



Article

Perception-Aware Planning for Active SLAM in Dynamic Environments

Yao Zhao ¹, Zhi Xiong ^{1,*}, Shuailin Zhou ¹, Jingqi Wang ¹, Ling Zhang ¹ and Pascual Campoy ²

¹ College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; yaozhaonc29@nuaa.edu.cn (Y.Z.); zhou03@nuaa.edu.cn (S.Z.); wangjingqi@nuaa.edu.cn (J.W.); zhanglingsnowman@nuaa.edu.cn (L.Z.)

² Computer Vision and Aerial Robotics Group, Universidad Politécnica de Madrid, 28006 Madrid, Spain; pascual.campoy@upm.es

* Correspondence: xiongzhi@nuaa.edu.cn

Abstract: This paper presents a perception-aware path planner for active SLAM in dynamic environments using micro-aerial vehicles (MAV). The “Next-Best-View” planner (NBVP planner) is combined with an active loop closing, which is called the Active Loop Closing Planner (ALCP planner). The planner is proposed to avoid both static and dynamic obstacles in unknown environments while reducing the uncertainty of the SLAM system and further improving the accuracy of localization. First, the receding horizon strategy is adopted to find the next waypoint. The cost function that combines the exploration gain and the loop closing gain is designed. The former can reduce the mapping uncertainty, while the latter takes the loop closing possibility into consideration. Second, a key waypoint selection strategy is designed. The selected key waypoints, instead of all waypoints, are treated as potential loop-closing points to make the algorithm more efficient. Moreover, a fuzzy RRT-based dynamic obstacle avoidance algorithm is adopted to realize obstacle avoidance in dynamic environments. Simulations in different challenging scenarios are conducted to verify the effectiveness of the proposed algorithm.



Citation: Zhao, Y.; Xiong, Z.; Zhou, S.; Wang, J.; Zhang, L.; Campoy, P. Perception-Aware Planning for Active SLAM in Dynamic Environments. *Remote Sens.* **2022**, *14*, 2584. <https://doi.org/10.3390/rs14112584>

Academic Editor: Francisco Javier Mesas Carrascosa

Received: 10 May 2022
Accepted: 24 May 2022
Published: 27 May 2022

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



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: perception-aware; key waypoints; path planning; active slam; loop closing; dynamic environments

1. Introduction

A Micro-Aerial Vehicle (MAV) is an intelligent vehicle that integrates MEMS technology, micro-electronics, computer technology, intelligent control and other technologies. It is characterized by low cost, small size, ease of operation and flexibility and is capable of performing various tasks such as reconnaissance [1], inspection [2], surveying and mapping [3–5] and search and rescue [6–8] in complex flight environments such as low altitude, indoor, marine environments and urban complexes. In recent years, simultaneous localization and mapping (SLAM) is one of the hot topics in the field of mobile robotics, and much attention has been paid to this research area. There are many classical SLAM frameworks and outstanding algorithms being used for autonomous navigation and localization [9–12]. SLAM focuses mainly on localization and map building rather than trajectory planning. When SLAM and trajectory planning are considered together, the problem becomes more complex and challenging, which is called active SLAM [13]. In active SLAM, a robot, such as an MAV or UGV [14–16], needs to solve a decision-making problems, in which a collision-free trajectory is planned to improve position estimation accuracy, as well as to perform other tasks, such as unknown environment exploration [17] and coverage planning [18]. This problem involves both SLAM and path-planning parts. The accuracy of localization must be taken into consideration during the planning process. Therefore, it is regarded as one of the most challenging problems in mobile robotics.

In this work, we focus more on active SLAM used in MAV. The purpose and motivation of this work is to plan a good trajectory to improve the localization performance of the

SLAM system, while performing a search and rescue task in unknown complex dynamic environments. In SLAM systems, loop closure detection is beneficial for improving localization accuracy. Hence, we propose a perception-aware path planning method in dynamic environments using MAV. The next best view planner is combined with an active loop closing strategy to make it an active SLAM system. This method increases the number of loop closures in the planned path as much as possible, while taking into account the efficiency of the search and rescue. In addition, a dynamic obstacle avoidance algorithm is introduced in this work to avoid various obstacles that may appear in dynamic environments.

The main contributions of this work are as follows:

- (1) A perception-aware path planning method is proposed, which combines the next best view planner with an active loop closing strategy to perform a trajectory that enables more accurate localization while performing search and rescue tasks;
- (2) A key-waypoint-based active loop closing strategy is proposed to improve the efficiency of active SLAM system;
- (3) Fuzzy RRT-based dynamic obstacle avoidance is integrated into the path planning method to improve the system performance in dealing with complex dynamic environments.

The rest of this paper is structured as follows. Section 2 provides an overview of related work in path planning and active SLAM. Then, Section 3 presents the system overview and explains the proposed algorithm in detail, including the next best view planner, the key-waypoints-based active loop closing, and a fuzzy-RRT-based dynamic obstacle avoidance algorithm. Subsequently, Section 4 provides qualitative and quantitative results of performance of the proposed method both on different cluster dynamic environments to demonstrate and analyze the effectiveness of the perception-aware path planning method and the active SLAM system. Moreover, discussion on the performance of different algorithms in different scenarios is presented in Section 5. Finally, conclusion and future work are discussed in Section 6.

2. Related Work

In the mobile robotics area, SLAM and path planning are two hot spot topics in the research community. Nowadays, with the development of SLAM algorithms becoming more sophisticated, more and more attention has been paid in the combination of path planning and SLAM [19–21]. Considering both of these makes the problem more complex and more challenging. In the field of exploration and coverage planning, a number of path planning algorithms have achieved very promising results, including the typical-sampling-based Receding horizon “Next Best View” planner [22], the frontier-based exploration [23] and more recent variants [17,19,24–32] based on them. For example, in [24], a novel Next-Best-View (NBV) planner which can perform full exploration and user-oriented exploration with inspection of the regions of interest using a mobile manipulator robot is proposed. In [30], a novel exploration method is proposed to generate a complete surface model in an unknown environment.

Moreover, to reduce the uncertainty of exploration and improve the accuracy of localization while performing path planning, active SLAM algorithms for autonomous, uncertainty-aware exploration, localization and mapping of unknown environments are proposed in [17,19,20,33,34]. In [17], a novel path planning algorithm for the autonomous, uncertainty-aware exploration and mapping of unknown environments using aerial robots is proposed to be able to efficiently explore unknown environments. In [19], an uncertainty-aware path planning strategy is proposed to achieve the autonomous aerial robotic exploration of unknown environments while ensuring mapping consistency on-the-go. In [20], the paramount importance of representing and quantifying uncertainty to the associated confidence of the robot’s location estimate is fully discussed. In [33], a novel strategy is proposed for the autonomous visual saliency-aware receding horizon exploration of unknown environments using aerial robots. In [34], a novel exploration strategy for MAVs is proposed to reduce map entropy regarding occupancy probabilities. Meanwhile, the Active SLAM algorithm is also important when it comes to coverage planning [18,35–37].

In [35], a hierarchical, hex-decomposition-based coverage planning algorithm is proposed for unknown obstacle-cluttered environments. In [35], a coverage path planning algorithm is proposed to find a collision-free path while covering every accessible region within an environment. In [36], an online coverage and inspection planning for 3D modeling is proposed to explore an unknown environment using an MAV. In addition, obstacle avoidance is seen as an integral part of path planning in both static and dynamic environments [28,38–41]. Specifically, a real-time perception-aware trajectory planner in dynamic environments is proposed in [38], which plans trajectories that avoid dynamic obstacles while also keeping them in the sensor field of view and minimizing the blur to aid in object tracking. A novel depth-based collision-avoidance method for aerial robots is proposed in [39] to ensure safety without sacrificing speed. A 3-D decentralized and asynchronous trajectory planner for UAVs that generates collision-free trajectories in environments with static obstacles, dynamic obstacles, and other planning agents is proposed in [40]. An environmental adaptive planner is proposed in [41] to adjust the flight aggressiveness. A robust and perception-aware replanning framework is proposed in [28] to support fast and safe flights.

The methods above enable systems to adapt to complex dynamic environments. The contributions proposed in this work differ from the above in the sense that we propose an improved receding horizon next best view planner to combine trajectory planning of unknown environments with key-waypoints-based active loop closing, together with dynamic obstacle avoidance algorithm to enhance the on-the-go mapping behavior of the MAV in a framework that remains computationally efficient. In this way, a perception-aware path planning based active SLAM is established and will be introduced in detail in the next section.

3. Method

3.1. System Overview

The framework of the system can be seen in Figure 1. First, the receding horizon strategy is adopted to perform path planning. Different to [22], in this work, the feature selection cost is considered to add to the cost function to calculate gain. In this way, the best waypoint is selected as the next waypoint among the candidates. Second, after the determination of waypoint, a key waypoint selection strategy is designed and a key-waypoints-based active loop closing planning (ALCP) method is proposed. In this method, key waypoints instead of all waypoints are selected to perform loop closing, which balances the efficiency of unknown space exploration with the accuracy of localization. Moreover, a dynamic obstacle avoidance algorithm is adopted to realize obstacle avoidance in dynamic environments.

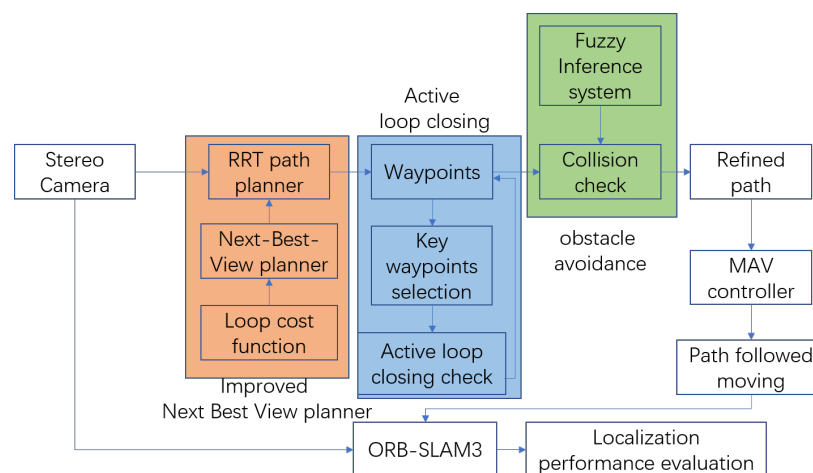


Figure 1. Framework of system.

3.2. Active Loop Closing Planner

Considering the exploration of unknown space, the proposed algorithm employs a sampling-based receding horizon path planning strategy and an active loop closing strategy to generate paths that cover the bounded space V , which is called the Active Loop Closing Planner (ALCP) in this work. A stereo camera is adopted as a sensing system to sense the surrounding environment and provide feedback to the MAV controller. All acquired information is included in a map representing the environment. A fuzzy RRT algorithm is adopted to generate an obstacle-free path. The details can be seen in Section 3.4. The map is adopted for the navigation of an MAV and deciding the direction of exploration.

The employed representation of the environment is an occupancy map dividing space V in occupied grids that can either be marked as free, occupied or unmapped. The resulting array of voxels is saved in an octree structure, enabling computationally efficient access such as checking for occupancy and checking for loop closing. Generally, paths are only planned in the known free space, thus providing collision-free navigation. However, when the previously passed waypoint is very close to the current MAV state, the MAV is planned to pass the previous waypoint again to form a loop, which is good for the performance of localization in the SLAM system. While a collision-free MAV state is denoted by S , a path is denoted by ϵ . The corresponding path cost from $k - 1$ to k is $c(\epsilon_k^{k-1})$. For a given occupancy map representing the world W , the set of visible and unmapped voxels from state S is denoted as $Vis(S)$. From the current state of the MAV, a geometric tree T is incrementally constructed in the space using the RRT algorithm. The resulting tree contains N_T nodes n , and its edges are given by collision-free paths ϵ . The information gain of a node $Gain(n)$ is the summation of the unmapped volume that can be explored at the nodes, as well as the feature observability. For node k , the corresponding $Gain(n_k)$ can be calculated by Equation (1):

$$Gain(n_k) = Gain(n_{k-1}) + \omega_1 Vis(S_k) e^{-\lambda c(\epsilon_k^{k-1})} + \omega_2 Loop(S_k) e^{-\lambda} \quad (1)$$

where λ is a tuning factor which penalizes high path costs [22,42], and ω_1 and ω_2 are weighting factors. $Loop(S_k)$ means the distance to the existing waypoints in the neighborhood around S_k . A trade-off between loop closing possibility and efficiency of exploration is made in the gain calculation above. After every replanning, the first segment of the branch to the best node is executed by MAV, and n_{best} is the node with highest gain. Each executed node is numbered, and the information of the node will be saved in the map. When the MAV is close to the saved node, it will be planned to pass the node again to form a loop. After one loop is formed, a time interval is set to prevent excessive re-exploration of known maps.

3.3. Key-Waypoints-Based Active Loop Closing

In SLAM systems, the repeated passage of the motion carrier through the same place will facilitate its own localization accuracy, which is called loop closing. In order to improve the accuracy of localization efficiently while exploring the space, a key-waypoints-based active loop closing method (Algorithm 1) is proposed in this section. As mentioned above in Section 3.2, each best node is determined to be the next waypoint. Each waypoint has the potential to be the loop closing point when the MAV passes again in the vicinity of the waypoint. However, considering the efficiency of the system, instead of all the waypoints, the selected waypoints, which are called key waypoints, are treated as potential loop-closing points. The criteria for the selection of key waypoints are as follows: (1) the change in yaw at the current waypoint is greater than 45 degrees from the yaw at the previous waypoint, and (2) five waypoints have been passed after the last key waypoint was selected. If one of the two criteria is satisfied, the current waypoint will be selected as the key waypoint. When the distance of the MAV from a key waypoint is less than a certain

threshold, the system will initiate the active loop closing process. The distance between the current position and key waypoint can be expressed by the following Equation (2):

$$d = \|p_{current} - p_{waypoint_i}\| \quad (2)$$

where d denotes the distance, $p_{current}$ denotes the current position and $p_{waypoint_i}$ denotes the position of the i th waypoint. To ensure the efficiency of the path planning, the active loop closing will be checked under the following situations: (1) after a certain time interval of planning, the threshold of interval is denoted by Th , and (2) after reaching the maximum value of iterations, if the iterative step cannot find a proper node with best gain. With the strategy above, the probability of loop closure in the path planning process can be increased while considering the planning efficiency.

Algorithm 1: Exploration Planner—Active Loop Closing

Input: Path planning times $Count_p$; Best nodes n_{best}
Output: Enable active loop closing flag $bFlag_{lc}$; Planned path ϵ

- 1 select key waypoints according to n_{best}
- 2 save key waypoints to wp_{key} as candidate active loop closing point
- 3 **If** $wp_{key} \neq \emptyset$ **then**
- 4 $bFlag_{lc} = true$
- 5 **end if**
- 6 **If** $Count_p = Th$ **and** $bFlag_{lc} = true$ **then**
- 7 $\epsilon = GetActiveLoopPath$
- 8 $Count_p = 0$
- 9 **break**
- 10 **end if**

3.4. Fuzzy-RRT-Based Dynamic Obstacle Avoidance

To guarantee collision avoidance in dynamic environments, a fuzzy RRT based planner is proposed to deal with dynamic obstacle avoidance (Algorithm 2). Fuzzy rules are designed, and a fuzzy inference system is adopted to output the steering response, which will be explained in detail in this section.

Algorithm 2: Fuzzy-RRT-based dynamic obstacle avoidance

Input: next best view planner; Obstacle distance threshold Th_{od}
Output: Refined path by fuzzy-RRT ϵ_{f-rrt}

- 1 generate path ϵ_{rrt} by *nextbestviewplanner* (Algorithm 3)
- 2 follow the planned path
- 3 calculate the distance from obstacle d_o in the path
- 4 **If** $d_o < Th_{od}$ **then**
- 5 generate refined path ϵ_{f-rrt} by fuzzy inference
- 6 **else**
- 7 $\epsilon_{f-rrt} = \epsilon_{rrt}$
- 8 **end if**
- 9 **return** ϵ_{f-rrt}

Fuzzy control is based on fuzzy theory, fuzzy linguistic variables and fuzzy logic inference [43]. It is essentially an intelligent control method that can be used to mimic human fuzzy inference and decision making processes in terms of behaviour. The process of implementing fuzzy control is as follows: (1) fuzzification of input and output variables to determine the values of linguistic and language variables; (2) determine the fuzzy membership function for each linguistic variable value; (3) establish fuzzy rules and perform fuzzy inference based on the rules; (4) defuzzification, where the amount of fuzziness is made precise by certain defuzzification methods. Fuzzification, fuzzy membership func-

tions and fuzzy rules will be described in detail in this section. In this work, the Mamdani inference method is used to perform fuzzy inference, and the centroid method is adopted as the defuzzification method. After defuzzification, the output is obtained for refining the current path. The fuzzy surfaces which represent the relationship between input and output are shown in Figures 2–4.

The fuzzy-RRT-based method presented in this work is based on RRT for global path planning and fuzzy inference rules for obstacle avoidance. RRT is a sampling-based method that grows a finite iteration random tree to explore the entire environment and incrementally find a connected path from the start to the end (Algorithm 3, line 7). However, in complex dynamic scenarios, due to the existence of dynamic obstacles, it is hard to find an obstacle-free path to connect to the tree. The planned path generated by the RRT algorithm is not always the optimum path with shortest distance from the current to the next best view waypoint. This is due to the fact that the generated random tree may go farther around in order to avoid obstacles during the replanning process, which will have a detrimental effect on exploration efficiency.

In the proposed fuzzy-RRT-based method, fuzzy inference is combined with traditional RRT for dynamic obstacle avoidance and to generate a path to the next-best-view waypoint efficiently at the same time. Specifically, the RRT algorithm is executed to perform global path planning at the first step, and then fuzzy inference rules are utilized to perform trajectory refinement at the second step when the obstacles are detected during flight following the planned path. Finally, an optimized path with several next best view waypoints is generated for MAV exploration.

Algorithm 3: Exploration Planner—Iterative Step

Input: Initial state S_0 ; Enable active loop-closing flag $bFlag_{lc}$
Output: Best gain: $Gain_{best}$; Planned path ϵ ; Path planning times $Count_p$

- 1 Initialize path planning times $Count_p$
- 2 Initialize T with S_0
- 3 Initialize best gain: $Gain_{best} = 0$
- 4 Initial root as best node: $n_{best} = n_0(S_0)$
- 5 Number of nodes in T : N_T
- 6 Initialize loop time: $t_{loop} = 0$
- 7 Waiting for planner call
- 8 **while** $N_T < N_{max}$ **or** $Gain_{best} = 0$ **do**
- 9 Add new node n_{new} to build T incrementally
- 10 $N_T = N_T + 1$
- 11 $t_{loop} = t_{loop} + 1$
- 12 **if** $t_{loop} > t_{max}$ **then**
- 13 **if** $bFlag_{lc}$ **then**
- 14 $\epsilon = GetActiveLoopPath$
- 15 **else**
- 16 $\epsilon = BackToPreviousPath$
- 17 **end if**
- 18 **break**
- 19 **end if**
- 20 **if** $Gain(n_{new}) > Gain_{best}$ **then**
- 21 $n_{best} = n_{new}$
- 22 $Gain_{best} = Gain(n_{new})$
- 23 **break**
- 24 **end if**
- 25 **end while**
- 26 $\epsilon = GetBestPath(n_{best})$
- 27 $Count_p = Count_p + 1$

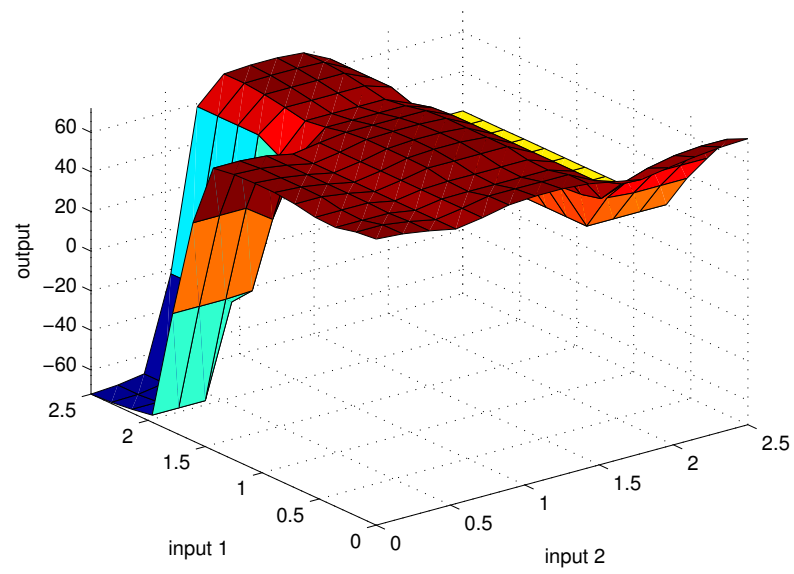


Figure 2. Fuzzy surface of relationship between input 1, input 2 and output.

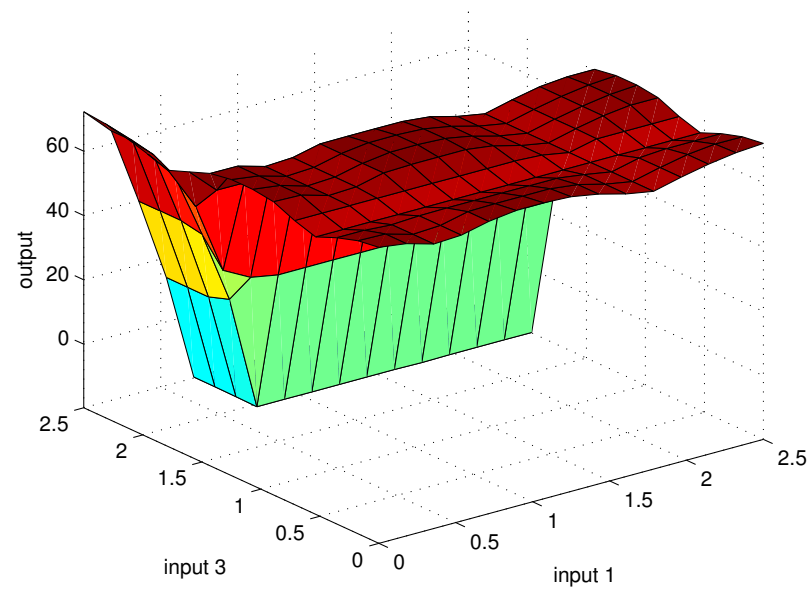


Figure 3. Fuzzy surface of relationship between input 1, input 3 and output.

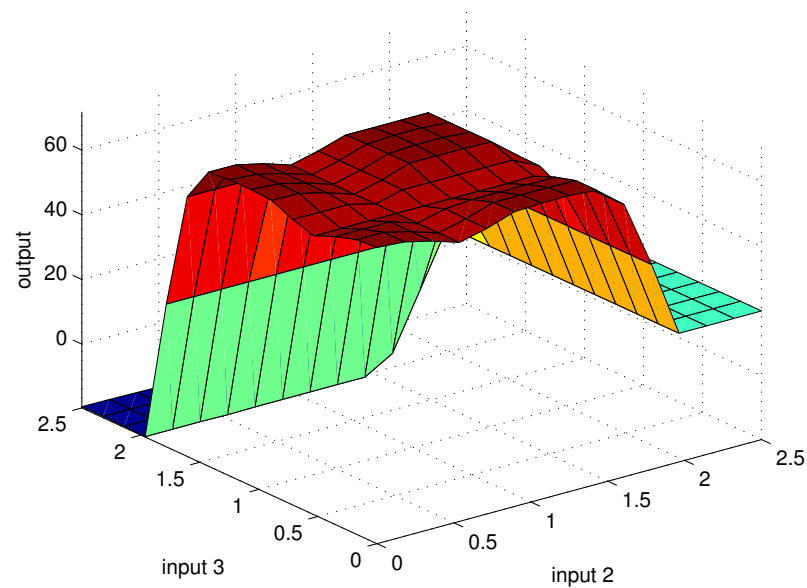


Figure 4. Fuzzy surface of relationship between input 2, input 3 and output.

The following are the fuzzy membership functions for inputs and outputs, together with the fuzzy rules designed for the fuzzy inference system in this work. The system is designed with three inputs and one output, as defined below:

Input 1: distance to obstacle in the flight direction;

Input 2: distance to obstacle at 45 degrees to the left of the direction of flight;

Input 3: distance to obstacle at 45 degrees to the right of the direction of flight;

Output: steering response based on the fuzzy inference.

Note that in the input, if the obstacle is far away, it is denoted by *large*. If the obstacle is close, it is denoted by *small*. If the obstacle is at a moderate distance, it is denoted by *medium*. While in output, we use the angle of turning left to represent the steering response. Similarly, if a small angle to the left is required, it is denoted by *positive small*. If a big angle to the right is required, it is denoted by *negative large*. If going straight is required, it is denoted by *zero*. The fuzzy rules for dynamic obstacle avoidance can be seen in Table 1. Although there are 27 rules in the fuzzy inference system, it could be trained offline and output online. Moreover, if the output is *negative large* or *positive large* according to the fuzzy rule list, it will perform replanning immediately instead of performing fuzzy inference, which improves the efficiency of the system.

As can be seen from the Table 1, if the distance to the obstacle in the flight direction is small, the distance to obstacle at 45 degrees to the left (right) or the direction of flight is not large, the system will perform replanning as traditional RRT methods do. That means only when the distance to the obstacle is not close, the fuzzy inference system would be adopted to perform dynamic obstacle avoidance. Actually, only 9 fuzzy rules are used instead of 27. In this work, triangular and trapezoidal membership functions are utilized for inputs and output of the fuzzy inference system, which can be seen in Figures 5–7.

For inputs, the x-axis denotes the distance, measured in meters, and the y-axis denotes the degree of membership, which takes on a value between zero and one. For the output, the x-axis denotes the angle, measured in degrees, and the y-axis denotes the degree of membership, which has the same range of values as the input membership function.

Table 1. Fuzzy rules for dynamic obstacle avoidance.

Rule No.	Input 1	Input 2	Input 3	Output
1	small	small	small	positive large (replan)
2	small	small	medium	positive large (replan)
3	small	small	large	negative large (replan)
4	small	medium	small	positive large (replan)
5	small	medium	medium	positive large (replan)
6	small	medium	large	negative large (replan)
7	small	large	small	positive large (replan)
8	small	large	medium	negative large (replan)
9	small	large	large	positive large (replan)
10	medium	small	small	positive large (replan)
11	medium	small	medium	negative large (replan)
12	medium	small	large	negative small
13	medium	medium	small	positive large (replan)
14	medium	medium	medium	positive large (replan)
15	medium	medium	large	negative small
16	medium	large	small	positive small
17	medium	large	medium	positive small
18	medium	large	large	positive small
19	large	small	small	positive large (replan)
20	large	small	medium	negative large (replan)
21	large	small	large	negative small
22	large	medium	small	positive large (replan)
23	large	medium	medium	positive large (replan)
24	large	medium	large	negative small
25	large	large	small	positive small
26	large	large	medium	positive small
27	large	large	large	zero

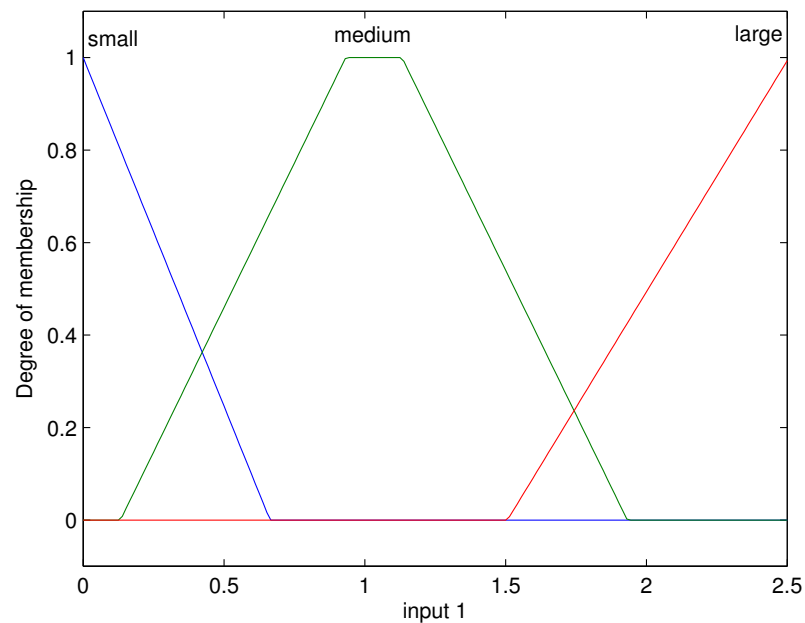


Figure 5. Membership function for input 1 of fuzzy inference system.

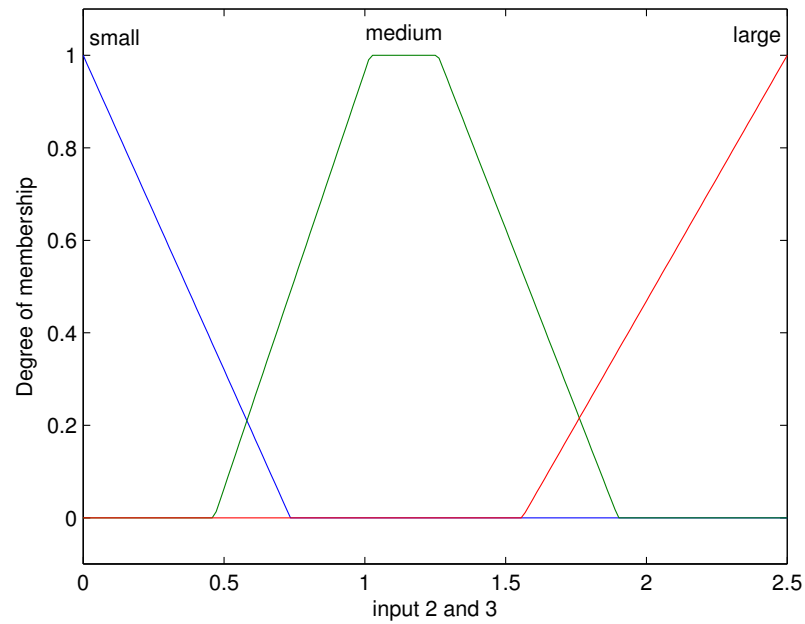


Figure 6. Membership function for input 2 and 3 of the fuzzy inference system.

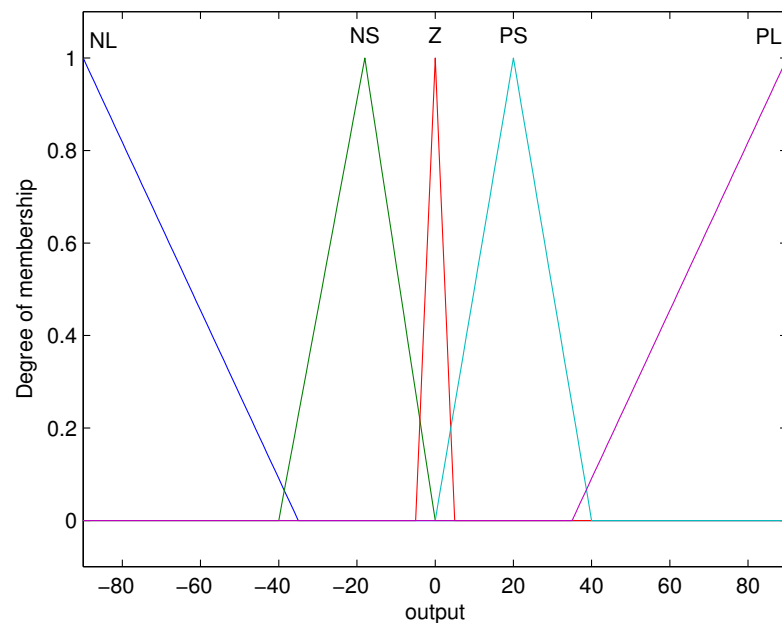


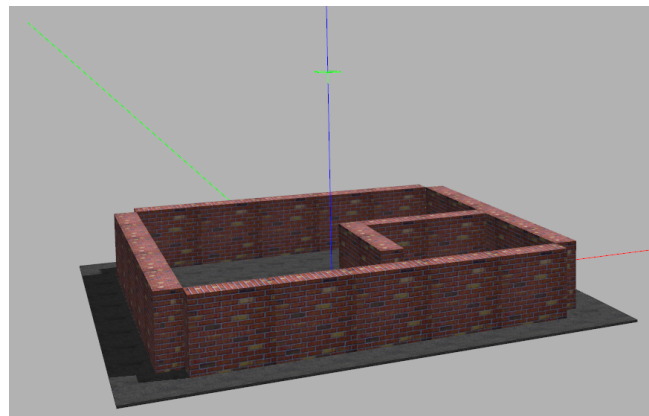
Figure 7. Output of the fuzzy inference system.

4. Results and Analysis

4.1. Experimental Settings

In order to systematically evaluate the potential of the proposed exploration planner, simulations in different scenarios have been performed. For simulation environments, a small-scale maze map, a medium-scale map and a large-scale map in different environments based on the Gazebo simulator [44] are selected to test the performance of the proposed method in different scales and environments, which are shown in Figure 8. The Gazebo simulator is used to provide the environment model, and the RotorS simulator [45] is used to provide the parameters of the MAV (see Figure 9), as well as the ground truth. All the

experiments were run on a laptop with Intel Core i7-10750H at 2.6 GHz. The experiments are repeated 10 times in each environment. Scenarios of different sizes are considered. The proposed algorithm is compared to the NBVP approach [22] and ORB-SLAM3 [12] is adopted to evaluate the accuracy of localization when MAV is carrying out a task. Similar to [22], the parameter λ in Equation (1) is set to 0.5.



(a)



(b)



(c)

Figure 8. Scenarios in different environments: (a) small-scale, (b) medium-scale, (c) large-scale.

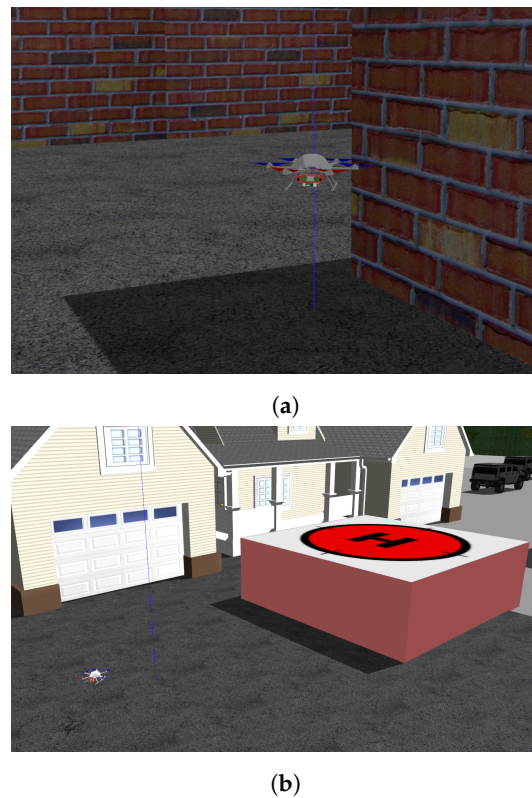


Figure 9. MAV in Gazebo simulator: (a) small-scale and (b) large-scale.

4.2. Results in Different Scenarios

In this section, we evaluate the localization accuracy after the adoption of the proposed active loop closing planning method when compared to the NBVP method, together with the comparison of real-time performance and the influence of different flight speeds on localization accuracy when following the planned path in different scenarios. The size of scenario 1 (Figure 8a) is about $15\text{ m} \times 12\text{ m} \times 3\text{ m}$, the size of scenario 2 (Figure 8b) is about $50\text{ m} \times 40\text{ m} \times 10\text{ m}$ and the size of scenario 3 (Figure 8c) is about $80\text{ m} \times 60\text{ m} \times 12\text{ m}$. In this work, scenario 1 is a small-scale scenario, and scenarios 2 and 3 are medium-scale and large-scale scenarios, respectively. The comparison of real-time performance in different scenarios can be seen in Table 2. Table 3 shows the quantitative evaluation in the small-scale scenario with different flight speeds. The results in Table 3 are for the small-scale scenario. Table 4 shows the quantitative evaluation in different scenarios.

Table 2. Comparison of time-consuming performance in different scenarios ($v = 0.2\text{ m/s}$).

Sequence	Planning Time		Active Loop Closing Time		Planning Times		Total Time Spent	
	NBVP (s)	ALCP (s)	NBVP (s)	ALCP (s)	NBVP	ALCP	NBVP (s)	ALCP (s)
<i>small-scale</i>	0.30	0.32	/	0.35	101	95	361.8	385.6
<i>medium-scale</i>	0.31	0.35	/	0.55	465	453	1161.8	1235.6
<i>large-scale</i>	0.35	0.41	/	0.69	851	783	1779.8	1865.2

Figures 10 and 11 show the trajectory error evaluated by ORB-SLAM3 in different scenarios. The dashed line represents the ground truth obtained from the Gazebo simulator, and the blue line represents the state estimation results obtained by ORB-SLAM3. The poor localization performance in the small-scale scenario is due to the fact that the scenario consists mainly of walls and is not rich in texture information. Figures 12–14 present the exploration results of different scenarios. The green lines in the figure represent the planned

path. As can be seen from Tables 1–4, we can find that in terms of the time required for one path planning session, the time spent on ALCP is a little bit more than that on the NBVP. The time spent on active loop closing is similar to that on path planning. The scenario, together with flight speed, will affect the performance of the localization, which will be discussed in detail in Section 5. The selection of the active loop closing check threshold will be discussed as well.

Table 3. Comparison of absolute trajectory RMSE and mean error with different flight speeds.

Speed	NBVP		ALCP	
	RMSE (m)	Mean (m)	RMSE (m)	Mean (m)
$v = 0.2$ m/s	3.25	2.56	3.21	2.87
$v = 0.5$ m/s	3.02	2.32	3.44	2.70
$v = 0.8$ m/s	4.89	3.66	4.19	3.52
$v = 1.2$ m/s	3.85	3.48	3.75	3.27
$v = 1.5$ m/s	10.56	5.05	3.36	2.82

Table 4. Comparison of absolute trajectory RMSE and mean error in different scenarios ($v = 0.2$ m/s).

Scenarios	NBVP		ALCP	
	RMSE (m)	Mean (m)	RMSE (m)	Mean (m)
<i>small-scale</i>	3.25	2.56	3.21	2.87
<i>medium-scale</i>	2.32	1.78	1.00	0.91
<i>large-scale</i>	4.89	3.66	3.45	3.49

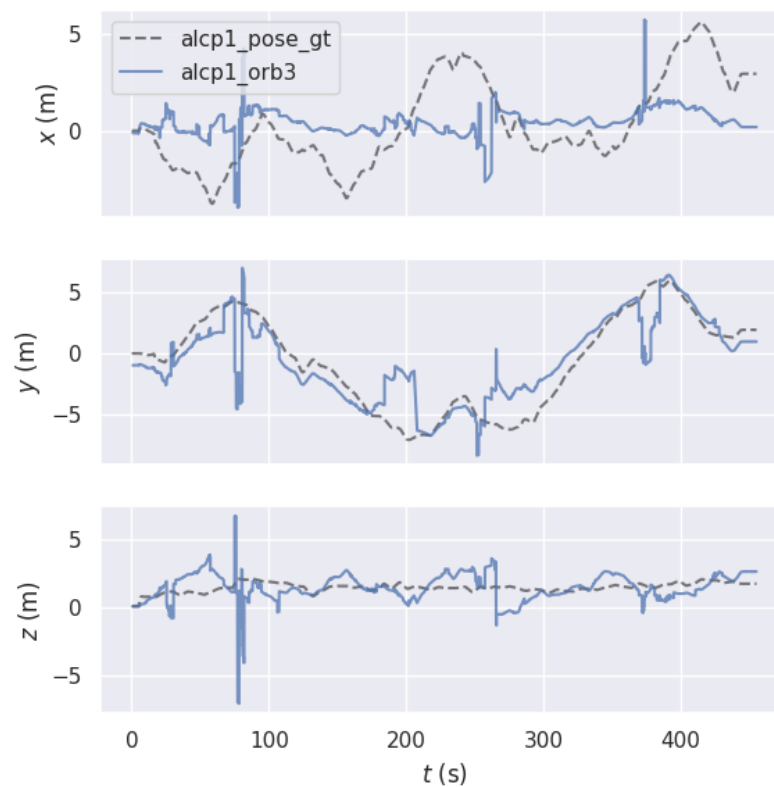


Figure 10. Trajectory of ORB-SLAM3 in the *small-scale* scenario.

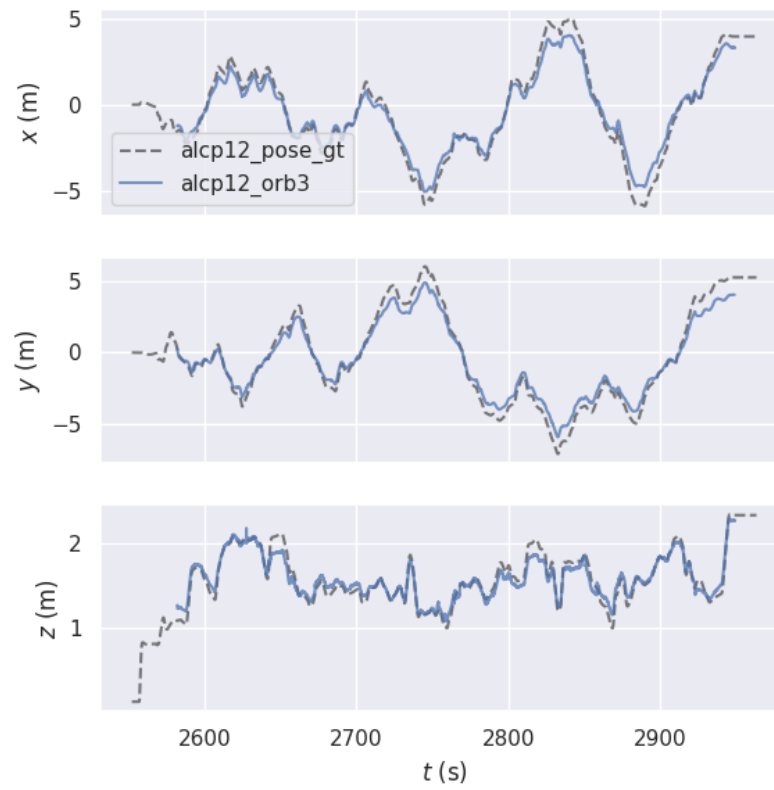


Figure 11. Trajectory of ORB-SLAM3 in the *medium-scale* scenario.

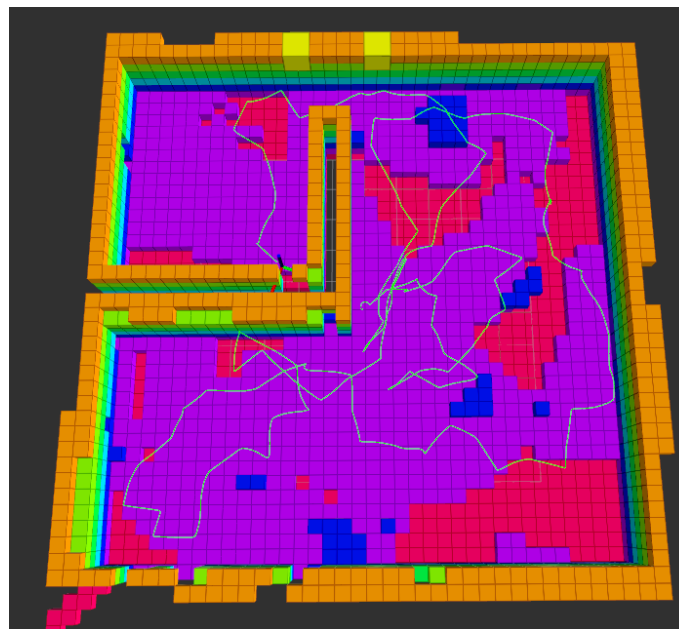


Figure 12. Exploration result in different environments (*small-scale*).

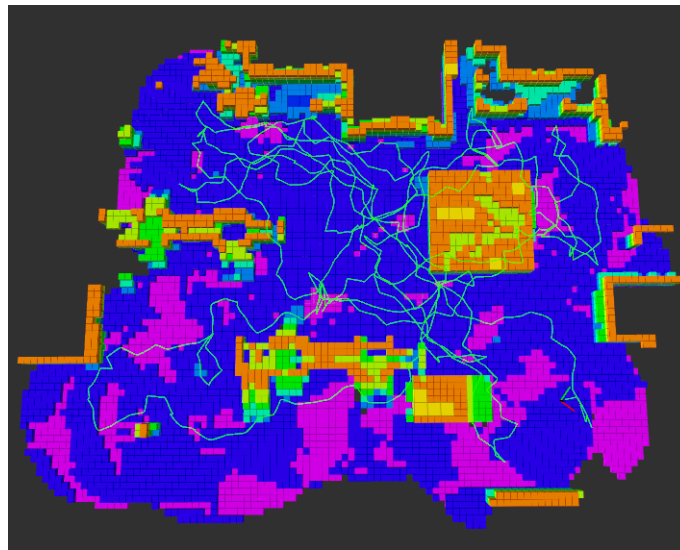


Figure 13. Exploration result in different environments (*middle-scale*).

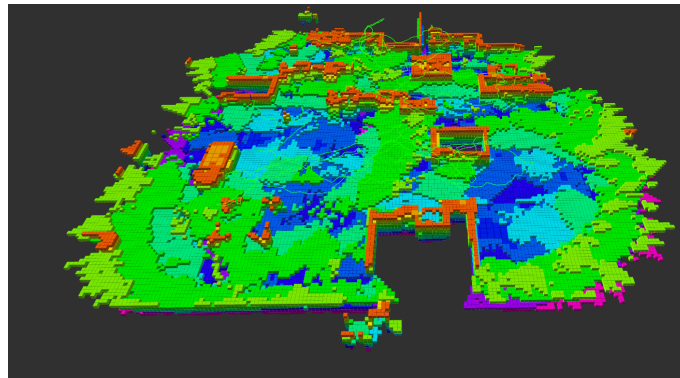


Figure 14. Exploration result in different environments (*large-scale*).

5. Discussion

In this section, we mainly discuss the real-time performance of the active loop closing planning method and influence of different active loop closing check threshold selection strategies, together with the comparison of the localization accuracy in different scenarios and different flight speeds mentioned in Section 4.

5.1. Comparison of Real-Time Performance

The time spent on ALCP is a bit more than that on NBVP. The average time spent on active loop closing is similar to that on path planning. It means the active loop closing does not significantly degrade the real-time performance system. However, we find that in large-scale scenarios, the loop closing check takes longer and longer due to the increasing number of key waypoint candidates during the system execution.

5.2. Comparison of Different Active Loop Closing Check Threshold Selection Strategies

Different active loop closing check threshold selection (such as after every 10, 15, 25, 25, 30 path planning iterations) will influence the performance of the localization. The higher the frequency of active loop closing, the smaller the localization error. However, as more loops are detected, the total time spent on exploration becomes longer. As the loop closing frequency decreases, the contribution of the active loop closing will become weaker, and the performance of the ALCP planner will be close to the NBVP planner. Table 5 shows the results of different active loop closing check threshold selection strategies in a small-scale scenario.

Table 5. Comparison of different active loop closing check threshold selection strategies ($v = 0.2$ m/s).

Check Frequency	ALCP			
	RMSE (m)	Mean (m)	Planning Times	Total Time Spent (s)
Every 10 path planning	2.94	1.89	113	484.7
Every 15 path planning	3.01	1.94	106	448.3
Every 20 path planning	3.25	2.56	95	385.6
Every 25 path planning	3.21	2.58	92	376.4
Every 30 path planning	3.86	2.73	101	370.2

5.3. Comparison of the Localization Accuracy in Different Scenarios

In terms of localization accuracy, the active loop closing path planning method differs in its ability to adapt to different scenarios. One of the reasons for this is, of course, the nature of the SLAM algorithm. Scenario 1 consists mainly of walls, and although ORB-SLAM3 is able to extract feature points for localization and mapping, the lack of rich texture features results in poor localization accuracy, even though the active loop closing strategy is adopted. In scenario 2, the accuracy is much better due to the rich texture information. As the flight speed increases, the localization accuracy without active loop closing will be slightly worse than that with active loop closing during the planning, but it is not the main factor. However, the small-scale scenario in the experiments consists mainly of walls and is not rich in texture information. It is not good for the localization performance of the SLAM system. So, the performance is poor even with active loop closing. Moreover, in the small-scale scenario, the increase in speed is also not good for localization because when MAV moves, the surrounding keypoints will change, keypoint matching is difficult. As the flight speed becomes faster, once the localization is lost or there is a large error in the state estimation, it is difficult for the localization error to converge quickly in a short period of time without active loop closing, which can lead to large error fluctuations or even failure of the feature-based SLAM algorithm. Notice that, in this work, we combine path planning and SLAM, and we pay more attention to localization performance instead of coverage rate during the exploration.

6. Conclusions

A perception-aware path planner, the Active Loop Closing Planner (ALCP), in dynamic environments using Micro-Aerial Vehicles is presented in this paper. The combination of path planning and active loop closing makes the system an active SLAM system. The planned trajectories can deal with obstacles in both static and dynamic unknown environments while active loop closing is beneficial to reduce uncertainty in the SLAM system and further improve the accuracy of localization. During the path planning, the loop closing cost is proposed to improve the gain cost function, and it takes the loop closing possibility into account. In addition, a key waypoints selection strategy is designed to list selected key waypoints, instead of all waypoints, as potential loop closing points to make the algorithm more efficient. In addition, a fuzzy-RRT-based dynamic obstacle avoidance algorithm is adopted to realize obstacle avoidance in dynamic environments. Simulations in different challenging scenarios are conducted to verify the localization accuracy of the proposed active loop closing path planning based SLAM algorithm.

Author Contributions: Material preparation, Y.Z., S.Z. and J.W.; methodology, Y.Z.; data collection and analysis, Y.Z. and S.Z.; validation, Y.Z., J.W. and S.Z.; writing—original draft preparation, Y.Z.; writing—review and editing, Z.X., L.Z. and P.C.; supervision, Z.X. and P.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the National Natural Science Foundation of China (Grant No. 62073163, 61873125), the Introduction plan of high end experts (Grant No. G20200010142), the Natural Science Fund of Jiangsu Province (Grant No. BK20181291), the Fundamental Research

Funds for the Central Universities (Grant No. NZ2020004), the Aeronautic Science Foundation of China (Grant No. ASFC-2020Z071052001), the Shanghai Aerospace Science and Technology Innovation Fund (Grant No. SAST2019-085), the 111 Project (Grant No. B20007) and the China Scholarship Council.

Data Availability Statement: Some or all data generated during the study may only be provided with restrictions upon request.

Acknowledgments: The authors would like to thank Liang Lu and Yiming Ding for the fruitful discussion.

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

References

1. Ceccarelli, N.; Enright, J.J.; Frazzoli, E.; Rasmussen, S.J.; Schumacher, C.J. Micro UAV path planning for reconnaissance in wind. In Proceedings of the 2007 American Control Conference, New York, NY, USA, 9–13 July 2007; pp. 5310–5315.
2. Quenzel, J.; Nieuwenhuisen, M.; Droschel, D.; Beul, M.; Houben, S.; Behnke, S. Autonomous MAV-based indoor chimney inspection with 3D laser localization and textured surface reconstruction. *J. Intell. Robot. Syst.* **2019**, *93*, 317–335. [\[CrossRef\]](#)
3. Dąbrowski, P.S.; Specht, C.; Specht, M.; Burdziakowski, P.; Makar, A.; Lewicka, O. Integration of multi-source geospatial data from GNSS receivers, terrestrial laser scanners, and unmanned aerial vehicles. *Can. J. Remote. Sens.* **2021**, *47*, 621–634. [\[CrossRef\]](#)
4. He, J.; Lin, J.; Ma, M.; Liao, X. Mapping topo-bathymetry of transparent tufa lakes using UAV-based photogrammetry and RGB imagery. *Geomorphology* **2021**, *389*, 107832. [\[CrossRef\]](#)
5. Wang, D.; Xing, S.; He, Y.; Yu, J.; Xu, Q.; Li, P. Evaluation of a New Lightweight UAV-Borne Topo-Bathymetric LiDAR for Shallow Water Bathymetry and Object Detection. *Sensors* **2022**, *22*, 1379. [\[CrossRef\]](#)
6. Lutz, P.; Müller, M.G.; Maier, M.; Stoneman, S.; Tomić, T.; von Barga, I.; Schuster, M.J.; Steidle, F.; Wedler, A.; Stürzl, W.; et al. ARDEA—An MAV with skills for future planetary missions. *J. Field Robot.* **2020**, *37*, 515–551. [\[CrossRef\]](#)
7. Bi, Y.; Lan, M.; Li, J.; Lai, S.; Chen, B.M. A lightweight autonomous MAV for indoor search and rescue. *Asian J. Control.* **2019**, *21*, 1732–1744. [\[CrossRef\]](#)
8. Wang, L.; Cavallaro, A. Microphone-array ego-noise reduction algorithms for auditory micro aerial vehicles. *IEEE Sensors J.* **2017**, *17*, 2447–2455. [\[CrossRef\]](#)
9. Qin, T.; Li, P.; Shen, S. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [\[CrossRef\]](#)
10. Forster, C.; Zhang, Z.; Gassner, M.; Werlberger, M.; Scaramuzza, D. SVO: Semidirect visual odometry for monocular and multicamera systems. *IEEE Trans. Robot.* **2016**, *33*, 249–265. [\[CrossRef\]](#)
11. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 834–849.
12. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.; Tardós, J.D. Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [\[CrossRef\]](#)
13. Lluvia, I.; Lazkano, E.; Ansuategi, A. Active mapping and robot exploration: A survey. *Sensors* **2021**, *21*, 2445. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Fraundorfer, F.; Heng, L.; Honegger, D.; Lee, G.H.; Meier, L.; Tanskanen, P.; Pollefeys, M. Vision-based autonomous mapping and exploration using a quadrotor MAV. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 4557–4564.
15. Meng, Z.; Sun, H.; Qin, H.; Chen, Z.; Zhou, C.; Ang, M.H. Intelligent robotic system for autonomous exploration and active SLAM in unknown environments. In Proceedings of the 2017 IEEE/SICE International Symposium on System Integration (SII), Taipei, Taiwan, 1–14 December 2017; pp. 651–656.
16. Qin, H.; Meng, Z.; Meng, W.; Chen, X.; Sun, H.; Lin, F.; Ang, M.H. Autonomous exploration and mapping system using heterogeneous UAVs and UGVs in GPS-denied environments. *IEEE Trans. Veh. Technol.* **2019**, *68*, 1339–1350. [\[CrossRef\]](#)
17. Papachristos, C.; Khattak, S.; Alexis, K. Uncertainty-aware receding horizon exploration and mapping using aerial robots. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 4568–4575.
18. Kan, X.; Teng, H.; Karydis, K. Online exploration and coverage planning in unknown obstacle-cluttered environments. *IEEE Robot. Autom. Lett.* **2020**, *5*, 5969–5976. [\[CrossRef\]](#)
19. Papachristos, C.; Mascarich, F.; Khattak, S.; Dang, T.; Alexis, K. Localization uncertainty-aware autonomous exploration and mapping with aerial robots using receding horizon path-planning. *Auton. Robot.* **2019**, *43*, 2131–2161. [\[CrossRef\]](#)
20. Rodríguez-Arévalo, M.L.; Neira, J.; Castellanos, J.A. On the importance of uncertainty representation in active SLAM. *IEEE Trans. Robot.* **2018**, *34*, 829–834. [\[CrossRef\]](#)
21. Deng, X.; Zhang, Z.; Sintov, A.; Huang, J.; Bretl, T. Feature-constrained active visual SLAM for mobile robot navigation. In Proceedings of the 2018 IEEE international conference on robotics and automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 7233–7238.

22. Bircher, A.; Kamel, M.; Alexis, K.; Oleynikova, H.; Siegwart, R. Receding horizon “next-best-view” planner for 3d exploration. In Proceedings of the 2016 IEEE international conference on robotics and automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1462–1468.
23. Yamauchi, B. A frontier-based approach for autonomous exploration. In Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation', Monterey, CA, USA, 10–11 July 1997; pp. 146–151.
24. Naazare, M.; Garcia-Rosas, F.J.; Schulz, D. Online Next-Best-View Planner for 3D-Exploration and Inspection with a Mobile Manipulator Robot. *IEEE Robot. Autom. Lett.* **2022**, *7*, 3779–3786. [[CrossRef](#)]
25. Papachristos, C.; Khattak, S.; Alexis, K. Autonomous exploration of visually-degraded environments using aerial robots. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 775–780.
26. Schmid, L.; Pantic, M.; Khanna, R.; Ott, L.; Siegwart, R.; Nieto, J. An efficient sampling-based method for online informative path planning in unknown environments. *IEEE Robot. Autom. Lett.* **2020**, *5*, 1500–1507. [[CrossRef](#)]
27. Zhang, Z.; Scaramuzza, D. Perception-aware receding horizon navigation for MAVs. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 2534–2541.
28. Zhou, B.; Pan, J.; Gao, F.; Shen, S. Raptor: Robust and perception-aware trajectory replanning for quadrotor fast flight. *IEEE Trans. Robot.* **2021**, *37*, 1992–2009. [[CrossRef](#)]
29. Zhang, Z.; Scaramuzza, D. Beyond point clouds: Fisher information field for active visual localization. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 5986–5992.
30. Song, S.; Jo, S. Surface-based exploration for autonomous 3d modeling. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 4319–4326.
31. Wang, Q.; Gao, Y.; Ji, J.; Xu, C.; Gao, F. Visibility-aware trajectory optimization with application to aerial tracking. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 5249–5256.
32. Lu, L.; Redondo, C.; Campoy, P. Optimal frontier-based autonomous exploration in unconstructed environment using RGB-D sensor. *Sensors* **2020**, *20*, 6507. [[CrossRef](#)]
33. Dang, T.; Papachristos, C.; Alexis, K. Visual saliency-aware receding horizon autonomous exploration with application to aerial robotics. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 2526–2533.
34. Dai, A.; Papatheodorou, S.; Funk, N.; Tzoumanikas, D.; Leutenegger, S. Fast frontier-based information-driven autonomous exploration with an mav. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 9570–9576.
35. Khan, A.; Noreen, I.; Habib, Z. Coverage path planning of mobile robots using rational quadratic Bézier spline. In Proceedings of the 2016 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 19–21 December 2016; pp. 319–323.
36. Song, S.; Kim, D.; Jo, S. Online coverage and inspection planning for 3D modeling. *Auton. Robot.* **2020**, *44*, 1431–1450. [[CrossRef](#)]
37. Galceran, E.; Carreras, M. A survey on coverage path planning for robotics. *Robot. Auton. Syst.* **2013**, *61*, 1258–1276. [[CrossRef](#)]
38. Tordesillas, J.; How, J.P. PANTHER: Perception-aware trajectory planner in dynamic environments. *IEEE Access* **2022**, *10*, 22662–22677. [[CrossRef](#)]
39. Lu, L.; Carrio, A.; Sampedro, C.; Campoy, P. A Robust and Fast Collision-Avoidance Approach for Micro Aerial Vehicles Using a Depth Sensor. *Remote Sens.* **2021**, *13*, 1796. [[CrossRef](#)]
40. Tordesillas, J.; How, J.P. MADER: Trajectory planner in multiagent and dynamic environments. *IEEE Trans. Robot.* **2021**, *38*, 463–476. [[CrossRef](#)]
41. Quan, L.; Zhang, Z.; Zhong, X.; Xu, C.; Gao, F. EVA-planner: Environmental adaptive quadrotor planning. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 398–404.
42. González-Banos, H.H.; Latombe, J.C. Navigation strategies for exploring indoor environments. *Int. J. Robot. Res.* **2002**, *21*, 829–848. [[CrossRef](#)]
43. Driankov, D.; Hellendoorn, H.; Reinfrank, M. *An Introduction to Fuzzy Control*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
44. Koenig, N.; Howard, A. Design and use paradigms for gazebo, an open-source multi-robot simulator. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No. 04CH37566), Sendai, Japan, 28 September–2 October 2004; Volume 3, pp. 2149–2154.
45. Furrer, F.; Burri, M.; Achtelik, M.; Siegwart, R. RotorS—A modular gazebo MAV simulator framework. In *Robot Operating System (ROS)*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 595–625.