

DigiForest: Digital Analytics and Robotics for Sustainable Forestry

Marco Camurri, Enrico Tomelleri, Matías Mattamala, Sebastián Barbas Laina, Martin Jacquet, Jens Behley, Sunni Kanta Prasad Kushwaha, Fang Nan, Nived Chebrolu, Leonard Freißmuth, Marvin Chayton Harms, Meher V.R. Malladi, Fan Yang, Jonas Frey, Cesar Cadena, Marco Hutter, Janine Schweier, Kostas Alexis, Cyril Stachniss, Maurice Fallon, Stefan Leutenegger

Marco Camurri

Dipartimento di Ingegneria Industriale, University of Trento, via Sommarive 9, 38123 Trento (TN), Italy e-mail: `marco.camurri@unitn.it`

Faculty of Engineering, Free University of Bozen-Bolzano, via Bruno Buozi 1, 39100 Bozen-Bolzano (BZ), Italy

Enrico Tomelleri

Faculty of Agricultural, Environmental and Food Sciences, Free University of Bozen-Bolzano, piazza Università 5, 39100 Bozen-Bolzano (BZ), Italy e-mail: `enrico.tomelleri@unibz.it`

Competence Centre for Mountain Innovation Ecosystems, via Bruno Buozi 1, 39100 Bozen-Bolzano (BZ), Italy

Matías Mattamala

School of Informatics, University of Edinburgh, 10 Crichton St, EH8 9AB, United Kingdom e-mail: `matias.mattamala@ed.ac.uk`

Sebastián Barbas Laina, Leonard Freißmuth

Technical University of Munich, School of Computation, Information and Technology Boltzmannstrasse 3, 85748 Garching, Germany e-mail: `{sebastian.barbas,l.freissmuth}@tum.de`

Martin Jacquet, Marvin Chayton Harms, Kostas Alexis

Department of Engineering Cybernetics, Norwegian University of Science and Technology, Elektro D/B2, Gløshaugen, Norway. e-mail: `{martin.jacquet,marvin.c.harms,konstantinos.alexis}@ntnu.no`

Jens Behley, Meher V.R. Malladi, Cyril Stachniss

Center for Robotics, University of Bonn, Nussallee 15, 53115 Bonn, Germany e-mail: `{jens.behley,meher.malladi,cyрил.stachniss}@igg.uni-bonn.de`

Sunni Kanta Prasad Kushwaha, Janine Schweier

Swiss Federal Research Institute WSL, Zürcherstrasse 111, 8903 Birmensdorf, Switzerland e-mail: `{sunni.kushwaha,janine.schweier}@wsl.ch`

Fang Nan, Fan Yang, Jonas Frey, Cesar Cadena, Marco Hutter

Robotic Systems Lab, ETH Zürich, Leonhardstrasse 21, 8092 Zürich, Switzerland e-mail: `{fannan,fanyang1,jonfrey,cesarc,mahutter}@ethz.ch`

Nived Chebrolu, Maurice Fallon

Oxford Robotics Institute, Department of Engineering Science, University of Oxford, 23 Banbury Rd, Oxford, OX2 6NN, United Kingdom e-mail: `{nived,mfallon}@robots.ox.ac.uk`

Stefan Leutenegger, Leonard Freißmuth

Mobile Robotics Lab, ETH Zürich, Department of Mechanical and Process Engineering, Leonhardstrasse 21, 8092 Zürich, Switzerland e-mail: `{lestefan,lfreissmuth}@ethz.ch`

Abstract Covering one third of Earth’s land surface, forests are vital to global biodiversity, climate regulation, and human well-being. In Europe, forests and woodlands reach approximately 40% of land area, and the forestry sector is central to achieving the EU’s climate neutrality and biodiversity goals; these emphasize sustainable forest management, increased use of long-lived wood products, and resilient forest ecosystems. To meet these goals and properly address their inherent challenges, current practices require further innovation. This chapter introduces DigiForest, a novel, large-scale precision forestry approach leveraging digital technologies and autonomous robotics. DigiForest is structured around four main components: (1) autonomous, heterogeneous mobile robots (aerial, legged, and marsupial) for tree-level data collection; (2) automated extraction of tree traits to build forest inventories; (3) a Decision Support System (DSS) for forecasting forest growth and supporting decision-making; and (4) low-impact selective logging using purpose-built autonomous harvesters. These technologies have been extensively validated in real-world conditions in several locations, including forests in Finland, the UK, and Switzerland.

1 Introduction

Forests cover nearly one third of the Earth’s land surface and are essential for biodiversity, climate regulation, and several other ecosystemic services. In Europe the share is even higher (close to 40%) and, as outlined by the EU Forest Strategy for 2030 [10], forestry has a key role in reaching EU’s climate neutrality and biodiversity goals, calling for more efficient and sustainable forest management.

Forest management practices differ from country to country, but generally follow a four-stage cycle. The first is surveying: given a forest plot or stand of interests, foresters record absolute position, Diameter at Breast Height (DBH), height, species, and health status of each tree, as well as additional traits of specific interest. Typically, collection of such quantities is performed manually in the field with portable tools (e.g., calipers for DBH and laser range finders for height) [22]. These surveys are generally accurate, but are subject to human operator’s ability, slow to carry out, and sparse (only limited areas are covered). More recently, techniques such as terrestrial laser scanning (TLS) [12], mobile laser scanning (MLS) [1] [33], and aerial laser scanning (ALS) are used to collect detailed 3D information on tree and stand structure. MLS has been demonstrated to be superior in tree estimation such as DBH, as coverage per unit time outweighs sampling accuracy [1].

The second stage involves the aggregation of the data collected to create (or update) a forest inventory. A forest inventory keeps track of all the relevant traits required for forest planning, such as DBHs, species and health. Inventories are made on-site during the data collection itself, or offline by means of specialized software to extract tree traits from point clouds [6], also aggregating longitudinal information.

The third stage is planning. Based on the inventory, foresters decide which interventions are more suitable for a stand, e.g., thinning, harvesting, or other measures to promote regeneration. Depending on the application, the aim may be

to stimulate growth, reduce competition, increase structural diversity, or secure timber production. Forest planning has traditionally relied on expert knowledge and paper records, but modern planning systems, known as Decision Support Systems (DSS), integrate computational models to estimate growth and production in different scenarios.

The fourth and final stage is carrying out forest interventions, which are determined by the planning step. Current forestry, specially in European countries, is shifting away from timber production via clear cuts in favor of continuous cover forestry. This approach aims to preserve ecological diversity while still targeting the production goals. Such effort requires the coordination of different activities such as thinning, pruning, planting, tending regeneration, apart from protecting against fire, browsing, and pests.

Among these, thinning is a silvicultural intervention in which a portion of trees is periodically removed to reduce competition and promote the growth, stability, and quality of the residual stand. By favoring well-formed trees and reducing density, thinning enhances the value of the timber, increases the resilience to biotic and abiotic stresses, and can improve biodiversity through greater structural heterogeneity. In European managed forests, thinning is typically scheduled at several stages during a rotation and adapted to site conditions and management objectives. Because the whole cycle is time and resource intensive, management usually runs on a five to ten year cycle.

With climate change and rising frequency and intensity of disturbance [39][45], there is growing pressure to shorten these cycles and make forest management more flexible. While modern remote sensing techniques provide large-scale aerial forest data to improve decision-making, over-canopy measurements lack the fidelity to improve current growth and production models. Therefore, there is an urging need for precision forestry solutions, which prioritize collecting fine-grained, localized under-canopy data while employing technological solutions that minimize the environmental impacts of this industry.

In this work, we present an innovative approach towards large-scale automated precision forestry based on digital analysis and robotic technologies developed in the context of the DigiForest EU Horizon project¹. An overview of DigiForest is shown in Fig. 1. Inspired by the four stages of current forestry practices, the approach is centered around four main components:

1. A detailed, tree-level data collection pipeline triggered by a human operator and executed by autonomous heterogeneous mobile robots. In addition to above-canopy ALS data acquisition, we focus on automated ground-level scanning by means of legged robots and small scale drones. While wheeled platforms have been used for this task in the past [41] [45], legged robots are now a mature technology promising superior mobility and reduced soil impact than wheeled and tracked vehicles. This technology is complemented by autonomous undercanopy aerial drones, which can effectively complement legged platforms in ground data acquisition as they can reach dense undergrowth areas but have more limited battery life. The

¹ <https://digiforest.eu/>

platforms exploit their sensing capabilities to reconstruct the forest via SLAM online or by communicating the data in a standardized format [8] to a map server running multi-session, multi-platform SLAM.

2. An automated tree trait extraction and online forest inventory for immediate report to the user in the field, or more detailed LiDAR processing capable of identifying trees as well as its constitutive components via panoptic segmentation. In contrast to most MLS or TLS based systems, the inventory is available online during the mission. The inventory system was demonstrated on the legged robots which were able to cover 1 ha in 20-30 min with DBH estimation accuracy of 2-3 cm.
3. A Decision Support System (DSS) performing growth forecast from forest inventory data, to support operator decision making.
4. Automated selective logging/cutting with small footprint, purpose-built, hydraulic harvesting machine. In conventional thinning, harvesting is a two-step cycle: (i) felling and processing trees along extraction trails, and (ii) forwarding the assortments to the roadside with a separate machine. Although effective, this approach relies on heavy harvesters and forwarders (15–20 t), which concentrate loads on forest soils. Repeated passes along trails can lead to rutting and soil compaction with long-term impacts on soil erosion and stand productivity. There is therefore a clear need for lighter, more versatile solutions that combine cutting and short-distance forwarding in a single machine, minimizing soil disturbance while operating in confined stands. To meet these needs, we developed the SAHA robot (Fig. 16) a 4.5-ton harvester optimized for selective thinning tasks, based on the RCM Harveri [25]

These components have been designed to operate independently from each other, such that each one could be used interchangeably with the forest management practice currently in use.

The remainder of the chapter is structured as follows. Section 2 describes the autonomy and mapping of the quadruped robot, as well as the autonomy stack for undercanopy flight, which includes a novel submapping based with trajectory adaptation upon loop closure, as well as a Neural Control Barrier Function (CBF) and MPC for reactive replanning in case of drift. Section 3 describes the online inventory system that was fully integrated with the quadruped platform, as well as the panoptic segmentation technique to segment trees and their subcomponents. Section 4 introduces an initial case study for the Decision Support System (DSS) which includes a growth simulator to support management decisions. Section 5 describes the semi-autonomous harvesters, including automation, navigation and grasping/cutting capabilities. Section 6 shows the experimental assessment of the systems presented in prior sections. Finally, Section 7 closes the chapter with final remarks and considerations for future works.

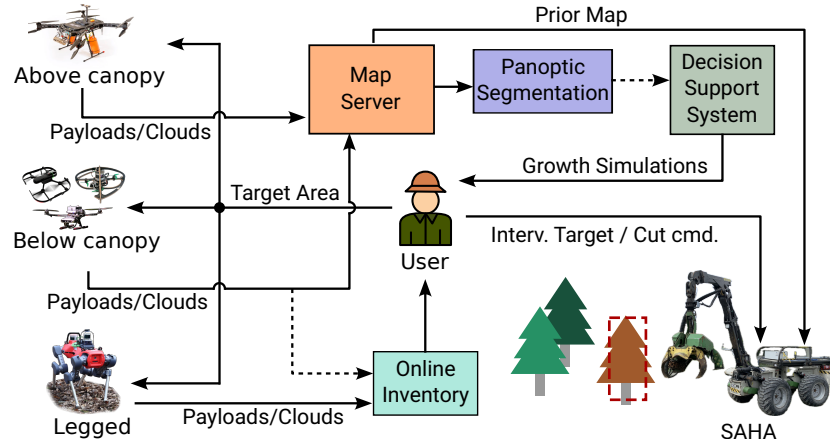


Fig. 1: Overview of the forest management approach proposed by DigiForest and data flow (solid lines indicate demonstration in the field, dashed lines indicate future integrations). The user initiates the data collection by providing a target area to survey. The maps from the ground units produce standardized payload maps for online tree inventory or multi-session mapping communicating them to a map server via a standardized interface. The maps are used to perform panoptic segmentation to produce tree traits, including its components such as crown and shrub, as well as to provide a prior map for the autonomous harvester. A decision support system uses the tree traits to produce growth simulation and predict the effect of an intervention. Finally, the operator decides the intervention to make and authorize the cut when the autonomous harvester has reached the site of intervention.

2 Autonomous Forest Navigation and Mapping

In this section, we describe the autonomous navigation and mapping system of our heterogeneous team of robots (legged, aerial, marsupial). Each robot category has specific complementary characteristics: legged platforms can cross a variety of obstacles and have long endurance, but cannot traverse through tall and dense undergrowth. Conversely, aerial platforms can rapidly navigate under-canopy but have limited payload and battery life. As each platform has its characteristic capabilities and requirements, we divide the remainder of the section by platform type. A summary of the platforms is shown in Table 1.

2.1 Legged Robot Navigation

For ground data acquisition, we developed and deployed an autonomy and forest inventory software stack on the ANYmal D quadruped platform. An overview of the

Name	Type	Dimension [cm]	Weight [kg]	Battery Life [min]
Avartek Boxer	UAV	$105 \times 105 \times 82$	17 (30 max)	120
RMF	UAV / Marsupial	$38 \times 38 \times 24$	1.45	8
MRL	UAV	$38 \times 38 \times 24$	1.7	15
BLK2FLY	UAV	$53 \times 60 \times 19$	2.6	10
ANYmal	Legged / Marsupial	$93 \times 53 \times 80$	50	90–120
SAHA	Harvester	$422 \times 220 \times 281$	4500	960+

Table 1: Overview of DigiForest robotic platforms.

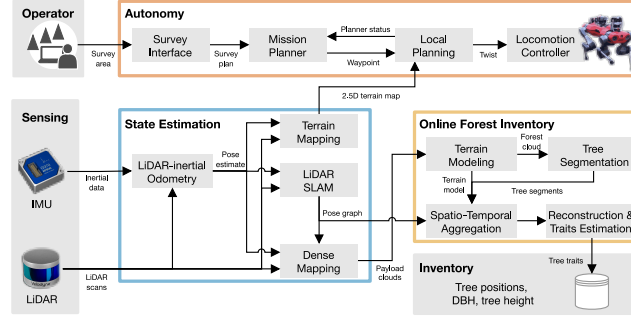


Fig. 2: System overview. The **Autonomy system** executes the mission specified by a survey reference provided by a human operator. **State estimation** provides a consistent scene representation and dense clouds used for estimating tree traits. **Forest inventory** segments the point cloud data and estimates tree traits from point cloud data. The main output of the system is a forest inventory database with the main attributes of the surveyed forest.

architecture is shown in Fig. 2: our solution incorporates a state-of-the-art LiDAR-based odometry and SLAM system [37], a mission planning and navigation system, and an online forest inventory pipeline, which we describe in Section 3.1. Further details and results from previous field trials in Finland and the UK are described in a prior work [32].

The ANYmal robot’s navigation system combines a reactive local planner with a global path-planning strategy. The local planner handles real-time obstacle avoidance and adapts to terrain changes. It outputs $SE(2)$ twist commands, which the robot’s lower-level control executes. Meanwhile, the global planner ensures long-term goal tracking and mission coverage. The planner leverages both geometric features (e.g., slope, elevation, obstacles) and semantic information (e.g., grass, mud) to make informed decisions in forest environments. This integration enables the robot to navigate safely and efficiently without becoming stuck (behind trees or obstacles) or deviating from its path.

2.1.1 Legged State Estimation

For state estimation (blue box in Fig. 2), we used a LiDAR-inertial factor graph-based odometry solution, VILENS [47], which provides high-frequency pose and velocity estimates (~ 200 Hz) in a fixed inertial frame. The system performs sensor fusion of relative pose displacements from LiDAR using the ICP algorithm [3], aided by preintegrated IMU measurements [15]. We calibrated the cameras, as well as Camera-IMU extrinsics using the Kalibr tool [20]. In addition, we developed a method for calibration LiDAR and camera using a checkerboard [18].

The factor graph-based LiDAR SLAM system from [43] is used to obtain a consistent estimate of the robot pose in a fixed global frame. Our SLAM system incrementally builds a pose graph, detects loop closure candidates within a radius of 10–15 m from the robot, and optimizes the pose graph in an incremental fashion using the iSAM2 algorithm [26].

We implemented a local sub-mapping module to accumulate all the incoming LiDAR scans within a fixed-distance window, generating dense maps we refer to as *data payloads* [8]. These data payloads are gravity-aligned and published at a lower frequency (~ 0.1 Hz), accumulating clouds for about every 20 m traveled.

The system also perform local terrain mapping and use its output as the main representation for local planning [34]. We used a 2.5D grid-based representation with a resolution of 4 cm.

2.1.2 Autonomy System

The input to the Autonomy System (orange box in Fig. 2) comes from the human operator, who defines the robot deployment plan from a graphical user interface (GUI) based on RViz using interactive markers (Fig. 3). Given the robot's initial pose, the GUI allows the operator to define a survey area and initialize a *survey plan* given by a *lawn mower pattern*, as done for TLS data collection protocols [13].

Once the survey plan from the operator is received, the mission planner manages its execution. The main task of the mission planner is scheduling the waypoints to be followed by the local planner, and executing any high-level action from the operator, such as safety stops. The local planner generates velocity commands for the low-level locomotion controller to reach a specified waypoint. We used a reactive local planner [31], which exploits the local terrain map to determine the action the robot executes.

The local planner outputs a 3 DoF linear and angular velocity command for a low-level RL-based locomotion controller. In our experiments, we used both the perceptive controller presented by Miki et al. [35], as well as the blind, learning-based controller from ANYbotics.

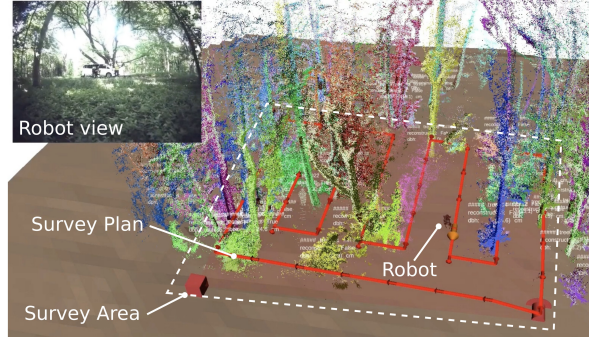


Fig. 3: Survey Interface: GUI for the operator, implemented on RViz using interactive markers. The operator defines the area to be surveyed, while the survey plan is automatically proposed by the Mission Planner. The segmented trees are shown in different colors.

2.2 Under-canopy Safe Navigation and Mapping

Unmanned Aerial Vehicles (UAV) for data acquisition above-canopy are becoming more commonly used in the forestry sector, as they provide a more flexible alternative to fixed-wing aircraft-mounted sensors for Aerial Laser Scanning (ALS) surveying. We employ a GNSS waypoint navigation plan to collect ALS data from an Avartek Boxer Hybrid UAV, to capture structural details that cannot be obtained from terrestrial scanning. The LiDAR collected from above-canopy is then fused with the data from under-canopy using the factor-graph based markerless approach described in [7].

In contrast to above-canopy ALS acquisition, autonomous under-canopy navigation is more challenging due to the presence of irregular obstacles, such as trunks and low branches, and the unreliability of GNSS data. To study the problem of safe navigation under-canopy in a general and versatile way, we tested our algorithms on three UAV systems: the MRL drone, which only relies on visual-inertial sensing, a collision-tolerant Resilient Micro Flyer (RMF) [40], and a commercial LiDAR-based drone, the Leica BLK2FLY.

The under-canopy navigation pipeline has four main components, illustrated in Fig. 4: a) a tightly coupled depth-visual-inertial SLAM, b) a large-scale volumetric mapping, c) an autonomous exploration and navigation process that relies on the volumetric maps, and d) a local collision avoidance controller.

2.2.1 Tightly coupled Depth-Visual-Inertial SLAM with Large-Scale Volumetric Occupancy Mapping

The UAV state estimation algorithm (Fig. 4.b) is based on OKVIS2 [28], a visual-inertial system which has been extended to integrate depth information to build dense

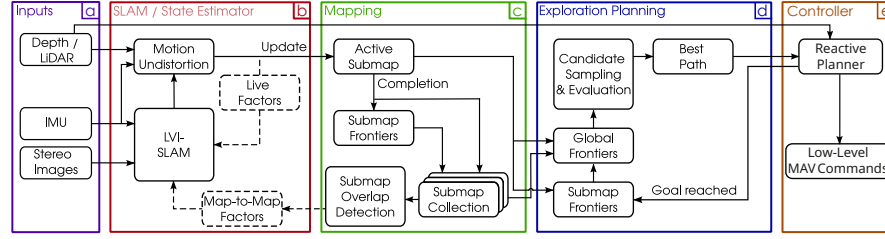


Fig. 4: Overall system architecture for autonomous exploration and mapping. The system leverages stereo images, IMU data and depth information from either a LiDAR or stereo images which are used for state estimation (b). The mapping module generates locally consistent submaps, that are stored together with the submaps’ local frontiers (c). These are later used in Exploration Planning (d) to compute global frontiers. After evaluating the frontiers for the best next drone pose, the path planner plans a trajectory to it, which the Model Predictive Controller (MPC) converts into attitude angles to control the UAV (e). The method can optionally leverage the submaps’ information to improve the state estimation accuracy.

volumetric occupancy submaps for both geometric reconstruction and path planning. The depth information can come from a LiDAR sensor or stereo vision approaches, such as [48].

The visual keypoints extracted from stereo cameras are tightly coupled with temporal IMU measurements to perform odometry estimates. The state estimator uses the poses to integrate depth information into the volumetric submaps, which are then used for downstream tasks. At the same time, the system can be configured to incorporate map-to-map and depth-to-map residuals into the state estimator, improving the odometry accuracy at the expense of higher computational requirements.

If the state estimator is configured to leverage the submap’s geometric information, we modify the factor-graph optimization as from our previous work [4]. In the Stein am Rhein field trials, we tested the proposed system and compared the reconstruction quality obtained by enabling and disabling the submap constraints. During the tests, the depth was obtained from either a LiDAR sensor (tested with the BLK2FLY) or from a Realsense D455 sensor (MRL UAV). All computations from the state estimator ran online.

2.2.2 Scalable Autonomous Drone Flight in the Forest with Visual-Inertial SLAM and Dense Submaps

For mapping (Fig. 4.c), the autonomous navigation system uses the same submaps from the estimator described in Section 2.2.1 for safe path planning. These submaps are constructed using the Supereight2 [19] multi-resolution occupancy mapping framework, where each voxel stores the log-odds probability of being occupied.

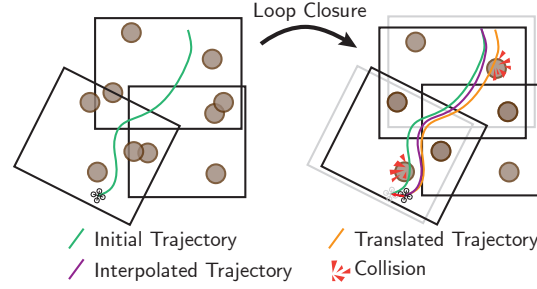


Fig. 5: Planned trajectory before loop-closure (left) and trajectory adaptation strategies after loop-closure (right), with obstacles as brown disks. The adaptation strategies are: no adaptation (green), rigid transformation (orange) and the proposed trajectory anchoring (purple). Without adaptation, the UAV flies towards its estimated pose before loop-closure (red arrow), which is unsafe. Rigidly transforming the trajectory does not account for the accumulated drift of each submap along it, leading to unsafe trajectories.

From this probability, regions within the submap are classified as free, occupied, or unknown. Unknown regions correspond to areas that are unobserved.

For path planning, we use the Informed-RRT* algorithm [21] and evaluate candidate path segments across all available submaps. A segment is considered valid if it lies entirely within free space in at least one submap. If the segment intersects an occupied or unknown region in a given submap, we check subsequent submaps. To ensure safety, we adopt a conservative approach by treating unknown regions as occupied during planning.

Once a safe path is found, it is converted into a trajectory that accounts for the UAV's velocity and acceleration constraints. This trajectory is anchored dynamically using updates from the state estimator. Since the estimator includes drift-correction mechanisms like loop closures, the trajectory must deform in real-time to remain consistent with the corrected state. Without this adaptation, discontinuities from loop closures could lead to erratic behavior or crashes, as the UAV might attempt to reach a trajectory point that now corresponds to a different physical location, as shown in Fig. 5. To address this issue, we deform the planned path based on changes in the underlying submaps, allowing the trajectory to elastically adapt and remain within known free space.

2.2.3 Efficient Submap-based Autonomous UAV Exploration using Visual-Inertial SLAM Configurable for LiDARs or Depth Cameras

Since the objective of the autonomous exploration algorithm is to visit previously unseen areas, we exploit the log-odds probability of occupancy to identify candidates to be explored.

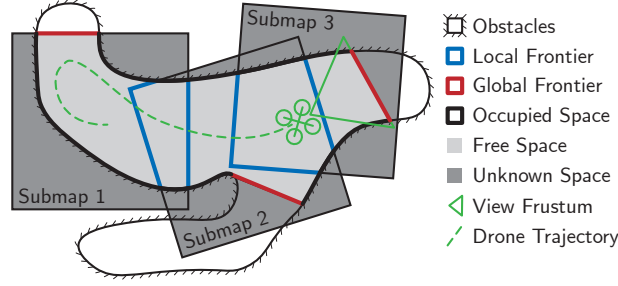


Fig. 6: Local-only (blue) and global (red) frontiers in a set of submaps. The local-only submap frontiers correspond to regions mapped in other submaps and are thus not considered to be global frontiers.

We begin by describing how the algorithm operates in a monolithic map before introducing the additional processes required for a submapping-based approach, as proposed in [38]. The exploration planner (Fig. 4.d) generates candidate positions near *map frontiers* i.e., regions likely to expand the known map if observed. In a volumetric occupancy map, a frontier is defined as the boundary between known free space and unobserved space.

From the UAV's current position, a path is planned to each candidate location. The estimated time to reach each candidate, t_j , is computed assuming the UAV travels at its maximum linear and angular velocities. At each candidate position, a 360° entropy-based raycast is performed to estimate the expected information gain, producing a *gain image*. Based on the sensor's horizontal field of view, a sliding window optimization is applied to identify the yaw angle that yields the highest gain g_j . The utility of each candidate is then calculated as $u_j = g_j/t_j$, effectively prioritizing candidates that maximize information gain per unit time. The candidate with the highest utility is selected as the next exploration goal (Fig. 4.e) and the process repeats once the goal has been reached.

In the submapping framework, the utility computation and raycasting steps remain the same but are applied over the entire collection of submaps. The primary difference lies in how global frontiers are defined for sampling candidate positions. Within a single submap, a frontier is any region where free space borders with unknown space or lies along the submap's boundary. However, in a multi-submap system, a frontier in one submap may no longer be valid in another if that region has already been observed. To handle this, we evaluate frontiers after a submap is completed. Using the known pose transformations T_{WM_k} for each submap, we test whether the local frontiers from one submap still correspond to unknown space in others. If the region is classified as free or occupied in any other submap, it is excluded as a sampling frontier. An illustrative example is shown in Fig. 6.

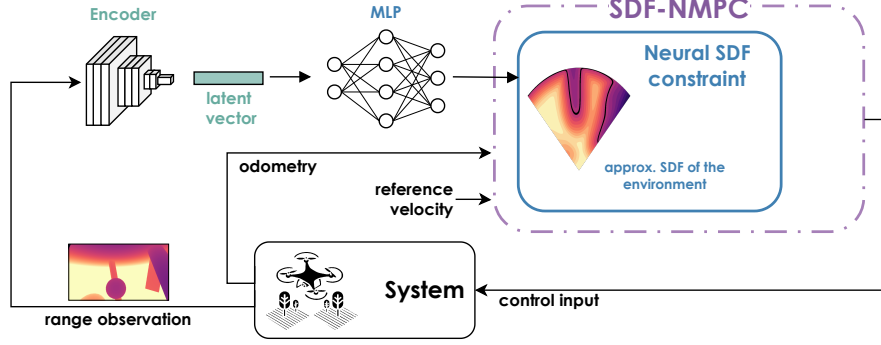


Fig. 7: A block-diagram of the SDF-NMPC method. The range image is processed via a neural network to encode the corresponding SDF within the sensor frustum. This SDF is used to parametrize a constraint in the NMPC, ensuring that trajectories satisfy a minimum distance to the obstacles.

2.2.4 Local Replanning for Collision Avoidance

Volumetric exploration planners provide paths that are nominally collision-free and typically rely on a voxel representation with a coarse resolution, so as to enable real-time computation. However, a coarse resolution makes it difficult to capture thin branches and twigs, posing risks of collisions and potential damage to the environment and/or the robot. Moreover, the volumetric map relies solely on the SLAM software and is therefore highly sensitive to systematic errors, such as position drift and estimation inaccuracies.

Consequently, it is important to introduce additional layers for collision avoidance by designing a complementary safety control layer that does not “blindly” track the planned trajectory. We therefore developed a safety layer able to process exteroceptive sensor data and assess whether actions are safe. Sensor input may come from an onboard LiDAR or a high-resolution depth camera, both of which provide detailed geometric information about the immediate surroundings.

Specifically, we use two methods designed to be used in a cascaded manner. The first method is a Nonlinear Model Predictive Controller (NMPC) exploiting a neural Signed Distance Field (SDF) as an explicit position constraint. It generates acceleration commands for tracking the reference trajectory. The second method relies on Control Barrier Functions (CBFs) to provide a last-resort safety filtering of the acceleration commands without relying on neural networks.

Each method has distinct advantages and trade-offs. The use of neural networks in the NMPC allows to work directly with high-dimensional sensor data. The method ensures a smooth behavior through predictive planning, but is computationally expensive. In contrast, the CBF is a reactive, computationally efficient formal approach with certified safety.

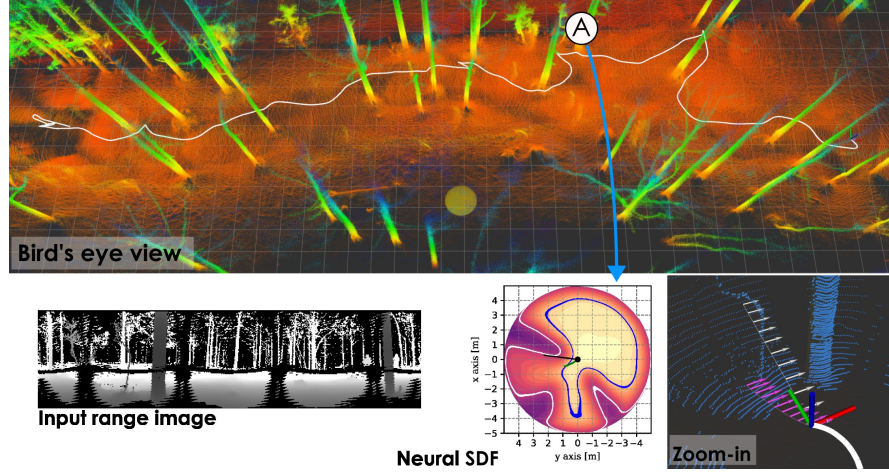


Fig. 8: Visualization of a LiDAR-based flight among trees. The lower part of the figure depicts a zoom-in of a specific time instance, along with the corresponding LiDAR input range image, and the $z_B = 0$ “slice” of the neural SDF encoding, with the obstacle line and the safety distance respectively in white and blue.

The NMPC is designed to track the reference trajectory provided by the exploration planner and provide acceleration commands. A neural network is used to process the sensor data directly and provide an SDF. This encoding is further used as a 3D position constraint in a local frame, while the reliance on only a local frame allows for improved resilience against odometry drift. An overview block diagram of the method, denoted SDF-NMPC, is pictured in Fig. 7.

First, the depth observations are encoded into a low-dimensional latent representation using a Variational Autoencoder (VAE). This latent encoding is used as an input for the neural network approximating the SDF, which is trained in a supervised manner by sampling. This training is performed fully offline, and this neural architecture has been shown to efficiently generalize to new environments without online fine-tuning.

The neural network is embedded as a constraint into the NMPC architecture. In the case of a constrained field-of-view sensor, additional position constraints are enforced to ensure that the system evolves in the currently visible space (such that the sensor is always pointed in the direction of motion to capture potential obstacles). Additionally, a theoretical analysis of the controller is proposed to derive recursive feasibility and stability conditions.

The SDF-NMPC was tested on the RMF in several forest environments to validate its collision-avoidance capability. During experiments, a human operator provided adversarial reference commands, intentionally trying to collide the robot with trees, branches, and bushes. Fig. 8 shows a visualization of such a trajectory, using the LiDAR and without resorting to any mapping system. The spatial encoding of the currently visible environment is depicted with the color gradient. Darker areas

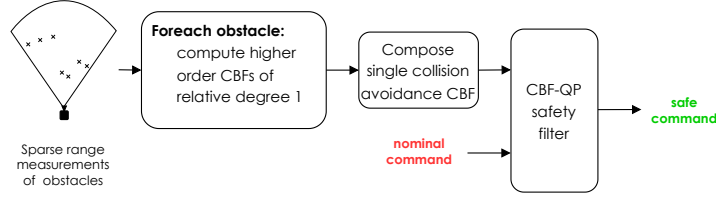


Fig. 9: An overview of the proposed composite CBF safety filter. The method synthesizes a single, composite CBF describe the avoidance of several point-wise obstacles. This CBF is in turn used as the constraint of a QP problem, finding the minimal deviation from a nominal input that ensures a safe behavior.

represent obstacles and unseen space, and the robot is constrained to navigate within the blue line, ensuring collision avoidance. This method ensures safe navigation against visible obstacles even in the case of failure of the mapping system, allowing for a safe recovery.

The neural NMPC yields an acceleration setpoint. Further, redundant safety is achieved by filtering this setpoint with a Control Barrier Function (CBF)-based safety filter. It is a computationally efficient reactive collision avoidance mechanism, synthesizing a composite CBF that describes safety with respect to the visible environment. It provides a formal safety certificate, which is used to filter out potentially unsafe actions from the higher-level controller.

An overview block diagram is presented in Fig. 9. From a conservatively sparsified point cloud measurement, a collection of high-order CBFs is computed, each describing safety with respect to the corresponding observed point. The composition method relies on computing an under-approximation of all those constraints. This continuous approximation yields a single, lumped constraint that describes safety with respect to all the visible obstacles. While the safety filter requires solving a Quadratic Programming, an analytical solution is available, making the method extremely computationally efficient.

2.3 Legged-Aerial Marsupial System

A marsupial system is a multi-agent setup consisting of an aerial robot and a quadruped. Combining the agility of the former with the endurance of the latter, this combination is particularly well-suited for forestry exploration, facilitating large-scale exploration while mitigating challenges related to difficult terrain. We designed a carrier platform and mounted on the quadruped to transport the aerial robot during walking phases. The carrier mechanism consists of two horizontal platforms and vertical to prevent lateral motion. A single actuator, powered by the battery of the quadruped, allow the mechanism to open and the drone to take off (Fig. 10).



Fig. 10: Deployment of a marsupial mission, including (left to right) the walking phase on easy terrain, the carrier mechanism opening, and the aerial robot starting the exploration mission.

The mechanism was validated by teleoperating the robot to an untraversable area and deploying the flying robot, which followed the exploration plan detailed in Section 2.2.

Testing in forested environments of co-planning and map sharing method, such as the one detailed on our prior work [49] is subject to future work. The method is able to assess whether a deployment of the aerial robot is necessary to perform the volumetric exploration (e.g., because of traversability issues) and, when the aerial robot is deployed, the two systems exchange a compressed shared map over wifi and individually perform mapping and exploration of the designed area.

3 Automated Forest Inventory

In this section, we describe technologies for estimating tree traits and constructing forest inventories using maps acquired during the robot’s navigation phase, either in real time (online) or post-mission (offline). We distinguish between three distinct processes: *Online inventory* generation occurs onboard one robot during a single navigation session, producing an inventory immediately usable by a forester (Sec. 3.1) *Multi-Mission SLAM* involves the aggregation and processing of LiDAR data collected across multiple missions and different robots to create a longitudinal data acquisition via multi-mission LiDAR SLAM. We based our LiDAR SLAM method on [37], which consistently reported up to 80% of correct loop closures even for scans that are 5 m apart and 60% when 10 m, effectively eliminating drift. The method was tested in four different forests in the UK, Finland and Switzerland. *Panoptic Segmentation* without real-time constraints, data provided by the Multi-Mission SLAM enables the use of more advanced and computationally intensive methods—such as panoptic segmentation for higher-fidelity analysis and interpretation of forest structure and tree traits (Sec. 3.2)

To support offline learning and evaluation of the last process, we collected a longitudinal dataset at three different time points. This dataset enables supervised training of learning-based algorithms for forest inventory tasks (Sec. 3.3).

3.1 Online Forest Inventory

Despite being robot-agnostic, our real-time forest inventory pipeline was primarily tested and developed for the legged robot and is build upon the output of its state estimator (Sec. 2.1), as described by Freißmuth et al. [16]. This pipeline first performs online tree segmentation and reconstruction during mission execution by combining LiDAR-based SLAM, a Tree Manager module, and a Hough Transform-based modeling procedure. The Tree Manager module applies an cloth simulation filtering to remove the ground from the SLAM payloads, then the normalized terrain is segmented out and DBSCAN and Non-Maxima Suppression are applied to the filtered cloud to distinguish tree trunk cylinders. A spatio-temporal aggregation then is used to incrementally build globally consistent maps and extract tree-level traits such as diameter at breast height (DBH) with accuracy comparable to terrestrial laser scanning, while operating without post-processing. The system was deployed on the ANYmal quadruped robot within the software stack described in Section 2.1 and validated in operational forest environments.

The outputs from the online forest inventory, available both as structured tabular files and through a dedicated SLAM interface, were integrated into QGIS, a widely used open-source Geographic Information System, allowing forestry researchers and practitioners to explore tree-level data and integrate it into their workflow (Fig. 11).

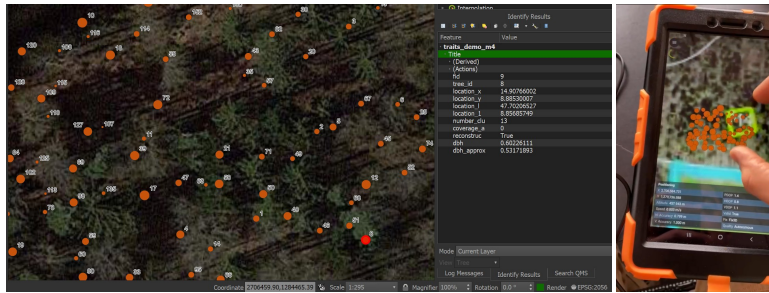


Fig. 11: Integration of the spreadsheet output from the online forest inventory [16] onto the QGIS software on PC (left) and tablet (right)

3.2 Longitudinal LiDAR Dataset

To facilitate training of semantic scene segmentation approaches for forestry applications, we collected and densely labeled a longitudinal dataset for forestry robotics called “DigiForests”, which also provides the foundation to assess the performance of forestry scene interpretation methods. We provide here a brief overview of the dataset and refer to our dataset paper [29] and website² for further details.

The dataset was collected in Oberwald forest, see Fig. 20, in three different campaigns: March 2023, October 2023, and July 2024. Therefore, our dataset provides data from different growth periods with varying amounts of shrub and under-canopy vegetation, which makes downstream tasks, like DBH estimation, potentially challenging due to the reduced visibility of the trunks.

The plots in the dataset show different levels of complexity in terms of traversability and clutter, but also differ in forest types, species, and silvicultural management regimes. For reference, a manually-built forest inventory includes measurements of tree DBH, length of clear wood, height of first green branch, and tree species.

To allow coverage of a large forest area, but also capture trunks and below-canopy scans at the same time, we used a different sensor setup employing similar LiDAR sensors used on the robotic platforms. Using a backpack carried by a human allows reaching challenging areas in the forest that are currently not reachable by legged robots, but also difficult for drones due to dense foliage. Therefore, our backpack solution is more practical for acquiring large amounts of LiDAR data in challenging scenarios with fewer restrictions on traversability or forest types, while featuring similar sensor configurations employed on the different robotic platforms. Differences in the positioning of the sensor, but also the number of LiDAR beams, are less relevant in the application scenario, where aggregated point clouds are used for a more fine-grained analysis as described in Sec. 3.3.

In particular, we used a backpack-mounted sensor package consisting of a Hesai XT32 LiDAR inclined at 45 degrees (March 2023), and a Hesai QT64 horizontally aligned (October 2023). In July 2024, we collected data with a combination of a horizontally aligned Hesai QT32 and a Hesai XT32 inclined at 45 degrees, to ensure maximum coverage of the tree stems and tree canopy. The sensor package was additionally equipped with a GNSS receiver for geo-referencing the data, and multiple cameras were employed for visual odometry.

Additionally, for the March 2023 and July 2024 campaigns, we used a UAV to record above-canopy aerial data covering the entire forest region shown in Fig. 20. For the March 2023 campaign, we flew a DJI Matrice M600 equipped with a sideways-mounted Velodyne HDL-32E while for the July 2024 campaign a Hesai XT-32M2X was used. We used on-board RTK GPS to globally geo-reference the drone via GNSS poses to register the corresponding backpack sensor point clouds from each plot.

The data was furthermore post-processed using SLAM techniques to generate aggregated point cloud maps. A key achievement is the spatial alignment of recorded data across seasons. To achieve sequence-to-sequence alignment across different

² The dataset is available at <https://digi-forest.eu/datasets>.

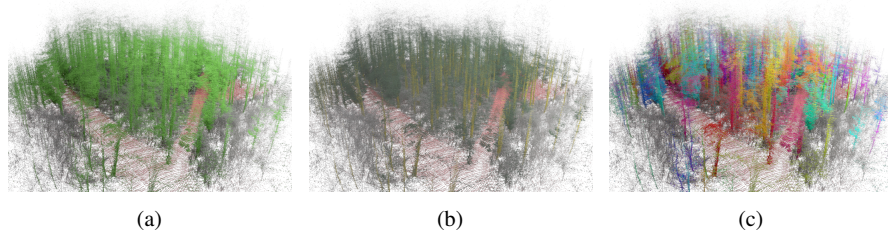


Fig. 12: The “DigiForests” dataset provides a longitudinal dataset of multiple forest plots recorded with a mobile LiDAR scanning system. The data has been recorded in three different seasons: Spring, Autumn, and Summer, with very different appearances in terms of vegetation. Besides a semantic annotation of the data into **tree**, **shrub**, and **ground** shown in (a), we also provide fine-grained annotations of **stem** and **canopy** of trees shown in (b), and instance annotations for trees shown by different colors in (c).

data collection seasons, we jointly optimized a pose graph over an individual plot’s trajectories from all sessions, globally aligning all point clouds spatially [7].

We manually labeled and verified all recording sequences using an extension of the point cloud annotation tool developed originally for the SemanticKITTI dataset [2]. Examples of the annotated data are shown in Fig. 12.

3.3 LiDAR-based Forest Panoptic Segmentation

For a more fine-grained analysis of the gathered LiDAR point cloud, we employ a panoptic segmentation approach to add the required information for downstream tasks, such as deriving tree traits, which are relevant for providing the decision support system for forestry management (see Sec. 4). Note that the offline processing allows for processing aggregated scans and additional post-processing that is not possible in an online setting. Furthermore, more computationally demanding approaches can be employed, which lead to a more complete forest inventory.

Our approach to panoptic segmentation builds on commonly employed semantic and instance segmentation for urban and agricultural environments. However, applying these techniques directly to forestry presents unique challenges due to the hierarchical and overlapping nature of the annotations, especially for the tree, i.e., stem and canopy classes, described in Sec. 3.2. To effectively leverage such approaches for forest scene interpretation, it is necessary to adapt these methods to handle the specific structure and requirements of our application.

To address these challenges, we designed a panoptic segmentation approach tailored to forest scenes captured by LiDAR sensors. For segmentation, we use Minkowski Engine [9], an auto-differentiation library for sparse tensors suitable for efficiently processing sparse 3D point clouds, which is crucial for handling the large, mostly empty spaces typical of LiDAR data. This leads to significant savings in

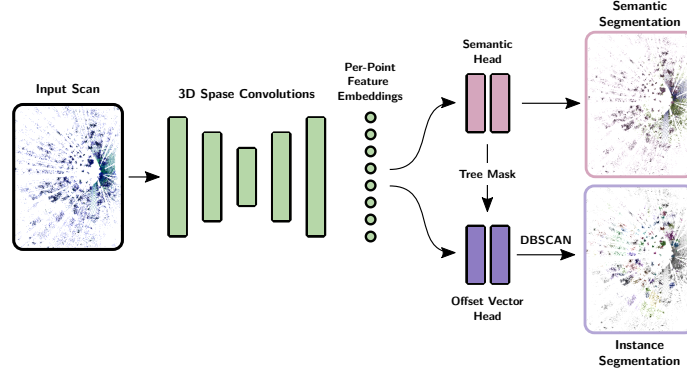


Fig. 13: Overview of our panoptic segmentation approach. We use a LiDAR scan as input for a sparse 3D convolutional encoder-decoder that generates a feature embedding per LiDAR point. The feature embeddings are passed to two multi-layer perceptron (MLP) heads, resulting in per-point semantics from the semantic head and instance segmentation from the offset vector head via clustering via DBSCAN.

both memory and computational resources compared to dense convolutional neural networks using voxel grids.

The network architecture of our approach is shown in Fig. 13. We use a MinkUNet architecture [9], which is an extension of the U-Net [44] model specifically optimized for sparse data using the Minkowski Engine. The encoder-decoder architecture produces a feature embedding vector of size 32 for each point in the input LiDAR scan. The feature embedding is then passed on to two network heads in parallel: a semantic head responsible for estimating the semantic classes and an offset vector head predicting offsets that can be used to derive instances. More specifically, the semantic segmentation head is a two-layer multi-layer perceptron (MLP) that predicts per-point probabilities over the classes: ground, shrub, stem, or canopy. Moreover, the offset vector head is a four-layer MLP predicting per-point three-dimensional offset vectors, which point to the geometric center of a potential tree instance.

For deriving individual tree instances, we first determine points classified as stem or canopy and then apply the predicted three-dimensional offset vectors to these tree points, shifting them towards the estimated center. The shifted points are then clustered using the DBSCAN [14] to group them in individual tree instances. More details on the approach can be found in our dataset paper [29]. Our method performs both semantic segmentation and tree instance segmentation, addressing the unique hierarchical labeling structure presented in Sec. 3.2.

4 Decision Support System

The Decision Support System (DSS) combines the robot-derived forest inventory with environmental data and management rules to support interventions at two horizons. In the short term, stand- and tree-level attributes (e.g., DBH, height, form, biomass/volume proxies, health) underpin choices about which trees to remove or retain and the expected economic return. In the long run, growth simulators model stand development under various climate scenarios and management paths, enabling multi-objective trade-offs between carbon sequestration, resilience, timber yield, protection, and recreation. (Fig. 14).

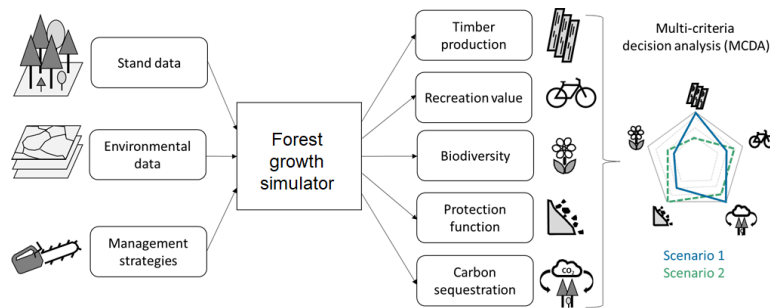


Fig. 14: Overview of the Decision Support System pipeline (adapted from Thrippleton/WSL).

As an initial case study, we used the climate-sensitive gap model ForClimV4.1 [5] for this kind of future forest growth estimation. We initialized it using manually collected ground truth reference data, including information about Diameter at Breast Height (DBH), tree species, and current health status of a small forest patch within the Stein am Rhein study area.

In total, the individual tree data collection included 800 trees, covering a wide variety of forest types (e.g., conifers/broadleaves; dense and light stands; continuous cover and rotation forestry) and will be made publicly available at a later stage. A snippet of the forest area map of the data collected is shown in Fig. 15.

We formulated potential forest management strategies together with the forester of the site and implemented them into the growth simulator. Simulations have been conducted for a time horizon of ca. 100 years, thereby considering historical climate data as well as different estimated climate scenarios (RCP4.5 and RCP8.5). The results will be coupled to indicator value functions allowing the calculation of future provision of biodiversity and ecosystem services. In future work, we intend to extend this approach to use data collected from robotic systems and extract individual tree parameters and process in the tree growth simulator.



Fig. 15: Individual tree data collected by team-WSL at Stein am Rhein, Switzerland as seen from the data inventory software.

5 Autonomous Harvesting



Fig. 16: The SAHA autonomous harvester.

The SAHA robot is designed to perform cutting and forwarding operations at the same time, minimizing the number of steps required in the harvesting process. The lightweight design of the robot minimizes the impact on the forest floor and allows it to traverse and operate in confined spaces, which is crucial for selective thinning. An active suspension chassis with four independently actuated legs provides stability on uneven ground. The chassis features an articulated steering mechanism, allowing relative rotation between the front and back parts of the vehicle [11]. The arm offers six degrees of freedom, of which four are hydraulically actuated, and carries a bidirectional harvester head.

A compact perception module combining a LiDAR, an RGBD camera, an IMU, and a GNSS receiver is mounted on the side of SAHA's arm, enabling localization, navigation, and tree detection. Onboard compute is provided by an NVIDIA Jet-

son Orin module integrated in the perception module and an industrial computer performing real-time control. A pipeline for autonomous operation of SAHA has been developed, with a number of its modules tested in the field, demonstrating fully automated cutting operations of target trees specified in the prebuilt map.

The following sections describe the main components of the autonomous harvester and their integration into a complete system.

5.1 Control Automation

Various controllers have been implemented for the robot, providing key functionalities such as terrain adaptation, driving control, and arm control. The terrain adaptation controller is adapted from our previous implementation [24] and consists of a cylinder force controller and a chassis pose controller. The chassis pose controller utilizes a Virtual Model Controller (VMC) to compute desired force and torque on the chassis. Then, hierarchical optimization is used to distribute the support forces of the legs to maintain stable contact when the robot operates on uneven forest ground. The desired support forces from individual legs are tracked by the cylinder force controller, which relies on the feedback from the chassis cylinder pressure measurements. For more details of the implementation, we refer the readers to our previous work [24].

While this optimization-based approach aims to find the most stable leg configuration subject to SAHA's kinematic limits, there exist difficult terrain patches that the balancing controller would fail to maintain stable ground contact. To quantify the limit of the balancing controller, we leverage massive parallel simulations. As seen in Figure 17, we spawn several SAHA robots on a randomly generated terrain. Each robot attempts to establish a stable ground contact. The terrain patches under the successful ones are marked as traversable and receive a traversability score based on the remaining margin for chassis pose control, while the others are marked as not traversable. Based on the simulation result, we use the result to train a geometric-based traversability prediction module similar to [17] and use it as the traversability cost in local navigation

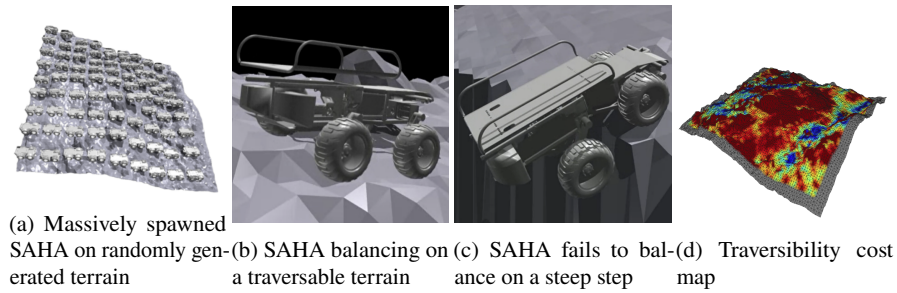


Fig. 17: SAHA traversability evaluation in parallel simulation.

Assuming the terrain adaptation controller handles the terrain uncertainties, the driving controller is designed based on a planar simplification of the kinematic model of the center-articulated chassis of SAHA. The driving controller enables SAHA to reliably track given reference targets. The derivation and convergence analysis is presented in [23].

Once SAHA arrives in the vicinity of the target tree, the arm must be accurately controlled to move the harvester head to the target tree trunk. While controlling hydraulically actuated robot arms is complex, we use a learning-based controller with online adaptation capabilities [36] to adapt to SAHA's hydraulic component dynamics at test time. As reported in [36], it reaches around 5 cm accuracy for end effector trajectory tracking, which provided adequate performance for harvesting.

5.2 Navigation in Scanned Forest

To autonomously navigate to the selected tree, we utilized the LiDAR map of the target area from the previous scans of the legged and aerial platforms. The SAHA robot localizes itself within the prebuilt map frame by matching the onboard LiDAR point cloud with the one from the map. Upon receiving a cutting target, a series of waypoints is generated between SAHA's current location and the target tree.

The waypoints are then used by a local path planner to generate a local path considering obstacles and traversability. Utilizing prebuilt motion primitives with offline simulation, the local navigation module selects collision-free motion plans that can be effectively tracked by SAHA's unique articulated steering system. During local planning, a real-time terrain analysis module generates traversability labels of the area surrounding SAHA based on geometric data. The traversability cost is considered by the local planner to simultaneously minimize risk while selecting the best primitive in the target direction. This capability ensures safe and efficient navigation in the complex and unpredictable forest environment. For more details on the local navigation algorithm, see [23]. We deployed the Traversability-Aware Navigation algorithm detailed in [23] in the Stein am Rhein forest. A picture of the navigation system running on the SAHA robot is shown in 18.

5.3 Map-Based Tree Grasping and Cutting

Using the prebuilt map, the user can specify the target tree location through a simple RViz interface. The state machine first evaluates the distance between the robot's current position and the specified tree to determine whether it is within the arm's reachable range. If the tree is out of reach, the autonomous navigation module is triggered to drive the robot closer to the specified location. Once the tree is within the arm's reachable range, the system activates SAHA's automatic cutting mode. The arm is then controlled to reach the target trunk, and once the clamp is in position,

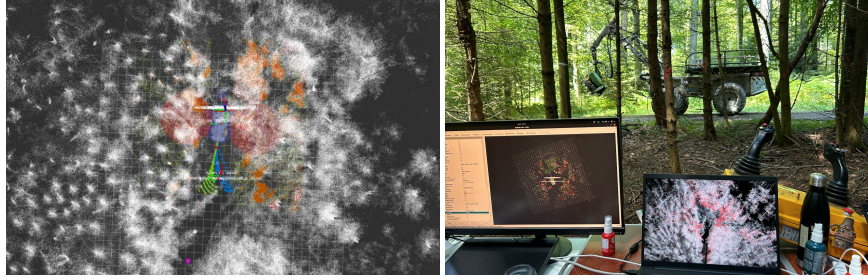


Fig. 18: SAHA navigating through the forest with a prior point cloud map. (Top) Prebuilt point cloud map in white, matched local point cloud in orange. Navigation target in purple, and motion primitives selected by local planner in green. (Bottom) picture of SAHA during autonomous navigation.

the operator is required to manually trigger the button to initiate cutting (Figure 19), ensuring safety.



Fig. 19: The SAHA robot releases the cut tree after manual triggering.

6 Experimental Evaluation

The technologies presented by DigiForest have been extensively tested in real-world scenarios in Finland, the UK, and Switzerland. One of the principal test sites used is the Oberwald forest near Stein am Rhein, Switzerland. The site represents a characteristic Central European forest environment with deciduous, coniferous, and mixed stands. The area is actively managed and provides suitable conditions for testing autonomous systems under realistic operational constraints, including varied terrain, dense vegetation, and canopy occlusion. For experimental purposes, the forest has been subdivided into plots based on dominant tree type: “D” for deciduous, “C” for coniferous, and “M” for mixed-species stands, as shown in Fig. 20

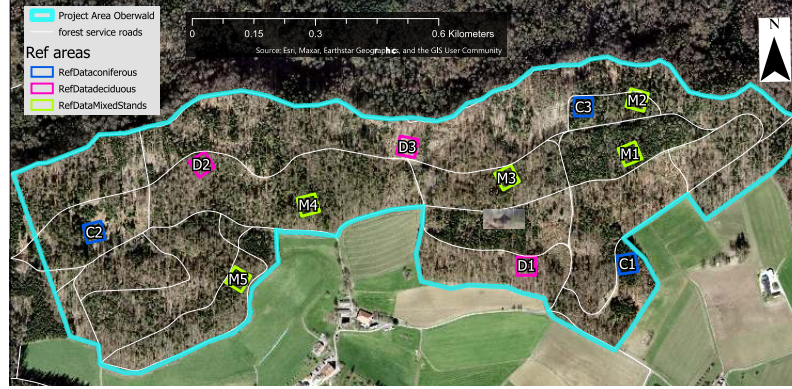


Fig. 20: Map of the Stein am Rhein test site plots. The area is divided into labeled plots based on dominant tree type: ■ deciduous (D), ■ coniferous (C), and ■ mixed-species (M).

6.1 Autonomous Legged Navigation

We deployed the quadruped robot in plot M4 of the Oberwald forest, as shown in Fig. 21. The plot was chosen because it presented less undergrowth and high grass than other plots in the forest, which were more challenging for the robot to traverse due to its comparable height (~ 60 cm). The ANYmal robot was deployed autonomously four times in the same plot, with each mission taking approximately 8 min. On each run, our forest inventory system was executed while the robot navigated autonomously in the field. We report 102.5 m average for the Mean Distance Between Interventions (MDBI) metric for a path of 152 m, indicating the robot operated autonomously for most of the mission. The sequences covered areas of up to 0.2 ha, consistently reporting around 40 trees on each mission, which is a sign of the repeatability of the forest inventory system.

The quality of the robot trajectory as well as the map estimated by a SLAM system depends on obtaining frequent loop closures. This is challenging in forest environments due to its self-repeating geometry as well as strong occlusions in sensor observations. In the work by Oh et al. [37], we analyze the capabilities several state-of-the-art LiDAR place recognition systems, and develop a pipeline for obtaining robust loop-closures with baselines ranging from 5–10 m. The pipeline provides reliable loop-closures by leveraging a candidate proposal models learned on forestry datasets [46], followed by a geometric verification step and a final refinement using Iterative Closest Point (ICP) registration.

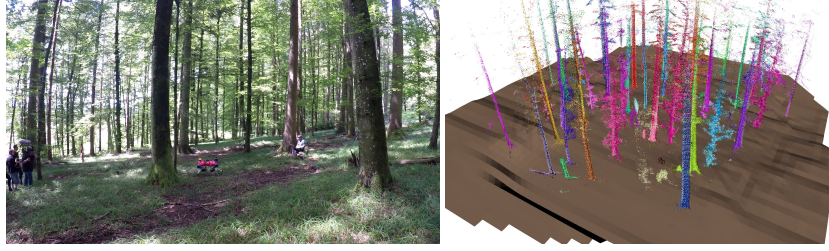


Fig. 21: Autonomous deployment of ANYmal in Oberwald: *Left*: ANYmal surveying the *M4* plot, with mixed trees. *Right*: Tree segmentation and digital terrain model after the mission.

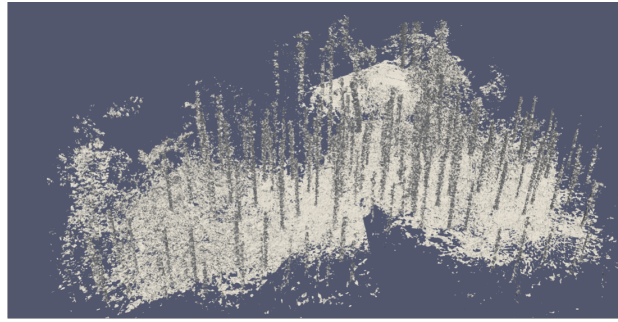


Fig. 22: 3D mesh reconstructed by BLK2FLY in Stein am Rhein.

6.2 Autonomous Undercanopy Flight

The autonomous exploration stack was tested in Oberwald with two depth sensing modalities: LiDAR on the BLK2FLY and depth from the MRL’s RealSense D455.

Since access to the computing unit onboard the BLK2FLY was unavailable, we streamed the data to a laptop in the field, where all computations on the exploration stack were performed. Whenever a new path was planned, it was sent to the BLK2FLY via a ROS interface, and the UAV continued with a regular exploration mission. An example of the 3D reconstruction obtained with the BLK2FLY from a test site in Stein am Rhein is shown in Fig. 22.

We also demonstrated our exploration algorithm with the MRL UAV. Two major tests were performed: demonstrating the benefits of performing Visual-Inertial SLAM with submap-to-submap alignment and exploration in the forest.

To demonstrate the benefits of incorporating geometric information into the state estimation, we performed two missions in the same area, one leveraging the submap constraints and another without them. In both missions, the depth was obtained from the RealSense D455. The mission was an exploration setup, and after 10 minutes (the UAV’s battery lifetime), the UAV was manually flown to the start position. Since

the end and start area are the same, we perform an ICP alignment to these submaps and estimate the error by the translation of the final transform. Even though the robots start and finish in the same area, there were no visual overlaps since the agents had opposing viewing directions, impeding loop-closures. The approach that had enabled the submap constraints achieved an end position error of 0.19 m while without submap constraints the final error was 0.37 m. A qualitative example of the effect of the submap constraints in the mesh reconstruction can be seen in Fig. 23. At the same time, we demonstrate that the proposed approach can effectively work with both depth from LiDAR or from cameras.

When comparing both approaches, Fig. 22 and Fig. 23, it is clear that the LiDAR approach produces a more accurate and complete reconstruction thanks to its larger field-of-view and the unmatched depth-sensing capabilities of this sensor. On average, the missions that relied on visual perception mapped an estimated volume of 706.69 m^3 , while with the BLK2FLY mapped volume is of 8612.32 m^3 .

A total of 10 autonomous exploration missions were performed, where no manual intervention by the safety pilot was required. This demonstrates that the proposed system is safe and can be used in forest scenarios, even when densely cluttered.

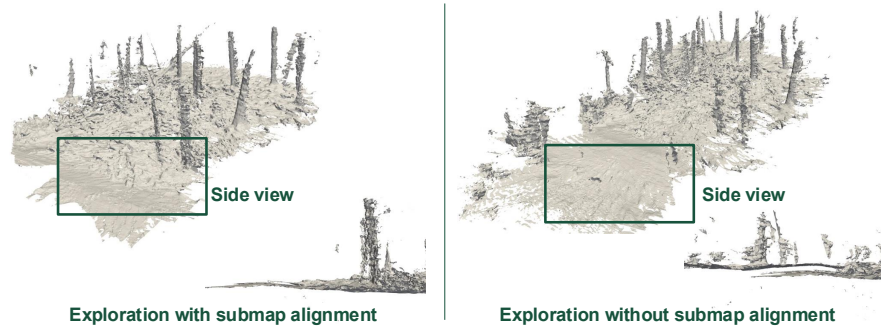


Fig. 23: 3D mesh reconstructed onboard by MRL drone in the field trip, Stein am Rhein. (Left) The first and the last submaps are well-aligned due to submap alignment factors in the factor graph optimization. (Right) The submaps are geometrically inconsistent without the submap alignment factor.

6.3 Panoptic Segmentation

For quantitative experimental evaluation, we used our dataset presented in Sec. 3.2. The dataset is split into a training dataset comprising 2 complete plots over all time points and the remaining plots from the first measurement campaign. For validation, i.e., hyper-parameter tuning, we use a complete temporal sequence of a single plot. For evaluation of the performance on a separate test set, we use a plot containing a

Table 2: Results for semantic interpretation on the test set of DigiForests. PQ is computed considering the ground, shrub, and tree classes.

Approach	Ground	Shrub	Stem	Canopy	Tree	PQ
MaskPLS [30]	65.9	53.9	-	-	57.0	58.9
Our approach	79.5	72.7	78.5	48.2	57.8	70.0

mixture of tree species and distributions to test the generalization performance of trained models. The clear spatial separation of train, validation, and test split allows for benchmarking methods in an unbiased fashion with reduced risk of overfitting.

For evaluating panoptic segmentation performance, we use the panoptic quality metric [27, 42], where the per-class panoptic quality PQ_c is given by

$$PQ_c = \begin{cases} \frac{\sum_{(p,g) \in TP} \text{IoU}(p, g)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}, & \text{if } c \text{ is thing} \\ \text{IoU}(p, g) & , \text{if } c \text{ is stuff} \end{cases}, \quad (1)$$

where p is the predicted and g is the ground truth segment of a class c , TP is the set of true positives, FP is the set of false positives, FN is the set of false negatives. With $|\cdot|$, we refer to the number of elements in the given set. For so-called thing classes, we follow the data assignment by Kirillov et al. [27], which treats each predicted instance segment p as a true positive if a ground truth segment g has an intersection over union (IoU) greater than 0.5. Conversely, predicted instance segments p without a matching ground truth segment are false positives, and ground truth segments g without an associated prediction are false negatives. For stuff classes, we compute the IoU between all points p assigned to class c and the points g with class c in the ground truth. The overall panoptic quality PQ over all classes C , *i.e.*, ground, tree, and shrub, are given by the average over class-wise panoptic qualities PQ_c given by:

$$PQ = \frac{1}{|C|} \sum_{c \in C} PQ_c. \quad (2)$$

Tab. 2 presents the test set results of our approach compared to a LiDAR-based panoptic segmentation approach, called MaskPLS [30], which was developed in the context of autonomous driving using a mask-based Transformer decoder to generate per-class binary masks. In comparison, our approach can also predict simultaneously the stem and canopy classes, which overlap with the tree class as explained in Sec. 3.2. Furthermore, our approach shows promising performance in comparison to the Transformer-based architecture, which might be caused by the amount of available training data. A qualitative example of the predictions is shown in Fig. 24.

The distinction of stem and canopy by the offline processing enables tackling scenarios where pure geometric approaches struggle due to small bushes and other vegetation close to the trees. The more fine-grained segmentation enables a more complete forest inventory, and enables deriving other forestry-related traits, such as

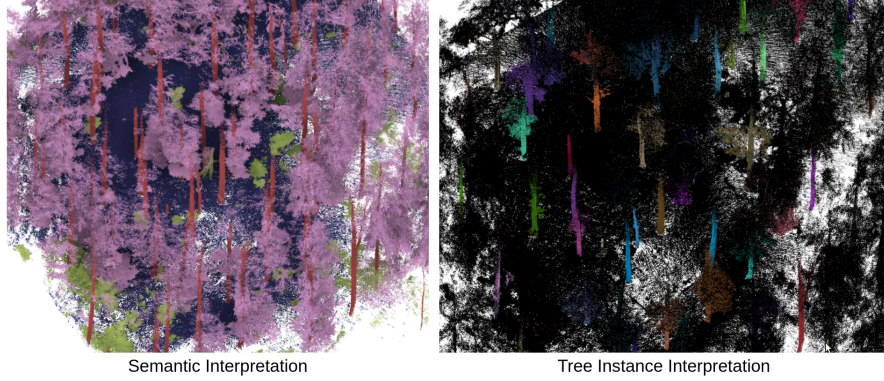


Fig. 24: Exemplary semantic and instance predictions generated with our panoptic segmentation approach showing the semantic classes: ground (blue), shrub (green), canopy (purple), and stem (ground). In the right part, the instance predictions are shown via a random coloring, where tree instances get the same color assigned.

Table 3: Results for tree DBH estimation using the DigiForests dataset. Root mean squared error (RMSE) is reported in cm.

Approach	Recall		RMSE	
	Per-plot Avg.	Overall	Per-plot Avg.	Overall
Geometric approach [16]	0.88	0.92	3.42	3.15
Cylinder fitting on stem segments	0.98	0.98	5.32	6.41

the clear wood percentage of the tree by using the stem region, the canopy volume due to the separation of the foliage, or the more complete estimation of the diameter at breast height. Tab. 3 shows the advantage with respect to the geometric online estimation of DBH onboard a robotic system [16] described in Sec. 2.1 In particular, the online approach cannot reliably handle some cases, where the offline processing and a more fine-grained distinction into stem segments can provide a DBH estimate, as shown by the higher per-plot average recall and overall recall.

In summary, our approach shows promising results compared to conventional LiDAR-based panoptic segmentation approaches employed in the context of autonomous driving. The approach is tailored for offline processing of aggregated scans, and a promising avenue for future work is an extension towards single scan panoptic segmentation and integration into a mapping framework to allow building semantic maps on board a robotic system.

7 Conclusions and Future Work

In this chapter, we have presented DigiForest, an innovative approach towards large-scale precision forestry that addresses the whole process of forest management, from data collection by a team of heterogeneous robots, its processing and transformation into a forest inventory, the integration of the inventory into a decision support system, and finally the autonomous tree harvesting via a driverless, purpose-build, hydraulic machine.

All these technologies have been validated in real operative scenarios, such as the Oberwald forest in Switzerland and other locations in Finland and the UK. The field trials served to test and validate key functionalities under realistic conditions, highlighting both the potential and the remaining technical challenges. Rather than replacing traditional forestry practices, the aim is to support them with more reliable, adaptable, and data-informed tools.

These developments reflect steady progress toward the integration of robotic perception and autonomy in forestry operations. These systems are being designed with forest-specific challenges in mind, including variable terrain, dense vegetation, and limited visibility.

Future work will develop across two main directions: making each individual component more robust (e.g., by testing in more challenging scenarios) and increasing the integration between components. For example, we plan to enhance the SAHA harvester’s global route planning and reactive obstacle avoidance by using real-time semantic terrain analysis to classify ground cover and hazards.

References

- [1] Sébastien Bauwens et al. “Forest Inventory with Terrestrial LiDAR: A Comparison of Static and Hand-Held Mobile Laser Scanning”. In: *Forests* 7.6 (2016). 127. ISSN: 1999-4907. DOI: 10.3390/f7060127.
- [2] J. Behley et al. “SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences”. In: *Intl. Conf. on Computer Vision (ICCV)*. 2019.
- [3] Paul J. Besl and Neil D. McKay. “A Method for Registration of 3-D Shapes”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 14.2 (1992), pp. 239–256. DOI: 10.1109/34.121791.
- [4] Simon Boche, Sebastián Barbas Laina, and Stefan Leutenegger. “Tightly-Coupled LiDAR-Visual-Inertial SLAM and Large-Scale Volumetric Occupancy Mapping”. In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2024, pp. 18027–18033. DOI: 10.1109/ICRA57147.2024.10610460.
- [5] Harald K. M. Bugmann. “A Simplified Forest Model to Study Species Composition Along Climate Gradients”. In: *Ecology* 77.7 (1996), pp. 2055–2074. DOI: <https://doi.org/10.2307/2265700>. eprint: <https://esajournals.onlinelibrary.wiley.com/doi/pdf/10.2307/2265700>. URL: <https://doi.org/10.2307/2265700>.

- //esajournals.onlinelibrary.wiley.com/doi/abs/10.2307/2265700.
- [6] Andrew Burt, Mathias Disney, and Kim Calders. “Extracting individual trees from lidar point clouds using treeseg”. In: *Methods in Ecology and Evolution* 10.3 (2019), pp. 438–445. DOI: <https://doi.org/10.1111/2041-210X.13121>.
 - [7] Benoît Casseau et al. “Markerless Aerial-Terrestrial Co-Registration of Forest Point Clouds using a Deformable Pose Graph”. In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2024, pp. 39–46. DOI: [10.1109/IROS58592.2024.10802448](https://doi.org/10.1109/IROS58592.2024.10802448).
 - [8] Nived Chebrolu and Maurice Fallon. *Report defining data format and software API for multi-robot map sharing*. Tech. rep. University of Oxford, 2023.
 - [9] C. Choy, J. Gwak, and S. Savarese. “4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks”. In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2019.
 - [10] European Commission. *Commission communication: New EU forest strategy for 2030*. 2021. URL: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021DC0572>.
 - [11] P.I. Corke and P. Ridley. “Steering kinematics for a center-articulated mobile robot”. In: *IEEE Transactions on Robotics and Automation* 17.2 (2001), pp. 215–218. DOI: [10.1109/70.928568](https://doi.org/10.1109/70.928568).
 - [12] Ralph O. Dubayah and Jason B. Drake. “Lidar Remote Sensing for Forestry”. In: *Journal of Forestry* 98.6 (June 2000), pp. 44–46. ISSN: 0022-1201. DOI: [10.1093/jof/98.6.44](https://doi.org/10.1093/jof/98.6.44).
 - [13] L Duncanson et al. “Global aboveground biomass product validation best practices protocol. Version 1.0”. In: *Best practice protocol for satellite derived land product validation. Land Product Validation Subgroup (Working Group on Calibration and Validation, Committee on Earth Observation Satellites)* (2020).
 - [14] M. Ester et al. “A density-based algorithm for discovering clusters in large spatial databases with noise.” In: *Proc. of the Conf. on Knowledge Discovery and Data Mining (KDD)*. 1996.
 - [15] Christian Forster et al. “On-Manifold Preintegration for Real-Time Visual-Inertial Odometry”. In: *IEEE Trans. Robot.* 33.1 (2017), pp. 1–21. DOI: [10.1109/TR0.2016.2597321](https://doi.org/10.1109/TR0.2016.2597321).
 - [16] Leonard Freißmuth et al. “Online Tree Reconstruction and Forest Inventory on a Mobile Robotic System”. In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2024.
 - [17] Jonas Frey et al. “Locomotion Policy Guided Traversability Learning using Volumetric Representations of Complex Environments”. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022, pp. 5722–5729. DOI: [10.1109/IROS47612.2022.9982190](https://doi.org/10.1109/IROS47612.2022.9982190).
 - [18] Lanke Frank Tarimo Fu, Nived Chebrolu, and Maurice Fallon. “Extrinsic Calibration of Camera to LIDAR Using a Differentiable Checkerboard Model”.

- In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2023, pp. 1825–1831. doi: 10.1109/IROS55552.2023.10341781.
- [19] Nils Funk et al. “Multi-resolution 3D mapping with explicit free space representation for fast and accurate mobile robot motion planning”. In: *IEEE Robot. Autom. Lett.* 6.2 (2021), pp. 3553–3560.
 - [20] Paul Furgale, Joern Rehder, and Roland Siegwart. “Unified temporal and spatial calibration for multi-sensor systems”. In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2013, pp. 1280–1286. doi: 10.1109/IROS.2013.6696514.
 - [21] Jonathan D Gammell, Siddhartha S Srinivasa, and Timothy D Barfoot. “Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic”. In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2014, pp. 2997–3004.
 - [22] G. J. Hamilton. *Forest Mensuration Handbook*. 1988.
 - [23] Jiangpeng Hu et al. “Motion Primitives Planning For Center-Articulated Vehicles”. In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2024, pp. 12702–12709.
 - [24] Marco Hutter et al. “Force control for active chassis balancing”. In: *IEEE/ASME Transactions On Mechatronics* 22.2 (2016), pp. 613–622.
 - [25] Edo Jelavic et al. “Harveri : A Small (Semi-)Autonomous Precision Tree Harvester”. In: *ICRA 2022 Workshop in Innovation in Forestry Robotics: Research and Industry Adoption*. 2022. URL: <https://openreview.net/forum?id=B0dgBn60QM5>.
 - [26] Michael Kaess et al. “iSAM2: Incremental smoothing and mapping using the Bayes tree”. In: *Int. J. Robot. Res.* 31.2 (2012), pp. 216–235. doi: 10.1177/0278364911430419.
 - [27] A. Kirillov et al. “Panoptic Segmentation”. In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2019.
 - [28] Stefan Leutenegger. “Okvis2: Realtime scalable visual-inertial slam with loop closure”. In: *arXiv preprint arXiv:2202.09199* (2022).
 - [29] Meher V.R. Malladi et al. “DigiForests: A Longitudinal LiDAR Dataset for Forestry Robotics”. In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2025.
 - [30] R. Marcuzzi et al. “Mask-Based Panoptic LiDAR Segmentation for Autonomous Driving”. In: *IEEE Robot. Autom. Lett.* 8.2 (2023), pp. 1141–1148.
 - [31] Matías Mattamala, Nived Chebrolu, and Maurice F. Fallon. “An Efficient Locally Reactive Controller for Safe Navigation in Visual Teach and Repeat Missions”. In: *IEEE Robot. Autom. Lett.* 7.2 (2022), pp. 2353–2360. doi: 10.1109/LRA.2022.3143196.
 - [32] Matías Mattamala et al. “Building Forest Inventories With Autonomous Legged Robots—System, Lessons, and Challenges Ahead”. In: *IEEE Transactions on Field Robotics* 2 (2025), pp. 418–436. doi: 10.1109/TFR.2025.3583972.
 - [33] Mikko Miettinen et al. “Simultaneous Localization and Mapping for Forest Harvesters”. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. 2007, pp. 517–522. doi: 10.1109/ROBOT.2007.363838.

- [34] Takahiro Miki et al. “Elevation Mapping for Locomotion and Navigation using GPU”. In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2022, pp. 2273–2280. DOI: 10.1109/IROS47612.2022.9981507.
- [35] Takahiro Miki et al. “Learning Robust Perceptive Locomotion for Quadrupedal Robots in the Wild”. In: *Sci. Robot.* 7.62 (2022). DOI: 10.1126/scirobotics.abk2822.
- [36] Fang Nan and Marco Hutter. “Learning adaptive controller for hydraulic machinery automation”. In: *IEEE Robot. Autom. Lett.* (2024).
- [37] Haedam Oh et al. “Evaluation and Deployment of LiDAR-based Place Recognition in Dense Forests”. In: *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. 2024, pp. 12824–12831. DOI: 10.1109/IROS58592.2024.10801297.
- [38] Sotiris Papatheodorou et al. “Efficient Submap-based Autonomous MAV Exploration using Visual-Inertial SLAM Configurable for LiDARs or Depth Cameras”. In: *arXiv preprint arXiv:2409.16972* (2024).
- [39] Marco Patacca et al. “Significant increase in natural disturbance impacts on European forests since 1950”. In: *Global Change Biology* 29.5 (2023), pp. 1359–1376. DOI: <https://doi.org/10.1111/gcb.16531>.
- [40] Paolo De Petris. “The Resilient Micro Flyer: a New Collision-tolerant Autonomous Robot”. MA thesis. University of Nevada, Reno, 2020.
- [41] Marek Pierzchała, Philippe Giguère, and Rasmus Astrup. “Mapping forests using an unmanned ground vehicle with 3D LiDAR and graph-SLAM”. In: *Computers and Electronics in Agriculture* 145 (2018), pp. 217–225. ISSN: 0168-1699. DOI: <https://doi.org/10.1016/j.compag.2017.12.034>.
- [42] L. Porzi et al. “Seamless Scene Segmentation”. In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2019.
- [43] Milad Ramezani et al. “Online LiDAR-SLAM for Legged Robots with Robust Registration and Deep-Learned Loop Closure”. In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2020, pp. 4158–4164. DOI: 10.1109/ICRA40945.2020.9196769.
- [44] O. Ronneberger, P. Fischer, and T. Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Proc. of the Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2015. URL: <https://arxiv.org/pdf/1505.04597.pdf>.
- [45] Jean-François Tremblay et al. “Automatic three-dimensional mapping for tree diameter measurements in inventory operations”. In: *Journal of Field Robotics* 37.8 (2020), pp. 1328–1346. DOI: <https://doi.org/10.1002/rob.21980>.
- [46] Kavisha Vidanapathirana et al. “LoGG3D-Net: Locally Guided Global Descriptor Learning for 3D Place Recognition”. In: *IEEE Int. Conf. Robot. Autom. (ICRA)*. 2022, pp. 2215–2221. DOI: 10.1109/ICRA46639.2022.9811753.
- [47] David Wisth, Marco Camurri, and Maurice Fallon. “VILENS: Visual, Inertial, Lidar, and Leg Odometry for All-Terrain Legged Robots”. In: *IEEE Trans. Robot.* 39.1 (2023), pp. 309–326. DOI: 10.1109/TR0.2022.3193788.
- [48] Haofei Xu et al. “Unifying flow, stereo and depth estimation”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 45.11 (2023), pp. 13941–13958.

- [49] Angelos Zacharia, Mihir Dharmadhikari, and Kostas Alexis. *Collaborative Exploration with a Marsupial Ground-Aerial Robot Team through Task-Driven Map Compression*. 2025. arXiv: 2509.07655 [cs.R0]. URL: <https://arxiv.org/abs/2509.07655>.