Track and Vertex Reconstruction at LHC

Wolfgang Liebig[1]

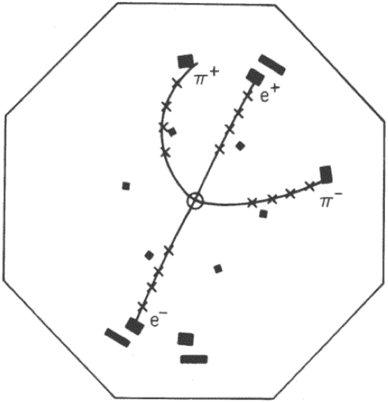
[1]Nikhef, Amsterdam

Terascale Workshop on LHC Detector Understanding
DESY, Hamburg, 2009-07-01

Definitions

Track reconstruction:

1. associate sets of hits to different charged particles in the event,
2. determine trajectory matching those hits and compute the best possible estimate of the track parameters



Computer reconstruction of Ψ' cascade decay at Mark I, SLAC, 1974

Definitions

Track reconstruction:

1. associate sets of hits to different charged particles in the event,
2. determine trajectory matching those hits and compute the best possible estimate of the track parameters

Vertex reconstruction:

1. associate sets of tracks to a common interaction point in space,
2. compute intersection point and classify type of interaction

Note:

- separation into tracking and vertexing is not trivial
- strategy split into *finding* and *fitting* – not always true

Definitions

Track reconstruction:

1. associate sets of hits to different charged particles in the event,
2. determine trajectory matching those hits and compute the best possible estimate of the track parameters

Vertex reconstruction:

1. associate sets of tracks to a common interaction point in space,
2. compute intersection point and classify type of interaction

Note:

- separation into tracking and vertexing is not trivial
- strategy split into *finding* and *fitting* – not always true

Tracks and Vertices in Physics

Charged particle track:

- physics analyses need 4-momenta and charge sign
- particle flow & identification: combine tracker information with calo, muons
- flavour tagging: b-tagging performance extremely sensitive to quality and precision of tracks
- B-physics, tau-leptons, etc.

Vertices:

- primary production vertex
- signal PV against pile-up
- identify photon conversions and decay of neutral hadrons
- precise primary and secondary vertex reconstruction for lifetime measurements and b-tagging

Tracks and Vertices in Physics

Charged particle track:

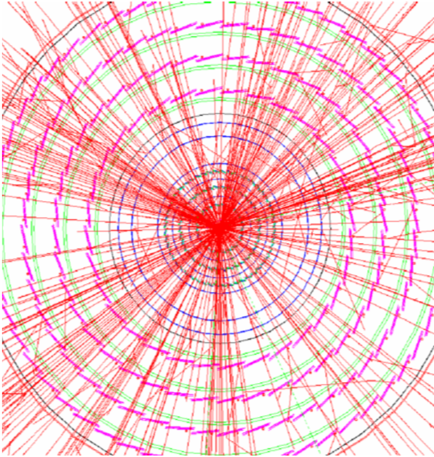
- physics analyses need 4-momenta and charge sign
- particle flow & identification: combine tracker information with calo, muons
- flavour tagging: b-tagging performance extremely sensitive to quality and precision of tracks
- B-physics, tau-leptons, etc.

Vertices:

- primary production vertex
- signal PV against pile-up
- identify photon conversions and decay of neutral hadrons
- precise primary and secondary vertex reconstruction for lifetime measurements and b-tagging

Particular Challenges at the LHC Detectors

- combinatorics: high track densities and numbers of read-out channels
- high precision: small hit errors require precise error propagation through detector
- distortions: very inhomogeneous dead material and magnetic field
- computing limits: event sizes and trigger rates demand efficient software techniques
- vertexing with pile-up: identify signal collision as 1 out of ~ 23



Outline

This lecture covers three main parts:

1. Track Reconstruction
   - Track Model and Parameter Estimators
   - Error Propagation and Material Effects
   - Trajectory Distortions
2. Vertex Reconstruction
   - Vertex Reconstruction
3. Detector-Specific Aspects - ATLAS and CMS
   - Detectors and Pattern Recognition
   - The Software
   - Calibration and Alignment
   - Understanding the First Data

part 1: Track reconstruction

The Essentials of Tracking

track model

transport in B-field and material corrections

parameter estimation

The track fit, linearisation

trajectory distortions

outliers, interactions, E-loss

measurement model

calibration and alignment

pattern recognition

combinatorics, fast versions of the above

- determines structure of ATLAS/CMS software
- differently affected by real (imperfect) detector

The Essentials of Tracking

track model

transport in B-field and material corrections

parameter estimation

The track fit, linearisation

trajectory distortions

outliers, interactions, E-loss

measurement model

calibration and alignment

pattern recognition

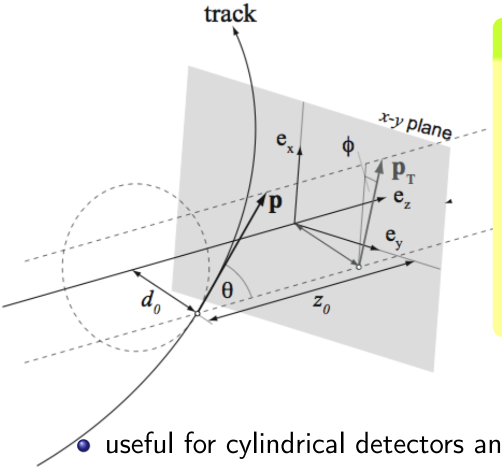
combinatorics, fast versions of the above

- determines structure of ATLAS/CMS software
- differently affected by real (imperfect) detector

Track Models

- The track model parameterizes the charged particle trajectory
- Track models have 5 parameters:
  - ▷ stable particle moving in stationary B-field in vacuum is described by 6 quantities (position, momentum)
  - ▷ however, initial position along trajectory is free
- 5 parameters expressed at an intersected reference surface
  - ▷ local coordinates l_1, l_2, angles ϕ, θ and curvature q/p
  - ▷ called 'LocalParameters' (ATLAS) or 'Global State' (CMS)
- In reality, the detector geometry affects the model through field configuration and material effects
  - ▷ *"propagation"* of parameters along trajectory

Track Parameters at Collider Detectors



Perigee parameterization:

- d_0 signed distance of closest approach to z axis
- z_0 z of closest approach
- ϕ_0 azimuthal angle at cl. app.
- θ polar angle of track
- q/p charge-signed curvature

- useful for cylindrical detectors and solenoidal B-field (B_z)
- basis for 4-vector parameterization in physics analysis (B_z)

Estimators and Track Parameters

- track parameter reconstruction starts with:
  - data set $\{m_i\}$, errors $\mathrm{cov}(\boldsymbol{m}) = V$
  - a model $\mathcal{P}_i(m_i, \boldsymbol{\lambda})$ with unknown par's $\boldsymbol{\lambda}$.
- need **estimator** for $\boldsymbol{\lambda}$:
  - best: smallest variance
  - unbiased: expectation value close to $\boldsymbol{\lambda}$
- for instance maximum likelihood estimator
- $\boldsymbol{\lambda}_{\mathrm{ML}}$ for which likelihood function

$$\mathcal{L}(\boldsymbol{\lambda}, \{m_i\}) = \prod_i \mathcal{P}_i(m_i; \boldsymbol{\lambda})$$

  becomes maximum
- method available through generic fitting algorithms like MINUIT
- However, track and vertex fitting does not use MINUIT.
- instead use linear estimator!

Gaussian-distributed Measurements

- usually several independent effects sum up to measurement resolution
- $\Rightarrow$ measurements distributed normally (Gaussian p.d.f.) around true values
- For Gaussian p.d.f.

$$\mathcal{P}_i(m_i, \boldsymbol{\lambda}) = \frac{1}{\sqrt{2\pi}} \exp\left[\frac{1}{2}\left(\frac{m_i - h_i(\boldsymbol{\lambda})}{\sigma_i}\right)^2\right]$$

- the least-squares estimator

$$\chi^2 = \sum_i \left(\frac{m_i - h_i(\boldsymbol{\lambda})}{\sigma_i}\right)^2 = -2 \ln \mathcal{L} + \text{const.}$$

- is the ML and therefore smallest-variance estimator

Linear Model

- We can approximate the track model by a linear model in the neighbourhood of the measurements:

$$h(\boldsymbol{\lambda}) = h_0 + H\boldsymbol{\lambda}$$

(under the condition that measurement errors vary little with $\boldsymbol{\lambda}$)

- minimum χ^2 condition ($\frac{d\chi^2}{d\boldsymbol{\lambda}} = 0$) gives the linear estimator

$$\hat{\boldsymbol{\lambda}} = CH^TV^{-1}(m - h_0), \quad C = \mathrm{var}(\hat{\boldsymbol{\lambda}}) = (H^TV^{-1}H)^{-1}$$

- Properties of linear LSEs are important in tracking: allow simple error propagation, χ^2 tests etc
- known as global fit: applies all meas. constraints at once

Applying Constraints Progressively

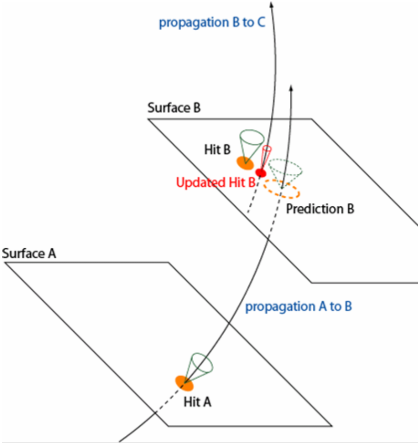
- alternative: sequentially apply measurement constraints m_k
- again, assuming linear model
- add a single data point k as a correction to previous state $k-1$

$$\begin{aligned} x_k &= x_{x-1} + K_k\big(m_k - h_k(x_{k-1})\big) \\ C_k &= (1 - K_k H_k) C_{k-1} \end{aligned}$$

- Kalman gain matrix

$$K_k = C_{k-1} H_k^T \left(V_k + H_k C_{k-1} H_k^T\right)^{-1}$$

- matrix of $\dim(m_k)$ to invert: fast!



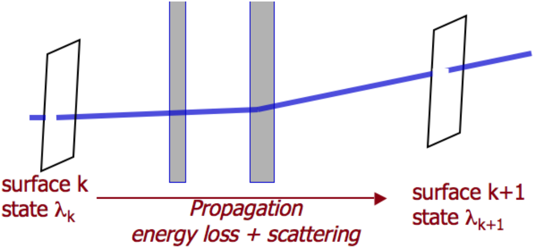
Kalman filter

The Kalman Filter

- developed originally for fast signal processing
  - time evolution of dynamical systems,
    for example updating rocket direction from radar signal
  - global fit needs to refit complete trajectory with every signal
- Kalman filter brings additional benefits to tracking:
  - local treatment of multiple scattering
  - use in local pattern recognition
  - integrating (non-Gaussian) energy loss in the track model
- will come back to each point
- Kalman filter also exists for vertexing

Parameter Propagation and Material Effects

- track parameters and associated covariances are changed by passage through B-field and material



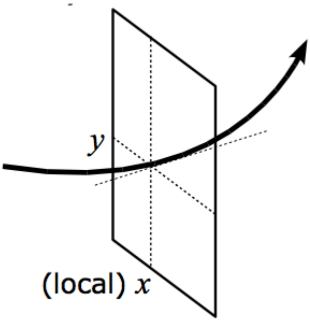
- separate track model: $\boldsymbol{\lambda}_k = f(\boldsymbol{\lambda})$ from meas't model: $h_k(\boldsymbol{\lambda}_k)$
- measurement model depends on k's geometry and sensor: h_k usually direct projection of measured coordinates
- energy loss is corrected deterministically, multiple scattering treated schochastically

Parameter Propagation

- equation of motion of particle

$$\frac{d^2\boldsymbol{r}}{ds^2} = \frac{q}{p}\left(\frac{d\boldsymbol{r}}{ds} \times \boldsymbol{B}(\boldsymbol{r})\right)$$

- helix approximation not sufficient: risk $1\,\%$ momentum bias (CMS)
- $\boldsymbol{B}(\boldsymbol{r})$ inhomogeneous: differential eq. can only be solved numerically
- ATLAS uses Runge-Kutta methods:
  - ▷ divide integration interval in steps;
  - ▷ each step becomes initial-value problem;
  - ▷ solve equation for each step independently
- high accuracy (short steps, many field look-ups) is cpu-costly!



Fast Track and Error Propagation

Adaptive step-length

- better than fixed step-length for B-field with regions of different inhomogeneity
- additional evaluation stage
  - ▷ to estimate local error of propagation
  - ▷ to trim step length for current position
- step acceptance criterion: local error $<$ error tolerance
- adaptive Runge-Kutta-Nyström

Propagation of error matrices

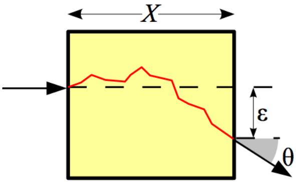
- purely numerical scheme: propagate set of auxiliary tracks with smeared parameters
- semi-analytical scheme: differentiate result of numerical integration in each step parallel transport of parameters and Jacobian elements
- semi-analytical much faster!
- both include gradients of E-loss and magnetic field

Material Effects: Energy Loss by Ionization

- Same effects which allow particle detection cause energy loss
- Energy loss depends very specifically on traversed medium, particle type and momentum
- mean specific energy loss described by Bethe-Bloch:

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

Material Effects: Coulomb Scattering



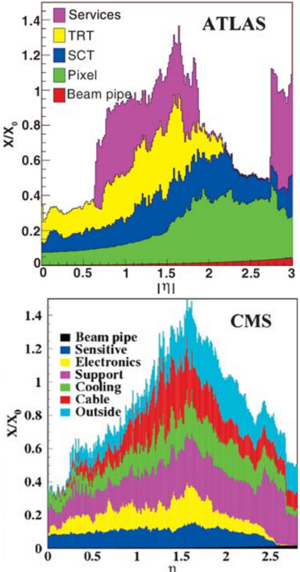
- charged particle deflected when passing through matter
- random deflection is result of many small-angle Coulomb scatterings on the nuclei
- Gaussian distribution for central $98\,\%$ given by Highland formula

$$\sigma(\theta) = \frac{13.6\,\mathrm{MeV}}{\beta c p} z \sqrt{x/X_0}\left(1 + 0.038 \ln\left(x/X_0\right)\right)$$

- expect $E(\varepsilon) = 0$, $E(\theta) = 0$. $\sigma(\theta)$ is proportional to $1/p$
- x/X_0: thickness of material in fraction of radiation length

Radiation Length

- radiation length X_0:
  mean distance over which electron loses $63\,\%$ of its energy
- also relevant for photons: survival probability is $1/e$ over $\frac{7}{9}X_0$.
- example: 300 μm Si gives $0.003\,X_0$
- ATLAS and CMS trackers are heavy!
- Consequences:
  - ▷ track fit needs good description of material effects
  - ▷ electron bremsstrahlung and photon conversions need to be reconstructed with the help of track detectors



Multiple Scattering in Global Fit

- break-point method: for each scattering plane add two scattering angles to track model: $h_i(\boldsymbol{\lambda}, \theta^{\text{scat}})$
- and a contribution to χ^2:

$$\chi^2 = \sum_i^{N_{\text{hits}}} \left(\frac{m_i - h_i(\boldsymbol{\lambda}, \theta^{\text{scat}})}{\sigma_i} \right)^2 + \sum_j^{N_{\text{planes}}} \left(\frac{E(\theta^{\text{scat}}) - \theta_j^{\text{scat}}}{\sigma^{\text{scat}}} \right)^2$$

- expectation value of scattering angles is $E(\theta^{\text{scat}}) = 0$
- However: introduces new parameter correlations, solution inverts large matrix $(\dim = 5 + 2N^{\text{scat}})$
- also needs 'smart' aggregation of detailed material onto planes
- method in wide use by ATLAS, not implemented in CMS.

Multiple Scattering in Kalman Filter

- state propagation

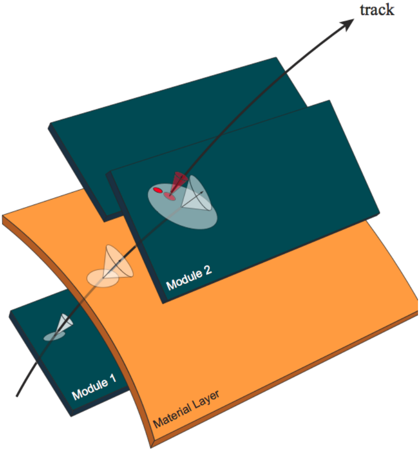
$$\boldsymbol{\lambda}_k^{k-1} = f_k^{k-1}(\boldsymbol{\lambda}_{k-1})$$

  does B-field integration
  and energy loss correction

- error propagation

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

- the process noise matrix Q_k reflects multiple scattering uncertainties in extrapolation from state k to $k+1$



Kalman Filter with State Propagation

- add a single data point

$$\begin{aligned}\boldsymbol{\lambda}_k &= \boldsymbol{\lambda}_k^{k-1} + K_k\left(m_k - h_k(\boldsymbol{\lambda}_k^{k-1})\right)\\ C_k &= \left(1 - K_k H_k\right)C_k^{k-1}\end{aligned}$$

- Kalman gain matrix

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

- Smoothing:
  run two filters in opposite directions and build weighted average to obtain track parameters and error at every surface

Quality of fit: Pull Quantities

Pull Distribution

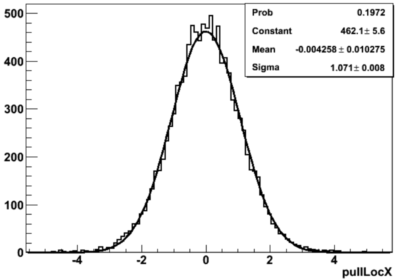
- verifies error estimate and unbiasedness of fit

$$\mathrm{pull}(y) = \frac{y - E(y)}{\sqrt{\mathrm{var}(y)}}$$

- pull should be distributed with mean 0, rms 1
- use 3 kinds of pull:
  - measurement pull (truth), tests input to fit
  - parameter pull (truth), tests track model
  - (measurement) residual pull

Residual Pull

- residual $r_k = m_k - h_k(\boldsymbol{\lambda})$
- pull $= r_k / R_k$ (R: cov. of residual)



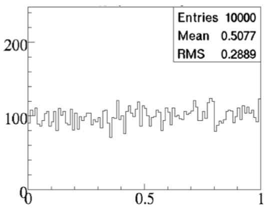
Chi-square and Robustness

Chi-square distribution

- if residual pull is good, χ^2 distribution should obey

$$E(\chi^2/\text{n.d.o.f.}) = 1$$

- n.d.o.f. $=$ # constraints – # params
- χ^2 probability should be flat



Outliers

- effects creating outliers:
  - ▷ error larger than expected, example: shortened drift or large cluster due to δ-electron
  - ▷ noise or wrong hit
- simplest robust estimator: reject largest residual pull, refit
- typical outlier cuts
  - ▷ reject hit if pull > 3.5
  - ▷ reject track if $\mathcal{P}(\chi^2) < 10^{-5}$
- avoid bias or degraded resolution

Gaussian-Sum Filter

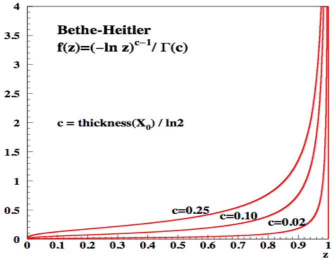
- Linear LSE only optimal in linear systems with Gaussian meas't errors and process noise
- Basic idea for non-Gaussian case:
  Keep advantages of LSE by
  describing general pdf's as mixture of Gaussian components

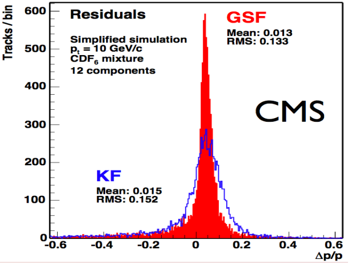
$$f(\lambda) = \sum_{i}^{N} w_i \phi(\lambda; \mu_i, V_i) \quad \text{with} \ \sum_i w_i = 1$$

- consists of N Kalman filters run in parallel
- After each update the weights are recalculated to reflect the compatibility with the measurement
- needs knowledge of noise distribution
- needs component reduction to control combinatorial explosion

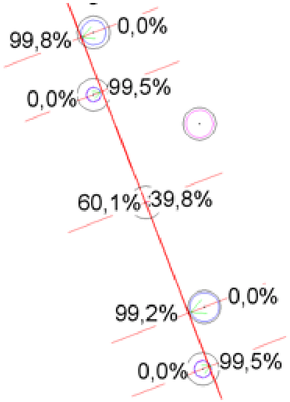
Gaussian-Sum Filter: Electrons

- Electron Bremstrahlung: very non-Gaussian noise !
- local extension of track model
  - ▷ brem. point in global track fit
  - ▷ Kalman filter with dynamic noise adjustment
- Gaussian-sum filter approximates Bethe-Heitler as Gaussian mixture
- GSF is very CPU costly
  - ▷ ATLAS & CMS use it only on electron candidate tracks

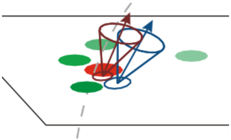




Deterministic Annealing Filter



- for extreme hit occupancy or noise
  - allow for several hits per layer to compete in track
  - use annealing iterations to find global minimum



- give assignment probability to hit i in layer

$$p_i = \frac{\phi_i(T)}{n\Lambda + \sum_j \phi_j(T)}$$

- weights ϕ_i calculated from residuals before next iteration, then: $T \to 1$ (annealing)
- multi-track fitter (in CMS): add competition between close-by tracks

part 2: Vertex reconstruction

Vertex Fitting

- techniques are very similar to track reconstruction:
  - data are N track parameter vectors + covariances
  - output: vertex position, track momentum vectors and the full covariance matrix
- measurement model $h(\boldsymbol{\lambda}, \boldsymbol{p})$ describes dependence of track parameters $\boldsymbol{\lambda}$ on vertex position $\boldsymbol{x}$ and momenta $\boldsymbol{p}$
  - is inherently non-linear
  - more unknown parameters: 3 + 3×N
  - process noise has no equivalent
- use again linear LSEs but need iterations

Vertex Least-Square Estimators

- total χ^2 from sum over N tracks

$$\chi^2 = \sum_i^N \left(\boldsymbol{\lambda}_i - h(\boldsymbol{x}, \boldsymbol{p}_i)\right) C_i^{-1} \left(\boldsymbol{\lambda}_i - h(\boldsymbol{x}, \boldsymbol{p}_i)\right)$$

- commonly use helix parameterisation in h and derivatives

$$D_i = \frac{\partial h(\boldsymbol{x}, \boldsymbol{p}_i)}{\partial \boldsymbol{x}} \qquad E_i = \frac{\partial h(\boldsymbol{x}, \boldsymbol{p}_i)}{\partial \boldsymbol{p}_i}$$

(will skip further details, see literature in appendix)

- linearisation of non-linear model

$$h(\boldsymbol{x}, \boldsymbol{p}_i) = h_0 + D_i \boldsymbol{x} + E_i \boldsymbol{p_i}$$

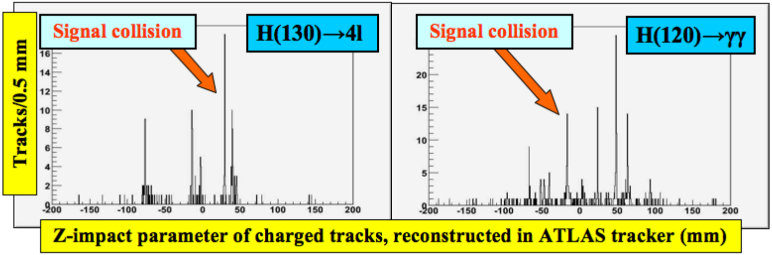
requires iteration and propagation of track parameters as vertex estimate moves

Vertex Least-Square Estimators

- practical methods avoid inverting large matrix (dim=3+3N)
- Billoir algorithm is a global fit exploiting the empty structure of $HV^{-1}H^T$ in LSE (ATLAS only)
- Kalman fit adds tracks one by one
- both methods can be used to perform or omit (faster) the calculation of the new track momenta at the vertex
- sequential addition of tracks to Kalman allows to exclude incompatible tracks during fit
- primary vertex reconstruction at LHC required new approach to vertex finding through fitting

Primary Vertex Reconstruction

- The beam spot is described by Gaussian parameters
  - $\sigma_x = \sigma_y = 0.015\,mm$, $\sigma_z = 56\,mm$, displaced from $(0, 0, 0)$
- pile-up: comparing to min. bias events signal events have higher track multiplicity and p_T
  - different methods of identifying the primary vertex
  - None of them gives 100 % efficiency for all channels
  - tracking distortions and beam spot displacement



Adaptive Multi-Vertex Fitter

- adaptive method works like Deterministic Annealing Filter:
  - ▷ downweights tracks according to compatibility with vertex
- For events with many close-by pile-up vertices: adaptive multi-vertex fitter
  - ▷ vertices compete against each other for track assignment
  - ▷ iterative annealing is used to approach a hard assignment
  - ▷ No prior assumption on number of primary vertices
- signal vertex taken as the one with highest $\sum (p_i^T)^2 / N_{\text{track}}$

Kinematic Fitting and constraints

Decay vertex fitting

- photon conversion or V^0 decay candidates
- full decay trees
- types of constraint:
  - ▷ invariant mass
  - ▷ direction of decaying particle
  - ▷ direction of decayed particles (collinearity)
- improves estimate when few measurement constraints available (1-2 charged tracks)

Applying exact constraints

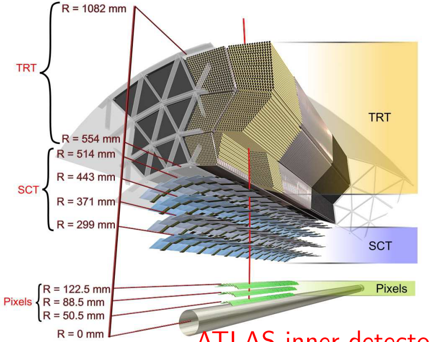
- Lagrange mutliplier (λ_j)
  - ▷ add extra term to χ^2:

$$\chi^2_+ = \lambda_j g_j(\boldsymbol{x}, \boldsymbol{p}_i)$$

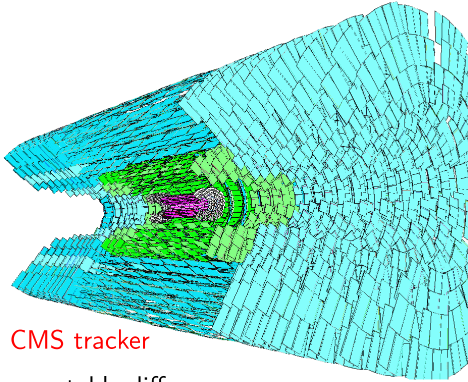
  - ▷ minimize χ^2 wrt. $\boldsymbol{x}, \boldsymbol{p}_i, \lambda_j$.
- Kalman filter
  - ▷ apply constraint as measurement-update with covariance 0

part 3: Detector-Specific Aspects

Tracking Detectors



ATLAS inner detector

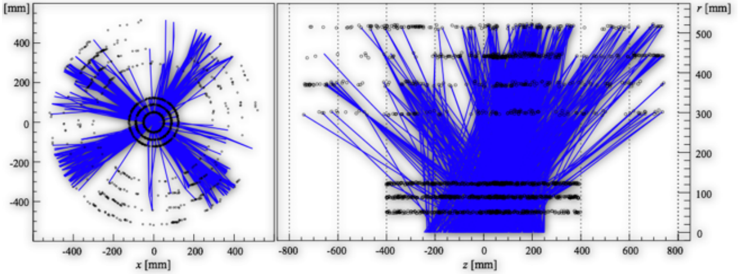


CMS tracker

- Two hermetic and precise trackers - notable differences:
  - ▷ magnetic field strength: 2 T vs. 4 T
  - ▷ pixel dimensions: 50×400 μm vs. 100×150 μm (in $R\phi \times z$ coord.)
- cause differences in parameter resolutions and tracking strategies

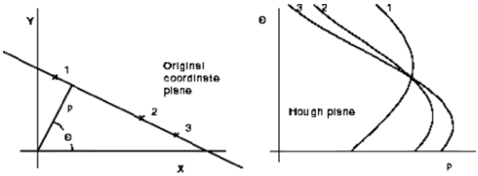
Pattern Recognition Strategies

- Two modi of operation:
  region of interest for high-level trigger or full event for off-line
- choice of track finding strategy depends on detector geometry
  - ▷ usually combination of seed finding and track following
- aim is high efficiency at low fake rates
- robustness against combinatoric problems and detector ambiguities



Global Pattern Recognition

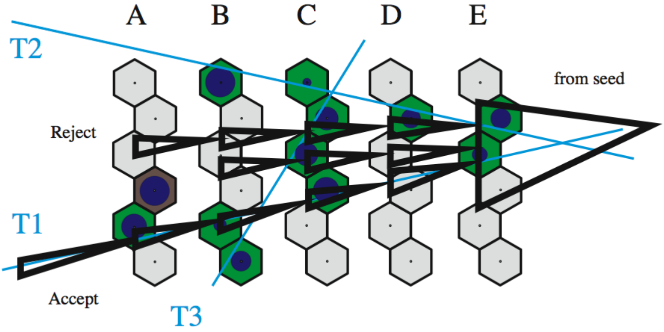
- Seed finding, usually restricted to a sub-detector
- main methods:
  - ▷ look-up tables or templates
  - ▷ Hough-transform
  - ▷ neural networks
- aware of event topology:
  - ▷ collisions (beam-spot)
  - ▷ cosmic rays (off-center)
  - ▷ beam halo passage, beam-gas collision
- robustness against large variations in multiplicity



- invert meas't function $f : \boldsymbol{\lambda} \to \boldsymbol{m}$
- measurements become hypersurfaces in parameter space
- they intersect at true parameters
- divide space into cells, find maxima

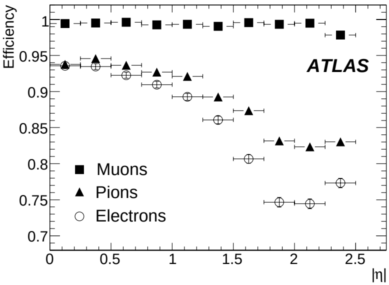
Pattern Recognition - Track Following

- follow seed candidate and search for hits in adjacent layers
- combinatorial track following:
  - branch seeds if more than one hit compatible
  - follow all seeds, evaluate candidates to reject bad ones
  - evaluation score built from number hits, holes and χ^2
- Kalman filter ideal for this, improves parameters with each hit

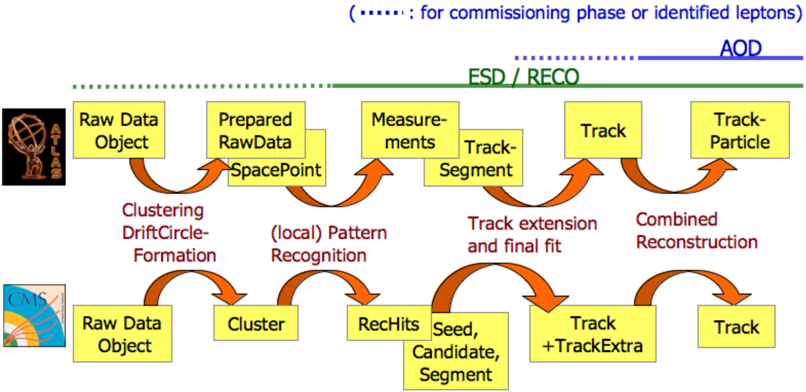


Efficiency and Fakes

- track efficiency says what fraction of true particles has been found
- determination from truth matching, either:
  - hit matching – certain fraction of hits are correctly associated (robust for high track densities → in use for inner trackers)
  - parameter matching – true and rec. parameters sufficiently close
- fake: not or only partially matched
- efficiency determination on data uses detector redundancy
- total efficiency ⟶ = detector eff. × reco eff.
- distinguishes if a track is "reconstructible" by software



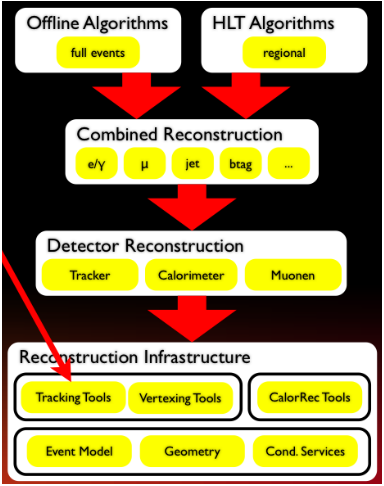
Event Data Model



- objects are mostly the same, just named differently!
- written to "event store", readable by downstream algs
- Final step is (selective) write to disk.

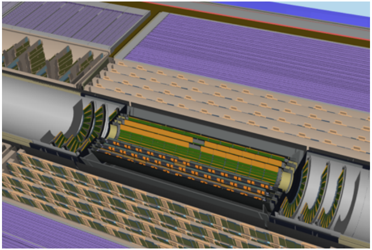
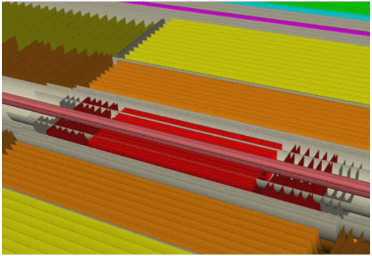
Reconstruction Software Design

- design principle: modularity!
  - ▷ reduce SW dependencies, maintainable over LHC life
  - ▷ performance, multiple use
- communicate through
  - ▷ common event model
  - ▷ abstract interfaces
- tracking/vertexing software written in very modular way



Track Reconstruction Geometry

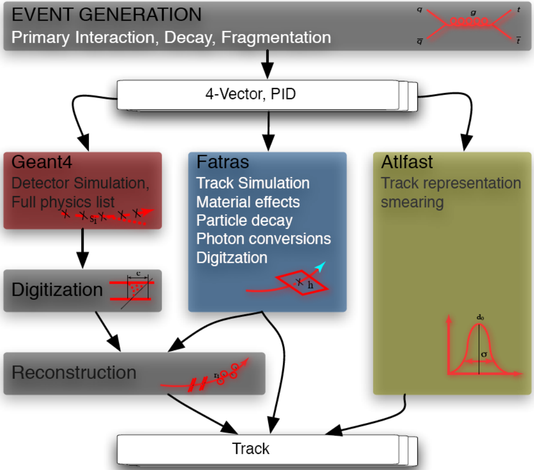
- G4 geometries too complex for reconstruction
  - ▷ simplified geometries provide material layers and field

- nodes reduced to $\mathcal{O}(10k)$ – factor $\sim 10^3$ wrt Geant4
- description of sensitive detectors identical

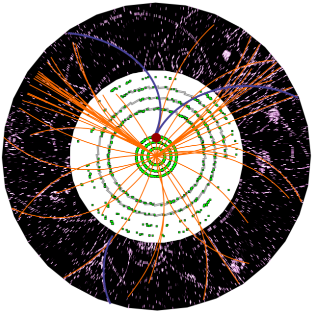
Fast Track Simulation

- ATLAS & CMS use reconstruction tools to improve fast simulation!
  - reco geometry, fast error propagation, PDG material effects
- nuclear interactions
- CMS parameterizes clustering and track finding effects
- ATLAS runs full reco chain starting from clustering
- $\mathcal{O}(100)$ faster than full Geant4

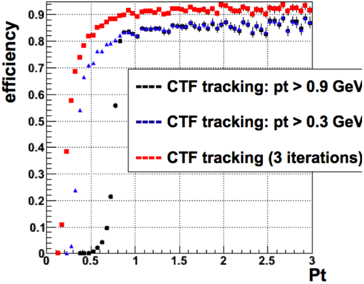


Track Reconstruction Strategies

- track search is iterated:
  - main inside-out search
  - following search for low p_T and non-pointing tracks on remaining hits
- ATLAS: NewTracking, iPatRec with Si- and TRT-seeded iteration



- CMS: Road Search and Combinatorial Track Finder
- fewer combinatorics and better efficiency than single search



Calibration and Tracking

Measurement Calibration

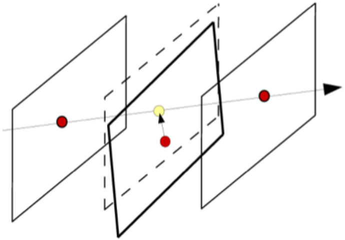
- need knowledge of track intersection to determine optimal meas't parameters m_k, V_k
- requires ATLAS & CMS to
  - re-calibrate during tracking
  - and iterate track fitting
- pixel and strip clusters:
  - bow, track-vs-Lorentz angle...
  - ambiguities, like ganged pixel
- drift tubes:
  - correct for drift time
  - solve left-right ambiguity

Software

- ATLAS & CMS software:
  - RIO_OnTrackCreator
  - pixel templates
- first data: usually apply "conservative" calibration

Alignment with Tracks

- detector positioning accuracy:
  - ▷ $\sim 100\,\mu m$ sensor modules
  - ▷ $\sim 1-5\,mm$ large structures
- BUT: intrinsic $5-150\,\mu m$
- large track statistics at LHC allows to align positions
- aided by optical alignment: ATLAS FSI, Rasnik; CMS LAS



Software Alignment

- details of structure: whole (L1), layer/disk (L2), module (L3)
- alignment given by 6 parameters per module!
  - ▷ ATLAS Pixel+SCT: 5832
  - ▷ CMS: 16588 modules
- correlations make it a computational challenge
- again global and iterative algorithms on the market

Alignment Algorithms

Global Algorithms

- minimize residuals

$$\chi^2 = \sum_j^{\text{tracks}} \sum_i^{\text{hits}} \frac{\left(m_{ij} - h_{ij}(\boldsymbol{p}, \boldsymbol{\lambda}_j)\right)^2}{\sigma_{ij}^2}$$

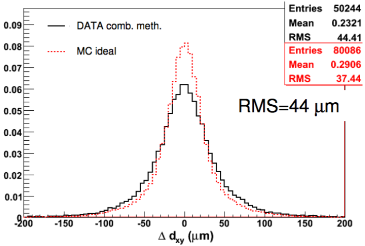
- LSE would invert huge matrix: $\dim(C) = 5N_{\text{tracks}} + 6M_{\text{modules}}$
- in practice: exploit sparse matrix structure in C
- ATLAS: invert $6M \times 6M$ matrix ("Global χ^2")
- CMS: solve $Cx + b = 0$ ("Millepede")

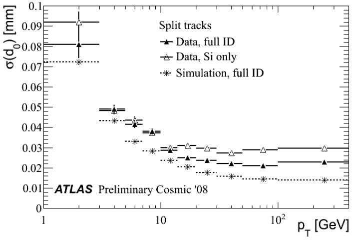
Local Algorithms

- local iterative method
  - ignores correlations
  - inverts M 6×6 matrices
  - needs many iterations
- Kalman filter algorithm
  - meas't model $h_k(\boldsymbol{p}, \boldsymbol{\lambda}_k^{k-1})$
  - no matrix inversion
  - updates all constants $\boldsymbol{p}$ and correlations: needs restriction for large $\boldsymbol{p}$

Alignment Status

- large 2008 cosmics data sample allows first tracker alignment
- performance: residuals $\sim 2\times$ intrinsic resolution
- high-level validation through cosmic track splitting method
  - ▷ latest results from trackers !
- studies with simulation from misaligned full detector
  - ▷ solve within cpu time (1-3h)
  - ▷ weak (not solvable) modes



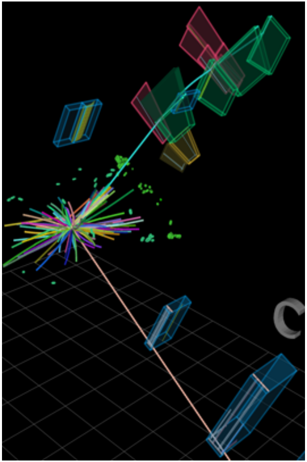


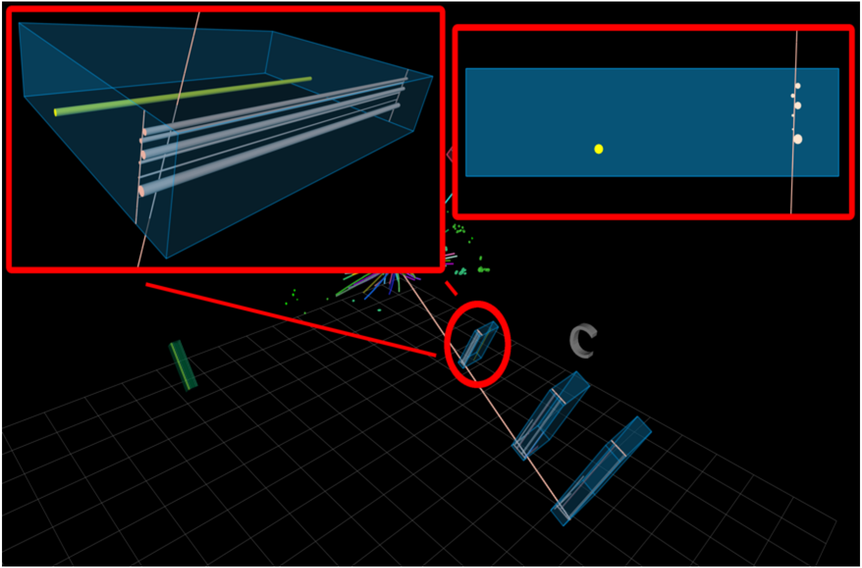
Realistic Detector Effects in First Data

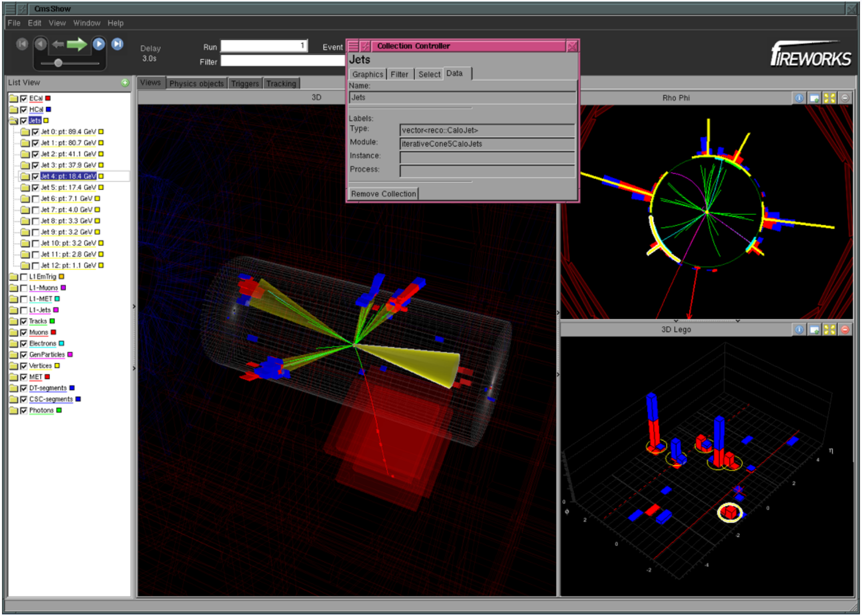
- distortions from real detector will affect all components of track and vertex reconstruction – in different ways
  - detector resolution, alignment constants, B-field, material effects, parameter tails
- understanding track reconstruction lays ground for fully understanding vertexing, b-tagging and finally physics analysis
- cosmic data have given a head start
  - especially for detector calibration and alignment
- however, reconstruction of collisions and vertices on the data will need much work to fully understand
  - means: needs manpower!

Event Displays

- indispensible for detector and software commissioning
  - online monitoring
  - understand physics events, visual debugging
- projective event displays: Atlantis, Iguana
- interactive event displays: VP1, CmsShow/Fireworks
- Virtual Point 1
  - based on Qt4 and Inventor/OpenGL
  - fully integrated in athena
- CmsShow/Fireworks
  - ROOT (GUI) + CMS-SW light






CmsShow
File Edit View Window Help
Delay
3.0s
Run
1
Event
Filter
FIREWORKS
List View
ECal
HCal
Jets
Jet 0: pt: 89.4 GeV
Jet 1: pt: 80.7 GeV
Jet 2: pt: 41.1 GeV
Jet 3: pt: 37.9 GeV
Jet 4: pt: 18.4 GeV
Jet 5: pt: 17.4 GeV
Jet 6: pt: 7.1 GeV
Jet 7: pt: 4.0 GeV
Jet 8: pt: 3.3 GeV
Jet 9: pt: 3.2 GeV
Jet 10: pt: 3.2 GeV
Jet 11: pt: 2.8 GeV
Jet 12: pt: 1.1 GeV
L1EmTrig
L1-Muons
L1-MET
L1-Jets
Tracks
Muons
Electrons
GenParticles
Vertices
MET
DT-segments
CSC-segments
Photons
Views
Physics objects
Triggers
Tracking
3D
Collection Controller
Jets
Graphics
Filter
Select
Data
Name:
Jets
Labels:
Type:
vector<reco::CaloJet>
Module:
iterativeCone5CaloJets
Instance:
Process:
Remove Collection
Rho Phi
3D Lego

Summary

- Track and vertex reconstruction should not be a 'black box' to the LHC physicist
  - ▷ hopefully not after this lecture!
  - ▷ very relevant to quality of data, esp. first data
- ATLAS and CMS are well positions for the challenging LHC track detector environment
- commissioning the track and vertex reconstruction needs manpower, skilled people and keep your eyes open!

Acknowledgements

Special thanks to:

Wolfgang Adam (HEPHY), Markus Elsing (CERN),
Sebastian Fleischmann (U Bonn), Wouter Hulsbergen (NIKHEF),
Kirill Prokofiev (CERN), Andi Salzburger (CERN)

4: Bibliography

General Track and Vertex Reconstruction

Pattern Recognition and Event Reconstruction

R. Mankel, arXiv:physics/0402039
R. Frühwirth, M. Regler, 'Data Analysis Techniques for HEP', Cambridge

Track and vertex fitting

P Avery, http://www.phys.ufl.edu/$\sim$avery/fitting.html

Track Fitting

Global Fitter:

G.Lutz, NIM A 273 (1988) 349 T

Kalman Fitter:

*R. Frühwirth, et al, NIM **A 262** (1987) 444*

Dynamic Noise Adjustment:

V. Kartvelishvili, IPRD06, Nucl.Phys B (proc)

Deterministic annealing filter:

*A. Strandlie et al, Comp.Phys.Comm. **133** (2000) 34*

Gaussian-sum filter

*R.Frühwirth, Comp.Phys.Comm. **154** (2003) 131*

Parameter Propagation and Material effects

Runge-Kutta propagation and material effects:

A. Salzburger et al, ATL-SOFT-PUB-2007-005
E. Lund et al, ATL-SOFT-PUB-2008-003
E. Lund et al, ATL-SOFT-PUB-2009-001
E. Lund et al, ATL-SOFT-PUB-2009-002

Tracking Geometry:

A. Salzburger et al, ATL-SOFT-PUB-2007-004

Fast track simulation:

A. Salzburger et al, ATL-SOFT-PUB-2008-001
CMS note submitted to CHEP 2009

Vertex Fitting

Billoir method:

*P. Billoir and S. Qian, NIM **A 311** (1992) 139-150*

Vertex Kalman filter:

*R. Frühwirth, Nucl. Ins. and Meth. **225** (1984) 352*

Adaptive vertex fitter:

*R. Frühwirth et al, NIM **A 502** (2003) 699,*
*R. Frühwirth et al, CERN-CMS-NOTE-2007-008 and J.Phys. **G 34**:N343 (2007)*

Adaptive multi-vertex fitter:

G. Piacquadio, ph.D. thesis in preparation, Freiburg 2009

Track Reconstruction Software

Atlas Event Data Model:

F. Akesson et al, ATL-SOFT-PUB-2006-004
A. Salzburger (ed.) et al, ATL-SOFT-PUB-2007-003
W. Liebig (ed.) et al, ATL-SOFT-PUB-2007-006

Atlas NewTracking and Performance:

A. Salzburger (ed.) et al, ATL-SOFT-PUB-2007-007
A. Salzburger (ed.) et al, ATL-INDET-PUB-2008-002

CMS tracking and performance:

B. Mangano, CERN-CMS-CR-2008-100

Vertex Reconstruction Software

Vertex Kalman filter (robust fitter):

*T. Speer et al, NIM **A 566**:149-152,2006 (Time '05)*

Vertex fitting in CMS:

T. Speer et al, CERN-CMS-NOTE-2006-032

Kinematic vertex fit and decay chain fit

K. Prokofiev et al, CHEP 2004 411

Gaussian-sum filter for vertex reconstruction

T. Speer et al, CHEP 2004 415.

Vertex reconstruction in ATLAS

E. Bouhova-Thacker et al, ATL-INDET-PUB-2009-001