

Search and Rescue under the Forest Canopy using Multiple UAS

Journal Title
XX(X):1–14
©The Author(s) 2019
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/

SAGE

Yulun Tian¹, Katherine Liu¹, Kyel Ok¹, Loc Tran², Danette Allen², Nicholas Roy¹ and Jonathan P. How¹

Abstract

We consider the problem of multi-robot search and rescue under the forest canopy. Forest is a particularly challenging environment for collaborative mapping and exploration, mainly due to the existence of severe perceptual aliasing which hinders reliable mutual localization and map fusion. Our proposed system features unmanned aerial vehicles (UAVs) with onboard sensing and autonomy. Each UAV runs a lightweight filtering algorithm for local state estimation, and a dynamic-aware frontier selection algorithm for fast exploration. The essential and computationally intensive task of collaborative simultaneous localization and mapping (CSLAM) is performed at a central ground station. To handle perceptual aliasing, we make use of stable landmarks extracted from trees which significantly improve precision and recall during place recognition. Furthermore, to recover from incorrect pairwise data associations during loop closure, we propose a novel procedure for global data association based on recently developed techniques on *cycle consistent* multiway matching. Our algorithm returns a global data association that is guaranteed to be cycle consistent, and is shown to significantly improve precision compared to the input pairwise associations. The overall multi-UAV system is extensively validated during real-world collaborative exploration missions in a forest at NASA Langley Research Center.

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. Small autonomous unmanned aerial vehicles (UAVs) have been frequently proposed as the next-generation solution for search and rescue missions under the forest canopy, such as shown in Hampton, VA in Figure 1. These vehicles can be rapidly deployed, can collaboratively explore expanses of terrain efficiently, and are agile enough to operate in reasonably thick forests.

Deploying a multi-UAV system for forest search and rescue presents a multitude of technical challenges. First, external localization services such as the Global Positioning System (GPS) are typically unavailable due to the overhead foliage. Peer-to-peer (P2P) communication, on the other hand, is unreliable and intermittent due to signal attenuation and limited transmission range. Consequently, all of sensing, estimation, and planning must happen onboard the vehicle during real-time search and rescue operations, while onboard sensors are affected by self similarities in the forest, i.e., visual aliasing, and lighting changes caused by the movement of the overhead foliage in the wind.

In this work, we present a multi-UAV system featuring drones with *onboard autonomy* that efficiently performs autonomous sensing, vehicle state estimation, local mapping, and exploration planning in a forest environment. The key idea behind our system is in building two different representations of the same environment to overcome the problem of visual aliasing and limited communication: a dense 3D volumetric grid representation for onboard trajectory planning and a compact tree-based representation

for efficient communication to a centralized ground station for inter-vehicle loop-closures and global map building.

The onboard sensing, estimation, and mapping modules are based on a 2D LIDAR sensor, where scans are matched for vehicle pose estimation, collected into a 3D grid for trajectory planning, and compressed into a tree-based map for efficient communication. We adopt a submap-based grid representation, i.e., build a collection of locally segmented grid maps as opposed to a single grid representation, so that the origins of the submaps can be optimized to correct for onboard estimation drifts upon loop-closures. The submap representation, similar to the sliding-window representation, also bounds the area that is updated after incorporating each new scan for computational efficiency, but the difference is that all of the submaps are kept so that the knowledge of all of the areas are kept for the trajectory planner that extends the coverage of a given search area. While we base our onboard planning on the memory intensive but useful occupancy grid representation, we compress each submap to an object-based map of trees by detecting tree trunks in the scans. We communicate the lightweight tree-based submaps to the ground station for multi-vehicle loop-closures and drift correction.

On the ground station, we solve for the global loop-closures between the submaps of the vehicles and build a joint map of the forest combining all of the information from the vehicles.

¹Massachusetts Institute of Technology, Cambridge, MA, USA

²NASA Langley Research Center, Hampton, Virginia, USA

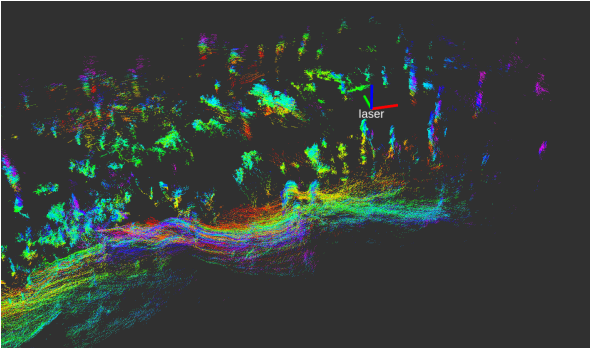
Corresponding author:

Yulun Tian

Email: yulun@mit.edu



(a) Image captured by external camera.



(b) Raw point cloud collected by onboard laser scanner. Individual point is colored according to its height from the ground. Vehicle pose estimated by onboard EKF is shown as the laser frame towards the upper-right.

Figure 1. Autonomous flight of one of our UAVs in a forested area within NASA Langley Research Center (LaRC). A complete video of two UAVs autonomous exploration and mapping is available at: <https://groups.csail.mit.edu/trg/videos/iser2018>.

We overcome the problem of self similarity between the trees by instead of matching based on visual similarity, by matching the relative locations of the tree trunks in the submaps. Our three step data association process first uses GLAROT signatures to quickly find potential loop-closures between submaps, then uses an object-based matching between pairs of submaps to remove inaccurate matches, and finally checks for cycle consistencies in the valid pairs to further reject false loop-closure matches. Based on only the verified loop-closures, the object-level associations are used to optimize over a final object-based map, i.e., a tree-based map of the forest, for the human operator at the ground station.

We demonstrate our system in a small forest at NASA LaRC, where a team of two quad-rotors are each assigned a search region to fully explore. We show that our tree-based map representation and cycle consistency checks result in improved precision in loop-closures. We also show that our dynamics aware exploration planner also outperforms a greedy exploration planner in covering the search area. Our autonomous multi-vehicle system shows promising results toward using team of UAVs for search and rescue in a forest.

The rest of the sections are organized as follows. We first discuss previous work in Section 2 and give an overview of the system in Section 3. We describe the onboard autonomy in Section 4 and the centralized global vehicle registration and map building in Section 5. We present real flight experiments and analyze the results in Section 6 and share insights learned in Section 7.

1.1. Notations and Preliminaries

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) are among the first to propose a centralized collaborative SLAM 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); Cieslewski et al. (2018). While distributed systems have better scalability compared to centralized ones, they do not leverage the opportunity to transfer expensive 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 the ones presented in Forster et al. (2013); Schmuck and Chli (2018).

The problems of search and exploration has been extremely well-studied. Standard approaches include nearest frontier selection (Yamauchi et al. 1998), next-best-view (NBV) (Gonzalez-Baos and Latombe 2002; Bircher et al. 2016), greedy or receding horizon information gain (Feder et al. 1999; Bourgault et al. 2002), or a mixture of these methods (Charrow et al. 2015). More recently, there have been results on exploring to find hidden objects in indoor environments (Aydemir et al. 2011). For rapid exploration using a UAV, Cieslewski et al. (2017) propose to select frontiers that are reachable with minimum change in velocity. Multi-robot exploration has also gained considerable attention over the years. Cesare et al. (2015) demonstrate a frontier-based exploration algorithm for multiple UAVs in an indoor environment. Other related work for multi-robot exploration focuses on deriving algorithms with provable performance guarantees (Singh et al. 2009; Corah and Michael 2017) and improving communication efficiency (Jensen 2018; Corah et al. 2019).

2.2. Multi-Robot Loop Closure and Data Association

Robust loop closure detection and data association form the backbone of any CSLAM system. Existing approaches for multi-robot loop closure and data association can be categorized into *pairwise* and *multiway* techniques.

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 (Neira and Tardos 2001; Zhou and Roullet 2006; Kaess and Dellaert 2009; Gil et al. 2010). 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 RANSAC (Fischler and Bolles 1981) for enhanced robustness.

Recently, *appearance-based* methods have shown great success in large scale place recognition (Paul and Newman 2010; Cummins and Newman 2011) 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 SIFT (Lowe 2004), SURF (Bay et al. 2006), BRIEF (Calonder et al. 2010), and ORB (Rublee et al. 2011). For laser-range data, similar descriptors have been developed which include Gestalt (Bosse and Zlot 2009), FPFH (Rusu et al. 2009), and 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) use objects detected by convolutional neural networks (CNN) during distributed pose graph optimization. Dubé et al. (2018) represent objects with segments in 3D point clouds and use 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; Öhman et al. 2008; Tang et al. 2015; Kukko et al. 2017). The present work also uses trees as landmarks in the map representation. However, unlike prior works that consider passive SLAM with a single robot, we focus on the more general scenario of collaborative SLAM which happens during multi-robot autonomous exploration.

In addition, state-of-the-art SLAM systems often implement a place recognition module which aggregates local features into a global descriptor for fast querying and similarity comparison. 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) develop the Geometric LANDmark RELations (GLARE) signature, which is 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 In robotics, multiway association (i.e., simultaneously considering multiple noisy pairwise

associations) has been proposed in the context of robust data association for pose graph SLAM (Indelman et al. 2014; Dong et al. 2015; Mangelson et al. 2018). A separate line of research closely related to our work addresses the problem of *cycle consistent* multi-robot data association (Aragues et al. 2011; Montijano et al. 2013). Inconsistencies can appear, for example, when a chain of spurious pairwise associations matches two landmarks observed by the same robot. Aragues et al. (2011) propose a heuristic based on cycle detection to detect and resolve these inconsistencies. In the computer vision community, cycle consistency has also gained considerable attention due to popular applications such as multi-shape matching (Nguyen et al. 2011; Huang and Guibas 2013) and multi-image matching (Zhou et al. 2015; Leonardos et al. 2017). 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 (Pachauri et al. 2013; Chen et al. 2014). In this work, we leverage these recent advances for cycle consistent data association for robust collaborative SLAM.

3. Overview

To enable independent exploration of large-scale GPS-denied forest environments, we equip each of our UAVs an onboard autonomy module that performs autonomous sensing, vehicle state estimation, local mapping, and exploration planning; see Figure 2a. 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.

On the offboard ground station, a more computationally capable computer is used to fuse 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, which consists of (i) *global data association*, which solves the underlying registration problem between trees observed across multiple submaps and possibly by different vehicles, and (ii) *landmark-based SLAM*, which jointly optimizes all vehicle poses and tree locations in the global map. See Figure 2b for the corresponding system diagram. Furthermore, the optimized vehicle poses are transmitted back to the vehicles to correct drift during local mapping. We note that the offboard map fusion process can happen in parallel to the onboard autonomous operations of each vehicle. This centralized architecture provides sufficient tolerance against network failures which would only delay potential map fusion, but will not hinder real-time onboard operations. We describe the centralized map fusion process in more detail in Section 5.

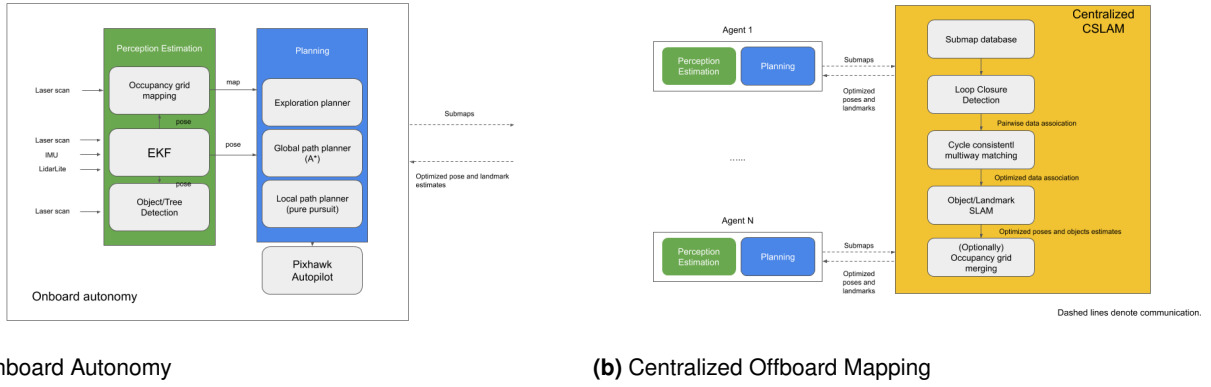


Figure 2. System architecture, **place holder to be optimized.** (a): Each vehicle achieves full autonomy by performing onboard sensing, vehicle state estimation, local mapping, and exploration planning. (b): A computationally capable offboard computer fuse noisy measurements from all vehicles into a globally consistent map of the world.

4. Onboard Autonomy

Each of our vehicles achieve 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 an object-based map compression scheme suitable for low-bandwidth communication (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, self-similarities between trees, and limiting field of views.

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) that provide additional information about the state of the vehicle.

We combine the raw measurements produced by the suite of onboard sensors in a laser-based Extended Kalman Filter (EKF) ([Bachrach et al. 2011](#)) that estimates the 6 degrees-of-freedom pose of the vehicle in the world. 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.

We adopt a *submapping* approach ([Leonard and Feder 2001](#)), in which a new submap is initialized empty on each vehicle after a fixed amount of time.* After receiving a new 2D LIDAR scan, we project the returned points from the local sensor frame to the world frame using real-time pose estimates provided by the EKF; see Figure 1b. The resulting point

clouds are then used to create and update a 3D occupancy grid representation, in our case an Octomap ([Hornung et al. 2013](#)), of the surrounding environment.

Instead of building a single map, we choose to create and update multiple submaps so that we can have a *deformable* map representation onboard the vehicles: while each submap is locally rigid, the relative transformations between the submaps can be optimized using the refined pose estimates after loop closure and CSLAM to correct for the errors accumulated during onboard vehicle estimation. While 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 our onboard motion and exploration planning.

Several state-of-the-art systems, e.g., ([Schmuck and Chli 2018](#)), use a sliding window approach for local mapping in order to bound the onboard memory usage. However, such approach of discarding old information is not suitable for our vehicles since they must keep track of all areas previously visited during exploration. In addition, we note that in practice the memory usage per vehicle is still relatively small since each vehicle is tasked with exploring a reasonably bounded area in the forest.

4.2. Frontier-based Planning

For the search and rescue problem, we adopt a frontier-based exploration algorithm, where a *frontier* refers to a location in the current map that is on the boundary of known and unknown space. Our choice is based on the low computational cost of frontier-based algorithms compared to algorithms based on maximizing information gain. After the frontier-based planner selects the best frontier to visit next, 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,

* Alternatively, the vehicle can initialize a new submap after traveling a fixed distance or after the estimation uncertainty of EKF grows beyond a threshold.

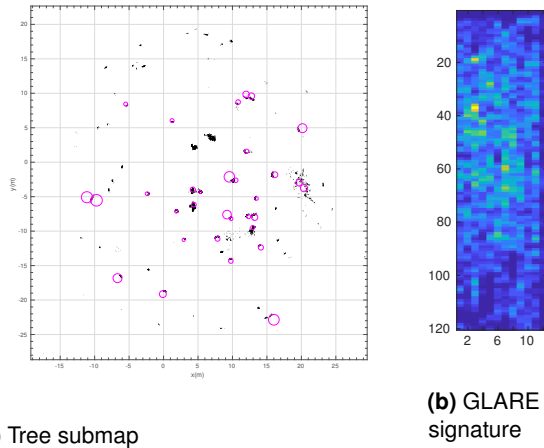


Figure 3. (a): Example submap generated from real forest data collected at NASA LaRC. Black dots show the raw laser returns over a period of 20 seconds. Each magenta circle shows a tree tracked within this submap, where the radius corresponds to the estimated tree stem radius. (b): The corresponding GLARE signature. Lighter color denotes higher value.

finding the best frontier for exploration using a UAV can be a challenging problem. Classic approaches (Yamauchi et al. 1998) usually select the closest frontier based on the Euclidean distance. 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 lead to 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 propose to 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

$$j(a) = j_\theta(a) + \lambda j_t(a), \quad (1)$$

where $j_t(\cdot)$ 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 λ is a balance parameter that trades off the two cost terms. 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. Because of the orientation cost term $j_\theta(\cdot)$, switching to the new frontier typically produces minimal heading changes and thus ensures 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 memory 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 address the issue of heavy payload, we turn to *object-based* representations, more specifically a *tree-based* representation (Kukko et al. 2017). Before communication, we compress each submap into a sparse collection of trees as show in Figure 3a. 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 then fitted to a circle in 2D that represents a tree trunk, by applying an algebraic fitting algorithm (Taubin 1991) that estimates a tree position and radius and returns a residual error. If the initial residual error after the algebraic fitting is sufficiently low, a geometric fitting based on Levenberg-Marquardt (Chernov 2010) that uses the algebraic fitting as initial guess is employed to refine the estimated tree parameters. A tree detection is accepted, if the final residual error after geometric fitting is less than 0.015, the tree has a radius greater than 0.1m, and the observed LIDAR beams cover more than 30% of the tree trunk.

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 0.5m 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 τ_{cull} . Here, τ_{cull} is a tunable parameter designed to cope with varying level of noise during tree detection.

5. Offboard Global Mapping

Recall that on the ground station, we construct a globally consistent map of the world by fusing identical trees detected across multiple submaps and subsequently optimizing the positions of all trees in the fused map. The overall problem is clearly combinatorial and thus extremely hard to solve optimally; on the other hand, 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 present the following centralized pipeline to efficiently (albeit approximately) solve the global mapping problem on the ground station; see Figure 2b. 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). To mitigate the effect of perceptual aliasing, the set of all pairwise associations are further optimized based on the criterion of cycle consistency (Section 5.3). Given the final data association result, we run landmark-based SLAM to optimize the estimates of all vehicle poses and tree locations 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) , their distance $\rho_{i,j}$ and absolute relative angle $\theta_{i,j}^+$ are computed. These are then assigned to bins in a 2D histogram $(\theta_{i,j}^+, \rho_{i,j}^t) \in \text{bin}(n_\theta, n_\rho)$, where $n_\theta \in [N_\theta]$ and $n_\rho \in [N_\rho]$ are indices corresponding to the quantization of $(\theta_{i,j}^+, \rho_{i,j}^t)$ in the range $[0, \rho_{\max}]m \times [0, \pi]\text{rad}$. To account for noise during tree detection, a Gaussian blur is applied to the histogram of each feature pair, and the GLARE descriptor is then the sum of the 2D histograms over all pairs. See Figure 3b for the GLARE signature extracted from the submap in Figure 3a. Following Giamou et al. (2017), we set the default histogram resolutions to be $N_\rho = 120$, $N_\theta = 12$.

To decide if two submaps are likely to form 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 $G^s, G^t \in \mathbb{R}^{N_\rho \times N_\theta}$,

$$SL_1(G^s, G^t) = \min_{0 \leq k < N_\theta} \sum_{i=0}^{N_\rho-1} \sum_{j=0}^{N_\theta-1} |G_{i,j}^t - G_{i,(j+k) \bmod N_\theta}^s|. \quad (2)$$

Each GLAROT query requires $N_\rho \times N_\theta^2$ operations which is very fast on standard CPUs. A pair of submaps (M^s, M^t) is declared to be a loop closure candidate, if their GLAROT distance $SL_1(G^s, G^t)$ is below a threshold ϵ_{GLAROT} . This procedure serves as an efficient filtering step to reduce the search space for loop closures to a small set of candidate matches. To verify if each candidate is a true loop closure, we apply a pairwise data association procedure which identifies the pairwise correspondences between trees in the two submaps. This is detailed in the next section.

5.2. Pairwise Loop Closure Verification

For each potential loop closure between a pair of submaps (M^s, M^t) , we verify if (M^s, M^t) 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^s and M^t could potentially come from different vehicles, classical pairwise association methods such as Nearest Neighbor or JCB 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 to alternative methods based on Generalized Hough Transform or RANSAC (Giamou et al. 2017). Let $\{o_i^s\}$ and $\{o_j^t\}$ represent the set of objects in M^s and M^t . In addition, let $\{p_i^s\}, \{p_j^t\}$ be the positions of these objects in the coordinate frames of M^s and M^t , respectively. A correspondence graph is an undirected graph in which each vertex $u_{i,j}^{s,t}$ denotes a hypothetical match between $o_i^s \in M^s$ and $o_j^t \in M^t$. Two vertices $u_{i,j}^{s,t}, u_{k,l}^{s,t}$ are connected by an edge if relative transformation is preserved, i.e., if,

$$\|(p_i^s - p_k^s) - (p_j^t - p_l^t)\|_2 \leq \epsilon_{\text{CG}}, \quad (3)$$

where ϵ_{CG} is an adjustable tolerance parameter.

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

$$\tilde{\Pi}_t^s(i, j) = \begin{cases} 1, & \text{if } \tilde{\pi}_t^s(i) = j \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

and m^s and m^t are the sizes of the two submaps. M^s and M^t are declared to be a loop closure, if $\tilde{\pi}_t^s$ contains sufficient amount of matches, specified by a tunable parameter τ_{CG} .

We make a note on the complexity of CG matching. Although the maximum clique problem is combinatorial in nature, in our case it could still be solved efficiently and in real-time, thanks to the small problem size owing to our object-based representation. To solve the maximum clique problem, we use the implementation provided by Konc and Janei (2007). Empirical runtime results are reported in the experiment section.

5.3. Globally Consistent Loop Closures

Given a collection of noisy pairwise associations $\{\tilde{\pi}_t^s\}$ between n submaps, we perform a final optimization on the pairwise associations to recover the *global* association $\{\pi_u^s\}$, which maps objects in each submap M^s to the *universe* U of objects, i.e., the set of all trees in the forest. Mathematically, recovering $\{\pi_u^s\}$ requires the pairwise associations $\{\tilde{\pi}_t^s\}$ 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* $\mathcal{G}$ (Leonardos et al. 2017) induced by $\{\tilde{\pi}_t^s\}$.[†] Each vertex in $\mathcal{G}$ represents an object o_i^s in a submap M^s . Two vertices o_i^s, o_j^t are connected by an edge if they are matched in the input associations, i.e., if $\tilde{\pi}_t^s(i) = j$. We note that these edges are inherently *transitive*: if o_i^s is matched to o_j^t and o_j^t is matched to o_k^t , then it must be true that o_i^s is matched to o_k^t . Fathian et al. (2019) show that $\{\tilde{\pi}_t^s\}$ is cycle consistent if and only if the corresponding data association graph consists of disjoint cliques. In this case, map fusion happens naturally by assigning each clique in $\mathcal{G}$ to a unique object in the universe U . See Figure 4a for an example.

In practice, however, $\{\tilde{\pi}_t^s\}$ often contain wrong matches that violate the cycle consistency principle. A typical example is shown in Figure 4b. 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

[†] $\mathcal{G}$ should be distinguished from the correspondence graph defined in the previous section for pairwise association.

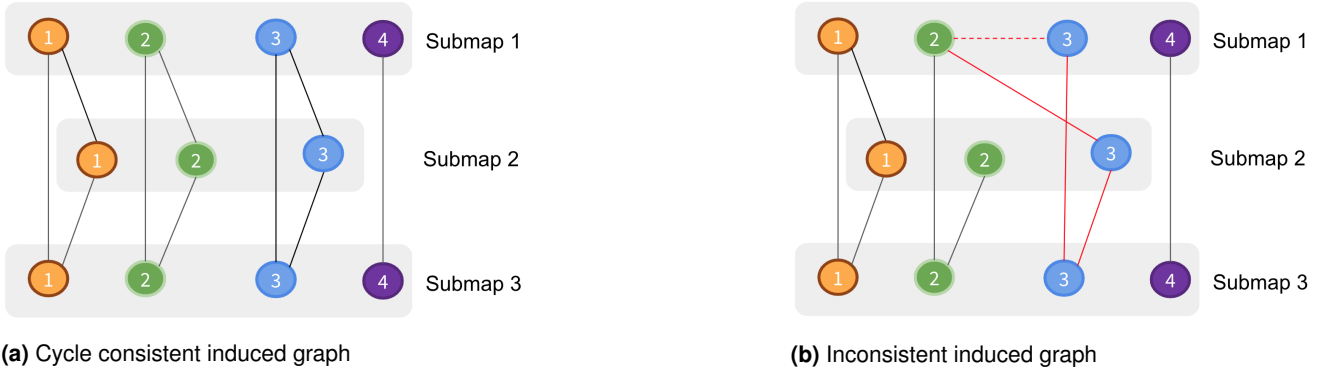


Figure 4. Example data association graphs. Each vertex (circle) represents an object (tree) in a submap. Two vertices are connected if the corresponding objects are matched in the input pairwise associations. **(a):** If the input associations are cycle consistent, objects matched together will form 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):** 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). To estimate the global data association, we hence need to find the closest cycle consistent associations to the input associations.

must find a cycle consistent set of pairwise associations $\{\pi_t^s\}$ that resembles the input associations $\{\tilde{\pi}_t^s\}$ as much as possible. Given $\{\pi_t^s\}$, it is again straightforward to recover the underlying global associations $\{\pi_u^s\}$, by assigning each clique in the induced data association graph of $\{\pi_t^s\}$ to a unique object. In the computer vision literature, this problem 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 Fathian et al. (2019) for its high precision and superior speed. CLEAR takes as input the aggregate matrix,

$$A = \begin{bmatrix} 0 & \tilde{\Pi}_2^1 & \cdots & \tilde{\Pi}_n^1 \\ \tilde{\Pi}_1^2 & 0 & \cdots & \tilde{\Pi}_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\Pi}_1^n & \tilde{\Pi}_2^n & \cdots & 0 \end{bmatrix}, \quad (5)$$

which corresponds to the adjacency matrix of $\mathcal{G}$. Let D denotes the (diagonal) degree matrix of $\mathcal{G}$. The Laplacian and normalized Laplacian of $\mathcal{G}$ are defined as follows,

$$L \triangleq D - A, \quad L_{\text{norm}} \triangleq D^{-\frac{1}{2}} L D^{-\frac{1}{2}}. \quad (6)$$

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_{norm} . In the second stage, an eigendecomposition is performed over L_{norm} to obtain an embedding vector $v_i^s \in \mathbb{R}^m$ for each vertex o_i^s in $\mathcal{G}$. These embeddings are then used to cluster vertices into m disjoint cliques using the Hungarian algorithm.

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

$$\pi_t^s = \pi_u^s \circ \pi_t^u, \quad (7)$$

where $\circ$ denotes the composition operator for permutations. More details regarding the algorithm can be found in (Fathian et al. 2019).

5.4. Global Landmark-based SLAM

After solving for the global data association $\{\pi_u^s\}$, we perform landmark-based SLAM that refines the estimates of all poses and objects in the fused map. To do this, we jointly optimize over the origin $x_t \in \text{SE}(2)$ of each submap M^t , which also corresponds to the vehicle pose when M^t is initialized, as well as the positions of the m objects $l_1, \dots, l_m \in \mathbb{R}^2$ in the environment. Recall that m is given as an output of global data association (Section 5.3).

Following the standard formulation of landmark-based SLAM, we consider two types of measurements during optimization. First, we use *odometry* measurements $z_{t+1}^t \in \text{SE}(2)$ 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 $\{\pi_u^s\}$. For each object o_i^s in map M^s , we retrieve the index k of the corresponding object in the universe via $k = \pi_u^s(i)$. An observation is then created between x_s and l_k , with a relative transformation constraint $p_i^s \in \mathbb{R}^2$.

After all variables and measurements are initialized, we use iSAM2 (Kaess et al. 2011) to carry out the global optimization. During optimization, we make use of robust Huber cost as the last layer of protection against outliers during data association. Finally, the optimized map origins $\{x_t\}$ are sent back to the vehicles to correct drift during local estimation, by realigning all submaps using the updated values of $\{x_t\}$.

6. Experiments

We evaluated the performance of the proposed system in both simulation and real-world flight tests. Both experiments used the same set of parameters for frontier exploration, but a few vehicle-related parameters differed to compensate for the differences in the simulation and the real world. For example, the limit on maximum allowed acceleration was set to be much higher in real-world experiments, in order to

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 are recorded. In all three cases, the proposed planner was able to fly faster and cover the area using much shorter amount of time compared to the baseline planner.

Area	Planner	Duration (sec)	Avg. Speed (m/sec)
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

compensate for external forces such as the wind that are not modeled in simulation.

6.1. Simulation Setup

We benchmarked our proposed system with the baseline in simulation. Vehicle dynamics and inertial measurements unit (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 the 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 30m 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. Frontier-based Exploration Results in Simulation

We evaluated the proposed frontier-based exploration planner in simulation. A single UAV was tasked with covering a 20m × 20m search area in a randomly generated forest. To benchmark our proposed planner, we also implemented a baseline planner that greedily selected the closest frontier (Yamauchi et al. 1998) without online update. For comprehensive evaluation, different search areas with varying search difficulty (e.g., density of obstacles) are assigned to the vehicle. For each search area, multiple exploration missions were carried out and average performances are recorded in 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 5 shows the trajectory of the proposed planner (a)-(d) and the baseline planner (e)-(h) in an example search area. The proposed produced a much smoother overall trajectory compared to the baseline. In addition, the proposed planner is clearly more time-efficient. At $t = 250$ sec, the proposed planner nearly completed the mission, while there was still a significant portion of the map uncovered by the baseline planner.

6.3. Outdoor Flight Setup

Real flight tests were performed in the forest at NASA LaRC, shown in Figure 1. Each vehicle was a modified DJI

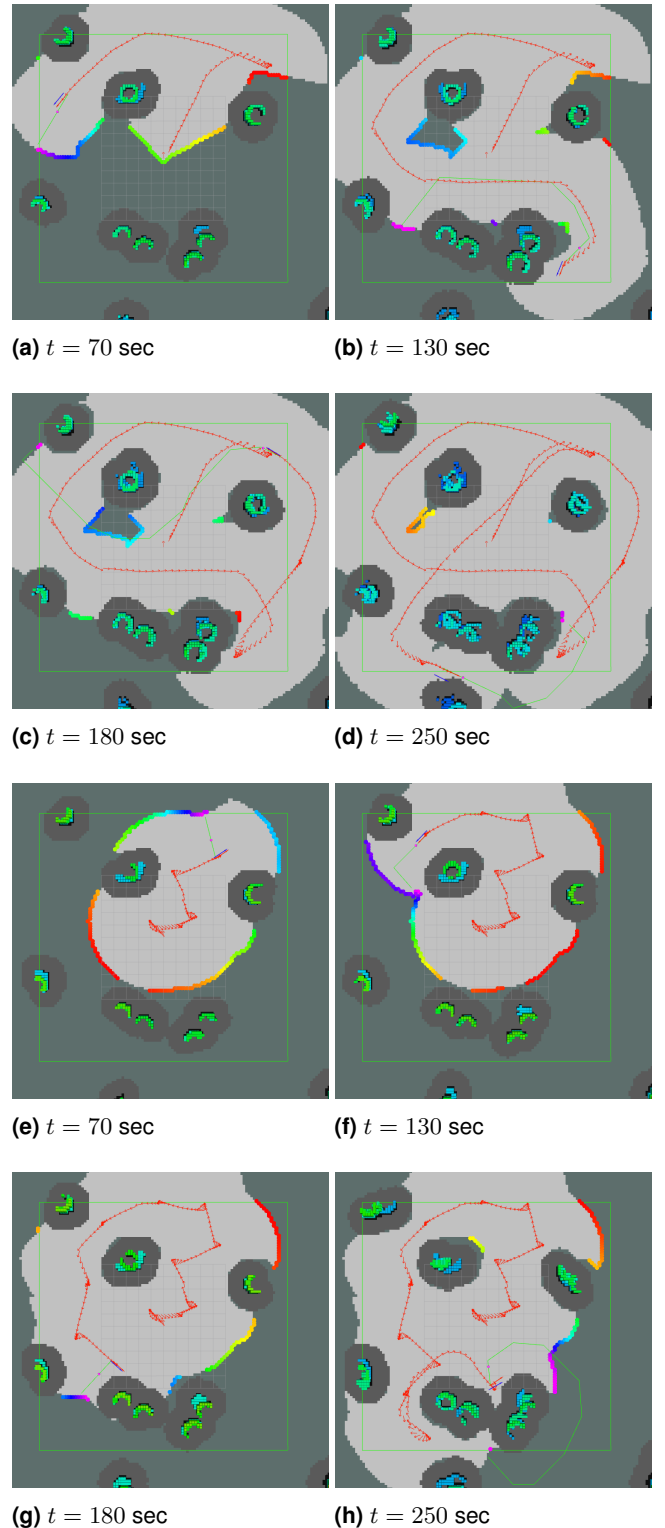


Figure 5. Vehicle trajectories and partial maps in simulation at different time steps (seconds). (a)-(d) show the proposed planner; (e)-(h) show the baseline planner. Green rectangle denotes the search area assigned to the vehicle. Colored point clouds shows the set of all frontiers with colors representing different costs (magenta to red in increasing cost). Red path shows the trajectory taken by the vehicle. The trajectory in (a)-(d) is evidently smoother and more time-efficient compared to the trajectory in (e)-(h).

F450 carrying a horizontally mounted Hokuyo UTM-30LX laser rangefinder, a Pixhawk PX4 unit providing inertial

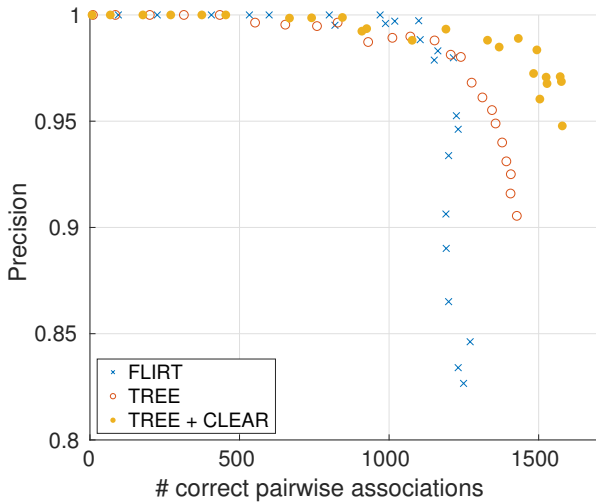


Figure 6. Precision vs. number of correct pairwise associations. The horizontal axis shows the number of correct pairwise associations (unnormalized recall), and the vertical axis shows the corresponding precision, i.e., percentage of correct pairwise associations within all proposed associations.

measurements and motor commands, a downward-facing LidarLite for altitude measurements, and an Intel NUC computer for on-board 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 LidarLite measurements were processed at 100 Hz. Communication with the ground station (for map fusion) is maintained through 5.8 GHz WiFi. In addition, the vehicles were commanded to fly at 1.8 m altitude with a maximum allowed velocity of 2.0 m/s, and a maximum acceleration of 0.4 m/s^2 .

6.4. Loop Closure and Data Association Results

We evaluated the performance of the proposed data association procedure using data collected by a single UAV in the forest. For this experiment, place recognition (Section 5.1) is disabled and loop closure is attempted on every pair of submaps in the database. As baseline, we implemented the approach in (Giamou et al. 2017) which uses stabilized FLIRT features obtained from DP-means clustering. Since ground truth (e.g., GPS) is not available, we instead use EKF estimates to classify if a proposed pairwise association is correct. Specifically, a pair of matched objects is declared to be a true match, if their positions as estimated by EKF in the world frame are sufficiently close. For short-duration flights, we expect this metric to be relatively accurate.

Figure 6 shows the precision-recall on an example forest sequence with 24 submaps in total. The horizontal axis shows the number of correct pairwise associations (unnormalized recall) recovered by the algorithm,[‡] and the vertical axis shows the corresponding precision, i.e., percentage of correct pairwise associations within all proposed associations. For each submap, an equal number of stabilized FLIRT features and tree landmarks were extracted. CG matching (Section 5.2) with varying tolerance threshold ϵ_{CG} were performed to obtain the resulting data points for FLIRT and trees. For trees,

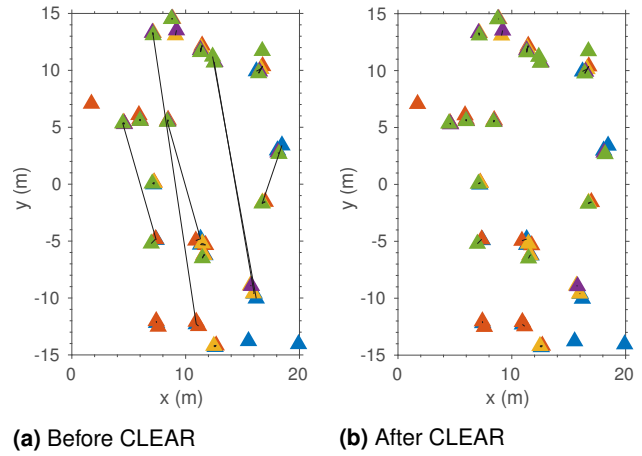


Figure 7. Part of data association graph before and after optimizing with CLEAR. Each triangle represents an object, where color corresponds the submap it belongs to. Two objects from different submaps are connected by an edge if they are matched in the given data association.

we further optimize the initial pairwise associations using the CLEAR algorithm (Section 5.3). In the forest, using tree landmarks unsurprisingly outperformed FLIRT features. Furthermore, including the additional stage of multiway matching further improved both precision and recall during data association.

Figure 7 shows part of an example data association graph (Section 5.3) in the forest dataset before and after optimizing with CLEAR. Each triangle represents an object, where color corresponds to the submap it belongs to. Two objects from different submaps are connected by an edge if they are matched in the given data association. For visualization purpose, the submaps are aligned in the world frame, i.e., each object is drawn at the corresponding position estimated by EKF. Thus, objects that corresponds to the same tree in the forest should appear close to each other. Due to perceptual aliasing, the initial associations from CG matching contained several wrong correspondences (long edges in Figure 7a). By jointly optimizing over all pairwise matches, CLEAR was able to reject these outliers (Figure 7b). Furthermore, the resulting data association is guaranteed to be cycle consistent. This means that the graph shown in 7b only contains disjoint cliques which makes subsequent map fusion possible; see Section 5.3.

6.5. Real-time Planning and Map Fusion Results

To demonstrate the proposed collaborative exploration and mapping system, we performed real-world flight tests in the NASA LaRC forest. The vehicles were started at different locations, with unknown relative positions. The mission was specified by non-overlapping search bounds of size $17\text{m} \times 20\text{m}$ for each vehicle. Each vehicle was tasked with observing the entire search region. The map was initially unknown; as the vehicles individually completed the coverage

[‡] We do not show recall as the total number of correct pairwise associations is unknown.

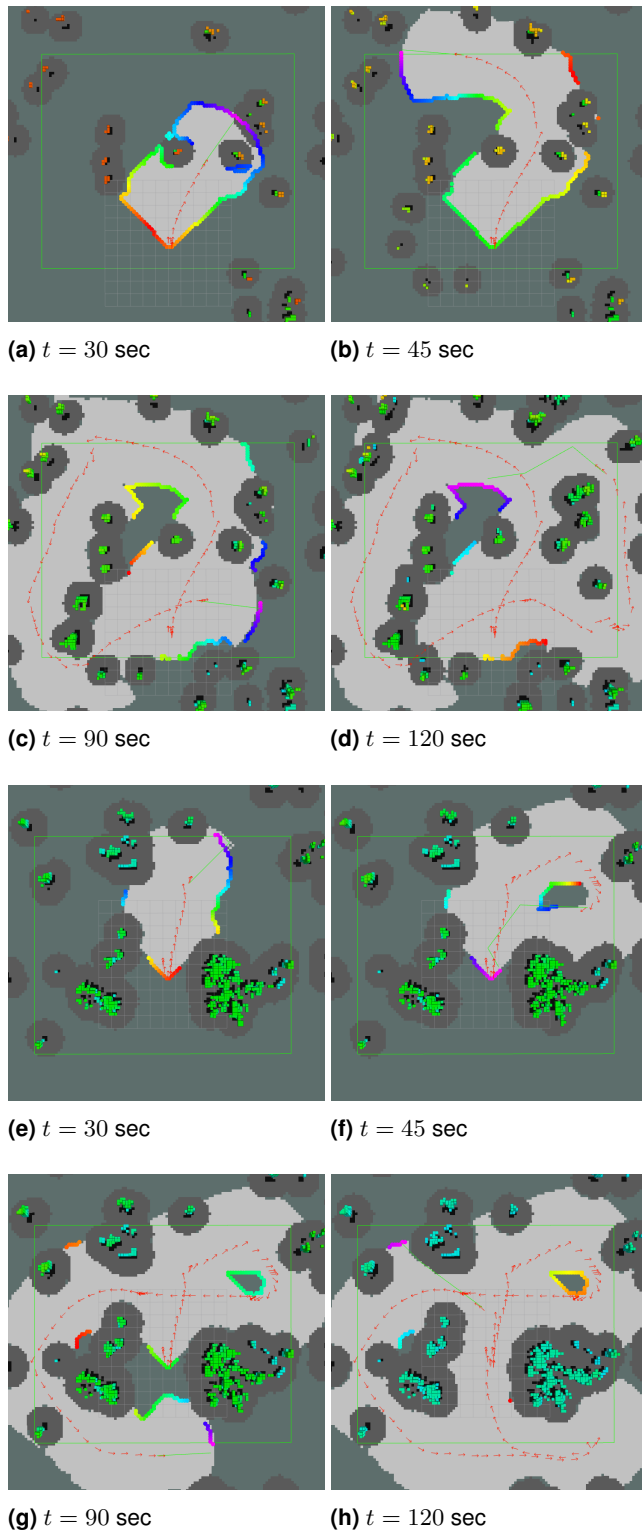


Figure 8. Vehicle trajectories in real flight tests at different time steps (seconds). (a)-(d) shows the trajectory of the first vehicle and (e)-(h) shows the second vehicle. Map merging happens off-board on the ground station at $t = 45$ sec.

task, they established inter-trajectory loop closures based on similar tree features observed during flight. The individual maps from both vehicles were then merged realtime on the ground station. Figure shows the final map that resulted from the example mission. The progress of planners throughout the flight is shown in Figure 8 and roughly aligned on-board

Table 2. Flight durations and average speeds for flight tests at NASA LaRC.

	Duration (sec)	Avg. Speed (m/sec)
Vehicle 1	122.89	2.66
Vehicle 2	135.25	2.65

camera images are shown in Figure 9; mission duration and average speeds are reported in Table 2. Note that due 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 velocities. Since the proposed planner preferred frontiers with smaller orientation change, we observed that the flight behavior exhibited bias towards a natural spiral pattern.

7. Main Experimental Insights

While one-step greedy frontier selection is computationally convenient, one of the goals in the search and rescue domain is to efficiently search the forest environment. We observed that choosing the nearest frontier in heavily occluded environments such as forests led to myopic and inefficient planning, especially if a new frontier can only be selected once the current frontier has been explored. Often times, the nearest frontier was located close to the previous frontier, which led to the vehicle flying a short distance, then stopping to select the next frontier to visit. An example of such inefficiencies is shown in Figure 5(e)-(h), where the dynamics-agnostic planner frequently stopped when performing search in the simulated forest, and covered the search window in a longer time than necessary.

A naive improvement to the frequent stopping problem could be solved by constantly re-planning, instead of requiring that the vehicle reach the intended frontier. However, constantly re-planning using only distance in the cost function may cause the vehicle to continuously command large changes in heading, which generally leads to flights that are less smooth. For slower quadrotor flight, this may be acceptable, but for time sensitive tasks such as search and rescue, we would like the vehicle to keep up search speed when possible. Additionally, limits on vehicle acceleration, coupled with the myopic planner results in the vehicle flying short, slower trajectories, as the trajectory to the next best (i.e. closest) frontier is short and does not allow the vehicle to pick up speed.

While we keep the computational tractability of one-step greedy frontier selection, our approach encodes the behavior of a longer-horizon, dynamics-aware planner by putting a heuristic cost on the deviation of a frontier from the vehicle's current heading vector. For example, in the real forest flights shown in Figure 8, the first vehicle first visited the top-right corner of the search window first, favoring the direction of the vehicle velocity, before heading to the bottom section in a naturally occurring spiral pattern.

While the new planner demonstrates a desirable emergent search behavior, we observe that the 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 (as in Figure 8, where the second vehicle misses the tear-dropped shaped section of unknown space). A further



Figure 9. Camera images from on-board GoPro camera at different time steps (seconds); First two rows show camera images from the first vehicle and the last two rows show images from the second vehicle. Map merging happens at $t = 45$. Frames are approximately aligned with the sensor data shown in Figure 8.

sophisticated approach may involve a clustering approach to encourage the planner to visit small patches of unobserved space. We leave this for future work.

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, Torii 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 (CVPRW'06)*.
- Aydemir A, Sjöö K, Folkesson J, Pronobis A and Jensfelt P (2011) Search in the real world: Active visual object search based on spatial relations. In: *Proceedings of the International Conference on Robotics and Automation (ICRA)*.
- 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. *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)*.
- 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 Carin L (2013) Dynamic clustering via asymptotics of the dependent dirichlet process mixture. In: *Advances in Neural Information Processing Systems*, 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*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability.
- 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 (IJRR)*.
- Cieslewski T, Choudhary S and Scaramuzza D (2018) Data-efficient decentralized visual SLAM. *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)*.
- Corah M and Michael N (2017) Efficient online multi-robot exploration via distributed sequential greedy assignment. In: *Robotics: Science and Systems (RSS)*.
- Corah M, OMeadhra C, Goel K and Michael N (2019) Communication-efficient planning and mapping for multi-robot exploration in large environments. *IEEE Robotics and Automation Letters (RAL)*.
- Cummins M and Newman P (2011) Appearance-only slam at large scale with FAB-MAP 2.0. *The International Journal of Robotics Research (IJRR)* 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.
- Feder HJS, Leonard JJ and Smith CM (1999) Adaptive mobile robot navigation and mapping. *The International Journal of Robotics Research, Special Issue on Field and Service Robotics* 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.
- Forster C, Lynen S, Kneip L and Scaramuzza D (2013) Collaborative monocular slam with multiple micro aerial vehicles. In: *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*.
- Gálvez-López D and Tardos 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, scar Reinoso, Ballesta M and Juli M (2010) Multi-robot visual slam using a rao-blackwellized particle filter. *Robotics and Autonomous Systems (RAS)*.
- Gonzalez-Baos HH and Latombe JC (2002) Navigation strategies for exploring indoor environments. *The International Journal of Robotics Research (IJRR)*.
- 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*.
- Himstedt M, Frost J, Hellbach S, Böhme HJ and Maehle E (2014) Large scale place recognition in 2D lidar scans using geometrical landmark relations. 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) Multi-robot 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 Prez 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, Snchez J, Prez P and Schmid C (2012) Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Jensen EA (2018) Restricted communication in online multi-robot exploration. In: *Proceedings of the Joint Conference on Artificial Intelligence (IJCAI)*.
- Kaess M and Dellaert F (2009) Covariance recovery from a square root information matrix for data association. *Robotics and Autonomous Systems (RAS)*.
- Kaess M, Johannsson H, Roberts R, Ila V, Leonard J and Dellaert F (2011) isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering. In: *Proceedings of the International Conference on Robotics and Automation (ICRA)*.
- Kallasi 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)*.
- Konc J and Janei D (2007) An improved branch and bound algorithm for the maximum clique problem. *MATCH - Communications in Mathematical and in Computer Chemistry*.
- Kukko A, Kaijaluo R, Kaartinen H, Lehtola VV, Jaakkola A and Hyyp J (2017) Graph slam correction for single scanner mls forest data under boreal forest canopy. *ISPRS Journal of Photogrammetry and Remote Sensing*.
- Leonard J and Feder HJ (2001) Decoupled stochastic mapping. *IEEE Journal of Oceanic Engineering*.
- 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*.
- 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 Sags C (2013) Distributed data association in robotic networks with cameras and limited communications. *IEEE Transactions on Robotics*.
- Morrison JG, Gavez-Lopez D and Sibley G (2016) Scalable multirobot localization and mapping with relative maps: Introducing moarslam. *IEEE Control Systems Magazine*.
- 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*.
- Neira J and Tardos JD (2001) Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*.
- 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*.
- 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*.
- 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, Snchez 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 (JFR)* .
- Schmuck P and Chli M (2018) CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams. In: *Journal of Field Robotics (JFR)*.
- 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 (JFR)* .
- Singh A, Krause A, Guestrin C and Kaiser WJ (2009) Efficient informative sensing using multiple robots. *Journal of Artificial Intelligence Research* .
- Tang J, Chen Y, Kukko A, Kaartinen H, Jaakkola A, Khoramshahi E, Hakala T, Hyypä J, Holopainen M and Hyypä H (2015) Slam-aided stem mapping for forest inventory with small-footprint mobile lidar. *Forests* .
- Taubin G (1991) Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* .
- Tedrake R (2014) Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems .
- Tipaldi GD and Arras KO (2010) Flirt-interest regions for 2D range data. In: *Proc. ICRA*. pp. 3616–3622.
- 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)*.
- 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)*.