Path Planning of Car-like Mobile Robot in Cluttered Environments

Hyun-ki KWON, Woo-jin CHUNG
(*Dept. of mechanical Engineering, Korea University, Seoul 136-701, Korea*)

Abstract – **We have proposed the KPP (Korea University Path Planner) in our prior work. The KPP in Ref. [6] is the path planning scheme of a car-like mobile robot in a parking environment. The objective of this paper is to investigate the advantages of the KPP through both quantitative and qualitative analysis. For comparison, numerical simulations have been carried out by the application of the KPP and the conventional Probabilistic Roadmap approach. The Probabilistic Roadmap approach is one of the widely used path planning schemes owing to its superior performance. This paper shows that the KPP shows outstanding performance from the viewpoints of optimality and computational efficiency.**

Key words – ***path planning; car-like mobile robot; Korea University path planner; non-holonomic planning***

Manuscript Number: 1674-8042(2011)supp1.-0009-04

doi: 10.3969/j.issn.1674-8042.2011.supp1.003

1 Introduction

Path planning for the automatic parking of car-like vehicles is one of main issues for future automobiles. The path planning of a car-like robot is difficult because of its non-holonomic constraints. Moreover, the path planner must consider obstacle avoidance and the environment. The purpose of path planning is to generate paths that connect the initial and target points while satisfying the constraints.

There are two main forms of search concerning automated parking path planners. One of them is sampling-based search, and the other is grid-based search. In the DARPA Urban Challenge[1], various non-holonomic path planning schemes such as heuristic A * and Rapidly-exploring Random Trees (RRT) were exploited to deal with path planning for automated parking.

The Stanford team[2] and the CMU team[3] demonstrated heuristic A * search. The MIT team[4] used the RRT scheme for path planning in the parking zone. RRT was presented in Ref. [5] and is suitable for non-holonomic path planning.

Another sampling-based method is the Probabilistic Roadmap Method (PRM)[6]. PRM constructs roadmap graphs by connecting sampling points.

The KPP method was presented in Ref. [7-9]. KPP computes the collision area of motion sets and provides non-holonomic paths. Details about the above methods and a comparative analysis will be given in section 2 and section 3.

Although various path planners have been proposed, very few attempts have been made regarding quantitative and qualitative analysis about existing path planning methods. Moreover, KPP needs to show its advantages and usefulness in a clear way. Therefore, the purpose of this study is to quantitatively analyze KPP in comparison with RRT and establish the former's superiority. Path planning simulation in a parking environment with the KPP and RRT methods was performed for this purpose.

This paper is organized in four sections. In section 2, we briefly explain the KPP and RRT methods; this is followed by a qualitative analysis of various path planners. In section 3, the results of the path planning simulations are presented. We conclude in section 4 with closing remarks and future plans for this research.

2 RRT and KPP methods

2.1 RRT

RRT method was presented in Ref. [5]. This scheme generates nodes by random sampling. Then, it connects the generated nodes to make a tree that involves the target and initial points.

Fig. 1 illustrates RRT scheme. This scheme performs simple iterations that extend the tree T by adding a new node qnew using a randomly-selected configuration. If qrand is selected by random sam-

* **Received:** 2011-05-18
Project supported: This work was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2009-(C1090-0902-0007)); Partly by the MKE under the Human Resources Development Program for Convergence Robot Specialists and by Basic Science Research Program through the NRF funded by the MEST (2010-0022609)
Corresponding author: Woo-jin CHUNG (smartrobot@korea.ac.kr)

pling, then the collision checking of qrand is performed. If qrand is confirmed to lie in a collision-free space, then RRT scheme adds the new node qnew that is slightly away from the previous node q. RRT scheme extends the node until the tree T includes the target point.

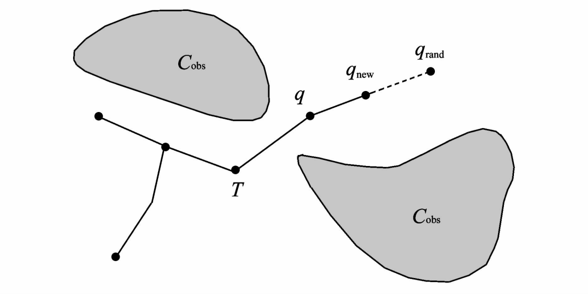

Fig. 1 The extend operation of RRT's method[3]

RRT is one of the most widely used path planners and is practically useful in non-holonomic conditions and complex environments. Because RRT is based on random sampling, the quality of the resultant path is not guaranteed.

2.2 KPP

KPP method was presented in Ref. [1]. It uses two basic motions of a car-like robot. One is the translation motion (TM), and the other is the rotation motion (RM) with the maximum steering angle for reducing the computational cost by simplifying the robot's motion patterns. Then, KPP method performs collision checking for calculating reachable regions.

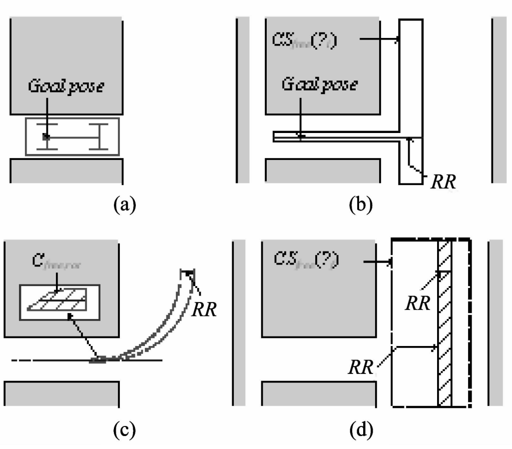

Fig. 2 Computing the reachable region

Fig. 2 shows the procedure for computing the reachable region (RR). Fig. 2(a) shows the initial pose and also indicates that KPP finds the reachable region from the target point and not the starting point. Fig. 2(b) illustrates $RR_{i,1}$, which means the reachable region under one translation motion. Fig. 2(c) shows how to calculate the RR of rotational motion; the hatched area of Fig. 2(d) is $RR_{i,2}$, which means the reachable region under translation motion after the rotational motion.

As we observed earlier, one set of motions consists of TM, RM and TM.

The KPP method restricts motion owing to the initial assumption of two basic motions. The reachable regions of a car-like robot are maximised when the maximum steering angle is used. The number of maneuvers can be decreased in a narrow parking environment when the reachable regions are increased. Therefore, there is no limitation on the robot's allowable motion in KPP. And, the candidate paths computed by KPP contain all of the possible motions.

All the candidate paths have their own path cost that is computed by a cost function that we define later. The path for which the cost is the minimum is selected as the optimal path.

The KPP method finds the path from the target point that is narrower than the starting point; it generates a candidate path for optimality.

2.3 Qualitative comparison of path planners

As noted in section 1, there are many studies about path planning for car-like robots. Details and a discussion of various non-holonomic path planning methods are given below.

In the DARPA Urban Challenge[1], a grid-based A* search scheme with a heuristic cost function was presented by both Stanford[2] and CMU[3] teams. Always, D* with heuristic cost schemes performed very well in practical environments; however, the performance of this search scheme strongly depends on the heuristic cost function.

A sampling-based path planner is another way to deal with the non-holonomic path planning problem. The PRM and RRT methods are well known as sampling-based search methods. Sampling-based methods can generate non-holonomic paths by considering non-holonomic constraints during node extension. RRT has a low computational cost and is widely used in recent studies. However, because it depends on random sampling, it cannot easily generate a path in a narrow environment for sampling-based methods. As shown in many studies such as Ref. [10] and [11], RRT is a useful solution for non-holonomic path planning.

KPP was proposed in Ref. [7-9]. It is powerful in parking environments because it starts to search for a path from the goal configuration. Also, KPP provides candidate paths for optimization. A major drawback of KPP is the growth in the computational cost that is caused by an increase in the number of motion sets.

As mentioned above, there are various non-holonomic path planners. We can consider the RRT method as the current state-of-the-art solution. Therefore, we have compared KPP with RRT using simulations in order to show the advantages of KPP and analyse the qualitative characteristics of KPP.

3 Simulations and results

3.1 Simulation environments

Fig. 3 illustrates the simulation environments. The environment of a parking lot can be modeled as two cases, namely, garage parking and parallel parking. Two parameters, a and b, are needed to determine the parking space. The width of the road is represented by a, and the width of the goal is b. The vehicle parameters are adopted from a midsized car. The length and the width are 4 860 and 1 800 mm.

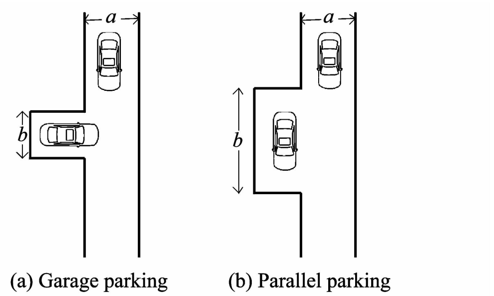

Fig. 3 Simulation environments

We define the cost function for calculating the optimal path as shown in (1).

$$C_{\text{total}} = \alpha(1/D_{\text{min,obs}}) + \beta N_{\text{str}} + \gamma N_{\text{mnvr}} + \delta D_{\text{trvl}}$$

In the cost function, $D_{\text{min,obs}}$ is the minimum distance between poses on the path and obstacles, N_{str} is the number of steering angle changes, N_{mnv} is the number of maneuverings, and D_{trvl} is the travel distance. α, β, γ and δ are the weights. The minimum cost path is optimal. The simulations shown here were performed on a 2.33 GHz Intel Core™2 duo.

3.2 Simulation results

Fig. 4 shows the computing time to obtain solution paths when the width of the parking lot b is changed from 3.2 m to 2.4 m. It is evident that RRT planner spends excessive time when b is small. The computing time of RRT increases to 57 seconds when b becomes smaller than 3 m. On the other hand, KPP successfully provides a solution with an acceptable computing time under 2 seconds.

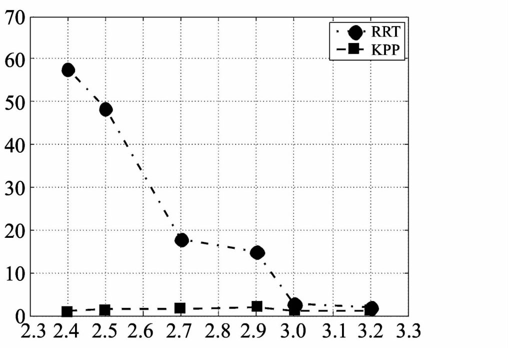

Fig. 4 Computing time versus the width of the parking lot b

The reason why the computing time of RRT increases when b becomes smaller is that RRT starts to find a node from the starting point. If the width of the target point becomes smaller, the chance of finding a node near the target point becomes smaller. However, because KPP starts to find a path from the target point, its computing time is independent of the width of the target point.

Fig. 5 shows the resultant path generated by the RRT scheme in the same environment as in Fig. 4. In Fig. 6, RRT has generated an inefficient path with overlapped nodes and additional changes in the steering angle. Thus, path planners based on random sampling such as RRT can generate an inefficient path, as shown in Fig. 5.

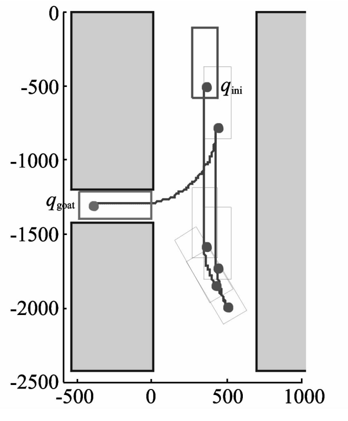

Fig. 5 Resultant path under RRT for a = 7 m and b = 2.4 m

Fig. 6 shows the computational times of KPP and RRT when a = 5 m and b = 2.5 m. The computing time of RRT changed every time while that of KPP was relatively constant. As shown in Fig. 7, the standard deviation under KPP was 0.206 1 and that under RRT was 44.696 9. Therefore, it can be said that KPP has better dependability than RRT in terms of the computational time.

Tab. 1 shows the cost of each path generated by KPP and RRT. The cost of the safest path generated by either method is shown in Tab. 1, which reveals that the cost of the safest path uneer RRT is 210.168, and that under KPP is 178.96. The shortest distance from the obstacle under KPP (resp., RRT) is 57.531 cm (resp., 48.302 cm). Thus, it is true that KPP generates a safer path than RRT.

Tab. 1 Costs of paths. $D_{min,obs}$

	$D_{min,obs}$	N_{str}	N_{mnvr}	D_{trvl}	C_{total}
RRT	48.302	3	0	1373.3	210.168
KPP(obs)	57.531	5	0	1407.2	178.96
KPP(str)	48.302	3	0	1373.3	-
KPP(trvl)	5.576	5	2	1291.8	-

KPP has generated a better path than RRT from the viewpoint of the minimum distance between the obstacle and the shortest path. Also, the paths of KPP are easier to control than those of RRT from the viewpoint of changing the steering angle.

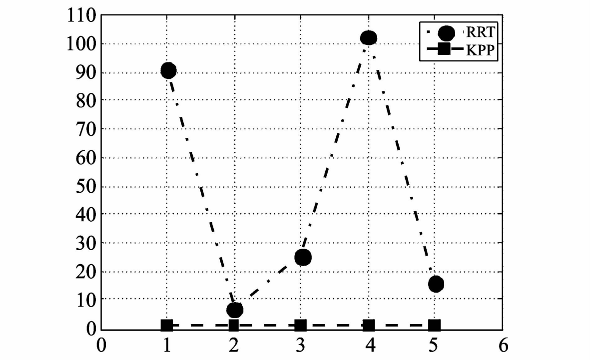

Fig. 6 Comparison of KPP and RRT in terms of the computational time

The safest path generated by KPP costs 26.79 cm, while the safest path under RRT corresponds to 4.939 cm. The latter path needs more than two changes in terms of maneuvers, while the path under KPP needs only one change. Thus, we can see that KPP affords better simplicity of control. Also, KPP has generated a shortest path of 1 913 cm, while RRT has generated a shortest path of 2 436.6 cm. Because KPP has generated the shorter path, we can see that KPP enables superior solutions than RRT.

4 Conclusions

In this paper, we compared KPP with RRT using path-planning simulation in parking environments. The comparison was performed in terms of the computational time, generality, simplicity, and optimality. Both KPP and RRT can successfully generate a path in both garage parking and parallel parking. Hence, we found that both have generality. We also found that the computational times of KPP are smaller than those of RRT for narrow target points. In addition, we noted that KPP affords better simplicity and optimality than RRT. In conclusion, KPP is a powerful non-holonomic path planner for car-like robots in automated parking.

References

[1] Martin Buehler, Sanjiv Singh, Karl Iagnemma. The darpa urban challenge: autonomous vehicles in city traffic. Springer, 2009.

[2] Dmitri Dolgov, Sebastian Thrun, Michael Montemerlo, et al. Path planning for autonomous vehicles in unknown semi-structured environments. *The International Journal of Robotics Research*, 2010, 29(5): 485-501.

[3] Maxim Likhachev, Dave Ferguson. Planning long dynamically-feasible maneuvers for autonomous vehicles. *The International Journal of Robotics Research*, 2009, 28(8): 933-945.

[4] Yoshiaki Kuwata, Gaston A. Firore, et al. Motion planning for urban driving using RRT. 2008 IEEE RSJ International Conference on Intelligent Robots and System, September, 2008.

[5] LaValle S M, Kuffner J J. Randomized kinodynamic planning. IEEE International Conference on Robotics and Automation, 1999:473-479.

[6] Kavraki L E, Svestka P, Latombe J C, et al. Probabilistic roadmaps for path planning in high dimensional configuration spaces. *IEEE Trans. on Robotics and Automation*, 1996, 12, (4): 566-580.

[7] Dalhyung Kim, Woojin Chung Shinsuk Park. Practical motion planning for car-parking control in narrow environment. IET Control Theory Appl., 2010:1-11.

[8] Dalhyung Kim, Woojin Chung. Motion planning for car-parking using the slice projection technique. 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1050--1055, Nice, France, September, 2008:22-26.

[9] Dalhyung Kim, Woojin Chung. Car-parking motion planning by the motion space approach. Proceedings of the 13th International Conference on Advanced Robotics (ICAR2007), Jeju, Korea, August 2007: 241-246.

[10] LaValle S M. Planning algorithms. Cambridge University, 2006:238.

[11] Howie Choset, Kevin M. Lynch, et al. Principles of robot motion. MIT Press, Cambridge, 2005.