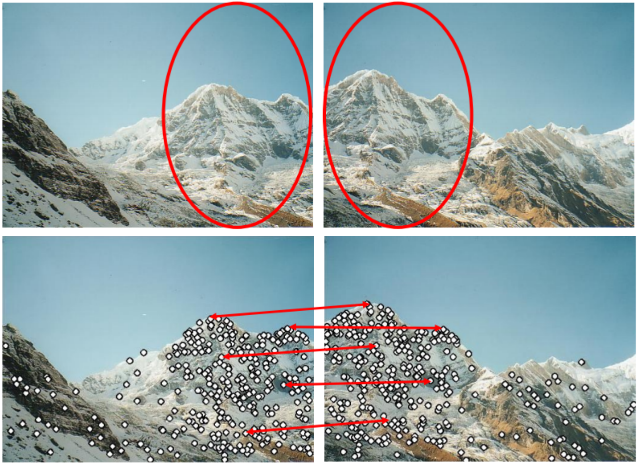
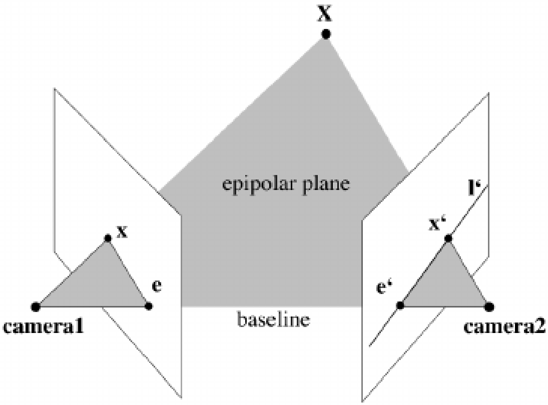
Encoderless Gimbal Calibration

October 18, 2017

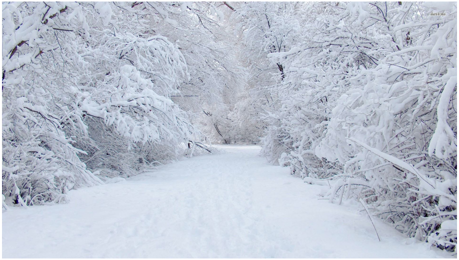
SLAM in a nutshell



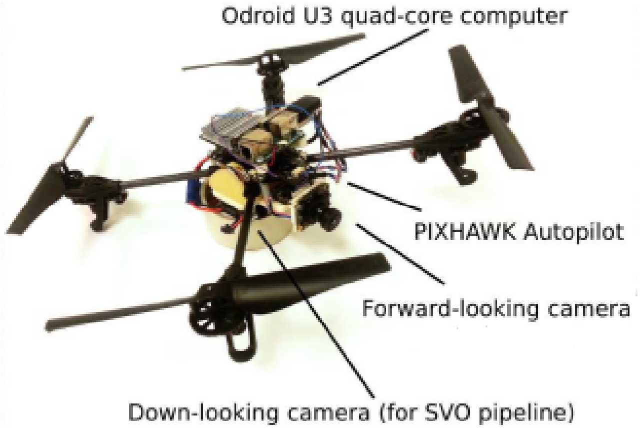
SLAM in a nutshell



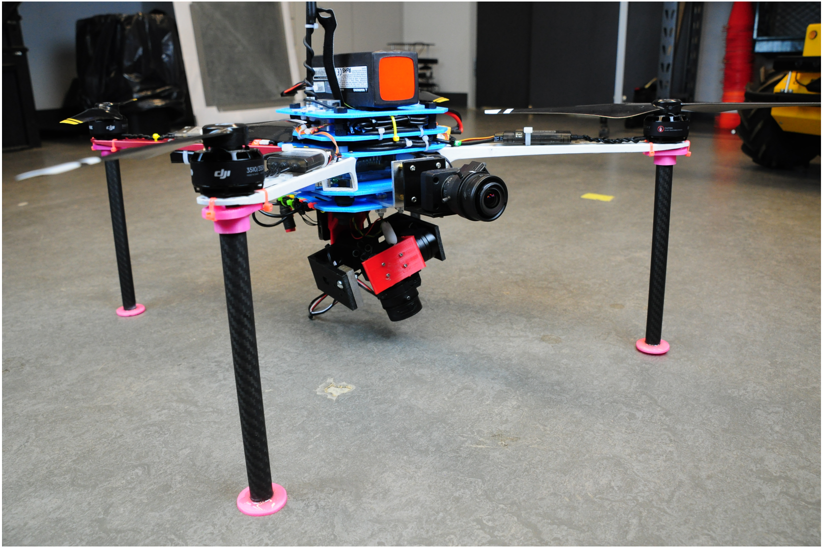
Motivation



Motivation


Odroid U3 quad-core computer
PIXHAWK Autopilot
Forward-looking camera
Down-looking camera (for SVO pipeline)

Motivation



Motivation

The key research goal we're after is **Active Vision**
https://youtu.be/UAU6nvNp7r0

Not so Fast!

But first we need a **Calibration**!

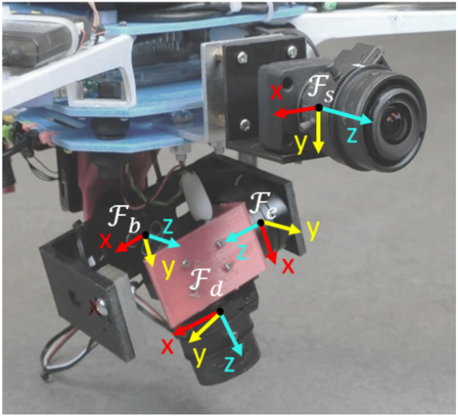
Arun and Jason already done this before:

- Calibration of a dynamic camera cluster for multi-camera visual SLAM (IROS 2016)
- Autonomous Active Calibration of a Dynamic Camera Cluster Using Next-Best-View (IROS 2017)

Previous work assumes the gimbal system has **encoders**.

Our new paper introduces an **encoder-less** overlapping gimbal calibration.

Camera Frames



$$\mathbf{T}^{d:s}_{\Theta,\beta_i} = \mathbf{T}^{d:e}_{\tau_d}\mathbf{T}^{e:b}_{\omega,\beta_i}\mathbf{T}^{b:s}_{\tau_s}$$

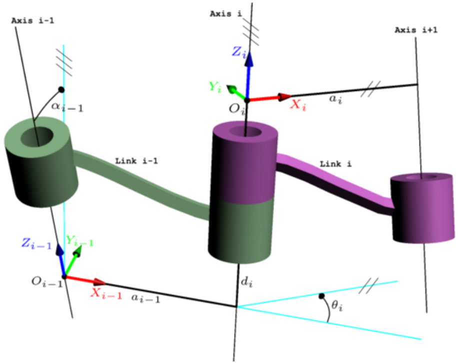
Homogeneous Transform Matrix

$$T^{B:A} = \left[\begin{array}{c|c} R & t \\ \hline \mathbf{0} & \mathbf{1} \end{array}\right] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$p^B = T^{B:W} p^W \tag{2}$$

$$T^B_{\text{obj}} = T^{B:W} T^W_{\text{obj}} \tag{3}$$

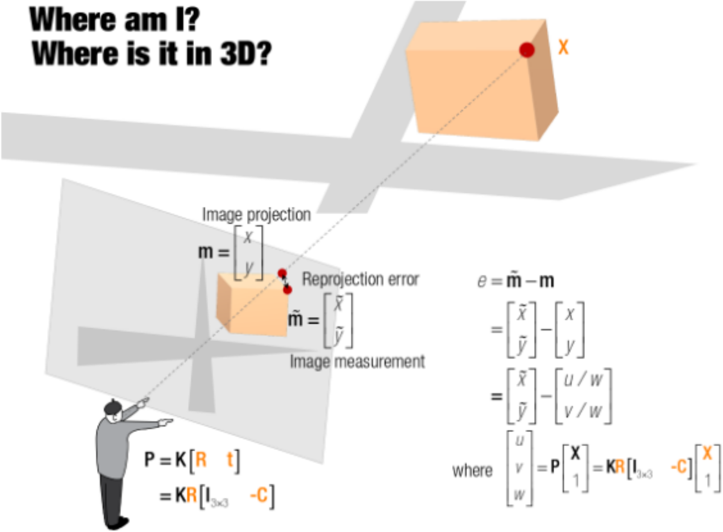
Denavit Hartenberg (aka DH parameters)



- d : offset along previous z to the common normal
- θ : angle about previous z, from old x to new x
- r : length of the common normal (aka a, but if using this notation, do not confuse with α). Assuming a revolute joint, this is the radius about previous z.
- α : angle about common normal, from old z axis to new z axis

Denavit Hartenberg (aka DH parameters)

$$
{}^{n-1}T_n = \left[\begin{array}{ccc|c} \cos\theta_n & -\sin\theta_n\cos\alpha_n & \sin\theta_n\sin\alpha_n & r_n\cos\theta_n \\ \sin\theta_n & \cos\theta_n\cos\alpha_n & -\cos\theta_n\sin\alpha_n & r_n\sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc|c} & R & & T \\ \hline 0 & 0 & 0 & 1 \end{array}\right]
$$


Where am I?
Where is it in 3D?
X
Image projection
$\mathbf{m} = \begin{bmatrix} x \\ y \end{bmatrix}$
Reprojection error
$\tilde{\mathbf{m}} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix}$
Image measurement
$\mathbf{P} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]$
$= \mathbf{K}\mathbf{R}[\mathbf{I}_{3\times3} \quad -\mathbf{C}]$
$e = \tilde{\mathbf{m}} - \mathbf{m}$
$= \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} x \\ y \end{bmatrix}$
$= \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u/w \\ v/w \end{bmatrix}$
where $\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{P}\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}[\mathbf{I}_{3\times3} \quad -\mathbf{C}]\begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$

Reprojection Error

Standard Reprojection Error:

$$\min_{R,C,X} \left|\left| \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} - \begin{bmatrix} u(R,C,X)/w(R,C,X) \\ v(R,C,X)/w(R,C,X) \end{bmatrix} \right|\right|^2 \tag{4}$$

Where we are optimizing over the rotation R, camera center C and 3D scene point X that minimizes the reprojection error.

Reprojection Error - Encoderless Gimbal Calibration version

$$e_j^d(\Theta, \beta_i) = z_j^d - \Psi^d(\mathbf{T}^{d:s}_{\Theta,\beta_i}\mathbf{p}_j) \tag{5}$$

$$e_j^s(\Theta, \beta_i) = z_j^s - \Psi^s((\mathbf{T}^{d:s}_{\Theta,\beta_i})^{-1}\mathbf{p}_j), \tag{6}$$

$$\begin{aligned} \Lambda(\Theta, \zeta) = \sum_{Z_i \in \Gamma} \sum_{j=1}^{|P_i^s|} & e_j^d(\Theta, \beta_i)^T e_j^d(\Theta, \beta_i) \\ & + e_j^s(\Theta, \beta_i)^T e_j^s(\Theta, \beta_i). \end{aligned} \tag{7}$$

$$\Theta^*, \zeta^* = \operatorname*{argmin}_{\Theta,\zeta} \Lambda(\Theta, \zeta) \tag{8}$$

OKVIS - Open Keyframe-based Visual-Inertial SLAM

Contributions:

- Visual and Inertial Odometry / SLAM
- Non-Linear Optimization with added error term for IMU
- Keyframe based
- Tightly coupled - jointly estimate camera pose with IMU

https://www.youtube.com/watch?v=TbKEPA2_-m4

OKVIS - State Vector

$$\mathbf{x}_{\mathrm{R}} := \left[{}_W\mathbf{r}_S{}^T, \mathbf{q}_{WS}^T, {}_S\mathbf{v}{}^T, \mathbf{b}_{\mathrm{g}}^T, \mathbf{b}_{\mathrm{a}}^T\right]^T \in \mathbb{R}^3 \times S^3 \times \mathbb{R}^9$$

position orientation speed IMU biases

$$\mathbf{x}^k_{\beta_k} = [\mathbf{x}^k \ \beta_k]^T$$

where $\beta_k = [\theta^k_1 \ \cdots \ \theta^k_L]$ are the L joint angles of the mechanism to be optimized.

- The cost function to be minimized is:

$$J(\mathbf{x}) := \underbrace{\sum_{i=1}^{I}\sum_{k=1}^{K}\sum_{j\in\mathcal{J}(i,k)} {\mathbf{e}_{\mathrm{r}}^{i,j,k}}^{T}\mathbf{W}_{\mathrm{r}}^{i,j,k}\mathbf{e}_{\mathrm{r}}^{i,j,k}}_{\text{visual}} + \underbrace{\sum_{k=1}^{K-1} {\mathbf{e}_{\mathrm{s}}^{k}}^{T}\mathbf{W}_{\mathrm{s}}^{k}\mathbf{e}_{\mathrm{s}}^{k}}_{\text{inertial}},$$

reprojection error
information
IMU error
information
all cameras
all frames
visible landmarks
all frames

- The authors use Ceres.

Results

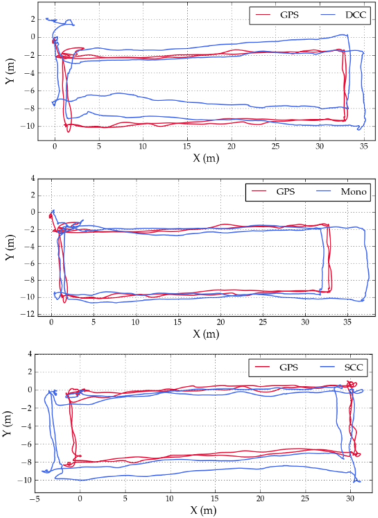


Fig. 3: Estimated trajectories with (a) Dynamic Camera Cluster, (b) Monocular, and (c) Static Camera Cluster Configurations.

Results

	DCC	Mono	SCC
Normalized Translation RMSE (%)	1.58×10^{-2}	1.24×10^{-2}	1.12×10^{-2}
Normalized Rotation RMSE (rad/m)	4.3×10^{-4}	2×10^{-4}	9.4×10^{-4}

TABLE V: Translation and rotation error of the three camera configurations, normalized by the total distance travelled.

Results

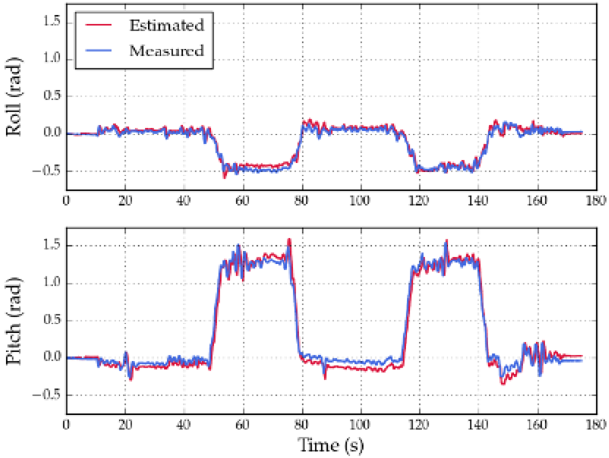


Fig. 5: Estimated gimbal joint angles compared to ground truth provided by the gimbal encoders.

Results

	DCC
Roll Angle RMSE (rad)	4.5×10^{-2}
Pitch Angle RMSE (rad)	9.2×10^{-2}

TABLE VI: Roll and Pitch RMSE for the DCC configuration.

Results

ICRA 2018 video: https://www.youtube.com/watch?v=v0OLmIKHQIM

Future Directions

Quadrotor Mapping with NBV?:
https://www.youtube.com/watch?v=54s6gGZLpJo