TKR Simulation and Reconstruction Survey of Current Projects

Simulation / Digitization

TkrReconTestSuite

Kalman Filter Vertexing

Comparing Recon to Monte Carlo

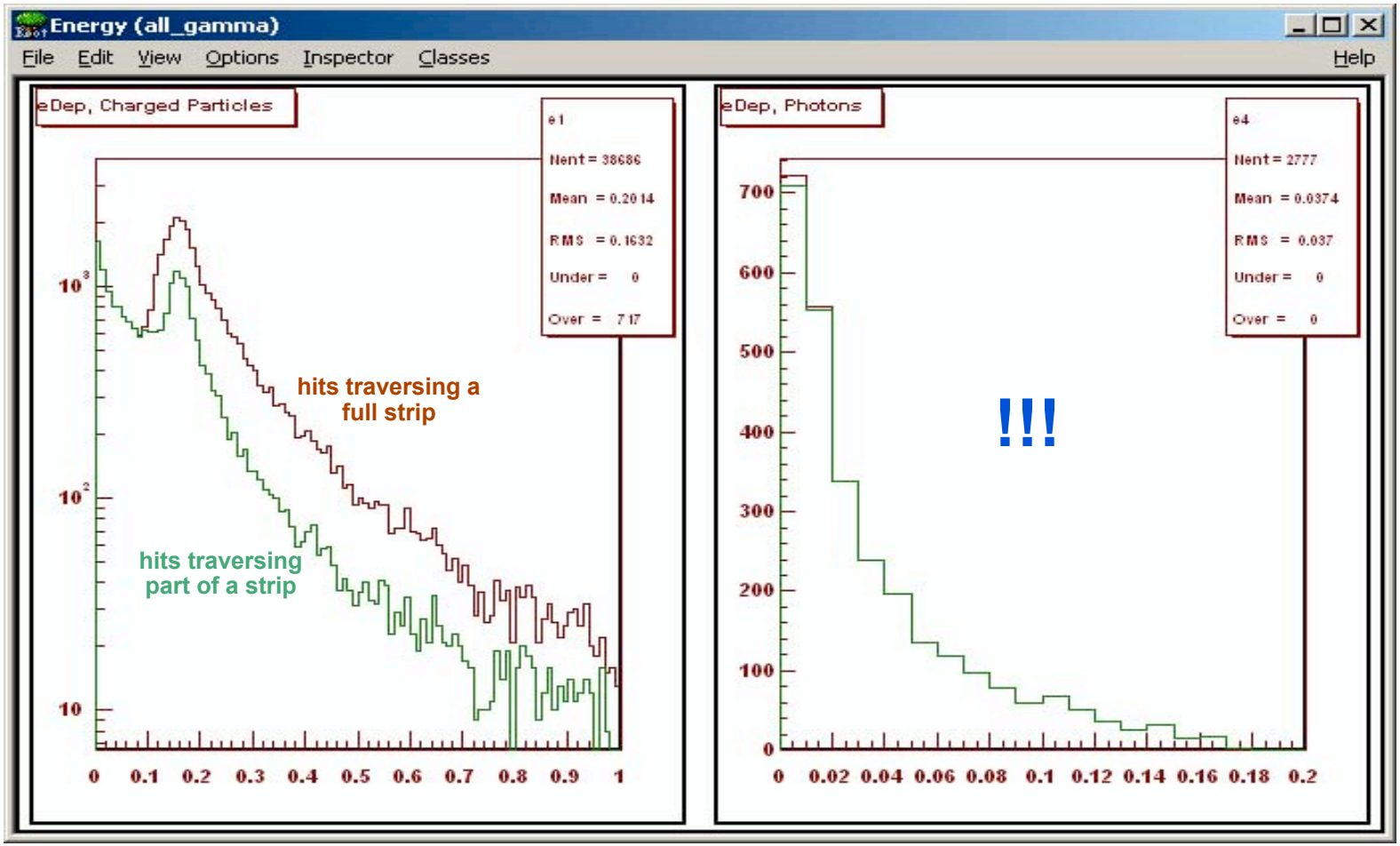
TkrRecon Algorithm Timing

The Crystal Ball

T. Usher

Simulation And Digitization

Energy Deposit by Photons!



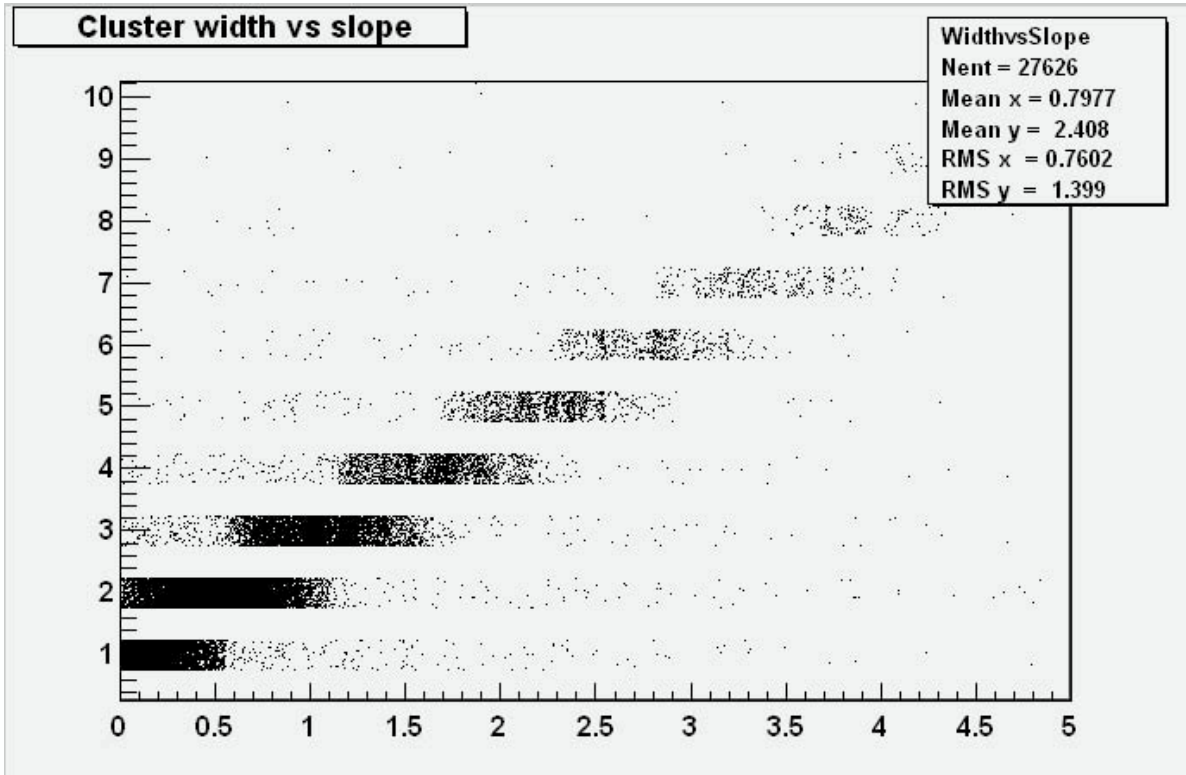
Energy from "photons" comes from Compton scatters below Geant range cutoff, and from stopping photons. We need to fix digitization to dump all this energy into one or two strips, rather than sharing across several strips. A shorter range cutoff will reduce this effect.

T. Usher

From Leon Rochester

Simulation And Digitization

Cluster Width

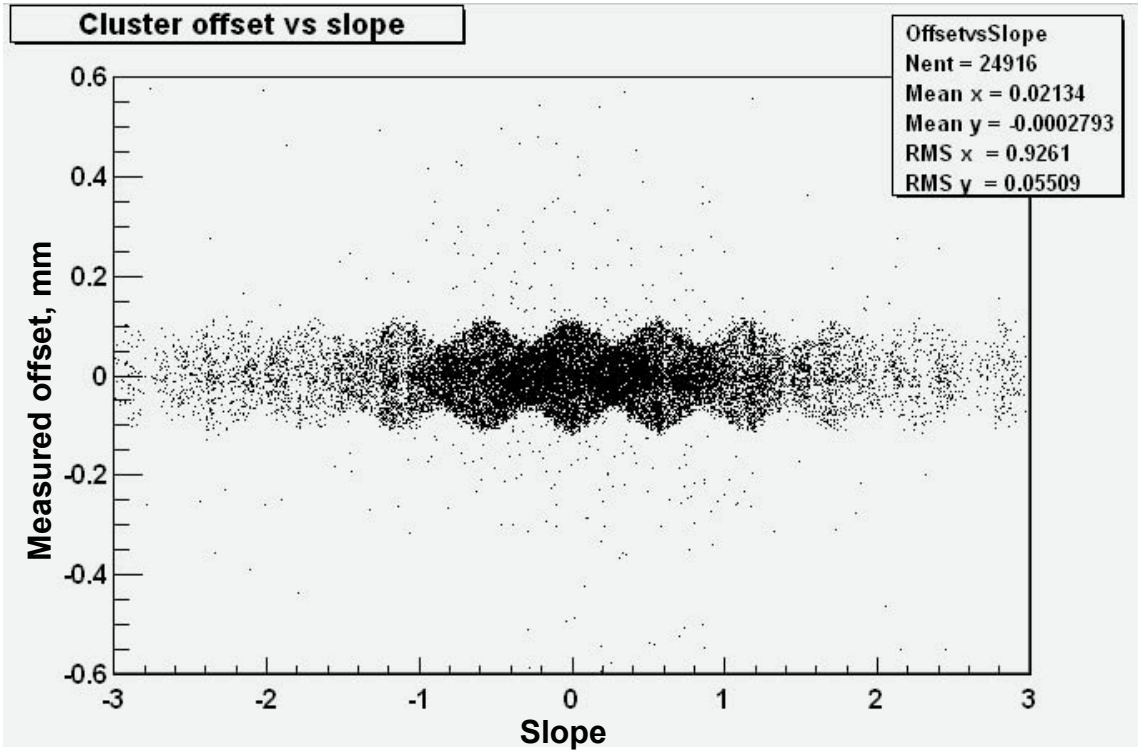


Cluster width vs. slope of track for 1-GeV muons over all angles. This plot can be compared to BTEM/BFEM data, or more directly, to EM/CU data to come.

From Leon Rochester

Simulation And Digitization

Cluster Center Measurement Error



Offset of measured cluster position from true hit position for 1-GeV muons at all angles. In principle, the average measurement error is less than 0.228/sqrt(12). There is no charge-sharing (important) or diffusion (less important) in the this digitization model.

T. Usher

From Leon Rochester

TkrReconTestSuite

- **Need:**
  - Tool for testing reconstruction algorithms
    - Want something to test the underlying logic of an algorithm
    - Test needs to be free from the complications of the "real world"
      - Multiple scattering
      - Production of secondaries
      - Etc.
    - Provides simpler environment for looking for and understanding problems
      - Are hits correctly attached to tracks
        - » e.g. crossing tower boundaries
      - Do tracks point in correct direction?
        - » e.g. track directions "pulled" to a preferred direction
      - Are vertex positions and directions correctly found
      - etc.
- **Idea:**
  - Generate tracks and associated hits independently of Monte Carlo
    - Tracks and/or hit positions specified via xml file

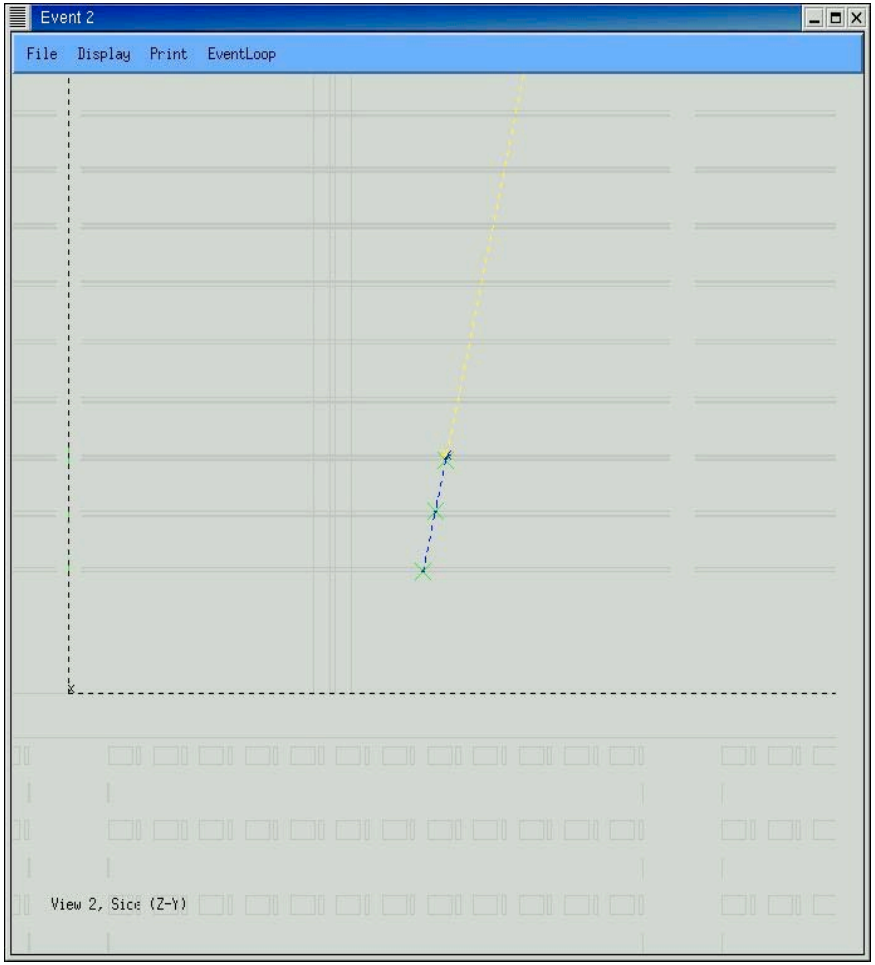
From Michael Kuss

TkrReconTestSuite

Example: Generating single hits

- **This mode generates hits at specified coordinates:**

```
<event>
  <tkrdigi  tower = "9"  xylayer =  "2"  x
   =   "10"  y =  "950" />
  <tkrdigi  tower = "9"  xylayer =  "1"  x
   =  "106"  y =  "922" />
  <tkrdigi  tower = "9"  xylayer =  "0"  x
   =  "190"  y =  "890" />
  <calcluster>
    <energy  value = "100"  units = "MeV"
   />
    <point  x = "-250"  y = "195"  z = "-
   30" />
  </calcluster>
</event>
```

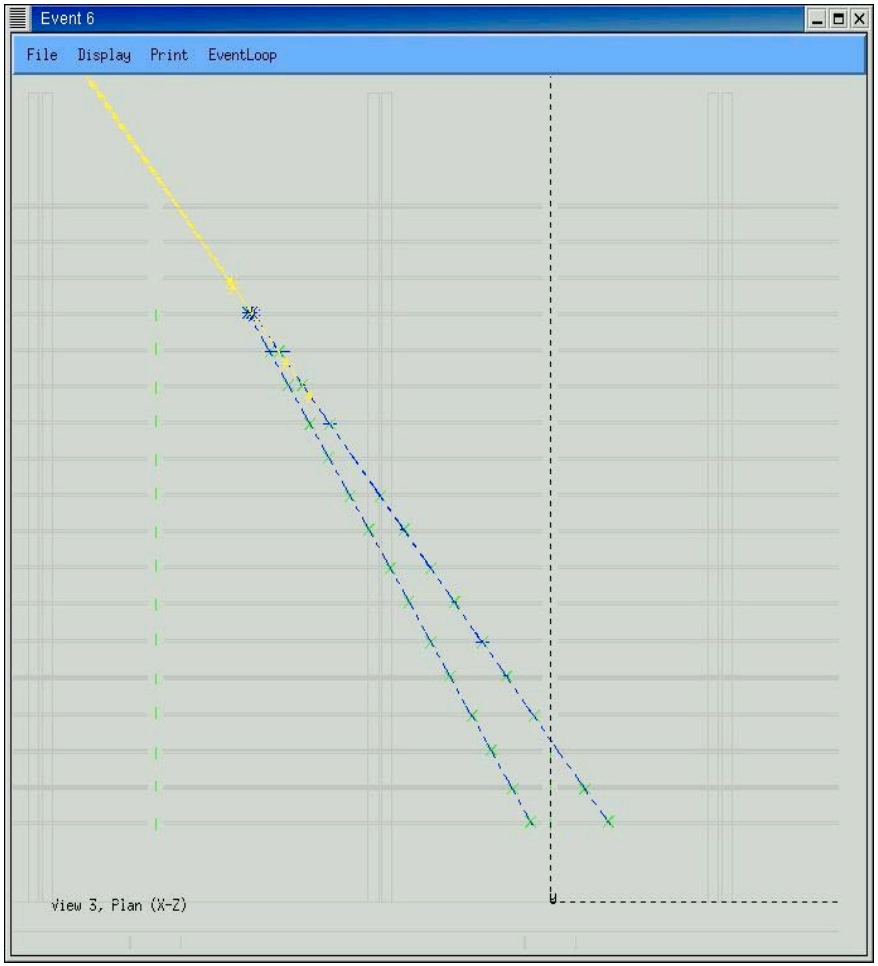


TkrReconTestSuite

Example: Generating hits by specifying tracks

- **This mode specifies tracks to generate hits:**

```
<!-- a "V" -->
<event>
  <particle>
    <energy  value = "100"  units = "MeV" />
    <point  x = "-300"  y = "100"  z = "550" />
    <angle  th = "2.5"  ph = "0.1" />
  </particle>
  <particle>
    <energy  value = "200"  units = "MeV" />
    <point  x = "-300"  y = "100"  z = "550" />
    <angle  th = "2.6"  ph = "0.2" />
  </particle>
</event>
```



- **Currently able to generate single events**
- **Now working to generate series of events**
- **Will become useful tool soon!**
  - **For code development**
  - **As part of System Tests (?)**

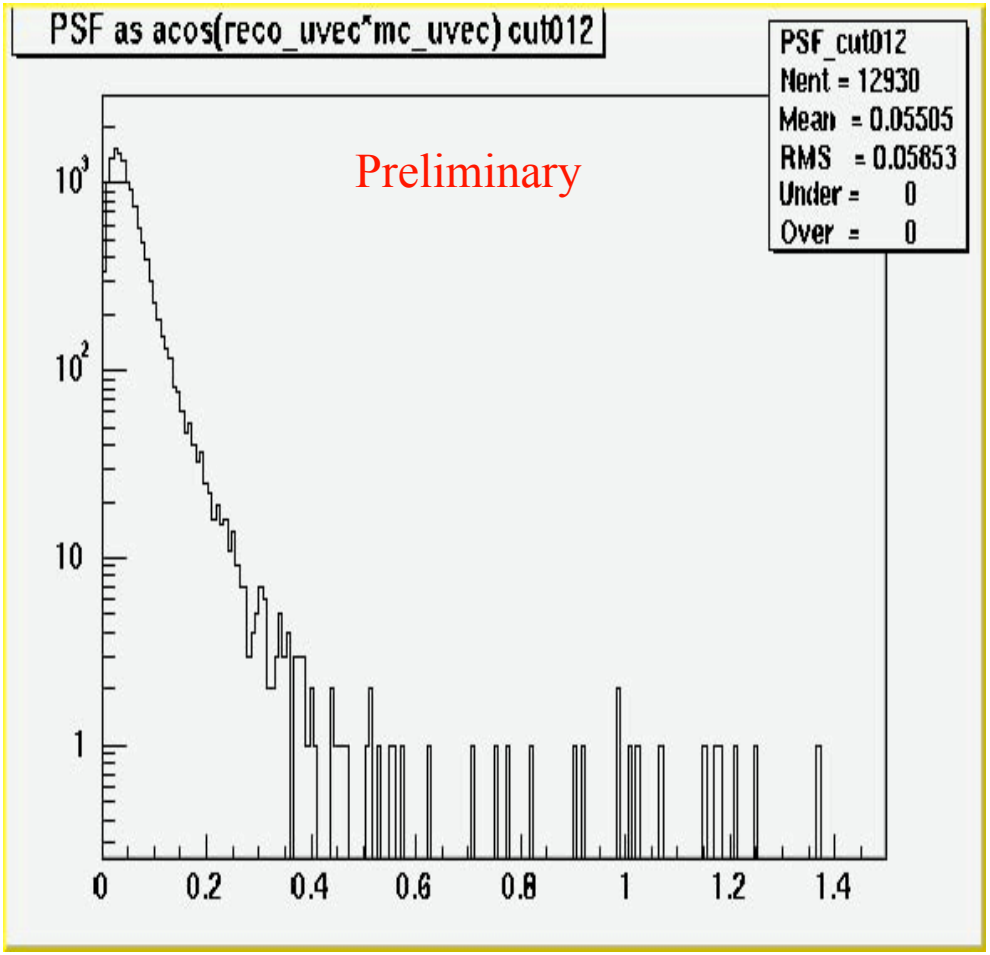
T. Usher

From Michael Kuss

Kalman Filter Vertexing

Very Preliminary Look

- **New Vertex Fit Algorithm**
  - Kalman Filter Approach
- **Example:**
  - Generated Sample:
    - Downward going gammas
  - Event selection:
    - Events with > 1 recon track
    - Vertices with 2 tracks
    - Look in Front section only
  - Preliminary results promising
    - Still much work to do before general release

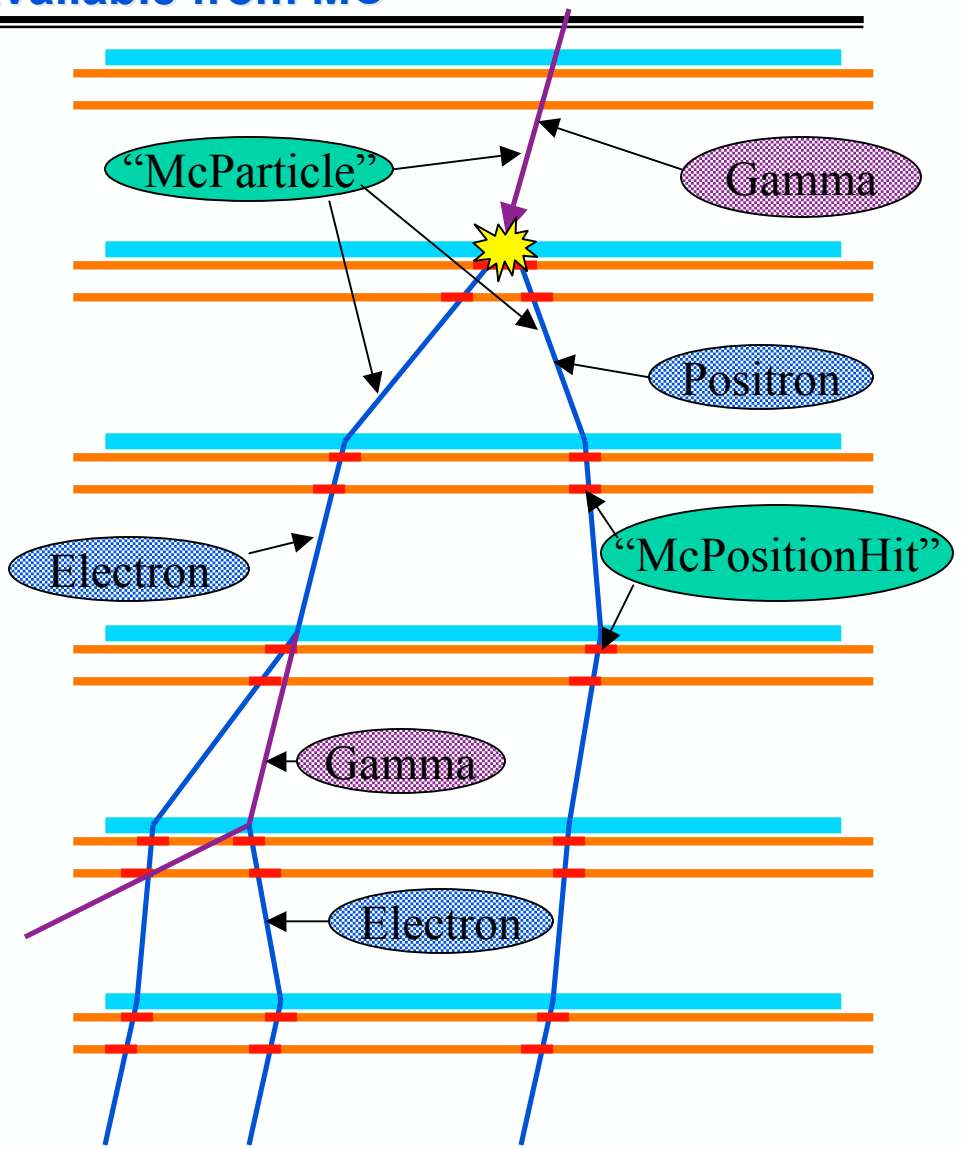


From Johann Cohen-Tanugi

Recon comparison to Monte Carlo

Information available from MC

- **McParticle**
  - One per particle produced in the event (If above Geant cutoff energy!)
  - Contains:
    - Particle type
    - Start/stop four momentum
    - Start/stop position
  - Arranged in tree structure
    - Points to its parent
    - List of pointers to daughters
- **McPositionHit**
  - One (at least) per hit in the silicon
  - Contains
    - Entry/Exit positions
    - Pointer to particle causing hit
    - Energy deposited by particle
- **McIntegratingHit**
  - Cal version of McPositionHit
- **Relational Tables**
  - Version useful here relates McPositionHits to reconstructed clusters

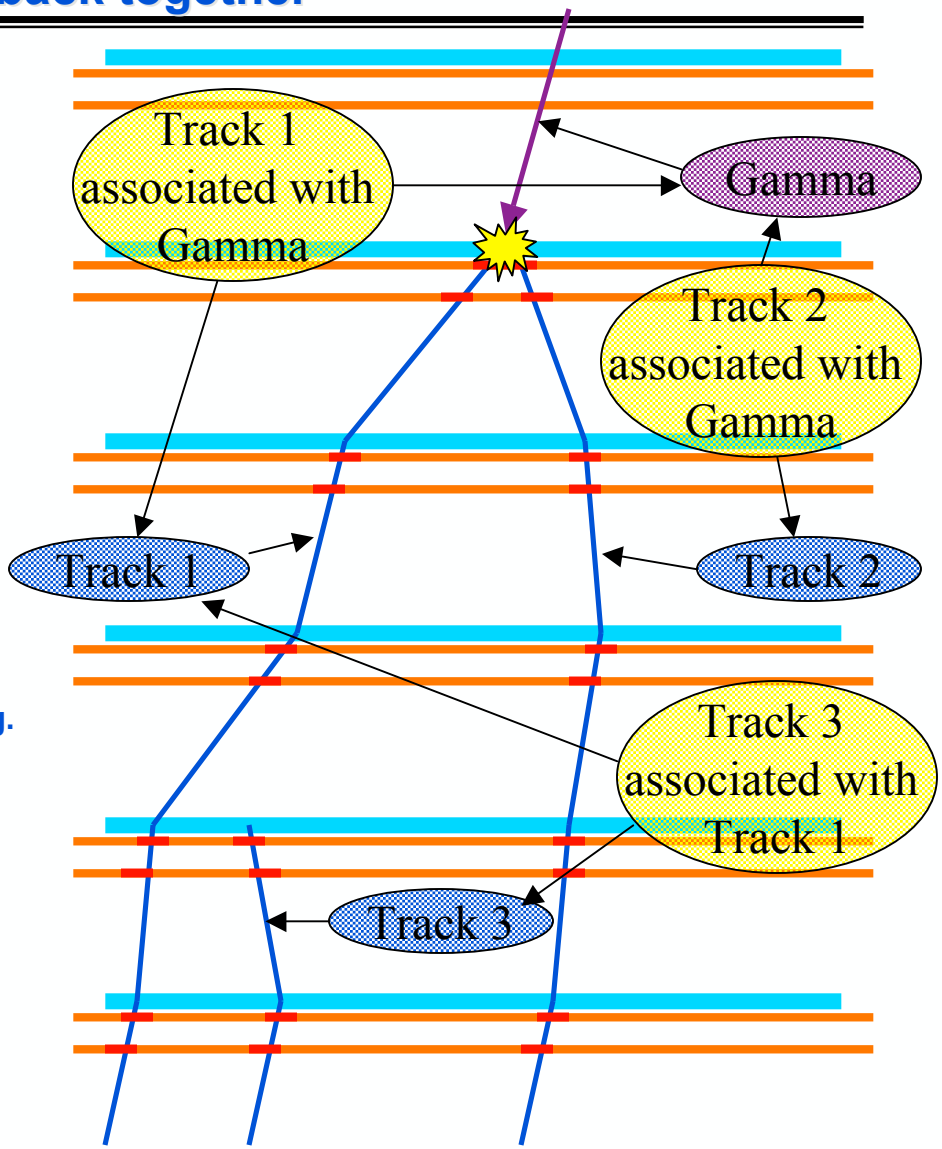


T. Usher

Recon comparison to Monte Carlo

Putting it back together

- **Step One**
  - Group McPositionHits by common McParticle Parent
  - Form base MC Tracks
- **Step Two**
  - McParticle tree puts event back together
    - Identify McPositionHits which form primary electron and positron tracks
    - Identify remaining McPositionHits associated with particles produced by electron or positron as they traverse the tracker
    - -or- as hits associated with background (e.g. backsplash from the calorimeter)
- **Step Three**
  - Use Relational Table to determine:
    - Did hit produce a cluster?
    - Was hit shared by another track?



Recon comparison to Monte Carlo

What can we do with this?

Currently, can do the following:

- **Compare basic event quantities:**
  - Number tracks per event
  - Total number of hits per event associated with gamma conversion
- **Compare basic tracking quantities:**
  - Number hits per track
  - Number of shared hits per track
  - Number of secondary hits associated per track
  - Track pointing
- **Compare basic vertexing quantities:**
  - Vertex position
  - Vertex pointing

Soon answer questions like:

- **Did track find the "right" hits?**
- **Should we find this track?**
- **Should we find this vertex?**

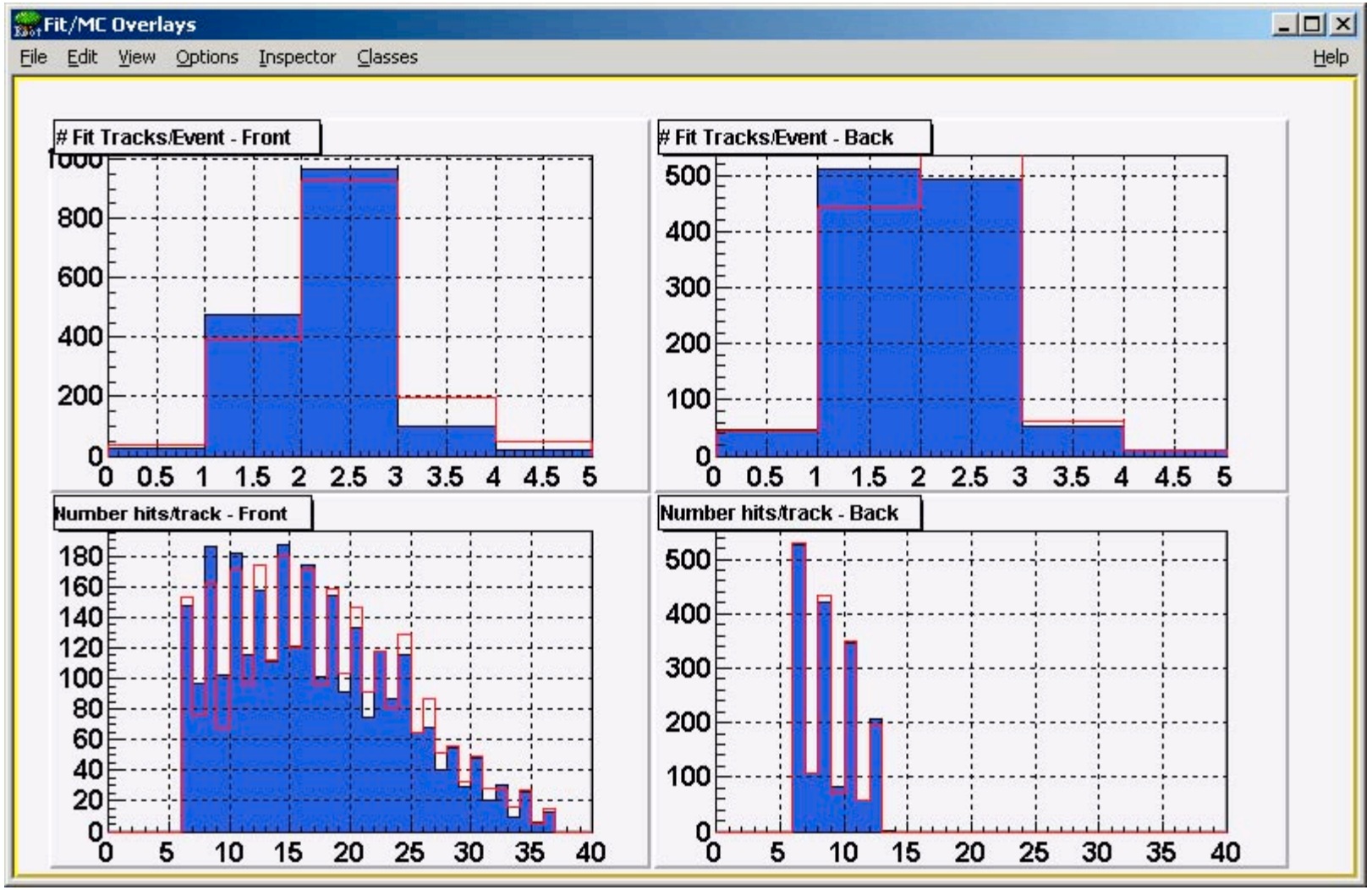
Goals:

- **Create MC tools for evaluating the performance of the TkrRecon Reconstruction**
- **Provide help in the development of new reconstruction algorithms**
- **Use the Monte Carlo to develop the concept of a "findable" track**

TkrRecon / Monte Carlo Comparisons

Examples of Monte Carlo prediction vs Recon output

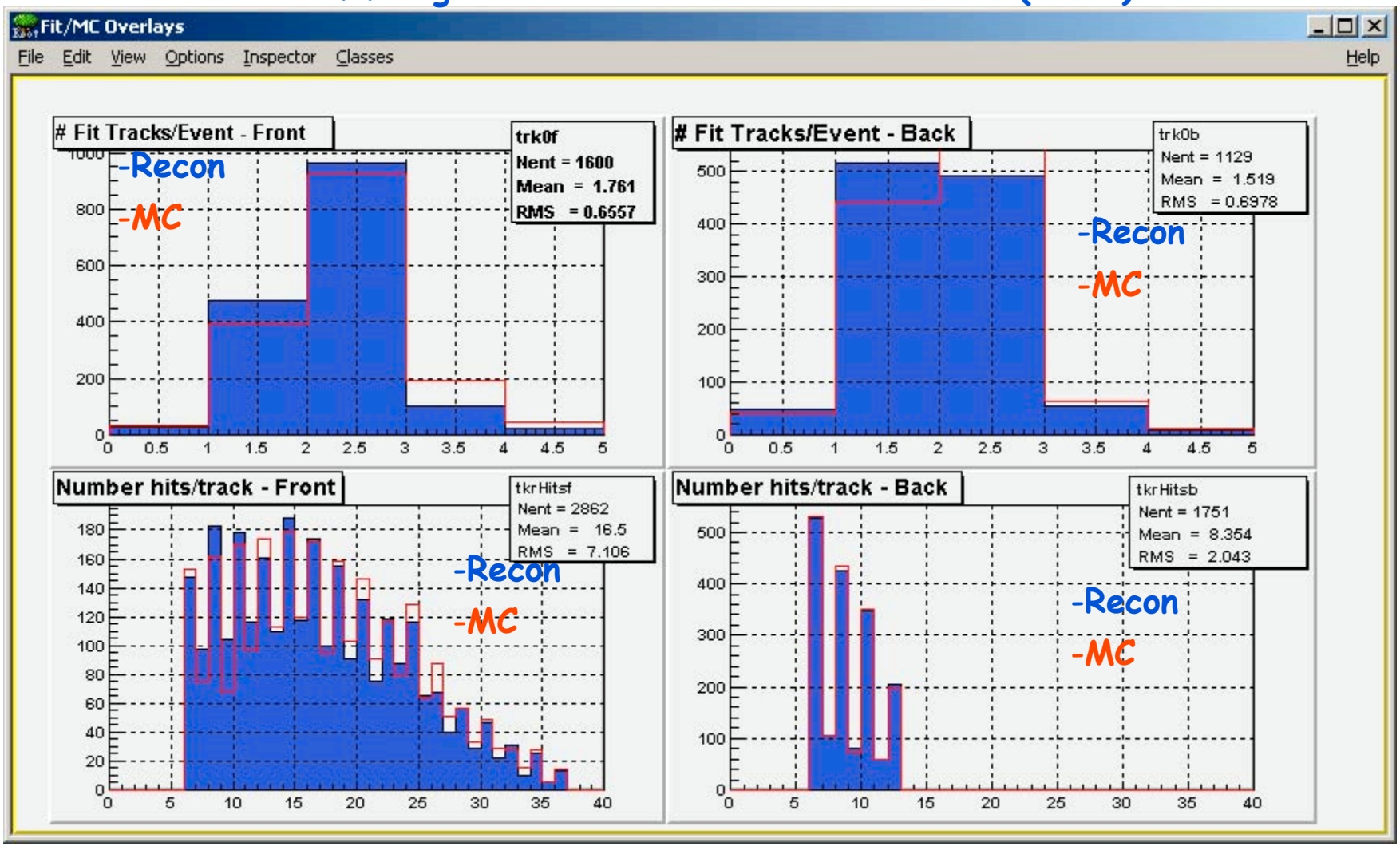
100 MeV Gammas generated into the cone –0.8 < cos(theta) < -1.0



TkrRecon / Monte Carlo Comparisons

Examples of Monte Carlo prediction vs Recon output

100 MeV Gammas generated into the cone –0.8 < cos(theta) < -1.0

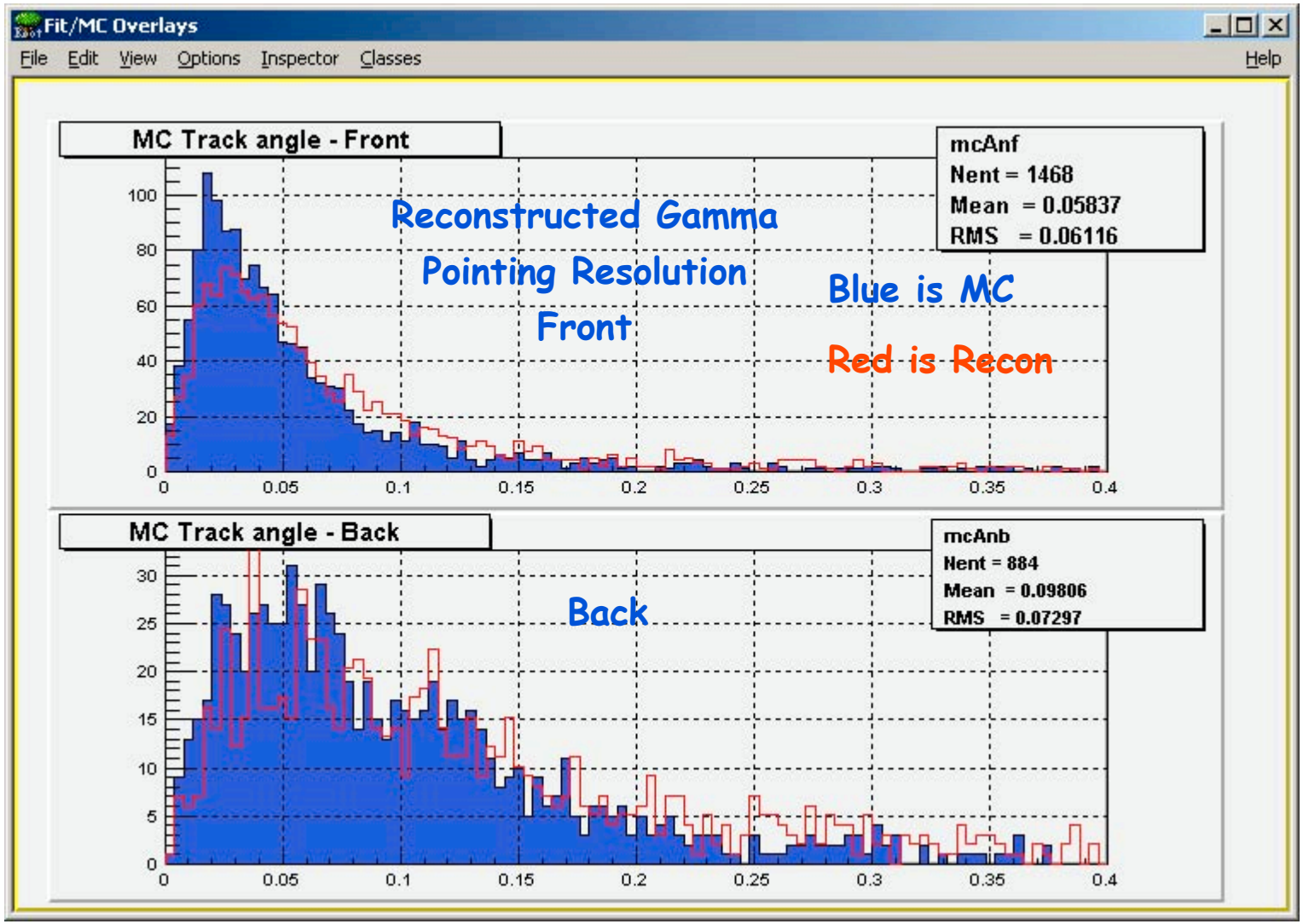


T. Usher

TkrRecon / Monte Carlo Comparisons

Examples of Monte Carlo prediction vs Recon output

100 MeV Gammas generated into the cone –0.8 < cos(theta) < -1.0

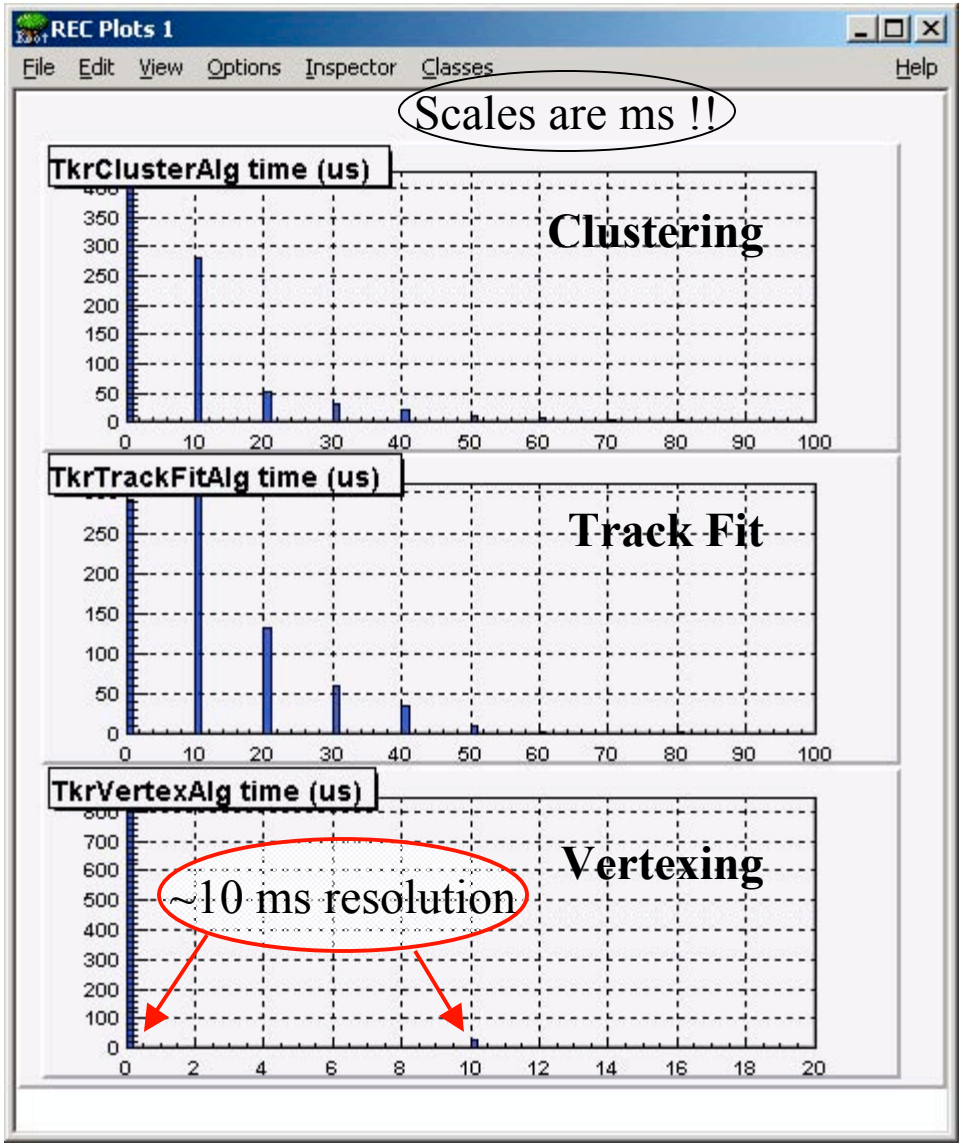


TkrRecon Algorithm Timing

- **Gleam Single Event Time**
  - All Gammas: average ~1 second / event (all events)
  - Dominated by event generation
  - TkrRecon reconstruction a close second!! (triggered events)
- **TkrRecon reconstruction not well bounded?**
  - Reported single event recon time 3600 seconds !!
  - TkrRecon people have seen events with 300 second recon times
  - Obviously, not allowable
- **Determining the root cause**
  - Use Gaudi Timing Services
    - ChronoStatSvc
  - Look at timing of individual TkrRecon Algorithms
    - Clustering
    - Track finding
    - Track fit
    - Vertex finding/fitting

TkrRecon Algorithm Timing

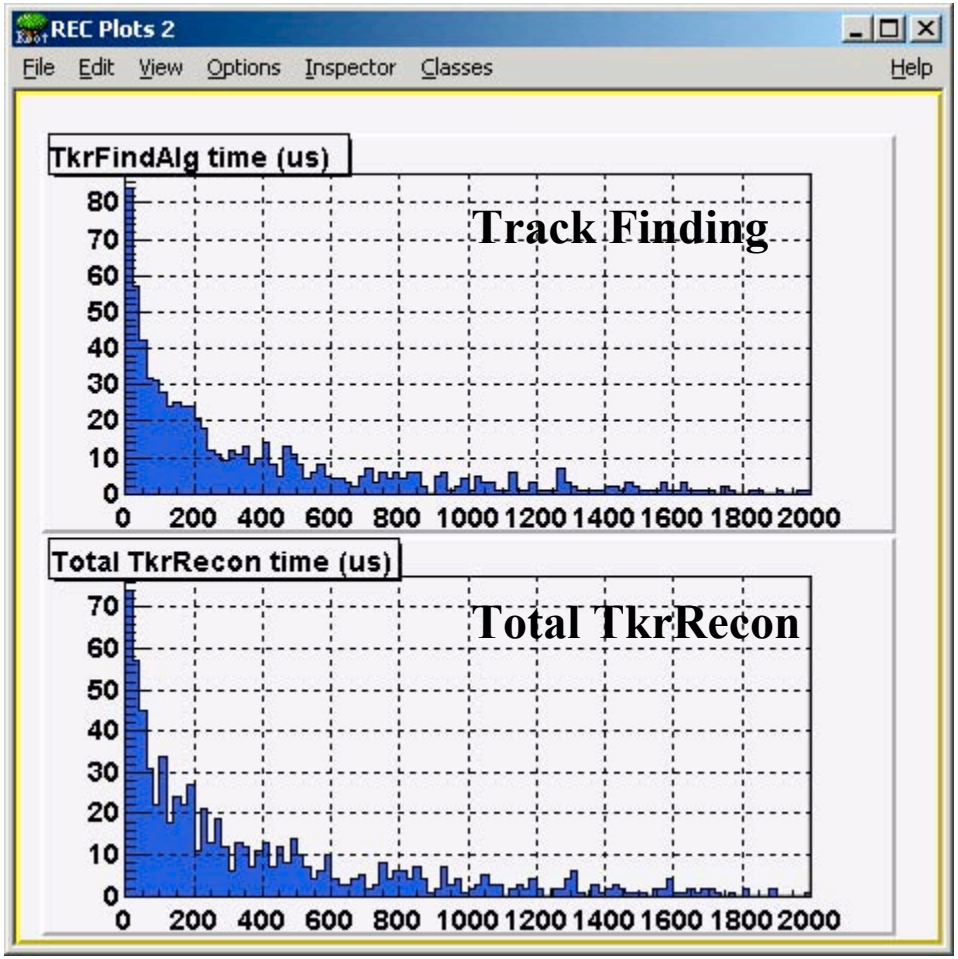
- **Look at timing for all_gamma run**
  - 5000 total events generated
  - Covers wide range of energies
  - Problem energy dependent?
- **First Discovery:**
  - ChronoStatSvc timing resolution
    - ~10 ms !!
    - Worse than expected (?)
- **Well behaved TkrRecon Algorithms**
  - Clustering
    - Average time ~ 10 ms
    - Longest time 630 us
  - Track Fit
    - Average time ~ 8 us
    - Longest time ~ 12 us
  - Vertexing
    - Average time ~250 us
    - Longest time ~10 ms



T. Usher

TkrRecon Algorithm Timing

- **Track Finding is the culprit**
  - Run only on triggered events
  - Average event time ~ 1 second
  - Maximum event time 300 seconds
  - Long tail visible in individual track finding times
  - Completely dominates the total TkrRecon reconstruction time
- **What is the problem?**
  - Default PatRec (combo) has cutoff to prevent it from running "wild"
  - Problem is most likely one level down from track finding
    - Track Propagator
  - Back under investigation after collaboration meeting!



T. Usher

Project List

Short Term (current)

- Code bug fixes, maintenance, etc.
  - e.g. fix runaway events
- Performance Studies
  - CDR (psf, background cuts, etc.)
  - Monte Carlo Verification
- Ongoing code development
  - TDS class design – next iteration
  - Simulation/Digitization
    - Bari Digi Alg
    - ToT
  - New algorithms
    - Neural Net pat rec running
    - Kalman Filter vertexing
    - etc.
  - Analysis tools
    - Track/vertex selection, etc.
- Monitoring / System Tests
  - Augment existing monitoring histograms
  - "fix" nightly build results
- Calibrations
  - Bad strips calibration for EM coming up

T. Usher

Medium Term (~ 6 months)

- Performance Studies
  - MC definition of "findable" track
  - Study edge regions
    - e.g. low energy gammas
- Ongoing code development
  - Event Filter
    - Shape and direction of photons
- Bug fixes, maintenance, etc.

Long Term (> 6 months)

- Alignment
- Real data
  - Nothing like data to burst a happy tracker's bubble

Plenty to keep us busy!!