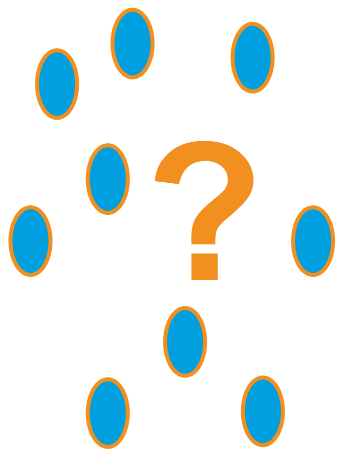
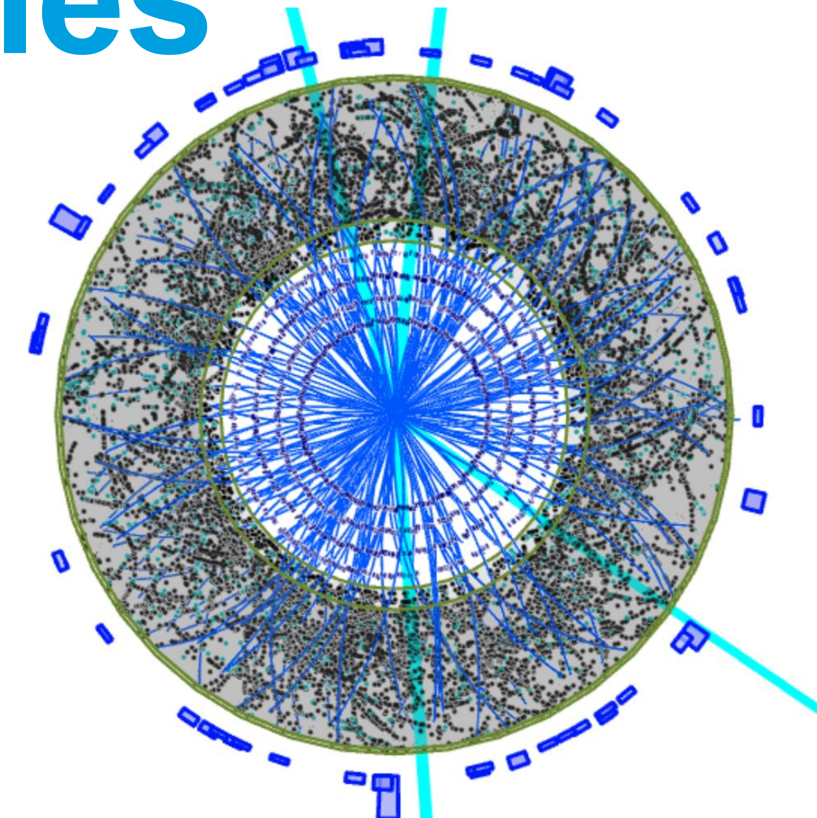
Tracking: Basic Principles

Or: How to Connect the Dots





Nick Styles, DESY
EDIT 2020
Hamburg, 17.02.2020

EDIT 2020

HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

Introduction

A few words before we start

Setting the context

This lecture is intended to give an overview of important concepts in track reconstruction

- Have tried to keep the level of mathematics I show explicitly small
- Only when it is useful/necessary for the conceptual understanding
- However this mathematics is clearly very important to understand when implementing or applying any of the methods discussed
- Will provide links at the end to places where complete and rigorous discussions of the mathematical underpinnings are discussed
- With that out of the way… let's begin!

Goals of Track Reconstruction

The “Why” before the “How”...

Why do we want to know about charged particles?

- They are a crucial aspect of a lot of Physics processes we want to study!
  - Large fraction of total momentum in collider events carried by charged particles
  - Many interesting final states are composed of charged particles
  - photons convert to charged e^+/e^- in material
  - etc...
- They have very useful properties as a “laboratory tool”
  - They can be steered by a magnetic field
  - Their properties can be determined via non-destructive measurements

Goals of Track Reconstruction

The "What" before the "How"...

What is it do we want to know about charged particles?

- Essentially we want to know their trajectory
- No magnetic field => Straight line!
  - Can compare where we expect them to go with where they actually go
  - Do our measurements match our predictions?
- With (typically solenoidal, uniform along z) magnetic field => Helix!
  - From curvature of helix, we can infer the momentum
- Measure the "Impact Parameter" with respect to a specific reference plane
- Also other, more specialized measurements possible depending on choice of detector design and technology

Track Parameterization

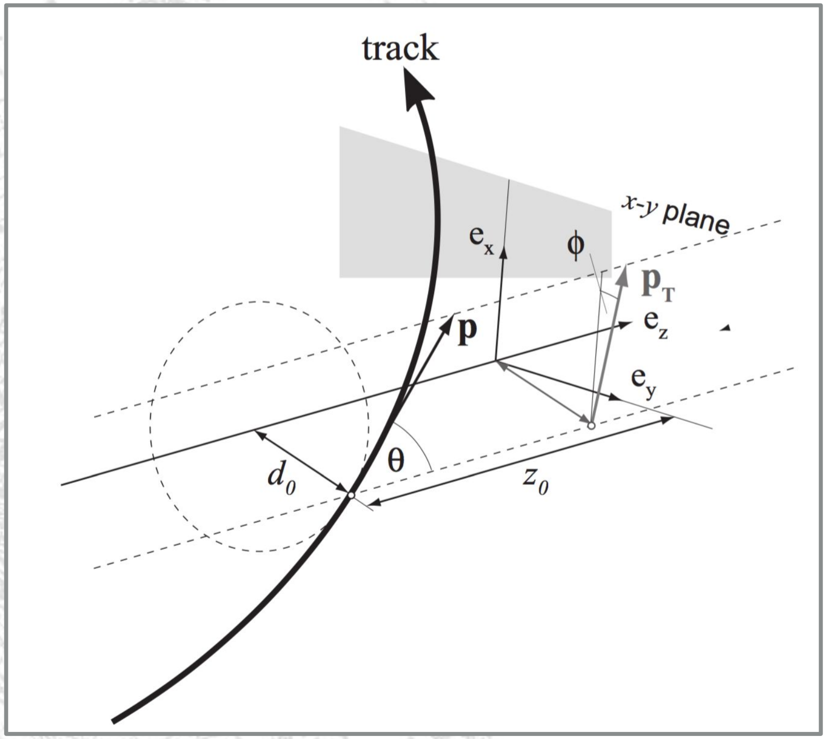
How to describe our Tracks

Need a way to encode the information about our trajectories

- This is our "Track Parameterization" - a typical parameterization with respect to a reference surface could be:

  $(d_0, z_0, \phi, \boldsymbol{\theta}, q/p)$

- This is a *special* version of this parameterization expressed on <u>perigee surface</u> (closest approach)
  - On this surface, first two parameters are transverse (d_0) and longitudinal (z_0) Impact Parameters
- Can also express at any "generic" surface
  - First two parameters become simply l_x and l_y - local coordinates on that surface



ϕ = azimuthal angle

$\boldsymbol{\theta}$ = polar angle

q/p = curvature*

*choose this rather than p itself, as errors are gaussian

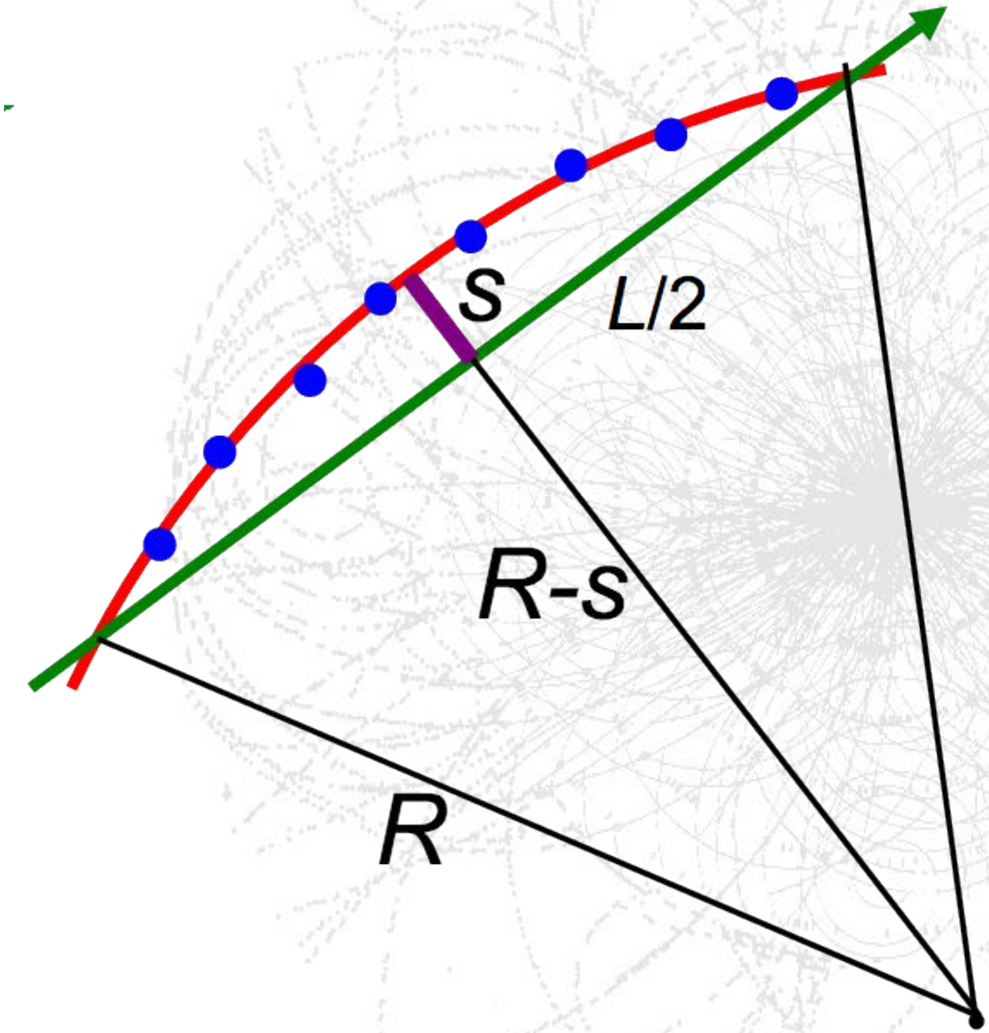
Track Parameterization

How to describe our Tracks

Need a way to encode the information about our trajectories

- Requiring knowledge about our surfaces is not always the most convenient…
- It is also possible to derive track parameterizations based on Curvilinear Coordinates - these are independent of any surface definition
  - For instance, (x, y, z, p_z, p_y, p_z)
- Typical just used as helpful “intermediate” format
  - Measurements in general will be with respect to a surface of some sort
  - Therefore predictions or expressions of “representative” track parameters are also typically in the same form
  - NB: This is assuming 3D tracking information - can of course be simplified for 2D case!

Momentum Resolution



Resolution:

$\delta p_T/p_T \propto \delta s/BL^2 \times p_T$

s is "sagitta", deviation from straight trajectory

p_T is momentum in transverse (bending) plane

B is magnetic field

From equations of motion of particle in Uniform B field:

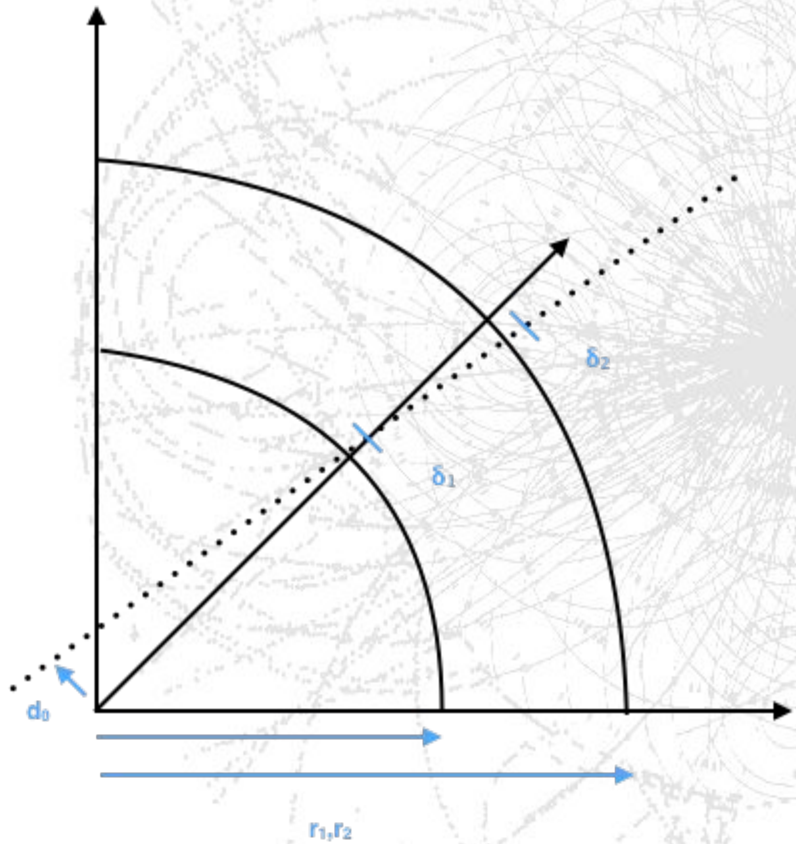
p_T [GeV/c] = 0.3 x B [T] x R [m]

$R \approxeq L^2/2s$

This gives us:

$\delta p_T/p_T = 8p_T/0.3BL^2 \times \delta s$

Impact Parameter Resolution



In a simplified system with 2 measurements, with uncertainties δ:

$$\delta_{d0}^2 = (r_1^2\delta_2^2 + r_2^2\delta_1^2)/(r_2 - r_1)^2$$

Both this and momentum resolution become more complicated when faced with reality…

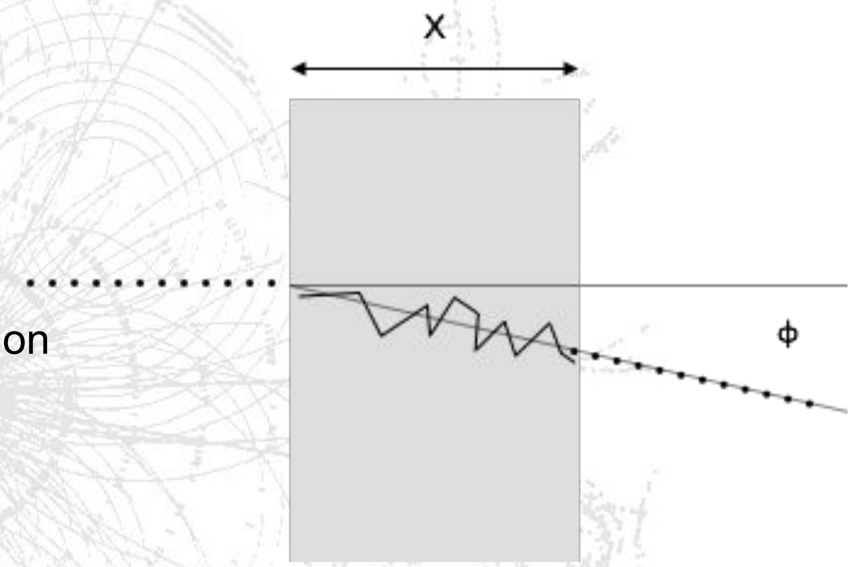
We'll revisit them later!

Multiple Scattering and Material Effects

Dealing with physical reality...

Particles traversing distance x/X0 through a material will undergo multiple Rutherford scattering-type interactions

- Random, stochastic process
  - Angular deflection of outgoing particle, θ_{MS}, follows an approximately gaussian distribution
  - Non-gaussian tail contribution ~2%, follows approximately $\sin^{-4}(\theta_{MS}/2)$ distribution
- Multiple Scattering contributions depends upon material properties and particle momentum (minimized at large momentum)



X_0 is "radiation length", characteristic property of material

$$\theta_{MS} = (13.6\ \mathrm{MeV}/\beta cp)\ z\ \sqrt{(x/X_0)}\ [1 + 0.0038\ln(x/X_0)]$$

Silicon has X_0 9.37 cm
Lead has X_0 0.5612 cm

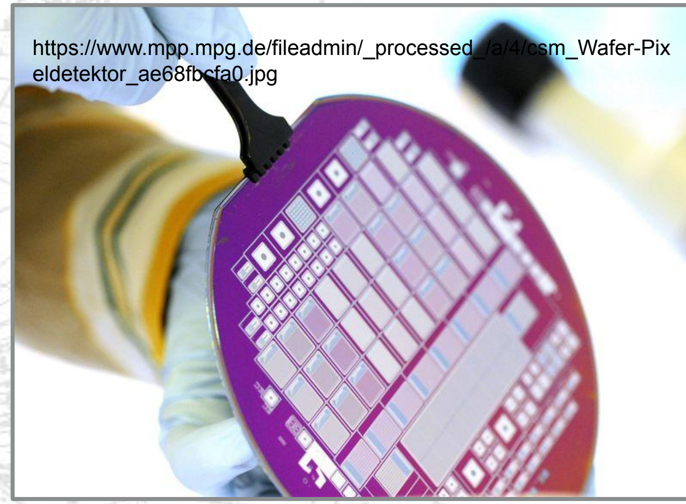
Data Preparation

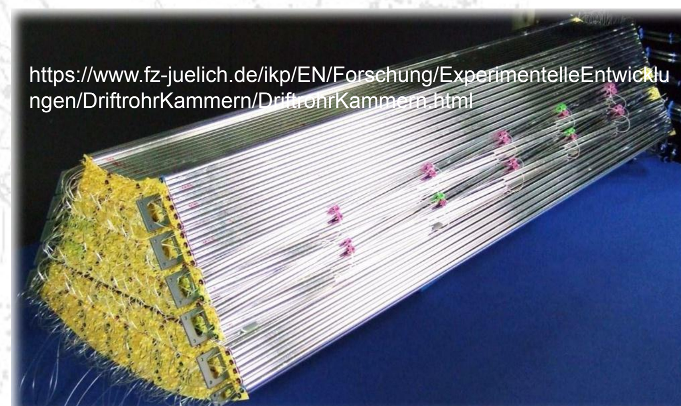
Measurements

What Type of Inputs can we use?

May have to deal with a few typical types of measurement

- Spatial measurements from highly-segmented semiconductor detectors
  - Segmentation in 1D (microstrip-type detectors) or 2D (pixel-type detectors)
  - Typically few measurements per track
- Drift time measurements from gaseous detector
  - Converted into distance of particle from “sense wire” - includes left/right ambiguity
  - Typically many measurements per track
- Functioning principles covered elsewhere ;-)

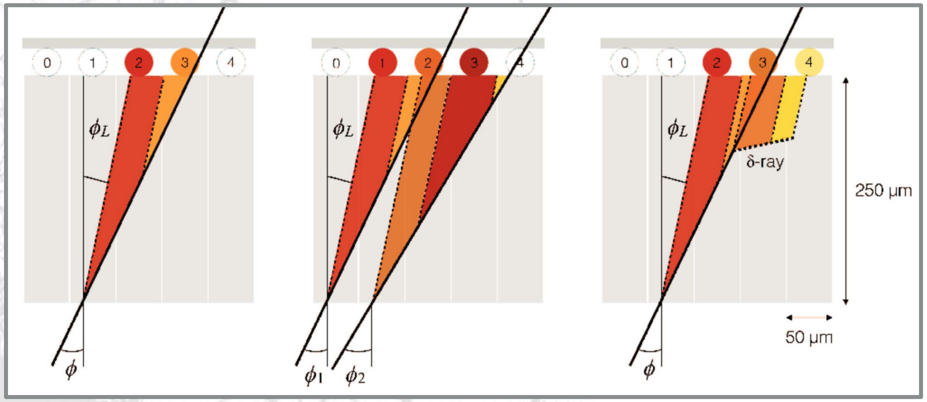




Clustering

From semiconductor detector outputs to tracking inputs

Single particle contribute charge to multiple detector channels



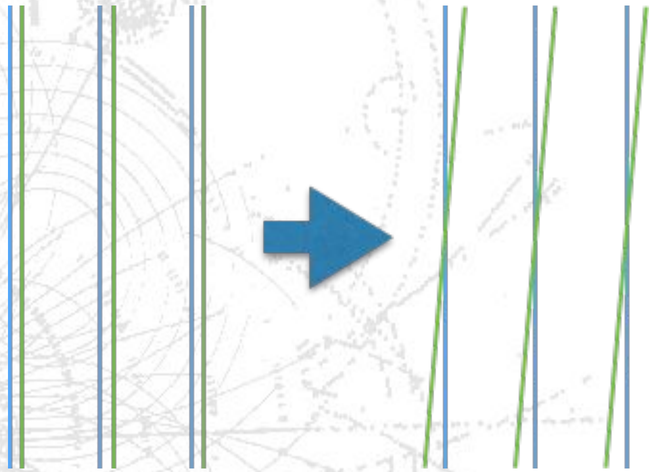
- Typically group channels with above-threshold charge deposits as a cluster
  - Effects of dead, noisy pixels, lorentz angles, must be accounted for
- Cluster information provides incident position estimate and uncertainties
  - Single channel resolution given by **pitch/√12**
- Information per channel can be digital ("on/off") or analogue (e.g. signal time over threshold)
  - The latter provides more information that can be used for calculating cluster "centre of gravity" => better position resolution

Further Data Preparation Techniques

Making the most of detector information

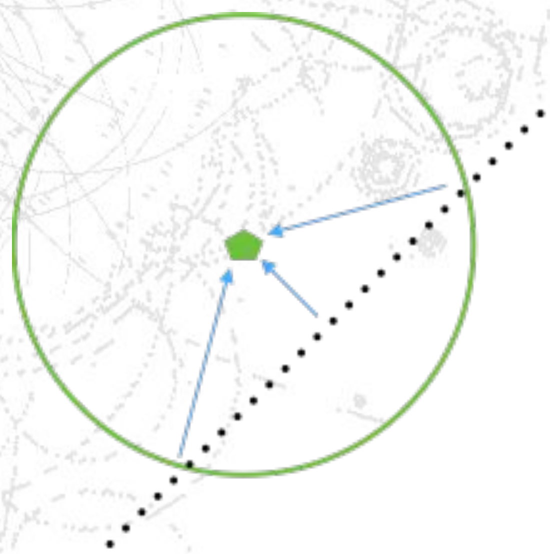
Stereo Angle Pairs

- Small rotation between pairs of strip sensors can improve precision in “long” direction
  - correlate which strip pair were hit
  - Caveat: Increasing stereo angle increases precision and rate of “Ghost Hits” (degenerate combinations)



Drift Circles

- Need to calibrate arrival times of charges to provide wire-to-track distance
  - Total amount of charge can also be used in some cases for particle Identification



Measurement Model

How to represent our measurements mathematically

$$m_k = h_k(q_k) + \gamma_k$$

$$H_k = \delta m_k / \delta q_k$$

measurement
track dependence model (e.g. on incident angle, etc)
track parameters (see later...)
error/noise term

Jacobian of track dependence model

$$G_k$$

measurement covariance

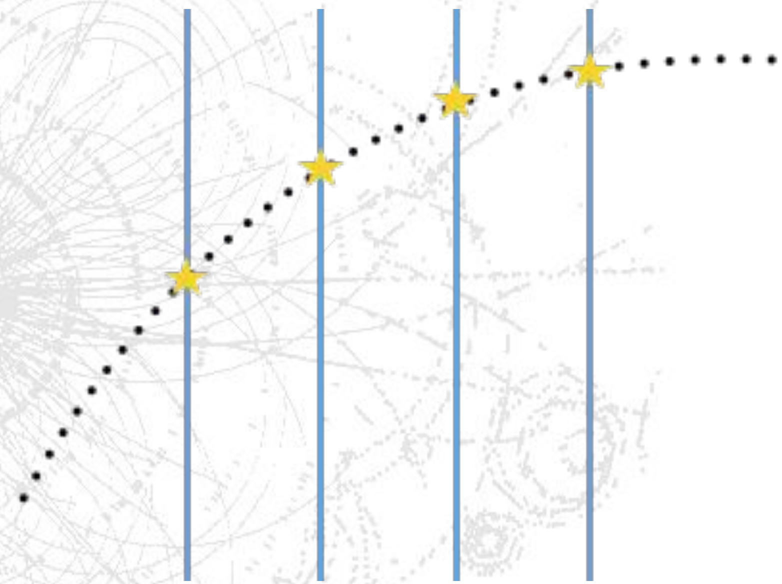
Finding Tracks

Let’s Start Simple

A toy example

In a trivial example, looks very easy to find a track… you need:

- Initial starting parameters
- Knowledge about detector layout
  - Where (e.g. which layer) to look for first/next hit
  - How much material is passed through
- A way to calculate Track Parameters and their uncertainties on the next surface
  - Often referred to as “Track Model”
- Simple! Well, let’s see...

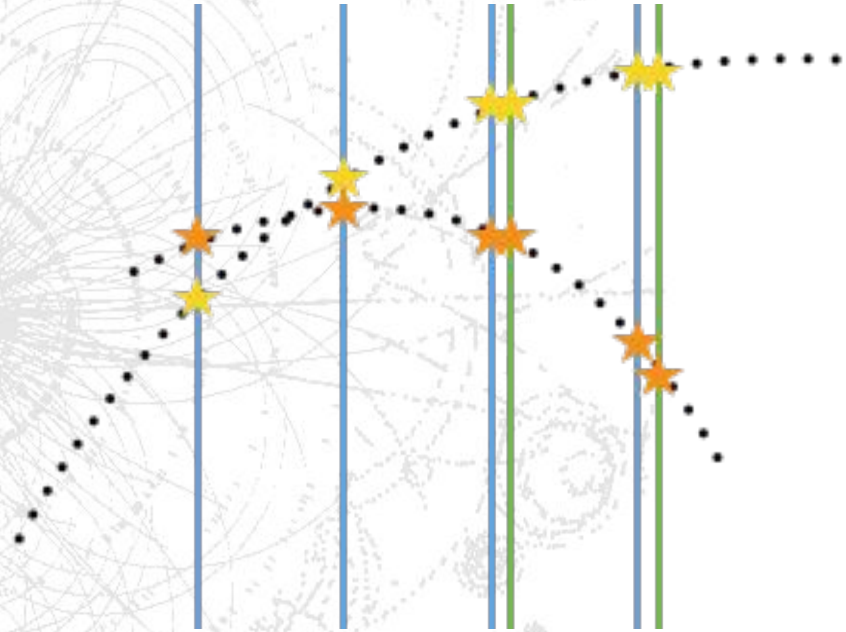


Let’s Start Simple

A toy example

Once things start getting more realistic...

- Initial starting parameters
  - Different choice of starting parameters can lead down completely different path
  - (Even in very low multiplicity scenarios can have noise, secondaries, etc)
  - Should aim to minimize attempts made down “wrong” paths
  - Use possible additional constraints from knowledge of physics, detector, initial particle distributions, etc to make sensible choices

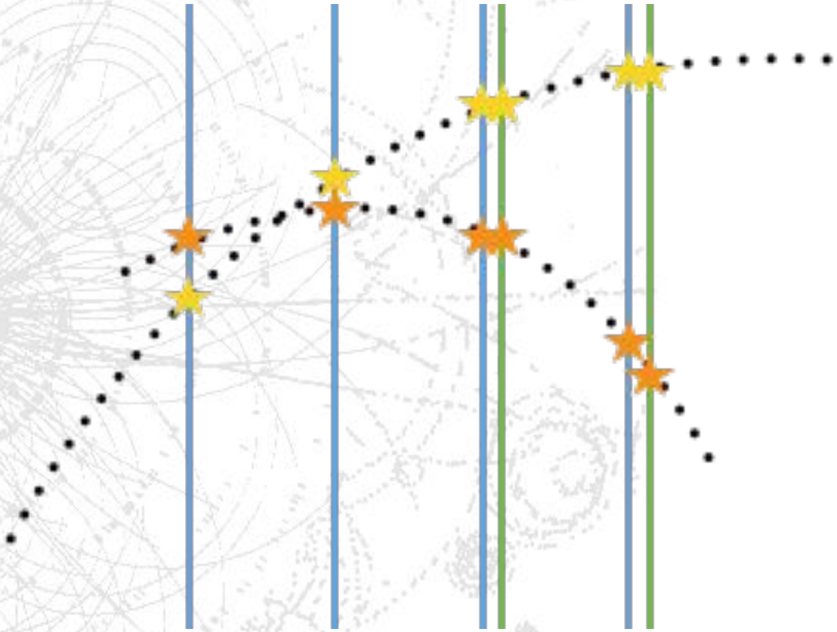


Let's Start Simple

A toy example

Once things start getting more realistic...

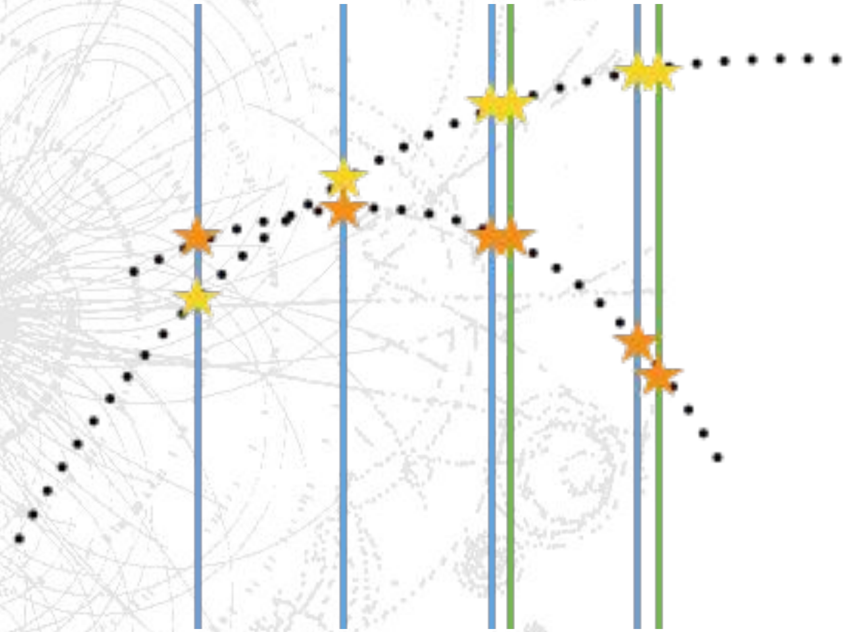
- Knowledge about detector layout
  - Where (e.g. which layer) to look for first/next hit
  - Different technologies per layer, barrel or endcap orientation (and transition between them), overlaps, tilt angles...
  - How much material is passed through
  - Very large local variations possible; need a way to store and retrieve information with appropriate granularity



Let’s Start Simple

A toy example

Once things start getting more realistic...

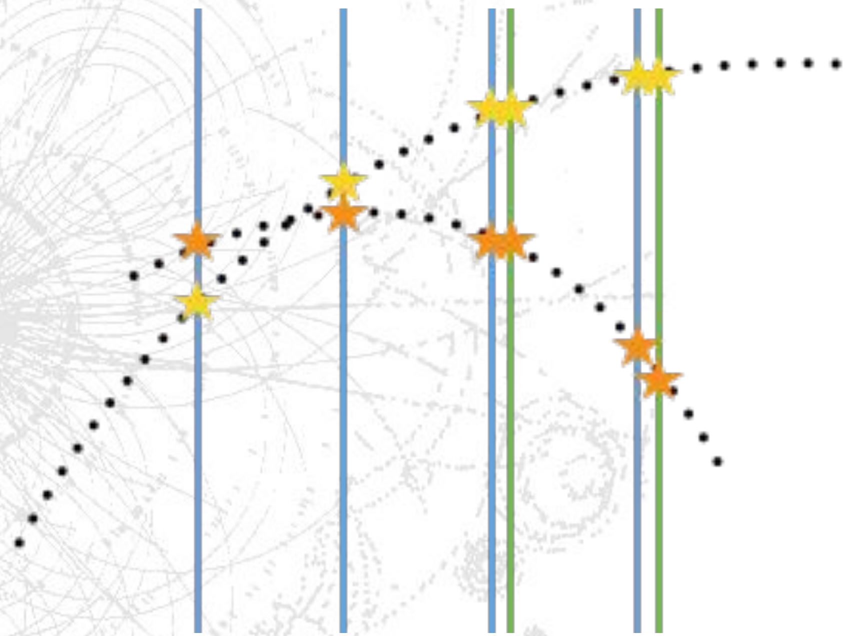


- A way to calculate Track Parameters and their uncertainties on the next surface
  - Often referred to as “Track Model”
  - With non-constant magnetic field, no analytic solution! Need to use numerical methods.

Let’s Start Simple

A toy example

Once things start getting more realistic...



- Simple! Well, let’s see…
  - Not so much!

Pattern Recognition

How to separate the “real” tracks from “everything else”

A realistic picture starts to look much more tricky...

- Compared to a toy situation, a hadron collider type event is very different
- By eye, seems impossible to find tracks in it...
  - Fortunately, we have algorithms that can do this very well!

Aside: Looks like a task for Machine Learning!

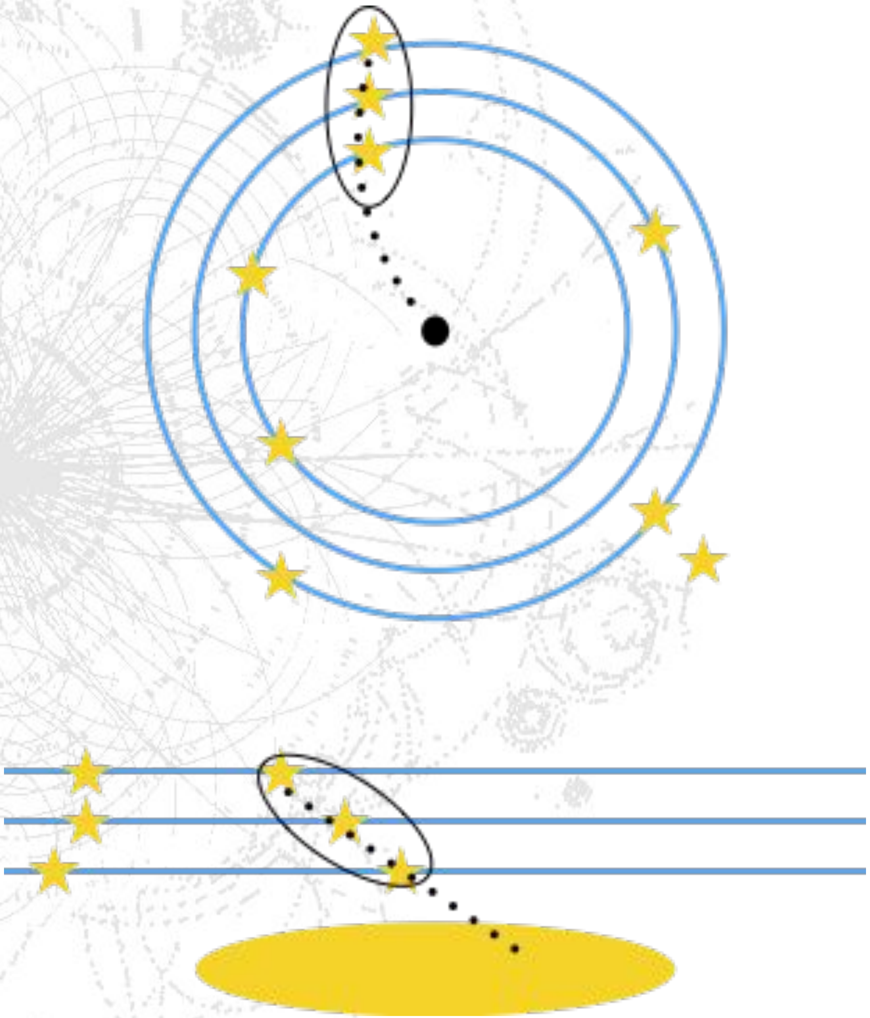
- As you can guess, people have been trying this!
  - Kaggle Tracking Machine Learning Challenge
- Overall not (yet) competitive with the tried-and-tested approaches present here
  - Something for the future, not so much for today

Local Approaches

"Following" a Track

"Seeding" the track

- Typical first step is to create track seeds
  - Group small number of compatible measurements
  - Provides initial rough estimate of track parameters
- Typically many more seeds than final tracks expected
  - Use knowledge about detector geometry, event topology, etc to reject "impossible" combinations as early as possible
  - In high multiplicity situations, book-keeping of hits may be needed



Local Approaches

"Following" a Track

Next step: Collecting compatible measurements along possible trajectories

- General procedure - look on next layer for hits
  - E.g. "hit road" based approach, propagate track parameters onto possible surfaces and check for hits
  - Various ways of deciding what is a "compatible" hit (is it on the expected sensor, does it pass a χ^2 criteria, etc…)
- May be multiple possibilities for compatible hits!
  - In this case, can either take "best one" or do a "combinatorial" approach - branch your track, and collect further hits according to both options
  - In latter case, will have more options later to choose between, but more "costly"
- Keep going until you reach the end of your detector
  - Congratulations, you now have a candidate track!

Kalman Filter

Progressive State Updates

Commonly-used method for estimating states of dynamic systems

- Combines predictions (based on underlying model and knowledge of prior state) and measurements to <u>provide more accurate state estimate than either individually</u>
  - Predictions alone accumulate increasingly large uncertainties due to stochastic processes along trajectory (multiple scattering, etc)
  - Measurements alone are “noisy”
- Nice feature: Need only the state estimate at prior step to have full information needed for the next step!
  - No need to keep track of full history; it is “encoded” in the state estimate plus its covariance
- “Real world” example: Combine telemetry data on thrust with GPS position to estimate the true position and velocity of a projectile

Kalman Filter

Progressive State Updates

$q_k = f_{k\|i}(q_i)$	$C_k = F_{k\|i} C_i F_{k\|i}^T$	$F_{k\|i} = \delta q_k / \delta q_i$

track states
track model

track states covariance
track model Jacobian

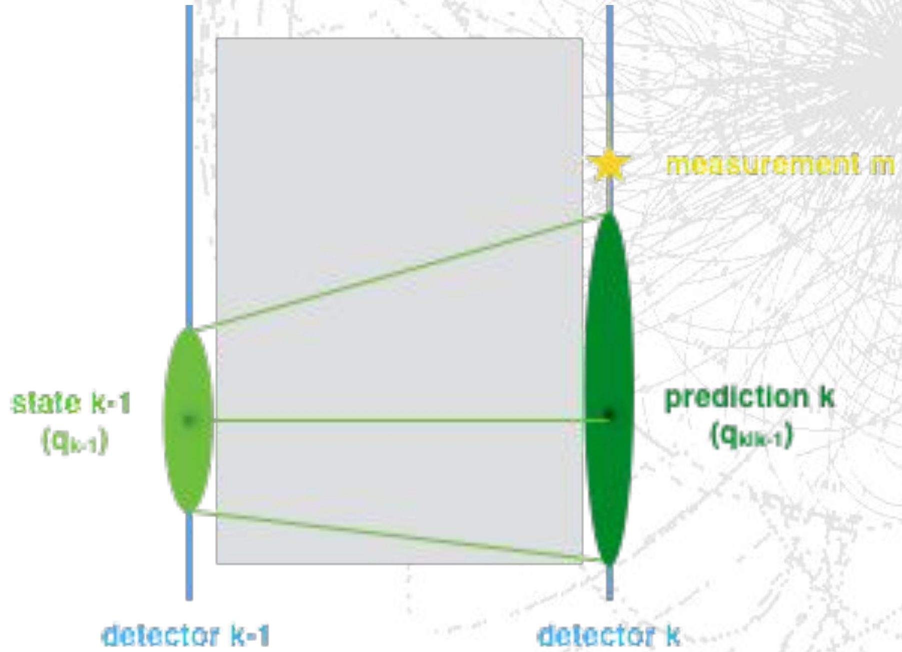
Kalman Filter

Progressive State Updates

$q_k = f_{k\|i}(q_i)$	$C_k = F_{k\|i} C_i F_{k\|i}^T$	$F_{k\|i} = \delta q_k / \delta q_i$

track states
track model

track states covariance
track model Jacobian



propagate prior state (q_{k-1}) onto next detector (k):

$$q_{k|k-1} = f_{k|k-1}(q_{k-1})$$

$$C_{k|k-1} = F_{k|k-1} C_{k-1} F_{k|k-1}^T + Q_k$$

Q_k is stochastic contribution (e.g. from Multiple Scattering)

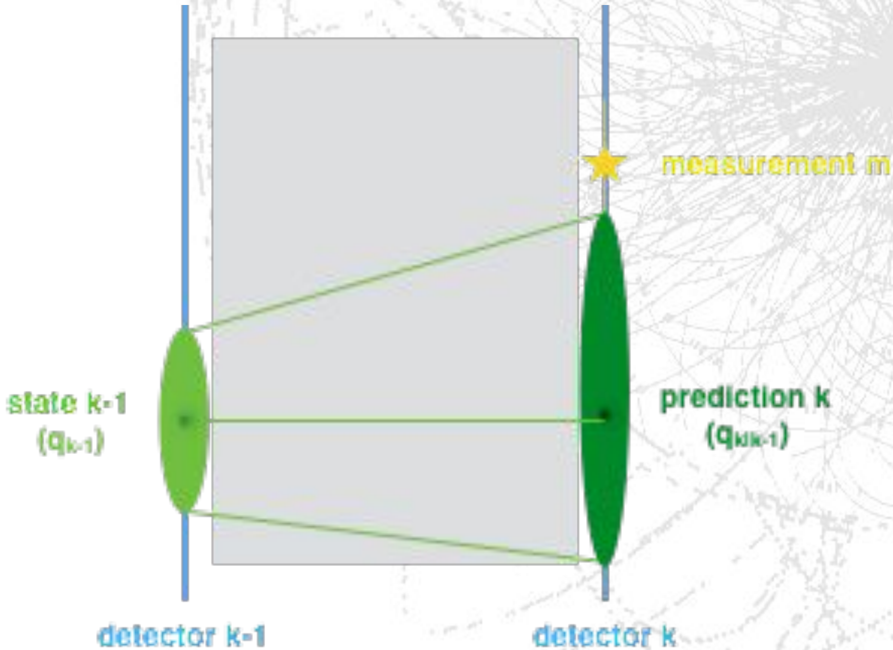
Kalman Filter

Progressive State Updates

$q_k = f_{k\|i}(q_i)$	$C_k = F_{k\|i} C_i F_{k\|i}^T$	$F_{k\|i} = \delta q_k / \delta q_i$

track states
track model

track states covariance
track model Jacobian



Gain matrix defines the combination of prediction with measurement:

$$K_k = C_{k|k-1} H_k^T (G_k + H_k C_{k|k-1} H_k^T)^{-1}$$

(Could be replaced by weighted mean)

Kalman Filter

Progressive State Updates

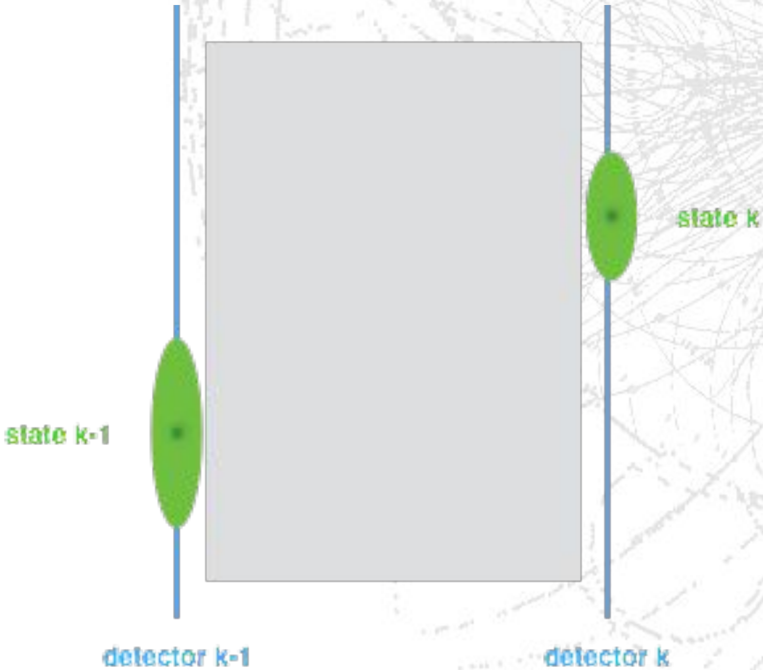
$$q_k = f_{k|i}(q_i)$$

$$C_k = F_{k|i} C_i F_{k|i}^T$$

$$F_{k|i} = \delta q_k / \delta q_i$$

track states
track model

track states covariance
track model Jacobian



Update prediction to get final parameter estimate q_k and C_k

$$q_k = q_{k|k-1} + K_k[m_k - h_k(q_{k|k-1})]$$

$$C_k = (I - K_k H_k) C_{k|k-1}$$

Repeat the procedure starting from q_k to get get q_{k+1}, and so on...

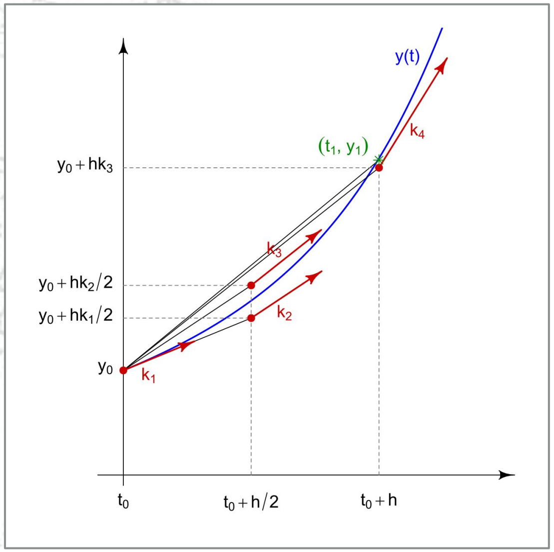
Propagating Parameters

Track Model and Extrapolation

- When extrapolating track parameters must account for multiple scattering effects on particle trajectory (increases direction uncertainty), but also energy loss due to material interactions (impacts curvature)

$$-\left\langle \frac{dE}{dx} \right\rangle = \frac{4\pi}{m_e c^2} \cdot \frac{nz^2}{\beta^2} \cdot \left(\frac{e^2}{4\pi\varepsilon_0} \right)^2 \cdot \left[\ln\left(\frac{2m_e c^2 \beta^2}{I \cdot (1-\beta^2)} \right) - \beta^2 \right]$$

- Must also account for magnetic field
  - d**p**/dt = q**v** x **B**
  - For a uniform field, simply use helix model
  - As mentioned earlier, no analytical solution in case of non-constant B-field
  - Estimate typically obtained via Runge-Kutta methods (or Runge-Kutta-Nyström)
  - Can be computationally expensive! Step size needs to be set carefully to an appropriate value for the application and conditions



from wikipedia (HilberTraum)

Global Approaches

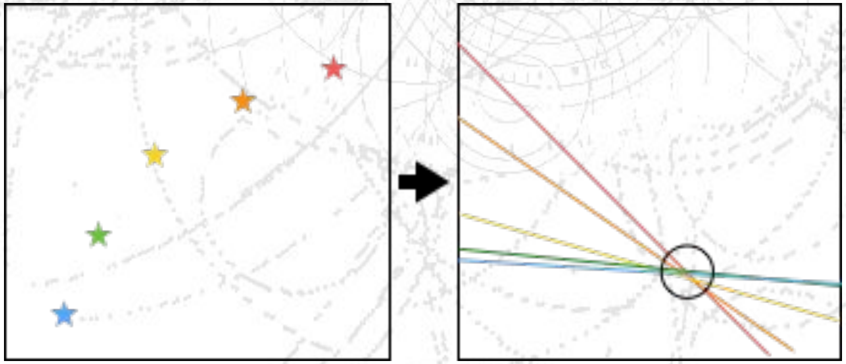
Looking at the whole picture...

Can use similar approaches to “feature extraction” in image processing

- Transform measurements into a “parameter space” allowing parameters to be found by simple maxima search (e.g. with histogramming methods)
  - e.g. Hough Transform, where hits become straight lines in **u,v** space
  - Initially developed for extracting tracks from bubble chamber images

$$\mathbf{u = x/(x^2+y^2) \quad v = y/(x^2+y^2) \Rightarrow v = -(x/y)u + (x^2+y^2/2y)}$$

- Particularly well suited for 2D tracking with many measurements
  - E.g. drift tube based detectors



Global Approaches

Looking at the whole picture...

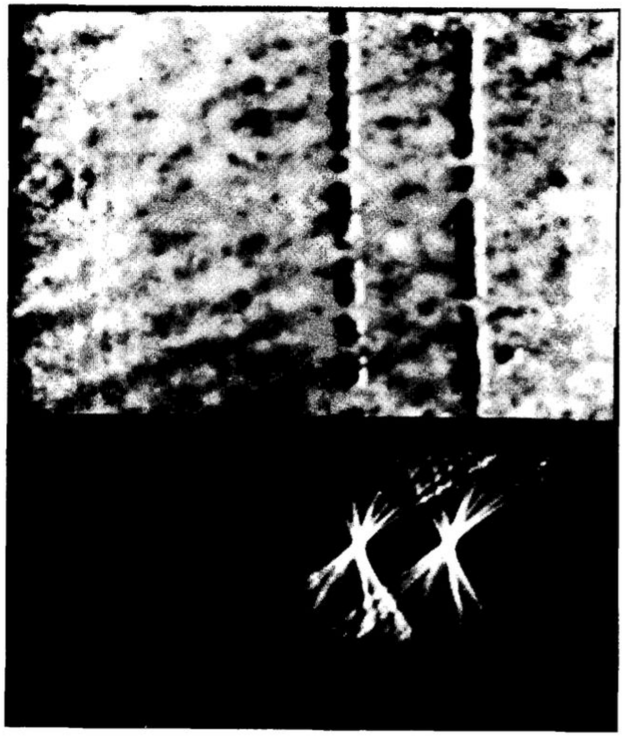


Fig. 2 A framelet giving a simple bubble pattern. The white pattern shows the portion of the bubble pattern detected by the electronics. The transform of the white pattern, drawn electronically, appears at the bottom.

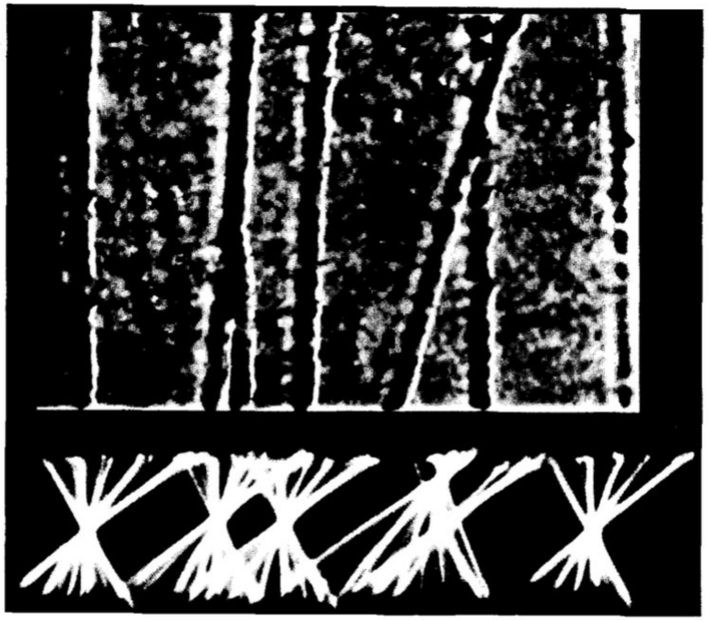


Fig. 3 A framelet giving a reasonably complex bubble pattern. The electronically-drawn transform appears at the bottom.

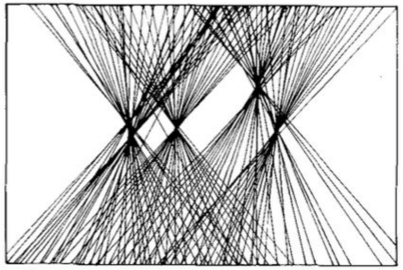


Fig. 4 A hand-drawn transform of the white pattern of Fig. 3. (The extreme right and extreme left tracks are not transformed.)

from: Machine Analysis of Bubble Chamber Pictures

P.V.C. Hough (Michigan U.)
Sep 1959

Fitting Tracks

Fitting Tracks

Now that you've found it...

Finding the the measurements belonging to a track is not the end of the story!

- May be advantageous to make simplifications during track finding
  - E.g. to allow early rejection for anything that is "not interesting" or not meeting some basic quality requirements
- May need to resolve "competing claims" on measurements between multiple track candidates before final hit content is known
- In such cases, a further step (generally referred to as fitting) is required to give best estimate of track parameters
  - Both at each measurement surface…
  - ...and also at any representative/defining surface, such as the Perigee

Kalman-Based Fit

Already half-way there….

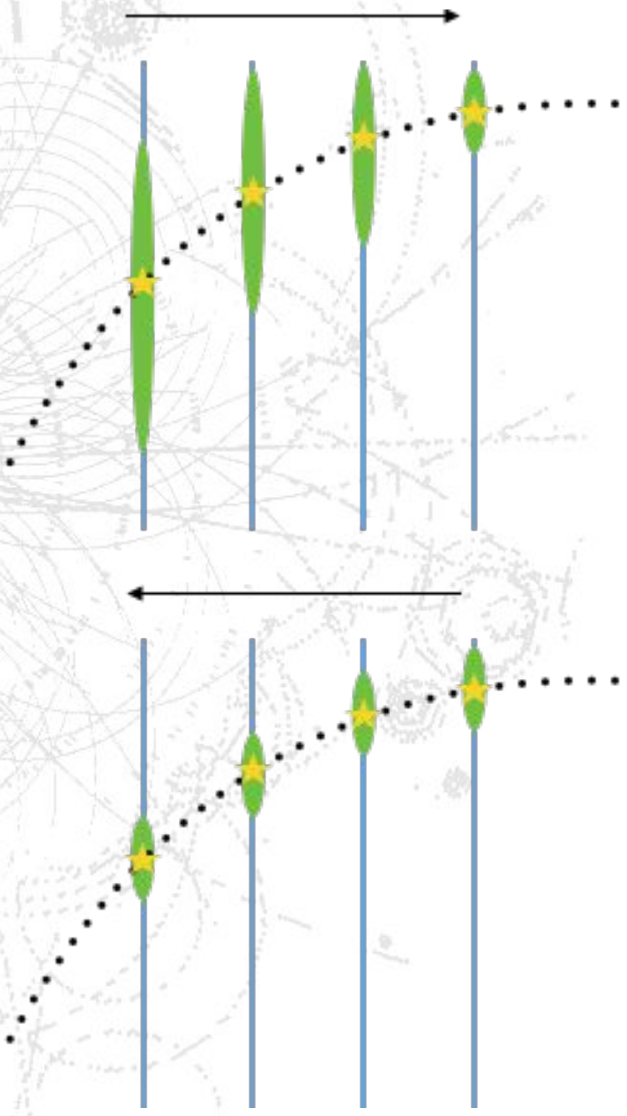
Our Kalman formalism already provides us with the framework for a track fit

- Due to progressive nature of process, only final step has "full" track information encoded in its state q_n
  - Therefore, a further stage going back along the track is needed to give best possible estimate at each surface
  - This backwards stage is referred to as the smoothing step

$$q_{k|n} = q_k + A_k(q_{k+1|n} - q_{k+1|k})$$

$$C_{k|n} = C_k - A_k(C_{k+1|k} - C_{k+1|k})A_k^T$$

$$A_k = C_k F^T_{k+1|k}(C_{k+1|k})^{-1}$$



Least-Squares Fit

Fitting based on χ^2 minimization

Another typical fitting approach which is frequently used

- Based on minimization of χ^2 function defined by track residuals and their uncertainties

$$\chi^2 = \sum_k r_k^T G_K^{-1} r_k \qquad r_k = m_k - d_k(p)$$

"residuals", i.e. difference between extrapolated local position and measurement

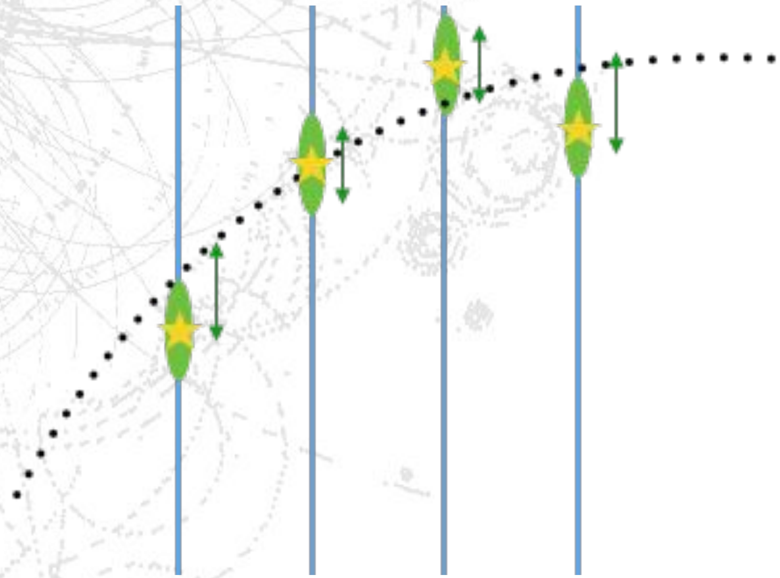
p represents defining ("global") track parameters;

d_k product of h_k and all prior f_{iji}

- Aim to find set of track parameters which minimizes χ^2

$$d\chi^2/dp = 0 \text{ with } p = p_0 + \delta p$$

p_0 is initial parameter estimate



Least-Squares Fit

Fitting based on χ^2 minimization

Another typical fitting approach which is frequently used

- Linearize the χ^2 function by performing a Taylor expansion and dropping terms beyond 1st order

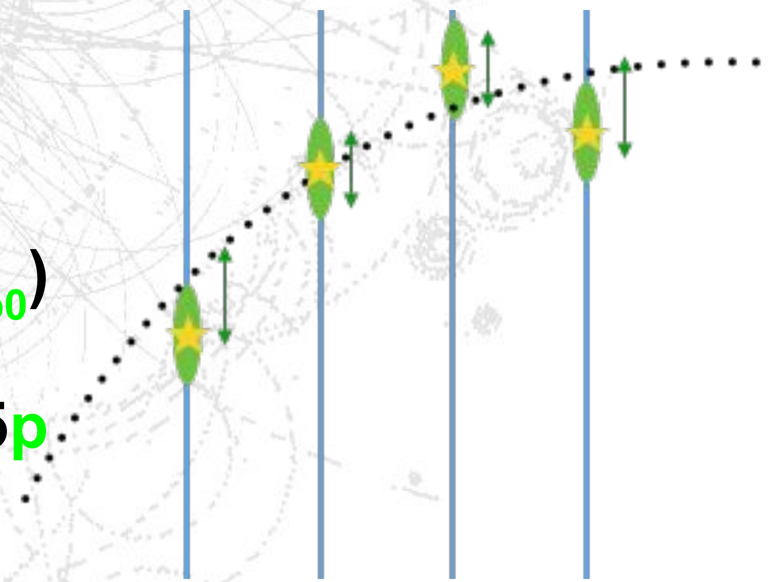
$$d_k(p_0 + \delta p) \rightarrow d_k(p_0) + D_k \delta p$$

Jacobian D_k is product of H_k and $F_{i|j}$ jacobians

- Rewriting the χ^2 minimization condition, we are left with the following to solve:

$$\delta p = (\sum_k D_k^T G_k^{-1} D_k)^{-1} (\sum_k D_k^T G_k^{-1} r_{k|p0})$$

First term directly gives us covariance of δp

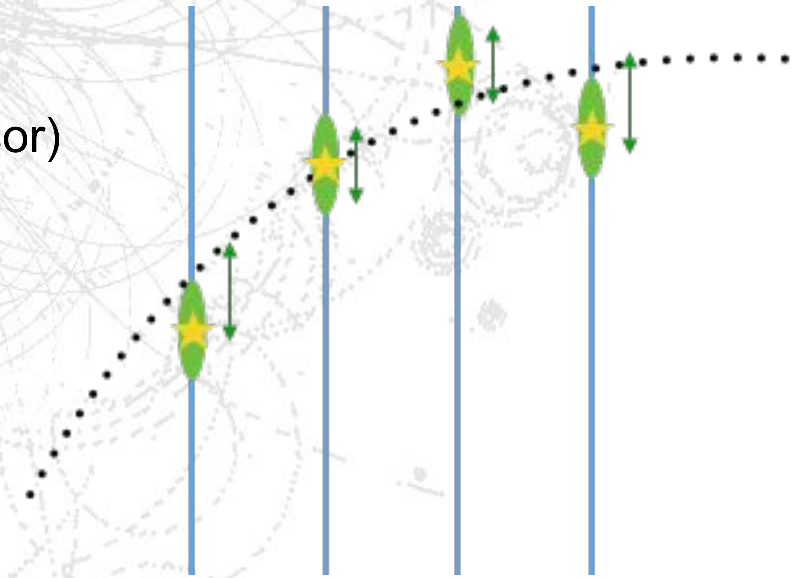


Least-Squares Fit

Fitting based on χ^2 minimization

Inclusion of Material Effects

- For small material effects, these can simply be included as additional uncertainties in $f_{k|i}$
  - However, can also add an explicit term for scattering angles to the χ^2 function
  - Can be useful if there are e.g large material structures to account for
- Add two additional parameters to be fit on each material surface (need not be a sensor)



$$\chi^2 = \sum_k r_k^T G_K^{-1} r_k + \sum_i \delta\theta_i^T Q_i^{-1} \delta\theta^i$$

$$r_k = m_k - d_k(p, \delta\theta_i)$$

Q_i is simply multiple scattering in x/X_0

Going further with fitting

Beyond the basics

Several additional techniques and optimizations can improve fit results

- Outlier removal
  - Initial candidate may include some erroneous measurements from e.g. noise or pattern recognition errors
  - Procedures can be put in place (based on e.g. contribution to overall χ^2) for these to marked as “outliers” such that they don’t bias final track parameters
- Dedicated electron energy loss treatment to account for bremsstrahlung energy losses
  - Allow for larger uncertainties in track model to account for curvature changes
  - Model non-gaussian energy loss from Bethe-Heitler formula by explicitly including multiple gaussian contributions => Kalman Filter becomes Gaussian Sum Filter
  - Care needed: Not always optimal for other particle types, therefore best combined with additional information allowing identification of electron candidates

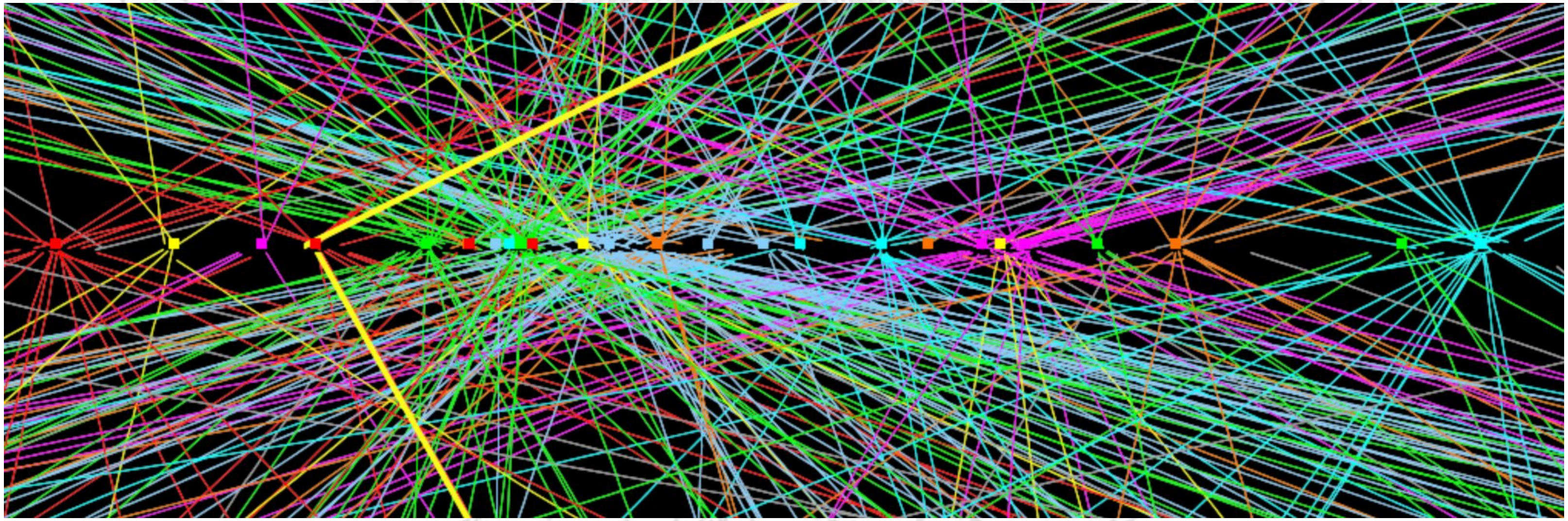
Reconstructing Vertices

Finding Vertices

Looking for the common origin

Reconstruction of primary and secondary vertices important for understanding underlying physics processes

- In collider experiments, often multiple interactions within single “Event” (referred to as pile-up interactions)
  - Understanding which tracks originate from a given interaction/process requires reconstruction of the vertex

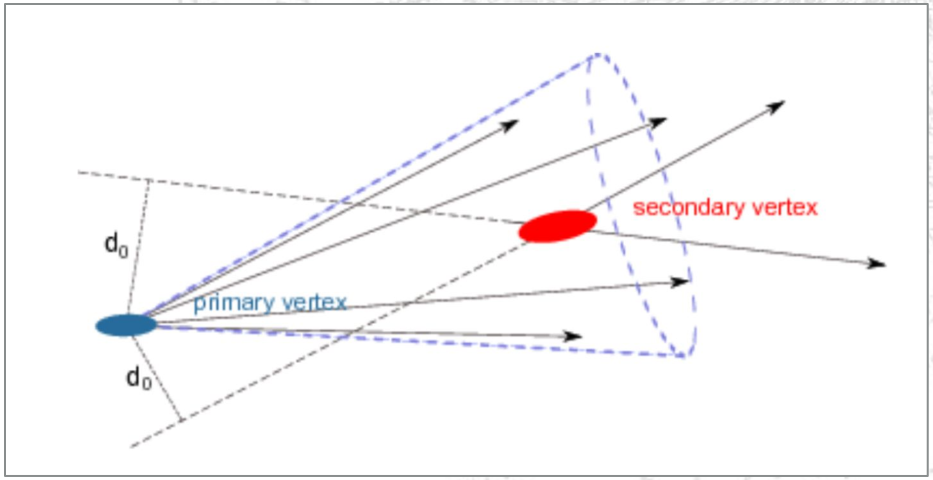


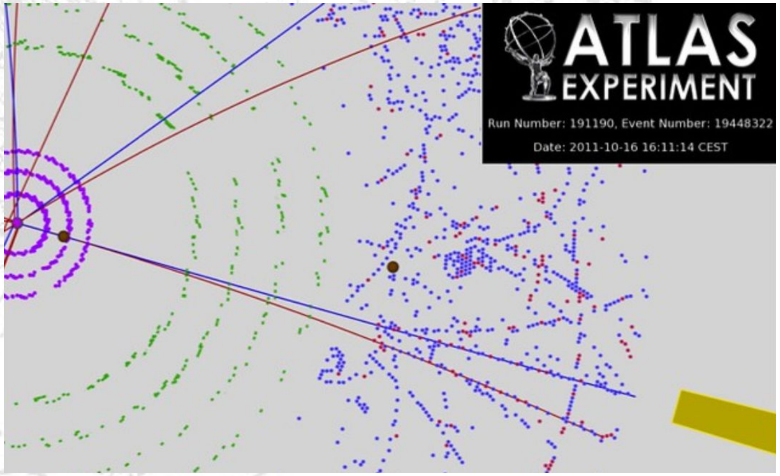
Finding Vertices

Looking for the common origin

Reconstruction of primary and secondary vertices important for understanding underlying physics processes

- Not only reconstruction of “Primary” interaction vertices, but also “Secondary” vertices important
  - Decays in flight of particles with significant lifetimes
  - Interactions with detector material; photon conversions or hadronic interactions





Vertex Reconstruction Techniques

What approaches are available

In general two step procedure similar to tracking - <u>Finding</u> and <u>Fitting</u>

- Finding: “Decide which tracks come from a common origin”
  - May use just simple geometrical methods, or also include kinematic information/constraints
- Fitting: “Determining the position of vertex and its covariance”
  - Essentially find a vertex solution that minimizes track-to-vertex distance for our track selection
- Like with track reconstruction, boundary between steps not always clear
  - Vertex reconstruction will typically implemented as either “fitting through finding” or “finding through fitting”
- Two widely-used techniques applied to this problem
  - <u>Billoir Fit</u> and <u>Kalman Fit</u>

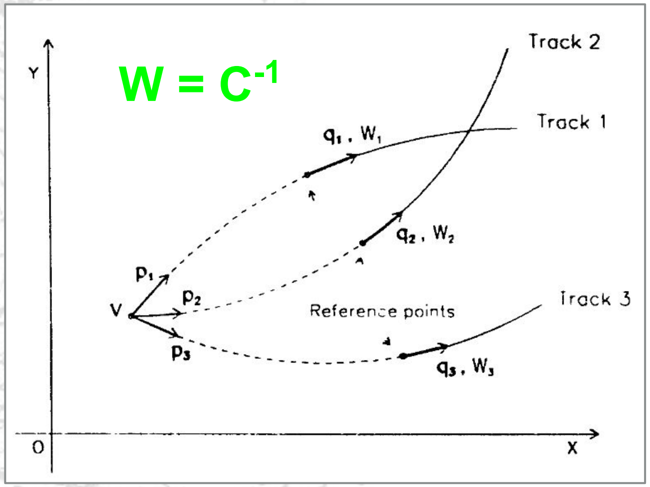
Vertex Fitting

Billoir Vs Kalman

- Billoir approach is based on least-squares technique like we saw for fitting tracks
  - Add explicit dependence on the vertex position (**V**) and the track momenta at the vertex ($\mathbf{p}_k$) to the track parameters

$$\chi^2 = \sum_k \Delta q_k^T C_K^{-1} \Delta q_k$$

$$\Delta q_k = q_k - f(V,p_k)$$



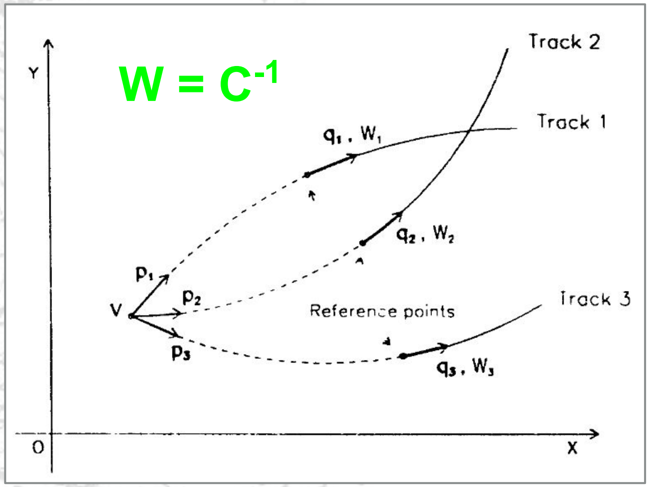
from "Fast vertex fitting with a local parametrization of tracks", Billoir, Qian, 1992

- Assuming linearity of dependence on **V** & $\mathbf{p}_k$ (within tracking errors), making first-order approximations, and exploiting matrix structure allows problem to be reduced to (relatively) simple set of matrices
  - Still typically requires an iterative procedure to arrive at solution
- Can be simplified further by dropping correlations between vertex position and momenta

Vertex Fitting

Billoir Vs Kalman

- Kalman approach uses our familiar formalism from earlier
  - State updates $\mathbf{q} \rightarrow \mathbf{q}_{k+1}$ now represent re-evaluation of parameters after addition of new track to the vertex
  - "Smoother" step corresponds to re-calculating momenta with final vertex position $\mathbf{V}_n$



from "Fast vertex fitting with a local parametrization of tracks", Billoir, Qian, 1992

- For the "Finding", tracks contributing too much to the vertex χ^2 can be dealt with by...
  - ...simply removing them from the pool of tracks to consider (potentially freed up for use by later vertices)
  - ... applying a weight to all tracks in the fit dependent on e.g. their χ^2 contribution (can be associated to more than one vertex potentially)
  - Latter approach lends itself to "Adaptive" procedure with e.g. Simulated Annealing

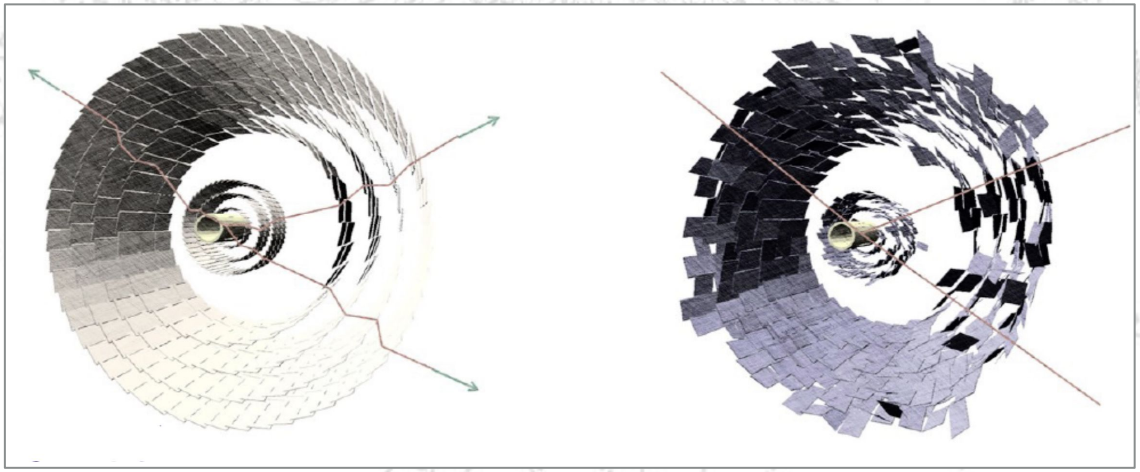
From Theory to Reality

Detector Alignment

Finding where your sensors really are...

Knowledge of precise location of sensitive elements can be important for achieving necessary track reconstruction performance

- Even very high placement accuracy can lead to displacements wrt nominal sensor positions which track reconstruction is sensitive to
  - Can degrade resolution on parameters, or even lead to biases
  - Surveys, optical alignment systems can help to understand these “misalignments”
  - Can also use the tracks themselves to understand this



Track-based Alignment

Back to fitting and residual minimization...

Can use an extension of our least-squares track fit to understand misalignments

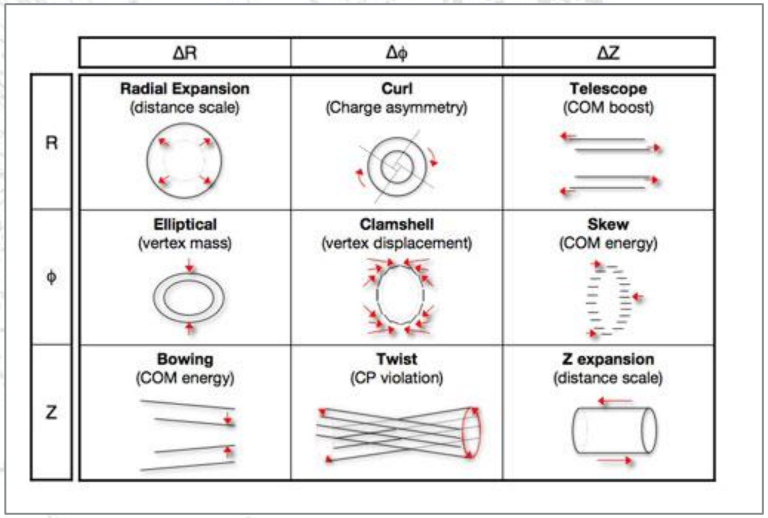
- Each alignable object typically has 6 alignment parameters; 3 rotational (R_i) and 3 translational (T_i) corresponding to physical degrees of freedom
- Minimize a χ^2 that depends not only on track parameters **p**, but also alignment parameters $\boldsymbol{\alpha}$ (global χ^2 alignment)
  - E.g. include a dependence on α in residual definition
- Solving using methods discussed previously now potentially involves very large matrices and a number of iterations
  - Computationally expensive; most efficient method may depend on the details of your detector
  - Possible to trade off time in matrix inversion against more iterations by removing dependence on **p** in **d/d$\boldsymbol{\alpha}$** (local χ^2 alignment)

Constrained Alignment

Using external knowledge to improve our alignment further

Some classes of misalignments may not be resolved by the methods on the previous slide, e.g. so-called "Weak Modes"

- Consider a correlated misalignment between detector layers
  - Can give a good fit χ^2 for a wrong trajectory by preserving helical track model
- Need to include external constraints to identify such effects
  - Constraints can be added to function χ^2 e.g. by considering it as a "pseudo-measurement"
- Various types of constraints possible
  - From independent detector system measurements (e.g. calorimeter energy)
  - From physics (e.g. mass constraints on resonance decay systems like J/Ψ or Z)

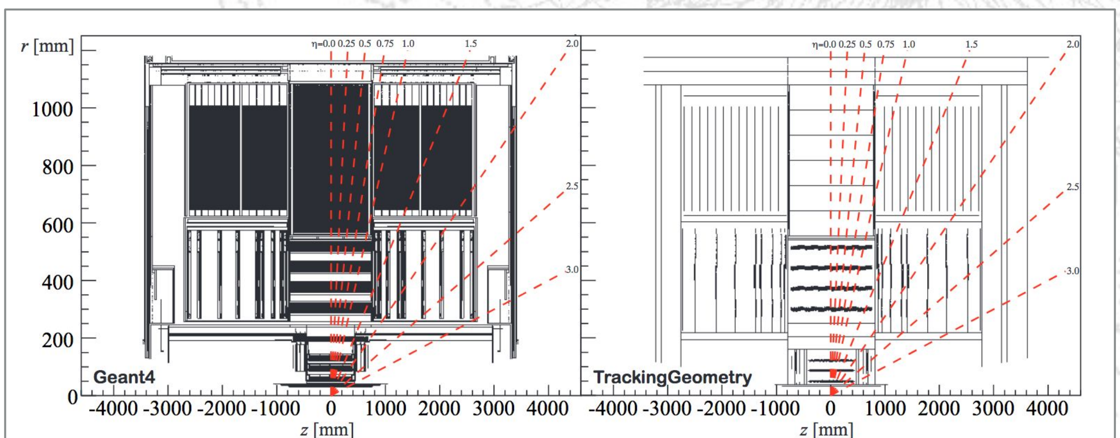


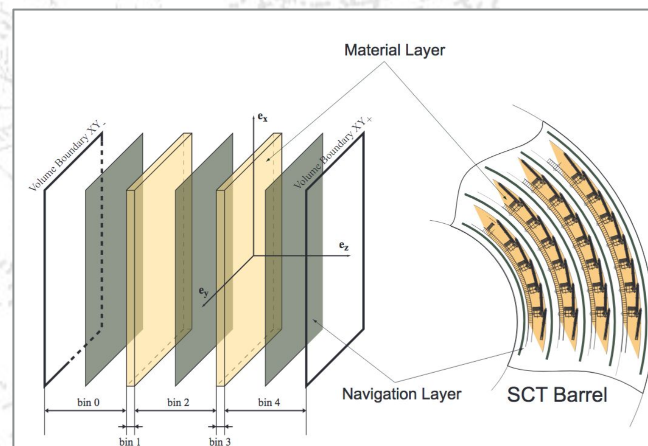
Understanding Detector Material

Representing the detector material

Describing detector material with appropriate accuracy important for good performance (both track precision and technical aspects)

- A highly detailed Geant4 (or similar) simulation with full best-knowledge material description is normally available for producing Monte Carlo samples
  - Using this for providing material in track propagation typically impractical
  - Simplified material description needed per surface known to track reconstruction
  - E.g. “observed” x/X_0 distribution binned in η/ϕ per layer





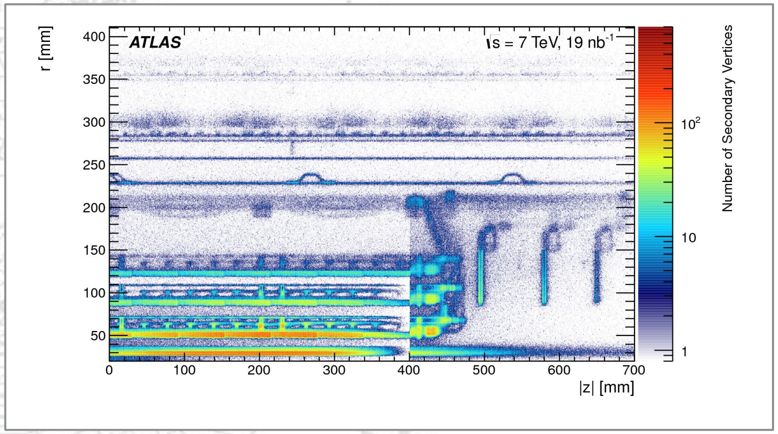
ATL-SOFT-PUB-2007-004

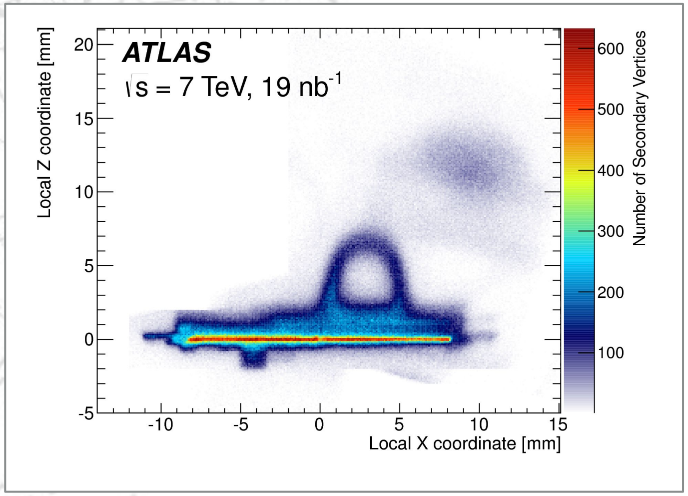
Understanding Detector Material

“Weighing” the detector with tracks

Typically don’t know how much material is *really* in a detector once it is built...

- Initial detailed simulation typically based on best engineering estimates
  - This will often underestimate the true picture, from small effects like extra cable lengths curling up, to larger contributions simply forgotten...
- Reconstructing secondary vertices from photon conversion and hadronic interactions allows this to be studied in detail
  - Compare number and position between data and Monte Carlo
  - Can use these comparisons to feed back into simulation model and improve the description





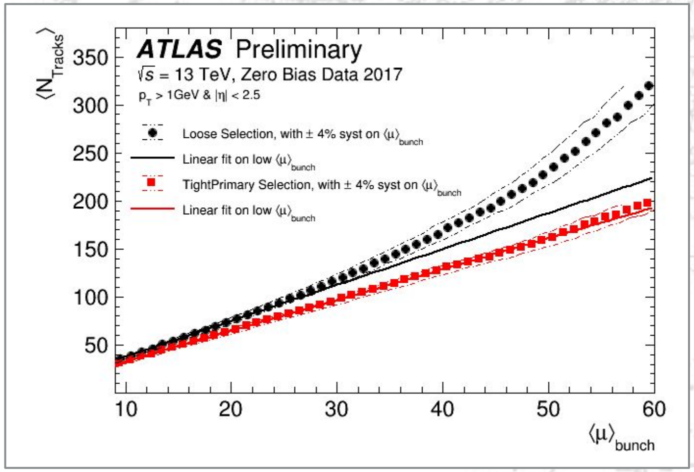
JINST 11 (2016) P11020

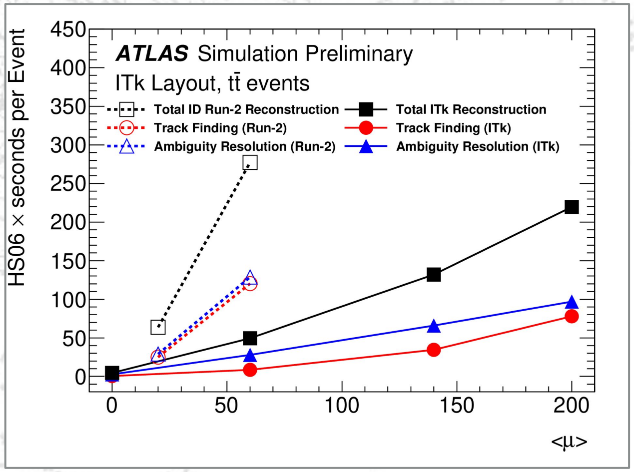
Coping with Pile-up

Mo' Data Mo' Problems...

Intensity frontier pushing towards ever-higher instantaneous luminosities

- More particles in the detector at one time makes track reconstruction trickier and more time-consuming
  - More genuine tracks to process, plus combinatorial challenge in pattern recognition results in super-linear scaling in both number of reconstructed tracks and CPU time
- Keeping excellent performance while sticking within CPU, memory, and disk space budgets is a big challenge for future collider experiments
  - New and fresh ideas very welcome! Maybe you have some?





Thank you for your attention!

Any questions?

Links

Reference material used in producing these slides

- Lecture By Salva Marti-Garcia
- Lecture by Pippa Wells
- Lecture by Markus Elsing
- Lecture Series by Wouter Hulsbergen
- Document by Are Strandlie and Rudolph Frühwirth
- (last two in particular are excellent references for all of the full mathematical treatments)