DARPA Sub-T: Lidar-Lidar Calibration

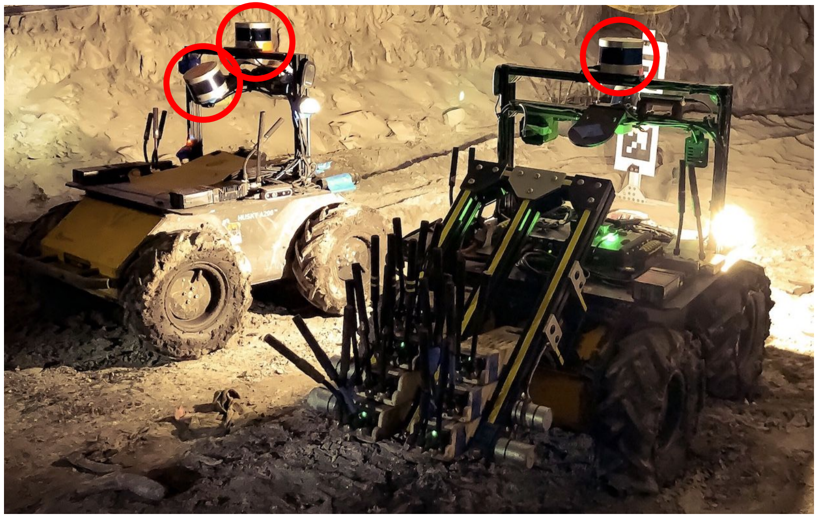
Dakota Wenberg, Aaron Ray, Ryan Sander

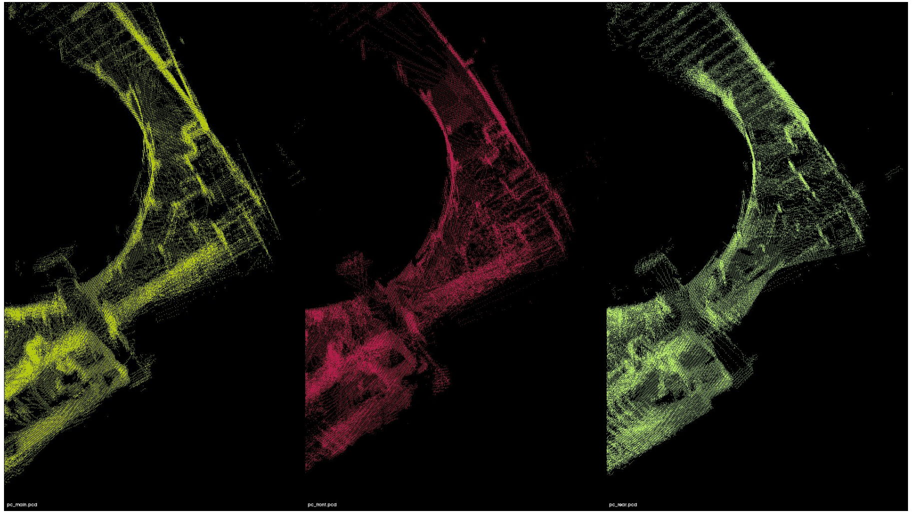
VNAV Final Project

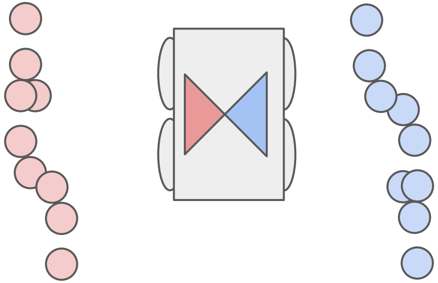
Problem Overview

- Build a consistent global point cloud

- If individual scans do not share keypoints, must **find lidar-lidar transform**







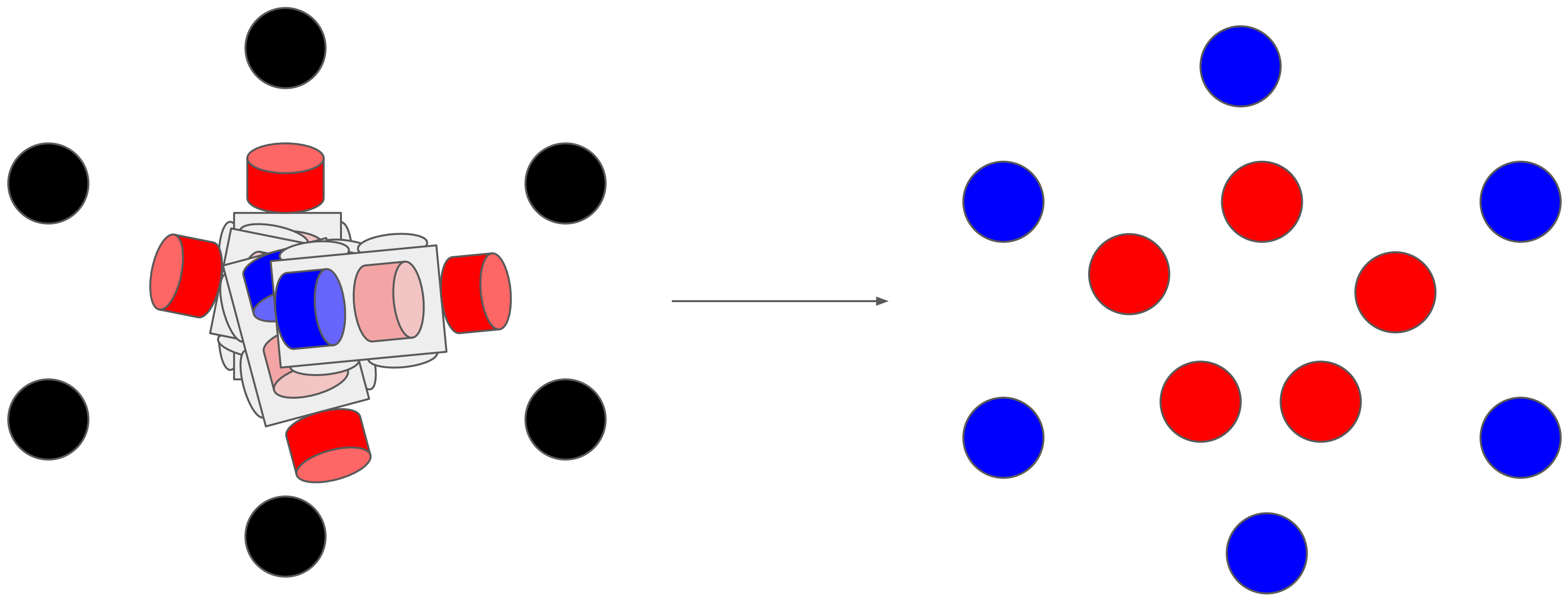
Clouds for each lidar may be disjoint

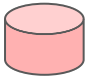
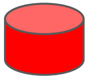
Preview

- How do we calibrate the Lidars? **Hand-Eye Calibration**
- **Solving** the Hand-Eye Calibration problem
- **Evaluating** calibration quality
- Next steps

Bad calibration example

- What happens to the global point cloud if the calibration is off?
  - Not just a rigid transformation!

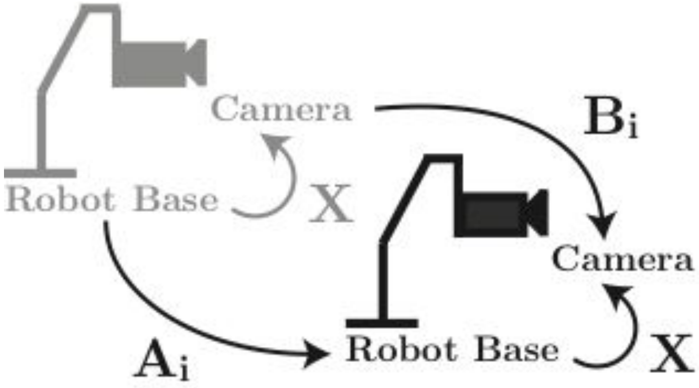


Robot thinks red lidar is at  but really it is at 

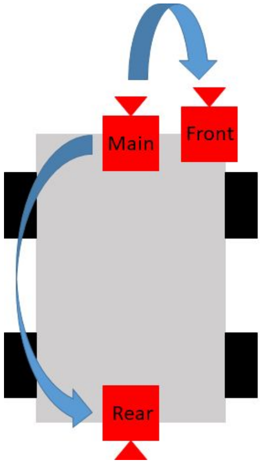
Resulting point clouds don’t match

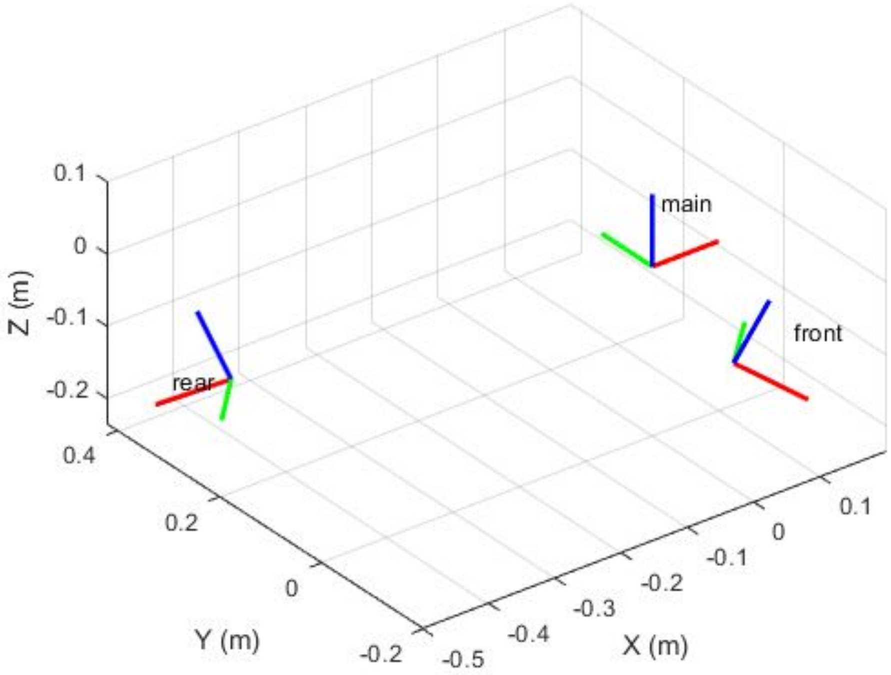
Our Hand-Eye Setup

“AX = XB”

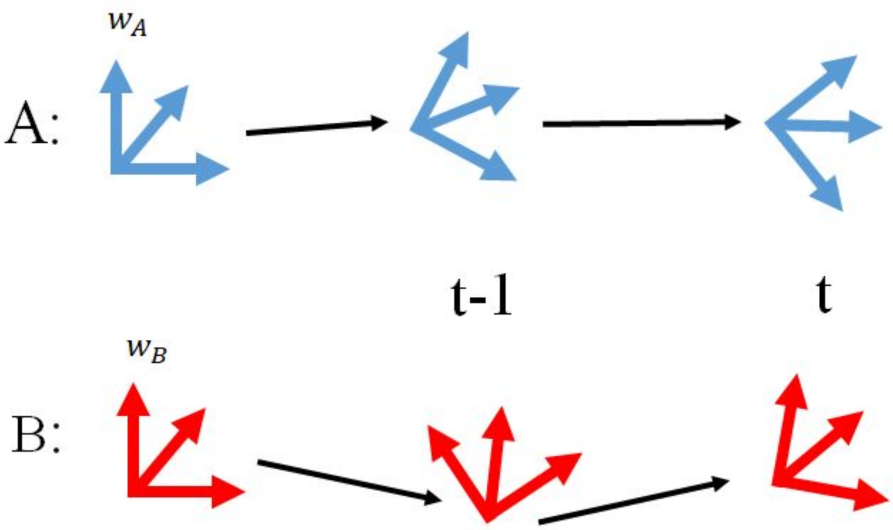


(from http://math.loyola.edu/~mili/Calibration/)





Hand-Eye Calibration



- Given: Two arbitrary trajectories

$$T_{A_t}^{A_{t-1}} = \left(T_{A_{t-1}}^{w_A}\right)^{-1} T_{A_t}^{w_A}$$

- Same for B

$$T_{B_t}^{B_{t-1}} = T_A^B T_{A_t}^{A_{t-1}} T_B^A$$

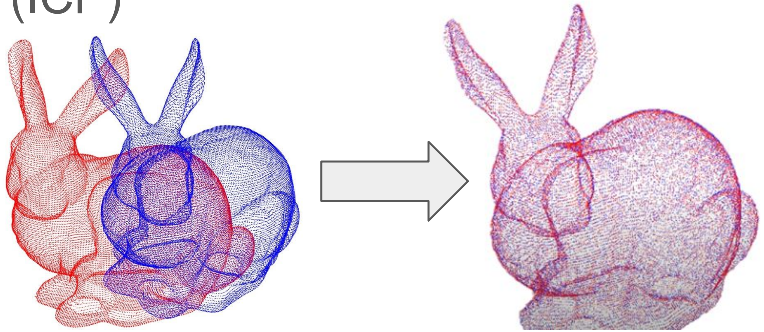
$$T_A^B = \left(T_B^A\right)^{-1}$$

$$\boxed{T_{B_t}^{B_{t-1}}} = \boxed{T_A^B}\,\boxed{T_{A_t}^{A_{t-1}}}\,\left(\boxed{T_A^B}\right)^{-1}$$

-Known -Unknown

ICP For Motion Estimation

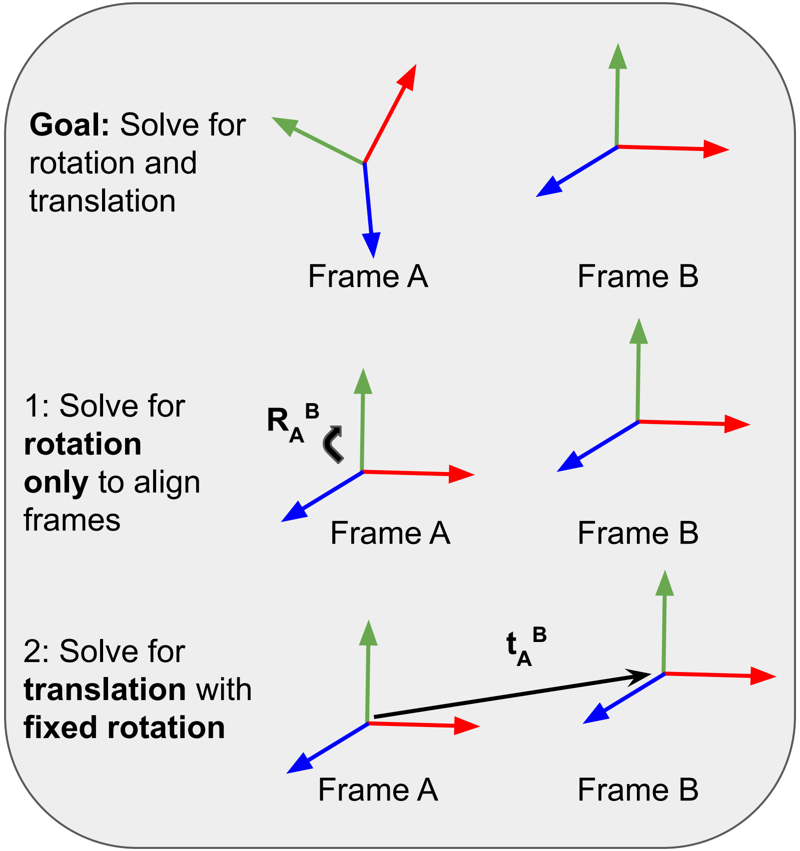
- AX = XB assumes we have measurements of motion (A and B). In practice, how do we get this? → Iterative Closest Point (ICP)
- Two point clouds are "best matched"
  - Point to point
- Rotation and translation is estimated
  - Minimizing point to point RMSE
- From this we can compute covariance
- **How do we use ICP?** We use the **variance** (diagonal) elements of the 6 x 6 ICP covariance matrix to eliminate certain data points whose variance exceeds a threshold amount.



(from https://www.youtube.com/watch?v=uz OCS_gdZuM&ab_channel=PhilippGlira)

Closed Form Solution-Based Methods

- Problem solved in closed form in two stages, as in Shiu and Ahmad [1] and Tsai and Lenz [2].
- **Advantages of these approaches:**
  - Does not require an iterative optimization procedure.
  - Closed-form solution may be easier to interpret intuitively.
- **Disadvantages of this approach:**
  - Many of these approaches fix rotation to solve for translation. This leads to errors in **rotation** affecting **errors** in translation.



Optimization-Based Solutions

- Ransac- esque sample consensus to find initial guess
- Nonlinear optimization to refine estimate
  - In our implementation, we focus on the local nonlinear optimization, because we have a good initial guess
- These optimization problems can be carried out using nonlinear techniques such as **Gauss-Newton** or **Levenberg-Marquadt**.

Unweighted Pose Estimation

- **Idea:** Minimize the **unweighted** sum of squared errors between our **pose estimate** and the **“round-trip” (chained) pose estimate**:

$$\begin{aligned}
\hat{\mathbf{T}}_A^B &= \arg \min_{\mathbf{T}_A^B \in \mathbf{SE}(3)} \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{B}_i \mathbf{T}_A^B \mathbf{A}_i^{-1} - \mathbf{T}_A^B||_F^2 \\
&= \arg \min_{\mathbf{T}_A^B \in \mathbf{SE}(3)} \frac{1}{N} \sum_{i=1}^{N} \mathrm{tr}(\mathbf{M}_i \mathbf{M}_i^T), \ \mathbf{M}_i = \mathbf{B}_i \mathbf{T}_A^B \mathbf{A}_i^{-1} - \mathbf{T}_A^B
\end{aligned}$$

Weighted Relative Pose Estimation (MLE Estimate)

- **Idea:** Some samples have more uncertainty than others in **rotation** and **translation.** To account for this uncertainty, we can assign less weight to our uncertain estimates using **rotation** and translation **variance estimates.**

$$\hat{\mathbf{T}}_A^B = \arg \min_{\mathbf{T}_A^B \in \mathbf{SE}(3)} \frac{1}{N} \sum_{i=1}^{N} r_i ||\mathbf{B}_i \mathbf{T}_A^B \mathbf{A}_i^{-1} - \mathbf{T}_A^B||_{\Omega_i}^2$$

$$= \arg \min_{\mathbf{T}_A^B \in \mathbf{SE}(3)} \frac{1}{N} \sum_{i=1}^{N} r_i \mathrm{tr}(\mathbf{M}_i \mathbf{\Omega}_i \mathbf{M}_i^T),$$

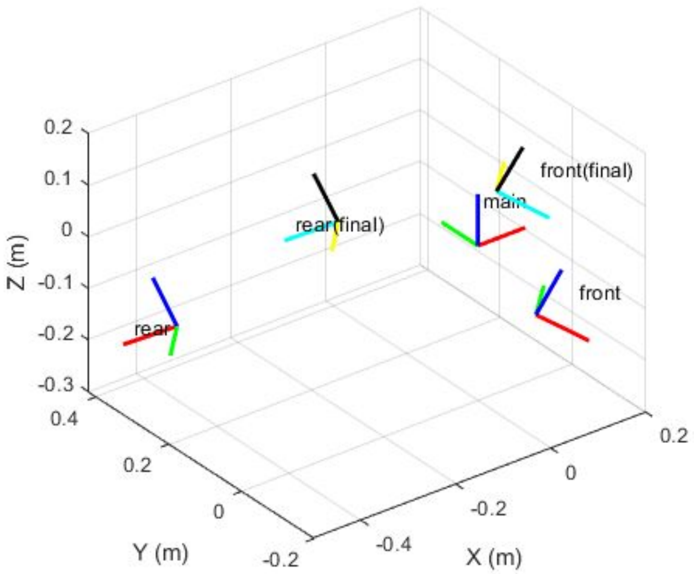
Where $\mathbf{M}_i = \mathbf{B}_i \mathbf{T}_A^B \mathbf{A}_i^{-1} - \mathbf{T}_A^B$, $\mathbf{\Omega}_i = \begin{bmatrix} \omega_i \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3^T & \rho_i \end{bmatrix}$

For some $\omega_i, \rho_i \in \mathbb{R}, r_i \in \{0, 1\}$

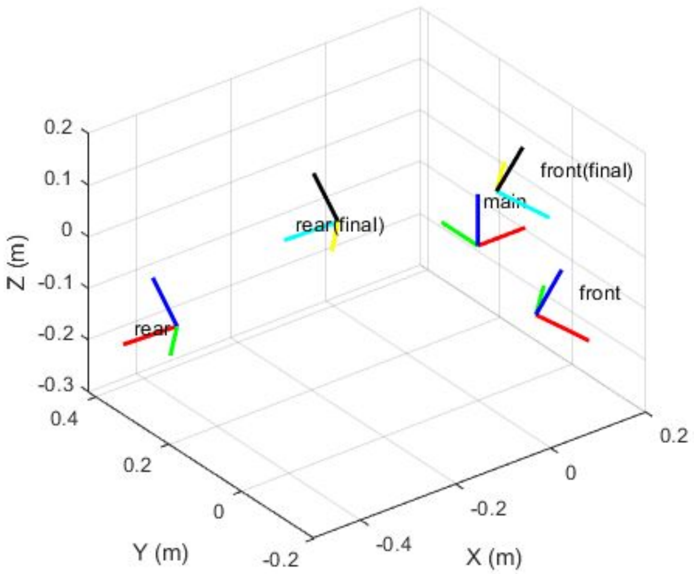
Dirty Details

- Time aligning
- Covariance
- ICP Covariance
- One of the lidars has occlusion on robot (can we tell numerically that the covariance of these ICP measurements is higher?)

Unweighted vs Weighted Calibration



Unweighted



Weighted

Evaluating Calibration Quality

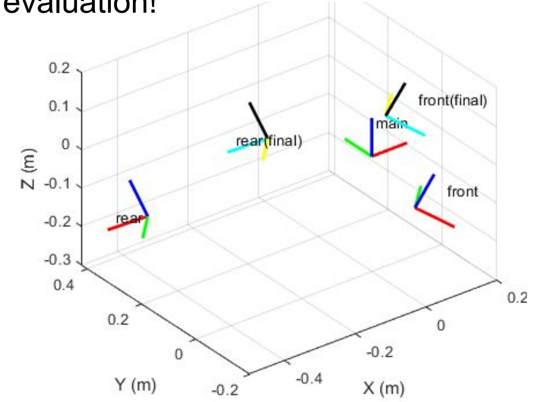
Can look at transform loop consistency over all data:

$$\hat{\mathbf{T}}_A^B = \arg \min_{\mathbf{T}_A^B \in \mathbf{SE}(3)} \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{B}_i \mathbf{T}_A^B \mathbf{A}_i^{-1} - \mathbf{T}_A^B||^2_{\Omega_i}$$

$$T_{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} ||\mathbf{B}_i \hat{\mathbf{T}}_A^B \mathbf{A}_i^{-1} - \hat{\mathbf{T}}_A^B||^2_{\Omega_i}}$$

Estimate	RMSE **R**, Weighted	RMSE **t**, Weighted	RMSE **R**, Unweighted	RMSE **t**, Unweighted
Initial ($\mathbf{T}_0$)	2.7793e-02	3.6588e-01	2.2885e-02	2.8327e-02
Final ($\mathbf{T}_F$)	2.7779e-02	3.6278e-01	2.2873e-02	2.8087e-02

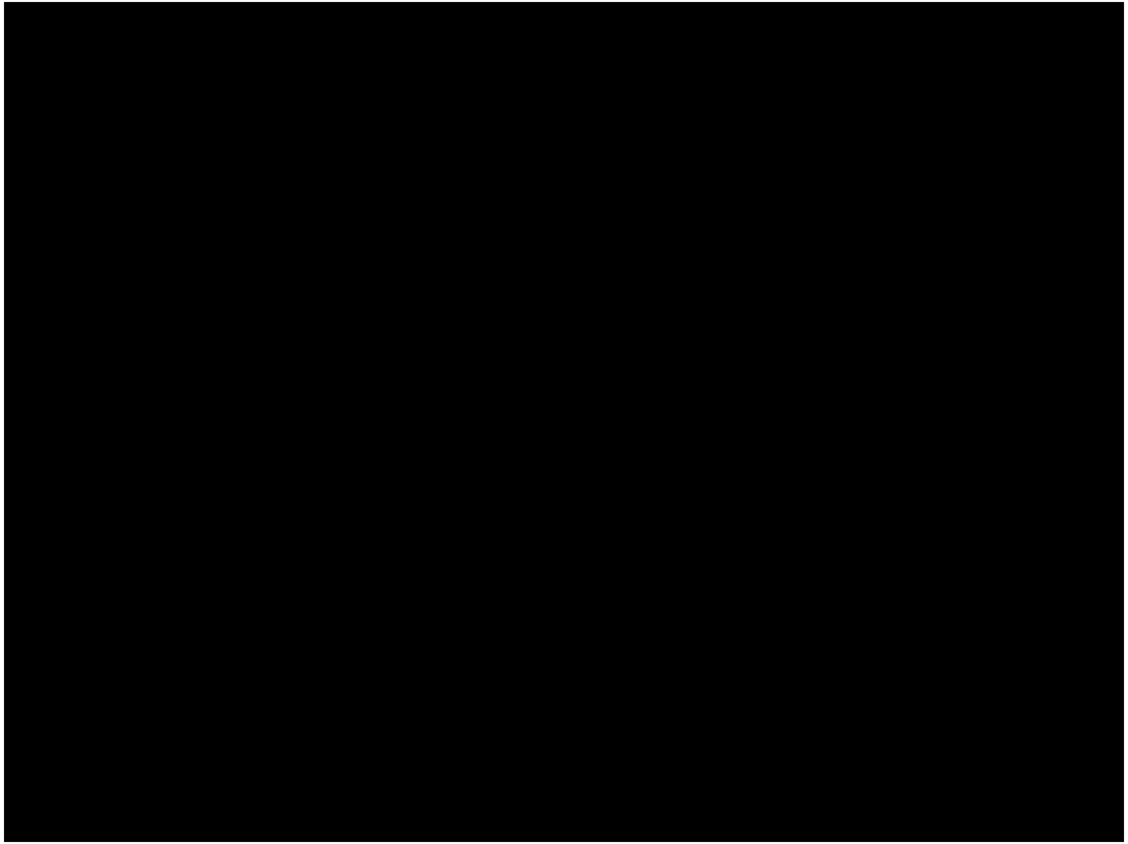
Choices about data weights affect optimization and evaluation!



Want an “external” method with separate assumptions from how we optimize

- Evaluate resulting point cloud

Building Point Clouds



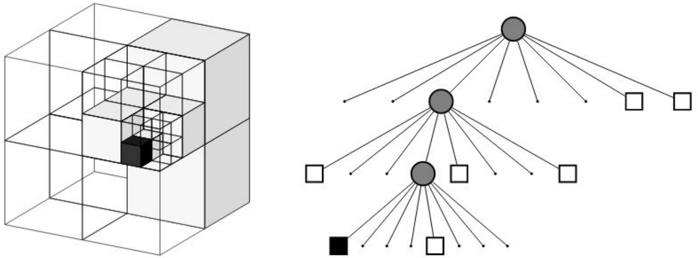


Fig. 2 Example of an octree storing free (shaded white) and occupied (black) cells. The volumetric model is shown on the left and the corresponding tree representation on the right.

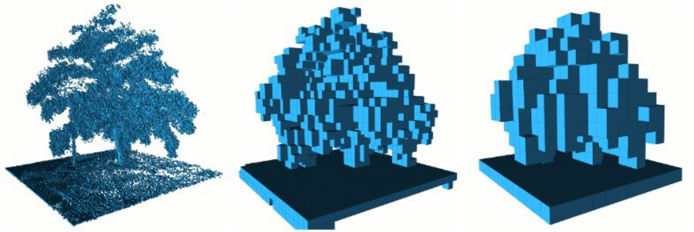
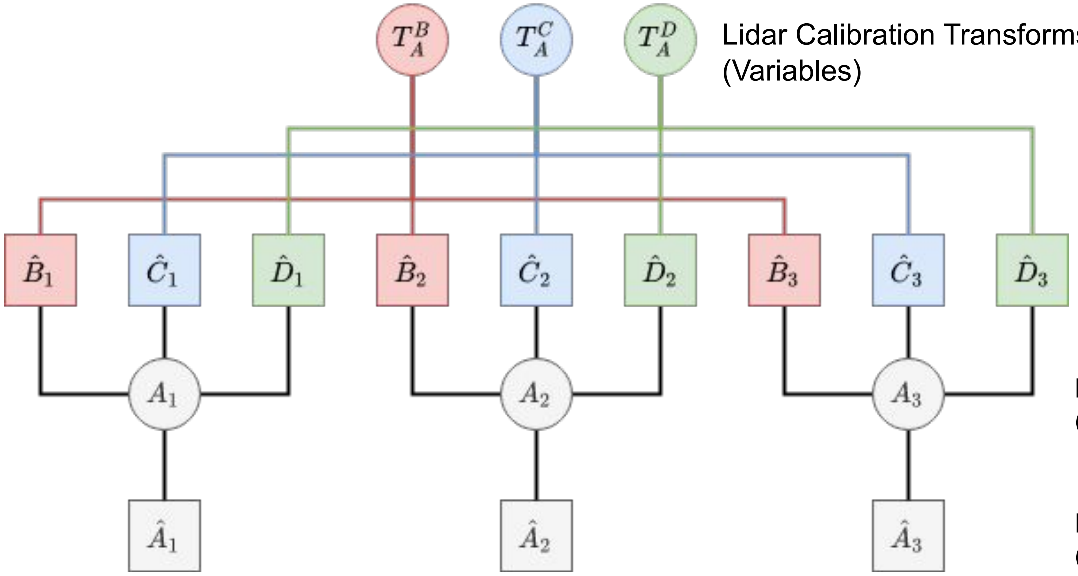


Fig. 3 By limiting the depth of a query, multiple resolutions of the same map can be obtained at any time. Occupied voxels are displayed in resolutions 0.08 m, 0.64 , and 1.28 m.

(A. Hornung et al, *"OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees" [4]*)

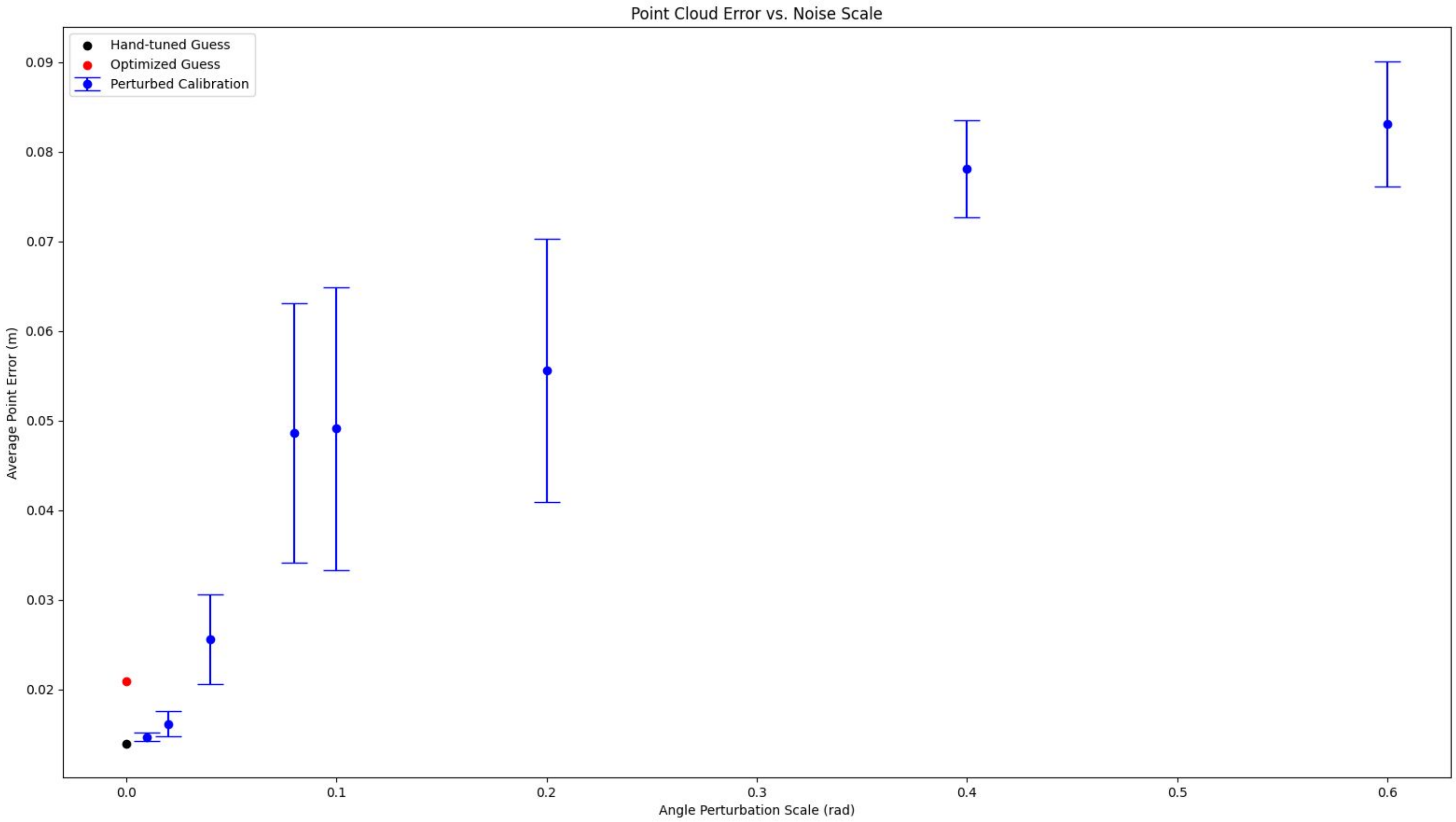
Improving Accuracy

- The manual calibration performed better on our point cloud metric
- Improve solution quality by **jointly estimating calibration and robot motion**
  - *Evaluation of Combined Time-Offset Estimation and Hand-Eye Calibration on Robotic Datasets*, Furrer, Fehr, Novkovic, Sommer, Gilitschenski, and Siegwart. 2017 [3].



Observed motion in each lidar frame is used to infer a single robot motion estimate at each timestep.

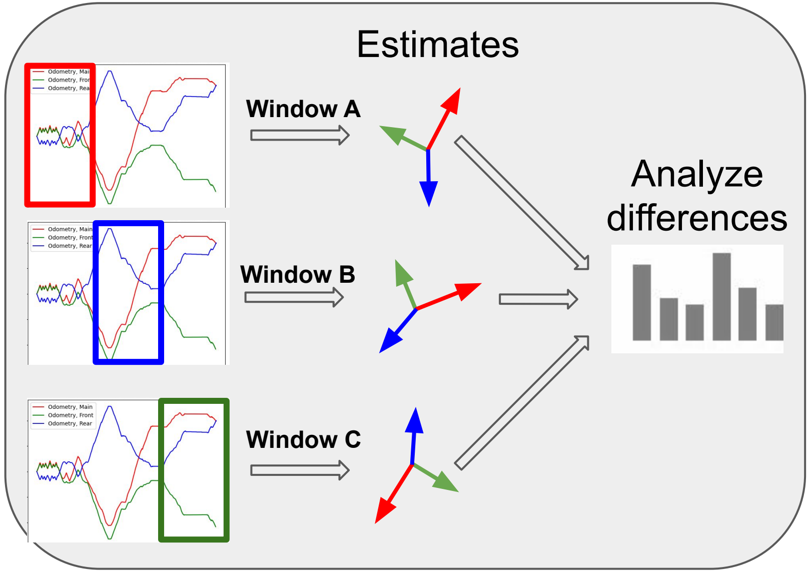
Point Cloud Accuracy Analysis



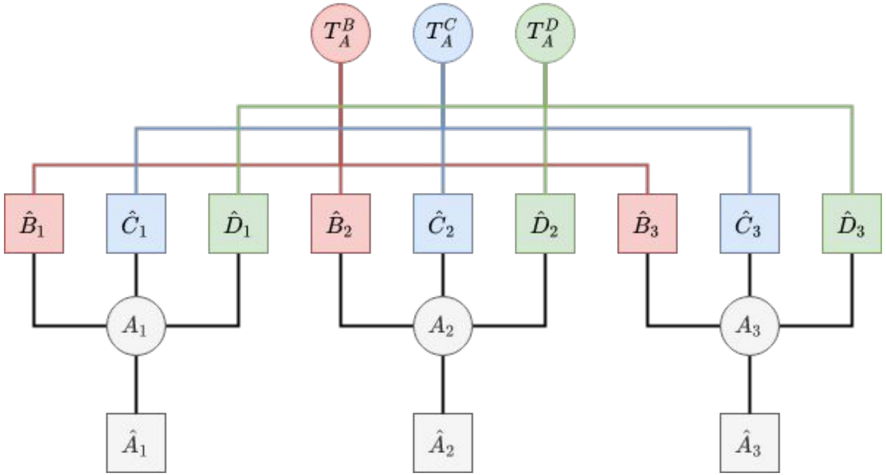
Future Work / Finishing Up

1. **Rolling average pose estimation** to see if transformation changes over time

2. Accuracy improvement through **joint** estimation of **rotation** and **translation**



[1]



[2]

Acknowledgements

Thank you to:

1. **Professor Luca Carlone and the VNAV staff** for their meaningful support and mentorship on this project and throughout the semester.
2. **Ben and Fadhil** for their insightful support with understanding our data from the DARPA Sub-T challenge.
3. **Mike and Phil** for collaborating with us on this project, enabling us to build a framework that complements their findings.

References

[1] Y. Shiu, S. Ahmad. “Calibration of Wrist-Mounted Robotic Sensors by Solving Homogeneous Transform Equations of the Form AX = XB.” In IEEE Transactions on Robotics and Automation, 5(1):16–29, 1989.

[2] R. Tsai, R. Lenz. “A New Technique for Fully Autonomous and Efficient 3D Robotics Hand/Eye Calibration.” In IEEE Transactions on Robotics and Automation, 5(3):345-358, 1989.

[3] Furrer, Fadri, et al. "Evaluation of combined time-offset estimation and hand-eye calibration on robotic datasets." Field and Service Robotics. Springer, Cham, 2018.

[4] Hornung, Armin, et al. "OctoMap: An efficient probabilistic 3D mapping framework based on octrees." Autonomous robots 34.3 (2013): 189-206.

Questions?

Appendix

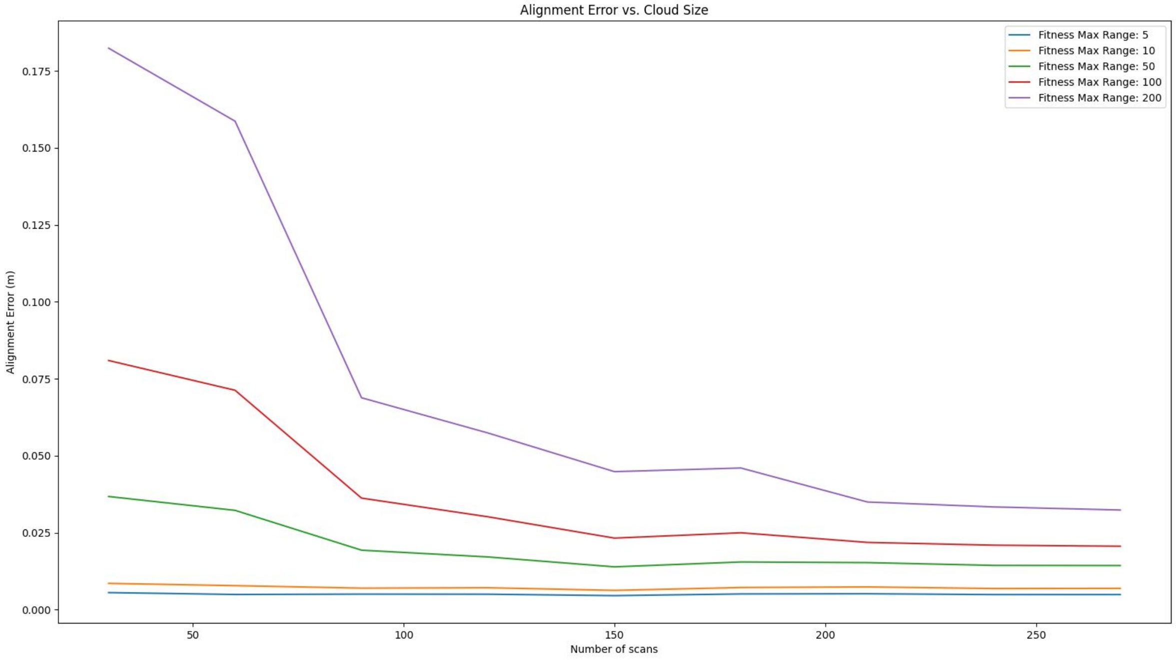
Computing Weights in Weighted Estimates

- Weights for **translation** ρ_i computed using the **translation inverse variance of the odometry data**.
- Weights for **rotation** ω_i computed using the **rotation inverse variance of the odometry data**.
- Rejection weights $\mathbf{r_i} \in \{0, 1\}$ indicate whether a sample should be rejected due to high uncertainty in the particularly sample, based off of whether: **max(diag(Σ_{ICP})) > rejection_threshold**.

Point Cloud Consistency

- Larger point clouds give more accurate validation

- Over long periods, drift in the map obscures alignment information



Accumulating Points

- How do we accumulate multiple scans into a single point cloud?

- Use an Octree!
  - Occupancy grid with multiple levels of resolution
  - Very efficient data structure
  - ROS integration with Octomap

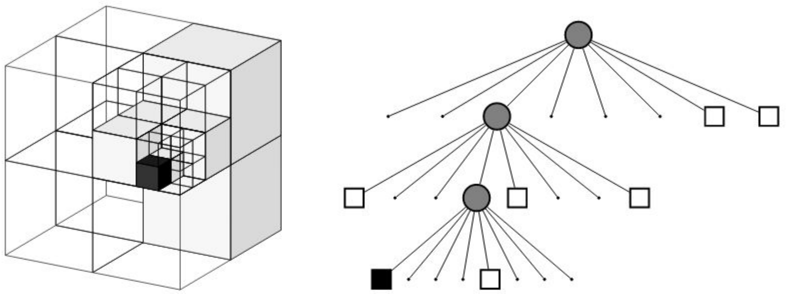


Fig. 2 Example of an octree storing free (shaded white) and occupied (black) cells. The volumetric model is shown on the left and the corresponding tree representation on the right.

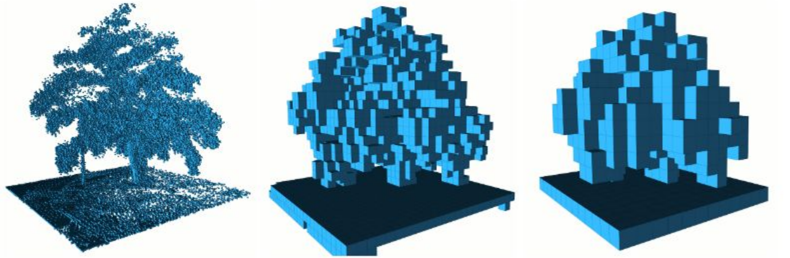


Fig. 3 By limiting the depth of a query, multiple resolutions of the same map can be obtained at any time. Occupied voxels are displayed in resolutions 0.08 m, 0.64 , and 1.28 m.

(A. Hornung et al, *"OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees"*)

SD Version of Movie