Evaluating the performance of a lidar-based autonomous flying system for flying inside forest

Aleksi Karhunen[1], Teemu Hakala[1], Väinö Karjalainen[1], Eija Honkavaara[1]

[1] Finnish Geospatial Research Institute in National Land Survey of Finland, Vuorimiehentie 5, 02150 Espoo, Finland - (aleksi.karhunen, teemu.hakala vaino.karjalainen, eija.honkavaara)@maanmittauslaitos.fi

Keywords: Autonomous flying, Forest, lidar, Path planning, Simultaneous localization and mapping, UAV

Abstract

In recent years, flying autonomously with UAVs has been a widely researched topic. While most autonomous flying systems use the Global Navigation Satellite Systems (GNSS) for localization, using GNSS under the forest canopy is not applicable. Thus other sensors such as lidars are needed. The objective of this study was to evaluate the performance of an autonomous flying system for flying inside boreal forests. Two open-sourced algorithms, IPC (Integrated Planning and Control framework) path planner and control algorithm, and LTA-OM (Long-Term Association Lidar-Inertial Odometry and Mapping) simultaneous location and mapping (SLAM) algorithm, were chosen as the base for the custom-built quadrotor system. Livox Mid-360 lidar was used as the sensor. The system was evaluated with a set of extensive experiments. First, a set of simulation experiments were conducted with multiple flights flown in multiple forests with different tree densities and varying target flight velocities. Second, the localization accuracy of LTA-OM was evaluated by measuring the end-point drift from two manually flown real-world flights. Lastly, 33 autonomous real-world flights were performed in two different forest plots with varying levels of difficulty and tree density as well as with varying target flight velocities. Based on real-world experiments, the performance of the system was somewhat promising inside the medium-difficulty forest, but poor inside the dense difficult forest. The success rate of real-world flights was 10/15 inside the medium-difficulty forest and 6/15 inside the difficult forest with a target flight velocity of 1 m/s. The flight distance on the real-world flights was 60 meters.

1. Introduction

Uncrewed aerial vehicles (UAVs) have been a widely researched topic both in academic research and in commercial organizations. Many autonomous UAVs utilize the Global Navigation Satellite Systems (GNSS) for localization. However, inside forests, localizing the UAV using GNSS can be problematic due to multipath effects and signal blockages caused by trees (Schubert et al., 2010). Furthermore, within a forest, the quadrotor must avoid obstacles, such as trees, branches, bushes, and other understory vegetation, which cannot be achieved without additional onboard sensors.

Many solutions have been proposed for flying autonomously in GNSS-denied environments. In recent years, most solutions use either a lidar, a camera, or a combination of both. However, large-scale experiments, especially inside real-world forests, are rarely performed. Often, the experiments are performed mainly inside simulated environments. Moreover, if real-world experiments are performed, the performance of the proposed system is not extensively reported. Often, the real-world flights are conducted in a sparse-looking forest without any mention of forest density. Additionally, the success rate of flights is rarely reported. Since the experiments, which are conducted when a new autonomous flight algorithm is proposed, are lacking, the performance of the algorithms inside forests is unknown.

Therefore, more extensive experiments within dense forests are needed. Karjalainen et al. (Karjalainen et al., 2023, Karjalainen et al., 2024) has performed more extensive experiments with a camera-based autonomous flight algorithm proposed by Zhou et al. (Zhou et al., 2022). 34 flights were performed inside forests with varying levels of difficulties. The success rate of flights varied from 47% inside a difficult forest with the original system to 100% inside a medium-difficulty forest and 87.5% with an improved system. To our knowledge, no such extensive testing scheme has been implemented for flying inside forests for lidar-based autonomous flight systems.

In this work, a set of extensive experiments is conducted with an autonomous flight system. The selected autonomous flying algorithm IPC was proposed by Liu et al. (Liu et al., 2024). The selected localization algorithm LTA-OM was proposed by Zou et al. (Zou et al., 2024). First, the feasibility of IPC was validated with a set of simulated flights inside three different forests with varying difficulties. Second, the feasibility of LTA-OM for localizing the quadrotor inside forests was validated with a manual flight experiment. Lastly, the performance of the whole system was evaluated with 33 real-world flights flown inside two different forests with varying difficulty.

This article is based on most parts on the Master's thesis of the first author Aleksi Karhunen (Karhunen, 2024).

2. Materials and Methods

2.1 Algorithms

The solution proposed by Liu et al. (Liu et al., 2024) was chosen since Liu et al. had performed real-world experiments inside forests. Although the forest density was not reported, and although based on the provided pictures, the test area seemed sparse, the ability to react quickly to new obstacles and the ability to handle dynamic obstacles was deemed promising.

The architecture of the whole system is presented in Figure 1. IPC is divided into frontend and backend. The frontend is responsible for local map construction and reference path searching. Local map construction is triggered every time a point

cloud is received and is based on a simplified version of ROG-Map (Ren et al., 2023). ROG-Map produces a robot-centric uniform grid local voxel map which is moved efficiently without copying any data in memory. In the simplified version of ROG-Map, no raycasting is used but instead, a voxel is marked as an obstacle if a lidar scan has hit the voxel. Liu et al. (Liu et al., 2024) introduced the temporal forgetting mechanism to ROG-Map where an obstacle is considered free if enough time has passed since the voxel was last been hit by a lidar scan. If the difference between the current time and the timestamp of the last hit is larger than the forgetting threshold, the voxel is considered free. The reference path searching is triggered every time a new goal command is obtained or if an obstacle is detected along the old reference path. In reference path searching, A* is used to find the shortest path to the goal and the reference path is formed by pruning away the redundant nodes to form the shortest piecewise reference path.

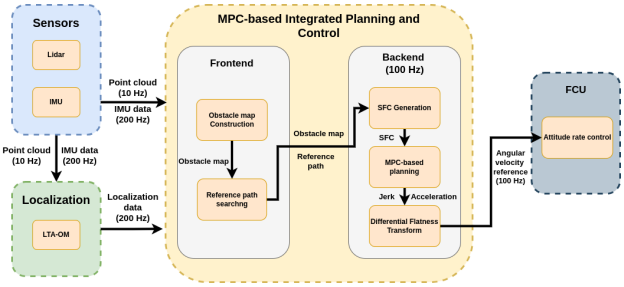


Figure 1. The architecture of the whole autonomous flying system. IPC and LTA-OM receive lidar and IMU data from sensors and IPC receives localization data from LTA-OM at IMU update rate of 200 Hz. IPC outputs angular velocity and thrust references at a rate of 100 Hz to the FCU which is responsible for controlling the quadrotor. Adapted from (Liu et al., 2024).

The backend of IPC is triggered at a constant rate of 100 Hz. First, two safe flight corridors (SFCs) (Liu et al., 2017) are created based on the obstacle map. Then, model predictive control (MPC) and the differential flatness property of quadrotors (Fliess et al., 1995) is used to generate a set of control inputs for the flight control unit (FCU). The MPC problem formulation provided in the source code of IPC by Liu et al. differs slightly from the original given in (Liu et al., 2024) and is formulated as:

$$\begin{aligned}\min_{\mathbf{u}_k} &\sum_{n=1}^{N} ||\mathbf{u}_{n-1}||^2_{\mathbf{R}_u} + \sum_{n=1}^{N-1}(||(\mathbf{p}_{\text{ref},n} - \mathbf{p}_n)||^2_{\mathbf{R}_p} \\ &+ ||(\mathbf{v}_{\text{ref},n} - \mathbf{v}_n)||^2_{\mathbf{R}_v} + ||\mathbf{a}_n||^2_{\mathbf{R}_a}) + ||(\mathbf{p}_{\text{ref},N} - \mathbf{p}_N)||^2_{\mathbf{R}_{p,N}}, \\ &+ ||\mathbf{v}_N||^2_{\mathbf{R}_{v,N}} + ||\mathbf{a}_N||^2_{\mathbf{R}_{a,N}} + \sum_{n=0}^{N-2} ||\mathbf{u}_{n+1} - \mathbf{u}_n||^2_{\mathbf{R}_c}\end{aligned} \tag{1}$$

where $||\mathbf{u}_{n-1}||^2_{\mathbf{R}_u}$ is the control efforts, $||(\mathbf{p}_{\text{ref},n} - \mathbf{p}_n)||^2_{\mathbf{R}_p} + ||(\mathbf{v}_{\text{ref},n} - \mathbf{v}_n)||^2_{\mathbf{R}_v} + ||\mathbf{a}_n||^2_{\mathbf{R}_a}$ are the reference path, velocity, and acceleration following error at reference positions $[\mathbf{p}_{\text{ref},1}, \ldots, \mathbf{p}_{\text{ref},N-1}]$ respectively, $||(\mathbf{p}_{\text{ref},N} - \mathbf{p}_N)||^2_{\mathbf{R}_{p,N}} + ||\mathbf{v}_N||^2_{\mathbf{R}_{v,N}} + ||\mathbf{a}_N||^2_{\mathbf{R}_{a,N}}$ are the reference path, velocity, and acceleration error at the reference position of the horizon length $\mathbf{p}_{\text{ref},N}$ respectively, and $||\mathbf{u}_{n+1} - \mathbf{u}_n||^2_{\mathbf{R}_c}$ is the control variation. The MPC problem is practically the same as presented in (Liu et al., 2024), but with greater flexibility in parameter selection. Using Equation 1 allows for setting different weights for the position, velocity, and acceleration error for the intermediate reference indexes and the last index. The two SFCs are used as the safe space constraint for the MPC problem. Using the differential flatness property, the set of jerk instructions generated from the MPC problem is transformed into a set of attitude commands and forwarded to PX4 flight controller software.

LTA-OM (Zou et al., 2024) was selected as the localization module based on the localization feasibility study presented in section 3.2. LTA-OM utilizes a variant of FAST-LIO2 (Xu et al., 2022) as the odometry module. The pose graph optimization (PGO) problem consists of nodes and constraints. Sequential nodes are constrained via odometry and adjacent keypoint factors, and non-sequential nodes are constrained by virtual loop closure keypoint factors. STD-LCD (Yuan et al., 2023) is used as the loop closure detection module. Loop closure detection is performed on every keyframe which accumulates all points from all previous n keyframes. From every keyframe, triangle descriptors are formed which are used to detect loop closure in a hierarchical manner. First, a hash table is used to form rough loop closure candidates. If the rough loop closures are geometrically consistent the keyframe is considered as a fine loop candidate. Lastly, from all the fine loop closure candidate keyframes, the one with the highest plane overlap ratio between the current frame and the candidate frame is selected as the final loop closure candidate. A false positive rejection module is activated before inserting the loop closure to the graph. If the loop closure passes the consistency check, the loop closure is added to the graph. GTSAM (Frank Dellaert and GTSAM Contributors, 2022) is used to solve the PGO problem.

Since odometry was published only at a lidar rate in the original source code of LTA-OM, the source code was modified to publish odometry data at the IMU rate. To avoid major restructuring of the source code, the inter-lidar odometry estimates were obtained by propagating the IMU measurements with a second-order integrator. The whole system operates on the Robot Operating System (ROS noetic).

2.2 Used Simulation Software and Hardware

The simulation experiments were conducted with the Gazebo physics simulator (Koenig and Howard, 2004) with PX4 software in the loop (SITL). The quadrotor used was a modified version of the Iris quadrotor provided in the PX4 SITL software. The quadrotor was modified to more closely represent the custom-built quadrotor used in real-world experiments by modifying the geometry of the collision and the weight and inertia parameters of the model. The weight of the simulated quadrotor with the lidar was set to be 1.8 kg. The Livox Mid-360 lidar used in the real-world experiments was modeled using an unofficial Mid-360 plugin (Vultaggio et al., 2023) which, inter alia, fixes the distortion of the point cloud which is present in the official Mid-360 plugin.

The real-world experiments were conducted with custom-built quadrotor hardware. The custom quadrotor was built into a quadrotor frame with a motor-to-motor distance of 350 mm. The prop length of the two-bladed propellers was 17.8 cm (8 inches). The onboard computer of the system was Minisforum EM780 with AMD Ryzen 7 7840U CPU, which weighed 268 grams. Pixhawk 6c Mini was used as the FCU. Livox Mid-360 and its built-in IMU, ICM40609, were used for SLAM and autonomous flight. Livox Mid-360 outputs 200 000 points a second in a non-repeating manner. The field of view of the lidar is 360° horizontally, 7° downwards, and 52° upwards. The

IMU outputs data at 200 Hz. The Livox Mid-360 weighed 265 grams. The whole system was powered with a 10 Ah battery. The weight of the whole system was 1875 grams with the battery and 1245 grams without the battery. The whole quadrotor system is shown in Figure 2.

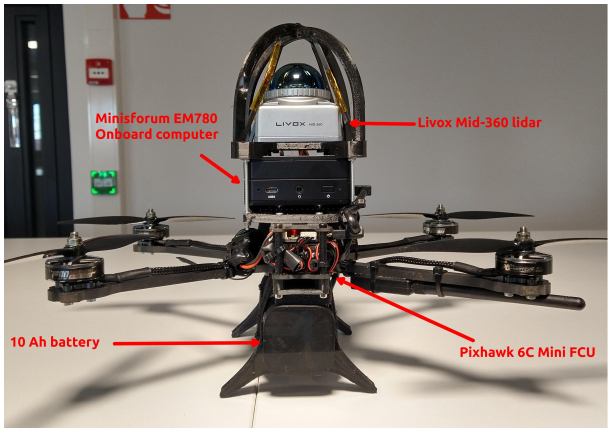


Figure 2. The custom-built quadrotor used in the real-world experiments.

2.3 Test Environments and Experimental Setup

The performance of IPC was first evaluated inside a simulated forest. To test the impact of different forest densities a total of 12 forests were generated. Four of the forests had a tree density of 0.1 trees/m², four forests had a tree density of 0.15 trees/m², and four forests had a tree density of 0.2 trees/m². The forest densities were chosen based on the complexity categories of the boreal forest introduced by Liang et al. (Liang et al., 2019), where easy forests have a density of 0.07 trees/m² with minimal understory vegetation, medium forests have a density of 0.1 trees/m² with sparse understory vegetation and difficult forests have a density of 0.2 trees/m² with dense understory vegetation. No understory vegetation was present in the simulated forests. The forests were generated by using a script by Oleynikova et al. (Oleynikova et al., 2016) where trees are placed inside the given area uniformly. A photorealistic Norway spruce model by Globe Plants (Globe Plants Team, 2022) was used.

To decrease the computational burden the forest size was limited to being 20 meters long and 10 meters wide. Additionally, the top of the trees were cut off. The forest was surrounded by walls to prevent the quadrotor from flying around the forest. The ground was completely flat in all the forests. Figure 3 shows an example of the simulated forests.

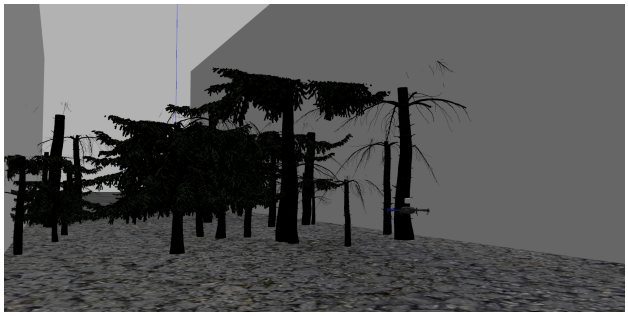


Figure 3. Forest number 4 with a density of 0.15 trees/m² and with the simulated quadrotor hovering in its starting position.

The performance of IPC was evaluated by flying five flights in each of the simulated forests with four different target velocities: 1.0 m/s, 3.0 m/s, 5.0 m/s, and 6.0 m/s. In total, 20 flights were performed inside every forest density with every target flight velocity. A precise position estimate was given to the autonomous flight algorithm during the experiments. The flight distance was 30 meters and the starting and goal localization height was set to 1 meter. The success rate and the true average velocity of the flight were measured. The flight was deemed as a success if the quadrotor managed to reach the goal point, and a failure if there was any collision during the flight.

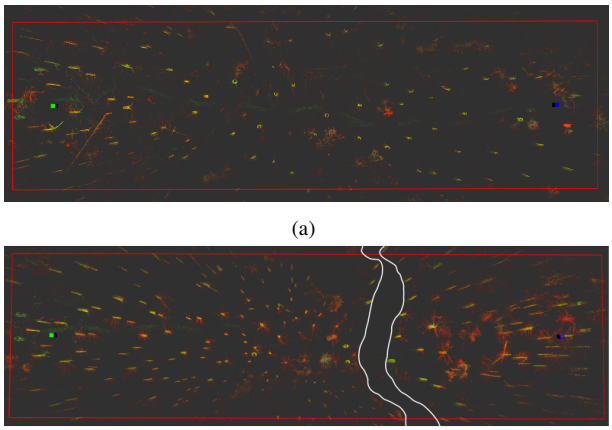
To evaluate the location accuracy of LTA-OM, a real-world manual flight experiment was conducted. The location accuracy of LTA-OM was compared against two other SLAM algorithms: SLOAM proposed by Chen et al. (Chen et al., 2020) and improved by Liu et al. (Liu et al., 2022), and GLIM proposed by Koide et al. (Koide et al., 2024), and one odometry algorithm: FAST-LIO2 proposed by Xu et al. (Xu et al., 2022). Since the Minisforum EM780 does not have a dedicated GPU, a version of GLIM for the CPU was used. This differs from GPU-accelerated GLIM by using VGICP (Koide et al., 2021) as the odometry module as well as a way to extract loop closure detection features. The experiment was conducted by flying the custom-built quadrotor inside a boreal forest in Paloheinä, Helsinki (60°15'28.4"N 24°55'19.9"E). The experiment was conducted by collecting flight data on two manually flown flights. The approximated flight distances of the flights were 150 meters and 420 meters. The starting and end locations of the flights were approximately the same. The end-point drift was measured for all algorithms for both flights.

The performance of the whole system was evaluated with a set of autonomously flown real-world experiments. The experiments were conducted in Paloheinä, Helsinki. The forest plots were classified by using the forest complexity categories by Liang et al. (Liang et al., 2019). The first forest plot was classified as a medium-difficulty forest. The approximated tree density was 0.104 trees/m² with sparse understory vegetation. The second forest plot was classified as a difficult forest. The approximated tree density was 0.222 trees/m² with sparse understory vegetation. Both forest plots consisted of mainly spruces with many dry low-hanging branches. Figure 4 shows the view from the starting location for both forest plots. Figure 5 shows the overhead view of the point clouds obtained from both forest plots.

(a) (b)

Figure 4. View towards the goal from the starting position of the medium-difficulty forest plot (a) and difficult forest plot (b).

In total 33 flights were flown. 15 flights were flown inside both forest plots with a target flight velocity of 1 m/s. Additionally, three flights were flown inside the medium-difficulty forest with a target flight velocity of 2 m/s. The goal location was approximately 60 meters forward from the starting location. The altitude of the starting location and goal was set at 1 meter.



(a)

(b)

Figure 5. Overhead view of the point cloud of the medium-difficulty forest plot (a) and difficult forest plot (b). The green cube depicts the starting position (0, 0, 1) and the blue cube depicts the goal position (60, 0, 1) both given in meters. The exact location of the goal position varied between flights since the initial orientation of the quadrotor varied slightly between the flights. The white lines in (b) depict the edges of a trail, after which the denser end forest starts.

The parameters of IPC were set as follows for the real-world experiments: The obstacle inflation of A* and the SFC shrinking parameter was set to 0.4 m. The horizon length and time step of MPC were set to 15 and 0.1 seconds, respectively. The maximum velocity, acceleration, and jerk were set for all directions to 10 m/s, 20 m/s², and 50 m/s³ respectively except for acceleration downwards which was set to 9.5 m/s². The MPC problem weight parameters of Equation 1 was set as follows: $\mathbf{R}_u = \text{diag}(0, 0, 0)$, $\mathbf{R}_p = \text{diag}(2500, 2500, 2500)$, $\mathbf{R}_v = \text{diag}(0, 0, 0)$, $\mathbf{R}_a = \text{diag}(0, 0, 0)$, $\mathbf{R}_{p,N} = \text{diag}(3500, 3500, 3500)$, $\mathbf{R}_{v,N} = \text{diag}(200, 200, 200)$, $\mathbf{R}_{a,N} = \text{diag}(200, 200, 200)$, and $\mathbf{R}_c = \text{diag}(1.0, 1.0, 1.0)$. The forgetting threshold was set to 30 seconds.

The performance was evaluated by measuring the success rate of flights, flight distance, flight time, point-to-point velocity, and true average velocity of the flights. A flight is considered a success if the system manages to fly to the goal point or if the system deems that the goal is unreachable near the goal point. Otherwise, the flight is deemed a failure. The true average velocity is calculated by taking the average velocity of the smoothed LTA-OM approximated path over the whole flight. The point-to-point average velocity is calculated by dividing the distance between the starting location and the goal location by the flight time. Using the point-to-point average velocity and the true average velocity of the flights, the time spent flying unnecessary long routes can be calculated by:

$$t_{\text{extra}} = \frac{d}{v_{\text{p-to-p}}} - \frac{d}{v_{\text{true}}}, \tag{2}$$

t_{extra} is the time spent flying unnecessary long routes, v_{true} is the true average velocity, $v_{\text{p-to-p}}$ is the point-to-point average velocity, and d is the flight distance. It is good to notice, that the t_{extra} cannot be 0 seconds if there are obstacles between the start and goal location. However, with a perfect system the t_{extra} should be at most a few seconds even in the difficult forests.

Trees/m² \ Target m/s	1.0	3.0	5.0	6.0
0.1	17/20	18/20	13/20	14/20
0.15	17/20	15/20	7/20	9/20
0.2	20/20	15/20	11/20	4/20

Table 1. The success rate of flights depending on the forest tree density and target flight velocity.

Forest \ Target m/s	1.0	3.0	5.0	6.0	All
Density 0.1, forest 1	5/5	5/5	3/5	2/5	15/20
Density 0.1, forest 2	5/5	4/5	3/5	5/5	17/20
Density 0.1, forest 3	4/5	5/5	3/5	2/5	14/20
Density 0.1, forest 4	3/5	3/5	4/5	5/5	15/20
Density 0.15, forest 1	5/5	5/5	0/5	2/5	12/20
Density 0.15, forest 2	4/5	5/5	5/5	4/5	**18/20**
Density 0.15, forest 3	5/5	4/5	2/5	3/5	14/20
Density 0.15, forest 4	3/5	1/5	0/5	0/5	**4/20**
Density 0.2, forest 1	5/5	4/5	2/5	2/5	13/20
Density 0.2, forest 2	5/5	4/5	4/5	2/5	15/20
Density 0.2, forest 3	5/5	4/5	4/5	0/5	13/20
Density 0.2, forest 4	5/5	3/5	1/5	0/5	9/20

Table 2. The success rate of flights depending on the individual forest. The forests are divided into three density groups: 0.1 trees/m², 0.15 trees/m², and 0.2 trees/m². The best and worst success rates of flights for all target flight velocities of individual forests are bolded.

3. Results and Discussion

3.1 Simulation Experiment

A summary of the success rates of the flights is presented in Table 1. In general, with lower target flight velocities, the density of the forests did not have an impact on the success rate of flights. However, with target flight velocities of 5.0 m/s and 6.0 m/s, the success rate of the flights decreased, especially with the target flight velocity of 6.0 m/s, where the success rate decreased from 14/20 in the sparsest forest to 4/20 in the densest forest. The success rate was additionally impacted by the target flight velocity. When the target flight velocity was 1.0 m/s or 3.0 m/s the success rate was 15/20 or over across all forest densities. With target flight velocities of 5.0 m/s and 6.0 m/s, the success rate was 14/20 or under across all forest densities, and under 10/20 for half of the forest densities.

The target flight velocities were not reached during the flights. With a target flight velocity of 1.0 m/s, the true average velocity was 0.75 m/s across all flights. With a target flight velocity of 3.0 m/s, the true average velocity was 1.9 m/s. With a target flight velocities of 5.0 m/s and 6.0 m/s, the corresponding true average velocities were 2.7 m/s and 3.0 m/s, respectively.

Table 2 presents the success rate of flights depending on the individual forest. The impact of the individual forest to the success rates of flights was larger than the impact of the forest density in general. With a forest density of 0.1 trees/m² 76.25% of flights were successful, with a forest density of 0.15 trees/m² 60% of flights were successful, and with a forest density of 0.2 trees/m² 62.5% of flights were successful. However, the differences between the success rates across individual forests were higher, especially in the forest density category of 0.15 trees/m². In fact, the easiest (forest 2 with a success rate of 18/20) and most difficult (forest 4 with a success rate of 4/20) forests had

a density of 0.15 trees/m². Thus based on the simulation experiments, it can be concluded that the configuration of the individual tree clusters has a larger impact on the success rate than the forest density in general. Of course, the denser the forest, the more likely it becomes that there will be a cluster of trees which is difficult to traverse through for IPC.

3.2 Manual Flight Experiment

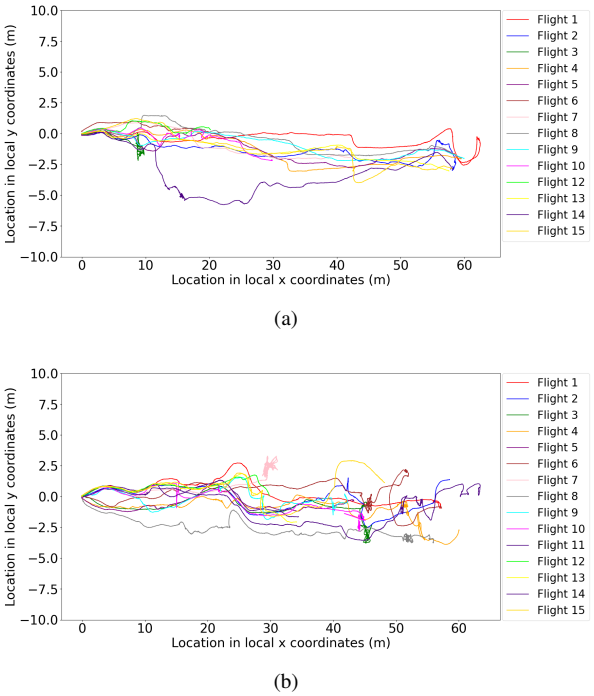
To evaluate the location accuracy of LTA-OM, the end-point drift from two manually flown flights was measured. The results are presented in Table 3 for the first 150-meter flight and in Table 4 for the second 420-meter flight. The tree detection of SLOAM failed for all flights due to the non-repetitive and non-gravity-aligned point cloud of the Livox Mid-360 lidar. Due to the backend failure, SLAOM fell back to using only the odometry module. LTA-OM and FAST-LIO2 achieved similar results across both flights while the location accuracy of GLIM was poor, especially on the longer second flight. Although FAST-LIO2 achieved slightly better results than LTA-OM across both flights, the difference is inside the measurement error. Moreover, for very long flights LTA-OM should achieve lower drift due to loop closure detection. A drift of only a couple tens of centimeters across both flights indicates that LTA-OM is well suited for autonomous flights.

3.3 Real-world Experiments

Table 5 presents an overview of all real-world flights. Figure 6 presents the LTA-OM approximated flight paths from the flights performed in both forest plots. Since the flight paths are given in the local coordinate system and since the starting location and orientation varied slightly in-between flights, the same location on the graph does not necessary correspond to the same location in the test area. The success rate of the system was somewhat promising inside the medium-difficulty forest with a target flight velocity of 1 m/s. 10/15 flights were successful. However, the flight performance inside the difficult forest was poor. Only 6/15 flights were successful. Table 6 presents the failure reasons for flights flown inside different forest plots with different target flight velocities. Only three flights were performed with the target flight velocity of 2 m/s inside the medium-difficulty forest.

In a failure due to a cloud of leaves, the quadrotor thrust a cloud of leaves from the ground with its propellers which were detected as obstacles by the system. This led the system to think that it was located inside an obstacle which caused the planner to move the starting location for the A* search to the closest free voxel in the obstacle map. This sometimes led to very aggressive control maneuvers which sometimes led to an MPC solver failure. This in turn sometimes led to a collision with the ground or with the surrounding obstacles. In a failure due to a collision with a tree, the system steered the quadrotor too close to a tree which resulted in a crash. In a failure due to ”not a numbers” (NaNs) being generated during the SFC generation process, the corners of the SFC polyhedron were defined as NaNs. This led to MPC generating NaN control inputs which turned off the motors. In unstable flying, too aggressively planned paths led to a prolonged period of MPC solver failure, which sometimes led to a crash or to the quadrotor shooting up toward the sky.

For the flights performed inside the medium-difficulty forest with the target flight velocity of 1 m/s, the reasons for failure were divided equally. The most common reason was a cloud of leaves being thrust from the ground. It is good to note that



(a)

(b)

Figure 6. The approximated flight paths of all flight performed inside the medium-difficulty forest (a) and difficult forest (b). The rosbag record of flight 11 inside the medium-difficulty forest was lost and thus omitted from (a). The rosbag recording of flight 7 inside the difficult forest was cut short, but the terminal point of that flight is very close to the one depicted in the graph (b).

most times a cloud of leaves being thrust from the ground did not not lead to a failure. Out of the 10 successful flights on five of them, a cloud of leaves was thrust alongside the quadrotor which did not lead to a failure. For the flights performed inside the difficult forest with the target flight velocity of 2 m/s, failure due to NaNs being generated by the SFC module was the most numerous one, followed by collision with a tree and unstable flying. Due to the ground being frozen, no leaves were thrust from the ground by the quadrotor.

For the flights inside the medium-difficulty forest, the difference between the average true average velocity and point-to-point average velocity was moderate, resulting, on average, 9.6 seconds of lost time over the 60-meter-long flight. However, the difference was much larger in the difficult forest. On average, the system lost 50.6 seconds over the 60-meter-long flight. The most prominent reason was unstable route planning. Since the A* module tries to always find the shortest path towards the goal, after replanning, two subsequent routes might be very different. An example of how different subsequently planned reference paths can be is shown in Figure 7. When this behavior repeats multiple times every few seconds the system starts to fly up-and-down or left-to-right in place for long periods. These periods usually last for 5-20 seconds, however in extreme cases such periods might last much longer. For example on flight seven inside the difficult forest, one in place flying period lasted for over six minutes. The locations of these in place flying periods can be seen in Figure 6 as tight zigzag in the position estimates.

All flights with a target flight velocity of 2 m/s failed. All in

Algorithm	Error x (m)	Error y (m)	Error z (m)	Total (m)
LTA-OM	-0.08	-0.10	0.15	0.19
GLIM	-0.07	0.25	-0.11	0.28
FAST-LIO2	-0.03	0.01	-0.13	0.13

Table 3. End-point error and total distance from end-point of tested SLAM algorithms of the first 150-meter flight. SLOAM was omitted from the table due to backend failure.

Algorithm	Error x (m)	Error y (m)	Error z (m)	Total (m)
LTA-OM	-0.01	-0.08	-0.02	0.08
GLIM	-0.28	1.00	0.36	1.10
FAST-LIO2	-0.00	-0.05	0.02	0.06

Table 4. End-point error and total distance from end-point of tested SLAM algorithms of the second 420-meter flight. SLOAM was omitted from the table due to backend failure.

Forest difficulty and target velocity	Success rate	Average point-to-point average velocity	Average true average velocity	Average t_{extra}
Med. forest, 1 m/s	10/15	0.68 m/s	0.76 m/s	9.3 s
Dif. forest, 1 m/s	6/15	0.44 m/s	0.70 m/s	50.6 s
Med. forest, 2 m/s	0/3	-	-	-

Table 5. An overview of all real-world flights. t_{extra} from Equation 2 is calculated by using the average point-to-point average velocity and true average velocity. The average true average velocity and point-to-point average velocity were only calculated for the successful flights.

Forest difficulty and target velocity	Hit a tree	Cloud of leaves	NaN SFC	Unstable flying	Total
Med. forest, 1 m/s	1	2	1	1	5
Dif. forest, 1 m/s	3	0	4	2	9
Med. forest, 2 m/s*	0	0	0	3	3*

Table 6. The failure reasons for flights flown inside different forest plots with different target flight velocities. Only three flights were performed inside the medium-difficulty forest with a target flight velocity of 2 m/s.

all, flying was very unstable right from the start. On all flights, the flight ended in a crash after approximately 15 meters due to constant and volatile A* reference path replanning which led to a long period of MPC failure.

3.4 Discussion

The system showed promising performance inside the simulator, but based on the real-world experiments the simulation experiments showed over-promising results. While based on the simulation experiment flights even with a target flight velocity of 6 m/s could be possible, based on the real-world experiments performing flights with a target flight velocity of 2 m/s or over would be unadvisable.

Based on the real-world experiment, the system showed encouraging performance, especially in the medium-difficulty forest. However, in the difficult forest environment, the performance was poor with under half of the flights being successful. Based on the estimated time spent flying unnecessary routes t_{extra}, reducing the time spent flying in place would reduce the flight times drastically, especially in more difficult forests. Decreasing the volatility of the A* module by for example making it follow the old reference path should decrease the time spent flying in place during flight. More stable reference path creation should also improve the general stability of the system which should lead to fewer failures due to extended MPC failure or due to a collision with a tree. Additionally, handling NaNs during the SFC generation should improve the reliability of the system. To be considered as a viable option for data collection inside forests, the reliability of the system should be improved and the flight times missions should be decreased.

4. Conclusions

In this study, the performance of a lidar-based autonomous flying system for flying a quadrotor inside boreal forests was evaluated. The system was built upon the IPC path planner proposed by Liu et al. and the LTA-OM SLAM proposed by Zou et al. The performance of the system was evaluated inside a set of simulated and real-world forests with varying levels of difficulty. Additionally, the location accuracy of LTA-OM was evaluated by flying two long manual flights and measuring the end-point drift. The system showed good performance inside the simulator, but the results of the real-world experiments were not as promising. Based on the real-world experiments, the system showed promising performance for the medium-difficulty forest, but poor performance for the difficult forest. However, with modifications to the system, the performance could be improved drastically. By making the A* reference path planning less volatile, and by introducing a way to handle NaNs during the SFC generation process should improve the reliability of the system as well as reduce the flight mission completion times. The end-point location accuracy of LTA-OM was in the range

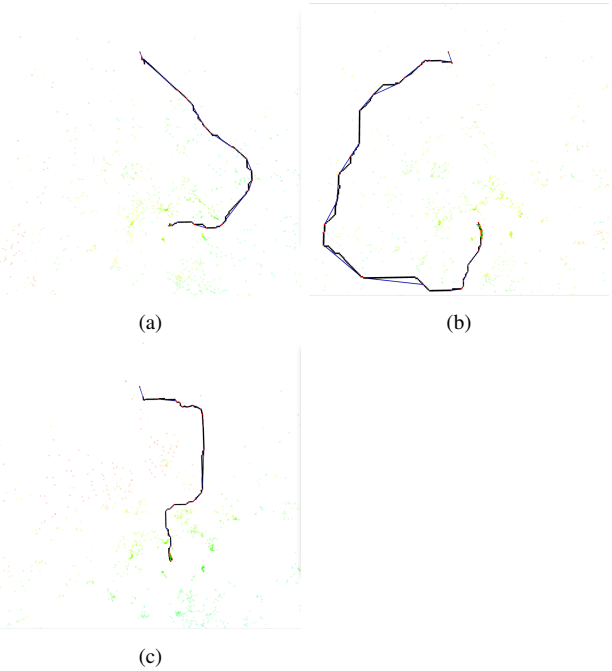


Figure 7. An example of volatile A* reference path planning from a flight done in a difficult forest. The black line depicts the optimal planned A* path and the blue line depicts the outputted reference path. The time interval between the first picture (a) and the last picture (c) was approximately 1.2 seconds. The distance from the current position of the quadrotor to the goal point was approximately 15 meters.

of a couple of tens of centimeters on flights of a few hundred meters.

The capability to fly autonomously within forest environments could simplify laborious forest data collection. The system could be used for example for stem measurements of trees, collection of understory point cloud or image data, or for bark beetle infestation detection to mention a few potential applications.

Acknowledgements

This research was funded by the Academy of Finland within project "Autonomous drone solutions for single tree-based forest management" (decision no. 359404). This study has been performed with affiliation to the Academy of Finland Flagship Forest–Human–Machine Interplay—Building Resilience, Redefining Value Networks and Enabling Meaningful Experiences (UNITE) (decision no. 357908).

References

Chen, S. W., Nardari, G. V., Lee, E. S., Qu, C., Liu, X., Romero, R. A. F., Kumar, V., 2020. SLOAM: Semantic Lidar Odometry and Mapping for Forest Inventory. *IEEE Robotics and Automation Letters*, 5(2), 612-619.

Fliess, M., Lévine, J., Martin, P., Pierre, R., 1995. Flatness and defect of non-linear systems: introductory theory and examples. *International Journal of Control*, 61(6), 1327–1361.

Frank Dellaert and GTSAM Contributors, 2022. borglab/gtsam. https://github.com/borglab/gtsam.

Globe Plants Team, 2022. Picea Abies - Norway Spruce (3D Model). https://globeplants.com/products/picea-abies-norway-spruce-3d-model.

Karhunen, A., 2024. Evaluating the applicability of autonomous lidar-based flying algorithms for flying with uncrewed aerial vehicles inside forests. Master's thesis, School of Electrical Engineering, Aalto University, Espoo, Finland.

Karjalainen, V., Hakala, T., George, A., Koivumäki, N., Suomalainen, J., Honkavaara, E., 2023. A DRONE SYSTEM FOR AUTONOMOUS MAPPING FLIGHTS INSIDE A FOREST – A FEASIBILITY STUDY AND FIRST RESULTS. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-1/W2-2023, 597–603.

Karjalainen, V., Koivumäki, N., Hakala, T., George, A., Muhojoki, J., Hyyppa, E., Suomalainen, J., Honkavaara, E., 2024. Autonomous robotic drone system for mapping forest interiors. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-2-2024, 167–172.

Koenig, N., Howard, A., 2004. Design and use paradigms for gazebo, an open-source multi-robot simulator. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, 3, 2149–2154 vol.3.

Koide, K., Yokozuka, M., Oishi, S., Banno, A., 2021. Voxelized gicp for fast and accurate 3d point cloud registration. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 11054–11059.

Koide, K., Yokozuka, M., Oishi, S., Banno, A., 2024. GLIM: 3D range-inertial localization and mapping with GPU-accelerated scan matching factors. *Robotics and Autonomous Systems*, 179, 104750.

Liang, X., Wang, Y., Pyörälä, J., Lehtomäki, M., Yu, X., Kaartinen, H., Kukko, A., Honkavaara, E., Issaoui, A. E., Nevalainen, O. et al., 2019. Forest in situ observations using unmanned aerial vehicle as an alternative of terrestrial measurements. *Forest ecosystems*, 6, 1–16.

Liu, S., Watterson, M., Mohta, K., Sun, K., Bhattacharya, S., Taylor, C. J., Kumar, V., 2017. Planning Dynamically Feasible Trajectories for Quadrotors Using Safe Flight Corridors in 3-D Complex Environments. *IEEE Robotics and Automation Letters*, 2(3), 1688-1695.

Liu, W., Ren, Y., Zhang, F., 2024. Integrated Planning and Control for Quadrotor Navigation in Presence of Suddenly Appearing Objects and Disturbances. *IEEE Robotics and Automation Letters*, 9(1), 899-906.

Liu, X., Nardari, G. V., Ojeda, F. C., Tao, Y., Zhou, A., Donnelly, T., Qu, C., Chen, S. W., Romero, R. A. F., Taylor, C. J., Kumar, V., 2022. Large-Scale Autonomous Flight With Real-Time Semantic SLAM Under Dense Forest Canopy. *IEEE Robotics and Automation Letters*, 7(2), 5512-5519.

Oleynikova, H., Burri, M., Taylor, Z., Nieto, J., Siegwart, R., Galceran, E., 2016. Continuous-time trajectory optimization for online uav replanning. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5332–5339.

Ren, Y., Cai, Y., Zhu, F., Liang, S., Zhang, F., 2023. ROG-Map: An Efficient Robocentric Occupancy Grid Map for Large-scene and High-resolution LiDAR-based Motion Planning. *arXiv preprint arXiv:2302.14819*.

Schubert, F. M., Fleury, B. H., Robertson, P., Prieto-Cerdeirai, R., Steingass, A., Lehner, A., 2010. Modeling of multipath propagation components caused by trees and forests. *Proceedings of the Fourth European Conference on Antennas and Propagation*, 1–5.

Vultaggio, F., d'Apolito, F., Sulzbachner, C., Fanta-Jende, P., 2023. SIMULATION OF LOW-COST MEMS-LIDAR AND ANALYSIS OF ITS EFFECT ON THE PERFORMANCES OF STATE-OF-THE-ART SLAMS. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-1/W1-2023, 539–545.

Xu, W., Cai, Y., He, D., Lin, J., Zhang, F., 2022. FAST-LIO2: Fast Direct LiDAR-Inertial Odometry. *IEEE Transactions on Robotics*, 38(4), 2053-2073.

Yuan, C., Lin, J., Zou, Z., Hong, X., Zhang, F., 2023. Std: Stable triangle descriptor for 3d place recognition. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 1897–1903.

Zhou, X., Wen, X., Wang, Z., Gao, Y., Li, H., Wang, Q., Yang, T., Lu, H., Cao, Y., Xu, C., Gao, F., 2022. Swarm of micro flying robots in the wild. *Science Robotics*, 7(66), eabm5954.

Zou, Z., Yuan, C., Xu, W., Li, H., Zhou, S., Xue, K., Zhang, F., 2024. LTA-OM: Long-term association LiDAR–IMU odometry and mapping. *Journal of Field Robotics*, 41(7), 2455-2474.

https://doi.org/10.5194/isprs-archives-XLVIII-2-W11-2025-153-2025