

Search and Rescue under the Forest Canopy using Multiple UAS

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

¹ Massachusetts Institute of Technology, Cambridge, MA 02139, USA,
`{yulun, katliu, kyelok, nickroy, jhow}@mit.edu`

² NASA Langley Research Center, Hampton, Virginia USA
`{loc.d.tran, danette.allen}@nasa.gov`

Abstract. We present an autonomous system consisting of multiple unmanned aerial vehicles (UAVs) for search and rescue under the forest canopy. Our vehicles can be rapidly deployed, can collaboratively explore expanses of terrain efficiently, and are agile enough to operate in reasonably thick forests. We demonstrate the ability to carry out GPS-denied exploration with on-board pose estimation, map inference, and path planning. In addition, we utilize a place recognition system that is able to handle perceptual aliasing unique to a forest environment, and fuse individual areas explored by each vehicle into a globally consistent map. We perform extensive evaluations in both simulation and real-world to demonstrate the effectiveness of our proposed system.

1 Motivation

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 at a time because manned aircraft cannot see through the overhead forest canopy. Small autonomous unmanned aircraft systems (UAS), or drones, have been frequently proposed 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 cover expanses of terrain quickly and are small enough to operate in reasonably thick forests.

2 Problem Statement

Using drones under the forest canopy presents several challenges. Firstly, GPS typically cannot penetrate the overhead foliage, which requires the vehicles to use on-board sensors and a mapping and localization system for state estimation. A second problem is that these vehicles are typically limited by battery life, which limits the search speed. Any search activity is accelerated by using multiple vehicles in parallel, but a multi-vehicle search requires co-ordination between the vehicles in both data fusion and planning. A third problem is that the search can be made arbitrarily slow with a poor strategy in terms of where to search next. Using more intelligent search strategies can dramatically accelerate the process.

In this paper we describe an experimental evaluation of a multi-UAS system for search and rescue under the forest canopy. We examine the ability of multiple

vehicles to carry out GPS-denied flight using laser-range finders for position estimation and map inference, plan trajectories, and fuse individual maps into a globally consistent model for exploration and search. We evaluate both a map construction process designed for search and exploration in the forest, and the corresponding search process itself.

3 Related Work

The problem of place recognition and map fusion under the forest canopy has received increased attention in recent years. Place recognition is a key component of any SLAM system and has a rich literature for single robot systems equipped with stereo or monocular camera [7,18,17], but these approaches typically require access to multiple images which would be costly to communicate over a network with limited bandwidth, and the high variability of illumination conditions in a forest environment creates challenges for purely image-based solutions.

Using laser-range data, Bosse and Zlot [4] developed a keypoint-based nearest neighbor algorithm for identifying loop closures in forest-type environments, but primarily using an underactuated 3D laser that would be a substantial challenge on a flight vehicle. Tipaldi et al. [22] developed fast laser interest region transform (FLIRT) features for both indoor and outdoor localization. Purely geometric approaches have been examined, including the Geometric LAndmark RElations (GLARE) signatures by Himstedt et al. [10] and the rotationally-invariant extension GLAROT by Kallasi and Rizzini [13]. None of these previous results take advantage of the strong prior structure present in forest environment.

The problems of search and exploration has been extremely well-studied and a comprehensive overview of all such work is outside the scope of this paper. The earliest approaches operated greedily on the nearest “frontier” in the map [23] or maximized expected coverage [20]. Frontier-based approaches have also been demonstrated for multi-UAV exploration in an indoor environment [6]. Later approaches to the exploration problem focused on either greedy [8] or receding horizon [5] information gain. Most recently, there have been results on exploring to find hidden objects [1], but primarily in indoor environments. Surprisingly, the majority of this prior work has not taken into account the dynamics of the vehicle in planning efficient search strategies.

4 Technical Approach

Our experimental investigation focuses on two primary areas, feature extraction for place recognition and map merging, and exploration-based search. We follow the approach described by Giamou et al. [9], extending to multiple vehicles that are communicating with a central ground station³. When a new laser scan or scans are received, features are extracted for place recognition and loop closure. Once a loop is closed, the merged map can be inspected by the first responders to plan a route for the rescue personnel.

³ Our experiments assume reliable communication to the ground station, which is generally not the case for operation in a forest environment. However, this is not a limiting assumption in that all of the local inference and map construction happen on-board the vehicle. The loss of the C2 link to the ground station could only delay potential map merges until the link is recovered.

4.1 Tree Feature Extraction and Map Merging

A forest is a challenging environment for cooperative SLAM due to strong perceptual aliasing and complex occlusions caused by a dense assembly of visually similar trees. Before place recognition can occur, we need to extract stable features from the laser scans. We first cluster the laser return; each cluster of points is then regressed to the parameters of a tree trunk circle $[x, y, r]$ using least squares. We accept a cluster of points as the observation of a tree, if the residual is less than 0.015, the tree has a radius greater than 0.1m, and the observed laser returns cover more than 30% of the arc of the tree trunk circle. Following [14], we correspond a detected tree at time t with a previously detected tree at time t' if the centers are within 1m, and the radii of the trees differ less than 10%.

Once the trees are extracted as features $\{p_i^t\}$ from a scan S^t , a GLARE signature [10] is computed, which is essentially a distribution over the pairwise Euclidean distance $\rho_{i,j}^t$ and angular distance $\theta_{i,j}^+$ between pairs of known trees. For every pair of features (p_i^t, p_j^t) their distance and relative angle in the scan frame $(\theta_{i,j}^t, \rho_{i,j}^t)$ 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_θ and n_ρ are a quantization of $(\theta^+, \rho_{i,j}^t)$. A Gaussian blur is applied to the histogram of each feature pair, and the GLARE descriptor is a sum over the 2D histograms for each pair. This histogram construction procedure is not rotationally invariant as it depends on an absolute orientation. Kallasi and Rizzini [13] extended the GLARE signature to be rotationally invariant in GLAROT. We use this extension for comparison of geometric signatures between scans.

After receiving laser scans from the flight vehicles, the ground station searches for loop closures in the entire history of scans. The geometric signatures described above are used to prune the search space by retaining only the top $k = 20$ candidate loop closures for each scan S^t . For each candidate loop closure, we perform correspondence graph (CG) matching [3] to verify the loop closure and identify the largest inlier set of correspondences. If the loop closure is accepted, a relative transformation constraint is computed and added to the global pose-graph for trajectory estimation. We use iSAM2 [12] as the back-end SLAM solver. Map merging happens once all UAVs establish a common reference frame through loop closure.

4.2 Exploration-based Search

Following [19], we formulate the planning problem as a trajectory optimization problem, where a motion planner selects a trajectory of some length that maximizes the probability of finding the target while avoiding observed obstacles. This approach can be represented formally as

$$a_t^* = \operatorname{argmin}_{a_t \in \mathcal{A}} j(b_t, a_t) + h(b_t, a_t), \text{s.t. } g(b_t, a_t) = 0,$$

which is repeatedly re-solved to select the optimal action a_t^* as the robot explores the environment. Here, a_t is an action to be executed at time t , chosen from a set \mathcal{A} of possible actions. The robot's belief b_t is a partial map built from

sensor measurements and its current position in that map. To ensure collision-free motion, we use a collision constraint $g(b_t, a_t)$, which returns 1 if the action intersects obstacles or unknown regions in the map, and returns 0 otherwise.

We parameterize the action space A as frontiers of the known map (i.e., the boundaries between known and unknown space). The total estimated cost of choosing action a_t , given b_t , is the sum of the action cost $j(b_t, a_t)$ and the heuristic estimate $h(b_t, a_t)$ of the cost remaining beyond the end of a_t . For exploration-based search, there is rarely a reason to bias one frontier over another in terms of future cost beyond a_t ; as a result, we set the heuristic cost $h(\cdot)$ to be the same for all frontiers and choose the action with the minimum instant cost described by $j(\cdot)$. Classical approaches design $j(\cdot)$ such that closer frontiers have lower costs than farther ones, e.g., $j(\cdot)$ returns the distance of the frontier. However, in practice it is often the case that the closest frontier lies behind the flight vehicle. In these cases, traditional cost functions lead to rapidly changing velocity and aggressive turning, which negatively impacts the overall search progress and map quality. On the other hand, purely information-theoretic models of $j(\cdot)$ use submodularity to bound the loss that comes from greedily choosing the next-best frontier; the cost of physically redirecting the vehicle to a new frontier is *not* submodular. In this paper, we propose a new search objective that better respects the vehicle dynamics,

$$j(b_t, a_t) = j_\theta(b_t, a_t) + \lambda j_t(b_t, a_t).$$

Above, $j_\theta(\cdot)$ denotes the orientation change from the vehicle's current heading of the frontier a_t . Incorporating orientation change in the cost function discourages the vehicle from excessive turning and hence produces a smoother overall trajectory. In addition, we include a separate, weighted cost term $j_t(\cdot)$ based on the classical Euclidean distance to the frontier. Intuitively, the user-specified weight λ encodes a *trade-off* between the smoothness and overall length of the trajectory. As we shall see in the experimental evaluations, the proposed objective function, with a suitably chosen weight, results in a longer but smoother and more time-efficient trajectory. Lastly, using an objective function based on orientation change enables *online* selection of the next-best frontier. Specifically, the currently selected frontier can be updated, whenever a new candidate with a significantly higher score is detected. In contrast, a dynamic-agnostic planner (e.g., with a purely distance-based objective) is less compatible with this approach, as it might continuously command large change in heading, leading to potentially unstable and dangerous flight behavior.

5 Results and 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 (Section 4.2), 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 compensate for external forces such as the wind that are not modeled in simulation.



Fig. 1: Two quadrotors exploring the forest at NASA Langley Research Center. A video of the flight available at <https://groups.csail.mit.edu/rrg/videos/isr2018>

5.1 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 [21] and a quadrotor model as described in [16]. To test the integration with the full stack, we also utilized the Pixhawk [15] 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. Laserscans and inertial measurements were passed to a laser-based Extended Kalman Filter (EKF) that incorporated incremental odometry measurements from iterative closest point (ICP) on two laser scans with inertial measurements. The vehicle state from the estimator was then used to build a map, which was used by our proposed planner. Finally, the simulated setpoints generated by the motion planner were passed to the Pixhawk SITL.

5.2 Frontier-based Planning Results in Simulation

We evaluated the proposed frontier-based exploration planner (Section 4.2) 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 [23] without online update. For a comprehensive evaluation, we assigned different search areas with varying search difficulty (e.g., density of obstacles) 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 2 shows the trajectory of the proposed planner (a)-(d) and the baseline planner (e)-(h) in an example search area. Note that using the proposed planner resulted in 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

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	Frontier Planner	Duration (sec)	Average 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

mission, while there was still a significant portion of the map uncovered by the baseline planner.

5.3 Outdoor Flight Setup

Real flight tests were performed in the forest at NASA LaRC, shown in Figure 1. Each vehicle was a modified DJI 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 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. Our software stack included a 2D scan-matcher and an EKF [2] for state estimation, 3D occupancy mapping using the Octomap library [11], path planning using A* search in the projected 2D map, as well as the exploration planner presented in Section 4.2. All of state estimation, mapping, and planning happened online and on-board the vehicles, and only map merging occurred online on an off-board ground station. Raw laser scans from each vehicle were communicated via 5.8 GHz WiFi to the ground station for map merging. 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/s².

5.4 Tree Feature Detection and Place Recognition Results

We evaluated the proposed tree feature detection and place recognition algorithm (Section 4.1) in a forest dataset collected at Middlesex Fells Reservation, MA [9]. This dataset contained traversals of the same forest trail in opposite directions, hence including a large number of loop closures. To detect ground truth loop closures, we estimated the sensor rig’s trajectory using the same 2D scan-matcher and EKF described in Section 5.3. The 281 m path was divided into two trajectories to simulate a cooperative SLAM scenario; see Figure 6(a). Figure 6(b)

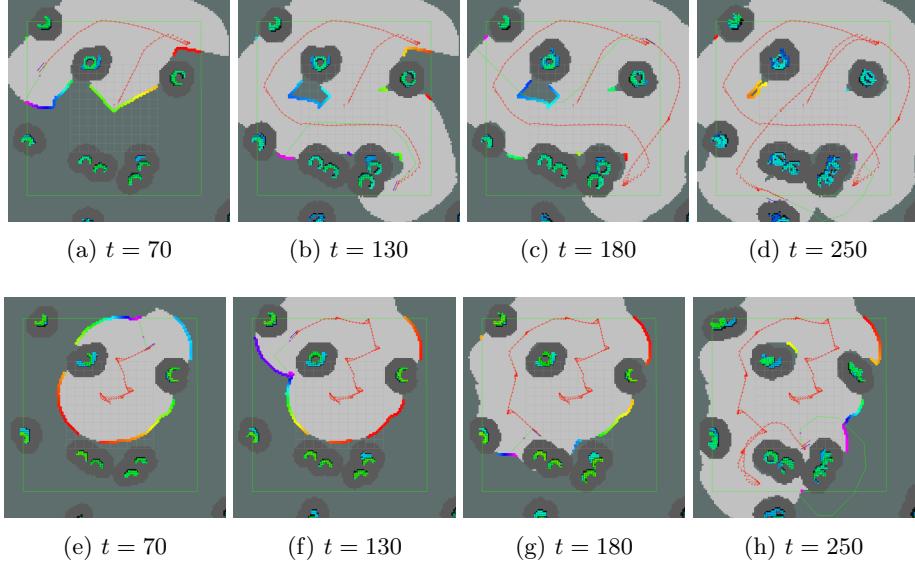


Fig. 2: Vehicle trajectories and partial maps in simulation at different time steps (seconds). (a)-(d) show the proposed planner; (e)-(h) show the baseline planner. Green rectangle denotes the search area assigned to the vehicle. Colored point clouds shows the set of all frontiers with colors representing different scores (magenta to red in decreasing score). Red path shows the trajectory taken by the vehicle. The trajectory in (a)-(d) is much smoother and more time-efficient compared to the trajectory in (e)-(h).

shows the resulting precision-recall curves of both the proposed approach and previous approach that directly clustered FLIRT point clouds without shape fitting to the semantic class of trees. In the forest environment, the performance of the proposed approach unsurprisingly outperformed the previous work. In our evaluation, we imposed a stricter criterion when detecting ground truth loop closures. Specifically, previous work [9] classified a proposed loop closure as a true positive if the corresponding two poses estimated by EKF are sufficiently close to each other. This criterion was sufficient for evaluation of place recognition performance which is the main focus of [9]. However, in our application, the discovered loop closures were eventually used by the back-end SLAM solver to refine trajectory estimate through pose-graph optimization. Hence, it is important to ensure that the relative transformations (factors in the pose-graph) extracted from the loop closures are accurate and do not contain outliers. To check this, we compared each extracted transformation with the transformation estimated by EKF. A proposed loop closure was declared a true positive if and only if the difference between the extracted transformation and the EKF estimate is sufficiently small in terms of both translation and rotation.

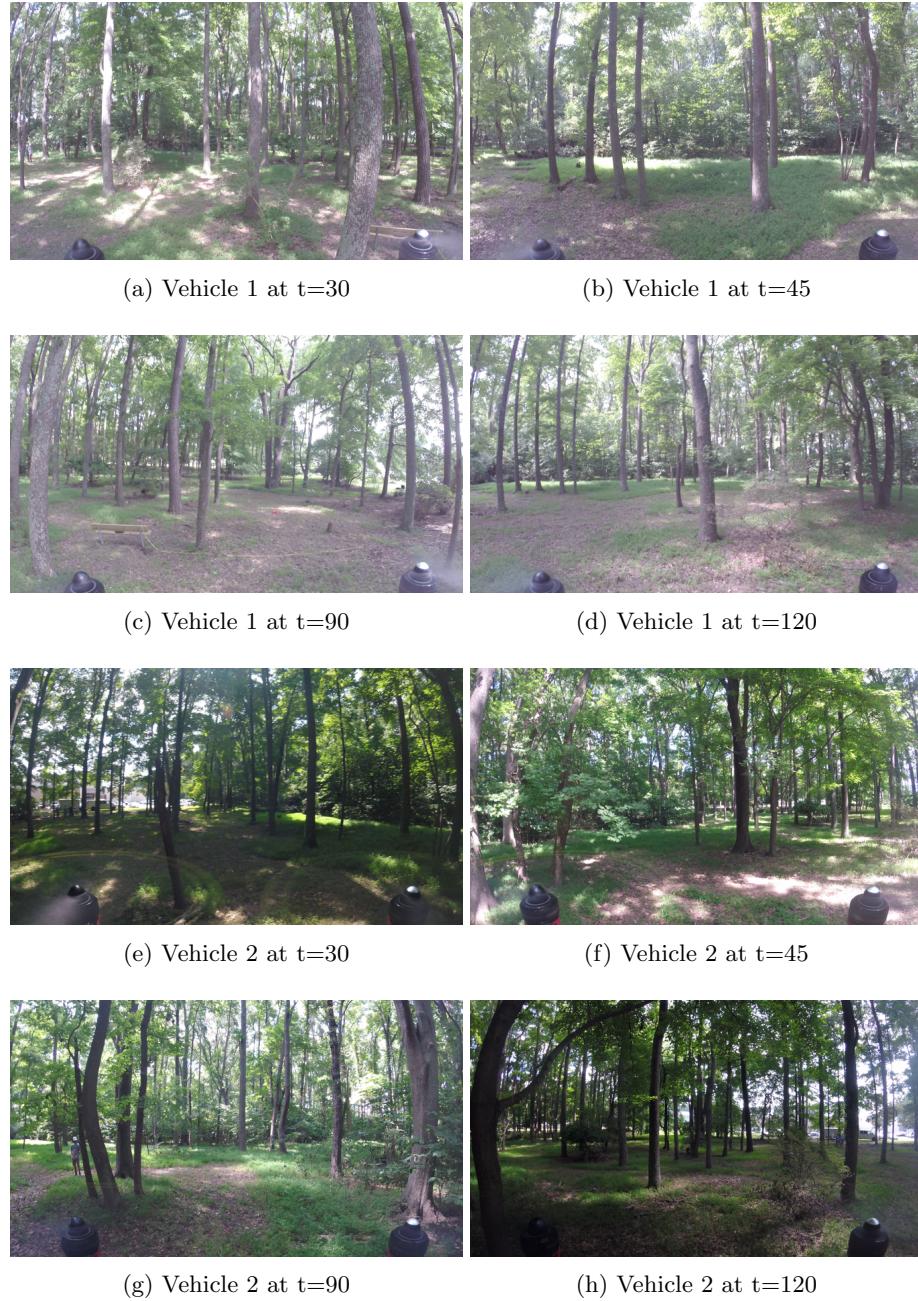


Fig. 3: Camera images from on-board GoPro camera at different time steps (seconds); First two rows show camera images from the first vehicle and the last two rows show images from the second vehicle. Map merging happens at $t = 45$. Frames are approximately aligned with the sensor data shown in Figure 4.

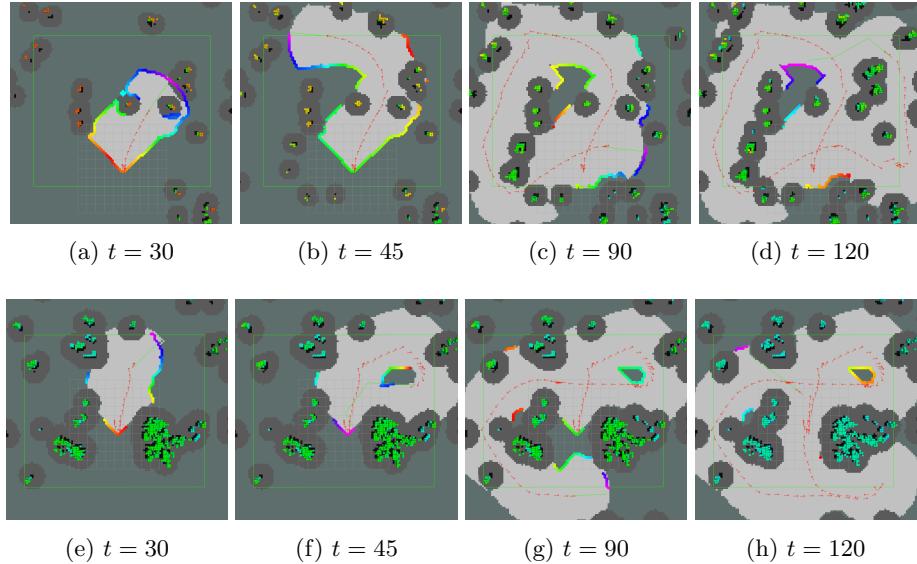


Fig. 4: Vehicle trajectories in real flight tests at different time steps (seconds). (a)-(d) shows the trajectory of the first vehicle and (e)-(h) shows the second vehicle. Map merging happens off-board on the ground station at $t = 45$.

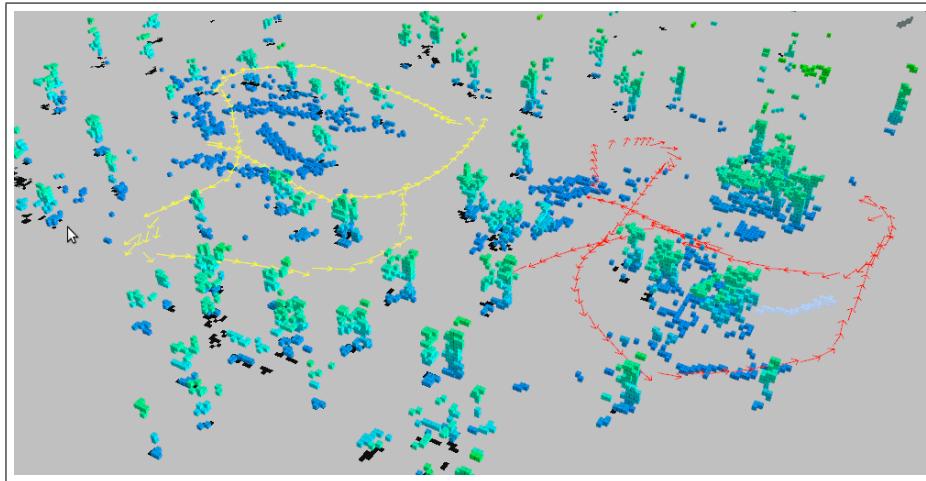


Fig. 5: The merged map resulting from collaborative exploration of the NASA LaRC forest by two vehicles. As the vehicles (paths shown in red and yellow) explore the center area of the forest, they establish inter-trajectory loop closures which enables the local maps to be merged. The resulting joint map is represented as a voxel grid with an altitude-based colormap for occupied cells (blue to green in increasing altitude), grey for unoccupied cells, and dark grey for unvisited cells (none in this figure).

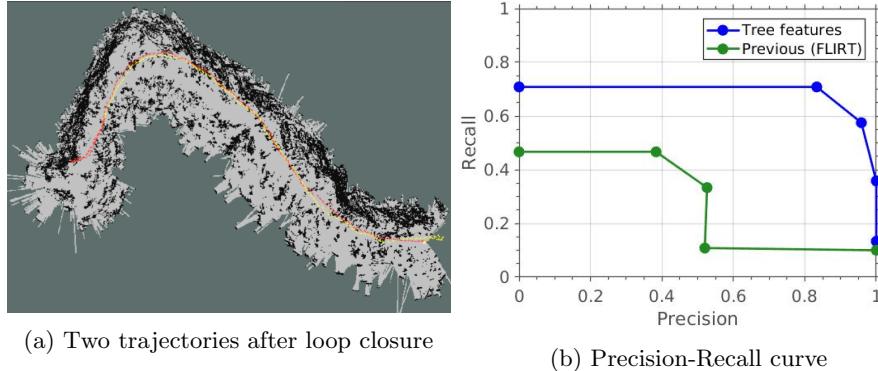


Fig. 6: Performance of the Tree features vs. clustered FLIRT features on the Middlesex forest dataset.

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

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

5.5 Real World Planning and Map Merging Results

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 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 5 shows the final map that resulted from the example mission. The progress of planners throughout the flight is shown in Figure 4 and roughly aligned on-board camera images are shown in Figure 3; 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.

6 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 2(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 4, 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 4, where the second vehicle misses the teardropped shaped section of unknown space). A further 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

1. Alper Aydemir, Kristoffer Sjöö, John Folkesson, Andrzej Pronobis, and Patric Jensfelt. Search in the real world: Active visual object search based on spatial relations. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2011.
2. Abraham Bachrach, Samuel Prentice, Ruijie He, and Nicholas Roy. RANGE—robust autonomous navigation in GPS-denied environments. *Journal of Field Robotics*, 28(5):644–666, 2011.
3. T. Bailey, E. M. Nebot, J. Rosenblatt, and H. F. Durrant-Whyte. Data association for mobile robot navigation: a graph theoretic approach. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2000.
4. M. Bosse and R. Zlot. Place recognition using keypoint voting in large 3D lidar datasets. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2013.
5. F. Bourgault, A.A. Makarenko, S.B. Williams, B. Grocholsky, and H.F. Durrant-Whyte. Information based adaptive robotic exploration. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 2002.
6. K. Cesare, R. Skeele, Soo-Hyun Yoo, Yawei Zhang, and G. Hollinger. Multi-uav exploration with limited communication and battery. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2015.

7. M. Cummins and P. Newman. Appearance-only slam at large scale with FAB-MAP 2.0. *The International Journal of Robotics Research*, 30(9):1100–1123, 2011.
8. H. J. S. Feder, J. J. Leonard, and C. M. Smith. Adaptive mobile robot navigation and mapping. *The International Journal of Robotics Research, Special Issue on Field and Service Robotics*, 18(7):650–668, July 1999.
9. M. Giamou, Y. Babich, G. Habibi, and J. How. Stable laser interest point selection for place recognition in a forest. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 2017.
10. M. Himstedt, J. Frost, S. Hellbach, H.-J. Böhme, and E. Maehle. Large scale place recognition in 2D lidar scans using geometrical landmark relations. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 2014.
11. Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: an efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.
12. M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2011.
13. F. Kallasi and D. L. Rizzini. 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)*, 2016.
14. Antero Kukko, Risto Kaijaluoto, Harri Kaartinen, Ville V. Lehtola, Anttoni Jaakkola, and Juha Hyppä. Graph slam correction for single scanner mls forest data under boreal forest canopy. *ISPRS Journal of Photogrammetry and Remote Sensing*, 132:199 – 209, 2017.
15. Lorenz Meier, Petri Tanskanen, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer, and Marc Pollefeys. Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision. *Autonomous Robots*, 33(1-2):21–39, 2012.
16. Daniel Mellinger, Nathan Michael, and Vijay Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. *The International Journal of Robotics Research*, 31(5):664–674, 2012.
17. M. Milford. Visual route recognition with a handful of bits. In *Proceedings of the Robotics: Science and Systems (RSS)*, 2012.
18. R. Paul and P. Newman. FAB-MAP 3D: Topological mapping with spatial and visual appearance. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2010.
19. Charles Richter and Nicholas Roy. Learning to plan for visibility in navigation of unknown environments. In *Proceedings of the International Symposium on Experimental Robotics (ISER)*, 2016.
20. C. Stachniss and W. Burgard. Exploring unknown environments with mobile robots using coverage maps. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, 2003.
21. Russ Tedrake. Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems. 2014.
22. G. D. Tipaldi, L. Spinello, and W. Burgard. Geometrical flirt phrases for large scale place recognition in 2D range data. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2013.
23. B. Yamauchi, A. Schultz, and W. Adams. Mobile robot exploration and map building with continuous localization. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 1998.