Post, Mark A. and Bianco, Alessandro and Yan, Xiu T. (2017) Autonomous navigation with ROS for a mobile robot in agricultural fields. In: 14th International Conference on Informatics in Control, Automation and Robotics (ICINCO), 2017-07-26 - 2017-07-28, Universidad Rey Juan Carlos. ,

Autonomous Navigation with ROS for a Mobile Robot in Agricultural Fields

Mark A. Post, Alessandro Bianco and Xiu T. Yan
University of Strathclyde, 75 Montrose St., Glasgow, U.K.
{*mark.post, alessandro.bianco, x.yan*}*@strath.ac.uk*

Keywords: Field Robotics, Navigation, Agriculture, ROS.

Abstract: Autonomous monitoring of agricultural farms and fields has recently become feasible due to continuing advances in robotics technology, but many notable challenges remain. In this paper, we describe the state of ongoing work to create a fully autonomous ground rover platform for monitoring and intervention tasks on modern farms that is built using inexpensive and off the shelf hardware and Robot Operating System (ROS) software so as to be affordable to farmers. The hardware and software architectures used in this rover are described along with challenges and solutions in odometry and localization, object recognition and mapping, and path planning algorithms under the constraints of the current hardware. Results obtained from laboratory and field testing show both the key challenges to be overcome, and the current successes in applying a low-cost rover platform to the task of autonomously navigating the outdoor farming environment.

1 INTRODUCTION

Interest in the application of robotic automation to agriculture has grown quickly in recent years, due to robotic technologies having reached a degree of maturity that allows autonomous experimentation in agricultural fields. Mobile robotic platforms have traditionally been very expensive in term of production costs with only the ability to perform simple tasks such as moving and recording images, and have been focused on data gathering and repetitive work in applications such as space exploration or factory mass production. Agricultural tasks are more challenging: in a non-structured environment, gathering information from the environment, harvesting randomly distributed crops, and simply navigating continuously for a long time away from a power source require a higher level of autonomous intelligence. It is only recently that improvements in sensor and actuator technologies and progress in data processing speed and energy have enabled robots to process enough information to reliably and affordably perform basic navigational tasks autonomously in an agricultural environment. The development and testing of navigation methods suitable for an autonomous agricultural data gathering robot is the focus of this paper.

In (Roßmann et al., 2009), (Krahwinkler et al., 2011) and (Roßmann et al., 2010), the authors propose a robotic platform which is able to move through a large forested area and produce a map of the trees. Their robot senses the environment by mean of GPS, stereo camera, and range scanner information. Since the GPS could not be always reliable under the canopy of trees, it is used as a first position estimation for a Kalman Filter. Aerial and satellite information are integrated in order to extract a better representation of the environment. Finally a virtual forest is extracted as a map of the area, and trees are classified according to their image.

In (Katupitiya et al., 2007), the authors propose a robotic platform for precision seeding. Their robotic architecture uses two high precision differential GPS units and an IMU to determine the exact location for the placement of the seeds. The path tracking algorithm monitors the movement with high precision in order to reduce the error of seed placement.

In (Henten et al., 2003b), (Henten et al., 2003a), (van Henten et al., 2002) the authors propose a robotic cucumber harvester. The platform is a robotic arm that moves along rails mounted in a greenhouse. The robot uses stereo vision information to detect the cucumbers hanging from the vines, determine whether they are ready for harvest and create a 3D representation of the environment. The 3D representation is used to allow the arm to plan a harvesting task. Another strawberry harvesting system have been developed in (Hayashi et al., 2010) and (Hayashi et al., 2012). These harvesting systems require that the arm

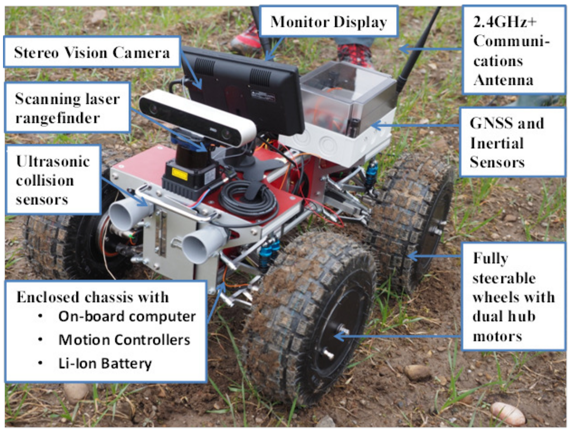

Figure 1: Rover Platform.

moves along predefined rails and resulting in a long harvesting time and still exhibit a high failure rate.

The purpose of our project is to combine several approaches to navigation and control into one autonomous rover, which shall be able to navigate autonomously in a farm field while constructing a map of the environment. The rover's main use is automated visual inspection and soil nitrogen level data gathering, but as an autonomously navigating platform it is designed with the capacity for other tasks such as harvesting and crop treatment. A soil testing payload and a controllable arm payload are currently being developed for use on this rover.

Our ongoing aim to create a low-cost, fully autonomous mobile robotic platform, which can provide useful services to farmers and make full use of modern robotic technology. In this paper, we describe the current state in development and technology of this practical rover that has the ability to localize itself through odometry and the ability to navigate reliably to locations in a farm field while using off the shelf parts in its construction without the need for high-priced inertial measurement, processing, and sensing hardware.

2 SYSTEM ARCHITECTURE

An image of the rover used in this work is shown in 1. The chassis is articulated to allow the front and back halves of the rover to rotate and compensate for balance in the advent of uneven rough terrain. Two wheels are attached to the front and back of the chassis, each with an integrated DC brush gearhead motor and controller within the wheel hub and with angular steering by means of small linear steering actuators. The space within the front half of the rover stores a 22 amp-hour lithium-polymer battery pack for power, and the space within the back half contains an NVidia Jetson TK1 single-board computer (SBC), an SPI-connected Arduino board that handles fast kinematic control of the drive and steering motors, and a SparkFun Razor attitude and heading reference system (AHRS) board. Four sensors allow the perception of the environment: the Razor AHRS board containing a three degree of freedom accelerometer, angular rate sensor, and magnetometer, a Hokuyo UTM-30LX-EW scanning laser rangefinder mounted on the front of the chassis, a Stereolabs ZED stereo vision camera mounted on top of the laser and a global navigation satellite system (GNSS) receiver mounted on top of the camera.

As the Jetson TK1 and ZED stereo camera are relatively inexpensive, the sensor and processing hardware has a total price of a few hundred Euros, only the laser can be considered as a fairly high-priced component in this system, and with the current interest in development of low-cost LIDAR technology it is expected that the cost of this component will decrease in the near future. Differential GNSS is not yet used due to the lack of internet access to reference stations in most farming environments. Two ultrasonic range sensors on the front of the rover provide additional capability to detect potential collisions. The sensors are small and leave the entire back half area on top of the rover for payloads, and the total weight of the rover platform is 15.5kg. Payloads currently being built for the rover include a laser-induced breakdown spectrometer (LIBS) system for measuring chemical components of soil such as Nitrogen, and a lightweight mechanical arm with three degrees of freedom for camera-based crop inspection and sample cutting.

Figure 2 contains a representation of the hardware components and their connections. Inertial sensors are connected directly via TTL serial to the Jetson TK1 SBC. The Hokuyo scanning laser rangefinder is connected via Ethernet, the ZED stereo camera by USB3, and the GNSS unit by USB2. All sensor processing is done on the Jetson TK1 SBC. Directional movement commands (speed and steering rate) are generated using an on-board planner and sent at a rate of 100Hz to the Arduino board for kinematic transformation. The Arduino board in turn calculates the required steering angles and motor speeds to execute the desired movement and transmits them via TTL serial and PWM to the motor controllers and steering actuators. Communications and telemetry from the rover is provided using an Intel Mini-PCI Wi-Fi card.

A software system allows the rover to autonomously navigate and perform tasks in the environment. The NVidia Jetson TK1 board runs Robot Operating System middleware (ROS Indigo) on an

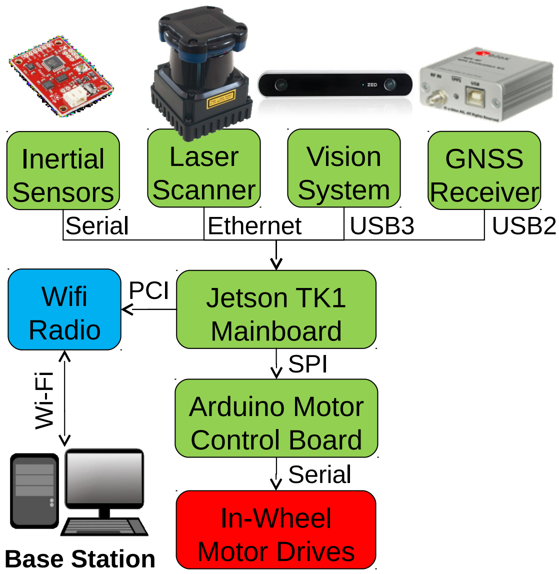

Figure 2: Hardware Architecture.

Ubuntu Linux 14.04 operating system environment, allowing convenient and free access to a large array of open-source software modules developed by the ROS community (with names noted when appropriate here), including many that implement results from other research projects. ROS nodes may be connected to sensors or actuators, or they may be intermediate nodes that process data. Sensor-connected nodes broadcast sensor information via topics in the ROS system, so that other nodes may use them as inputs for computation, actuator-connected nodes read information from ROS topics and output commands to their controllers, and other nodes simply process information from topics while broadcasting new topics. Our system contains four ROS sensor nodes: an IMU node (razor_imu_9dof), a ZED stereo camera node (zed-ros-wrapper), a laser scanner node (hokuyo_node) and a GPS node (nmea_navsat_driver).

Several intermediate nodes that are available as part of the ROS community perform various navigation processing tasks as follows:

- A transform broadcaster publishes information regarding the relative movement of the rover parts.
- A GPS transform node (navsat_transform_node) converts the GPS coordinates to approximate 2-D orthogonal coordinates in a fixed Cartesian Coordinate System parallel to the rover plane of operation.
- A Hector-SLAM node (Kohlbrecher et al., 2011) converts laser readings into odometry estimation according to the displacement of laser-detected objects.
- A visual odometry node converts information from the stereo camera into estimation of robot movements. and extended Kalman Filter node (Moore and Stouch, 2014) fuses all the odometry information and the velocity and orientation information from the IMU to produce one final global odometry estimate.
- An RTAB-MAP node (Labbé and Michaud, 2014) uses the final odometry estimate and the camera data to reconstruct a 3-D and 2-D representation of the environment
- A map-server node manages complex maps by decomposing them into multiple smaller parts and determines when to use the various map parts.

Several new ROS nodes were written specifically for the agricultural rover platform to facilitate outdoor agricultural navigation tasks in the context of farm monitoring as follows:

- A custom-designed controller node sets wheel speed for all four wheels, controls the wheel angles through the linear actuators for steering, and feeds back wheel and steering odometry via the SPI interface to the Arduino board.
- As most existing path planners are precise in nature and do not perform well in uncertain or changing environments, a path planner node was created with a simplified algorithm for choosing uncomplicated paths efficiently through open spaces such as farm fields.
- A global planner node takes goal information and produces a plan though the 2-D map that allows the rover to reach its destination, a local planner takes a global plan and sends command to the wheel controller, moreover it stops the rover in the advent of unmapped obstacles such as passing animals, a navigator node store the set of goals that a user may input into the rover and oversees the execution of the plan.
- Finally a connector node connects ROS to an external user interface, such interface is the point of access of a farmer and it offers the ability to look at the re-constructed map and input high level goals to the rover, such as sampling at multiple destinations.

These new ROS nodes perform the key functions that are the focus of the remainder of the paper. Figure 3 shows an overview of the architecture of the software system.

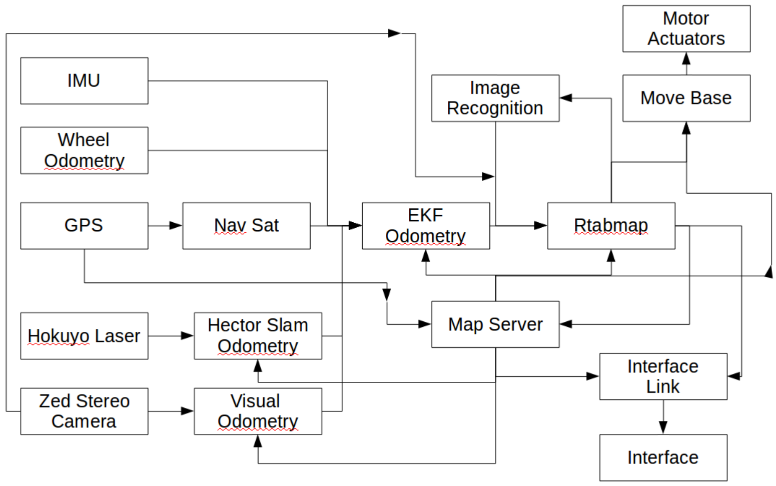

Figure 3: Software Architecture.

3 SENSOR FUSION FOR ODOMETRY

Due to the focus on reducing costs of the rover platform and sensors, rover perception is short in range and low in precision, and due to the focus on outdoor mobility, wheel odometry by itself is too unreliable for mapping purposes and GNSS localization alone is considered to be too inaccurate for short (less than 3m) distances. To obtain accurate movement odometry, a combination of visual odometry using RTAB-Map, laser-based odometry using Hector-SLAM, and wheel odometry is used. This information is fused using the Extended Kalman Filter and robot localization model available in the ROS robot_localization package (ekf_localization_node).

The greatest challenge for robotic navigation in farming fields is the uniformity of terrain and external features. Each part of a field or orchard looks very much the same visually or in terms of solid obstacles, and even when navigating down crop rows, visual measurement of distances is a challenge. The most precise sensor on the rover is a Hokuyo UTM-30LX-EW laser scanner, which performs very well in an indoor environment with walls and well-defined obstacles, but cannot provide consistent tracking in a sparse, unchanging or empty environment. Initially, visual odometry and mapping using the ZED stereo camera and RTAB-Map was evaluated as a single sensory method, producing a good-quality 3D representation of our lab and some outdoor areas in which features are rich in complex, recognizable objects such as the grove of trees shown in Figure 4. However, experimenting with RTAB-MAP software under different system configurations made it became clear that small errors in the odometry estimation caused the 3-D objects to shift in space and produce very bad reconstructions, and in outdoor environments it is generally not possible to reproduce good quality 3-D maps like these in real time.

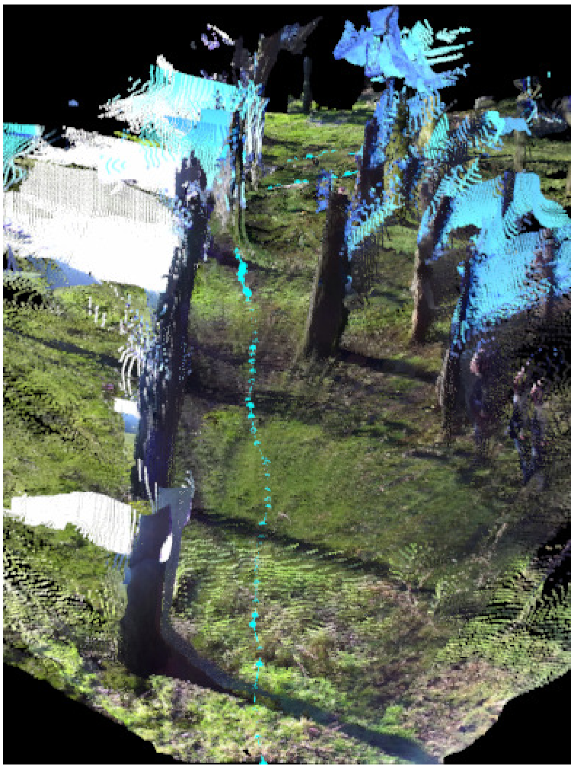

Figure 4: High-quality 3-D reconstruction of a grove of trees on a farm from RTAB-Map. Visual odometry is consistent only while differences between locations in the scene are obvious and discernible.

RTAB-Map has not been extensively tested in outdoor environments, and the use of appearance-based mapping requires large numbers of distinct features for accurate odometry and loop closure. RTAB-Map does not provide any correction on the odometry estimation between two successive frames, only large loop closures are subject to optimization once a map has been created. In the agricultural context the greatest challenge is traversal across farmers' fields, where very few nearby visual references are available for odometry and loop closure. Figure 5 shows a navigational map obtained from a farm field with a fenced boundary, in which good visual odometry is maintained as long as distinct objects around the boundary are present. This represents a relatively easy map to obtain at a speed of approximately 0.5 m/s due to the diversity of features available and navigation is straightforward as long as the boundary is in view of the rover. A more difficult situation is shown in 6, where the farm field boundary is grassy and lacking in distinct features. A slower forward speed of 0.2 m/s or lower is necessary to perform mapping while travelling along the boundary of this field, and visual odometry is less consistent. It is still possible to maintain good odometry when travelling along the perimeter of this field as long as some features are tracked well. However, as soon as the boundary is out of the rover's view when entering the centre of the field, odometry is easily lost and the quality of the map is compromised.

To evaluate the quality of odometry in an outdoor environment, a test was run outdoors in which five points were chosen in a 100 squared meter area and their relative position from a starting point was measured. The rover was manually driven through the five points, and the odometry estimation was recorded. Table 1 reports the average error between real position and recorded odometry under different configurations of the extended Kalman Filter node. The left column highlights the sensor information that has been fused in the EKF node, and the right column contains the average error in meters. The smallest error recorded in this test was 1.77 meters.

Our experiment confirmed that while the laser scanner was the most reliable sensor indoors, provides little reliable information in a large outdoor area. The irregular shapes of the plants does not only offer a challenge to Hector-Slam node in the reconstruction of a 2-D laser map, but the uneven terrain adds noisy shifts in the laser readings. As 3D reconstruction attempts proved that the error was too large to produce a full 3-D map of the area under test, emphasis in the mapping process is now being shifted to producing a reliable 2-D obstacle map that can be overlaid on a topographic elevation map.

4 LOCAL PLANNER

In our original design we planned to use the global planners and local planners available for the move_base node of ROS, which provides a general infrastructure to integrate different pairs of global and local planners. The node requires a 2-D occupancy map and a set of goal locations to compute a global plan and output velocity commands in real time to the drive motors until the destination is reached. Initially, the basic Navfn global planner in ROS was tested on the rover since it provides valid planning around mapped obstacles. However, it was found that nearly all ROS-based planners made critical assumptions about the underlying mobility hardware, in particular that commands and odometry feedback would be executed immediately with negligible time lag and that small incremental movements would be possible. As the rover platform uses four-wheel steering and takes time to accelerate and turn, the planning algorithms would repeatedly issue commands for small, incremental angular adjustments (or in the case of the ROS teb_local_planner, back-and-forth steering movements) and when this failed due to either actuator lag or lack of precise movement, execute recovery behaviours. To remedy this problem, we developed a new and simplified planner that relaxed the constraints on timing and precision in favour of a best-effort approach.

For trial testing, we created a very simple algorithm: at the start the rover rotates on the spot and aligns itself along the direction of the global path, then it tries to follow the path by steering when needed either when the direction of the curves changes too much or the rover moves too far away from the plan. Once the destination is reached a signal is sent to the navigator nodes which may provide another goal or halt the rover. The planner uses five states of operation with appropriate commands sent to the drive motors for the duration of each state: Start, Forward, Backward, Pause, and Rotation. A flowchart of the process the planner uses when moving from the current movement state to the next movement state is given in Figure 7.

Making use of the high-quality odometry information available by using Hector SLAM and the scanning laser rangefinder in the lab environment, this planner showed much better reliability on the rover platform. In the trial for the planner, seven different points were set spaced far apart on the lab floor and the rover was directed to visit them autonomously in order. Table 2 reports the distance between the set goals and the point the rover stopped at, and an average error of 15cm was recorded at the points the rover stopped. Figure 8 shows the 2-D Hector SLAM and

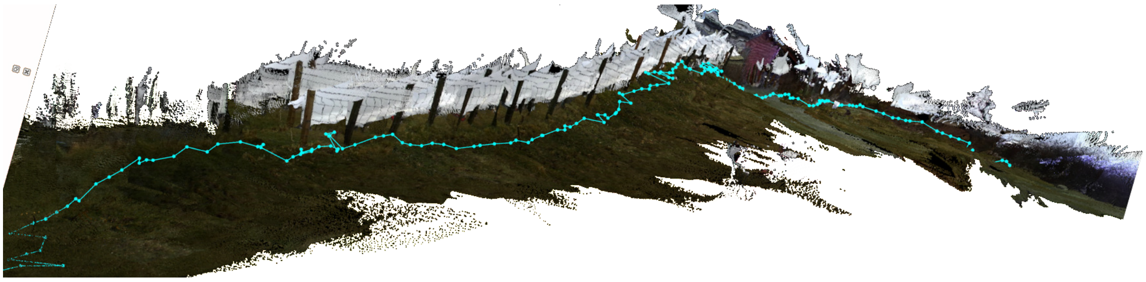

Figure 5: High-quality 3-D reconstruction of a farm field with fenced border from RTAB-Map. As long as the border is in view, visual odometry can be maintained.

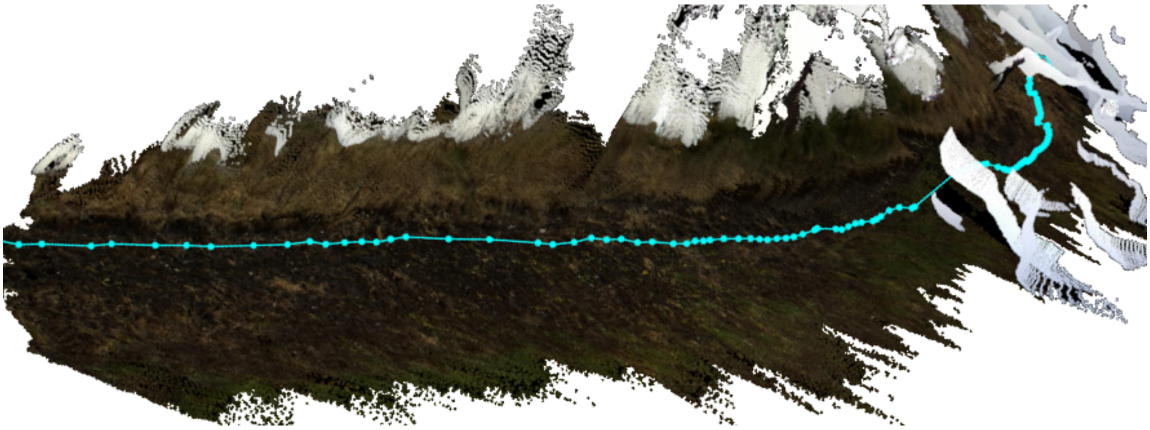

Figure 6: High-quality 3-D reconstruction of a farm field with grassy border from RTAB-Map. The border of the field provides barely enough discernible references to maintain visual odometry.

Table 1: The average fused odometry error in localizing at five points in a 100 squared meter outdoor area with respect to the different sensors fused by the EKF node.

Laser Odometry	Visual Odometry Linear Velocity	Visual Odometry Angular Velocity	IMU Orientation	IMU Angular Velocity	IMU Linear Acceleration	GPS	Wheel Odometry Linear Velocity	Wheel Odometry Angular Velocity	Error in Meters
absolute									6.81
	absolute	absolute							2.22
absolute		absolute							4.01
absolute	absolute								4.31
absolute	absolute	absolute							2.74
absolute				absolute					5.13
absolute					absolute				2.95
absolute			differential						7.14
absolute			differential	absolute	absolute				6.03
absolute								absolute	7.92
absolute							absolute		5.25
absolute							absolute	absolute	5.96
			absolute			absolute			4.61
absolute						differential			7.99
absolute			differential			differential			4.26
absolute	absolute	absolute			absolute				8.10
absolute	absolute	absolute	differential	absolute	absolute	differential			4.47
absolute	absolute	absolute	differential	absolute	absolute	differential	absolute	absolute	4.99
differential	absolute	absolute	absolute	absolute	absolute	absolute			1.77
differential	absolute	absolute	absolute			absolute			1.23
differential	absolute	absolute	absolute	absolute	absolute	absolute	absolute	absolute	3.87

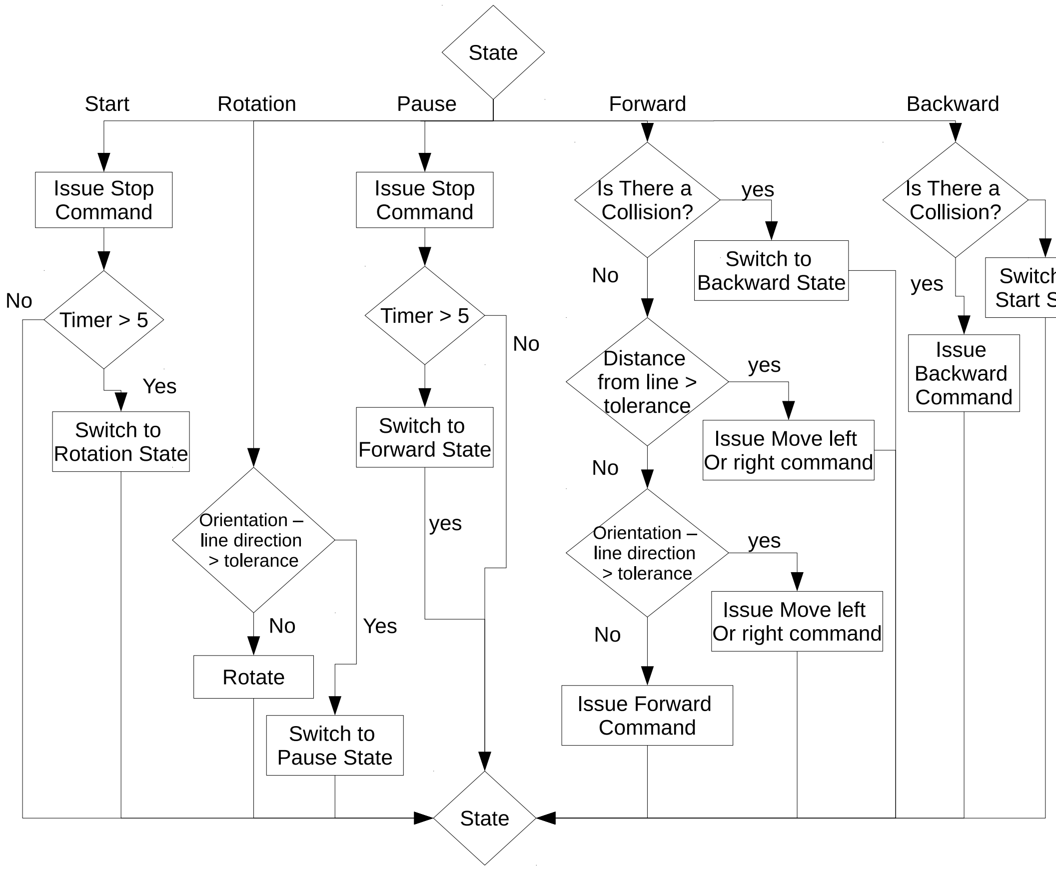

Figure 7: Flowchart of the navigational planner process. The planner sends commands to the drive motors while in each of five states, and state transitions are performed where indicated.

reconstructed 3-D RTAB-Map maps of our lab and the testing path: S marks the starting point and each numbered goal point is marked G. The goals are structured in order of the difficulty of reachability:

1. from S to G1 there is a straight path
2. from G1 to G2 a turn on the spot is involved
3. from G3 to G4 there is a sharp corner
4. from G5 to G6 there is a narrow path
5. from G6 to G7 the path includes a narrow door

The same experiment was executed in an outdoor area. Unfortunately, due to the low mapping quality and the higher position uncertainty of the rover in outdoor environment, the rover could reach only the first goal with an error of 1-2 meters and was not able to reach the following ones. Table 3 reports the distance between the set goal and the point the rover stopped at. During these test the GPS system was not used, since it was missing at the time and the experiment was not repeated.

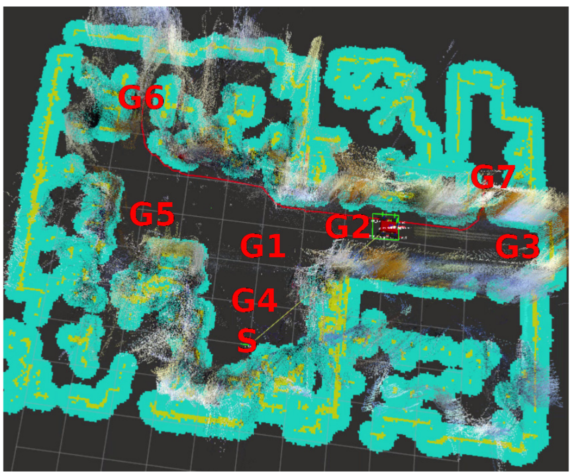

Figure 8: Reconstructed map of our lab and testing paths. The picture was taken when the rover was attempting the navigation from G6 to G7, but G7 was never reached as the rover could not cross the door.

5 CONCLUSION

In this paper we introduced a practical and cost-

Table 2: The navigation error in the lab environment for each one of the six goals, depending on the enabled sensors (NR stands for Not Reached).

Laser Odometry	Other Sensors	Measure	Goal 1 Error (cm)	Goal 2 Error (cm)	Goal 3 Error (cm)	Goal 4 Error (cm)	Goal 5 Error (cm)	Goal 6 Error (cm)	Goal 7 Error (cm)
absolute			0	20	20	NR	NR	NR	NR
absolute	IMU	Angular Velocity	0	5	15	0	0	0	NR
absolute	IMU	Linear Acceleration	25	10	35	NR	NR	NR	NR
absolute	IMU	Both above measures	0	20	35	0	20	NR	NR
absolute	Visual Odom.	Linear Velocity	20	30	5	5	NR	NR	NR
absolute	Visual Odom.	Angular Velocity	0	10	25	0	0	20	NR
absolute	Visual Odom.	All Velocities	0	50	20	5	40	20	NR
absolute	Wheel Odom.	Linear Velocity	70	30	30	0	NR	NR	NR
absolute	Wheel Odom.	Angular Velocity	10	60	25	150	25	0	NR
absolute	Wheel Odom.	All velocities	5	NR	NR	NR	NR	NR	NR

Table 3: The navigation error in garden environment for one goal located 15 meters ahead of the starting position, depending on the enabled sensors. NR stands for Not Reached.

Laser Odometry	Other Sensors	Measure	Goal Error in meters
absolute			1.5
absolute	IMU	Angular Velocity	2
absolute	IMU	Linear Acceleration	3
absolute	Visual Odom.	Linear Velocity	0.5
absolute	Visual Odom.	Angular Velocity	NR
absolute	Wheel Odom.	Linear Velocity	1
absolute	Wheel Odom.	Angular Velocity	1.5

-effective robotic rover platform for autonomous environment monitoring in agriculture, and discussed several key technologies implemented in ROS for odometry and localization, identification of objects, and, path planning under limited kinematic precision. Our results to date regarding navigation indicate that the rover can successfully reach set navigational points with high precision so long as accurate and real-time localization is available such as that provided by Hector SLAM against fixed obstacles. We have also established that the methods used for indoor and feature-rich localization and odometry are not suitable for use outdoors in uniformly-visual farm fields. We are nonetheless able using a low-cost rover platform with a minimal sensor set to traverse navigational goals efficiently and quickly using a simple, efficient, and kinematically-tolerant planning algorithm for our rover platform.

The next steps in our ongoing work to develop this platform include the integration of differential GNSS localization between a local base station and the rover using the recently-released u-blox C94M8P devices as a supplement for odometry in feature-poor field environments, and the integration of soil sampling and visual measurement technologies to allow autonomous monitoring activities to be tested in the field.

ACKNOWLEDGEMENTS

This work was made possible and supported by grants from the Science and Technology Facilities Council Newton Fund. The authors gratefully acknowledge the work of the Rutherford Appleton Laboratories (RAL) Autonomous Systems Group for the design and construction of the mechanical platform for the rover, the James Hutton Institute for providing field test facilities in support of this research, and the work of Jonathan Watson, Giacomo Corvi, Kyle Burnett, Jennifer Miller, and Finlay Harris on setting up and testing RTAB-Map algorithms in ROS.

REFERENCES

Hayashi, S., Shigematsu, K., Yamamoto, S., Kobayashi, K., Kohno, Y., Kamata, J., and Kurita, M. (2010). Evaluation of a strawberry-harvesting robot in a field test. *Biosyst. Eng.*, 105:160–171.

Hayashi, S., Yamamoto, S., Saito, S., Ochiai, Y., Kohno, Y., Yamamoto, K., Kamata, J., and Kurita, M. (2012). Development of a movable strawberry-harvesting robot using a travelling platform. In *Proc. Int. Conf. Agric. Eng. CIGR-AgEng 2012, Valencia, Spain*.

Henten, E. V., Tuijl, B. V., Hemming, J., Kornet, J., and Bontsema, J. (2003a). Collision-free motion planning for a cucumber picking robot. *Biosyst. Eng.*, 86:135–144.

Henten, E. V., Tuijl, B. V., Hemming, J., Kornet, J., Bontsema, J., and van Os, E. A. (2003b). Field test of an autonomous cucumber picking robot. *Biosyst. Eng.*, 86:305–313.

Katupitiya, J., Eaton, R., and Yaqub, T. (2007). Systems engineering approach to agricultural automation: New developments. In *1st Annual IEEE Syst. Conf.*, pages 298–304.

Kohlbrecher, S., Meyer, J., von Stryk, O., and Klingauf, U. (2011). A flexible and scalable slam system with full 3d motion estimation. In *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE.

Krahwinkler, P., Roßmann, J., and Sondermann, B. (2011). Support vector machine based decision tree for very high resolution multispectral forest mapping. In *2011 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2011, Vancouver, BC, Canada, July 24-29, 2011*, pages 43–46.

Labbé, M. and Michaud, F. (2014). Online global loop closure detection for large-scale multi-session graph-based SLAM. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014*, pages 2661–2666.

Moore, T. and Stouch, D. (2014). A generalized extended kalman filter implementation for the robot operating system. In *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer.

Roßmann, J., Jung, T. J., and Rast, M. (2010). Developing virtual testbeds for mobile robotic applications in the woods and on the moon. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 18-22, 2010, Taipei, Taiwan*, pages 4952–4957.

Roßmann, J., Schluse, M., Schlette, C., Bücken, A., Krahwinkler, P., and Emde, M. (2009). Realization of a highly accurate mobile robot system for multi purpose precision forestry applications. In *The 14th Int. Conf. Adv. Robotics*, pages 133–138.

van Henten, E. J., Hemming, J., van Tuijl, B. A. J., Kornet, J. G., Meuleman, J., Bontsema, J., and van Os, E. A. (2002). An autonomous robot for harvesting cucumbers in greenhouses. *Auton. Robots*, 13(3):241–258.