2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Cooperative Perception for Autonomous Vehicle Control on the Road: Motivation and Experimental Results

Seong-Woo Kim, Zhuang Jie Chong, Baoxing Qin, Xiaotong Shen,
Zhuoqi Cheng, Wei Liu, and Marcelo H. Ang Jr.

Abstract— **In this paper, we attempt to develop a reusable framework of cooperative perception for vehicle control on the road that can extend perception range beyond line-of-sight and beyond field-of-view. For this goal, the following problems are addressed: map merging, vehicle identification, sensor multi-modality, impact of communications, and impact on path planning. We provide experimental results using a self-driving vehicle and manned vehicles equipped with the cooperative perception systems that we propose and implement.**

I. INTRODUCTION

Thanks to the recent technology advances, the performance, convenience, and safety of modern vehicles has been greatly improved [1]. Moreover, the maturity of this technology has reached a level enough to enable a fully autonomous vehicle to drive complying with urban traffic laws [2]. The distinct aspect of autonomous vehicles from normal intelligent vehicles is to be controlled by motion planning algorithms based on the information sensed by a machine not a human. Since decision making and actuator control are initiated by the sensing information, the quality and quantity of sensing information about environments play a key role for fast and safe vehicle control [3].

For good and sufficient sensing information, there are two primary approaches: long range and wide angle sensor equipment, and cooperative perception [4], [5]. The high performance sensors provide an immediate response sensing time or large area sensing capability, but whose price is prohibitively expensive for economic viability, and sensing area is limited by line-of-sight. One of the major alternatives is cooperative perception that can obtain far distance information by exchanging local information via communications. The advantageous aspects of the cooperative perception are as follows. Firstly, the sensing area can be extended to the boundary of networked vehicles. Secondly, the prices of sensors and radio devices are affordable. Lastly, beyond-line-of-sight sensing is possible depending on the network connectivity. It was shown that this cooperative sensing approach improves traffic flow and safety [6], [7].

S. Kim is with Singapore-MIT Alliance for Research and Technology, Singapore, sungwoo@smart.mit.edu.

Z. Chong, B. Qin, X. Shen, Z. Cheng, W. Liu, and M. H. Ang Jr. are with the Department of Mechanical Engineering, National University of Singapore, Singapore, {chongzj, baoxing.qin, shen_xiaotong, A0094123, liu_wei, mpeang}@nus.edu.sg.

This research was supported by the Future Urban Mobility project of the Singapore-MIT Alliance for Research and Technology (SMART) Center, with funding from Singapore's National Research Foundation.

In this paper, we aim at developing a reusable framework of cooperative perception that can extend perception range beyond line-of-sight and beyond field-of-view, which is applicable to autonomous vehicle control on the road. However, many practical challenges should be addressed and solved in advance such as map merging, vehicle identification, communication uncertainty, sensor multi-modality, and path planning. This paper provides our answers to overcome these problems through implementation and experiments on urban road using a self-driving vehicle and manned vehicles equipped with the cooperative perception systems that we propose. The primary contribution of this paper can be summarized as 1) our proposed system can provide an online multi-modality map to path planner for autonomous vehicle control on the road, which is merged from more than three driving vehicles, 2) this is verified through experiments on the road, and 3) we address practical problems that need to be investigated further.

The remainder of this paper is as follows. Section II presents the overall description of the proposed cooperative perception system, and our solving approaches to build the system. Section III provides experimental results. We conclude this work in Section IV.

II. METHODOLOGY

A. Overview

The key idea of the proposed cooperative perception system is to share local sensing information with other vehicles via communications. For this purpose, a number of sub-problems should be considered such as a) map merging problem, and b) vehicle identification problem. In addition, c) the impact of communications, d) the impact on path planning, and e) sensor multi-modality should be also considered. In this section, we present our methods to cope with a) and b). In Sec. III, c), d), and e) will be addressed with experimental results.

B. Map Merging

For the movement of an autonomous vehicle, the vehicle must have a map and know where it is located on the map, which are called spatial awareness. In case of unknown environment, map building and localization should be performed at the same time, which is called Simultaneous Localization and Map building (SLAM) [8]. For multi-vehicle spatial awareness, map merging is additionally necessary along with map building and localization [9]. To realize the map

978-1-4673-6357-0/13/$31.00 ©2013 IEEE

merging, we use an occupancy grid map defined as $g : \mathbb{R}^2 \rightarrow \mathbb{R}$, where the point $g(x, y)$ can be defined in several ways such as the height of obstacle at the point (x, y), or belief that the point is free. The occupancy grid map is typically used for spatial awareness framework of mobile robotics.

To merge two different maps into one, a coordinate transformation method is necessary to move a position on a coordinate to other coordinate. Figure 1(a) shows the concept of this coordinate transformation. The bounding box depicts the boundary of occupancy grid map. θ is a relative angle between two maps. A certain position (x, y) on the map of the preceding vehicle $i + 1$ can be transformed to the coordinate of ego vehicle i by the following matrix [10]:

$$T_{(x_{i+1},y_{i+1},\theta)}(x, y) = \begin{bmatrix} \cos\theta & -\sin\theta & x_{i+1} \\ \sin\theta & \cos\theta & y_{i+1} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (1)$$

where (x_{i+1}, y_{i+1}) corresponds to p_{i+1} in Fig. 1(a). Note that the key information to merge two different maps is the relative pose, i.e., position (x_{i+1}, y_{i+1}) and angle θ.

There have been several methods to obtain the relative pose between maps. However, without any common coordinate assumption between vehicles, this problem is not straightforward to solve [11]. Given an initial relative pose between vehicles, it is possible to fuse local and remote map by using the conventional Cooperative SLAM, as long as all observations and control inputs are shared among vehicles [12]. However, the assumption is quite limited in practice, particularly on the road.

An alternative approach is to find the relative pose that maximizes the overlap area between maps, which is a overlap matching method [10], [13]. In practice, the overlap matching method has also limitations to apply map merging on the road, because of time-invariant or unobvious matching features, and insufficient overlap area due to long safety gap for collision avoidance. In this work, we use a relative pose measurement based approach, whose principle can be essentially summarized as follows. Firstly, the initial relative pose is obtained by on-board range sensors. However, the value is not stable, particularly of the relative angle θ. To cope with the problem, we adjust the angle initially estimated by on-board sensors by using the tangential angle of desired path, which corresponds to p_l in Fig. 1(a). We will evaluate this method in Sec. III.B.

Including the relative pose estimation step, our whole process of map merging method consists of six steps: 1) Ego vehicle movement detection, 2) other vehicle detection, 3) vehicle identification, 4) vehicle pose estimation, 5) delay compensation, and 6) map merging. Fig. 1(b) shows the sequence of whole process. Among all these steps, we address the vehicle identification problem in more detail.

C. Vehicle Identification

Since wireless devices commercially available are typically omni-directional, it is difficult and challenging to know who sent a message that arrived at ego vehicle, particularly when no common coordinate is assumed such as a network

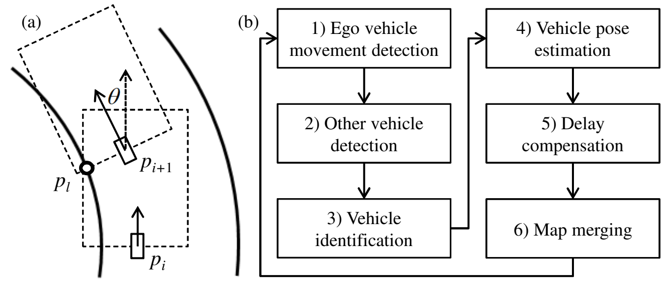


Fig. 1. To merge two different maps, the relative pose between them are necessary. p_l is a projection of p_{i+1} on desired path or lane.

camera behind the windshield. If more than one vehicles are detected at the same time, each vehicle should be identified before map merging. The principle to solve this problem is that common information between a sender and a receiver vehicle is necessary to match the senders of messages with recognized vehicles. In this work, we use a method using a speed as the common information.

A sender vehicle transmits 1) its unique network identity, and 2) its speed measured by its local odometry attached to sensing information. If ego vehicle is the successor of the sender vehicle, the speed of sender can be recognized by its on-board range sensor such as a laser scanner. Ego vehicle can recognize what the network identify of the preceding vehicle, by comparing the speed profiles between the remotely received and locally measured speed.

Let N_i be the set of neighboring vehicles of detected by on-board sensors of a vehicle i. $v_\tau^{j,i}$ is the speed of a vehicle j measured by ego vehicle i at time τ. Let $v_{t-w:t}^{j,i} = \{v_\tau^{j,i} : t - w < \tau \leq t\}$, where w is non-negative. r_τ^i is the speed included in the message arrived at ego vehicle i at time τ. Likewise, $r_{t-w:t}^i = \{r_\tau^i : t - w < \tau \leq t\}$. The vehicle that sent the message to ego vehicle i can be obtained by the following method:

$$j^*(i, t, w) = \arg\max_{j \in N_i} S(v_{t-w:t}^{j,i}, r_{t-w:t}^i), \quad (2)$$

where S is a function for measuring similarity between two input sequences, and t is current time. j^* is the estimated predecessor among neighboring vehicles N_i. j^* is typically a form of network identity for addressing. The control parameter w has a trade-off between false positive and response time. We evaluate this method in Sec. III.D.

D. Path Planer for Autonomous Driving

In this work, our self-driving vehicle follows the path generated by the path planning algorithm RRT* [14]. From the perspective of online path planning for autonomous driving, a path planner keeps finding a feasible path to arrive at a short-term destination, i.e., a waypoint, while moving toward the final destination [15]. The principle of path finding in RRT* is random sampling on the perceived spatial map. The path planner keeps looking for the less cost path until the next periodic deadline. This non-increasing cost of new solution is called as anytime characteristic, which is essential for time-constraint decision making on dynamic environments for

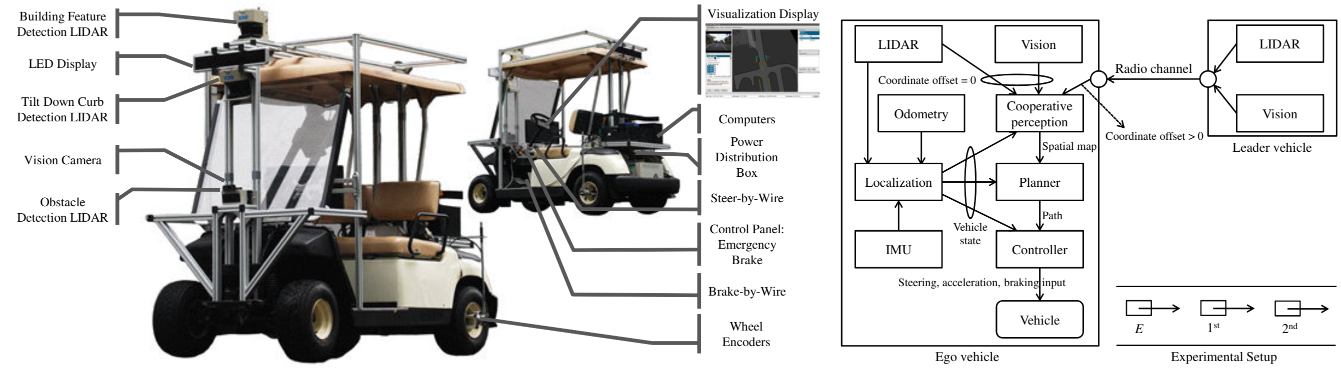


Fig. 2. (a) Self-driving robot used for all experiments. Since the robot development was initiated with the intention of economically viability, the robot was designed with minimal and off-the-shelf components, e.g., 2D LIDARs, webcam, two desktop computers, and no expensive INS-GPS navigation system. (b) System architecture of the proposed frameworks, and basic experimental setup consisting of one self-driving vehicle E as an ego vehicle, and two manned vehicles as leaders.

autonomous driving on the road. If no feasible path is found, the path planner initiates braking. In Sec. III.E, we show that the proposed system can provide an online map to path planner sufficiently enough to control the self-driving vehicle through real experiments on the road. From the next section, we evaluates all above-mentioned approaches considering sensor multi-modality and communications.

III. Experimental Results

A. System Architecture

We conducted all experiments with a self-driving vehicle in Figure 2(a), which is equipped with 2D LIDARs (LIght Detection And Ranging), a vision camera and wireless interface 802.11n. The software architecture of this system was established on Robot Operating System (ROS) suite and using only open source libraries. Detailed specifications are available in [16].

Figure 2(b) shows the overall system architecture and experimental setup. In our system, the sensing information delivered from remote vehicles is logically identical to the information from local sensors. The primary difference is only a coordinate offset. In case of remote information, the coordinate is shifted or rotated using the transformation matrix with respect to the relative pose between ego and remote vehicle.

To evaluate our proposal, we developed a cooperative perception (CP) package consisting of one 2D LIDAR, one webcam, Li-Po battery pack, and ROS-Ubuntu-based computer. Any vehicle with this CP package can participate in the cooperative perception. In this experiment, we used two CP packages for two manned vehicles. One type of vision camera was used for all three vehicle. In contrast, two kinds of LIDARs were used. The self-driving vehicle and the second leader were equipped with SICK LMS 291 LIDAR. Hokuyo UTM 30LX was used for the first leader, which enables to evaluate the characteristics of inter-vehicle sensor multi-modality.

In the system for this experiment, 2D LIDAR is mounted horizontally for vehicle detection and tracking. Importantly, the LIDAR is used to obtain the relative pose for map merging, as mentioned in Sec. II.B. A single laser scan is segmented into different pieces, and classification is performed on these pieces to find segment corresponding to vehicles. The classified vehicles are then approximated by line segments, from which vehicle poses can be calculated. To fuse vision images from different vehicles together, Inverse Perspective Mapping method (IPM) [17] was used, which can help to obtain a bird's eye view of the road surface.

The rightmost bottom of Fig. 2(b) shows the basic experimental setup with one self-driving vehicle E as an ego vehicle, and two manned vehicles as leaders. In the figure, the second leader transmits its sensing information to the first leader via wireless communications, while moving forward. The first leader merges the remote information with its local sensing information, and then transmits the merged information to the ego vehicle, while moving forward as well.

B. General Evaluation

To evaluate our proposal, let us bring the self-driving vehicle, and two manned vehicles equipped with CP packages on the road. Figure 3 shows the snapshots of occupancy grid map obtained from our map merging method using LIDAR at sharp curve and tortuous road. The red, green, and blue dots correspond to scan detection points of ego vehicle, the first, and the second leader, respectively. The sky-blue horizontal line is the scan results from tilt-down laser of for curb detection and collision avoidance [18]. This merged map was obtained online for path planner of the self-driving vehicle. Figure 9 shows another snapshots using our method along with various scenarios, particularly considering sensor multi-modality with vision sensors.

Let us evaluate the performance of the map merging methods first. The performance metric is the average position estimation error. A center position of lane was used as a reference value for performance metric. In general, the position is used for a desired path. However, note that the position is not a ground truth, which adds some errors that do not exist in practice. We have to compensate this error,

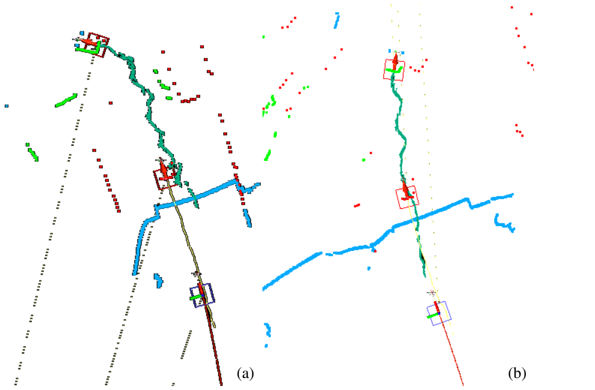


Fig. 3. Snapshots of occupancy grid map obtained online from the proposed map merging method using LIAR at (a) T-junction, and (b) Tortuous road. The horizontal sky-blue dots show the scan results from tilt-down laser for curb detection and collision avoidance.

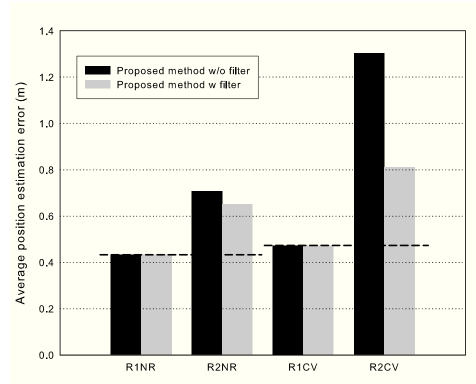


Fig. 4. Average position estimation error according to the algorithms. R1 and R2 represent the first and second leader, respectively. NR and CV represent the normal road and curve, respectively.

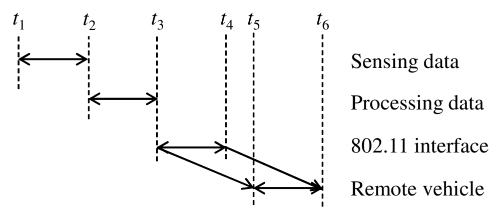


Fig. 5. Timing diagram of cooperative perception. For example, in case of LIDAR only, 2916 bytes/frame at 20 Hz. We measured $t_6 - t_3$ as communication delay. Note that $t_3 - t_2$ is a processing delay.

which is described later in detail.

Figure 4 shows measurement results. In Fig. 4, R1NR represents the first leader at the Normal Road that can be classified as a straight lane. Likewise, R2CV represents the second leader at CurVe. The position of the first leader can be estimated almost equally regardless of a selected method and road situation, because the position of the first leader is obtained from an on-board range sensor. However, each method performs differently from the second leader, particularly at curve.

Note that there are some errors of estimating the first leader in the figure, although the position of the first leader is obtained from an on-board sensor, because we used a center position of lane as a reference value. However, the leaders do not move exactly according to the position. Therefore, the position estimation errors of leaders have to be compensated by as much as this first leader's position error.

Finally, the average position estimation error of the second leader are 0.22 m at a normal lane, and 0.34 m at curve in our experiments, comparing to position estimation using an on-board sensor. Note that other techniques such as scan or image matching can be also used for map merging on the road scenarios. We will consider evaluating comparative performance in the next step.

C. Communication Impact

The wireless communication inherently has unpredictable delay. This delay is highly susceptible to the interference from others or environment. However, the others and environment cannot be fully controlled and predicted in practice. This delay unpredictability is a significant problem that directly affects the coordinate offset. To characterize this delay, we conducted six different experiments with the follow configurations: 1) laser only, 2) vision only with raw data, 3) compressed data, 4) processed data, 5) fusion with laser and compressed vision, and 6) with laser and processed vision.

To transmit our sensing data, we use standard message types in ROS. Firstly, our laser scan data is transmitted in the form of *LaserScan*. Each scan contains 721 laser beams. The total amount of information per frame is 2916 bytes/frame = 25 bytes (message header) + 7 bytes (message description) + 721 beams × float32 (4 bytes). The frame is transferred at 20 Hz. Secondly, there are three types of vision only cooperative perception with raw, compressed, and processed data. The raw data is a 640×360 (=230,400) bytes image without any post-processing. Thirdly, the compressed data is the result of PNG compression. It takes averagely 8.052 ms, minimally 2 ms, and maximally 14 ms to compress and decompress a 640×360 image on our implemented system. Fourthly, the processed data contains meta data such as lane information, which is represented by a cloud of point and transmitted as *PointCloud* in ROS. The size of the message varies depending on the extracted information. However, it is usually less than 5000 bytes, much smaller than the raw image. Lastly, we performed two additional experiments to evaluate sensor multi-modality with laser and compressed vision data, and laser and processed vision data.

Figure 6 shows the delay components over time. We measured $t_6 - t_3$ as a communication delay. Likewise, $t_3 - t_2$ is a processing delay. The communication and processing delay are directly affected by system design parameters such as data size, sensor type, communication protocol, or processing algorithm. We analyze the characteristics and impact of these delay components along with six configurations.

Table I shows the communication delay measurement results on the road with the self-driving vehicle, first and second leader via IEEE 802.11n wireless communications

TABLE I

DELAY MEASUREMENTS WITH THE SELF-DRIVING VEHICLE, FIRST AND SECOND LEADER VIA IEEE 802.11N.

	(ms)	Ego vehicle	1st to E	2nd to E
1) LIDAR	Average	0.226	4.390	13.574
	Stdev.	0.020	10.799	21.311
	Min.	0.119	0.986	5.798
	Max.	0.836	196.247	207.976
2) Vision (raw data)	Average	4.084	854.600	1643.238
	Stdev.	0.662	230.979	433.526
	Min.	2.524	390.037	1067.346
	Max.	6.012	2623.092	4223.861
3) Vision (Compressed)	Average	16.877	36.756	92.600
	Stdev.	3.030	72.580	132.562
	Min.	11.924	26.745	71.482
	Max.	23.850	1230.779	2904.557
4) Vision (Processed)	Average	0.227	17.127	24.562
	Stdev.	0.015	93.953	103.866
	Min.	0.158	1.064	5.638
	Max.	0.697	1109.996	1241.517
5) LIDAR + Vision (Compressed)	Average	18.696	116.935	159.123
	Stdev.	2.153	583.127	592.603
	Min.	12.165	26.085	71.601
	Max.	26.025	6998.621	8484.887
6) LIDAR + Vision (Processed)	Average	0.504	54.849	71.630
	Stdev.	0.028	323.612	429.999
	Min.	0.265	1.846	8.037
	Max.	0.685	3487.738	6172.385

according to the above six different sensor configurations. To obtain this data, we conducted separate experiments with experimental setup of Fig. 2(b). In Table I, Ego vehicle, 1st to E, and 2nd to E columns represent a internal processing delay, one-hop communication delay from the first leader to ego vehicle, two-hop communication delay from the second leader via the first leader to ego vehicle. In this work, we do not consider the case where the second leader directly communicates with ego vehicle. Hence, the scheme of opportunistic transmission can be considered to improve communication performance.

Note that this communication delay can be quantified and abstracted as the coordinate offset. Based on this measurement results, we can decide the proper coordinate offset value to compensate the communication delay. In this work, we use an explicit solution to compensate the delay impact by using real measurement data. This explicit solution can be online-processed fast.

Table II provides the selected table of coordinate offset with respect to the vehicle speed based on Table I. We used this table as a coordinate compensation function with respect to the odometry input of ego vehicle. From the perspective of average delay, the position error of one hop is 12 cm at 100 km/h in case of LIDAR only. The worst case position error becomes 5.45 m at 100 km/h. This is good enough to be used as control-purpose information, depending on the goal or task. However, communication delay becomes significantly

TABLE II

IMPACT OF COMMUNICATION DELAY ON POSITION ESTIMATION ERROR.

	(m)	20 km/h	40	60	100
LIDAR	Average	0.024	0.049	0.073	0.122
	Worst	1.090	2.181	3.271	5.451
Processed vision	Average	0.095	0.190	0.285	0.476
	Worst	6.167	12.333	18.499	30.833
LIDAR + P. Vision	Average	0.305	0.609	0.914	1.524
	Worst	19.372	38.744	58.117	96.861

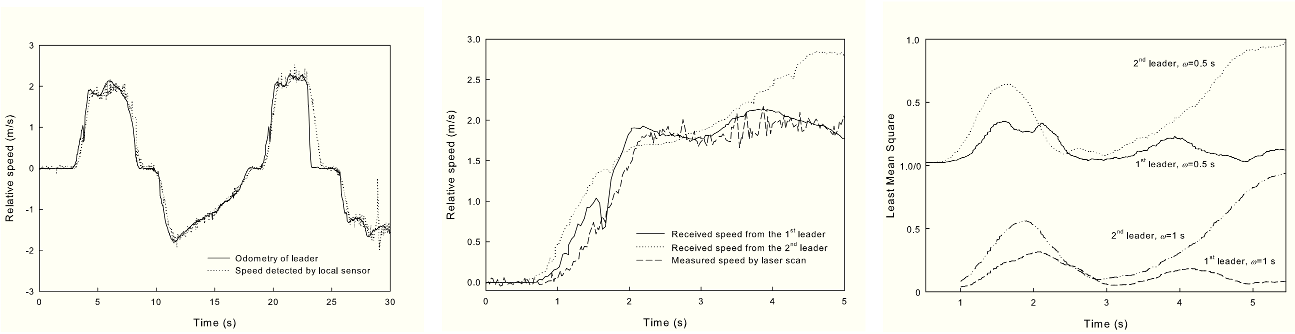
uncertain, as the size of data increases. For example, in case of fusion with LIDAR and processed vision, the position error of the worst case delay is 96.861 m at 100 km/h in Table II. Conclusively, sensor configuration must be determined depending on the application and its purpose. The worst case delay or loss of messages must be fully considered, and clearly dealt with a proper solution. For example, collision avoidance should be performed by local sensing information. Instead, the remote information could be utilized for long-term perspective path planning such as a decision problem between early lane changing and lane keeping [6].

D. Vehicle Identification

In Sec. II.C, we proposed the vehicle identification method based on matching the speed remotely delivered from other vehicles with the speed detected by local sensors of ego vehicle. To verify the feasibility of the proposal, we performed several experiments. Firstly, Figure 6 provides the received and measured speed in case of one leader and ego vehicle. In this experiment, a self-driving vehicle was used as a leader, because CP package has no odometry. For the same reason, one vehicle with CP package was used as ego vehicle.

In Fig. 6(a), the solid line shows the set of odometry values from other vehicles via wireless communications. With this data only, ego vehicle cannot know whether the vehicle is the immediate leader or not. The dotted line shows the measured speed using LIDAR. Ego vehicle definitely knows that the dotted value is the speed of the immediate leader by the method proposed in Sec. II.C. In this figure, two values are well matched with each other despite some errors caused by communication delay, or sensor disturbance. Average speed estimation error between the measured and remotely received speed is 0.175 m/s, which is accurate enough to be used for the purpose of map merging on the inter-vehicle cooperative perception framework.

Fig. 6(b) shows the scenario of Fig. 2(b). In other words, ego vehicle can receive the speed information from the first and second leader at the same time. In particular, we focus on a stop-to-run situation, one of the most typical situations triggering vehicle identification. In Fig. 2(b), a vehicle with CP package was used as ego vehicle, and the self-driving vehicle was used as the first leader. The remaining vehicle with CP package was used as the second leader. One difference is that the second leader was sending its speed already collected. For this purpose, we measured the odometry value once in this scenario before the experiment.



(a) (b) (c)

Fig. 6. (a) Single vehicle tracking based on vehicle speed, where the solid line is the speed delivered from a remote neighbor vehicle, and the dashed line is the speed measured by on-board LIDAR. Average speed estimation error between the measured and remotely received speed is 0.175 m/s. (b) Ego vehicle is receiving the speed from the first and second leader at the same time. Ego vehicle cannot know which value comes from the first leader. (c) Results using the same data of (b) with $w = 0.5$ s and 1 s.

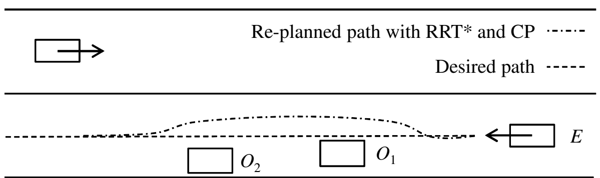


Fig. 7. Experiment scenario for on-line path planning with RRT* and cooperative perception.

Let us revisit Fig. 6(b). Ego vehicle cannot know which value comes from the first leader by using the raw measurement data. Fig. 6(c) shows the results using the measured speed of the first leader, and the equation in Sec. II.C with $w = 0.5$ s and 1 s. By using this, ego vehicle could identify that the messages of the solid line comes from the first leader. Note that there is a trade-off with respect to w. The longer w enables to track more accurately, but takes more time to obtain the first result. On the other hand, the shorter w can increase the false positive of vehicle identification. We notice that the extension to more than three vehicles converts this problem into a more complex. Scalability of the approach to the lager team will be evaluated in the next step.

E. Impact on Path Planning

The primary goal of experiments of this subsection is to show that the proposed system performs well on the self-driving vehicle while driving autonomously on urban road.

Figure 7 shows the experiment configuration to evaluate this goal. Based on the configuration, we performed experiments with four different scenarios. In the first scenario, an obstacle O_1 is located in front of ego vehicle E, which prevents E from following the desired path, i.e., the center of lane. As a result, path replanning is triggered to pass by the obstacle O_1. By using the newly generated path like the dash-dotted line in Fig. 7, the vehicle can pass by the obstacle.

Figure 8(a) shows one snapshot of path planning using RRT*. The big rectangle represents the local cost map of ego vehicle for RRT*, which is bounded by the range of local sensor of ego vehicle. In the local cost map, the blue area is the least cost area, namely, the lane where the vehicle can move. The purple area is higher cost area than the blue, because this is the opposite lane. The red area is infinite cost area where the vehicle cannot move to. The red arrow represents the next waypoint.

The next scenario is that O_1 is equipped with CP package, and thus O_1 can send its sensing information to E, i.e., with cooperative perception. Fig. 8(b) shows one snapshot of this scenario. The path planner can obtain candidate paths even beyond local sensing area, by random sampling with cooperative perception. The third scenario is that another obstacle O_2 is located in front of O_1. However, O_2 is located at closer to the walk road, which is not perceived by the local sensors of E. In other words, O_2 is a hidden obstacle. These experiments were performed with the self-driving vehicle at National University of Singapore (NUS) at night.

Fig. 8(c) and (d) are the snapshots of the experiments. Fig. 8(e) shows the RRT* path planning without cooperative perception, where the green line represents the newly confirmed path to the next waypoint. From the new waypoint, RRT* will choose a set of candidate paths via random sampling for the next short-term destination, which is represented by yellow lines. In (e), without realizing the existence of second obstacle or vehicle, path planner just generated the candidate paths with a few samples, and confirmed the next path quickly. Fig. 8(f) shows the same scenario with cooperative perception. In this experiment, O_1 took a role of the first leader with CP package. Compared to Fig. 8(e), the second obstacle is visible. Therefore, the path planner can know the existence of O_2, and path planing responds differently, as we can see the figure.

From the perspective of vehicle control, thanks to the extended sensing range through cooperative perception, earlier obstacle detection and traffic flow prediction are possible, which can give impacts on the result of path planning depending on the goal or strategy. Accordingly, the control inputs generated by path planner can also become different. New or improved path planing methods would be further investigated to fully utilize this longer and see-though view map provided by cooperative perception.

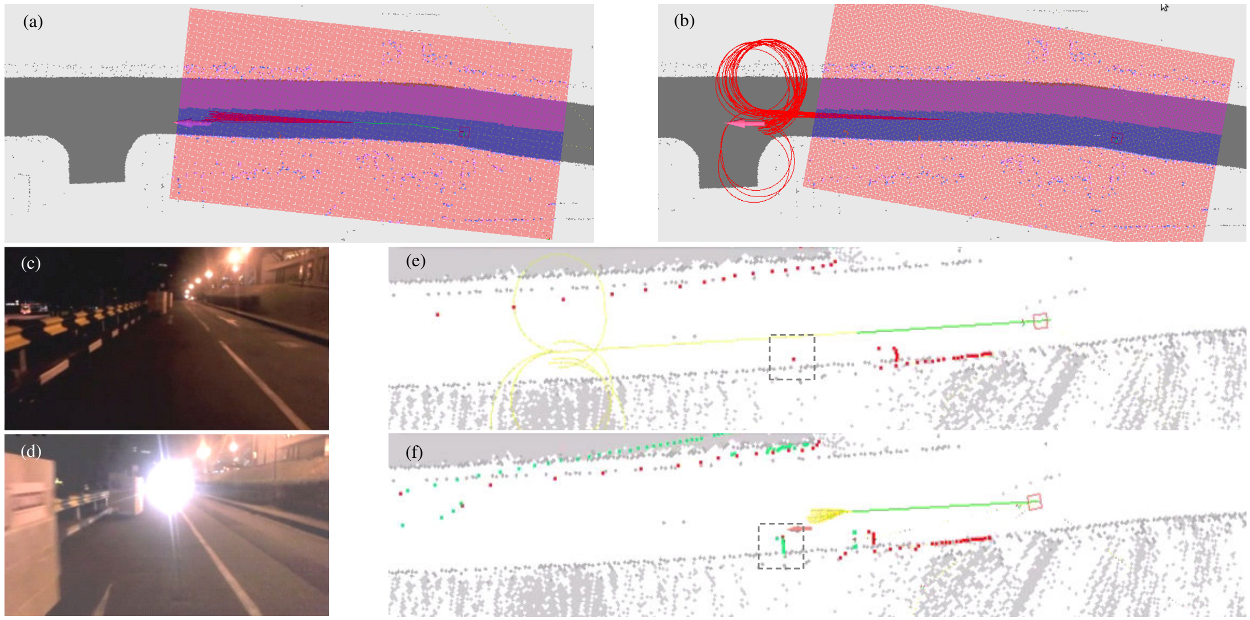


Fig. 8. (a) and (b) show the snapshots of RRT* path planning without and with cooperative perception, respectively. (c) shows the vision view of the self-driving vehicle at starting point. In (d), the vehicle is autonomously passing by the first obstacle according to the path replanning with cooperative perception. In the opposite lane, a car is coming. (e) and (f) are one snapshot of RRT* path planning during the autonomous driving with/without the cooperative perception, respectively. The yellow line represents the set of candidate paths that are chosen by random sampling.

IV. Conclusion and Future Work

In this paper, we proposed a cooperative perception framework applicable enough to control a self-driving vehicle. The proposed framework considers map merging problems, communication uncertainty, sensor multi-modality, vehicle identification, and path planning issues. To verify the feasibility and practicality of the framework, we conducted experiments with a self-driving vehicle as well as manned vehicles. Through experiments on the road, we demonstrated our framework performed well good enough to control a self-driving vehicle. Future works include comparative evaluation of map merging methods, scalability evaluation of vehicle identification, and quantification of impact on path planning.

References

[1] S.-W. Kim, E.-D. Lee, M.-E. Choi, H.-Y. Jeong, and S.-W. Seo, "Design optimization of vehicle control networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 7, pp. 3002–3016, Sep. 2011.

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

[3] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. S. Huang, S. Karaman, O. Koch, Y. Kuwata *et al.*, "A perception driven autonomous urban robot," *International Journal of Field Robotics*, vol. 25, no. 10, pp. 727–774, 2008.

[4] L. Merino, F. Caballero, J. M. de Dios, J. Ferruz, and A. Ollero, "A cooperative perception system for multiple UAVs: Application to automatic detection of forest fires," *Journal of Field Robotics*, vol. 23, no. 3-4, pp. 165–184, Apr. 2006.

[5] A. Rauch, F. Klanner, R. Rasshofer, and K. Dietmayer, "Car2X-based perception in a high-level fusion architecture for cooperative perception systems," in *IEEE Intelligent Vehicles Symposium*, June 2012, pp. 270–275.

[6] S.-W. Kim, G.-P. Gwon, S.-T. Choi, S.-N. Kang, M.-O. Shin, I.-S. Yoo, E.-D. Lee, and S.-W. Seo, "Multiple vehicle driving control for traffic flow efficiency," in *IEEE Intelligent Vehicles Symposium*, June 2012, pp. 462–468.

[7] B. van Arem, C. J. G. van Driel, and R. Visser, "The impact of cooperative adaptive cruise control on traffic-flow characteristics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 429–436, Dec. 2006.

[8] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99–110, 2006.

[9] K. Konolige, D. Fox, B. Limketkai, J. Ko, and B. Stewart, "Map merging for distributed robot navigation," in *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, vol. 1. IEEE, 2003, pp. 212–217.

[10] A. Birk and S. Carpin, "Merging occupancy grid maps from multiple robots," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1384–1397, 2006.

[11] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Putting the'I'in'Team': an ego-centric approach to cooperative localization," in *Robotics and Automation, IEEE International Conference on*, vol. 1. IEEE, 2003, pp. 868–874.

[12] X. Zhou and S. Roumeliotis, "Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case," in *Intelligent Robots and Systems, IEEE/RSJ International Conference on*. IEEE, 2006, pp. 1785–1792.

[13] G. Dedeoglu and G. Sukhatme, "Landmark-based matching algorithm for cooperative mapping by autonomous robots," in *Distributed autonomous robotic systems 4*. Citeseer, 2000.

[14] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the RRT*," in *Robotics and Automation, IEEE International Conference on*, May 2011.

[15] Y. Kuwata, G. Fiore, J. Teo, E. Frazzoli, and J. How, "Motion planning for urban driving using RRT," in *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, Sep. 2008, pp. 1681–1686.

[16] Z. J. Chong, B. Qin, T. Bandyopadhyay, T. Wongpiromsarn, B. Rebsamen, P. Dai, S.-W. Kim, M. H. Ang Jr., D. Hsu, D. Rus, and E. Frazzoli, "Autonomy for mobility on demand," in *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, Oct. 2012.

[17] H. A. Mallot, H. H. Bülthoff, J. Little, and S. Bohrer, "Inverse perspective mapping simplifies optical flow computation and obstacle detection," *Biological cybernetics*, vol. 64, no. 3, pp. 177–185, 1991.

[18] B. Qin, Z. Chong, T. Bandyopadhyay, M. H. Ang, E. Frazzoli, and D. Rus, "Curb-intersection feature based monte carlo localization on urban roads," in *Robotics and Automation, IEEE International Conference on*. IEEE, 2012, pp. 2640–2646.

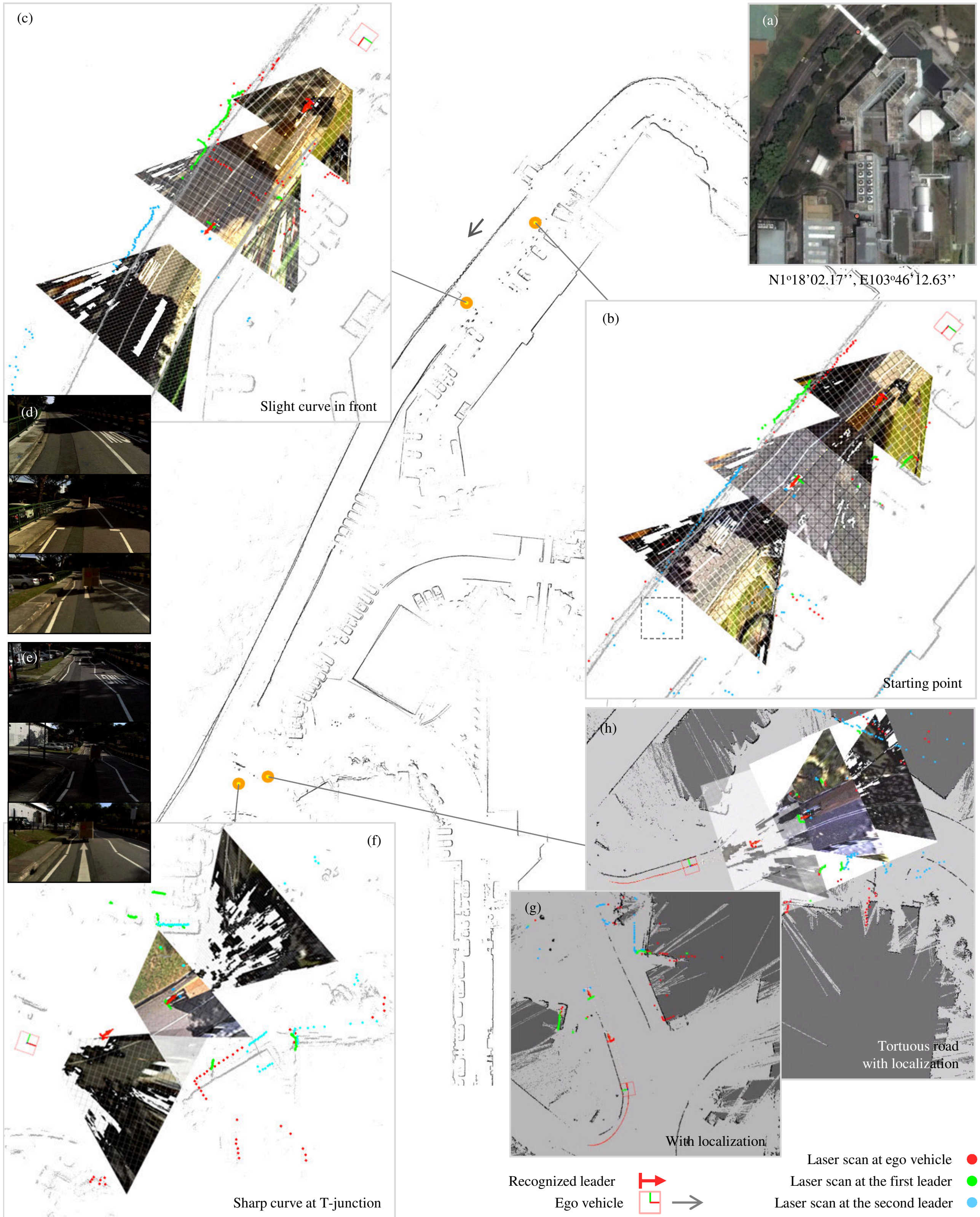


Fig. 9. Snapshots of map merging on the road, where the background is a map established by SLAM using LIDAR. (a) shows the sky view of the road from Google Map™. Ego vehicle is represented by the red box where the red line shows forward direction. In front of the red box, the thick red arrows shows the pose of leaders detected by LIDAR. The red, green, and blue dots show scanning results of SICK at ego vehicle, Hokuyo at the first leader, and SICK at the second leader. The road includes (b) straight lane at starting point, (c) slight curve, (f) sharp curve at T-junction, and (h) uphill and tortuous road. (d) and (e) are raw vision data of (c) and (b), respectively. Meta data extraction from vision is challenging due to numerous shadows on the road, because of many tress at the sidewalk and sidelight before sunset without any cloud. (g) shows laser scan and localization at T-junction. (h) shows laser scan, vision, and localization results at the same time in the uphill and tortuous road. In (b), note that an incoming car in the opposite lane is detected by the only second leader's LIDAR, not the first and ego vehicle. In (f), ego vehicle can see ahead traffic situations even beyond sensing angle at sharp curve.