SILVANUS

D4.3 - Demonstration of ground vehicle navigation to and from frontline wildfire spread

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 101037247

Project Acronym	SILVANUS
Grant Agreement number	101037247 (H2020-LC-GD-2020-3)
Project Full Title	Integrated Technological and Information Platform for Wildfire Management
Funding Scheme	IA – Innovation action

DELIVERABLE INFORMATION

Deliverable Number:	D4.3
Deliverable Name:	SILVANUS D4.3
Dissemination level:	CO
Type of Document:	R
Contractual date of delivery:	31/03/2023 (M18)
Date of submission:	20/03/2023
Deliverable Leader:	Thomas Lowe
Status:	Final
Version number:	V0.5
WP Leader/ Task Leader:	WP4 – CMCC F/T4.5 CSIRO
Keywords:	GV, Navigation, Localisation, 3D maps, Foliage density
Abstract:	The aim of the deliverable is to explain in detail the implementation and demonstration of the ground robot technology that has been developed for vehicle navigation to and from frontline wildfire spread, using the tracked TITAN vehicle with CSIRO's navigation hardware and software payload. The deliverable outlines the functional capabilities of the ground vehicles when deployed within forest terrain.

Lead Author(s):	Thomas Lowe, CSIRO, Richard Revak (3MON)
Reviewers:	PEGASO, VTG

Disclaimer

All information in this document is provided “as is" and no guarantee or warranty is given that the information is fit for any particular purpose.

The user there of uses the information at its sole risk and liability. For the avoidance of all doubts, the European Commission has no liability in respect of this document, which is merely representing the authors view.

Document History			
Version	**Date**	**Contributor(s)**	**Description**
V0.1	10.03.2023	CSIRO	Table of content released
V0.2	20.03.2023	CSIRO	First consolidated draft of the deliverable released
V0.3	20.03.2023	3MON	Additional contribution on the ground robot navigation included
V0.4	24.03.2023	CSIRO	Deliverable finalised and released for internal view
V0.5	31.03.2023	CSIRO	Final deliverable draft released for submission

List of Contributors

Partner	Author(s)
CSIRO	Thomas Lowe, Milad Ramezani, Nicholas Lawrence
3MON	Richard Revak

List of acronyms and abbreviations

ACRONYM	Description
AP	Action Point
CA	Consortium Agreement
DoA	Description of Action
DX.Y	Deliverable X. Y (X refers to the WP and Y to the deliverable in the WP)
EAB	External Advisory Board
EC	European Commission
ECAS	European Commission Authentication Service
EU	European Union
EIM	Exploitation and IP Manager
GA	General Assembly
IPRs	Intellectual Property Rights
KoM	Kick-off Meeting
KPI	Key Performance Indicators
PAC	Project Administrative Coordinator
PM	Project Manager
PQP	Project Quality Plan
RP	Reporting Period
SC	Steering Committee
QAC	Quality Assurance Coordinator
QAM	Quality Assurance Manager
QAP	Quality Assurance Plan
SIC	Scientific and Innovation Coordinator
TL	Team Leader
ToC	Table of Contents
WP	Work Package
WPL	Work Package Leader

List of beneficiaries

No	Partner Name	Short name	Country
1	UNIVERSITA TELEMATICA PEGASO	PEGASO	Italy
2	ZANASI ALESSANDRO SRL	Z&P	Italy
3	INTRASOFT INTERNATIONAL SA	INTRA	Luxembourg
4	THALES	TRT	France
5	FINCONS SPA	FINC	Italy
6	ATOS IT SOLUTIONS AND SERVICES IBERIA SL	ATOS IT	Spain
6.1	ATOS SPAIN SA	ATOS SA	Spain
7	EMC INFORMATION SYSTEMS INTERNATIONAL	DELL	Ireland
8	SOFTWARE IMAGINATION & VISION SRL	SIMAVI	Romania
9	CNET CENTRE FOR NEW ENERGY TECHNOLOGIES SA	EDP	Portugal
10	ADP VALOR SERVICOS AMBIENTAIS SA	ADP	Portugal
11	TERRAPRIMA - SERVICOS AMBIENTAIS SOCIEDADE UNIPESSOAL LDA	TP	Portugal
12	3MON, s. r. o.	3MON	Slovakia
13	CATALINK LIMITED	CTL	Cyprus
14	SYNTHESIS CENTER FOR RESEARCH AND EDUCATION LIMITED	SYNC	Cyprus
15	EXPERT SYSTEM SPA	EAI	Italy
16	ITTI SP ZOO	ITTI	Poland
17	Venaka Treleaf GbR	VTG	Germany
18	MASSIVE DYNAMIC SWEDEN AB	MDS	Sweden
19	FONDAZIONE CENTRO EURO-MEDITERRANEOSUI CAMBIAMENTI CLIMATICI	CMCC F	Italy
20	EXUS SOFTWARE MONOPROSOPI ETAIRIA PERIORISMENIS EVTHINIS	EXUS	Greece
21	RINIGARD DOO ZA USLUGE	RINI	Croatia
22	Micro Digital d.o.o.	MD	Croatia
23	POLITECHNIKA WARSZAWSKA	WUT	Poland
24	HOEGSKOLAN I BORAS	HB	Sweden
25	GEOPONIKO PANEPISTIMION ATHINON	AUA	Greece
26	ETHNIKO KENTRO EREVNAS KAI TECHNOLOGIKIS ANAPTYXIS	CERTH	Greece
27	PANEPISTIMIO THESSALIAS	UTH	Greece
28	ASSOCIACAO DO INSTITUTO SUPERIOR TECNICO PARA A INVESTIGACAO E DESENVOLVIMENTO	IST	Portugal
29	VELEUCILISTE VELIKA GORICA	UASVG	Croatia

No	Partner Name	Short name	Country
30	USTAV INFORMATIKY, SLOVENSKA AKADEMIA VIED	UISAV	Slovakia
31	POMPIERS DE L'URGENCE INTERNATIONALE	PUI	France
32	THE MAIN SCHOOL OF FIRE SERVICE	SGPS	Poland
33	ASSET - Agenzia regionale Strategica per lo Sviluppo Ecosostenibile del Territorio	ASSET	Italy
34	LETS ITALIA srls	LETS	Italy
35	Parco Naturale Regionale di Tepilora	PNRT	Italy
36	FUNDATIA PENTRU SMURD	SMURD	Romania
37	Romanian Forestry Association - ASFOR	ASFOR	Romania
38	KENTRO MELETON ASFALEIAS	KEMEA	Greece
39	ELLINIKI OMADA DIASOSIS SOMATEIO	HRT	Greece
40	ARISTOTELIO PANEPISTIMIO THESSALONIKIS	AHEPA	Greece
41	Ospedale Israelitico	OIR	Italy
42	PERIFEREIA STEREAS ELLADAS	PSTE	Greece
43	HASICSKY ZACHRANNY SBOR MORAVSKOSLEZSKEHO KRAJE	FRB MSR	Czechia
44	Hrvatska vatrogasna zajednica	HVZ	Croatia
45	TECHNICKA UNIVERZITA VO ZVOLENE	TUZVO	Slovakia
46	Obcianske zdruzenie Plamen Badin	PLAMEN	Slovakia
47	Yayasan AMIKOM Yogyakarta	AMIKOM	Indonesia
48	COMMONWEALTH SCIENTIFIC AND INDUSTRIAL RESEARCH ORGANISATION	CSIRO	Australia
50	FUNDACAO COORDENACAO DE PROJETOS PESQUISAS E ESTUDOS TECNOLOGICOS COPPETEC	COPPETEC	Brazil

Table of Contents

Index of figures

Index of tables

No table of figures entries found.

Executive Summary

The purpose of this report is to present the technology and functional capability of the robotic platform developed by the CSIRO team to combat against the spread of wildfire and enable firefighters the option to deploy mitigation actions against wildfire. The functional capabilities of the ground navigation vehicle include the capability to autonomously navigate to and from frontline wildfire spread through self-exploration of the forest terrain, to autonomously send data to a base station and then over a REST interface to a SILVANUS server, to autonomously estimate the foliage density in the neighbourhood of the ground vehicle (also to be referred to as ground robot), to autonomously segment the acquired map into individual trees, and to autonomously localise the robot to prior maps.

All the above functional capabilities have been tested, validated, and demonstrated in the presence of smoke, trees, uneven terrain, dusty ground and significant scrub in the field tests.

Additionally, the use of ground commercial solutions to be deployed within the context of SILVANUS has also been highlighted by 3MON.

Following an exhaustive experimentation process that has been carried out, it was noted that the operating temperatures of the ground vehicle was not suitable for the deployment of these robots to the extreme perimeter of the forest fire spread. However, with the installation of the onboard camera and LiDAR infrastructure available within the robot platform, it has been agreed to export the image sequences to the SILVANUS cloud for the execution of the fire detection module using the deep-learning components that have been developed. Following the preliminary integration, the next stage of ground vehicle development will enable executing the deep-learning components at the edge and validate the effectiveness of the fire detection components that are being installed on the ground vehicle platform. Therefore, in the current deliverable, the scope of the demonstrate is limited to commanding the robot to travel a significant distance around the trees to explore towards the expected fire front, we then place a virtual safe boundary on the map and demonstrate commanding the vehicle to explore along the length of this boundary before returning. The vehicle has only limited knowledge of the terrain before starting the mission and must navigate autonomously.

During the mission, the vehicle constructs spatial maps of the visited areas. We demonstrate how these maps are then used for identification of trees and estimation of foliage density in the vicinity of the fire.

1 Introduction

The extremely hazardous environment around forest fires is a motivation for the use of autonomous ground vehicles to replace people in the most dangerous locations leading to the safety of the firefighters. Ground robots have the potential to withstand higher temperatures, to operate for longer and in larger numbers than humans, at less risk to human lives. To complement the capabilities of Unmanned Aerial Vehicles in fires, Unmanned Ground Vehicles provide a longer battery duration, they are unaffected by the heavy winds that often accompany fires, they do not require flight authority and do not present a falling hazard for any nearby workers.

The challenge with autonomous ground vehicles is that the same hazardous environment that motivates their use also means it is a challenging environment to navigate and negotiate. Specifically, forested environments contain complex undergrowth, fallen trees, uneven and loose terrain, and dynamic objects such as compliant and swaying branches. Moreover, in wildfire scenarios there is the potential for mud (if water has been used on the fire), smoke, dust and wind, in addition to heat. These all present their own challenges.

2 Deliverable

In this deliverable we target the challenge of how to autonomously navigate a robust ground robot from a human-safe location up to a robot safe-distance zone from an imagined fire. We currently cannot operate in the presence of a real fire but make use of the on-board return images and 3D map to allow a human at the base station to judge where this safe zone boundary would be. The operator then directs the robot to explore along this safe distance boundary for a distance, and then to return to the start point.

We present the technologies required in this task, then we go on to present the following additional capabilities in their own sections:

1. The capability to localise the robot to a previously generated map of the environment. This is necessary with any repeated inspection of the fire zone, and for whenever the robot is recharged or otherwise moved and needs to return to the scene.
2. The capability to analyse the generated 3D maps to identify individual trees and their foliage density. This provides tree and leaf density information, which informs the user on terrain navigability and is an indicator of the potential rate of wildfire spread through the canopy.
3. The capability to send images and location information back to the base station, and automatically pass this over the REST interface to a server, such as the SILVANUS platform.

3 SILVANUS UGV development

3.1 CSIRO Navigation Capability

In the target application of the SILVANUS project, CSIRO is developing a robot system that can traverse natural environments in order to perform wildfire threat assessment (including tasks such as foliage density estimation for assessing the amount of potential fuel) and make direct observations of fire-affected areas. The system should offer some autonomous capabilities, such as basic navigation and exploration, as well as allowing for direct input from a human operator in real time. The system should be capable of operations in natural, unstructured outdoor environments in the presence of dust and potentially GNSS-denied conditions.

Navigation in complex, especially natural, environments remain a significant robotics challenge. Deploying a robot in a novel natural environment requires:

1. Physically capable hardware (the robot platform itself),
2. Appropriate perception systems for perceiving and interpreting the environment in a variety of conditions, and processing this data into a suitable representation, and
3. Autonomy software for planning and decision making to safely navigate the robot through the environment.

CSIRO has a long history of developing robot systems for operating in complex environments. Relevant recent efforts include CSIRO's participation in the DARPA Subterranean Challenge, a competition event that required a heterogeneous robot team controlled by a single human operator to autonomously explore an unknown underground environment [1] . The effort for this project led to the development of a complete robot system capable of operating multiple vehicles in natural and man-made underground environments, including terrain complexities like smoke. The team used a mix of ground (both tracked and legged) vehicles and aerial systems. SILVANUS will build on this progress to operate in more complex outdoor environments and develop tailored forest analysis algorithms for assisting wildfire prevention and control.

3.1.1 Robotic platform

Some of the physical robot systems used in the subterranean challenge are well-suited to complex outdoor environments. The BIA5 OzBot Titan All Terrain Robot (ATR) is a tracked robot ground vehicle, designed for operations in challenging environments. The system is powered by $LiFePO_4$ batteries, has a total mass of approximately 90kg, and can travel across rough terrain (step height up to 25cm) with typical operating speed up to 0.75 m/s. The Titan ATR is capable of traversing light forest, long grass and uneven natural terrain. Due to the extensive testing conducted with these robots, they are now considered among the most robust and capable platforms in the CSIRO fleet. The ATRs are expected to act as the primary demonstration vehicles for the SILVANUS project. Section 4 shows several images of the Titan ATR and the operator ground station deployed in the field demonstration.

In addition to the mobile platform hardware, the system being used at CSIRO for the SILVANUS project includes a wireless networking solution and a base station for controlling multiple robots with a single human operator. The networking solution utilises deployable battery-powered nodes to construct a mesh communication network in-situ, that allows robots to communicate in real time with the base station. This system could be modified to make use of alternative networks such as terrestrial phone networks when they are available.

3.1.2 Perception system

As part of the efforts for the Subterranean Challenge, significant research and development was focused on creating a modular perception system to be used on a variety of robotic platforms in a range of environmental conditions, capable of operating without global positioning systems. This research led to the development of the CatPack, a self-contained perception system consisting of multiple visual- and infrared-spectrum cameras, an inertial measurement unit (IMU), a spinning LiDAR and associated computer hardware for processing data online. The CatPack can be fitted to different platforms and provides out-of-the-box mapping and localisation capabilities using on-board sensors and processing.

The CatPack system utilises CSIRO's Wildcat Simultaneous Localisation and Mapping (SLAM) algorithms, that offer class-leading mapping and localisation performance based on LiDAR and IMU data [2] . The system provides low-drift, sub-metre position and map estimation in suitable conditions.

Although a detailed description of the algorithmic basis of Wildcat is beyond the scope of this report, a brief description may aid in understanding the advantages and limitations of this approach. Wildcat is essentially an optimisation process that attempts to reconcile the spatial LiDAR measurements with the motion data captured by the IMU. The problem is treated as a graph, where edges describe the relationships between recorded data and the estimated trajectory of the robot. Laser data is processed into a spatially-voxelised representation, where each occupied voxel contains a surfel (an estimated surface element characterised by location and shape), that summarises multiple laser returns. The algorithm performs optimisation over the constructed graph, alternating between minimising error on the estimated trajectory of the robot and aligning new measurements to the existing voxel map. Further details can be found in [2] .

Although the system was primarily designed for use in rigid environments, the CatPack has demonstrated strong localisation and mapping performance in natural terrain. Creating more flexible map representations that consider the deformable nature of most natural terrain remains an open challenge that will be further explored later in this project.

3.1.3 Autonomous navigation

Given the localisation and map estimates from the CatPack perception system, the on-board navigation stack processes this data into representations that can be used for planning and executing safe paths and exploring previously unseen terrain. The system is based on the Robot Operating System (ROS) and was designed as a modular navigation system that can be applied to a wide range of platforms. The system has been successfully deployed on a variety of platforms, including custom hexapods, third-party commercial legged platforms and large tracked and wheeled vehicles. Further, the navigation system can operate with or without direct communication to the base station.

The goal of the complete human-robot system is to combine the skills and experience of the human operator with the capabilities of the robot team. Meeting this goal requires an interface that provides the operator with a clear understanding of the robot and the remote situation and allows them to command the robot at a range of autonomy levels. Currently, the human operator is able to interact with the robot from low-level velocity commands through a gamepad up to high level behaviour commands. The ground station offers a real-time view of the robot's scene understanding through visual displays such as point cloud and traversability estimates, robot status information and direct feedback including streaming video. Examples of the user interface in practice can be seen in the demonstration example later in this report. The base station provides the operator with an awareness of both the robot and the environment, and the ability to intervene when required.

In the SILVANUS project, the onboard navigation stack makes use of a behaviour-based approach, that allows the operator to request a range of higher-level behaviours such as exploring a new region, travelling to an operator-defined target position, or returning to communicate information with the base station. The operator can command multiple sequential or simultaneous tasks and assign priorities to each. Tasks are then automatically allocated either to different robots in the team or to a single robot based on capabilities and current status (such as completing nearby tasks first in order to maximise efficiency) using a market-based consensus voting strategy. For the SILVANUS project, this allows a operator to, for example, command the robot to reach a specified location, then explore a target region to identify fire affected areas or dangerous fuel loads, and then return to base to report this information. The operator can update or modify tasks for any robot in communication range at any time.

Addressing the behaviour goals requires the robot to autonomously navigate through the environment. The maps constructed by the Catpack perception system are further processed to estimate traversability of the terrain for local and global planning. Again, a detailed discussion is beyond the scope of this report, but further details can be found in [3] . The system is capable of operating in complex terrain, and can autonomously plan safe paths, update them as new information arrives, and replan when required.

3.2 CSIRO Re-localisation Capability

Bushfires are a recurring problem in many parts of the world. These devastating fires are difficult to control and can have catastrophic consequences. In recent years, there has been increasing interest in using robots to help fight these fires. However, one significant challenge in using robots in this context is the robots' "wake-up" problem or re-localisation.

The “wake-up” problem in robotics refers to determining a robot's current location (re-localisation) within an area of the environment that has already been mapped. This is an essential task for robots, as it allows them to continue operating their tasks even if they have temporarily lost track of their position. One common example of the “wake-up” problem is where a robot that has been turned off or lost power must determine its location upon being restarted. Robots wake-up solutions often involve a combination of sensory data, such as from cameras or lidar, and prior knowledge of the environment, such as maps or landmarks.

3.2.1 CSIRO re-localisation approach

In fire hazards, the smoke and haze can significantly degrade the visibility, making robots re-localisation challenging for vision-based systems to operate effectively. In contrast, lidar uses laser beams to create 3D maps of the environment, which is less susceptible to such environmental disturbances. Hence, our re-localisation approach is based on lidar measurements.

Our prior map, created by Wildcat SLAM [2] , is a pose graph consisting of robots’ poses (nodes) and the edges in between. In short, Wildcat integrates lidar and inertial measurements within a sliding window localisation and mapping module. Wildcat generates undistorted sub-maps which are further stored as the prior map. The prior map can be loaded at the control station or on the robot’s computing unit to be used for re-localisation of robots exploring within the prior map.

Providing the prior map as described above, the re-localisation problem can be defined as finding the maximum similarity between the newly generated sub-map, namely query point cloud, by comparing it with the point clouds in the prior map. To this end, we train a deep lidar place recognition model encoding point clouds into feature space. Our network allows us to generate descriptors for point clouds which are invariant to rotation, viewpoints and temporal changes in the environment. Additionally, the CNN-based architecture of our model is quite light, making the model suitable for online operation. For optimum performance in unstructured environments (forests which is the target environment in Silvanus), we trained our network in a large-scale bushwalk dataset [4] . The pre-trained model along with dependencies were included in a ducker image which can be easily deployed on the robots. For re-localisation, once the most similar point cloud from the prior map is selected (course localisation), our network can create several semantically local features with their positions that we use for point cloud registration (fine localisation), estimating the relative transformation between the query point cloud and the top-candidate point cloud. The following figure displays the re-localisation process between few query point clouds (in orange) and the top-candidate point clouds (in cyan) from the prior map. The corresponding local features are linked with green lines.

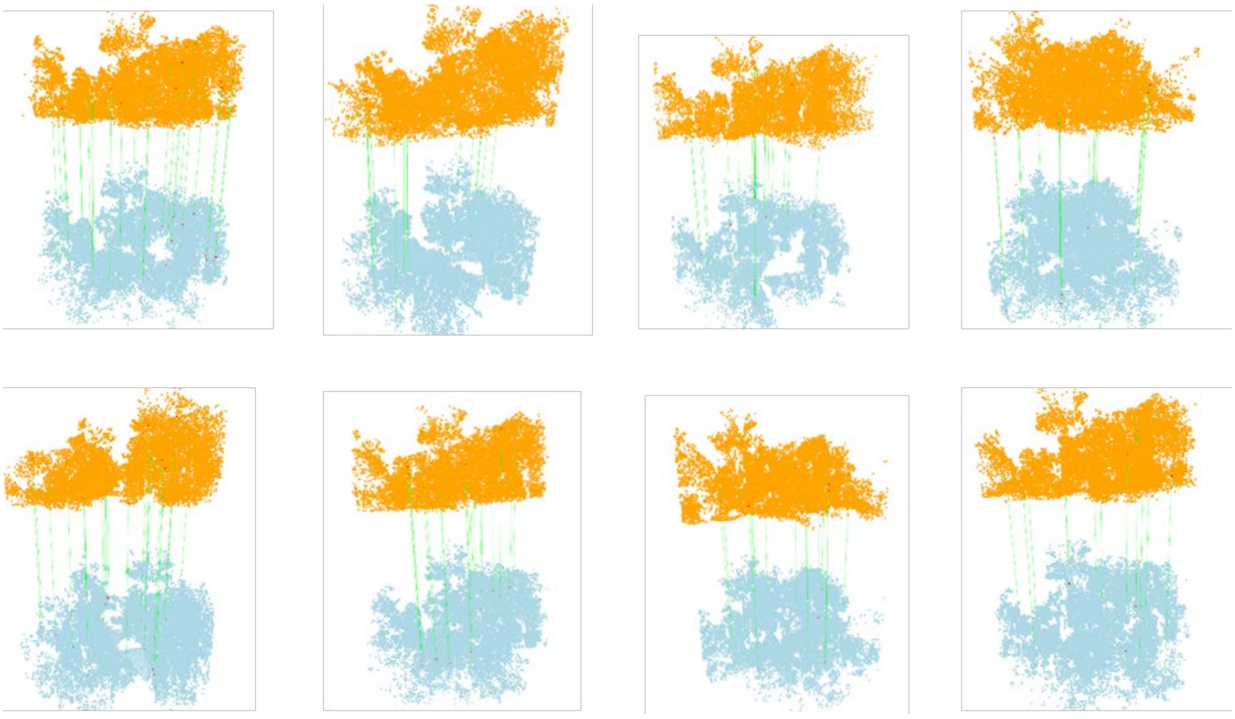


Figure 1 - Showcases of submap registration after place recognition using our deep lidar re-localisation pipeline. the cyan cloud indicates the top candidate from database, while the cloud visualised on top is the query point cloud.

3.2.2 CSIRO Re-localisation showcase

Our re-localisation module communicates with our lidar SLAM system Wildcat through Robotics Operating System (ROS). For the robot exploring the environment, upon requesting re-localisation, the query sub-map (generated by Wildcat) and the sub-maps existing in the prior map are fed into the model. By performing a forward pass using the weights and benefiting from a fast search approach such as kd-tree, the top candidate is selected and further the transformation relative to the top-candidate coordinate system is estimated. Figure below demonstrates the incorporation of the re-localisation module with Wildcat. When a robot moves in a known environment (the map generated from any early session) (left), upon a successful re-localisation (middle), the current pose graph is merged into the prior map (right), allowing the robot to operate concerning the prior map for resuming unfinished tasks.

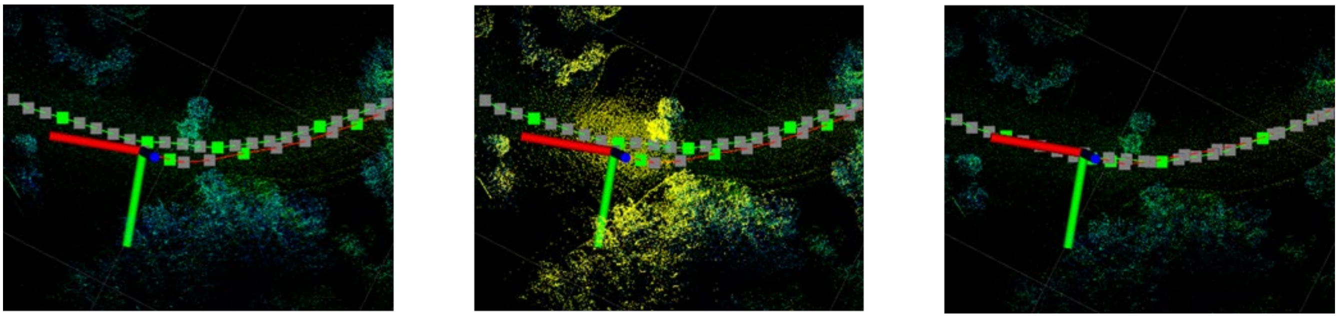


Figure 2 - Wake-up showcase. Left: The robot (indicated by the RGB axes) moves within the prior map containing nodes (children in grey and roots in green). Middle: Robot wakes up by re-localising with respect to the prior map. Right: Upon re-localisation, the robot merges into the prior map contuning the remaining tasks.

Our re-localisation system can be used to merge the pose graph created by the operating robot (path shown in red in the following figure) into the prior map, allowing the robot to work in the same coordinate system.

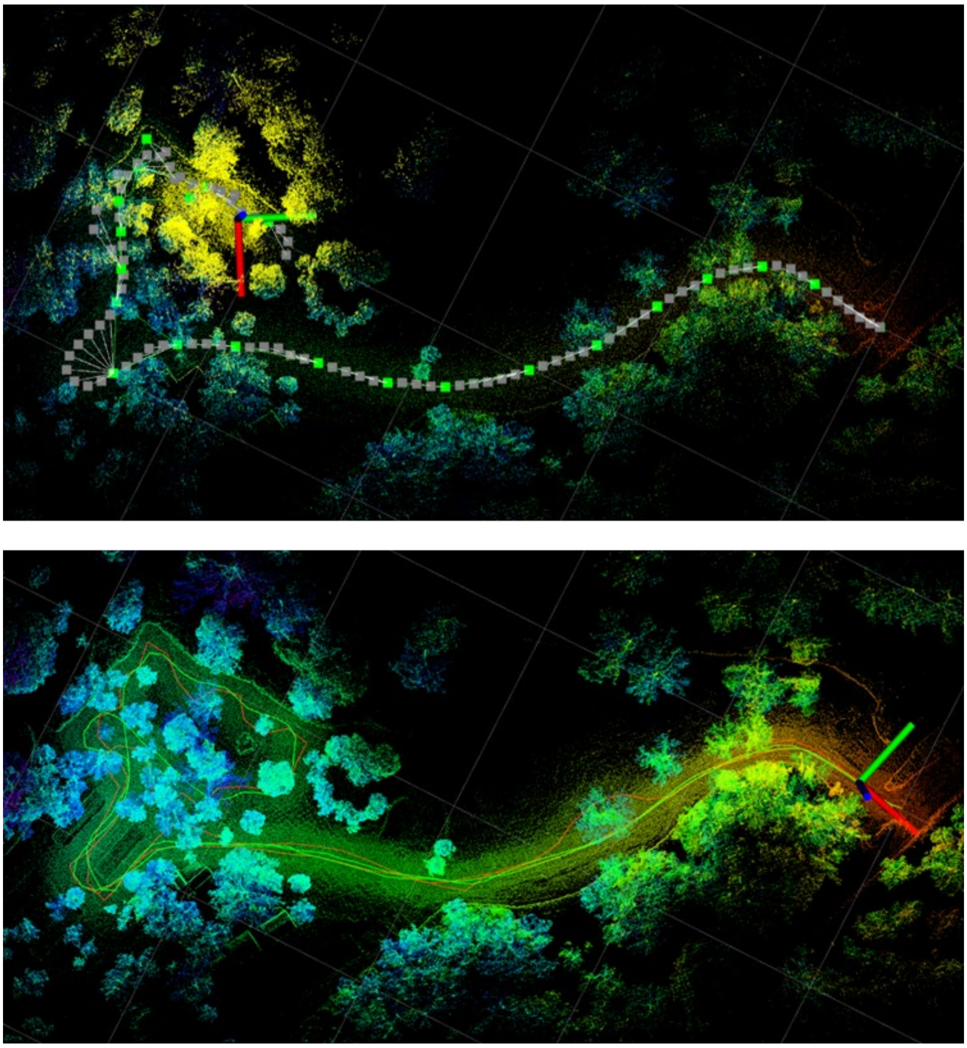


Figure 3 - Operating robot re-localises itself with respect to the existing map. Bottom: Upon re-localisation, the current map will merge into the existing map resulting in a globally consistent map.

3.3 CSIRO Forest Analysis Capability

When the robot has returned back to the human-safe area, the 3D map can be downloaded and analysed to provide indices representing the foliage density and tree density across the map. This derived information is helpful in understanding the rate and risk of canopy fire spread, and in assessing the navigability of each area. This is because high densities of trees are less navigable than sparser areas. In later development it is planned for the analysis to be available on-the-fly, but in this deliverable, it is performed after the robot has returned.

The lidar map is a point cloud (a set of 3D surface points observed by the onboard lidar) together with an estimated robot trajectory. Together these are converted to a ray cloud, which is the set of rays from the lidar source to the observed surface point.

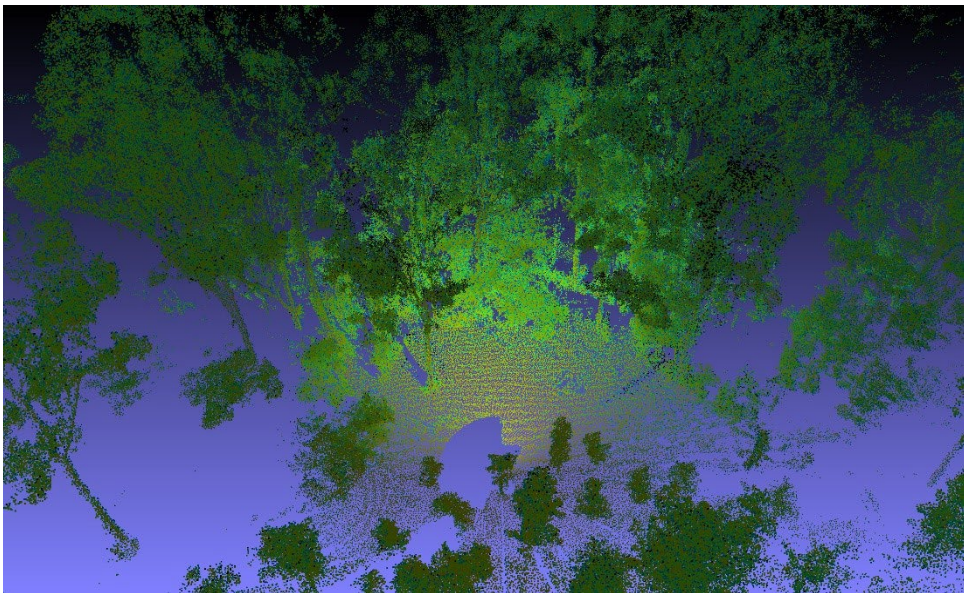


Figure 4 - Ray cloud of forested terrain obtained from UGV at our QCAT site.

We are then able to process this ray cloud with the open source raycloudtools library [5] that we continue to develop at CSIRO. The latest developments provide most of the tools required to analyse the forest. Since they are built as command line functions, they are easily applied in sequence using bash scripting. We now describe these commands approximately in the order that they are applied:

rayextract terrain

This command reconstructs the ground as a triangular mesh. This task is challenging due to the wide range of point densities within the cloud. We reconstruct the ground by applying a conical kernel to the point cloud as depicted below. Every point that the cone contacts from below becomes a vertex of the ground mesh.

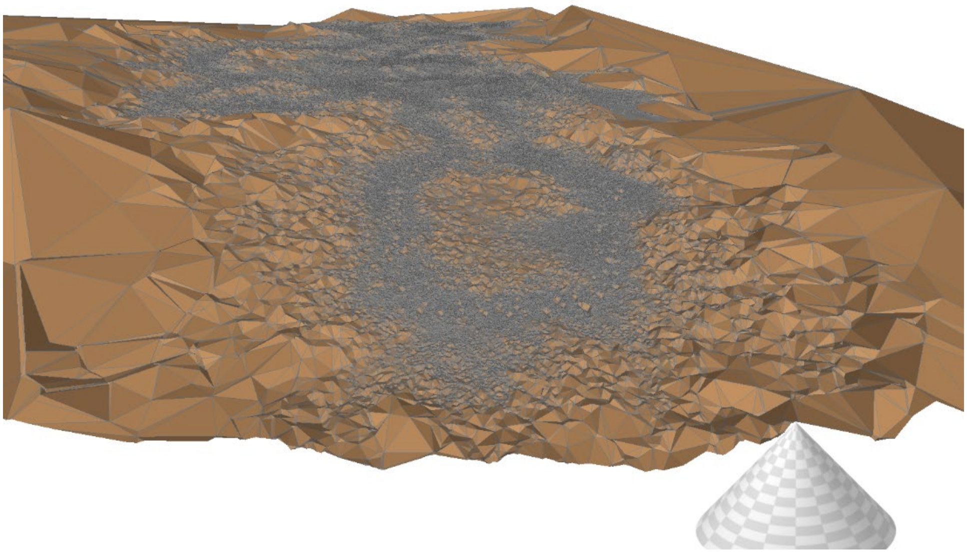


Figure 5 - Example triangular mesh ground generated from the point cloud points that contact the cone.

This technique is insensitive to point density and only requires one parameter – the cone gradient – which we keep as 1.

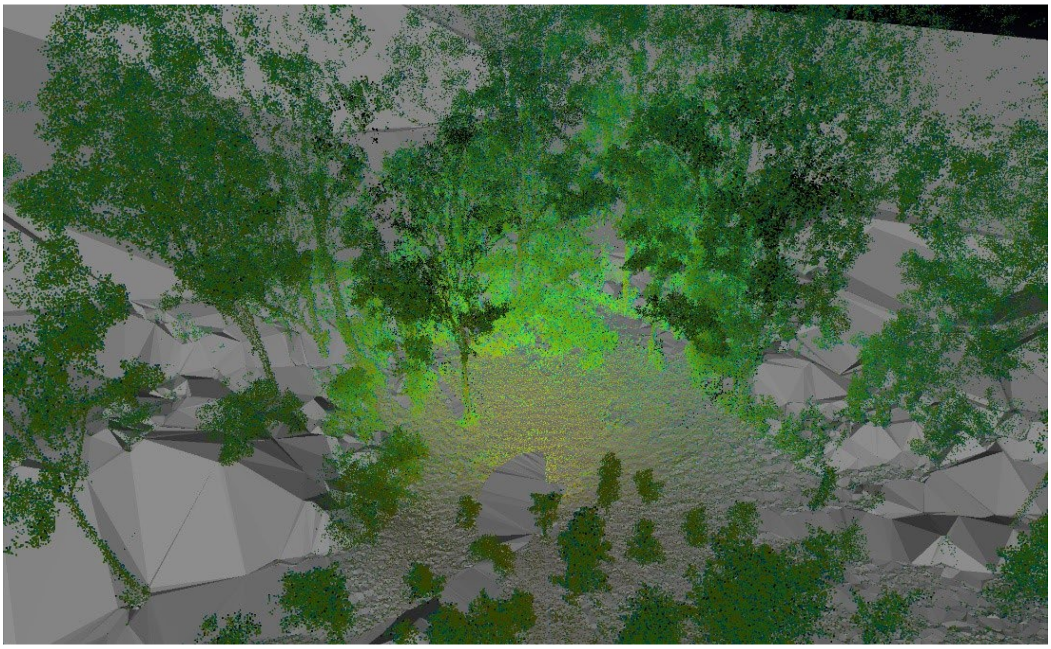


Figure 6 - Forested terrain point cloud overlaid on the reconstructed ground terrain mesh.

rayextract trees

This command segments the ray cloud, using the ground terrain as a reference, such that there is a unique point colour for each independent tree.

Trees are designated as the structures that reach above a specified height from ground level (here 1.5 m), all lower structures are treated as undergrowth or terrain. Dijkstra's shortest path algorithm is then employed from each ground point, in order to follow the lidar points up the trees and along their branches. This provides a robust means of segmenting the trees that is faster than 3rd party solutions (such as TreeSeg) and does not require the trees to be clearly spatially separated.

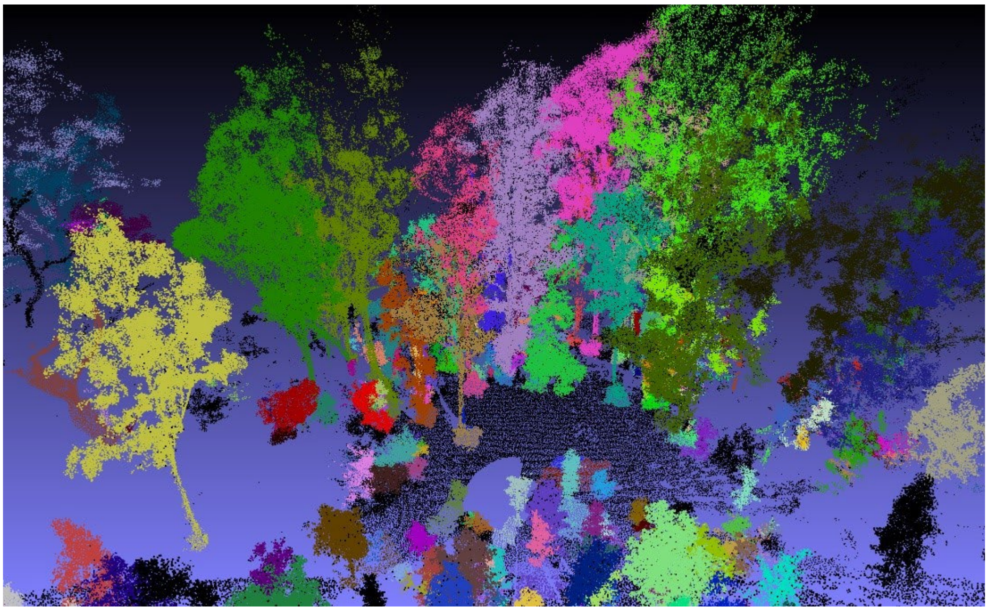


Figure 7 - Forested terrain point cloud, coloured by individual tree, in the presence of dense canopy overlap.

This tree segmentation can be used to estimate the number of trees per square metre in the operational area, which is indicative of both fire spread risk and of traversability risks.

Another factor indicating fire spread risk is foliage density. We demonstrate the work on this front with the following commands.

raycolour branches

This command colours the ray cloud by two separate indicators of the points belonging to a branch rather than a leaf.

Firstly, the return intensity from the lidar is used. For the Velodyne VLP16 lidar that we use, the return intensity is already normalised for distance, so approximates the reflectivity of the surface. There is typically a different degree of reflectivity of the lidar's ~900 nm wavelength light between foliage and wood/bark. We convert this measure into the red colour channel for each point.

Secondly, we estimate the cylindricality of the neighbouring points around each point in the cloud. Nearby points that are cylindrical are more likely to represent a branch than points that have a more spherical distribution. We measure this cylindricality by a novel application of the second moment of the scatter matrix of the neighbouring points. This `area measure' is intermediate between the trace and the determinant of the matrix, and allows us to measure cylindricality without performing an expensive eigenvalue decomposition. This cylindricality index is converted to the green colour channel per point.

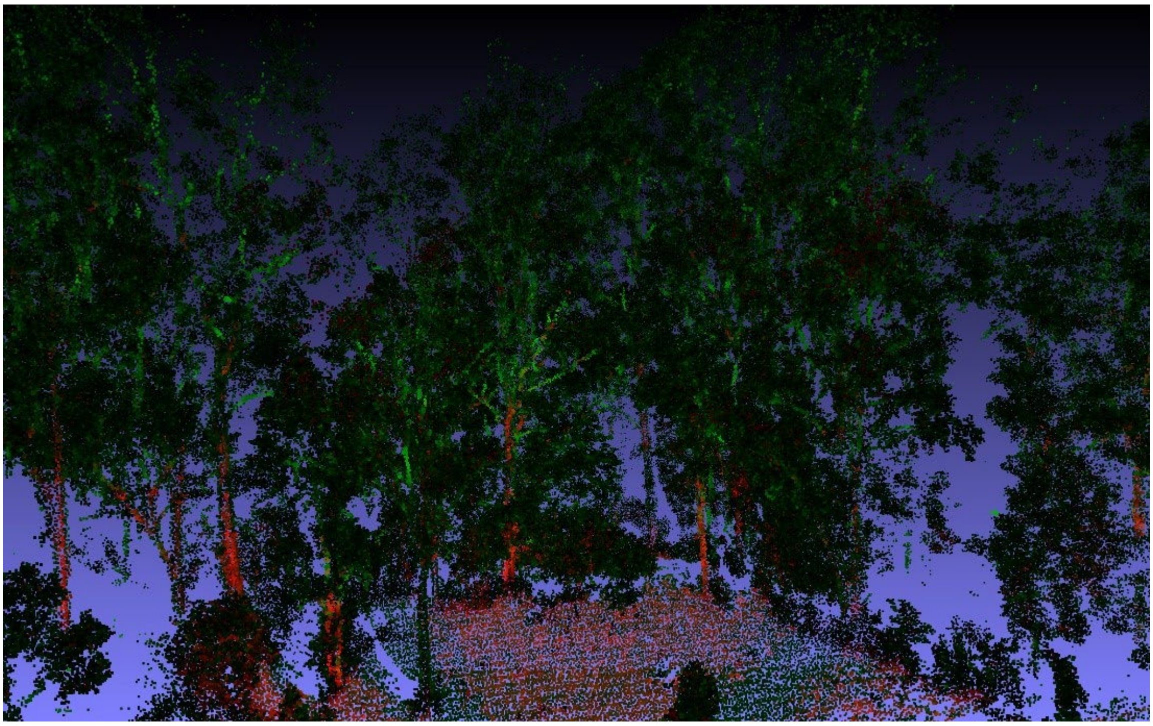


Figure 8 - QCAT forest cloud coloured by return intensity (red) and cylindricality (green).

The reason for the two indices is that return intensity is only reliable on the wider branches and trunks, while cylindricality only works well on the thinner branches. The two indices therefore complement each other and we can now remove most of the woody material from the ray cloud by cutting out points with a given linear combination of the red and green channels.

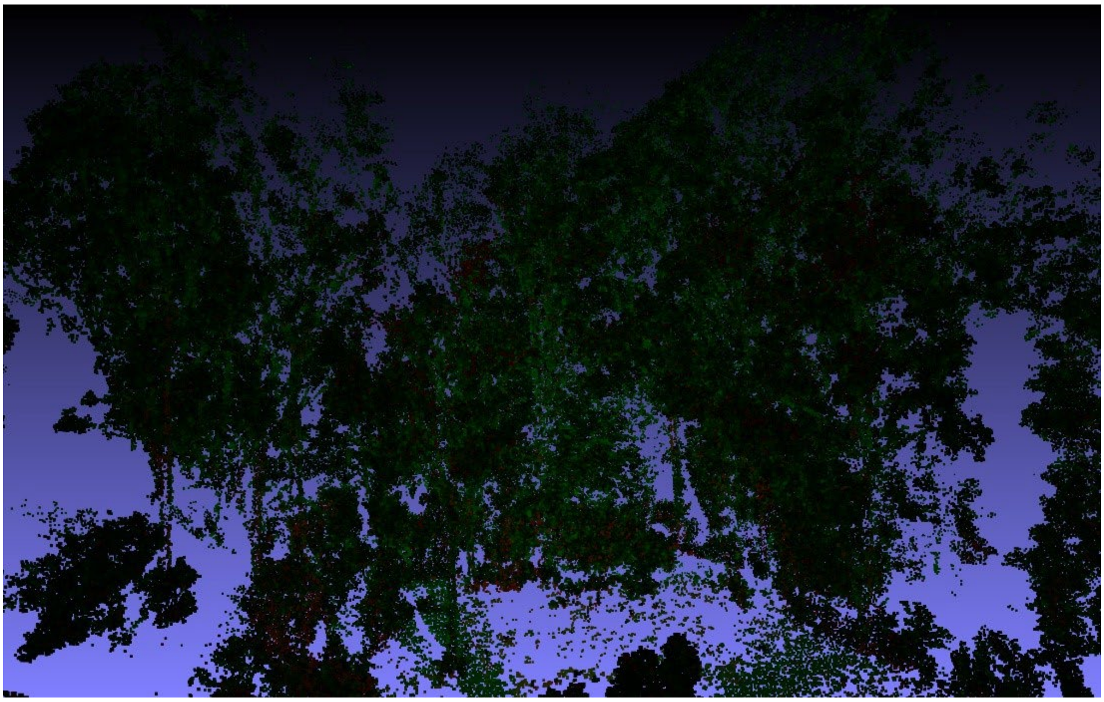


Figure 9 - Removing all points where r+g > 0.3, using the raysplit colour command leaves primarily foliage and ground.

raysplit ground distance

We are almost ready to estimate the canopy foliage density in the map, however the remaining points also contain undergrowth and terrain. We use this command to remove all points that are within a specified distance of the ground. We used 1.5 m in this case.

rayrender top density_rgb

This command renders the remaining ray cloud (canopy foliage) in plan view by the estimate of foliage density. Foliage density is measured in one-sided leaf area per volume, and the rendered image integrates over the vertical axis, which gives the one-sided leaf area per horizontal area, also known as the Leaf Area Index (LAI). The lowest LAI values are rendered transparent in order that the image can be overlaid on any existing map or satellite image of the area.

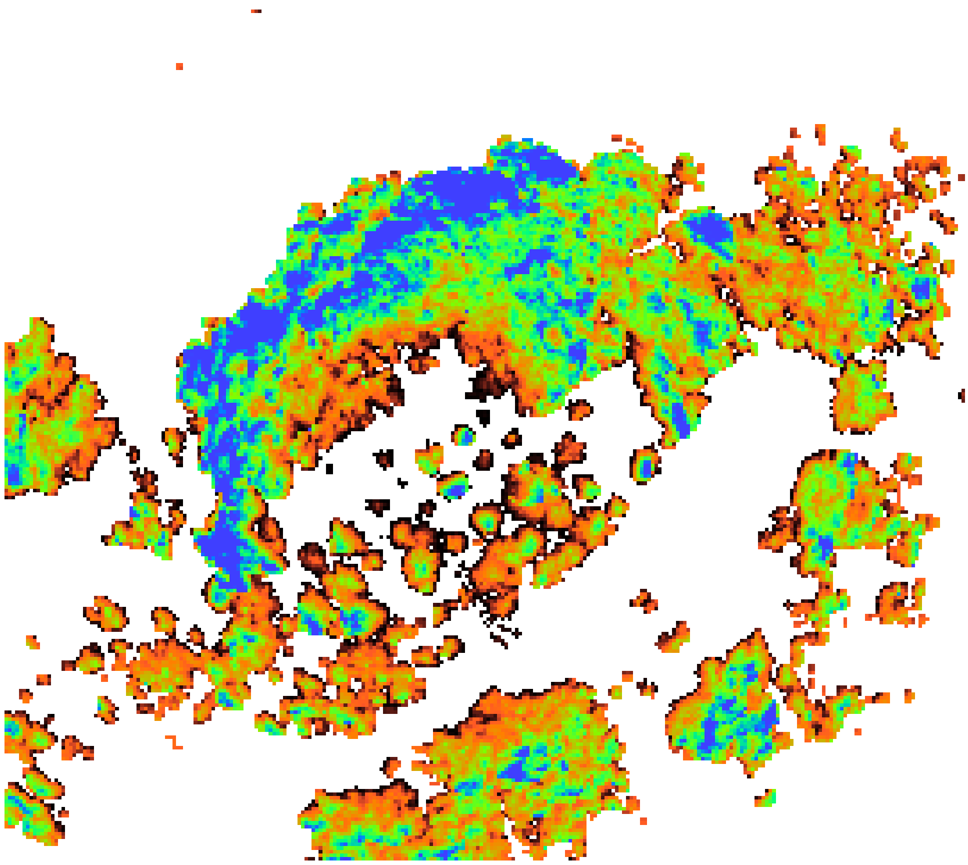


Figure 10 - Leaf Area Index estimate of canopy foliage, from red to blue. The crescent of blue represents the crescent of tall, dense trees in the previously shown QCAT ray cloud.

The foliage density is estimated per voxel of the voxelised space, using the formula that we validated in the research paper [6] :

$$\rho_c = g\frac{(n-1)m}{n\Sigma_{i=1}^{n}x_i}$$

Here g is a constant representing the leaf angle distribution, we use a spherical distribution model where g=2. n is the number of rays entering the voxel and m is the number of rays that end within the voxel (so intersect a leaf). x_i is the path length of the ray from where it enters the voxel to where it ends or exits the voxel.

In cases where voxels have very few intersecting rays, the numerator and denominator of the above ratio are both small and so the estimation is inaccurate. We compensate in the case where n<10 by including contributions to the numerator and denominator from the neighbouring voxels until n=10. This ensures a minimum accuracy at the expense of spatial fidelity in these few areas. The frequency of low-accuracy voxels is reported back by the rayrender command so that the user is made aware that the data is too sparse to be reliable.

3.4 3MON

The company 3MON is using robot Colossus that has been made by company Shark Robotics as presented in Figure 11. This robot is not able to operate independently and must be operated by human or programmed to move according to waypoints (the waypoint navigation solution is available only for military version now, but according to results of the pilot in Slovakia, it would be integrated to civil robots made by Shark Robotics too). The robot can by remotely controlled with a distance up to 500 m. The robot is equipped with a 360° thermal camera and optic camera with zoom with video transmission to the operator. It can be also equipped with a CBRN detector.

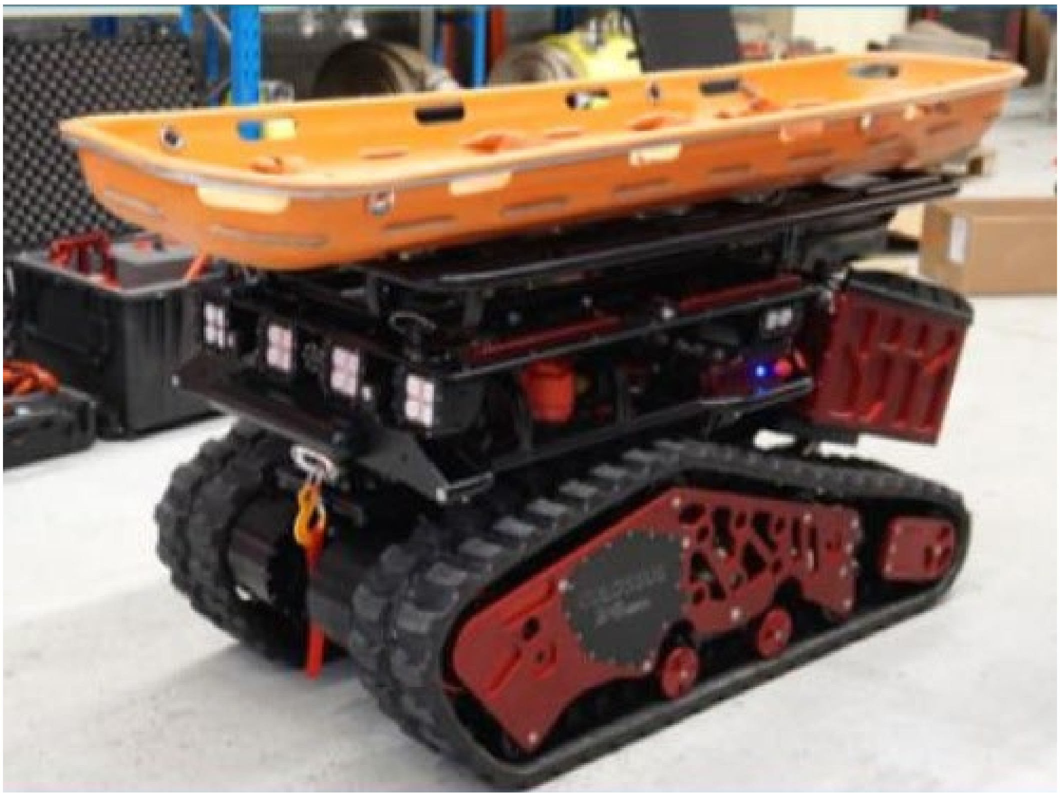


Figure 11 - Robot with stretchers for transporting victims

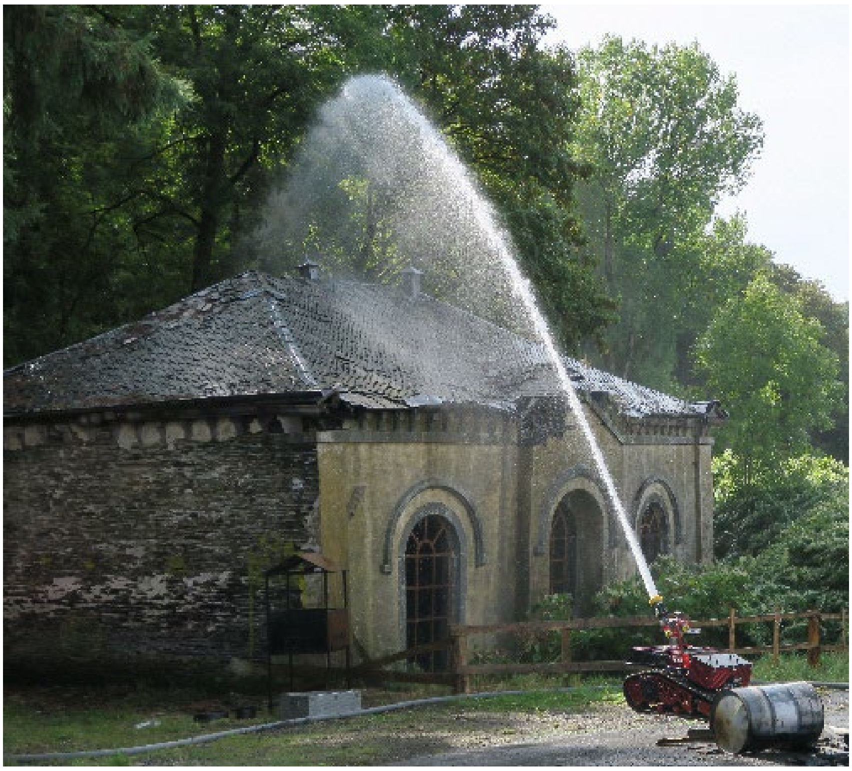


Figure 12 - Robot with water cannon

For the forest fires it can by used for firefighting but also for hauling fire equipment (fire houses, hand tools…) or injured person on stretchers as showcased in Figure 12 . The maximum payload of the robot is 500 kg. For pulling or pushing the payload can be up to 1000 kg. The robot can be used also for clearing path for fire trucks from fallen trees or other objects. For firefighting it can by quickly reequipped with water cannon with flow up to 3000l/min and can be used to protect buildings or infrastructure from the fire front and can withstand temperatures up to 600°C.

The outcome of the pilot and experiences of the deployment of the robot on the pilot will determine possible additional development of the robot to be more suitable for forest fires.

4 Navigation demonstration and performance

In this section we demonstrate the robot capability. We place the robot in an area with many young trees, with scrub and grass, earth and some rocks making up the terrain. In this image the rotating Velodyne lidar sensor pack can be seen mounted on the tracked ATR vehicle

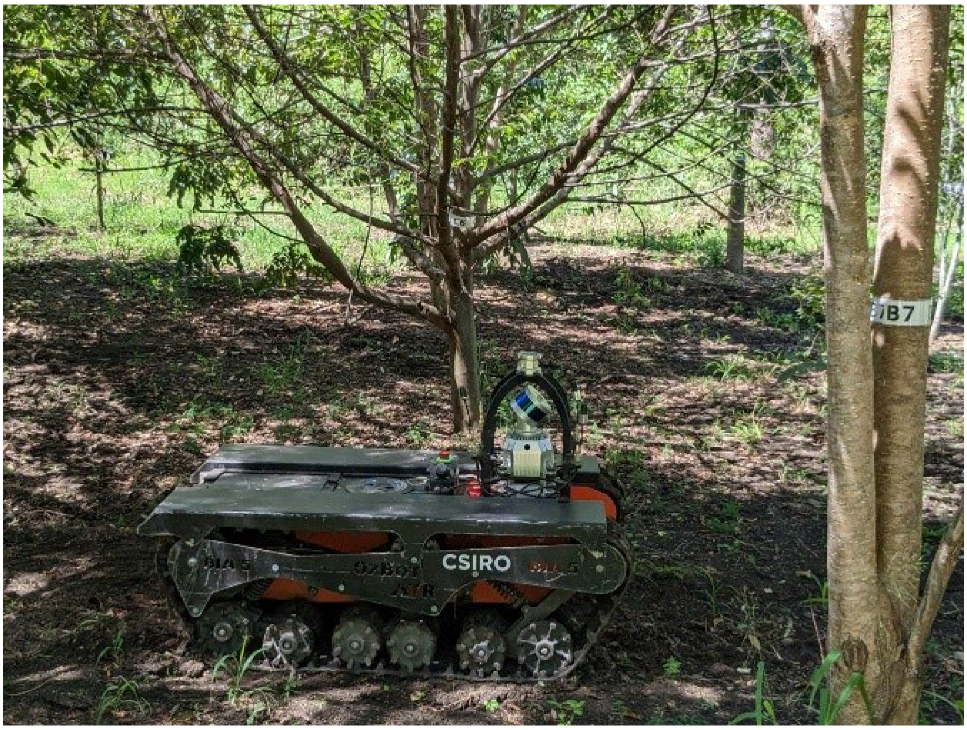


Figure 13 - The tree, scrub and grass environment used in this demonstration. The sensor and navigation payload can be seen on the right side of the ATR.

All UGV commands are sent from the base station at a safe distance from the fire front. The base station does not need to have line of sight to the vehicle as it includes images from the vehicle's front camera and visualises the robot location using the 3D lidar map, shown in plan view.

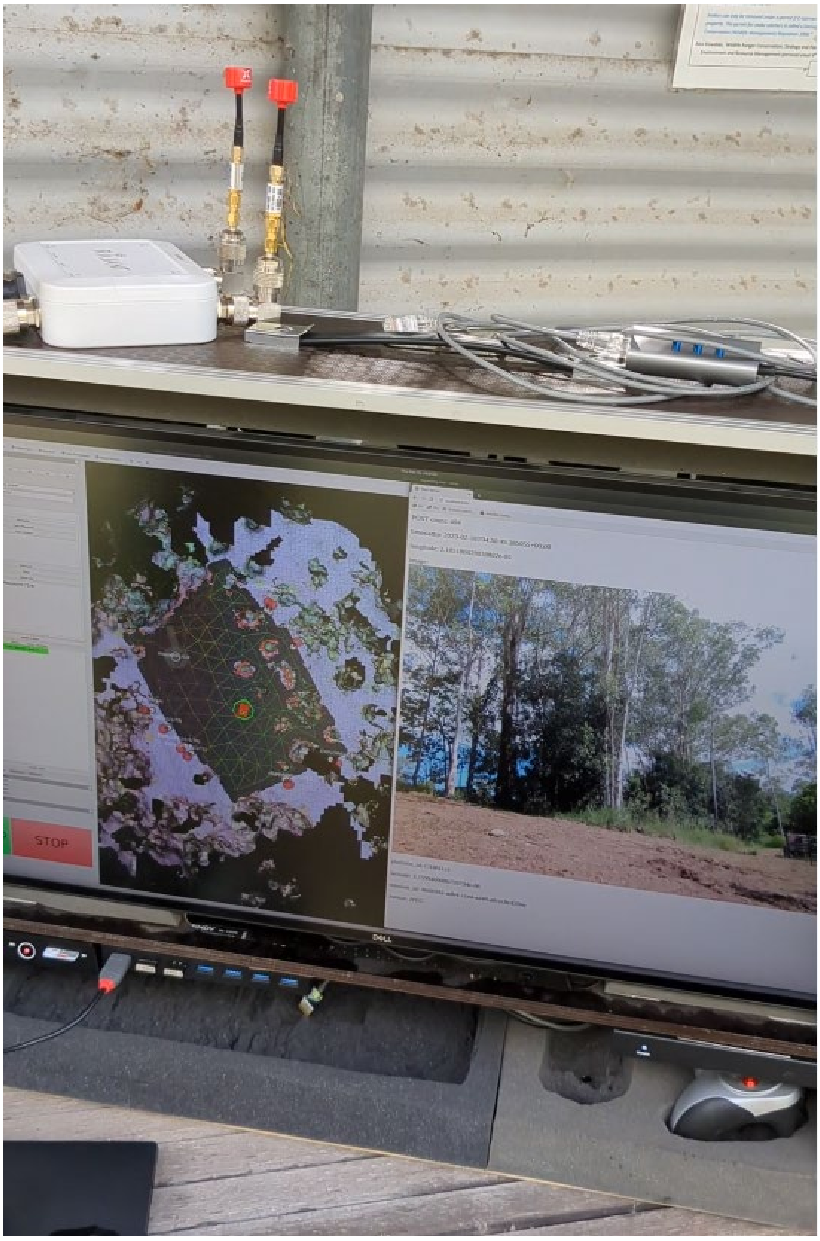


Figure 14 - Base station showing command interface (left), map plan view (middle) and front camera view (right). The communications link is above the screen, and teleoperations controller shown bottom right.

We represent the closest that the robot can get to the fire front by the following orange barrier

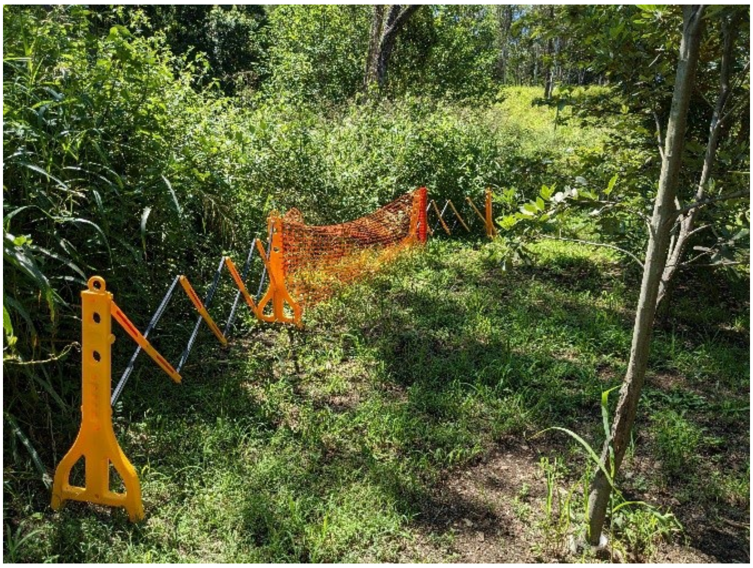


Figure 15 - Demonstration of the ground robot proximity to the wildfire

The robot is powered up at approximately 40 m from this front line. The operator at the base station is then able to see the local map generated by the robot and direct it to navigate in the direction of the fire front. This single request placed on the map is sufficient for the robot to navigate through the environment:

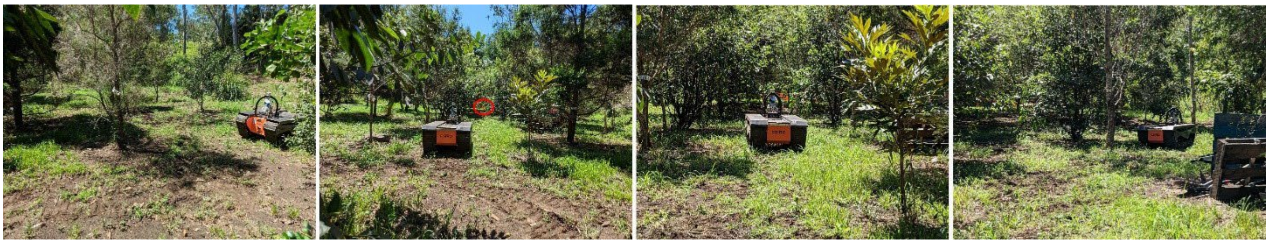


Figure 16 - Left: start location, followed by sequence towards the fire front. The distant fire front is circled in red in the second image.

The UGV is building a map of the environment as it traverses it. This means that it re-plans its path to the target on the fly, and has to turn around if a route turns out to be a dead end on closer inspection.

There will always be situations where the robot reaches a tight spot and cannot find a way to proceed. This occurred half-way through the outbound journey and so the UGV stopped and awaited further instructions.

Our interface is capable of addressing these situations. The UGV's state is observed by the operator at the base station, and can either manually specify a sequence of locations to escape the tight area, or teleoperate the robot using a control pad. In both cases the operator does not need to observe the robot through line of sight, they have the video feed from the forward camera available on-screen together with the plan-view map of the area and its obstacles.

In this demonstration the vehicle was not in sight of the operator and the operator chose to teleoperate the robot approximately 4 m to escape the tight area, and then the mission was continued.

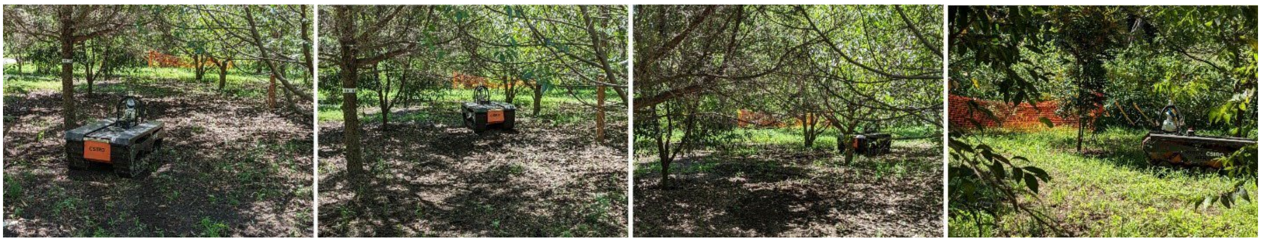


Figure 17 - UGV approaching fire front and reaching it (right).

In a real fire situation the end location may have various objectives. It may be the closest point that is safe for the UGV to operate at, it may be a location that is repeatedly updated by the operator in order to observe the fire, or to look for signs of smoke. Or it may be a broad location that needs to be mapped. In each of these cases it is helpful for the robot to be able to explore this hazardous region once it has arrived. We therefore now run a short exploration command for the robot now that it has reached the destination.

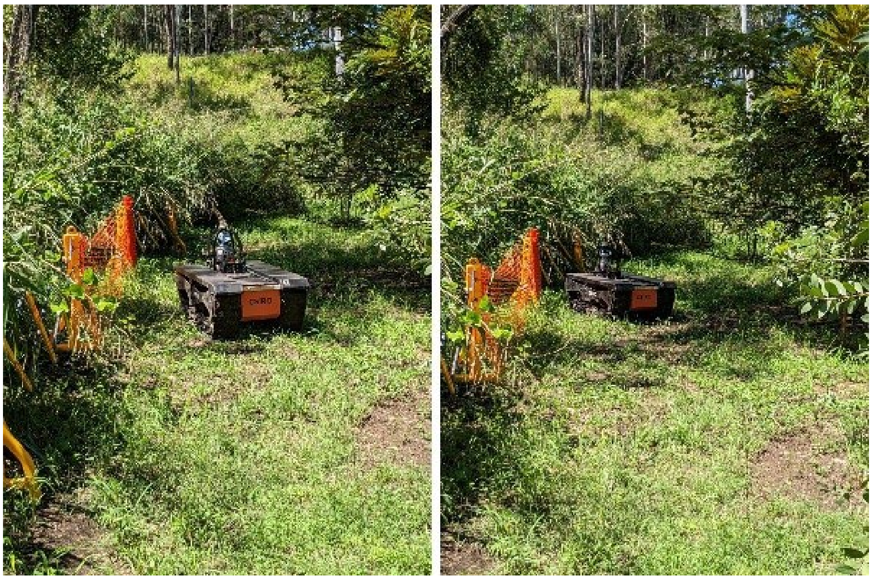


Figure 18 - Explore mode seeks areas that have not been mapped.

Once the operator is satisfied that the UGV has observed what needs observing, then the single command is sent to return to the start location. In this demonstration the robot navigated back without getting stuck or having to turn back. This is primarily due to the robot now having a map of the region between start and destination, so any re-planning would only be required if the environment has changed since the outbound traversal.

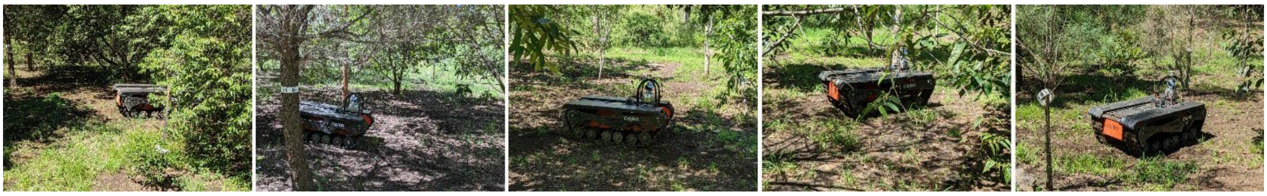


Figure 19 - Return traversal of the UGV was fully autonomous and without any back tracking, due to the outbound acquired map.

The following image shows the base station visualisation on the return journey. The command interface is the left panel, and the requested return point is seen as the grey disk and arrow in the central plan-view map. This map also contains the vehicle (red) and its planned path through the terrain (red path). The currently observed vegetation is in purple, and its estimated unexplored regions and directions are shown as white and red spots. The fire front was at the rightmost point of the path.

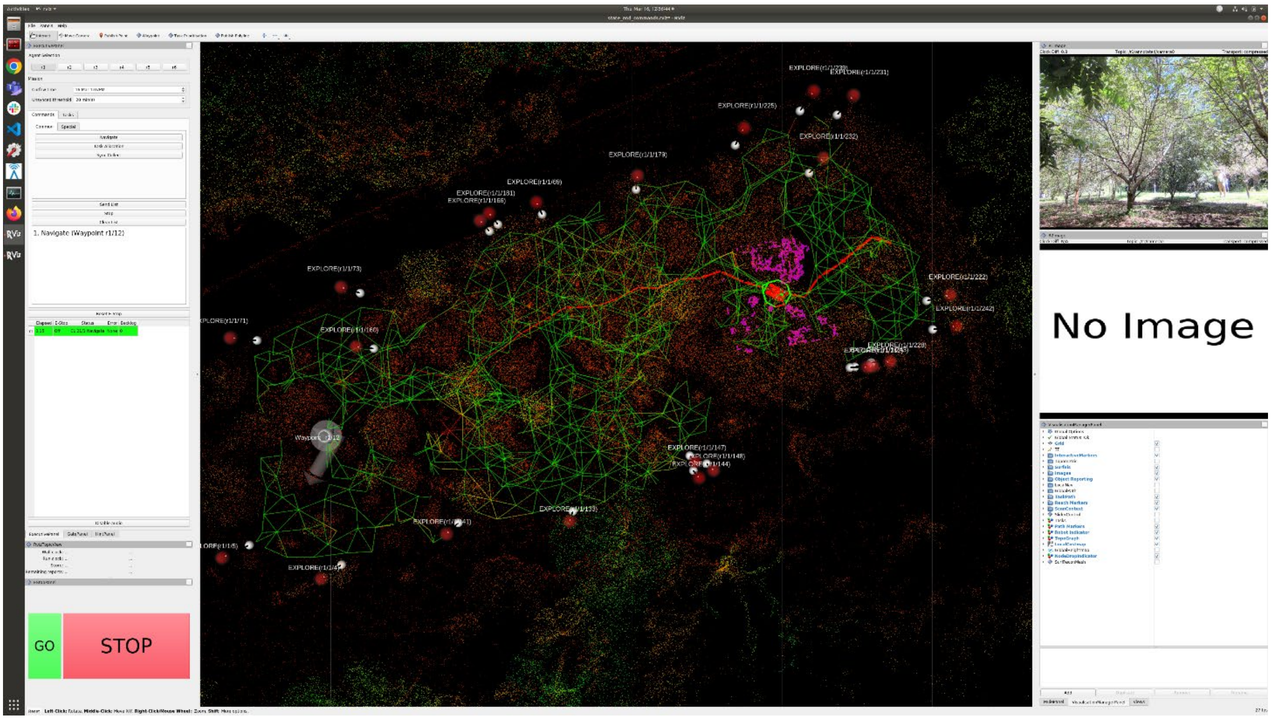


Figure 20 - Basestation showing autonomous return journey (red path) from frontline (right) back to human-safe location (grey disk).

The right panel shows the front view of the onboard cameras. Side camera views are also available if required.

The navigation mesh is rendered in green and shows the traversable areas as a triangle mesh. This mesh indicates the difficulty of the task, with some narrow routes between the many mapped trees (dark red points). This full map is only available on the return journey, as it is being generated as it navigates on the outward journey. If the terrain had been previously mapped (such as on a previous day) then this map can be made available for outward navigation using the new capabilities described in Section 3.3.

The following image is a 3D plot of the generated and coloured point cloud map, sliced at approximately a metre above the ground to cut out obscuring trees, and with the full trajectory shown from blue to red:

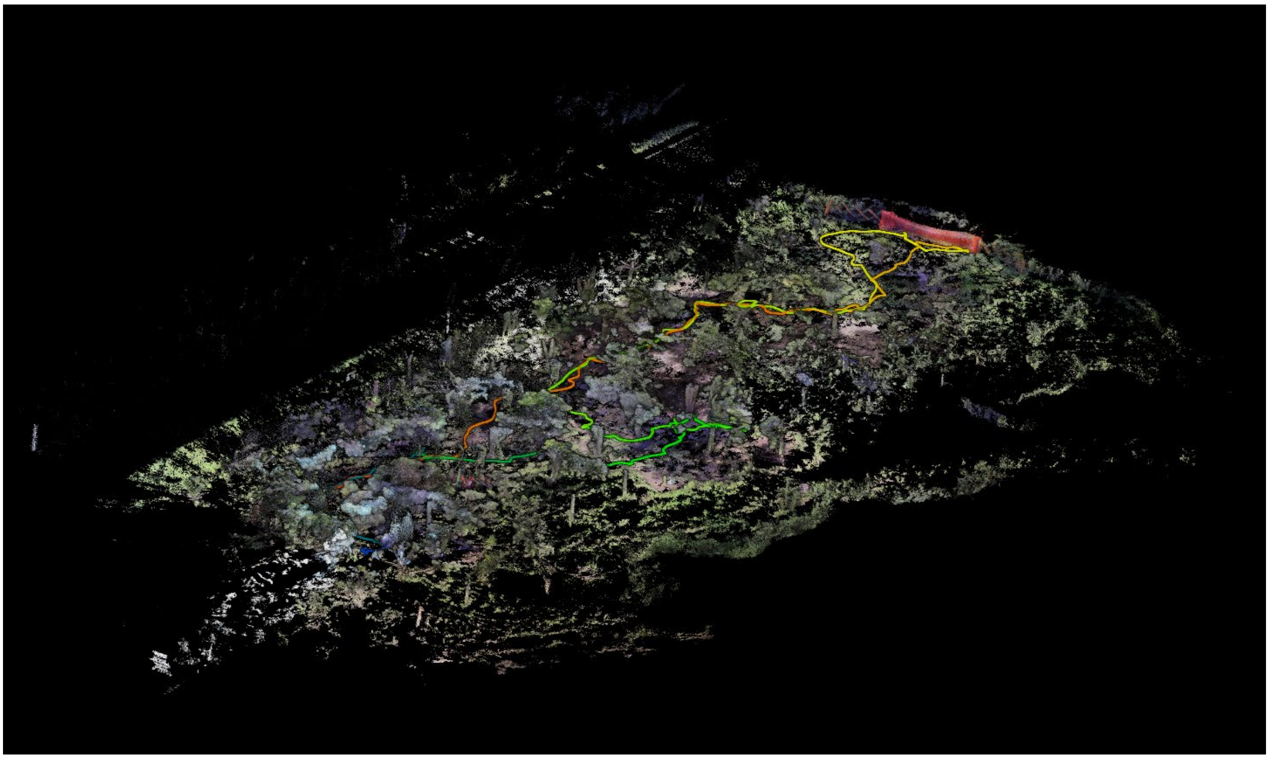


Figure 21 - Isometric view of the full path traversed by the UGV from blue to red. Start and end point: bottom left, fire front: top right.

The green part of the trajectory shows the outward route where it reached a point where it couldn't find a viable route out and so required a short period of teleoperation to get out of that tight location. This outward journey is more of an exploratory path as the map is being generated on the fly. The return journey (orange-red) is more direct due to being able to plan using the full map.

In order to achieve this autonomous navigation, we were required to remove some of the taller clumps of grass in the environment, which are currently treated as rigid obstacles. This report’s field test therefore confirms the necessity of our current research into classifying non-rigid obstacles in order to be able to drive through taller grass, weeds, bracken and similar undergrowth.

Overall, we were satisfied with the results of this field test, for this month 18 deliverable.

5 Integration methodology with SILVANUS platform

The UGVs have onboard computing for mapping, localisation, control and real-time navigation. Information that is useful to the operator can be sent back to the base station using the ROS message interface. Existing data that is passed to the base station includes a map of the terrain, the estimated traversibility cost, path plans and the robot's current location.

It is expected that the base station is configured to have a network connection to a server that hosts the SILVANUS platform, whether that be a full internet connection or a local private Wi-Fi connection. The most useful data for analysis of the fire situation can therefore be passed from the base station to the SILVANUS platform. This is performed by converting the relevant messages to the web REST interface, where they are pushed and transferred through HTTP, which uses the standard TCP/IP protocol.

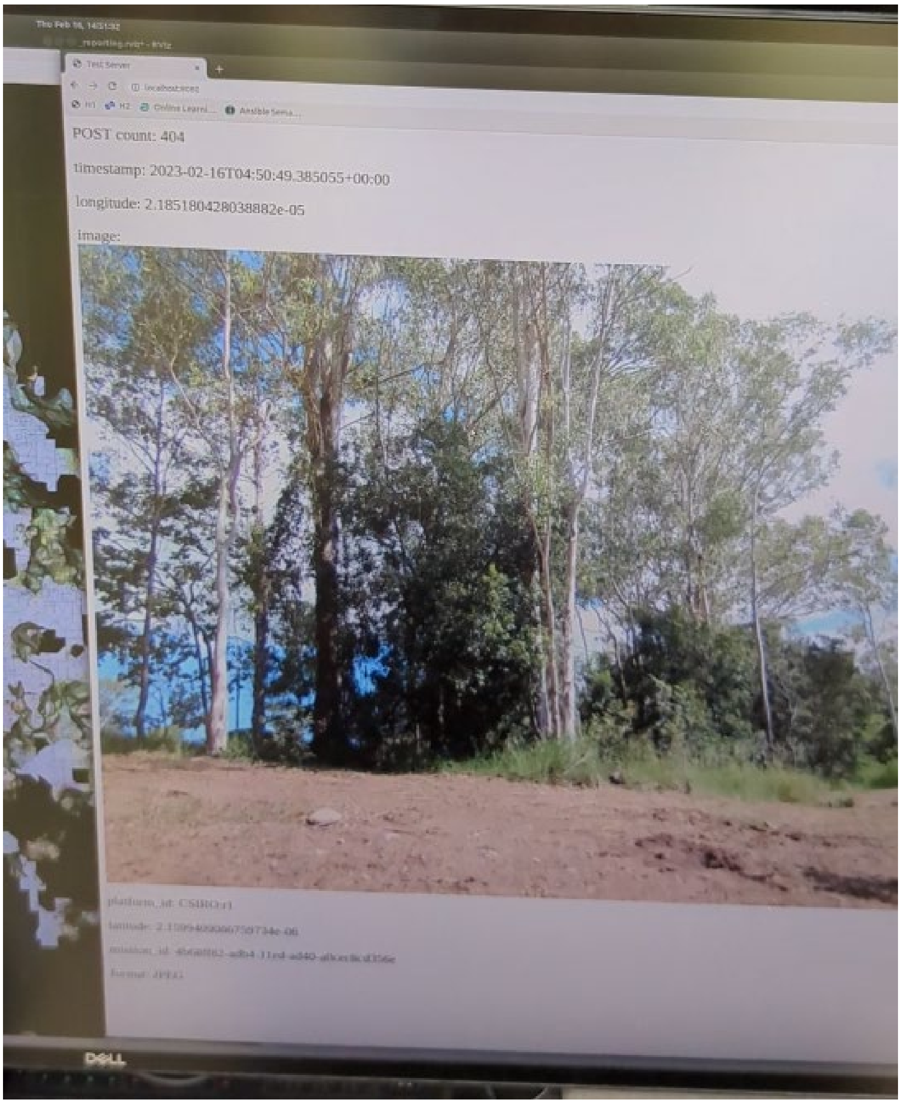
Specifically, a bridge node (silvanus_bridge) has been written, which is a python node designed to listen for the ROS messages arriving at the base station and forward them over the REST interface. The current set of messages are:

- The mission ID
- The platform ID
- The time stamp
- The estimated longitude and latitude of the UGV
- The image data and its image format

The REST post message is then of the form:

```
data = {
  "mission_id": mission_id,
  "platform_id": platform_id,
  "timestamp": timestamp,
  "latitude": latitude,
  "longitude": longitude,
  "format": "JPEG",
}
files = {
  "image": ("sample.jpeg", msg.jpeg_data, "image/jpeg"),
}
```

The UGV is designed to operate in GPS denied environments, so the conversion of its estimated location into longitude/latitude assumes that the start location of the robot has been geolocated. This data is passed in as launch parameters: gate_longitude, gate_latitude and gate_bearing. With this input data, the conversion is performed using Python's GeoPy library.



POST count: 404

timestamp: 2023-02-16T04:50:49.385055+00:00

longitude: 2.185180428038882e-05

image:

platform_id: CSIRO-r1

latitude: 2.1599400006759734e-06

mission_id: 4b68f862-adb4-11ed-ad40-a0cec8cd356e

format: JPEG

Figure 22 - Photo of base station, with browser running example server. It directly displays the data arriving through the REST interface, updating every 10 seconds

6 Conclusions

The deliverable reports upon the activities carried out in SILVANUS related to the overall functional capabilities of the ground vehicle. The key aspects of the robot include the autonomous navigational ability to explore the forest terrain, the ability to capture and transmit images collected in the field back to the SILVANUS cloud and the ability to represent the environment in a 3D map. Additionally, the use of two commercially available robots have also been identified to be demonstrated within the SILVANUS pilot activities. The operational effectiveness of the robots will be subsequently evaluated within the context of WP9 (piloting of Phase A, B and C) activities.

7 References

[1] N. Hudson, F. Talbot, M. Cox, J. Williams, T. Hines, A. Pitt, B. Wood, D. Frousheger, K. Lo Surdo, T. Molnar, R. Steindl, M. Wildie, I. Sa, N. Kottege, K. Stepanas, E. Hernandez, G. Catt, W. Docherty, B. Tidd, B. Tam, S. Murrell, M. Bessell, L. Hanson, L. Tychsen-Smith, H. Suzuki, L. Overs, F. Kendoul, G. Wagner, D. Palmer, P. Milani, M. O'Brien, S. Jiang, S. Chen and R. Arkin, "Heterogeneous Ground and Air Platforms, Homogeneous Sensing: Team CSIRO Data61's Approach to the DARPA Subterranean Challenge," Field Robotics, vol. 2, pp. 595-636, 2022.

[2] M. Ramezani, K. Khosoussi, G. Catt, P. Moghadam, J. Williams, P. Borges, F. Pauling and N. Kottege, "Wildcat: Online continuous-time 3D lidar-inertial SLAM," arXiv preprint arXiv:2205.12595, 2022.

[3] T. Hines, K. Stepanas, F. Talbot, I. Sa, J. Lewis, E. Hernandez, N. Kottege and N. Hudson, "Virtual surfaces and attitude aware planning and behaviours for negative obstacle navigation," IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 4048-4055, 2021.

[4] J. Knights, K. Vidanapathirana, M. Ramezani, S. Sridharan, C. Fookes and P. Moghadam, "Wild-Places: A Large-Scale Dataset for Lidar Place Recognition in Unstructured Natural Environments," arXiv preprint arXiv:2211.12732, 2022.

[5] T. D. Lowe and K. Stepanas, "RayCloudTools: A Concise Interface for Analysis and Manipulation of Ray Clouds," IEEE Access, vol. 9, pp. 79712-79724, 2021.

[6] T. Lowe, P. Moghadam, E. Edwards and J. Williams, "Canopy density estimation in perennial horticulture crops using 3D spinning lidar SLAM," Journal of Field Robotics, vol. 38, no. 4, pp. 598-618, 2021.