Smart Spline-Based Robot Navigation on Several Homotopies: Guaranteed Avoidance of Potential Function Local Minima

Roman Lavrenov
Intelligent Robotic Systems Laboratory, Higher Institute for Information Technology and Information Systems (ITIS), Kazan Federal University, 35 Kremlyovskaya street, Kazan, 420008, Russian Federation
E-mail: lavrenov@it.kfu.ru
http://kpfu.ru/robolab.html

Abstract

Potential function based methods provide powerful solutions in tasks of local and global path planning. They are characterized implementation simplicity, but suffer from navigation function local minima. In this paper we propose a modification of our original spline-based planning algorithm. Voronoi-based approach provides a good initial path as first iteration. A new safety criterion is integrated into path planning to guarantee path safety. The modified algorithm was implemented in Matlab environment and demonstrated significant advantages over the original algorithm.

Keywords: robotics, algorithm, modelling, mobile robot, path planning, Voronoi graph, potential field, optimization criteria, virtual experiments.

1. Introduction

Path planning, which is a fundamental feature of a mobile vehicle autonomous navigation, is concerned with automatic planning of a collision-free path between initial and target configurations. The classical path planning problem ("piano movers problem") is defined for complete a priori knowledge about environment. A robot is informed about its own shape, start and target position and orientation, and a set of obstacles in the environment. The robot searches for a continuous path from its initial position to the target position, while avoiding collisions with static obstacles along its way. The path should satisfy some particular criteria of its optimality.

One of the classical yet still popular approaches for dynamical solution of a collision avoidance problem is a potential field approach[1]. This technique is based on an artificial potential field generated at a global or a local level with an attractive pole in a role of a target and obstacles being represented with repulsive surfaces[2]. Then a robot follows the potential gradient toward its minimum[3]. The major advantages of potential field methods are simplicity and a capability of applying them reactively for mobile and stationary obstacles negotiation. Its typical and significant drawbacks include path oscillations (for certain obstacle configurations) and existence of local minima that attract the robot and then keep it captured inside.

Previously[4] we had proposed a path planning algorithm for a car-like mobile vehicle that could attain a target configuration from its initial configuration within a well-known environment providing a smooth obstacle free path. In order to improve the performance of the original algorithm, to add flexibility for path optimization and a possibility for further fast dynamic replanning, we integrate Voronoi Graph approach into our algorithm[5]. The new approach was tested in our previous work and shown the results, which significantly overperformed the original algorithm.

2. Spline-based robot navigation with original potential field approach

The spline-based method, proposed by Magid et.al. about a decade ago[4] navigates a car-like robot in a planar known environment populated with static obstacles. It considers an omnidirectional circle-shape robot that is reduced to a point in a 2D configuration space. Each obstacle is represented with a set of intersecting circles of different sizes. To provide a collision free path, a repulsive potential function has a high value inside an obstacle and on its boundary and a small value in free space. High potential function value in the obstacle's center pushes all points of a path outside in order to minimize path cost during local optimization procedure. The path cost function sums up three components:

$$F(q) = \gamma_1 T(q) + \gamma_2 R(q) + \gamma_3 L(q) \qquad (1)$$

Where T(q) function takes into an account all N obstacles of the environment and their influence (as potential field) on the robot along the entire path. The example of influence of obstacles to potential field is demonstrated in Fig.1. R(q) is a function for smoothness property of the path. And L(q) function accumulated path length.

The original algorithm of spline-based path planning works iteratively, beginning with start point S and target point T, and utilizes the environment obstacles as its input data. An initial path is suggested as a straight line between points S and T. Eq. (1) sets a current path cost, which is further optimized with Nelder-Mead Simplex Method[6] in order to minimize the path cost and provide a next spline with +1 increase in its flexibility degree and complexity as well. A resulting better path serves as an initial guess for the next iteration.

The original method succeeds to provide a collision free smooth path for any complexity of the environment only if each obstacle is approximated with a single circle. However, when the obstacles are approximated with a number of intersecting circles, the intersections introduce local maxima of the global potential field. If an initial spline passes through intersection of several circles that form a single obstacle, the cost function F(q) may succeed pushing the spline out of intersection area and

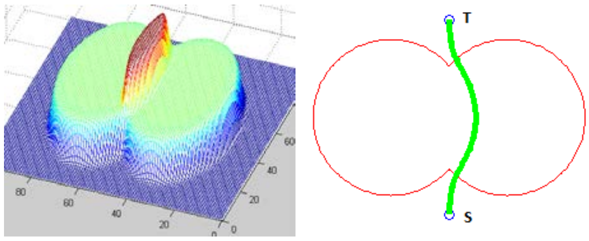

Fig. 1. A global potential field of the environment (left) and a final path between start (S) and target (T) positions that was suggested by the original algorithm.

then fail in escaping from a local minima at further iterations (Fig. 1, on the right).

3. Adding path safety parameter

We introduce an additional path quality evaluation criterion - average distance from obstacles. While navigating in obstacle populated environments, a robot should maximize its distance from the obstacles, and this feature is integrated into T(q) component of our cost function. However, in some operations, there may be a requirement to stay at some defined distance from obstacle boundaries. This criterion may be required for the range-limited sensors of the robot. A distance of the robot from all obstacles of environment is calculated in each configuration q(t) along the parametrically defined path as follows:

$$q(t) = \min_{\forall c \in C} \sqrt{(x(t) - x(c))^2 + (y(t) - y(c))^2} - r(c) \qquad (2)$$

Here C is a set of all circular obstacles c with the center at (x(c), y(c)) and radius r(c); further, these elementary circular obstacles may intersect to form compound obstacles. Average distance (α) is predefined. The effect of the average distance on the cost function Eq. (1) we can calculate with:

$$A(q) = \sum_{\forall t \in [0,1]} (\alpha - q(t))^2 \qquad (3)$$

All four criteria are combined into the cost function:

$$F(q) = \gamma_1 T(q) + \gamma_2 R(q) + \gamma_3 L(q) + \gamma_4 A(q) \qquad (4)$$

Figure 2 shows the result of path planning methods with and without the new criteria. In both cases, we searched a path from *S* to *T*. Blue points on the trajectory indicate control points for constructing a spline[10]. Thus, in the environment in Figure 2, the path was found in 4 iterations: top image shows a path without the new criteria, while a path with new criteria with parameters α=5, γ4=1 appears in the bottom. The algorithm with a new criterion builds a safer path. The red arrows in the figure indicate the places where the path became farther from the obstacles.

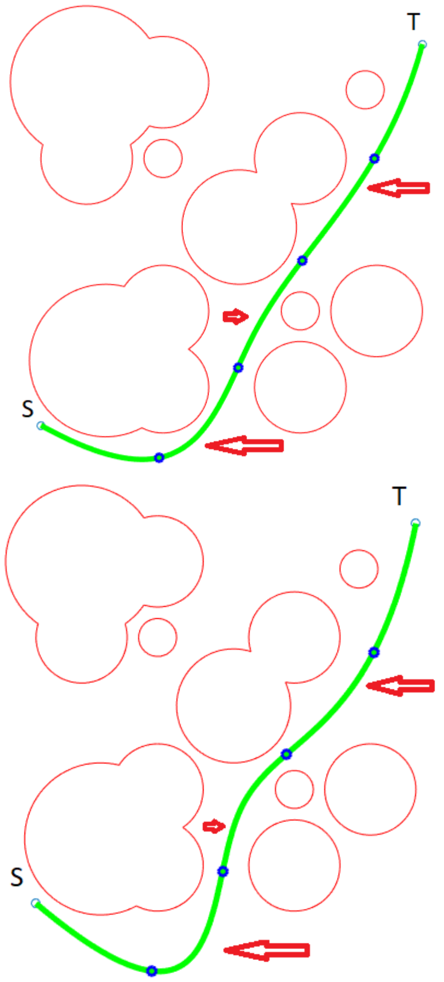

Fig. 2. Path calculated by the original algorithm without (top) and with the new criteria (bottom). The red arrows mark the points of interest where the new criteria influence is clearly visible.

Unfortunately, despite the use of a new criterion route optimization is performed only locally, the effect of the additional parameter is also local. The path remains in the same homotopy group even in the case when it is possible to bypass any obstacle from the other side.

4. Voronoi graph based solution

In order to provide a good initial spline that could be further improved locally with regard to user selection of the cost weights, we apply Voronoi graph approach[7]. With the three steps we prepare the environment by grouping circles into obstacles[10]: register circle obstacles to form a single compound obstacle, find outer and inner boundaries of each compound obstacle, and remove inner boundaries. For example, in Fig.3 (in the left image) the three inner obstacle boundaries will be removed.

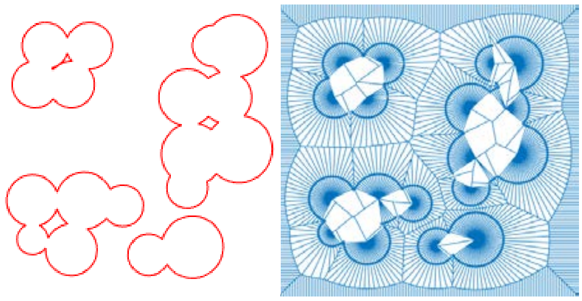

Fig. 3. Inner and outer boundaries of obstacles (left) and the process of Voronoi graph calculating (right).

Next, Voronoi graph is constructed based on a classical approach[11]. Upon obtaining Voronoi graph VG, we connect start position S and target position T points to VG and apply Dijkstra algorithm[8]. For example in Fig. 4 you can see created Voronoi graph as tiny red lines and calculated path as fil red line. Any path (S, T) on Voronoi graph VG is guaranteed to be collision free and maximally safe with regard to distance from obstacle boundaries, and thus could provide a good initial spline for the original spline-based method[4]. A small set of special points is utilized to form via points for the initial spline[10] (blue points in Fig. 4 and Fig.5).

5. Conclusions and future work

In this paper, we have presented a method for calculating a smooth and safe path for mobile robot in static planar environment. A significant modification of our original spline-based path planning algorithm for a car-like mobile vehicle navigation helps avoiding local minima problem and adds more flexibility for path optimization. We have integrated a Voronoi graph approach into the first iteration of spline optimization. Furthermore, Voronoi graph provides an opportunity to vary paths from different homotopy classes with regard to dynamic changes in optimization function criteria as well as supports a fast dynamic replanning in a case when an initially off-line selected path becomes unavailable. The algorithm was implemented in Matlab environment and its results were compared with our original algorithm. The new approach requires less optimization iterations that the original algorithm due to a smart selection of an initial spline. While the original algorithm fails to find an existing path in complicated environments with multiple concave obstacles, its smart version was successful in all simulated tests. For example, original approach cannot find path in environment shown in Fig.5. Voronoi-based method find the path by 3 iteration.

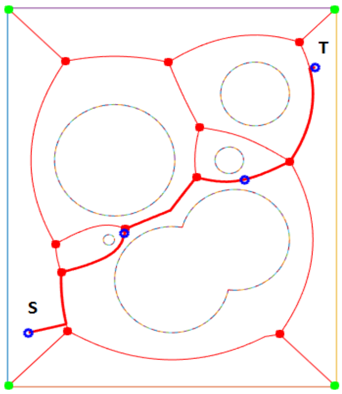

Fig. 4. Path within Voronoi graph.

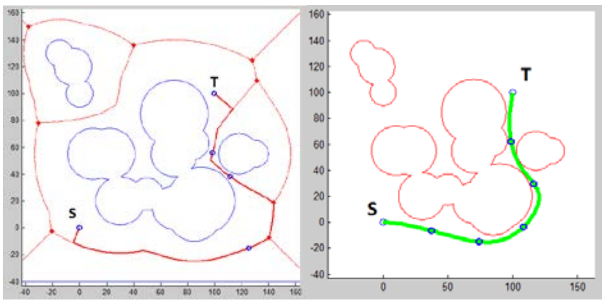

Fig. 5. Path on Voronoi graph (left) and resulting path after three iteration with the modified algorithm (right).

As our future work, we plan to integrate a number of additional parameters of 3D environments into a cost function. We also plan to create a C++ library of the method and further extend the proposed method for the navigation of a heterogeneous robotic team operating in an urban search and rescue scenario[9].

Acknowledgements

This work was partially supported by the Russian Foundation for Basic Research (RFBR) and Ministry of Science Technology & Space State of Israel (joint project ID 15-57-06010). Part of the work was performed according to the Russian Government Program of Competitive Growth of Kazan Federal University.

References

1. S. Fleury, P. Soueres, J.-P. Laumond, and R. Chatila, Primitives for smoothing mobile robot trajectories. *IEEE transactions on robotics and automation*, **11**(3) (1995) 441-448.
2. O. Khatib and B. Siciliano, *Springer handbook of robotics*. (Springer, 2016).
3. E. Magid, T. Tsubouchi, E. Koyanagi, and T. Yoshida, Building a search tree for a pilot system of a rescue search robot in a discretized random step environment. *J. of Robotics and Mechatronics*, **23**(1) (2011) 567-581.
4. E. Magid, D. Keren, E. Rivlin, and I. Yavneh, Spline-based robot navigation. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (2006), pp. 2296-2301.
5. E. Magid, R. Lavrenov and I. Afanasyev, Voronoi-based trajectory optimization for UGV path planning. *Int. Conf. on Mechanical, System and Control Engineering* (2017), pp. 383-387.
6. J. C. Lagarias, J. A. Reeds, M. H. Wright and P. E. Wright, Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM J. on optimization*, **9**(1) (1998) 112-147.
7. R. Lavrenov, F.Matsuno and E. Magid, Modified spline-based navigation: guaranteed safety for obstacle avoidance. *In: Ronzhin A., Rigoll G., Meshcheryakov R. (eds) Interactive Collaborative Robotics. Lecture Notes in Computer Science*, 10459 (Springer, Cham, 2017) 123-133.
8. H. M. Choset, *Principles of robot motion: theory, algorithms, and implementation* (MIT press, 2005).
9. E. Magid, R. Lavrenov, M. Svinin and A. Khasianov, Combining Voronoi graph and spline-based approaches for a mobile robot path planning. *In Lecture Notes in Electrical Engineering* (Springer International Publishing, 2018), in press.
10. E. W. Dijkstra, A note on two problems in connexion with graphs. *Numerische mathematik* **1**(1) 269-271.
11. A. Andreychuk and K. Yakovlev, Applying MAPP Algorithm for Cooperative Path Finding in Urban Environments. *In: Ronzhin A., Rigoll G., Meshcheryakov R. (eds) Interactive Collaborative Robotics. Lecture Notes in Computer Science*, 10459 (Springer, Cham, 2017) 1-10.