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
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.

Source: KITTI dataset
State of the Art: **Calibration in the Wild**

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


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


**Egomotion-based:** extrinsic calibration between egomotion sensors

RQE Calibration: Overview

Our approach is:

• appearance-based,
• recovers up to \( \text{Sim}(3) \) 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 \( \text{Sim}(3) \) 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 $\mathcal{F} \rightarrow L$ to $\mathcal{F} \rightarrow C$, $\Xi = [x_L \ y_L \ z_L \ \phi_L \ \theta_L \ \psi_L \ s]^T$.

Given: camera poses and associated covariance, 

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

2D lidar measurements and 

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

$$\mathbf{p}_{L_k}^{(n)} = \begin{bmatrix} x_k^{(n)} & y_k^{(n)} & 0 & 1 \end{bmatrix}^T.$$

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

Apply kinematic chain to transform point cloud to global frame:

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

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

Have: set of 3D points $\hat{\mathbf{x}}_{G,k}^{(n)} \in \hat{\mathbf{X}}$ each with covariance $\Sigma_k^{(n)}$, in global frame.
Key Concept: Entropy

We use Renyi Quadratic Entropy to optimize point cloud quality.

Information Theory

Shannon Entropy:

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

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

Statistical Mechanics

Gibbs Entropy:

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

Intuition: measures progress towards equilibrium, often implying uniformity.

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

$$H[P] = \frac{1}{1 - \alpha} \log \sum_{i=1}^{\Omega} p_i^\alpha$$
Derivation: Cost Function

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

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

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

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

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

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

Minimize entropy with respect to calibration parameters:

$$\Xi^* = \arg\min_{\Xi} H(\Xi | Y, \hat{X}).$$
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:

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

Simplifications:
- Ignore constants and monotonic logarithm,
- Remove double-counting,
- Store points in $kd$-tree and only consider points within some radius:
  $$\mathcal{N}(\hat{x}_i - \hat{x}_j, \Sigma_i + \Sigma_j + 2\sigma^2 \mathbf{I}) \approx 0 \text{ if } ||\hat{x}_i - \hat{x}_j|| \geq 2k \left(\max(\lambda_1(\Sigma_i), \lambda_1(\Sigma_j)) + \sigma^2\right)$$

Final cost function:

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

controls cost function accuracy versus computation time
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:

<table>
<thead>
<tr>
<th>Environment</th>
<th>$x$ [mm]</th>
<th>$y$ [mm]</th>
<th>$z$ [mm]</th>
<th>$\phi$ [deg]</th>
<th>$\theta$ [deg]</th>
<th>$\psi$ [deg]</th>
<th>Scale [$\times 10^{-3}$]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Room</td>
<td>2.8 (2.5)</td>
<td>3.1 (2.5)</td>
<td>5.2 (3.5)</td>
<td>0.22 (0.12)</td>
<td>0.051 (0.043)</td>
<td>0.24 (0.15)</td>
<td>0.33 (0.30)</td>
</tr>
<tr>
<td>Underground Parking Lot</td>
<td>4.5 (4.4)</td>
<td>4.8 (4.1)</td>
<td>5.2 (4.5)</td>
<td>0.37 (0.22)</td>
<td>0.11 (0.11)</td>
<td>0.37 (0.17)</td>
<td>1.2 (0.9)</td>
</tr>
<tr>
<td>Plane City</td>
<td>4.1 (2.1)</td>
<td>5.2 (4.2)</td>
<td>4.0 (3.9)</td>
<td>0.38 (0.23)</td>
<td>0.18 (0.06)</td>
<td>0.35 (0.23)</td>
<td>0.69 (0.55)</td>
</tr>
<tr>
<td>Quadratic Forest</td>
<td>4.6 (2.6)</td>
<td>3.9 (1.6)</td>
<td>2.9 (3.3)</td>
<td>0.32 (0.20)</td>
<td>0.074 (0.54)</td>
<td>0.35 (0.27)</td>
<td>0.73 (0.32)</td>
</tr>
<tr>
<td>Triangle Array</td>
<td>3.0 (1.9)</td>
<td>2.9 (1.6)</td>
<td>4.6 (3.5)</td>
<td>0.64 (0.60)</td>
<td>0.10 (0.07)</td>
<td>0.61 (0.58)</td>
<td>0.47 (0.26)</td>
</tr>
</tbody>
</table>
Experiments: Setup

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

• 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

Non-overlapping FOV

Point cloud before calibration

Point cloud after calibration

<table>
<thead>
<tr>
<th>Calibration results</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$ [mm]</td>
</tr>
<tr>
<td>---------------------</td>
</tr>
<tr>
<td>Initial Guess</td>
</tr>
<tr>
<td>Trial I</td>
</tr>
<tr>
<td>Trial II</td>
</tr>
<tr>
<td>Trial III</td>
</tr>
<tr>
<td>Trial IV</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>$\mu$ ($\sigma$)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Calibration results</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$ [mm]</td>
</tr>
<tr>
<td>---------------------</td>
</tr>
<tr>
<td>Initial Guess</td>
</tr>
<tr>
<td>Trial I</td>
</tr>
<tr>
<td>Trial II</td>
</tr>
<tr>
<td>Trial III</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>$\mu$ ($\sigma$)</td>
</tr>
</tbody>
</table>
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.
Entropy-Based $\text{Sim}(3)$ Calibration of 2D Lidars to Egomotion Sensors

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

MFI 2016, Baden-Baden, Germany