Automated Monocular Exploration with DonkeyCar

Noah Carver, Tim Oates

Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County

Goal & Motivation

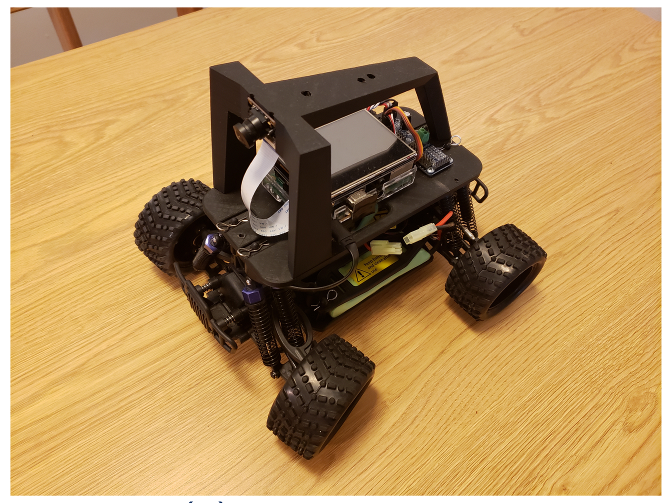
Robotics platforms tend to cost thousands of dollars, making them unreachable to underfunded research programs.
This project aims to perform Automated Exploration - similar to DARPA's Subterranean Challenge - using a cost effective Robotics platform.

The Donkey Car Platform

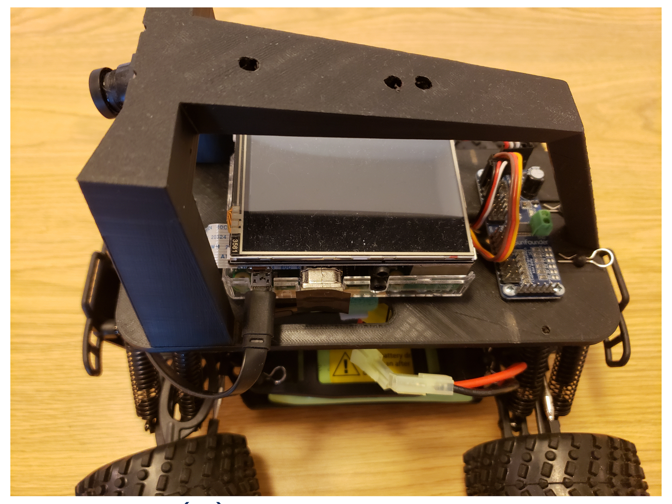
The Donkey Car is inexpensive and feature light:

- Grainy, low frames per second Monocular Camera - no depth estimation
- Non-servo Drive train - no feedback metric on forward motion
- No Odometer - Robot doesn't know where it's going or how fast

However, the car only costs around $200 and is easily assembled from readily available parts:
Just a remote control car, a Raspberry PI & PiCam and a PCA9685 motor driver.



(a) The Donkey Car



(b) Control Circuitry

Autonomous Exploration

- **Simultaneous Localization and Mapping (SLAM):** create map of environment and pinpoint robot location
- **Local planning:** given an incomplete map, get robot from point A to point B
- **Global planning:** choose local planning destinations that will cause complete & efficient exploration of an unknown environment.

Contact Information

Email: {ncarver1,oates}@umbc.edu

Monocular SLAM

SLAM is generally done with binocular camera systems or LiDaR, both of which allow the robot to create depth maps instantaneously. Monocular SLAM only uses **one camera**:

- Requires multiple images from different positions
- Feature extraction finds notable areas in the image
- Tries to detect those features in other images
- Comparing the positions of similar features reveals differences in position of the robot

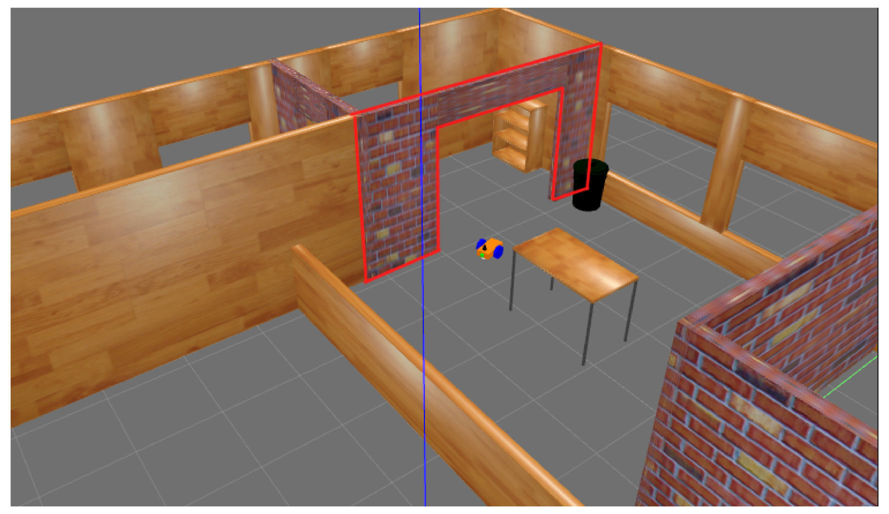


Figure: Exploring in Simulation

Ackermann Steering

Incapable of turning on the spot, requires forward motion to turn.
Ackermann robots are incapable of turning in place.
Tuned Elastic Band (TEB) Local Planning Algorithm:

- Requires an Occupancy Grid and Continuous Odometry
- Produces a high level plan (green line) and a low level plan (red arrows)
- Low level plan can react to minor changes in the Occupancy Grid.

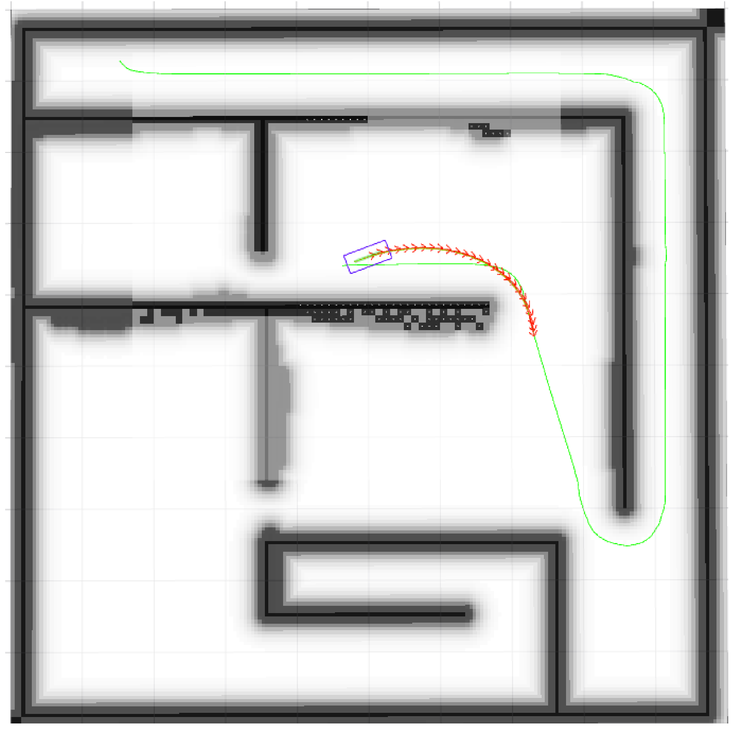
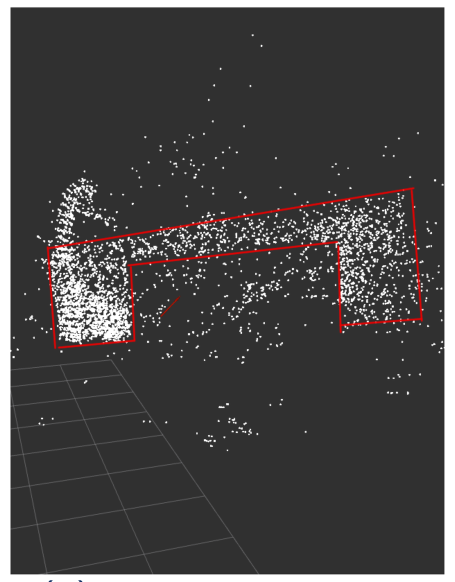


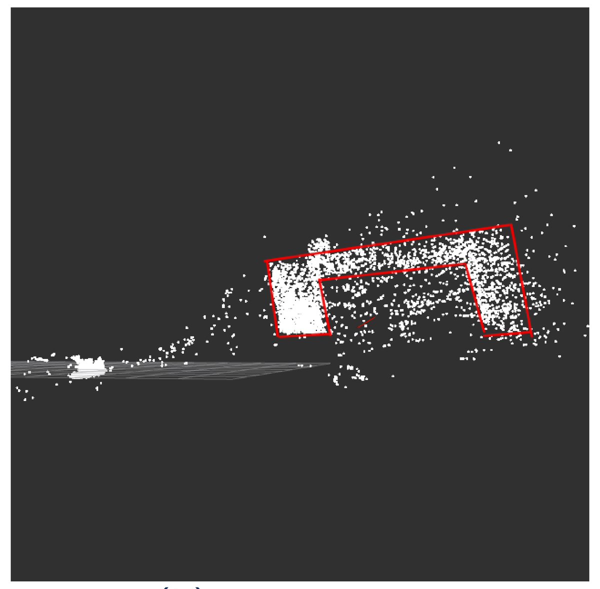
Figure: TEB Local planner on 2d Occupation Grid

Results

The local planner performed very well, but only when assuming knowledge of the environment. The SLAM algorithm could not provide that.



(a) Example Point Map



(b) A Closer Look

Monocular Slam found accurate points, note the archway in the above point maps. This is the brick archway visible in the simulation.
However, because dense features are needed to determine the differences in the robot's position, surfaces as feature-sparse as the wood used in simulation caused the robot to lose it's bearings.

Conclusion & Future Work

Current Monocular SLAM technology is not enough for a robot to create a world map in which to navigate.

- Employ Single Frame Depth Extraction techniques to increase the density and accuracy of 3d readings
- Modify SLAM feature detection to find more features on homogeneous surfaces
- Find inexpensive and accessible LiDaR / Binocular vision solutions that can run on a Raspberry Pi

Acknowledgements

Thanks to Dr. Oates for sponsoring my summer research

References

[1] Raul Mur-Artal and Juan D Tardós.
Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras.
IEEE Transactions on Robotics, 33(5):1255–1262, 2017.

[2] W Roscoe.
Donkey car: An opensource diy self driving platform for small scale cars.

[3] Christoph Rösmann, Frank Hoffmann, and Torsten Bertram.
Integrated online trajectory planning and optimization in distinctive topologies.
Robotics and Autonomous Systems, 88:142–153, 2017.