



---

# 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 16<sup>th</sup>!

# 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

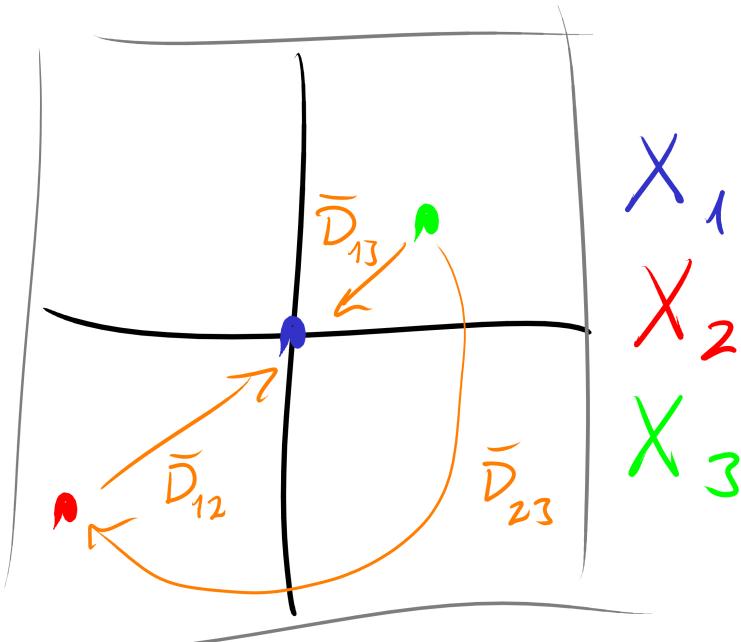
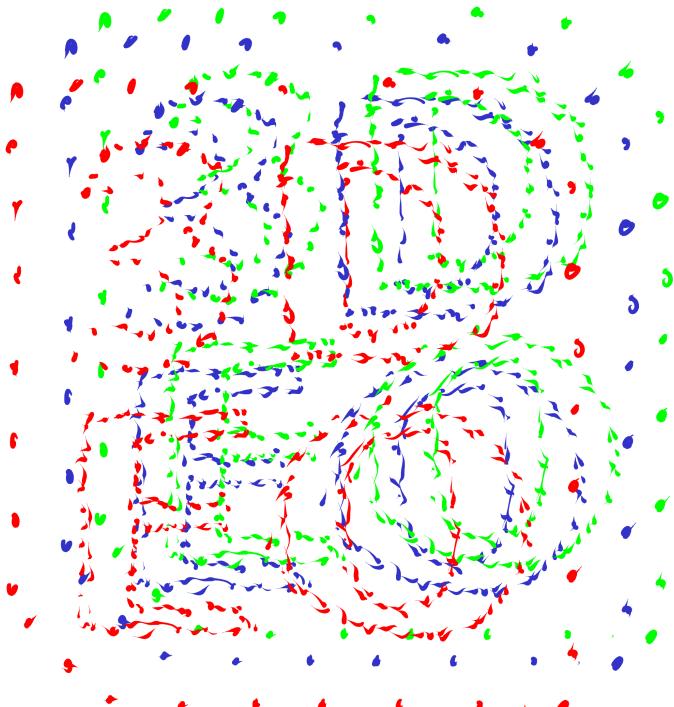
# Global registration from pairwise registration

# 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.

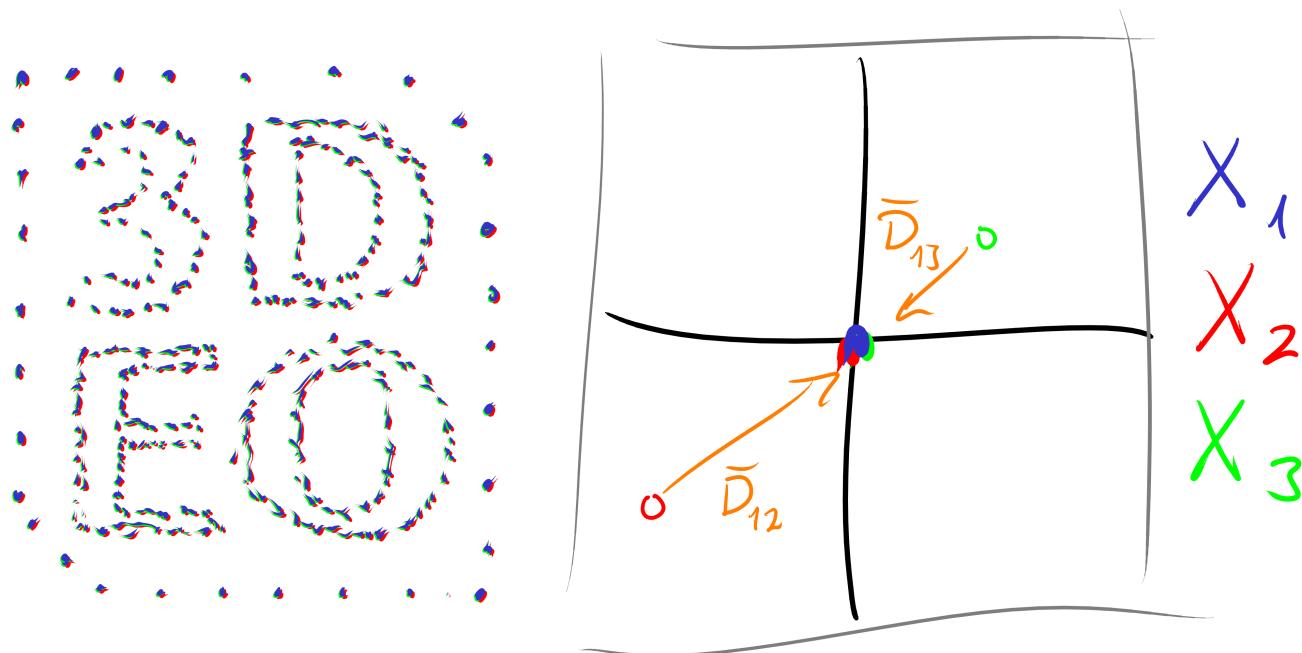
# Global registration from pairwise registration

Easy to align 2 scans, hard to align 3



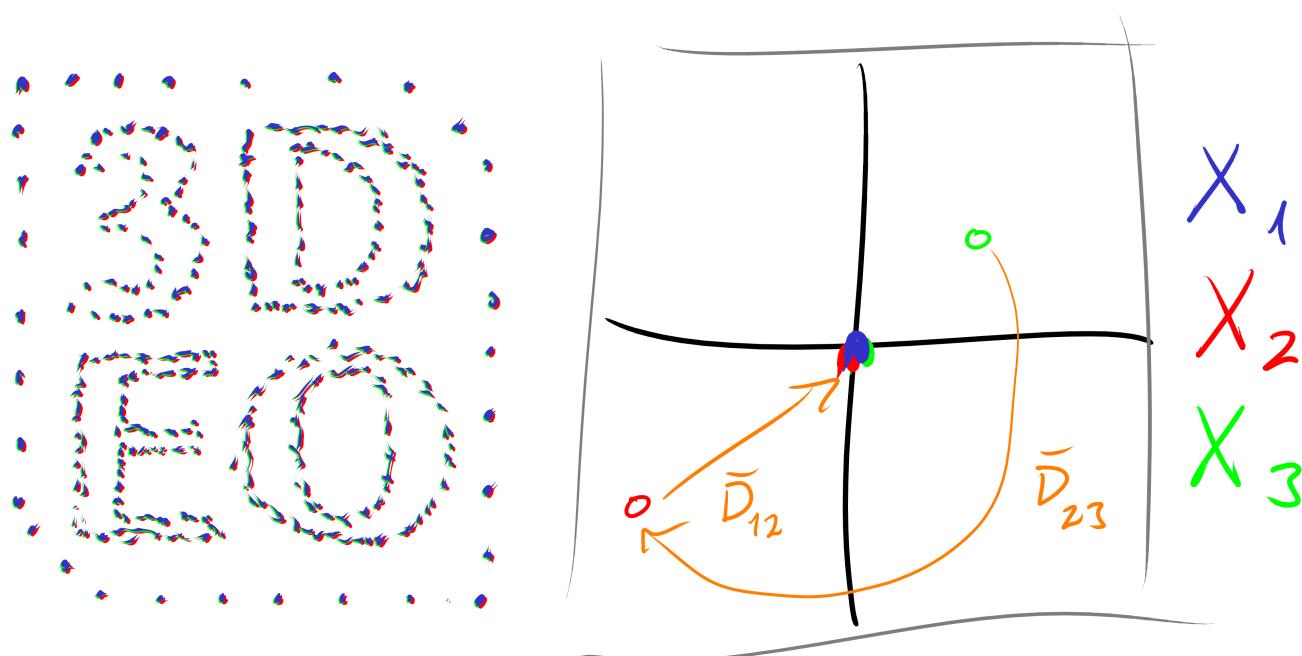
# Global registration from pairwise registration

Easy to align 2 scans, hard to align 3



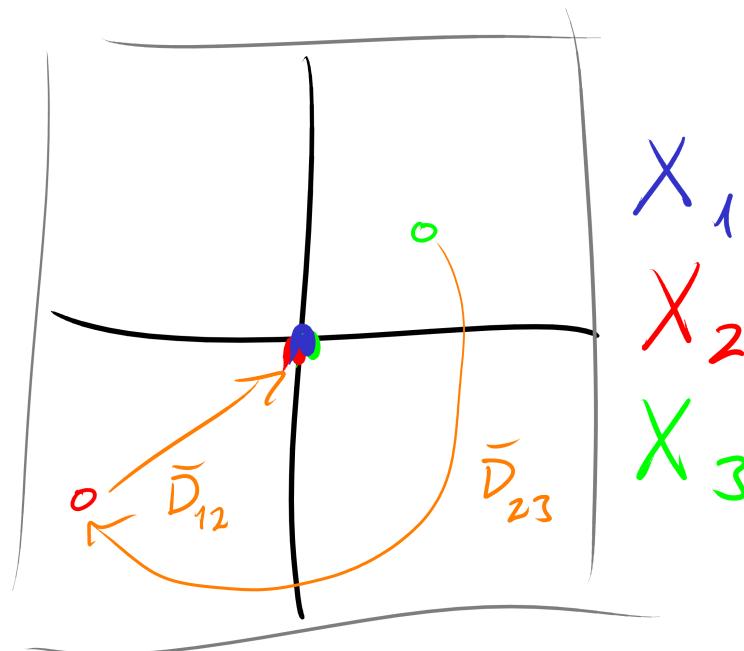
# Global registration from pairwise registration

Easy to align 2 scans, hard to align 3



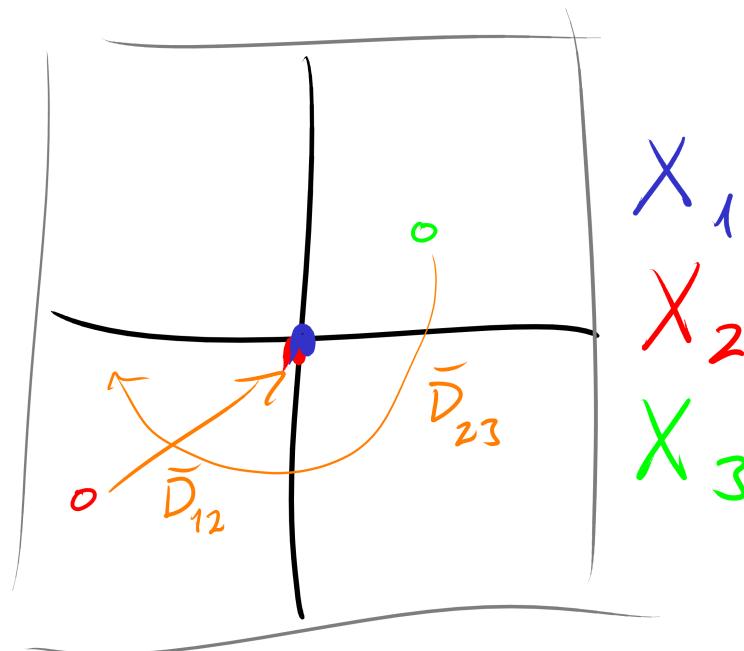
# Global registration from pairwise registration

Easy to align 2 scans, hard to align 3



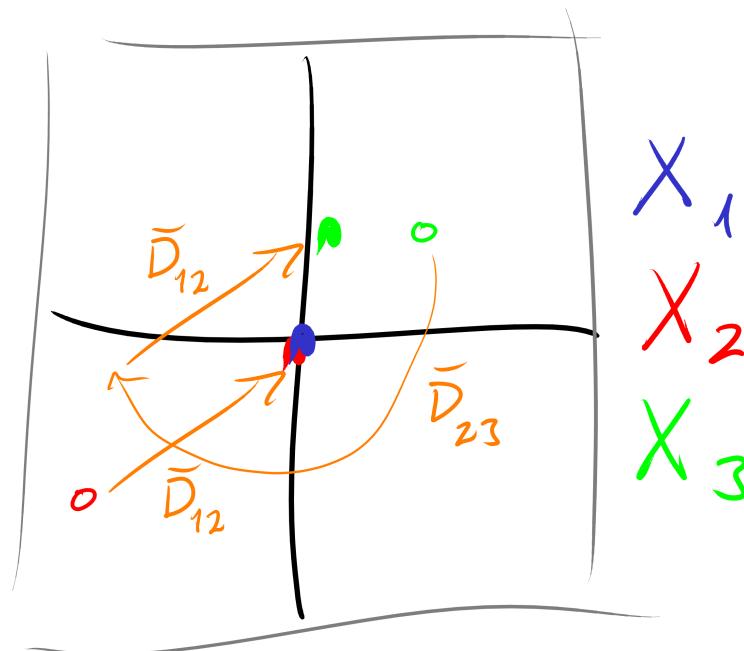
# Global registration from pairwise registration

Easy to align 2 scans, hard to align 3



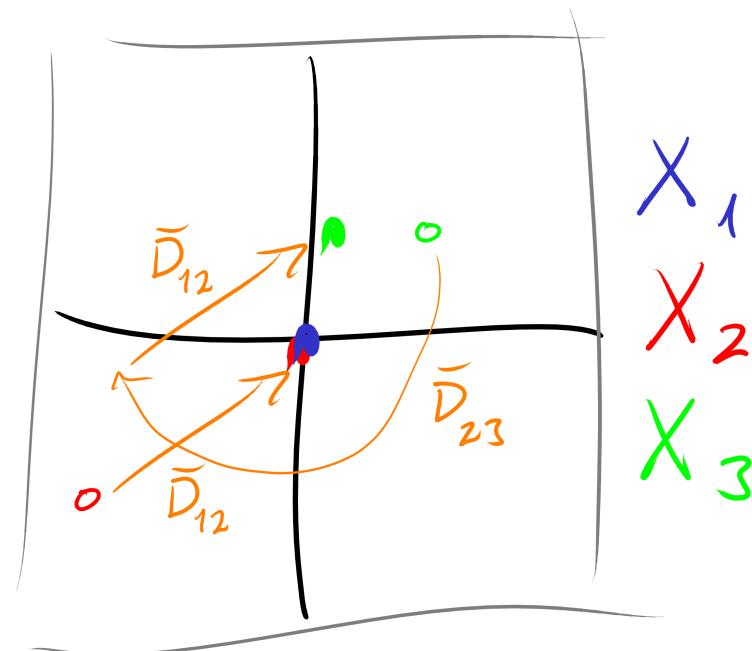
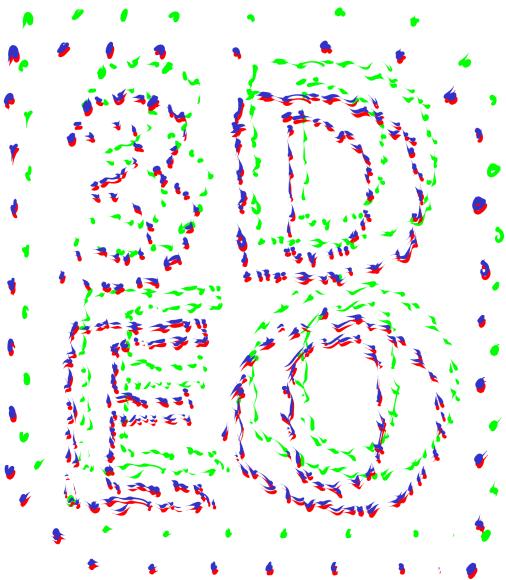
# Global registration from pairwise registration

Easy to align 2 scans, hard to align 3



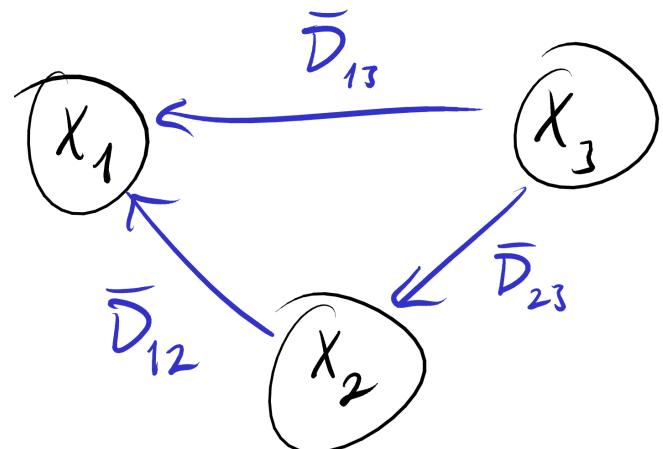
# Global registration from pairwise registration

Easy to align 2 scans, hard to align 3



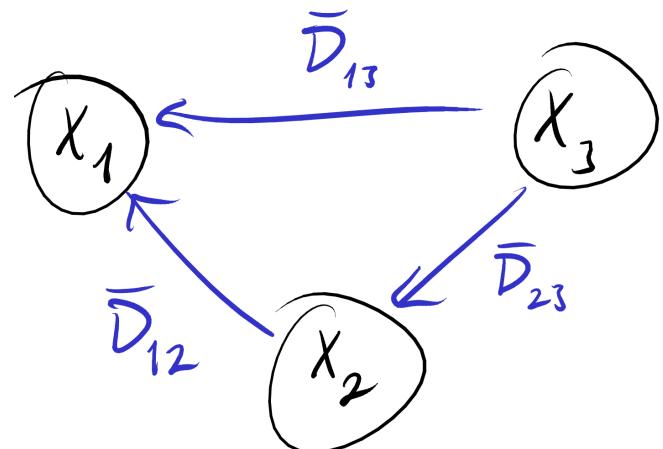
# Global registration from pairwise registration

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

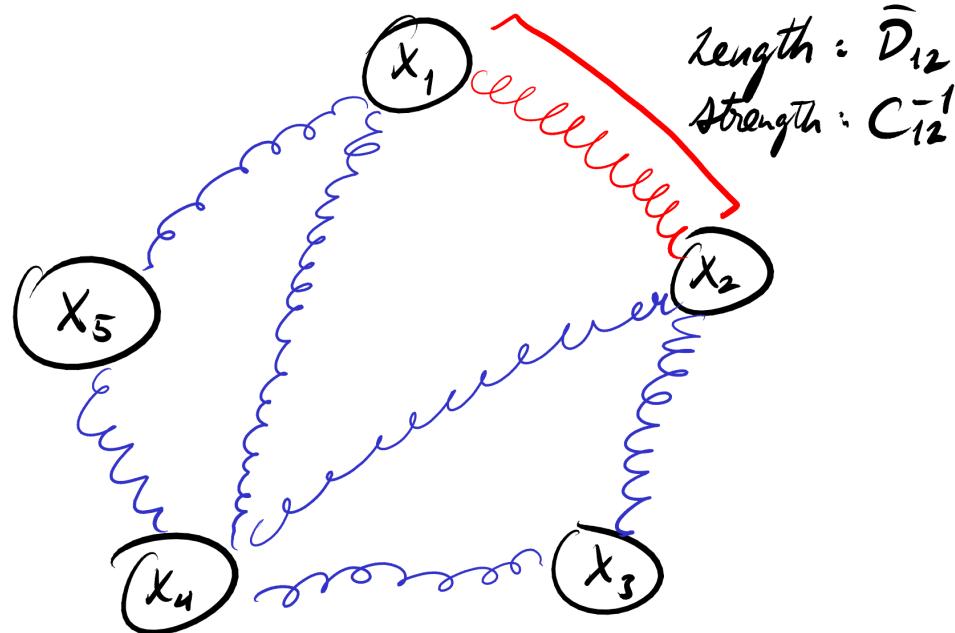
Straightforward to move one scan to align with another, less so to move  $n$  scans to align with each other.



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

# Global registration from pairwise registration

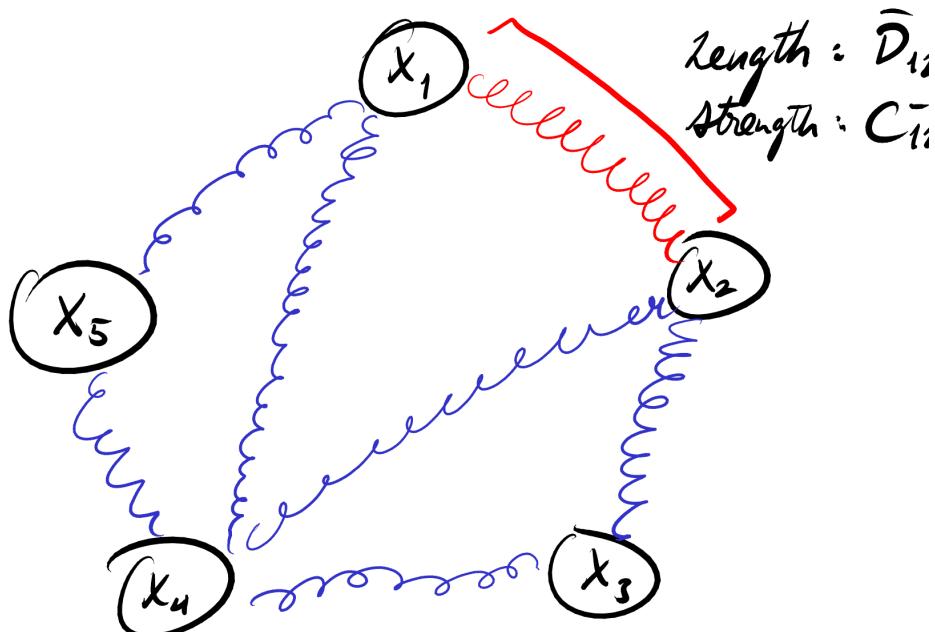
Pose graph optimization.



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

# 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)

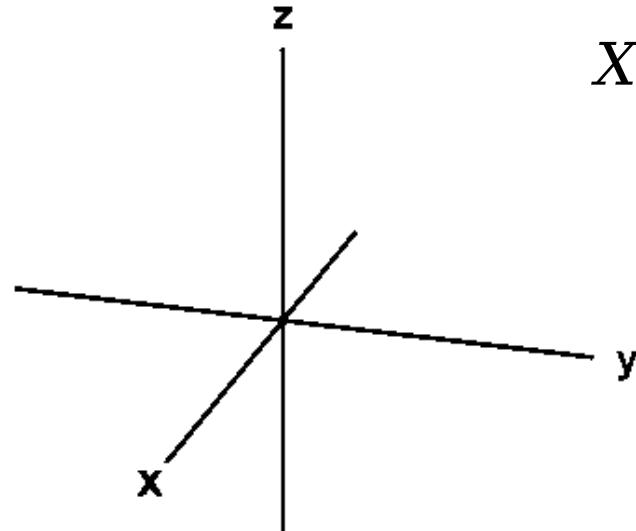
$$\underset{\{X_1, \dots, X_n\}}{\text{minimize}} \quad \sum_{i,j} \left( \bar{D}_{ij} - (X_i - X_j) \right)^T C_{ij}^{-1} \left( \bar{D}_{ij} - (X_i - X_j) \right)$$

# Global registration from pairwise registration

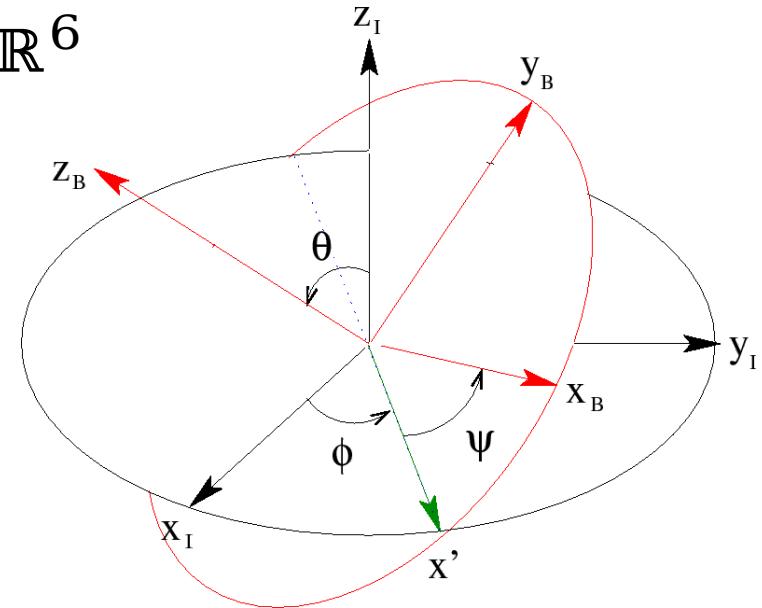
$$X_i = \begin{pmatrix} x_i \\ y_i \\ z_i \\ \theta_i \\ \varphi_i \\ \psi_i \end{pmatrix} \in \mathbb{R}^6$$

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

# Global registration from pairwise registration

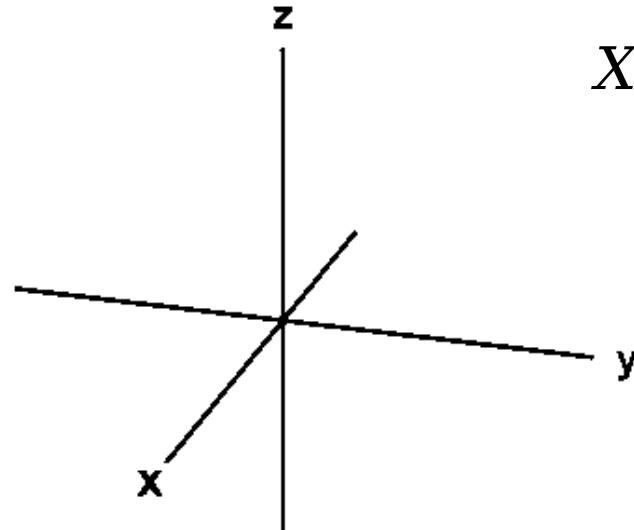


$$X_i = \begin{pmatrix} x_i \\ y_i \\ z_i \\ \theta_i \\ \varphi_i \\ \psi_i \end{pmatrix} \in \mathbb{R}^6$$

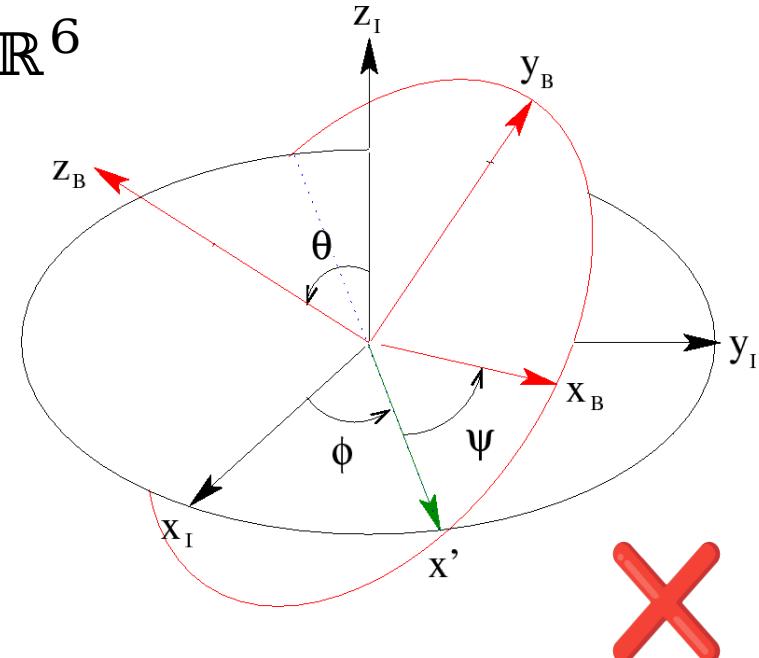


$$\underset{\{X_1, \dots, X_n\}}{\text{minimize}} \quad \sum_{i,j} \left( \bar{D}_{ij} - (X_i - X_j) \right)^T C_{ij}^{-1} \left( \bar{D}_{ij} - (X_i - X_j) \right)$$

# Global registration from pairwise registration



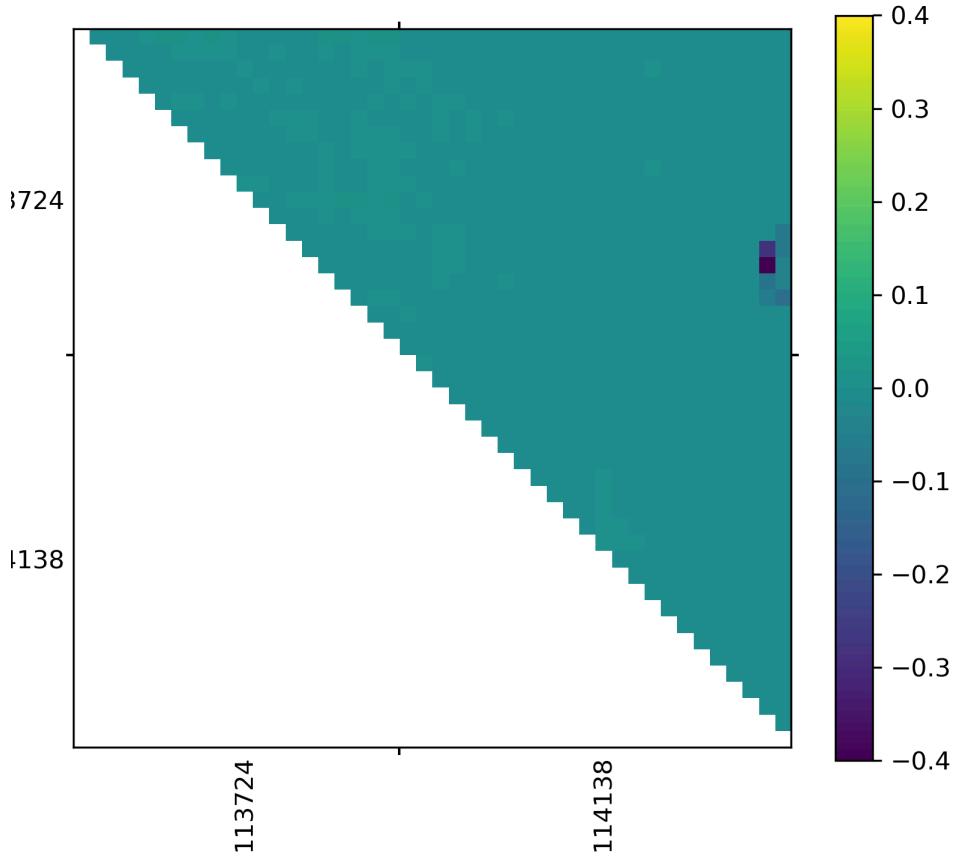
$$X_i = \begin{pmatrix} x_i \\ y_i \\ z_i \\ \theta_i \\ \phi_i \\ \psi_i \end{pmatrix} \in \mathbb{R}^6$$



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

# Global registration from pairwise registration

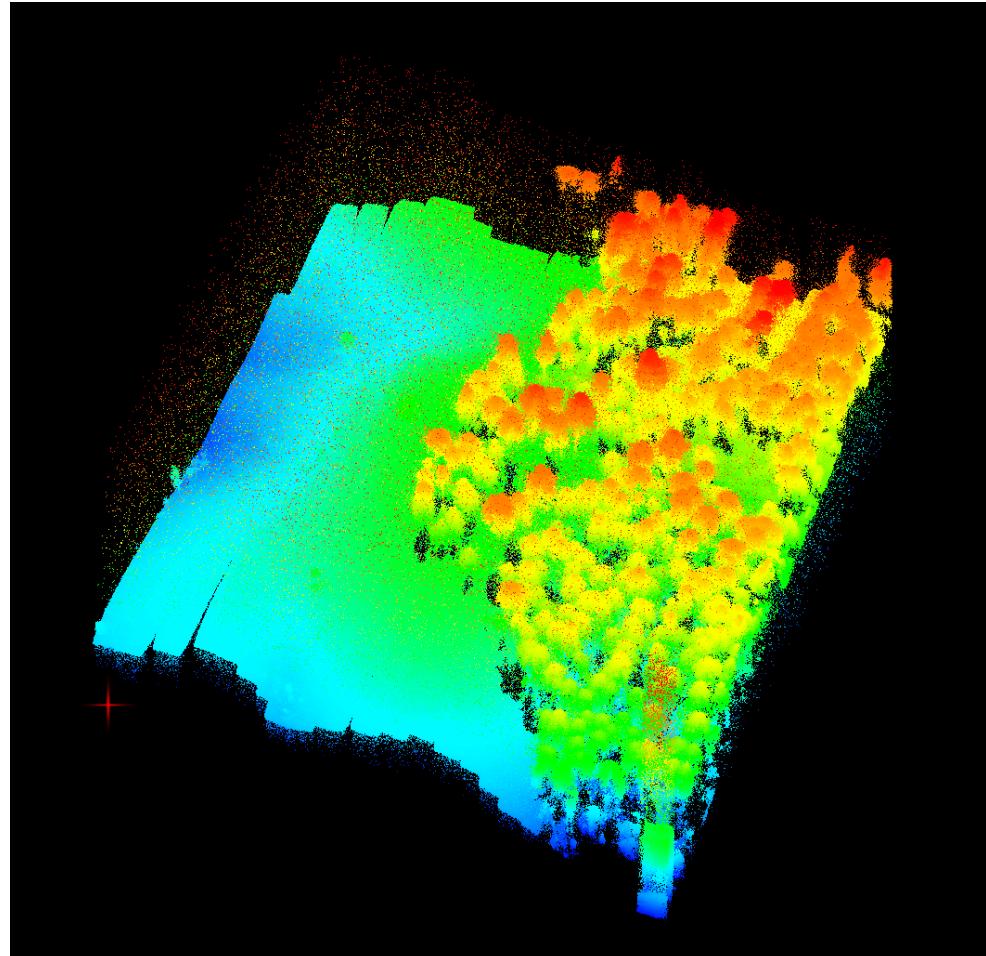
Pruning and weighting.



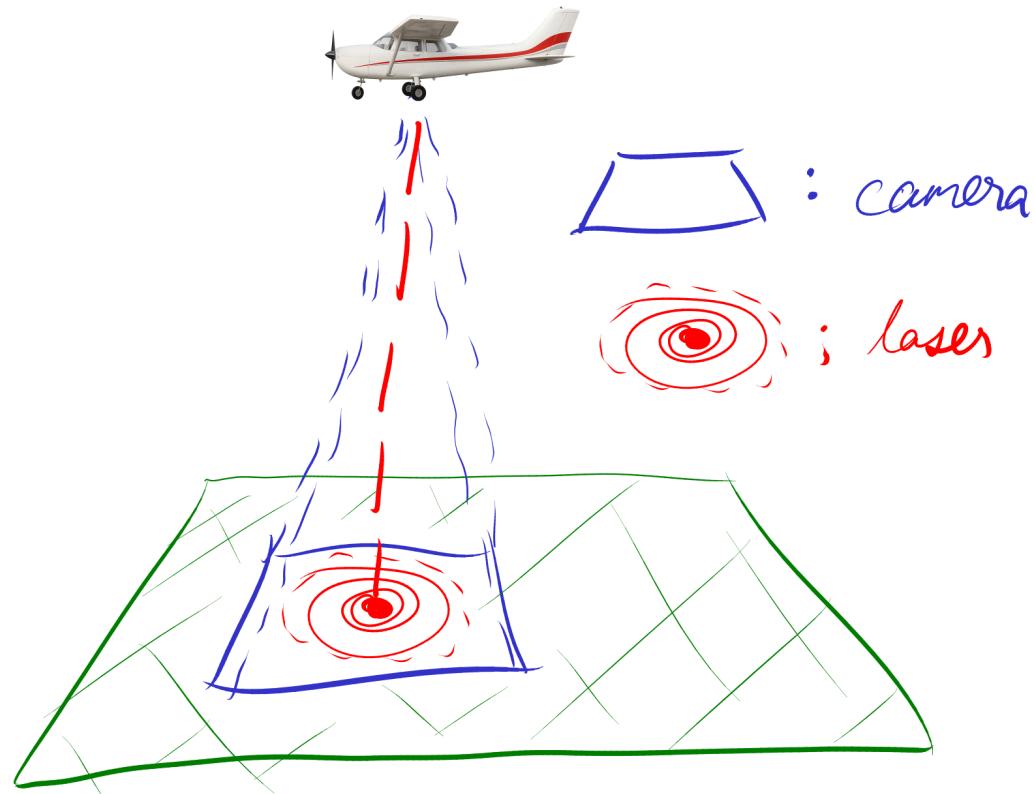
- After optimization, some springs (pairwise registrations) are stretched
- Lots of redundancy in the graph reveals poor pairwise registrations
- Weight those springs less in optimization step or remove them

# Spot modeling

Analyzing 20230627\_095732\_cuchillo1\_scan00091.bpf

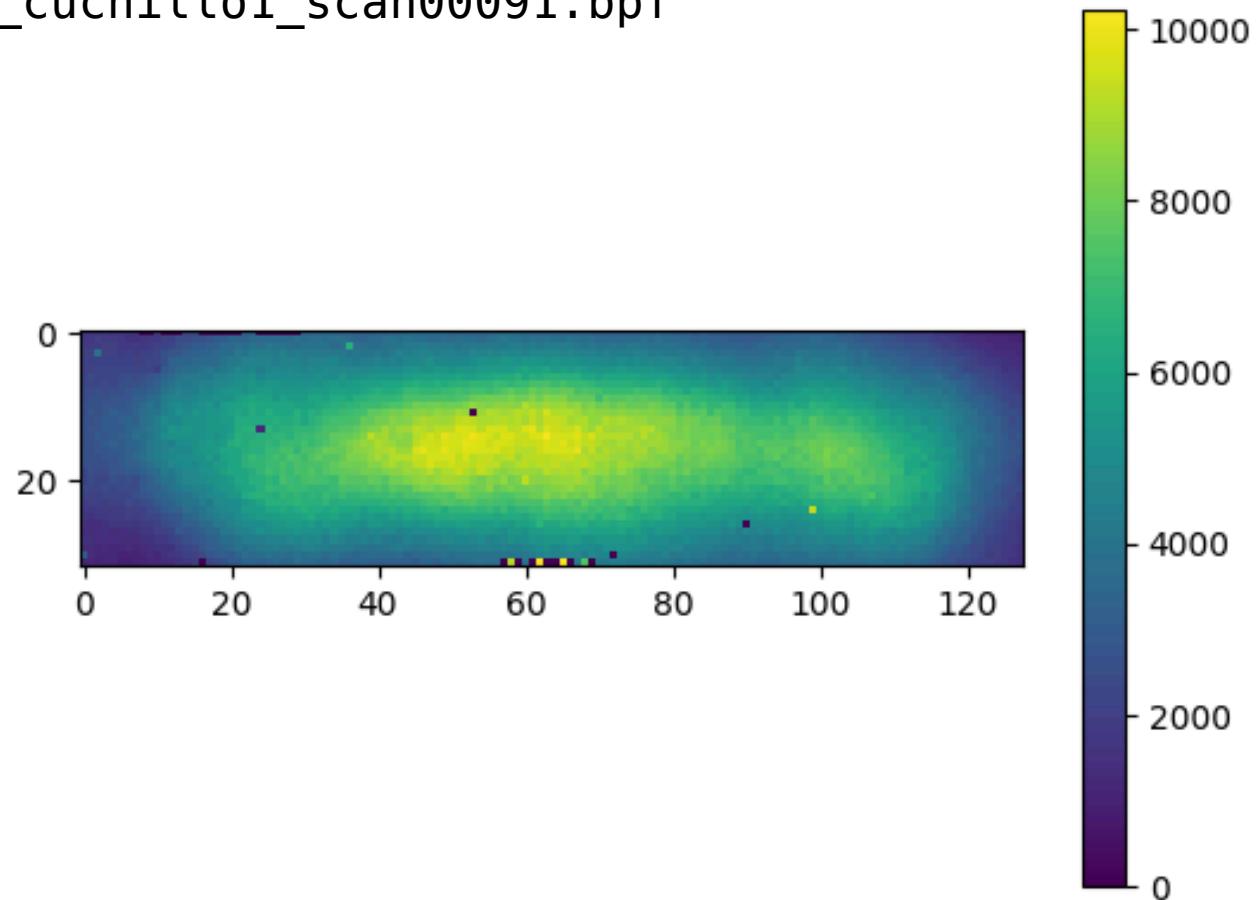


# Spot modeling



# Spot modeling

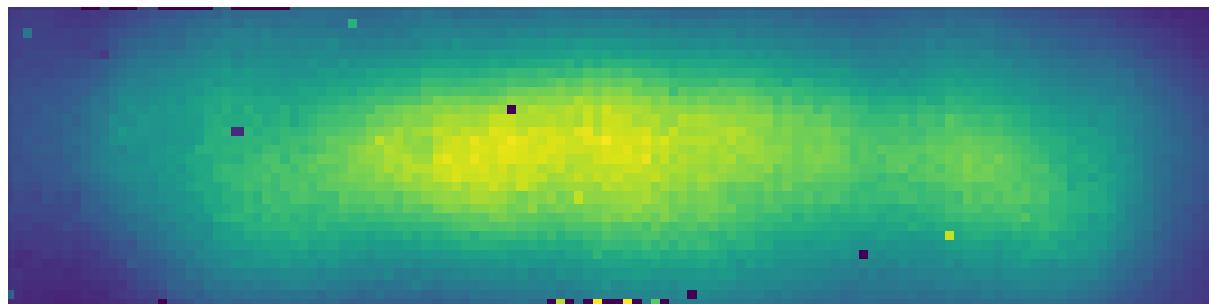
Total photon detections per pixel during  
20230627\_095732\_cuchillo1\_scan00091.bpf



# Spot modeling

Analyzing 20230627\_095732\_cuchillo1\_scan00091.bpf

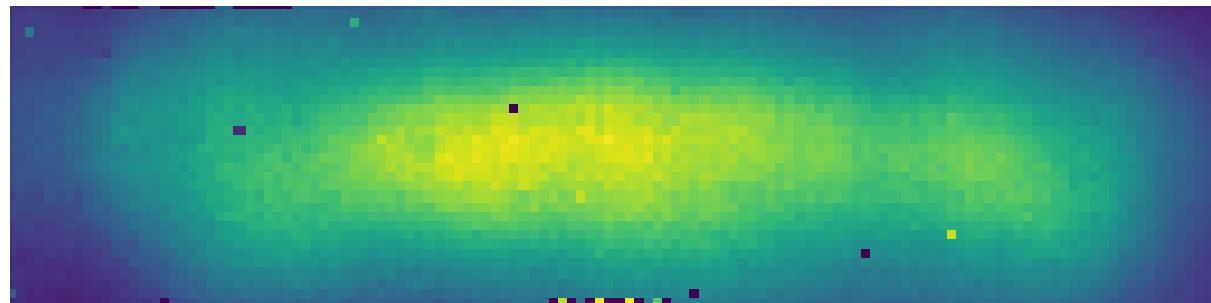
- 30 frame moving average pixel bitmap



# Spot modeling

Analyzing 20230627\_095732\_cuchillo1\_scan00091.bpf

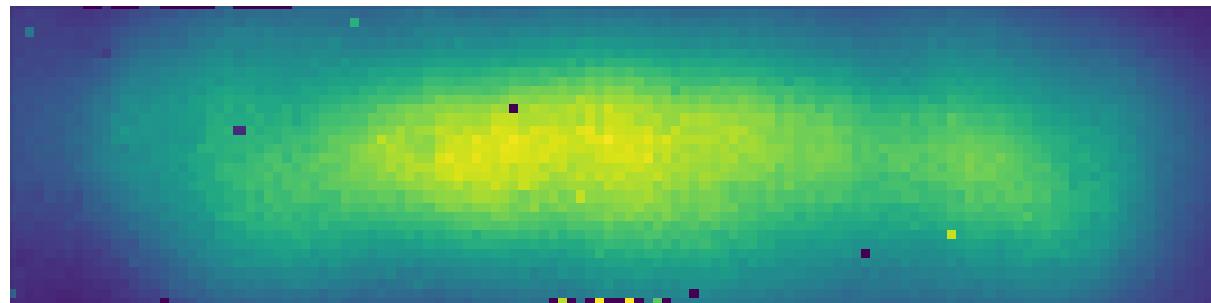
- 30 frame moving average pixel bitmap
- Model with Gaussian Mixture Model (1 component)



# Spot modeling

Analyzing 20230627\_095732\_cuchillo1\_scan00091.bpf

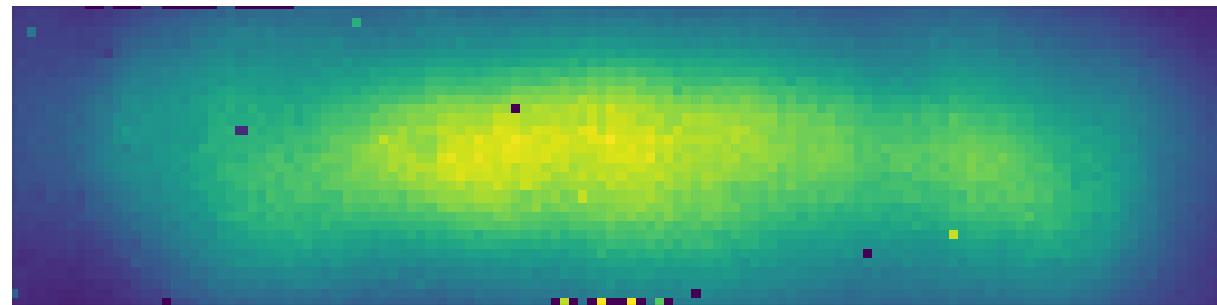
- 30 frame moving average pixel bitmap
- Model with Gaussian Mixture Model (1 component)
- Model with GMMIs



# Spot modeling

Analyzing 20230627\_095732\_cuchillo1\_scan00091.bpf

- 30 frame moving average pixel bitmap
- 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

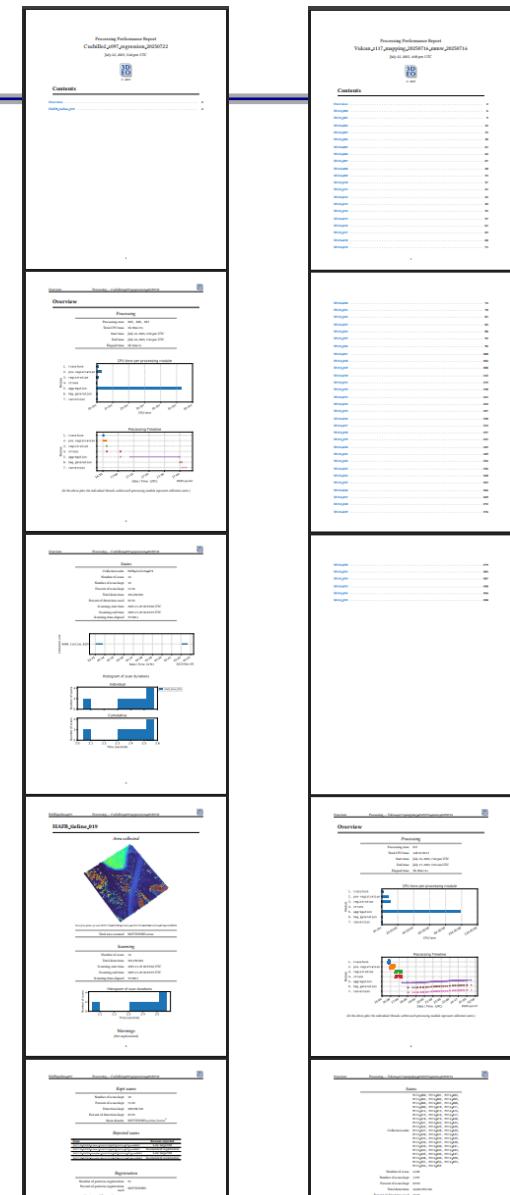
Not to be confused with sensor performance report!

# Processing performance report

Not to be confused with sensor performance report!

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

Located in <acadia-output-directory>/qc/processing\_report



# Processing performance report

3D  
EO

## Processing Performance Report

Vulcan\_z117\_mapping\_20250716\_mmw\_20250716

*July 22, 2025, 4:08 pm UTC*



© 2025

## Contents

---

<a href="#">Overview</a> . . . . .	2
<a href="#">M112_000</a> . . . . .	6
<a href="#">M112_001</a> . . . . .	9
<a href="#">M112_002</a> . . . . .	12
<a href="#">M112_003</a> . . . . .	15

# Processing performance report

## Scans

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,  
Collection units 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

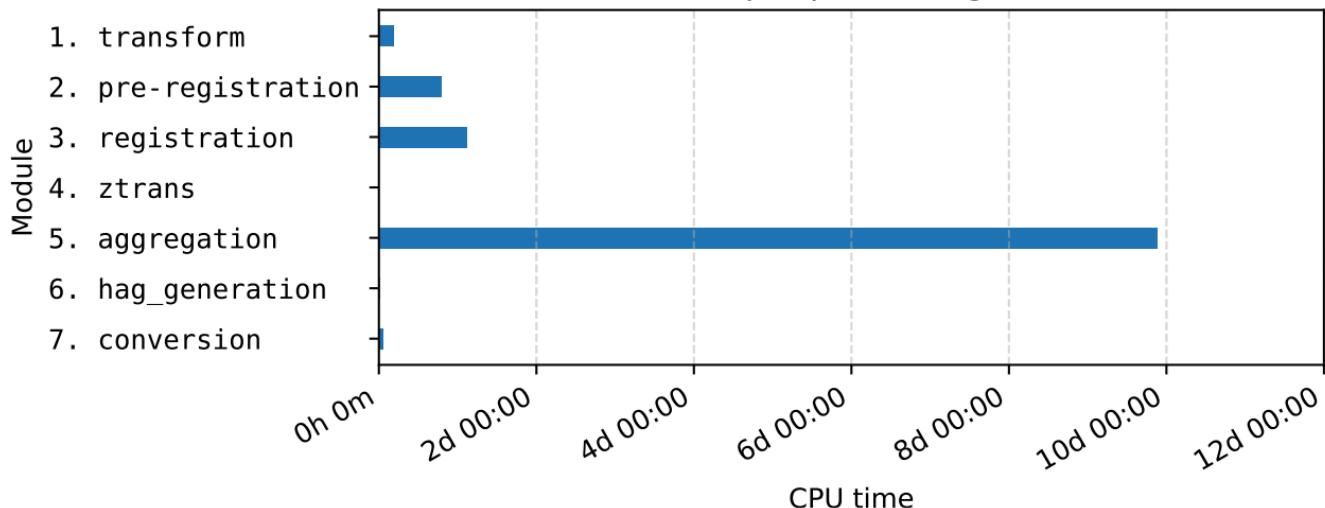
## *Processing*

---

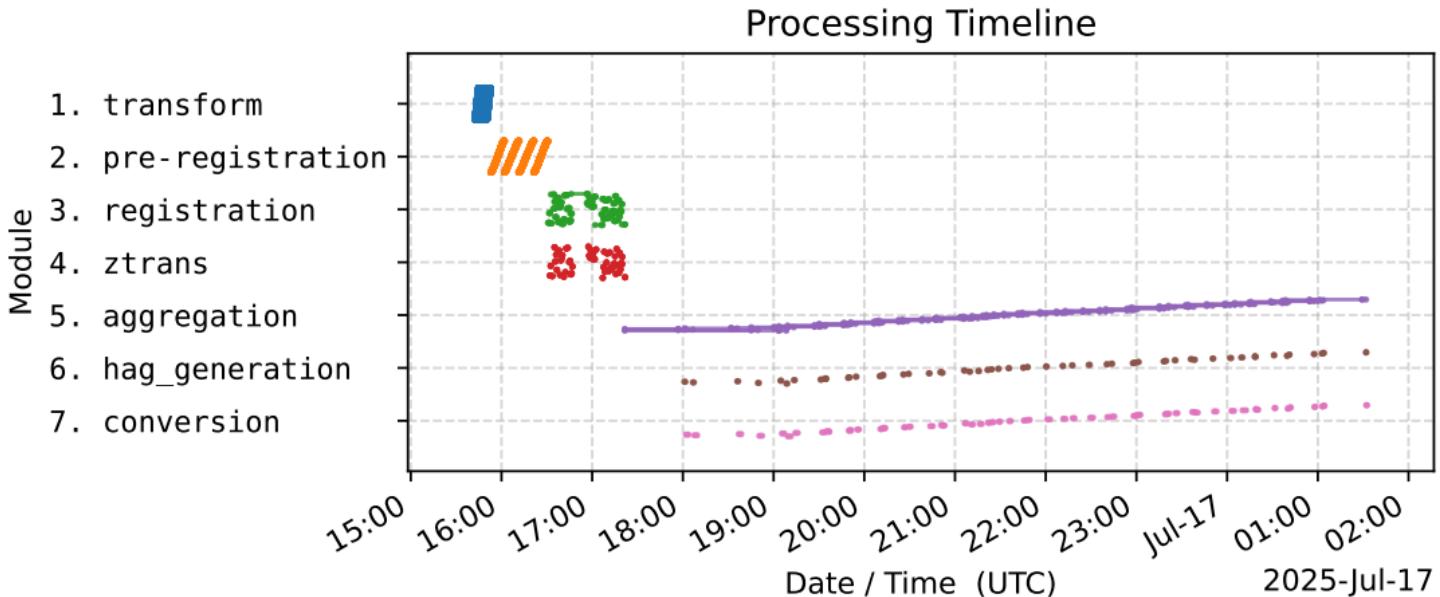
Processing runs	631
Total CPU time	12d 01:30:17
Start time	July 16, 2025, 7:42 pm UTC
End time	July 17, 2025, 5:32 am UTC
Elapsed time	9h 50m 11s

---

CPU time per processing module



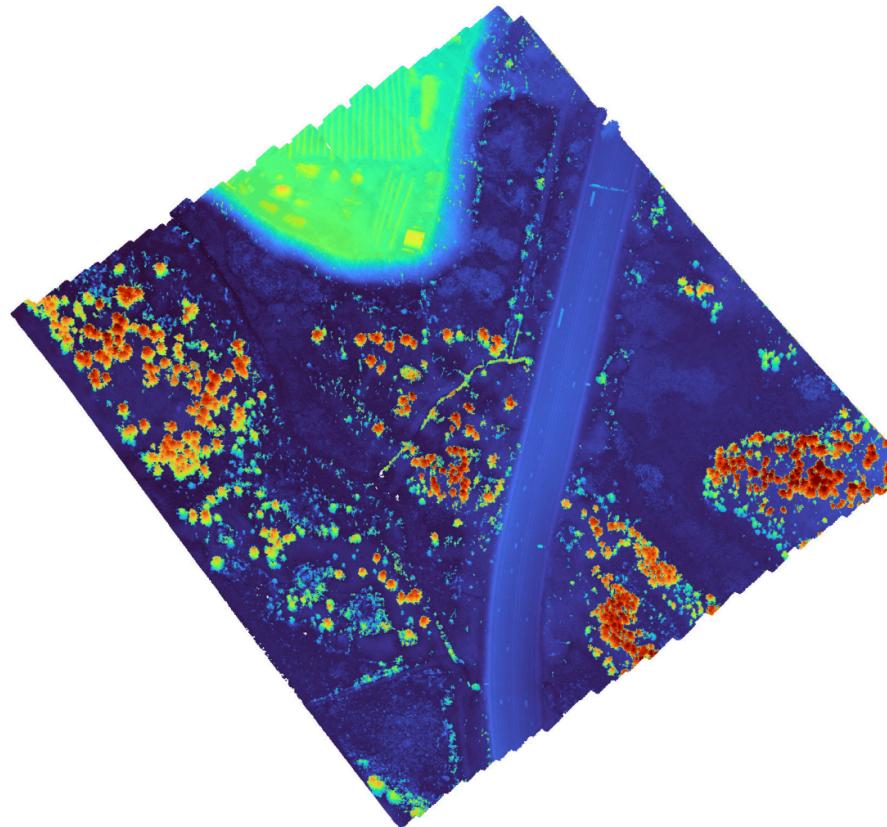
# Processing performance report



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

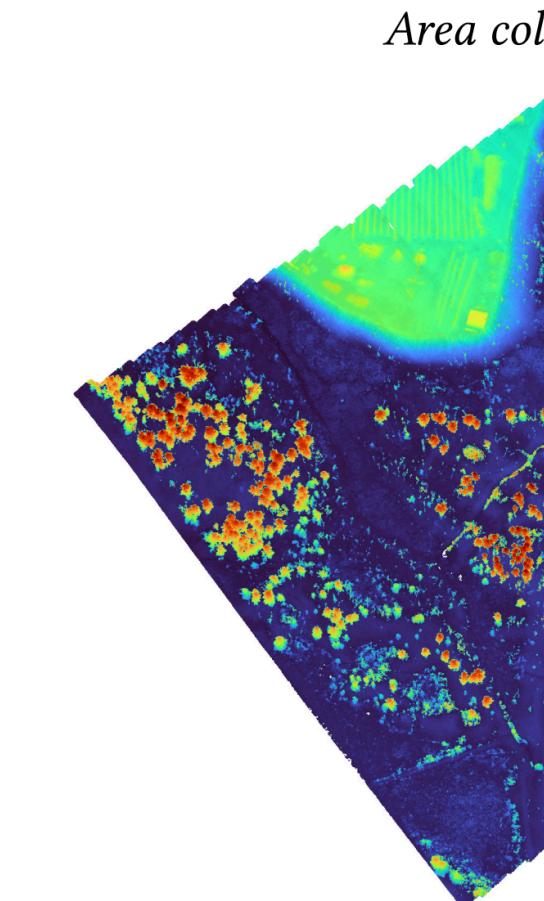
# Processing performance report

*Area collected*



*View from plane of scan 20231129\_023905\_primary\_Cuchillo2\_HAFB\_tieline\_019\_scan00059*

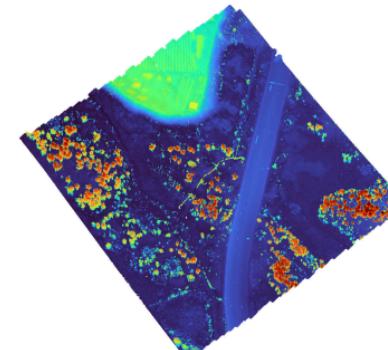
# Processing performance report



HAFB.tieline.019 Processing – Cuchillo2,z097\_regression\_20250722

## HAFB.tieline.019

*Area collected*

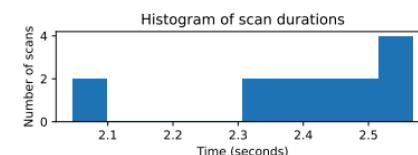


*View from plane of scan 20231129\_023905\_primary.Cuchillo2,HAFB.tieline.019.scan00059*

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



### Warnings

(Not implemented)

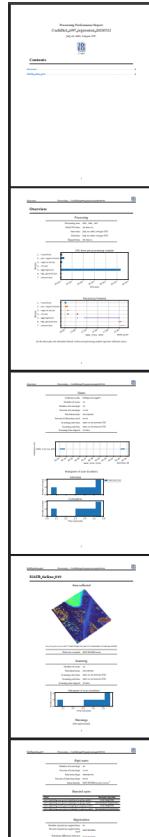
# Processing performance report

## *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



## L<sup>A</sup>T<sub>E</sub>X 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