



Slope inspection under dense vegetation using LiDAR-based quadrotors

Received: 17 November 2024

Accepted: 22 July 2025

Published online: 11 August 2025



Check for updates

Wenyi Liu^{ID}¹, Yunfan Ren¹, Rui Guo¹, Vickie W. W. Kong², Anthony S. P. Hung², Fangcheng Zhu¹, Yixi Cai^{ID}¹, Huajie Wu^{ID}¹, Yuying Zou¹ & Fu Zhang^{ID}¹✉

Landslides pose significant threats to residents' safety and daily lives. To mitigate such risks, flexible debris-resisting barriers are constructed and regularly inspected, a task known as slope inspection. Traditional manual inspections are costly and difficult due to steep terrains and dense vegetation. Unmanned aerial vehicle (UAV) equipped with LiDAR and cameras offers high mobility, making them well-suited for slope inspections. However, existing UAV solutions lack comprehensive frameworks to handle dense vegetation, including robust localization, high-precision mapping, small and dynamic obstacle avoidance, and cluttered under-canopy navigation. To address these challenges, we develop a LiDAR-based quadrotor with a comprehensive software system. Our quadrotor features assisted obstacle avoidance, enabling it to autonomously avoid intricate obstacles while executing pilot commands. Field experiments conducted in collaboration with the Hong Kong Civil Engineering and Development Department demonstrate our quadrotor's ability to avoid small obstacles and maneuver in dense vegetation, validating its practical potential for slope inspection.

Landslides present significant risks to the safety and daily lives of residents. In regions characterized by coastal and mountainous terrain, such as Hong Kong, climate hazards, including extreme rainfall and typhoons, contribute to an average of approximately 300 landslides per year. Some of these events can lead to severe consequences as shown in Fig. 1a. The Geotechnical Engineering Office (GEO) of the Civil Engineering and Development Department (CEDD) is tasked by the Government of Hong Kong Special Administrative Region to manage the landslide risk for Hong Kong. The GEO has implemented a Slope Safety System¹ to achieve the mandate given. One of the key strategies in the Slope Safety System is to adopt an engineering approach to stabilize existing substandard man-made slopes and mitigate landslide risk from natural hillsides. As regards the latter, it has been the practice to erect flexible debris-resisting barriers at strategic locations to intercept landslide debris coming from uphill and hence protect the public and infrastructure downhill, as shown in Fig. 1b.

These barriers, however, are challenging to access due to their mid-slope placement and environmental changes over time (e.g., the growth of vegetation and trees). Therefore, regular inspections are important to ascertain the proper functioning of these barriers, particularly steel flexible barriers, a task known as slope inspection. Any anomalies, such as accumulation of debris behind barriers (Fig. 1c1) or severe corrosion of the steel components, may call for subsequent maintenance and repair works. These inspections encompass a comprehensive assessment of various aspects related to the flexible debris-resisting barriers, including the steel components, such as the wire ropes positioned atop the barriers (Fig. 1c2), the inclined wire ropes (Fig. 1c3), and the supporting posts (Fig. 1c4), to ensure the stability of the barriers. Diligently conducting these inspections ensures that the flexible barrier can effectively mitigate the impact of debris when it becomes necessary.

Currently, the maintenance agent of the debris-resisting barrier employs a manual inspection approach. Maintenance access is constructed along the flexible debris-resisting barriers on the slope to

¹Department of Mechanical Engineering, The University of Hong Kong, Hong Kong, China. ²Geotechnical Engineering Office, Civil and Engineering Development Department, The Government of Hong Kong SAR, Hong Kong, China. e-mail: fuzhang@hku.hk

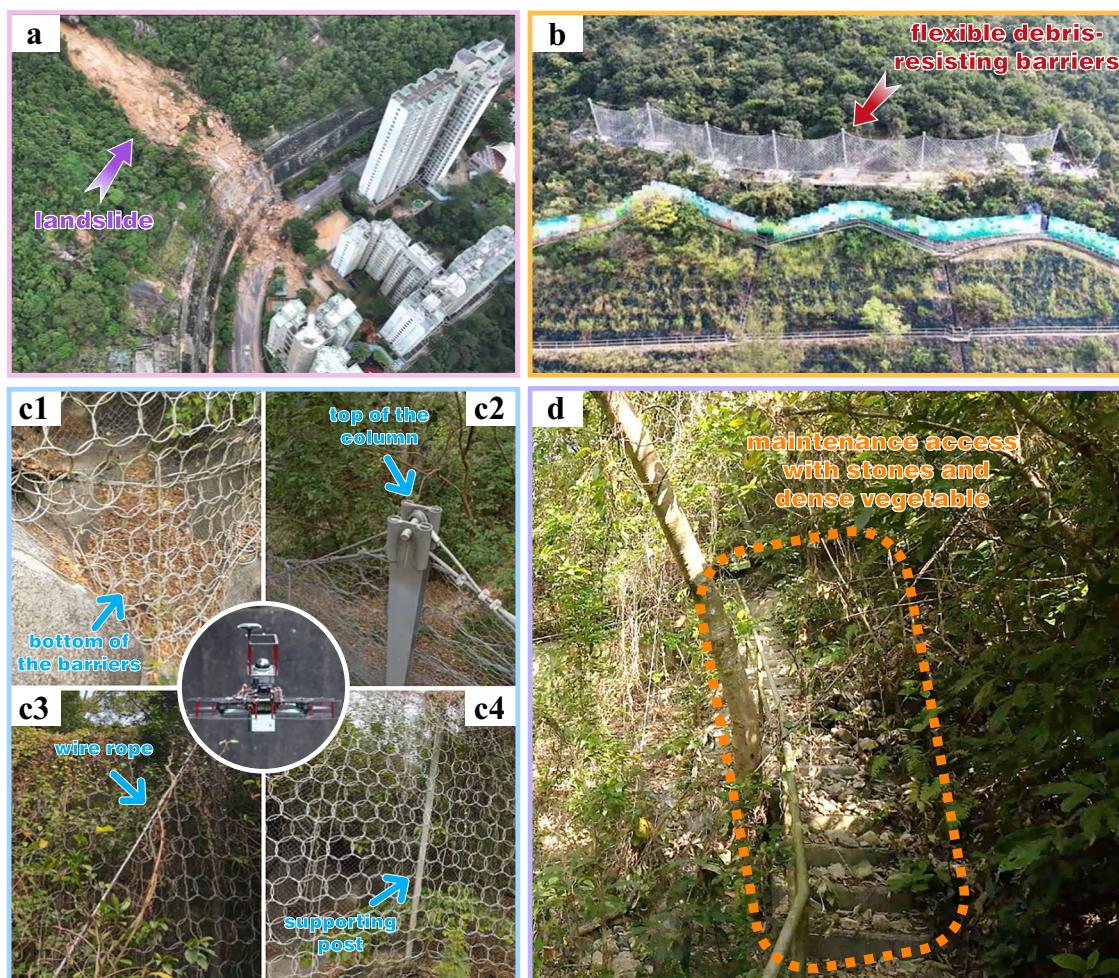


Fig. 1 | The environments and inspection targets of slope inspection. **a** A slope in Hong Kong that suffered a significant landslide in September 2023³¹ due to Typhoon Haikui³⁰. **b** Flexible debris-resisting barriers constructed on a slope to

stop landslides. **c** Inspection targets of the flexible debris-resisting barriers. **d** Inspection maintenance access with stones and dense vegetation.

provide safe access for the inspecting personnel. However, the location of these barriers necessitates the construction of long maintenance access to access these barriers from public roads. As shown in Fig. 1d, the lengthy maintenance access, combined with the steep terrain, leads to high construction costs and imposes an increased workload on the inspecting personnel. Additionally, dense vegetation and hostile environments (e.g., steep rugged stairs, insects) are not suitable for inspecting personnel to stay. Furthermore, the maintenance access could be misused by unauthorized personnel to access nearby residential areas, which have raised considerable concerns among the residents.

Compared to manual inspection, unmanned aerial vehicles (UAVs)-based inspection possesses unique advantages and potential applications in slope inspections. Firstly, UAVs offer the capability to navigate complex terrains, covering wide areas and accessing areas that are difficult for humans or ground mobile robots to reach. They also provide rich terrain and obstacle data through onboard sensors such as LiDAR and cameras. Additionally, the deployment of UAVs reduces the need for human labor, mitigates operational risks, and enhances work efficiency. Moreover, the utilization of UAVs can potentially eliminate the necessity for access construction and maintenance, resulting in substantial cost savings and reduced criminal activities.

However, applying UAVs to slope inspections in dense vegetation presents several challenges. Firstly, for platform, the mechanical

structure of the UAV must be systematically designed, considering factors such as maneuverability in narrow areas, flight time, and the types of onboard sensors required for effective inspections. Secondly, for localization, traditional GPS-based methods become unreliable under the canopy. UAVs need to rely on onboard sensors such as LiDAR and cameras to achieve accurate localization. Thirdly, for mapping, it is necessary to use data from onboard sensors to rapidly and accurately update information about thin objects in the map, such as thin tree branches, wire ropes, and fine nets on barriers. Finally, for planning and control, UAVs must effectively mitigate wind disturbances while responding smoothly and quickly to avoid dynamic objects, enabling safe navigation amidst swaying tree branches and other disturbances in the wild.

Many works^{2–9} deploy UAVs for inspection tasks on structures such as bridges, power lines, and buildings. These UAVs are capable of conducting inspections in areas away from obstacles. Limitations such as the larger aircraft size or the lack of comprehensive navigation algorithms render these approaches unsuitable for narrow-area inspection. Some studies and commercial UAVs incorporate additional mechanical structures to access narrow areas, such as protective enclosures around the propellers^{10–14}, foldable arms^{15–19}, or robotic arms equipped with cameras^{20–22}. However, UAVs with such mechanisms generally lack obstacle avoidance capabilities, making them impractical for slope inspection under dense vegetation. Studies in refs. 23–27 attempt to control UAVs through narrow gaps by

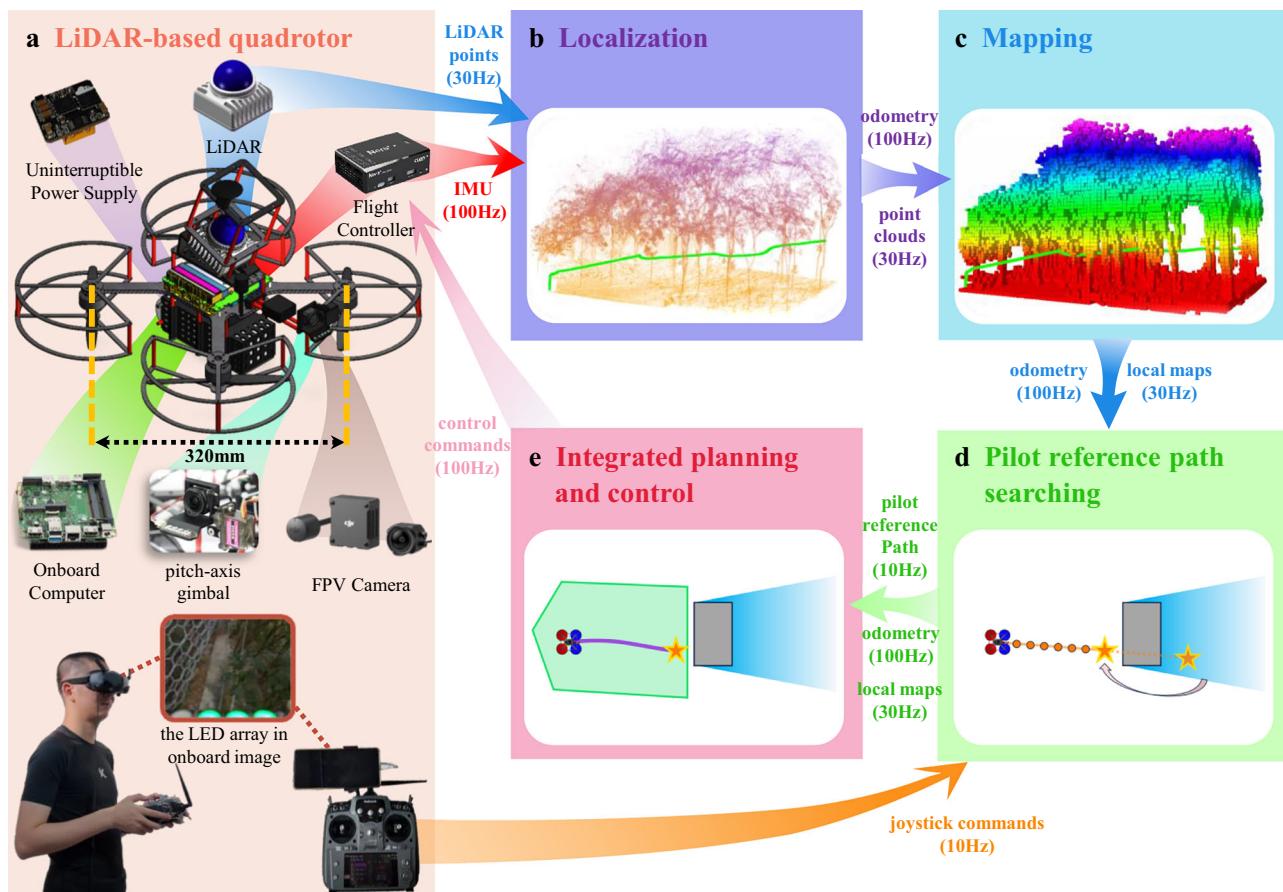


Fig. 2 | System overview of our LiDAR-based quadrotor for slope inspection. **a** The detailed hardware configuration of our LiDAR-based quadrotor. **b** The localization module employs Fast-LIO2, which utilizes an iterative error state Kalman filter and incremental kd-Tree. **c** The mapping module constructs a robocentric occupancy grid map, ROG-Map, with improvements tailored for slope

environments. **d** The pilot reference path searching generates a coarse reference path from the pilot's joystick commands. **e** Our quadrotor adopts an integrated planning and control framework (IPC), which directly generates control commands (e.g., angular velocity references and throttle) from the coarse reference path by solving a model predictive control (MPC) problem.

whole-body motion planning (i.e., SE(3) planning), but in dense vegetation, the complexity and irregularity of obstacles make it challenging to optimize SE(3) trajectories. Currently, commercial UAVs such as DJI Mavic 3²⁸ and Skydio 2 plus²⁹, equipped with multiple sensors and employing navigation algorithms for obstacle avoidance in sparser environments, demonstrate practicality. However, the stereo disparity method used by the DJI Mavic 3 is susceptible to lighting changes, while the learning-based approach of the Skydio 2 plus suffers from data scarcity in extreme environments such as dense vegetation, resulting in lower accuracy of visual depth measurements compared to LiDAR for slope inspection. In addition to the low accuracy of these visual measurements, their obstacle avoidance functionality restricts UAVs from flying in narrow areas due to the paramount concern for safety. As far as we know, no existing research or commercial UAV has presented an effective solution that can be applied for slope inspection under dense vegetation.

To address the challenges of slope inspection under dense vegetation, we develop a LiDAR-based quadrotor with a comprehensive software system, as shown in Fig. 2 (Supplementary Movie 1). The quadrotor's design fully considers the requirements for slope inspection, incorporating a high-performance navigation system that enables the quadrotor to maneuver through narrow spaces and agilely avoid small obstacles such as thin branches and ropes. An onboard high-definition digital video transmission system delivers a low-latency image feed with a delay of only 40 ms, providing timely feedback for the pilot on inspection targets. Given that slopes in Hong Kong currently lack digital models, predefining inspection targets for

autonomous inspection is impossible. Therefore, our quadrotor operates in a human-in-the-loop mode. In this mode, the pilot determines the inspection targets impromptu from the transmitted video and commands the quadrotor's high-level flight direction via the joysticks on the remote controller. On the other hand, the quadrotor navigation system attempts to fly along the specified direction meanwhile avoiding obstacles on the way. This mode enables the pilot to focus solely on the inspection task without concerns about avoiding obstacles, thus enhancing inspection efficiency. During the inspection process, our quadrotor produces a high-accuracy 3D point cloud reconstruction of the environments in real-time. The 3D models are instrumental in constructing digital assets and facilitating further data analysis.

To assess the feasibility of our system solution for slope inspection, we first conduct functional tests on our quadrotor in non-operational scenarios (Supplementary Movie 2). Subsequently, invited by CEDD for visual inspection of flexible debris-resisting barriers and other geotechnical features, we deploy our quadrotor in six field environments (Supplementary Movies 3 and 4), including five flexible debris-resisting barriers located in dense vegetation and one slope that experienced a landslide caused by a rainstorm. In all these six field experiments, our quadrotor effectively accomplishes the assigned inspection tasks. Additionally, benchmark comparisons with DJI Mavic 3 regarding assisted obstacle avoidance flight (Supplementary Movie 5), demonstrate that our LiDAR-based quadrotor outperforms the most advanced commercial UAVs. Particularly, our quadrotor can avoid small obstacles on actual slopes and navigate safely through

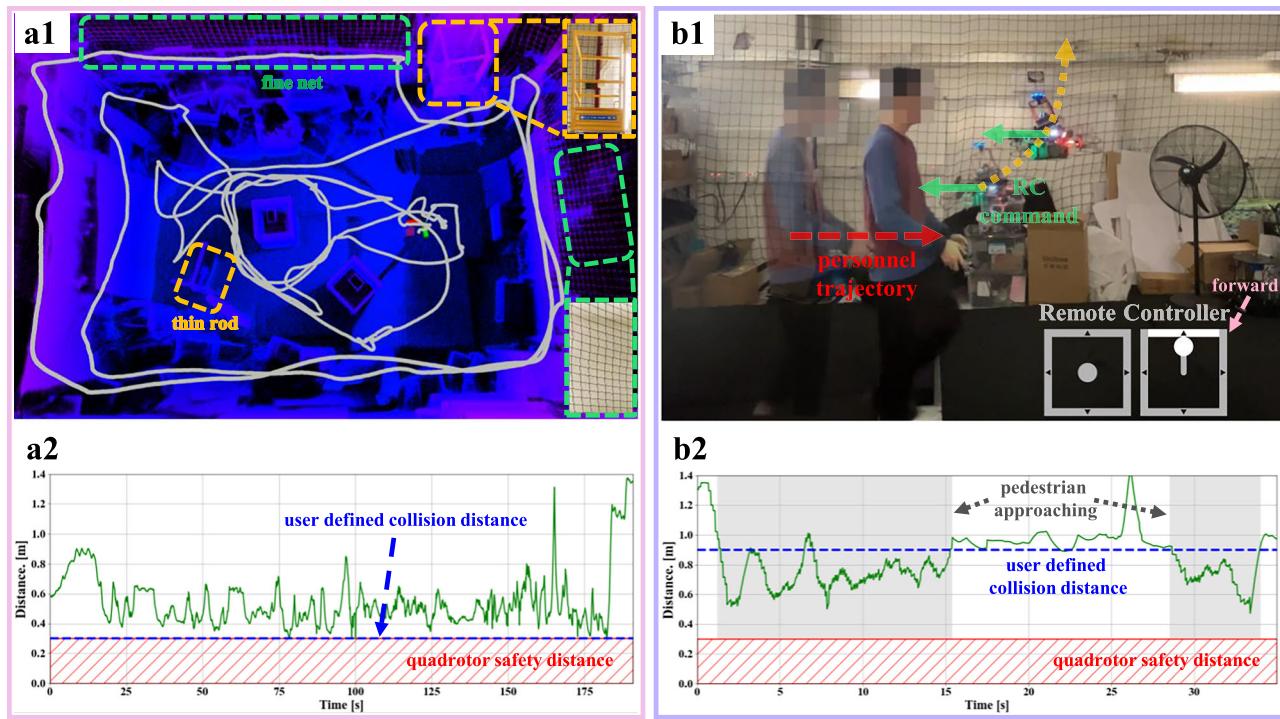


Fig. 3 | Functional tests of the assisted obstacle avoidance function in non-operational scenarios. **a** Assisted obstacle avoidance flight in a narrow environment. In a1, the gray curve represents the quadrotor's flight trajectory, which is about 99.46 m long. The green box represents the fine nets, and the orange box represents a scaffold with thin rods. In a2, the distance between the quadrotor's center and the nearest obstacles is greater than the quadrotor's size during the

flight. **b** The third person view of the assisted obstacle avoidance flight in dynamic environment. In b1, as the personnel approaches the quadrotor, the quadrotor moves away from the personnel to avoid collisions, despite adverse joystick commands. In b2, as the personnel approaches the quadrotor, the distance between the quadrotor's center and the nearest obstacles remains constantly greater than the quadrotor's size throughout the flight.

dense vegetation. As a result, this work represents a significant milestone in transitioning LiDAR-based quadrotors from laboratory settings to real-world applications, pioneering a new standard for UAV-based inspection in hazardous and hard-to-access environments.

Results

We demonstrate the assisted obstacle avoidance capabilities of our LiDAR-based quadrotor through extensive real-world experiments. First, we evaluate its performance in non-operational scenarios, confirming that our quadrotor can navigate safely through narrow spaces and avoid dynamic objects. Subsequently, our quadrotor is deployed to six actual slopes with dense vegetation in Hong Kong. Throughout all six field experiments, our quadrotor successfully performs close-up photo inspections of flexible debris-resisting barriers while navigating safely in complex environments. Finally, we conduct comparative experiments in the field environments with DJI Mavic 3 to further showcase the assisted obstacle avoidance capabilities of our quadrotor.

Functional tests in non-operational scenarios

We conduct functional tests of the quadrotor's assisted obstacle avoidance function in two typical non-operational scenarios, namely the narrow environment and the dynamic environment.

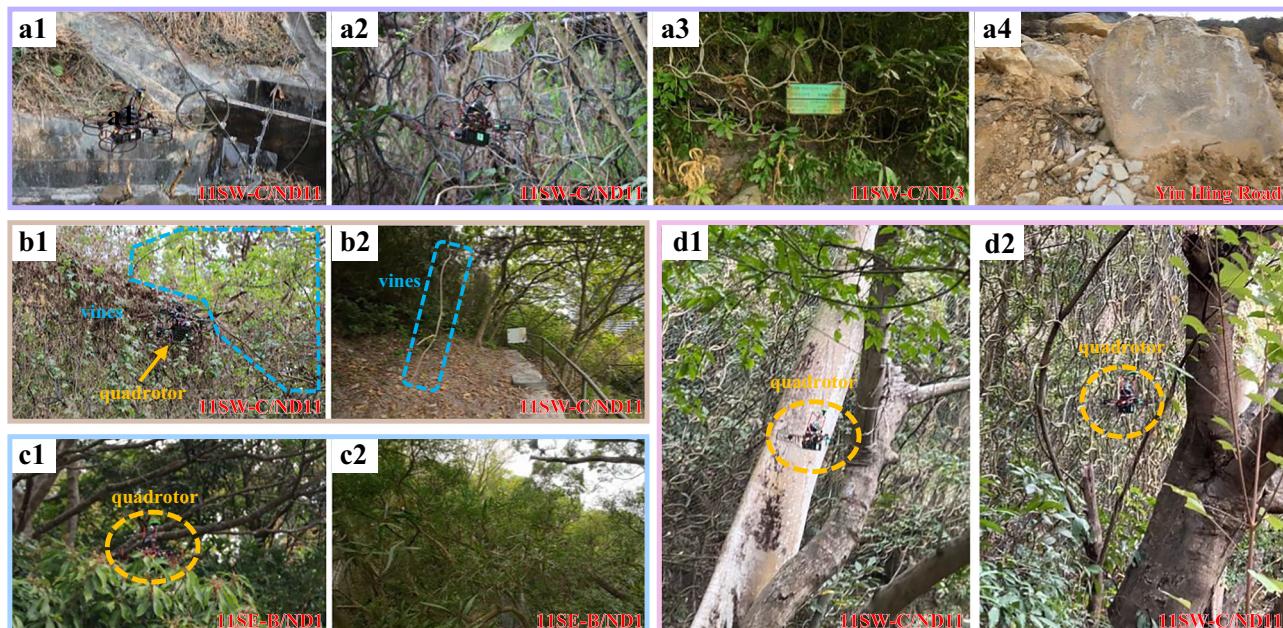
In the narrow environment, the surroundings are enclosed by fine nets, with several boxes and thin rods in the middle, as shown in Fig. 3a1. To enhance the ability to navigate through narrow areas, we set the user-defined obstacle avoidance distance (represented by the blue line in Fig. 3a2) equal to the quadrotor's size. As shown in Fig. 3a2, the distance d_{min} between the quadrotor's center and the nearest obstacles remains greater than the quadrotor's size, suggesting a successful obstacle avoidance throughout the whole test. The

achievement of obstacle avoidance is primarily attributed to the robustness and effectiveness of the navigation algorithm (see "Methods"). The mapping module constructs high-resolution occupancy maps, effectively updating the corresponding grid cells in the map with small objects (e.g., fine nets or thin rods) detected by the LiDAR scans. Additionally, the pilot reference path searching module efficiently generates a collision-free reference path based on joystick commands. Subsequently, the integrated planning and control (IPC) module achieves high-precision control, ensuring the quadrotor safely arrives at the target point.

Next, we evaluate the assisted obstacle avoidance function in dynamic environment, as illustrated in Fig. 3b1, where the personnel and the quadrotor move towards each other. Under the assisted obstacle avoidance function, the quadrotor ignores the remote control commands directing it towards the personnel and instead maneuvers to avoid him. In this experiment, the user-defined obstacle avoidance distance is set to 0.9 m to provide the quadrotor with a larger clearance for reacting and avoiding dynamic objects. As depicted in Fig. 3b2, although the motion of the dynamic object leads to the distance between the quadrotor and the obstacle being smaller than the user-defined obstacle avoidance distance, the distance remained bigger than the quadrotor safety distance. This is achieved by the low system latency of our navigation algorithm (see "Methods"). Upon receiving a LiDAR scan, the mapping module promptly updates the obstacles in the probability and inflated map. Subsequently, the pilot reference path searching module generates a new local goal in the Known Free region and a reference path reaching the new goal. This reference path is then leveraged by the IPC module to enable the quadrotor to rapidly respond and navigate towards a safe area. By successfully conducting these two test scenarios, we validate the feasibility of the quadrotor's assisted obstacle avoidance function.

Table 1 | Flight data from six field experiments

Experiment No.	Flexible Barrier No. as referred by CEDD	Location	Trajectory Length (m)	Flight Time (min:sec)
1	11SW-C/ND3	Victoria Road, Pokfulam	133.49	7:30
2	11SW-C/ND6	Victoria Road, Pokfulam	79.31	3:43
3	-	Yiu Hing Road	425.99	6:55
4	11SE-B/ND1	Lei Yue Mun Estate	274.68	8:28
5	11SE-B/ND2	Lei Yue Mun Estate	154.34	5:24
6(part 1)	11SW-C/ND11	Victoria Road, Pokfulam	186.18	8:25
6(part 2)	11SW-C/ND11	Victoria Road, Pokfulam	227.20	8:28

**Fig. 4 | Our quadrotor performs slope inspection in field environments.** The text in the lower-right corner of the image corresponds to the experimental location in Table 1. **a** Quadrotor performing close-range inspection of flexible debris-resisting

barriers (a1–a3) and stones caused by landslides (a4). **b** Quadrotor avoiding thin dropping vines. **c** Quadrotor maneuvering through dense tree canopies. **d** Quadrotor navigating through narrow tree branches.

We invite readers to watch Supplementary Movie 2, to get a more intuitive understanding of the functional tests of our quadrotor in non-operational scenarios.

Field experiments in slope scenarios

To validate our system's suitability for slope inspection under dense vegetation environments, we collaborate with the CEDD to deploy the quadrotor in the field and utilize the onboard camera for close-range visual inspection of the barriers. We conduct experiments at six different locations, including five slopes covered with dense vegetation and one slope recently experienced a landslide as a result of the 2023 Hong Kong rainstorm and floods caused by the landfall of Typhoon Haikui³⁰. The six field experiments are summarized in Table 1. The third field experiment, conducted next to the Yiu Hing Road, involves a slope that has just experienced a landslide³¹, where the quadrotor conducts close observation of the barriers and stones in a relatively wide space. The other five field experiments consist of slopes with dense vegetation, where the quadrotor performs close-range photographic inspection of the flexible debris-resisting barriers in narrow spaces.

With the assistance of obstacle avoidance functionality, our quadrotor assists the pilot in close-range visual inspection of flexible debris-resisting barriers and stones resulting from landslides (Fig. 4a), maneuvering through dense tree canopies (Fig. 4b), avoiding thin

dropping vines (Fig. 4c), and navigating through narrow tree branches (Fig. 4d). Ultimately, our quadrotor completes all six field experiments, demonstrating its suitability for slope inspection in dense vegetation environments. The flight trajectory of each experiment is superimposed on the slope drawing provided by the CEDD, the point cloud map built online by our navigation system, and the example photos taken during the inspection are shown in Fig. 5. These outputs provide the Hong Kong CEDD with detailed and valuable information about the inspected areas, enabling thorough analysis and assessment of the flexible debris-resisting barriers' condition.

Throughout these six field experiments, the quadrotor benefited from the accurate perception of thin objects provided by the LiDAR sensor. This accurate perception allows our quadrotor to construct high-resolution local occupancy grid maps (map size: $10 \times 10 \times 10$ m, resolution: 0.05 m) in the mapping module and update the occupancy status of grid cells containing moving branches and other thin objects. Moreover, thanks to the assisted obstacle avoidance function, our quadrotor is capable of safely approaching inspection targets, such as flexible debris-resisting barriers, at distances as small as the quadrotor's size. This enables the quadrotor to capture high-definition images for subsequent detailed analysis. Additionally, despite encountering varying degrees of natural wind disturbances (average wind speed: 4.17 m/s), the quadrotor effectively suppresses these disturbances without compromising flight performance, due to

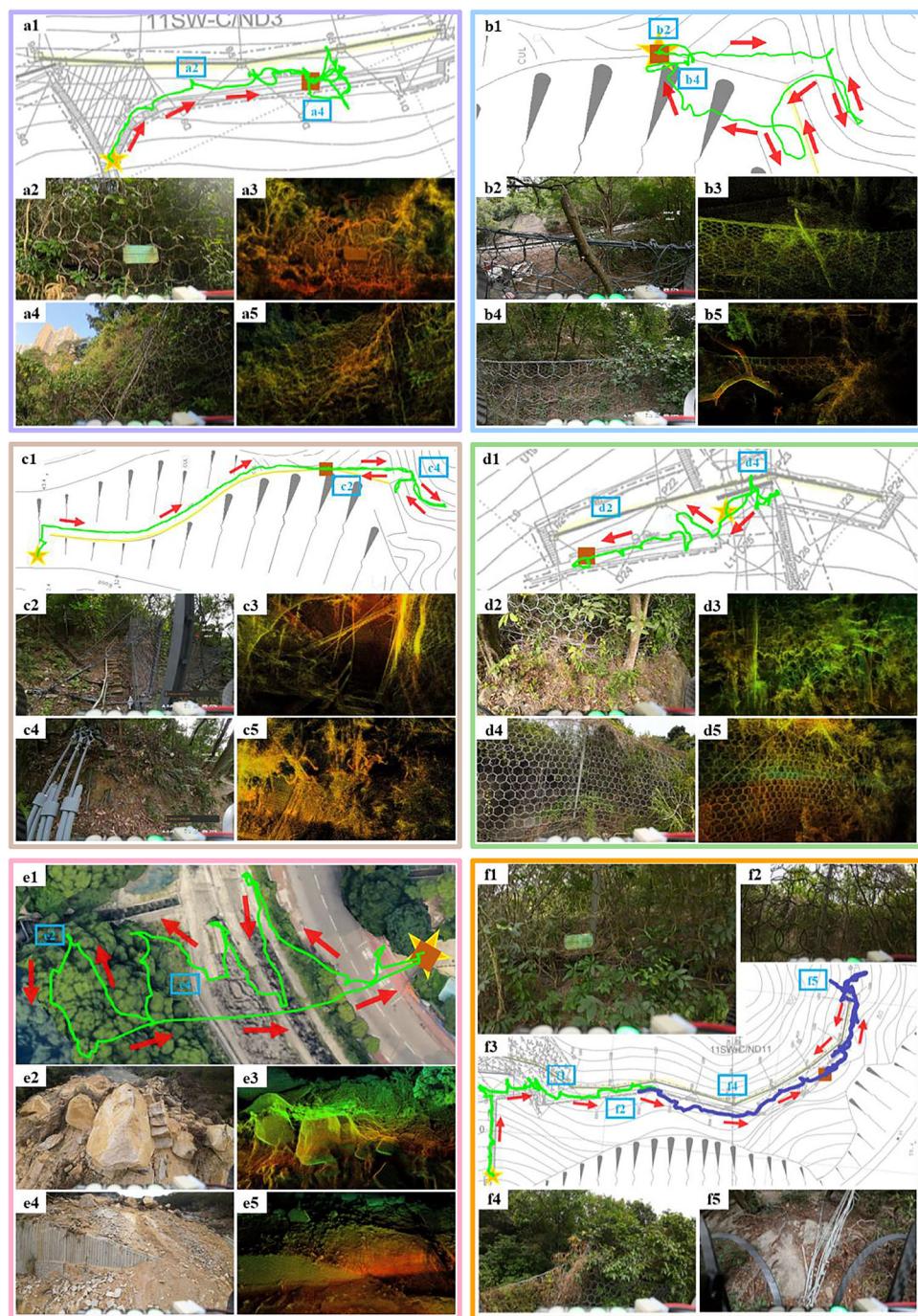


Fig. 5 | Flight data of our LiDAR-based quadrotor during slope inspection. The green and purple curve represents the flight trajectory of the quadrotor, the yellow star represents the take-off point, and the brown box represents the landing point. **a** The field experiment of 11SW-C/ND3 slope next to Victoria Road, Pokfulam. **b** The field experiment of 11SE-B/ND2 slope next to Lei Yue Mun Estate. **c** The field

experiment of 11SE-B/ND1 slope next to Lei Yue Mun Estate. **d** The field experiment of 11SW-C/ND6 slope next to Victoria Road, Pokfulam. **e** The field experiment of a slope next to Yiu Hing Road. **f** The field experiment of 11SW-C/ND11 slope next to Victoria Road, Pokfulam.

the real-time generation of optimal control actions by the IPC module.

To gain a more comprehensive understanding of our quadrotor's real-world performance in field experiments, we invite readers to watch Supplementary Movies 3 and 4. The movie provides a detailed showcase of our quadrotor's performance during the field experiment of the 11SW-C/DN11 slope next to Victoria Road, Pokfulam. Moreover, we present first-person videos recorded during the other five field experiments.

Benchmark with commercial drones

DJI Mavic 3²⁸, as one of the most advanced commercial drones, is equipped with up to eight wide-angle cameras and incorporates the advanced pilot assistance system (APAS 5.0) algorithm for high-level flight assistance, enabling omnidirectional obstacle sensing. It offers three obstacle avoidance modes: Brake, Normal Bypass, and Nifty Bypass, allowing pilots to customize the settings based on the environments and their preferences. In Brake mode, the DJI Mavic 3 comes to an immediate stop if in the flight direction an obstacle is detected. In

Normal Bypass and Nifty Bypass modes, the DJI Mavic 3 can bypass obstacles, but in Nifty Bypass mode, it maintains a smaller clearance with obstacles, so possessing a higher passability but also a higher risk of collision. In sum, among these three modes, the Brake mode provides the highest level of flight safety, followed by the Normal Bypass mode, and the Nifty Bypass mode performs the least effectively. However, in terms of accessibility in narrow areas, the order is reversed, with Nifty Bypass mode performing the best, Normal Bypass mode being in the middle, and Brake mode performing the worst.

To further validate the suitability of our quadrotor for slope inspection in dense vegetation, we conduct experiments on the 11SE-B/ND1 slope next to Lei Yue Mun Estate to compare its obstacle avoidance function with DJI Mavic 3 in different modes. When operating in Normal Bypass modes, DJI Mavic 3 exhibits highly conservative behavior in dense vegetation environments, as shown in Fig. 6a. In such an environment, despite a clear feasible corridor in the commanded direction, the DJI Mavic 3 prioritizes safety by maintaining a hover in place. The Brake mode behaves similarly to the Normal Bypass mode, as it remains hovering too. While the Brake mode prioritizes the most on safety thus being highly conservative, our testing revealed that when flying towards fine nets, it failed to execute the necessary stop maneuver, resulting in a collision, as shown in Fig. 6b1. This deficiency can be attributed to its limited perception capabilities, particularly when dealing with thin objects. In contrast, as depicted in Fig. 6b3 and Fig. 6b4, our LiDAR-based quadrotor effectively perceives the presence of fine nets ahead, actively refusing to follow joystick commands to fly towards them, thus ensuring flight safety.

In Nifty Bypass mode, DJI Mavic 3 can fly in relatively open areas on maintenance access. However, it should be noted that the flight safety in this mode is the worst, as the rotor blades are prone to collide with tree leaves and thin branches, as shown in Fig. 6c1. Besides, DJI Mavic 3 can easily give the pilot a false feeling of loss of control. For example, it often refuses to follow the pilot's commands when in proximity to the inspection targets, while exhibits unexpected large maneuvers and long flight distances when otherwise. Moreover, despite Nifty Bypass mode being the most aggressive, the pilot still lacks control when navigating narrow areas where the available space is less than twice the size of the DJI Mavic 3. Moreover, DJI Mavic 3 is even more susceptible to collisions and crashes while operating in this mode. When directed by the pilot to fly in the forward downward region (Fig. 6c2, it fails to detect and avoid the wire rope shown in Fig. 6c3, resulting in a collision depicted in Fig. 6c4).

Overall, in dense vegetation environments, the Brake mode and Normal Bypass mode of the DJI Mavic 3 prove to have insufficient ability to navigate through dense crowded vegetation environments and encounter frequent immediate stops in the presence of clear flight passage. Moreover, even in the most conservative Brake mode, the DJI Mavic 3 fails to perceive fine nets, resulting in collisions. The Nifty Bypass mode of the DJI Mavic 3 has improved passability in dense vegetation, but at the cost of much lower safety level, often leading to collisions with small objects, such as tree leaves, tree branches, and wire ropes. On the other hand, our LiDAR-based quadrotor performs exceptionally well in the same test scenario at the 11SE-B/ND1. As shown in Fig. 6d, our quadrotor maneuvers agilely through narrow areas while still roughly following the pilot's commands. Furthermore, our quadrotor effectively avoids collisions with small objects such as thin tree branches (Fig. 6d3), successfully executing the necessary stop maneuvers when encountering fine nets (Fig. 6d4). The reliable performance of our quadrotor is a result of the carefully engineered system and its modules. First, thanks to the high-precision measurement capabilities of LiDAR and the high-resolution (0.05 m) of the mapping module, our quadrotor can effectively perceive fine nets and classify them as occupied within the probability map. Second, the pilot reference path searching

module effectively generates collision-free reference paths based on joystick commands, thus avoiding collisions with small obstacles (e.g., thin tree branches and fine nets) in the environment. Third, the high-precision control and low-latency characteristics of the IPC module (which generates angular velocity and throttle control commands with a delay of less than 3.5 ms) enable our quadrotor to swiftly and safely reach the target point.

Finally, in terms of dynamic obstacle avoidance, as shown in Fig. 6e, in Nifty Bypass mode, the DJI Mavic 3 is able to perceive dynamic obstacle ahead but fails to avoid it, further reducing its safety assurance. In comparison, our quadrotor can successfully evade slow-moving dynamic objects, as shown in Fig. 3b1. This performance advantage is primarily attributed to the pilot reference path search and the IPC modules. The pilot reference path search actively generates a reference path free of obstacles in less than 1 ms, while the IPC module computes control actions in real-time at a high frequency of 100 Hz, enabling the quadrotor to respond rapidly to dynamic obstacles. The complete comparison experiments between our quadrotor and the DJI Mavic 3 can be found in the Supplementary Movie 5.

Discussion

A key advantage of our LiDAR-based quadrotor lies in its superior obstacle avoidance capabilities, enabling it to avoid various obstacles in slope environments, including thin tree branches, wire ropes, and fine nets on barriers. Notably, our quadrotor can detect fine nets composed of 4 and 10 mm metal wires within 3.53 m and detect 16 mm wire ropes within 7.10 m. This feature establishes a foundation for human-in-the-loop operations. In slope inspection under dense vegetation, where prior maps are often unavailable, the human-in-the-loop mode enables pilots to make real-time decisions and direct the quadrotor toward specific targets while it autonomously avoids obstacles during flight. This mode provides a high degree of flexibility and adaptability, making it suitable for inspection tasks in unknown and unstructured environments, particularly in scenarios where inspection targets must be decided impromptu, such as post-landslide slope inspection.

Despite these promising results, some challenges remain. Our quadrotor's reliance on the pilot reference path, while enhancing flexibility, continues to limit its scalability in large-scale inspection. Future work could incorporate machine learning algorithms to facilitate inspection target detection and task execution, aiming to reduce pilot efforts. Moreover, while the system performed well under normal wind disturbances (wind speed less than 5.56 m/s), its stability under extreme weather conditions, such as strong winds or heavy rain, requires further validation. Since these conditions are common in landslide-prone areas, enhancing the quadrotor's resilience to adverse environments is imperative.

In conclusion, our LiDAR-based quadrotor offers an efficient and reliable solution for slope inspection under dense vegetation. Its exceptional obstacle avoidance capabilities in cluttered environments and flexible human-in-the-loop inspection mode also hold significant potential for broader applications, such as agriculture³², infrastructure development³³, industrial inspections⁷, and search and rescue operations^{34,35}. Moving forward, we aim to increase the system's automation, utilizing the identified inspection target points and point cloud models from initial human-in-the-loop flights to support subsequent automated inspections. In this enhanced mode, our quadrotor would autonomously navigate to each inspection target point within the same environment, thereby reducing the operational burden on the pilot. We believe that with further automated operation, this LiDAR-based quadrotor will become increasingly effective in supporting inspection tasks across complex, large-scale environments.

Methods

The system overview of our LiDAR-based quadrotor for slope inspection is shown in Fig. 2. For platform, considering that LiDAR

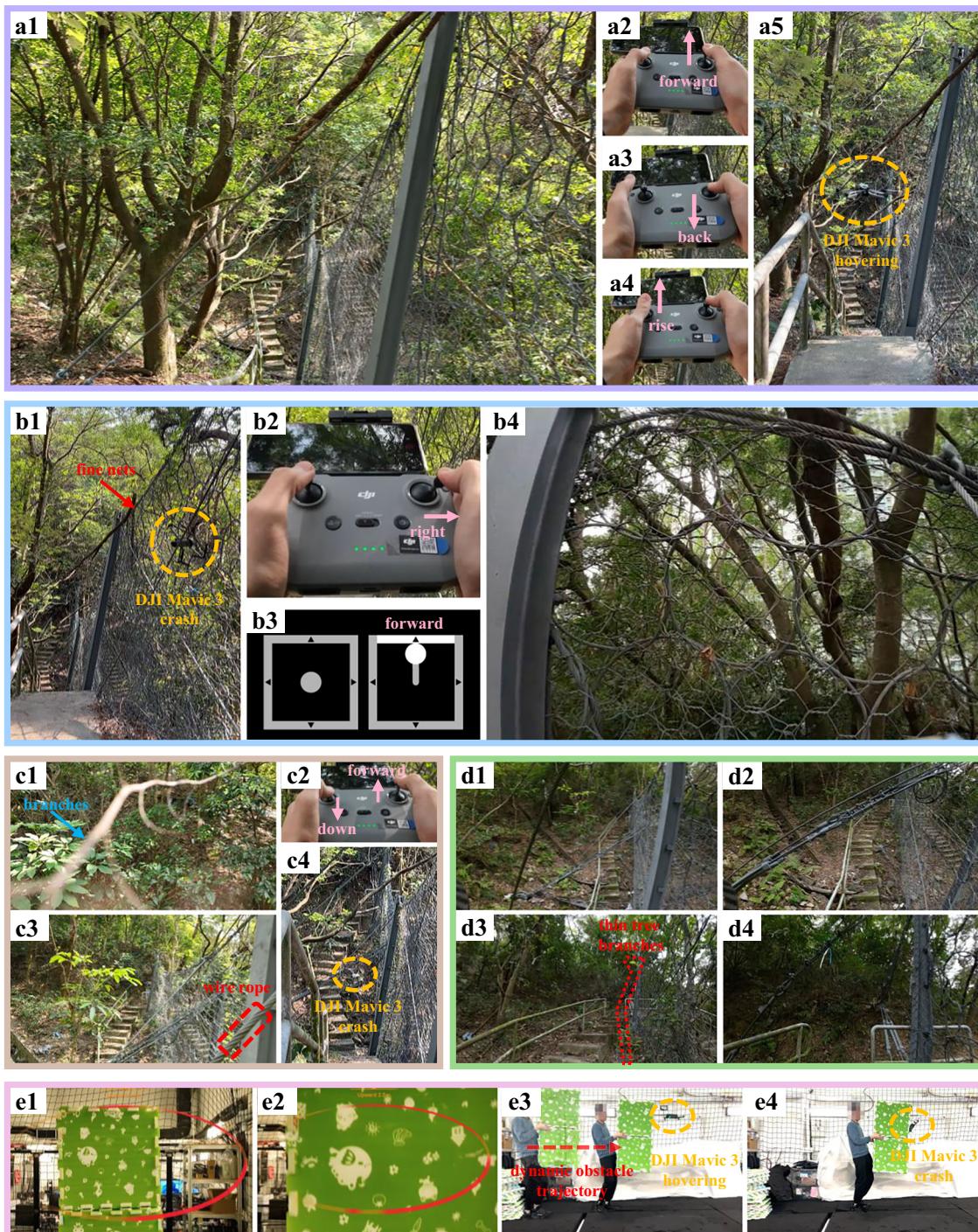


Fig. 6 | Benchmark with DJI Mavic 3. **a** DJI Mavic 3 exhibiting conservative behavior in Normal Bypass mode. Similar behavior is observed in Brake mode within the same environment. Despite a clear feasible corridor in the commanded direction (a1), the DJI Mavic 3 fails to follow the pilot's commands (a2–a4) and remains hovering in place (a5). **b** Comparison of obstacle avoidance between the DJI Mavic 3 and our quadrotor when encountering fine nets. DJI Mavic 3 collides with the fine nets (b1) under the right command (b2). However, our quadrotor

maintains a safe distance from the fine nets (b4) under the forward command (b3). **c** Obstacle avoidance performance of DJI Mavic 3 in Nifty Bypass mode. In this mode, it is easy to collide with branches (c1). When the pilot controls DJI Mavic 3 to fly forward and downward (c2), it collides with the wire rope (c3) and crashes (c4). **d** First-person view pictures taken by our quadrotor when inspecting the 11SE-B/ND1 slope next to Lei Yue Mun Estate. **e** In Nifty Bypass mode, DJI Mavic 3, despite its ability to perceive dynamic objects, cannot avoid them, resulting in crashes.

can directly provide high-precision 3D measurements compared to visual sensors, we developed a LiDAR-based quadrotor. For localization, we employ our previous work, FAST-LIO2³⁶, a robust LiDAR-inertial odometry framework capable of operating in GPS-denied environments. For mapping, we enhance ROG-Map³⁷, which constructs real-time high-resolution sliding occupancy grid maps.

By incorporating three enhancement techniques, namely the unknown grid cells inflation, infinite points ray casting, and incremental frontier update, our quadrotor can effectively identify the correct grid occupancy states in case of no LiDAR returns or with returns caused by thin objects, allowing for efficient avoidance of potential obstacles in dense vegetation. For path

searching, we search a pilot reference path consisting of two segments, ensuring alignment with the pilot's intent while avoiding static and dynamic obstacles. For planning and control, we employ the IPC framework³⁸, enhancing safety in dense vegetation by generating safe flight corridor (SFC) exclusively within Known Free space. To the best of our knowledge, we are the first to propose a practical and comprehensive UAV solution for slope inspection under dense vegetation.

The authors affirm that human research participants provided informed consent for publication of the images in Figs. 2, 3, and 6.

LiDAR-based quadrotor

The detailed hardware configuration of our LiDAR-based quadrotor is shown in Fig. 2a. Given that LiDAR, unlike GPS and cameras, can operate independently of external devices and remains unaffected by environmental conditions, our quadrotor utilizes the Livox Mid-360 LiDAR³⁹, weighing only 265 g, as its primary sensor. This LiDAR employs a non-repetitive scanning approach, accumulating data over time to generate dense point clouds. Additionally, its 360-degree horizontal and 59-degree vertical fields of view enable a wide range of scene perception, enhancing the robustness of the state estimation. For computation, we employ the Intel NUC minicomputer equipped with an Intel i7-1260P CPU (operating at 4.7 GHz), providing substantial computational power for real-time processing tasks. To enable real-time observation of the flight process by the pilot and capture photos and videos of specific areas, we use an FPV camera DJI O3 Air Unit⁴⁰, a high-definition digital video transmission system, with the DJI Goggles 2⁴¹. This video transmission system offers an impressively low-latency of 40 ms, providing timely feedback on the surrounding environment for the pilot. Moreover, it supports recording 4 K videos at a high frame rate of 120 Hz, enabling in-depth post-analysis after the flights. In order to enlarge the field of view (FOV) of the FPV camera, so as to observe both the top and bottom of the flexible debris-resisting barriers on the slope, we install the camera on a pitch-axis gimbal. The pitch angle of the gimbal is commanded by the remote controller in real-time during the flight. We use an LED array to keep track of the quadrotor's battery level in real-time. The LED array consists of four LEDs, indicating the battery percentage. They display green for battery percentage over 40%, red for battery percentage between 25% and 40%, and flash red for battery percentage below 25%. The LED arrays are installed in front of the camera and their status is visible in the streamed video, so the remote operator can initiate a return flight before the battery is out. To enhance quadrotor safety, carbon fiber propeller guards are meticulously designed and installed to minimize the risk of propeller-related accidents. These guards protect both the quadrotor and the pilot, ensuring safe and reliable operation. To reduce deployment time, we design an uninterrupted power supply, ensuring uninterrupted power supply to onboard computers and electronics during battery replacement. Additionally, the quadrotor is equipped with the CUAV Nora+ flight controller⁴², complemented by the CUAV NEO 3 with the NEO-M9N module for GPS data collection (note that GPS data is not used for localization).

After these devices are integrated into the airframe composed of carbon plates and aluminum columns, we conduct tests on the developed quadrotor. The quadrotor features a motor-to-motor distance (i.e., wheelbase) of 320 mm and overall dimensions of 422 mm in length, 422 mm in width, and 285 mm in height. With a total mass of 2.1 kg, the quadrotor achieves a thrust-to-weight ratio of 3, ensuring efficient and stable flight performance. Additionally, the maximum flight time is measured to be 12 min. In the actual field tests, our quadrotor has an average flight speed of 1.5 m/s, corresponding to an approximate flight distance of 1080 m. In practice, this flight distance may vary depending on specific weather conditions and flight characteristics.

Localization

The localization module plays a crucial role by providing real-time state estimation (position, velocity, and attitude) to the quadrotor's controller. Additionally, it transforms the LiDAR points in the local frame into the world frame to construct subsequent navigation maps. In this work, our adapted FAST-LIO2 utilizes the 30 Hz raw point cloud from LiDAR and the 100 Hz IMU. By performing IMU pre-integration, FAST-LIO2 can provide low-latency state estimation at a frequency of 100 Hz with a latency of less than 1 ms. Additionally, it generates world frame registered point clouds at a scan rate of 30 Hz, producing about 200,000 points per second with a latency of less than 10 ms.

Mapping

The mapping module is responsible for constructing high-resolution navigation maps based on the odometry and world frame registered point clouds provided by the localization module. These maps accurately represent the terrain and obstacles, ensuring safe quadrotor navigation through dense vegetation. In our previous work, ROG-Map³⁷, a zero-copy map sliding strategy is utilized to maintain two local occupancy grid maps (OGMs). The first local map is the probability map, where grid cells are classified as Occupied, Unknown, or Known Free based on the occupancy probability updates derived from ray casting. The second local map is the inflated map, utilized for robot navigation in configuration space by inflating obstacles. This map adopts an incremental update mechanism, with grid cell states being classified as either Inflation or No Inflation, depending on the probability values from the probability map. In this work, we introduce three enhancements to ROG-Map specifically tailored for dense vegetation environments: unknown grid cells inflation, infinite points ray casting, and incremental frontiers update.

Unknown grid cells inflation. In motion planning, UAVs are often treated as point masses, and the occupied inflation radius r_{occ} is employed to expand the obstacles in the inflated map. This inflation enables the UAV to avoid known obstacles during planning. However, existing methods^{43–49} often make the simplistic assumption that unknown areas are traversable. In reality, unknown areas may contain obstacles, presenting potential safety risks. To ensure a higher level of safety, it is necessary to not fly in unknown regions and further expand Unknown grids in the probability map by a radius r_{unk} to consider the UAV size and safety clearance. Due to the unknown grid cells inflation, a grid cell state in the inflated map is redefined as Occupied Inflation, Unknown Inflation, and No Inflation.

The method of unknown grid cells inflation is similar to the incremental update mechanism used in the original ROG-Map, which is achieved by maintaining a counter for each grid cell. Therefore, each grid cell in the inflated map maintains two non-negative counters, namely N_{occ} and N_{unk} , which define the grid cell states. A grid cell is considered to be in the Occupied Inflation if $N_{occ} > 0$. If $N_{occ} = 0$ but $N_{unk} > 0$, the cell is classified as Unknown Inflation. Only when both $N_{occ} = 0$ and $N_{unk} = 0$, the cell regarded as No Inflation. Considering the inflation efficiency, it is necessary to determine the set of inflated grid cells for each Occupied or Unknown Grid Cell in advance. Such sets, denoted as \mathbf{I}_{occ} and \mathbf{I}_{unk} , are expressed as grid cell coordinates offsets relative to the input Occupied or Unknown Grid Cell, and are computed offline from the respective inflation radius r_{unk} and r_{occ} . Note that the unknown inflation radius r_{unk} may differ from the occupied inflation radius r_{occ} to give more flexibility. These offset coordinates are then added to the coordinates of the input grid cell to determine this grid cell's inflated occupied and unknown neighbors. During the initialization of the inflated map, all grid cells of the probability map are Unknown, therefore, N_{occ} is set to 0, and N_{unk} is set to the size of \mathbf{I}_{unk} (including the grid cells at the map boundary, since spaces outside the map boundary are also Unknown). Figure 7a illustrates a simplified 2D process of updating from a probability map to an inflated map. Once

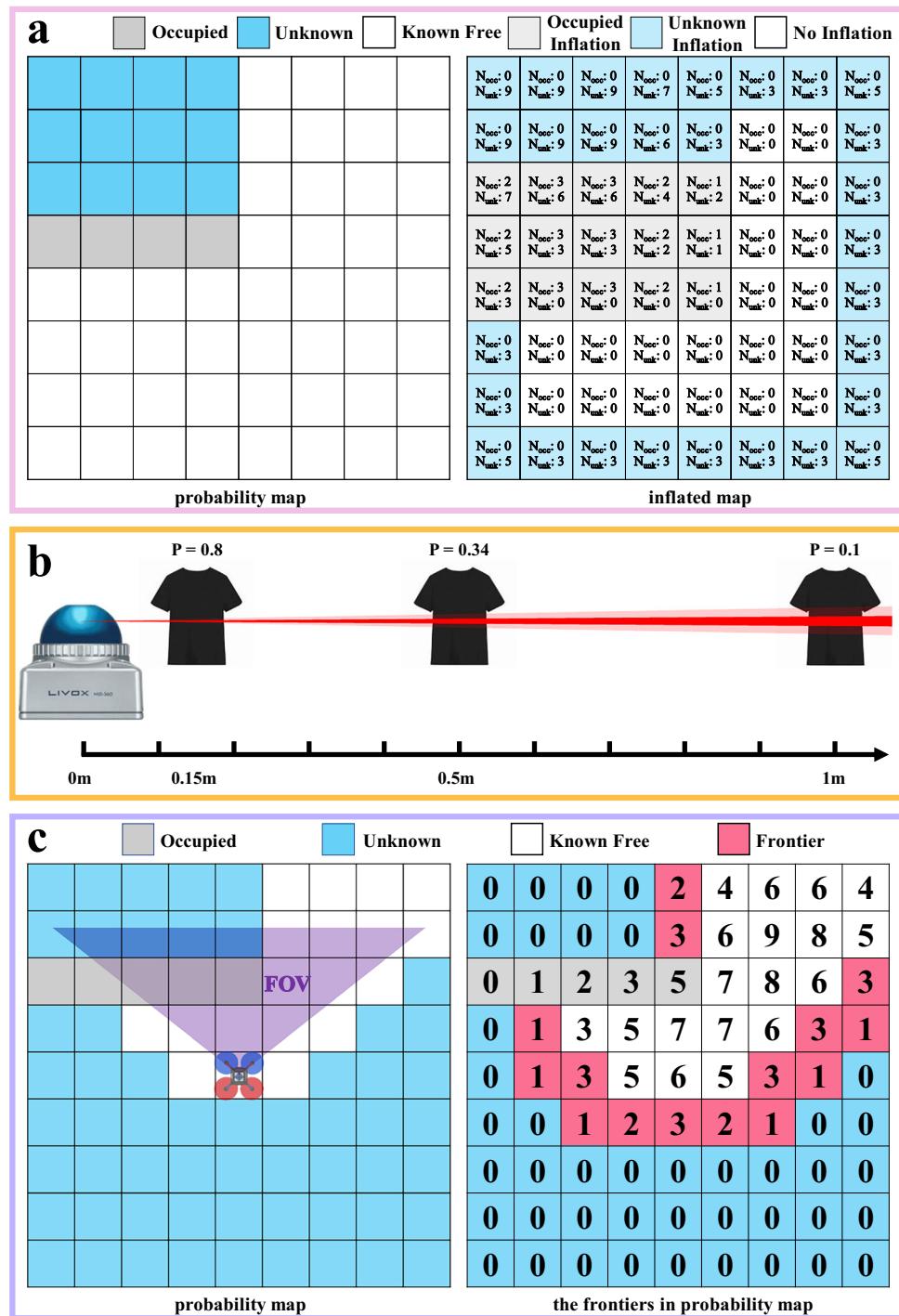


Fig. 7 | Three enhancements of the mapping module. **a** The unknown grid cells inflation in the 2D case. In this case, both the resolution of the probability map and the inflated map are set to 0.1 m, and the inflation radius, r_{occ} and r_{unk} , are both set to 0.2 m. As a result, the quantities of I_{occ} and I_{unk} are equal, with both being 9. **b** Characteristics of Livox Mid-360 LiDAR scanning on clothing. As the distance

increases, the ratio of invalid measurement points to the total measurement points decreases. When the distance exceeds 1 m, this ratio approaches zero, indicating that nearly all measurement points are considered valid. **c** The incremental frontiers update in the 2D case.

occurring a change from an else state to Occupied in the probability map, the corresponding grid cell in the inflated map, along with its inflated neighboring grid cells defined by \mathbf{I}_{occ} , increments N_{occ} by 1. Conversely, if the state changes from Occupied to else, N_{occ} is decremented by 1. Similarly, when the state changes from an else state to Unknown, the corresponding grid cell in the inflated map, along with its inflated neighboring grid cells defined by \mathbf{I}_{unk} , increments N_{unk} by 1.

Conversely, if the state changes from Unknown to else, N_{unk} is decremented by 1. By maintaining these two counters, we can effectively inflate the Unknown grid cells and update the inflated map in real-time based on changes of the grid cell state in the probability map.

Infinite points ray casting. As an active sensor, LiDAR perceives the environment by emitting laser beams and receiving their return pulses.

These laser beams that provide measurement are known as valid measurement points. However, certain situations can result in invalid measurement points. Firstly, when the LiDAR faces the sky, it is unable to measure distances due to the absence of returns. These LiDAR beams are referred to as infinite points. Secondly, when LiDAR scans nearby objects, the returned pulses are lumped into the pulses reflected by the LiDAR internal parts (e.g., prisms, glasses), causing the pulse return due to nearby objects to be indistinguishable from that due to the LiDAR internal parts. Points causing such a phenomenon are defined as nearby blind points. In the Livox Mid-360 LiDAR³⁹ (and also many other LiDARs), the two cases are not distinguished, both leading to a point at LiDAR origin (i.e., invalid measurements). The inability to distinguish infinite points can cause a large number of grid cells lying in the direction of the sky or far buildings having their grid cell state not updated and remain Unknown. To update these grid cells to a Known Free state, we need to distinguish the infinite points from nearby blind points and fully utilize them for ray casting.

We analyze the characteristics of nearby blind points when scanning objects with the Livox Mid-360 LiDAR through an experiment. In the experiment, we scan a cloth at different distances within LiDAR's fixed FOV and record the ratio of invalid measurement points to the total number of measurements. As shown in Fig. 7b, when the cloth is located within a distance of 1m, the ratio of invalid measurements gradually decreases as the distance increases. Beyond 1m, the proportion stabilizes and approaches zero. These results indicate that the nearby blind points caused by objects within 1m account for approximately 10% to 80% invalid measurements, rendering the infinite points indistinguishable in such cases. Only when there are no close-proximate objects (e.g., within 1m), the invalid measurements can be considered as infinite points.

Based on the characteristics of nearby blind points analyzed above, we can extract infinite points by verifying the presence of close-proximate Occupied grid cells in the probability map. Specifically, when receiving LiDAR data, we extract the valid measurement points and invalid measurement points. For the valid measurement points, we perform ray casting to update the probabilities of the grid cells corresponding to obstacles. Next, for each invalid measurement point, we check if there are occupied grid cells within a distance of 1m along its ray direction. If no occupied grid cells are found within the 1-m ray, we identify this invalid measurement point as an infinite point. Subsequently, within the local map region, we use these identified infinite points for ray casting to update the probability map. This process, known as infinite points ray casting, effectively extracts infinite points from the invalid measurement points, enabling the updating of probabilities for grid cells in the direction of the sky.

Incremental frontiers update. Due to the inability of the LiDAR sensor to scan the internal regions of obstacles and the limited LiDAR FOV, numerous grid cells in the probability map will be marked as Unknown. Leveraging frontiers can effectively reduce the computation time required for SFC generation, as frontiers represent the boundary of the unknown region and are typically much fewer in quantity than Unknown grid cells. In the work⁵⁰, frontiers are defined as Known Free grid cells adjacent to Unknown grid cells. We slightly modify this definition and define frontiers as Unknown grid cells adjacent to Known Free grid cells. Considering that SFC is generated in the probability map, we need to label the frontiers in the probability map. Specifically, we introduce a counter N_f for each grid cell, indicating the number of Known Free grid cells among itself and its 26 neighbor grids. Fig. 7c illustrates a simplified 2D Incremental frontiers update process. During the initialization of the probability map, all grid cells are Unknown (including areas outside the map range), therefore, N_f is set to zero. When a grid cell state changes from Known Free to an else state, both the grid cell and its 26 neighboring grids decrement N_f by 1. Conversely, when the grid cell state changes from an else state to

Known Free, both the grid cell and its 26 neighboring grid cells increment N_f by 1. If an Unknown grid cell has its N_f less than 27 but greater than zero, this grid cell is classified as a frontier. Incremental frontiers update allows us to replace a large number of Unknown grids with a small number of frontier grids, thereby reducing the computation time required for SFC generation.

Pilot reference path searching

In slope inspection, the pilot reference path searching module plays a pivotal role in assisted obstacle avoidance. This module first calculates a local goal and the quadrotor's yaw reference based on the pilot's joystick commands and odometry, followed by searching a reference path on the inflated map.

The calculation of the local goal and the quadrotor's yaw reference is shown in Fig. 8a. To align with the pilot's flight expectations in the first-person view, the joystick commands are treated as velocity command \mathbf{v}_{joy} in the quadrotor's yaw frame (a frame with only a rotation in the yaw direction to the world frame, with zero pitch and roll). Let $\mathbf{R}(\psi, 0, 0)$ be the rotation of the yaw frame with respect to the world frame, where ψ is the quadrotor's yaw angle, the local goal in the world frame \mathbf{p}_g and the quadrotor's yaw reference ψ_r can be computed as:

$$\begin{aligned}\mathbf{p}_g &= \mathbf{R}(\psi, 0, 0) \cdot \mathbf{v}_{joy} \cdot \Delta t + \mathbf{p}_{odom} \\ \psi_r &= \psi + w_{\psi, r} \cdot \Delta t\end{aligned}\quad (1)$$

where \mathbf{p}_{odom} is the quadrotor's current position, Δt is the joystick commands period (e.g., Δt is 0.1 s when the joystick commands frequency is 10 Hz).

After calculating the local goal and the quadrotor's yaw reference, the pilot reference path \mathbf{P} is searched on the inflated map. This reference path consists of two segments. The first segment is a reference path in the Inflation region (comprising both Occupied Inflation and Unknown Inflation areas) \mathbf{P}_{inf} , which guides the quadrotor away from inflation regions caused by dynamic objects (e.g., small branches or wires swaying by the wind) and reach a position \mathbf{p}_s in No Inflation regions. The second segment is the reference path in the No Inflation region \mathbf{P}_{no_inf} , which continues to guide the quadrotor toward the local goal \mathbf{p}_g .

The detailed generation of the two path segments is as follows: For \mathbf{P}_{inf} initially, \mathbf{p}_s is set to \mathbf{p}_{odom} and \mathbf{P}_{inf} remains empty. If \mathbf{p}_{odom} is in Inflation state, \mathbf{P}_{inf} starts at \mathbf{p}_{odom} and connects to the nearest No Inflation grid \mathbf{p}_s using a breadth-first search (Fig. 8b1). Then, \mathbf{P}_{no_inf} connects \mathbf{p}_s to the updated local goal \mathbf{p}_{gn} . Initially, \mathbf{p}_{gn} is set to \mathbf{p}_g . If \mathbf{p}_g is in No Inflation state but there are Inflation grids along the connecting line to \mathbf{p}_s , the farthest No Inflation grid along this line is selected as \mathbf{p}_{gn} (Fig. 8b2). If \mathbf{p}_g is in Inflation state, a feasible local goal is first found using a breadth-first search (Fig. 8b3), and then the updated local goal is adjusted as in the previous case. Finally, the complete \mathbf{P} is obtained as the union of the two path segments \mathbf{P}_{inf} and \mathbf{P}_{no_inf} .

It is worth noting that in the inflated map, we do not track or predict the dynamic objects' movements. Instead, dynamic objects are treated together with static obstacles and subjected to inflation operations. The lack of dynamic objects tracking and prediction are compensated by the high planning and control rate of our overall framework, which can avoid dynamic objects in a purely reactive manner. While a larger inflation radius can provide the quadrotor with more reaction time and distance to avoid dynamic objects, it significantly reduces the available space in the inflated map, consequently decreasing maneuverability in narrow areas. Therefore, different obstacle avoidance requirements can be met by adjusting the occupied inflation radius r_{occ} (i.e., the user-defined obstacle avoidance distance) in the inflated map. For instance, setting r_{occ} to approach the quadrotor's radius allows for flight in narrow areas. Alternatively,

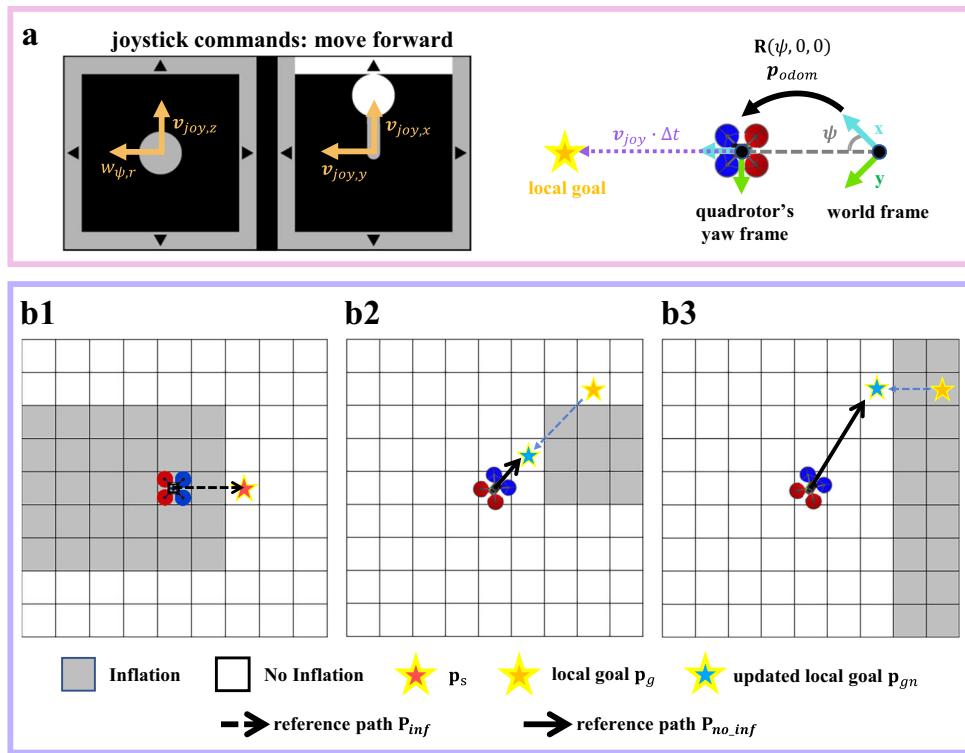


Fig. 8 | Demonstrations of the pilot reference path searching module for assisted obstacle avoidance. **a** Mapping the joystick commands to reference velocity in the quadrotor's yaw frame v_{joy} and reference yaw angular velocity $w_{joy,r}$. Then the local goal p_g and the quadrotor's yaw reference ψ_r is identified. **b** Schematic of reference path searching. If the quadrotor's current position being

in an Inflation (Occupied Inflation or Unknown Inflation) grid cell, a reference path P_{inf} is first searched to leave the inflation area (b1). If the p_g is in No Inflation state but occluded by Inflation grid cells, the farthest visible No Inflation grid cell will be identified as the new p_g (b2). If the p_g is in Inflation state, a nearby No Inflation grid cell will be identified as the new p_g (b3).

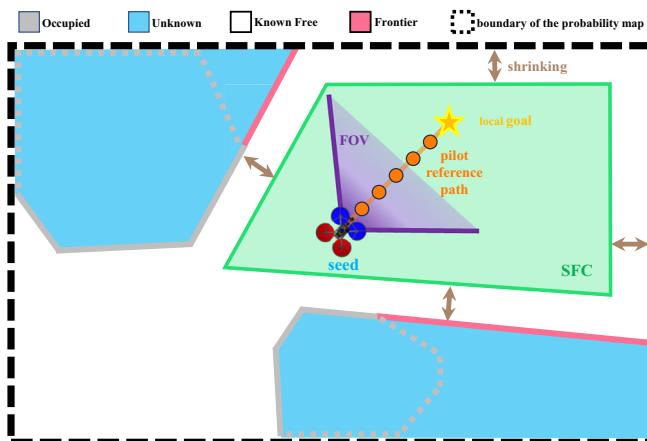


Fig. 9 | The SFC generation considering the unknown areas. The grid cells along the pink lines represent Frontiers, while the grid cells along the gray lines represent Occupied. The orange line represents the pilot reference path, and the green polygon represents the generated SFC.

setting r_{occ} to three times the quadrotor's radius can effectively evade dynamic objects.

Integrated planning and control

The IPC module directly generates throttle and angular velocity references for quadrotor using odometry, probability map, and the pilot reference path. In our previous work³⁸, IPC computes control actions for the quadrotor at a frequency of 100 Hz, allowing for rapid

response to dynamic obstacles. Moreover, by integrating planning and control into a model predictive control (MPC) problem, IPC enhances the quadrotor's ability to suppress external disturbances. These features allow the quadrotor to swiftly avoid small obstacles and resist natural wind disturbances in slope environments.

In this work, we further enhance flight safety in dense vegetation by generating SFC only within Known Free areas. Since no SFC can be generated along P_{inf} in the Inflation area, we only consider generating SFC along P_{no_inf} in the No Inflation area, which resides in the Known Free space of the probability map. Considering that the quadrotor's flight speed in narrow spaces is low, we generate only one convex polyhedron as the SFC to reduce the computation load. We input the grid cells that are identified as Occupied or Frontiers, instead of Occupied or Unknown for improved efficiency, along with the seed, which is the point on the path P_{no_inf} that is closest to the quadrotor's current position, to the polyhedron generation method⁵¹. The obtained polyhedron will intersect with the local map boundary (since the spaces enclosed by Occupied and Frontiers are Known Free when they are within the local map, see Fig. 7c) and shrink by the quadrotor's size. A simplified 2D scenario for SFC generation is illustrated in Fig. 9.

Following³⁸, the reference path and SFC are used as the reference positions and hard constraints, respectively, for the MPC problem. The optimal control actions from the MPC are then transformed into commands for the quadrotor's actuators through differential flatness⁵², which are finally tracked by a lower-level controller implemented on the autopilot to produce motor commands.

Data availability

All data is available in the main text. The source data presented in the manuscript have been deposited in Figshare and can be accessed at

<https://doi.org/10.6084/m9.figshare.29527079>. Additional requests for data should be addressed to F.Zhang.

Code availability

Source code of this work has been provided on the GitHub repository. This code is freely accessible. All the information needed to install and use it, as well as any updates, can be found here: https://github.com/hku-mars/slope_inspection.

References

1. Hong kong slope safety system. <https://hkss.cedd.gov.hk/hkss/en/slope-safety-in-hong-kong/slope-safety-system/overview/index.html>.
2. Castelar Wembers, C. et al. Lidar-based automated UAV inspection of wind turbine rotor blades. *J. Field Robot.* **41**, 1116–1132 (2024).
3. Chen, S., Laefer, D. F., Mangina, E., Zolanvari, S. I. & Byrne, J. UAV bridge inspection through evaluated 3d reconstructions. *J. Bridge Eng.* **24**, 05019001 (2019).
4. Jordan, S. et al. State-of-the-art technologies for UAV inspections. *IET Radar, Sonar Navigation* **12**, 151–164 (2018).
5. Li, Z. et al. Design and application of a uav autonomous inspection system for high-voltage power transmission lines. *Remote Sens.* **15**, 865 (2023).
6. Winkvist, S., Rushforth, E. & Young, K. Towards an autonomous indoor aerial inspection vehicle. *Ind. Robot.: Int. J.* **40**, 196–207 (2013).
7. Nikolic, J. et al. A UAV System for Inspection of Industrial Facilities. In *2013 IEEE aerospace conference*. 1–8 (IEEE, 2013).
8. Car, M., Markovic, L., Ivanovic, A., Orsag, M. & Bogdan, S. Autonomous wind-turbine blade inspection using lidar-equipped unmanned aerial vehicle. *IEEE Access* **8**, 131380–131387 (2020).
9. Bolourian, N. & Hammad, A. Lidar-equipped UAV path planning considering potential locations of defects for bridge inspection. *Autom. Constr.* **117**, 103250 (2020).
10. Aucone, E., Geckeler, C., Morra, D., Pallottino, L. & Mintchev, S. Synergistic morphology and feedback control for traversal of unknown compliant obstacles with aerial robots. *Nat. Commun.* **15**, 2646 (2024).
11. Briod, A., Kornatowski, P., Zufferey, J.-C. & Floreano, D. A collision-resilient flying robot. *J. Field Robot.* **31**, 496–509 (2014).
12. Salaan, C. J. O. et al. Close visual bridge inspection using a UAV with a passive rotating spherical shell. *J. Field Robot.* **35**, 850–867 (2018).
13. Elios3. <https://www.flyability.com/elios-3>.
14. Cleo dronut x1 pro. <https://cleorobotics.com>.
15. Falanga, D., Kleber, K., Mintchev, S., Floreano, D. & Scaramuzza, D. The foldable drone: a morphing quadrotor that can squeeze and fly. *IEEE Robot. Autom. Lett.* **4**, 209–216 (2018).
16. Hu, D., Pei, Z., Shi, J. & Tang, Z. Design, modeling and control of a novel morphing quadrotor. *IEEE Robot. Autom. Lett.* **6**, 8013–8020 (2021).
17. Xu, M. et al. Biomimetic morphing quadrotor inspired by eagle claw for dynamic grasping. *IEEE Trans. Robot.* **40**, 2513–2528 (2024).
18. Bucki, N., Tang, J. & Mueller, M. W. Design and control of a midair-reconfigurable quadcopter using unactuated hinges. *IEEE Trans. Robot.* **39**, 539–557 (2022).
19. Bucki, N. & Mueller, M. W. Design and Control of a Passively Morphing Quadcopter. In *2019 International Conference on Robotics and Automation (ICRA)*. 9116–9122 (IEEE, 2019).
20. Jimenez-Cano, A., Braga, J., Heredia, G. & Ollero, A. Aerial Manipulator for Structure Inspection by Contact from the Underside. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. 1879–1884 (IEEE, 2015).
21. Ikeda, T. et al. Stable camera position control of unmanned aerial vehicle with three-degree-of-freedom manipulator for visual test of bridge inspection. *J. Field Robot.* **36**, 1212–1221 (2019).
22. Yamada, T. et al. Exploration of UAV Rope Handling and Flight in Narrow Space Strategies, Using a Three-arm Aerial Manipulator System. In *2022 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. 677–682 (IEEE, 2022).
23. Neunert, M. et al. Fast Nonlinear Model Predictive Control for Unified Trajectory Optimization and Tracking. In *2016 IEEE international conference on robotics and automation (ICRA)*. 1398–1404 (IEEE, 2016).
24. Wang, Z., Zhou, X., Xu, C. & Gao, F. Geometrically constrained trajectory optimization for multicopters. *IEEE Trans. Robot.* **38**, 3259–3278 (2022).
25. Han, Z. et al. Fast-racing: an open-source strong baseline for se3 planning in autonomous drone racing. *IEEE Robot. Autom. Lett.* **6**, 8631–8638 (2021).
26. Ren, Y., Liang, S., Zhu, F., Lu, G. & Zhang, F. Online Whole-Body Motion Planning for Quadrotor Using Multi-Resolution Search. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 1594–1600 (IEEE, 2023).
27. Wu, T., Chen, Y., Chen, T., Zhao, G. & Gao, F. Whole-body control through narrow gaps from pixels to action. Preprint at <https://arxiv.org/abs/2409.00895> (2024).
28. Dji mavic 3. <https://www.dji.com/support/product/mavic-3>.
29. Skydio 2 plus. <https://www.skydio.com/skydio-2-plus-enterprise>.
30. 2023 hong kong rainstorm and floods. https://wikipedia.org/wiki/2023_Hong_Kong_rainstorm_and_floods.
31. Hk rocked by landslides, road subsidence. <https://news.rthk.hk/rthk/en/component/k2/1717372-20230908.htm>.
32. Maddikunta, P. K. R. et al. Unmanned aerial vehicles in smart agriculture: applications, requirements, and challenges. *IEEE Sens. J.* **21**, 17608–17619 (2021).
33. Seo, J., Duque, L. & Wacker, J. Drone-enabled bridge inspection methodology and application. *Autom. Constr.* **94**, 112–126 (2018).
34. Mishra, B., Garg, D., Narang, P. & Mishra, V. Drone-surveillance for search and rescue in natural disaster. *Comput. Commun.* **156**, 1–10 (2020).
35. Lyu, M., Zhao, Y., Huang, C. & Huang, H. Unmanned aerial vehicles for search and rescue: a survey. *Remote Sens.* **15**, 3266 (2023).
36. Xu, W., Cai, Y., He, D., Lin, J. & Zhang, F. Fast-llo2: fast direct lidar-inertial odometry. *IEEE Trans. Robot.* **38**, 2053–2073 (2022).
37. Ren, Y., Cai, Y., Zhu, F., Liang, S. & Zhang, F. Rog-map: an efficient robocentric occupancy grid map for large-scene and high-resolution lidar-based motion planning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 8119–8125 (2024).
38. Liu, W., Ren, Y. & Zhang, F. Integrated planning and control for quadrotor navigation in presence of suddenly appearing objects and disturbances. *IEEE Robot. Autom. Lett.* **9**, 899–906 (2023).
39. Livox mid-360. <https://www.livoxtech.com/mid-360>.
40. Dji o3 air unit. <https://www.dji.com/o3-air-unit>.
41. Dji goggles 2. <https://www.dji.com/goggles-2>.
42. Cuav nora+. <https://www.cuav.net/en/nora-plus/>.
43. Hart, P. E., Nilsson, N. J. & Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **4**, 100–107 (1968).
44. LaValle, S. Rapidly-exploring random trees: a new tool for path planning. *Res. Report* 9811 (1998).
45. Karman, S. & Frazzoli, E. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **30**, 846–894 (2011).
46. Ren, Y. et al. Bubble Planner: Planning High-Speed Smooth Quadrotor Trajectories Using Receding Corridors. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 6332–6339 (IEEE, 2022).

47. Kong, F., Xu, W., Cai, Y. & Zhang, F. Avoiding dynamic small obstacles with onboard sensing and computation on aerial robots. *IEEE Robot. Autom. Lett.* **6**, 7869–7876 (2021).
48. Zhou, X., Wang, Z., Ye, H., Xu, C. & Gao, F. Ego-planner: an esdf-free gradient-based local planner for quadrotors. *IEEE Robot. Autom. Lett.* **6**, 478–485 (2020).
49. Zhou, B., Gao, F., Wang, L., Liu, C. & Shen, S. Robust and efficient quadrotor trajectory generation for fast autonomous flight. *IEEE Robot. Autom. Lett.* **4**, 3529–3536 (2019).
50. Yamauchi, B. A Frontier-Based Approach for Autonomous Exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'*. 146–151 (IEEE, 1997).
51. Liu, S. et al. Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments. *IEEE Robot. Autom. Lett.* **2**, 1688–1695 (2017).
52. Mellinger, D. & Kumar, V. Minimum Snap Trajectory Generation and Control for Quadrotors. In *2011 IEEE international conference on robotics and automation*. 2520–2525 (IEEE, 2011).

Acknowledgements

This paper is published with the permission of the Head of the Geotechnical Engineering Office and the Director of Civil Engineering and Development, the Government of the Hong Kong Special Administrative Region. This work was supported by the Hong Kong Civil Engineering and Development Department (CEDD). The authors would like to thank Mr. Sammy Cheung from the Hong Kong CEDD for his support, insightful discussions and valuable suggestions on improving the writing of this article. His expertise greatly contributed to the quality and clarity of our work. The authors would also like to express their gratitude to Mr. Ruize Xue, Mr. Bowen Wang, Mr. Xuesong Han, Ms. Liuyu Shi and Ms. Minghe Chen for their valuable assistance during the experiments. This project has received funding from Hong Kong Research Grants Council (RGC) General Research Fund (GRF) 17204523 and a DJI donation.

Author contributions

F.Zhang proposed the initial idea of the research. With the advice of F.Zhang, W.L. designed the complete system framework and experiments. W.L., Y.R. and F.Zhu implemented all software modules. With the contribution of R.G., A.H, Y.C., Y.Z., H.W., and F.Zhang, W.L. performed

the UAV experiments. F.Zhang, V.K. and A.H. provided funding and supervised the research.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41467-025-62801-y>.

Correspondence and requests for materials should be addressed to Fu Zhang.

Peer review information *Nature Communications* thanks the anonymous reviewers for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025