LOW COST COMPUTER VISION FOR AUTONOMOUS SAFETY GROUND VEHICLES

Jacob Hartzer
Texas A&M University, College Station, Texas, United States

1. INTRODUCTION

The North Texas Tollway Authority is an organization that runs and maintains toll roads and road infrastructure in the North Texas Area. They are responsible for collecting tolls and to use those tolls to fund projects and services connected to the infrastructure and use of these toll roads. This includes servicing the roadway itself, mobile repairs or service, and the monitoring of roads for safety and traffic conditions. This allows the NTTA to respond and adapt to changing conditions throughout the day and to better serve those who use their roads.

Included in the monitoring of roads, the NTTA will freely respond to accidents or break-downs on the road. In addition to road work, this means NTTA needs to have the ability to access and safely cordon portions of the road to work or provide assistance to drivers.

In order to achieve this need, NTTA has two attenuator trucks that are capable of withstanding a rear or side impact that are used to protect road workers while they are servicing the road or assisting drivers while not constricting access to the road itself. Additionally, there are numerous response trucks that are used in service but that also can also be used to block off a lane from the rear. Shown in *Figure 1*, the service truck in addition to the attenuator truck can be used to block off the entire shoulder and one lane while still allowing access to work within the lane.

Figure 1. Attenuator Truck (orange) with Service Truck (black)

These attenuator trucks and service trucks are vital to the safety of the NTTA workers due to the unpredictability of drivers at all times of day. However, there have been issues with the number of vehicles that have collided with these trucks. Despite working as designed to protect the workers, the attenuator trucks have long downtimes after collision and a high cost to repair. At times, both trucks have been out of service, preventing the workers from being able to safely service the road or aid drivers. And while the response trucks can also be used to block traffic and protect workers, collisions with these trucks have the same issue of large costs for repairs.

NTTA has found that placing cones in a wedge shape in front of the vehicles drastically reduces the number of collisions with the trucks by giving drivers additional warning that they are approaching road work. This is especially effective around corners and roads with high speed limits. The tradeoff is the placement of cones requires the workers to manually drop off and pick up the cones up to 120 feet away from the protection of the service and attenuator trucks. Due to the high risk of personal injury to workers, this was deemed unacceptable. Therefore, the NTTA desired a device that would be able to solve this problem.

2. PROBLEM DEFINITION

2.1 Objective

The objective of this research project was to develop a robotic platform that is capable of automatically placing cones in a defined wedge shape within the starting lane. This system should be able to place cones in a wedge following the curvature of lane lines and return them after a user-given signal. Also, this system should be able to simple to use and not require significant training.

2.2 Constraints

This platform must be capable of operating on concrete highway surfaces like those of North Texas toll roads. This includes concrete with gravel, cracks, and other small surface features. The system cannot rely on magnetic sensors, due to fiber-optic cable in the road. The system cannot require infrastructure to be laid on the highway. And most importantly, the system must be portable, reusable, and cost less than $1500 per cone unit.

3. PLATFORM DESIGN

3.1 Preliminary Platform Design

The first solution developed by the lab made use of a 1/5th scale remote controlled racing buggy that would use magnetometers to determine heading and make use of road mapping information to place itself within the lane. However, it was determined that the magnetometer data had significant noise from the fiber-optic cables laid in the NTTA toll roads. This made this sensor infeasible. The buggy was much faster than was required, and this also made it difficult to accurately control. Finally, a GPS that meets the cost constraint of the system would not be able to meet the accuracy constraint for making a proper wedge shape for deflecting drivers. As such, this solution was deemed infeasible. The buggy was able to be put to use on a different NTTA project, but it was determined that a redesign of the base platform would be required.

Figure 2. 1/5th Scale Racing Buggy

3.2 Improved Platform Design

In order to meet the requirements and constraints of the project with the additional information learned from the first iteration of the design, a new design was created. It was decided that this new design could be significantly slower than the first iteration, could not use magnetometers or GPS, and would ideally be able to move in any direction for the implementation of obstacle avoidance. As such, the new platform was designed around an omnidirectional base kit available from SuperDroid Robots.

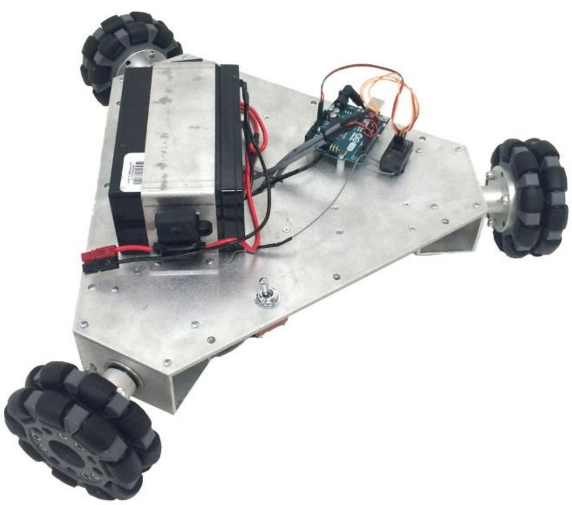

Figure 3. Omni-Base from SuperDroid

This base supports the Omni-directional wheels that NTTA requested after the first demo, and is capable of much more accurate position control due to encoders on each of the wheel motors. Outfitted with RoboteQ motor controllers, this platform is capable of closed-loop speed control with zero steady state error. Additionally, the size of this platform made it much easier to transport and is a good shape for mounting sensors and a collapsible cone when in use.

4. MOTOR CONTROL

Controlling the motors proved to be a rather difficult task. The motor controllers were not designed to communicate with a raspberry pi system, but due to cost limitations, this was the most feasible option.

Using a package called *roboteq-indigo*, it is possible to communicate with the RoboteQ controllers by sending serial commands via USB. This package also maintains code for naming and identifying separate motor controllers if multiple are used. Because each controller only has inputs for two motors, this proved extremely useful. An additional package called *roboteq_python* was used to retrieve ROS messages for velocity commands and translate them into commands to the *roboteq-indigo* package. These commands are given as a twist message, where the (x, y, theta) values are interpreted as velocities instead of positions. Combining these two packages made it possible to control the robot base using ROS messages in python. This information flow is summarized in *Figure 4.*

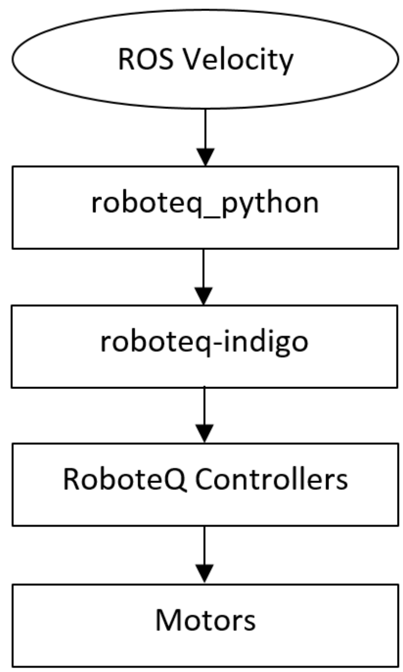

Figure 4. Velocity Control Flow Diagram

5. OPEN LOOP POSITION CONTROL

5.1 Open Loop Testing

Using the closed loop velocity control process explained in the previous section, it was possible to relatively accurately control the position of the base using open-loop control. By giving the controllers a velocity input for a given period of time, a particular reference position was achievable. This process is summarized in *Figure 5.*

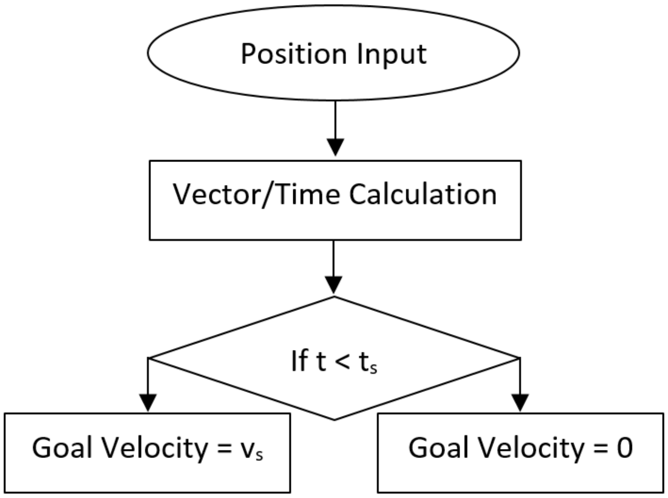

Figure 5. Open Loop Control Flow Diagram

When tested on flat concrete with minimal bumps, this process was found to have a horizontal position error of 0.83 % or 1 foot for every 80 travelled. This is the key dimension as the robot cannot drift outside the lane else it will collide with oncoming traffic and fail to properly make the wedge shape to control the flow of vehicles. Because of this low value for a relatively inexpensive control system, further testing was done on actual highway surfaces as shown in *Figure 6.*

Figure 6. Platform Running on the Road

When tested on actual highway surfaces, it was found that errors in initial positioning and the uneven surface of the highway as exemplified in *Figure* 7 caused an accumulation of error. The slope and hard plastic wheels caused the base to lose its initial heading and turn. This caused an accumulation of error over long distances, leaving the platforms 8 feet horizontally from their desired positions at 120 feet, which equates to 6.67% error. This is an unacceptable level of error because the base could drift into an oncoming lane as shown in *Figure 8*.

Figure 7. Surface Defects seen by Platform

Figure 8. Accumulated Error

An additional issue that arises from just using open loop control is the requirement that the initial position and trajectory be very accurate. Small errors in initial position and angle accumulate very quickly, and this is quite unreasonable given the ideal use case by road workers. As such, there cannot be reliance on the user to place the robot exactly and align the robot with the lines of the road. Moreover, this system does not account for drastic slopes in the road or curvature of the lanes. Therefore, despite its initial accuracy, open loop control is simply not robust enough for such a complex scenario.

It was concluded from these tests that the base would require some form of feedback control that is both inexpensive and simple to control for a user once implemented.

6. LANE DETECTION ALGORITHM

The simplest and cheapest method of feedback for the mobile platform was computer vision. Using a small, base-mounted camera, the cone platform would be able to detect lane lines and curvature in order to correct its position within the lane to more accurately create a wedge shape.

6.1 Method

The lane detection algorithm method was selected after conducting a literature review of inexpensive methods of implementing lane detection. In particular, the work done by Shibuya (2017) and Kaur and Kumar (2015) was instrumental in outlining the processes for computer vision lane detection used in this project. The image processing algorithm follows the flowchart shown in *Figure 9.*

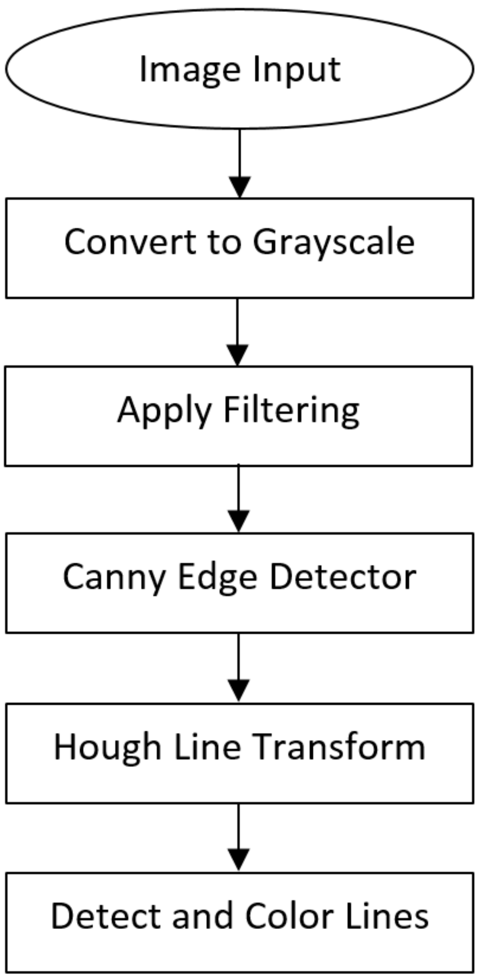

Figure 9. Lane Detection Flowchart

In this case, the detection of white lines is not necessarily easier with color data, and this can slow down processing time. Therefore, the input image is converted to grayscale to simplify the computer vision processing.

Figure 10. Camera Input Converted to Grayscale

A simple Gaussian blur is applied in order to reduce the effects of noise in the image from shaking as well as to remove smaller features of the image without having to downsize the image.

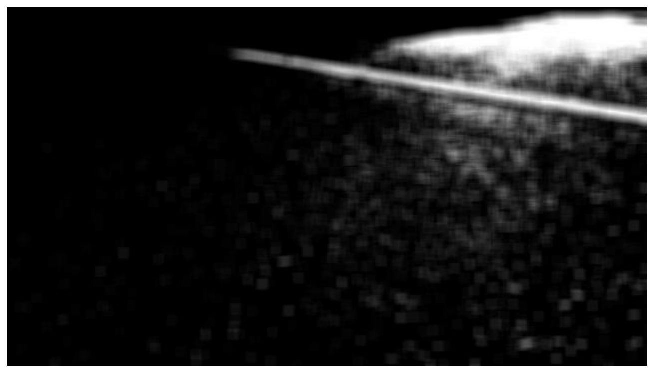

Figure 11. Applied Gaussian Blur

The canny edge detection algorithm available in the OpenCV package is then used to detect edges. These edges are shown in *Figure 12* by drawing points where edges were found.

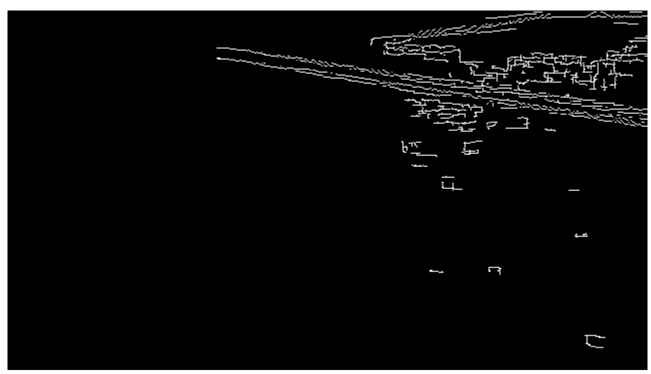

Figure 12. Applied Canny Edge Detection

Finally, using Hough line transforms, it is possible to find lines from the edges. These lines were filtered by a minimum length and expected location within the image. All detected lines using the tuned parameters are shown in green with the blue line being the predicted actual location of the lane line. This process works for multiple angles and times of day, but is currently limited to daytime.

Figure 13. Results of Hough Line Transform

7. GEOMETRIC POSITION ALGORITHM

To make use of the lane detection algorithm, it is necessary that the platform be able to predict its position within the lane using the detected lane line. This is done using homogenous frame transformation and the pinhole camera model.

7.1 Method

To determine the position of the lane line with respect to the platform, the setup shown in *Figure 14* was used.

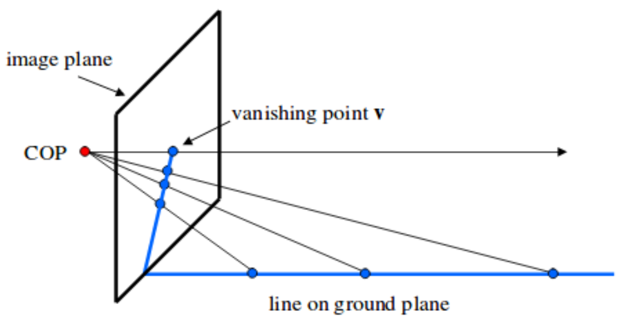

Figure 14. Basis for Geometric Estimation

This algorithm assumes the center and orientation of the platform with respect to the ground plane is given by rotation matrix R and translation vector t. These are constants found by the position of the camera on the platform. Assume a point on the ground plane is given by

$$P_g = [X_g, Y_g, Z_g] \tag{1}$$

This point can be represented in the camera frame by

$$P_c = R \cdot P_g + t \tag{2}$$

By definition, the center of the camera in camera coordinates is

$$C_c = [0, 0, 0] \tag{3}$$

Therefore, in ground coordinates the camera position must be

$$C_g = -R^T \cdot t \tag{4}$$

The ground plane is defined as

$$Z_g = 0 \tag{5}$$

Given a pixel in camera frame position

$$q = [u, v] \tag{6}$$

It is possible to rewrite it in homogenous image coordinates

$$Q = [u, v, 1] \tag{7}$$

Therefore, its position in camera coordinates is given by

$$Q_c = K^{-1} Q \tag{8}$$

Where K is the intrinsic parameters matrix of the camera model. The same point in world coordinates is

$$Q_g = R^T \cdot Q_c - R^T \cdot t \tag{9}$$

All the points P_γ that lie on the line connecting the given camera pixel and the camera center are given by

$$P_\gamma = C_g + \gamma \cdot (Q_g - Q_c) \tag{10}$$

Therefore, the intersection of this three-dimensional line and the ground plane can be used to find the location and orientation of the line with respect to the center of the platform by applying this method to the two ends of the found line from the camera image.

Using this method, it is possible to provide a feedback controller the location and orientation of the platform within the lane, which can then use this information to correct the trajectory of the robot.

8. PLATFORM POSITION CONTROL

Using the information provided by the geometric position algorithm, simple closed-loop correction using a proportional controller was implemented to correct the distance and angle of the platform with respect to the lane line. This controller used a desired distance and orientation as a reference input, and changed its trajectory based on a proportional control from the error in distance from reference.

Implementation of this algorithm was relatively successful, whereby the platform was able to maintain a distance and orientation from a lane line over the course of a mission. The error was significantly reduced to within 6 inches of the final desired horizontal location.

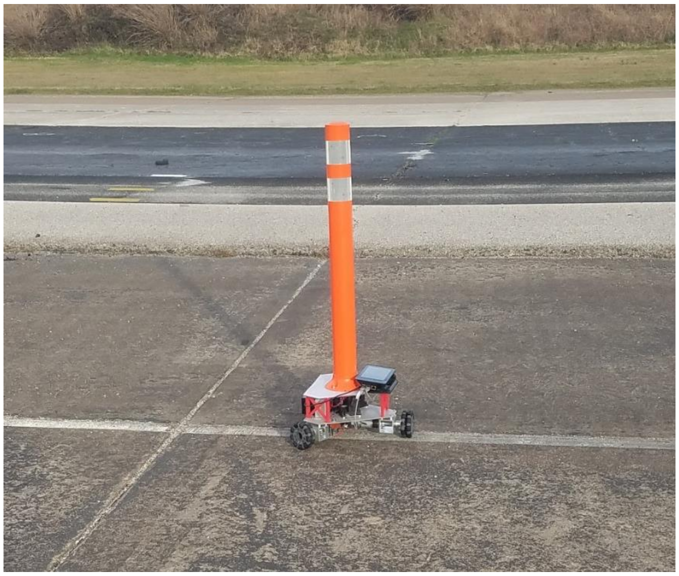

Figure 15. Cone Stopping on Line with Feedback Control

The biggest issue at the moment is the platform's reliance on a steady white line to perform its feedback control. Moving forward, this dependence shall be removed.

9. FUTURE WORK

The current implementation of the platform is limited in many regards, the largest being only operable in situations with solid white lines. As such, implementing more robust algorithms that can handle yellow lines as well as stripes is a must. Moving forward, a Kalman filter will be implemented to increase certainty in position estimation as well as to help handle the low certainty regime when there are gaps in the lane lines.

Additionally, it is desirable to replace the current vision image with an infrared, so that an infrared light may illuminate the lane lines at night. This has the added benefit of reducing error due to shadows, direct sunlight, or the headlights of moving vehicles.

Finally, a user-friendly initialization is to be implemented. Currently, the user must input distances and generally point the platform in the correct direction. Moving forward, it is desired to use a camera mounted on the deployment truck that would initialize the location of the platform and give it initial curvature data for the lane. This would simplify deployment for the end-users and increase accuracy of localization.

10. CONCLUSIONS

In conclusion, this research project was able to take the current issue faced by NTTA and implement a robotic, Omni-directional platform that is capable of localizing and placing cones in a wedge shape in order to increase road worker safety. Improvements were made on the open loop control of the platform using computer vision feedback that was able to significantly reduce the error to within project tolerances. Moving forward, the project will continue to refine the computer vision algorithm as well as implement simplifications that will aid the end user and reduce overall error.

REFERENCES

Kaur, Gurveen and Dinesh Kumar. *Lane Detection Techniques: A Review.* 2015

North Texas Tollway Authority. *North Texas Tollway Authority: What We Do*. 2019

OpenCV. *Hough Line Transform.* 2019

Redcat Racing. *RAMPAGE XB-E 1/5 SCALE ELECTRIC BUGGY.* 2019

Shibuya, Naoki. *Finding Lane Lines on the Road*. 2017

SuperDroid Robots. *Programmable Tri-Wheel Vectoring Robot Kit - IG32 Medium Duty SB.* 2019