TPC reconstruction methods for the ILC

Martin Killenberg

universitätbonn

3. TPC Analysis Jamboree
Orsay, 12. May 2009

EUDET

Overview

1. Track Finding
   - Topological Track Finder
   - Hough Transform
   - Kalman Filter

2. Background in the TPC

3. The Bunch Train Problem

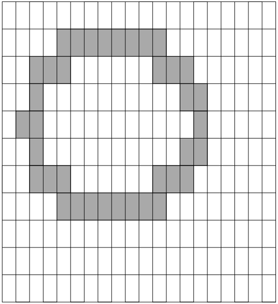
4. Track Fitting
   - Track Seeding
   - χ^2-Minimisation
   - The Likelihood Fitter
   - Kalman Filter

N.B. This talks will focus on MarlinTPC.
LEP-Tracking (wrapped Fortran code) used in Particle Flow analyses is not considered.

Topological Track (and Hit) Finding

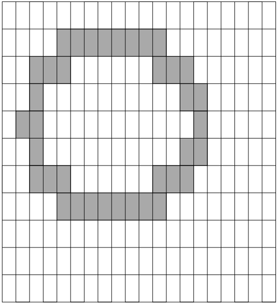
Search for contiguous areas on the pad plane

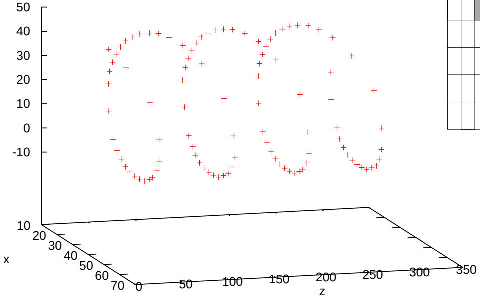
+ Independent of trajectory, not track hypothesis
+ Works in 3D
+ Fast and robust algorithm
− No seed parameters for track fit
− No separation of crossing tracks (yet)



Search for contiguous areas on the pad plane

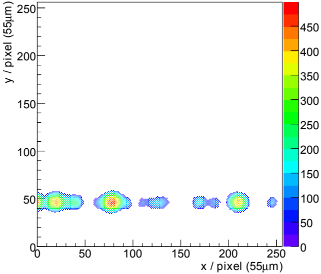
- + Independent of trajectory, not track hypothesis
- + Works in 3D
- + Fast and robust algorithm
- − No seed parameters for track fit
- − No separation of crossing tracks (yet)



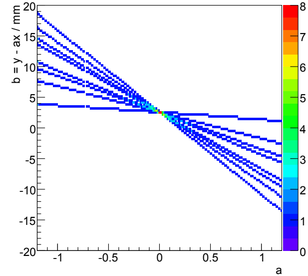


Hough Transform

Single track event



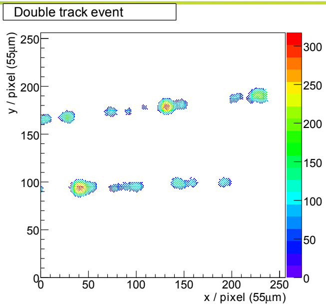
Hough space

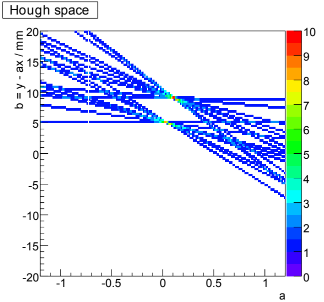


Example: Straight line (currently implemented)

- Track parametrisation: $y = a \cdot x + b$
- Hough space: $b = y - x \cdot a$
- Each hit (xy pair) corresponds to one line in parameter space.
- All parameter space lines intersect in one points, corresponding to the ab pair representing the track.

Hough Transform Algorithm

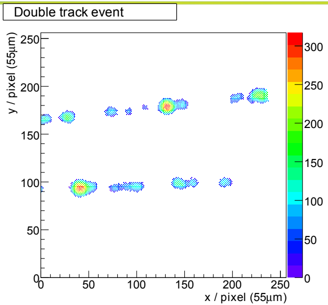


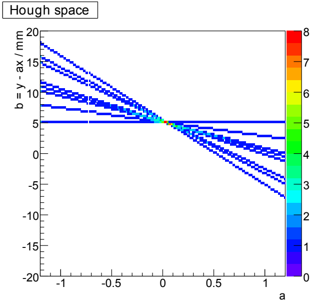


- Fill histogram (one b for each a per hit)
- Search for maximum
- Multiple tracks → multiple maxima
  - Start with absolute maximum
  - Remove all hits from found track
  - Search for next maximum
  - . . .

+ Runs on 2D (3D) hits
  ⇒ Independent from (pad) geometry
+ Multi track capable
+ Initial track parameters for track fit
− Bad double track separation
− Performance depends on histogram cell size
− Huge, multi-dimensional "histograms" for 3D helix parametrisations

Hough Transform Algorithm





- Fill histogram (one b for each a per hit)
- Search for maximum
- Multiple tracks → multiple maxima
  - Start with absolute maximum
  - Remove all hits from found track
  - Search for next maximum
  - . . .

\+ Runs on 2D (3D) hits
  ⇒ Independent from (pad) geometry

\+ Multi track capable

\+ Initial track parameters for track fit

– Bad double track separation

– Performance depends on histogram cell size

– Huge, multi-dimensional "histograms" for 3D helix parametrisations

universitätbonn

Kalman Filter

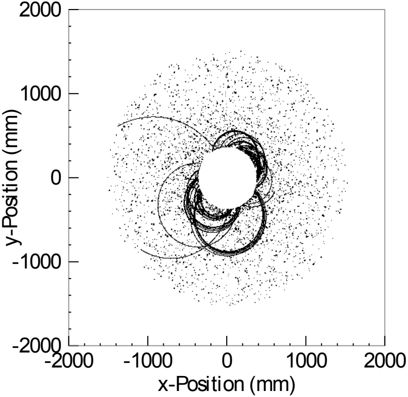
Improved version of a *track following* algorithm.

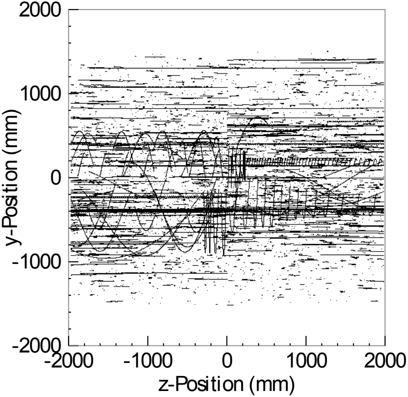
- Starts from a track seed
- Predicts hit in next layer (pad row)
- Includes found hit in track parameters to improve next prediction

\+ Combined track finding and fitting
\+ Can include detector material ($\mathrm{d}E/\mathrm{d}x$ and multiple scattering)
− Requires track seed + covariance matrix

Main Background: Micro Curlers

Background from 100 bunch crossings:





Study by Adrian Vogel

Properties of micro curlers:

- Low energetic particles spiral up.
- Some pads are *blind* for a long time (up to 60 μs).
- Only relatively few pads are affected.

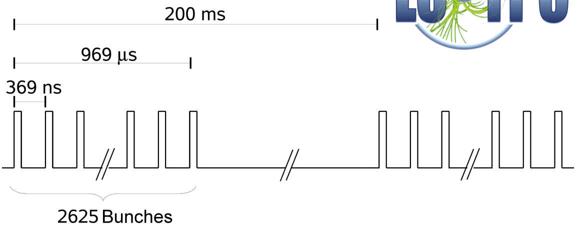
Issues to discuss:

- **Data reduction**
  Suppress micro curlers on the front end electronic:
  All channels that see a continuous signal over $\mathcal{O}$(μs) are ignored.

- **Pattern recognition and track fitting**
  No real problem for the pattern recognition, but both should be aware which channels were blind.

The Bunch Train Problem

- Bunch spacing: 369 ns
- Bunch train length: 1 ms
- TPC readout time: ≈ 60 µs



⇒ One large picture of the bunch train

⇒ Data from $\mathcal{O}(150)$ bunch crossings simultaneously in the TPC

⇒ z-position is ambiguous to $\pm n \cdot v_{\mathrm{drift}} \cdot t_{\mathrm{BX}}$

Front end electronics can

either store the hole bunch train, read out in 199 ms pause

or ship each pulse asynchronously

The data structure has to be adapted.

The Bunch Train Problem (2)

Problem: Marlin is event based.

Two solutions

1. Store complete bunch train in one event
   - \+ All information available in event, existing algorithms can be used
   - – Very large event, only 150 bunch crossings can contribute
2. Store data at end plate for each bunch crossing
   - \+ Only relevant data kept in memory
   - – Data from several "events" have to be combined, algorithms have to be adapted

MarlinTPC digitisation can produce both types of data.
Reconstruction can only handle first type.

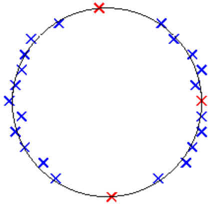
Not addressed yet: Matching TPC track with the correct bunch crossing

- Matching with inner tracker and / or calorimeter
- Influence of possible confusions on tracking efficiency / resolution

The track seeders provide starting values for numerical fits.

Simple track seeder

- Approximate helix projection as circle through 3 points



Linear regression (straight tracks only)

- Calculate track parameters analytically

Track seeder

- Transform helix parameters to linearise problem
- Calculate parameters using regression
- If errors are included this could be used as analytical track fitter.

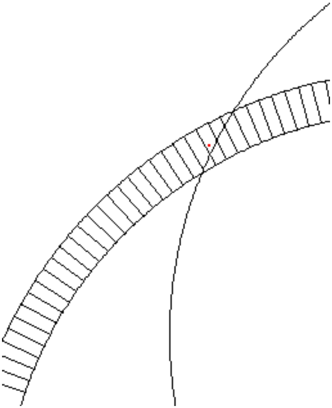
χ^2-Minimisation

Numerical minimisation of χ^2-Function:

$$\chi^2 = \sum_i \left(\frac{res_{i,xy}^2}{\sigma_{i,xy}^2} + \frac{res_{i,z}^2}{\sigma_{i,z}^2} \right)$$

Methods to calculate residuals:

- Along pad row
- Along coordinate axes
- Perpendicular to the track



+ "bread and butter" algorithm
+ χ^2 is simple quality test for the fit
− may require inversion of large matrices
− needs pad response corrections

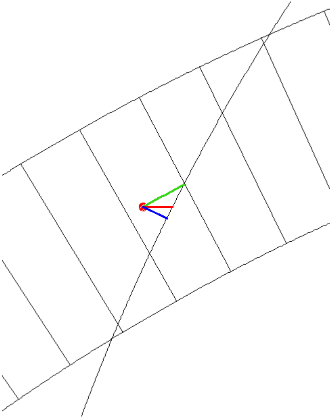
χ^2-Minimisation

Numerical minimisation of χ^2-Function:

$$\chi^2 = \sum_i \left(\frac{res^2_{i,xy}}{\sigma^2_{i,xy}} + \frac{res^2_{i,z}}{\sigma^2_{i,z}} \right)$$

Methods to calculate residuals:

- Along pad row
- Along coordinate axes
- Perpendicular to the track



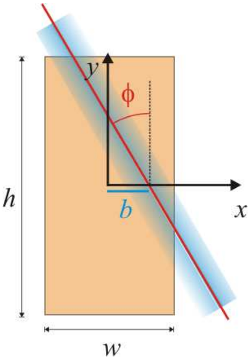
+ "bread and butter" algorithm
+ χ^2 is simple quality test for the fit
– may require inversion of large matrices
– needs pad response corrections

The Likelihood Fitter

- The pad response can only be calculated correctly if angle of track wrt. pad row is known.
- This cannot be done only on hit basis

⇒ Do it globally for the whole track

- Calculate likelihood of charge distribution per pad row, assuming Gaussian distribution
- Sum up log(likelihood) on all pad rows
- Maximise the global log(likelihood)



+ Includes gain fluctuations
+ Can include distortions due to field inhomogeneities
+ Reproduces original track very well
– Time consuming numerical calculations
– Numerical stability depends on correct calibration parameters

Kalman Filter

Filter calculates *state vector* of a dynamical system for a given "time"[1].

State vector in a detector layer:

- Position (two parameters)
- Direction (two parameters)
- Curvature (one parameter)

Steps:

- *Prediction:* Extrapolate the hit
- *Filtering:* Include the current measurement
- *Smoothing:* Propagate backwards to the previous measurements

+ No inversion of large matrices
+ Multiple scattering can be included
+ Easy to remove single hits (outliers)

− Requires track seed + covariance matrix

[1] *time* in this context means "layer" or "pad row"

Summary

- **Track finding algorithms**
  - Topological track finder
  - Hough Transform
  - Kalman Filter
- **Track fitting algorithms**
  - Analytical methods
  - Numerical χ^2 minimisation
  - Likelihood fit
  - Kalman filter
- **Special issues**
  - Background (micro curlers)
  - Event overlay
- Existing algorithms mainly for prototype data
  - No kink finding
  - No $\mathrm{d}E/\mathrm{d}x$ effects on track
  - Test performance in Particle Flow algorithms