Search and rescue under the forest canopy using multiple UAVs

Yulun Tian\textsuperscript{1}\textsuperscript{*}, Katherine Liu\textsuperscript{1}, Kyel Ok\textsuperscript{1}, Loc Tran\textsuperscript{2}, Danette Allen\textsuperscript{2}, Nicholas Roy\textsuperscript{1} and Jonathan P. How\textsuperscript{1}

Abstract
We present a multi-robot system for GPS-denied search and rescue under the forest canopy. Forests are particularly challenging environments for collaborative exploration and mapping, in large part due to the existence of severe perceptual aliasing which hinders reliable loop closure detection for mutual localization and map fusion. Our proposed system features unmanned aerial vehicles (UAVs) that perform onboard sensing, estimation, and planning. When communication is available, each UAV transmits compressed tree-based submaps to a central ground station for collaborative simultaneous localization and mapping (CSLAM). To overcome high measurement noise and perceptual aliasing, we use the local configuration of a group of trees as a distinctive feature for robust loop closure detection. Furthermore, we propose a novel procedure based on cycle consistent multiway matching to recover from incorrect pairwise data associations. The returned global data association is guaranteed to be cycle consistent, and is shown to improve both precision and recall compared with the input pairwise associations. The proposed multi-UAV system is validated both in simulation and during real-world collaborative exploration missions at NASA Langley Research Center.

Keywords
Multi-robot systems, field robotics, SLAM

1. Introduction
Lost hikers are often within a mile or two of the last point of detection for extended periods of time, but are undetected for hours because manned aircraft cannot see them through the overhead forest canopy. Instead, a team of small autonomous unmanned aerial vehicles (UAVs) can be deployed under the tree canopy to gain better visibility during search and rescue missions in forest areas; these vehicles can be rapidly deployed, can collaboratively explore expanses of terrain efficiently, and are agile enough to operate in reasonably thick forests, such as shown in Figure 1.

Deploying a multi-UAV system for forest search and rescue presents a multitude of technical challenges. First, signals from the Global Positioning System (GPS) typically cannot penetrate the overhead forest canopy. Thus, for independent and real-time operations, vehicles must be able to perform GPS-denied flights using onboard sensing, estimation, and planning. Second, a collaborative mapping system would require a map fusion procedure that accurately detects overlaps between individual maps and fuses these maps into a globally consistent model of the environment. In the forests, however, correctly matching multiple overlapping maps is a challenging problem due to severe perceptual aliasing caused by the visual similarities of trees. Lastly, communication constitutes another prerequisite for collaborative exploration and mapping. However, in forest search and rescue scenarios, communication is typically unreliable and intermittent due to signal attenuation and limited transmission range.

In this work, we present a multi-UAV system for efficient exploration in large-scale unknown forest environments. Our vehicles are equipped with onboard autonomy that efficiently performs autonomous sensing, vehicle state estimation, local mapping, and exploration planning. When communication is available, map fusion is achieved by performing collaborative simultaneous localization and mapping (CSLAM) at a central ground station. We overcome

\textsuperscript{*}Corresponding author:
Yulun Tian, Massachusetts Institute of Technology, 70 Vassar Street, Cambridge, MA 02139-4307, USA.
Email: yulun@mit.edu

\textsuperscript{1}Massachusetts Institute of Technology, Cambridge, MA, USA
\textsuperscript{2}NASA Langley Research Center, Hampton, VA, USA
Fig. 1. Autonomous flight of one of our UAVs in a forested area within NASA Langley Research Center (LaRC). (a) Image of one of our UAVs during a collaborative mapping mission in a forest. (b) Raw point cloud collected by onboard laser scanner. Each individual point is colored according to its height from the ground. The vehicle pose estimated by the onboard extended Kalman filter (EKF) is represented as the laser coordinate frame. A complete video of two UAVs autonomously exploring and collaboratively mapping the forest is available at: https://groups.csail.mit.edu/rrg/videos/isere2018.

the challenge of limited communication bandwidth by leveraging two different map representations: a dense 3D volumetric grid representation for onboard trajectory planning, and a compact tree-based representation for lightweight communication with the ground station during CSLAM.

Our onboard sensing, estimation, and mapping modules are based on a 2D LIDAR sensor, where laser scans are aligned frame-to-frame for incremental vehicle pose update, collected into a 3D occupancy grid for trajectory planning, and compressed into a tree-based map for lightweight communication. We adopt a submap-based grid representation (Leonard and Feder, 2001), i.e., a collection of locally segmented grid maps as opposed to a single occupancy grid, which allows the origin of each submap in the world to be optimized to correct for onboard estimation drifts after loop closure. Our submapping approach also bounds the area that is updated after incorporating each new scan, but unlike the sliding-window mapping approach, e.g., by Schmuck and Chli (2018), all previous submaps are stored and used by a dynamics-aware frontier-based planner during exploration. While dense local submaps are useful for onboard planning processes, naive transmission of rich geometric occupancy information poses high bandwidth requirements. For lightweight communication, we compress local maps into object-based maps by detecting tree stems in each laser scan.

On the ground station, we perform CSLAM to jointly optimize all submap origins, which determine how each submap should be positioned in the world frame, as well as tree positions in the fused map. To do so, we first detect overlaps across individual submaps by performing loop closure detection. Instead of directly matching individual trees, which is prone to error due to their visual similarities, we match groups of trees using their relative positions and orientations as a unique signature. To gain additional robustness against perceptual aliasing, we include a novel stage during our data association process that jointly optimizes the set of all pairwise matches based on the criterion of cycle consistency (Aragues et al., 2011). Based on the final tree-to-tree correspondences between submaps, we perform landmark simultaneous localization and mapping (SLAM) that jointly optimizes all submap origins and tree positions in the global map. This global map may then be inspected by the human operators at the ground station, and transmitted back to the vehicles to correct estimation drifts during onboard state estimation and mapping.

We demonstrate the proposed system in a small forest at NASA Langley Research Center (LaRC), where a team of two UAVs are each assigned a search region to completely explore (Figure 1). Our experimental results show promising progress towards the practical use of multi-UAV systems for search and rescue under the forest canopy.

1.1. Contributions

In this article, we propose a novel collaborative mapping and exploration system that overcomes the unique challenges associated with forest search and rescue. It is widely recognized that data association (e.g., matching landmarks or feature points) is among the hardest problems for single and multi-robot SLAM. In a forest, data association is even harder due to large amount of noise caused by moving branches and leaves, and severe perceptual aliasing between different parts of the environment. To overcome this problem, we utilize a novel tree-level map representation, and leverage both pairwise and multiway matching techniques to improve the reliability of landmark association and guarantee cycle consistency. Using precision recall as the performance critical metric, we show that our approach significantly outperforms the baseline approach in a real-world forest environment.

In addition, for search and rescue, the efficiency to explore a designated search area is among the most crucial
performance factors. Classical closest-frontier-based exploration strategy is inefficient for a UAV, as it requires frequent heading changes to visit frontiers that are behind the vehicle. To address this issue, we utilize a heuristic for selecting frontiers that better respects vehicle dynamics. Using a high-fidelity simulation environment, we demonstrate the advantages of this approach over standard frontier-based exploration, in terms of both average vehicle speed and time efficiency.

An early version of this work was presented at ISER 2018 (International Symposium on Experimental Robotics) (Tian et al., 2018). This paper extends our previous work with the following contributions.

1. Extension to a deformable submap-based representation for local mapping (Section 4.1). Using a high-fidelity simulation environment, we show that the approach effectively corrects onboard estimation drift accumulated by an extended Kalman filter (EKF).
2. Novel use of cycle consistent multiway matching for data association during multi-robot landmark SLAM (Section 5.3). We show that this approach further improves precision-recall performance compared with our previous work.
3. Additional experimental results that demonstrate the usefulness of the proposed multi-UAV exploration and mapping system (Sections 6 and 7).

1.2. Outline

The rest of this article is organized as follows. We first review previous work in Section 2 and give an overview of the proposed system in Section 3. We describe in detail the onboard autonomy in Section 4 and the offboard CSLAM pipeline in Section 5. We present simulation and real flight experiments and analyze the results in Sections 6 and 7. Finally, we share main experimental insights learned and suggest future work in Section 8.

2. Related work

2.1. Multi-robot mapping and exploration

Multi-robot localization and mapping has been an active field of research. For a comprehensive survey, see also Saeedi et al. (2016), Choudhary et al. (2017), Schmuck and Chli (2018), and the references therein. Forster et al. (2013) were among the first to propose a centralized CSLAM system for multiple UAVs. Each UAV runs visual odometry onboard and relays its keyframes and relative pose estimates to a ground station. The ground station detects loop closures among local maps and performs map fusion when necessary. Schmuck and Chli (2018) extend this framework by sending optimized keyframes and map points back to the vehicles to improve accuracy of local mapping.

On the other hand, Dong et al. (2015), Morrison et al. (2016), and Schuster et al. (2019) propose to run full SLAM onboard each vehicle. The incurred computation costs are further reduced in distributed architectures, where each robot only optimizes its local map and shares the compressed map or boundary poses with each other, see Cunningham et al. (2013), Choudhary et al. (2017), and Cieslewski et al. (2018).

While distributed systems have better scalability compared with centralized ones, they do not leverage the opportunity to transfer expensive onboard operations to the ground station which has more computational power and is not limited by resource constraints. Furthermore, for search and rescue, a fused map ultimately has to be created on the ground station and presented to the first responders. For these reasons, in this work we choose to implement a centralized architecture similar to those presented in Forster et al. (2013) and Schmuck and Chli (2018).

The problem of multi-robot exploration has been well studied in the literature. Commonly used exploration strategies include nearest frontier selection (Yamauchi et al., 1998), next-best view (Bircher et al., 2016; González-Baños and Latombe, 2002), greedy or receding horizon information gain (Bourgault et al., 2002; Feder et al., 1999), or a mixture of these methods (Charrow et al., 2015). More recently, exploration using a single or multiple UAVs has become increasingly popular (Cesare et al., 2015; Cieslewski et al., 2017; Heng et al., 2015; Shen et al., 2012; Yoder and Scherer, 2016). Among these works, Cesare et al. (2015) demonstrated a frontier-based exploration algorithm for multiple UAVs in an indoor environment. To accelerate exploration, Cieslewski et al. (2017) proposed to select frontiers that are reachable with minimum change in velocity. In this work, we draw similar intuitions and implement a dynamics-aware strategy for exploration.

Another line of research focuses on coordinated multi-robot exploration. Burgard et al. (2000, 2005) embed coordination into the exploration strategy by reducing the utilities of frontier cells that are seen by other robots. Experiments show that this approach results in more efficient exploration compared with the uncoordinated approach of Yamauchi et al. (1998). The same approach was also adopted and extended by later works (Fox et al., 2006; Stachniss et al., 2008). This work differs with the aforementioned works in two aspects. First, earlier works frequently assume that loop closure can be performed straightforwardly, e.g., via scan matching. In a forest, however, standard scan matching tends to fail due to large amount of noise caused by moving branches and leaves. In this work, we propose and demonstrate a CSLAM system that leverages prior structures within the environment to achieve reliable loop closure and map fusion. Second, Burgard et al. (2005) and Stachniss et al. (2008) also relied on a central hub to assign frontiers to robots. In practice, this centralized approach sometimes create unnecessary latency in the system (e.g., a vehicle could become idle while waiting for the next frontier from the ground station). We overcome this problem by implementing a distributed
exploration strategy where each vehicle independently explores a non-overlapping search region.

2.2. Multi-robot loop closure and data association

One major contribution of this work is the design of a reliable multi-robot loop closure and map fusion system for the forest environment. In this section, we review existing techniques on multi-robot loop closure and data association, which can be largely categorized into pairwise and multiway methods.

2.2.1. Pairwise methods. Detecting correspondences between two sets of feature points is a well-studied problem in robotics. Classical geometry-based methods such as nearest neighbor, maximum likelihood, and joint compatibility branch and bound (JCBB) associate pairs of landmarks based on their Euclidean or Mahalanobis distances in a common reference frame (Gil et al., 2010; Kaess and Dellaert, 2009; Neira and Tardos, 2001; Zhou and Roumeliotis, 2006). When a common reference frame is not available, methods based on local geometric features (Cunningham et al., 2012), generalized Hough transform (Paz et al., 2005), or maximum common subgraph (MCS; Bailey et al., 2000) can be used instead. Many of the above techniques can be combined with random sample consensus (RANSAC) (Fischler and Bolles, 1981) for enhanced robustness.

Recently, appearance-based methods have shown great success in large-scale place recognition (Cummins and Newman, 2011; Paul and Newman, 2010) and distributed visual SLAM (Cieslewski et al., 2018). Many of these methods start by detecting local features from input images or laser scans. Each feature point is typically associated with a descriptor. For visual data, popular descriptors include scale-invariant feature transform (SIFT; Lowe, 2004), speeded up robust features (SURF; Bay et al., 2006), binary robust independent elementary features (BRIEF; Calonder et al., 2010), and ORB (oriented FAST and rotated BRIEF; Rublee et al., 2011). For laser-range data, similar descriptors have been developed which include Gestalt (Bosse and Zlot, 2009), fast point feature histograms (FPFH; Rusu et al., 2009), and fast laser interest region transform (FLIRT; Tipaldi and Arras, 2010).

Although feature extraction is very fast, the number of features needed for tracking and loop closure detection may still pose a challenge for real-time computation and communication. To address this issue, recent works begin to use more lightweight representations based on semantic and object-level models. Choudhary et al. (2017) used objects detected by convolutional neural networks during distributed pose graph optimization. Dubé et al. (2018) represented objects with segments in 3D point clouds and used data-driven descriptors for loop closure detection.

In the forest environment, prior works have recognized trees as the distinctive objects that can aid localization and mapping (Asmar et al., 2006; Kukko et al., 2017; Öhman et al., 2008; Tang et al., 2015). In this work, we also use trees as informative features for data association and loop closure. However, unlike prior works such as Kukko et al. (2017), we develop on top of tree features a complete data association procedure that leverages both pairwise and cycle consistent multiway matching. In our experiments, we demonstrate our system in a real-world multi-robot exploration task beneath the forest canopy.

In addition, state-of-the-art SLAM systems often implement a place recognition module that aggregates local features into a global descriptor for fast detection of potential loop closures. For images, notable examples include bag of visual words (Gálvez-López and Tardos, 2012), vector of locally aggregated descriptors (VLAD; Jégou et al., 2010), Fisher vectors (Jégou et al., 2012; Perronnin et al., 2010), and deep-learning-based methods such as NetVLAD (Arandjelović et al., 2016). For laser scans, Himstedt et al. (2014) developed the Geometric LAndmark RElations (GLARE) signature, which was later extended by Kallasi and Rizzini (2016) to be rotationally invariant in GLAROT. Note that comparing global descriptors only yields a set of candidate matches. State-of-the-art SLAM systems still employ a geometric verification step to validate the proposed associations (see, e.g., Mur-Artal and Tardós, 2017).

2.2.2. Multiway methods. For robust data association in robotics, multiway association that simultaneously considers multiple noisy pairwise associations is often used. One prominent example is the problem of rejecting outlier loop closures in pose graph SLAM (Dong et al., 2015; Indelman et al., 2014; Mangelson et al., 2018). Among the different ways of performing multiway association, a family of techniques closely related to our work attempts to achieve cycle consistent multi-robot data associations (Aragues et al., 2011; Montijano et al., 2013). Specifically, these works seek to resolve inconsistencies in the data associations; for example, such inconsistencies can appear when a chain of spurious pairwise associations matches two landmarks observed by the same robot. Aragues et al. (2011) proposed a heuristic based on cycle detection to detect and resolve these inconsistencies.

Similarly in the computer vision community, cycle consistency has also gained considerable attention owing to popular applications such as multi-shape matching (Huang and Guibas, 2013; Nguyen et al., 2011) and multi-image matching (Leonardos et al., 2017; Zhou et al., 2015). Principled approaches based on spectral relaxation (Pachauri et al., 2013), semidefinite relaxation (Chen et al., 2014), distributed consensus (Leonardos et al., 2017), and spectral clustering (Fathian et al., 2019) have been proposed for solving this problem, and performance guarantees for exact matching are proved under certain noise models.
(Chen et al., 2014; Pachauri et al., 2013). In this work, we propose a novel application of cycle consistent multiway matching for fusing tree landmarks during multi-robot data association and SLAM.

3. Overview

To enable independent exploration of large-scale GPS-denied forest environments, we equip each of our UAVs with an onboard autonomy module that performs autonomous sensing, vehicle state estimation, local mapping, and exploration planning; see Figure 2(a). We adopt a filtering-based approach to infer the vehicle state, while merging incoming laser scans into an occupancy grid for real-time obstacle avoidance and exploration planning. When communication is available, each vehicle also transmits its local observations to a central ground station to contribute to the collaborative mapping process. To cope with the limited communication bandwidth, these local observations are first compressed into lightweight submaps consisting only of detected tree objects in the environment. A detailed description of each component of the onboard autonomy module is provided in Section 4.

The ground station performs CSLAM, which fuses noisy measurements (in the form of tree-based submaps) from different vehicles into a globally consistent map of the world. This is achieved via a two stage process that consists of: (i) global data association, which recovers the underlying correspondences between trees observed across multiple submaps and possibly by different vehicles; and (ii) landmark SLAM, which jointly optimizes all submap origins and tree positions in the global map. See Figure 2(b) for the corresponding system diagram. The optimized submap origins are transmitted back to the vehicles to correct estimation drifts during local mapping. We note that the onboard CSLAM can happen in parallel to the onboard autonomous operations of each vehicle. This centralized architecture provides sufficient tolerance against network failures which in our architecture would only delay potential map fusion, but will not hinder real-time onboard operations. We describe the centralized CSLAM pipeline in detail in Section 5.

4. Onboard autonomy

Each of our vehicles achieves full onboard autonomy by estimating the vehicle pose in real-time and simultaneously optimizing over a local geometric map for obstacle avoidance and exploration planning. The rest of this section is organized as follows. We first describe our approach for laser-based vehicle pose estimation and 3D occupancy grid mapping (Section 4.1). We then describe our frontier-based exploration planning algorithm (Section 4.2), and lastly our object-based map compression scheme suitable for low-bandwidth communication in the forest (Section 4.3).

4.1. State estimation and local mapping

Forests are challenging environments for lightweight onboard sensing. Vision-based sensors, such as RGB cameras that provide rich information about the environment, suffer from occlusions between trees, large lighting changes with movements of foliage in the wind, visual similarities between trees, and limited fields of view.

Following Giamou et al. (2017), we instead adopt a 2D Hokuyo LIDAR as our main onboard sensor as it provides a large field of view (270°), accurate depth information, and robustness to visual aliasing and lighting changes, and is also lightweight enough to be carried by a smaller UAV. In addition to the LIDAR, we also utilize a single-point downward-facing laser altimeter and an inertial measurement unit (IMU), both of which provide additional information about vehicle state.

We combine the raw measurements produced by the suite of onboard sensors in a laser-based EKF (Bachrach et al., 2011) that estimates the six-degree-of-freedom pose of the vehicle. We first estimate the incremental motion between two consecutive laser scans using the iterative closest point (ICP) algorithm, and subsequently fuse this estimate with height measurements from the laser-based altimeter and inertial measurements produced by the IMU. Owing to dense geometric information provided by laser scans as well as the close similarity between consecutive scans, ICP in practice often provides accurate estimate of incremental vehicle motion. Furthermore, ICP is also computationally efficient, which makes it favorable for high-rate state estimation. In our implementation, we run ICP onboard at the same rate as raw laser scan comes in (40 Hz).

Each vehicle uses the real-time state estimates from EKF to build a map of the explored region. In our previous work (Tian et al., 2018), each onboard map was a 3D occupancy grid produced using Octomap (Hornung et al., 2013). While occupancy grid is a popular map representation especially for the purpose of path planning, one of its drawbacks is that it is not easily amenable to correction of previous mapping errors. In our case, such corrections are often needed to account for onboard estimation drifts, which is important for the vehicle to have a more accurate understanding of the environment (e.g., in terms of explored and unknown space) and adjust its decision intelligently to maximize search efficiency.

In this work, we extend our onboard map representation to a submap-based representation (Leonard and Feder, 2001). Each onboard map is stored as a collection of smaller 3D occupancy grids. This approach makes the onboard map deformable: while each submap is locally rigid, the relative transformations between submaps can be optimized using the refined pose estimates provided by CSLAM. This allows us to correct the errors accumulated during onboard vehicle estimation. An additional benefit is that at any time, the vehicle only needs to update its current submap which is computationally more efficient.
In our implementation, we initialize a new empty submap (Octomap) on each vehicle after a fixed amount of time. To update the current submap, we transform each incoming laser scan from the local sensor frame to the coordinate frame of the submap. The resulting point cloud is then used to update the 3D octomap. Although each submap only provides local information of a small area, the union of submaps contains the same amount of information as a single global map. Therefore, while we only update the current submap during onboard mapping, we use the union of the submaps for onboard motion and exploration planning.

Several state-of-the-art systems (e.g., Schmuck and Chli, 2018) used a sliding window approach for local mapping to limit the onboard memory usage. However, such approach that discards old information is not suitable for our vehicles because they must keep track of all areas previously visited during exploration. In addition, we note that in practice the memory usage per vehicle is relatively small as each vehicle is tasked with exploring a reasonably bounded search region.

4.2. Frontier-based planning

We adopt a frontier-based algorithm for onboard exploration, where a frontier refers to a location in the current map that is on the boundary of known and unknown space. We select frontier-based exploration as it is computationally lightweight, which makes it easy to run distributed onboard each vehicle. After the frontier-based planner selects the next frontier to visit, a motion planner based on A* search (Hart et al., 1968) is used on the projected 2D occupancy map to plan the optimal collision-free path for the vehicle. Exploration finishes when there are no more frontiers left in a designated search area.

While frontier-based algorithms are computationally efficient and, thus, more suitable for lightweight platforms, finding the best frontier for exploration using a UAV can be a challenging problem. Classic approaches usually select the closest frontier based on the Euclidean distance (Yamauchi et al., 1998). However, when using maps built with our onboard mapping system, it is often the case that the closest frontier lies directly behind the vehicle, in the blindspot of the 270° field-of-view LIDAR. Selecting these frontiers would yield rapidly changing vehicle orientations that ultimately lead to a poor mapping performance and a lack of progress towards searching uncovered ground.

In this work, we select frontiers based on a hybrid cost function that accounts for both the distance to frontiers and the vehicle dynamics, similar to the technique in Cieslewski et al. (2017). We define the cost associated to a frontier $a$ as

$$J(a) = J_d(a) + \lambda J_\theta(a)$$

where $J_d(\cdot)$ is the classic Euclidean distance of $a$, $J_\theta(\cdot)$ is the change in orientation required to reach $a$ from the vehicle’s current heading, and $\lambda$ is a balance parameter that trades off the two cost terms. In practice, the choice of $\lambda$ depends heavily on the flight speed. At lower speed, our vehicle can handle more aggressive heading change, which means that $\lambda$ can be increased to place more weight on the distance to frontier. In our outdoor experiments, we have commanded both vehicles to fly at a constant speed of 2.0 m/s. At this speed, we observed that our choice of $\lambda = 0.5$ leads to the best empirical performance.

Directly incorporating the orientation change in the cost function discourages the vehicle from excessive turning and hence produces a smoother overall trajectory. In addition, accounting for orientation change also enables continuous update of the next-best frontier, i.e., at any time during flight, the planner can replace the currently selected frontier with a new frontier whose cost is significantly lower. Owing to the orientation cost term $J_\theta(\cdot)$, switching to the new frontier typically produces minimal heading changes and, thus,
encourages safe flight behavior. In contrast, a dynamics-agnostic planner, e.g., with a purely distance-based cost, has to wait until the vehicle reaches the current frontier to be able to select the next frontier, which is not time efficient.

4.3. Tree-based map compression

While dense volumetric maps provide accurate geometric information for path planning, they are not data efficient and, hence, are not suitable for communication over a low-bandwidth wireless network. Other representations such as 3D point clouds (Schuster et al., 2019) or feature-based maps (Schmuck and Chli, 2018) can be more lightweight, but both can still result in potentially heavy data payload.

To achieve lightweight communication, we turn to object-based representations, more specifically a tree-based representation (Kukko et al., 2017). Before transmission, we compress each submap into a sparse collection of trees as shown in Figure 3(a). This is done by first clustering beams in each LIDAR scan using Dirichlet process clustering (DP-means) (Campbell et al., 2013). Each cluster of points is first projected to the 2D plane, and then fitted to a circle that represents a tree trunk. We first apply an algebraic fitting algorithm (Taubin, 1991) that estimates a tree stem radius. (b) The corresponding GLARE signature. Horizontal and vertical axes show discretizations with respect to orientation and distance, respectively. Lighter color denotes higher values.

Fig. 3. (a) Example tree submap generated from real forest data collected at NASA LaRC. Black dots show the raw laser returns over a period of 20 seconds. Each magenta circle shows a tree tracked within this submap, where the radius corresponds to the estimated tree stem radius. (b) The corresponding GLARE signature. Horizontal and vertical axes show discretizations with respect to orientation and distance, respectively. Lighter color denotes higher values.

For improved stability, we also perform tree tracking and culling within each submap. At every time step, a newly detected tree is combined with the closest previously detected tree in the same submap if their estimated positions are within 0.5 m and their radii differ less than 10%. If the above two conditions hold, the observation count of the original tree is incremented by one; if not, a new tree is initialized in the current submap using parameters of the newly detected tree. Before transmitting to the ground station, we optionally perform culling inside the submap, by removing trees with observation counts less than a threshold \( \tau_{\text{cull}} \). Here, \( \tau_{\text{cull}} \) is a tunable parameter designed to cope with varying level of noise during tree detection.

Our current tree-detection algorithm assumes that trees can be represented as cylinders in three dimensions (or circles after projection to two dimensions). In practice, we find that this approximation works well if the vehicle is flying at an altitude at which most tree trunks can be seen clearly. On the other hand, performance of this method will degrade for denser forests with more branches and leaves. In these scenarios, more sophisticated detection algorithms are needed to improve robustness.

5. Offboard CSLAM

On the ground station, we construct a globally consistent map of the world by fusing identical trees detected across multiple submaps and performing landmark SLAM over the fused trees. The overall problem is combinatorial in nature and, thus, difficult to solve optimally; however, practical applications, e.g., search and rescue, often require a solution that is efficient and, ideally, real-time. In order to meet such performance demand, we make use of the following centralized pipeline to efficiently (albeit approximately) solve the global mapping problem on the ground station; see Figure 2(b). First, we identify potential matches between pairs of submaps, i.e., potential loop closures, using a compact descriptor developed for 2D point landmarks (Section 5.1). We then verify each potential match by solving for the pairwise correspondences between trees in the two submaps (Section 5.2). Finally, the set of all pairwise associations are verified based on cycle consistency (Section 5.3) to overcome perceptual aliasing. Given the final data association results, we perform landmark SLAM to jointly optimize all submap origins and tree positions in the global map (Section 5.4).

5.1. Fast loop closure candidate detection

To efficiently detect potential loop closures between pairs of submaps, we build a GLARE (Himstedt et al., 2014) descriptor for each received submap on the ground station. GLARE is an efficient method for encoding relative landmark (i.e., tree) positions in a single map. For
every pair of trees \((i, j)\), we compute their distance \(\rho_{ij}\) and absolute relative angle \(\theta_{ij}\). We then assign these values to bins in a 2D histogram \((\theta_{ij}^+, \rho_{ij}^+)) \in \text{bin}(n_0, n_\rho)\), where \(n_0 \in \{1, 2, \ldots, N_t\}\) and \(n_\rho \in \{1, 2, \ldots, N_\rho\}\) are indices corresponding to the quantization of \((\theta_{ij}, \rho_{ij})\) in the range \([0, \rho_{\text{max}}]\) m \times [0, \pi]\) rad. To account for noise during tree detection, we apply a Gaussian blur with standard deviation 0.1 to the histogram of each feature pair. The GLARE descriptor is computed as the sum of the 2D histograms over all pairs. Following Giamou et al. (2017), we set the default histogram resolutions to be \(N_\rho = 120, N_\theta = 12\). In our experiments, we have not observed significant impact of the discretization parameters on the performance of loop closure detection. Figure 3(b) shows the GLARE signature corresponding to the submap in Figure 3(a).

To determine whether two submaps are likely to be a loop closure, we compare their GLARE descriptors using the GLAROT procedure proposed by Kallasi and Rizzini (2016). GLAROT provides a rotation-invariant distance metric, by computing the shifted \(L_1\) distance between two GLARE descriptors \(G^i, G^j \in \mathbb{R}^{N_\rho \times N_\theta}\),

\[
SL_1(G^i, G^j) = \min_{0 \leq l < N_\rho} \sum_{i=0}^{N_\rho-1} \sum_{j=0}^{N_\rho-1} |G^i_{l,j} - G^j_{l+(i+l) \mod N_\rho}|\tag{2}
\]

Each GLAROT query requires \(N_\rho \times N_\rho^2\) operations, which is very fast on standard CPUs. A pair of submaps \((M^i, M^j)\) is declared to be a loop closure candidate, if their GLAROT distance \(SL_1(G^i, G^j)\) is below a threshold \(\varepsilon_{\text{GLAROT}}\). This process serves as an efficient filtering step that reduces the search space for loop closures to a small set of candidate matches. To verify whether each candidate is a true loop closure, we apply a pairwise data association procedure which identifies the tree-to-tree correspondences between the two submaps.

5.2. Loop closure verification

For each potential loop closure between a pair of submaps \((M^i, M^j)\), we verify if \((M^i, M^j)\) forms a true loop closure by solving an element-wise data association problem that returns the correspondences between trees in the two submaps. Note that as \(M^i\) and \(M^j\) could potentially come from different vehicles, classical pairwise association methods such as nearest neighbor or JCBB, which rely on a common reference frame could not be used. Instead, we perform correspondence graph (CG) matching (Bailey et al., 2000), which is observed to work better compared with alternative methods based on the generalized Hough transform or RANSAC (Giamou et al., 2017). Let \([o_i^j]\) and \([o_j^i]\) represent the set of objects in \(M^i\) and \(M^j\). In addition, let \([p_i^j]\) and \([p_j^i]\) be the positions of these objects in the coordinate frames of \(M^i\) and \(M^j\), respectively. A CG is an undirected graph in which each vertex \(u_{ij}^{st}\) denotes a hypothetical match between \(o_i^j \in M^j\) and \(o_j^i \in M^i\). Two vertices \(u_{ij}^{st}, u_{ki}^{st}\) are connected Two vertices \(u_{ij}^{st}, u_{kj}^{st}\) are connected by an edge (i.e., they are geometrically consistent) if relative distances between landmarks are preserved, i.e., if,

\[
|\|p_i^j - p_k^j\|_2 - \|p_j^i - p_l^i\|_2| \leq \epsilon_{\text{CG}},
\]

where \(\epsilon_{\text{CG}}\) is an adjustable tolerance parameter to account for noise in real measurements.

Given the CG, we find the maximal set of pairwise compatible correspondences between \(M^i\) and \(M^j\) by finding the maximum clique. The resulting data association can be written as a partial permutation \(\tilde{\pi}_t\) which maps a subset of objects in \(M^i\) to \(M^j\). We use the tilde notation as a reminder that \(\tilde{\pi}_t\) could still contain wrong associations, e.g., owing to strong perceptual aliasing in the forest. For numerical computations in the next section, each \(\tilde{\pi}_t\) is also represented as a partial permutation matrix \(\tilde{\Pi}_t^j \in \{0, 1\}^{m^t \times m^j}\), where

\[
\tilde{\Pi}^j_t(i, j) = \begin{cases} 
1, & \text{if } \tilde{\pi}_t(i) = j \\
0, & \text{otherwise}
\end{cases}
\]

and \(m^t\) and \(m^j\) are the sizes of the two submaps. Here \(M^i\) and \(M^j\) are declared to be a loop closure, if \(\tilde{\pi}_t\) contains sufficient matches, specified by a tunable parameter \(\epsilon_{\text{CG}}\).

We make a note on the complexity of CG matching. Although the maximum clique problem is combinatorial in nature, our object-based representation allows us to solve it efficiently and in real-time. To solve the maximum clique problem, we use the implementation provided by Konc and Janežič (2007). Empirical runtime results are reported in Section 7.

5.3. Globally consistent loop closures

Given a collection of noisy pairwise associations \(\{\tilde{\pi}_t^j\}\) between \(n\) submaps, we perform a final optimization to recover the global association \(\{\pi_u^j\}\), which maps objects in each submap \(M^u\) to the universe \(U\) of objects, i.e., the set of all trees in the forest. Recovering \(\{\pi_u^j\}\) requires the pairwise associations \(\{\tilde{\pi}_t^j\}\) to be cycle consistent, i.e., the composition of pairwise mappings along any cycle of submaps should result in the identity mapping (Nguyen et al., 2011).

To better understand the importance of cycle consistency, we look at the data association graph \(G\) (Leonardos et al., 2017) induced by \(\{\tilde{\pi}_t^j\}\). With a slight abuse of notation, we let each vertex in \(G\) represent an object \(o_i^j\) in a submap \(M^j\). Two vertices \(o_i^j, o_j^i\) are connected by an edge if they are matched in the input associations, i.e., if \(\tilde{\pi}_t(i) = j\). We note that these edges are inherently transitive: if \(o_i^j\) is matched to \(o_j^i\) and \(o_j^i\) is matched to \(o_k^j\), then it must be true that \(o_i^j\) is matched to \(o_k^j\). Tron et al. (2017) showed that \(\{\tilde{\pi}_t\}\) is cycle consistent if and only if the corresponding data association graph consists of disjoint cliques, and furthermore no two objects from the same submap appear in the same clique. In this case, map fusion happens naturally by assigning each clique in \(G\) to a unique object in the universe \(U\). Figure 4(a) provides an example.

Fig. 4. Example data association graphs. Each vertex (circle) represents an object (tree) in a submap. Two vertices are connected if the corresponding objects are matched in the input pairwise associations. (a) Cycle consistent induced graph: if the input associations are cycle consistent, objects matched together will form disjoint cliques. In this case, it is straightforward to read out the global data association by assigning each clique to an object in the universe. (b) Inconsistent induced graph: in practice, a chain of spurious associations (red edges) might violate cycle consistency and incorrectly fuse two objects in the same submap (red dashed edge). We remove such noisy associations based on cycle consistency.

In practice, however, \{π′_i\} often contain noisy matches that violate the cycle consistency principle; an example is shown in Figure 4(b). After invoking the transitive property of all edges, a chain of spurious associations (red edges) would incorrectly fuse two objects belonging to the same submap, which contradicts the assumption that each submap contains distinct objects. Therefore, for map fusion, we must find a cycle consistent set of pairwise associations \{π_i\} that resembles the input associations \{π′_i\} as much as possible. Given \{π_i\}, it is again straightforward to recover the underlying global associations \{π_s\}, by assigning each clique in the induced data association graph of \{π_i\} to a unique object. In the computer vision literature, this procedure is known as cycle consistent multiway matching (Pachauri et al., 2013).

Among all algorithms proposed for cycle consistent multiway matching, we choose CLEAR (consistent lifting, embedding, and alignment rectification) proposed by Fahim et al. (2019) for its high precision and superior speed. CLEAR takes as input the aggregate matrix,

\[
A = \begin{bmatrix}
0 & \tilde{\Pi}_1^1 & \cdots & \tilde{\Pi}_1^n \\
\tilde{\Pi}_1^1 & 0 & \cdots & \tilde{\Pi}_2^n \\
\vdots & \vdots & \ddots & \vdots \\
\tilde{\Pi}_1^n & \tilde{\Pi}_2^n & \cdots & 0
\end{bmatrix}
\]

which corresponds to the adjacency matrix of G. Let D denote the (diagonal) degree matrix of G. The Laplacian and normalized Laplacian of G are defined as follows,

\[
L \triangleq D - A, \quad L_{\text{rm}} \triangleq D^{-\frac{1}{2}}LD^{-\frac{1}{2}}
\]

From here, the algorithm proceeds in two stages. In the first stage, the number of objects m in the universe is estimated from the spectrum of the normalized Laplacian \(L_{\text{rm}}\). In the second stage, an eigendecomposition is performed over \(L_{\text{rm}}\) to obtain an embedding vector \(v'_i \in \mathbb{R}^m\) for each vertex \(o'_i\) in G. These embeddings are then used to cluster vertices into m disjoint cliques using the Hungarian algorithm. When computing the embedding vectors, we use a block singular value decomposition (SVD) method that leverages the separable structure of the underlying data association graph. To further accelerate computation, we also implemented the greedy method for clustering as suggested in Fathim et al. (2019).

The final global data association \{π_s\} is obtained by assigning vertices in each clique to a unique object in the universe, i.e., \(π_s(i) = c\) if vertex \(o'_i\) belongs to clique \(c\). In addition, \(π_i\) can also be used to recover a cycle consistent pairwise matching simply via

\[
π_i = π_s \circ π_i^0
\]

where \(\circ\) denotes the composition operator for permutations.

5.4. Global landmark SLAM

After solving for the global data association \{π_s\}, we perform SLAM to refine the estimates of all poses and objects in the fused map. To do this, we jointly optimize over the origin \(x_i \in \text{SE}(2)\) of each submap \(M_i\), which also corresponds to the vehicle pose when \(M_i\) is initialized, as well as the positions of the m objects \(l_1, \ldots, l_m \in \mathbb{R}^2\) in the environment. Recall that \(m\) is estimated by CLEAR; see Section 5.3.

Following the standard formulation of landmark SLAM, we consider two types of measurements during optimization. First, we use odometry measurements \(z'_{i+1} \in \text{SE}(2)\) extracted from EKF, which links consecutive submaps belonging to the same vehicle. For two submaps from different vehicles, an odometry measurement is not available as these vehicles do not share a common coordinate system.
Second, we also consider observation measurements which links a map origin to landmarks observed in this submap. These measurements can be extracted from the global data association \( \pi^c_{su}(i) \): for each object \( \sigma^c_i \) in map \( M^c \), we retrieve the index \( c \) of the corresponding tree via \( c = \pi^c_{su}(i) \). An observation is then created between \( x_i \) and \( l_i \), with a relative transformation constraint specified by \( p^s_u(\cdot) \in \mathbb{R}^2 \), where \( p^s_u(\cdot) \) is the observed position of this tree in the coordinate frame of map \( M^c \).

After all variables and measurements are initialized, we use iSAM (Kaess et al., 2008) to carry out the nonlinear optimization. Finally, the optimized map origins \( \{x_i\} \) are transmitted back to the vehicles to correct estimation drifts during local estimation, by realigning all onboard submaps using the updated estimates of \( \{x_i\} \).

6. Simulation experiments

In order to evaluate the performance of the proposed tree detection, exploration, and SLAM algorithms over many trials, we conducted experiments in simulations and used largely the same set of parameters as in the following real-world experiments. Certain parameters were different to compensate for the differences in the simulation and the real world, e.g., the limit on maximum acceleration was set much higher in real-world experiments to compensate for unmodeled external forces such as wind.

6.1. Simulation setup

To evaluate tree detection under different settings of sensor noise and forest density (Section 6.2), we generated random forests in 2D where tree positions are sampled from a 2D Poisson point process (Chiu et al., 2013) and tree radii are sampled uniformly from 0.1–0.3 m. We then simulated laser range measurements corrupted by zero-mean Gaussian noise for a robot with a 30 m sensing range, a 270° field of view, and a 0.25° angular resolution (same as real-world experiments).

To evaluate the proposed exploration and SLAM algorithms (Sections 6.4–6.5), we also leveraged a high-fidelity simulation environment based on the Unity game engine. Vehicle dynamics and IMU readings for a simulated vehicle were generated using the Drake toolbox (Tedrake, 2014) and a quadrotor model as described in (Mellinger et al., 2012). To test integration with the full stack, we also utilized the Pixhawk (Meier et al., 2012) software-in-the-loop (SITL) to simulate the motor commands from the Pixhawk, which were fed back into the Drake dynamics model. We used the Unity game engine to simulate 2D laser scans in a randomly generated forest environment at roughly 30 Hz with a 270° field of view and 30 m range. All sensor measurements were simulated with low noise. The simulated setpoints generated by the motion planner were passed to the Pixhawk SITL.

6.2. Tree-detection results in simulation

Figure 5(a) shows an example random forest in two dimensions together with the tree trunks detected by our algorithm. Using the simulator, we evaluated the precision of tree detection with respect to increasing sensor noise and forest density (i.e., average number of trees per unit area). For each level of noise and density, we performed 500 random simulations and recorded the average precision. A tree detection is classified as a true positive if the estimated position is within 0.5 m of ground truth and the error of estimated radius is less than 30%. As shown in Figure 5(b), the precision of tree detection tends to be more sensitive under low noise and less so under higher noise. On the other hand, precision also degrades as the forest density increases, as shown in Figure 5(c). We observed that many failure cases in dense forests are caused by the Dirichlet process clustering algorithm incorrectly merging points belonging to nearby trees. In these cases, more sophisticated tree-detection methods are required to further improve robustness.

6.3. Loop closure candidates detection results in simulation

Using the same simulator as a previous section, we evaluated the effectiveness of GLARE and GLAROT (Section 5.1) at detecting loop closure candidates, under increasing forest density. We recorded the average GLAROT distances for scan pairs that either share sufficient overlap in the fields of view (and hence likely to be loop closures), or do not share any overlap in the fields of view. In Figure 6, we refer to the former as “true pairs” and the latter as “false pairs”. Intuitively, the larger the gap is between the GLAROT distances of true pairs and false pairs, the more effective GLAROT is at identifying loop closures candidates. In our experiments, we used the default discretization of \( N_\rho = 120, N_\theta = 12 \), and a reduced Gaussian blur with standard deviation 0.01. We observed that moderate change in discretization does not lead to significant change in performance.

As expected, in Figure 6, the GLAROT distances for true pairs (i.e., loop closure candidates) is consistently lower than false pairs, with a reasonable margin under all levels of forest density. In particular, the margin between the two curves is larger in sparser forests, indicating that GLAROT is more effective in this case. This is because as the forest gets denser, occlusions happen more frequently, effectively reducing the sensor’s field of view. We note that this issue can be mitigated by aggregating information from multiple scans, which is already implemented in our current system via the use of submaps.

6.4. Exploration results in simulation

We evaluated the proposed frontier-based exploration planner in simulation. A single UAV was tasked with covering
Fig. 5. (a) Example simulation of a 2D random forest. The robot is placed at the origin, with a 270° field of view that does not cover the fourth quadrant. Black circles denote ground-truth tree trunks. Blue dots show simulated laser returns. Magenta circles show detected tree trunks. (b) Precision of tree detection with increasing sensor noise, under fixed forest density of 0.2/m². (c) Precision of tree detection under increasing forest density, with sensor noise standard deviation fixed at 0.05 m. For each level of noise and density, we performed 500 random simulations and recorded the average precision.

Fig. 6. Average GLAROT distances for simulated scan pairs that either have sufficient overlap in the fields of view (true pair) or not (false pair), as a function of increasing forest density. Each data point is the result of averaging over 100 random simulations. The margin between the GLAROT distances is larger in sparser forests, indicating that GLAROT is more effective at identifying loop closure candidates in this case. We note that results under density level larger than 0.6/m² is omitted, as the margin between the two curves stays relatively constant.

Table 1. Comparison with baseline planner in three different search areas in simulation. Each planner was evaluated multiple times inside each area and average completion time and flight speed were recorded. In all three cases, the proposed planner was able to maintain a higher velocity while covering the search area more efficiently, resulting in a much shorter average exploration time compared with the baseline planner.

| Area | Planner | Duration (s) | Average speed (m/s) |
|------|---------|--------------|---------------------|
| 1    | Proposed | 313.50       | 0.84                |
| 1    | Baseline | 504.87       | 0.69                |
| 2    | Proposed | 340.52       | 0.82                |
| 2    | Baseline | 448.91       | 0.71                |
| 3    | Proposed | 302.41       | 0.80                |
| 3    | Baseline | 477.96       | 0.72                |

Takes for the vehicle to complete the exploration mission. As demonstrated by Table 1, the proposed planner clearly outperformed the baseline planner in terms of both the total time to complete the mission and the average speed during flight. Figure 7 shows the trajectory of the proposed planner (a)–(d) and the baseline planner (e)–(h) in an example search area. As expected, the proposed planner produced a much smoother overall trajectory compared with the baseline and completed the mission in a shorter period of time.

6.5. SLAM results in simulation

We evaluated the capability of our submap-based map representation (Section 4.1) to correct onboard estimation drifts. In practice, such capability is needed to make sure that the final map produced by the ground station is as accurate as possible. Similar to the previous section, the UAV
was tasked with exploring a 20 m × 20 m search area in simulation. The vehicle initialized a new submap every 5 seconds.

Figure 8(a) shows the ground-truth trajectory (yellow) and the trajectory estimated by EKF (blue). As expected, the EKF estimates suffered from accumulated estimation drifts. To correct the drifts, we applied our SLAM pipeline described in Section 5 to optimize the origins of individual submaps, and refined the estimated trajectory based on the optimized submap origins. The resulting trajectory (red) mostly matched the ground truth, confirming the accuracy of the SLAM solution.

To provide additional quantitative results, we also use the absolute trajectory error (ATE) (Sturm et al., 2012), defined as the average distance between ground-truth poses and estimated poses, as the performance critical metric to evaluate our system. The results are shown in Figure 8(b). Owing to estimation drifts, the ATE associated to EKF eventually exceeded 0.25 m; in contrast, performing SLAM effectively bounded the ATE below 0.03 m.

7. Flight experiments

Real flight tests were performed in the forest at NASA LaRC, as shown in Figure 1. To test the proposed real-time planning and CSLAM algorithms, we deployed a team of two quadroters to perform the multi-agent mapping and coverage task. We report qualitative planning and CSLAM performance, as well as analysis of data payload size and algorithm runtimes (Section 7.2). We note that the reported real-time planning results were the same as in our previous work (Tian et al., 2018), with the frontier-based planning occurring in a single onboard map. Nevertheless, all CSLAM results were updated to incorporate the improvements (notably the submap-based representation and cycle consistency multiway matching) made in this work. In addition, we performed extensive offline analysis on the global data association methods described in this work using the data collected by a single vehicle from the same experiment (Section 7.3).

7.1. Outdoor flight setup

Each vehicle was a modified DJI F450 carrying a horizontally mounted Hokuyo UTM-30LX laser rangefinder, a Pixhawk PX4 unit providing inertial measurements and motor commands, a downward-facing LidarLite for altitude measurements, and an Intel NUC computer for onboard computation. The Hokuyo produced laser measurements at a rate of 40 Hz over an angular field of view of 270° with 0.25° angular resolution. The inertial measurements and Lidar-Lite measurements were processed at 100 Hz. The vehicles
were commanded to fly at 1.8 m altitude with a maximum velocity of 2.0 m/s, and a maximum acceleration of 0.4 m/s². Communication with the ground station was maintained via 5.8 GHz WiFi; alternatives for peer-to-peer (P2P) communication, e.g., long-range (LoRa) radios, exist for larger search regions. A new submap was initialized on each vehicle after 5 seconds. Other parameters used during onboard and offboard operations were: $\lambda = 0.5$ (Section 4.2), $\tau_{\text{cull}} = 3$ (Section 4.3), $\epsilon_{\text{GLAROT}} = 1.5$ (Section 5.1), $\epsilon_{\text{CG}} = 0.15$ (Section 5.2), and $\tau_{\text{CG}} = 7$ (Section 5.2).

### 7.2. Real-time planning and CSLAM results

In the outdoor experiments, our vehicles were started at different locations with unknown relative positions. The mission was specified by non-overlapping search area of size 17 m × 20 m for each vehicle. Each vehicle was tasked with observing the entire search region. Each search area was initially unknown; as the vehicles individually completed the coverage task, they established inter-trajectory loop closures based on similar configurations of trees observed during flight. The individual submaps collected during the experiment from both vehicles were then fused real-time on the ground station. The progress of the planners throughout the flight is shown in Figure 9 and matching onboard camera images are shown in Figure 10.

Since the proposed planner preferred frontiers with smaller orientation change, we observed that the resulting flight trajectory exhibited bias towards a natural spiral pattern. Vehicle 1 completed the exploration mission after 122 seconds, with an average flight speed of 2.66 m/s; vehicle 2 completed the mission after 135 seconds, with an average flight speed of 2.65 m/s. Note that owing to measurement noise, the calculated average speeds were higher than the maximum allowed speed (2.0 m/s) even after applying a low-pass filter to the raw velocity measurements.

Figure 11 shows the evolution of the global object (tree) map on the ground station. For cleaner visualization, we only show tree landmarks that have been observed in at least three submaps. At $t = 30$ s (Figure 11(a)), no inter-trajectory loop closure was detected and the two vehicle trajectories were arbitrarily aligned. The first inter-trajectory loop closure was established at $t = 45$ s and map fusion happened subsequently (Figure 11(b)). Afterwards, all newly received submaps from the two vehicles were consistently fused into the global map (Figure 11(c)–(d)). In addition, we also rendered the fused global 3D occupancy grid at $t = 120$ s, by aligning all submaps in the world frame using their optimized origins in Octomap (Hornung et al., 2013). The resulting global map is shown in Figure 12. Note that, in practice, constructing the global occupancy grid would require vehicles to transmit their full local occupancy grids which could be communication intensive. In contrast, our tree-based map representation shown in Figure 11 captures all the essential geometric information in each submap and is still lightweight enough for low-bandwidth communication.

In Figure 11(d), while we successfully fused multiple submaps into a consistent global map, the fused map nevertheless contains duplicate trees. For example, some overlapping magenta circles correspond to a single tree in the actual forest. This issue exemplifies the trade-off between
The International Journal of Robotics Research 39(10-11)

Fig. 9. Vehicle trajectories and partial maps in real flight tests at different time steps (seconds): (a) $t = 30$ s; (b) $t = 45$ s; (c) $t = 90$ s; (d) $t = 120$ s; (e) $t = 30$ s; (f) $t = 45$ s; (g) $t = 90$ s; (h) $t = 120$ s; (a)–(d) show the trajectory of the first vehicle; (e)–(h) show the second vehicle. Since the proposed planner preferred frontiers with smaller orientation change, the flight trajectory exhibited bias towards a natural spiral pattern.

**Fig. 9.** Vehicle trajectories and partial maps in real flight tests at different time steps (seconds): (a) $t = 30$ s; (b) $t = 45$ s; (c) $t = 90$ s; (d) $t = 120$ s; (e) $t = 30$ s; (f) $t = 45$ s; (g) $t = 90$ s; (h) $t = 120$ s; (a)–(d) show the trajectory of the first vehicle; (e)–(h) show the second vehicle. Since the proposed planner preferred frontiers with smaller orientation change, the flight trajectory exhibited bias towards a natural spiral pattern.

**precision** and **recall** as two performance critical metrics for data association. In our system, we use conservative thresholds when matching trees within and across submaps. While this approach allows us to achieve high precision, it also results in degrading recall (i.e., missing certain associations), which ultimately adds duplicate trees to the map. However, we note that although missing associations is undesirable, it does not impact the geometric consistency of inferred loop closures. This is because to infer the relative transformation between two submaps in two dimensions, at minimum one only needs to identify two correct landmark associations. Therefore, although our algorithm sometimes failed to associate **all** trees between two submaps, it still associated enough of them to correctly infer the underlying relative transformation, which is the most critical factor for CSLAM and map fusion. This can be done by further optimizing upstream modules (e.g., tree detection and pairwise association), or by performing an additional step to fuse overlapping landmarks after CSLAM, which we leave to future work.

To further demonstrate the efficiency of the proposed CSLAM framework, we also evaluated the incurred runtime and data payloads. Figure 13 shows the total runtime of GLAROT (Section 5.1), CG matching (Section 5.2), CLEAR (Section 5.3), and iSAM (Section 5.4), each as a function of the number of submaps received at the ground station. We observed that all modules demonstrated satisfactory speeds; in particular, GLAROT and iSAM were computationally very efficient and, hence, capable of running in real-time. In comparison, the pairwise and multiway data association steps (using CG matching and CLEAR, respectively) incurred higher computation costs. In practice, however, CG matching can be made real-time by storing and reusing previously computed matches. On the other hand, the CLEAR algorithm could also be accelerated by implementing a block SVD method that exploits the separable structure of the underlying data association graph (Fathian et al., 2019). Still, as the number of submaps grows, we note that multiway association is likely to become the computation bottleneck of the overall system. Thus, reducing the computational cost of this step, via an incremental algorithm that can reuse previously computed results, remains an important avenue for future work. In principle, CG matching has exponential complexity owing to the need to solve a maximum clique problem. In our specific application, however, our use of object-level representations and aggressive culling drastically reduce the input size to CG matching (each submap typically contains less than 30 tree landmarks). As a result, in our outdoor experiment each matching requires at most 0.015 s, making it possible to process incoming submaps in real-time. On the other hand, the runtime of CLEAR eventually exceeded 0.5 s as
Fig. 10. Images from onboard GoPro camera at different time steps (seconds): (a) $t = 30$ s; (b) $t = 45$ s; (c) $t = 90$ s; (d) $t = 120$ s; (e) $t = 30$ s; (f) $t = 45$ s; (g) $t = 90$ s; (h) $t = 120$ s; (a)–(d) show images from the first vehicle; (e)–(h) show images from the second vehicle. Frames are approximately aligned with the vehicle trajectories shown in Figure 9.
more submaps are added. We currently use CLEAR as a batch technique to periodically optimize global data association. An important avenue for future research is the development of incremental methods that is able to reuse results computed previously.

Finally, Figure 14 shows the total data payloads received at the ground station as a function of the number of submaps. Our calculation did not include additional communication incurred by network protocol overhead. Note that the growth in data payloads was not exactly linear with respect to the number of submaps. This is because each submap contained different number of trees depending on the specific area it covered. Overall, the proposed tree-based map representation incurred minimal communication (at most 8 kB with 48 submaps) demonstrating its usefulness for low-bandwidth and real-time communication during search and rescue.

7.3. Data association results

Post-flight, we performed additional offline evaluations of the proposed pairwise and multiway data association algorithms (Sections 5.2–5.3) using data collected by vehicle 2 (yellow trajectory in Figure 12). For this experiment, GLAROT (Section 5.1) was disabled and loop closure was attempted on every pair of submaps received at the ground station. To benchmark our methods, we also implemented the approach of Giamou et al. (2017), which uses stabilized FLIRT features obtained from DP-means clustering. Ground-truth (e.g., GPS) data was not available in our forest datasets; instead, we used estimates provided by onboard EKF to verify proposed pairwise associations. Specifically, a pair of matched trees was declared to be a true match if their positions as estimated by EKF in the world frame were sufficiently close. For short-duration flights, we expect this metric to be relatively accurate. For longer flights, the accuracy of this metric will degrade as drifts accumulate in the EKF estimate. A better evaluation method should utilize more accurate source of information (e.g., GPS) or uncertainty-aware metrics (e.g., Mahalanobis distance), which we leave to future work.

Figure 15 shows the resulting precision and unnormalized recall (number of correct pairwise associations), which are two commonly used performance metrics when evaluating the quality of data association. For each submap, an equal number of stabilized FLIRT features (Giamou
et al., 2017) and tree landmarks were extracted. CG matching (Section 5.2) with varying tolerance threshold \( \epsilon_{\text{CG}} \) were performed to obtain the resulting data points for FLIR T and trees. For trees, we further optimized the initial pairwise associations with the CLEAR algorithm (Section 5.3). In the forest, the proposed tree-detection algorithm unsurprisingly outperformed stabilized FLIR T features. Furthermore, CLEAR was able to improve both precision and recall by simultaneously rejecting false associations and suggesting more correct associations. This result demonstrates the usefulness of cycle consistent multiway association as an additional step for CSLAM.

To provide more insights on the benefits of cycle consistent multiway matching, Figure 16 shows part of the data association graph before and after optimizing with CLEAR. Each triangle represents a tree colored according to the submap it belongs to. Two trees from different submaps are connected by an edge if they are matched in the given data association. For visual purpose, all submaps are aligned in the world frame, i.e., each object is drawn at the corresponding position estimated by EKF. Thus, objects that correspond to the same tree in the forest should appear close to each other. Owing to perceptual aliasing, the initial associations from CG matching contained several wrong correspondences, shown as the long edges in Figure 16(a). By jointly optimizing over all pairwise matches, CLEAR was able to reject these outliers as shown in Figure 16(b). Furthermore, the resulting data association is guaranteed to be cycle consistent. This means that the graph shown in Figure 16(b) only contains disjoint cliques which makes subsequent map fusion possible; see Section 5.3.

8. Conclusions and future work

We have presented a collaborative exploration and mapping system for multi-UAV search and rescue under the forest canopy. Our UAVs were equipped with onboard autonomy that reliably performed sensing, pose estimation, local mapping, and exploration planning. CSLAM was performed at a central ground station. To cope with the limited communication bandwidth, we used a map compression scheme that compressed onboard dense occupancy grids into sparse tree-based maps before transmission. During CSLAM, we used the relative positions and orientations between multiple trees as a unique signature to detect loop closures. To
be further improved. In this work, we have not explicitly addressed the issue of outliers during data association and loop closure detection. In our outdoor experiment, we rarely observed outlier loop closures due to the conservative thresholds used in CG matching. Nonetheless, under higher noise regimes (e.g., in denser forests with more branches and leaves), errors during data association and loop closure are more likely to happen. In these scenarios, one needs to implement outlier rejection schemes for both intratrijectory and inter-trajectory loop closures. As an example, the robust map fusion procedure of Mangelson et al. (2018) can be directly applied on top of our current system.

On the planning side, currently each vehicle performs frontier-based exploration within its own search region. While the dynamics-aware frontier selection was able to produce smoother and more time-efficient search behavior, we also observed that its inherent bias to follow the current velocity vector sometimes results in the vehicle flying past a small patch of unobserved space, and having to return later in the mission. Possible ways of addressing this problem include encouraging the planner to prioritize small patches of unobserved space, or incorporating longer-horizon planning during exploration. In addition, currently each of our vehicles performs independent exploration within its assigned region, without much interaction with others (except during CSLAM). Since the search regions do not overlap with each other, explicit coordination is not required to complete the overall exploration task. Nevertheless, our system could still benefit from higher-level coordination among the robots to be more flexible at, e.g., assigning vehicles to different search areas at different times.

Finally, most procedures in our system currently take place in the projected 2D map. In our experiments, we observed that the 2D assumption is generally sufficient for sparse forests. Nevertheless, such assumption could become restrictive in denser environments, for example when feasible 2D paths no longer exist. We note that most modules in our system can be extended to three dimensions at the expense of increased computation costs, without requiring many modifications. Such an extension will also be considered for future work.

Acknowledgements
The authors gratefully acknowledge Matthew Giamou for early works on this project, and Chester Dolph, Brian Duvall, and Zak Johns for assistance during outdoor experiments. The authors also would like to thank Kasra Khosoussi and Kaveh Fathian for insightful discussions during the preparation of this manuscript, as well as two anonymous reviewers for their constructive feedback that further improved this work.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This
work was supported in part by the NASA Convergent Aeronautics Solutions project Design Environment for Novel Vertical Lift Vehicles (DELIVER), the ONR (BRC award N000141712072), and ARL DCIST (Cooperative Agreement Number W911NF-17-2-0181).

Notes
1. Alternatively, one can initialize a new submap after the vehicle travels a fixed distance or after the estimation uncertainty of EKF grows beyond a threshold.
2. Here $\mathcal{G}$ should be distinguished from the CG defined in the previous section for pairwise association.
3. We did not report CSLAM results in simulation, because our simulation currently only supports single vehicle exploration. Nevertheless, when evaluating estimation accuracy, the number of vehicles is nearly irrelevant because we can always assume the trajectories are produced by a single vehicle. In addition, see Section 7.2 for real-world CSLAM results with two UAVs.
4. We do not show recall as the total number of correct pairwise associations in this dataset is unknown.

**ORCID iD**

Yulun Tian [http://orcid.org/0000-0002-9017-3022](http://orcid.org/0000-0002-9017-3022)

**References**

Aragues R, Montijano E and Sagues C (2011) Consistent data association in multi-robot systems with limited communications. In: Robotics: Science and Systems (RSS).

Arandjelović R, Gronat P, Tori A, Pajdla T and Sivic J (2016) NeTVLAD: CNN architecture for weakly supervised place recognition. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR).

Asmar DC, Zelek JS and Abdallah SM (2006) Tree trunks as landmarks for outdoor vision SLAM. In: 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPR’06).

Bachrach A, Prentice S, He R and Roy N (2011) RANGE–robust autonomous navigation in GPS-denied environments. *Journal of Field Robotics* 28(5): 644–666.

Bailey T, Nebot EM, Rosenblatt J and Durrant-Whyte HF (2000) Data association for mobile robot navigation: A graph theoretic approach. In: Proceedings of the International Conference on Robotics and Automation (ICRA).

Bay H, Tuytelaars T and Van Gool L (2006) SURF: Speeded up robust features. In: Proceedings of the European Conference on Computer Vision (ECCV).

Bircher A, Kamel M, Alexis K, Oleynikova H and Siegwart R (2016) Receding horizon “next-best-view” planner for 3D exploration. In: Proceedings of the International Conference on Robotics and Automation (ICRA).

Bosse M and Zlot R (2009) Keypoint design and evaluation for place recognition in 2D LIDAR maps. In: Robotics and Autonomous Systems (RAS).

Bourgault F, Makarenko A, Williams S, Grocholsky B and Durrant-Whyte H (2002) Information based adaptive robotic exploration. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS).

Burgard W, Moors M, Fox D, Simmons R and Thrun S (2000) Collaborative multi-robot exploration. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).

Burgard W, Moors M, Stachniss C and Schneider FE (2005) Coordinated multi-robot exploration. *IEEE Transactions on Robotics* 21(3): 376–386.

Calonder M, Lepetit V, Strecha C and Fua P (2010) BRIEF: Binary robust independent elementary features. In: Daniilidis K, Maragos P and Paragios N (eds.) Proceedings of the European Conference on Computer Vision (ECCV).

Campbell T, Liu M, Kulis B, How JP and Carlin L (2013) Dynamic clustering via asymptotics of the dependent Dirichlet process mixture. In: Advances in Neural Information Processing Systems, vol. 26.

Cesare K, Skeele R, Yoo SH, Zhang Y and Hollinger G (2015) Multi-UAV exploration with limited communication and battery. In: Proceedings of the International Conference on Robotics and Automation (ICRA).

Charrow B, Liu S, Kumar V and Michael N (2015) Information-theoretic mapping using Cauchy–Schwarz quadratic mutual information. In: Proceedings of the International Conference on Robotics and Automation (ICRA).

Chen Y, Guibas L and Huang Q (2014) Near-optimal joint object matching via convex relaxation. In: Proceedings of the International Conference on Machine Learning (ICML).

Chernov N (2010) *Circular and Linear Regression: Fitting Circles and Lines by Least Squares* (CRC Monographs on Statistics & Applied Probability). Chapman & Hall.

Chiu S, Stoyan D, Kendall W and Mecke J (2013) *Stochastic Geometry and Its Applications (Wiley Series in Probability and Statistics)*. New York: John Wiley & Sons, Inc.

Choudhary S, Carlone L, Nieto C, Rogers J, Christensen HI and Dellaert F (2017) Distributed mapping with privacy and communication constraints: Lightweight algorithms and object-based models. *The International Journal of Robotics Research* 36(12): 1286–1311.

Cieslewski T, Choudhary S and Scaramuzza D (2018) Data-efficient decentralized visual SLAM. In: Proceedings of the International Conference on Robotics and Automation (ICRA).

Cieslewski T, Kaufmann E and Scaramuzza D (2017) Rapid exploration with multi-rotors: A frontier selection method for high speed flight. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS).

Cummins M and Newman P (2011) Appearance-only SLAM at large scale with FAB-MAP 2.0. *The International Journal of Robotics Research* 30(9): 1100–1123.

Cunningham A, Indelman V and Dellaert F (2013) DDF-SAM 2.0: Consistent distributed smoothing and mapping. In: Proceedings of the International Conference on Robotics and Automation (ICRA).

Cunningham A, Wurm KM, Burgard W and Dellaert F (2012) Fully distributed scalable smoothing and mapping with robust multi-robot data association. In: Proceedings of the International Conference on Robotics and Automation (ICRA).

Dong J, Nelson E, Indelman V, Michael N and Dellaert F (2015) Distributed real-time cooperative localization and mapping using an uncertainty-aware expectation maximization approach. In: Proceedings of the International Conference on Robotics and Automation (ICRA).

Dubé R, Cramariuc A, Dugas D, Nieto J, Siegwart R and Cadena C (2018) SegMap: 3D segment mapping using data-driven descriptors. In: Robotics: Science and Systems (RSS).
Fathian K, Khosoussi K, Lusk P, Tian Y and How JP (2019) CLEAR: A consistent lifting, embedding, and alignment rectification algorithm for multi-agent data association. *arXiv Preprint arXiv:1902.02256*.

Feder HJS, Leonard JJ and Smith CM (1999) Adaptive mobile robot navigation and mapping. *The International Journal of Robotics Research* 18(7): 650–668.

Fischler MA and Bolles RC (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24(6): 381–395.

Forster C, Ko J, Konolige K, Limketkai B, Schulz D and Stewart B (2006) Distributed multirobot exploration and mapping. *Proceedings of the IEEE* 94(7): 1325–1339.

Gálvez-López D and Tarados JD (2012) Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics* 28(5): 1188–1197.

Giamou M, Babich Y, Habibi G and How J (2017) Stable laser interest point selection for place recognition in a forest. In: *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*.

Gil A, Óscar Reinoso, Ballesta M and Juliá M (2010) Multirobot visual SLAM using a Rao-Blackwellized particle filter. In: *Robotics and Autonomous Systems (RAS)*.

González-Baños HH and Latombe JC (2002) Navigation strategies for exploring indoor environments. *The International Journal of Robotics Research* 21(10–11): 829–848.

Hart PE, Nilsson NJ and Raphael B (1968) A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* 4(2): 100–107.

Heng L, Gotovos A, Krause A and Pollefeys M (2015) Efficient visual exploration and coverage with a micro aerial vehicle in unknown environments. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*.

Himstedt M, Frost J, Hellbach S, Böhme HJ and Maehle E (2013) Distributed monocular SLAM with multiple micro aerial vehicles. In: *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*.

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

Huang QX and Guibas L (2013) Consistent shape maps via semidefinite programming. In: *Proceedings of the Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*.

Indelman V, Nelson E, Michael N and Dellaert F (2014) Multirobot pose graph localization and data association from unknown initial relative poses via expectation maximization. In: *Proceedings of the International Conference on Robotics and Automation (ICRA)*.

Jégou H, Douze M, Schmid C and Pérez P (2010) Aggregating local descriptors into a compact image representation. In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jégou H, Perronnin F, Douze M, Sánchez J, Pérez P and Schmid C (2012) Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(9): 1704–1716.

Kaess M and Dellaert F (2009) Covariance recovery from a square root information matrix for data association. In: *Robotics and Autonomous Systems (RAS)*.

Kaess M, Ranganathan A and Dellaert F (2008) ISAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics* 24(6): 1365–1378.

Kallas F and Rizzini DL (2016) Efficient loop closure based on FALKO LIDAR features for online robot localization and mapping. In: *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*.

König J and Janežič D (2007) An improved branch and bound algorithm for the maximum clique problem. MATCH - Communications in Mathematical and in Computer Chemistry 58: 569–590.

Kukko A, Kaijaluo R, Kaartinen H, Lehtola VV, Jaakkola A and Hyypä J (2017) Graph SLAM correction for single scanner MLS forest data under boreal forest canopy. *ISPRS Journal of Photogrammetry and Remote Sensing* 132: 199–209.

Leonard J and Feder HJ (2001) Decoupled stochastic mapping. *IEEE Journal of Oceanic Engineering* 26(4): 561–571.

Leonardos S, Zhou X and Daniilidis K (2017) Distributed consistent data association via permutation synchronization. In: *Proceedings of the International Conference on Robotics and Automation (ICRA)*.

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60: 91–110.

Mangelson J, Dominic D, Eustice R and Vasudevan R (2018) Pairwise consistent measurement set maximization for robust multi-robot map merging. In: *Proceedings of the International Conference on Robotics and Automation (ICRA)*.

Meier L, Tanskanen P, Heng L, Lee GH, Fraundorfer F and Pollefeys M (2012) Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision. *Autonomous Robots* 33(1–2): 21–39.

Mellinger D, Michael N and Kumar V (2012) Trajectory generation and control for precise aggressive maneuvers with quadrotors. *The International Journal of Robotics Research* 31(5): 664–674.

Montijano E, Aragues R and Sagás C (2013) Distributed data association in robotic networks with cameras and limited communications. *IEEE Transactions on Robotics* 29(6): 1408–1423.

Morrison JG, Gavez-Lopez D and Sibley G (2016) Scalable multirobot localization and mapping with relative maps: Introducing MOARSLAM. *IEEE Control Systems Magazine* 36(2): 75–85.

Mur-Artal R and Tardós JD (2017) ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics* 33(5): 1255–1262.

Neira J and Tardos JD (2001) Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation* 17(6): 890–897.

Nguyen A, Ben-Chen M, Welnicka K, Ye Y and Guibas L (2011) An optimization approach to improving collections of shape maps. *Computer Graphics Forum* 30(5): 1481–1491.

Öhman M, Miettinen M, Kannas K, Jutila J, Visala A and Forsman P (2008) Tree measurement and simultaneous localization and mapping system for forest harvesters. In: *Field and Ser-
vice Robotics (Springer Tracts in Advanced Robotics, Vol. 42). New York: Springer, pp. 369–378.
Pachauri D, Kondor R and Singh V (2013) Solving the multi-way matching problem by permutation synchronization. In: Proceedings of the International Conference on Neural Information Processing Systems (NIPS).
Paul R and Newman P (2010) FAB-MAP 3D: Topological mapping with spatial and visual appearance. In: Proceedings of the International Conference on Robotics and Automation (ICRA).
Paz LM, Pinies P, Neira J and Tardos JD (2005) Global localization in SLAM in bilinear time. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS).
Perronnin F, Liu Y, Sánchez J and Poirier H (2010) Large-scale image retrieval with compressed Fisher vectors. In: Proceedings of the conference on Computer Vision and Pattern Recognition (CVPR).
Rublee E, Rabaud V, Konolige K and Bradski G (2011) ORB: An efficient alternative to SIFT or SURF. In: Proceedings of the International Conference on Computer Vision (ICCV).
Rusu RB, Blodow N and Beetz M (2009) Fast point feature histograms (FPFH) for 3D registration. In: Proceedings of the International Conference on Robotics and Automation (ICRA).
Saeedi S, Trentini M, Seto M and Li H (2016) Multiple-robot simultaneous localization and mapping: A review. Journal of Field Robotics 33(1): 3–46.
Schmuck P and Chli M (2018) CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams. Journal of Field Robotics 36(4): 763–781.
Schuster MJ, Schmid K, Brand C and Beetz M (2019) Distributed stereo vision-based 6D localization and mapping for multi-robot teams. Journal of Field Robotics 36(2): 305–332.
Shen S, Michael N and Kumar V (2012) Autonomous indoor 3D exploration with a micro-aerial vehicle. In: 2012 IEEE International Conference on Robotics and Automation.
Stachniss C, Mozos O and Burgard W (2008) Efficient exploration of unknown indoor environments using a team of mobile robots. Annals of Mathematics and Artificial Intelligence 52: 205–227.
Sturm J, Engelhard N, Endres F, Burgard W and Cremers D (2012) A benchmark for the evaluation of RGB-d SLAM systems. In: International Conference on Intelligent Robots and Systems.
Tang J, Chen Y, Kukko A, et al. (2015) SLAM-aided stem mapping for forest inventory with small-footprint mobile LIDAR. Forests 6(12): 4390.
Taubin G (1991) Estimation of planar curves, surfaces, and non-planar space curves defined by implicit equations with applications to edge and range image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 13(11): 1115–1138.
Tedrake R (2014) Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems. https://drake.mit.edu/
Tian Y, Liu K, Ok K, et al. (2018) Search and rescue under the forest canopy using multiple UAS. In: Proceedings of the International Symposium on Experimental Robotics (ISER).
Tipaldi GD and Arras KO (2010) FLIRT - interest regions for 2D range data. In: Proceedings of ICRA, pp. 3616–3622.
Tron R, Zhou X, Esteves C and Daniilidis K (2017) Fast multi-image matching via density-based clustering. In: IEEE International Conference on Computer Vision (ICCV).
Yamauchi B, Schultz A and Adams W (1998) Mobile robot exploration and map building with continuous localization. In: Proceedings of the International Conference on Robotics and Automation (ICRA).
Yoder L and Scherer S (2016) Autonomous Exploration for Infrastructure Modeling with a Micro Aerial Vehicle. New York: Springer.
Zhou X, Zhu M and Daniilidis K (2015) Multi-image matching via fast alternating minimization. In: Proceedings of the International Conference on Computer Vision (ICCV).
Zhou XS and Roumeliotis SI (2006) Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS).