Article

Artificial Potential Field with Discrete Map Transformation for Feasible Indoor Path Planning

Muhammad Zulfaqar Azmi ^{1,*} and Toshio Ito ²¹ Graduate School of Engineering and Science, Shibaura Institute of Technology, Saitama 337-8570, Japan² Department of Machinery and Control Systems, Shibaura Institute of Technology, Saitama 337-8570, Japan; tosi-ito@shibaura-it.ac.jp

* Correspondence: nb18504@shibaura-it.ac.jp

Received: 25 November 2020; Accepted: 11 December 2020; Published: 16 December 2020

Abstract: This work considers the path planning problem of personal mobility vehicle (PMV) for indoor navigation using the Artificial Potential Field (APF) method. The APF method sometimes suffers from an infinite loop problem during the planning phase when the goal is blocked by obstacles with certain characteristics. To address the issue, this study deploys the map augmentation method for replanning. When infinite loop situations occur, the map is transformed and the search for drivable path is initiated. The method successfully generates a feasible trajectory when the map is rotated at a certain angle. The scenario of successful planning is shown in the result.

Keywords: map transformation; artificial potential field; lidar data

1. Introduction

Automated Driving Systems (ADSs) have the potential to reduce the number of accidents, lower emissions, transport the physically-impaired, and reduce the driver's workload [1]. Based on the Society of Automotive Engineers (SAE), ADS describes the variety of commonly used terms in intelligent transportation field, e.g.: autonomous, self-driving, driverless, etc. [2].

Most ADSs break down several tasks of automated driving and, by utilizing an array of sensors and algorithms, they can be solved accordingly. Path planning is one of the essential requirements for all ADSs. The path planning stage of a vehicle requires it to obey traffic rules within road networks while a robot has no specific requirements to follow [3].

The scenario of path planning involves planning in known and unknown scenarios. Further enhancement involves planning for static and dynamic obstacles. Various perception sensing devices, for example, camera or Light Detection and Ranging (LiDAR) sensors have been used to scan the surroundings. Then, the data are processed by Simultaneous Localization and Mapping (SLAM) algorithm to divide the data into free space and fixed obstacles [4]. The result is known as a static map. The static map provides a piece of very essential information for the path planning algorithm as it needs to generate a traversable path of the vehicle based on its position with respect to the space within the static map. One of the drawbacks, however, as the area gets larger, is that the computational requirement tends to get higher.

The transformation of one type of space to another type is one of the techniques to reduce computational complexity. The transformation method is typically used by machine learning algorithms. For example, in Support Vector Machine, the field is transformed using the kernel trick, which increases the dimension of the field [5]. In data science, the data are augmented to increase data variation [6]. It is found that, when augmented data are used to test the non-augmented model, the classification error is more prominent despite the same batch of images being used.

There are many techniques that have been adopted to solve the path planning problem. The blind search technique, i.e., Breadth First Search (BFS) [7] and Depth First Search (DFS) [8], traverses every single state available until the feasible solution is found. They are typically used to solve maze problems. BFS and DFS are simple to be implemented, but they require a lot of computational time if the given area is huge. Alternatively, there are other more efficient methods and they can be divided into four basic algorithms: graph-search based planners, sampling-based planners, interpolating curve planners, and numerical optimization approaches [3]. Graph-based planners, for example Dijkstra Algorithm and A*, utilized the graph searching algorithm to search for a solution by traversing between grid or states. The sampling-based planners sample the configuration space and return solution when connectivity between them exists. The commonly used sampling-based planner's algorithm includes the Probabilistic Roadmap Map (PRM) and Rapidly Exploring Random Tree (RRT) [9]. The interpolating curve planners add a new set of waypoints given a range of reference points. This technique typically increases the path continuity, which increases comfort and allows smoother motion. Polynomial, Bezier, and Splines are some of the well-known interpolation algorithms. Finally, numerical optimization approaches use an algorithm like Particle Swarm Optimization (PSO) [10] or Genetic Algorithm (GA) to optimize a given objective function subjected to any constrained variable, for example, vehicle kinematic constraints or any environmental constraints. Each of these algorithms can be used independently and can be combined based on the application.

Among the path planning algorithms, the graph search-based planner often deals with an occupancy grid or any approximation of a discrete cell-grid space. For example, the Dijkstra Algorithm locates the single-source shortest path. However, the algorithm also suffers from performance issues. Its extension, the A* algorithm, solves this problem by taking the heuristic function into consideration [11]. This enables fast node search. However, as the method searches for the shortest path, the generated trajectory tends to lead the vehicle to be too close to the obstacle. In the case of vehicles operating indoors, driving too close to any objects may be hazardous, as accidents may occur due to the possibility of humans exiting doors. Thus, an obstacle-distancing method for the vehicle is needed.

A common method to penalize proximity between vehicles and any known landmark in the map is by using a potential field [12]. One of them is the Artificial Potential Field (APF) method. Some studies employ APF to create the potential of road boundaries which allows the vehicle to maintain within its lane on the road, simultaneously allowing a safe distance for any sudden appearing object [13,14]. It also has been used to define human personal space [15]. However, the method may also yield trajectory near wall proximity unless further parameter tuning is needed. At the same time, it is computationally expensive if the number of points is large. Nevertheless, it can be optimized by discretizing the field [16].

This paper introduces an alternative APF path planning consideration for an environment known a priori. The main objective of the research carried out is to develop an effective fast-planning method for indoor vehicle navigation. The proposed solution was verified by simulated studies. The main contribution of this paper is the consideration of a map transformation method when the path is not found during the primary planning stage.

2. Problem Definition

The path planning problem involves solving the task of finding an optimal collision-free path between a starting point and the target point. It also considers the characteristics of the obstacle, for example, the size and shape of the object, or landmarks within a map.

Figure 1 shows the top view illustration of an $\mathbb{R}^3$ space. $\mathbb{R}^3$ contains two objects, C_1^{obs} and C_2^{obs} that occupy the free space, each assumed as a landmark. A collision-free path will allow the vehicle to traverse past these landmarks along the traversable trajectory from point S_0 to S_g .

Figure 1. Landmarks within the map before being discretized.

To address the aforementioned problem, continuous space is discretized. However, this will result in the shape of the object being constrained by the pixel that it occupies. For example, in Figure 2, the cell being marked as $\times$ has blocked the direct path between the two points.

Figure 2. Discretized map with the original orientation.

By transforming the map, and discretizing it again, the occupied cell will yield a different shape compared to the original orientation. Figure 3 illustrates a map rotated at an angle of 90° clockwise. Note that object C_2^{obs} occupies a 2-by-2 cell, similar to its non-rotated counterpart in Figure 2, while the object C_1^{obs} has a different shape. The path between the point S_0 to S_g is also not blocked by any cells.

Figure 3. Discretized map with the transformation applied.

In a discretized space, the generated path may not be smooth. However, it is common for the path smoothing to be performed after a path is found [17].

During APF path planning, when the obstacles and the goal are in close proximity, the computation of the potential map might result in one global minimum (where the goal is located) and multiple local minima. Figure 4 illustrates the difference between local and global minimum.

Figure 4. Illustration of local minimum and global minimum.

In this case, it is possible for the planner to become stuck between the local minima. Consider the following configuration in Figure 5 where this issue is depicted.

Figure 5. The planner suffers from infinite-loop.

Since the goal is close to both landmarks, the planner will fall into a local minimum as it heads towards them. As the planner is aware that it has yet to arrive at the true goal, it will keep on repeating the same previous subroutine. However, as it is unable to escape the area of the local minimum, the computation will keep repeating infinitely, therefore causing infinite loop problem.

The modification of grid's shape by rotation such as shown in Figure 6, allows the possibility for the planner to escape the local minimum and break from the infinite loop.

Figure 6. Augmentation of Figure 5 rotated at 71° clockwise.

3. Proposed Method

3.1. Artificial Potential Field

The artificial potential field (APF) method is inspired by the electrical charges' concept [18], whereby the objects in the configuration space, where the vehicle is traveling, are presumed to emit potential charges. The goal or target position is assumed to generate an attractive force that pulls the ego vehicle towards it. On the other hand, the obstacle creates a repulsive force that pushes the vehicle away. These forces are calculated in reference to the vehicle's position. These forces will define the behavior of the vehicle as it moves towards the goal, while avoiding any obstacle. The repulsive force will also prevent the vehicle from 'hugging the wall', which is the behavior that some other path planning method suffers [12]. Figure 7 shows the illustration of both force field.

Figure 7. Attractive and Repulsive force field.

The total potential is defined as the sum of two potentials, i.e., attractive and repulsive, given as

$$U(q) = U_{att}(q) + U_{rep}(q) \quad (1)$$

where $U_{att}(q)$ is the attractive potential generated at the goal position, with respect to each cell in the configuration space, as follows:

$$U_{att}(q) = \frac{1}{2} \delta \cdot d(q, q_g) \quad (2)$$

where $\delta > 0$ is a positive constant that determines the strength of the attractive field, also known as gravitational constant, and $d(q, q_g) = \|q - q_g\|$ is the Euclidean distance between each cell $q := (x, y)$ in the configuration space and the goal position, $q_g := (x_g, y_g)$.

Repulsive potential affects every cell in the configuration space as follows:

$$U_{rep}(q) = \begin{cases} \frac{1}{2} \alpha \cdot \frac{1}{d(q, q_{obs})} - \frac{1}{Q^*} & \text{if } d(q, q_{obs}) \leq Q^* \\ 0 & \text{if } d(q, q_{obs}) > Q^* \end{cases} \quad (3)$$

where $\alpha > 0$ is a positive weighting constant of the repulsive field. Meanwhile, $d(q, q_{obs}) = \min_{Q' \in Q_{obs}} \|Q - Q'\|$ represents the smallest distance between the vehicle's current position, $Q = (x_{start}, y_{start})$, and the obstacle position, $q_{obs} := Q_{obs} = (x_{obs}, y_{obs})$. The repulsive potential is defined as zero outside the distance of the influence of the obstacle (Q^*) and positive when it is inside.

3.2. Transformation of Configuration Space

When the planner is trapped within an endless loop, the configuration space will be transformed. The transformation involves rotating the original point cloud map, discretizing it, and maps updating

the goal and vehicle's initial location to the newly rotated map's coordinate frame. For a given set of position (x, y) in the normal space, the rotated position is calculated using

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4)$$

where R denotes 2×2 rotation matrices and t is the 2×1 translation matrices, both are captured when the point cloud map is rotated. This space, therefore, is referred to as $\mathbb{R}^L$ from here on.

The value of (x, y) used in Equations (1)–(3) are substituted with transformed position values, $(x^L, y^L) \in \mathbb{R}^L$, accordingly.

3.3. Path Planning

After the total potential map is obtained, the path searching will start. Figure 8 shows the overall algorithm to obtain the feasible path, S . However, the generated path will typically oscillate, especially during the planning of a straight corridor in an indoor-constrained space [19]. Consequently, a low pass filter is used to smoothen the path.

Figure 8. Flowchart algorithm 1: Obtaining feasible trajectory.

3.4. Savitzky–Golay Filter

In this study, the Savitzky–Golay filter is employed to reduce path oscillation. A Savitzky–Golay filter is a data smoothing algorithm based on local least-squares polynomial approximation [20]. The filter is able to preserve the original structure of the signal while removing noise. There are three inputs required for the implementation: noisy signal, the order of polynomial, and frame-size. The best fit of these values is estimated using the trial and error method [21].

For example, a set of waypoints trajectory $S = \{s_i\} = \{(x_i, y_i)\}, i = 1, 2, \dots, n$, is expressed as a set of positive odd integer, m frames with coefficients, C_i , given as follows:

$$sf_i = \sum_{i = -\frac{(m-1)}{2}}^{\frac{m-1}{2}} C_i s_i \quad (5)$$

where s_i and sf_i represent unfiltered and filtered data, respectively. To the author's knowledge, there is no clear definition of the best value suitable to be used, thus, as previously mentioned, the final result will be based on trial and error estimation.

4. Experimental Design

4.1. Hardware Setup

A 4-wheel personal mobility vehicle (PMV), as shown in Figure 9 is used to collect data. A custom platform is attached to the front of the PMV where the Velodyne VLP-16 LiDAR (by Velodyne Lidar Inc., San Jose, CA, USA) is placed on top of it. DELL G7 laptop with NVIDIA GTX1060Ti (by Dell Technologies Inc., Round Rock, TX, USA), Ubuntu 18.04 operating system, and Robot Operating System (ROS) Melodic are used for collecting and processing the LiDAR point cloud data.

Figure 9. Hardware set up.

4.2. Data Collection Method

All data are obtained within the Shibaura Institute of Technology, Omiya Campus' indoor area. During raw point cloud data collection, the PMV is operated at a constant speed to reduce the speed error that may cause drift when generating point cloud map. The point cloud data are processed offline and the Normal Distribution Transform (NDT) mapping method is used to generate a 3D functional map, with a total of 6783 points. 3D data are collected because the height information is important for human rideable PMV. To reduce the number of obstacle points to be computed, the map is discretized to a 2D grid map, and the height information is embedded to the pixel color value. The dimension for each pixel is $0.25 \text{ m} \times 0.25 \text{ m}$.

4.3. Path Planning

Once the 2D grid map is obtained, the risk potential is calculated. The starting and the goal position are chosen arbitrarily and, for this study, the goal is located at the end of the floor's corridor. The trajectory is calculated after the risk map is available and, if the planner fails to do so, the map is transformed until the feasible path is found. The path is then transformed back to the original orientation.

5. Results and Discussion

The results of the point cloud mapping before and after being discretized to 2D grid cells are shown in Figures 10 and 11, respectively, with the ground and ceiling's information being filtered out.

Figure 10. 3D Point Cloud Map.

Figure 11. Point cloud map centered to 2D grid cells.

During the process, the pixel size information and the minimum and maximum value of x and y of the 3D point cloud map is recorded. This will be used during the conversion from the transformed orientation back into the original orientation.

Figure 12 illustrates the colorbar that is used as the indication for the potential value at each cell. The blue region shows the area with low potential value and the red region signifies as the high-risk area.

Figure 12. Potential value's colorbar for indication.

Figure 13 shows the original map without any discretization and transformation applied. For this scenario, the starting position is not blocked by any walls or obstacle, thus the planning is straightforward.

Figure 13. Non-discretized potential map.

Figure 14 shows the altered potential map of Figure 13, discretized to $0.25 \text{ m} \times 0.25 \text{ m}$. The attractive force creates a low potential while the obstacles are encapsulated by high potential charges. The start position is located at (228, 143), and the goal location is at the end of the corridor at (102, 44). Similarly, the planning is also straightforward in this case.

Figure 14. A feasible path is found from (228, 143) to (102, 44).

The path of before and after smoothing is shown in Figure 15. It can be observed that, before smoothing, the path exhibits a zig-zag pattern.

Figure 15. Path before and after filtering for Figure 14.

Now, the starting position is shifted to position (36.3, 23.2) in the original map. In this case, the planner fails to find a path, as shown in Figure 16. This is due to the planner suffering from an

infinite loop between two walls. A point to note is that the planner is heading straight to the wall. This is the nature of the APF algorithm; as it searches for the feasible path, it will stay as close as possible to the boundaries of the potentials near the wall. The reason is due the algorithm attempt to minimize the distance between the vehicle and the obstacle.

Figure 16. Non-discretized potential map—failed scenario.

When discretized to $0.25 \text{ m} \times 0.25 \text{ m}$ grid, as shown in Figure 17, the starting position is now at (242, 135); however, the planner still fails to find a path. This is because the planner also suffers from an infinite loop, as it is stuck between the wall and the pillar at the corner of the area. As the 2D map is a downscaled version of the 3D point cloud map, the planner has added a tolerance value of plus-minus one-pixel value to the starting position—for example, (242, 136), (243, 135) and so on. However, even with the tolerance, the planner still fails, as shown in Figure 17.

Figure 17. Failed scenario at (242, 135), before the map is rotated.

Figure 18 shows the transformed Figure 17. The transformation matrix is obtained by the brute force method, which is rotating the map by 1-degree increment until the path is found. The matrix T , based on Equation (4), is used to recalculate the transformed starting and goal position. The value depends on the increment of rotation. However, the result did not yield a round value. Thus, the number will first be rounded down. For the first trial, the starting value is at (196, 133) and the goal is at (40, 158). As shown in the same figure, the planner is still stuck in the similar area to that of the previous scenario before the map is rotated.

The next trial involves rounding up the transformed value. The x -coordinate starting value is rounded up to 197. Thus, the next starting position is at (197, 133). The goal remains unchanged. The result is illustrated in Figure 19.

It can be observed that a feasible path is found. The reason is that the 2D grid cell map is the scaled-down integer representation of the original 3D point cloud map. This also indicates that the size of the grid cell may affect the performance of the APF. Furthermore, as the consideration of the planner is based on the size of the discretized grid, the resulted path will ensure an added distance between the vehicle and any obstacle. The obtained path before and after smoothing is displayed in Figure 20. Inverse transformation is applied to return the path to its original orientation.

Figure 18. Transformed map—however, search fails.

Figure 19. Transformed map, the path found.

Figure 20. Transformed map before and after smoothing.

6. Conclusions

This study proposed a transformed-map approach to the global path planning method. As the point cloud map is computationally demanding, it is proposed to downscale the map to 2D grid cells where the pixel color corresponds to the height. APF is employed to search for the feasible path by computing the potential value at each cell. However, the method may suffer from the infinite loop when the planner falls between obstacles with certain characteristics. To solve this problem, the map is transformed by rotating the original map to a certain angle. The APF value is recalculated for the transformed map. It is found that varying the pixel value can affect the performance of the path planner.

Author Contributions: Conceptualization, M.Z.A.; methodology, M.Z.A.; investigation, M.Z.A.; software, M.Z.A.; validation, M.Z.A.; data curation, M.Z.A.; formal analysis, M.Z.A.; resources, T.I.; supervision, T.I.; writing—original draft preparation, M.Z.A.; writing—review and editing, M.Z.A. and T.I.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yurtsever, E.; Lambert, J.; Carballo, A.; Takeda, K. A Survey of Autonomous Driving: Common Practices and Emerging Technologies. *IEEE Access* **2020**, *8*, 58443–58469. [\[CrossRef\]](#)
2. Chan, C-Y. Advancements, prospects, and impacts of automated driving systems. *Int. J. Transp. Sci. Technol.* **2017**, *3*, 208–216. [\[CrossRef\]](#)
3. González, D.; Pérez, J.; Milanés, V.; Nashashibi, F. A Review of Motion Planning Techniques for Automated Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2016**, *4*, 1135–1145. [\[CrossRef\]](#)
4. Sun, N.; Yang, E.; Corney, J.; Chen, Y. Semantic Path Planning for Indoor Navigation and Household Tasks. TAROS 2019: Towards Autonomous Robotic Systems. In *Lecture Notes in Computer Science*; Althoefer K., Konstantinova, J., Zhang, K., Eds.; Springer: London, UK, 2019; Volume 11650, pp. 191–201.
5. Amari, S.; Wu, S. Improving support vector machine classifiers by modifying kernel functions. *Neural Netw.* **1999**, *12*, 783–789. [\[CrossRef\]](#)
6. Perez, L.; Wang, J. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *arXiv* **2017**, arXiv:1712.04621.
7. Jose, S.; Antony, A. Mobile robot remote path planning and motion control in a maze environment. In Proceedings of the IEEE International Conference on Engineering and Technology, Coimbatore, India, 17–18 March 2016; pp. 207–209.
8. Shwail, S.H.; Karim, A.; Turner, S. Probabilistic multi robot path planning in dynamic environments: A comparison between A* and DFS. *Int. J. Comput. Appl.* **2013**, *7*, 29–34.
9. Kim, Y.N.; Ko, D.W.; Suh, I.H. Confidence random tree-based algorithm for mobile robot path planning considering the path length and safety. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1–10. [\[CrossRef\]](#)
10. Chen, X.; Li, Y. Smooth Path Planning of a Mobile Robot Using Stochastic Particle Swarm Optimization. In Proceedings of the International Conference on Mechatronics and Automation, Luoyang, China, 25–28 June 2006; pp. 1722–1727.
11. Duchoň, F.; Babinec, A.; Kajan, M.; Beňo, P.; Florek, M.; Fico, T.; Jurišica, L. Path Planning with Modified a Star Algorithm for a Mobile Robot. *Procedia Eng.* **2014**, *96*, 59–69.
12. Dolgov, D.; Thrun, S.; Montemerlo, M.; Diebel, J. Path Planning for Autonomous Vehicles in Unknown Semi-structured Environments. *Int. J. Rob. Res.* **2010**, *29*, 485–501. [\[CrossRef\]](#)
13. Raksincharoensak, P.; Hasegawa, T.; Nagai, M. Motion Planning and Control of Autonomous Driving Intelligence System Based on Risk Potential Optimization Framework. *Int. J. Automot. Eng.* **2016**, *7*, 53–60. [\[CrossRef\]](#)
14. Hamid, U.Z.A.; Zamzuri, H.; Rahman, M.A.A.; Saito, Y.; Raksincharoensak, P. Collision avoidance system using artificial potential field and nonlinear model predictive control: A case study of intersection collisions with sudden appearing moving vehicles. In Proceedings of the International Symposium on Dynamics of Vehicles on Roads and Tracks, Rockhampton, Australia, 14–18 August 2017.

15. Toyoshima, A.; Nishino, N.; Chugo, D.; Muramatsu, S.; Yokota, S.; Hashimoto, H. Autonomous Mobile Robot Navigation: Consideration of the Pedestrian's Dynamic Personal Space. In Proceedings of the IEEE International Symposium on Industrial Electronics, Cairns, Australia, 13–15 June 2018; pp. 1094–1099.
16. Lazarowska, A. Discrete Artificial Potential Field Approach to Mobile Robot Path Planning. *IFAC-Pap. OnLine* **2019**, *52*, 277–282. [[CrossRef](#)]
17. Klančar, G.; Seder, M.; Blažič, S.; Škrjanc, I.; Petrović, I. Drivable Path Planning Using Hybrid Search Algorithm Based on E* and Bernstein-Bézier Motion Primitives. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, 1–15. [[CrossRef](#)]
18. Kumar, P.B.; Rawat, H.; Parhi, D.R. Path planning of humanoids based on artificial potential field method in unknown environments. *Expert Syst.* **2019**, *36*, 7655–7678. [[CrossRef](#)]
19. Zhu, Q.; Yan, Y.; Xing, Z. Robot Path Planning Based on Artificial Potential Field Approach with Simulated Annealing. In Proceedings of the International Conference on Intelligent Systems Design and Applications, Jinan, China, 16–18 October 2006; pp. 622–627.
20. Schafer, R.W. What Is a Savitzky–Golay Filter? [Lecture Notes]. *IEEE Signal Process. Mag.* **2011**, *28*, 111–117. [[CrossRef](#)]
21. Acharya, D.; Rani, A.; Agarwal, S.; Singh, V. Application of adaptive Savitzky–Golay filter for EEG signal processing. *Perspect. Sci.* **2016**, *8*, 677–679. [[CrossRef](#)]

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).