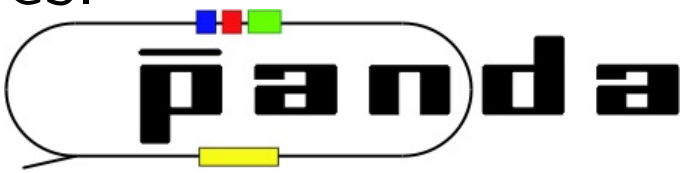
Benchmarking of P_z reconstruction in the STT

Walter Ikegami Andersson

Uppsala University
on behalf of the $\overline{\text{P}}$ANDA collaboration

$\overline{\text{P}}$ANDA Collaboration Meeting
June 24-28, 2019
GSI



Outline

- Recap on Pz recontruction algorithm
- New method: Recursive Annealing Fit
- Efficiency and Resolution
- Outlook

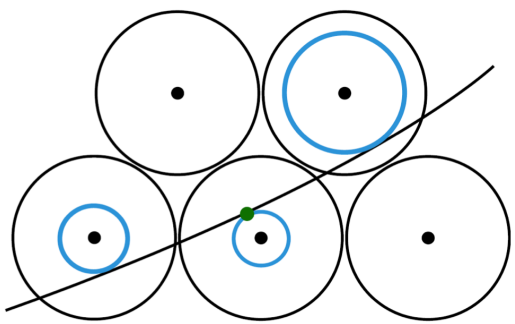
The $\overline{\text{P}}$ANDA Straw Tube Tracker

STT specifications	
Total straws	4224
Axial layers	15-19
Stereo layers	8
Stereo angle	$\pm$ 2.9 deg

Numbers taken from
STT design report

Isochrone radius

Radial distance from track to wire



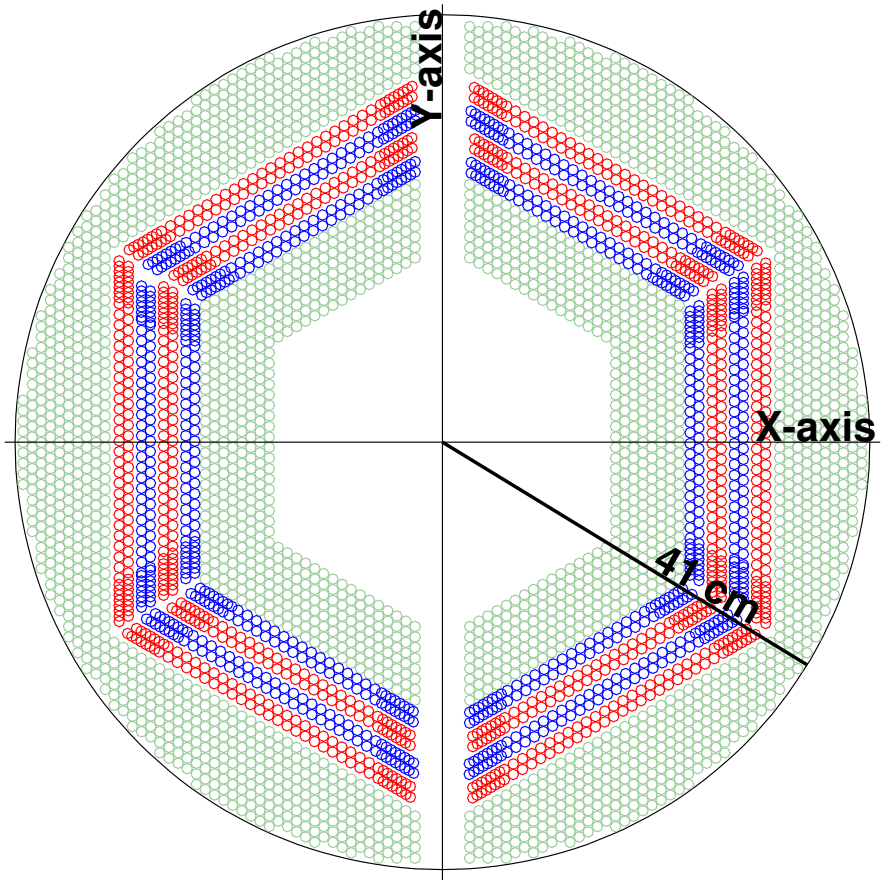


Figure: Cross sectional view of STT
Green - parallel straw
Red, blue - skewed straw

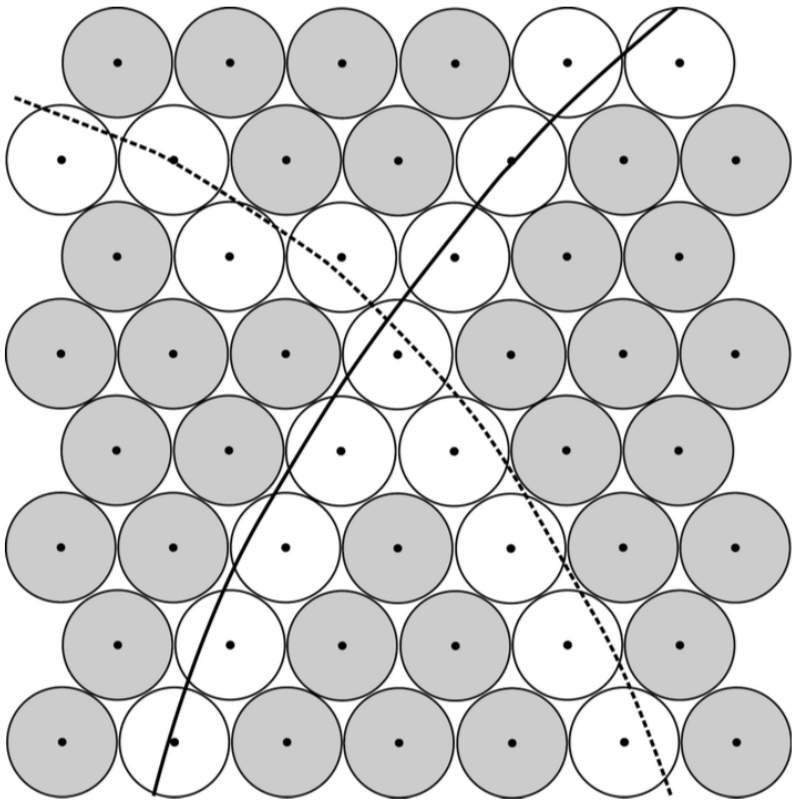
Tracking algorithm dedicated for STT

Track reconstruction algorithm using only STT.
(J. Schumann, Forschungszentrum Jülich)

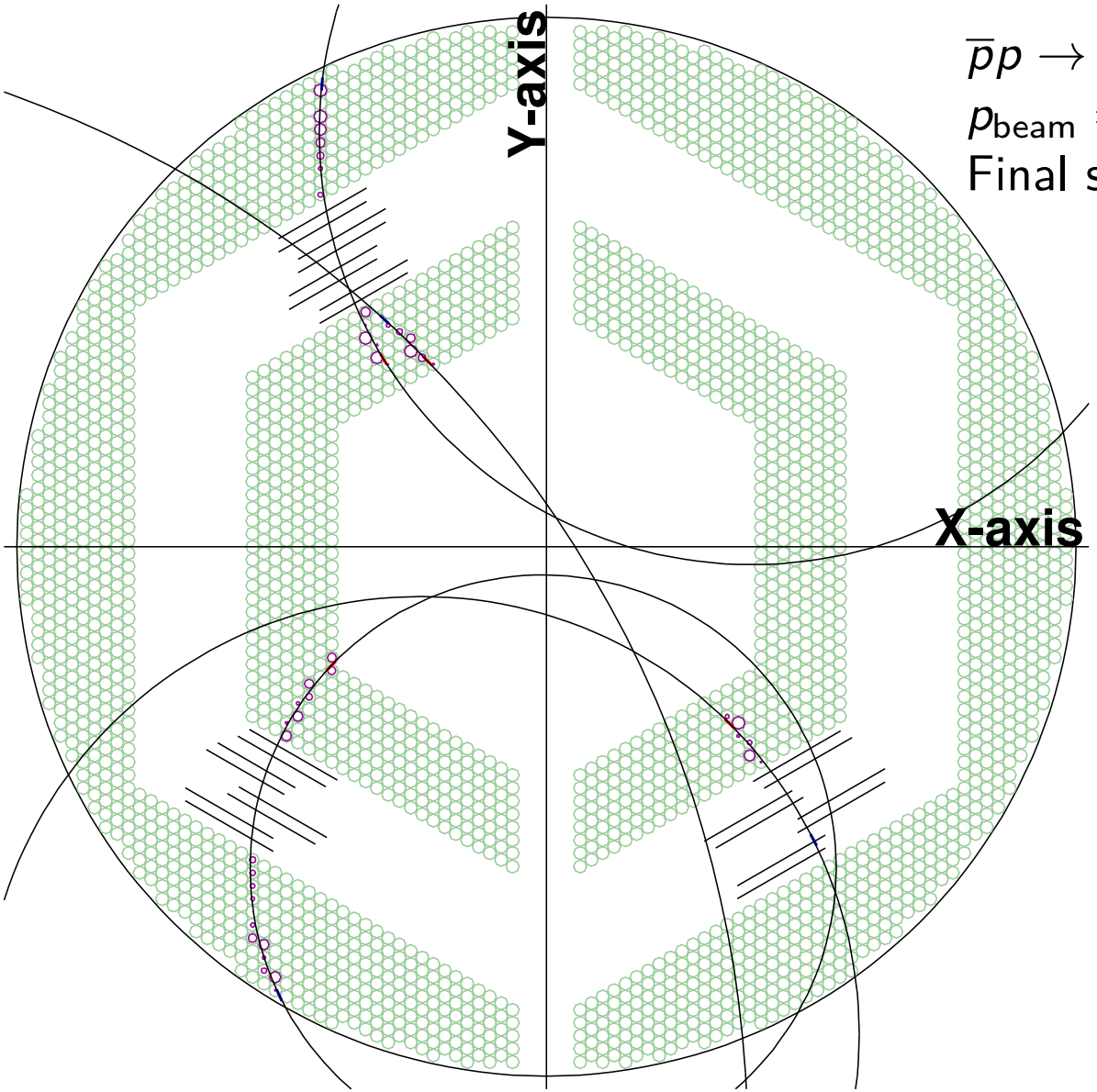
1. Cluster hits in parallel straws into tracklets (neighboring relations)
2. Riemann fit using isochrone corrected hits
3. Assign skewed straw hits to track

Output: Riemann track object for each track, containing circle fit in xy-plane

Must include skewed straws to reconstruct p_z

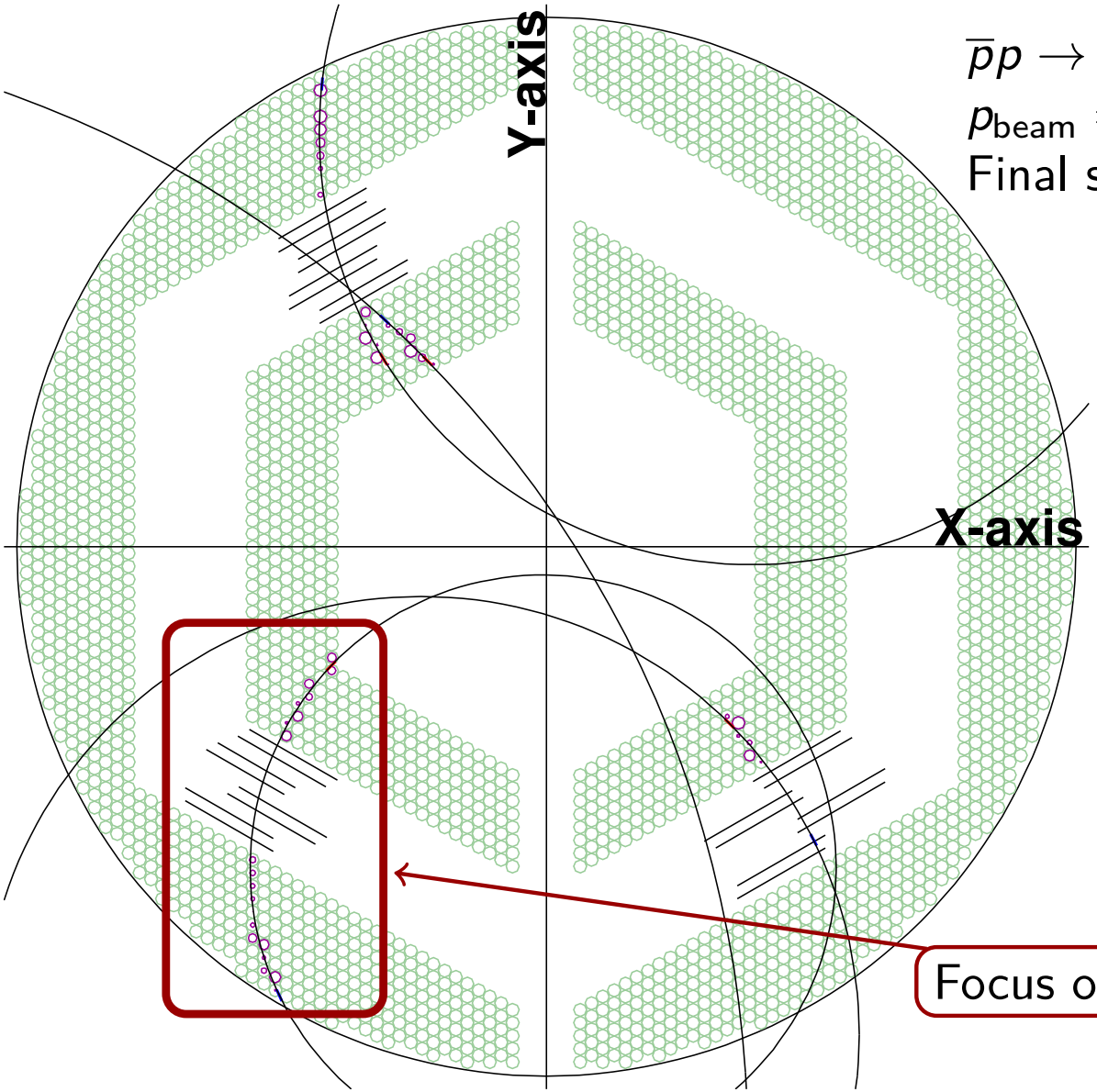


Longitudinal position from skewed straws



$\overline{p}p \rightarrow \overline{\Lambda}\Lambda$ generated at
$p_{\text{beam}} = 1.64$ GeV/c
Final state: $\overline{p}\pi^+ p\pi^-$

Longitudinal position from skewed straws



$\overline{p}p \to \overline{\Lambda}\Lambda$ generated at
$p_{\text{beam}} = 1.64$ GeV/c
Final state: $\overline{p}\pi^+ p\pi^-$

Longitudinal position from skewed straws

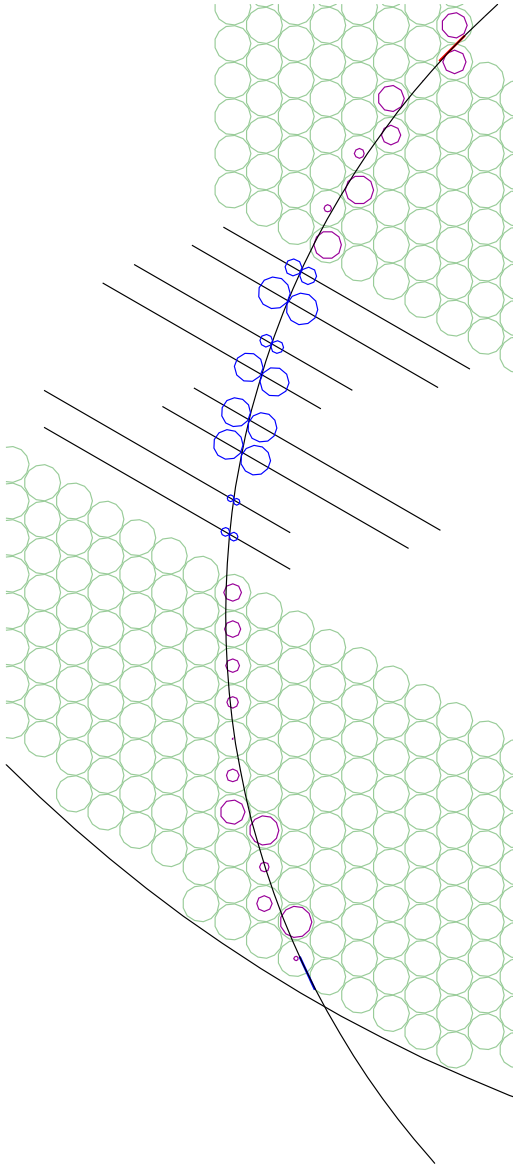
The method:

1. Extract isochrone radius in skewed straw
2. Center of isochrone gives $z-$position
3. Generate all possible isochrone positions
4. Calculate (z, ϕ)

Ambiguity: Each straw gives two possible (z, ϕ)

Solve ambiguity

Develop algorithms to find true collinear (z, ϕ) hits

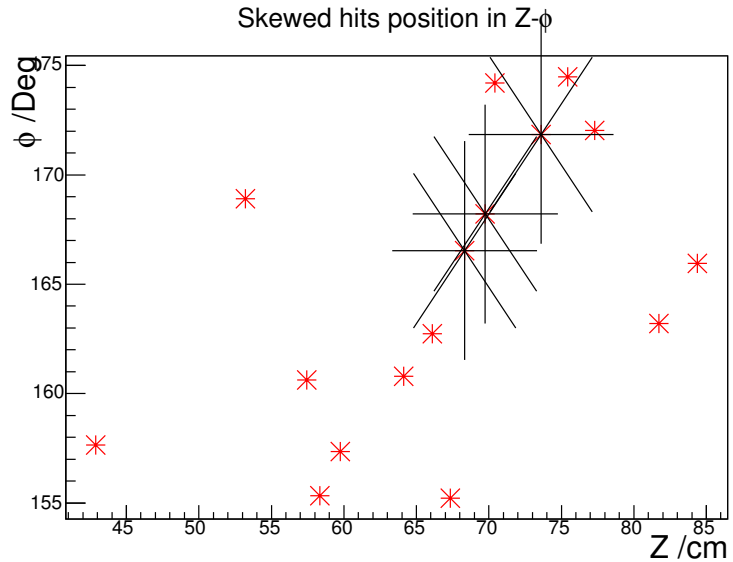


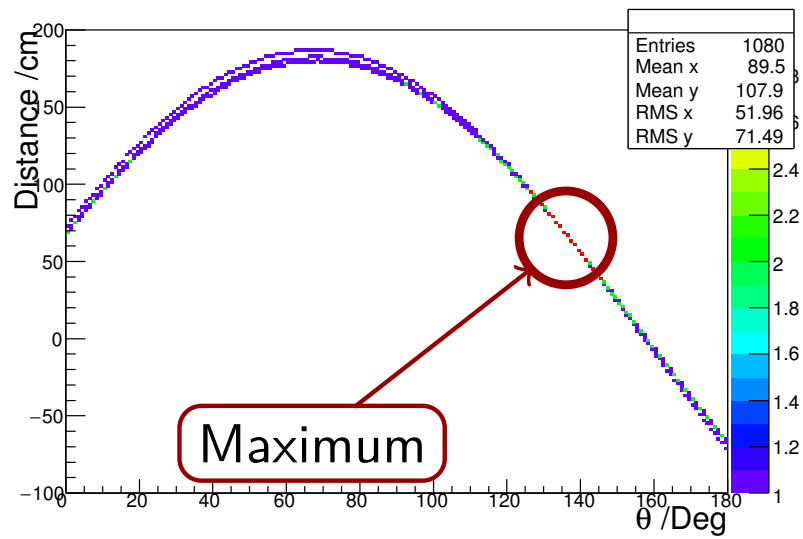
Method 1: Hough transform

The method:

1. Isochrone centers in $z - \phi$ space
2. Generate set of all lines
3. Parameters $\rightarrow$ accumulator space
4. Repeat for all points
5. Voting procedure $\rightarrow$ true line

True line found in maximum!



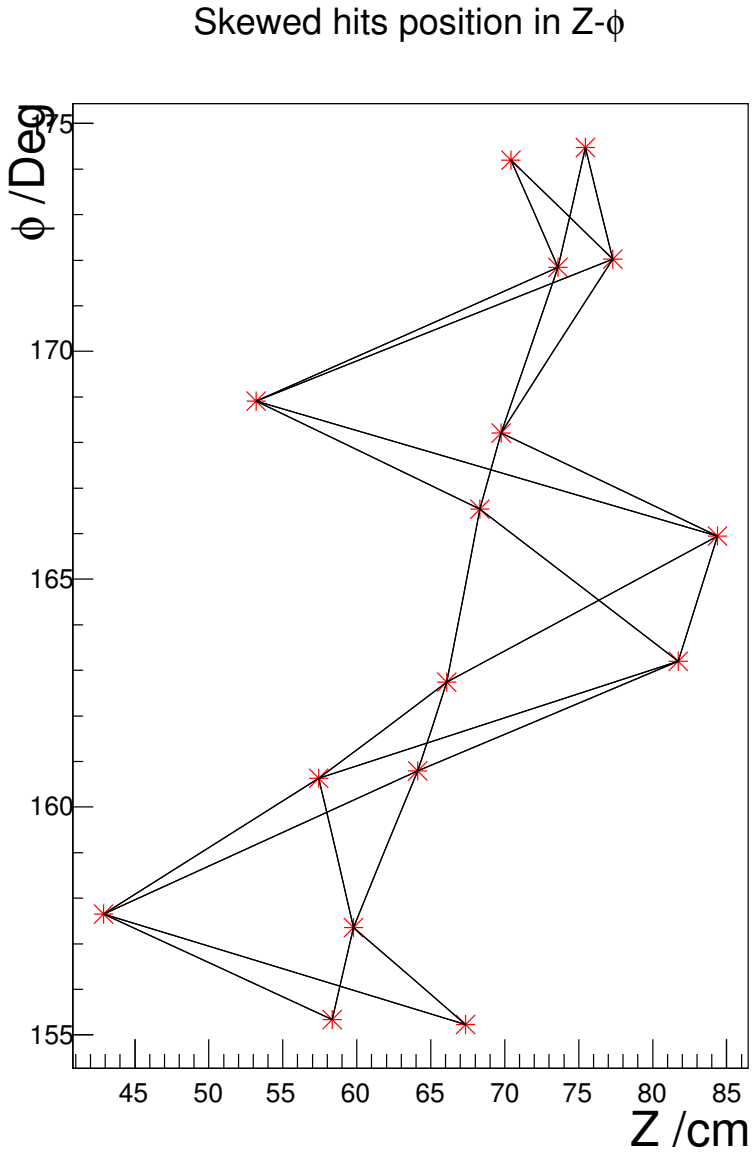


Method 2: Combinatorics

The method:

1. Calculate all lines between (z, ϕ) points in neighboring skewed straws
2. Calculate angle between all possible neighboring lines
3. Ignore paths where $\theta < 90°$ $\rightarrow$ reduces number of combinations
4. Choose path with $\min(\sum \theta_i - 180°)$

Hits in final path chosen as true hits



Method 3: Recursive Annealing Fit

The strategy:

- Fit a line to all (S, z) hits (MVD and STT)
- Remove (S, z) hit (only skewed STT) with largest z residual
- Make a new line fit
- Repeat until one (S, z) hit has been rejected in every skewed straw

Benchmarking: Quantities

Two quantities studied in the benchmarking:

- p_l resolution
  Done with the TrackingQA task of PandaRoot
- (S, z) selection efficiency

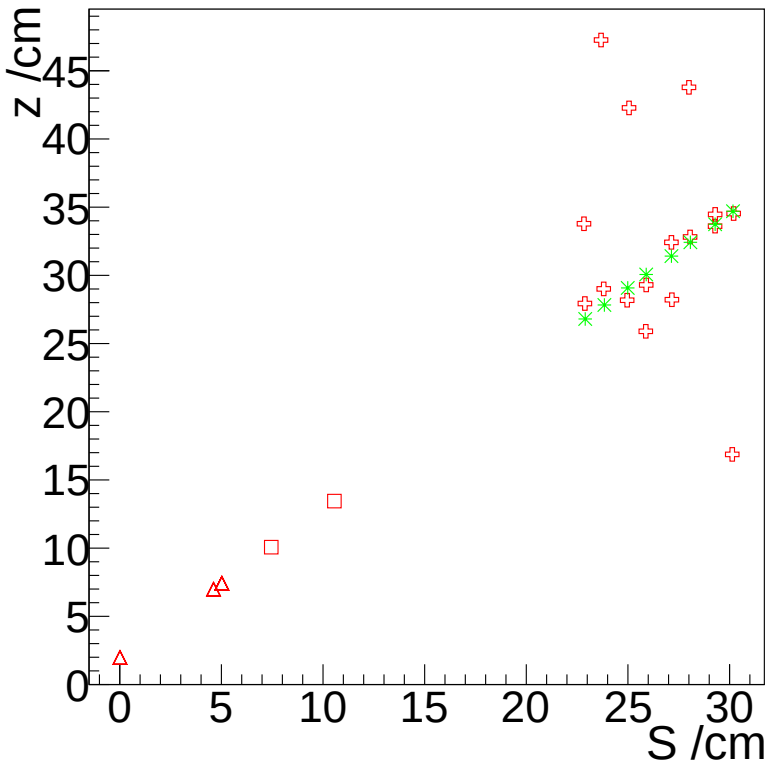
Benchmarking: Quantities

Two quantities studied in the benchmarking:

- p_l resolution
  Done with the TrackingQA task of PandaRoot
- (S, z) selection efficiency

Custom QA task developed for this purpose:

- For each skewed STT hit, get MC point
- Transform MC point from (x, y, z) to (S, z)
- Compare (S, z) (red) from alignment with MC point (green)
- Red point closest to MC point considered *true*



Benchmarking: Samples

The data samples used for benchmarked are the following:

- Two DPM samples generated at 2 GeV/c and 15 GeV/c
- At 2 GeV/c, solenoid field strength at 1 T

	DPM @ 2.0 GeV/c	DPM @ 15.0 GeV/c
Events	1000	1000
MC tracks	2339	2832

- Both Primary Track Finder and SttCellTrackFinder used to provide prefit tracks
- Primary Track Finder - study performance including MVD hits
- SttCellTrackFinder - study performance of local STT tracking

Benchmarking: Efficiency

	Combi. Path Finder		Hough Transform		Rec. Ann. Fit	
p_t method	Primary	SttCell	Primary	SttCell	Primary	SttCell
DPM @ 2.0 GeV/c	80.2	84.3	66.3	83.9	93.7	92.4
DPM @ 15 GeV/c	78.1	83.3	66.0	82.0	91.7	91.0

- Combinatorial Path Finder and Hough Transform have similar efficiencies $\sim 80\%$
- Low efficiency when using the Hough Transform with Primary Track Finder
- Recursive Annealing Fit highest efficiency with $> 90\%$

Benchmarking: Resolution

Resolution obtained from standard TrackingQA task:

- Calculate relative p_l error distribution
- Take FWHM as resolution parameter

	Combi. Path Finder		Hough Transform		Rec. Ann. Fit	
p_t method	Primary	SttCell	Primary	SttCell	Primary	SttCell
DPM @ 2.0 GeV/c	0.164	0.180	0.140	0.280	0.176	0.188
DPM @ 15 GeV/c	0.096	0.132	0.096	0.128	0.080	0.112

- Again Recursive Annealing Fit has best resolution, mostly
- For 2 GeV/c using Primary Track Finder, Hough Transform has best resolution.
  The same sample has the lowest efficiency

Summary and Outlook

- Three algorithms to solve left-right ambiguity in STT skewed straws have been developed
  - Combinatorial path finding
  - Hough Transform
  - Recursive Annealing Fit
- The Recursive Annealing Fit found to have highest efficiency
- Hough Transform found to have best p_l resolution at 2.0 GeV/c when MVD hits are included
- Recursive Annealing Fit could be extended to reject fake MVD hits as well

Summary and Outlook

- Three algorithms to solve left-right ambiguity in STT skewed straws have been developed
  - Combinatorial path finding
  - Hough Transform
  - Recursive Annealing Fit
- The Recursive Annealing Fit found to have highest efficiency
- Hough Transform found to have best p_l resolution at 2.0 GeV/c when MVD hits are included
- Recursive Annealing Fit could be extended to reject fake MVD hits as well

Thank you for your attention!

Backup

Method 1: Hough transform

Find geometric shapes in images.

- Helix trajectory $\rightarrow$ straight line in $z - \phi$ space
- Line parameters in xy-plane, slope k and intercept m
  - $y(x) = kx + m$

Problem: The intercept parameter m unbound.

Hesse normal form

$$r = x\cos\theta + y\sin\theta$$

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r}{\sin\theta}\right)$$

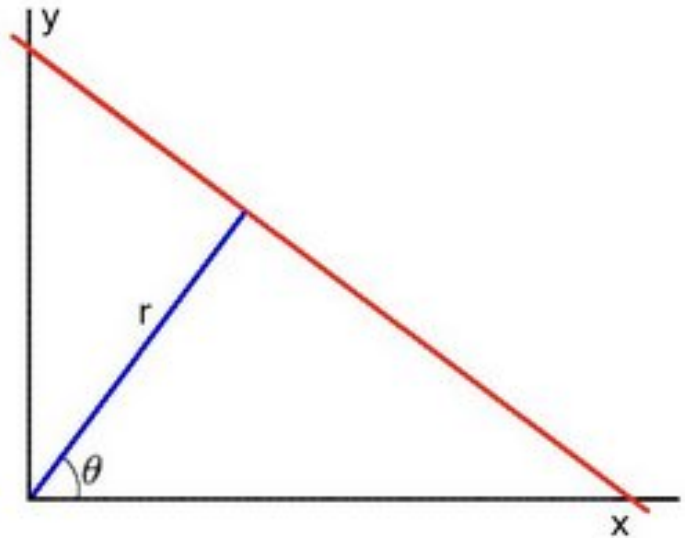
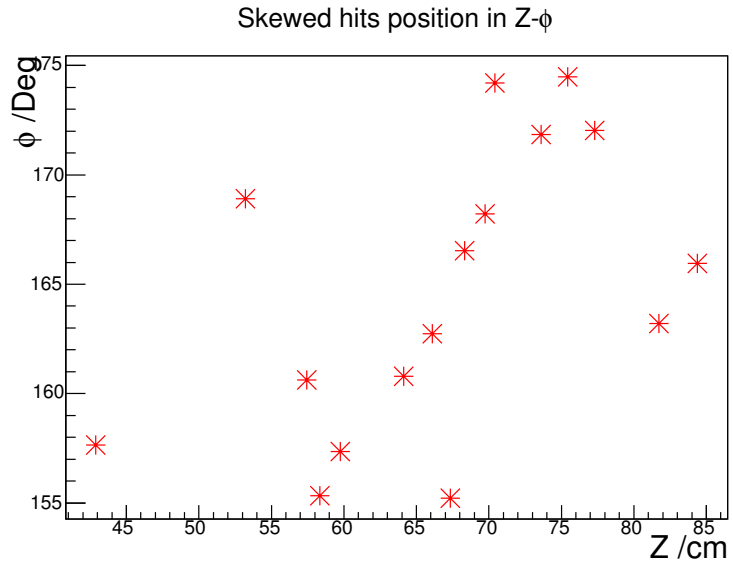


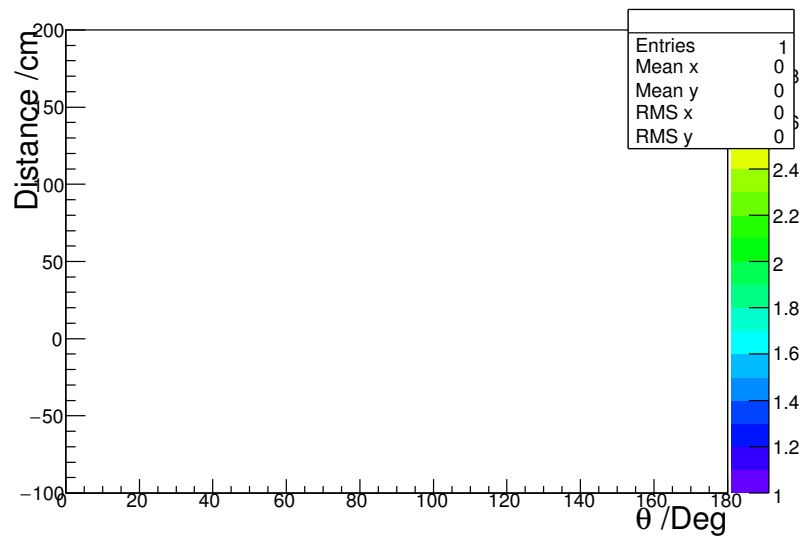
Figure: Blue line perpendicular to red line and crosses the origin

Method 1: Hough transform

The method:

1. Isochrone centers in $z - \phi$ space

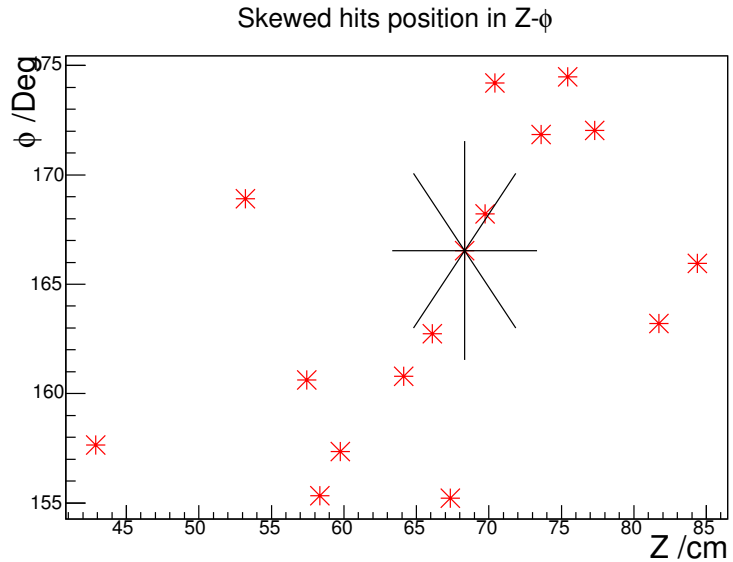


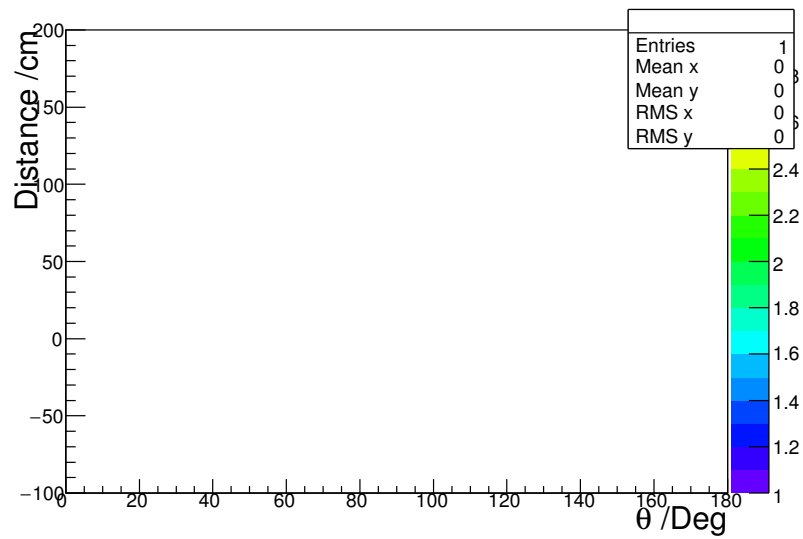


Method 1: Hough transform

The method:

1. Isochrone centers in $z - \phi$ space
2. Generate set of all lines

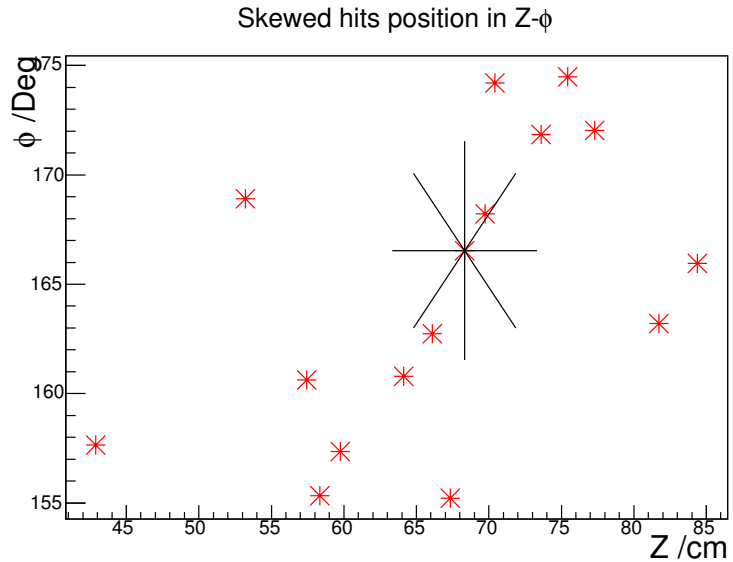


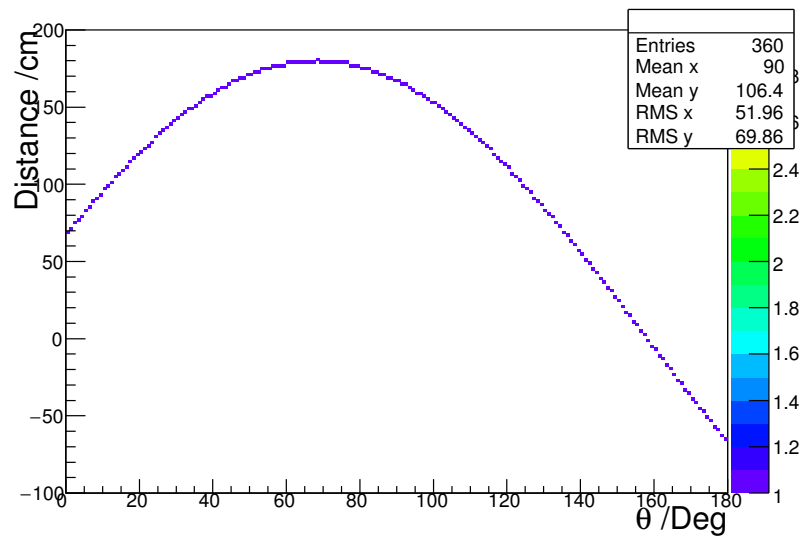


Method 1: Hough transform

The method:

1. Isochrone centers in $z - \phi$ space
2. Generate set of all lines
3. Parameters $\rightarrow$ accumulator space

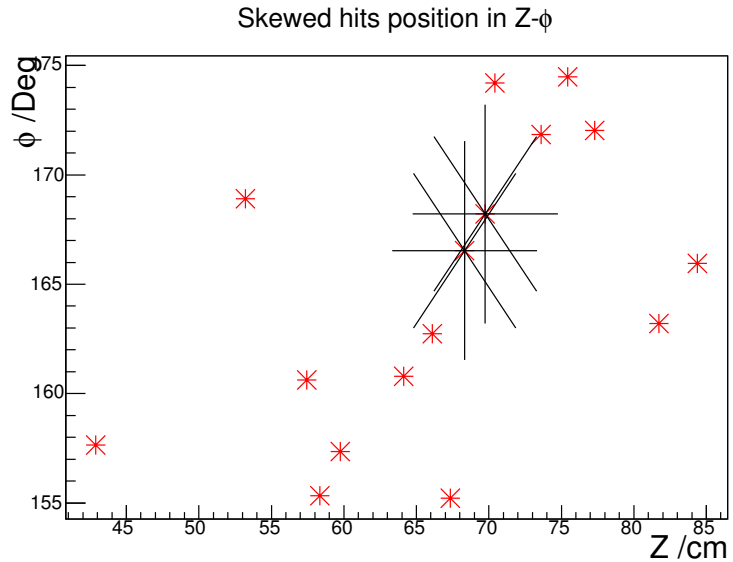


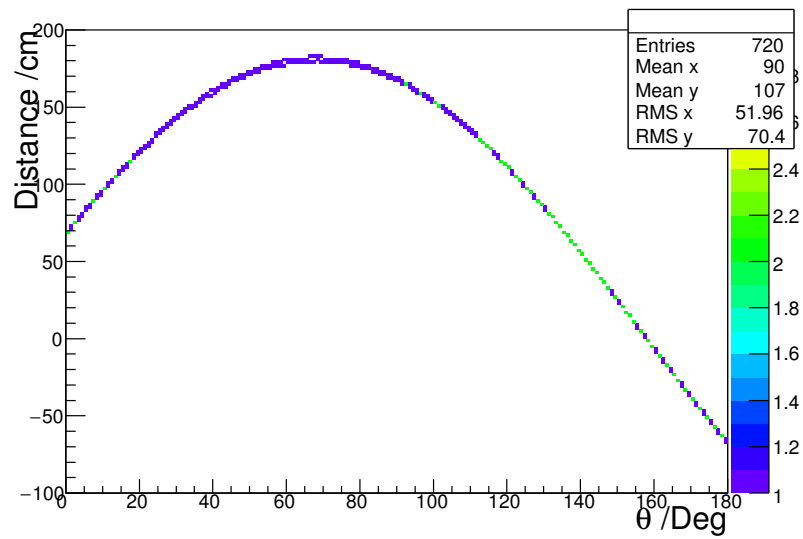


Method 1: Hough transform

The method:

1. Isochrone centers in $z - \phi$ space
2. Generate set of all lines
3. Parameters $\rightarrow$ accumulator space
4. Repeat for all points

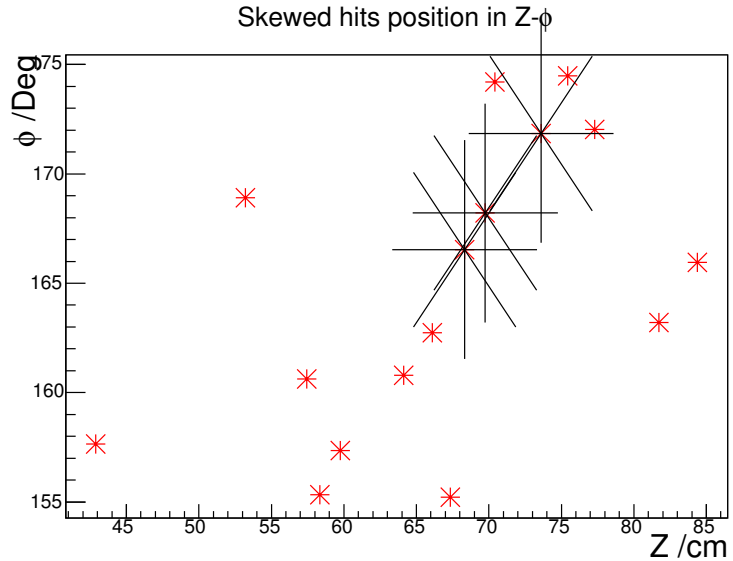


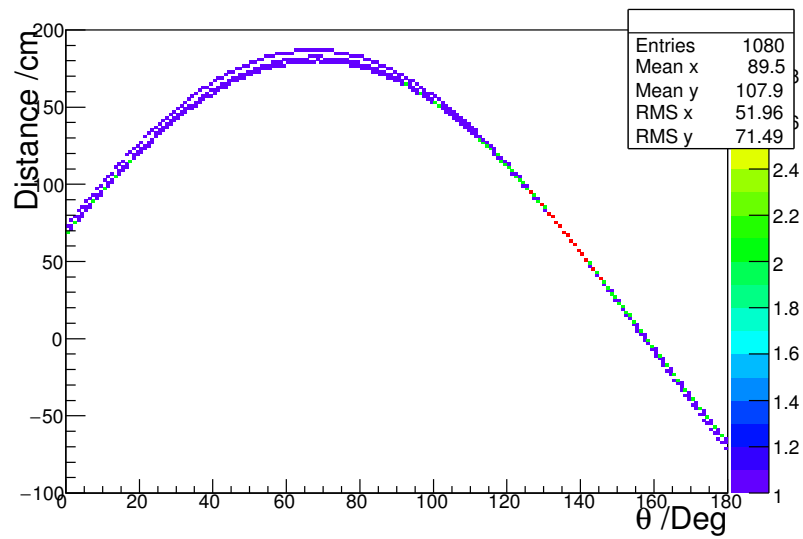


Method 1: Hough transform

The method:

1. Isochrone centers in $z - \phi$ space
2. Generate set of all lines
3. Parameters $\rightarrow$ accumulator space
4. Repeat for all points



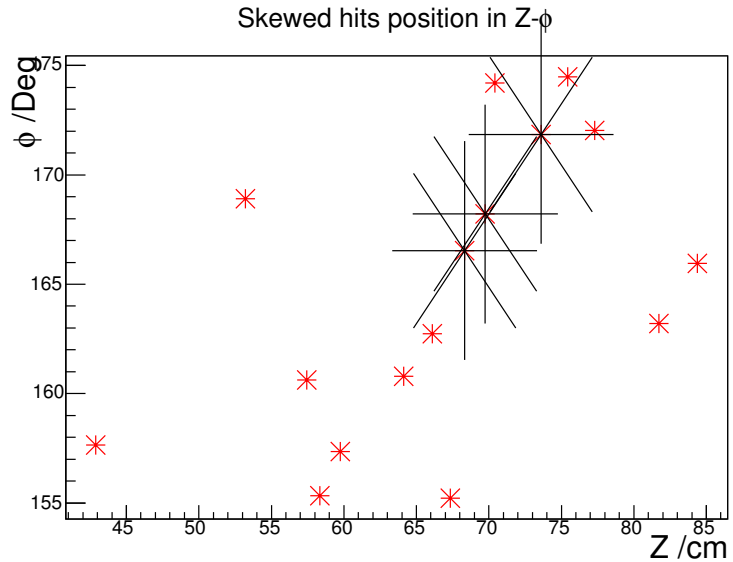


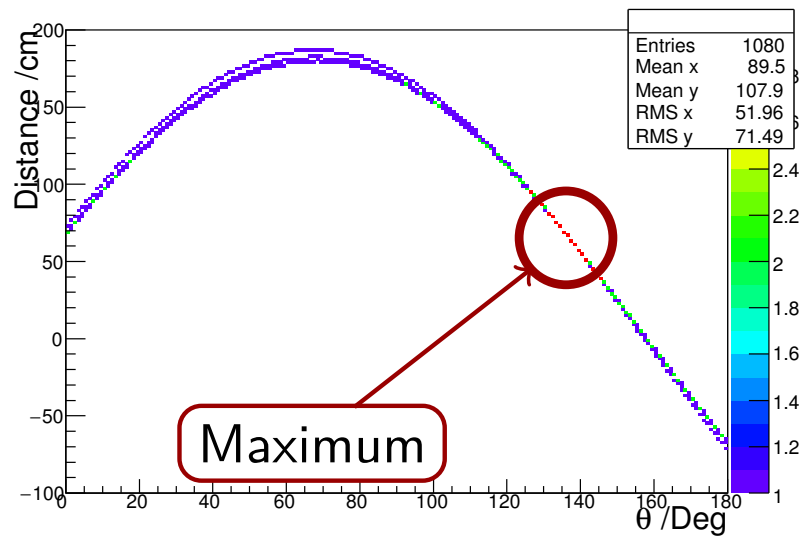
Method 1: Hough transform

The method:

1. Isochrone centers in $z - \phi$ space
2. Generate set of all lines
3. Parameters $\rightarrow$ accumulator space
4. Repeat for all points
5. Voting procedure $\rightarrow$ true line

True line found in maximum!





Method 1: Hough transform - our track

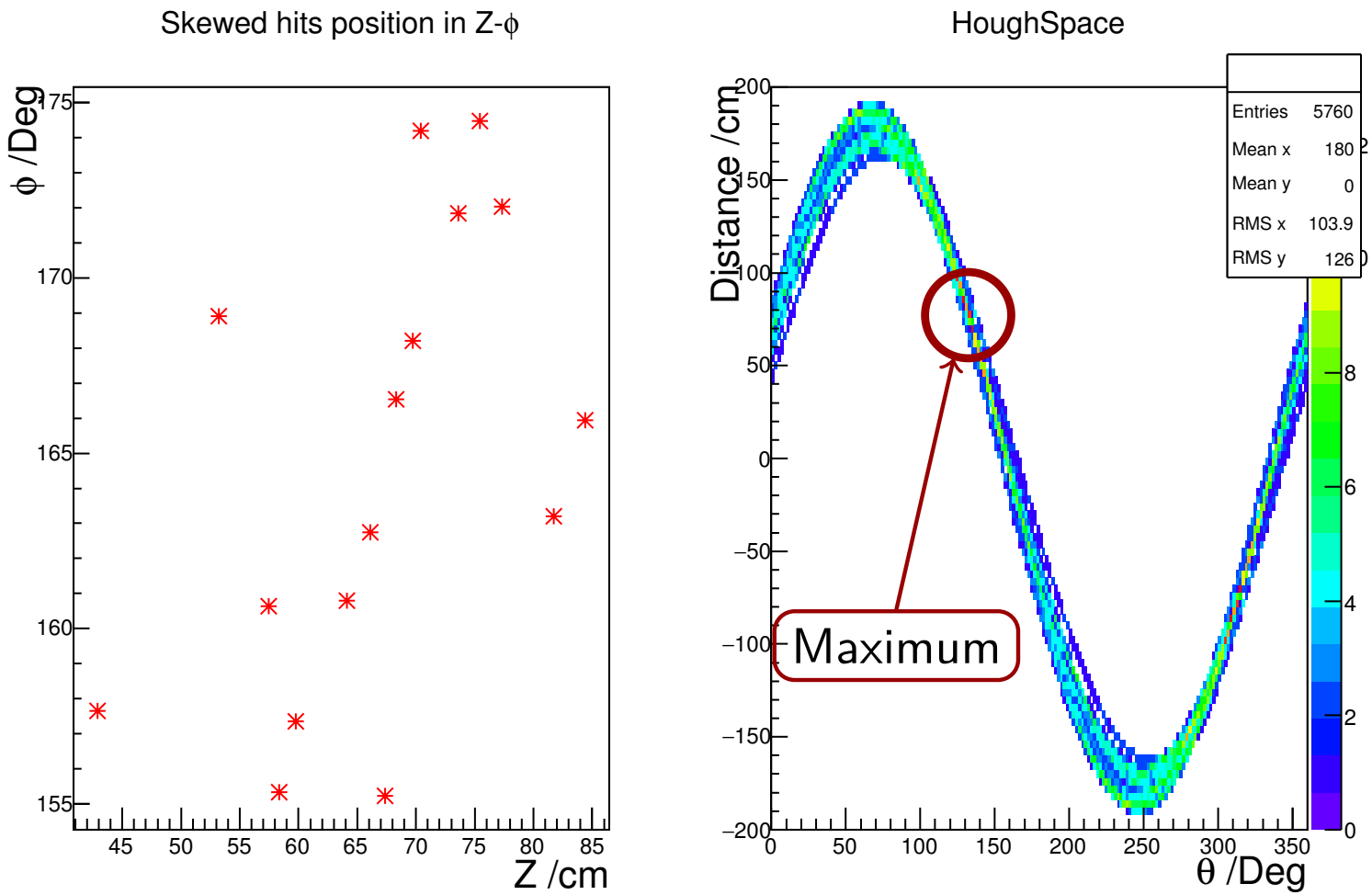


Figure: 360 lines generated for each data point in steps of 1° in θ

Method 1: Extracting helix angle

The method:

1. Calculate point of closest approach (POCA) from hits to true line
2. Accept hit with smallest POCA
3. Straight line fit with selected (z, ϕ) coordinates

Finish

The slope of the fitted line yields the helix angle. z_0 and p_z can now be extracted!

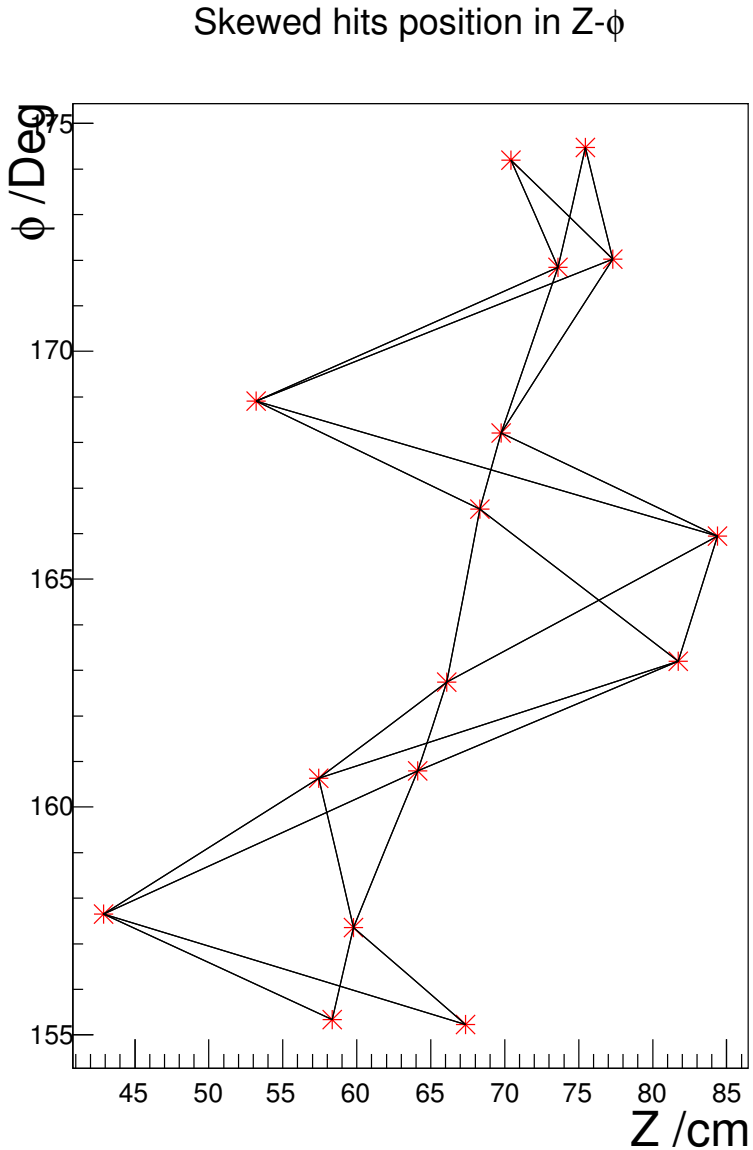
Method 2: Combinatorics

The method:

Method 2: Combinatorics

The method:

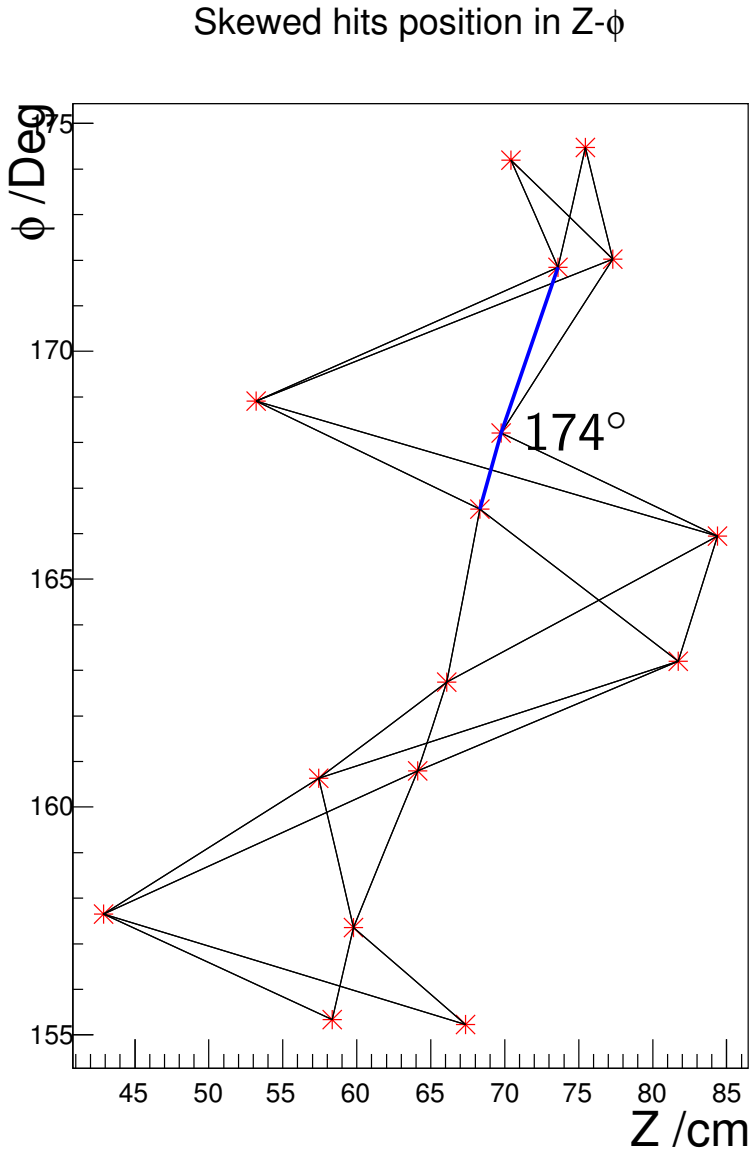
1. Calculate all lines between (z, ϕ) points in neighboring skewed straws



Method 2: Combinatorics

The method:

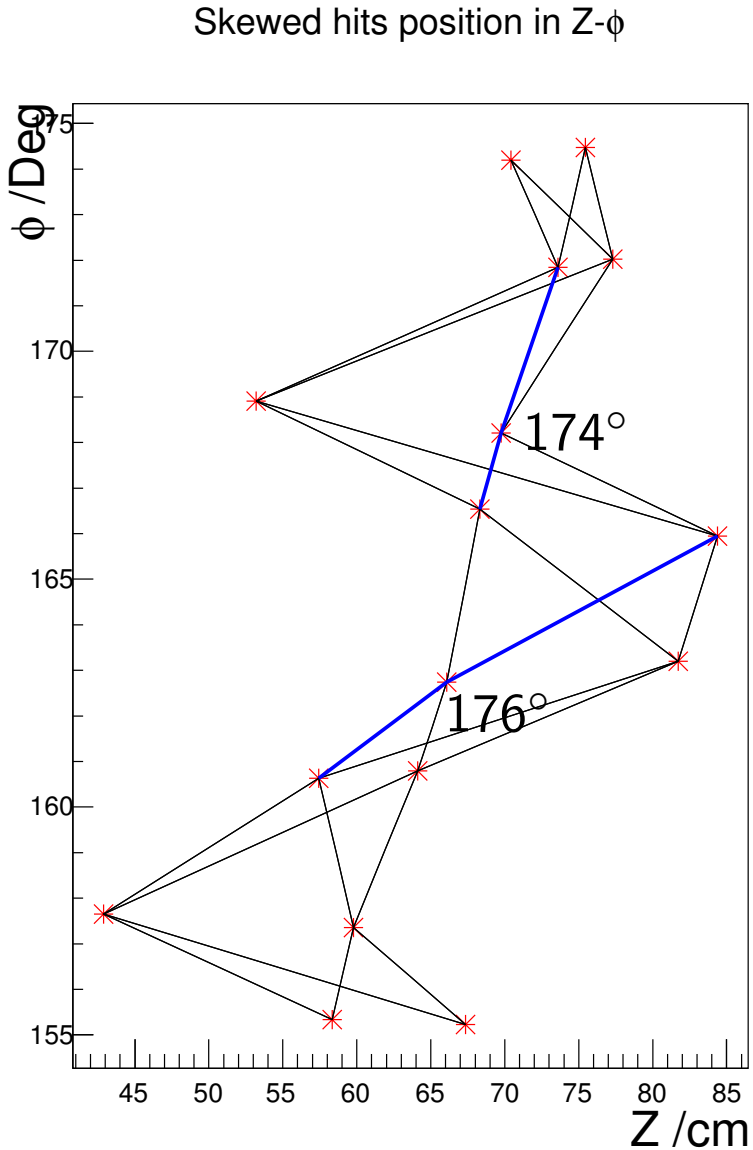
1. Calculate all lines between (z, ϕ) points in neighboring skewed straws
2. Calculate angle between all possible neighboring lines



Method 2: Combinatorics

The method:

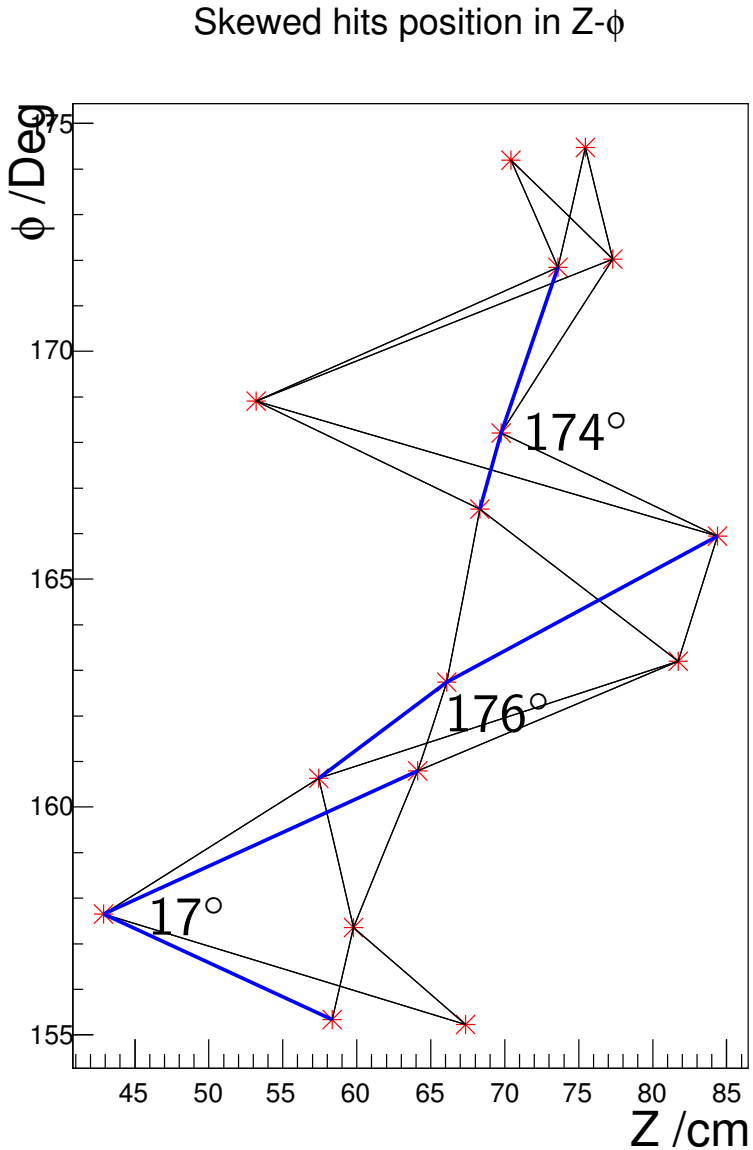
1. Calculate all lines between (z, ϕ) points in neighboring skewed straws
2. Calculate angle between all possible neighboring lines



Method 2: Combinatorics

The method:

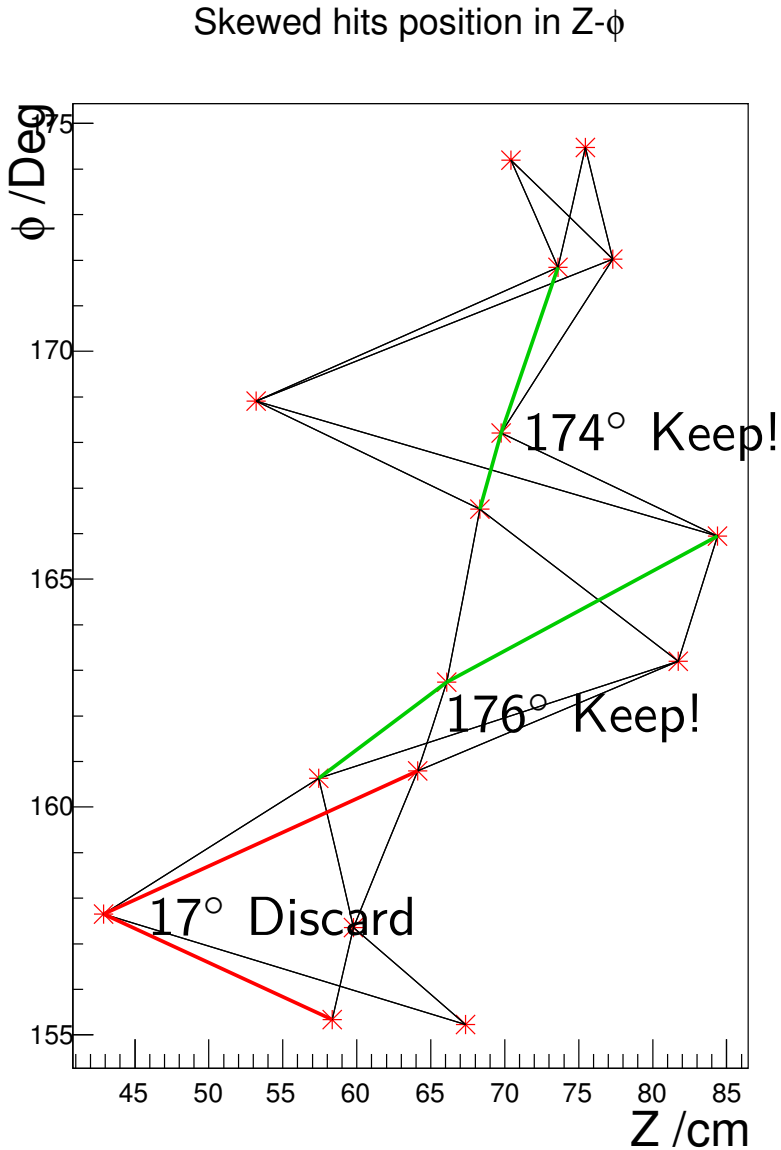
1. Calculate all lines between (z, ϕ) points in neighboring skewed straws
2. Calculate angle between all possible neighboring lines



Method 2: Combinatorics

The method:

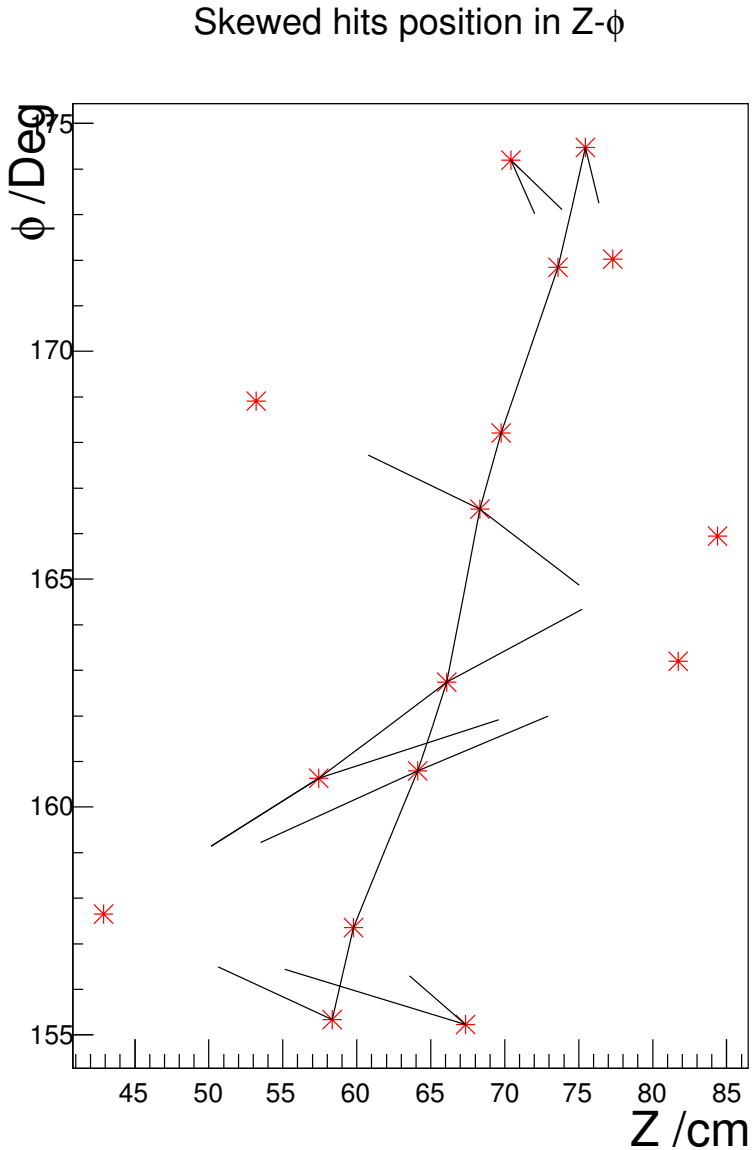
1. Calculate all lines between (z, ϕ) points in neighboring skewed straws
2. Calculate angle between all possible neighboring lines
3. Ignore paths where $\theta < 90°$ $\rightarrow$ reduces number of combinations



Method 2: Combinatorics

The method:

1. Calculate all lines between (z, ϕ) points in neighboring skewed straws
2. Calculate angle between all possible neighboring lines
3. Ignore paths where $\theta < 90°$ $\rightarrow$ reduces number of combinations

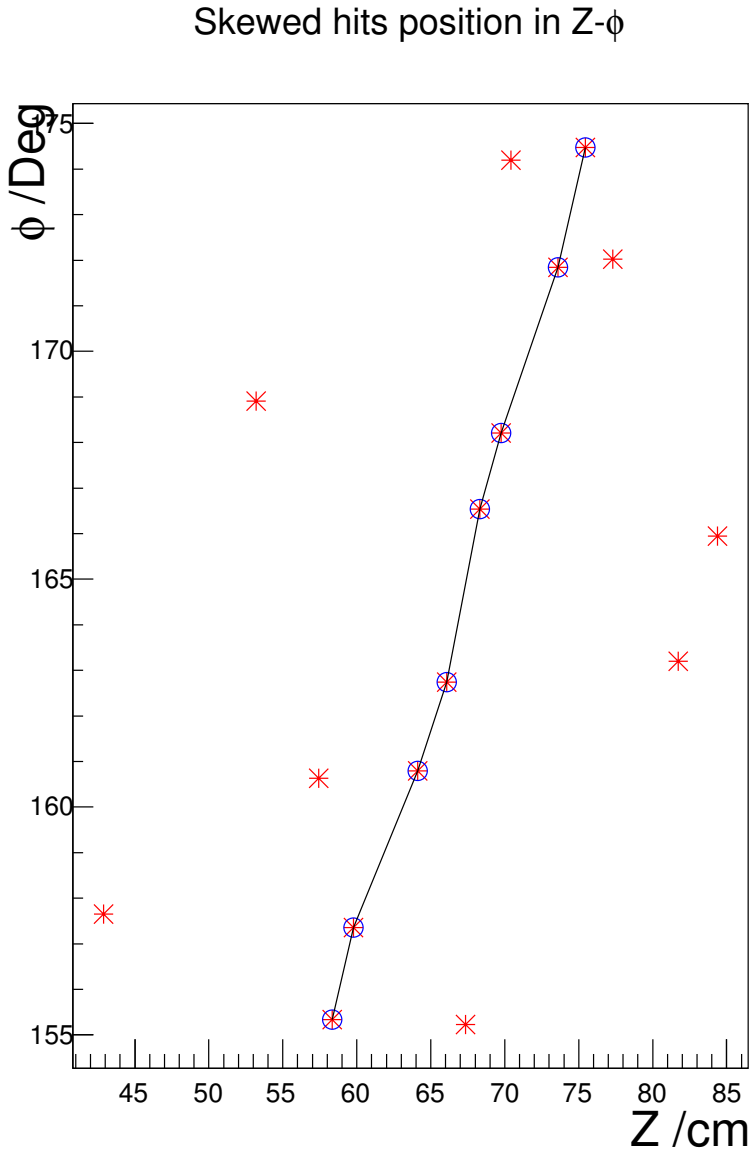


Method 2: Combinatorics

The method:

1. Calculate all lines between (z, ϕ) points in neighboring skewed straws
2. Calculate angle between all possible neighboring lines
3. Ignore paths where $\theta < 90°$ $\rightarrow$ reduces number of combinations
4. Choose path with $\min(\sum \theta_i - 180°)$

Hits in final path chosen as true hits



PzFinder - Code structure

- PndSttSkewStrawPzFinderTask.cxx
  - PndTrack - Standard $\overline{\text{P}}$ANDA track object
  - PndTrackCand - PndSttHits belonging to track
  - PndRiemannTrack - Riemann circle parameters to track
- PndSttSkewStrawPzFinder.cxx
  - MoveSkewedHitstoCircle
    - Calculates all possible (z, ϕ) in skewed straw
  - HoughTrueIsoFinder
    - Fills accumulator space, find maximum, rejects fake hits with POCA
  - LineCombiIsoFinder
    - Generates lines, calculates angles, find best path
  - PzLineFitExtract
    - Simple line fit to true (z, ϕ) hits and extracts helix angle