Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

OGBPS: Orientation And Gradient Based Path Smoothing Algorithm For Various Robot Path Planners

Xiaotong Wang, Biao Hu, Meng Zhou*

***Abstract*— It is significant to plan a smooth trajectory for high-speed wheeled mobile robots working in a cluttered environment. The trajectory generated by most of the previously proposed path planners is not smooth enough for robot motion, especially under kino-dynamic constraints. An improved smoothing algorithm is proposed in this work as a solution for most of the previously proposed path planners to deal with the rugged paths, which may cause abrupt and angular turns of robots. The improved solution we proposed could be applied to many mainstream path planners (like Theta*, A*, RRT, RRT*, RRT#, SORRT*, PRM) as a post-smoothing algorithm, which is called orientation and gradient-based path smoothing (OGBPS). The OGBPS algorithm is derived from both orientation-angle-based and gradient-based path deformations to obtain a high-quality path. The objective of path deformations in this work is to improve path smoothness, lower maximum curvature and path length. Sufficient simulation experiments are well conducted to demonstrate the effectiveness of our approach. It is verified that the proposed algorithm can improve the quality of the previous path while respecting the kino-dynamic constraints through experiments. The simulation results indicate the advantages (smaller maximum curvature and smaller path length) of the proposed algorithm compared with several state-of-the-art smoothing algorithms.**

I. INTRODUCTION

Path planning has always been a hot research issue in robotics field. Classical mainstream path planning algorithms have been utilized very widely for decades. Lots of path planning algorithms have been proposed by many researchers [1]–[4]. However, most of the state-of-the-art path planners such as A* algorithm [2], probabilistic road-map (PRM) [4] and rapidly exploring random tree (RRT) [3] have the potential limitations on abrupt and angular turns, which increases the difficulties of robot motion. Since the wheeled mobile robots at high velocities cannot take turns abruptly when sharp path or angular turns are given, at this situation, they have to stop and reorient. Hence, a path smoothing algorithm can be utilized to reduce abrupt turns, improve the path quality while respecting the kino-dynamic constraints at the same time in these scenarios. A smooth path is definitely requisite for robot motion planning which enables the robot to move at nearly constant velocity.

To improve the path quality generated by state-of-the-art path planning algorithms, many researchers have presented numerous feasible techniques in recent years. The existing mainstream post-smoothing algorithms can be divided into the different categories which is as following: combining different planner results [5-6], smoothing through utilizing Bézier curve [7-8], gradient asymptotical optimization strategy [9-10] and others [11-12]. A novel hybrid algorithm [5] is proposed for high-dimensional freedom path smoothing, which merges the paths of different path planners via classical graph search to get a new feasible path with higher quality especially in aspects of path length, smoothness and energy. The authors verified the effects of the algorithm in the experiments with configuration space up to 12 degrees of freedom (DOFs). A hybridization algorithm variant [6] is proposed by Ryan Luna, which combines parts of two or more path solutions into one form a shorter path and makes the algorithm asymptotically optimal. It converges to the shortest path over time. Due to the combination of multiple path planning methods, this kind of algorithm has the disadvantage of long computation time. A new curve smoothing method is proposed [7]. The method is proposed for a path planner with curvature constraints. The location of the node from the planner will be updated. When the trajectory is stable, Bézier curve is adopted. For the purpose of meeting the optimal requirements, the position of each control point curvature constraint is determined. However, the path smoothed by this method sometimes may be too close to obstacles in the environments to guarantee safety. An efficient and analytical continuous curvature path smoothing algorithm [8] has been proposed by Kwangjin Yang et al, which is based on parametric cubic Bézier curve. The algorithm only needs to obtain the information of the maximum curvature constraint. Therefore, the algorithm has high computational efficiency and is easy to achieve. The results confirm that the algorithm is effective in computing continuous curvature paths and satisfies an upper bound condition with bounded curvature constraints, the generated trajectory requires less control to track and minimize control input variables. Although the algorithm satisfies the maximum curvature constraints, the distance from the obstacles in the environment are not considered in the smoothing process. Covariant Hamiltonian Optimization for Motion Planning (CHOMP) optimization algorithm [9] is proposed, which is a path refinement algorithm using co-variant gradient technique to improve the quality of sampling-based trajectory. Their optimization technique not only optimizes the trajectory in the high configuration space dynamics but also is feasible in converging over a wider range of input paths with comparison to previous path optimization algorithm. Their method works greatly with holonomic robot systems,

*This work is supported by the National Natural Science Foundation of China under Grant 51805021 and Grant 91848103.

*Corresponding author.
The authors are with College of Information Science and Technology, Beijing university of chemical technology, Beijing, China
zhoumeng@mail.buct.edu.cn(M.Zhou)

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

but doesn't fit as well as non-holonomic systems, such as wheeled robots. GRIPS post-smoothing algorithm has been proposed in [10], which is based on path gradient asymptotical optimization. The gradient-informed path smoothing (GRIPS) algorithm uses the gradients of obstacle distance to adjust the path. The GRIPS algorithm works well most of time, but sometimes doesn't perform perfectly when the original trajectory has large radian turns. Short-cutting smoothing algorithm [11] has been proposed by D.Hsu et al, which has been employed to a robot manipulator in a scenario cluttered with obstacles. The algorithm deletes redundant path points and connects directly by steer function. However, short-cutting algorithm deletes so many path points which may not be redundant that the path through short-cutting seemed weird in several cases.A extension approach of path smoothing [12] for a variety of robot path planners is presented, which maintains straight paths of the robot straight and smooths only at the sharp turns of path. But the extension approach is only advantageous in narrow passages environment.

The previously proposed path smoothing techniques have some major limitations, i.e., several post-smoothing algorithms may be insufficient in the accuracy of the optimized path goal point in several cases, moreover, some of the smoothing path algorithms have not take topological constraints into account.

The proposed approach is inspired by the work of CHOMP [9] algorithm and GRIPS [10] algorithm, which are both based on path gradient asymptotical optimization strategy. The advantage of asymptotic optimization based on distance gradient is combined with the advantage of asymptotic optimization based on orientation angle in proposed OGBPS algorithm. Simulation experimental results confirm that OGBPS algorithm is superior to other smoothing algorithms while subjecting to kino-dynamic and topological constraints.

The contributions of this paper are summarized as following. The difference of orientation angle has been introduced into the path-smoothing algorithm in a novel way. The experimental results prove that the effect of OGBPS algorithm satisfies compared with state-of-the-art smoothing algorithms especially in the aspect of success-failure rate and path quality (path length and maximum curvature). Moreover, The OGBPS algorithm can cooperate with various mainstream path planners: The proposed OGBPS algorithm can smooth the path well generated by miscellaneous path planners (like Theta*, A*, PRM, RRT, RRT*, RRT#, SORRT*), which allows it to be a good choice for various mobile robot applications.

The overall structure of this paper is as follows. Problem overview is described in Section II. Then, OGBPS algorithm design is presented in Section III. Next, simulation results and analysis are presented in IV. The conclusion and future work are elaborated in Section V.

II. PROBLEM OVERVIEW

Wheeled mobile robots utilize various path planners when the messages of goal state are given(randomized sampling based path planners [4] or grid-based search algorithm [2], [13]) to compute a path from start state to goal state. A path P consists of a sequence of vertices state, in other words, a collection of the start states S_s to the goal state S_g, $P = \{S_1 = S_s, \ldots, S_N = S_g\}$. The connection of each state is configured by different steer functions to guarantee the kinematic and kino-dynamic constraints of wheeled robots, which are acknowledged as a two point boundary value problem (2PBVP) [14]. The motion steer function, which is also called two-point BVP solver, is to plan a series of feasible and stable motions of two adjacent path points in the configuration space and get a feasible continuous trajectory. Each state is computed by path planners from open motion planning library (OMPL) [15]. The parameters of the steer functions are formulated with respect to the robots' constraints. The rugged path with angular and sharp turns brings difficulties to motion steer functions. According to the previous statement, the objective of OGBPS algorithm is to apply between path planners and motion steer functions to smooth the path with less angular turns.

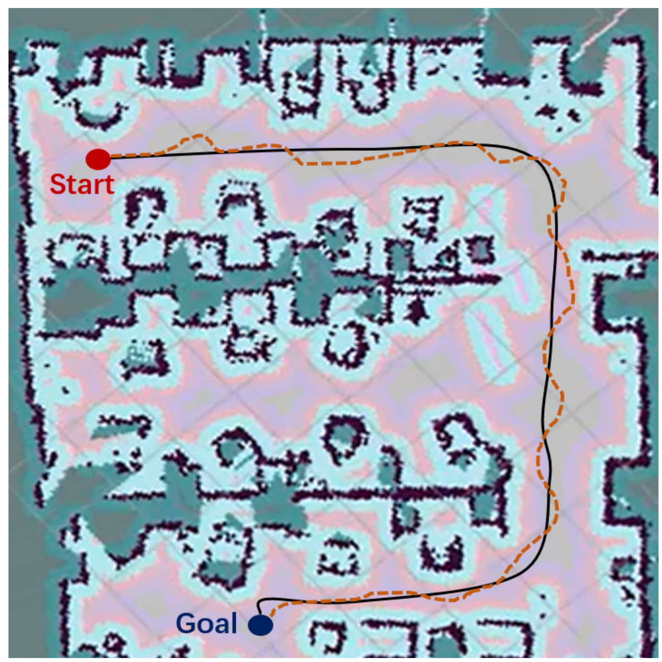

Fig. 1: Schematic diagram of the path (solid line) smoothed by OGBPS algorithm and the original path (dashed line) generated by path planners.

III. OGBPS ALGORITHM DESIGN

The algorithm proposed in this work is to smooth the path while decreasing the path length, higher and maximum curvature in the path. The OGBPS algorithm first determines the moving degree of each point according to the gradient of the distance between the path points and the obstacles as well as the difference of the orientation angle between the adjacent path points, which goes through K iterations. During the K iterations, if a local minimum of a non-path point is found in the path, add it to the path. Then the classical Bellman-Ford algorithm [16] is utilized to select and delete redundant vertices in the path. The goal of utilizing Bellman-Ford algorithm is reducing the number of 2PBVP problems that need to be solved as well as computing time.

The Euclidean distance of obstacle between robots can be computed as follows:

$$D[m] = \min_{o \in O} \|o - m\|_2. \tag{1}$$

where O indicates the collection of coordinates of occupied grid cells, while m depicts the robot's path continuous vertex position.

The gradient $\bigtriangledown D$ of the configuration search space is defined as follows:

$$\bigtriangledown D[m] = \begin{pmatrix} \dfrac{D[m.x-\varepsilon, m.y] - D[m.x+\varepsilon, m.y]}{2\varepsilon} \\ \dfrac{D[m.x, m.y-\varepsilon] - D[m.x, m.y+\varepsilon]}{2\varepsilon} \end{pmatrix} \tag{2}$$

where a pair of coordinates x_i, y_i is utilized to indicate the horizontal and vertical positions on the grid map respectively, while sufficiently small $\varepsilon > 0$. The objective of using $\bigtriangledown D$ is to modify each path vertex position as far as possible from obstacles.

The path deformation size is designed as follows:

$$S = \frac{(\eta + \lambda[m]) \bigtriangledown D[m]}{D[m]} \tag{3}$$

where η is a parameter which can be set to adjust the deformation amplitude. During each iteration, η is discounted by constant discount factor γ. The objective of introducing $\bigtriangledown D[m]$ is to smooth the path as far away from the obstacles as possible while taking orientation angle difference into account. The deformation parameter $\lambda[m]$ based on orientation angle difference to adjust the deformation amplitude is defined as follows:

$$\lambda[m] = \frac{\beta|(\theta_{i+1} - \theta_i)|}{\pi} \tag{4}$$

where β is a parameter which can be set to adjust the deformation amplitude and θ indicates the orientation angle of each vertex position. The orientation angle θ_i (as shown in Fig.2) is defined as follows:

$$\theta_i = arctan\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right) \tag{5}$$

The objective of introducing $\lambda[m]$ is to make different degree adjustments depending on the orientation angle difference of the path vertices and lower the higher curvature producing angular and sharp turns. The path with higher curvature which producing angular turns brings the wheeled robots physical limitations on accelerations and velocities. If the distance to the obstacles is more closer and the orientation angle difference is more larger, the amplitude of adjustment is more larger.

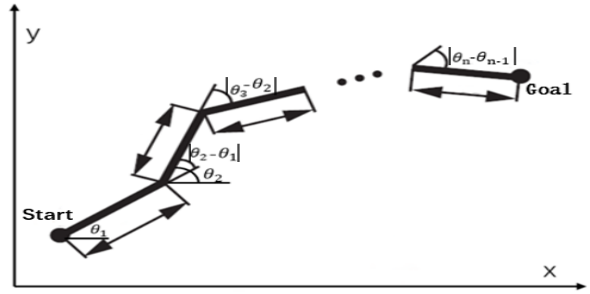

Fig. 2: Schematic diagram of orientation angle $\theta_i, i \in [1, n]$.

As stated above, the pseudocode of OGBPS algorithm is presented as follows:

Algorithm 1 OGBPS smoothing algorithm

function OGBPS ALGORITHM(β, P, η)
  $\beta \leftarrow \beta_0, \eta \leftarrow \eta_0$
  for $k = 1...K$ **do**
    for $i = 1...n$ **do**
      calculate $\bigtriangledown D[p_i]$
      calculate $\lambda[p_i] = \dfrac{\beta|(\theta_{i+1} - \theta_i)|}{\pi}$
      calculate $S_i = \dfrac{(\eta + \lambda[p_i]) \bigtriangledown D[p_i]}{D[p_i]}$
      $P_i \leftarrow P_i - S_i$
    end for
    $\eta \leftarrow \gamma\eta$
    $update(P)$
    $update(\theta_i)$
    for $q = \in (p_i, p_{i+1})$ **do**
      if $D[q]$ *is local minimum* **then**
        $P \leftarrow APPEND(P, q)$
      end if
      updateP
      updateθ_i
    end for
    return P
  end for
  function BELLMAN-FORD ALGORITHM(P)
    Build direct acyclic graph $G = (Vertices, Edges)$
    return *monophyletic shortest path P*
  end function
  return P
end function

When the original path has been processed by the OGBPS smoothing algorithm, steer functions are utilized to generate steady motion. Reed Shepp steer function [17] is utilized to deal with the kino-dynamic feasibility in this work, which is a steering technique generating constant tangential velocity for car-like vehicles or wheeled robots. The motion planning between two adjacent path vertex points is implemented by steer function. The steer function they present computes motion with upper-limited curvature and upper-limited curvature derivative. Reeds Sheep steer function is a classical

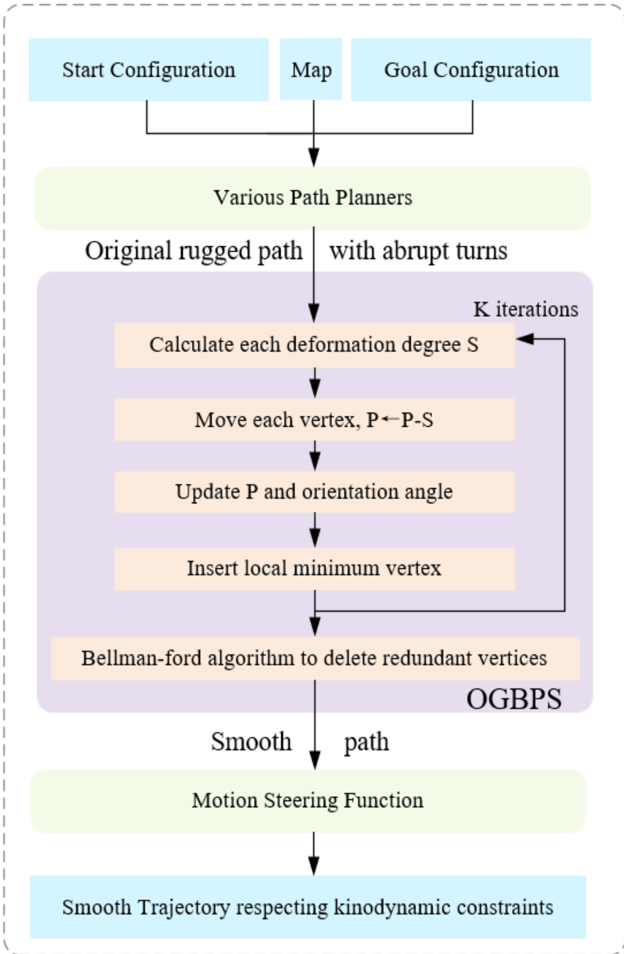

Fig. 3: The proposed OGBPS algorithm can cooperate with most of the mainstream global path planners like A* algorithm, Theta*, Probabilistic Road-map, RRT, RRT*, SORRT* and RRT#.

motion planning function which can steer both forward and backward utilized especially for non-holonomic system like wheeled mobile robots while respecting kino-dynamic constraints. Reeds Shepp steer function generates continuous trajectories for the kinematic system which is as follows:

$$\begin{Bmatrix} x' \\ y' \\ \alpha' \\ k' \end{Bmatrix} = \begin{Bmatrix} cos\alpha \\ sin\alpha \\ k \\ 0 \end{Bmatrix} d + \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{Bmatrix} d_k \tag{6}$$

where x, y indicates the midpoint position of wheel rear axle, α indicates the heading angle of robots, k indicates the curvature. The operator $(')$ denotes the respective derivative with respect to the arc length. The inputs of steer function are the heading direction d and the change of curvature d_k.

IV. SIMULATION RESULTS AND ANALYSIS

All experiments are well performed in C++. QT platform is utilized as visualization platform of output results. With respect to the path generation, we utilize the mainstream path planners and path smoothing algorithms from the accomplishment of Open Motion Planning Library (OMPL) [15]. The simulation environment is designed as a grid map which is consisting of square cells. Each grid cell has two status, occupied or free. Each path smoothing algorithm has four result status, path not found, missed the exact goal, collided and correct. The performance of different post-smoothing algorithms is evaluated by four metrics, the planning accuracy rate (the planning status), maximum curvature, path length and planning time. Path length and maximum curvature effect the actual traversal time of a path. Following longer paths with higher curvature is time-consuming, because the angular and sharp turns will lead to frequent acceleration and unstable motion.

As shown in Fig.4, while comparing with Theta*, which is recognized as smoothing A* algorithm, the angular path generated by A* algorithm through OGBPS smoothing algorithm is significantly smoother and more reasonable. The parameters of all experiments presented in this work are set to: $\beta = 10, \eta = 0.5, K = 5, \gamma = 0.8$. The radius of wheeled robots is set to 3.5m in simulation environment. A random environment of 50*50m is utilized to validate the effects of OGBPS algorithm. OGBPS algorithm effectively avoids as many turns as possible. Simulation results indicate that the OGBPS algorithm can be effectively applied to grid-based path planners (like A*) and obtain a smooth path with less abrupt and angular turns.

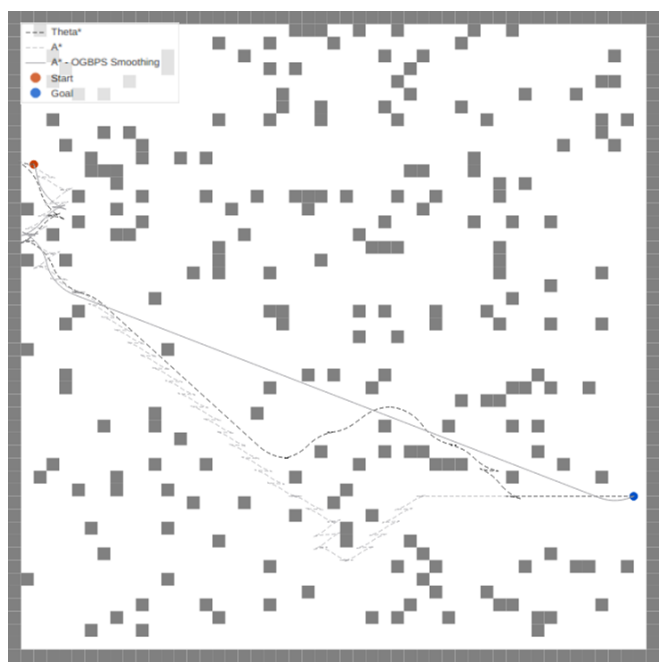

Fig. 4: The simulation experimental results of different path generated by A* path planner, Theta* path planner and the path which generated by A* path planner smoothed through OGBPS smoothing algorithm.

As shown in Fig.5, a self-created corridor environment of size 50m × 50m is utilized for visual comparison. Result

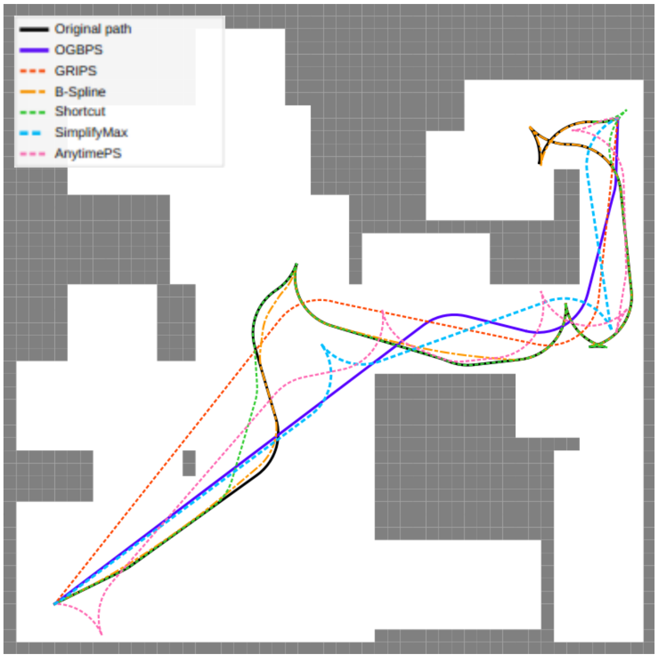

Fig. 5: Visual comparison of different path smoothing algorithms (GRIPS, B-Spline, Shortcut, Simplify Max, Anytime PS) utilized in the same path generated by RRT algorithm with Reeds-Sheep steer function through experiment.

indicates the actual effects of the OGBPS algorithm in the path generated by sampling-based path planning algorithm RRT while comparing with other path smoothing algorithms (B-spline which resembles with [8], Simplify-Max, Shortcut [11], Anytime PS [6], GRIPS [10]). The results show the advantages of combining our post-smoothing approach with the advantages of fast computation of sampling-based path planning algorithm. It is obvious that the proposed OGBPS algorithm surpasses other path smoothing algorithm in the visual comparison simulation experimental results due to reducing angular and sharp turns.

The plots in Fig.6-8 summarize the performance metrics of 6 different post-smoothing algorithms respectively on path length, maximum curvature and planning time (the plot results have removed planning status of collided and missed the exact goal). The experiments are implemented in 10 different random generated corridor environments which are resemble to Fig.5 and size of 50m*50m. Each post smoothing algorithm cooperates respectively with 7 different path planners (A* algorithm [2], Theta* [13] , Probabilistic Road-map (PRM) [4] , RRT, RRT*, SORRT*, RRT# [18]) 10 times. Each dot of different colors in every column represents separately a path result generated and smoothed by corresponding path planner and path smoothing algorithm in each random-generated environment. According to Fig.6-8, the conclusions can be drawn as following: OGBPS, Simplify-Max and GRIPS outperform other path smoothing algorithms in terms of path length; OGBPS and GRIPS have obvious advantages in terms of maximum curvature; most algorithms perform similarly in terms of planning time, except that OGBPS and GRIPS have a slightly longer planning time. Table I elaborates the detailed numeral statistics results of 70 experiments correspond to the plots of Fig.6, Fig.7 and Fig.8. The detailed numeral statistics confirm that OGBPS algorithm outperforms other smoothing algorithm especially on the aspect of planning accuracy rate. From what has been analyzed above, the plot and table turn out that OGBPS algorithm optimizes the path to the more high-quality than other post-smoothing algorithms according to performance metrics on path length and maximum curvature.

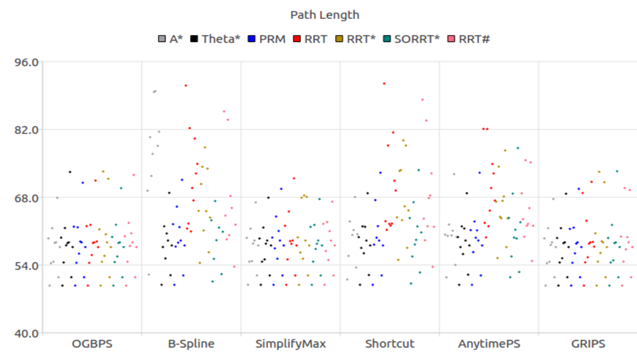

Fig. 6: Visual comparisons between various smoothing algorithms on the performance of path length.

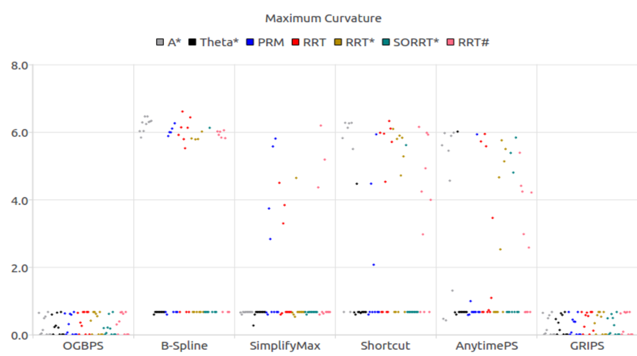

Fig. 7: Visual comparisons between various smoothing algorithms on the performance of maximum curvature.

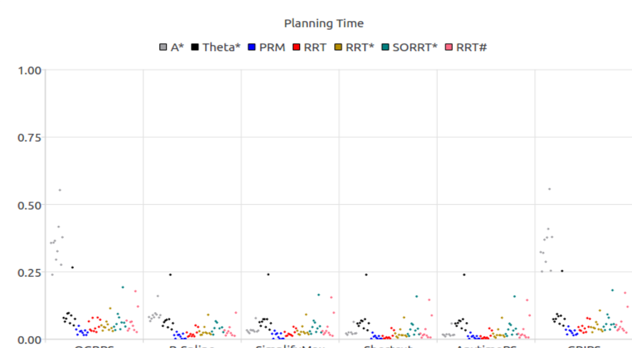

Fig. 8: Visual comparisons between various smoothing algorithms on the performance of planning time.

TABLE I: Detailed statistics summarize the results of 6 different smoothing algorithms cooperating with 7 different path planners 10 times respectively. Metrics on Max-Curvature, Path Length and Planning Time are shown in the format $mean \pm std\ deviation$

	OGBPS	B-Spline	SimplifyMax	Shortcut	AnytimePS	GRIPS
Path not found	**0**	**0**	**0**	**0**	**0**	**0**
Missed exact goal	**0**	**0**	**0**	**0**	**0**	**0**
Collided	**0**	**0**	4	**0**	3	1
Correct	**70**	**70**	66	**70**	67	69
Max-Curvature	**0.29 ± 0.29**	3.36 ± 2.71	1.26 ± 1.47	2.77 ± 2.44	2.30 ± 2.18	0.31 ± 0.29
Path Length (m)	**58.33 ± 5.93**	66.65 ± 13.19	**58.31 ± 5.39**	63.53 ± 9.01	62.85 ± 7.49	**58.32 ± 5.99**
Planning Time (s)	0.10 ± 0.12	0.04 ± 0.04	0.04 ± 0.04	**0.03 ± 0.04**	**0.03 ± 0.04**	0.10 ± 0.12

V. CONCLUSION AND FUTURE WORK

In this study, a path smoothing method which called OGBPS has been presented, which utilizing the path asymptotical optimization based on orientation angle difference and gradient of distances, while it is subject to kinodynamic constraints by Reed Shepp steer function. Abundant comparative experiments have been implemented well, and the experimental results demonstrate that OGBPS algorithm outperforms other state-of-the-art smoothing algorithms in the aspect of planning accuracy rate and path quality (path length and maximum curvature). As experimental diagram shows, the path through smoothing algorithm obtains higher-quality and seems more reasonable when compared with the original path and the path smoothed by other state-of-the-art post-smoothing algorithms.

In the future, we hope to combine the core step of OGBPS algorithm with A* path planner (or other similar grid-based search path planner), not as a post-smoothing algorithm any more.

REFERENCES

[1] Jean Arcady Meyer and David Filliat. Map-based navigation in mobile robots: A review of map-learning and path-planning strategies. *Cognitive Systems Research*, 4(4):283–317, 2003.

[2] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[3] Steven M. Lavalle and James J. Kuffner. Randomized kinodynamic planning. *International Journal of Robotics and Research*, 15(5):378–400, 1999.

[4] Sun Zheng, David Hsu, Tingting Jiang, Hanna Kurniawati, and John H. Reif. Narrow passage sampling for probabilistic roadmap planning. *IEEE Transactions on Robotics*, 21(6):1105–1115, 2005.

[5] Barak Raveh, Angela Enosh, and Halperin Dan. A little more, a lot better: Improving path quality by a simple path merging algorithm. *IEEE Transactions on Robotics*, 27(2):365–371, 2011.

[6] Ryan Luna, Ioan A. Sucan, Mark Moll, and Lydia E. Kavraki. Anytime solution optimization for sampling-based motion planning. *2013 IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[7] Reinis Cimurs, Jaepyung Hwang, and Il Hong Suh. Bezier curve-based smoothing for path planner with curvature constraint. In *IEEE International Conference on Robotic Computing*, 2017.

[8] Kwangjin Yang and S Sukkarieh. An analytical continuous-curvature pathsmoothing algorithm. *IEEE Transactions on Robotics*, 2010.

[9] N Ratliff. Chomp:gradient optimization techniques for efficient motion planning. *IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[10] Eric Heiden, Luigi Palmieri, Sven Koenig, O. Arras Kai, and Gaurav S. Sukhatme. Gradient-informed path smoothing for wheeled mobile robots. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[11] D. Hsu, J. C. Latcombe, and S. Sorkin. Placing a robot manipulator amid obstacles for optimized execution. In *IEEE International Symposium on Assembly and Task Planning*, 1999.

[12] Abhijeet Ravankar, Ankit A. Ravankar, Yukinori Kobayashi, and Takanori Emaru. Path smoothing extension for various robot path planners. In *International Conference on Control*, 2017.

[13] Kenny Daniel, Alex Nash, Sven Koenig, and Ariel Felner. Theta*: Any-angle path planning on grids. *Journal of Artificial Intelligence Research*, 39(1):533–579, 2010.

[14] C. Xie, J. Van Den Berg, S. Patil, and P. Abbeel. Toward asymptotically optimal motion planning for kinodynamic systems using a two-point boundary value problem solver. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[15] I. A. Sucan, M. Moll, and E. E. Kavraki. The open motion planning library. *IEEE Robotics and Automation Magazine*, 19(4):72–82, 2012.

[16] Jennifer C. Dela Cruz, Glenn V. Magwili, Juan Pocholo E. Mundo, Giann Paul B. Gregorio, Monique Lorraine L. Lamoca, and Jasmin A. Villasenor. Items-mapping and route optimization in a grocery store using dijkstra's, bellman-ford and floyd-warshall algorithms. In *2016 IEEE Region 10 Conference (TENCON)*, 2016.

[17] Th. Fraichard, A. Scheuer, R. Desvigne, and Rhne Alpes Gravir. From reeds and shepp's to continuous-curvature paths. *IEEE Transactions on Robotics*, 20(6):1025–1035, 2004.

[18] Oktay Arslan and Panagiotis Tsiotras. Use of relaxation methods in sampling-based algorithms for optimal motion planning. In *IEEE International Conference on Robotics and Automation*, 2013.