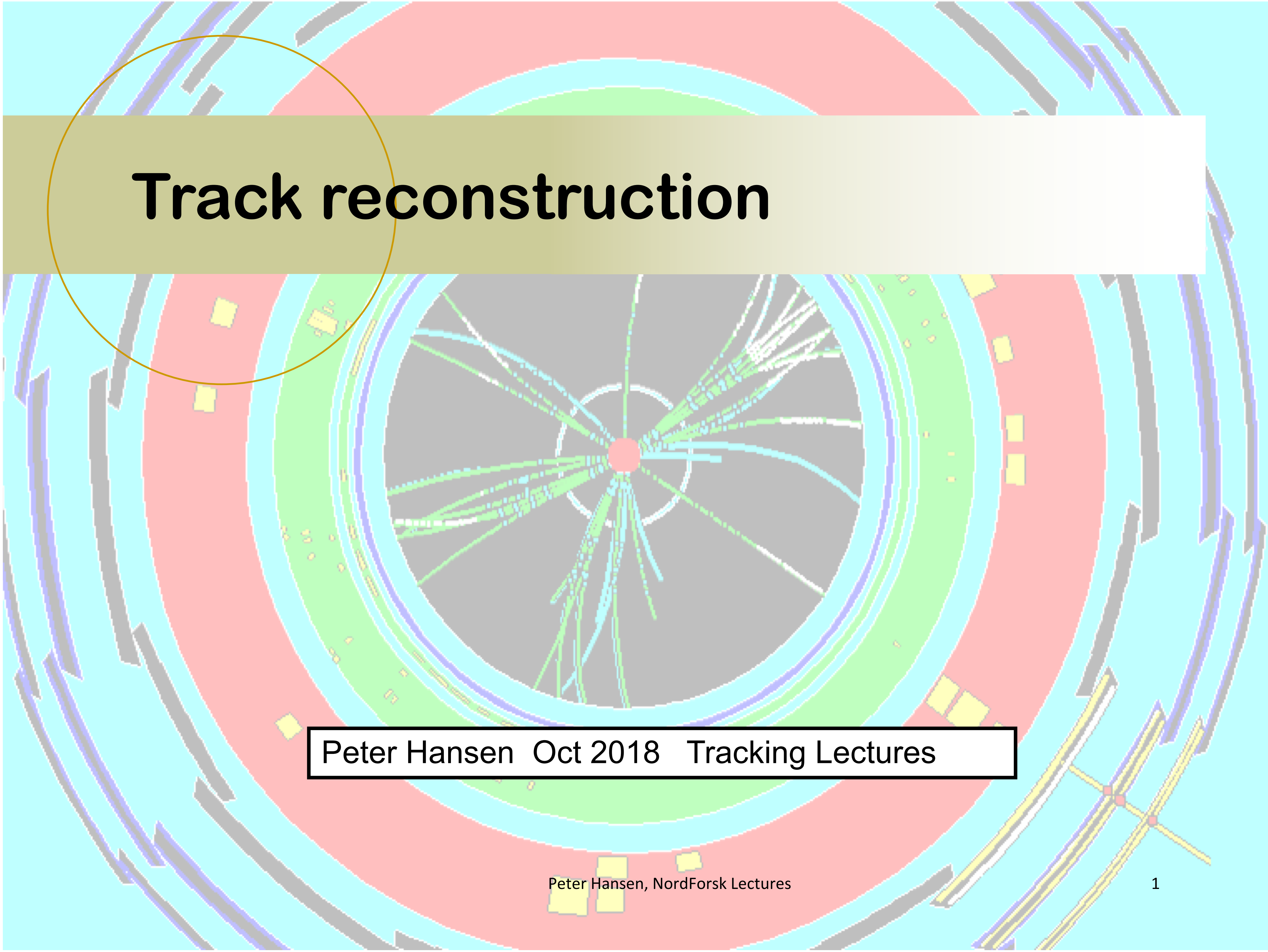
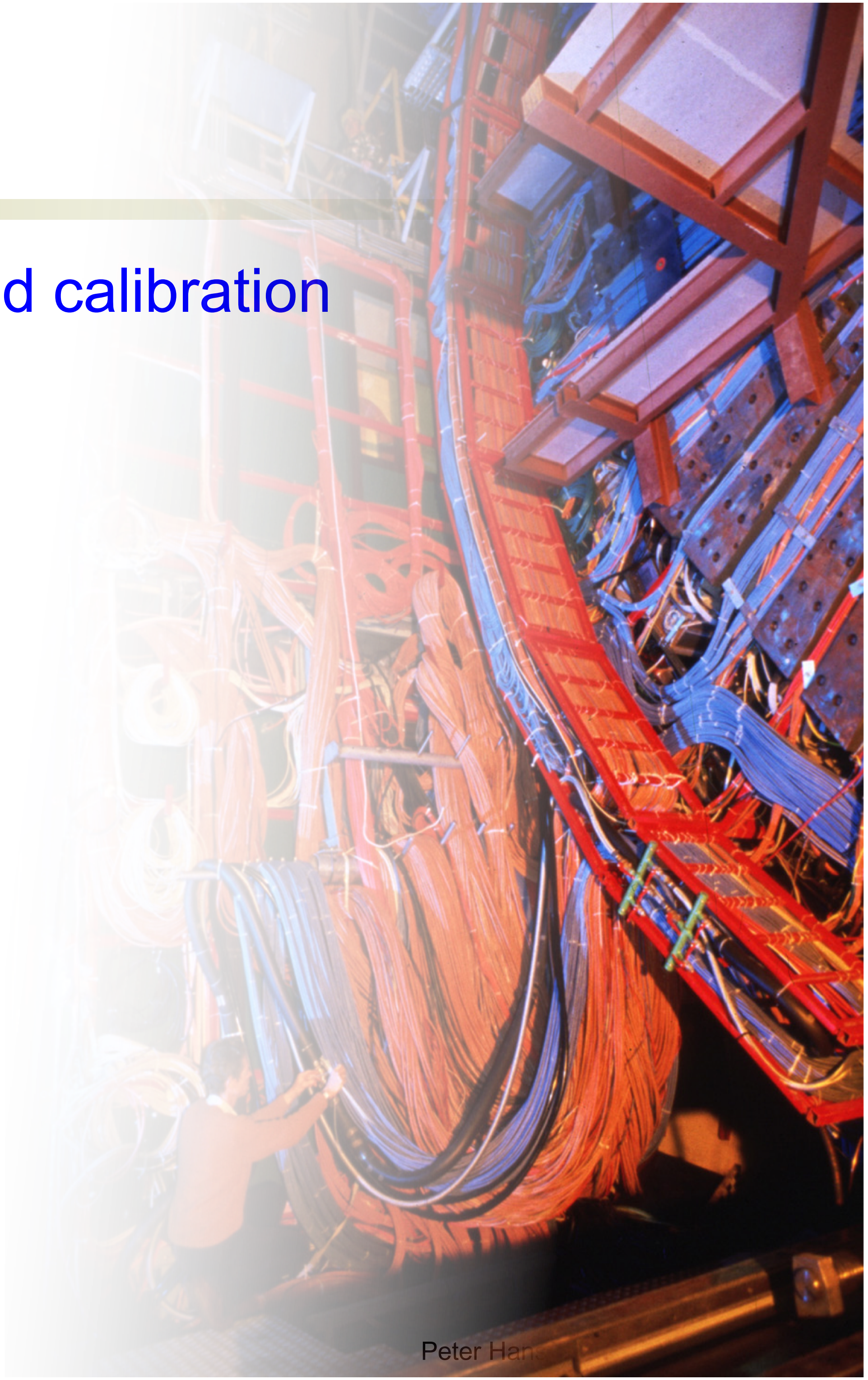
Track reconstruction



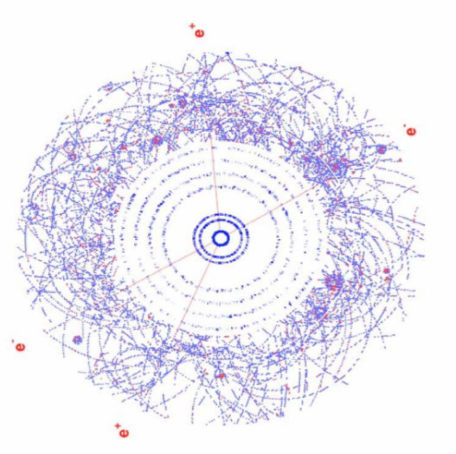
Peter Hansen Oct 2018   Tracking Lectures

Overview

- Spacepoint formation and calibration
- Pattern recognition
- Track fitting methods
- Vertexing
- Alignment
- High pile-up mitigation

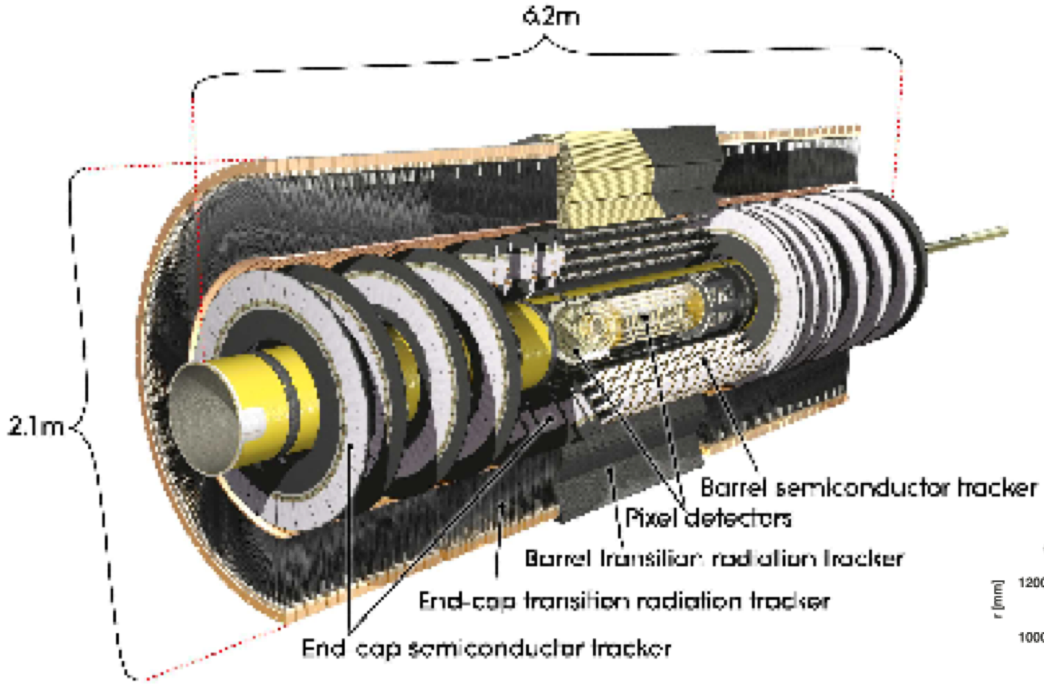


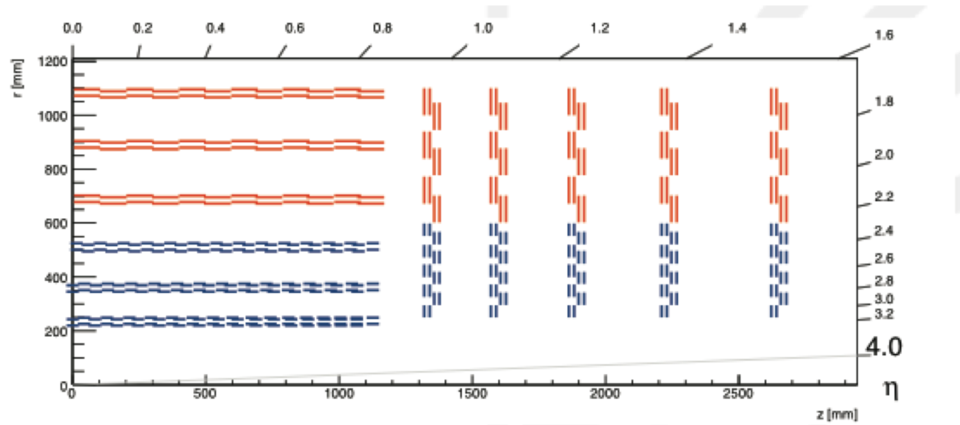
The tracking challenge



- Every second, 40 million beam-crossings are happening at the LHC, producing thousands of tracks from typically 60 individual collisions. About 1 kHz of the crossings are selected for later processing.
- Because of the high track density and high momentum very many channels are needed, causing rather large amounts of material in the tracking detectors.
- Thus, we need highly efficient and error-tolerant track-finders and –fitters, good calibration and alignment methods, robust vertexing and particle identification.
- In the future, the extrapolated computing technology cannot keep up with the flood of data using existing algorithms. New techniques will be necessary.

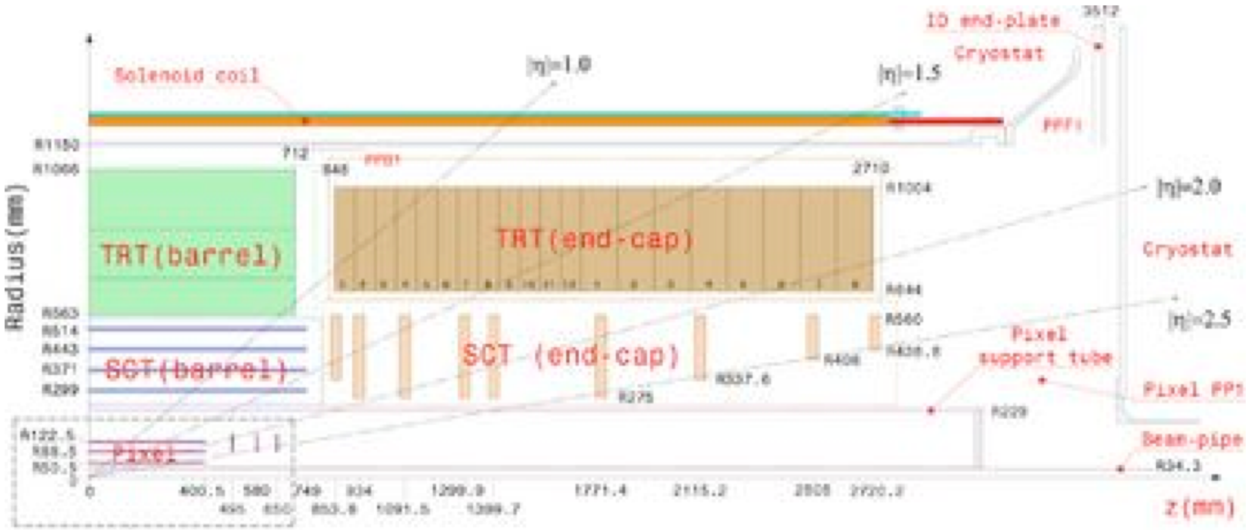
ATLAS and CMS Inner Trackers



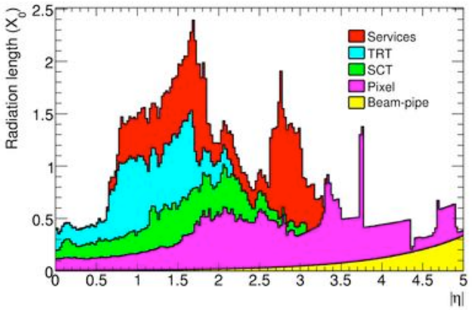


Many ATLAS and CMS examples are used this lecture. A general principle is to build detector planes roughly perpendicular to the tracks…

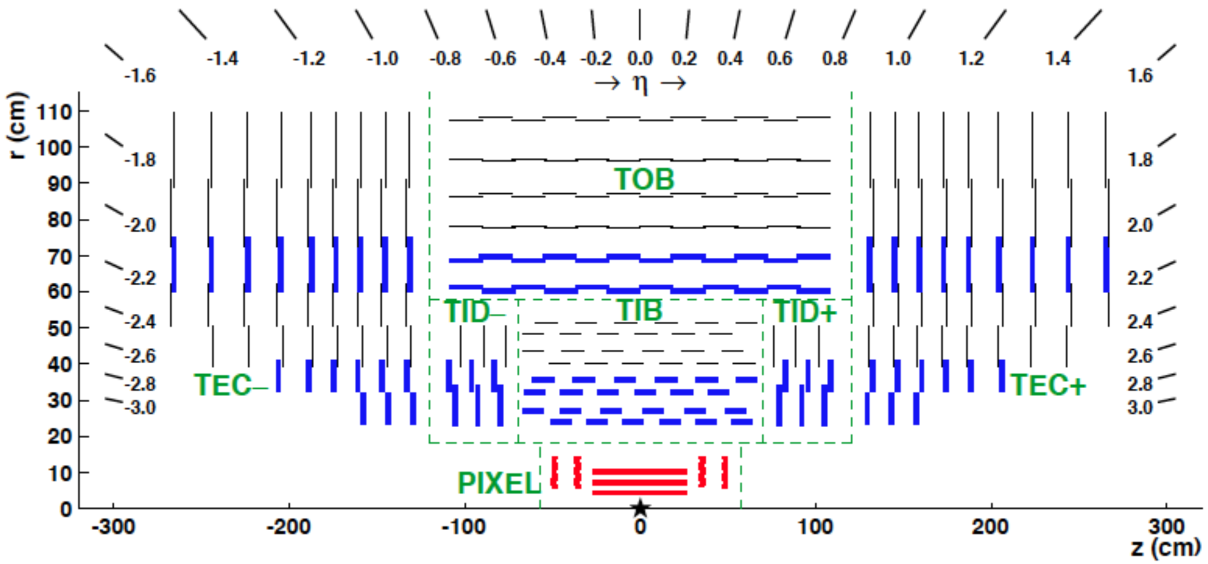
The Inner Detectors



The ATLAS ID

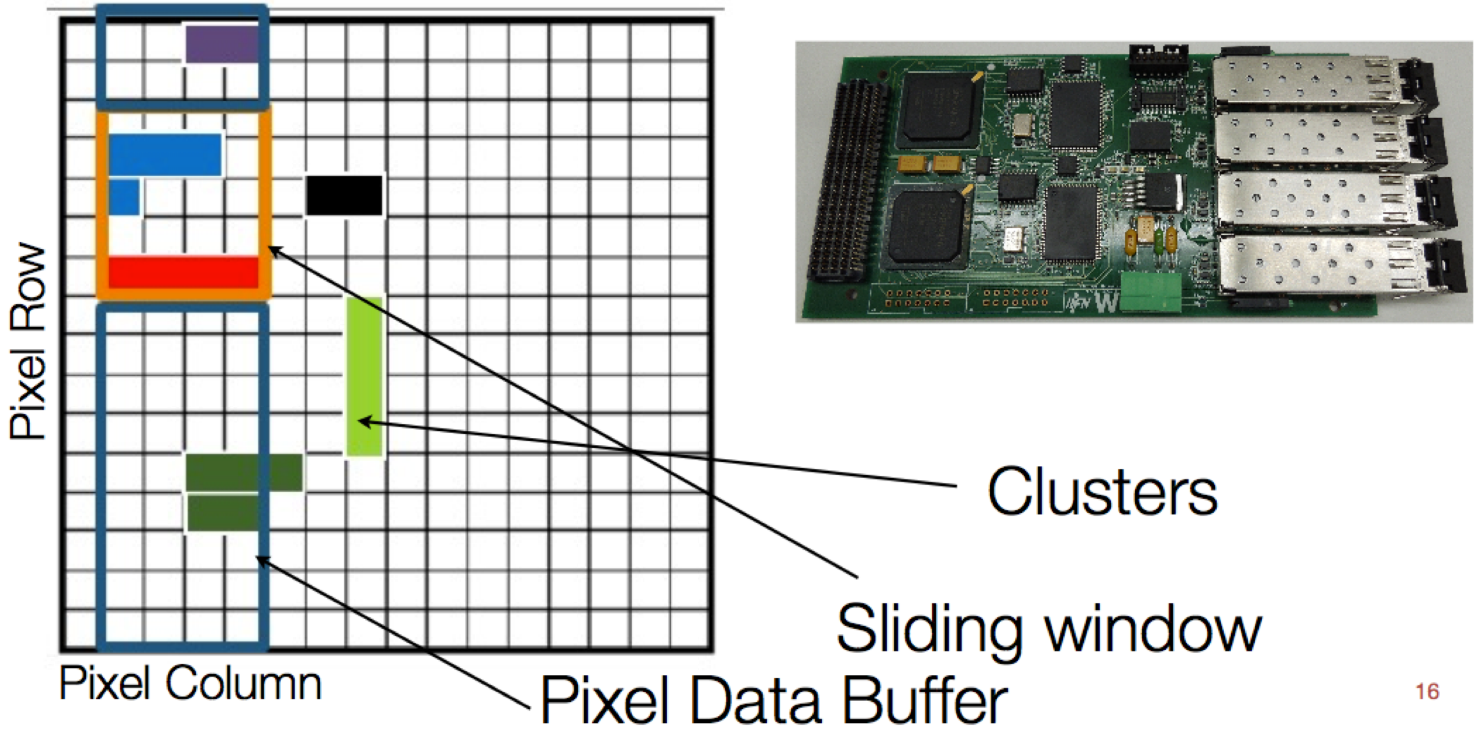


ATLAS ID material



The CMS silicon tracker

Space-point reconstruction



Clustering of pixel cells performed in hardware by the ATLAS Fast Track Trigger

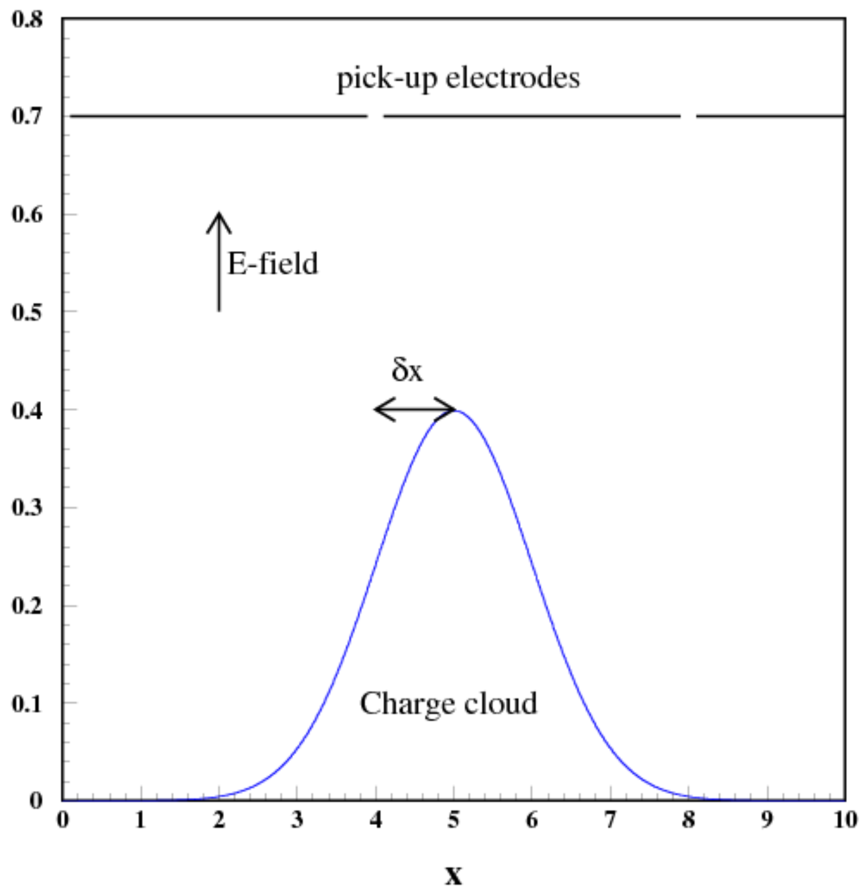
Spacepoint formation

- Tracking detectors register "hits" from signals induced on pickup electrodes by an *electron cloud* made by a track.
- In case of a hit on only one electrode, the precision is $\Delta / \sqrt{12}$ (Δ = the electrode size).
- Much better is it if the signal is distributed over two electrodes.

Then $$\frac{\delta x}{\Delta} = f(P1, P2, w)$$

giving higher accuracy, but you need to know both the pulse-heights P and the cloud width w.

Blum, Ronaldi: TPC tracking book



Spacepoint formation

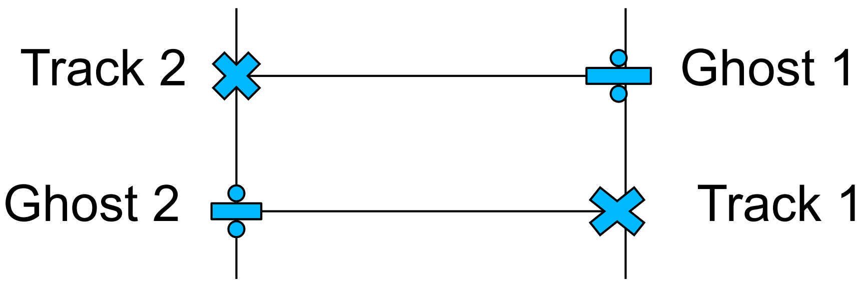
- In general, a track will give rise to a “cluster” of cell signals and their barycenter is a popular estimator of the position.
- The pulse-heights, P, must exceed a certain threshold and the electrodes must share a side, or at least a corner, forming a cluster. Summing over cluster cells, we get the barycenter:

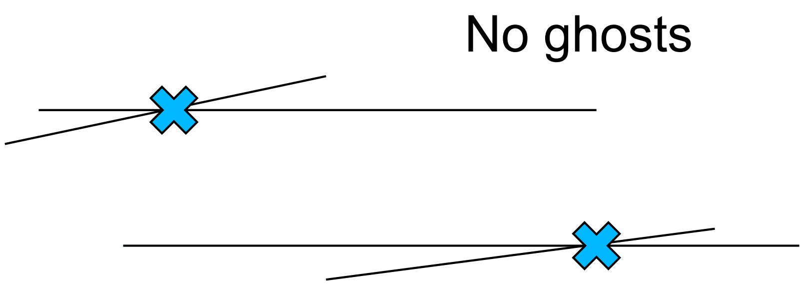
$$x = \sum P_i x_i / \sum P_i$$

(some use only the cells at the *cluster edge*, anyways you have to correct for finite cell size effects)

Stereo view

- If you do not have pixels, only wires or strips, what about the second coordinate?
- In strip detectors double sided wafers are often used with strips on both sides having an angle between them. But large angles gives ghost hits!
- At high track densities, 20-80 mrad is a good choice, avoiding too many ghost hits, having good resolution in the bending plane and still some resolution in the second coordinate.



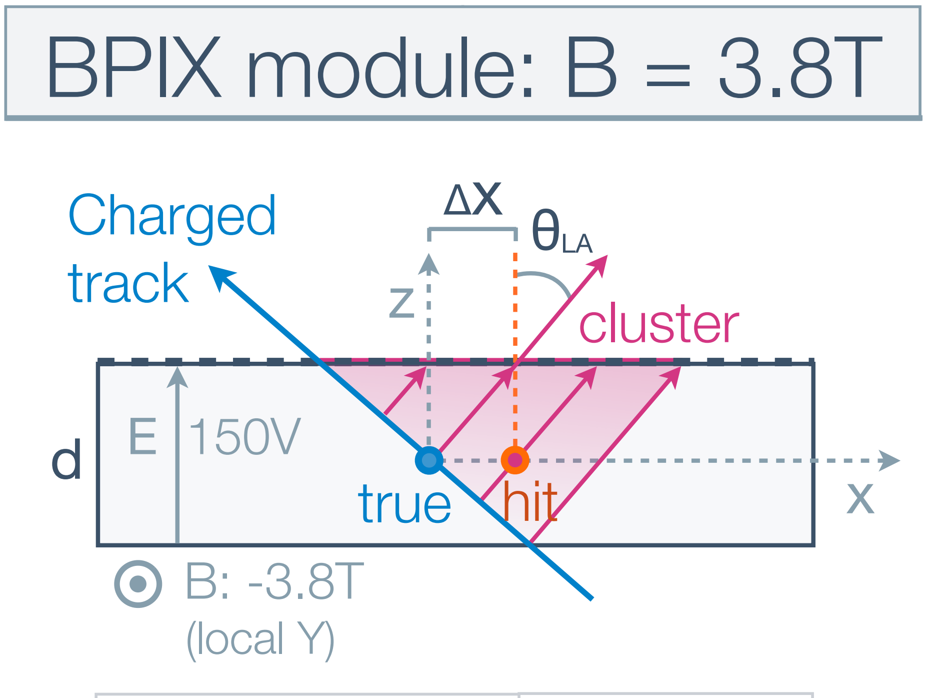


Spacepoint calibration

- In general we must know *the response function* , the probability distribution of induced pulse-heights for a given track impact
- (Actually, it would be lovely to know the inverse: the pdf for the track, given the pulse-heights. But we can not get that from test-beam..)
- The response function may vary from channel to channel and even vary in time. It must be *calibrated from data*.

Lorentz angle and defects

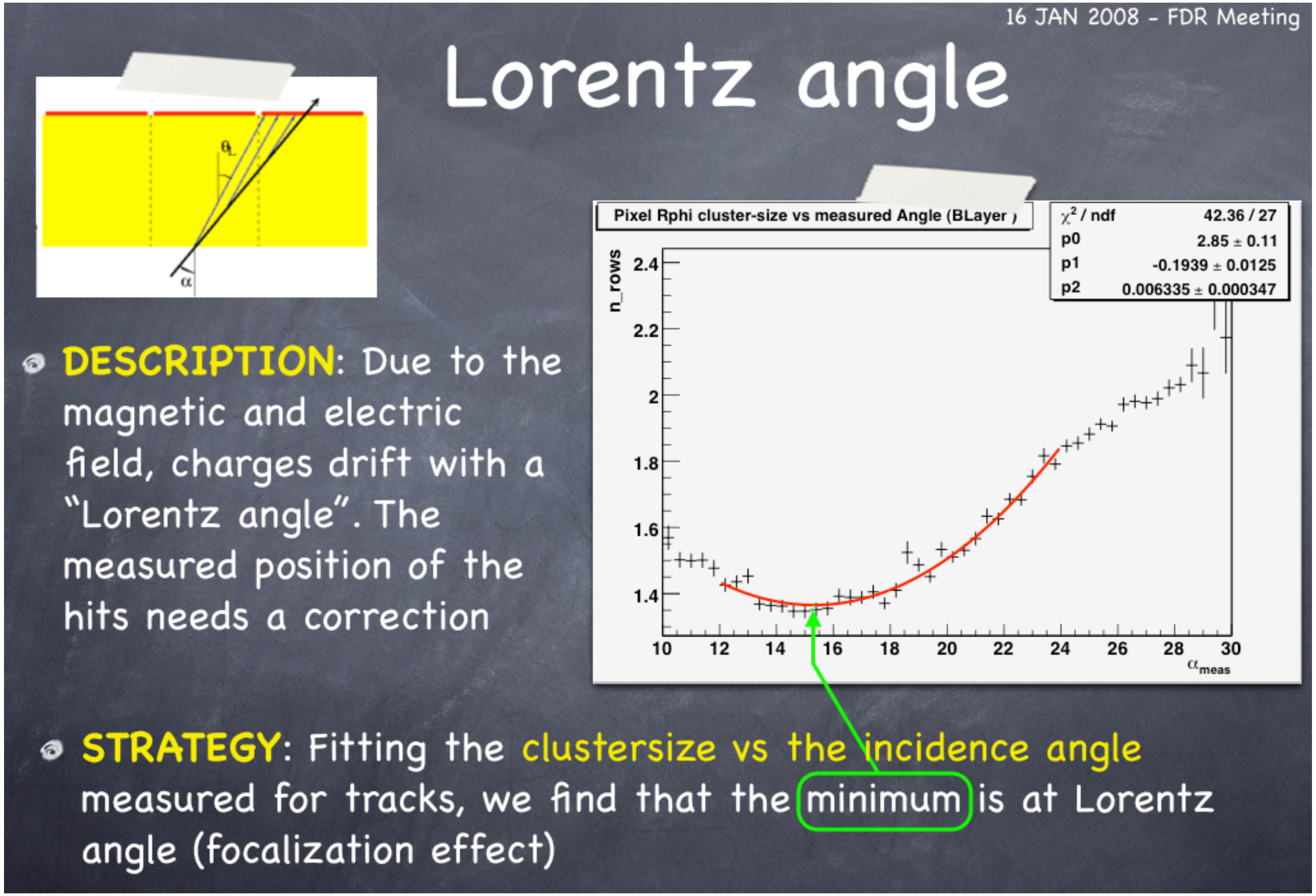
- Due to the Lorentz force from the B-field, the electron drift direction in silicon sensors is rotated by a Lorentz angle. This needs to be corrected for to get the true hit position.
- Another complication is the possibility of local radiation damage to pixels or strips biasing the barycenter.



- In CMS, all this is handled by comparing the observed charge distributions with a simulated template for a sample of possible true tracks, where defects are accounted for.

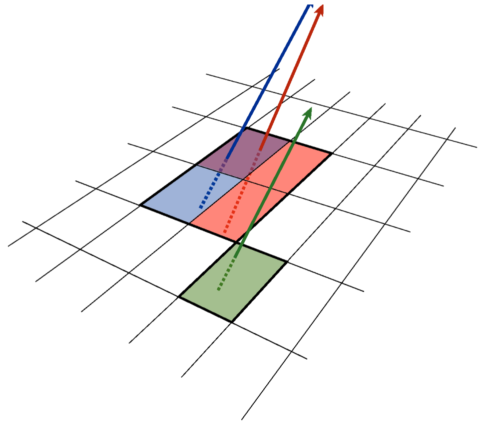
Lorentz angle

- In ATLAS, the Lorentz angle is extracted from the cluster size vs incident angle in the first tracking iteration. (from Simone Montesano)



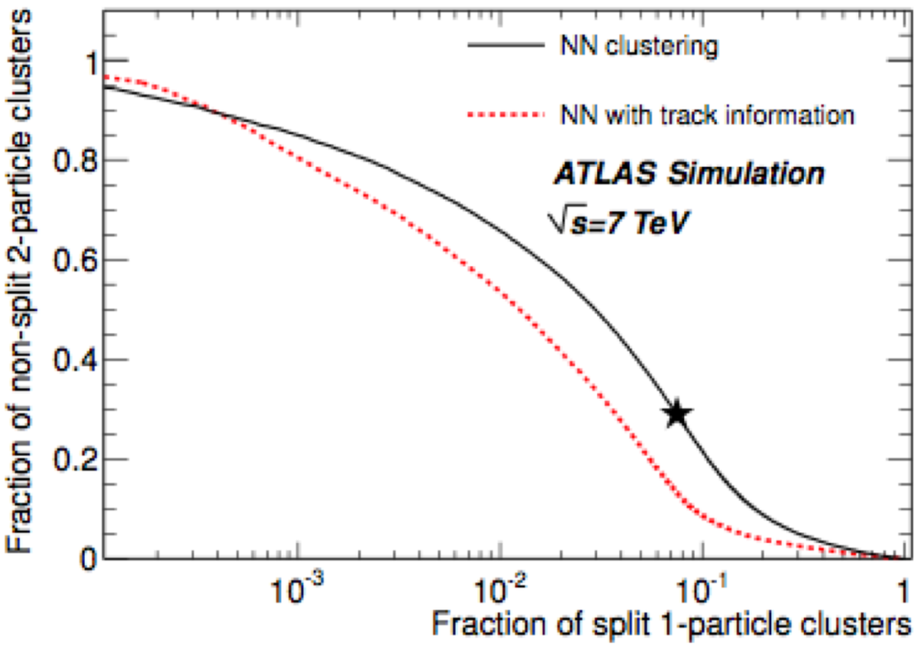
Splitting of merged clusters

- At high track densities, clusters of fired detector cells from two different tracks may merge.



- For example, a jet with p_T=1TeV has typically only 0.1mm between two tracks at the innermost ATLAS pixel layer.
- ATLAS uses an NN algorithm to split pixel clusters again (Prokofieff and Selbach 2012)
- Uses charge, shape, previous layer, incident angle
- The cell charge in pixel detectors is estimated using Time-over-Threshold.

Splitting merged clusters



Improvements in Run2:

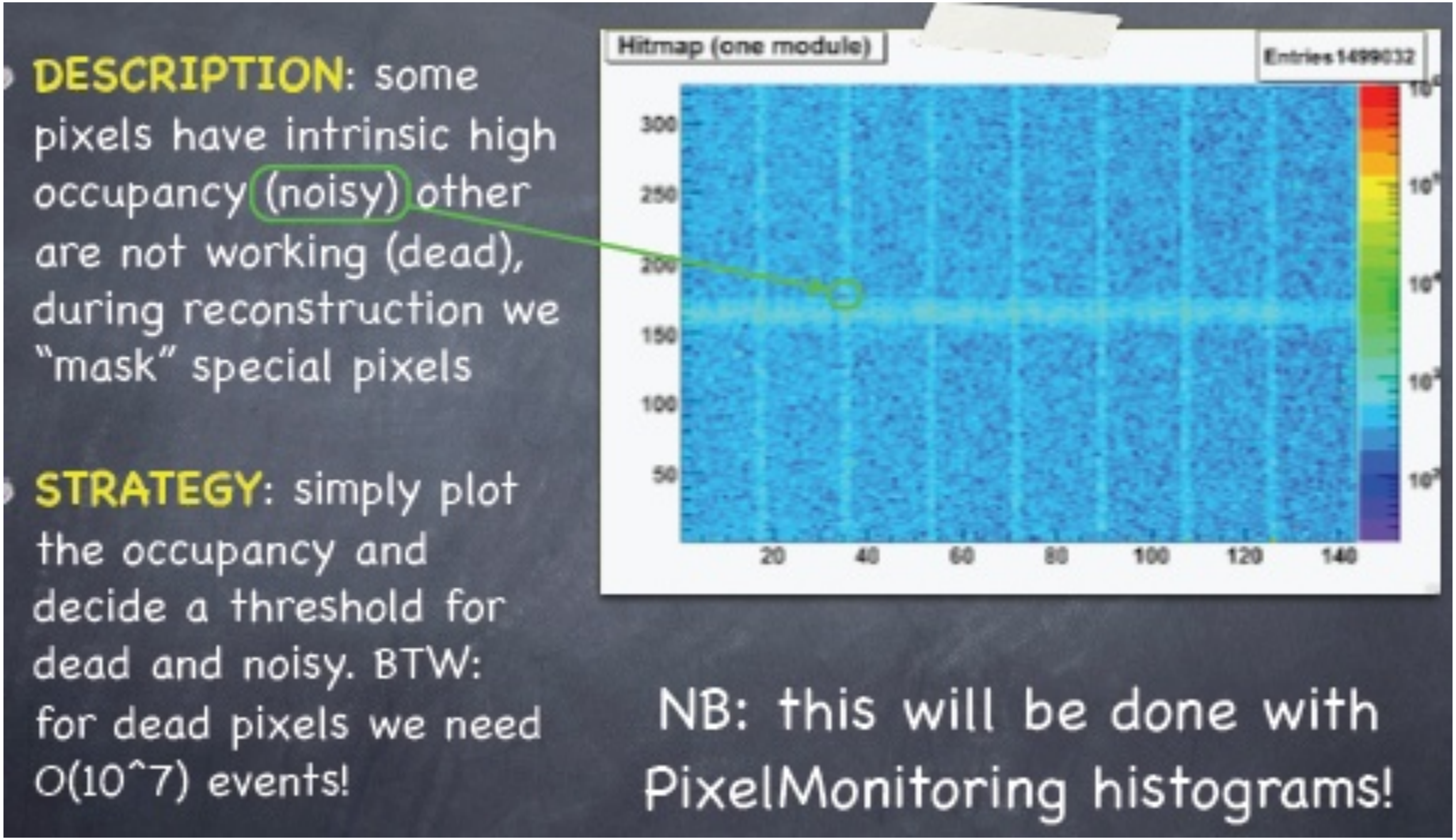
- A NN evaluates if a pixel cluster is *shareable.*
- Such can be shared without penalty (see later on “score”)
- Clusters are first split *after* Pass 1 track reco, taking advantage of track info.

This yields:

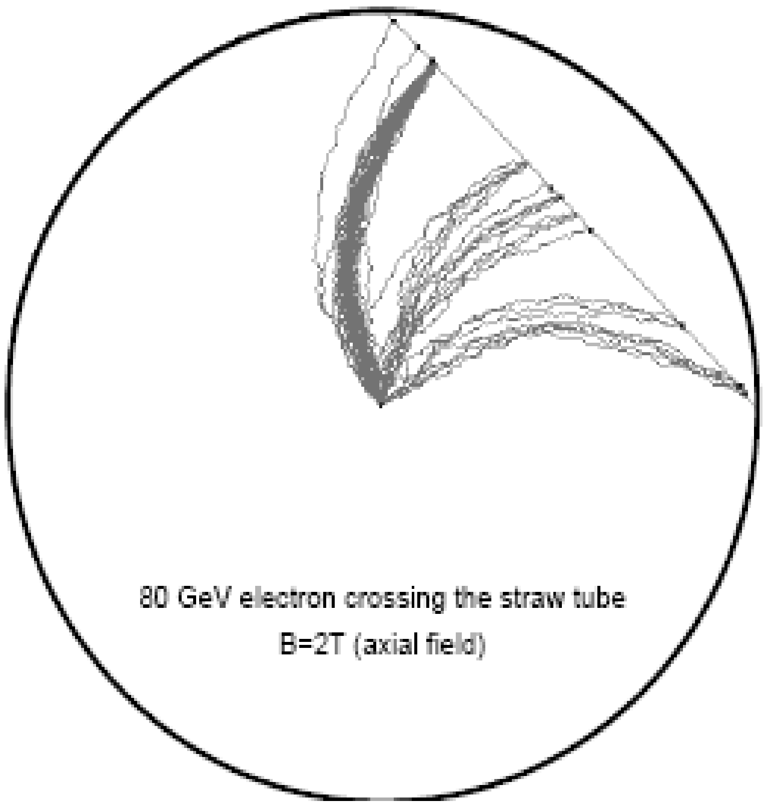
- a 10-17% improvement in track reconstruction efficiency in jet cores,
- a 7-13% increase in b-tagging efficiency
- a significant reduction of CPU (factor 4 when joined by other improvements in Run2 reco).

Dead and noisy channels

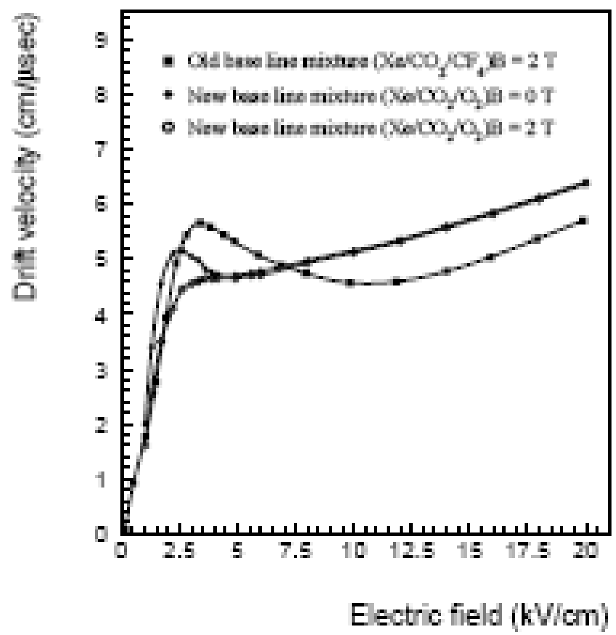
- Any clustering algorithm must handle dead or noisy channels to avoid false clusters.



Spacepoints in drift-tubes

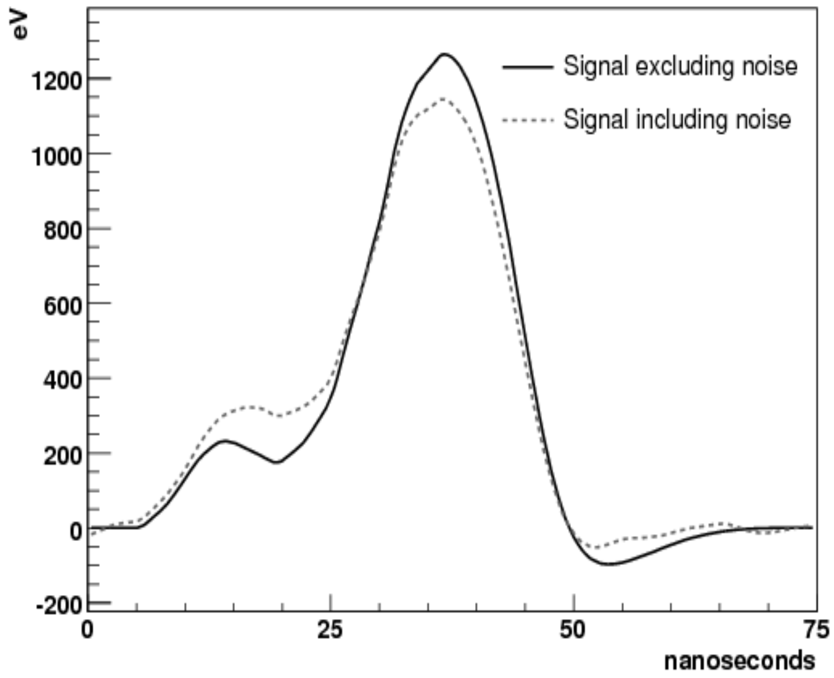


The ATLAS TRT flags time-bin t where the signal exceeds some threshold. Must calibrate the distance R(t-t0) from the track to the each wire.



ATLAS TRT

Refinements of drift radii



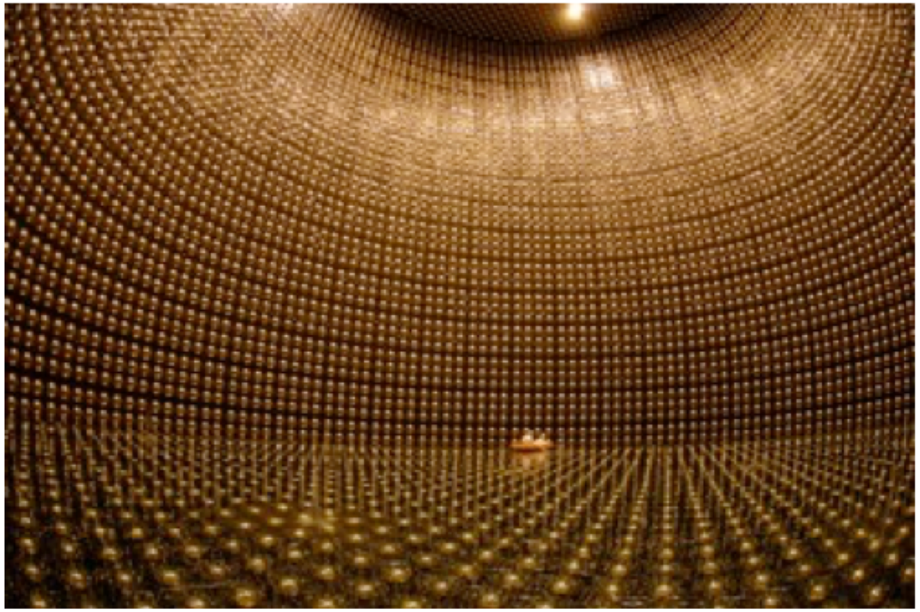
ATLAS TRT

Large pulses will trigger the threshold sooner for the same track impact -> small correction for large time-over-threshold or High-Threshold hit.

At a track refit, the track impact along the wire, angle and other info is available.

Small corrections for time-of-flight, signal propagation and other effects can be made at this point.

Big water Cherenkov detectors



The "*space-points*" are here
the signals on each PMT:
the charges on the anode
and their arrival times.

For calibration Super-K needs
for each single PMT:

1) The gain = charge / photo-electron.

2) The quantum and collection efficiency.

3) The timing calibration and resolution.

4) The background level.

In addition it needs the water transparency, temperature, the geo-magnetic field etc at each point in space.

A variety of light sources, radioactive sources and even small linear accelerators are used in the calibration.

From space-points to tracks

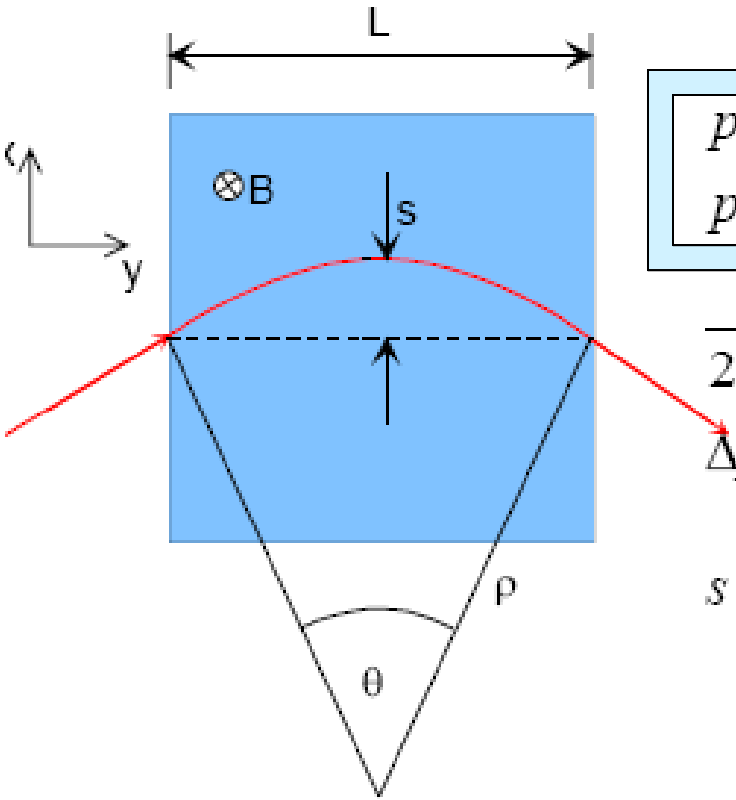
- Given a collection of space-points we need to group together those space-points that belong to a track and determine the tracks features.

- The important feature of a track is its momentum, so we open a parenthesis on momentum measurement

- Then we will look at tracking at trigger level

- Then study two track fit algorithms:
  the Kalman filter and the global chi-squared fit.
  Adaptive global methods are also discussed.

Momentum measurement (..



$$p_T = qB\rho$$

$$p_T\ (\mathrm{GeV/c}) = 0.3B\rho \quad (\mathrm{T\cdot m})$$

$$\frac{L}{2\rho} = \sin\theta/2 \approx \theta/2 \quad \rightarrow \quad \theta \approx \frac{0.3L\cdot B}{p_T}$$

$$\Delta p_T = p_T \sin\theta \approx 0.3L\cdot B$$

$$s = \rho(1-\cos\theta/2) \approx \rho\frac{\theta^2}{8} \approx \frac{0.3}{8}\frac{L^2 B}{p_T}$$

This and next three slides are from Christian Jorams summer student lectures

Momentum accuracy

the sagitta s is determined by 3 measurements with error $\sigma(x)$:

$$s = x_2 - \frac{x_1 + x_3}{2}$$

$$\left.\frac{\sigma(p_T)}{p_T}\right|^{meas.} = \frac{\sigma(s)}{s} = \frac{\sqrt{\frac{3}{2}}\sigma(x)}{s} = \frac{\sqrt{\frac{3}{2}}\sigma(x)\cdot 8 p_T}{0.3\cdot BL^2}$$

for N equidistant measurements, one obtains

(R.L. Gluckstern, NIM 24 (1963) 381)

$$\left.\frac{\sigma(p_T)}{p_T}\right|^{meas.} = \frac{\sigma(x)\cdot p_T}{0.3\cdot BL^2}\sqrt{720/(N+4)}$$ (for $N \geq \approx 10$)

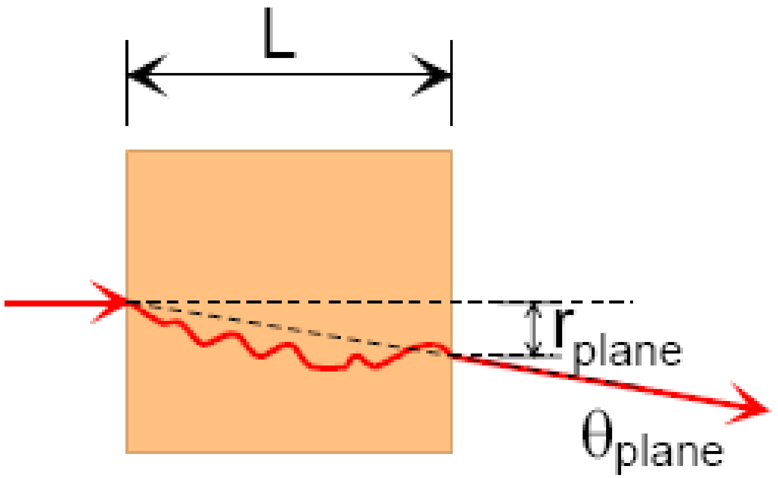
ex: p_T=1 GeV/c, L=1m, B=1T, $\sigma(x)$=200μm, N=10

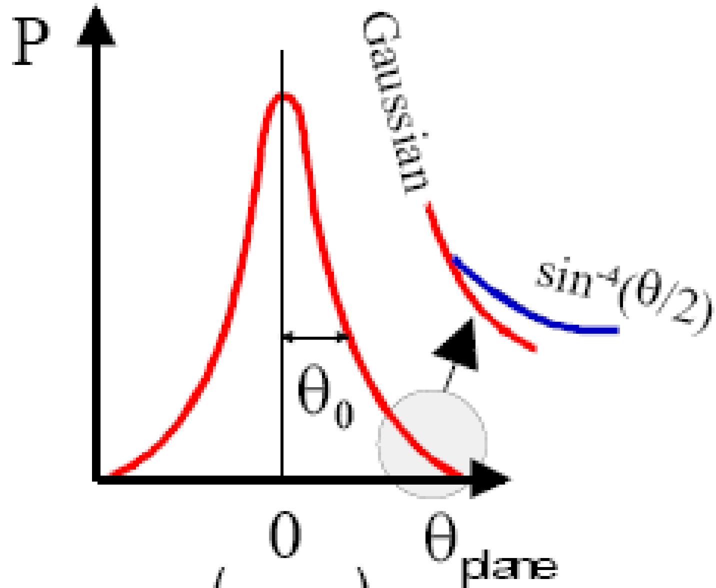
$$\left.\frac{\sigma(p_T)}{p_T}\right|^{meas.} \approx 0.5\%$$ ($s \approx 3.75$ cm)

Multiple scattering

Sufficiently thick material layer

→ the particle will undergo multiple scattering.





$$\theta_0 = \theta_{plane}^{RMS} = \sqrt{\langle \theta_{plane}{}^2 \rangle} = \frac{1}{\sqrt{2}} \theta_{space}^{RMS}$$

$$P(\theta_{plane}) = \frac{1}{\sqrt{2\pi}\theta_0} \exp\left\{ -\frac{\theta_{plane}{}^2}{2\theta_0^2} \right\}$$

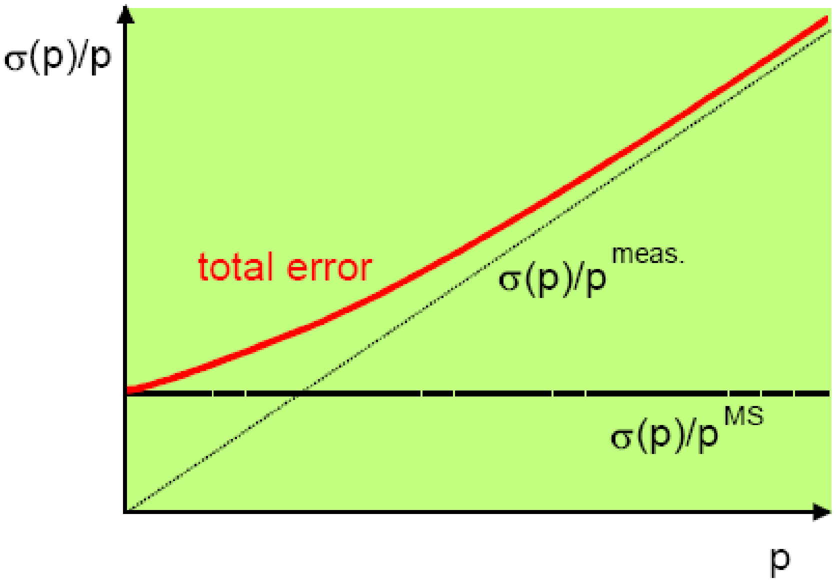
Total momentum error ..)

contribution from multiple scattering

$$\Delta p^{MS} = p \sin\theta_0 \approx p \cdot 0.0136 \frac{1}{p}\sqrt{\frac{L}{X_0}}$$

$$\left.\frac{\sigma(p)}{p_T}\right|^{MS} = \frac{\Delta p^{MS}}{\Delta p_T} = \frac{0.0136\sqrt{\frac{L}{X_0}}}{0.3BL} = 0.045\frac{1}{B\sqrt{LX_0}}$$

independent of p !



Fast pattern recognition

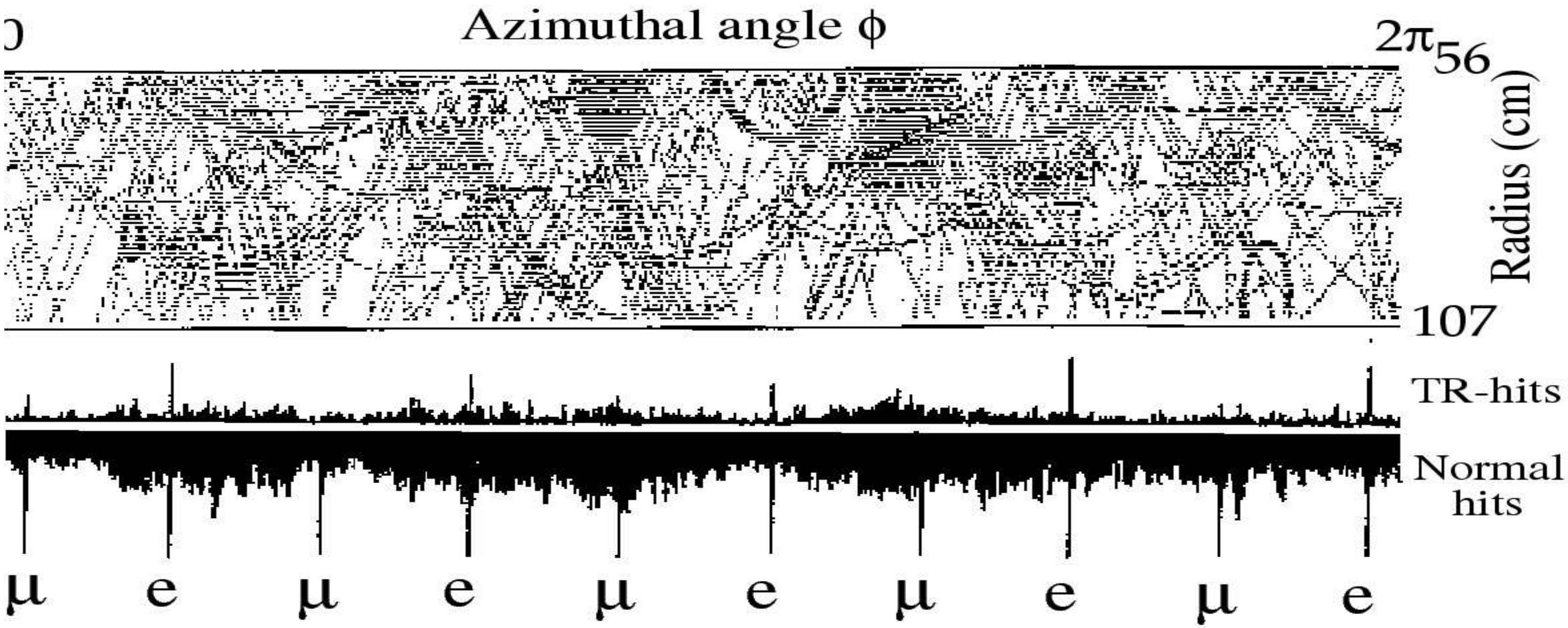
- How to associate a subset of hits to a track at trigger level?

- *Predefined templates*, i.e. patterns of fired cells defining an allowed track. Used in fast trigger algorithms.
- The *cell tower* is an example from the calorimeter world.

- *Hough transform* is another method. For straight tracks in two dimensions, each hit corresponds to a straight line in the slope-intercept plane. Peaks in this plane where many lines intercept reveal the hits-on-tracks.

Simple Hough transform

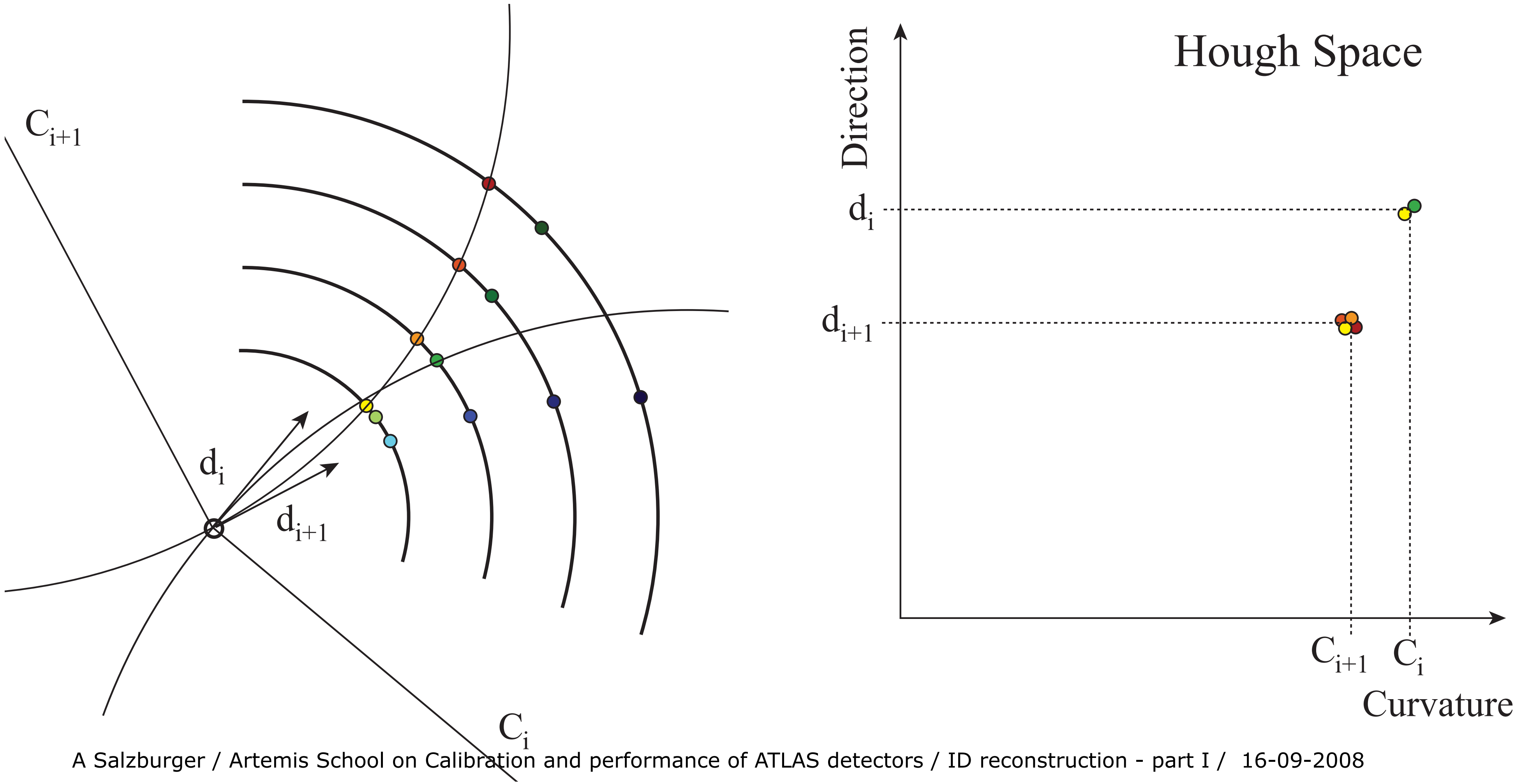
- Histogram methods may provide fast seeds for high momentum tracks – here an example from the ATLAS TRT:



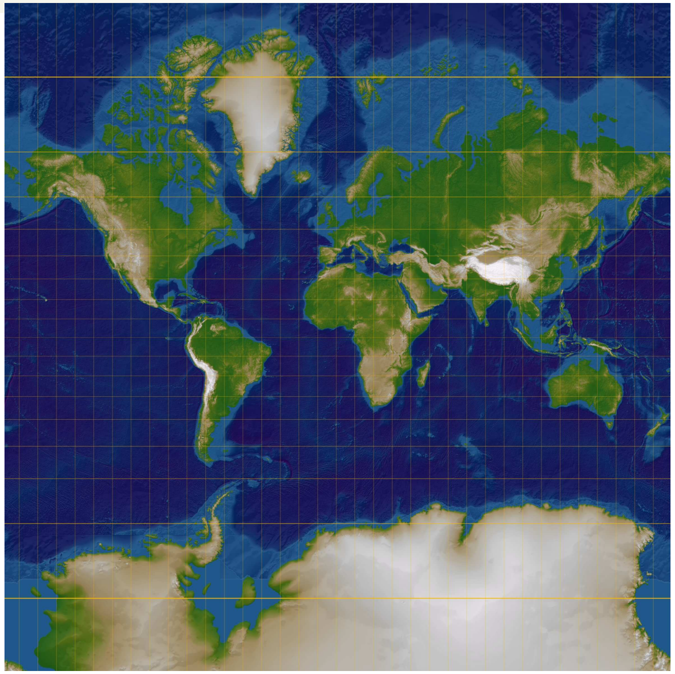
ATLAS TRT

General Hough transform

- Scan in two dimensions (d_i=-$C_i r$+d_{hit})
- Count number of compatible hits.



Conformal mapping

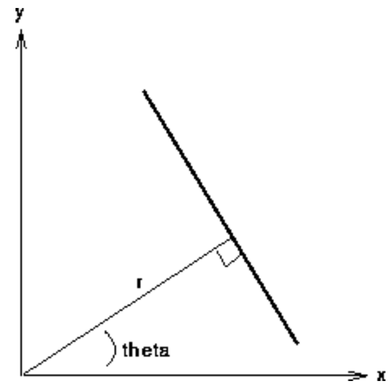


- Preserves angles, but not lengths.
- The points:

$$u=x/(x^2+y^2)$$
$$v=y/(x^2+y^2)$$

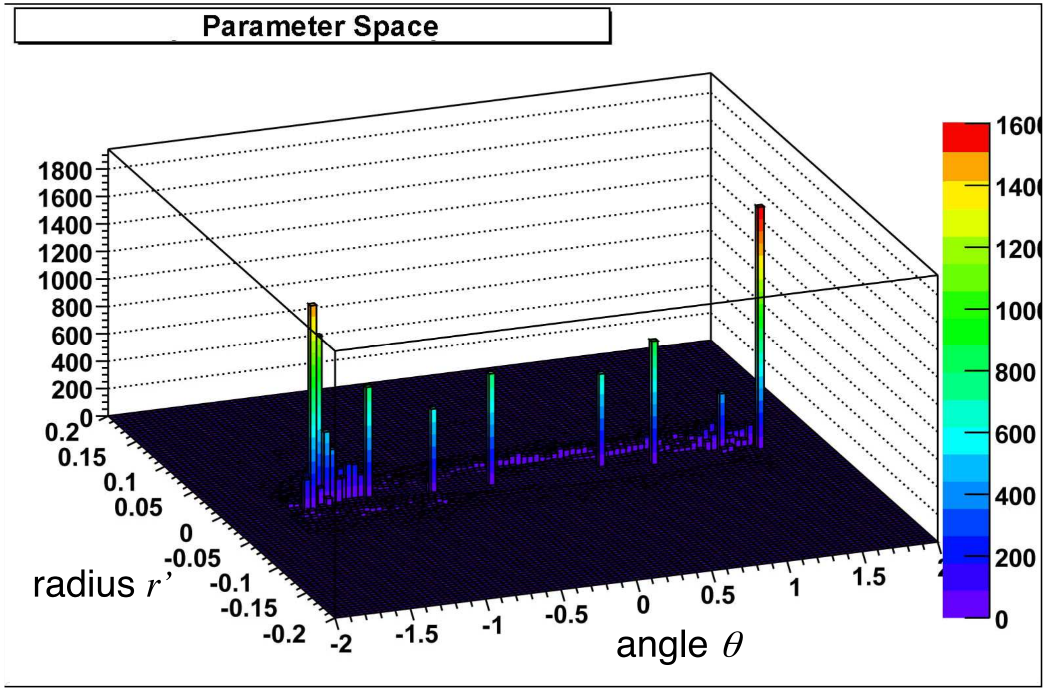
will lie on a straight line in (u,v) space for hits on a circle passing through (x,y)=(0,0).

- A straight line can be characterized by its point closest to the origin, with distance *d* ("DOCA") and direction θ.



Conformal mapping

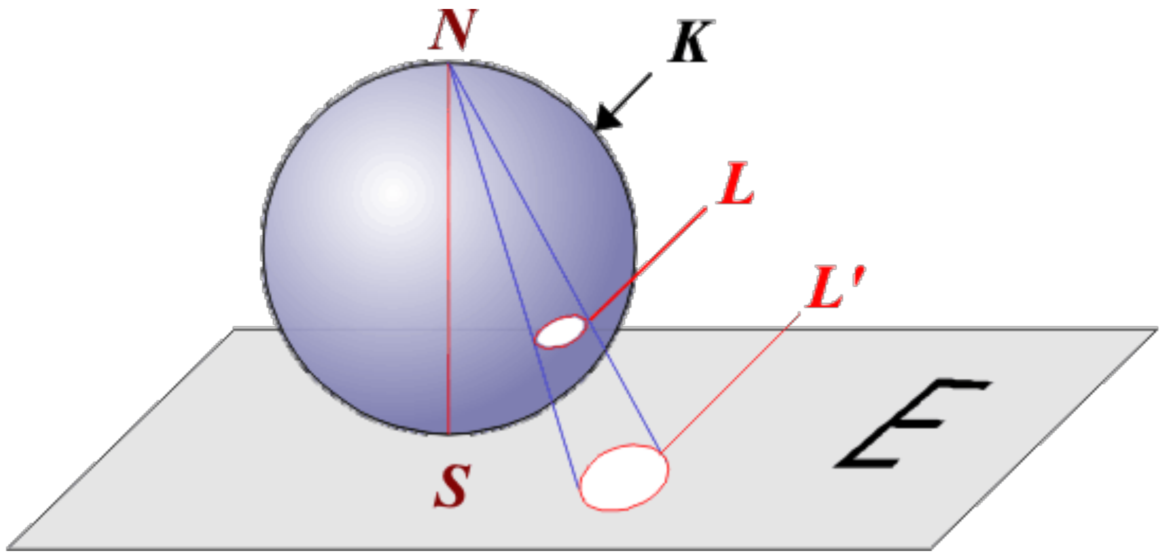
- For each hit (u,v), all d and θ of lines passing through the hit are entered in a histogram and the local peaks are found.
- From these we immediately get the circle parameters.



- Used by BES III, Belle II and PANDA.
- Even implemented on FPGA.

Riemann track fit

- Another fast non-iterative circle fit is the Riemann fit where a circle in a 2-D plane is transformed into a plane in 3-D that intersects the Riemann sphere

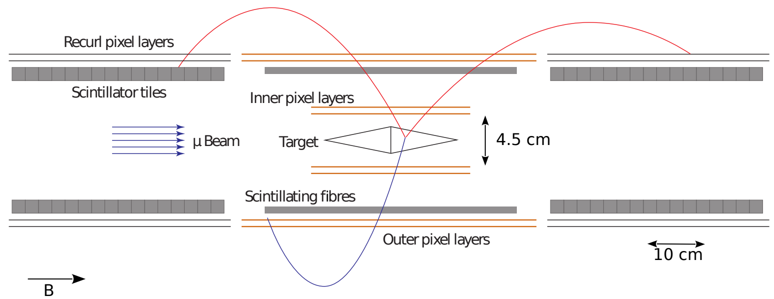


- The parameters of the plane through *L* are quickly found as a linear combination of the hit coordinates, and these can then be mapped to the circle parameters of *L'*.

R. Frühwirth and A.Strandlie, J.Phys.Cond 762(2016)012032 arXiv:1606.048802

Multiple scattering fit

- In many experiments, like mu3e, MS dominates resolution:

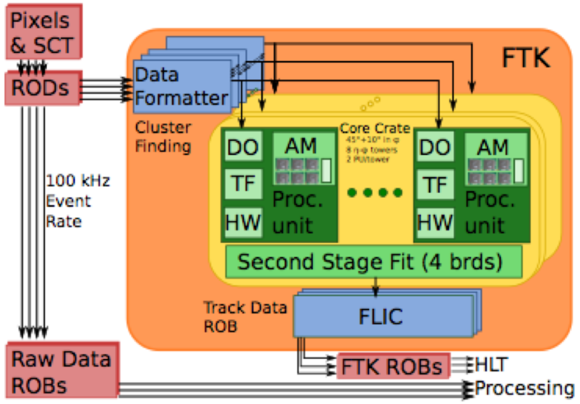


- A triplet of hits provides 6 constraints, but a helix with MS in the middle plane is described by 8 parameters (6 helix parameters plus two MS angle projections)
- The missing constraints can be supplied by minimizing:

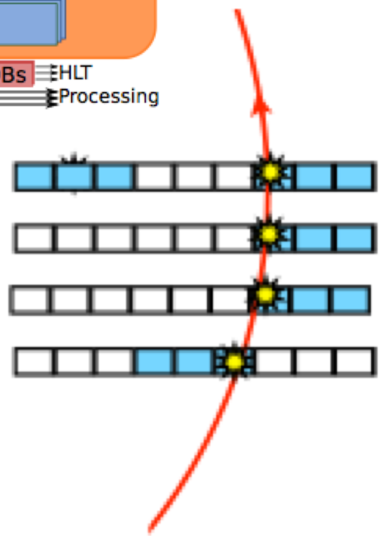
$$\chi^2(R_{3D}) = \frac{\Phi_{MS}(R_{3D})^2}{\sigma_\varphi^2} + \frac{\Theta_{MS}(R_{3D})^2}{\sigma_\theta^2}$$

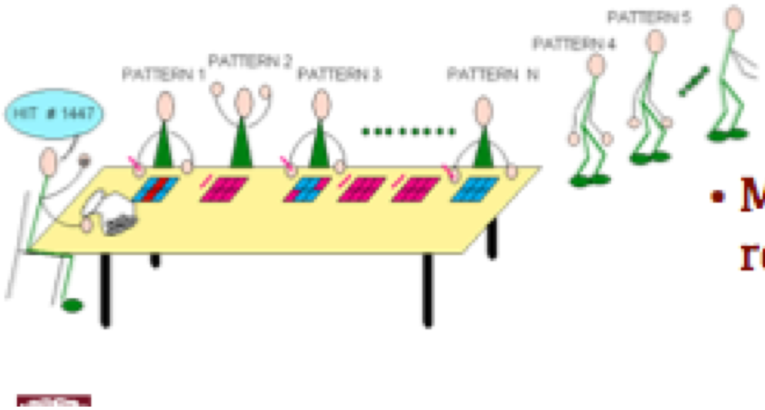
where R_{3D} is the helix radius and the two angles are the azimuthal and polar scattering angles, respectively. This minimalization leads to a fast online estimate of R_{3D} .

Associative memory – ATLAS FTK



The *associative memory* where each hit is seen by all possible templates is the most advanced example.





- Hits are ganged into Super Strips (SS)
  - Roughly 15x36 pixels/16strips per SS @ 70 int/x-ing
- **Custom associative memory chips** are used to compare hits to $O(10^9)$ patterns **simultaneously**
  - Pattern matching finished as soon as all hits are read
- Matched patterns (Roads) are then fit to reject bad roads
  - Most matches are fake, need fits to reduce bad rate

The state vector

- Let at each detector surface the track be given by a vector (position, direction, 1/p), along with its uncertainties:

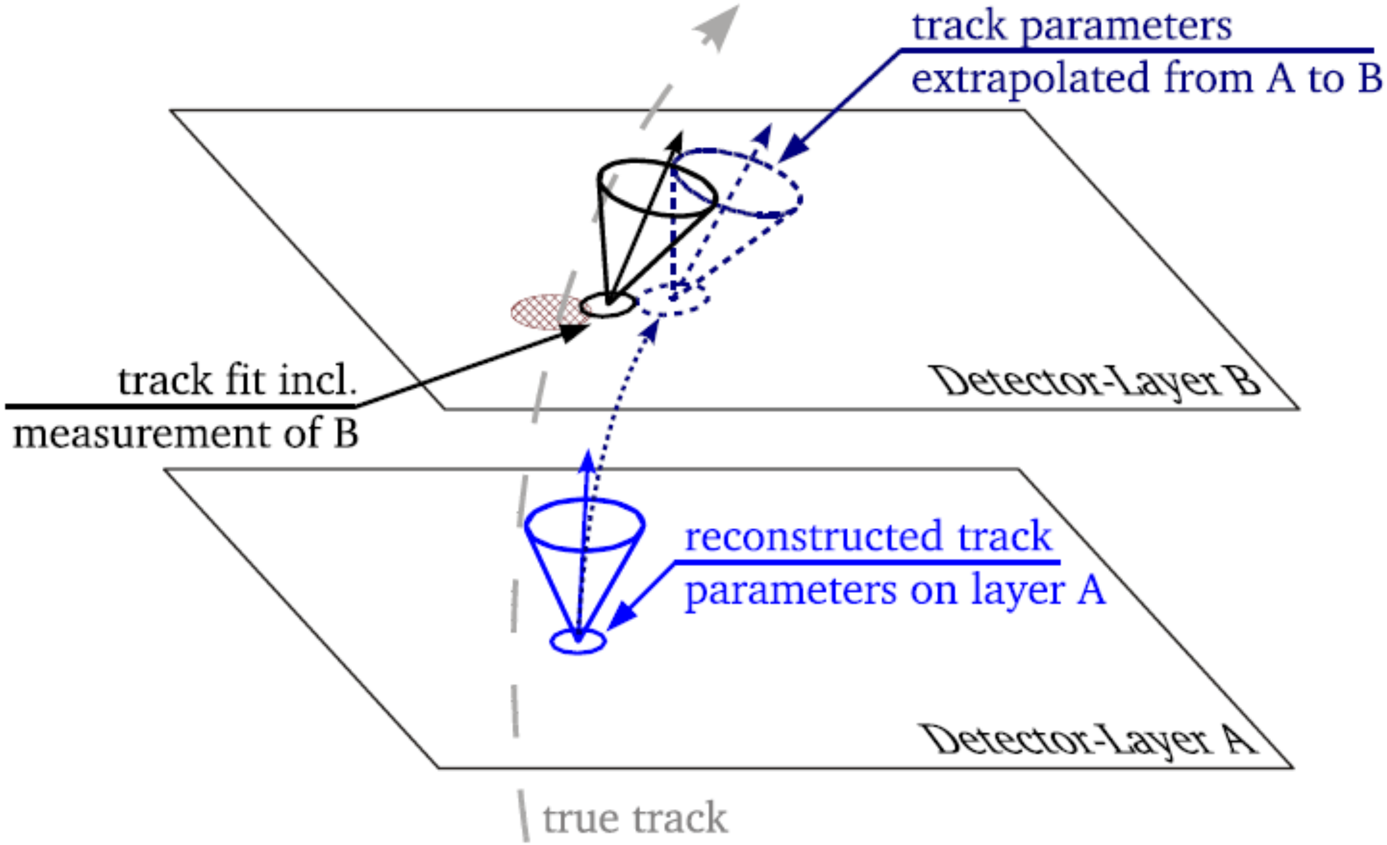


Figure from ATLAS reconstruction group

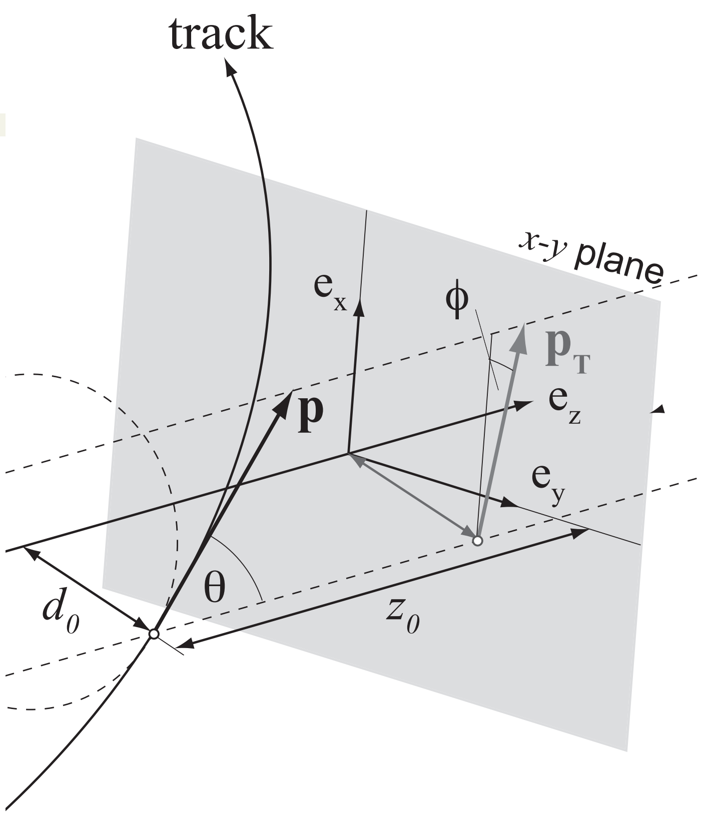
Helix parametrization

- An example of a *state vector* is *helix parameters*, where 90°-λ is the track angle to the B field, R is the radius, s the path length and h is a sign. This gives the trajectory:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x^0 + R\cdot(\cos(\alpha_0 + \frac{hs}{R}\cdot\cos\lambda) - \cos\alpha_0) \\ y^0 + R\cdot(\sin(\alpha_0 + \frac{hs}{R}\cdot\cos\lambda) - \sin\alpha_0) \\ z^0 + s\cdot\sin\lambda \end{pmatrix}$$

- A track in a detector with cylinder symmetry is a collection of helices at each “cylinder surface”.

Perigee parameters



- The perigee parameters

$$x = (\varphi_0, d_0, z_0, \theta, \frac{q}{p})$$

are often used to describe the track state at the closest approach to the beam (z) axis.

- q/p is measured with approximately gaussian uncertainty.
- d_0 has a sign convention following that of the angular momentum of the track wrt the z axis

The projection matrix H

- To compare with measurements m, the track state x needs to be mapped onto "measurement space". We linearize:
- H=δm/δx, where H is the projection matrix

(assuming for simplicity that x=0 corresponds to m=0)

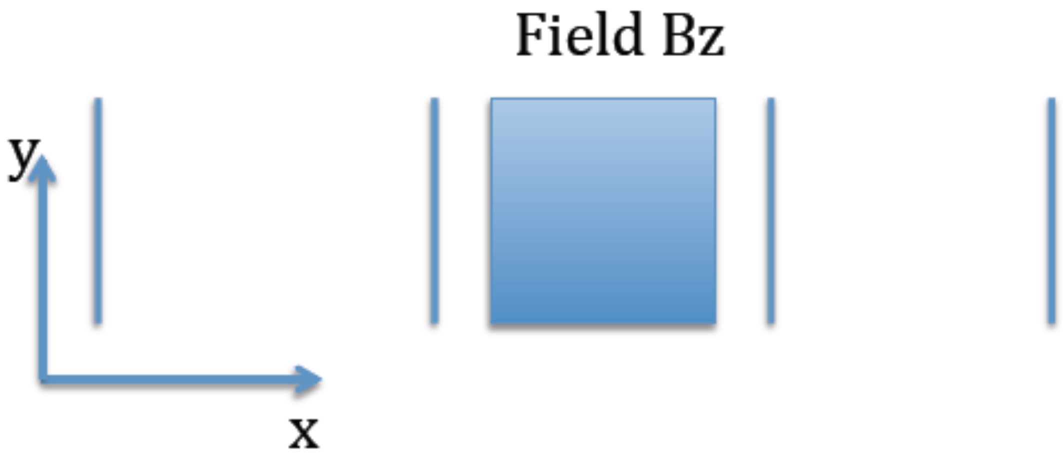
- Let a set of strips form a small angle α with the x axis. The track parameters are x and y at each plane.

$$H\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -\sin\alpha & \cos\alpha \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}$$

produces the y'-coordinate perpendicular to the tilted strip. This yields immediately the raw hit strip number m.

Spectrometer example

4 y-z pixel planes distance between planes d



$$\tan(\Delta\varphi) = 0.3 \times \int B_z dx \times q/p = b(q/p)$$

Using units of Tesla,m, and GeV/c

Spectrometer example

Projection matrix (linearized)

measurements

State vector at plane 1

$$m = \begin{bmatrix} z_1 \\ \dots \\ z_4 \\ y_1 \\ \dots \\ y_4 \end{bmatrix}, \qquad x = \begin{bmatrix} z \\ z' \\ y \\ y' \\ q/p \end{bmatrix}, \qquad H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & d & 0 & 0 & 0 \\ 1 & 2d & 0 & 0 & 0 \\ 1 & 3d & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d & 0 \\ 0 & 0 & 1 & 2d & bd/2 \\ 0 & 0 & 1 & 3d & 3bd/2 \end{bmatrix}$$

The Kalman filter

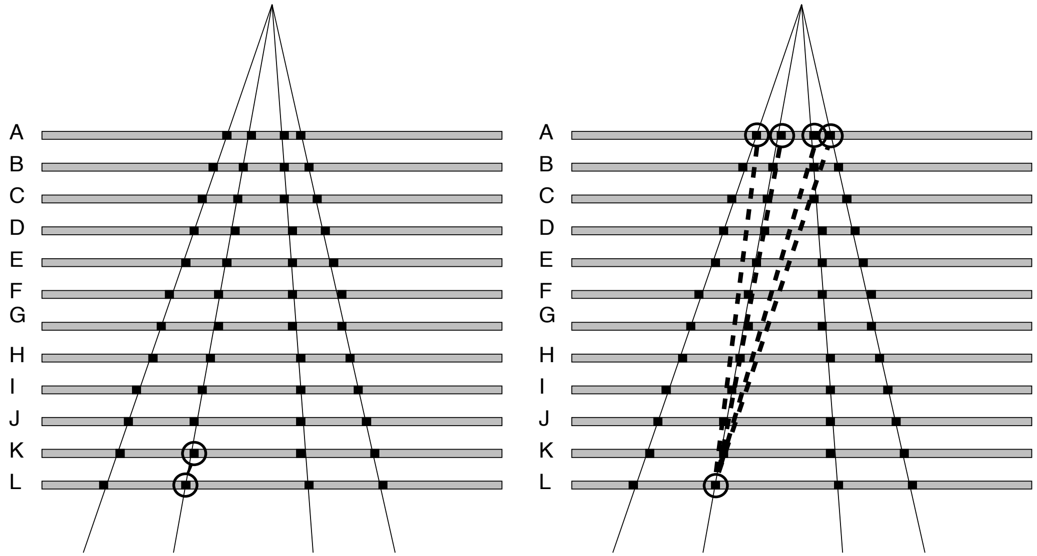
- Determines the track state vector dynamically from measurements at each detector surface.
- These are either discarded or used to update the existing state vector.
- Needs only inversion of small matrices. Fast.
- Can account for noise, multiple scattering and energy loss at each surface. Efficient.
- Is equivalent to the least squares fit, but provides pattern recognition integrated in the fit.

Seeding

- We need track seeds with a high efficiency and modest fake rate. Many possible strategies:

Figure from Mankel arXiv:0402039v1

Start from two outer layers:



Connect all inner layer hits to each outer

- ATLAS and CMS use the inner pixel layers for seeds and then proceed outwards for track finding

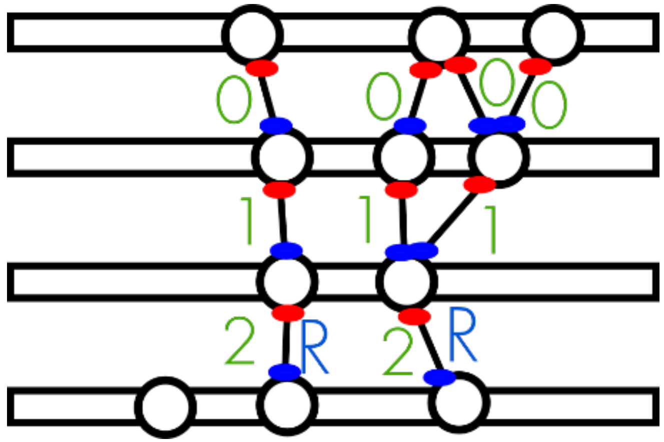
Cellular automaton seeding

Generic lib available!

https://github.com/HSF/TrickTrack

- CMS and Belle II have cellular automatons running in parallel GPU threads

- Cell: allowed doublet of hits.
- Cells start in state=0
- If there is an outer neighbour in the same state, then state++
- Continue until states do not change
- Select eg quadruples by chosing state=2 start cells



Great improvement in CMS fake rate:

1/100 wrt HLT 2016

-30% wrt triplet propagation method.

The propagator F

- Let the track transport from layer k-1 to k be given by

$$x_k = f(x_{k-1})$$

- Let the predicted state be denoted by a tilde. If f is not already linear in x, we Taylor expand it:

$$\tilde{x}_k = F_k x_{k-1},$$

$$C_k^{k-1} = F_k C_{k-1} F_k^T + Q_k$$

where C_k is the covariance matrix for the predicted state and Q contains the additional random perturbations in the step, such as multiple scattering and energy loss.

Covariance matrices V and C

- A pair of random variables x_i and x_j has the covariance matrix:

$$V_{ij} = E((x_i - E(x_i)) \times (x_j - E(x_j))$$

It is symmetric and have diagonal elements equal to the variances of the x'es.

Off-diagonal elements describe the degree of correlation between x_i and x_j.

- Any set of functions f_i of the x's has (to lowest order in a Taylor expansion) the covariance matrix:

$$C_{ij}^{f} = \sum_{kl} \frac{\partial f_i}{\partial x_k} \frac{\partial f_j}{\partial x_l} V_{kl}$$

- This is the chain rule.

The propagator F – simple example

- The F propagator is exactly the same as the transfer matrix of accelerator physics.
- For our example spectrometer we have the z projection propagation from the second to the third plane (a drift space in accelerator language).

$$\tilde{z}_3 = F_3 z_2 = \begin{bmatrix} 1 & d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_2 \\ z'_2 \end{bmatrix},$$

$$C_3 = F_3 C_2 F_3^T + Q.$$

$$C_2 = \begin{bmatrix} \sigma_m^2 & \sigma_m^2 / d \\ \sigma_m^2 / d & 2\sigma_m^2 / d^2 \end{bmatrix}, \qquad Q = \begin{bmatrix} \theta_{MS}^2 d^2 & \theta_{MS}^2 d \\ \theta_{MS}^2 d & \theta_{MS}^2 \end{bmatrix}$$

The propagator F – complex case

- In regions with an inhomogenous B field, the preferred method is Runge-Kutta integration. Here, the trajectory derivatives are sampled at a number of intermediate positions, weighted so that the error is 5th power in h, the small time-step to the next plane:

$$y' = f(t, y), \qquad y_0 = y_0(t_0)$$

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = f(t_n, y_n)$$

$$k_2 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1)$$

$$k_3 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2)$$

$$k_4 = f(t_n + h, y_n + hk_3)$$

The propagator F – complex case

- Let us try a grossly nonlinear case y=exp(t):

$$y' = y, \qquad y_0 = 1$$

$$k_1 = 1$$

$$k_2 = 1 + \frac{h}{2}$$

$$k_3 = 1 + \frac{h}{2}(1 + \frac{h}{2})$$

$$k_4 = 1 + h + \frac{h^2}{2}(1 + \frac{h}{2})$$

$$y_1 = 1 + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 1 + h + \frac{h^2}{2} + \frac{h^3}{6} + \frac{h^4}{24}$$

- Victory!

The residual r

- The difference between a measurement m and its prediction by the track state, Hx, is called the residual:

$$r_k^{k-1} = m_k - H_k \tilde{x}_k, \quad R_k^{k-1} = V_k + H_k C_k^{k-1} H_k^T$$

V is the *covariance matrix* of the measurements

R is the *covariance matrix* of the residuals.

(Note that the contribution from the track is here *added* to the measurement variance. The measurement is not used yet. If the hit contributes to the track, the track variance is instead *subtracted* from the residual variance).

At this point you can reject a measurement m_k on the basis of r_k^2/R_{kk} . This is the *pattern recognition* part of the KF.

Updating the state vector:

- A way to update the track state with the newly added hit is to take a weighted average of the predicted track state and the state suggested by the new measurement:

$$x_k = C_k((C_k^{k-1})^{-1} x_k^{k-1} + H^T V_k^{-1} m_k)$$

$$C_k^{-1} = (C_k^{k-1})^{-1} + H^T V_k^{-1} H$$

Updating the state vector:

- An equivalent way is to use the gain matrix K:

$$K_k = C_k^{k-1} H^T (V_k + H_k C_k^{k-1} H_k^T)^{-1}$$

$$\tilde{x}_k = \tilde{x}_k^{k-1} + K_k (m_k - H_k \tilde{x}_k^{k-1})$$

$$C_k = (1 - K_k H_k) C_k^{k-1}$$

Filtered residuals

- The filtered residual, its covariance and χ2 are

$$r_k = (1 - H_k K_k) r_k^{k-1}, \; R_k = (1 - H_k K_k) V_k, \; \chi^2 = r^T R^{-1} r$$

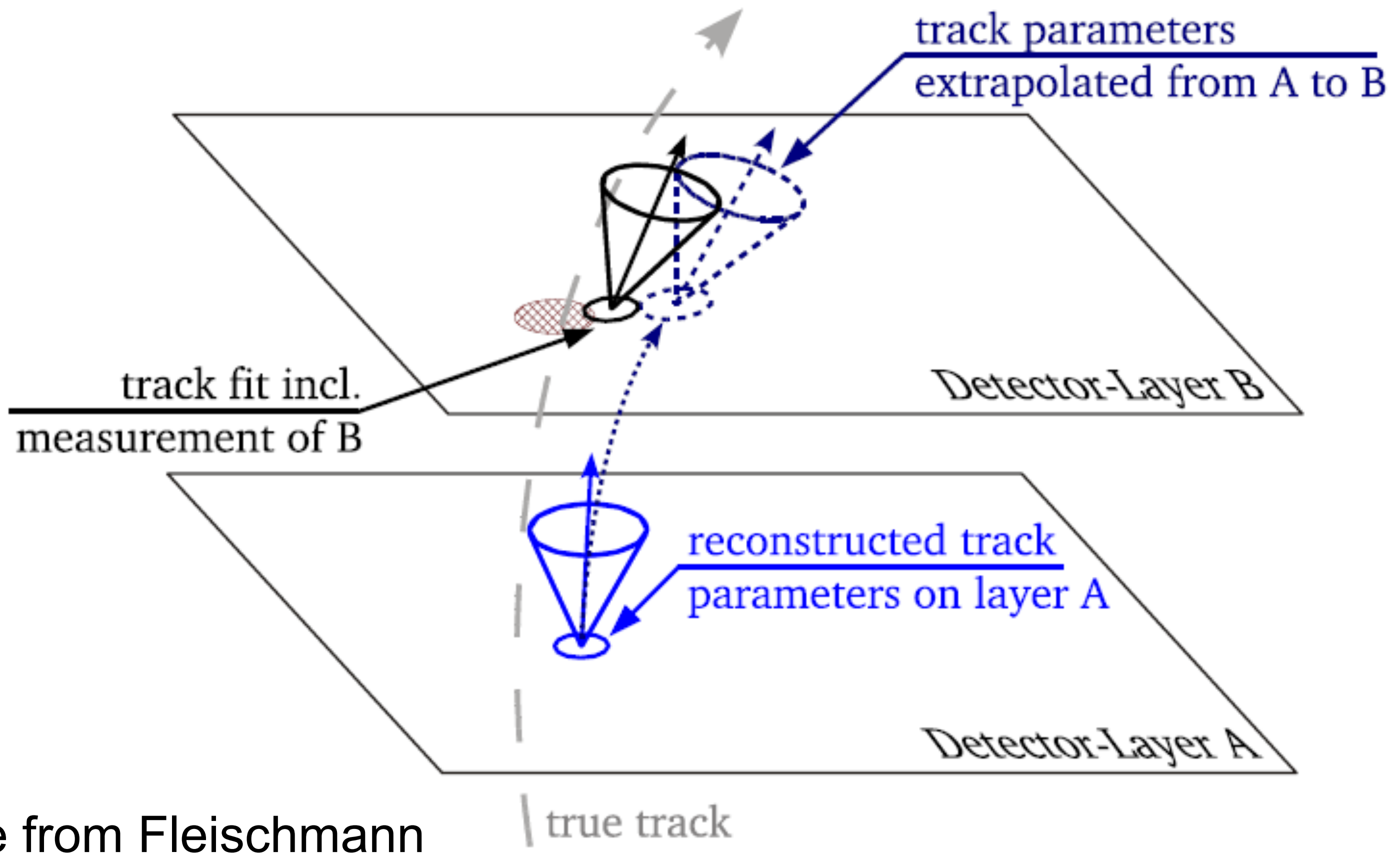


Figure from Fleischmann

Smoothing

- We have reached the end with n hits. Now the procedure is repeated backwards. This is used to update the state at each surface k with the information from all the other:

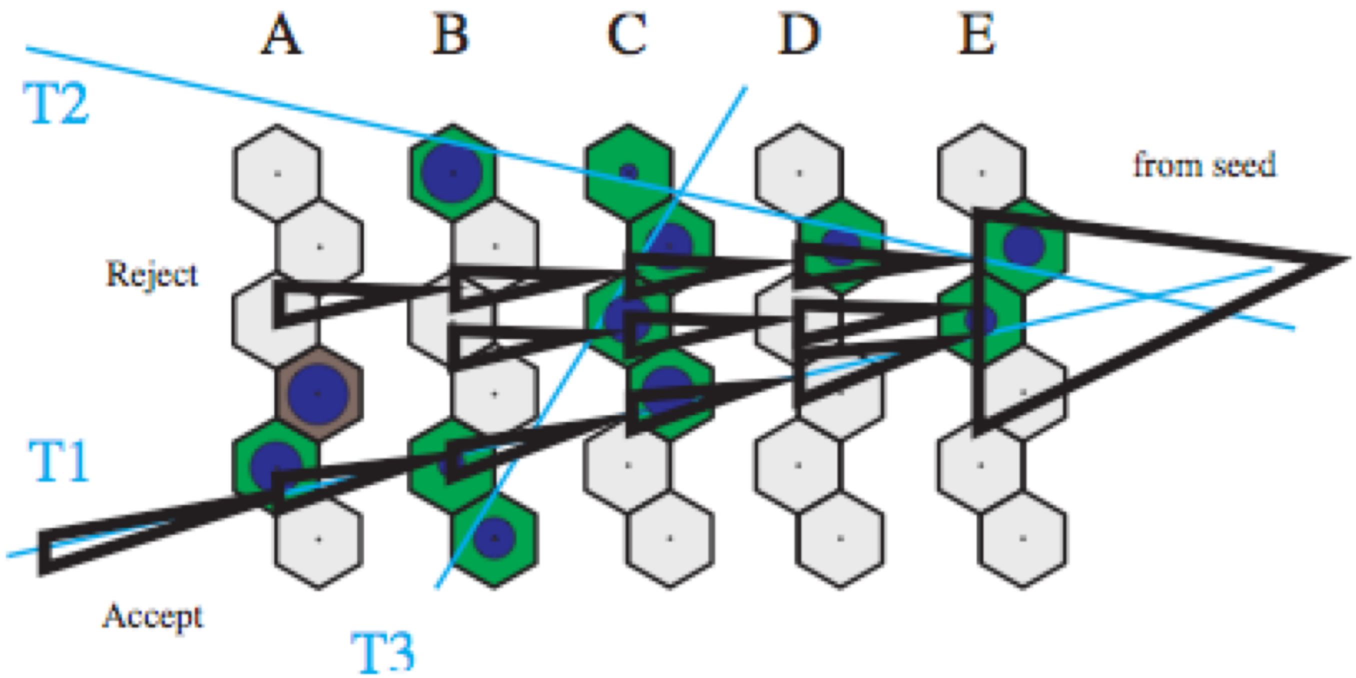
$$C_{k|n}^{-1} = C_{k|k}^{-1} + (C_{k|k+1}^{b})^{-1}$$

$$x_{k|n} = C_{k|n}(C_{k|k}^{-1} x_{k|k} + (C_{k|k+1}^{b})^{-1} x_{k|k+1})$$

- Finally the state at the innermost surface is extrapolated to the perigee, and this result is used in further analysis.

Combinatorial Kalman Filter

- For tracking in dense track environments, the nearest hit might not be the best.
- The Combinatorial Kalman Filter (Mankel 1997) keeps many options open for propagating a seed until one of them conquers



Outlier removal and iterations

- Several iterations are normally used:

- The smoother fit can use updated space-points (applying small corrections depending on track parameters).

- At the smoother step, outliers contributing a large χ2 can be removed (due to δ-electrons, nearby tracks or noise)

- The easiest tracks (high pT, many hits-on-track) are reconstructed first and their hits removed

- The remaining hits are now fitted with more relaxed requirements.

Finally repeat outside-in, using seeds in the outer layers, to pick up long-lived decays and photon conversions.

Global (Newton-Raphson) fits

- The global least-squares fit requires to know in advance which hits belong to the track.
- It minimizes the weighted sum of distances between the fitted track and the assigned hits, adjusting the track states at each surface.
- It is mathematically equivalent to the Kalman smoother for a fixed selection of hits on a track.
- If all measurement errors are Gaussian, it is also equal to the maximum likelihood fit.

Global fits

- In the approximation where the expected measurements are linear in the track parameters x, we minimize:

$$\chi^2 = (\overline{m} - H\overline{x})^T V^{-1} (\overline{m} - H\overline{x})$$

where m is a vector of measurements at *all the surfaces.*

- The solution is:

$$\overline{x} = (H^T V^{-1} H)^{-1} H^T V^{-1} \overline{m}$$

For normally distributed m, this is also the maximum likelihood estimate of the parameters.

The factor

$$(H^T V^{-1} H)^{-1} = \left(\frac{1}{2} \frac{\delta^2 \chi^2}{\delta \overline{x}^2}\right)^{-1}$$

is also the covariance matrix C of the track parameters.

Newton-Raphson fit

- If the projection h(x) is not linear, we can Taylor expand around an initial value x_0 obtaining approximately:

$$\frac{d\ \chi^2}{dx}(x_0) = 2H^T V^{-1}(m - h(x_0))$$

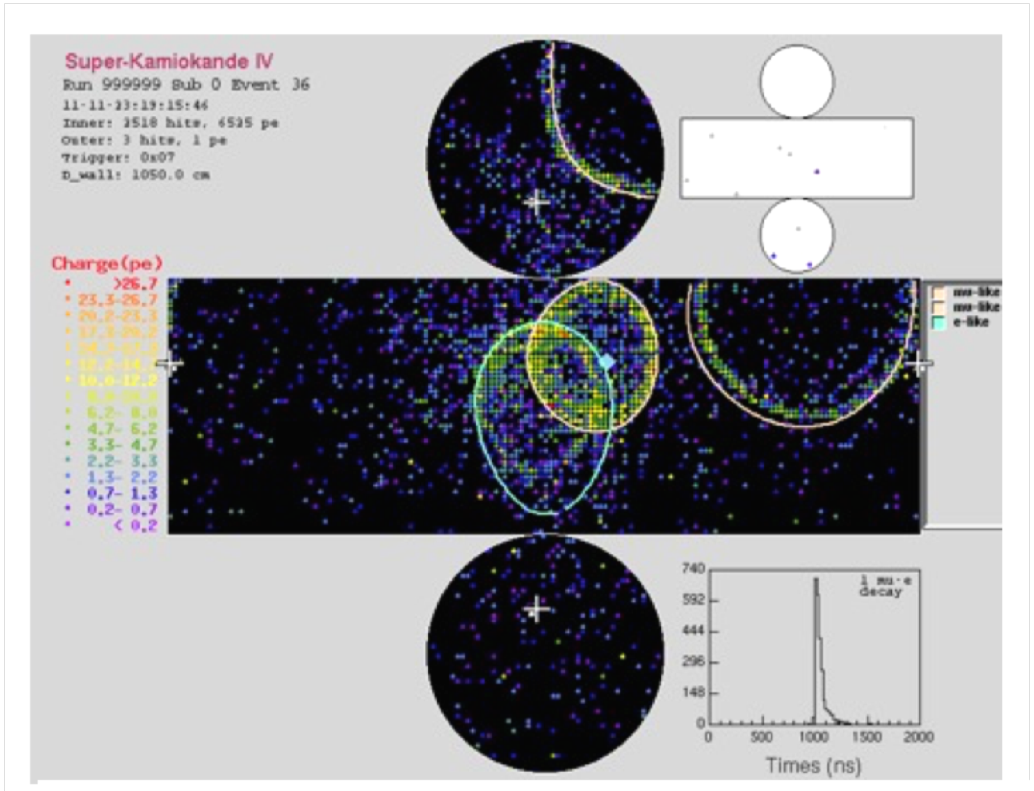
$$\frac{d^2\chi^2}{dx^2}(x_0) = 2H^T V^{-1} H = Cov^{-1}(x_0)$$

we insert now instead $$x_1 = x_0 - \left(\frac{d^2\chi^2}{dx^2}\right)^{-1} \frac{d\chi^2}{dx}$$

x_1 may not exactly minimize χ^2 – but it is, after all, better than x_0. Thus we iterate until $|x_n - x_{n-1}| < \varepsilon$.

This is basically MINUIT.

Track fits in Super-Kamiokande



- Reconstruct Cherenkov rings
- Extract track parameters X=
- Vertex position
- Momentum and direction
- Particle ID

By maximizing the total likelihood:

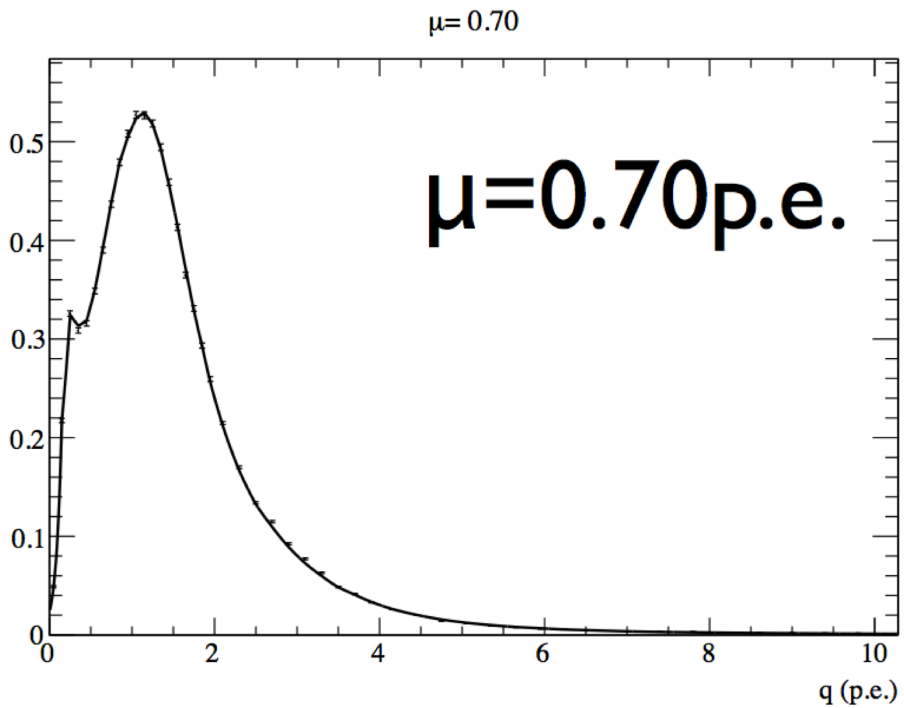
i-th PMT's charge, time

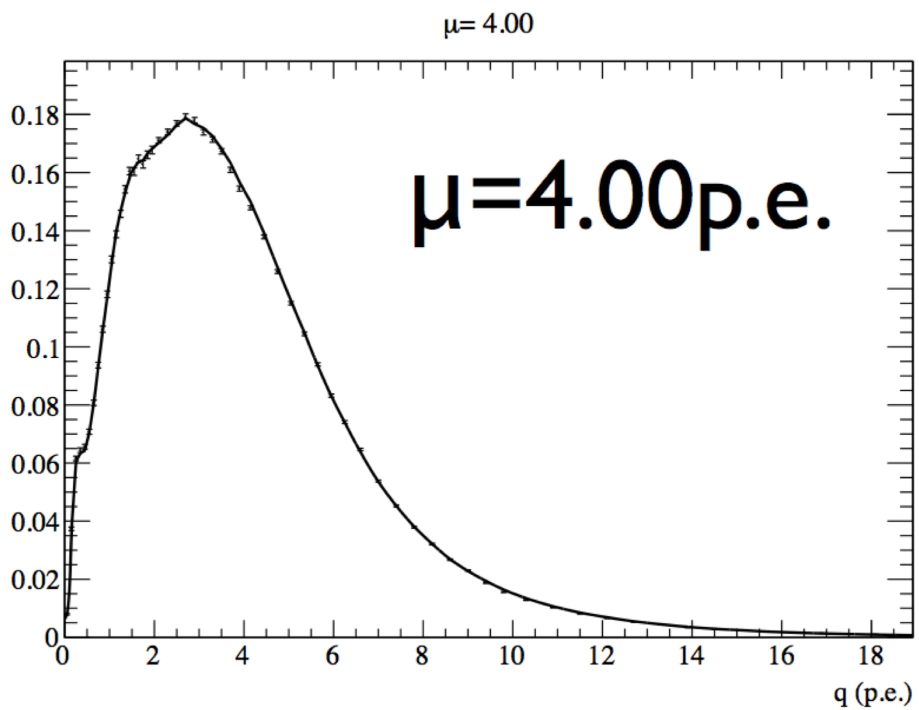
$$L(\mathbf{x}) = \prod_{i}^{\text{unhit}} P(i\text{unhit}|\mathbf{x}) \prod_{i}^{\text{hit}} P(i\text{hit}|\mathbf{x}) f_q(q_i|\mathbf{x}) f_t(t_i|\mathbf{x})$$

Unhit probability   Hit probability   Charge likelihood   Time likelihood

Track fits in Super-Kamiokande

- The individual pdf's are far from Gaussian !
- So good seeding is very important to avoid local minima and CPU consumption is very high





Dealing with multiple scattering

- The global chi-squared track fit can allow at each scattering plane a MS angle treated as an extra track parameter with a contribution to χ^2 of $(\theta / \theta_0)^2$

- Alternatively we can introduce correlations between surfaces in the covariance matrix V:

$$V \to V + S \Theta S^T \qquad S = \partial r / d\theta \quad \Theta_{jj} = \theta_{0j}^2$$

- MS is approximately Gaussian.

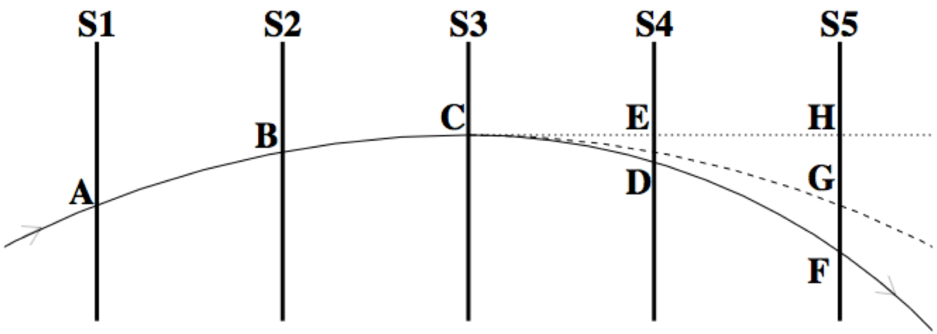
- *The Global Chisquared and the Kalman Filter only work optimally with Gaussian deviations from expectations.*

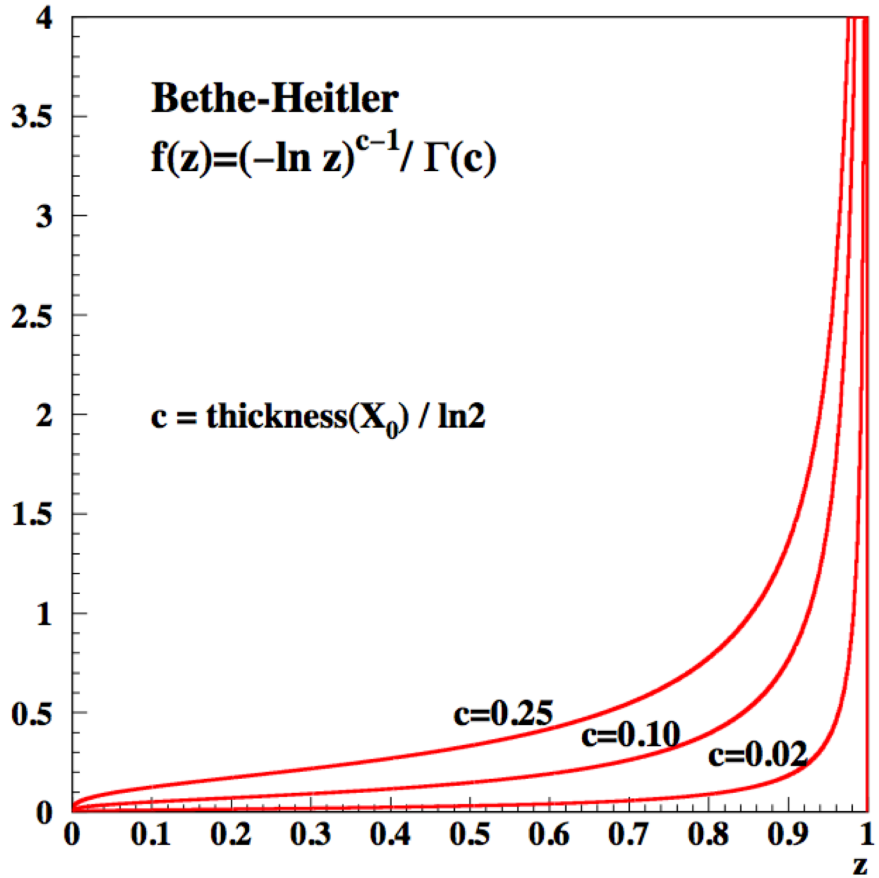
Dealing with Non Gaussian errors

- Special methods are needed to take care of *non-gaussian* influences. Typical example is hard photon radiation where the probability density for the electron to retain a fraction z of its energy follows the Bethe-Heitler law

$$f(z) = (-\ln z)^{c-1} / \Gamma(c)$$

$$c = X_{rl} / \ln 2$$





Gaussian Sum Filter

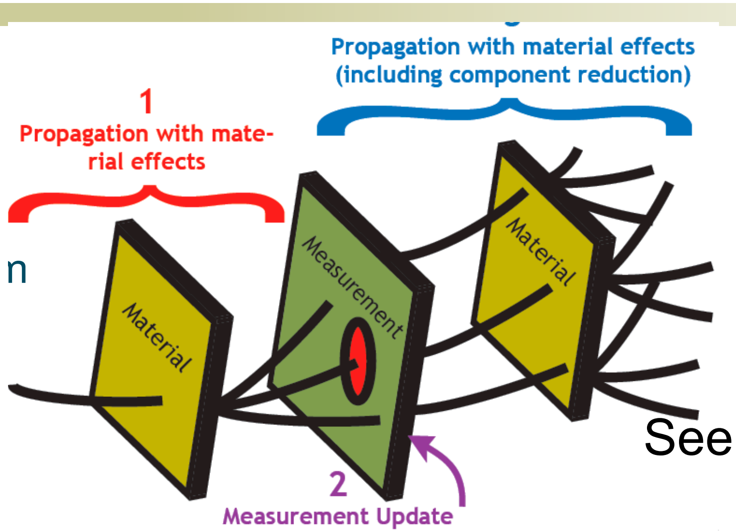


Figure from Salzburger lectures

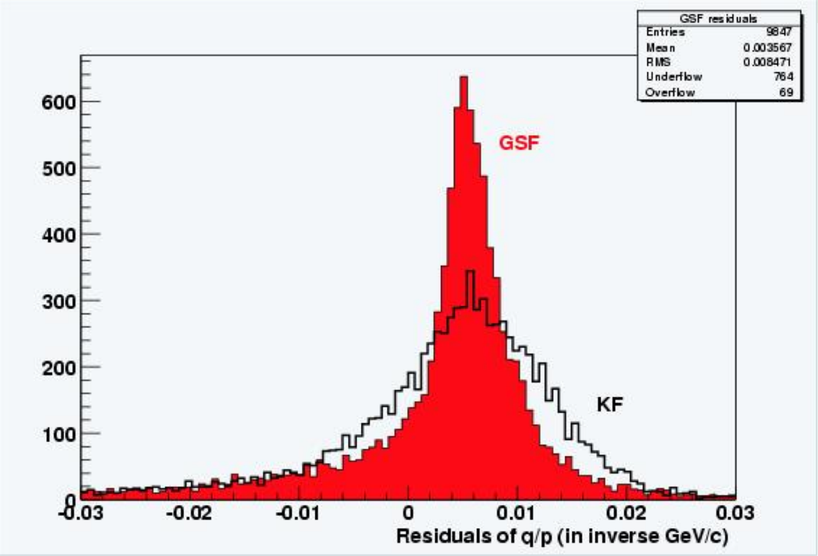
See eg R. Frühwirth and S. Frühwirth-Schnatter, 1998

- Branch the Kalman filter at each surface into parallel paths using a finite number of different Gaussian errors.
- This is the same as modelling e.g. the Bethe-Heitler as a sum of Gaussians

$$f(z) = \sum_{i}^{Nmax} g_i \phi(z, \mu_i, \sigma_i)$$

where the weights g_i, the average, μ_i, and variance, σ_i^2, of the energy are determined beforehand from simulation.

Gaussian Sum Filter



From A. Strandlie, CMS simulation 2003

- Effectively the track state branches out into a number of possibilities at each plane.
- *Component reduction*, must be carried out at some point to keep the number of branches from exploding.
- The resulting algorithm is very efficient in recovering from hard bremsstrahlung, but is also very CPU consuming. Often restricted to electron candidates.

Global optimization

- In case of competing assignments of hits to a given track candidate - or to noise - what is the optimal assignment?

- In essence this is the travelling salesman’s problem. It is not sure that the nearest hit is the best choice. The problem should be tackled by minimizing a total energy function.

- In the Elastic Arms Algorithm a number of “deformable track templates” must first be found. These should also include a “noise template”. The number of tracks stays fixed, but which hits are assigned to which tracks is not yet fixed.

Global optimization

- A "metric" M_{ia} is defined, typically as the squared distance from hit i to track template a.
- One could try to minimise to minimize a "total energy":

$$E = \sum_{a}^{Tracks} \sum_{i}^{Hits} \left[S_{ia} M_{ia} \right]$$

where the "assignment strength" S_{ia} is either 0 or 1.

- However, optimizing the S_{ia}'s is tricky since the energy-landscape is very "spiky" with lots of local minima.

Elastic arms and annealing

- This is tackled by annealing and fuzzy assignment strength:

$$S_{ia} = \frac{e^{-\beta M_{ia}}}{e^{-\beta\lambda} + \sum_{a=1}^{Tracks} e^{-\beta M_{ia}}}$$

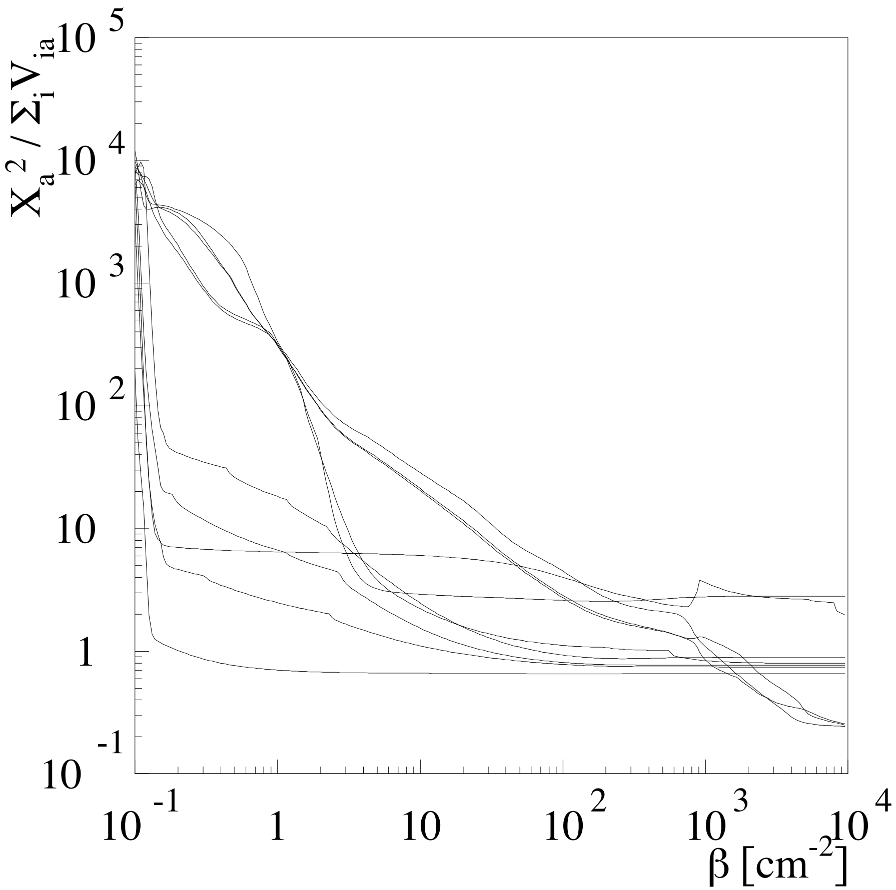
where $\beta = 1/T$ and λ is a "chisquared cut" of the order 10.

S_{i0} is here the assignment strength to noise ($M_{i0}=\lambda$)

- We now start at a high "temperature" where the S_{ia}'s are relatively large, even for distant hits. Few local minima.
- The track parameters are then iterated to the global minimum of E using its derivatives a la Newton-Raphson

Elastic arms and annealing

- We then lower the temperature (by eg 5%), repeat and continue until T<<1.
- At this point, all the S_{ia}'s take ~discreet values of 0 and 1.



Chisquared for 10 muon tracks in HERA-B with decreasing temperature. (From Borgmeier Diploma Thesis 1996)

More is found in R.Mankels review (arXiv:040239v1) from 2004.

Deterministic Annealing Filter

- A problem with a *global method* like Elastic Arms is that the number of tracks must be known beforehand.
- It gives you better hit sharing, but not better track finding efficiency.

- Therefore Frühwirth and Strandlie proposed to modify the (*local*) Kalman filter using an assignment probability S_{ik} for assigning hit i in plane k to the current track, thus giving all hits a say in the propagation of a given seed.

DAF assignment probabilities

- The assignment probability for each of n_k measurements in layer k to the current track is assumed to be proportional to a multivariate Gaussian:

$$\phi_k^i = \phi(m_k^i; H_k x_k, TV_k^i + H_k C_k^* H_k^T)$$

σ

where x here is the smoothed track state, but without involving layer k in the fit, and T a temperature parameter (the last term is the "track contribution" to the error, which can often be ignored).

This is nothing but the likelihood for a track to produce a given hit using scaled measurement errors. However, what we want is the posterior probability of the track parameters.

DAF assignment probability

- Allowing for the hypothesis that no hit is produced by the track in layer k, we normalise the assignment probability as:

$$S_i^k = \frac{\phi_k^i}{\sum_j^{n_k} \left(\Lambda_k^j + \phi_k^j\right)}$$

- The cut term may be parametrized as

$$\Lambda_k^j = \frac{1}{(2\pi)^{\dim(m)} \sqrt{T \det V_k^i}} \exp\left(-\frac{\lambda}{2T}\right)$$

where λ acts as a χ^2 cut-off at low temperature.

(Frühwirth and Strandlie, Comp.Phys.Comm 120,197 (1999))

DAF algorithm

- The filtered state can take several measurements per detector layer into account by using their weighted mean.

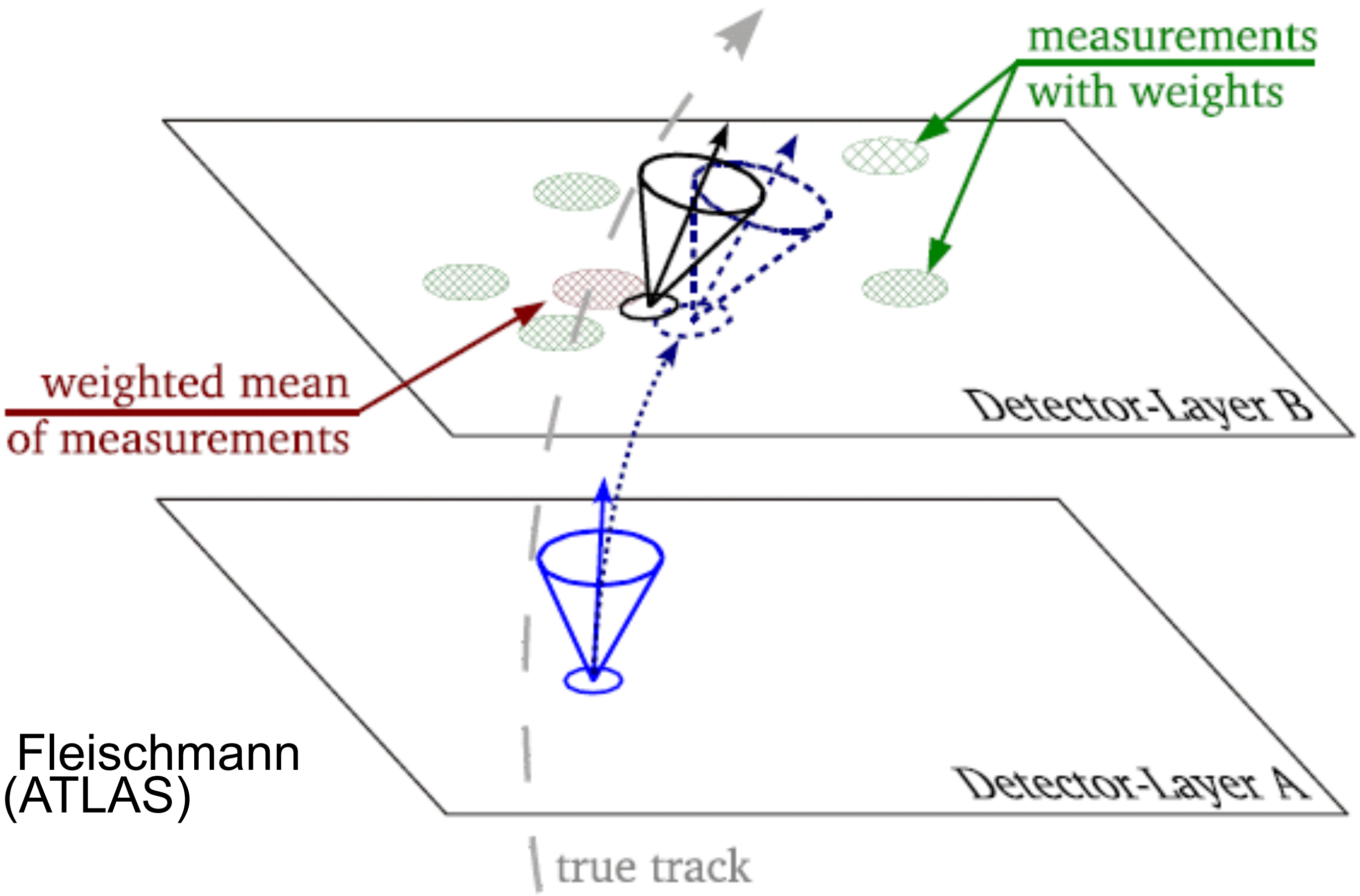
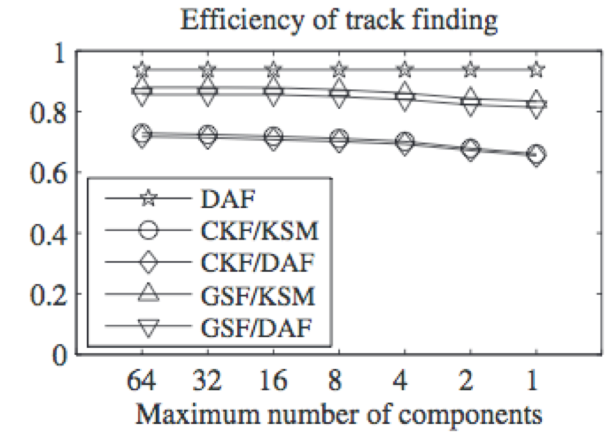
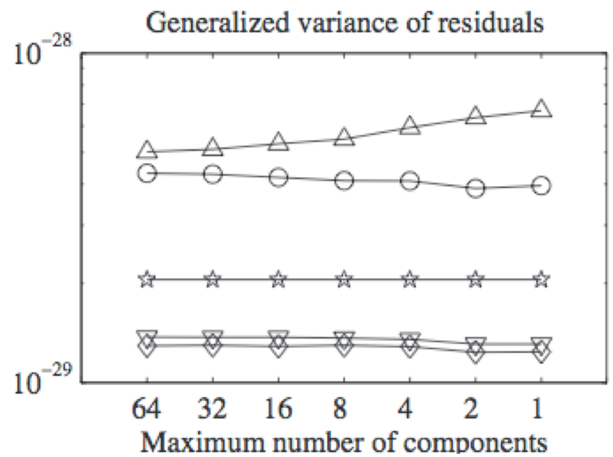


Figure from Fleischmann
PhD thesis (ATLAS)

DAF in practice

- The Deterministic Annealing Filter is reported to be especially effective in finding the best left-right choices in drift tubes.
- It can be used as an ”afterburner” and may significantly improve momentum resolution.





Simulated performance in an “ATLAS-like” setup of the DAF, either in standalone mode or as a track fitter following a CKF or GSF track finder. (Frühwirth and Strandlie, 2006)

DAF as a multi-track fitter

- It may be extended it to a multi-track fitter with several filters propagating in parallel.

  (Frühwirth and Strandlie, Comp.Phys.Comm,133(2000)34).

- In this case, the normalisation of assignment probabilities needs to be changed so that the sum runs over all accepted tracks competing for the measurements.

- The procedure again starts at a high temperature and iterates with decreasing tolerance, but without working with a fixed number of tracks. Candidates are rejected along the way.

Track scoring

- To resolve competition among tracks for the same hits, a "score" is used to decide the further fate of a track.

- ATLAS uses a combination of sub-scores:
  - Number of precision hits
  - Number of outlier hits
  - Holes (track passing through live sensor with no signal)
  - Shared hits (penalized if hit is not "shareable").
  - Total χ^2 per degree of freedom
- In a *second pass* the hits not yet assigned to a track may be reconsidered with larger tolerances to form, for example, low pT tracks or tracks from secondary interactions (long lived decays).

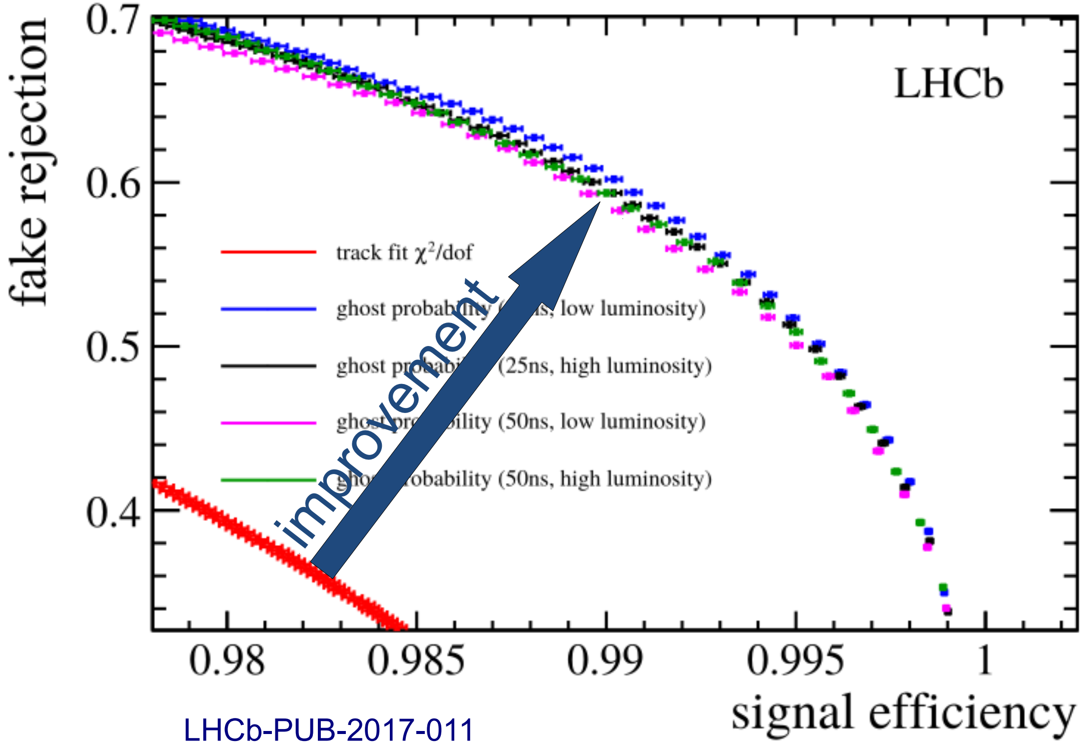
Track scoring

LCHb has in Run2 split the trigger in two levels, which enables online alignment and calibration. But CPU bandwidth is an issue.

To speed up, a shallow neural net has been designed (using the TMVA framework) for on-line rejection of fake tracks :

22 inputs -> m linear nodes ->
m non-lin nodes –>
m linear nodes ->
m non-lin nodes ->
Decision

The result is a 16% saving

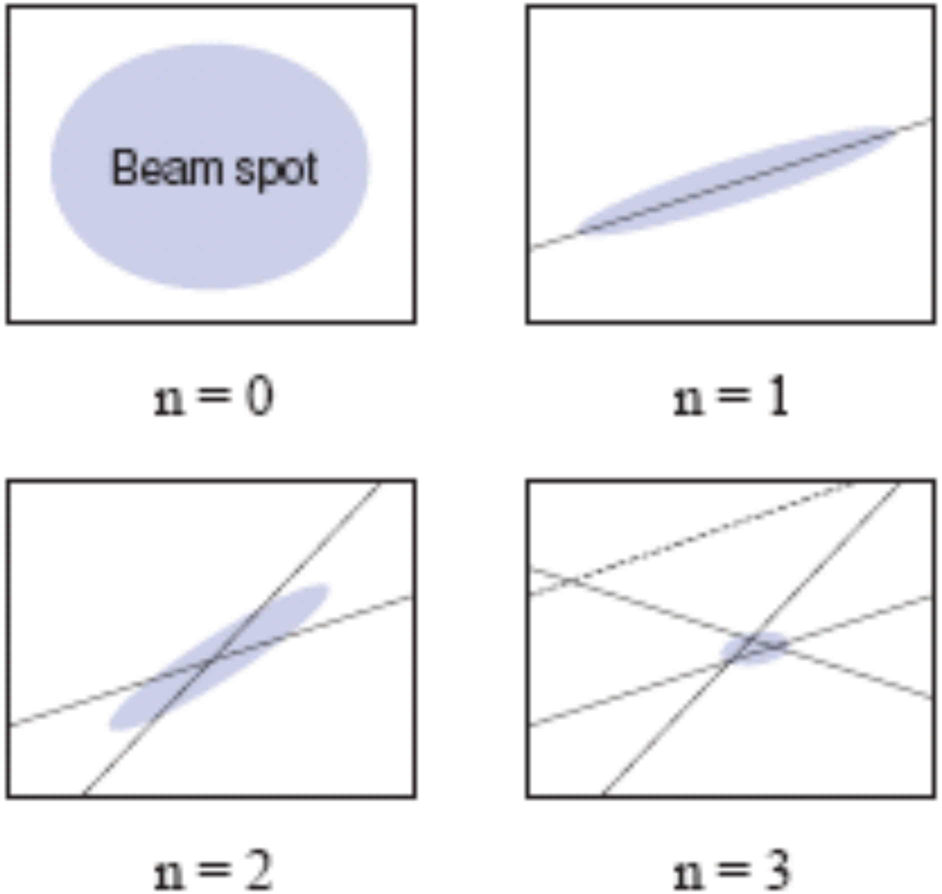


LHCb-PUB-2017-011

Finding the primary vertex

- Every few minutes the ”beam-spot” with high concentration of track perigees is recalculated.
- Hereafter, just two tracks suffices to provide an accurate seed for a vertex Kalman Filter.

Figure from Mankel



Finding the first vertex seed

- Great care must be taken in this very first step.

- ZEUS: candidate track pairs were ranked according to how many other pairs they agreed with. The best pair then started the chi-squared fit.

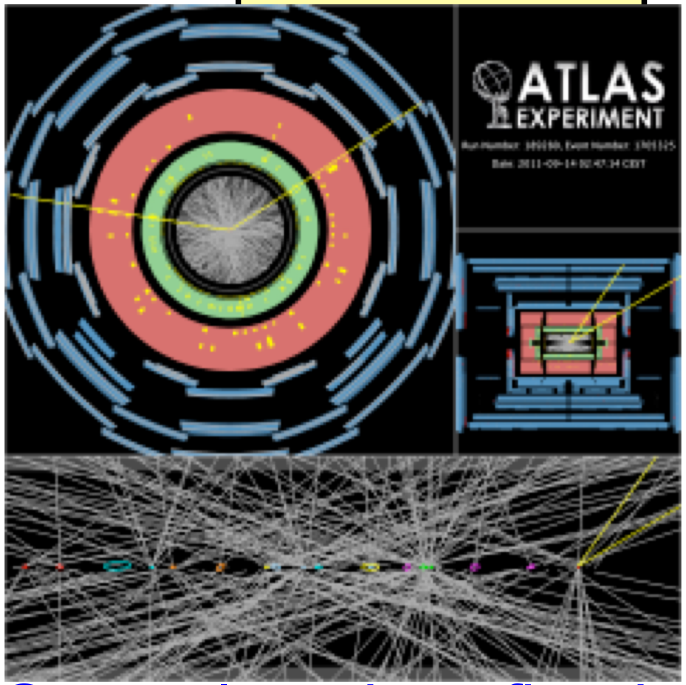
- CMS: finds regions with a high density of track pair crossings. Each track pair is weighted by a decreasing function of the distance between their two perigees. The position with the largest weight is the seed.

ATLAS Multi Vertex Finder

INPUT

Tracks

Track selection



Seeding tracks

Seed Finder

New vertex is seeded

Adaptive Multi Vertex Fitter

Vertex is fit together with all previous ones

Remaining tracks

Figure from Saltzburger

- Several vertices fitted simultaneously (20 vertices above)
- Several iterations with decreasing tolerance for assigning a track to a vertex a la the Multitrack DAF.

Billoir vertex fit

- The alternative to a Kalman Filter is the Newton-Raphson least squares fit (for vertices called a Billoir fit):
- It requires that the collection of tracks associated with each vertex is known in advance.
- Not only the vertex is fitted, but also the track momenta, this time with the constraint that they should all come from the same vertex point. This yields improved momenta.

Billoir vertex fit

- Let $\overline{v} = (x_v, y_v, z_v)$ be the vertex position for n tracks
- Let $\overline{p}_i = (p_{xi}, p_{yi}, p_{zi})$ be the i'th track momentum
- Let $\overline{x}_i = F(\overline{v}, \overline{p}_i)$ be the 5 track parameters of the i'th track at some reference surface.

x_i

- To first order in a Taylor series: $F = F(\overline{v}_0, \overline{p}_{0i}) + D_i \delta\overline{v} + E_i \delta\overline{p}_i$
- v_0 and p_{0i} are estimates of the vertex and track momenta.
- Let $\delta\overline{x}_i = \overline{x}_{i,meas} - F(\overline{v}_0, \overline{p}_{0i})$ and V_i be the covariance matrix for δx_i where *i* is the track number.
- Then

$$\chi^2 \approx \sum (\delta\overline{x}_i - D_i\delta\overline{v} - E_i\delta\overline{p}_i)^T V_i^{-1} (\delta\overline{x}_i - D_i\delta\overline{v} - E_i\delta\overline{p}_i)$$

Billoir vertex fit

Tatjana Lenz master thesis

- Like in the track fit, we solve the 3+3n equations minimising χ^2:

$$\bar{v} = \bar{v}_0 + (A - \sum B_i C_i^{-1} B^T{}_i)^{-1} (\bar{t} - \sum (B_i C_i^{-1})^T \bar{u}_i)$$

$$\bar{p}_i = \bar{p}_{0i} + C_i^{-1}(\bar{u}_i - B_i^T \delta\bar{v})$$

$$A = \sum D_i^T V_i^{-1} D_i \qquad B_i = D_i^T V_i^{-1} E_i \qquad C_i = E_i^T V_i^{-1} E_i$$

$$\bar{t} = \sum D_i^T V_i^{-1} \delta\bar{x}_i \qquad \bar{u}_i = E_i^T V_i^{-1} \delta\bar{x}_i$$

- The real work for the programmer is in the initial calculation of D and E - and in the initial guess of v_0

Billoir vertex fit

- Now interchange $\bar{v}_0$ with $\bar{v}$ and continue until convergence
- The covariance of the fitted parameters is at each step:

$$\mathrm{cov}(\bar{v}) = \left(A - \sum B_i C_i^{-1} B_i^T\right)$$

$$\mathrm{cov}(\bar{p}_i) = C_i^{-1} + (B_i C_i^{-1})^T \mathrm{cov}(\bar{v}) B_i C_i^{-1}$$

$$\mathrm{cov}(\bar{v}, \bar{p}_i) = -\mathrm{cov}(\bar{v}) D_i E_i^{-1}$$

- We also get correlations between the track momenta:

$$\mathrm{cov}(\bar{p}_i, \bar{p}_j) = \delta_{ij} E_j^{-1} - E_i^{-1} D_i^T \mathrm{cov}(\bar{v}, \bar{p}_j)$$

Exploiting external constraints

- If you have some partial prior knowledge about the beam collision position b, just add an extra contribution to the χ^2 , which will change its derivatives:

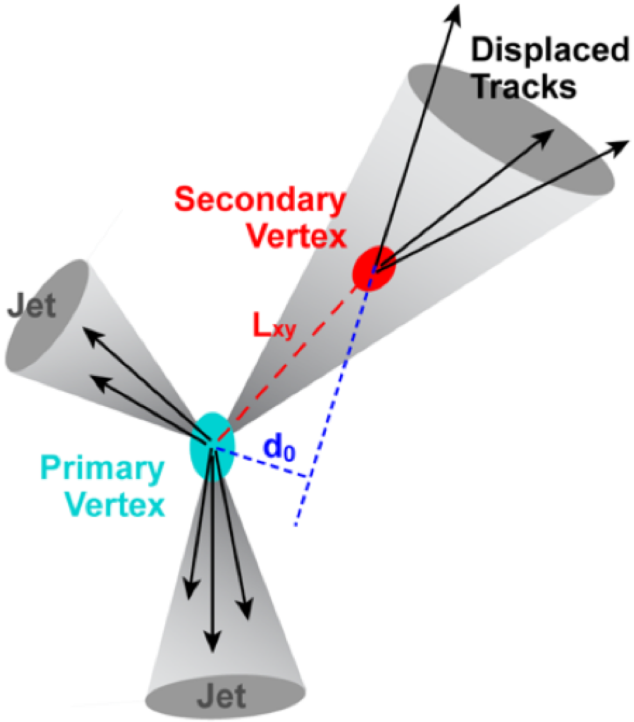
$$\delta\chi^2 = (\bar{v} - \bar{b})^T V_b^{-1} (\bar{v} - \bar{b})$$

- If you have an exact constraint, like momentum conservation, you can use the method of Lagrange Multipliers:

$$\delta\chi^2 = -\bar{\lambda} \bullet \bar{d}(\bar{v}, \sum \bar{p}_i)$$

- Where λ are 3 new arbitrary fit parameters and d is some function that in principle is exactly zero.

b-jet tagging



- Jets with a B-hadron can be identified by the lifetime (~1.5 ps) and high mass of the b quark (~4.2 GeV)

From the ATLAS B-physics group

using secondary vertices

- One way to find b-jets is to reconstruct the decay chain:

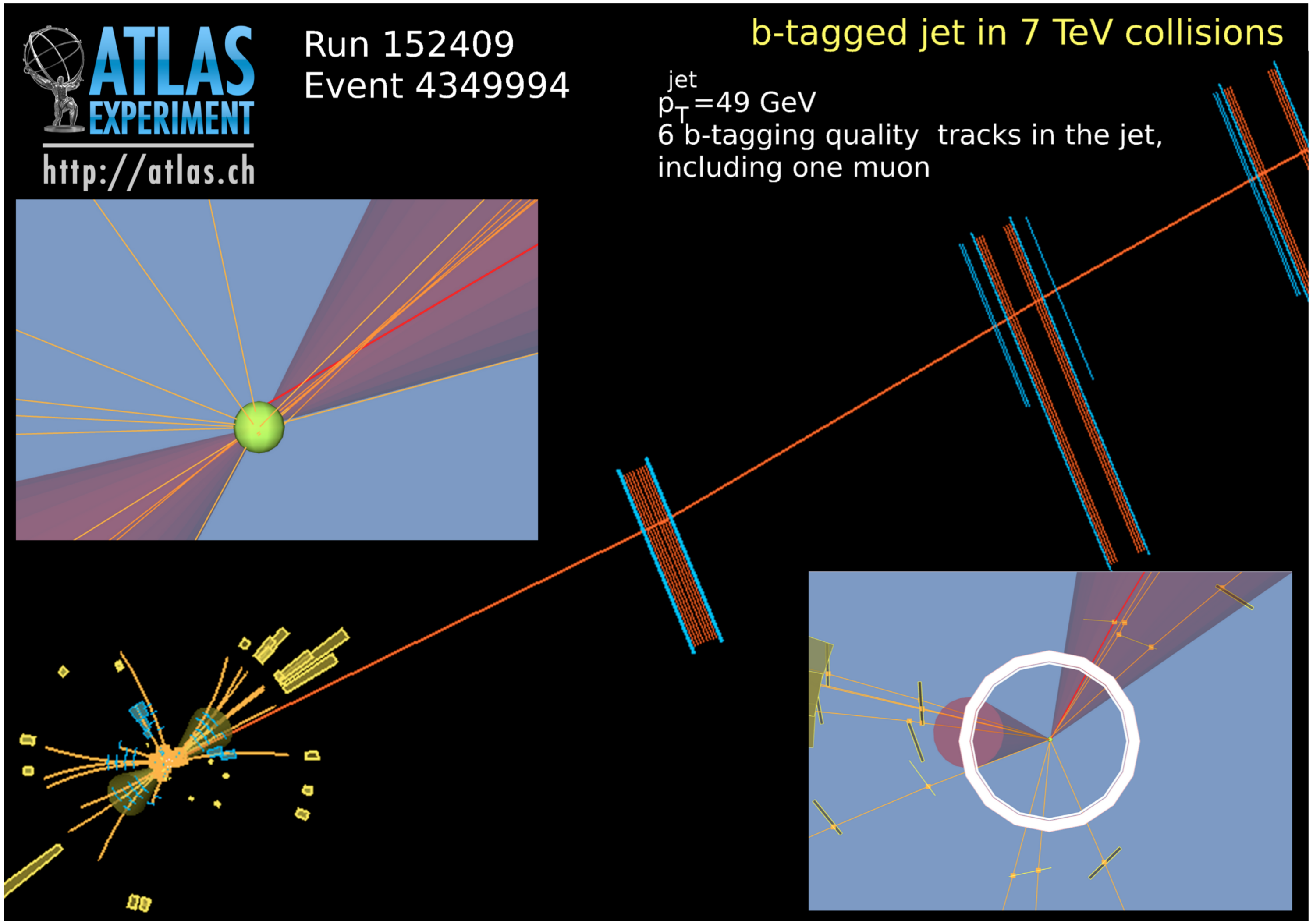
  b-jet -> B^-+X -> D^0+X+Y -> K^-+X+Y+Z

Where X are b-quark fragmentation particles, Y are other particles from B^- decay and Z other particles from D decay.

- Another way is to use multi-variate techniques on the number of found vertices along the jet-axis, their distances from PV, the mass and number of tracks at each vertex etc to discriminate between b-quarks and lighter partons.

Using a lepton tag

- Due to the high B-meson mass, its leptonic decay (~10%) has a higher p_T^{rel} than leptons from light parton jets

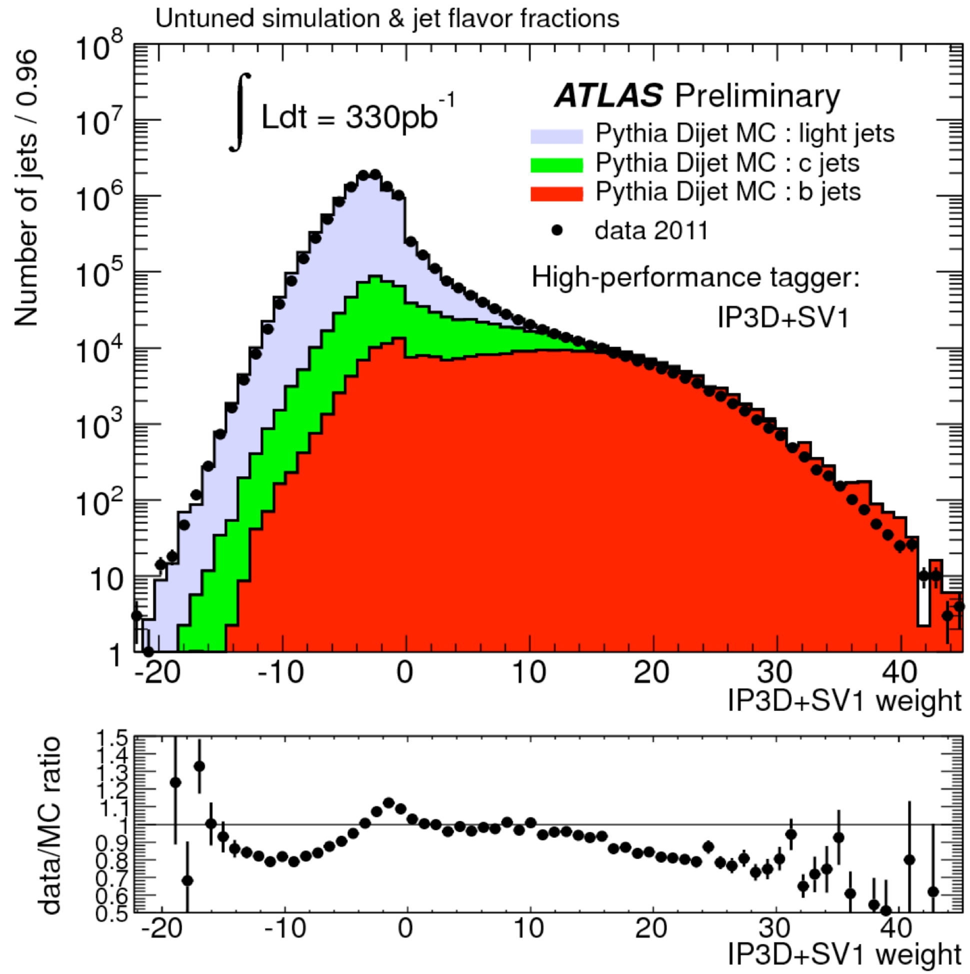


Combining all of it

- ATLAS combines the $S=d_0/\Delta d_0$ for all tracks in a jet:

$$P(jet) = \Pi \cdot \sum_{i=0}^{N_{trk}} \frac{-\ln(\Pi)^i}{i!} \quad where \quad \Pi = \prod_{i \in jet} \int_{S_i}^{+\infty} f(S)dS$$

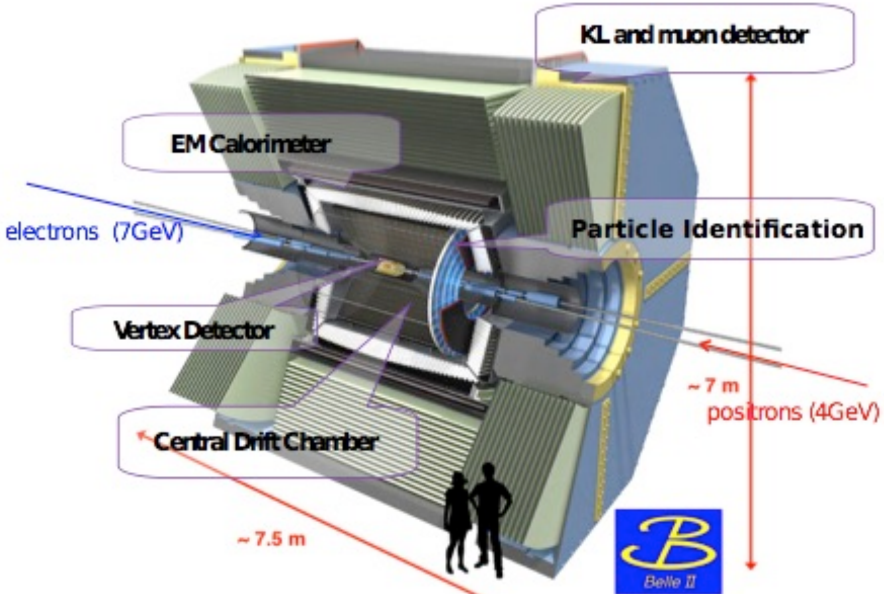
S is the impact parameter significance and f(S) its light jet probability.
The P(jet) estimator has many nice properties.

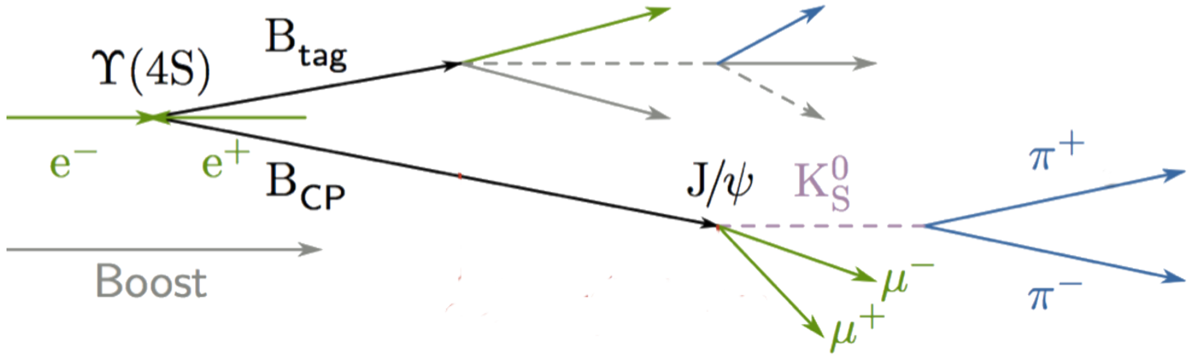


- Finally combine everything:

combined Likelihood
or Multi-Variate Analysis

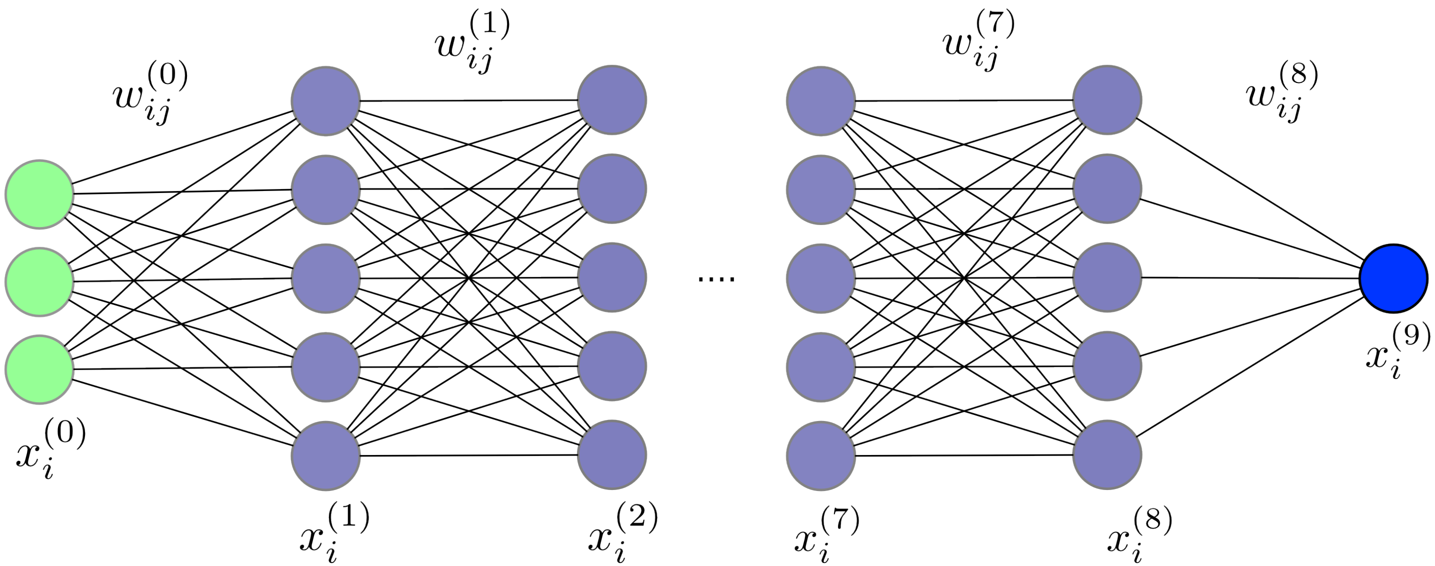
B-tagging at Belle II





B-tagging at Belle II

140 input features: kinematics, PID, Pvalues ..



$$x_i^{(k+1)} = \sigma^{(k+1)} \left(\sum_{j=1} w_{ij}^{(k)} x_j^{(k)} + w_{i0}^{(k)} \right)$$

B-tagging at Belle II

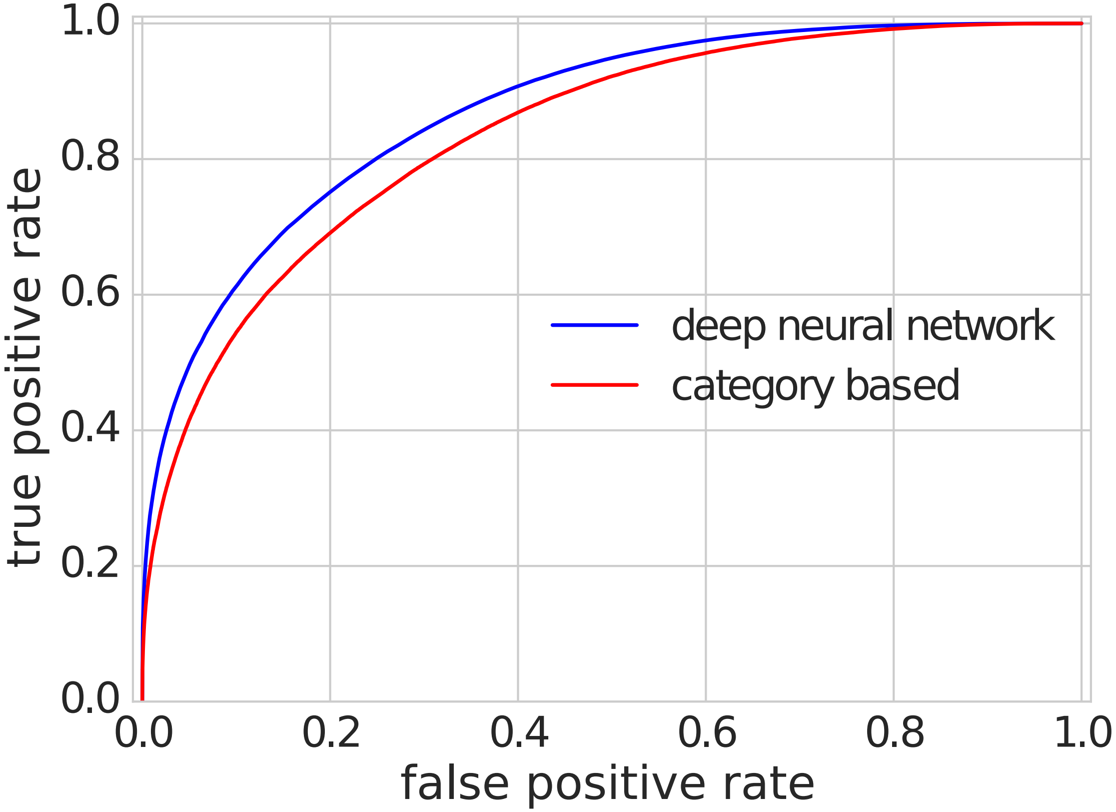
$t_n \in \{0,1\}$

- compare network output y_n and true information t_n
- cross-entropy
  $E(\mathbf{w}) = -\sum_{n=1}[t_n \ln y_n + (1-t_n)\ln(1-y_n)]$

- weight update
  $\Delta w^i = -\eta \nabla E_n(w^i)$
- weight update with momentum
  $\Delta w^i = -\eta \nabla E_n(w^i) + \mu \Delta w^{i-1}$

Train on 12 million events

B-tagging at Belle II



Lagrange Multipliers

- Let x be the track parameters of the two tracks that form a conversion candidate. The constraints must be expressed as some functions, H(x), being exactly zero.
- We again expand around an approximate solution x_A:

$$\frac{\partial \overline{H}}{\partial \overline{x}}(\overline{x}-\overline{x}_A)+\overline{H}(\overline{x}_A)=\overline{D}\Delta\overline{x}+\overline{d}=\overline{0}$$

- In a photon conversion, the e+e- emerge parallel from a common point, and the expression is like

$$\overline{d}=\overline{H}(\overline{p}_1,\overline{p}_2,\overline{r}_1,\overline{r}_2)_A=(\frac{\overline{p}_1}{E_1}-\frac{\overline{p}_2}{E_2},\overline{r}_1-\overline{r}_2)_A$$

- Where the p’s and r’s refer to the start points of the tracks.

Lagrange Multipliers

- The function to be minimized is now (dropping vector bars):

$$\chi^2 = (x - x_0)^T V_0^{-1} (x - x_0) + 2\lambda^T (D(x - x_A) + d)$$

- The minimum is found in the space of the track parameters x=(p,r) and the real constants λ.
- The ”0” refer to the unconstrained solution from the track fits and the “A” to the previous iteration of this fit.
- The solutions have to be iterated since the constraint equations were linearized.
- MINUIT does all of that behind the scenes, but is not normally possible to use in a reconstruction program.

Solution to the constrained fit

- (put derivatives of chi2 to zero and solve by substitution):

$$x = x_0 - V_0 D^T \lambda$$

$$\lambda = V_D (D(x_0 - x_A) + d)$$

$$V_D = (D V_0 D^T)^{-1}$$

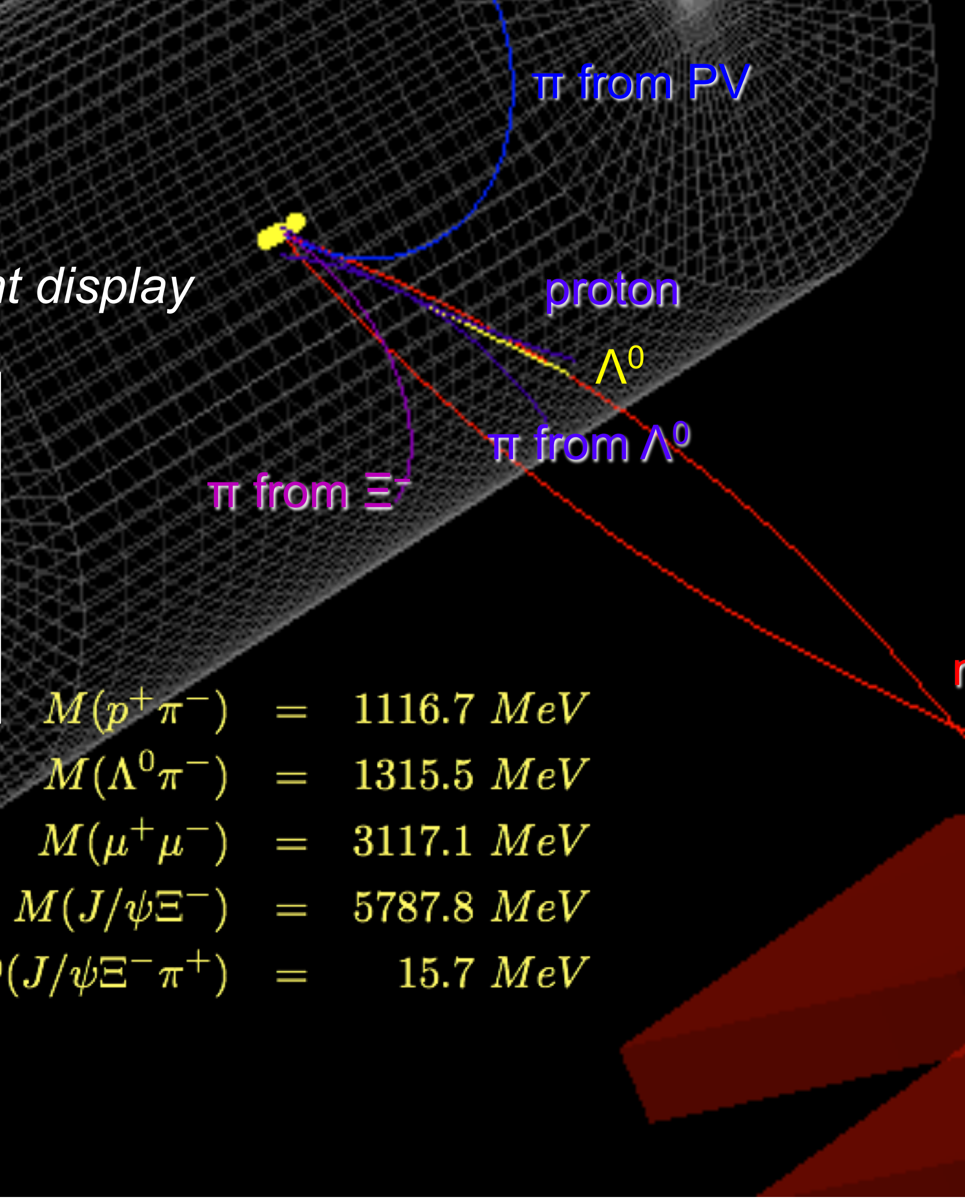
$$V_x = V_0 - V_0 D^T V_D V_0$$

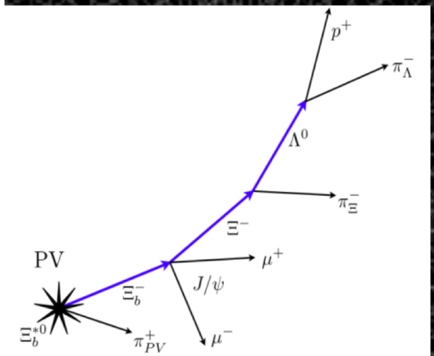
$$\chi^2 = \lambda^T V_D^{-1} \lambda$$

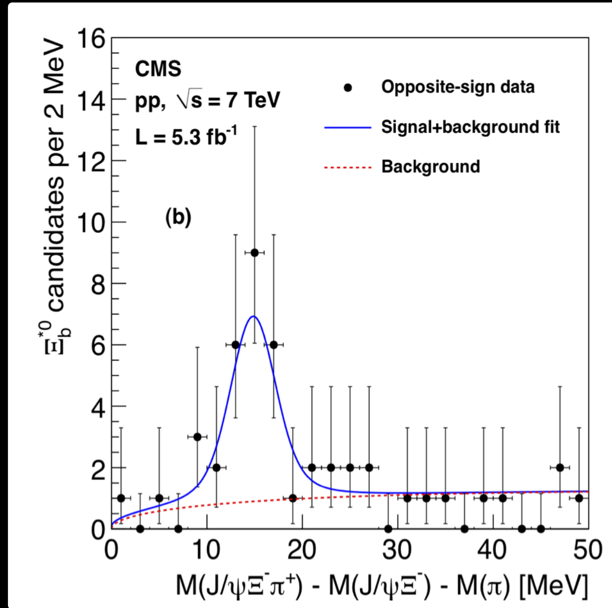
New ParticleDiscovery

The Ξ_b^{*0} involves elegant cascade that CMS tracker handles beautifully.

Candidate event display







$$
\begin{aligned}
M(p^+\pi^-) &= 1116.7\ MeV \\
M(\Lambda^0\pi^-) &= 1315.5\ MeV \\
M(\mu^+\mu^-) &= 3117.1\ MeV \\
M(J/\psi\Xi^-) &= 5787.8\ MeV \\
Q(J/\psi\Xi^-\pi^+) &= 15.7\ MeV
\end{aligned}
$$

Alignment

- In order to have high resolution and unbiased tracking, the detector elements must be correctly aligned.

- This is partly achieved by optical survey, and for example laser alignment systems, to track short-term movements.

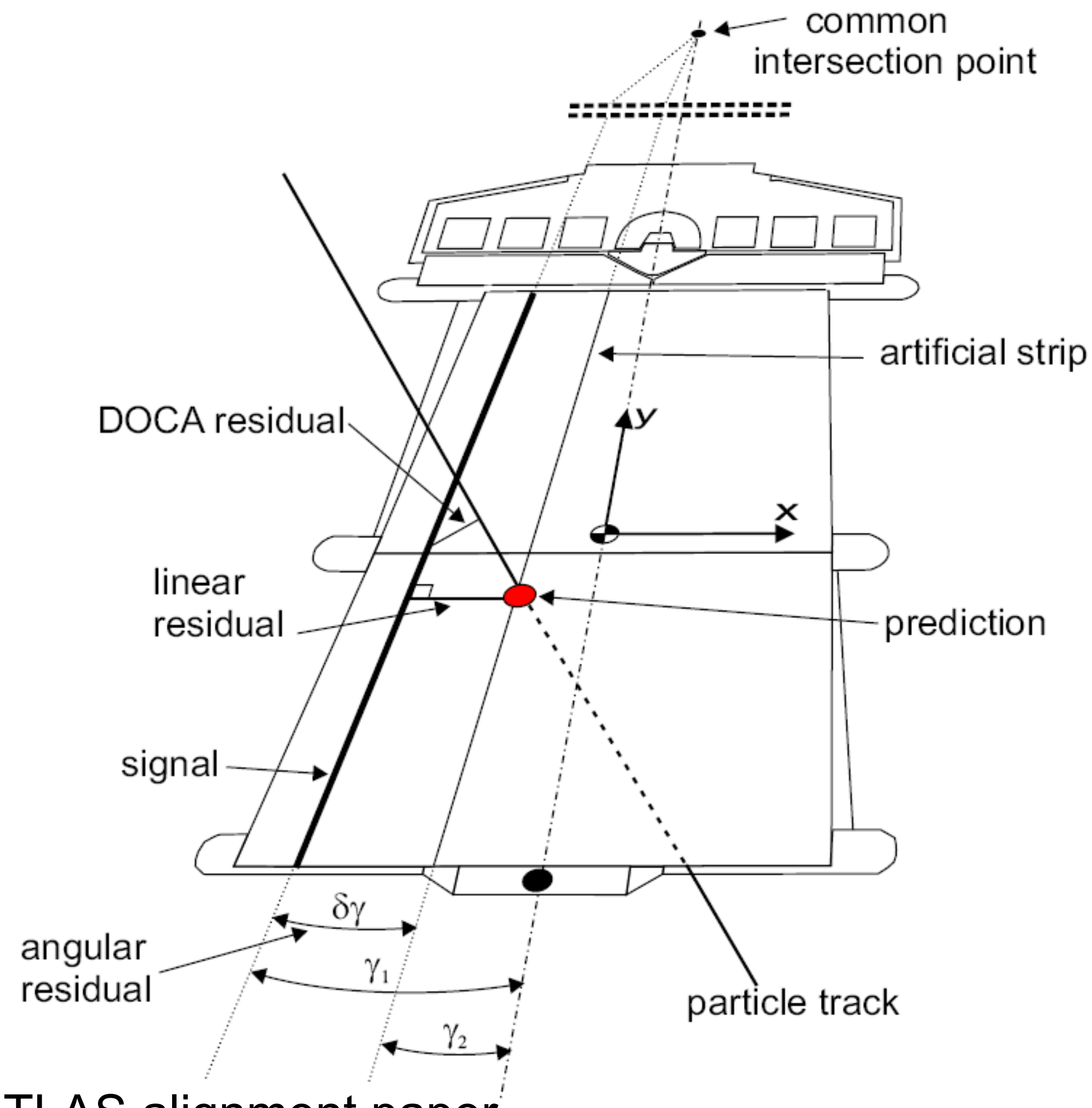
- The ultimate alignment precision, however, is achieved by using the fitted tracks themselves.

Local alignment with tracks

- How to determine the alignment corrections to the position and orientation of each detector element from the reconstructed tracks?
- Consider a single measured coordinate y_i and a track model $y=h(\mathbf{x},\boldsymbol{\alpha})$, where $\mathbf{x}$ are the track parameters and $\boldsymbol{\alpha}$ are the alignment corrections.
- A straight forward estimate of $\delta\alpha_i$ is simply the

  average residual $\langle r_i = y_i - h(x,\alpha)\rangle$
- If the considered plane does not take part in the fitted track, r_i is called an unbiased residual.
- This *"local alignment"* requires in general many iterations because the correlations between planes induced by the fitted tracks are ignored with this method.

Alignment with tracks



From ATLAS alignment paper

Global alignment with tracks

- In the *"global"* approach we define a total χ^2 of a large track sample:

$$\chi^2 = \sum_{tracks} r^T R^{-1} r$$

$$r(x,\alpha,m) = m - h(x,\alpha)$$

where r are the residuals, α the alignment parameters of the detector elements and x the individual track parameters.

We would like to simultaneously minimise χ^2 both with respect to the millions of x's and to the many α's.

Sounds impossible, but it isn't!

Global alignment with tracks

- After fitting for the track parameters, x, we have to first order the *total* derivative wrt the alignment :

$$\frac{d\chi^2}{d\overline{\alpha}} = 2\sum_{tracks} A^T R^{-1} \overline{r}$$

where *A* is the *partial* derivative of *r wrt α.*

- If R is diagonal, the derivative receives only contributions from local detector elements for which A=δr/δα is non-zero.

- Finding δα so that total derivative be zero thus results in M coupled equations, just like for the Billoir vertex fit.

Global alignment with tracks

- The M equations to be solved are :

$$\frac{d\chi^2}{d\overline{\alpha}} = -\frac{d^2\chi^2}{d\overline{\alpha}^2}\Delta\overline{\alpha}$$

- CMS minimises the distance between the two sides of the equation with a program called MINRES.
- ATLAS calculates eigenvectors and eigenvalues of the second derivative, exploiting the sparseness of this matrix, in order to do a fast inversion.

Global alignment with tracks

- The explicit solution to the alignment problem is thus

$$\Delta\alpha = -\left(\frac{d^2\chi^2}{d\alpha^2}\right)^{-1}\frac{d\chi^2}{d\alpha}$$

or, assuming r is linear in α,:

$$\Delta\alpha_i = -\left[\sum_{tracks}\frac{\partial\overline{r}}{\partial\alpha_i}R^{-1}\frac{\partial\overline{r}^T}{\partial\alpha_j}\right]^{-1}\sum_{tracks}\left[\frac{\partial\overline{r}}{\partial\alpha_j}R^{-1}r^T\right]$$

where $R = V - HCH^T$ is the covariance matrix of the residual vector of a track. See ATL-INDET-PUB-2007-009.

Eigen-modes of distortion

- Let

$$b = -\frac{1}{2}\frac{d\chi^2}{d\overline{\alpha}}, \qquad A = \frac{1}{2}\frac{d^2\chi^2}{d\overline{\alpha}^2}$$

Then the covariance of the corrections is C(Δα)=A^{-1} .

Since this matrix has an inverse, it can be diagonalized and written in terms of its eigenvectors u and eigenvalues d.

$$C_{kl}(\Delta\overline{\alpha}) = \sum_j^M \frac{1}{d_j} u_k^{(j)} u_l^{(j)}$$

where

$$\Delta\overline{\alpha} = \sum_j^M \frac{1}{d_j} (u^{(j)T} b) u^{(j)}$$

Eigen-modes of distortion

- The eigenvectors are collective orthogonal distortions of the detector. The change in the χ2 due to the correction Δα receives independent contributions from each mode j:

$$\Delta\chi^2 = -2\sum_{j}^{M}\frac{1}{d_j}(u^{(j)T}b)^2$$

- Thus we can identify and correct the contributions from the ortho-normal modes independently of each other.

Weak modes

Clearly there is a problem if d_j=0.
Small eigenvalues of A correspond to small $d\chi^2/d\alpha$. These are called ”weak modes”, poorly constrained by the data.

There are some distortions which cannot be seen in the residuals but still may spoil the momentum measurement.

Weak modes

The only way out is to use external constraints.
If, for example, an optical survey yields the alignment shift α_{survey} with precision σ, you could add a piece

$$\Delta\chi^2 = (\alpha - \alpha_{survey})^2/\sigma^2$$

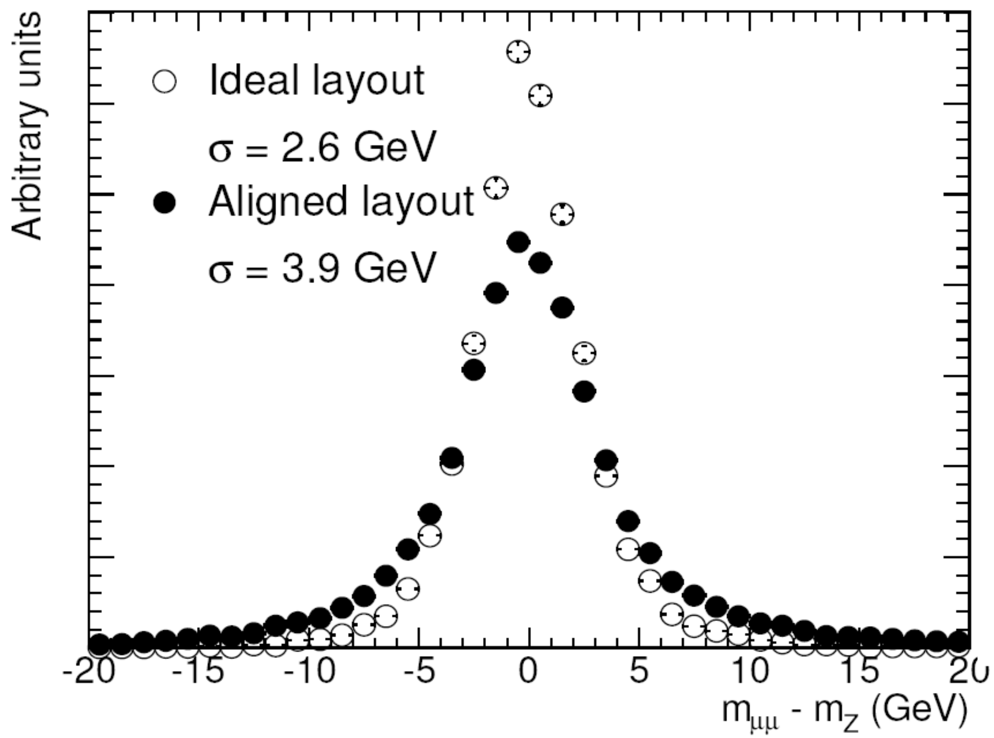
In general, if relations g(α)=0 exist with covariance G, add:

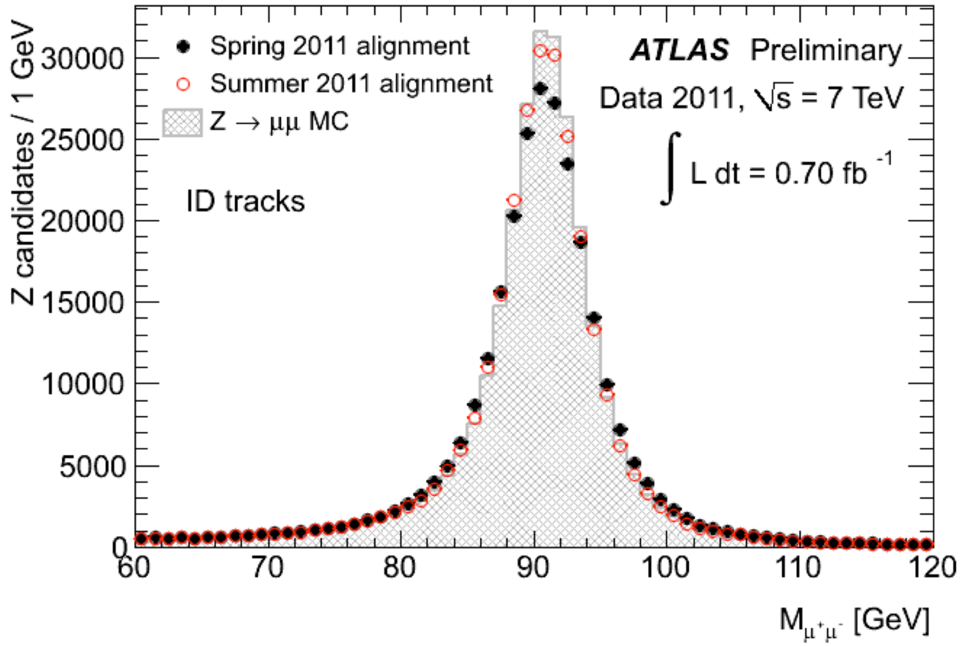
$$\Delta\chi^2 = g^T G g$$

Exact constraints, such as energy-momentum conservation, are best taken into account with *Lagrange multipliers*.

Effect of weak mode misalignment

- ATLAS saw the troubles from weak modes already in simulation. Extra constraints from cosmics and combined detectors helped a lot.
- 2008 2011

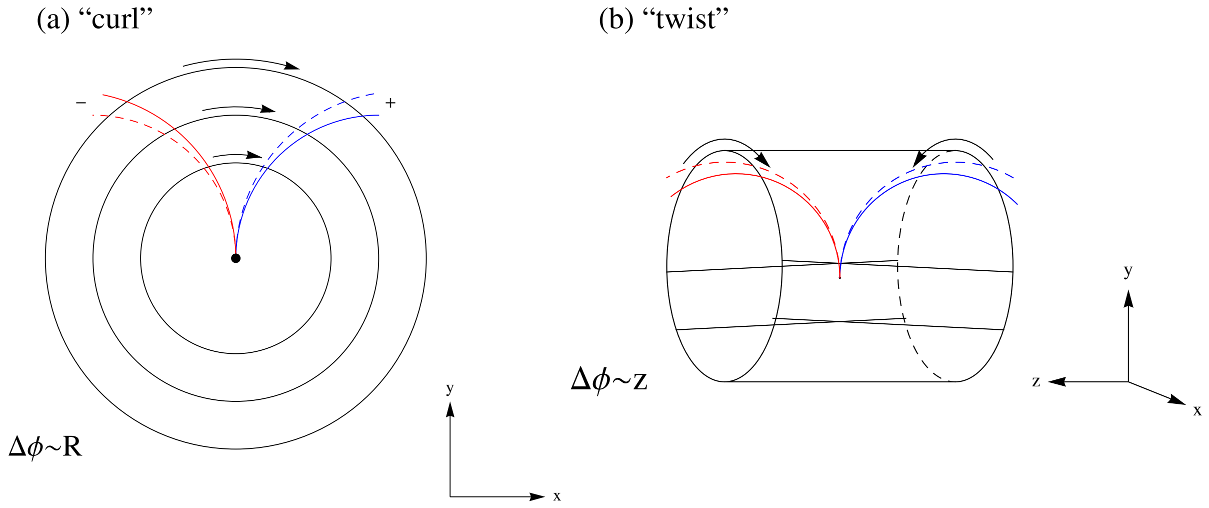




Curls and twists (q anti-sym)

- A rotation in φ proportional with R of the various layers would approximately conserve the helix-shape but bias the momentum (different for positive and negative charge).

  That is called a curl.

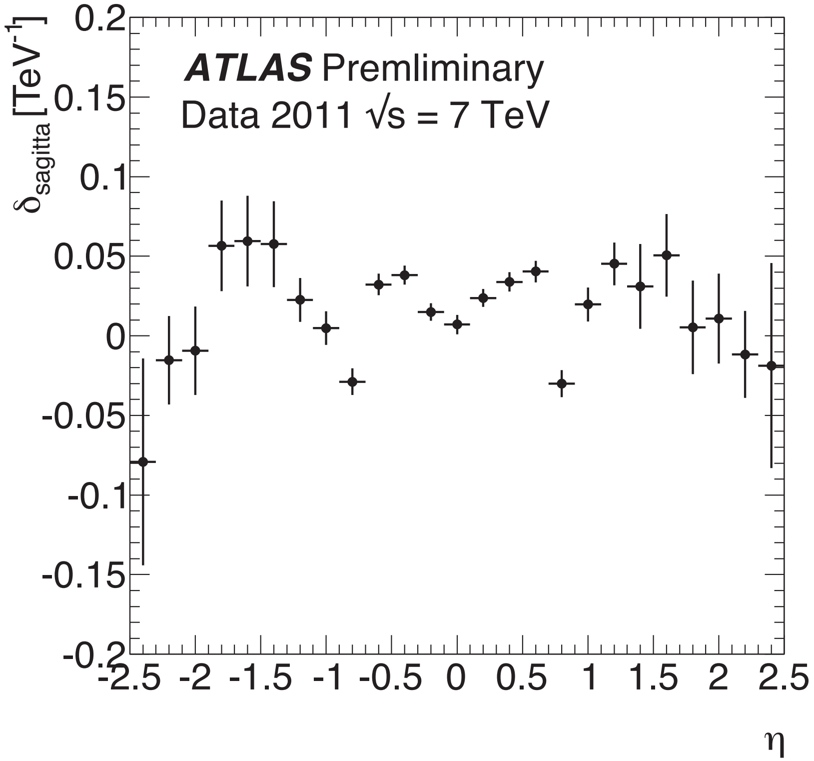
- 

- A rotation of the end of a cylinder layer would approximately conserve the helix-shape but bias the momentum in an η dependent way. That is called a twist.

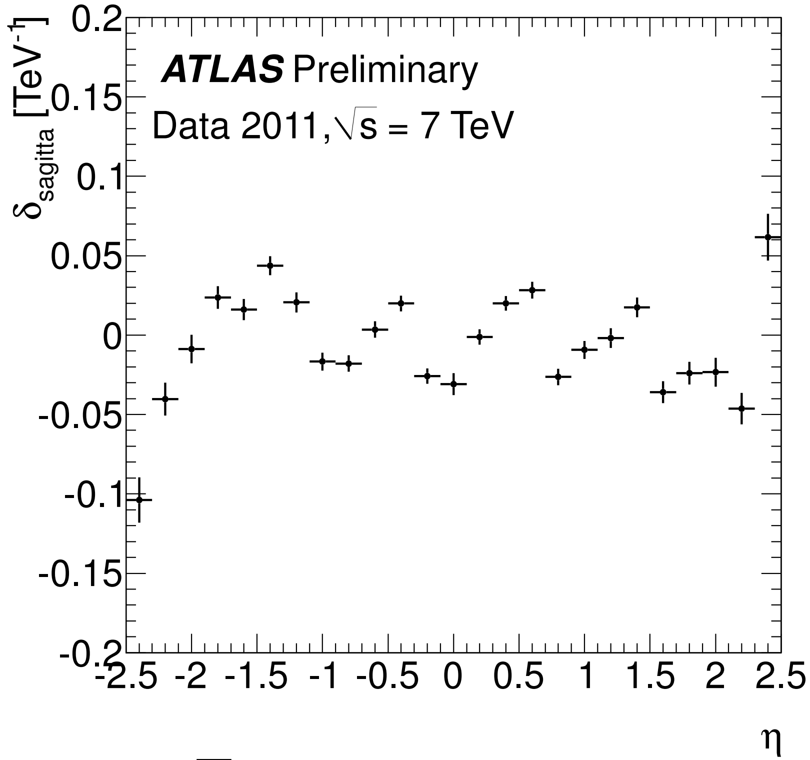
From the ATLAS Silicon alignment group, Bruckman et al

Checking for twists and curls

- No effect of twists and curls seen after alignment
- (Note: measurement uncertainty alone gives a resolution of 0.6 TeV-1)



E/p (e+ e-)

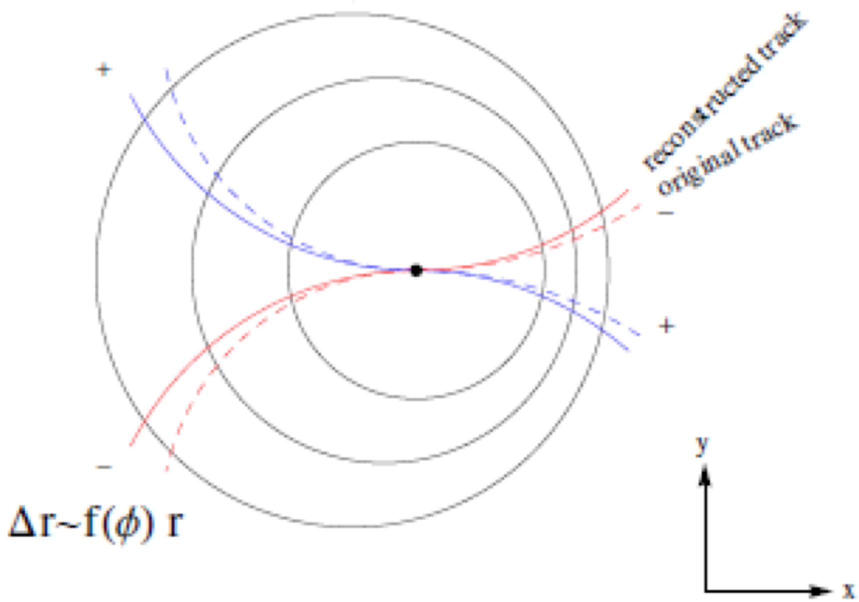


Z->mu+mu- mass

$$q/p_{Corr} = q/p_{Rec}(1 - qp_T\delta_{sagitta})$$

Radial distorsions

- There are also weak mode distortions affecting charges symmetrically – but different at different phi.



From the ATLAS Silicon alignment group, Bruckman et al

Checking radial deformations

- Phi dependence of low mass resonances

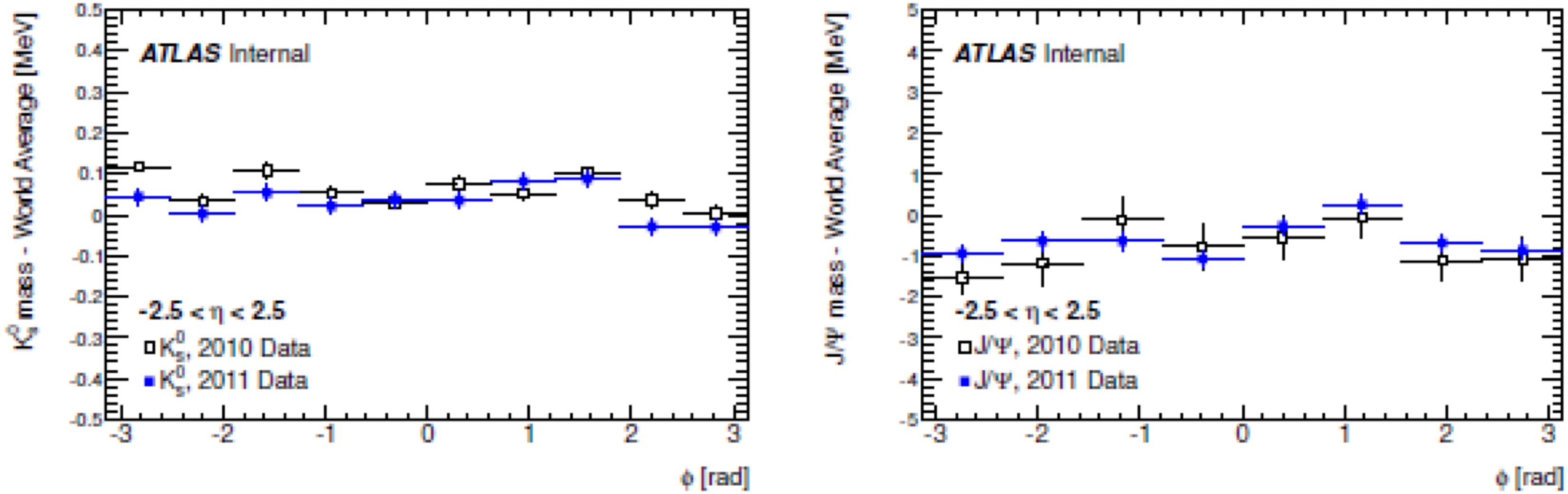
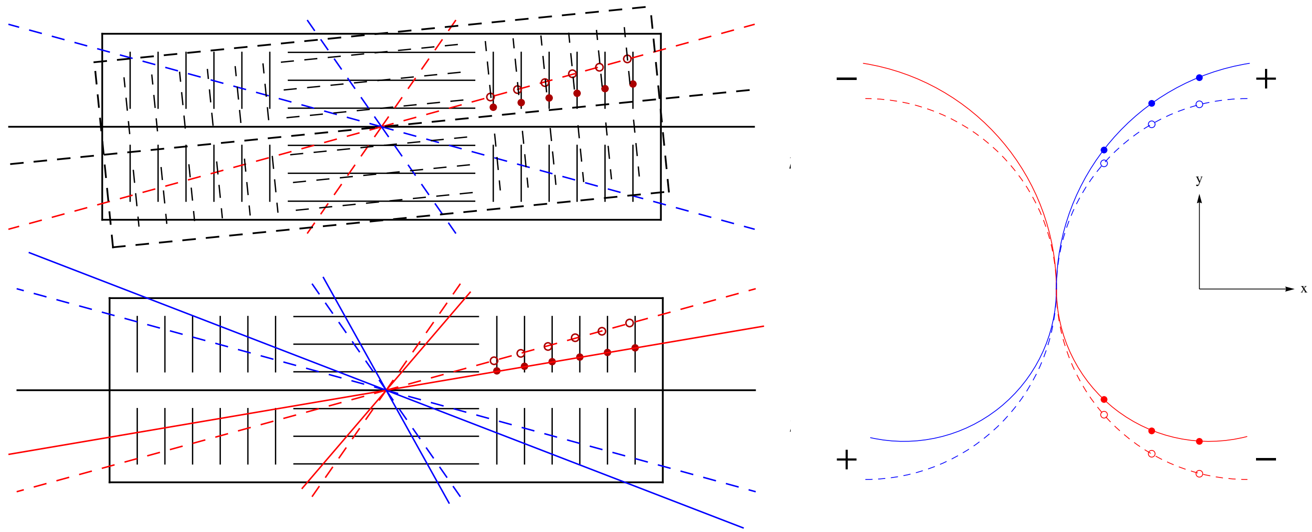


Figure 8: $m_{K_S^0}(\phi_{K_S^0})$ (left) and $m_{J/\psi}(\phi_{J/\psi})$ (right) as a function of ϕ-direction of the resonance.

From the ATLAS Silicon alignment group, Bruckman et al

B field rotations

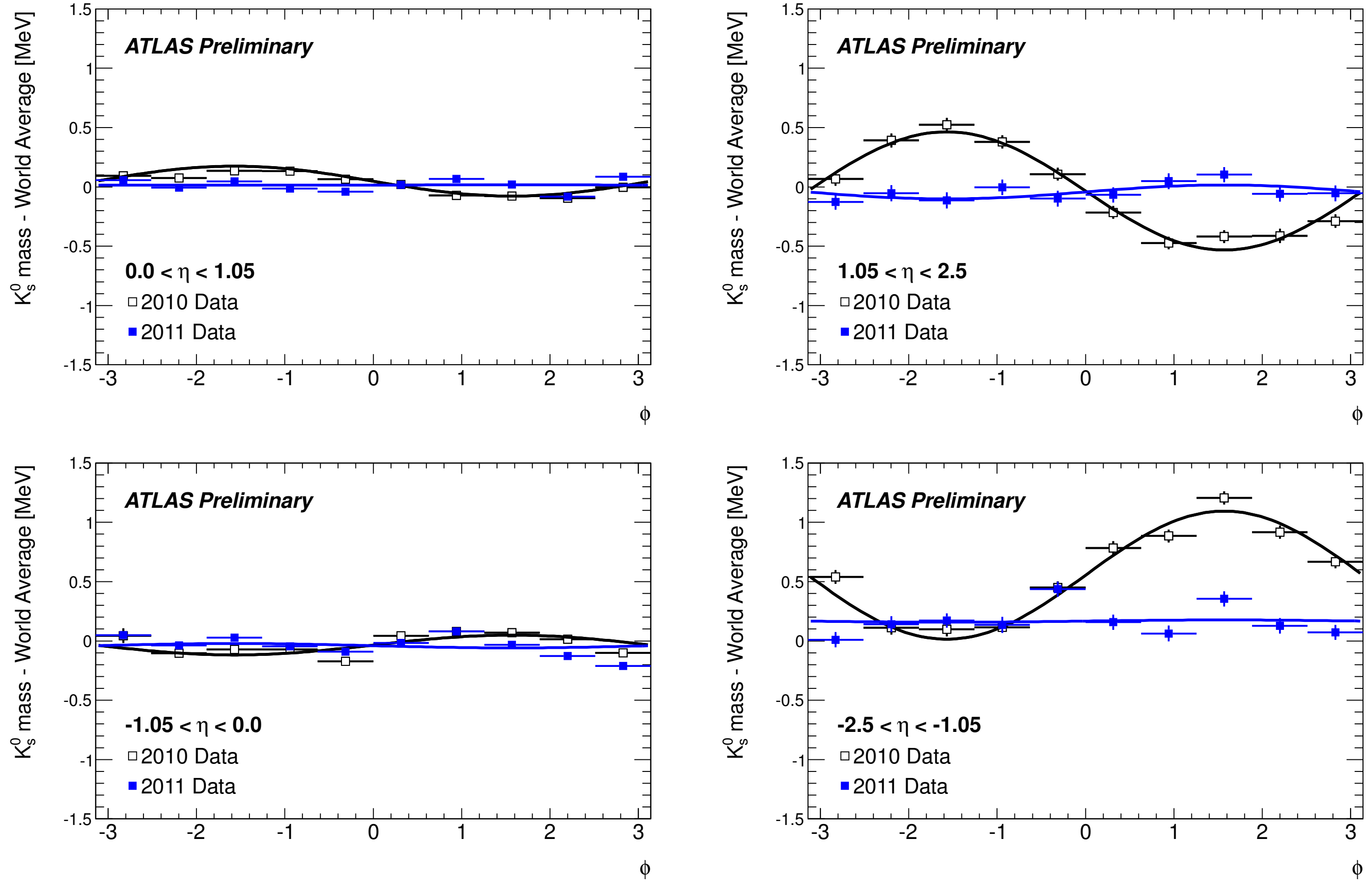


$$p \rightarrow p(1 - cot\theta sin\varphi \alpha_{rot})$$

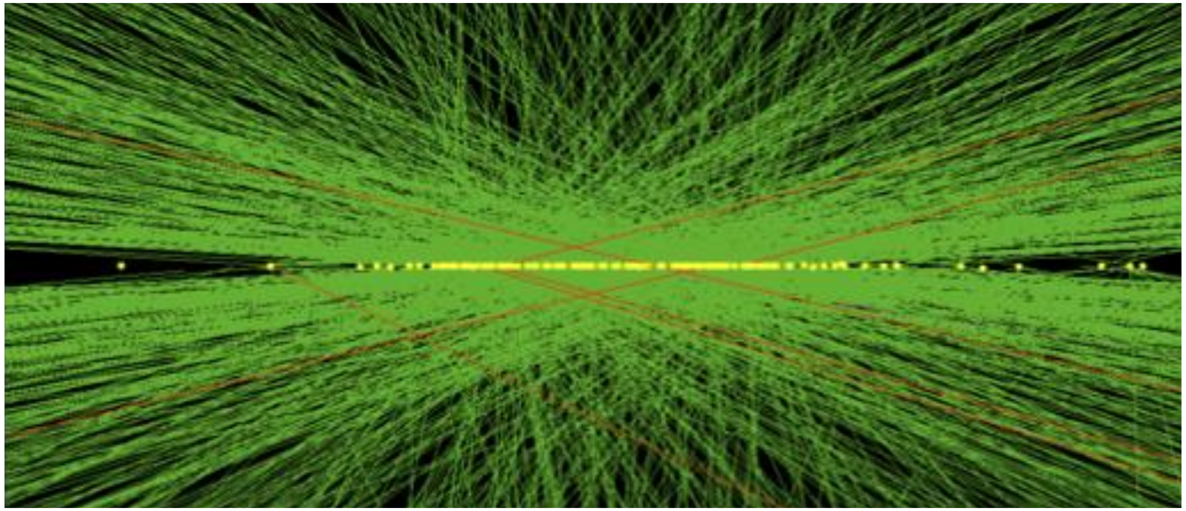
From the ATLAS Silicon alignment group, Bruckman et al

Correcting for B field rotation

- The needed extra constraint is here provided by the K0 mass.
- The rotations in data are found from interpolation among simulated rotated samples.

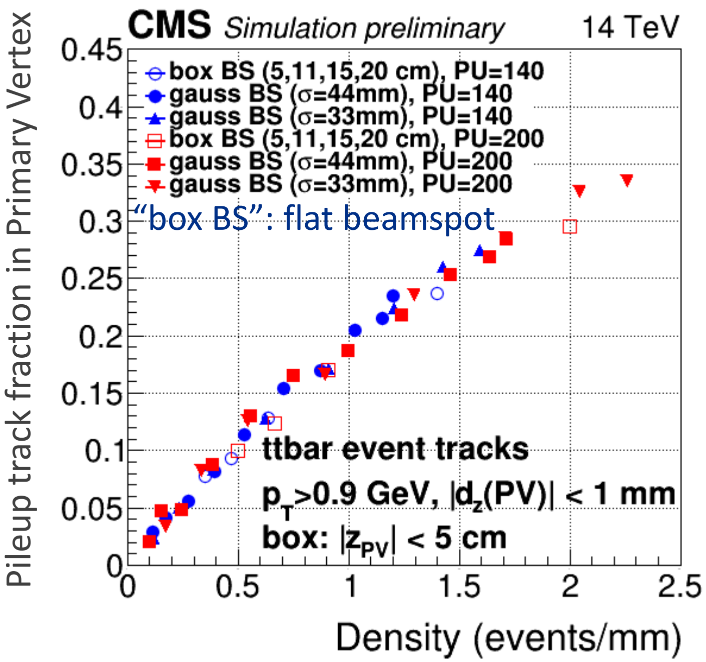


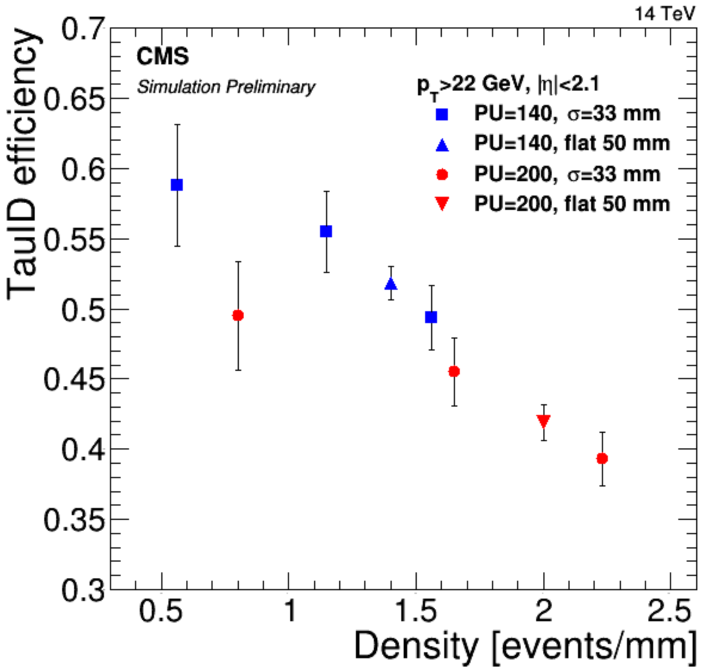
The pile-up challenge



Even with new high granularity trackers, the pile-up 200 at HL-LHC will significantly reduce physics reach.

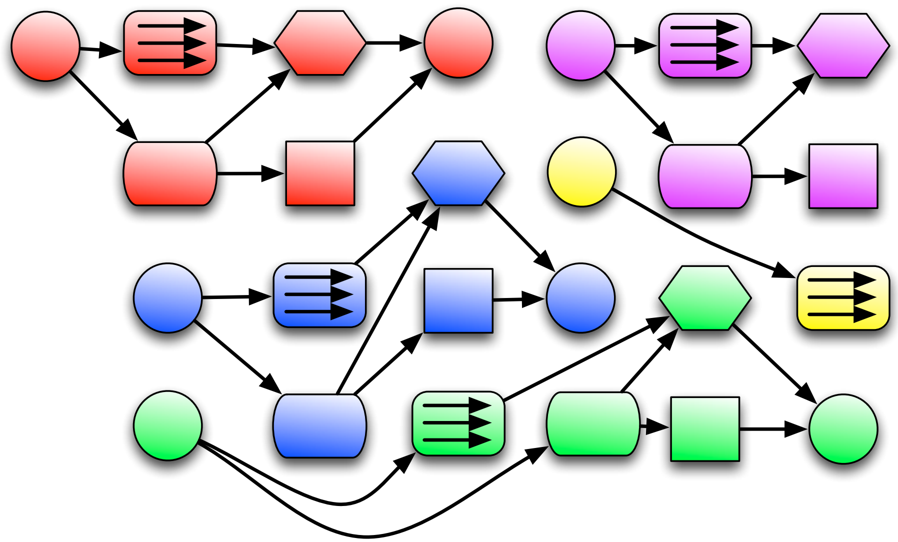
This goes both for performance and sheer processing capacity





Multithreading

- Because of power limitations, we are stuck with a clock speed of 3 GHz.
- However, the number of cores still grows (eg Xeon-Phi has 64 cores and 256 threads)
- The threads share memory and can be swapped in and out fast. This way you avoid idling and save memory.
- But all the software has to be changed so that shared services know who they are serving and respond accordingly

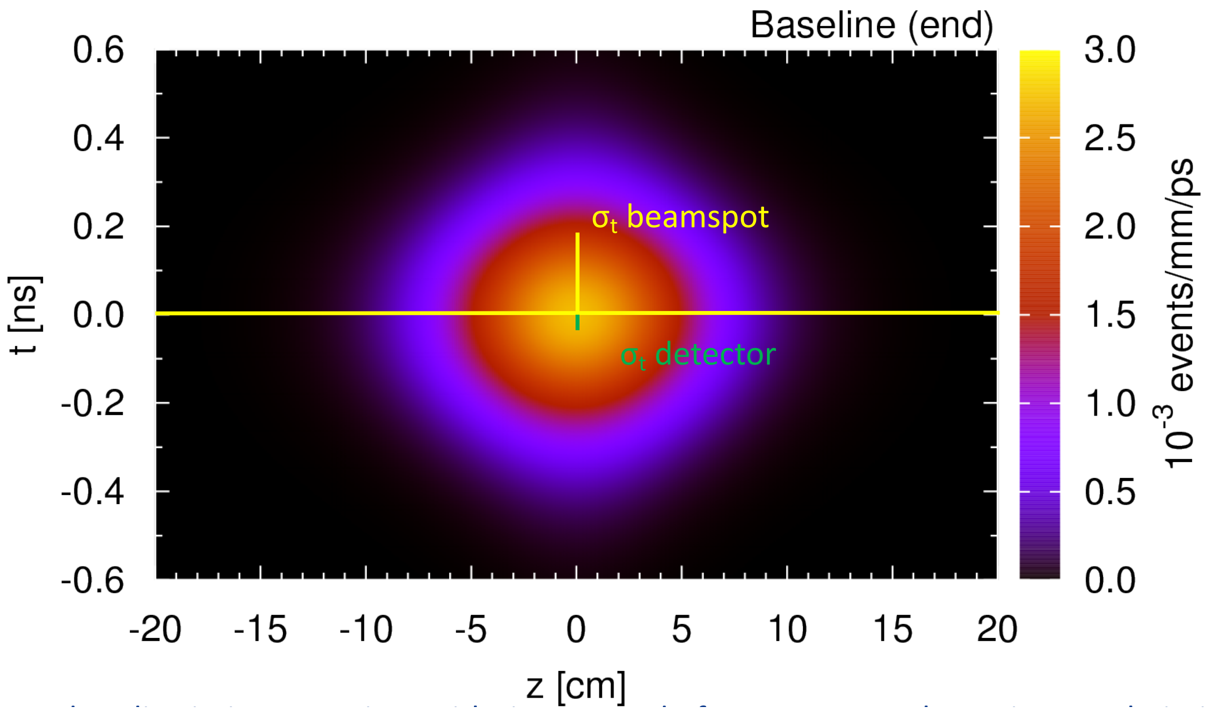


Cartoon of event processing
- each event is a different colour
- each shape is a different algorithm

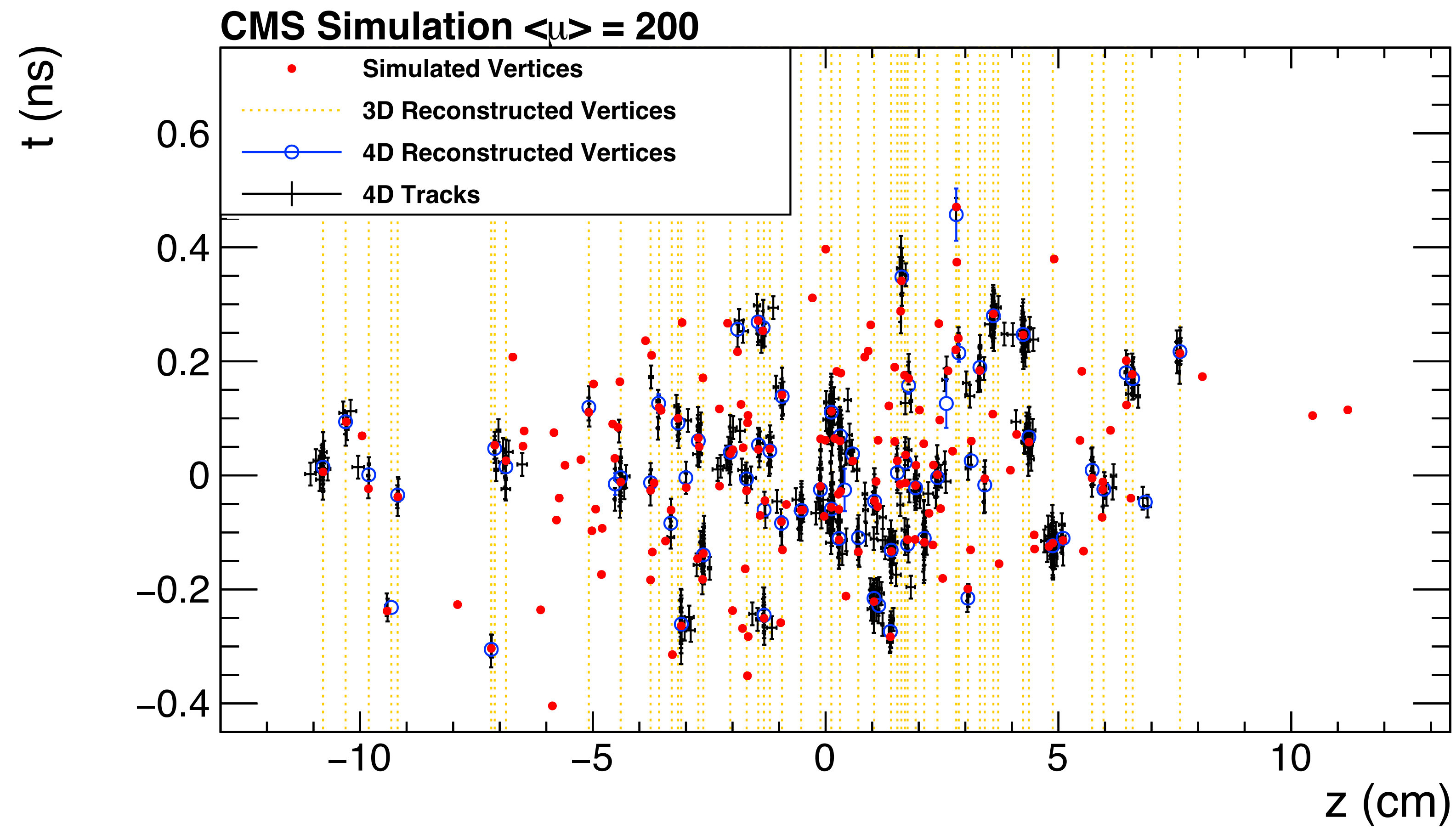
New timing layers for HL-LHC

- CMS: enclose entire tracker.
- Barrel: Thin crystals with SiPM, σ_t = 30ps
- Endcap: Low Gain Avalance Dev 3mm^2, 30-50ps
- ATLAS: 2.4 < η < 4.2. LGAD 1mm^2, σ_t = 30-50ps

Low gain -> thin detector -> high dV/dt -> good timing



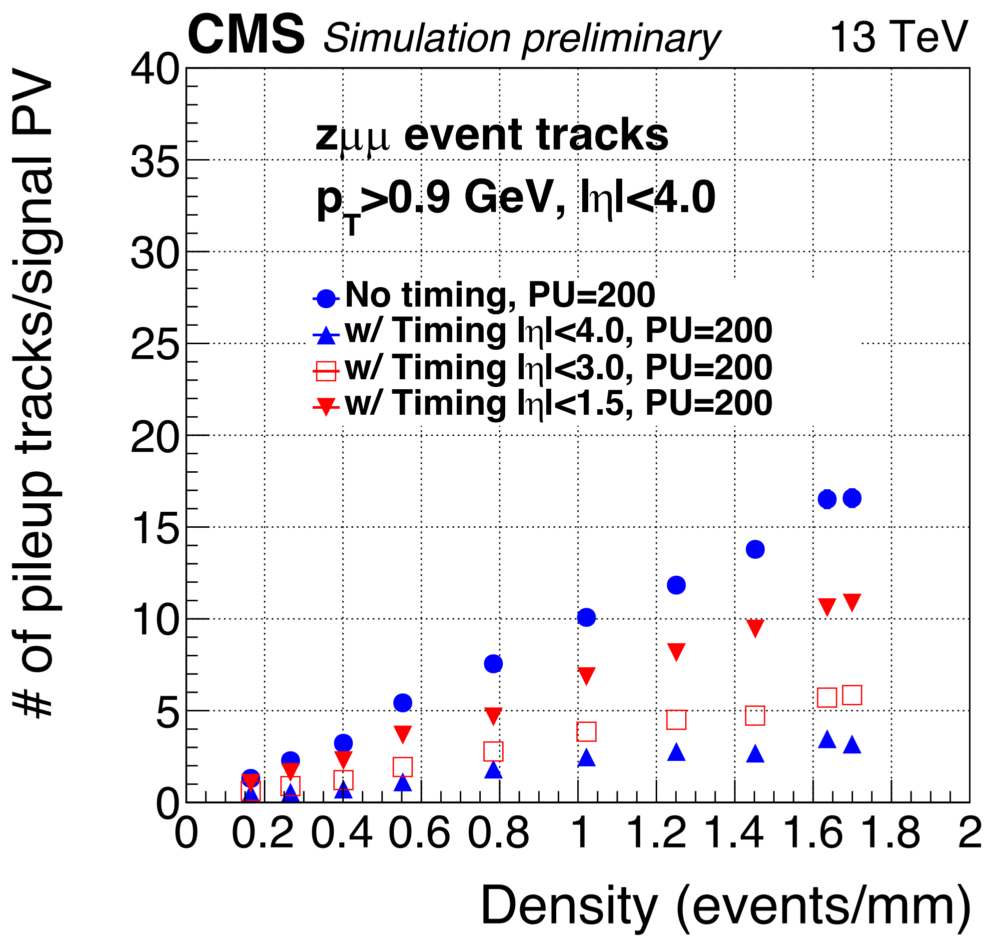
Timing in reconstruction

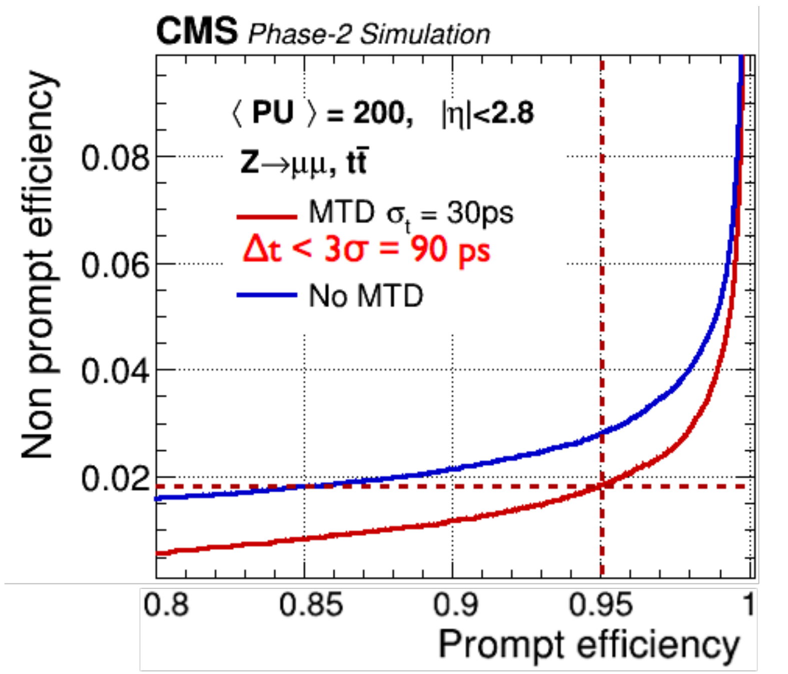


Great reduction of confusion. Everything benefits: isolation, MET, b-tag…

Timing info can be included in the Kalman smoother step

Timing in analysis

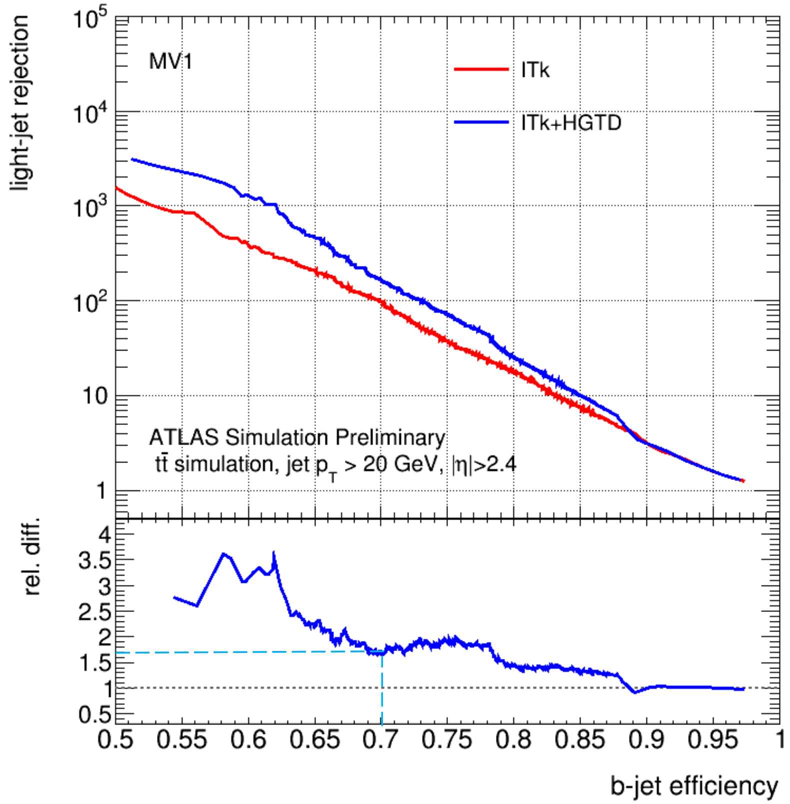


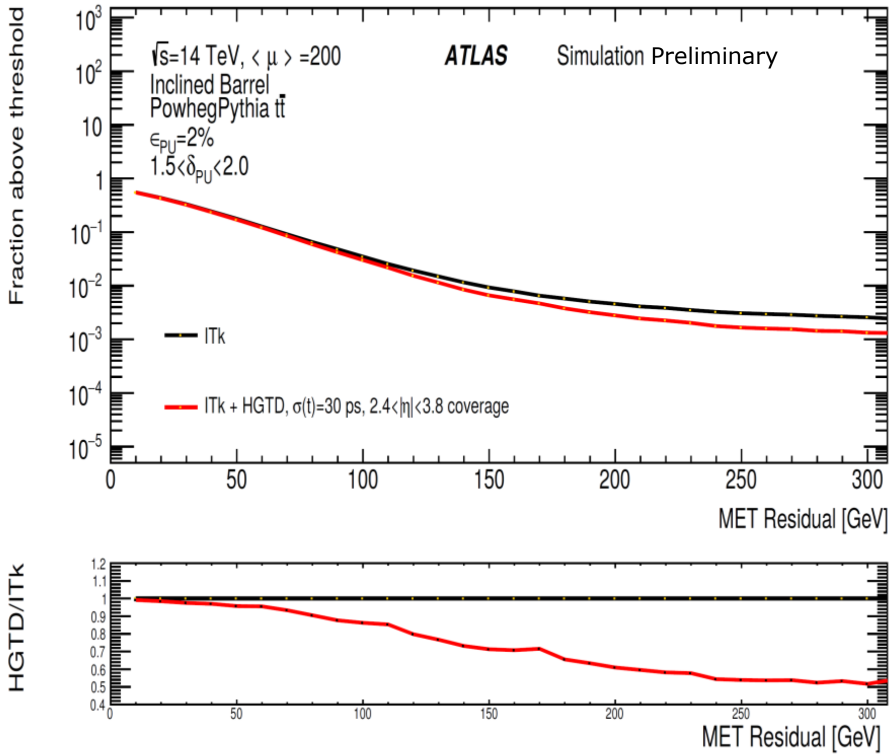


Increase in efficiency of isolated objects.
Benefit seems to scale with timing coverage.

Combine with calo-timing and get also better H->γγ vertexing

Timing in analysis





ATLAS forward light jet rejection 70% improved

Significant MET improvements.

New possibilities for heavy stable particle searches.

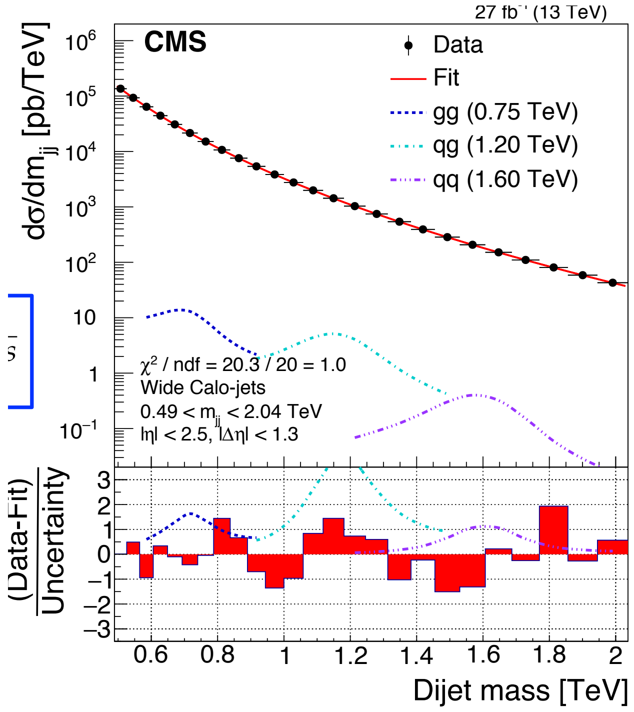
Data scouting

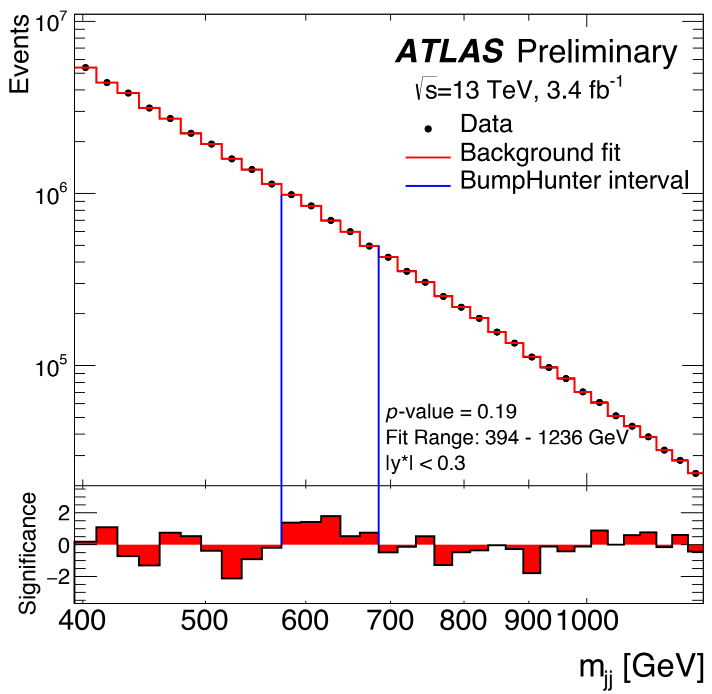
Even with the planned detector upgrades, the increasing luminosity will force higher trigger thresholds, especially for jet based triggers.

This may cause us to miss possible new physics with weak couplings.

Already now, all calibration and alignment has moved upstream to online HLT and calibration farms.

Trend at LHC is to also move suitable analysis to the trigger level:



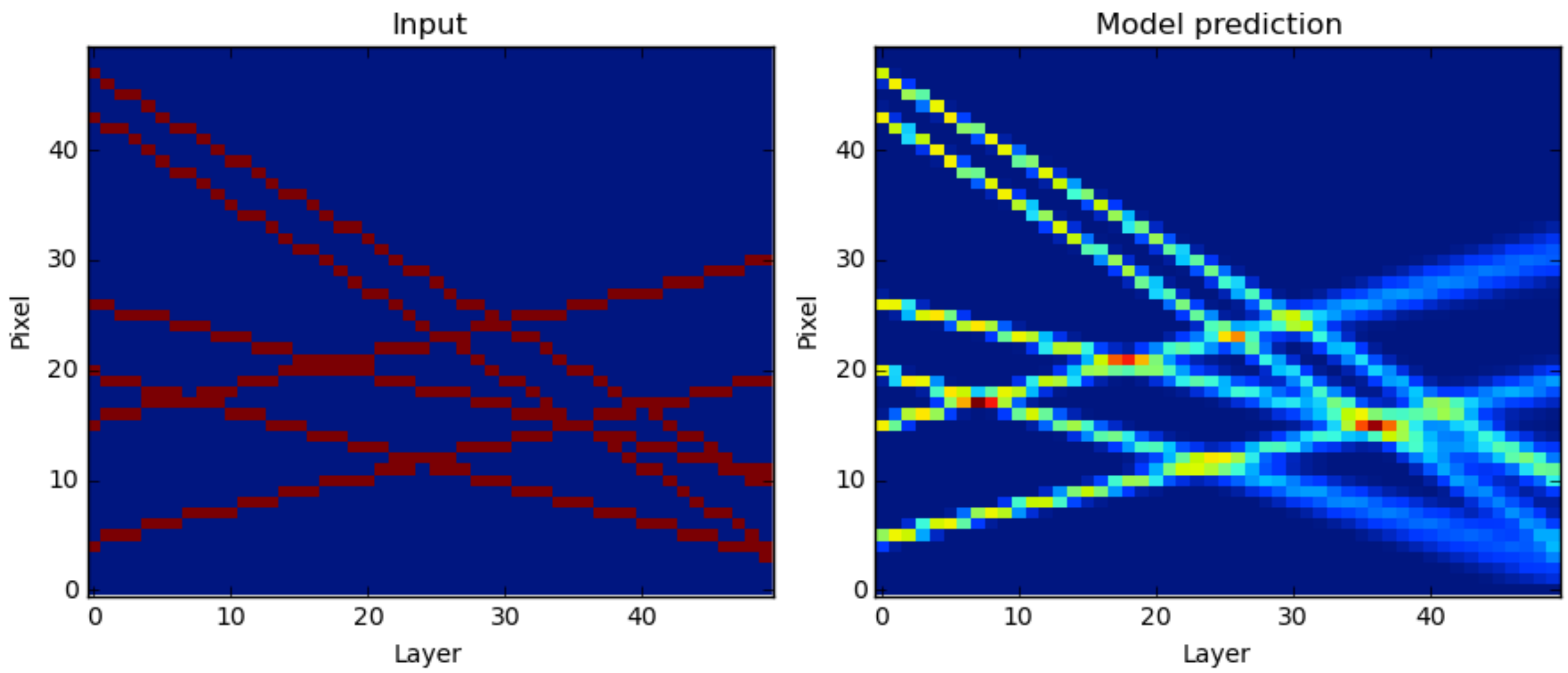


Machine learning in tracking

HEP.TrkX is a US based initiative to develop generic machine learning algorithms for track reconstruction.

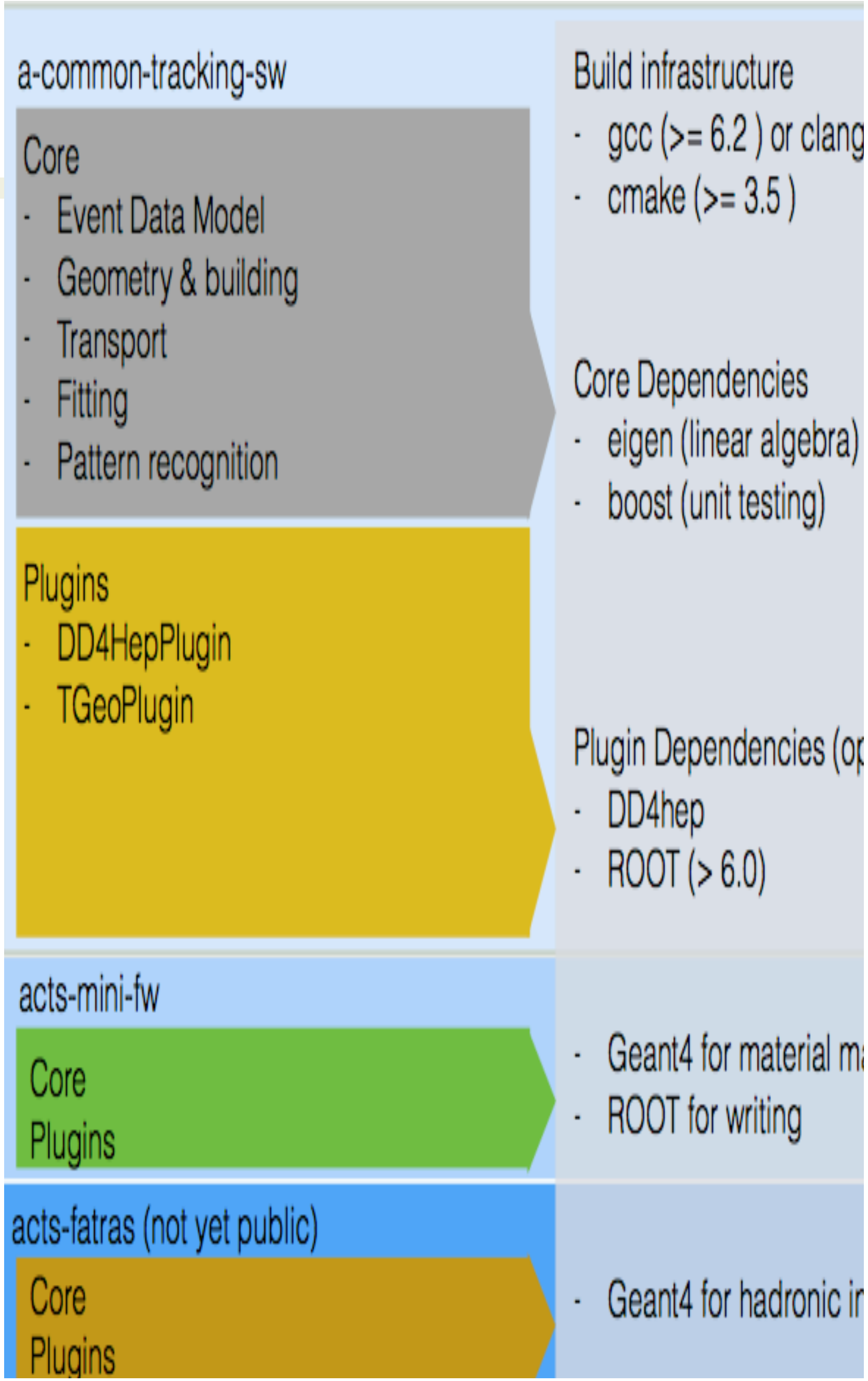
An initial study shows the potential of a recurrent (LSTM) and a convolutional (CNN) network and the combination of the two.
The LSTM acts like a fast Kalman Filter and the CNN is good for parameter estimation

EPJ Web of Conferences **150**, 00003 (2017)
Connecting The Dots/Intelligent Trackers 2017



Tracking toolbox

ACTS is an initiative (mainly by ATLAS people) to provide a generic toolbox of tracking algorithms for future experiments



http://acts.web.cern.ch/ACTS/

https://gitlab.cern.ch/acts/a-common-tracking-sw

https://its.cern.ch/jira/projects/ACTS/summary

Summary of particle tracking

- Precise spacepoints is the most important thing!
- Need to be continously calibrated and aligned – online.
- Trigger level tracking uses pre-fabricated *templates* or Hough transforms.
- Fits find the max likelihood track and vertex states. The Kalman Filter, Gaussian Sum Filter and chisquared minimization are the standard methods.
- Further refinements are possible using global methods.
- In future high lumi scenarios, the increase in combinatorics will be mitigated by adding a time-dimension.