A first look at telescope track finding with fast Hough transformation

BTTB Workshop '15, 19.-21.01.15

C. Kleinwort - DESY

Overview

- Introduction
  - Track finding
  - Hough transformation
- Fast Hough Transformation (FHT)
- Application to telescope data

Track finding (I)

- Track finding (or pattern recognition)
  - For each particle in the detector find back the set of hits (measurements) produced by that particle
  - It's an art
- Track fitting
  - For set of measurements determine optimal set of track parameters (usually linear least squares)
  - It's a technique

Track finding (II)

- Two general approaches
- Global: "Top-Down"
  - Start with all hits in the event
  - Try to subdivide into smaller structures
  - Potentially large combinatorics (e.g. noise)
  - Robust against missing hits (e.g. low efficiency)
- Local: "Bottom-Up"
  - Build small structures from few close by hits
  - Try to combine into larger structures

Hough transformation (I)

- A global method based on the transformation of measurements into a (track) parameter space
- Track finding uses usually simple track models, e.g. for (locally) constant magnetic field:
  - A line in a plane: 2 parameters
  - A circle in a plane: 3 parameters
  - A helix in space: 5 parameters
- In (2-5 dimensional) parameter space a subspace of "acceptable" tracks is defined

Hough transformation (II)

- Each measurement defines a manifold in this subspace
  - Set of all track parameter combinations compatible with the measurement
- Intersection of two manifolds
  - These track parameters describe both measurements
- Track finding means to look for multiple intersections in the parameter subspace

Hough transformation (III)

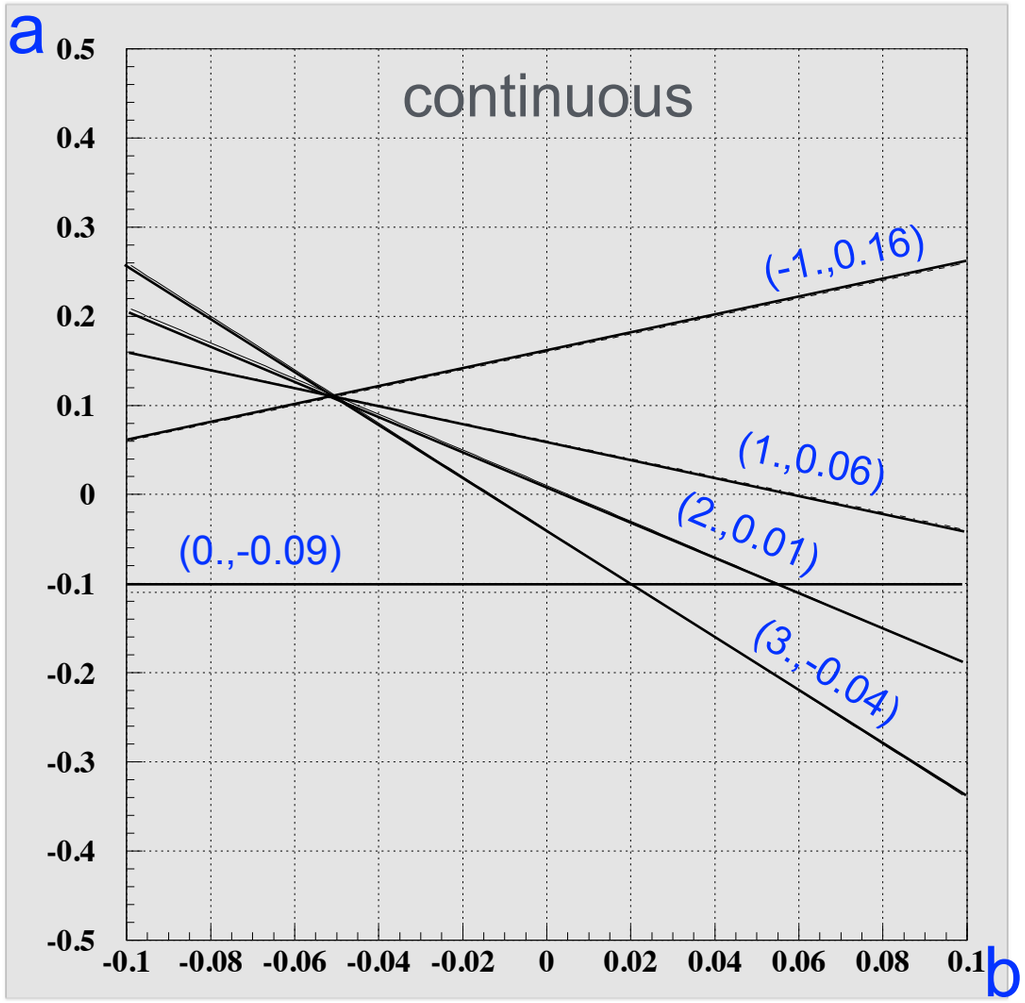
- In practice the subspace in quantised ("binned")
  - n-dimensional histogram
  - Each bin contains the number of manifolds (measurements) passing the bin
  - Track finding means to look for (significant) peaks in this histogram
- Result
  - Lists of measurements with coarse track parameter estimates (from bin center)

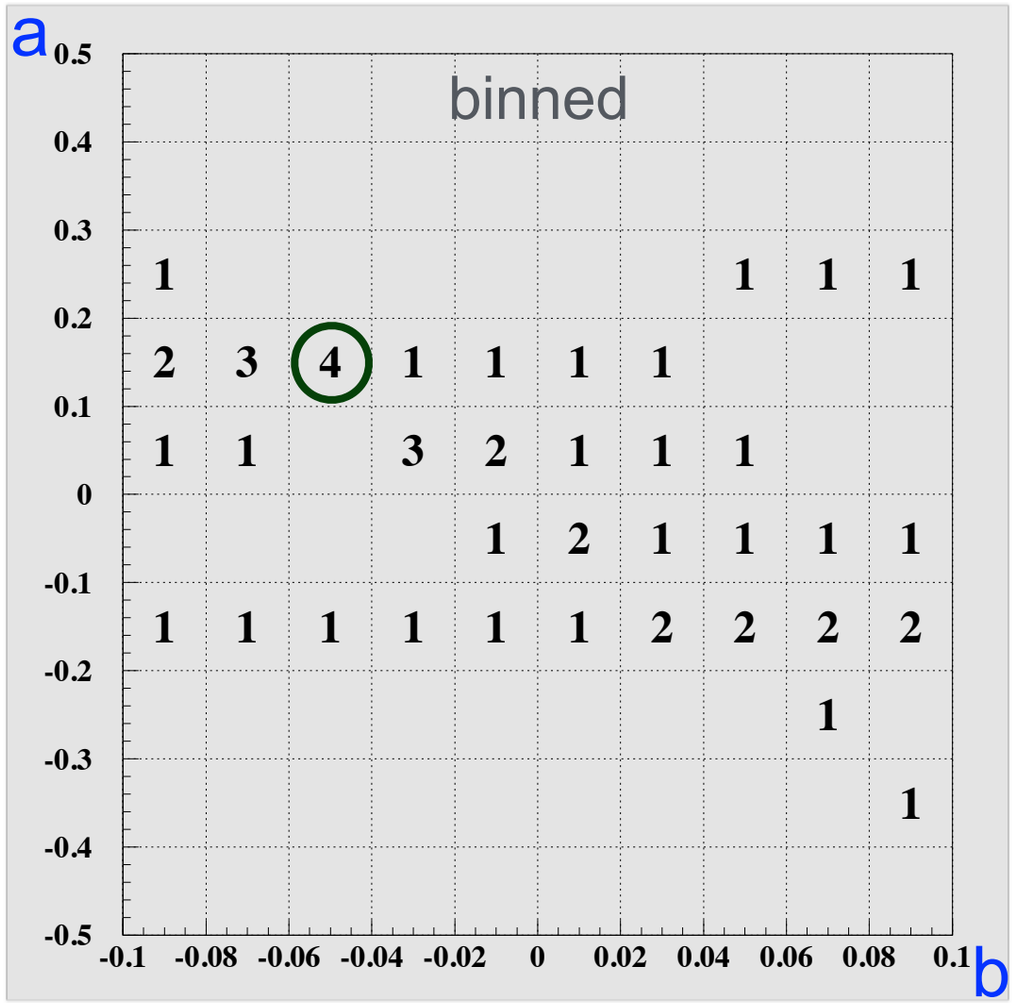
Hough transformation (IV)

- ★ Example in XY plane
  - Particles move straight and roughly in x direction
  - Track model: $y(x) = a + b \cdot x$
  - Detector measures points (x_i, y_i)
  - Each measurement defines line in 2D parameter space (a,b): $a(x_i, y_i, b) = y_i - b \cdot x_i$
  - "Toy" measurements
    - (-1.,0.16), (0.,-0.09), (1.,0.06), (2.,0.01), (3.,-0.04)
  - Look for tracks with
    - a in [-0.5, 0.5], b in [-0.1, 0.1], 10 bins each

Hough transformation (V)

Hough subspace (a,b) from the 5 toy measurements

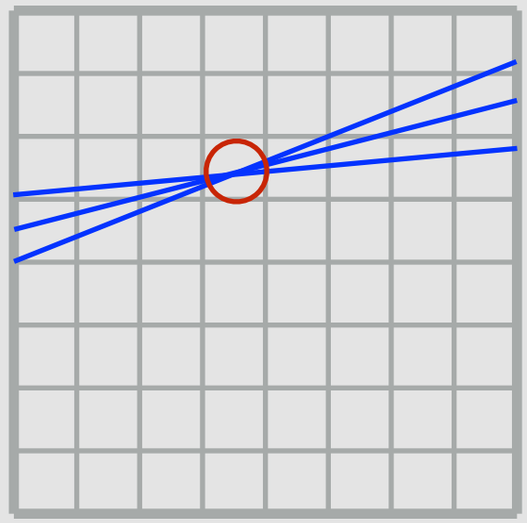




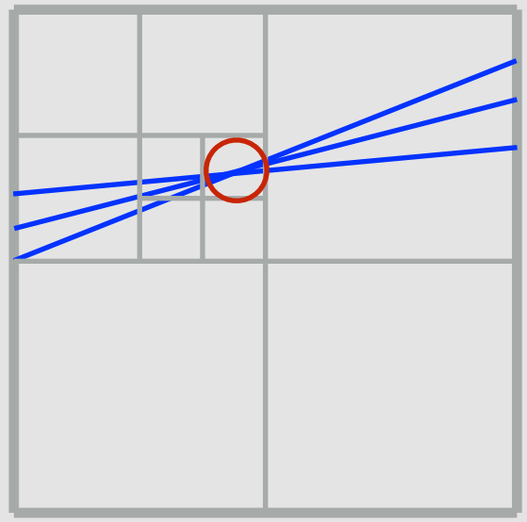
Found track with 4 measurements in bin at (a,b)=(0.15, -0.05)

Fast Hough transformation

- Patterns (measurements) are transformed into (manifolds in) parameter space
- Usual parameter space
  - large number of bins
  - searching for peaks
- Fast Hough Transformation (FHT)
  - start with few bins, e.g. one
  - subdivide (only) bins with some minimal content



8^2=64 bins



$3 \cdot 2^2$=12 bins

Implementation

- Based on:
  - H. Li et al., Fast Hough Transform: A Hierarchical Approach, Computer Vision, Graphics and Image Processing, **36**, 139-161 (1986)
- Hypercubes
- Hyperplanes
- Sequence
- Splitting level

Hypercubes

- The n-dimensional parameter space is mapped into the unit hypercube: $-0.5 < p_i < 0.5$ (i=1..n)
- Will be split in each dimension into 2 child cubes
- For each child the patterns with intersecting transforms have to be determined
- Subdivide recursively cubes with minimal content until child contains valid solution (single track)

Hyperplanes (I)

- Restrict to patterns described by hyperplanes (**n**,ρ) in parameter space $\rho = \mathbf{p} \cdot \mathbf{n} = \sum p_i \cdot n_i$
  - Unit normal vector **n**
  - Distance of closest approach ρ, use d = -ρ
- Simple intersection test with unit hypercube
  - For point of closest approach (ρ·**n**) maximal component < 0.5: $\max\left(\left|\rho \cdot n_i\right|\right) = \left|d\right| \cdot \max\left(\left|n_i\right|\right) < 0.5$
- Transform d (-ρ) into child 'i' as unit hypercube
  - With child at $\mathbf{c}_i$: $d_i = \underbrace{2 \cdot d}_{\text{scale}} + \underbrace{\mathbf{n} \cdot \mathbf{c}_i}_{\text{shift}}$

Hyperplanes (II)

- ⋆ Parametrisation for beam telescope (**B**=(0,B_y,0))
  - ‣ Tracks move mainly in z direction, use x(z), y(z)
- ⋆ Linear range adjustment (x,y,z) → (v,w,u)
- ⋆ Use series of Legendre polynomials $\sum p_i \cdot L_i(u)$
  - ‣ L_i project range [-1, +1] to [-1, +1] $\Longrightarrow$ u in [-1, +1]
    - ✦ Parameters p_i in [-0.5, +0.5] $\Longrightarrow$ v, w in [-0.5, +0.5]
  - ‣ X meas. (bending plane) $v(u) = \sum_{i=0}^{i=2} p_{i+1} \cdot L_i(u)$ (parabola)
  - ‣ Y measurement $w(u) = \sum_{i=0}^{i=1} p_{i+4} \cdot L_i(u)$ (line)
  - ‣ L_i are orthogonal $L_0(u) = 1,\ L_1(u) = u,\ L_2(u) = 1.5 \cdot u^2 - 0.5$

Hyperplanes (III)

- Description of hit (space point)
  - By pair of hyperplanes in 5 (4@B=0) dimensional parameter space

Sequence (I)

- ★ Input
  - The positions of all (up to now) unused hits
- ★ Preprocessing of input
  - Count number of detector planes, stop if too few
  - Get min(z), max(z) for dynamic scaling of z to u
  - Static scaling of x,y to v,w according to steering
- ★ For each hit construct hyperplanes
  - v(u): $\mathbf{l}_v = (L_0(u), L_1(u), L_2(u), 0., 0.),\ \mathbf{n}_v = \mathbf{l}_v / |\mathbf{l}_v|,\ d_v = -v / |\mathbf{l}_v|$
  - w(u): $\mathbf{l}_w = (0., 0., 0., L_0(u), L_1(u)),\ \mathbf{n}_w = \mathbf{l}_w / |\mathbf{l}_w|,\ d_w = -w / |\mathbf{l}_w|$

Sequence (II)

- ⋆ Construct root hypercube
  - From hits where both hyperplanes intersect
- ⋆ Recursively split hypercube
  - For all child cubes check intersection with hyperplane pairs including some overlap to avoid binning effects: $|d| < 0.75 / \max(|n_i|)$
  - Consider only child cubes containing some minimum number of planes (relative to total number, e.g. 80%)

Sequence (III)

- Sort child cubes by decreasing number of planes and increasing distance spread $\langle d^2 \rangle$
  - Don't want to analyse all of them, but most promising first
- If minimum (splitting) level (e.g. 5) reached
  - Calculate hit density (#hits/track length in planes)
  - Stop and return list of hits in cube if:
    efficiency (0.8) < density < purity (1.2)
- If maximum (splitting) level (e.g. 8) not reached
  - Split child cube (unless max #cubes (e.g. 250) exceeded)

Sequence (IV)

- If some candidate has been found
  - Remove hits from list of unused ones
  - Iterate with first step (counting planes …)
- Postprocessing of candidates
  - E.g. planes with multiple hits: skip, clean up, ..

Splitting level

- Defines resolution
  - $\Delta v = \Delta w \sim 1 / 2^{level}$
- Higher level
  - improved 2 track resolution, noise rejection
  - but start to loose hits due to distortions (e.g. misalignment)
- Need compromise
  - E.g. with $x_{range} = y_{range} = 20$ mm level = 5 corresponds to $\Delta x = \Delta y = 0.18$ mm

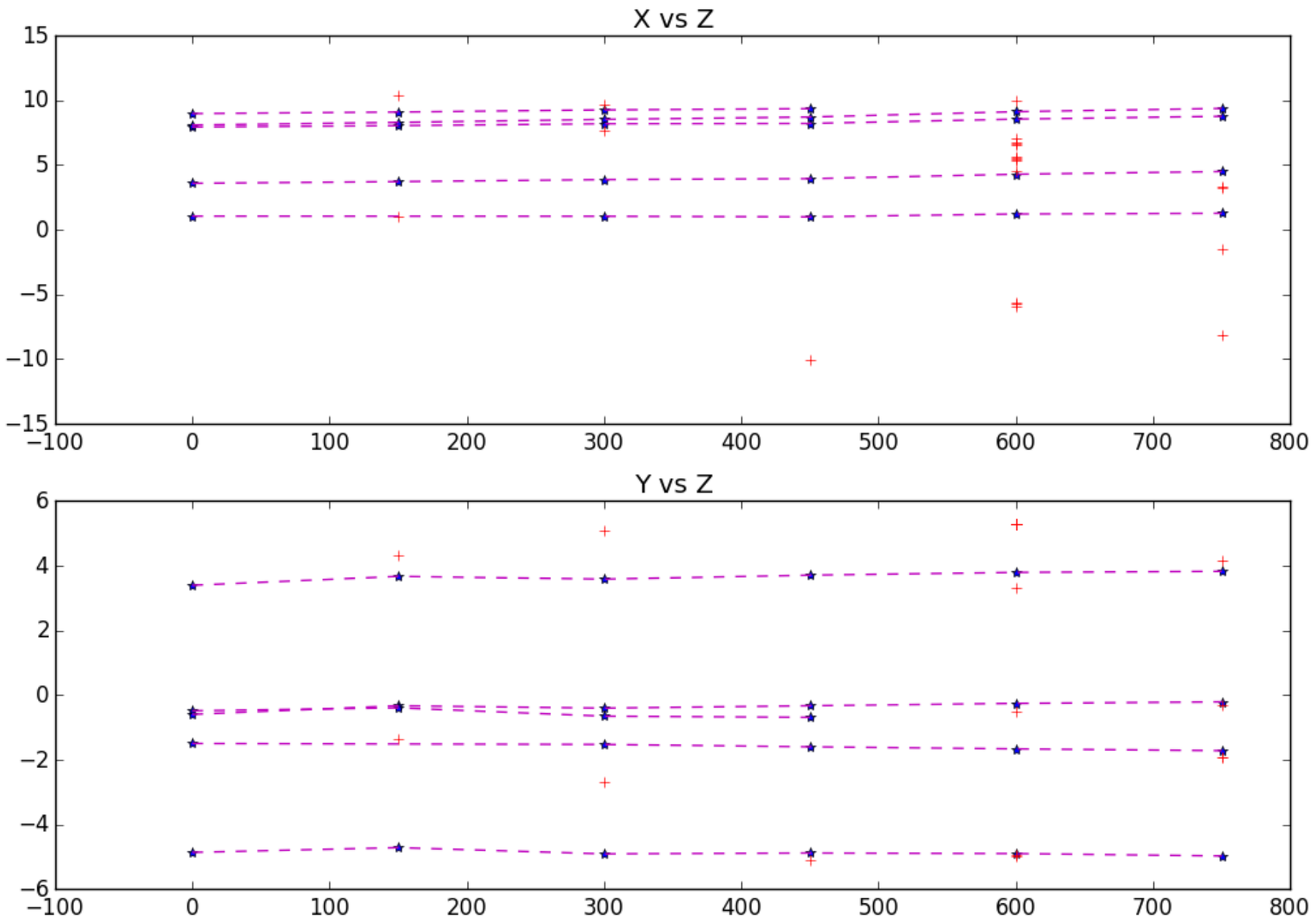
FHT for telescope data (I)

- FHT in ILCSOFT
  - MarlinTPC processor `(RowBasedFastHoughTransformation)` implements track finding with FHT for the LPTPC (TPC prototype for ILD in DESY test beam 24/1)
  - Described in lcnote LC-TOOL-2014-006
  - Assumes track propagation in x direction and constant magnetic field in z direction
  - Could be used after coordinate transformation (x,y,z) to (z,x,y) for telescope data too

FHT for telescope data (II)

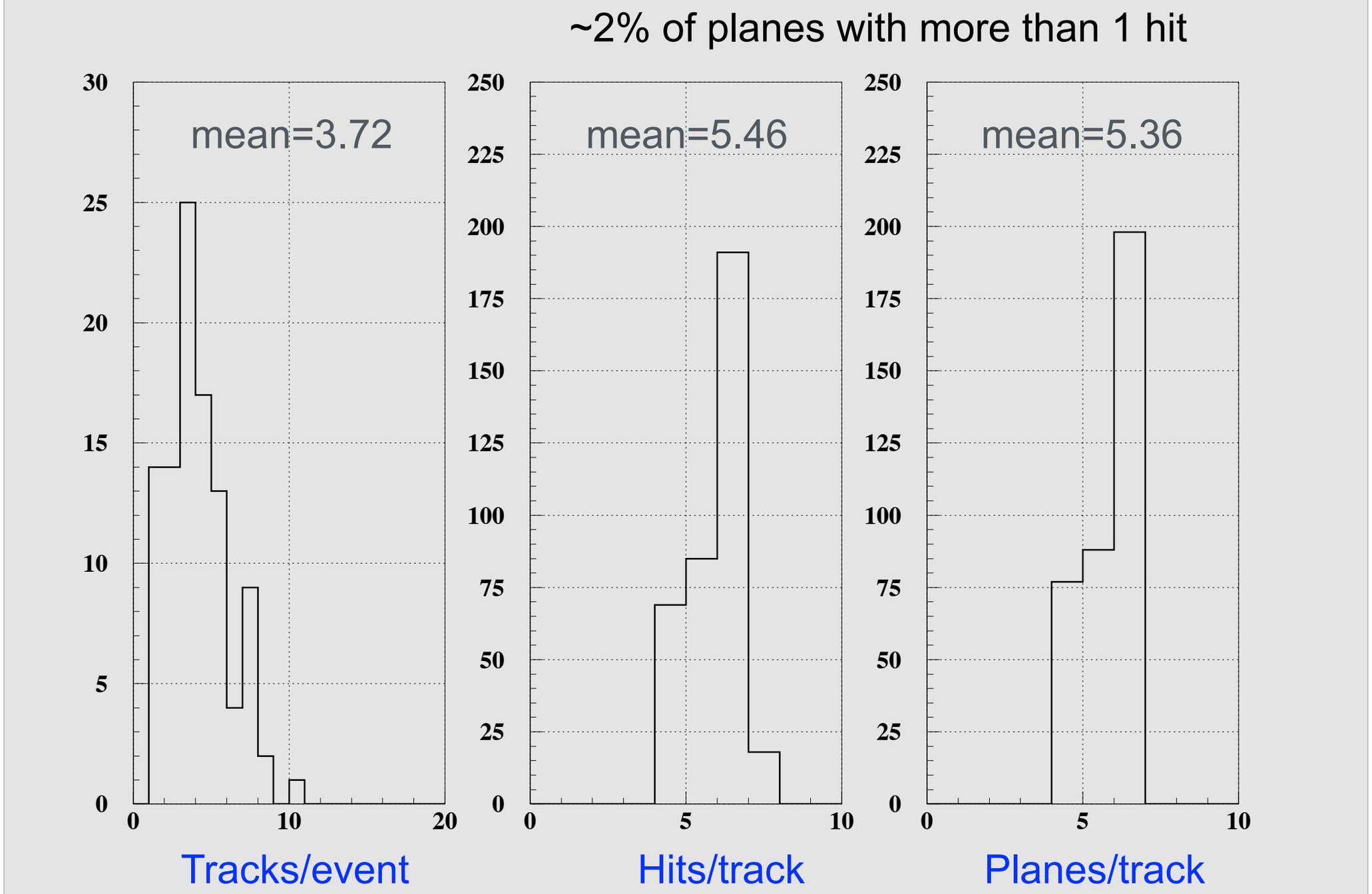
- EUDet telescope data
  - Got hundred events with and without magnetic field from Alexander Morton (Thanks again)
    - B off, 6 planes at: 0., 150., 300., 450., 600., 750.
    - B on, 6 planes at: 0., 30., 60., 260., 285., 310.
      - Coarsely aligned
- First tests
  - Ran FHT from ILCSOFT after coordinate transformation with only minor steering parameter adjustments on that telescope data
    - Minimum number of planes: 4

FHT with telescope, B off (I)

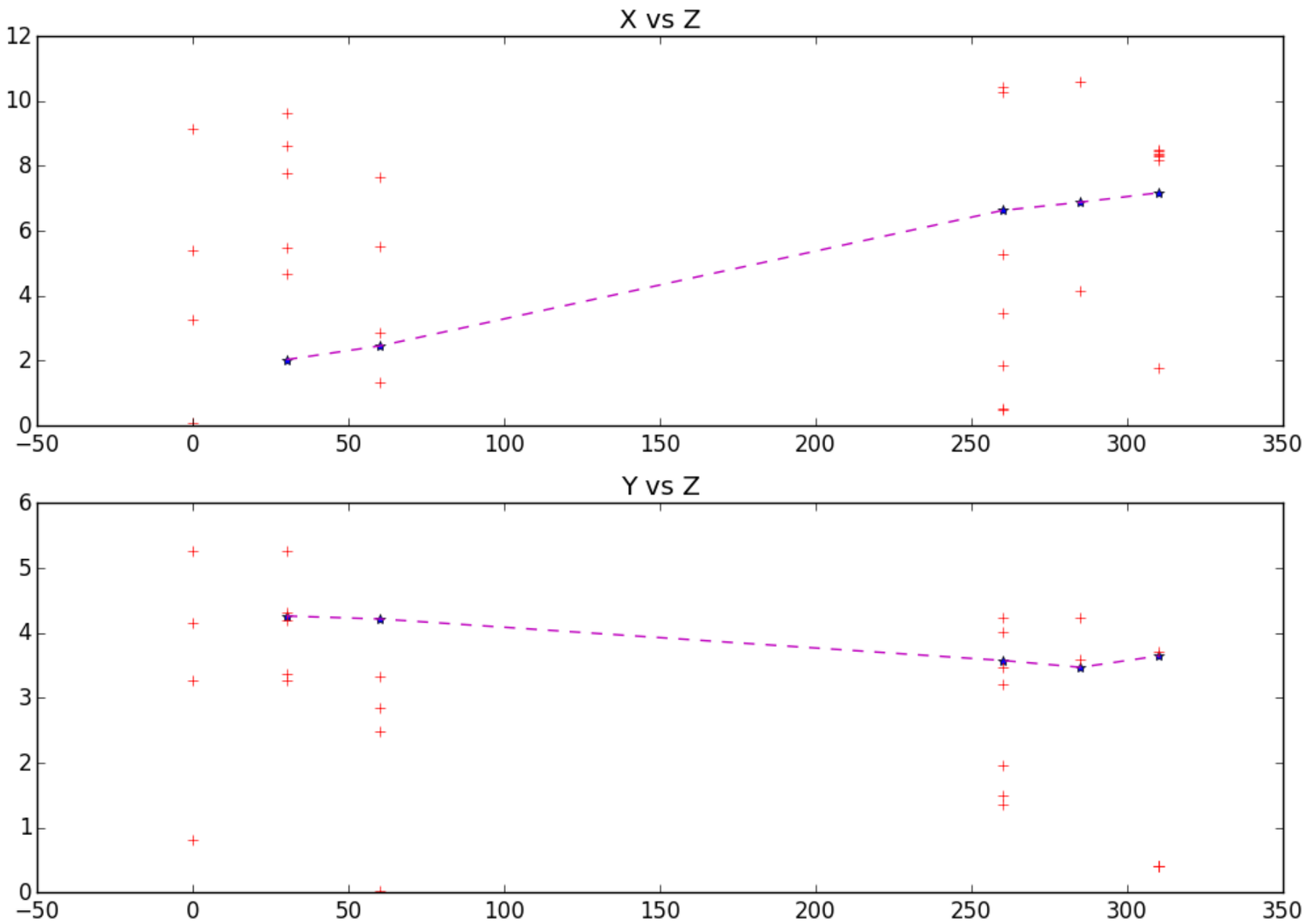


Event with 5 tracks and some noise hits

FHT with telescope, B off (II)



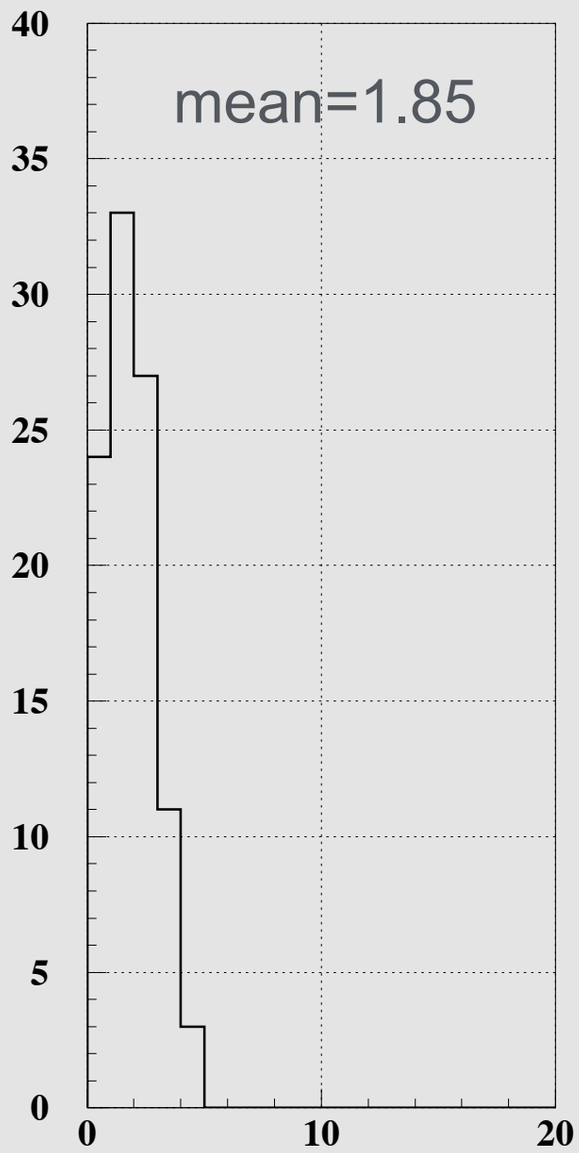
FHT with telescope, B on (I)



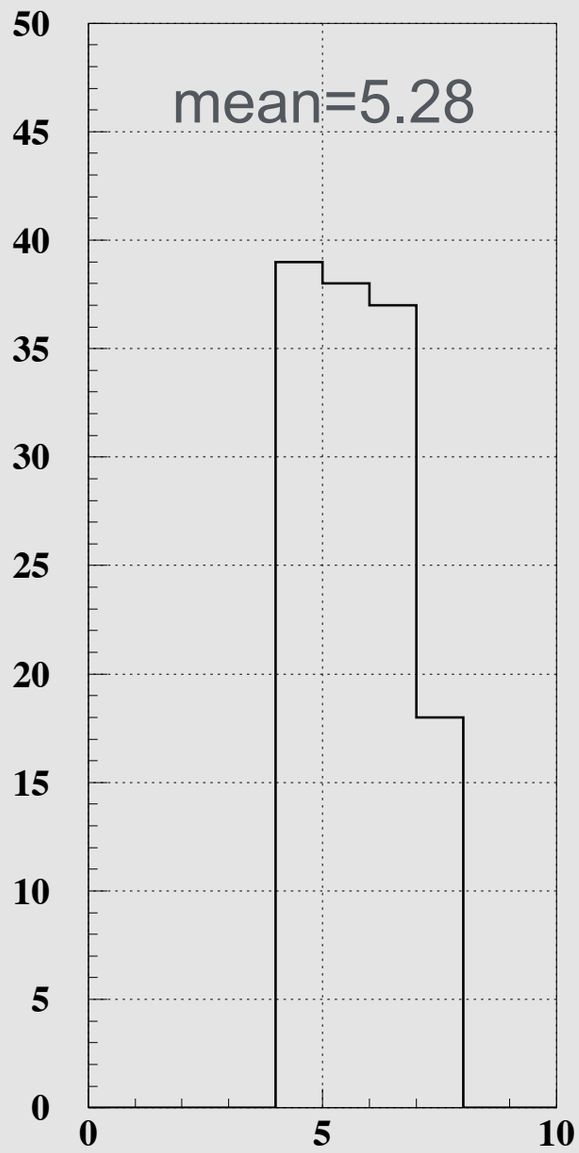
Event with 1 track and many noise hits

FHT with telescope, B on (II)

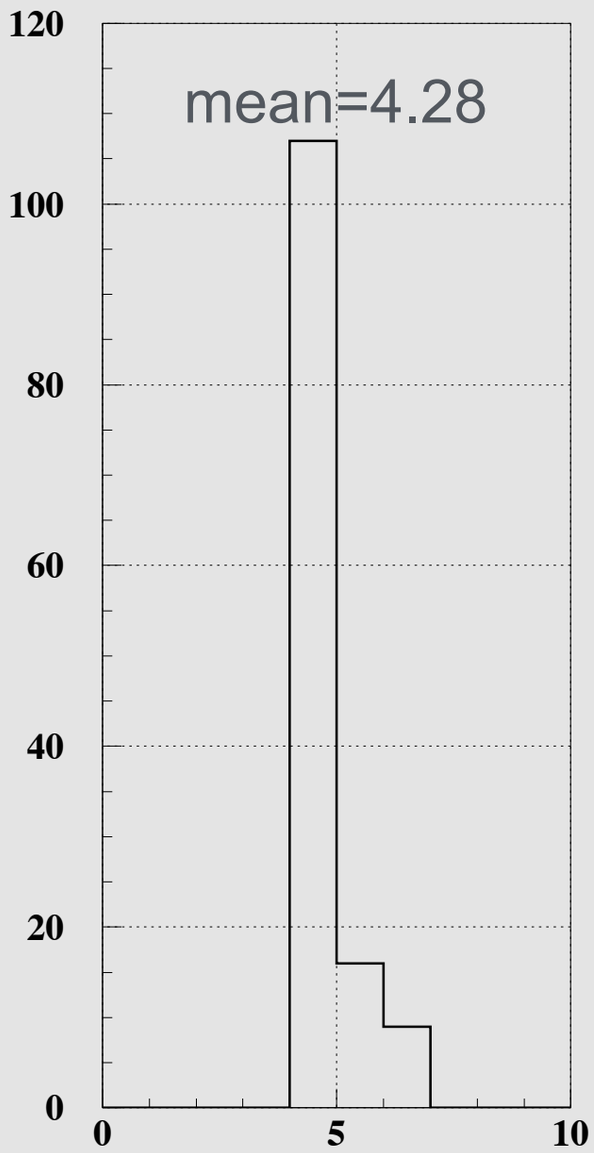
~20% of planes with more than 1 hit



Tracks/event



Hits/track



Planes/track

Summary

- Fast Hough Transformation from ILCSOFT used on telescope data
- First observations
  - Runs fine (out of the box) on field off data
  - Field on data suffers from high(er) noise
- Proper implementation needs some more work
  - Proper noise filtering, adjustment of steering parameters, ..
  - Any volunteer?