Track and Vertex Reconstruction in CMS
Baseline and advanced algorithms

Wolfgang Adam
Inst. f. Hochenergiephysik der
Österr. Akademie d. Wissenschaften

DESY Computing Seminar, 11/7/05

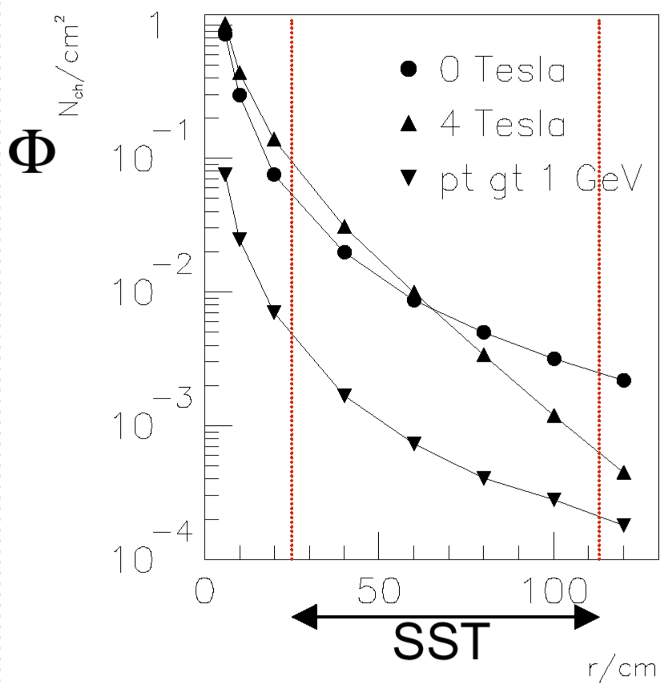
LHC Tracking Environment

pp-collisions

- Collision rate 40 MHz; $\sqrt{s}$ = 14 TeV
- Design luminosity 10^{34} $cm^{-2}s^{-1}$
  Low luminosity scenario 2x1033 $cm^{-2}s^{-1}$
- Typically 20 events pile up superimposed on signal
- Typically 2000 Tracks / bunch crossing

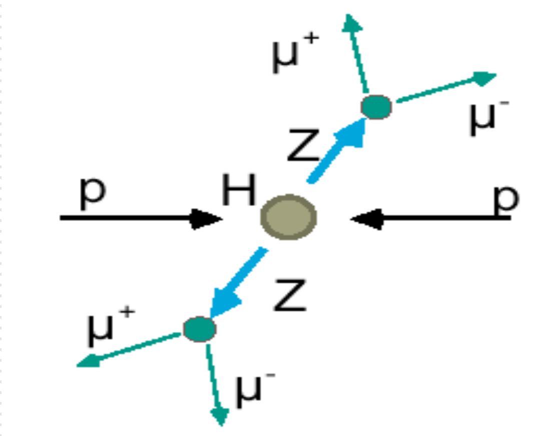
Charged track density

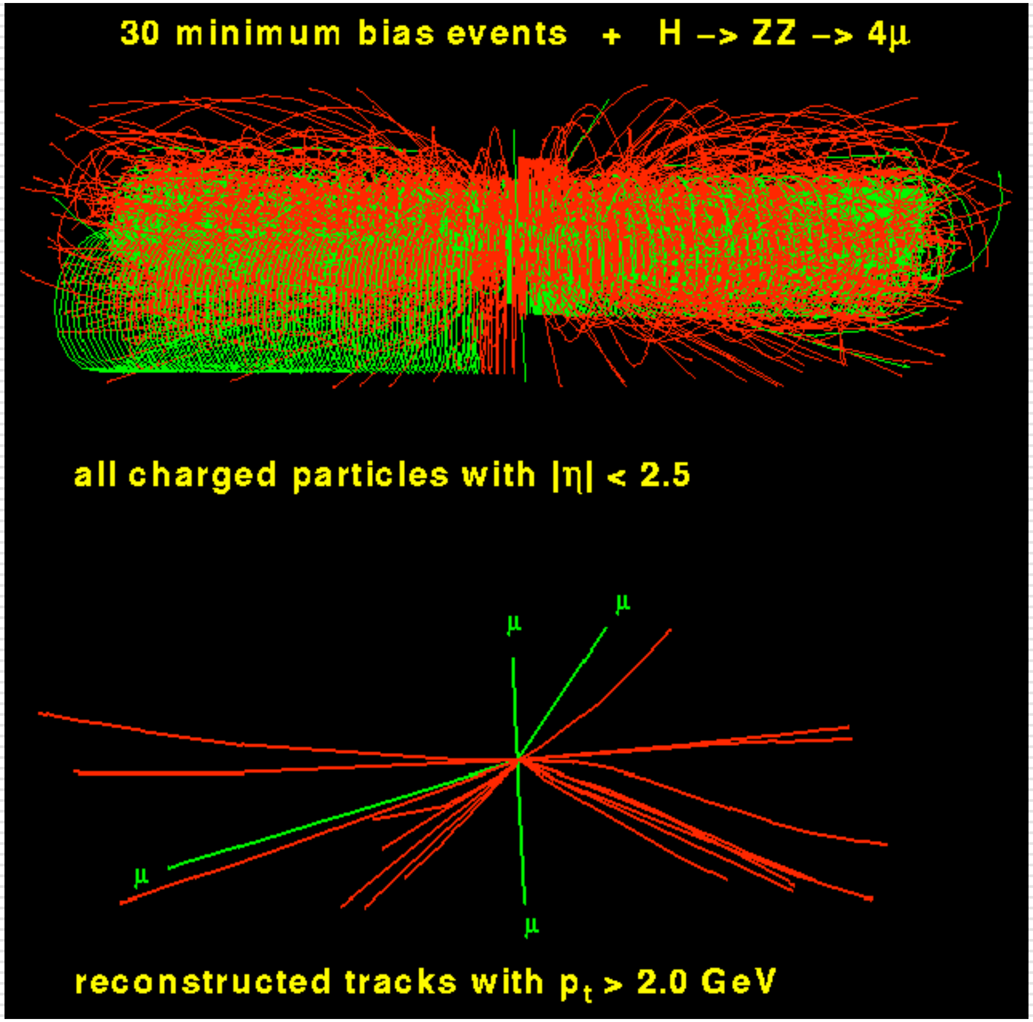
- In solenoidal field of 4T
  - ~ 1 / cm^2 / 25ns at r = 10cm
  - ~ 0.1 / cm^2 / 25ns at r = 25cm
  - ~ 0.01 / cm^2 / 25ns at r = 60cm



LHC Tracking Environment

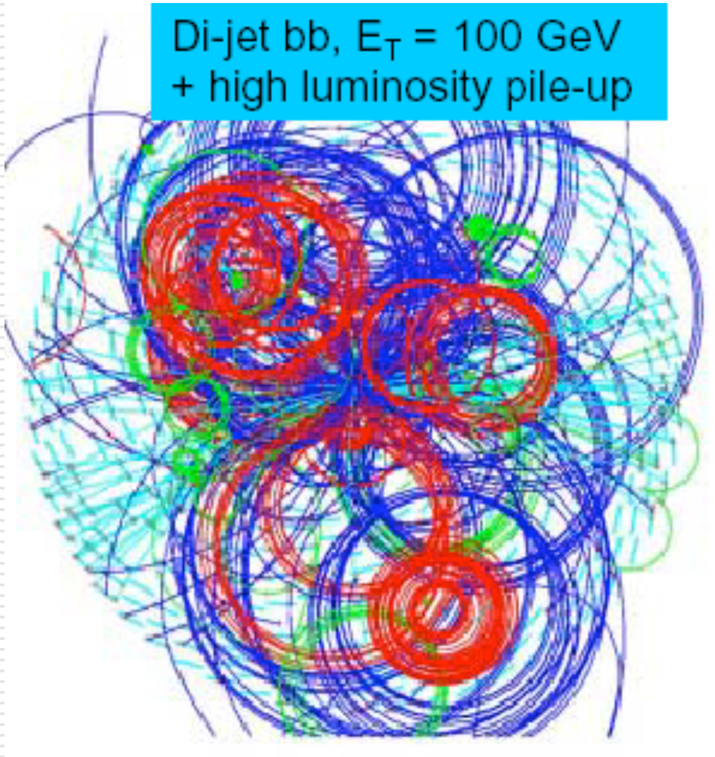
Example:



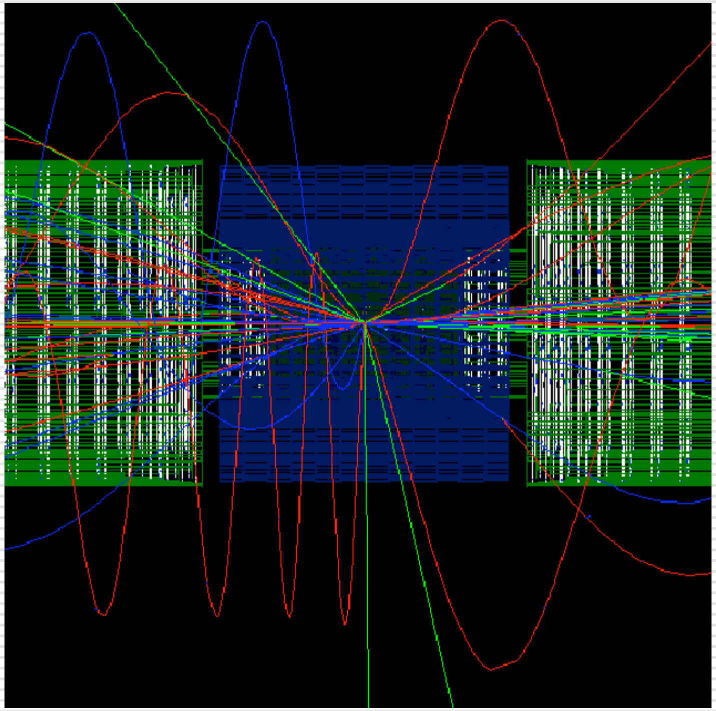


at high luminosity

LHC Tracking Environment



180GeV Higgs event



LHC Tracking Environment

Implications

- **High occupancy**
  - Pile up events
  - Possible remnants of previous crossings (for slow detectors)
  - Spiraling tracks
- **Radiation damage**
  - Potential degradation of efficiency and resolution

Physics requirements

- **Highly efficient reconstruction of tracks**
- **High momentum resolution for**
  - Mass reconstruction
  - Charge identification
- **High spatial resolution at impact point**
  - Secondary vertex reconstruction
  - Heavy flavour tagging

The CMS Tracker

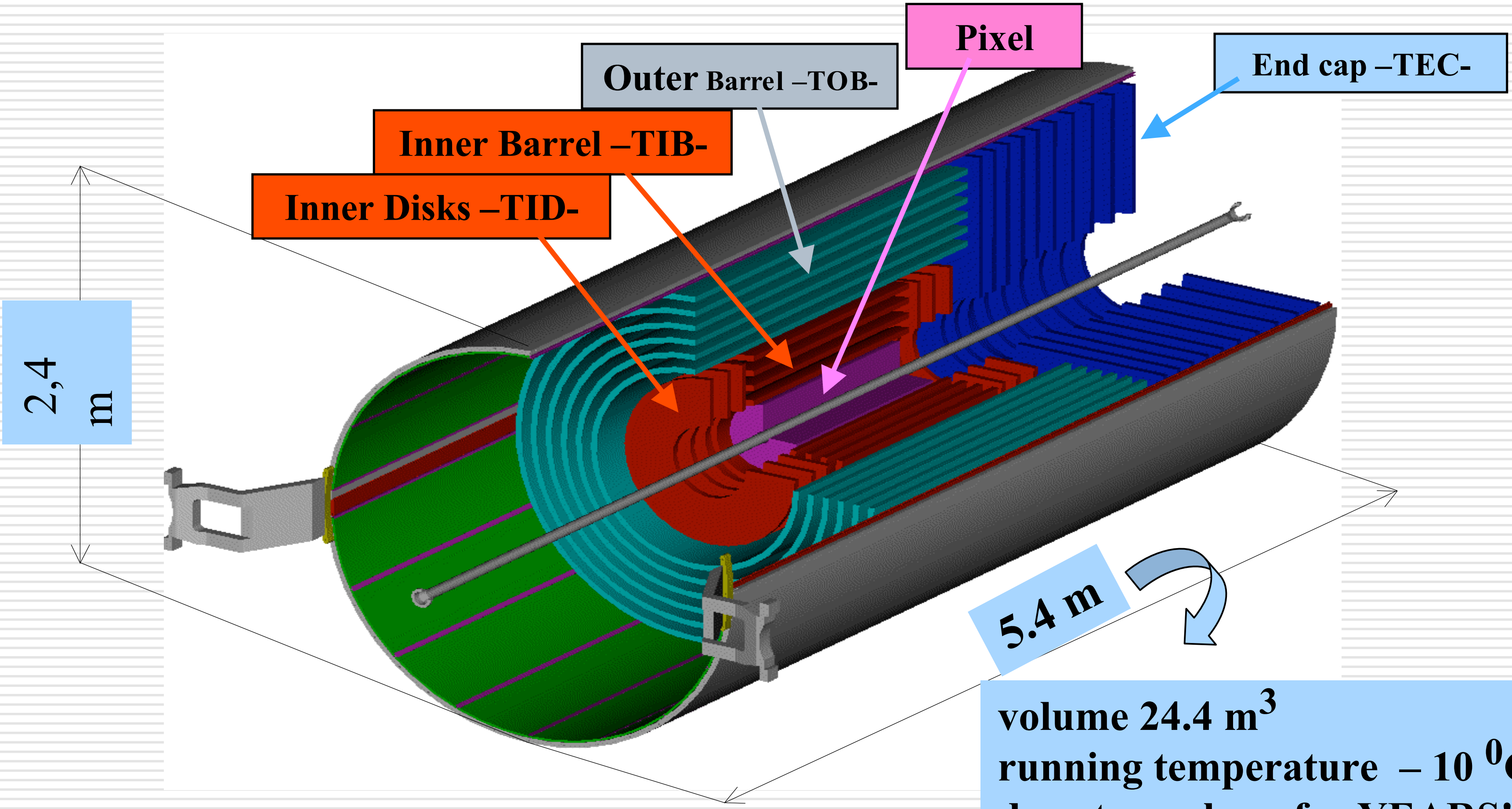
Full-Silicon solution for the inner tracking

- High hit resolution and 2-track separation
- Fast response
- High granularity
- High redundancy
- > 220 m^2 of silicon sensors

Layout

- Classical scheme with cylindrical barrel and planar end cap disks
- Pixel detectors close to interaction point
- Single sided strip detectors with increasing pitch in outer layers
- Stereo-configuration of strip detectors for a subset of layers

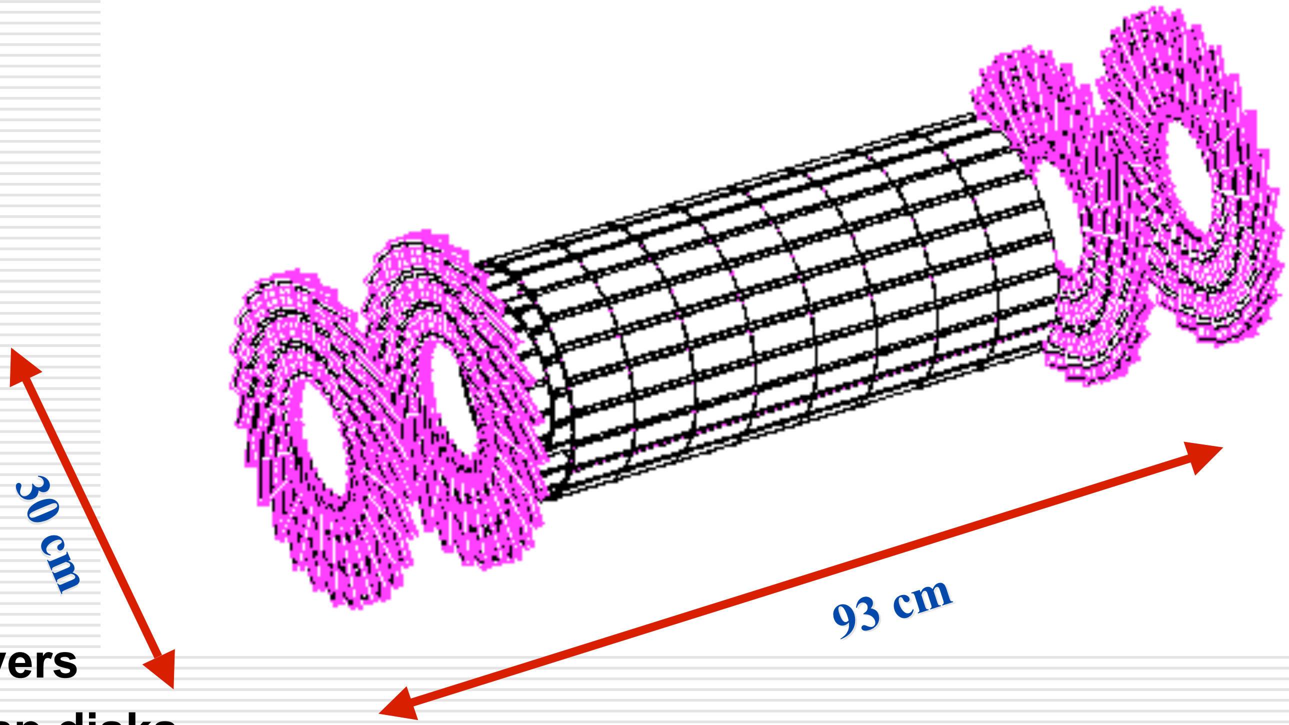
The CMS Tracker



volume 24.4 m^3
running temperature – 10 ^{0}C
dry atmosphere for YEARS!

The CMS Tracker

Pixel detector



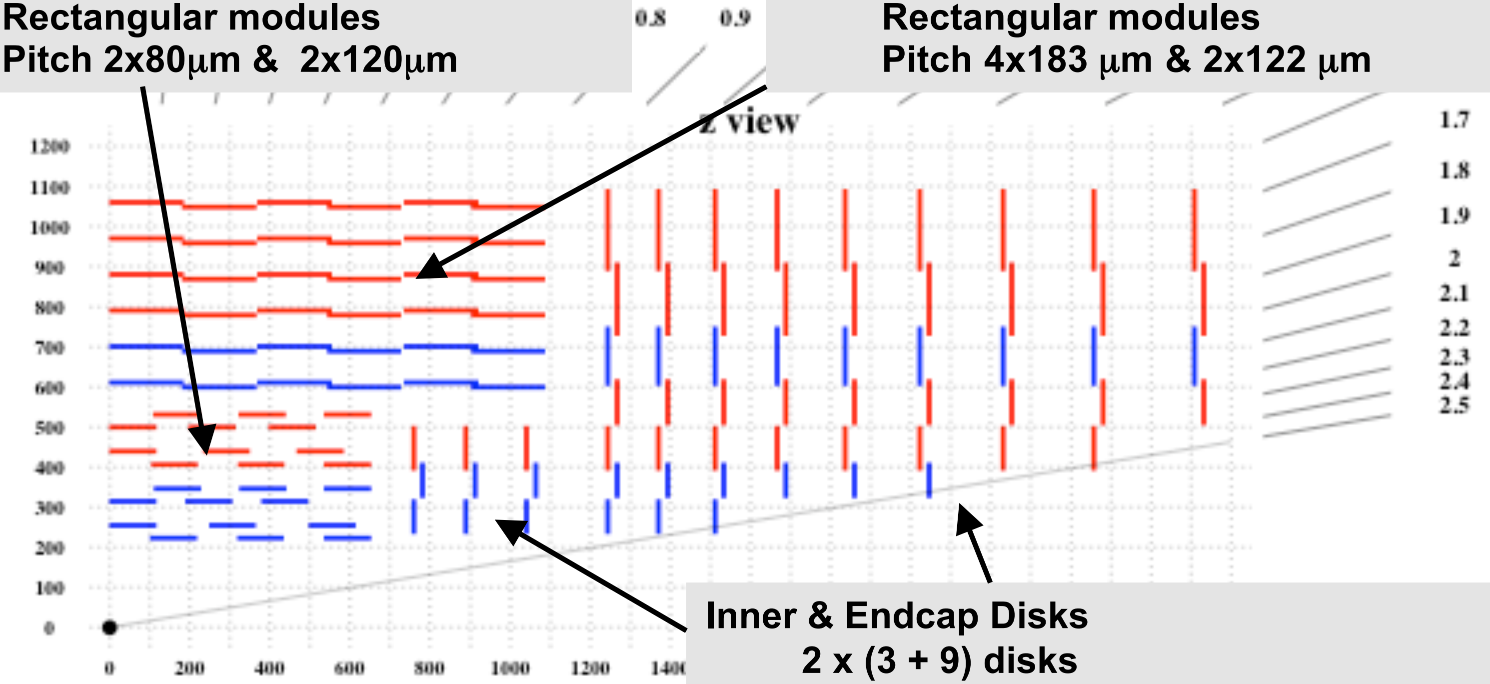
- 3 barrel layers
- 2 x 2 endcap disks

- Pixel size 100x150 μm^2
- > 5x10^7 channels

The CMS Tracker

Inner Barrel
- **4 layers (2 in stereo configuration)**
- **Rectangular modules**
- **Pitch 2x80μm & 2x120μm**

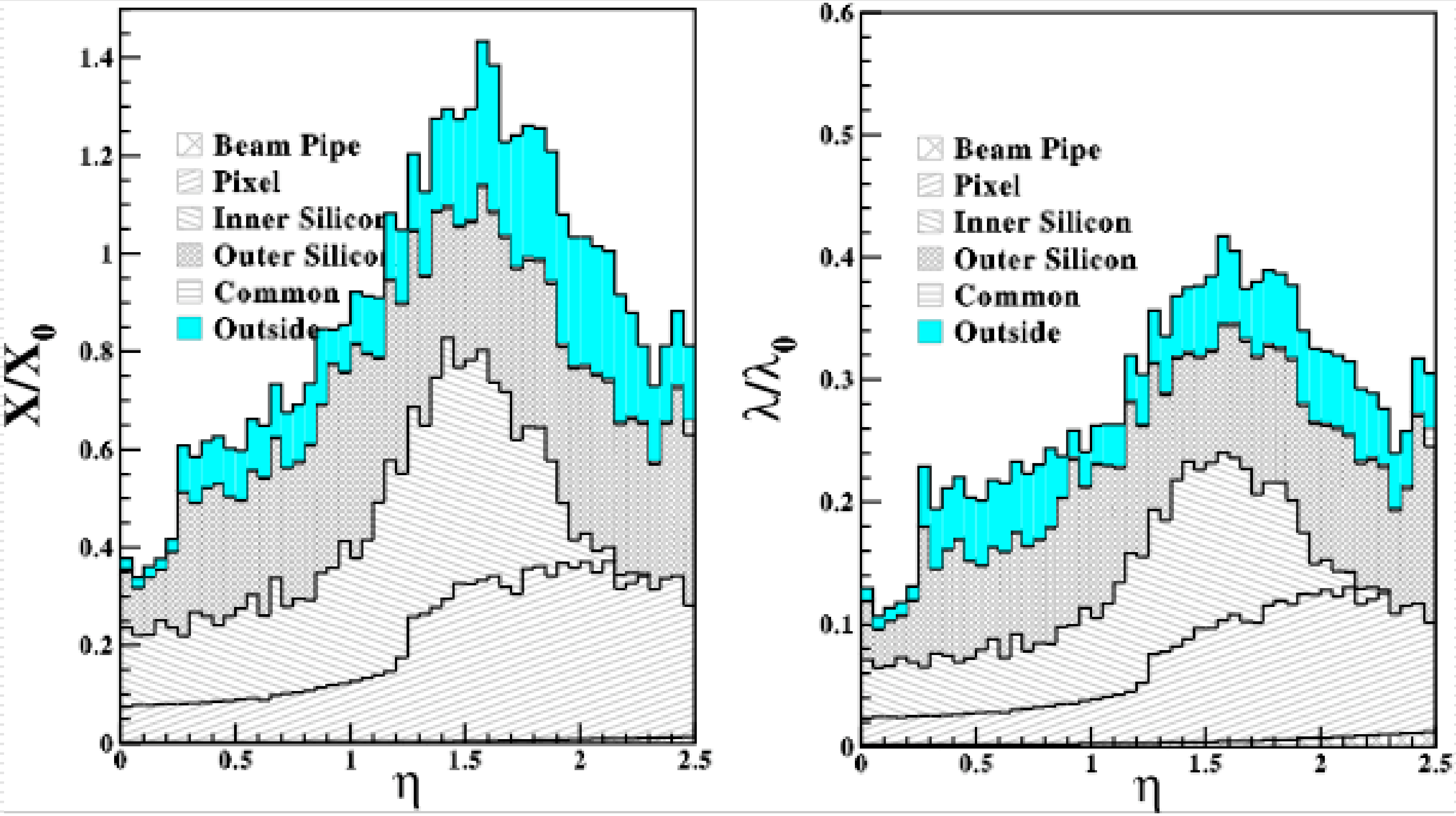
Outer Barrel
- **6 layers (2 in stereo configuration)**
- **Rectangular modules**
- **Pitch 4x183 μm & 2x122 μm**



Inner & Endcap Disks
- **2 x (3 + 9) disks**
- **Wedge-shaped modules in 7 rings**
- **3 rings in stereo configuration**
- **Mean pitch ranges from ~95μm to ~185μm**

The CMS Tracker

Distribution of material



Multiple scattering
Bremsstrahlung for electrons

Hadronic interactions:
shorter pion tracks

Definitions & Utilities

Track model

- **Inside the tracker volume: ~ uniform magnetic field**
  - Trajectories (segments) described by helices
- **Outside the tracker volume:**
  - Need numerical propagation using a field map

- **State of a trajectory with one constraint (reference surface) described by 5 parameters.**
  - Global:
    - Q/p, two orthogonal co-ordinates in the plane perpendicular to the trajectory, global phi (azimuthal) and theta (w.r.t. beam axis) angle only used for description of the covariance matrix in a global frame
    - Parameters: cartesian
  - Local (on a plane):
    - In cartesian co-ordinate system (z normal to plane):
    - Q/p, dx/dz, dy/dz, x, y

Definitions & Utilities

Propagation

- **Different propagators are implemented and easy to exchange**
  - “Analytical” with helix model
  - Detailed propagation as in detector simulation (“GEANE”)
  - Runge-Kutta + volume navigation in preparation
- **For reconstruction within tracker volume**
  - Propagation to planes and cylinders
  - Assumes constant magnetic field within one propagation step
  - Helix model - analytical solution to error propagation (numerical differentiation available for cross checks)
  - Simplified description of the material distribution
    - Assumed to be concentrated on active layers
    - Takes into account multiple scattering and energy loss (radiative for electrons, by ionization for all other particles)

Definitions & Utilities

Measurements

- **(OO representation) of each detector module can provide**
  - Clusters of signal channels
  - Estimation of position and its covariance matrix
    - On a reference surface within the silicon (Lorentz angle!)
    - Virtual “double” modules in stereo layers provide already combined hits
    - A refined estimate can be obtained using a guess of the track direction
  - Transformation from the “strip” / “pixel” system to the local cartesian system
  - Projection matrix for the conversion from the (local) track parameter space to the space of the measurement

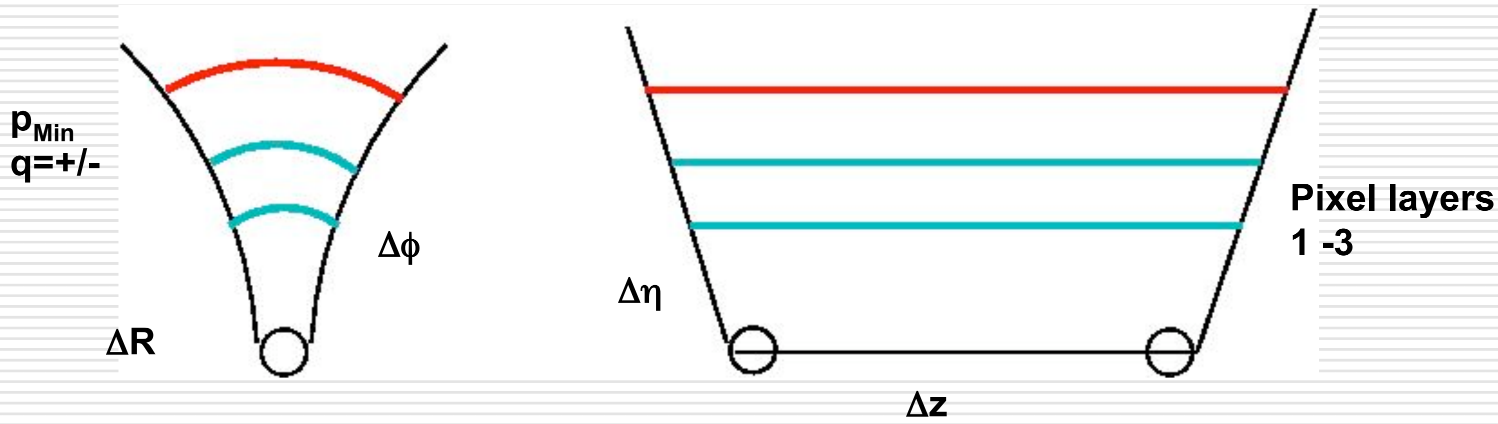
Track Reconstruction - Baseline

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

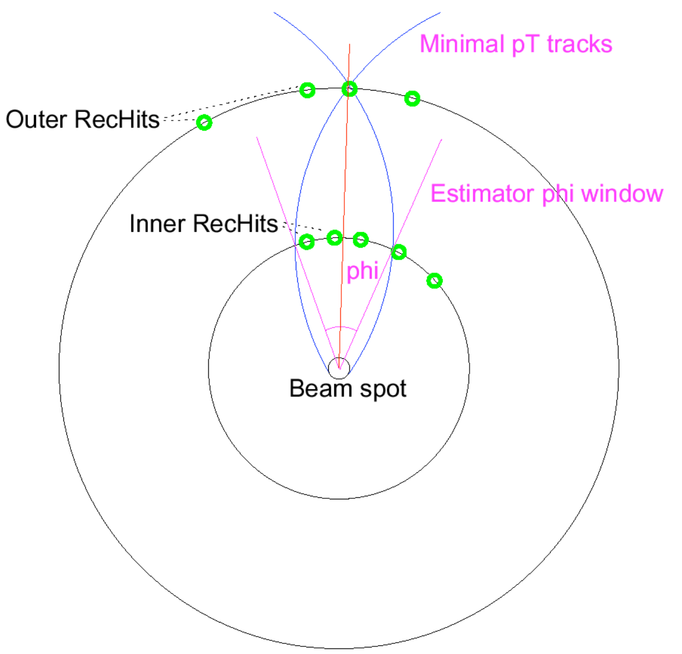
- **Needs to be fast and efficient**
- **Baseline seeding starts at the innermost layers**
  - Effects of higher track density are more than compensated by the high granularity of the pixel detector
- **Fast hit pair finding: starts with primary (outermost) hit**
  - can be restricted to a region of interest



Seed Generation

- **Complemented by second hit in other layers**
  - Fast "geometrical" search (p_T-cut, interaction region)
  - Multiple scattering important for low p_T -> fast parametrization
  - Number of seeds for tracking can be drastically reduced by using primary vertex estimate (z-constraint drops from 30cm to sub-mm)

- **Efficiency for hit pair finding ~ 100%**
- **Time small (O 10%) w.r.t. full track reconstruction**



Seed Generation

Alternatives:

- **“Outside-in” seeding**

- **“External” seeds provided by other sub-detectors:**

  Electromagnetic calorimeter

  - Based on pixel hits compatible with ECAL cluster and primary vertex

  Muon system

  - Tracker hits compatible with a fit of a standalone track in the muon system and the primary vertex

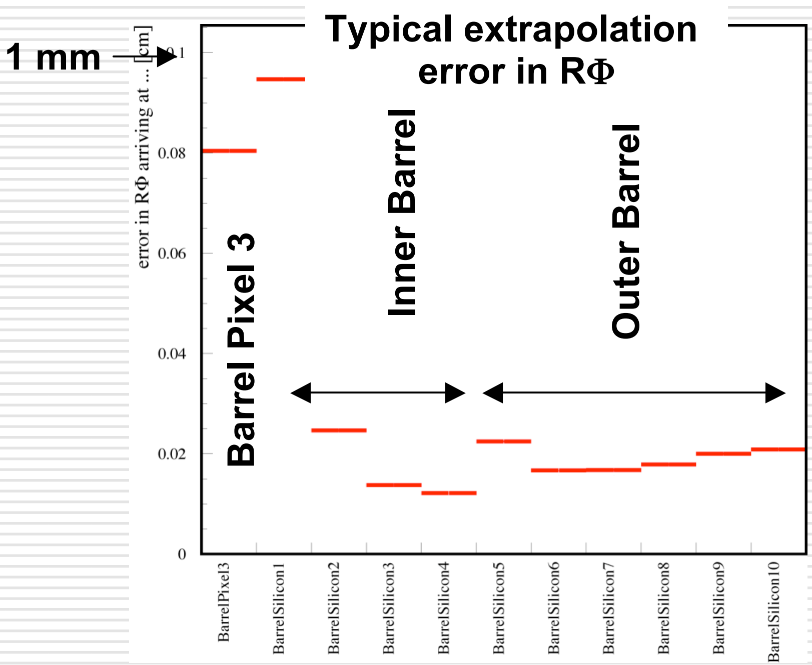
Trajectory Building

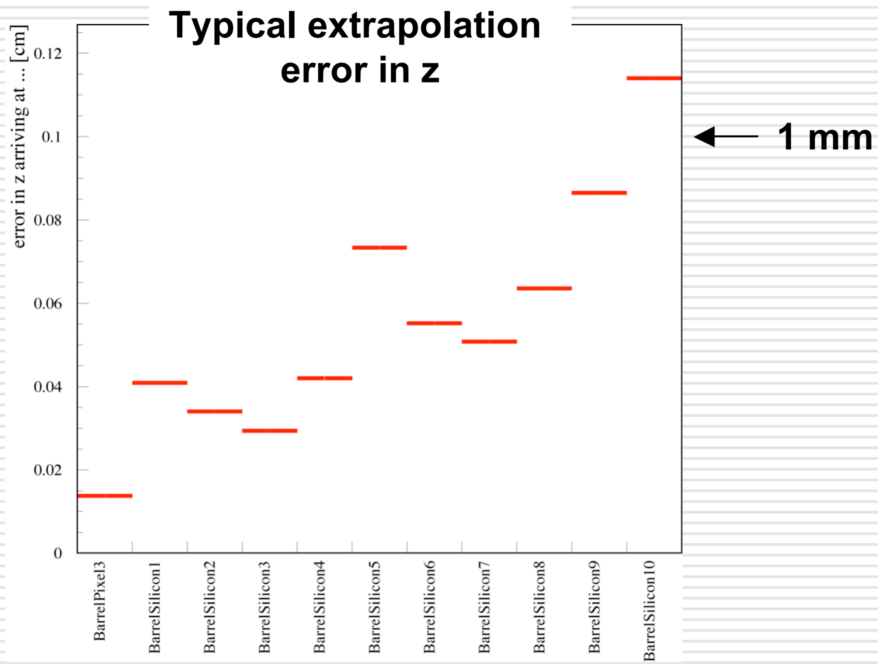
- **Based on a Kalman filter for simultaneous trajectory extension and hit selection.**
- **Selection is delegated to the classes describing the relevant components in the tracker structure**

- **Layer-to-layer navigation**
  - Layers are ~ hermetic → small number of candidate layers for next propagation step
  - Layers can be asked for lists of individual Si modules compatible with the current track parameters
  - The implementation is specific for each detector part (cylinders/disks, “turbine”/”staggered” types, …)
  - Several implementations of the same layer type can co-exist (e.g. using different search strategies or different grouping of modules)

Trajectory Building

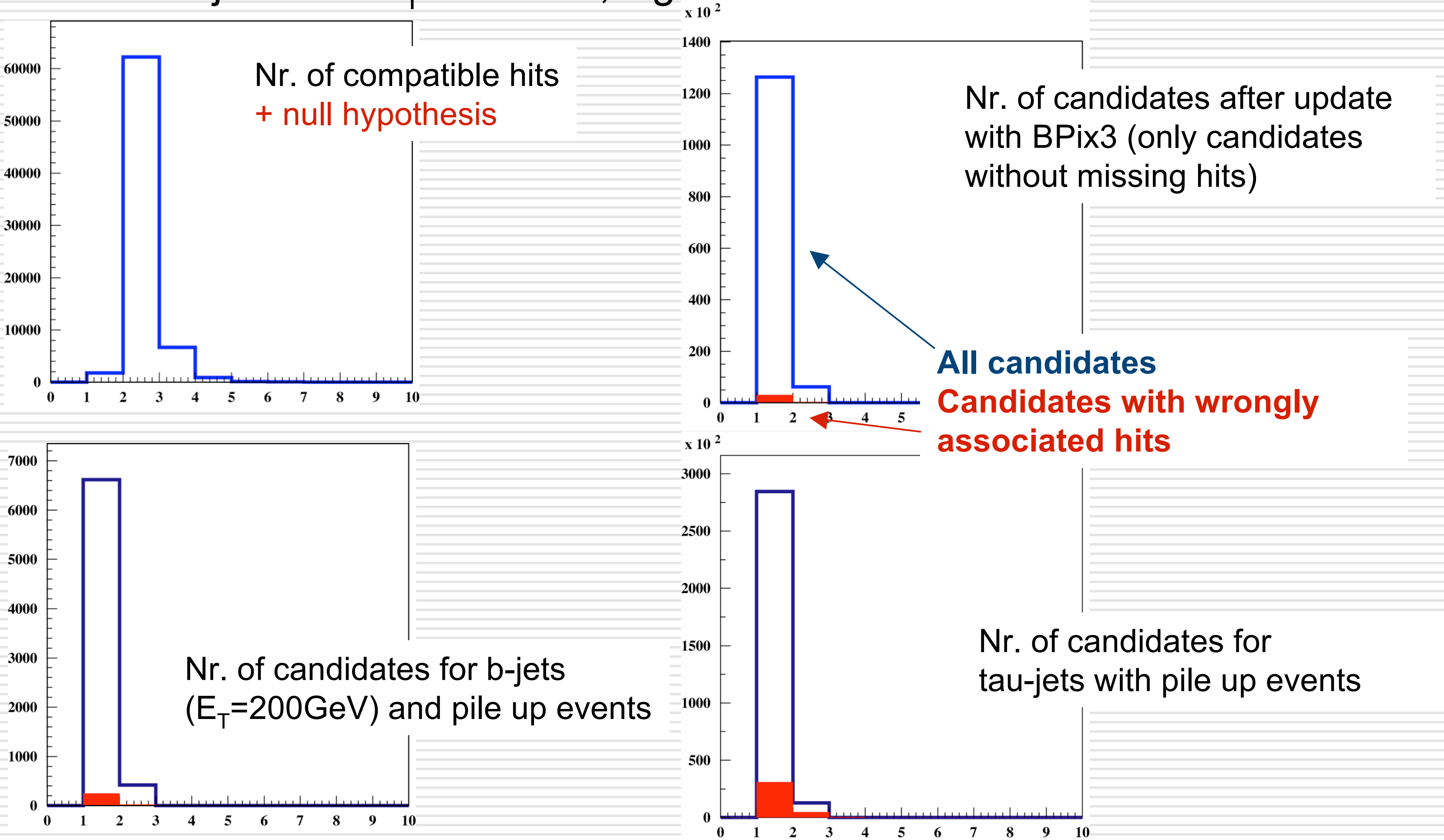
- **Update with new measurements**
  - Each Si module provides hits compatible with the current track parameters
  - Each hit leads to a new candidate trajectory
  - Possible inefficiency is taken into account by adding a "null" hypothesis: trajectory extension without adding a valid hit





Trajectory Building

- **Looking at one of the worst cases: updates at 3rd barrel pixel layer**
  - B-jets with E_T=100GeV, signal event only, counts / seed



Trajectory Building

- **Control of combinatorics: rejection & selection**
  - Propagation stops if number of layers without hit (total and consecutive) exceed a cut or no further compatible module is found
  - An intermediate resolution of ambiguities between candidates with an active / a missing hit can be performed
  - Active candidates are ranked based on the total χ^2 and the number of missing hits
  - Only the N best candidates are retained for further propagation

- **Final cleaning of ambiguities / seed**
  - Search for sets of mutually exclusive candidates (based on fraction of shared hits)
  - The best candidate in each set is retained

Track Fitting & Smoothing

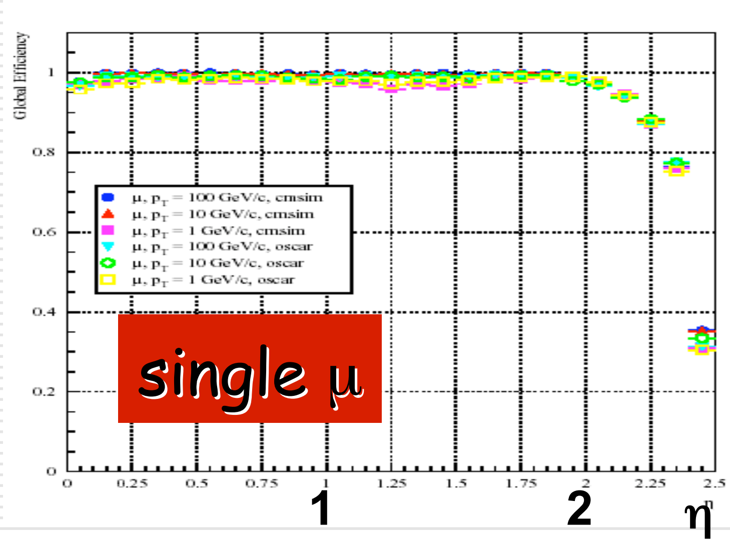
- **Candidate trajectories are refitted and smoothed using a KF**
  - Uses the gain matrix formulation of the KF
- **“Forward” = inside-out fit**
  - Removes possible biases from the vertex constraint used in seeding
  - Uses better estimation of track angle when retrieving position information from the seeding hits
  - Results in optimal estimate of track parameters at the end of the trajectory
- **“Smoothing” = outside-in fit**
  - Smoothed states = weighted mean of forward fit (including measurement) + backward fit (prediction)

Track Reconstruction

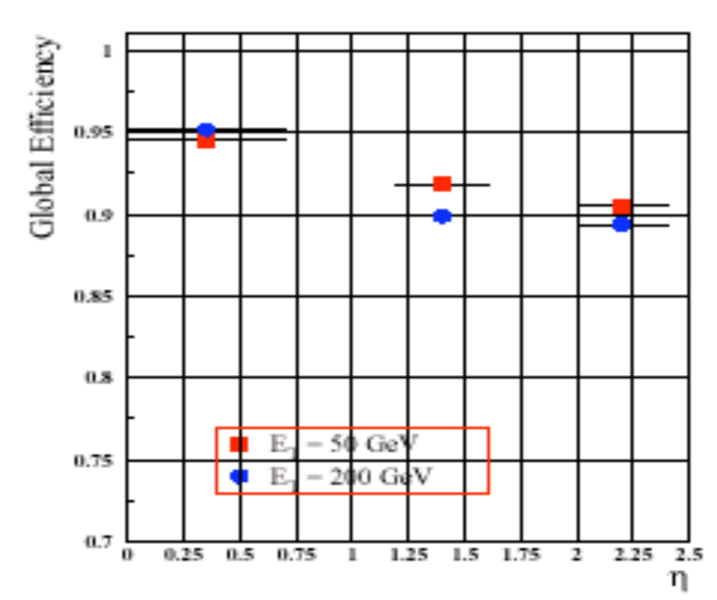
Efficiency

Some pion tracks are too short for reconstruction due to hadronic interactions

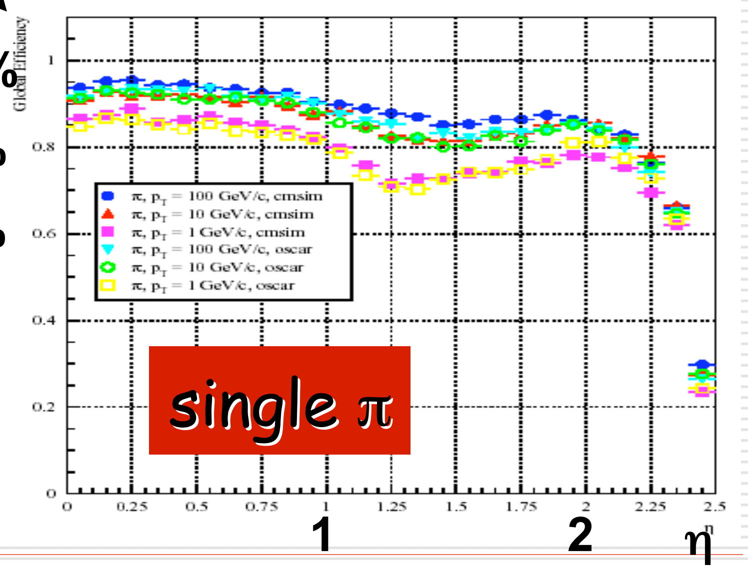
100%
80%
60%



No significant loss inside jet cones

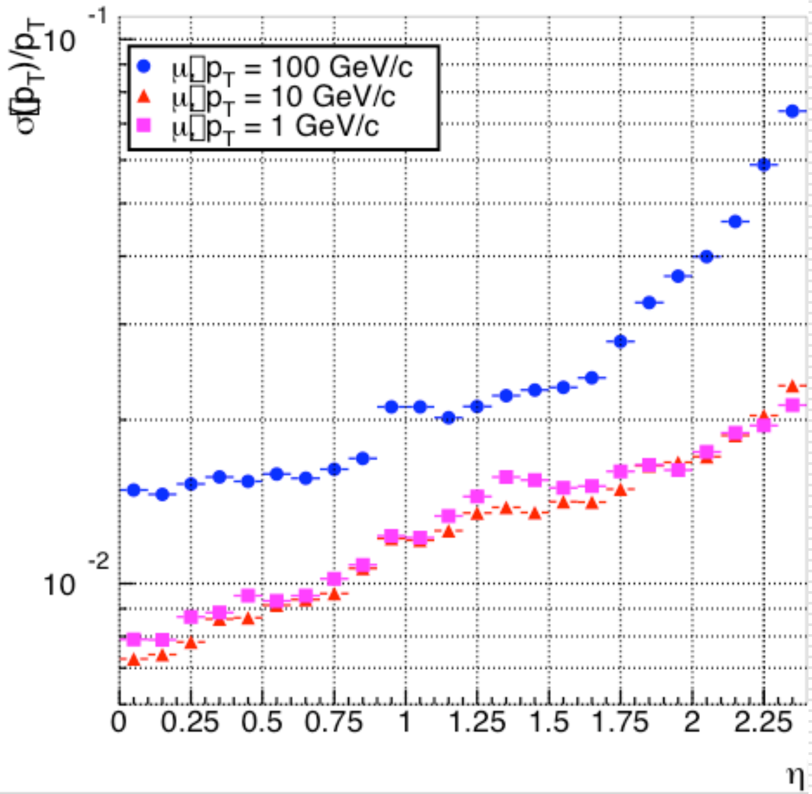


100%
80%
60%

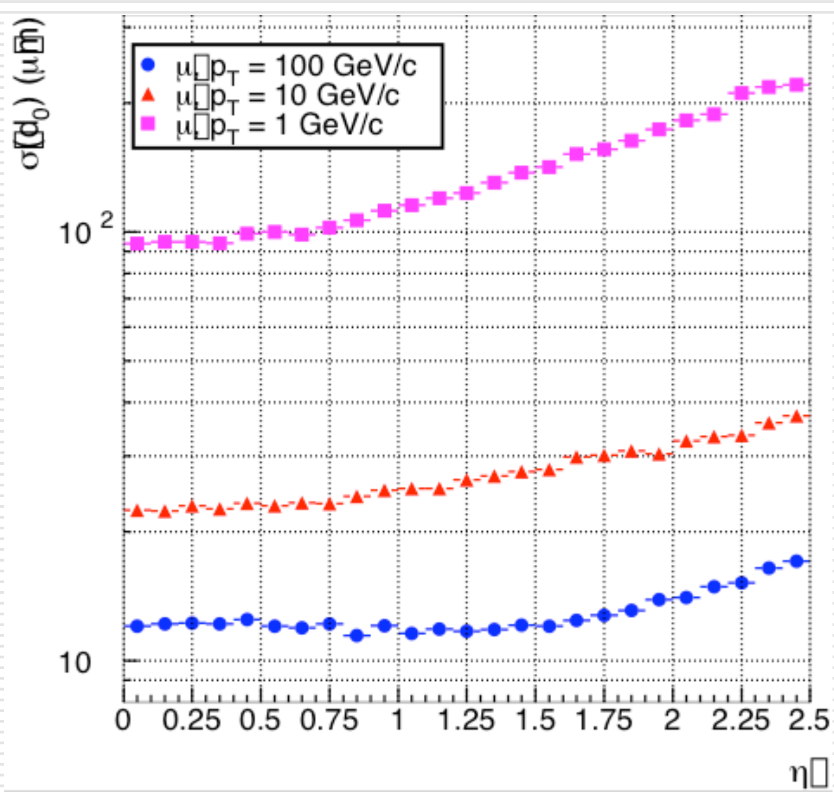


Track Reconstruction

Relative resolution, p_T

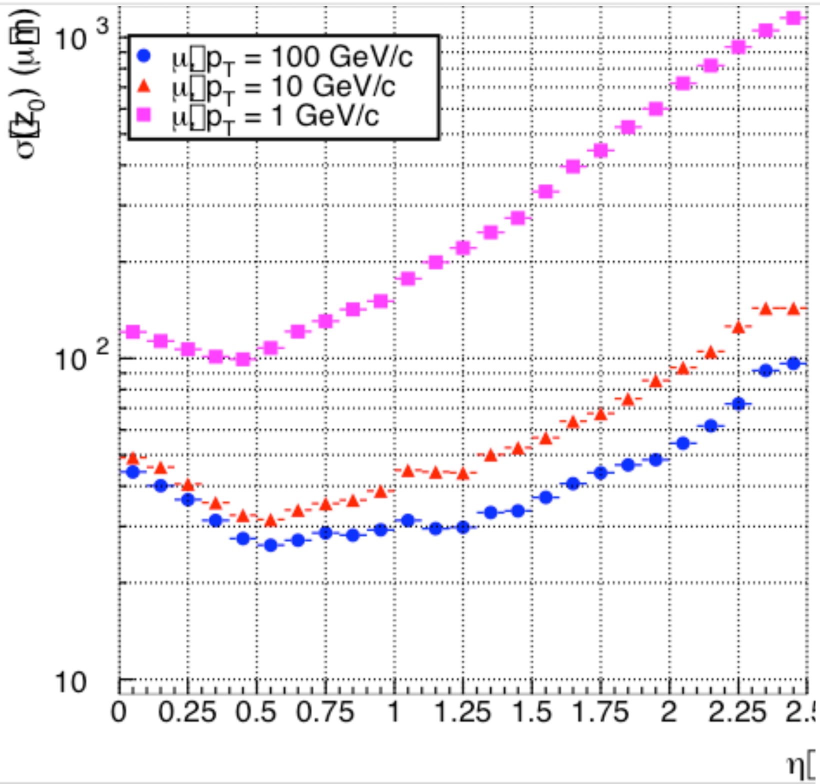


Transverse impact parameter resolution

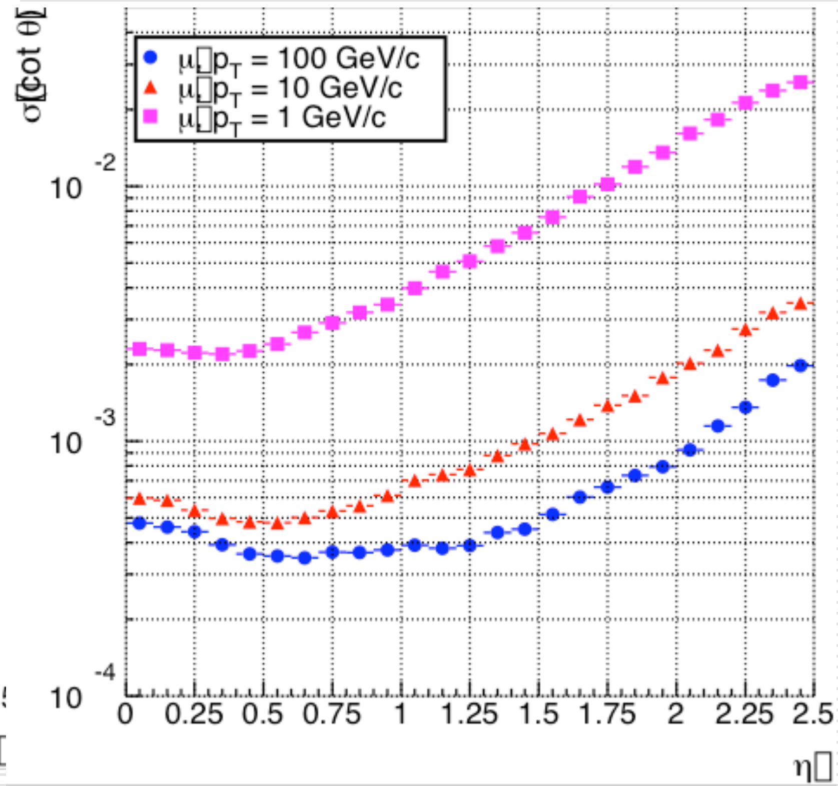


Track Reconstruction

Longitudinal impact parameter resolution



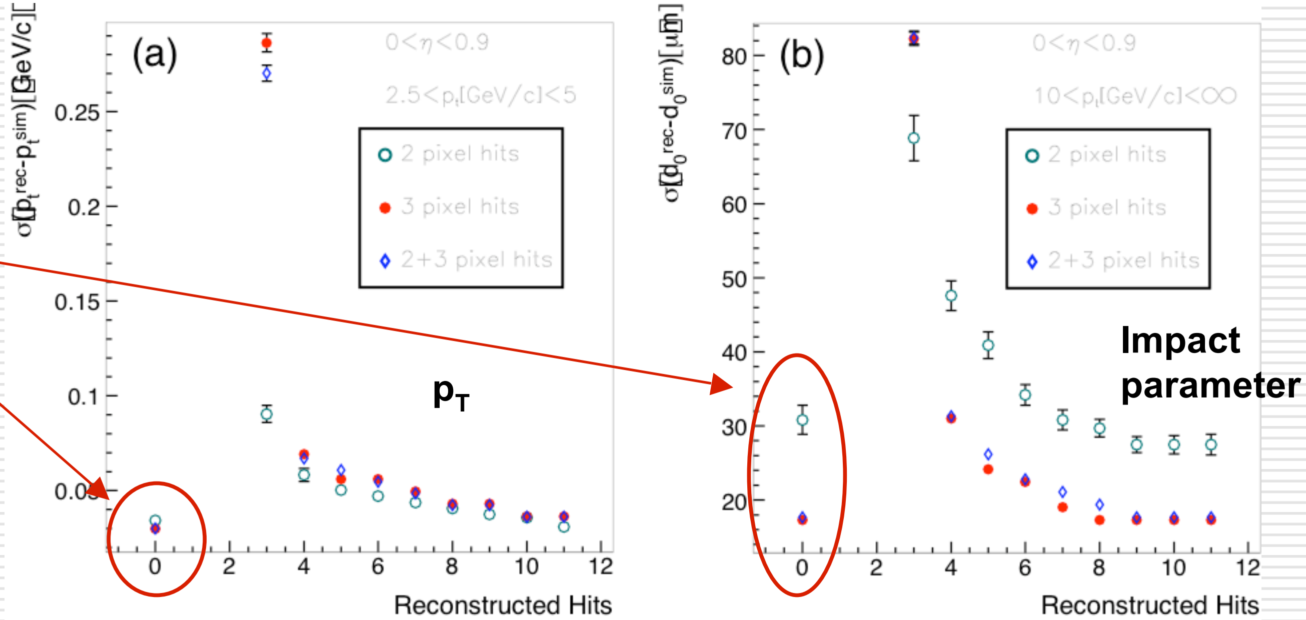
Resolution on cot(θ)



Track Reconstruction - Special Modes

- **Regional tracking**
  - Region of interest derived from other detectors / low-level trigger objects: e.g. limits in phi, eta, p_T
  - Seeding and trajectory building is restricted to this region
- **Partial reconstruction**
  - Don't need full tracker resolution e.g. for isolation criteria
  - Stop trajectory building when errors are small enough

full tracking



Resolutions

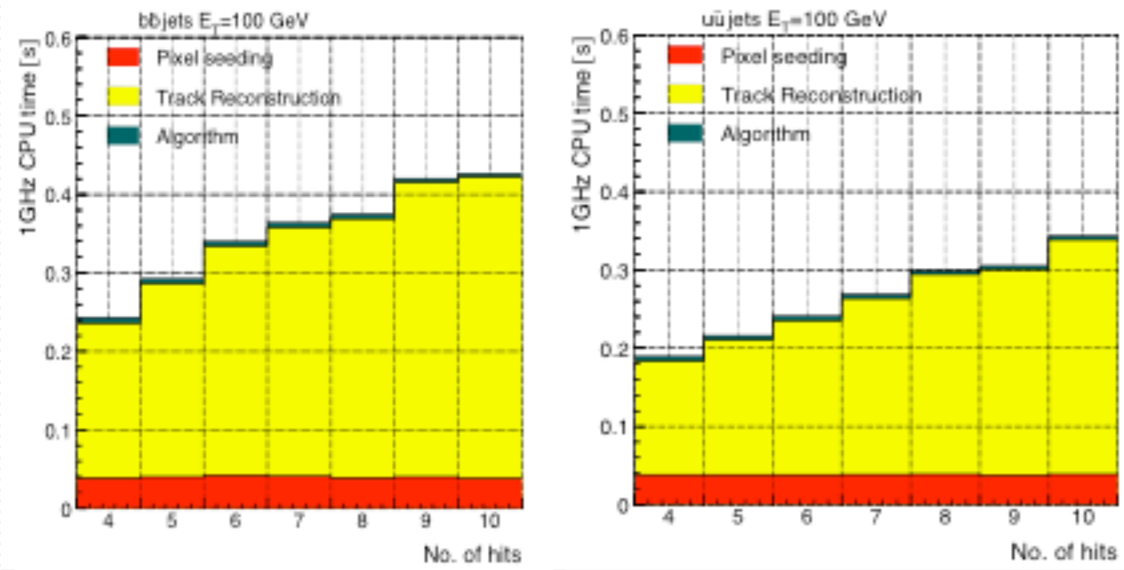
Track Reconstruction

- Timing for b-tagging in high level trigger
- jets with E_T=100GeV, low luminosity pile up

B-tag

Track finding & fitting

Seeding



- CPU time can be reduced (with some loss of efficiency) by using track segments in the pixels

Track Reconstruction

Summary default track reconstruction

- **Advantages**
  - Based on simple, well-known algorithms
  - Efficient & robust
  - Few “free parameters”
  - Works with minor retuning even for heavy ion collision
- **Drawbacks**
  - The limit on the number of candidates in combinatorial trajectory building is a compromise between speed and the risk to loose the right track
  - No differentiation between noise and hits from other tracks
  - Hard assignment is known to be sub-optimal in dense environments

The modular structure of the reconstruction software and the polymorphism of basic elements like hits and track states allow for easy implementation, testing and benchmarking of new algorithms

Deterministic Annealing Filter

Basic idea (equivalent to elastic arms):

- **allow for several hits / layer to compete in a track**
- **In each layer, use assignment probabilities p_i for each hit:**

$$f(r;0,V) \Rightarrow \sum_1^{n_H} p_i f(r_i;0,V_i); p_i \in [0,1]$$

r = residual w.r.t. prediction

- **Run fitter & smoother, update p_i based on distance to combined forward & backward prediction**

$$p_i = \frac{\varphi(r_i;0,V)}{c + \sum_j \varphi(r_j;0,V)}$$

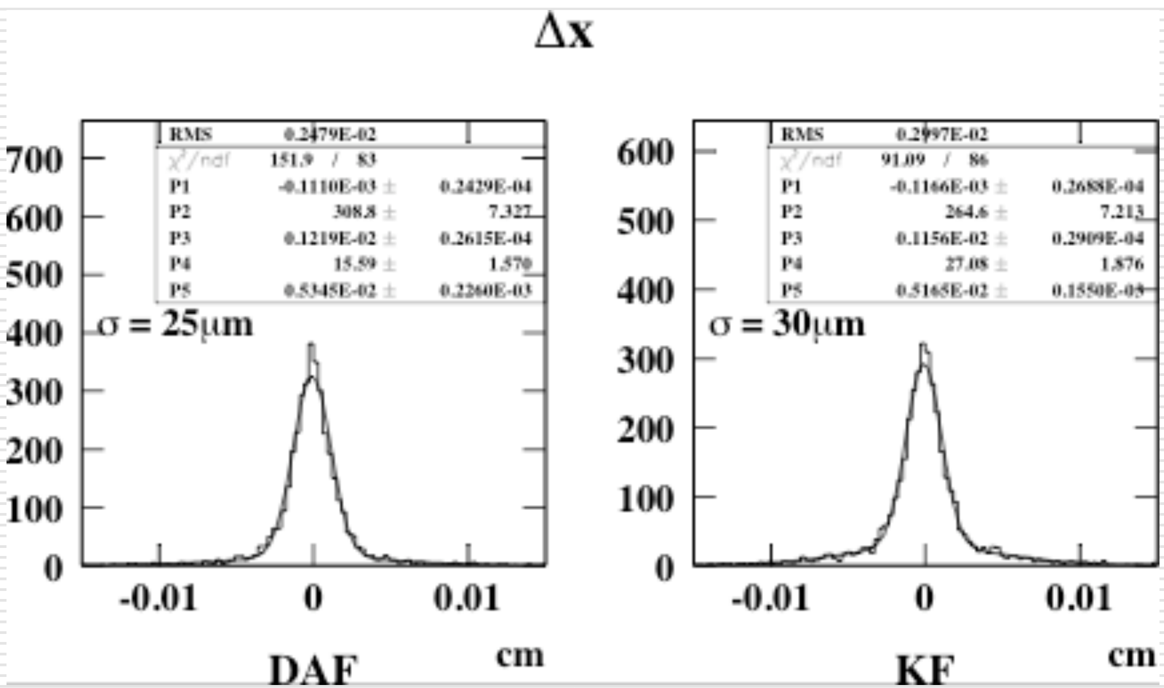
c serves as cutoff - stabilizes fit in case all hits are incompatible

- **Fitter & smoother are iterated until convergence**
- **In order to avoid local minima use annealing: V ⇒ α V**
- **Decrease α (~ temperature) to 1. Hard assignment (α→0) yields worse resolutions!**

Deterministic Annealing Filter

- Event sample: b-jets in the central part, E_T=200GeV
- Candidate hits were collected using a road search (standard trajectory building is using a KF!)

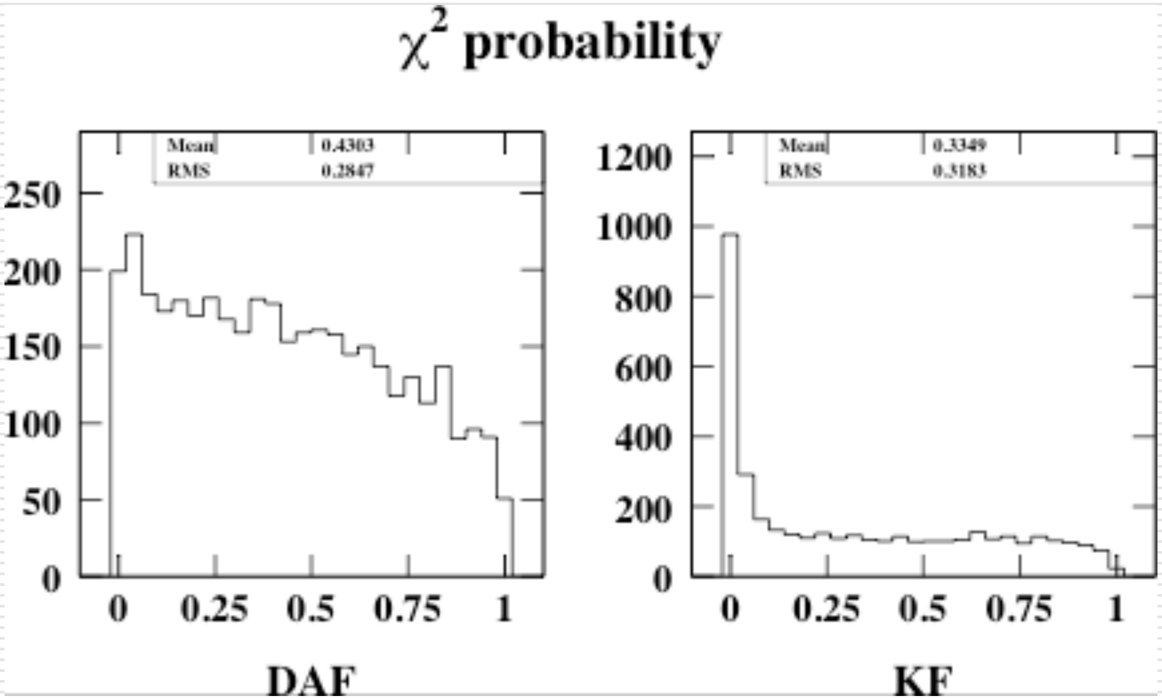
- Transverse impact parameter: tails are reduced



Deterministic Annealing Filter

- **Track quality criteria:**
  - End fit with soft assignment (a=1) →
    use effective number of hits = sum of weights

- **χ^2 probability: almost flat for DAF**



Multi-Track Fitter

Extend to competition between hits AND tracks

- **matrix of assignment probabilities p_{ij}, $i=1\ldots N_{hit}$, $j=1\ldots N_{track}$**
- **3 options:**
  - For each j, normalise Σi → DAF
  - For each i, normalise Σj → equivalent to an elastic arms algorithm with competing tracks
  - Normalise Σij → full competition, MTF
- **As for the DAF a cutoff used**
- **As for the DAF the fit is iterated and the temperature is lowered according to an annealing schedule**

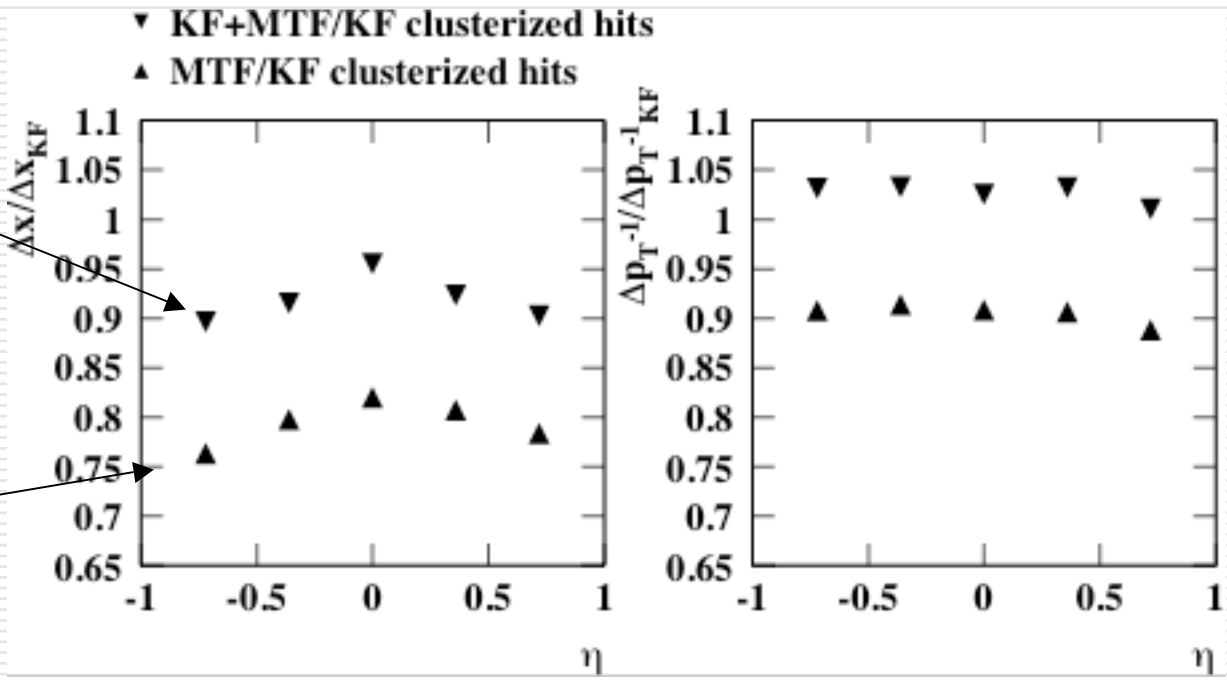
- **The MTF depends crucially on the quality of the track candidates in the first iteration:**
  - Different scenarii have been tried: initialisation with KF or DAF

Multi-Track Fitter

- **Results from an simplified simulation:**
  - 2 collimated muon tracks, helix model

Initialized with KF

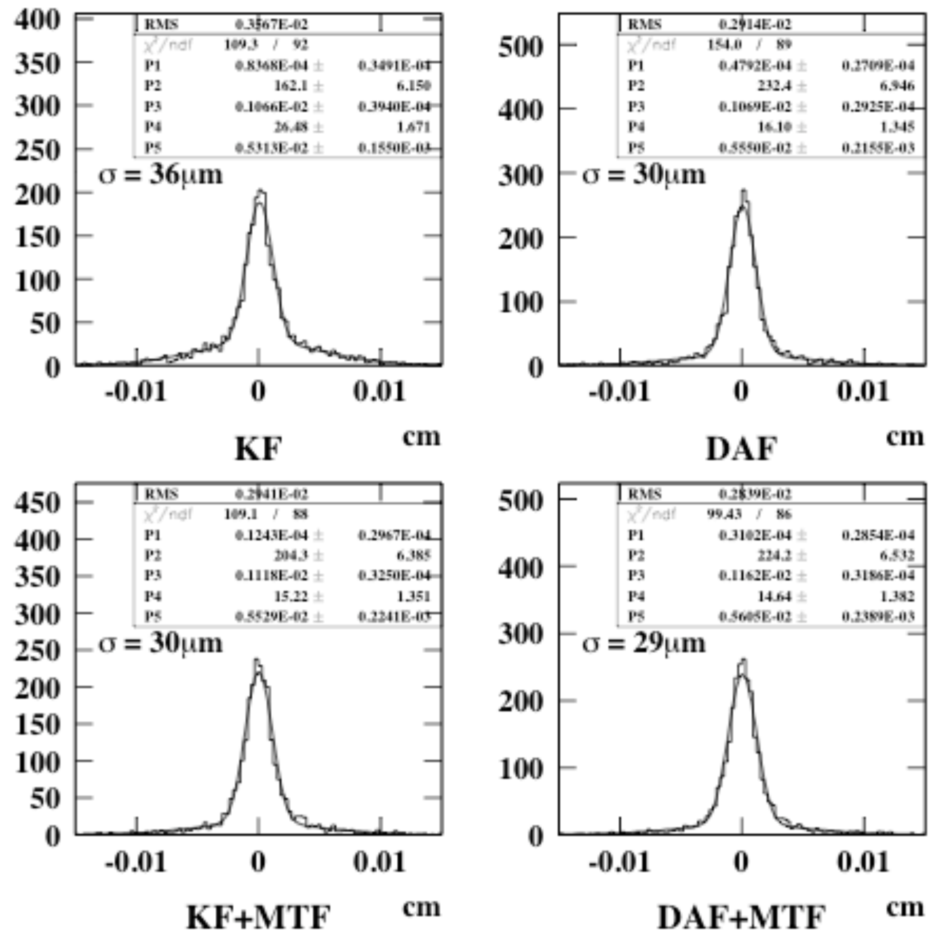
Initialized on true values



- **Sensitivity to initialization**

Multi-Track Fitter

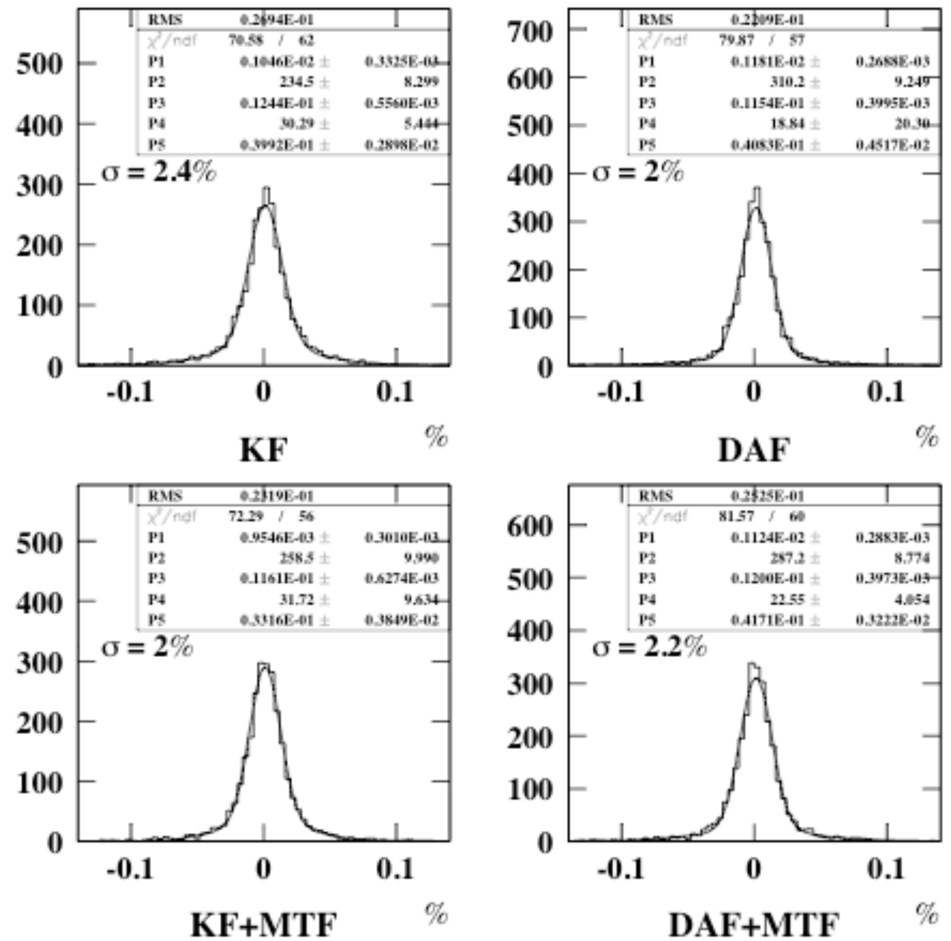
- **Transverse impact parameter**



- **Full simulation: highly boosted 3-prong tau decays**

Multi-Track Fitter

- Inverse transverse momentum



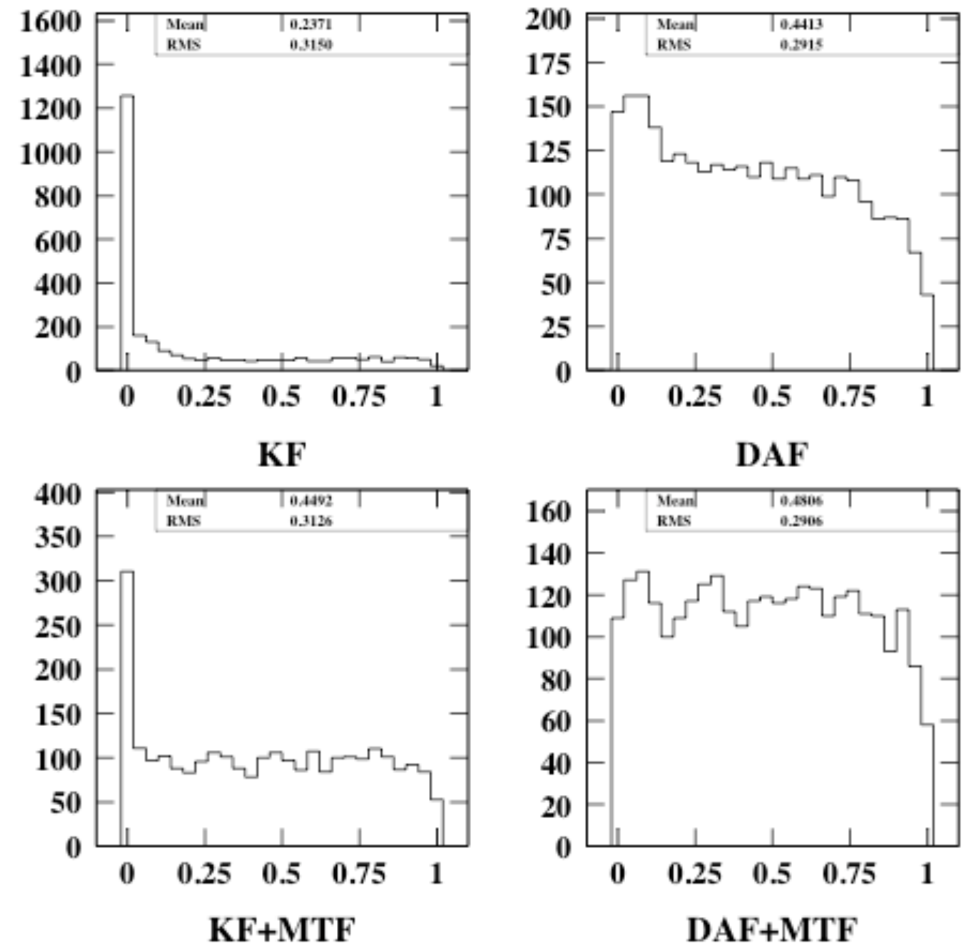
Multi-Track Fitter

- χ^2 probability

Conclusion:

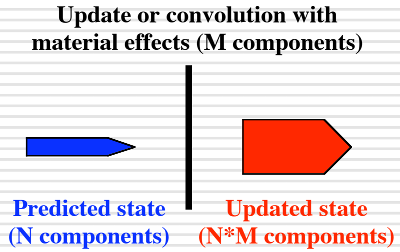
- I.P. improved
- No gain for p
- Better error estimates

- Environment still too “easy” for MTF?



Gaussian Sum Filter

- **Linear least-square estimators (== KF) are only optimal in linear systems with Gaussian measurement errors & process noise**
- **Basic idea for non-Gaussian case:**
  - Keep advantages of KF (fast, modular, no need for numeric minimization) by describing general pdf's by mixtures of Gaussian components $f(x) = \sum w_i \varphi(x; \mu_i, V_i); \sum w_i = 1$
  - Propagation & updates resemble N Kalman filters run in parallel
  - Inclusion of new multi-Gaussian components → multiplication of the total number of states



  - After each update the weights are recalculated to reflect the compatibility with the measurement → non-linear, "adaptive" filter

Gaussian Sum Filter

For use in real applications need

- **Prior knowledge about the noise distributions (difference to DAF, EA etc.!) and description in terms of a Gaussian mixture**
  - Fit parameters (w_i, μ_i, σ_i) by minimizing distance between the known distribution and the Gaussian mixture

- **Control the combinatorial explosion in the number of components: eliminate or combine components to keep total number low**
  - Simplest approach: keep N components with the highest weight
    - Looses too much information in the tails of the distribution
  - Better: define distance between components and recluster
    - E.g. combine two closest components and reiterate

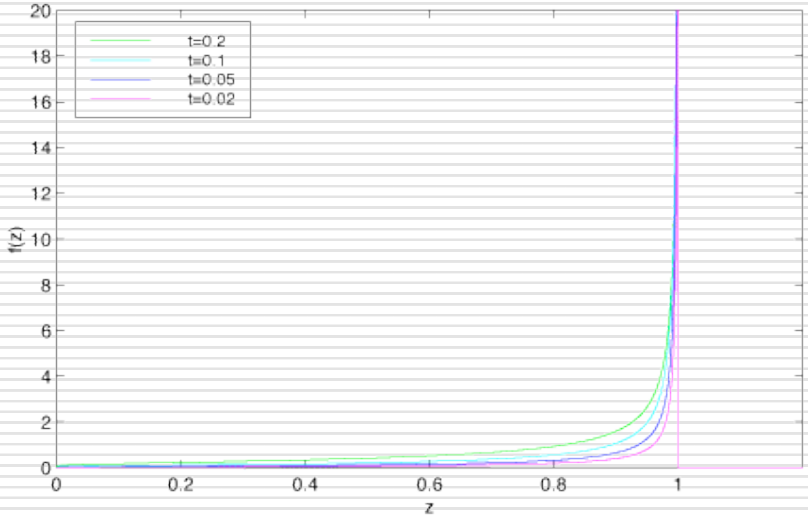
Gaussian Sum Filter

Implemented in CMS for electron reconstruction in the tracker

- **Radiative energy loss of electrons is highly non-Gaussian**

Bethe-Heitler model
t thickness in units of X_0
z remaining fraction of energy

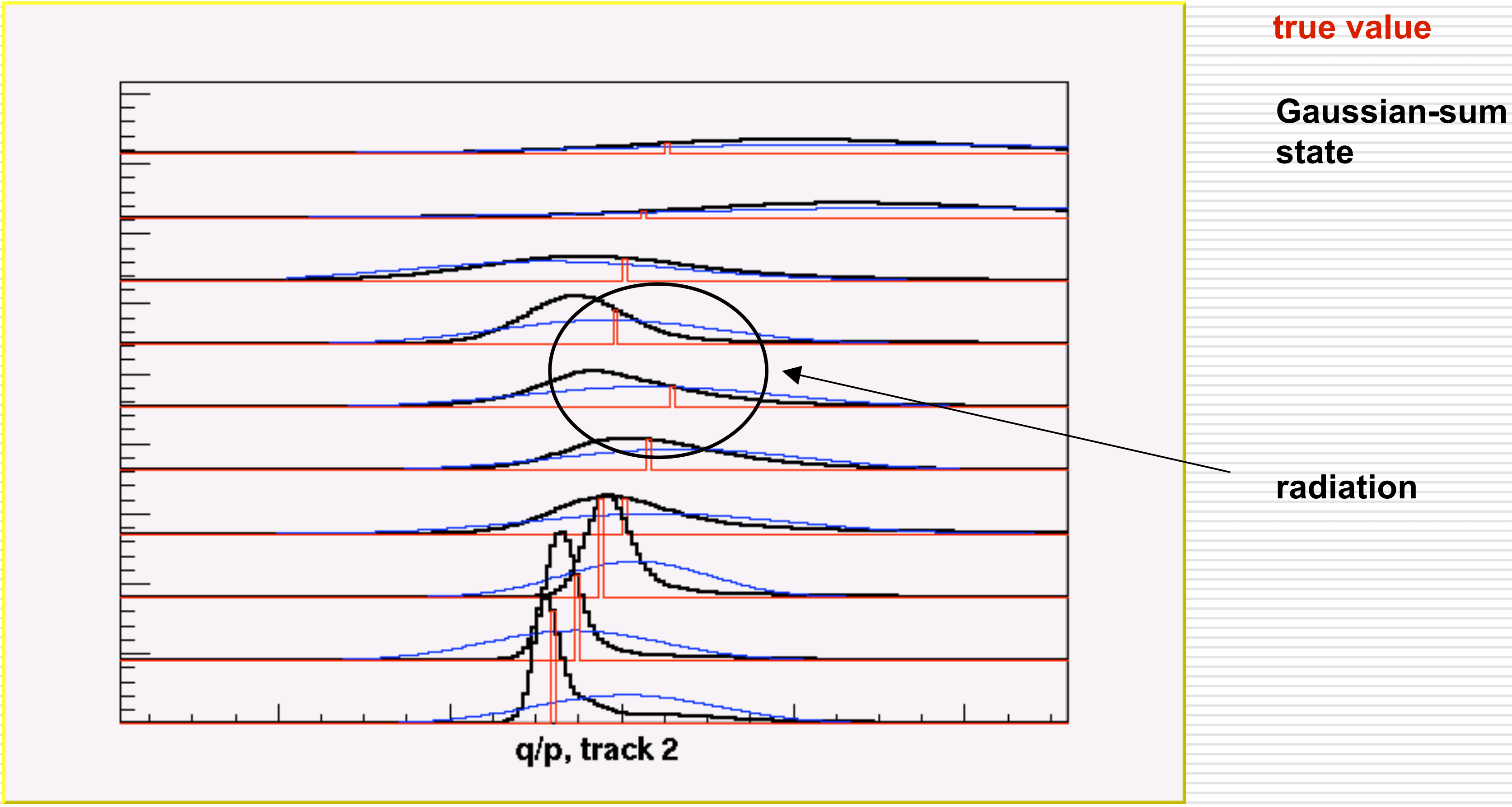
$$f(z) = \frac{(-\ln z)^{c-1}}{\Gamma(c)}, c = t/\ln 2$$



- **Implementation:**
  - Multi-Gaussian states "hidden" in object describing the track parameters on a surface: standard application see equivalent single-Gaussian state
  - Bethe-Heitler model available by switching propagator class

Gaussian Sum Filter

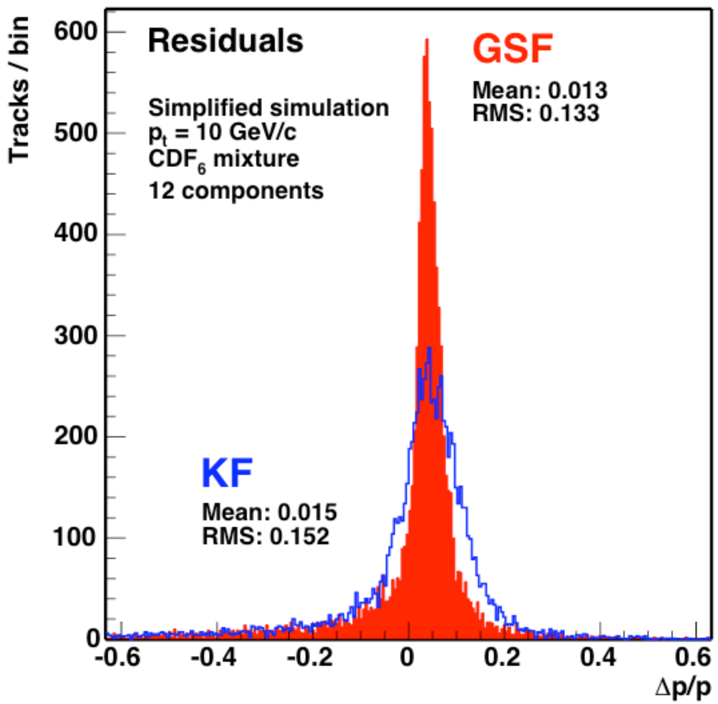
Evolution of inverse momentum for a single track

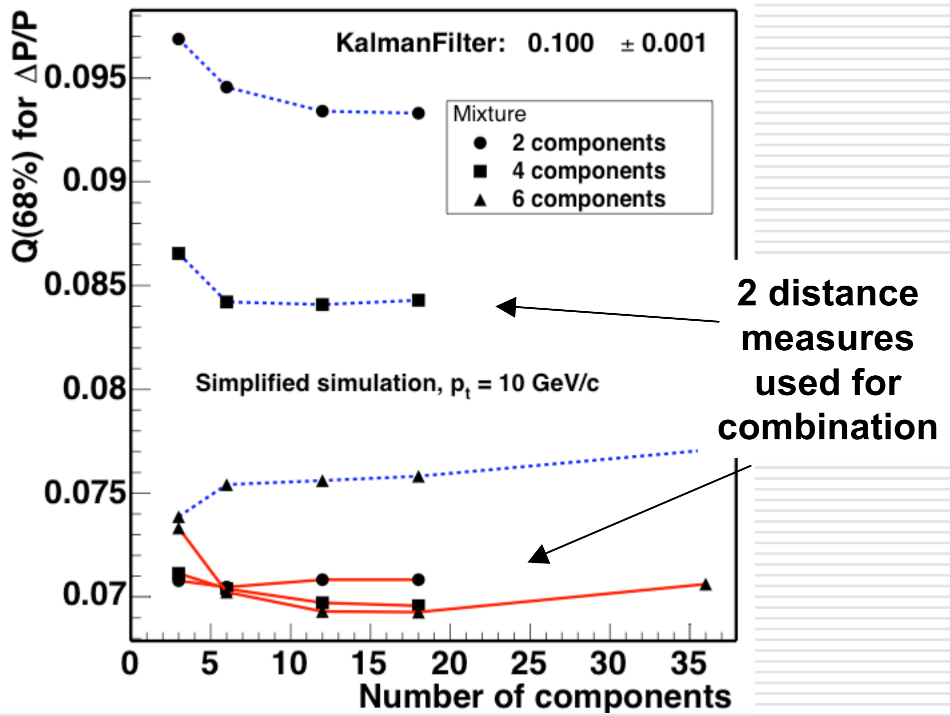


Gaussian Sum Filter

"Ideal" simulation:

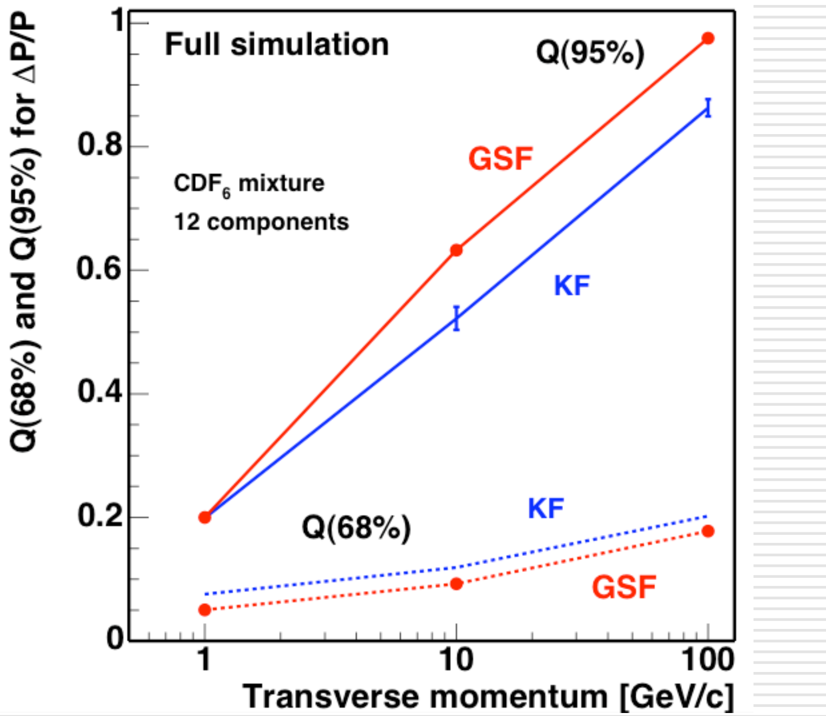
- Helix, Gaussian measurement errors, Bethe-Heitler model
- 6 components in radiation model
- Nr. of components during fitting limited to 12

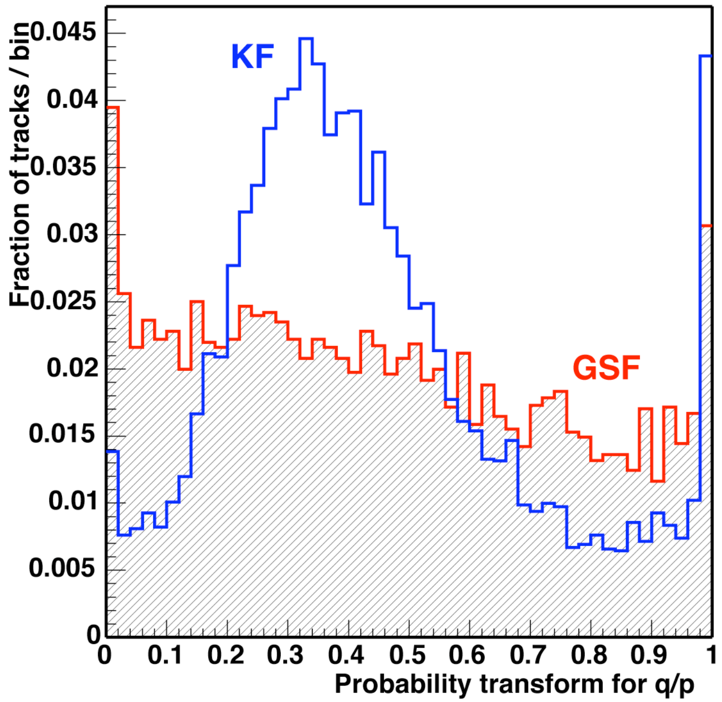




Gaussian Sum Filter

Detailed simulation (detector and interaction with matter)



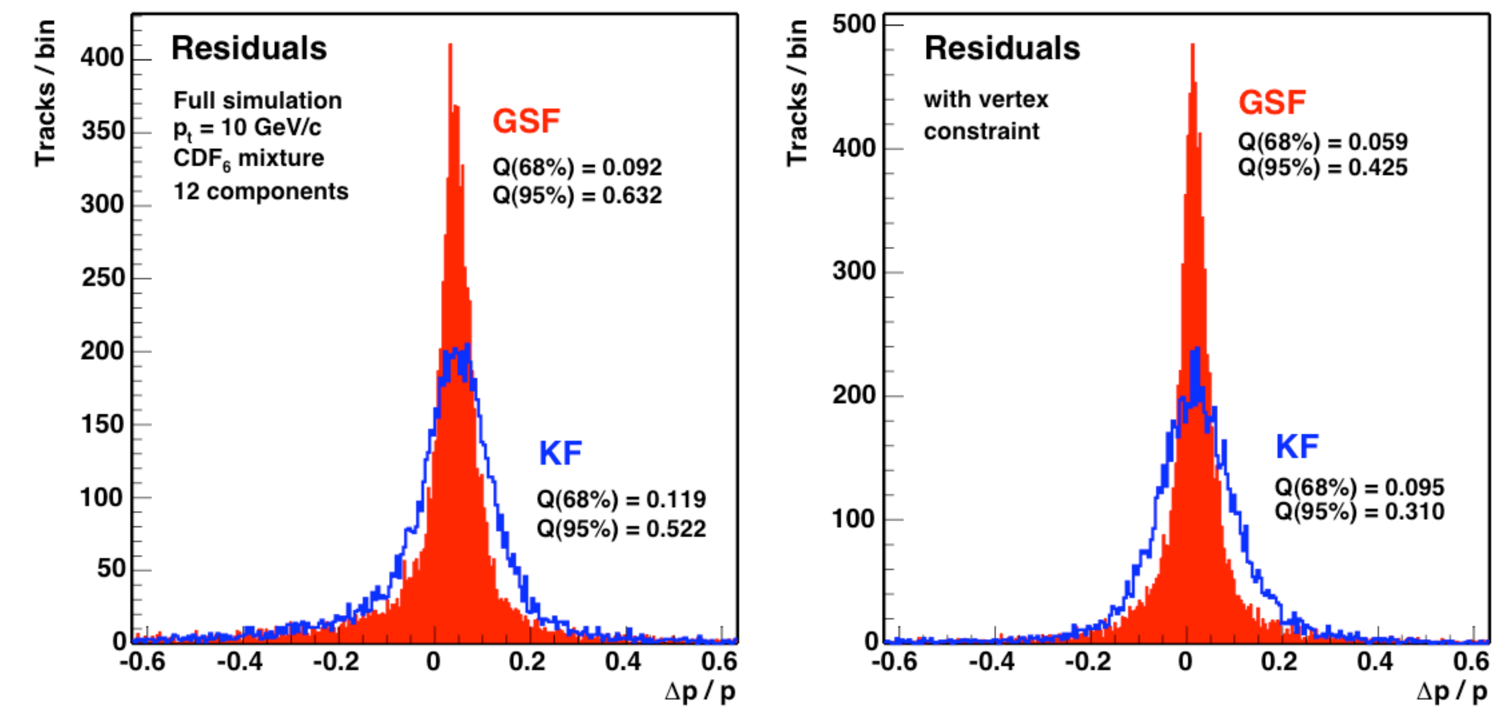


Tails in GSF residuals were traced to tails in the pixel measurements

Probability transform of the residual: should be flat, if distribution of the estimate is correctly described

Gaussian Sum Filter

- Track fit has no handle on radiation in innermost layers ⇒ use interaction region / primary vertex constraint

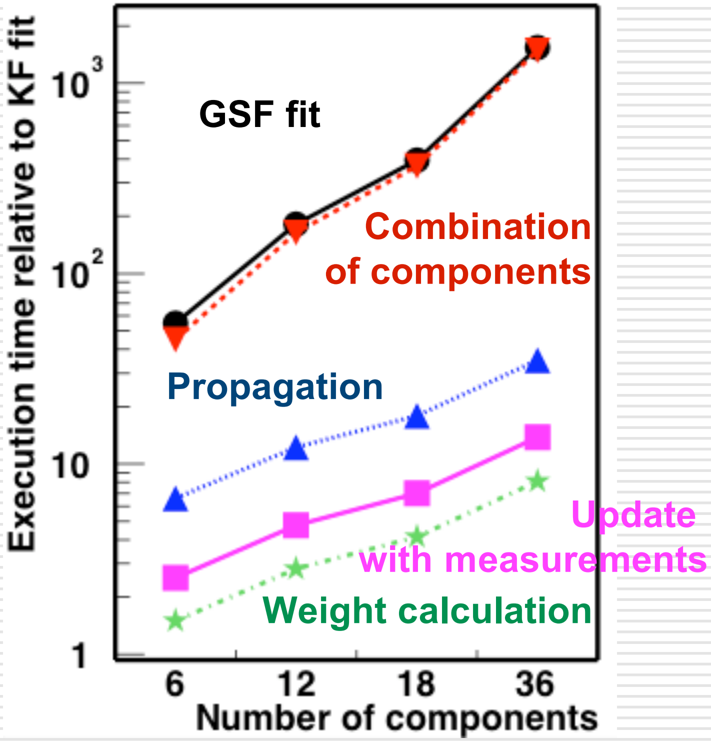


Distributions become more Gaussian
GSF can extract some more information

Gaussian Sum Filter

For electron reconstruction

- Provides good estimate of pdf of track parameters
- Can yield better resolutions than KF
- Needs good modelling of process noise (for propagation and measurements)
- CPU time dominated by control of combinatorics
  - Use on preselected tracks
  - Better strategies for combination of components?



- Other possible appliations:
  - Non-gaussian measurement errors (tails in residuals)
  - Modelling of inhomogeneous distribution of material

Vertex Reconstruction

Stages in vertex reconstruction

- **Vertex finding**
  - Association of tracks to a vertex
- **Vertex fitting**
  - Estimation of vertex position (and track parameters at vertex)
- **Kinematic fitting**
  - Refinement of estimates using kinematic information (masses)

} Kalman filter

Areas of application

- **Primary vertex reconstruction**
  - (very) high multiplicity
  - Needed as reference for many other reconstruction steps
- **Secondary vertex reconstruction**
  - Low multiplicity
  - Track association and vertex separation problem
  - Needed for reconstruction of long-lived particles

Vertex Reconstruction

Track parametrization for vertex fitting

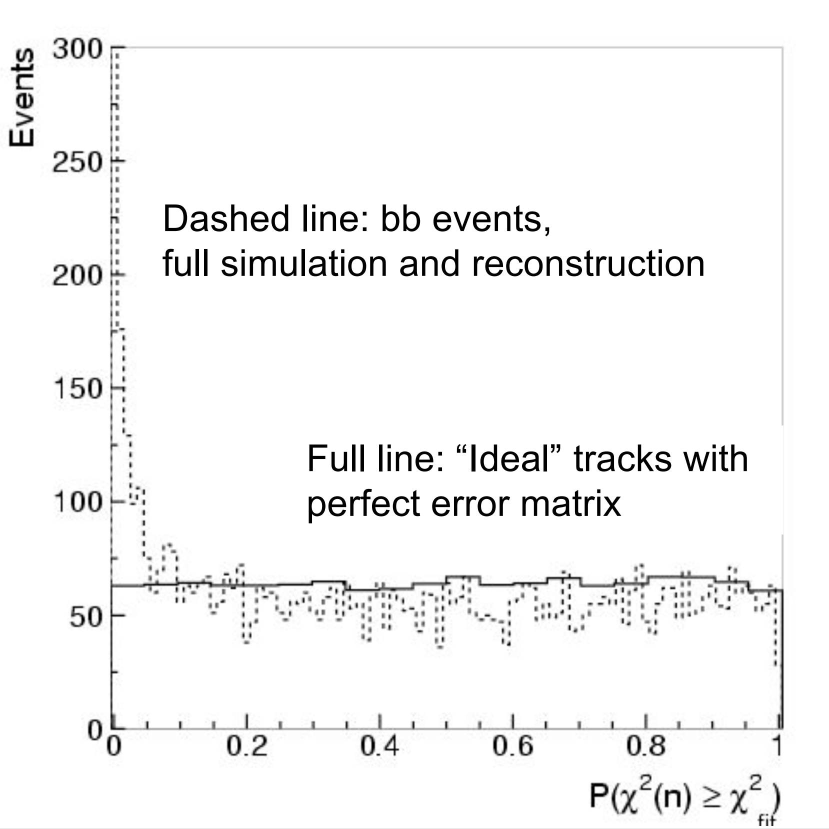
- Choose suitable parameters for linearization
- Defined close to vertex candidate
- With fast, analytical conversion from standard track parameters

- Perigee parameters
  - Signed curvature ρ
  - Signed transverse impact parameter d0
  - Polar angle at perigee θ
  - Azimuthal angle at perigee ϕ
  - Longitudinal coordinate at perigee z

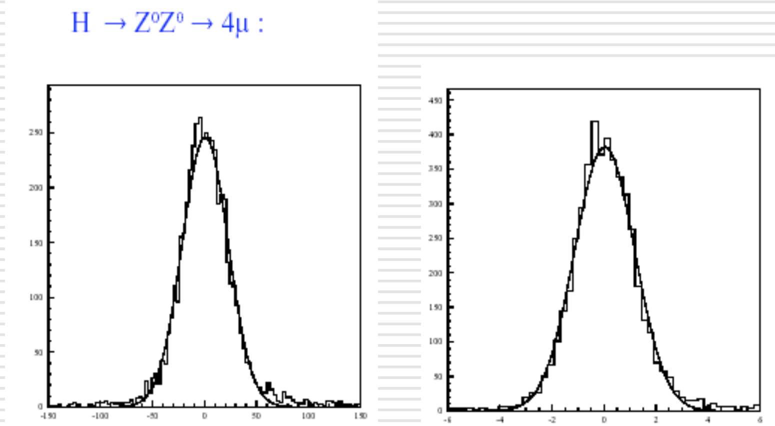
Least Squares Vertex Fitting

Kalman filter applied to primary vertex fit

Vertex fit χ^2 probability



Z-position for H→ZZ→4μ



Resolution
(22.2 ± 0.3) μm

Pull
1.16 ± 0.02

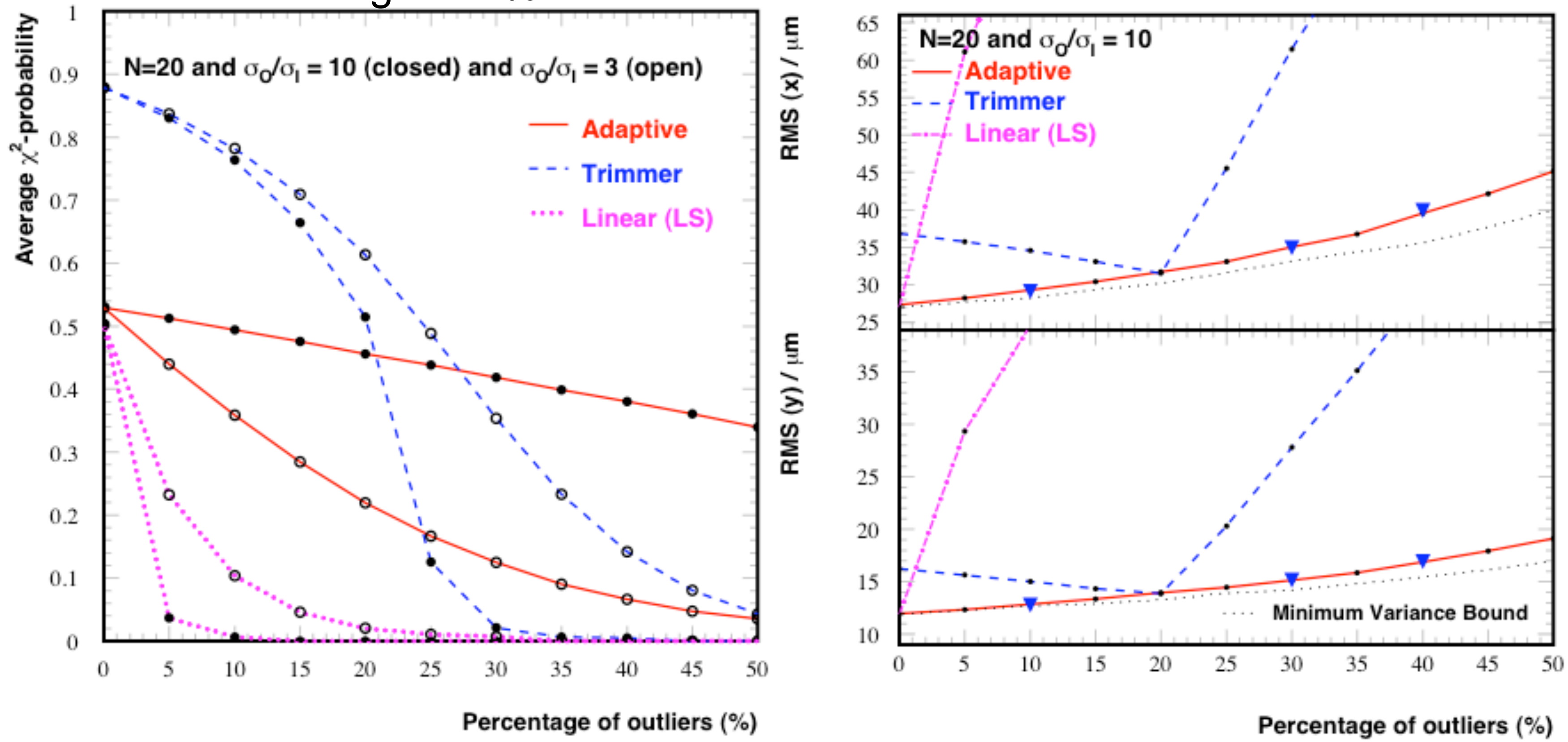
Robust Vertex Fitting

Two basic groups of robust reconstructors have been studied

- **Trimming algorithms remove the least compatible tracks**
  - E.g. a constant fraction of the tracks
  - Or according to a constant probability cut
    - The default CMS primary vertex reconstructor removes tracks with <5% probability
- **Adaptive vertex fitter**
  - Same idea as for deterministic annealing filter:
  - Assign weight according to compatibility
  - Introduce cutoff
- **In all cases an iterative fit is needed. For the adaptive fitter an annealing scheme is used.**

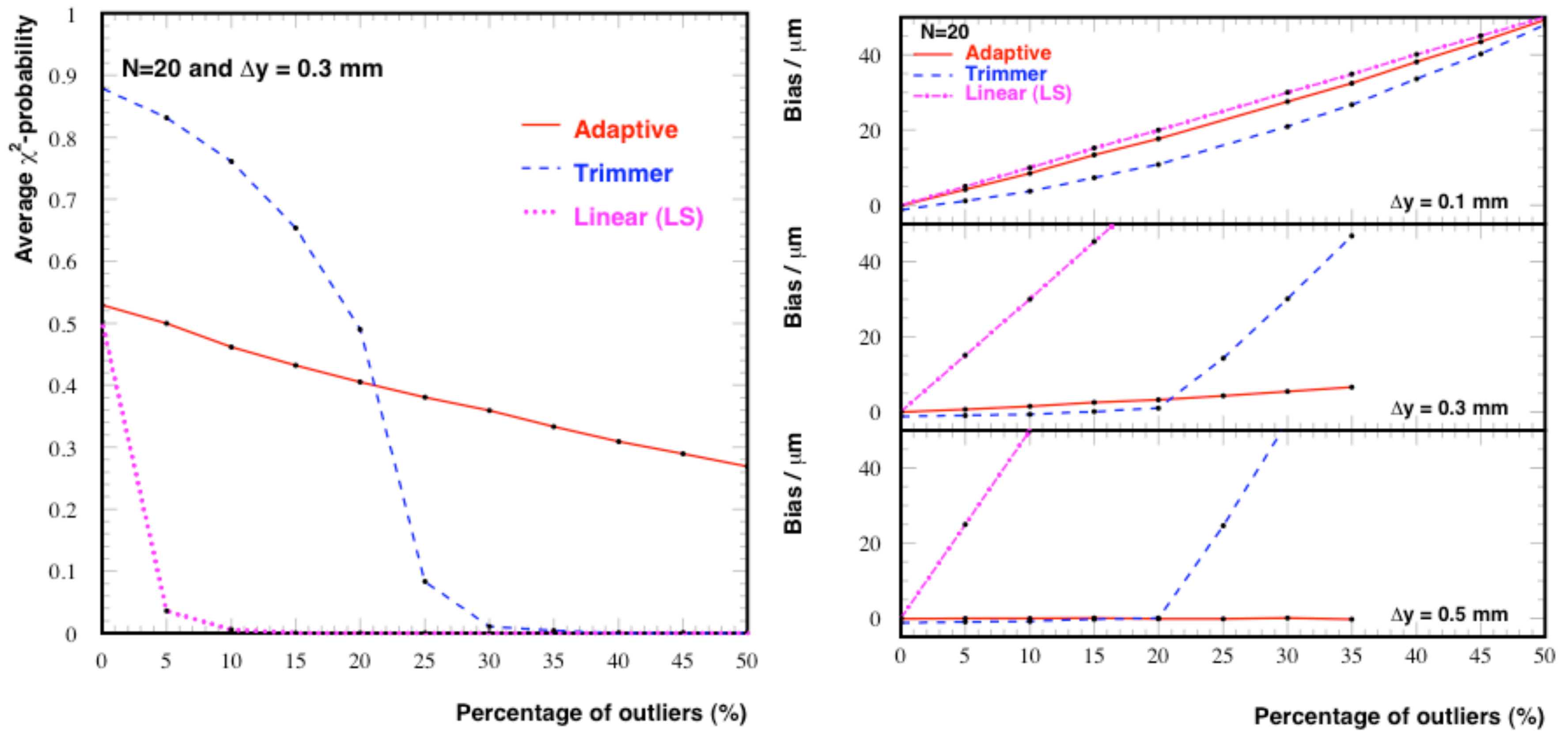
Robust Vertex Fitting

- **Simplified simulation of tracks at a vertex**
  - High multiplicity = “primary vertex” scenario
  - Type 1 outliers = tracks with underestimated errors
  - Trimming at 20% of tracks



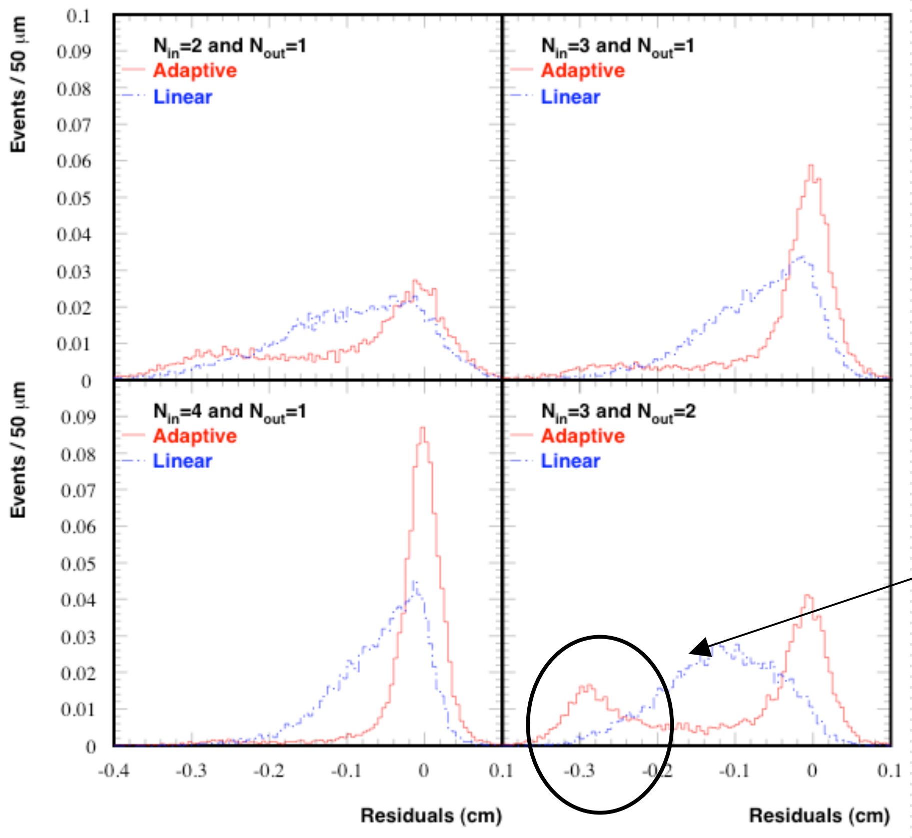
Robust Vertex Fitting

- As before, but with
  - Type 2 outliers = tracks from a second vertex



Robust Vertex Fitting

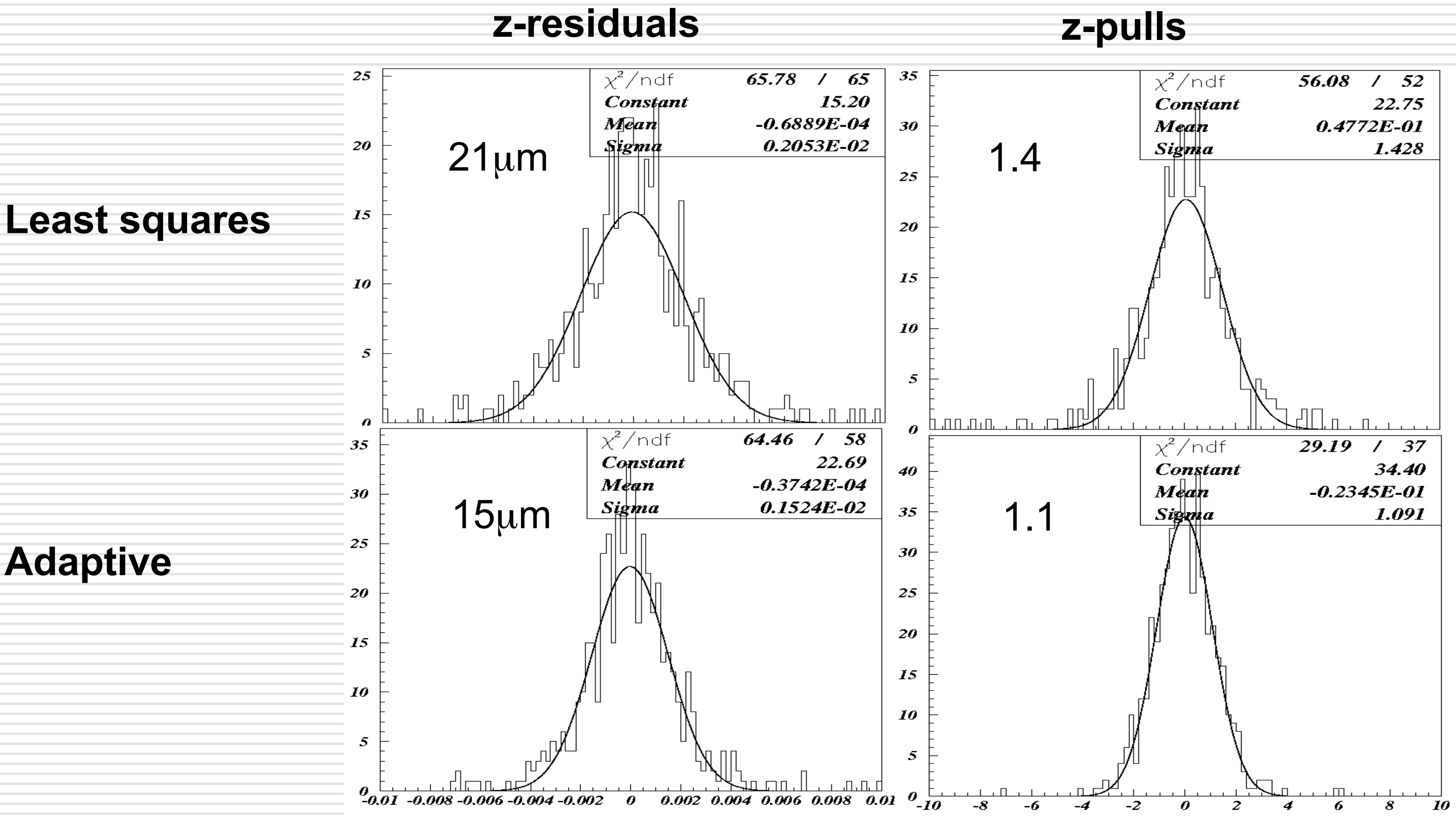
- **As before, but**
  - Low multiplicity = "secondary vertex" scenario



Second peak corresponds to primary vertex

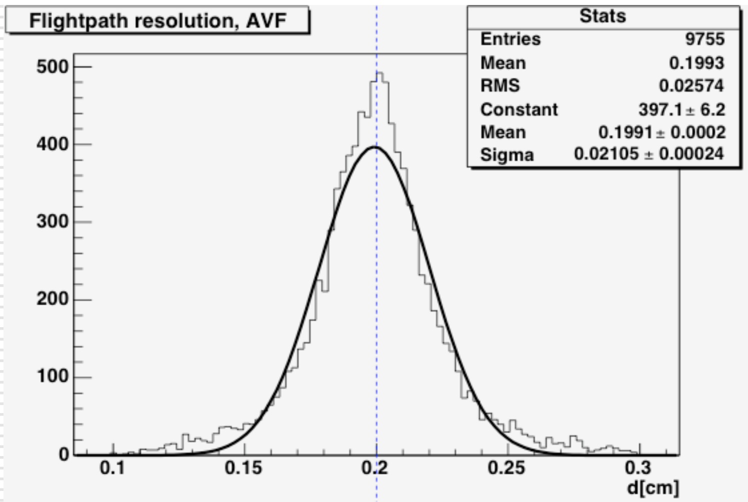
Adaptive Vertex Fitting

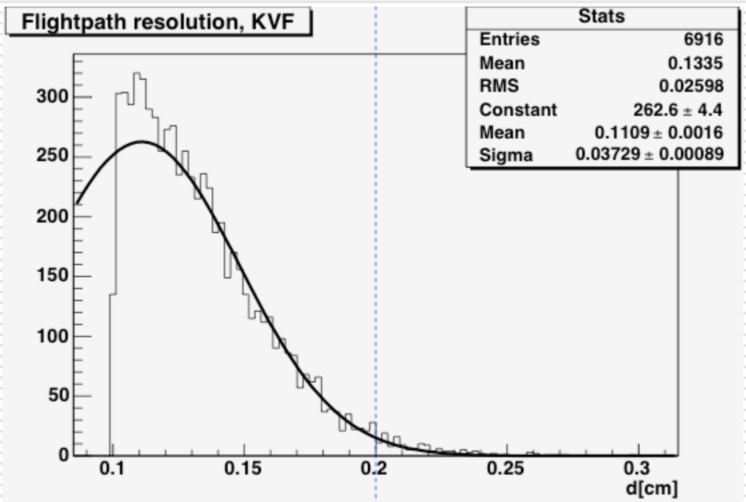
- Robustness against type 1 outliers
- Full simulation H→ZZ→eeμμ, only primary tracks

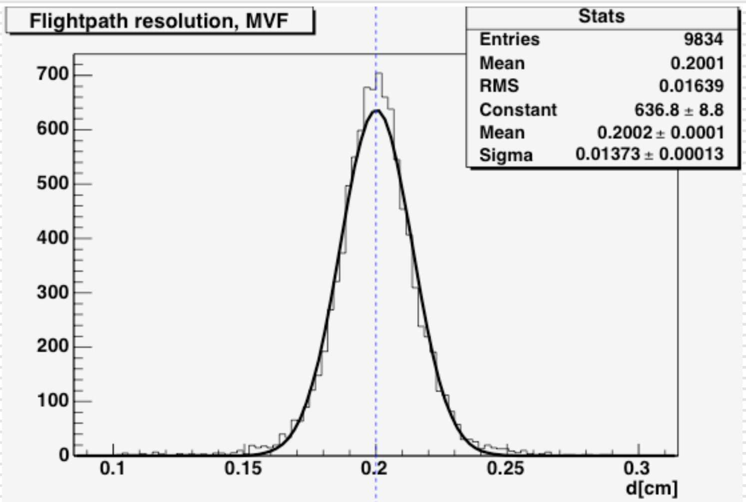


Multi Vertex Fitting

- **Try algorithm similar to multi-track fit:**
  - Adaptive weights / vertex / track
  - Several vertices compete for a track
- **Result from simplified simulation**
  - 5 + 3 tracks from 2 vertices separated by 2mm
  - Initialization with 2 tracks exchanged between vertices







Gaussian Sum Vertex Fitting

- **Previous algorithms: adapt to unknown noise distribution**
- **Alternative: as in the GSF for tracks, use prior knowledge**
  - Using input tracks obtained with a GSF
  - Using parametrizations of the resolution
  - Adding a track to the vertex increases the number of components in the vertex → need combination
  - Filtering and smoothing has been implemented
    - A combination of the predicted states of the “forward” and “backward” fit can be used to update track parameters

- **Simplified simulation**
  - 4 tracks simulated with two components
    - Nominal and 10x nominal resolution (fractions 0.9 / 0.1)
  - Comparing KF (using nominal resolution) to GSF

Gaussian Sum Vertex Fitting

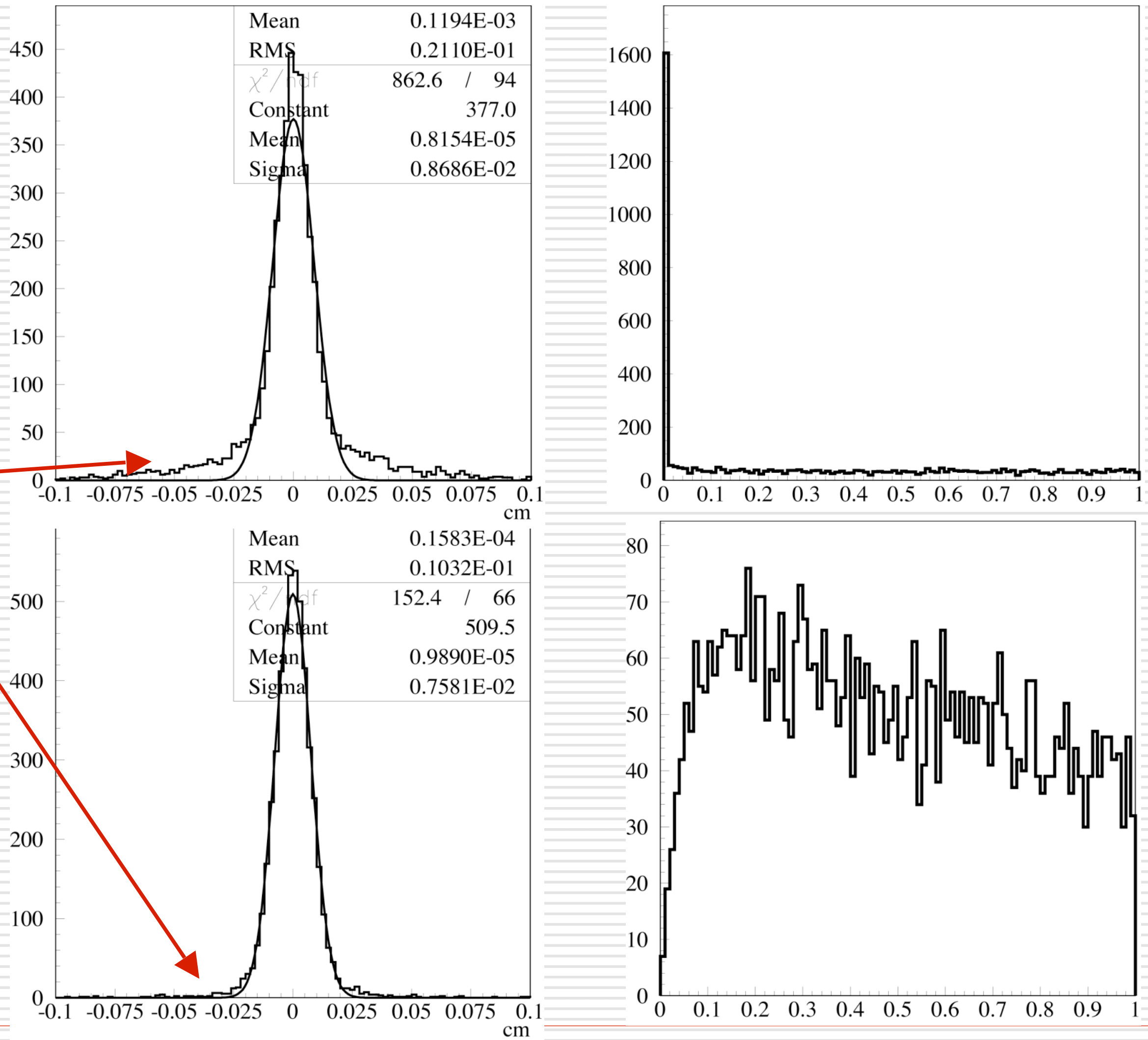
x-residuals

χ^2 probabilities

Least squares

Tails disappear

Gaussian sum



Conclusions & Outlook

- **A multitude of tracking and vertexing algorithms have been implemented and tested in the CMS reconstruction program (only a selection could be presented today).**
- **Essential points:**
  - The modular design allows to easily exchange single blocks in the reconstruction chain (finding, fitting, …)
  - New versions of the basic building blocks - measurements and states - can be introduced and are usable by existing modules
  - The availability of a simplified simulation - with controlled topology and process / measurement noise - is essential for the rapid development of new algorithms.
  - The results of all standard algorithms are available to the user in a uniform way (requests for collections of objects with defined configuration of parameters and algorithmic components) - either from persistent store or reconstructed on demand

Conclusions & Outlook

- **Adaptive algorithms show a clear advantage over classical solutions**
  - They are robust
  - They are (much) slower than a plain least squares fit, but faster than combinatorial approaches
  - More testing and tuning will be needed on detailed simulation (and data!)
- **Two classes of algorithms have been shown**
  - Algorithms without prior knowledge of outlier distributions
    - Weights are adjusted in an iterative process
    - Measurements can compete for association in the fitted object, and vice versa
    - Global minimum is found using annealing
  - Algorithms using a description of non-Gaussian components
    - General pdf's modelled using a mixture of Gaussian components
    - Weights are adjusted to measurements in a single pass using a Bayesian approach

Conclusions & Outlook

Current priority in track reconstruction:

- **Alignment**
  - Simulation of realistic misalignment scenarios
  - Use of alignment hardware (lasers)
  - Several track-based alignment algorithms in development
  - Non-trivial task: ~ 16000 elements x 6 degrees of freedom!
- **Calibration**
  - Realistic use of calibration constants read from a DB
  - Gains, noise levels, detector status, …

CMS has started working on a new EDM

- **New software framework**
  - Persistency and access to reconstructed objects
  - Scheduling and configuration of reconstruction
  - …
- **Still in prototype phase**

References

A short list of CMS related notes and publications. For more details and information on the original papers please see their list of references.

- CMS Collaboration, TriDas project, Technical Design Report vol. 2, CERN LHCC 02/26
- T. Boccali et al, "Vertex Reconstruction Framework and its implementation for CMS", CMS Conference Report 2003/035
- W. Waltenberger, "Adaptive Multi-vertex fitting", CMS Conference Report 2004/062
- W. Adam et al, "Reconstruction of electrons with the Gaussian-sum filter in the CMS tracker at the LHC", CMS Note 2005/001
- T. Speer and R. Frühwirth, "A Gaussian-sum Filter for Vertex Reconstruction", CMS Note 2005/005
- J. D'Hondt et al, "Sensitivity of Robust Vertex Fitting Algorithms", CMS Note 2004/002

- CMS Collaboration, TDR on physics and reconstruction, in preparation

Acknowledgments

What I have presented today is the result of several years of work of many CMS physicists and students, and I would like to acknowledge the contribution of all the members of the CMS tracking and vertexing groups!

Special thanks to R. Frühwirth, T. Todorov and P. Vanlaer.

And, of course, many thanks to

Andreas Meyer &
Andreas Gellrich

for inviting me to this seminar!