Internship Project Report

Matthew Ward

August 4, 2025

Introduction

- BYU Applied and Computational Math (DS & ML, April 2026)
- Biophysics Simulation Group—computer vision and competition dataset curation
- Music
- Handball
- Data engineer intern with you until August 16th!

Introduction

A few goals to discuss.

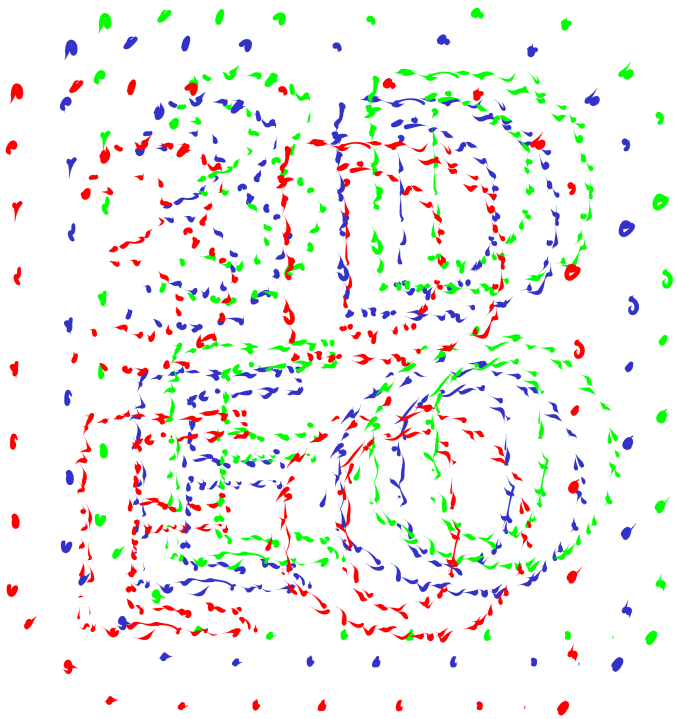
1. Infer global registrations from pairwise registrations
2. Model laser spot illumination as a Gaussian from sensor data
3. Generate a processing performance report

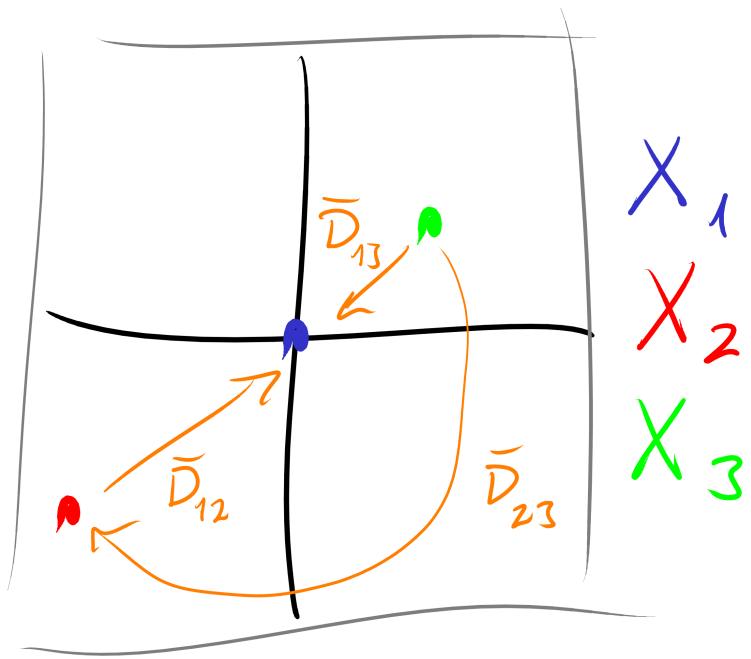
Global registration from pairwise registration

Global registration from pairwise registration

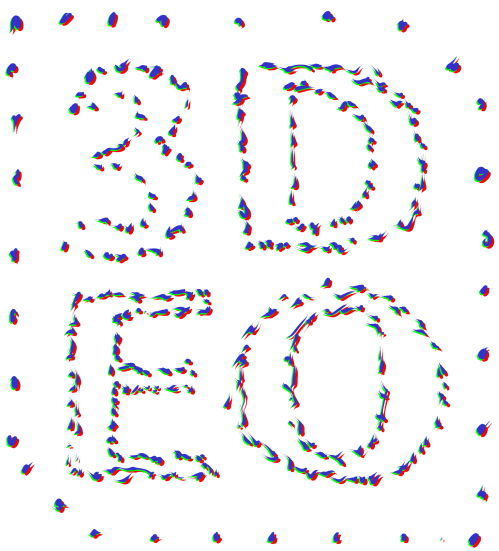
Straightforward to move one scan to align with another, less so to move n scans to align with each other.

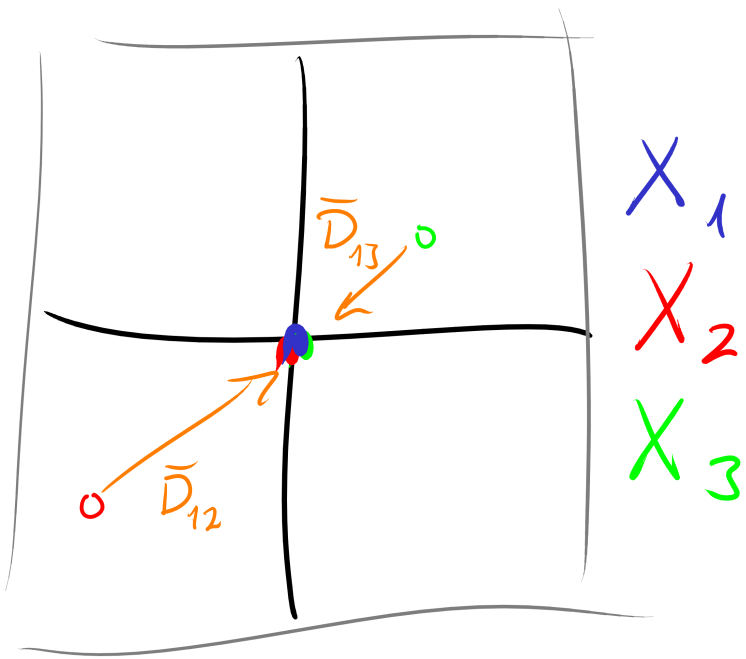
Global registration from pairwise registration



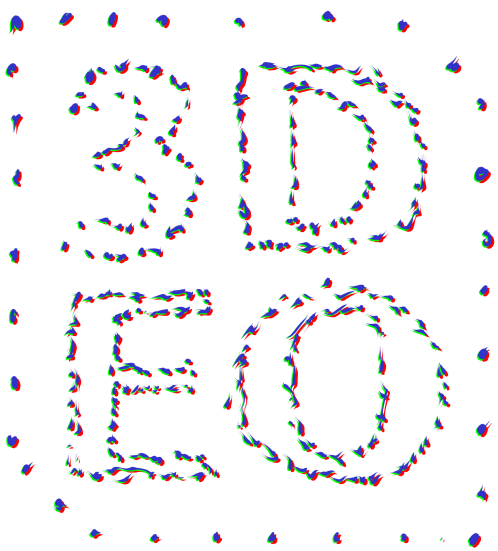


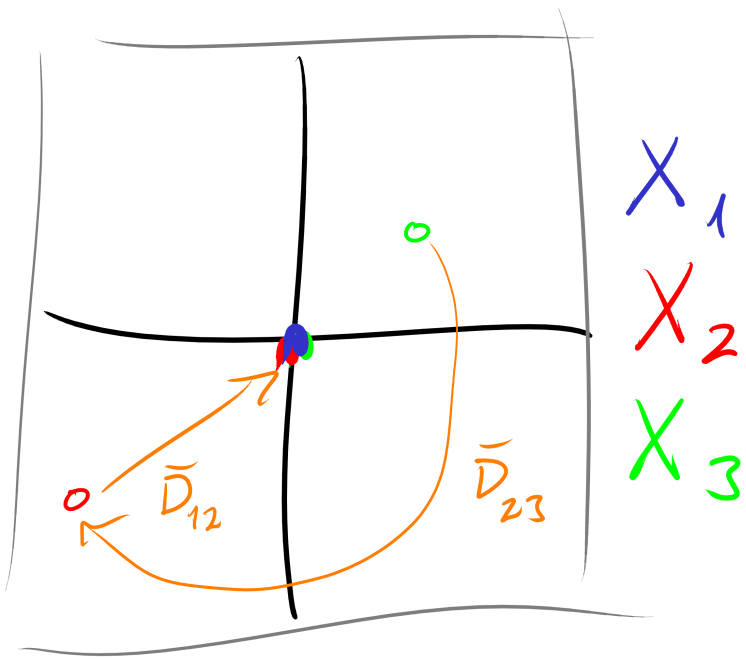
Global registration from pairwise registration



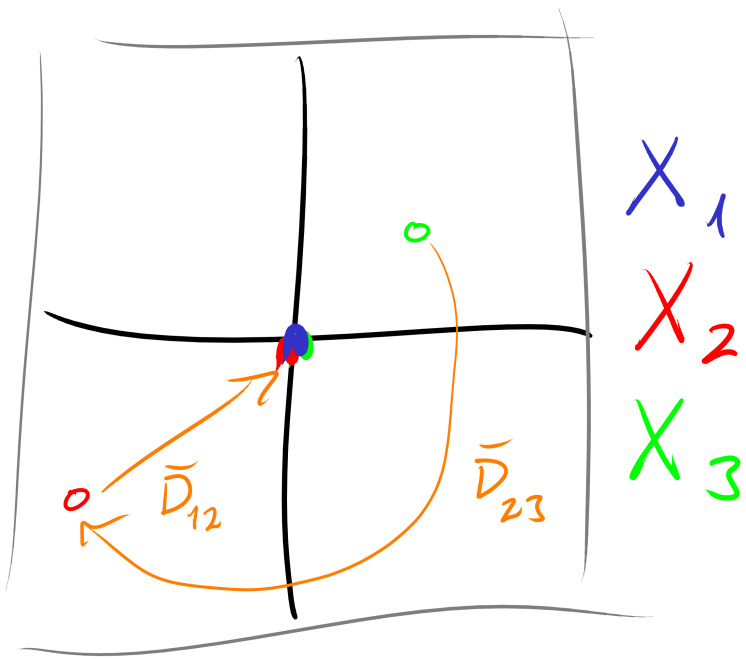


Global registration from pairwise registration

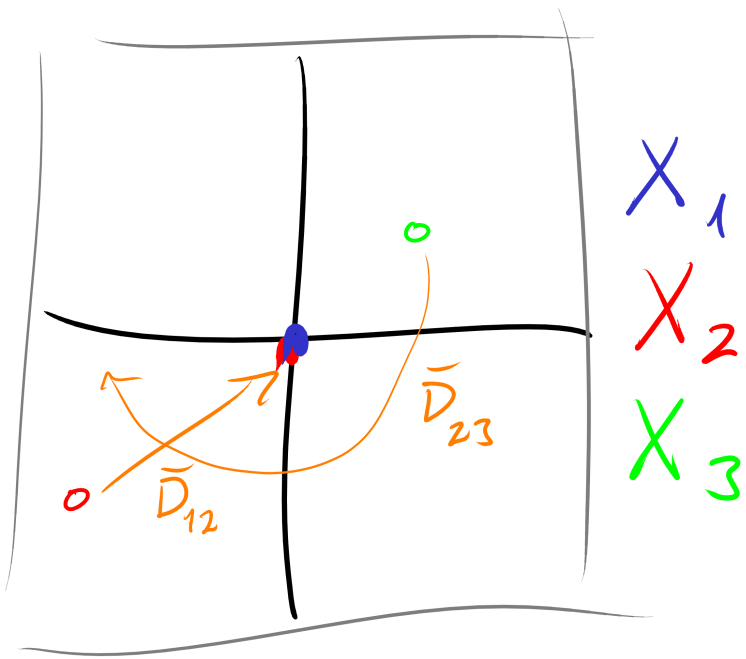




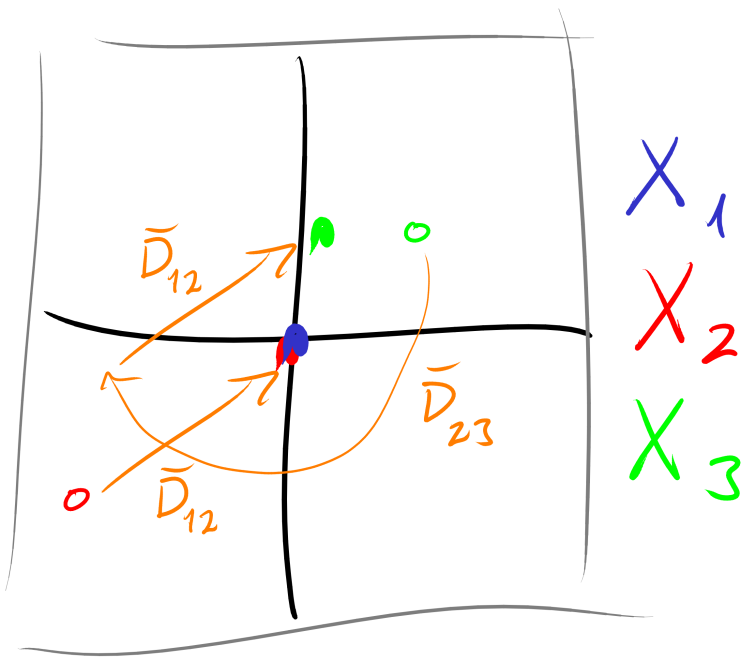
Global registration from pairwise registration



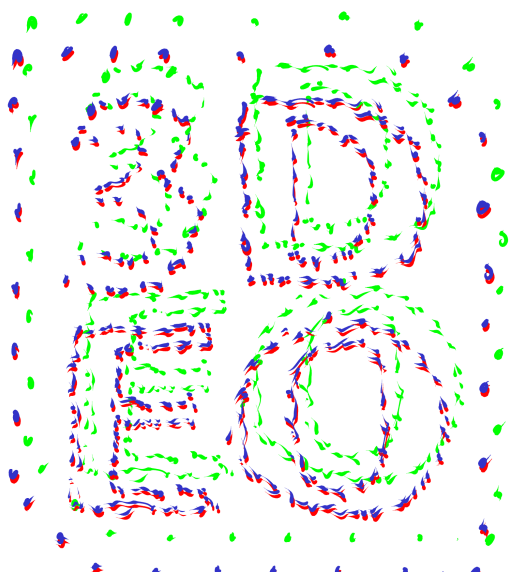
Global registration from pairwise registration

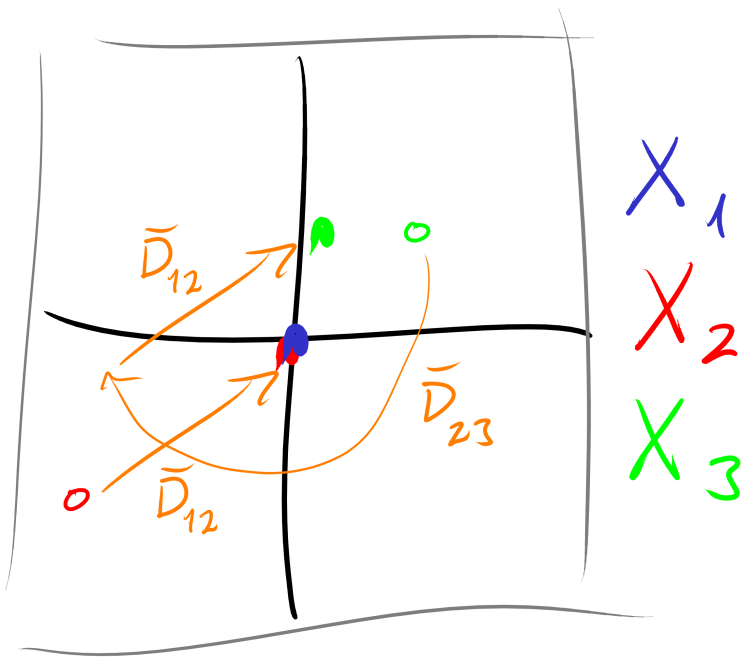


Global registration from pairwise registration



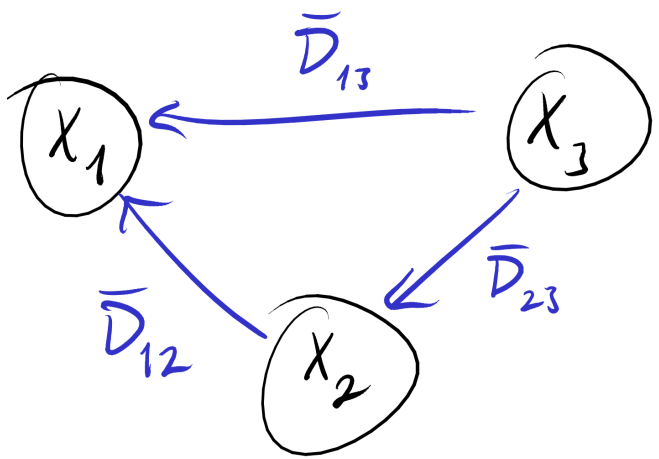
Global registration from pairwise registration





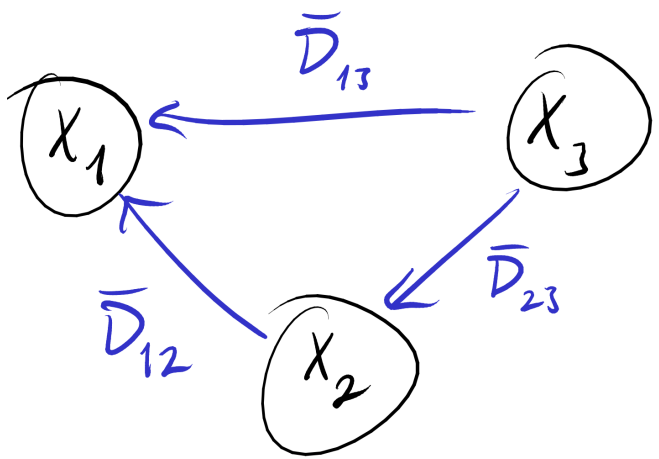
Global registration from pairwise registration

Straightforward to move one scan to align with another, less so to move n scans to align with each other.



Global registration from pairwise registration

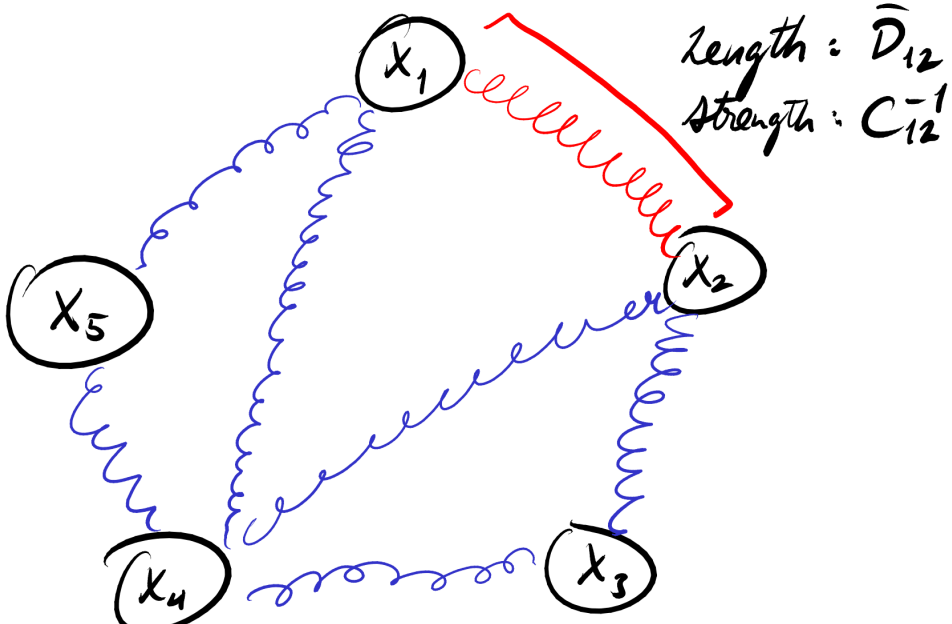
Straightforward to move one scan to align with another, less so to move n scans to align with each other.



Unfortunately, $\bar{D}_{13} \neq \bar{D}_{12}\bar{D}_{23}$.

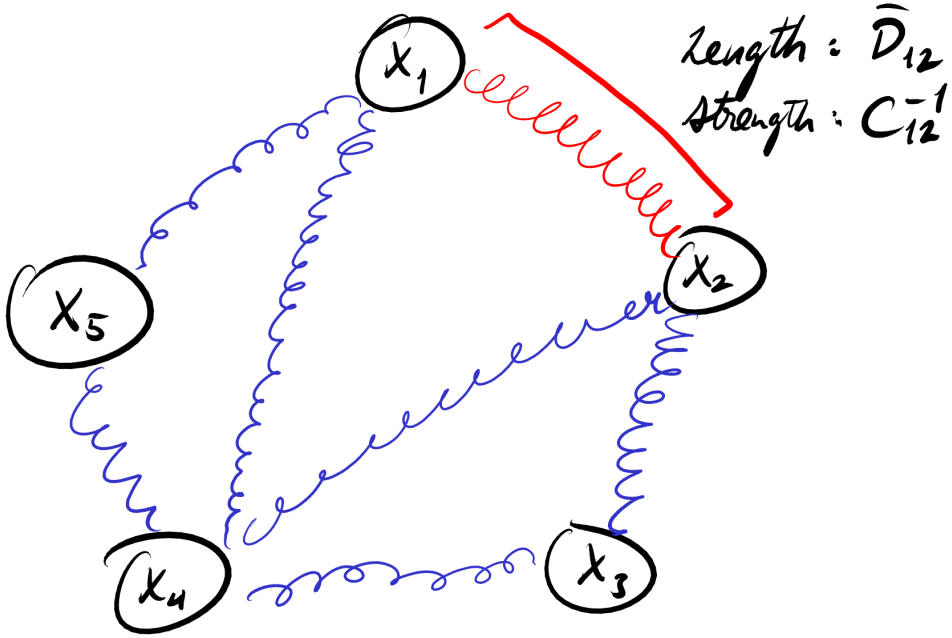
Global registration from pairwise registration

Pose graph optimization.



Global registration from pairwise registration

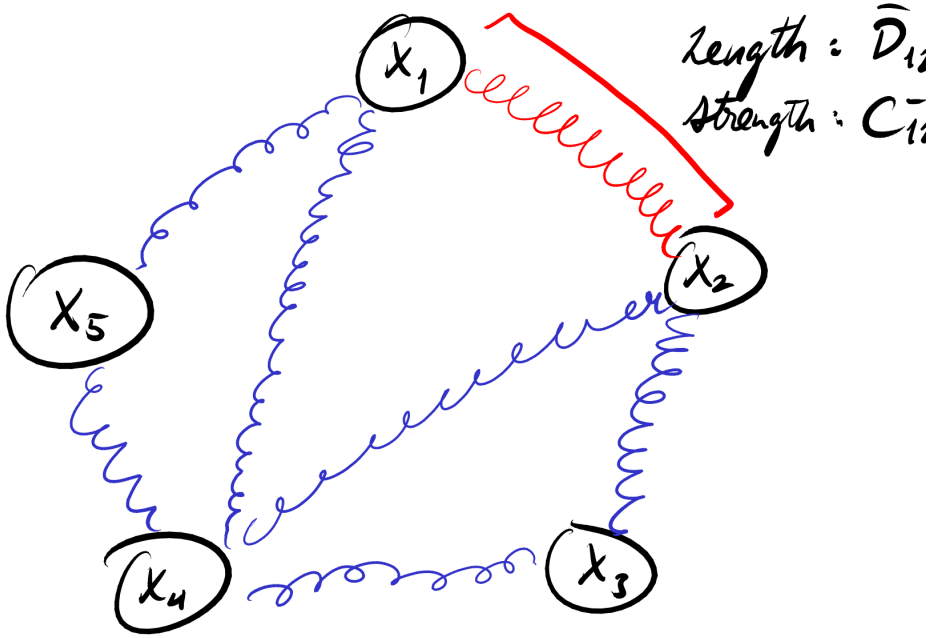
Pose graph optimization.



$$\underset{\{X_1,\ldots,X_n\}}{\text{minimize}} \quad \sum_{i,j} \left(\bar{D}_{ij} - \left(X_i - X_j\right)\right)^T C_{ij}^{-1} \left(\bar{D}_{ij} - \left(X_i - X_j\right)\right)$$

Global registration from pairwise registration

Pose graph optimization.



- Constant stretchiness
- Model stretchiness manually. $(i, j) \rightarrow C_{ij}^{-1}$
- Model stretchiness with machine learning (Optuna)

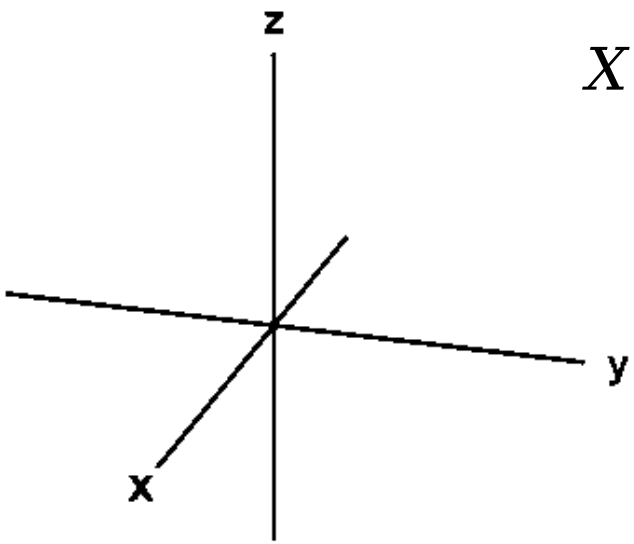
$$\underset{\{X_1,\ldots,X_n\}}{\text{minimize}} \quad \sum_{i,j} \left(\bar{D}_{ij} - \left(X_i - X_j\right)\right)^T C_{ij}^{-1} \left(\bar{D}_{ij} - \left(X_i - X_j\right)\right)$$

Global registration from pairwise registration

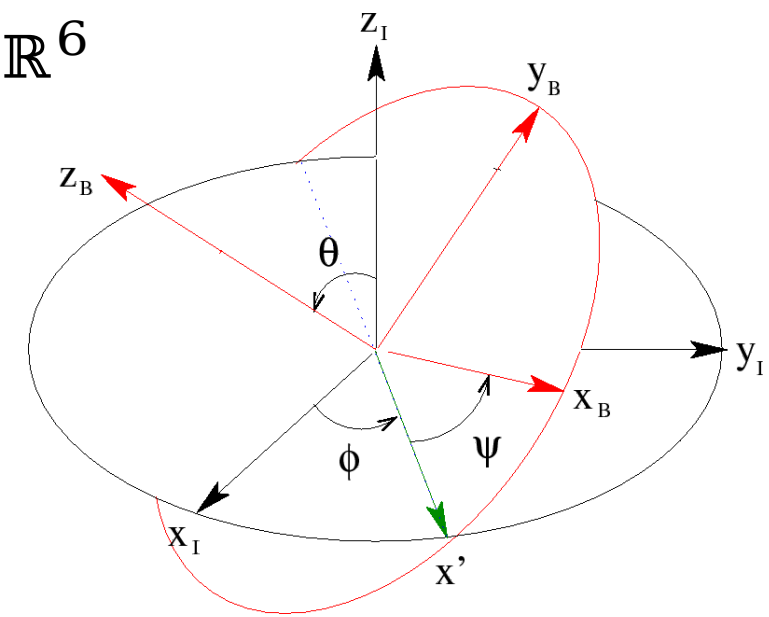
$$X_i = \begin{pmatrix} x_i \\ y_i \\ z_i \\ \theta_i \\ \varphi_i \\ \psi_i \end{pmatrix} \in \mathbb{R}^6$$

$$\underset{\{X_1,\dots,X_n\}}{\text{minimize}} \quad \sum_{i,j} \left(\bar{D}_{ij} - \left(X_i - X_j\right)\right)^T C_{ij}^{-1} \left(\bar{D}_{ij} - \left(X_i - X_j\right)\right)$$

Global registration from pairwise registration

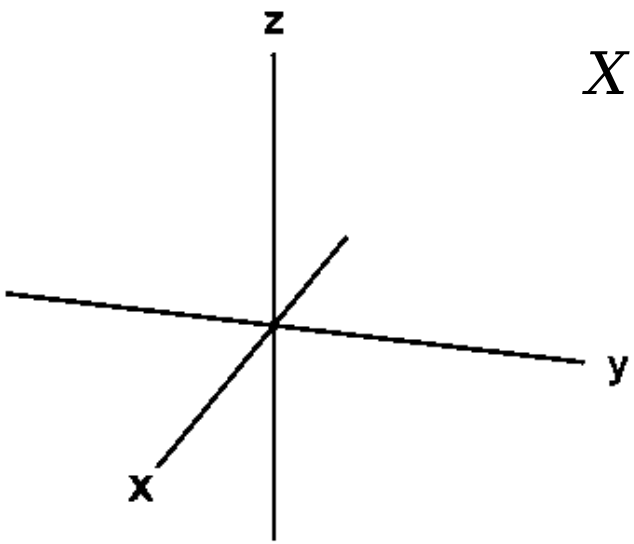


$$X_i = \begin{pmatrix} x_i \\ y_i \\ z_i \\ \theta_i \\ \varphi_i \\ \psi_i \end{pmatrix} \in \mathbb{R}^6$$

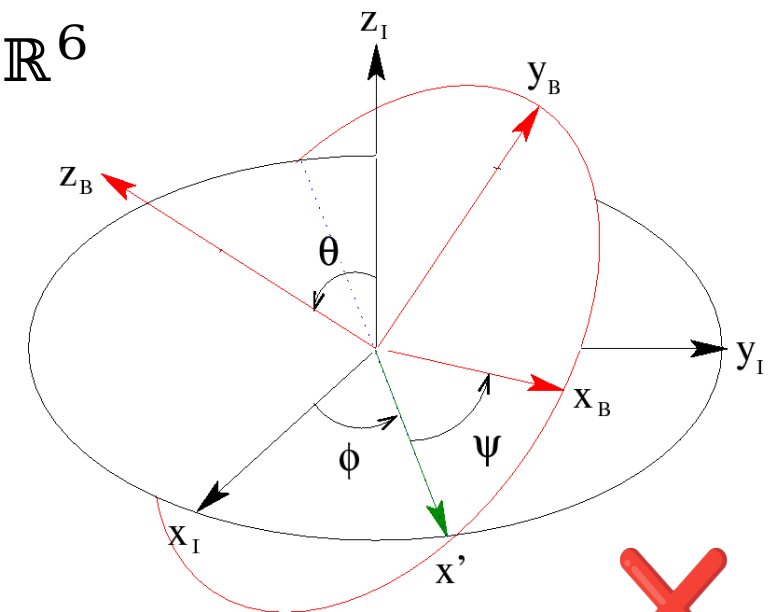


$$\underset{\{X_1,\ldots,X_n\}}{\text{minimize}} \sum_{i,j} \left(\bar{D}_{ij} - \left(X_i - X_j\right)\right)^T C_{ij}^{-1} \left(\bar{D}_{ij} - \left(X_i - X_j\right)\right)$$

Global registration from pairwise registration



$$X_i = \begin{pmatrix} x_i \\ y_i \\ z_i \\ \theta_i \\ \phi_i \\ \psi_i \end{pmatrix} \in \mathbb{R}^6$$

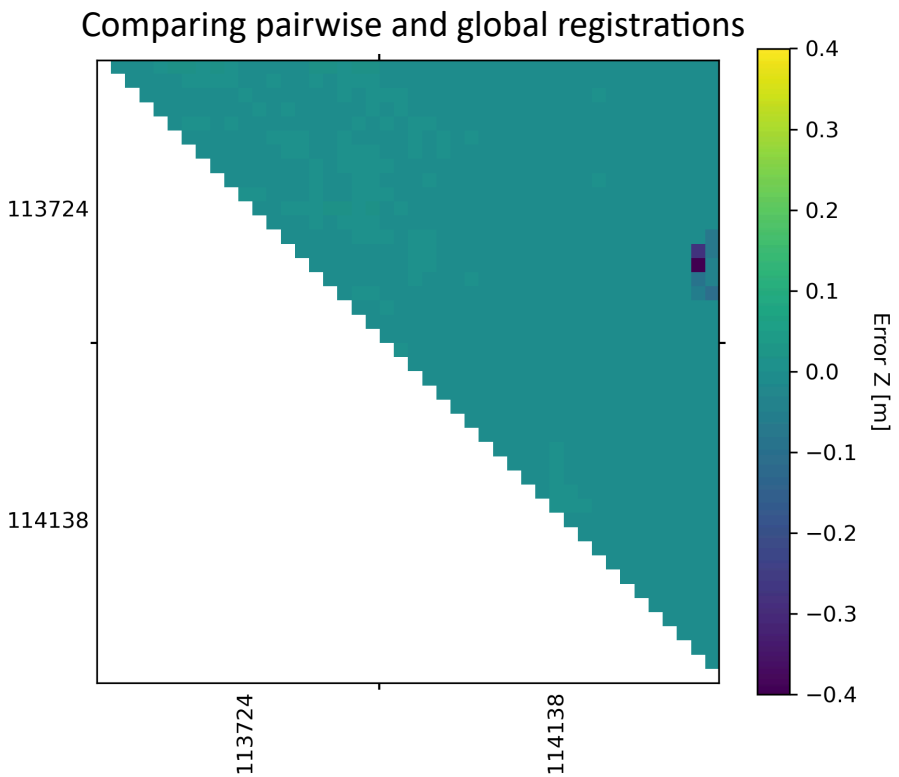


✅ ❌

$$\underset{\{X_1,\ldots,X_n\}}{\text{minimize}} \sum_{i,j} \left(\bar{D}_{ij} - \left(X_i - X_j\right)\right)^T C_{ij}^{-1} \left(\bar{D}_{ij} - \left(X_i - X_j\right)\right)$$

Global registration from pairwise registration

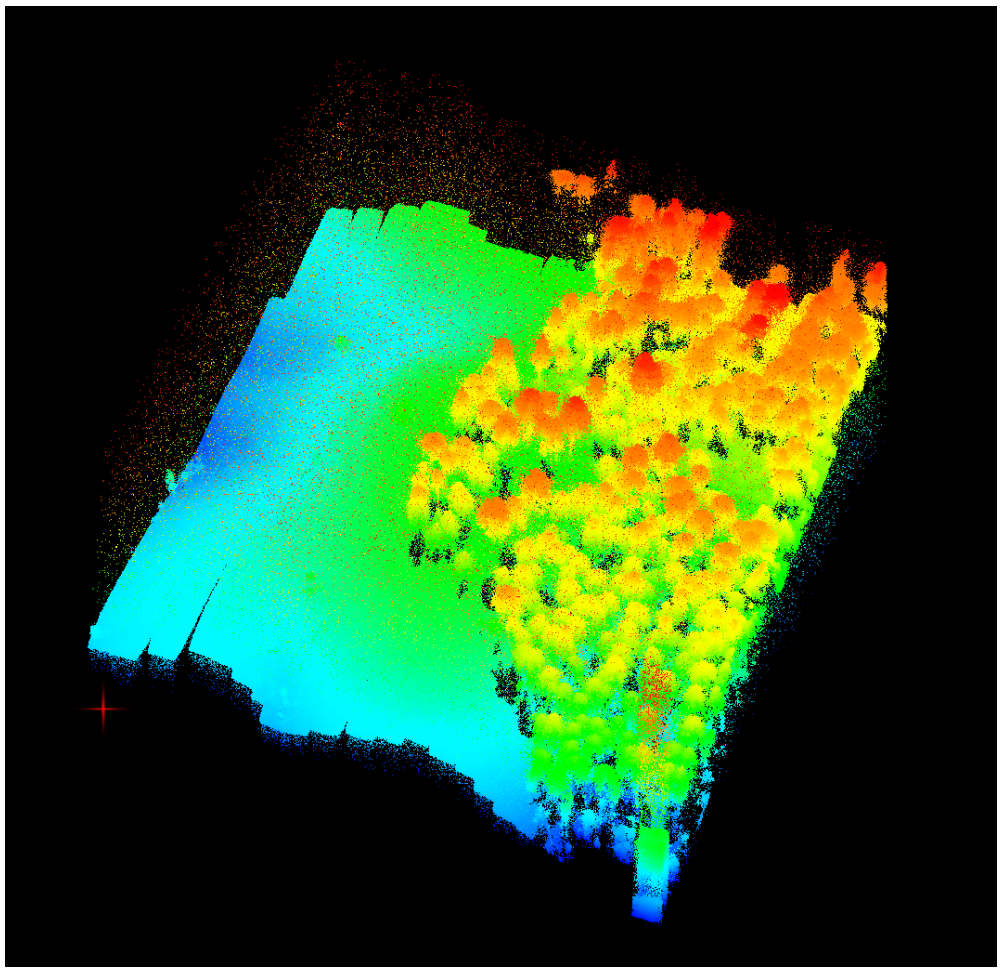
Pruning and weighting.



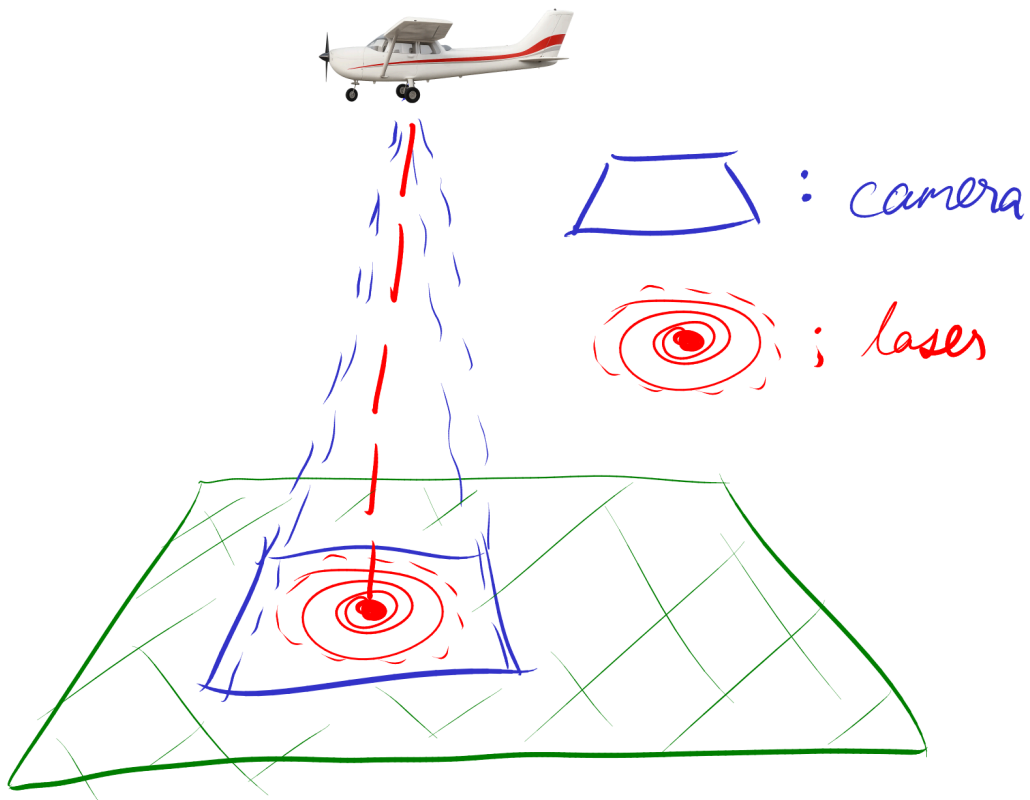
- After optimization, some springs (pairwise registrations) are stretched
- Lots of redundancy in the graph reveals poor pairwise registrations
- Weight those springs less in optimization step or remove them

Spot modeling

Analyzing 20230627_095732_cuchillo1_scan00091.bpf



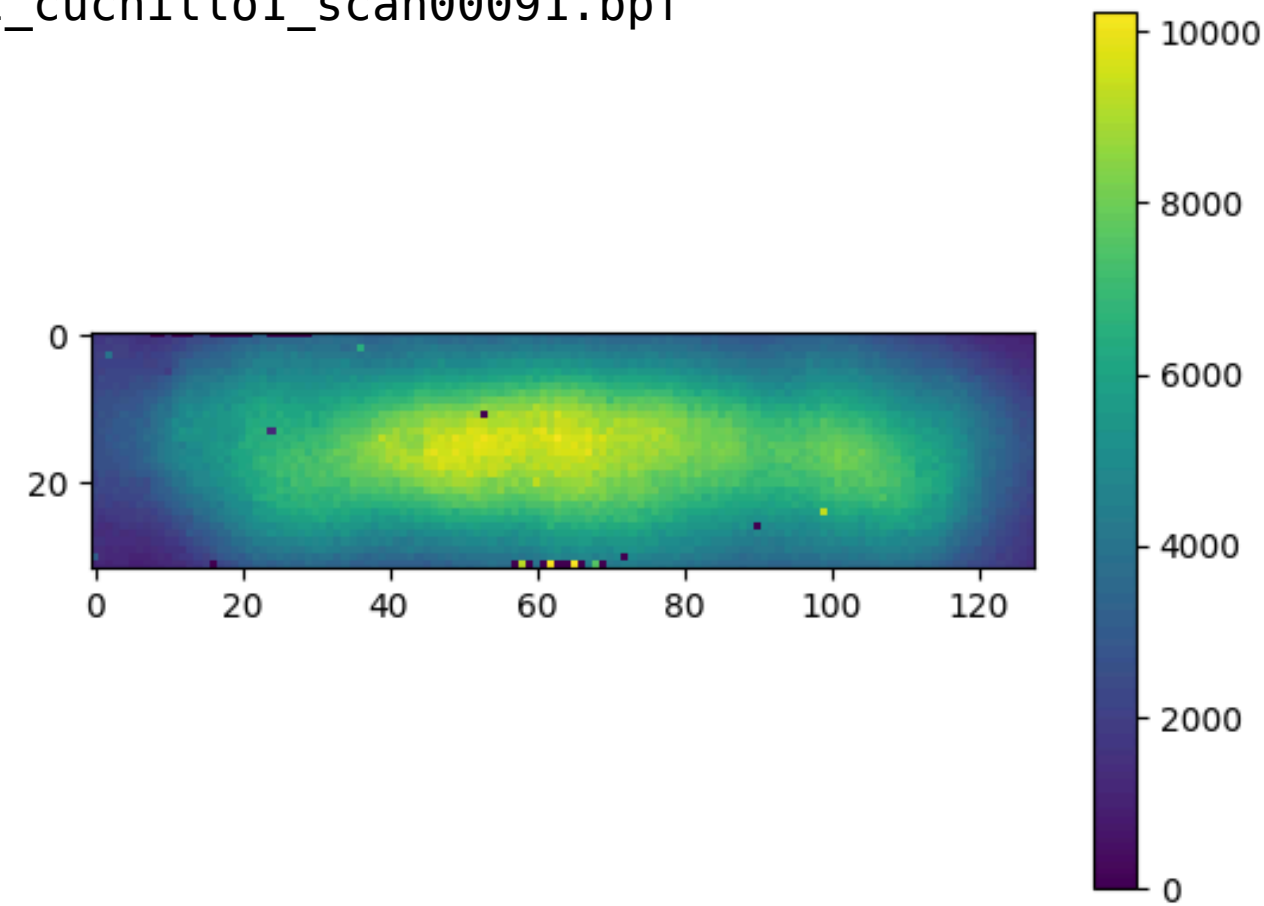
Spot modeling



Spot modeling

Total **photon detections per pixel** during

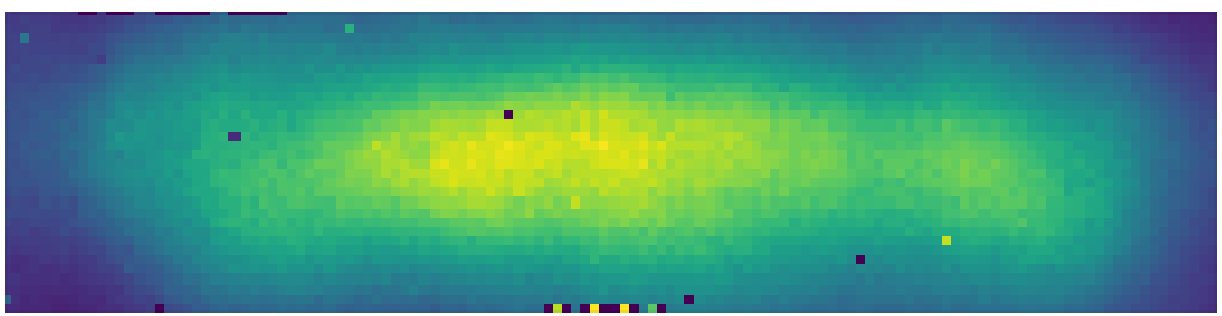
`20230627_095732_cuchillo1_scan00091.bpf`



Spot modeling

Analyzing `20230627_095732_cuchillo1_scan00091.bpf`

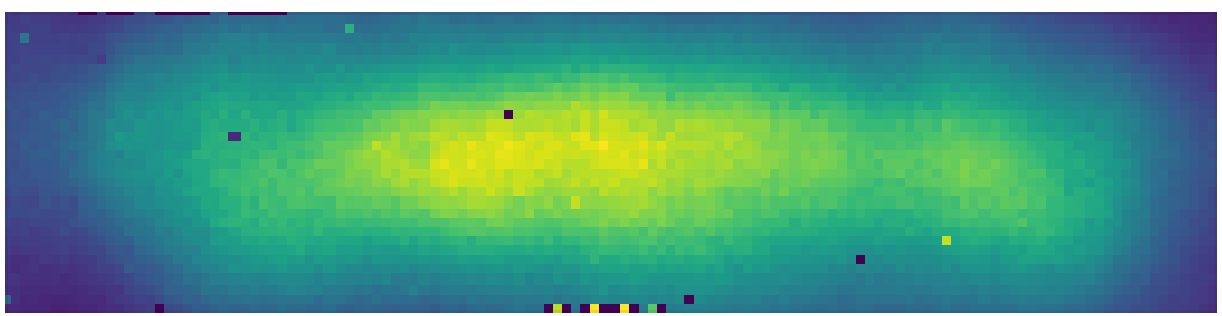
- 30 frame moving average pixel hitmap



Spot modeling

Analyzing `20230627_095732_cuchillo1_scan00091.bpf`

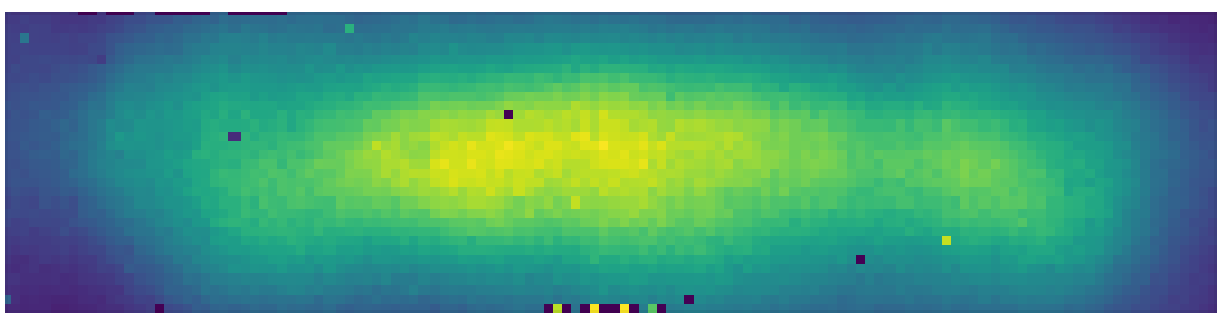
- 30 frame moving average pixel hitmap
- Model with Gaussian Mixture Model (1 component)



Spot modeling

Analyzing `20230627_095732_cuchillo1_scan00091.bpf`

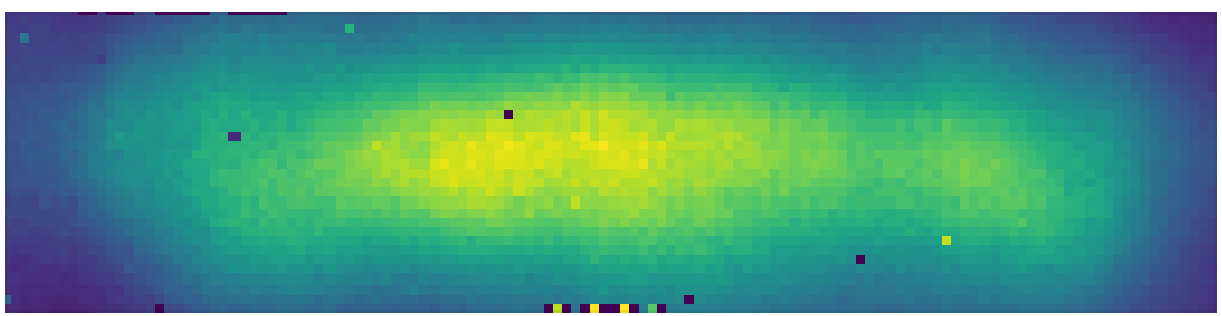
- 30 frame moving average pixel hitmap
- Model with Gaussian Mixture Model (1 component)
- Model with GMMis



Spot modeling

Analyzing `20230627_095732_cuchillo1_scan00091.bpf`

- 30 frame moving average pixel hitmap
- Model with Gaussian Mixture Model (1 component)
- Model with GMMis
- Model with GMMis, keep only 1% of detections



Spot modeling

Truncated multivariate normal likelihood

Questions v5 modeling

Matthew_Ward 2 May 22

Hello,

I am trying to fit the mean and covariance of a 2D normal distribution to some data with the complication that my data is truncated. I only have observations within a window, although the underlying distribution really is normal.

Some astronomers had this same problem (among others) while fitting GMMs and made an expectation maximization algorithm that I've successfully used to solve this problem (Filling the gaps: Gaussian mixture models from noisy, truncated or incomplete samples 1), but I'd like to try Bayesian inference as well.

As far as I understand, PyMC only has a *univariate* truncated normal (pymc.TruncatedNormal), so I'm trying to define my own multivariate truncated normal with constant bounds of truncation. I'm really struggling to implement the normalization constant part. Here's an example, where the third and fourth-to-last lines don't actually work since they use scipy to show what I'm imagining.

```
import os
import pymc as pm
from scipy import stats
import numpy as np
rng = np.random.default_rng()

lower_bounds = np.zeros(2)
upper_bounds = np.array([128, 32])
```

Processing performance report

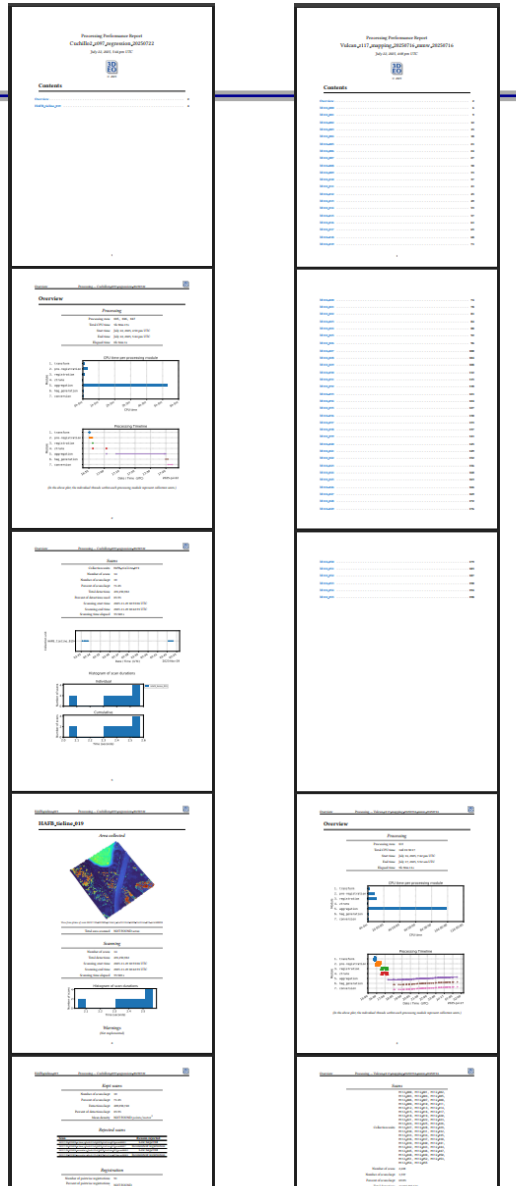
Not to be confused with sensor performance report!

Processing performance report

Not to be confused with sensor performance report!

A consise, readable .pdf report summarizing key insights from processing.

Located in `<acadia-output-directory>/qc/processing_report` after processing is complete



Processing performance report

Processing Performance Report

Vulcan_z117_mapping_20250716_mmw_20250716

July 22, 2025, 4:08 pm UTC

Contents

Processing performance report

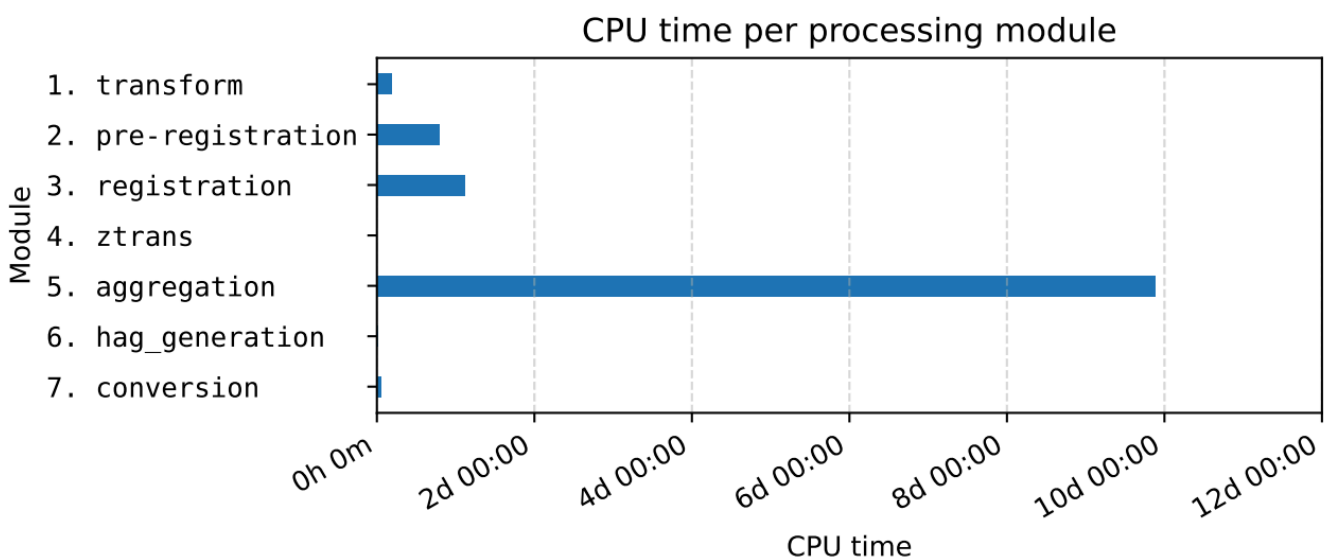
Scans

Collection units	M112_000, M112_001, M112_002, M112_003, M112_004, M112_005, M112_006, M112_007, M112_008, M112_009, M112_010, M112_011, M112_012, M112_013, M112_014, M112_015, M112_016, M112_017, M112_018, M112_019, M112_020, M112_021, M112_022, M112_023, M112_024, M112_025, M112_026, M112_027, M112_028, M112_029, M112_030, M112_031, M112_032, M112_033, M112_034, M112_035, M112_036, M112_037, M112_038, M112_039, M112_040, M112_041, M112_042, M112_043, M112_044, M112_045, M112_046, M112_047, M112_048, M112_049, M112_050, M112_051, M112_052, M112_053, M112_054, M112_055
Number of scans	2,228
Number of scans kept	1,537
Percent of scans kept	69.0%
Total detections	44,045,593,544
Percent of detections used	70.5%
Scanning start time	2025-05-29 11:48:24 UTC
Scanning end time	2025-05-29 11:56:27 UTC
Scanning time elapsed	0h 7m 54s

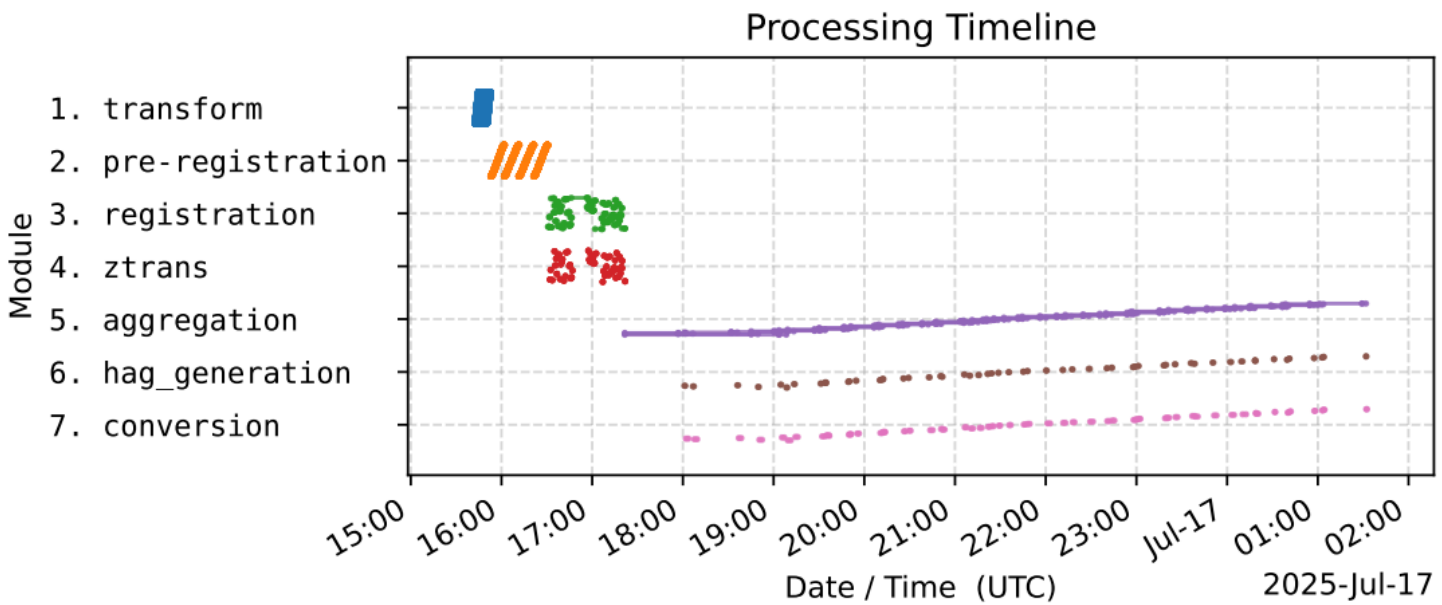
Processing performance report

Processing

Processing runs	631
Total CPU time	12d 01:30:17
Start time	July 16, 2025, 7:42 pm UTC
End time	July 17, 2025, 5:32 am UTC
Elapsed time	9h 50m 11s



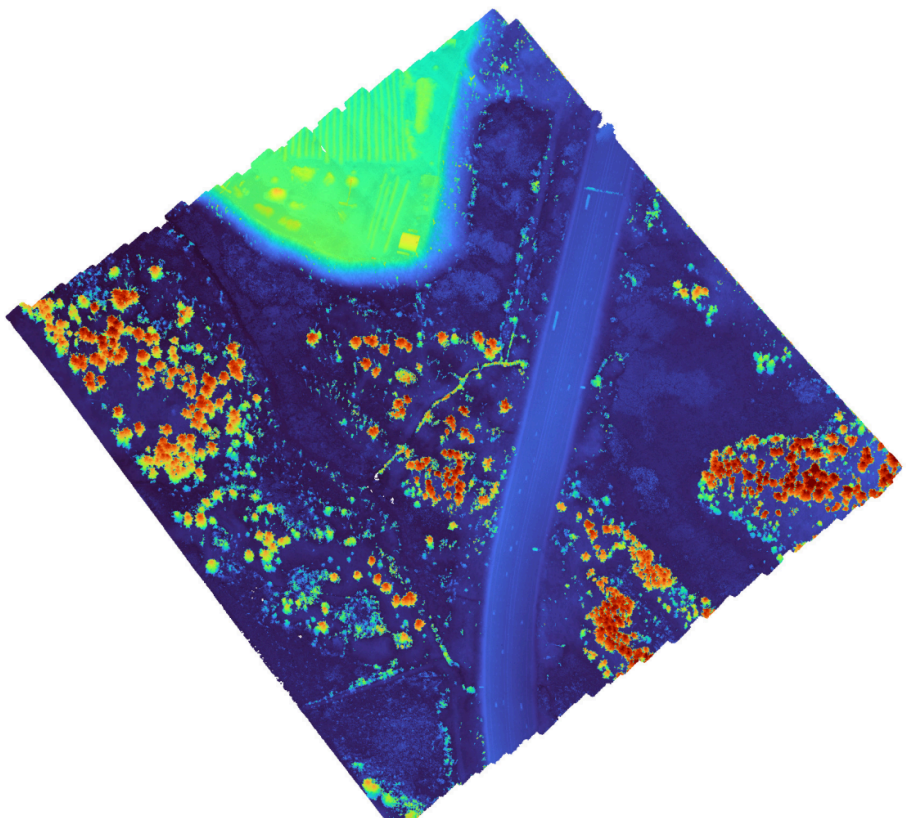
Processing performance report



(In the above plot, the individual threads within each processing module represent collection units.)

Processing performance report

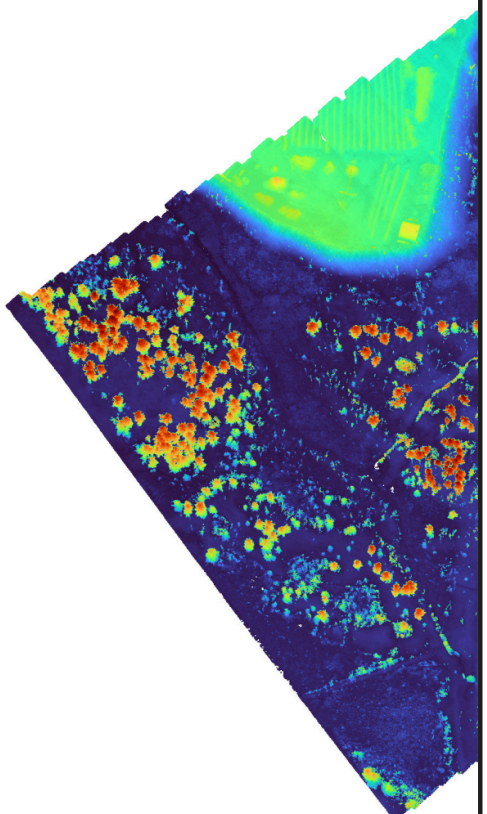
Area collected



View from plane of scan 20231129_023905_primary_Cuchillo2_HAFB_tieline_019_scan00059

Processing performance report

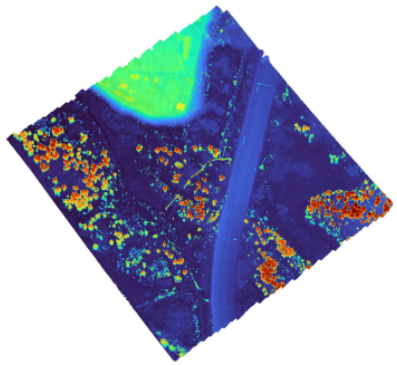
Area col



View from plane of scan 20231129_023905_primar

HAFB_tieline_019

Area collected



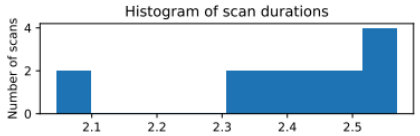
View from plane of scan 20231129_023905_primary_Cuchillo2_HAFB_tieline_019_scan00059

Total area scanned	NOT FOUND acres

Scanning

Number of scans	14
Total detections	491,250,982
Scanning start time	2023-11-29 02:33:06 UTC
Scanning end time	2023-11-29 02:42:35 UTC
Scanning time elapsed	33.540 s

Histogram of scan durations



Warnings

(Not implemented)

4

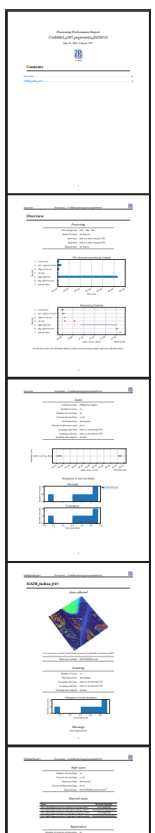
Processing performance report

Rejected scans

Scan	Reason rejected
20231129_023028_primary_Cuchillo2_HAFB_tieline_019_scan00052	Low target fill
20231129_023028_primary_Cuchillo2_HAFB_tieline_019_scan00057	Inconsistent registration
20231129_023028_secondary_Cuchillo2_HAFB_tieline_019_scan00052	Low target fill
20231129_023905_secondary_Cuchillo2_HAFB_tieline_019_scan00069	Inconsistent registration

Processing performance report

Single target report



Mapping report

Processing performance report

Processing directory

say, `albert:/shares/processed/cuchillo/flightData/FlatCreek`

↓

.json of filepaths and statistics

to be used in report

↓

LaTeX report

↓

.pdf report

References

Global from pairwise registration

- zreg_ncc/python (master) on Bitbucket
- Presentation I gave on pose graph registration
- Ideas related to the above presentation
- Lu and Milios paper "Globally Consistent Range Scan Alignment for Environment Mapping", April 1997. Introduces these ideas in the context of robotics
- Borrmann et al. paper "The Efficient Extension of Globally Consistent Scan Matching to 6 DoF", June 2008. Extends concepts in the above paper to 3D. We aren't using this paper's ideas, but it helped me understand what was going on better

Laser spot modeling

- Write-up Describes background and my work thus far on this
- I have the git repo, but the code is unfinished, so I haven't pushed to Bitbucket. Should I push?

Processing performance report

- processing-performance (master) on Bitbucket
- acadia (master) on Bitbucket Apptainers in apptainer_creation and slurm bricks in processing_workflow
- python_3deo/fileIO/readGSOF.py Used to collect certain scanning information