Track Finding with Hough Transformation for TPC Data

Isa Heinze

LC Forum, Munich 2011

July 14, 2011

Outline

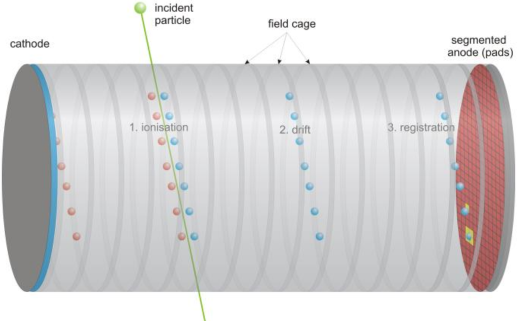
1. Time Projection Chambers
2. Track Reconstruction
3. Efficiency Studies
4. Conclusion and Outlook

Working Principle of Time Projection Chambers

A time projection chamber (TPC) consists of a volume filled with gas with an electrical field applied to it.

Working Principle

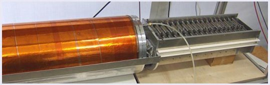
- A charged Particle traverses the volume and ionizes the gas inside the TPC.
- Due to the electric field the electrons drift towards the anode.
- Readout of the electrons at the anode.



Time Projection Chambers

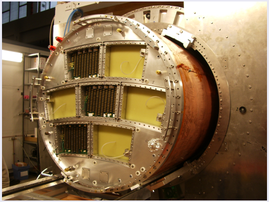
MediTPC

- Ø: 27 cm; length 80 cm
- Maximum drift length: 66 cm
- Pad plane: 60 x 80 mm^2



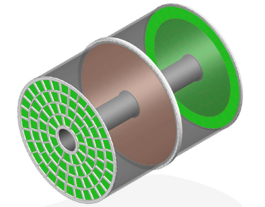
Large Prototype

- Ø: 77 cm; length: 61 cm
- room for up to seven modules

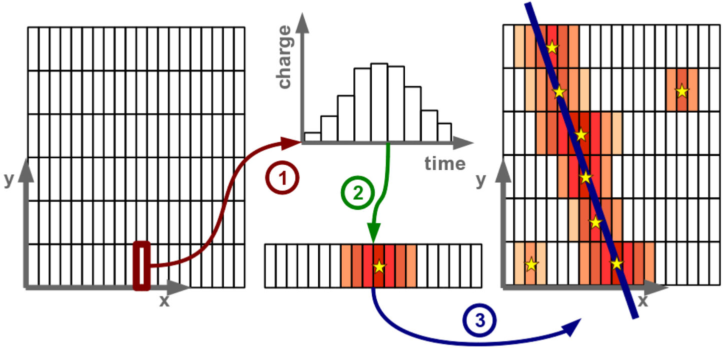


ILD TPC

- Ø: 3.8 m; length 4.3 m
- Tracking system in the future ILD detector



Track Reconstruction



1. For each pad a charge spectrum is recorded; search for pulses in the charge spectrum.
2. Pulses on adjacent pads in the same row are combined to hits (3D space points).
3. Hits are combined to tracks; tracks are fitted.

Pattern Recognition with Hough Transformation

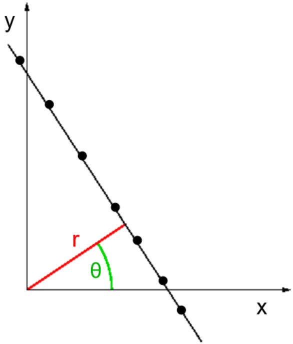
- Global method: all hits enter into the algorithm at the same time and in the same way.
- Works for any track shape which can be described by one set of parameters.
- here: either straight line (without magnetic field) or helix (with magnetic field).

Hough Transformation: Straight Lines

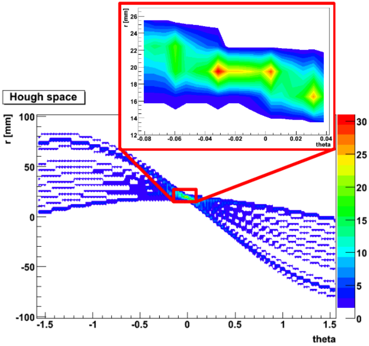
- Track Model: straight line
- Two parameters needed to describe straight line. Here: r, θ

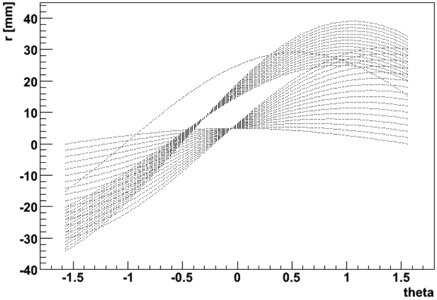
$$\begin{aligned} y(x) &= -\frac{\cos\theta}{\sin\theta}\cdot x + \frac{r}{\sin\theta} \\ \Rightarrow r(\theta) &= \cos\theta \cdot x + \sin\theta \cdot y \end{aligned}$$

- Hit positions are inserted in the function $r(\theta)$.
- If the hits are on a straight line the functions belonging to these hits intersect in one point in 2D Hough Space.
- The point of intersection delivers the parameters of the straight line.



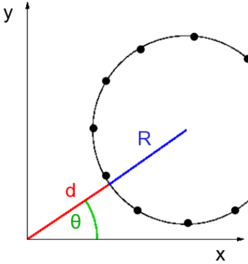
Hough Transformation: Straight Lines





Hough Transformation: Circles

- Track model: circle
- Three parameters needed to describe circle. Here: d, θ, R
- Same procedure as for straight lines
- Intersection of functions $R(d,\theta)$ in 3D Hough Space

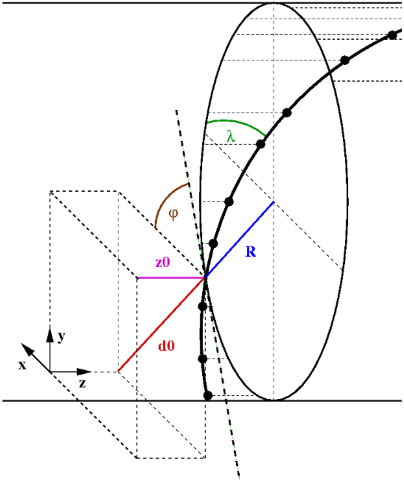


$$\begin{aligned} R^2 &= (x - x_M)^2 + (y - y_M)^2 \\ x_M &= (R + d)\cdot\cos\theta \\ y_M &= (R + d)\cdot\sin\theta \\ \Rightarrow R(d,\theta) &= \frac{x^2 + y^2 + d^2 - 2d\cdot(x\cdot\cos\theta + y\cdot\sin\theta)}{2\cdot(x\cdot\cos\theta + y\cdot\sin\theta - d)} \end{aligned}$$

$\Rightarrow$ 3D Hough Space

Hough Transformation: Helix

- Track model: helix
- Five parameters needed to describe a helix
- LCIO parameter: φ, $\Omega = 1/R$, d_0, z_0, $\tan\lambda$
- Functions would intersect in 5D Hough Space
- Search is done in two projections
  - xy projection (straight line or circle)
  - sz projection (straight line), where s is the arc length in the xy projection

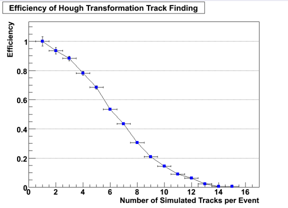


Simulation for track finding in multi track events

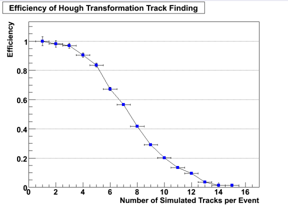
- Volume of MediTPC: 60 x 80 x 666 mm^3
- Straight lines
- Track parameters are created randomly in the volume.
- Tracks are simulated such that they are vertical ($\pm 30^\circ$).
- Y positions are the center of the pads.
- X and z positions are calculated accordingly with the track parameters.
- 12 hits per track (MediTPC like)

Track Finding Efficiencies

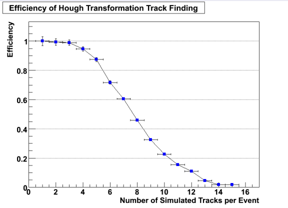
a: 0 wrong hits allowed



b: 1 wrong hits allowed



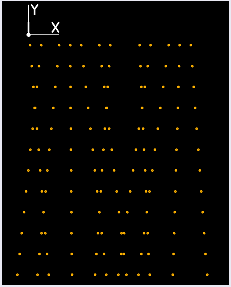
c: 2 wrong hits allowed



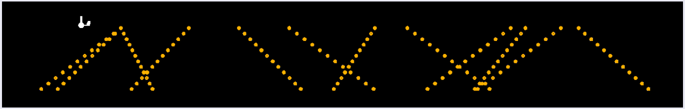
- 1000 simulated events with n tracks
- Volume: MediTPC
- Track was found correctly if all hits (**a**), all but one hit (**b**), all but two hits (**c**) were found correctly.

Example: Straight Line

Simulated tracks in xy projection

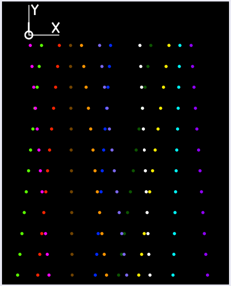


Simulated tracks in zy projection

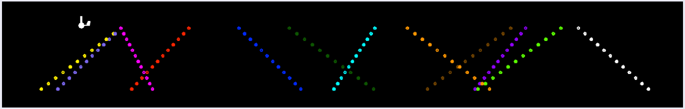


Example: Straight Line

Reconstructed tracks in xy projection



Reconstructed Tracks in zy projection

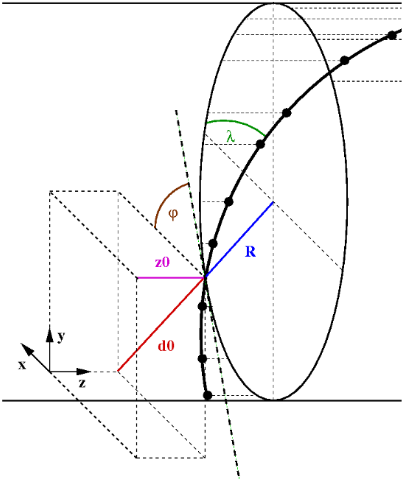


Track Finding Efficiencies

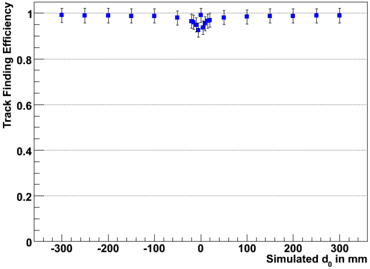
- Hits might not be assigned to the right track if
  - two tracks cross each other.
  - two tracks are very close to each other.
- With very few hits per track the points of intersection in the Hough space might not be defined well.
- Especially in events with many tracks fake tracks might be found.

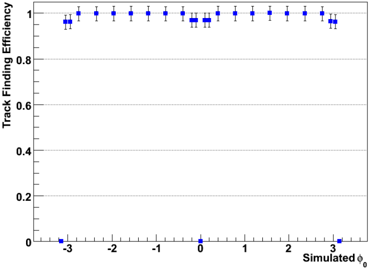
Simulation for Parameter Scans

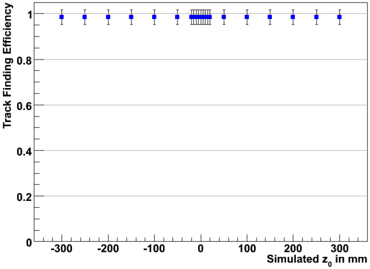
- 1000 events
- 1 track per event
- 200 hits per track (ILD TPC like)
- Y-Positions of hit in the center of pads rows.
- Straight lines ($\Omega = 0$)
- One track parameter fixed, others randomly chosen.
- A track is defined to be found correctly when all hits belonging to that track were found correctly.

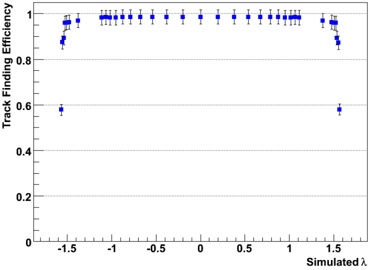


Track Finding Efficiencies









Track Finding Efficiencies

- For one track per event the track finding efficiencies is almost one for nearly all parameter regions.
- Tracks can not be found if they are nearly parallel to the pad rows ($\varphi \approx 0$, $\varphi \approx -\pi$ or $\varphi \approx \pi$) or the drift direction ($\lambda \approx -\frac{\pi}{2}$ or $\lambda \approx \frac{\pi}{2}$).
- The Hough transformation delivers two track candidates for each track. For d_0 close to 0 these two candidates are very close to eachother which leads to a lower track finding efficiency in this region.
- Excluding tracks with parameters in the critical regions leads to an even better track finding efficiency.

Conclusion and Outlook

Conclusion

- A Hough transformation for track finding was implemented.
  - It works for straight lines, helix segments and curler.
  - More than one track per event can be found.
  - Noise hits are rejected.
  - Hits do not need to be exctly on the track.
- Implemented as a library, which is currently used in MarlinTPC, but it can in principle be used in any other software framework.

Outlook

- Efficiency studies ongoing (Multi Track Events with 200 hits per track, helices, track separation, fake tracks).
- Improve the track finding algorithm.
  - Speed needs to be increased, ideas how to improve this exist.
  - Optimize algorithm to find tracks with higher efficiency.