

# A DRONE SYSTEM FOR AUTONOMOUS MAPPING FLIGHTS INSIDE A FOREST – A FEASIBILITY STUDY AND FIRST RESULTS

V. Karjalainen<sup>1</sup>, T. Hakala<sup>1</sup>, A. George<sup>1</sup>, N. Koivumäki<sup>1</sup>, J. Suomalainen<sup>1</sup>, E. Honkavaara<sup>1\*</sup>

<sup>1</sup>Finnish Geospatial Research Institute in National Land Survey of Finland, Vuorimiehentie 5, 02150 Espoo, Finland  
- (vaino.karjalainen, teemu.hakala, anand.george, niko.koivumaki, juha.suomalainen, eija.honkavaara)nls.fi

**KEY WORDS:** Autonomous flying, Forest, Visual-Inertial Odometry, Mapping, Path Planning.

## ABSTRACT:

In recent years, drones have gained wide popularity in forest research and operational applications. Over the forest canopy, where Global Navigation Satellite Systems (GNSS) are available, the flights are already highly automated. However, under the canopy of dense forests, the flights still need active manual control by a human pilot due to missing GNSS signal and obstacles. The objective of this study was to design and implement a prototype of a drone autonomously flying inside a forest for future boreal forest research purposes by utilizing open-source algorithms. Based on a literature survey, EGO-Planner-v2 with VINS-Fusion localization and stereo-depth camera-based mapping was chosen as the base of the implemented prototype. The algorithms were first tested in a simulator and later a custom drone hardware was built to evaluate the performance and suitability in real boreal forest environments. The evaluation criteria for the performance were the success of the mission, the reliability of the obstacle avoidance, and the accuracy of the localization. Based on the results, the performance of the prototype was promising, but in dense forests, the sensing of small needleless branches and leafless understory vegetation needs to be improved to increase reliability. In a dense spruce forest, nine of 19 test flights were successful, when approximate flight distances varied between 35 m and 80 m. In the longest of those test flights, the error of the VINS-Fusion estimate of the trajectory length was approximately 1 m.

## 1. INTRODUCTION

In recent years, the autonomously flying of drones has been an actively studied topic in both commercial and academic organizations. Most autopilots can fly autonomously in open areas where Global Navigation Satellite Systems (GNSS) are available. However, inside dense forests, the localization of the drone cannot rely on GNSS due to poor quality or missing GNSS signal due to signal blocks or multipath effects caused by trees (Schubert et al., 2010). Furthermore, inside forests, the drone also has to avoid various obstacles such as trees, bushes, branches, and understory vegetation on its path.

Recent literature has presented various solutions for autonomous flying in GNSS-denied environments, and they differ in sensor setup and algorithms. Although many of those solutions have been tested only in simulators or designed to fly in indoor environments, the literature also contains several solutions with open source code, which have been successfully flight tested in forest environments. These open source real-environment tested methods were the focus of this study. The solution proposed by Liu et al. (2022) was targeted purely at large-scale autonomous flights in forest environments. The system used an inertial measurement unit (IMU) and a stereo camera to run visual-inertial-odometry (VIO) and 3D light detection and ranging (LiDAR) to run a semantic simultaneous localization and mapping (SLAM) algorithm, which was used to correct drift in VIO-estimate. The flight tests were performed in a North American pine forest. The solution proposed by Campos-Macías et al. (2021) used a stereo tracking camera to run VIO and a point cloud from an active stereo depth camera to create a voxel map. The solution concentrated on computational efficiency and was able to run in real-time with a low-power CPU. In addition to indoor environments, the system was tested in a sparse mixed

forest. The solution proposed by Loquercio et al. (2021) concentrated on high-speed autonomous flying in unknown environments. The system forwent on maintaining a map, and different sensing, mapping, and planning subtasks were replaced by a single neural network that produced feasible trajectories from depth images. The drone had an active stereo depth camera to produce the depth images, and a stereo tracking camera to run VIO. The system was successfully flight tested in various environments that contained also Central-European mixed forests. The solution proposed by Zhou et al. (2022) was designed for swarm flights. The designed system used only one stereo camera to run both mapping and VIO, and the swarm flights were successfully tested in a dense bamboo forest.

The objective of this study was to design and implement a prototype of a drone flying autonomously under the canopy for forest research purposes in boreal forest environments utilizing open source codes. However, none of the open-source solutions had been tested in dense boreal forest environments. Furthermore, for most of the systems, the reliability of the navigation had not been evaluated either. The system proposed by Loquercio et al. (2021) was the only one whose reliability was evaluated in the original paper by performing the same flight task multiple times. In our prototype, existing open-source algorithms were utilized, and the suitability of those solutions for boreal forest environments was evaluated. The major research questions were the performance and potential of the system in boreal forest environments as well as further development needs. The system was first implemented and tested in a simulator, followed by altogether 39 flights in real forest environments in Finland.

This article is based on the Master's thesis of the first author Väinö Karjalainen (2023).

\* Corresponding author

## 2. MATERIALS AND METHODS

### 2.1 Algorithms

The solution by Zhou et al. (2022), was chosen as the base of the implemented prototype, since it was proven to work with only one stereo camera. That is beneficial, since in drone applications, minimizing both weight and battery consumption is a crucial component. Cameras are typically lighter than LiDARs and their typical power consumption is smaller (Reddy Cenkeramaddi et al., 2020). Even though the solution had been flight tested only in a bamboo forest, the test forest of the original article was also the densest among the found open-source solutions. The planner algorithm was named in Github as EGO-Planner-v2. Even though the EGO-Planner-v2 is a swarm-supporting approach, in this study, a single drone method was implemented. The algorithm is running on top of the Robot Operating System (ROS). The mapping module of EGO-Planner-v2 is based on probabilistic occupancy grid maps (Moravec and Elfes, 1985). The mapping module uses depth images produced by the stereo camera as an input and the system adopts the fixed-sized circular buffers for maintaining the local map as proposed by Usenko et al. (2017). The maps also have a virtual floor and a virtual ceiling. The virtual floor defines the minimum altitude of the grid map, and the virtual ceiling defines the maximum altitude of the grid map. Virtual floor and virtual ceiling can be used for restricting the altitude where the system is allowed to plan paths.

In trajectory planning, EGO-Planner-v2 uses a trajectory representation called MINCO (minimum control) (Wang et al., 2022) designed for differentially flat systems, such as quadrotors (Mellinger and Kumar, 2011). Trajectories are represented as piece-wise polynomial splines, where spatial and temporal parameters are decoupled (Zhou et al., 2022). The trajectories are optimized by minimizing the weighted sum of metrics defined for smoothness and time. The feasibility of the trajectory is forced by deforming the trajectory shape to avoid obstacles and restricting the magnitudes of trajectory velocity, acceleration, and jerk to fulfill the dynamical constraints of the drone. For computational practicality, the trajectory constraints are enforced via integrals of penalty functions with large penalty weights, and integrals are evaluated by a finite sum of equally spaced samples along the timeline. The optimization of time-dependent objectives  $J$  in one piece of the trajectory can be represented in simplified form by

$$\min \sum_x \lambda_x J_x \quad (1)$$

where where  $J_x$  are penalty terms and  $\lambda_x$  are corresponding weights. Subscripts  $x = \{s, t, d, o, u\}$  stand for smoothness, total time, dynamical feasibility, obstacle avoidance, and uniform distribution of constraint evaluation points, respectively. In addition to these, EGO-Planner-v2 has also penalty terms for swarm operations, but those are omitted here. A\* is used as the global path-searching algorithm given as an input for the optimization.

In the original article by Zhou et al. (2022) an open-source VIO-algorithm, VINS-Fusion (Qin et al., 2018, 2019), was used to provide the pose estimate to EGO-Planner-v2. VINS-Fusion used grayscale stereo images and IMU to track the position and orientation of the drone. VINS-Fusion detects Shi-Tomasi corner features (Shi and Tomasi, 1994), and tracks them

between frames with KLT Tracker (Lucas and Kanade, 1981). VINS-Fusion also uses IMU preintegration, i.e. position, velocity, and orientation are integrated from IMU measurements between two frames. VINS-Fusion also adopts an online temporal calibration method for inertial and visual measurements (Qin and Shen, 2018), which can be used for estimating the time offset  $t_d$  between the camera and the IMU online. In the swarm operations of the original article, the drones used also relative distances measured with Ultra Wide Band (UWB) sensors to correct the drift in pose estimate, but with a single drone that function is omitted.

For trajectory tracking, an open-source tracking controller (ZJU FAST Lab Team, 2022) included in the source code of Zhejiang University's drone building course was used. The tracking controller gets setpoints for position, velocity, acceleration, and yaw from the planner, and converts them to attitude and thrust setpoints that are forwarded to PX4 flight controller software (PX4 Community, 2020). The complete system architecture of the prototype is presented in Figure 1.

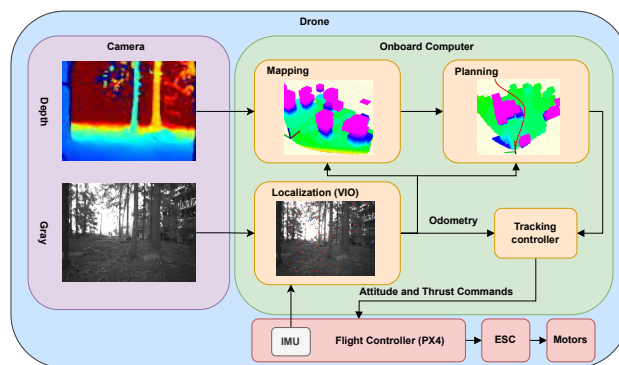


Figure 1. The system architecture used in this study (Karjalainen, 2023). In comparison to the original EGO-Planner-v2 article by Zhou et al. (2022), UWB-based Drift correction and Wireless trajectory broadcast network modules for swarm operations were omitted in this study.

### 2.2 Drones

Simulated tests were performed in Gazebo simulator (Koenig and Howard, 2004) with PX4 running software in the loop (SITL). The simulated drone platform was 3DR Solo, whose Gazebo model was included in the PX4 software. The drone had an approximately 33 cm radius with propellers included. A simulated Intel RealSense D435 camera was attached to the drone. For simulating the D435, a Gazebo ROS plugin Pal Robotics Development Team (2019) published by PAL Robotics was used. The simulated camera was publishing both depth images and grayscale stereo images at a frequency of 30 Hz. The resolution of all images was  $640 \times 480$  pixels. The simulated drone is presented in Figure 2.

In the physical robot drone prototype an onboard computer ASUS PN51-E1 with AMD Ryzen 7 5700U processor was used. The used PX4-compliant autopilot was Hex Cube black, which contained also the IMU. For mapping and VIO, an Intel RealSense D435 camera (Intel Realsense Development Team, 2023)(Keselman et al., 2017) was used. RealSense D435 is based on active stereoscopy, which allows it to output high-quality depth images. The camera was publishing both depth images and grayscale stereo images at a frequency of 30 Hz and the resolution of all images was  $640 \times 480$  pixels. The



Figure 2. The simulated drone.

camera-IMU system was calibrated with an open-source camera calibration toolbox, kalibr (Furgale et al., 2012)(Furgale et al., 2013)(Maye et al., 2013). In the calibration, the intrinsic and extrinsic parameters of the left and right grayscale cameras and the time shift between the camera and the autopilot IMU were calculated from recorded data where Aprilgrid (Olson, 2011) was used as a calibration target. The computer, the camera, and the autopilot were attached to a 330 mm drone frame. The propellers had three pieces of 6.5 cm long blades each. The measured weight of the drone was 1248 g without the batteries, and 1613 g with batteries for the motors and the onboard computer included. An overview of the used drone hardware is presented in Figure 3.

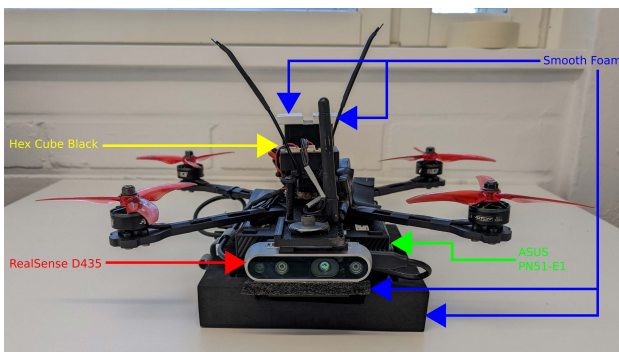


Figure 3. The drone hardware used in the flight tests.

### 2.3 Test environments and experimental setups

The performance of the system was first evaluated in simulated forests. The two main components of the system, obstacle avoidance and VIO, were first evaluated separately in randomized forests with artificial spruce models. The forest generation process was adapted from the script by Oleynikova et al. (2016). As tree models, photorealistic Norway spruce models by Globe Plants were used (Globe Plants Team, 2022). A total of 12 different forests were generated. The tree densities of the forests were 0.1 trees/m<sup>2</sup>, 0.15 trees/m<sup>2</sup>, and 0.2 trees/m<sup>2</sup> so that there were four individual forests per density. An example of a simulated forest with a density of 0.2 trees/m<sup>2</sup> is shown in 4.

The performance of the obstacle avoidance was tested with four different maximum flying velocities: 1.0 m/s, 1.5 m/s, 2.0 m/s, and 2.5 m/s. For every forest and velocity combination five flights were performed, so there were in total 20 flights with every density and velocity combination. In the obstacle avoidance tests, the drone got precise pose estimates from the simulator instead of the VIO. The flight distance was 25 m.

To evaluate the performance of VINS-Fusion during obstacle avoidance, flight data from six flights were recorded. From

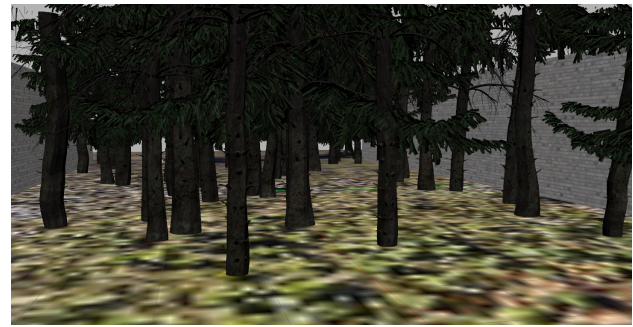


Figure 4. Example of simulated forest with a density of 0.2 trees/m<sup>2</sup>.

those, three flights were performed with a maximum velocity of 1.0 m/s and three flights with 2.0 m/s. Since VIO is not a deterministic process, the estimation result varies between runs even if the input data remains exactly the same. For that reason, VINS-Fusion estimation was performed 20 times for every flight data recording. From those, the estimation was run 10 times by using the online estimation for  $t_d$  and 10 times with a fixed value for  $t_d$ . The used fixed  $t_d$  value was acquired by roughly testing different values in the simulator. At the end of every estimation, the estimated position and orientation were compared against the ground truth values from the simulator.

After successful flights in artificial forests, the system was tested in its entirety in a simulated forest that was generated based on the point cloud data from a real pinewood forest in Finland. The data was processed and the 3D model was created with Agisoft Metashape (Metashape Development Team, 2021). A screenshot from the forest is presented in Figure 5. The purpose of these tests was not only to validate the safe joint operation between VIO and planner but also to evaluate the performance of the system in longer flights with a computationally lighter environment. The goal was 90 m forward and 8 m up from the takeoff location since the flying direction was up the slope. In total 10 test flights were performed, and the maximum flying velocity was set to 2.0 m/s.



Figure 5. Simulated forest generated from real point cloud data.

After validating the safe operation and correct performance of the system in a simulator, its performance and suitability for operating in boreal forest environments were evaluated using real physical hardware. The success of the mission, the reliability of the obstacle avoidance, and the accuracy of the localization were used as evaluation criteria. The performance of the system was tested in three different forest environments in Southern Finland. Tests were carried out between January 2023



and February 2023, under leaf-off conditions, and in different weather conditions, and adjusting the maximum flying speed and the distance to the goal during the tests. The takeoff pose of the drone was defined as the origin of the coordinate system, where the x-axis was pointing forward from the drone, the y-axis was pointing left from the drone, and the z-axis was pointing up from the drone. The navigation goal was then defined by sending three-dimensional coordinates of the goal point to the system.

The first test environment was a sparse mixed forest. In total, nine test flights were performed there, and the flight distances varied between 18 m and 23 m. The maximum flight velocity was 0.5 m/s in the first test flight and 1.0 m/s in the rest of the flights. The weather was sunny, but the sun was not pointing toward the camera. Figure 6 presents the overview of the test environment. The second test environment was a park woodland with dense understory vegetation and uneven ground. Since the tests were performed during the winter, the vegetation was leafless. Again, nine test flights were performed, and the flight distances varied between 21 m and 40 m. The maximum flight velocity varied between 1.0 m/s and 1.5 m/s. The weather was partially cloudy, and on some flights, the sun was pointing toward the camera. An overview of the second test area is presented in Figure 7. The last test environment was a dense and snowy spruce forest. In this environment, 19 test flights were performed and the flight distances varied between 35 m and 80 m. The maximum flight velocity was 1.0 m/s in all test flights. The weather during the tests was partially cloudy, but dense forest blocked the direct sunlight during the sunny moments. For this environment, a rough estimate of the forest density was formed with a backpack laser scanner. In the first part of the flight test area the forest was very dense consisting mainly of young spruces with low branches. The estimated average forest density in that area was approximately 0.238 trees/m<sup>2</sup>. After approximately 35 m the forest had two small open areas, that were used as goal locations in shorter flights. After the open areas, the forest was sparser than in the beginning. The estimated average forest density in that area was approximately 0.165 trees/m<sup>2</sup> consisting mainly of older trees with fewer low branches. An overview of the third test area is presented in Figure 8.

The experiments tested the ability of the system to detect and avoid trees, branches, and understory vegetation. For some of the flights, also the precision and accuracy of the VIO were evaluated by comparing the precision of the landing point in repeated flights and measuring the distance from the takeoff location to the landing point with a laser range meter.

### 3. RESULTS AND DISCUSSION

#### 3.1 Simulation results

The system was first implemented and validated using the Gazebo simulator. Based on the results, increasing the forest density or flying velocity decreased the success rate of obstacle avoidance. With a maximum flying velocity of 1.0 m/s, the system was able to navigate without collisions in forests with a density of 0.15 trees/m<sup>2</sup>. With the highest maximum flying velocity and forest density, 2.5 m/s and 0.2 trees/m<sup>2</sup>, only six of the 20 flights were successful.

In the evaluation of VINS-Fusion, the average position error over all flights was 0.49 m with fixed  $t_d$  and 0.85 m with using built-in online  $t_d$  estimation. In both cases, the average position



Figure 6. Sparse mixed forest.



Figure 7. Drone flying in park woodland with dense leafless understory vegetation.

error was biggest in the x-direction, which was also the direction where most of the moving happened. The average position error in the x-direction was 0.39 m with fixed  $t_d$  and 0.80 m with using online  $t_d$  estimation. The standard deviation between the estimations was also biggest in the x-direction, 0.20 m with fixed  $t_d$  and 0.17 m with using online  $t_d$  estimation. The average orientation error was biggest in the yaw-direction, 0.67° with online estimated  $t_d$  and 0.74° with fixed  $t_d$ .

For the last simulation test flights in the forest model generated from the photogrammetric point cloud data, the  $t_d$  was fixed. The system managed to fly all 10 test flights without collisions. The average position error of the ten flights was 1.32 m. The average orientation error was again highest in the yaw-direction, 0.8°.



Figure 8. Dense snowy spruce forest.

### 3.2 Real forest test results

The prototype performed well in the sparse mixed forest, and out of nine test flights eight were successful. In these cases the drone autonomously calculated its path and avoided any obstacles in its way. In the only failed flight the drone failed to find the path to the goal position and was manually landed from hovering mode.

In the park woodland with dense understory vegetation only three out of nine test flights were successful. The most common reason for the failures was a late detection of small branches of the leafless understory vegetation leading either to collisions or late emergency stops in situations where the planner was unable to find a new path even after the emergency stop. Two flights failed also since the drone considered the given goal position to be under the ground. However, there were also successful flights with identical goal points and takeoff locations, which indicates that there was a deviation between the z-coordinate of VIO-estimations of the flights.

In the dense spruce forest, nine out of 19 test flights were successful. This performance convincingly showed that the system is capable to operate in quite challenging conditions. The most common reason for the failures was again a late or completely missed detection of small dry branches. Figures 9 and 10 illustrate the problem by presenting a left stereo image and depth image respectively approximately a second before a collision with the small branches in front of the drone. As can be seen from the illustration of the local map in Figure 11, the system did not detect the branches and the grid map in front of the drone was clear. In three flights the reason for the failure was the instability of the VIO-estimate, and once the drone tried to go under the ground straight after the takeoff due to poor global path returned by the planner.

Figures 12, 13, and 14 illustrate examples of flight paths in successful autonomous flights in sparse mixed forest, park woodland, and dense spruce forest respectively. The maps and the paths are seen from up, so paths going through obstacles mean that the drone navigated under a low branch.



Figure 9. Left stereo image before a collision with the small branches.

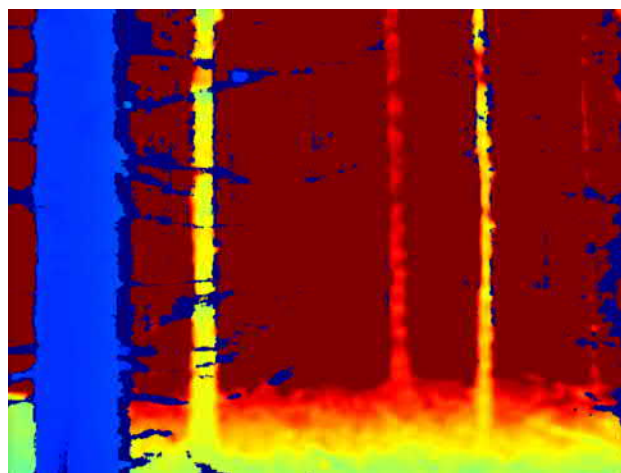


Figure 10. Depth image before a collision with the small branches.

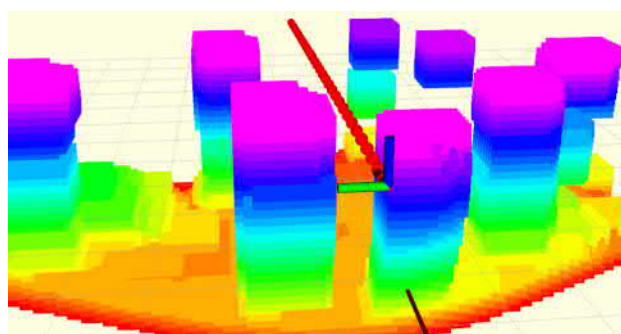


Figure 11. The local map and planned path before a collision with the small branches.

The precision of the VINS-Fusion estimate was roughly evaluated by performing three flights with identical takeoff locations and goals both in sparse mixed forest and dense spruce forest. In the mixed forest, the goal of the three flights was 18 m forward from the takeoff location. The longest distance between the landing points was approximately 0.5 m. In the dense spruce forest, the goal of the three flights was 60 m forward and 1.5 m left from the takeoff location. The longest distance between the



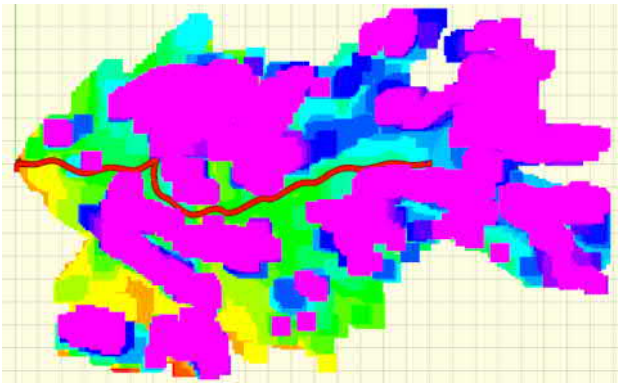


Figure 12. Example of flight path in sparse mixed forest.

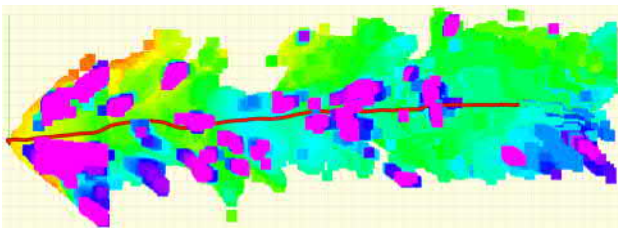


Figure 13. Example of flight path in park woodland.

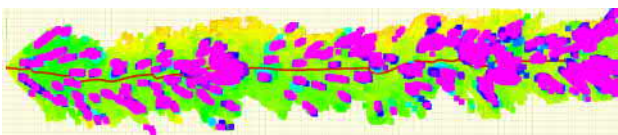


Figure 14. Example of flight path in dense spruce forest.

landing points was approximately 2.2 m. In both cases, the majority of the deviation was in y-direction. Hence the deviation can be caused by differences in initial headings, since in the 60 m long flight even a  $1^\circ$  difference in the initial heading leads to approximately 1 m difference in the position at the end of the flight. A rough estimate of the accuracy of the VINS-Fusion estimate was formed with a laser range meter. In the three sparse mixed forest flights, the range meter reading from the takeoff location to the approximate center of landing points of 18 m flights was exactly 18 m. In the dense spruce forest measuring the distance with a range meter was difficult due to trees and branches in the path. However, in the longest test flight VINS-Fusion estimated the flight distance to be 79.6 m, and the distance measured from the takeoff location to the landing location was 78.6 m.

The performance of the prototype was promising, but development is needed to increase the reliability of obstacle avoidance in dense boreal forests with tiny leafless branches. At the hardware level, improved methods to detect thin branches need to be tested. Using only one stereo camera makes the sensor setup energy efficient and light, but comes with the drawback of limited field-of-view. Using LiDAR or  $360^\circ$  camera would offer a better view of the surrounding trees and branches. Although the evaluation methods for the performance of VINS-Fusion were approximate, its accuracy and precision were satisfying for flights with this distance. However, in the snowy spruce forest, the VINS estimate became unstable in three tests. Two times the instability started straight after the initialization during the takeoff, and only once during the flight. The unstabil-

ity problems were not detected in the two earlier environments, so the possibility of environmental reasons or hardware breaks cannot be excluded. The performance and reliability of VINS-Fusion could also increase by performing a new calibration for the sensor setup or implementing a hardware synchronization between the stereo camera and the IMU.

#### 4. CONCLUSIONS AND FURTHER WORK

In this study, a prototype of a drone flying autonomously under the canopy for forest research purposes in boreal forest environments was designed and implemented. The prototype was built by utilizing existing open-source algorithms, EGO-Planner-v2 and VINS-Fusion, and its suitability and performance were evaluated in various Finnish forest environments. In the best-case scenario, the prototype was able to successfully navigate and avoid obstacles even in dense spruce forests, but the detection of thin branches needs to be improved to achieve reliable and robust flight performance.

The implemented prototype can be used for example for further research on improving the performance of the system and developing applications on top of it. Potential ways to further improve the system are, for example, combining a high-level path planner performing area coverage to the system or improving the VIO estimate with loop closures and GNSS-fusion. Also, the implementation of the full swarm-based solution would be of interest. More testing in different seasons would also be relevant.

The ability to fly inside a forest opens up completely new possibilities to digitalize forest analysis and offer enhanced ecosystem services. Potential future applications of the system in forest research include direct forest inventory measurements inside the forest (e.g. tree species and stem parameters), image data collection of understory vegetation for biodiversity research, analysis of stem quality e.g. to detect damages in the bark of the trees due to insect pests, as well as a search of wild-food such as mushrooms and berries, to mention but a few potential applications.

#### 5. ACKNOWLEDGEMENTS

This research was funded by the Academy of Finland within projects “Finnish UAV Ecosystem” (decision no. 337018) and Fireman (decision no. 346710). 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. 337127).

We also want to deliver acknowledgments to Professor Ville Kyrki from Aalto University’s Department of Electrical Engineering and Automation for supervising the Master’s thesis of the first author Vainö Karjalainen to which this article is based.

#### References

Campos-Macías, L., Aldana-López, R., de la Guardia, R., Parra-Vilchis, J. I., Gómez-Gutiérrez, D., 2021. Autonomous navigation of MAVs in unknown cluttered environments. *Journal of Field Robotics*, 38(2), 307–326. doi.org/10.1002/rob.21959.

- Furgale, P., Barfoot, T. D., Sibley, G., 2012. Continuous-time batch estimation using temporal basis functions. *2012 IEEE International Conference on Robotics and Automation*, 2088–2095. doi.org/10.1109/ICRA.2012.6225005.
- Furgale, P., Rehder, J., Siegwart, R., 2013. Unified temporal and spatial calibration for multi-sensor systems. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1280–1286. doi.org/10.1109/IROS.2013.6696514.
- Globe Plants Team, 2022. Picea Abies - Norway Spruce (3D Model). Globe Plants. <https://globeplants.com/products/picea-abies-norway-spruce-3d-model> (22 December 2022).
- Intel Realsense Development Team, 2023. Depth Camera D435. Intel Corporation. <https://www.intelrealsense.com/depth-camera-d435/>.
- Karjalainen, V., 2023. Vision-based navigation for autonomous drone flights inside a forest. Master's thesis, Aalto University. School of Electrical Engineering. <http://urn.fi/URN:NBN:fi:aalto-202305213281>.
- Keselman, L., Woodfill, J. I., Grunnet-Jepsen, A., Bhowmik, A., 2017. Intel(r) realsense(tm) stereoscopic depth cameras. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1267–1276. doi.org/10.1109/CVPRW.2017.167.
- 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. doi.org/10.1109/IROS.2004.1389727.
- 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. doi.org/10.1109/LRA.2022.3154047.
- Loquercio, A., Kaufmann, E., Ranftl, R., Müller, M., Koltun, V., Scaramuzza, D., 2021. Learning high-speed flight in the wild. *Science Robotics*, 6(59), eabg5810. doi.org/10.1126/scirobotics.abg5810.
- Lucas, B. D., Kanade, T., 1981. An iterative image registration technique with an application to stereo vision. *IJCAI'81: 7th international joint conference on Artificial intelligence*, 2, 674–679. <https://hal.science/hal-03697340/>.
- Maye, J., Furgale, P., Siegwart, R., 2013. Self-supervised calibration for robotic systems. *2013 IEEE Intelligent Vehicles Symposium (IV)*, 473–480. doi.org/10.1109/IVS.2013.6629513.
- Mellinger, D., Kumar, V., 2011. Minimum snap trajectory generation and control for quadrotors. *2011 IEEE International Conference on Robotics and Automation*, 2520–2525. doi.org/10.1109/ICRA.2011.5980409.
- Metashape Development Team, 2021. Agisoft Metashape Software, Version 1.7. Agisoft LLC. <https://www.agisoft.com/downloads/installer/>.
- Moravec, H., Elfes, A., 1985. High resolution maps from wide angle sonar. *Proceedings. 1985 IEEE international conference on robotics and automation*, 2, IEEE, 116–121. doi.org/10.1109/ROBOT.1985.1087316.
- 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. doi.org/10.1109/IROS.2016.7759784.
- Olson, E., 2011. Apriltag: A robust and flexible visual fiducial system. *2011 IEEE International Conference on Robotics and Automation*, 3400–3407. doi.org/10.1109/ICRA.2011.5979561.
- Pal Robotics Development Team, 2019. Intel RealSense Gazebo ROS plugin. PAL Robotics S.L. [https://github.com/pal-robotics/realsense\\_gazebo\\_plugin](https://github.com/pal-robotics/realsense_gazebo_plugin).
- PX4 Community, 2020. PX4-Autopilot, Version 1.11.0. <https://github.com/PX4/PX4-Autopilot>.
- Qin, T., Li, P., Shen, S., 2018. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4), 1004–1020. doi.org/10.1109/TRO.2018.2853729.
- Qin, T., Pan, J., Cao, S., Shen, S., 2019. A general optimization-based framework for local odometry estimation with multiple sensors. *arXiv preprint arXiv:1901.03638*. doi.org/10.48550/arXiv.1901.03638.
- Qin, T., Shen, S., 2018. Online temporal calibration for monocular visual-inertial systems. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3662–3669. doi.org/10.1109/IROS.2018.8593603.
- Reddy Cenkeramaddi, L., Bhatia, J., Jha, A., Kumar Vishkarma, S., Soumya, J., 2020. A survey on sensors for autonomous systems. *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 1182–1187. doi.org/10.1109/ICIEA48937.2020.9248282.
- Schubert, F. M., Fleury, B. H., Robertson, P., Prieto-Cerdeira, 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.
- Shi, J., Tomasi, 1994. Good features to track. *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 593–600. doi.org/10.1109/CVPR.1994.323794.
- Usenko, V., von Stumberg, L., Pangercic, A., Cremers, D., 2017. Real-time trajectory replanning for MAVs using uniform B-splines and a 3D circular buffer. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 215–222. doi.org/10.1109/IROS.2017.8202160.
- Wang, Z., Zhou, X., Xu, C., Gao, F., 2022. Geometrically constrained trajectory optimization for multi-copters. *IEEE Transactions on Robotics*, 38(5), 3259–3278. doi.org/10.1109/TRO.2022.3160022.
- Zhou, X., Wen, X., Wang, Z., Gao, Y., Li, H., Wang, Q., Yang, T., Lu, H., Cao, Y., Xu, C. et al., 2022. Swarm of micro flying robots in the wild. *Science Robotics*, 7(66), eabm5954. doi.org/10.1126/scirobotics.abm5954.
- ZJU FAST Lab Team, 2022. px4ctrl. Zhejiang University. [https://github.com/ZJU-FAST-Lab/Fast-Drone-250/tree/master/src/realflight\\_modules/px4ctrl](https://github.com/ZJU-FAST-Lab/Fast-Drone-250/tree/master/src/realflight_modules/px4ctrl).