Behavioral Trajectory Planning for Motion Planning in Urban Environments

Wonteak Lim
Automotive Engineering
Hanyang University
Seoul, Korea
lwt1849@gmail.com

Seongjin Lee
Automotive Engineering
Hanyang University
Seoul, Korea
seongjin90@gmail.com

Kichun Jo
Driving Assistance Research
Valeo
Paris, France
kichun.jo@gmail.com

Myoungho Sunwoo
Automotive Engineering
Hanyang University
Seoul, Korea
msunwoo@hanyang.ac.kr

***Abstract*—This paper presents a behavioral trajectory planning for efficient motion planning in urban environments. For the autonomous driving in these environments, there are various complicated driving situations because of many combinations of road types, multiple lanes, traffic rules, and static and dynamic obstacles. Moreover, since these are changed dynamically, real-time trajectory planning is more intricate. To overcome this problem, we proposed behavioral trajectory planning which generates a macro-scale trajectory to define a spatiotemporal boundary of vehicle motion. This planner focuses on searching for a driving maneuver considering lane structure and traffic flow. As a result, this trajectory can reduce a searching area to generate a motion-level trajectory. To utilize this advantages, we apply a numerical optimization method for a local trajectory. The proposed algorithm was implemented and evaluated by urban scenarios.**

I. Introduction

Autonomous driving is the technology that makes cars understand driving environments and navigate own path without a driver's input. For collision-free driving in a dynamic environment, a car must predict motions of surrounding vehicles and determine own future trajectory, which is called *path planning* or *motion planning* [1], [2]. In this, behavior planning is a core module to select a driving maneuver and draw a blueprint of a vehicle motion to guide autonomous driving [3]. However, it is a complex task to determine a safe and comfortable one in urban environments because of various combinations of road types, traffic rules, and traffic participants (vehicles, bicycles, bikes, and pedestrians). In particular, the dynamic characteristics of them is a key factor to make the behavior planning problem more intricate [4]–[6].

To solve this problem, a state-machine based behavior planner [7]–[9] and a unified planner [10], [11] are widely used. The state-machine based method is a behavior planning algorithm to divide driving maneuvers into multiple independent states by semantic analysis. For example, the semantic level for urban driving includes a lane keeping, lane change, U-turn, and intersection driving. Each state has own local path planner for the particular driving purpose. Due to the states, the method can make the complicated problem of entire path planning into multiple simplified sub-problems. Thus, each local planner can focus on own purpose. However, it is vague to define individual states for a specific driving maneuver because most maneuvers contain other maneuver's motions. For example, a maneuver for a lane change includes a lane change motion as well as a lane keeping one before and after the lane change. Moreover, since this method must switch types of local path planner whenever to select a different behavior state, a discrete change of ego's trajectory can occur at that time. This dramatically decreases driving comfort and safety. To overcome these limitations, researchers used the unified path planning which does not separate the trajectory planning into two hierarchical modules (a behavior and a local planner). By using only an unified planning algorithm in various driving situations, the overall structure of the planning becomes simple. Moreover, there is not a transient region due to a change of behavior states. On the other hand, this algorithm requires extensive computation to search for the global optimal trajectory because it has to consider a highly coupled problem by mixing the complex driving elements which include vehicle dynamics, traffic rules, and surrounding object's movements.

For the efficient motion planning, behavior planning has to represent a more specific maneuver than the state-machine method and the complexity of it should be less than the unified planner's one. For these reasons, this paper proposed *behavioral trajectory planning* which provides a driving maneuver as a form of a trajectory. This is a macroscopic-scale trajectory to represent a specific maneuver using sequential sets of a spatiotemporal node and its boundary. Among the various urban driving considerations of road types, road geometry, traffic rules, participants, and vehicle dynamics, this planner focuses on the maneuver-related elements such as road types, traffic rules, and movements of other traffic participants. Based on this process, the behavioral trajectory defines a constrained spatiotemporal area to generate a motion-level trajectory. In other words, the enclosed space helps local planner quickly converge on the optimal point by reducing a variety of local optimal regions. The trajectory can be generated by considering the boundary of the behavioral trajectory and motion-related elements which are not strongly considered in the upper planner. Since this motion planning manages the driving elements by dividing the maneuver-related and the motion-related ones in a hierarchical architecture, the whole complexity of motion planning decreases. Furthermore,

978-1-5386-1526-3/17/$31.00 ©2017 IEEE

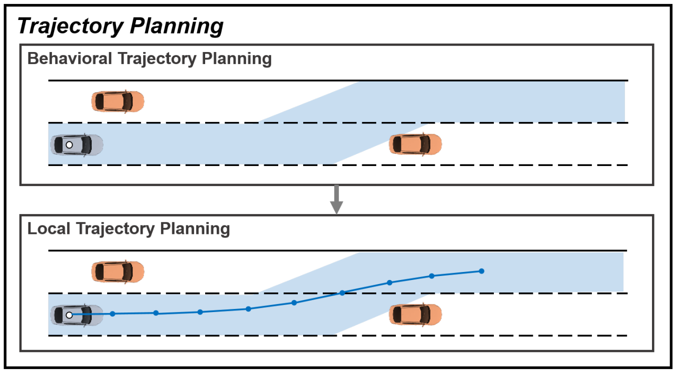


Fig. 1. The overall process of trajectory generation is divided into behavioral trajectory planning and local trajectory planning. The blue area and line refer a behavioral trajectory and a local trajectory, respectively.

the proposed algorithm needs only a single local trajectory planner regardless of various behavioral selections. For the performance evaluation, we tested the algorithm in lane change scenarios because a behavior decision of a lane change was a challenging problem in an urban environment.

This paper is organized as follows. Section II explains the overall architecture of the proposed trajectory planning. Then, Section III describes how to generate a behavioral trajectory. Based on the trajectory, Section IV makes a motion-level trajectory, called local trajectory. This algorithm is evaluated in Section V. In Section VI, we provide the conclusion.

II. Overview

For safe driving in urban environments, an autonomous car must determine their future trajectory considering vehicle dynamics, various road types, traffic rules, and traffic participants (vehicles, bikes, bicycles, and pedestrians). Combinations of these elements make trajectory planning problem more complicated. Moreover, multiple lanes increase maneuver variants. In these situations, finding a global optimum trajectory requires high computation because there are a lot of local minimums. To overcome the problem, this paper proposed the *behavioral trajectory planning*. The overall process for the total motion planning is described in Fig. 1.

The *behavioral trajectory* is a macroscopic maneuver to represent the outline of an ego's future trajectory. To determine the trajectory, this algorithm focuses on maneuver-related elements such as a structure of multiple lanes, motions of other vehicles, and traffic rules among all of the considerations for urban driving. The other elements (vehicle dynamics and driving boundary) are simplified to understand the macroscopic driving environments. Since these elements can be more efficiently modeled as discrete models rather than continuous mathematical models, the behavior algorithm uses a sampling-based trajectory generation method. The result of this algorithm provides a spatiotemporal boundary to generate the motion-level trajectory.

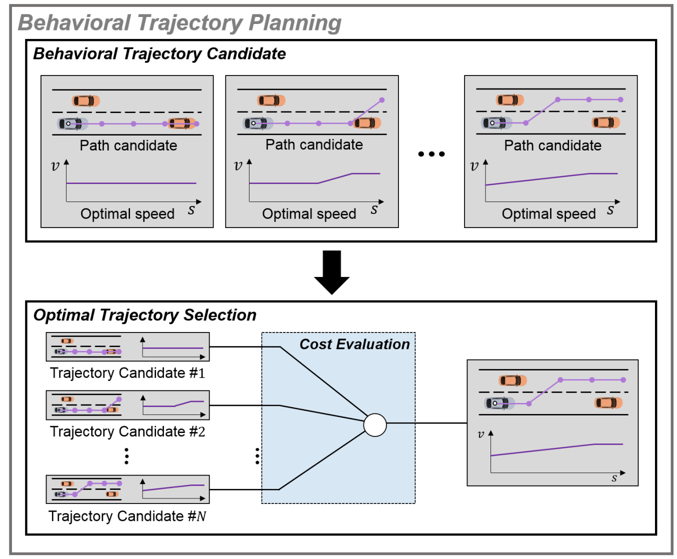


Fig. 2. Behavioral trajectory planning has three sequential steps: path candidate generation, optimal speed profile generation, and trajectory selection.

Due to the reduced searching area, *a local trajectory planner* focuses on the specific region to find the final trajectory. The trajectory can be generated only considering vehicle dynamics, boundaries, and surrounding objects with the exception of lane and traffic rule information. Since the conditions for the trajectory generation are same as trajectory generation in a single lane, various local trajectory methods can be applied in this system. For efficient calculation of vehicle dynamics, this paper uses the optimization-based trajectory generation method [12].

III. Behavioral Trajectory Planning

Behavioral trajectory planning determines a specific maneuver for a safe and comfort driving based on the macroscopic information such as road network, multiple-lane structure, traffic rule, and distribution of surrounding vehicles. To manage these elements, sampling-based planning method [10], [11] is used. The trajectory samples have own path and speed profile. The path is generated following the road shape and lane disposition. The speed profile is optimized by considering movements of surrounding vehicles and traffic rules.

The *behavioral trajectory* provides a guidance region by using sequential sets of a spatiotemporal node [13] and a boundary. In other words, each set means that an ego's position constraint at a certain time. These sets are constructed by *Path-Velocity Decomposition (PVD)* method [14], [15]. Based on this, the behavioral trajectory is generated by following three steps: path candidate generation, speed profile generation of each path candidate, and the final trajectory selection. Fig. 2 describes the process of the *behavioral trajectory planning*.

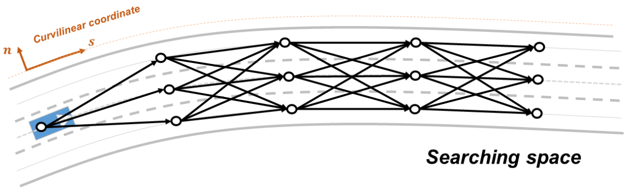


Fig. 3. Searching space is a base grid to generate path candidates. This space is configured in the curvilinear coordinate. Each node is placed on all of the lanes and at the interval of ΔS_{int}.

A. Path Candidate Generation

1) Searching Space: Path candidates refer to various spatial maneuvers where an ego vehicle can drive on structured roads. To generate various path candidates, searching space is defined as shown in Fig. 3. This space is a lane-based grid to abstract a structure of multiple lanes. Each grid is designed by a directed graph in the curvilinear coordinate [16]. In the graph, nodes are generated at each lane and intervals of ΔS_{int} in s-direction of the curvilinear coordinate. The interval ΔS_{int} is determined by a constant time gap. In other words, ΔS_{int} is proportional to ego's speed. Edges of the graph are connected between neighborhood nodes except for nodes at the same position s.

2) Feasible Path Candidates: A feasible path means that an ego vehicle can pass through the path without a collision. To generate feasible paths from the searching space graph, it requires two steps: a collision check of each edge and path candidate extraction. A collision check is to remove the edges overlapping static obstacles. Since a static obstacles occupy a specific region permanently, the edge is infeasible path segment. However, edges on dynamic obstacles are feasible segments because they do not keep staying on the areas. Thus, dynamic obstacles are not considered in this step. The final path candidates are extracted from the feasible edges. Since these candidates do not contain temporal information, speed information will be added to them in the next step.

B. Speed Profile Generation

Each path candidate means a safe guidance line if there is no moving object. However, there are many moving objects in urban environments. These environments also have various speed-related elements such as speed limits, speed bump, and stop line. To manage these elements, the optimal speed profile is added into each path candidate. This profile makes each maneuver avoid a collision with dynamic obstacles within traffic rules and physical limits of the ego vehicle. The optimal profile is generated by two steps: speed limit profile generation and optimal speed search. The speed limit profile V_{limit} is a table containing the speed limit along the s axis of a road geometry. This limit is determined by the minimum value between regulation speed V_{reg} and the maximum speed V_{acc} by the lateral acceleration. The speed V_{acc} can be calculated by

$$v_{acc} = \sqrt{\frac{a_{lat}}{k}} \quad (1)$$

where k and a_{lat} refer to road curvature and the allowable lateral acceleration which is selected by a physical limit and driving comfort.

Based on this profile, the optimal speed search determines the best speed profile considering a physical limit (longitudinal acceleration limit) and dynamic objects. For an efficient search, this algorithm utilizes the Space-Time domain (S-T domain) [5], [17]. In this domain, there are two axes: time and space axis. These axes mean a predictive time and a position along a specific path. The origin point (0,0) refers to the current position of an ego vehicle. Moreover, dynamic obstacles can be represented by the gray region as shown in Fig. 4. Thus, the problem is defined as finding the optimal speed profile from the origin to the prediction time $T_{predict}$ within the collision-free area (white region) in Fig. 4.

To solve this problem, *Hybrid A** [18] is applied. This algorithm is a sampling-based search algorithm to expand a graph using a vehicle model. This has three sequential steps. In the first step, speed samples are generated between the minimum speed v_{min} and maximum speed v_{max}. These speeds are determined by the maximum longitudinal acceleration a_{max} and the speed limit of the speed limit profile described in the previous step. In addition, the number of samples n_{vsam} is a configuration parameter to adjust the speed resolution. The second step is to check if each sample collides with a dynamic obstacle (gray region). If there is a collision in a speed sample, this sample is rejected. The final step is cost evaluation. In this step, the performance index f_{speed} of each sample is calculated by weighted sum of reference cost f_r, acceleration cost f_a, and jerk cost f_j as following

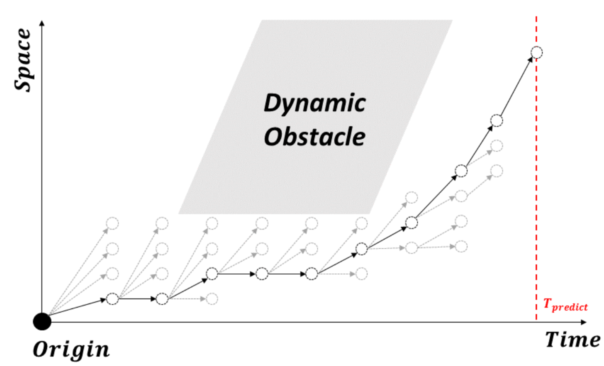


Fig. 4. Hybrid A* searches the optimal speed profile in Space-Time domain (ST domain).

$$f_{speed} = w_r f_r + w_a f_a + w_j f_j \tag{2}$$

where w_r, w_a, and w_j are weights of a reference, acceleration, and jerk cost respectively. The reference cost f_r is the difference between the speed limit in the speed limit profile (the reference) and the sample speed. Based on this cost, this search algorithm has a characteristics to follow the maximum speed of a road. The acceleration cost f_a and jerk cost f_j are inserted for driving comfort as following

$$f_a = |v_{limit} - v_{sample}| \tag{3}$$

$$f_a = \frac{|v_{sample} - v_{t-1}|}{\Delta t} \tag{4}$$

$$f_j = \frac{|v_{sample} - 2v_{t-1} + v_{t-2}|}{\Delta t} \tag{5}$$

where v_{sample}, v_{limit}, and v_t refer to a sampling speed for graph expansion of Hybrid A*, the limit speed, and the speed at time t, respectively.

C. Trajectory Selection

Trajectory candidates are feasible maneuvers considering a driving situation. However, each candidate has different characteristics of a driving lane, a travel distance, and consistency of the previous trajectory. To select the best one in these candidates, the algorithm uses the performance index of each behavioral candidate which is calculated by

$$f_{sel} = w_{LC} C_{LC} + w_{dis} C_{dis} + w_{cons} C_{cons} \tag{6}$$

where C_{LC}, C_{dis}, and C_{cons} are cost values for lane change, travel distance, and consistency. The lane change cost C_{LC} is the number of lane changes in one behavioral trajectory candidate. It is calculated by

$$C_{LC} = \sum_{k=1}^{N_{path}-1} |Lane(n_k) - Lane(n_{k-1})| \tag{7}$$

where N_{path} is the number of behavioral path's nodes and $Lane(n_k)$ is the lane index of the node n_k. This cost can adjust tendency to do a lane change because the cost increases when the trajectory tries to do a lane change.

The travel distance cost C_{dis} measures how far the trajectory can drive in a prediction time $T_{predict}$. The cost is modeled by

$$C_{dis} = V_{limit} T_{predict} - \int_{t=0}^{T_{predict}} v(t) dt \tag{8}$$

where $v(t)$ is the speed at the time t.

Since drivers tend to prefer the trajectory with the longest travel distance, this cost decreases when the travel distance increases.

The consistency cost C_{cons} is an index to prevent an unnecessary change of a behavioral trajectory. To measure the difference between the previous and current behavioral trajectories, the cost is defined as

$$C_{cons} = \sum_{j=0}^{N_b-k} |X_t^j - X_{t-1}^{j+k}| \tag{9}$$

where X_t^j is the re-sampled node from a trajectory candidate at the same time interval. t and j refer to the index of time and node. The index k is the index shifter to synchronize two trajectories generated at different time frames.

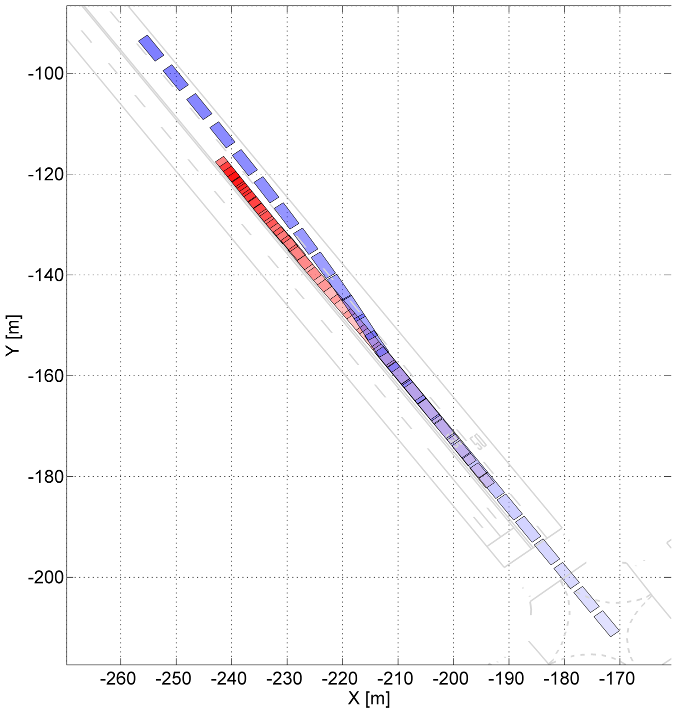
IV. Local Trajectory Planning

A *local trajectory* is a micro-scale trajectory to determine a concrete path and speed for autonomous driving. The optimal trajectory must be generated within the behavioral boundary. In other words, the behavioral trajectory reduces the searching boundary and provides the reference trajectory considering road structure, traffic rules, and a movement pattern of dynamic objects. For these reason, local trajectory planner only focuses on dynamic objects within the region, vehicle dynamics, and the behavioral boundary. Due to the behavioral trajectory, a local trajectory problem in multiple lanes can be simplified as a single lane trajectory problem. To solve this problem, there are various previous methods [19]–[22]. Since the environments for local trajectory generation can be modeled as a continuous mathematical model by vehicle dynamics, the *numerical optimization based trajectory generation* [20] is used to find the optimal trajectory with constraint and boundary conditions in this paper. This method has a strong advantage of fast convergence if optimization-related functions are guaranteed to be convex functions.

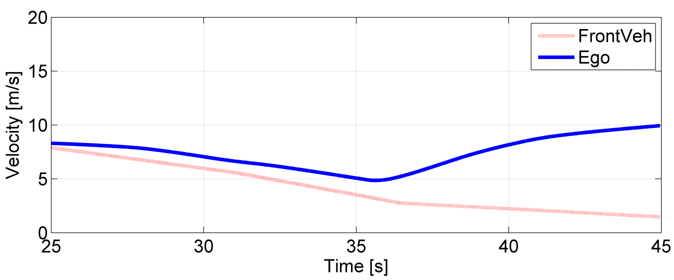
V. Results

The behavioral trajectory determines a macroscopic trajectory and a boundary to represent a driving maneuver. Based on the information, local trajectory planning generates a motion-level trajectory to guide ego's motion. To evaluate the algorithm, we implemented this algorithm into our simulation system based on C++. This system had a vehicle dynamic model to emulate vehicle's motion and longitudinal and lateral motion control modules to make this model follow the input trajectory. In addition, positions and speed profiles of other vehicles can be configured in this simulator. Each planner was executed at the period of 50 ms on Intel Core i5. The behavioral trajectory planning was evaluated by following lane change scenarios.

The first one was the situation where the front vehicle (red) slowed down in multiple lanes as shown in Fig. 5. Although a



(a) Vehicle trajectory



(b) Vehicle speed

Fig. 5. This is the result of the scenario 1. Although the front vehicle (red) slowed down, the ego vehicle (blue) followed the front one by 35 s. However, The ego one started to overtakes it because its speed became too slow. The degree of a lane change cab be adjusted by w_{LC}, w_{dis}, and w_{cons}.

road's speed limit was 40 km/h, the ego vehicle was following the front one by 35 s. It was because the lane change cost C_{LC} and the consistency cost C_{cons} prevented a lane change. The tendency of a lane change was able to be tuned by the distance weigh w_{dis}. At $t = 35$ s, the behavioral trajectory planning selected a trajectory for a lane change and accelerated the speed as described in Fig. 5 (b). The whole trajectory of the ego (blue) and front one (red) was plotted in Fig. 5 (a). Each box meant its position at each time frame.

The other case was more complicated situation. There were two surrounding vehicles (green and red) which enclosed the ego vehicle (blue) as shown in Fig. 6. Moreover, since their speeds were same, the ego vehicle had no choice but to keep an own lane with maintaining a constant time gap to the front

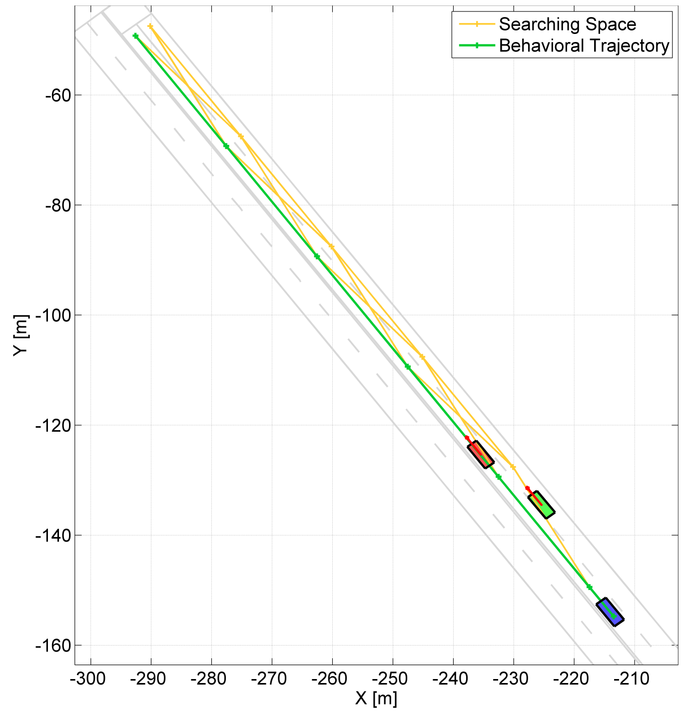


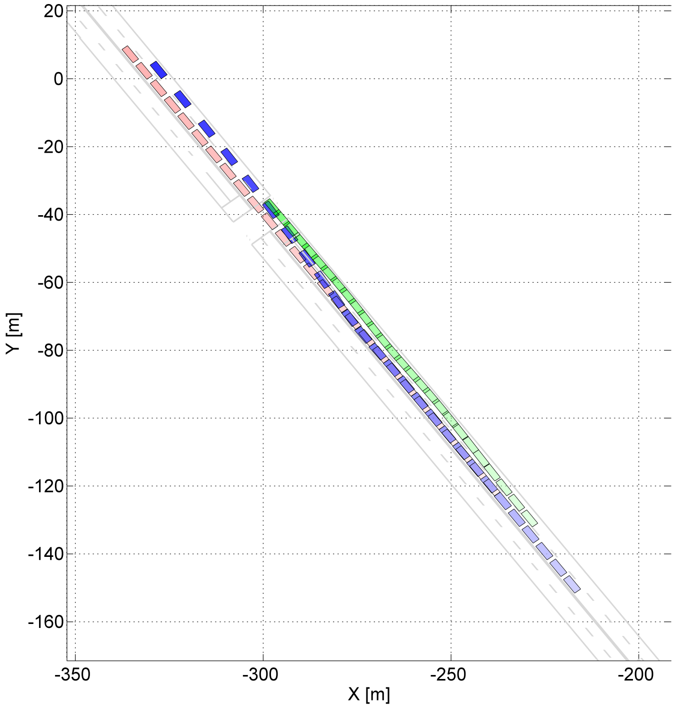
Fig. 6. This was a snap shot of the scenario 2 at $t = 10$ s. There were two surrounding vehicles (red and green) and one ego (blue). In this situation, the searching space and behavioral trajectory were described by yellow and green lines.

vehicle. The side vehicle (green) started to slow down, and the spacing between the front vehicle (red) and the side one (green) increased. When the behavioral planning estimated that the spacing was sufficient to do a lane change, the ego (blue) increased its speed and moved to the right lane as shown in Fig. 7.

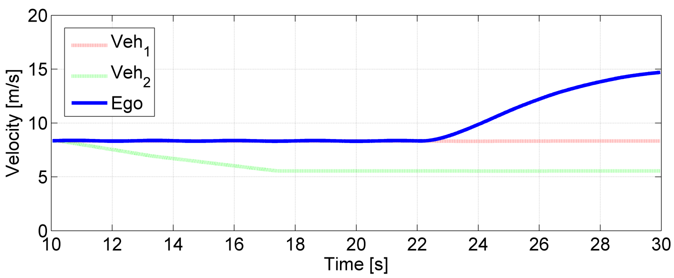
VI. Conclusion

This paper proposes a *behavioral trajectory planning* algorithm for urban autonomous cars. The algorithm determines a macroscopic trajectory to guide an autonomous car. Due to the reduced spatiotemporal region of the trajectory, a local trajectory planner can find the optimal trajectory, more efficiently. In addition, since the algorithm uses a single of a local trajectory planner, we can keep the simple architecture regardless of various driving maneuvers. We evaluated the performance of the algorithm based on the lane change scenarios. In these situations, the behavioral trajectory planning generated appropriate and acceptable maneuvers.

In the future work, we have a plan to evaluate the performance in more complex scenarios and verified by real driving tests. In addition, it is an important to determine the adequate size of the searching space considering computation and a prediction time.



(a) Vehicle trajectory



(b) Vehicle speed

Fig. 7. This is the result of the scenario 2. At $t = 10$ s, the ego vehicle kept own lane because surrounding vehicle enclosed it. When the speed of the green one decreased and the spacing between two surrounding vehicles became sufficient, the ego started to do a lane change with acceleration. In this case, a speed limit was 60 km/h (16.7 m/s).

Acknowledgment

This work was supported by the following: National Research Foundation through the Ministry of Education and Science Technology, Republic of Korea (grant number 2011-0017495); BK21 Plus Program of the Ministry of Education, Republic of Korea (grant number 22A20130000045); Industrial Strategy Technology Development Program of the Ministry of Knowledge Economy, Republic of Korea (grant number 10039673); Energy Resource RD Program of the Ministry of Knowledge Economy, Republic of Korea (grant number 2006ETR11P091C); Korean Ministry of Land, Infrastructure and Transport, (grant number15RTRPB067379-03); Industrial Strategic Technology Development Program of the Ministry of Trade, Industry and Energy, Republic of Korea (grant numbers 10042633 and 10060068); International Collaborative Research and Development Program through the Ministry of Trade, Industry and Energy, Republic of Korea (grant number N0001992).

References

[1] D. González, J. Perez, V. Milanés, and F. Nashashibi, “A review of motion planning techniques for automated vehicles”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016.

[2] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, “A survey of motion planning and control techniques for self-driving urban vehicles”, *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[3] J. Wei, J. M. Snider, T. Gu, J. M. Dolan, and B. Litkouhi, “A behavioral planning framework for autonomous driving”, in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, IEEE, 2014, pp. 458–464.

[4] M. Brannstrom, F. Sandblom, and L. Hammarstrand, “A probabilistic framework for decision-making in collision avoidance systems”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 637–648, 2013.

[5] J. Johnson and K. Hauser, “Optimal longitudinal control planning with moving obstacles”, in *Intelligent Vehicles Symposium (IV)*, 2013, pp. 605–611.

[6] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, and G. Hoffmann, “Stanley: The robot that won the darpa grand challenge”, *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[7] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, *et al.*, “Autonomous driving in urban environments: Boss and the urban challenge”, *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[8] J. Kim, K. Jo, K. Chu, and M. Sunwoo, “Road-model-based and graph-structure-based hierarchical path-planning approach for autonomous vehicles”, *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 228, no. 8, pp. 909–928, 2014.

[9] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, *et al.*, “Junior: The stanford entry in the urban challenge”, *Journal of Field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.

[10] T. Gu, J. M. Dolan, and J.-W. Lee, “On-road trajectory planning for general autonomous driving with enhanced tunability”, in *Intelligent Autonomous Systems 13*. Springer, 2016, pp. 247–261.

[11] M. McNaughton, C. Urmson, J. M. Dolan, and L. Jin-Woo, “Motion planning for autonomous driving with a conformal spatiotemporal lattice”, in *Robotics and Au-*

tomation (ICRA), 2011 IEEE International Conference on, 2011, pp. 4889–4895.

[12] J. Ziegler, P. Bender, T. Dang, and C. Stiller, “Trajectory planning for bertha; a local, continuous method”, in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, 2014, pp. 450–457.

[13] J. Ziegler and C. Stiller, “Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios”, *Intelligent Robots and Systems*, pp. 1879–1884, 2009.

[14] K. Kant and S. W. Zucker, “Toward efficient trajectory planning: The path-velocity decomposition”, *The International Journal of Robotics …*, vol. 5, no. 3, pp. 72–89, 1986.

[15] T. Fraichard, “Trajectory planning amidst moving obstacles: Path-velocity decomposition revisited”, *Journal of the Brazilian Computer Society*, vol. 4, no. 3, 1998.

[16] J. Kim, K. Jo, W. Lim, M. Lee, and M. Sunwoo, “Curvilinear-coordinate-based object and situation assessment for highly automated vehicles”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1559–1575, 2015.

[17] J. Johnson and K. Hauser, “Optimal acceleration-bounded trajectory planning in dynamic environments along a specified path”, in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 2035–2041.

[18] Dolgov, Dmitri, *et al.*, “Path planning for autonomous vehicles in unknown semi-structured environments”, *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501, 2010.

[19] J. Ziegler and C. Stiller, “Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios”, in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, IEEE, 2009, pp. 1879–1884.

[20] J. Ziegler, P. Bender, T. Dang, and C. Stiller, “Trajectory planning for bertha—a local, continuous method”, in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, IEEE, 2014, pp. 450–457.

[21] T. Gu, J. Snider, J. M. Dolan, and J.-w. Lee, “Focused trajectory planning for autonomous on-road driving”, in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, IEEE, 2013, pp. 547–552.

[22] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, “Optimal trajectory generation for dynamic street scenarios in a frenet frame”, in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, IEEE, 2010, pp. 987–993.