

Autonomous Landing Spot Detection for Unmanned Aerial Vehicles Based on Monocular Vision

Yanan Cao¹, Minglei Li¹, Li Xu¹, Mingqiang Wei²

¹ College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, 211106 Nanjing, China - (caoyanan, minglei li, sz2204056)@nuaa.edu.cn

² College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, 211106 Nanjing, China - (mqwei@nuaa.edu.cn)@nuaa.edu.cn

Keywords: Monocular vision, Landing spot detection, SLAM, UAV.

Abstract

Enhancing the autonomous landing capability of unmanned aerial vehicles (UAVs) is of great significance for improving their operational efficiency and field survival capabilities. To this end, we propose a real-time autonomous landing spot detection method for UAVs. Firstly, the pose of the UAV at any given time and the initial three-dimensional point cloud of the scene are estimated using simultaneous localization and mapping (SLAM) techniques. Then, since the initial point cloud is sparse and cannot be used for terrain analysis, we generate a voxel-based elevation map of the scene, which can establish interconnectivity among the points. Finally, we propose a shift-box strategy to comprehensively analyze various terrain factors in the elevation map, determine the landing spot for UAVs, and update the landing spot in real time. The UAV flight experiments conducted in the real world have demonstrated the effectiveness and real-time performance of the proposed method.

1. Introduction and Related Work

Landing spot detection is important for unmanned aerial vehicles (UAVs) in many applications, such as wilderness rescue, reconnaissance, geological exploration, disaster relief, and environmental monitoring. They need to accurately identify, approach, and land in safe spots without human intervention. At present, research on autonomous landing spot detection of UAVs is very challenging without prior knowledge (Yubo et al., 2021).

Global Positioning System (GPS) can provide three-dimensional spatial location parameters for UAVs (Laiacker et al., 2013). However, relying solely on GPS signals cannot obtain terrain and obstacle information in the flight area. Moreover, in environments such as forests, valleys, and areas with high-density buildings, GPS navigation signals may be interfered with, leading to inaccuracies in the navigational data and consequently affecting the reliability of the system.

Lidar can obtain three-dimensional point cloud data of the scene. By designing a candidate plane extraction algorithm for point cloud data, flat areas that meet landing requirements can be selected (Johnson et al., 2002; Scherer et al., 2012; Xing et al., 2020). However, such sensors are expensive, heavy, and not suitable for small UAVs. In contrast, optical cameras have advantages such as being lightweight and low-cost, making them suitable as environmental sensors for UAVs (Harshit et al., 2022; Scaramuzza et al., 2014; Kong et al., 2014). However, challenges still exist in measuring the position and attitude of the UAV, as well as determining the landing spot, when acquiring video images through onboard cameras in the absence of GPS navigation signals.

In the past few years, there have been many studies on vision-based autonomous landing of UAVs, which can be divided into three categories based on technical means: methods based on visual marker recognition, methods based on ground feature

extraction and matching, and methods based on feature-based visual odometry.

1) Methods based on visual marker recognition. This type of method utilizes computer vision algorithms to identify specific landmarks placed on the landing spot and solves the flight pose parameters of the UAV through feature extraction and matching operations (Patrino et al., 2019; Cabrera-Ponce and Martinez-Carranza, 2017; Xin et al., 2022). Chen et al. (2017) used an improved R-CNN neural network to recognize landmarks. However, the landing spot detection method based on landmarks requires designing different indicator landmarks according to different application scenarios, and the detection algorithm also needs to be designed based on the geometric characteristics of the indicator landmarks. Therefore, the UAV can only achieve a safe landing in a given scenario. This type of algorithm has poor stability and robustness and is inappropriate for certain scenarios, such as post-disaster search and rescue operations, where it is not feasible to pre-install landmarks. Therefore, UAVs must have the ability to autonomously analyze the surrounding environment and select a safe landing spot.

2) Methods based on ground feature extraction and matching. This method uses the visual system of UAVs to obtain images of the pre-landing spot, extract ground features (such as edges, corners, etc.), match them with ground templates or previously established maps, and then determine the ideal landing spot based on the matching results (SUO et al., 2020). Miller et al. (2008) created an image reference database for drone runways. When the drone arrives at the flight area, the flight scene information is matched with the information collected by the processor on board the drone to obtain the distance and attitude angle of the drone relative to the runway. This method does not require pre-setting indicating landmarks. However, this type of method requires high accuracy of the ground template. If there is a significant difference between the ground template and the actual scene, the matching results may be incorrect, resulting in the inability to accurately find the ideal landing spot.

3) Methods based on feature-based visual odometry. This type of method acquires real-time images of the flight area through the UAV's visual system, derives the relative positions and attitudes between sequence images based on projection geometric relationships, and restores the motion trajectory of the UAV (Zeng et al., 2022; Engel et al., 2012). Forster et al. (2015) from the Federal Institute of Technology in Zurich proposed a multi-rotor UAV landing spot recognition method that utilizes the elevation map framework (Fankhauser et al., 2014) to establish an elevation model of the scene. The method uses a semi-direct visual odometry (SVO) based on a monocular camera to estimate the UAV's pose.

When dealing with the challenges of safely landing UAVs, not only should the position and attitude information of UAVs be considered, but the three-dimensional terrain of the landing spot is also an important aspect of environmental perception.

Yang et al. (2018) at Northwestern Polytechnical University conducted research on the landing of UAVs in unknown areas. They proposed a new map representation method, which combines the three-dimensional features of the unknown area's point cloud and uses the region segmentation method to analyze terrain, achieving recognition of obstacles and flat areas. Mittal et al. (2019) selected the landing spot for UAVs by establishing a digital elevation model (DEM) of the scene. This method relies on the Canny operator for edge extraction and still has limitations.

In this paper, we propose a real-time autonomous landing spot detection method for UAVs using a monocular camera, which does not rely on satellite signals and prior ground knowledge. The simultaneous localization and mapping (SLAM) technique is used to construct the initial point cloud of the scene and estimate the pose of the UAV in real time. To conduct terrain analysis, we voxelize the acquired sparse point cloud using an octree-based method to incrementally build the elevation map of the scene. Finally, we propose a shift-box strategy to effectively utilize the elevation map to analyze terrain and comprehensively consider various terrain factors to determine and update the landing spot in real time.

2. System Overview

The workflow of the proposed autonomous landing spot detection method consists of an elevation mapping module and a landing spot selection module. An illustration of the approach is shown in Figure 1.

In the elevation mapping module, we make use of the ORB-SLAM2 (Mur-Artal and Tardós, 2017) framework to estimate the pose of the UAV and incrementally construct an initial three-dimensional point cloud of the scene in real time. This initial point cloud provides a rough representation of the scene structure, serving as a data source for the subsequent module. To address the sparsity of the initial point cloud, we utilize a voxelization method based on octree to generate an elevation map of the scene, which can establish connectivity among points. By analyzing the properties and occupancy probability of each cell in the elevation map, the geometric changes of the three-dimensional terrain are examined. The elevation map can be dynamically updated and maintained as the point cloud is incrementally constructed.

In the landing spot selection module, firstly, we design a ground filtering mechanism based on the RANdom SAmple Consensus (RANSAC) algorithm to extract the horizontal ground surface in the elevation map. The centers of cells in the elevation map are used as the input for the RANSAC algorithm, instead of the original three-dimensional point cloud, which greatly improves the algorithm's running speed. Then, we propose a shift-box algorithm to cover the entire ground surface and determine the most suitable landing spot for the UAV by calculating the flatness in each box. Furthermore, the landing spot will be dynamically adjusted based on the flight position of the UAV to ensure that it prioritizes the position closest to the UAV.

3. Methodology

3.1 Elevation Mapping

Once the UAV receives a landing command, it explores the current scene. In the first module, the input is real-time images obtained by the visual sensor on the UAV. The ORB-SLAM2 algorithm is a visual SLAM algorithm based on feature points, which utilizes the geometric relationships between sequence images to calculate the position and pose of UAVs, and simultaneously constructs the three-dimensional point cloud structure of the current scene. The sparse point cloud obtained by the ORB-SLAM2 algorithm presents limitations in the task of landing spot detection, and the storage of point cloud data occupies a large amount of memory. However, for the UAV landing task, algorithms usually need to have high computational efficiency. Therefore, we adopt a hierarchical data structure based on octree to voxelize the three-dimensional point cloud in order to establish a map with connected regions for the subsequent rapid landing spot detection task.

The octree recursively divides the space into eight equally sized cubes, each corresponding to a node, and each node can represent a region, called voxels. If a voxel is occupied in three-dimensional space, the corresponding node in the octree is initialized. By using the hierarchical structure of the octree, it is possible to quickly locate the desired spatial area, reduce search scope, and improve query efficiency. However, during the flight of UAVs, the environment below is constantly changing, and simple discrete occupancy labels usually cannot fully describe the environmental state, making it difficult to capture the uncertainty and variability of the environment. Therefore, it is necessary to use probabilistic modeling methods to quantify the uncertainty of the environment, achieve more precise modeling of the environmental state, and provide a more reliable basis for subsequent landing spot selection decisions.

Assuming the observed data at time t_1, t_2, t_3 is z_1, z_2, z_3 , the probability $P(n|z_{1:t})$ of a leaf node n being occupied is:

$$P(n|z_{1:t}) = \left[1 + \frac{1 - P(n|z_t)}{P(n|z_t)} \frac{1 - P(n|z_{1:t-1})}{P(n|z_{1:t-1})} \frac{P(n)}{1 - P(n)} \right]^{-1} \quad (1)$$

where $P(n|z_{1:t-1})$ = the observations at previous moments
 $P(n)$ = prior probability (typically taking a value of 0.5)

By using the log-odds notation, Equation (1) can be rewritten as

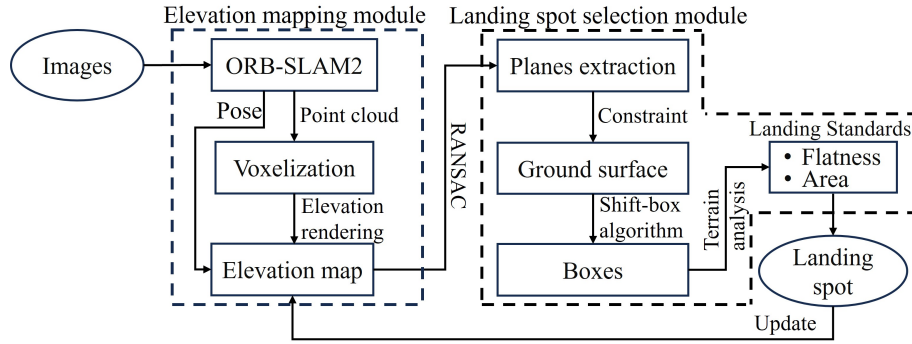


Figure 1. Workflow of the autonomous landing spot detection method for UAVs.

$$L(n|z_{1:t}) = L(n|z_{1:t-1}) + L(n|z_t) \quad (2)$$

with

$$L(n) = \log\left[\frac{P(n)}{1 - P(n)}\right] \quad (3)$$

If the probability of the current node exceeds 0.5, it is considered occupied; if it is below 0.5, it is considered idle. The depth of the octree is set to 16. As can be seen from Equation (2), we have accumulated the observation data at each moment, so when the environment changes, the occupancy status of nodes will also be updated accordingly. Finally, according to the height data within each voxel, we perform color rendering to generate an elevation map of the scene.

3.2 Landing Spot Selection

When the UAV receives a landing command, it needs to select a large enough and flat spot on the ground surface as the pre-landing spot.

3.2.1 Ground Filtering Mechanism: In the landing spot selection module, we first utilize the RANdom SAMple Consensus (RANSAC) algorithm to extract planes from the elevation map. The input of the RANSAC algorithm is the center of each cell in the elevation map. Since the elevation map is the result of our voxel resampling of the three-dimensional point cloud, the data complexity is reduced, resulting in a significant enhancement in the computational efficiency of RANSAC in plane extraction. Furthermore, the cell centers within the elevation map are uniformly arranged, thereby simplifying the parameter configuration process for the RANSAC algorithm, as only integer multiples of the elevation map resolution need to be taken into account when setting the threshold. This makes the plane extraction more convenient and efficient. Figure 2 shows the voxel resampled regular point cloud and the plane extraction process.

Then, since each plane has a normal vector, the horizontal ground surface among these planes can be identified under the constraint of the direction of the normal vector. We calculate the angle θ_i between each plane and the horizontal plane by calculating the angle between the normal vector \vec{n}_i of each plane and the unit vector \vec{n}_0 perpendicular to the horizontal plane:

$$\theta_i = \arccos \frac{(\vec{n}_i \cdot \vec{n}_0)}{|\vec{n}_i| \cdot |\vec{n}_0|} \quad (4)$$

When the angle approaches 0, the plane in which the normal vector \vec{n}_i resides is considered to be the ground surface.

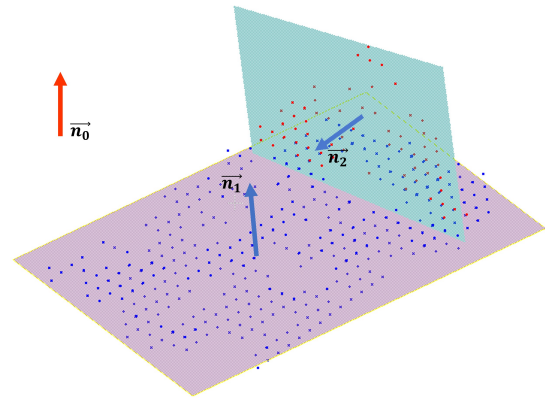


Figure 2. Ground surface extraction.

3.2.2 Landing Spot Selection: To find a large enough and flat landing spot on the ground surface, we propose a shift-box strategy to cover the entire ground surface and analyze the flatness of the terrain. This strategy is described in the following.

Firstly, we define a voxel cell G , which is located within the range of the elevation map and has two-dimensional coordinates (x, y) . Let us consider using R as the distance threshold to form a set of cells $T(x_0, y_0, R)$ around cell G . Each cell in set T has a two-dimensional coordinate (x, y) in the elevation map. Then, we create a box with a specific edge length according to the following formula:

$$S_k = \sum_{(x,y) \in T(x_0, y_0, R)} \|x - x_0\|^2 + \|y - y_0\|^2 \quad (5)$$

As shown in Figure 3, the box shifts across the ground surface with a certain iteration step size. S_k represents the set of cells within the box after k movements. In order to guarantee that the landing spot has a sufficiently large area, the area A_c of the box determined by the distance threshold R must satisfy the following criteria:

$$A_c \geq 2 * A_d \quad (6)$$

where A_d represents the area occupied by the UAV on the ground.

By moving the box on the ground with a certain iteration step size, we can compute the flatness of all areas in the elevation map. This is achieved by calculating the sum of squared height differences for all cells contained in the box after each move.

Let us define the set of cells in the box S_k as B , and the height value of each cell as Z . We calculate the flatness of the area on the ground represented by the box according to the following formula:

$$P_k = \sum_{i \in B} \sum_{j \in B} \|Z_j - Z_i\|^2 \quad (7)$$

where P_k denotes the flatness of the area represented by the box after k movements. A lower flatness value indicates a flatter area. Finally, the region where the box with the highest flatness is located is identified as the landing spot. When boxes with the same flatness appear in the updated elevation map, the landing spot will be dynamically adjusted to the location of the nearest box to the UAV. From the construction of the elevation map, it is likely that there are cells in some boxes that do not have height values. This situation occurs because the SLAM system is unable to effectively recover map points in areas lacking texture. Considering the safety during the landing of the UAV, we directly set the flatness value within this type of box to P_{max} to ensure that the UAV lands in a spot with clear terrain information.

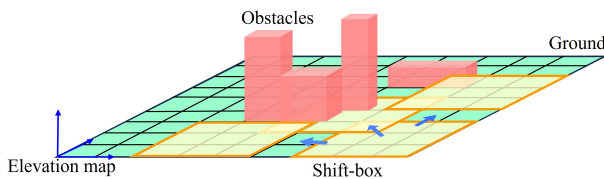


Figure 3. Move the box to cover the entire ground.

4. Experiments

4.1 Experimental Platform

DJI Mini 3 quadcopter drone is used as the platform for data collection in this paper. It is equipped with a barometer and a monocular camera sensor with 48 million pixels. The size of the quadcopter drone after deployment (including blades) is 251mm x 362mm x 72mm.

We configured the experimental environment on a desktop computer equipped with a 12-core 2.50GHz Intel Core™ i5-12400 CPU and a Linux operating system with 15.0 GB of memory. Meanwhile, to visualize the experimental results, we conducted data processing and visualization development in the ROS operating system.

4.2 Outdoor Flight Experiments

We conducted experiments in two sets of scenarios separately. In the first set of scenarios, the drone flew at an altitude of 20 meters above ground level, using an elevation map with a cell resolution of approximately 1 meter. In the second set

of scenarios, the drone flew at an altitude of 60 meters, with an elevation map resolution of approximately 0.67 meters. The images of the two scenarios captured by the camera on the drone are shown respectively in the top row of Figure 4 and Figure 5. The elevation maps and landing spot detection results at corresponding times are depicted in the second row of Figure 4 and Figure 5. The position of the landing spot at each moment corresponds to the location depicted by the red box in the actual scene. The highlighted spot in the elevation map represents the landing spot, and the green line and coordinate axis positioned above the elevation map represent the drone's flight trajectory and its current pose, respectively. It can be seen from the experimental results that the drone accurately detected a large enough and flat landing spot in both scenarios and could update the position of the nearest landing spot in real time. When the drone prepared to land, it approached a way-point vertically above the detected landing spot and subsequently slowly descended.

We conducted experimental tests on the accuracy of landing spot detection in four outdoor scenes listed in Table 1. The accuracy is the ratio of the number of grid cells detected by our method that meet the landing standards to the number of grid cells in the actual flat area. Table 1 reveals that in relatively simple scenes, our method exhibits a higher accuracy in detecting the landing spot. However, in scenes with higher complexity, where the ground details are more intricate, and the point cloud recovered by SLAM contains more noise, the detection accuracy is likely to be affected. In areas with high-density buildings, there are often many walls, obstacles, and complex structures, which may lead to more obstruction and affect the accuracy of the localization and mapping. Meanwhile, the construction of elevation maps in such scenes becomes more complex, consequently diminishing the precision of landing spot detection. In regions lacking textures, the SLAM system is unable to accurately extract features and thus unable to construct a complete point cloud map, resulting in increased localization errors and insufficient information within the map, which ultimately affects the construction of the elevation map and the accuracy of landing spot detection.

Experimental scenarios	Detection accuracy
Simple scene	95%
Multi-obstacle scene	89%
Scene with high-density buildings	41%
Texture-deficient scene	69%

Table 1. Accuracy of landing spot detection in different scenarios.

4.3 Runtime and Computational Efficiency

During operation, the elevation mapping module and landing spot selection module use one processing core. The visual sensor on the UAV captures scene videos at a frame rate of 30fps. Table 2 lists the timing measurements. On average, it takes 0.25 seconds to construct an elevation map for a frame of point cloud, and approximately 0.47 seconds to detect the landing spot in the current elevation map. However, the time consumption depends greatly on the scope and complexity of the scene, as an increase in the scene information will lead to higher computational costs for both the elevation mapping and the detection algorithm.

To study the applicability of the SLAM technology in UAV

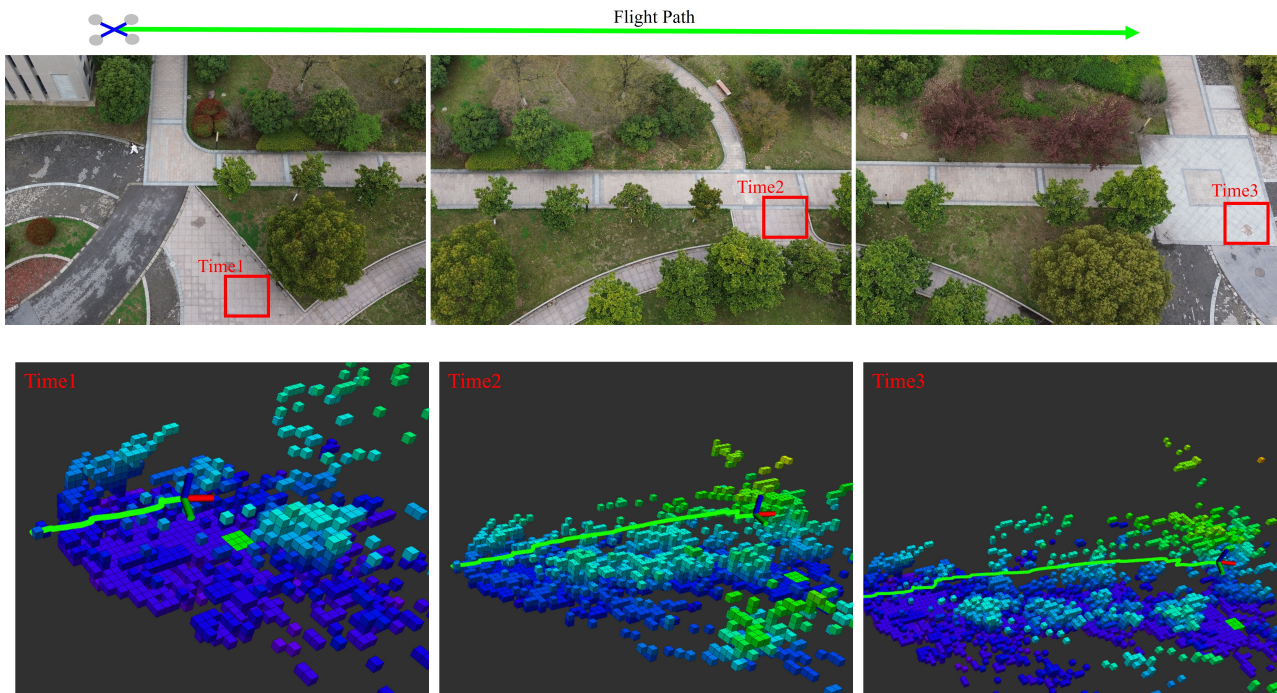


Figure 4. Elevation maps and landing spot detection results in scenario 1.

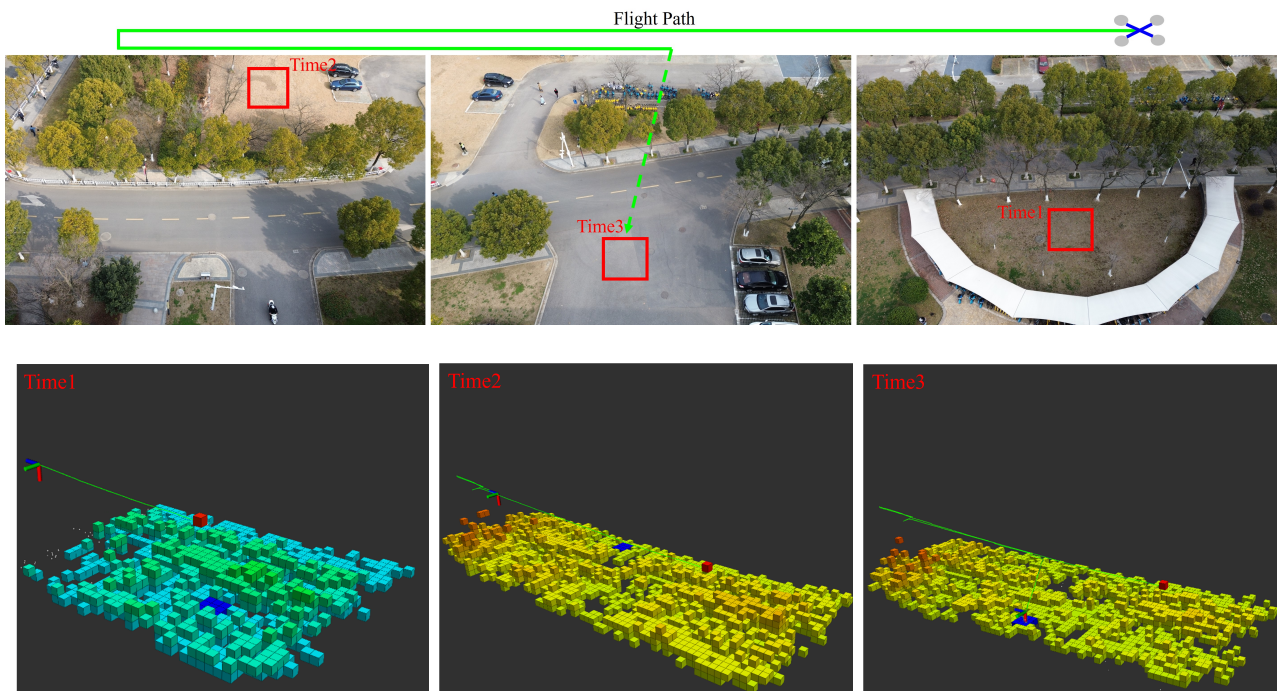


Figure 5. Elevation maps and landing spot detection results in scenario 2.

Stages	Time(s)
Elevation map establishment	0.25
Elevation map update	0.22
Landing spot detection	0.47
Landing spot update	0.27

Table 2. Timing measurements.

landing spot detection and the performance of our method in multiple scenarios, we conducted computational efficiency testing experiments in the four scenarios listed in Table 1. As shown in Table 3, areas with dense buildings have complex

structures and may contain a large number of feature points, which increases the computational complexity of feature extraction and matching in SLAM systems, thereby increasing CPU and memory usage. On the contrary, for areas lacking texture, the computational burden of feature point extraction and matching will be reduced, thus requiring fewer computational resources. For the landing spot detection module, compared to areas with simple scene structures, areas with dense buildings require more grid cells when establishing elevation maps, which increases the utilization of CPU and memory resources during elevation mapping and the subsequent selection of the

Experimental scenarios	SLAM		Landing spot detection	
	CPU (%)	Mem (%)	CPU (%)	Mem (%)
Simple scene	11.7	3.9	3.3	0.7
Multi-obstacle scene	11.9	4.0	1.0	0.6
Scene with high-density buildings	12.3	4.2	8.7	0.7
Texture-deficient scene	11.2	3.8	3.7	0.7

Table 3. Computational efficiency in various experimental scenarios

landing spot.

5. Conclusion

In summary, this paper proposes a real-time autonomous landing spot detection method for UAVs. The novelty of the method is that the scene is modeled using a voxel-based probabilistic update approach, which can overcome the problem of the sparse point cloud and represent changes in the environment in real time. In addition, a shift-box strategy is proposed to analyze terrain in the elevation map and identify a safe landing spot. Experiments show that our method has high accuracy in multiple scenarios, and possesses high real-time processing capabilities, thus indicating a significant potential for applications.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (No. 42271343) and the Fund of State Key Laboratory of Remote Sensing Information and Image Analysis Technology of Beijing Research Institute of Uranium Geology (6142A01210403).

References

Cabrera-Ponce, A. A., Martinez-Carranza, J., 2017. A vision-based approach for autonomous landing. *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, IEEE, 126–131.

Chen, J., Miao, X., Jiang, H., Chen, J., Liu, X., 2017. Identification of autonomous landing sign for unmanned aerial vehicle based on faster regions with convolutional neural network. *2017 Chinese Automation Congress (CAC)*, IEEE, 2109–2114.

Engel, J., Sturm, J., Cremers, D., 2012. Camera-based navigation of a low-cost quadcopter. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2815–2821.

Fankhauser, P., Bloesch, M., Gehring, C., Hutter, M., Siegwart, R., 2014. Robot-centric elevation mapping with uncertainty estimates. *Mobile Service Robotics*, World Scientific, 433–440.

Forster, C., Faessler, M., Fontana, F., Werlberger, M., Scaramuzza, D., 2015. Continuous on-board monocular-vision-based elevation mapping applied to autonomous landing of micro aerial vehicles. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 111–118.

Harshit, H., Kushwaha, S., Jain, K., 2022. Geometric Features Interpretation of Photogrammetric Point Cloud from Unmanned Aerial Vehicle. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 10, 83–88.

Johnson, A. E., Klumpp, A. R., Collier, J. B., Wolf, A. A., 2002. Lidar-based hazard avoidance for safe landing on Mars. *Journal of guidance, control, and dynamics*, 25(6), 1091–1099.

Kong, W., Zhou, D., Zhang, D., Zhang, J., 2014. Vision-based autonomous landing system for unmanned aerial vehicle: A survey. *2014 international conference on multisensor fusion and information integration for intelligent systems (MFI)*, IEEE, 1–8.

Laiacker, M., Kondak, K., Schwarzbach, M., Muskardin, T., 2013. Vision aided automatic landing system for fixed wing uav. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2971–2976.

Miller, A., Shah, M., Harper, D., 2008. Landing a uav on a runway using image registration. *2008 IEEE International Conference on Robotics and Automation*, IEEE, 182–187.

Mittal, M., Mohan, R., Burgard, W., Valada, A., 2019. Vision-based autonomous UAV navigation and landing for urban search and rescue. *arXiv 2019. arXiv preprint arXiv:1906.01304*.

Mur-Artal, R., Tardós, J. D., 2017. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*.

Patruño, C., Nitti, M., Petitti, A., Stella, E., D’Orazio, T., 2019. A vision-based approach for unmanned aerial vehicle landing. *Journal of Intelligent & Robotic Systems*, 95(2), 645–664.

Scaramuzza, D., Achtelik, M. C., Doitsidis, L., Friedrich, F., Kosmatopoulos, E., Martinelli, A., Achtelik, M. W., Chli, M., Chatzichristofis, S., Kneip, L. et al., 2014. Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments. *IEEE Robotics & Automation Magazine*, 21(3), 26–40.

Scherer, S., Chamberlain, L., Singh, S., 2012. Autonomous landing at unprepared sites by a full-scale helicopter. *Robotics and Autonomous Systems*, 60(12), 1545–1562.

SUO, W., HU WG, B. et al., 2020. Research on flight control of quadrotor UAV based on visual image. *Laser Technology*, 44(4), 452–458.

Xin, L., Tang, Z., Gai, W., Liu, H., 2022. Vision-based autonomous landing for the UAV: A review. *Aerospace*, 9(11), 634.

Xing, W., Zhu, L., Li, C., XU, Y., Zhao, G., Song, A., 2020. Autonomous landing site selection method for lidar-based UAVs. *Journal of Electronic Measurement and Instrumentation*.

Yang, T., Li, P., Zhang, H., Li, J., Li, Z., 2018. Monocular vision SLAM-based UAV autonomous landing in emergencies and unknown environments. *Electronics*, 7(5), 73.

Yubo, L., Haohan, B., Wenhao, L., Ying, H., 2021. Survey of uav autonomous landing based on vision processing. *Advances in Intelligent Networking and Collaborative Systems: The 12th International Conference on Intelligent Networking and Collaborative Systems (INCoS-2020) 12*, Springer, 300–311.

Zeng, Q., Luo, Y., Sun, K., Li, Y., Liu, J., 2022. Review on SLAM technology development for vision and its fusion of inertial information. *Journal of Nanjing University of Aeronautics & Astronautics/Nanjing Hangkong Hangtian Daxue Xuebao*, 54(6).