32nd Electric Vehicle Symposium (EVS32)
Lyon, France, May 19 - 22, 2019

Re-Plannable Parking Path Planning for Real Driving Based on Utility-Guided Sampling Method

Seongjin Lee[1], Wonteak Lim, Myoungho Sunwoo
[1]*Hanyang Unversity, Wangsimniro 222, Seongdonggu, Seoul, Korea*
seongjin90@gmail.com

Executive Summary

In commercialized automated parking systems, a parking path is generated from the vacant parking space detected by ultrasonic or camera sensors which produce sensing noise. Since this sensing noise affects the detection of parking space adversely, parking path planning under perceptional uncertainty is a challenging problem. To solve the conundrum, this paper proposes a re-plannable parking path planning algorithm by combining a sampling method with utility theory. Uncertain parking space can be formulated deterministically through the sampling method, and one of the sampled paths is selected according to utility theory. Furthermore, the proposed method can provide an adaptable path in an uncertain parking space.
Keywords: autonomous vehicle, safety, smart

1 Introduction

Automotive industries have increasingly provided drivers with a variety of functions for safety and convenience. An automated parking system that parks autonomously without driver intervention is able to save time and reduce the risk of accidents. The process of such a parking system is divided into three steps: recognizing vacant parking space with ultrasonic or camera sensors, creating a safe path to reach the detected parking space, and controlling the vehicle to follow the generated path. Among them, parking path generation is one of the core technologies which is responsible for generating a path that can securely and accurately locates at the detected parking space.
Since the relative position of the parking space is utilized to generate a parking path, the parking space position must be accurate and consistent to precisely locate the vehicle at the parking space. In commercial systems, Around View Monitoring systems(AVM) or ultrasonic sensors, which are limited in reducing noise, are used to detect parking spaces. When using these sensors, detection error arises as a result of propagation from the sensing noise. Eventually, the sensing noise adversely affects the recognition of the parking space in question and renders the parking space uncertain. If a parking path is generated from the inconsistent parking space, the vehicle is prevented from reaching the actual position. In addition, this path can lead to a system failure that may cause the vehicle to stop controlling itself or even cause an accident. Since inconsistent errors cannot be completely eliminated, this uncertainty should be considered in the planning process. Furthermore, the parking path planner should be able to find another path when the vehicle does not reach the actual position.
Over the last few years, some researchers have developed advanced planning algorithms that take uncertainty into account. One of the solutions, Sampling-Based Planning (SBP) [1–6], has been greatly successful in solving motion planning problems, since SBPs provide fast solutions for difficult problems by taking a randomized sampling approach. In other words, SBPs are useful in discarding the current path and re-planning the next in uncertain environment.
Bry et al. [1] represent external uncertainty as the probability of the ego position, and then samples nodes in the probability boundaries. On the other hand, Vitus et al. [2] and Missiuro et al. [3] propose motion

planning algorithms that consider environmental uncertainty as a probability of obstacle boundaries, then sample nodes by avoiding the obstacle boundaries. [4,5] suggest a partially observable Markov decision process (POMDP) framework to sample nodes for SBP, whereas Burns, et al. [6] propose a sampling method to reflect environmental uncertainty directly into the planning process by applying utility theory. These algorithms consider uncertainty during the sampling process when constructing the searching-tree of SBPs. However, SBPs require an enormous amount of sample nodes to operate well in narrow environments, which are often the case for parking. The searching-tree, which consists of these many samples, has limitations in planning and re-planning the path in real-time.

In the case of parking, there is a certain pattern of paths based on the relative position to the parking space. For example, when we do left-sided perpendicular parking, we drive first to the upper left side of the parking space and then move back towards the parking space. In this context, it is more useful to sample pattern paths rather than an enormous number of nodes. When finding a parking pattern, there are three classes: Geometric methods, Tree-searching methods, and Optimization methods. The first class [7–9] finds a solution by combining geometric primitives such as straight lines, circles, and spirals, etc. These methods are simple but only operate well in a predefined situation. The second class [10–12] uses tree-searching methods such as Hybrid-A* or RRT. However, since these methods require tremendous number of nodes constructing a tree, these take a long time to execute. The third class is for optimization-based methods. These methods find a parking path by solving Model Predictive Control (MPC) [13–15] or Optimal Control Problem (OCP) [16, 17]. MPC-based methods make the parking planning problem general, although they cannot be executed in real-time when the problem is difficult. On the other hands, OCP-based methods have a shorter execution time compared to MPC-based methods, and these are able to find a solution regardless of the location of the vehicle.

The sampling-based method is necessary from the perspective of re-planning coming from the rapid production of new solutions. However, the sampling-based method is performed within a tremendous amount of time when the size of the searching tree becomes enormous in the narrow environment. On the other hand, optimization-based methods hold the advantage of finding a parking path in well-known environments. By combining these two kinds of methods, the new searching tree can be constructed. Through utility theory, the optimal path can be selected in the new searching tree. Therefore, taking advantages of these three kinds of methods is necessary to park in the desired parking space in a precise manner in real-driving. In this paper, we propose a parking planning algorithm that can plan and re-plan from uncertain perceptional information by combining a sampling method, an optimization method and utility theory. The proposed algorithm has the advantage of significantly reducing the failure rate of autonomous parking, and re-planning the parking path when parking failed.

We have introduced the limitations of the current parking system as well as the contribution of the proposed method in Section 1. The remainder of this paper is arranged as follows. The overview and the details of the proposed method are provided in Section 2. The proposed method is then evaluated with experiments in Section 3. This paper is then concluded in Section 4.

2 Approach

The proposed algorithm is divided into three parts: Parking Space Sampling, Path Candidate Generation, and Optimal Path Selection. At the first step, *Parking Space Sampling*, we propose the perceptional error model to represent the probability of the parking space position that comes from the perception system by utilizing Gauss error model. From the probability of the parking space position, new parking spaces, which are likely true positions, are sampled. In the *Path Candidate Generation*, parking paths are generated to reach each sampled parking spaces. An OCP-based method can find the parking path regardless of the vehicle location within few milliseconds. In the final step, *Optimal Path Selection*, the optimal path most likely to reach the true parking space position is determined by applying utility theory. A utility function is proposed which considers the probability of each path candidate, the consistency with the previous path, and the current position of the vehicle in question.

2.1 Parking Space Sampling

Since position uncertainty is propagated from commercial sensors that generate noisy measurements, a parking space is always uncertain in its position even though the detection algorithm may be accurate. We assume that the planner uses the deterministic perception model that does not consider uncertainty. Instead, the planner uses deterministically sampled parking space information from a perception probability model to interpret an uncertainty-free world model. We now present the perception probability model which is formulated with Gauss error function.

2.1.1 Perception Probability Model

We assume that the noise level gets higher when the measured distance is longer. For example, a parking space which is detected near the ego vehicle has a higher true probability than a farther one. We restrict

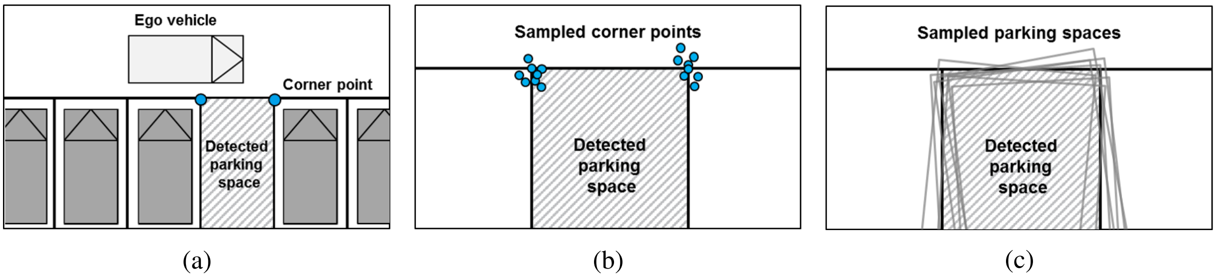

Figure 1: A process of parking space sampling

our model to translational axis, but the same approach could be extended to rotational axis. This is done using the cumulative distribution function (CDF) of the Gaussian:

$$P(measurement) = 1 - \frac{1}{2}(1 + erf(\frac{d}{\sigma\sqrt{2}})) \tag{1}$$

where erf is the Gauss error function and d is the distance to the measured position. This represents an approximation of the true probability of the measurements.

2.1.2 Sampling Process

When the parking space is detected from the perception system, the system first looks for corner points of the parking space (see Fig.1-(a)). The corner points manifest the true probability through the perception probability model. By using a complementary true probability of the corner points, the positions of the corner points are sampled on the x and y axes as shown in Fig.1-(b). The number of the sampled corner points increases according to its probability. In Fig.1-(c), it is shown that the sampled parking spaces are the connection of two corner points.

2.2 Path Candidate Generation

To find a path candidate from the ego position to each sampled destination, optimization-based methods, which are called Optimal Control Problems (OCP), are used in the path candidate generation. One of the OCP-based methods [16] can solve the problem within a few milliseconds and allows the path to be found regardless of the ego position. Furthermore, unlike other methods, this method automatically calculates a direction switching point. For these reasons, the OCP-based method is used to generate path candidates for each sampled destination.

2.2.1 Vehicle Kinematic Model

Since a vehicle is in low speed during parking, the tire slips of a given vehicle can be ignored. The vehicle kinematic model is used to describe vehicle motion. The vehicle model is simplified to have one front wheel and one rear wheel. This model has a state composed of $[x, y, \theta]^T$, where $[x, y]$ is the center position of the rear axle, and θ denotes the direction of the vehicle. The mathematical equation of the kinematic model is described by

$$\dot{\mathbf{q}} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} v\cos\theta \\ v\sin\theta \\ v\frac{\tan\delta}{L} \end{pmatrix} \tag{2}$$

where v and δ are control inputs that relate to the velocity and steering angle, and L indicates the wheelbase of the vehicle. To reduce computation time, this kinematic model can be simplified by using Path-Velocity-Decomposition [18]. With this decomposition, time can be ignored. Velocity can be written as $v = D\frac{ds}{dt}$, where s is the path length and $D \in \{-1, 1\}$ indicates the vehicle direction (i.e. forward or backward). The simplified equation can be written as,

$$\mathbf{q}^{'} = \begin{pmatrix} x^{'} \\ y^{'} \\ \theta^{'} \end{pmatrix} = \begin{pmatrix} D\cos\theta \\ D\sin\theta \\ Du_l \end{pmatrix} \tag{3}$$

where $(\cdot)'$ denotes the derivative with respect to the path length s, and the new control input u_l is equal to $\tan\delta/L$. To apply discrete optimization, equation (3) can be discretized with respect to the path length s by using second order Runge-Kutta discretization. The discretized equation can be written as,

$$\mathbf{q}_{i+1} = \begin{pmatrix} x_{i+1} \\ y_{i+1} \\ \theta_{i+1} \end{pmatrix} = \begin{pmatrix} x_i + D\eta_i \cos(\theta_i + D\frac{\eta_i u_{l_i}}{2}) \\ y_i + D\eta_i \sin(\theta_i + D\frac{\eta_i u_{l_i}}{2}) \\ \theta_i + D\eta_i u_{l_i} \end{pmatrix} \tag{4}$$

where the control input $\mathbf{u}$ of (4) is comprised of u_l and step length η.

2.2.2 Obstacles

To generate a collision-free path, obstacles need to be expressed as constraints. If obstacles can be described to convex polygons, the computation time of optimization will be reduced. Therefore, obstacles are regarded as convex polygons. If the obstacle is non-convex, the obstacle would be divided into a set of convex polygons through convex decomposition [19]. All convex polygonal obstacles are described being larger than their original size. During optimization, obstacles are transformed into inequality constraints in the cost function. The inequality constraints check whether the vehicle is within the obstacle boundary or not.

2.2.3 Optimal Control Problem

To find a set of optimal control inputs, the optimal control problem should be formulated. This can be done by the constrained static optimization problem.

$$\underset{u_i}{\text{argmin}} \quad l_{O_i}(\mathbf{q}_{i+1}) \tag{5}$$

$$\text{s.t.} \quad \mathbf{q}_{i+1} = \mathbf{f}(\mathbf{q}_i, \mathbf{u}_i, D) \tag{6}$$

$$\mathbf{h}_P(\mathbf{q}_{i+1}) \leq 0 \tag{7}$$

$$\mathbf{u}_{min} \leq \mathbf{u}_i \leq \mathbf{u}_{max} \tag{8}$$

where the objective function (5) is expressed as,

$$l_{O_i}(\mathbf{q}_i) = r_\theta e_{\theta_i}^2 + \mathbf{e}_{P_i}^T \mathbf{R} \mathbf{e}_{P_i}. \tag{9}$$

The objective function is comprised of error terms with weighting terms. Error terms between the current state to the final state include the distance error in direction x, y and the heading angle error with the starting position $\mathbf{q}_S$. The weighting term $\mathbf{R}$ and r_θ with respect to each error should be positive definite and positive value, respectively. There is one equality constraint and two inequality constraints. The first constraint in (6) is equal to (4) for the vehicle motion constraint. The two inequality constraints in (7), (8) are for collision and control inputs, respectively. The collision constraint can be formulated with the Minkowski sum [19], and input constraint should be within the physical limits of the vehicle, which are the maximum steering angle and the maximum step length at one optimization iteration.

2.2.4 Direction Switching Point

To find a path before arriving at the starting position, the vehicle may exhibit changes in the direction. There are two heuristic rules for directional change. The first rule is to switch the direction when the vehicle cannot move in that direction. This means that the vehicle will collide with obstacles. The second rule is to change the direction when the current result of optimization $l_{O_i^*}$ is larger than the previous result $l_{O_{i-1}^*}$. The larger value of the cost function refers to a more distant position from the start position. Therefore, the vehicle changes direction in order to reduce the distance error with the start position.

2.3 Optimal Path Selection

To select the optimal path candidate, the planner must use limited information to identify the best possible path. To achieve this, we use the formalization of Bernoullian utility [20]. Based on utility theory, every beneficial path has positive utility, which exhibits a high probability of reaching the parking space. Conversely, every useless path has negative cost. Negative cost is computed from the consequences of a path failure such as being too far from the ego vehicle, or not reaching the designated parking space.

Expected utility combines the notions of utility and cost, weighted by the probability that the path is beneficial. The expected utility of each path candidate can be formulated as

$$P(p = true) \cdot U(p = true) + (1 - P(p = true)) \cdot C(p = far) \tag{10}$$

where, p indicates the path candidate, the function U measures utility and the function C measures cost. $P(\cdot)$ measures the true probability of the path candidate. This can be calculated in the same way as the parking space, since the path candidate is generated from each sampled parking space. The path with the greatest expected utility maximizes rewards and minimizes risks of misplacement at a given parking space.

2.3.1 Utility

The function $U(\cdot)$ measures the preference of the path candidate. To calculate utility, we consider whether the vehicle can reach to the actual parking space through the path candidate and how close the path candidate is to the ego vehicle. The first term (U_{dst}) can be expressed by the distance between the currently detected parking space and the end point of the path candidate. The second term (U_{ego}) can be formulated as the distance between the position of the ego vehicle and the first point of the path candidate. Total utility can be calculated through the weighted sum of each utility term.

$$U(p) = w_{dst} \cdot U_{dst} + w_{ego} \cdot U_{ego} \tag{11}$$

where U_{dst} refers the utility of the distance between the currently detected parking space and the end point of the path candidate, and U_{ego} refers to the utility of the distance between the position of the ego vehicle and the first point of the path candidate. w_{dst} and w_{ego} are weighting term for each utility term. U_{dst} and U_{ego} can be normalized through Gauss error function.

$$U(p) = \frac{1}{2}(1 - erf(\frac{d}{\sigma\sqrt{2}})) \tag{12}$$

where, d refers the distance of each utility term, and σ denotes the covariance of each term. Here, σ can be a tuning factor which determine the accuracy of the equipped perception system.

2.3.2 Cost

The cost function $C(\cdot)$ determine the non-preference of each path candidate. There are three types of cost: Destination cost (C_{dst}), Ego cost (C_{ego}), and Consistent cost (C_{const}). The first two terms are measured in the same way as the distance in each utility term. Consistent cost refers to the consistency between the previous and current optimal path candidate. Consistent cost can be formulated the distance of each path point between the previous path and the current path. To make sure that the cost has negative value, we use a modified Gauss error function as,

$$C(p) = -\frac{1}{2}(1 + erf(\frac{d}{\sigma\sqrt{2}})) \tag{13}$$

Similar to utility function, d denotes the distance of each cost term, and σ is able to tune the cost term. Total cost can be measured through the weighted sum of each cost term.

2.3.3 Re-planning Strategy

When the expected utility is negative, or when the currently detected parking space is far from the previous one with differences in x-direction ϵ_x, y-direction ϵ_y, and heading angle ϵ_θ, all path candidates are useless. This means that all the sampled parking spaces are significantly different from the actual parking space. At that time, the vehicle must be stopped to resample parking spaces from the currently detected parking space.

3 Simulation Results

To verify the re-planning capability of the proposed algorithm and the feasibility of the re-planning strategy, simulation scenarios are investigated, which imitate the real-driving situation using our system that are equipped with AVM sensors. The dimensions of the vehicle are set identically to the electric vehicle ZOE. The parameters are given in Table 1, where l represents the length and w the width of the vehicle. The simulation vehicle parks at the parking space with length l_p and width w_p. The maximum

Table 1: Parameters for the vehicle and the parking space

$l(m)$	$w(m)$	$L(m)$	$l_p(m)$	$w_p(m)$
4.084	1.730	2.845	5.0	2.0

Table 2: Parameters for the proposed algorithm

N	$\sigma_{U_{dst}}(m)$	$\sigma_{U_{ego}}(m)$	$\sigma_{C_{dst}}(m)$	$\sigma_{C_{ego}}(m)$	$\sigma_{C_{const}}(m)$	$\epsilon_x(m)$	$\epsilon_y(m)$	$\epsilon_\theta(\text{deg})$
30	0.2	0.05	0.2	0.05	1	0.2	0.2	2.0

control input for path candidate generation u_l is given by $u_{l_{max}} = \frac{\tan(\delta_{max})}{L} = 0.2rad$. The step length is limited to $\eta \in [0.1m, 0.3m]$. The parameters for parking space sampling and optimal path selection are also shown in Table 2. The parameters can be used to adjust the quality of re-planning. A higher value of N leads to a robust planner from a large number of sampled parking spaces. Furthermore, smaller values of ϵ_x, ϵ_y, and ϵ_θ mean that the algorithm terminates in a more precise manner.
As shown in Fig.2, the parking space is modelled in the shape of polygons where each point has a noise. When parking, the detected parking space (gray rectangle) moves with this noise, and the noise level varies with its speed. The final parking space (black rectangle) eventually differs from the initially-detected parking space. The total error between the initial parking space and the actual parking space is $(0.37m, 0.06m, 2.5\,\text{deg})$ in x, y-direction and heading angle, respectively.

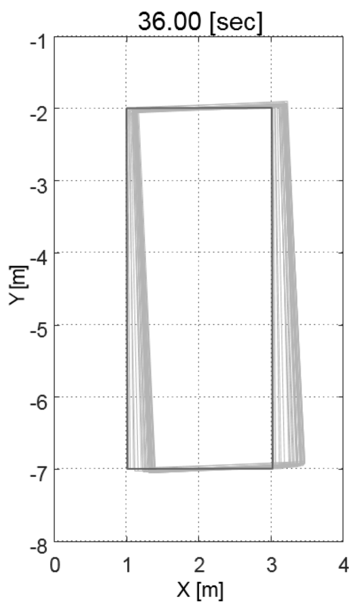

Figure 2: A simulation scenario that imitates the real-driving car equipped with AVM sensors

Fig.3 shows the sequence of automated parking without re-planning. The ego vehicle (green rectangle) starts at $(0m, 0m)$ and the parking space is located at $(2m, -6m)$. Due to sensor noise, the parking space is detected at $(2.34m, -5.89m)$ and the parking path (blue line) has been planned to reach this imprecise parking space. Since the ego vehicle tracks this inaccurate path, it is eventually mislocated at the initially-detected parking space with an error $(0.34m, 0.11m, 2.40\,\text{deg})$ in x, y-direction and heading angle, respectively.
On the other hand, the sequence of the proposed algorithm is shown in Fig.4 to 6. The overall process from parking space sampling to optimal path selection is shown in Fig.4. The gray rectangles are the sampled parking spaces and the gray paths are the path candidates. Each path candidate has an expected utility calculated by its probability, utility, and cost. The blue path is the optimal path from Section 2.3 that has a maximum expected utility. At time $18.0s$, $28.8s$, and $33.3s$ in Fig.5, it is shown that the proposed algorithm selects the optimal path that is close to the parking space currently being detected which is likely to be the true parking space. However, since the vehicle at $33.4s$ in Fig.6 has erred over ϵ_x and ϵ_θ, the proposed algorithm samples parking spaces again and re-generates path candidates to reach the right parking space. Eventually, the vehicle parks close to the true parking space with an error

$(0.09m, 0.05m, 0.57\,\text{deg})$. To check the capability of the proposed algorithm, we have evaluated it 100 times in the noisy environment. The results are verified by two criteria: average error and worst-case error. As shown in Table 3, The average error of the proposed algorithm was reduced by (74.2%, 66.6%, 62.7%) percent, and in worst-case was reduced by (55.8%, 60.4%, 59.6%) percent compared to planning without re-planning.

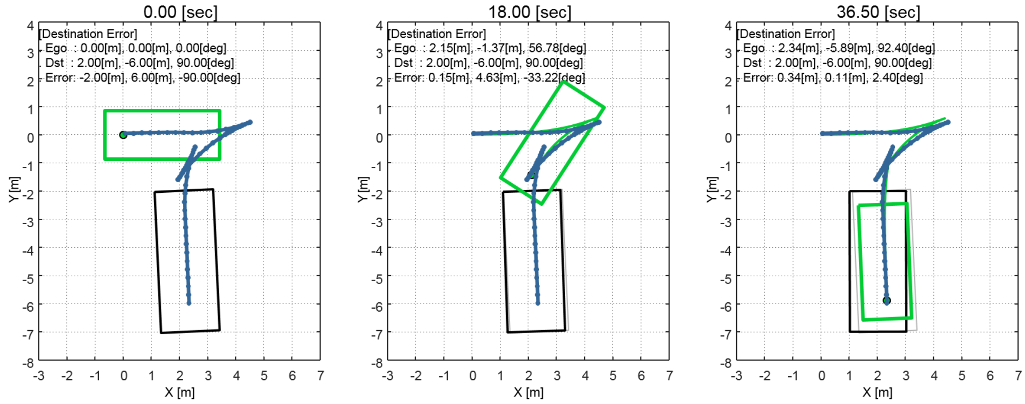

Figure 3: Automated parking without re-planning; the planner does not try to plan a path again even though the detected parking space is different from the one detected first.

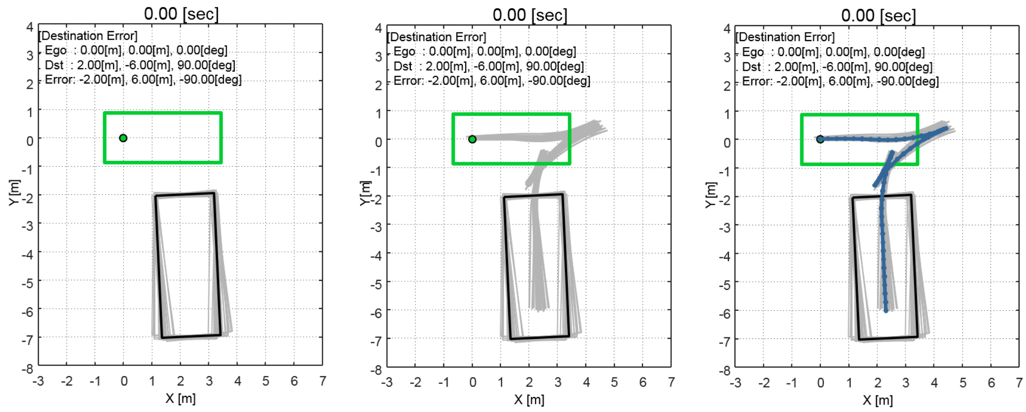

Figure 4: Overall process of the proposed algorithm from parking space sampling to optimal path selection

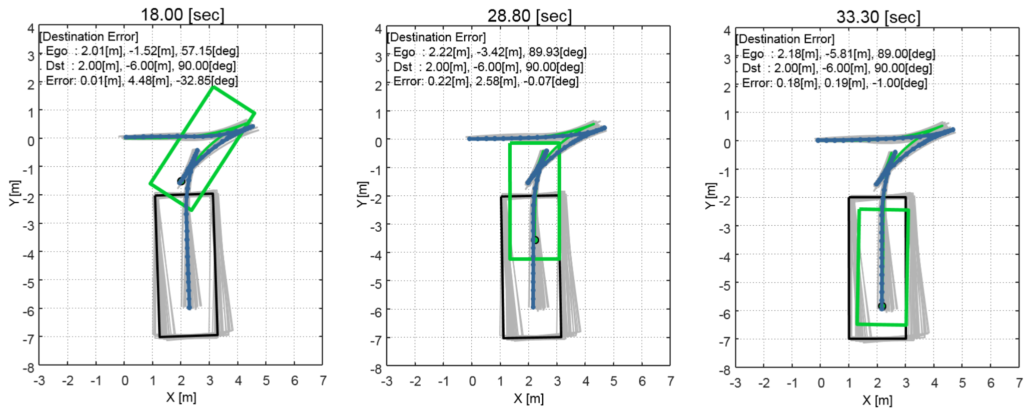

Figure 5: The effect of optimal path selection; the proposed algorithm selects the optimal path

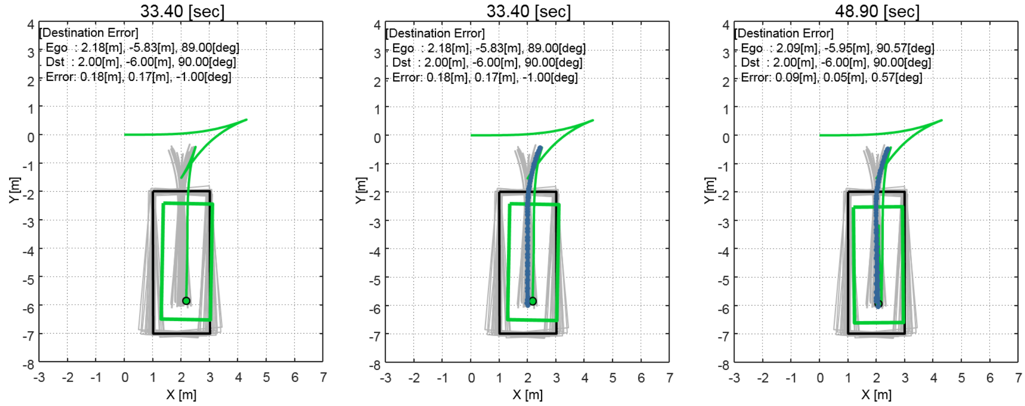

Figure 6: The effect of re-planning strategy; the proposed algorithm tries to re-sample parking spaces and re-generates path candidates

Table 3: The comparison of the average error between without re-planning and with re-planning

error	$x_{avg}(m)$	$y_{avg}(m)$	θ_{avg}(deg)	$x_{worst}(m)$	$y_{worst}(m)$	θ_{worst}(deg)
without replanning	0.31	0.18	2.33	0.42	0.32	3.81
proposed	0.08	0.06	0.87	0.19	0.13	1.54

4 Conclusion

In this paper, a re-plannable parking path planning algorithm is proposed, which operates in uncertain environments. The detected parking space from the perception system is sampled according to its probability for reaching actual parking space. Then, the parking path candidates are generated to reach each sampled parking spaces. At the end, the optimal parking path candidate is selected through the utility function that considers the true probability of the currently detected parking space and the consistency of the selected path candidate.

Simulation results show the capability and the feasibility of the proposed algorithm. The proposed algorithm can reach the actual parking space even though the input of the planner, which is the currently detected parking space, is continuously being changed. Through the proposed algorithm, the vehicle can be located in the actual parking space with reduced errors than the parking path planner without re-planning. Extensive simulations prove that the proposed algorithm is able to provide feasible paths for various noisy scenarios. Nevertheless, future work will address planner implementation subject to various sensor models and dynamic obstacles.

Acknowledgment

This work was financially supported by the BK21 plus program (22A20130000045) under the Ministry of Education, Republic of Korea, the Industrial Strategy Technology Development Program (No. 10039673, 10060068, 10042633, 10079961, 10080284), the International Collaborative Research and Development Program (N0001992) under the Ministry of Trade, Industry and Energy (MOTIE Korea), and National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (No. 2011-0017495).

References

[1] A. Bry and N. Roy, “Rapidly-exploring Random Belief Trees for motion planning under uncertainty,” in *2011 IEEE International Conference on Robotics and Automation*, pp. 723–730, May 2011.

[2] M. P. Vitus and C. J. Tomlin, “Belief Space Planning For Linear, Gaussian Systems In Uncertain Environments,” *IFAC Proceedings Volumes*, vol. 44, pp. 5902–5907, Jan. 2011.

[3] P. E. Missiuro and N. Roy, “Adapting probabilistic roadmaps to handle uncertain maps,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 1261–1267, May 2006.

[4] R. Alterovitz, T. Simeon, and K. Goldberg, “The Stochastic Motion Roadmap: A Sampling Framework for Planning with Markov Motion Uncertainty,” in *Robotics: Science and Systems III*, Robotics: Science and Systems Foundation, June 2007.

[5] A.-a. Agha-mohammadi, S. Chakravorty, and N. M. Amato, “FIRM: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements,” *The International Journal of Robotics Research*, vol. 33, pp. 268–304, Feb. 2014.

[6] B. Burns and O. Brock, “Sampling-Based Motion Planning With Sensing Uncertainty,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 3313–3318, Apr. 2007.

[7] M. F. Hsieh and U. Ozguner, “A parking algorithm for an autonomous vehicle,” in *2008 IEEE Intelligent Vehicles Symposium*, pp. 1155–1160, June 2008.

[8] W. Lim, J. Kim, K. Jo, Y. Jo, and M. Sunwoo, “Turning Standard Line (TSL) Based Path Planning Algorithm for Narrow Parking Lots,” Apr. 2015.

[9] K. Jolly, R. Sreerama Kumar, and R. Vijayakumar, “A Bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits,” *Robotics and Autonomous Systems*, vol. 57, pp. 23–33, Jan. 2009.

[10] J. Yoon and C. D. Crane, “Path planning for Unmanned Ground Vehicle in urban parking area,” in *2011 11th International Conference on Control, Automation and Systems*, pp. 887–892, Oct. 2011.

[11] L. Han and S. Mita, “Unified path planner for parking an autonomous vehicle based on RRT,” in *2011 IEEE International Conference on Robotics and Automation*, pp. 5622–5627, May 2011.

[12] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, “Path Planning for Autonomous Vehicles in Unknown Semi-structured Environments,” *The International Journal of Robotics Research*, vol. 29, pp. 485–501, Apr. 2010.

[13] K. Sakaeta, T. Oda, K. Nonaka, and K. Sekiguchi, “Model predictive parking control with on-line path generations and multiple switching motions,” in *2015 IEEE Conference on Control Applications (CCA)*, pp. 804–809, Sept. 2015.

[14] B. Li and Z. Shao, “A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles,” *Knowledge-Based Systems*, vol. 86, pp. 11–20, Sept. 2015.

[15] T. Kobayashi and S. Majima, “Real-time optimization control for parking a vehicle automatically,” in *IVEC2001. Proceedings of the IEEE International Vehicle Electronics Conference 2001. IVEC 2001 (Cat. No.01EX522)*, pp. 97–102, 2001.

[16] P. Zips, M. Bck, and A. Kugi, “Optimisation based path planning for car parking in narrow environments,” *Robotics and Autonomous Systems*, vol. 79, pp. 1–11, May 2016.

[17] M. Yamamoto, Y. Hayashi, and A. Mohri, “Garage parking planning and control of car-like robot using a real time optimization method,” in *(ISATP 2005). The 6th IEEE International Symposium on Assembly and Task Planning: From Nano to Macro Assembly and Manufacturing, 2005.*, pp. 248–253, July 2005.

[18] K. Kant and S. W. Zucker, “Toward Efficient Trajectory Planning: The Path-Velocity Decomposition,” *The International Journal of Robotics Research*, vol. 5, pp. 72–89, Sept. 1986.

[19] I.-K. Lee, M.-S. Kim, and G. Elber, “The Minkowski sum of 2d curved objects,” in *Proceedings of Israel-Korea Bi-National Conference on New Themes in Computerized Geometrical Modeling*, vol. 5, pp. 155–164, 1998.

[20] N. E. Jensen, “An Introduction to Bernoullian Utility Theory: I. Utility Functions,” *The Swedish Journal of Economics*, vol. 69, no. 3, pp. 163–183, 1967.

Authors

Seongjin Lee (S16) received a B.S. in Automotive Engineering in 2015 from Hanyang University in Seoul, Korea. From 2015, he has been working in the Automotive Control and Electronics Laboratory (ACE Lab), at the Department of Automotive Engineering at Hanyang University, doing research on implementation of autonomous cars. His work focuses on behavioral decision, trajectory planning and vehicle motion control for highly automated vehicles.

Wonteak Lim (S14) received a B.S. degree in Mechanical Engineering in 2013 from Hanyang University in Seoul, Korea, where he is currently working on attaining a Ph.D. degree with the Automotive Control and Electronics Laboratory (ACE Lab). His research interests include trajectory planning, vehicle control, and real-time embedded systems for intelligent transportation systems. His research includes trajectory planning algorithms, a system integration, and software platform design for autonomous vehicles.

Myoungho Sunwoo (M81) received a B.S. in Electrical Engineering from Hanyang University in 1979, and an M.S. in Electrical Engineering from the University of Texas at Austin in 1983, and a Ph.D. in System Engineering from Oakland University in 1990. He joined General Motors Research (GMR) Laboratories, Warren, and MI in 1985 and has worked in the area of automotive electronics and control for 30 years. During his nine-year tenure at GMR, he worked on the design and development of various electronic control systems for powertrains and chassis. Since 1993, he has led research as Professor at the Department of Automotive Engineering at Hanyang University. His work has focused on automotive electronics and controls such as modeling and control of internal combustion engines, design of automotive distributed real-time control systems, intelligent autonomous vehicles, and automotive education programs.