ICP based Software Alignment at Lumi

November 2018 Progress Report

Roman Klasen | roklasen@uni-mainz.de

Software Tasks

Misalignment Determination

Determination of component positions with:

- measurement data
- cosmic rays
- etc.

This is done with PndLmdSensorAligner.

Software Alignment

Use align matrices for particle reconstruction

This is done with the new FairRoot AlignManager class presented in the computing session.

Software Alignment

We distinguish two related but very different concepts:

<u>During Simulation / Pre-Experiment</u>

Generate mc tracks (or similar) using a “wrong” geometry just like a real detector would produce. The tracks will be off w.r.t. to their “real” position. Use this to study how your analysis software handles a realistic, misaligned geometry. This can be done two different ways, see slides 8 and 9.

<u>For Real Measurement Data / During Experiment</u>

Once built, use the alignment parameters obtained from survey etc on *real measurement data*. This accounts for misaligned detector parts and produces reconstructed tracks that are closer to the real tracks than without alignment. ***This is the main goal of software alignment.***

Shift Detector vs. Shift Data

Shift Detector

- Realistic Detector Acceptance
- Realistic scenario for Track Finder, Fitter etc.
- Reco Macro need to only load Geometry from TGeoManager (which handles Align.)
- But need to generate MC Data again (esp. If you want multiple misaligned geometries)

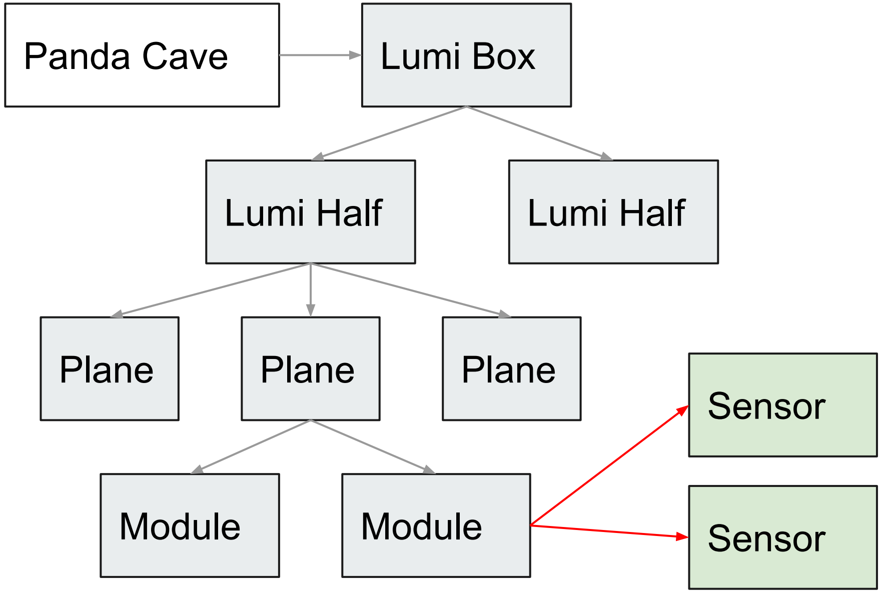
Shift Data

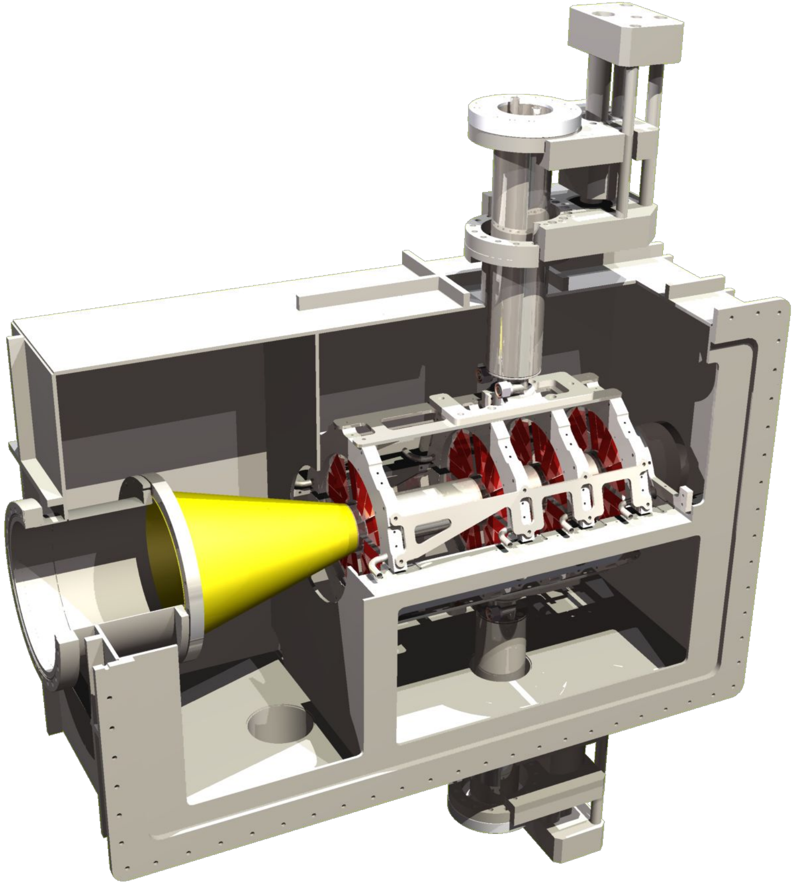
- Can use existing MC data
- Wrong detector acceptance may lead to implausible tracks:
- Don't see some tracks that should be there
- See tracks that can't be there
- Reco Macros must account for Misalignment

I'll be using both methods in this talk.

Lumi Example

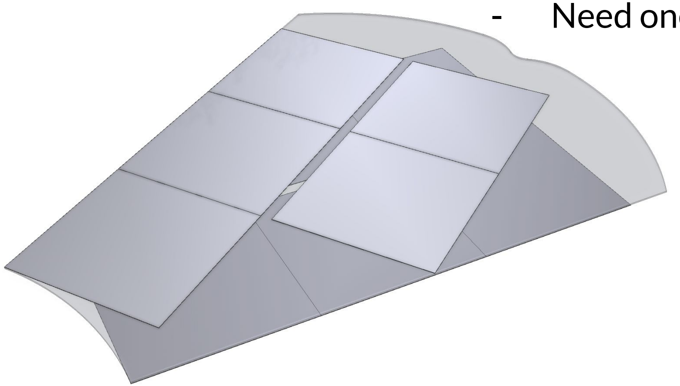
Alignable Components

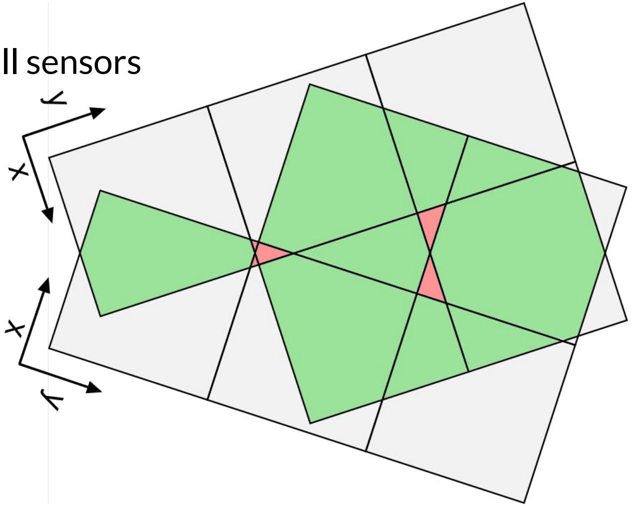




Sensor Alignment with overlapping Areas

- Use overlaps to get matrix s1 -> s2
- Go from sensor to sensor to reach all sensors
- Need one sensor as reference





Point-cloud based Alignment with ICP

We treat the hits on the front and back sensors as point clouds.

After filtering and selection, two clouds with N elements remain, each point in cloud A corresponds to a point in cloud B.

The transformation from cloud A to cloud B is called M, and it's the transformation matrix from sensor A to sensor B.

We use an iterative closest point algorithm that finds the optimum transformation matrix. Differences to the design matrices can stem from:

- Detector resolution
- Wrong pair filtering
- Wrong particle (or their angles an entry)

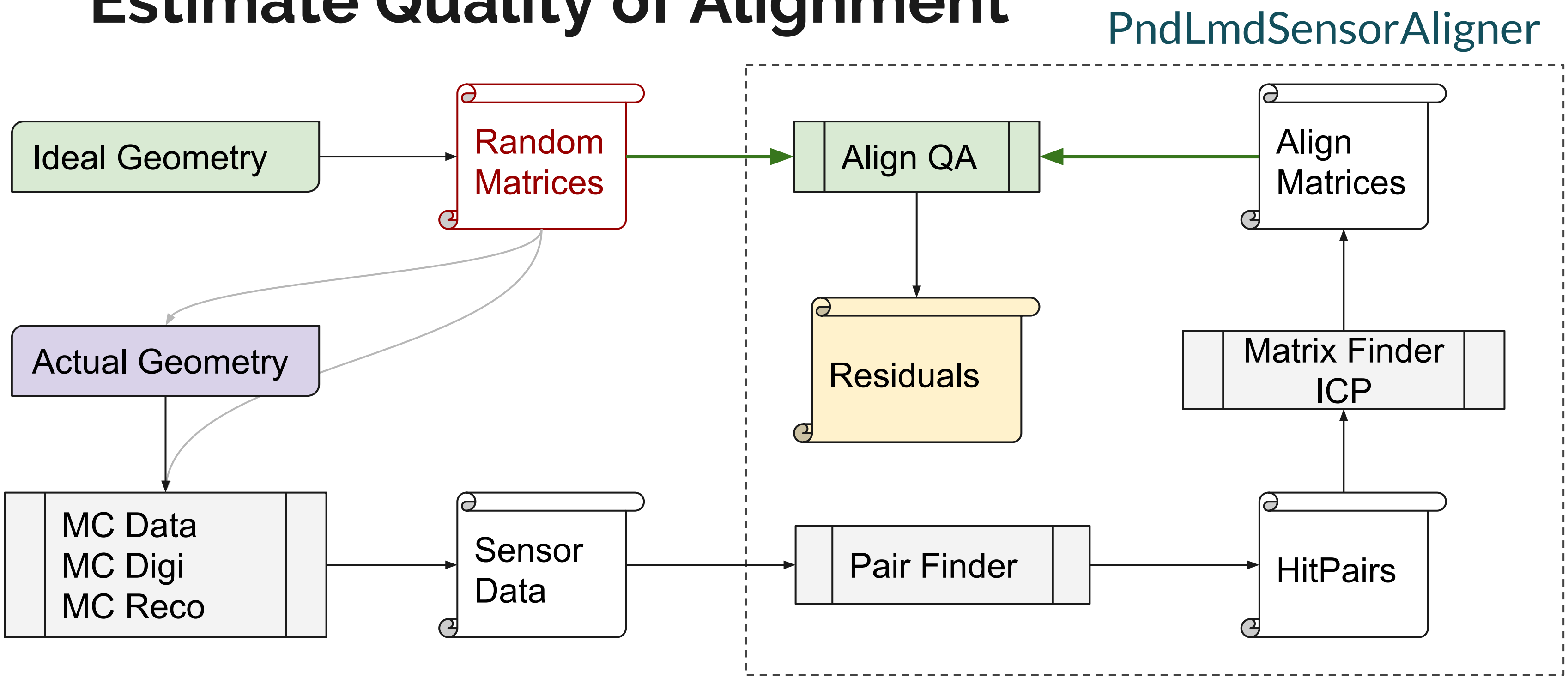
Software Parameters

- We used multiple Geometries
- We Misalign Sensors only
- We allow XY Shift and z-Rotation only
- Enough data for ~ 10^5 pairs/area

We can reach all sensors just be stepping from sensor to sensor by their overlapping areas

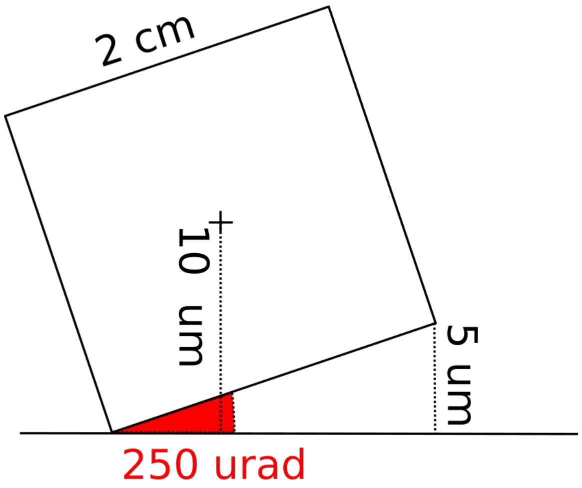
We compare the found alignment matrices with the ones provided to the simulation

Estimate Quality of Alignment



Multiple Sets of random Matrices

Use a fixed relation shift -> rotation



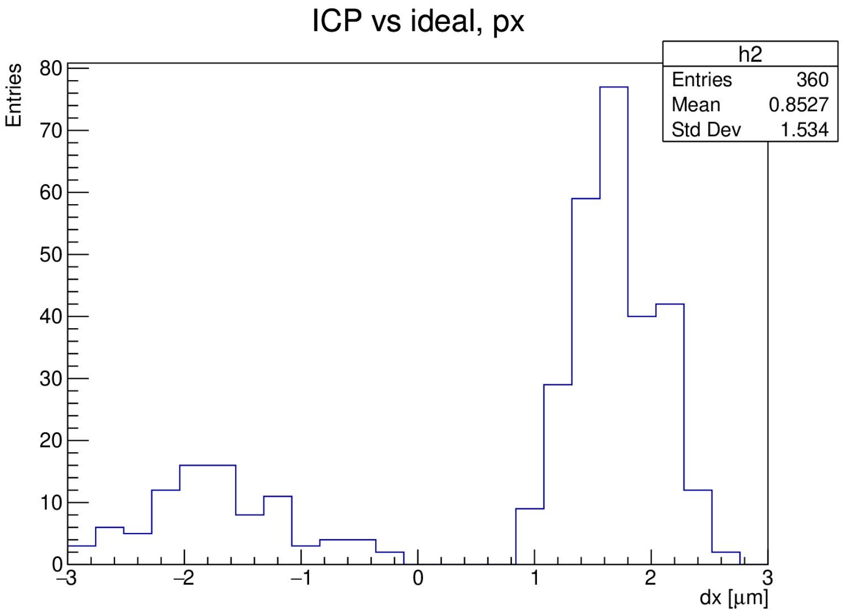
We'll use these shorthands

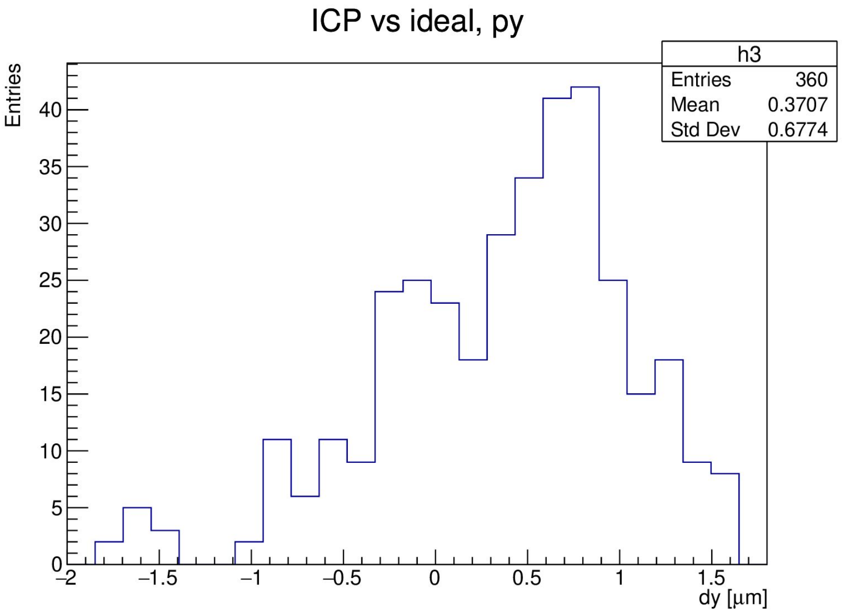
- **0µ** : perfect geometry
- 10µ : 10µm shift, 250µrad rot
- 50µ : 50µm shift, 1.25mrad rot
- **100µ** : 100µm shift, 2.5 mrad rot
- **200µ** : 200µm shift, 5.0 mrad rot
- 250µ : 250µm shift, 7.5 mrad rot

Comparison: Individual Sensor Matrices

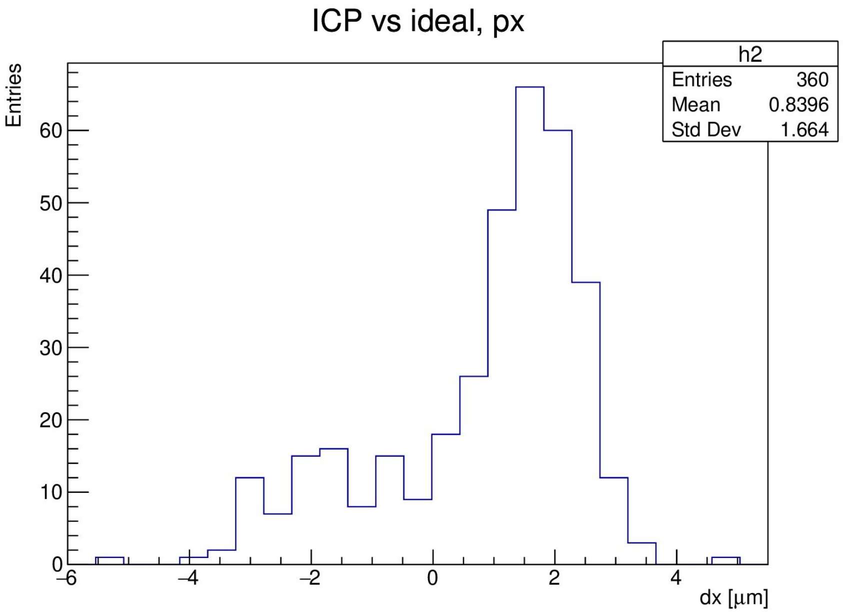
For every overlapping area we determine the overlap matrix and compare it to the design misalignment matrix

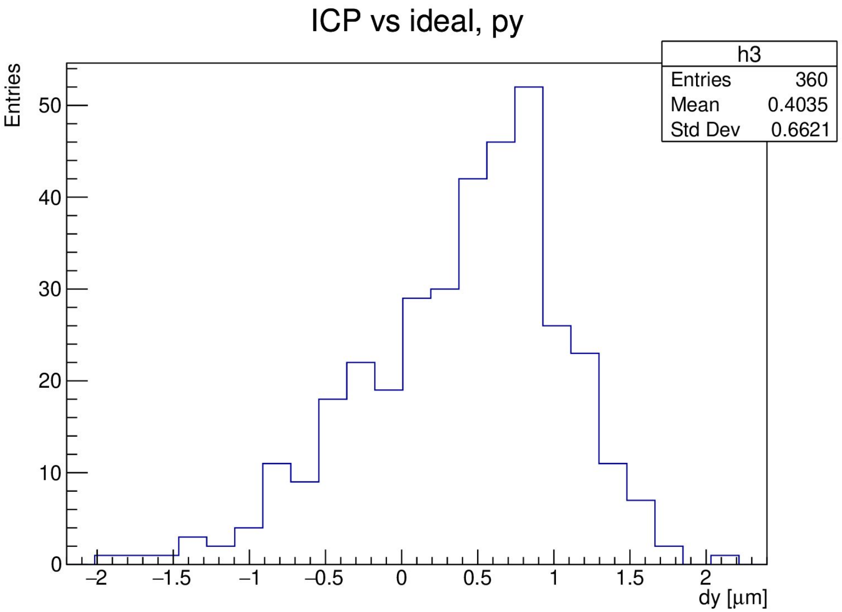
0µ - perfect geometry



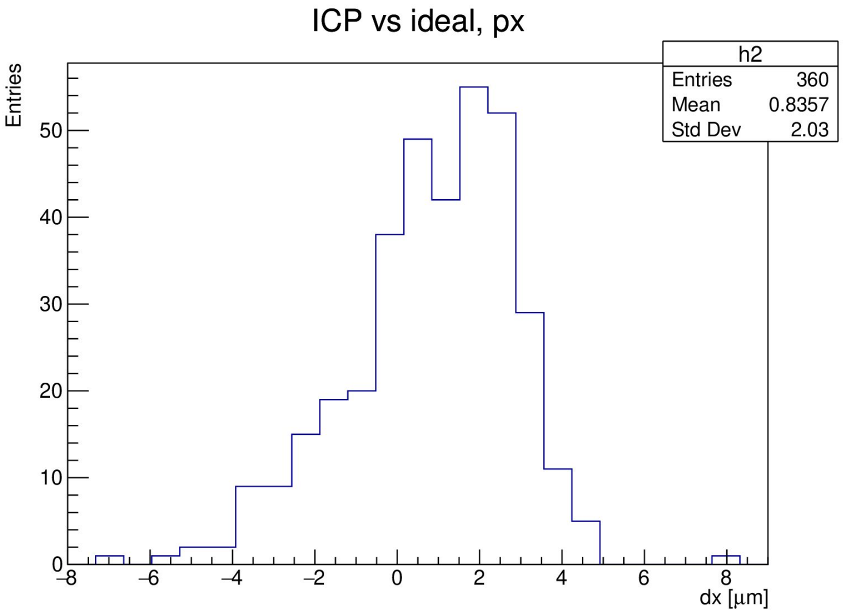


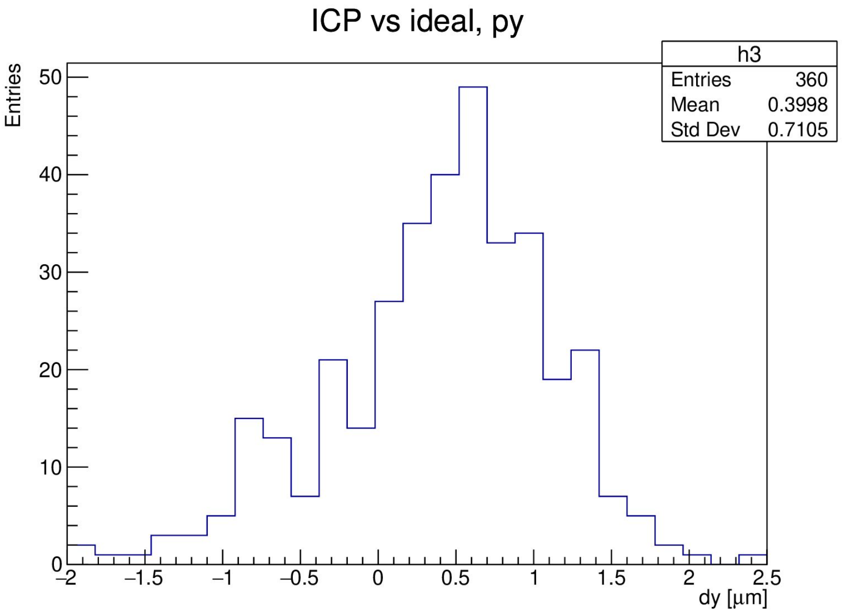
100µ - misaligned geometry





200µ - misaligned geometry



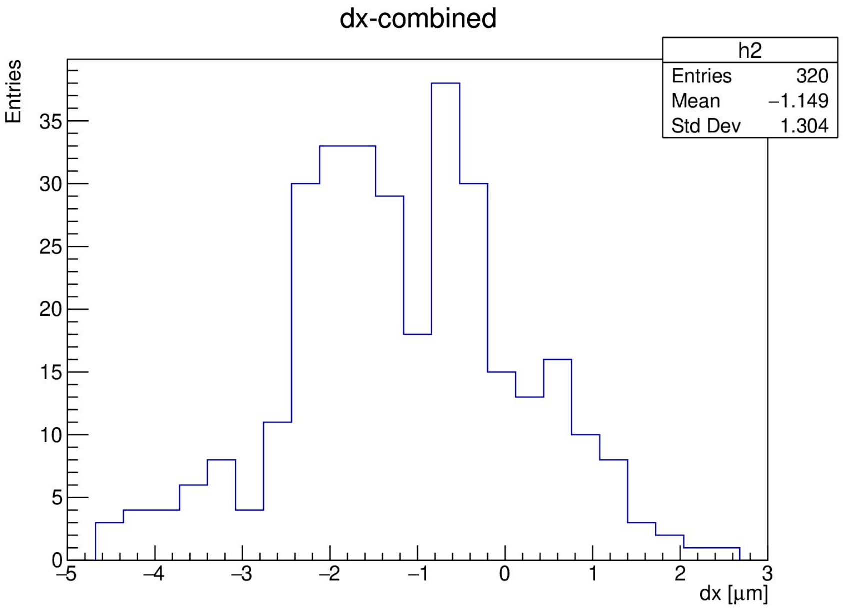


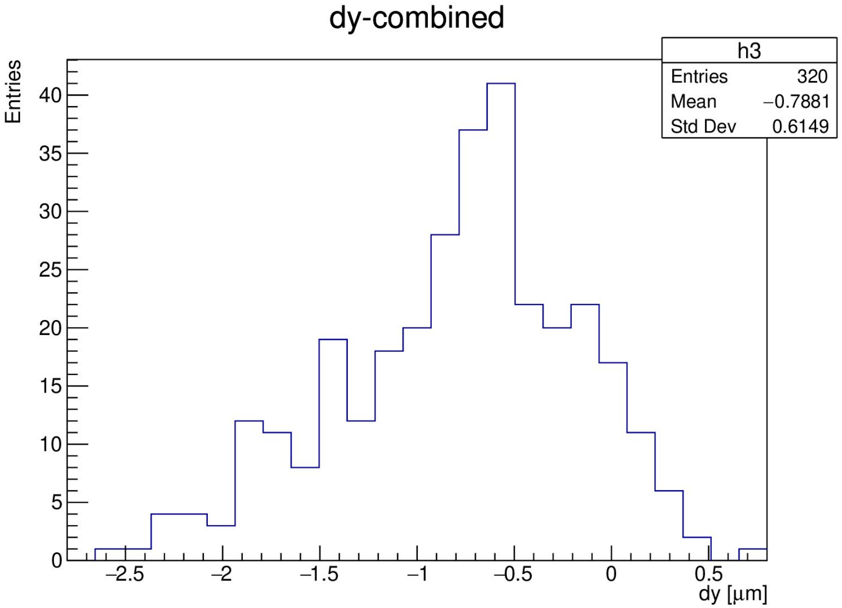
Comparison: Combined Matrices

For all modules, we calculate the resultant matrices from our references sensor to all other sensors on that module.

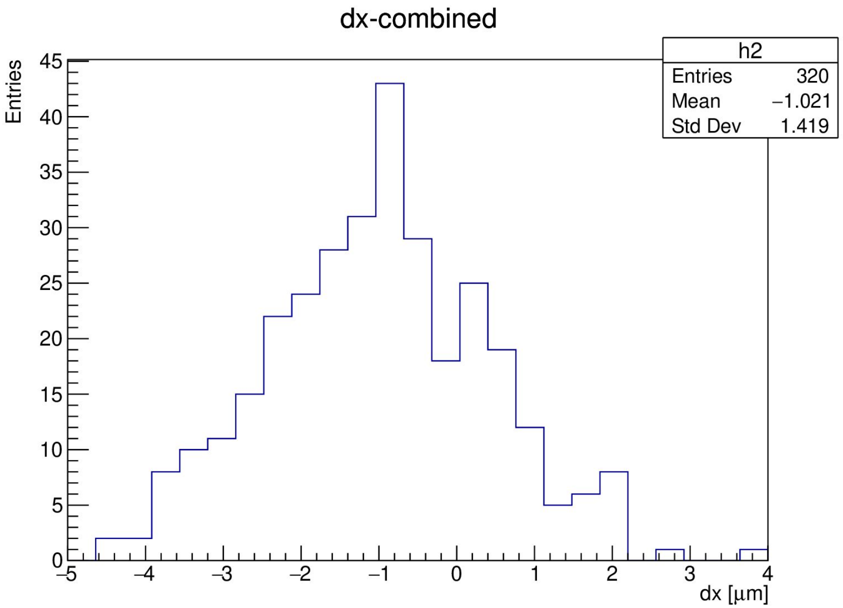
That means there are fewer combined matrices that individual matrices, and each combined matrix consists of several individual matrices.

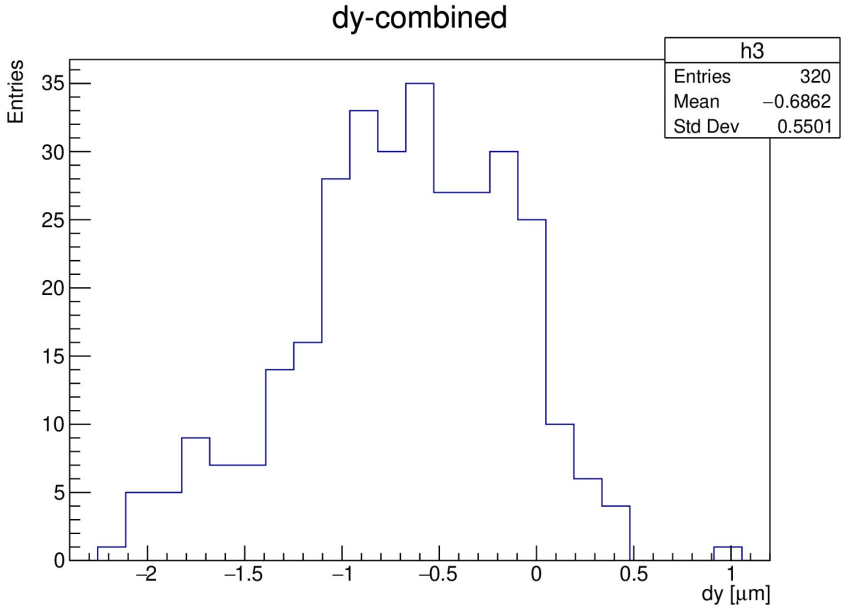
0µ - perfect geometry



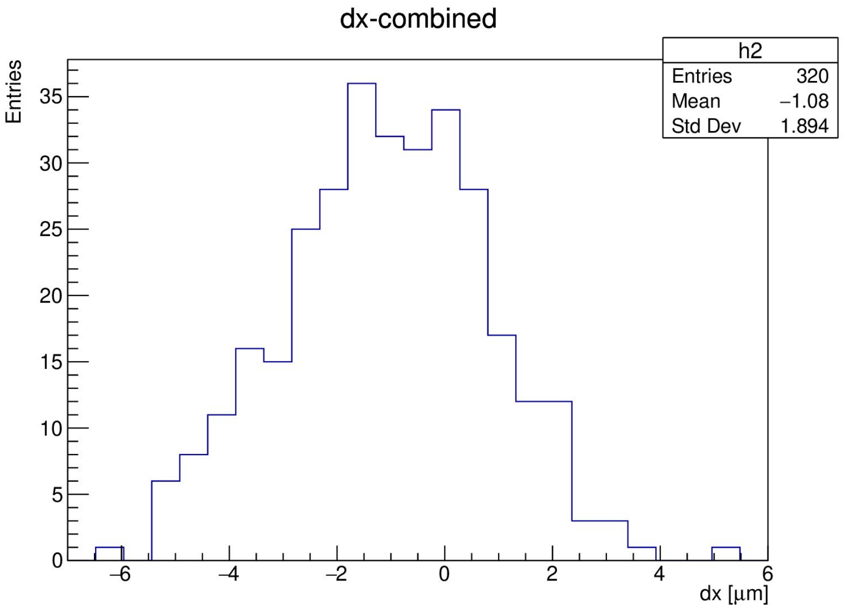


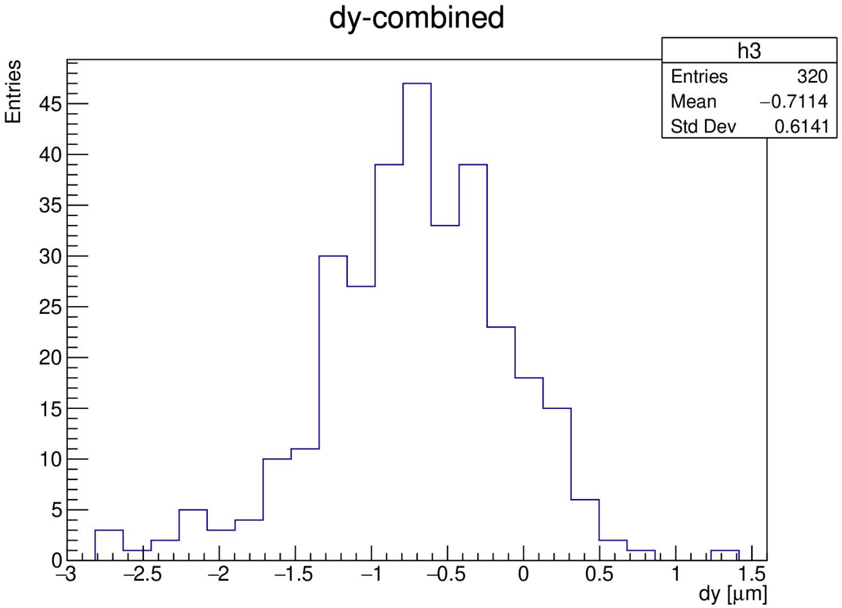
100μ - misaligned geometry





200μ - misaligned geometry



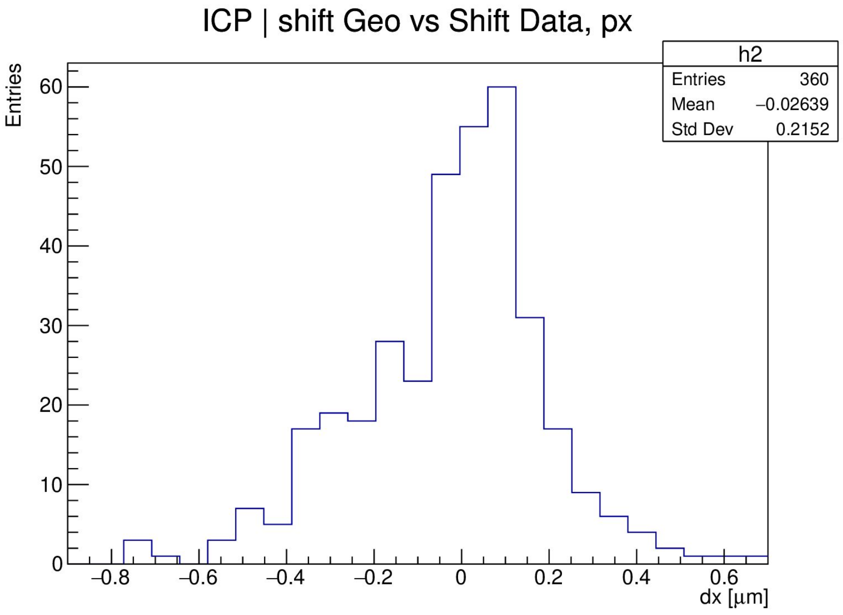


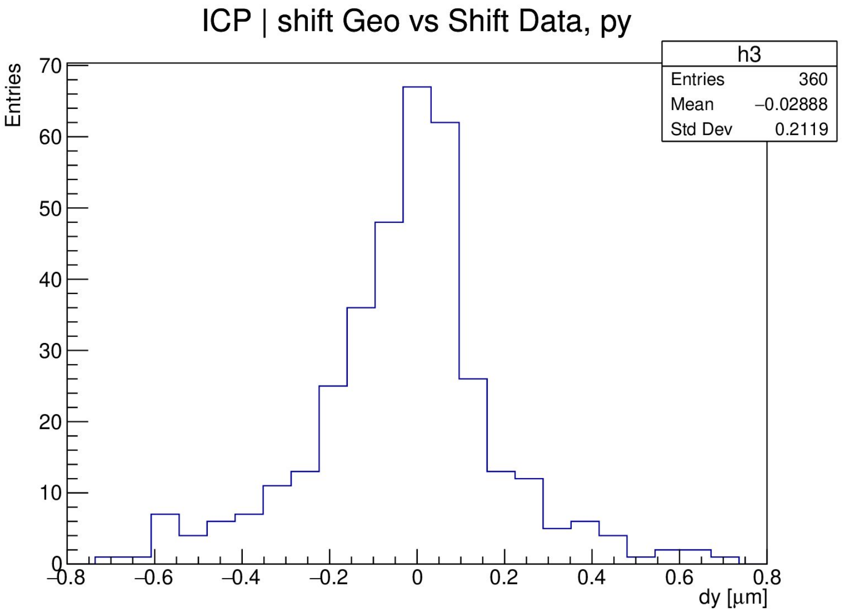
Comparison: Shift Data vs. Shift Geometry

For the single case of 100u misalignment, I compare the resultant individual overlap matrices obtained with shifted geometry with the matrices from shift data.

Due to time constraints, I will not show other misalignments or combined matrices.

100µ - Comparison Shift Geo vs Shift Data





Conclusion

Software alignment using the overlapping areas of two sensors using an ICP algorithm works.

The results are consistent even when the misalignment between two sensors is large.

Both methods (shift geometry and shift data) yield similar results.

Thank you for your attention!