# Autonomous Vehicular Traffic Control (OnTRAC)

Raghav Bhardwaj, Nicholas Miele, Daniel Revie, Poornima Suresh, Kevin Zheng

{rb721, ngm37, djr199, ps778, kz114} @scarletmail.rutgers.edu

Advisor: Prof. Waheed Bajwa

RUTGERS THE STATE UNIVERSITY OF NEW JERSEY

## Background and Motivation

Antiquated features of traffic flow control such as traffic lights still the dominate the vast majority of roads and highways. This requires drivers to bring their vehicles to a complete stop before they resume driving, which causes a delay that propagates throughout the road. This, along with outdated infrastructure that has been in place for decades, are primary causes of traffic congestion. Our method for addressing this issue is to remove the root of the problem: we make use of connected vehicles to eliminate the need to completely stop at an intersection and drastically reduce the delay propagated through the line of vehicles.

## Abstract

This solution is designed to work in parallel with the advent of self-driving vehicles. Using sensors that would be present on autonomous vehicles already, our system would have the approaching vehicles communicate and produce a local map of the intersection with relative distances from each other. This information can then be used to generate a procedure for passing through the intersection by adjusting the speeds of the vehicles. The goal would be to minimize the speed changes necessary to weave the streams of traffic between each other. Our project entails the development of the algorithm, a software simulation, and finally a scaled test using physical hardware with retrofitted RC cars.

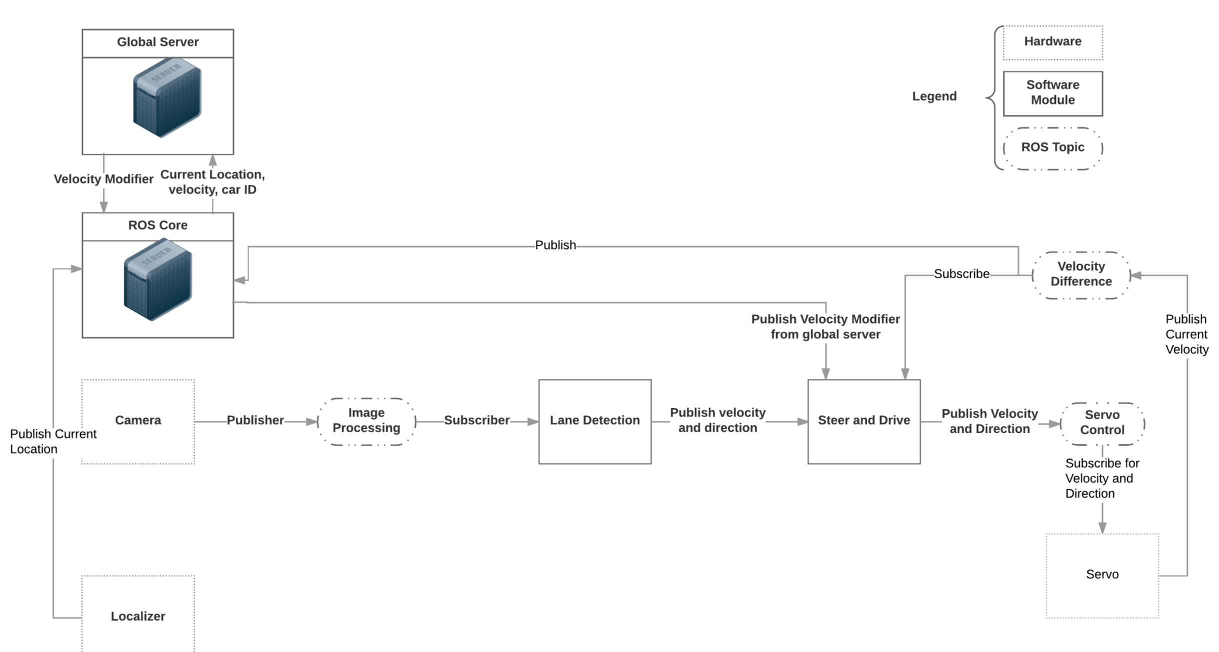

*Figure 1: System diagram*

**Objective**: OnTRAC aims to reduce traffic by making cars weave through intersections without stopping for red lights. A central server at the intersection would mediate traffic ensuring contention is reduced.

## Acknowledgements

We would like to thank the ECE department for offering us this opportunity, as well as Professor Bajwa for his guidance and encouragement.

## Methodology

### Hardware

Real-time deterministic control of steering and closed loop feedback of drive speed using hall effect sensors done by Arduino. Raspberry Pi issues commands to the Arduino via i2c interface. Steering is done using traditional rack-and-pinion-like steering as seen on current automobiles.

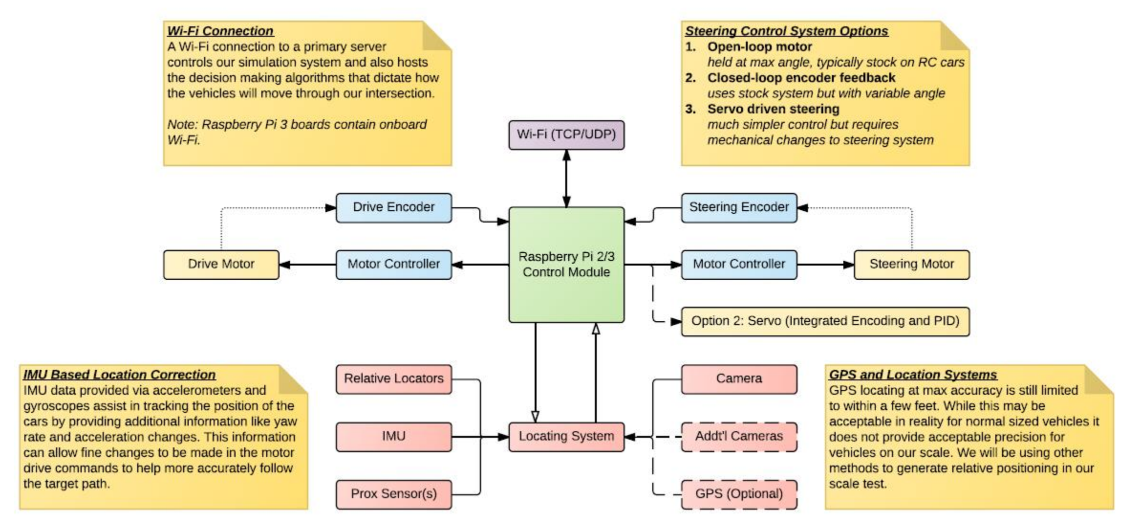

*Figure 2: Hardware flowchart*

### Server Velocity Regulation:

- As cars approach an intersection, a virtual probe representing each car is sent out towards the intersection which determines if it would collide with another vehicle in the intersection.
- If a no collision scenario is determined then the car continues to proceed at it's current speed otherwise the server forces the car to slow down just enough to avoid the collision after which the car returns to it's normal speed.
- This approach causes the cars to weave through each other at the intersection without any collisions assuming no hardware failure occurred

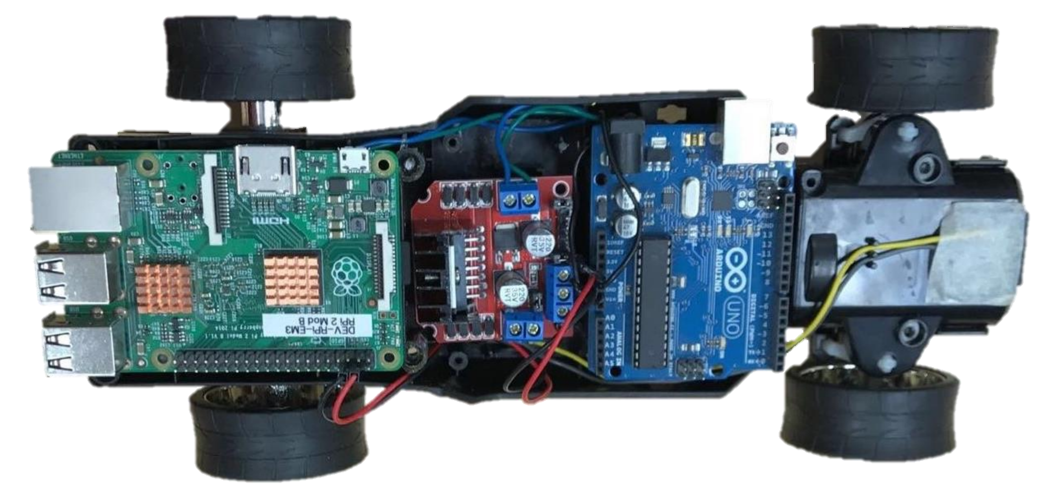

*Figure 3: Modified RC car controlled by Arduino (right) and Raspberry Pi (left)*

### Computer Vision

- The lanes on our track are represented by white tape on a dark surface
- Edge detection using contours in OpenCV allows us to detect curved and straight lanes
- Segmenting each frame and finding the centers of each line contour gives us more precise estimations of the car's position relative to the lane
- If only one lane is visible, car is instructed to turn in opposite direction get back on track
- Blurry frames rejected to avoid bad samples

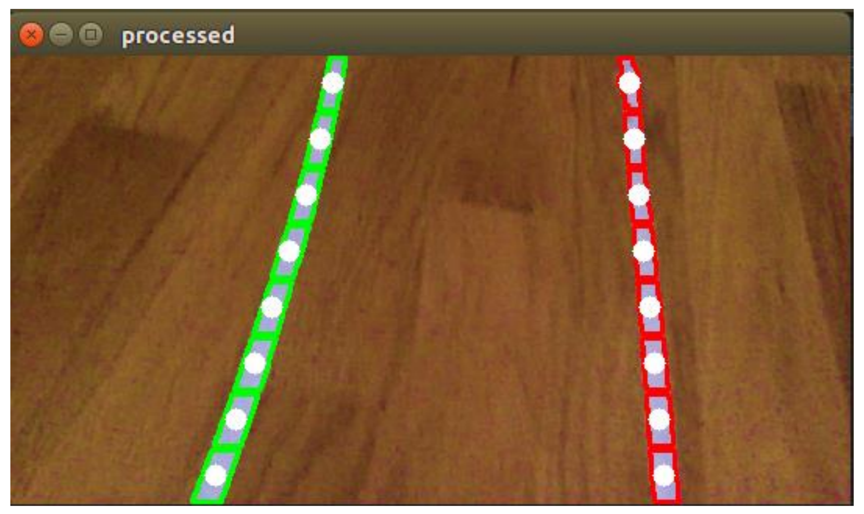


*Figure 4: Lane detection with contours in OpenCV*

## Research Challenges

- **Hardware**: optimizing real-time PID control loop, interfacing hardware control to camera and server driving software.
- **Computer Vision**: feature detection should be rigorous enough to identify straight and curved lanes, and should correct for angle and shift offsets
- **Communication**: Creating a client instance on raspberry pi's to communicate the vehicles current position and velocity to the server. Ensuring that server receives the right data and identification from the car and can reply with velocity updates.
- **Algorithm**: Formulating a robust velocity regulation algorithm to handle high levels of traffic saturation.
- **Connecting the physical and virtual models**: Since RC cars do not have the ability to decelerate and accelerate as dramatically as a real car, we assumed constant maximum speeds and deceleration to ensure the model was suited for the RC car and its limitations.
- **Scalability**: In order to demonstrate a larger volume of traffic, we pushed the virtual simulation to display a higher level of saturation requiring tighter constraints.

## Results

The cars in the simulation were able to course through the intersection in an interleaved configuration without crashing. We have expanded this to multilane intersections. The modified RC cars are steered in response to the server and the image processing code in order to stay within the lanes and pass through the intersection.

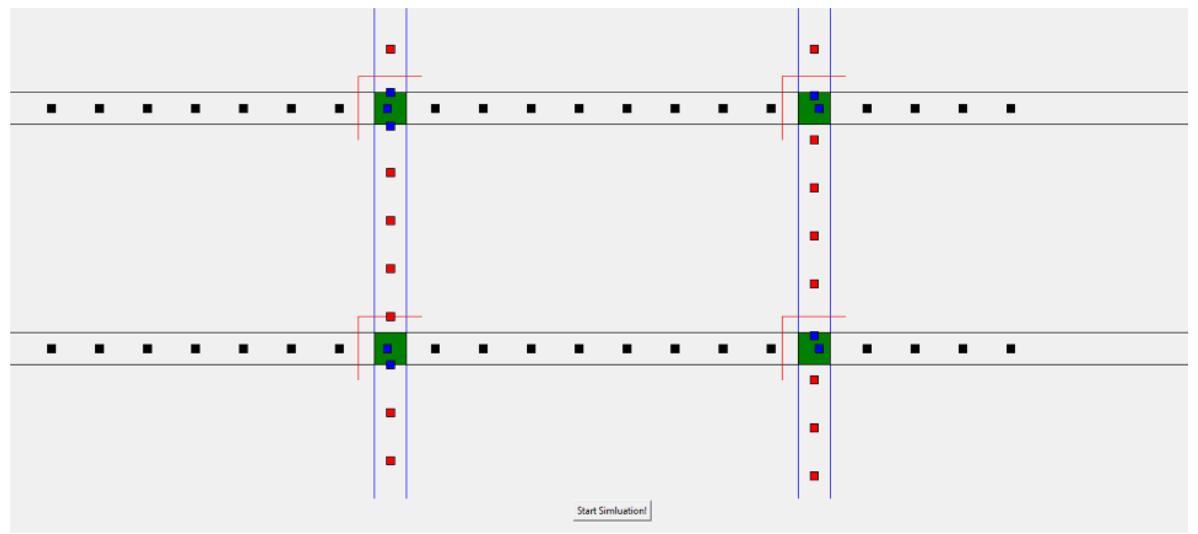

*Figure 5: Simulation of algorithm with Python GUI*

## Future Improvements

- Rather than having a central server mediate traffic, we plan to have the communication decentralized by having cars communicate directly. This would save us the cost for setting up servers on intersections and help immensely with scalability.
- The lane detection algorithm that we are employing currently is robust, it still falls short in many scenarios. Replacing the current algorithm with a trained Convolutional Neural Network would greatly improve scalability.
- Hardware control could improve with higher performance sensors. Currently we are limited to 4 pulses per revolution of the wheel. Real vehicles possess sensors with significantly higher pulse resolution on the order of 72+ pulses per revolution.

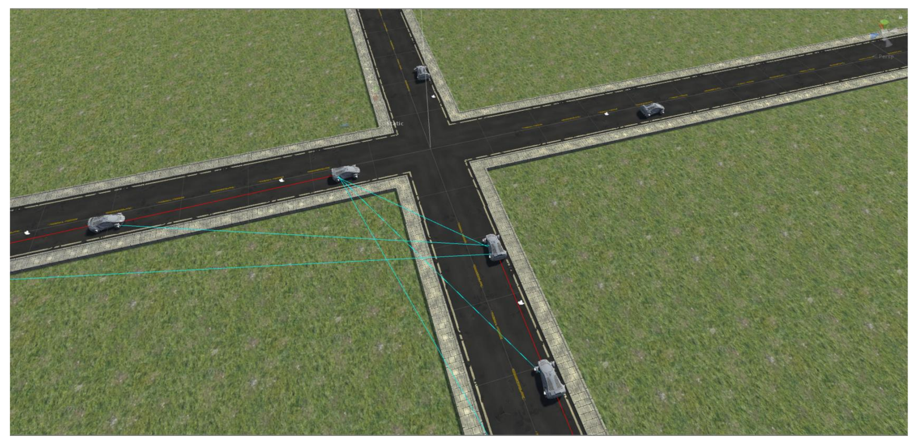

*Figure 6: Unity 3D simulation of cars with proximity sensors following our algorithm*

## References

[1] Tao Jiang, Srdjan Petrovic, Uma Ayyer, Anand Tolani, Sajid Husain. Self-Driving Cars: Disruptive or Incremental? Issue 1 June 2015. http://cet.berkeley.edu/wp-content/uploads/Self-Driving-Cars.pdf

[2] University of Texas-Austin Autonomous Intersection Management: http://www.cs.utexas.edu/~aim/

[3] K. Dresner and P. Stone "A Multiagent Approach to Autonomous Intersection Management", Volume 31. http://www.jair.org/papers/paper2502.html ]