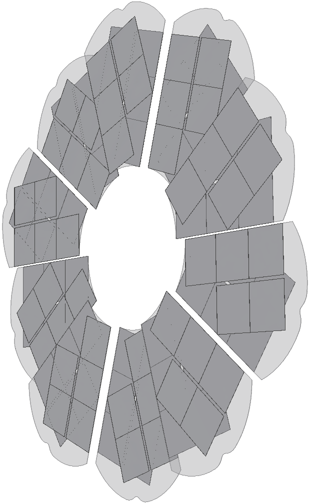
Alignment within one module of $\overline{\text{P}}$ANDA Luminosity Detector

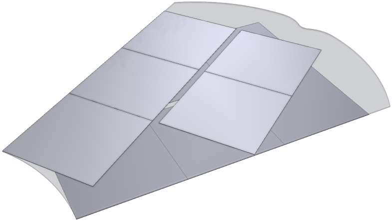
R. Klasen, M. Fritsch, P. Jasinski

Helmholtz-Institut Mainz
Johannes Gutenberg-Universität Mainz

PANDA Collaboration Meeting
December 9, 2013

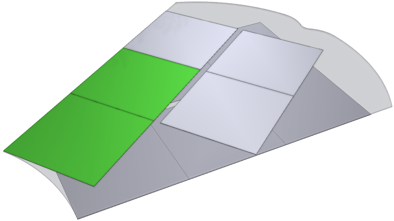
Luminosity pixel detectors





Task

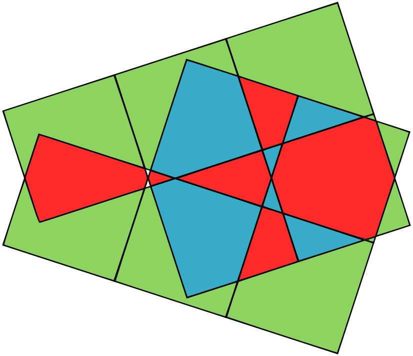
- pixel sensors glued on thin diamond plane from both sides
- quality fluctuations $\rightarrow$ only few sensors grouped to arrays of $(2 \times 4)cm^2$ (green), rest will be $(2 \times 2)cm^2$ sensors (grey)
- full disc coverage $\rightarrow$ 36° angle between sensors
- exact position and rotation of sensors must be known for track reconstruction



Pixel coincidence approach

Using overlapping area

- sensors will overlap partially
- use overlapping area to determine the alignment of each sensor pair



Pixel coincidence approach

Pixel coincidence approach

- charged particle will (ideally) activate single pixel in every sensor
- pixels on front and back must be close (depends on angle of entry and energy of particle)
- sensors have discrete pixels $\rightarrow$ approach will have certain maximum accuracy

transformation matrices

- translation + rotation + scaling can be expressed as single homogenous 4×4 matrix
- $\Rightarrow$ alignment of two sensors can be expressed as 4×4 matrix!
- hopefully, no scaling here (sensors should have same size)
- algorithm needed to find matrix from set of pixel hit pairs

getting info from 4 × 4 transformation matrix

Translation matrix:

$$T(x, y, z) = \begin{pmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotation matrix:

$$R(\alpha, \beta, \gamma) = \begin{pmatrix} \cos[\beta]\cos[\gamma] & \cos[\gamma]\sin[\alpha]\sin[\beta] - \cos[\alpha]\sin[\gamma] & \cos[\alpha]\cos[\gamma]\sin[\beta] + \sin[\alpha]\sin[\gamma] & 0 \\ \cos[\beta]\sin[\gamma] & \cos[\alpha]\cos[\gamma] + \sin[\alpha]\sin[\beta]\sin[\gamma] & -\cos[\gamma]\sin[\alpha] + \cos[\alpha]\sin[\beta]\sin[\gamma] & 0 \\ -\sin[\beta] & \cos[\beta]\sin[\alpha] & \cos[\alpha]\cos[\beta] & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

α: rotation about x axis
β: rotation about y-axis
γ: rotation about z-axis

Small steps $\Rightarrow$ small angle approximation. We get:

Rotation matrix

$$R(\alpha, \beta, \gamma) = \begin{pmatrix} 1 & -\gamma & \beta & 0 \\ \gamma & 1 & -\alpha & 0 \\ -\beta & \alpha & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Transformation matrix in readable form

approximated

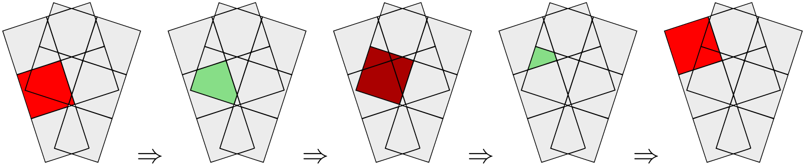
$$M(x,y,z,\alpha,\beta,\gamma) = \begin{pmatrix} 1 & -\gamma & \beta & x \\ \gamma & 1 & -\alpha & y \\ -\beta & \alpha & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

in our case, α and β and z are $\approx 0 \Rightarrow$ use time stamp as z coordinate

Final matrix

$$M(x,y,z,\alpha,\beta,\gamma) = \begin{pmatrix} 1 & -\gamma & 0 & x \\ \gamma & 1 & 0 & y \\ 0 & 0 & 1 & time \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

that way, matrix even tells time of flight between sensor layers



- go from sensor sensor 1 via matrix 1to2 to sensor 2
- go from sensor sensor 2 via matrix 2to3 to sensor 3

- matrix from 1to3 = 2to3 * 1o2

this can be done for all sensors ⇒ all sensors are aligned

Iterative Closest Point Algorithm

- algorithm to align two clouds of points in three dimensions (i.e. translation, rotation, scaling)
- here: determine transformation matrices from one sensor to every other sensor
- 9 overlapping areas $\rightarrow$ 9 matrices

$\Rightarrow$ position of every sensor with respect to reference sensor is known

Iterative Closest Point Algorithm

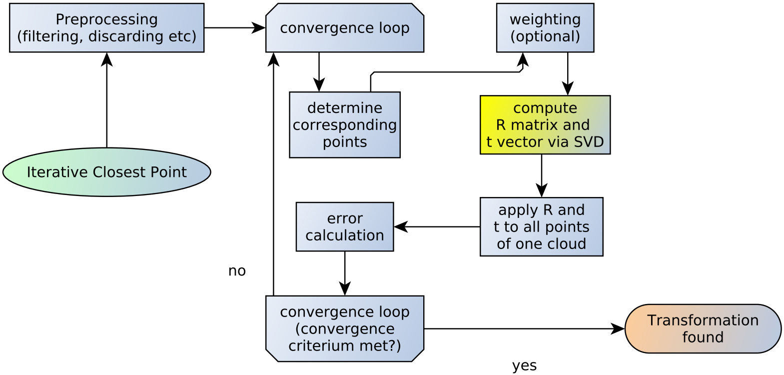
Inputs:

- points from two raw scans
- initial estimation of the transformation
- criteria for stopping the iteration

Output:

- refined transformation

Iterative Closest Point Algorithm



Singular Value Decomposition

compute
R matrix and
t vector via SVD

Singular Value Decomposition

- is a factorization of a matrix
- decompose transformation matrix
  $\rightarrow$ rotation + translation + scaling

Procedure

- subtract center of mass from every point
- point sets then are:

$$X' = x_i - \mu_x = x_i'$$
$$P' = p_i - \mu_p = p_i'$$

Singular Value Decomposition

- construct matrix W:

$$W = \sum_{i=1}^{N_p} x'_i p'^T_i \tag{1}$$

- and denote singular value decomposition (SVD) of W by:

$$W = U \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{pmatrix} V^T$$

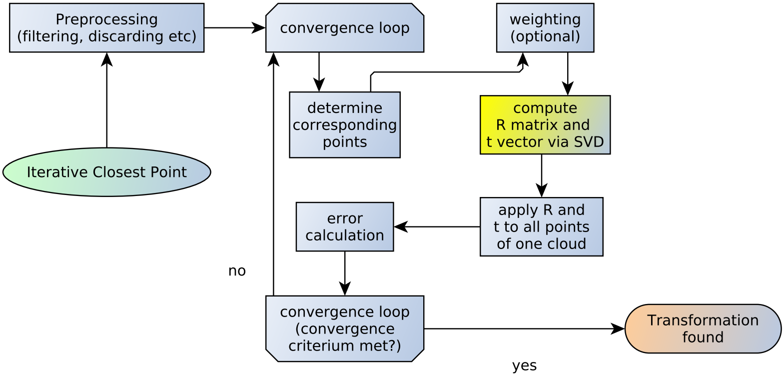
where $U, V \in \mathbb{R}^{3\times 3}$ are unitary and $\sigma_1, \sigma_2, \sigma_3$ are singular values of W

- rotation and translation are:

$$R = UV^T$$

$$t = \mu_x - R\mu_p$$

Iterative Closest Point Algorithm



Termination criteria

algorithm uses loop $\Rightarrow$ termination criterion required!

- maximum number of iterations achieved
- difference between M_i and $M_{i+1} <$ threshold

additional checks for quality:

- number of iterations < 4 (but only if initial matrix is known!)
- RMS distance (front pixel $\rightarrow$ back pixel) $< 0,5$ pixels

Testing

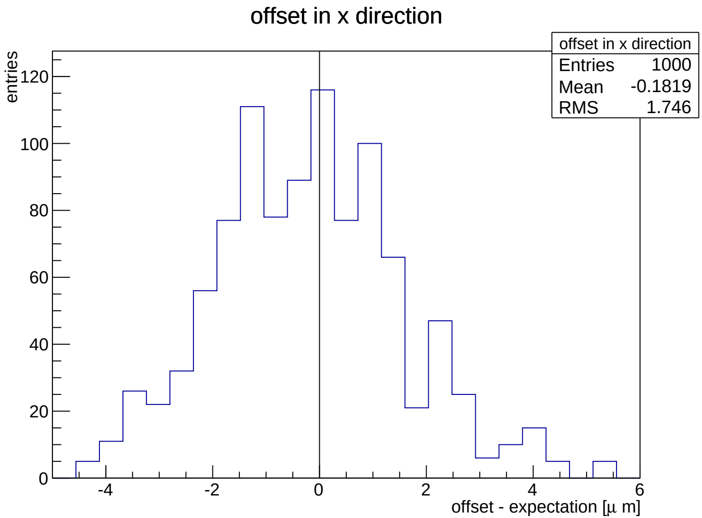
Pre-processing

- simulation of perfectly aligned sensors (no shift in x and y, no rotation)
- firing lots of events
- sorting to hit pairs (one pixel hit on front correlates to exactly one on back)
- event sorter must be very thorough!

Testing Procedure

- dividing available pairs to 1000 bunches (by bootstrapping)
- using $\approx$ 100.000 pixel pairs (best case) for ICP
- difference from ideal matrix should be small!

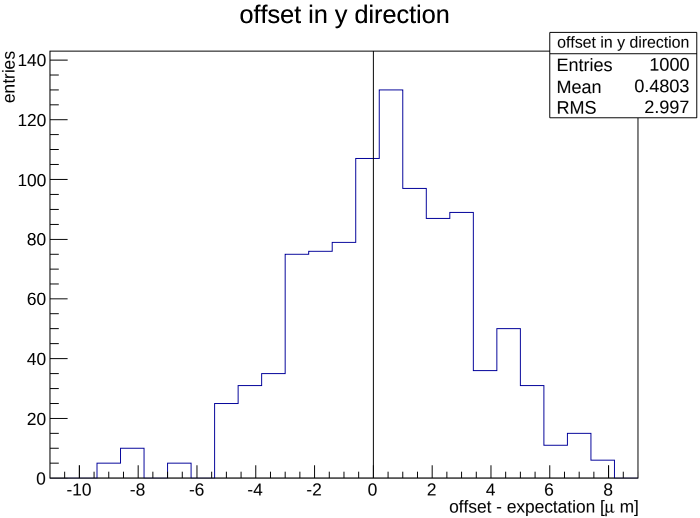
taking ≈ 10.000 elements from pool of ≈ 100.000 per entry



result should be as close to 0 as possible

resultant alignment, matrix 1 to 2

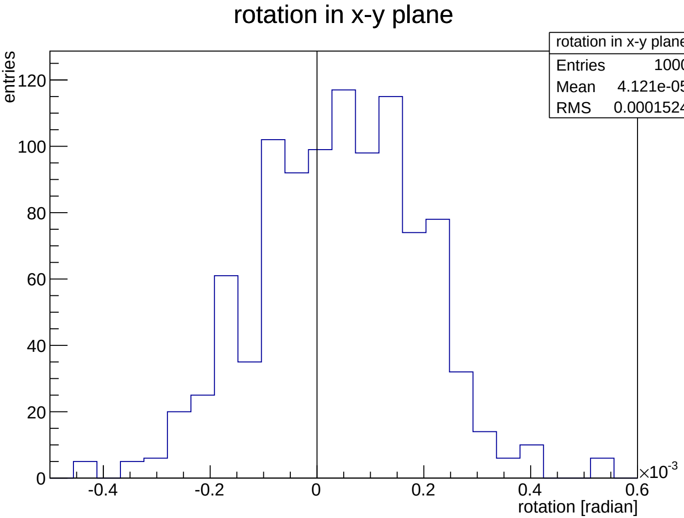
taking $\approx$ 10.000 elements from pool of $\approx$ 100.000 per entry



result should be as close to 0 as possible

resultant alignment, matrix 1 to 2

taking ≈ 10.000 elements from pool of ≈ 100.000 per entry



result should be as close to 0 as possible

That means:

Result:

- simulated geometry can be aligned on micrometer and sub-miliradian scale!

example for transformation from one sensor to its neighbour

direction	value	error
x	$-0.18\mu m$	$1.75\mu m$
y	$0.48\mu m$	$2.99\mu m$
rotation	$0.041 mrad$	$0.152 mrad$

Conclusion

Pixel coincidence method

- using charged particles tracks and responding pixels
- accuracy surpasses fraction of pixel length
- simultaneously firing pixels must be close (rough estimate of alignment must be known)
- high statistics $\rightarrow$ high accuracy

Iterative Closest Point Algorithm

- finds transformation matrix to transform one set of points onto a corresponding set of points
- quality heavily depends on quality of event pre-processing