Tracking with SODTracker

Fred Blanc and Steve Wagner

University of Colorado

SiD Tracking meeting

March 17th, 2006

➢ SODTracker: Introduction

➢ Performance studies: efficiency and resolution

➢ Code availability

SODTracker

- Track finding and fitting in the Silicon Outer Detector (SOD)

- Method:

  1. Find seed tracks in the Vertex Detector

     - seed tracks from hit combinations in Vertex Detector

     - seed tracks from MCParticle true information (i.e. cheat track)

  2. Extend seed tracks by adding hits in Tracker

     - currently works in barrel only

     - option to add random noise for efficiency studies

  3. Fit tracks

     - circle fitter

     - Kalman filter fitter (not available yet)

  4. Insert SODTrack (implementation of Track) in event

Performance studies

A) Track finding **efficiency**

B) Circle fit **resolution** (and bias)

- Input data:

  - **single-track Monte-Carlo**

    - 2GeV, 10GeV, 20GeV ($\theta=90^\circ$) pions

      pi_Theta90_2GeV_SLIC_v1r9p3_sidaug05.slcio
      pi_Theta90_10GeV_SLIC_v1r9p3_sidaug05.slcio
      pi_Theta90_20GeV_SLIC_v1r9p3_sidaug05.slcio

    - 1-10GeV, $\theta = 4$-176° pions

      pions_Theta4-176_1-10GeV_SLIC_v1r9p3_sidaug05.slcio

  - **physics events**

    - $e^+e^- \rightarrow ZZ$ *panpyZZ-16-500_SLIC_v1r9p3_sidaug05.slcio*

    - $e^+e^- \rightarrow Zh$ *panpyZh120-0-500_SLIC_v1r9p3_sidaug05.slcio*

Track finding efficiency

- Run on single track MC

  – $\theta=90^o$ (barrel only)

  – 2GeV, 10GeV, and 20GeV tracks

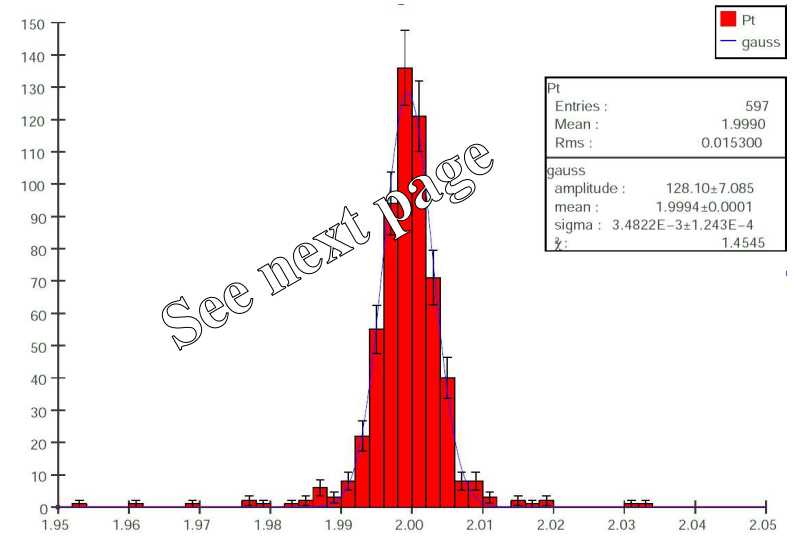- Reject events in which the track has interacted with the detector material

- Efficiency = **99.2% at 2GeV**
  99.9% at 10GeV
  99.6% at 20GeV

- no visible reduction in efficiency when noise is added (>100 hits/layer)

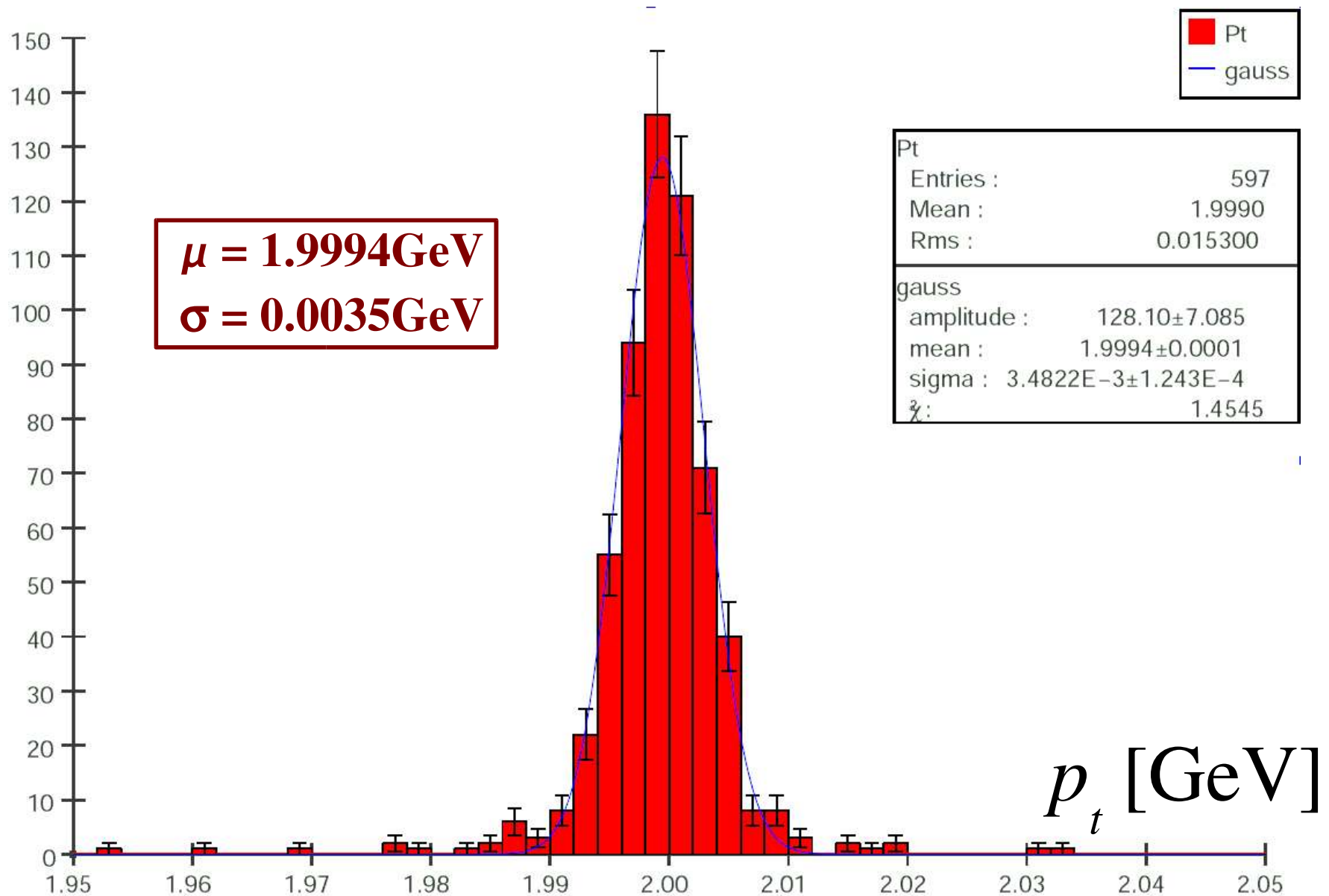**SOD track finding efficiency (single track)**
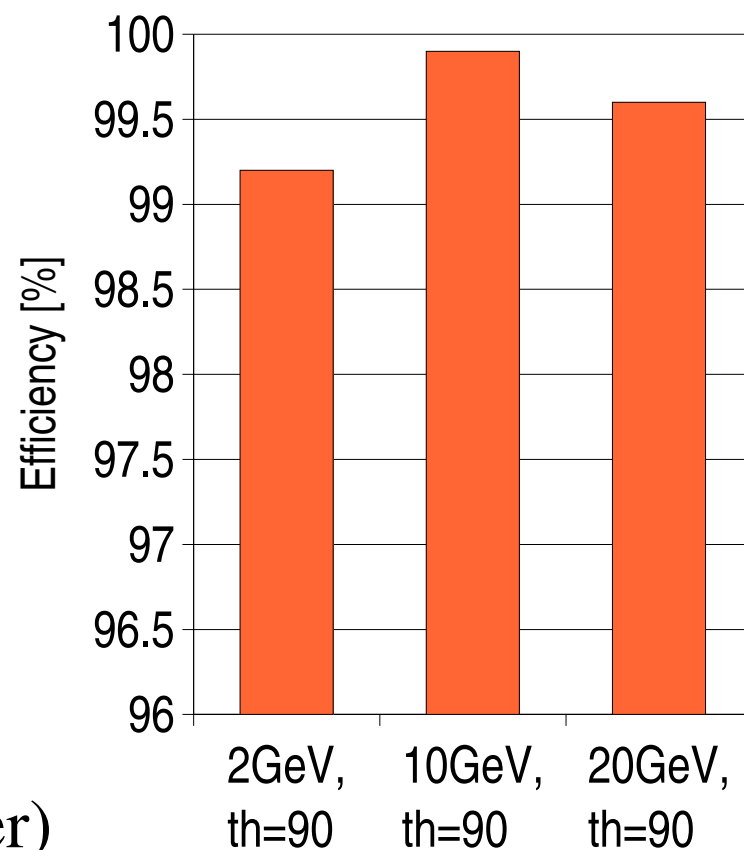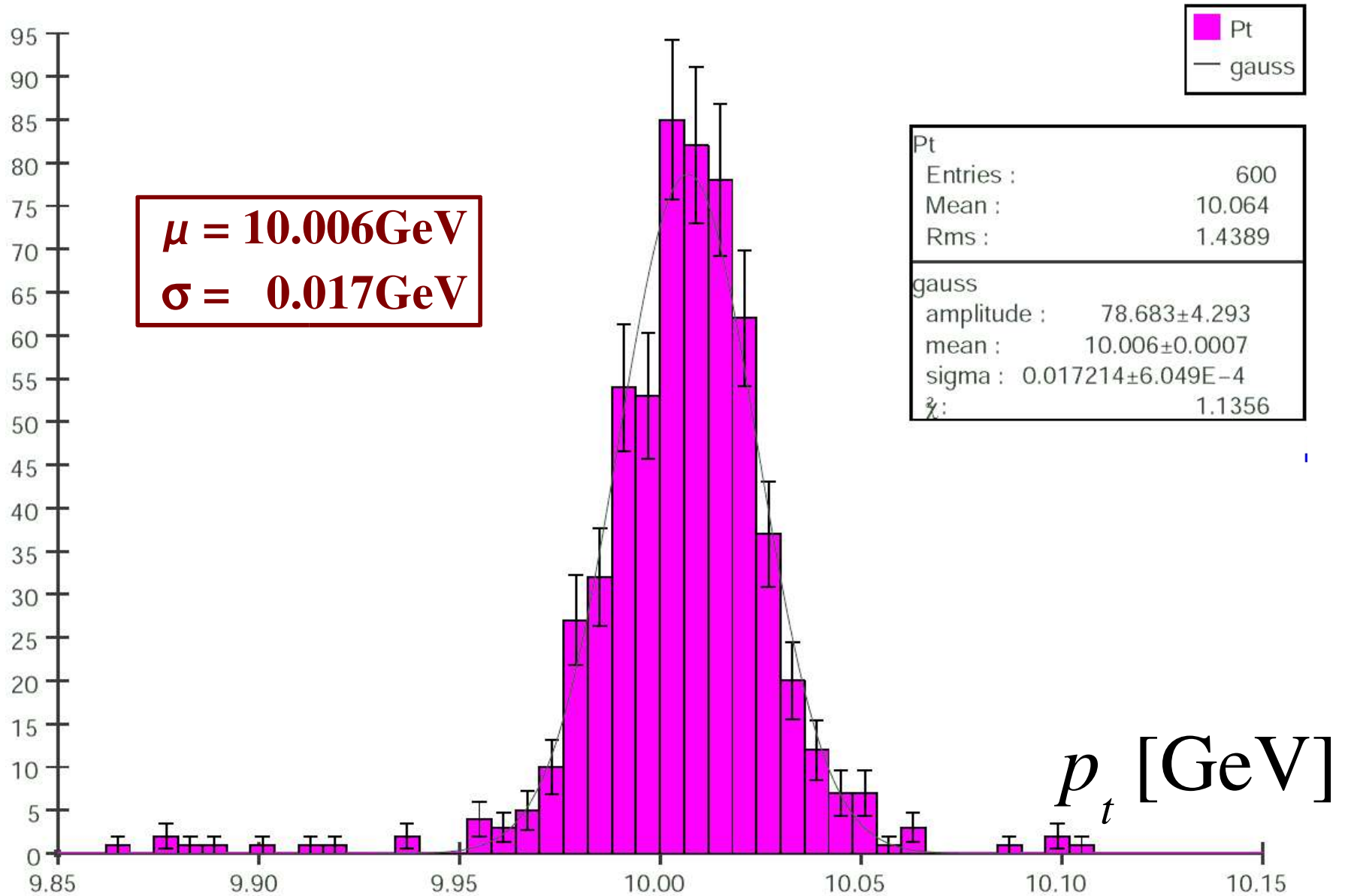
Momentum resolution

- ### Run on single track MC

  - $\theta=90^{o}$ (barrel only)

  - 2GeV, 10GeV, and 20GeV tracks

- ### Apply circle fit

- Plot fitted momentum

- Gaussian fit to distribution (see next pages)

  ➔ good fit and good χ^2 in all 3 samples

  ➔ $\Delta p/p = 0.17\%$ => $\sigma_p(p=10\text{GeV}) = 17\text{MeV}$

  ➔ small bias relative to generated momentum (~0.07%@20GeV)

    => circle fit is already a good approximation!
    => Kalman filter fit should improve further

Momentum resolution (2GeV pions, θ=90º)

$\mu = 1.9994\text{GeV}$

$\sigma = 0.0035\text{GeV}$

Pt
Entries :	597
Mean :	1.9990
Rms :	0.015300

gauss
amplitude :	128.10±7.085
mean :	1.9994±0.0001
sigma :	3.4822E−3±1.243E−4
χ^2 :	1.4545

p_t [GeV]

Momentum resolution (10GeV pions, θ=90°)

μ = 10.006GeV

σ =   0.017GeV

Legend: Pt, — gauss

Pt

Entries :	600
Mean :	10.064
Rms :	1.4389

gauss

amplitude :	78.683±4.293
mean :	10.006±0.0007
sigma :	0.017214±6.049E−4
χ^2 :	1.1356

p_t [GeV]

Momentum resolution (20GeV pions, θ=90°)

$\mu = 20.014\text{GeV}$
$\sigma = 0.033\text{GeV}$

p_t [GeV]

Pt ■
gauss —

Pt	
Entries :	698
Mean :	17.459
Rms :	7.2613

| gauss	
amplitude :	67.113±3.637
mean :	20.014±0.001
sigma :	0.033297±0.001147
χ^2 :	1.0347
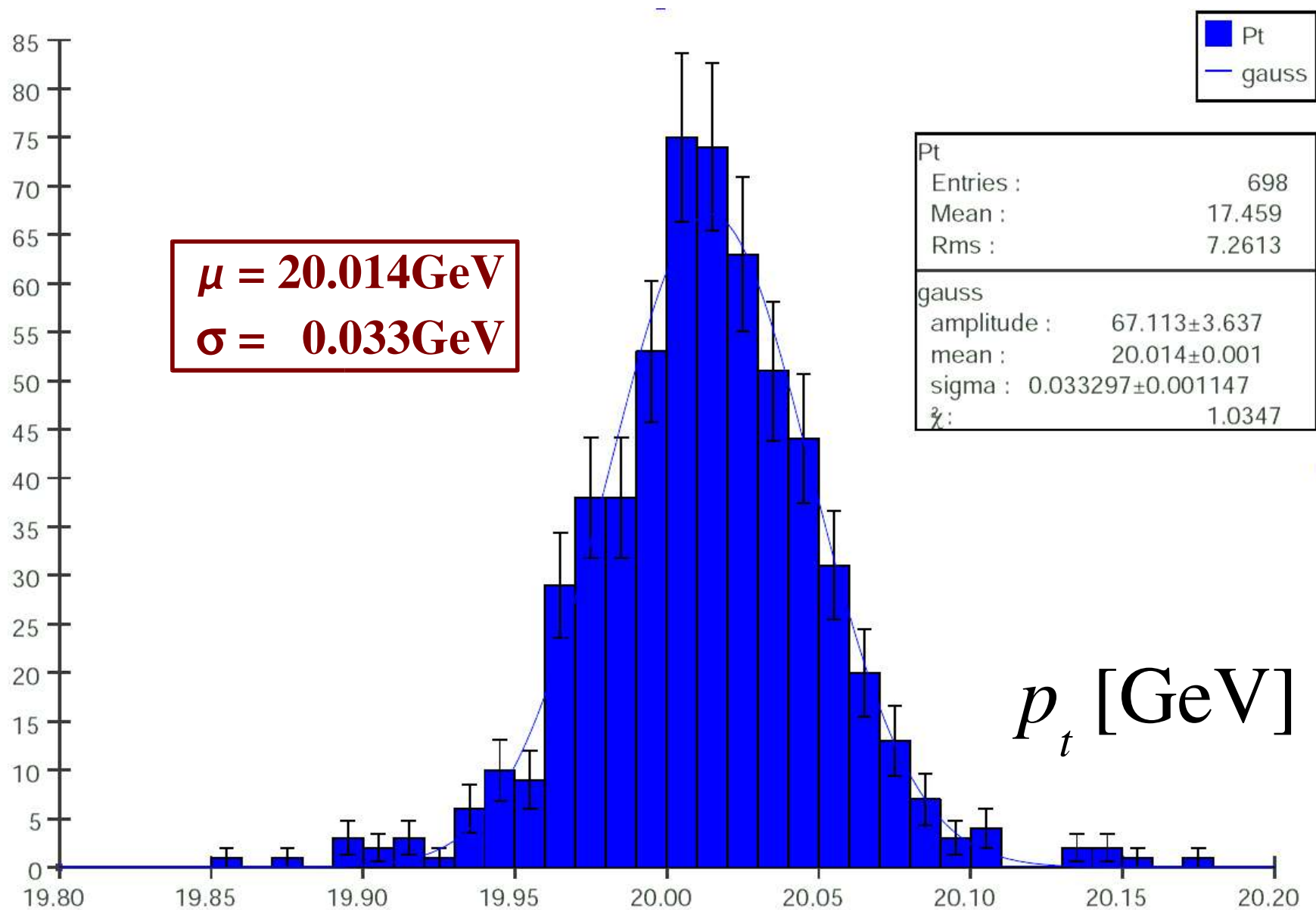
SODTracker package

- Code ported from hep.lcd to org.lcsim (Steve Wagner)

- SODTrack output inserted in event
  (SODTrack implements *org.lcsim.event.Track*)

- Created package SODTracker

- Committed SODTracker to CVS (in org/lcsim/contrib/)
     => available for all to use

- Package also contains test driver in test/TestSOD.java

- Tested on single track and physics samples
     => no crash (running on single-track and physics events)
     => visualize SODTracks on event display
          (and in event browser)

Event Display from $e^+e^- \rightarrow Zh$ MC
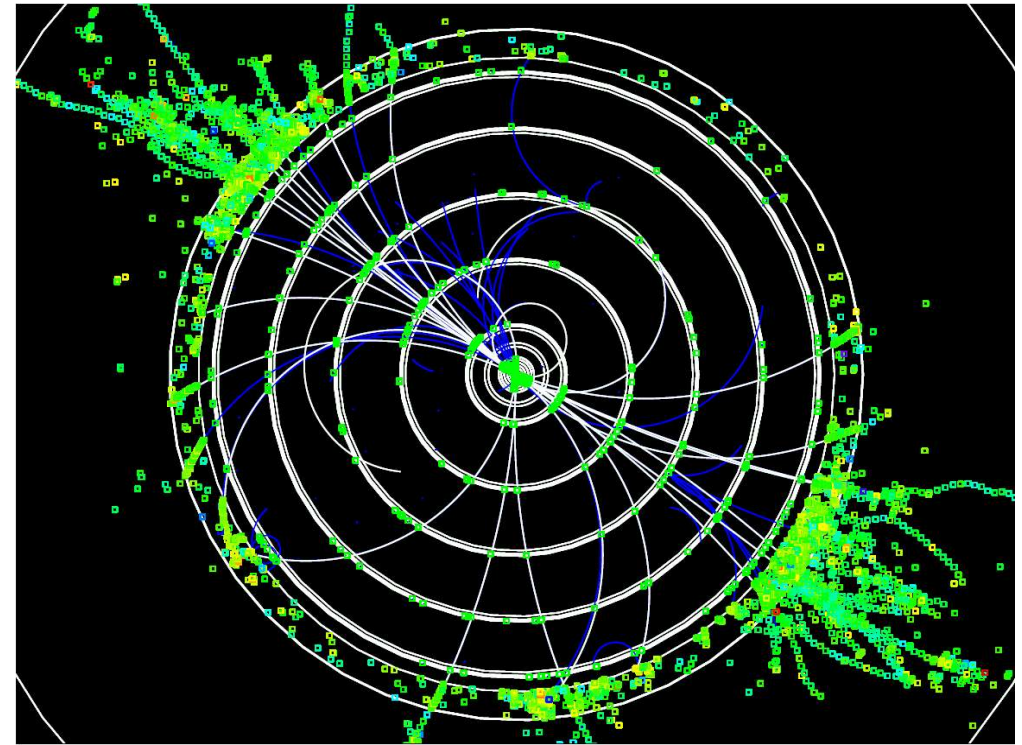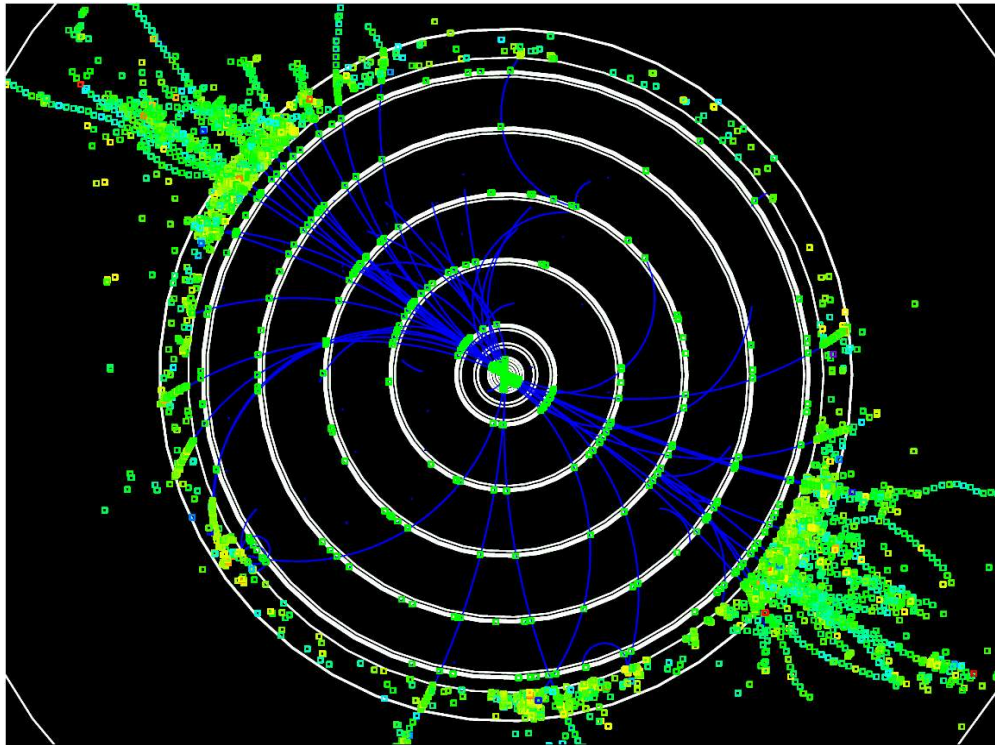
- Display only tracks in barrel
- Most barrel tracks are reconstructed
- Few missing tracks probably at larger azimuthal angle

Summary

- New SODTracker package:

  - track finding and fitting in the Tracker

  - high efficiency (>99% on single-track MC)

  - good resolution, and very small bias (circle fit)

  - tracks inserted in event => can be used in event reconstruction

- Welcome comments and suggestions

- Next step:

  - Extend circle fit to full helix fit with Kalman filter
    => should recover already small bias in p_t (<0.1%)