Information Engineering Express
International Institute of Applied Informatics
2020, Vol. 6, No. 1, 59 – 69

Transient Virtual Obstacles for Safe Robot Navigation in Indoor Environments

Abhijeet Ravankar [*], Ankit A. Ravankar [†],
Yohei Hoshino [*], Michiko Watanabe [*], Arpit Rawankar [‡]

Abstract

Service robots are expected to work in real world scenarios which are dynamic in nature. The traversable and non-traversable passages of the environment might change at different times. For example, it is common for some of the passages or areas in the map to be inaccessible due to cleaning, repair works, or other reasons. On the other hand, some passages in the environment could have a high congestion of people compared to other passages. This requires a feature to alter the path of the robots to address these dynamic changes, instead of only relying on the shortest paths criterion. To this end, this paper proposes Transient Virtual Obstacles (TVO): a method to actively block desired paths by using virtual obstacles. The virtual obstacles can be placed or removed by a user anywhere in the map for programmable intervals. The robot's plan their paths considering the virtual obstacles, thereby blocking certain paths, and planning the desired trajectories through specific paths. Another major advantage of the proposed TVO algorithm is that the existing robot path planner does not needs to be modified. The proposed TVO is tested in both simulation and real environments and results are discussed highlighting the merits of safe mobile robot navigation.

Keywords: Path Planning, Robot Navigation, Virtual Obstacles.

1 Introduction

Service mobile robots are expected to work in human-centric environments to replace the dull (ex. cleaning), dangerous (ex. handling toxic materials), and demanding (ex. lifting heavy items) tasks. Navigation is a critical component of such service mobile robots. Unlike well-controlled research environments and simple areas, real-world scenarios are dynamic in nature. For example, passages of the environment are routinely blocked for cleaning. During this time, mobile robots cannot traverse the passage and must re-plan an alternate route to the goal. The passages may also be blocked for other purposes. Moreover, it is

[*] Kitami Institute of Technology, Kitami, Japan
[†] Hokkaido University, Sapporo, Japan
[‡] Vidyalankar Institute of Technology, Mumbai, India

common for some of the passages to have high congestion at specific times. For example, passages around cafeteria in universities are generally congested during lunch hours. Mobile robots are equipped with path planners which generally plan trajectories based on the shortest path criteria. Unlike people, robots do not have access to the information about the blocked paths or congested areas. If the shortest path planned by the robot happens to pass through the blocked passages of the map, the robot must re-plan its path towards the goal. On the other hand, if the path planned by the robot passes through the congested areas of the map, a robot will have to avoid many dynamic obstacles for which it be stopping several times adversely affecting the quality of service (QoS) by taking longer time in navigation. Considering the weight and speed of the robot, mobile robot navigation through congested passages could be dangerous as there is a risk of dashing into people or accidents due to sudden stops to avoid collision. Similarly, forceful navigation through blocked passages is unsafe.

To this end, this paper proposes Transient Virtual Obstacles (TVO), a method to actively block desired paths by using virtual obstacles. The virtual obstacles can be placed or removed by a user anywhere in the map for desired intervals. For example, users can block certain paths around the cafeteria only during the lunch hours forcing the robot to plan paths through less congested passages. The robot's plan their paths considering the virtual obstacles, thereby blocking certain paths, and planning the desired trajectories through specific paths. Another major advantage of the proposed TVO algorithm is that the existing robot path planner does not need to be modified. Since the dynamic scenarios are easily comprehended by humans, the proposed method actively blocks certain areas of the map through virtual obstacles so that the global path planner of the mobile robot automatically generates a safer path. A virtual obstacle is not a real obstacle in the map. Moreover, it is transient and does not alter the actual map used by the robot.

2 State of the Art

Mobile robot path planning is a well studied problem. An extensive review of various path planning techniques can be found in [1]. Path planning is generally done in two stages. First, is the 'global planning' stage, in which, a path is searched from the start to the goal location considering only the static obstacles in the map. Once the robot starts navigation on the global path, the second stage of 'local planning' comes into picture, in which, kinematically feasible and smooth [2] trajectories for local collision with dynamic entities like moving people and other robots are generated in real-time. At the global stage, criteria like shortest paths is generally emphasized. At the local planning stage, the emphasis is mainly on collision avoidance[3, 4] cases. It is assumed that robots have a Simultaneous Localization and Mapping (SLAM) module to build a map and localize themselves in it [5, 6, 7]. The case of multi-robot navigation has been extensively discussed in [8, 9, 10, 11, 12]. Some researchers have focussed on realizing a robot that acquires logical recognition of space [13]. A number of path planning algorithms have been proposed, and a detailed summary of those algorithms can be found in [14], [15], and [16]. Among these, $A*$ algorithm [17], D* algorithm [18], probabilistic roadmap planner (PRM) [19], rapidly exploring random tree (RRT) [20], and potential fields [21] algorithms are the most widely used techniques today.

Many practical works which use virtual obstacles have been proposed in the state-of-the-art. Vision based collision avoidance by plotting a virtual obstacle on depth map has

Algorithm 1: Grid-map Generation

```
1  Function grid_mapping($\{l_{t-1,i}\}, x_t, z_t$)
2      if $\mathrm{m_i}$ in range $\mathrm{z_t}$ then
3          $P \leftarrow 0$
4          $r \leftarrow \sqrt{(x_i - x)^2 + (y_i - y)^2}$
5          $\phi \leftarrow \tan^{-1}\left(\frac{y_i - y}{x_i - x}\right) - \theta$
6          $k \leftarrow \mathrm{argmin_j}|\phi - \theta_{j,sens}|$
7          if $r > \min(z_{\max}, z_t^k + \frac{\eta}{2})$ or $|\phi - \theta_{k,\mathrm{sens}}| > \frac{\Gamma}{2}$ then
8              $P \leftarrow l_0$
9          if $z_t^k < z_{\max}$ and $|r - z_t^k| < \frac{\eta}{2}$ then
10             $P \leftarrow l_{\mathrm{occupied}}$
11         if $r \leq z_t^k$ then
12             $P \leftarrow l_{\mathrm{free}}$
13         $l_{t,i} \leftarrow l_{t-1,i} + P - l_0$
14     else
15         $l_{t,i} \leftarrow l_{t-1,i}$
16     return($l_{t,i}$)
```

been proposed in [22]. In [23], authors have proposed to integrate virtual obstacle concept with a potential-field-based method to maneuver cylindrical mobile robots in unknown or unstructured environments. In [24], a safety aware robot coverage motion planning with virtual-obstacle-based navigation has been proposed. In [25], authors have tacked the path-switching problem using virtual obstacles.

3 Virtual Obstacles in Map

This section explains the addition and removal of virtual obstacles in the map. Since this process is based on changing the values of certain areas of the map to values representing the obstacle values, we first discuss the grid-map generation which facilitates this task.

3.1 Grid-map Generation

A grid-map quantizes the world into equally sized partitions called cells (m_i). A cell can be free $p(m_i) = 0$, occupied $p(m_i) = 1$, or unknown $p(m_i) = 0.5$. A static map is assumed in which all the cells are independent. The probability distribution of the map is given by the product over the cells, $p(m) = \prod_i p(m_i)$. If sensor data $z_{1:t}$ and the poses $x_{1:t}$ of the sensor are given, the aim is to estimate the map, $p(m|z_{1:t}, x_{1:t}) = \prod_i p(m_i|z_{1:t}, x_{1:t})$.

The pseudo-code of grid mapping is shown in Algorithm 1. An inverse measurement model (Algo.1, Line 7) is used for robots equipped with range finders. Assuming the robot's pose to be $(x, y, \theta)^{\mathrm{T}}$, lines 7 to 16 calculates the inverse model [26, 27]. The inverse model first determines the beam index k and the range r for the center-of-mass of the cell m_i. In programming, log-odds ratio $l_{t,i}$ and l_0 are used and defined as, $l_{t,i} = \log\frac{p(m|z_{1:t}, x_{1:t})}{1 - p(m|z_{1:t}, x_{1:t})}$, and $l_0 = \log\frac{p(m_i=0)}{p(m_i=1)} = \log\frac{p(m_i)}{1 - p(m_i)}$.

In Algorithm 1, the thickness of obstacles, and the width of a sensor beam are represented as η and Γ, repsectively. Many algorithms for grid mapping have been proposed [26, 27] and other cost-map based mapping algorithms can also be used [26, 27].

Algorithm 2: Adding a Virtual Obstacle in the Map

```
Function virtual_obs_add($M, (x_1, y_1), (x_2, y_2), G_M, T$)
    $T \leftarrow$ current timestamp
    $i = y_1,\ j = x_1$
    while $i < y_2$ do
        while $j < x_2$ do
            $M[i][j] \leftarrow 1$
            $G_M[i][j] \leftarrow 1$
            $j \leftarrow j + 1$
        $i \leftarrow i + 1$
```

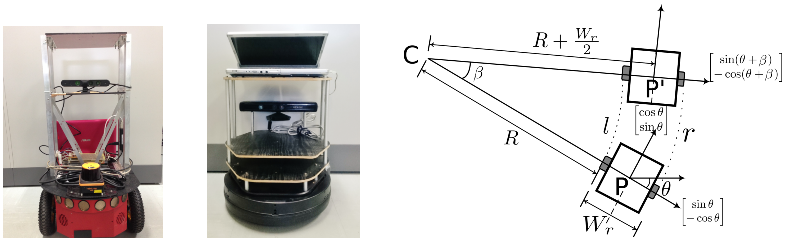
3.2 Addition and Removal of Virtual Obstacles

As explained in the previous section, a grid-map represents obstacles with a value 1, and free space as 0. The entire map is realized as a matrix M of size $W \times H$, where W and H are the width and height of the map, respectively. It is assumed that the top-most coordinate of the map is (0,0) with height increasing in the bottom direction, and width increasing in the right direction. The addition of a virtual obstacle between (top-left) coordinates (x_1, y_1) and bottom-right coordinates (x_2, y_2) to the map is accomplished by setting the appropriate values of the matrix with obstacle value of 1. This is shown in Algorithm 2. The input to the function are the current map M, the top-left and bottom-right coordinates where the virtual obstacle has to be placed, a tracker map G_M, and a timestamp T. The tracker matrix G_M keeps a track of all the addition and removal of virtual obstacles in the map, and has the same dimension as the matrix M. The timestamp T keeps a track of when the virtual obstacle was added. The procedure of removing the obstacle is similar by setting appropriate values of the matrix to value of free-space i.e. 0.

3.3 Affect of Virtual Obstacles on Global Path

The addition of virtual obstacles in the map affects the global path generation. The virtual obstacles can be added manually over certain areas of the map with high influx of people, or closed passages. Detection of people congestion in passages can be done by using external cameras or other sensors. Any global planner can be employed and it will automatically exclude path generation through the virtual obstacles. Our implementation uses A* [17] which is a famous algorithm for path planning. Let G $= (V, E)$ is a graph with non-negative edge distances, and h is an admissible heuristic. Let S be the start location and G be the goal node of the . If d(v) is the shortest distance from S to v seen so far, then d(v) $+$ h(v) gives an estimate of the distance from S to v, and similarly from v to G. The queue of nodes Q $= (V_{d1}, V_{d2}, \cdots, V_{dn})$ sorted by d(v) $+$ h(v) is the A* path from S to G.

To facilitate virtual obstacle addition in certain passages, the timestamp T value can be used. First, the virtual obstacle is added in the map using Algorithm 2. Next, a check is performed if desired time 't' has passed since the addition of the virtual obstacle, after which, the obstacle is removed by setting the value to free-space value again. Using the same feature, obstacles can be added for certain time periods.


C
$R+\frac{W_r}{2}$
β
P'
$\begin{bmatrix}\sin(\theta+\beta)\\-\cos(\theta+\beta)\end{bmatrix}$
l
$\begin{bmatrix}\cos\theta\\\sin\theta\end{bmatrix}$
r
R
P
θ
$\begin{bmatrix}\sin\theta\\-\cos\theta\end{bmatrix}$
W_r

(a) (b) (c)

Figure 1: Robots used in the experiment, and their motion model. (a) Pioneer P3DX robot, (b) Turtlebot robot, (c) Motion Model.

4 Experiments and Results

Two experiments were carried in real environment to test the influence of virtual obstacles in path planning. We used Pioneer P3DX robot (Fig.1(a)) and Turtlebot robot (Fig.1(b)). Both the robots were equipped with distance sensors (RGBD sensor and UHG-08LX laser range sensor). The distance sensor is accurate within ±30 mm within 1 m, and within 3% of the detected distance between 1 and 8 m. The angular resolution is approximately 0.36 degrees.

4.1 Motion Model

We first describe the motion model of the robots. Both robots are differential drive robots. The distance between the left and the right wheel is W_r, and the robot state at position P is given as $[x, y, \theta]$. From Figure 1(c), turning angle β and the radius of turn R is calculated as,

$$\begin{gathered} r = \beta \cdot (R + W_r),\ l = \beta \cdot R,\ \therefore \beta = \frac{r-l}{W_r}, \\ R = \frac{l}{\beta}, \beta \neq 0. \end{gathered} \tag{1}$$

The coordinates of the center of rotation (C, in Figure 1(c)), and new heading θ' are calculated as,

$$\begin{bmatrix} Cx \\ Cy \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} - \left(R + \frac{W_r}{2} \right) \cdot \begin{bmatrix} \sin\theta \\ -\cos\theta \end{bmatrix}. \tag{2}$$

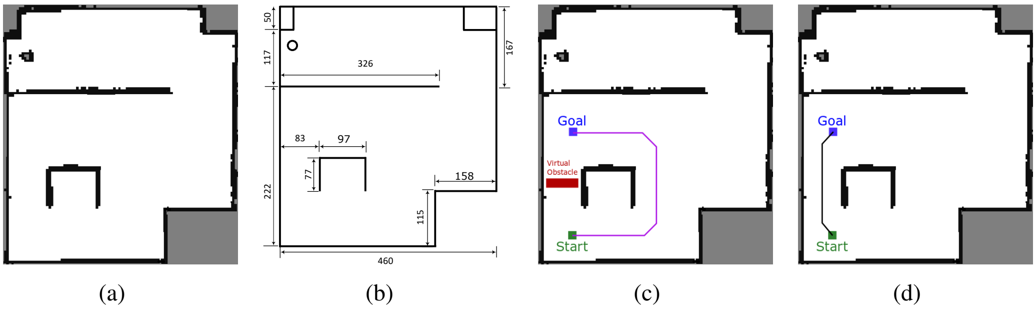
$$\theta' = (\theta + \beta) \mathrm{mod} 2\pi, \tag{3}$$

from which the coordinates of the new position P' are calculated as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} Cx \\ Cy \end{bmatrix} - \left(R + \frac{W_r}{2} \right) \cdot \begin{bmatrix} \sin\theta' \\ -\cos\theta' \end{bmatrix}, \beta \neq 0 \implies r \neq l. \tag{4}$$

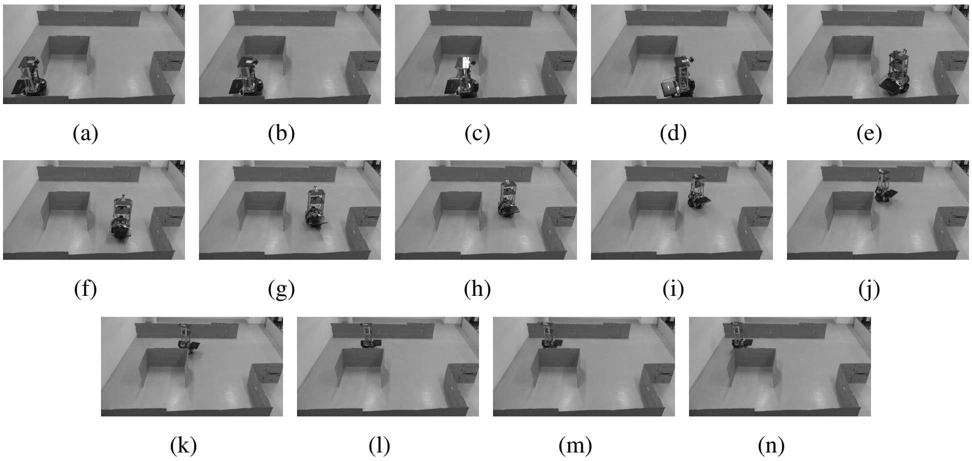
If $r = l$, i.e., if the robot motion is straight, the state parameters are given as $\theta' = \theta$, and,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + l \cdot \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}, (l = r). \tag{5}$$



(a) (b) (c) (d)

Figure 2: Experiment-1. (a) Gridmap, (b) Map dimensions, (c) Path planning with TVO at $t = 5$ min. The shortest path is planned avoiding the virtual obstacle. (d) Path planning with TVO at $t = 25$ min. The transient virtual obstacle was effective only during time period ($t = 0$ min to $t = 20$) min. Hence, the path is planned without the virtual obstacle.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n)

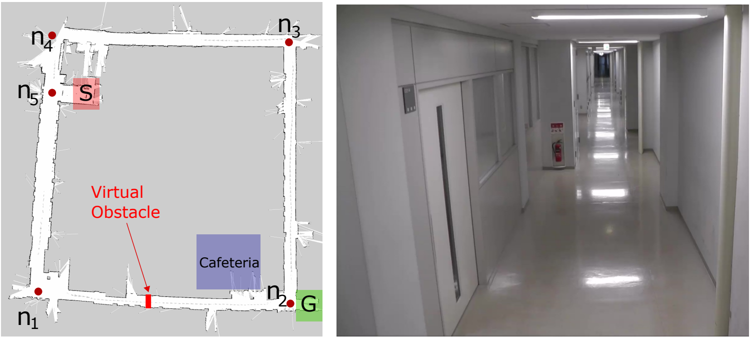
Figure 3: Navigation of P3DX robot considering TVO shown in Fig.2(c) in successive time (a) to (n). The robot avoids shorter path on the left as virtual obstacle was effective.

4.2 Experiment 1

This section explains the first experiment conducted to test the proposed TVO algorithm. Figure 2 shows the setup of the experiment. Figure 2(a) shows the gridmap of the test environment used to test the proposed TVO. The dimensions of the map are 444 × 490 pixels. The white portions of the map represents open space, whereas the black grids represents the obstacles. The dimensions of the map are given in Fig.2(b).

The start and goal locations of the robot are marked in Fig.2(d) in green and blue color, respectively. The top-left corner of the map in Fig.2(d) is marked $(0,0)$. Pioneer P3DX robot (Fig.1(a)) was used in this experiment, and its start location was set to $(64,435)$ whereas the goal location was set to $(66,240)$. A* algorithm [17] was chosen for path planning, however, any other algorithm can also be chosen. The grid based navigation had one unit cost for forward, back, left, and right movement, whereas, for diagonal movement the cost was $\sqrt{2}$ units.

At the start of the experiment virtual obstacle was placed in the map between the locations $x = 15, y = 328$ (top-left) and $x = 75, y = 345$ (bottom-right). The dimensions of the virtual obstacle was 60×17. The virtual obstacle is shown as a red block in Fig.2(c).



(a) (b)

Figure 4: Experiment-2. (a) Gridmap of the environment. 'S' and 'G' are the start and goal locations, respectively. The locations of cafeteria and virtual obstacle are marked. (b) Section of environment showing a section of passage.

The obstacle was placed for a duration of $T = 20$ min. The robot was programmed to start 5 min after the induction of virtual obstacle at $t = 5$ min. Figure 2(c) shows the planned path of the robot from start to the goal location in purple color. It is clear that the path is generated considering the virtual obstacle in the map. It is also the shortest path considering the virtual obstacle.

Figure 2(d) shows the results of path planning at $t = 25$ min. The transient virtual obstacle was effective only during time period ($t = 0$ min to $t = 20$) min. Hence, the path shown in black color is planned without the virtual obstacle this time.

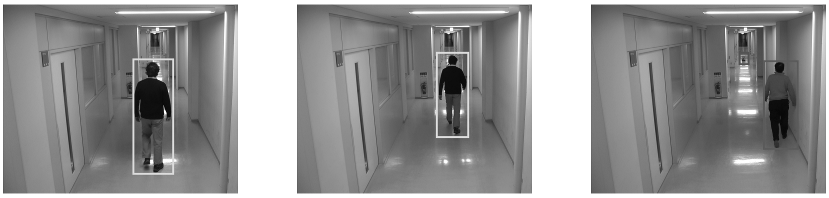
Figure 3 shows the navigation of P3DX robot considering TVO shown in Fig.2(c). Figure 3(a) shows the start location of the robot, and Fig.3(b) to Fig.3(m) shows the snapshots at different times until the robot reaches its goal location shown in Fig.3(n). The robot avoids shorter path on the left as virtual obstacle was effective at the time of navigation. Further, once the paths have been planned and virtual obstacle still exists, cache based approaches [28] can be used for efficiency.

4.3 Experiment 2

The second experiment was conducted in a more realistic scenario of the engineering building of Hokkaido University. This building has a cafeteria, and there are many students accessing the cafeteria during the lunch time of of the university (around 12:00 PM). This causes more congestion along the passage (Fig.4(b)) on which the cafeteria is located. The aim of this experiment was to block the passage along the cafeteria during the peak lunch time of the university, so that the robot goes through free passages. We used Turtlebot robot shown in Fig.1(b). This is also a differential drive robot, and its motion model is same as given in Section 4.1.

The grid-map of the building environment with turn-nodes $n_1, \cdots, n_5$ is shown in Fig.4(a). Robot's start (S), and goal (G) locations are marked along with the cafeteria. The experiment was conducted during the peak lunch-time of university (11:30-13:30) to test path planning in congestion. Similar to Experiment 1 shown in Section 4.2, A* algorithm [17] was chosen for path planning.

The total size of the grid-map in Fig.4(a) was (1279 × 1327) pixels. A virtual obstacle was added between top-left coordinates (576,1161) and bottom-left coordinates (596,1214).



(a) (b) (c)

Figure 5: Congestion in the passage marked n_1n_2 in Fig.4(a).

Table 1: Generated Paths With and Without Transient Virtual Obstacles.

Method	Path (Refer Fig.4(a))	Path Length
Without Virtual Obstacle	$S \rightarrow n_5 \rightarrow n_1 \rightarrow n_2 \rightarrow G$	2038
With Virtual Obstacle	$S \rightarrow n_5 \rightarrow n_4 \rightarrow n_3 \rightarrow n_2 \rightarrow G$	2450

This obstacle was added to prevent robot path generation through the passage near the cafeteria which generally has high congestion of people. The path planned without the proposed TVO was: $\{ S \rightarrow n_5 \rightarrow n_1 \rightarrow n_2 \rightarrow G\}$. Notice that this passes through the congested passage n_1n_2. On the other hand, the path planned by the robot considering the virtual obstacle using the proposed TVO was: $\{ S \rightarrow n_5 \rightarrow n_4 \rightarrow n_3 \rightarrow n_2 \rightarrow G\}$. Figure 5 shows the timely snapshots of congestion on the passage along the cafeteria. The path planned by the robot using the proposed TVO method clearly avoids this congested passage n_1n_2 on which the virtual obstacle was placed.

As summarized in Table 1, the shortest path length without considering the virtual obstacle was 2038 units. Whereas, the path length while considering virtual obstacles was 2450 units. Notice that, this path is longer than the shortest path. However, the virtual obstacle was placed to avoid path generation through crowded passage. Hence, it was safer than the path planned by the traditional method.

5 Conclusion

This paper presented transient virtual obstacles. The method induces virtual obstacles to alter the paths of the robots. Virtual obstacles can be placed at places which are either blocked or risky due to high traffic of people. We proposed a way to add and remove the virtual obstacles in the map. The virtual obstacles are transient. For this, a timestamp feature enables automatic removal of the virtual obstacle after certain time, allowing blocking a certain path for certain time period only. Any global path planner can be used and it automatically generates a global path avoiding the virtual obstacles. Thus, relatively safe paths can be generated for mobile robots easily. The advantages of the proposed method were verified with two experiments in different scenarios.

6 Discussion

Several approaches for mobile robot navigation have been proposed. Most of these algorithms assume a static environment. However, real-world scenarios are dynamic in nature. Therefore, there are scenarios in which some of the paths needs to be actively blocked for safety of mobile navigation. To this end, this paper proposed using virtual obstacles which are transient in nature, and can be programmed easily. A major advantage of the proposed method is that the path planner does not need to be altered as virtual obstacles are added and removed in the map. This enables a powerful feature to block or guide robots through specific passages without using actual obstacles, altering path planners, or reprogramming.

References

[1] A. Ravankar, A. Ravankar, Y. Kobayashi, Y. Hoshino, and C.-C. Peng, "Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges," *Sensors*, vol. 18, no. 9, p. 3170, Sep 2018. [Online]. Available: http://dx.doi.org/10.3390/s18093170

[2] A. Ravankar, A. Ravankar, A. Rawankar, Y. Hoshino, and Y. Kobayashi, "Itc: Infused tangential curves for smooth 2d and 3d navigation of mobile robots," *Sensors*, vol. 19, no. 20, p. 4384, Oct 2019. [Online]. Available: http://dx.doi.org/10.3390/s19204384

[3] S. B. Liu, H. Roehm, C. Heinzemann, I. Ltkebohle, J. Oehlerking, and M. Althoff, "Provably safe motion of mobile robots in human environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 1351–1357.

[4] B. Penin, P. R. Giordano, and F. Chaumette, "Minimum-time trajectory planning under intermittent measurements," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 153–160, Jan 2019.

[5] A. A. Ravankar, Y. Hoshino, A. Ravankar, L. Jixin, T. Emaru, and Y. Kobayashi, "Algorithms and a framework for indoor robot mapping in a noisy environment using clustering in spatial and hough domains," *International Journal of Advanced Robotic Systems*, vol. 12, no. 3, p. 27, 2015.

[6] A. Ravankar, A. A. Ravankar, Y. Hoshino, T. Emaru, and Y. Kobayashi, "On a hopping-points svd and hough transform based line detection algorithm for robot localization and mapping," *International Journal of Advanced Robotic Systems*, vol. 13, no. 3, p. 98, 2016.

[7] A. A. Ravankar, A. Ravankar, T. Emaru, and Y. Kobayashi, "Line segment extraction and polyline mapping for mobile robots in indoor structured environments using range sensors," *SICE Journal of Control, Measurement, and System Integration*, vol. 13, no. 3, pp. 138–147, 2020.

[8] T. Regev and V. Indelman, "Multi-robot decentralized belief space planning in unknown environments via efficient re-evaluation of impacted paths," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 5591–5598.

[9] A. Ravankar, A. Ravankar, Y. Kobayashi, Y. Hoshino, C.-C. Peng, and M. Watanabe, "Hitchhiking based symbiotic multi-robot navigation in sensor networks," *Robotics*, vol. 7, no. 3, p. 37, Jul 2018. [Online]. Available: http://dx.doi.org/10.3390/robotics7030037

[10] A. Ravankar, A. A. Ravankar, Y. Kobayashi, and T. Emaru, "Can robots help each other to plan optimal paths in dynamic maps?" in *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, Sep. 2017, pp. 317–320.

[11] A. Ravankar, A. A. Ravankar, Y. Hoshino, and Y. Kobayashi, "On sharing spatial data with uncertainty integration amongst multiple robots having different maps," *Applied Sciences*, vol. 9, no. 13, p. 2753, Jul 2019. [Online]. Available: http://dx.doi.org/10.3390/app9132753

[12] A. Ravankar, A. A. Ravankar, Y. Kobayashi, and T. Emaru, "Avoiding blind leading the blind," *International Journal of Advanced Robotic Systems*, vol. 13, no. 6, p. 1729881416666088, 2016.

[13] M. Fujita, Y. Gotoy, N. Nidez, K. Satohx, and H. Hosobe, "Toward a robot that acquires logical recognition of space," *Information Engineering Express*, vol. 3, no. 4, pp. 1–10, 2017.

[14] D. Delling, P. Sanders, D. Schultes, and D. Wagner, "Engineering route planning algorithms," in *Algorithmics of Large and Complex Networks*, ser. Lecture Notes in Computer Science, J. Lerner, D. Wagner, and K. Zweig, Eds. Springer Berlin Heidelberg, 2009, vol. 5515, pp. 117–139. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-02094-0_7

[15] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006, available at http://planning.cs.uiuc.edu/ [Accessed: 11-02-2016].

[16] J.-C. Latombe, *Robot Motion Planning*. Norwell, MA, USA: Kluwer Academic Publishers, 1991.

[17] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 4, no. 2, pp. 100–107, July 1968.

[18] A. Stentz and I. C. Mellon, "Optimal and efficient path planning for unknown and dynamic environments," *International Journal of Robotics and Automation*, vol. 10, pp. 89–100, 1993.

[19] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 4, pp. 566–580, Aug 1996.

[20] S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning," Tech. Rep., 1998.

[21] Y. Hwang and N. Ahuja, "A potential field approach to path planning," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 1, pp. 23–32, Feb 1992.

[22] A. Aggarwal, A. Kukreja, and P. Chopra, "Vision based collision avoidance by plotting a virtual obstacle on depth map," in *The 2010 IEEE International Conference on Information and Automation*, June 2010, pp. 532–536.

[23] and M. H. Ang and H. K. and, "Virtual obstacle concept for local-minimum-recovery in potential-field based navigation," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, April 2000, pp. 983–988 vol.2.

[24] C. Luo, S. X. Yang, H. Mo, and X. Li, "Safety aware robot coverage motion planning with virtual-obstacle-based navigation," in *2015 IEEE International Conference on Information and Automation*, Aug 2015, pp. 2110–2115.

[25] K. Ueno, T. Kinoshita, K. Kobayashi, , and K. Watanabe, "Development of a robust path-planning algorithm using virtual obstacles for an autonomous mobile robot," *Journal of Robotics and Mechatronics*, vol. 27, no. 3, pp. 286–292, 2015.

[26] A. A. Ravankar, A. Ravankar, C. Peng, Y. Kobayashi, and T. Emaru, "Task coordination for multiple mobile robots considering semantic and topological information," in *2018 IEEE International Conference on Applied System Invention (ICASI)*, April 2018, pp. 1088–1091.

[27] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[28] A. Ravankar, A. A. Ravankar, Y. Kobayashi, C. Peng, and T. Emaru, "Real-time multi-robot path planning revisited as a caching problem," in *2018 IEEE International Conference on Applied System Invention (ICASI)*, April 2018, pp. 350–353.