

# Autonomous Aerial Inspection using Visual-Inertial Robust Localization and Mapping

Lucas Teixeira, Ignacio Alzugaray and Margarita Chli

**Abstract** With recent technological breakthroughs bringing fully autonomous inspection using small Unmanned Aerial Vehicles (UAVs) closer to reality, the community of Robotics has actively been developing the real-time perception capabilities able to run onboard such constraint platforms. Despite good progress, realistic deployment of autonomous UAVs in GPS-denied environments is still rudimentary. In this work, we propose a novel system to generate a collision-free path towards a user-specified inspection direction for a small UAV using monocular-inertial sensing only and performing all computation onboard. Estimating both the previously unknown scene and the UAV’s trajectory *on the fly*, this system is evaluated on real experiments outdoors in the presence of wind and poorly structured environments. Our analysis reveals the shortcomings of using sparse feature maps for planning, highlighting the importance of robust dense scene estimation proposed here.

## 1 Introduction

The remarkable agility of small Unmanned Aerial Vehicles (UAVs) has been drawing growing interest in tasks, such as aerial surveying and industrial inspection (e.g. of wind turbines<sup>1</sup>). Their employment in open fields can be automated using pre-defined GPS waypoints. However, such methods do not only assume the absence of any obstacles at the pre-specified flight altitude, but also the availability of reliable GPS signals, restricting their applicability. As a result, most missions employing UAVs today, resort to manually driven flights around the structures of interest, albeit limiting dramatically the inherent agility of UAVs to the pilot’s field of view and judgement of clearance to structures.

---

e-mail: lteixeira@mavt.ethz.ch · aignacio@student.ethz.ch · chlim@ethz.ch  
Vision for Robotics Lab (V4RL), ETH Zurich, Switzerland

<sup>1</sup> <http://aeroworks2020.eu>

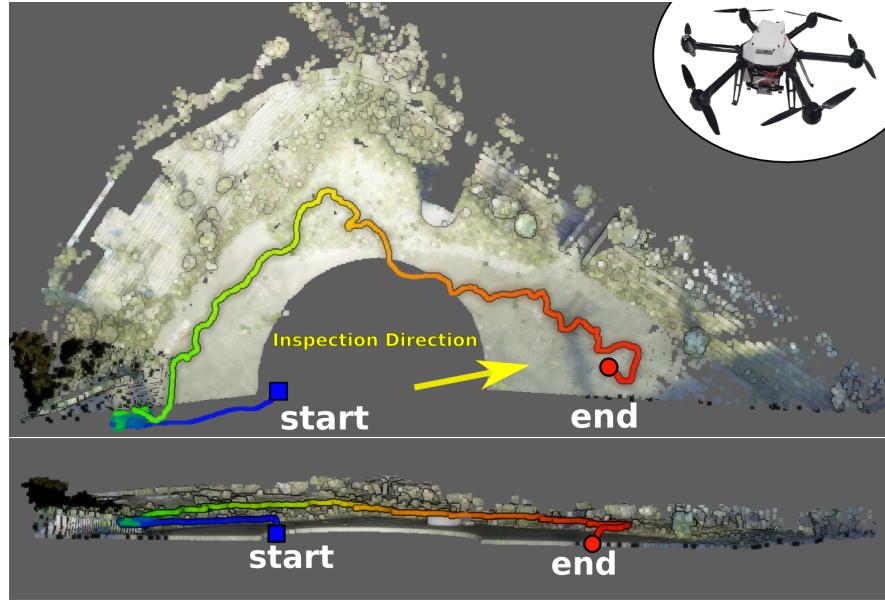
Combining promising state-of-the-art building blocks, in this paper, we present a novel system capable of generating autonomously collision-free trajectories for aerial inspection in potentially GPS-denied environments. Despite the ability of stereo image processing in providing reliable scene depth estimates, monocular sensing in preferred onboard small UAVs as the stereo baseline most often proves too small to be effective in general missions, such as to provide clearance from structures. Factoring in the possibility of external disturbances, such as wind gusts, a clearance range of 10-20m is recommended. As a result, the minimal setup of monocular-inertial sensing is typical for Simultaneous Localization And Mapping (SLAM) onboard small UAVs, respecting the platform's constraining payload and computational capabilities.

Starting off without any prior knowledge of the scene, the proposed system builds on monocular-inertial SLAM to obtain a sparse map of the UAV's surroundings and estimate a denser scene representation as in [18]. Employing the Monocular-Inertial SLAM based Planner (MISP) [2], demonstrated to be suitable for aerial inspection in simulation, our system plans the UAV's path to a pre-specified inspection direction using its current estimate of the scene, as shown in Figure 1. The structure to be inspected is assumed to be a static manifold with enough free space around to safely fly. Exploiting the power of MISP to generate a collision-free path via continuous re-planning, any structures encountered are inspected and used as a source of localization cues. Evaluating MISP in real experiments for the first time, we demonstrate that our denser scene representation provides a far more practical alternative to feature-based maps used in [2].

## 2 Related work

The complex task of automating inspection requires that the robot's spatial perception and path-planning capabilities are synchronized and able to cope with the uncertainties arising in a real mission. While there is a vast body of literature addressing either robotic perception or planning, it is due to the inherent difficulty in dealing with uncertainties in both processes at the same time, that there is only a handful of works addressing their combination and employment in real scenarios. One of the first works to demonstrate successful path-planning for UAVs was the framework of [1], which used Rapidly-exploring Random Belief Trees [6] on a pre-acquired map of visual features to predict subsequent scene measurements. The prohibitive cost of this approach, however, requires a base station for off-board computations, while its employment of known maps limits its applicability to general tasks. Tackling path-planning for an incrementally built map using SLAM, the monocular setup of [15] employed a set of heuristic rules to indirectly evaluate the quality of the generated paths, leading to an improvement in performance of the SLAM system in small scale scenarios.

In the most naive approach for collision avoidance, ultrasound range finders are typically used as a last-minute resort to avoid obstacles (e.g. in parking guidance



**Fig. 1** The top view (top) and the side view (bottom) of a successful run from the start position towards the inspection direction, until the pilot decides to take over (labelled as “end”). Note that the planner chooses to follow the structure, while planning the robot’s path towards the inspection direction as the structure is a rich source of visual cues, and thus ensures the system’s robustness. The color of the trajectory represents the time. The platform used for all the experiments in this paper is the AscTec Neo, shown in the inset.

for cars applications). However, it is depth and/or stereo cameras that are mostly often employed to enable more sophisticated obstacle avoidance strategies, as they provide denser scene information. Nonetheless, the limited range of both ultrasound or RGBD sensors as well as their sensitivity to the environment renders them often impractical in general scenarios.

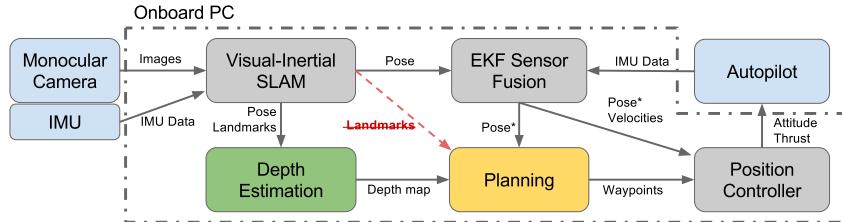
Investigating the applicability of high performing real-time stereo matching algorithms, we tested ELAS [9] and SGBM [10] to estimate a stereo-based scene reconstruction. However, as predicted theoretically, the stereo baseline is relatively too small with respect to the depth of the scene in the mid-range inspection addressed here, producing poor scene estimates, thus rendering such systems unsuitable. As aforementioned, the monocular-inertial setup employed in this work is typical in UAV navigation, however, it poses great challenges in robust and denser scene estimation. The most recent works of monocular ORB-SLAM [16], and monocular-inertial ROVIO [5] and OKVIS [13], demonstrated that robust robot localization using a sparse set of visual landmarks can be performed in real-time even without using cues from a secondary (stereo) camera. ROVIO, for instance, tracks only about 20 visual features per frame, while OKVIS can track about 200. While ROVIO’s map is clearly too sparse for meaningful use during path-planning, OKVIS’s map often provides a more preferable (i.e. denser) distribution features in space. Denser

real-time monocular scene estimation approaches, such as LSD-SLAM [8] offer more complete scene representations, but are too noisy to use for path-planning.

In this work, we employ the low-cost approach for denser scene representation from monocular views of [18]. The outlier removal, smoothing and interpolation of the SLAM landmarks performed by this algorithm overcome the problem of uneven distribution of landmarks and noisy measurements, resulting to a favourable map for path planning. Employing MISP [2], the generated UAV trajectory explicitly considers the motion constraints of a robot with a monocular setup, improving the overall performance of the SLAM system.

### 3 System

Our system for aerial inspection using visual and inertial sensing cues builds on top of the open-source ETHZ-ASL ROS stack for visual-inertial autonomous flights<sup>2</sup>. This framework permits the use of the three sensors that we have onboard our UAV; a monocular camera with an Inertial Measurement Unit (IMU) attached to it and another IMU inside the body of the UAV (at its center). The former IMU and the camera are embedded in the same circuit for time-synchronization as in the Intel RealSense ZR300 Camera. The latter IMU is used by the UAV’s autopilot system inside the Attitude Controller.



**Fig. 2** The architecture of our system. Using both the feeds from the external IMU and monocular camera, and the autopilot’s IMU, the UAV’s onboard computer runs the autonomous flight stack (gray blocks), which estimates the UAV’s pose and a map of landmarks of the UAV’s surroundings. Instead of naively using the landmarks to plan the UAV’s path to the goal (using the red-dashed arrow), our experiments demonstrate that introducing a denser scene estimation module (green block) results to more robust and accurate performance for aerial inspection.

Using the sensor information from all onboard sensors, the proposed system, illustrated in Figure 2, is able to accurately estimate on the fly and in real-time the UAV’s motion while building a map of the environment. Using such information, the system plans and executes a collision-free path to the user-defined inspection direction, while following closely the structure in front of the UAV (i.e. any structure with visual features).

<sup>2</sup> [http://wiki.ros.org/asctec\\_mav\\_framework](http://wiki.ros.org/asctec_mav_framework)

To this end, the autonomous flight stack, depicted in gray in Figure 2, is composed of a visual-inertial SLAM system, the keyframe-based odometry algorithm OKVIS [13], the EKF-based Multi-Sensor Fusion algorithm [14], and the linear model-predictive position controller of [12]. This stack is well tested and used in several works, such as [3, 17]. Employing a variant of MISP [2], here we modified the original path-planning algorithm to enable its practicality and feasibility in real-life experiments. Having been only illustrated in simulation so far, The original MISP uses directly the landmarks provided by the visual-inertial SLAM (OKVIS), as illustrated by the red dashed arrow. In this work, we propose to use a robust scene depth estimation module as we demonstrate that the naive use of the pure visual-inertial SLAM landmarks does not provide an accurate enough scene representation in real missions. All of the components of the proposed pipeline run in the onboard UAV’s CPU.

In the rest of this section, a brief description of the MISP algorithm is presented, together with our strategy for real-time and robust depth estimation and their integration within the proposed pipeline.

### 3.1 Path planning based on MISP

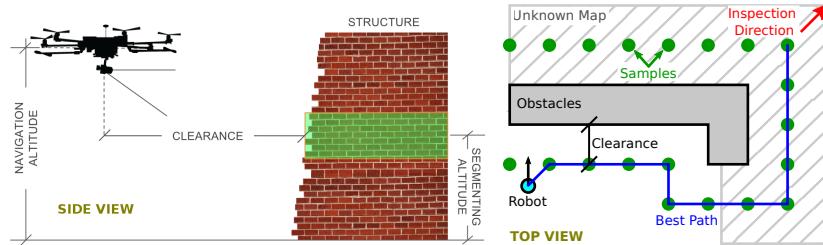
MISP [2] is a path planning algorithm specially designed for small UAVs with limited computational capacity using monocular-inertial localization and mapping. The algorithm exploits the structure of an *a priori* unknown map to guide the robot towards a known goal position, adapting the navigation as new areas are explored by means of an quick re-planning policy. The efficiency of the algorithm relies on the reuse of most of the underlying information between consecutive planning iterations by limiting its reaction to only local changes in the map.

MISP facilitates the monocular localization by planning around the obstacles in the map, while tracking and orienting the robot towards the visual features on them. This path planning leads to a tangential navigation with respect to the local obstacles that is most suitable for the inspection task proposed in this work. The algorithm is originally designed as point-to-point planner, whereas the mission proposed in this work does not consist of reaching a specific location, but rather inspecting the structure that the UAV is facing at the time that MISP gets initialized. For this reason, here we adapt the original algorithm to be guided by a generic and user-defined inspection direction instead of a goal location as depicted in Figure 3. In [2], MISP achieves good results in an inspection task in a simulated environment. Here, we address in detail the limitations of original algorithm when it is applied to real scenarios in the following sections.

MISP employs a probabilistic 3D grid map representation [11] to register the obstacles in the map on-the-fly. The obstacles in the map are considered sources of visual features and thus, the accurate tracking of such features has critical impact in the performance of the employed vision-based SLAM system. Inspired by [7],

MISP generates position samples around to the map obstacles at a given clearance distance  $\rho$  enforcing a collision-free navigation.

In this work, we restrict the algorithm to plan in a plane at a fixed navigation altitude  $h_{nav}$ , as proposed originally in [2]. For the proposed inspection task, we also define the segmenting altitude  $h_{seg}$  at which the structure of the environment is considered of interest and thus any obstacles at this altitude are used to generate the position samples as previously described (See Figure 3).



**Fig. 3** In an inspection task, the MISP segments part of the inspected structure at a given segmenting altitude  $h_{seg}$ . The segmented part of the structure, in green, is then used to generate paths keeping a constant clearance at navigation altitude  $h_{nav}$ . On the right, the MISP computes the best path towards the inspection direction. The structure of the environment in the unknown map areas is inferred based on the current map, relying on the efficient re-planning policy to adapt the navigation to upcoming changes.

The position samples surrounding the obstacles of interest are connected to other neighbouring position samples via collision-free paths generating a graph in which the current robot and goal position are also included. The graph is then searched for the best path connecting the current position to the goal. The weights of the graph are designed so that the resulting best path favours the robot to navigate close to the obstacles at the given predefined clearance distance  $\rho$ , even if it implies a larger overall travelled distance towards the goal.

We redefine the “goal” in the original MISP implementation to use an inspection direction instead, guiding the navigation of the robot while avoiding any obstacles in the way. We implement this modification by setting the goal location beyond the reach of the inspection area. The mission is considered completed as soon as the robot leaves the inspection area guided by the inspection direction while flying autonomously.

In each planning iteration, the segments of the best path closer to the current robot position are determined by the known local map, whereas any segments further away are generated by inferring the structure of the unknown map areas subject to changes. As a result, we only execute the path up to a given short travelled distance, usually a couple of meters, and subsequently repeat a planning iteration including any new updates of the map. This receding horizon strategy, inspired by [4], allows the robot to re-plan its path and adapt its navigation as new areas in the map are explored.

The heading of the robot is defined to be almost perpendicular to the inspected structure’s surface to maximize the parallax of the perceived visual features, ensuring this way to boost the robustness of the SLAM system during the mission. The heading direction deviates slightly (e.g.  $\sim 15^\circ$ ) from the nominal perpendicular heading towards the navigation direction in order to safely detect upcoming changes in the structure. The original MISP implementation defines an additional set of so-called “recovery behaviours” to overcome other endangering situations when navigating close to the obstacles. Since this work focuses on medium-large range inspection, we do not consider such heuristics. Although the structure is assumed to be a manifold, small gaps can be effectively handled by MISP, since it allows the robot to move from one obstacle to another inside of the inspection area, as well as the dense depth estimation, that is able to fill small areas without visual information in the map.

### **3.2 Robust and Dense Depth Estimation**

The original MISP implementation [2] naively assumes that the raw landmarks estimated during SLAM are outlier-free, well distributed across the scene, and dense enough to allow a good enough estimate of the world using a probabilistic 3D voxel grid map at a coarse resolution. While these assumptions may hold in simulated environments, in real scenarios noisy scene estimates and textureless areas are common enough to cause large deviations of the resulting behaviour from the nominal expected performance, as illustrated in the experimental results presented in Section 4.

Considering the limited computational capacity and payload restrictions onboard a small UAV, in this work we focus on the already available monocular and inertial sensing cues to generate a denser scene representation, making the most of the onboard sensor suite. Inspired by the mesh-based scene representation of [18], we employ this method, which was specifically designed for aerial inspection to create a mesh out of the landmarks estimated by SLAM and then perform an outlier elimination and smoothing of this mesh. This system assumes that the area in the field of view is a continuous surface without small or thin objects protruding, although the surface can be non-planar and have small discontinuities, which most often holds in reality (see experimental setup of Section 4). Note that, although this mesh-based scene representation is also capable of providing surface normals, this feature is not used in this work.

As depicted in Figure 8, the difference of the map estimation when using only raw landmarks as opposed to the mesh-based depth estimation pipeline [18] is clearly noticeable. This mesh is computed using only the raw landmarks and the camera pose estimates from SLAM. Firstly, the 3D landmarks are checked for neighbourhood support (in their depth estimates) in order to filter out unreliable landmarks, i.e. the ones with much larger uncertainty within their surroundings. The remaining landmarks are projected onto the image plane corresponding to the current view-



**Fig. 4** Aerial (top) and ground photos (bottom) of the test site. Aerial image source: Swiss Panorama

point and a mesh is constructed via a 2D Delaunay Triangulation. An outlier detection algorithm then removes any triangles that do not fulfil the algorithm's smoothness criteria; very oblique mesh-triangles and spikes in the mesh are rejected. The remaining mesh-vertices are smoothed with respect to their depth using a Laplacian filter, before interpolation is applied by a rasterization algorithm. The outcome of this pipeline is a smooth mesh representation of the scene, providing one depth measurement per pixel in the image space and resulting into a depth map with the same resolution as the camera image. For all our tests, we use the default parameters provided in the open source implementation of the authors<sup>3</sup>. Please refer to the original paper [18] for detailed explanation.

## 4 Experiments

In order to evaluate the proposed system, we carried experiments out in the ruins of the Hardturm Stadium in Zurich visible in Figure 4. We choose to inspect the largest structure of the stadium's stands (*aka* bleachers). This structure is about 8m tall and forms an almost continuous surface covering a  $100\text{m} \times 100\text{m}$  area. Ground-truth data was recorded for the test site by capturing a laser point cloud of the scene using the Leica MS50 Station Theodolite. The ground-truth for the UAV's trajectory in each run, was generated by post-processing each sequence with OKVIS [13] using unbounded optimization times and extended optimization windows.

During the experiments the algorithms face additional challenges, such as light wind gusts. The wind does not only interfere with the UAV's trajectory, but also with the environment due to the fact that trees and bushes cover a large area in front of the structure. Additionally, there is a featureless white tent in the area which adds

---

<sup>3</sup> [https://github.com/VIS4ROB-lab/mesh\\_based\\_mapping](https://github.com/VIS4ROB-lab/mesh_based_mapping)

difficulty in performing SLAM robustly as this can occupy a large portion of the UAV's field of view at times.

Once the algorithm is initialized, the pilot hands over all control of the UAV's motion to the proposed system and takes back the control either at the end of the route or if the pilot judges that the UAV enters a dangerous situation (e.g. flies too close to a structure). The inspection direction is set as shown in Figure 7.

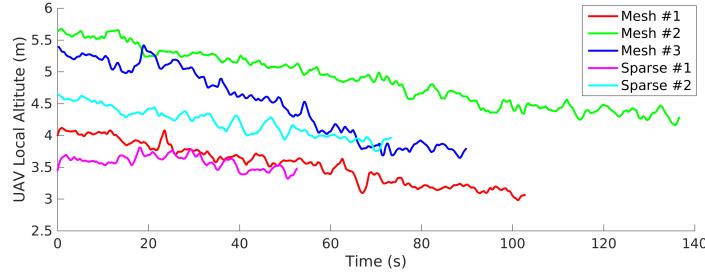
Following a few test-flights, we set the clearance to 10m for this setup and the current wind conditions. Making sure that the UAV is facing the structure to be inspected at the beginning of each mission, the origin of the UAV's coordinate system is set at the position where the SLAM module is initialized. This initial position also determines the segmenting altitude, while the navigation altitude is set at 2m higher due to the sensor's slightly downward-tilted configuration as illustrated in Figure 3. For fairness of comparisons amongst different experiments, the segmenting altitude is set to 2m measured by the UAV's barometer and the navigation altitude at 4m meters high. Note that the barometer readings can be affected by the wind, so these settings can only be approximate. In this setup, the rest of this section analyses the capability of our system in respecting the navigating altitude and the clearance. Further on, the quality of the generated map and the UAV trajectories are discussed, on the basis of five different flights with the same starting pose and inspection direction; three of these using the mesh-based robust dense depth estimation and two using only the SLAM sparse landmark map as a representation of the scene.

#### **4.1 Navigation Altitude**

A constant navigation altitude is important not only for collision avoidance during the navigation, but also for acquiring images with sufficient overlap, crucial for robust navigation and effective inspection. Figure 5 illustrates the altitude measured during each of the five flights. There is a clear drift tendency of descending UAV altitude, from the SLAM origin and increasing with respect to the travelled distance and the experiment's duration. As a result, the mesh-based flights have similar descent rate, but larger absolute drift than the sparse-based flights, because the travelled distance is also longer (see Table 1).

#### **4.2 Clearance**

Although MISP aims to maintain a constant clearance to the structure, in practice it is a difficult task for the position controller to stabilize the UAV in presence of wind, specially when all computations are based on noisy pose and map estimates. As a result, some variance on the actual clearance is expected in real experiments, but the average clearance achieved should be similar to the specified. Most importantly, the UAV should never fly too close to the structure for safety reasons.



**Fig. 5** The ground-truth of the navigation altitude throughout each flight, revealing drift in all trials increasing with travelled distance and flight duration.

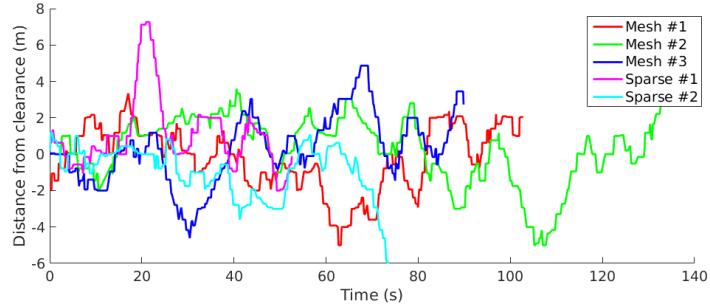
Flight-label	Mesh #1	Mesh #2	Mesh #3	Sparse #1	Sparse #2
Duration (s)	103	137	90	53	74
Distance (m)	120	125	102	62	77

**Table 1** Duration and travelled distance for all five flights used in this analysis. Note that the sparse-based flights are aborted prematurely as they are unable to safely complete the mission.

Extracting the same segmented region from the scene ground truth and in order to evaluate the actual clearance during flights, we build a 2D map (i.e. top view) with the same 1m-resolution as the probabilistic 3D voxel grid representation used in the planner, essentially constructing a binary image. Using this image we calculate the distance field to estimate the actual distance of the UAV (from the trajectory's ground truth) to the closest part of the structure. Figure 6 plots the distance of the actual position of the UAV to the desired clearance (here, set to 10m). The mesh- and the sparse-based approaches exhibit comparable performance, probably due to the fact that the test site was overall sufficiently textured, except the tent region. Self-similar structure, however, e.g. arising from the vegetation around the structure of interest, generates erroneous feature matches. While these can be filtered out to a certain degree during mesh-based flights, for both sparse-based flights this proves fatal as discussed in Section 4.3.

### 4.3 Trajectory Generation and Scene Estimation

Figure 7 depicts the 3D trajectories of all five flights, as MISP plans a path, such that the UAV inspects and follows the structure in the field of view. Each experiment started at the square marker. An experiment is finished either when the UAV reaches the boundaries of the experimental area marked with a disc in Figure 7 or when the UAV's trajectory reveals large errors in the estimation processes and the pilot decides to take over (marked with a star) for safety reasons. More specifically, when the UAV loses the structure of interest from its field of view, it is a point of no return revealing that a path is planned on an invalid map containing too many outliers, and this leads to the decision of ending the experimental flight prematurely. Another



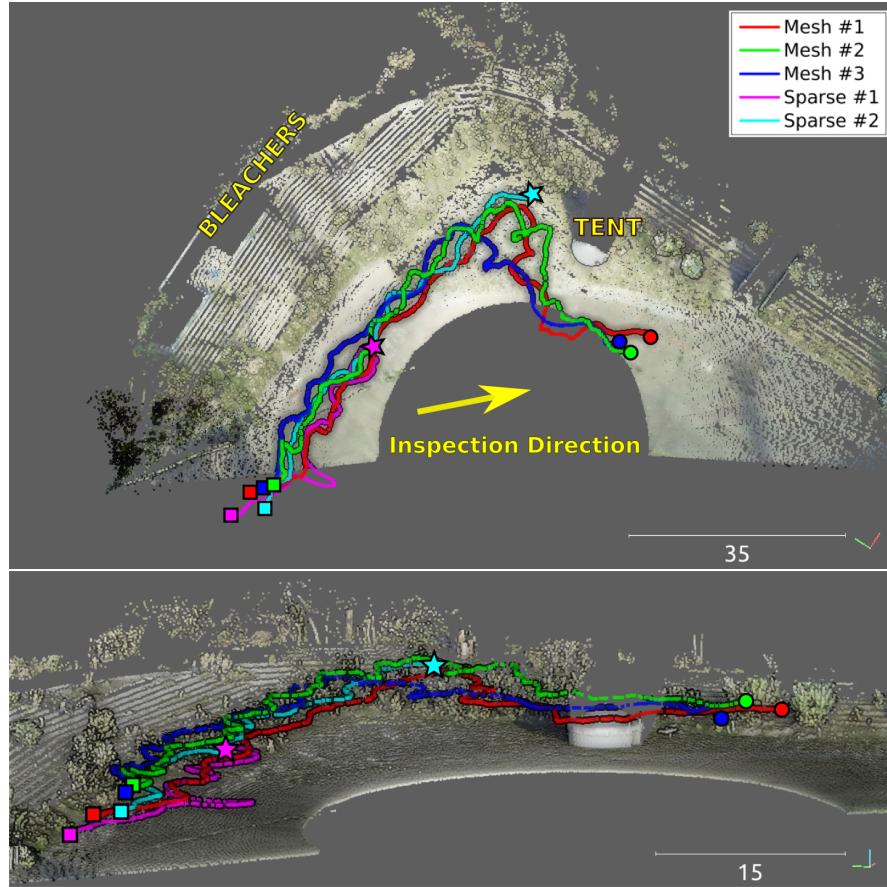
**Fig. 6** Distance between the actual UAV's position and the desired clearance set for the trajectory generation. While sometimes this error is large, in the mesh-based flights on average it approaches zero. During flight Sparse #2, the UAV gets too close from the structure and the safety pilot took over.

case that leads to premature take-over by the pilot is the segmentation of the ground plane (i.e. containing no free space) due to vertical drift. This is very problematic situation that can be more common in our test setup, where we fly close to the ground, however typically aerial inspection is conducted at high altitudes practically eliminating this failure case.

As clearly depicted in Figure 7, both experiments relying on the sparse scene estimation have to be interrupted by the pilot, while all flights employing the mesh-based alternative are able to reach the boundaries of the flying area. In fact, mesh-based dense depth estimation exploits the advantage of accessibility to a denser scene estimation accumulated over time (i.e. multiple views of the same area), outliers can be removed effectively resulting to a far less noisy scene estimation than the sparse counterpart. Generally, most of the outliers in the scene estimation arise during inaccurate registration of the current view to the map, following inaccurate pose estimation from SLAM. Such outliers are very unlikely to be replaced by correct measurements later on in the trajectory, when employing sparse depth estimation traditionally used in SLAM.

Figure 8 shows the probabilistic 3D grid map at the moment of the failure of flight Sparse #1. During navigation, an outlier landmark suddenly gets estimated to be very close to the current UAV position. As a result, the UAV is forced to move away from it to maintain the nominal clearance, rotating during this manoeuvre to face to the direction of this new obstacle that does not really exist, losing in the way the track of the actual structure to be inspected. In the bottom image of Figure 8, the dramatically improved quality of the map when using the mesh-based map is evident.

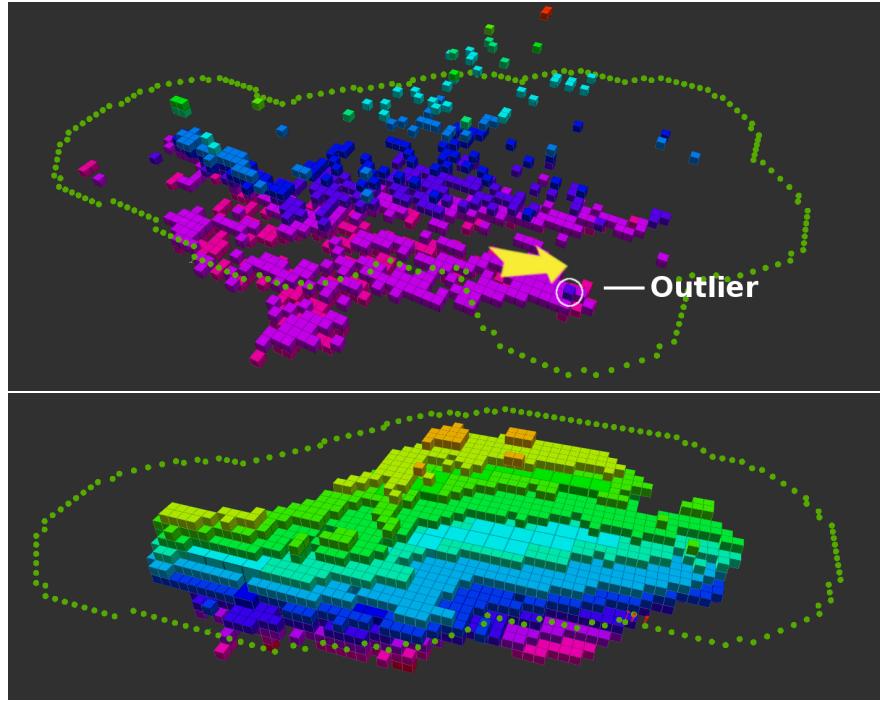
The top-left image of Figure 9 illustrates the sparse SLAM map as estimated at the moment of the failure of flight Sparse #2. In this case, this sparse estimation of the test site does not provide enough information to permit continuing the SLAM estimation during the UAV's rotation necessary for turning at the corner visible also in Figure 7, with the planned trajectory expanding through the white tent – hence if the pilot would not take over, the UAV would collide with the tent.



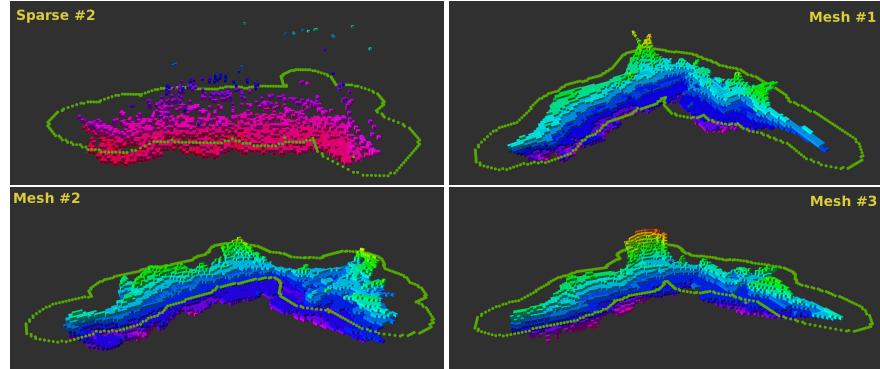
**Fig. 7** All the runs of the proposed pipeline with either options of using the sparse landmark map or the mesh-based scene representation, shown in the top view (top) and a side view (bottom). All experiments start from roughly the same position (marked with a square) and have the same inspection direction. All mesh-based runs autonomously reach the boundaries of the experimental area (marked with a disc), while both sparse-based runs end prematurely (marked with a star).

## 5 Conclusion

This paper proposes a novel system composed of state-of-the-art vision-based perception and planning able to run onboard a small UAV to perform autonomous aerial inspection. Without any *a priori* knowledge of the environment, the UAV's surroundings are estimated in real-time and the UAV's trajectory to inspect the structure of interest is constantly re-planned as newly perceived scene information enters the system. The evaluation of this system on real and challenging experiments outdoors reveals the power of the proposed method to deal with noisy estimates in the perception of the UAV's motion and the scene.



**Fig. 8** The sparse scene estimation (top) at the moment when flight Sparse #1 fails due to the falsely perceived outlier very close to the UAV (yellow arrow is UAV's pose at that moment). As soon as the outlier is encountered a new path is planned (green nodes) around the erroneous obstacle estimate, causing the UAV to lose track of the structure of interest. For comparison, the denser map estimated during the successful flight Mesh #3 is superimposed in the bottom image.



**Fig. 9** The final 3D octomaps retrieved during different flights.

**Acknowledgements** This research was supported by the Swiss National Science Foundation (SNSF, Agreement no. PP00P2\_157585) and EC's Horizon 2020 Programme under grant agreement n. 644128 (AEROWORKS).

## References

1. Achtelik, M.W., Lynen, S., Weiss, S., Chli, M., Siegwart, R.: Motion and Uncertainty Aware Path Planning for Micro Aerial Vehicles. *Journal of Field Robotics (JFR)* **31**(4) (2014)
2. Alzugaray, I., Teixeira, L., Chli, M.: Short-term uav path-planning with monocular-inertial slam in the loop. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2017)
3. Bircher, A., Kamel, M., Alexis, K., Burri, M., Oettershagen, P., Omari, S., Mantel, T., Siegwart, R.: Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots. *Autonomous Robots* **40**(6), 1059–1078 (2016)
4. Bircher, A., Kamel, M., Alexis, K., Oleynikova, H., Siegwart, R.: Receding horizon "next-best-view" planner for 3d exploration. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 1462–1468. IEEE (2016)
5. Bloesch, M., Omari, S., Hutter, M., Siegwart, R.: Robust visual inertial odometry using a direct ekf-based approach. In: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, pp. 298–304. IEEE (2015)
6. Bry, A., Roy, N.: Rapidly-exploring random belief trees for motion planning under uncertainty. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2011)
7. Cover, H., Choudhury, S., Scherer, S., Singh, S.: Sparse tangential network (SPARTAN): Motion planning for micro aerial vehicles. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2013)
8. Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: Proceedings of the European Conference on Computer Vision (ECCV) (2014)
9. Geiger, A., Roser, M., Urtasun, R.: Efficient large-scale stereo matching. In: Proceedings of the Asian Conference on Computer Vision (ACCV) (2010)
10. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* **30**(2), 328–341 (2008)
11. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: OctoMap+: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots* **34**(3) (2013)
12. Kamel, M., Stastny, T., Alexis, K., Siegwart, R.: Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system. In: Robot Operating System (ROS) The Complete Reference Volume 2. Springer (2017)
13. Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., Furgale, P.: Keyframe-based visual-inertial odometry using nonlinear optimization. *International Journal of Robotics Research (IJRR)* (2015)
14. Lynen, S., Achtelik, M.W., Weiss, S., Chli, M., Siegwart, R.: A Robust and Modular Multi-Sensor Fusion Approach Applied to MAV Navigation. In: Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS) (2013)
15. Mostegel, C., Wendel, A., Bischof, H.: Active monocular localization: Towards autonomous monocular exploration for multirotor mavs. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). IEEE (2014)
16. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics (T-RO)* (2015)
17. Oleynikova, H., Burri, M., Taylor, Z., Nieto, J., Siegwart, R., Galceran, E.: Continuous-time trajectory optimization for online uav replanning. In: Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS) (2016)
18. Teixeira, L., Chli, M.: Real-Time Mesh-based Scene Estimation for Aerial Inspection. In: Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS) (2016)