Journal of Intelligent & Robotic Systems (2020) 99:65–77
https://doi.org/10.1007/s10846-019-01112-z

Hybrid Path Planning Based on Safe A* Algorithm and Adaptive Window Approach for Mobile Robot in Large-Scale Dynamic Environment

Xunyu Zhong [1] · Jun Tian [1] · Huosheng Hu [2] · Xiafu Peng [1]

Received: 29 May 2019 / Accepted: 22 October 2019 / Published online: 6 January 2020

Abstract
When mobile robot used in large-scale dynamic environments, it face more challenging problems in real-time path planning and collision-free path tracking. This paper presents a new hybrid path planning method that combines A* algorithm with adaptive window approach to conduct global path planning, real-time tracking and obstacles avoidance for mobile robot in large-scale dynamic environments. Firstly, a safe A* algorithm is designed to simplify the calculation of risk cost function and distance cost. Secondly, key path points are extracted from the planned path which generated by the safe A* to reduce the number of the grid nodes for smooth path tracking. Finally, the real-time motion planning based on adaptive window approach is adopted to achieve the simultaneous path tracking and obstacle avoidance (SPTaOA) together the switching of the key path points. The simulation and practical experiments are conducted to verify the feasibility and performance of the proposed method. The results show that the proposed hybrid path planning method, used for global path planning, tracking and obstacles avoidance, can meet the application needs of mobile robots in complex dynamic environments.

Keywords Mobile robot · Hybrid path planning · Safe A* algorithm · Adaptive window approach · Key path points

1 Introduction

Path planning is a key technique of autonomous mobile robot and a hot issue in mobile robot navigation research [1–4]. Generally, the path planning can be classified into global path planning and local path planning, depending on the nature of the environments and goals [5].

The Global Path Planning (GPP) for known environment has been extensively studied by scholars and researchers [6]. Thereinto, A* algorithm is a very effective method used to search the shortest path in grid map, and it is also one of the most widely used algorithms at present [7–10]. However, the path planned by traditional A* algorithm is adjacent to obstacles, which is not conducive to avoiding collision risk when robot tracking this planned path. Bayili and Polat [11] proposed a path search algorithm in computer games, namely limited-damage A*, which considers damage as a feasibility criterion and takes the collision risk into account to get a safer path. Park et al. created a safe Global Path Planning (SGPP) for mobile robots, which is called Modified A* Algorithm [12]. The computing model of configuration space was used to handle potential risk and the risk cost is introduced into the heuristic function of A*. Aine et al. used the guidance ability of different heuristic functions to develop the Multi-Heuristic A* algorithm for multi-objective planning of mobile robots in complex environments [13].

When the environments contain dynamic changes, Likhachev et al. proposed a real-time A* search algorithm (D* algorithm) [14, 15] to deal with the changes. Moreover, Toll and Geraerts developed a Dynamically Pruned A* algorithm for re-planning in navigation meshes [16]. Dakulovi, et al. proposed a two-way D* algorithm for path planning and re-planning based on weighted cost map [17]. However, D* algorithm and re-planning approach are not suitable for large dynamic environments, their path search time will multiply with the increase of map size [18]. In addition, the path generated by A* algorithm is composed of adjacent grid nodes, and the distance between each path node is very small. In the process of path tracking, it is difficult for a robot to

✉ Xunyu Zhong
zhongxunyu@xmu.edu.cn

[1] School of Aerospace Engineering, Xiamen University, Xiamen, Fujian, China

[2] School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK

achieve smooth path tracking as there are many path nodes and very short distance between the adjacent nodes.

When the environment is unknown, the robots need to deploy sensors for real-time environmental detection and adopt Local Path Planning (LPP) method to avoid obstacles, such as artificial potential field, RRT algorithm, dynamic window approach (DWA) [19–24]. In [25], the DWA method was improved and applied to dynamic obstacles avoidance. This method built the grid model of dynamic obstacles with their velocity prediction, and then made collision avoidance planning for virtual obstacles. In [26], based on DWA, a Guided Dynamic Window Approach (GDWA) was proposed, in which the velocity commands could be chosen to satisfy all the constraints and maximize an objective function to trade off speed, clearance and weighted local orientation based on the local navigable region. Based on the principle of rolling window, we proposed the real-time motion planning methods based on environment modeling, analysis and adaptive window [27, 28] to satisfy the requirements of obstacles avoidance in dynamic complex environments, which can be directly applied in different mobile robots without the difficulty in establishing velocity space needed in the DWA method.

In general, GPP can obtain the optimal path in static environments, but unable to deal with the dynamic changes, and the re-planning approach is hard to get real-time performance in large-scale environments. LPP can be applied to unknown dynamic environments with the real-time obstacles avoidance ability, but is easy to fall into local minimum and cannot guarantee the global optimization. The Hybrid Path Planning (HPP) method aims to combine these two methods to achieve the purpose of global optimization and dynamic avoidance for the robot navigation in large-scale dynamic environments. Lu et al. used Global Voronoi diagram and Local search D* algorithm to create a HPP algorithm for Mobile Robot navigation [29]. In [30], the visual graph based global path planning was combined with potential field method to obtain the shortest path and improve the obstacle avoidance safety for robots. In [31], a global dynamic path planning method integrated with A* algorithm and DWA was proposed for both the required global optimal path and real-time obstacles avoidance, which however requires the accurate dynamic characteristics of the robot for the calculation of the velocity space. At present, to the best of our knowledge, the research of HPP is inadequate to achieve global path planning and simultaneous path tracking and obstacles avoidance (SPTaOA).

With the increasing application of mobile robots in large-scale dynamic environments, such as workshop in factory, airport, or shopping malls, they face with more challenging problems in real-time path planning and SPTaOA. This is the motivation of our work, and the main contribution of this paper is that 1) a safe A* algorithm is designed, and the method of finding & extracting key path points is proposed for smooth path tracking; 2) real-time motion planning for obstacle avoidance based on adaptive window approach is proposed; 3) the HPP based on the safe A* and real-time motion planning is formulated for global path planning and SPTaOA through the switching of the key path points.

The rest of this paper is organized as follows. In Section 2, the global path planning based on safe A* and the method for key path points extraction are presented. In Sections 3, dynamic window approach is proposed for the real-time motion planning, i.e. local path planning or obstacle avoidance. In Sections 4, the method of key path points switching is proposed for simultaneous path tracking and obstacles avoidance. And Sections 5 describes the integrated architecture of the proposed hybrid path planning. Then in Section 6, simulation verification and practical application are conducted to show the feasibility and the performance of the proposed methods. Finally, a brief conclusion is given in Section 7.

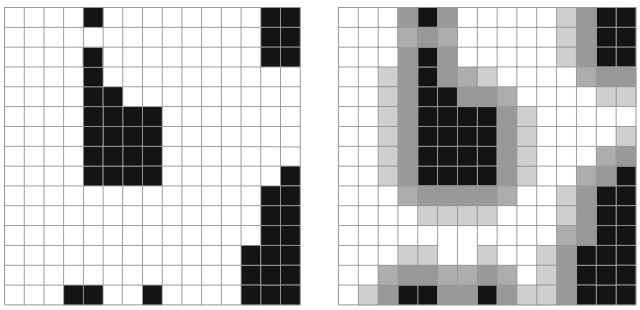
2 Global Path Planning and Key Path Points

Based on the conventional A* algorithm, a safe global path planning method is designed by using configuration space approach (C-space). Then, in order to solve the problem that the planned path has too many grid nodes, key path points are extracted to facilitate the smooth path tracking of the robot.

2.1 Safe Global Path Planning

The initial grid map of the environment is obtained by using the SLAM method, then C-space is used to enlarge the obstacles according to the radius of the robot to obtain a C-space grid map, shown in Fig. 1a. The following risk function $R(n)$ is defined to convert the C-space map to the safe grid map.

$$R(n)\cdot = \cdot\alpha \cdot / \cdot r^2 \tag{1}$$



(a) C-space grid map (b) Safe grid map

Fig. 1 Diagram of the grid maps

Where r represents the distance between the node n and the nearest obstacle node, and $\alpha > 0$ is the risk factor.

The risk cost of all nodes is evaluated by $R(n)$ to obtain the safe grid map with risk areas, as shown in Fig. 1b. It is clear that a layer of risk nodes with decreasing gray values will be generated near obstacle nodes.

Then, the cost function of the traditional A* algorithm is modified as

$$F(n)\cdot = \cdot G(n) \cdot + \cdot H(n) \cdot + \cdot R(n) \tag{2}$$

where $G(n)$ is the actual cost from the starting point **S** to the current node n, $H(n)$ is the estimated cost from the current node n to the goal point **G** (calculated by Manhattan distance), $R(n)$ is the risk value of the current node n.

On the safe grid map, the global path with high security can be obtained using the safe A* algorithm with formula (2) for path planning (by searching in the upper, right, lower and left directions). The obtained global path is denoted as

$$Path\cdot = \cdot\{S, \cdot P_1, \cdot P_2, \cdot \ldots \cdot, \cdot P_w, \cdot G\} \tag{3}$$

where ($\mathbf{P_1}$, $\mathbf{P_2}$, ..., $\mathbf{P_w}$) are the all connected grid nodes of the global path.

As shown in Fig. 2a, the shortest path can be obtained when $R(n)$ is not taken into account; and a relatively safer path can be obtained when risk assessment R(n) is taken into account, as shown in Fig. 2b.

2.2 Key Path Points of the Global Path

The key path points are represented as the subset of ***Path***, i.e.

$$KPath\cdot = \cdot\{S, \cdot KP_1, \cdot KP_2, \cdot \ldots \cdot, \cdot KP_m, \cdot G\} \tag{4}$$

where ($\mathbf{KP_1}$, $\mathbf{KP_2}$, ..., $\mathbf{KP_m}$) $\subset$ ($\mathbf{P_1}$, $\mathbf{P_2}$, ..., $\mathbf{P_w}$).

The method and steps of finding & extracting key path points ***KPath*** from ***Path*** are as follows.

Step 1: Store the starting point **S** in set ***KPath*** as the first key path point and record it as **O**.

Step 2: Set the next path point (node) of **O** as **M**, then judge whether **M** is the goal point **G**, and jump to Step 5 if **M** is the goal point.

Step 3: Set the next path point (node) of **M** as **N**, then judge whether **N** is the goal point **G**, and jump to Step 5 if **N** is the goal point.

Step 4: Calculate the risk value $R(i)$ of any node i on the line segment from **O** to **N**. If $R(i) \geq \varepsilon$, then set **M** as the next key path point after **O**, and add it to ***KPath***, and then set **O** = **M**, jump to Step 2; If $R(i) < \varepsilon$, then move **M** one node backwards along ***Path***, and then return to Step 3.

Step 5: When finding the goal point **G**, all key path points are found, then add **G** to the set ***KPath*** as the last key path point.

The above process of finding & extracting key path points is shown in Fig. 3 as a example. When in the process of path tracking of ***Path***, the key path points in ***KPath*** will be regarded as the real-time subgoal point $\mathbf{G_t}$ of the robot.

As shown in Fig. 3b, when judging whether the line segment ON passes through an obstacle node (i.e. exist $R(i) > \varepsilon$ of any grid node i on the segment ON), the coordinates of node i on the line segment ON need to be calculated first. The grid coordinates of node **O** are denoted as (O.x, O.y), and the coordinates of node **N** are (N.x, N.y). Then we can calculate dx and dy using the following equations.

$$\begin{aligned} \mathrm{d}x &= \mathrm{N}.x - \mathrm{O}.x \\ \mathrm{d}y &= \mathrm{N}.y - \mathrm{O}.y \end{aligned} \tag{5}$$

1) As shown in Fig. 4a, when $|\mathrm{d}x| \geq |\mathrm{d}y|$, the number of nodes on the line segment ON is $s = |\mathrm{d}x| - 1$, and set $\Delta y = |\mathrm{d}y/\mathrm{d}x|$. Then the coordinates of the i-th node on the line segment ON can be calculated as

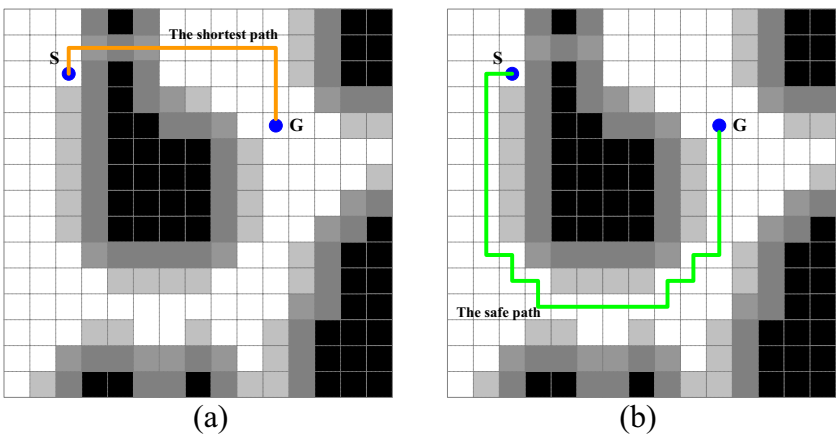
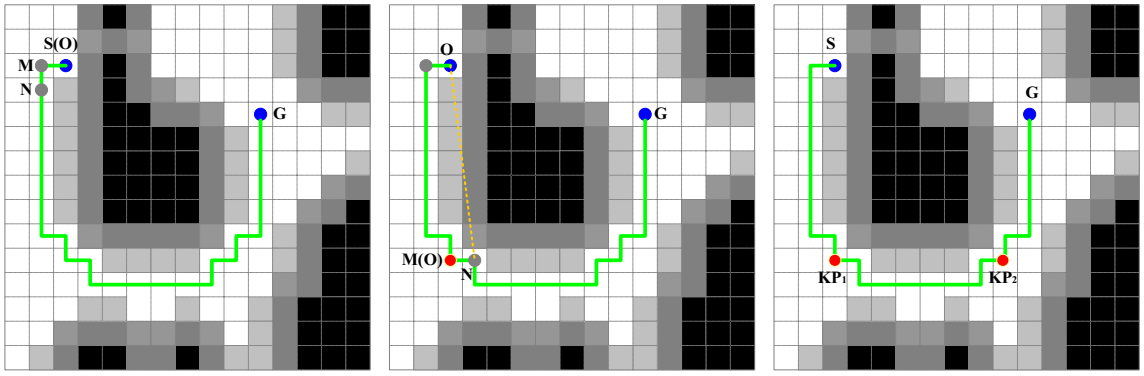


Fig. 2 Diagram of the shortest path and the safe path



(a)Beginning (b) Intermediate (c) Result

Fig. 3 Diagram of the process of finding & extracting key path point

$$\begin{aligned} x_i &= \mathrm{O}.x + \mathrm{sign}(\mathrm{d}x) \cdot i \\ y_i &= \mathrm{O}.y + \mathrm{sign}(\mathrm{d}y) \cdot \lfloor \Delta y \cdot i + 0.5 \rfloor \end{aligned} \tag{6}$$

Where sign(·) is the sign function, $\lfloor \cdot \rfloor$ is the integral function, $i = 1,2,...,s$.

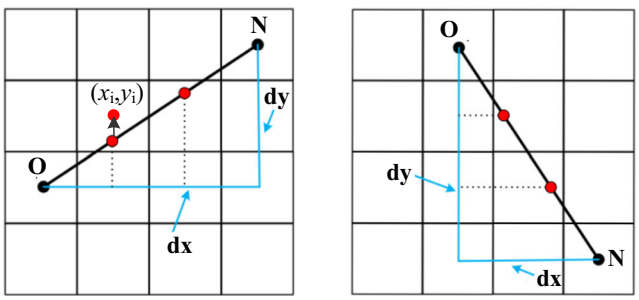
2) As shown in Fig. 4b, when $|\mathrm{d}x| < |\mathrm{d}y|$, the number of nodes on the line segment ON is $s = |\mathrm{d}y|-1$, and set $\Delta x = |\mathrm{d}x/\ \mathrm{d}y|$. Then the coordinates of the i-th node on the line segment ON can be calculated as

$$\begin{aligned} x_i &= \mathrm{O}.x + \mathrm{sign}(\mathrm{d}x) \cdot \lfloor \Delta x \cdot i + 0.5 \rfloor \\ y_i &= \mathrm{O}.y + \mathrm{sign}(\mathrm{d}y) \cdot i \end{aligned} \tag{7}$$

where $i = 1,2,...,s$, (x_i,y_i) are the grid coordinates of the node i on the line segment ON.

3 Motion Planning for Obstacle Avoidance

During the path tracking of mobile robot, it is necessary to avoid new obstacles using Local Path Planning. In this paper, a improved real-time motion planning based on adaptive window approach is used for obstacles avoidance. The planning-control period of the robot is T.



(a) |dx|≥|dy| (b) |dx|<|dy|

Fig. 4 Coordinates of the nodes on the line segment ON

3.1 Real-Time Detection of Local Environment

As shown in Fig. 5, the robot uses 2D LiDAR to detect the current information of obstacles, assuming that the LiDAR coordinate system coincides with the robot coordinate system. The maximum range of measure is d_{max}, the field of view is $[\Phi_{min}, \Phi_{max}]$, the angular resolution is Δ_Φ; the corresponding scanning angle on the robot heading (θ_R) is recorded as Φ_{rob}. After each scan, the range measurements obtained by the LiDAR are $\{d_1, d_2, ..., d_N\}$, the scanning angle of the distance d_i ($i = 1,2,...,N$) is $\Phi_i = \Phi_{min} + (i\text{-}1)\ \Delta_\Phi$.

3.2 The Travelable Direction in Planning Window

As shown in Fig. 6, a virtual planning window is designed within the range of LiDAR, its size (i.e. radius) is r_{win}. In the planning window, the distances $\{d_1, d_2, ..., d_N\}$ are represented as $\{l_1, l_2, ..., l_N\}$, where $l_i = \{\Phi_i, dw_i\}$ is denoted with the

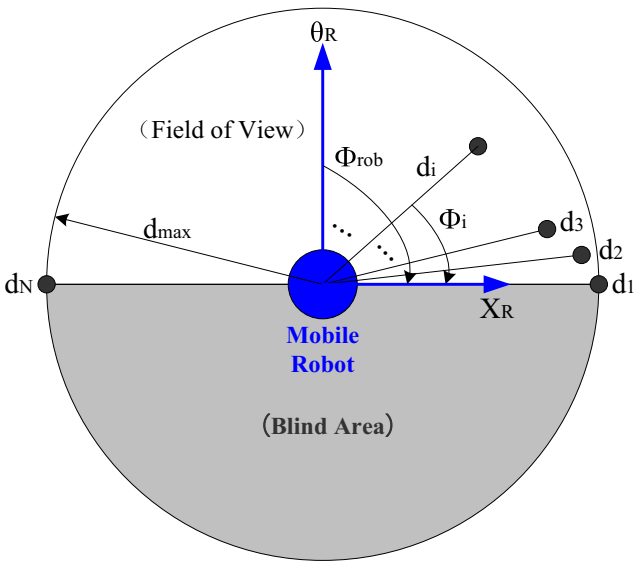


Fig. 5 Diagram of the distance measurement of LiDAR

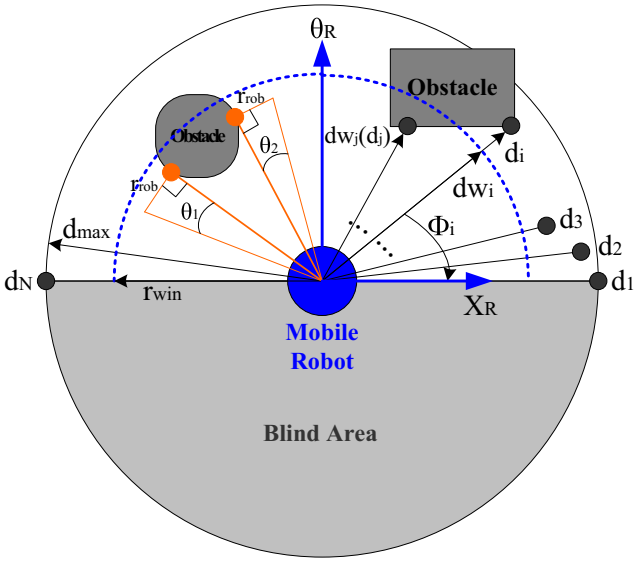


Fig. 6 Diagram of the planning window and the adjustment of the measured distances

scanning angle Φ_i and the corresponding distance dw_i (i = 1,2,...,N).

1) Firstly, d_i will be adjusted to obtain dw_i according to the comparison with r_{win}. The method is: if $d_i \geq r_{win}$, then set $dw_i = r_{win}$; otherwise set $dw_i = d_i$.
2) According to the radius of the robot (r_{rob}), the edges of obstacles need to be expanded, and consequently dw_i will be adjusted again. The method is: for any $dw_j < r_{win}$ (j = 1,2,...,N), if $dw_{j+1} = r_{win}$, then we calculate $\theta_1 = \arctan(r_{rob}/dw_j)$, and let all the distances in the angle θ_1 (i.e. $[\Phi_j, \Phi_j + \theta_1]$) equal to dw_j; if $dw_{j-1} = r_{win}$, then calculate $\theta_2 = \arctan(r_{rob}/dw_j)$, and let all the distances in the angle θ_2 (i.e. $[\Phi_j, \Phi_j - \theta_2]$) equal to dw_j, too.

In the planning window, an example of adjusting the initial range $\{d_1, d_2, ..., d_N\}$ is shown in Fig. 6. Therefore, in the planning window, for any $l_i = \{\Phi_i, dw_i\}$, if $dw_i = r_{win}$, it means that the robot is travelable in the direction Φ_i (i.e. without obstacle).

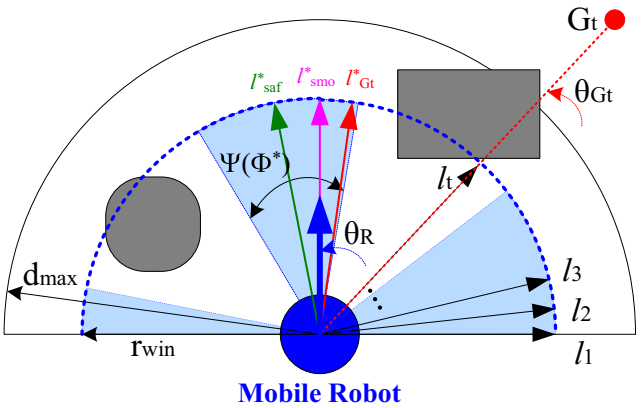


Fig. 7 Selection of the moving direction for next moment

3.3 The Moving Direction for the Next Moment

In the planning window, the set of all travelable directions is called Travelable Angle Region of the robot at the current moment, which denoted as

$$\Psi_{path} = \Psi_{path1} + \Psi_{path2} + \ldots \cdot = \cdot\{(\Phi_i, \cdot dw_i) | \Phi_{\min} \leq \Phi_i \leq \cdot \Phi_{\max} \cdot and \cdot dw_i = \cdot r_{win}, \cdot i = 1, 2, \ldots, N\} \quad (8)$$

As shown in Fig. 7, the current pose of the robot is (x_R, y_R, θ_R), and its current subgoal is $G_t(x_{Gt}, \cdot y_{Gt})$. Then the angle relative to the heading of the robot is $\theta_{GR} \cdot = \cdot \theta_{Gt} - \theta_R$, where $\theta_{Gt} \in [-\pi, \cdot\pi]$, calculated as follow:

- when $x_{Gt} \geq x_R$, the value is

$$\theta_{Gt} \cdot = ^{\arctan}((y_{Gt} - y_R)/(x_{Gt} - x_R)) \quad (9)$$

- when $x_{Gt} < x_R$, $y_{Gt} \geq y_R$, the value is

$$\theta_{Gt} \cdot = ^{\arctan}((y_{Gt} - y_R)/(x_{Gt} - x_R)) + \pi \quad (10)$$

- when $x_{Gt} < x_R$, $y_{Gt} < y_R$, the value is

$$\theta_{Gt} \cdot = ^{\arctan}((y_{Gt} - y_R)/(x_{Gt} - x_R)) - \pi \quad (11)$$

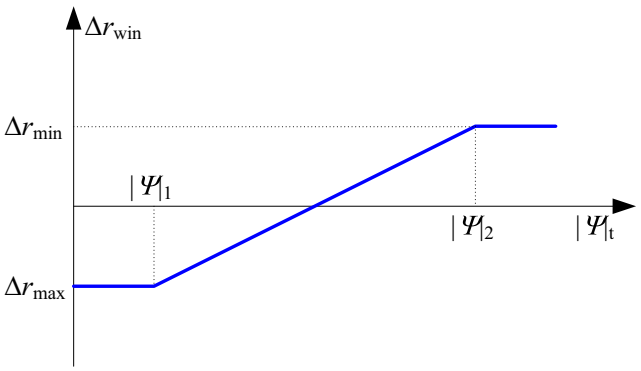


Fig. 8 Functional relation between Δr_{win} and $|\Psi|_t$

Given the nearest scanning measurement to θ_{Gt} is $l_t = \{\Phi_t, dw_t\}$ (where $\Phi_t \approx \Phi_{rob} + \theta_{GR}$), then, in order to move to the subgoal $\mathbf{G_t}$, the optimal direction for moving at the next moment is

$$\Phi^*_{Gt} = \arg \min_{\phi_i \in \psi_{path}} (|\Phi_i - \Phi_t|) \tag{12}$$

The corresponding distance on direction Φ^*_{Gt} is dw_{Gt}, the travelable angle region where dw_{Gt} locates is denoted as $\Psi(\Phi^*)$. If $dw_{Gt+1} = r_{win}$,..., $dw_{Gt+m} = r_{win}$, $dw_{Gt+m+1} < r_{win}$, then $\Phi^*_{lef} = \Phi_{Gt+m}$ is the left edge of $\Psi(\Phi^*)$; If $dw_{Gt-1} = r_{win}$,..., $dw_{Gt-m} = r_{win}$, $dw_{Gt-m-1} < r_{win}$, then $\Phi^*_{rig} = \Phi_{Gt-m}$ is the right edge of $\Psi(\Phi^*)$. For better safety, robot should be as far away from obstacles as possible (especially the dynamic obstacles). So, the safest direction in $\Psi(\Phi^*)$ is

$$\Phi^*_{saf} = \arg \max_{\phi_i \in \Psi(\phi^*)} \left(\min \left(\left| \Phi_i - \Phi^*_{lef} \right|, \left| \Phi_i - \Phi^*_{rig} \right| \right) \right) \tag{13}$$

Considering the smoothness of the robot's movement, the smoothest direction in $\Psi(\Phi^*)$ is

$$\Phi^*_{smo} = \arg \min_{\phi_i \in \psi(\phi^*)} (|\Phi_i - \Phi_{rob}|) \tag{14}$$

As shown in Fig. 7, in the travelable region $\Psi(\Phi^*)$, considering both safety and smoothness, Φ^*_{Gt} is adjusted as

$$\Phi^* = k_{Gt}\Phi^*_{Gt} + k_{saf}\Phi^*_{saf} + k_{smo}\Phi^*_{smo} \tag{15}$$

where $k_{Gt} \geq 0$, $k_{saf} \geq 0$, $k_{smo} \geq 0$, and $k_{Gt} + k_{saf} + k_{smo} = 1$. Φ^* is the final moving direction for the next moment.

However, if the distance between the subgoal $\mathbf{G_t}$ and the robot is less than r_{win}, i.e. $|RG_t| < r_{win}$, and $|RG_t| < dw_t$, then the direction of the point $\mathbf{G_t}$ is directly taken as the moving direction of the robot for the next moment, i.e. let $\Phi^* = \Phi_t$.

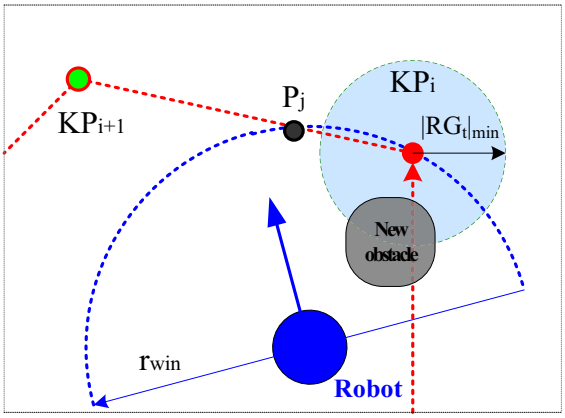
3.4 The Expected Velocity for the Next Moment

According to the moving direction Φ^* and the distance dw_{rob} on the robot's heading θ_R, the motion control of the robot is designed as

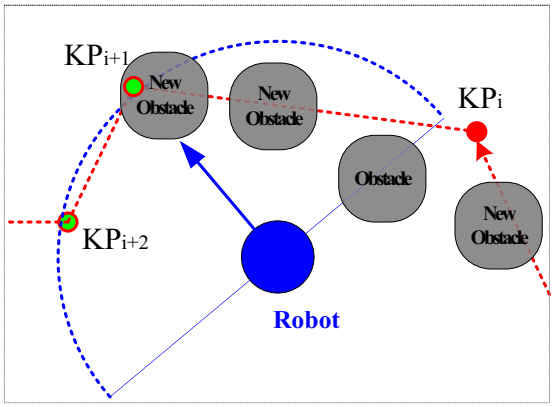
$$v = k_v \cdot dw_{rob} \cdot \cos(\Phi^* - \Phi_{rob}) \tag{16}$$

$$\omega = k_\omega \cdot (\Phi^* - \Phi_{rob}) \tag{17}$$

where v is the expected linear velocity, ω is the expected angular velocity, k_v and k_ω are the positive parameters. And if $\cos(\Phi^* - \Phi_{rob}) < 0$, then let $\cos(\Phi^* - \Phi_{rob}) = 0$. In order to prevent overshoot, the linear velocity is restricted to $v \leq |RG_t|/T$ (when $\mathbf{G_t}$ is located in $\Psi(\Phi^*)$), and the angular velocity is restricted to $\omega \leq \Phi^*/T$.



(a) The subgoal $\mathbf{G_t}$(=$\mathbf{KP}_i$) will be switched to $\mathbf{KP}_{i+1}$



(b) The subgoal $\mathbf{G_t}$(=$\mathbf{KP}_i$) will be switched to $\mathbf{KP}_{i+2}$

Fig. 9 Diagram of the switching of the subgoal

3.5 Adaptive Adjustment of the Planning Window

Let $|\Psi|_t$ as the value of the currentΨ_{path}, which is proportional to the total number of the element i whose distance $dw_i = r_{win}$ in $\{l_1, l_2, ..., l_N\}$, $i = 1,2,...,N$. Considering the weight of direction, the calculation is as follows

$$|\Psi|_t = \sum_{i=1}^{N} |_{l_i \in \Psi_{path}} k_\Psi \cos(|\Phi_i - \Phi_{rob}|) \tag{18}$$

where $k_\Psi > 0$, and if $|\Phi_i - \Phi_{rob}| > 90°$ then let $|\Phi_i - \Phi_{rob}| = 90°$.

According to the value $|\Psi|_t$, the rules for the adaptive adjustment of the planning window are as follow: 1) When $|\Psi|_t$ is large, increase r_{win}, expand the planning window to find better direction Φ^*; 2) When $|\Psi|_t$ is small, reduce r_{win}, decrease the planning window to find the travelable path in narrow space. Let $r_{win}(t)$ be the current value of r_{win}, then the value of the next moment is

$$r_{win}(t+1) = r_{win}(t) + \Delta r_{win} \tag{19}$$

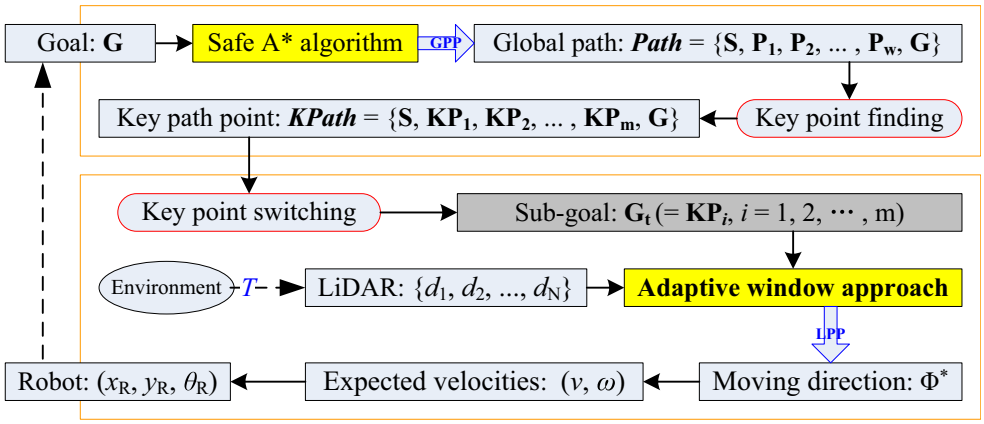


Fig. 10 Overall architecture of the proposed hybrid path planning

The upper and lower bounds are constrained by

$$r_{rob} \cdot < \cdot r_{win}(t+1) \cdot < \cdot d_{\max} \tag{20}$$

And Δr_{win} will be adjusted adaptively according to $|\Psi|_t$. The calculation is as follows:

- when $|\Psi|_t < |\Psi|_1$,

$$\Delta r_{win} \cdot = \cdot \Delta r_{\min} \tag{21}$$

- when $|\Psi|_1 \le |\Psi|_t \le |\Psi|_2$,

$$\Delta r_{win} \cdot = ((\Delta r_{\max} - \Delta r_{\min}) / (|\Psi|_2 - |\Psi|_1))(|\Psi|_t - |\Psi|_2) + \Delta r_{\max} \tag{22}$$

- when $|\Psi|_t > |\Psi|_2$,

$$\Delta r_{win} \cdot = \cdot \Delta r_{\max} \tag{23}$$

where $\Delta r_{\max} > 0$, $\Delta r_{\min} < 0$, the parameters $|\Psi|_2 > |\Psi|_1 > 0$. The relation between Δr_{win} and $|\Psi|_t$ is shown in Fig. 8.

4 Path Following Based on Key Path Points

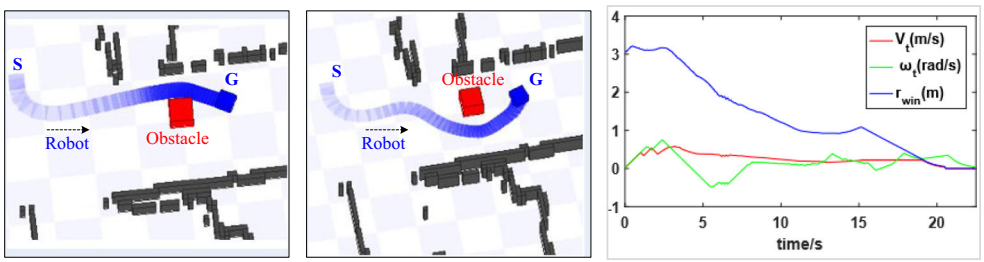
In the navigation of mobile robot, the key path points in ***KPath*** = {**S**, $\mathbf{KP_1}$, $\mathbf{KP_2}$, ..., $\mathbf{KP_m}$, **G**} are taken successively as subgoal $\mathbf{G_t}$ for the real-time motion planning, which can achieve smooth tracking of the global path and obstacles avoidance simultaneously.

However, in order to achieve the coordinated integration of the path tracking and the obstacles avoidance, what's important is that the robot needs to switch the key path points as $\mathbf{G_t}$ according to the current information of environment. The method and steps are as follows.

Step 1: When the robot moving from the starting point **S**, it takes $\mathbf{KP_1}$ as the subgoal, that is, set $\mathbf{G_t} = \mathbf{KP_1}$.

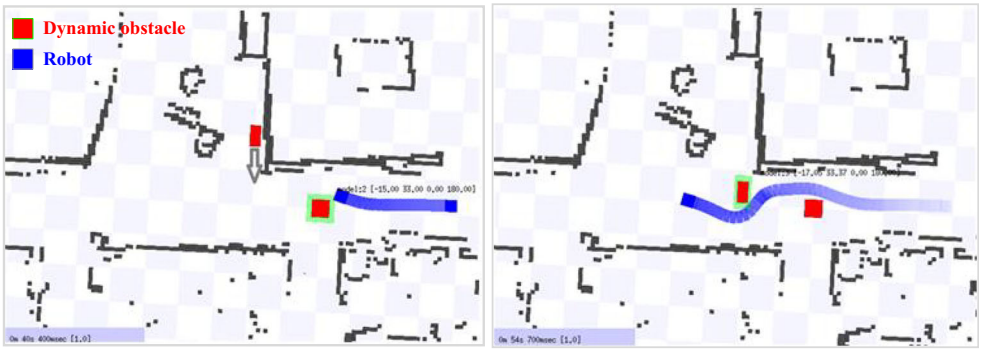
Step 2: When the distance between the robot and the subgoal $\mathbf{G_t}$ (=$\mathbf{KP}_i$) is less than the threshold $|RG_t|_{\min}$, the subgoal will be switched to the next key path point, that is, set $\mathbf{G_t} = \mathbf{KP}_{i+1}$.

Step 3: When $\mathbf{G_t} = \mathbf{KP}_i$ and a path node $\mathbf{P}_j$ on the section ($\mathbf{KP}_i$,..., $\mathbf{P}_j$,..., $\mathbf{KP}_{i+1}$) of ***Path*** appears in Ψ_{path} of the planning window, the subgoal will be switched to the next key path point, that is, set $\mathbf{G_t} = \mathbf{KP}_{i+1}$ (As shown in Fig. 9a).



(a) The track when k_{saf}=0.1 (b)The track when k_{saf}=0.3 (c) The curves of (v, ω) and r_{win}

Fig. 11 Results of motion planning in static environment



(a)Avoiding the First Dynamic Obstacle (b) Avoiding the second Dynamic Obstacle

Fig. 12 Results of motion planning in dynamic environment

Step 4: When $\mathbf{G_t} = \mathbf{KP}_i$ and the key path point $\mathbf{KP}_{i+j}$ ($j > 0$) appears in Ψ_{path} of the planning window, the subgoal will be switched to $\mathbf{G_t} = \mathbf{KP}_{i+j}$. (As shown in Fig. 9b).

Using the above methods, the current subgoal is continuously switched to the following key path points until $\mathbf{G_t} = \mathbf{G}$.

In the subgoal switching, the distance comparison is used to determine whether the path node $\mathbf{P}_j$ (or the key point $\mathbf{KP}_j$) is located in Ψ_{path}. For example, if the distance between the robot $\mathbf{R}$ and the path node $\mathbf{P}_j$ is $|\text{RP}_j|$, the angle between the line RP_j and the heading of the robot is $\angle(\text{RP}_j)$ (the corresponding scanning angle is $\Phi_{\text{RP}j}$), the measured distance corresponding to this direction is $dw_{\text{RP}j}$, then the path node $\mathbf{P}_j$ is located in Ψ_{path} if $|\text{RP}_j| < dw_{\text{RP}j}$. Similarly, the key point $\mathbf{KP}_j$ can be judged using this distance comparison method.

5 The Process of Hybrid Path Planning

Combining the above elements, the process of hybrid path planning is shown in Fig. 10. For the real-time navigation in large-scale dynamic environments, the path planning and path tracking are executed separately. That is, whenever a goal point $\mathbf{G}$ is given, the global path is first planned by the safe A* algorithm and its key points are extracted using key point finding algorithm. Then, the robot uses adaptive window approach (motion planning) with key point switching to track the global path and avoid obstacles.

This is a new HPP formulated with safe A* algorithm and adaptive window approach for global path planning and SPTaOA, and the advantage is that it is a general method for mobile robots i) without the modeling requirement for velocity space computing which needed in the DWA method [31], ii) can be used in dynamic environments in all sizes, even in a large-scale environment (i.e. will not affected by the size of the environment).

6 Experiments and Results

6.1 Simulation and Verification

ROS (Robot operating system) is used for the simulation experiments. The robot is simulated as a block with diameter of 0.5 m. The maximum linear velocity is set to 0.5 m/s and the maximum angular velocity is set to 0.8 rad/s. The motion planning period $T = 50$ ms. $k_v = k_\omega = 1.0$.

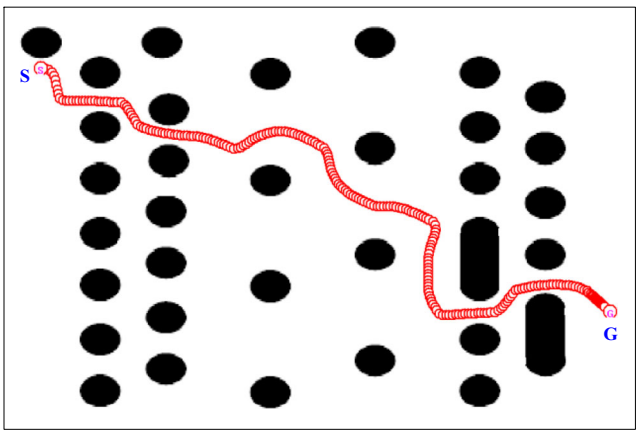


Fig. 13 The result when $r_{\text{win}} = 1.8$ m and remain unchanged

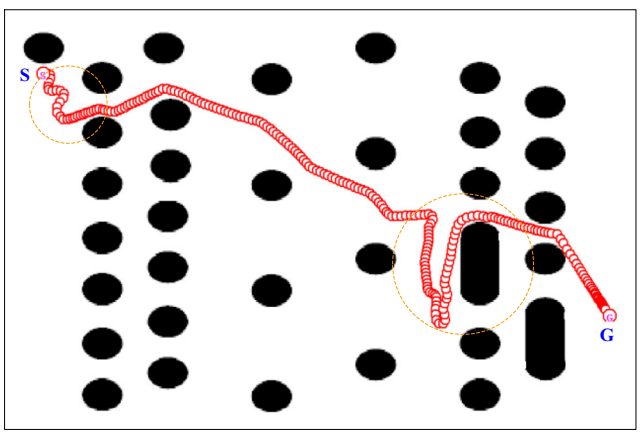


Fig. 14 The result when $r_{\text{win}} = 7$ m and remain unchanged

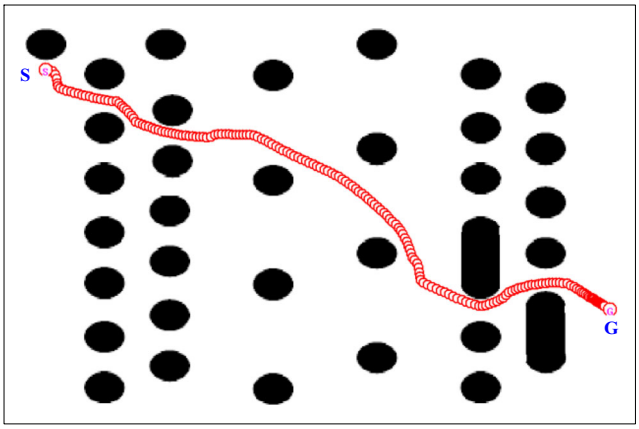
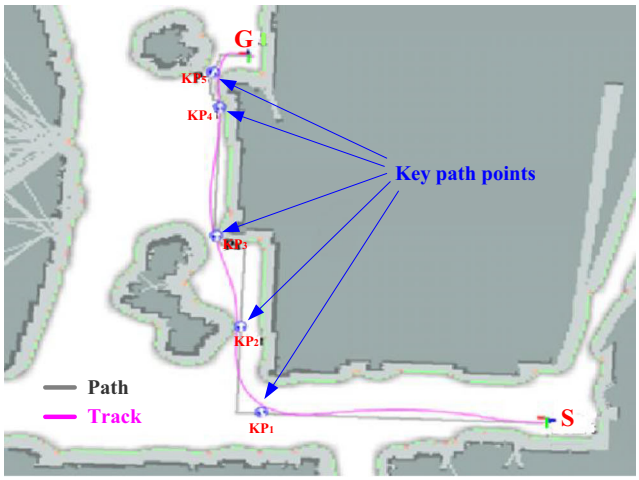


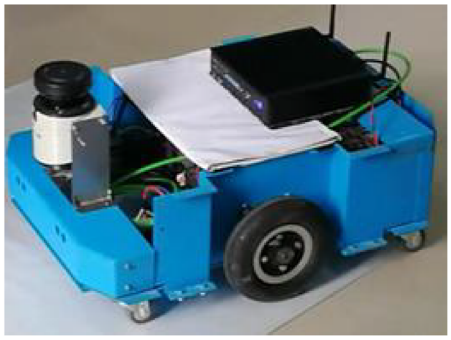
Fig. 15 The result when r_{win} is adjusted adaptively



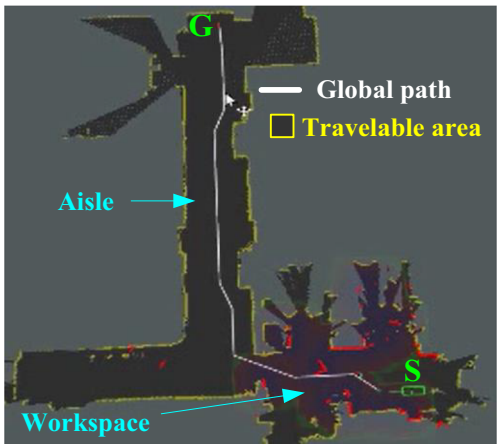
(a) The environment has not changed

(b) The environment has changed (has new obstacles)

Fig. 16 Global path planning and tracking



(a) AGV robot



(b) Grid map and global path

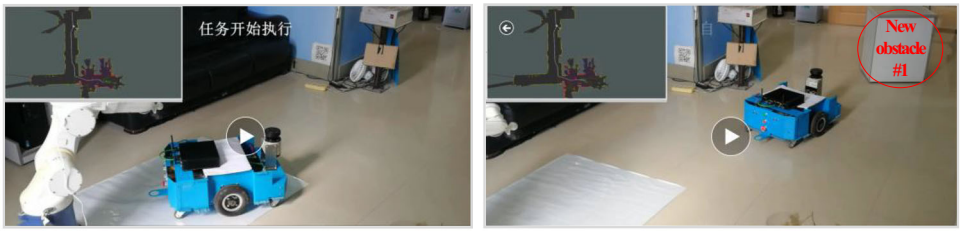
Fig. 17 AGV robot and planned global path

6.1.1 Verification of the Motion Planning

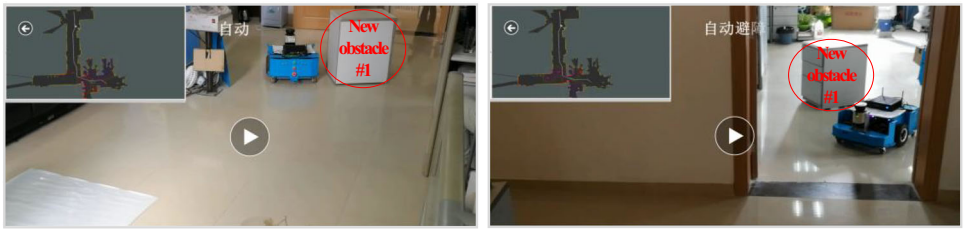
In order to verify the performance of the motion planning based on adaptive window approach, the simulation results of the robot in unknown static environment are given as shown in Fig. 11. The red block represents the dynamic obstacle, and the blue represents the track of the robot from the point **S** to point **G** result from the real-time motion planning. Where, Fig. 11a is the track when ($k_{\text{Gt}} = 0.6$, $k_{\text{smo}} = 0.3$, $k_{\text{saf}} = 0.1$), Fig. 11b is the track when ($k_{\text{Gt}} = 0.5$, $k_{\text{smo}} = 0.2$, $k_{\text{saf}} = 0.3$). It can be seen that the track (with bigger parameter k_{saf}) has a certain distance from obstacles, which is of benefit to safety. Figure 11c is the curves of the velocities (v, ω) of the robot and the radius of the planning window.

As shown in Fig. 12, on the way to the goal, the robot encounters two dynamic obstacles successively. The experimental results show that when the obstacle suddenly appear in front of the robot, the robot can change its heading quickly, and safely avoid the moving obstacle. So the motion planning method has good adaptability for dynamic environment.

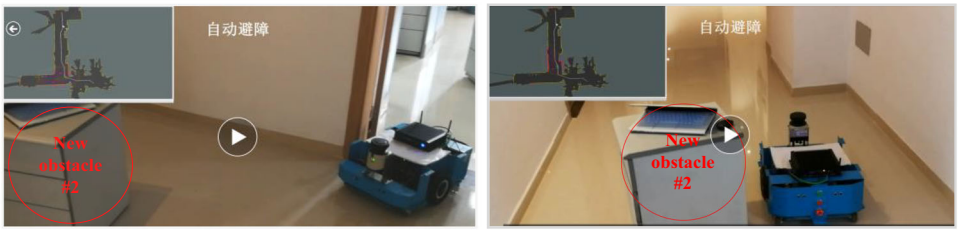
Furthermore, using the motion planning, if the planning window is too small, as shown in Fig. 13, the robot can not avoid obstacles in time; if the planning window is too large, as shown in Fig. 14, the robot may not find the optional moving



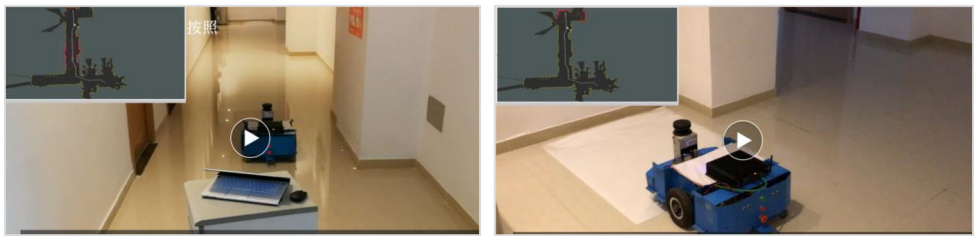
(a)Starts from the starting point (b) Encountering a new obstacle in path tracking



(c) Avoiding the obstacle (d) Returns to the path after bypassing the obstacle



e) Encountering another new obstacle at the door (f)Avoiding the obstacle



(g) Moving along the path after bypassing the obstacle (h)Arrived at the goal point smoothly

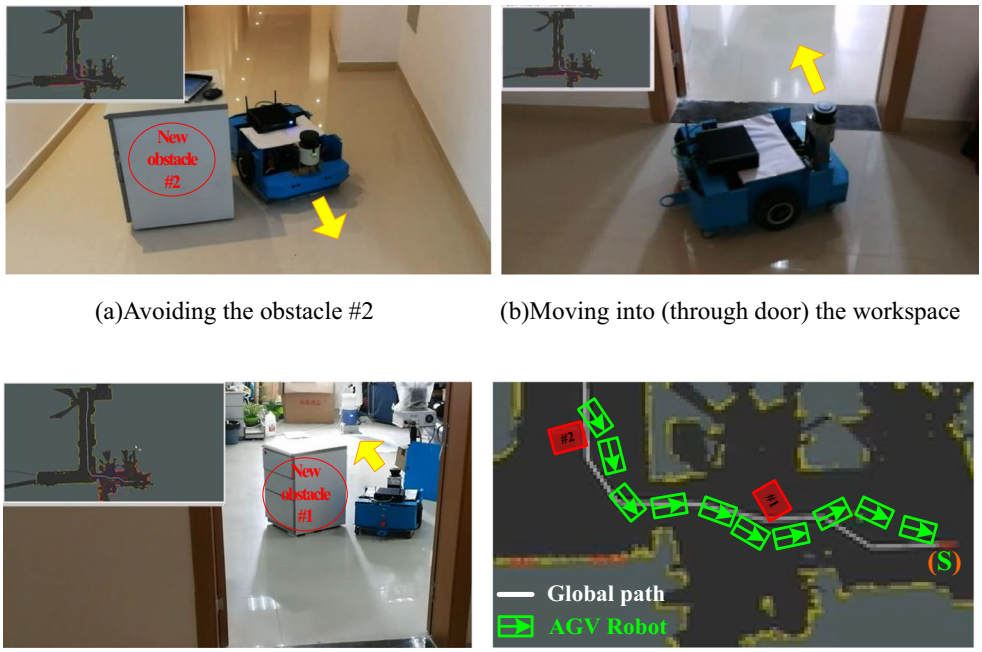
Fig. 18 The snapshots of the AGV robot video of path tracking

direction (only when walking along a wall). When using the adaptive window approach, the robot can dynamically adjust the size of the planning window according to the environmental information, the resulting track is shown in Fig. 15. As can be seen, the track in Fig. 15 is significantly better than that in Fig. 13 and Fig. 14.

6.1.2 Verification of Global Path Tracking and Obstacle Avoidance

The result of global path planning in static environment is shown in Fig. 16a. It can be seen that the path planned by the safe A* is not the shortest one, but it keeps a certain safe distance from the obstacles. Then, using the real-time motion planning and the switching of the key path point for subgoal, the robot tracks the global path smoothly and finally arrives at the goal point **G** without collision. The results show that the methods of motion planning and subgoal switching can effectively achieve simultaneous path tracking and obstacles avoidance (SPTaOA) in static environment.

To further verify the impact of new obstacles on path tracking, different obstacles are added around the key points in the scenario of Fig. 16b (map and global path are the same as Fig. 16a. The final result of path tracking is shown in Fig. 16b. It can be seen that no matter the new obstacle is in front, behind or even completely covering the key point, the



(a)Avoiding the obstacle #2 (b)Moving into (through door) the workspace

(c)Avoiding the obstacle #1 (d)The track of AGV robot

Fig. 19 The snapshots of the video of path tracking and track of the AGV robot

robot can always avoid the obstacles in time and switch to the next key point as subgoal, and finally returns to the global path for the goal **G**. Therefore, the results show that the adaptive window approach and the key point switching can effectively achieve simultaneous path tracking and obstacles avoidance (SPTaOA) in dynamic environment.

6.2 The Tests of Practical Application

As shown in Fig. 17, the application tests were carried out on a autonomous AGV robot (mobile industrial robot) using LiDAR-based SLAM. Figure 17a is a picture of the AGV robot (drived by brushless DC motors, MAX 3000RPM) using LMS111 LiDAR and on-board Industrial Computer with i7–4500 CPU and 4GB RAM; The software is desiged based on ROS. Figure 17b is the grid map of the working environment and the global path obtained by using the safe A* algorithm. The maximum velocity of the AGV's driving wheel is able to reach 1.0 m/s. And in the tests, we set the motion planning period as $T = 80$ ms, and $k_v = k_\omega = 1.0$.

With regard to the planned global path of Fig. 17b, the process of real-time path tracking and obstacle avoidance are shown in Fig. 18 (snapshots (a)-(h)). It can be seen that the AGV robot, using the motion planning, starts from the starting point **S**, guided by the key path points, can immediately avoid the emerging obstacles, and then return to the planned path to reach the goal point **G** smoothly.

When AGV arrived at the goal point **G**, we set the starting point **S** as new goal point. The process of path tracking starting from the avoidance of obstacle #2 are shown in Fig. 19. Where Fig. 19d is the track of AGV robot to show the path transverse by the robot to reach the goal successfully. It can be seen that the AGV robot guided by the key path points of global path, can avoid the new obstacles #2 and #1 again, then reach the goal point (the original **S**) smoothly.

Statistical analyses of the tests reveal that the motion planning algorithms (i.e. LPP) is executed in less than 12 ms on the on-board Industrial Computer (i7–4500 CPU, 4GB RAM), and the LPP algorithms with key point switching is executed in less than 17 ms. However, if the robot rurnning A* algorithm in every motion planning period for GPP, the time consume is more than 20-50 ms, and it will exceeds the preset period T with the size increase of environment.

7 Conclusion

In this paper, the global path planning, tracking and obstacle avoidance for mobile robot in large-scale dynamic environments are studied. In order to meet the requirement of real-time navigation, a hybrid planning method is proposed, which uses the safe A* algorithm for global path planning and motion planning for path tracking and obstacle avoidance. More specifically, the designed safe A* algorithm simplifies the cost function which has risk evaluation. The real-time motion planning method based on adaptive window is able to achieve smooth and collision-free global path tracking through the proposed strategy of the extraction and switching of key path points as the subgoal. Experimental results show that this hybrid path planning method combines effective A* and real-

time motion planning, which can meet the practical application requirement of mobile robots in large-scale dynamic environments.

Although GPP and LPP has been well studied separately, this is the first HPP formulated with safe A* algorithm and adaptive window approach for global path planning and SPTaOA, which without the modeling requirement for velocity space computing and can be used in dynamic environments free to the size of the scenes. However, as many parameters are determined by experiences and simulation experiments, our future work will be focused on how to optimize them effectively.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China (NO. 61703356, 61305117), Industry-University Cooperation Project (IUCP) of Fujian Province (NO. 2017H6021), Fundamental Research Funds for the Central Universities(NO. 20720190129).

References

1. Lim, Z.W., Hsu, D., Lee, W.S.: Adaptive informative path planning in metric spaces[J]. Int. J. Robot. Res. **35**, 283–300 (2015)
2. Jie, J., Khajepour, A., Melek, W.W., et al.: Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints[J]. IEEE Trans. Veh. Technol. **66**(2), 952–964 (2017)
3. Bhattacharya, S., Ghrist, R., Kumar, V.: Persistent homology for path planning in uncertain environments[J]. IEEE Trans. Robot. **31**(3), 578–590 (2017)
4. Murillo, M., Sánchez, G., Genzelis, L., et al.: A real-time path-planning algorithm based on receding horizon techniques[J]. J. Intell. Robot. Syst. **91**, 445–457 (2018)
5. Low, E.S., Ong, P., Cheah, K.C.: Solving the optimal path planning of a mobile robot using improved Q-learning [J]. Robot. Auton. Syst. **115**, 143–161 (2019)
6. Mac, T.T., Copot, C., Tran, D.T., et al.: Heuristic approaches in robot path planning: a survey[J]. Robot. Auton. Syst. **86**, 13–28 (2016)
7. Kala, R., Shukla, A., Tiwari, R.: Fusion of probabilistic a* algorithm and fuzzy inference system for robotic path planning[J]. Artif. Intell. Rev. **33**(4), 307–327 (2010)
8. Persson, S.M., Sharf, I.: Sampling-based a* algorithm for robot path-planning[J]. Int. J. Robot. Res. **33**(13), 1683–1708 (2014)
9. Chen, W., Zhang, T., Zou, Y.: Mobile robot path planning based on social interaction space in social environment[J]. Int. J. Adv. Robot. Syst. **15**(3), 1–10 (2018)
10. Zhang, H.M., Li, M.L., Yang, L.: Safe path planning of Mobile robot based on improved a* algorithm in complex terrains[J]. Algorithms. **11**(4), 44–62 (2018)
11. Bayili, S., Polat, F.: Limited-damage a*: a path search algorithm that considers damage as a feasibility criterion[J]. Knowl.-Based Syst. **24**(4), 501–512 (2011)
12. Park, J.H, No, J.H., Huh, U.Y.: Safe global path planning of Mobile robots based on modified A* algorithm[M]// the 8th international conference on robotic, vision, Signal Processing & Power Applications, pp. 99–105. Springer Singapore (2014)
13. Aine, S., Swaminathan, S., Narayanan, V., et al.: Multi-Heuristic A*[J]. Int. J. Robot. Res. **35**(1–3), 224–243 (2016)
14. Likhachev, M., Koenig, S.: A generalized framework for lifelong planning a* search[C]// fifteenth international conference on international conference on automated planning and scheduling, pp. 99–108. AAAI Press (2008)
15. Likhachev, M., Ferguson, D., Gordon, G.: Anytime search in dynamic graphs[J]. Artif. Intell. **172**(14), 1613–1643 (2008)
16. Toll, W.V., Geraerts, R.: Dynamically pruned A* for re-planning in navigation meshes[C]// IEEE/RSJ international conference on Intelligent Robots & Systems. IEEE (2015)
17. Dakulovi, M., Petrovi, et al.: Two-way D* algorithm for path planning and replanning[J]. Robot. Auton. Syst. **59**(5), 329–342 (2011)
18. Ammar, A., Bennaceur, H., Châari, I., et al.: Relaxed Dijkstra and a* with linear complexity for robot path planning problems in large-scale grid environments.[J]. Soft. Comput. **20**(10), 4149–4171 (2016)
19. Montiel, O., Sepúlveda, R., Orozco-Rosas, U.: Optimal path planning generation for Mobile robots using parallel evolutionary artificial potential field[J]. J. Intell. Robot. Syst. **79**(2), 237–257 (2015)
20. Moon, C.B., Chung, W.: Kinodynamic planner dual-tree RRT (DT-RRT) for two-wheeled Mobile robots using the rapidly exploring random tree[J]. IEEE Trans. Ind. Electron. **62**(2), 1080–1090 (2015)
21. Zaid, T., Qureshi, A.H., Yasar, A., et al.: Potentially guided bidirectionalized RRT* for fast optimal path planning in cluttered environments[J]. Robot. Auton. Syst. **108**, 13–27 (2018)
22. Taheri, E., Ferdowsi, M.H., Danesh, M.: Fuzzy Greedy RRT Path Planning Algorithm in a Complex Configuration Space[J]. Int. J. Control. Autom. Syst. **16**(6), 3026–3035 (2018)
23. Aykut, Z., Volkan, S.: Follow the gap with dynamic window approach[J]. Int. J. Semant. Comput. **12**(01), 43–57 (2018)
24. Fox, D., Burgard, W., Thrun, S.: The dynamic window approach to collision avoidance[J]. IEEE Rob. Auto. Mag. **4**(1), 23–33 (2002)
25. Seder, M., Petrovic, I.: Dynamic window based approach to mobile robot motion control in the presence of moving obstacles[C]// IEEE international conference on robotics and automation, pp. 1986–1991. IEEE (2007)
26. Li, G.Y., Wu, Y.Y., Wei, W.: Guided dynamic window approach to collision avoidance in troublesome scenarios [C].China: Proceedings of the 7th World Congress on Intelligent Control and Automation (2008)
27. Zhong X., Peng X., Miao M.: Planning of robot paths through environment modelling and adaptive window [J]. J. Huazhong Univ. of Sci. and Tech. (Natural Science Edition). **38**(6), 107–111 (2010)
28. Chen, Y., Wang, X., Hong, S., et al.: Motion planning implemented in ROS for mobile robot[C]// Control and decision conference, pp. 7149–7154. IEEE (2017)
29. Lu, M.C., Hsu, C.C., Chen, Y.J., et al.: Hybrid path planning incorporating global and local search for Mobile robot[J]. IEEE. **7429**, 668–671 (2012)
30. Imran, M., Kunwar, F.: A Hybrid Path Planning Technique Developed by Integrating Global and Local Path Planner[C]// 2016 International Conference on Intelligent Systems Engineering (ICISE). IEEE (2016)
31. Cheng, C., Xiaoyang, H., Jiansheng, L., et al.: Global Dynamic Path Planning Based on Fusion of Improved A* Algorithm and Dynamic Window Approach [J]. J. Xi'an Jiaotong Univ. **51**(11), 137–143 (2017)

Xunyu Zhong received the Ph.D. degree in control theory and control engineering from Harbin Engineering University, China, in 2009. He is currently an associate Professor with the Department of Automation, Xiamen University, China. His current research interests include robot path/motion planning, SLAM, visual servo and intelligent autonomous mobile robot.

Jun Tian received the M.Sc. degree in Marine Engineering from Jimei University, Xiamen, China, in 2008. He is currently working toward the Ph.D. degree in Xiamen University. His research mainly focuses on motion planning and SLAM of mobile robot.

Huosheng Hu (SM'01) received the M.Sc. degree in industrial automation from Central South University, Changsha, China, in 1982, and the Ph.D. degree in robotics from the University of Oxford, Oxford, U.K., in 1993. Currently, he is a Professor with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K., leading the Robotics Research Group. And he is a Guest Professor in Xiamen University. He has authored over 500 research articles published in journals, books, and conference proceedings. His research interests include autonomous mobile robots, human–robot interaction, multi-robot collaboration, pervasive computing, sensor integration, intelligent control, cognitive robotics, and networked robots. Prof. Hu is a Fellow of IET & InstMC, and a Chartered Engineer in U.K. He currently serves as Editor-in-Chief for the Int. Journal of Automation and Computing, Editor-in-Chief of MDPI Robotics Journal, and an Executive Editor for the Int. Journal of Mechatronics and Automation.

Xiafu Peng received the M.Sc. and Ph.D. degrees in control science from the Harbin Engineering University, in 1994 and 2001, respectively. He is currently a Professor with the Department of Automation, Xiamen University at Xiamen. His current research interests include the navigation and motion control of robots. Prof. Peng is a Fellow of the Fujian Association for the advancement of Automation and Power, and a Senior Member of the Chinese Institute of Electronics. He is the recipient of the provincial/ministerial Scientific and Technological Progress Award.