

Integrated Planner for Autonomous Driving in Urban Environments Including Driving Intention Estimation



Hatem Darweesh

Abstract Thousands are killed every day in traffic accidents, and drivers are mostly to blame. Autonomous driving technology is the ultimate technological solution to this problem. There are still many unresolved problems with autonomous driving technology, such as navigating complex traffic situations. One of the reasons is detecting other drivers' intentions. Planning, which determines the movement of autonomous vehicles, is the cornerstone of autonomous agent navigation. Planning applications consist of multiple modules with different interfaces. Another challenge is the lack of open-source planning projects that allow cooperation between development teams globally. In this chapter, I will introduce two approaches to the planning problem. The first is developing of an open-source, integrated planner for autonomous navigation called Open Planner. It is composed of a global path planner, intention predictor, local path planner, and behavior planner. The second is a novel technique for estimating the intention and trajectory probabilities of surrounding vehicles, which enables long-term planning and reliable decision-making. Evaluation was achieved using simulation and field experimentation.

1 Introduction

There are several causes for the high number of traffic fatalities and injuries; rapid urbanization, low safety standards, insufficient law enforcement, people driving while distracted, fatigued, or under the influence of drugs or alcohol, speeding and failure to wear seat belts or helmets. Drivers are responsible for most of these accidents, as opposed to equipment failure or road hazards.

Many researchers are trying to tackle this problem by eliminating the human factor, using state-of-the-art technology to replace human drivers with a collection of sensors, hardware, and software, in order to achieve automated driving. Self-driving vehicles (autonomous vehicles) are close to becoming a reality. Interest in this field was sparked by the DARPA Challenge in 2005 [1] and DARPA Urban Challenge in

H. Darweesh (✉)

Nagoya University, Furo-cho, Chikusa-ku, Nagoya, Japan
e-mail: hatem.darweesh@g.sp.m.is.nagoya-u.ac.jp

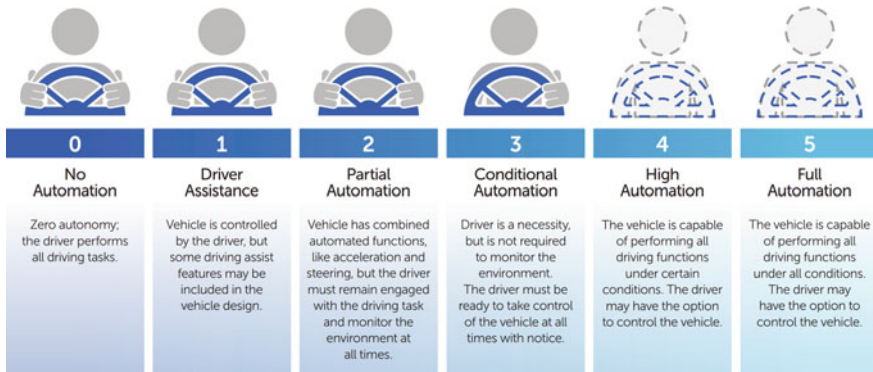


Fig. 1 Automation levels and milestones for autonomous driving, as defined by the Society of Automotive Engineers (SAE)

2007 [2]. However, after more than a decade of development, autonomous driving (AD) is still far from achieving its ultimate automation objectives. Figure 1 shows the milestones for autonomous driving, starting with the classical 100% human driver (no automation) to futuristic, 100% computer automated systems (full automation).¹

Autonomous driving is still a difficult problem due to the vast number of possible situations that can occur in dynamic driving environments. Here, I propose a solution to one of the most challenging autonomous driving tasks, which is `planning`. My research objective is to develop an open-source framework for planning which can achieve the following goals:

- Fosters international collaboration
- Provides a complete autonomous driving software package
- Integrates the various planning modules so that they are usable in most autonomous systems
- Supports a broad range of platforms, such as stand alone APIs and within ROS
- Supports a broad range of maps, so that the system will support standard and open-source map formats
- Supports a broad range of environments, such as indoors, outdoors, structured streets, and off-road
- Incorporates intention awareness

¹U.S. Department of Transportation, Automated Driving Systems 2.0, A Vision for Safety, <https://www.nhtsa.gov/>, [Online; accessed December 2019].

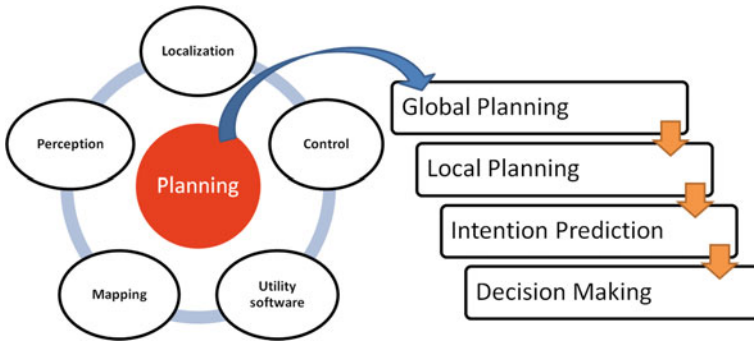


Fig. 2 Level 4+ autonomous driving software stack, showing the complexity of the planning part

1.1 Problem Definition

Planning consists of multiple modules as shown in Fig. 2. Integrating these together correctly is a challenging task.

Another challenge facing autonomous driving is the complex social interaction scenarios. In addition, location and culture difference pose additional challenges:

- In some countries, vehicles travel on the left side of the road, while in others on the right side
- Infrastructure differs from country to country and even from city to city
- Driving rules and habits differ
- Social interaction rules are different
- Traffic laws are different.

1.2 Proposed Solution

1.2.1 OpenPlanner: An Open-Source, Integrated Planner

The implementation of the open-source, integrated planner introduced in this work is called *OpenPlanner*. Its architecture is illustrated in Fig. 3. It includes a global planner that generates global reference paths using a vector (road network) map. The local planner then uses this global path to generate an obstacle-free, local trajectory from a set of sampled roll-outs. It uses various costs, such as collision, traffic rules, transition, and distance from center, to select the optimal trajectory. An intention and trajectory estimator calculates the probabilities associated with other vehicles’ intentions and trajectories, while the behavior generator uses predefined traffic rules and sensor data to function as a decision-maker.

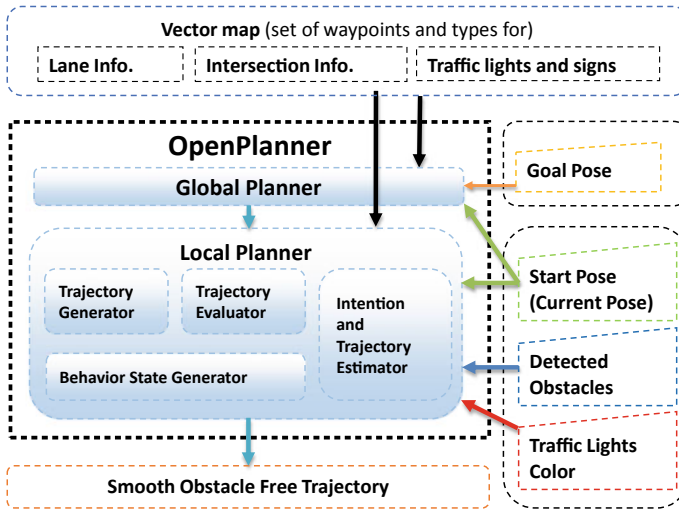


Fig. 3 Architecture of the proposed integrated planner

OpenPlanner has been used by many international research teams, which have made diverse and innovative contributions by applying it to a wide range of autonomous driving planning problems around the World, for example:

- **The Roboat project:** Using a fleet of autonomous boats [11] on Amsterdam's canals as transportation solution²
- **Autonomous driving map editor:** Road network map editor developed at the University of Konkuk, Korea [10]
- **ADAS Demo:** An Advanced Driver Assistance System (ADAS) demo for a major Hong Kong-based technology company

The integrated planner contributions could be seen in Table 1. It compares OpenPlanner to the top open-source planners currently available.

1.2.2 Trajectory and Intention Estimation

I have developed a novel method of estimating the probabilities of the intentions and trajectories of surrounding vehicles, using an existing behavior planner with a particle filter. This estimation process is a very important function of the planner, allowing it to handle complex traffic situations.

²Roboat Project, MIT and AMS, <https://roboat.org/>, [Online; accessed December 2019].

Table 1 Comparison of OpenPlanner with leading open-source planners

Feature	Planner					
	Open Motion Planning Library (OMPL)	ROS navigation stack	Open robotics design & Control Open-RDC	Mobile robot programming toolkit MRPT	Apollo AD planner	Open planner
Platform support	Robots	Robots	Robots	Robots	Autonomous vehicles	Robots + Autonomous vehicles
Environment support	Indoor	Indoor, sidewalk	Indoor, sidewalk	Indoor	Pre-driven structured roads	Indoor, outdoor, public road, highway
Library APIs	Independent	–	–	Independent	–	Independent
ROS support	Yes	Yes	Yes	Via bridge	–	Yes
Map support	Cost map + point cloud	Cost map	Cost map	Cost map + point cloud	Customized OpenDRIVE	Vector Map, KML, OpenDRIVE, Lanelet2
Global planning	Yes	–	–	–	Yes	Yes
Local planning	Yes	Yes	Yes	Yes	Yes	Yes
Behavior planning	Custom	–	–	–	Yes	Yes

1.3 Chapter Structure

In Sect. 2, the components of OpenPlanner will be explained in detail, with intention and trajectory estimation discussed at length in Sect. 3. The relationship between OpenPlanner and the real-world data circulation is explained in Sect. 4. Finally, in Sect. 5, this chapter is concluded and future work is introduced.

2 Integrated Planner for Autonomous Navigation

The open-source, integrated planner introduced in this section can be used for autonomous navigation of mobile robots in general, including autonomous driving applications. It is designed to use road network map items such as lanes, traffic lights, traffic signs, intersections, and stop lines, which is one of its main advantages over other open-source as shown in Table 1. In this section, different components of the planner and experimental results are introduced.

2.1 Global Planning

The global planner handles path routing. It takes the vector map, a start position and a goal position as input and then finds the shortest or lowest cost path using dynamic programming [9]. The global planner used by OpenPlanner can support complicated vector maps. Dynamic programming is used to find the optimal path from start to goal positions as in Fig. 4.

2.2 Trajectory Planning

A local trajectory planner is a set of functionality that generates a smooth trajectory which can be tracked by path-following algorithms, such as Pure Pursuit [3]. For OpenPlanner, Roll-out generation approach in Fig. 5 is adapted, in which the behavior generator can demand a re-plan at any time to generate fresh, smooth, roll-out trajectories. The sampled roll-outs are divided into three sections as shown in Fig. 6.

2.3 Behavior Planning

The behavior state generation module of OpenPlanner functions as the decision-maker of the system. It is a finite state machine in which each state represents a traffic situation. Transitions between states are controlled by intermediate parameters calculated using current traffic information and pre-programmed traffic rules. Figure 7 shows the currently available states in the OpenPlanner system.



Fig. 4 Searching the map for the shortest path. Line color indicates route cost. Cost is represented by distance; green color is the closest distance to Start and red color is the max distance reached at the Goal position

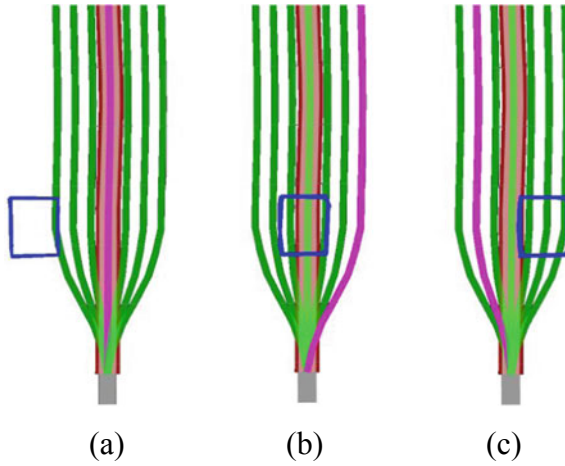


Fig. 5 Local Planner in action, in **a** the central trajectory is free, in **b** obstacle blocks the central trajectory so the most feasible one is the right-most trajectory, and in **c** the most feasible one is the second trajectory on the left

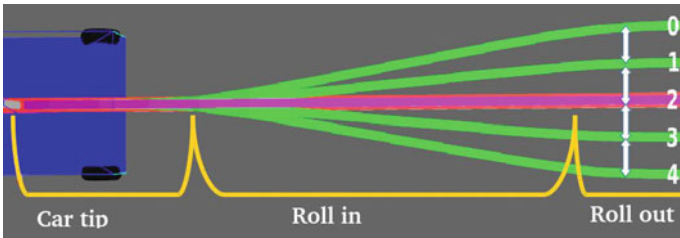


Fig. 6 Sections for generating roll outs

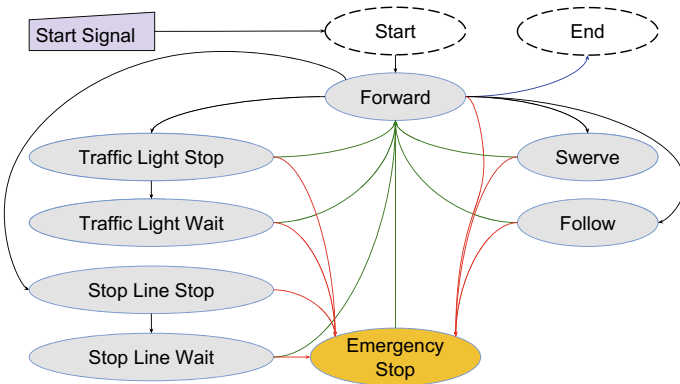


Fig. 7 Current system behavior states

2.4 Experiment Results

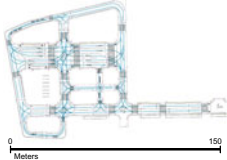
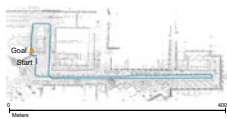
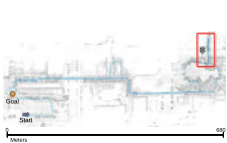
In this work, multiple robotics platforms are used, such as Ackerman-based steering robot, differential drive robot, and a real vehicle. These platforms are used in several experimental environments such as indoor, outdoor, structured public roads, and simulation environment. Table 2 shows the conducted experiments' maps and results.

2.5 Conclusion

The integrated planner was able to achieve the design objectives by successfully generating plans for multiple platforms in various dynamic environments. Also it targeted most of the open-source challenges we aimed for. Figure 8 shows the challenges and how OpenPlanner tackled these challenges.

As a result of providing this work as open-source, continuous feed back from the open-source community is received. There are diverse and innovative contributions

Table 2 Experimental results for proposed integrated planner

Experiment	Environment	Results
Simulation		<ul style="list-style-type: none"> • The planner can correctly generate trajectories and select a suitable behavior • The global planner was able to find the shortest path on a complex map
Nagoya University		<ul style="list-style-type: none"> • The planner is able to perform in real time at a minimum 10Hz, with an average of 100 objects detected • The planner is able to generate smooth trajectories and avoid obstacles even in a narrow passage • The planner is able to handle different traffic situations
Tsukuba Challenge		<ul style="list-style-type: none"> • The planner achieves automatic rerouting even when perception-based behavior is not available • The planner successfully navigates through a busy, unfamiliar environment

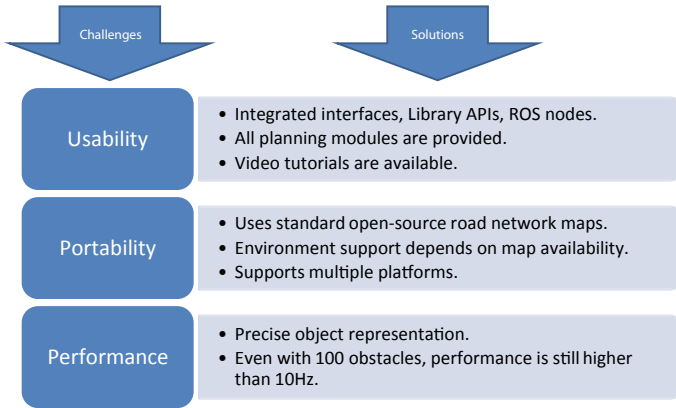


Fig. 8 Solution provided by OpenPlanner for the main challenges of developing an open-source planner for Autonomous Navigation systems

using OpenPlanner to a wide range of planning problems from around the World, for example:

- **The Roboat project:** MIT & AMS, Netherlands [11]
- **Autonomous driving open-source map editor:** Developed at the University of Konkuk, Korea [10]
- **Campus self driving carts:** University of California San Diego, USA [6]
- **ADAS Demo:** An Advanced Driver Assistance System (ADAS) demo, HKPC Hong Kong³

3 Behavior Planner Based Intention and Trajectory Estimation

Predicting with a high level of confidence what other vehicles are doing is essential for autonomous driving, and is one of the basic functions of a successful planner. Actions of other vehicles also include their probable intentions and future trajectories. The planner uses this information to generate suitable plans, i.e., ego-vehicle actions and trajectories.

The proposed solution is an intention and trajectory probability estimation algorithm, which utilizes a behavior planner [4] working in passive mode, wrapped in a multi-cue particle filter [7] for uncertainty modeling.

The main contribution is the development of a new method for estimating the intentions and trajectories of surrounding vehicles which can accurately handle most

³HKPC, HKPC web page, <https://www.hkpc.org/en/>, [Online; accessed December 2019].

Table 3 Issues with conventional particle filter estimation algorithms and contributions of the proposed estimation method by addressing these problems

Challenges using a conventional particle filter for estimation	Contributions of proposed approach
Difficult to model the complex motion parameters of surrounding vehicles	Behavior planner is used as the motion model
The state consists of both continuous and discrete variables	Multi-cue particle filter is used with variable confidence factors
Since the state dimensions are high, thousands of particles are needed to capture the posterior distribution	Separate particle filters are used for each intention and trajectory combination
Multi-cue particle filter uses joint distribution to aggregate the weights of different sensing cues, leading to particle deprivation	Weighted sum of different cues is used instead of the joint distribution

complex urban driving situations in real time. This is accomplished by using a behavior planner as the complex motion model, and integrating it with a non-parametric probabilistic filter (a multi-cue particle filter) to handle uncertainty. Table 3 highlights the main issues with the conventional approach and shows how these problems are solved.

3.1 *Solution Approach*

Figure 9 shows the system architecture of the proposed intention and trajectory estimation system. The proposed algorithm consists of three main parts; a trajectory extractor, a passive behavior planner, and multi-cue particle filters.

3.2 *Passive Behavior Planner*

The passive behavior planner is a stripped-out version of the integrated planner described in Sect. 2. The integrated planner is so flexible that unnecessary modules such as the Global Planner, Intention Estimation, and Object Tracking could be eliminated. It is called a passive planner because no feedback is available, thus the control signal does not have a direct impact on the observed state, and it has become similar to an open-loop planner.

Figure 10a shows the estimated intention states. For each intention state in Fig. 10a, there should exist a behavior that models the intention inside the passive behavior planner, Fig. 10b. Multiple behaviors can model single intention and vice versa.

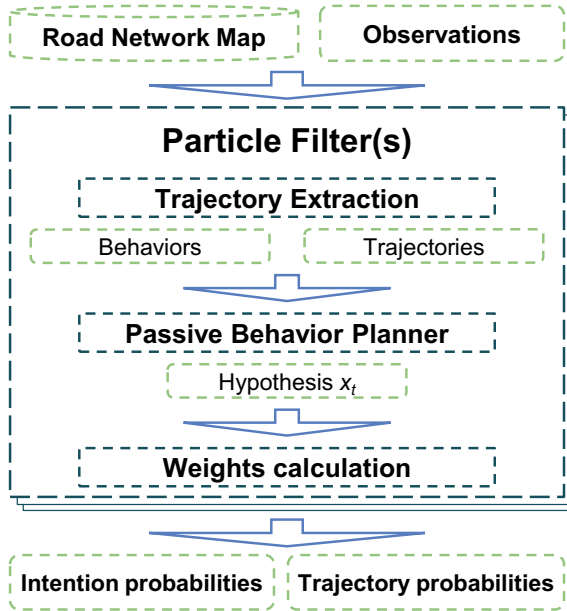


Fig. 9 Proposed intention and trajectory estimation system. Multi-cue particle filters use a passive behavior planner as a motion model

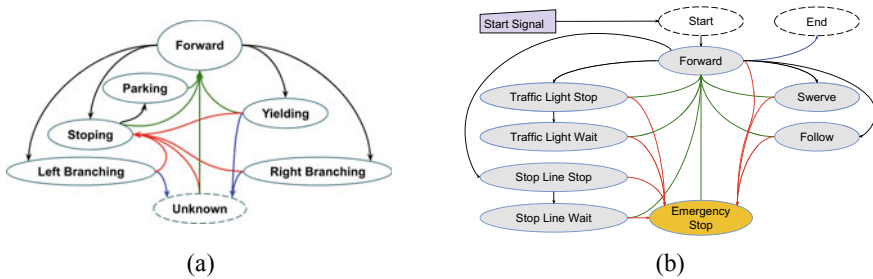


Fig. 10 Comparison of intentions to be estimated (a), to behaviors modeled by the passive behavior planner (b)

3.3 Uncertainty Modeling Using Particle Filter

The basic idea behind particle filtering is to approximate the belief state $b(x_t)$ at time t by a set of weighted samples χ_t as in Eq. (1) [5]. Here, x_t is the state vector, where z is the observation vector, χ_t^i means the i th sample of state x_t . Equation (1) enables computing the posterior probability using importance sampling. Because it is hard to sample from the target distribution, importance distribution $q(x)$ is used, then weighted according to Eq. (2).

$$P(x_t|z_{1:t}) \approx \sum_{i=1}^{N_s} w_t^i \delta(x_t, \chi_t^i) \quad (1)$$

$$w_t^i \propto \frac{P(x_{1:t}^i|z_{1:t})}{q(x_{1:t}^i|z_{1:t})} \quad (2)$$

In particle filter, the sample of a posterior distribution is called particles and denoted as χ in Eq. (3) with M the maximum number of particles. Here, each particle $x_t^{[m]}$ is a concrete instantiation of the state at time t .

$$\chi_t := x_t^{[1]}, x_t^{[1]}, \dots, x_t^{[m]} \quad (3)$$

Including the hypothesis x_t in the particle set χ_t requires that it is proportional to the Bayes filter posterior belief $b(x_t) = P(x_t|z_t, u_t)$ as in relation (4), where z is the observation and u is the control signal.

$$x_t^{[m]} \propto P(x_t|Z_t, u_t) \quad (4)$$

The particle importance factor (particle's weights) is denoted as $w_t^{[m]}$, which is the probability of the measurement z_t under the particle $x_t^{[m]}$ and is given by Eq. (5).

$$w_t^{[m]} = P(z_t|x_t^{[m]}) \quad (5)$$

Here, two main modifications are introduced to the original particle filter in [8]. The first one is the use of a Passive behavior planner as the state transition distribution, as in Eq. (6). The second is the Multi-cue particle filter to allow the use of several weak measurement cues to be accumulated into a strong estimator, as in Eq. (7).

$$x_t^{[m]} \simeq x(z_t, x_{t-1}, M) \quad (6)$$

$$w_t^{[m]} = p(z_t|x_t^{[m]}) = \alpha_p \cdot P(z_{p,t}|x_t^{[m]}) + \alpha_d \cdot p(z_{d,t}|x_t^{[m]}) + \alpha_v \cdot p(z_{v,t}|x_t^{[m]}) \\ + \alpha_a \cdot p(z_{a,t}|x_t^{[m]}) + \alpha_s \cdot p(z_{s,t}|x_t^{[m]}) \quad (7)$$

3.4 Estimation Algorithm

The proposed algorithm consists of three steps; trajectory extraction (initialization), particle sampling, and measurement update (weight calculation).

For the detected vehicle, all possible driving trajectories from the road network map is extracted. Then two additional trajectories to represent branching right and branching left are added.

The next step is the particle sampling to create hypothesis state x_t^m using the state transition $p(x_t | u_t, x_{t-1}^m)$. State transition probability follows the motion assumed for each particle which is represented by the passive behavior planner working as an expert driver.

Finally, in the weight calculation step, the sampled particles are filtered by calculating how far the hypothesis distribution is from the measurement sensing cues. The weight for each particle is calculated against the sensing data, such as position, direction, velocity, acceleration, and turn signal information using Eq. (8).

$$w_t^{[m]} = \alpha_p \cdot w_{p,t}^{[m]} + \alpha_d \cdot w_{d,t}^{[m]} + \alpha_v \cdot w_{v,t}^{[m]} + \alpha_a \cdot w_{a,t}^{[m]} + \alpha_s \cdot w_{s,t}^{[m]} \quad (8)$$

3.5 Evaluation Results

The proposed intention and trajectory estimation system was evaluated using multiple simulated driving situations. The objective of these evaluations is to demonstrate that the proposed method can accurately estimate trajectories and intention probabilities in various driving scenarios, such as three-way intersection, four-way intersection, intersection with and without stop signs, and bus stops. These driving scenarios are depicted from the National Highway Traffic Safety Administration (NHTSA) report.⁴


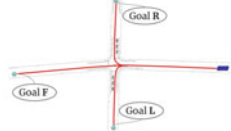
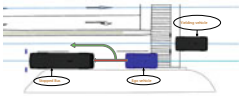
The results in Table 4 show that the proposed method has the ability to accurately discriminate between various possible intentions and trajectories in a variety of complex driving situations.

4 Real-World Data Circulation and Social Impact

The idea of RWDC is based on openly creating, organizing, and sharing data, which capture the problem enabling teams to explore more possibility offline, analyzing existing solutions, and developing new ones based on the openly shared data. After development, another data set is created and shared which helps bench-marking the problem's solution. Data circulation creates a modern and unique way for international collaboration to tackle the most challenging technological problems. The work introduced in this chapter fits perfectly to the definition and support the RWDC concept. Section 4.1 shows how the integrated planner (OpenPlanner) relates to the RWDC concept. Additional project developed under the supervision of the RWDC Leading program at Nagoya University is introduced in Sect. 4.2 as another example of the utilization of the data circulation concept.

⁴U.S. Department of Transportation, Pre-crash Scenario Typology for Crash Avoidance Research, <https://www.nhtsa.gov/>, [Online; accessed December 2019].

Table 4 Experimental results for intention and trajectory estimation

Experiment	Environment	Results
Three way intersection		<ul style="list-style-type: none"> • The estimator successfully assigns higher probabilities to the correct intention and trajectory for both Goal F and Goal L
Four way intersection		<ul style="list-style-type: none"> • The method shows successful estimation of intention and trajectory when there is sufficient sensing information • The method is tested in different situations, such as the vehicle stopping and not stopping at a stop line, and using or not using a turn signal • There was only one situation in which the estimator could not predict the trajectory, which was when the vehicle stopped at the stop line and no turn signal information was available. However, even the best human driver can't predict where the vehicle will go next in this situation
Bus stop (parking & yielding)		<ul style="list-style-type: none"> • The proposed method successfully estimates the parking intention of the bus and the intention of the vehicle in the other lane as to whether it would or would not yield. This is important because the planner needs to decide whether to wait behind the bus or to pass it

4.1 Autonomous Driving Planning

The introduced integrated planner utilizes the RWDC concept in two perspectives. First, data driven which is illustrated in Fig. 11. The second, as an open-source project, which is shown in Fig. 12.

The development of OpenPlanner contributed to the society as an open-source application. It has so far achieved the following:

Fig. 11 Relationship between OpenPlanner and RWDC

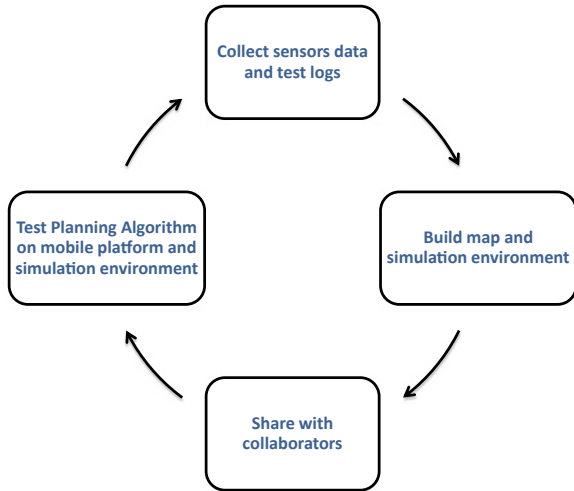
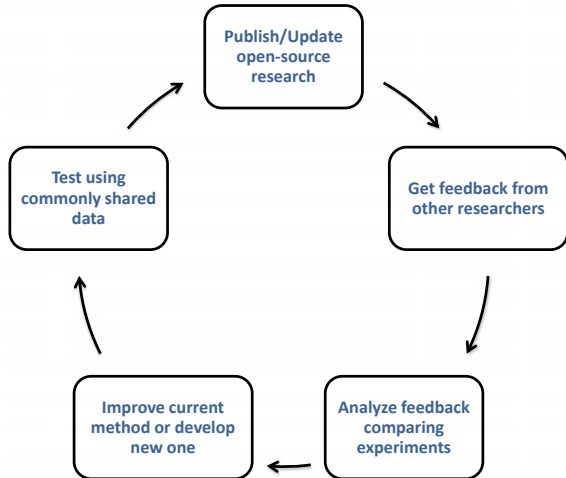


Fig. 12 OpenPlanner and other open-source code projects as examples of RWDC



- Hundreds of users, as well as feedback from the autonomous mobility community
- International collaboration with several teams in multiple countries, working on different goals
- Experiment and data sharing between development teams to improve the platform and create safer autonomous driving systems
- Use of the planner in multiple projects which directly improve daily life, such as the Roboat project.

4.2 Automated Road Network Mapping (ASSURE Maps)

Accurate road network maps are an essential part of reliable autonomous driving systems. ASSURE Maps project focused on developing a method of automatically building High Definition (HD) road network maps for autonomous vehicles. The goals of the ASSURE Maps project (and the origin of the project's name) were as follows:

- Accurate road network maps
- Secure cloud service
- Smart mapping tools
- Updated maps
- Rich details
- Evaluated results

The ASSURE Maps system consists of two main modules. The first module uses Automatic Map Generation (AMG) APIs, and the second uses Smart Mapping Tools (SMT), as shown in Fig. 13. ASSURE's AMG APIs function as the internal engine that loads data logs (LiDAR, camera images, GPS data and odometry) and extracts map semantic information. The smart mapping tools module functions as a review tool which helps users control the output of the AMG APIs and create data sets for the system's machine learning-based components.

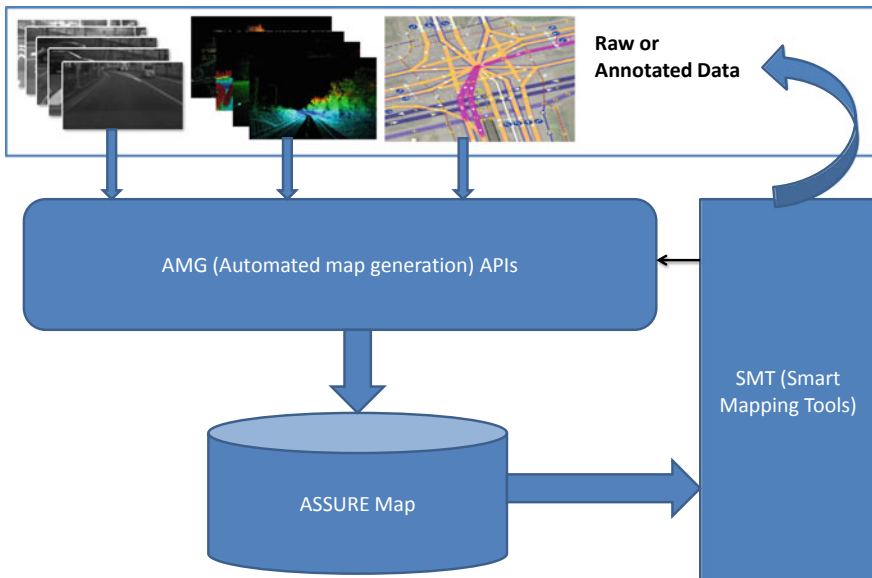


Fig. 13 ASSURE Maps Architecture

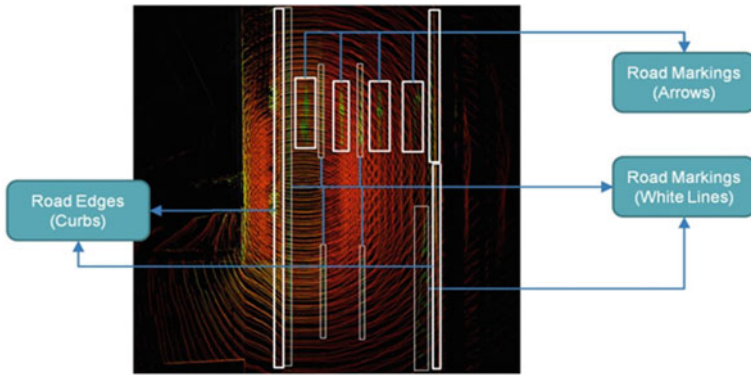


Fig. 14 LiDAR-based detection of the map data (curbs, markings, and lines)

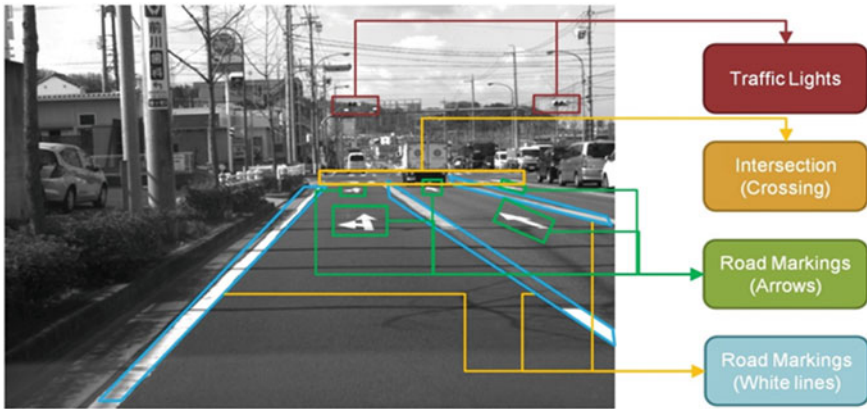


Fig. 15 Visual detection of map data (traffic lights, intersections, markings, and lines)

Experimental results in Fig. 14 show the extracted map items from the LiDAR data. The AMG APIs successfully detect curbs, lines, and markings. Figure 15 shows the detection of traffic light, intersections, markings, and lines using camera images only.

The relationship between ASSURE Maps and RWDC is twofold. The first is the data circulation that occurs at the core of the system development process. Figure 16 shows how data are utilized within the ASSURE mapping system. The second is based on the business model. Figure 17 shows the utilization of customer's data to improve the detection and mapping algorithms.

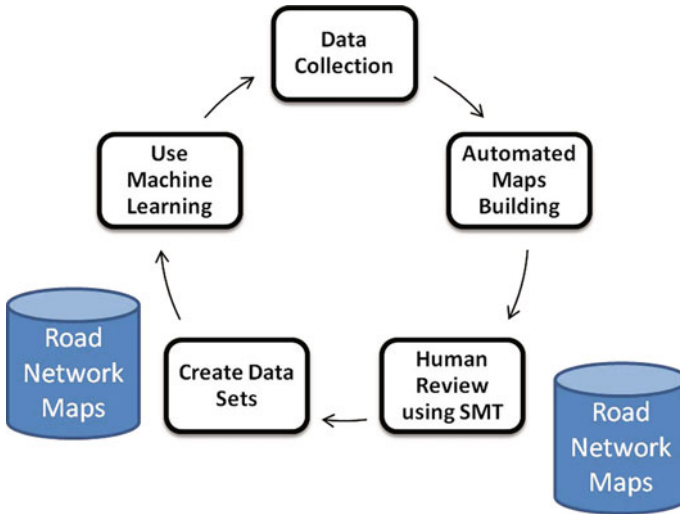


Fig. 16 Relationship between ASSURE maps and RWDC from a system development perspective

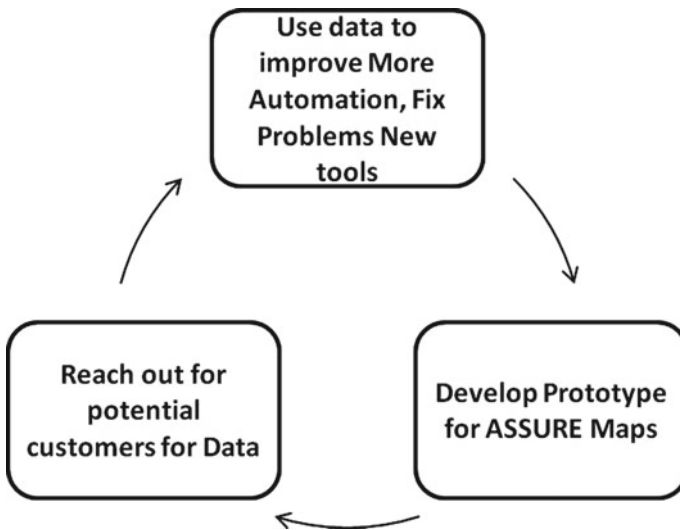


Fig. 17 Relationship between ASSURE Maps and RWDC from a business perspective

5 Conclusion and Future Work

In this chapter, the development of a complete, integrated, open-source planner for autonomous driving was introduced. The implementation of this planner, **OpenPlanner**, is open-source, so the robotics community can freely take advantage of it, use it, modify it, and build on it. OpenPlanner relies heavily on precisely created road network maps, which improve autonomous driving safety. Multiple experiments were conducted to show the planner functionalities. Continuous feedback from the open-source community indicates that OpenPlanner is useful not only for autonomous driving but also for a wide range of robotics applications.

In addition, intention and trajectory estimation method was introduced for predicting the actions of surrounding vehicles by associating a probability with each intention and trajectory. This is a very important step before decision-making in an autonomous driving system's planning process. A behavior planner is used to model the expected motion of surrounding vehicles, and particle filters are associated with each intention and trajectory. The results show that the proposed method has the ability to accurately discriminate between various possible intentions and trajectories in a variety of complex driving situations.

Achieving a speed of more than 30 km/h on a public road is one of our future goals. Another idea is to use MDP to calculate optimal motion actions rather than the current optimization method. Finally, for intention estimation filtering, using other probabilistic methods such as an HMM could improve performance dramatically.

Acknowledgments This work was supported by the OPERA project of the Japan Science and Technology Agency and by Nagoya University.

References

1. Buehler M, Iagnemma K, Singh S (2007) The 2005 DARPA grand challenge: the great robot race, 1st edn. Springer
2. Buehler M, Iagnemma K, Singh S (2009) The DARPA urban challenge: autonomous vehicles in city traffic, 1st edn. Springer
3. Coulter RC (1992) Implementation of the pure pursuit path tracking algorithm. Carnegie-Mellon University, Technical report
4. Darweesh H, Takeuchi E, Takeda K, Ninomiya Y, Sujiwo A, Morales LY, Akai N, Tomizawa T, Kato S (2017) Open source integrated planner for autonomous navigation in highly dynamic environments. *J Robot Mechatron* 29(4):668–684
5. Ghahramani Z (2001) An introduction to hidden Markov models and Bayesian networks. In: *Hidden Markov models: applications in computer vision*. World Scientific, pp 9–41
6. Paz D, Lai PJ, Harish S, Zhang H, Chan N, Hu C, Binnani S, Christensen H (2019) Lessons learned from deploying autonomous vehicles at UC San Diego. EasyChair Preprint no. 1295, EasyChair
7. Shen C, Van den Hengel A, Dick A (2003) Probabilistic multiple cue integration for particle filter based tracking. In: *Proceedings of VIIth digital image computing, techniques and applications*, pp 10–12
8. Thrun S, Burgard W, Fox D (2005) *Probabilistic robotics*. MIT Press

9. Thrun S, Montemerlo M, Dahlkamp H, Stavens D, Aron A, Diebel J, Fong P, Gale J, Halpenny M, Hoffmann G et al (2006) Stanley: the robot that won the DARPA grand challenge. *J Field Robot* 23(9):661–692
10. Tun WN, Kim S, Lee JW, Darweesh H (2019) Open-source tool of vector map for path planning in autoware autonomous driving software. In: *Proceedings of 2019 IEEE international conference on big data and smart computing*, pp 1–3
11. Wang W, Gheneti B, Mateos LA, Duarte F, Ratti C, Rus D (2019) Roboat: an autonomous surface vehicle for urban waterways. In: *Proceedings of 2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp 6340–6347