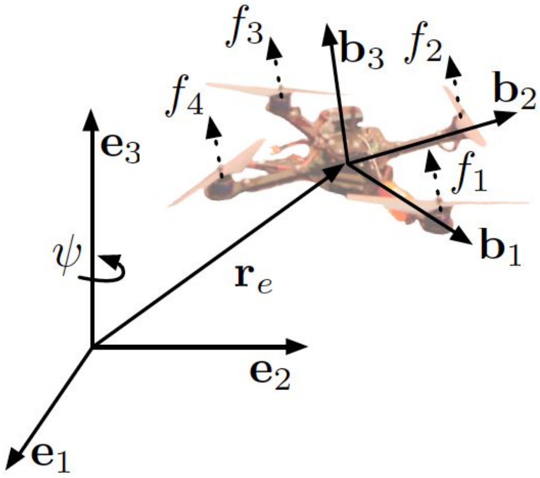
Quadrotor State Estimation and Obstacle Detection

Robot Autonomy Project

Cole, Job, Erik, Rohan

I. Dynamics
II. Differential Flatness
III. Planning
IV. Control Architecture
V. State Estimation (EKF)
VI. Sensors
VII. SLAM (RTAB Map)
VIII. Obstacle Detection
IX. Video

Quadrotor Dynamics



Control inputs:

$$[f\ \mathrm{M}]^T, f \in \mathbb{R}, \mathrm{M} \in \mathbb{R}^3$$

State:

$$[x\ v\ R\ \Omega]^T$$

Dynamics:

$$\dot{x} = v$$
$$m\dot{v} = mge_3 - fRe_3$$
$$\dot{R} = R\widehat{\Omega}$$
$$J\dot{\Omega} + \Omega \times J\Omega = \mathrm{M}$$

Differential Flatness

Dynamics:

$$\dot{x} = v$$
$$m\dot{v} = mge_3 - fRe_3$$
$$\dot{R} = R\widehat{\Omega}$$
$$J\dot{\Omega} + \Omega \times J\Omega = \mathrm{M}$$

Pick outputs: $y = y(x, u, \dot{u}, \ldots, u^{(p)})$

Such that:

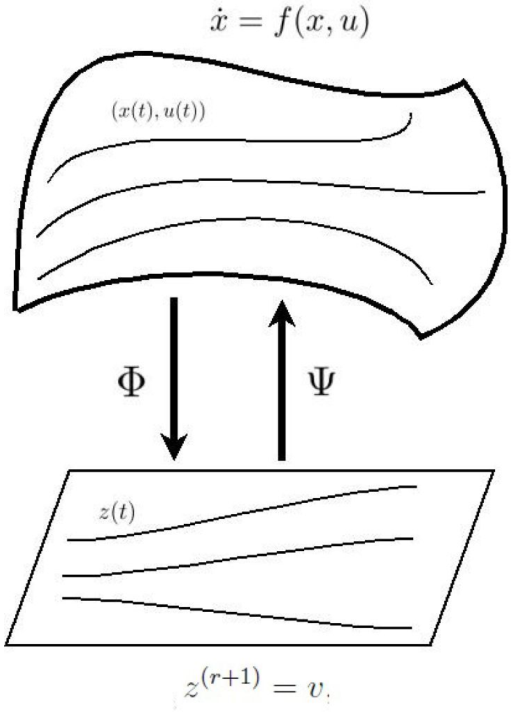
$$\begin{aligned} x &= x(y, \dot{y}, \ldots, y^{(q)}) \\ u &= u(y, \dot{y}, \ldots, y^{(q)}). \end{aligned}$$

Any 4 of the following 6 can serve as flat outputs:

X
Y
Z
Phi
Theta
Psi

Murray, Richard M., Muruhan Rathinam, and Willem Sluis. "Differential flatness of mechanical control systems: A catalog of prototype systems." *ASME international mechanical engineering congress and exposition*. 1995.

Differential Flatness



Any 4 of the following 6 can serve as flat outputs:

X
Y
Z
Phi
Theta
Psi

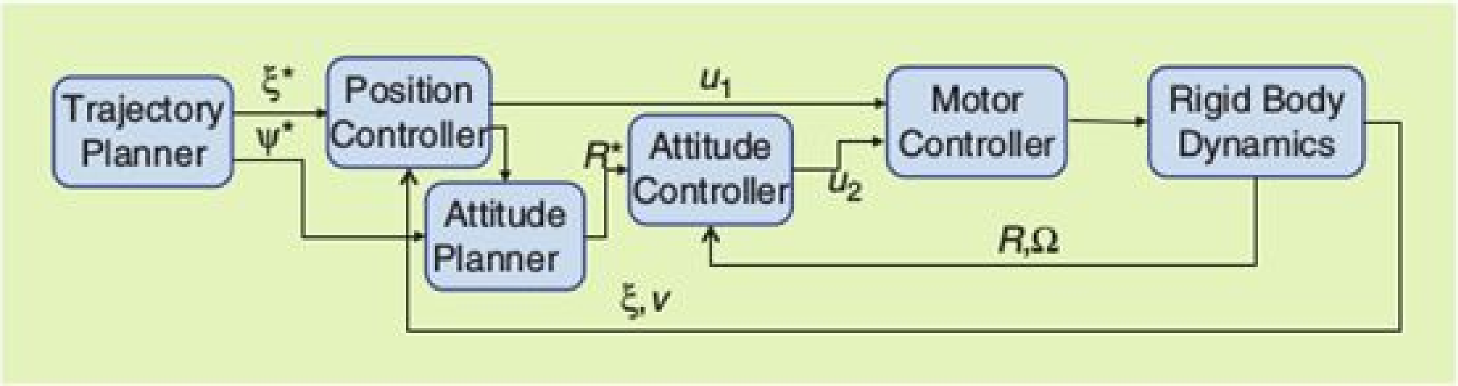
Trajectory Planning:

$$\min \int_{t_0}^{t_m} \mu_r \left\| \frac{d^{k_r} \mathbf{r_T}}{dt^{k_r}} \right\|^2 + \mu_\psi \frac{d^{k_\psi} \psi_\mathbf{T}}{dt^{k_\psi}}^2 dt$$

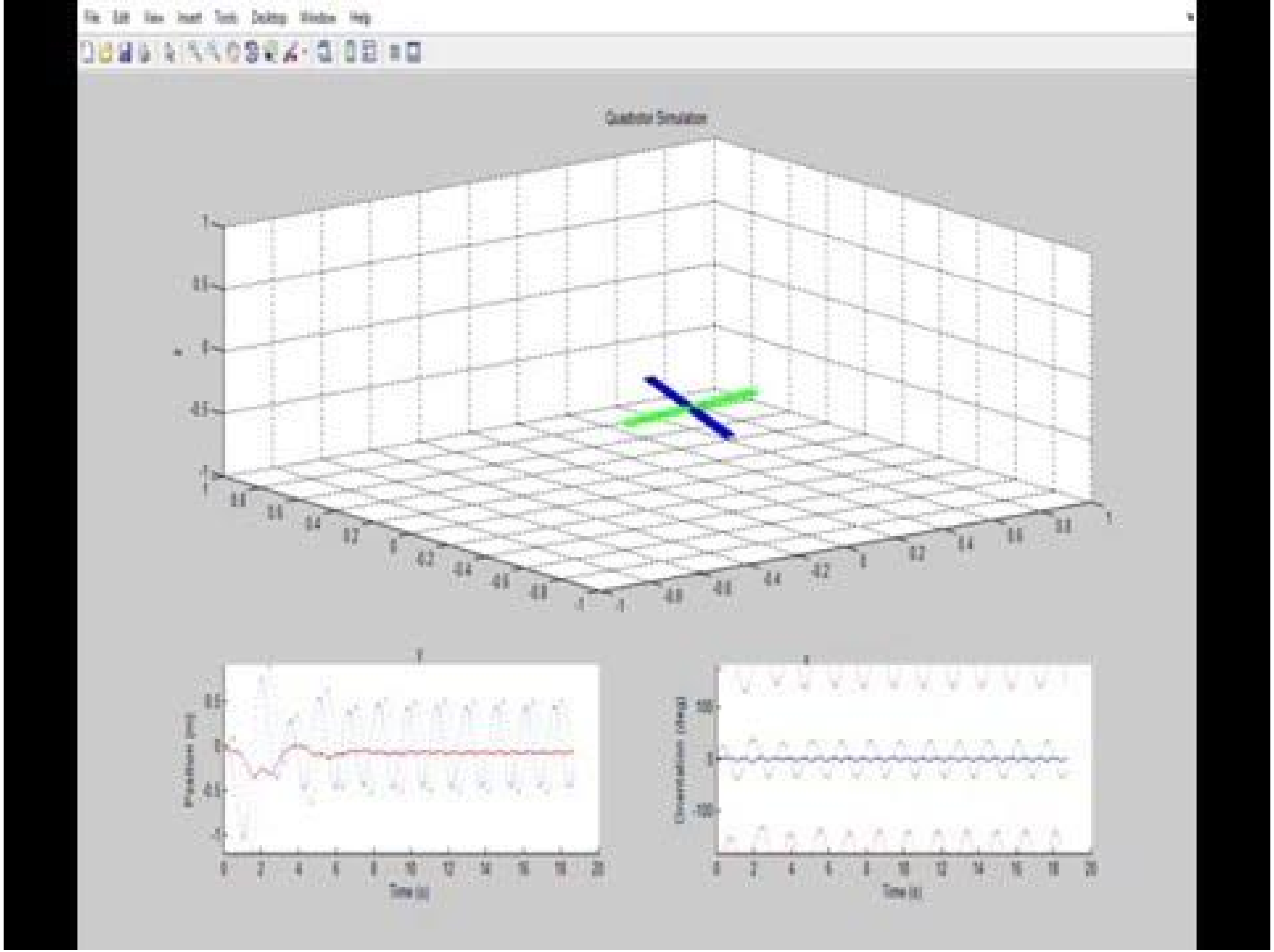
kr = 4, kψ = 2

Mellinger, Daniel, Nathan Michael, and Vijay Kumar. "Trajectory generation and control for precise aggressive maneuvers with quadrotors." *The International Journal of Robotics Research* (2012): 0278364911434236.

CONTROL ARCHITECTURE



Mahony, Robert, Vijay Kumar, and Peter Corke. "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor." IEEE Robotics & amp amp Automation Magazine 19 (2012): 20-32.

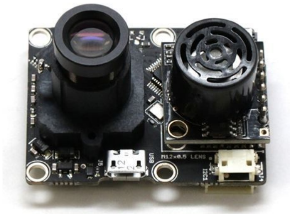

File Edit View Insert Tools Desktop Window Help
Quadrotor Simulation
Position (m)
Time (s)
Orientation (deg)
Time (s)

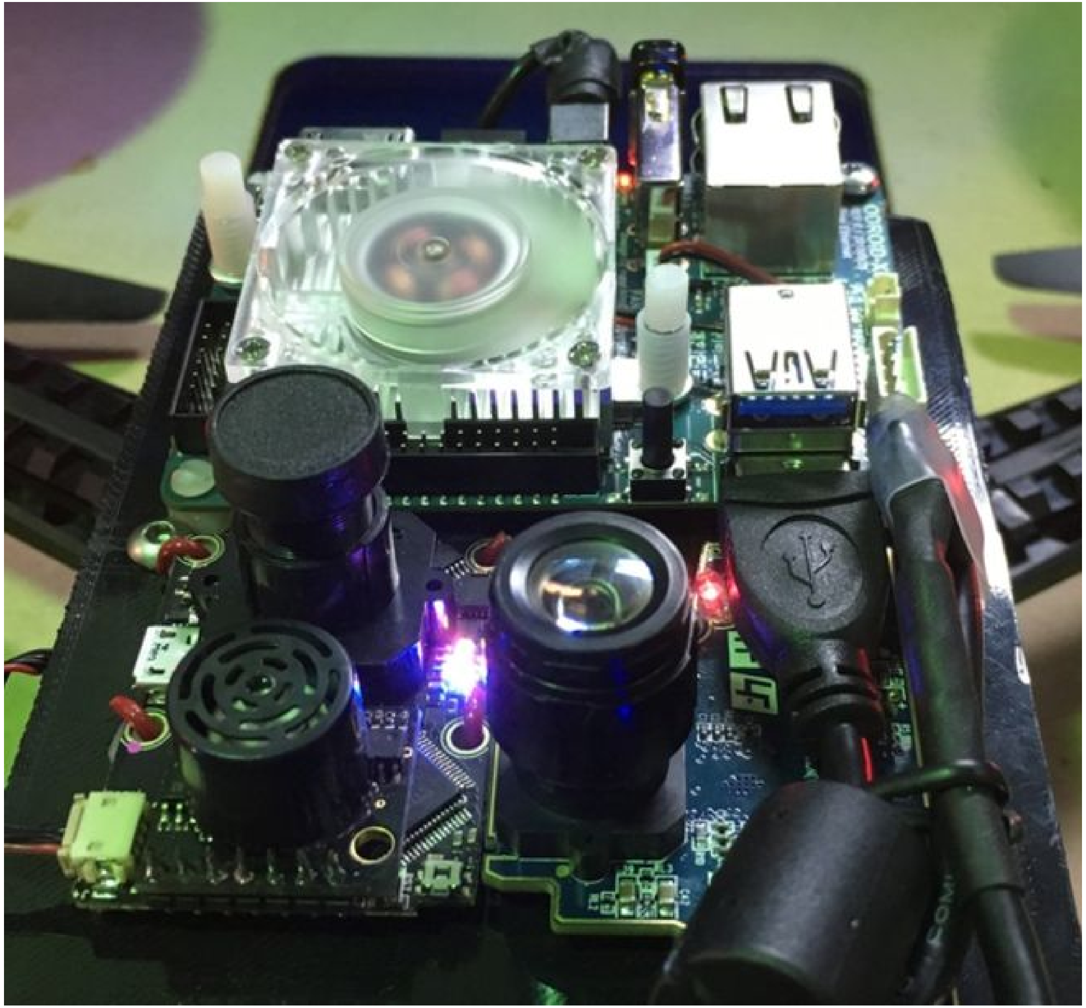
Sensors

Camera

IMU

Height Sensor





Extended Kalman Filter

Kalman Filtering: Computations

Notation:

A_t : *Motion Model*

B_t : *Control Input Model*

μ_t : *State Mean*

Σ_t : *State Variance*

Q_t : *Motion Model Noise*

C_t : *Observation Model*

R : *Observation Noise*

K_t : *Kalman Gain*

$z_t - C_t\bar{\mu}_t$: *Innovation*

$$\text{KalmanFilter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$$

Prediction

$$\bar{\mu}_t = A_t \mu_{t-1} + B u_t$$

$$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^\top + Q_t$$

Gain

$$K_t = \bar{\Sigma}_t C_t^\top (C_t \bar{\Sigma}_t C_t^\top + R\ )^{-1}$$

Update

$$\mu_t = \bar{\mu}_t + K_t(z_t - C_t \bar{\mu}_t)$$

$$\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$$

Slide courtesy Kris Kitani

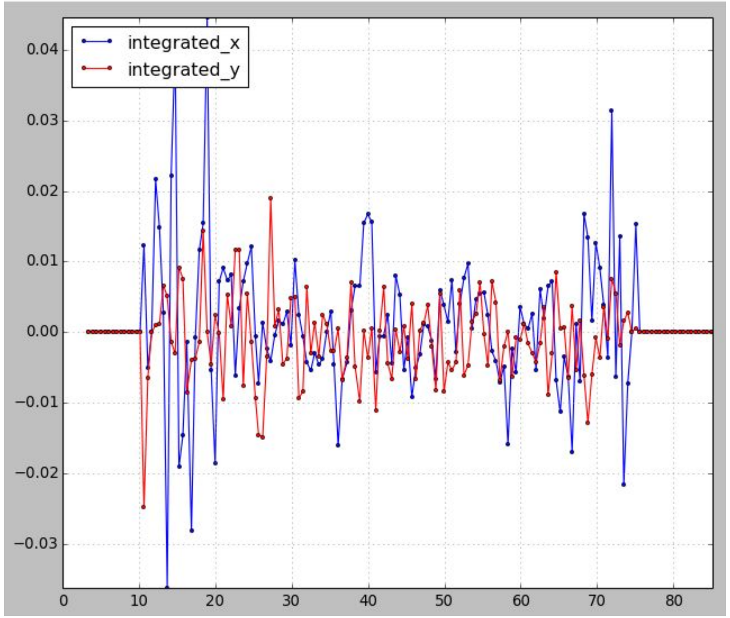
State Estimation with Optical Flow

$$v_x = \frac{T_z x - T_x \mathrm{f}}{Z} - \omega_y \mathrm{f} + \omega_z y + \frac{\omega_x xy - \omega_y x^2}{\mathrm{f}}$$

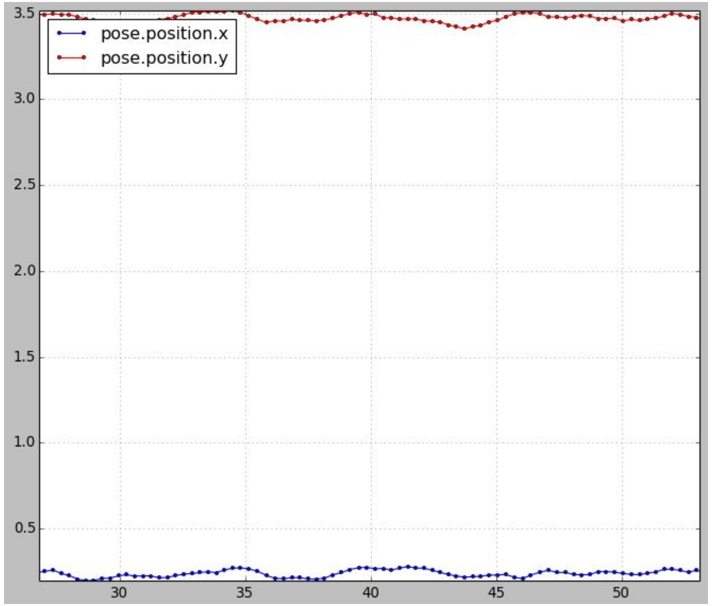
$$v_y = \frac{T_z y - T_y \mathrm{f}}{Z} + \omega_x \mathrm{f} - \omega_z x + \frac{\omega_x y^2 - \omega_y xy}{\mathrm{f}}$$

$$\frac{\mathbf{flow}}{\Delta \mathrm{time}} = \mathbf{v} = \mathrm{f}\frac{\mathrm{Z}\mathbf{V} - V_z \mathbf{P}}{\mathrm{Z}^2}$$

State Estimation with Optical Flow

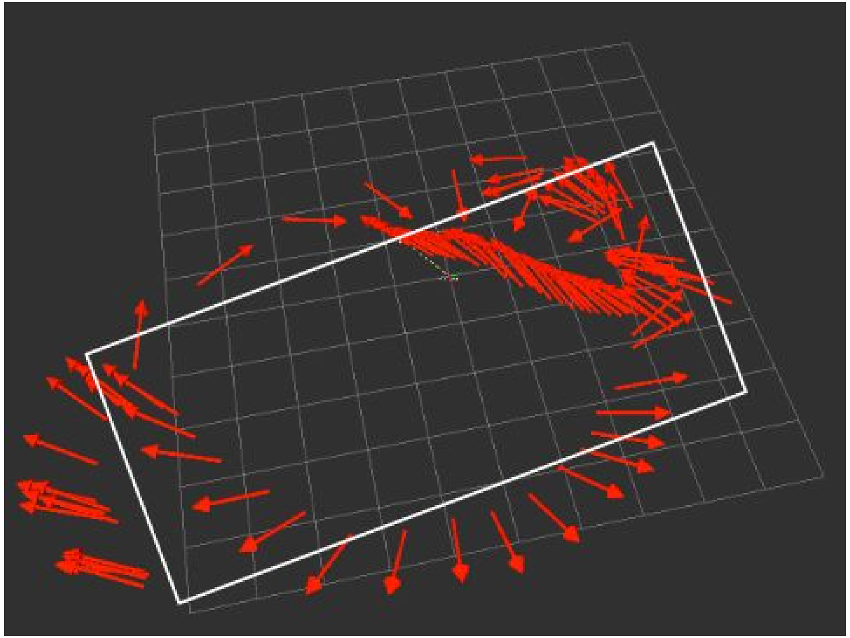


Velocity Updates from Optical Flow Camera

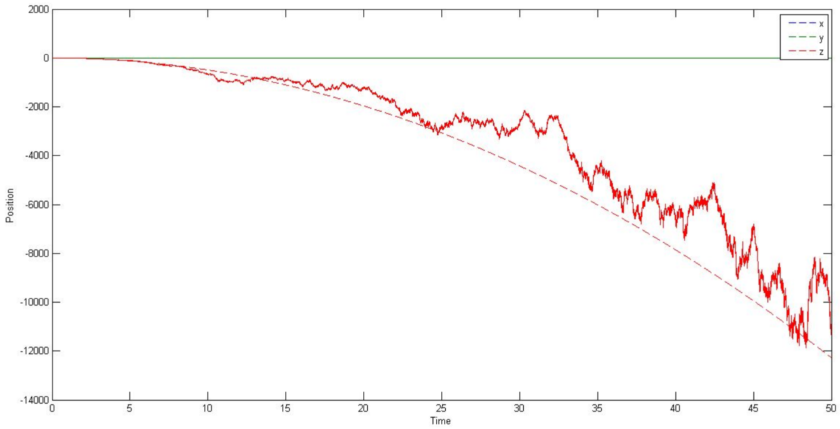


Position Updates from EKF

State Estimation with Optical Flow



Odometry Readings

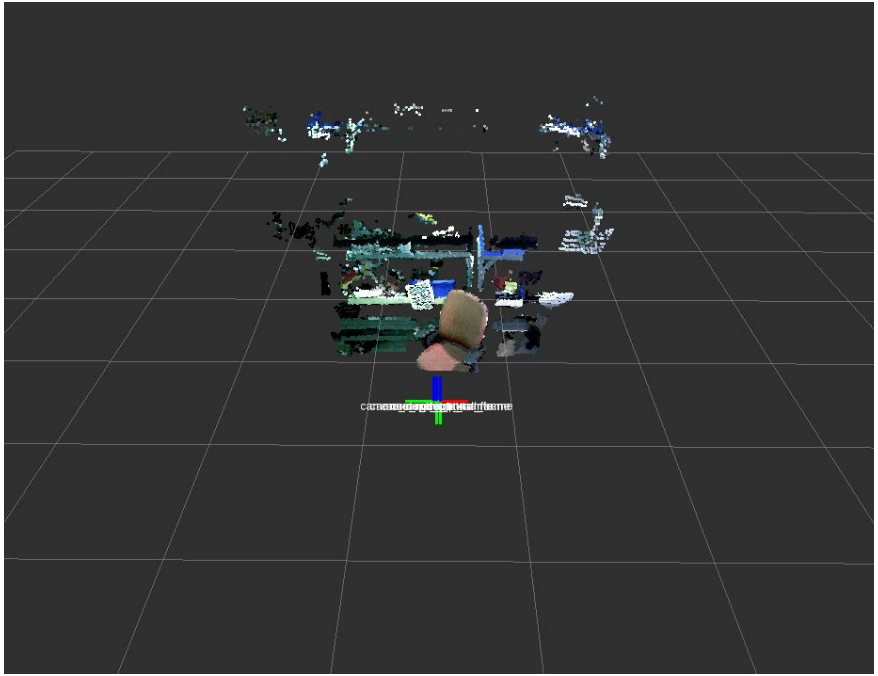


Linear Drift with Time in Simulation

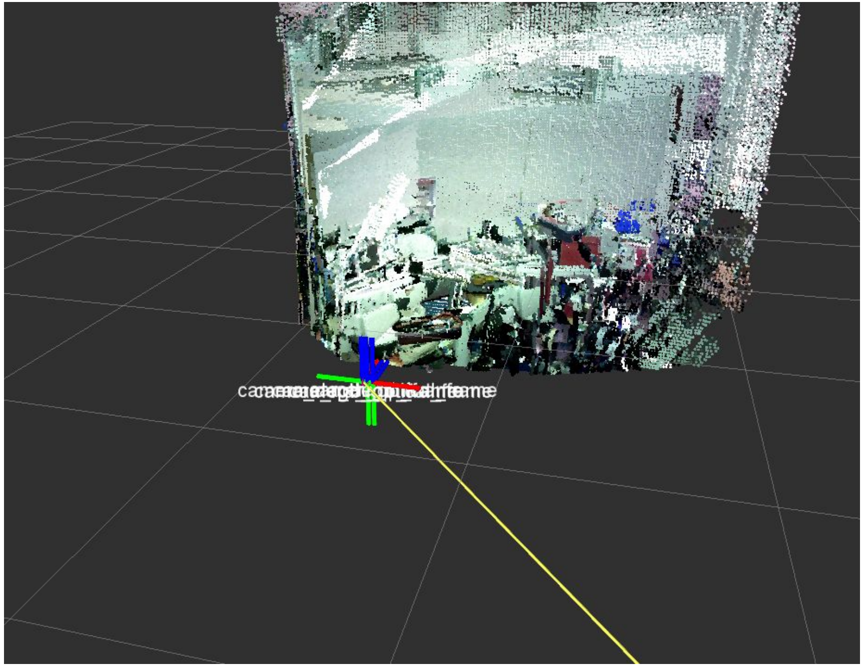
RTAB-Map

- Graph and Node based System
- Gathers RGB and Depth information
- OpenNI handles point clouds
- Uses visual words to detect loop closures

RGB-D SLAM

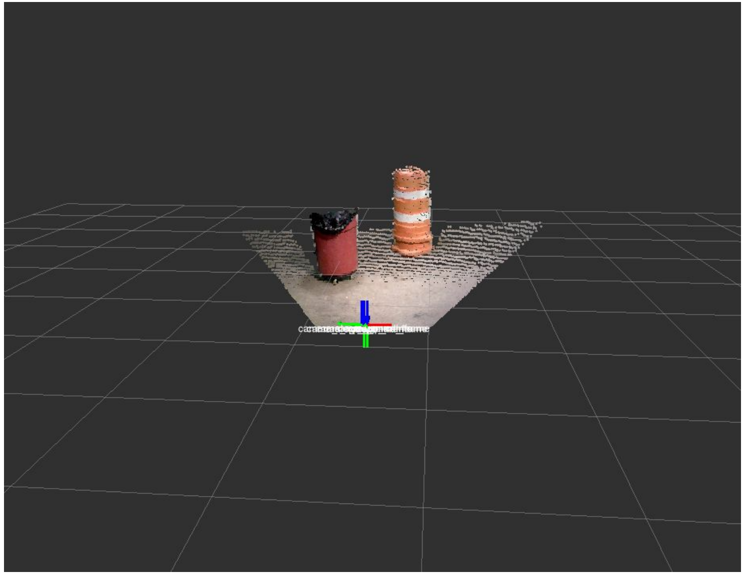


Static Map

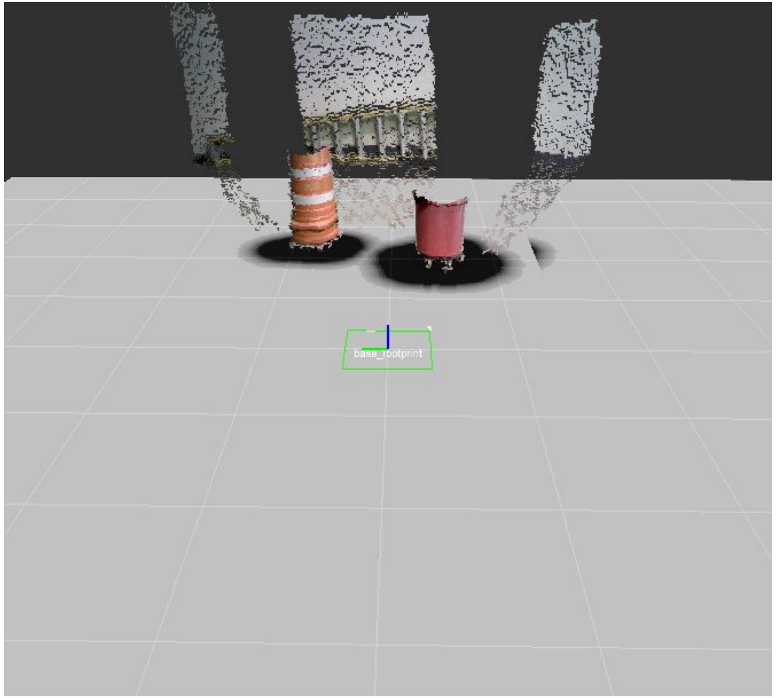


Dynamic Map Building

Obstacle Detection



RBG-D Point Cloud Data



Occupancy Grid

Visualization of Costmap and State Estimation