國立臺灣大學電機資訊學院資訊工程研究所

碩士論文

Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Master Thesis

基於光學雷達自駕車之修復系統

Self-recovery System for LiDAR-based Autonomous Driving

王若芸
Ruo-Yun Wang

指導教授：逄愛君博士
Advisor: Ai-Chun Pang, Ph.D.

中華民國 106 年 9 月
September 2017

doi:10.6342/NTU201704349

國立臺灣大學碩士學位論文
口試委員會審定書

基於光學雷達自駕車之修復系統

Self-recovery System for LiDAR-based Autonomous Driving

本論文係王若芸君（學號 R04922118）在國立臺灣大學資訊工程學系完成之碩士學位論文，於民國 106 年 9 月 29 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

逄愛君

莊清智（指導教授） 徐宏民

系主任 莊永裕

doi:10.6342/NTU201704349

致謝

非常感謝逄老師的指導，給予我許多建議並引導我朝正確的研究方向前進，對於過去沒有甚麼研究經驗的我來說，真的有非常大的助益。同時感謝詩梵學姊和彥儒學長願意抽空和我討論研究相關的問題，讓我能思考得更加深入周全，還有清智學長和小泉學長的鼓勵和經驗分享以及研究室的大家和我的家人朋友們的幫助和鼓勵，才能順利完成我的論文。

doi:10.6342/NTU201704349

中文摘要

近年來隨著各種科技發展的成熟，自駕車不再被視為科幻電影和小說中的幻想，許多知名公司及車商像是 Google，Intel，Apple，Tesla，BMW 等皆已積極投入自駕車研發的市場，其牽涉到人工智慧、深度學習、先進視覺、感測、雲端處理、物聯網等多項複雜的技術，所以要實現自駕車普及化仍需至少十年以上甚至更長的時間，然而自駕車仍是未來汽車發展的趨勢。在運作上自駕車需要多種感測器協助偵測周遭環境，並運用偵測的數據分析決策及行為，大部分自駕車最主要的感測器為光學雷達 (LiDAR)，但現階段對光學雷達 (LiDAR) 故障的應對方式為緊急停車並等待支援，相當被動且喪失自駕車的優勢，此篇提出兩種機制並運用車對車傳輸 (V2V Communication) 來進行修復，目的是讓發生故障的自駕車無須人未支援便能安全的抵達目的地，最後透過模擬, 驗證兩種機制對故障自駕車修復的效果皆能達到目標，並分析它們對無線網路的影響。

關鍵字：車對車通訊、自駕車、光學雷達

doi:10.6342/NTU201704349

Abstract

For the past few years, the Self-Driving Car (SDC) is not regarded as a fantasy depicted in science fiction along with various kinds of techniques are mature.Many well-known companies and automakers such as Google, Intel, Apple, Tesla and BMW have put a lot of effort into developing SDC.The technologies involve AI, deep learning, computer vision, cognitive sensor, cloud computing and IoT thus the SDC becomes common and affordable at least ten years, even longer time.However, the SDC is undoubtedly a future trend for driving and needs various kinds of sensors to detect the nearby environment. Additionally, it uses sensing data in order to plan actions. On most SDCs, LiDAR acts as the main sensor. However, However, we discovered that the current approach to resolving a LiDAR malfunction is very passive and causes the SDC advantage to be lost. In this thesis, we propose two strategies in order to deal with LiDAR failure through V2V Communication. The aim is to recover from the breakdown and safely drive the SDCs to their destination without human support. Finally, we validated the performance of both strategies through simulation and provide insights for wireless networks.

Keywords：V2V Communication, Self-Driving Car, LiDAR

doi:10.6342/NTU201704349

Contents

doi:10.6342/NTU201704349

doi:10.6342/NTU201704349

List of Figures

doi:10.6342/NTU201704349

List of Tables

doi:10.6342/NTU201704349

Chapter 1

Introduction

1.1 Introduction

The Self-Driving Car (SDC), also called autonomous vehicle or driverless vehicle, is no longer a futuristic idea. Since the end of 2014, most car manufacturing companies have been developing their own version of the SDC and investing many funds in its development [1]. Although many companies and automakers declare that their SDCs will appear on the market in the 2020s to 2030s, the SDC become common and affordable, probably in the 2030s to 2040s [2].These companies include Google Waymo, Intel, Mobileye, Uber, and Apple Titan. , while the Automakers automakers include Toyota, Ford, BMW, Tesla, Nissan, and Volvo.

Compared with traditional vehicle we can estimate some potential benefits and costs [3].From the benefit viewpoint, the SDC will provide safety, better driving experience, traffic efficiency, green environment.According to the statistical report of KPMG, which is a professional service company [2].Driver error contributes to more than 90 percent of traffic accidents, the SDC will reduce crashes 90 percent.Without controlling the car, a driver can rest or work in the car. May increase road capacity and reduce traffic congestion.For the environment,may reduce pollution and improve fuel efficiency.Also suit for sick, blind, drunk and poor driving skill driver.By the angle of costs and problem, the SDC will bring higher costs such as various onboard sensors, additional risks for system failure, security and privacy concern for hacking and cyber attack.Somebody will lose jobs such as taxi, bus and truck driver.Finally, we need new policy and insurances.

Without a driver control, the SDC relies on multiple sensors and internal High-Definition map (HD map) to plan actions. The primary sensor of the SDC is light detection and ranging (LiDAR) which is responsible for determining distances to obstacles and building

doi:10.6342/NTU201704349

dynamic 3D point cloud maps of the surrounding environment.It is a powerful sensor with high accuracy and offers a 360-degree field of view.In Figure 1.1, we show the placement of sensors about Google Waymo Self-driving car [4] and its LiDAR is mounted on the roof and rotates 360^o horizontal and 26.9^o vertical to scan the nearby environment about radius of 120 m.However, However, hardware devices have a limited life-span. The LiDAR life-span is usually 3-5 years and is influenced by the SDC running frequency. LiDAR malfunctioning can cause serious strain on the SDC. By using the remaining sensors, the SDC cannot know the entire information of the nearby environment. The breakdown SDC may execute incorrect actions and cause serious accidents. More specifically, for others sensors, the radar has a long range of approximately 200 m,however its accuracy is not enough. Even if we place the radar in all the directions of the SDC, the low accuracy may have an effect on the decision. On the contrary, Ultrasonic can provide accurate data, but has a short range of approximately 1-10 m. The camera has some drawbacks; namely, it is easily affected by weather and light conditions, such as rain, night, fog and snow, and the detection of the distance is not accurate enough.

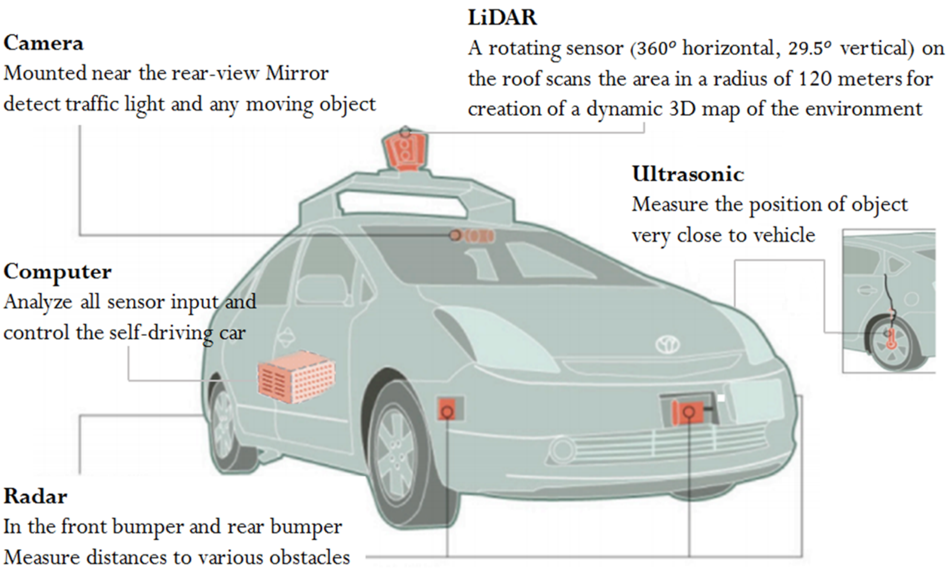


Figure 1.1: Google Waymo Self-driving car

Many AI technologies are used to detect sensor failures and estimate sensor bias from analyzing sensed data [5].In this thesis, we do not describe these technologies about detecting sensor failure in detail and just classify LiDAR failure into three types.We show

doi:10.6342/NTU201704349

the reaction for different types of LiDAR failure as follows:

- *Type 1 - Data with little bias*

  The system can tolerate LiDAR bias within 5 cm and use fault tolerance to recover.

- *Type 2 - Data with a serious bias*

  Emergency stop.

- *Type 3 - Can not sense data*

  Emergency stop.

In the next section of motivation, we would focus on LiDAR failure about type 2 and type 3 which are not recovered by system its own.

1.2 Motivation

The developing market of the SDC can classify into three areas separately Carry Passengers, Delivering Service and In-Car Entertainment.

1. *Carry Passengers*

   This area mainly provides service to passengers such as self-driving bus,self-driving taxi about ride-sharing or ride-hailing and personal SDC.

2. *Delivering Service*

   This area mainly delivers the cargo.Many companies such as Amazon, DeNA and Takkyubin, Otto and Uber are manufacturing delivering service.

3. *In-Car Entertainment*

   Without controlling the car people can do any thing on the SDC.It brings about the market of In-Car Entertainment such as providing audio or video.

The scenario about this paper focuses on delivering service.It is that a SDC which provides delivering service is running on the road in an urban environment and its major sensor ”LiDAR” suddenly malfunctions.Unfortunately, the way to face this situation

doi:10.6342/NTU201704349

today is passive like that SDC broadcasts the message to notice its neighbors before emergency stops.As we know, the most important requirement of a delivery service is that the cargo must be delivered to the destination. When a situation occurs that pre-vents this requirement from being reached, the breakdown SDC needs to wait for another car to carry out the remaining delivery services. However, this approach may cause plenty of traffic problems as follows:

(i) *Unpredictable waiting time* : the road rescue time takes at least 30-40 minutes in an urban environment. The number of freight company delivering trucks is the limit and the supporting car which is assigned by the freight company may not be dis-patched immediately. The breakdown SDC may cause traffic congestion problems, if it waits while being parked on the road.

(ii) *Requiring large human support and losing the advantage of the SDC* : when the supporting car arrives, we need to discharge the goods and transfer them onto the supporting car, reset the delivery path, and tow or drive the breakdown SDC to a repair spot. All the operations are completed by human support staff, and moving the goods on the road consumes time and energy, in addition to being very dangerous.

From the above description we know that there are many drawbacks with regard to current approaches. We propose a way to recover from LiDAR failure and let the breakdown car arrive at the its destination by utilizing only its own non-human support. The challenges of designing the recovery approach are as follows : (i) Ensuring that the breakdown SDC can arrive at the destination. ; (ii) Considering safety and avoiding collisions. ; (iii) Considering traffic congestion.

In this paper, we propose a self-recovery system for LiDAR malfunction on the SDC through V2V communication. The objective is to minimize the total time of the breakdown SDC costs the drive from the departure to the destination. The contributions of this paper are summarized as follows. Firstly, we show the scenario about major sensor of the SDC malfunctions on delivering service and prove it is a critical problem. Secondly, we propose strategies to recover the failure through V2V communication and let breakdown car arrive at the destination with its own. The way would consider safety, traffic congestion and

doi:10.6342/NTU201704349

avoid collision. Finally, due to the simulation, we evaluate the method and give insight to wireless networks.

1.3 Thesis Organization

The remainder of this thesis is organized as follows. In Chapter II, the background of High-Definition Map (HD Map) and the architecture and operation about the SDC are introduced. In Chapter III, we propose three potential recovering ways and compare them. In Chapter IV, we describe the system model and problem formulation. In Chapter V, we present the solution design. In Chapter VI, the simulation results are presented and the performance evaluations are discussed. Finally, we conclude the thesis in Chapter VII.

doi:10.6342/NTU201704349

Chapter 2

Background

In this chapter, we introduce the HD map, which is an essential ingredient for SDC, and illustrates the SDC architecture and operation.

2.1 High-Definition Map

The HD map is a highly detailed, three-dimensional, computerized map specifically for the SDC, in order to support safe and more efficient operation.Unlike traditional navigation map such as Google map which is served for people about 10-25 m accuracy.The human can use their keen sense and smart brain to interpret abstract mark in the conventional map.HD map is served for the SDC, not the human being.Therefore, the contents of the HD map require very detail and the location and coordination also require high accuracy with a precision of approximately 10-20 cm.”If we want to have autonomous cars everywhere, we have to have digital maps everywhere” said chief technology officer at Mobileye.Many companies consider that the HD map will play a key role for the SDC. A few commercial HD map implementations, such as HERE [6] and TomTom [7], are already in existance. Figure 2.1 shows HERE HD live map.

The contents containing in HD map can be classified into two different layers by update frequency [8].One is a static layer, while the another one is the a dynamic layer.The contents belonging to the static layer is as follows:

- *lane model* : lane line, lane slope, lane curve, lane center line and lane separation point
- *lane object* : traffic sign, traffic light, stop line, crosswalk, division island, roadside building and tree

doi:10.6342/NTU201704349

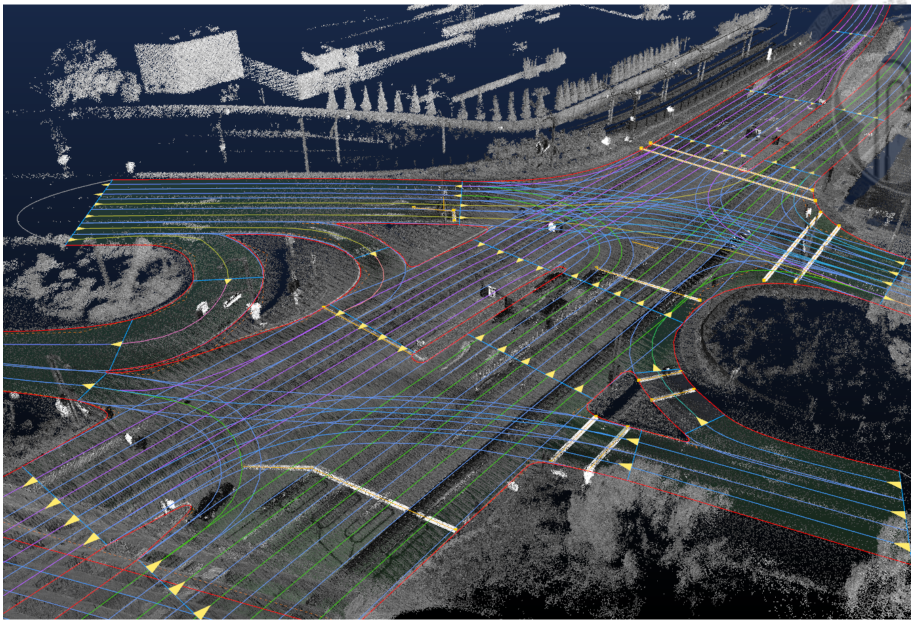


Figure 2.1: HERE HD live map

- *lane attribute* : lane speed limit and lane high limit

The static layer is generated by a map company and is updated once per month. Map company uses its collective cars which contain at least two Lidars and four cameras to collect data. The generation of the static layer needs human support and spends more time.The contents of the dynamic layer contain some real-time traffic information, such as traffic flow conditions, congestion, construction, accident and local weather, and are updated once per minute. The dynamic layer is generated by live data from roadside sensors, government agencies and SDCs travelling on the road which analyze their sensing data generally camera data before submitting them to the cloud based on the map company's specification.

HERE presents a specification in [9] where sensor data is submitted as messages of various contents.Each message consists of four elements :

1. *Envelope* : information on the transmitting vehicle includes Submitter (OEM or system vendor name) and vehicle ID
2. *Path* : spatial information includes position, speed and their time stamp
3. *Path events* : proper sensor values such as vehicle status (fuel state, emergency braking), sign recognition (congestion, construction, accident) and environment status

doi:10.6342/NTU201704349

(road surface type)

4. *Path media* : video or audio

With HD map sensors on the SDC such as LiDAR, the SDC can concentrate on scanning the moving objects on the road, detecting traffic change, and receiving real-time traffic information continuously.

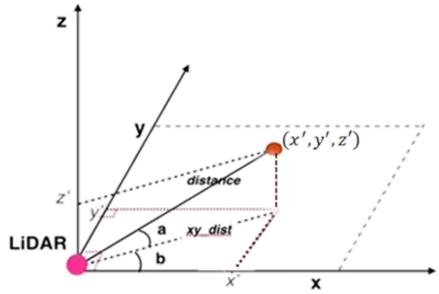
2.2 Architecture and Operation

Developing the SDC needs to integrate and fuse many technologies such as Big Data, Machine Learning, Artificial Intelligence, Cloud computing and Computer Vision. There are three basic components in SDC operation; namely, Sensing, Perception and Decision [10].

Sensing is in charge of sensing the surrounding environment of the SDC by using various types of sensing techniques.The common onboard sensors include LiDAR, Radar, camera, Ultrasonic, GPS, IMU and Wheel Odometer. All sensing data will be analyze by computer [11].LiDAR is a prominent SDC sensor and responsible for determining distances to obstacles and building 3D point cloud maps of the environment by using its laser scanners, which emit light beams. Each laser scanner with fixed vertical angles collects the reflective point distance and intensity and records its corresponding time stamp and azimuth. The most biggest LiDAR company today is Velodyne and it releases many types of LiDAR such as HDL-16E , HDL-32E and HDL-64E [12].For example, Google Waymo SDC uses Velodyne HDL-64E which is 64 laser scanners, frequency 10-15 Hz, scanning range 120 m, 360-degree horizontal FOV and 29.6-degree vertical FOV.The principle behind creating a 3D point cloud map is that the SDC uses distances, vertical angle, and azimuth in order to calculate coordination, and uses intensity in order to determine the texture of the object and mark the point type. The principle of transforming raw data to three-dimensional coordination displays on Figure 2.2.

Perception consists of two functions; namely, Localization and Object Recognition. Localization finds the position of the SDC and Object Recognition determines the kind of

doi:10.6342/NTU201704349



$$(x', y', z') = (\cos(b) * xy\_dist, \sin(b) * xy\_dist, \sin(a) * distance)$$

$$xy\_dist = \cos(a) * distance$$

Figure 2.2: The principle about LiDAR three-dimensional coordination

obstacles. We describe the integral operation of the SDC on Figure 2.3. The SDC uses IMU, Wheel Odometer and the previous pose obtained from the internal map in order to predict the current pose. IMU is generally with six-axis processing components including three-axis acceleration detector and three-axis angular velocity detector.The acceleration detector checks the force on four surfaces and the angular velocity detector calculates acceleration.Wheel Odometers are mounted on two front wheels and record their total revolution and the angle of turning left and turning right.The SDC can predict current pose by using the previous pose on the internal map, acceleration getting from IMU, ongoing distance and the turning angle from Wheel Odometer.

However, the prediction pose is not accurate enough. We also need the the support of GPS, camera, and LiDAR, in order to localize the SDC.As As we know, LiDAR can transform a reflective point to 3D coordination and the origin is the LiDAR position. With these 3D point clouds, we can use the Iterative Closest Point (ICP), which is one of the widely used algorithms in aligning three-dimensional models.For each point in the new 3D point cloud match, the closest point in the point cloud of the internal map computes the MSE due to the match, and adjusts the pose. We can obtain an exact pose after many iterations. Finally, we use sensor fusion techniques, such as a Kalman Filter, in order to fuse the GPS, IMU, Wheel Odometer, camera and LiDAR data [13] and obtain the new pose.The precision of the new pose can reach up to 10-20 cm.

By using Automated Mapping Technology, we map all points to a new coordination based on the new pose.Object Recognition is mainly carried out by camera and LiDAR.Computer vision technology is used to analyze the image and identify the signal. AI

doi:10.6342/NTU201704349

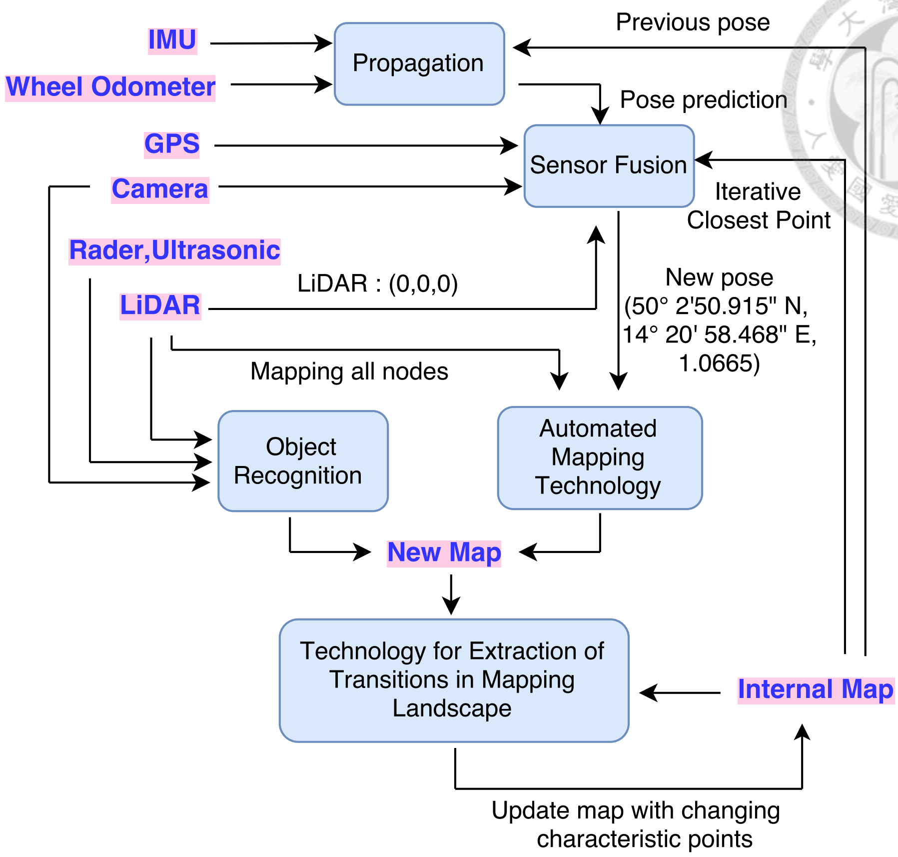


Figure 2.3: The operation of the SDC

and machine learning are used in order to train the point cloud profile generated by LiDAR and identify the type of object. The new map integrates the results of Automated Mapping Technology and Object Recognition. We do not update the entire internal map and just update the changing characteristic points. The changing characteristic transition points in the mapping landscape are obtained by Technology, by comparing the new map to the internal map. The changing characteristic points are generally various dynamic nodes such as other vehicles, pedestrians, bikes and any moving object.

In the Decision step the SDC analyzes the results of the past two components and uses the internal HD map for path planning, action prediction, and obstacle avoidance.

doi:10.6342/NTU201704349

Chapter 3

Overview of Potential Approaches

Currently, the breakdown is handled by making an emergency stop upon LiDAR failure, since there is no other method to recover from this situation. Below, we will propose various approaches that can potentially achieve a recovery from these types of LiDAR failure.

- *Using Backup Sensor*
- *Using High-Definition map*
- *Using V2V support*

3.1 Using Backup Sensor

This is the simplest way to recover from LiDAR failure by just switching on the backup onboard LiDAR. However, the cost of LiDAR is very expensive and amounts to approximately $60,000-$80,000.In order to enable the SDC to become mainstream and affordable, the cost is a critical point to be addressed. Although this is the simplest approach, from a cost viewpoint, it is infeasible for ubiquitous manufacturing in the future.

3.2 Using High-Definition Map

The breakdown SDC downloads the HD map in the cloud in order to exchange the internal map, and uses the new map in order to keep running. However, There are two reasons due to which it is infeasible to use the HD map for recovery. First, the update frequency happens at a very small scale. Secondly, the content of the HD map in the cloud does not contain dynamic nodes.

doi:10.6342/NTU201704349

3.3 Using V2V Support

The concept is that The adjacent SDC with high related sensing data can transfer its information to the breakdown SDC. When the breakdown SDC receives the current information, it can update its internal map and use the new map to plan actions.Considering instantaneity, a neighboring SDC can immediately transfer its information sensing by its sensors through V2V communication. With consideration to traffic fluency, stopping on the road may cause traffic congestion; however, the condition of stopping is that there is no SDC in close proximity to the breakdown SDC. If there is no other SDC travelling close to the break-down SDC, the probability of traffic congestion is low. We consider that using V2V support is the most suitable way to recover from LiDAR failure in the SDC, with consideration to information instantaneity and traffic flow.

What is the look about the information would the neighboring SDC transmitting to the breakdown SDC? In the past chapter, we illustrate the operation about the SDC in detail and observe that all sensing data by various sensors would analyze and fuse to the new map.The internal map uses the changing characteristic points getting from the new map comparing to the internal map to update the current state.We aggregate the changing characteristic points into an updating file like Figure 3.1.When the breakdown SDC receives the updating file, it can update its old internal map, use the new map to plan actions and keep running on the road.

The updating file is a LAS file which is the common format to record point cloud and includes the information about changing characteristic points during past 100 ms within radius of 120 m.The time and the area is based on LiDAR scanning range and scanning frequency.The LAS file is a binary file and Figure 3.2 shows the content of the updating file.In this file, the point information contains the 3D coordination, point class and point intensity.

The recovering procedure about how to fulfill two main functions without normal LiDAR is shown in Figure 3.3. In Object Recognition, the breakdown SDC can still use its camera and points on the updating file, including point class information such as the

doi:10.6342/NTU201704349

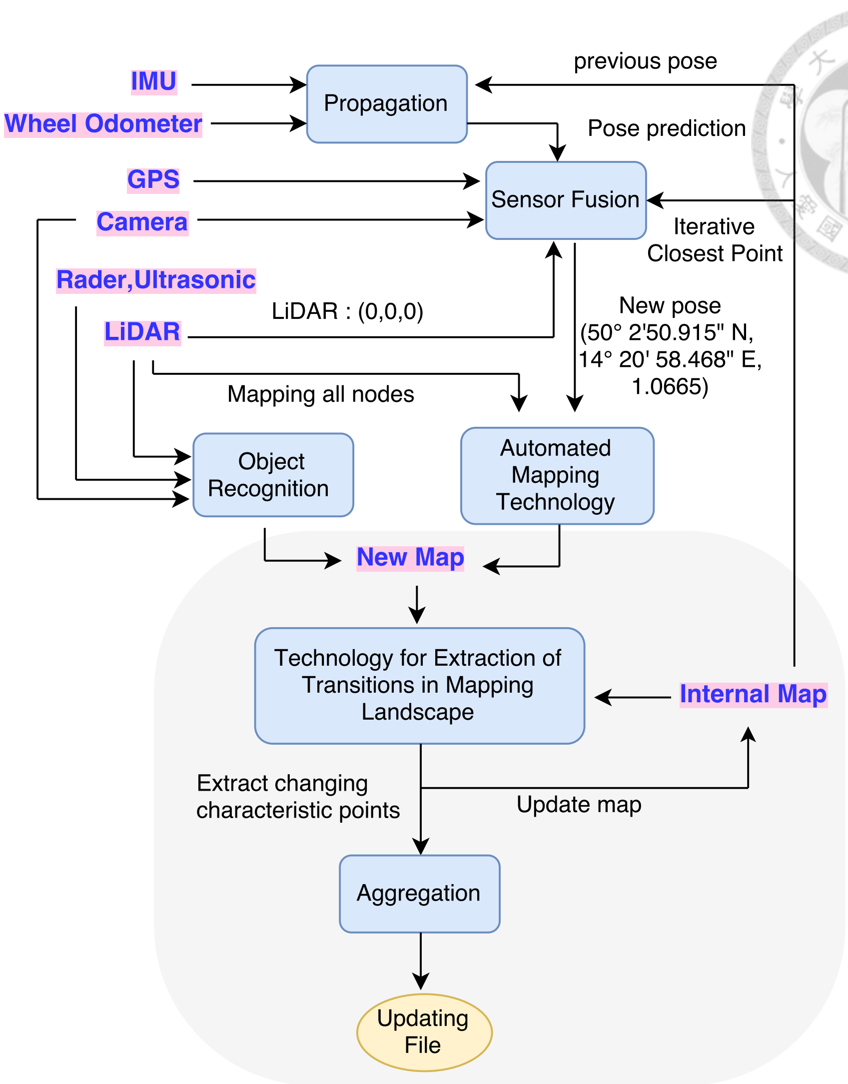


Figure 3.1: The generation of the updating file

vehicle or other obstacles. In the Localization part, the breakdown SDC can also use its IMU, Wheel Odometer, GPS, and previous pose obtained from the internal map in order to predict the pose, and apply the ICP in order to match the point cloud on the updating file and internal map. Finally, by integrating these results and sensing data through sensor fusion, we obtain the accurate location of the breakdown SDC. Therefore, by using the remaining sensors and the information in the updating file, we Therefore, by using the remaining sensors and the information in the updating file, we can update the changing part and accurately calculate the location of the breakdown car on the map. After updating the map to its current state and determining the location of the breakdown car on the map, it can plan actions and keep travelling on the road.

doi:10.6342/NTU201704349

```
4c41 5346 0000 0000 5952 114e 0d0f 2934
a051 4909 2d3f 587e 0100 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 4c69 4441 5220
416e 616c 7973 7420 6279 2050 726f 4c6f
```


Figure 3.2: The content of updating file

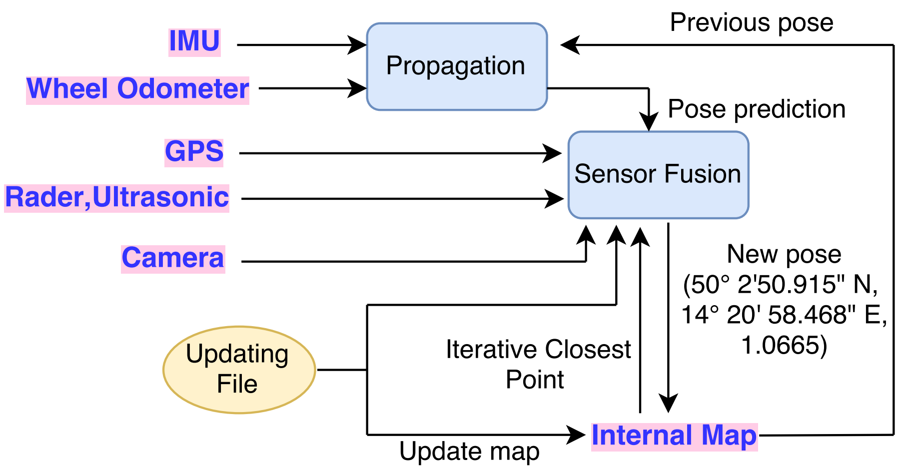


Figure 3.3: The recovery of the LiDAR malfunction

In this chapter, we propose three potential recovering approaches and discuss their feasibility.We consider that using V2V support is the most suitable way to recover from LiDAR failure in the SDC, with consideration to information instantaneity and traffic flow. Below, we will design a recovery system based on V2V support.

doi:10.6342/NTU201704349

Chapter 4

System Model and Problem Formulation

4.1 System Model

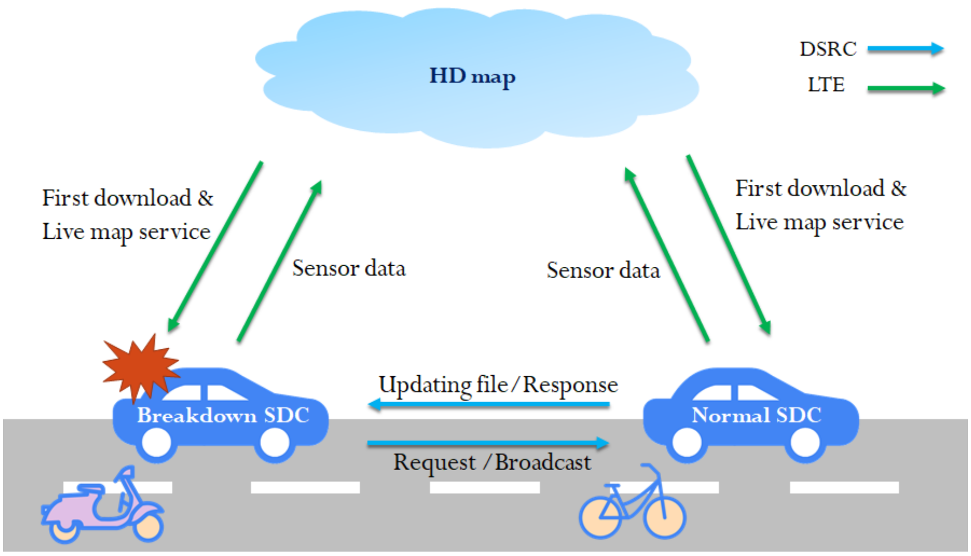


Figure 4.1: System model

The system model shows in Figure 4.1.In this system, there are numerous vehicles including normal SDC, breakdown SDC, bikes, and motorcycles travelling on the road in an urban environment. The SDCs downloads the HD map, from departure to destination, before starting to drive. When the LiDAR on the SDC breakdowns, the SDC will immediately broadcast to its nearby SDCs, in order to notify them about the situation, and then it will stop. When the nearby SDCs receive the message, they transmit the updating file to the breakdown SDC. After the breakdown, the SDC receives the updating file, and uses it to update the internal map and even apply the data to updating the HD map in the cloud, while continuing its travel to the destination. During the trip, the HD map in the cloud still provides live map services for all SDCs regardless of whether the SDC is normal or breakdown. We use the Dedicated Short Range Communication (DSRC), also known as

doi:10.6342/NTU201704349

IEEE 802.11p, which is a suite of standards for communication between vehicles, and uses LTE to communicate between vehicles and the cloud. Our requirements of this system is as follows. Firstly, the breakdown SDC must arrive to the destination. Secondly, the breakdown SDC should avoid collision during the trip.

As shown in Table 4.1, the set V consists of all kinds of the vehicles in the system. Before SDCs start, the system initiates some flags to record their current states. It sets F_{stop}, $F_{breakdown}$ and F_{resume} to zero. Let $N_{neighbor}$ denote the number of neighboring normal SDC for the breakdown SDC. The definition of collision is that when moving, the breakdown SDC cannot obtain the updating file and the distance between the front of the vehicle is smaller than the smallest safe distance; thereby, collision occurs. Equation (4.1) specifies the smallest safe distance d_{safe} for the breakdown SDC in V , where g and u are the gravitational acceleration and the coefficient of asphalt road friction, respectively.

$$d_{safe} = \frac{v^2}{2 \times g \times u} + 0.2 \times v \tag{4.1}$$

We calculated angle in order to determine the relative position of the breakdown SDC and other vehicles. Equation (4.2) specifies the angle between the breakdown SDC and the neighboring vehicle to determine whether the vehicle is in the front of the breakdown SDC, where A represents the current location of the breakdown SDC, A' represents the previous location of the breakdown SDC and B represents the location of another vehicle.The angle θ is a vector from A to A' and a vector from B to A.When it is over the threshold, then it means B is in front of A.

$$\theta = |\angle(\overrightarrow{A'A}, \overrightarrow{AB})| \tag{4.2}$$

Various metrics have been proposed in order to evaluate the design method in this system. These metrics can be classified into two aspects by separat-ing the recovery efficiency for the breakdown SDC, and the impact on the wireless network. For the break-

doi:10.6342/NTU201704349

down SDC, we use two evaluation metrics. The first one is total time. The total time of the breakdown SDC, costs the drive from the departure to the destination. The second metric is the collision rate, which is the rate of breakdown SDC collisions that occur during the trip. The collision rate represents the ratio of the number of collisions to the total number of estimated times. For the wireless network, we used four evaluation metrics. The first one is broadcast overhead, which is the total number of broadcast packets transmitted during the trip in order to recover the system; the second one is the Broadcast burst, which is the maximum broadcast overhead in a unit of time. The third one is the Transmission overhead, which is the total amount of times that the normal SDCs transmits its updating files to the breakdown SDC during the trip, for the purpose of system recovery; The fouth one is transmission burst which is the maximum transmission overhead in a unit of time. The unit of time is set to 100 ms, since the synchronization interval of the DSRC channel is 100 ms and the period of transmission and broadcast is also 100 ms.

4.2 Problem Formulation

Our objective is to minimize the total time subject to the constraint r_c < 5%. Input is the set V and outputs are $time$, $transmission$, $broadcast$, $burst_b$ and $burst_t$. We have some assumptions for the problem as follows. Firstly, normal SDCs, bikes and motorcycles do not crash with the breakdown SDC. Secondly, the quality of the DSRC channel is good which means that the channel is not congested. Finally, the path loss about the radio shadow caused by obstacles does not severely influence the transmission. The notations are summarized in Table 4.1.

doi:10.6342/NTU201704349

Table 4.1: Summary of notation

Symbol	Definition
V	The set of the vehicles
F_{stop}	The flag of stop state for the breakdown SDC
$F_{breakdown}$	The flag of breakdown state for the breakdown SDC
F_{resume}	The flag of resume state for the breakdown SDC
$N_{neighbor}$	The number of neighbors
$N_{collision}$	The number of vehicles crashed by the breakdown SDC
r_c	The collision rate of the breakdown SDC
d_{safe}	The smallest safety distance
$d_{neighbor}$	The distance of neighbor
d_{min}	The shortest distance between the breakdown SDC and another normal SDC
$burst_t$	Transmission burst
$burst_b$	Broadcast burst
$transmission$	Transmission overhead
$time$	Total time
$broadcast$	Broadcast overhead
u	The coefficient of asphalt road friction
g	Gravitational acceleration
v	The speed of vehicle
θ	The angle determine whether the vehicle is in the front of the breakdown SDC

doi:10.6342/NTU201704349

Chapter 5

Method Design

In this chapter, we present our solution design for the recovery.When occurring breakdown, the system would immediately operate.We propose two strategies separately the Decision and the Non-Decision.The main difference between these two strategies is about the behavior of the breakdown SDC.Two different methods would propose and depict in the following sections.

5.1 Decision

As the name implies, the concept of the Decision strategy is that the breakdown SDC is active, in order to decide which normal SDCs should transmit the updating file to it. The procedure is as follows. (i) When the breakdown occurs, the SDC immediately broadcasts to nearby vehicles in order to notify that $F_{breakdown}$ is one, its location , and stops on the road and sets F_{stop} to one. (ii) When the nearby normal SDCs obtain the broadcast message, they calculate the distance between the breakdown SDC. If the distance is lower than $d_{neighbor}$, they send the distance to the breakdown SDC. (iii) When the breakdown SDC obtains these distances, it chooses a normal SDC with d_{min} and notifies it in order to send its file. (iv) When the chosen SDC gets the message, it sends its updating files to the breakdown SDC. (v) When the breakdown SDC receives the updating file, it uses the file and resumes its travel on the road and sets F_{stop} to zero and F_{resume} to one. (vi) If $N_{neighbor}$ is zero or or the breakdown SDC cannot receive any updating file to use, the breakdown SDC still stops on the road and sets F_{stop} to one and F_{resume} to zero.

The broadcast to notify the nearby SDC in step i is periodic. In order to reduce the transmission overhead, the normal SDC cannot send its updating file to the breakdown SDC when the breakdown SDC is waiting for the traffic light. We show the detail pseudo

doi:10.6342/NTU201704349

code about the Decision strategy on Algorithm 2.In the trip, the system continuously detects whether the LiDAR on the SDC is normal or breakdown.The detect breakdown function shows on Algorithm 1.When the system detect breakdown, the recovering strategy would start.In line 7, the system examines whether the breakdown SDC arrives to the destination.If it successfully arrives to the destination, the system would terminate and record the total costing time.In line 16, the system checks the condition of collision.The equation of $R.V_b = R.V_j$ states that the breakdown SDC b is on the same road with vehicle j.The formula of $\theta \leq \delta$ states that the vehicle j is in the front of the breakdown SDC b.In line 24, it reduces transmission overhead and $V_b.wait() \neq 0$ represents the breakdown SDC b is not waiting for traffic light.In line 27, the system determines the stop condition.When there are not any vehicle nearby the breakdown SDC, then stop it on the road.In line 33, the system determines the resume condition.When there are any vehicles nearby the breakdown SDC, then resume it to run on the road.

5.2 Non-Decision

Unlike Decision, the concept of the Non-Decision strategy is that the breakdown SDC does not decide which normal SDC should transmit its updating file to it. The procedure is as follows. (i) when the breakdown occurs, the SDC immediately broadcasts to nearby vehicles in order to notify that $F_{breakdown}$ is one, its location , and stops on the road and sets F_{stop} to one. (ii) when the nearby normal SDCs receive the broadcast message, they calculate the distance between the breakdown SDC. If the distance is lower than $d_{neighbor}$, they send their files to the breakdown SDC. (iii) when the breakdown SDC receives these files, it merges their files for use, and resumes to run on the road and sets F_{stop} to zero and F_{resume} to one. (iv) If $N_{neighbor}$ is zero or or the breakdown SDC cannot receive any updating file to use, the breakdown SDC still stops on the road and sets F_{stop} to one and F_{resume} to zero.

The broadcast for notifying the nearby normal SDC in step i is event triggered. When the number of received updating files is lower than the previous unit of time or the state of the breakdown SDC is “stopping”, the breakdown SDC rebroadcasts the message

doi:10.6342/NTU201704349

in order to inform its nearby vehicles. It uses the same Decision strategy in order to reduce transmission overhead. We show the detail pseudo code about the Non-Decision strategy on Algorithm 3.In line 15, the system checks whether the breakdown SDC is waiting for the traffic light in order to reduce transmission overhead.In line 19, the system checks the condition of collision such like the Decision strategy.In line 25, the system determines the stop condition.In line 31, the system determines the resume condition.In line 35, the system determine the condition of rebroadcasting and $S.V_b$ represent the speed of the breakdown SDC b.

Algorithm 1 Detect breakdown

function $Detect\ breakdown$
    When detect breakdown then $F_{breakdown} \leftarrow 1$
    if $F_{breakdown} = 1$ **then**
        $V_b.stop()$
        $F_{stop} \leftarrow 1$
        $V_b.broadcast()$
        $broadcast++$
    end if
    return $F_{stop}, F_{breakdown}, broadcast$
end function

doi:10.6342/NTU201704349

Algorithm 2 Decision

Input: The vehicles set V on the map
Output: $time, transmission, broadcast, burst_b$

Initialize: $F_{stop} \leftarrow 0, F_{breakdown} \leftarrow 0, F_{resume} \leftarrow 0, N_{neighbor} \leftarrow 0, N_{collision} \leftarrow 0, burst_b \leftarrow 0, transmission \leftarrow 0, broadcast \leftarrow 0, time \leftarrow 0$
for $i = 1$ to $duration$ **do**
  if $F_{breakdown} = 0$ **then**
    $Detect\ breakdown$
  end if
  if $F_{breakdown} = 1$ **then**
    if $V_b.reach() = 1$ **then**
      $time \leftarrow i$ and $exit()$
    end if
    $N_{neighbor} \leftarrow 0, N_{collision} \leftarrow 0$
    for all $j \in V$ **do**
      if $dist(V_b, V_j) < d_{neighbor}$ **then**
        if $type.V_j = SDC$ **then**
          $V_j.broadcast()$ and $N_{neighbor} + +$
        else if $type.V_j = bike$ or $type.V_j = motorcycle$ **then**
          if $d < d_{safe}$ and $R.V_b = R.V_j$ and $\theta \leq \delta$ **then**
            $N_{collision} + +$
          end if
        end if
      end if
      Find the shortest d from all neighbors and assign to d_{min}
    end for
    $V_b.broadcast()$
    if $V_b.wait() \neq 0$ and $N_{neighbor} > 0$ **then**
      $V_{d_{min}}.transmit(V_b)$
    end if
    if $N_{neighbor} = 0$ and $F_{stop} = 0$ **then**
      $F_{stop} \leftarrow 1, F_{resume} \leftarrow 0$
      if $N_{collision} > 0$ and $S.V_b \neq 0$ **then**
        occur collision and $exit()$
      end if
      $V_b.stop()$
    else if $N_{neighbor} > 0$ and $F_{stop} = 1$ **then**
      $F_{stop} \leftarrow 0, F_{resume} \leftarrow 1$
      $V_b.resume()$
    end if
    if $F_{stop} = 0$ and $S.V_b \neq 0$ **then**
      $broadcast \leftarrow broadcast + N_{neighbor} + 1$
      $transmission + +$
    end if
    $V_b.broadcast()$ and $broadcast + +$
  end if
end for
Find maximum $N_{neighbor}$ from all duration and assign $N_{neighbor} + 2$ to $burst_b$
return $transmission, broadcast, time, burst_b$

doi:10.6342/NTU201704349

Algorithm 3 Non-Decision

Input: The vehicles set V on the map
Output: $time, transmission, broadcast, burst_t$
Initialize: $F_{stop} \leftarrow 0, F_{breakdown} \leftarrow 0, F_{resume} \leftarrow 0, N_{neighbor} \leftarrow 0, N_{collision} \leftarrow 0, temp \leftarrow 0, burst_t \leftarrow 0, transmission \leftarrow 0, broadcast \leftarrow 0, time \leftarrow 0$
for $i = 1$ to $duration$ **do**
    if $F_{breakdown} = 0$ **then**
        $Detect\ breakdown$
    end if
    if $F_{breakdown} = 1$ **then**
        if $V_b.reach() = 1$ **then**
            $time \leftarrow i$ and $exit()$
        end if
        $temp \leftarrow N_{neighbor}, N_{neighbor} \leftarrow 0, N_{collision} \leftarrow 0$
        for all $j \in V$ **do**
            if $dist(V_b, V_j) < d_{neighbor}$ **then**
                if $type.V_j = SDC$ **then**
                    $N_{neighbor}++$
                    if $V_b.wait() \neq 0$ **then**
                        $V_j.transmit(V_b)$
                    end if
                else if $type.V_j = bike$ or $type.V_j = motorcycle$ **then**
                    if $d < d_{safe}$ and $R.V_b = R.V_j$ and $\theta \leq \delta$ **then**
                        $N_{collision}++$
                    end if
                end if
            end if
        end for
        if $N_{neighbor} = 0$ and $F_{stop} = 0$ **then**
            $F_{stop} \leftarrow 1, F_{resume} \leftarrow 0$
            if $N_{collision} > 0$ and $S.V_b \neq 0$ **then**
                occur collision and $exit()$
            end if
            $V_b.stop()$
        else if $N_{neighbor} > 0$ and $F_{stop} = 1$ **then**
            $F_{stop} \leftarrow 0, F_{resume} \leftarrow 1$
            $V_b.resume()$
        end if
        if $S.V_b = 0$ or $(S.V_b \neq 0$ and $temp > N_{neighbor})$ **then**
            $V_b.broadcast()$ and $broadcast++$
        end if
        if $F_{stop} = 0$ and $S.V_b \neq 0$ **then**
            $transmission \leftarrow transmission + N_{neighbor}$
        end if
    end if
end for
Find maximum $N_{neighbor}$ from all duration and assign it to $burst_t$
return $transmission, broadcast, time, burst_t$

doi:10.6342/NTU201704349

Chapter 6

Simulation

In this chapter, we validate the proposed two recovering ways and evaluate the performance in terms of our metrics under different density.The simulation map was downloaded from OpenStreetMap (OSM) which is an open world map.The map of OSM not only contains road topology but also includes traffic light with real timing.Simulation of Urban Mobility (SUMO), which is a free and open traffic simulation suite, was used as the traffic simulator. The simulation was developed in MATLAB and needs to extra setup the library of Traffic Control Interface (TraCI4) which allows the user to interact with SUMO in a client-server scenario in which Matlab acts as the client and SUMO as the server.

Figure 6.1: The road topology of the map

6.1 Simulation Environment

The simulation environment consisted of one square kilometer of the Taipei City Zhongzheng District, which has one breakdown SDC, some normal SDCs, bikes, and motorcycles. The number of SDCs within the area have a density of 100, 200, or 300, and correspond to the

doi:10.6342/NTU201704349

same number of bikes plus motorcycles. The average density within one square kilometer in the urban environment is 200.The value of 100 represents the off-peak hour, and 300 represents the rush hour. The odometer of the breakdown SDC, from departure to destination, is 1.5 km and breakdown SDC breakdown after travelling 150 m from departure.

Table 6.1: Parameter Settings

Parameter	**Description**	**value**
$maxSpeed_{SDC}$	Max speed of SDC	16 m/s
$maxSpeed_{motorcycle}$	Max speed of motorcycle	16 m/s
$maxSpeed_{bike}$	Max speed of bike	6 m/s
$d_{neighbor}$	The distance of neighbor	80 m
d_{LiDAR}	The scanning range of LiDAR	120 m
d_{DSRC}	The transmission range of DSRC	300 m
$f_{broadcast}$	The frequency of broadcast	100 ms
$f_{transmission}$	The frequency of transmitting updating file	100 ms
$s_{broadcast}$	The size of broadcast packet	100 $bytes$
s_{update}	The size of updating file	53 KB
L_v	The most longest length of vehicle	18 m
D	Data rate	27 $Mbps$
δ	The threshold of the θ	90^o
u	The coefficient of asphalt road friction	0.8
g	Gravitational acceleration	9.8 m/s^2

The simulation settings are summarized in Table 6.1.In this table, we specifically explain the size of updating file, the distance of neighbor and the threshold of the θ.In the first part about size, the size of one scan generating point clouds is about 40 MB.After filtering unnecessary parts such as tree and building, the size of point clouds reduces to 0.8 MB and then using LASzip which is the common lossless compression for LAS file with compression rate about 5-15 times.Finally, the size of point clouds is about 53 KB.Because the content of updating file is the changing characteristic points from one scan, the size of updating file is not over 53 KB.In the second part about the neighbor, the condition of neighbor is that the LiDAR scanning area of the SDC should cover the breakdown SDC and the width of the lane which the breakdown SDC locates and should contain the distance of the smallest safe distance plus the longest vehicle length in front of the break-

doi:10.6342/NTU201704349

down SDC to avoid the collision.The length of the most longest vehicle is about 18 m.We discover that if the distance between the SDC and the breakdown SDC is lower than 80 m, the condition can be satisfied and the SDC can be a neighbor.Finally, the value of the threshold θ refers to [14].

6.2 Simulation Results

In this section, we evaluate the performance of proposed two recovering methods based on the metrics introduced on chapter four and compare them.For each density, we created one hundred different routes. Each route has the same path and odometer for the breakdown SDC, and a random path for other vehicles.

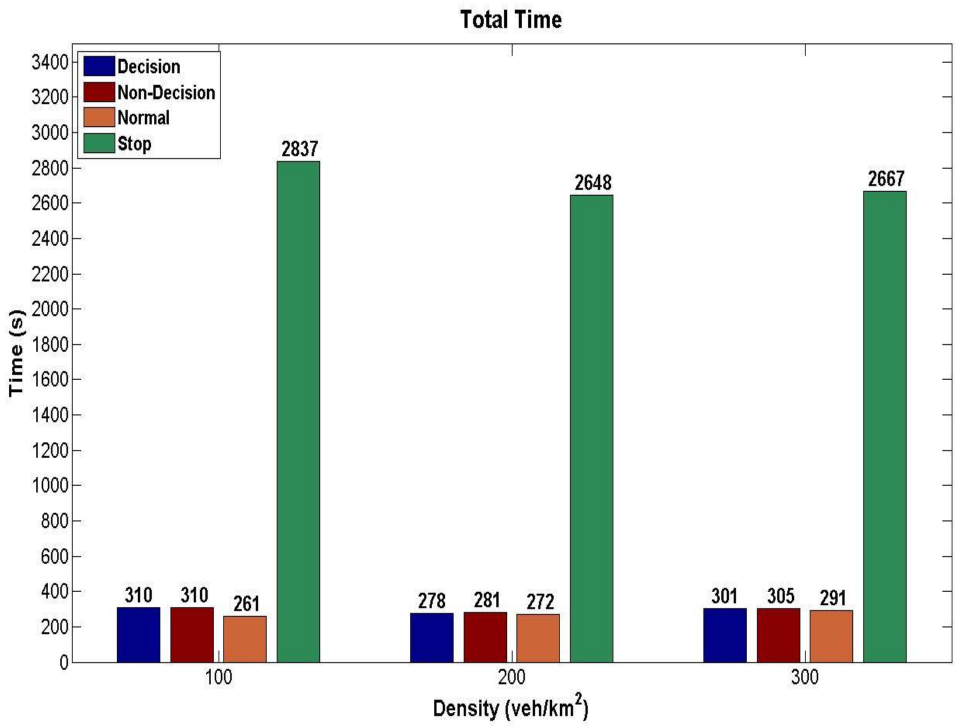


Figure 6.2: The total time

On Figure 6.2, we show the total time for two methods, normal mode, and emergency stop for each density. The normal mode represents that the breakdown SDC does not breakdown during the trip. Emergency stop is the current approach. The results of these four ways are the average of one hundred routes for each density on the map. We can observe that the performance of both methods for these densities is similar. Although the

doi:10.6342/NTU201704349

total time of both methods is higher than the normal mode, the gap between them is small, and does not exceed 5% based on the total time of the normal mode. By using the current approach the total time would be very high, in comparison to other methods. This includes road rescue time of approximately 30 minutes, discharge and carry time of approximately 10 minutes, and remaining trip time. The Decision and Non-Decision approaches can improve the total time by at least 88-90%. For the normal mode, when the density is bigger, the total time is higher due to the traffic flow. For other approaches, we observed that the total time for a density of 100 is the longest. Since this wastes a lot of time for waiting, vehicles travel into the neighboring area. We also observe that the total time for the density of 300 was longer than that 200 due to the traffic flow.

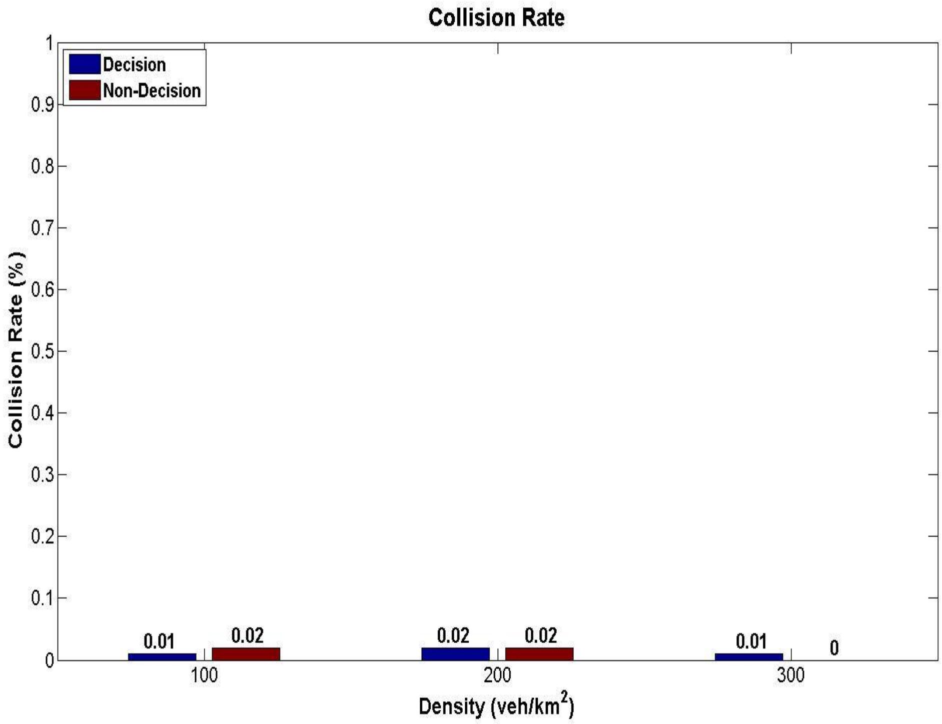


Figure 6.3: The collision rate

The collision rate shows on Figure 6.3.The results were obtained by the estimated number of routes, with collisions occurring in one hundred routes. We observed that the collision rates of both methods were similar, and did not exceed our 5% constrain. Actually, the collision rate would be lower than the results due to support from remain-ing sensors such as camera and radar.

doi:10.6342/NTU201704349

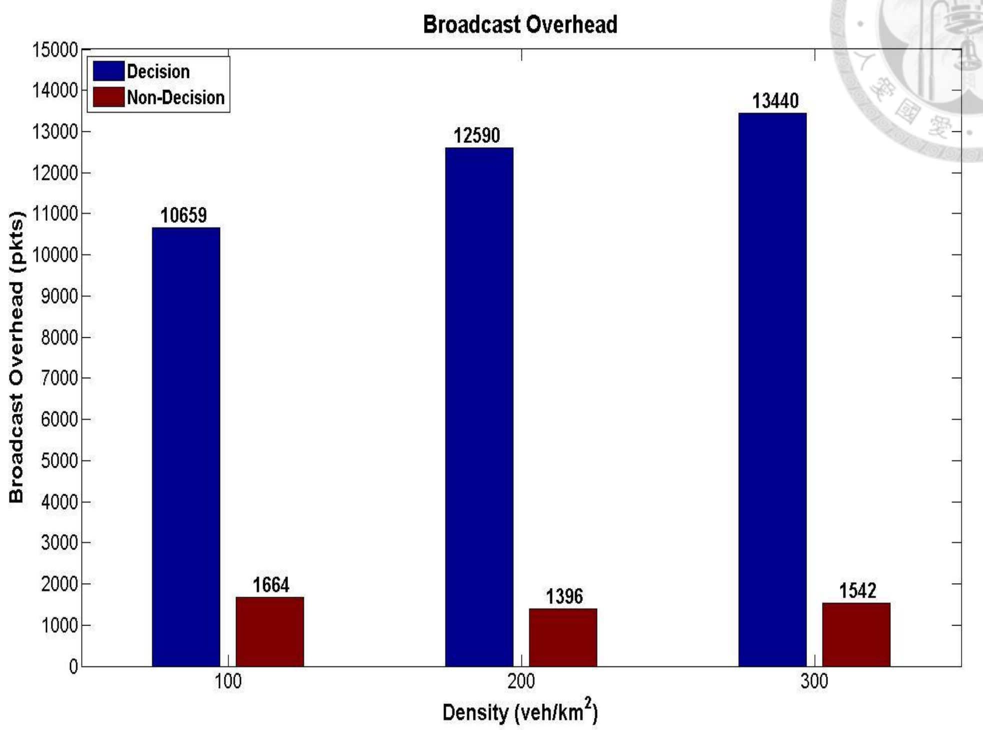


Figure 6.4: The broadcast overhead

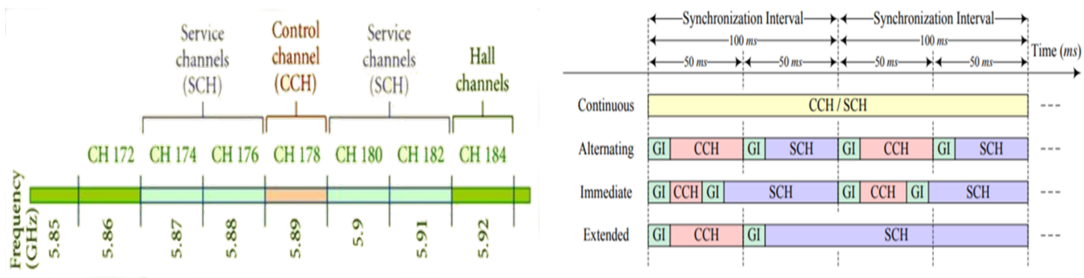
For the broadcast overhead on Figure 6.4, we also averaged one hundred routes for each density on the map. The broadcast overhead of Decision was much higher than that of Non-Decision. Apparently, the reason was that the Decision strategy is periodic instead of event triggered, like Non-Decision. When the density was higher, the number of nearby SDCs was bigger and the broadcast overhead was also higher. However, the gap between the broadcast overhead of Non-Decision on different densities was small, since the timing of broadcast was event triggered. In the broadcast overhead, we only know the total number of broadcast packets during the trip and it is hard to observe the effect on the wireless network. Therefore, we should focus on the broadcast overhead in a unit of time and analyze the load of the channel.

Table 6.2: Decision : Maximum broadcast overhead in the unit of time

Density (veh/km^2)	**Average** ($pkts$)	**Max** ($pkts$)
100	13	23
200	19	25
300	20	27

doi:10.6342/NTU201704349

For the broadcast burst in Table 6.2, we just aim at the Decision strategy. The reason is that Non-Decision only sends one broadcast packet in a unit of time, at most, and the impact on the wireless network is low. The results include the average of the maximum number of broadcast packets in a unit of time for about one hundred routes with different density. The maximum of these is the maximum number of broadcast packets in one unit of time. When the density is higher, the number of nearby SDCs is bigger and the result is higher, we would directly focus on the maximum value of 27 pkts in order to discuss whether the DSRC channel can sustain the number in one unit of time.



(a) Channel allocation

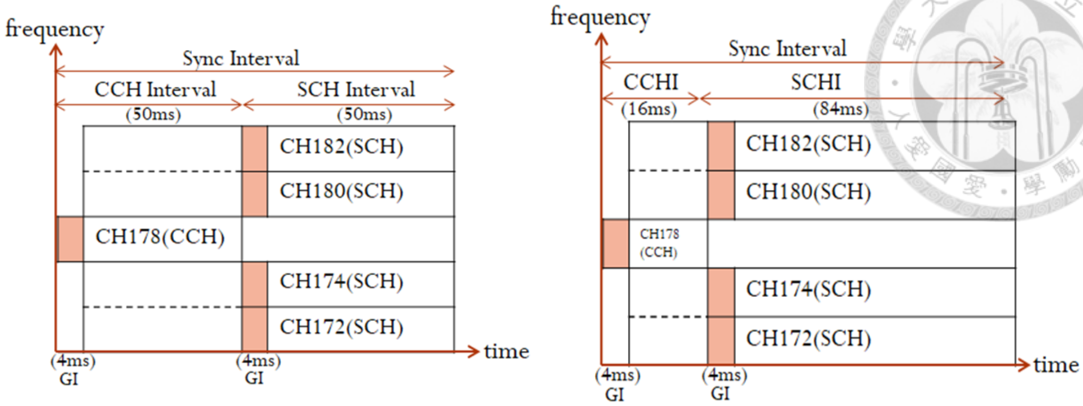
(b) Channel access option

Figure 6.5: Channel allocation and Channel access option of DSRC

The DSRC 5.9 GHz spectrum was divided into smaller operational channels as shown in Figure 6.5(a).It contains one control channel (CCH) and four service channels (SCH).The channel 178 as a CCH is exclusively used to transmit safety messages while the channel 174, 176, 180 and 182 as a SCH is typically used to communicate IP-based services.The channel 184 as a the High Availability Low Latency (HALL) channel is being left for future use and channel 172 is unused [15].

There are four types of channel access options as shown in Figure 6.5(b).The default option is alternating access which the synchronization interval (SI) was evenly composed of the CCH interval (CCHI) and the SCH interval (SCHI) [15].When it switches to the other channel, it should wait for a Guard Interval (GI) is default 4 ms.For the default alternating access, the remaining time of transmitting 27 pkts is 46 ms by CCHI minus GI as shown in Figure 6.6(a).What is the need of costing time about transmitting 27 pkts considering CSMA/CA using in the DSRC ?

doi:10.6342/NTU201704349



(a) Alternating access (b) Immediate access

Figure 6.6: Channel access scheme

In Table 6.1, we know the size of a broadcast packet is 100 bytes and the data rate is 27 Mbps, so it would cost 30 μs to transmit one broadcast packet.The common time and packet size using on CSMA/CA are as follows : $slot\ time = 9$ μs, $DIFS = 34$ μs, $SIFS = 32$ μs, $RTS = 20$ bytes, $CTS = 14$ bytes and $ACK = 12$ bytes.Now, we compute the time of transmitting 27 pkts in the unit of time.One broadcast packet for the breakdown SDC to its neighbor : $DIFS + Backoff + RTS + SIFS + CTS + SIFS + DATA + SIFS + ACK = 34 + 2 \times 9 + 6 + 32 + 4 + 32 + 30 + 32 + 4 = 192$ μs and then 25 broadcast packets for all neighbors to the breakdown SDC : $DIFS + Backoff + RTS + SIFS + CTS + SIFS + DATA + SIFS + ACK = 4692$ μs.Finally, one broadcast packet for the breakdown SDC to its neighbor : $DIFS + Backoff + RTS + SIFS + CTS + SIFS + DATA + SIFS + ACK = 34 + 0 \times 9 + 6 + 32 + 4 + 32 + 30 + 32 + 4 = 174$ μs.The total time is $192 + 4692 + 174 = 5.058$ ms and far less than 46 ms.It can be seen that the maximum broadcast burst can also be sustained by the DSRC channel.

For the transmission overhead on Figure 6.7, we also average one hundred routes for each density. The transmission overhead of Non-Decision was much higher than that of Decision, since the Non-Decision strategy is that all neighbors transfer their files to the breakdown SDC instead of selecting the closest SDC, like Decision. When the density is higher, the number of nearby SDCs is larger and the transmission overhead of Non-Decision is also higher. However, the gap between the transmission overhead of Decision on different densities is small. The reason is that, regardless of the number of neighbors,

doi:10.6342/NTU201704349

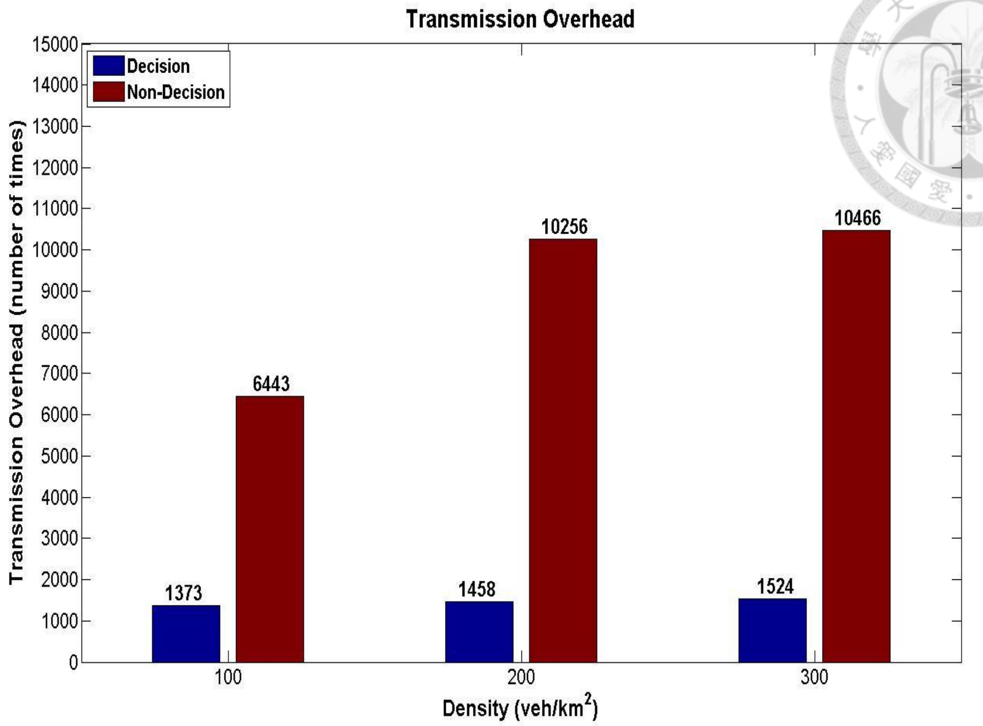


Figure 6.7: The transmission overhead

the breakdown SDC only selects the closest SDC to transfer the file. In the transmission overhead, we are difficult to observe the effect on the wireless network. Therefore, we would focus on the transmission overhead in one unit of time and analyze the load of the channel.

Table 6.3: Non-Decision : Maximum transmission overhead in the unit of time

Density (veh/km^2)	**Average** ($number\ of\ times$)	**Max** ($number\ of\ times$)
100	11	21
200	17	23
300	18	25

For the transmission burst in Table 6.3, we just aimed at the Non-Decision strategy. The reason was that Decision only sends one updating file, at most, in one unit of time, and the impact on the wireless network is low. The data include the average of the maximum number of transmission times in one unit of time, which is approximately one hundred routes with different density, with their maximum being the maximum number of transmission times in one unit of time. When the density was higher, the number of nearby

doi:10.6342/NTU201704349

SDCs was bigger. We would analyze whether these transmission bursts could be loaded by the SCH. In the DSRC, it enables to use four SCHs in parallel.We firstly focus on the default option of alternating access as shown in Figure 6.6(a).Each SCH exists 46 ms to transfer the updating file.In Table 6.1, we know the size of the updating file is 53 KB and the data rate is 27 Mbps, so it would cost 16 ms to transmit a file.In the alternating access, we only can transmit 8 files in the unit of time.Apparently, it is not enough for this system and the minimum average transmission burst is at least 11.We must adjust to another access options such as immediate access which allows channel switching at any time.The remaining time about SI eliminates two GIs is 92 ms thus we have 92 ms to distribute to CCH and SCH.The distribution shows on Figure 6.6(b).Using this distributing way which SCH gets 80 ms and CCH gets 12 ms to use, the system can transfer the larger number of files.It can transmit 20 files in the unit of time.However, if the transmission burst exceeded 20, it had still occurred packet loss.

In summary, for the total time and the collision rate, the performance of this two methods is similar.For the impact on the wireless network, the strategy of Non-Decision is not ideal and the overhead of SCH is very huge.In the default alternating access, it is infeasible.We adjust to immediate access and it can transfer more files in the unit of time.However, when the transmission burst exceeds 20, some files still transfer failure.When different SDC chooses the same SCH to transfer, it may generate interference and if there are other services also need to use SCH, it may not transfer successfully on the busy SCH.Therefore, the strategy of the Non-Decision is applicable for the condition of low density.Although the broadcast burst of the Decision is more than ten times higher than the Non-Decision.The impact on the CCH is very low.So using the strategy of the Decision, the effect on the wireless network is much lower than the Non-Decision and the performance of recovering is similar.

doi:10.6342/NTU201704349

Chapter 7

Conclusion

In this thesis, we investigated the reaction of the SDC after LiDAR malfunction, and discovered that it was currently very poor. Consequently, the ad-vantages of the SDC are being lost. Therefore, we proposed two strategies in order to recover LiDAR breakdown for the SDC. Both strategies apply information transfer from the nearby SDC through V2V Communication. We validated both strategies by simulation and found that they were able to reach our targets. Additionally, we analyzed their effect on the wireless network.

doi:10.6342/NTU201704349

Bibliography

[1] T.Jiang, S.Petrovic, U.Ayyer, A.Tolani, and S.Husain. Self-Driving Cars : Disruptive or Incremental. In *UC Berkeley engineering leadership professional program*, pages 3–22, 2015.

[2] KPMG. Self-Driving Cars : The Next Revolution. pages 1–36, 2012.

[3] Victoria Transport Policy Institute. Autonomous Vehicle Implementation Predictions Implications for Transport Planning. pages 1–24, 2017.

[4] Google Waymo. We are building a safer driver for everyone. 2017. Available at https://waymo.com/.

[5] E.Khalastchi, M.Kalech, and L.Rokach. A hybrid approach for fault detection in autonomous physical agents. In *International Conference on Autonomous agents and multi-agent systems (AAMAS)*, pages 941–948, 2014.

[6] HERE. HERE HD Live Map. 2017. Available at https://here.com/en/products-services/products/here-hd-live-map.

[7] TomTom. TomTom HD map & RoadDNA. 2017. Available at https://automotive.tomtom.com/products-services/hd-map-roaddna.

[8] K.Massow et al. Deriving HD maps for highly automated driving from vehicular probe data. In *IEEE International Conference on Intelligent Transportation Systems*, pages 1745–1752, 2016.

[9] HERE. Vehicle Sensor Data Cloud Ingestion Interface Specification (v2.0.2). 2015. Available at https://lts.cms.here.com/static-cloud-content/Company_Site/2015_06/Vehicle_Sensor_Data_Cloud_Ingestion_Interface_Specification.pdf.

doi:10.6342/NTU201704349

[10] K.Jo, J.Kim, D.Kim, C.Jang, and M.Sunwoo. Development of Autonomous Car—Part I: Distributed System Architecture and Development Process. In *IEEE Transactions on Industrial Electronics*, volume 61, pages 7131–7140, Dec 2014.

[11] RAND Corporation. Autonomous Vehicle Technology - A Guide for Policymakers. pages 1–214, 2016.

[12] Velodyne. velodyne LiDAR. 2017. Available at http://velodynelidar.com.

[13] S.Bauer, Y.Alkhorshid, and G.Wanielik. Using High-Definition maps for precise urban vehicle localization. In *IEEE International Conference on Intelligent Transportation Systems*, pages 492–497, 2016.

[14] M.T.Sun, W.C.Feng, T.H.Lai, K. Yamada, H.Okada, and K.Fujimura. GPS-based message broadcast for adaptive inter-vehicle communications. In *IEEE Vehicular Technology Conference*, pages 2685–2692, 2000.

[15] IEEE Standard for Wireless Access in Vehicular Environments (WAVE) – Networking Services. In *IEEE Std 1609.3-2016*, pages 1–160, April 2016.

doi:10.6342/NTU201704349