The GLAST Event Reconstruction: What's in it for DC-1?

Digitization Algorithms
Calorimeter Reconstruction
Tracker Reconstruction
ACD Reconstruction
Plans

GLAST Ground Software Workshop
Tuesday, July 15, 2003

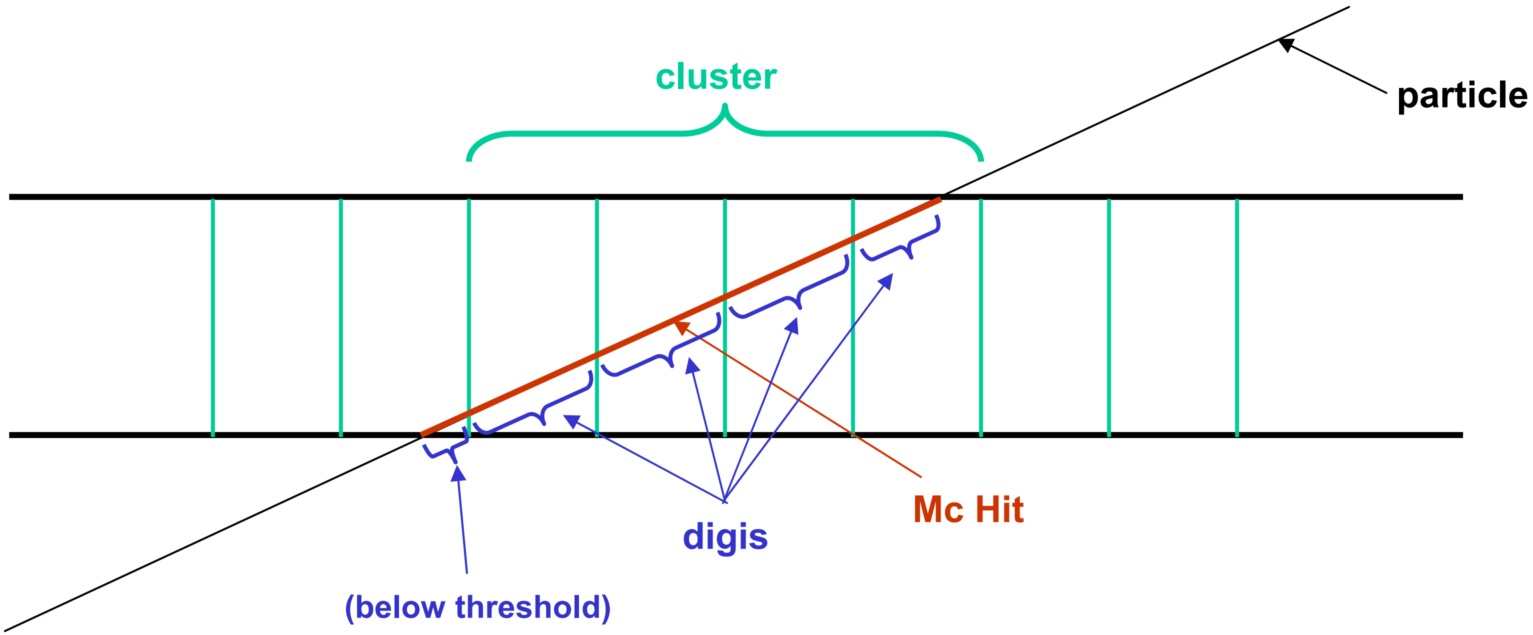
Tracy Usher
(representing all the recon folks!)

Digitization Algorithms

- **Digitization Algorithms**
  - Geant4 tracks particles through the LAT and records interactions in "active" volumes
    - McIntegratingHits in the Calorimeter crystals
    - McPositionHits in the tracker silicon layers and the ACD tiles
  - Digitization algorithms then convert this MC information to simulate the electronics output
- **Three Algorithms:**
  - Tracker Digitization (TkrSimpleDigiAlg)
    - Converts McPositionHits to tracker digi output (hit strips, ToT)
    - (see slides)
  - Calorimeter Digitization (CalDigiAlg)
    - Converts McIntegratingHit information to digi output (ADC output)
    - (see slides)
  - ACD Digitization (AcdDigiAlg)
    - Etc…

Tracker Simulation / Digitization

Geant4 treats the entire silicon plane as a unit. Energy is deposited with "landau" fluctuations. Using this information, the digitization algorithm then determines which strips are hit.



Tracker Simulation / Digitization

- McPositionHit is input
- Digitization Algorithms:
  - Two Digitization Algorithms exist:
    - In the Simple Digitization (SimpleDigiAlg) algorithm energy deposited in the silicon is divided according to the path length (no fluctuations).
      - Time-over-threshold is linear in deposited energy
    - The Bari Digitization (BariDigiAlg) is a complete electronics simulation which accounts for fluctuations in deposited energy as the particle traverses the silicon, etc.
      - More detail – much slower execution speed
  - Currently merging two algorithms into one package
    - Common features in one algorithm
      - Bad strips, failed layers, noisy strip generation, etc.
    - Flip of switch to change from one algorithm to the other
      - Easy comparison of the two – cross check results
  - DC-1 will run SimpleDigiAlg
    - Primary issue is execution speed
    - Will use merged algorithm if ready soon
- TkrDigi is output

Digitization Algorithms

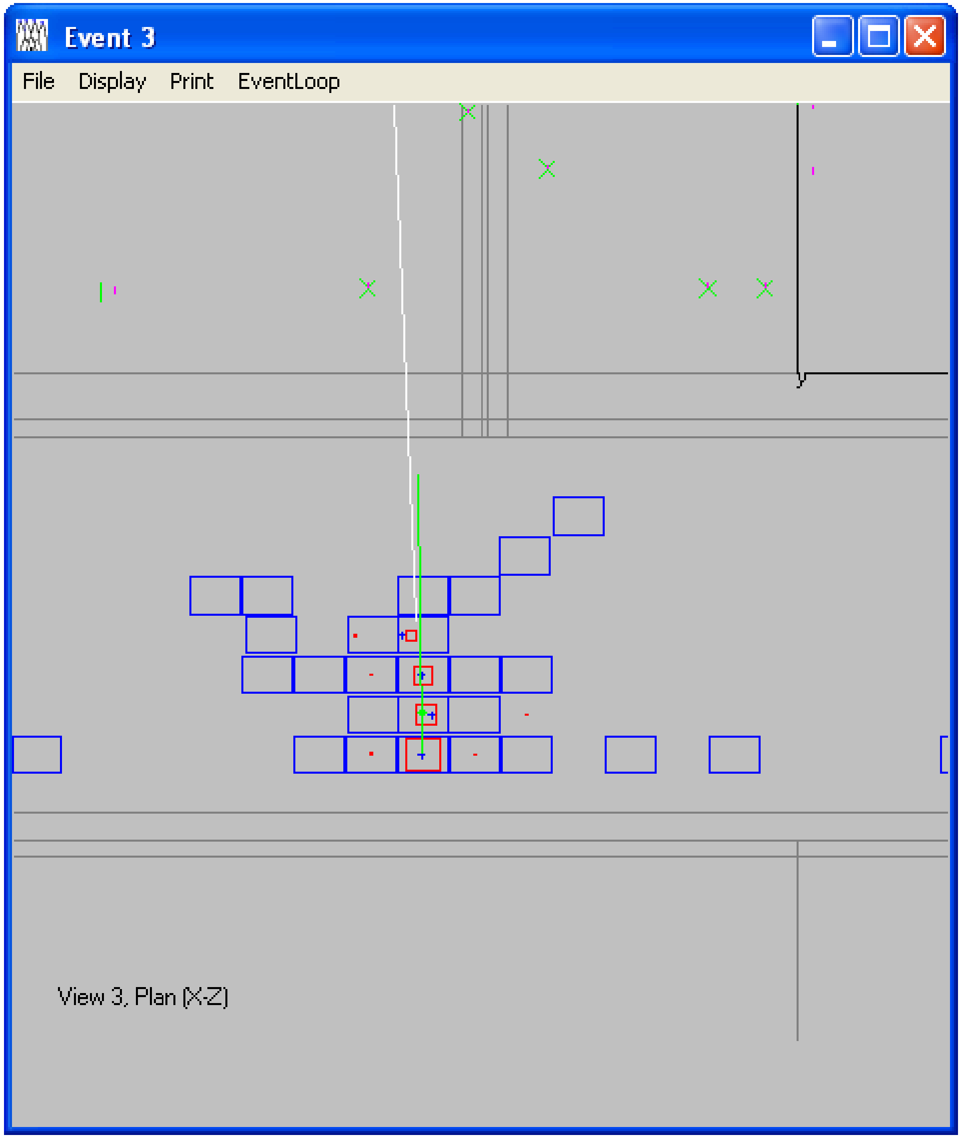
- Calorimeter Digitization
  - McIntegratingHit is input
  - Cal Digitization:
    - For deposit in a crystal segment, take into account light propagation to the two ends and apply light taper based on position along the length
      Keep track of direct deposit in the diode
      Add noise to the diode
      Combine (with the appropriate scale factor) with crystal deposits
    - Add noise to "unhit" crystals; save those above threshold
    - Convert to ADC units and pick the appropriate readout range for hits above threshold
  - CalDigi is output

- ACD Digitization
  - McPositionHit is input
  - ACD digitization:
    - Energy deposited converted to PMT output (with corrections)
    - Tile "hit" if above threshold
  - AcdDigi is output

Calorimeter Reconstruction

- **Basic Reconstruction Goals:**
  - Reconstruct the energy of the incident gamma ray
  - Reconstruct its direction
  - Reconstruct its position within the calorimeter

- **This section thanks to**
  - Mark Strickman
  - Berrie Giebels



Calorimeter Reconstruction

- Step 1: For Each Crystal:
  - Convert ADC scale to Energy measured at each crystal end
    - Gain and Pedestal defined from muon (ground) and cosmic ray (flight) calibrations
    - Integral nonlinearity defined from charge injection calibration (not available for DC-1)
  - Centroid position of event in crystal
    - Position approximately proportional to the “signal asymmetry:”
      - $(s_2 - s_1)/(s_2 + s_1)$ Linear Taper
      - $\mathrm{Log}(s_2/s_1)$ Exponential Taper
      - Where $s_{1,2}$ are the signals from the diodes on each end of the crystal
    - Current energy estimate assumes a linear response as a function of position in the crystal.
      - Asymmetry derived position will ultimately be used to improve crystal energy estimate (not available for DC-1)

- Step 2: For Each Layer:
  - Sum the individual crystal energies

Calorimeter Reconstruction

- Step 3: Energy Reconstruction Algorithms
  - Start with the total raw energy obtained from the sum over layers
    - Note that this raw energy does not take into account losses:
      - In the tracker
      - In the inter-tower cracks
      - From leakage out of the calorimeter
  - Get the correction to the total raw energy
    - Currently three algorithms available (more on these in next slides):
      - Shower profile fitting
      - Shower profile model (due to Bill Atwood)
      - Last layer correlation method (due to Berrie Giebels)
    - All three algorithms currently run in parallel
    - For DC-1 rely, primarily, on the Shower Profile Model
- Step 4: Estimation of the incident position and direction
  - Uses a two-dimensional energy-weighted centroid position in each layer

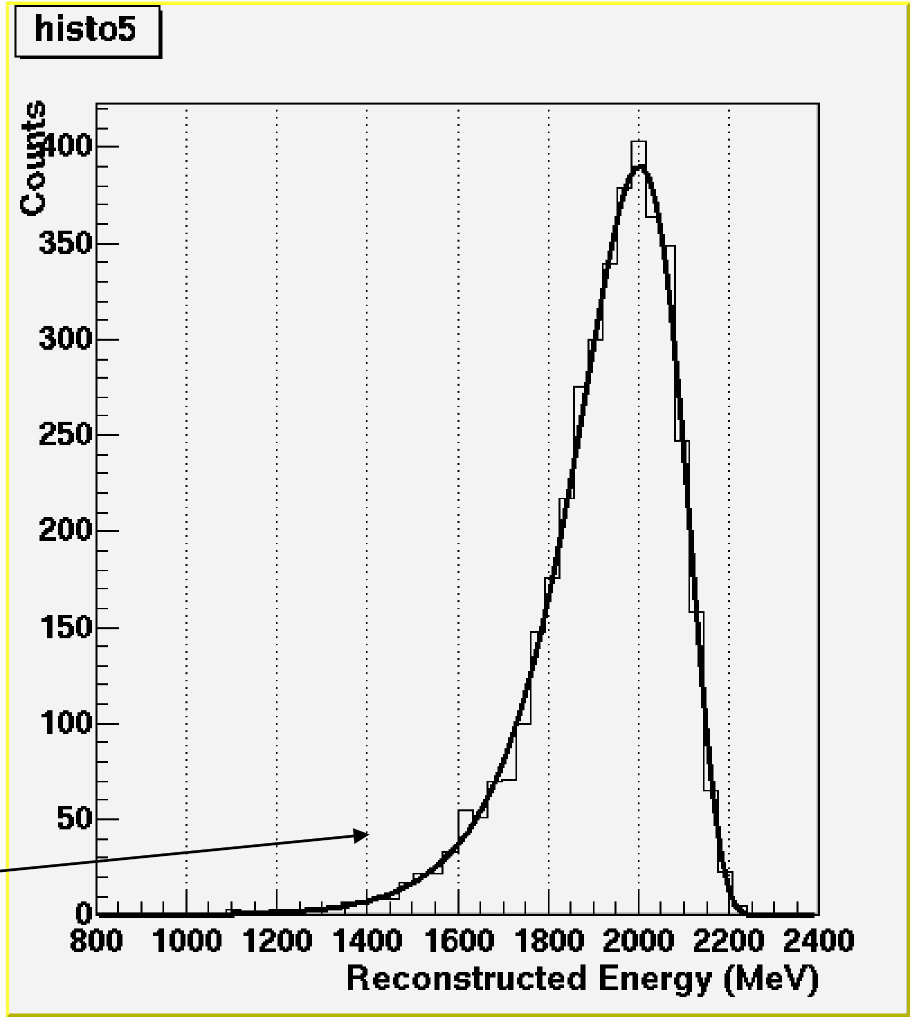
Calorimeter Reconstruction

- Energy Reconstruction Correction Algorithms
  - Shower profile fitting
    - "Standard" approach
    - Works well, especially at high energies
    - Not very dependent on Cal geometry details
      - Except for the intertower cracks
    - But…
      - Needs modification to work at all incident angles
      - Needs modification to account for inter-tower cracks
  - Shower profile model (method due to Bill Atwood)
    - Uses a fixed (not fitted) shower profile model with corrections for inter-tower cracks
      - Valid over a wide range of energies and angles (especially low energies)
    - But…
      - Requires significant tuning
      - Cal folks have not yet compared performance in detail to other algorithms

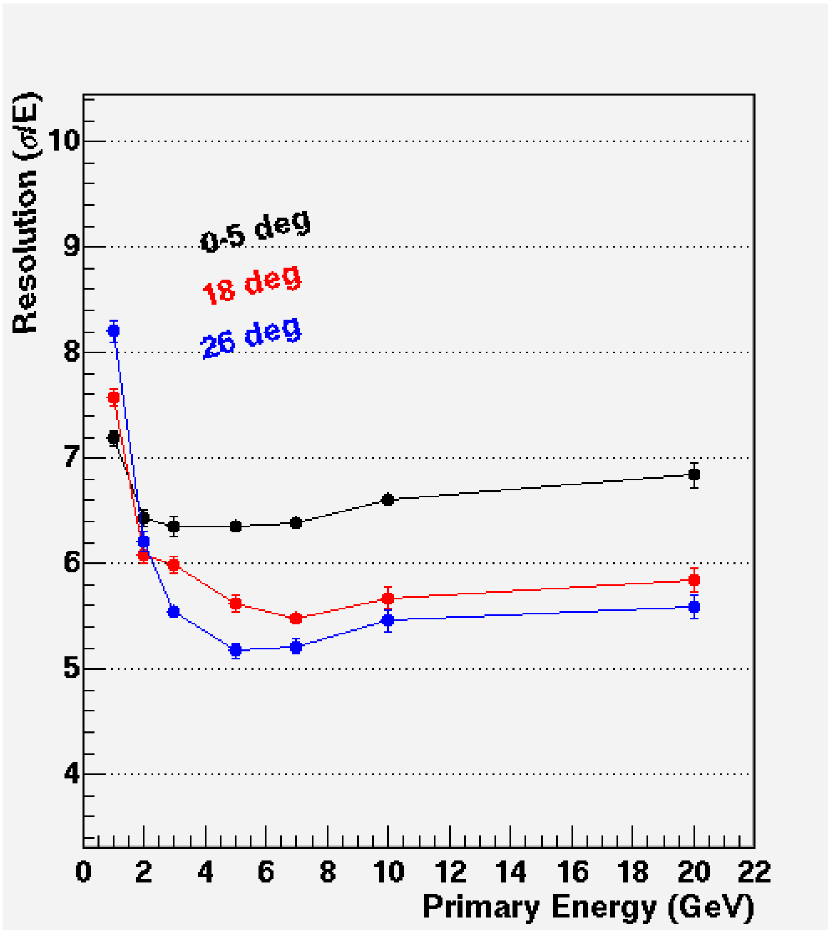
Calorimeter Reconstruction

Energy Reconstruction Correction Algorithms

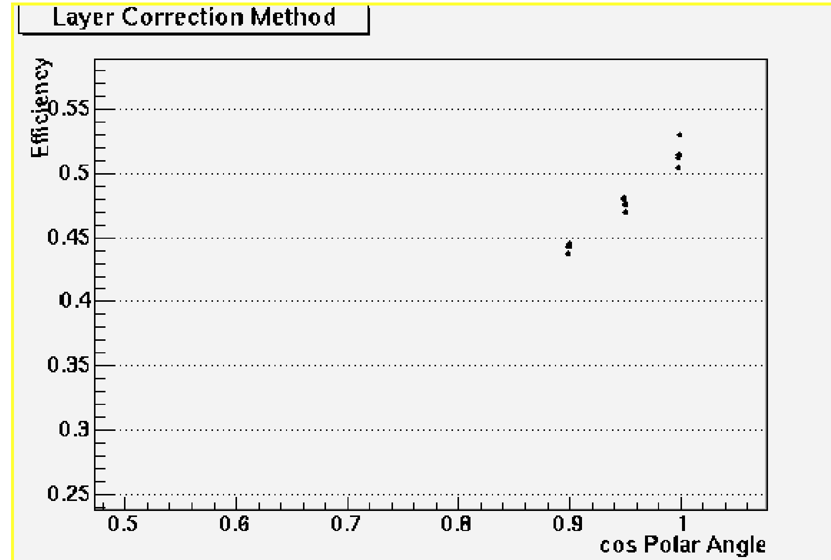
- Last layer correlation method: Use correlation between leakage and energy deposited in last layer of calorimeter to estimate leakage energy
  - Generally produces better resolution than profile fitting
  - But…
    - Stops working when shower profile max is outside of Cal
      - > 50 GeV at normal incidence
    - Does not work well at low energy
    - Parameters are dependent on Cal geometry
- Sample of reconstructed 2 GeV photon peak from 26° incident beam using last layer correlation



Calorimeter Reconstruction
Last Layer Correlation Method Performance



Resolution vs. Energy and incident angle



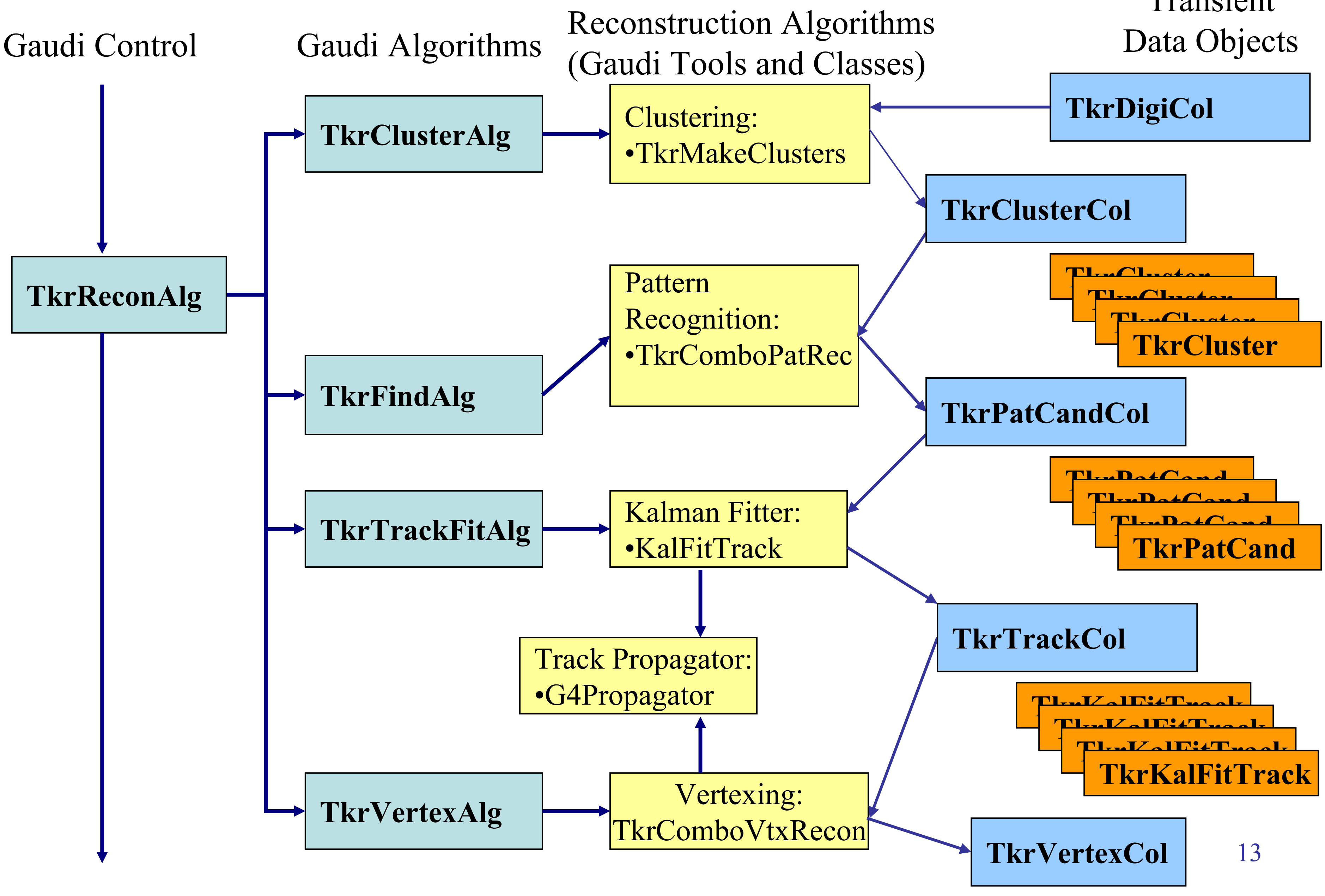
Correlation method efficiency (fraction of triggered events for which recon energy is produced) vs. polar angle for seven energies from 1-20 GeV. Inefficiency is due to failure of algorithm near cracks.

Berrie Giebels hard at work on this!

Tracker Reconstruction

- For DC-1, TkrRecon will go mostly "as is"
  - Use Default Recon (see diagram)
    - Cluster hit strips using standard Clustering algorithm
    - Find candidate tracks using "Combo" Pattern Recognition
    - Fit the track candidates with the Kalman Filter track fit
    - Combine tracks to form vertices using the "Combo" Vertexing Algorithm
    - Track propagation with G4Propagator
      - Tied directly to the G4Geometry (and constants)
      - Gismo getting phased out
  - Utilities
    - TkrGeometrySvc – tied to DetModel
    - Alignment – not needed for DC-1
    - Bad strips – not needed for DC-1
    - Failure modes – not needed for DC-1
  - Use the Iterative Recon?

Tracker Reconstruction Diagram

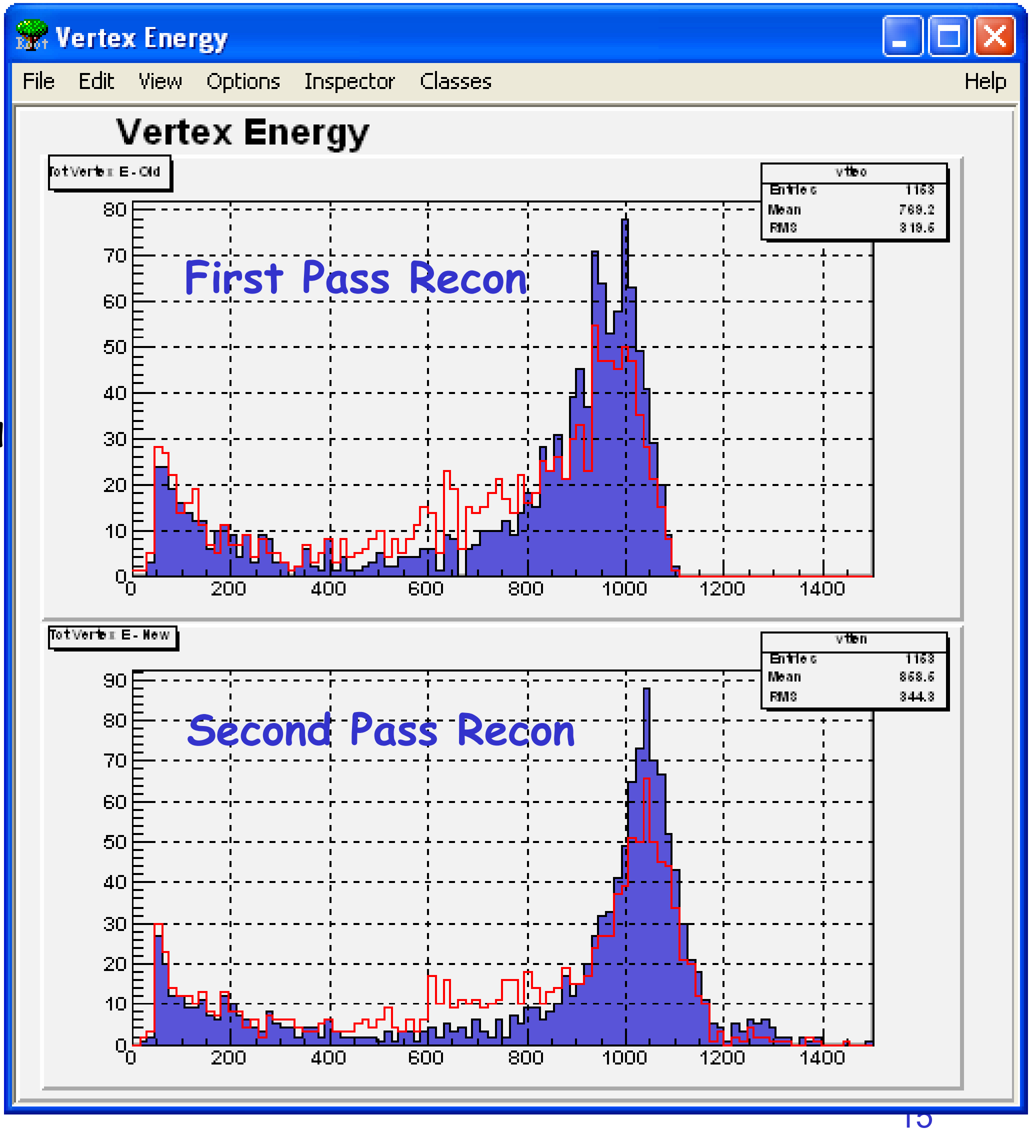


Tracker Reconstruction
Iterative Tracker Recon

- What is it?
  - The Iterative Recon is a mechanism for allowing parts of the Tracker Reconstruction software to be called more than once per event
  - In particular, existing pattern recognition tracks can be refit and the vertex algorithm re-run
- Why is it needed?
  - The Calorimeter would like the output of TkrRecon when running the energy correction algorithms.
  - At the same time, TkrRecon wants the best energy estimate from the Calorimeter to get the best track fits and, subsequently, the best vertices
  - The Iterative Recon solves this problem by providing the Calorimeter Recon with sufficient tracking information to get an improved energy estimate, which can be fed back to the track fit and vertexing algorithms
  - The process can be repeated as many times as the user likes (in principal)
- Will it be ready in time?
  - Code already in the last three GlastRelease tags
  - Only needs to be activated by modifying the standard recon job options file
- Do we want/need it?

Tracker Reconstruction
Iterative Tracker Recon

- Example using "WAgammas"
  - 1 GeV
  - 5° cone about normal
  - Into 6 m^2 area containing Glast
- Plot Energy of the reconstructed vertex
  - Red Histograms - energy of "best" vertex
  - Blue Histograms – include energy of second track if not part of vertex
- In General
  - Reconstructed vertex energy improves
  - Some details to be understood
    - Shift in energy above 1 GeV probably still fallout (in this version of Gleam at least) of Tkr/Cal displacement
    - High energy tail?



Tracker Reconstruction

- DC-1 To Do list:
  - Merged digitization
    - Not really needed for DC-1 but want to have done so that both digi algorithms will merge into the same development path
  - Iterative Recon part of the main reconstruction loop?
    - Need to find source of observed energy shift in final result
  - Try to find and remove remaining hardwired numbers
    - Goal: pass the Leon TkrRecon displacement test
  - Update the TkrRecon documentation
    - Mainpage
    - Doxygen
    - Inline comments in updated classes
  - General housekeeping
  - General bug hunting
    - Check out "anomalous" events
    - Etc.

- Aim for updates to reconstruction algorithms in DC-2
  - If anything can be found to be better than existing recon!

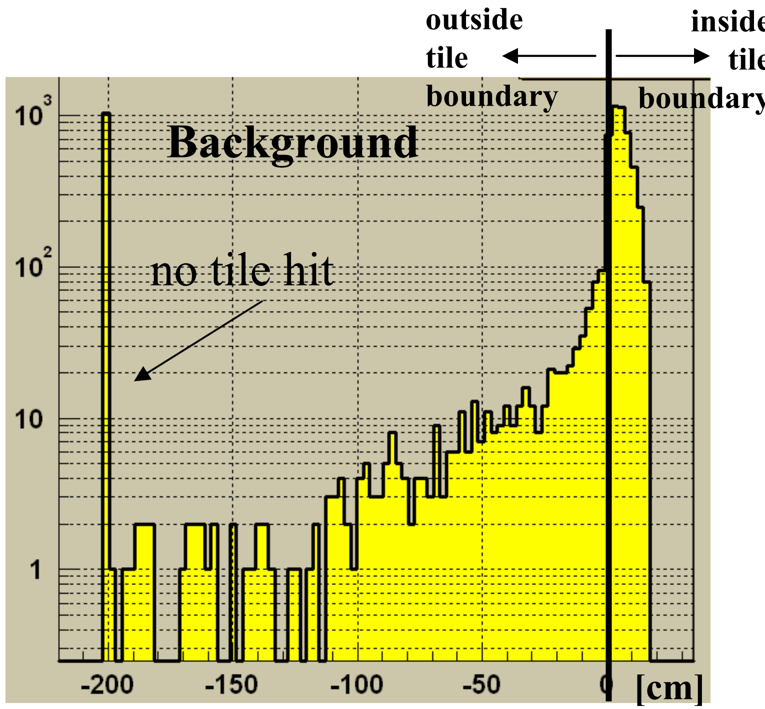
ACD Recon

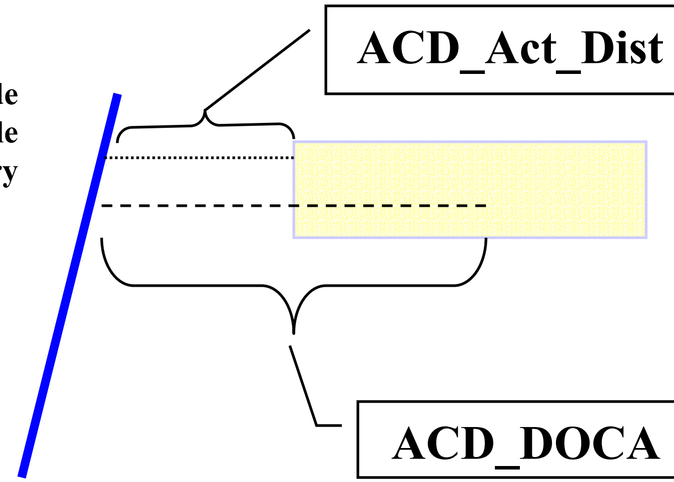
Primary outputs (both originally designed by Bill Atwood):

1) Active Distance:

measures distance from edge

(done once for entire ACD, and by region)





2) **Distance of Closest Approach (DOCA):** measure distance from the center of a tile.
Done also for different regions of the ACD, since tile size varies.

Recon also provides: energy deposition estimate and counts of tiles above threshold by region.

(slide courtesy of Heather Kelly)

Reconstruction
General Questions for DC-1

- Cal Recon
  - When to stop “development” of energy reconstruction algorithms?
- Tkr Recon
  - Do we want to include the iterative recon in DC-1?
- Event Summary
  - TDS output class to summarize the event?
  - What should be in this class?
- …

Recon – The Result