_f}{I_z}\delta_f\\ v_x\cos\psi-v_x\tan\beta\sin\psi\\ v_x\sin\psi+v_x\tan\beta\cos\psi\end{bmatrix} \tag{4.9}$$

The control input (u) is the steering angle (δ_f), and the output is set to be the position of the vehicle in the inertial reference frame. Therefore, the trajectories can be generated based on the output variables as follows:

$$\eta=\begin{bmatrix}0&0&0&1&0\\0&0&0&0&1\end{bmatrix}\cdot\xi=\begin{bmatrix}X&Y\end{bmatrix}^T \tag{4.10}$$

For the lateral stability of the vehicle, as a constraint, maximum yaw rate ($\dot{\psi}_{max}$) is defined by the maximum allowable lateral acceleration and the longitudinal velocity, as given in equation 4.11.[Kou 08, Gordon 03]

$$\left|\dot{\psi}\right|\le\dot{\psi}_{\max}=\frac{a_{y,limt}}{v_x} \tag{4.11}$$

For the purpose of discrete-time implementation, the state equation 4.9 is discretized using the Euler method as follows:

$$\xi_{k+1}=\xi_k+\Delta T\cdot f_k\left(\xi_k,u_k\right)=f^{dt}\left(\xi_k,u_k\right) \tag{4.12}$$

Given the current state ξ_0 at the current time step t, MPC computes the optimal control sequence $[u_k]$ with the receding horizon principle that solves the following optimization problem:

$$
\begin{aligned}
&\textit{Minimize} \quad J_{mpc}\left(\left[\xi_k\right]_{k=0}^{N}, \left[u_k\right] \right) \\
&\textit{subject to} \quad (i) \quad \xi_{k+1} - f^{dt}(\xi_k, u_k) = 0 \\
&\qquad\qquad (ii) \quad d_{\min,k} - d_{safety} \geq 0 \\
&\qquad\qquad (iii) \quad \left|\dot{\psi}_k\right| - \frac{a_{y\max}}{v_x} \leq 0 \\
&\qquad\qquad \textit{where } k = 0, \ldots, N
\end{aligned}
\tag{4.13}
$$

In equation 4.13, N is the length of the look-ahead horizon, k is the time step within the look-ahead horizon. Equation (i) is the equality condition related to the dynamics of the vehicle. Equation (ii) is the bound constraint on the control input, and equation (iii) is the constraints on the yaw rate. The cost function to be minimized is defined as shown in equation 4.14.

$$
\begin{aligned}
&J_{mpc} = \phi_{mpc}\left(\Delta\eta_N\right) + \sum_{k=0}^{N-1} L_{mpc}\left(\Delta\eta_k, u_k\right) \\
&\phi_{mpc} = \frac{1}{2}\Delta\eta_N{}^T P_{mpc}\Delta\eta_N \\
&L = \frac{1}{2}\Delta\eta_k{}^T Q_{mpc}\Delta\eta_k + \frac{1}{2}u_k{}^T R_{mpc}u_k
\end{aligned}
\tag{4.14}
$$

where, $\Delta\eta_k$ represents the deviation from the reference trajectory $\eta_{ref,k}$ at time step k, i.e. $\Delta\eta_k = \eta_{ref,k} - \eta_k$. Reference trajectory associated with lead vehicle is defined on local coordinate attached to the longitudinal vehicle axle, as shown in figure 4.7. P_{mpc}, Q_{mpc} and R_{mpc} are positive semi-definite constant weighting matrices.

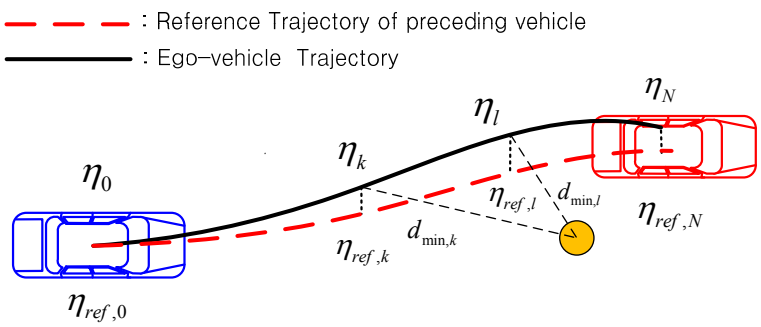


Figure 4.7 Definition of the predictive distance deviation ($\Delta\eta_k$)

In equation 4.14, the first term(ϕ_{mpc}) penalizes the deviation from the reference along the horizon, and the second term(L) penalizes the control input in terms of energy consumption. Therefore, equation 4.14 can be seen as a cost function for following a lead vehicle while minimizing the input-effort based on the constant weighting matrices.

To solve this optimization problem, the equality and the inequality constraints are incorporated into the cost function as the follows:

$$J_a = \phi(\Delta\eta_N) + \sum_{k=0}^{N-1}\begin{bmatrix} L(\Delta\eta_k, u_k) + \lambda_k^T\{\xi_{k+1} - f^{dt}(\xi_k, u_k)\} \\ +\mu_{obs}G_{obs}(d_{\min}) + \mu_{ay}G_{ay}(\dot{\psi}) \end{bmatrix} \tag{4.15}$$

where, λ_k^T is a sequence of Lagrange multiplier vectors. $G_u(d_{min})$ and $G_c(\dot{\psi})$, given in equation 4.16 and 4.17, are penalty functions for the inequality constraints on the control input and yaw rate, respectively. μ_{obs} and μ_{ay} are the weighting parameters for the penalty functions.

$$G_{obs}(d_{\min}) = \begin{cases} \frac{1}{2}\left|\frac{1}{d_{\min}} - \frac{1}{d_{safety}}\right|^2 & \text{if } \frac{1}{d_{\min}} - \frac{1}{d_{safety}} > 0 \\ 0 & \text{else} \end{cases} \tag{4.16}$$

$$G_{ay}(\dot{\psi}) = \begin{cases} \frac{1}{2}\left|\left|\dot{\psi}\right| - \frac{a_{y,sat}}{v_x}\right|^2 & \text{if} \quad \left|\dot{\psi}\right| - \frac{a_{y,sat}}{v_x} > 0 \\ 0 & \text{else} \end{cases} \tag{4.17}$$

The optimal input sequence $[\xi_k]_{k=0}^{N}$ can be calculated by an online optimization. A first-order gradient method is applied to Hamiltonian equation, as follows :

$$H_k = L\left(\tilde{\eta}_k, u_k\right) + \lambda_k^T f^{dt}\left(\xi_k, u_k\right) + \mu_{obs} G_{obs}(d_{\min}) + \mu_{ay} G_{ay}\left(\dot{\psi}\right) \tag{4.18}$$

The total variation in the cost function can be written as follows:

$$\begin{aligned} dJ_a = & \left(\frac{\partial J}{\partial x_N} - \lambda_N^T\right) dx_N + \frac{\partial H_0}{\partial x_0} dx_0 + \frac{\partial H_0}{\partial u_0} du_0 \\ & + \sum_{k=1}^{N-1} \left\{ \left(\frac{\partial H_k}{\partial x_k} - \lambda_k^T\right) dx_k + \frac{\partial H_k}{\partial u_k} du_k \right\} \end{aligned} \tag{4.19}$$

The Lagrange multiplier vectors are defined as follows:

$$\lambda_N^T = \frac{\partial \phi}{\partial \xi_N}, \ \lambda_k^T = \frac{\partial H_k}{\partial \xi_k} \tag{4.20}$$

The total variation in the cost function and Hamiltonian equation can be rewritten as follows:

$$dJ_a = \sum_{k=0}^{N-1} \frac{\partial H_k}{\partial u_k} du_k + \lambda_0^T d\xi_0 \tag{4.21}$$

$$\frac{\partial H_k}{\partial u_k} = u_k^T R + \lambda_{k+1}^T \frac{\partial f_k}{\partial u_k} \tag{4.22}$$

The optimal control steering inputs are calculated by an online optimization. Using Hamiltonian equation and Lagrange multiplier, the optimization

process is executed as follows :

$$u_k = \begin{cases} u_2, \textit{previous optimal input} & k = 1 \\ u_{k-1} + du_k & k = 2, \ldots, N \end{cases} \tag{4.23}$$

Gradient descent method is used to solve the MPC online optimization problem [Kim 02]. This method is suitable for real-time implementation. The optimization sampling rate is 0.01 seconds and prediction steps are varied according to the distance of the preceding vehicle. A cycle of optimizing process is set to be 0.1 seconds for real-time control.

4.4.3. Longitudinal Control

This paper focuses on lateral control. This section would only briefly describe the longitudinal controller algorithm for the simulation and test. As shown in Figure 4.8, cruise controller was composed of two parts: upper level controller and lower level controller. The upper level controller determines the desired acceleration based on the clearance and relative velocity considering ride quality and safety [Moon 09]. And the lower level controller regulates throttle and brake input to follow the desired acceleration using model free control [Kim 12].

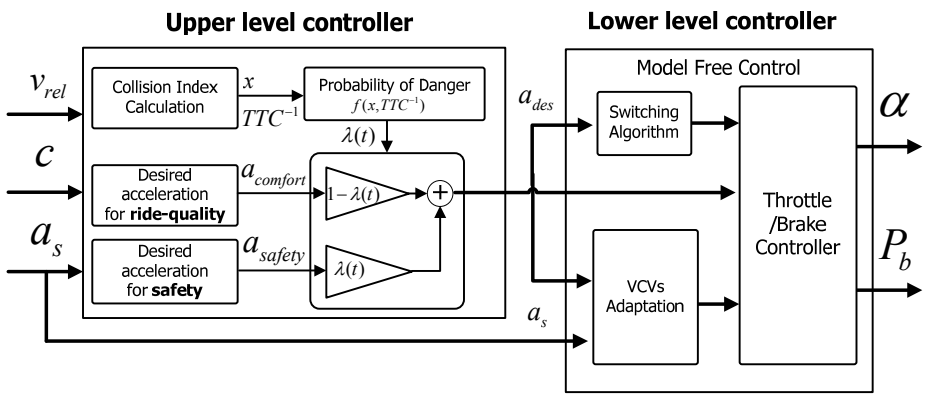


Figure 4.8 Scheme of an integrated longitudinal driving/safety control

4.5. Test Results

4.5.1. Simulation Results

To validate the performance of proposed algorithm, we carry out simulation test using vehicle dynamic software CARSIM and Matlab/Simulink. In the test, a lead vehicle travels along the straight road at a constant speed of 30km/h. The ego-vehicle follows the lead vehicle automatically. In order to verify the performance of obstacle avoidance capability, two missions were

added to simulation scenario as shown in figure 4.9. The first mission is passing next to cones that cover the right half of lane. And second mission is avoiding the oncoming vehicle on the opposite lane when the first mission was almost complete. In this test, the maximum allowable lateral acceleration and marginal distance from obstacles are 2m/s^2 and 0.5m.

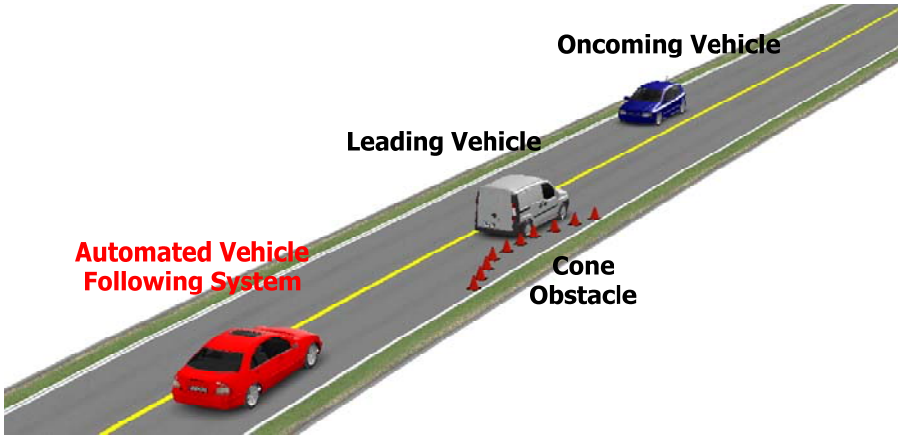
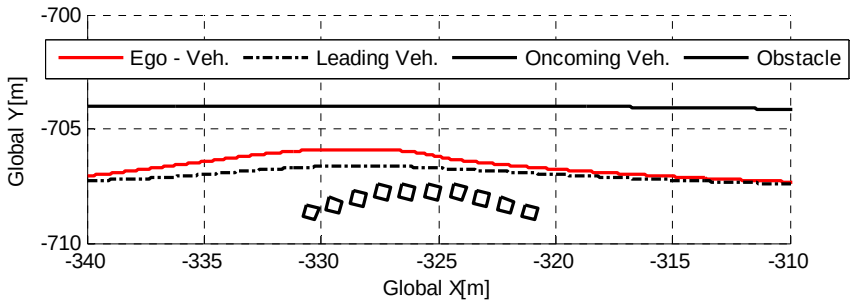
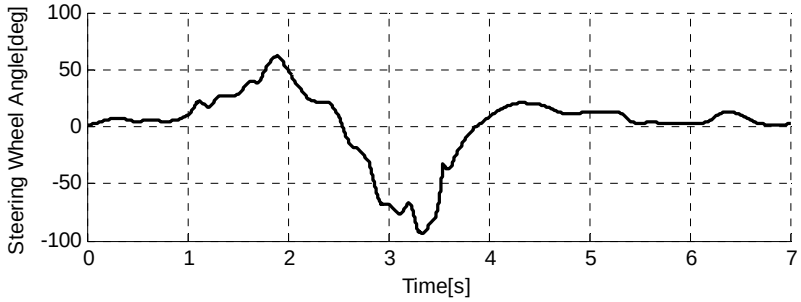


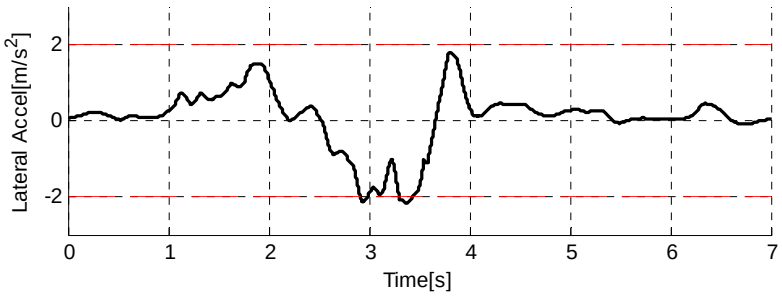
Figure 4.9 Simulation Scenario



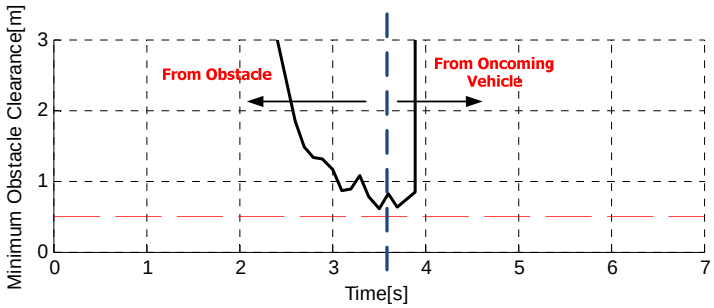
(a) Trajectories of each vehicles



(b) Steering Wheel Angle[deg]



(c) Lateral Acceleration[m/s^2]



(d) Minimum Distance from Obstacle [m]

Fiugre 4.10 Simulation Results

Simulation results are presented in figure 4.10. Figure 4.10 (a) shows the steering wheel angle calculated by model predictive steering controller for following the lead vehicle and avoiding obstacle. Figure 4.10 (b) and (c) show lateral acceleration and minimum distance from obstacles. In this test, minimum distance and lateral acceleration do not exceed the pre-defined threshold value. According to these results, our system can ensure the safety from unexpected obstacles while following the preceding vehicle automatically.

4.5.2. Vehicle Test Results

To verify whether the proposed algorithm is robust and can be implemented in real-time, we carry out vehicle-level test using the vehicle mentioned in section 4.2. In the test, the driver of preceding vehicle drives along the trajectory as shown figure 4.11 and freely adjusted the speed within 40km/h. In addition, two missions were added to the trajectory in order to validate the robustness of tracking algorithm and obstacle avoidance capability. The first mission is to pass a speed bump. Abnormal measurements can be occurred by the ego-vehicle's pitch motion while passing through the bumps. So we can confirm the robustness of tracking algorithm through this mission. And second mission is to pass the construction area. In this mission, Obstacle

avoidance capabilities can be validated.

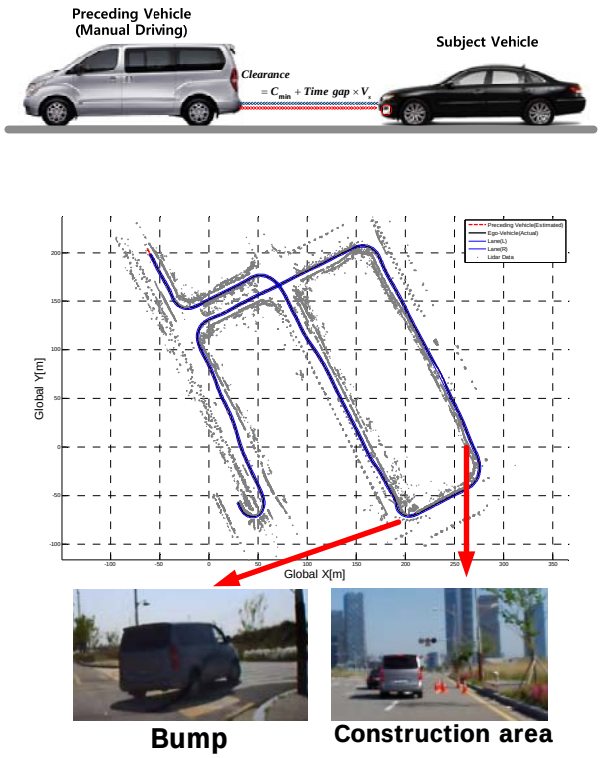
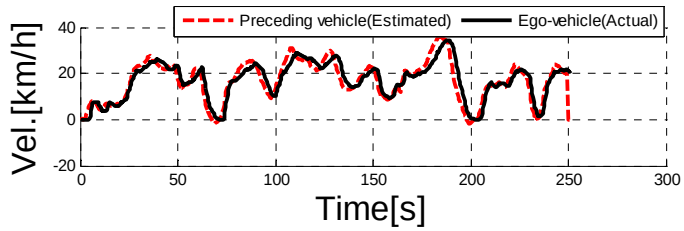


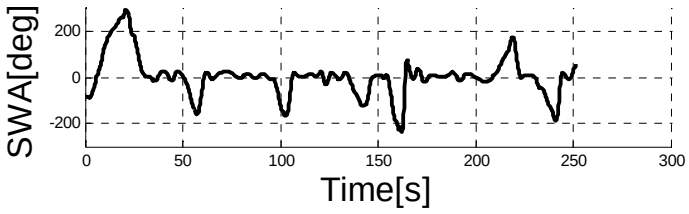
Figure 4.11 Test Scenario and Trajectory of the Preceding Vehicle

Our system accomplished this test scenario and test results are presented in figure 4.12. Figure 4.12 (a) shows the velocity of two vehicles: Preceding vehicle and Ego-vehicle. The Ego-vehicle drives automatically at the almost same speed as the preceding vehicle in order to maintain the predefined time gap. Figure 4.12 (b) shows the steering angle determined to follow the lead vehicle and avoid obstacle. Figure 4.12 (c) shows a constraint of the maximum yaw rate (Dashed line) and an actual yaw rate of ego-vehicle (Solid line). And the graph implies that our algorithm satisfies the yaw rate constraint. The maximum allowable lateral acceleration for lateral stability is set to be $2m/s^2$. Figure 4.12 (d) shows the deviation from the preceding

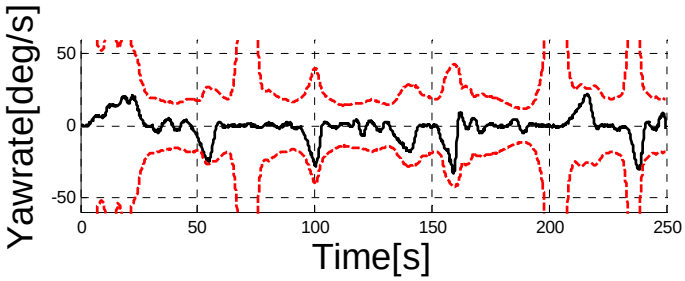
vehicle's trajectory. This is calculated using the estimated global position of the leader vehicle based on the laser scanning data. The standard deviation of lateral error stays within 30 cm. The standard width of a lane is 3.6m and the width of the car is about 1.8 m, so our system is possible to keep the lane if preceding vehicle is keeping the lane.



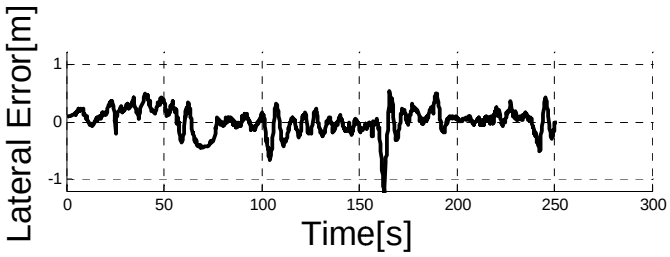
(a) Velocity



(b) Steering Wheel Angle (Ego-vehicle)



(c) Yawrate (Ego-vehicle)



(d) Lateral Distance Error

Figure 4.12 Vehicle test results

Figure 4.13 presents the scene of our test. To show that the system works automatically, the driver puts hands behind his head in the upper-left corner of Fig 4.13. And the result graph of vehicle tracking is presented in the upper-right corner of Fig.4.13. Red dots are point cloud of lidar and squares are ego-vehicle(black) and preceding vehicle(bule).

Figure 4.13 The scene of the experiment

Figure 4.14 presents the actual trajectory of ego-vehicle (Dashed line) and the estimated trajectory of preceding vehicle(Solid line). And the obstacles, obtained by sensor module, are expressed as gray dots. As shown the graph, ego-vehicle drove almost the same path as preceding vehicle to follow it while avoiding obstacles. From this result, it can be known that model predictive approach is very useful to follow the vehicle and avoid obstacles.

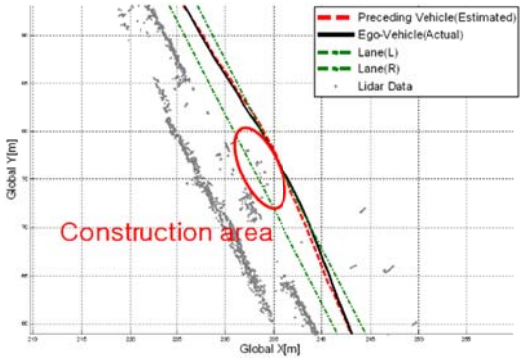


Figure 4.14 Trajectories of preceding vehicle and ego-vehicle

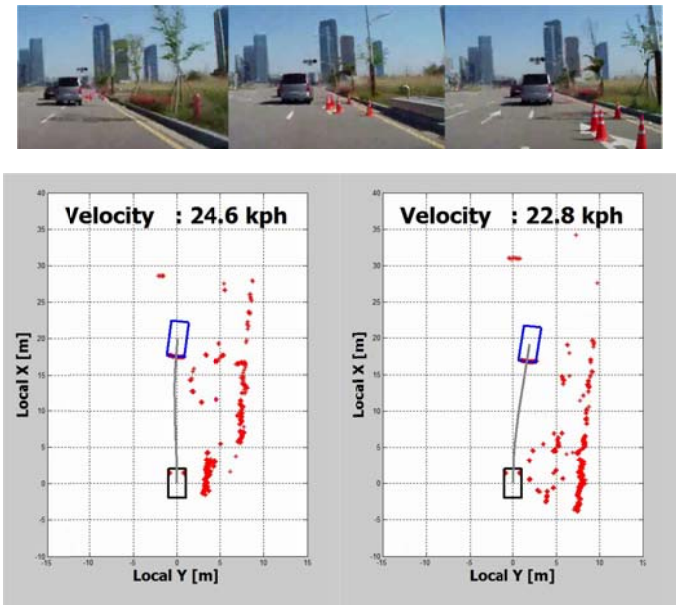


Figure 4.15 Test scene and generated path during the construction area

Chapter 5

Conclusions and Future Works

This dissertation has proposed an automated driving algorithm which is capable of automated driving on urban with guaranteed safety. The proposed algorithm consisted of the following two steps: a lane-level localization using AVM camera and MPC based automated vehicle control.

A lane-level localization algorithm consists of three parts: lane detection, position correction and localization filter. In lane detection, a well-known lane detection algorithm is adapted to be applicable to the commercial AVM module. To correct the vehicle position using detected lane and a digital map, an ICP-based map-matching algorithm is used, and an EKF is applied to fuse the map-matching result with the vehicle sensors' data. In order to improve the reliability of the proposed localization algorithm, a covariance estimator for ICP and a validation gate are designed in the localization filter. The lane-level localization algorithm has been successfully implemented on a test vehicle. Tests have been conducted on a proving ground. Vehicle test results have revealed that a precision within a few centimeters can be achieved, which is sufficient for automated vehicle control. However, there still remain challenging driving situations in terms of lane-level localization on urban

roads. Such situations include low- visibility conditions of the lane, lack of longitudinal position correction over a long period, increasing nonlinearity due to unusual maneuvering, and so on. Enhancing and verifying the proposed algorithm to achieve good accuracy for automated vehicle control in challenging driving situations on urban roads are the topics of our future research.

The MPC based control algorithm consists of two parts: a target tracking algorithm and a vehicle control algorithm. Vehicle tracking algorithm classifies the lead vehicle from a lidar point clouds based on the geometrical configuration and estimates its states using extended Kalman filter. The following algorithm calculates the optimal steering angle using model predictive control approach. Using a preceding vehicle information as initial guess of MPC can minimize the variations of constraints. It helps robust and safe real-time implementation of automated control algoritm. The performance of the proposed algorithm has been investigated via computer simulation and vehicle tests. The test result shows that our algorithm satisfies the performance required for urban automated driving.

Bibliography

[Allgöwer 00] Allgöwer, F., & Zheng, A. (Eds.). (2000). Nonlinear model predictive control. Basel: Birkhäuser

[Aly 08] Aly, Mohamed. "Real time detection of lane markers in urban streets."Intelligent Vehicles Symposium, 2008 IEEE. IEEE, 2008.

[Anderson 10] ANDERSON, S. J., PETERS, S. C., PILUTTI, T. E. & IAGNEMMA, K. 2010. An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios. International Journal of Vehicle Autonomous Systems, 8, 190-216

[Artus 03] G. Artus, P. Morin, C. Samson. "Tracking of an omnidirectional target with a unicycle-like robot: control design and experimental results". INRIA Technical report No 4849, June 2003

[Besl 92] Besl, Paul J., and Neil D. McKay. "Method for registration of 3-D shapes."Robotics-DL tentative. International Society for Optics and Photonics, 1992.

[Betaille 10] D. Betaille and R. Toledo-Moreo, "Creating enhanced maps for lane-level vehicle navigation," IEEE Transactions on Int. Transp. Systems,vol. 11, pp. 786–798, 2010.

[Bonnifait 01] Bonnifait, Philippe, et al. "Data fusion of four ABS sensors and GPS for an enhanced localization of car-like vehicles." Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on. Vol. 2. IEEE, 2001.

[Broggi 12] Broggi, Alberto, et al. "The VisLab Intercontinental Autonomous Challenge: an extensive test for a platoon of intelligent vehicles." International Journal of Vehicle Autonomous Systems 10.3 (2012): 147-164.

[Broggi 14] Broggi, Alberto, et al. "PROUD-Public road urban driverless test: Architecture and results." Intelligent Vehicles Symposium Proceedings, 2014 IEEE. IEEE, 2014.

[BSP 10] https://www.tu-braunschweig.de/stadtpilot

[CAUGV 03] Committee on Army Unmanned Ground Vehicle Technology,“Technology Development for Army Unmanned Ground Vehicles”, The National Academies Press,Washington, D.C., 2002

[Censi 07] Censi, Andrea. "An accurate closed-form estimate of ICP's covariance."Robotics and Automation, 2007 IEEE International Conference on. IEEE, 2007.

[Chen 10] A. Chen, A. Ramanandan, and J. A. Farrell, “High-precision lane-level road map building for vehicle navigation,” Position Location and Navigation Symp., pp. 1035–1042, 2010.

[Cui 14] Cui, Dixiao, et al. "Real-time global localization of intelligent road vehicles in lane-level via lane marking detection and shape registration." Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on. IEEE, 2014.

[Chang 13] Chang, Yu-Lung, Li-You Hsu, and Oscal T-C. Chen. "Auto-Calibration Around-View Monitoring System." Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th. IEEE, 2013.

[Chong 14] Chong, Z. J., et al. "Synthetic 2d lidar for precise vehicle localization in 3d urban environment." Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE, 2013.

[Chandak 02] P. Chandak, “Study and implementation of follow-the-leader – (BearcatIII)”. Master Thesis, University of Cincinnati, 2002.

[Chen, 92] Y. Chen, G. Medioni. ”Object Modeling by Registration of Multiple Range Images,” Proc. of the 1992 IEEE Intl. Conf. on Robotics and Automation, pp. 2724-2729, 1991.

[Cowan 03] N. Cowan, O.Shakernia, R.Vidal and S. Sasty. “Vision based follow the leader”, IROS, Lausanne, 2003

[Das 01] A.K.Das, R.Fierro, V.Kumar. “Real-time vision based control of a nonholonomic Mobile Robot”. ITTT Int. Conf. On Robotic & Automation, May 2001

[David 02] David A. Forsyth and Jean Ponce. Computer Vision: A modern approach. Prentice Hall, 2002

[Diehl 02] Diehl, M., Bock, H. G., Schloder, J. P., Findeisen, R., Nagy, Z., & Allgöwer, F.(2002). Realtime optimization and nonlinear model predictive control of process governed by differential-algebraic equations. Journal of Process Control, 12, 577–585.

[Dietmayer 05] Dietmayer, Klaus, et al. "Roadway detection and lane detection using multilayer laserscanner." Advanced Microsystems for Automotive Applications 2005. Springer Berlin Heidelberg, 2005. 197-213.

[Donald 93] B. Donald, P. Xavier, J. Canny, and J. Reif. Kinodynamic motion planning. Jounal of the Assocation for Computing Machinery, 40(5):1048–1066, November 1993.

[Douglas 73] Douglas, David H., and Thomas K. Peucker. "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature."Cartographica: The International Journal for Geographic Information and Geovisualization 10.2 (1973): 112-122.

[Du 04] J. Du, J. Masters, and M. Barth, “Lane-level positioning for in-vehicle navigation and automated vehicle location (avl) systems,” Proc IEEE Int. Transp. Systems, pp. 35–40, 2004.

[Falcone 08] FALCONE, P., BORRELLI, F., TSENG, H. E., ASGARI, J. & HROVAT, D. 2008. Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation. International journal of robust and nonlinear control, 18, 862-875

[Fang 07] Fang, Hui, Ming Yang, and Ruqing Yang. "Ground texture matching based global localization for intelligent vehicles in urban environment." Intelligent Vehicles Symposium, 2007 IEEE. IEEE, 2007.

[Farwell 99] M. Farwell, D. Caldwell. “RTK based vehicle tracking and unmanned operation for Agriculture”

[Findeisen03] Findeisen, R., & Allgöwer, F. (2004). Computational delay in nonlinear model predictive control. In Proc. int. symp. adv. control of chemical processes ADCHEM'03.

[Franke 13] Franke, Uwe, et al. "Making bertha see." Proceedings of the IEEE International Conference on Computer Vision Workshops. 2013.

[Gehrig 98] Gehrig, S.K,"A trajectory-based approach for the lateral control of car following systems", 1998 IEEE International Conference on (Volume:4)

[Google 15] https://www.google.com/selfdrivingcar/

[Gordon 03] Tim Gordon, Mark Howell, Felipe Brandao, "Integrated Control Methodologies for Road Vehicles", Vehicle System Dynamics, 2003, Vol. 40, pp.157-190

[Gray 13] GRAY, A., GAO, Y., HEDRICK, J. K. & BORRELLI, F. Robust predictive control for semi-autonomous vehicles with an uncertain driver model. Intelligent Vehicles Symposium (IV), 2013 IEEE, 2013. IEEE, 208-213.

[Gruyer 14] Gruyer, Dominique, Rachid Belaroussi, and Marc Revilloud. "Map-aided localization with lateral perception." Intelligent Vehicles Symposium Proceedings, 2014 IEEE. IEEE, 2014.

[HAC 14] http://www.businesskorea.co.kr/english/news/industry/6610-autonomous-vehicles-hyundai-motor-group-supports-students-applying-their-ideas

[Han 01] W. Han, Y. T. Chin. "Target tracking and pursue using laser scanner". 2nd Asian symposium on Industrial Automation and Robotics BITEC,May 2001

[Han 13] Han, Seung-Beom, Jong-Hwan Kim, and Hyun Myung. "Landmark-based particle localization algorithm for mobile robots with a fish-eye vision system."Mechatronics, IEEE/ASME Transactions on 18.6 (2013): 1745-1756.

[Haralick 98] Haralick, R. (1998). Propagating covariance in computer vision. Performance Characterization in Computer Vision, 1, 95–115.

[Hata 14] Hata, Alberto Y., Fernando Santos Osorio, and Denis F. Wolf. "Robust curb detection and vehicle localization in urban environments." Intelligent Vehicles Symposium Proceedings, 2014 IEEE. IEEE, 2014.

[Hillel14] Hillel, Aharon Bar, et al. "Recent progress in road and lane detection: a survey." Machine Vision and Applications 25.3 (2014): 727-745.

[Isogai 09] Isogai, Akira, et al. "Lidar Based Lane Recognition." 16th ITS World Congress and Exhibition on Intelligent Transport Systems and Services. 2009.

[IVFC 13] http://sites.ieee.org/itss/files/2014/04/v16n2April2014.pdf

[Levinson10] Levinson, Jesse, and Sebastian Thrun. "Robust vehicle localization in urban environments using probabilistic maps." Robotics and Automation (ICRA), 2010 IEEE International Conference on. IEEE, 2010.

[Kammel 08] Kammel, Sören, and Benjamin Pitzer. "Lidar-based lane marker detection and mapping." Intelligent Vehicles Symposium, 2008 IEEE. IEEE, 2008

[Karush 12] S. Karush, “They’re working: Insurance claims data show which new technologies are preventing crashes,” Status Report of the Insurance Institute for Highway Safety—Highway Loss Data Institute, vol. 52, no. 5, July 2012.

[Kim 02] H. Kim, D. Shin and S. Sastry, “Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles”, in America Control Conference, 2002, Proceedings of the 2002, vol.5, 2002, pp3576-3581

[Kim 11] BJ, Kim., KS, Yi, “Driver Risk Monitoring with Enhanced Path Prediction”, KSAE, pp 1809~1815.

[Kim 12] HK, Kim, KS, Yi, “Design of a Model Reference Cruise Control Algorithm”, SAE International Journal of Passenger Cars- Electronic and Electrical SystemsSeptember 2012 vol. 5no. 2 440-449

[Kou 08] Youseok Kou, Huei Peng, DoHyun Jung, “Worst-case evaluation for integrated chassis systems”, Vehicle System Dynamics, Vol. 46, 2008, pp.329-340

[Lee 02] C.-Y. Lee “Real Time Tracking of an unpredictable target Amidst Unknown obstacles”, ICARCV 2002.

[Lefebvre 04] T. Lefebvre, H. Bruyninckx, and J. D. Schutter, “Kalman filters for nonlinear systems: a comparison of performance,” The International Journal of Control., vol. 77, no. 7, 2004

[Lu 04] G. Lu, J. Huang, M. Tomizuka, "Vehicle lateral control under fault in front and/or rear sensors: final report', Research Report, UCBITS-PRR-2004-36, 2004.

[Madhavan 98] Madhavan, Raj, M. W. M. G. Dissanayake, and Hugh F. Durrant-Whyte. "Autonomous underground navigation of an LHD using a combined ICP-EKF approach." Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on. Vol. 4. IEEE, 1998

[Mallios 14] Mallios, Angelos, et al. "Scan matching SLAM in underwater environments."Autonomous Robots 36.3 (2014): 181-198.

[Marin 14] Marin, Luis, et al. "Event-Based Localization in Ackermann Steering Limited Resource Mobile Robots." Mechatronics, IEEE/ASME Transactions on 19.4 (2014): 1171-1182.

[Martinelli 08] Martinelli, Francesco. "Robot localization: comparable performance of EKF and UKF in some interesting indoor settings." Control and Automation, 2008 16th Mediterranean Conference on. IEEE, 2008.

[Moon 09] SW. Moon, IK. Moon, KS. Yi,” Design, tuning, and evaluation of a full-range adaptive cruise control system with collision avoidance”, Control Engineering Practice volume 17, Issue 4, 2009

[Nagy 07] Nagy, Z. K., Franke, R., Mahn, B., & Allgöwer, F. (2007). Real-time implementation of nonlinear model predictive control of batch processes in an industrial framework. In Assessment and future directions of NMPC. Berlin: Springer., p. 429.

[Nashashibi 08] Nashashibi, F., “Laser-based vehicles tracking and classification using occlusion reasoning and confidence estimation”, Intelligent Vehicles Symposium, 2008 IEEE, pp 847 - 852 .

[Ndjeng 11] A. N. Ndjeng, D. Gruyer, S. Glaser, and A. Lambert, “Low cost imu-odometer-gps ego localization with imm for unusual maneuvers,” Information Fusion (Elsevier), vol. 12, no. 4, pp. 264–274, 2011.

[Norgaard 00] M. Norgaard, N. K. Poulsen, and O. Ravn, “Advances in derivative-free state estimation for nonlinear systems,” in Revised edition of the technical report IMM-REP-1998-15, Technical University of Denmark, 2000

[NHTSA 13] NHTSA, "Preliminary Statement of Policy Concerning Automated Vehicles," Washington, 2013.

[Ohtsuka 04] Ohtsuka, T. (2004). A continuation/GMRES method for fast computation of receding horizon control. Automatica, 40, 563–574.

[Otsu 75] Otsu, Nobuyuki. "A threshold selection method from gray-level histograms."Automatica 11.285-296 (1975): 23-27.

[Park 03] Park, Soon-Yong, and Murali Subbarao. "An accurate and fast point-to-plane registration technique." Pattern Recognition Letters 24.16 (2003): 2967-2976.

[Plamen 08] Plamen Petrov, “Nonlinear Adaptive Control of a Two-Vehicle Autonomous Convoy Using a Look-Ahead Approach”,ISPRA'08 Proceedings of the 7th WSEAS International Conference on Signal Processing, Robotics and Automation

[Prakhya 15] Prakhya, Sai Manoj, et al. "A closed-form estimate of 3D ICP covariance." Machine Vision Applications (MVA), 2015 14th IAPR International Conference on. IEEE, 2015.

[Pomerleau 13] Pomerleau, François, et al. "Comparing ICP variants on real-world data sets."Autonomous Robots 34.3 (2013): 133-148.

[PROUD 13] http://vislab.it/proud/

[Qin 96] S. Qin and T. Badgwell. An overview of industrial model predictive control technology. Proceedings of CPC-V, 1996.

[Rezaei 07] Rezaei, Shahram, and Raja Sengupta. "Kalman filter-based integration of DGPS and vehicle sensors for localization." Control Systems Technology, IEEE Transactions on 15.6 (2007): 1080-1088.

[Rhudy 13] Rhudy, M., and Y. Gu. "Understanding Nonlinear Kalman Filters Part I: Selection of EKF or UKF." Interactive Robotics Letters (2013).

[Rusinkiewicz 01] Rusinkiewicz, Szymon, and Marc Levoy. "Efficient variants of the ICP algorithm." 3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on. IEEE, 2001.

[Schreiber 13] Schreiber, Markus, Carsten Knoppel, and Ulrik Franke. "Laneloc: Lane marking based localization using highly accurate maps." Intelligent Vehicles Symposium (IV), 2013 IEEE. IEEE, 2013.

[SHALOM 87] BAR-SHALOM, Y. & FORTMANN, T. E. 1987. Tracking and data association, Wiley-Interscience.

[Singh 12] Singh, Krishna Kant, Kirat Pal, and M. J. Nigam. "Shadow detection and removal from remote sensing images using NDI and morphological operators."Int J Comput Appl 42.10 (2012): 37-40.

[Skog 09] Skog, Isaac, and Peter Händel. "In-car positioning and navigation technologies—A survey." Intelligent Transportation Systems, IEEE Transactions on 10.1 (2009): 4-21.

[Suganuma 11] Suganuma, Naoki, and Takahiro Uozumi. "Precise position estimation of autonomous vehicle based on map-matching." Intelligent Vehicles Symposium (IV), 2011 IEEE. IEEE, 2011

[Swaroop 96] Swaroop, D Hedrick, J.K. “String stability of interconnected systems” Automatic Control, IEEE Transactions on volume:41, Issue: 3, 1996

[Takehiko 98] Takehiko Fujioka, “Vehicle Following Control in Lateral Direction for Platooning” Vehicle System Dynamics: International Journal of VehicleMechanics and Mobility Volume 29, Supplement 1, 1998

[Tao 13] Tao, Zhenning, et al. "Lane marking aided vehicle localization. " Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on. IEEE, 2013.

[Teck 04] Teck Chew Ng, “Vehicle Following with Obstacle Avoidance Capabilities in Natural Environments”, Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on volume:5

[Thrun 10] S. Thrun. What we're driving at, 2010.

[Toledo 07] R. Toledo-Moreo, M. Zamora-Izquierdo, B. Ubeda-Miarro, and A. Gomez-Skarmeta, "High-integrity imm-ekf-based road vehicle navigation with low-cost gps/sbas/ins," IEEE Transactions on Intelligent Transportation Systems, vol. 8, no. 3, pp. 491–511, 2007.

[Trenholm 15] R. Trenholm. Nokia sells Here maps business to carmakers Audi, BMW and Daimler, 2015.

[Urmson 08] Urmson, Chris, et al. "Autonomous driving in urban environments: Boss and the urban challenge." Journal of Field Robotics 25.8 (2008): 425-466.

[Wang 14] Wang, Kai, Yun-Hui Liu, and Luyang Li. "A simple and parallel algorithm for real-time robot localization by fusing monocular vision and odometry/AHRS sensors." Mechatronics, IEEE/ASME Transactions on 19.4 (2014): 1447-1457..

[Wang 13] Wang, Lei, Paul D. Groves, and Marek K. Ziebart. "GNSS shadow matching: improving urban positioning accuracy using a 3D city model with optimized visibility scoring scheme." Navigation 60.3 (2013): 195-207.

[Xu 06] Xu, Li, et al. "Shadow detection and removal in real images: A survey."Computer Vision Lab, Dept. of Computer Science and Engineering, Shanghai JiaoTong University, Shanghai (2006).

[Zlot 14] Zlot, Robert, and Michael Bosse. "Place recognition using keypoint similarities in 2D lidar maps." Experimental Robotics. Springer Berlin Heidelberg, 2009.

초 록

도심도로 상황에서 어라운드 뷰 모니터링 시스템 기반 차량의 위치 추정 및 모델 예측 기반 자율 주행 차량 제어

최근 주요 자동차 업체들은 수년 안에 자율주행 자동차를 판매하겠다고 공표하였다. 그리고 이러한 차량들이 합법적으로 공공도로를 달리기 위한 법들도 통과되었다. 심지어 일부 선도기업들에서는 제한적으로 자율 주행을 경험해 볼 수 있는 새로운 차량들을 출시하였다. Distronic Plus (Mercedes-Benz), Driving Assistant Plus (BMW), Highway Driving Assist (Hyundai Motor Company) 와 같은 시스템들이 대표적인 제한적 자율주행 시스템이다. 이러한 시스템 탑재한 차량은 운전자가 어떠한 조작을 하지 않아도 자동으로 전방차량과의 거리를 유지 할 뿐 아니라 조향 어시스트를 통해서 차선 유지를 해준다. 하지만 아직 이러한 시스템은 운전자가 핸들을 잡고 있는 동안에만 작동하고 운전자가 일정시간 이상 아무 조작을 하지 않으면 작동을 멈춘다. 하지만 제한적인 작동 조건을 고려하더라도, 이러한 시스템이 운전자에게 편안함과 안전을 제공하는 것은 분명하다.

현재 출시된 자율주행 시스템이 제한적으로 작동하게 된 주된

이유는 다양한 도로 환경에서 양산 차량의 환경인지 시스템의 성능이 강건성과 유용성 측면에서 요구 성능을 만족시키지 못하기 때문이다. 사실 이러한 문제는 고정밀 관성 항법시스템과 3 차원 레이져 스캐너와 같은 최신의 센서로 대부분 해결가능하다. 하지만 이러한 시스템을 양산 차량에 적용하기에는 비용문제로 인하여 또다른 장벽이 존재한다.

그러므로 본 논문에서는 현재 양산되어 가격적으로 경제적이고 기술적으로 완성된 차량용 센서들을 조합하여 도심도로에서 완전한 자율 주행이 가능한 시스템개발을 목표로 한다. 전체 알고리즘은 차선 수준 측위와 모델 예측 제어 기반 차량 제어로 구성 된다.

차선 수준 측위는 디지털 지도 대비 수 cm 수준의 정확도로 차량의 위치를 추정하는 것을 의미한다. 저가의 GPS 를 이용하여 이러한 위치 정확도를 만족시키기 위해서는 센서 융합 알고리즘이 필수적이다 본 논문에서 차선 수준 측위를 위한 센서 융합 알고리즘은 저가형 GPS 이외에 어라운드 뷰 모니터링 시스템.과 차량 센서를 추가적으로 활용한다. 이를 위해 제안한 알고리즘은 차선 인식, 위치 보정 그리고 측위 필터로 구성된다. 차선인지 시스템은 AVM 모듈을 기반으로 하여 획득한 차량 주변 영상 정보를 이용하여 정확하게 차선의 위치를 인지한다. 이 위치 정보와 디지털 지도 정보를 이용하여 위치 보정 알고리즘은 iterative closest point (ICP) 기법을 이용하여 차량의 위치 보정 량을 계산한다. 이렇게 계산된 위치 정보는 확장 형 칼만 필터를 기반으로 하여 차량 센서와 저가형 GPS 정보를 융합하여 최종적인 차량의 위치를 추정하게 된다. 이때 좀더 높은 정밀도의 센서 정보 융합을 위하여 Haralick’s method 를 이용하여

위치 보정 량의 공분산을 계산하였다.

고속도로와 대조적으로 도심도로에서는 주변 차량들과 상대적으로 작은 거리를 유지하며 주행한다. 따라서 레이더,라이다 그리고 카메라와 같은 센서들은 주변 차량들에 의해 매우 제한을 받는다. 이러한 경우 갑자기 나타나는 장애물에 의해 자율주행 차량은 쉽게 위험에 빠지게 된다. 이러한 문제를 극복하기 위해서 본 논문에서는 전방 차량의 거동 정보를 자 차량의 제어에 활용하는 방법에 대해 연구하였다. 이러한 접근 방식은 먼저 전방 차량의 요 모션을 포함한 거동 정보를 라이다 센서를 이용하여 정확히 예측하고 이를 바탕으로 전방 차량을 정확히 추종 하는 궤적을 생성한다. 이는 곧 도심 도로 자율주행 제어 문제를 제약조건을 만족시키며 최적으로 경로를 추종하는 문제로 귀결시킨다. 본 논문에서는 모델 예측 기반 제어 기법을 이용하여 이 문제를 해결하였다.

본 논문에서 제안된 자율주행 차량을 위한 측위와 제어 알고리즘의 성능은 컴퓨터 시뮬레이션과 실차 실험을 통해 검증되었다. 실험 결과는 제안한 방법이 수 cm 수준의 측위 정확도와 도심 도로에서 자율주행 할 수 있는 강건한 차량 제어 성능을 가졌음을 보여주었다.

주요어: 자율 주행 자동차, 차량 측위, 차선 지도 생성, 맵 매칭, 반복 최근 접점, 모델 예측 제어, 차량 인지 및 추종

학　번: 2008-22869