Journal of Systems Architecture 97 (2019) 258–268

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

ECASS: Edge computing based auxiliary sensing system for self-driving vehicles

Xiong Wang[a], Tianpeng Wei[a], Linghe Kong[a,*], Liang He[b], Fan Wu[a], Guihai Chen[a]

[a] Shanghai Jiao Tong University, China
[b] University of Colorado Denver, USA

ARTICLE INFO

Keywords:
Autopilot systems
Self-driving vehicle
Trajectory
GPS

ABSTRACT

Self-driving vehicles, combining automobiles with autopilot systems, enable intelligent and safe driving. Self-driving vehicles can achieve accurate automatic navigation, trajectory tracking, and automatic overtaking by using GPS, radars, and inertial measurement unit (IMU). Among them, overtaking is essential in order to avoid excessive waiting time and improve the traffic efficiency. When following a large truck or bus, the self-driving vehicle cannot ensure the safe overtaking because the line-of-sight (LOS) range detected by the radar and camera is blocked, thus unable to perceive the surrounding environment accurately. A commonly adopted mitigation is to follow the truck or bus at a reduced speed, at the cost of reduced traffic efficiency and more traffic jams. To mitigate this deficiency, this paper develops an auxiliary sensing system using edge computing to locate nearby vehicles for self-driving vehicles, called ECASS. Specially, infrastructure deployed along the road like servers are utilized to accurately locate vehicles according to GPS and wireless information such as WiFi or DSRC. Subsequently, the server will transmit the localization information of nearby vehicles to the self-driving vehicle, based on which it can determine the driving state for the next moment despite of the obstruction. Extensive simulations verify that ECASS based trajectory is much closer to the real trajectory than GPS. Especially when GPS error is set within 10 m, ECASS can reduce the mean absolute localization error from more than 7 m to about 3 m.

1. Introduction

In recent years, self-driving vehicles have attracted extensive attention from both industry and academia because of its safety and travelling efficiency [1–3]. In particular, with the increase in the number of vehicles, the injured persons and deaths in traffic accidents are also increasing. As a critical technique in future vehicles, autopilot can substantially decrease the traffic accidents induced by human factors and thus enhance the travelling safety. The traffic sharing based on the automatic driving technique can effectively alleviate the traffic congestion and pollution problems [4,5].

Researchers have designed various detection algorithms based on the image and signal processing, to accurately sense the environments around the self-driving vehicle [6–8]. The self-driving vehicle will then determine the driving state for the next moment, such as overtaking, acceleration/deceleration, or travelling at the original speed based on the acquired environmental information.

Compared to acceleration/deceleration, or travel at the original speed, the execution of overtaking for self-driving vehicles becomes much more complicated. It mainly consists of three steps. Firstly, the self-driving vehicle changes the lane according to the planned trajectory. Secondly, it drives along the overtaken vehicle at a prescribed lateral distance. Finally, it will return to the original lane in front of the overtaken vehicle [9,10]. Majority of research work on this problem has focused on the planning or prediction of the overtaking trajectory [9,11,12].

Available literature has planned their overtaking trajectories under the premise that the radar or camera integrated in the self-driving vehicle can detect and track obstacles without buses or trucks around when overtaking. However, under some special circumstances, the camera and radar may be blocked by the truck or bus in front or behind, rendering the self-driving vehicle partially or even completely unknown about the surroundings. Therefore, it has to follow the truck or bus at a reduced speed [13]. Obviously, this approach incurs increased time consumption and traffic congestion.

However, GPS based localization errors can reach up to 10 m, and even localization errors in map-matching based GPS suffer from 5 m [14,15]. Such a large localization error may cause the wrong driving state adoption for self-driving vehicles, risking the traffic safety. In result, the auxiliary sensing system named ECASS is developed to provide accurate vehicle localization information in the proximity of the self-

* Corresponding author.
E-mail address: linghe.kong@sjtu.edu.cn (L. Kong).

https://doi.org/10.1016/j.sysarc.2019.02.014
Received 5 October 2018; Received in revised form 21 January 2019; Accepted 13 February 2019
Available online 13 February 2019
1383-7621/© 2019 Elsevier B.V. All rights reserved.

driving vehicle based on edge computing, when it cannot accurately locate nearby vehicles only using the camera and radar.

In particular, when the self-driving vehicle cannot accurately sense the surrounding environments due to obstruction, it will send a request to nearby servers, so as to inform these servers of obtaining the current vehicle localization near the self-driving vehicle. In these servers, an information fusion algorithm with regard to the wireless signal and GPS is designed to estimate the position of nearby vehicles. Finally, the location of each vehicle near the self-driving vehicle will be transmitted to the self-driving vehicle. The self-driving vehicle determines the driving state for the next moment based on the acquired localization information, despite of the obstruction. In this paper, the driving state includes overtaking, changing lanes, acceleration, deceleration, braking, and travelling at the original speed.

The contributions of this paper are listed as below:

- We propose an edge computing based framework to assist self-driving vehicles to achieve accurate nearby vehicle localization and tracking when self-driving vehicles cannot accurately sense the surrounding environment.
- In roadside servers, an fusion algorithm related to GPS and wireless signal information is developed to measure the location of vehicles near the self-driving vehicle. According to these acquired localization information, the self-driving vehicle can determine the driving state for the next moment even when partially or completely blocked.
- Extensive simulations are carried out to demonstrate ECASS's high efficiency. Compared to GPS, ECASS based localization is much closer to the real vehicle position, especially when the GPS based localization error becomes larger.

The paper is organized as follows: Section 2 presents the related literature on self-driving vehicles. The preliminaries are introduced in Section 3, including the motivation and problem statement. Then, the system overview, algorithm design, and the selection strategy of servers are presented in Section 4, and we present the performance evaluation in Section 5. Finally, we conclude this paper in Section 6.

2. Related work

Up to now, there exists substantial research work related to the autopilot technique, including the hardware design such as detection radar and camera, algorithm design in terms of information fusion from these hardware, and the trajectory plan in the travelling process [16–18].

Hardware Hardware is the fundamental part for autopilot. High-quality hardware is able to perceive nearby information more accurately [19,20]. For example, Mercedes-Benz equip S-Class S 500 INTELLIGENT DRIVE with close-to-production sensor hardware. In particular, vision and radar sensors combined with digital maps are employed to sense nearby traffic conditions [21], and this system has been tested in an autonomous manner from Mannheim to Pforzheim, Germany. As a promising technique, lidar can also perceive the environment in the same way as radar. Due to much shorter wavelength, the high resolution and reliability render it a necessity for driverless cars in the future [2]. However, the size, complexity, and cost of the current generation of lidar sensors hinder its commercialization. Therefore, extensive academic and industry research has attempted to make lidar sensors smaller, easier to manufacture, and cheaper [6,22].

Information fusion algorithm After obtaining the information about nearby environments, how to deal with these massive information becomes crucial for self-driving vehicles [23,24]. Based on the stereo camera system, Franke et al. [25] present the vision algorithms for object recognition and tracking, free-space analysis, traffic light recognition, lane recognition, as well as self-localization. Further, in order to realize an accurate visual understanding of complex urban street scenes, a benchmark suite and large-scale dataset named Cityscapes is introduced to train and test pixel-level and instance-level semantic labeling. Both detailed analysis and performance evaluation have been carried out based on the proposed benchmark. Unlike camera based detection, a signal processing algorithm based on radar is designed to estimate the speed and size of vehicles in [26]. Finally, some research work proposes to build a vehicle detection system fusing radar and vision data [27].

Trajectory plan algorithm According to the accurate understanding of surrounding environments, self-driving vehicles can determine the driving state for the next moment [28,29], such as braking, acceleration, changing the lane, or overtaking. Among them, overtaking is the most complicated process. To achieve safe overtaking, Milanes et al. [30] develop a fuzzy-logic based controller to control the lateral movement and longitudinal movement of vehicles. Meanwhile, a stereo vision system is applied to detect any preceding vehicle and trigger the autonomous overtaking manoeuvre. Furthermore, a mathematical model and adaptive controller for autonomous overtaking maneuver is presented in [9]. Especially, an adaptive control scheme is designed to allow tracking the desired trajectories with unknown velocity of the overtaken vehicle compared to previous work. The authors of [31] proposed an path planning scheme for the self-driving car under the complex environments. It mainly consists of three parts, respectively as the novel path representation, the collision detection and the path modification. Finally, a multiple-goal reinforcement learning framework is constructed to tackle multiple criteria for overtaking in [12]. Simulation results demonstrate the high efficiency of the proposed strategy.

3. Preliminaries

In this section, we present the motivation behind ECASS and the problem statement.

3.1. Motivation

Nowadays, the traffic condition becomes increasingly complicated in metropolises. Therefore, there exist several critical challenges for self-driving vehicles to deal with various kinds of traffic scenarios. For example, the self-driving vehicle can be easily blocked by the truck or bus in front or behind, as shown in Fig. 1, in which case the self-driving vehicle is travelling behind the truck. Then, the radar or camera mounted on the top of the self-driving vehicle cannot detect and locate any obstacle in region 1, due to be blocked by the truck in front.

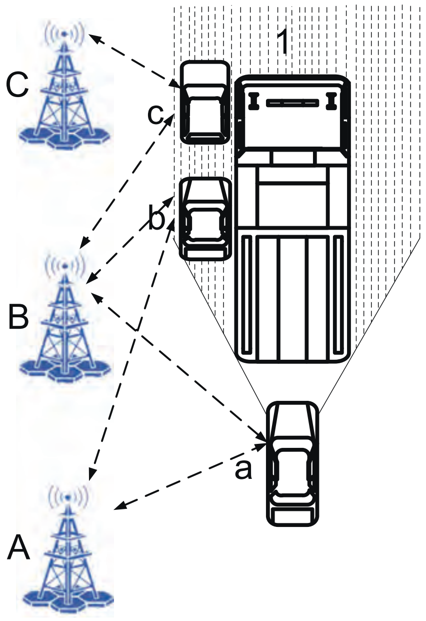

Fig. 1. The scenario when the self-driving vehicle is blocked.

Consequently, the self-driving vehicle cannot determine the travelling state for the next moment because it cannot accurately perceive the surrounding environment. The traditional strategy that self-driving vehicles adopt is to follow the truck or bus in front at a reduced speed. Therefore, this mechanism will incur much more time consumption and traffic congestion, thus resulting in more pollution.

3.2. Problem statement

In recent years, edge computing has been proposed to provide faster network response and more safety guarantee using the open platform integrated with networking, computing, storage and application close to objects or data sources, compared to data center based computing [32]. In this paper, we combine self-driving vehicles with edge computing to accurately locate vehicles near the self-driving vehicle in case of obstruction.

In self-driving vehicles, one second is divided into n time slots. For each slot, the self-driving vehicle will execute instructions from vehicle controllers, to ensure the travelling safety. Assuming at time t, the self-driving vehicle waits for the driving state instruction for the next moment–$t + 1/n$. Then, the vehicle controller will determine the travelling state for the next moment according to the acquired localization information.

As in Fig. 1, the self-driving vehicle, referred to as vehicle a, is blocked by the truck in front. Although equipped with the camera and radar, the self-driving vehicle still cannot detect and locate vehicles b and c on the left side of the truck. The roadside infrastructure like servers are proposed to assist vehicle a to detect and locate vehicles b and c through information interaction. According to these acquired information, the self-driving vehicle a can determine the driving state for the next moment.

To accurately locate vehicles near the self-driving vehicle, GPS localization information and wireless signals from vehicles will be delivered to nearby servers. According to the wireless signal information, the server can obtain the angle of arrival (AOA) relative to the vehicle [33]. In the presence of obstruction, an accurate vehicle localization algorithm for self-driving vehicles can be developed based on the fusion of GPS localization and AOA information. Assuming the true location for vehicle b is $b_{t,x}$ and $b_{t,y}$, the measured position $b_{m,x}$ and $b_{m,y}$ based on the designed vehicle localization algorithm should satisfy the formula as below.

$$\min\left(\sqrt{(b_{t,x} - b_{m,x})^2 + (b_{t,y} - b_{m,y})^2}\right). \tag{1}$$

4. System design

The framework of ECASS is shown in Fig. 2. In this section, the workflow of ECASS is presented, followed by the introduction of vehicle localization algorithm. Finally, we present the selection scheme of servers near the self-driving vehicle.

4.1. System overview

In the proposed system, every self-driving vehicle is integrated with GPS and one wireless antenna such as WiFi antenna or DSRC antenna. The GPS localization information of vehicles is delivered to nearby servers through wireless communication. The wireless antenna in vehicle a is referred to as An_a. An_a will select two nearby servers to communicate with, so as to accurately locate vehicle a. The workflow of ECASS is presented as follows, as shown in Fig. 3.

If the self-driving vehicle can accurately sense the surrounding environment according to the traffic condition obtained from the camera, radar, and IMU, then it plans the driving state based on the decision algorithm. Otherwise, it will send a request to roadside severs for obtaining nearby vehicle location. Subsequently, these nearby servers will estimate the vehicle's location near the self-driving vehicle using the

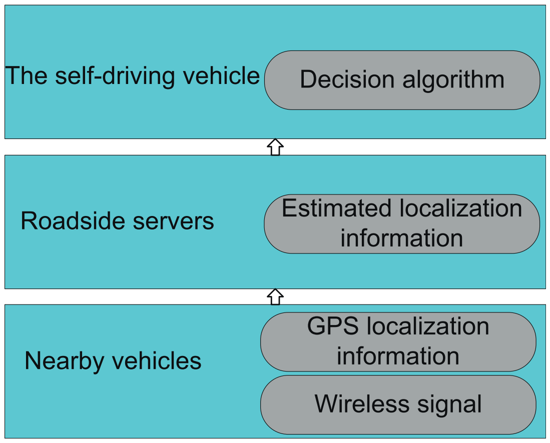

Fig. 2. The framework of ECASS.

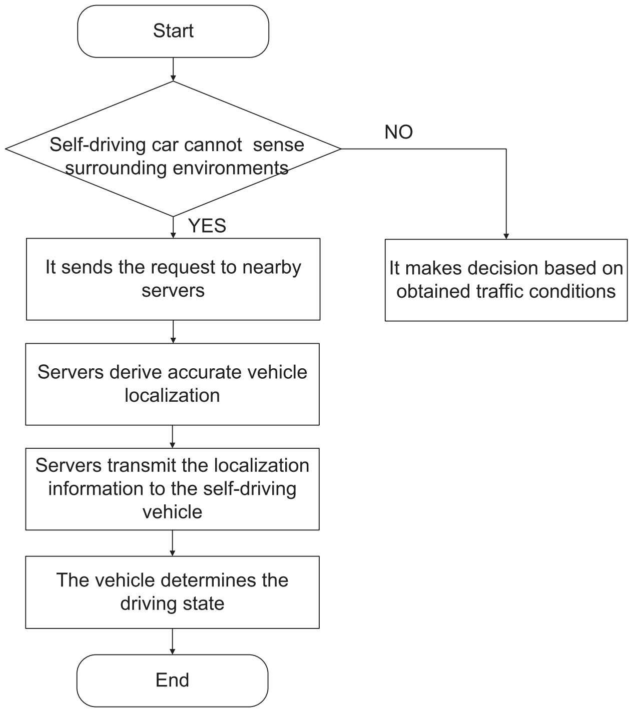
Start
Self-driving car cannot sense surrounding environments
NO
YES
It sends the request to nearby servers
It makes decision based on obtained traffic conditions
Servers derive accurate vehicle localization
Servers transmit the localization information to the self-driving vehicle
The vehicle determines the driving state
End

Fig. 3. The workflow of ECASS.

designed information fusion algorithm. The self-driving vehicle will receive nearby vehicles' location information from these servers. Finally, it can determine whether to overtake, decelerate, or accelerate based on the obtained localization information.

We focus on the location measurement of vehicles on the server side, which is based on the fusion of GPS and wireless signal information transmitted from vehicles. Therefore, the time overhead can also be cut down since the vehicle location estimation is executed on the server side. In this paper, we call it the vehicle localization algorithm.

4.2. Design of the vehicle detection algorithm

To accurately locate vehicles in the blocked area for the self-driving vehicle, edge computing based on the roadside infrastructure is utilized. As shown in Fig. 4(a), the wireless antenna An_b on vehicle b communicates with two nearby servers B and C simultaneously. Vehicle b delivers its own GPS localization information to nearby servers B and C. Specifically, servers can harness the incoming wireless signals to derive the

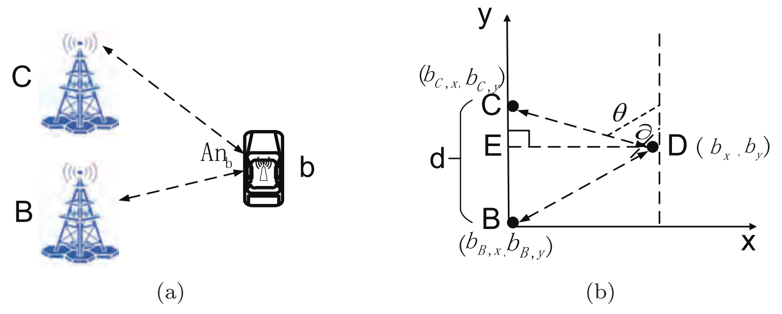

Fig. 4. Communication between the vehicle and servers.

AOA [34]. On the server side, the location and AOA information can be fused to improve the localization accuracy of vehicle b according to the designed localization algorithm. Meanwhile, servers will transmit their own location information and IDs to vehicle b, based on which it can realize the selection of nearby servers.

For ease of understanding, Fig. 4(a) can be abstracted into Fig. 4(b). In Fig. 4(b), points B and C represent server B and C, respectively. The wireless antenna An_b is placed at point D. The connection line between points B and C is set as the Y axis. And the line perpendicular to the line BC is set as the X axis. b_x and b_y represent the abscissa and ordinate of the antenna An_b. These localization information can be obtained from GPS embedded in the vehicle. Meanwhile, the coordinates of server B are denoted as $b_{B,x}$ and $b_{B,y}$, while $b_{C,x}$ and $b_{C,y}$ are the coordinates of server C. The distance between two servers is set to d.

We consider the scenario that the wireless antenna communicates with two nearby servers. Specifically, An_b transmits the GPS localization information b_x and b_y to servers B and C. Therefore, a geometric triangle $\triangle BCD$ can be established, as shown in Fig. 4(b). Assuming that the AOA from D to B is α and the AOA from D to C is θ according to the received wireless signals, then the following equation can be established.

$$d_{BD} \times \sin(\pi - \alpha) = d_{CD} \times \sin(\theta), \tag{2}$$

where d_{BD} represent the length of line segment BD, and d_{CD} denote the length of line segment CD. This equation can be formulated as below.

$$\begin{aligned}&\sqrt{(b_x - b_{B,x})^2 + (b_y - b_{B,y})^2} \times \sin(\pi - \alpha)\\&= \sqrt{(b_x - b_{C,x})^2 + (b_y - b_{C,y})^2} \times \sin(\theta).\end{aligned} \tag{3}$$

In the meantime, the following equation can also be established according to the law of cosines.

$$\begin{aligned}&2 \times \cos(\alpha - \theta) \times \sqrt{(b_x - b_{B,x})^2 + (b_y - b_{B,y})^2} \times\\&\sqrt{(b_x - b_{C,x})^2 + (b_y - b_{C,y})^2} = (b_x - b_{B,x})^2 + (b_y - b_{B,y})^2 +\\&(b_x - b_{C,x})^2 + (b_y - b_{C,y})^2 - d^2.\end{aligned} \tag{4}$$

Finally, we can also get another equation: The sum of the length of line segment BE and the length of line segment CE is equal to the length of line segment BC, which can be formulated as:

$$\begin{aligned}&\sqrt{(b_x - b_{C,x})^2 + (b_y - b_{b_{C,y}})^2} \times \cos(\theta) +\\&\sqrt{(b_x - b_{B,x})^2 + (b_y - b_{B,y})^2} \times \cos(\pi - \alpha) = d.\end{aligned} \tag{5}$$

These equations hold when the GPS based localization and AOAs are accurate. However, as mentioned above, even map-matching based GPS localization suffers from meters of errors. The error of measured AOAs are at a level of several degrees. Therefore, there exists a gap between the length of line segment BC and the sum of the length of line segment BE and the length of line segment CE. The aim of the designed localization algorithm in ECASS is to minimize the gap between these two values, which can be formulated as:

$$\begin{aligned}&\min\left(\sqrt{(b_x - b_{b_{C,x}})^2 + (b_y - b_{C,y})^2} \times \cos(\theta)\right.\\&\left.+\sqrt{(b_x - b_{B,x})^2 + (b_y - b_{B,y})^2} \times \cos(\pi - \alpha) - d\right).\end{aligned} \tag{6}$$

There exist some constraints on the optimization problem. Firstly, Eqs. 3 and 4 should be satisfied at the same time. Secondly, θ should be larger than 0, yet smaller than $\pi/2$. In conclusion, the optimization is related to the antenna An_b mounted on the vehicle b, and the optimization problem can be converted into:

$$\begin{aligned}&\min\left(\sqrt{(b_x - b_{B,x})^2 + (b_y - b_{B,y})^2} \times cos(\pi - \alpha) +\right.\\&\left.\sqrt{(b_x - b_{C,x})^2 + (b_y - b_{C,y})^2} \times cos(\theta) - d\right),\\&s.t.\\&2 \times cos(\alpha - \theta) \times \sqrt{(b_x - b_{b_{B,x}})^2 + (b_y - b_{B,y})^2} \times\\&\sqrt{(b_x - b_{C,x})^2 + (b_y - b_{C,y})^2} = (b_x - b_{B,x})^2 + (b_y - b_{B,y})^2 +\\&(b_x - b_{C,x})^2 + (b_y - b_{C,y})^2 - d^2,\\&\sqrt{(b_x - b_{B,x})^2 + (b_y - b_{B,y})^2} \times sin(\pi - \alpha)\\&= \sqrt{(b_x - b_{C,x})^2 + (b_y - b_{C,y})^2} \times sin(\theta),\\&0 \le \alpha - \theta \le \pi,\\&\pi/2 \le \alpha \le \pi,\\&0 \le \theta \le \pi/2,\\&d_{L,S} \le b_x \le d_{L,s} + B_w,\\&0 \le b_y,\end{aligned}$$

where B_w represents the width of the road, $d_{L,S}$ denotes the distance between the server and the road. For each request, the optimization operation is executed on the server side once, and then send the estimated vehicle location to the self-driving vehicle. Through collecting these localization information from nearby servers, the self-driving vehicle can determine its driving state based on the decision strategy.

4.3. The selection of nearby servers

The proposed system assumes that nearby servers are employed to assist the self-driving vehicle realize vehicle detection and localization, and they can communicate with each other. As designed in Section 4.2, two nearest servers on the same side are selected as the assistant infrastructure to locate the vehicle accurately. This is because the closer to the vehicle, the stronger wireless signal from the vehicle can be received by servers. Further, it contributes to more accurate AOA estimation.

As demonstrated in the framework, servers, within the communication coverage of self-driving vehicles, will send their information including their ID and localization to the self-driving vehicle. Subsequently, the self-driving vehicle will select two nearest servers on the same side to assist accurate vehicle localization. As shown in Fig. 5, the antenna An_a will select servers B and C since they are the two nearest servers when the self-driving vehicle is at position 1. After a period of time, when at position 2, servers C and D are selected. Detailed algorithm for the server selection are demonstrated in Algorithm 1 .

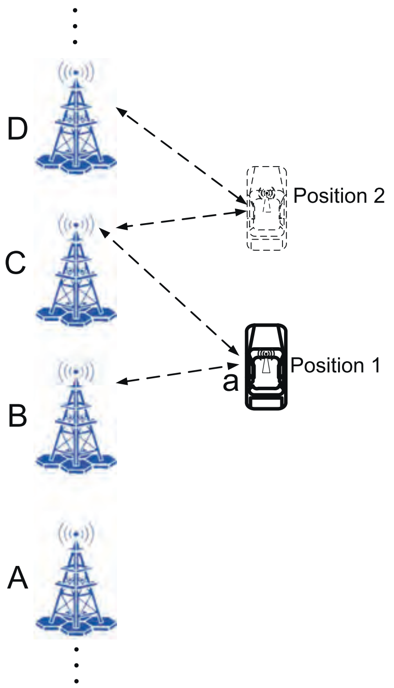

Fig. 5. The travelling process when passing through one server.

```
Algorithm 1: The server selection algorithm.
input : $(x_{v,a}, y_{v,a})$: the position of vehicle $a$; $S$: the set of servers
        that are in the communication range of the vehicle $a$; $N$:
        The number of servers in set $S$; $S_i$: server $i$; $(x_{s,i}, y_{s,i})$: the
        position of server $i$
output: $S_{pre}$: the set of communication base stations
1  $d_{\min} \leftarrow \sqrt{(x_{v,a} - x(s,1))^2 + (y_{v,a} - y(s,1))^2}$ ;
2  $S_{pre} \leftarrow \emptyset$;
3  $m \leftarrow 0$;
4  $k \leftarrow 0$;
5  while $(j < N)$ do
6      $d \leftarrow \sqrt{(x_{v,a} - x(s,j))^2 + (y_{v,a} - y(s,j))^2}$ ;
7      if $(d < d_{\min})$ then
8          $d_{\min} \leftarrow d$;
9          $k \leftarrow m$;
10         $m \leftarrow j$;
11     $j \leftarrow j + 1$;
12 $S_{pre} \leftarrow S_{pre} \cup S_m$;
13 $S_{pre} \leftarrow S_{pre} \cup S_k$;
14 return $S_{pre}$;
```

5. Performance evaluation

Extensive simulations are conducted to verify the high efficiency of ECASS.

5.1. Simulation settings

The simulations are developed by MATLAB. As shown in Fig. 6, there are two lanes in the same direction. Each lane width is set as 3 m. The distance between two adjacent servers along the road is set as 200m. The distance from the server to the nearest lane is set as 10 m. Vehicles travel along the X axis. Therefore, the coordinates of the first server can be set as (0,0), and the coordinates of the second server can be set as (200,0).

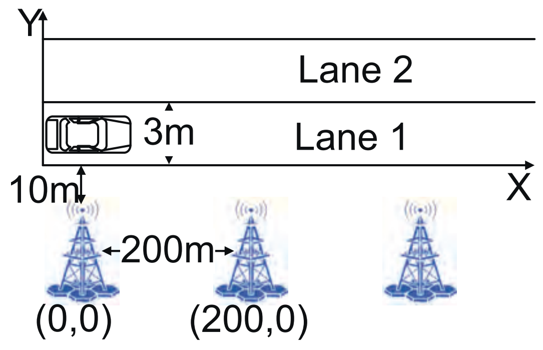

Fig. 6. The settings in the simulations.

5.2. Simulation results

In the simulation part, we evaluate ECASS with various velocities, different GPS errors, and different number of vehicles.

5.2.1. Simulation with various velocities

First of all, we mimic the scenario of only one vehicle travelling at a constant speed of 10 m/s. The vehicle moves in a straight line of lane 1. The GPS localization error is set within 2 m, and the angle estimation error based on wireless signals is set within 5°. According to the velocity, real AOA values of the antenna related to two nearby servers are equal to $arctan(11.5/10t)$ and $\pi - arctan(11.5/(200 - 10t))$, as shown in Fig. 7(a) and (b), respectively.

The simulation results about the trajectory tracking are shown in Fig. 8(a). Obviously, ECASS based trajectory tracking is closer to the vehicle real trajectory than GPS based trajectory.

The absolute error between ECASS based position and real position is equal to $\sqrt{(b_{t,x} - b_{m,x})^2 + (b_{t,y} - b_{m,y})^2}$ for vehicle b, and the absolute error between GPS position and real position is equal to $\sqrt{(b_{t,x} - b_x)^2 + (b_{t,y} - b_y)^2}$. These two kind of absolute errors are referred to as $E_{ECASS-Real}$ and $E_{GPS-Real}$, respectively.

The CDF of $E_{ECASS-Real}$ and $E_{GPS-Real}$ along the trajectory is plotted in Fig. 8(b). From this figure, it can be observed that the absolute localization errors based on ECASS are smaller than that based on GPS. When the GPS localization errors are set within 2 m, the mean absolute error between ECASS based position and real position is about 1 m, which is 0.4 m less than that between GPS based position and real position. Consequently, it verifies the high localization accuracy of the proposed algorithm.

With regard to the angle estimation, the simulation results are plotted in Fig. 9(a) and (b), for the AOA between the vehicle with the server behind, and the AOA between the vehicle and the front server, respectively. Both figures verify that ECASS based angles are much closer to the real angles and remain much stabler than the estimated angles based on wireless signals.

In the meantime, we also investigate the angle values based on the server selection scheme when passing through one server in the travelling process. The simulation results are shown in Fig. 10(a) and (b). It clearly demonstrates that the measured angles based on ECASS can track the real angle much more accurately compared to wireless signal based angles along the trajectory. Even when the vehicle passes through one server, ECASS based angle estimation is still much more accurate.

Above simulations are based on the constant velocity, which cannot be satisfied in most actual scenarios. Therefore, we investigate the scenario that the vehicle velocity changes over time. The set velocities are shown in Fig. 11(a) and (b), respectively, in which the highest speed of the vehicle is limited to 20 m/s. As shown in the left figure, the vehicle

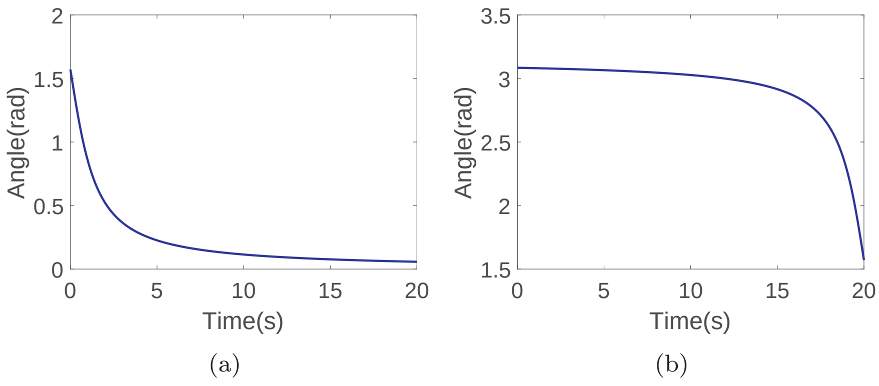

Fig. 7. The angle values when the vehicle follows the uniform motion in a straight line.

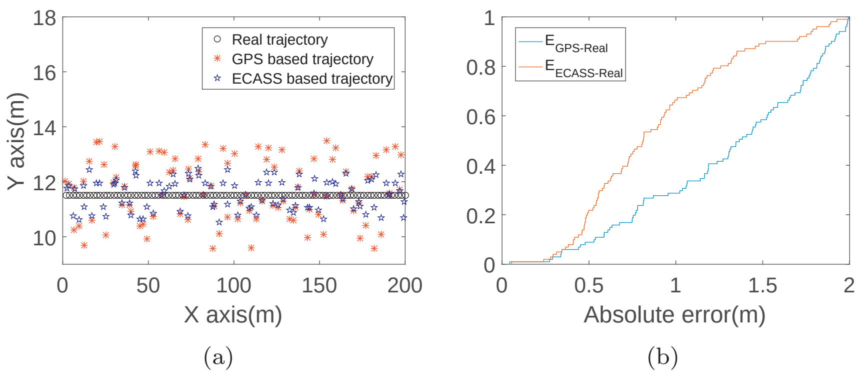

Fig. 8. The trajectory tracking results when the vehicle follows the uniform motion in a straight line.

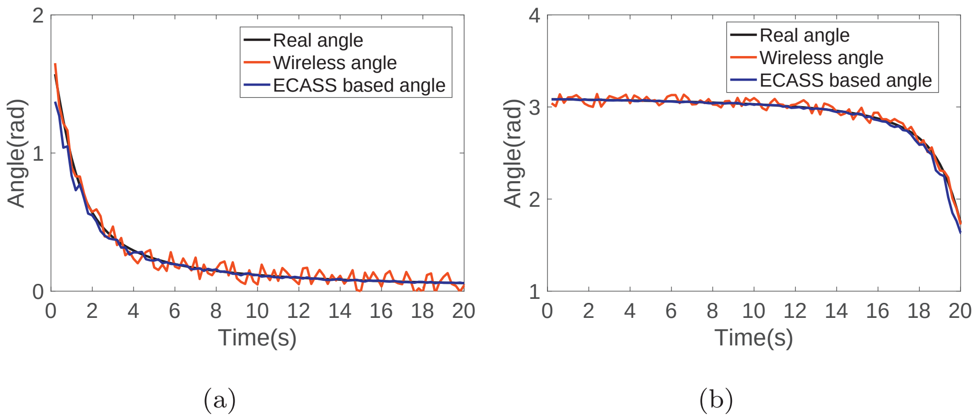

Fig. 9. The angle values when the vehicle follows the uniform motion in a straight line.

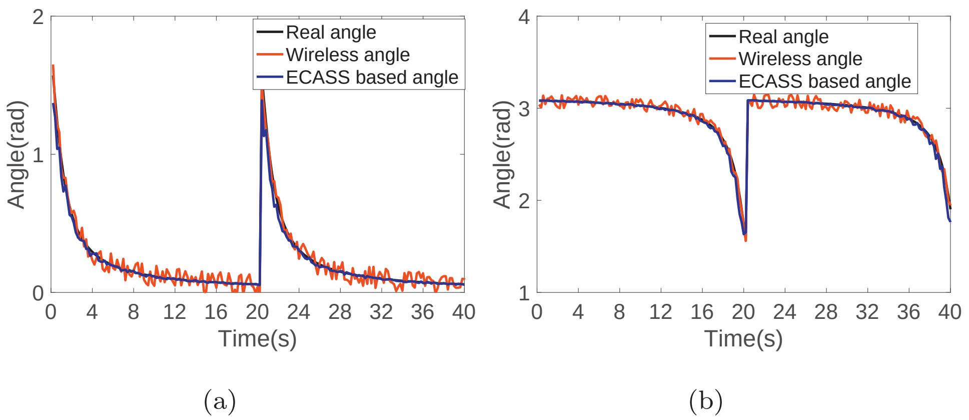

Fig. 10. The angle values when passing through one server.

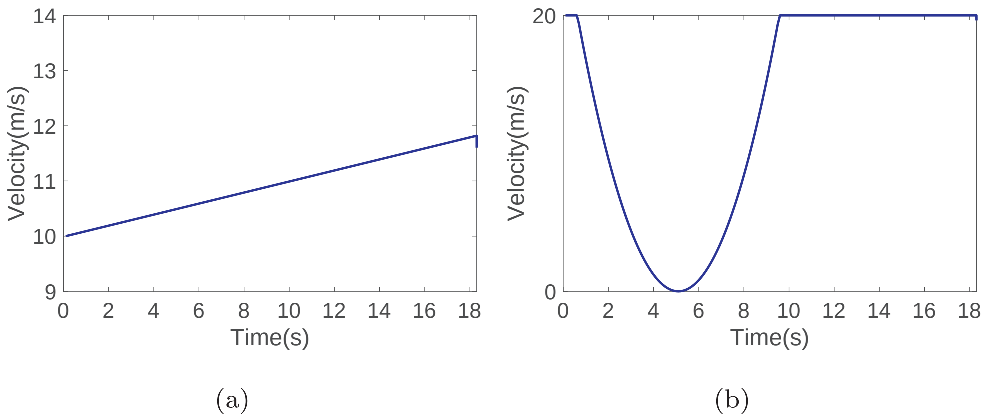

Fig. 11. The set velocity.

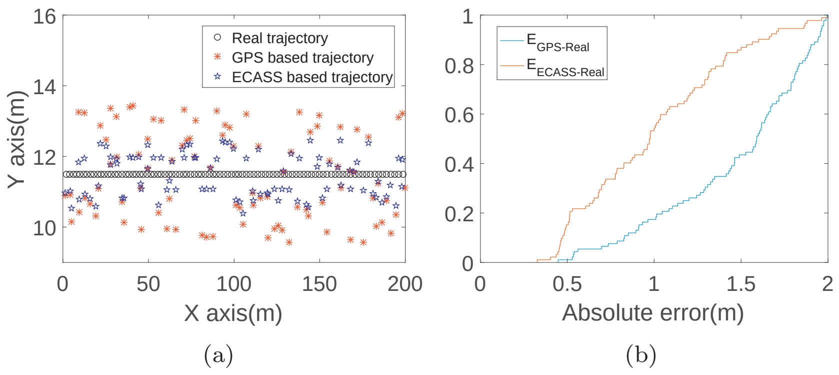

Fig. 12. The trajectory tracking when travelling with the uniform acceleration motion.

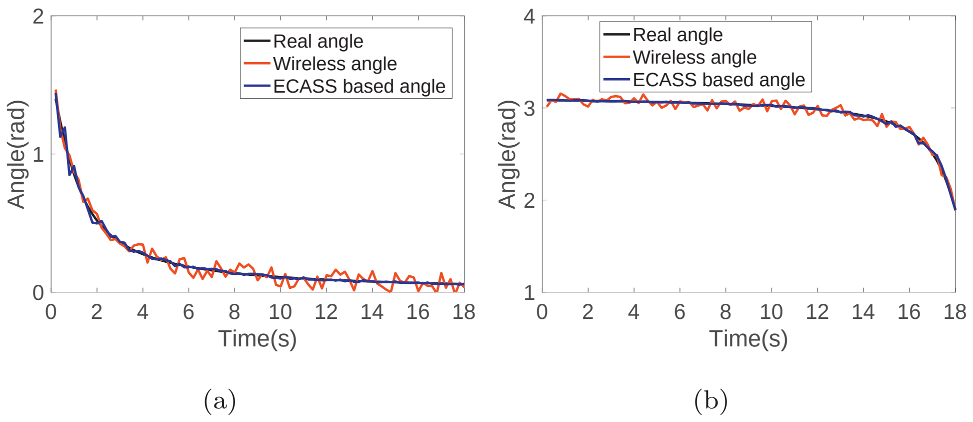

Fig. 13. The angle values when travelling with the uniform acceleration motion.

follows the uniform acceleration motion, and the velocity in the right figure is irregular, which is closer to realistic scenarios.

The simulation results are plotted in Fig. 12(a) and (b). ECASS based vehicle trajectory shows a superior performance in terms of both the trajectory tracking and absolute error compared to those based on GPS information. In the meantime, the mean absolute error based on ECASS remains about 0.4 m less than that based on GPS.

The angle estimation is shown in Fig. 13(a) and (b). We can observe that ECASS based angle measurements nearly match with the real angles, while angles estimated based on wireless signals fluctuate in a large range.

Finally, we carry out simulations based on the velocity set in Fig. 11(b). The simulation results shown in Fig. 14(a) and (b) verify that ECASS based trajectory is more accurate than GPS based trajectory even when the vehicle velocity is irregular.

Therefore, it can be concluded that when the GPS localization error is set within 2 m, ECASS based trajectory tracking shows a superior performance compared to GPS based trajectory despite of various vehicle speeds.

5.2.2. *Simulation with different GPS errors*

Above simulations are carried out with the GPS localization error set within 2 m. In this subsection, we will explore the simulation scenario with different GPS localization errors since the GPS localization accuracy can be influenced by different factors, such as the refraction effect caused by ionosphere and multipath effect.

Firstly, we set the GPS localization error within 6 m. Fig. 15(a) and (b) demonstrate the simulation results in terms of the localization accuracy. ECASS based trajectory is much closer to the real trajectory compared to that based on GPS. The mean absolute error is 2 m less than that based on GPS.

The angle measurements are shown in Fig. 16(a) and (b). Both two AOAs based on ECASS show an accurate estimation and stable perfor-

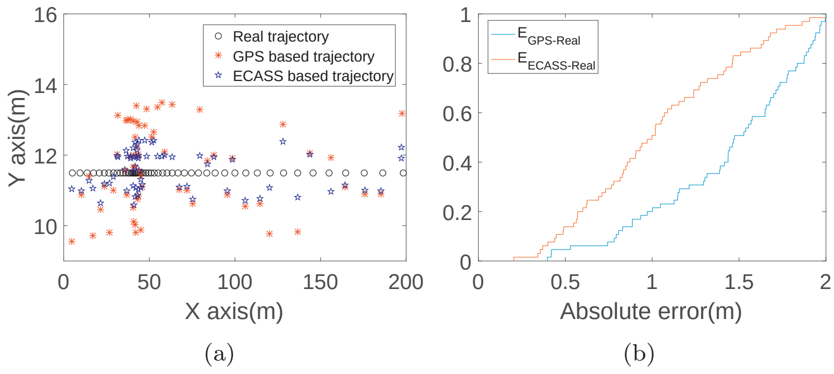

Fig. 14. The trajectory tracking when travelling with irregular velocities.

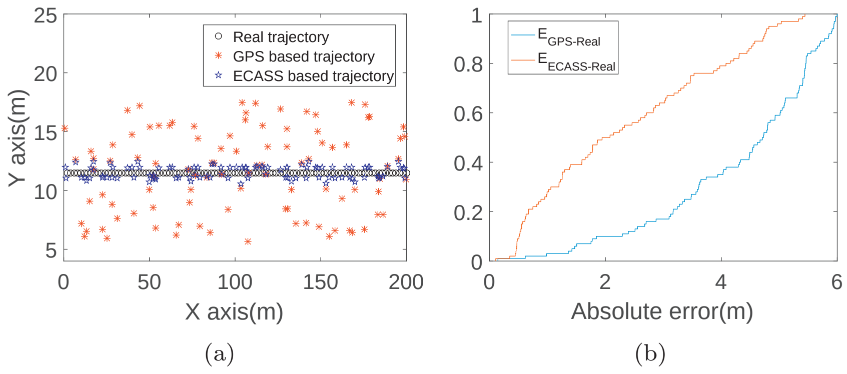

Fig. 15. The trajectory tracking when the GPS error is set within 6 m.

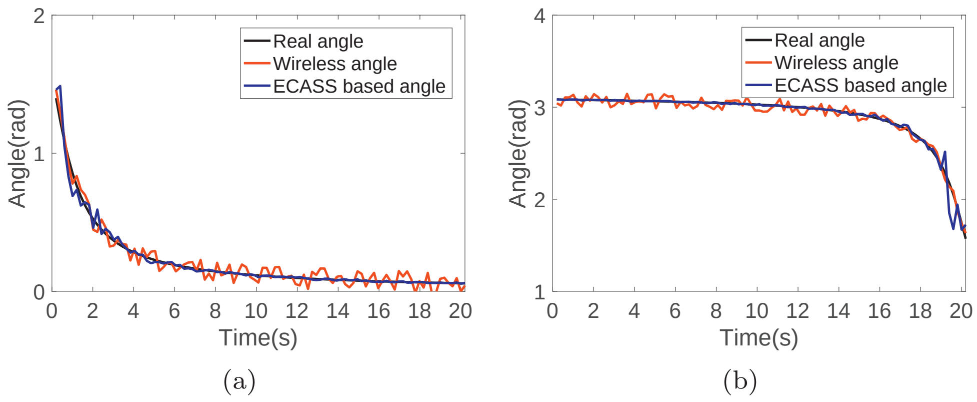

Fig. 16. The angle values when the GPS error is set within 6 m.

mance compared to the estimated AOAs based on wireless signals during the simulation time.

Finally, Fig. 17(a) and (b) plot the tracked trajectory when the GPS localization error is set within 10 m. The vehicle trajectory based on ECASS shows a much better performance than the trajectory based on GPS. The mean absolute error based on the designed algorithm is about 3 m, which is 4.1 m less than that based on GPS.

Finally, it is observed that the accuracy improvement based on ECASS can be enhanced with larger GPS localization errors. Although the mean absolute error based on the designed algorithm is about 3 m when the GPS error is set within 10 m. We can see that this relatively large error is mainly caused by the localization error in X axis. In convention, there should exist a large safety distance between two nearby vehicles. Therefore, these much smaller errors in comparison with the safety distance have a small influence on the driving state determination. Yet, the large GPS localization error will lead to wrong determinations. We can conclude that ECASS can work more efficiently when the GPS errors become larger.

5.2.3. Simulation with two vehicles

In the subsection, we will deploy two vehicles to demonstrate the high efficiency and robustness of ECASS. The velocities set for these two vehicles are shown in Fig. 11(a) and (b). In order to simulate more complex scenarios, vehicle 1 will change the lane in the 10 second with the initial location set in lane 2, while vehicle 2 will change the lane in the 4 second. The initial position of vehicle 1 is set as (0,0), and the initial position of vehicle 2 is set as (20,0).

The simulation results for vehicle 1 are plotted in Fig. 18(a) and (b). It clearly demonstrates that the trajectory based on ECASS performs much better than GPS based trajectory. It can be seen that the trajectory can also accurately track the real trajectory even when the vehicle changes its lane in the travelling process.

The simulation results of the estimated angle values under the scenario of two vehicles are shown in Fig. 19(a) and (b). It verifies that ECASS can achieve a high accuracy of AOA estimation even when the vehicle changes its lane.

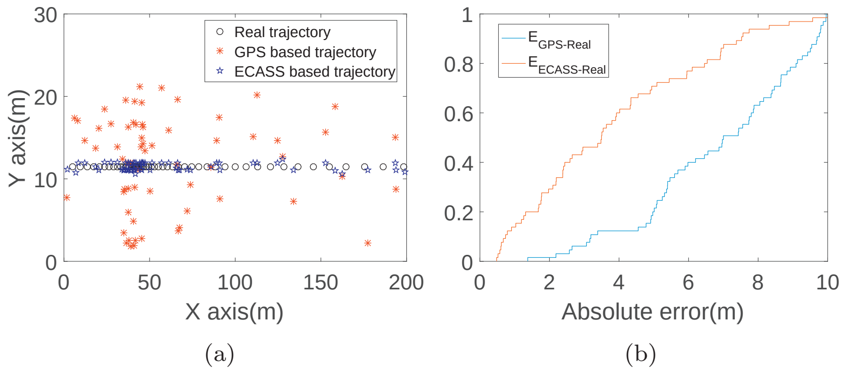

Fig. 17. The trajectory tracking when the GPS error is set within 10 m.

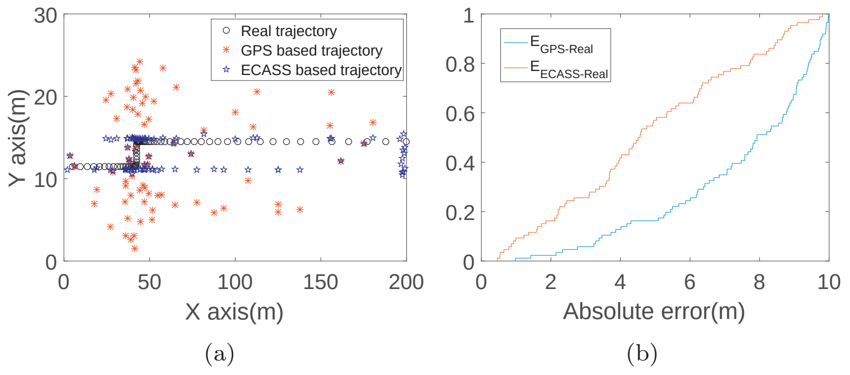

Fig. 18. The trajectory tracking when setting two vehicles.

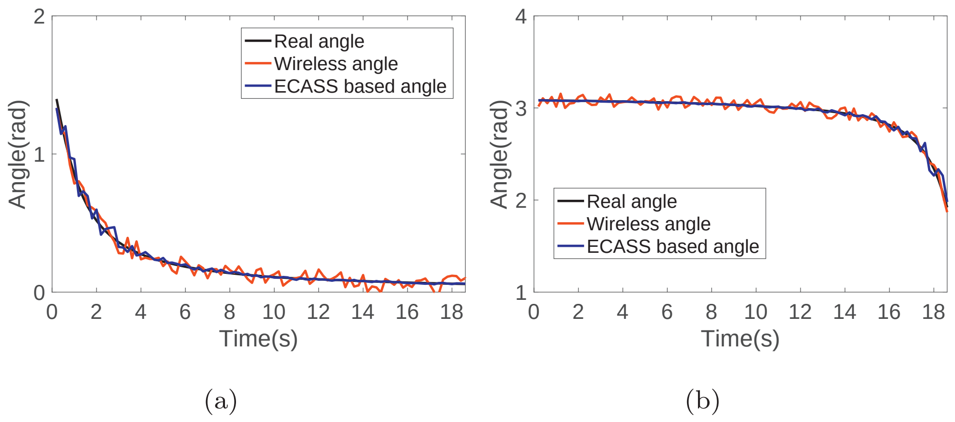

Fig. 19. The angle values when deploying two vehicles.

6. Conclusion and future work

With the rapid development of the autopilot technique, self-driving vehicles has attracted substantial attention from both industry and academia. Majority of available literature has focused on the hardware design such as vehicle borne radar and camera, algorithm design related to the information fusion, and vehicle trajectory planning. However, most trajectory planning algorithms are based on the assumption that radar, camera, and IMU can perceive the environment around the self-driving vehicle. Nevertheless, in real traffic scenes the vehicle may be blocked by the truck or bus ahead or behind, the radar or camera integrated within the vehicle cannot sense the surrounding environments accurately. In addition, GPS based localization accuracy cannot ensure the safety for automatic driving.

Therefore, we combine self-driving vehicles with edge computing to realize nearby vehicle detection and localization when self-driving vehicles are partially or even completely blocked by trucks or buses. Leveraging the roadside infrastructure, accurate vehicle localization near the self-driving vehicle can be achieved. These information will be delivered to the self-driving vehicle, based on which it can determine the driving state for the next moment. Extensive simulation results have verified the high efficiency of the proposed system.

Although we have proposed a novel framework using the roadside infrastructure to locate vehicles in the proximity of the self-driving vehicle, there exist some deficiencies, which are listed as below.

- Efficient communication strategies are necessary to cut down the transmission time consumption since the time cost plays a vital role in the reaction of autopilot system.
- In the future work, the vehicle localization algorithm should be designed in combination with the trajectory plan, in order to deal with various kinds of traffic conditions.
- Instead of simulations, real trace experiments should be carried out to test the designed scheme in actual scenarios.

Acknowledgments

This work is partly supported by National Key Research and Development Program grant 2016YFE0100600 and NSFC 61672349, 61672353, 61472252.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.sysarc.2019.02.014.

References

[1] G.H. Lee, F. Faundorfer, M. Pollefeys, Motion estimation for self-driving cars with a generalized camera, in: in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2746–2753.
[2] E. Ackerman, Lidar that will make self-driving cars affordable [news], IEEE Spectr. 53 (10) (2016) 14.
[3] J. Carlson, Mapping large, in: Urban Environments with GPS-aided Slam.
[4] H. Xiong, L.N. Boyle, Drivers' adaptation to adaptive cruise control: examination of automatic and manual braking, IEEE Trans. Intell. Transp. Syst. 13 (3) (2012) 1468–1473.
[5] J.V. Mierlo, G. Maggetto, E.V. de Burgwal, R. Gense, Driving style and traffic measures-influence on vehicle emissions and fuel consumption, Proc. Inst. Mech. Eng. Part D 218 (1) (2004) 43–50.
[6] R.W. Wolcott, R.M. Eustice, Visual localization within LIDAR maps for automated urban driving, in: in: International Conference on Intelligent Robots and Systems, IEEE, 2014, pp. 176–183.
[7] M.A. Brubaker, A. Geiger, R. Urtasun, Lost! leveraging the crowd for probabilistic visual self-localization, in: in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3057–3064.
[8] J. Xing, H. Dai, Z. Yu, A distributed multi-level model with dynamic replacement for the storage of smart edge computing, J. Syst. Archit. 83 (2018) 1–11.
[9] P. Petrov, F. Nashashibi, Modeling and nonlinear adaptive control for autonomous vehicle overtaking, IEEE Trans. Intell. Transp. Syst. 15 (4) (2014) 1643–1656.
[10] C.M. Sosa-Reyna, E. Tello-Leal, D. Lara-Alabazares, Methodology for the model–driven development of service oriented IoT applications, J. Syst. Archit. 90 (2018) 15–22.
[11] A. Houenou, P. Bonnifait, V. Cherfaoui, W. Yao, Vehicle trajectory prediction based on motion model and maneuver recognition, in: in: International Conference on Intelligent Robots and Systems, IEEE, 2013, pp. 4363–4369.
[12] D.C.K. Ngai, N.H.C. Yung, A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers, IEEE Trans. Intell. Transp. Syst. 12 (2) (2011) 509–522.
[13] E. Coelingh, S. Solyom, All aboard the robotic road train, IEEE Spectr. 49(11).
[14] M.A. Quddus, W.Y. Ochieng, R.B. Noland, Current map-matching algorithms for transport applications: state-of-the art and future research directions, Transp. Res. Part C 15 (5) (2007) 312–328.
[15] N.M. Drawil, H.M. Amar, O.A. Basir, et al., GPS localization accuracy classification: a context-based approach, IEEE Trans. Intell. Transp. Syst. 14 (1) (2013) 262–273.
[16] A.K.S. Rajan, A. Feucht, L. Gamer, I. Smaili, et al., Hypervisor for consolidating real-time automotive control units: its procedure, implications and hidden pitfalls, J. Syst. Archit. 82 (2018) 37–48.
[17] T. Fanni, L. Li, T. Viitanen, C. Sau, R. Xie, F. Palumbo, L. Raffo, H. Huttunen, J. Takala, S.S. Bhattacharyya, Hardware design methodology using lightweight dataflow and its integration with low power techniques, J. Syst. Archit. 78 (2017) 15–29.
[18] P. Pelliccione, E. Knauss, R. Heldal, S.M. Ågren, P. Mallozzi, A. Alminger, D. Borgentun, Automotive architecture framework: the experience of volvo cars, J. Syst. Archit. 77 (2017) 83–100.
[19] P. Falcone, F. Borrelli, J. Asgari, H.E. Tseng, D. Hrovat, Predictive active steering control for autonomous vehicle systems, IEEE Trans. Control Syst. Technol. 15 (3) (2007) 566–580.
[20] Q. Li, N. Zheng, H. Cheng, Springrobot: a prototype autonomous vehicle and its algorithms for lane detection, IEEE Trans. Intell. Transp. Syst. 5 (4) (2004) 300–308.
[21] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C.G. Keller, et al., Making bertha drive-an autonomous journey on a historic route, IEEE Intell. Transp. Syst. Mag. 6 (2) (2014) 8–20.
[22] N. Jeong, H. Hwang, E.T. Matson, Evaluation of low-cost LiDAR sensor for application in indoor UAV navigation, in: in: Sensors Applications Symposium, IEEE, 2018, pp. 1–5.
[23] L. Meier, P. Tanskanen, L. Heng, G.H. Lee, F. Fraundorfer, M. Pollefeys, PIXHAWK: a micro aerial vehicle design for autonomous flight using onboard computer vision, Auton. Robots 33 (1–2) (2012) 21–39.
[24] J. Wei, J.M. Snider, J. Kim, J.M. Dolan, R. Rajkumar, B. Litkouhi, Towards a viable autonomous driving research platform, in: in: Intelligent Vehicles Symposium, IEEE, 2013, pp. 763–770.
[25] U. Franke, D. Pfeiffer, C. Rabe, C. Knoeppel, M. Enzweiler, F. Stein, R. Herrtwich, Making bertha see, in: in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2013, pp. 214–221.
[26] S.J. Park, T.Y. Kim, S.M. Kang, K.H. Koo, A novel signal processing technique for vehicle detection radar, in: in: MTT-S International Microwave Symposium Digest, Vol. 1, IEEE, 2003, pp. 607–610.
[27] G. Alessandretti, A. Broggi, P. Cerri, Vehicle and guard rail detection using radar and vision data fusion, IEEE Trans. Intell. Transp. Syst. 8 (1) (2007) 95–105.
[28] S.J. Anderson, S.C. Peters, T.E. Pilutti, K. Iagnemma, An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios, Int. J. Veh. Auton. Syst. 8 (2–4) (2010) 190–216.
[29] S. Glaser, B. Vanholme, S. Mammar, D. Gruyer, L. Nouveliere, Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction, IEEE Trans. Intell. Transp. Syst. 11 (3) (2010) 589–606.
[30] V. Milanés, D.F. Llorca, J. Villagrá, J. Pérez, C. Fernández, I. Parra, C. González, M.A. Sotelo, Intelligent automatic overtaking system using vision for vehicle detection, Expert Syst. Appl. 39 (3) (2012) 3362–3373.
[31] U. Lee, S. Yoon, H. Shim, P. Vasseur, C. Demonceaux, Local path planning in a complex environment for self-driving car, in: in: IEEE 4th Annual International Conference on Cyber Technology in Automation, Control, and Intelligent Systems, IEEE, 2014, pp. 445–450.
[32] S. Sardellitti, G. Scutari, S. Barbarossa, Joint optimization of radio and computational resources for multicell mobile-edge computing, IEEE Trans. Signal Inf. Process. Netw. 1 (2) (2015) 89–103.
[33] C. Yang, H.R. Shao, WiFi-based indoor positioning, IEEE Commun. Mag. 53 (3) (2015) 150–157.
[34] C. Chen, Y. Chen, Y. Han, H.Q. Lai, K.R. Liu, Achieving centimeter-accuracy indoor localization on wifi platforms: a frequency hopping approach, IEEE Internet Things J. 4 (1) (2017) 111–121.

Xiong Wang received the B.S. degree in electronic information engineering from the Wuhan University of Science and Technology in 2013 and the master's degree in information and communication engineering from the Huazhong University of Science and Technology in 2016. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. His research interests include wireless networks and mobile computing.

Tianpeng Wei is currently studying in Shanghai Jiao Tong University as a junior student. He majors in computer science.

Linghe Kong received the B.E. degree from Xidian University in 2005, the Dipl.-Ing. degree from TELECOM SudParis in 2007, and the Ph.D. degree from Shanghai Jiao Tong University in 2012. He was a Post-Doctoral Fellow with Columbia University and McGill University. He is currently a Research Professor with Shanghai Jiao Tong University, China. His research interests include wireless communications and mobile computing.

Liang He is an assistant professor at University of Colorado Denver. His research mainly focuses on cyber-physical systems, IoTs, and mobile computing. Before joining UCD, he worked as a research fellow at The University of Michigan at Ann Arbor, MI, USA, with Prof. Kang G. Shin, as a Research Scientist at Singapore University of Technology and Design, Singapore, with Dr. Yu (Jason) Gu, and as a research assistant at University of Victoria, Canada, with Prof. Jianping Pan. He is a senior member of IEEE and a member of ACM.

Fan Wu is a professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. He received the BS degree in Computer Science from Nanjing University, in 2004, and the PhD degree in Computer Science and Engineering from the State University of New York at Buffalo, in 2009. He has visited the University of Illinois at Urbana-Champaign (UIUC) as a Post Doc Research Associate. His research interests include wireless networking and mobile computing, algorithmic game theory and its applications, and privacy preservation. He has published more than 100 peer-reviewed papers in technical journals and conference proceedings.

Guihai Chen received the B.S. degree from Nanjing University in 1984, the M.E. degree from Southeast University in 1987, and the Ph.D. degree from the University of Hong Kong in 1997. He is a Distinguished Professor with Shanghai Jiaotong University, China. He had been invited as a Visiting Professor for many universities, including the Kyushu Institute of Technology, Japan, in 1998, the University of Queensland, Australia, in 2000, and Wayne State University, USA, from 2001 to 2003. He has a wide range of research interests with focus on sensor network, peer-to-peer computing, and high performance computer architecture.