Graduate School Form
30 Updated 12/26/2015

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Zhi Huang

Entitled
Vehicle Sensor-based Pedestrian Position Identification in V2V Environment

For the degree of Master of Science in Electrical and Computer Engineering

Is approved by the final examining committee:

Stanley Chien
Co-chair

Yaobin Chen
Co-chair

Lauren A. Christopher

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): Stanley Chien

Approved by: Brian King 11/21/2016

Head of the Departmental Graduate Program Date

VEHICLE SENSOR-BASED PEDESTRIAN POSITION IDENTIFICATION IN V2V ENVIRONMENT

A Thesis

Submitted to the Faculty

of

Purdue University

by

Zhi Huang

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Electrical and Computer Engineering

December 2016

Purdue University

Indianapolis, Indiana

I dedicate my thesis work to my family and my advisors. A special feeling of gratitude to my loving parents, Fubo Huang and Mingxia Sun whose words of encouragement and push for tenacity ring in my ears. My sister Sai Huang and my sister's husband Cong Feng has never left my side and are very special to me. their unique insight always helps me a lot on my research and my life path as well.

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisors Dr. Stanley Yung-Ping Chien and Dr. Yaobin Chen who have supported me throughout the process by the useful comments, remarks and engagement through the learning process of this master thesis. I will always appreciate all they have done, especially Dr. Stanley Yung-Ping Chien for helping me develop my technology skills. Also, many thanks to Dr. Yaobin Chen and Dr. Lauren Christopher for the many hours of proofreading.

PREFACE

This thesis titled Vehicle Sensor-Based Pedestrian Position Identification in V2V Environment is a work as partial fulfillment for the degree of Master of Science in Electrical and Computer Engineering, Purdue University. The research was conducted in Transportation Active Safety Institute from August 2015 to November 2016.

One year ago, I started this project with heartfelt enthusiasm; however this journey turned out to be tougher than I expected. This thesis is the report and the summary of this long process. It expresses my vision of using mathematical and statistical perspective to find the solution of Intelligent Transportation System Problems, and analyzed the computational complexity in the algorithm layer.

Personally, I envolved in Intelligent Vehicle and Transportation System research 2 years ago. When I was a research assistant in University of Michigan, I got to familiar with V2V (Vehicle-to-vehicle) communication and Intelligent System at that moment. I felt excited when I was manipulating the smart devices and learning the algorithms and protocols. During 2 years studying and training, I grasped the important ideas and concepts in vehicle related research, especially when I joined into TASI (Transportation Active Safety Institute) group 1 year ago. I felt really grateful that I could join TASI community and exchange my ideas with my colleagues and professors.

Finally, I hope readers can enjoy this article and find some valuable entry points to do further studies.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

V2V	Vehicle-to-vehicle
AEB	Autonomous Emergency Braking
PAEB	Pedestrian Autonomous Emergency Braking
DSRC	Dedicated Short Range Communication
GPS	Global Positioning System
CCD	Charge-coupled Device
ESR	Electronically Scanning Radar
DBSCAN	Density Based Spatial Clustering of Applications with Noise

ABSTRACT

Huang, Zhi. M.S.E.C.E., Purdue University, December 2016. Vehicle Sensor-Based Pedestrian Position Identification in V2V Environment. Major Professor: Stanley Yung-Ping Chien.

This thesis presents a method to accurately determine the location and amount of pedestrians detected by different vehicles equipped with a Pedestrian Autonomous Emergency Braking (PAEB) system, taking into consideration the inherent inaccuracy of the pedestrian sensing from these vehicles. In the thesis, a mathematical model of the pedestrian information generated by the PAEB system in the V2V network is developed. The Greedy-Medoids clustering algorithm and constrained hierarchical clustering are applied to recognize and reconstruct actual pedestrians, which enables a subject vehicle to approximate the number of the pedestrians and their estimated locations from a larger number of pedestrian alert messages received from many nearby vehicles through the V2V network and the subject vehicle itself. The proposed methods determines the possible number of actual pedestrians by grouping the nearby pedestrians information broadcasted by different vehicles and considers them as one pedestrian. Computer simulations illustrate the effectiveness and applicability of the proposed methods. The results are more integrated and accurate information for vehicle Autonomous Emergency Braking (AEB) systems to make better decisions earlier to avoid crashing into pedestrians.

1. INTRODUCTION

Recently, vehicle sensor-based smart systems have been studied extensively due to the wide spreading of the high-speed internet connection and advanced control theory. For example, autonomous cruise control system, lane departure warning system, blind spot detection system, and even fully autonomous vehicle system are all available in the market to cater customers needs. However, safety is always the priority feature that we need to concern in the first place. This thesis emphasizes the important of the safety and proposed a method to avoid car crash with pedestrians thus save peoples life.

1.1 Definition of V2V-PAEB System

As many automobile companies have announced incorporating Autonomous Emergency Braking (AEB) into their automobiles in the near future, pedestrian recognition systems based on onboard vehicle sensors, such as radar, camera, LiDAR, etc., will be available on more vehicles. If a vehicle can send its sensor detected pedestrian information to nearby vehicles through the Vehicle-to-Vehicle (V2V) communication network, receiving vehicles may be able to use this information as early pedestrian detection and reduce the chance of crashes.

The V2V communication based on DSRC (Dedicated Short Range Communication) technology has been studied extensively in recent years [1]. Many efforts have been made to use this technology to improve road safety. Meanwhile, there have also been developments in Pedestrian Autonomous Emergency Braking (PAEB) technology, which can provide autonomous braking when there is an eminent frontal crash to a vehicle, pedestrian, or bicyclist if the driver fails to apply braking or applies insufficient braking [2], [3]. The PAEB system uses radar, camera, and LiDAR sensors individually or in conjunction with one another to detect the presence and the location of the object in front of the vehicle [4], [5]. For example, the authors in [4] proposed a LiDAR and vision-based approach for pedestrian detection and tracking.

PAEB system performance has been improved significantly in recent years and been offered as an option on many vehicles. It is certain that all vehicles will be equipped with V2V communication capability and PAEB features in the future. However, there will also be a long period of time where vehicles with and without the PAEB and V2V technology will coexist on the road.

If V2V works in conjunction with PAEB, this system is referred to V2V-PAEB system. One of the problems for this system is that when a subject vehicle receives many pedestrian position information messages from other vehicles, it does not know if each pedestrian reported by one vehicle is the same as the pedestrian reported by other vehicles. Therefore, it is necessary to create a method in order to accurately determine the actual amount of pedestrians. The main goal of this thesis is to develop an efficient method for accurately identifying the exact positions and the amount of pedestrians from data provided by multiple vehicles equipped with PAEB systems in the V2V communication network environment.

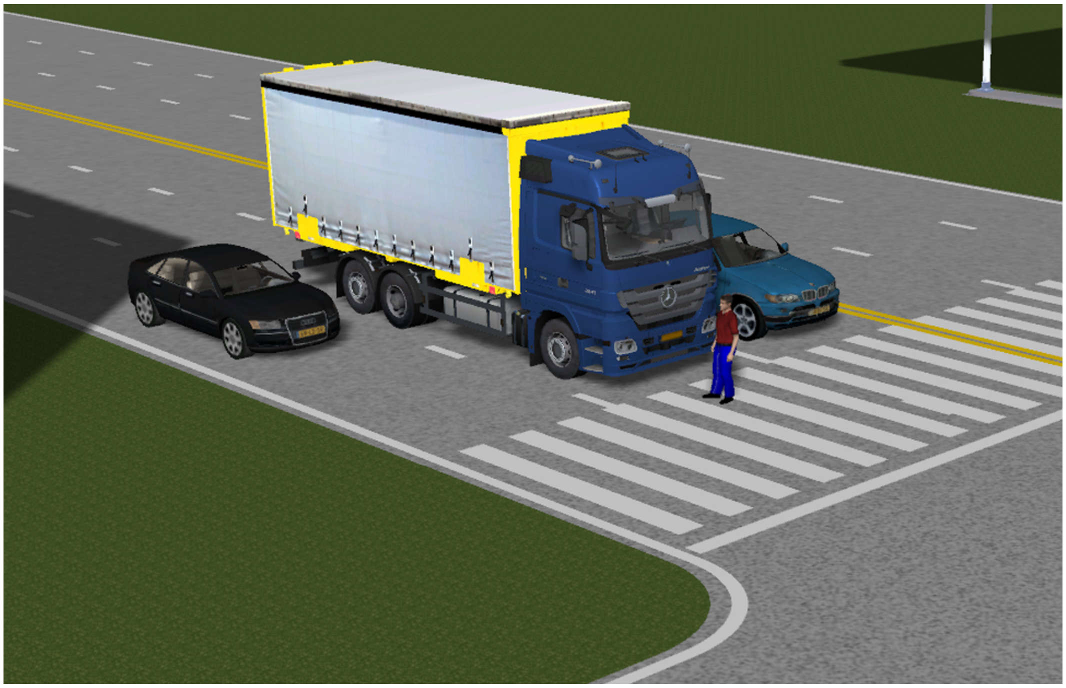


Fig. 1.1.: The truck obscures the right car and the pedestrian.

In this thesis, an efficient method for accurately identifying the exact positions and the number of pedestrians is a clustering problem. A clustering method "Greedy-Medoid" is proposed in chapter 3 and tested in different scenarios.

1.2 V2V-PAEB System Safety Features

There are significant safety benefits when the PAEB system is integrated into V2V communication systems. The benefits can be achieved by empowering every V2V enabled vehicle to make PAEB decisions based on the PAEB sensory data from other nearby vehicles. Figure 1.1 shows a scenario to demonstrate the usefulness of an integrated V2V and PAEB (V2V-PAEB) system. When the black car on the right lane is moving forward and a pedestrian is crossing the street, the pedestrian and the black car cannot see each other since their views are obscured by the truck in the middle lane. It is possible that the black car may collide with the pedestrian since it may be too late for the black car to brake after its PAEB system sees the pedestrian.

In a V2V-PAEB environment, the position and the trajectory of the pedestrian can be detected by the truck and the car on the left lane and can be transmitted through the V2V network to the black car on the right lane long before the black car can see the pedestrian. This enables the black car on the right lane to use the received pedestrian information to make safety decisions earlier.

Figure 1.2 shows an example of V2V-PAEB environment in a busy intersection.

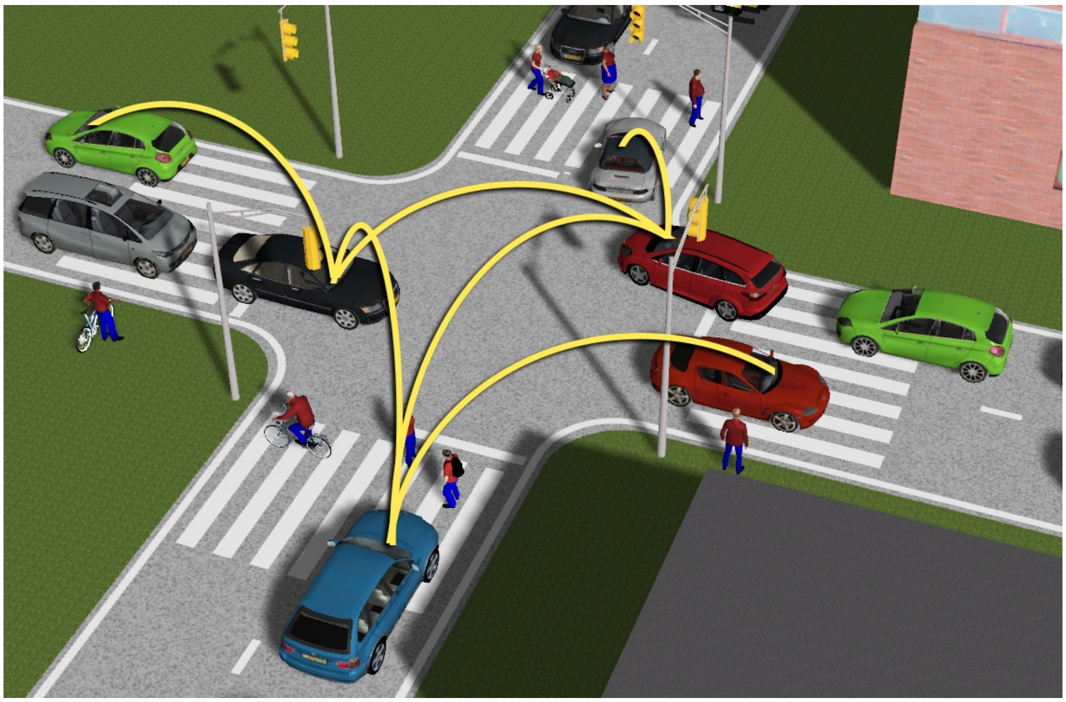


Fig. 1.2.: V2V-PAEB environment in a busy intersection. Curved lines connecting cars represent the V2V communication.

The keys for the successful operation of the collaborated V2V-PAEB are:

(1) To make each vehicle broadcast its own PAEB detected pedestrians information and receive pedestrians information from nearby vehicles through the local V2V network.

(2) To be able to extract accurate location and trajectory information of pedestrians from the V2V messages from many different sources in real-time.

Figure 1.3 shows an information process of the V2V-PAEB system on each vehicle. The goal of this work is to develop an algorithm that enables each V2V enabled vehicle to construct pedestrians locations and trajectory information accurately from the pedestrians information sent from several nearby V2V-PAEB enabled vehicles.

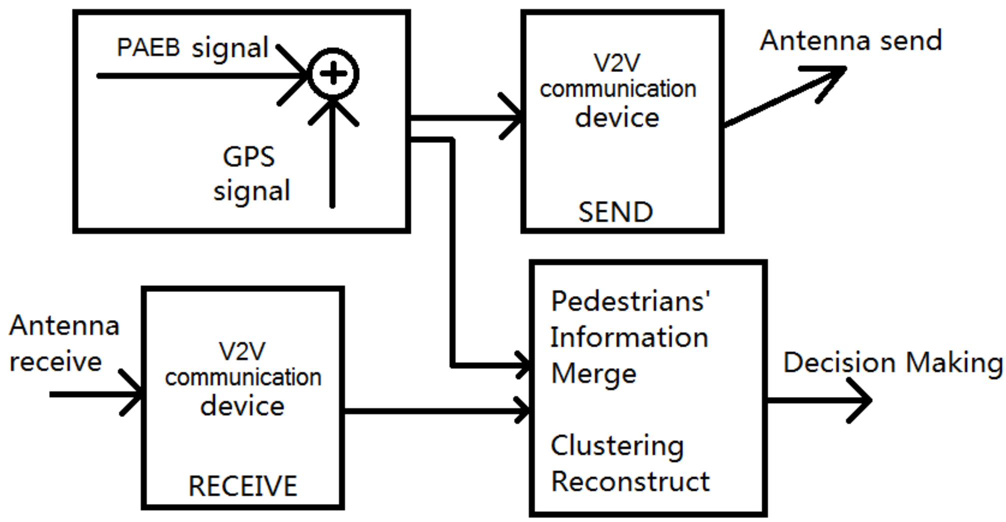


Fig. 1.3.: V2V-PAEB pedestrian safety decision-making process on each vehicle.

Figure 1.3 demonstrates the whole procedure of a single V2V-PAEB enabled vehicle in a single time step. The vehicle constantly uses active sensors to generate PAEB signal, adding with the GPS information about itself location, then send to the V2V network through the antenna or DSRC device. Meanwhile, the vehicle also receives a variety of incoming messages which are containing the similar information as it sends out. By merging its own PAEB data and others PAEB data, the vehicle proceeds the clustering analysis in real-time. The vehicle finally estimate how many actual pedestrians in total and their locations. By finding the locations of pedestrians, the vehicle then make decisions to avoid accident to pedestrians.

1.3 V2V-PAEB Model Structure

The work described in this paper is built on the prior V2V-PAEB research effort described in [6]. As the predecessor of Figure 1.3, Figure 1.4 shows the architecture of the V2V-PAEB system described in [6]. The architecture assumes that V2V enabled vehicles can broadcast their PAEB system detected pedestrians position information as a V2V message, and can receive pedestrians position V2V messages broadcasted from other nearby vehicles. Each vehicle makes safety decisions (warning/braking) by predicting potential collisions based on the pedestrians locations obtained from its own PAEB system and received V2V messages.

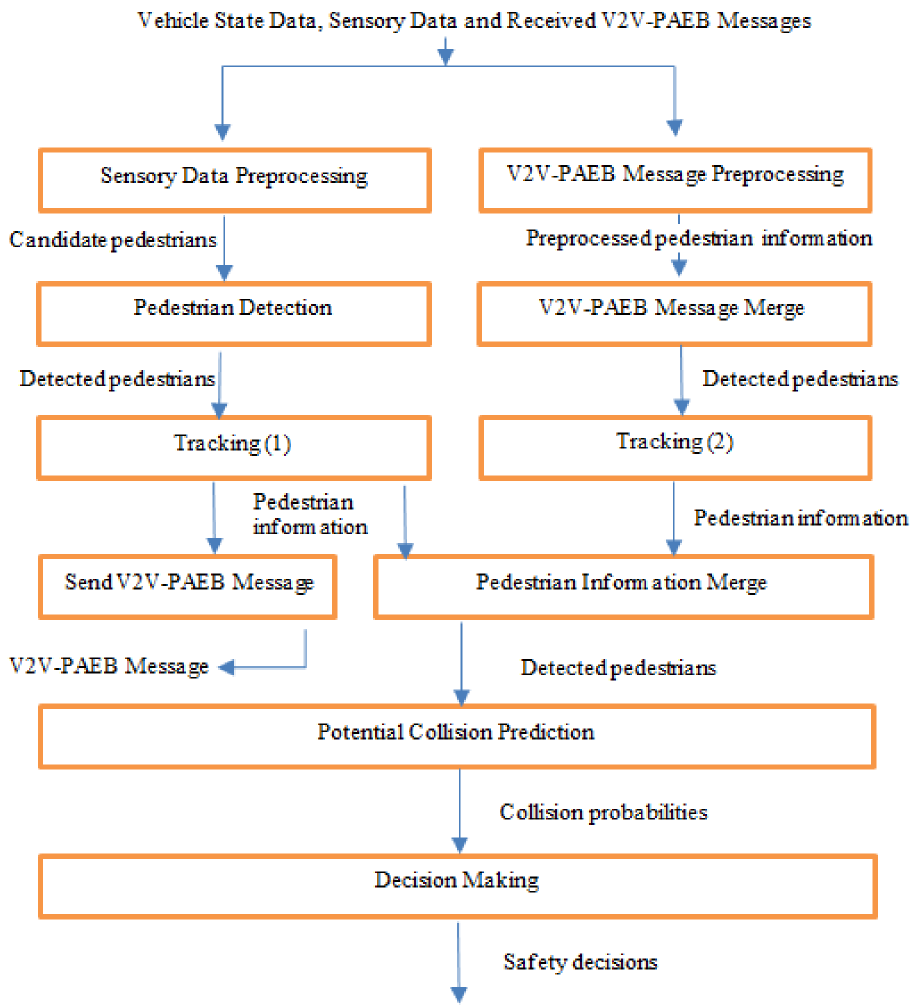


Fig. 1.4.: V2V-PAEB system on each vehicle proposed in [6].

The flowchart in Figure 1.4 shows the necessary subtasks to make the V2V-PAEB system work. Each block in Figure 1.4 represents a specific problem that needs to be addressed in order to make the V2V-PAEB system function properly. One specific block, Pedestrian Information Merge, presents an interesting problem. When n pedestrians and m vehicles are in a small area, each vehicle can potentially see 0 to n pedestrians and can broadcast the pedestrians positions through the V2V network. Due to the errors introduced by the inaccuracy of a vehicles GPS and PAEB sensors, different vehicles may generate different pedestrian locations for the same pedestrian. There is a high possibility that nm pedestrian positions are broadcasted in the V2V network. Assuming that each pedestrian is seen by at least one vehicle, and each vehicle does not necessarily see all pedestrians, how to determine the location of n pedestrians from m V2V messages by m vehicles is a major issue raised but not solved in [6]. This paper describes a method for the block Pedestrian Information merge. The method enables each V2V-PAEB enabled vehicle to construct pedestrians location information accurately from the pedestrians information received from nearby vehicles.

In order to extract real pedestrian information in a large set of PAEB messages in the V2V network, the nature of the errors in the data need to be investigated. Wang, T. et al. [7] described human tracking using Delphi ESR-Vision Fusion in complex environments. They built a radar-vision fusion system utilizing a 77GHz 2D Delphi Electronically Scanning Radar (ESR) and a CCD camera. They described their radar error distribution results. A simple uniform error distribution will be taken into consideration in the experiments of this paper.

Based on our best knowledge, there is no published work on data fusion (reconstructing pedestrians from PAEB information) in a V2V network provided from multiple vehicles. This paper attempts to develop a data fusion (pedestrians signal reconstruction/clustering) algorithm to address this problem. The meaning of this process is trying to compromise the false positive (type I error, false alarm) and false negative (type II error, miss). Since so many pedestrian signals are shared which must be overlapping with each other, the system is trying to merge the pedestrian signals in order to reduce both false positive and false negative of the avoidance decisions.

1.4 Avoidance Decisions

Figure 1.5 shows the avoidance decisions. In Figure 1.5 (a), False negative (type I error, or miss) is introduced to explain the situation while the car is fail to detect an presented pedestrian and thus will lead to an accident. In Figure 1.5 (d), False positive (type II error, or false alarm) is introduced to explain the situation while car is accidentally detect an absent pedestrians and conduct an avoidance decision (either an emergent stop, slow down, or a maneuvering which the subject car is trying to prevent a crash can be treated as an avoidance decision). Figure 1.5 (b) and (c) are both the correct decisions.

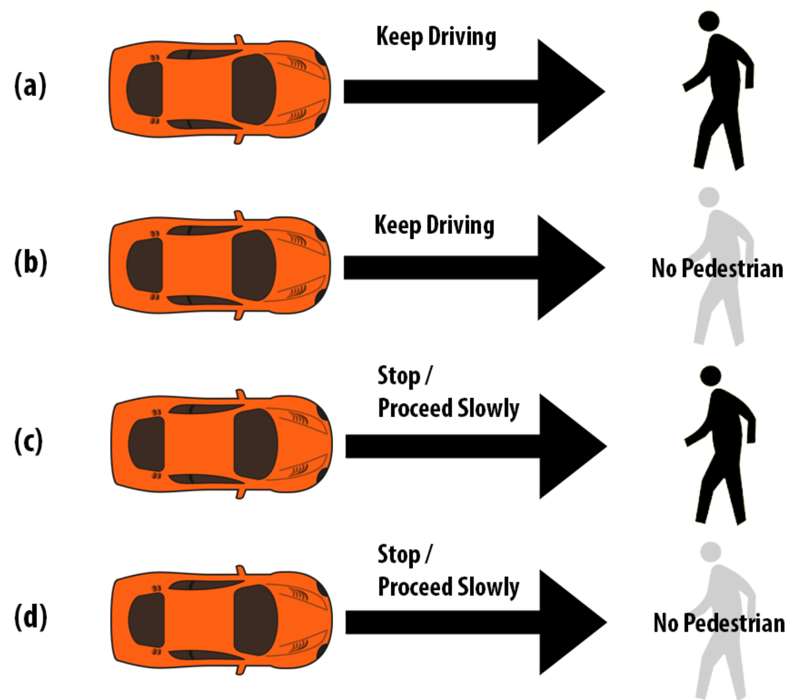


Fig. 1.5.: Avoidance decisions. (a) False negative (type I error, miss), a pedestrian is in the front while response is nothing from the decision machine. (b) Correct decision. Keeping driving without any pedestrian in the front. (c) Correct decision. A pedestrian is presented and been detected by the system, thus proceed an avoidance decision. (d) False positive (type II error, false alarm), no pedestrian in the front while the system still believe there's an pedestrian and thus proceed an avoidance decision.

Another advantage of this system is that it is not required to know how accurate that the pedestrian signals are, comparing with the ground truth original pedestrians' locations. The Greedy-Medoids clustering algorithm (will be discussed later) is trying to gather the nearest pedestrian signals sent from different vehicles as one cluster, then choose the representative data point as the medoid. Thus the distribution of the noisy pedestrian signals are not required as a prior knowledge to the classification (clustering) process.

Finally, the proposed method will be tested in 3 different cases: (1) 2 near pedestrian scenario, (2) dense pedestrians scenario, and (3) sparse pedestrians scenario. All those 3 cases represent different situations when cars meet pedestrians. More scenarios (simulation cases) which are discussed in [8] can also apply on the proposed algorithm. The computational complexity will be studied in order to know if the algorithm is solvable in polynomial time and if it can run in real-time.

This thesis is organized in three parts. Section II describes a mathematical model of pedestrian information broadcasted by various senders. In Section III, an algorithm is proposed to cluster pedestrians information from different vehicles and to find the approximate number of pedestrians. The computational complexity of the algorithm is analyzed. Conclusions are given in Section IV.

2. MATHEMATICAL MODEL OF PEDESTRIANS IN V2V MESSAGE

2.1 Constructing a Local Cartesian Coordinate System

In order to model the pedestrians and show their relative positions for the further study, it is necessary to construct a local Cartesian coordinate system. In that sense, the location of the vehicles and pedestrians can be expressed as the x, y, z values on 3-D Cartesian coordinate, or x, y values on 2-D Cartesian coordinate. The benefit of using local Cartesian coordinate system rather than Global Positioning System (GPS) is because the local Cartesian coordinate system is better to describe a local environment since it is more nature comparing with the GPS uses longitude, latitude, and altitude. However, usually in the intelligent transportation system, GPS data is the only information about location that the vehicle system can derive. Thus to construct a local Cartesian coordinate system, a conversion from GPS geodetic coordinates to local Cartesian coordinates is needed.

The conversion from GPS geodetic coordinates to local Cartesian coordinate system is the conversion between geodetic and ECEF (earth-centered, earth-fixed) coordinates. Following fomulars shows the conversion from geodetic coordinates (latitude ϕ, longitude λ, height h) to ECEF Cartesian coordinate system x, y, and z.

$$\begin{aligned} x &= (N(\phi) + h)\cos\phi\cos\lambda \\ y &= (N(\phi) + h)\cos\phi\sin\lambda \\ z &= \big(N(\phi)(1 - e^2) + h\big)\sin\phi \end{aligned} \tag{2.1}$$

where,

$$N(\phi) = \frac{a}{\sqrt{1 - e^2\sin^2\phi}} \tag{2.2}$$

Where a is the semi-major axis and e is the first numerical eccentricity of the ellipsoid. $N(\phi)$ is the distance from the surface to the z-axis [9].

Given the formular 2.1 and 2.2, it is clear that from the GPS geodetic coordinates we can derive the ECEF Cartesian coordinate, then for a specific scenario, the local Cartesian coordinate system will be constructed.

2.2 The Pedestrians Information in V2V Message

It is assumed that the pedestrian trajectory information detected by a PAEB system can be broadcasted on the V2V network. The information of each pedestrian in a V2V message can be represented by a list of parameters. Let x_x, x_y, and x_z be the coordinates of the vehicle in a local Cartesian coordinate system L. Let x_{PAEB_x}, x_{PAEB_y}, and x_{PAEB_z} be the distances of the PAEB detected pedestrian to the vehicle in x_x, x_y, and x_z directions in L, respectively. The origin of L can be chosen anywhere, but for convenience, x, y, and z axis are assigned to be in parallel with GPS longitude, latitude and altitude respectively, where the zero references of x, y, and z axis point to East, North, and the right hand rule direction, respectively, as depicted in Figure 2.1. According to the property of PAEB sensors (GPS, Radar, camera, etc.), some of the parameters can be considered as random variables with corresponding distribution functions. Table 2.1 shows the example parameters used in this paper.

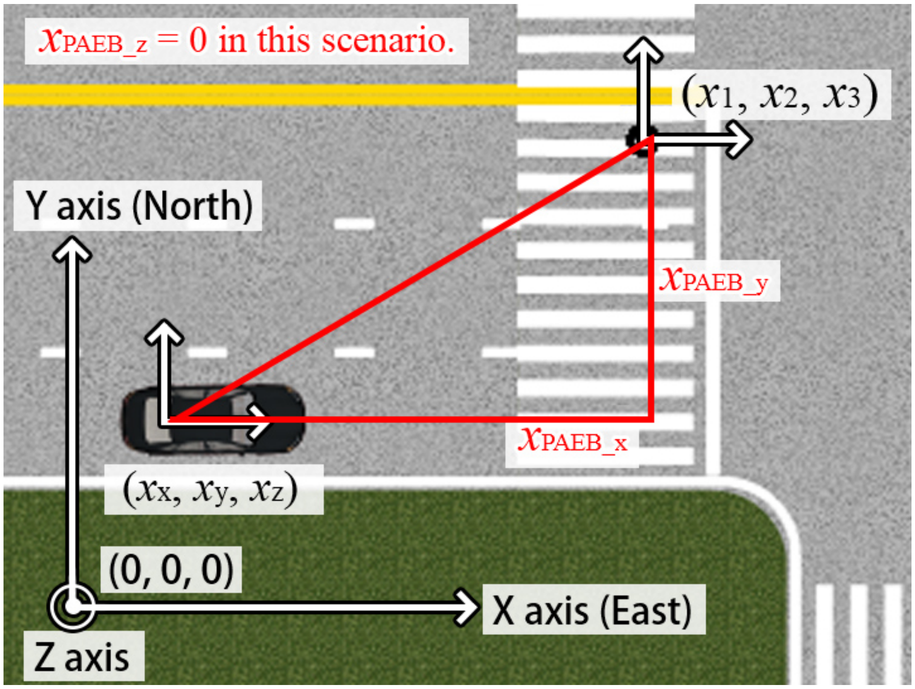


Fig. 2.1.: Relationship between x_x, x_y, x_z, x_{PAEB_x}, x_{PAEB_y}, x_{PAEB_z}, and x_1, x_2, x_3.

x_x, x_y, and x_z can be obtained from vehicles GPS data. x_{PAEB_x}, x_{PAEB_y}, and x_{PAEB_z} can be obtained and calculated by PAEB sensors. Then x_1, x_2, and x_3 can be calculated. Random variables X_1 to X_3 include the errors generated by GPS and PAEB sensors. It is assumed that X_1 to X_3 are the inputs to the proposed work in this paper. There can be many more parameters such as speed, heading, acceleration, etc. To demonstrate the idea, only three input parameters are used in the vector.

The ground truth pedestrian refers to the accurate position and information of a human being. Considering the inherent error introduced in sensor detected pedestrians, ground truth pedestrians can be the benchmark to evaluate the error distribution and quantify the error as well. In this thesis, it is intuitive that vehicles and their sensors do not know the ground truth information, all they can do is using the sensor to “estimate” the ground truth pedestrian. Each ground truth pedestrian can be

Table 2.1: Information of Pedestrians Detected by PAEB

Parameter	**Ground Truth**	**Sensor detected information (with noise)**	**Parameter value**
$Vehicle\ ID$	-	-	$Vehicle\ ID$
$x\ coordinate$	P_1	$X_1 = X_x + X_{PAEB\_x}$	$x_1 = x_x + x_{PAEB\_x}$
$y\ coordinate$	P_2	$X_2 = X_y + X_{PAEB\_y}$	$x_2 = x_y + x_{PAEB\_y}$
$z\ coordinate$	P_3	$X_3 = X_z + X_{PAEB\_z}$	$x_3 = x_y + x_{PAEB\_y}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Ground Truth is the actual information of a pedestrian. Uppercase X represents random variables; lowercase x represents the realization of corresponding random variables.

sensed as 1 or 0 pedestrian (Will address this assumption in chapter 3), which will be the "*pedestrian signal*".

2.3 Mathematical Model of Message Fusion

Let a pedestrians information sent to the V2V network by a car be a *pedestrian signal* (denote as $\boldsymbol{P_{signal}}$, $\boldsymbol{P_{signal}}$ in plural also. $\boldsymbol{P_{signals}}$ is represented by a matrix that contains multiple $\boldsymbol{P_{signal}}$ detected by multiple cars). $\boldsymbol{P_{signal}}$ can be considered as the pedestrian ground truth information, (denoted as $\boldsymbol{P_{input}}$) added to the noise (error) introduced by the PAEB and V2V communication. Note that in a real world situation, such as road testing, $\boldsymbol{P_{input}}$ cannot be obtained. This can be demonstrated in Figure 2.2. It is assumed that cars 2, 3 and 4 use their PAEB sensors to detect a pedestrian and broadcast the pedestrian information through V2V messages; car 1 can receive these V2V messages. However, the three V2V messages received show

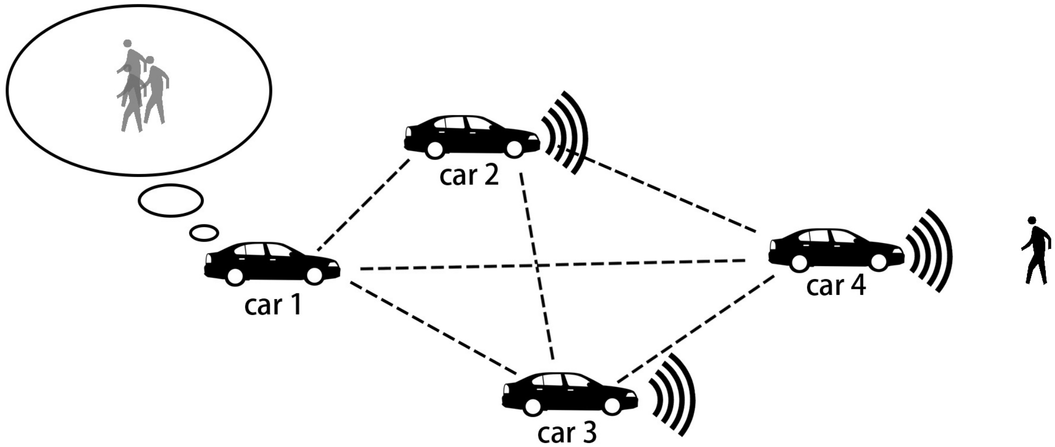


Fig. 2.2.: V2V communication combines with PAEB system.

pedestrians at three different locations. It is not clear to car 1 if there are one pedestrian or three pedestrians.

As the ground truth and pedestrian signal are discussed previously, since the differences of pedestrian description of the same pedestrian are due to the inaccuracy of vehicles GPS and PAEB sensors, we model the sensor errors by noises. Therefore, $\boldsymbol{P_{signal}}$ is the sum of the ground truth pedestrian information and the noise (as depicted in Figure 2.3), which means $\boldsymbol{P_{signal}}$ is not the real attributes of one pedestrian, but is a pedestrian with some attributes (locations, etc.) that vehicle system believes.

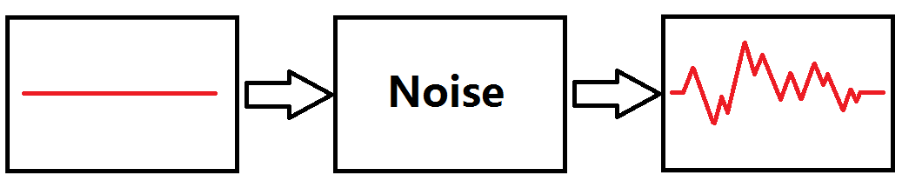


Fig. 2.3.: $\boldsymbol{P_{signal}}$ is a ground truth pedestrians information plus noise.

2.4 Description of $\boldsymbol{P_{signals}}$

Since the goal of the pedestrian message fusion is to extract the estimated pedestrian ground truth information and try to reduce the incorrect information, the resulting$\boldsymbol{P_{output}}$ will be compared to the ground truth, $\boldsymbol{P_{input}}$. $\boldsymbol{P_{input}}$ can be expressed as a vector of pedestrians $\boldsymbol{P_1}$ to $\boldsymbol{P_n}$:

$$\boldsymbol{P_{input}} = \begin{bmatrix} \boldsymbol{P_1} & \boldsymbol{P_2} & \cdots & \boldsymbol{P_n} \end{bmatrix} \tag{2.3}$$

Each pedestrian $\boldsymbol{P_i}$, where $i = 1, 2, \ldots, n$, is also a column vector which contains several variables such as P_1, P_2, P_3 in Table 2.1, and is shown in Equation 2.4. For example, $P_{i,1}$ represents the ground truth variable P_1 for the i^{th} pedestrian.

$$\boldsymbol{P_i} = \begin{bmatrix} P_{i,1} \\ P_{i,2} \\ P_{i,3} \end{bmatrix} \tag{2.4}$$

Due to differences in sensor types and different sensors used on different vehicles, different noises are applied to the ground truth vector $\boldsymbol{P_i}$ generating different $\boldsymbol{P_{signal}}$ by different vehicle makes and models. Let $\boldsymbol{f}\,(\cdot)$ represent the noise and *vehicle ID* injection process, a parameter in each $\boldsymbol{P_{signal}}$ can be described as in Equation 2.5,

$$\boldsymbol{Q_{n,m}} = \begin{bmatrix} vehicle\ ID \\ x_{n,m,1} \\ x_{n,m,2} \\ x_{n,m,3} \end{bmatrix} = f\left(\boldsymbol{P_i}\right) = f\left(\begin{bmatrix} P_{i,1} & P_{i,2} & P_{i,3} \end{bmatrix}^T\right) \tag{2.5}$$

where $\boldsymbol{Q_{n,m}}$ represents the n^{th} pedestrian detected by vehicle m, which is a "$\boldsymbol{P_{signal}}$". The process $\boldsymbol{f}\,(\cdot)$ can either be the error process from PAEB hardware in the real world or be the error process generated by computer simulations.

Elements in $\boldsymbol{Q_{n,m}}$ have the same meaning as $\boldsymbol{P_i}$ in Equation 2.4, except a *vehicle ID* inserted into the first row in $\boldsymbol{Q_{n,m}}$ in order to distinguish which vehicle generates $\boldsymbol{P_i}$. Each $\boldsymbol{P_{signal}}$ (i.e. $\boldsymbol{Q_{n,m}}$) has a unique random vector to describe its ground truth values with added noises:

$$\overrightarrow{X}_{n,m} = \begin{bmatrix} X_{n,m,1} & X_{n,m,2} & X_{n,m,3} \end{bmatrix}^T \tag{2.6}$$

$x_{n,m,1}$ to $x_{n,m,3}$ in Equation 2.5 are the realizations of random variables $X_{n,m,1}$ to $X_{n,m,3}$, as shown in Equation 2.6 with respect to the n^th pedestrian detected by vehicle m.

Thus, $\boldsymbol{P_{signals}}$ that contains all pedestrian messages from all vehicles can be expressed as,

$$\boldsymbol{P_{signals}} = \boldsymbol{F}\left(\boldsymbol{P_{input}}\right) = \boldsymbol{F}\left(\begin{bmatrix} \boldsymbol{P_1} & \boldsymbol{P_2} & \cdots & \boldsymbol{P_n} \end{bmatrix}\right) = \begin{bmatrix} \boldsymbol{Q_{1,1}} & \boldsymbol{Q_{1,2}} & \cdots & \boldsymbol{Q_{1,m}} \\ \boldsymbol{Q_{2,1}} & \boldsymbol{Q_{2,2}} & \cdots & \boldsymbol{Q_{2,m}} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{Q_{n,1}} & \boldsymbol{Q_{n,2}} & \cdots & \boldsymbol{Q_{n,m}} \end{bmatrix} \tag{2.7}$$

where $\boldsymbol{F}(\cdot)$ contains multiple processes of $f(\cdot)$ in Equation 2.5. $\boldsymbol{P_{signals}}$ is a matrix that contains all $\boldsymbol{P_{signal}}$ which are detected by m vehicles simultaneously (in the same V2V message processing timestamp). Inside the matrix, each column represents a V2V message containing multiple $\boldsymbol{P_{signal}}$ detected by one vehicle. For example, $\boldsymbol{Q_{2,4}}$ represents the 2nd $\boldsymbol{P_{signal}}$ detected by 4th vehicle (each vehicle has a unique ID).

Note that even though $\boldsymbol{Q_{1,1}}$ and $\boldsymbol{Q_{1,2}}$ have the same first subscript "1", it does not mean that they are the same pedestrian. They represent the first pedestrian detected by each car.

Different cars may detect different numbers of pedestrians. The number of rows n is the maximum number of pedestrians detected by one car. If another car detects less than n pedestrians, say r pedestrians, its $r+1, r+2, \ldots, n$ elements will be substituted by 0 and will not be considered in the subsequent calculations. For example, assuming that there are five ground truth pedestrians. If car 1 can only detect four $\boldsymbol{P_{signal}}$ and car 2 can only detect three $\boldsymbol{P_{signal}}$, both car 3 and car 4 can detect five $\boldsymbol{P_{signal}}$, then the matrix can be represented by,

$$\boldsymbol{P_{signals}} = \begin{bmatrix} \boldsymbol{Q_{1,1}} & \boldsymbol{Q_{1,2}} & \boldsymbol{Q_{1,3}} & \boldsymbol{Q_{1,4}} \\ \boldsymbol{Q_{2,1}} & \boldsymbol{Q_{2,2}} & \boldsymbol{Q_{2,3}} & \boldsymbol{Q_{2,4}} \\ \boldsymbol{Q_{3,1}} & \boldsymbol{Q_{3,2}} & \boldsymbol{Q_{3,3}} & \boldsymbol{Q_{3,4}} \\ \boldsymbol{Q_{4,1}} & 0 & \boldsymbol{Q_{4,3}} & \boldsymbol{Q_{4,4}} \\ 0 & 0 & \boldsymbol{Q_{5,3}} & \boldsymbol{Q_{5,4}} \end{bmatrix} \tag{2.8}$$

2.5 Clustering Part

The clustering part is to find the actual number of pedestrians with estimated locations (see dashed box in Figure 2.4). This part can also refer to pedestrians' information merge part shown in both Figure 1.3 and Figure 1.4.

Clustering analysis is a task to group many data points into several classes (clusters) [10]. Each cluster has a clustering center. The input of the clustering part is $\boldsymbol{P_{signals}}$ and the output is estimated pedestrians $\boldsymbol{P_{output}}$. The goal is to generate the estimated pedestrians ($\boldsymbol{P_{output}}$) from $\boldsymbol{P_{signals}}$ to match $\boldsymbol{P_{input}}$. Clustering is the process $\boldsymbol{g(\cdot)}$ to convert $\boldsymbol{P_{signals}}$ to $\boldsymbol{P_{output}}$:

$$\boldsymbol{P_{output}} = g\left(\boldsymbol{P_{signals}}\right) = \begin{bmatrix} \boldsymbol{P'_1} & \boldsymbol{P'_2} & \cdots & \boldsymbol{P'_j} \end{bmatrix} \tag{2.9}$$

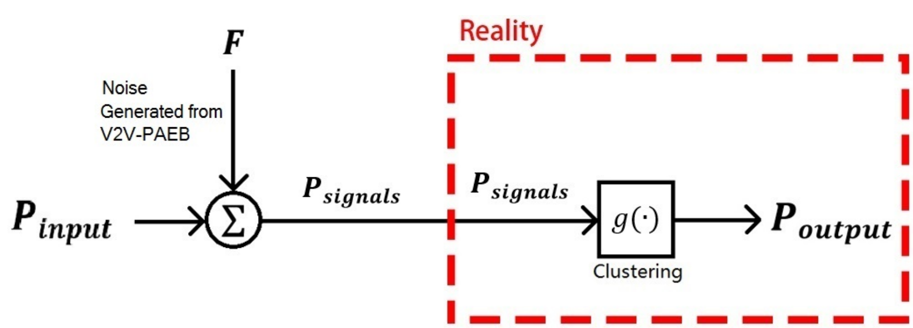


Fig. 2.4.: Mathematical model of the fusion (clustering) of the pedestrian information received from multiple vehicles.

As shown in Figure 2.4, the significance of the clustering process is trying to refine the original pedestrians, by providing a bunch of $\boldsymbol{P_{signal}}$, thus estimates the more accurate positions of pedestrians, reduces both false positive and false negative, helps vehicles make better decisions as discribed in Figure 1.5 (b), (c).

3. GREEDY-MEDOIDS CLUSTERING ALGORITHMS

3.1 Determine the Number of the Clusters

An example scenario at a road intersection is shown in Figure 3.1. There are four cars (A, B, C and D) and five pedestrians (pedestrians 1, 2, 3, 4, and 5). Suppose in a time step, car A can detect pedestrians 2, 3, 4, 5, car B can detect pedestrians 1, 2, 3, car C can detect pedestrians 1, 4, 5, and car D can detect pedestrian 2, 3, 4. The matrix $\boldsymbol{P_{signals}}$ is

$$\boldsymbol{P_{signals}} = \begin{bmatrix} \boldsymbol{Q_{1,1}} & \boldsymbol{Q_{1,2}} & \boldsymbol{Q_{1,3}} & \boldsymbol{Q_{1,4}} \\ \boldsymbol{Q_{2,1}} & \boldsymbol{Q_{2,2}} & \boldsymbol{Q_{2,3}} & \boldsymbol{Q_{2,4}} \\ \boldsymbol{Q_{3,1}} & \boldsymbol{Q_{3,2}} & \boldsymbol{Q_{3,3}} & \boldsymbol{Q_{3,4}} \\ \boldsymbol{Q_{4,1}} & 0 & 0 & 0 \end{bmatrix} \tag{3.1}$$

Vehicle ID 1, 2, 3, 4 are used to represent $\boldsymbol{P_{signals}}$ detected by cars A, B, C, and D in Equation 3.1, respectively.

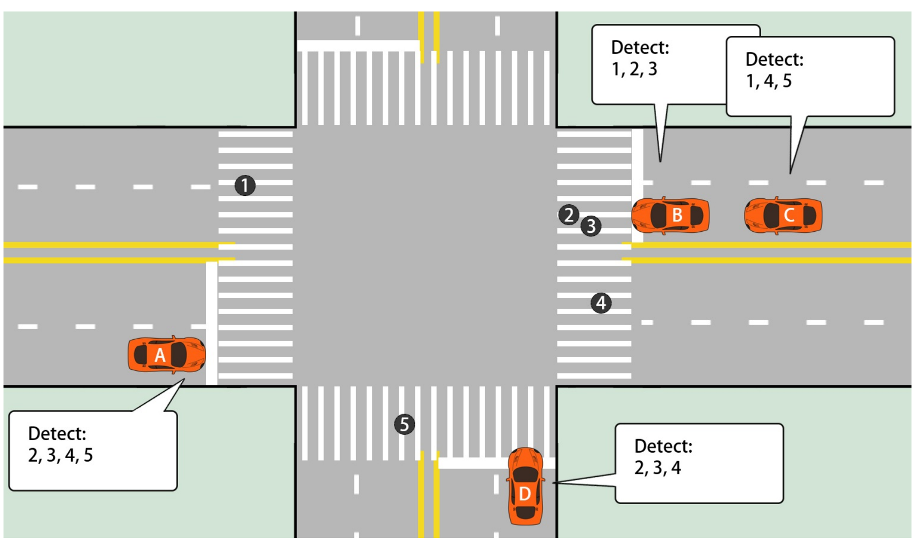


Fig. 3.1.: A road intersection scenario.

Figure 3.2 shows the locations of pedestrians based on $\boldsymbol{P_{signals}}$ in Equation 3.1. Each $\boldsymbol{P_{signal}}$ is marked by the ID of the car that generates it. Table 3.1 shows the exact locations of all $\boldsymbol{P_{signal}}$ in a local Cartesian coordinate system shown in Figure 3.2.

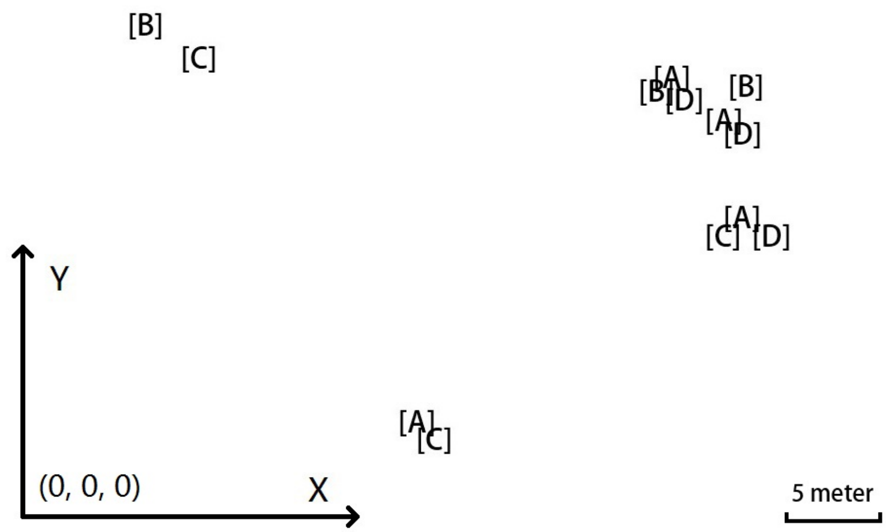


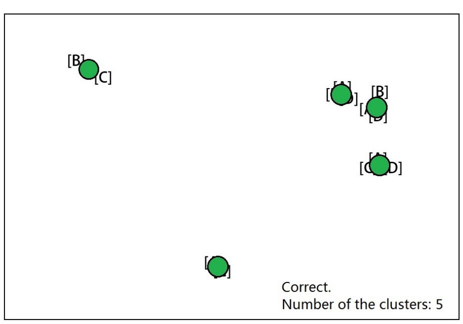
Fig. 3.2.: A road intersection scenario.

Suppose car C receives $\boldsymbol{P_{signals}}$ that contains the information from other cars, without knowing actually how many pedestrians are on the road. To reconstruct pedestrians ($\boldsymbol{P_{output}}$) based on Figure 3.2, car C needs to generate $\boldsymbol{P_{output}}$ using the clustering analysis to split $\boldsymbol{P_{signals}}$ into several clusters, and then find the center of each cluster. Then the set of these clustering centers becomes $\boldsymbol{P_{output}}$. The correct and incorrect clustering processes with five and four clustering centers are shown in Figure 3.3(a) and Figure 3.3(b), respectively.

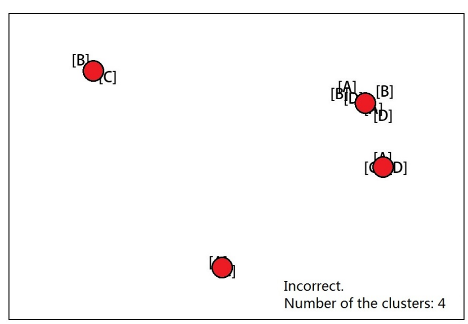
Table 3.1: 5 Pedestrians' Parameter

$\boldsymbol{P_{signal}}$ **ID**	**Detected by which vehicle ($Vehicle\ ID$)**	**Location (x, y, z) (center of the pedestrian) (meter)**
1	A	(33.719, 23, 0)
2	A	(36.469, 20.688, 0)
3	A	(37.5, 15.625, 0)
4	A	(20.469, 4.875, 0)
5	B	(6.406, 25.75, 0)
6	B	(32.906, 22.188, 0)
7	B	(37.656, 22.438, 0)
8	C	(9.219, 24, 0)
9	C	(36.469, 14.688, 0)
10	C	(21.469, 3.906, 0)
11	D	(34.406, 21.75, 0)
12	D	(37.5, 20, 0)
13	D	(39, 14.688, 0)

$\boldsymbol{P_{signal}}$ ID is used for distinguishing each $\boldsymbol{P_{signal}}$ in this example.



(a) Correct clustering process.



(b) Incorrect clustering process.

Fig. 3.3.: Correct (a) and incorrect (b) clustering results.

Since pedestrian 2 is close to pedestrian 3 (could be the same walking speed and same direction/heading), it is reasonable that the clustering algorithm classifies them into one class. However, it will fail to identify a pedestrian. In statistical hypothesis testing, it is a false negative error [11].

To prevent false negative errors, additional information $Vehicle\ ID$ is used in the clustering process. Assuming that one pedestrian cannot be detected by a PAEB sensor as two pedestrians, assumption 1 is adopted (see dashed oval area in Figure 3.4):

(Assumption 1) If there exist two $\boldsymbol{P_{signal}}$ with the same sender ID ($Vehicle\ ID$), they cannot be clustered into the same cluster (classified as one pedestrian).

Based on Assumption 1, $\boldsymbol{P_{signals}}$ in the dashed oval area in Figure 3.4 cannot be in one cluster. This assumption can be applied in clustering process to classify $\boldsymbol{P_{signals}}$ better since the $Vehicle\ ID$ information is available and useful. The Greedy-Medoids clustering algorithm described in next section uses this assumption.

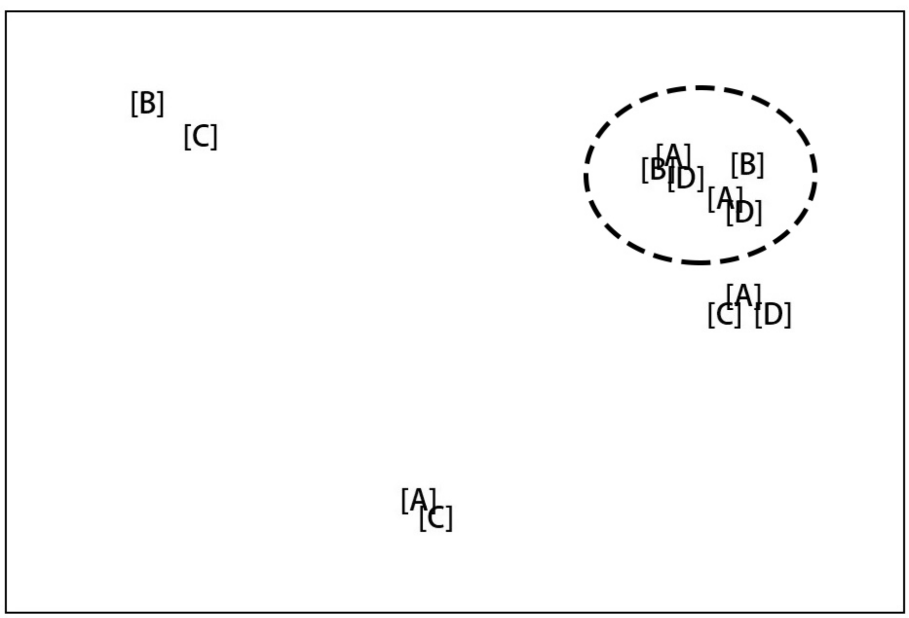


Fig. 3.4.: Two pedestrians have several similar $\boldsymbol{P_{signal}}$ detected by cars.

3.2 Greedy-Medoids Clustering Algorithm Approach

To cluster $\boldsymbol{P_{signals}}$ and to reconstruct pedestrians from multiple vehicles' V2V messages, typical clustering algorithms such as K-Means and K-Medoids [12], [13] cannot be applied since they require the number of the clusters in advance. In V2V-PAEB scenarios, the number of pedestrians is not known. Even density-based clustering methods, for example, DBSCAN (Density Based Spatial Clustering of Applications with Noise), such as [14], [15] yield good approaches to discover the number of clusters, however, they do not make use of the important knowledge of Vehicle ID, which is not applicable in the V2V-PAEB pedestrian information merging problem. These clustering methods provide useful information for Greedy-Medoid clustering algorithm.

With reference to statistical hypothesis testing [11], say a type I error is an absent pedestrian while the response is existed. A type II error is an existed pedestrian while the response is absent (see Figure 3.5).

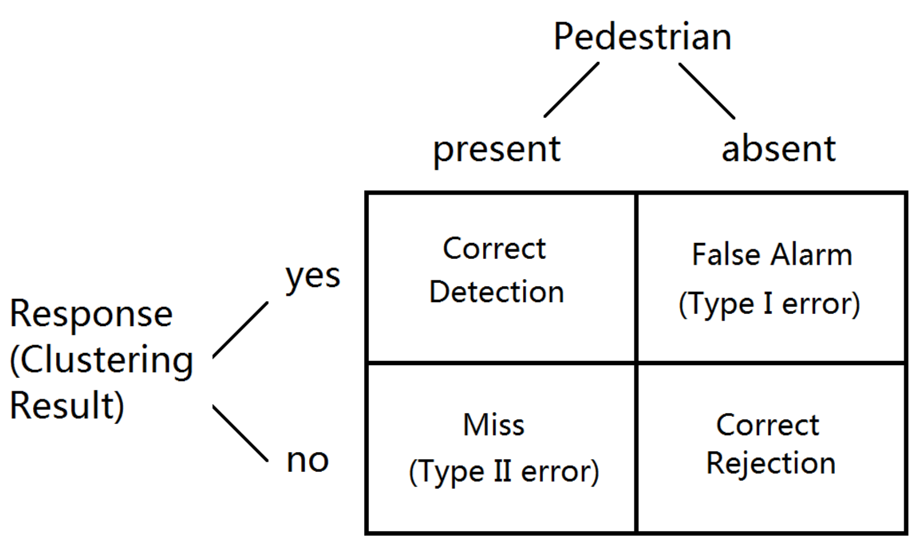


Fig. 3.5.: Statistical hypothesis testing applies on PAEB detection.

Type I and type II errors are also called "false positive" and "false negative", respectively. Decreasing both type I error (false positive) and type II error (false negative) is desirable in the clustering process. However, they cannot be decreased at the same time. To make a compromise, a threshold parameter will be used to balance the type I and type II errors. Based on this approach, we can make the following assumption:

(Assumption 2) If a $\boldsymbol{P_{signal}}$ is far from any other $\boldsymbol{P_{signal}}$, then it can be considered as in another cluster.

Therefore, a distance function $dist(o, p)$ needs to be defined to measure the distance between two $\boldsymbol{P_{signal}}$ o and p. To quantify the distance between two $\boldsymbol{P_{signal}}$, we use Euclidean distance since it is widely used in [12]. Note that $vehicle\ ID$ is not used in distance calculation.

A medoid is defined as a representative $\boldsymbol{P_{signal}}$ of a cluster. Following are the detailed steps of Greedy-Medoids clustering algorithm for determining the number of the clusters and clustering centers.

Algorithm 1 GREEDY-MEDOIDS CLUSTERING

1: $\boldsymbol{PrimeVehicle, MedoidList} = INITIALIZATION(\boldsymbol{D_{Threshold}})$
2: $\boldsymbol{MedoidList} = ASSIGNMENT(\boldsymbol{PrimeVehicle, MedoidList})$
3: $\boldsymbol{MedoidList} = UPDATING(\boldsymbol{MedoidList})$
4: $\boldsymbol{MedoidList} = ASSIGNMENT(\boldsymbol{0, MedoidList})$
5: **return** $\boldsymbol{MedoidList}$

Algorithm 2 INITIALIZATION($\boldsymbol{D_{Threshold}}$)

1: Find a vehicle (call it the $\boldsymbol{PrimeVehicle}$) which sends the most number of $\boldsymbol{P_{signal}}$.
2: Define all $\boldsymbol{P_{signal}}$ detected by the $\boldsymbol{PrimeVehicle}$ as initial medoids and store them into a list called $\boldsymbol{MedoidList}$.

Discussion: The first ASSIGNMENT uses the following greedy strategy in line 21-23:

$$dist(\boldsymbol{P_{current}}, \boldsymbol{M}) \underset{D_1}{\overset{D_0}{\gtrless}} dist(\boldsymbol{S}, \boldsymbol{M}) \tag{3.2}$$

$\boldsymbol{D_0}$: Keep $\boldsymbol{S}$ and reject $\boldsymbol{P_{current}}$.

$\boldsymbol{D_1}$: Reject $\boldsymbol{S}$ and add $\boldsymbol{P_{current}}$.

Algorithm 3 ASSIGNMENT($\boldsymbol{PrimeVehicle}, \boldsymbol{MedoidList}$)

Require: $\boldsymbol{D_{Threshold}}$

1: **for** each vehicle $\boldsymbol{V}$ except the $\boldsymbol{PrimeVehicle}$ **do**
2: Initialize an empty list $\boldsymbol{WaitingList}$.
3: Append all $\boldsymbol{P_{signal}}$ detected by $\boldsymbol{V}$ into $\boldsymbol{WaitingList}$.
4: **while** $\boldsymbol{WaitingList}$ is not empty **do**
5: Initialize an empty list $\boldsymbol{MatchedList}$.
6: Delist a $\boldsymbol{P_{signal}}$ from $\boldsymbol{WaitingList}$ and call it $\boldsymbol{P_{current}}$.
7: **for** each medoids $\boldsymbol{M}$ from $\boldsymbol{MedoidList}$ **do**
8: **if** $dist(\boldsymbol{P_{current}}, \boldsymbol{M}) \leq \boldsymbol{D_{Threshold}}$ **then**
9: Append $\boldsymbol{M}$ into $\boldsymbol{MatchedList}$.
10: **end if**
11: **end for**
12: **if** $\boldsymbol{MatchedList}$ is empty **then**
13: Append $\boldsymbol{P_{current}}$ into $\boldsymbol{MedoidList}$. ▷ Set $\boldsymbol{P_{current}}$ as a new medoid.
14: **else**
15: ▷ $\boldsymbol{MatchedList}$ is not empty.
16: **for** each $\boldsymbol{M}$ from sorted $\boldsymbol{MatchedList}$ **do**
17: **if** the cluster having a medoid $\boldsymbol{M}$ (call it M cluster) does not contain a $\boldsymbol{P_{signal}}$ (call it $\boldsymbol{S}$) sent by vehicle $\boldsymbol{V}$ **then**
18: Assign $\boldsymbol{P_{current}}$ to the $\boldsymbol{M}$ cluster.
19: **goto** line 4.
20: **else**
21: **if** $dist(\boldsymbol{P_{current}}, \boldsymbol{M}) < dist(\boldsymbol{S}, \boldsymbol{M})$ **then**
22: Assign $\boldsymbol{P_{current}}$ to $\boldsymbol{M}$ cluster.
23: Move $\boldsymbol{S}$ from $\boldsymbol{M}$ cluster to $\boldsymbol{WaitingList}$.
24: **goto** line 4.
25: **else**
26: **goto** line 16.
27: **end if**
28: **end if**
29: **end for**
30: **end if**
31: **if** $\boldsymbol{P_{current}}$ still remains unassigned status **then**
32: Append $\boldsymbol{P_{current}}$ into $\boldsymbol{MedoidList}$. ▷ Set $\boldsymbol{P_{current}}$ as a new medoid.
33: **end if**
34: **end while**
35: **end for**

By applying this strategy, the algorithm will traverse all vehicles' messages except the message from $\boldsymbol{PrimeVehicle}$. The algorithm stops when all $\boldsymbol{P_{signal}}$ are assigned to clusters.

There is an issue with this strategy. Suppose after ASSIGNMENT, a cluster is shown as in Figure 3.6 where the medoid is circle 1 that comes from the $\boldsymbol{PrimeVehicle}$ and circles 2 to 6 are assigned to circle 1. It appears that choosing circle 3 as the medoid is much better since it is closely surrounded by other $\boldsymbol{P_{signal}}$ while circle 1 is more likely a noise.

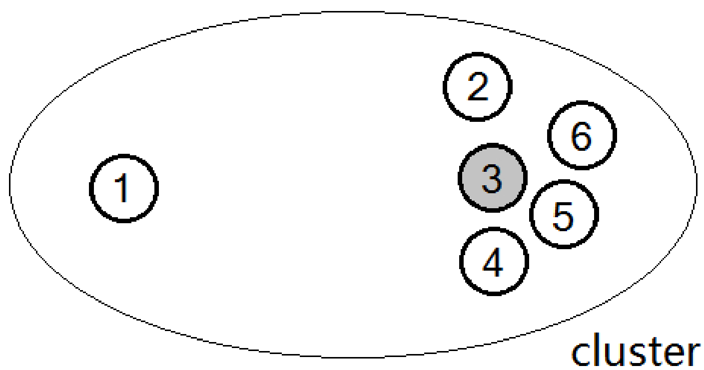


Fig. 3.6.: Different choose of medoid in a cluster.

Remember the reason for choosing circle 1 as the medoid was because circle 1 comes from the $\boldsymbol{PrimeVehicle}$. If circle 3 was initially observed by the $\boldsymbol{PrimeVehicle}$, circle 3 would be the medoid. To eliminate this randomness, UPDATING and the second ASSIGNMENT are added.

Discussion: UPDATING finds a better initial medoid set with respect to each cluster, which has minimum total distance error in the cluster. This step also eliminates the randomness of the observer $\boldsymbol{PrimeVehicle}$ in INITIALIZATION, which strongly affects the result ($\boldsymbol{MedoidList}$) in the first ASSIGNMENT.

Figure 3.7 shows the flow chart of the ASSIGNMENT step.

Algorithm 4 UPDATING($\boldsymbol{MedoidList}$)

Initialize an empty list $\boldsymbol{BetterMedoidList}$.
for each cluster $\boldsymbol{C}$ **do**
  $\boldsymbol{MinimumError} \leftarrow +\infty$
  for each $\boldsymbol{P_{signal}}$ $\boldsymbol{p}$ from cluster $\boldsymbol{C}$ **do**
    $\boldsymbol{error} \leftarrow \sum_{\boldsymbol{o} \in cluster\boldsymbol{C}} dist(\boldsymbol{o}, \boldsymbol{p})$ ▷ $\boldsymbol{o}$ represents all other $\boldsymbol{P_{signal}}$ from cluster $\boldsymbol{C}$.
    if If $\boldsymbol{error} < \boldsymbol{MinimumError}$ **then**
      $\boldsymbol{MinimumError} \leftarrow \boldsymbol{error}$
      $\boldsymbol{BetterMedoid} \leftarrow \boldsymbol{p}$
    end if
  end for
  Append $\boldsymbol{BetterMedoid}$ into $\boldsymbol{BetterMedoidList}$.
end for
$\boldsymbol{MedoidList} \leftarrow \boldsymbol{BetterMedoidList}$

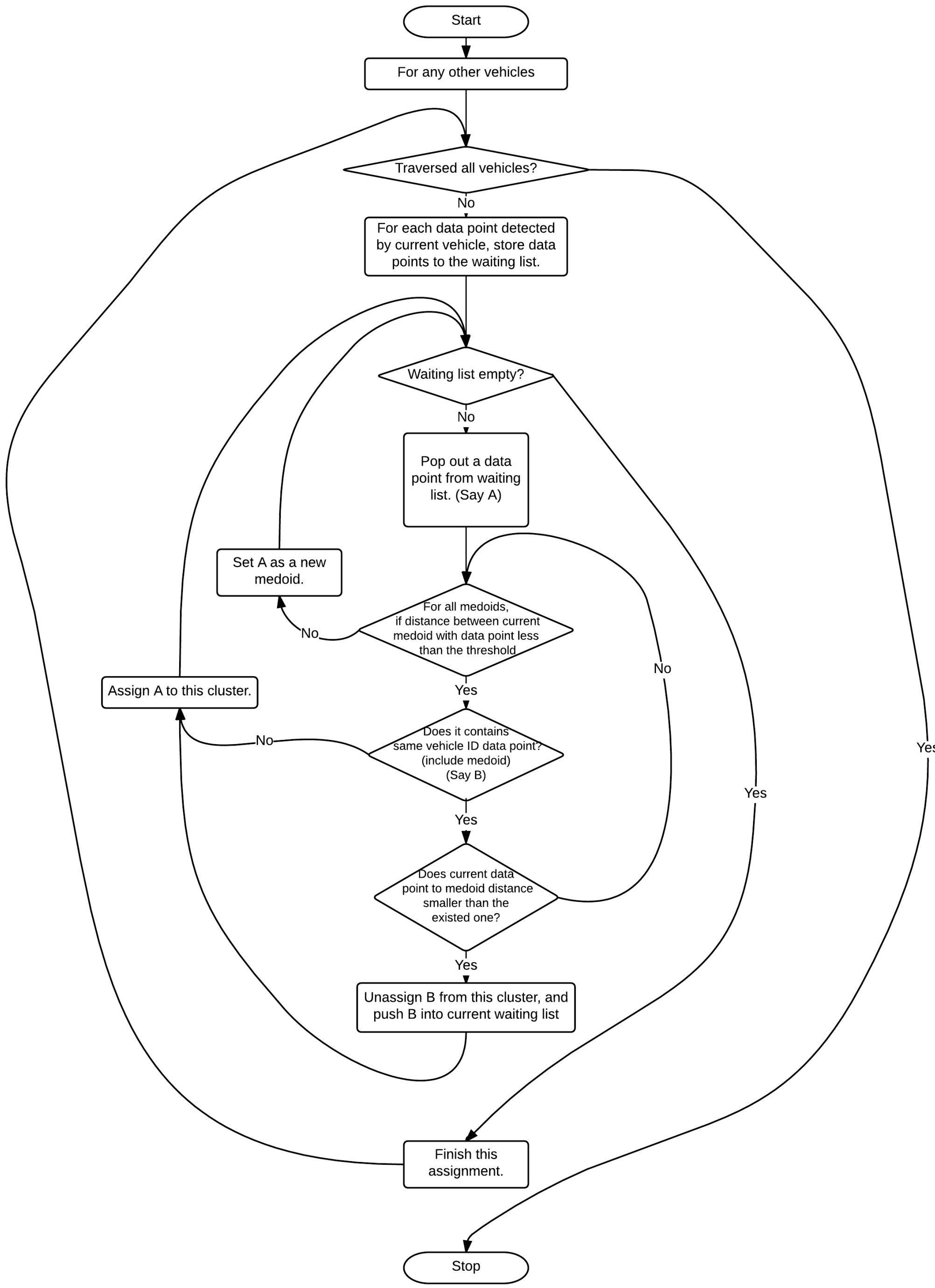


Fig. 3.7.: Flowchart of the ASSIGNMENT step.

The second ASSIGNMENT with a better set of initial medoids in $\boldsymbol{MedoidList}$ generated in UPDATING without any $\boldsymbol{PrimeVehicle}$.

Based on above steps, the approximate number of pedestrians is reconstructed ($\boldsymbol{P_{output}} = \boldsymbol{MedoidList}$) based on the pedestrian information provided by near-by vehicles through the V2V network. The number of pedestrians in $\boldsymbol{P_{output}}$ and $\boldsymbol{P_{input}}$ can be interpreted as follows:

If the value $\boldsymbol{D_{Threshold}}$ is fairly small, as well as a small noise for $\boldsymbol{P_{signal}}$ generation, then it is very likely that $\boldsymbol{P_{output}} = \boldsymbol{P_{input}}$.

If the value $\boldsymbol{D_{Threshold}}$ is fairly small, but the random noise for $\boldsymbol{P_{signal}}$ generation is large, then it is very likely that $\boldsymbol{P_{output}} > \boldsymbol{P_{input}}$. It means a ground truth pedestrian is considered as more than one pedestrian due to large differences in $\boldsymbol{P_{signals}}$ of the same ground truth pedestrian.

If the value $\boldsymbol{D_{Threshold}}$ is large, then it is very likely that $\boldsymbol{P_{output}} < \boldsymbol{P_{input}}$. It means more than one ground truth pedestrians are considered as one pedestrian.

3.3 Correctness of the Greedy-Medoids Clustering Algorithm

To show the correctness of the Greedy-Medoids clustering algorithm, it is necessary to prove that the target vehicle is able to make correct decision based on clustering result.

Note that the goal of the Greedy-Medoids clustering algorithm is to filter out the noise by refining a small group of representative data points (medoids). The key point is, after filtering, the representative data points are very close to the ground truth.

In ASSIGNMENT, a greedy approach is used to assign $\boldsymbol{P_{signal}}$ to the nearest medoid. Greedy algorithms do not always yield optimal solutions [16], but they are efficient in a real-time communication (V2V communication) system, since each time line 12-32 in ASSIGNMENT either assign a $\boldsymbol{P_{signal}}$ to the nearest acceptable medoid, or let this $\boldsymbol{P_{signal}}$ become a new medoid. Eventually, all $\boldsymbol{P_{signal}}$ will be assigned to their clusters while the maximum distance among all clusters is less than threshold

$\boldsymbol{D_{Threshold}}$. Thus, it is easy to show that $\boldsymbol{P_{signal}}$ is either a representative data point for decision-making, or a noisy data point. By increasing or decreasing the distance threshold $\boldsymbol{D_{Threshold}}$, the number of the representative data points (medoids) will decrease or increase correspondingly.

By grouping each cluster with different $\boldsymbol{P_{signal}}$ detected by different vehicles inside of each cluster, one clustering center (medoid) will represent one estimated pedestrian. (Based on Assumption 1)

After UPDATING, a group of medoids with minimum distance in each corresponding cluster will be generated. It can be shown intuitively that these medoids are able to offer good knowledge to make a decision compared to unfiltered raw data $\boldsymbol{P_{input}}$.

3.4 Simulation Example

Simulations were conducted using the Matlab and PreScan software by applying the Greedy-Medoids clustering algorithm to the scenarios shown in Figure 3.1, Figure 3.9 and Figure . In order to be able to show the results in a 2D plot, the Euclidian distance in x, y, z coordinate was used as the unit of $\boldsymbol{D_{Threshold}}$.

Simulation case 1: 2 near pedestrian scenario

Figure 3.8 shows the simulation results of $\boldsymbol{P_{output}}$ with $\boldsymbol{D_{Threshold}} = 4$ meters. The number of the pedestrians in $\boldsymbol{P_{output}}$ matches that in $\boldsymbol{P_{input}}$. The location of each pedestrian in $\boldsymbol{P_{output}}$ (red cross medoid in Figure 3.8) is a good estimate of that in $\boldsymbol{P_{input}}$ (black circle in Figure 3.8). As shown in Figure 3.8, compared to the original pedestrians in Figure 3.1, we can see the Greedy-Medoids clustering algorithm is acceptable without any prior knowledge of the number of the pedestrians.

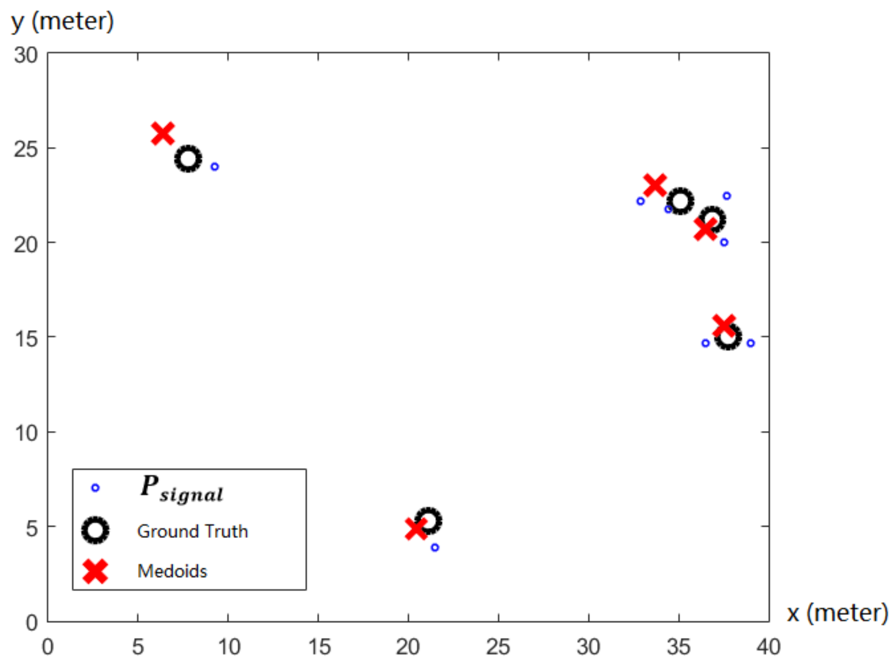


Fig. 3.8.: Clustering result with $\boldsymbol{D_{Threshold}} = 4$ meters. Note that medoids are also $\boldsymbol{P_{signal}}$.

Simulation case 2: Dense pedestrians scenario

There are six cars and nine pedestrians in an intersection. It is assumed that the error between all $\boldsymbol{P_{signal}}$ and the ground truth $\boldsymbol{P_{input}}$ are uniformly distributed within $[-1.5, 1.5]$ meters in both x and y directions. $\boldsymbol{P_{outout}}$ was generated by the Greedy-Medoid clustering algorithm with $\boldsymbol{D_{Threshold}} = 4$ meters (see Figure 3.10). The result demonstrates that the medoids of each cluster is close to a ground truth pedestrian.

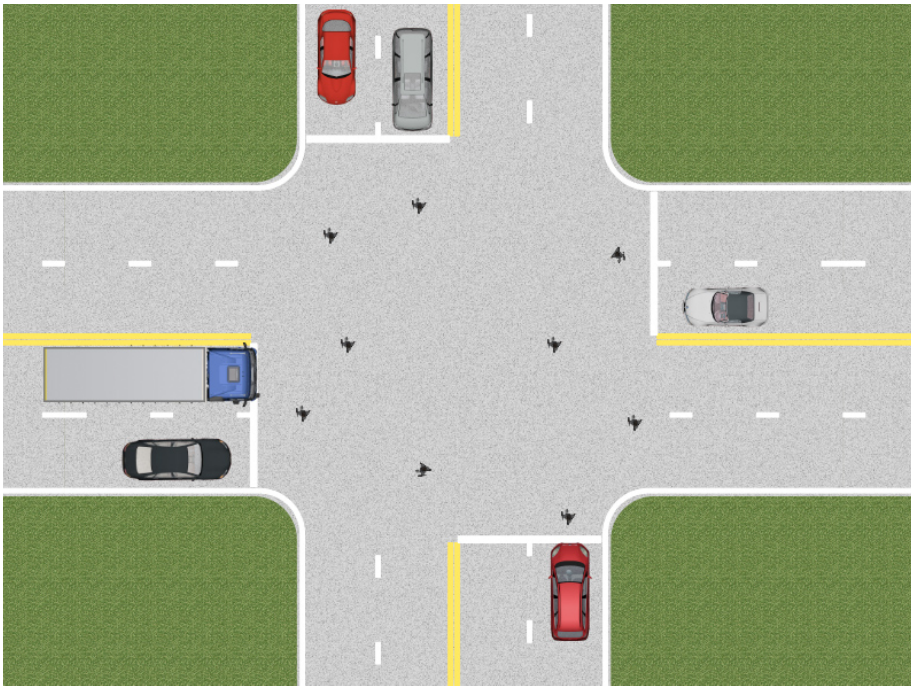


Fig. 3.9.: A scenario with 6 cars and 9 pedestrians.

From Figure 3.10, we can see the representative data points (medoids) are very close to ground truth pedestrian. It also filters out a large number of $\boldsymbol{P_{signal}}$ which are not likely a pedestrian. This result is a good example that shows the Greedy-Medoid clustering algorithm can reduce false positive and remain false negative.

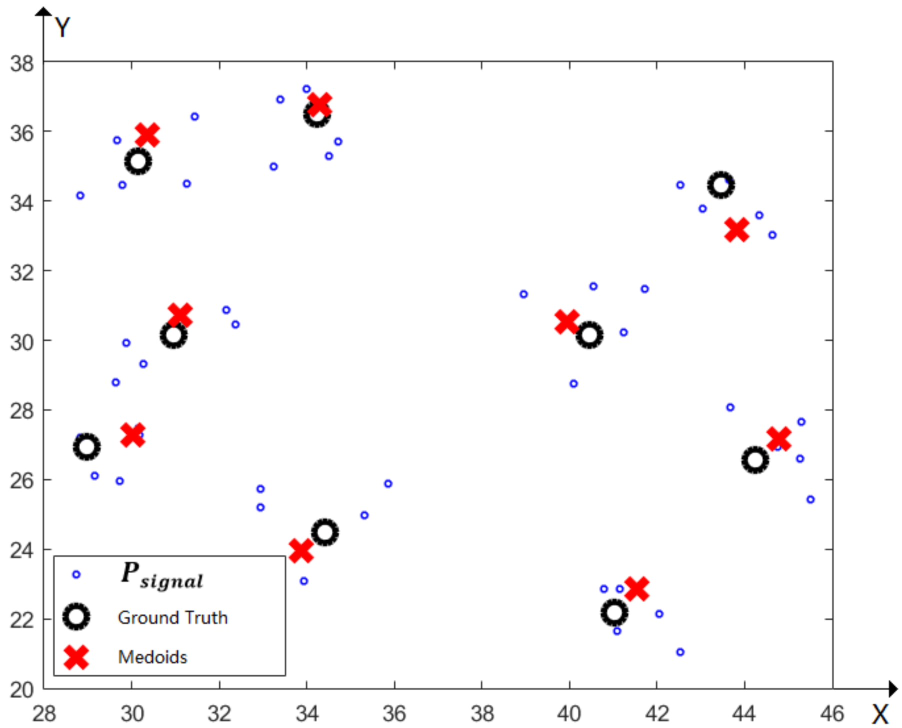


Fig. 3.10.: Clustering result of 6 cars 9 pedestrians scenario. Error= $[-1.5, 1.5]$, $\boldsymbol{D_{Threshold}} = 4$ meters.

Simulation case 3: Sparse pedestrian scenario

This example shows seven cars and five pedestrians in a two-lane road, where each car sees only one pedestrian (see Figure 3.4). The purpose of this scenario is to explain extreme cases that a car can only detect 1 or 0 pedestrians. Figure 3.11 and Figure 3.12 show the sparse $\boldsymbol{P_{signal}}$ cases with $\boldsymbol{D_{Threshold}} = 4$ meters and 1 meter, respectively. As the distance threshold $\boldsymbol{D_{Threshold}}$ is decreased from 4 meters to 1 meter, the number of the clusters increased (see Figures 3.11 and 3.12). If the threshold is fairly small, since none of the $\boldsymbol{P_{signal}}$ are adjacent to each other, final $\boldsymbol{P_{output}}$ is exactly the $\boldsymbol{P_{signals}}$.

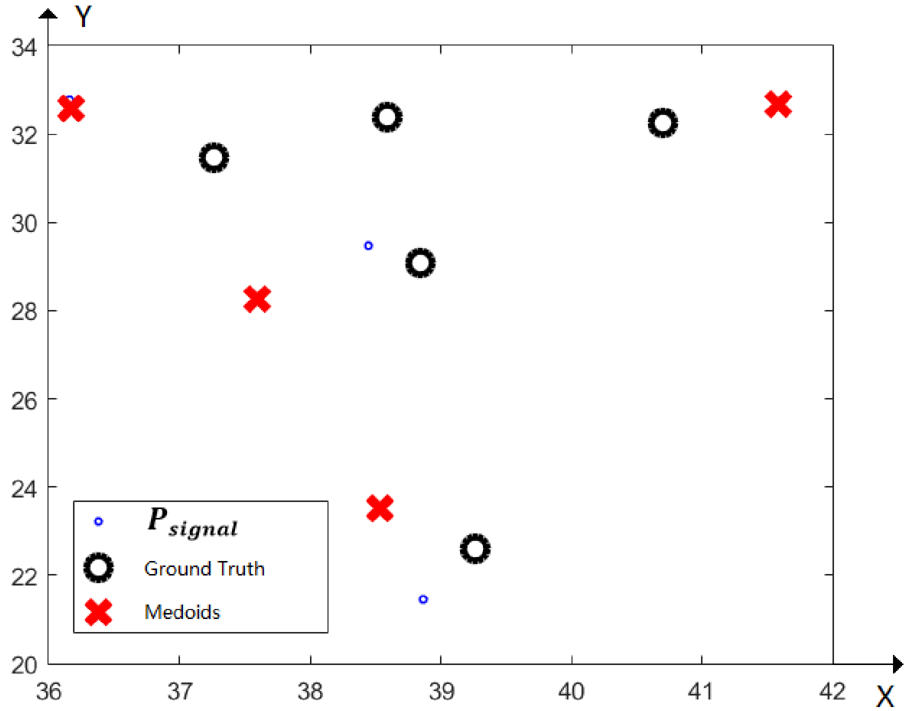


Fig. 3.11.: Clustering result of 7 cars 5 pedestrians' scenario. Error= $[-1.5, 1.5]$, $D_{Threshold} = 4$ meters.

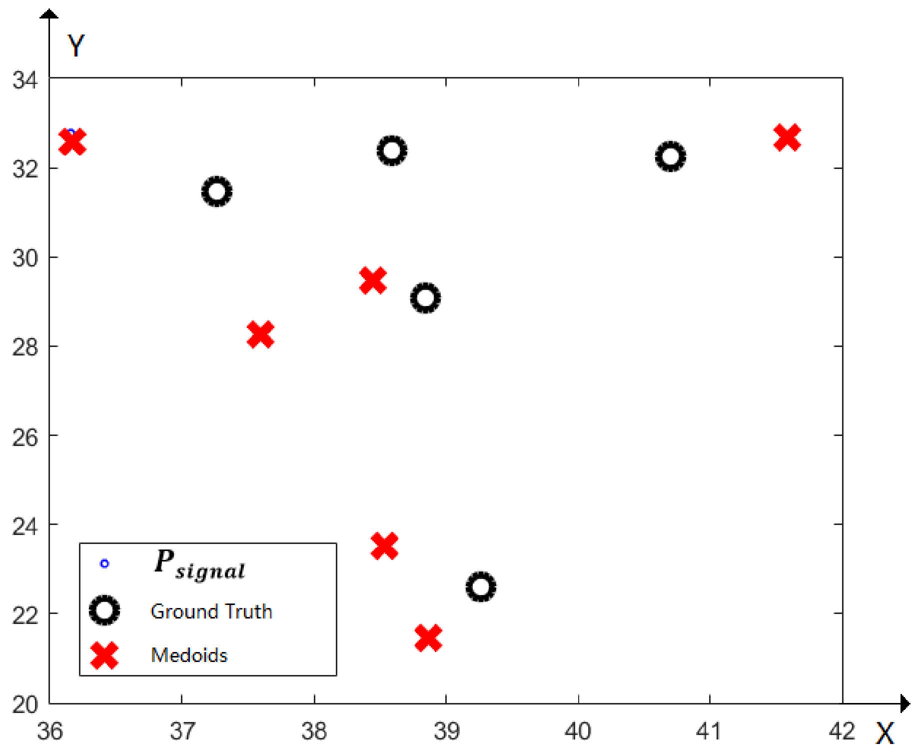


Fig. 3.12.: Clustering result of 7 cars 5 pedestrians' scenario. Error= $[-1.5, 1.5]$, $\boldsymbol{D_{Threshold}} = 1$ meter.

3.5 Computational Complexity Assessment

This section briefly discusses the running time assessment of ASSIGNMENT in the Greedy-Medoids clustering algorithm. Since V2V communication is a real-time process, focusing more on the time complexity takes precedence over the memory space complexity.

Suppose we can calculate the distance between two $\boldsymbol{P_{signal}}$ in linear time, then the worst case complexity of ASSIGNMENT can be expressed as,

$$(m-1)\left(\frac{wl_m(wl_m+1)}{2}(nlogn+p+1)\right) \tag{3.3}$$

where n is the number of the clusters (medoids), m is the number of the vehicles, wl_m is the length of the $\boldsymbol{WaitingList}$ with regards to vehicle m, and p represents the number of $\boldsymbol{P_{signal}}$ in one cluster.

Equation 3.3 shows that for each vehicle except $\boldsymbol{PrimeVehicle}(m-1)$, considering the worst case, each data $\boldsymbol{P_{current}}$ from $\boldsymbol{WaitingList}$ will be replaced by next $\boldsymbol{P_{current}}$ (push data point S back into $\boldsymbol{WaitingList}$, line 21-23 in ASSIGNMENT). So for each vehicle we need traverse at most $\frac{wl_m(wl_m+1)}{2}$ times. For each loop, the distances between $\boldsymbol{P_{current}}$ and the current Medoids need to be sorted (line 16), which can cost $nlogn$ in optimum running time [16]. After sorting, line 17 in the algorithm costs linear time $O(p)$, and the decision can be made in $O(1)$ time, where $O(\cdot)$ is well-known O-notation representing an asymptotic upper bound [16]. Note that the number of clusters (medoids) may also increase in each loop. But in the worst case n cannot be greater than the total number of $\boldsymbol{P_{signals}}$. Let k be the total number of $\boldsymbol{P_{signals}}$, Equation 3.3 can be rewritten as,

$$O(m \cdot wl_m^2 \cdot (nlogn + p)) \tag{3.4}$$

Since $m \cdot wl_m = k =$ number of the $\boldsymbol{P_{signals}}$ and $n \leq k, p \leq k$, Equation 3.4 can be expressed as,

$$O(k^3 logk) \tag{3.5}$$

Equation 3.5 is the upper bound running time of ASSIGNMENT. Similarly, the complexity is $O(k^2)$ in INITIALIZATION and $O(k^3)$ in UPDATING. Then the upper bound complexity of the GREEDY-MEDOIDS CLUSTERING should be,

$$O(k^2 + k^3(2 \cdot logk + 1)) = O(k^3 logk) \tag{3.6}$$

According to the specification of DSRC [17], [18], 0.1 seconds is the time interval of two consecutive timestamps (10 message/second). It is necessary to make sure the Greedy-Medoids clustering algorithm can generate results in less than 0.1 seconds.

Even though the running time of the Greedy-Medoids clustering algorithm grows rapidly when the number of the $\boldsymbol{P_{signals}}$ increases, in many scenarios with sparse pedestrians (for example, less than 5 people in a road intersection), the Greedy-Medoids clustering algorithm can cluster $\boldsymbol{P_{signals}}$ and reconstruct pedestrians very well within the given 0.1 seconds.

4. HIERARCHICAL CLUSTERING APPROACH - AN EFFICIENT WAY BASED ON ASSUMPTION 1

In Section III, an intuitive Greedy-Medoid Clustering Algorithm is proposed to solve Assumption 1 and Assumption 2 problem for identifying and classifying $\boldsymbol{P_{signals}}$ based on K-Medoids clustering and greedy approach. In this section, based on Assumption 1, an efficient hierarchical clustering algorithm is ultilized and a matrix based solution is proposed.

4.1 Matrix Based Solution

The matrix based solution (especially 2 dimensional distance matrix) is a popular and efficient way to illustrate the idea intuitively while solving the problem quick and smart. Usually the distance matrix, a 2-D array, contains the distance between each data point, are symmetric. It describes the relationship between each data points. In many different constraint clustering analysis, distance matrix is very useful to tackle the problem.

To illustrate the matrix based solution method, a case is adopted for demonstrate the idea and the meaning of the inside data in matrix.

Considering a scenario with 3 pedestrians (say ped $p1, p2, p3$) and 4 cars (say car A, B, C, D), suppose each vehicle can sense all pedestrians, then the total number of $\boldsymbol{P_{signals}}$ is $3 \times 4 = 12$. Table 4.1 shows the relationship of each $\boldsymbol{P_{signal}}$. Figure 4.1 shows the visualized data. $\boldsymbol{P_{signals}}$ is simulated by a uniform distribution ($[-1.5, 1.5]$ meters).

Table 4.1: 3 Pedestrians 4 Cars' Parameters

$\boldsymbol{P_{signal}}$ ID	**Locations** (x, y)	**Detected by which vehicle** ($Vehicle\ ID$)	**Pedestrian**
1	(6.644199, 17.282080)	A	$p1$
2	(5.792567, 17.193826)	B	$p1$
3	(5.697515, 18.725224)	C	$p1$
4	(3.995532, 17.816467)	D	$p1$
5	(8.252002, 3.673744)	A	$p2$
6	(7.435576, 4.865974)	B	$p2$
7	(8.095334, 5.337754)	C	$p2$
8	(8.918196, 3.170066)	D	$p2$
9	(11.912089, 16.507965)	A	$p3$
10	(10.333733, 17.614958)	B	$p3$
11	(10.758729, 19.076866)	C	$p3$
12	(11.115410, 17.488841)	D	$p3$

The order of $p1, p2, p3$ does not taken seriously consideration. All data follow uniform distribution $[-1.5, 1.5]$ meters on both x and y axes.

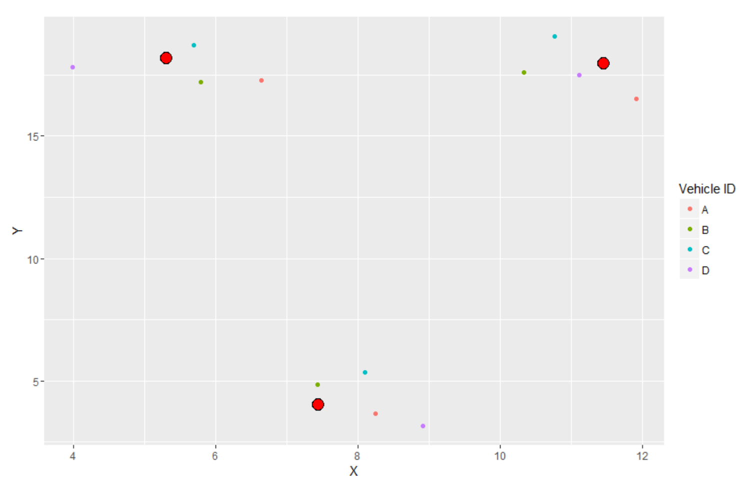


Fig. 4.1.: 3 pedestrians 4 cars' scenario with 12 $\boldsymbol{P_{signal}}$ and ground truth data points.

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.000000	0.856193	1.725942	2.702038	Inf	12.4413000	12.0321526	14.2940541	Inf	3.7045196	4.488944	4.4759885
2	0.856193	0.000000	1.534345	1.901845	13.7419590	Inf	12.0776311	14.3678600	6.157837	Inf	5.311178	5.3310121
3	1.725942	1.534345	0.000000	1.929401	15.2667108	13.9678082	Inf	15.8850789	6.598269	4.7673055	Inf	5.5571775
4	2.702038	1.901845	1.929401	0.000000	14.7693651	13.3995955	13.1349397	Inf	8.023967	6.3414029	6.879640	Inf
5	Inf	13.741959	15.266711	14.769365	0.0000000	1.4449792	1.6713692	0.8351682	Inf	14.0957807	15.605763	14.1087205
6	12.441300	Inf	13.967808	13.399596	1.4449792	0.0000000	0.8110837	2.2526136	12.472975	Inf	14.594273	13.1483053
7	12.032153	12.077631	Inf	13.134940	1.6713692	0.8110837	0.0000000	2.3186149	11.804288	12.4795893	Inf	12.5207731
8	14.294054	14.367860	15.885079	Inf	0.8351682	2.2526136	2.3186149	0.0000000	13.669782	14.5140839	16.012928	Inf
9	Inf	6.157837	6.598269	8.023967	Inf	12.4729751	11.8042882	13.6697816	0.000000	1.9278592	2.815936	1.2636515
10	3.704520	Inf	4.767306	6.341403	14.0957807	Inf	12.4795893	14.5140839	1.927859	0.0000000	1.522432	0.7917856
11	4.488944	5.311178	Inf	6.879640	15.6057634	14.5942728	Inf	16.0129276	2.815936	1.5224320	0.000000	1.6275890
12	4.475989	5.331012	5.557177	Inf	14.1087205	13.1483053	12.5207731	Inf	1.263652	0.7917856	1.627589	0.0000000

Fig. 4.2.: 12 by 12 $\boldsymbol{P_{signal}}$ Euclidean distance relationship matrix.

The matrix based solution is, for example, considering $\boldsymbol{P_{signal}}$ 1 and $\boldsymbol{P_{signal}}$ 5 are both detected by vehicle A, then $\boldsymbol{P_{signal}}$ 1 and $\boldsymbol{P_{signal}}$ 5 cannot be classified into one cluster. Under this condition, we can plot a matrix where it has 12 rows and 12 columns (same number of $\boldsymbol{P_{signal}}$), and measure the Euclidean 2D distances of each pair of $\boldsymbol{P_{signal}}$. "Inf" means they cannot be classified into one cluster, as set them infinity. See Figure 4.2.

Figure 4.2 illustrates the connections between each $\boldsymbol{P_{signal}}$ data points based on Euclidean distances. Data should be clustered automatically based on there strong/weak connections. (Say the smaller value that the distance have, then the stronger their connection is)

Based on this matrix, a hierarchical clustering under Assumption 1 is proposed to cater this problem.

4.2 Hierarchical Clustering

The hierarchical clustering is a method of clustering analysis to cluster dataset in a hierarchical way. In hierarchical clustering, agglomerative and divisive are two typical strategies to do the clustering. In general, the merges and splits are determined in a greedy manner. The results of hierarchical clustering are usually presented in a dendrogram. [19] In this paper, Ward's method [20] is used for merging the data points together and grows up till reaching the root [21], [22], where Calinski-Harabasz Pseudo F-Statistic [23] is adopted to determine the number of the clusters.

Based on Table 4.1 example, a "bottom up" agglomerative clustering is applied. Figure 4.3 shows the hierarchical relationship of the dataset under the condition of Figure 4.2. The reason we cluster data point 6, 7, 5, 8 in one group is because the relationship between any pair of these data points are relatively close, which verifies the usefulness of the distance matrix and the "infinity distance" constraint.

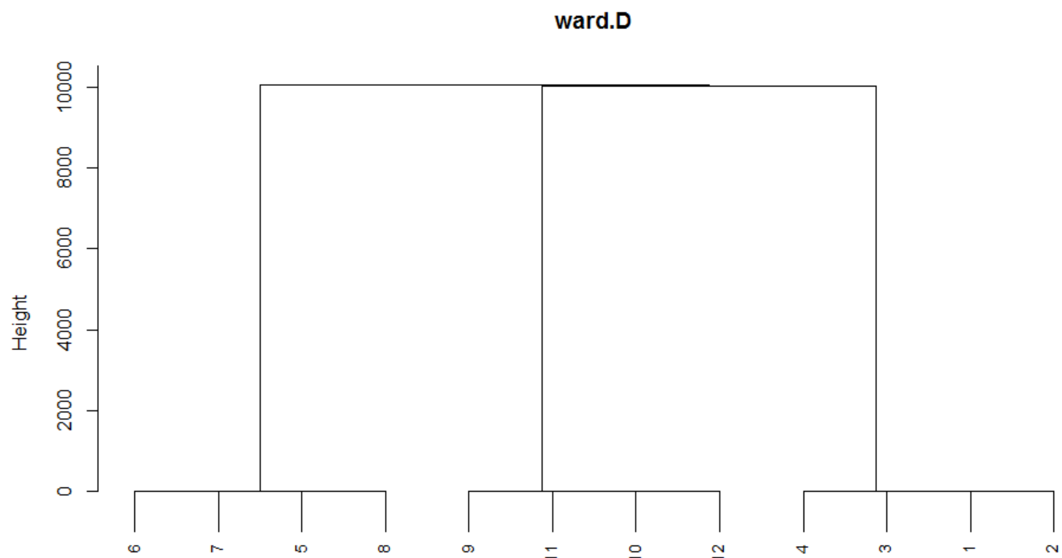


Fig. 4.3.: 3 pedestrians 4 cars' Ward's method hierarchical clustering dendrogram.

4.3 Determine the Number of the Clusters

Comparing with the Greedy-Medoid Clustering Algorithm which it use $\boldsymbol{D_{Threshold}}$ to determine the number of the clusters, as an semi-supervised machine learning, hierarchical clustering method still need an approach to estimate the number of the clusters. There're several popular methods to determine the number of the clusters:

Elbow method [24]; Average silhouette method [25]; Gap statistic method [26], etc. As mentioned above, in this thesis, Calinski-Harabasz Pseudo F-Statistic [23] is adopted.

Based on our experiments, such 3 pedestrians 4 cars scenario does not show a good quality of the estimation due to the number of the signals is insufficient. More densed case will provides us a good estimation of the clusters.

Simulation case 5:

To test this hierarchical clustering method, another scenario with 8 pedestrians and 10 cars are evaluated. $\boldsymbol{P_{signals}}$ is simulated by a uniform distribution ($[-3, 3]$ meters). As shown in Figure 4.4, the original $\boldsymbol{P_{signals}}$ are really hard to tell where they come from.

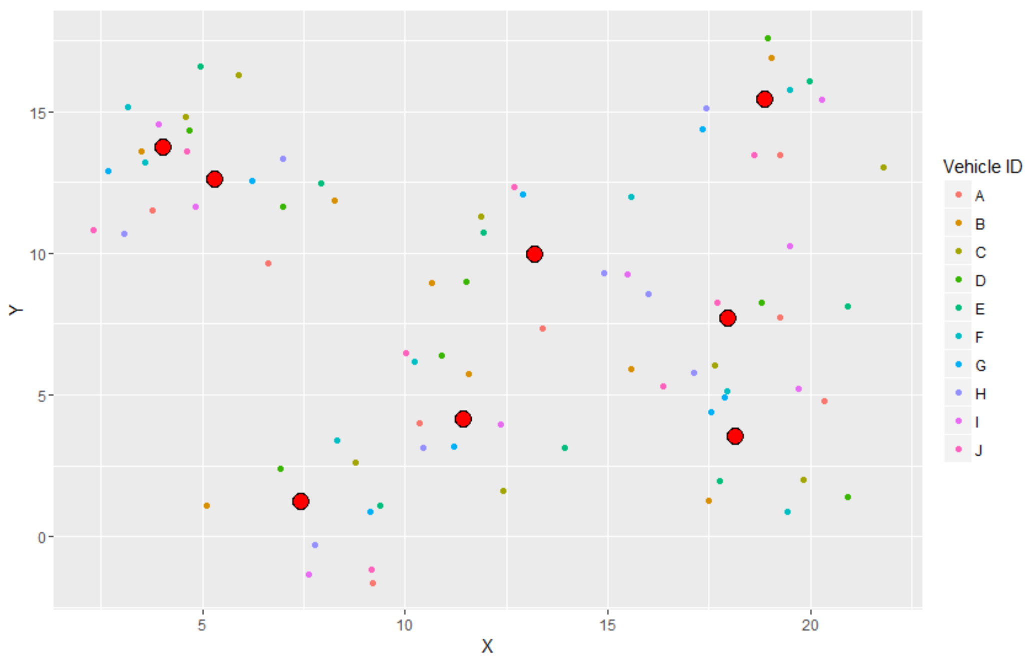


Fig. 4.4.: 8 pedestrians 10 cars' scenario with 80 $\boldsymbol{P_{signal}}$ and ground truth data points.

Figure 4.5 shows the hierarchical relationship of the dataset under the condition of Figure 4.4. By Letting the cut (number of the clusters) equals to from 2 to 79 (number of the clusters cannot be less than 2 and greater or equal to 80), the gap statistics (where the number of Monte Carlo (bootstrap) samples = 50) is shown in

Figure 4.6, and it concludes that when $k = 7$, i.e. the number of the clusters = 7, hierarchical clustering approach yields the optimal solution for this scenario. The final result of the Figure 4.4 data points aggregation (based on Euclidean distance) is shown as Figure 4.7. The final result of the estimated pedestrians (choose the mean value of each group) is shown as Figure 4.8.

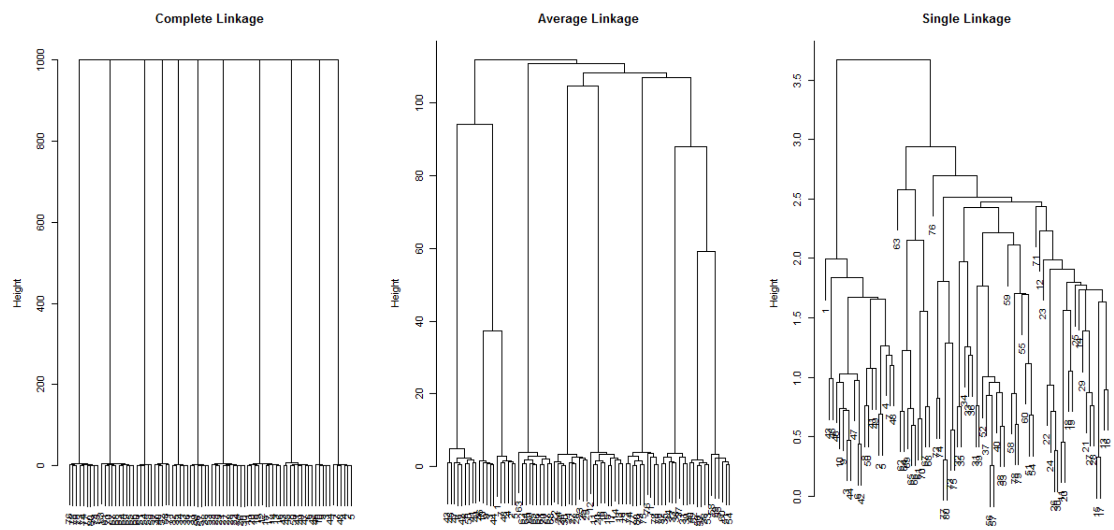


Fig. 4.5.: 8 pedestrians 10 cars' complete-link, average-link, and single-link hierarchical clustering dendrogram.

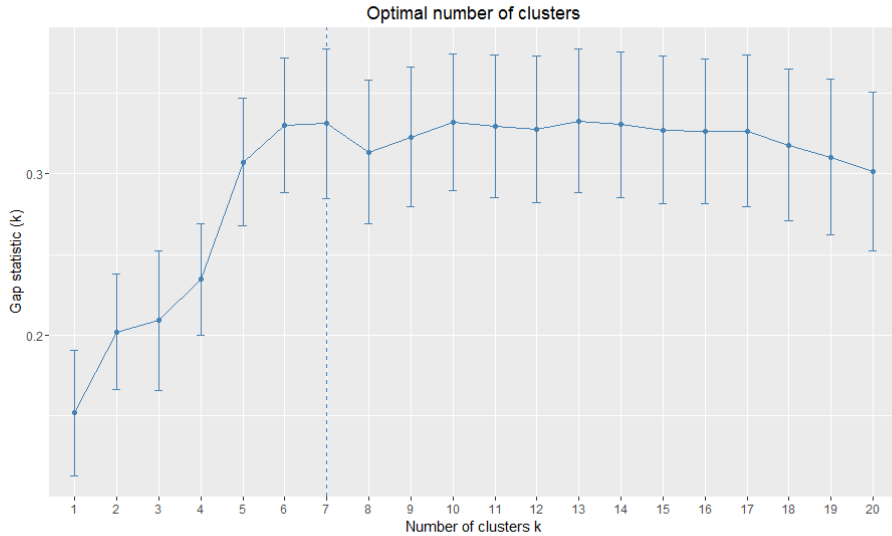


Fig. 4.6.: 8 pedestrians 10 cars' gap statistics with cluster number from 2 to 20 (21 - 79 is omitted). Number of Monte Carlo (bootstrap) samples = 50.

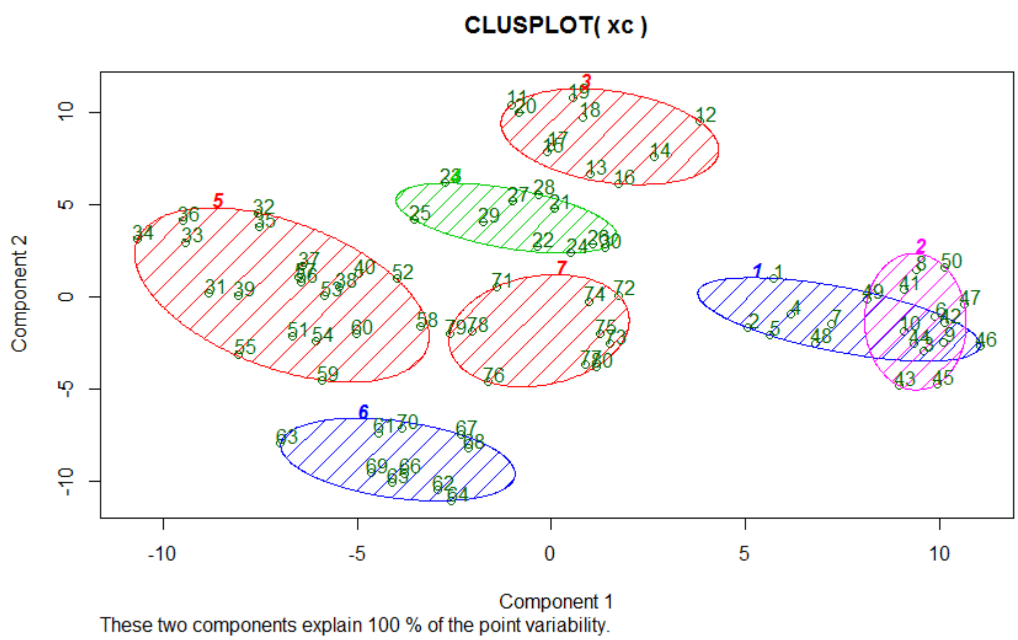


Fig. 4.7.: 8 pedestrians 10 cars' scenario matrix data points aggregation (based on Euclidean distance).

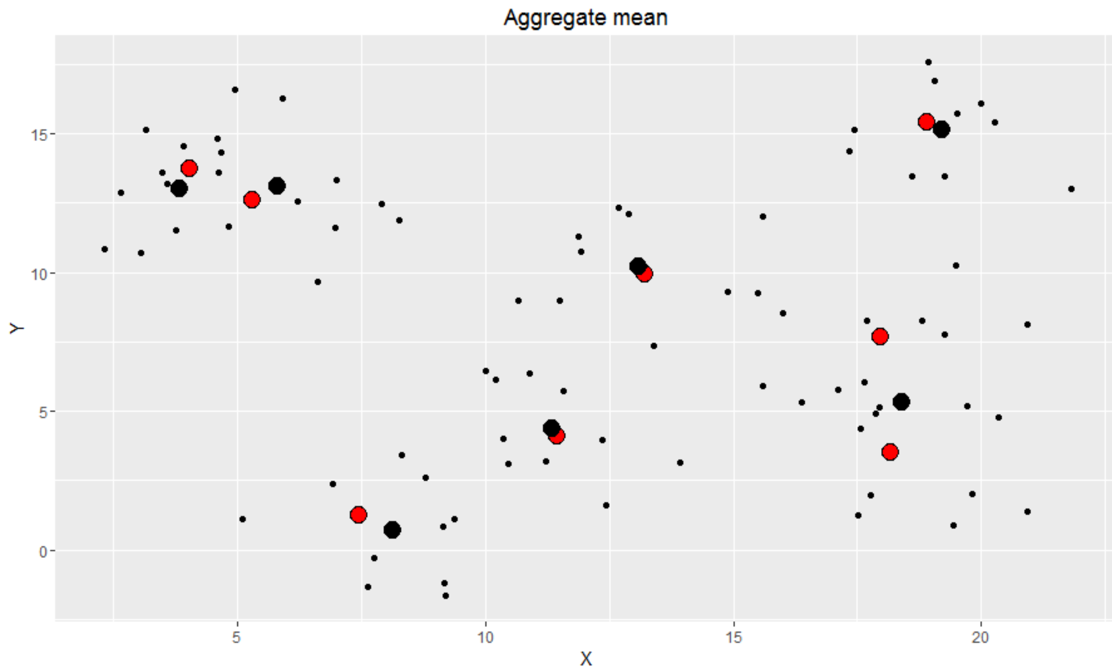


Fig. 4.8.: 8 pedestrians 10 cars' scenario estimating result. Mean value estimated pedestrians are the big black circles, ground truth are the big red circles, $\boldsymbol{P_{signals}}$ are the small black dots.

From Figure 4.8, it is easily to be observed that the quality of hierarchical clustering, even the signal applies strong noises, is really robust and works pretty well.

4.4 Computational Complexity and the Comparision with Greedy-Medoid Clustering Algorithm

It has been proved that the time complexity of average-link hierarchical clustering is $O(n^2 \cdot logn)$ [27]. Comparing with Greedy-Medoid Clustering Algorithm, hierarchical clustering method create a dendrogram first, then by cutting them into separate groups, hierarchical clustering can obtain the best estimated number of the clusters based on the gap statistics. Also, by utilizing the Assumption 1 in matrix based approach, hierarchical clustering algorithm solves the pedestrians' signals clustering problem without any former knowledge about the dataset nor the number of the groups it should be divided. By utilizing gap statistics, hierarchical clustering is able to determine the number of the cluster and the groups of the data points.

However, Greedy-Medoid Clustering Algorithm provides an easier understanding on how the data association procedure works. Greedy-Medoid Clustering algorithm also describes the problem based on the distance threshold, which provide a more nature way to aggregate/disperse data points comparing with the gap statistics. Thats being said, hierarchical clustering method still provides a smarter, more efficient approach which could be used in the real world.

5. CONCLUSION

As PAEB and V2V technologies are becoming mature, sending PAEB detected pedestrian information to the V2V network provides a potential benefit to make safety decisions earlier and more effective. This paper has provided a solution for a specific pedestrian data fusion problem in the V2V-PAEB system. A mathematical model of the pedestrian information generated by the PAEB system in the V2V network was introduced. The proposed Greedy-Medoids clustering algorithm enables a subject vehicle to approximate the number of pedestrians and their estimated locations from a large number of pedestrian alert messages by many nearby vehicles through the V2V network and the subject vehicle itself. The simulation results have demonstrated the effectiveness and applicability of the proposed method.

This thesis also briefly introduce the hierarchical clustering and distance matrix apply on the V2V-PAEB problem. A simulation is given to verify the usefulness of this technique. A brief comparison between Greedy-Medoid clustering algorithm and hierarchical clustering is introduced.

Both results of Greedy-Medoid clustering and hierarchical clustering can be useful for PAEB system to make the warning/braking decisions earlier and hence, improving its pedestrian safety performance. The same idea can be applied to other objects (such as bicyclists) on the road.

6. SUMMARY

The purpose of this thesis is to estimate the actual number and geolocation of pedestrian signals during processing the AEB messages in a V2V network. The result can help the AEB system to make safety relate action more accurately. Vehicles can respond to potential collision to pedestrians early do avoid driving into those dangerous areas.

This paper is trying to envision the near future transportation system when all cars are mounted on V2V devices and be able to communicate and exchange messages. At that moment, the shared information needs to filter out duplicated data and errors. Thus, clustering analysis will be considered as an approach to do this process. In that sense, Greedy-Medoid clustering algorithm provides a effective way to estimate pedestrians from various signals. Hierarchical clustering and number of the clusters estimation also provide an applicable way to solve this specific problem.

LIST OF REFERENCES

LIST OF REFERENCES

[1] Q. Xu, T. Mak, J. Ko, and R. Sengupta, “Vehicle-to-vehicle safety messaging in dsrc,” in *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pp. 19–28, ACM, 2004.

[2] E. Rosen, “Autonomous emergency braking for vulnerable road users,” in *IRCOBI Conference Proceedings*, no. IRC-13-71, 2013.

[3] E. Coelingh, A. Eidehall, and M. Bengtsson, “Collision warning with full auto brake and pedestrian detection-a practical example of automatic emergency braking,” in *13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 155–160, IEEE, 2010.

[4] C. Premebida, G. Monteiro, U. Nunes, and P. Peixoto, “A lidar and vision-based approach for pedestrian and vehicle detection and tracking,” in *2007 IEEE Intelligent Transportation Systems Conference*, pp. 1044–1049, IEEE, 2007.

[5] M. Szarvas, U. Sakai, and J. Ogata, “Real-time pedestrian detection using lidar and convolutional neural networks,” in *2006 IEEE Intelligent Vehicles Symposium*, pp. 213–218, IEEE, 2006.

[6] B. Tang, “Pedestrian protection using the integration of v2v communication and pedestrian automatic emergency braking system,” *Purdue University, M.S.E.C.E Thesis*, 2015.

[7] T. Wang, R. Aggarwal, and A. Somani, “Human tracking using delphi esr-vision fusion in complex environments,” in *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*, p. 198, The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2015.

[8] M. Liu, S. Chien, and Y. Chen, “Improve road safety using combined v2v and pre-collision systems,” in *24th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, 2015.

[9] R. Burtch, “A comparison of methods used in rectangular to geodetic coordinate transformations,” in *American Congress on Surveying and Mapping (ACSM)*, Citeseer, 2006.

[10] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*, vol. 344. John Wiley & Sons, 2009.

[11] D. J. Sheskin, *Handbook of parametric and nonparametric statistical procedures.* CRC Press, 2003.

[12] C. M. Bishop, “Pattern recognition,” *Machine Learning*, vol. 128, 2006.

[13] L. Kaufman and P. Rousseeuw, *Clustering by means of medoids*. North-Holland, 1987.

[14] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Kdd*, vol. 96, pp. 226–231, 1996.

[15] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: The algorithm gdbscan and its applications," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 169–194, 1998.

[16] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, vol. 6. MIT press Cambridge, 2001.

[17] Y. J. Li, "An overview of the dsrc/wave technology," in *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, pp. 544–558, Springer, 2010.

[18] K. Ansari, C. Wang, L. Wang, and Y. Feng, "Vehicle-to-vehicle real-time relative positioning using 5.9 ghz dsrc media," in *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th*, pp. 1–7, IEEE, 2013.

[19] L. Rokach and O. Maimon, "Clustering methods," in *Data mining and knowledge discovery handbook*, pp. 321–352, Springer, 2005.

[20] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244, 1963.

[21] W. H. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *Journal of classification*, vol. 1, no. 1, pp. 7–24, 1984.

[22] A. Siddharthan, "Christopher d. manning and hinrich schutze. foundations of statistical natural language processing. mit press, 2000. isbn 0-262-13360-1. 620 pp. $64.95/£ 44.95 (cloth).," 2002.

[23] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[24] D. J. Ketchen and C. L. Shook, "The application of cluster analysis in strategic management research: an analysis and critique," *Strategic management journal*, vol. 17, no. 6, pp. 441–458, 1996.

[25] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[26] T. Hastie, R. Tibshirani, and G. Walther, "Estimating the number of data clusters via the gap statistic," *J Roy Stat Soc B*, vol. 63, pp. 411–423, 2001.

[27] H. Schütze and C. Silverstein, "Projections for efficient document clustering," in *ACM SIGIR Forum*, vol. 31, pp. 74–81, ACM, 1997.