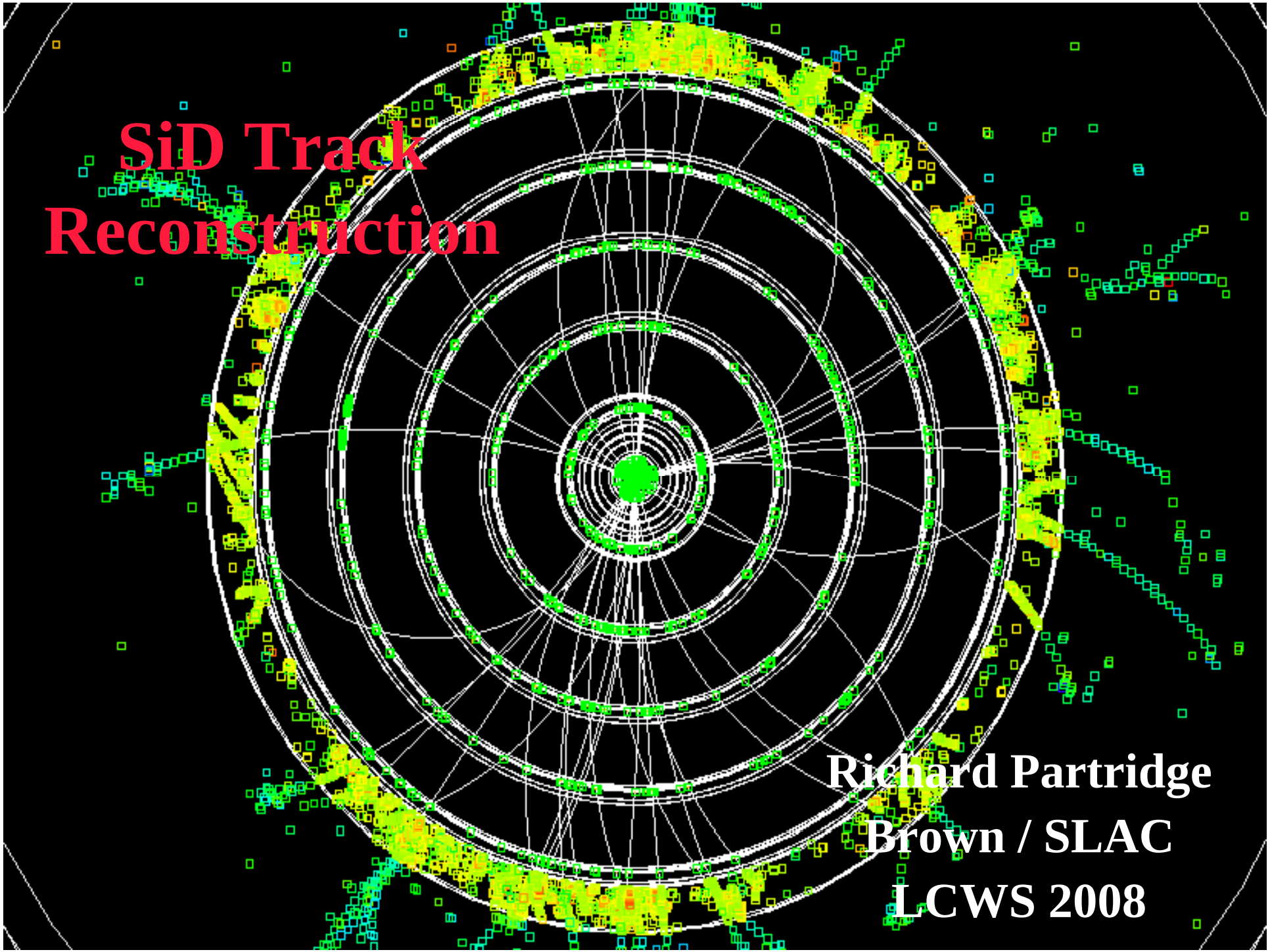
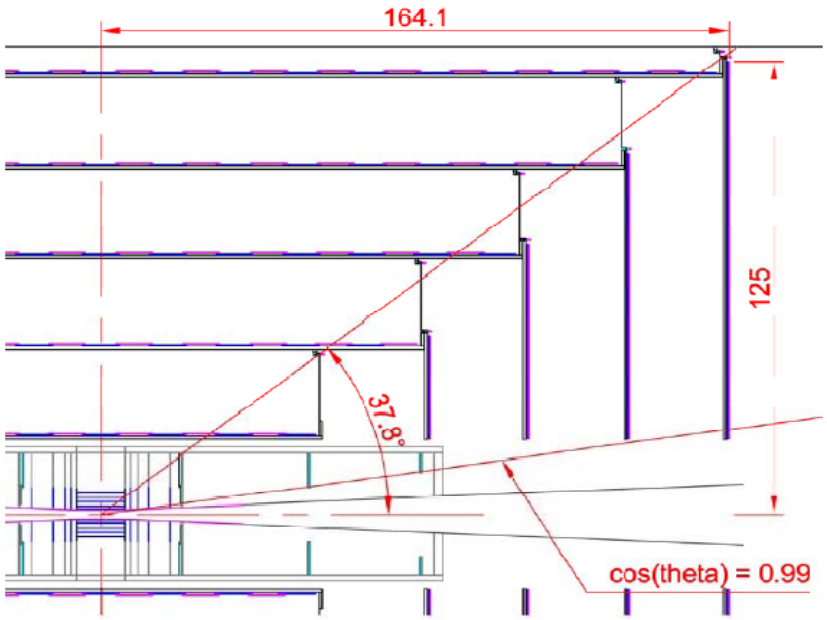
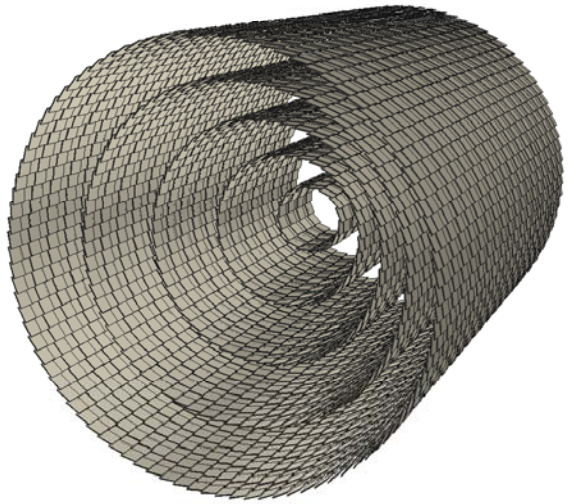

SiD Track
Reconstruction
Richard Partridge
Brown / SLAC
LCWS 2008

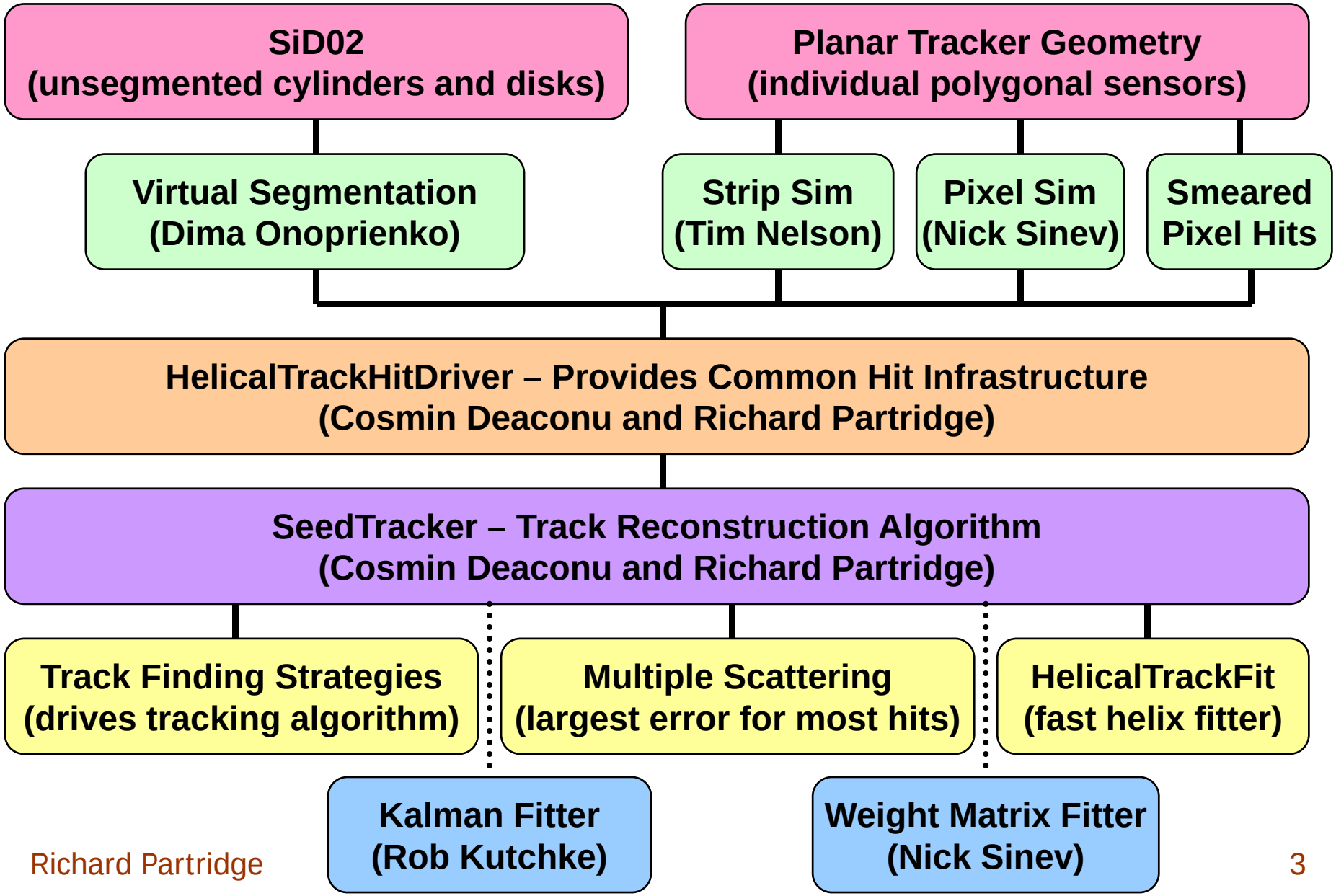
Overview

- SiD proposes an all-Si Tracker
  - 5 barrel + 7 disk pixel inner detector
  - 5 barrel (axial strip) + 4 disk (stereo strip) outer detector
- New tracking software developed using org.lcsim framework
  - Supports any combination of pixel, axial strip, and stereo strip layers in barrel or disk geometry
  - Detector segmentation can be either cylinders/disks with virtual segmentation or individual planar sensors
  - Digitization options include full simulation of charge collection for strips and pixel detectors (using field map)
  - Flexible track finding developed explicitly for tracker design studies



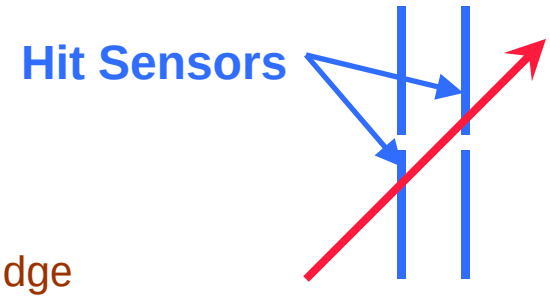


Track Reconstruction Flow Chart



Common Hit Infrastructure

- Tracker hit infrastructure is rather weak in LCIO
  - Key features (e.g. sensor orientation and endpoints of a strip) hit are missing
  - Hit errors are represented by a 3x3 covariance matrix, which is intrinsically singular for 1 coordinate (strip) or 2 coordinate (pixel) measurements
- Virtual segmentation and full digitization algorithms extend LCIO conventions, but take different approaches
- Common hit infrastructure developed to shield the tracking code from the details (and changes) in the hit digitization
- Infrastructure also provides extensive support for stereo hits
  - Forms stereo hits (and ghost hits) from nearby non-parallel strips
  - Adjusts hit position for track direction
  - Uncertainty in hit position includes uncertainty in the track direction

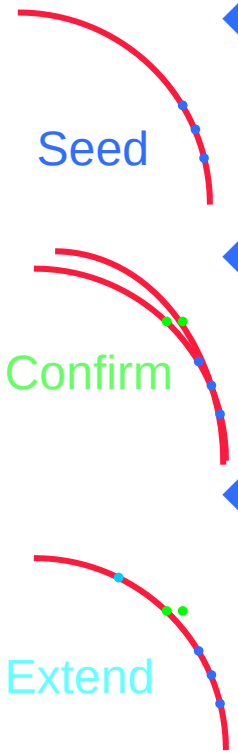


Common Hit Infrastructure provides robust handling of stereo hits, including cases like the one show here

SeedTracker Philosophy

- Track finding is guided by a set of user defined "Strategies"
  - A strategy defines layers to be used, their roles, and constraints (e.g. p_T>x)
- All pattern recognition code is agnostic as to the type of hit
  - No differentiation between pixel or strip, barrel or forward sensors
- Multiple Scattering must be accounted for in track finding
  - Superb intrinsic pixel/strip resolution ⇒ MS errors will typically be dominant
- A fast helix fitter, HelicalTrackFitter, plays a central role
  - This is the only piece of code that needs to understand the differences between pixels and strips, barrels and disks, etc.
- All decisions based on a global χ^2 from fits, constraints, etc.
  - No internal parameters or tuning is required if tracker geometry changes
  - Constraint example: if $(|z_0| > z_0^{max})$ $\chi^2 = \chi^2 + (|z0| - z_0^{max})^2 / \sigma^2(z_0)$
- Maximize flexibility for detector design optimization
  - No aspect of the detector geometry is hard coded
  - Automated strategy generation further simplifies performing design studies

SeedTracker Algorithm



- Track finding begins by forming all possible 3 hit track seeds in the three "Seed Layers" (specified in the strategy)
  - Brute force approach to finding all possible track seeds
- Typically require the presence of a hit in a "Confirmation Layer" (specified in the strategy)
  - Significantly reduces the number of candidate tracks to be investigated
- Add hits to the track candidate using hits on the "Extension Layers" (specified in the strategy)
  - Discard track candidates that have fewer that the minimum number of hits specified in the strategy
  - If two track candidates share more than one hit, best candidate is selected
- Upon each attempt to add a hit to a track candidate, a helix fit is performed and a global χ^2 is used to determine if the new track candidate is viable
- Hooks for user-defined diagnostics at all decision points

Track Finding Strategy

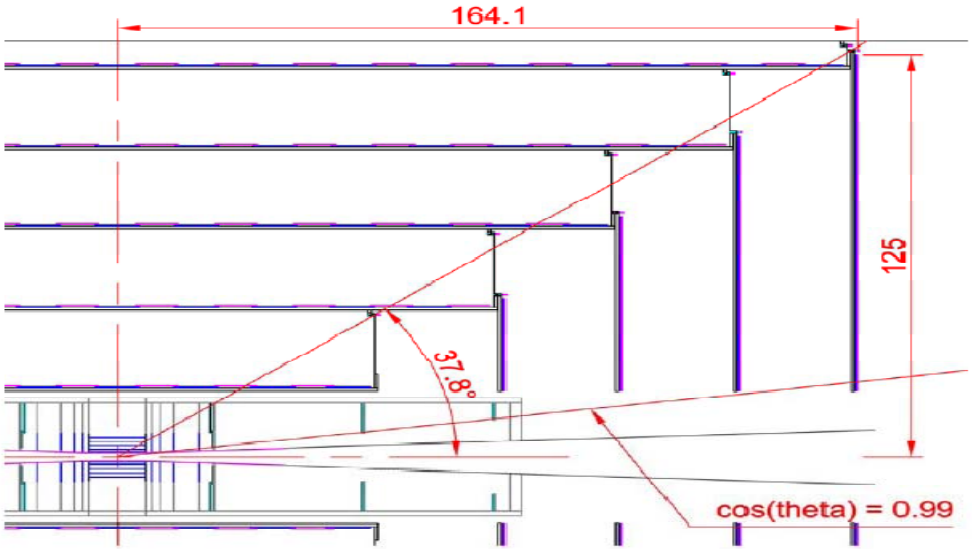
- The user interacts with the track reconstruction program by specifying one or more “strategies”
- Strategies identify:
  - Layers to be used in track finding
  - Role of each layer (seed, confirm, extend)
  - “Cutoffs” on helix parameters (p_T, d_0, z_0) where a χ^2 penalty is introduced
  - Minimum number of confirmed hits and total hits
  - Cut on global χ^2
  - “Bad Hit χ^2” – a χ^2 increase exceeding this amount will flag hit as suspect
- Tracking code processes all strategies sequentially
  - Final list of tracks is the union of all distinct tracks
- Strategies can most easily specified using an xml file, but may be hard coded if desired

Example xml Strategy File

```
<?xml version="1.0" encoding="UTF-8"?>
<StrategyList>
  <Strategy name="OutsideInBarrel">
    <!--Cutoffs-->
    <MinPT>1.0</MinPT>
    <MinHits>7</MinHits>
    <MinConfirm>1</MinConfirm>
    <MaxDCA>10.0</MaxDCA>
    <MaxZ0>10.0</MaxZ0>
    <MaxChisq>50.0</MaxChisq>
    <BadHitChisq>15.0</BadHitChisq>
    <!--Layers-->
    <Layer type="Extend" layer_number="1" detector_name="SiVertexBarrel" be_flag="BARREL" />
    <Layer type="Extend" layer_number="2" detector_name="SiVertexBarrel" be_flag="BARREL" />
    <Layer type="Extend" layer_number="3" detector_name="SiVertexBarrel" be_flag="BARREL" />

    ...

    <Layer type="Confirm" layer_number="2" detector_name="SiTrackerBarrel" be_flag="BARREL" />
    <Layer type="Seed" layer_number="3" detector_name="SiTrackerBarrel" be_flag="BARREL" />
    <Layer type="Seed" layer_number="4" detector_name="SiTrackerBarrel" be_flag="BARREL" />
    <Layer type="Seed" layer_number="5" detector_name="SiTrackerBarrel" be_flag="BARREL" />
  </Strategy>
  <Strategy name="OutsideInEndcap">

    ...

  </Strategy>

...
</StrategyList>
```

Strategy Builder

- Finding an optimal set of strategies that provides complete coverage turns out not to be so easy
  - Many distinct sets of layers are required, especially in the forward region
  - Requires carefully examining possible track paths looking for coverage holes
  - Typically need ~20 strategies to have full coverage for baseline tracker design to find ≥7 hit tracks with p_T > 1.0 GeV for 100% detector efficiency
- Strategy list needs to be re-optimized whenever:
  - Change detector geometry
  - Change helix cutoffs
  - Change number of hits required



- Strategy Builder automates creation of optimized strategy list

Strategy Builder Algorithm

- Starting point is a file of simulated events that you want to use for optimization
  - Simulated top events would be a good choice since they have a wide variety of particles in them
- Findable MC particles are identified based on helix cutoffs and minimum hit requirements
- All potential combinations of seed / confirm layers are identified and ranked by a weighted frequency of occurrence
  - Weighting can favor / disfavor using particular layers
  - Weighting can also favor combinations with greater adjacency between layers
  - Can optionally randomly discard MC hits to simulate detector inefficiency
- Strategy builder chooses the highest ranked strategy
  - Remove MC Particles from top ranked strategy and re-do ranking
  - Iterate until desired track finding efficiency is achieved

Multiple Scattering

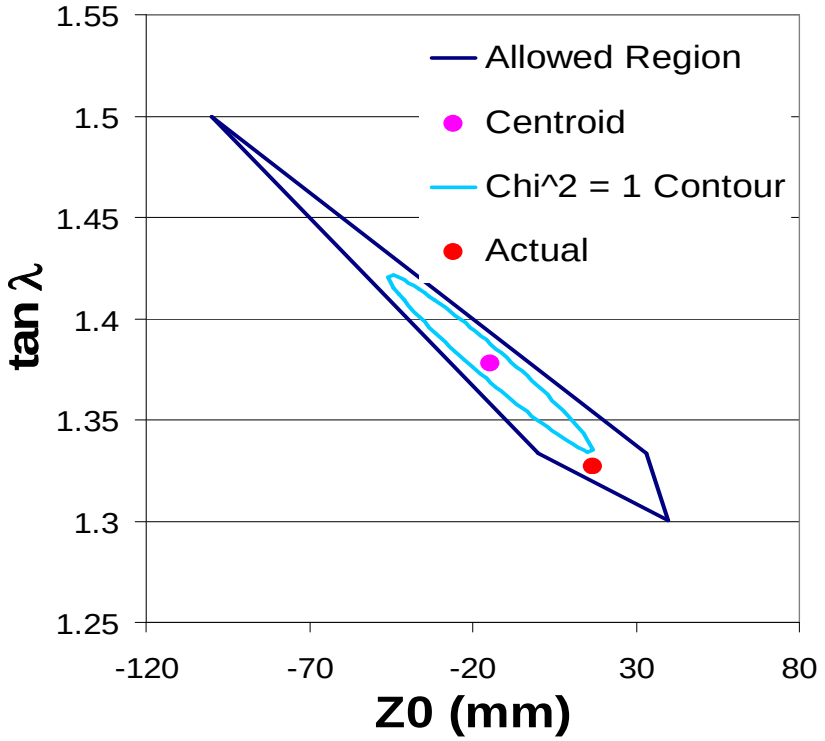
- Multiple scattering errors dominate for most tracks, so having a reasonable estimate of these errors is critical to form a sensible χ^2 discriminator
- SeedTracker constructs a model of the tracker material
  - All material that derives from a common element in the compact.xml description is lumped together
  - Material is modeled as either a cylinder or disk depending on aspect ratio as seen from the origin
- Multiple scattering errors are assigned to each hit
  - Tracks are assumed to originate from the point of closest approach
  - For a given hit, the multiple scattering error is the cumulative error from all the material crossed in getting from the point of closest approach to the hit
- Multiple scattering correlations are ignored for track finding
  - This isn't really true - a given multiple scattering will systematically affect all subsequent hits – but shouldn't have a big effect on track finding

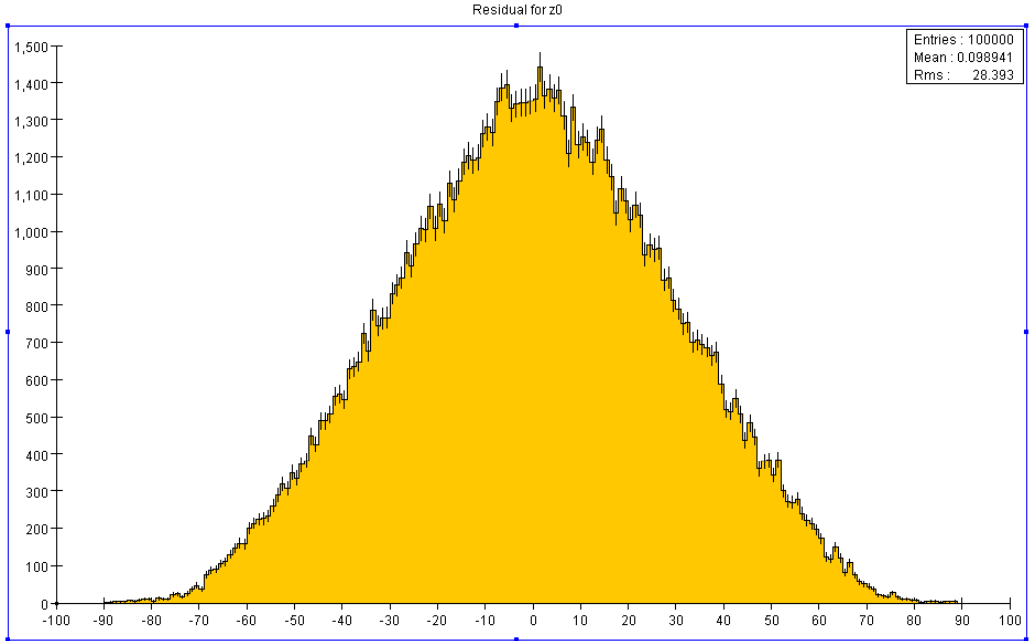
Fast Helix Fitting

- Approximate helix by fitting a circle in x – y and a line in s – z
  - Circle fit uses Karimaki algorithm
  - s-z fit uses a straight line fitter or ZSegment fitter when there are <2 3D hits
- First fit for a given 3 hit seed is performed without MS errors
  - First determination of the helix parameters $\omega \equiv 1/R$, d_0, ϕ_0, z_0, and $\tan(\lambda)$
- Calculate the MS errors for each hit using this helix
- Perform a second helix fit including MS errors
- If necessary, calculate a constraint χ^2 to estimate the increase in χ^2 needed to pull helix into compliance with the constraint
  - Kinematic constraints: $p_T > p_T^{min}$, $|d_0| < d_0^{max}$, $|z_0| < z_0^{max}$
  - Geometric constraints: ensure helix is consistent with strip endpoints
  - Example: if ($|z_0| > z_0^{max}$) $\chi^2 = \chi^2 + (|z0| - z_0^{max})^2 / \sigma^2(z_0)$
- Reject seeds that fail the χ^2 cut
  - χ^2 cut is applied to the sum of the fit χ^2 and the constraint χ^2

Helix Fitting with only Strip Hits

- Strips are bounded in z ⇒ for 2 or more strip layers there are constraints on the helix paramaters z_0 and tanλ
  - Results in a polygonal allowed region in z_0 – tanλ parameter space
  - "Fit parameters" are taken from centroid of allowed region
  - Covariance matrix calculated assuming all points in allowed region of parameter space are equally probable





z_0 Residual Distribution

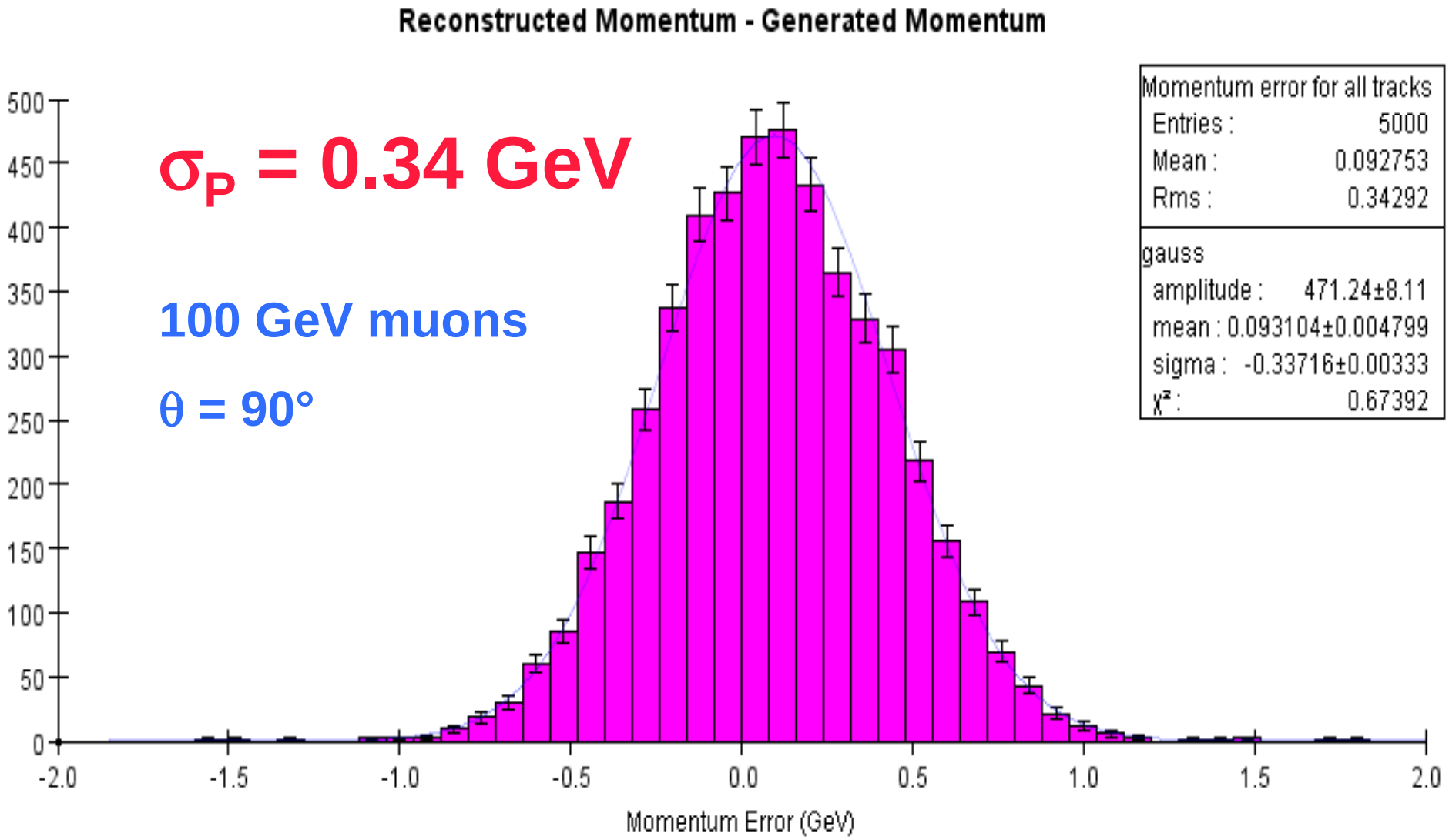
Track Fitting

- Reconstructed tracks are saved into the event with helix parameters and covariance matrices obtained from the fast helix fitter
- These fits are not true helix fits
  - Separate circle / line fits instead of a true helix fit
  - Multiple scattering correlations not included
- Two track fitting approaches have been pursued in SiD
  - Kalman filter track fitter
  - Weight matrix track fitter
- Additional work is required before we can perform true helix fits on the reconstructed tracks
- Goal is to have at least one helix fitter running by the time the LOI is submitted

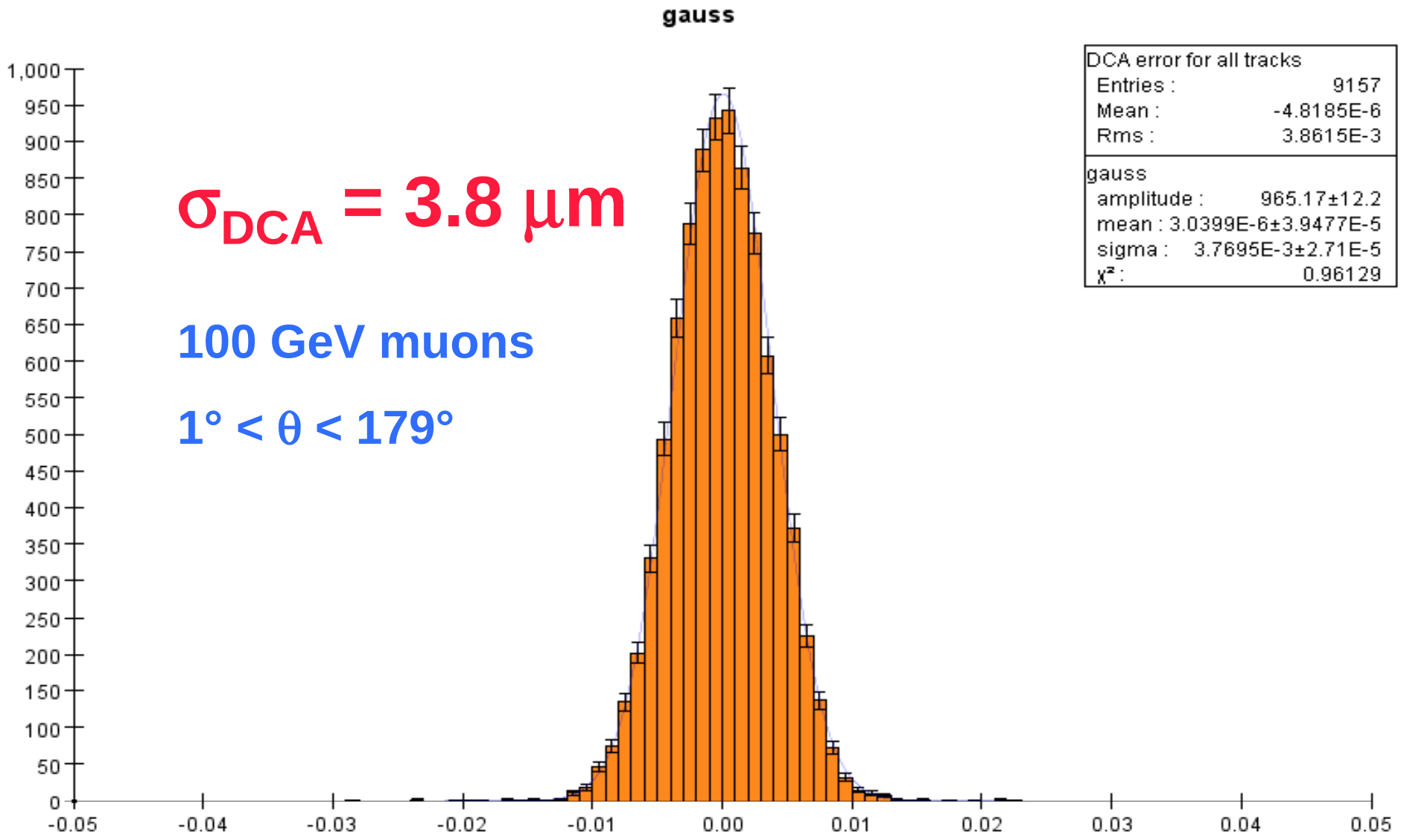
Tracking Resolution Study

- Tracker resolution has been studied using samples of single muons at various momenta and angles
- Resolution measured by comparing the measured track parameters with the MC parameters for the muon
- Some caveats on these results:
  - Results are from fast fitter used in track finding – not a true helix fitter
  - Expect that these are “worst case” results

Momentum Resolution



DCA Resolution

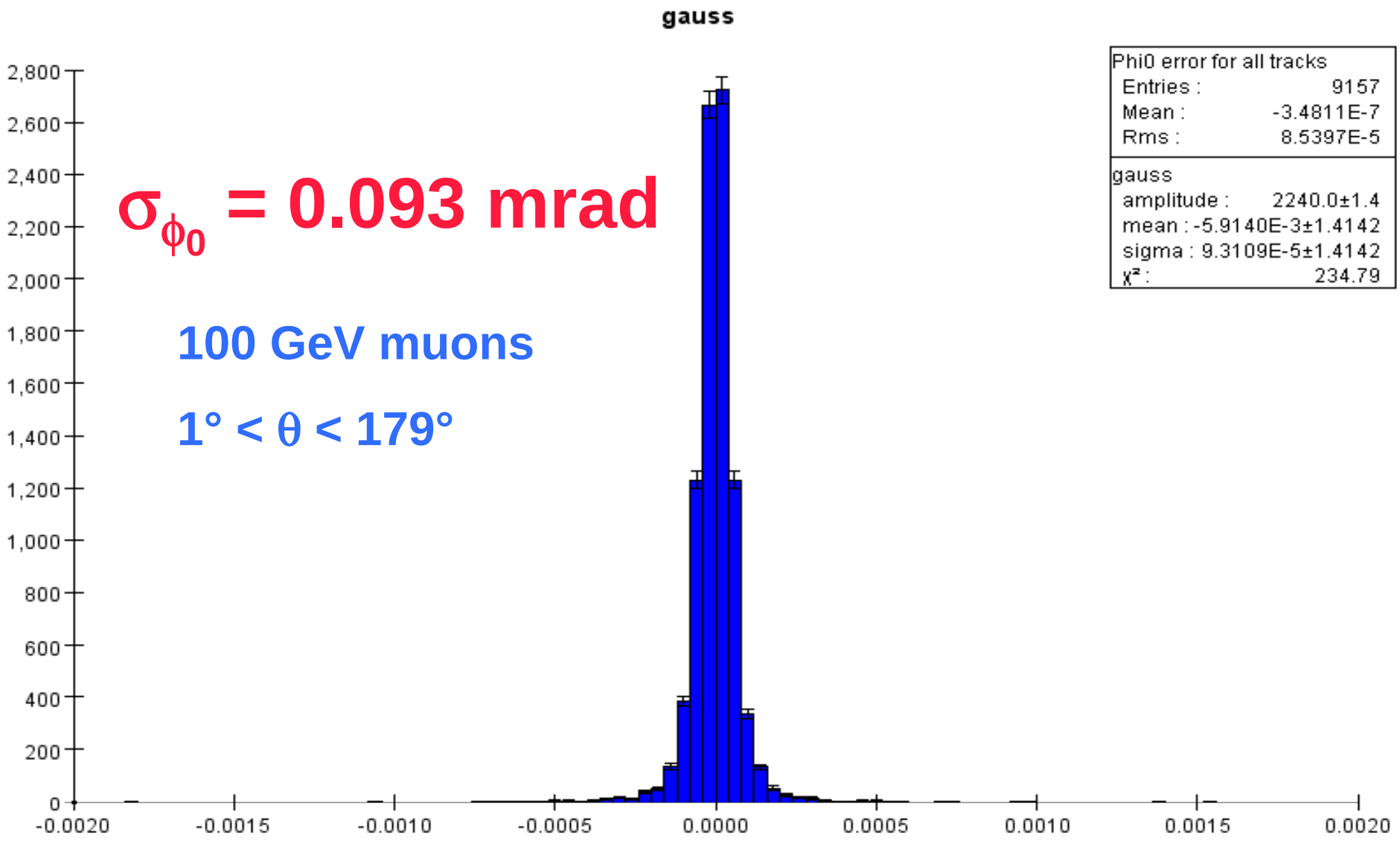


$\sigma_{DCA} = 3.8\ \mu m$

100 GeV muons

$1° < \theta < 179°$

ϕ_0 Resolution

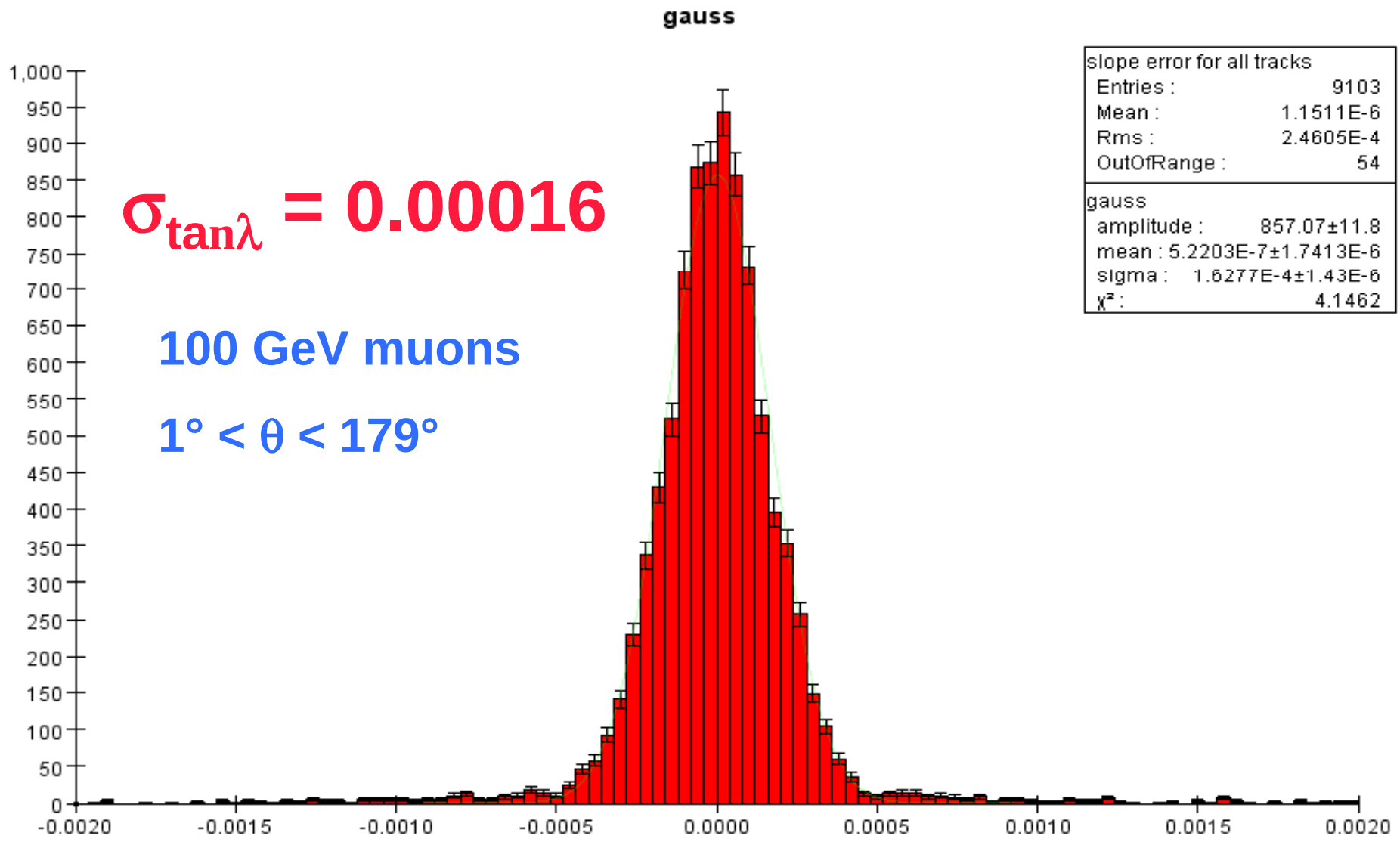


σ_{ϕ_0} = 0.093 mrad

100 GeV muons

1° < θ < 179°

tan(λ) Resolution

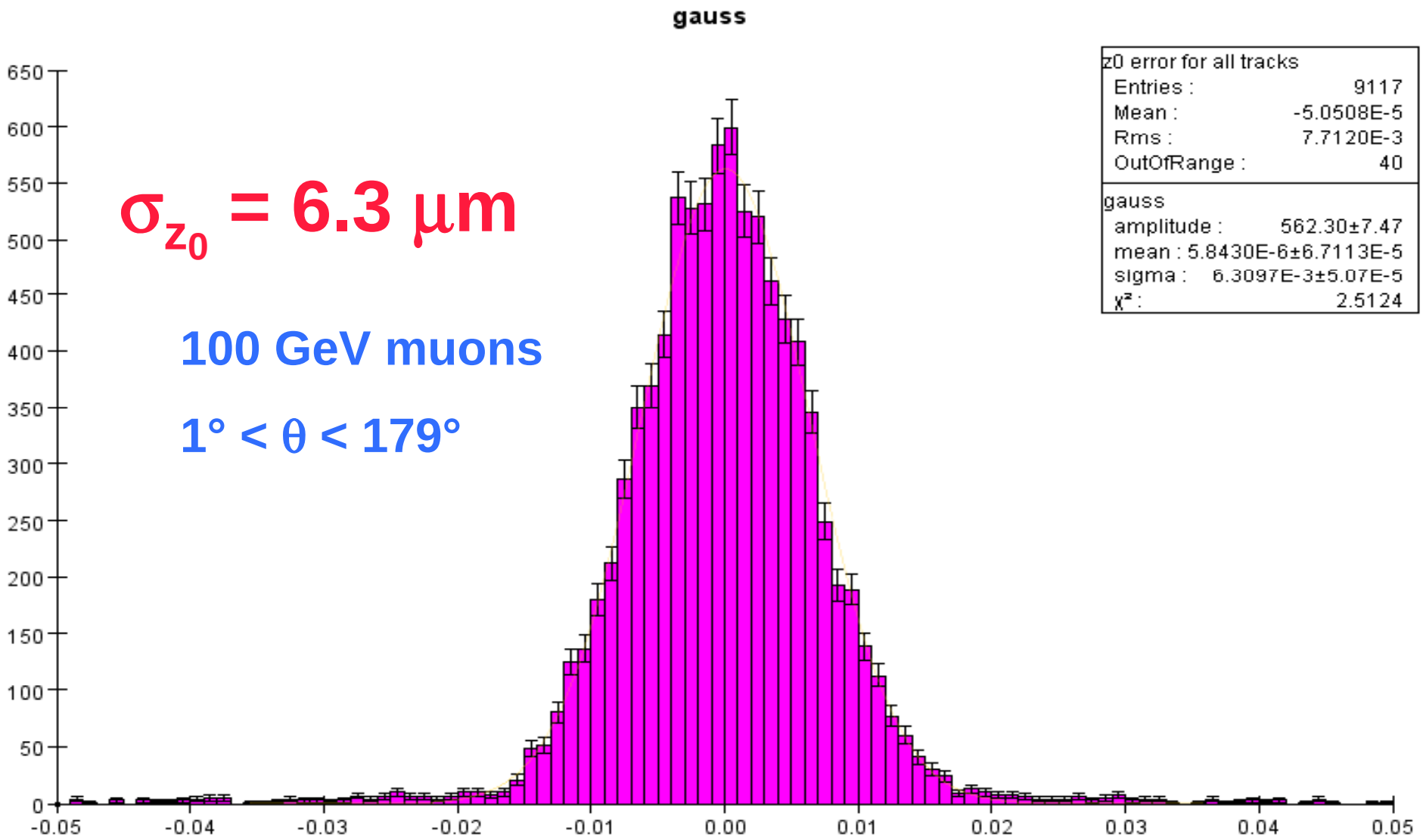


$\sigma_{\tan\lambda} = 0.00016$

100 GeV muons

$1° < \theta < 179°$

z_0 Resolution



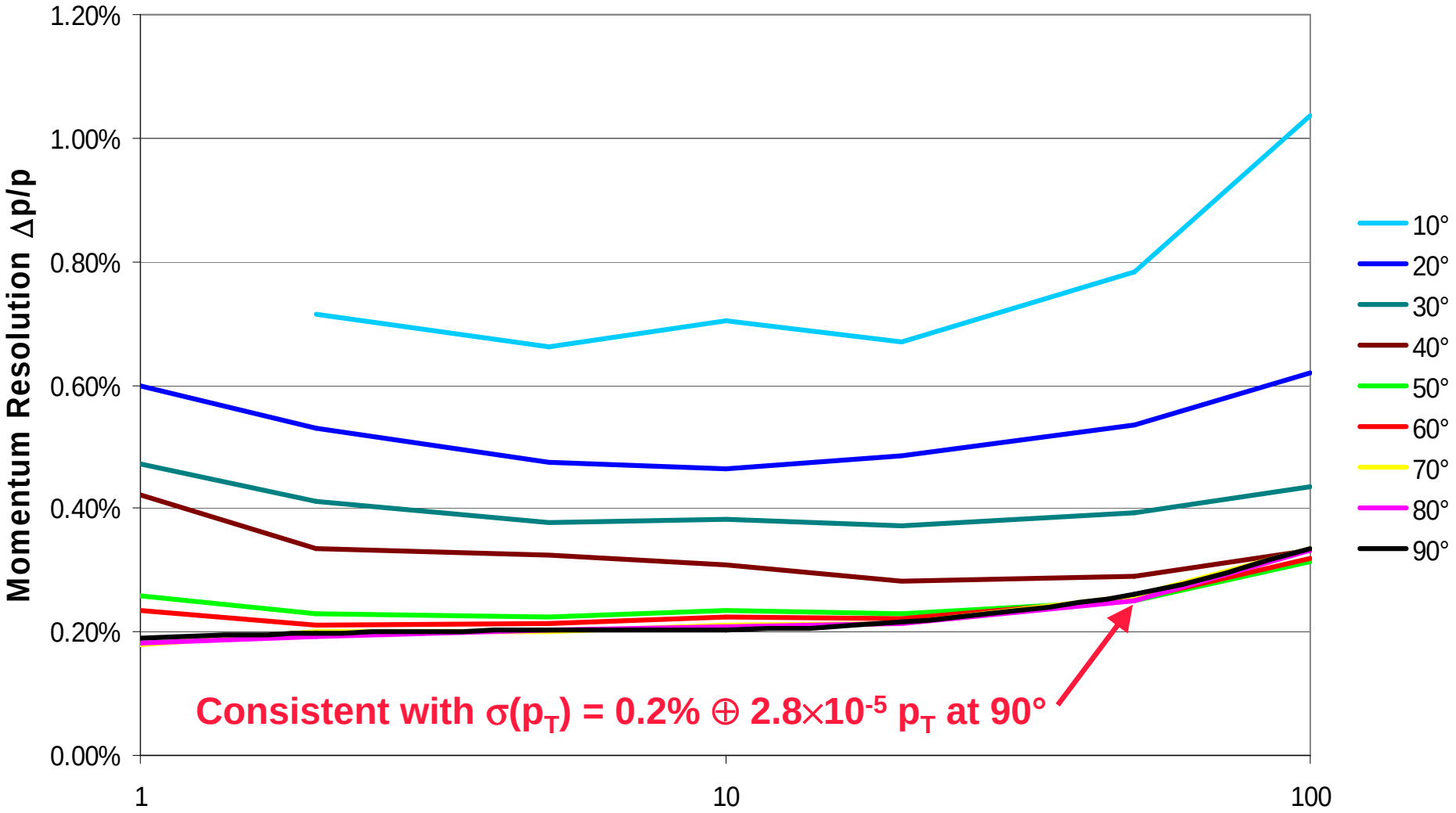
$\sigma_{z_0} = 6.3\ \mu m$

100 GeV muons

$1° < \theta < 179°$

Momentum Resolution

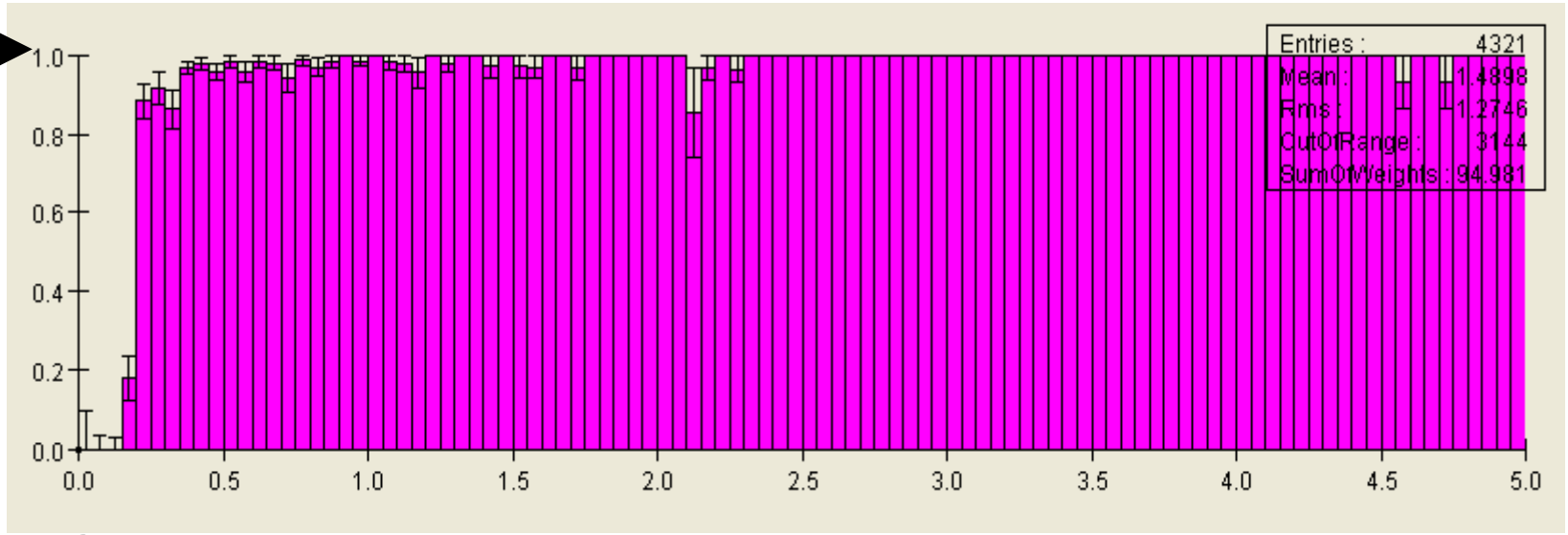
- Good momentum resolution everywhere!



Efficiency vs p_T for 6-Jet $t\bar{t}$ Events

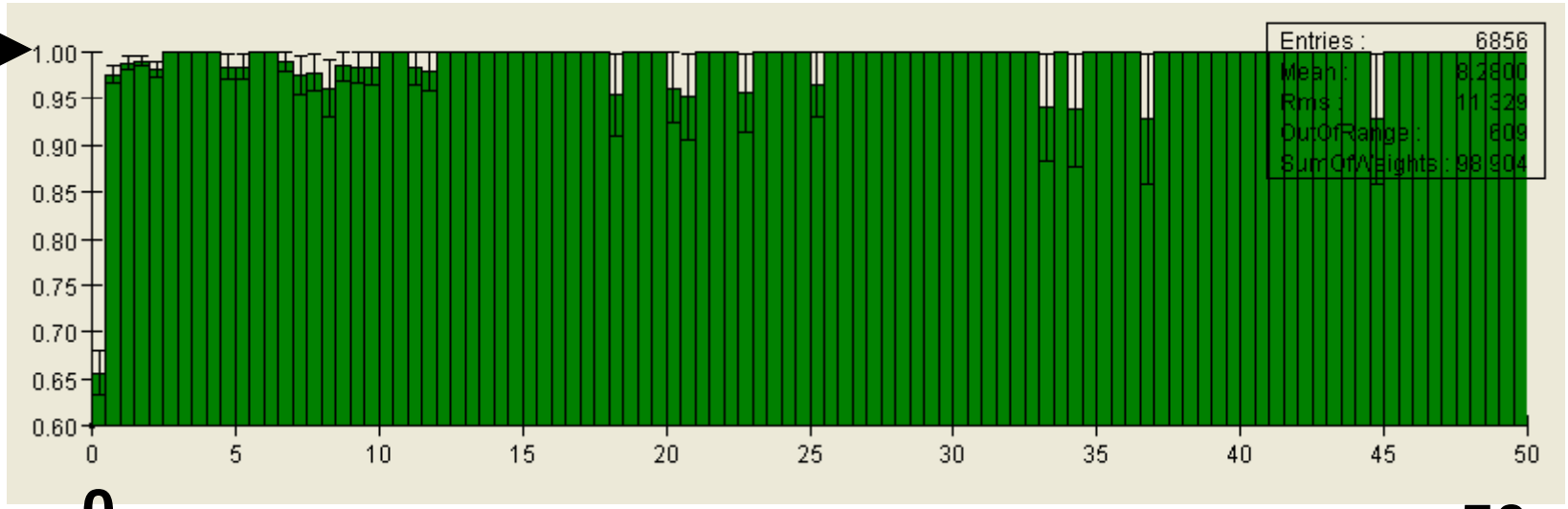
100% →

Efficiency for tracks satisfying 1 cm IP cut



0 **5**

100% →

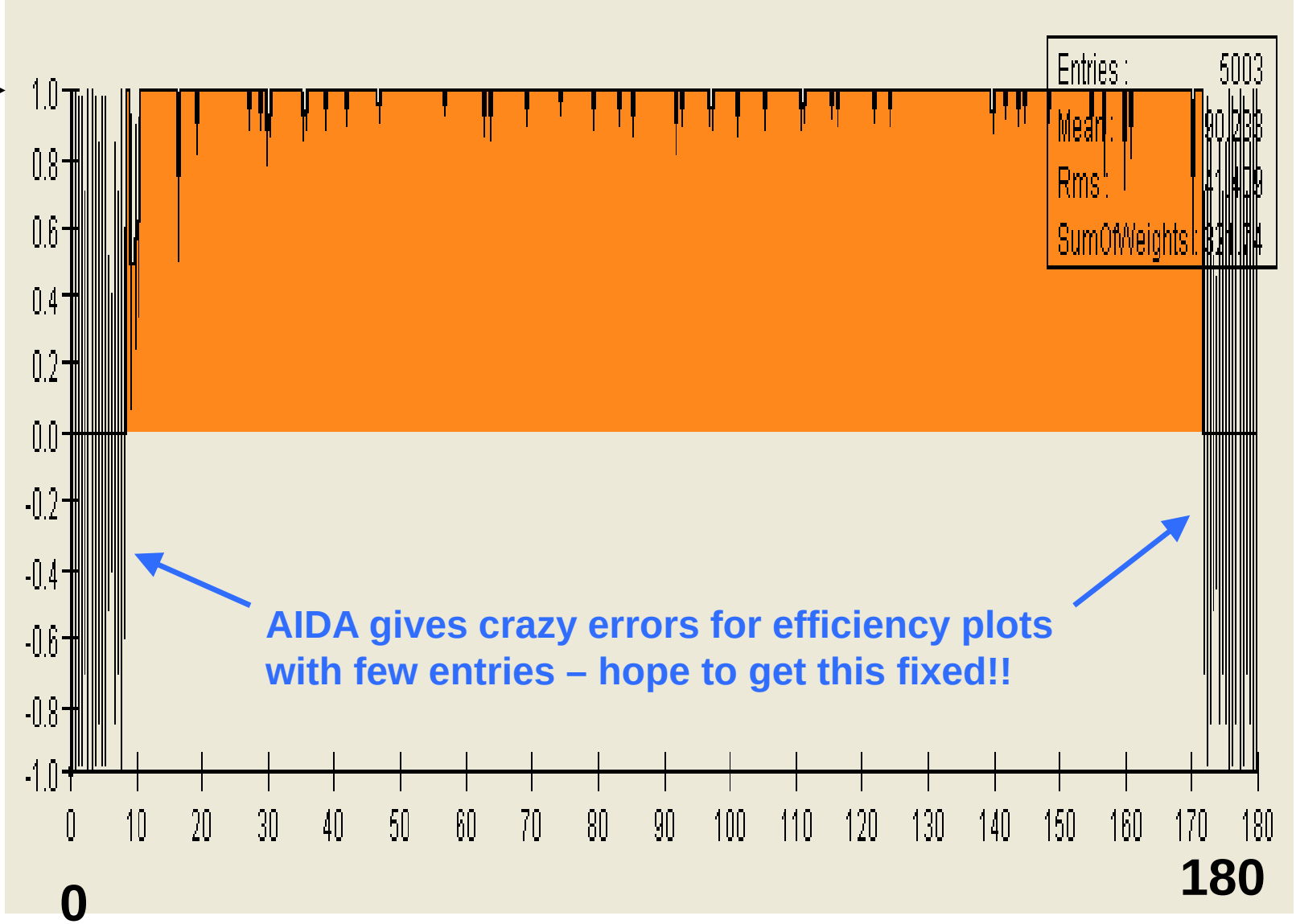


0 **50**

Efficiency vs θ for 6-Jet $t\bar{t}$ Events

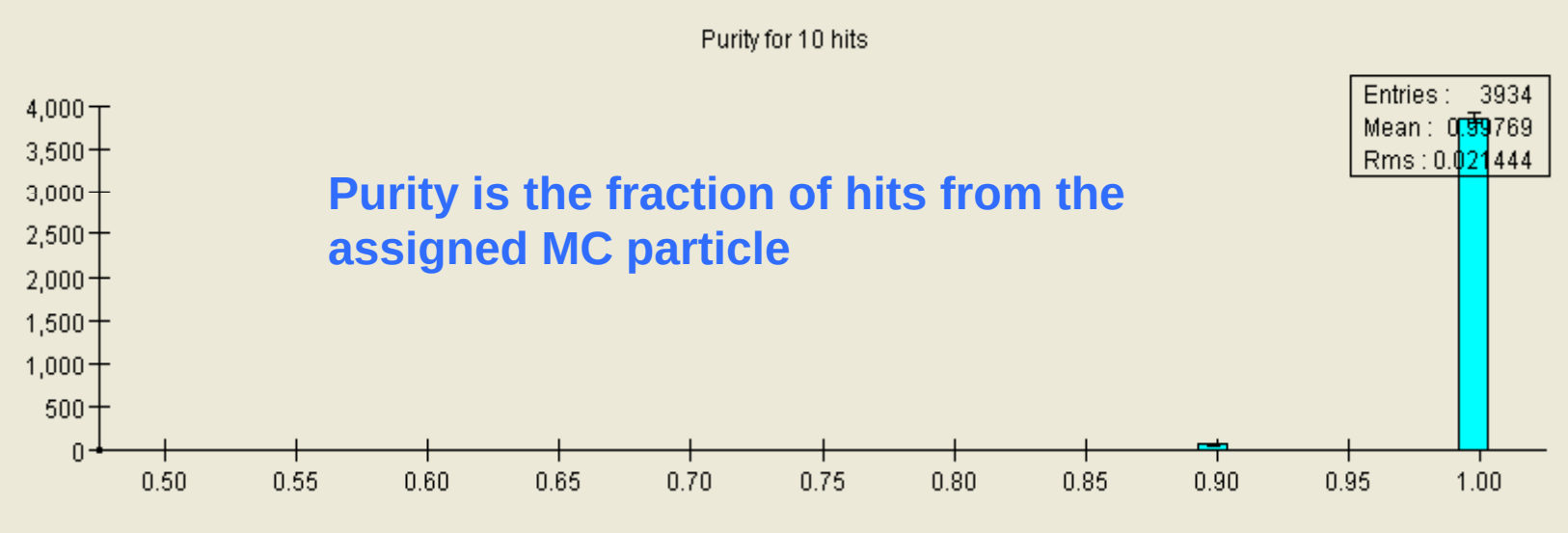
100%

Efficiency for tracks satisfying 1 cm IP cut



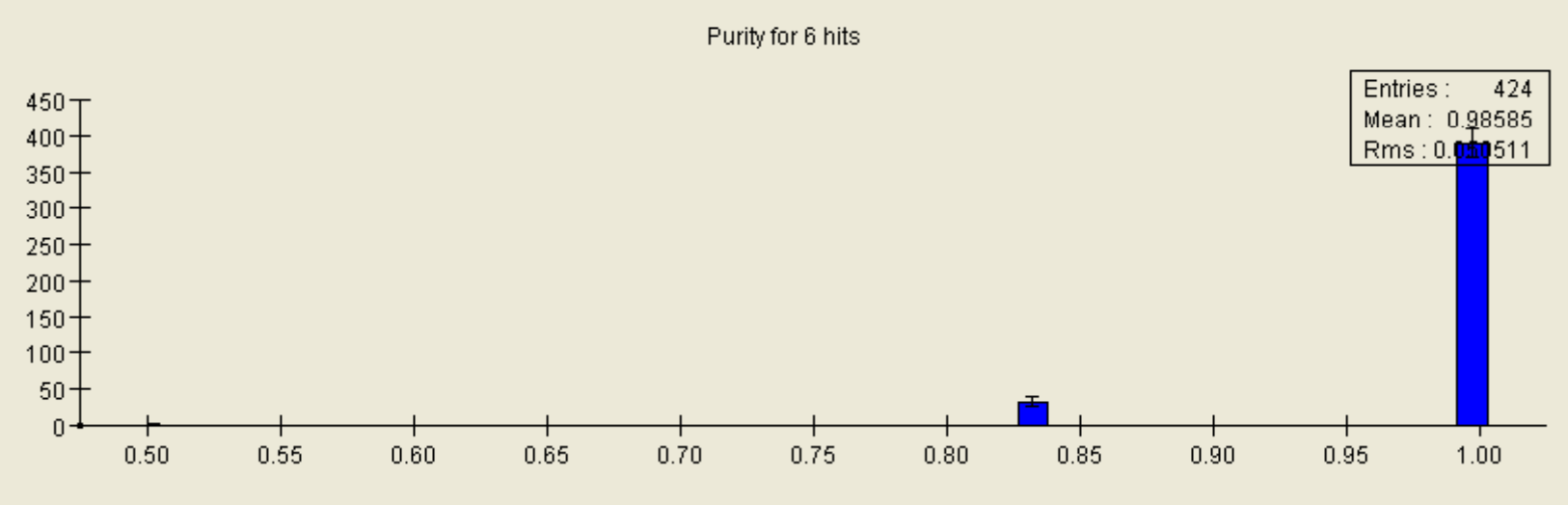
Purity for for 6-Jet $t\bar{t}$ Events

10 Hit Tracks



Purity is the fraction of hits from the assigned MC particle

6 Hit Tracks



Summary

- SiD has developed a new track reconstruction code in the org.lcsim framework with a number of innovative features
  - Algorithm expressly developed for silicon tracker design studies
  - New (?) algorithm for performing full helix fit for set of axial hits
  - 3D stereo hit positions / covariance matrix calculated including dependence on track angle and helix uncertainties (new?)
  - All decision based on χ^2 cut – no internal tuning parameters
  - User controls track finding controlled through xml strategy file
  - Strategy builder automates generation of strategies
- Code has been used to characterize SiD tracker performance
  - Outstanding resolution over full solid angle
  - Full coverage with high efficiency for findable tracks
  - Tracks have high purity with very small fake rate