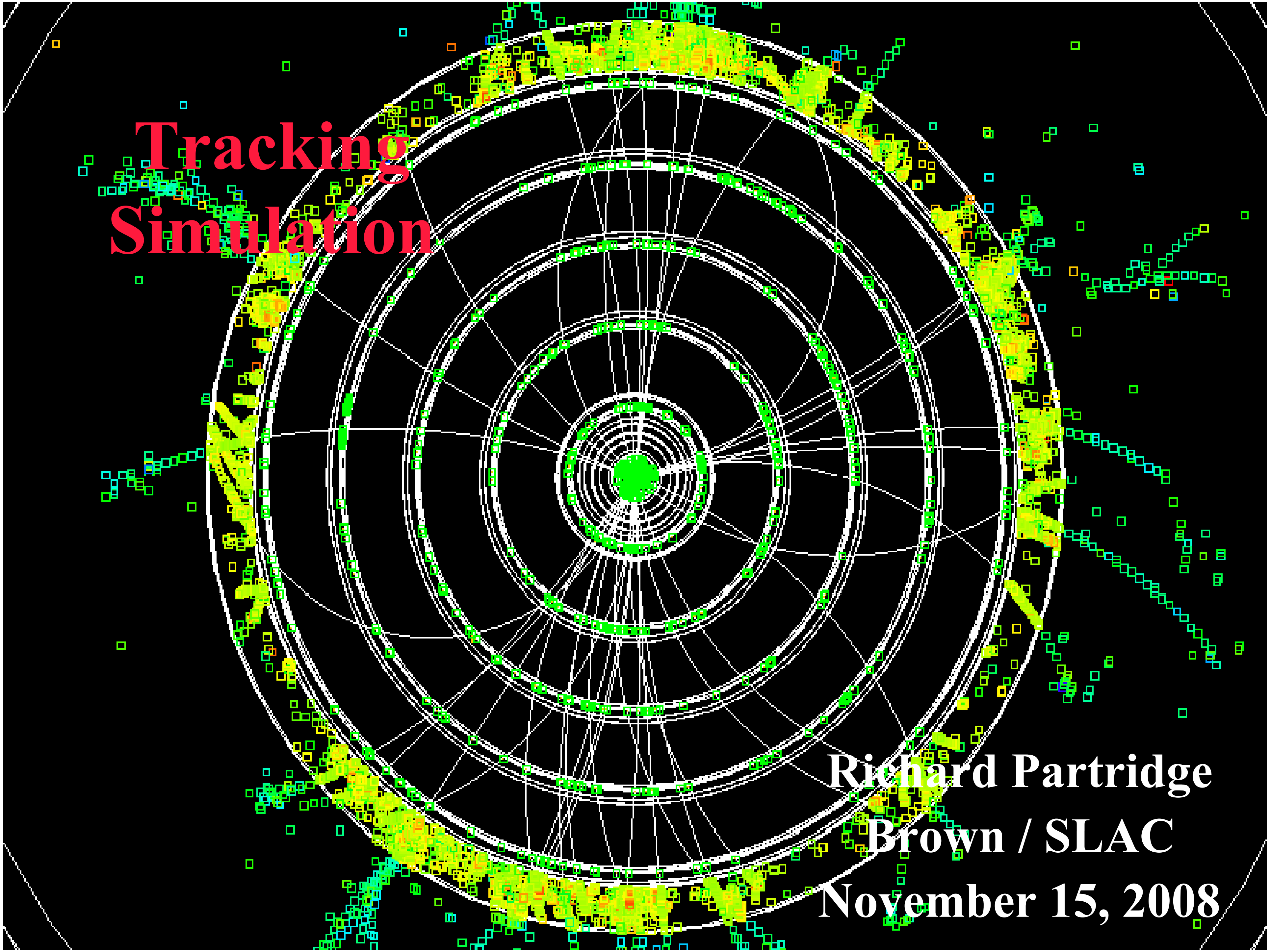


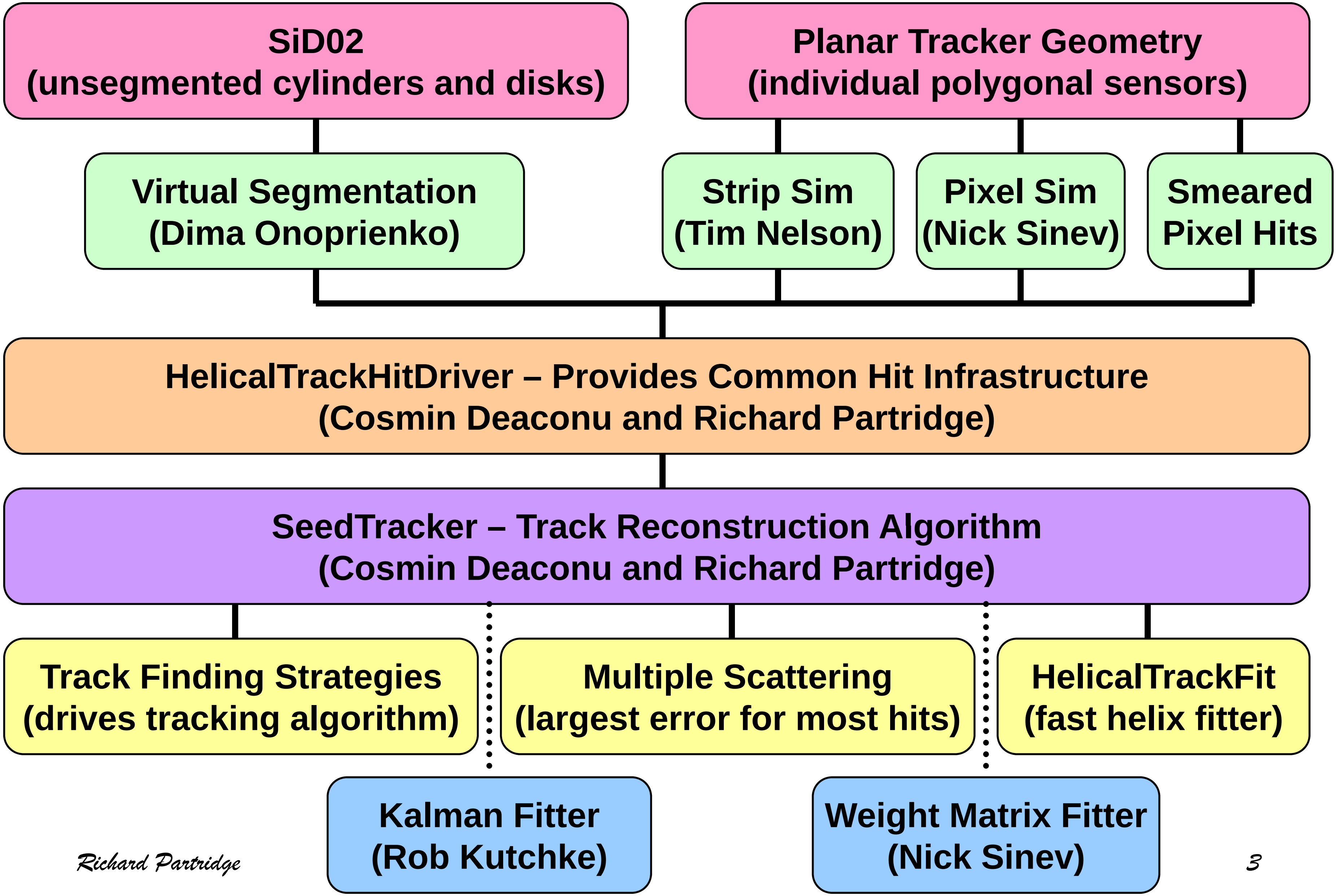
Tracking Simulation

Richard Partridge
Brown / SLAC
November 15, 2008

Outline

- Introduction to track reconstruction for SiD
  - For more details, see talk at Boulder SiD workshop
- Changes since the Boulder workshop
- Tracking resolution study
- Tracking efficiency
- Summary and future plans

Track Reconstruction Flow Chart



SiD02
(unsegmented cylinders and disks)

Planar Tracker Geometry
(individual polygonal sensors)

Virtual Segmentation
(Dima Onoprienko)

Strip Sim
(Tim Nelson)

Pixel Sim
(Nick Sinev)

Smeared
Pixel Hits

HelicalTrackHitDriver – Provides Common Hit Infrastructure
(Cosmin Deaconu and Richard Partridge)

SeedTracker – Track Reconstruction Algorithm
(Cosmin Deaconu and Richard Partridge)

Track Finding Strategies
(drives tracking algorithm)

Multiple Scattering
(largest error for most hits)

HelicalTrackFit
(fast helix fitter)

Kalman Fitter
(Rob Kutchke)

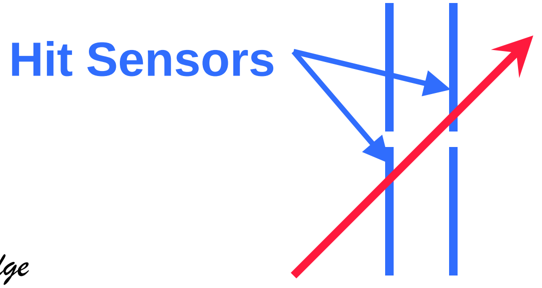
Weight Matrix Fitter
(Nick Sinev)

Hit Digitization

- Hit digitization turns GEANT energy deposits into tracker hits
- Several digitization packages are available:
  - Virtual segmentation divides cylinders / disks into virtual sensors allowing different segmentation schemes to be compared without re-running GEANT
  - Strip digitization provides detailed simulation of charge collection, digitization, and clustering of strips
  - Pixel digitization provides similar functionality as the strip digitization, and includes the ability to derive charge collection from pixel field maps
  - Pixel smearing allows Gaussian smearing of pixel hits – mostly used before the pixel digitization code was integrated into tracking
- Track reconstruction code works with either virtual segmentation or full digitization packages

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

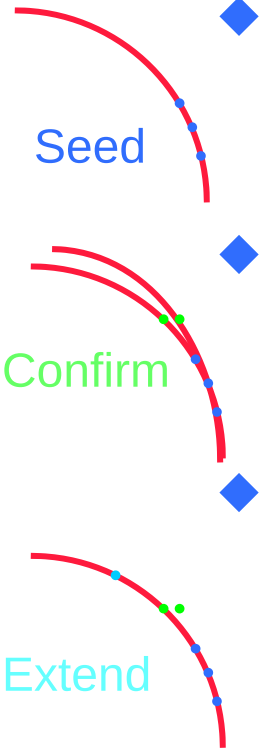


Common Hit Infrastructure provides robust handling of stereo hits, including cases like the one show here

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
  - High efficiency and robust track finding achieved by brute force

SeedTracker Algorithm

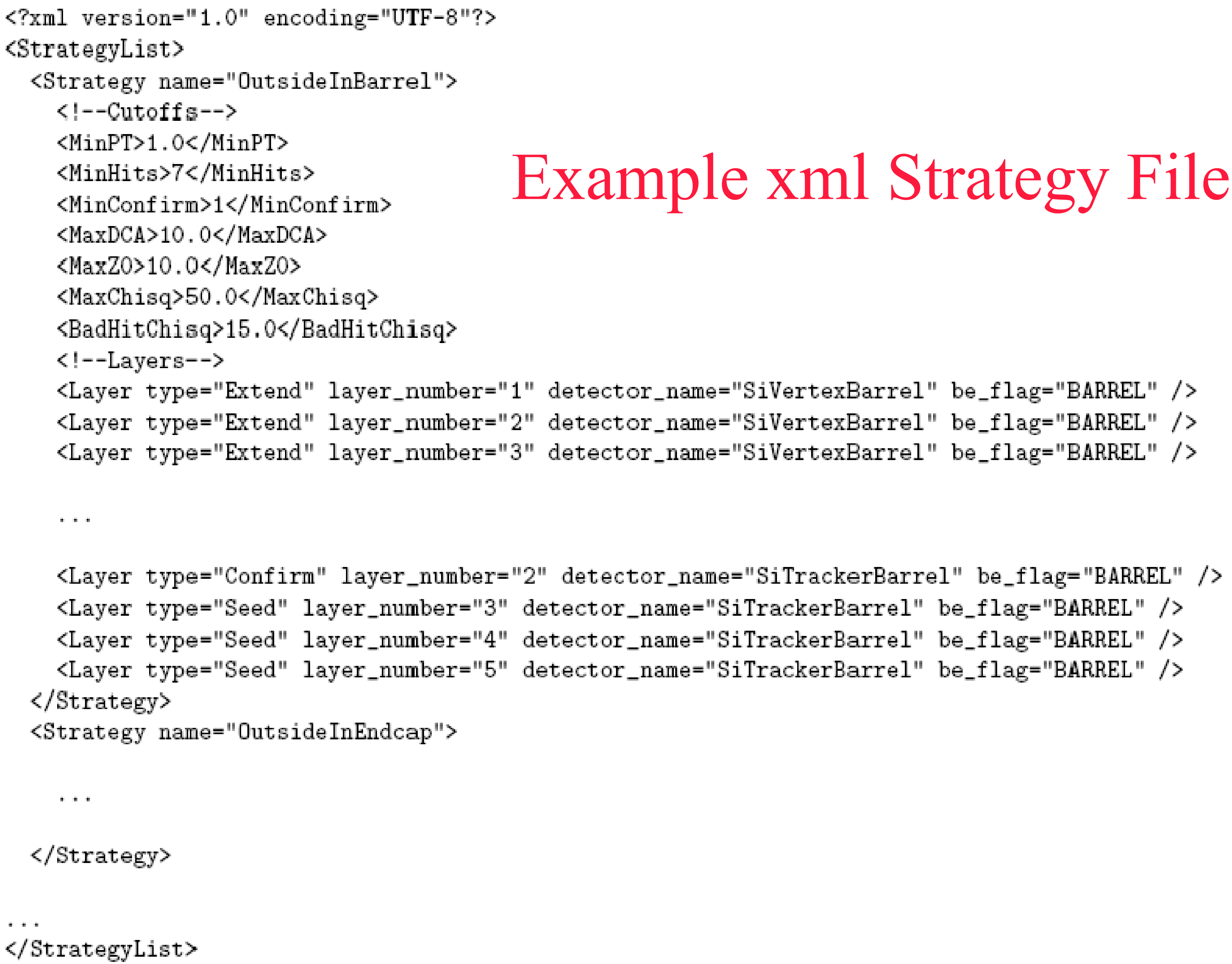


- Track finding begins by forming all possible 3 hit track seeds in the three “Seed Layers” (specified in the strategy)
  - Brute force approach to finding all possible track seeds
- Typically require the presence of a hit in a “Confirmation Layer” (specified in the strategy)
  - Significantly reduces 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

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

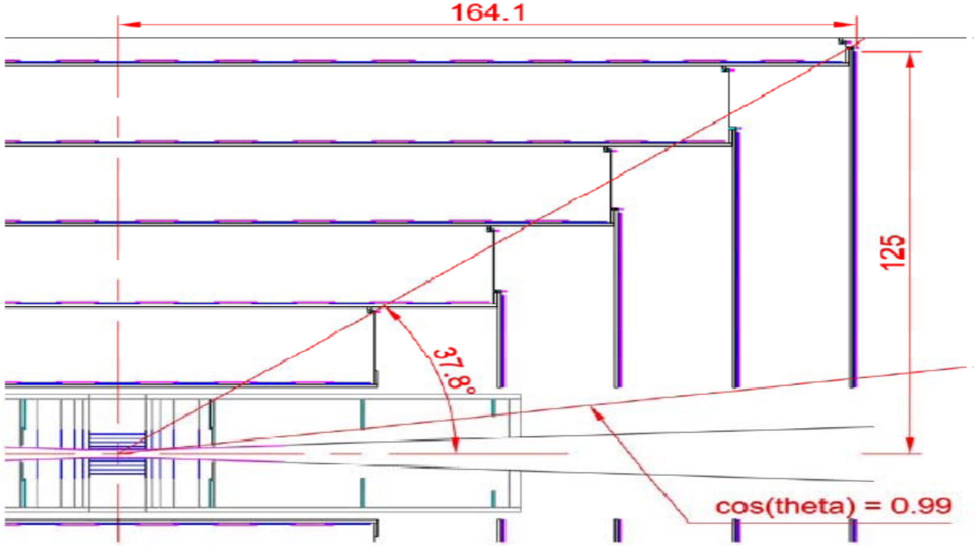
Example xml Strategy File



```
<?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

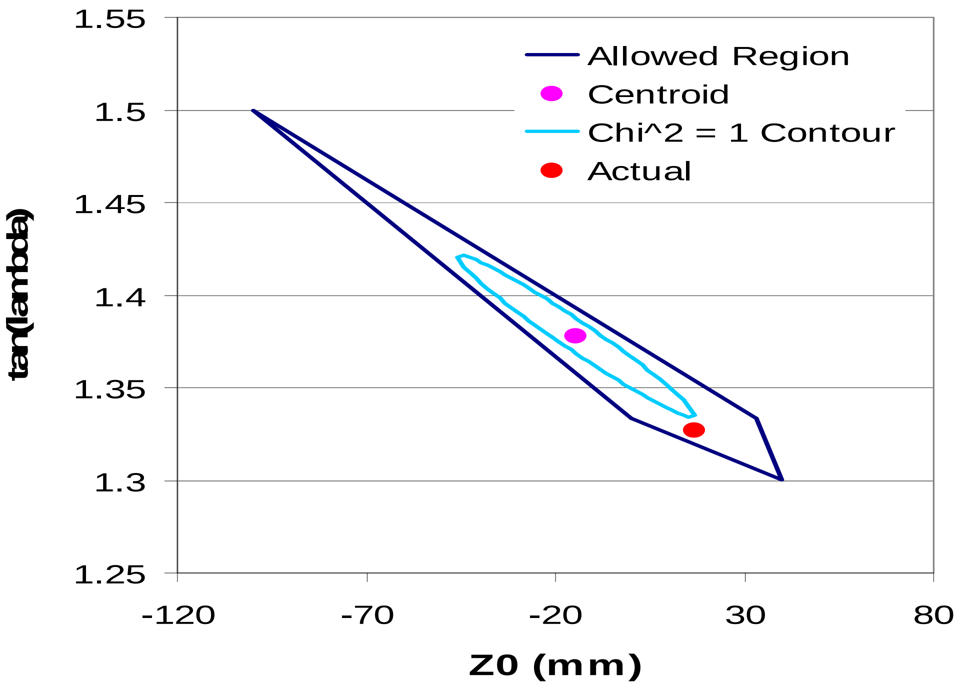
Multiple Scattering

- 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 it's 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 Kariaki algorithm to determine the track parameters $\omega \equiv 1/R$, d_0, and ϕ_0
- Determine the z_0 and $\tan(\lambda)$ track parameters: $z = z_0 + s \tan(\lambda)$
  - If there are >1 pixel hits, do a straight-line fit using only the pixel hits
  - If there are 0 pixel hits, do a ZSegment fit using all strip hits
  - If there is 1 pixel hit, treat the pixel hit as a short strip and do a ZSegment fit

The ZSegment fitter uses the observation that for ≥2 strip hits, there is a polygonal allowed region in the z_0 – $\tan(\lambda)$ plane (in some cases there is no allowed region)



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

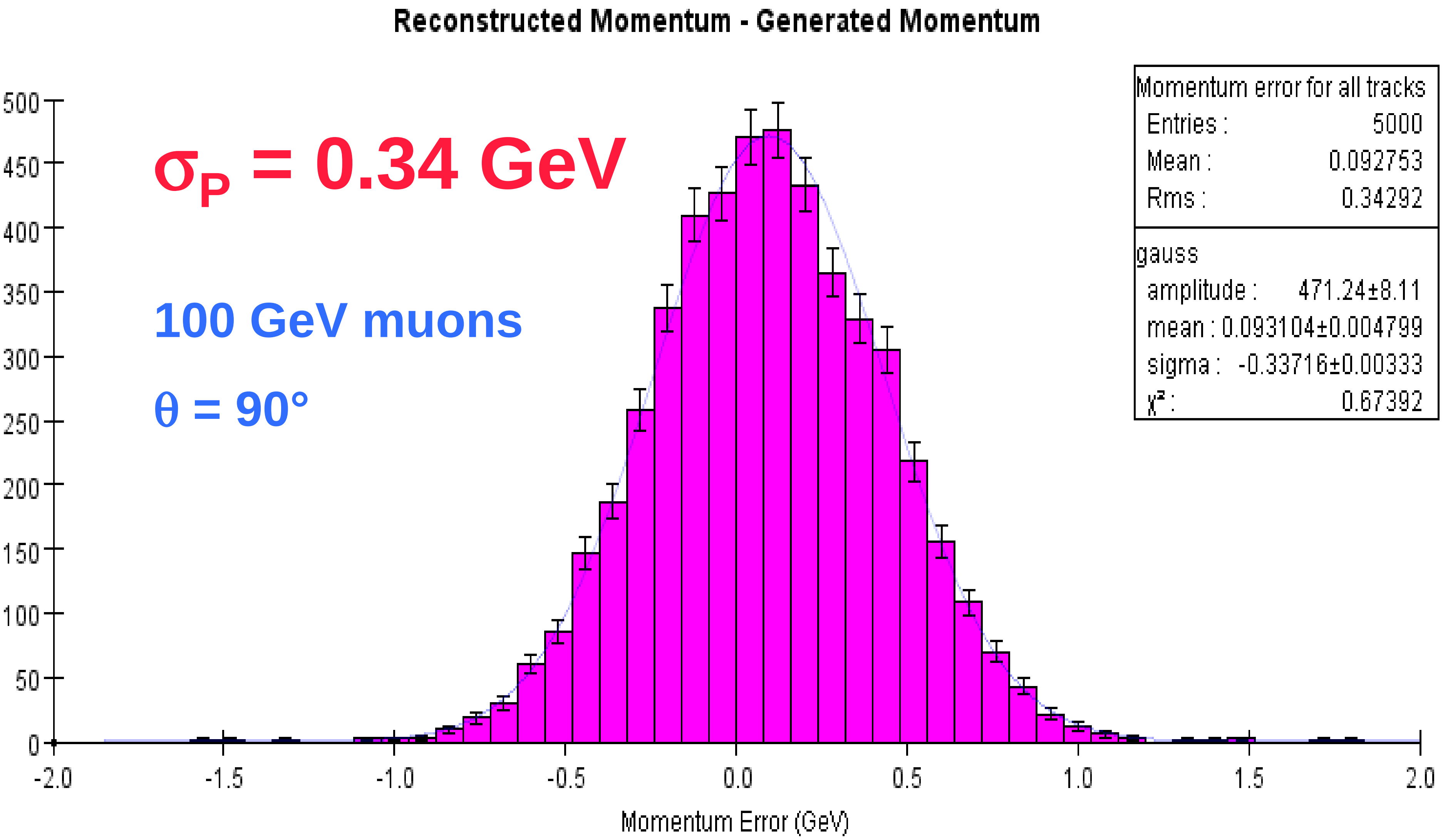
What's New Since Boulder?

- At the Boulder SiD workshop, we saw the very first SiD PFA results with full tracking
  - Some degradation in PFA resolution seen compared to cheated tracking
  - Further study indicated largest source of degradation was the $p_T > 1$ GeV constraint used in full tracking
- Since the workshop, several improvements have been made
  - Strategies re-tuned to implement $p_T > 200$ MeV constraint
  - Moved to SiD02 detector (far forward tracking disks are now pixels)
  - Finalized virtual segmentation issues (stereo angle, segmentation, pixel size)
  - A few bugs that gave occasional track finding errors were fixed
  - Substantial improvement in PFA results compared to Boulder
  - New LCRelations persist correspondence between tracker hits and MC particles
- A standard track reconstruction driver has been developed
  - org.lcsim.recon.tracking.seedtracker.ReconTracking.SiD02ReconTrackingDriver
  - Reconstruction using this driver is well underway for the LOI benchmarking samples

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
  - Hit digitization using old virtual segmentation package, which puts hits in center of strip / pixel associated with no smearing
  - Expect that these are “worst case” results – but are probably close to what can be achieved

Momentum Resolution

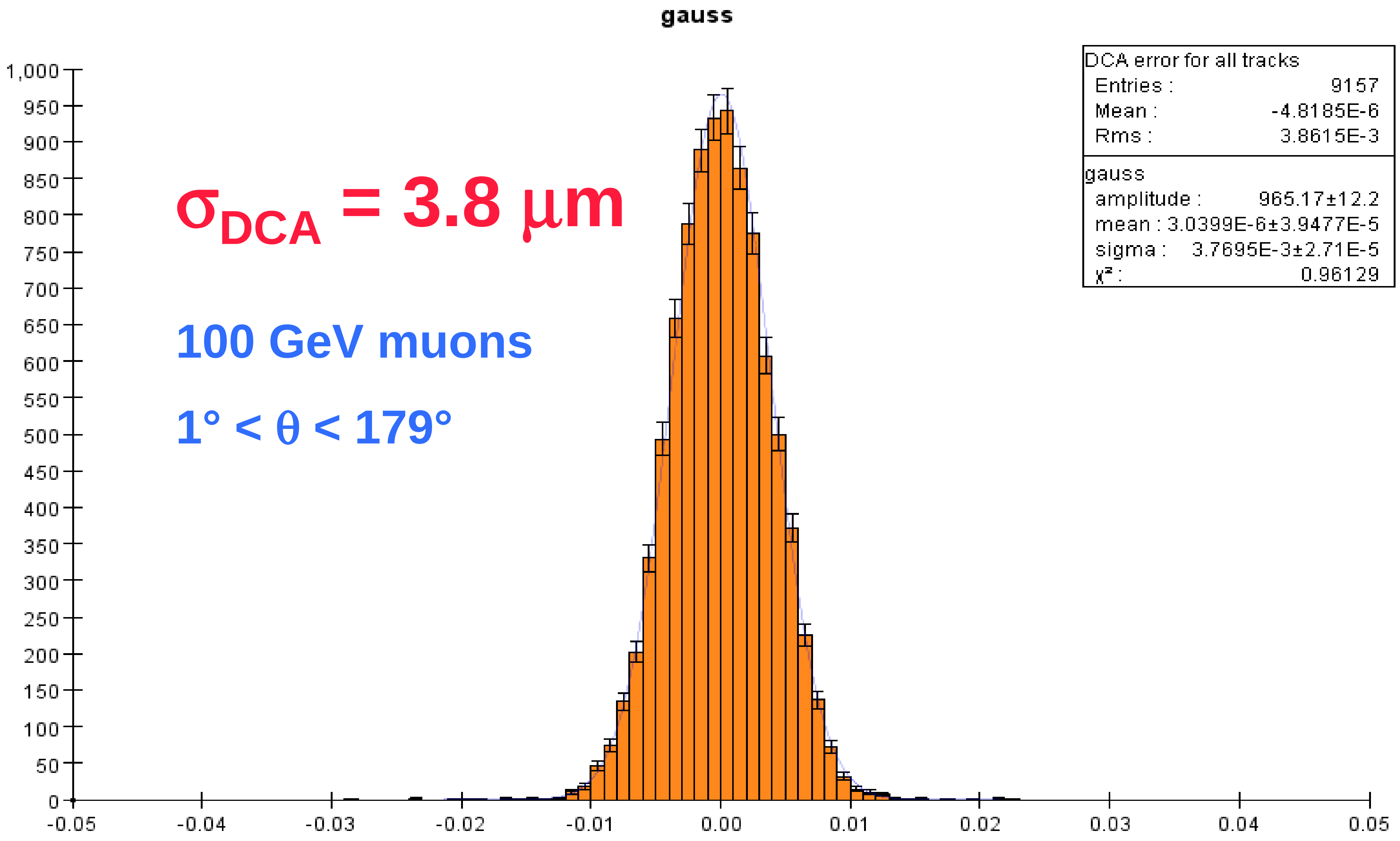


σ_P = 0.34 GeV

100 GeV muons

θ = 90°

DCA Resolution

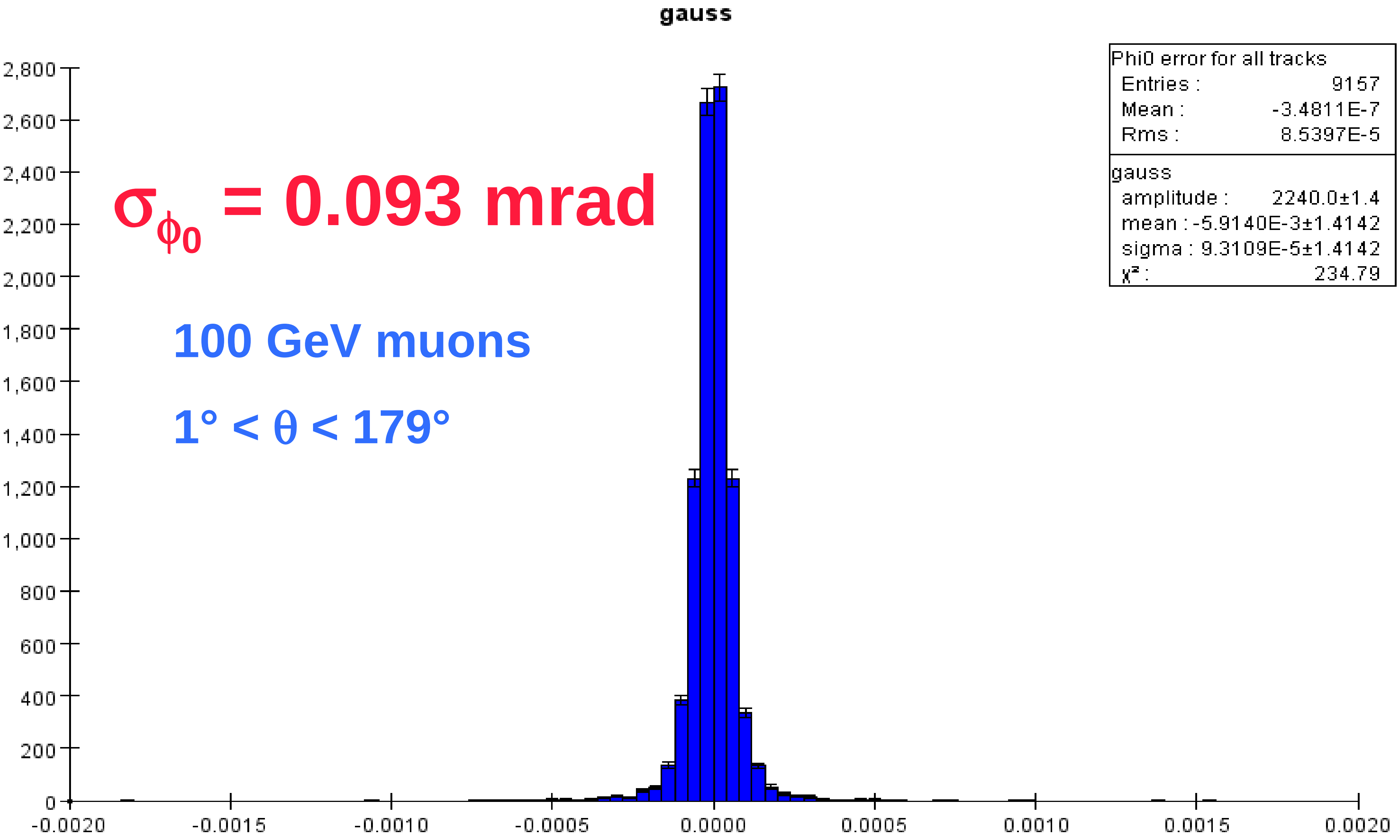


σ_{DCA} = 3.8 µm

100 GeV muons

1° < θ < 179°

ϕ_0 Resolution

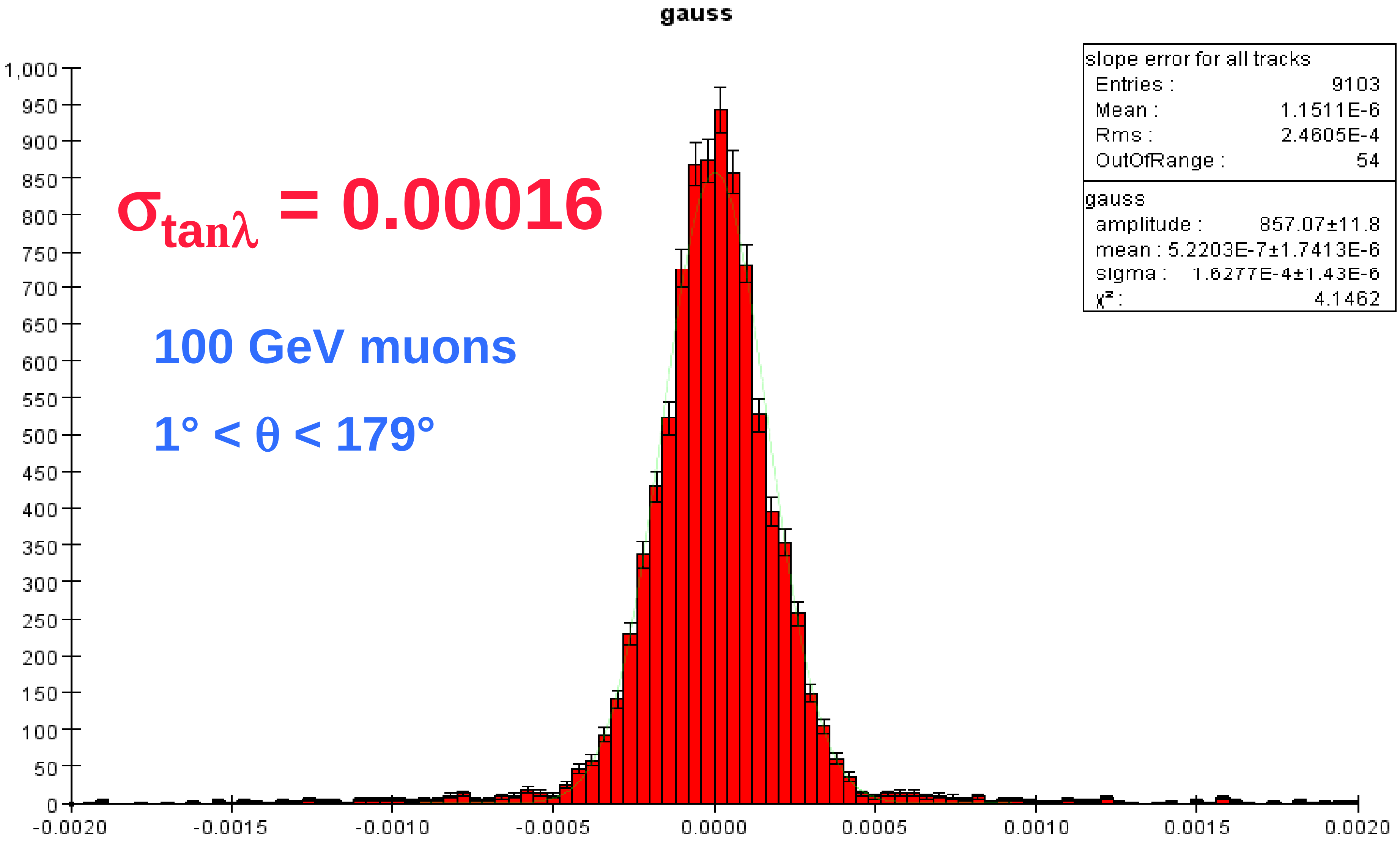


σ_{ϕ_0} = 0.093 mrad

100 GeV muons

1° < θ < 179°

tan(λ) Resolution



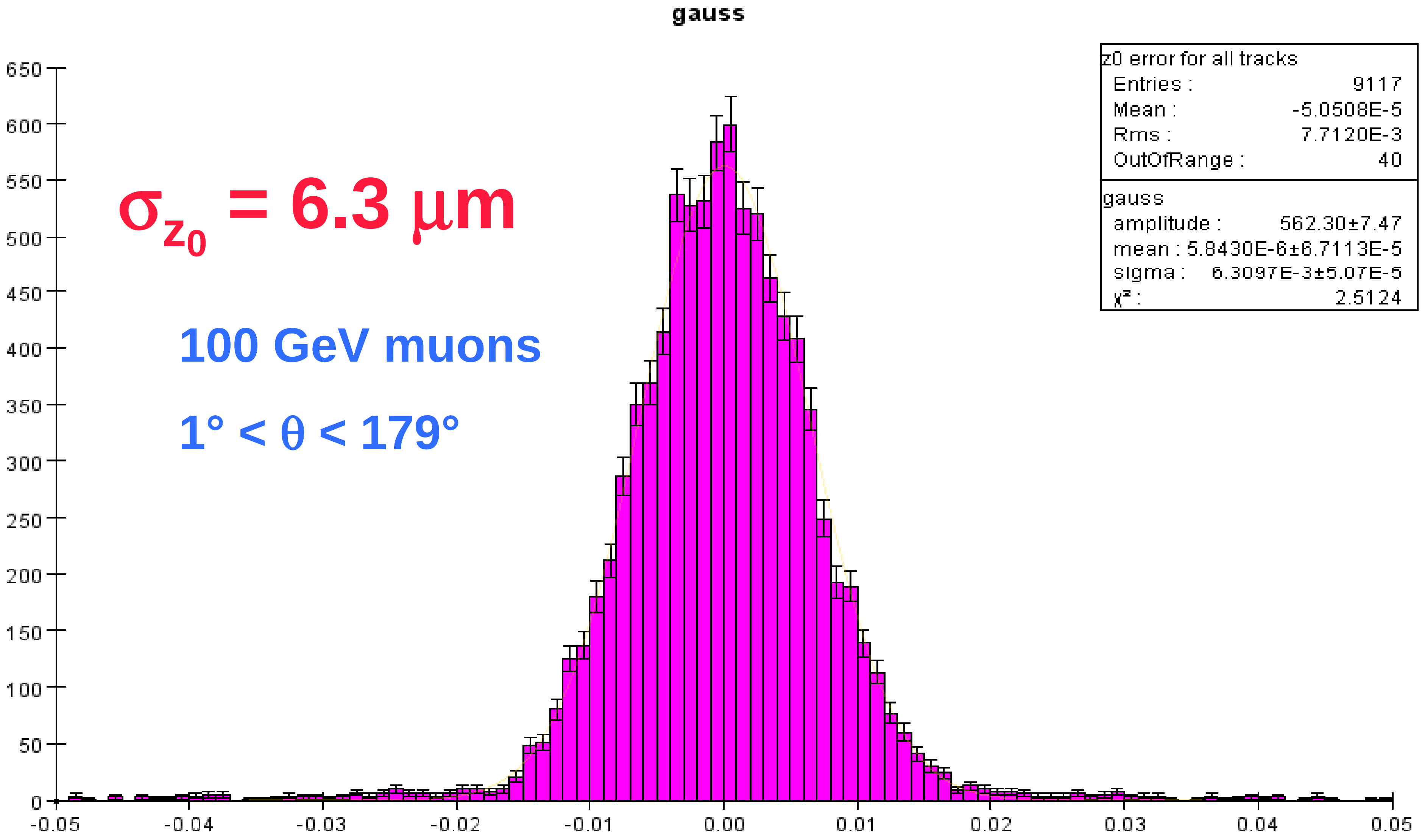
```
slope error for all tracks
 Entries :                    9103
 Mean :                  1.1511E-6
 Rms :                   2.4605E-4
 OutOfRange :                   54

gauss
 amplitude :           857.07±11.8
 mean : 5.2203E-7±1.7413E-6
 sigma :     1.6277E-4±1.43E-6
 χ² :                       4.1462
```

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

100 GeV muons

1° < θ < 179°

z_0 Resolution



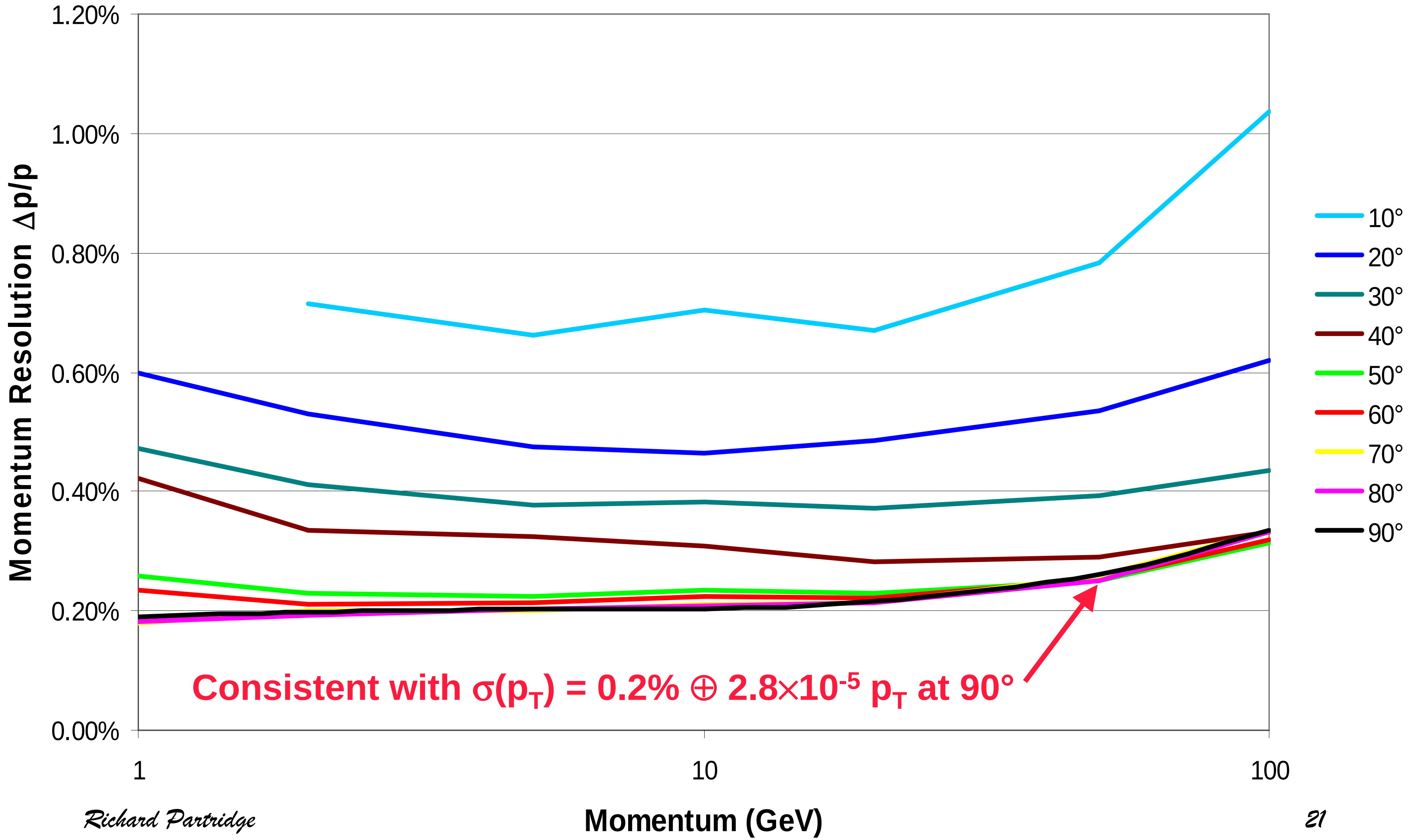
$\sigma_{z_0} = 6.3\ \mu m$

100 GeV muons

$1° < \theta < 179°$

Momentum Resolution

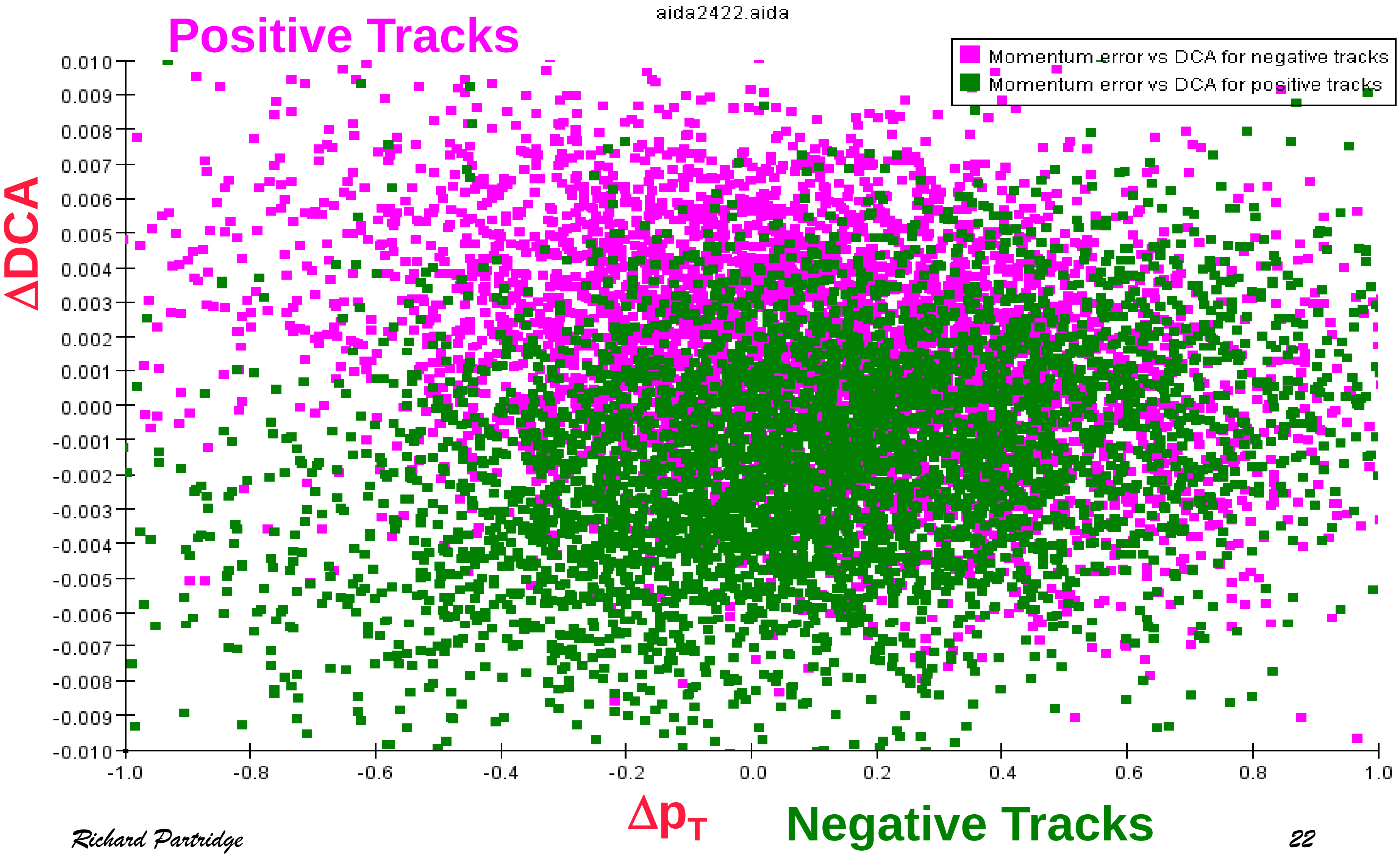
- Good momentum resolution everywhere!



Consistent with $\sigma(p_T) = 0.2\% \oplus 2.8\times10^{-5}\ p_T$ at 90°

Still Need a True Helix Fitter

- Some offsets, charge asymmetries seen



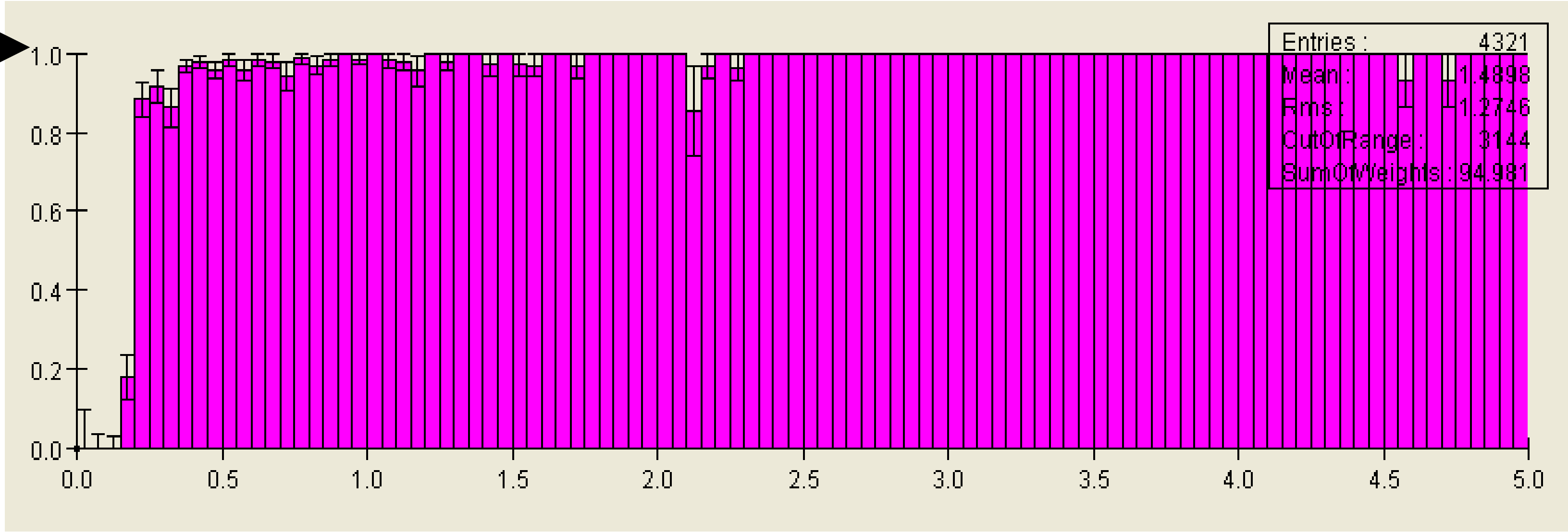
Pull Summary Table

- RMS of pull distributions are consistently >1
- Track fit errors are underestimated by a factor of ~2
- Not all that bad given the approximations in the fast helix fitter
  - No correlations in multiple scattering errors
  - No propagation of circle fit uncertainties into s – z line fits (or vice versa)
  - Approximations made to include disks in s – z line fits
  - No energy loss corrections

Helix Parameter	RMS Pull
ω	2.07
δ	1.70
φ_0	2.15
$\tan(\lambda)$	1.80
z_0	2.32

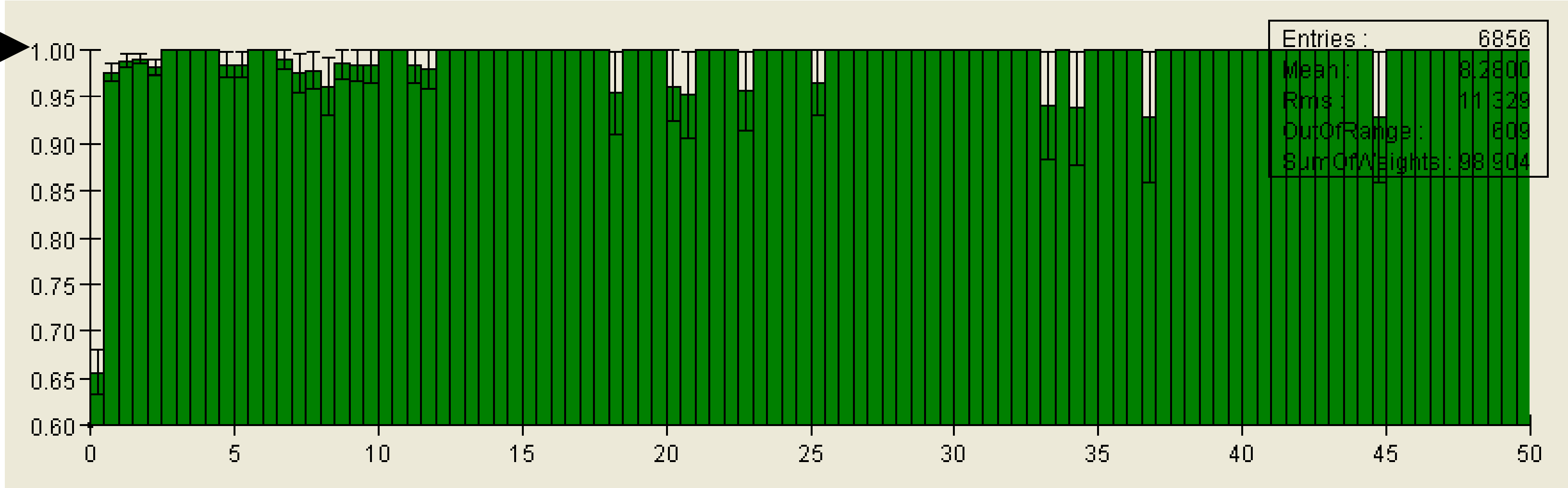
Efficiency vs p_T

100% →



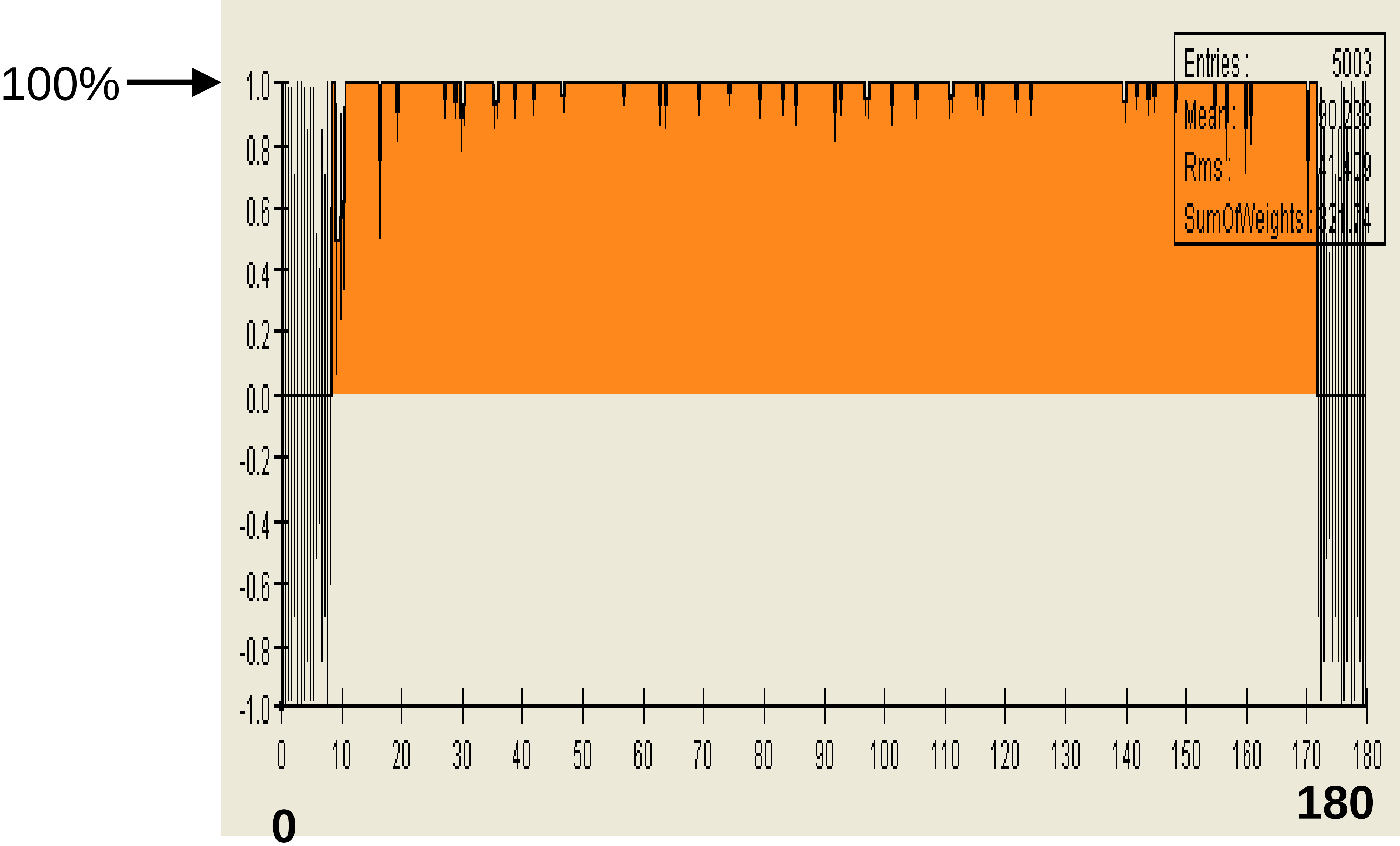
0 5

100% →

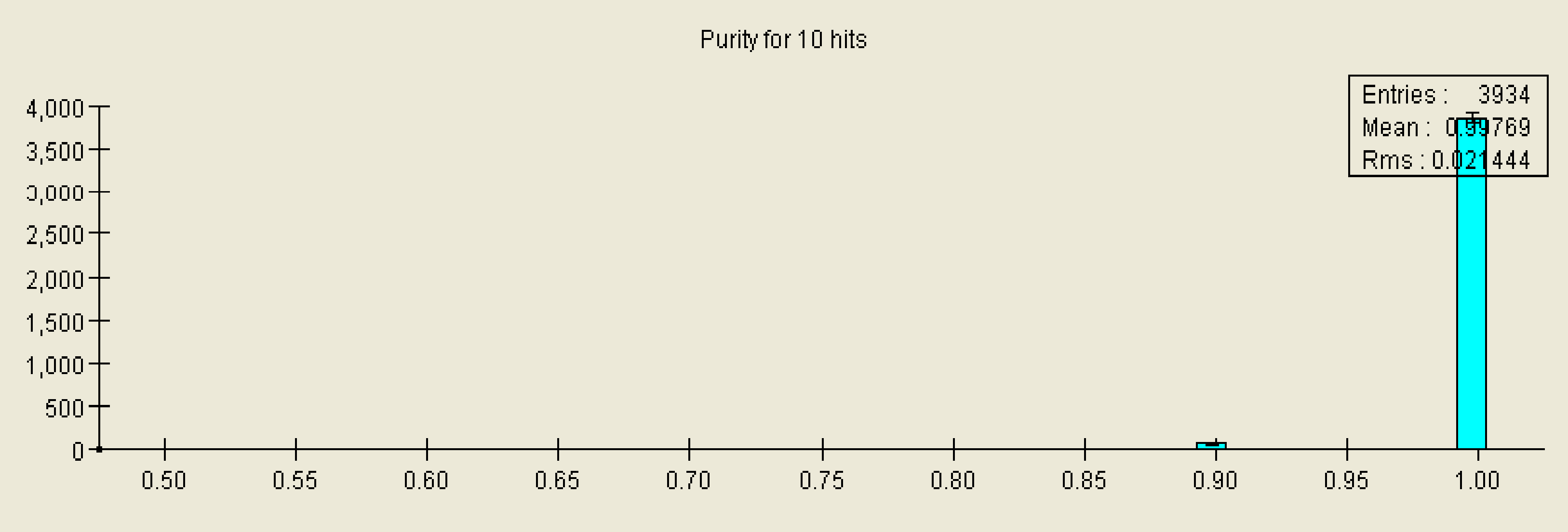


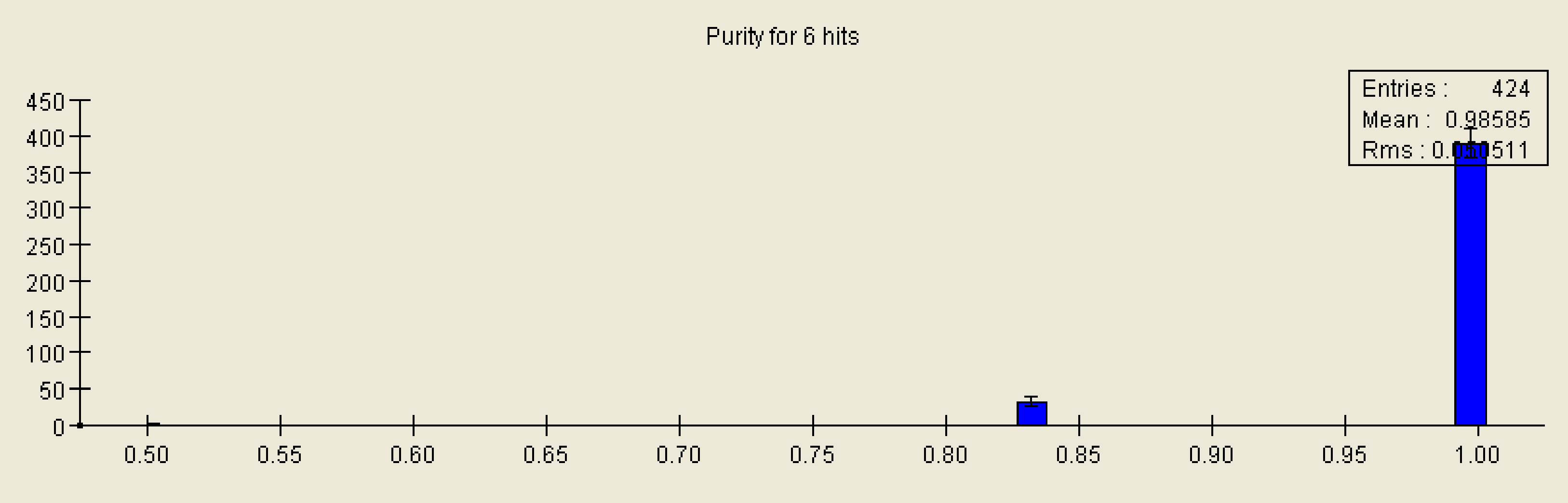
0 50

Efficiency vs θ



Purity


Purity for 10 hits
Entries : 3934
Mean : 0.99769
Rms : 0.021444
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
1.00


Purity for 6 hits
Entries : 424
Mean : 0.98585
Rms : 0.0[illegible]511
450
400
350
300
250
200
150
100
50
0
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
1.00

Summary and Future Work

- Track reconstruction algorithm seems to be working as advertised
    - Good resolution over full solid angle
    - Full coverage with high efficiency for findable tracks
    - Tracks have high purity with very small fake rate
- Track reconstruction is running as part of the reconstruction program running on the LOI benchmarking samples
- A number of areas for future work
    - Implement new virtual segmentation package with smeared hits
    - Integrate with calorimeter assisted tracking to pick up secondaries
    - True helix fits using Kalman and weight matrix fitters
    - Study effect of machine backgrounds on tracking performance
    - Apply tracking code to the task of detector optimization