Online Calibration For SLAM
(Amazon Lab126)
Advisor : Prof Michael Kaess

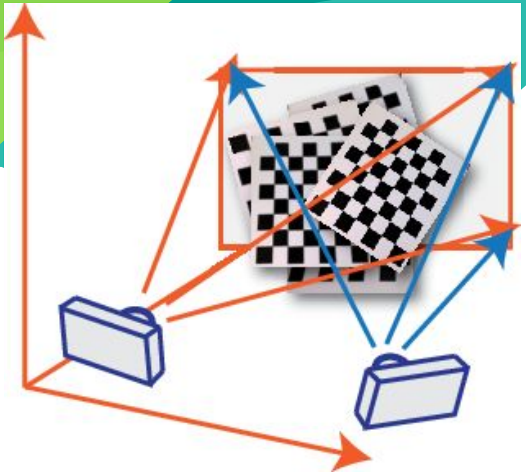
| Jingguang Zhou | Syed Ashar Javed |

Background

- Stereo Cameras
  - Assumption: parallel

- Stereo Relative Extrinsics
  - Vibration, Collision
  - Changes over time

Background



Offline Calibration:

- OpenCV
- Kalibr (J. Maye et al. 13'IVS, Joern et al. 16'ICRA)
- CamOdoCal (Heng et al. ETH, 13'IROS 14'ICRA 15'JFR)

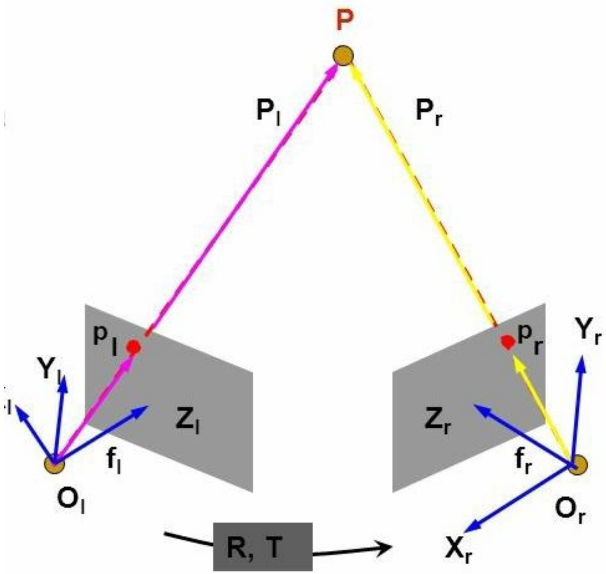
Problem:

- Offline
- Need a chessboard
- CamOdoCal: pre-build a map (>= 1 hour)

Background

Parallel?

No. So we compute the relative extrinsics for calibration

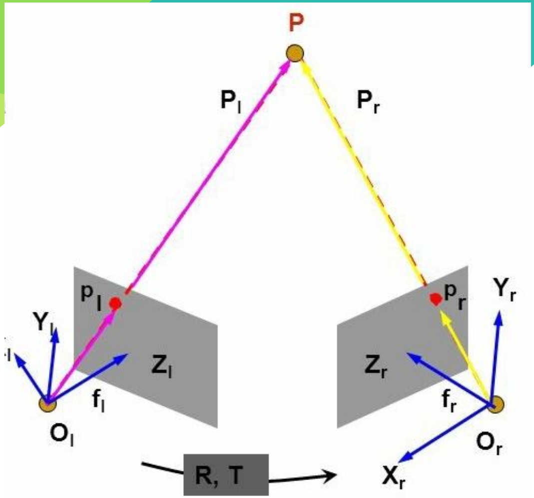


Solutions

- Online Calibration for Stereo SLAM:
- **Front-End**
- Geometric optimization w.r,t Epipolar function
- Correspondence-aware features

- **Back-End**
- Stereo Batch-Optimization
  - Err1 = z - h(pose, mappoints)
  - Err2 = z - h(pose, mappoints, R, t)

Geometric Optimization



- $x_2 E x_1 = 0$ E => [R t]
  - DLT?
  - Five-point method?
- Problem: Two Steps
- One step: optimize geometric distance directly w.r.t **[R t]**

$$d(\mathbf{x}_i', \hat{\mathbf{x}}_i') = \left((x_i'/w_i' - \hat{x}_i'/\hat{w}_i')^2 + (y_i'/w_i' - \hat{y}_i'/\hat{w}_i')^2 \right)^{1/2}$$

$$\mathbf{J}_i(\hat{\mathbf{t}}, \hat{\mathbf{R}}) = [\frac{\partial \mathbf{r}_i(\hat{\mathbf{t}}, \hat{\mathbf{R}})}{\partial \delta \boldsymbol{\theta}} \quad \frac{\partial \mathbf{r}_i(\hat{\mathbf{t}}, \hat{\mathbf{R}})}{\partial \delta \mathbf{t}}],$$

$$\frac{\partial \mathbf{r}_i(\hat{\mathbf{t}}, \hat{\mathbf{R}})}{\partial \delta \boldsymbol{\theta}} = -(\mathbf{f}_i')^T \lfloor \hat{\mathbf{t}} \times \rfloor \hat{\mathbf{R}} \lfloor \mathbf{f}_i \times \rfloor,$$

$$\frac{\partial \mathbf{r}_i(\hat{\mathbf{t}}, \hat{\mathbf{R}})}{\partial \delta \mathbf{t}} = [(\mathbf{f}_i')^T \lfloor \mathbf{b}_1 \times \rfloor \hat{\mathbf{R}} \mathbf{f}_i \quad (\mathbf{f}_i')^T \lfloor \mathbf{b}_2 \times \rfloor \hat{\mathbf{R}} \mathbf{f}_i].$$

High-precision Online Markerless Stereo Extrinsic Calibration Ling et al. Hkust, 16'IROS.

Solutions

- Online Calibration for Stereo SLAM:
- **Front-End**
- Geometric optimization w.r.t Epipolar function
- Correspondence-aware features

- **Back-End**
- Stereo Batch-Optimization
  - Err1 = z - h(pose, mappoints)
  - Err2 = z - h(pose, mappoints, R, t)

DL-based features

- Combine DL and Epipolar Constraints
  - ***Still follow*** $\mathbf{x}_2 \mathbf{E} \mathbf{x}_1 = 0$ $\mathbf{E} \Rightarrow [\mathbf{R}\ \mathbf{t}]$
  - But use DL to estimate $(\mathbf{x}_1, \mathbf{x}_2)$ better

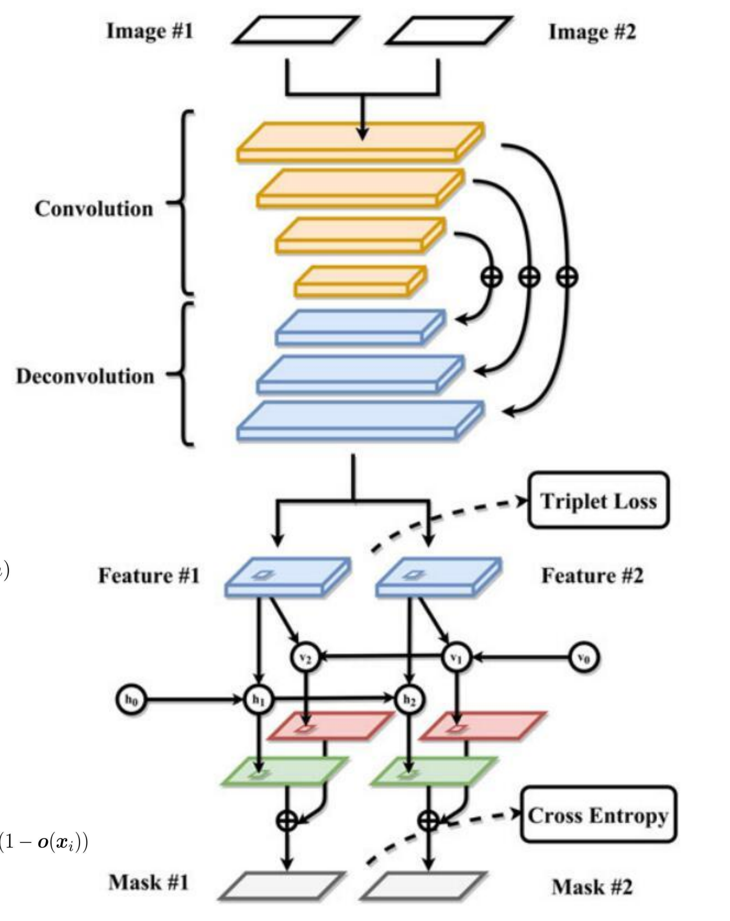
DL-based features

- Combine DL and Epipolar Constraints
  - ***Still follow*** $\mathbf{x}_2\mathbf{E}\mathbf{x}_1 = 0$ **E => [R t]**
  - But use DL to estimate ($\mathbf{x}_1$, $\mathbf{x}_2$) better

$$L_{metric} = \sum_i \max(0, d_{\boldsymbol{x}_i,\boldsymbol{x}_{i,+}} - d_{\boldsymbol{x}_i,\boldsymbol{x}_{i,-}} + m)$$
$$d_{\boldsymbol{x}_1,\boldsymbol{x}_2} = (\mathrm{sign}(\boldsymbol{f}_1(\boldsymbol{x}_1)) - \mathrm{sign}(\boldsymbol{f}_2(\boldsymbol{x}_2)))^2$$

- Network outputs:
  - Keypoints + Binary Descriptors

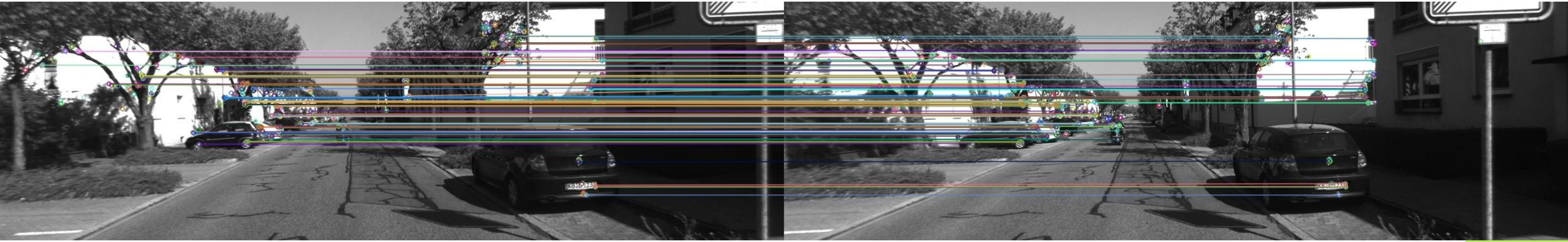
$$L_{mask} = L_{ce}(\boldsymbol{o}_1, \boldsymbol{x}) + L_{ce}(\boldsymbol{o}_2, \boldsymbol{x}_+)$$
$$L_{ce}(\boldsymbol{o}, \boldsymbol{x}) = -\sum_i(\alpha_1 c_{\boldsymbol{x}_i} \log(\boldsymbol{o}(\boldsymbol{x}_i)) + \alpha_2(1 - c_{\boldsymbol{x}_i})\log(1 - \boldsymbol{o}(\boldsymbol{x}_i)))$$



Geometric Correspondence Network for Camera Motion Estimation, Jiexiong Tang et al. 18'RA-L

Experiment

Experiment



Experiment

R(degree) = [1, 3, 1.5]

R(deg) = [1, 3, 1.5]	Data Pair 0	Kitti Pair 1	Kitti Pair 2	Kitti Pair 3
Eight-point	0.0012 0.0007 0.0007	33.5877 0.0193 0.0552	33.6488 0.0285 0.0803	0.0385 0.0043 0.0142
Five-point & ransac	0.9970 3.0405 1.4473	1.4888 5.1309 1.5166	0.7540 7.8351 1.9219	1.5491 6.0329 0.9743
Geometric Optimization(NLS)	1.0219 3.0151 1.4466	1.0090 1.8294 1.4373	1.0024 4.4919 1.1790	0.7490 3.1005 1.4094

Experiment

R(deg) = [1, 3, 1.5]	Kitti Pair 1	Kitti Pair 2	Kitti Pair 3
Five-point & ransac + ORB	2.0814 8.4261 0.9234	0.1321 1.7543 2.0075	0.4641 1.9125 1.6142
Five-point & ransac + DL	1.4888 5.1309 1.5166	0.7540 7.8351 1.9219	1.5491 6.0329 0.9743
Geometric Optimization(NLS) + ORB	1.5539 2.0433 1.4570	1.3351 6.1566 0.9303	0.5984 2.1586 0.9827
Geometric Optimization(NLS) + DL	1.0090 1.8294 1.4373	1.0024 4.4919 1.1790	0.7490 3.1005 1.4094

Discussion

- Online Calibration for Stereo SLAM:
- **Front-End**
- Geometric optimization w.r.t Epipolar function
- Correspondence-aware features

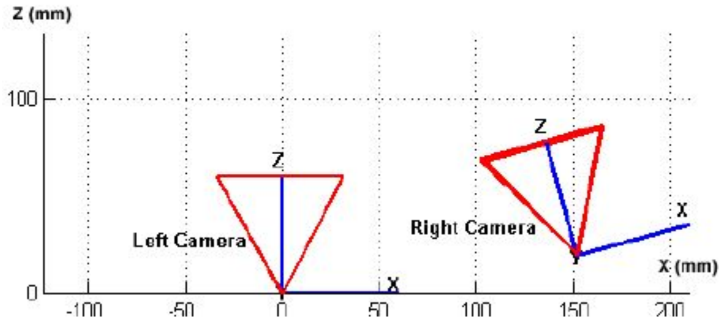
- **Future**
- More stable algorithms
- Less error on Kitti data

Backend overview

- **Problem statement**
- Factor graph formulation
- Evaluation: simulation setup
- Results, analysis & reformulation
- Detecting miscalibration
- Future work

Problem statement

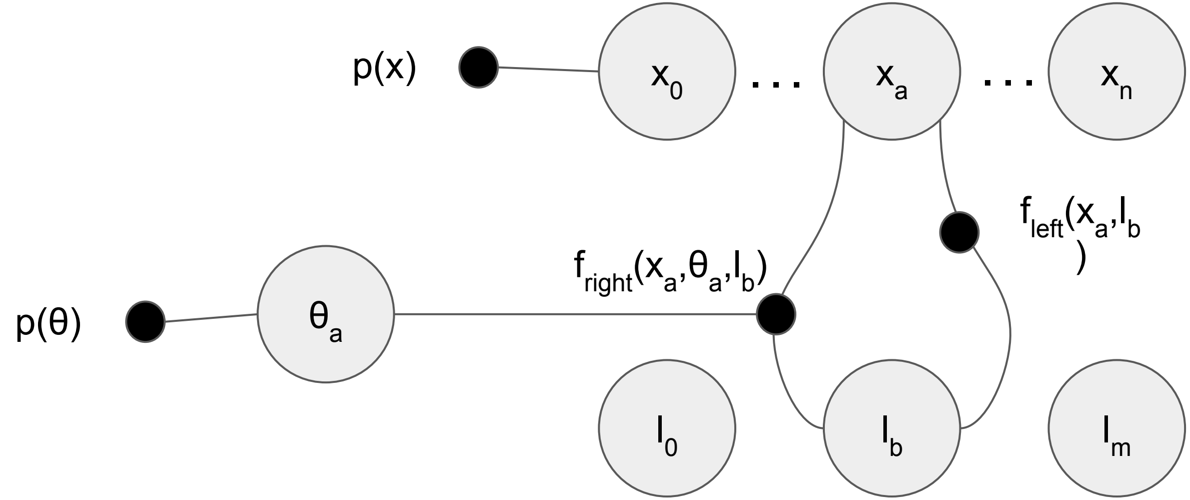
- Thermal expansion of rig or minor bumps change relative pose of stereo pair
- Need to estimate stereo extrinsics (mainly R)
- Expected errors in range 1e-2 to 5e-2 in degrees



Backend overview

- Problem statement
- **Factor graph formulation**
- Evaluation: simulation setup
- Results, analysis & reformulation
- Detecting miscalibration
- Future work

Factor graph formulation



x : Pose

l : Landmark

θ : Stereo extrinsics

p(·) : Prior factor on variable

f(·) : Measurement factor

Factor graph formulation

Maximizing data (X) likelihood given measurements(Z)

$$x = [x_0, x_1, ...x_n]$$
$$l = [l_0, l_1, ...l_m]$$
$$\theta = [R_{calib}, t_{calib}]$$
$$X = [x, l, \theta]$$
$$\eta = [R_{cam}, t_{cam}]$$

$$P(X|Z) = \prod_{i=1}^{j \times k \times l} P(z_i|X)$$

j^{th} landmark, k^{th} frame, l^{th} camera

$$z_i \in R^{j \times k \times l}$$

$$P(z_i|X) = exp(-||z_i - h_i(X)||^2_{\Sigma}))$$

Factor graph formulation

$$\arg\max_X \prod_{i=1}^{j\times k\times l} P(z_i|X)$$

$$\arg\min_X \sum_{i=1}^{j\times k\times l} ||z_i - h_i(X)||_\Sigma^2)$$

$$\arg\min_\Delta ||A\Delta - b||_2^2)$$

$$h(X) = z_{est}(x, l, \theta)$$

$$z_{est}^L = K^L \phi_\eta^{-1}(l)$$

$$z_{est}^R = K^R \phi_\eta^{-1}(\phi_\theta(l))$$

z_{est} → Estimated measurement
ϕ_θ → Stereo extrinsic transformation matrix
ϕ_η → Robot-world transformation matrix

Backend overview

- Problem statement
- Factor graph formulation
- **Evaluation: simulation setup**
- Results, analysis & reformulation
- Detecting miscalibration
- Future work

Evaluation

- Need a ground truth to compare against which itself is very precise
- Need to run controlled ablations

Answer: Simulation

Simulation setup

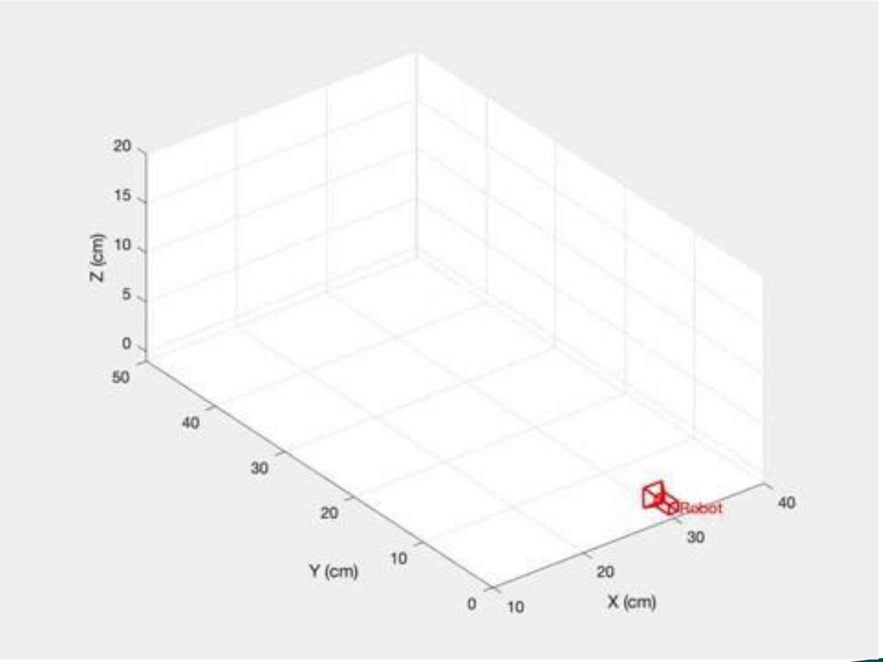
- The robot is shipped with known extrinsics θ^{factory}
- Before simulation, the extrinsics change to θ^{deviated}
- Measurements are made with the new extrinsics θ^{deviated} but the factor graph is run with the initial values θ^{factory}
- Task of optimization is to retrieve θ^{deviated}

Simulation setup

- Robot moving on the ground
- Observes 3D feature points on the way
- Factor graph optimization is run in GTSAM after 16 timesteps
- Process is repeated 1000 times and calibration errors are averaged

Simulation setup

Sample Trajectory



Simulation setup

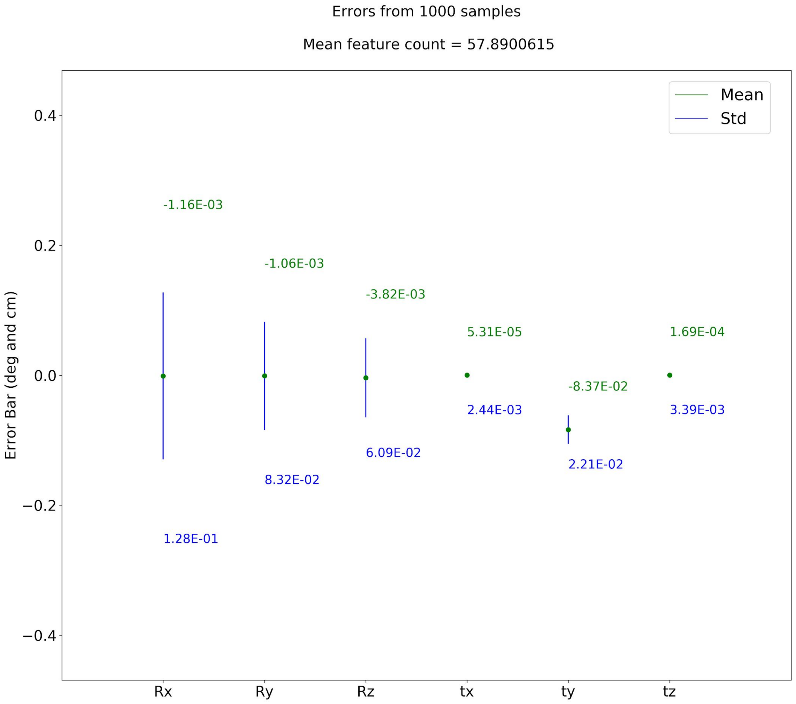
Ablations:

- No of features seen
- Deviation of extrinsics
- Measurement noise
  - Measurement rounding
  - Measurement noise
- Odometry noise

Backend overview

- Problem statement
- Factor graph formulation
- Evaluation: simulation setup
- **Results, analysis & reformulation**
- Detecting miscalibration
- Future work

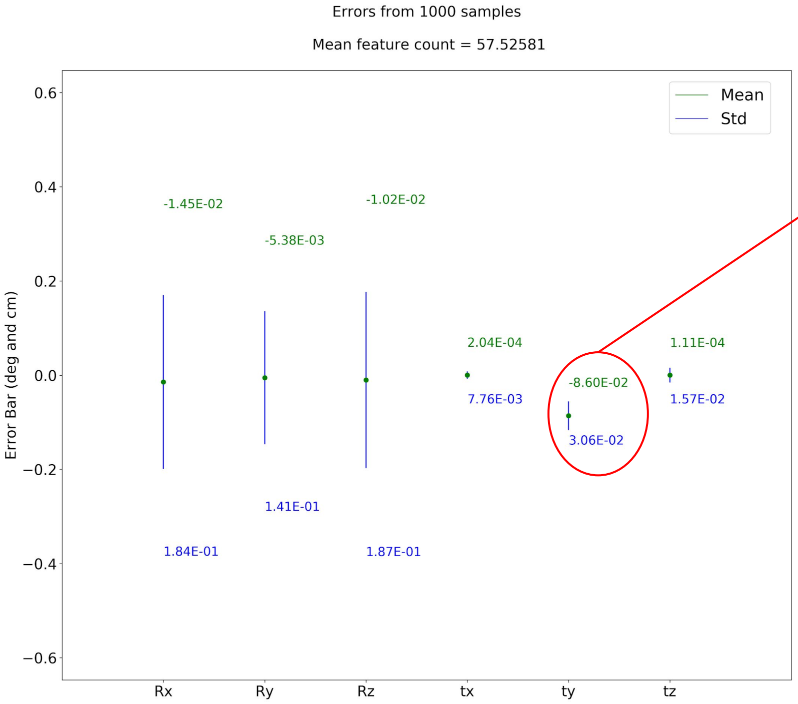
Results and analysis



Base Experiment

- 58.5 features seen on average
- 1.1 deg extrinsic deviation
- 1 pixel measurement noise
- Feature rounding on
- Odometry noise is on

Results and analysis



Baseline changes. Needs constrained optimization

High extrinsic deviation

- 57.5 features seen on average
- 2.3 deg extrinsic deviation
- 1 pixel measurement noise
- Feature rounding on
- Odometry noise is on

Constrained optimization

How can we prevent the baseline from changing:

- Strong prior on initial estimate

Idea: Place a very low variance on the initial relative translation estimate

Problem: Multiplicative Gaussian terms in the likelihood $\rightarrow$ numerical instability

Constrained optimization

How can we prevent the baseline from changing:

- Equality constraints during optimization

Idea: Enforce hard equality constraint on the relative translation variable

Problem: Increases the optimization time substantially

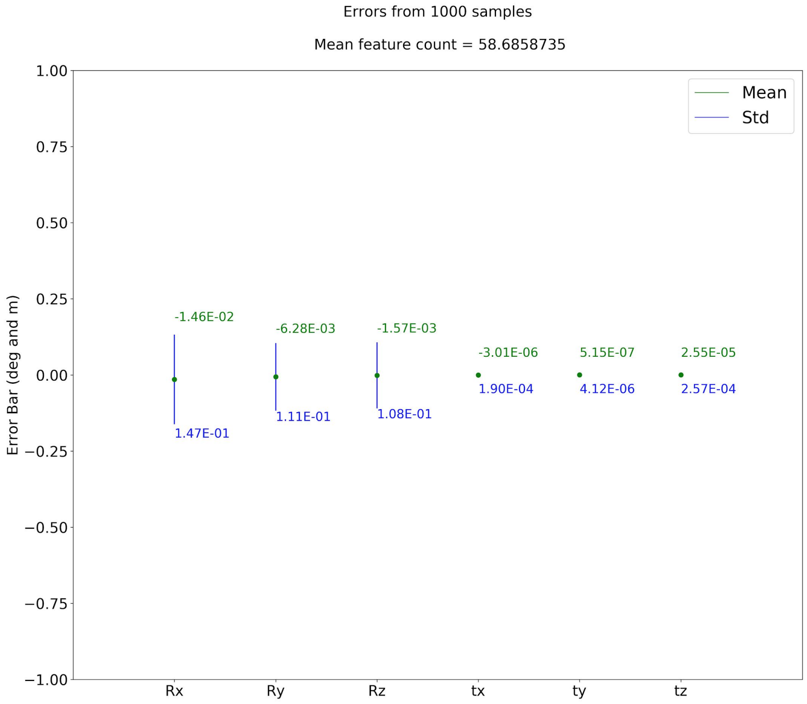
Constrained optimization

How can we prevent the baseline from changing:

- Optimize only over the rotation manifold

Idea: Use ground truth translation in the measurement factor

Updated results

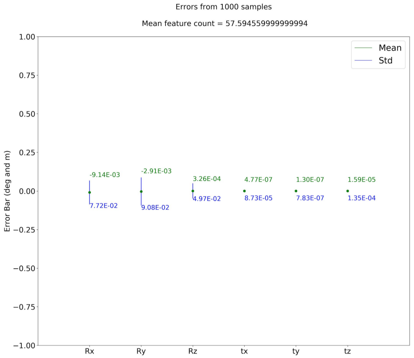


Base Experiment

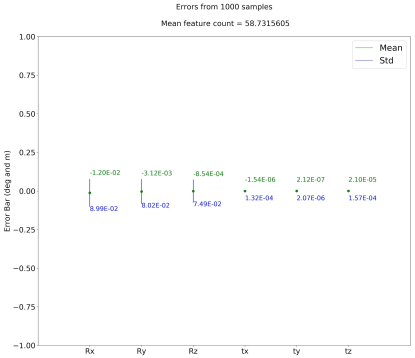
- 58.6 features seen on average
- 1.1 deg extrinsic deviation
- 1 pixel measurement noise
- Feature rounding on
- Odometry noise is on

Ablations

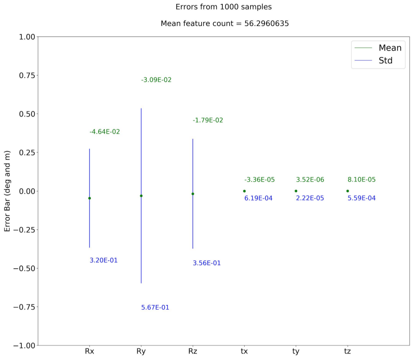
Extrinsic Deviation



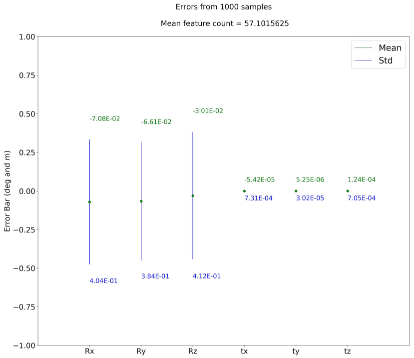
0.6 deg



1.1 deg



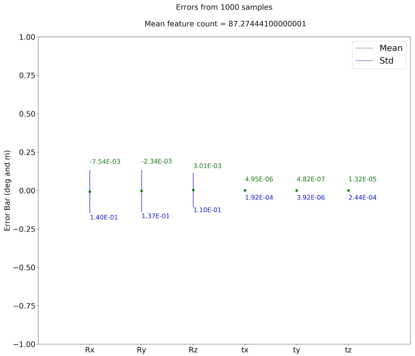
1.7 deg



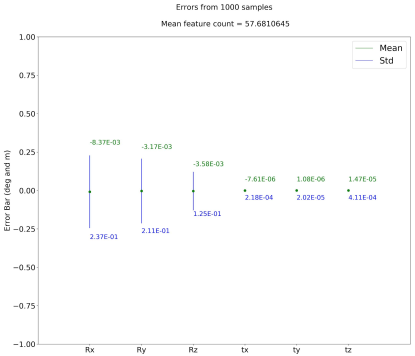
2.3 deg

Ablations

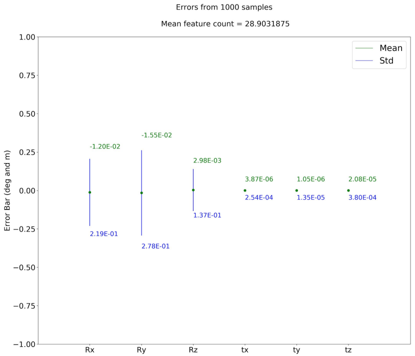
Average Feature Count



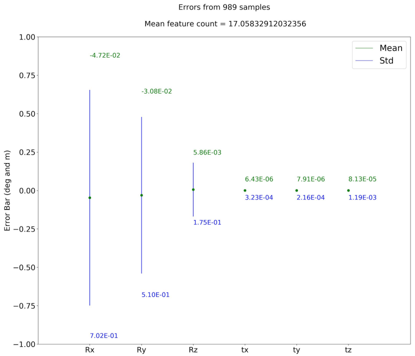
87



58



29



17

Backend overview

- Problem statement
- Factor graph formulation
- Evaluation: simulation setup
- Results, analysis & reformulation
- **Detecting miscalibration**
- Future work

When to re-calibrate?

How can we know when to re-calibrate:

- Information theoretic metrics
- Reprojection error metric
- Epipolar error metric

When to re-calibrate?

How can we know when to re-calibrate:

- Information theoretic metrics

Fisher Information Matrix (FIM):

Amount of information in an observable random variable X about an unknown parameter θ

When to re-calibrate?

How can we know when to re-calibrate:

- Information theoretic metrics

For Gaussians,

FIM = inverse(Marginal Covariance of θ)

$$\left(\bar{\Sigma}_{\theta}^{-1}\right)$$

When to re-calibrate?

How can we know when to re-calibrate:

- Information theoretic metrics

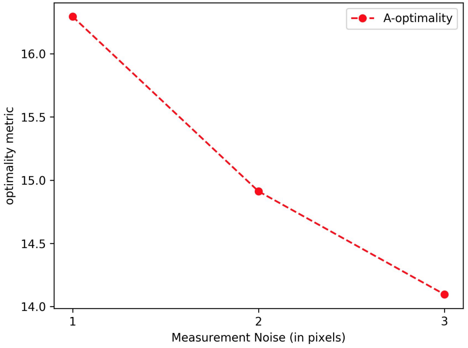
1. A-optimality: $\operatorname{trace}\left(\overline{\boldsymbol{\Sigma}}_\theta^{-1}\right)$
2. D-optimality: $\det\left(\overline{\boldsymbol{\Sigma}}_\theta^{-1}\right)$
3. E-optimality: $\max\left(\operatorname{eig}\left(\overline{\boldsymbol{\Sigma}}_\theta^{-1}\right)\right)$

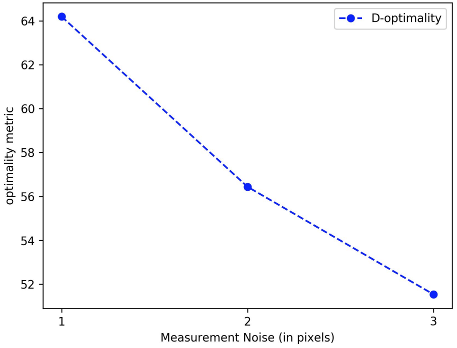
When to re-calibrate?

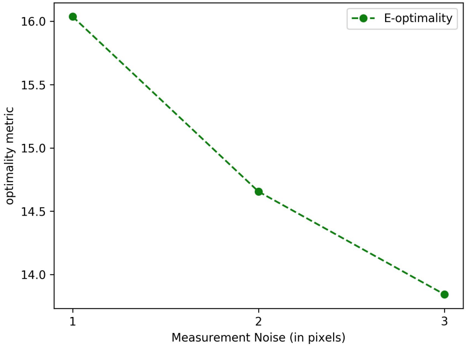
How reliable are these metrics for detecting the need to re-calibrate:

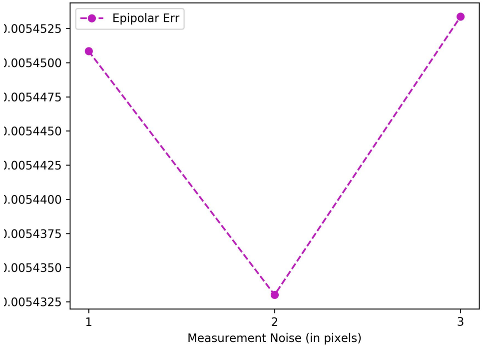
Do the metrics vary as we add sources of noise to the factor graph?

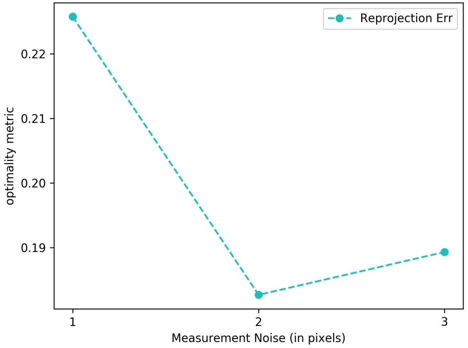
When to re-calibrate?



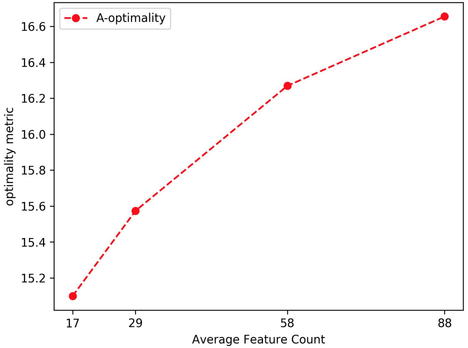


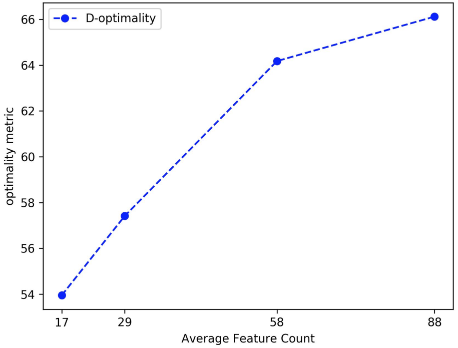


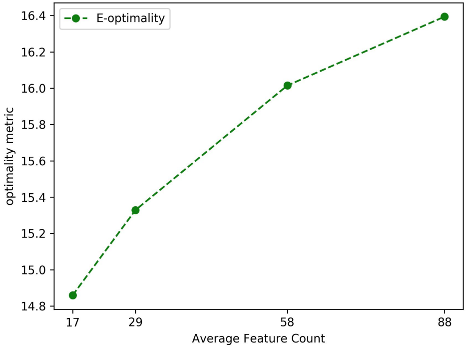


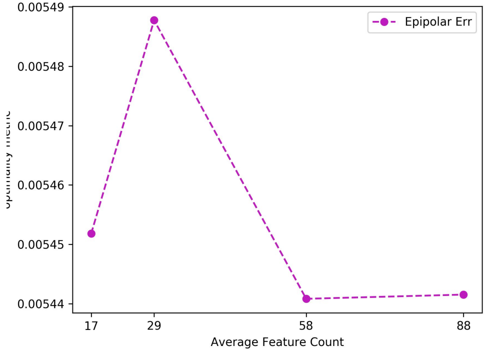


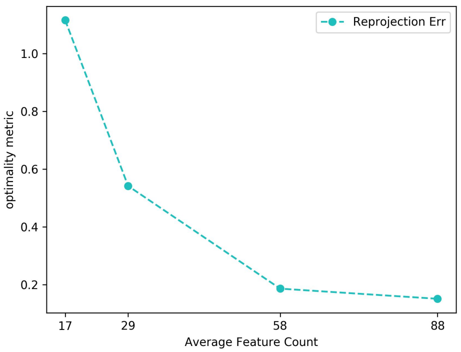
When to re-calibrate?



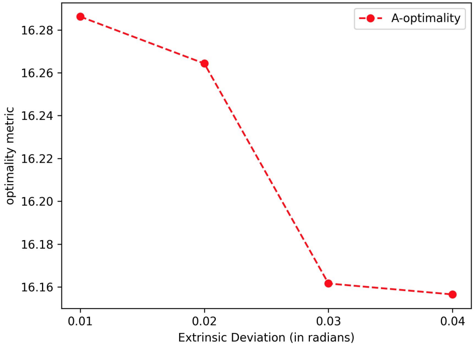


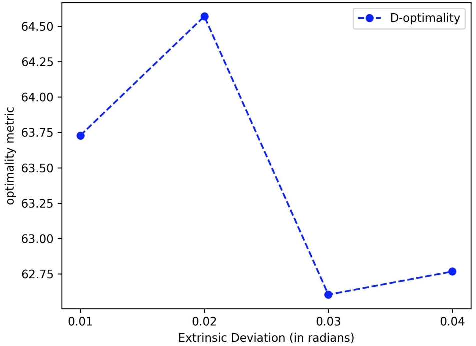


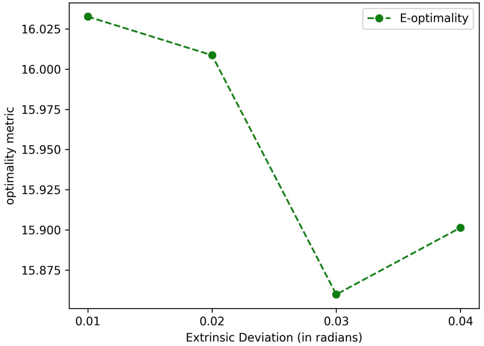


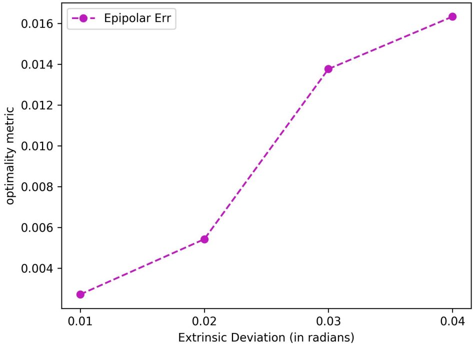


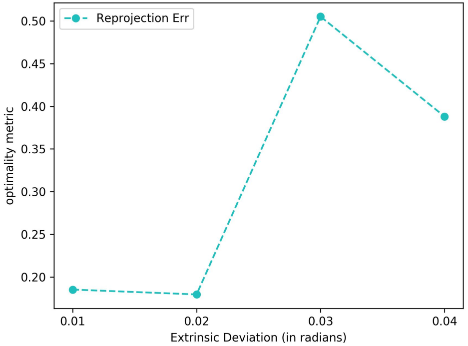
When to re-calibrate?











Can detect miscalibration

Future directions

- Segment selection for batched calibration
- Using IMU data
- Handling arbitrary translational constraint

Thank you