Osprey: Multi-Session Autonomous Aerial Mapping with LiDAR-based SLAM and Next Best View Planning

Rowan Border
Oxford Robotics Institute (ORI)
Department of Engineering Science
University Of Oxford
rborder.robots@gmail.com

Nived Chebrolu
Oxford Robotics Institute (ORI)
Department of Engineering Science
University of Oxford
nived@robots.ox.ac.uk

Yifu Tao
Oxford Robotics Institute (ORI)
Department of Engineering Science
University of Oxford
yifu@robots.ox.ac.uk

Jonathan D. Gammell
Oxford Robotics Institute (ORI)
Department of Engineering Science
University of Oxford
gammell@robots.ox.ac.uk

Maurice Fallon
Oxford Robotics Institute (ORI)
Department of Engineering Science
University of Oxford
mfallon@robots.ox.ac.uk

Abstract

Aerial mapping systems are important for many surveying applications (e.g., industrial inspection or agricultural monitoring). Aerial platforms that can fly GPS-guided preplanned missions semi-autonomously are already widely available but fully autonomous systems can significantly improve efficiency. Autonomously mapping complex 3D structures requires a system that performs online mapping and mission planning. This paper presents *Osprey*, an autonomous aerial mapping system with state-of-the-art multi-session LiDAR-based mapping capabilities. It enables a non-expert operator to specify a bounded target area that the aerial platform can then map autonomously over multiple flights. Field experiments with Osprey demonstrate that this system can achieve greater map coverage of large industrial sites than manual surveys with a pilot-flown aerial platform or a terrestrial laser scanner (TLS). Three sites, with a total ground coverage of 2528 m^2 and a maximum height of 27 m, were mapped in separate missions using 112 minutes of autonomous flight time. True colour maps were created from images captured by Osprey using pointcloud and NeRF reconstruction methods. These maps provide useful data for structural inspection tasks.

1 Introduction

Autonomous aerial mapping systems can provide crucial information for a variety of applications, including industrial inspection, construction surveying and agricultural monitoring. For some tasks (e.g., terrain

Figure 1: The Osprey autonomous aerial mapping system surveying a multi-building industrial site.

mapping) it is sufficient for an aerial platform with GPS navigation to fly along a fixed preplanned trajectory while capturing data with a camera or LiDAR. Many semi-autonomous aerial systems without online mapping or mission planning capabilities already provide this functionality (e.g., the DJI Mavic 3). Mapping structures (e.g., buildings or bridges) is more complex and requires a fully autonomous system capable of adaptively planning a flight trajectory online from live sensor measurements. This allows the aerial platform to safely manoeuvre around unknown environments and capture complete maps of complex structures.

Fully autonomous aerial mapping systems can map large structures more quickly than ground-based systems (e.g., a terrestrial laser scanner (TLS) or robot platform) and obtain greater coverage by flying above and around tall structures that would otherwise be unobservable. Their autonomy enables them to operate without oversight or communication from a human pilot, thereby reducing the cost of data acquisition as well as allowing them to map hazardous environments that are not safely accessible (e.g., nuclear waste facilities). The ultimate aim for many of these systems is for an untrained user to specify an area or object of interest that the aerial platform can then autonomously map without human intervention.

The industrial importance of fully autonomous aerial mapping is already being demonstrated by commercial systems (e.g., Emesent Autonomy, Leica BLK2FLY and Skydio 3D Scan). These are being deployed to survey mining tunnels, tall buildings and cellphone towers. Mapping missions typically exceed battery flight time and as a result some commercial systems can incrementally build a map over multiple flights.

Research on fully autonomous aerial mapping systems has advanced significantly in the past decade. Shen et al. (2012) presented a 2D LiDAR-based autonomous aerial mapping system and demonstrated it in an indoor environment. Heng et al. (2014) presented a vision-based system and demonstrated it mapping small indoor and outdoor environments. Yoder and Scherer (2016) presented a system capable of mapping large outdoor structures with a 2D LiDAR and demonstrated it surveying a bridge. Autonomous mapping systems have since been developed using a variety of different platforms, sensor payloads and mapping pipelines. They have been demonstrated indoors (e.g., Bircher et al., 2016, 2018), on small outdoor structures (e.g., Teixeira et al., 2018), large outdoor structures (e.g., Song et al., 2020, 2021) and underground (e.g., Papachristos et al., 2019). The capabilities of state-of-the-art research systems is exemplified by the aerial platforms fielded in the DARPA Subterranean Challenge (Chung et al., 2023), which autonomously mapped underground environments as part of heterogeneous robot teams.

This paper presents *Osprey*, an autonomous aerial lidar-based mapping system capable of surveying large outdoor structures over multiple flights (Fig. 1). We incorporated a custom sensor payload, called *Frontier*, onto a DJI M600 aerial platform to deploy our autonomous multi-session mapping pipeline. The state-of-the-art capabilities of Osprey are demonstrated by its ability to autonomously map several different sites, including a large (52x29x27m) disaster site in 42 minutes over consecutive two flights.

The key contributions of this work are the development of a fully autonomous aerial mapping system, which integrates existing state-of-the-art approaches for multi-sensor odometry, multi-session LiDAR-based mapping, online mission planning, and efficient motion planning. We also present the results of field experiments demonstrating its capabilities for mapping large outdoor structures. A discussion of these experiments identifies outstanding challenges for autonomous aerial mapping and future research directions. To the best of our knowledge the only other research systems with comparable capabilities are those deployed in the DARPA Subterranean Challenge, which were demonstrated in underground and indoor environments (e.g., by Team Explorer; Scherer et al., 2022 and Team CSIRO Data61; Hudson et al., 2022).

Field experiments with Osprey were conducted at the Fire Service College, a training center for emergency responders, where the system autonomously mapped three separate large industrial sites, which had a total ground coverage of 2528 m^2 and a maximum height of 27 m. The sites were also mapped using manual pilot-flown missions and a Leica BLK360 TLS to provide comparative results. Quantitative and qualitative analysis of these results demonstrates the state-of-the-art autonomous mapping capabilities of Osprey as it captured maps of all the sites with equivalent or better coverage than the pilot-flown missions and the Leica BLK360 surveys. True colour reconstructions of the sites were also created from camera images captured by the Frontier sensor payload, using either a projection pipeline to colour the pointcloud map or SiLVR (Tao et al., 2024), a Neural Radiance Field (NeRF) reconstruction approach.

This paper is structured as follows: Section 2 discusses related work on autonomous aerial mapping systems, with a focus on the different sensor configurations and mapping pipelines used. Section 3 presents the aerial platform used for Osprey, a DJI M600 with our custom sensor payload, Frontier, and the Rajant WiFi modules we used for communication. Section 4 presents the Osprey mapping system, beginning with an overview of the pipeline and followed by a description of each component. Section 5 presents the sites mapped during field experiments at the Fire Service College. Section 6 evaluates the mapping performance of Osprey on these sites in comparison to the pilot-flown missions and Leica BLK360 surveys. It also presents the true colour reconstructions. Section 7 discusses the capabilities of Osprey and identifies directions for future work. Section 8 summarises the contributions of this work and lessons learned from developing Osprey.

2 Related Work

Autonomous aerial mapping systems require five key components: (i) a sensor payload to perceive the environment, (ii) an odometry algorithm to estimate the platform motion, (iii) a mapping algorithm to aggregate sensor measurements into a combined representation, (iv) a mission planning algorithm to determine where measurements should be captured from and (v) a motion planning algorithm to identify safe paths through the environment. This section discusses related work on autonomous aerial mapping systems and their approaches for each component. The review focuses specifically on real world systems and excludes approaches that have only been demonstrated in simulation or motion capture environments.

2.1 Sensor Payload

Most aerial mapping systems use a sensor payload comprised of an Inertial Measurement Unit (IMU) and either a camera and/or LiDAR. IMU measurements can provide high frequency estimates of the payload attitude while the visual or LiDAR measurements are used to obtain a more robust translation estimate and reconstruct the environment. Measurements from the different sensors used must be time synchronised either in hardware or software before being processed. The sensors used typically depend on the environment being mapped and the payload capacity of the aerial platform.

LiDAR-Inertial payloads (Batinovic et al., 2021; Hudson et al., 2022; Ohradzansky et al., 2022) are best suited for environments with distinctive geometric structures (e.g., large buildings) and environments with poor illumination or texture where visual perception may fail (e.g., underground). However, LiDARs typically weigh over 500 grams and can not be lifted by smaller multirotor drones, which usually have a maximum

payload capacity of a few hundred grams. Larger drones can carry these payloads and easily navigate around outdoor structures but may struggle to manoeuvre in narrow indoor or underground environments.

Visual-Inertial payloads are suitable for highly textured and well illuminated environments where the platform can stay relatively close to the surfaces being mapped (e.g., indoors or in small outdoor scenes). Many of these systems use a single stereo camera (Bircher et al., 2016, 2018; Kompis et al., 2021) mounted at a downwards angle, but some combine a stereo camera with a downward-facing monocular camera (Tabib et al., 2022) or an optical flow sensor (Heng et al., 2014) that is used solely for odometry. Teixeira et al. (2018) use a single monocular camera mounted at a downwards angle. These payloads are usually smaller and lighter than LiDAR-Inertial configurations, which allows them to be mounted on small aerial platforms that can navigate narrow passages more easily. Cameras are also typically less expensive than LiDARs, but multiple cameras are required to obtain the 360° horizontal field-of-view provided by most LiDARs.

Some systems combine measurements from all three sensing modalities as LiDAR-Visual-Inertial payloads. Earlier systems (Shen et al., 2012; Yoder and Scherer, 2016) use a 2D LiDAR and monocular camera configuration while more recent systems (Papachristos et al., 2019; Nguyen et al., 2020; Scherer et al., 2022; Tranzatto et al., 2022) use a 3D LiDAR and monocular camera. Using both LiDAR and visual sensors provides redundancy when operating in environments with both poorly textured surfaces and indistinct geometric structures where solely relying on either sensor may cause the odometry to fail. The disadvantage of these payloads is an increased weight, computational cost and the complexity of synchronising measurements from multiple sensors. A few systems use single-sensor payloads, either vision-only (Song et al., 2020, 2021) or LiDAR-only (Rouček et al., 2022). These are simpler as no time synchronisation is required between sensors but are less robust to drift during rapid motions due to the lack of high frequency IMU measurements.

The Osprey mapping system presented in this paper uses a custom LiDAR-Visual-Inertial sensor payload. This provides the sensor measurements necessary to obtain a robust multi-sensor odometry estimate and produce accurate true colour 3D maps of the target sites.

2.2 Odometry

An odometry algorithm estimates the incremental platform motion using measurements from the sensor payload. Systems with LiDAR-Inertial payloads typically use odometry algorithms that register incoming LiDAR pointclouds to a rolling local map and use the IMU measurements to provide a high frequency motion estimate. Batinovic et al. (2021) and Ohradzansky et al. (2022) use Google Cartographer (Hess et al., 2016), which obtains a motion estimate by scan matching LiDAR pointclouds with a local submap. This is created by aggregating pointclouds using initial motion estimates from the IMU measurements. Hudson et al. (2022) use odometry from Wildcat (Ramezani et al., 2022), which performs surfel-based scan matching within sliding-window submaps and integrates IMU measurements in a continuous-time representation.

Systems with Visual-Inertial payloads use either Extended Kalman Filter (EKF)-based (e.g., ROVIO; Bloesch et al., 2015) or graph-based (e.g., OKVIS or VINS-Mono; Leutenegger et al., 2014; Qin et al., 2018) odometry algorithms to obtain a motion estimate from tightly coupled visual and IMU measurements.

LiDAR-Visual-Inertial systems typically combine tightly coupled LiDAR-Inertial and Visual-Inertial estimators in a lightly coupled manner (e.g., Super Odometry or CompSLAM; Zhao et al., 2021; Khattak et al., 2020). Doing so retains the accuracy and resiliency of the tightly coupled motion estimates while exploiting the computational efficiency of a lightly coupled estimate.

The vision-only system presented by Song et al. (2020, 2021) uses odometry from ORB-SLAM (Mur-Artal et al., 2015). The LiDAR-only system from Rouček et al. (2022) uses odometry from A-LOAM, an extension of LOAM (Zhang and Singh, 2014), which obtains a motion estimate by extracting features from LiDAR pointclouds and matching them between scans.

Osprey uses a multi-sensor fusion odometry algorithm to estimate the platform pose from IMU and LiDAR measurements (Wisth et al., 2023). It provided reliable motion estimates at the large sites being mapped.

2.3 Mapping

A mapping algorithm aggregates the incremental sensor measurements captured from the moving platform into a map of the environment. Most systems use a volumetric representation in the form of a 3D voxel grid. The encoded information can include measurement occupancy, observation states, occlusion states and distances to other voxels. OctoMap (Hornung et al., 2013) is a popular implementation which encodes the occupancy probability of each voxel; while Voxblox (Oleynikova et al., 2017) encodes information on the distance to occupied voxels by integrating measurements into a Truncated Signed Distance Field (TSDF) map representation. Some published approaches developed their own volumetric mapping algorithms. Shen et al. (2012) use a 2.5D occupancy grid map where each floor in an indoor environment is represented by a separate occupancy grid. Heng et al. (2014); Yoder and Scherer (2016) use custom implementations of a 3D occupancy grid and Scherer et al. (2022) present an implementation based on OpenVDB (Museth, 2013).

Volumetric representations are ideal for encoding the occupancy and exploration states of regions in an environment but the fidelity of the encoded surface information is constrained by the voxel grid resolution and using a high resolution typically incurs a high computational cost. Song et al. (2021) address this limitation by combining a volumetric representation with a surface representation that uses surfels to encode information about surface confidence. Tabib et al. (2022) use a novel Gaussian Mixture Model (GMM)-based map representation that combines elements of both volumetric and surface representations. Hudson et al. (2022) instead use a combined pointcloud and surfel representation. Rouček et al. (2022) use the pointcloud map representation provided by A-LOAM. Song et al. (2021) and Hudson et al. (2022) also incorporate pose graph updates from their SLAM algorithms (i.e., ORB-SLAM and Wildcat SLAM, respectively) into their map representations to correct odometry drift.

Osprey uses a graph-based SLAM algorithm to aggregate LiDAR pointclouds into a combined map and correct for odometry drift by detecting geometric loop closures (Ramezani et al., 2020). A LiDAR-based localisation algorithm (Kim et al., 2022) provides the ability to perform mapping over multiple flights.

2.4 Mission Planning

A mission planning algorithm determines where the platform should capture measurements from in order to obtain a complete map of an environment. These algorithms typically adopt a Next Best View (NBV) planning approach which generates a set of potential views and then selects a next view from this set to obtain the best improvement in the current map. Some systems use view planning algorithms that generate potential views with sampling-based path planning algorithms. Bircher et al. (2016, 2018); Papachristos et al. (2019) sample views with Rapidly-exploring Random Tree (RRT; LaValle, 1998) and Song et al. (2020) use RRT* (Karaman and Frazzoli, 2011). This approach generates views that are well distributed within free space but does not explicitly consider the scene structure and therefore produces many views with limited value for extending the map.

Nguyen et al. (2020) use GBPlanner (Dang et al., 2019) and Tranzatto et al. (2022) use GBPlanner2 (Kulkarni et al., 2022). These view planning algorithms sample views within a bounding box around the platform using Rapidly-exploring Random Graph (RRG; Karaman and Frazzoli, 2011) and evaluate the value of these views for extending the map. GBPlanner2 extends GBPlanner by creating an adaptively sized bounding box from the local map instead of using a fixed size and only evaluating a subset of the sampled views.

Many systems with volumetric representations use view planning algorithms that generate potential views directly from specific voxel types, which helps to ensure their utility for extending the map. These can be occupied voxels (Yoder and Scherer, 2016; Teixeira et al., 2018), surface frontier voxels (i.e., occupied voxels with unknown neighbours; Kompis et al., 2021; Scherer et al., 2022) or exploration frontier voxels (i.e., free

voxels with unknown neighbours; Heng et al., 2014; Song et al., 2021; Batinovic et al., 2021; Rouček et al., 2022; Ohradzansky et al., 2022). Systems that do not maintain a volumetric representation use other view generation methods. Tabib et al. (2022) generate views using motion primitives and Hudson et al. (2022) propose views by identifying point-based frontiers (Williams et al., 2020).

Once a set of potential views is generated a particular next best view is selected from the set based on a utility metric. Some methods simply select the view closest to the platform (Heng et al., 2014; Rouček et al., 2022) or one that lies a given inspection direction (Teixeira et al., 2018). Most systems with volumetric map representations attempt to maximise coverage of specific voxel types relative to the travel cost of reaching a view. These can be unknown voxels (Bircher et al., 2016, 2018; Papachristos et al., 2019; Tranzatto et al., 2022), surface frontier voxels (Yoder and Scherer, 2016; Kompis et al., 2021; Scherer et al., 2022) or exploration frontier voxels (Song et al., 2020, 2021; Batinovic et al., 2021; Ohradzansky et al., 2022). Some view selection metrics also consider surface quality (Kompis et al., 2021) or platform momentum (Scherer et al., 2022). Song et al. (2020, 2021) extend the selection of a next best view to a next best trajectory by optimising the path to a chosen view to maximise coverage of exploration frontier voxels (Song et al., 2020) and low confidence surfels (Song et al., 2021). Tabib et al. (2022) select a motion primitive to reduce map uncertainty and Hudson et al. (2022) select a view based on reachability metrics.

Mission planning for Osprey is performed by a NBV planning algorithm that operates directly on the LiDAR measurements (Border and Gammell, 2024). Measurement-based frontiers are identified by performing a density-based classification of each measurement. Potential views are generated to observe these frontiers and a next best view is chosen to observe the most frontiers per unit of travel distance.

2.5 Motion Planning

Motion planning is a lower level task which identifies a specific collision-free path for the platform to traverse through the environment to reach a view selected by the mission planner. Systems that use sampling-based path planners to generate potential views typically use the collision-free path to the chosen view already provided by the existing planning graph. Meanwhile, systems with voxel-based view generation methods often create their own planning graphs by connecting view-based vertices with collision-free edges and find a path through the graph using A* search (Hart et al., 1968) or Dijkstra's algorithm (Dijkstra, 1959). Yoder and Scherer (2016) use the SPARTAN algorithm (Cover et al., 2013), Teixeira et al. (2018) use MISP (Alzugaray et al., 2017) and Rouček et al. (2022) use a spatial exploration approach (Bayer and Faigl, 2019). Other systems use local planning methods including vector fields (Heng et al., 2014), direct vision-based control (Ohradzansky et al., 2022) and motion primitives (Kompis et al., 2021; Tabib et al., 2022). Scherer et al. (2022) plan paths with RRT-Connect (Kuffner and La Valle, 2000), Batinovic et al. (2021) use RRT, Shen et al. (2012); Song et al. (2021) use RRT* and Hudson et al. (2022) use a motion planner from Emesent.

Osprey uses a sampling-based planner that can quickly identify collision-free paths for the platform, often in less time than RRT or RRT*, and continuously improves its solution within the remaining planning time (Strub and Gammell, 2022).

3 Aerial Platform

The Osprey aerial platform (Fig. 2) is a DJI M600 drone with a custom sensor payload, called *Frontier*[1], that was developed at the Oxford Robotics Institute (ORI). A Rajant WiFi module provides wireless communication between the platform and an operator. The DJI M600 is a hexrotor drone with an integrated flight controller, IMU and GPS. It has a flight time of up to 30 minutes when carrying the 3 kg payload, which includes the Frontier, a Rajant module and an external battery.

[1]More details on the payload can be found here: `https://ori.ox.ac.uk/labs/drs/frontier-payload/`

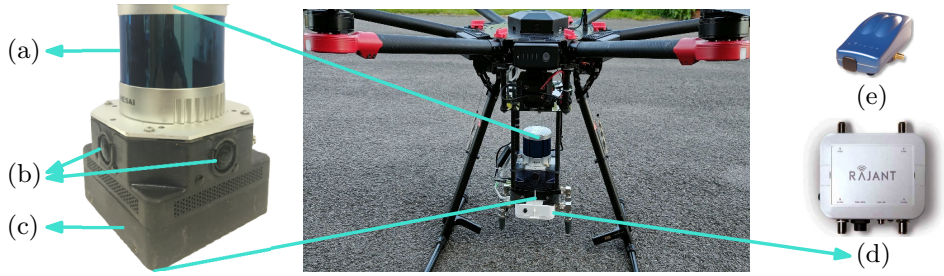


Figure 2: The Osprey aerial platform, a DJI M600 drone with a Frontier and Rajant WiFi module mounted underneath it. The Frontier consists of (a) a Hesai QT64 LiDAR, (b) a SevenSense Core Research sensor module with an IMU and three 1.6 megapixel colour fisheye cameras, and (c) an Intel NUC computer. Communication with the operator is provided by a wireless mesh network comprised of (d) a Rajant BreadCrumb ES1 module on Osprey and (e) Rajant BreadCrumb DX2 modules distributed around a target site.

Table 1: Specifications for the LiDAR, IMU and cameras used by the custom Frontier sensor payload.

	Hesai QT64 LiDAR	Sony IMX-273 Cameras	Bosch BMI085 IMU
Horizontal Field-of-View [°]	360	126	—
Vertical Field-of-View [°]	104.2	92.4	—
Horizontal Resolution [pt or px]	600	1440	—
Vertical Resolution [pt or px]	64	1080	—
Frequency (Hz)	10	20	400

The Frontier is comprised of a Hesai QT64 LiDAR (Fig. 2a), a Sevensense Core Research sensor module (Fig. 2b) with a Bosch BMI085 IMU and Sony IMX-273 colour fisheye cameras on three sides, and an Intel NUC computer with a 12-core Intel i7 CPU and 32GB of RAM (Fig. 2c). Table 1 presents the specifications for these sensors. The wide vertical field-of-view of the Hesai LiDAR allows the platform to fly above tall structures while still retaining enough visibility of surfaces below it for good odometry and mapping performance. The fisheye cameras were not used for the online mapping system but their images were recorded and used to produce colour reconstructions offline.

The camera intrinsics and camera-IMU extrinsics were calibrated with Kalibr (Furgale et al., 2013). The camera-LiDAR extrinsics were calibrated with DiffCal (Fu et al., 2023). Extrinsics between the Frontier and DJI base frames were manually calibrated. This transformation was necessary when sending commands from the Frontier to the DJI flight controller. The sensors are all connected to the NUC via Ethernet and time synchronised with PTP. All processing for the system is performed onboard the drone using the NUC.

Communication between the platform and an operator was achieved using Rajant WiFi modules, which create a reconfigurable wireless mesh network and allowed reliable communication to be maintained between the platform and operator when mapping large sites. One Rajant BreadCrumb ES1 module (Fig. 2d) was mounted on the aerial platform, another was connected to the operator laptop and intermediate connectivity was provided by Rajant BreadCrumb DX2 modules (Fig. 2e) distributed around the target site. Connectivity was required to send high level commands (e.g., to start or stop a mission) to the platform and monitor the system for safety but was not critical for the autonomous mapping, which ran entirely on the Frontier.

4 Osprey Mapping System

The Osprey mapping pipeline (Fig. 3) can be divided into five key components: (i) Sensor Payload, (ii) Odometry, (iii) Mapping, (iv) Mission Planning and (v) Motion Planning. The Sensor Payload captures measurements from the IMU, LiDAR and cameras.

In the Odometry component, the Visual Inertial Legged/Lidar Navigation System (VILENS) algorithm (Wisth et al., 2023) obtains a robust motion estimate by tightly fusing IMU and LiDAR measurements in a

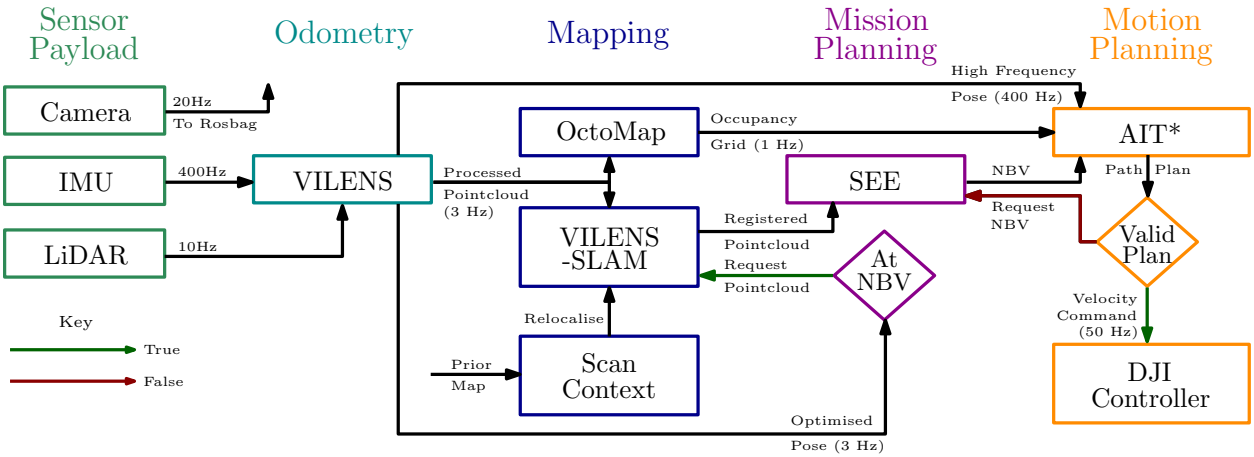


Figure 3: Flowchart showing an overview of the five key components of the Osprey mapping pipeline: Sensor Payload, Odometry, Mapping, Mission Planning and Motion Planning. The rectangular boxes represent sensor inputs or algorithms and the connections between them denote data pipelines. The diamonds represent boolean decisions, with green arrows denoting true outcomes and red arrows denoting false outcomes.

factor graph. Motion undistortion is also applied to the raw LiDAR measurements, which are then published as processed pointclouds. These are integrated into an OctoMap representation of the environment, which is used for collision checking by the motion planning algorithm.

In the Mapping component, the VILENS-SLAM algorithm (Ramezani et al., 2020) integrates undistorted LiDAR pointclouds from VILENS into a pose graph and detects geometric loop closures to correct for odometry drift. A combined map is created by aggregating every registered pointcloud in the pose graph in a common reference frame. The most recent registered cloud and pose graph are continuously published. An implementation of ScanContext (Kim et al., 2022) provides the relocalisation capabilities necessary to map large areas over multiple flights.

In the Mission Planning component, the Surface Edge Explorer (SEE; Border and Gammell, 2024), a NBV planning algorithm, directs the platform to capture measurements from views around a target site with the aim of obtaining complete coverage of visible surfaces. It processes the pointclouds captured at these chosen views and adds the measurements to its internal map representation. Boundaries between complete and incomplete surfaces are identified and potential views are generated to extend coverage of the map. A next best view is iteratively selected from this set of potential views and sent to the motion planning algorithm.

In the Motion Planning component, the Adaptively Informed Trees (AIT*; Strub and Gammell, 2022) algorithm plans a collision-free path for the platform to traverse between its current pose and the next best view chosen by SEE. It uses the OctoMap occupancy grid for collision checking. If AIT* finds a valid path then velocity commands are generated for the platform and sent to the DJI flight controller. If a valid path is not found then AIT* requests an alternative view from SEE.

The following subsections provide detailed descriptions of the algorithms used in each component of the mapping pipeline. The values used for the key parameters of each algorithm are presented in Section 5.2.

4.1 Sensor Payload

Measurements are captured from the LiDAR at 10 Hz, the IMU at 400 Hz and the cameras at 20 Hz. All of the sensors are synchronised with the onboard computer via PTP to ensure millisecond accurate time alignment between the measurements, which is critical for reliable operation of the VILENS odometry algorithm. The LiDAR and IMU measurements are sent to VILENS for processing but the camera images are only used for post-processed colour reconstructions and are saved directly to a rosbag.

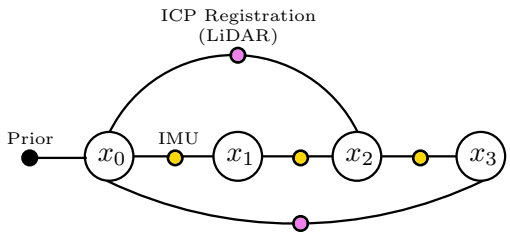


Figure 4: An illustration of the VILENS factor graph, showing the states, x_i, and the factors connecting them from the IMU (yellow) and ICP registration (pink) modules.

4.2 Odometry

The odometry algorithm incrementally estimates the platform pose as it moves around a target site. This estimate is computed from the sensor measurements and provides the initial pose for LiDAR pointclouds that are combined by the mapping algorithm. The Osprey system uses VILENS (Wisth et al., 2023), a multisensor fusion approach that obtains a robust and reliable odometry estimate by integrating tightly coupled measurements from different sensors into a factor graph with GTSAM[2] (Dellaert and Kaess, 2017). The DJI flight controller separately maintains an estimate of its own state using onboard IMU and GPS sensors but this is not used for odometry or mapping as it is less accurate than VILENS and can degrade significantly when the platform has limited GPS connectivity (e.g., when flying close to large structures).

VILENS uses a modular approach to integrate measurements from different sensors. In the large structured sites mapped by Osprey the best results were obtained using the IMU and ICP registration modules. The IMU module provides the core of the factor graph by pre-integrating IMU measurements between consecutive states at the full frequency of the IMU (Fig. 4). This provides an initial constraint on the motion between states and produces a high frequency pose estimate that can be used to generate control commands (e.g., for the DJI flight controller). The IMU pose estimates are also used to apply motion undistortion to the LiDAR pointclouds, which improves the performance of the ICP registration module.

The ICP module uses a scan-to-submap registration approach. This typically produces a more robust transformation estimate than scan-to-scan registration as a submap provides more structural detail than a single pointcloud. The submap is created by aggregating pointclouds captured from poses within 25 m of the current pose into a local map. The map density is constrained by enforcing a minimum separation distance between the included poses and downsampling the final map to 30000 points. The submap is divided into a *dense* region, within 4 m of the current pose, where the separation distance is 0.25 m and a *sparse* region where the separation distance is 2 m. This ensures that more detailed information is retained for nearby structures and improves the ICP accuracy without significantly increasing the computational cost.

Incoming pointclouds are also downsampled to 30000 points and then registered against the submap to obtain transformation estimates. This typically occurs at a lower frequency than the LiDAR output due to the computational cost of the ICP registration. ICP registration factors are defined by the relative transformation of a registered pointcloud to the current submap reference frame and connect their corresponding non-consecutive states in the factor graph (Fig. 4). The factor graph is solved after adding each ICP factor to provide an optimised pose estimate at the frequency of the registration module.

4.3 Mapping

The mapping algorithm aggregates sensor measurements captured as the platform moves through the environment into a combined map. The pose estimate provided by the odometry algorithm is used to provide an initial alignment of these measurements in a common reference frame. The Osprey system uses VILENS-SLAM (Ramezani et al., 2020), a pose-graph based mapping algorithm with loop closure capabilities.

[2] `https://gtsam.org/`

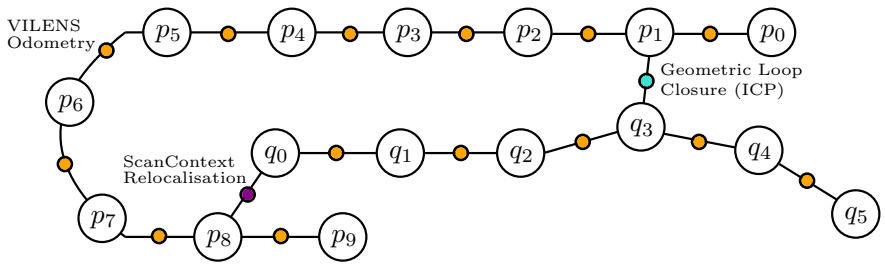


Figure 5: An illustration of a VILENS-SLAM pose graph created from a multi-session mapping mission. Nodes associated with the first session are denoted as p_i and those from a subsequent session are denoted as q_i. Odometry factors from VILENS (orange) connect consecutive nodes in the same session. Geometric loop closure factors (light blue) can exist between nodes in the same or different sessions, but only an inter-session loop closure is shown for simplicity. A relocalisation factor (purple) is added between nodes in different sessions after ScanContext identifies a successful match.

The VILENS-SLAM pose graph (Fig. 5) is implemented as a factor graph using GTSAM. Nodes in the graph represent LiDAR pointclouds captured at different poses. Edges between consecutive nodes represent odometry factors, which are provided by pose estimates from VILENS. New pointclouds are added to the pose graph at a set distance interval (e.g., every 1 m) or when requested by the mission planning algorithm. They are processed by VILENS for motion undistortion before being added to the pose graph, which improves the accuracy of the captured map.

VILENS-SLAM uses geometric loop closures to correct for odometry drift in the map. Loop closure detection is proximity-based and occurs when the pose of a newly added pointcloud is within a given radius of existing poses in the graph. When a loop closure candidate between two poses is identified an ICP registration is performed between their associated pointclouds. If this registration is successful then the loop closure is considered valid and the computed transformation between the two pointclouds is added to the graph as a loop closure factor (Fig. 5). Registration success is determined by convergence of the transformation and the number of inliers in the final result. The factor graph is then optimised to account for the new factor and the poses of the pointclouds in the pose graph are updated. The transformation between the VILENS odometry and VILENS-SLAM map frames is also updated to correct for drift in the VILENS pose estimate.

4.3.1 Multi-session Mapping

Our system can extend pose graphs across multiple sessions in order to map large sites. This requires the ability to relocalise the platform within an existing map. VILENS-SLAM uses an implementation of ScanContext (Kim et al., 2022) to perform place recognition between new LiDAR pointclouds and those integrated in an existing pose graph. ScanContext creates a unique descriptor for each pointcloud by segmenting the cloud into bins defined by circular sectors and rings, which are centered on the pointcloud origin. The descriptor is defined by the greatest z-axis value of measurements within each bin. This descriptor is robust to rotational variations around the z-axis, which allows the system to relocalise when the yaw orientation of the aerial platform differs between visits to the same location.

The extension of an existing map when starting a new flight as part of an ongoing mission proceeds as follows. The existing SLAM map and pose graph are loaded from files and ScanContext descriptors are computed for all the individual pointclouds. Descriptors generated for online pointclouds from the LiDAR are matched with those from the pointclouds in the pose graph. When a valid match is found an ICP registration is performed between the matched pointclouds to verify the match and obtain a transformation estimate. If the ICP registration converges to a good solution then the relocalisation is successful. The online pointcloud is added to the pose graph and connected to the node associated with its matched pointcloud from the existing map by a relocalisation factor (Fig. 5), which uses the transformation obtained from the ICP registration. The integration of new pointclouds into the map then proceeds as previously described.

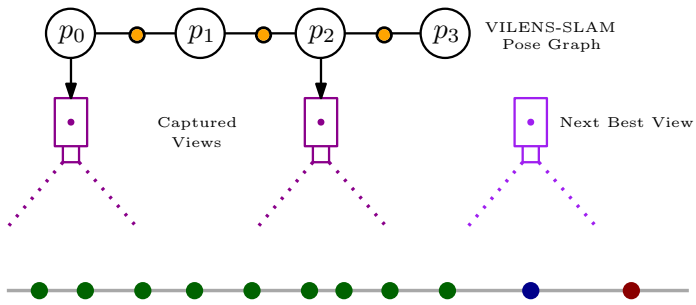


Figure 6: An illustration of how SEE, the mission planning algorithm, integrates with the pose graph created by VILENS-SLAM, the mapping algorithm. Views captured by SEE are denoted by sensor icons (dark purple) and correspond to reference clouds in the pose graph. Measurements processed by SEE lie on a surface (grey line) and are classified as core (green dots), frontier (blue dot) or outlier (red dot) points. The next best view chosen to capture the surface around the frontier is shown as a sensor icon (light purple).

4.4 Mission Planning

The mission planning algorithm determines where measurements should be captured from in order to improve map coverage. Osprey uses SEE (Border and Gammell, 2024), a measurement-direct NBV approach that plans views to obtain a minimum measurement density from all surfaces within a bounded volume of space. The platform is placed facing a target structure and the bounding volume is then defined by the user relative to this takeoff location as an axis-aligned bounding box that is large enough to encompass the structure.

SEE starts planning potential views after takeoff by capturing and processing a registered LiDAR pointcloud from VILENS-SLAM. The pointcloud measurements are integrated into the unstructured pointcloud representation used by SEE and individually classified based on the local density of neighbouring points within a resolution radius. Measurements with a minimum density are classified as *core* points and represent complete surfaces. Measurements with insufficient density are *outliers* and denote incomplete surfaces. The boundary between complete and incomplete surfaces is defined by *frontier* points, which are outlier points with core point neighbours (Fig. 6).

Potential views are generated from the frontier points to extend the map coverage by capturing new measurements at the boundary between complete and incomplete surfaces. Each view is generated to observe the surface around its associated frontier point at a specified distance and with an orientation orthogonal to the local surface in order to maximise coverage. If a frontier point is occluded from its associated view then an optimisation approach is used to find an alternative unoccluded view.

SEE iteratively chooses a next best view from the set of potential views to extend map coverage. This next best view is chosen to provide the greatest improvement in surface coverage per unit of travel distance. The chosen view is sent to the motion planning pipeline, which identifies a collision-free path and commands the platform to reach the view. When the next best view is reached the platform stops and SEE captures a new registered LiDAR pointcloud from VILENS-SLAM for processing. SEE continues to plan new potential views and select next best views until all measurements are classified as either core or outlier points.

The version of SEE used for Osprey extends our previous work (Border and Gammell, 2024) by making the point classification more robust to measurement noise, adding the ability to integrate loop closures and enabling multi-session mission planning. The misclassification of points on complex surfaces (e.g., thin or pointed structures) is reduced by evaluating the distribution of the neighbouring points around a measurement in addition to their density. SEE integrates loop closures from VILENS-SLAM into its map representation by receiving pose graph updates when a loop closure is added and computing the transformation between the previous and current poses of each pointcloud in its internal map. The corresponding measurements in the pointclouds are then transformed and reprocessed by SEE as if they were newly added.

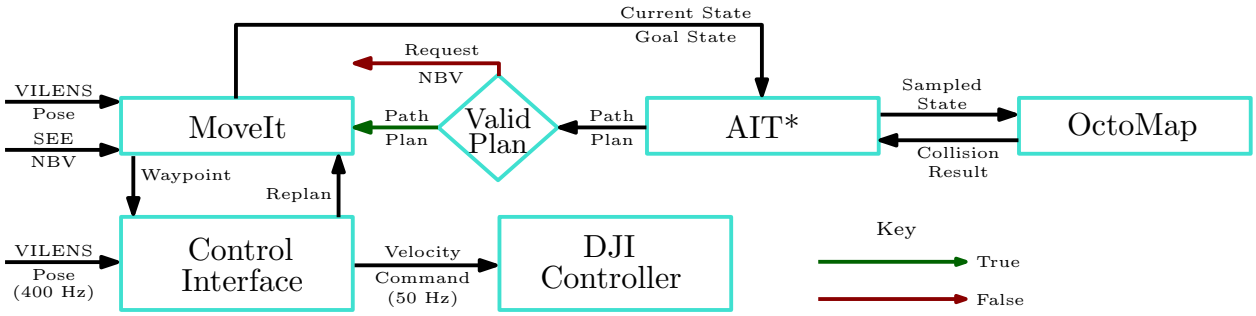


Figure 7: A flowchart overview of the motion planning pipeline. The constituent algorithms are represented by rectangular boxes and the connections between them represent data communications. Diamond boxes denote decisions, with green arrows indicating true outcomes and red arrows indicating false outcomes.

Multi-session mission planning capabilities were added to SEE by enabling it to export its current observation state at the end of a flight and import a previous state when starting a new flight as part of an ongoing mission. When beginning a new flight that will extend an existing map, SEE is started with the same parameters as the last flight and imports the previous observation state. It waits for ScanContext to relocalise against the existing map and then selects a next best view from the imported observation state.

4.5 Motion Planning

The motion planning algorithm identifies a collision-free path for the platform to follow from its current pose to the selected next best view. Osprey uses AIT* (Strub and Gammell, 2022), an almost-surely asymptotically optimal sampling-based planner, to efficiently plan trajectories between views. AIT* uses informed sampling to obtain an approximation of the state space solely within a constrained region where an improvement to the current path could be found. It is able to efficiently evaluate new paths through the informed samples by simultaneously calculating and exploiting a problem-specific cost heuristic that aims to minimise the path length. It finds an initial path first and then continuously refines its informed state approximation and cost heuristic to identify more efficient paths within the remaining allocated planning time. This enables AIT* to find efficient paths faster than other sampling-based planners (e.g., RRT, RRT*).

Osprey uses the implementation of AIT* provided with the Open Motion Planning Library (OMPL[3]; Sucan et al., 2012) and uses MoveIt[4] (Coleman et al., 2014) to interface between AIT* and the DJI M600 flight controller. Collision checking is performed using an OctoMap occupancy grid, which has a 0.5 m voxel resolution and is updated with registered pointclouds from VILENS-SLAM at 1 Hz.

The planning pipeline works as follows. MoveIt receives a next best view from SEE and the current platform pose from VILENS. These 6 DoF poses are transformed into 4 DoF states consisting of a position and yaw angle. AIT* then attempts to plan a path between the start state (i.e., the current platform state) and the goal state (i.e., the next best view state) within a maximum allowed planning time of 10 s.

If a valid path is not found (e.g., because the goal state is too close to an obstacle) then an alternative view is requested from SEE. If a valid path is found then the first state along the trajectory is sent to the control interface, which generates velocity commands for the DJI flight controller. The platform stops when each state is reached (i.e., the VILENS pose is within 1 m of the target state) and a new path to the goal is planned from the current state. This helps prevent collisions with previously unobserved obstacles. The platform continues moving between states and replanning its trajectory until the goal state is reached.

The control interface generates velocity commands for the DJI flight controller at 50 Hz based on the difference between the current platform state, which is updated from the high frequency VILENS pose at

[3] `https://ompl.kavrakilab.org/`
[4] `https://moveit.ros.org/`

Figure 8: Photographs of the *Industrial Building A* site. It is a two storey building, 30x16x12m (LxWxH) in size, with a balcony surrounding the upper storey, which is partially enclosed on one side. There is a concrete fire escape on one corner of the building, a garbage chute (red) attached to one side, a concrete chimney and various metal structures on the roof.

Figure 9: Photographs of the *Industrial Building B* site. It is a five storey building, 27x20x21m (LxWxH) in size, with a balcony surrounding the fourth storey and two concrete fire escapes attached to different sides of the building. There is also an exposed metal fire escape with a garbage chute attached to the building, which is comprised of thin metal structures and proved challenging to map.

Figure 10: Photographs of the *Rig 5* site. It is a large site, 52x29x27m (LxWxH) in size, which contains multiple structures and is staged as a disaster scenario. The site is comprised of a seven storey building next to a long three storey building with an exposed interior on one side and a metal fire escape at one end. There is a large rubble pile next to these buildings with several upright metal girders within it.

400 Hz, and the next target state. The platform is commanded to accelerate at constant rate, a, until the maximum allowed velocity, v, is reached or it nears the target state and begins decelerating to a stop.

5 Field Experiments

Field experiments with Osprey were conducted at the Fire Service College, Moreton-in-the-Marsh, Gloucestershire, UK. The facility is a training center for firefighters with several standalone buildings ideal for testing the Osprey autonomous mapping system. Field experiments were conducted at three of these sites ranging from a medium-sized two storey industrial building (*Industrial Building A*; Fig. 8) to a large five storey industrial building with several external fire escapes (*Industrial Building B*; Fig. 9) and an even larger site containing multiple structures that was staged as a disaster scenario (*Rig 5*; Fig. 10).

5.1 Experiments

Three types of experiment were performed at each site: a survey-grade scan using a TLS, a pilot-flown mission and an Osprey autonomous mapping mission. A Leica BLK360 was used to obtain a survey-grade map of each site by registering individual scans around the site. The pilot-flown missions were performed by an author who is an experienced pilot and flew the platform around each site while the SLAM algorithms

ran passively onboard. The pilot empirically judged when the map was complete. These experiments were performed to provide comparative results for evaluating the performance of the Osprey system.

The Osprey missions were conducted with an operator overseeing progress on a laptop and a safety pilot monitoring the aerial platform, ready to take control if necessary. These autonomy missions proceeded as follows. The aerial platform was placed at one end of a site and a bounding box, which defined the target area for mission planning, was specified by the operator. The bounding box was set large enough to encompass the entire site, with a minimum height of 2 m and a maximum height greater than the tallest structure.

An autonomous mission was initiated by the operator using the automatic takeoff capability of the DJI M600. After takeoff, the platform would climb to a height of 3 m and the autonomous mapping system would start. The platform would then be directed by the Osprey system to fly autonomously around the site. The safety pilot monitored the platform continuously and would intervene if there appeared to be a risk of collision. Interventions only occurred on a few occasions when the drone deviated from its planned trajectory due to wind gusts or poor performance of the flight controller (e.g., due to degraded GPS accuracy). In these cases the safety pilot would take control, fly the platform to a safe minimum distance of at least 3 m from any structure and wait until the gust passed or the flight controller performance improved (e.g., due to better GPS accuracy). Control was then returned to the Osprey system and the autonomous mission was resumed. When the platform battery level was insufficient to continue flying the operator would stop the autonomous mission planning and manually land the platform.

After landing, the platform batteries would be replaced and a new flight could be initiated to extend the current map as part of the same autonomous mission. At the start of this next flight the Osprey system would load the current map from file and relocalise within it using ScanContext before activating the autonomous mission planning. This relocalisation would typically occur during takeoff, but in some cases the safely pilot needed to manually fly the platform around the takeoff area until the relocalisation was successful. The mission planning would then be started. When the mission planning algorithm determined that the target area had been completely mapped it would stop and return control to the safety pilot for a manual landing.

5.2 Parameters

The algorithms used the same parameters in all of the Osprey experiments, with only the bounding box being site-specific. The key parameters for each algorithm are presented in Table 2. VILENS, the odometry algorithm, ran its ICP module on every third pointcloud from the LiDAR (i.e., at a frequency of 3 Hz). The scan-to-submap ICP registration was successful if the difference in translation between iterations converged to less than 0.01 m and the difference in rotation converged to less than 0.001 rad within 30 iterations.

VILENS-SLAM, the mapping algorithm, integrated new pointclouds from the LiDAR into its pose graph every 1 m, which provided sufficient map density while maintaining a reasonable computational cost. The detection radius for geometric loop closures between nodes in the pose graph was 3 m, which was large enough to correct for odometry drift while mitigating the risk of false loop closure matches. A geometric loop closure was considered successful if the ICP registration met the same convergence criteria used by VILENS and there were at least 5000 inliers in the final ICP result.

SEE, the mission planning algorithm, used a view distance of 10 m for potential views in order to obtain high coverage of the target site from each captured view. The resolution radius of 1.5 m and target density of 5 points per m^3 were chosen to ensure that the boundaries between complete and incomplete surfaces could be reliably identified on the large structures being observed while mitigating the effects of measurement noise.

The motion planning pipeline, which is built around AIT*, used a maximum distance of 5 m between waypoints to ensure the platform did not attempt to traverse unobserved regions without replanning its path. The maximum velocity of 1 m per s and acceleration of 0.1 m per s^2 were chosen based on limitations of the DJI flight controller and to ensure that the safety pilot could take control of the platform if required.

Table 2: The key parameters used by each of the constituent algorithms in the Osprey mapping system.

Odometry (VILENS)		Mapping (VILENS-SLAM)		Mission Planning (SEE)		Motion Planning (AIT*)	
ICP Module Freq. [Hz]	3	Reference Cloud Dist. [m]	1	View Dist. [m]	10	Max Waypoint Dist. [m]	5
ICP Max Trans. Diff. [m]	0.01	Loop Closure Radius [m]	3	Resolution Radius [m]	1.5	Max Velocity [$\mathrm{m\,s^{-1}}$]	1
ICP Max Rot. Diff. [rad]	0.001	Loop Closure Min Inliers	5000	Target Density [points $\mathrm{m^{-3}}$]	5	Acceleration [$\mathrm{m\,s^{-2}}$]	0.1

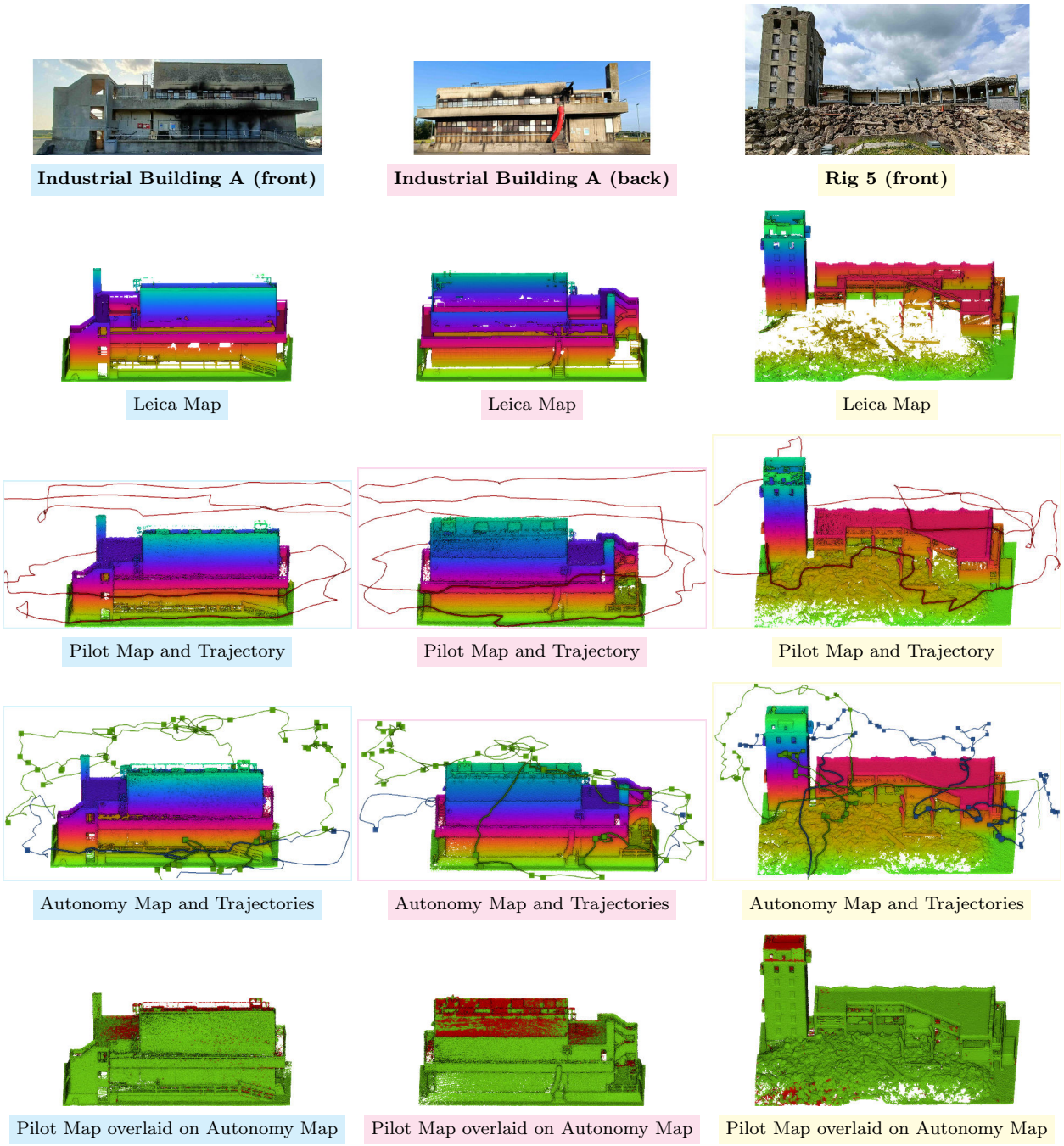


Figure 11: Qualitative results showing the pointcloud maps obtained by field experiments conducted at Industrial Building A and Rig 5 (Autonomy 1 mission). The last row shows a comparison between the Autonomy map (green) and the Pilot map (red and green).

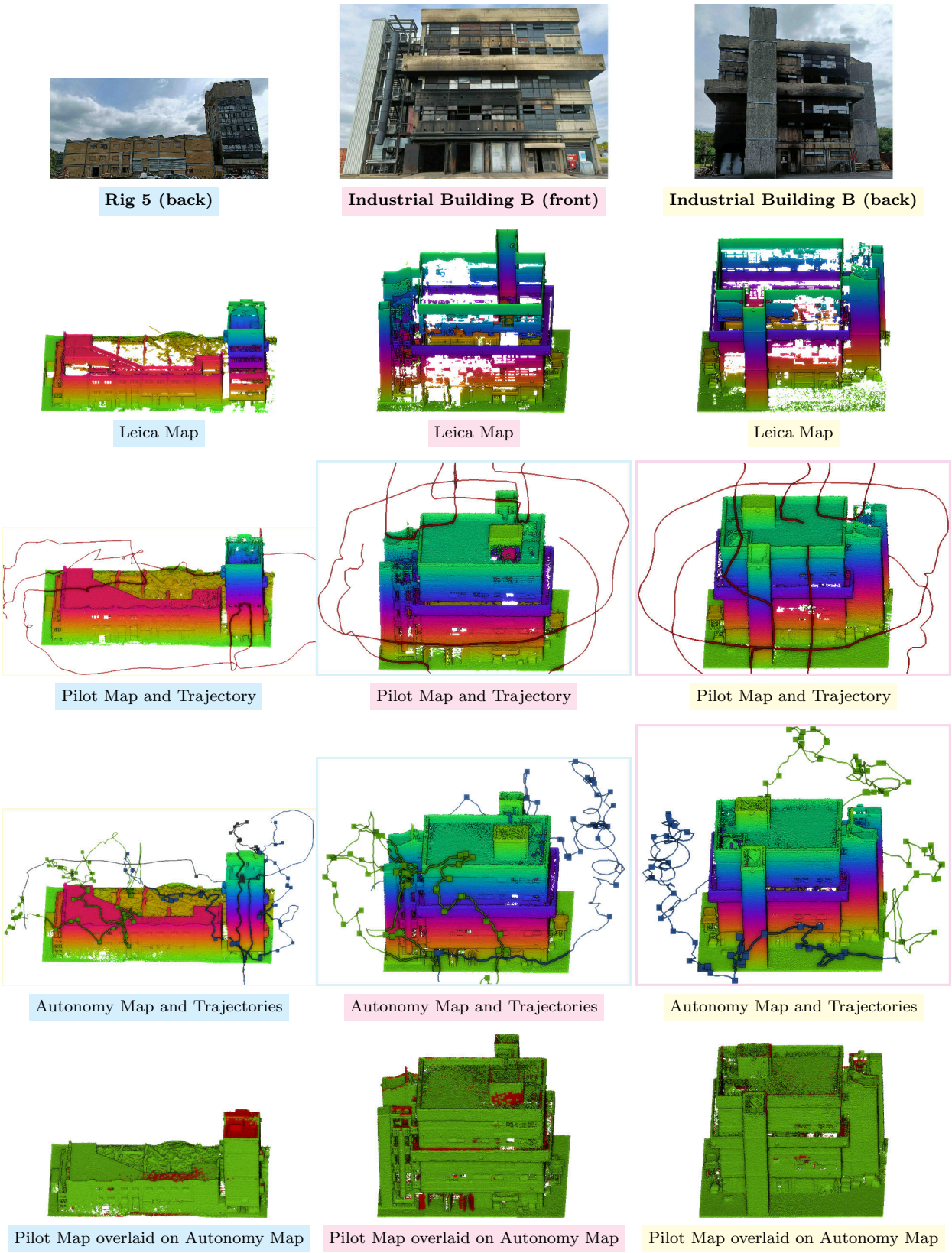


Figure 12: Qualitative results showing the pointcloud maps obtained by field experiments conducted at Rig 5 (Autonomy 2 mission) and Industrial Building B. The last row shows a comparison between the Autonomy map (green) and the Pilot map (red and green).

Table 3: Quantitative metrics for the field experiments, measuring the efficiency of each mapping approach and the quality of maps obtained at each site. Two autonomy missions were carried out at Rig 5.

	Industrial Building A			Industrial Building B			Rig 5			
	Leica	Pilot	Autonomy	Leica	Pilot	Autonomy	Leica	Pilot	Autonomy 1	Autonomy 2
Scans / Flights	19	**1**	2	9	**1**	2	23	**1**	2	3
Travel Distance [m]	—	**400**	512	—	**476**	541	—	**445**	651	824
Mission Time [s]	7980	**493**	1956	3780	**535**	2271	9660	**538**	2498	3323
Flight Speed [$\mathrm{m\,s^{-1}}$]	—	**0.81**	0.26	—	**0.89**	0.24	—	**0.83**	0.26	0.25
View Planning Time [s]	—	—	10	—	—	24	—	—	39	37
Path Planning Time [s]	—	—	175	—	—	144	—	—	89	308
Coverage [%]	83.4	87.6	**93.8**	75.5	89.7	**95.7**	75.4	85.0	91.1	**92.9**
Accuracy [m]	—	**0.04**	0.05	—	0.07	**0.06**	—	0.08	**0.06**	0.09

6 Evaluation

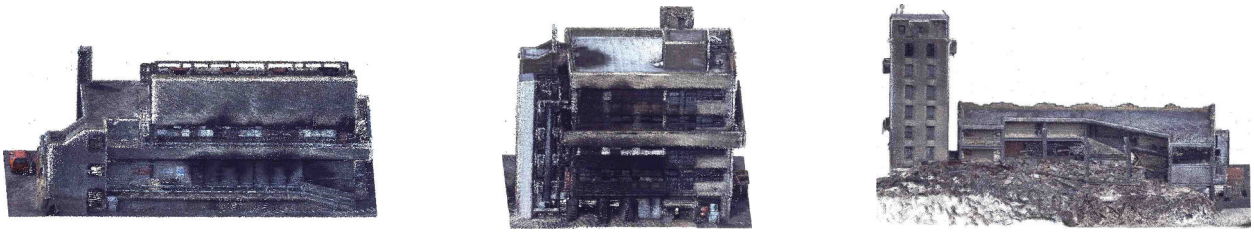
The mapping performance of Osprey is evaluated qualitatively and quantitatively based on the presented field experiments. We compare the quality and efficiency of its mapping with results obtained from the pilot-flown missions and Leica BLK360 TLS.

Figures 11 and 12 present the pointcloud maps obtained by the mapping approaches for each of the three sites. Maps from each approach were aligned into a common reference frame using CloudCompare[5]. Each column presents the maps for one site, viewed from either the front or back. The first row presents a photograph of the site from the corresponding perspective. Rows 2–4 present the maps obtained by the Leica BLK360, pilot-flown mission and autonomous Osprey mission, respectively. These maps are coloured with a z-axis colourmap and also show the flight trajectories of the platform for the aerial missions. The squares along the autonomy trajectories denote the views captured by SEE. The last row shows a coverage comparison between the aligned autonomy and pilot maps. This is computed by matching points in the autonomy map with their nearest neighbours in the pilot map. Green points have a match closer than 10 cm and indicate good consistency between the maps. Red points have no match within 10 cm and denote areas that were missed in the piloted mission.

Table 3 presents quantitative metrics to evaluate the efficiency of each mapping approach and the quality of maps obtained. The travel distance measures the total distance flown by the aerial platform during a mapping mission, from takeoff to landing for each flight. The mission time is the total scanning time taken for each Leica BLK360 scan or the overall flight time for the aerial missions, from takeoff to landing for each flight. The flight speed is the average speed of the platform during the aerial missions. The view and path planning times are specific to the Osprey autonomy missions. They measure the total computation time taken by the mission and motion planning algorithms. The computation time of the odometry and mapping algorithms was not measured separately as they ran continuously in real-time for the duration of a mission.

The quality metrics quantify the coverage and accuracy of the captured maps. Map coverage is measured by comparing the individual maps obtained by each approach with a combined map created by aggregating the pointclouds from all of the maps in a common reference frame. Points in the combined map are matched with their nearest neighbours in each individual map; those with a match closer than 10 cm are considered covered and those without are uncovered. The coverage metric is computed as the percentage of points in the combined map that are covered by an individual map. The mapping accuracy of the pilot-flown and Osprey autonomy missions is evaluated relative to the Leica BLK360 maps as it provides survey-grade mapping accuracy. The accuracy metric measures the mean distance between points in the Leica BLK360 map of a site and their nearest neighbour matches in the pilot-flown or autonomy map.

[5] `https://www.cloudcompare.org/`



Industrial Building A | Industrial Building B | Rig 5

Figure 13: Pointcloud maps of each site, captured by the Osprey autonomy missions, that were colourised using the camera images. This colourisation was performed offline using a custom pipeline based on COLMAP.

Figure 14: Two novel views of Rig 5 rendered by our *SiLVR* NeRF-mapping system in post-processing.

6.1 Pointcloud and NeRF Colour Reconstructions

The LiDAR on the Frontier device overlaps with three fisheye colour cameras, which we used for offline true colour reconstruction. We considered two different methods: projecting colour from the camera images onto the pointcloud map (Fig. 13) and producing a NeRF reconstruction using SiLVR[6] (Fig. 14; Tao et al., 2024).

The projected colour reconstructions were created with a custom pipeline, which uses COLMAP[7] (Schonberger and Frahm, 2016) to optimise the SLAM-estimated camera poses and intrinsic parameters. The camera images are then projected onto the pointcloud map from their optimised poses to produce a true colour map that preserves the metric accuracy of the original map. When colour is projected onto a point from multiple images the result is averaged and points without projected colour are removed. This does not produce photorealistic results due to variable image exposure (e.g., along the building edges in Fig. 13).

SiLVR creates NeRF reconstructions by using the raw LiDAR pointclouds, camera images and pose graph trajectory from VILENS-SLAM. The NeRF technique for differentiable volume rendering was first introduced by Mildenhall et al. (2020). Our approach builds upon Nerfacto (Tancik et al., 2023) by adding LiDAR measurements to regularise the depth and surface normals in the NeRF using LiDAR range images. In particular this improves the planarity of uniformly textured surfaces. The poses from VILENS-SLAM are used to initialise COLMAP, which refines them in a bundle adjustment. Large sites can be reconstructed by segmentation into submaps using Spectral Clustering (Von Luxburg, 2007). SiLVR produces photorealistic reconstructions with high fidelity and can achieve promising metric accuracy due to the LiDAR regularisation.

7 Discussion

The results (Figs. 11 and 12, Table 3) show that the Osprey autonomous mapping system obtained the greatest map coverage for all the sites. This clearly demonstrates the value of mapping sites with an aerial

[6] `https://ori-drs.github.io/projects/silvr/`
[7] `https://colmap.github.io/`

platform and using an autonomous system with online mission planning capabilities. The Leica BLK360 is clearly unable to map areas that could not be seen from the ground such as the roofs of the buildings. We did not have access to a Leica BLK2FLY drone, which may address these limitations.

Our key demonstration is that the Osprey autonomy missions obtained greater coverage than the pilot-flown missions. This difference is most significant on the roofs and balconies of the buildings (Figs. 11 and 12, last row). The pilot had to fly the platform high over the building roofs in order to maintain the direct line-of-sight visibility needed for safe operation and had no knowledge of structures on the roofs that were not visible from the ground. This made it challenging to determine if they had been completely mapped. The balcony interiors were difficult to map, particularly for the pilot, as from most angles they were occluded by the balcony walls and it was challenging for the pilot to determine from their perspective on the ground if the interiors were visible to the LiDAR. The results show that many of these limitations for the pilot-flown missions can be overcome by using a state-of-the-art autonomous aerial mapping system such as Osprey.

In terms of mapping efficiency, Osprey took less time than the Leica scanner surveys but took longer and flew further than the pilot-flown missions (Table 3). The Leica scanner captured measurements at several locations around a site with each scan taking 7 minutes. The Leica surveys took an average of three times longer than the Osprey autonomy missions. The difference in mission times between the Osprey autonomy and pilot-flown missions can be attributed to a number of factors. The Osprey autonomy system held position while performing view and path planning but this computation time was relatively insignificant — only 1% of the overall mission time for view planning and 8% for path planning. A more significant factor is the difference in average flight speed, which was three times slower than for the pilot-flown missions. Osprey was constrained to fly slower than the piloted missions to mitigate the risk of the platform deviating from its planned trajectory when flying near large structures, where limited GPS connectivity could degrade the accuracy of the DJI flight controller state. This slower flight speed provided the safety pilot with enough time to safely prevent collisions.

The difference in travel distance and trajectory smoothness between the Osprey autonomy and pilot-flown missions can be attributed to the use of *a priori* site knowledge for the piloted missions. The pilot preplanned their flights by walking around a site before takeoff and empirically judging what trajectory would provide the best coverage. This preplanning allowed the pilot to fly smoother and shorter trajectories that could be completed in a single flight. Incrementally planning potential views online with SEE enabled Osprey to obtain greater map coverage than the pilot-flown missions, while achieving a similar map accuracy, but produced longer and less smooth trajectories as the mission planning algorithm made locally optimal decisions based on incomplete site knowledge. This could potentially be improved by planning a sequence of next best views instead of a single view and applying a smoothness constraint on the trajectory between them.

The travelling required to land and takeoff between flights also increased the overall distance of the multi-session Osprey missions. We include the distance required to manually fly the platform to a safe landing location after reaching a critical battery level and the distance autonomously flown by the platform to resume its mission after replacing the batteries. This distance was sometimes significant, particularly for larger Rig 5 site, as when the drone was flying high above the opposite side of the building from the safety pilot they would have to fly the platform over the building while maintaining sufficient height for good visibility and identify a nearby location clear of obstacles for landing. It accounted for 23% of the overall travel distance in the second autonomy mission at Rig 5, which required two battery changes. This distance could be reduced in future work by autonomously identifying a safe landing location (e.g., Bartolomei et al., 2022).

The overall performance of Osprey demonstrates the successful integration of its constituent algorithms. The VILENS odometry algorithm provided reliable pose estimates that facilitated the accurate registration of LiDAR measurements into a graph-based SLAM map by VILENS-SLAM, which uses loop closures to correct for odometry drift. The SEE mission planning algorithm directed the platform to improve map coverage by capturing measurements from chosen views and collision-free paths between these views were planned by the AIT* motion planning algorithm. LiDAR-based place recognition with ScanContext made the multi-session mapping possible by relocalising the platform within the existing SLAM map for an ongoing mission. This

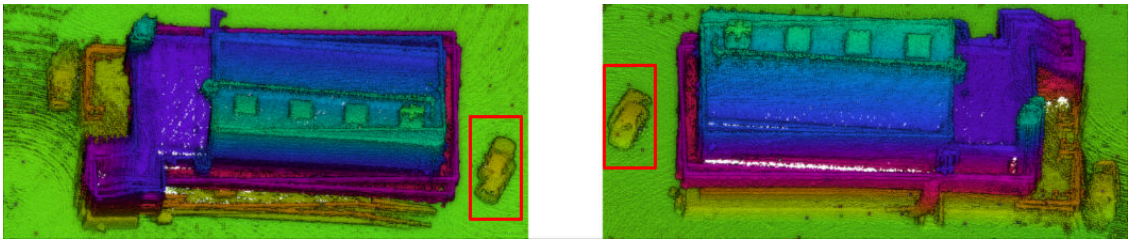


Figure 15: Two views of a pointcloud map with SLAM loop closure correction *disabled* at Industrial Building A. It illustrates that pointcloud inconsistency occurs when there is uncorrected odometry drift (e.g., the misaligned car shown in the red boxes) and highlights the importance of highly accurate online SLAM for autonomous mapping.

was the only combination of approaches tried when developing Osprey as they were well suited to the task of mapping large outdoor structures. All of the algorithms were able to run concurrently on the Frontier device without bottlenecks. The average computational load for VILENS was $\sim 28\%$ of the CPU and for VILEN-SLAM this was $\sim 1.8\%$. SEE typically used $\sim 25\%$ of the CPU when processing new measurements and AIT* used $\sim 3.1\%$ when planning paths. The efficiency of the system could be further improved by sharing a single map representation between the algorithms. The performance of the individual algorithms is discussed in detail in the following subsections.

7.1 Sensor Payload

The perception capabilities of the aerial platform were well suited for mapping the target sites. The wide vertical field-of-view LiDAR made it possible to fly above tall structures and acquire more complete maps. The main limitation of this LiDAR is its maximum range of 20 m, which made it necessary to fly relatively close to structures for robust odometry and could make it challenging to map sites with large gaps between structures. Newer LIDAR models are likely to relax this trade-off in the future.

An alternative design could have been to use multiple LiDARs mounted at different angles, a hemispherical LiDAR or a spinning LiDAR configuration (e.g., Hudson et al., 2022) to achieve more complete coverage. Cameras with a greater dynamic range would provide better image exposure for visual odometry and point-cloud colouring — in particular to avoid overexposure of surface edges facing the sky (Fig. 13).

7.2 Odometry

The VILENS odometry algorithm provided a robust incremental motion estimate during the field experiments by tightly integrating high-frequency IMU measurements and lower frequency LiDAR measurements in a factor graph. The wide vertical field-of-view of the LiDAR ensured that there was always sufficient coverage of unique geometric structures for the ICP registration module to reliably converge to a good solution. All of the mapped sites were standalone structures, even though some contained multiple connected buildings, and therefore there was always enough 3D structure within range of the LiDAR to provide robust odometry. This would not necessarily be the case when mapping larger sites that contain disconnected structures. In this scenario it may be beneficial to include visual feature tracking (Wisth et al., 2023) or GPS measurements (Beuchert et al., 2023) in the VILENS odometry estimate, both of which have been successfully demonstrated on other platforms in different environments.

7.3 Mapping

The VILENS-SLAM algorithm produced high quality pointcloud maps of the sites by incorporating geometric loop closures to correct for drift in the VILENS odometry estimate. The robustness of these loop closures was aided by low drift in the VILENS odometry, which in turn required the place recognition to find less frequent and more confident loop closure proposals. The importance of loop closure corrections is demonstrated in

Fig. 15, where the loop closure mechanism was disabled and this produced misalignments in the pointcloud map (e.g., the misaligned car shown in the red boxes).

For multi-session mapping (Section 4.3.1), the ScanContext algorithm was usually reliable at relocalising the platform within a prior map during takeoff. Occasionally it was necessary for the safety pilot to fly the platform around the area of the existing map until a place recognition match could be identified. This occurred during test flights at Industrial Buildings A and B, where the sides of the building were sometimes not distinctive enough for a valid match and it was necessary to fly to a different location to find one. Relocalisation robustness could also be improved with visual place recognition (e.g., Maffra et al., 2018).

7.4 Mission Planning

The SEE mission planner reliably directed the platform to capture highly complete maps of the target sites. It was able to capture measurements from regions not covered by the pilot-flown missions or Leica surveys (e.g., the building roofs). It generally obtained better coverage of the balconies and other surfaces with restricted visibility but sometimes failed to capture measurements from areas that were considered unobservable after too many unsuccessful attempts. Some of these areas could possibly have been successfully observed at a closer range. Therefore, adding the ability to adapt the view distance after a chosen view fails to capture new measurements around a target frontier point may improve the coverage attainable by SEE.

SEE plans potential views to incrementally expand the coverage of measurements on visible surfaces. This works well when the surfaces in a target area are interconnected and coverage can be extended in all directions until a complete map is obtained. However, larger sites often contain multiple structures that are only connected by the ground. SEE could observe these sites by including the ground in its target area but this may not be the most efficient method for extending coverage between structures. Future work could investigate adding a volumetric exploration strategy to SEE for planning missions at sites with unconnected structures, which may be more efficient than planning solely based on measurement coverage. Adding an adaptive view distance would also be beneficial for observing surfaces in narrow gaps between structures.

7.5 Motion Planning and Control

The AIT* path planning algorithm could reliably plan efficient trajectories for the platform to traverse between chosen views. However, the executed trajectory sometimes deviated from the planned trajectory by as much as 1 m due to poor performance of the DJI flight controller, which separately estimates its own state using onboard IMU and GPS sensors. The accuracy of this flight controller state is dependent on the number of visible GPS satellites and can degrade significantly when the platform flies close to large structures.

Our control interface generated velocity commands for the DJI flight controller using the VILENS odometry, as this was more reliable, but the DJI flight controller sometimes produced different velocities than were commanded when its own internal state was inaccurate (e.g., due to low GPS accuracy). This reduced the safe flight speed of the drone and we needed to set maximum velocity and acceleration limits that were lower than desired. Switching to a flight controller that accepts input from an external state estimator (i.e., VILENS), such as PX4[8], would enable closed-loop control based on a single reliable state estimate.

8 Conclusion

This paper presents Osprey, an autonomous aerial mapping system with state-of-the-art multi-session mapping capabilities. Osprey integrates existing state-of-the-art approaches for multi-sensor odometry, LiDAR-based SLAM, mission planning, motion planning, and multi-session relocalisation into a complete pipeline capable of autonomously mapping large structures over multiple flights. The key contributions of this work are the development of Osprey, a demonstration of its mapping capabilities on several large outdoor sites

[8] `https://px4.io/`

and a discussion of outstanding challenges that should be addressed when developing future systems. A dataset[9] of the field experiments has been made publicly available to benefit future research.

The field experiments conducted with Osprey demonstrate that it is able to autonomously map large industrial sites and achieves greater coverage than pilot-flown missions and surveys with a TLS. It is quicker than carrying out a TLS survey and has equivalent map accuracy to the pilot-flown missions. Future improvements to the system could also allow Osprey to autonomously map sites at similar flight speeds to a human pilot and with comparable travel distances. Another promising direction for future work, beyond improving the individual Osprey system, would be to extend these capabilities to multiple platforms, which would enable the collaborative mapping of even larger and more complex sites.

Acknowledgments

The authors would like to thank Wayne Tubby, Matt Towlson and Arundathi Shanthini at ORI for building the Frontier sensor payload and mounting infrastructure for the DJI M600. This research was funded by Horizon Europe through the Digiforest project [101070405] and by UK Research and Innovation and EPSRC through ACE-OPS: From Autonomy to Cognitive assistance in Emergency OPerationS [EP/S030832/1].

References

Alzugaray, I., Teixeira, L., and Chli, M. (2017). Short-term UAV path-planning with monocular-inertial SLAM in the loop. *IEEE International Conference on Robotics and Automation*, pages 2739–2746.

Bartolomei, L., Kompis, Y., Teixeira, L., and Chli, M. (2022). Autonomous Emergency Landing for Multicopters using Deep Reinforcement Learning. *IEEE International Conference on Intelligent Robots and Systems*, pages 3392–3399.

Batinovic, A., Petrovic, T., Ivanovic, A., Petric, F., and Bogdan, S. (2021). A Multi-Resolution Frontier-Based Planner for Autonomous 3D Exploration. *IEEE Robotics and Automation Letters*, 6(3):4528–4535.

Bayer, J. and Faigl, J. (2019). On autonomous spatial exploration with small hexapod walking robot using tracking camera Intel RealSense T265. *European Conference on Mobile Robots*, pages 1–6.

Beuchert, J., Camurri, M., and Fallon, M. (2023). Factor Graph Fusion of Raw GNSS Sensing with IMU and Lidar for Precise Robot Localization without a Base Station. *IEEE International Conference on Robotics and Automation*, pages 8415–8421.

Bircher, A., Kamel, M., Alexis, K., Oleynikova, H., and Siegwart, R. (2016). Receding Horizon Next-Best-View Planner for 3D Exploration. *IEEE International Conference on Robotics and Automation*, pages 1462–1468.

Bircher, A., Kamel, M., Alexis, K., Oleynikova, H., and Siegwart, R. (2018). Receding horizon path planning for 3D exploration and surface inspection. *Autonomous Robots*, 42(2):291–306.

Bloesch, M., Omari, S., Hutter, M., and Siegwart, R. (2015). Robust visual inertial odometry using a direct EKF-based approach. *IEEE International Conference on Intelligent Robots and Systems*, pages 298–304.

Border, R. and Gammell, J. D. (2024). The Surface Edge Explorer (SEE): A measurement-direct approach to next best view planning. *International Journal of Robotics Research*, pages 1–25.

Chung, T. H., Orekhov, V., and Maio, A. (2023). Into the Robotic Depths: Analysis and Insights from the DARPA Subterranean Challenge. *Annual Review of Control, Robotics, and Autonomous Systems*, 6:477–502.

Coleman, D., Sucan, I., Chitta, S., and Correll, N. (2014). Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study. *Journal of Software Engineering for Robotics*, 5(1):1–14.

[9] `https://dynamic.robots.ox.ac.uk/datasets/oxford-osprey`

Cover, H., Choudhury, S., Scherer, S., and Singh, S. (2013). Sparse Tangential Network (SPARTAN): Motion planning for micro aerial vehicles. *IEEE International Conference on Robotics and Automation*, pages 2820–2825.

Dang, T., Mascarich, F., Khattak, S., Papachristos, C., and Alexis, K. (2019). Graph-based Path Planning for Autonomous Robotic Exploration in Subterranean Environments. *IEEE International Conference on Intelligent Robots and Systems*, pages 3105–3112.

Dellaert, F. and Kaess, M. (2017). *Factor Graphs for Robot Perception*. Foundations and Trends in Robotics, Vol. 6.

Dijkstra, E. W. (1959). A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1(1):269–271.

Fu, L. F. T., Chebrolu, N., and Fallon, M. (2023). Extrinsic Calibration of Camera to LIDAR using a Differentiable Checkerboard Model. *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Furgale, P., Rehder, J., and Siegwart, R. (2013). Unified Temporal and Spatial Calibration for Multi-Sensor Systems. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–7.

Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths.

Heng, L., Honegger, D., Lee, G. H., Meier, L., Tanskanen, P., Fraundorfer, F., and Pollefeys, M. (2014). Autonomous Visual Mapping and Exploration With a Micro Aerial Vehicle. *Journal of Field Robotics*, 31(4):654–675.

Hess, W., Kohler, D., Rapp, H., and Andor, D. (2016). Real-time loop closure in 2D LIDAR SLAM. *IEEE International Conference on Robotics and Automation*, pages 1271–1278.

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206.

Hudson, N., Talbot, F., Cox, M., Williams, J., Hines, T., Pitt, A., Wood, B., Frousheger, D., Lo Surdo, K., Molnar, T., Steindl, R., Wildie, M., Sa, I., Kottege, N., Stepanas, K., Hernandez, E., Catt, G., Docherty, W., Tidd, B., Tam, B., Murrell, S., Bessell, M., Hanson, L., Tychsen-Smith, L., Suzuki, H., Overs, L., Kendoul, F., Wagner, G., Palmer, D., Milani, P., O'Brien, M., Jiang, S., Chen, S., and Arkin, R. (2022). Heterogeneous Ground and Air Platforms, Homogeneous Sensing: Team CSIRO Data61's Approach to the DARPA Subterranean Challenge. *Field Robotics*, 2(1):595–636.

Karaman, S. and Frazzoli, E. (2011). Sampling-based Algorithms for Optimal Motion Planning. *International Journal of Robotics Research*, 30(7):846–894.

Khattak, S., Nguyen, H., Mascarich, F., Dang, T., and Alexis, K. (2020). Complementary Multi-Modal Sensor Fusion for Resilient Robot Pose Estimation in Subterranean Environments. *International Conference on Unmanned Aircraft Systems*, pages 1024–1029.

Kim, G., Choi, S., and Kim, A. (2022). Scan Context++: Structural Place Recognition Robust to Rotation and Lateral Variations in Urban Environments. *IEEE Transactions on Robotics*, 38(3):1856–1874.

Kompis, Y., Bartolomei, L., and Chli, M. (2021). Fully Autonomous Live 3D Reconstruction with an MAV. *9th International Conference on 3D Vision*, pages 3–4.

Kuffner, J. J. and La Valle, S. M. (2000). RRT-connect: an efficient approach to single-query path planning. *IEEE International Conference on Robotics and Automation*, pages 995–1001.

Kulkarni, M., Dharmadhikari, M., Tranzatto, M., Zimmermann, S., Reijgwart, V., De Petris, P., Nguyen, H., Khedekar, N., Papachristos, C., Ott, L., Siegwart, R., Hutter, M., and Alexis, K. (2022). Autonomous Teamed Exploration of Subterranean Environments using Legged and Aerial Robots. *IEEE International Conference on Robotics and Automation*, pages 3306–3313.

LaValle, S. M. (1998). Rapidly-Exploring Random Trees: A New Tool for Path Planning. Technical report, Iowa State University, Ames.

Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., and Furgale, P. (2014). Keyframe-Based Visual-Inertial SLAM using Nonlinear Optimization. *The International Journal of Robotics Research*, 34.

Maffra, F., Chen, Z., and Chli, M. (2018). Viewpoint-Tolerant Place Recognition Combining 2D and 3D Information for UAV Navigation. *IEEE International Conference on Robotics and Automation*, pages 2542–2549.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2020). NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *European Conference on Computer Vision*.

Mur-Artal, R., Montiel, J. M., and Tardos, J. D. (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163.

Museth, K. (2013). VDB: High-resolution sparse volumes with dynamic topology. *ACM Transactions on Graphics*, 32(3).

Nguyen, H., Mascarich, F., Dang, T., and Alexis, K. (2020). Autonomous Aerial Robotic Surveying and Mapping with Application to Construction Operations. *arXiv*.

Ohradzansky, M., Rush, E., Riley, D., Mills, A., Ahmad, S., McGuire, S., Biggie, H., Harlow, K., Miles, M., Frew, E., Heckman, C., and Humbert, J. (2022). Multi-Agent Autonomy: Advancements and Challenges in Subterranean Exploration. *Field Robotics*, 2(1):1068–1104.

Oleynikova, H., Taylor, Z., Fehr, M., Siegwart, R., and Nieto, J. (2017). Voxblox: Incremental 3D Euclidean Signed Distance Fields for on-board MAV planning. *IEEE International Conference on Intelligent Robots and Systems*, pages 1366–1373.

Papachristos, C., Khattak, S., Mascarich, F., Dang, T., and Alexis, K. (2019). Autonomous aerial robotic exploration of subterranean environments relying on morphology-aware path planning. *International Conference on Unmanned Aircraft Systems*, pages 299–305.

Qin, T., Li, P., and Shen, S. (2018). VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020.

Ramezani, M., Khosoussi, K., Catt, G., Moghadam, P., Williams, J., Borges, P., Pauling, F., and Kottege, N. (2022). Wildcat: Online Continuous-Time 3D Lidar-Inertial SLAM. *arXiv*.

Ramezani, M., Tinchev, G., Iuganov, E., and Fallon, M. (2020). Online LiDAR-SLAM for Legged Robots with Robust Registration and Deep-Learned Loop Closure. *IEEE International Conference on Robotics and Automation*, pages 4158–4164.

Rouček, T., Pecka, M., Čížek, P., Petříček, T., Bayer, J., Šalanský, V., Azayev, T., Heřt, D., Petrlík, M., Báča, T., Spurný, V., Krátký, V., Petráček, P., Baril, D., Vaidis, M., Kubelka, V., Pomerleau, F., Faigl, J., Zimmermann, K., Saska, M., Svoboda, T., and Krajník, T. (2022). System for multi-robotic exploration of underground environments CTU-CRAS-NORLAB in the DARPA Subterranean Challenge. *Field Robotics*, 2(1):1779–1818.

Scherer, S., Agrawal, V., Best, G., Cao, C., Cujic, K., Darnley, R., DeBortoli, R., Dexheimer, E., Drozd, B., Garg, R., Higgins, I., Keller, J., Kohanbash, D., Nogueira, L., Pradhan, R., Tatum, M., Viswanathan, V., Willits, S., Zhao, S., Zhu, H., Abad, D., Angert, T., Armstrong, G., Boirum, R., Dongare, A., Dworman, M., Hu, S., Jaekel, J., Ji, R., Lai, A., Lee, Y. H., Luong, A., Mangelson, J., Maier, J., Picard, J., Pluckter, K., Saba, A., Saroya, M., Scheide, E., Shoemaker-Trejo, N., Spisak, J., Teza, J., Yang, F., Wilson, A., Zhang, H., Choset, H., Kaess, M., Rowe, A., Singh, S., Zhang, J., Hollinger, G., and Travers, M. (2022). Resilient and Modular Subterranean Exploration with a Team of Roving and Flying Robots. *Field Robotics*, 2(1):678–734.

Schonberger, J. L. and Frahm, J. M. (2016). Structure-from-Motion Revisited. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113.

Shen, S., Michael, N., and Kumar, V. (2012). Autonomous indoor 3D exploration with a micro-aerial vehicle. *IEEE International Conference on Robotics and Automation*, pages 9–15.

Song, S., Kim, D., and Choi, S. (2021). View Path Planning via Online Multiview Stereo for 3-D Modeling of Large-Scale Structures. *IEEE Transactions on Robotics*, pages 1–19.

Song, S., Kim, D., and Jo, S. (2020). Online coverage and inspection planning for 3D modeling. *Autonomous Robots*, 44(8):1431–1450.

Strub, M. P. and Gammell, J. D. (2022). Adaptively Informed Trees (AIT*) and Effort Informed Trees (EIT*): Asymmetric bidirectional sampling-based path planning. *The International Journal of Robotics Research*, 41(4):390–417.

Sucan, I., Moll, M., and Kavraki, L. E. (2012). The Open Motion Planning Library (OMPL). *IEEE Robotics & Automation Magazine*, 19(December):72–82.

Tabib, W., Goel, K., Yao, J., Boirum, C., and Michael, N. (2022). Autonomous Cave Surveying With an Aerial Robot. *IEEE Transactions on Robotics*, 38(2):1016–1032.

Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., Mcallister, D., Kerr, J., and Kanazawa, A. (2023). Nerfstudio: A Modular Framework for Neural Radiance Field Development. *ACM SIGGRAPH*.

Tao, Y., Bhalgat, Y., Fu, L. F. T., Mattamala, M., Chebrolu, N., and Fallon, M. (2024). SiLVR: Scalable Lidar-Visual Reconstruction with Neural Radiance Fields for Robotic Inspection. *IEEE International Conference on Robotics and Automation*.

Teixeira, L., Alzugaray, I., and Chli, M. (2018). Autonomous Aerial Inspection Using Visual-Inertial Robust Localization and Mapping. *Springer Proceedings in Advanced Robotics*, 5:191–204.

Tranzatto, M., Dharmadhikari, M., Bernreiter, L., Camurri, M., Khattak, S., Mascarich, F., Pfreundschuh, P., Wisth, D., Zimmermann, S., Kulkarni, M., Reijgwart, V., Casseau, B., Homberger, T., De Petris, P., Ott, L., Tubby, W., Waibel, G., Nguyen, H., Cadena, C., Buchanan, R., Wellhausen, L., Khedekar, N., Andersson, O., Zhang, L., Miki, T., Dang, T., Mattamala, M., Montenegro, M., Meyer, K., Wu, X., Briod, A., Mueller, M., Fallon, M., Siegwart, R., Hutter, M., and Alexis, K. (2022). Team CERBERUS Wins the DARPA Subterranean Challenge: Technical Overview and Lessons Learned. *arXiv*.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Williams, J., Jiang, S., O'Brien, M., Wagner, G., Hernandez, E., Cox, M., Pitt, A., Arkin, R., and Hudson, N. (2020). Online 3D Frontier-Based UGV and UAV Exploration Using Direct Point Cloud Visibility. *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 263–270.

Wisth, D., Camurri, M., and Fallon, M. (2023). VILENS: Visual, Inertial, Lidar, and Leg Odometry for All-Terrain Legged Robots. *IEEE Transactions on Robotics*, 39(1):309–326.

Yoder, L. and Scherer, S. (2016). Autonomous Exploration for Infrastructure Modeling with a Micro Aerial Vehicle. *Field and Service Robotics*, 10:427–440.

Zhang, J. and Singh, S. (2014). LOAM: Lidar Odometry and Mapping in Real-time. *Robotics: Science and Systems*.

Zhao, S., Zhang, H., Wang, P., Nogueira, L., and Scherer, S. (2021). Super Odometry: IMU-centric LiDAR-Visual-Inertial Estimator for Challenging Environments. *IEEE International Conference on Intelligent Robots and Systems*, pages 8729–8736.