Track fitting in HEP experiments

Salvador Martí-García
IFIC-Valencia
COMCHA School
Barcelona October 2019

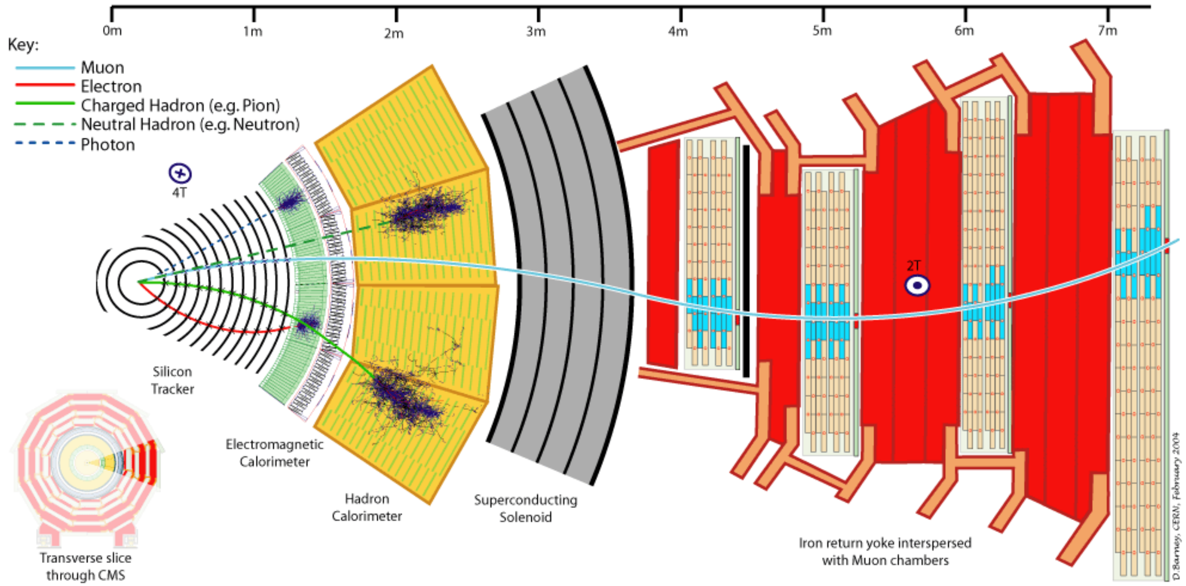
[Usain Bolt (JAM) in Berlin 2009]

Outline

- Track fitting
  - Basic ideas & concepts
  - Basic formulae
  - Pattern recognition
  - Track fitting with Kalman filter

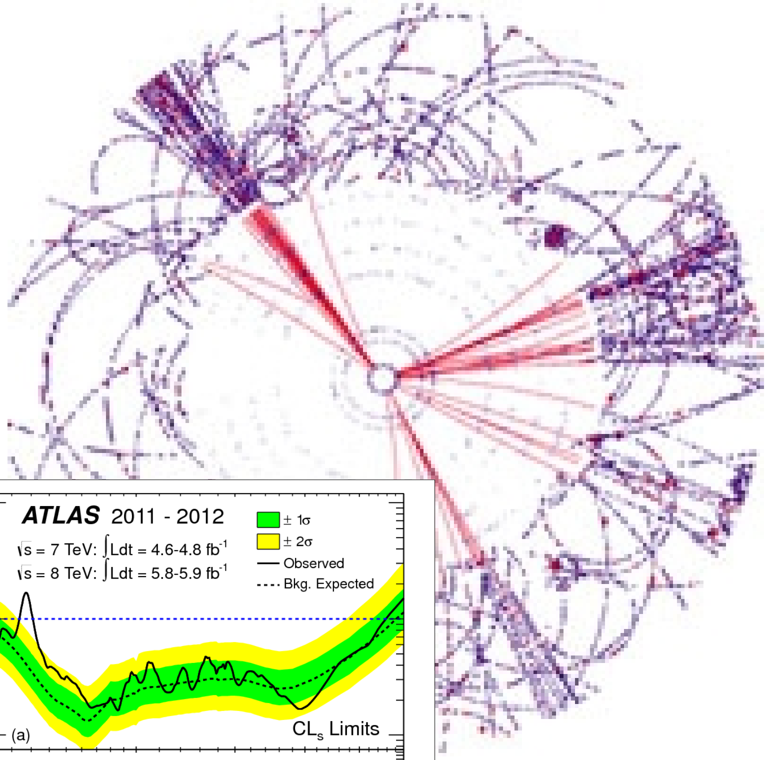
Data analysis flow in HEP experiments

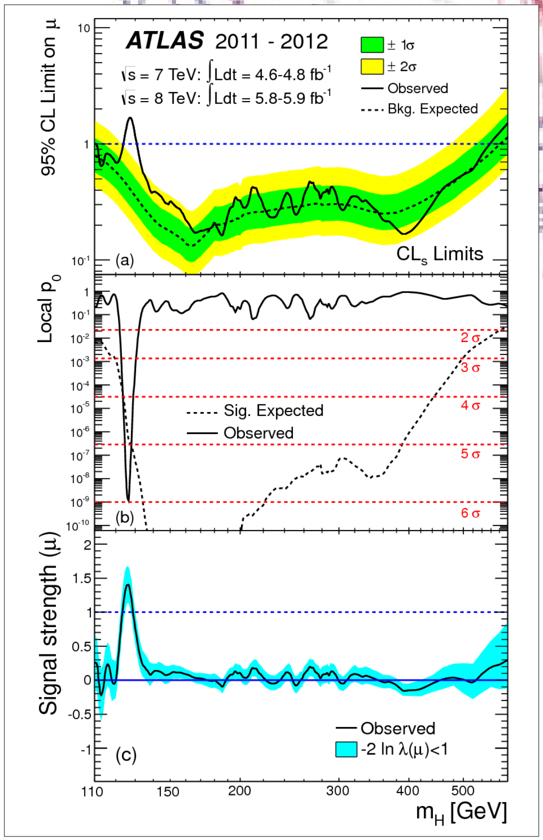
- Goal is to record the data registered by sensors when beams collide



Data analysis flow in HEP experiments

- Sensors react to the passage of particles and produce signals
  - Usually as electric pulses
  - Digitization: convert those pulse into digits
- Trigger
  - Whenever an interesting event happens
    - Whatever “interesting” means
- Record the data
  - In digital format
  - In disk or tape
- Event reconstruction
  - Tracker hits → tracks
  - Calorimetry → energy deposition
  - Bear in mind the calibration, geometry, etc.
- Event analysis & selection
  - According to the reconstructed objects
- Physics results
  - Eureka !

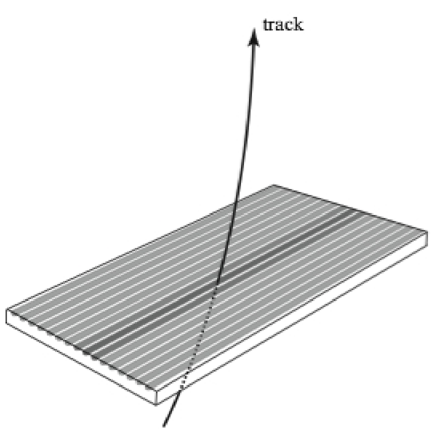


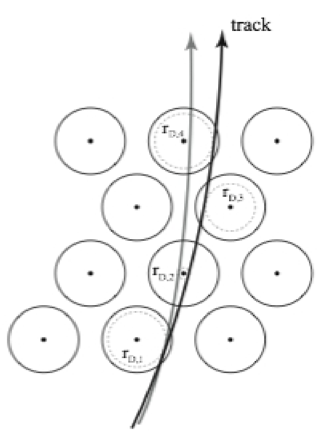


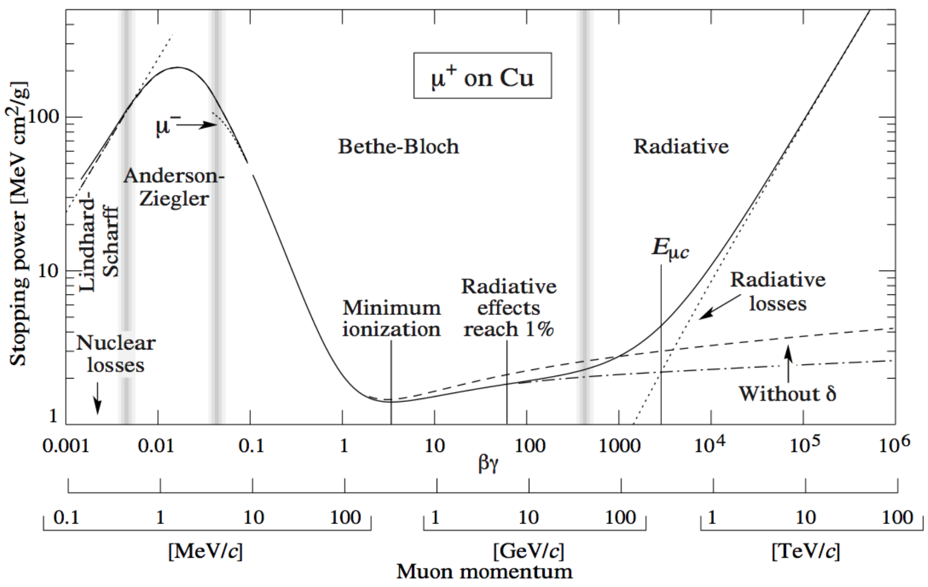
Introduction: tracking what for ?

- Tracking allows to determine the properties of those charged particles present in an experiment
  - Where is the particle ?
  - Where does it go ?
  - Which is its velocity ?
- Tracking is possible because charged particles interact with detector material
  - Energy loss by ionization: radiation detection
    - Bethe-Bloch formula

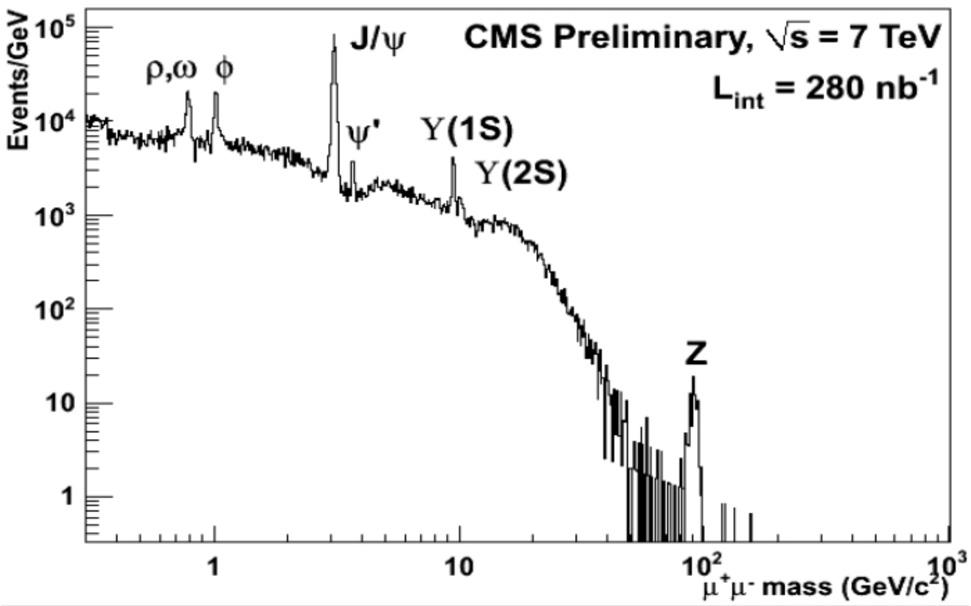
$$\left\langle \frac{dE}{dx} \right\rangle = -K z^2 \frac{Z}{A} \frac{1}{\beta^2} \left[\frac{1}{2} \ln \frac{2 m_e c^2 \beta^2 \gamma^2 T_{max}}{I^2} - 2\beta^2 - \frac{\delta(\beta\gamma)}{2} \right]$$

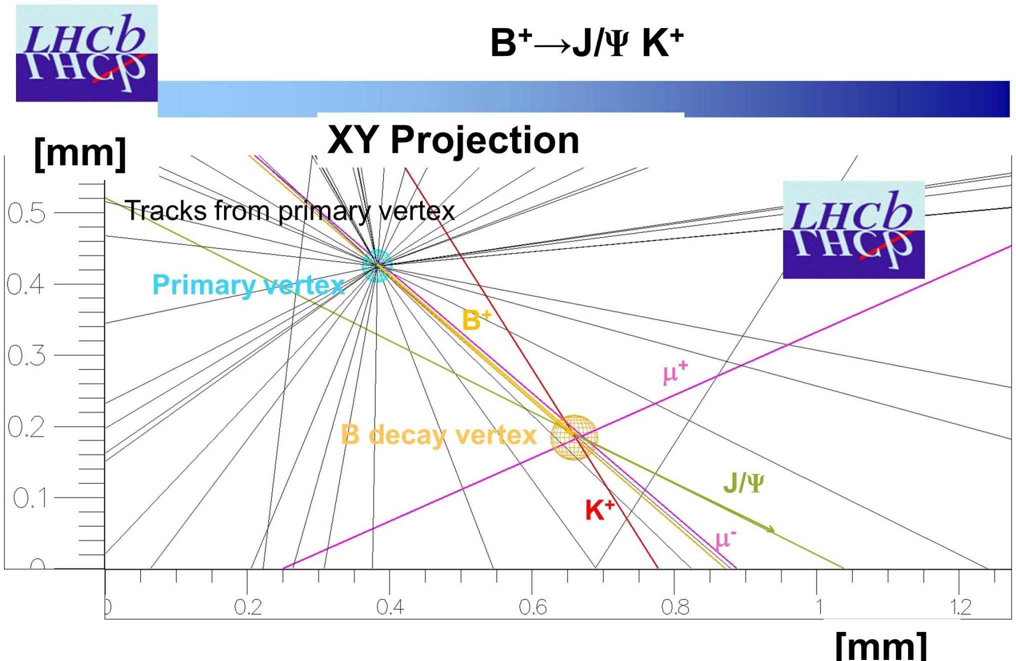






- A good performance of the Track Fitting is a key ingredient of the success of the physics program of the HEP experiments
  - An accurate determination of the charged particles properties is necessary
    - Invariant masses have to be determined with precision and well estimated errors
    - Secondary vertices must be fully reconstructed: evaluate short lifetimes
    - Kink reconstruction: on flight decays

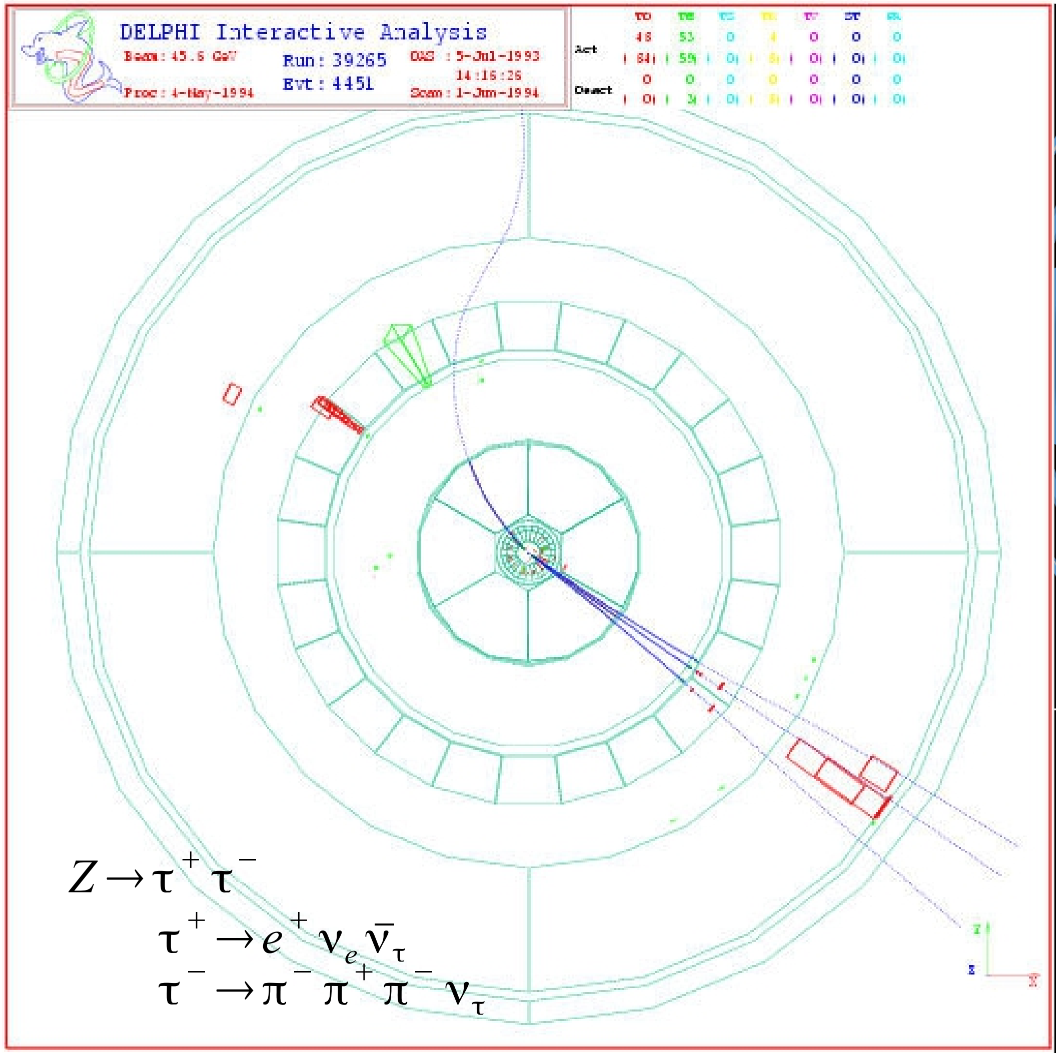


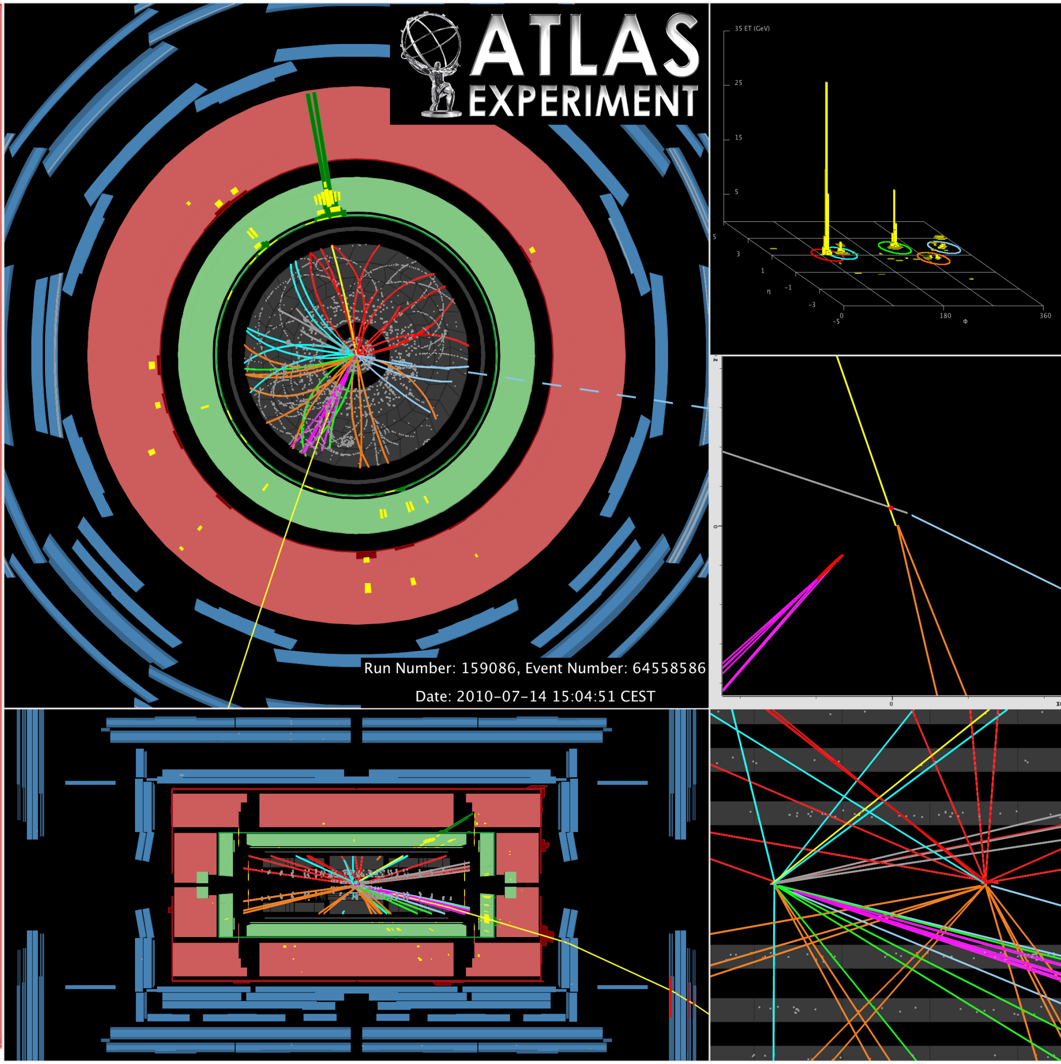


Tracks from PV are forced to come from PV

Introduction: tracking what for ?

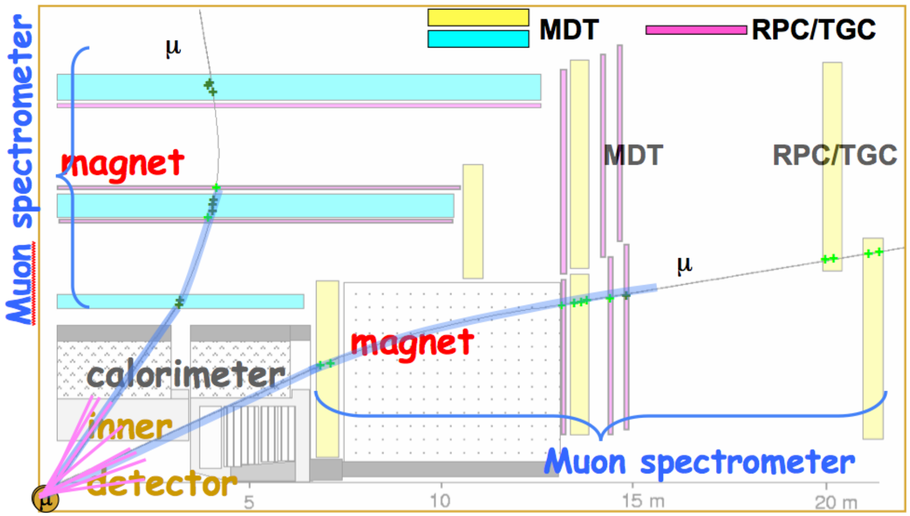
- Tracking allows to determine the properties of those charged particles present in an experiment





Introduction: tracking what for ?

- Challenges for the tracking systems of the LHC detectors
  - Momenta of particles in the final state ranging from MeV to TeV
  - High multiplicity of charged particles (up to 1000 for $\mathscr{L} \sim 10^{34}cm^{-1}s^{-1}$)
    - Even higher for heavy ion collisions
  - Large background from secondary activities of the particles
  - Multiple Coulomb Scattering in detector frames, supports, cables, pipes...
  - Complex modular tracking systems combining different detection technologies, different resolutions
  - Resolutions that vary as a function of the momentum (p), polar angle (θ) or pseudorapidity (η)
  - Very high event rates leading to large amount of data
    - with demanding requirements of CPU and storage → Tracking CPU budget

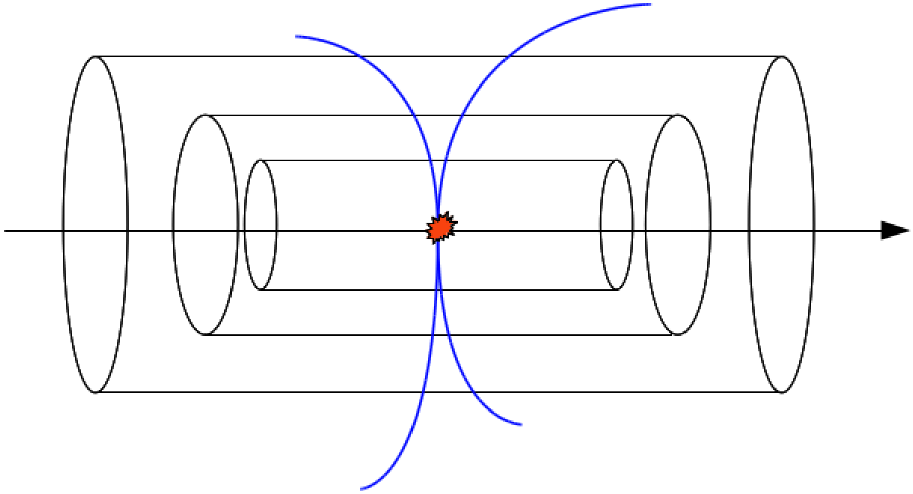


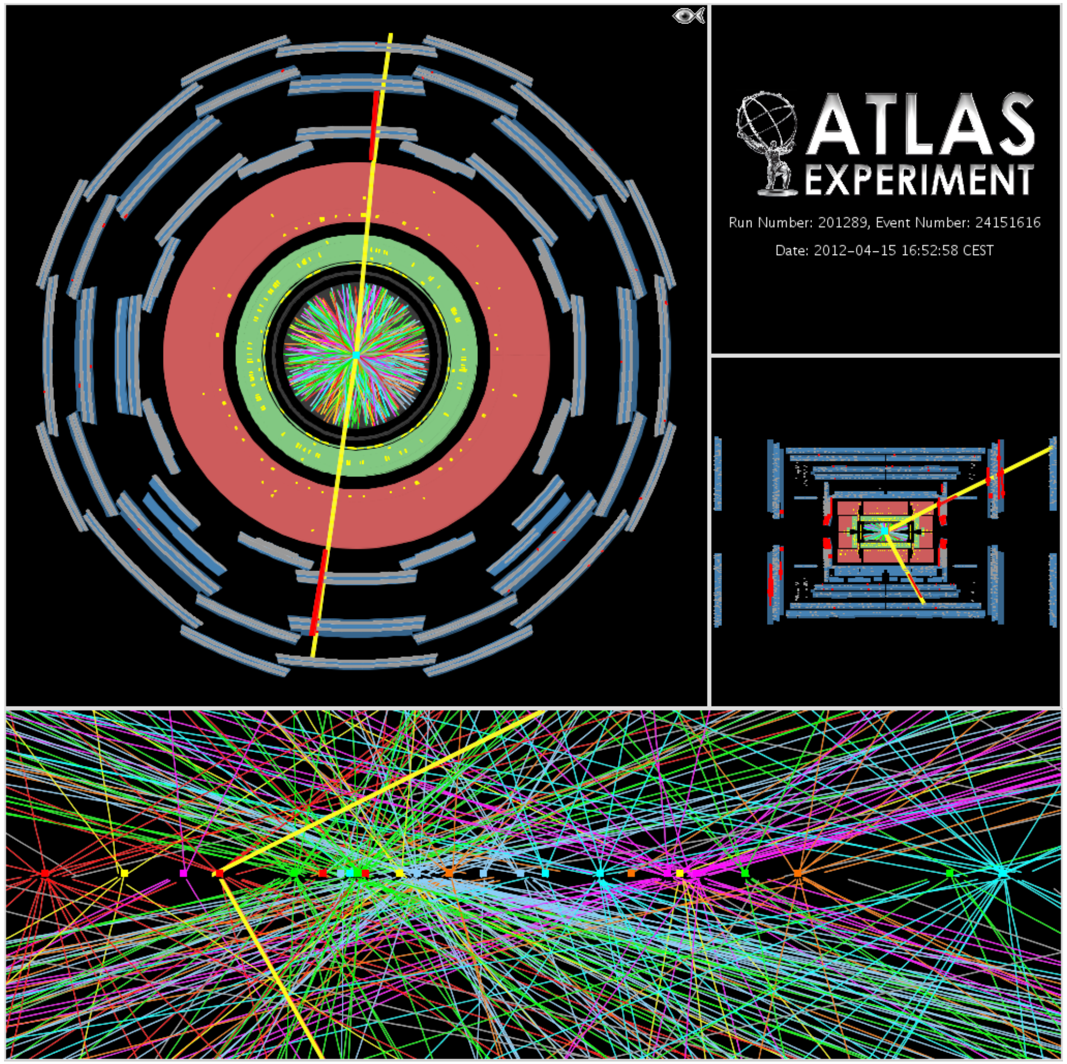
Introduction: tracking what for ?

- Finding where the particle was originated tell us much about the physics: primary vertex, secondary vertex or material interactions

Vertex fitting capabilities depend on tracking performance (specially in impact parameter and space point resolution)

Primary vertex

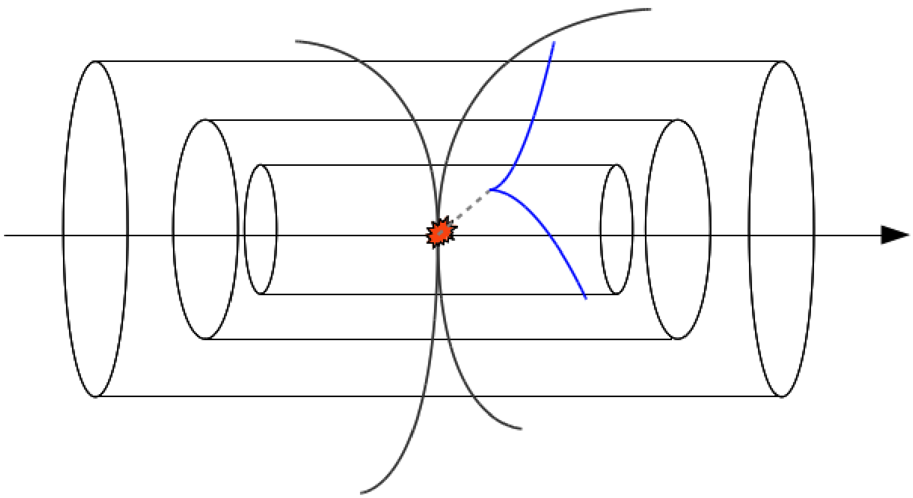


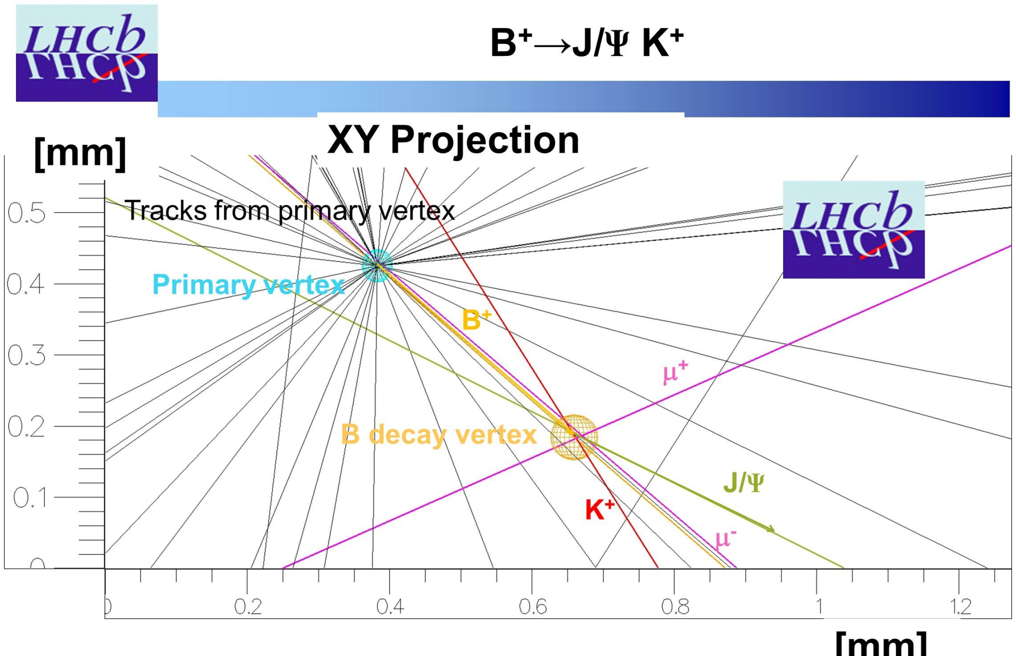


- Finding where the particle was originated tell us much about the physics: primary vertex, secondary vertex or material interactions

Vertex fitting capabilities depend on tracking performance (specially in impact parameter and space point resolution)

Secondary vertex: particle decay





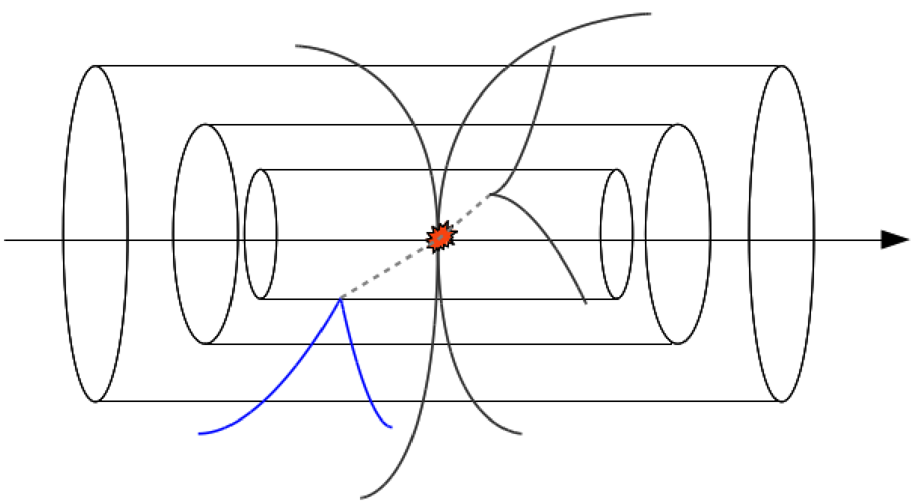
Tracks from PV are forced to come from PV

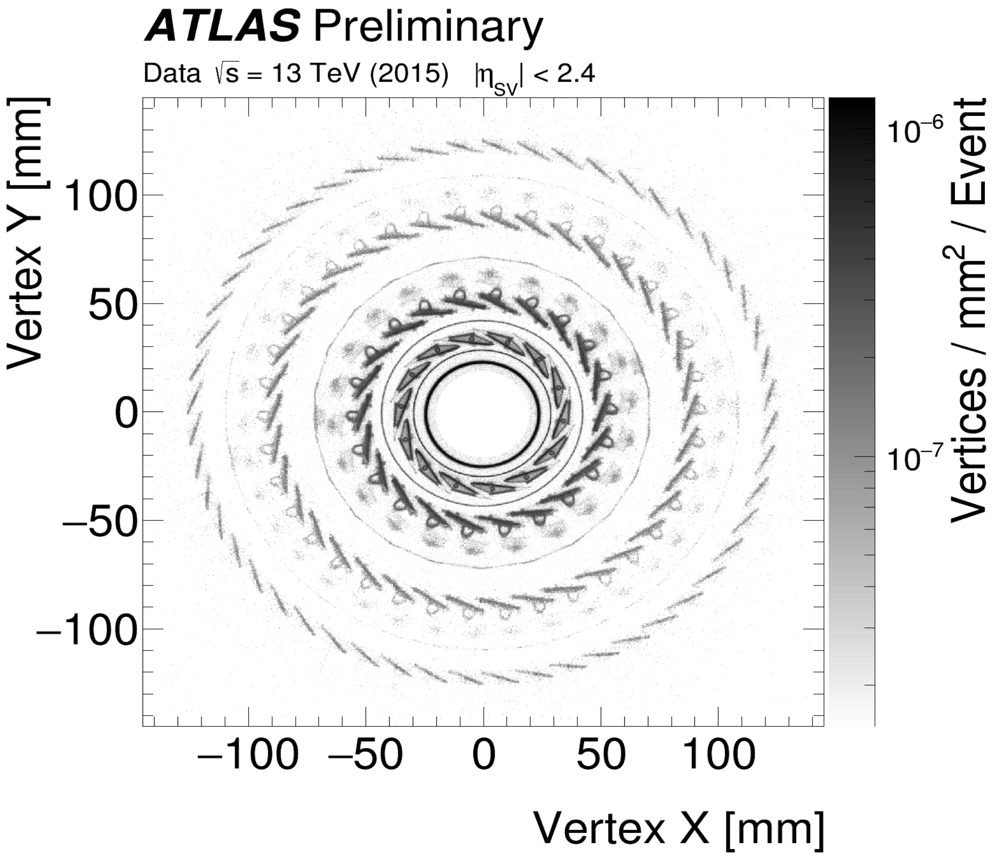
Introduction: tracking what for ?

- Finding where the particle was originated tell us much about the physics: primary vertex, secondary vertex or material interactions

Vertex fitting capabilities depend on tracking performance (specially in impact parameter and space point resolution)

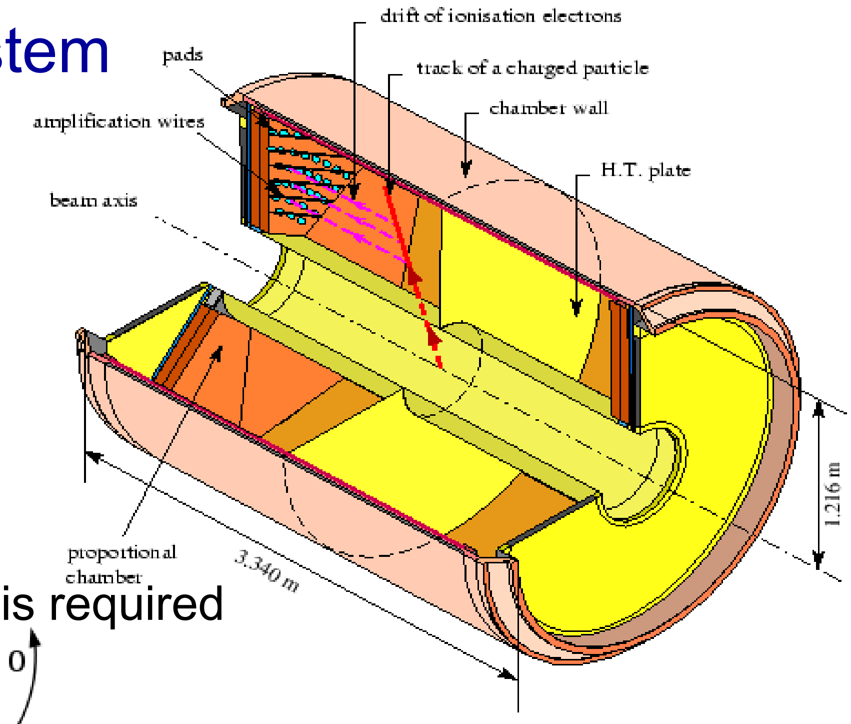
Secondary vertex: material interaction



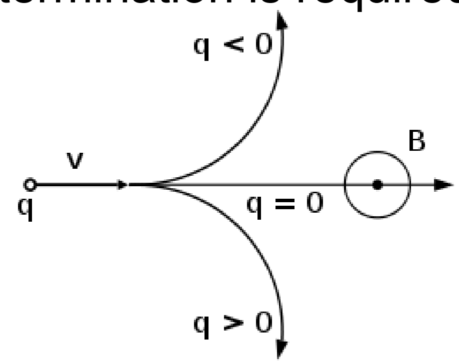


Basic ingredients

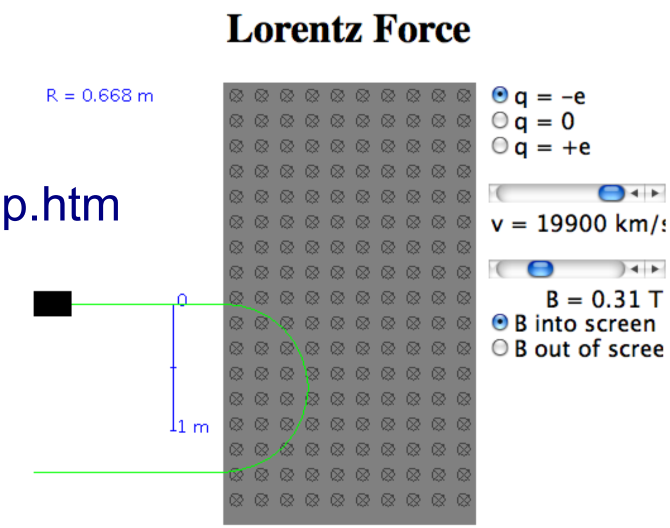
- Basic ingredients of the tracking system
  - Charged particles (+ve or -ve)
    - |q| = 1, 2 (e, μ, π, k, p, α, d,...)
  - Ionization detector
    - Continuous (e.g.: gas detectors)
    - Discrete (e.g.: silicon planar detectors)
  - Magnetic field (no strictly necessary)
    - Necessary if momentum determination is required
    - Lorentz force



$$\vec{F} = q(\vec{E} + \vec{v} \times \vec{B})$$



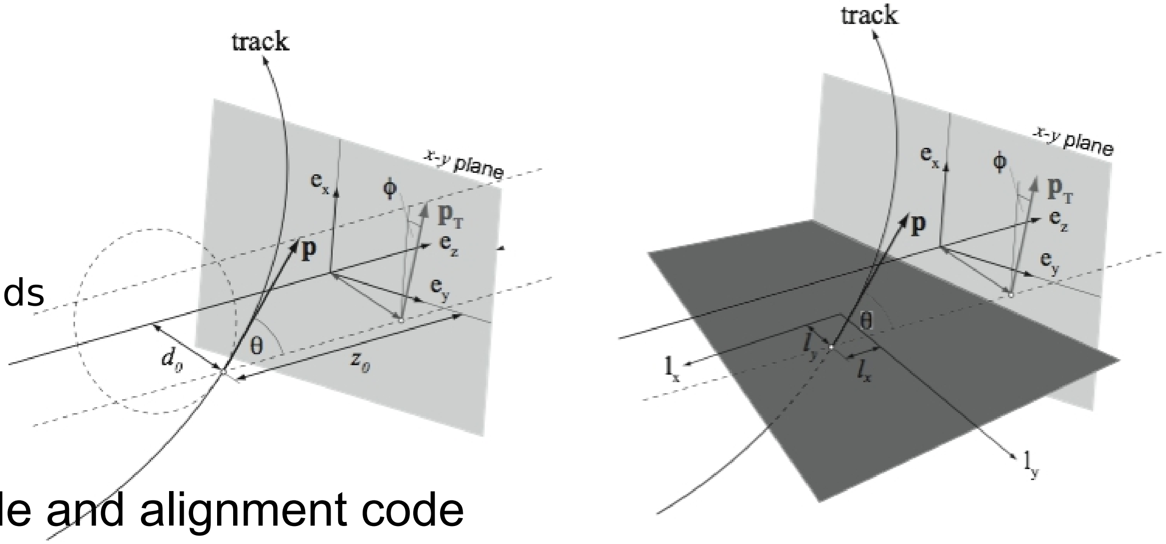
- Example: Nice Java applet
  - http://www.lon-capa.org/~mmp/kap21/cd533capp.htm
- Usually E=0 inside detectors or quite small
  - Negligible effects on tracks
  - E > 0 necessary for ionization charge collection
- The bending of the trajectory is due to B field



Track parameters

- A trajectory can be parametrized with just 5 parameters at a surface
  - x, y, φ, θ, v
- The track extrapolation to detector surfaces usually requires a different parametrization
  - Optimization
    - Track parameters given in the local reference frame of the surface
  - Error matrix propagation !
- The track is characterized by its 5 parameters as given at the "perigee surface" & using the global reference coordinate system
  - $d_0, z_0, \varphi_0, \theta_0, q/p$
  - $d_0, z_0, \varphi_0, \cot\theta_0, q \cdot p_T$
  - $d_0, z_0, \varphi_0, \eta, q/p$

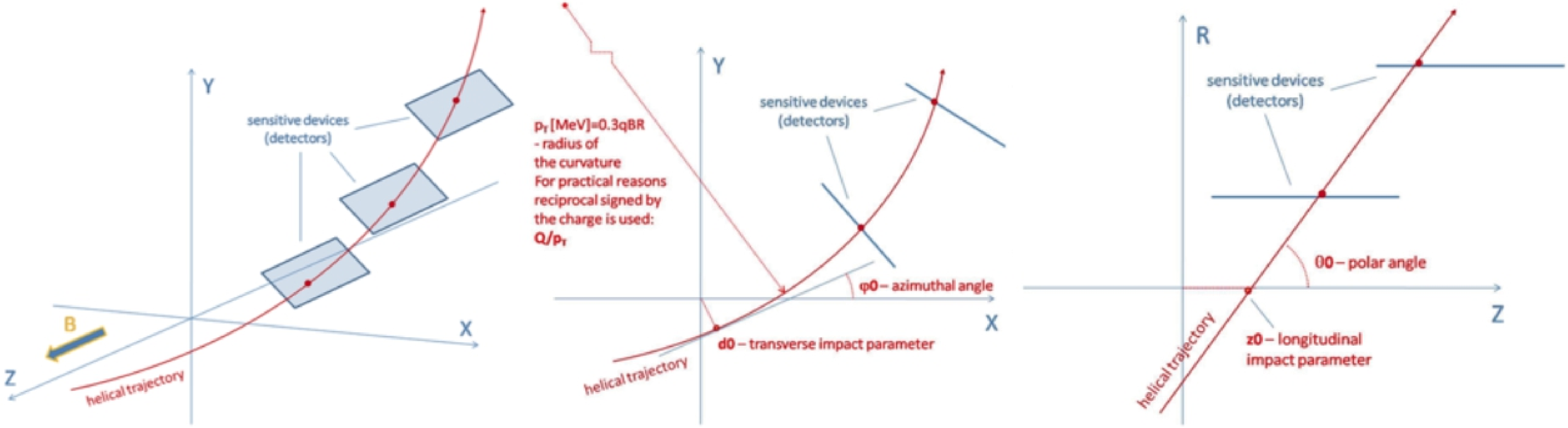
The choice of parametrization depends on the detector layout



- Track extrapolation
    - Heavily used in tracking code and alignment code

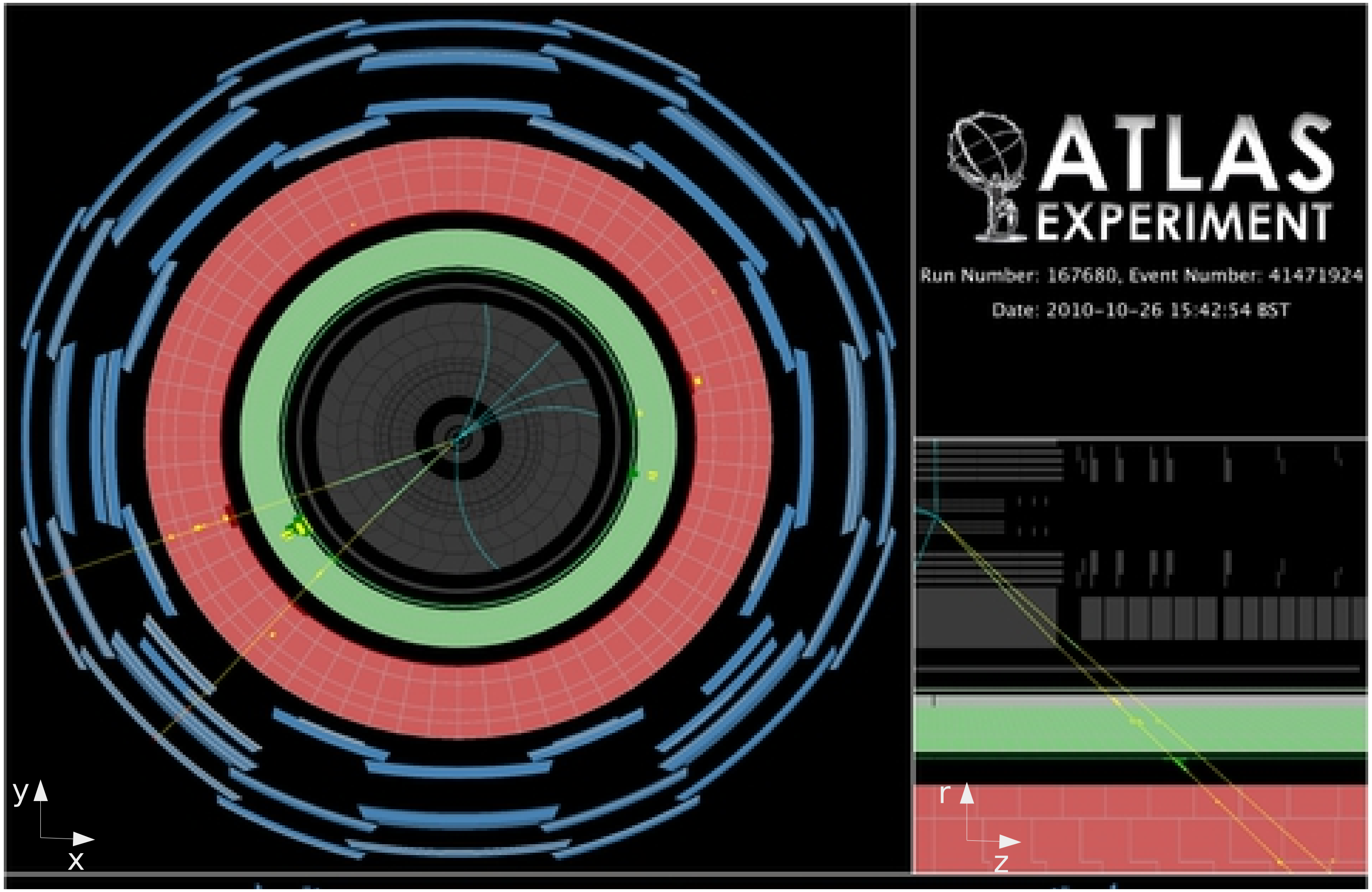
Track parameters

- Remember: The track is characterized by its 5 parameters as given at the “perigee surface”
  - At each sensor surface one can use a different parametrization
    - Track parameters given in the local reference frame of the surface
    - Error/Covariance matrix



The choice of parametrization depends
on the detector layout

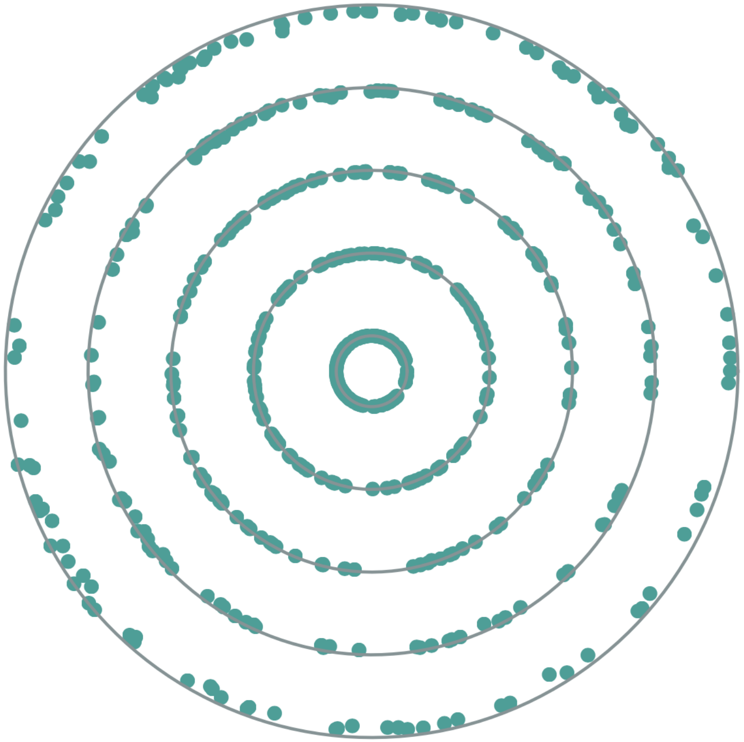
Motion of a charged particle in a uniform magnetic field



$$\vec{F} = q\,\vec{v} \times \vec{B} \qquad\qquad p_T(GeV/c) = 0.3\,q\,B(T)\,\rho(m)$$

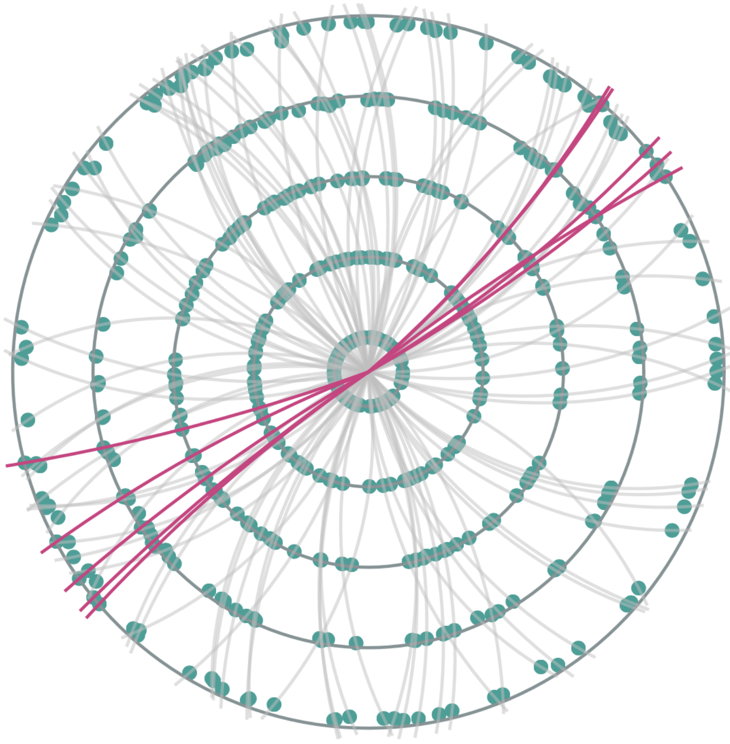
Pattern recognition

From hits



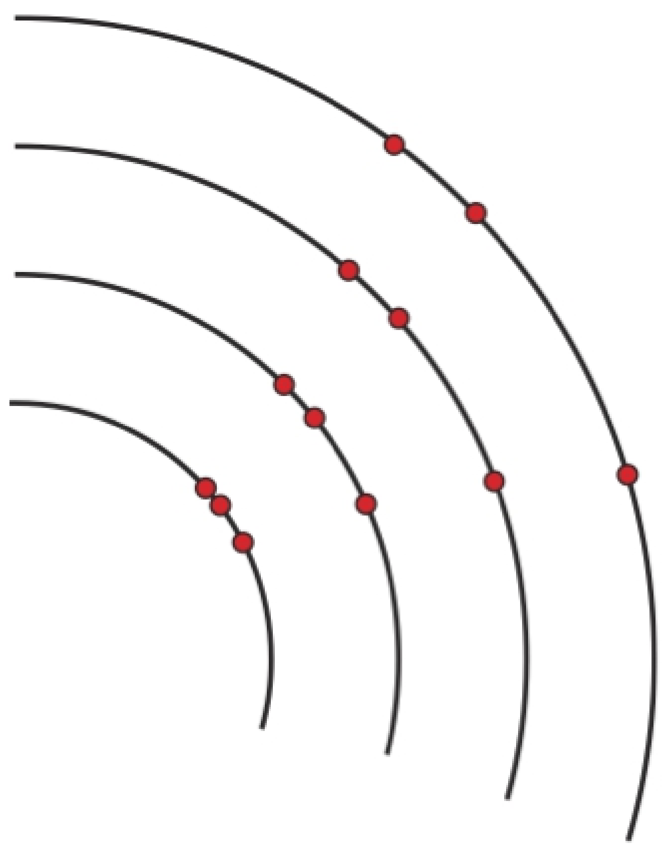
⇒

...to tracks



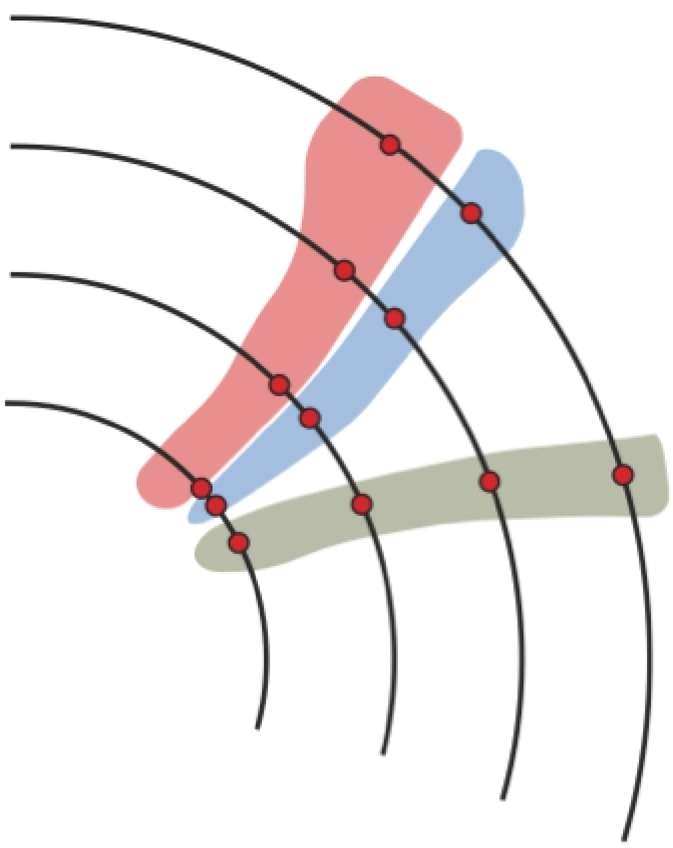
Pattern recognition

- The main goal of the pattern recognition is to associate hits to tracks
  - Efficient: all hits
  - Robust: no noise and no hits from other tracks
- Pattern recognition is a field of applied mathematics
  - It makes use of statistics, cluster analysis, combinatorial optimization, etc
  - The choice of the algorithm depends heavily in the type of measurements
    - 2D vs 3D points
  - And in the track model
    - Detector shape and B field
  - Hough space transform, template matching, minimum spanning tree, local pattern recognition
- Hit-to-track association
  - Defined by pattern recognition
  - Later altered by tracking
    - Removing bad hits & outliers
  - Noisy channels tend to be the “party spoilers”



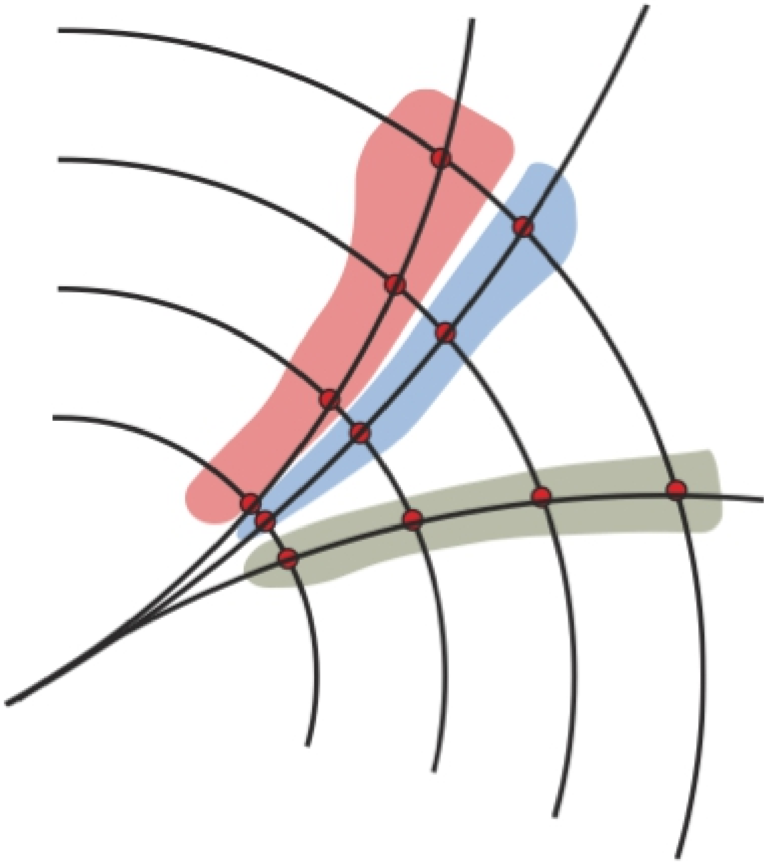
Pattern recognition

- The main goal of the pattern recognition is to associate hits to tracks
  - Efficient: all hits
  - Robust: no noise and no hits from other tracks
- Pattern recognition is a field of applied mathematics
  - It makes use of statistics, cluster analysis, combinatorial optimization, etc
  - The choice of the algorithm depends heavily in the type of measurements
    - 2D vs 3D points
  - And in the track model
    - Detector shape and B field
  - Hough space transform, template matching, minimum spanning tree, local pattern recognition
- Hit-to-track association
  - Defined by pattern recognition
  - Later altered by tracking
    - Removing bad hits & outliers
  - Noisy channels tend to be the “party spoilers”



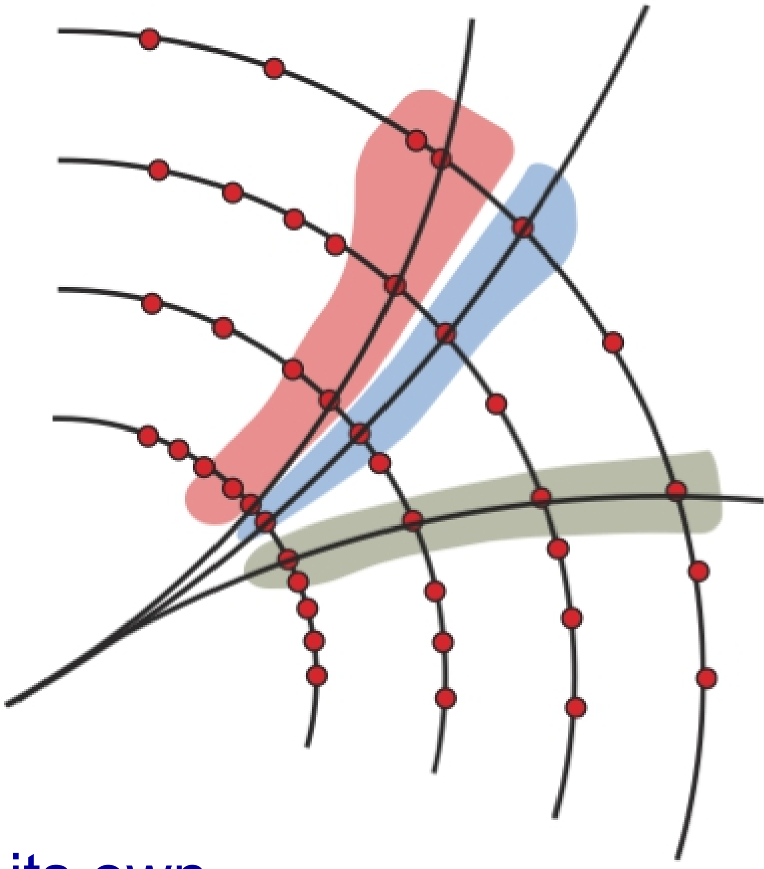
Pattern recognition

- The main goal of the pattern recognition is to associate hits to tracks
  - Efficient: all hits
  - Robust: no noise and no hits from other tracks
- Pattern recognition is a field of applied mathematics
  - It makes use of statistics, cluster analysis, combinatorial optimization, etc
  - The choice of the algorithm depends heavily in the type of measurements
    - 2D vs 3D points
  - And in the track model
    - Detector shape and B field
  - Hough space transform, template matching, minimum spanning tree, local pattern recognition
- Hit-to-track association
  - Defined by pattern recognition
  - Later altered by tracking
    - Removing bad hits & outliers
  - Noisy channels tend to be the "party spoilers"



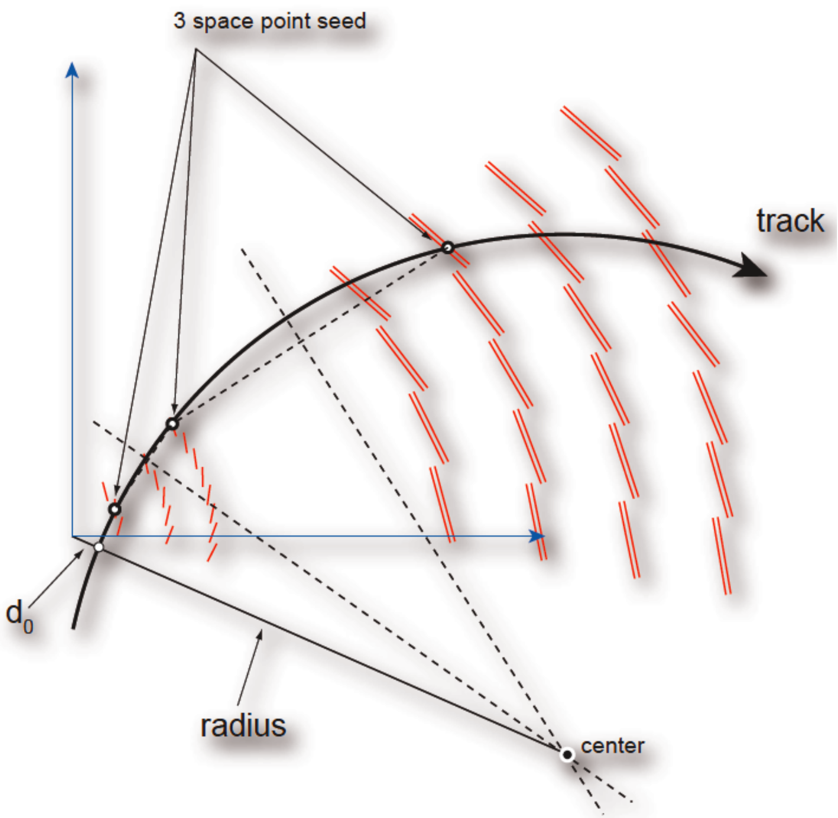
Pattern recognition

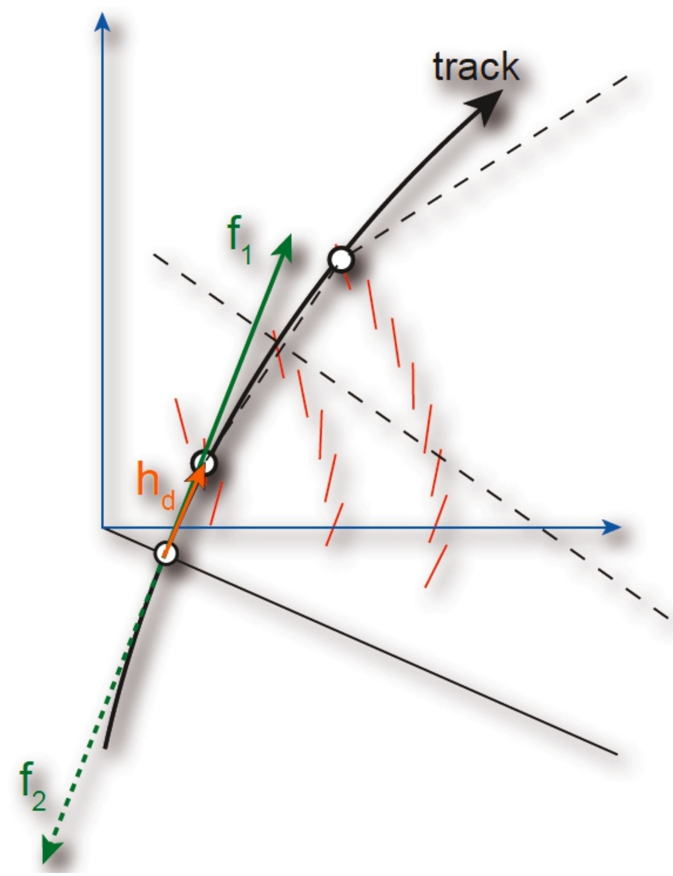
- The main goal of the pattern recognition is to associate hits to tracks
  - Efficient: all hits
  - Robust: no noise and no hits from other tracks
- Pattern recognition is a field of applied mathematics
  - It makes use of statistics, cluster analysis, combinatorial optimization, etc
  - The choice of the algorithm depends heavily in the type of measurements
    - 2D vs 3D points
  - And in the track model
    - Detector shape and B field
  - Hough space transform, template matching, minimum spanning tree, local pattern recognition
- Hit-to-track association
  - Defined by pattern recognition
  - Later altered by tracking
    - Removing bad hits & outliers
  - Noisy channels tend to be the "party spoilers"
- In summary: pattern recognition is an art on its own



Pattern recognition:

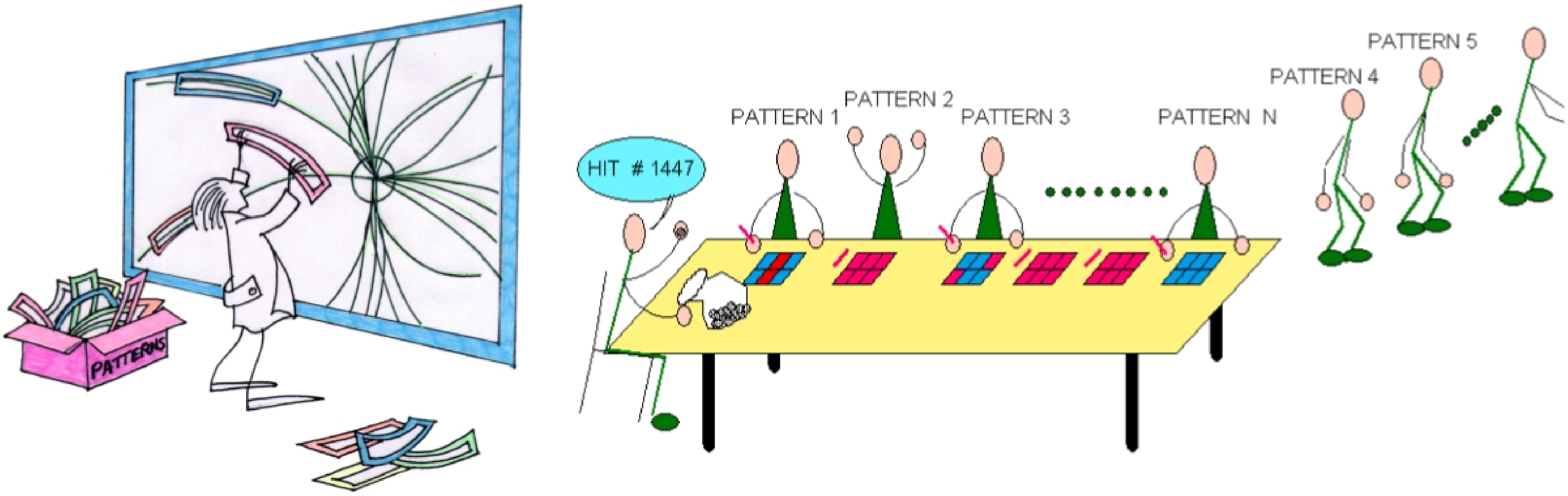
- 3 points seed:
  - Adding other measurements: (inside-out or outside-in) may use 3 consecutive measurements (compute a circle) and extrapolate the track (outwards or inwards) attaching near-by measurements





Pattern recognition

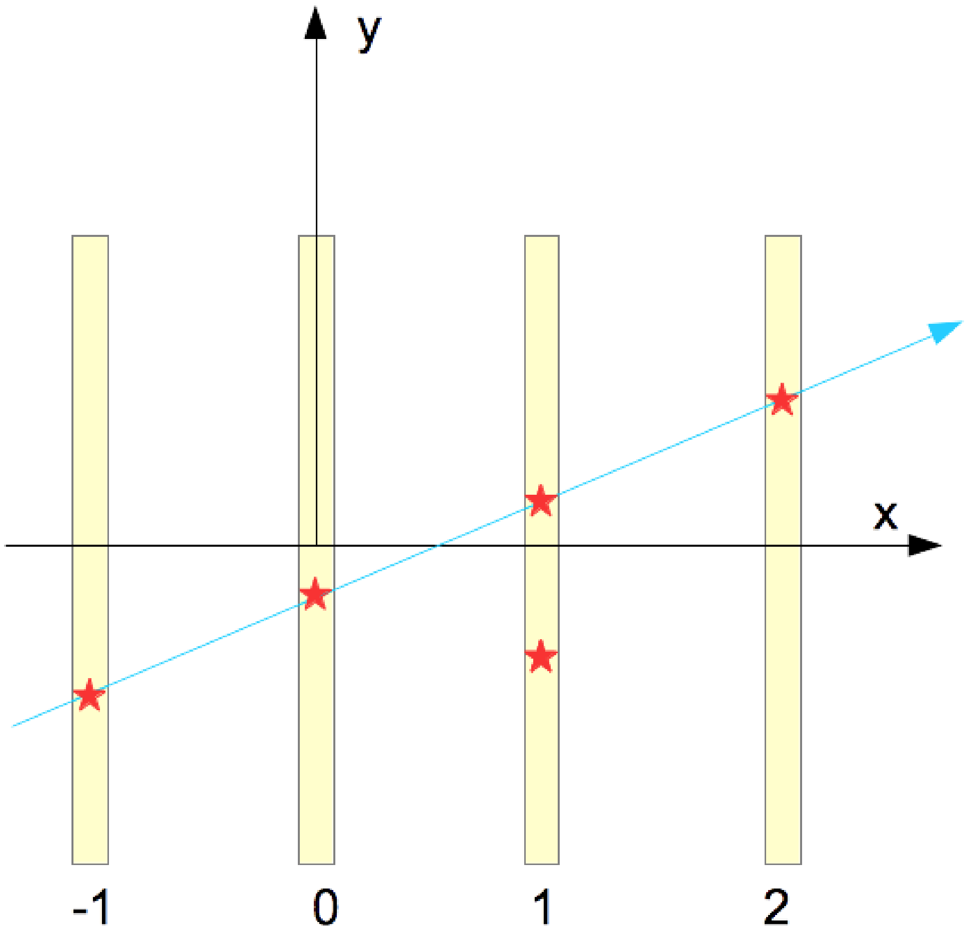
- It is possible to perform an *online* pattern recognition for a fast online tracking
  - Why fast tracking ?
    - Online one has a limited time to decide if the event is stored or discarded
    - A finite set of track topologies is used.
      - p_T based and possible routes from collision point
  - Possibility to implement a “fast tracking” based trigger
    - Trigger on secondary vertices → online B-tagging



Pattern recognition: Hough transform

- Hough transform it is a technique for digital image detection
  - It can detect the points that belong to a line (straight, circle, ellipse, helix...)
    - So the points coordinates satisfy the line equation
  - The Hough space has as many dimensions as the number of parameters to determine
    - Straight line (2D): 2. Circle (2D): 3. Helix (3D): 5
    - Then take all possible tuples (of track parameters) that will pass by each point
      - Infinite combinations → discretize &/or use constraints (e.g.: particles were originated at the center of the detector).
      - Count how many times a given parameter tuple is possible / find intersections
      - Select the most frequent parameter tuple (more intersections)
        - Use the points for the track fitting
        - Initial track parameters → use most frequent parameter tuple
- Example: straight line
  - Points are given as many available (x, y) tuples. 2D space: $y = x \tan\theta + y_0$
  - Lines are given as ($\tan\theta$, y_0) tuples Hough space: $y_0 = y - x \tan\theta$
  - Solve: draw lines in Hough space and check for intersections

Pattern recognition: Hough transform



Pattern recognition: Hough transform

Table 2: List of hits recorded in this event. In total 5 hits: 4 hits from the track plus a noise hit. Units are arbitrary.

x	y
-1	-0.18
0	0.39
1	1.02
1	0.77
2	1.57

Pattern recognition: Hough transform

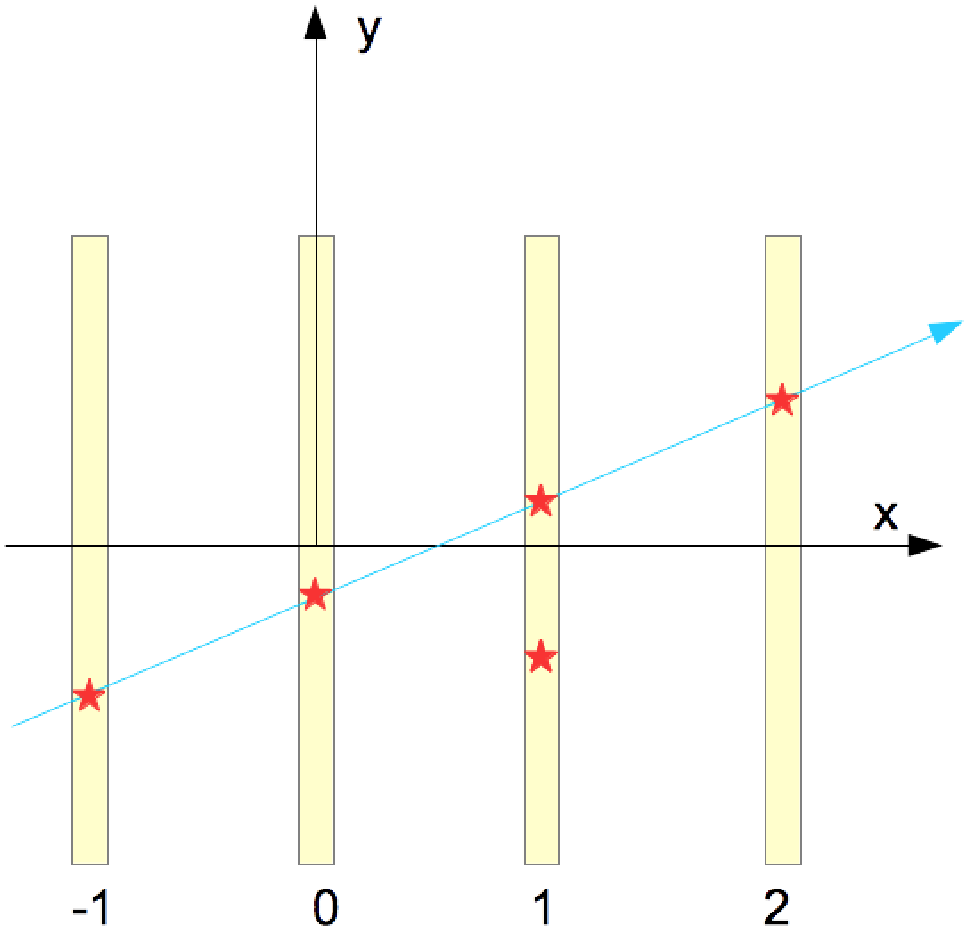


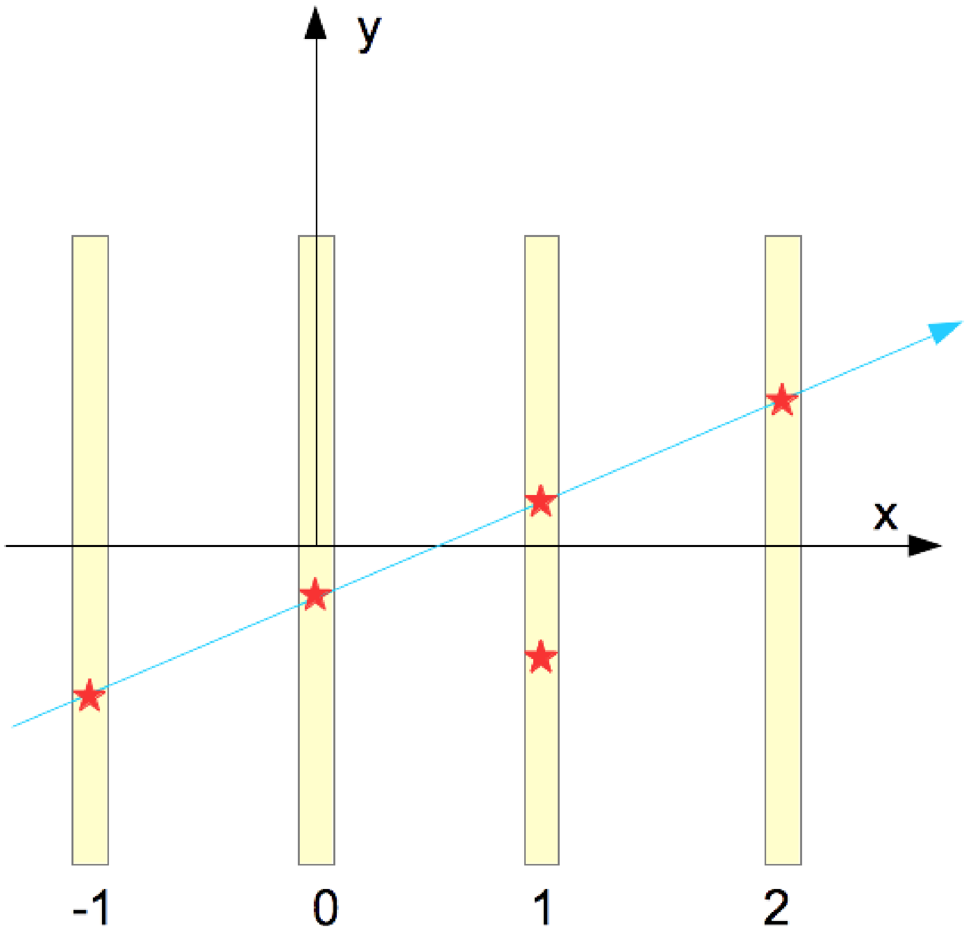
Table 2: List of hits recorded in this event. In total 5 hits: 4 hits from the track plus a noise hit. Units are arbitrary.

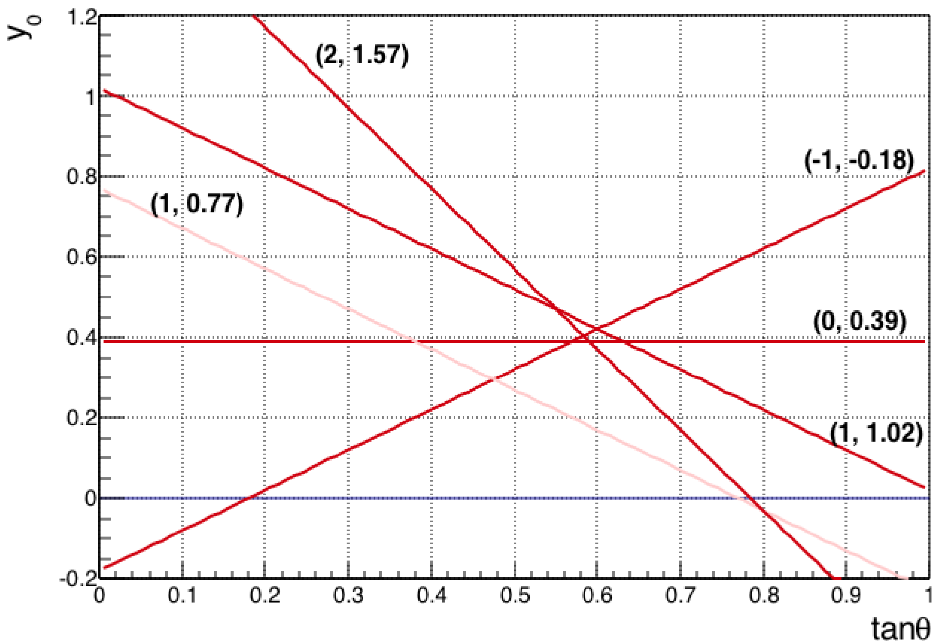
x	y
-1	-0.18
0	0.39
1	1.02
1	0.77
2	1.57

Pattern recognition: Hough transform

Table 2: List of hits recorded in this event. In total 5 hits: 4 hits from the track plus a noise hit. Units are arbitrary.

x	y
-1	-0.18
0	0.39
1	1.02
1	0.77
2	1.57





2D space: $y = x \tan\theta + y_0$

Hough space: $y_0 = y - x \tan\theta$

Pattern recognition: Hough transform

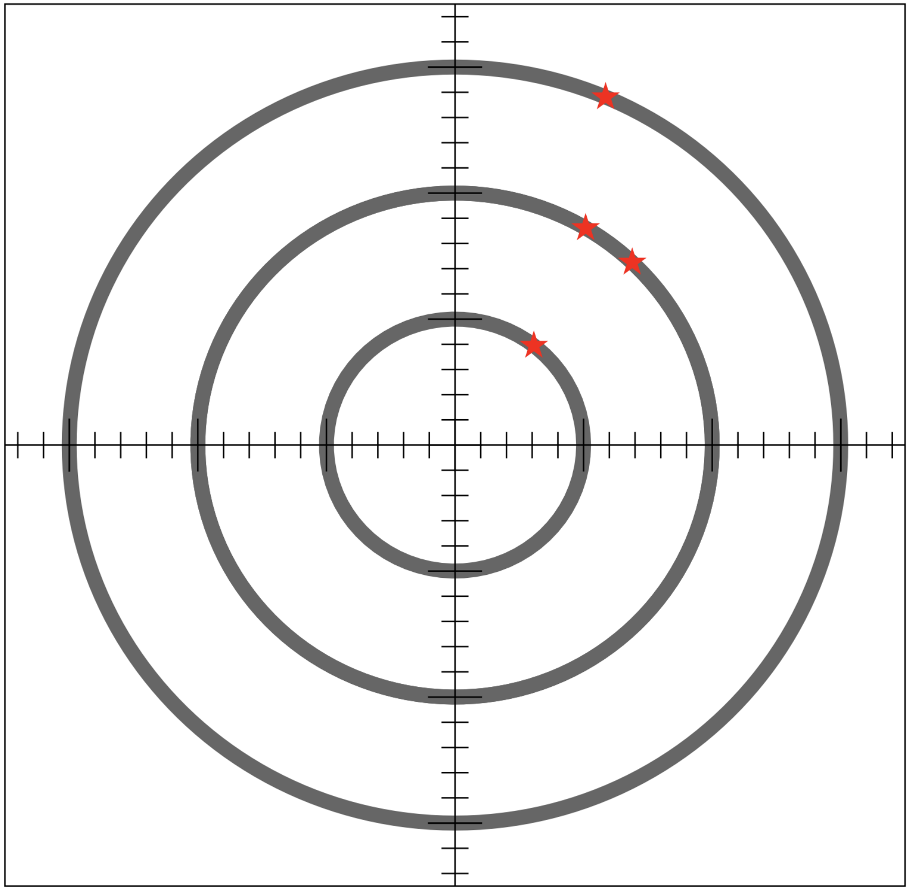


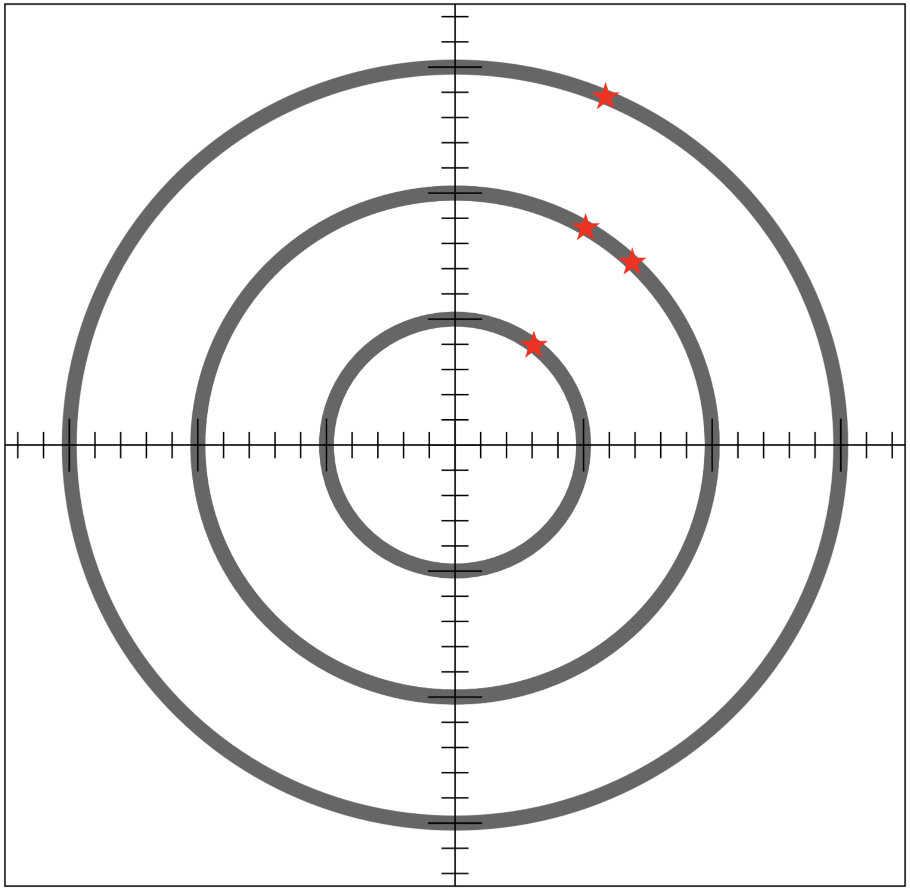
Table 3: List of hits recorded in this event. In total 4 hits: 3 hits from the track plus a noise hit.

x [mm]	y [mm]
61.3	79.0
101.6	172.3
137.8	145.0
117.1	276.2

$$r^2 - 2\,r\,(\rho + d_0)\sin(\alpha - \phi_0) + (\rho + d_0)^2 = \rho^2$$

$$(d_0 \to 0) : r - 2\,\rho\sin(\alpha - \phi_0) = 0$$

Pattern recognition: Hough transform

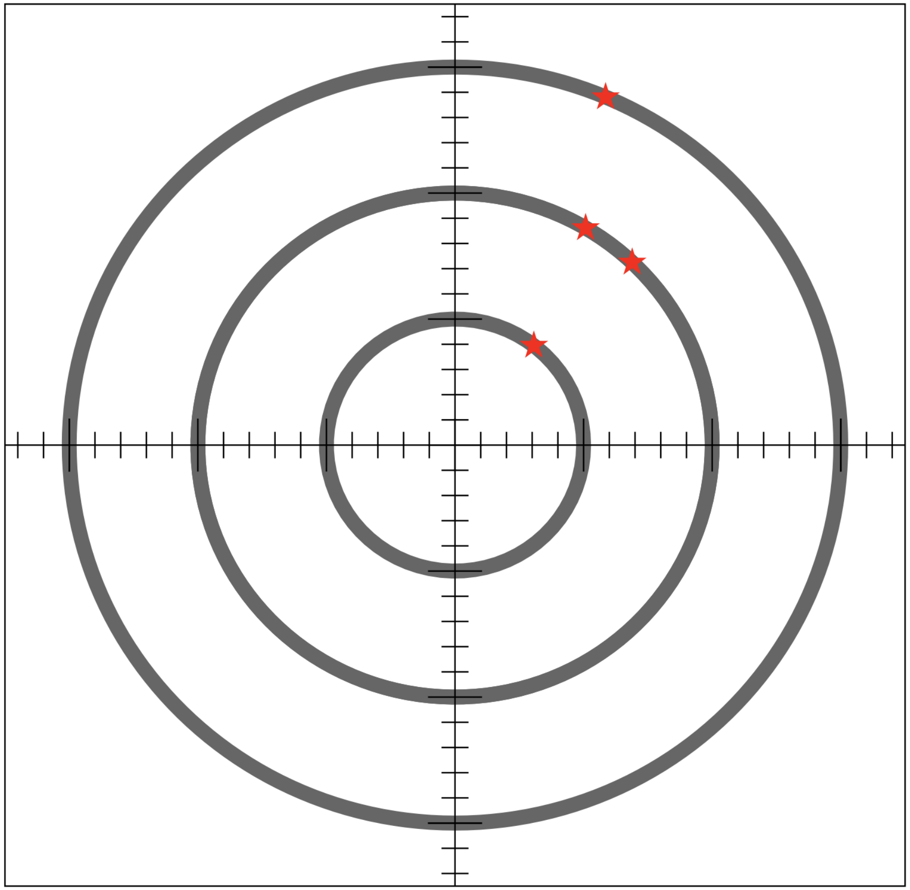


x [mm]	y [mm]	r [m]	α [rad]
61.3	79.0	0.1	0.9107
101.6	172.3	0.2	1.0381
137.8	145.0	0.2	0.8107
117.1	276.2	0.3	1.1698

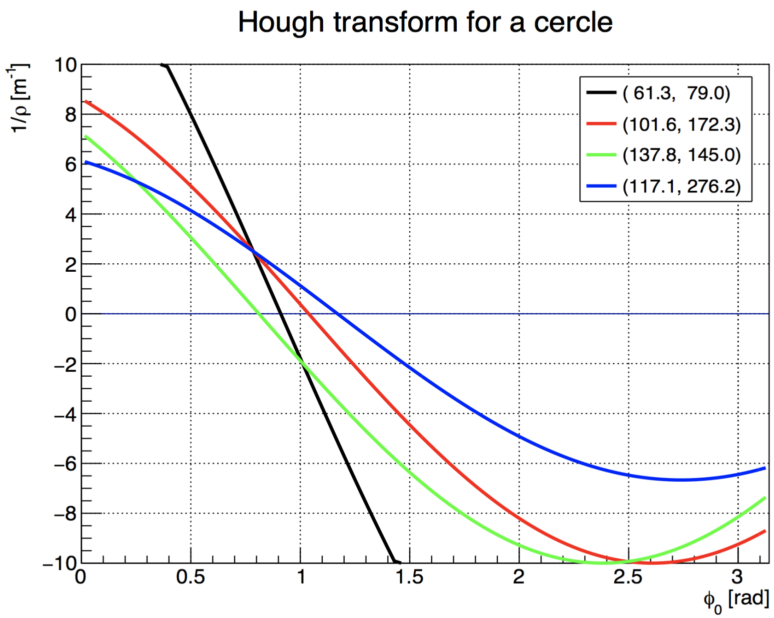
$$r^2 - 2\ r\ (\rho + d_0)\sin(\alpha - \phi_0) + (\rho + d_0)^2 = \rho^2$$

$$(d_0 \to 0) : r - 2\ \rho\sin(\alpha - \phi_0) = 0$$

Pattern recognition: Hough transform



x [mm]	y [mm]	r [m]	α [rad]
61.3	79.0	0.1	0.9107
101.6	172.3	0.2	1.0381
137.8	145.0	0.2	0.8107
117.1	276.2	0.3	1.1698

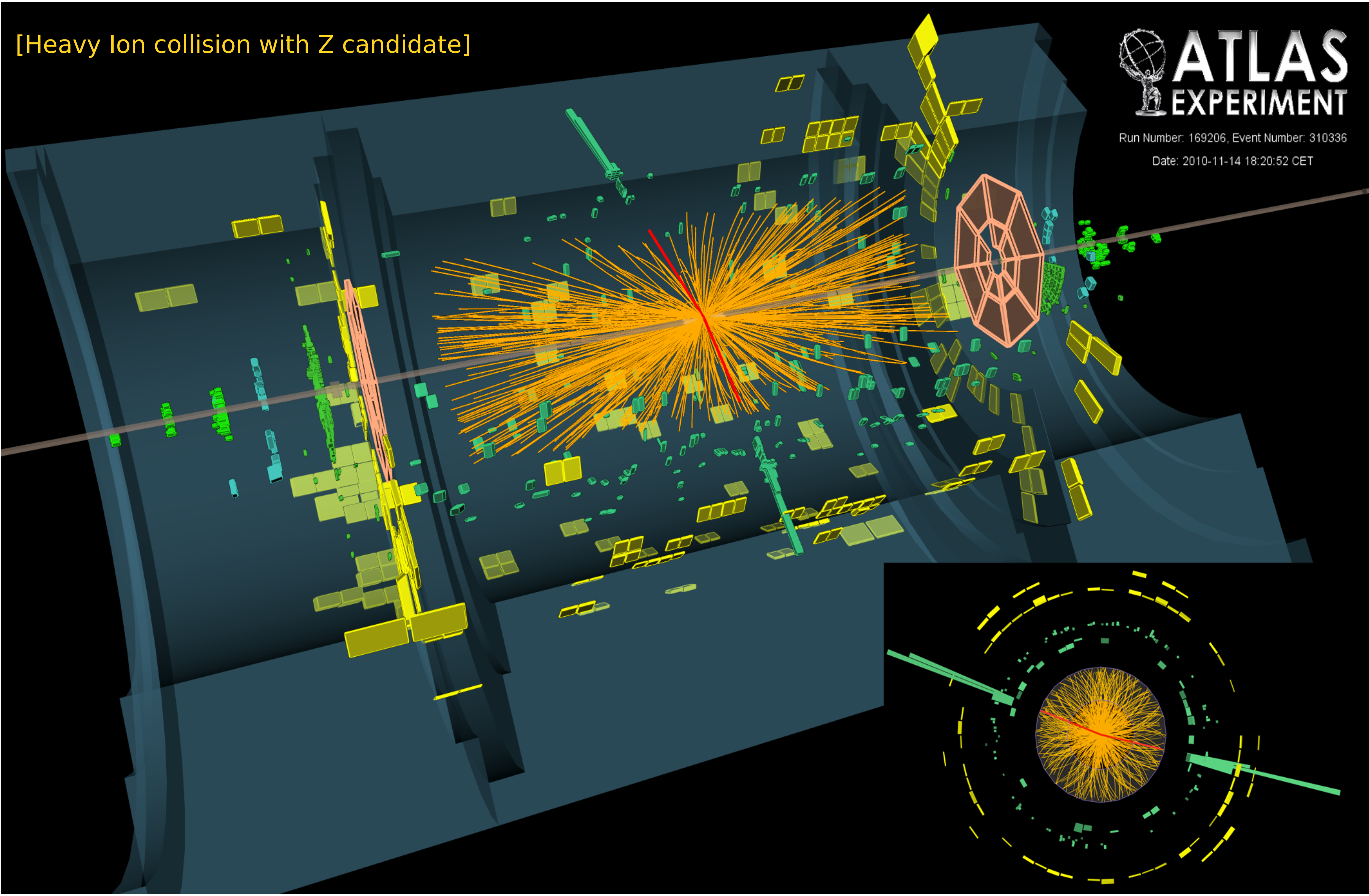


$$r^2 - 2\,r\,(\rho + d_0)\sin(\alpha - \phi_0) + (\rho + d_0)^2 = \rho^2$$

$$(d_0 \to 0) : r - 2\,\rho\sin(\alpha - \phi_0) = 0$$

$$\frac{1}{\rho} = \frac{2\,\sin(\alpha - \phi_0)}{r}$$

Example of event with many tracks



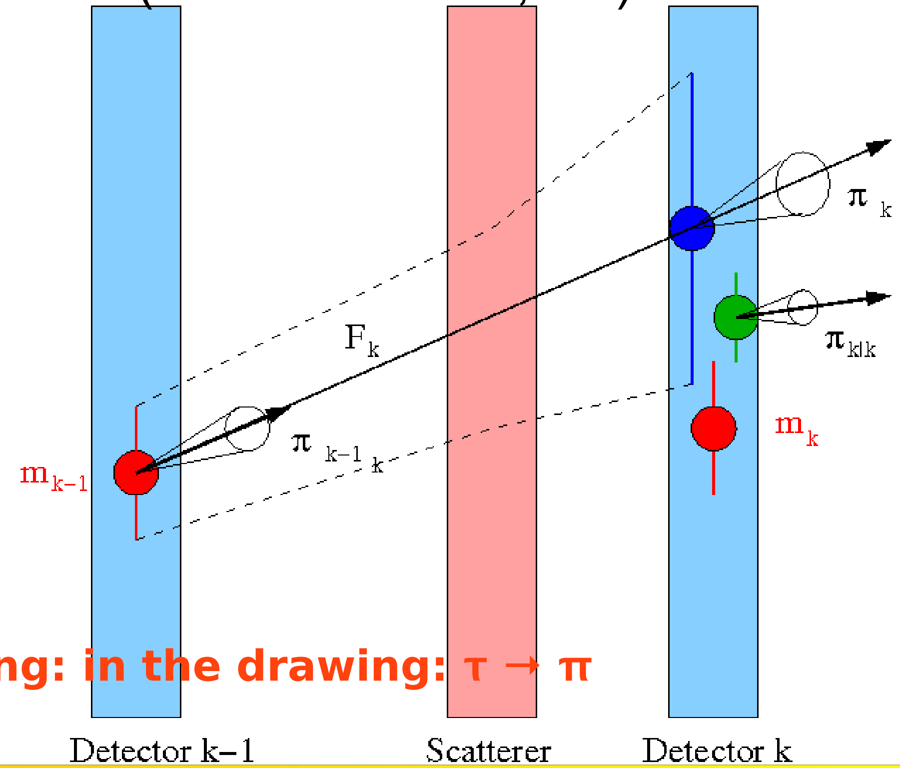
Track fitting with Kalman filter

- The Kalman filter was developed by R.E. Kalman during the 1950's
  - To solve differential matrix equations without matrix inversions
  - It is a method of estimating the states of dynamic systems
    - Applied by the NASA in the rocket trajectory control for the Apollo program
    - Military applications: compute plane trajectory by radar tracking
- Assumption:
  - The trajectory of a particle between two adjacent surfaces is described by a deterministic function plus random disturbances (material effects, etc)
  - The system equation: propagates the estate in one surface to the next

$$\tau_k = F_k(\tau_{k-1}) + P_k \boldsymbol{\delta}_k \qquad \langle \boldsymbol{\delta}_k \rangle = 0 \qquad Cov(\boldsymbol{\delta}_k) = Q_k$$

  - The measurement equation: mapping the track in the surface and considers some measurement error

$$\boldsymbol{m}_k = H_k(\boldsymbol{\tau}_k) + \varepsilon_k \qquad \langle \boldsymbol{\varepsilon}_k \rangle = 0 \qquad Cov(\boldsymbol{\varepsilon}_k) = V_k$$



Warning: in the drawing: τ → π

Track fitting with Kalman filter

- The aim is to estimate the track parameters from the observations
  - From *k-1* observations and a k^{th} measurement: obtain a new *k* estimate

$$\left\{\{\boldsymbol{m}_1,\ldots,m_{k-1}\},\quad \boldsymbol{\tau}_{k-1}\right\} \;+\; \boldsymbol{m}_k \quad\rightarrow\quad \boldsymbol{\tau}_k$$

  - **Prediction** $\tau_{k|k-1}=F_k(\tau_{k-1})+P_k\boldsymbol{\delta}_k$
    - and its covariance matrix (error): $C_{k|k-1}=F_k C_{k-1|k-1}F_k^T+P_k Q_k P_k^T$
  - **Filtering**, based on $\tau_{k/k\text{-}1}$ and m_k:
    - It consists in minimizing the following:

$$L(\boldsymbol{\tau}_k)=(\boldsymbol{m}_k-H_k(\boldsymbol{\tau}_k))^T V_k^{-1}(\boldsymbol{m}_k-H_k(\boldsymbol{\tau}_k))+(\boldsymbol{\tau}_{k|k-1}-\boldsymbol{\tau}_k)^T C_{k|k-1}(\boldsymbol{\tau}_{k|k-1}-\boldsymbol{\tau}_k)$$

    - The solution should be well known by now:

$$\boldsymbol{\tau}_{k|k}=\boldsymbol{\tau}_{k|k-1}-\left[(H_k^T V^{-1}H_k)+C_{k|k-1}\right]^{-1}\left[H_k^T V^{-1}(\boldsymbol{m}_k-H_k(\boldsymbol{\tau}_k))\right]$$

    - And its covariance matrix (error):

$$C_{k|k}=\left[(H_k^T V^{-1}H_k)+C_{k|k-1}\right]^{-1}$$

  - The residual is thus:

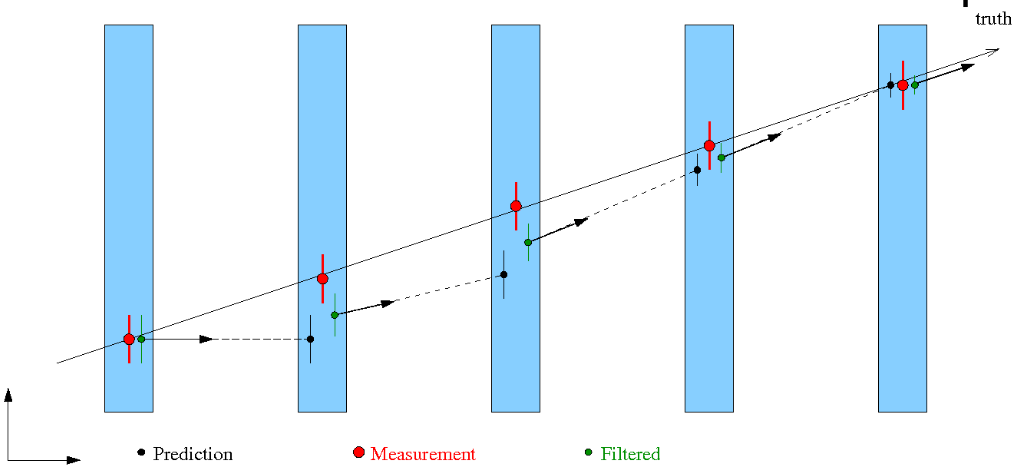
$$\boldsymbol{r}_{k|k}=\boldsymbol{m}_k-H_k\boldsymbol{\tau}_{k|k}$$

    - Which allows to compute a χ^2 in order to test the goodness of the fit

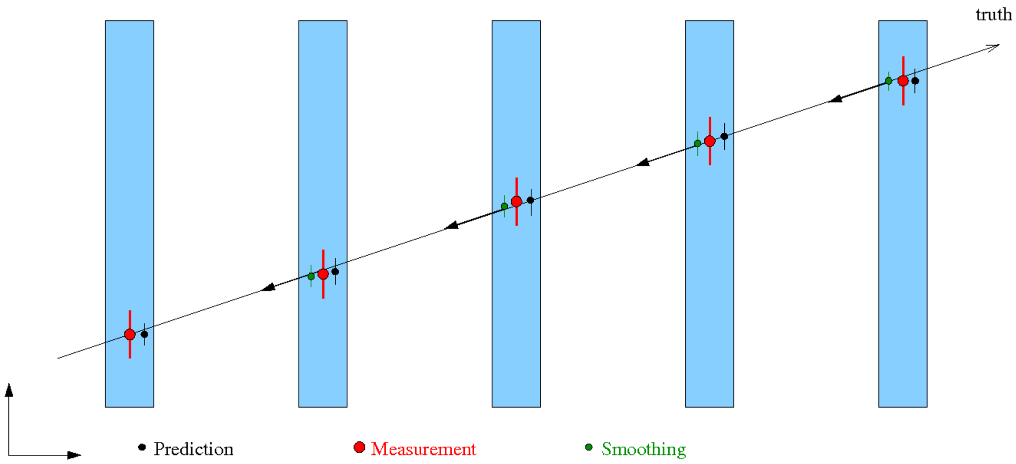
      that needs some smoothing. $\chi^2_{k|k}=\boldsymbol{r}_k^T V_k^{-1}\boldsymbol{r}_k \qquad \chi^2=\sum_k \chi_k^2$

Track fitting with Kalman filter

- Estimate of the track parameters and state at the detector surfaces
  - Filtering from estimate *k*-1 to *k*
    - Outer points estimates have more information than inner points

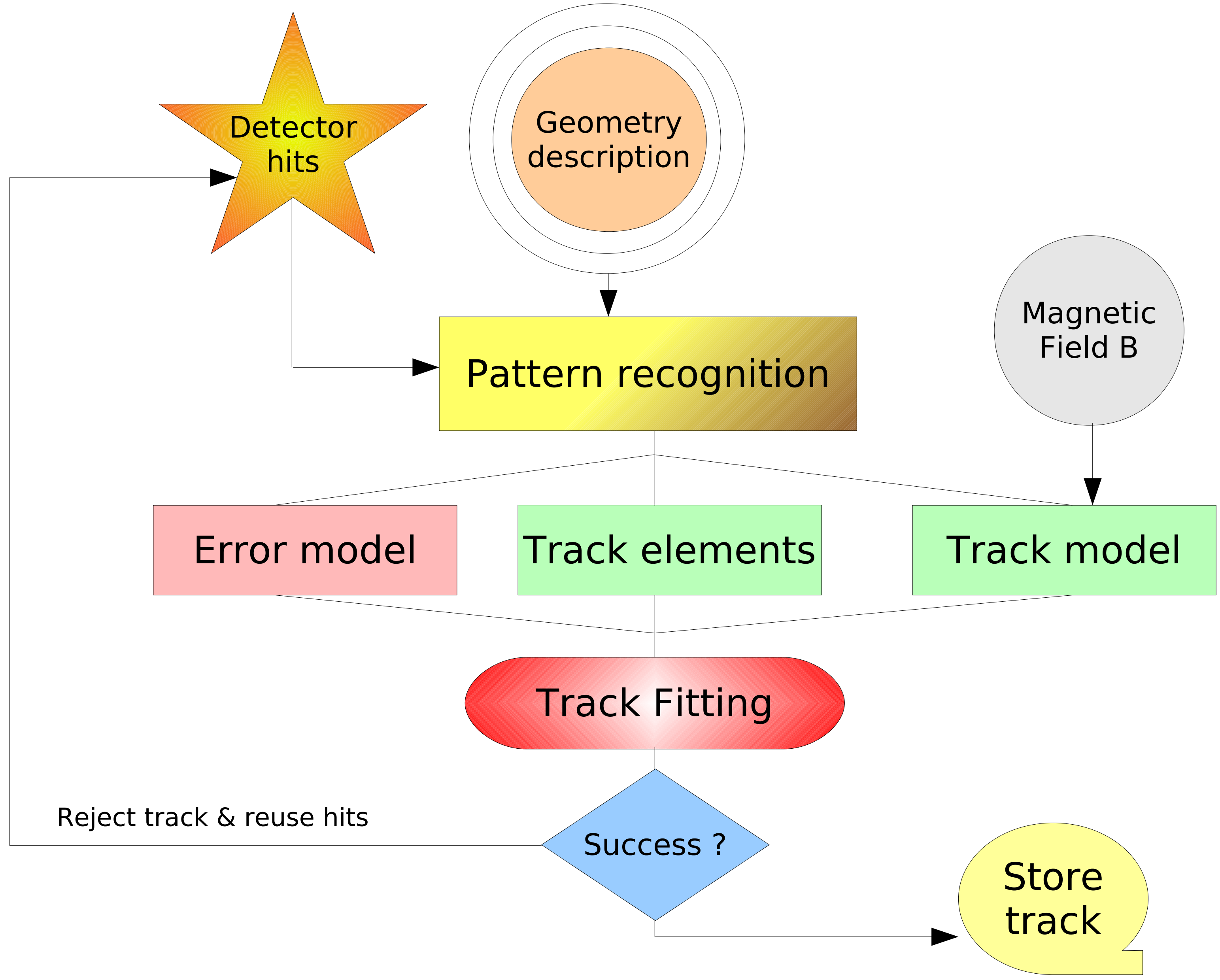


  - Smoothing: from estimate k to k-1 (sort of backward filter)
    - All points estimates have the same information



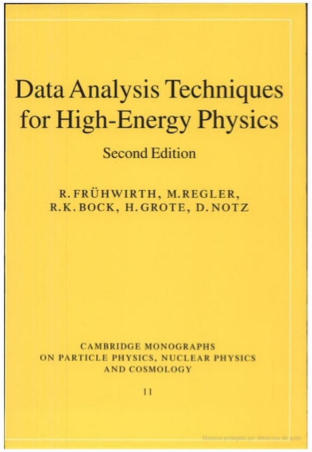
Track fitting summary

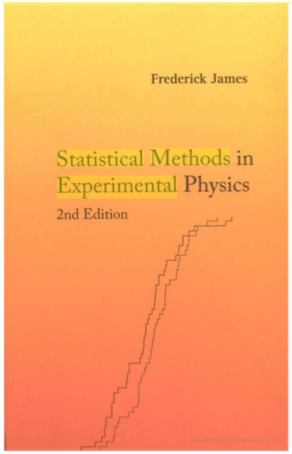
- From detector hits to particle trajectories

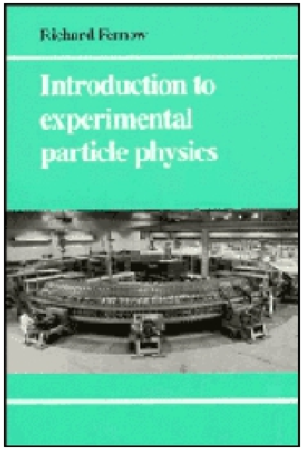


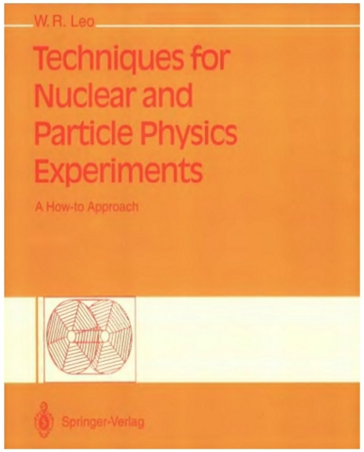
Bibliography

- “Data Analysis Techniques for High-Energy Physics”
  - R. Früwirth et al.
- “Statistical Methods in Experimental Physics”
  - F. James
- “Introduction to experimental particle physics”
  - R. Fernow
- “Techniques for Nuclear and Particle Physics Experiments”
  - W.R. Leo
- “Inner Detector Reconstruction: tracking”
  - A. Salzburger
  - Artemis school, 15-19 September 2008, MPI Munich, Germany

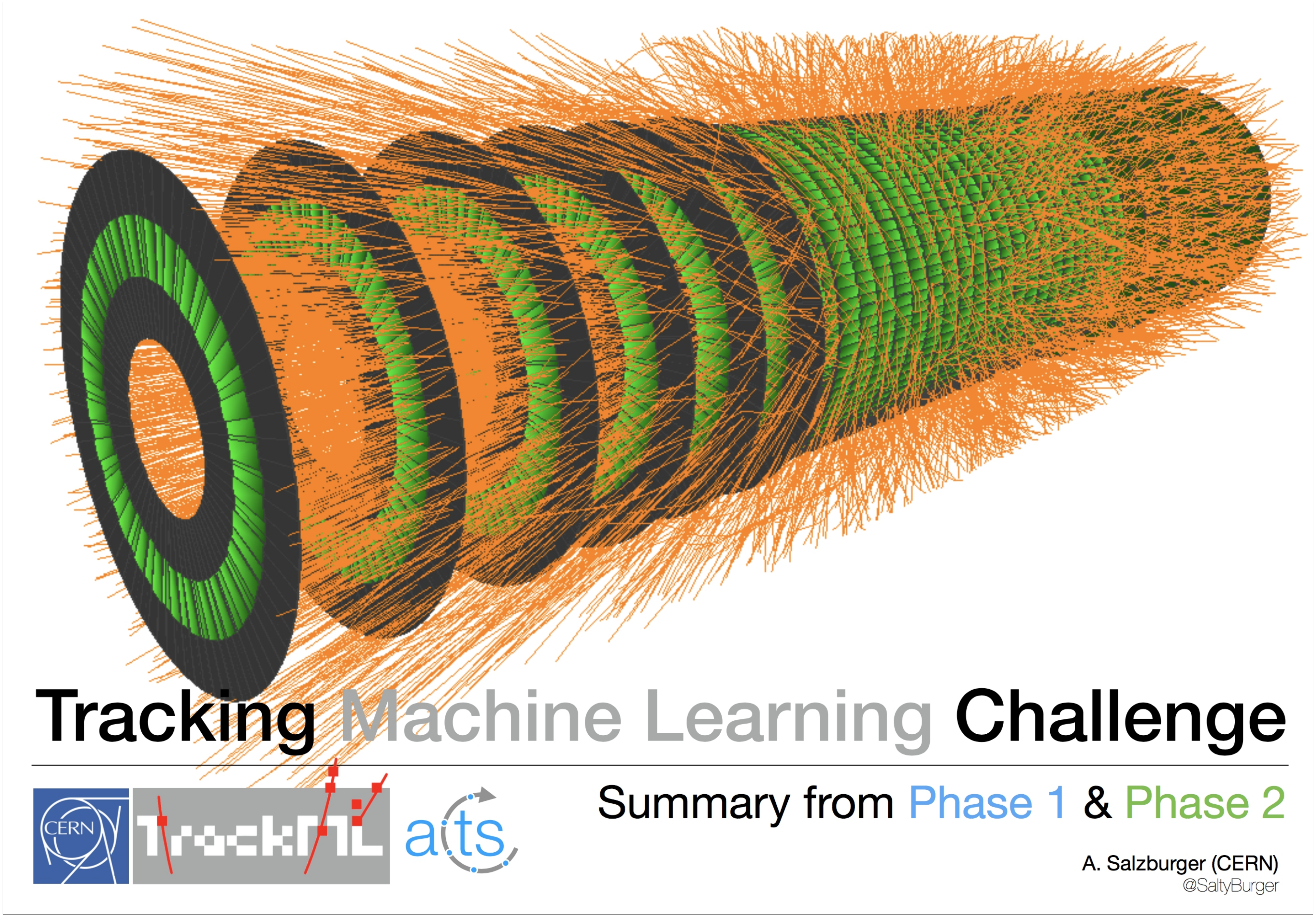






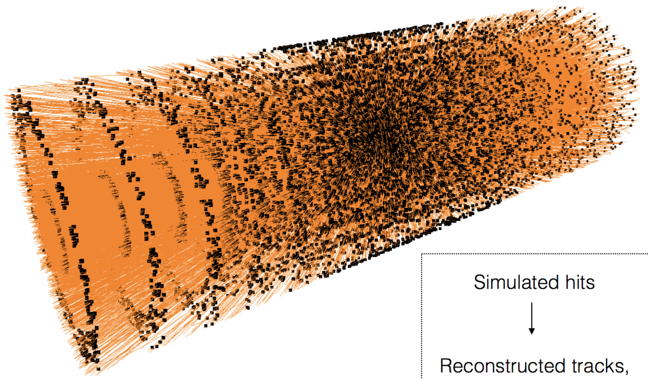


Tracking ML challenge



Tracking ML challenge

The challenge

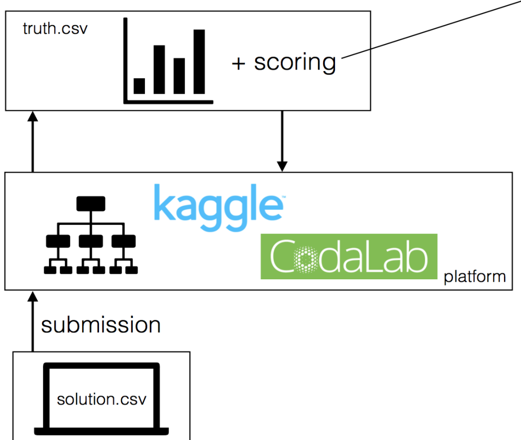


Simulated hits

↓

Reconstructed tracks, Sequence of labelled hits

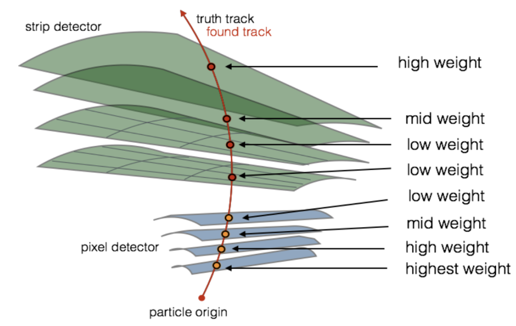
Submission

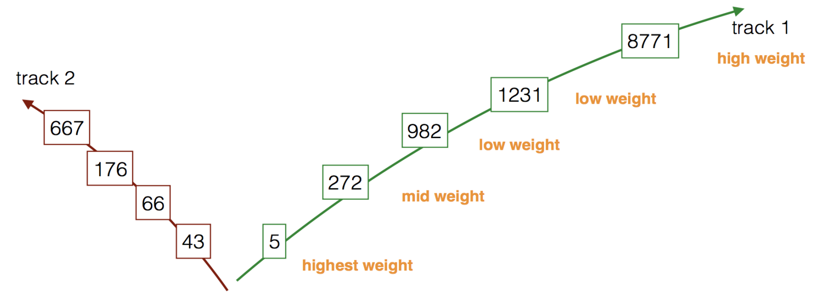


hit_id	track_id
5	1
272	1
982	1
1231	1
8771	1
43	2
66	2
176	2
667	2

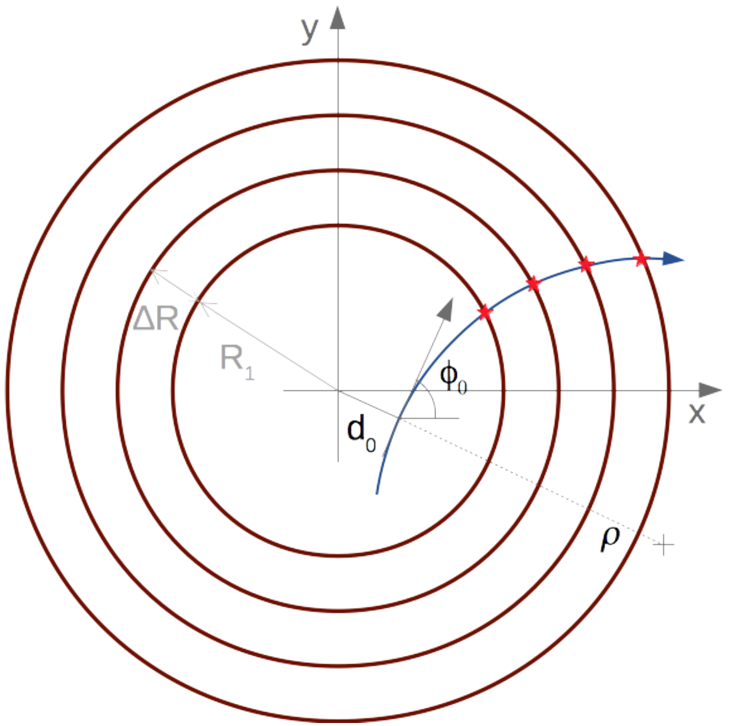
participant

hits on track have **weights**

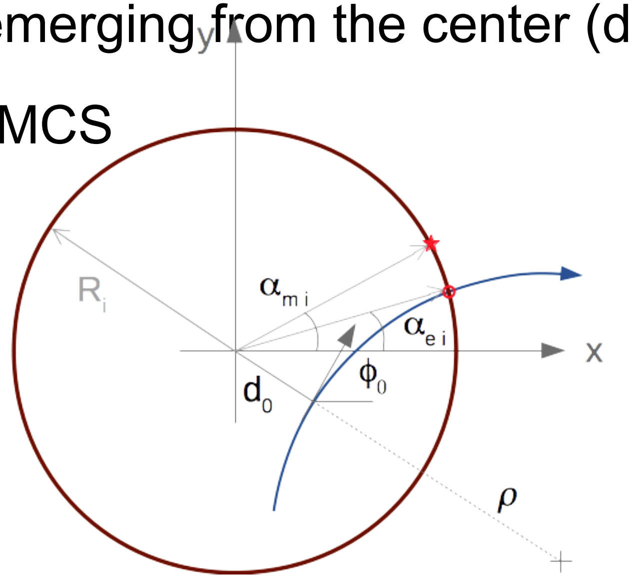




Example: circular tracks



- Consider circular tracks in 2D (X,Y)
- 4 circular sensor layers centered at (0,0)
  - First layer at R_1, the rest uniformly spaced by ΔR
  - Resolution: σ (same for all layers)
- Track parameters: d_0, ρ & φ_0
- Estimate track parameter uncertainties by inverting the track fit matrix of the Chi2 fit
  - Residuals: arc difference between measurements and extrapolations
  - Tracks emerging from the center (d_0 = 0)
  - Neglect MCS



$$r^2+2r(\rho-d_0)\sin(\alpha-\phi_0)+(\rho-d_0)^2=\rho^2$$

$$\boldsymbol{R}=\begin{pmatrix} R_1(\alpha_{m1}-\alpha_{e1}) \\ (R_1+\Delta R)(\alpha_{m2}-\alpha_{e2}) \\ (R_1+2\Delta R)(\alpha_{m3}-\alpha_{e3}) \end{pmatrix}$$

$$\delta\tau=-\left[\left(\frac{dr}{d\tau}\right)^T V^{-1}\left(\frac{dr}{d\tau}\right)\right]^{-1}\left[\left(\frac{dr}{d\tau}\right)^T V^{-1} r(\tau_0)\right]$$