Entropy-Based *Sim*(3) Calibration of 2D Lidars to Egomotion Sensors

Jacob Lambert, Lee Clement, Matthew Giamou, and Jonathan Kelly

MFI 2016, Baden-Baden, Germany

Institute for Aerospace Studies
UNIVERSITY OF TORONTO

STARS LABORATORY

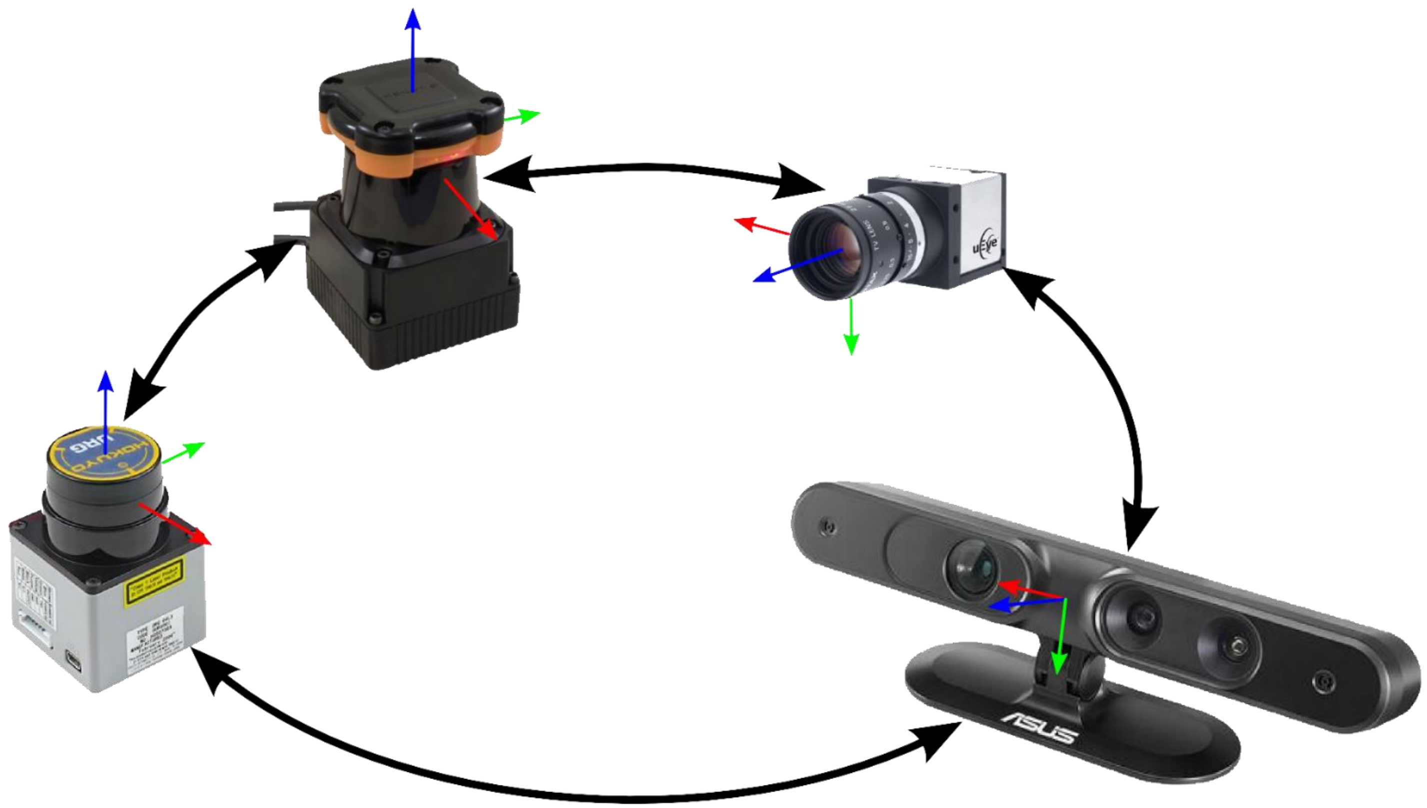
Motivation: Multi-sensor platforms

Multi-sensor platforms are increasingly common in autonomous mobile robotics.

Sensors must be spatially calibrated for data fusion

Extrinsic Sensor Calibration

Manually recovering geometric transformation between sensors **cannot** be done accurately and reliably.



Must use data-driven techniques for calibration

Traditional Method: **Calibration Targets**

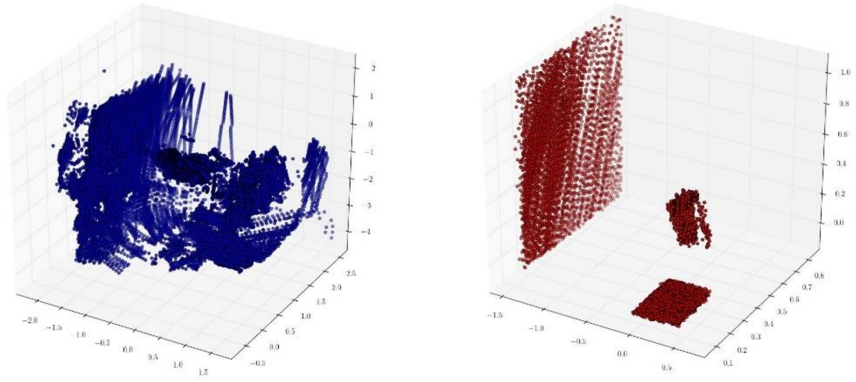
Supervised calibration approach, often requiring specific:

- calibration targets,
- sensor configurations,
- environments or
- trajectories.

State of the Art: **Calibration *in the Wild***

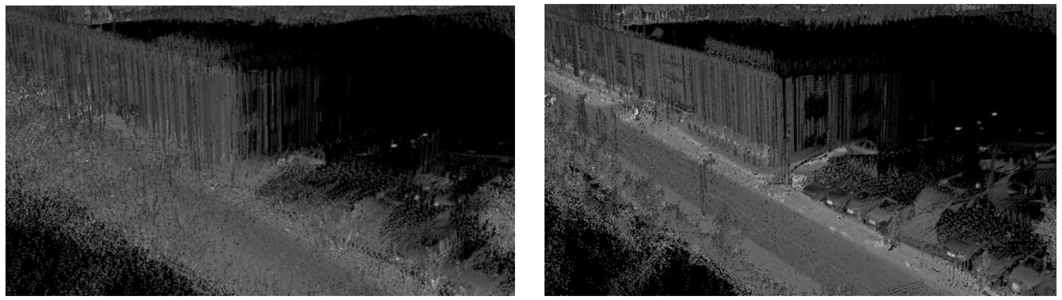
Feature-based: 2D Lidar - IMU - Stereo spatiotemporal calibration

J. Rehder, R. Siegwart and P. Furgale, "A General Approach to Spatiotemporal Calibration in Multisensor Systems," in *IEEE Transactions on Robotics*, 2016.



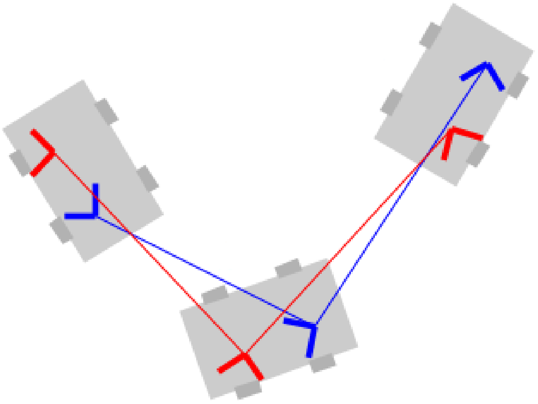
Appearance-based: 3D Lidar-2D Lidar extrinsic calibration

W. Maddern, A. Harrison and P. Newman, "Lost in translation (and rotation): Rapid extrinsic calibration for 2D and 3D LIDARs," *ICRA, 2012*.



Egomotion-based: extrinsic calibration between egomotion sensors

J. Brookshire and S. Teller,"Extrinsic Calibration from Per-Sensor Egomotion," in *Robotics:Science and Systems VIII* , 1, MIT Press, 2013.



RQE Calibration: Overview

Our approach is:

- appearance-based,
- recovers up to *Sim(3*) calibration parameters (3D+Scale),
- Calibrates a lidar (2D or 3D) to egomotion sensor (Camera, IMU*, GNS, etc.) pair,
- poses no restrictions on sensor configuration (FOV) and
- performs reliably *in the wild,* for a broad range of urban and natural environment.

Main contributions are:

- adaptation of RQE-based cost function for *Sim*(3) calibration,
- validation of the cost-function through simulations,
- experimental validation of the 2D Lidar to monocular camera, including non-overlapping FOV case, and
- motivation for the development of a fully spatiotemporal calibration algorithm through entropy.

Derivation: Kinematic Chain

Want: $\mathbf{T}_{C,L}$, from $\underrightarrow{\mathcal{F}}_L$ to $\underrightarrow{\mathcal{F}}_C$, $\boldsymbol{\Xi} = [x_L \;\; y_L \;\; z_L \;\; \phi_L \;\; \theta_L \;\; \psi_L \;\; s]^T$.

Given: camera poses and associated covariance**,**

$$\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_K\}, \quad \mathbf{y}_k = [x_k \; y_k \; z_k \; \phi_k \; \theta_k \; \psi_k]^T \rightarrow \mathbf{T}_{G,C_k}, \quad \mathbf{Q}_k$$

2D lidar measurements and

$$\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_K\}, \quad \mathbf{z}_k = \{\mathbf{z}_k^{(1)}, \mathbf{z}_k^{(2)}, ..., \mathbf{z}_k^{(N)}\}, \quad \mathbf{z}_k^{(n)} = \left[x_k^{(n)} \;\; y_k^{(n)}\right]^T,$$

$$\mathbf{p}_{L_k}^{(n)} = \left[x_k^{(n)} \;\; y_k^{(n)} \;\; 0 \;\; 1\right]^T.$$

a reasonable guess for $\mathbf{T}_{C,L}$.

Apply kinematic chain to transform point cloud to global frame:

$$\hat{\mathbf{p}}_{G,k}^{(n)} = h^{-1}(\mathbf{p}_{L_k}^{(n)} \mid \mathbf{y}_k, \boldsymbol{\Xi}) = \mathbf{T}_{G,C_k} \mathbf{T}_{C_k,L_k} \mathbf{p}_{L_k}^{(n)},$$

$$\boldsymbol{\Sigma}_k^{(n)} = \mathbf{J}_k^{(n)} \mathbf{Q}_k \mathbf{J}_k^{(n)T}, \quad \mathbf{J}_k^{(n)} = \frac{\partial h^{-1}(\mathbf{x}_{L_k}^{(n)} | \mathbf{y}_k, \boldsymbol{\Xi})}{\partial \mathbf{y}_k}.$$

Have: set of 3D points $\hat{\mathbf{x}}_{G,k}^{(n)} \in \hat{\mathbf{X}}$ each with covariance $\boldsymbol{\Sigma}_k^{(n)}$, in **global frame**.

Key Concept: **Entropy**

We use **Renyi Quadratic Entropy** to optimize point cloud quality.

Information Theory

Shannon Entropy:

$$H\left[P\right] = \sum_{i=1}^{\Omega} p_i \log \frac{1}{p_i}$$

Intuition: quantifies the uncertainty related with drawing a measurement from the distribution.

Statistical Mechanics

Gibbs Entropy:

$$H\left[P\right] = -k_B \sum_{i=1}^{\Omega} p_i \log p_i$$

Intuition: measures progress towards equilibrium, often implying uniformity.

All Renyi Entropy of order *alpha* are equivalent **in terms of optimization**:

$$H\left[P\right] = \frac{1}{1-\alpha} \log \sum_{i=1}^{\Omega} p_i^{\alpha}$$

Derivation: Cost Function

Apply **Parzen-Window Density Estimation** on set $\hat{\mathbf{x}}_{G,k}^{(n)} \in \hat{\mathbf{X}}$:

$$p(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^{M} \mathcal{N}(\mathbf{x} - \hat{\mathbf{x}}_i, \mathbf{\Sigma}_i + \sigma^2 \mathbf{I}).$$

Isotropic kernel capturing lidar measurement uncertainty

Choose **Renyi Quadratic Entropy** (RQE), where $\alpha = 2$:

$$H[\hat{\mathbf{X}}] = -\log \int p(\mathbf{x})^2 d\mathbf{x}.$$

Integral for the **convolution of two Gaussians** has a closed form solution:

$$H[\hat{\mathbf{X}}] = -\log \left(\frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \mathcal{N}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j, \mathbf{\Sigma}_i + \mathbf{\Sigma}_j + 2\sigma^2 \mathbf{I}) \right).$$

Minimize entropy with respect to calibration parameters:

$$\mathbf{\Xi}^* = \underset{\mathbf{\Xi}}{\operatorname{argmin}} \; H(\mathbf{\Xi} \mid \mathbf{Y}, \hat{\mathbf{X}}).$$

Algorithm: Computational Enhancements

Problem: the cost function is computationally expensive:

- Computing entropy contribution of all point pairs = $O(N^2)$
- 1 minute of data for a typical 2D Lidar is more than a million points.

Starting cost function:

$$H[\hat{\mathbf{X}}] = -\log\left(\frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\mathcal{N}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j, \mathbf{\Sigma}_i + \mathbf{\Sigma}_j + 2\sigma^2\mathbf{I})\right)$$

Simplifications:

- Ignore constants and monotonic logarithm,
- Remove double-counting,
- Store points in kd-tree and only consider points within some radius:

$$\mathcal{N}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j, \mathbf{\Sigma}_i + \mathbf{\Sigma}_j + 2\sigma^2\mathbf{I}) \approx 0 \text{ if } ||\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j|| \geq 2k\left(\max\left(\lambda_1(\mathbf{\Sigma}_i), \lambda_1(\mathbf{\Sigma}_j)\right) + \sigma^2\right)$$

controls cost function accuracy versus computation time

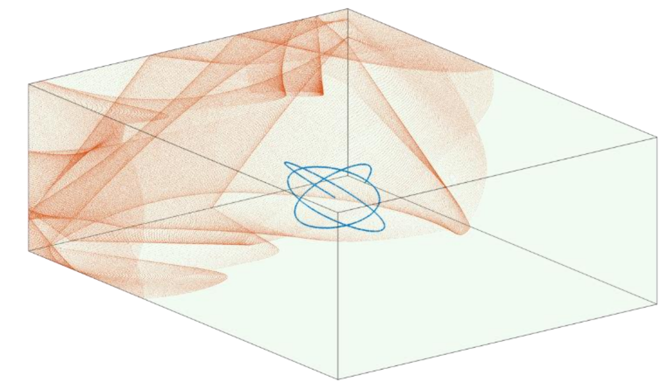
Final cost function:

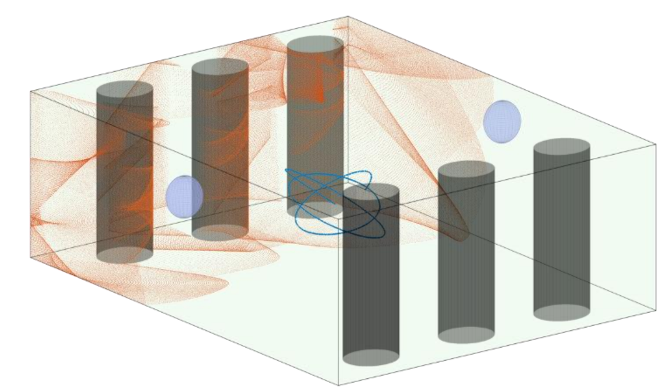
$$H[\hat{\mathbf{X}}] = -\sum_{i=1}^{M}\sum_{j=i}^{M}\mathcal{N}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j, \mathbf{\Sigma}_i + \mathbf{\Sigma}_j + 2\sigma^2\mathbf{I})$$

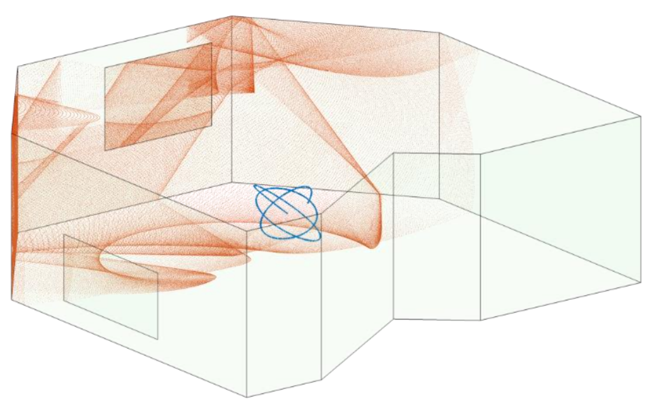
Simulations: Setup

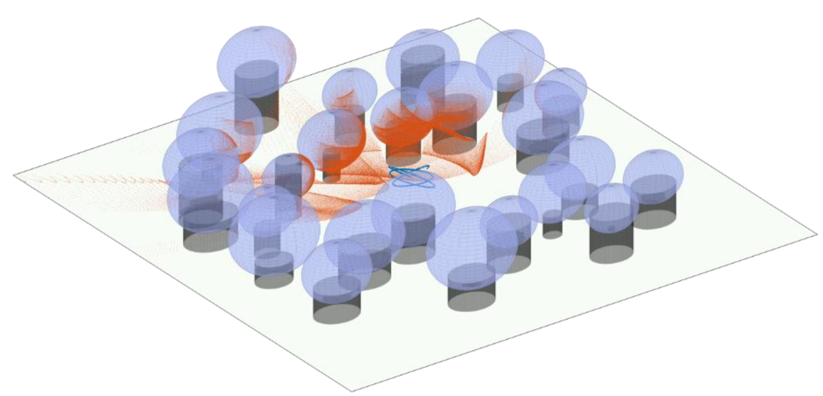
Egomotion sensor reports relative poses:
- pseudo-random sinusoid trajectories,
- 50 mm standard deviation on position,
- 1 degree standard deviation on orientation.

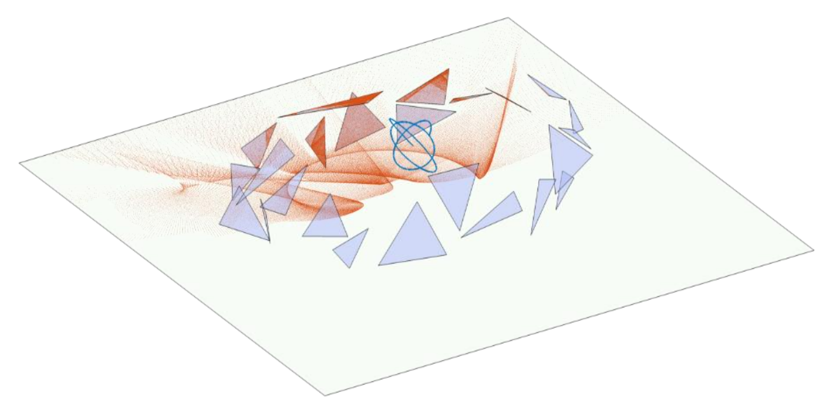
Rigidly attached **lidar sensor:**
- modelled as a Hokuyo UTM-30LX,
- 40 hz scan rate,
- 240 degree field of view and 20 meters range,
- angular resolution of 0.25 degrees per beam,
- 50 mm standard deviation on range measurements.


(a) Simple Room


(b) Underground Parking Lot


(c) Plane City

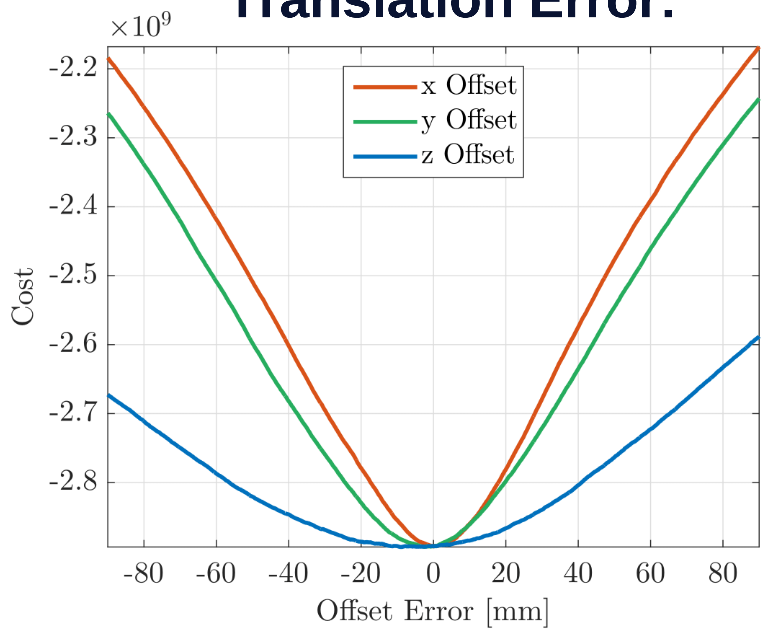

(d) Quadratic Forest


(e) Triangle Array

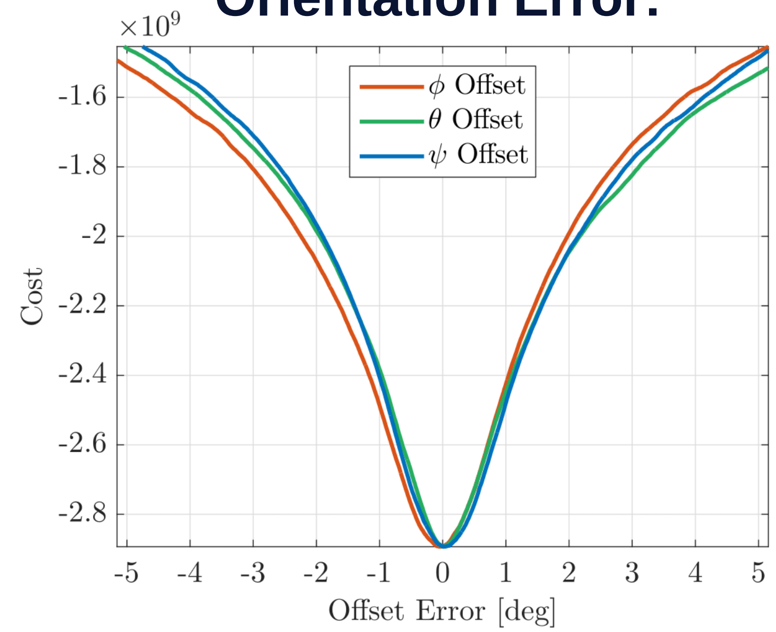
Simulations: Results

Cost function validation: Single parameter variation with others held at their true value:

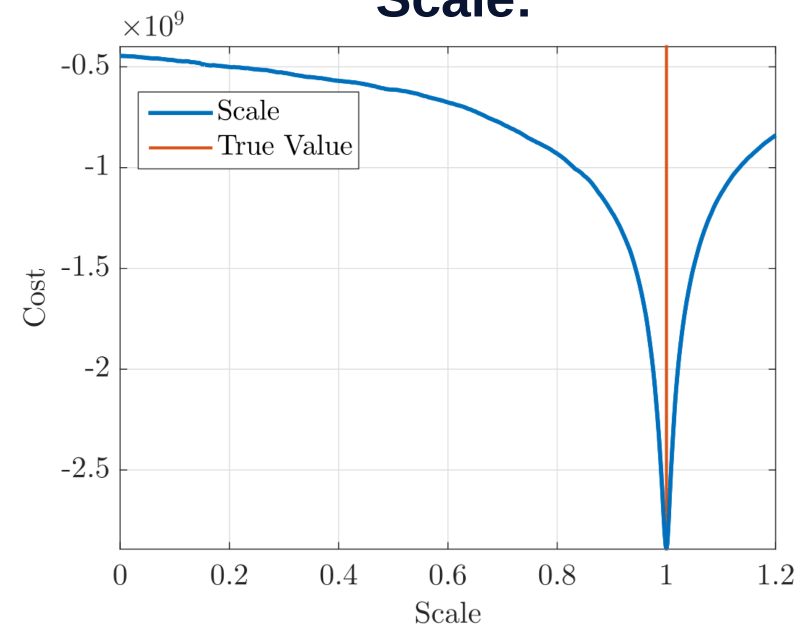
Translation Error:



Orientation Error:



Scale:

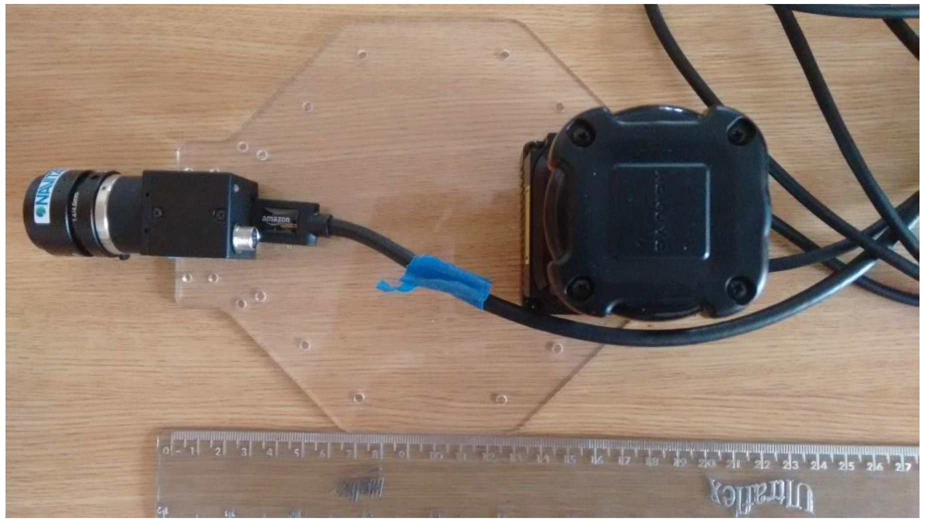


Global optimization: average error over 10 different trajectories.

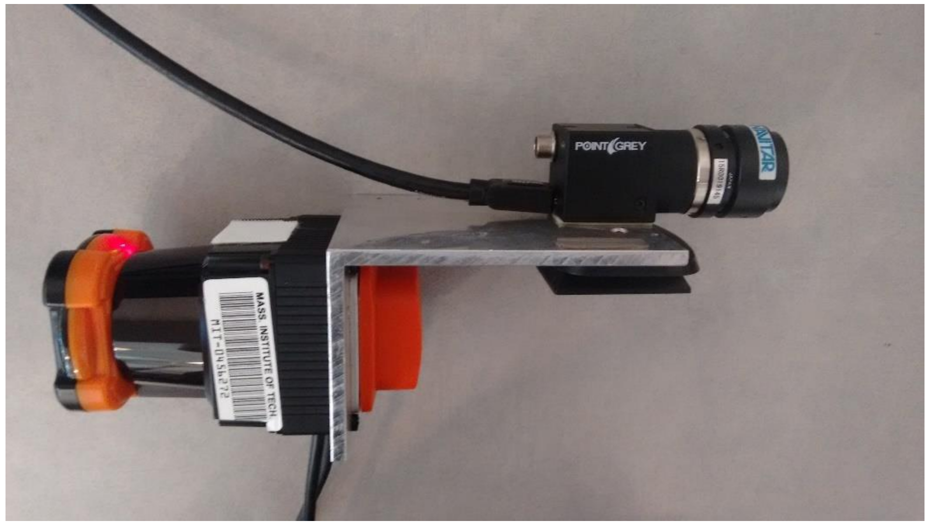
	Average absolute error – μ (σ)						
Environment	x [mm]	y [mm]	z [mm]	ϕ [deg]	θ [deg]	ψ [deg]	Scale [$\times 10^{-3}$]
Simple Room	2.8 (2.5)	3.1 (2.5)	5.2 (3.5)	0.22 (0.12)	0.051 (0.043)	0.24 (0.15)	0.33 (0.30)
Underground Parking Lot	4.5 (4.4)	4.8 (4.1)	5.2 (4.5)	0.37 (0.22)	0.11 (0.11)	0.37 (0.17)	1.2 (0.9)
Plane City	4.1 (2.1)	5.2 (4.2)	4.0 (3.9)	0.38 (0.23)	0.18 (0.06)	0.35 (0.23)	0.69 (0.55)
Quadratic Forest	4.6 (2.6)	3.9 (1.6)	2.9 (3.3)	0.32 (0.20)	0.074 (0.54)	0.35 (0.27)	0.73 (0.32)
Triangle Array	3.0 (1.9)	2.9 (1.6)	4.6 (3.5)	0.64 (0.60)	0.10 (0.07)	0.61 (0.58)	0.47 (0.26)

Experiments: Setup

Hokuyo UTM-30LX lidar and **PointGrey Flea3** camera; two configurations:



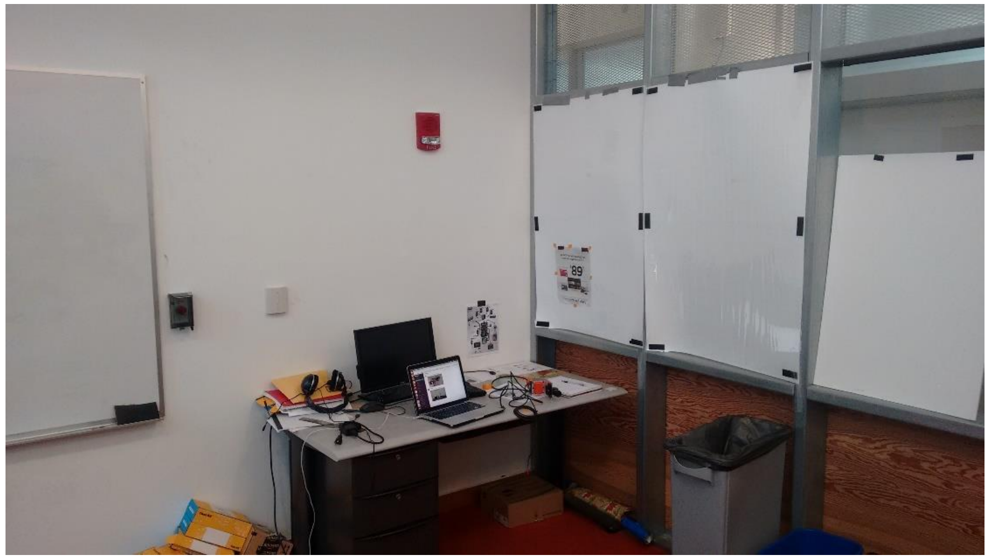
(a) Overlapping FOV



(b) Non-overlapping FOV

- **200 fps camera** synced with **40 hz lidar** according to ROS timestamps,
- Camera pose estimation up to scale through **ORB-SLAM2** (open-source).

Data collected in a cluttered office space in **MIT's Strata Center**:



Experiments: Results

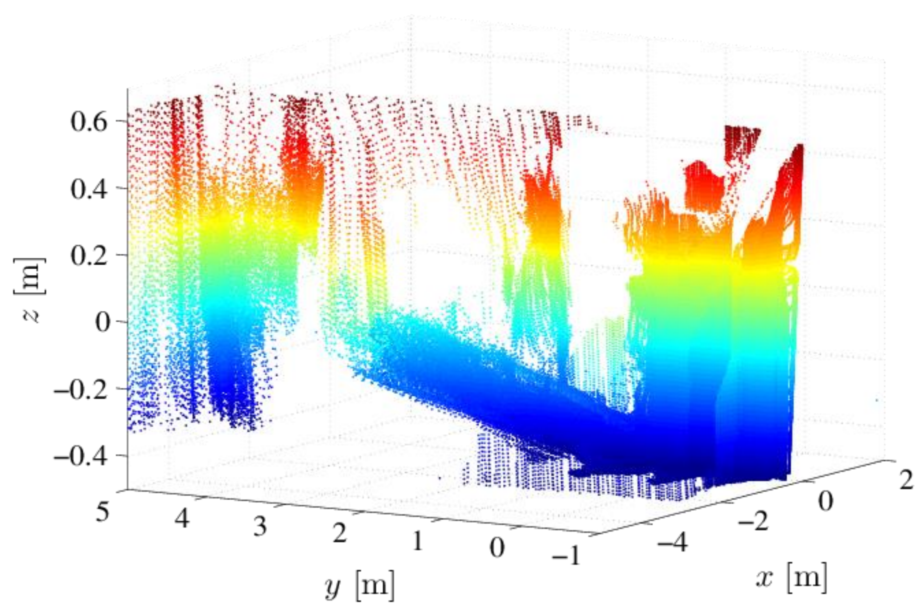
Overlapping FOV

	Calibration results						
	x [mm]	y [mm]	z [mm]	ϕ [deg]	θ [deg]	ψ [deg]	Scale [$\times 10^{-3}$]
Initial Guess	**160.0**	**0.0**	**-50.0**	**0.400**	**-90.00**	**0.00**	**-90.00**
Trial I	-178.2	-3.8	-45.8	90.58	-0.10	-90.74	0.506
Trial II	182.5	-2.8	-50.9	-90.22	0.14	-90.12	0.509
Trial III	173.6	-2.9	-47.8	89.84	-0.27	-90.04	0.506
Trial IV	187.0	-4.9	-54.2	-89.49	-0.02	-90.27	0.511
μ (σ)	180.3 (5.0)	-3.6 (0.8)	-49.7 (3.2)	-90.03 (0.41)	0.06 (0.15)	-90.29 (0.27)	0.508 (0.002)

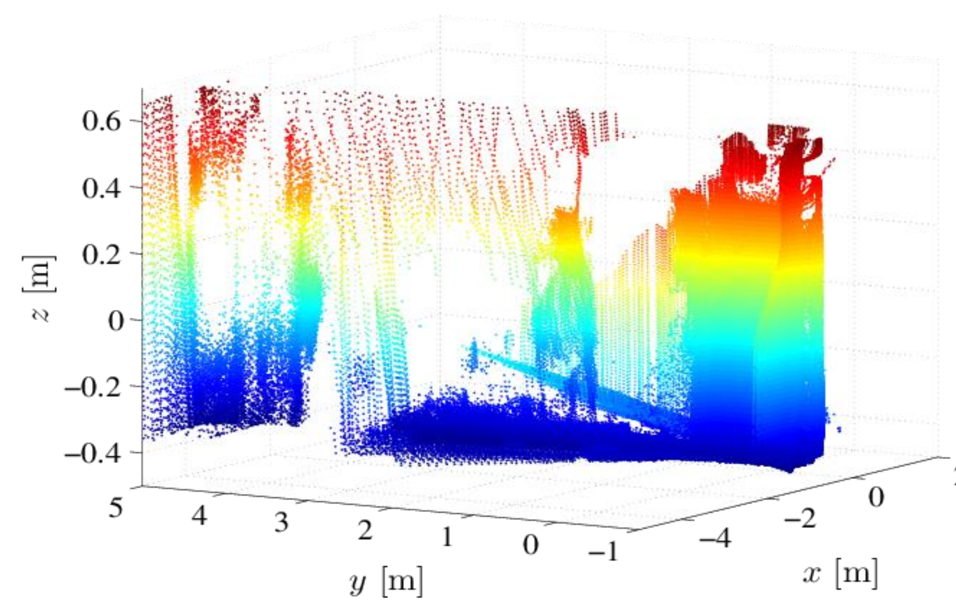
Non-overlapping FOV

	Calibration results						
	x [mm]	y [mm]	z [mm]	ϕ [deg]	θ [deg]	ψ [deg]	Scale [$\times 10^{-3}$]
Initial Guess	**50.0**	**0.0**	**-250.0**	**180.00**	**0.00**	**-90.00**	**0.500**
Trial I	45.2	0.5	-202.1	180.13	-1.11	-88.89	0.2149
Trial II	42.2	-1.2	-202.8	180.73	-1.29	-88.93	0.2159
Trial III	44.0	0.2	-204.2	180.43	-1.78	-88.89	0.2156
μ (σ)	43.8 (1.2)	-0.2 (0.8)	203.0 (0.85)	180.43 (0.24)	-1.39 (0.28)	-88.90 (0.02)	0.2156 (0.0005)

Point cloud **before** calibration



Point cloud **after** calibration



Institute for Aerospace Studies
UNIVERSITY OF TORONTO

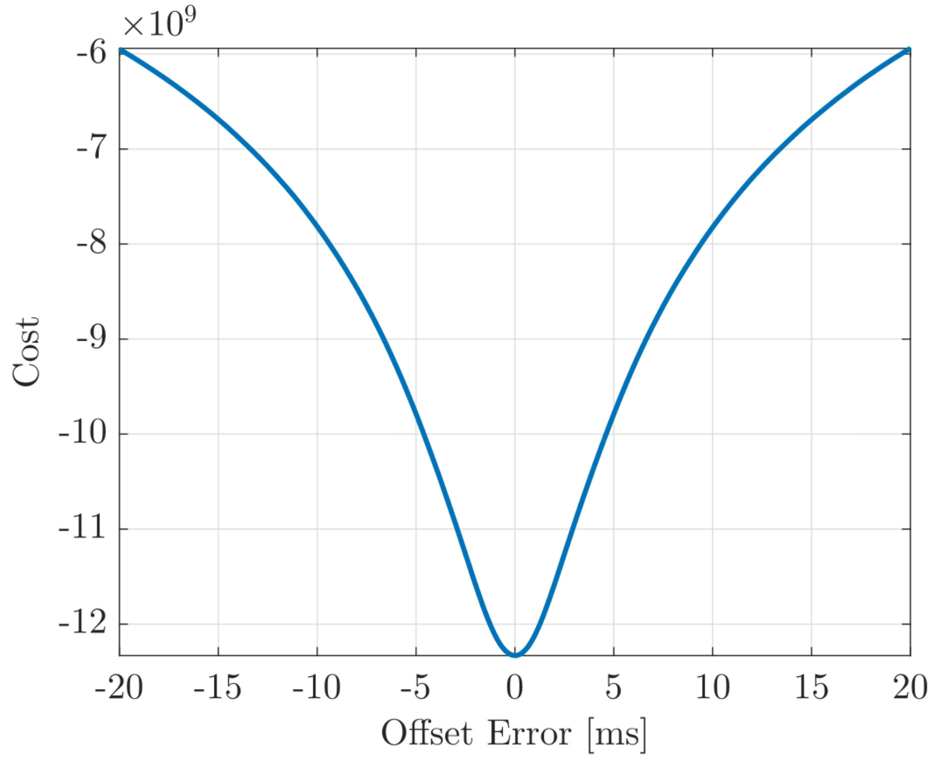
Future Work: Temporal Calibration

Sensor's internal time-delays result in a **temporal offset between data streams,** clearly affecting the reliability of the algorithm.

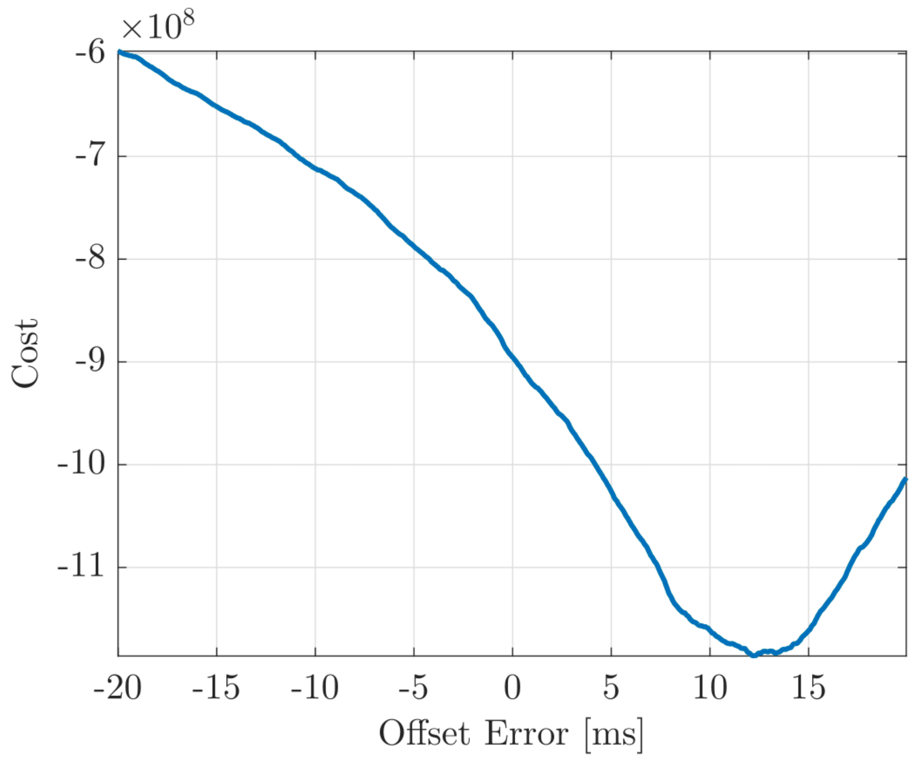
Can we adapt RQE calibration to include a temporal offset parameter?

One option: **pre-calibration through entropy minimization.**

Sim(3) parameters at **true values**



Sim(3) parameters away from **true values**



Simultaneous, **spatiotemporal calibration** could prove more reliable.

Conclusion

In this paper, we show that **RQE calibration** can recover the *Sim*(3) calibration parameters between 2D lidars and monocular cameras.

This appearance-base technique:

- calibrates lidars to a variety of egomotion sensors**,**
- operates in a broad range of structured environments,
- does not restrict sensor configuration,
- requires no preprocessing.

Future Work includes:

- implementing a fully spatiotemporal calibration algorithm,
- testing different sensor pairs,
- calibrating internal IMU parameters,
- parallelizing of cost function evaluation on GPUs,
- releasing an open-source implementation.

Thanks! Questions?

Email: **jacob.lambert@mail.utoronto.ca**
Website: **starslab.ca**