CCCC: Traversable Terrain Detection

Rahul Amara, Josh Hejna, Anish Singhani, Mokshith Voodarla

PROJECT BACKGROUND

Autonomous Navigation for the *Masses*

Consumer robots *need* to be be **smart**

Consumer robots still *need* to be **affordable**...

...which means reliable sensors like **3D LiDAR** are *impractical*

Goals

1. Detect untraditional obstacles (ie. glass, railing) which LiDAR can't see
2. Stick to affordable components
   a. i.e. replace 3D LiDAR with camera+2D LiDAR

CLEARPATH JACKAL

with Jetson TX2

Upgraded with:

- RPLidar A1 (low-cost 2D LiDAR)
  - ~$100
- Wide-Angle USB Camera
  - ~$100

Also includes:

- BNO055 IMU (Gyro)
- Wheel Encoders

PIPELINE

From Image to Useable Navigation Input

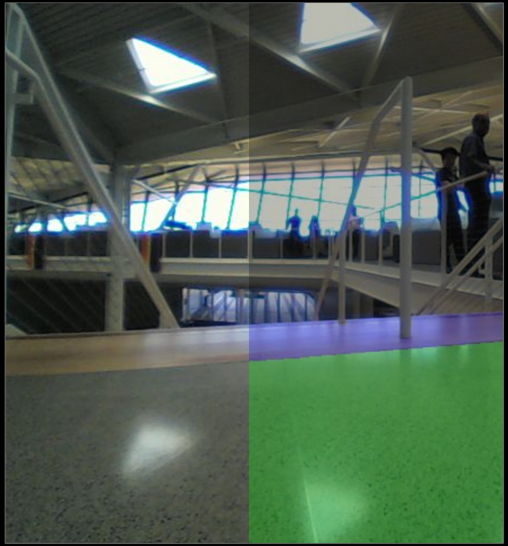
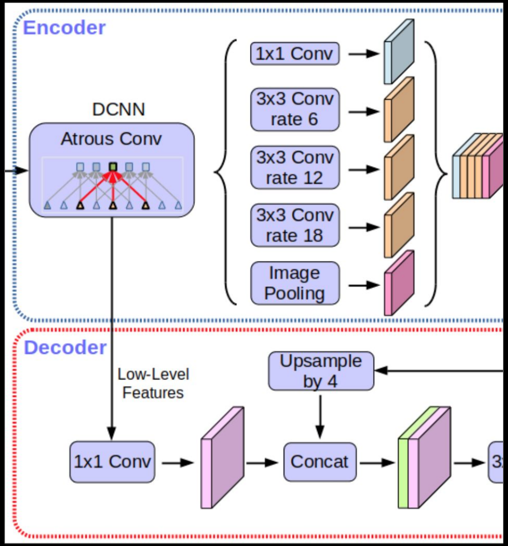


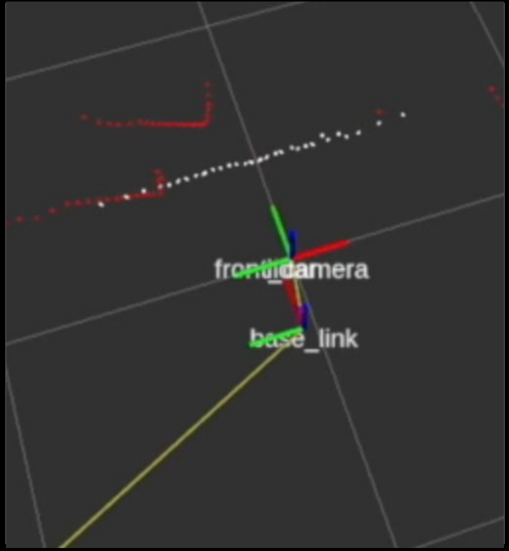
Image Input

3600 augmented images w/ labels



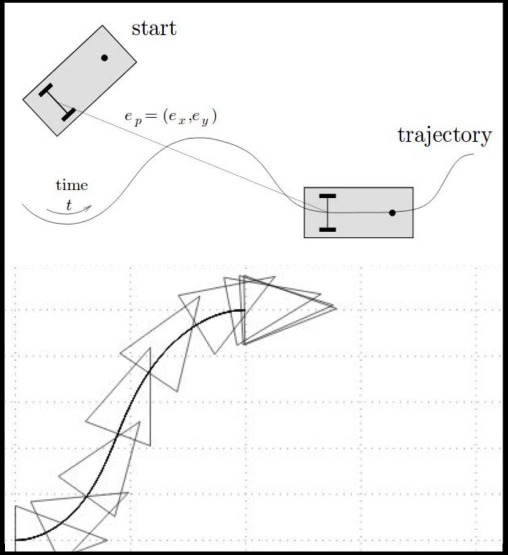
Segmentation Model

DeepLab V3 w/ ResNet 50 V2



LaserScan Generation

ROS-compatible obstacles



Navigation Stack

Custom stack based on NLRT

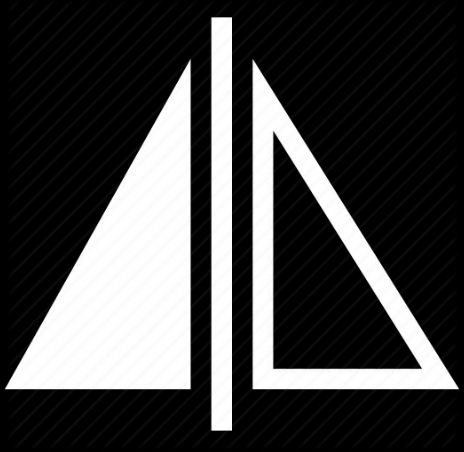
DATA COLLECTION AND AUGMENTATION

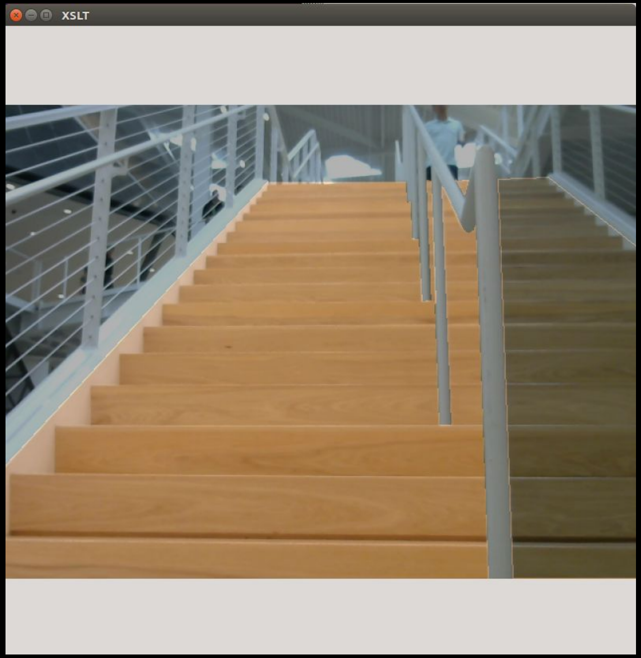
480 Images collected

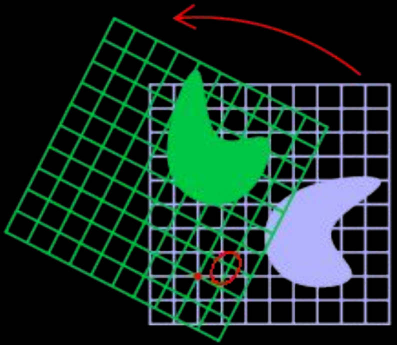
5 Classes

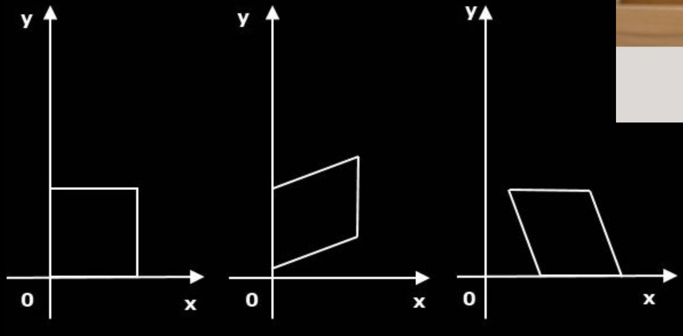
Augmentation to 3600+ images
- Rotations
- Shears
- Horizontal Flips

Labeled with custom SXLT Labeling Tool









Original object

After y shear

After x shear

SEMANTIC SEGMENTATION MODEL

Semantic Segmentation requires more compute than image classification, object detection

Goal is to separate all unsafe and safe space in front of robot

Constraints:

1. Relatively lighter weight model
2. Can perform reasonably accurate segmentation
3. Runs at a decent framerate

TESTED MODELS

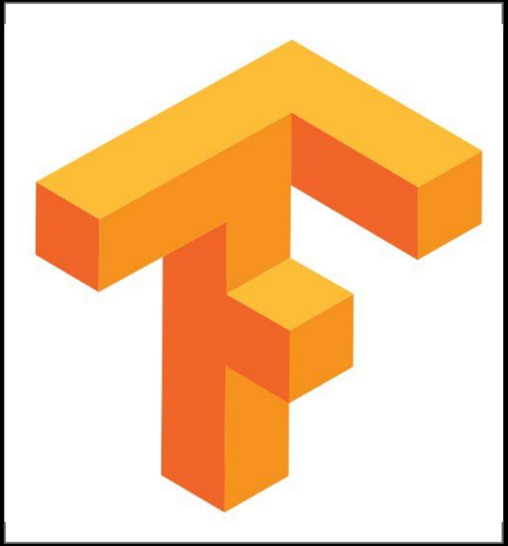
	mIOU	Isolated FPS	Implementation
Fully-Convolutional AlexNet	0.65	~1 (with TensorRT)	Caffe
E-Net	0.84	~4.5 (no TensorRT)	TensorFlow
CV + Random Forest Classifier	0.84*	~12 (no TensorRT)	OpenCV
DeepLab V3 (ResNet-50 v2 Feature Extractor)	0.94	~4.5 (no TensorRT)	TensorFlow

* mIOU for one class, other models were multiclass

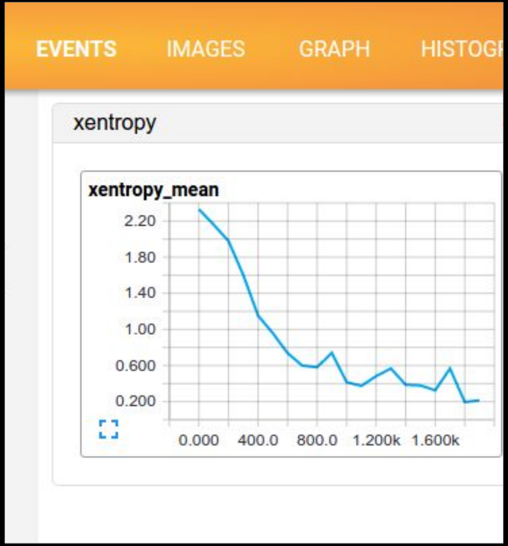
DEEPLAB V3

TensorFlow Workflow

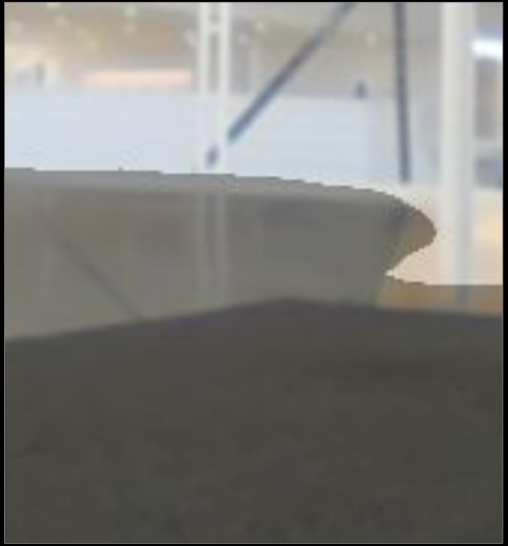
Data Collection saved to TFRecords data format



TensorFlow Model Loading and Implementation



TensorBoard Training Visualization



Final Model Inferencing on Jetson TX2 GPU

MAIN PROBLEM FACED

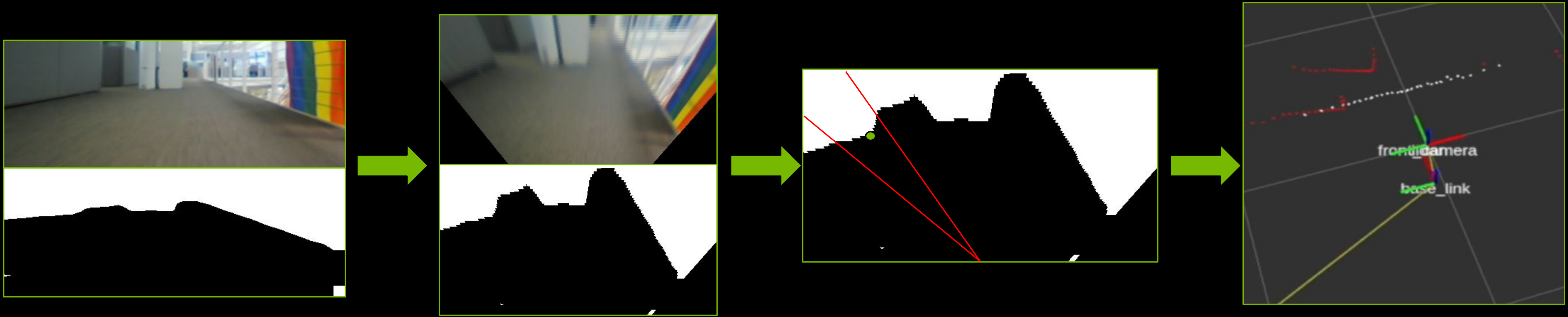
During Implementation on Jetson TX2

Varied inference results on Jetson CPU/GPU vs regular CPU or GPU

- Downgrading to CUDA 8 and TensorFlow 1.3 fixes this

LASERSCAN GENERATION

Inference Segmentation to Laser Scan (for SLAM)



1. Segmentation Data

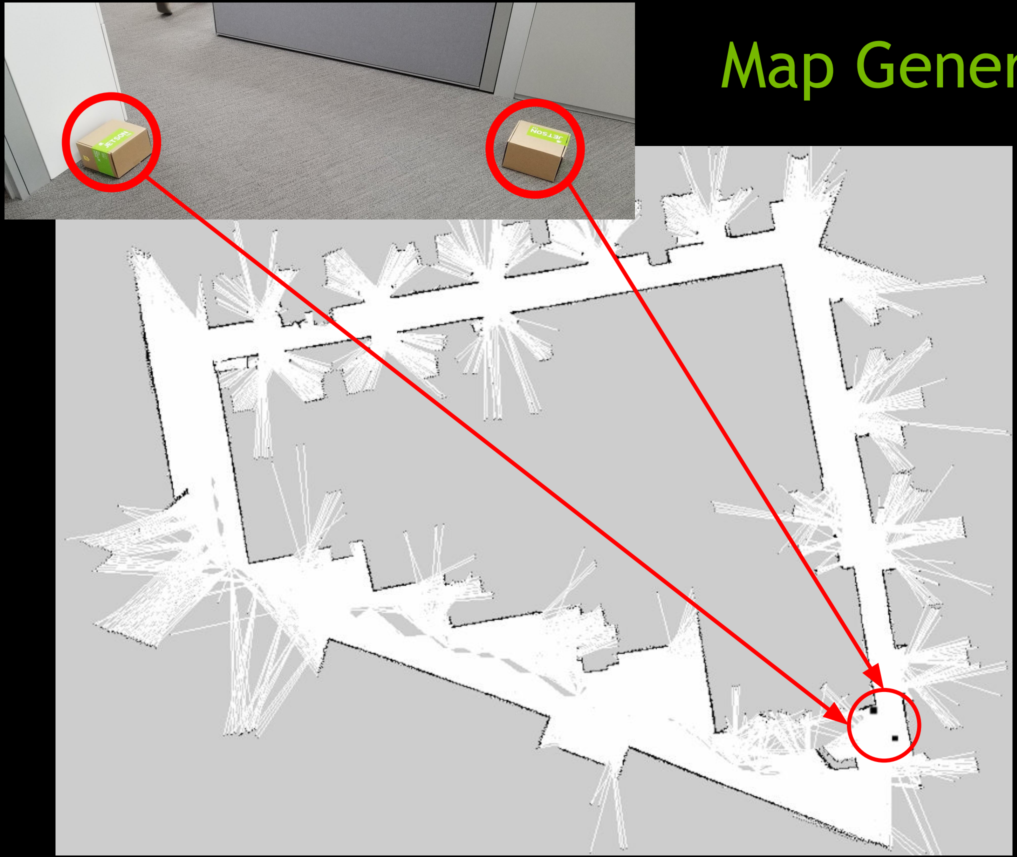
2. Top Down Transform

3. Sector Based Point Extraction

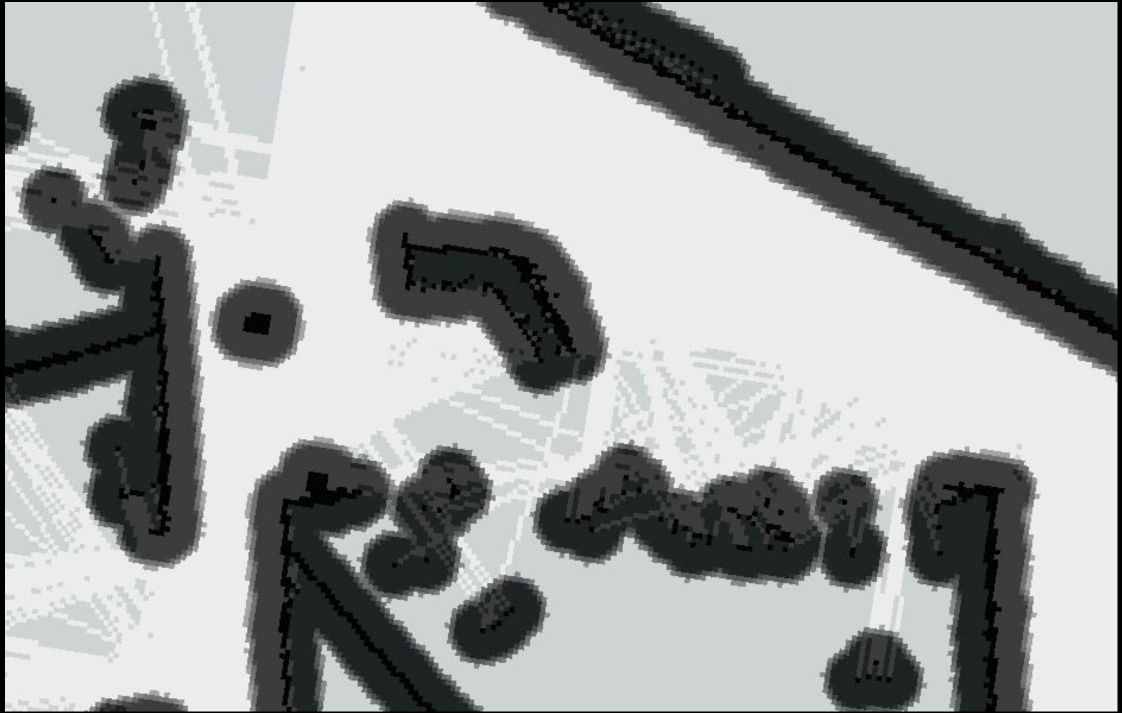
4. Formatting into Laser Scan

NAVIGATION STACK

Map Generation and Usage



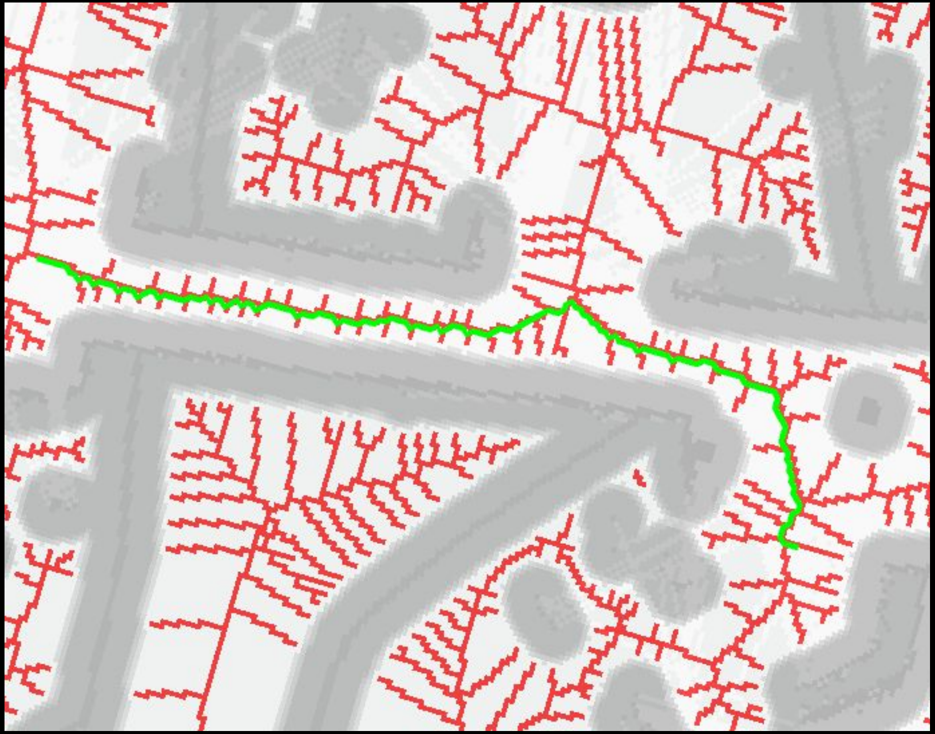
SLAM Mapping
(Includes semantic segmentation)



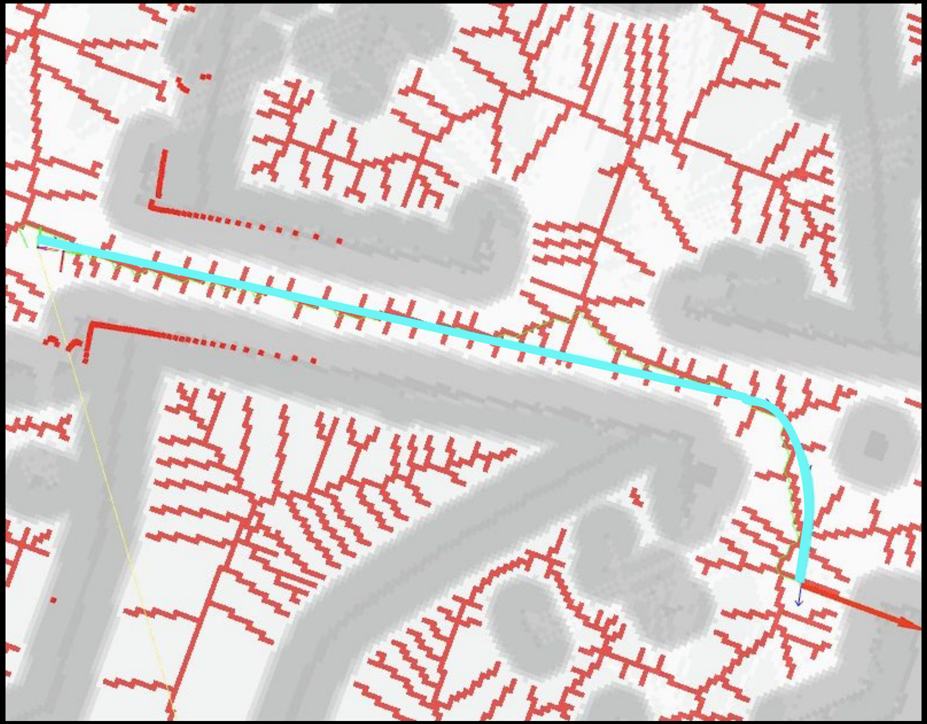
Configuration Space

NAVIGATION STACK

Path Planning



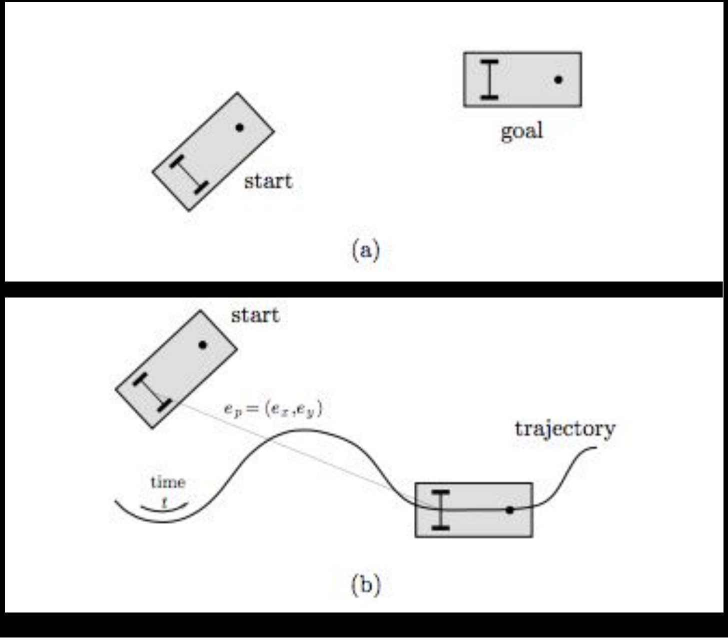
Voronoi Path Planning

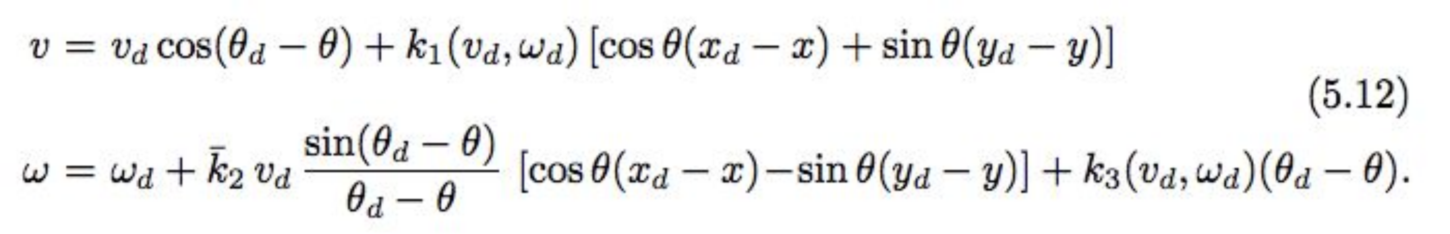


Cubic Hermite Spline Fitting

NAVIGATION STACK

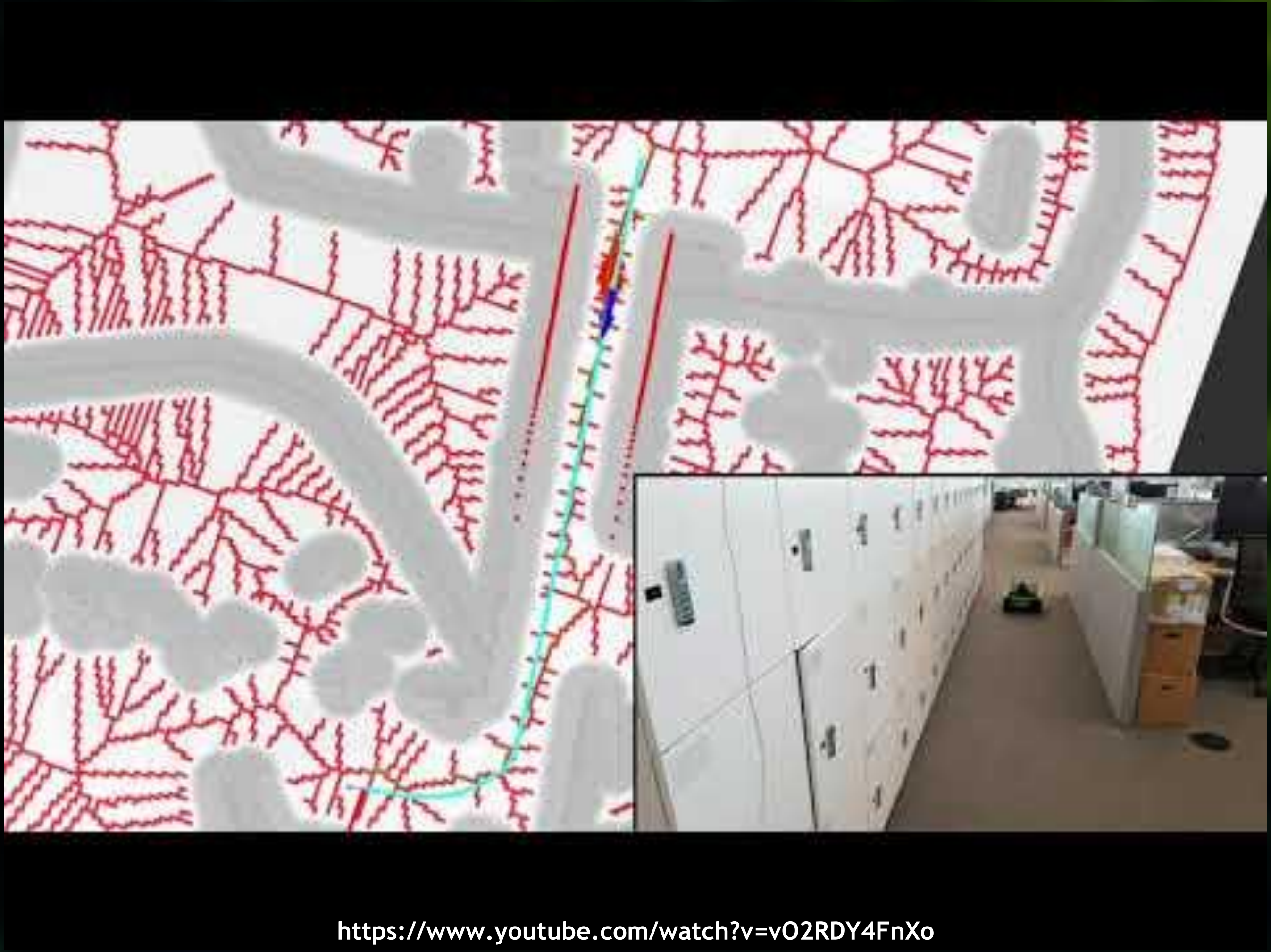
Path Following (Nonlinear Reference Tracking)





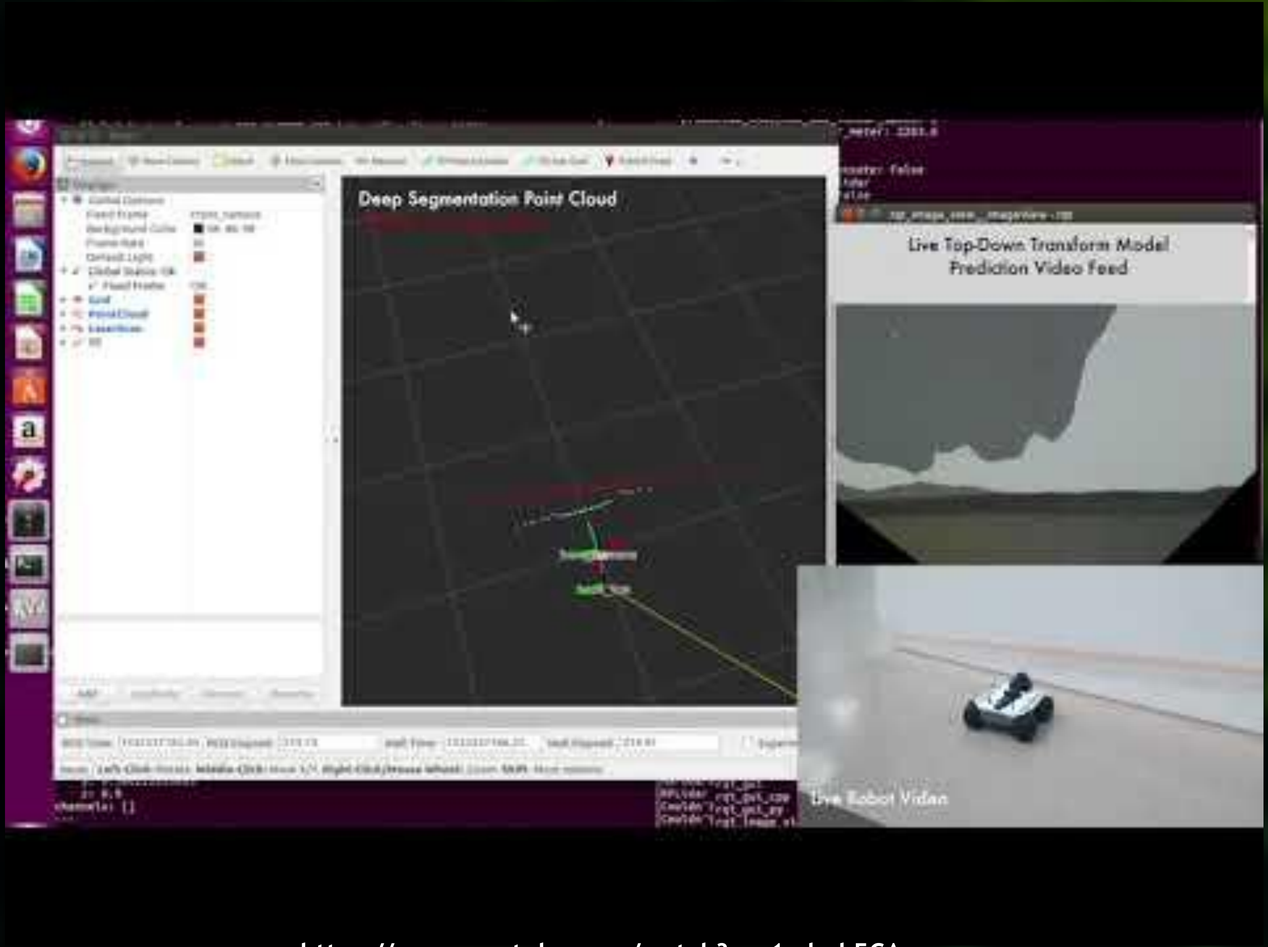
$$v = v_d \cos(\theta_d - \theta) + k_1(v_d, \omega_d)\left[\cos\theta(x_d - x) + \sin\theta(y_d - y)\right]$$

$$\omega = \omega_d + \bar{k}_2\, v_d \,\frac{\sin(\theta_d - \theta)}{\theta_d - \theta}\left[\cos\theta(x_d - x) - \sin\theta(y_d - y)\right] + k_3(v_d, \omega_d)(\theta_d - \theta). \tag{5.12}$$



https://www.youtube.com/watch?v=vO2RDY4FnXo

Navigation Stack Demo


Deep Segmentation Point Cloud
Live Top-Down Transform Model Prediction Video Feed
Live Robot Video

https://www.youtube.com/watch?v=s1mhek5CAnw

Segmentation to Pointcloud Demo

NVIDIA®