Abstract

The nexus of robotics and autonomous systems and artificial intelligence (AI) has the potential to change the nature of human guided exploration of indoor and outdoor spaces. Such autonomous mobile robots can be incorporated into a variety of applications, ranging from logistics and maintenance, to intelligence gathering, surveillance, and reconnaissance (ISR). One such example is that of a tele-operator using the robot to generate a map of the inside of a building while discovering and tagging the objects of interest. During this process, the tele-operator can also assign an area for the robot to navigate autonomously or return to a previously marked area/object of interest. Search and rescue and reconnaissance abilities could be immensely improved with such capabilities.

The goal of this research is to prototype and demonstrate the above autonomous capabilities in a mobile ground robot called Explorer51. Objectives include: (i) enabling an operator to drive the robot non-line of sight to explore a space by incorporating a first-person view (FPV) system to stream data from the robot to the base station; (ii) implementing automatic collision avoidance to prevent the operator from running the robot into obstacles; (iii) creating and saving 2D and 3D maps of the space in real time by using a 2D laser scanner, tracking, and depth/RGB cameras; (iv) locating and tagging objects of interest as waypoints within the map; (v) autonomously navigate within the map to reach a chosen waypoint.

To accomplish these goals, we are using the AION Robotics R1 Unmanned Ground Vehicle (UGV) rover as the platform for Explorer51 to demonstrate the autonomous features. The rover runs the Robot Operating System (ROS) onboard an NVIDIA Jetson board, connected to a Pixhawk controller. Sensors include a 2D scanning LiDAR, depth camera, tracking camera, and an IMU. Using existing ROS packages such as Cartographer and TEB planner, we plan to implement ROS nodes for accomplishing these tasks. We plan to extend the mapping ability of the rover using Visual Inertial Odometry (VIO) using the cameras. In addition, we will explore the implementation of additional features such as autonomous target identification, waypoint marking, collision avoidance, and iterative trajectory optimization. The project will culminate in a series of demonstrations to showcase the autonomous navigation, and tele-operation abilities of the robot. Success will be evaluated based on ease of use by the tele-operator, collision avoidance ability, autonomous waypoint navigation accuracy, and robust map creation at high driving speeds.

Introduction

Semi-autonomous and fully-autonomous vehicle design is becoming more common in military technology. Unmanned aerial vehicles (UAVs) as well as unmanned ground vehicles (UGVs) are poised to take over battlefields as the technology for these systems rapidly advances. Fully-autonomous vehicles perform tasks with minimal direct user input, essentially transferring the decision-making power from the human to the robot. The fact that systems designed for complete autonomy must be able to make a variety of decisions on their own makes them very complex to design. Meanwhile, semi-autonomous vehicles typically perform some, usually simple, tasks autonomously, but rely on the assistance of a remote operator to perform riskier tasks.

The benefits of these systems are numerous and aim to reduce operational risk in unfamiliar environments. This lack of risk along with the ability to pursue objectives relentlessly has changed the conduct of military operations [1]. For example, being in hostile environments without knowledge of the operational space presents a potentially dangerous position for humans

to be in. However, if an unmanned vehicle can provide reconnaissance data to the humans before they enter the environment, the situation becomes much less dangerous.

Previous University of Virginia capstone teams have developed a 3D-printed unmanned ground vehicle ("the rover") consisting of commercial off-the-shelf parts. This design made the rover relatively cheap to build as well as simple enough for nontrained personnel to maintain and operate. The ability to operate both indoors and outdoors while providing real-time data to the operator was considered essential for the rover. To achieve this, the rover was fitted with a LIDAR that could scan the space around the rover and send the data back to the operator via WiFi. Using this data, a map of the rover's local environment and the pose of the rover within the map is generated during operation with Google Cartographer's simultaneous localization and mapping (SLAM) algorithm. Lastly, in an attempt to avoid damaging the rover, obstacle avoidance was implemented that would stop the rover in the event of a potential front-end collision.

Objectives and Methods

Rover design goals included improving the ability and performance of a rover with the purpose of exploring uncharted areas which are too dangerous for humans. The test area for the rover was the Link Lab Arena at the University of Virginia.

The first step to prepare the rover for realistic scenarios is to allow an operator to comfortably drive the robot non-line-of-sight. The initial robot design had no way of doing this, and this goal is essential for non-human exploration. This will be evaluated by successfully navigating the robot through routes of tables and hallways which are out of view of the operator.

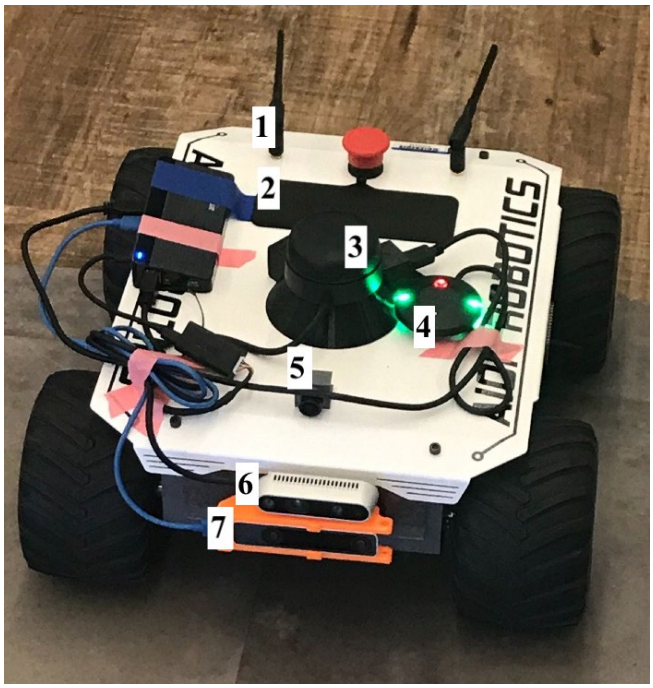
Improving the rover's mapping capabilities is the second goal. If the rover's mission is at all successful, the operating personnel will need an accurate scan of the area providing important intelligence. Additionally, it was determined in discussions with the MITRE team that the rover should be able to flag and save points of interest, or waypoints, within the map for later investigation or avoidance. To this end, the rover should have the ability to save a realistic map after exploring an indoor region, be able to localize itself within a map, and be able to save waypoints at input locations in the map.

For the next step in improving the rover's performance, it was determined that it was common for the operator to accidentally run the rover into walls or other obstacles while exploring in realistic indoor conditions. This could come from several factors, including a lagging connection, poor conditions, or excess speed. A safe collision avoidance feature that halts the robot's motion if a collision is imminent and enables quick and smooth recovery would provide for seamless exploration and lesser risk of damage to the rover.

If all of these goals are accomplished, there is still a disconnect between what the operator sees while exploring and what the rover sees and saves for later. A two-dimensional map provides some insight into the rover's surroundings, but a three-dimensional map would greatly improve this connection for improved human interpretation as well as more accurate visual inertial odometry, which is the robot's process of placing itself within its surroundings. The integration of three-dimensional mapping would accomplish this goal.

To further expand the rover's operational capabilities, it should be able to autonomously navigate toward previously placed waypoints on a map, or back to the initial position. This would involve both global and local path planning, pose and velocity control, and obstacle avoidance.

Current Design



1) 2.4 GHz External Antennae. 2) Battery compartment and interior access. 3) LiDAR. 4) HERE M8N GPS Module 5) First-person camera. 6) Intel RealSense D435 depth camera. 7) Intel RealSense T265 tracking camera.

Hardware

It was decided early on to transition from the original 3D-printed rover to the AION ROBOTICS R1 UGV. Both provide similar internal hardware, but the R1 has a larger, more rugged chassis with room to house additional hardware components. Both are controlled by a Pixhawk running ArduPilot software paired with an Nvidia Jetson TX2 microcontroller. The Jetson communicates with the Pixhawk, which is most directly responsible for handling input from a radio control (RC) transmitter and controlling the motors, via the MAVLink protocol. The Jetson also runs some form of the Ubuntu operating system, on top of which the Robot Operating System (ROS) is used to integrate with the rover's various sensors.

The onboard LiDAR is the RPLIDAR A2. Compared to the previous chassis, the LiDAR is positioned in a higher and more central location on the rover, allowing a larger range of LiDAR scans. Scans from the LiDAR are accessible to the Jetson through a ROS node.

Two Intel cameras, the RealSense D435 depth camera and the T265 tracking camera, were added in order to enhance the rover's three-dimensional mapping capabilities. The depth camera provides RGB and depth images of its surroundings, useful for creating a visual 3D map. The tracking camera includes two fisheye cameras, an additional inertial measurement unit (IMU), and on-board SLAM. All of these data are also accessed via ROS.

To allow more remote tele-operation of the rover outside of the operator's line of sight, an analog camera was mounted to the front of the chassis. This camera was connected to an additional radio, separate from the either of the Pixhawk or Jetson microcontrollers. Video feed from this radio could thus be streamed directly to analog monitors or headsets with lower latency than with Wi-Fi.

Software

The on-board Jetson computer ran Robot Operating System (ROS), a flexible framework for writing robotics programs that allows Ubuntu Linux to control the physical components of the rover. ROS was originally selected by a prior UVA capstone team because it can intuitively support a wide range of robotics algorithms. Like the team before us, the package Google Cartographer was used for simultaneous localization and mapping (SLAM) within a 2D space. Cartographer uses SLAM to transform the output data from the LiDAR sensor to generate and visualize an overhead map of the space in real time, displayed in Rviz. We continued the use of Google Cartographer for 2D SLAM and MAVLink override messages to implement obstacle avoidance.

While the ArduPilot software already implements simple object avoidance with LiDAR, it requires the LiDAR to be directly plugged into the Pixhawk, leaving it unavailable to be used for other purposes, such as 2D SLAM.

Moving to a 3D space, a Real-Time Appearance-Based Mapping (RTAB-Map) algorithm was used to create a 3D of the rover's surroundings. RTAB-Map uses an incremental approach to detect and close loops. The SLAM algorithm looks a thet new image and compares it to existing elements in the live map. Based on the appearance of the image in combination with its iodimetry, RTAB-Map is able to decide, detect and fix if a loop in the map exists.

Timed Elastic Band planner (TEB planner) is used for anonymous navigation and obstacle avoidance, it locally optimizes the rover's trajectory with respect to the trajectory execution time.

Results

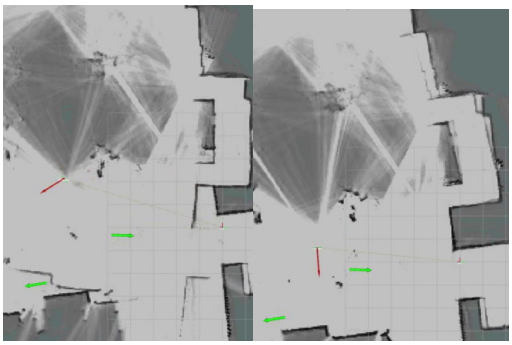
Obstacle Avoidance

The obstacle avoidance code from the previous project had two main drawbacks: Both forward and backward controls locked when an obstacle in front was detected, and did not stop the new rover from crashing. The control lock was relaxed by checking if new distance readings were the same as previous ones. The new rover had a faster top speed, enabling drivers to crash into obstacles faster than the avoidance node could detect them. Increasing the distances the node checks for obstacles fixed this problem for the newer rover.

The modifications made to the obstacle avoidance were not tested rigorously, but were able to consistently detect objects as thin as a table leg (2") and stop the rover's forward motion. The robot however did find it difficult to account for transparent objects in its path, such as glass windows.

Area Mapping

Previous work on the rover had solely used data from a 2D LiDAR to generate its maps, and faced issues with their map shifting after the rover turned. After some tweaking of Cartographer configuration files, short-term map quality improved. The 2D map has still been seen to drift after some more extended operation, but this appears to self-correct after some time.



An example of a drifting map (left), where the hallway entrance in the bottom right on a second visit does not match up with how it was originally mapped. After a delay, this error is fixed (right).

This implementation may work for the simple task of navigating around a space, but is incapable of determining which objects actually occupy the space. For this purpose, we chose to employ a D435 depth camera for its RGB and depth imaging capabilities, in order to create a 3D point cloud of its field of view.

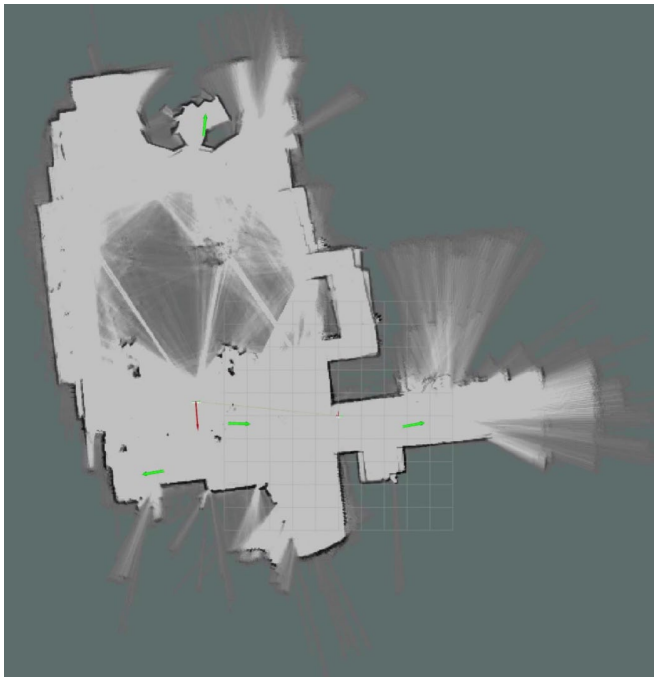
Images provided by the RealSense D435 camera showing both an RGB image (left) and the depth view of the same scene generated by its depth cameras (right).

The D435, along with real-time appearance-based mapping (RTAB-Map) software, allows the rover to generate 3D colored maps of its surroundings. In basic testing, this feature worked very well to generate maps that allowed for easy identification of the objects within; however, because it relies solely on previous images, this method is slow. Full testing of this feature paired with the tracking camera and attached to the moving rover has yet to be conducted.

Waypoint Saving

A waypoint saver node was written to listen if a predefined button on the RC transmitter is pressed and, if so, to remember the rover's current position and orientation. These values are then used to create a marker which is both saved by the node for later use and published for visualization in software such as rviz. Rviz can then display the markers as points of interest on

the map. By simply changing the trigger from a button press to another action, the robot could potentially perform the entire process autonomously.



Waypoint markers (green) within a 2-dimensional map of the arena created by Cartographer's SLAM algorithm with LiDAR input.

Visual Inertial Odometry

The previous SLAM implementation used by the rover relied solely on the laser scan data provided by the LiDAR. The relatively low sample rate of the LiDAR being used caused the map generated by Google Cartographer to slip any time that the rover turned too quickly or objects moved while the rover was in motion, causing the rover to lose its pose and be unable to navigate the previously mapped environment autonomously.

To alleviate this issue, we planned to use visual-SLAM (V-SLAM) using an Intel RealSense T265 tracking camera to assist Cartographer in determining the pose of the rover. Compared to implementing wheel odometry or using the IMU that already exists in the Pixhawk, the T265 camera comes with V-SLAM out of the box which would make it relatively simple to implement into the existing code base on the rover. This method is valuable in that the robot's pose and velocity can be estimated using a minimum of one camera, and it serves as a cheap alternative to relying on GPS and LiDAR-based odometry to obtain visual information [2].

Unfortunately, this feature was not able to be tested on the rover, so it is unclear how much the slipping problem was mitigated.

Autonomous Navigation

An overarching goal was to provide the rover with autonomous navigation capable of traveling within a map to reach a chosen waypoint. A ROS navigation stack was implemented that uses odometry and sensor data to output velocity directions back to the rover. Timed Elastic Band (TEB) planner package is a plugin to this navigation stack that would provide the rover

with autonomous navigation capabilities. TEB was selected because it locally optimizes the path of the rover against a calculated trajectory that updates as the rover's surroundings change. It is able to reroute the rover's projected path to the specified waypoint as the environment changes and new obstacles arise. Unfortunately, this package was never fully implemented, so it is difficult to speculate on its effectiveness without any trials.

Recommendations/ Future Work

Full implementation and testing of the features we were unable to complete would be a good starting place to continue this project. These features include: autonomous waypoint navigation using TEB Planner, full-scale 3D mapping with the Intel RealSense D435 and T265 cameras, and object of interest tagging using object detection. Explorer51's next steps would have included integrating the two RealSense cameras in order to quickly obtain robust three-dimensional maps of the surrounding space, allowing for more advanced algorithms to determine optimal pathing and waypoint navigation under fully-autonomous navigation. These would have been the biggest areas of development had the project not been cut short by COVID-19.

Conclusion

Continued development of a multimodal UGV has the potential to further improve robot operation in critical scenarios. Accurate obstacle avoidance measures and area mapping capabilities pave the way for both semi- and fully-autonomous driving modes, minimizing the risk of damage to the robot and maximizing the amount of information gathered as the robot explores its surroundings. The ability to save and display two- and three-dimensional maps of the space provides tangible data for the robot and the operator to make optimal decisions in time-sensitive operations, as well as a foundation to identify targets and waypoints of interest for further analysis.

Given more time to test and improve the current rover's design and functional capabilities, using such a robot would be beneficial in many ISR applications as a supplementary resource. Being able to search a previously unexplored area and pinpoint targets with close precision, all without employing human actors until necessary, would be invaluable in a multitude of high-risk scenarios.

Acknowledgement

The Explorer51 team would like to thank Nathan Gaul, Grant Showalter, Andy Chapman, Michael Balazs, and the rest of MITRE Corporation for their guidance and valuable contributions to this project.

References

[1] Office of the Secretary of Defense. 2005. "Unmanned aircraft systems roadmap, 2005-2030." apps.dtic.mil/docs/citations/ADA445081

[2] D. Scaramuzza and Z. Zhang, "Visual-inertial odometry of aerial robots," arXiv:1906.03289

Author Information
Gabriel Argush, Student, Department of Engineering Systems and Environment, University of Virginia.
William Holincheck, Student, Department of Engineering Systems and Environment, University of Virginia.
Jessica Krynitsky, Student, Department of Engineering Systems and Environment, University of Virginia.
Brian McGuire, Student, Department of Engineering Systems and Environment, University of Virginia.
Dax Scott, Student, Department of Engineering Systems and Environment, University of Virginia.
Charlie Tolleson, Student, Department of Engineering Systems and Environment, University of Virginia.
Madhur Behl, Assistant Professor, Department of Computer Science and Department of Engineering Systems Environment, University of Virginia.