Pixel TPC simulation and reconstruction

Kees Ligtenberg

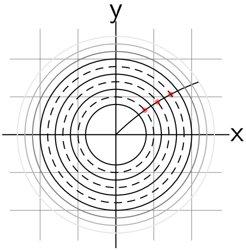
Nikhef lepton collider meeting

27 March 2017

Outline

1. Simulation
2. Track finding
3. Track fitting

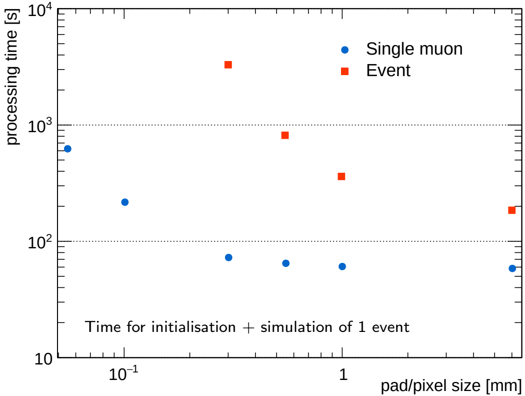
Simulation of pads within ilcsoft



Volumes are organised as tube shaped layers, there are no pad columns

- Detector is described by DD4HEP geometry
- Geant4 processes interactions of particle(s) from gun or event
- Single hit in TPC is deposited if energy above threshold (32eV) in a single pad. Position of pad centre crossing is recorded
- TPC hits that are smeared by expected resolution in TPCDigiProcessor are used as input for reconstruction

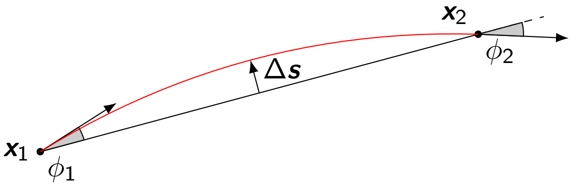
Simulating 55 × 55 μm^2 pixels as small pads costs too much processing time



- Processing time increases rapidly at smaller pixel sizes

Interpolate pixels in larger volumes

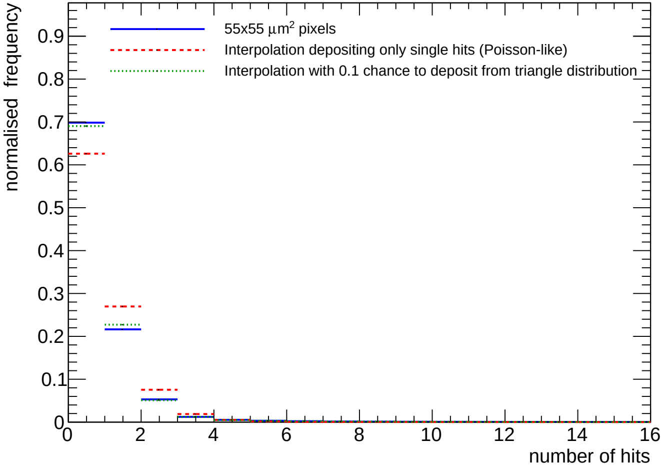
- The processing time can be sped up by approximating the $55 \times 55\ \mu\text{m}^2$ pixels over larger, e.g. $0.99 \times 0.99\ \text{mm}^2$ (18 pixels), volumes
- Register point and direction at entry and exit of volume
- Approximate the circular track with a parabola within the volume



Distribution of hits along the track

Ionization in gas follows roughly a Landau distribution

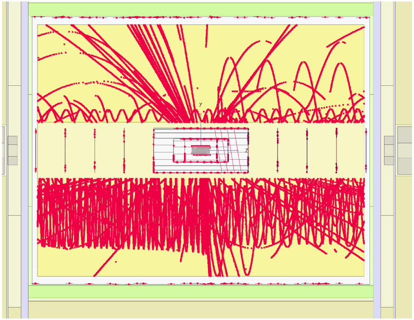
Distribute hits with a $P(N_{\text{hits}} = N) \simeq 0.1 \cdot \frac{2N}{N_{\text{total}}^2}$ chance to deposit multiple hits



Simulation of pad hits compared to pixel hits

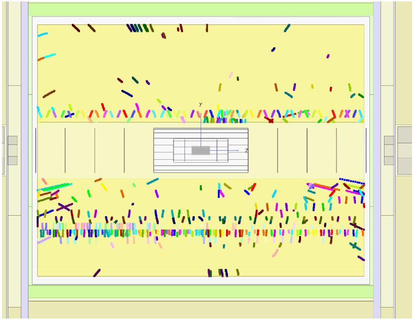
Pad hits	Pixel hits
6 mm $\times$ 1 mm	0.55μm $\times 0.55\mu$m
Exactly one hit per layer	Multiple or no hits per layer
22 electrons per hit	1 electron per hit
Only diffusion in ϕ and z	Diffusion in r, ϕ and z
$\sim$200 hits per track	$\sim$10 000 hits per track

Track finding for pads using Clupatra



Tracker hits

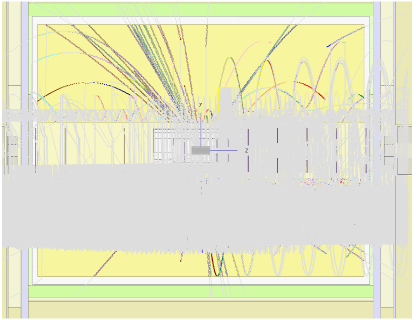
Track finding for pads using Clupatra



Seeds

1. Seed finding
   - Uses nearest neighbour clustering by distance in a pad row range of 15 rows

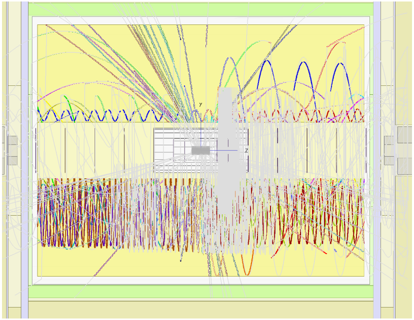
Track finding for pads using Clupatra



Seed fit and extended

1. Seed finding
   - Uses nearest neighbour clustering by distance in a pad row range of 15 rows
2. Fit track to seeds
   - use first, middle and last hit to initialise track parameters
3. Extend track inwards (and outwards)
   - Uses Kalman filter (Kaltest) in MarlinTrk, see track fitting

Track finding for pads using Clupatra

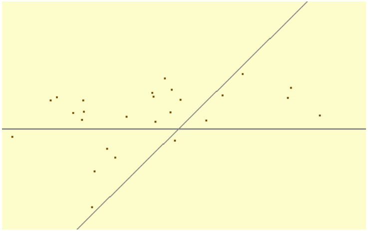


Track fit

1. Seed finding
   - Uses nearest neighbour clustering by distance in a pad row range of 15 rows
2. Fit track to seeds
   - use first, middle and last hit to initialise track parameters
3. Extend track inwards (and outwards)
   - Uses Kalman filter (Kaltest) in MarlinTrk, see track fitting
4. Combine curled segments

Issues when applying pad-track-finding to pixel-hits

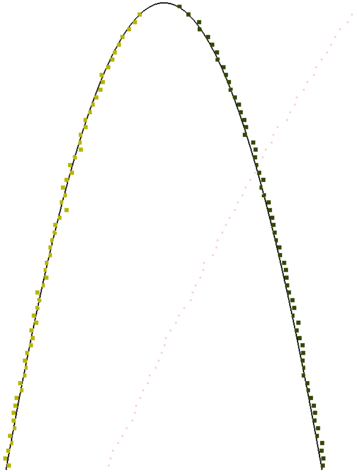
- Seed finding: computational complexity of nearest neighbour clustering scales as $\mathcal{O}(N^2)$
  Unsuitable for many thousands of pixel hits
- Track fit: initialise Kalman filter with first, middle and last hit
  3 hits do not fix the track tight enough, first hits can pull the track fit in the wrong direction



Track finding for pixel TPC

- Perform clustering by ϕ (Hough-transform like)
  - Fill histogram of hits by ϕ in pad row range of 750 rows
  - Maximum bin is cluster with track candidate if more than 200 hits
  - construct a straight line from the detector center to the average position
  - Cut hits on distance from this line (10mm in ϕ and 3mm rz)
  - initialise track fit with this line

Track fitting for pads



Track fit:

- Calculate intersection of helix with layer in coordinates (ϕ, z)
- Add closest hit to fit if $\chi^2 < \chi^2_{\text{threshold}} (=35)$

For curled (low momentum) tracks, cluster inward and outward parts separately and merge

Fit tracks by Extended Kalman filter

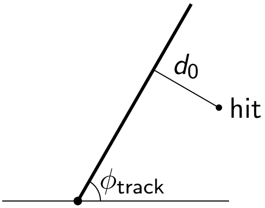
Fit track by an Extended Kalman Filter: a recursive fitting algorithm working in steps:

- Predict using state-propagator $\boldsymbol{a}_k^{k-1} = \boldsymbol{f}_k(\boldsymbol{a}_k)$
- Update with measurement $\boldsymbol{m}_k$ using state-to-measurement projector $\boldsymbol{h}_k(\boldsymbol{a}_k^{k-1})$
- ...

For pad track fitting

- $\boldsymbol{a}_k$ contains track parameters $(d_\rho, \phi_0, \kappa, d_z, \tan\lambda)$
- $\boldsymbol{m}_k$ is defined for pads as coordinates of a cylindrical surface (ϕ, z)
  Pixel hits are also smeared in the radial direction r direction
  Measurement vector has large error for tracks (almost) in z direction
  Difficult to extend towards multiple layers per element for pixels

Track fitting for pixel hits



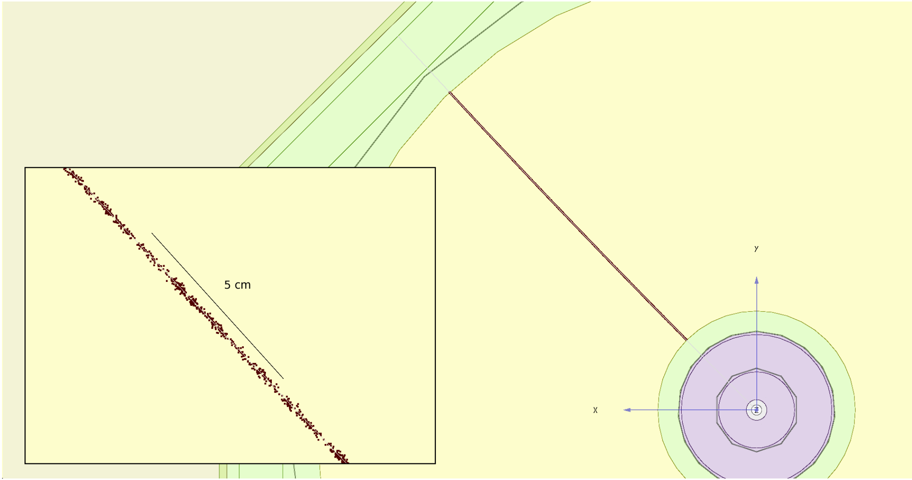
Define alternative measure with $\boldsymbol{m}_k$ as a function of $\boldsymbol{a}_k^{k-1}$

$$\boldsymbol{m}_k(\boldsymbol{a}_k^{k-1}) = \begin{pmatrix} d_0 \\ z \end{pmatrix} = \begin{pmatrix} \Delta x \sin(\phi_{\text{track}}) - \Delta y \cos(\phi_{\text{track}}) \\ z_{\text{hit}} + \tan\lambda(\Delta x \cos(\phi_{\text{track}}) + \Delta y \sin(\phi_{\text{track}}), \end{pmatrix}$$

the distance to the track d_0 better represents the measurement

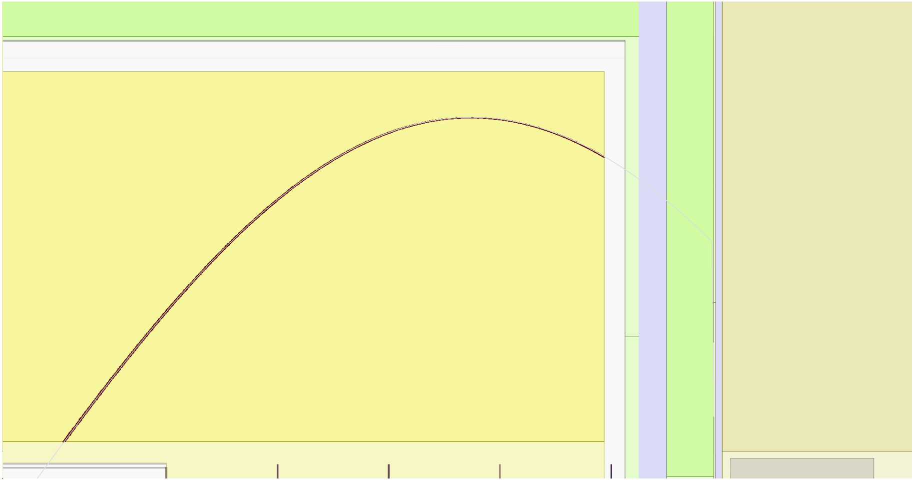
Fit of straight track

50 GeV muon



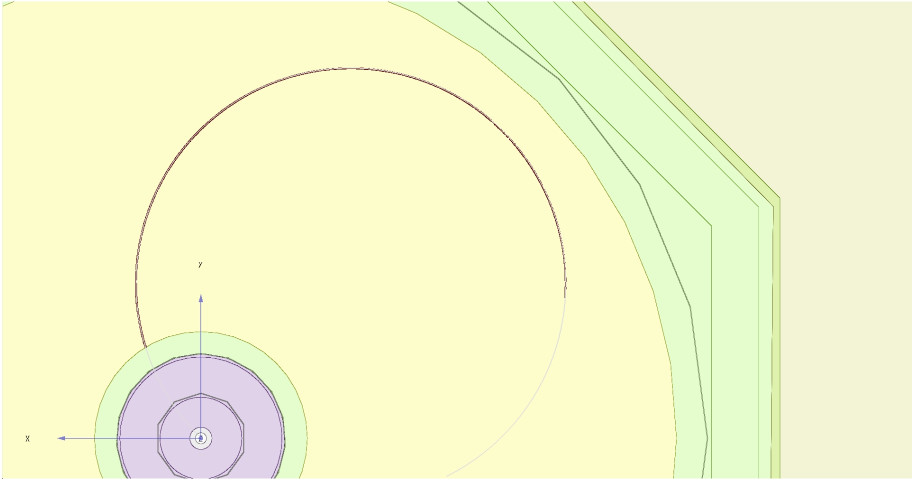
Fit of curled track

1 GeV muon without energy loss



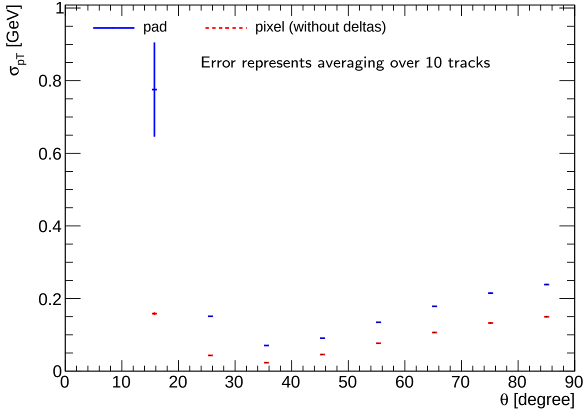
Fit of curled track

1 GeV muon without energy loss



Momentum resolution from track fit

50 GeV muon

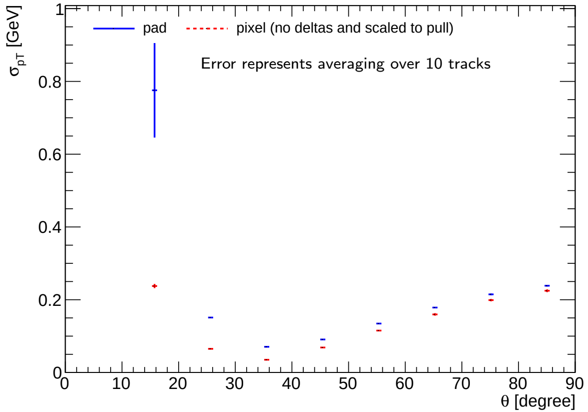


Pad pull is ~ 1.05

Pixel pull is $\sim$1.5 without delta electrons

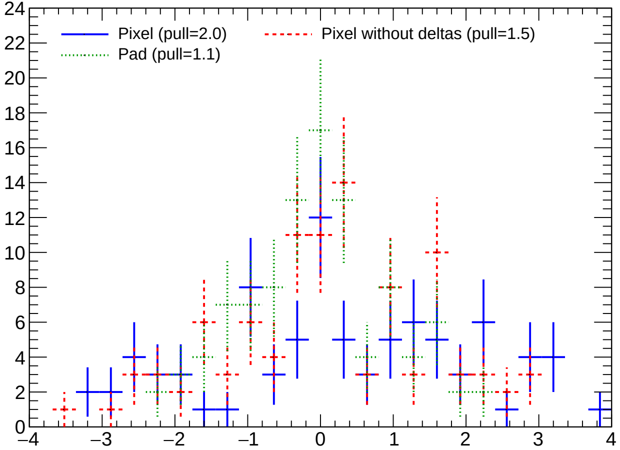
Momentum resolution from track fit

50 GeV muon



Pixel pull without delta electrons scaled to pull ∼1

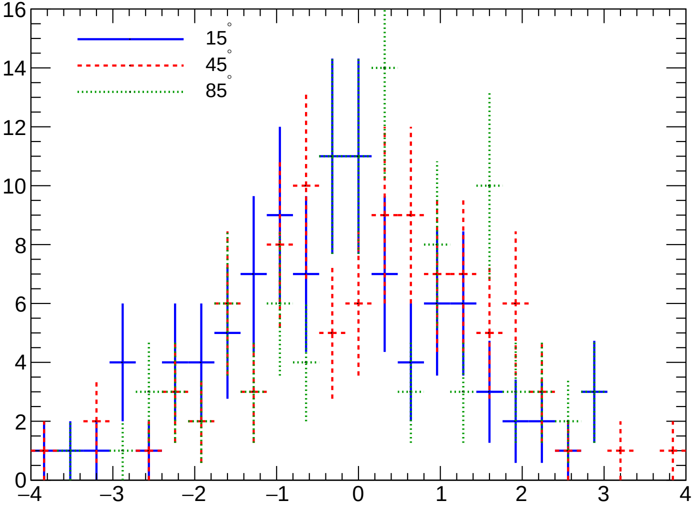
Pull of ω for a 50 GeV muon track at 85°



Conclusion

- Track finding was adapted for the larger number of pixel hits
- A less restrictive measurement vector was implemented for Kalman fitting
- Todo:
    - Fix pull of track fit

Pull of 50 GeV muon on pixel tpc without delta electrons



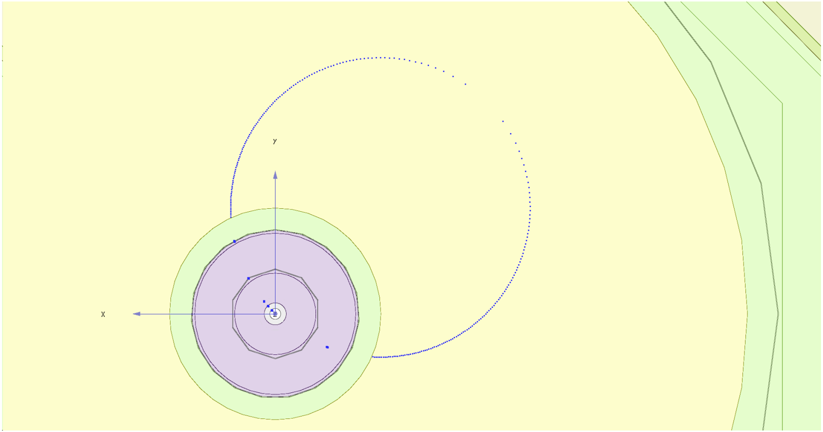
Extended Kalman filter

Recursive fitting algorithm to find state vector $\boldsymbol{a}_k$ and covariance $\boldsymbol{C}_k$ at site k from a series of measurements $\boldsymbol{m}_k$ by procedure:

- Predict
  - $\boldsymbol{a}_k^{k-1} = \boldsymbol{f}_{k-1}(\boldsymbol{a}_{k-1})$, where $\boldsymbol{f}_k(\boldsymbol{a}_k)$ is the state-propagator
  - $\boldsymbol{C}_k^{k-1} = \boldsymbol{F}_{k-1}\boldsymbol{C}_{k-1}\boldsymbol{F}_{k-1}^T + \boldsymbol{Q}_{k-1}$, where $\boldsymbol{F}_{k-1} = \frac{\partial \boldsymbol{f}_{k-1}}{\partial \boldsymbol{a}_{k-1}}$, and $\boldsymbol{Q}_k$ the covariance of the process noise
- Update
  - $\boldsymbol{a}_k = \boldsymbol{a}_k^{k-1} + \boldsymbol{K}_k \left(\boldsymbol{m}_k - \boldsymbol{h}_k(\boldsymbol{a}_k^{k-1})\right)$, where $\boldsymbol{h}_k(\boldsymbol{a}_k)$ the projector, $\boldsymbol{K}_k = \boldsymbol{C}_k^{k-1}\boldsymbol{H}_k^T(\boldsymbol{V}_k + \boldsymbol{H}_k\boldsymbol{C}_k^{k-1}\boldsymbol{H}_k^T)^{-1}$, $\boldsymbol{H}_k = \frac{\partial \boldsymbol{h}_k}{\partial \boldsymbol{a}_k^{k-1}}$, and $\boldsymbol{V}_k$ the coveriance of the measurement noise
  - $\boldsymbol{C}_k = \left((\boldsymbol{C}_k^{k-1})^{-1} + \boldsymbol{H}_k^T\boldsymbol{G}_k\boldsymbol{H}_k\right)^{-1}$, where $\boldsymbol{G}_k = (\boldsymbol{V}_k)^{-1}$
- (Smooth...)

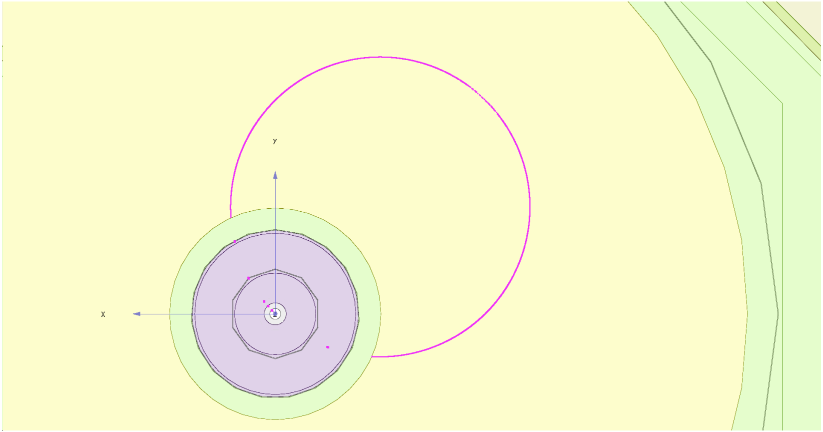
See: Keisuke Fujii, Extended Kalman Filter, The AFCA-SIM-J Group

Pad simulation of 700 MeV muon



Hits are only at layer crossings
This was solved for pixel hits by adding an interpolator

Pixel simulation of 700 MeV muon



Hits are only at layer crossings
This was solved for pixel hits by adding an interpolator

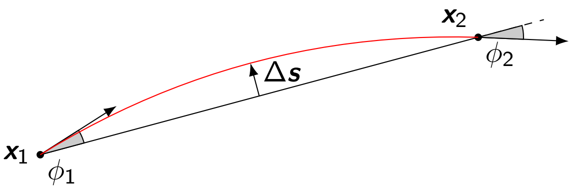
Parabolic interpolation

The position $\boldsymbol{x}(t)$ between the points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is parametrised as a function of $0 \geq t \geq 1$

$$\boldsymbol{x}(t) = \boldsymbol{x}_1 + t(\boldsymbol{x}_2 - \boldsymbol{x}_1) + 4t(1-t)\Delta\boldsymbol{s}, \tag{1}$$

where $\Delta\boldsymbol{s}$ is the deflection midway given by

$$|\Delta\boldsymbol{s}| = \frac{|\boldsymbol{x}_2 - \boldsymbol{x}_1|}{4}\sin(\Delta\phi_{12}/2). \tag{2}$$



Curled segements in schematic pad layout