



# Towards Visual Inspection of Distributed and Irregular Structures: A Unified Autonomy Approach

Vignesh Kottayam Viswanathan<sup>1</sup> · Björn Lindqvist<sup>1</sup> · Sumeet Gajanan Satpute<sup>1</sup> · Christoforos Kanellakis<sup>1</sup> · George Nikolakopoulos<sup>1</sup>

Received: 29 March 2023 / Accepted: 29 August 2023 / Published online: 27 September 2023  
© The Author(s) 2023

## Abstract

This paper highlights the significance of maintaining and enhancing situational awareness in Urban Search and Rescue (USAR) missions. It focuses specifically on investigating the capabilities of Unmanned Aerial Vehicles (UAV) equipped with limited sensing capabilities and onboard computational resources to perform visual inspections of apriori unknown fractured and collapsed structures in unfamiliar environments. The proposed approach, referred to as First Look Inspect-Explore (FLIE), employs a flexible bifurcated behavior tree that leverages real-time RGB image and depth cloud data. By employing a recursive and reactive synthesis of safe view pose within the inspection module, FLIE incorporates a novel active visual guidance scheme for identifying previously inspected surfaces. Furthermore, the integration of a tiered hierarchical exploration module with the visual guidance system enables the UAV to navigate towards new and unexplored structures without relying on a map. This decoupling reduces memory overhead and computational effort by eliminating the need to plan based on an incrementally built, error-prone global map. The proposed autonomy is extensively evaluated through simulation and experimental verification under various scenarios and compared against state-of-art approaches, demonstrating its performance and effectiveness.

**Keywords** Unmanned Aerial Vehicles (UAVs) · Inspect-explore · Urban search and rescue robotics · Aerial autonomy

## 1 Introduction

Robot autonomy, at its current stage, has consolidated extensive and multi-disciplinary use cases by providing a profitable and efficient mode of fulfilment of missions [1, 2]. Primarily due to the increased capability and ease of onboard autonomy to be adapted and implemented based on the mission

scenario, autonomous robots have prevailed to serve the needs put forth by industries worldwide. Focusing on the field of aerial robotics, UAVs are widely used in the field of construction for structural health monitoring [3, 4], in the energy industry for inspection of industrial machines such as wind turbines [5], power plants [6] and towards power-line inspection [7]. Moreover, UAV is also finding applications in GPS-denied environments for exploration and mapping of subterranean caves [8–10], for Urban Search and Rescue (USAR) operations [11, 12] and for inspection of mining vehicles [13].

In the context of Urban Search and Rescue USAR operations in urban environments [14–16], the primary aspect of the mission is aimed to improve situational awareness through gathering images and generating a 3D map of collapsed structures found in the deployment zone. This is achieved via an external inspection of the existing structures. The use of autonomous aerial robots in such situations leads to a reduced response time, a robust operational framework and a viable platform for functioning in occluded and beyond line-of-sight scenarios.

---

✉ Vignesh Kottayam Viswanathan  
vigkot@ltu.se

Björn Lindqvist  
bjolin@ltu.se

Sumeet Gajanan Satpute  
sumsat@ltu.se

Christoforos Kanellakis  
chrkan@ltu.se

George Nikolakopoulos  
geonik@ltu.se

<sup>1</sup> Robotics and AI, Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Luleå 97187, Sweden

As the UAV technology advances rapidly, autonomous drones have now become available in smaller sizes and are cost-effective while possessing enhanced sensing capabilities. Nevertheless, the flight time of several drones is constrained due to the augmented payload suite carried on board. This is in contrast to the expected performance of UAVs during standard USAR operations. Consequently, a research area emerges that addresses the challenge of completing a mission with a resource-limited platform thereby improving operational duration. The present study introduces a novel framework that seeks to make a meaningful contribution to the field of autonomous systems by implementing a bifurcated autonomy approach that leverages input data from a singular front-facing vision sensor to complete a mission. Thus, improving flight time by limiting the principal autonomy required for the mission.

## 2 Background and Motivation

The task of gathering information in an unknown environment necessitates the onboard autonomy to fulfil primarily two objectives: *Exploration*, to build 3D map of the surrounding environment and, *Inspection*, to gather dense representation of the 3D structures located within the previously built map.

### 2.1 Related Works

Previous works [17, 18] implement an exploration-guided sampling-based techniques to build a volumetric map of the unknown environment. In their works, the authors model the environment through growing a Rapidly Exploring Random Tree (RRT) to compute collision free paths and optimal viewing strategy formulated as a Next-Best-View (NBV) problem. In [19–21], the authors present a modified version of frontier-driven approach [22] used for exploration and mapping of unknown environments. The authors had proposed a surface-frontier based exploration framework which is aimed to construct complete 3D volumetric models.

On the other hand, surface-based mapping strategy [23] encompass feedback from reconstructed surfaces, in an online fashion, for generation of view-points. In [23], the authors combine the exploring efficiency from a volumetric perspective with consideration of the quality of the observed surface to ensure quality 3D modelling of the structure. Similarly, in [24], the authors present an online sampling-based informative path planner which focuses on growing a solitary RRT-tree to explore and complete coverage of a 3D environment. The surface quality is determined through the formulation of gain function modelled from the weighted score of Truncated Signed Distance Field (TSDF) values of neighbouring constructed surfaces. A similar approach, in

view of Multi-View Stereo MVS 3D modelling, has been addressed by [25–27]. In [26], the authors present a form of *explore-then-exploit* framework wherein a coarse 3D model is built from an initial flight and then a detailed and dense reconstruction trajectory is computed. In [27], the authors propose a reconstruction-heuristic based informative path planning solution.

While, volumetric strategies address building a volumetric map of an unknown environment, the quality of the map obtained is often affected by the resolution of the voxel utilized during planning. Moreover, for large-scale environments, high voxel resolutions results in high computational and memory overhead as a direct result of the ray-cast operations needed to be done to ensure visibility of mapped volumes and to ensure collision-free paths. Progressively built global map are also subject to compounded localization inaccuracies, and thus can result in the map diverging from reality. This can affect view-planning solutions and are often limited to small use-cases.

Moreover, frontier-driven approaches compute information gain from current sensor observations and thus indirectly draw assumptions of partial environment visibility at point of initialization. Thus, if a structure is located outside of current field-of-view, the utility gain approaches to zero and results in premature termination. Surface-based methods require online reconstruction for generation of view path resulting in higher computational and memory overhead for large-scale complex environments. Additionally, existing frameworks, primarily sampling and frontier based approaches, assume a spatially bounded operational region around the target structure under inspection and thus are not suitable for carrying inspection of multiple structures spread across the operational region.

In light of above shortcomings, this work focuses on decoupling the need to plan on global map whilst ensuring a safe and detailed visual inspection of all structures located in the operational region. Building on a parallel approach taken by earlier frameworks to track observed and unobserved surfaces through occupied voxels, this work extends the independence from a volumetric representation by utilizing the stream of RGB images collected during inspection to identify previously inspected surfaces. Thus, by ensuring a reactive and an environment-governed observation, the proposed baseline autonomy, seeks to address both qualitative and quantitative aspects of view-planning, in-effect unifying both inspection and exploration objectives, around unknown structures in an unknown environment.

### 2.2 Contributions

This work builds upon the inspection methodology previously investigated in [28, 29], and extends it with comprehensive experimental and simulation evaluations. In addition

to the contributions of the previous work, the current article proposes a synergistic formulation of inspect-explore autonomy with a vision-based guidance module that leverages RGB images collected during inspection to direct the UAV towards new and uninspected structures. Drawing inspiration from state-of-the-art research and in alignment with the objective of enhancing situational awareness in unfamiliar environments during USAR operations, we present a novel framework with the following contributions:

1. A novel map-independent vision-guided unified inspect-explore framework with reactive synthesis of view poses based on instantaneous sensor information. This is aimed towards autonomous detection and inspection of nearby collapsed structures targeted towards USAR use-case to improve and update in-situ situational awareness.
2. A recursive view planning policy drawing merit from the First-Look formulation, composed of a dual-purpose safety layer intended to keep the UAV safe and to operate in a stable manner. The policy is structured around the reactive scheme to remain robust against large gaps or holes on locally viewed surfaces and enables a dynamic reconfiguration of the view poses based on the instantaneous sensor input. The policy is augmented with an active scene recognition framework that utilizes catalogued RGB images, captured via the onboard stereo optical sensor, to cross-check for previously inspected surfaces. Subsequently ensuring a robust performance against localization drift by decoupling the need to view the plan on a continuously updated map.
3. A novel structured hierarchical survey policy that accounts for behaviour modifications required during exploration. The policy is formulated to take into account the decision to explore, such as exploring during proximity inspection, i.e. to continue visual inspection and prevent premature termination, which is useful for inspection of structures with large surface deformities or gaps/holes, during the initial search for structures, i.e. to detect and localize available structures for inspection and finally, accounting for a travelling search policy through backtracking. The combination of exploration policies ensures a comprehensive survey of the deployed zone for additional structures.
4. Extensive experimental and simulated studies were conducted to assess the efficacy and to present validation proof of the proposed inspect-explore autonomy. The experimental evaluation is performed with the autonomy deployed on a UAV and the criteria focus on the extent of close visual inspection conducted in an unknown indoor environment around 3D EverBlock structures constructed to replicate the distribution and condition of structures in a USAR scenario. The eval-

uation of the autonomy via simulation addresses the inspect-explore mission around realistically modelled fractured and collapsed 3D structures in various arrangements within GAZEBO simulation platform. Comparative analysis of both quantitative and qualitative performance characteristics with a map-based approach is also presented.

The rest of this article can be summarized as follows. In Section 3 the problem statement addressed in this work is described. The proposed methodology is defined in Section 4 along with the utilized low-level autonomy, the environment setup in Section 5. Section 6 provides an analytical discussion of the results obtained. Finally, Section 7 address the limitations faced in this work and Section 8 presents the conclusions drawn and the future scope of this work.

### 3 Problem Definition

Fundamentally, this work considers an unbounded volume  $\mathcal{V} \in \mathbb{R}^3$  within which the occupied volume by the structures,  $S = \{S_1, S_2, \dots, S_m\}$  with  $m \in \mathbb{Z}^+$  is represented as  $\mathcal{V}_{oi}$  corresponding to each  $S_i$  with  $i = \{1, 2, \dots, m\}$ . The aim of the presented article is to address the primary task of executing close visual inspection of multiple a priori unknown structures via a decoupled approach. The position of the UAV is given as  $\mathbf{p} = [x \ y \ z]^T \in \mathbb{R}^3$ . The desired view orientation to maintain is given as yaw component of the orthogonal rotational group,  $\psi \in \mathbb{SO}(1)$ . The view pose,  $\xi = [\mathbf{p} \ \psi] \in \mathbb{R}^3 \times \mathbb{SO}(1)$  is generated subjected to satisfying desired photogrammetric overlap conditions, represented as  $\gamma_H, \gamma_V \in \mathbb{R}^+$  along the horizontal and vertical axis of inspection respectively, and the desired inspection distance  $r_m \in \mathbb{R}^+$ . The outcome of the inspection policy is to find a set of safe view poses,  $\{\xi\}$ . Considering a platform equipped with a single optical stereo sensor, let  $\mathbf{P}_c \in \mathbb{R}^3$  be the raw depth point cloud representation obtained with a sensor range of  $z_c \in \mathbb{R}$ . To ensure a generation of surface-adaptive inspection view pose, a limited Field of View (FOV)-based view cone model is considered with the viewing angle given as  $\sigma \in \mathbb{R}$ . Let  $\mathcal{I}$  be the RGB image frame captured by the sensor at each  $\xi$ .

Thus, the problem definition can be structured as follows. Initializing in a deployed region with a volume  $\mathcal{V}$ , the subsequent tasks are to be achieved: (a) to plan a safe and unique set of inspection configurations,  $\{\xi\} \forall S$ , subject to photogrammetric constraints, (b) to obtain a dense volumetric representation of the reconstructed mesh  $\mathcal{V}_{oi} \forall S_i$  located within  $\mathcal{V}$ , and (c) the planning is based only on the environment knowledge gained from  $\mathbf{P}_c$  and  $\mathcal{I}$ .

## 4 Proposed Methodology

In this work, we propose a system to enable close external visual inspection of a priori unknown fractured or collapsed structures in an unknown environment. Specifically, we target the use of UAVs with limited onboard resources. The presented strategy unifies a recursive inspection view planning policy with a structured hierarchical exploration aimed to ensure an effective survey of the deployed environment to detect and locate nearby structures. Utilizing the images collected, the inspection module is reinforced with a scene recognition scheme allowing the autonomy to be aware of previously inspected surfaces. The unified architecture is based on instantaneous sensor information and thus remains independent of the need of a progressively built global map. As a result of the techniques implemented, the system remains robust against localization inaccuracies along with inheriting the advantage of reduced memory overhead and lower computational cost during the mission. Figure 1 depicts an overview of the proposed unified inspect-explore architecture.

### 4.1 Preliminaries

Let  $\mathcal{B} \in \mathbb{R}^3$  be the body frame attached to the UAV on which the equipped optical sensor is given as  $\mathcal{O} \in \mathbb{R}^3$ . Let  $\mathcal{W} \in \mathbb{R}^3$  be the fixed global reference frame. The pose of the UAV is defined as  $\xi = [p_{uav} \ \psi_{uav}] \in \mathbb{R}^4$  comprising of the translational states  $p_{uav} = [x \ y \ z] \in \mathbb{R}^3$  and the yaw orientation  $\psi_{uav} \in \mathbb{R}$  of the UAV. Figure 2 presents the utilized frames of reference in this work.

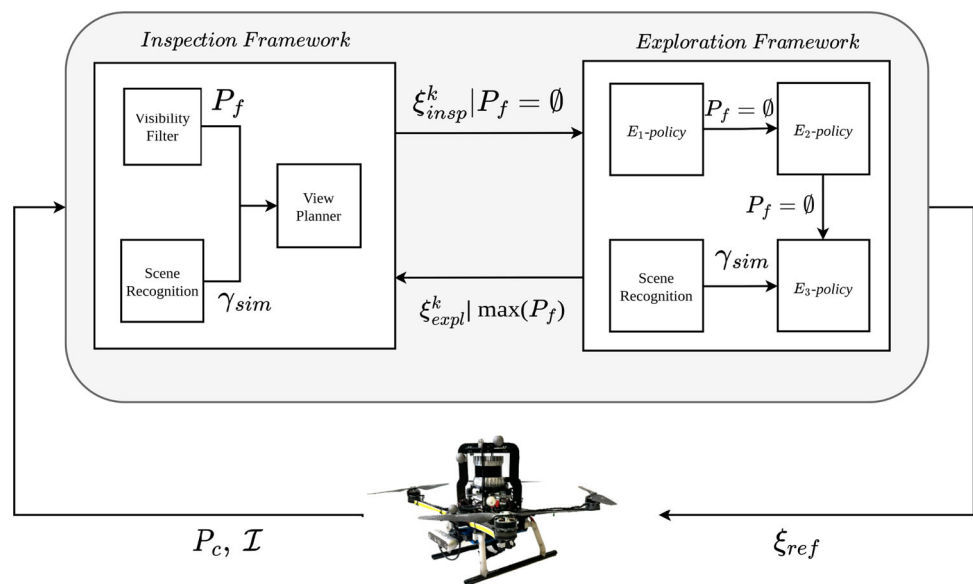
### 4.2 Inspection Framework

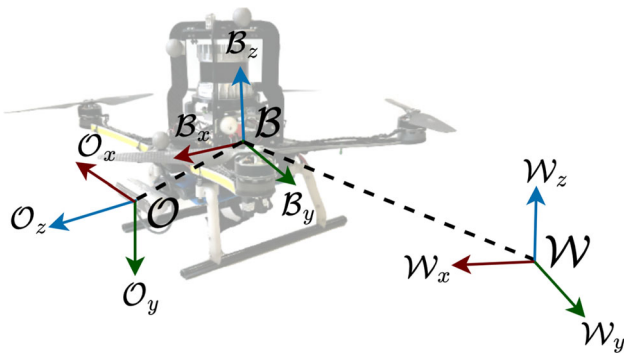
The inspection behaviour of the UAV is modelled subjective to certain constraints, such as (a) satisfying desired horizontal and vertical overlap between consecutive inspection poses, (b) maintaining a desired distance from the collapsed structure during the inspection and avoiding any endangerment due to obstructing protrusions from the structure and (c) maximizing the information gained during the mission through cataloguing and identifying previously inspected surfaces from the images gathered during the inspection.

Let  $P_c \in \mathbb{R}^3$  be the detected depth point cloud by the onboard stereo camera given relative to  $\mathcal{O}$ . To remove noise and downsample  $P_c$  to a manageable size, a voxel grid filter is utilized, represented as **ViewabilityFilter()** function in Fig. 1. Let  $P_f \in \mathbb{R}^3 \subseteq P_c$  be the downsampled and transformed point cloud set obtained from the output of the voxel grid obtained in  $\mathcal{W}$ . A primary aspect of an inspection algorithm is to maintain a view orientation perpendicular to the surface being observed. As such, a visibility condition is enforced, where only points within a viewing angle of  $\sigma$  are considered for subsequent view planning. Figure 3(a) and (b) depicts the sequence of operations performed to obtain the final sampled set of visible points  $P_s \subseteq P_f \in \mathbb{R}^3$ .

In this work, a unit directional vector is generated from the UAV to the current candidate point of the surface being observed. A k-dimensional tree is constructed using  $P_f$  to obtain the nearest 3D point  $p_{poi} \in \mathbb{R}^3$ , representative of the structure under inspection, from  $p_{uav}$  considered at the  $k^{th}$  instant. Let  $v_x, v_y, v_z \in \mathbb{R}^3$  be the normalized directional vectors of the UAV along X, Y and Z axes respectively. Figure 3(c) presents the subsequent  $p_{poi}$  being obtained

**Fig. 1** Synoptic architecture of the proposed inspect-explore autonomy





**Fig. 2** Graphical representation of the frames of references utilized in this work

from  $P_s$ . Equation 1a-1c presents the modelling of directional vectors of the UAV. As per the ROS convention,  $n_z = [0 \ 0 \ 1]^T \in \mathbb{R}^3$  denotes the normalized up-vector of the camera, parallel to the orientation of  $B_z$ .

$$v_x = \frac{p_{poi} - p_{uav}^k}{\|p_{poi} - p_{uav}^k\|} \tag{1a}$$

$$v_y = \text{cross}(n_z, v_x) \tag{1b}$$

$$v_z = \text{cross}(v_y, v_x) \tag{1c}$$

Let  $\alpha \in \mathbb{R}^+$  be the Horizontal FOV of the onboard camera. According to the desired horizontal overlap factor  $\gamma_H$ , the necessary overlap distance  $O_H \in \mathbb{R}^+$  to be maintained can be formulated as trigonometric problem by determining the camera footprint for a given  $r_m$ ,  $\alpha$  and  $\gamma_H$  parameters. Thus, solving for the relative distance,  $O_H$ , to maintain the desired overlap characteristics ensures the view-planner satisfies

desired photogrammetric properties. Equation 2 presents the mathematical description to determine the necessary overlap distance. Figure 4(a) provides a graphical illustration of the modelled horizontal overlap characteristics.

$$O_H = f(p_{uav}^k, p_{poi}, \gamma_H, \alpha) \tag{2}$$

where,

$$f(p_{uav}^k, p_{poi}, \gamma_H, \alpha) = 2 \tan \frac{\alpha}{2} \|p_{poi} - p_{uav}^k\| - 2 \gamma_H \tan \frac{\alpha}{2} \|p_{poi} - p_{uav}^k\|$$

Similarly, given  $\beta \in \mathbb{R}^+$  representing the Vertical FOV of the camera and  $O_V \in \mathbb{R}^+$ , Eq. 2 can be modified and re-written as in Eq. 3. Figure 4(b) provides a graphical illustration of the modelled vertical overlap characteristics.

$$O_V = f(p_{uav}^k, p_{poi}, \gamma_V, \beta) \tag{3}$$

where,

$$f(p_{uav}^k, p_{poi}, \gamma_V, \beta) = 2 \tan \frac{\beta}{2} \|p_{poi} - p_{uav}^k\| - 2 \gamma_V \tan \frac{\beta}{2} \|p_{poi} - p_{uav}^k\|$$

From Eqs.1 and 2, the proposed view planning policy can be formulated as follows,

$$p_{uav}^{k+1,k} = p_{uav}^{k,k} + v_y O_H \tag{4}$$

$$\psi_{uav}^{k+1,k} = \arctan(v_x(2), v_x(1)) \tag{5}$$

**Fig. 3** Graphical representation of the implemented point-cloud down sampling process during visual inspection

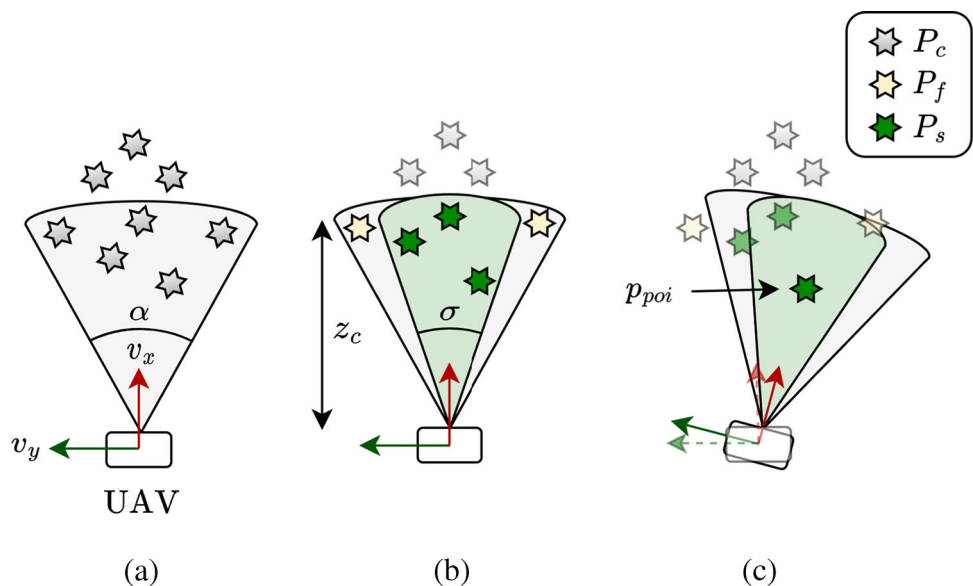


Figure 4 shows a visualization of the modelled photogrammetric constraints to plan view pose during inspection. Together Eqs. 4 and 5 denote the reference view pose  $\xi_{ref}$  fed to the tracking controller. This formulation allows the UAV to be resilient against the presence of gaps or holes along the surface of the structure due to the search for  $p_{poi}$ . Equation 4 is executed recursively with the sensor information being updated as the UAV moves thereby allowing the planner to adapt the inspection path and the required view orientation to the profile of the structure being inspected.

To maintain the desired viewing distance and to avoid collisions with parts of the structure, i.e maintaining a safe distance, a dual-purpose safety layer with a desired distance  $r_m \in \mathbb{R}$  is incorporated into Eq. 4. Equation 4 can thus be appended to its final form in Eq. 6,

$$p_{uav}^{k+1,k} = p_{uav}^{k,k} + v_x r_m + v_y O_H \tag{6}$$

The use of  $r_m$  binds the UAV to maintain the desired distance along the view direction during each iteration. With the UAV’s orientation being guided by the use of  $p_{poi}$ , the presence of projections lying within the field-of-view of the camera is considered within the formulation in Eq. 6. Thus, ensuring that the UAV is able to safely navigate around available structures. The holistic operations of the inspection framework is represented as **ViewPlanner()** function in Fig. 1.

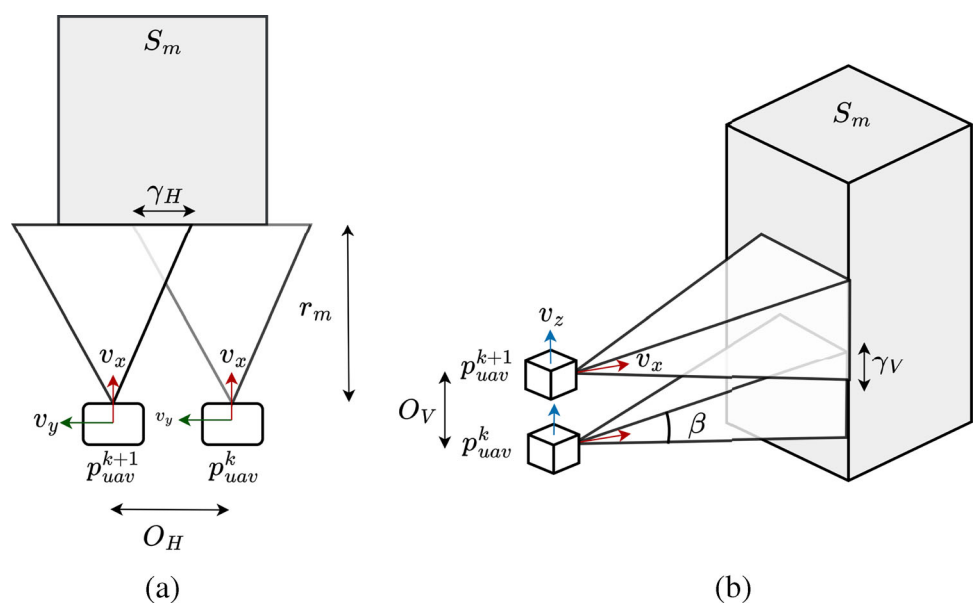
### 4.3 Active Scene Recognition

The view planning policy presented in Eq. 6 executes a loop-by-loop inspection path around the structure. Thus, in order

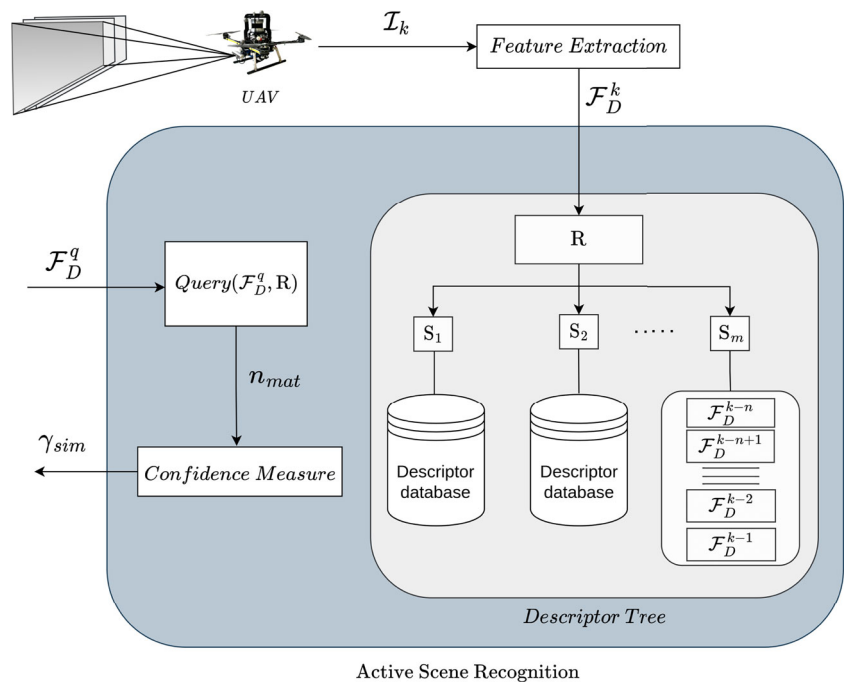
to fulfil the vertical overlap during inspection and to prevent the redundant generation of view poses, the images captured during inspection are used to provide a quantitative measure of the inspected surfaces. Figure 5 represents the approach taken towards the implementation of Active Scene Recognition (ASR) in this work. To achieve recognition of previously inspected structures, the image frames,  $\mathcal{I}_k$ , taken at each inspection view pose is used to extract the corresponding feature descriptors,  $\mathcal{F}_D^k$ . Thus, the augmented descriptor matrix  $\mathcal{F}_{Di}$  for the current structure  $S_i$  under inspection can therefore be formulated as  $\mathcal{F}_{Di} = \{\mathcal{F}_D^1, \mathcal{F}_D^2, \dots, \mathcal{F}_D^k\}$ . Let  $T$  denote a well-ordered set defined as  $T(S!)$ . Thus, in presence of multiple structures, the expanding descriptor tree  $T$  would inherit the order based on the sequence of structures being visited.

For this module, we utilize a low-level Scale-invariant Feature Transform (SIFT) [30] keypoint descriptor to perform feature extraction. During inspection, we model ASR for continuous matching policy, such that for every query image descriptor  $\mathcal{F}_D^k$  captured from each inspection pose, it is compared against the candidate augmented descriptor matrix  $\mathcal{F}_D^c$  extracted from the images captured from the point of engagement of inspection at each loop until  $N$  horizon. The similarity score  $\gamma_{sim}^{insp} \in \mathbb{R}$  is evaluated through the implementation of the Lowe’s match filter [30] on the set of  $n_{mat} \in \mathbb{R}$  matches obtained between the query and candidate descriptors. Feature matching is performed via Fast Library for Approximate Nearest Neighbors (FLANN) [31] based k-nearest neighbour search with the filtered set of good matches represented as  $n_{knn} \in \mathbb{R}$ . The filter is represented as *Confidence Measure* sub-module in Fig. 5. Let  $\gamma_{thresh}^{insp} \in \mathbb{R}$  be the threshold score above which we consider two scenes to be similar. Thus, Eq. 7 formulates the derived similarity

**Fig. 4** Graphical representation of the modelled photogrammetric constraints for view pose synthesis during inspection



**Fig. 5** Graphical diagram depicting the framework within Active Scene Recognition module



condition between any two image frames evaluated during inspection.

$$\gamma_{sim}^{insp} = \frac{n_{knn}}{n_{mat}} \geq \gamma_{thresh}^{insp} \tag{7}$$

Thus, at each  $p_{uav}^k$ , the module stores the corresponding descriptor information of the image captured along with the pose of the UAV at that instant. As mentioned before, the novelty of the proposed methodology is mainly derived from the map-independent approach implemented for inspect-explore planning operations. During inspection of multiple distributed structures, it is critical for the framework to identify previously inspected structures or surfaces to prevent the execution of redundant inspection behaviour. **SceneRecognition()** function, shown in Fig. 1, captures the formulated behaviour in this module.

Primarily, classical methods mainly utilize a bounded volumetric representation of visited surfaces through the use of voxels to advance the search of frontiers towards the unique and unexplored regions. The First Look Inspect-Explore (FLIE) autonomy makes use of candidate RGB images captured and catalogued during inspection to filter out previously seen surfaces. The descriptor tree built during inspection is used to direct the UAV towards new and previously uninspected structures during  $E_3$ -stage of exploration (Section 4.4). When  $P_f \neq \emptyset$  during  $E_3$  exploration, i.e. when a structure is present within the current sensor range,  $F_D^q$  is extracted from the current image frame and is used to query with the descriptor tree to check if the surface has been previously inspected. Let  $\gamma_{thresh}^{expl}$  be the threshold score

below which a structure is considered to be new. Equation 8 presents the condition formulated for the identification of new and uninspected structures during  $E_3$  exploration.

$$\max_{m=1}^M (\gamma_{sim}^{expl}) \forall S_m \in R \leq \gamma_{thresh}^{expl} \tag{8}$$

### 4.4 Exploration Framework

During an external inspection of fractured structures, the autonomy takes into consideration the potential presence of large gaps or holes due to broken walls and collapsed ceilings along the inspection route. The driving factor is to ensure the inspection behaviour remains robust against discontinuities in order to prevent premature termination of visual inspection. Thus, if in the course of executing visual inspection of a structure, in absence of any detected surfaces lying within the current field of view of the stereo camera, that is when  $p_{poi} = \emptyset$  the exploration behaviour is structured to ensure effective re-engagement with the inspecting structure. Moreover, during mission initialization, a comprehensive survey of the immediate vicinity around the UAV is necessary to improve in-situ knowledge in order to tag nearby structures for inspection. A typical USAR operational scenario often has multiple structures situated around the deployed zone. As such, to ensure visual inspection of all the target structures present, the need to explore the unknown region around inspected structures is important.

Thus, to address the aforementioned challenges, the exploration behaviour is structured into three main policies. The initial policy  $E_1$ , functions to survey and detect nearby

extensions of the surface of the inspecting structure based on decomposing the forward view space of the UAV. This is carried out when in the event of inspecting a structure, there are no detected  $p_{poi}$  present within the current field of view of the camera, shown in Fig. 6(a). Therefore, ensuring that the UAV regains visual lock of the surface profile of the structure from the current position. The necessary reference yaw angle,  $\psi_{ref}$ , during  $E_1$  search is bounded between  $[-\frac{\pi}{2}, \frac{\pi}{2}]$  from current view orientation, shown in Fig. 6(b). The desired decomposition,  $m \in \mathbb{Z}^+$ , is dependent on  $\alpha$ .

$$\psi_{ref}^{k+1,k} = \psi_{ref}^k + \alpha, \forall m \tag{9}$$

where,

$$m \approx \frac{\pi}{0.5 \alpha}$$

Let  $G \in \mathbb{R}$  be the information gained, which in this case corresponds to the size of  $P_f$  obtained at each view pose during surveying. The target view poses to be maintained by the UAV is modelled to face towards the direction of  $\max\{G\}$ , shown in Fig. 6(c).

Let  $E_2$  be the secondary survey policy. In a situation where  $G = \emptyset$  at the end of  $E_1$  search or at the initialization of the mission, as portrayed in Fig. 7(a), the exploration behaviour

is escalated to encapsulate  $360^\circ$  search space around the UAV, shown in Fig. 7(b). In addition to that,  $E_2$  policy is flagged when no prior inspection behaviour is registered such as at the event of initialization of the mission. As shown in Eq. 9,  $E_2$  follows a similar formulation, although in this case,  $m \approx \frac{2\pi}{\alpha}$ . Figure 7(c) presents the target view-pose for  $\max\{G\}$  at the end of  $E_2$  policy.

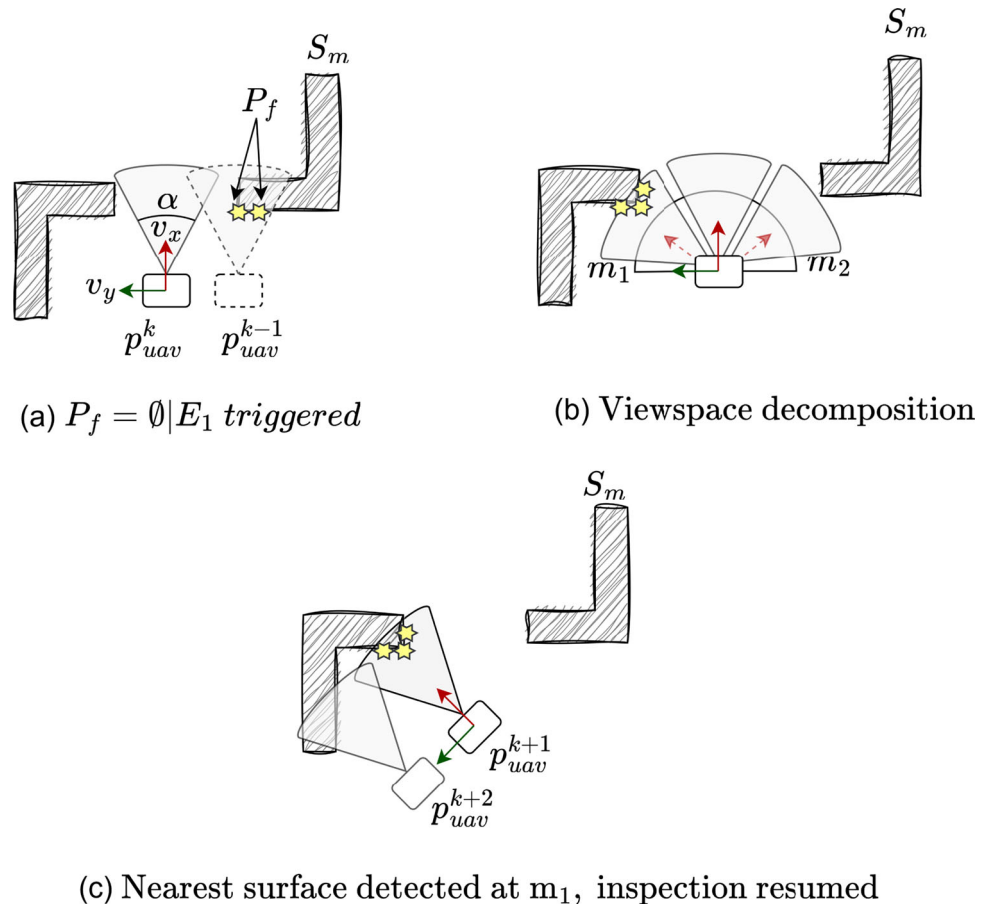
If no new structures are detected at the end of  $E_2$ , i.e.  $G = \emptyset$ , or when the condition of  $\max\{P_f(z)\} < p_{uav}(z)$ , i.e when no more potential extension of the structure is observed above the current position of the UAV, the exploration behaviour is again escalated to its last and final stage. Let  $E_3$  be the tertiary survey policy. During  $E_3$ , the UAV is directed to back-track through a stored repository of visited inspection view poses,  $\xi_{insp} \subseteq \xi$ , with an offset of  $180^\circ$  to each view orientation registered at the candidate pose, shown in Fig. 8(a). In Eqs. 10 and 11,  $j$  corresponds to the size of the stored repository of inspection poses.

$$\xi_{uav}^{k+1,k} = \xi_{insp}^{j-k,k} \tag{10}$$

where the required view orientation is modelled as shown below:

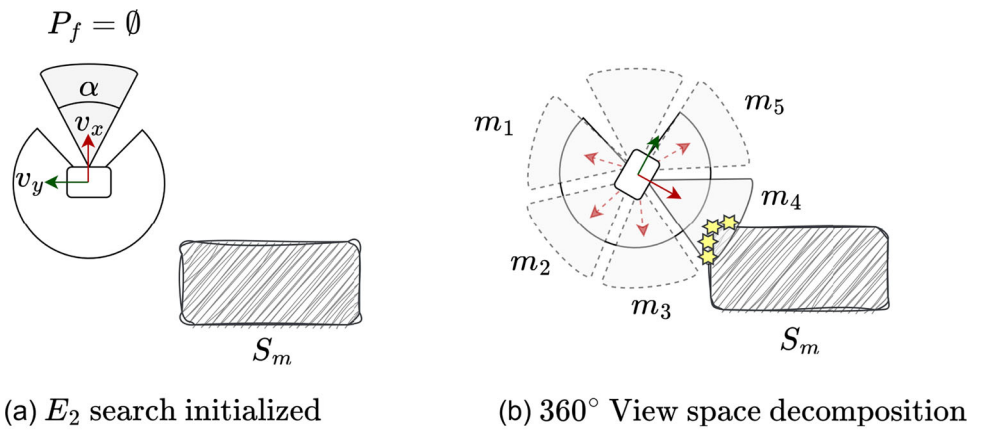
$$\psi_{uav}^{k+1,k} = \pi + \psi_{insp}^{j-k,k} \tag{11}$$

**Fig. 6** Illustrative representation of the sequence of behaviour executed during  $E_1$  search





**Fig. 7** Graphical depiction of the commanded behaviour during execution of  $E_2$  policy



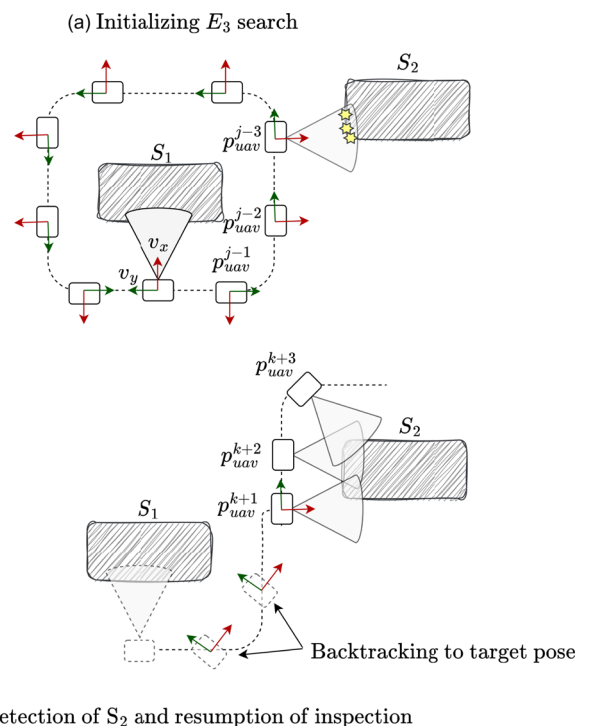
(c) Structure detected at  $m_4$ , inspection policy executed

For Eq. 10,  $\xi_{insp}$  is modelled to access the planned poses at the base level of the inspection loops. Figure 8(b) presents a graphical representation of the executed behaviour of navigating towards the detected structure  $S_2$  at the end of  $E_3$  search.

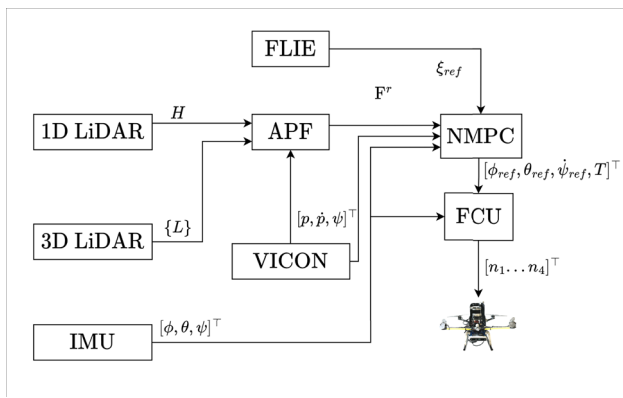
## 5 Mission Configuration

### 5.1 Low Level Autonomy

Evaluation of the proposed autonomy during experimental trials is accompanied by VICON motion capture system providing indoor localization of the UAV. Let the  $\mathbf{x}_{uav}^{vicon} = [\mathbf{p}, \dot{\mathbf{p}}, \mathbf{q}, \dot{\mathbf{q}}]$  be the full state vector of the UAV provided by the motion capture system, where  $\mathbf{p} = [x, y, z]$ ,  $\dot{\mathbf{p}} = [v_x, v_y, v_z]$ ,  $\mathbf{q} = [q_x, q_y, q_z]$ ,  $\dot{\mathbf{q}} = [\omega_x, \omega_y, \omega_z]$ . In this work, we implement a high-level Nonlinear Model Predictive Control (NMPC) that provides control inputs in the form of angle and thrust commands,  $\mathbf{u} = [\phi_{ref}, \theta_{ref}, \psi_{ref}, T]$ , to the Flight Control Unit (FCU) translating the control commands to respective motor velocities,  $\mathbf{n} = [n_1, n_2, n_3, n_4]$  to reach the reference pose (Fig. 9). The low-level autonomy is supplemented with Artificial Potential Field (APF) guided



**Fig. 8** Illustration of commanded behaviour to be executed by the UAV during  $E_3$  search



**Fig. 9** Overview of the implemented controller architecture used in this work

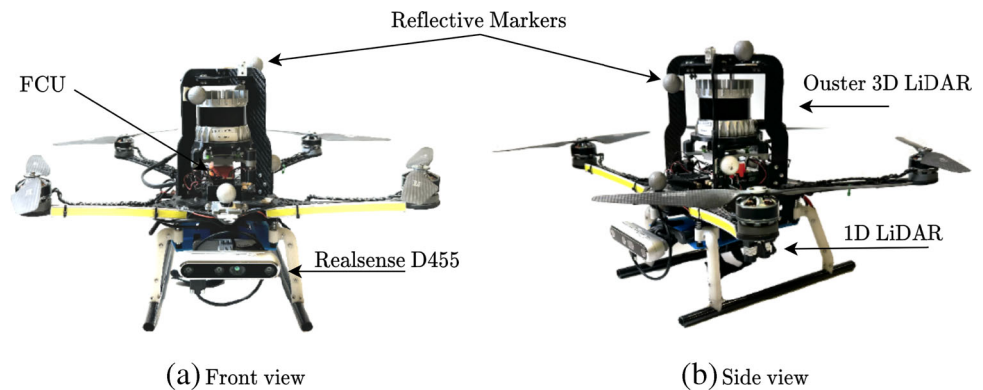
collision avoidance scheme which considers local 3D LiDAR point cloud  $\{L\}$  and range information  $H \in \mathbb{R}$  along  $\mathcal{B}_Z$ -axis from the single beam LiDAR. The APF provides repulsive potential  $F^r = [F_x, F_y, F_z] \in \mathbb{R}$ . A comprehensive discussion on the utilized control methodology can be found in [11, 32].

### 5.2 System Overview

The UAV is equipped with an IntelRealsense D455 stereo camera, a LidarLite v3 single beam lidar to maintain altitude, an Velodyne Ouster 3D LiDAR, an Intel NUC i5 8365U computational board running Robot Operating System (ROS) Noetic and Ubuntu 20.04 operating system (Fig. 10). In addition to that, the platform carries 32GB internal memory and a PixHawk FCU. The  $RGB$  frames capture information within  $90^\circ$  Horizontal FOV and  $65^\circ$  Vertical FOV at 30 Frames Per Second (FPS). The depth point cloud information is down-sampled using a voxel grid filter a leaf size of 0.2 m.

The simulation is executed in an Intel i7 9700K desktop PC with Nvidia Quadro P6000 Graphical Processing Unit (GPU) and 64 GB memory. In the experiments carried out, The PC runs ROS Noetic and Ubuntu 20.04. The framework is written completely in Python language.

**Fig. 10** Graphical depiction of the experimental platform used in this work. The system components mentioned above are demarcated in this figure



### Algorithm 1 FLIE Autonomy.

```

while initialize do
  if  $P_f = \emptyset$  then
     $G^k(P_c) = E_2(\xi_{uav}^k)$ 
     $\xi_{insp} \leftarrow \arg \max(G(P_c, \xi_{expl}))$ 
  else
    /* begin inspection
     $P_f = \text{VisibilityFilter}(P_c)$ 
    while within inspection distance do
       $\gamma_{sim} = \text{SceneRecognition}(I_{c,t})$ 
       $\xi_{insp}^{k+1,k} = \text{ViewPlanner}(P_f, \xi_{uav}^k)$ 
    if  $\gamma_{sim} \geq \gamma_{thresh}$  and  $abs(p_{uav}^k - p_{loop}) \leq \gamma_h$  then
       $p_{uav}^k = p_{uav}^k + v_z \cdot O_V$ 
      Reset  $\gamma_{thresh}$ 
    if  $\max\{P_f(z)\} < p_{uav}(z)$  then
       $G^k(P_c) = E_3(\xi_{uav}^k)$ 
      if  $G \neq \emptyset$  then
         $\xi_{insp} \leftarrow \arg \max(G(P_c, \xi_{expl}))$ 
    else
      /* Switch to exploration mode
    if  $P_f = \emptyset$  then
       $G^k(P_c) = E_1(\xi_{uav}^k)$ 
      if  $G \neq \emptyset$  then
         $\xi_{insp} \leftarrow \arg \max(G(P_c, \xi_{expl}))$ 
      else
         $G^k(P_c) = E_2(\xi_{uav}^k)$ 
        if  $G \neq \emptyset$  then
           $\xi_{insp} \leftarrow \arg \max(G(P_c, \xi_{expl}))$ 
        else
           $G^k(P_c) = E_3(\xi_{uav}^k, \xi_{insp}, \gamma_{sim})$ 
          if  $G \neq \emptyset$  then
             $\xi_{insp} \leftarrow \arg \max(G(P_c, \xi_{expl}))$ 
          else
            initialize = False
            Land
    
```

The behaviour of the proposed FLIE autonomy is provided in Algorithm 1. For the simulation study presented, we consider the desired inspection distance  $r_m = 5\text{m}$ , the horizon for the augmented descriptor database during inspection to be  $N = 5$ , the desired horizontal and vertical overlap parameters to be  $\gamma_H = 0.8$  and  $\gamma_V = 0.5$ . The similarity threshold value during inspection  $\gamma_{thresh}^{insp}$  is set as 0.3 and the threshold value

during exploration  $\gamma_{thresh}^{expl}$  is set as 0.1. The sensor range for  $z_c$  is set to a default value of 10metre and  $\sigma$  is set to be 30 deg.

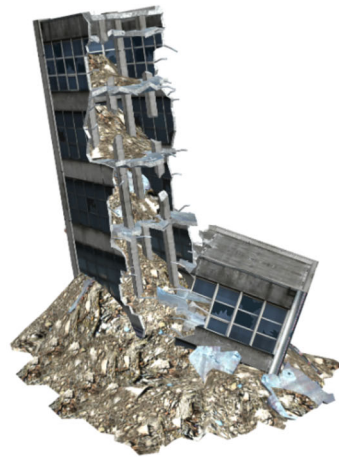
For experimental evaluation, consideration for the limited operational region is taken into account through the generation of a rectangular bounding box with dimensions  $8 \times 4\text{m}$  ( $L \times B$ ) to prevent registration of walls and other equipment. We consider the desired inspection distance  $r_m = 0.9\text{m}$ , the horizon for the augmented descriptor database during inspection to be  $N = 4$ , the desired horizontal and vertical overlap parameters to be  $\gamma_H = 0.8$  and  $\gamma_V = 0.5$ . The similarity threshold value during inspection  $\gamma_{thresh}^{insp}$  is set as 0.2 and the threshold value during exploration  $\gamma_{thresh}^{expl}$  is set as 0.1. During experimental trials, it was noted that lower bound of the similarity score primarily lies within the range of 0.2–0.3. Considering the influence of lighting conditions and noise in optical sensor data, the threshold parameter was set to 0.2 to account for the elements and to ensure proper functioning of the ASR module in real-life conditions. The sensor range for  $z_c$  is set to a conservative value of 2m and  $\sigma$  is set to be 60 deg. Due to safety considerations, the inspection mission was designed to be carried out to be fixed at a single altitude. Thus, no vertical overlap was executed and the effect

of which can be visualized in Figs. 21 and 28. Moreover, as a safety measure due to indoor conditions, ASR module score is supplemented with a conservative positional requirement equal to one  $O_H$  to prevent premature termination as a result of false identifications. However, ASR module results presented in Figs. 23 and 30 show no false positives being reported.

### 5.3 Environment

The simulation is performed using RotorS UAV GAZEBO simulator [33] and ROS [34]. The virtual environment is built using open-source 3D models available in Gazebo [35]. Figure 11 depicts the different models used in this work to emulate an inspect-explore mission scenario around distributed fractured structures. The experimental evaluation is conducted in an indoor laboratory. To emulate open-space sensor behaviour, a virtual bounding box is constructed to isolate the walls from being represented via depth point clouds. We use *EverBlock* building blocks to create structures for visual inspection. The evaluation of the proposed autonomy is performed in both simulation and experimental trials. For experimental evaluations, we consider two main

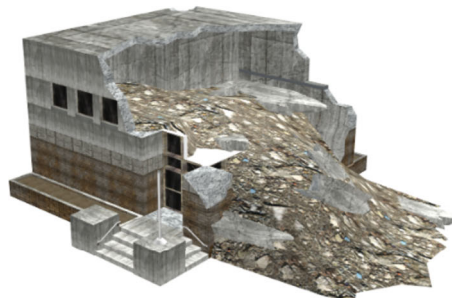
**Fig. 11** Various fractured and dilapidated structures utilized in the virtual environment to evaluate the proposed FLIE autonomy



(a) Collapsed industrial building



(b) Collapsed firestation



(c) Collapsed police station



(d) Collapsed house

scenarios. In scenario one, as shown in Fig. 12, we implement a three-structure inspection use-case wherein the structures are built as individual segmented blocks with varying gaps between them to emulate a discontinuous surface profile. For the second scenario, we consider a two-structure use-case, with one structure possessing a highly irregular shape with large gaps in between any two columns of blocks and a secondary smaller one. Figure 13 represents the structure set up for experimental trials.

The results on computational and memory usage presented are given with respect to percentiles of the total available cores and memory capacity available with regard to each respective system. We utilize open-source *RTAB-Map* [36] and *CloudCompare* software for post-processing and to create the dense RGB-D reconstruction presented in this work. To generate 3D TSDF mesh of the inspected structures at the end of a run, we utilize *Voxblox* [37], a volumetric mapping library to evaluate the qualitative performance of the inspection planner. We also make use of *CloudCompare*, an open-source point-cloud processing project. A video of the experimental evaluation can be found in <https://youtu.be/iYmOJuq1H3g>.

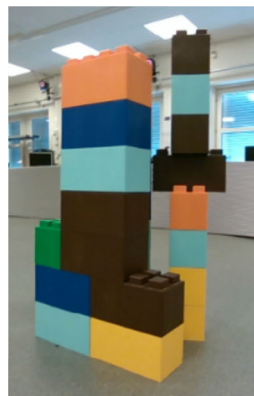
## 6 Results and Discussions

Figure 14 shows the inspection route executed by the FLIE autonomy along with the filtered point cloud  $P_f$  utilized for view planning for the simulated scenario around multiple fractured structures. The influence of irregular surface profiles on the inspection path can be seen alongside the profile-adaptive inspection behaviour of the proposed autonomy on the planned inspection path. Despite the presence of large surface gaps or holes in the 3D models used, the inspection autonomy can be seen as capable to be resilient against such occurrences. Figure 15 presents the desired inspection

distance being maintained throughout the simulated mission. The observed large but infrequent deviations from the desired value is due to the planner executing a correction behaviour when protrusions or gaps from the locally viewed surface are present. The presence of which is reflected in the evaluation of  $p_{poi}$  with a higher value corresponding to the presence of gaps or holes in the current field of view and a lower value corresponding to a protrusion from the object being inspected. In Fig. 16, the performance of ASR module during inspection is provided. Each spike crossing  $\gamma_{thresh}^{insp}$  represents an increment in the inspection height at the instant the UAV recognizes the previously inspected surface which corresponds to the completion of the inspection loop. Figure 17 presents the performance of the ASR module during exploration in order to identify new and uninspected structures for inspection. Since ASR input is considered only when  $P_f \neq \emptyset$ , i.e when a structure lies within the sensor field of view, the data points reflect scoring values at a specific epoch during the simulation run. Referencing the timeline of events in Fig. 16 and comparing with Fig. 17, we can infer successful instances of recognition of previously inspected structures, indicated with high  $\gamma_{sim}^{expl}$  scores and identification of new and uninspected surfaces, indicated with a low score below the threshold value. It is observed that at the end of the  $E_3$  survey around the fourth structure, ASR performs as expected, recognizing the previous inspected structure and finally terminating the mission as a result of no new structures found within the vicinity of the current building

The performance characteristics of FLIE modules with respect to computational load and memory overhead during the simulation run is provided in Figs. 18 and 19. Predominantly, the module utilize less than 20% of available computational power with the median consumption of INSP module being  $\approx 9\%$ , EXPL module being  $\approx 12\%$  and the ASR module being  $\approx 1\%$ . From Fig. 19, the higher median consumption of  $\approx 1.5\%$  for ASR module compared to the other two modules is expected. This is due to the descriptor tree being expanded

**Fig. 12** Experimental setup of the *EverBlock* structures for the case of three distributed structure scenario



(a) Structure 1

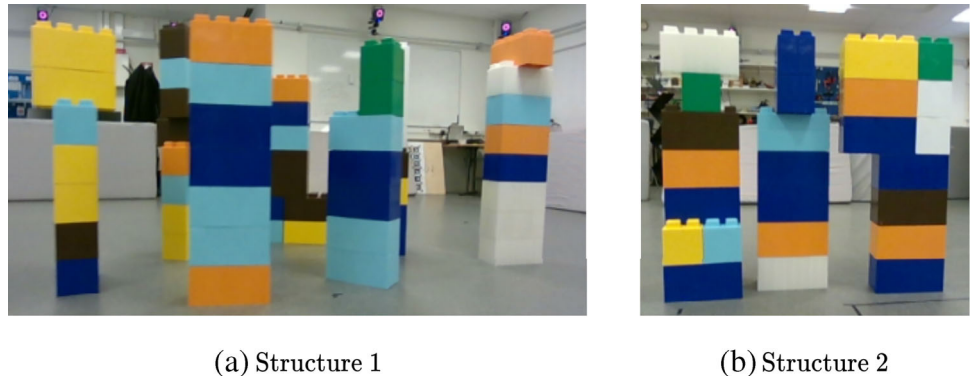


(b) Structure 2

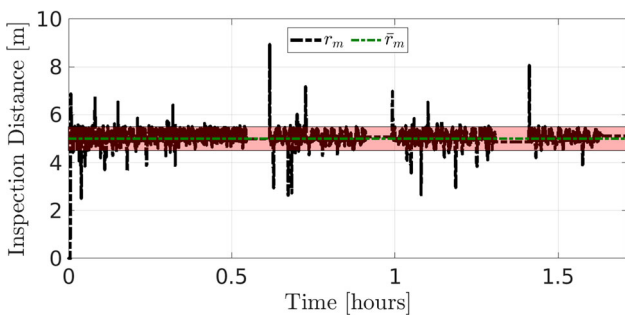
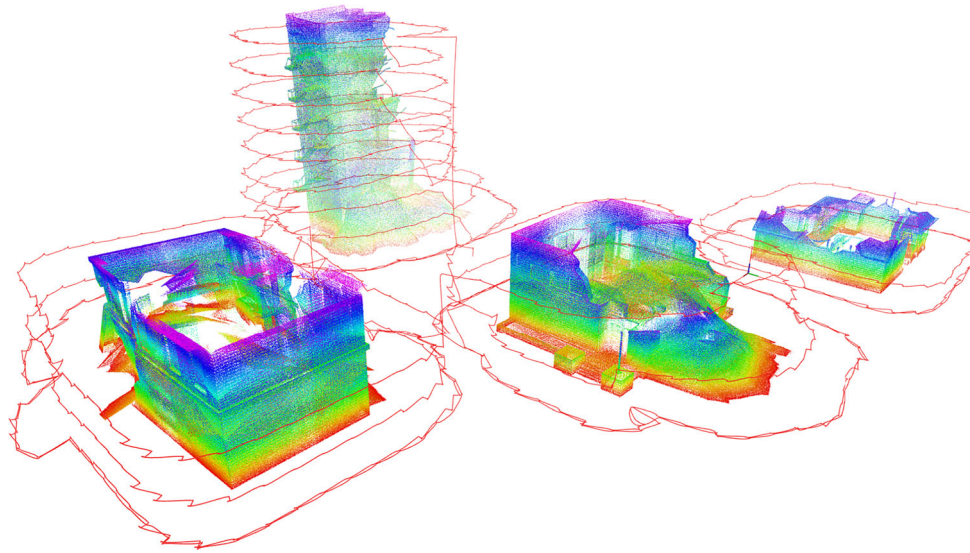


(c) Structure 3

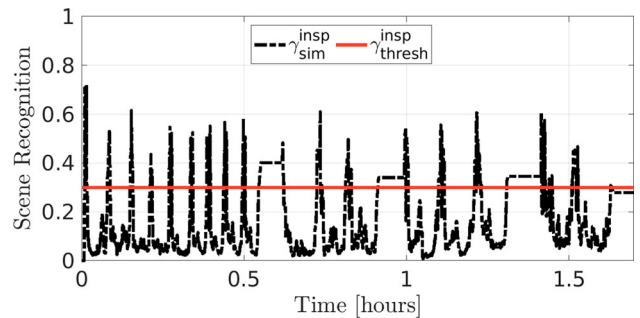
**Fig. 13** Experimental setup of the *EverBlock* structures for the case of two distributed structure scenario with a highly segmented surface profile



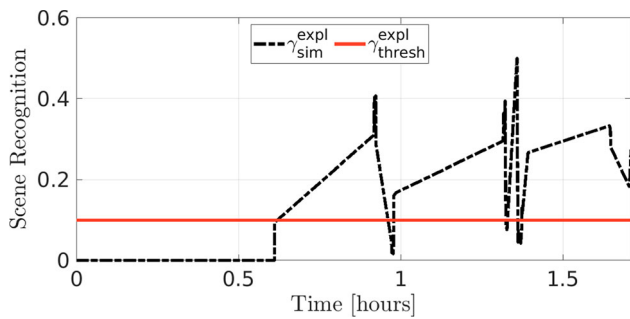
**Fig. 14** 3D inspection trajectory obtained during simulation of FLIE autonomy in presence of four fractured objects located in a distributed manner



**Fig. 15** Performance of desired inspection distance being maintained throughout inspection. In *red* band, the buffer range of  $r_m \pm 0.5m$  considered to update the view pose is shown. In *green*, the mean inspection distance maintained throughout the simulated period is shown



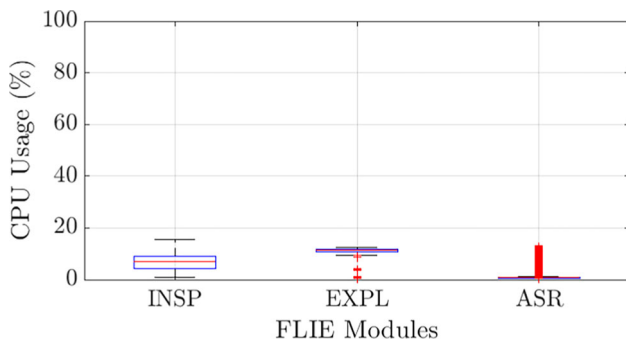
**Fig. 16** Graphical depiction of performance of Active Scene Recognition module during inspection. In *black*, is the similarity score obtained against current and query image and in *orange*, the threshold value considered for similar surfaces during the simulation run is given



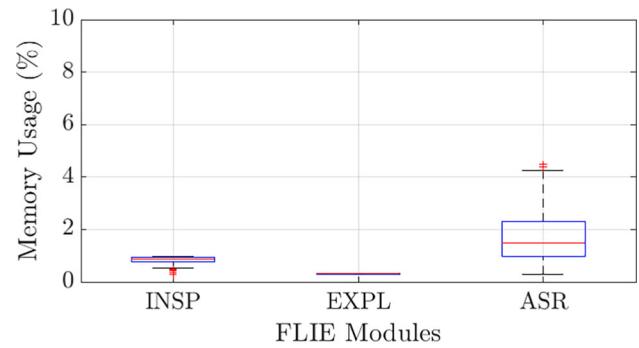
**Fig. 17** Graphical depiction of performance of Active Scene Recognition module during exploration. In *black*, is the similarity score obtained and in *red*, the threshold value considered for previously inspected structures during the simulation run is shown

during the inspection as a result of the augmented descriptor matrix.

Figure 20 displays the planned inspection route around the structures overlaid with the executed trajectory by the UAV. Figure 21 presents the reconstructed mesh through RGB-D images collected during the experimental run for the case of three structures. The mesh visualized can be seen to have a dense reconstruction in the regions of the executed visual inspection loop. However, since a vertical overlap was not executed, i.e an increment in the inspection height to satisfy desired overlap parameters, the top portion of the structure can be seen to have a sparse reconstruction. On the hand, it is evident that the UAV detects and inspects all three structures located in its operational vicinity. The total inspected volume obtained Fig. 21 is 4.014 cubic metres. The inspection distance maintained during inspection is given in Fig. 22. The UAV keeps an average inspection distance of 0.7706 m. We implement a buffer distance of  $\pm 0.2m$  from  $r_m$  indicated in Fig. 22 to update the inspection pose. During inspection of the *EverBlock* structures, we can infer that the autonomy corrects and maintains the required inspection distance based on the traced saw-tooth profile after



**Fig. 18** Performance characteristics of the FLIE modules with respect to computational load required during simulation. INSP represents the inspection module, EXPL refers to the exploration module and the active scene recognition module is shown as ASR

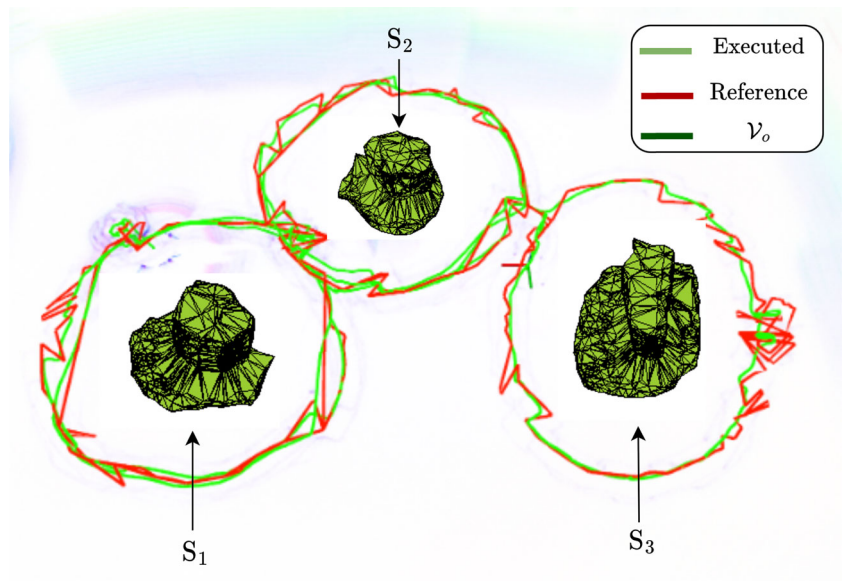


**Fig. 19** Performance characteristics of the FLIE modules with respect to memory overhead consumed during the simulation run. INSP represents the inspection module, EXPL refers to the exploration module and the active scene recognition module is shown as ASR

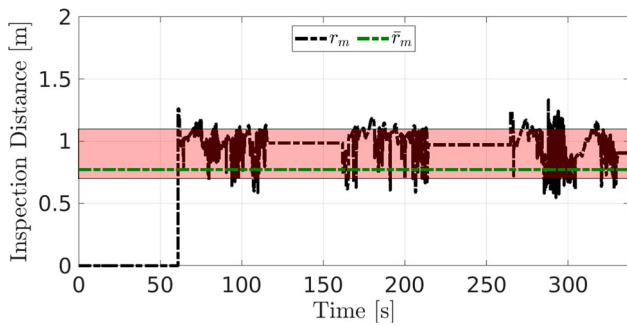
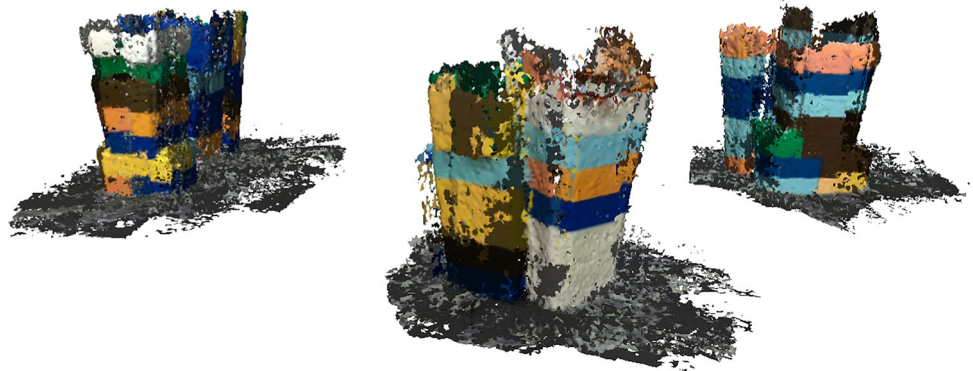
exceeding the buffer value. Since the framework depends on the depth point clouds to find the representative  $p_{poi}$ , the presence of noise can affect the performance of the inspection framework. This is reflected in the figure, where it is seen as large infrequent spikes deviating from the desired baseline inspection distance. Towards the end of the inspection run around the third object, the effect of noisy point clouds can also be seen in the commanded inspection pose oscillating back and forth. Figure 23 presents the performance of the ASR module for the inspection around three distributed structures. The instances of  $\gamma_{sim}^{insp}$  crossing  $\gamma_{thresh}^{insp}$  during the initial period of inspections is a result of ASR continuously comparing the current image frame with the augmented descriptor matrix being built until  $N$  horizon from the start of inspection of the structure. Subsequent to this, the similarity score drops below the threshold value until the UAV approaches the inspection engagement position again where the similarity score increases until it crosses the threshold value and proceeds to execute  $E_3$  survey. This behaviour is repeated for the other two structures.

Figure 24 depicts the performance of ASR module during exploration, i.e in search and identification of new and uninspected structures. At the epoch of successful inspection of structure 1,  $S_1$ , the autonomy executes  $E_3$  policy. Subsequent to which structure 2,  $S_2$ , is detected and inspection is carried out. Comparing timelines between Figs. 23 and 24, it can be observed that during  $E_3$  search around the second structure, it successfully recognizes structure 1, indicated with the highest  $\gamma_{sim}^{insp}$  score of 0.35 and proceeds to detects and recognize the third structure,  $S_3$ , as new and uninspected. It is shown with a low similarity score of 0.08. Due to reduced battery level, the experimental run had to be terminated at the end of inspection of  $S_3$  without completing a  $E_3$  search. This can be cross-referenced in Fig. 23 at the same timeline where  $\gamma_{sim}^{insp}$  crosses the threshold value indicating recognition of previously inspected surface and subsequently executing the tertiary exploration policy.

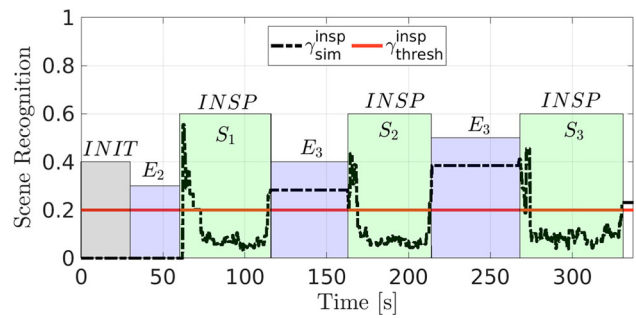
**Fig. 20** Graphical plot of the executed inspection route around three structures. In *red*, the reference inspection path generated by the planner is shown along with the actual travelled path shown here in *green*



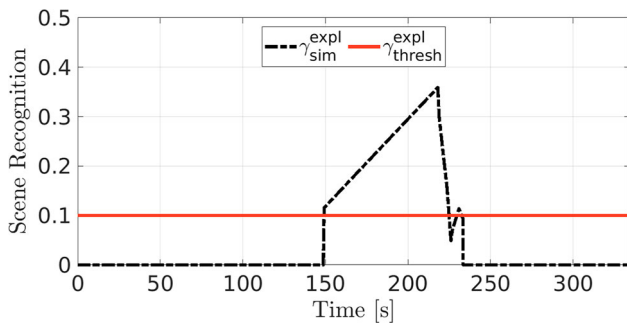
**Fig. 21** RGB-D reconstruction of the inspected structures for the case of three structure inspection



**Fig. 22** Graphical plot of the inspection distance maintained during the case of three distributed structures. In *black*, is the actual inspection distance maintained during inspection. In *green*, the mean inspection distance maintained throughout the run is shown. The buffer distance of  $r_m \pm 0.2m$  considered to update the inspection pose is shown here in *red*



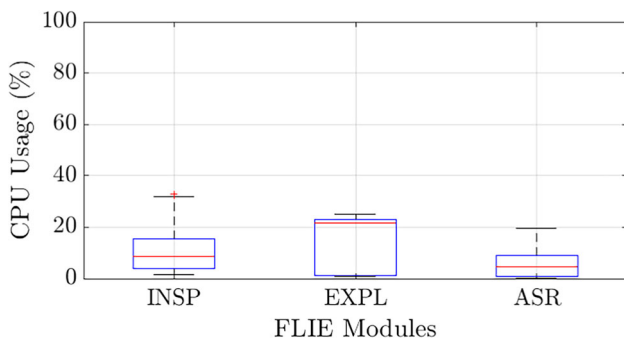
**Fig. 23** Performance visualization of ASR module during inspection run. INIT, shown here in *grey*, denotes the initialization period for the experimental run. The specific exploration behaviours exhibited are referenced to as  $E_2$  and  $E_3$ , shown in *blue*, during their corresponding execution period. The regions of exhibited inspection behaviour are denoted as INSP and shown in *green*



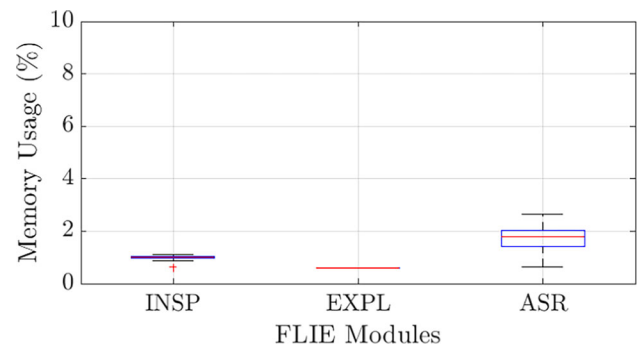
**Fig. 24** Performance visualization of ASR module during exploration for the case of three structures is shown. In *black*, the obtained similarity scores between the query descriptor and the descriptor tree during the run is shown. While in *red*, the threshold value considered for a structure to be unique is shown

Figures 25 and 26 display the computational and memory load respectively consumed during the experimental run around three structures. The usage of available computational resources for all three modules follow a similar trend as seen in Fig. 18. From the Fig. 25, we can infer that the median consumption of INSP module to be  $\approx 9\%$ . The EXPL module is observed to have the highest consumption of all three reserving  $\approx 22\%$  with ASR module using  $\approx 5\%$  of the available CPU resources. From Fig. 26, the effect of the implemented descriptor-tree-based scene recognition module can be observed with the ASR module possessing the highest percentile of memory usage out of the three modules with the median overhead  $\approx 2\%$ . Both INSP and EXPL modules present low usage percentiles of 5% and 3% respectively. Reflecting on the previous result obtained during the simulation run in Fig. 19, the observed behaviour in Fig. 26 can be inferred to follow a similar trend.

Figure 27 displays the planned inspection route around the two structures overlaid with the actually executed trajectory by the UAV. The inspection autonomy can be seen to



**Fig. 25** Performance characteristics of the FLIE modules with respect to computational load required for the scenario of inspection of three structures. INSP represents the inspection module, EXPL refers to the exploration module and the active scene recognition module is shown as ASR



**Fig. 26** Performance plot of the resulting memory overhead during the experimental run for the scenario of three distributed structures. INSP, EXPL and ASR terms retain their aforementioned definitions

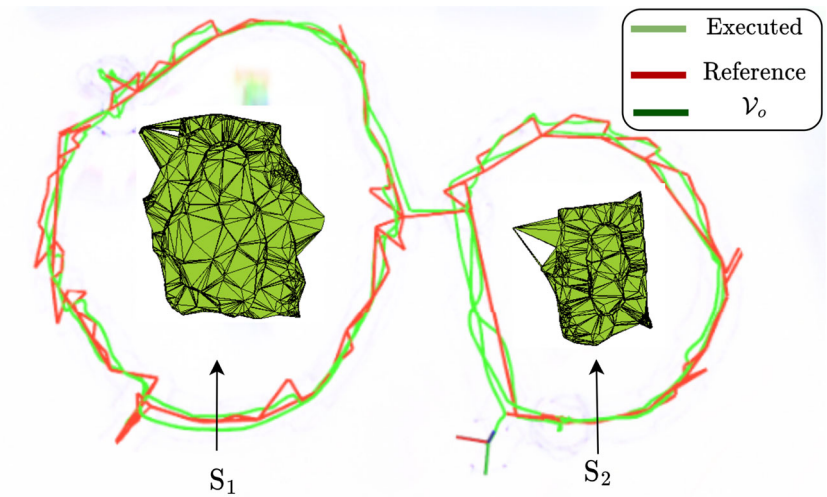
be resilient against the effect of a highly segmented surface profile of  $S_1$  and completes the desired task successfully. In addition to that, the planned path can be observed to adapt to the locally viewed surface apart from exhibiting corrections to maintain desired inspection distance, seen in the figure with a sharp deviation from previous inspection behaviour. In Fig. 28, the reconstructed 3D mesh from collected RGB-D images during inspection is shown. As the mission configuration was designed for the UAV to complete one inspection pass, the results obtained possess characteristics similar to Fig. 21 with dense mesh seen around the inspection regions and sparse representations on the upper portion of the structures. The total inspected volume obtained Fig. 28 is 6.5173 cubic metres.

The maintained inspection distance during the inspection run is shown in Fig. 29. A mean viewing distance of 0.7497 m is observed throughout the duration of the mission. The correction characteristics mentioned above can be clearly inferred from the saw-tooth profile of inspection distance when it exceeds the buffer range. The few outliers in the form of sharp spikes, during inspection, are caused primarily due to the presence of noise where the evaluation of  $p_{poi}$  is affected. However, we can observe the inspection behaviour revert back within bounds immediately after the event.

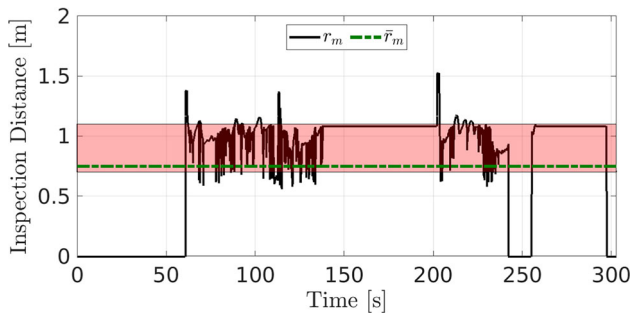
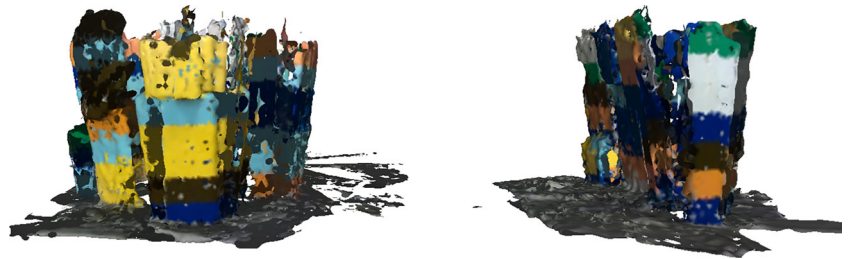
Figure 30 shows the performance of ASR module during inspection mode. The specific behaviour exhibited during the mission is demarcated in the plot. Similar to Figs. 16 and 23, the autonomy executes  $E_2$  search after initialization to detect and tag available structures in its current vicinity. Subsequent to this, the expected high similarity scores during the initial period of inspection are attributed to the comparison performed on the actively built descriptor tree until the desired  $N$  horizon after which it falls below the threshold value. As ASR recognizes a previously inspected surface, at the end of the inspection loop, indicated in the figure with a high similarity score, it proceeds to execute  $E_3$  search. This is observed in Fig. 31, where the framework identifies the second structure as new and uninspected, indicated with a low



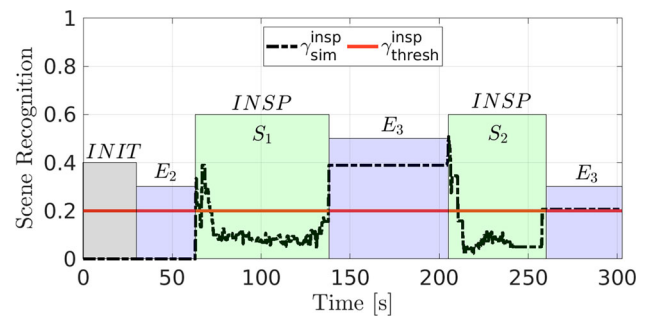
**Fig. 27** Graphical plot of the executed inspection route around two structures. In *red*, the reference inspection path generated by the planner is shown along with the actual travelled path shown here in *green*



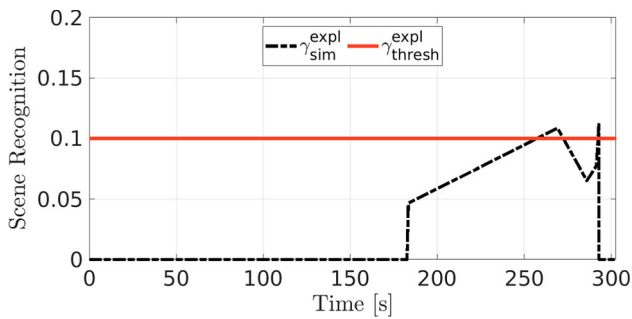
**Fig. 28** RGB-D reconstruction of the inspected structures for the case of dual structure inspection



**Fig. 29** Graphical plot of the inspection distance maintained during the case of two distributed structures. In *black*, is the actual inspection distance maintained during inspection. In *green*, the mean inspection distance maintained throughout the run is shown. The buffer distance of  $r_m \pm 0.2m$  considered to update the inspection pose is shown here as *red* band



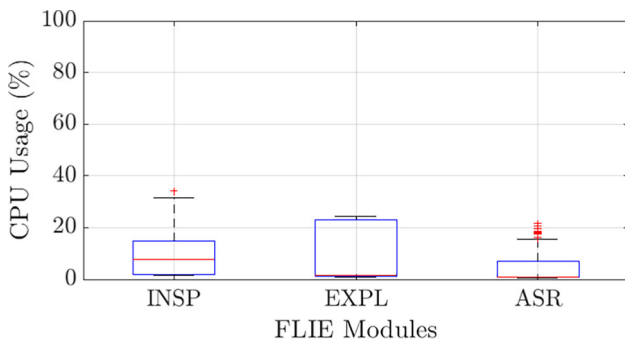
**Fig. 30** Performance visualization of ASR module during inspection run for the two structure scenario is shown. INIT, shown here in grey, denotes the initialization period for the experimental run. The specific exploration behaviours exhibited are referenced to as  $E_2$  and  $E_3$ , shown in blue, during their corresponding execution period. The regions of exhibited inspection behaviour are denoted as INSP and shown in green



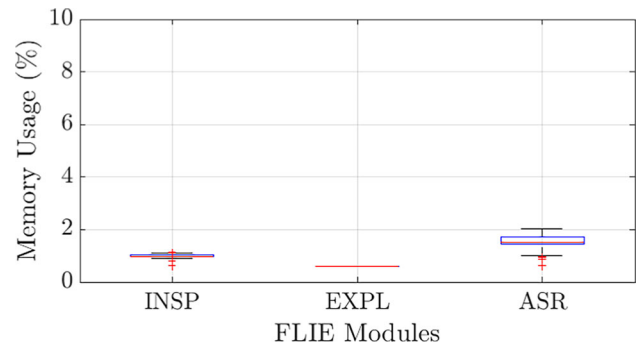
**Fig. 31** Performance visualization of ASR module during exploration for the case of two structures is presented. In black, the obtained similarity scores between the query descriptor and the descriptor tree during the run is shown. While in red, the threshold value considered for a structure to be unique is shown

score, and proceeds to inspect the said structure. The autonomy exhibits the expected response with the recognition of the previously inspected surface at the end of the inspection loop and proceeds to execute  $E_3$  policy. Referencing timelines with Fig. 30, the recognition of  $S_1$  during  $E_3$  search is inferred with a score above the threshold value in Fig. 31 and ignores  $S_1$  as expected. Faulty registration of noise as the presence of a potential new structure can be observed with the ASR providing a low score during exploration. This leads to the autonomy to carry out inspection by travelling to the target pose at which instant the mission was terminated manually. The effect of this outlier can be observed in Fig. 27 with the UAV seen at its final position en-route to the falsely registered structure.

Figures 32 and 33 present the computational and memory overhead reserved during the mission run for the second scenario. The mean consumption observed for both the plots follow previously seen behaviour. INSP module has the highest median percentile of CPU usage of  $\approx 10\%$  followed by EXPL module with  $\approx 2\%$  and ASR exhibiting the lowest median consumption of  $\approx 1\%$  and a maximum usage of  $\approx$



**Fig. 32** Performance characteristics of the FLIE modules with respect to computational load required during inspection of two structures. INSP represents the inspection module, EXPL refers to the exploration module and the active scene recognition module is shown as ASR



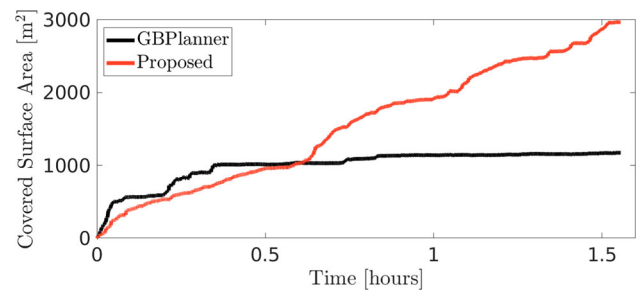
**Fig. 33** Performance plot of the resulting memory overhead during the experimental run for the scenario of two distributed structures. INSP, EXPL and ASR terms retain their aforementioned definitions

18%. In terms of memory overhead, ASR reserves a median amount of  $\approx 1.8\%$  followed by INSP module with  $\approx 1\%$  and EXPL module with  $\approx 0.8\%$ .

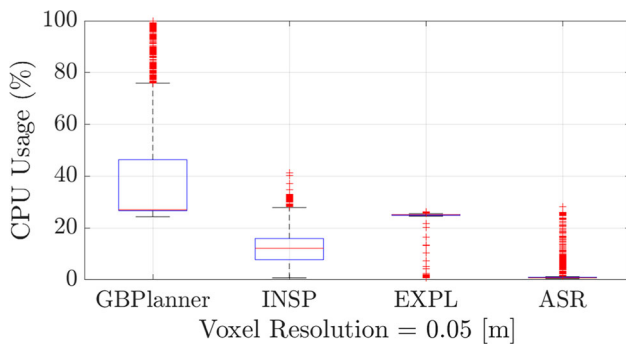
### 6.1 Performance Comparison

The authors would like to note that reproducible state-of-art frameworks in a similar field was restricted to exploration-centric methodology, primarily the work presented in [10]. Frontier-based methods such as the work in [38], Next Best View planner [17] and Exploration planner [39] faced compatibility issues. Because of the compounding effect resulting from the use of a different sensor configuration, the contrast in the density of the sampled sensor point-cloud information and the difference in intended use-case, *3D bounded volume vs open-world*, the comparison presented against [10] is not in equal-grounds. However, the framework in [10] was tuned to the best of our capability to ensure an overall better performance in the simulated urban world. Moreover, implementation of the related state-of-art open-source repositories [24, 40] with *UnrealEngine* [41] and *AirSim* [42] is currently incompatible with our established *Gazebo* [35] and *ROS* [34] setup.

Figures 34, 35 and 36 present the comparison characteristics for a desired TSDF mesh resolution of 0.05 m of the

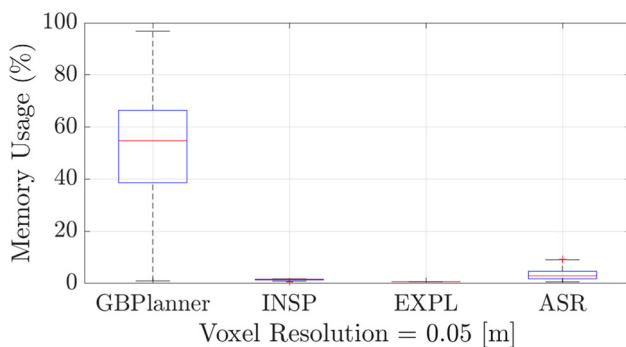


**Fig. 34** Graphical performance plot between *GBPlanner* (black) and *Proposed* (red) autonomy for covered surface area with a desired TSDF voxel resolution of 0.05 m during the simulation run

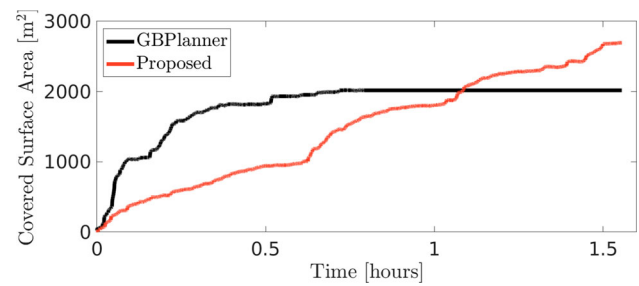


**Fig. 35** Performance characteristics *GBPlanner* and *FLIE* modules in terms of consumed computational resources during the simulation run for a desired TSDF voxel resolution of 0.05 m

structures mentioned prior in Fig. 11. Figure 34 presents the total covered surface area determined over the length of the simulation run by the two frameworks. While *GBPlanner* initially attains higher coverage, as is the expected behaviour from an exploration framework, the proposed autonomy surpasses it and continues to register new and uninspected surfaces. An expected correlation can be seen with regards to the chosen TSDF voxel size and computational burden from Figs. 35 and 36. In Fig. 35, *FLIE* modules show an overall median utilization of approximately 13% compared to *GBPlanner* which utilized approximately 27% of available CPU resources. A similar trend can be observed in Fig. 36 where *FLIE* modules have been recorded to consume an overall median amount of 1.7% compared to the significantly higher amount of 54% for *GBPlanner* in terms of memory overhead during planning. The stark contrast in the registered computational load for *GBPlanner* is mainly due to the planning implemented in a global map. Thus, refining the resolution of the required TSDF voxel size increases the necessary map size to be stored in memory as well as the



**Fig. 36** Performance characteristics *GBPlanner* and *FLIE* modules in terms of total memory overhead during the simulation run for a desired TSDF voxel resolution of 0.05 m

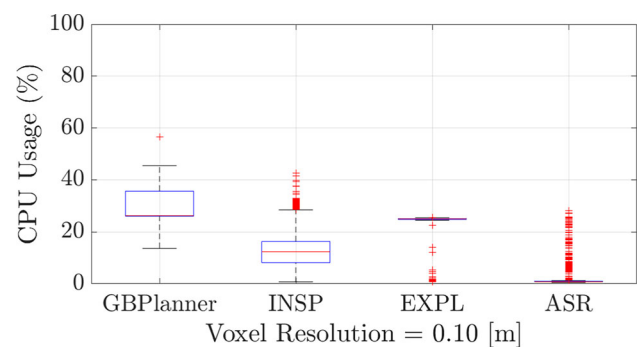


**Fig. 37** Graphical performance plot between *GBPlanner* (black) and *FLIE* (red) autonomy for covered surface area with a desired TSDF voxel resolution of 0.1 m during the simulation run

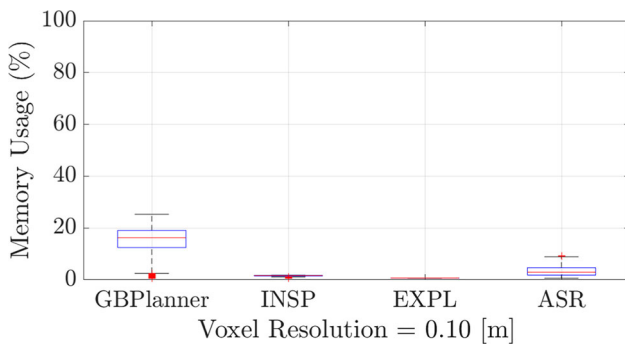
necessary computations performed for collision checks and path planning in a dense map.

Figures 37, 38 and 39 present the comparison characteristics for a desired TSDF mesh resolution of 0.1 m. As in Fig. 34, *GBPlanner* exhibits similar performance with an initially higher coverage within a shorter time but plateaus for the remaining period in Fig. 37. The proposed framework can be seen to register higher inspected surfaces gradually over time and surpasses *GBPlanner* to complete inspection of all structures available within the deployed region.

From Fig. 38, the *FLIE* modules register approximately the same as in Fig. 35, an overall median amount of 13% compared to *GBPlanner* which registered a similar rate as before, around 26%. However, the maximum consumption has dropped to  $\approx 45\%$  from  $\approx 75\%$ , as seen in Fig. 35, for *GBPlanner* whereas for *FLIE* modules it stays relatively the same. In Fig. 39, the effect of increasing desired voxel size is more visible. *GBPlanner* registers  $\approx 16\%$  of median memory consumption compared to *FLIE* modules which registered an overall median consumption of  $\approx 1.7\%$ , similar to Fig. 36. The similarity in performance characteristics of the proposed *FLIE* autonomy for both runs can be attributed to the map-decoupled approach addressed in the work. Thus,



**Fig. 38** Performance characteristics *GBPlanner* and *FLIE* modules in terms of consumed computational resources during the simulation run for a desired TSDF voxel resolution of 0.1 m

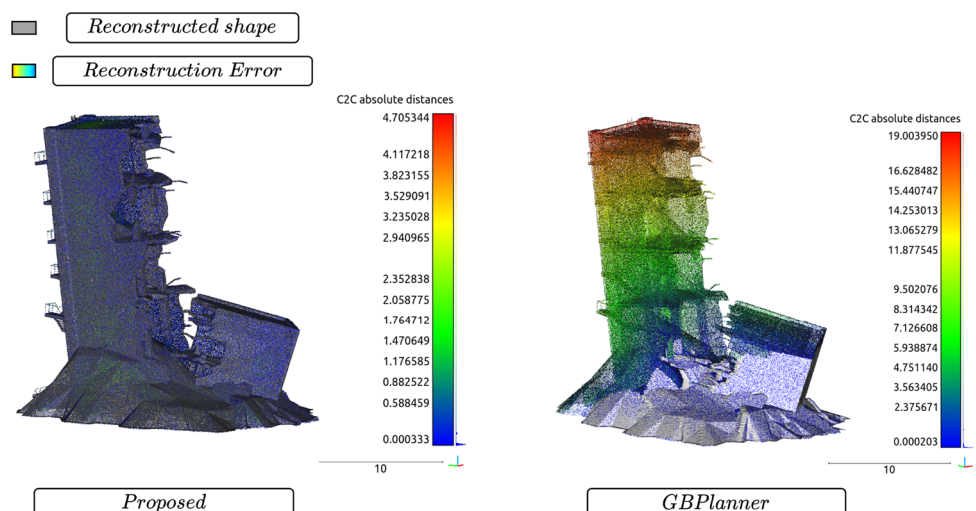


**Fig. 39** Performance characteristics *GBPlanner* and *FLIE* modules in terms of total memory overhead during the simulation run for a desired TSDF voxel resolution of 0.1 m

isolating the performance of the autonomy based on variations in desired map characteristics.

Figures 40, 41, 42 and 43 present a qualitative analysis of the performance of both frameworks in terms of 3D surface reconstruction error when compared with ground truth data. The proposed framework scores higher with a maximum reconstruction error of 4.7 m compared to *GBPlanner* with a maximum error 19.0 m in Fig. 40. Similarly, *FLIE* framework produces reconstruction error value of 3.7 m compared to *GBPlanner* with an error of 10.59 m in Fig. 41, a 3.4 m maximum deviation compared to 6.19 m in Fig. 42 and a value of 2.7 m compared to 3.12 m in Fig. 43. Influenced by sensing restrictions of a stereo camera, the most common regions of reconstruction error can be seen to be the top-flat surface for all four meshes, with additional regions located deep into the voids and outside sensor range in each level for the case of inspecting the industrial building. Thus, across

**Fig. 40** Qualitative analysis of the covered surface area (voxel size = 0.05 m) of the collapsed industrial building in terms of reconstruction error with ground truth mesh



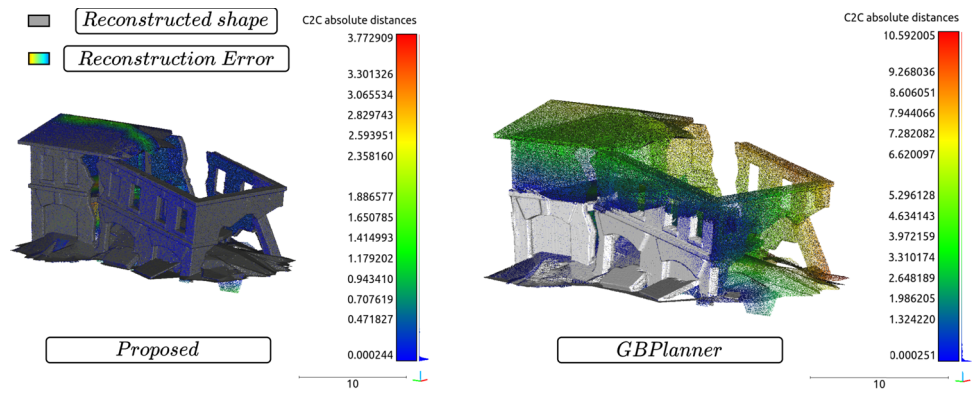
(a) Collapsed industrial building

both quantitative and qualitative aspects of measured performance, the proposed *FLIE* framework achieves much higher score than *GBPlanner*.

### 7 Limitations

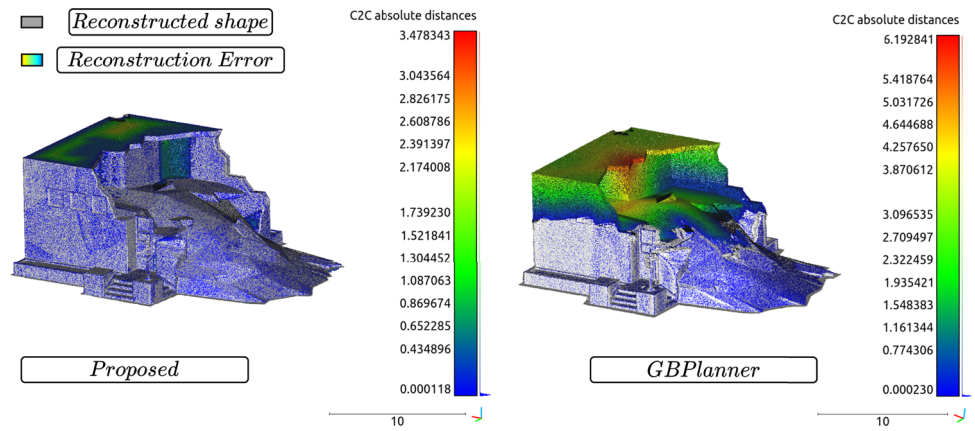
The proposed baseline autonomy provides a planning solution based on a single optical sensor, primarily, RGB image and depth point-cloud sensor measurements. As such, it is susceptible to false-positives measurements which can affect view-planning. To address this, the current autonomy down-samples raw point-cloud measurements through a centroid-based voxel grid prior to its utilization in view-planning. However, during experimental trials, presence of noise in depth-cloud was seen during view-planning irrespective of the use of the voxel grid filter. Moreover, the utilized platform features a fixed and undamped setup of the optical sensor. Thus, vibrations induced by on-board motors or external disturbances caused by wind-gusts acting on the UAV can impact mission performance. Through the use of a gimbal mechanism and an external observer as in [43], external influences can be isolated and avoided. External uncertainties in localization is a primary concern in the field of robotics and can affect mission performance. This work partially address the limitation faced by presenting a map-decoupled approach for view-planning. However, it is regarded as part of future scope to incorporate vision-based localization source for field trials as the scene recognition provides a synergistic compatibility with a vision-based localization module which can benefit from the loop-closure behaviour exhibited during inspection.

**Fig. 41** Qualitative analysis of the covered surface area (voxel size = 0.05 m) of the collapsed fire station in terms of reconstruction error with ground truth mesh

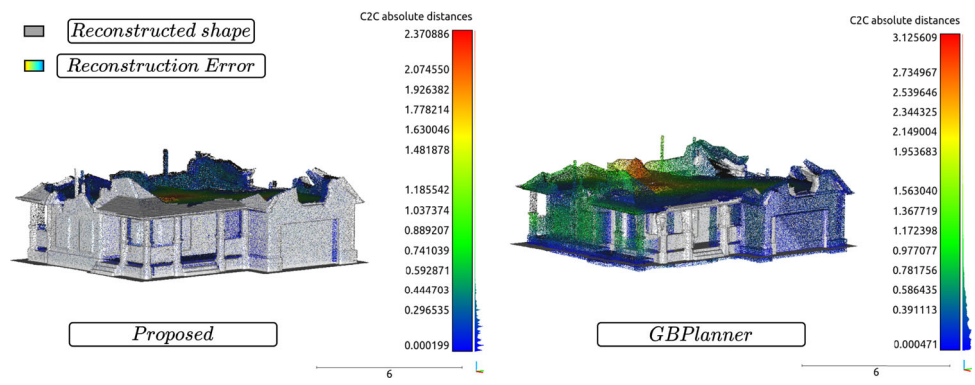


(b) Collapsed fire station

**Fig. 42** Qualitative analysis of the covered surface area (voxel size = 0.05 m) of the collapsed police station in terms of reconstruction error with ground truth mesh



**Fig. 43** Qualitative analysis of the covered surface area (voxel size = 0.05 m) of the collapsed house in terms of reconstruction error with ground truth mesh



## 8 Conclusions and Future Work

In this work, we presented a unified aerial inspect-explore autonomy capable to execute a close visual inspection of structures along with a staged global exploration strategy to detect and tag available structures. The proposed autonomy is shown to be robust against surface discontinuities such as the presence of large gaps or holes in addition to exhibiting a safe and stable inspection behaviour around fractured structures. Moreover, catalogued images collected during inspection is used to successfully recognize previously inspection structures and identify potential new structures. Thus, proving the efficacy of a map-independent planning approach. The current framework is shown to be able to operate in a completely unknown environment to inspect apriori unknown structures with an overall low computational and memory footprint compared to map-based state-of-art approach. The advantages of a reactive pose synthesis scheme are presented with extensive studies conducted both in simulation and experimentally around severely damaged structures and highly segmented columns of *EverBlock* structures. The proposed *FLIE* autonomy scores higher across the board for both qualitative and quantitative performance characteristics based on the comparison presented. As part of future work, field deployments are considered to incorporate visual-inertial based localization with the ASR module to transition to real-life scenarios and to address odometry drift by leveraging collected image database.

**Acknowledgements** This work has been partially funded by the European Union's Horizon 2020 Research and Innovation Programme under the Grant Agreement No. 101003591 NEXGEN SIMS. The authors acknowledge Ilias Tevetzidis contribution towards this work by assuming the role of safety pilot during experimental trials.

**Author Contributions** Vignesh Kottayam Viswanathan: Development of manuscript, primary research contributor and experimental work; Björn Lindqvist: Experimental work, manuscript contributions and advisory role; Sumeet Gajanan Satpute, Christoforos Kanellakis and George Nikolakopoulos: Manuscript contributions and advisory.

**Funding** Open access funding provided by Lulea University of Technology. This work has been partially funded by the European Union's Horizon 2020 Research and Innovation Programme under the Grant Agreement No. 101003591 NEXGEN SIMS.

**Availability of data and materials** ROSbags of sensor and submodule output data from the experiments can be made available at the suggestion of the reviewers and editors.

**Code Availability** Not applicable

## Declarations

**Ethics approval** Not applicable

**Consent to participate** All authors have read and agreed to publish this work.

**Consent for publication** All authors have read and agreed to publish this work.

**Conflict of interest** The authors declare that they have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, 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 changes were made. 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/4.0/>.

## References

1. Viswanathan, V.K., Satpute, S., Banerjee, A., Nikolakopoulos, G.: Nonlinear model predictive control based cooperative stereo-visual coverage of an asteroid. In: 2022 American control conference (ACC), pp. 5360–5367 (2022). IEEE
2. Viswanathan, V.K., Papadimitriou, A., Banerjee, A., Mansouri, S.S., Nikolakopoulos, G.: Exogenous disturbance estimation for autonomous navigation around small celestial bodies. In: 2022 IEEE 61st conference on decision and control (CDC), pp. 3760–3766 (2022). IEEE
3. Khaloo, A., Lattanzi, D., Jachimowicz, A., Devaney, C.: Utilizing uav and 3d computer vision for visual inspection of a large gravity dam. *Front. Built Environ.* 31 (2018)
4. Chan, B., Guan, H., Jo, J., Blumenstein, M.: Towards uav-based bridge inspection systems: A review and an application perspective. *Struct. Monit. Maint.* 2(3), 283–300 (2015)
5. Kanellakis, C., Mansouri, S.S., Fresk, E., Kominiak, D., Nikolakopoulos, G.: Cooperative uavs as a tool for aerial inspection of large scale aging infrastructure. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp. 5040–5040 (2018). IEEE
6. Addabbo, P., Angrisano, A., Bernardi, M.L., Gagliarde, G., Mennella, A., Nisi, M., Ullo, S.L.: Uav system for photovoltaic plant inspection. *IEEE Aerosp. Electron. Syst. Mag.* 33(8), 58–67 (2018)
7. Zhang, Y., Yuan, X., Li, W., Chen, S.: Automatic power line inspection using uav images. *Remote Sens.* 9(8), 824 (2017)
8. Patel, A., Lindqvist, B., Kanellakis, C., Nikolakopoulos, G.: Fast planner for mav navigation in unknown environments based on adaptive search of safe look-ahead poses. In: 2022 30th Mediterranean conference on control and automation (MED), pp. 545–550 (2022). IEEE
9. Mansouri, S.S., Kanellakis, C., Fresk, E., Lindqvist, B., Kominiak, D., Koval, A., Sopsakis, P., Nikolakopoulos, G.: Subterranean mav navigation based on nonlinear mpc with collision avoidance constraints. *IFAC-PapersOnLine* 53(2), 9650–9657 (2020)
10. Dang, T., Tranzatto, M., Khattak, S., Mascari, F., Alexis, K., Hutter, M.: Graph-based subterranean exploration path planning using aerial and legged robots. *J. Field Robot.* 37(8), 1363–1388 (2020). Wiley Online Library

11. Lindqvist, B., Kanellakis, C., Mansouri, S.S., Agha-mohammadi, A.-A., Nikolakopoulos, G.: Compra: A compact reactive autonomy framework for subterranean mav based search-and-rescue operations. *J. Intell. Robot. Syst.* **105**(3), 1–21 (2022)
12. Kumar, A.: Real-time performance comparison of vision-based autonomous landing of quadcopter on a ground moving target. *IETE J. Res.* pp. 1–18 (2021)
13. Viswanathan, V.K., Satpute, S.G., Lindqvist, B., Kanellakis, C., Nikolakopoulos, G.: Experimental evaluation of a geometry-aware aerial visual inspection framework in a constrained environment. In: 2022 30th Mediterranean conference on control and automation (MED), pp. 468–474 (2022). IEEE
14. Kerle, N., Nex, F., Gerke, M., Duarte, D., Vetrivel, A.: Uav-based structural damage mapping: A review. *ISPRS Int. J. Geo-inform.* **9**(1), 14 (2019)
15. Lauterbach, H.A., Koch, C.B., Hess, R., Eck, D., Schilling, K., Nüchter, A.: The eins3d project-instantaneous uav-based 3d mapping for search and rescue applications. In: 2019 IEEE international symposium on safety, security, and rescue robotics (SSRR), pp. 1–6 (2019). IEEE
16. Cheng, M.-L., Matsuoka, M., Liu, W., Yamazaki, F.: Near-real-time gradually expanding 3d land surface reconstruction in disaster areas by sequential drone imagery. *Autom. Constr.* **135**, 104105 (2022)
17. Bircher, A., Kamel, M., Alexis, K., Oleynikova, H., Siegwart, R.: Receding horizon path planning for 3d exploration and surface inspection. *Auton. Robot.* **42**(2), 291–306 (2018)
18. Brogaard, R.Y., Andersen, R.E., Kovac, L., Zajaczkowski, M., Boukas, E.: Towards an autonomous, visual inspection-aware 3d exploration and mapping system for water ballast tanks of marine vessels. In: 2021 IEEE international conference on imaging systems and techniques (IST), pp. 1–6 (2021). IEEE
19. Yoder, L., Scherer, S.: Autonomous exploration for infrastructure modeling with a micro aerial vehicle. In: *Field and service robotics*, pp. 427–440. Springer, (2016)
20. Faria, M., Ferreira, A.S., Pérez-Leon, H., Maza, I., Viguria, A.: Autonomous 3d exploration of large structures using an uav equipped with a 2d lidar. *Sens.* **19**(22), 4849 (2019)
21. Song, S., Kim, D., Jo, S.: Online coverage and inspection planning for 3d modeling. *Auton. Robot.* **44**(8), 1431–1450 (2020)
22. 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'*, pp. 146–151 (1997). IEEE
23. Song, S., Jo, S.: Surface-based exploration for autonomous 3d modeling. In: 2018 IEEE international conference on robotics and automation (ICRA), pp. 4319–4326 (2018). IEEE
24. Schmid, L., Pantic, M., Khanna, R., Ott, L., Siegwart, R., Nieto, J.: An efficient sampling-based method for online informative path planning in unknown environments. *IEEE Robot. Autom. Lett.* **5**(2), 1500–1507 (2020)
25. Smith, N., Moehrl, N., Goesele, M., Heidrich, W.: Aerial path planning for urban scene reconstruction: A continuous optimization method and benchmark (2018)
26. Hepp, B., Nießner, M., Hilliges, O.: Plan3d: Viewpoint and trajectory optimization for aerial multi-view stereo reconstruction. *ACM Trans. Graph. (TOG)* **38**(1), 1–17 (2018)
27. Song, S., Kim, D., Choi, S.: View path planning via online multi-view stereo for 3-d modeling of large-scale structures. *IEEE Trans. Robot.* **38**(1), 372–390 (2021)
28. Viswanathan, V.K., Satpute, S.G., Lindqvist, B., Nikolakopoulos, G.: First-look enabled autonomous aerial visual inspection of geometrically fractured objects in constrained environments. In: 2022 IEEE 31st international symposium on industrial electronics (ISIE), pp. 295–300 (2022). IEEE
29. Viswanathan, V.K., Satpute, S.G., Nikolakopoulos, G.: Flie: First-look enabled inspect-explore autonomy towards visual inspection of unknown distributed and discontinuous structures. *IEEE Access* (2023)
30. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**, 91–110 (2004)
31. Muja, M., Lowe, D.G.: Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)* **2**(331–340), 2 (2009)
32. Lindqvist, B., Haluska, J., Kanellakis, C., Nikolakopoulos, G.: An adaptive 3d artificial potential field for fail-safe uav navigation. In: 2022 30th Mediterranean conference on control and automation (MED), pp. 362–367 (2022). IEEE
33. Furrer, F., Burri, M., Achtelik, M., Siegwart, R.: RotorS—A modular gazebo mav simulator framework. *Robot operating system (ROS): The complete reference (Volume 1)*, pp. 595–625. Springer, Cham (2016)
34. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y., et al.: Ros: an open-source robot operating system. In: *ICRA workshop on open source software*, vol.3, p. 5. Kobe, Japan (2009)
35. Design and use paradigms for gazebo, an open-source multi-robot simulator. In: 2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)(IEEE Cat. No. 04CH37566), vol. 3, pp. 2149–2154 (2004). IEEE
36. Labbé, M., Michaud, F.: Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *J. Field Robot.* **36**(2), 416–446 (2019)
37. Oleynikova, H., Taylor, Z., Fehr, M., Siegwart, R., Nieto, J.: Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)* (2017)
38. Zhou, B., Zhang, Y., Chen, X., Shen, S.: Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning. *IEEE Robot. Autom. Lett.* **6**(2), 779–786 (2021)
39. Selin, M., Tiger, M., Duberg, D., Heintz, F., Jensfelt, P.: Efficient autonomous exploration planning of large-scale 3-d environments. *IEEE Robot. Autom. Lett.* **4**(2), 1699–1706 (2019). <https://doi.org/10.1109/LRA.2019.2897343>
40. Feng, C., Li, H., Gao, F., Zhou, B., Shen, S.: Predrecon: A prediction-boosted planning framework for fast and high-quality autonomous aerial reconstruction. *arXiv preprint arXiv:2302.04488* (2023)
41. Epic Games: Unreal Engine. <https://www.unrealengine.com> (2023)
42. Shah, S., Dey, D., Lovett, C., Kapoor, A.: Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In: *Field and service robotics*. [arXiv:1705.05065](https://arxiv.org/abs/1705.05065) (2017)
43. Papadimitriou, A., Jafari, H., Mansouri, S.S., Nikolakopoulos, G.: External force estimation and disturbance rejection for micro aerial vehicles. *Expert Syst. Appl.* **200**, 116883 (2022)

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

**Vignesh Kottayam Viswanathan** is currently pursuing his PhD at the Robotics and AI team at the Department of Computer Science, Electrical and SpaceEngineering, Luleå University of Technology, Sweden. He received his Master's degree in Space Science and Technology from Luleå University of Technology, Sweden with specialization in Space Technology and Instrumentation in 2021. Vignesh's research direction is focused on improving situational awareness by enabling

autonomous inspection behavior of multiple objects. His research applications span across terrestrial and space disciplines.

**Björn Lindqvist** is currently pursuing his PhD at the Robotics and AI Team at the Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Sweden, working in the field of aerial robotics. He received his Master's Degree in Space Engineering with a specialization Aerospace Engineering from Luleå University of Technology, Sweden, in 2019. Björn's research has so far been focused on collision avoidance and path planning for single and multi-agent Unmanned Aerial Vehicle systems, as well as field applications of such technologies. He has worked as part of the JPL-NASA led Team CoSTAR in the DARPA Sub-T Challenge on subterranean UAV exploration applications, specifically in the search-and-rescue context.

**Sumeet Gajanan Satpute** (Member, IEEE) received master's degree in electrical engineering with a specialization in control systems from the Veermata Jijabai Technological Institute (VJTI), India, and the Ph.D. degree from the Onboard Space Systems Group, Luleå University of Technology (LTU), Luleå, Sweden. He is currently a Postdoctoral Researcher with the Robotics and Artificial Intelligence Group, LTU. His current research interests include multiple spacecraft formation, control and path planning problems, coverage and inspection of infrastructures, and autonomous planetary explorations with multiple agents.

**Christoforos Kanellakis** received the Ph.D. degree from the Control Engineering Group, Luleå University of Technology (LTU), Sweden, and the Diploma degree from the Department of Electrical and Computer Engineering, University of Patras (UPAT), Greece, in 2015. He is currently a Postdoctoral Researcher with the Department of Computer Science, Electrical and Space Engineering, LTU. He also works in the field of robotics, focusing on the combination of control and vision to enable robots perceive and interact with the environment.

**George Nikolakopoulos** was working as a Project Manager and a Principal Investigator in several R&D&I projects funded by the EU, ESA, Swedish, and the Greek National Ministry of Research. In 2013, he has established the bigger outdoors motion capture systems in Sweden, and most probably in Europe, as part of the FROST Field Robotics Laboratory, Luleå University of Technology, Luleå, Sweden. He is currently a Professor on robotics and automation with the Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology. His work is focusing in the area of robotics and control applications, while he has a significantly large experience in creating and managing European and National Research Projects. He is the Coordinator of H2020-ICT AEROWORKS project in the field of aerial collaborative UAVs and H2020-SPIRE project DISIRE in the field of integrated process control. His published scientific work includes more than 150 published international journals and conferences in the fields of his interest. In 2003, he has received the Information Societies Technologies (IST) Prize Award for the Best Paper that promotes the scopes of the European IST (currently known as ICT) sector. In 2014, he has received the 2014 Premium Award for Best Paper in IET Control Theory and Applications, (Elsevier) for the research work in the area of UAVs. In 2014, he has been nominated as a LTU's Wallenberg candidate, one out of three nominations from the University and 16 in total engineering nominations in Sweden. His publications in the field of UAVs have received top recognition from the related scientific community, while have been several times listed in the TOP 25 most popular publications in Control Engineering Practice (Elsevier). George Nikolakopoulos is acting Chair on Robotics and AI, a Professor on Robotics and Automation at the Department of Computer Science, Electrical and Space Engineering at Luleå University of Technology. His work is focusing in the area of Robotics, Control Applications and Cyberphysical Systems. His published scientific work includes more than 150 published International Journals and Conferences in the field. He has been Associate Editor and Reviewer of Several International Journals and Conferences, as well as a member of the ARTEMIS scientific Council in the European Commission.