APPLICATIONS OF AUTONOMOUS NAVIGATION IN NEXT-GENERATION MARS ROVERS

Sarah Dlouhy, Ethan Arneson
Advised by: Dr. Kurt Kosbar
Missouri University of Science and Technology, Rolla MO, USA

ABSTRACT

This paper describes a module used to provide autonomous navigation and obstacle avoidance to a teleoperated prototype Mars rover designed to compete in the 2019 University Rover Challenge. For the competition's Autonomous Traversal task, the rover must be capable of traversing difficult desert terrain in search of visual waypoints. Our design uses a custom Navigation Board (NavBoard), a mobile robotics computer, and a sensor capable of producing a dense point cloud. NavBoard provides quaternion-based orientation data, distance measurements from a 1D LiDAR system, and GPS data over ethernet to a mobile robotics computer. This computer derives a 3D point cloud from a three-headed collinear stereoscopic camera then processes that data along with the data from NavBoard to determine the correct action to navigate through sparsely mapped terrain.

INTRODUCTION

The Missouri S&T Mars Rover Design Team (MRDT) is an undergraduate student organization that designs and constructs a teleoperated, electric rover to compete in the annual University Rover Challenge (URC) hosted by the Mars Society [1]. Each team is expected to compete in tasks wherein they simulate manned missions to Mars. The competition is broken into four tasks: Science, where teams analyze rock and soil samples using a myriad of on-board sensors in search of signs of life; Extreme Retrieval and Delivery, where the rover must deliver objects to astronauts in rocky and otherwise difficult terrain; Equipment Servicing, wherein the rover's operator must manipulate a panel featuring buttons, knobs, etc. to test fine motor control of a robotic arm; and finally, Autonomous Traversal.

For the past three years, URC has contained a task which requires teams to create a system capable of traversing rocky and otherwise difficult desert terrain, sparsely covered in various foliage common to the Utah desert. The rovers must autonomously navigate to approximate GPS coordinates, and then search for a tennis ball which designates the exact target location the judges determined. Eight waypoints were given, with the first four being worth 10 points each and the last four being worth 15 points each, giving a total of 100 points for the task. Competition rules also required each team to provide a visual signal on-rover when each waypoint is reached [2]. The task is confined to area of approximately 1 square kilometer.

The development of the autonomy system began with the selection of an on-board computer to execute the autonomy code, and sensors to ensure proper location and orientation of the rover at any given point in time. MRDT's system consists of a combination of custom and off-the-shelf products. Figure 1 is a picture of a late-stage prototype of the autonomy box which contains the autonomy system.

Figure 1 - The rover, Valkyrie, with Autonomy Box

Autonomous-capable vehicles have operated in large cities with reasonably lax laws regarding autonomous navigation. After tragic incidents like the one in Tempe, Arizona in March of 2018 [4], many jurisdictions have made their relevant legislation stricter. Large cities are typically chosen as test sites because there are more densely labeled, and their maps are more frequently updated. Research groups, like one from MIT [5], have created vehicles capable of navigating roads without predefined maps, by using data from a suite of on-board sensors. However, many of these efforts are confined to developed roads, unlike the off-road application faced in the URC.

This autonomy system is designed to work cohesively with the rest of the communications stack and operates on a 900MHz band to communicate with the remote base station. This allows for the rover to communicate over long distances without direct line of sight. The base station is a remote command and control station, and is used in other tasks at URC to teleoperate the rover; for autonomy, it is only used to initiate the autonomy system and supply GPS waypoints to the system.

This paper focuses on the hardware and software used to create an autonomous navigation system capable of traversing largely unmapped off-road terrain. Many of the team's systems are based on common practice both within industry, and amongst researchers relevant to this system, and are being expanded upon by the former parties listed and hobbyists alike. All of our code and schematics are available on the MRDT github page [6].

HARDWARE

The autonomy system contains a System-on-Module (SOM) as the mobile robotics computer, a custom three-headed collinear stereoscopic camera, a 1D LiDAR unit, headlights for signaling arrival at a target location, and it also relies on information from NavBoard.

SOM

The selected SOM [7] contains 32GB solid state storage, six processor cores, 8GB memory, Wi-Fi and Ethernet connectivity, and a 256core GPU. This unit was selected for the integrated GPU as it allows for CUDA support, giving the rover the ability to perform fast calculations for efficient computer vision processing and handling of dense 3D point clouds.

Stereoscopic Camera

The stereoscopic camera (called Tricam) is a custom model created from three USB webcams as a primary vision system with the intent of having two baselines simultaneously creating two point clouds in order to visualize objects between one and twenty meters away [8]. This was chosen as the primary vision system due to the wide visual range and advantages over LiDAR and similar sensors as a primary system, such as ease of obtaining a 3D point cloud and getting useful data from point clouds during the day in bright sunlight on terrain that has high reflectivity, such as sand.

LiDAR

The LiDAR unit was an off-the-shelf product chosen as a secondary vision system for its ten meter range, greater than 1KHz sample rate, and seventy percent reflectivity coefficient[9], making it ideal for operating in bright conditions. Additional sheathing was also added in the mount to further protect against bright sunlight. A 1D unit pointed at an angle towards the ground was used in favor of something with more dimensions because the LiDAR was designed to be a last resort due to the unreliability of data from these point clouds in bright light- if Tricam could not identify a hazard like a cliff or a steep hill, the data from the LiDAR unit was to be used to stop the rover and give the SOM time to generate a new point cloud from Tricam in order to create a new best course of action for the rover[8].

NavBoard

NavBoard is a custom printed circuit board featuring a GPS unit, inertial measurement unit (IMU), and interface for the LiDAR unit, all mounted on a 120 MHz microcontroller which transmits the data to the autonomy unit over an ethernet network using a custom UDP implementation. The selected GPS unit supports an active antenna for improved satellite acquisition and supports serial communication to and from the unit.

NavBoard runs an Extended Kalman Filter to generate a quaternion from the outputs of the IMU which allows the autonomy system to find pitch, heading, and roll at any given time with better accuracy than using instantaneous measurements[10]. The LiDAR interface is an I2C connection between NavBoard and the LiDAR unit, receiving the distance in centimeters from the target location.

SOFTWARE

The software for the autonomy system consists of a finite-state machine to ensure the system is executing the correct underlying code. The underlying code facilitates the operation of the state machine including driving to GPS coordinates, object tracking, obstacle avoidance, search pattern, as well as control from base station and telemetry back to base station.

State Machine

The main core of the autonomy system is packaged into one of six states of a simple state machine, illustrated in figure 2: Idle, Point-to-Point navigation (P2P), Search Pattern, Ball

Tracking, Goal, and Obstacle Avoidance. The Idle and Goal states are fairly self-explanatory, with the Idle state being just that and the Goal state being the state in which the rover stops and signals to the judges that the waypoint marker has been reached. The other four will be expanded upon later in this section. Each state was created as a separate but interconnected module within the system as a whole.

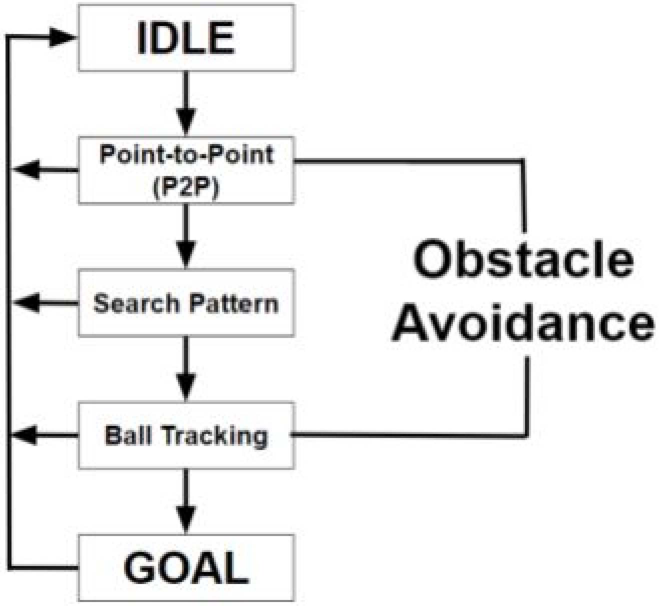

Figure 2 - The Autonomy State Machine

Point-to-point

The P2P module was designed using an application of haversine to determine the distance from the rovers current location to the target location, as well as simpler trigonometry to determine the heading needed to direct the rover from its current location to the target location. The math begins by converting the latitude and longitude values of the GPS locations to radians to allow proper usage of python math functions, then utilizing equations two through nine in the Appendix.

Search Pattern

An Archimedean spiral was chosen for our search pattern because the distance between each arc of the spiral is constant, as shown in equation 1,

$$r = a + b\theta \tag{1}$$

allowing us to exploit that in our implementation and make the distance *b* fit within Tricam's field of view. GPS coordinates corresponding to points on the spiral that were generated from a point that was considered to be within the radius of acceptance from the given approximate GPS coordinate as the center.

Ball Tracking

Ball tracking was achieved by using a simple color mask. This mask isolated yellows within a certain range similar to a tennis ball defined with their RGB values. This mask ran each frame of a live video feed through a series of dilations and color conversions to isolate each pixel within the frame. Color bounds were carefully chosen to contrast with green foliage present at the task site, as small green shrubs like sage bushes had been falsely identified as tennis balls in previous years. Due to heavier rains than average, small sunflowers that were not present in previous years had bloomed, making foliage a much greater problem than it had been prior to the 2019 URC. These sunflowers were falsely identified as tennis balls, rendering the mask useless.

Obstacle Avoidance

Obstacle avoidance was created with the intention of navigating around difficult terrain using visual odometry. Tricam has a dense enough point cloud to use for navigation purposes, similar to methods of Simultaneous Localization and Area Mapping (SLAM) such as ORB-SLAM [12] and PL-SLAM [13]. Our chosen approach did not involve pure visual odometry and instead handled a live point cloud, straying from SLAM due to its unreliability in largely unmapped outdoor spaces, as we tied everything back to GPS coordinates.

Tricam's point cloud was used to create a snapshot of the environment the rover was about to traverse. A point cloud can be created from anything the camera could possibly detect, like the height and depth of large rocks, or anything perpendicular to the z-axis when using camera coordinates, though it could not detect things like a gap hidden between two large, but otherwise traversable, boulders. Tricam accomplished this by breaking up a generated point cloud into sections on the xy plane, averaging out the distance on the z-axis of each section, then calculating the gradient between sectors-whichever had the steepest gradient was to be avoided.

To compensate for unforeseen obstacles, a 1D LiDAR system pointed at the ground at an angle such that the beam would hit the ground at a point approximately twelve inches in front of the front set of wheels was added to the avoidance system [8]. If the distance taken from the LiDAR data is measured to be outside of a certain margin of error, the rover is programmed to stop and search for a safer route.

CONCLUSION

The core design principles used to create MRDT's off-road autonomous navigation system were far more robust than they were in previous years. After solving countless issues arising from legacy code, the rover proved to be able to navigate through difficult desert terrain. The system was functional throughout each of the three legs of the task completed before a mechanical failure prevented the team from being able to progress past the fourth leg.

Progress is being made towards using more visual odometry in favor of relying purely on GPS location in the future. Furthermore, the team is searching for ways to refine our current systems, such as making Tricam easier to calibrate, using neural networks to identify objects instead of color masks, and finding a more reliable GPS unit. The overall goal of this project is to create a reliable navigation system capable of operating in Mars-like environments and assisting humans in everyday operations on a different planet.

REFERENCES

[1] Mars Society, "University Rover Challenge Description," http://urc.marssociety.org, June 2019.
[2] Mars Society, "2019 University Rover Challenge Rules," http://urc.marssociety.org/home/requirements-guidelines, September 2018.
[3] Rausch, Andrew, "A Picture of Valkyrie's Autonomy Box," Internal Resource. May 2019.
[4] Wakabayashi, Daisuke, "Self-Driving Uber Car Kills Pedestrian in Arizona, Where Robots Roam," New York Times, March 2018.
[5] Ort, Teddy, Paull, Liam, and Rus, Daniela, "Autonomous Vehicle Navigation in Rural Environments without Prior Detailed Maps." Massachusetts Institute of Technology. May 2018.
[6] "Missouri S&T Mars Rover Design Team." *GitHub*, Mars Rover Design Team, github.com/MissouriMRDT.
[7] NVidia Corp., "Jetson TX2 Specifications" https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-tx2/, July 2019.
[8] Dlouhy, Sarah, "Autonomy Theory." Internal Resource. February 2019.
[9] Garmin, "LiDAR Lite-v3HP Operation Manual," April 2018.
[10] Marins, J. L., Yun, Xiaoping, Bachmann, E. R., McGhee, R. B., and Zyda, M.J., "An Extended Kalman Filter for Quaternion-Based Orientation Estimation Using MARG Sensors," Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, Maui, Hawaii, USA, Oct. 29 - Nov. 03, 2001, pp.2003-2011.
[11] Dlouhy, Sarah. "Simplified State Machine Diagram." Internal Resource. February 2019.

[12] Mur-Artal, Raoul , Montiel, J. M. M., and Tardos, Juan D., "ORB-SLAM : A Versatile and Accurate Monocular SLAM System," IEEE Transactions on Robotics, 2015. Feb 2015.
[13] Gomez-Ojeda, Ruben, Zuñiga-Noël, David, Moreno, Francisco-Angel, Scaramuzza, Davide, and Gonzalez-Jimenez, Javier. "PL-SLAM: A Stereo SLAM System through the Combination of Points and Line Segments," April 2018.

APPENDIX

In equations 2, 3, 4 and 7, A is the latitude and longitude value for the originating GPS coordinate, and B is the latitude and longitude value for the target GPS coordinate. D is the distance measured in kilometers. H is the target heading, converted from radians to degrees and then turned into a value beginning at zero degrees for due north and increasing in the clockwise direction.

$$d_{lon} = B_{lon} - A_{lon} \tag{2}$$

$$d_{lat} = B_{lat} - A_{lat} \tag{3}$$

$$a = sin(\frac{d_{lat}}{2})^2 + cos(A_{lat}) * cos(B_{lat}) * sin(\frac{d_{lon}}{2})^2 \tag{4}$$

$$c = 2 * sin^{-1}(a) \tag{5}$$

$$D = c * 6371 \tag{6}$$

$$H = tan^{-1}(\frac{sin(d_{lon}) * cos(B_{lat})}{cos(A_{lat}) * sin(B_{lat}) - sin(A_{lat}) * cos(B_{lat}) * cos(d_{lon})}) \tag{7}$$

$$H = H * \frac{180}{\pi} \tag{8}$$

$$H = (H + 360) \% 360 \tag{9}$$