J Electr Eng Technol.2016; 11(5): 1441-1448
http://dx.doi.org/10.5370/JEET.2016.11.5.1441

ISSN(Print) 1975-0102
ISSN(Online) 2093-7423

Path Planning for Autonomous Mobile Robot Based on Safe Space

Jong-Hun Park* and Uk-Youl Huh†

Abstract – Path planning is essential for the autonomous navigation of mobile robots. Researchers have been working on ensuring the safe navigation of mobile robots; however, it is impossible to secure the absolute safety of a mobile robot without environmental information. Nevertheless, passive safety of a mobile robot can be secured. With the aim of ensuring safe path planning of a mobile robot, a safety space is proposed in this work by using the parameters of stopping distance and hazard point. Mobile robots should formulate path plans to bypass crossroads or corner areas where their field of view is limited, and they should also be capable of reducing their movement speed to secure safe driving. We demonstrate through extensive simulations that the developed SGPP (safe global path planning) method outperforms the classical A* and PRM (probabilistic roadmap) algorithms. It improves the navigation time and length of the robot path in comparison to the PRM algorithm and reduces the navigation time by up to 26% compared to the classical A* algorithm for safe navigation. In addition, results of an experiment conducted on a real robot show that the SGPP method finds a safe path with limited velocity for safe navigation.

Keywords: Path planning, Stopping distance, Hazard point, Safe space, Safe navigation

1. Introduction

Since the occurrence of the Fukushima nuclear disaster in Japan in 2011, there has been an increasing need for autonomous robots that can cope with hazardous situations. In particular, concerns for the development of practical robots that can work in severe environments have increased. The DARPA (US Defense Advanced Research Projects Agency) Robotics Challenge (DRC) has been initiated as an effort to achieve this goal and held every year since 2012.

The design of autonomous mobile robots should focus more on their safe navigation than on their travel time or distance. The safety of robots has been an issue that has been addressed for a long time in the field of mobile robots, and most works have focused on the task of collision avoidance [1, 2].

The potential field method presented by Borenstein and Koren creates a path based on the attraction and repulsion forces, which facilitates convenient mathematical analyses and application and thus makes it useful for being employed as an obstacle avoidance method [3]. The VFH (vector field histogram) method was developed to improve the practical application of the potential field method and to solve the local minimum problem [4-6].

The abovementioned methods exhibited excellent obstacle avoidance capabilities in static environments. However, in dynamic environments, they could not secure the safety of mobile robots. Fujimura and Samet proposed a static analysis method for avoiding dynamic obstacles [7]. Further, Ko and Lee presented a method that considers only the speed of moving obstacles [8]. Ge and Cui proposed a new potential field method by considering the speeds of the robot and obstacles [9]. The DWA (dynamic window approach) method has been widely used for obstacle avoidance and fast driving in local path planning [10].

However, these methods cannot secure the safety of robots in real environments, because of the partial information on the environment obtained through the limited FoV (field of view) of the sensor. With respect to the problem of the limited FoV, Fraichard and Asamark discussed the safety issue on the basis of the ICS (inevitable collision states) in a static environment [11]. Krishna et al. performed the speed decomposition of a path in a dynamic environment [12].

The braking ICS approach method proposed recently by Bouraine et al. became the basis for safety-based driving [13]. In their study, Donald et al. added physical analyses and proposed a kinodynamics approach [14, 15]. Studies related to ICS were inspiring achievements for safety issues; however, the major disadvantage of this approach is that the operation time increases in a complex environment. Therefore, it does not secure safe navigation in real time.

In this paper, we propose SGPP (safe global path planning) by considering the safe configuration space. This space is the passive safety space of a mobile robot. Passive safety implies that the robot is guaranteed to be at rest in the event of a collision. To secure the safe path planning of a mobile robot, the stopping distance (free running distance, braking distance) and hazard point (i.e., vertex point) are

† Corresponding Author: Dept. of Electrical Engineering, Inha University, Korea. (uyhuh@inha.ac.kr)
* Dept. of Robot Engineering, Inha University, Korea. (patch_hanl@nate.com)

Received: October 29, 2015; Accepted: March 8, 2016

used in this work. The limited velocity for SGPP is also determined by considering the safety space. SGPP is performed using the A* algorithm with limited velocity.

The paper is organized as follows. In section 2, a safe configuration space is defined in a global environment. In section 3, the proposed SGPP method implemented using the A* algorithm based on the concept of safety space is presented. In section 4, the feasibility of the proposed method is validated through a simulation. In section 5, the proposed SGPP method is validated through comparisons of its performance with those of other methods in actual experiments in a real-world environment. Finally, conclusions of this study on SGPP for a mobile robot are presented in section 6.

2. Definition of Safe Space

A safety space includes all the possible kinematic positions of the robot that would ensure its safe distance from the surrounding obstacles, it also refers to all the spaces that would allow maintenance of minimum safety distances of the robot from the obstacles. Finally, the definition of safe space contains the minimum space required for stopping of the mobile robot to prevent its collision with an obstacle.

2.1 Safe configuration space

A safe configuration space is generated in order to ensure the passive safety of a mobile robot in its path planning. This passive safety is aimed at securing the minimum safety of the robot in a limited FoV [16]. The configuration space (C) of a mobile robot in two-dimensional (2D) space can be divided into a free space ($\mathrm{C_{free}}$) and configuration space obstacles (CO) [17]. The configuration space is the space of the robot that includes all possible configurations of its position. Therefore, the position of the robot represents a point in the workspace. Then, the path planning problem is simplified regardless of the kinematic condition of the robot. The configuration space obstacle is given as

$$\mathrm{CO_i} = \{q \in C : R(q) \cap O_i \neq \emptyset\}, i = \{1, \dots, \mathrm{k}\} \tag{1}$$

where R(q) is the position of the robot, O_i are the obstacles, and k is the number of obstacles. $\mathrm{CO_i}$ will be an extension of obstacles in the configuration space according to the size of the robot. The remaining area, i.e., the area excluding the area of the obstacles in the configuration space, is termed the free space, and it can be expressed as

$$\mathrm{C_{free}} = \mathrm{C} - \cup_{\mathrm{i}=1}^{\mathrm{k}} \mathrm{CO_i} \tag{2}$$

where $\mathrm{C_{free}}$ is the set of configurations in which the robot does not intersect any obstacles. In this free space, the robot can move around freely; however, the safety of the robot in this space cannot be guaranteed. This is because in free space, the robot would need space to stop or space to avoid obstacles before it collides with them [13]. Moreover, such a space should be expanded further if no space were available for sensor information. This area is called the risk space $\mathrm{R_s}$, and the remaining space, i.e., the space excluding the risk space, is called the safe space $\mathrm{S_s}$. The following Eq. (3) relates the risk space and the safe space.

$$\mathrm{S_s} = \mathrm{C_{free}} - \mathrm{R_s} \tag{3}$$

The space of the risk area should be generated by considering two distance concepts. The first is the stopping distance of the robot when it is moving at its highest speed, and the second is the risk distance of the robot on account of an unavailable FoV.

2. Calculation of Risk Distance via Stopping Distance

The stopping distance ($\mathrm{d_s}$) can be expressed by using the free running distance ($\mathrm{d_f}$) and the braking distance ($\mathrm{d_b}$). The free running time is the time required by a robot to recognize obstacles and to start the braking action; the distance that the robot moves during this time is called the free running distance. The distance that the robot moves from the start of braking until coming to a complete stop is called the braking distance. The stopping distance of robot moving at its maximum velocity can be calculated as

$$\begin{gathered} \mathrm{d_s} = \mathrm{d_f} + \mathrm{d_b} \\ \mathrm{d_f} = (\mathrm{t_s} + \mathrm{t_p}) \times \mathrm{v_{max}} \\ \mathrm{d_b} = \frac{\mathrm{v_{max}}^2}{2\mathrm{fa}} \end{gathered} \tag{4}$$

where $\mathrm{t_s}$ is the time required to acquire sensor information, t $\mathrm{t_p}$ is the time taken to process the information in the processor, $\mathrm{v_{max}}$ is the maximum velocity of the mobile robot, f is the friction force, and a is the acceleration.

The value of $\mathrm{d_f}$ is dependent on the characteristics of the sensor and the performance of the processor of the robot, whereas the value of $\mathrm{d_b}$ is dependent on the speed of the mobile robot, friction force of the surface, and performance of the brake. The value of $\mathrm{d_s}$ increases with an increase in the speed of the mobile robot. To prevent the collision of the robot with unidentified objects encountered in the area of a limited FoV or corners, the additional concept of risk distance is required.

2.3 Hazard point

Although the mobile robot might be equipped with the sensors required for driving, there exist spaces where the sensor information cannot be obtained. Examples of such

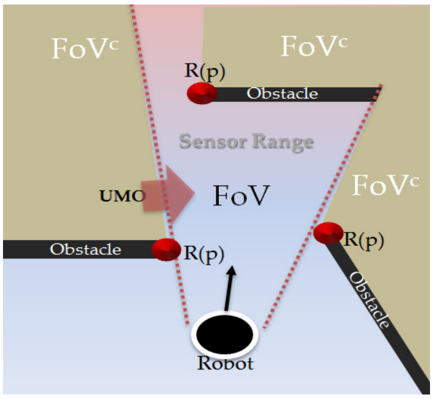


Fig. 1. Limited field of view of mobile robot

spaces include the corner areas of buildings in an urban space. In such spaces, there is a potential risk of collision, and therefore, slow driving or a safe bypass plan is needed. Fig. 1 below illustrates an example of a limited FoV of the mobile robot.

The mobile robot can handle a collision with obstacles in the space secured by the FoV, as illustrated in Fig. 1; however, the space that is outside the FoV^c(complement of FoV), which is the space unavailable for sensor information, is termed hazardous space, in which a mobile robot might collide with unidentified obstacles.

The FoV is the space that a robot could reach with its view, whereas the FoV^c is the space that is beyond the FoV of a robot, i.e., where the robot cannot reach.

Vertex R(p) denotes the risk points in the workspace. The robot may collide with a UMO (unidentified moving obstacle) that appears in the FoV^c. Fig. 2 shows the stopping and risk distances from obstacles in the configuration space.

Connecting the vertices perpendicular to the vertical segment d_s extended from the segment $\overline{AB}$ yields the segment $\overline{A'B'}$, which is parallel to the segment $\overline{AB}$. Further, connecting the vertices perpendicular to the vertical segment d_s extended from the segment $\overline{BC}$ yields the segment $\overline{B'C'}$, which is parallel to the segment $\overline{BC}$.

The intersection of segments $\overline{A'B'}$ and $\overline{B'C'}$ is called the virtual risk point (R'_p). The risk distance d_h from R_p to R'_p can be termed the hazard distance.

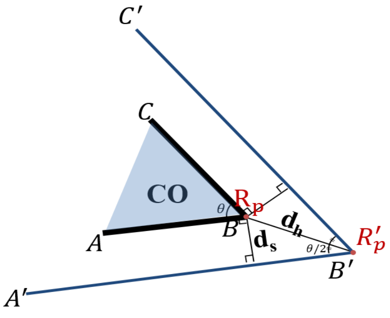


Fig. 2. . Safe distances from obstacle

$$d_h = \eta\left(\frac{d_s}{sin(\theta/2)}\right) \tag{5}$$

where η is the safety gain. With a decrease in the interior angle R_p, the risk distance d_h increases, and vice versa. The risk distance can be calculated in accordance with the magnitude of the corner angle. The range of the interior angle is $0° < \angle B^{\circ} < 180°$.

Algorithm 1 described below is an algorithm for checking the risk space, with the global map given in vector format.

Algorithm 1. Check of risk space in configuration space

```
Input: $\forall Obstacle \subset \mathbb{R}^2$
Output: Risk space ($R_s$)
for $\forall Obstacle$ do
    calc $CO_i\ i = \{1, \dots, k\}$
end for
for $\forall CO$ do
        for $\forall C_{free}$ do
            $d_s = d_f + d_b$
            $d_h = d_s / sin(\theta/2)$
            calc risk point $R'_p(d_s, d_h)$
        end if
    $R_s = \cup_{i=1}^{k} R'_{pi}$
end for
```

Configuration space obstacles are created with all kinds of polygonal shapes with regards to the size of a mobile robot. For all kinds of configuration space obstacles, R_p is sized up and the risk space is checked.

2.4 Limited velocity according to risk space

The risk distance is calculated in the configuration space as described in section 2.3. It is then possible to calculate the limited velocity of the mobile robot. The risk area R_s (x) can be defined as

$$R_s(\boldsymbol{x}) = \begin{cases} \left(1 - \frac{(\mathrm{D - robot_{radius}})}{d_h}\right)^2, & D < \mathrm{d_h} : Rsik\ area \\ \quad 1, & otherwise \end{cases} \tag{6}$$

Where D is the distance between the risk point and the position of the robot. R'_p denotes a specified risk point, which is necessary to secure at least the minimal safety of the robot. The value of $R_s(x)$ lies in the interval of $0 < R_s(x) < 1$.

Fig. 3 shows the risk space of a rectangular obstacle. This figure shows a 2D map for path planning. The cell decomposition method is used to generate a risk map according to Algorithm 1 and Eq. (6).

A safe cell is a free space from the conflict. it takes the value of 0, an obstacle cell takes the value of 1, and a risk cell takes a value of $0 < R_s(x) < 1$. The risk values are 1 in a corner of the obstacle and they converge to zero as the robot moves farther from the corner. In particular, the risk value indicates a wider area at the vertices. The risk space of the mobile robot given in Eq. (6) can be used to express the limited velocity of the robot as follows:

$$V_{limit}(\boldsymbol{x}) = \begin{cases} V_{max}\big(1 - R_s(\boldsymbol{x})\big), & \boldsymbol{x} \notin Obstacle \\ 0, & otherwise \end{cases} \tag{7}$$

where V_{max} is the maximum velocity of the mobile robot. Fig. 4 below shows results of simulation of the limited velocity of the mobile robot for various shapes of the obstacle.

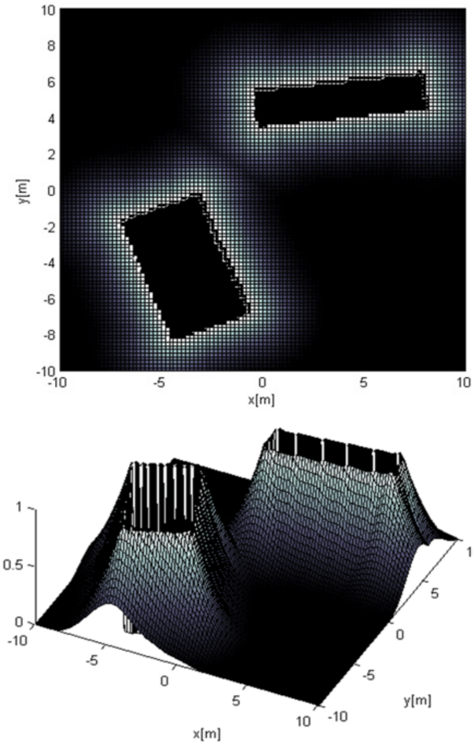
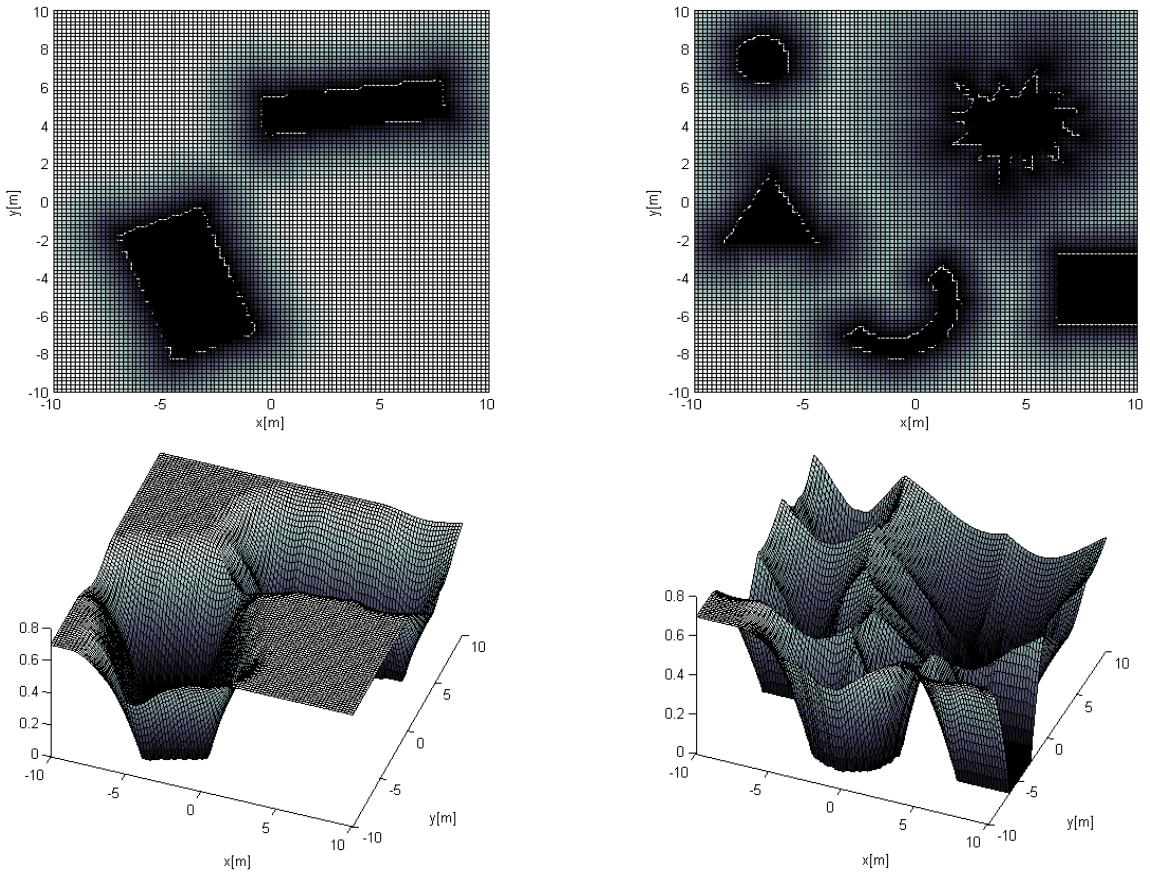


Fig. 3. Risk space of rectangular obstacle

3. SGPP using A* Algorithm

Next, we perform safety path planning by using a grid map considering the risk index. Algorithm 2 below is the SGPP method implemented using the A* algorithm in consideration of the safety cost.



Limited velocity in the case of rectangular obstacle (a)

Limited velocity in the case of obstacles of various shapes (b)

Fig. 4. Limited velocity according to risk space

Algorithm 2 will find the safest and most optimal path by using a grid map containing the risk information. The input is a graph of the coordinates of the robot. The coordinates of the robot are termed nodes n, which lie in the free space. n_best denotes the current node. OpenList refers to the priority queue, and ClosedList contains all processed nodes. The safety cost function is expressed as

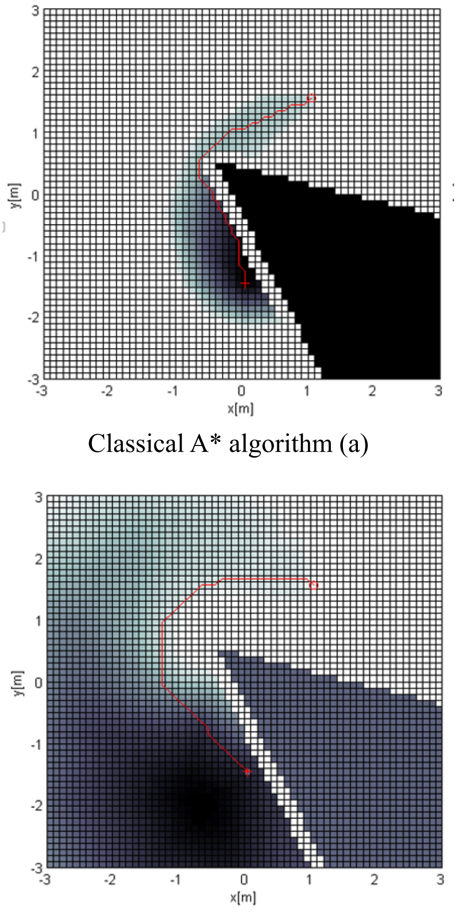
Algorithm 2. SGPP method implemented using A*

```
Input: A graph with risk cell
Output: A path between start and goal nodes
do
    Pick $n_{best}$ from $OpenList$ such that $s(n_{best}) \leq s(n), \forall n \in OpenList$.
    Remove $n_{best}$ from $OpenList$ and add to Closed$List$.
    If $n_{best} == q_{goal}$, EXIT.
    Expand $n_{best}$ : for all $x$ that are not in Closed$List$.
    if x ∉ OpenList then
        add $x$ to $OpenList$
     else if $c(n_{best}) + g(n_{best}(S_s = 1 - R_s)) < c(x)$ then
        inset $n_{best}$ into $x's$ backpointer
    end if
while($OpenList! = empty$)
```

$$s(n) = c(n_{best}) + g(n_{best}(S_s = 1 - R_s) * Ed) \qquad (8)$$

where $c(n_{best})$ is the sum of safety costs from the start node to the current node and $g(n_{best}(S_s = 1 - R_s) * Ed)$ is the estimated safety cost from the start node to the target node. Here, Ed denotes the value of the Euclidean distance from the current node to the target node.

The estimated safety cost from the current node to the target node will be a product of the safety cost and the value of the Euclidean distance. A node will be created by exploring the safest peripheral nodes, from the start node to the target node. Fig. 5 shows a comparison of the global path results of the SGPP method with those of the classical A* algorithm.



Classical A* algorithm (a)

SGPP method (b)

Fig. 5. Comparison of global paths generated by classical A* algorithm and SGPP method

4. Simulation Results

The MATLAB and ARIA simulators were adopted for the mobile robot Pioneer3DX (Mobile Robots Inc.). The applied maximum speed, maximum acceleration, and maximum deceleration were 1 m/s, 0.3 m/s^2, and 0.6 m/s^2, respectively. The map employed was a 2D map with a 100×100 resolution, and the lattice size was set as 30 cm/grid.

In this study, the SGPP method was implemented using the A* algorithm based on the concept of safety space, as described in section 3. Here, we present an extensive simulation study conducted for evaluating the efficiency of the proposed method.

The objective of this simulation is two-fold: first, we wanted to compare the performances of the proposed method, classical A* algorithm, and the PRM (probabilistic roadmap) algorithm [18] in safety space. The DWA (dynamic window approach) [19] has commonly been used for local path planning.

Fig. 6 shows the results of global path planning performed by each of these three methods in safety space.

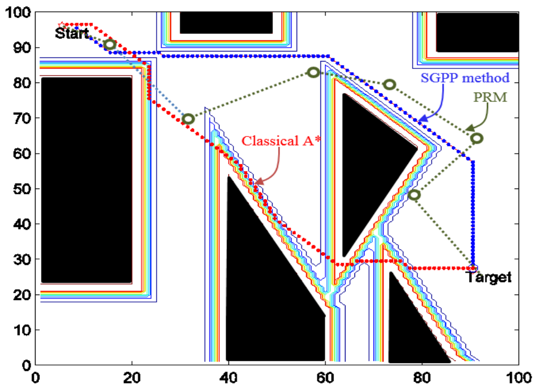
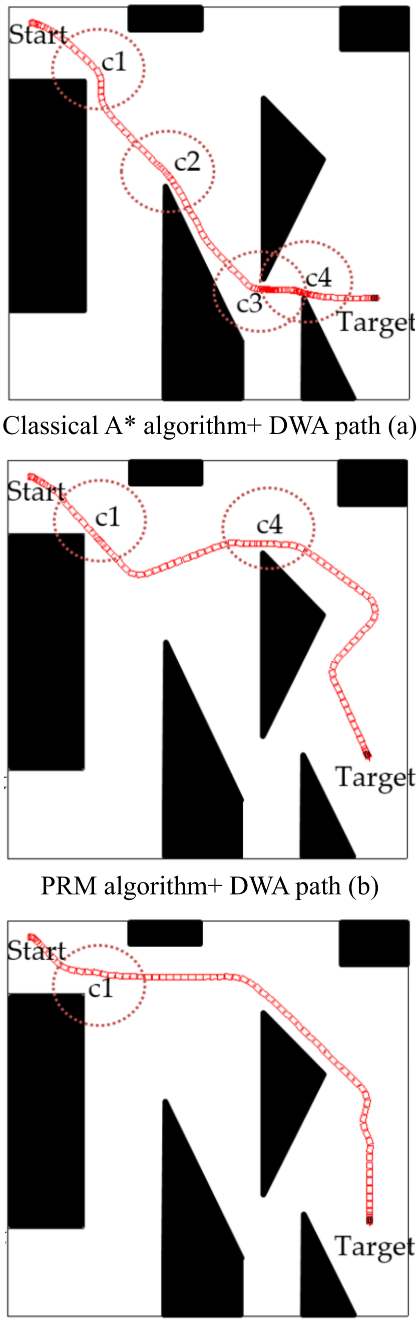


Fig. 6. Global path planning with limited velocity (safety space)



Classical A* algorithm+ DWA path (a)

PRM algorithm+ DWA path (b)

SGPP method + DWA path (a)

Fig. 7. Results of paths generated by the three considered methods

Table 1. Comparison of performance of the three considered algorithms in safety space. (The risk value is the sum of the risk values of the path of the mobile robot on the map. The risk values are as mentioned in Eq. 6.)

	Risk value	Navigation time [s]	Length of path [m]	Mean velocity [m/s]
Classical A*	25.2	61	37	0.58
PRM	6.5	51.5	46	0.86
Proposed method	0	45	38.3	0.87

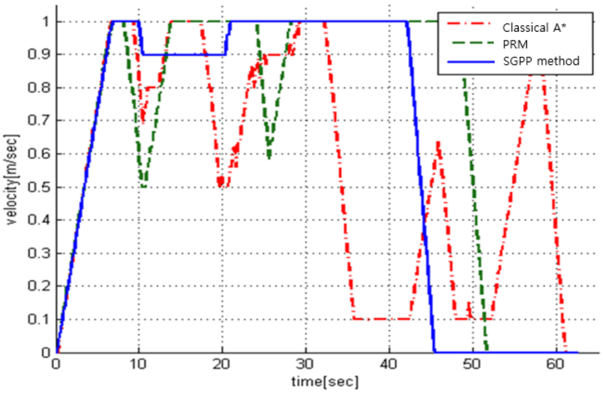


Fig. 8. Velocity profiles for paths created by the three methods

Fig. 7 shows the navigation paths of the mobile robot using the created waypoints shown in Fig. 6.

As shown in Figs. 6 and 7, all three methods rendered paths for the mobile robot to drive from the starting point (8, 95) to the destination point (90, 27) without collision.

The classical A* algorithm (Fig. 7(a)) employed for the simulation created the shortest path for driving the robot; however, the robot reduced its speed in the corner area (Fig. 7(a) c1-c4). In the classical A* algorithm, the navigation time increased because the robot reduced its speed whenever it encountered a corner, as illustrated by the red dashed line in Fig. 8.

The PRM algorithm shown in Fig. 7(b) also yielded a path that reduced the speed of the robot in the corner areas (Fig. 7(b) c1, c4), and owing to the somewhat longer path, it also resulted in a longer navigation time.

Fig. 7(c) shows the created path simulated by the proposed method, where the robot also reduced its speed in the first corner area (Fig. 7(c) c1); however, it then drove at maximum velocity.

As illustrated in Table 1, the proposed method rendered more favourable outcomes than the PRM method in all respects. The length of the path created by the proposed method increased by 3% compared to that created by the classical A* algorithm; however, in the case of the proposed method, the robot drove 26% faster, with an average moving speed of 0.87 m/s. The proposed method was shown to perform the safest and fastest navigation to the goal.

5. Experiments

We attempted to experimentally demonstrate the feasibility of the proposed safe path planning method through its real-world implementation in mobile robots. The method was implemented in C++ using Microsoft Visual Studio and tested on the Pioneer3DX robot,

Table 2. Comparison of simulated results for the three considered methods

	Risk value	Navigation time [s]	Length of path [m]	Mean velocity [m/s]
Classical A*	26	38	7.5	0.23
PRM	31.6	47	8.3	0.16
Proposed method	0	34	9.4	0.27

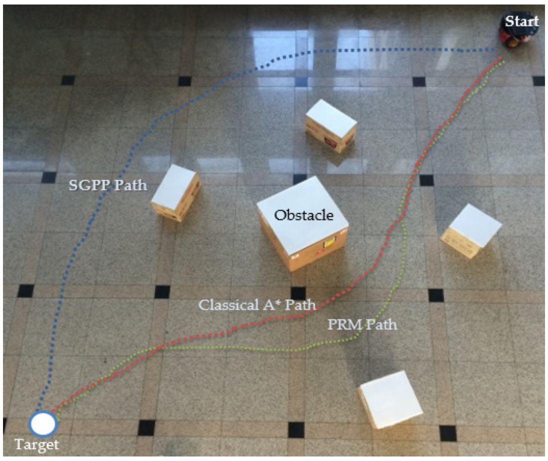


Fig. 9. Experiment with rectangular obstacles

operating under a version of Windows CE.

A snapshot of the experimental environment is shown in Fig. 9. We conducted experiments in an indoor environment (5.6 m × 5.6 m). The five obstacles in this environment were static. Fig. 9 shows the comparative results obtained from an experiment in which the three methods — the proposed SGPP method, PRM algorithm, and classical A* algorithm were executed separately from the start point (0, 0) to the targeted destination point (50, 50). Even though a conclusion may review the main results or contributions of the paper, do not duplicate the abstract or the introduction. For a conclusion, you might elaborate on the importance of the work or suggest the potential applications and extensions.

The proposed method created paths that were 21% and 12% longer than those created by the classical A* algorithm and PRM algorithm, respectively; however, it enabled comparatively faster driving that reduced the running time by 11.5% and 28.7%, respectively, compared to these two methods. It also allowed for safe navigation, with the safety cost reduced in the risk area in the experiment.

As shown by the results of these experiments, the proposed method exhibited better outcomes in terms of the navigation time and safe driving.

Fig. 10 shows the velocity profiles obtained from the three methods employed in the experiment. The average speeds obtained for the classical A* algorithm, PRM algorithm, and proposed method were 0.23 m/s, 0.16 m/s, and 0.27 m/s, respectively; that is, the proposed method rendered the fastest average velocity.

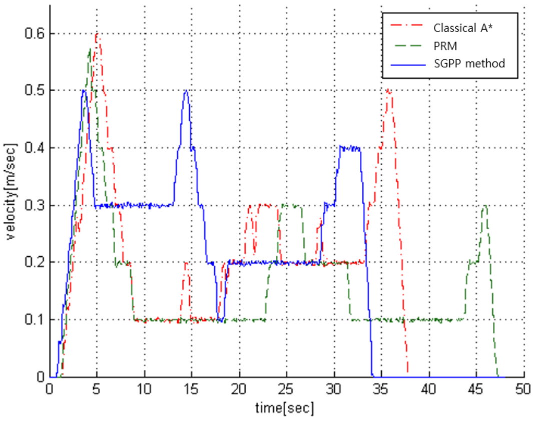


Fig. 10. Velocity profiles for the three methods

The path created by the proposed method was somehow longer than those created by the other two methods. However, the path created by the proposed method allowed safe driving with the highest speed, which reduced the overall running time.

6. Conclusion

In this paper, a safe and fast path planning method was presented that employed the concept of safety space and the A* algorithm. The stopping distances and the risk information at corner points were generated quantitatively, and a safe path was created by incorporating the A* algorithm.

In the planning of a global path, paths that bypass corner areas or crossroads can be created to reduce the risk of collision with obstacles and to secure a safe and fast path to the destination.

This method would specifically be useful for robots that employ inexpensive sensors or for occasions in which places of increased uncertainty might be encountered.

In this study, the performance of the developed SGPP method in terms of bypassing of spaces such as corners and reducing the running speed of mobile robots was validated through simulations and experiments. The SGPP method showed superior performance to the other considered algorithms in terms of the safety cost and navigation time.

The next research step is to employ the proposed safe local path planning method in performing tasks that are more complex, such as navigation in dynamic environments.

References

[1] Ullah, I., et al., Integrated tracking and accident avoidance system for mobile robots. International Journal of Control, Automation and Systems, 2013.

11(6): p. 1253-1265

[2] Becker, M., C.M. Dantas, and W.P. Macedo. Obstacle avoidance procedure for mobile robots. in ABCM Symposium series in Mechatronics. 2006.

[3] Borenstein, J. and Y. Koren, Real-time obstacle avoidance for fast mobile robots. Systems, Man and Cybernetics, IEEE Transactions on, 1989. 19(5): p. 1179-1187.

[4] Borenstein, J. and Y. Koren, The vector field histogram-fast obstacle avoidance for mobile robots. Robotics and Automation, IEEE Transactions on, 1991. 7(3): p. 278-288.

[5] Ulrich, I. and J. Borenstein. VFH+: Reliable obstacle avoidance for fast mobile robots. in Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on. 1998. IEEE.

[6] Mabrouk, M. and C. McInnes, Solving the potential field local minimum problem using internal agent states. Robotics and Autonomous Systems, 2008. 56(12): p. 1050-1060.

[7] Fujimura, K. and H. Samet, A hierarchical strategy for path planning among moving obstacles [mobile robot]. Robotics and Automation, IEEE Transactions on, 1989. 5(1): p. 61-69.

[8] Nak Yong, K. and L. Bum Hee. Avoidability measure in moving obstacle avoidance problem and its use for robot motion planning. in Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on. 1996.

[9] Ge, S.S. and Y. Cui, Dynamic motion planning for mobile robots using potential field method. Autonomous Robots, 2002. 13(3): p. 207-222.

[10] Fox, D., et al. A hybrid collision avoidance method for mobile robots. in Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on. 1998. IEEE.

[11] Fraichard, T. and H. Asama, Inevitable collision states — a step towards safer robots? Advanced Robotics, 2004. 18(10): p. 1001-1024.

[12] Madhava Krishna, K., R. Alami, and T. Siméon, Safe proactive plans and their execution. Robotics and Autonomous Systems, 2006. 54(3): p. 244-255.

[13] Bouraine, S., T. Fraichard, and H. Salhi, Provably safe navigation for mobile robots with limited field-of-views in dynamic environments. Autonomous Robots, 2012. 32(3): p. 267-283.

[14] Donald, B., et al., Kinodynamic motion planning. Journal of the ACM (JACM), 1993. 40(5): p. 1048-1066.

[15] LaValle, S. M. and J. J. Kuffner, Randomized kinodynamic planning. The International Journal of Robotics Research, 2001. 20(5): p. 378-400.

[16] Park, J.-H., J.-H. No, and U.-Y. Huh. Safe Global Path Planning of Mobile Robots Based on Modified A* Algorithm. in The 8th International Conference on Robotic, Vision, Signal Processing & Power Applications. 2014. Springer.

[17] Lozano-Perez, T., Spatial planning: A configuration space approach. Computers, IEEE Transactions on, 1983. 100(2): p. 108-120.

[18] Denny, J., et al. A Region-Based Strategy for Collaborative Roadmap Contruction. in Algorithmic Foundations of Robotics XI, Proceedings of the Eleventh Workshop on the Algorithmic Foundations of Robotics (WAFR 2014), Springer Tracts in Advanced Robotics, to appear. 2014.

[19] Fox, D., W. Burgard, and S. Thrun, The dynamic window approach to collision avoidance. Robotics & Automation Magazine, IEEE, 1997. 4(1): p. 23-33.

Jong-Hun Park He Received the M.Sc. degree in Electrical Engineering from Inha University in 2004. He now pursuing a Ph.D. degree in the Department of Robot Engineering, Inha University of Incheon, Korea. His current research interests include path planning and Robotics.

UK-Youl Huh He Received the M.Sc. degree in Electrical Engineering from Inha University in 2004. He now pursuing a Ph.D. degree in the Department of Robot Engineering, Inha University of Incheon, Korea. His current research interests include path planning and Robotics.