Track Reconstruction in the CMS Tracker

Frank-Peter Schilling (CERN/PH)
D-CMS Meeting
Hamburg, 21/02/2006

• Introduction
• Baseline algorithm: Kalman Filter
• Advanced algorithms
• Road Search
• Recent developments
• Alignment
Outline

- **Baseline algorithm (Combinatorial Kalman Filter)**
  - Algorithm description
  - Performance
  - Special cases (regional/partial tracking (HLT))

- **Advanced algorithms: Adaptive Filters**
  - Deterministic Annealing Filter, Multi-track Filter
  - Gaussian Sum Filter

- **Alternative approach: RoadSearch**

- **Recent Developments**
  - Pixel-less tracking
  - Inclusion of hit pairs in overlap regions
  - (Tracking for cosmics)
  - (Reconstruction of V0’s)

- **A few words on track-based Alignment**
Introduction

- High hit resolution and granularity
- A few (10+3 barrel layers), but precise (10-50μm) measurements
- 4T solenoidal magnetic field: Pt>0.6 GeV to reach outer layer
- Large track multiplicity (pileup!), most hits from low p junk
- A lot of material: multiple scattering!

Physics Requirements:
1) High efficiency
2) Momentum resolution
3) Impact parameter resolution

For details on HW, see talk of K.Klein
Material Budget

- Radiation lengths
- Interaction lengths

- Multiple scattering
- Bremsstrahlung for electrons
- Hadronic interactions
- Kills tracks!
5(8) layer crossing probability: muons

- Muons mostly unaffected
5(8) layer crossing probability: pions

- Pions suffer substantial losses
- Must consider material effects during pattern recognition!
- No sense to track outside-in (for primary tracks):
  - Up to ~20% don’t reach outside!
Baseline Algorithm

Standard algorithm in CMS: Combinatorial Kalman Filter

- Equivalent to global least-squares minimization
  - optimal estimator if model is linear and random noise is gaussian
  - For non-linear models / non-gaussian noise, optimal linear estimator
- Local: one track reconstructed at a time
- Recursive: parameters updated with each successive hit
- Energy loss / multiple scattering can be taken into account

Modular Building blocks for track reconstruction:

- Starting point
  - Seed generation
- Pattern recognition
  - Trajectory building
- Reduction of remaining combinatorics
  - Trajectory cleaning
- Parameter estimation
  - Track fitting and smoothing
Seed generation

- Defines the starting point for pattern recognition
  - Seeds should constrain all 5 track parameters:
    - reasonable search region
    - sufficiently close to true values: linear regime
- Needs to be fast and efficient
- Baseline seeding:
  - innermost layers: Pixel
  - High track density compensated by high granularity
- Fast hit pair finding: start with primary hit in outermost Px layer
  - Can be restricted to region of interest
Seed generation (cont.)

- Complemented by 2nd hit in other layers and vertex constraint ($\Delta z=30\text{cm}$, $\Delta R=1\text{mm}$) → 2 2d hits + vertex
- Fast geometrical search
- Hit finding efficiency ~100%
- Time small (~10%) w.r.t full track reco

- Alternative seedings (not discussed here):
  Outside-in (e.g. photon conversions)
  External seeds (ECAL or Muon + Vertex)
Trajectory Building

- Based on Kalman filter
  - Simultaneous trajectory extension and hit selection
- Propagation from layer to layer, accounting for energy loss and multiple scattering (requires efficient layer navigation)
- Propagation of track to next layer, search for compatible hits
  - New trajectories constructed with updated parameters + errors for each compatible hit
  - In addition one further trajectory without hit to account for inefficiencies
  - All trajectories propagated to next layer
- Procedure repeated until outermost layer is reached
- To avoid bias, all trajectories propagated in parallel
- Parameters
  - Max. number of candidates retained per step (ranked in $\chi^2$)
  - Number of missing hits
Example: Pattern recognition in the Barrel

- Most seeds are composed of a hit pair in Px layers 1+2
- 1st step: Propagation to Px layer 3
  - Trajectory not yet well defined; uncertainties \( \sim 500\) (80) \(\mu m\) in \( r_\phi (rz) \)
  - Few fakes, mostly 1 (+1 invalid) compatible hits, thanks to Pixel granularity

\[ \text{Number of compatible hits} \]

\[ \text{Number of trajectories} \]

Fakes
Pattern recognition in the Barrel (cont.)

- 2\textsuperscript{nd} step: Propagation to TIB layer 1
  - Uncertainties of predicted state increases (800 / 400 \(\mu\text{m}\)) due to large extrapolation distance (~13 cm) and small lever arm of initial trajectories (~6 cm)
  - More compatible hits due to bigger occupancy in strip detector

- From TIB layer 2 on, uncertainties reduced (trajectories ~well defined); many trajectories with spurious hits discarded
• Uncertainties in $r_{\phi}$ and $rz$ planes per layer
• Fraction of trajectory candidates with at least one spurious hit
Pattern recognition in endcap

- Navigation more complex than in barrel
  - E.g. for high $\eta$ tracks leaving PX disk 2, all 3 TID disks and 3 of TEC disks could be compatible: many trajectory candidates
  - Large propagation distances possible: more spurious hits
- Once in TEC, situation improves

- Compatible hits when leaving PX disk 2
- Fraction of trajectories with spurious hit
Trajectory cleaning

- Resolve ambiguities to avoid double counting of tracks

- Ambiguities may arise from
  - One seed leading to >1 trajectory candidates
  - A given track is reconstructed starting from different seeds

- Based on fraction of shared hits $f$:
  - $f = \frac{N_{\text{shared}}}{\min(N_1,N_2)}$
  - If $f > 0.5$ for a given pair of tracks, the one with the smaller number of hits is discarded (if $N_1=N_2$, the one with the bigger $\chi^2$)

- Cleaning applied twice:
  - On all tracks resulting from a single seed
  - On all tracks from all seeds
Track fitting and smoothing

- For each trajectory, building stage results in collection of hits and estimate of track parameters, but
  - full information only available at last hit
  - estimate can be biased by constraints applied at seeding stage
  - Therefore, a re-fit is performed, implemented as a combination of a Kalman filter and smoother

- Filter: is Initialized at innermost hit with seeding estimate
  - Covariance matrix scaled by large factor to remove seeding bias

- Iterative processing of hit list:
  - Re-evaluation of hit position estimate
  - Update of track parameters and covariance matrix
  - Trajectory propagation, modification of parameters and cov. matrix according to estimates for energy loss and multiple scattering

- Smoothing: 2\textsuperscript{nd} filter outside-in
  - Smoothed states: weighted mean of forward and backward fits
Performance: Efficiency for Muons

- **Two definitions of efficiency used:**
  - Algorithmic efficiency: efficiency of pattern recognition (defined wrt sim. Tracks which are reconstructable: no of PX/strip hits, pt, etc)
  - Global efficiency: efficiency for all tracks with pt>pt-cut and production vertex inside beam-pipe (includes acceptance, hit eff. etc)

- **Cuts:** Pt>0.9 GeV. at least 8 hits. at most one missing hit

- Pattern recognition fully efficient
  - Loss of acceptance at large $\eta$
  - $\eta$~0: alignment of Px ladders
Performance: Efficiency for Pions

- Algorithmic efficiency reduced at low pt due to elastic scattering
- Lower global efficiency due to hadronic interactions in tracker material (tracks don’t reach outside)
Tuning: Efficiency vs fake rate for b jets

- Algorithmic eff ~95%
- Global eff ~80-90%
- Fake rate <1%

- Tuning of normalized $\chi^2$ and/or N-hit
- Fake rate explodes for N-hit<7
Performance: Resolutions (muons)

- Transverse impact parameter $d_0$
- Longit. impact parameter $z_0$

- $10\mu$ resolution in $d_0$ at 100 GeV: pixel hit resolution
- Degrading at lower $p_t$ due to multiple scattering
- $z_0$ resolution improving from $\eta=0$ up to $\sim0.5$ due to widening of $P_x$ clusters, improving resolution
Resolutions (cont.)

- Azimuthal angle $\phi$
- Polar angle: $\cot(\theta)$
• **Transverse momentum**

- Pt resolution ~1-2% in barrel
- At 100 GeV, tracker material accounts for 20-30% of Pt resolution
- At lower Pt, dominated by multiple scattering
- Small Pt bias in endcap due to B-field inhomogenities not (yet) accounted for
Special modes

- What has been shown so far represents the current CMS default
- This is what you will get (also on DST) if you request
  `RecQuery("CombinatorialTrackFinder")`
- **Documentation:** CMS-Note-2006/041, CMS-Note-2006/026, PTDR1

Special modes:

- **limited reconstruction to save CPU (HLT):**
- **Regional reconstruction**
  - Region of interest (ROI) derived from L1 trigger
  - Seeding and pattern recognition limited to this region
- **Partial reconstruction**
  - Don’t need full resolution e.g. for isolation
  - Stop trajectory building when errors are small enough
Partial reconstruction (HLT)

- Pt resolution
- Impact parameter resolution

![Graphs showing Pt resolution and impact parameter resolution for different ranges of p_t and η.]

Full reconstruction

- Reasonable precision already with 5 hits

(see DAQ TDR)
Advanced algorithms

- **Advantages of default reconstruction**
  - Based on simple, well-known algorithms
  - Efficient and robust
  - Few parameters
  - Works (with retuning) even for Heavy Ion collisions

- **Drawbacks:**
  - Limit on number of candidates in trajectory building is compromise between speed and risk to loose right track
  - No differentiation between noise and hits from other tracks
  - Hard hit assignment sub-optimal in dense environments

- **Advanced algorithms: Adaptive filters**
  - Avoid hit assignment errors at high track density (1+2)
  - Consider non-gaussian tails (e.g. Bremsstrahlung) (3)
  1. Deterministic Annealing Filter (DAF)
  2. Multi-track Filter (MTF)
  3. Gaussian Sum Filter (GSF)
Adaptive Filters: DAF and MTF

- Dense track environments e.g. in b- or tau-jets:
  - Hit degradation due to contamination from nearby tracks
  - Large hit multiplicity in search window: wrong hit assignment
- Try soft hit assignment during pattern recognition

- Deterministic Annealing Filter (DAF): CMS-IN-2003/043
  - Iterative Kalman Filter
  - Competition between hits on same surface to belong to track
  - Soft Assignment probabilities 0…1
  - Fitter and smoother iterated until convergence
  - To avoid local minima use annealing

- Multi-track filter (MTF): CMS-IN-2003/042
  - Extension to concurrent multi track fit
  - Competition between tracks and hits (assignment prob. matrix)
  - Each hit can belong to each of several tracks
Deterministic Annealing Filter (DAF)

- B-Jets in the barrel, Pt=200 GeV
- Transverse impact parameter resolution

<table>
<thead>
<tr>
<th>RMS</th>
<th>0.279E-02</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\chi^2$/ndf</td>
<td>151.9 / 83</td>
</tr>
<tr>
<td>P1</td>
<td>-0.1110E-03 ± 0.2429E-04</td>
</tr>
<tr>
<td>P2</td>
<td>308.8 ± 7.327</td>
</tr>
<tr>
<td>P3</td>
<td>0.1219E-02 ± 0.2615E-04</td>
</tr>
<tr>
<td>P4</td>
<td>15.59 ± 1.570</td>
</tr>
<tr>
<td>P5</td>
<td>0.5345E-02 ± 0.2260E-03</td>
</tr>
</tbody>
</table>

$\sigma = 25 \mu m$

<table>
<thead>
<tr>
<th>RMS</th>
<th>0.297E-02</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\chi^2$/ndf</td>
<td>91.09 / 86</td>
</tr>
<tr>
<td>P1</td>
<td>-0.1166E-03 ± 0.2688E-04</td>
</tr>
<tr>
<td>P2</td>
<td>264.6 ± 7.213</td>
</tr>
<tr>
<td>P3</td>
<td>0.1156E-02 ± 0.2909E-04</td>
</tr>
<tr>
<td>P4</td>
<td>27.08 ± 1.876</td>
</tr>
<tr>
<td>P5</td>
<td>0.5165E-02 ± 0.1550E-03</td>
</tr>
</tbody>
</table>

$\sigma = 30 \mu m$

- DAF: tails are reduced
Gaussian Sum Filter (GSF)

- Linear least square estimators (==KF) only optimal in linear systems with Gaussian measurement errors and process noise.
- GSF used for electron reconstruction in CMS
  - Bremsstrahlung highly non-gaussian

Basic idea:
- Non-linear generalization of KF
- Describe non-gaussian probability density functions (pdf’s) by mixture of multivariate Gaussian pdf’s
- Main component: Core of distributions
- Tails: One or more additional Gaussians
- Weighted sum of several Kalman Filters, run in parallel
- At each step, convolution of state vector mixture with energy loss mixture ⇒ exponential rise of number of components
- Way out: collapsing of components which are “close”
Gaussian Sum Filter (cont.)

- Application of GSF to electron reconstruction
- Bethe-Heitler model:
  - $f(z)$: pdf of electron energy loss
  - $t$: path length (units of rad. length)
  - $z$: remaining energy fraction
  
  $$f(z) = \frac{[-\ln z]^{c-1}}{\Gamma(c)} ; \quad c = \frac{t}{\ln 2}$$

- Fit parameters of Gaussian mixture to known energy loss distribution
GSF electron reconstruction (cont.)

- 6 component mixture for energy loss
- Number of components limited to 12 in fit
- Momentum residuals:
- \( q/p \) pulls (both at TiP surface):

- Clear improvement in momentum resolution
  (but similar to KF at high Pt)
RoadSearch Algorithm

- **New development pursued by the USCMS group**
  - Complementary to Kalman Filter
  - Robust tracks, in particular at start-up
  - Always good to have alternative

- **Basics of RoadSearch algorithm:**
  - Tracker subdivided in “Rings” in phi at given \((r,z)\)
  - Seeds built from hits in predefined inner and outer seed ring combination (RoadSeed) passing \(\Delta\phi\) cut
  - RoadSeed: all lin. Extrapolations of inner/outer seed ring combinations compatible with beam spot
  - Collect hits (cloud) in window around trajectory in road
  - Clean hit collection; final track fit
Efficiency for single muons with $P_T=100$ GeV

- Better RS efficiency in fwd region (no PX requirement)
- Meanwhile compensated by KF Pixelless seeding (see later)
RoadSearch (cont.)

- Efficiency for $H \rightarrow ZZ \rightarrow ee\mu\mu$
- RS inefficiency at low pt, mainly in barrel (roads too narrow?)
- Timing:

<table>
<thead>
<tr>
<th>sample</th>
<th>mean number of tracks</th>
<th>time per event</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CTF</td>
<td>RS</td>
</tr>
<tr>
<td>single muon</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$h \rightarrow ZZ \rightarrow ee\mu\mu$</td>
<td>33.7</td>
<td>29.8</td>
</tr>
<tr>
<td>$W \rightarrow \mu\nu + pileup$</td>
<td>43.3</td>
<td>40.7</td>
</tr>
<tr>
<td>$b$ jets ($120 \leq p_T \leq 170$ GeV)</td>
<td>60.0</td>
<td>56.2</td>
</tr>
</tbody>
</table>

- No fake rate studies yet
- Work in progress
Other recent developments

In addition to the new RoadSearch approach, extensions and improvements of existing KF tracking are ongoing:

- Tracking with overlaps
- Tracking without Pixels
- Cosmics tracking (see talk of M. Stoye)
- (V0 tracking)

... and of course porting of the track reconstruction to the new software framework CMSSW
Tracking with overlaps

- Standard CTF uses only one hit per layer, even if tracks cross overlap region between two modules, leaving two hits.
- Tracking with overlaps potentially interesting e.g. for alignment!

- ~0.5…1 more hits per track found
- Performance similar (efficiency, resolution)
Pixel-less tracking (seeding)

- Standard tracking uses seeds from Pixel detector
- At the CMS start-up, there will be no pixel!
- Implement alternative seeding, using the innermost layers of strip tracker

- SiliconStrip hits have position error bigger than silicon ones.
- The innermost SiStrip layer is farther from the beam line than the pixel one.

⇒ Bigger number of trajectory candidates
In order to reduce the number of seeds generated and the number of calls to the trajectory builder:

Optimization of the layers set used during seed generation

The best arrangement (between seed efficiency and track reconstruction speed) resulted in this layers combination:

- TIB1 and TIB2 layers
- 2 innermost rings of TID1 + complete TID2 + 2 innermost rings of TID3
- 2 innermost rings of TEC2 and TEC3
Pixel-less seeding: performance

- Higher efficiency in fwd region
- mixed seeding coming!
- Impact parameter resolution degraded

Pt resolution similar (strip leverarm)

5000 events have been simulated without pixel “dead” material. There are no apparent improvements in the resolution.
A few words on Alignment

CMS tracker consists of ~16,000 indiv. Modules
- Knowledge of position and orientation should be comparable or better than intrinsic resolution
- Laser alignment: only for of larger structures in TIB / TOB / TEC
  - See talk by M. Thomas
- Determination of ~100k alignment parameters to 10μ necessary
- Only possible with track-based alignment!

Three algorithms presently studied in CMS:
- Kalman filter (Vienna, Aachen), CMS-Note-2006/022
  - See talk by M. Weber
- Millepede (Hamburg), CMS-Note-2006/011
  - See talk by M. Stoye
- HIP (Helsinki, CERN), CMS-Note-2006/018

Summary of work documented in PTDR Vol. 1, section 6.6
Status: Software

- Simulation of Misalignment
  - Development of two “Misalignment scenarios” (short- and long-term)
  - Documented in CMS-Note-2006/008
  - Used for many PTDR physics studies, see also CMS-Note-2006/029

- Common Software Framework for track-based Alignment
  - Presently implemented in ORCA
  - Documented in CMS-IN-2005/051
  - Used for interfacing alignment algorithms to CMS software

- Software developments relevant for alignment
  - Track refit at DST level (~25 ev/sec), if only relevant tracks are refitted, e.g. $\mu$ from $W \rightarrow \mu \nu$
  - miniDST format for alignment (retain only relevant tracks): improvement in performance (~75 ev/sec) and disk space (~1/100) precursor of alignment HLT stream?!

Only these make large scale alignment possible with reasonable turnaround!
HIP Algorithm (Helsinki, CERN)

- Linearized $\chi^2$ minimisation
- Derivatives of impact point on sensor w.r.t. alignment parameters
- No correlations between sensors, no large matrices
- Implemented for indiv. sensors as well as for composite objects (rods, ladders etc.)

- Alignment of 720 Pixel barrel modules

- Short term misalignment scenario *10 (CMS startup)

Current status doc. in CMS-Note-2006/018
Impact of misalignment

- Misalignment implemented at reconstruction level (ORCA) by moving/rotating modules/layers etc
- Can be studied even at DST level using track refitter

Two misalignment scenarios developed for PTDR studies:
- "first data" scenario
  - Situation at LHC start-up (first few 100 pb-1)
  - Construction information, LAS, pixel aligned with tracks
- "long term" scenario
  - After first few fb-1 have been taken
  - Tracker aligned at the sensor level to \( \approx 20 \ \mu \text{m} \)

<table>
<thead>
<tr>
<th></th>
<th>Pixel</th>
<th></th>
<th>Silicon Strip</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Barrel</td>
<td>Endcap</td>
<td>Inner Barrel</td>
<td>Outer Barrel</td>
</tr>
<tr>
<td>First Data Taking Scenario Modules</td>
<td>13</td>
<td>2.5</td>
<td>200</td>
<td>100</td>
</tr>
<tr>
<td>Ladders/Rods/Rings/Petals</td>
<td>5</td>
<td>5</td>
<td>200</td>
<td>100</td>
</tr>
<tr>
<td>Long Term Scenario Modules</td>
<td>13</td>
<td>2.5</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>Ladders/Rods/Rings/Petals</td>
<td>5</td>
<td>5</td>
<td>20</td>
<td>10</td>
</tr>
</tbody>
</table>

CMS-Note-2006/008
CMS-Note-2006/029
Impact of misalignment

- Single muons with $Pt=100$ GeV (typical scale for LHC physics, resolutions not dominated by multiple scattering)

- Inefficiency in barrel, if alignment unc. not added to meas. error
- Worse in TID region (larger initial uncertainty from mounting)
- $Pt$ resolution worse by factor $\sim 5$ for short-term scenario
Impact of Misalignment

- Transverse and longit. Impact parameter resolution

<table>
<thead>
<tr>
<th>$\sigma(d_0)$ vs $\eta$, $p_T = 100$ GeV/c</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>$\sigma(d_0)$ [\mu m]</strong></td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0.5</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1.5</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>2.5</td>
</tr>
</tbody>
</table>

- $d0$ resolution $\sim 9, 35, 20 \ \mu m$ (ideal, short term, long term)
- Note: Pixel detector assumed aligned even in short term scenario
Impact of Misalignment

- No b-tagging performance with currently assumed assembly precision for pixel

- Fast Pixel alignment mandatory (also to provide reference for strip alignment)!

![Graph showing impact of misalignment with different alignment conditions](image)
Conclusions

- CMS has (had?) a modular / oo oriented track reconstruction
  - Details of detector geometry hidden from reconstruction
  - Modular structure allows easy exchange of components
- Efficient baseline track reconstruction based on Kalman filter
- Advanced algorithms available (building upon baseline KF)
  - Soft assignment algorithms
  - Gaussian sum filter
- Recent (ongoing) developments
  - Overlaps
  - Pixel-less seeding
  - Cosmics
  - Tracking of V0’s
  - RoadSearch
  - Porting to CMSSW!
- Alignment crucial for physics performance of tracker!

Additional manpower in tracking/alignment welcome!!