ANAGLyPh: Autonomous NAviGation Learning Platform

Josh Wolper
Jeff Bulick
Y Nguyen

“ Hands-on experience is the best way to learn about all the interdisciplinary aspects of robotics.”

- Rodney Brookes

Motivation

- Multi-robot systems is an exploding research area
- VADER Laboratory work to date has been almost entirely in simulation
- The state of the field motivates the need for real-world experiments

Motivation (cont’d)

- Existing commercial solutions for robotics are expensive ($20,000+)
- *k*-existing commercial vehicles for multi-robot research is prohibitively expensive
- Want to develop a platform that is
  - Affordable
  - Works outdoors in unstructured environments
  - Has rich 3D perception solution
  - Very high processing power
  - Leverages the latest software tools to facilitate application development
- First solution: RoSCAR (Robot Stock Car Autonomous Racing)
  - Developed for a robotics course Fall 2013
  - Students wrote programs to follow an indoor track using visual feedback

Structure of Presentation

- Introduction
- Hardware
- Software
- Comparison of Exteroceptive Sensors
- Conclusion & Future Extension

RoSCAR Overview

- Based on Traxxas Slash 2WD 1/10 scale RC car
- On board motherboard running Ubuntu 12.04
- Applications for RoSCAR written in Matlab and uses Spread for IPC
- Feedback is limited
  - Wheel encoder gives linear velocity
  - Camera gives RGB-D visual feedback
    - Only works indoors
    - 30 Hz
  - No inertial feedback
  - No measurements to bound odometry error

Our New Solution: ANAGLyPh

Nuccar
Intel Nuc
Asus Xtion camera

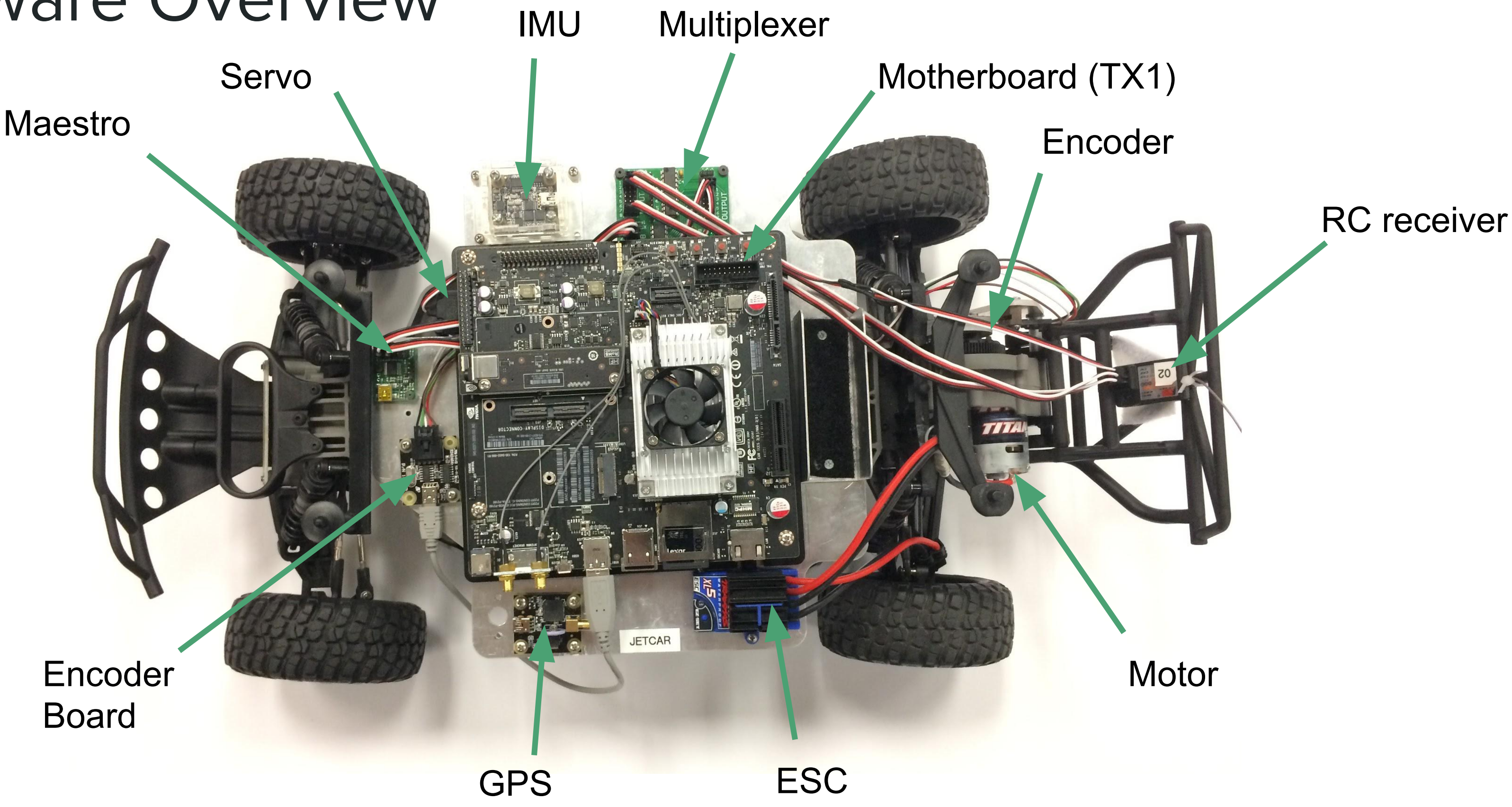
Jetcar
Jetson TX1
ZED camera

Project Goals

- Transition to ROS (Robot Operating System)
  - “Industry Standard”
  - Clean and fast interprocess communication with ROS and large number of packages supported, but...
  - ... steep learning curve
- Integrate additional sensing to support outdoor operations
  - Angular velocity feedback
  - Measurement feedback
- Integrate and evaluate competing exteroceptive sensors
  - Asus Xtion RGB-D PrimeSense
  - Stereo Labs ZED

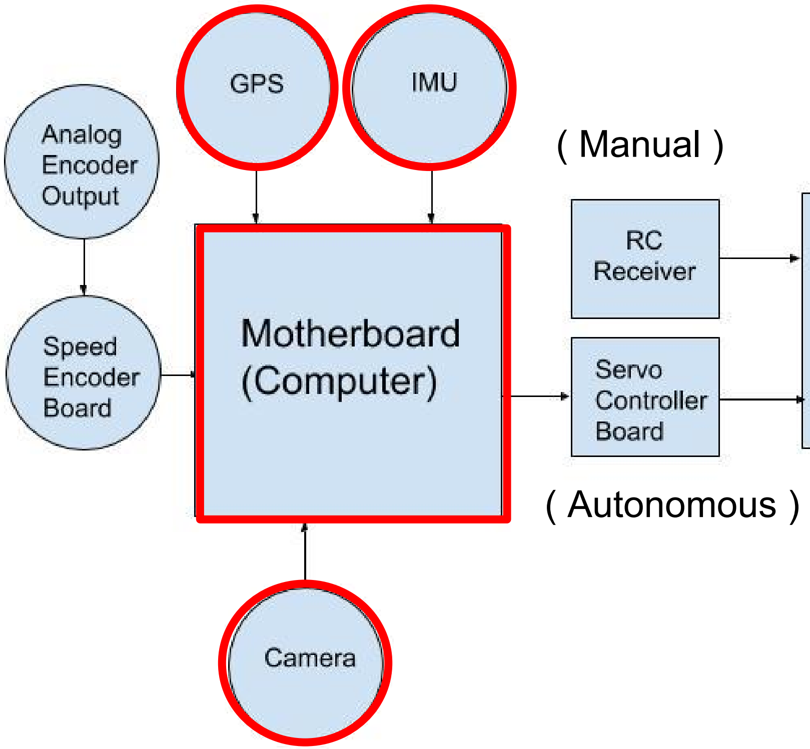
Hardware

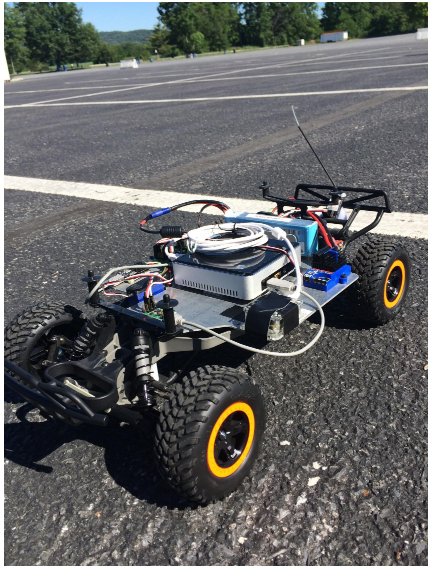
Hardware Overview



ANAGlyPh System Design

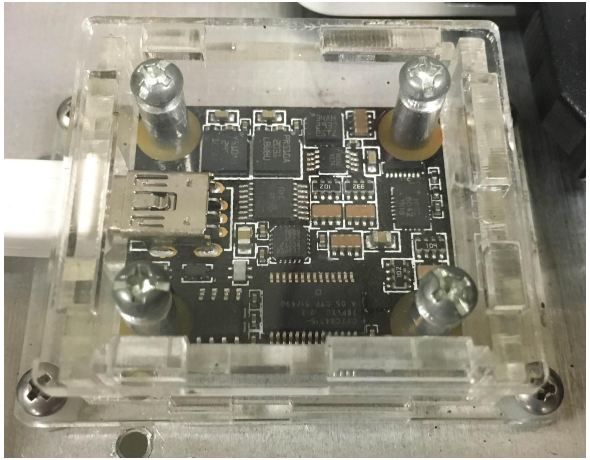
- Improvements on Roscar



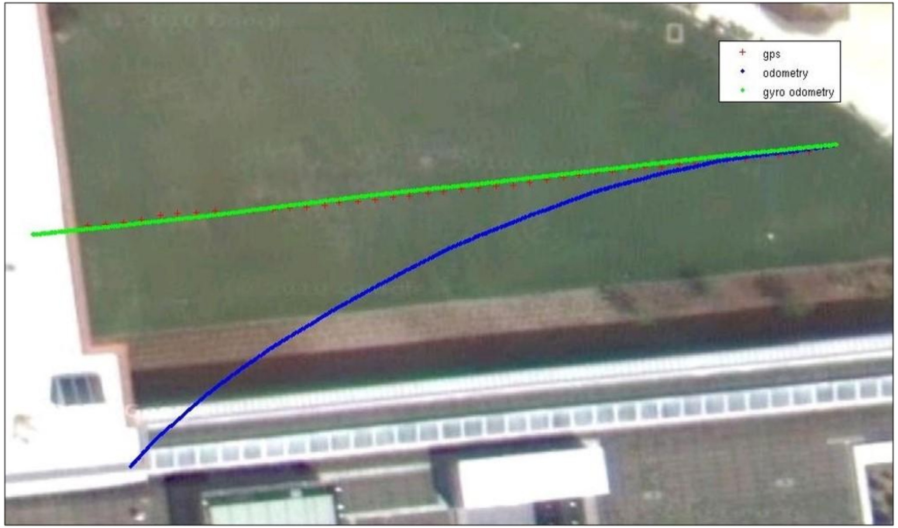


IMU Incorporation

- angular velocity (yaw z-axis) for gyro-corrected odometry
- Wheel slippage, skidding, sliding, etc., handled



IMU Comparison



Improving Odometry

- Kinematic predictions, minimizing error

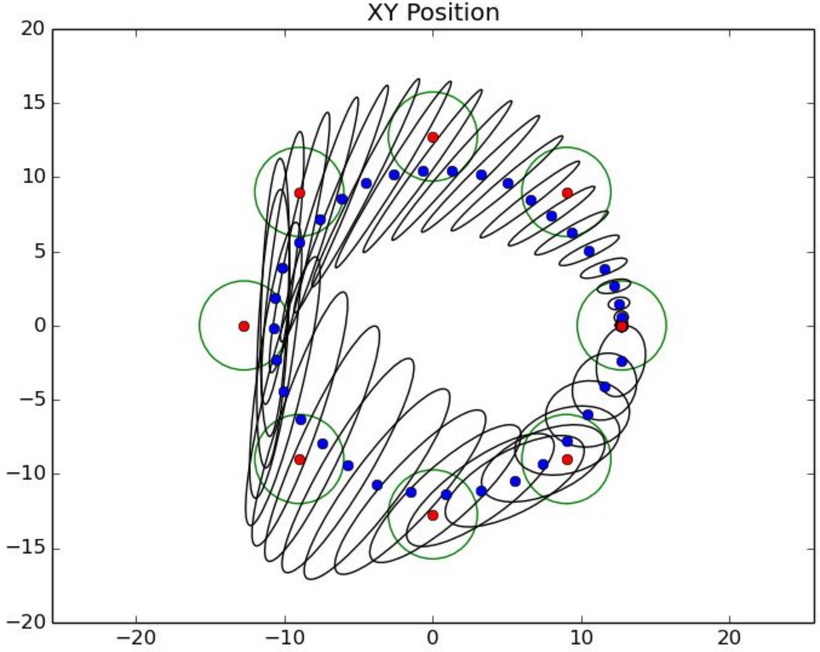
v = linear velocity
ω = angular velocity

$$\Delta\theta = \omega * \Delta t$$

$$x_{t+1} = x_t + (v\, cos(\theta_t) * \Delta t)$$

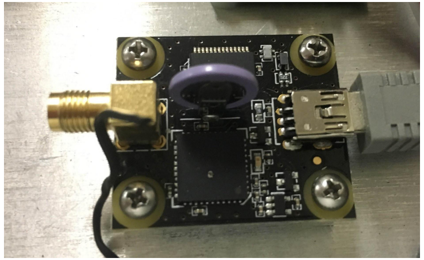
$$y_{t+1} = y_t + (v\, sin(\theta_t) * \Delta t)$$

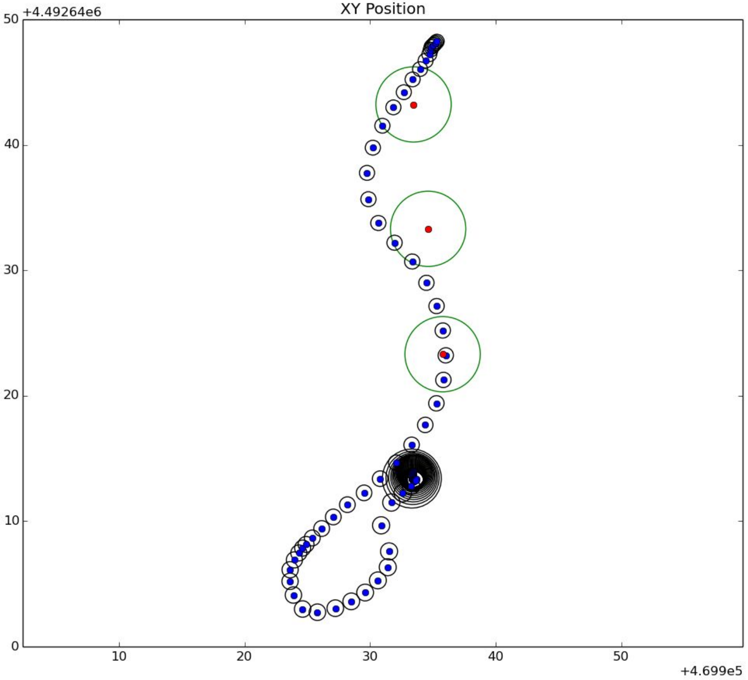
$$\theta_{t+1} = \theta_t + \Delta\theta$$



GPS Incorporation

- Transform to Universal Transverse Mercator (UTM) coordinate system
- UTM 2D slices of Earth's surface area
- Bound the odometry error





Roscar versus ANAGLyPh Computers

Roscar Setup

- CPU: Gigabyte GA-H61N-USB3 Intel H61 Express Chipset Mini ITX DDR3 800 Intel - LGA 115 Motherboard, Intel Core i3-3220T processor
- Power: Up to 90 Watts

Nuccar

Intel Nuc

- mini PC
- 1 x Core i5 5010U / 2.1 GHz
- Dimensions: 4.5 in x 4.4 in x 1.4 in
- 7.5W TDP
- Best for general purpose

Jetcar

Jetson TX1

- GPU: 1 TFLOP/s 256-core with NVIDIA Maxwell™ Architecture
- CPU: 64-bit ARM® A57 CPUs
- Up to 15W TDP
- Better for graphical processing

Overview of Hardware: Cameras

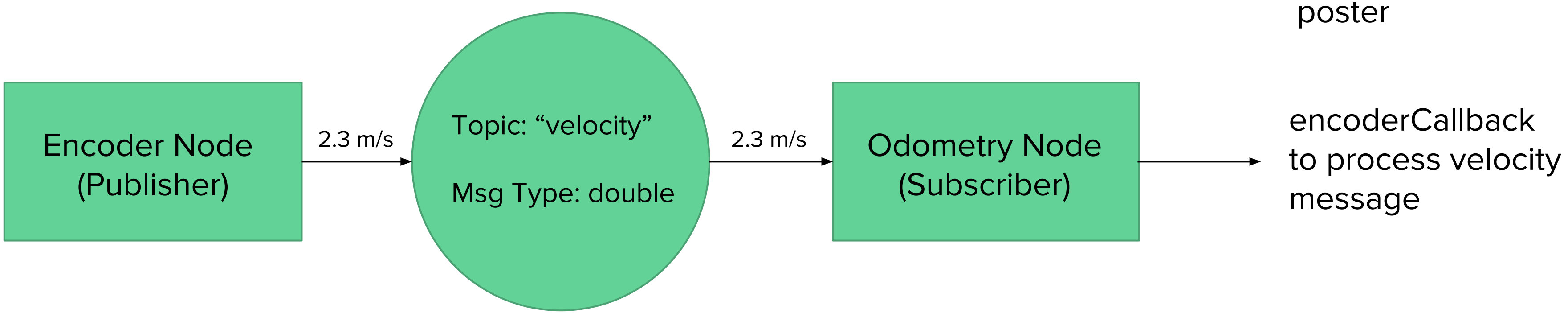
- Asus PrimeSense (structured light)
  - Extremely cheap but restricted to indoor use
- ZED Camera (stereo vision)
  - Interested in low price and outdoor capability
  - Object needs texture for camera to pick up
  - Stereo much more affordable, Bumblebee $2-3,000

Software

The Robot Operating System (ROS)

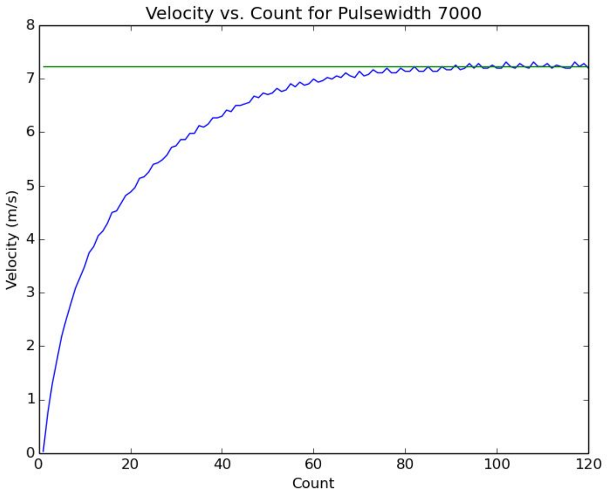
- Great for inter-process communication (IPC)
- Nodes
- Message types and topics
- Publishers
- Subscribers and Callback Functions

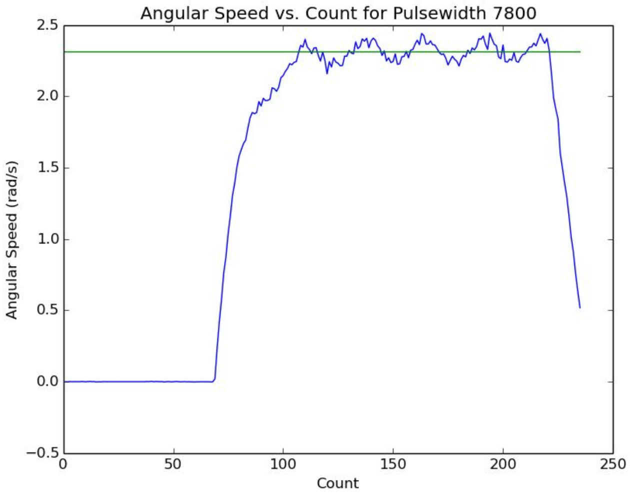
ROS Jade Distribution poster



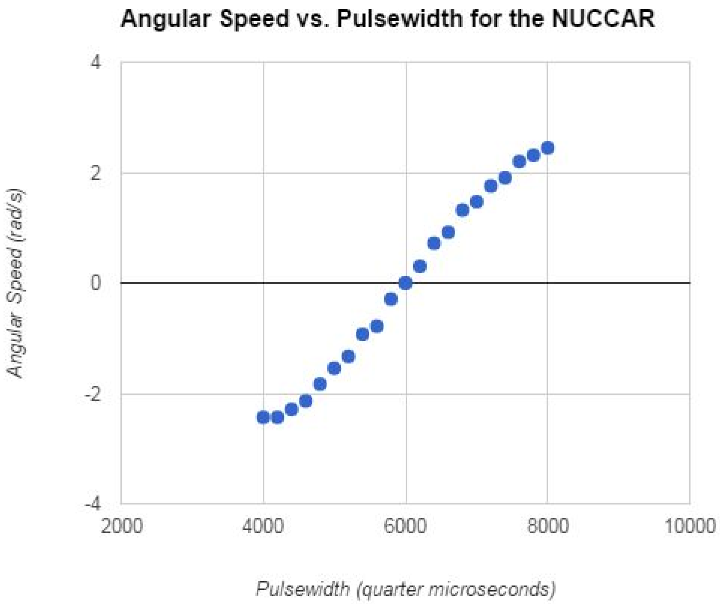
Calibrating the Vehicle Kinematics

- Motor and steering servo take in pulsewidths (PWMs)
- Need way to convert linear velocity (m/s) and angular velocity (rad/s) to PWM
    1. Command desired PWM until steady state
    2. Plot encoder data for linear, IMU data for angular
    3. Take mean at steady state

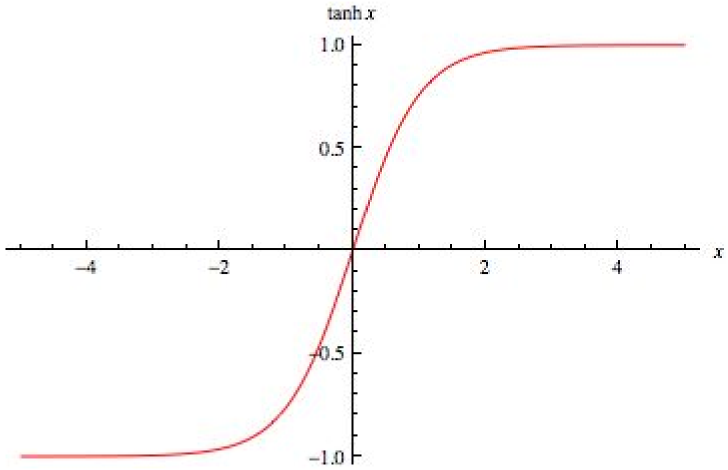




Found a nonlinear trend



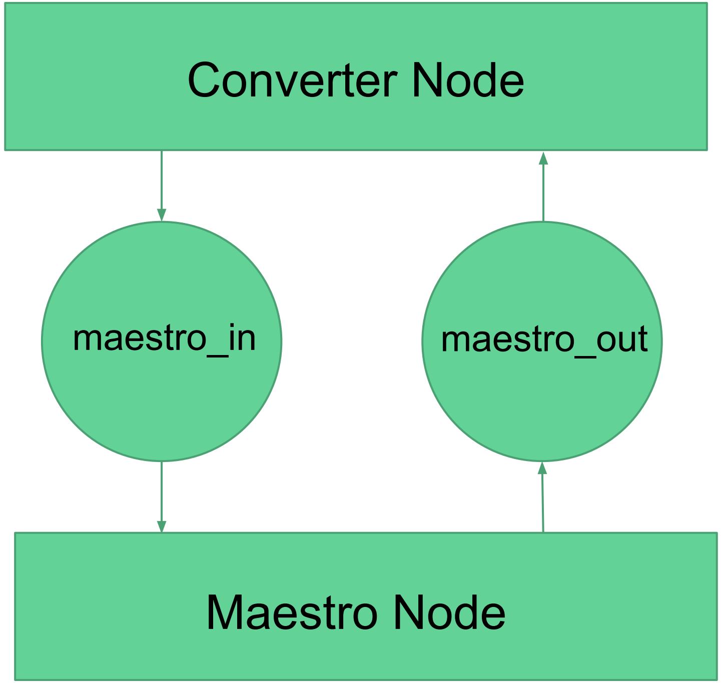
Our Empirical Data



Hyperbolic Tangent Function

Servo Controller (Maestro) and Converter Nodes

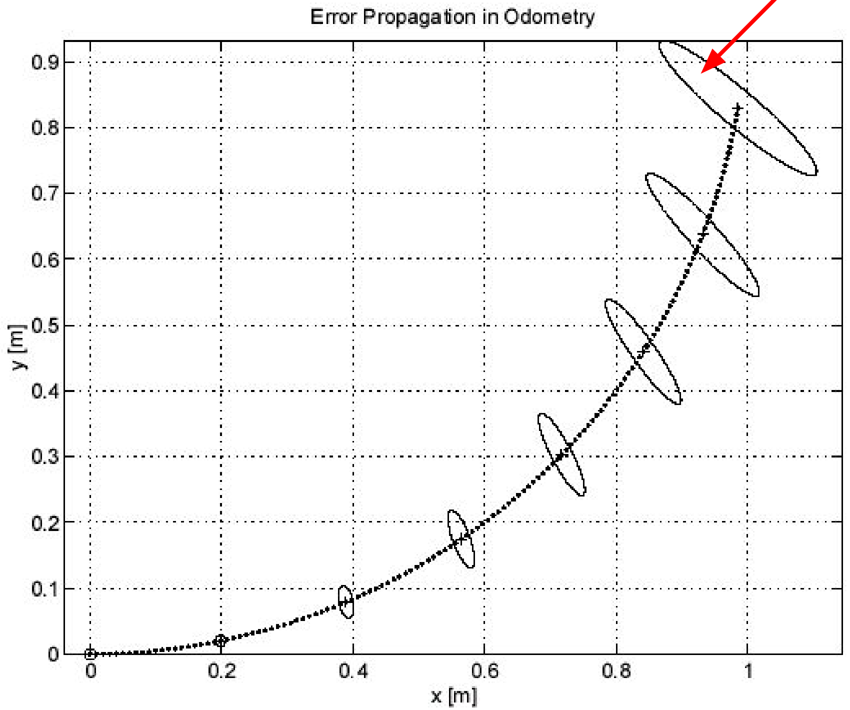
- Converter converts m/s and angular speeds into pulsewidths and sends to the maestro
- Maestro sends readings of current pulsewidth back out to converter and converter converts PWs to m/s or angular speed



Localization Using a Kalman Filter

- Where are we?
- Odometry error grows unbounded with time
- KF presents nice way to:
  - Keep track of the error
  - Bound the error
- The prediction-correction approach:
  - Make prediction of odometry (based on encoder and IMU)
  - Correct this estimate using a measurement (GPS northing and easting)

covariance ellipse



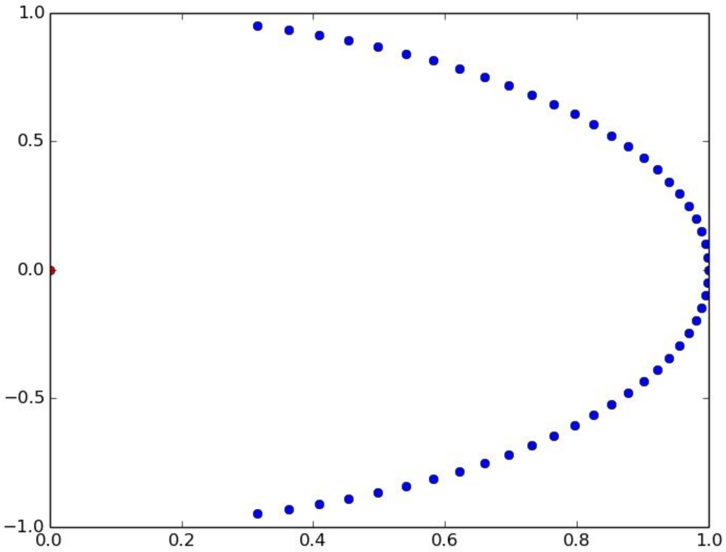
Unbounded error growth represented by error ellipses

Ready to go!

- Now the RoSCAR platform has all the hardware and software it needs for students to start writing interesting applications!
  - Application 1: Waypoint Navigation
  - Application 2: 3D Mapping using camera sensor to generate point clouds

Waypoint Navigation Node

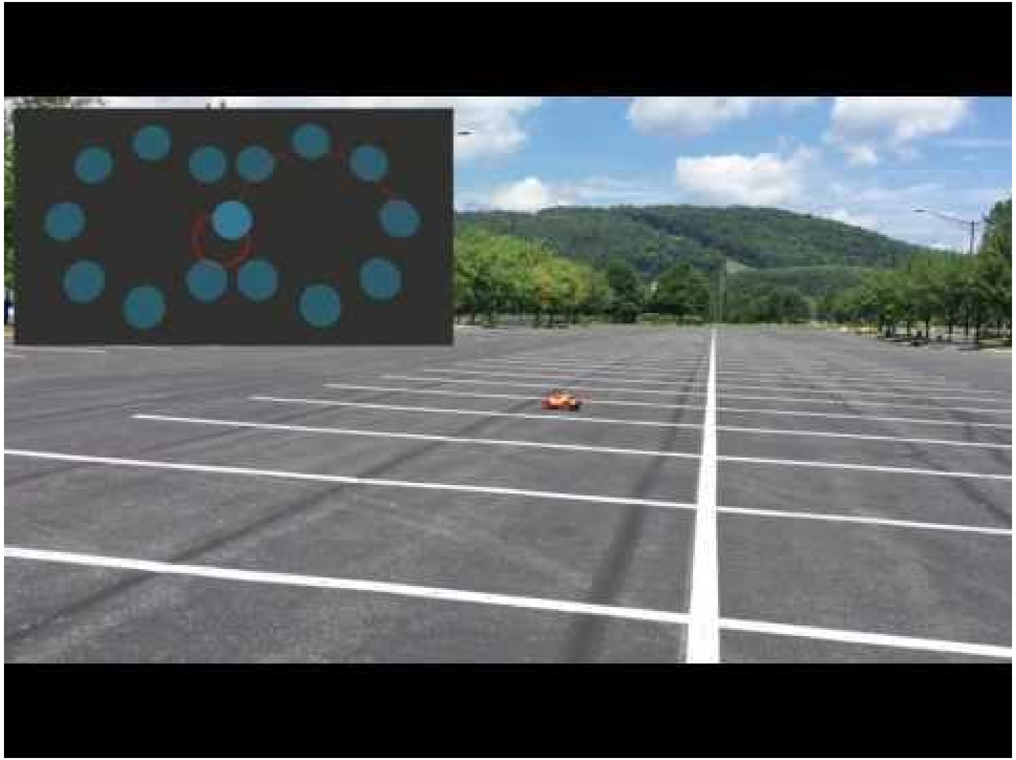
- Sample based algorithm
  - Keep linear velocity constant
  - Sample a range of angular velocities, project trajectories at some point in the future
  - Choose trajectory that gives lowest Euclidean distance from next waypoint and command that angular velocity



Proportional Controller

- Get a desired angular velocity from waypoint follower, ω_d
- Have a current angular velocity, ω_c
- Empirically determine some gain, α
- Then, combine these to proportionally control the car based on error:
    - $\omega_{commanded} = \omega_c + \alpha(\omega_d - \omega_c)$

GPS Waypoint Nav. Example



3D Mapping Example (Video)

Comparison of Exteroceptive Sensors

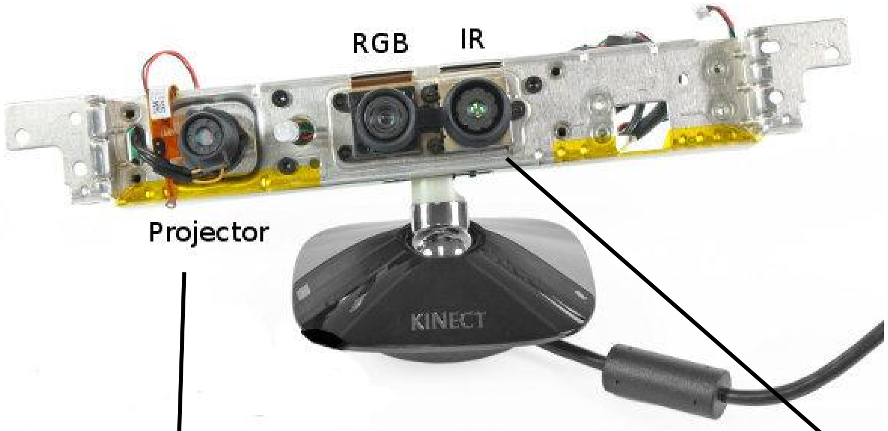
Two Methods of Point Cloud Generation

- Method 1: Structured Light
  - Asus Xtion RGB-D Camera
- Method 2: Stereo Vision
  - ZED Stereo Camera

Asus camera

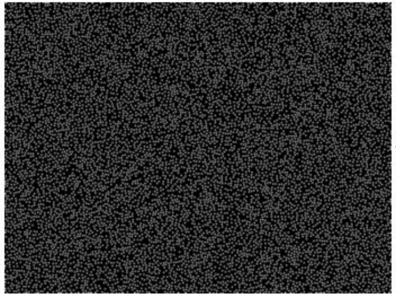
ZED Stereo Camera

Structured Light Point Cloud Generation

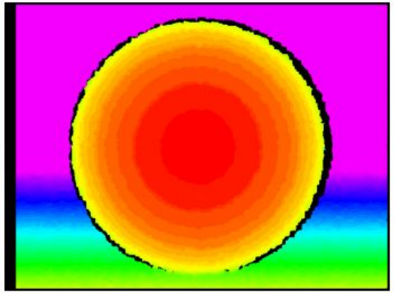


$$Z = \frac{b \cdot f}{\Delta}$$

Z = depth
b = stereo baseline
f = focal length
Δ = disparity

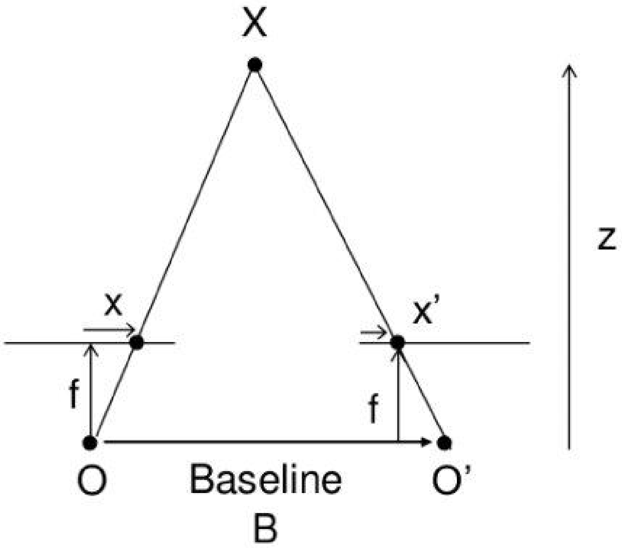

Projected Pattern


Received Pattern


Disparity Map

Stereo Vision Point Cloud Generation

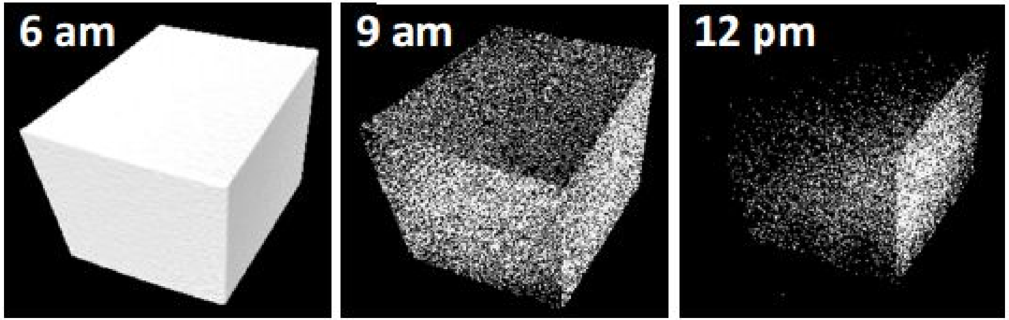
- Same depth calculation, but with different disparity
  - Disparity = Δ = x - x'



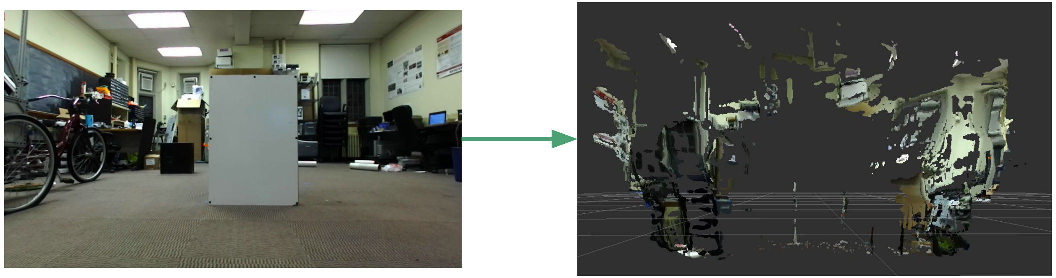
Example anaglyph: two stereo images superimposed and tinted in different colors

Limitations to Each Approach

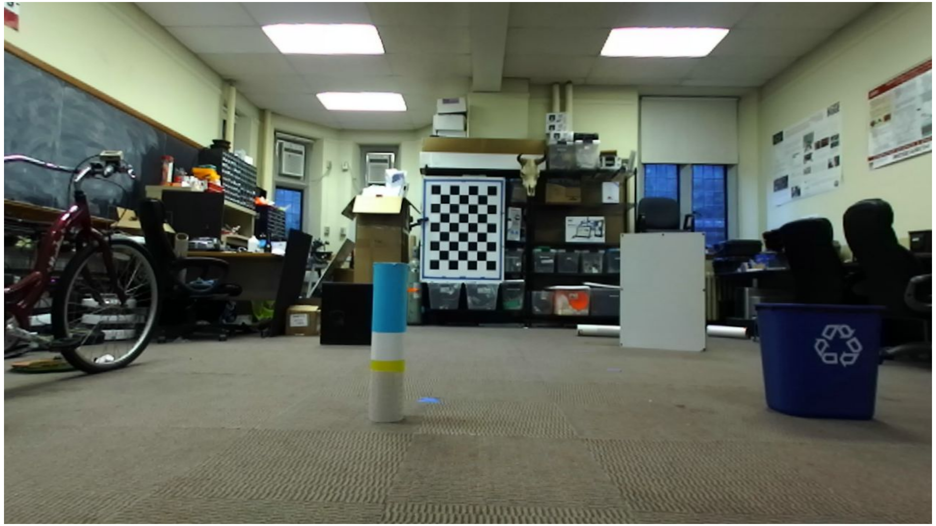
- Structured Light:
  - Sunlight is much stronger than the projector and interferes with the light patterns, **SL does not work well outside!**
- Stereo Vision:
  - Disparity is calculated by matching points in each image; thus, if the object has no texture **points cannot be matched and depth cannot be calculated**



http://www1.cs.columbia.edu/CAVE/publications/pdfs/Gupta_ICCV13a.pdf



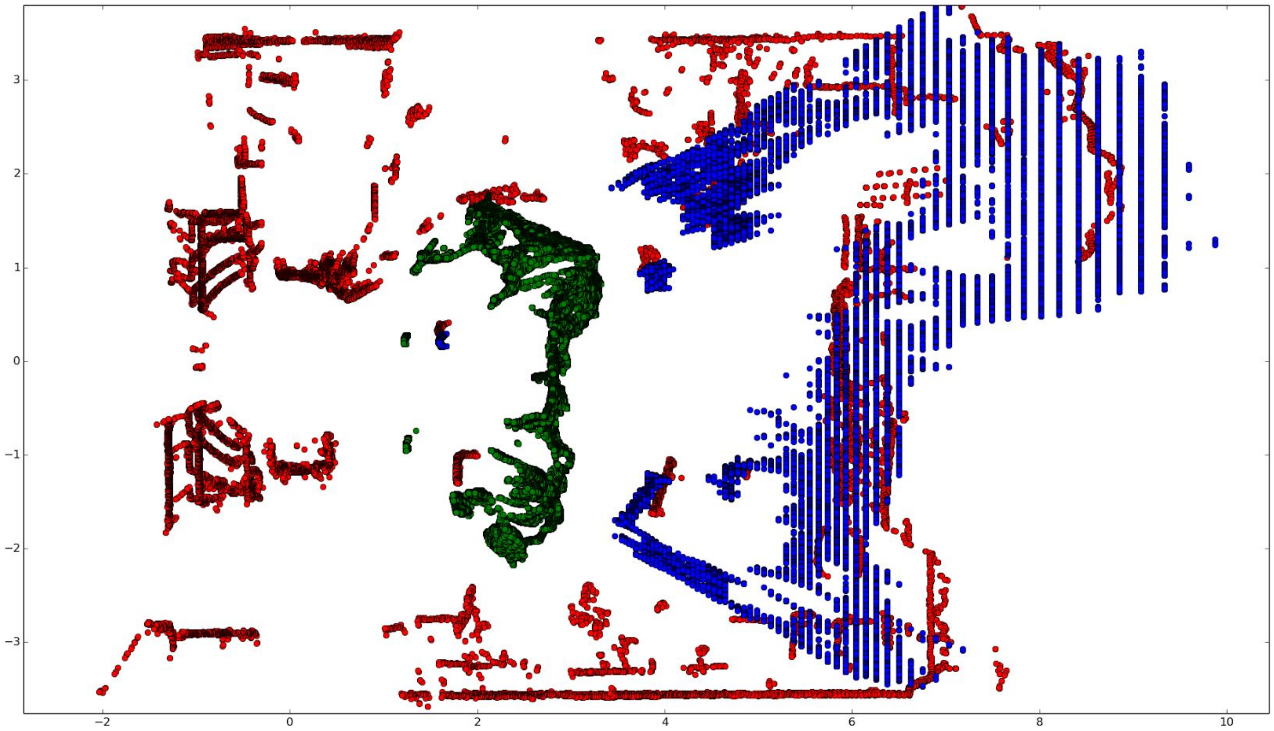
Comparing the Cameras



In lab object setup

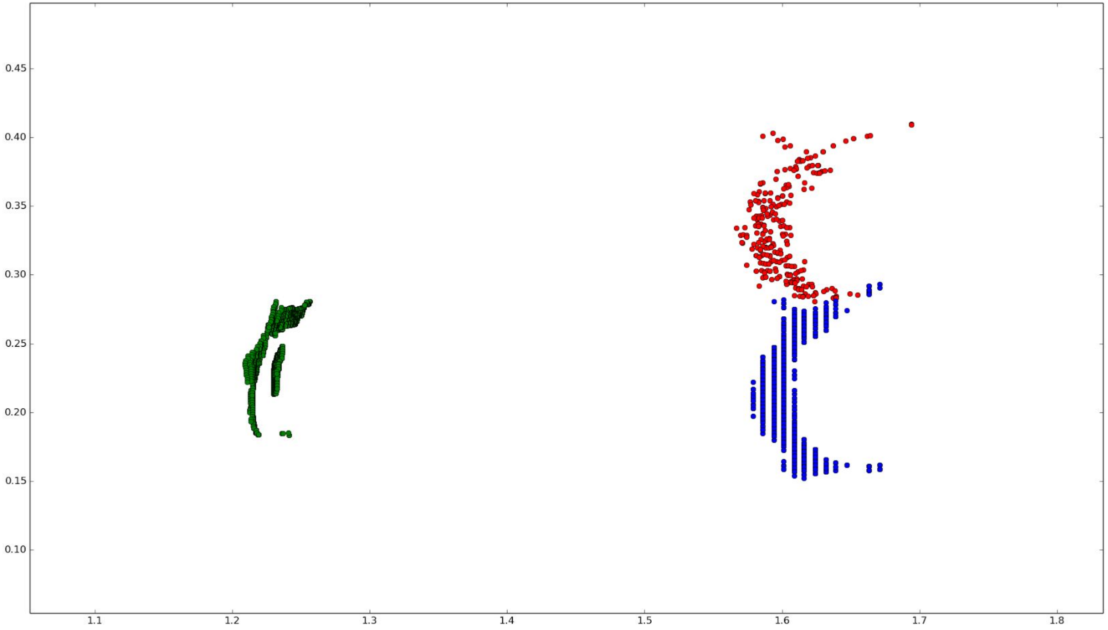
Velodyne 360 degree LiDAR sensor for ground truth

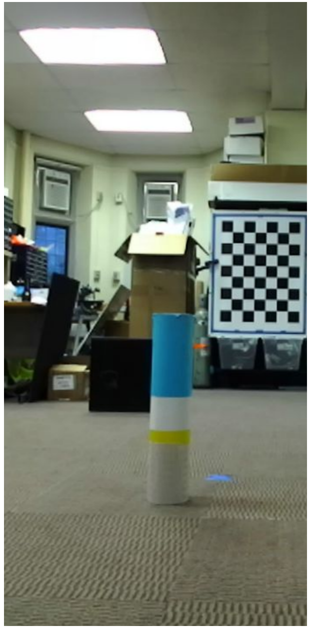
2D Projection of Point Cloud



Velodyne = red
ZED = green
Asus Xtion = blue

Cylinder Comparison





Conclusion & Future Extension

Improvements over RoSCAR

- Upgraded to ROS
  - Richer IPC
  - Extensive open source libraries
- Low cost with better tech specs
  - Nuccar: **$1,066.09**
  - Jetcar: **$1,351.55**
- IMU drastically improves odometry
- GPS bounds error and command predefined locations
- Indoor and outdoor computer vision

Sensor Comparison

	Asus Xtion RGB-D Camera	Stereo Labs ZED Camera
Metrics:		
Price	$167	$449
Max Range	~3.5 meters	~3 meters
Mean Depth Error	μ = 0.127m	μ = 0.367m
Working Conditions:		
Works in sunlight:		✓
Works in complete darkness:	✓	
Works on textured surfaces:	✓	✓
Works on untextured surfaces:	✓	

Future Work

- Servos with feedback
- Eliminate error from starting stationary
- Improve/reduce experimental calibration
- Use ZED for 3D mapping
- Proportional controller for linear velocity control
- Bug in GPS element of kalman filter

Acknowledgements

Special thanks to :

John Spletzer

Dylan Schwesinger

Lehigh CSE Department

National Science Foundation (NSF)

Works Cited

"Pulse Width Modulation Used for Motor Control." *Basic Electronics Tutorials Pulse Width Modulation Comments*. N.p., 10 Sept. 2013. Web. 27 July 2016.

"Wiki." *Jade/Installation/Ubuntu*. N.p., n.d. Web. 27 July 2016. <http://wiki.ros.org/jade/Installation/Ubuntu>.

"OS - Linux." - *Phidgets Support*. N.p., n.d. Web. 27 July 2016. <http://www.phidgets.com/docs/OS_-_Linux>.

"Ubuntu Documentation." *WifiDocs/Adhoc*. N.p., n.d. Web. 27 July 2016. <https://help.ubuntu.com/community/WifiDocs/Adhoc>.

"Main Page." *Eigen*. N.p., n.d. Web. 27 July 2016. <http://eigen.tuxfamily.org/index.php?title=Main_Page#Download>.

"How to Install CUDA on NVIDIA Jetson TX1 - Slothparadise." *Slothparadise*. N.p., 24 Nov. 2015. Web. 27 July 2016. <http://www.slothparadise.com/how-to-install-cuda-on-nvidia-jetson-tx1/>.

"Prebuilt Binaries for Linux." - *Point Cloud Library (PCL)*. N.p., n.d. Web. 27 July 2016. <http://pointclouds.org/downloads/linux.html>.

"Stereolabs/zed-ros-wrapper." *GitHub*. N.p., n.d. Web. 27 July 2016. <https://github.com/stereolabs/zed-ros-wrapper>.

Questions?

Testing at Goodman Athletic Complex