TkrRecon 2005

GLAST Track Reconstruction:
What's old and what's new since DC1

The GLAST Tracker Software Group
Bill A., Johann C-T., Leon R., Tracy U.

The Big Picture

Task: Reconstruct particle trajectories within the GLAST Tracker enabling γ-ray reconstruction and background rejection

Note: Background rejection is extremely important; mistakes made during reconstruction lead directly to background contamination and background events occur at a rate several 1000's of times that of the γ-ray signal.

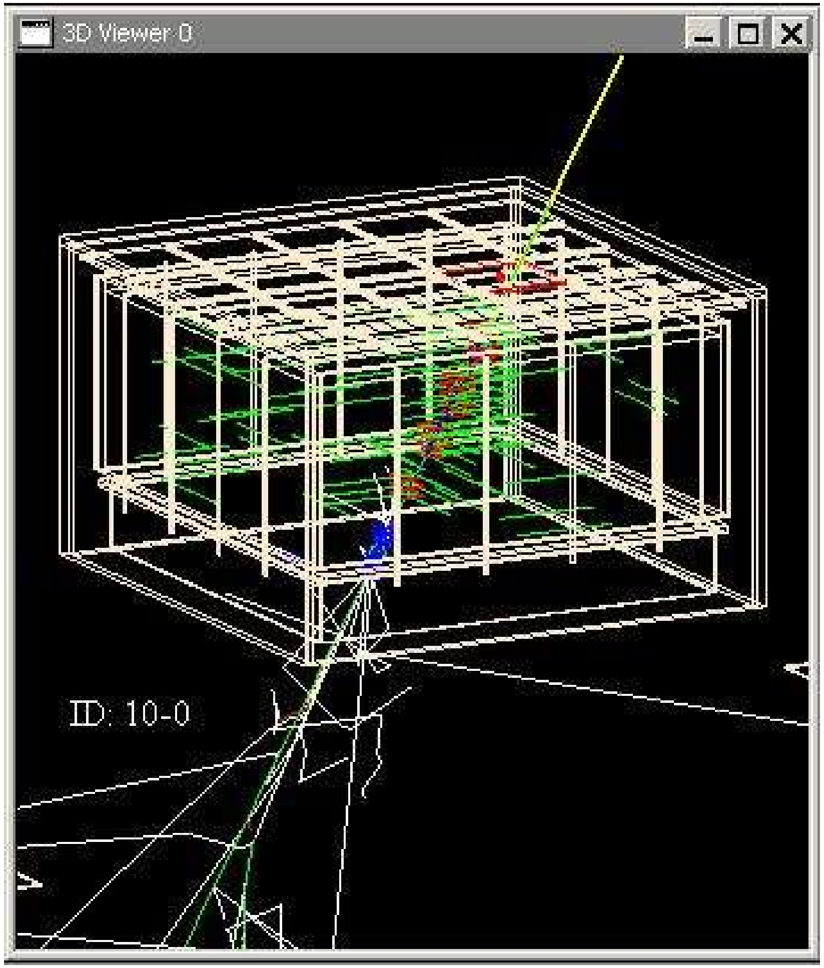
Starting Point: DC1 TkrRecon

This was reasonably successful track reconstructor but:

a) Hard-coded control parameters and geometry
b) Overly complex structure – hard to read and maintain
c) Incomplete description of the track objects

Topics

- Geometry
- Calibration
- TDS Objects
  - Clusters
  - Track objects
- Reconstruction
  - Track Finding
  - Track Fitting
  - "Vertexing"
- "Sea Trials" Redux



Geometry

- Initializes itself using the propagator (mostly!)
- Finds out
  - what kind of towers, how many, where they are
  - relation of trays, layers and planes
- To do:
  - Free the code from fixed arrays
    - IndexedVector
    - Allocate multidimensional arrays at dynamically

Using Calibration Data

Implemented:

- Hot/Dead strips (xml)
- Splits (xml)
- Tot charge-injection (root)

Still to finish

- Tot muon scale (root) – almost there
- Alignment
  - text file implemented

Making Clusters

- Clusters are now made by a TkrUtil tool, which goes through all the hit strips and groups them into contiguous clusters.
- Dead strips adjacent to hit strips are included in the cluster, up to 10.
- New features:
  - TkrId, status bits
  - Number of bad strips in the cluster
  - First guess at the "Mip"s
  - Bad Clusters

Clusters: a Bit more Detail

Corrected ToT ("Mip"s)

- Raw ToT is corrected by charge-injection constants and muon calibration scale, to yield an equivalent number of Mips
  - For 1-strip clusters, this is well defined.
  - For clusters of 3 or greater strips, those constants are chosen which produce the smallest corrected value, except that the edge strips are excluded
  - For 2-strip clusters, the average is used
- Path length correction can't be done at this stage
  - requires knowledge of track angles

Bad Clusters

- Every time a bad-strips calibration is encountered, the cluster tool generates bad clusters from the known dead strips.
- These are stored for use during pattern recognition

Track Objects Simplified

- **Old**
  - TkrFitHit
  - TkrFitMatrix
  - TkrFitPar
  - TkrFitPlane
  - TkrFitTrackBase
  - TkrKalFitTrackBase
  - TkrKalFitTrack
  - TkrPatCand
  - TkrPatCandHit
  - TkrRecInfo
  - TkrTrackTab
- **New**
  - TkrTrackParams
  - TkrTrackHit
  - TkrTrack

Track Finding: Clusters to Space Points

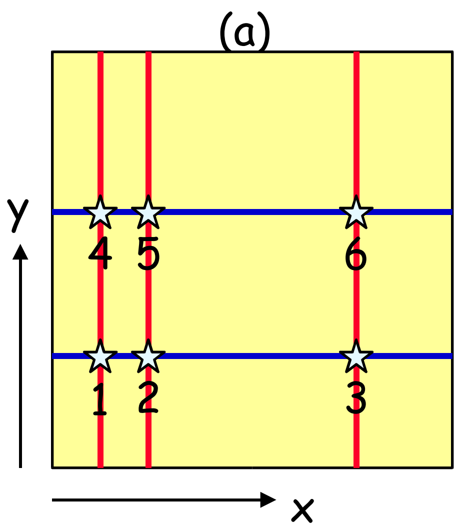
The tracker provides a 3-in-a-row trigger: 3 consecutive X-Y layers firing within a microsecond.

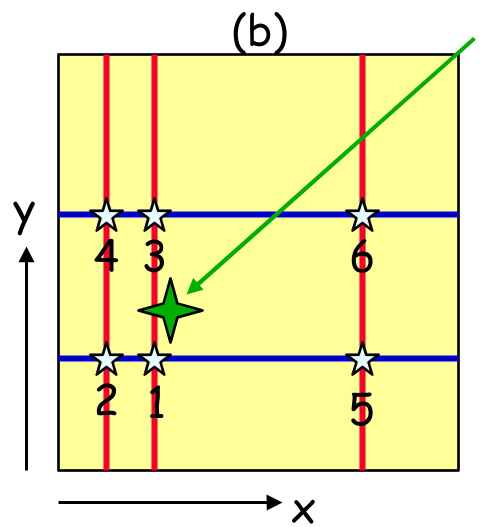
Each X-Y layer yields possible "space points."

Step one: build an object which can cycle over the allowed X-Y pairing in a given GLAST measuring layer

a) Ordered just as they come X's then Y's (numbers on plots below)
b) Ordered with reference to closeness to a given space point

(nextHit)





(x,y) Point about which to search

(nearestHit)

From Space Points to Track Hypothesises

Three Approaches under investigation:

1) "Combo" - Cycle over space points - build an ordered list of tracks

2) "Link & Tree" - Join space points allowing branches - forming a tree-like structure

3) "Neural Net" - Link close by space points (forming "neurons"). Link neurons by rules weighting linkages.

First method is "track by track" -

Pro's: Simple to understand (although details add complications)

Con's: Find bogus tracks early on throws off the remaining by missing assigning hits

Can be quite time consuming depending of depth of search

Methods 2) & 3) are global -

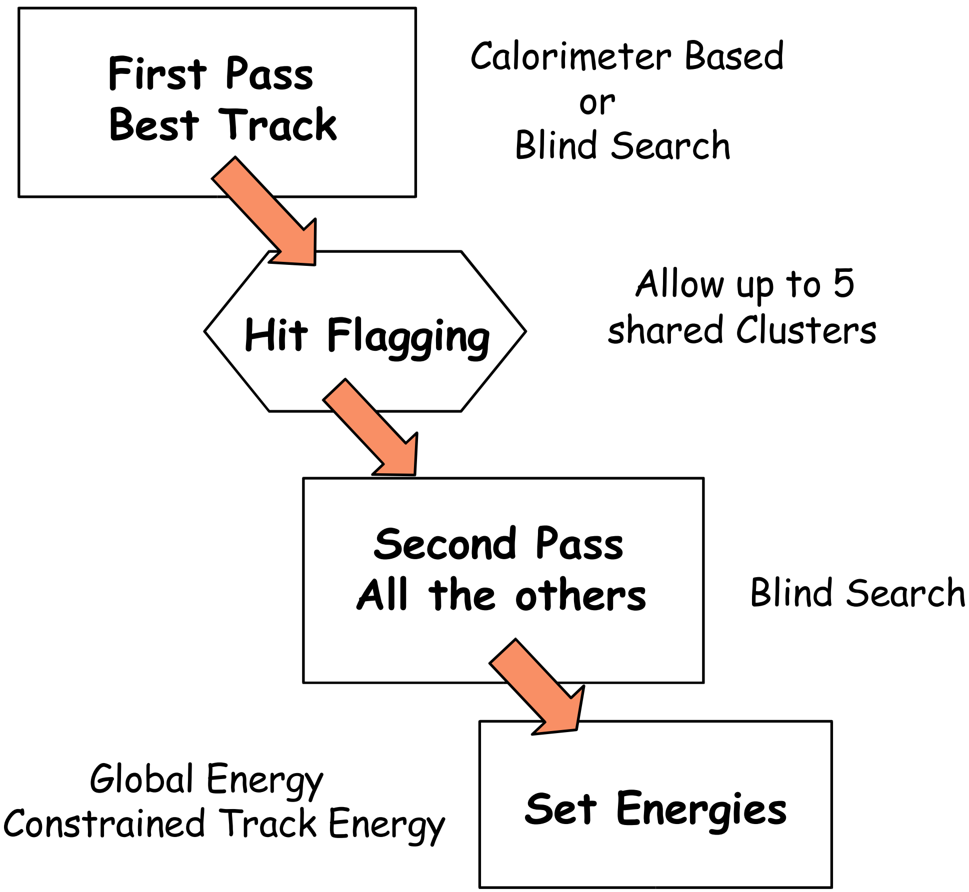
Pro's: Optimized finding across entire event

Con's: Neural Nets can be quite time consuming

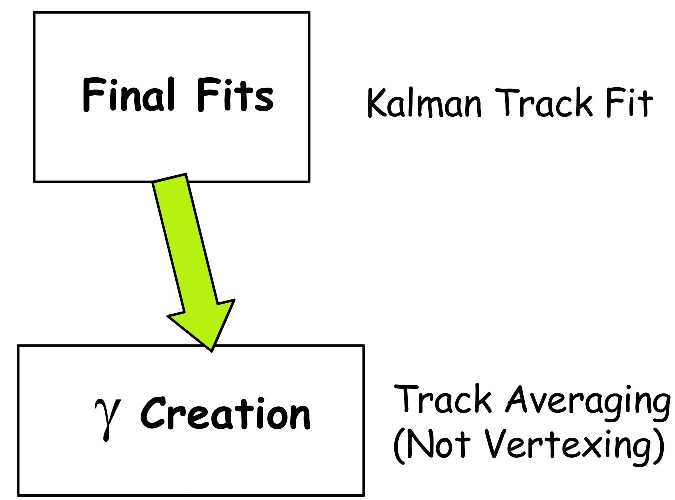
Combo is currently the most advanced, and has been the default. It is also the only one that has been substantially reworked since DC1.

Combo PatRec - Kalman Overview

Pattern Recognition



Track Fitting



Combo PatRec in Detail

Starting Layer: One furthest from the calorimeter

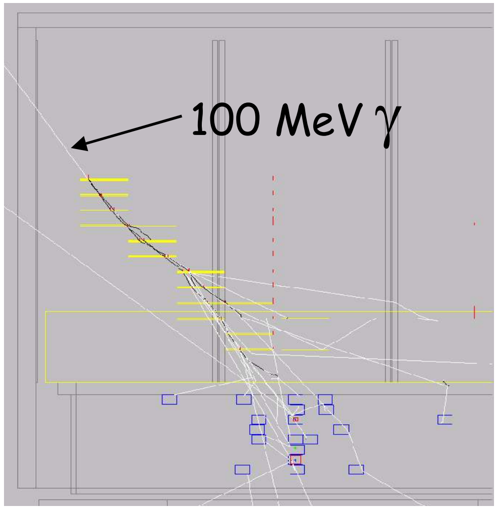
Two Strategies:

1) Calorimeter Energy present ⟹ use energy centroid as a space point!

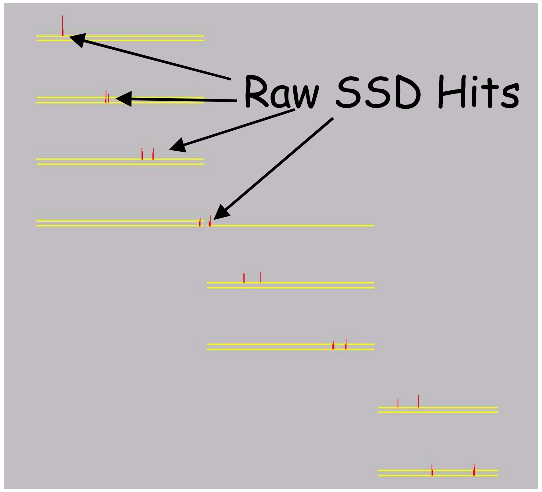
2) Too little Cal. Energy ⟹ use only track points

Combo Pattern Recognition - Processing an Example Event:

The Event as produced by Geant



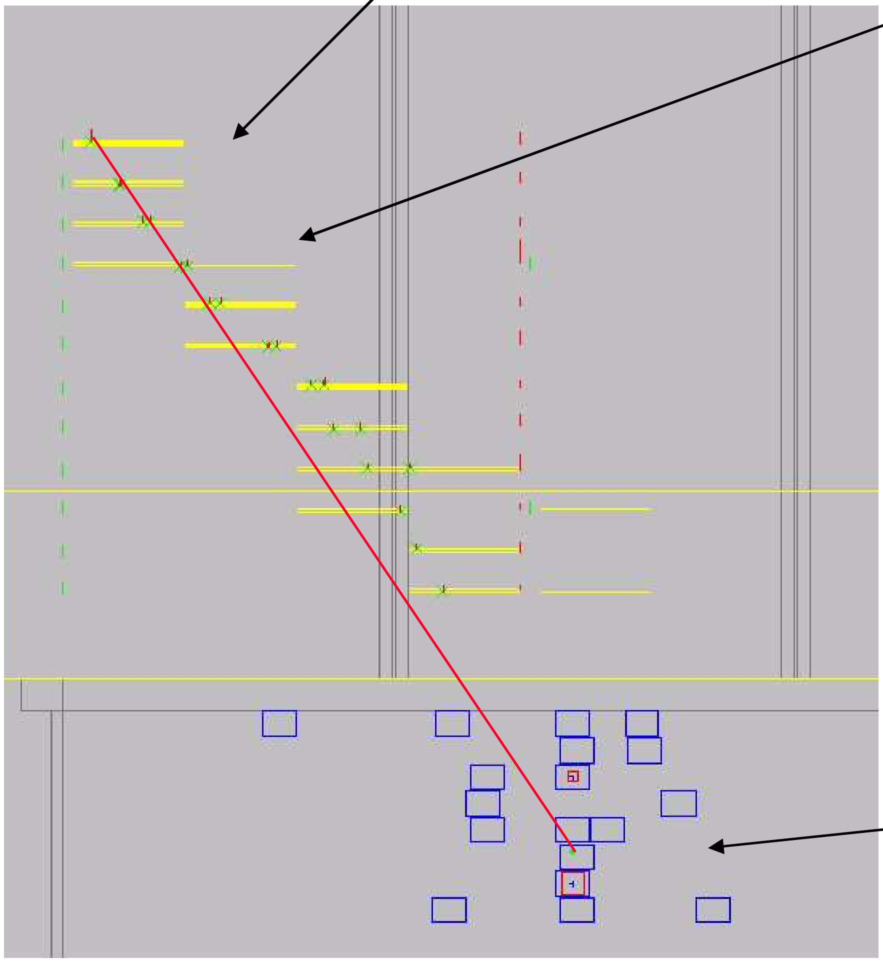
What we see



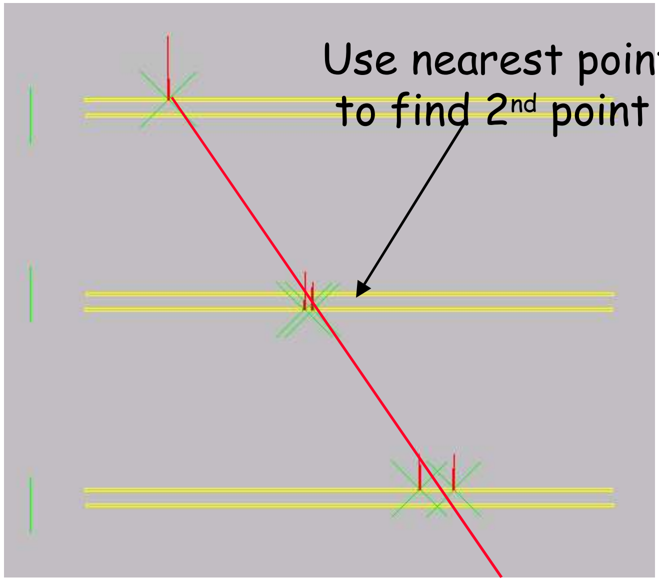
Combo PatRec, cont.

Start with points in outer most layer

First Guess: connect point with Cal. Centroid



Use nearest point to find 2nd point

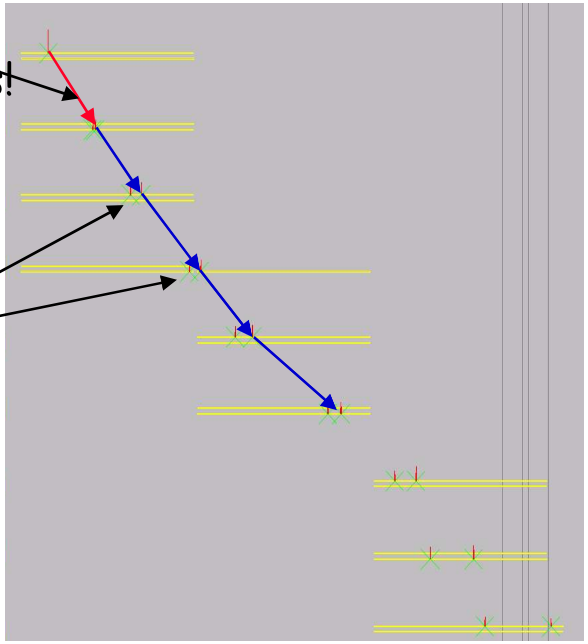


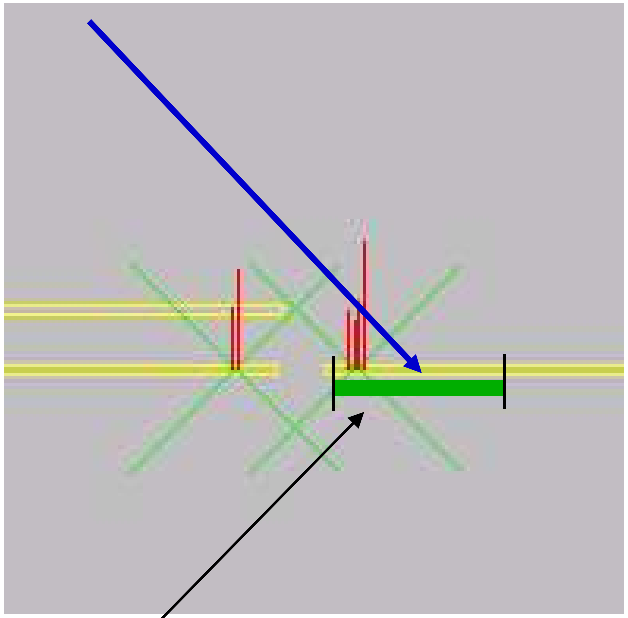
Sufficient Cal. Energy (42 MeV) Use Cal. centroid

Combo PatRec, cont.

Initial Track Guess:
connect first 2 points!

Project and add hits along the track within the search region





The search region is set by propagating the track errors through the GLAST geometry.

The default region is 9σ (set very wide at this stage)

Combo PatRec, cont.

New feature:

- Hits are now recorded even if there is no associated cluster in the plane. If there is no cluster, the following tests are performed, in order, and status bits are assigned:
  - Are we in a failed plane?
  - Are we close† to the active edge of a tower?
  - Are we close to the edge of active silicon?
  - Are we close to a bad cluster?
- Hits that fail all the tests are marked. A certain number of such hits are allowed in a track (setable, currently, none).
- Algorithm needs work
  - changing order of tests
  - use distance to edge of cluster, not center

†currently, within 2 sigmas of predicted error

Combo PatRec, cont.

The blind search proceeds similar to the calorimeter-based search

- 1st point found found - tried in combinatoric order
- 2nd point selected in combinatoric order
- First two points used to project into next layer -
- 3rd point is searched for -
- If found and accepted, track is built by "finding - following" as with calorimeter search, adding hits in each plane.

In this way a list of tracks is formed.

Crucial to success is ordering the list!

Combo PatRec, cont.

Track Selection Parameter Optimization

Ordering Parameter

Q = Track-Quality - C_1* Start-Layer - C_2*First-Kink - C_3*Hit-Size - C_4*Leading-Hits

Track_Quality: "No. Hits" - χ^2 : track length (track tube length) - how poorly hits fit inside it

Start-Layer: Penalize tracks for starting late

First-Kink: Angle between first 2 track segments / Estimated MS angle

Hit-Size: Penalize tracks made up of oversized clusters (see Hit Sharing)

Leading-Hits: These are unpaired X or Y hits at the start of the track. This protects against noise being preferred.

Status: Current parameters set by observing studying single events.
Underway – program to optimize parameters against performance metrics underway (Brian Baughman)

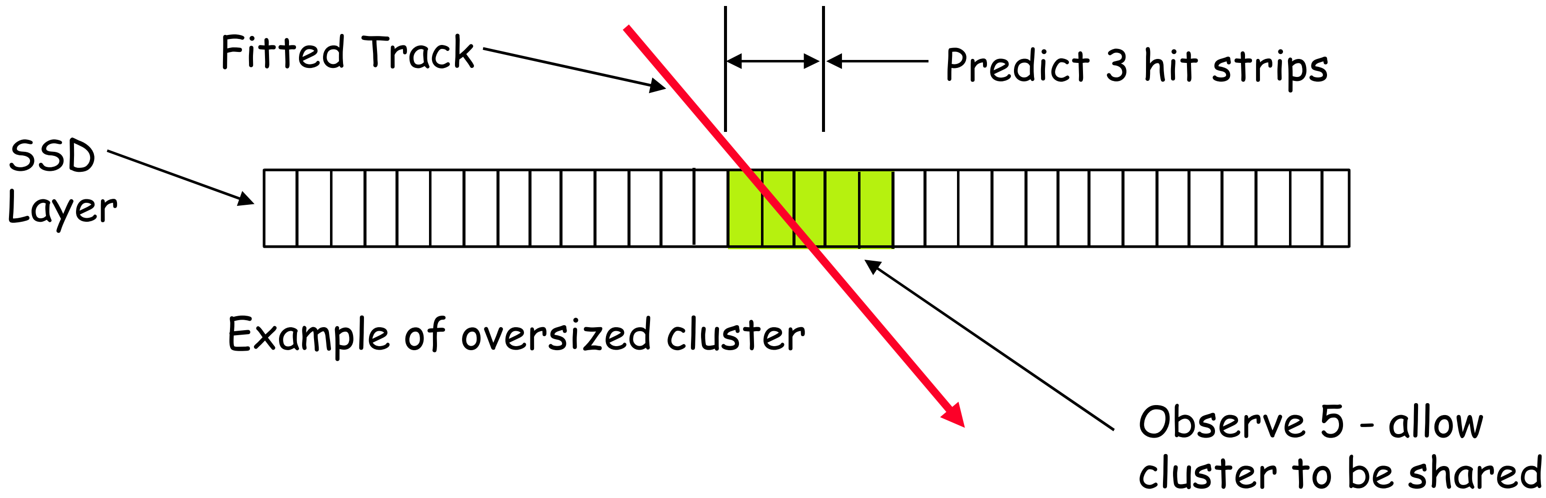
Combo PatRec, cont.

<u>Hit Flagging</u> (Allowed Hit Sharing)

In order not to find the same track, at most 6 (default) clusters can be shared

The first X and Y cluster (nearest the conversion point) is always allowed to be shared

Subsequent clusters are shared depending on the cluster width and the track slopes.



Combo Pat. Rec. Control Parameters

Control Name	Internal Variable	Def. Value	Description
MinEnergy	m_minEnergy	30.	Min. energy to use for setting search regions
SigmaCut	m_sigmaCut	9.0	Sigma cut for picking up points
FirstTrkEnergyFrac	m_1stTkrEFrac	.8	First track energy fraction
MinTermHitCount	m_termHitCnt	16	Min. no. of hits on best track to terminate search
MaxNoCandidates	m_maxCandidates	10	Max. allowed number of candidate tracks
MaxChisq	m_maxChiSqCut	40.	Max allow Combo Pat. Rec. Chisq. (1st fit)
NumSharedFirstClusters	m_hitShares	6	Number of first clusters which can be shared
MaxNumberTrials	m_maxTrials	50	Max. number of trial candidates to test
FoVLimit	m_PatRecFoV	.19	Minimum cos(theta) for track trials
MinCosKink	m_minCosKink	.7	Minimum cos(theta) for a track kink
MaxTripletRes	m_maxTripRes	30.	Max. un-normalized residual for first 3 TkrPoints
UniqueHits	m_minUniHits	4	Min. number of unique hits required on a track
MinPatRecQual	m_minQuality	10	Min. Track PR quality to accept
MaxFirstGaps	m_maxFirstGaps	1	Max. number of allowed gaps in the first 3 XY points
MaxTotalGaps	m_maxTotalGaps	2	Max. total number of XY gaps in the track
EnergyType	m_energyType	Default	Energy types: DEFAULT, CALONLY, USER, MC Default = Tkr+Cal with constraint, others self explanatory and over ride resetting energy at later stages USER: MinEnergy is used
Direction	m_searchDirection	Downwards	Direction in which to search for tracks: TopDown or BottomUp
AddLeadingHits	m_leadingHits	true	Flag to include leading hit (clusters) on the track
ReverseLayerPenalty	m_reverseLayerPenalty	1	Don't search all the way to the top
MaxDeltaFirstLayer	m_maxDeltaFirstLayer	1	If one long track has been found, don't look more than delta layers away for any others

A Kalman Filter for GLAST

- What a Kalman Filter is and how it works.
- Overview of Implementation in GLAST
- Validation (or Sea Trials)

Reference: Data Analysis Techniques in HEP by Fruthwirth et al, 2000

Kalman Filter

The Kalman filter process is a successive approximation scheme to estimate parameters

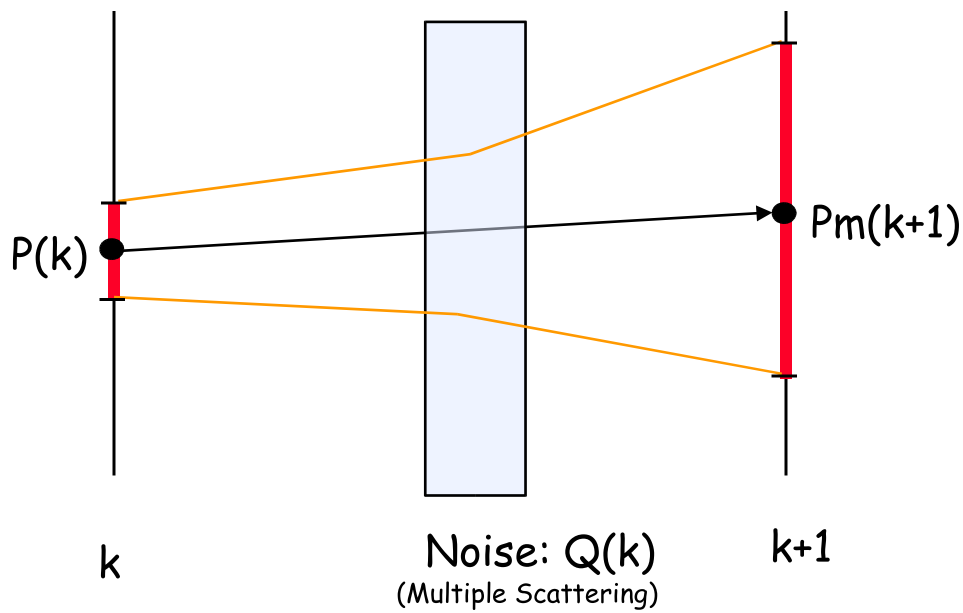
Simple Example: 2 parameters - intercept and slope: $x = x_0 + S_x * z$; $P = (x_0 , S_x)$

Errors on parameters x_0 & S_x (covariance matrix): $C = \begin{bmatrix} Cx\text{-}x & Cx\text{-}s \\ Cs\text{-}x & Cs\text{-}s \end{bmatrix}$

$Cx\text{-}x = \langle (x-x_m)(x-x_m) \rangle$

In general

$C = \langle (P - P_m)(P-P_m)^T \rangle$



Propagation:

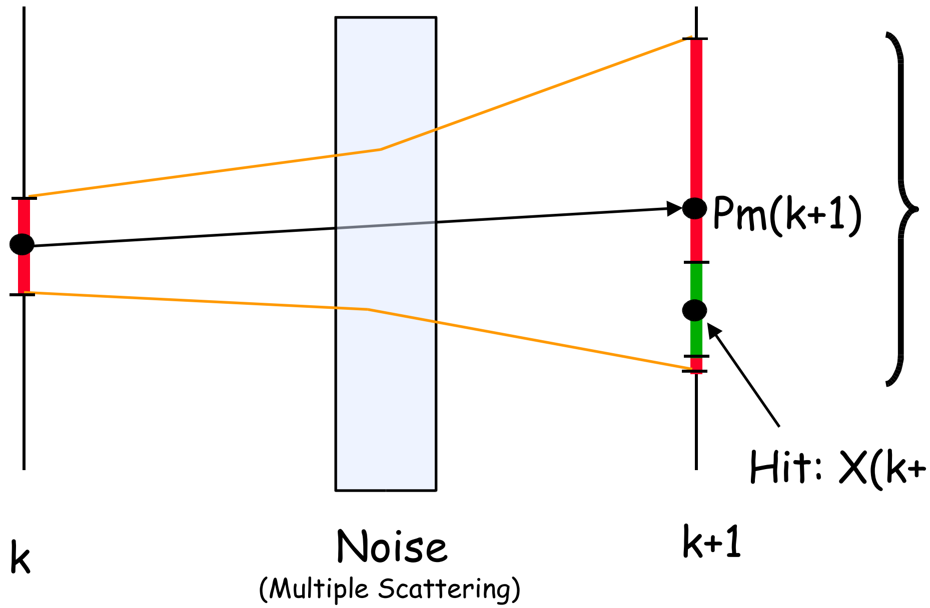
$x(k+1) = x(k)+Sx(k)*(z(k+1)-z(k))$

$Pm(k+1) = F(\delta z) * P(k)$ where

$$F(\delta z) = \begin{bmatrix} 1 & z(k+1)-z(k) \\ 0 & 1 \end{bmatrix}$$

$Cm(k+1) = F(\delta z) *C(k) * F(\delta z)^T + Q(k)$

Kalman Filter (2)



Form the weighted average of the k+1 measurement and the propagated track model: Weights given by inverse of Error Matrix: C^{-1}

$$P(k+1) = \frac{Cm^{-1}(k+1)*Pm(k+1) + V^{-1}(k+1)*X(k+1)}{Cm^{-1}(k+1) + V^{-1}(k+1)} \quad \text{and} \quad C(k+1) = (Cm^{-1}(k+1) + V^{-1}(k+1))^{-1}$$

Now its repeated for the k+2 planes and so - on. This is called FILTERING - each successive step incorporates the knowledge of previous steps as allowed for by the NOISE and the aggregate sum of the previous hits.

Kalman Filter (3)

We start the FILTER process at the conversion point

BUT... We want the best estimate of the track parameters
at the conversion point.

Must propagate the influence of all the subsequent Hits backwards to the beginning of the track - Essentially running the FILTER in reverse.

This is call the SMOOTHER & the linear algebra is similar.

Residuals & χ^2:

Residuals: $r(k) = X(k) - Pm(k)$

Covariance of r(k): $Cr(k) = V(k) - C(k)$

Then: $\chi^2 = r(k)^T Cr(k)^{-1} r(k)$ for the k^{th} step

GLAST Kalman Filter Organization

- Kalman Filter Details
- Define
  - State and measurement vector
  - Covariance matrices
- Abstract definitions for:
  - **F**, **Q**, **H** matrices
- Kalman Filter Utility Class
  - Implements Kalman Filter equations
  - Operates on vectors and matrices passed to it
    - Does not need to know actual definitions
    - Could be used for ANY experiment (FLW)
- Gaudi Tool to drive the track fit
  - Choose specific implementations of abstract state, measurement, etc, vectors and matrices
  - Provides freedom to switch fit conditions via job options parameters
    - Fit measured only, or measured and non-measured
    - Turn on or off multiple scattering matrices
    - Change energy loss mechanisms
    - Change implementations of measured errors
    - etc.
  - Implement several "special" run modes

- Basic Definitions
  - State Vector $\mathbf{p} = \{x, m_x, y, m_y\}$
  - Covariance matrix for $\mathbf{p}$: C
  - Measurement vector $\mathbf{m}$ = {x or y} or {x,y}
  - Error matrix for $\mathbf{m}$: W
  - Transport matrix $\mathbf{F}_k$: takes $\mathbf{p}$ from k-1 to k
  - "Process Noise" matrix $\mathbf{Q}$ – the multiple scattering matrix
  - Projection matrix $\mathbf{H}$

- Prediction
  - State Vector: $\mathbf{p}_{k|k-1} = \mathbf{F}_k \mathbf{p}_{k-1|k-1}$
  - Covariance: $C_{k|k-1} = \mathbf{F}_k C_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k$

- Filter Step
  - State Vector: $\mathbf{p}_{k|k} = C_{k|k}[C_{k|k-1}^{-1}\mathbf{p}_{k|k-1} + \mathbf{H}_k^T \mathbf{W}_k \mathbf{m}_k]$
  - Covariance: $C_{k|k} = [C_{k|k-1}^{-1} + \mathbf{H}_k^T \mathbf{W}_k \mathbf{H}_k]^{-1}$

- Smooth Step
  - State Vector: $\mathbf{p}_{k|n} = \mathbf{p}_{k|k} - A_k(\mathbf{p}_{k+1|k} - \mathbf{p}_{k+1|n})$
  - Covariance: $C_{k|n} = C_{k|k} - A_k(C_{k+1|k} - C_{k+1|n})A_k^T$
  - With: $A_k = C_{k|k}\mathbf{F}_{k+1}^T C_{k+1|k}^{-1}$

Kalman Filter Control Parameters

Control Parameter	Internal variable name	Default Value	Description
HitEnergyType	m_HitEnergyType	"eRadLoss"	Determines energy loss mechanism used during the track fit: radiation energy loss (default), Bethe-Bloch, or use Monte Carlo particle energy
ParticleName	m_ParticleName	"e-"	Used to extract particle mass for energy loss mechanism
DoMultScatMat	m_MultScatMat	true	Diagnostic fitting: turn off multiple scattering in the track fits
FitMeasHitOnly	m_FitMeasOnly	true	Selects "projection" matrix, used to turn on fitting of measured direction only, or both measured and non-measured directions
MeasHitErrorType	m_HitErrorType	SlopeCorrected	Selects which measurements errors are used in fit
RunSmootherMemory	m_RunSmootherMemory	false	Turns on finding of the number of "segment" points
MinSegmentHits	m_MinSegmentHits	4	Sets the starting hit for segment hit finding
SegmentMinDelta	m_SegmentRes	0.005	Minimum change in track params χ^2 to stop search for "segment" point
RunRecursiveFit	m_RunRecursiveFit	false	Turns on Kalman Filter fit energy determination
FracDifference	m_fracDifference	0.01	Minimum change in track fit χ^2 to stop track fit energy determination
MaxIterations	m_maxIterations	5	Maximum iterations before stopping track fit energy determination
RunResidualsFit	m_RunResidualsFit	false	Turns on Kalman Filter residuals fitting

Combo Vertex

Algorithm: (The name is a misnomer - Historical)

- Determined where the vertex should fall in Z.
  - Trying to determine this by simply using the tracks is error prone since the tracks are often ~ parallel to each other
  - Use instead knowledge of relative conversion probabilities and the hit topologies at the heads of the track

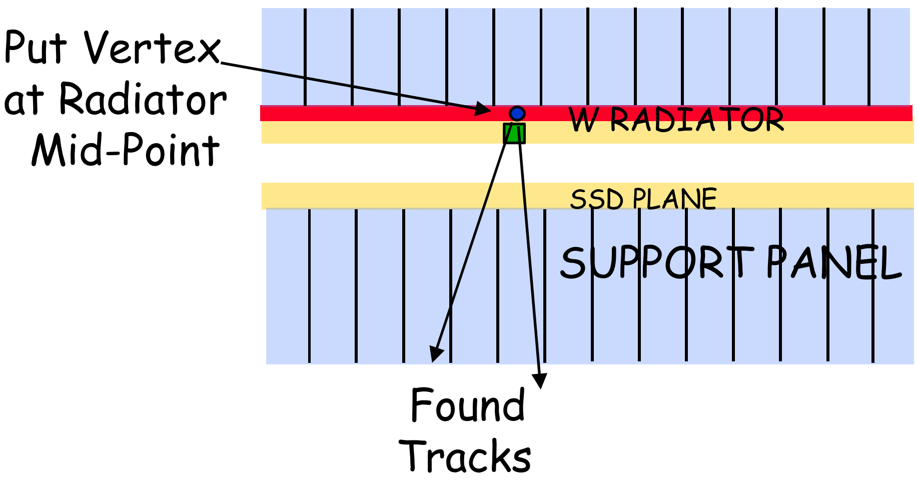
- Average the Track Parameters using their covariance matrices

- Compute the χ^2 for the pairing and form a "Vertex Quality" (computed using the extrapolation distances to the vertex and the χ^2 of the association)

- Loop over all possible 2-track pairings, selecting the pairing with the highest quality.

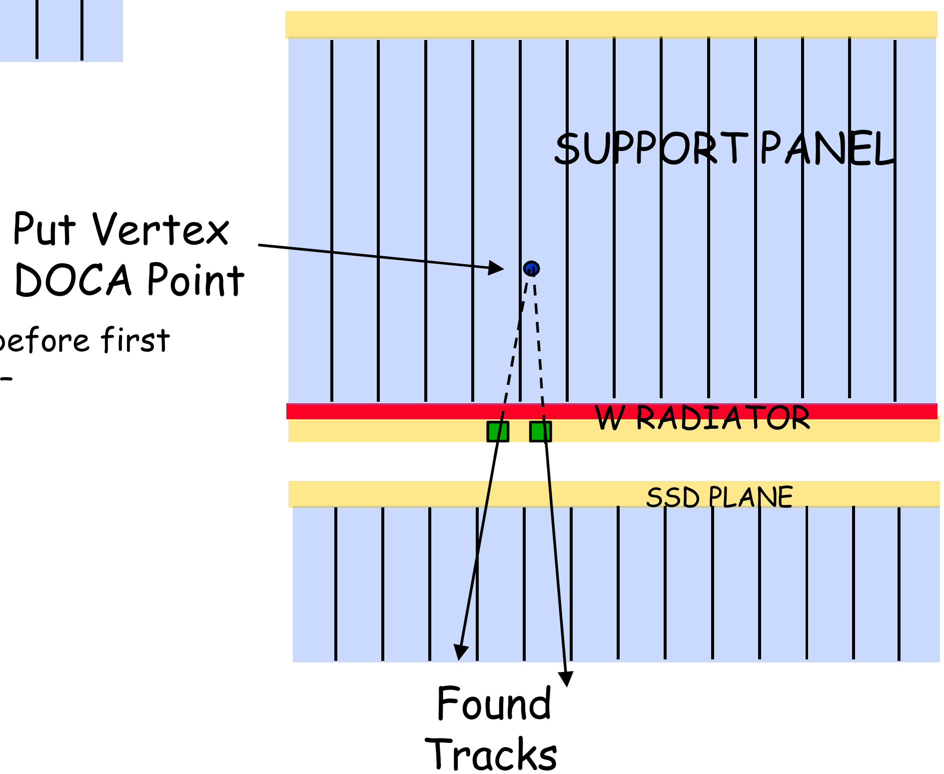
- If no pairing is possible – form a 1-track "vertex" with the parameters of that track.

Choosing the Z location for the Vertex



Preferred Solution

If 2 Tracks share the same first hit and the Cluster Size is no more then 2 Strips and the first hit directly proceed a W radiator
- ZVTX = middle of W Radiator prior to first hit



Next Best Solution

If DOCA location of 2 tracks lies before first hits but is after the next layer up –
- ZVTX = DOCA – Z

All Other Case

Put Vertex at Z location of start of the 1st Track

Combining the 2 Tracks

Multivariate Averaging: $P_{Pair} = \dfrac{C_1^{-1} \cdot P_1 + C_2^{-1} \cdot P_2}{C_1^{-1} + C_2^{-1}}$

$$P_{Pair} = (C_1^{-1} + C_2^{-1})^{-1} \cdot (C_1^{-1} \cdot P_1 + C_2^{-1} \cdot P_2)$$

$$C_{Pair} = (C_1^{-1} + C_2^{-1})^{-1}$$

where $\boldsymbol{P_i}$ are the parameter vectors of the combination(Pair) and tracks ($\boldsymbol{P_1}$ and $\boldsymbol{P_2}$) and $\mathbf{C}_i$ are the covariance matrices

And

$$\chi^2 = (P_1 - P_{Pair})^T C_{\mathrm{Res}1}^{-1} (P_1 - P_{Pair}) + (P_2 - P_{Pair})^T C_{\mathrm{Res}2}^{-1} (P_2 - P_{Pair})$$

where

$$C_{\mathrm{Res}1,2} = C_{1,2} - C_{Pair}$$

The parameter vectors are ***(x, S***

Control Parameters for Combo Vertexing

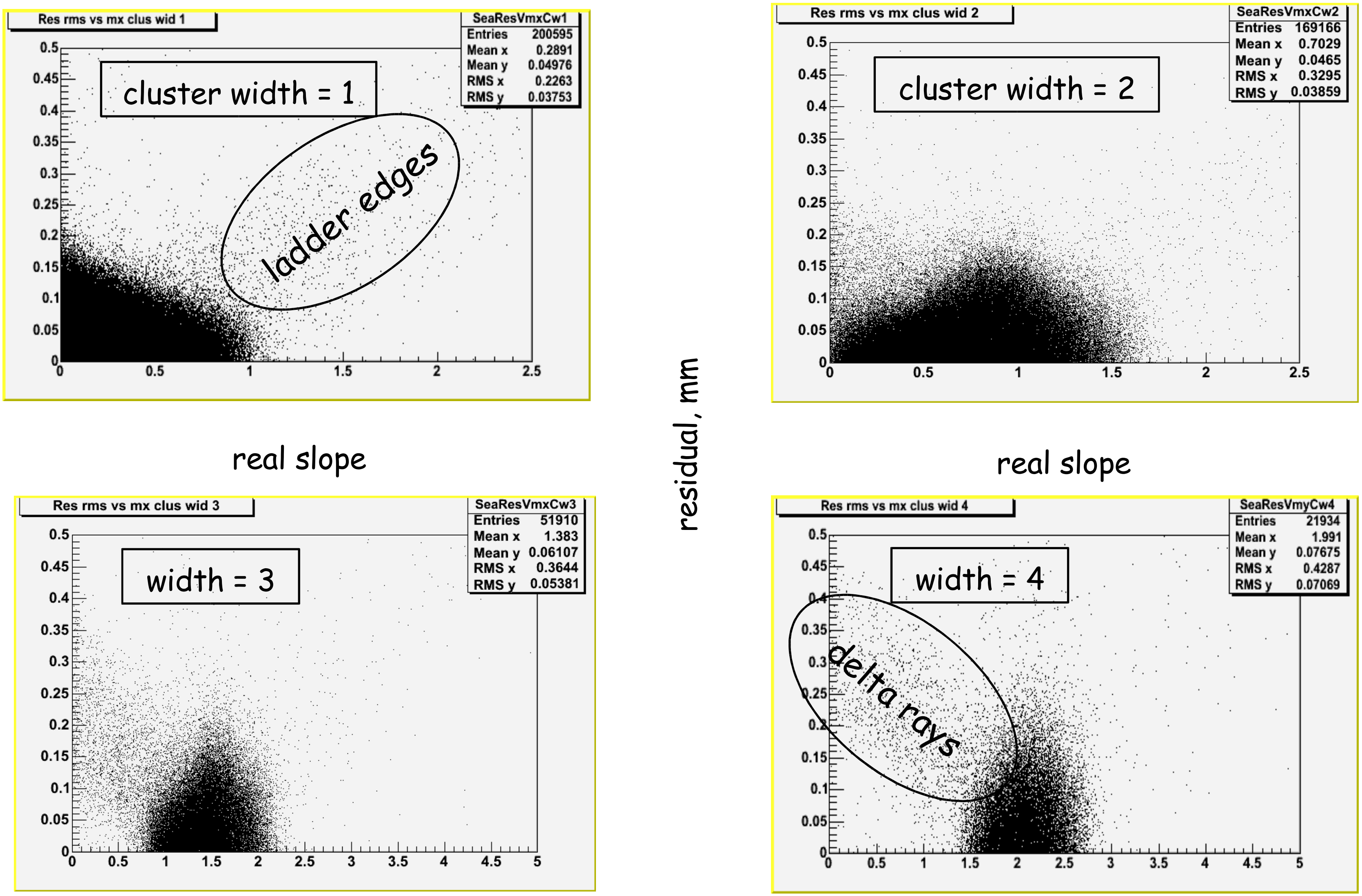
Control Name	Internal Variable	Def. Value	Description
MaxDOCA	m_maxDOCA	20	Maximum allowed distance of closest approach between prospective track pairs
MinQuality	m_minQuality	-100.	Min. vertex quality. The quality parameter is used to order possible pairings. Per-force it is < 0.

Track Residuals Fit

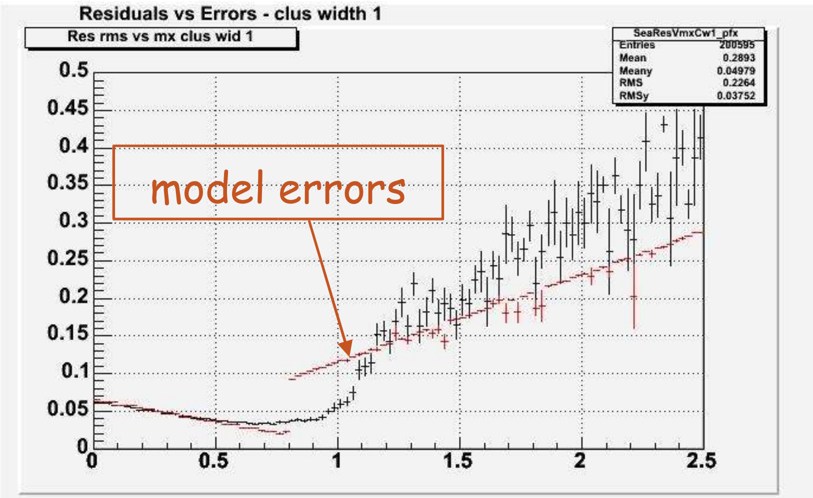
New tool from Tracy that allows tracks to be refit removing one hit at a time, allowing calculation of "unbiased" track hit residuals:

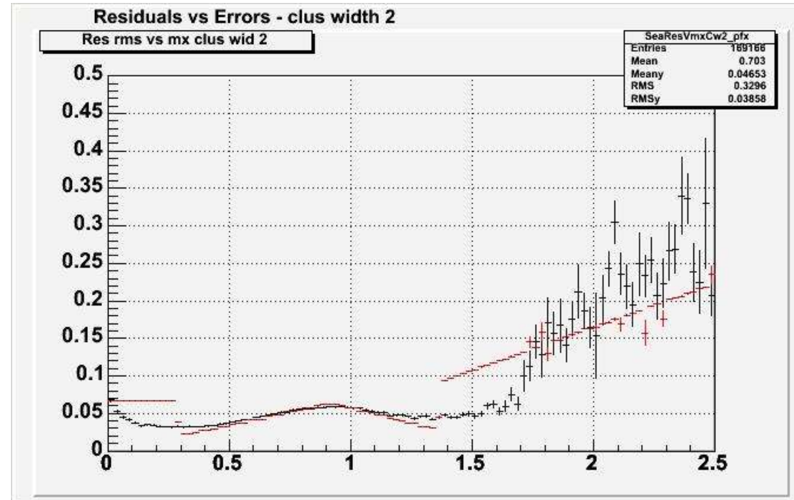
- Loop through hits:
  - Remove hit from the fitting procedure
    - Set bit
  - Refit track
    - Store Smoothed parameters for removed hits in the unused "RevFilter" parameters
- This method allows comparison with real data. No MC input necessary.
- To turn on this feature
  - ToolSvc.KalmanTrackFitTool.RunResidualsFit=true;

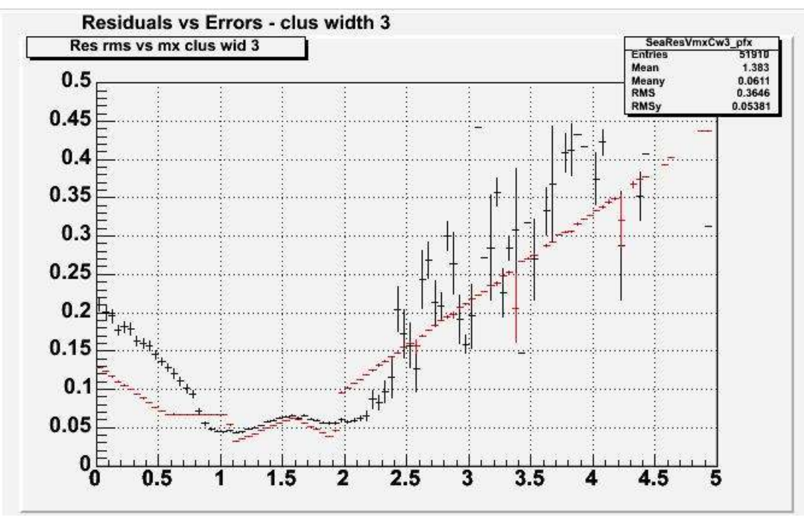
Deviations from fit, 10-GeV muons

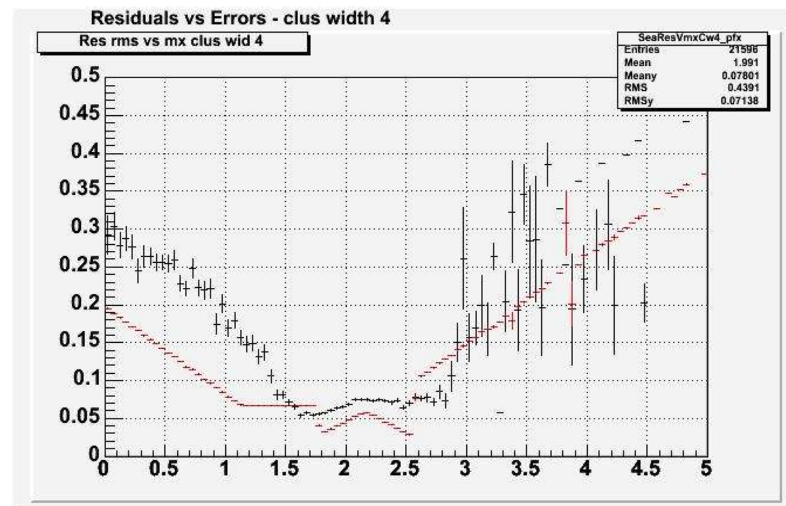


Profiles



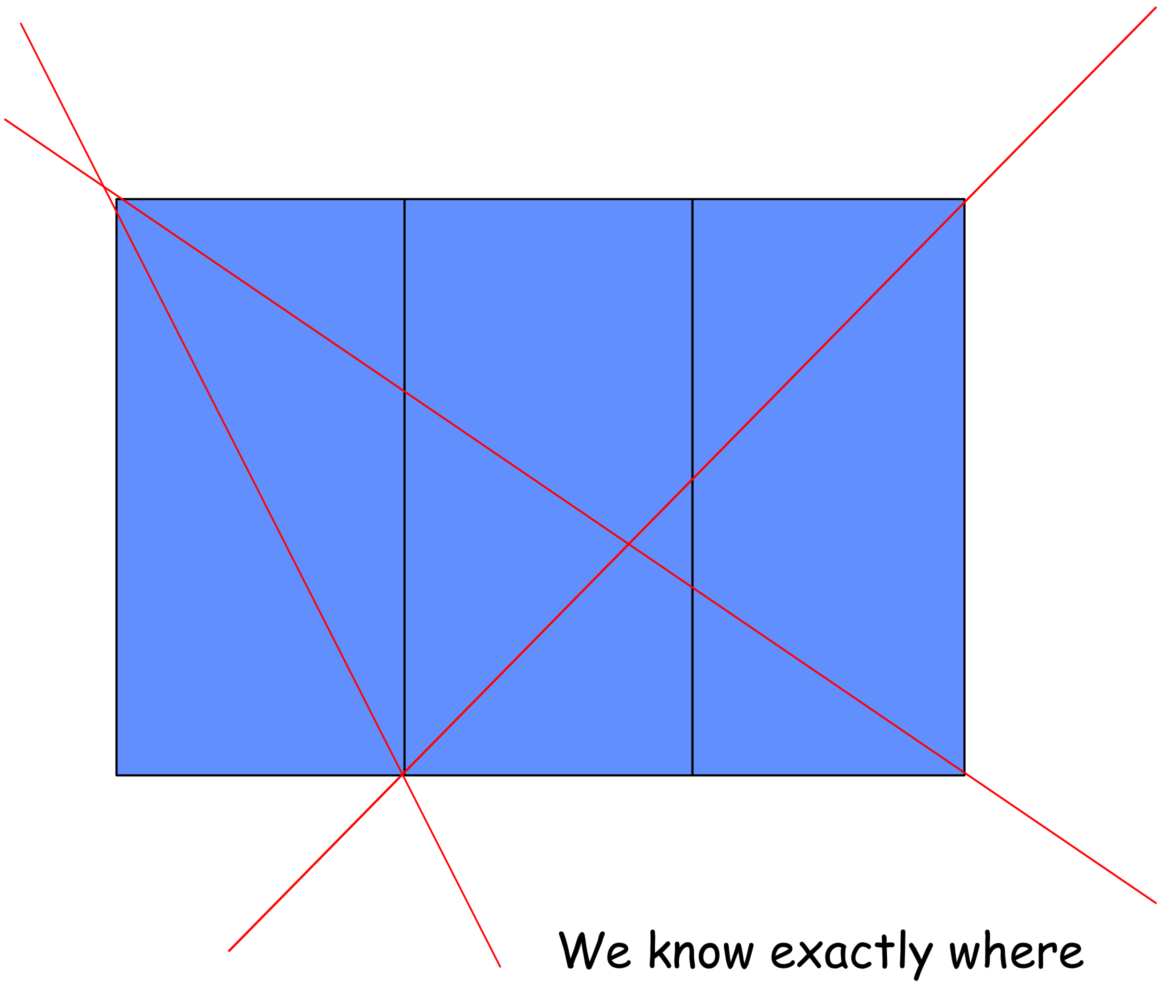






calculation uses RMS

What's happening? There are magic slopes…



We know exactly where
these tracks went.

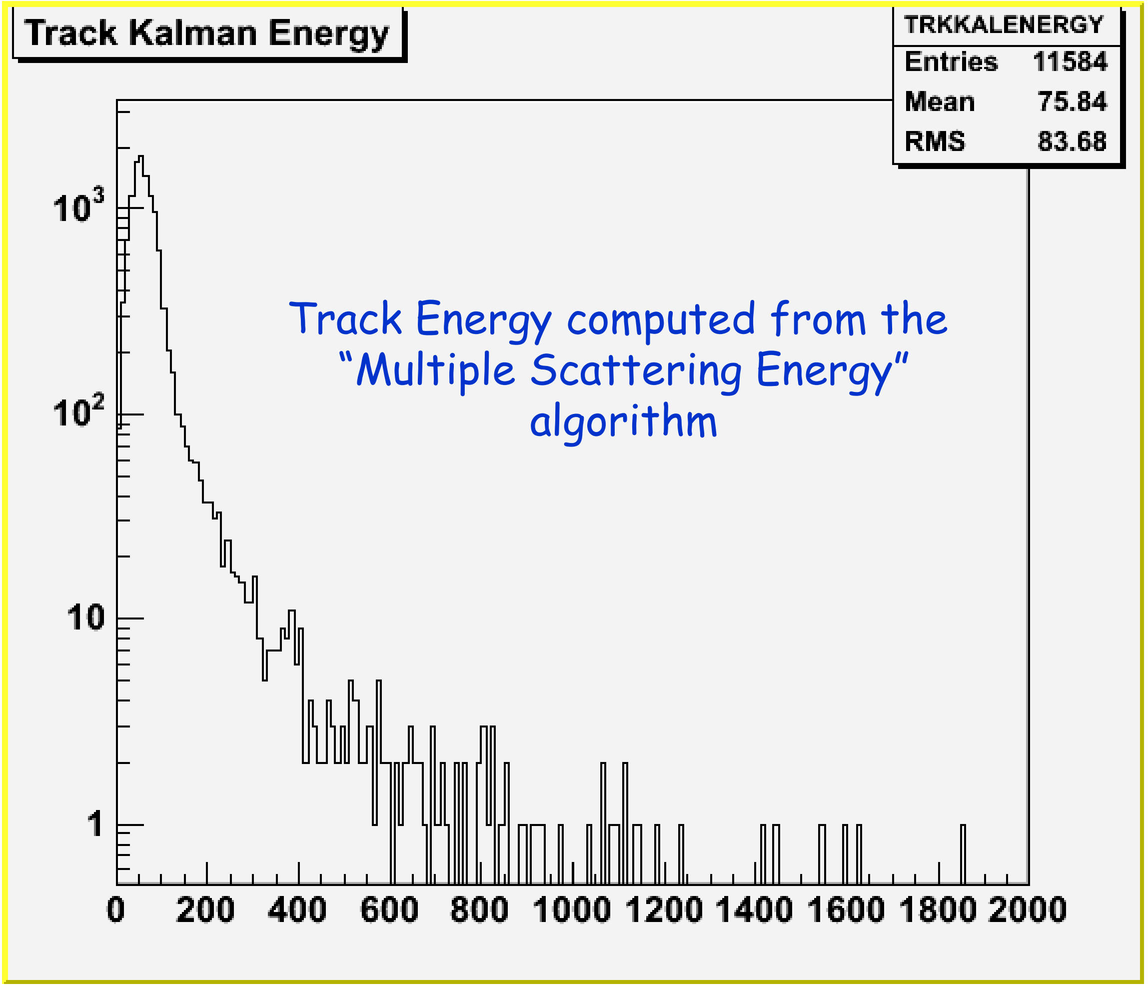
Track Fit Energy

Two Approaches

- Both estimate the track energy by using the multiple scattering
  - Will run out of steam around 1 GeV
- "Multiple Scattering Energy"
  - Computes the mean deflection angle between segments along a track
  - Converts this to an energy via PDB forumlae
- Kalman Filter
  - Uses the Kalman Filter track fit to estimate the track energy
    - Scale the track energy based on the χ^2 of the track fit
    - Refit the track with this new track energy
    - Repeat until the change in χ^2 is small
- Switch between the two:
  - ToolSvc.KalmanTrackFitTool.RunRecursiveFit=true;

Turns on Kalman Filter energy

Track Fit Energy



Number "Segment" Hits

- Question: how many hits on a track contribute to the track parameters (pointing) at the beginning of the track?

- Use the "smoothing" step of the Kalman Filter to answer this question:
  - Store the fully fit track parameters at the first hit
  - Run smoothing step from hit n on the track
  - Compare the resulting track parameters to the stored parameters
    - If change is negligibly small then stop
    - Hit n-1 is the last of the "segment" hits
  - If resulting change is "large", repeat procedure for hit n+1

- To turn on this feature:
  - ToolSvc.KalmanTrackFitTool.RunSmootherMemory=true;

- Control parameters:
  - ToolSvc.KalmanTrackFitTool.MinSegmentHits=4;
  - ToolSvc.KalmanTrackFitTool.SegmentMinDelta=0.005;

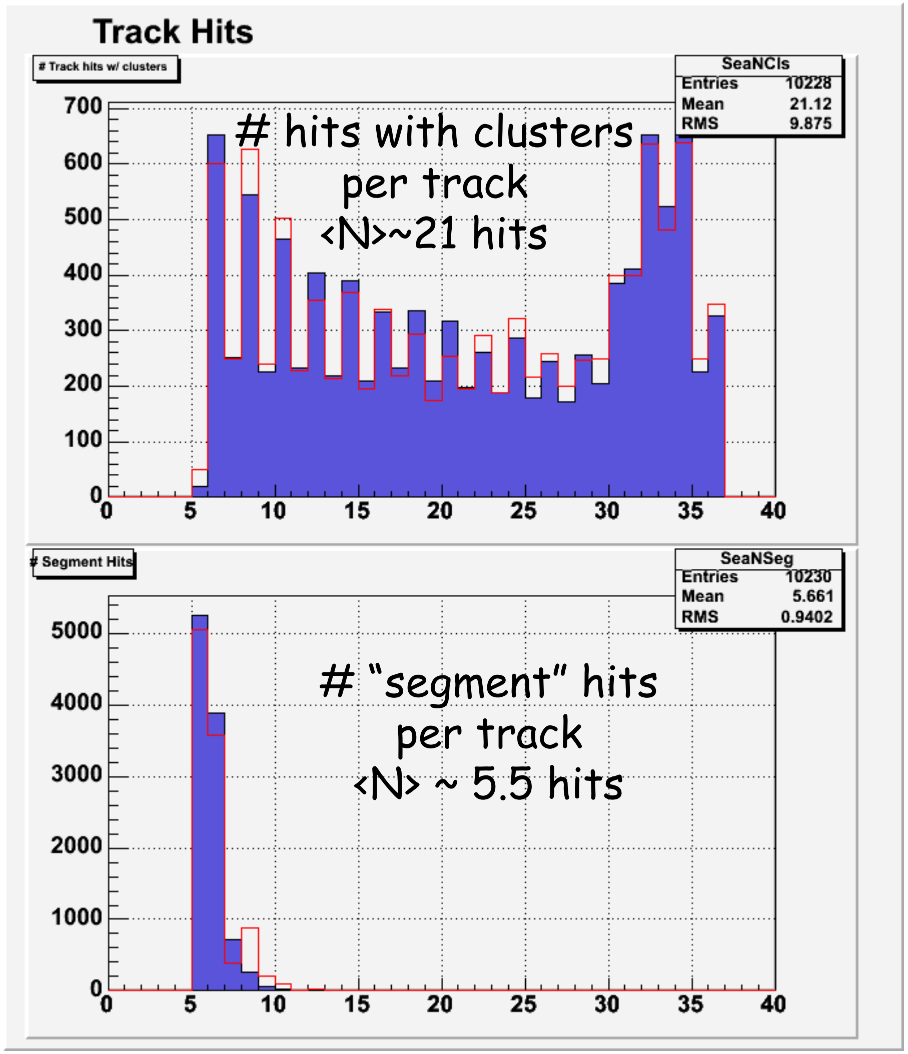
Sets first hit to start procedure

χ^2 change to stop procedure

Track Hits

100 MeV Muons – upper hemisphere

- Number of hits (with clusters) associated to tracks for 100 MeV muons
- Determine how many hits along the track contribute to the pointing at the first hit.
  - # "Segment" hits
  - Found by progressively running "smooth" phase of track fit until the hit found which has not significant effect on the fit parameters at the first track hit
- Plots contain
  - Blue fill/Black lines: MC Hits
  - Red lines: Combo Pat Rec Hits

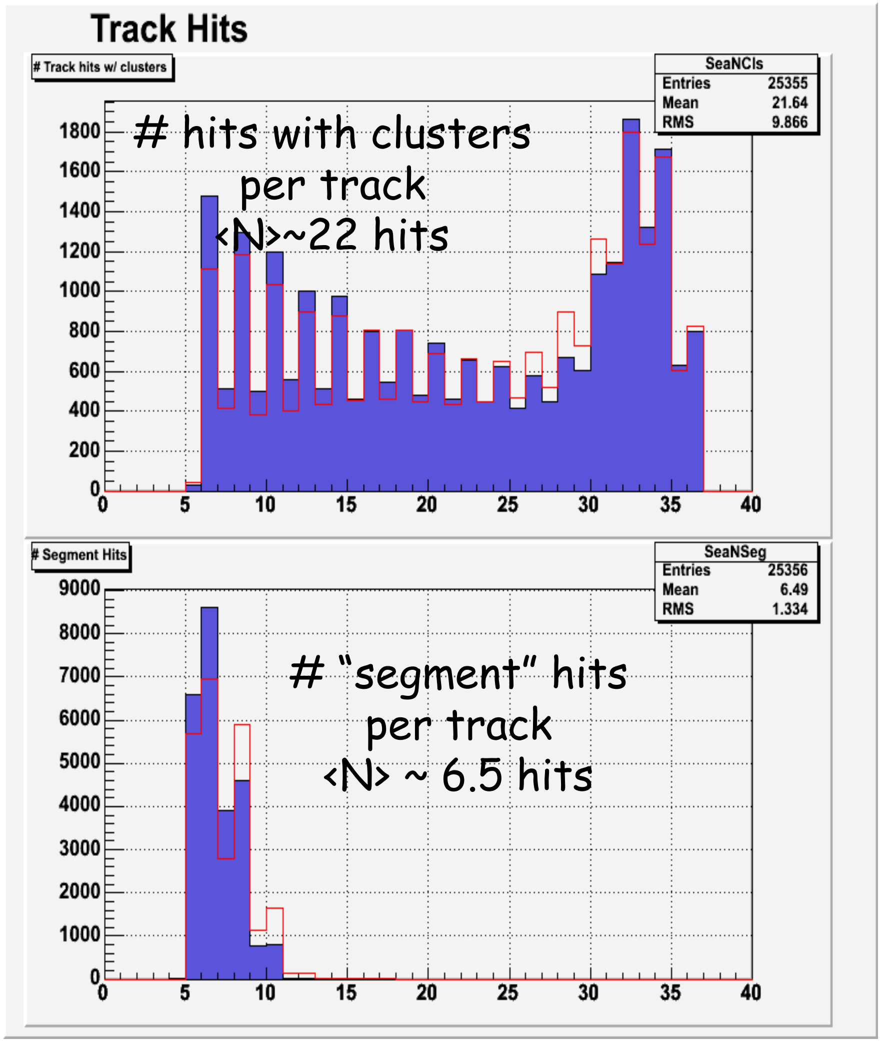


GLAST

Track Hits

500 MeV Muons – upper hemisphere

- Number of hits (with clusters) associated to tracks for 100 MeV muons
- Determine how many hits along the track contribute to the pointing at the first hit.
  - # "Segment" hits
  - Found by progressively running "smooth" phase of track fit until the hit found which has not significant effect on the fit parameters at the first track hit
- Plots contain
  - Blue fill/Black lines: MC Hits
  - Red lines: Combo Pat Rec Hits

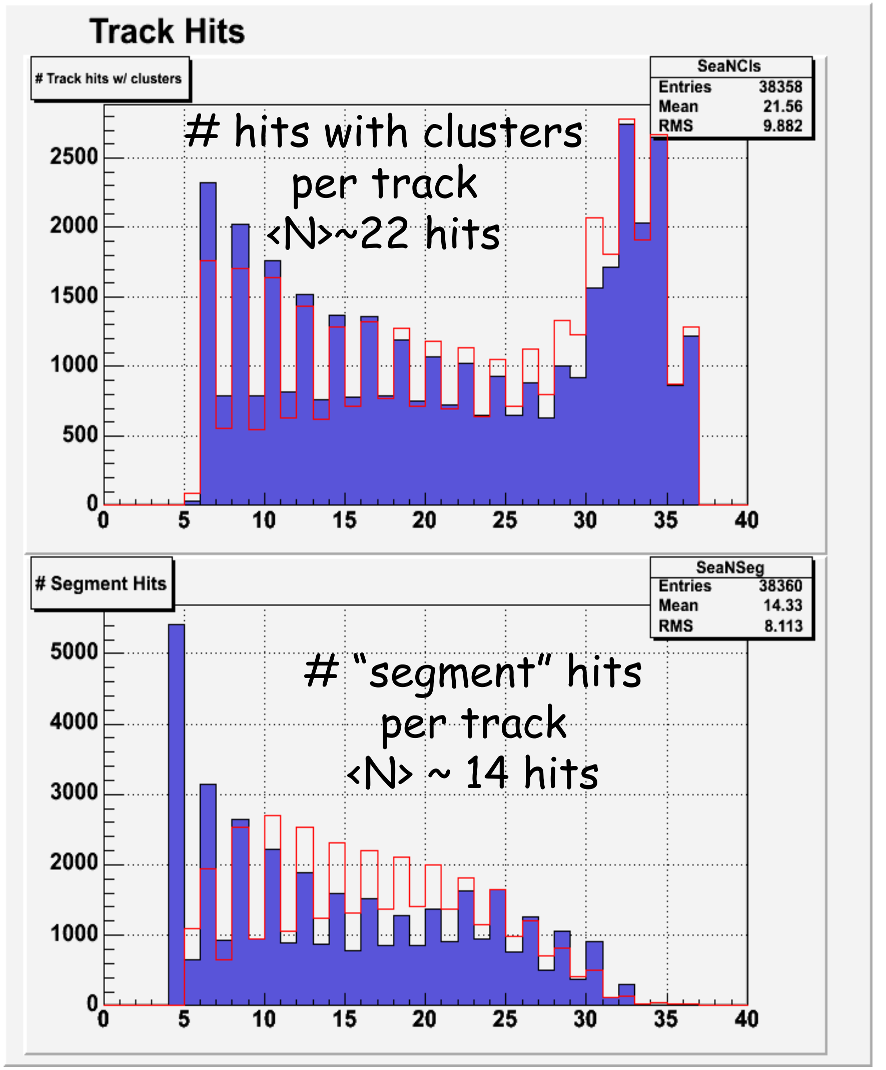


GLAST

Track Hits

10 GeV Muons – upper hemisphere

- Number of hits (with clusters) associated to tracks for 100 MeV muons
- Determine how many hits along the track contribute to the pointing at the first hit.
  - # "Segment" hits
  - Found by progressively running "smooth" phase of track fit until the hit found which has not significant effect on the fit parameters at the first track hit
- Plots contain
  - Blue fill/Black lines: MC Hits
  - Red lines: Combo Pat Rec Hits

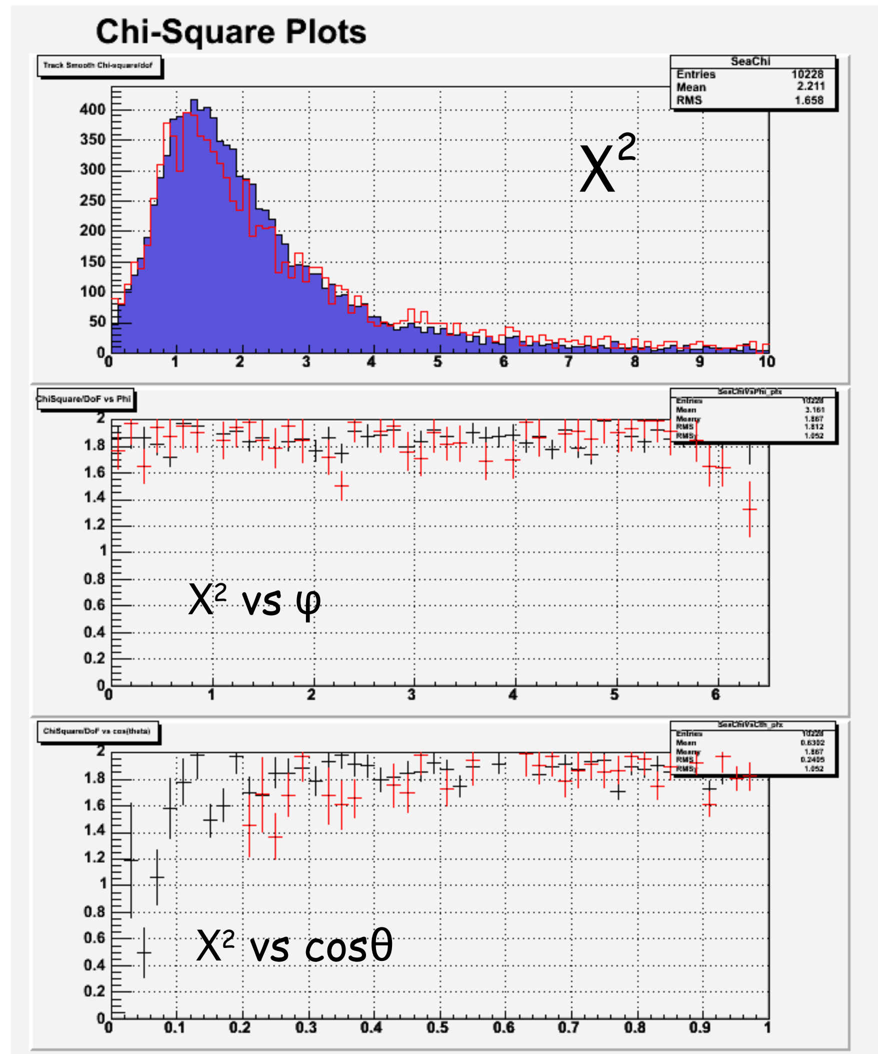


GLAST

Sea Trials – Track χ^2 Plots

100 MeV Muons – upper hemisphere

- Use χ^2 as test of Kalman Filter Fitting procedure
  - Show that we understand hit errors as a function of angle
- Plots contain
  - Blue fill/Black lines: MC Hits
  - Red lines: Combo Pat Rec Hits

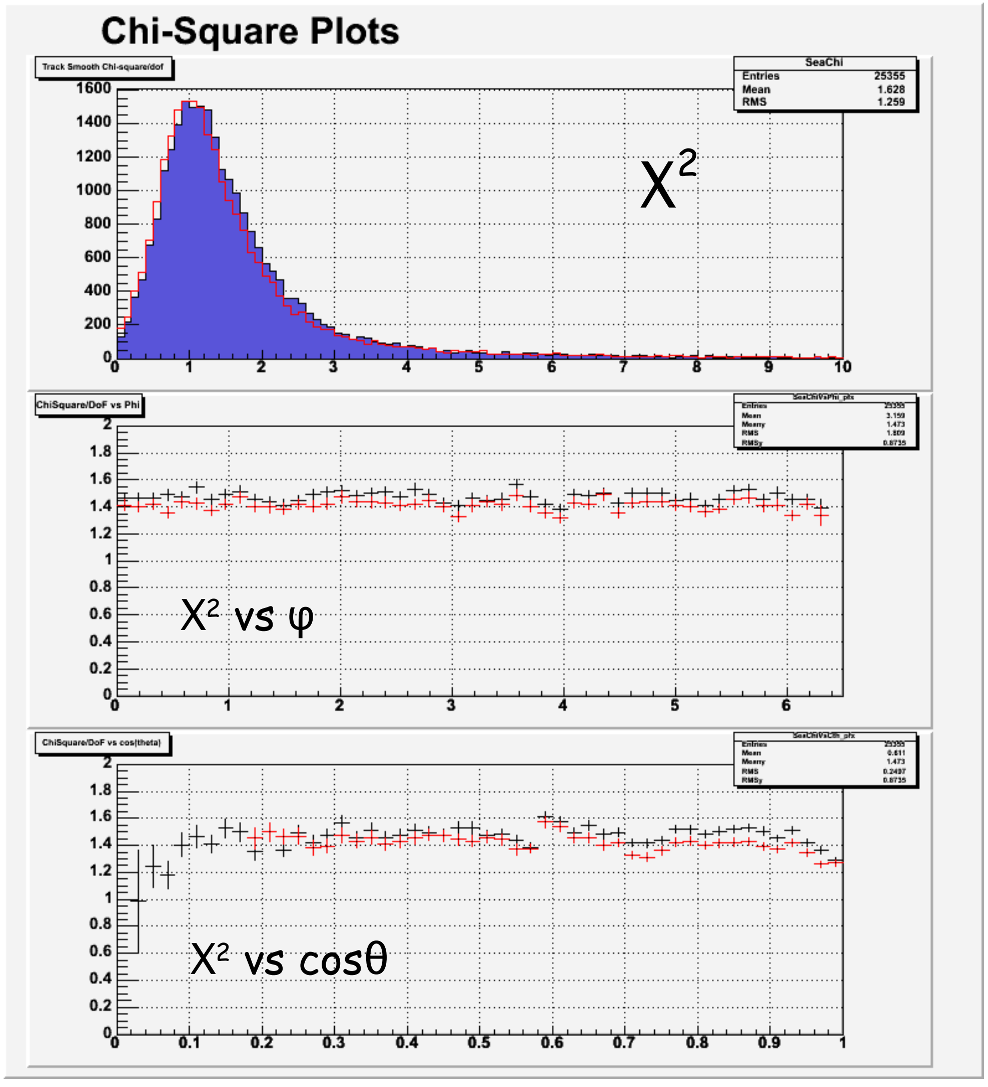


GLAST

Track χ^2 Plots

500 MeV Muons – upper hemisphere

- Use χ^2 as test of Kalman Filter Fitting procedure
  - Show that we understand hit errors as a function of angle
- Plots contain
  - Blue fill/Black lines: MC Hits
  - Red lines: Combo Pat Rec Hits

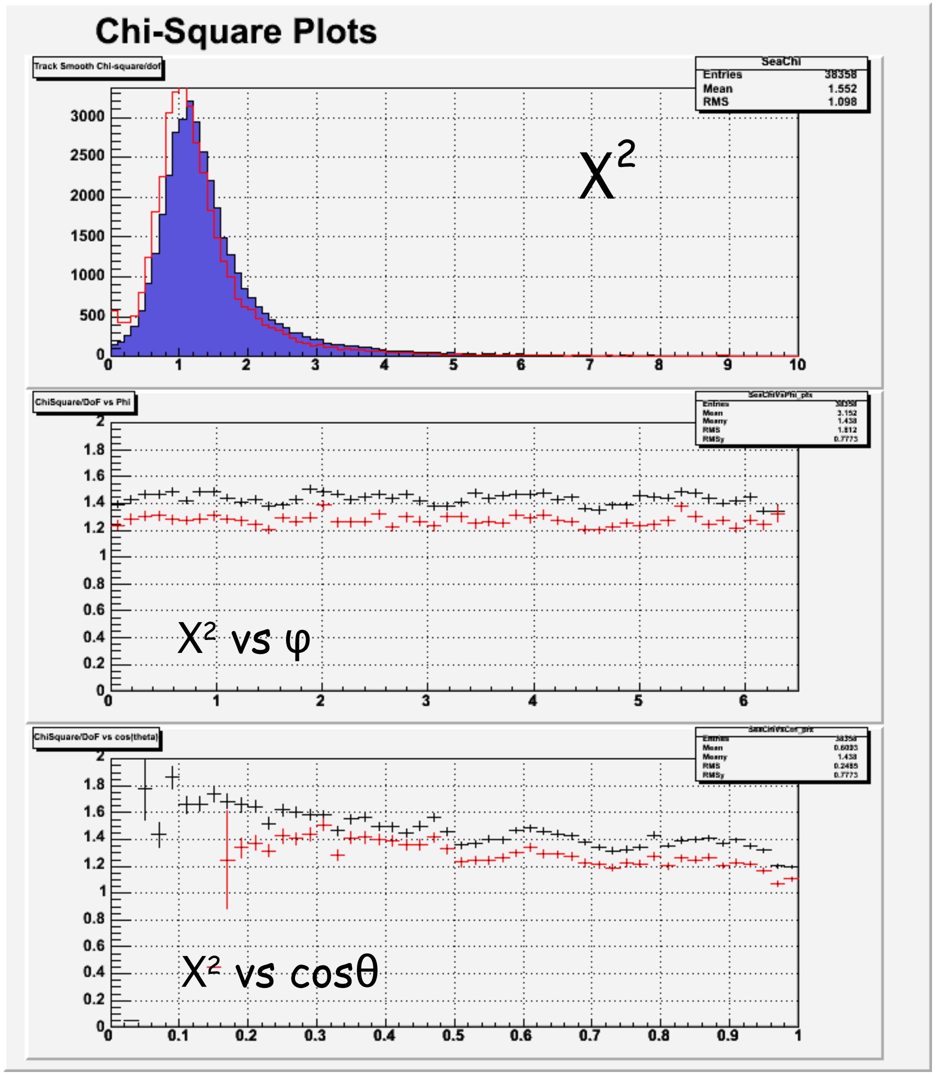


GLAST

Track χ^2 Plots

10 GeV Muons – upper hemisphere

- Use χ^2 as test of Kalman Filter Fitting procedure
  - Show that we understand hit errors as a function of angle
- Plots contain
  - Blue fill/Black lines: MC Hits
  - Red lines: Combo Pat Rec Hits

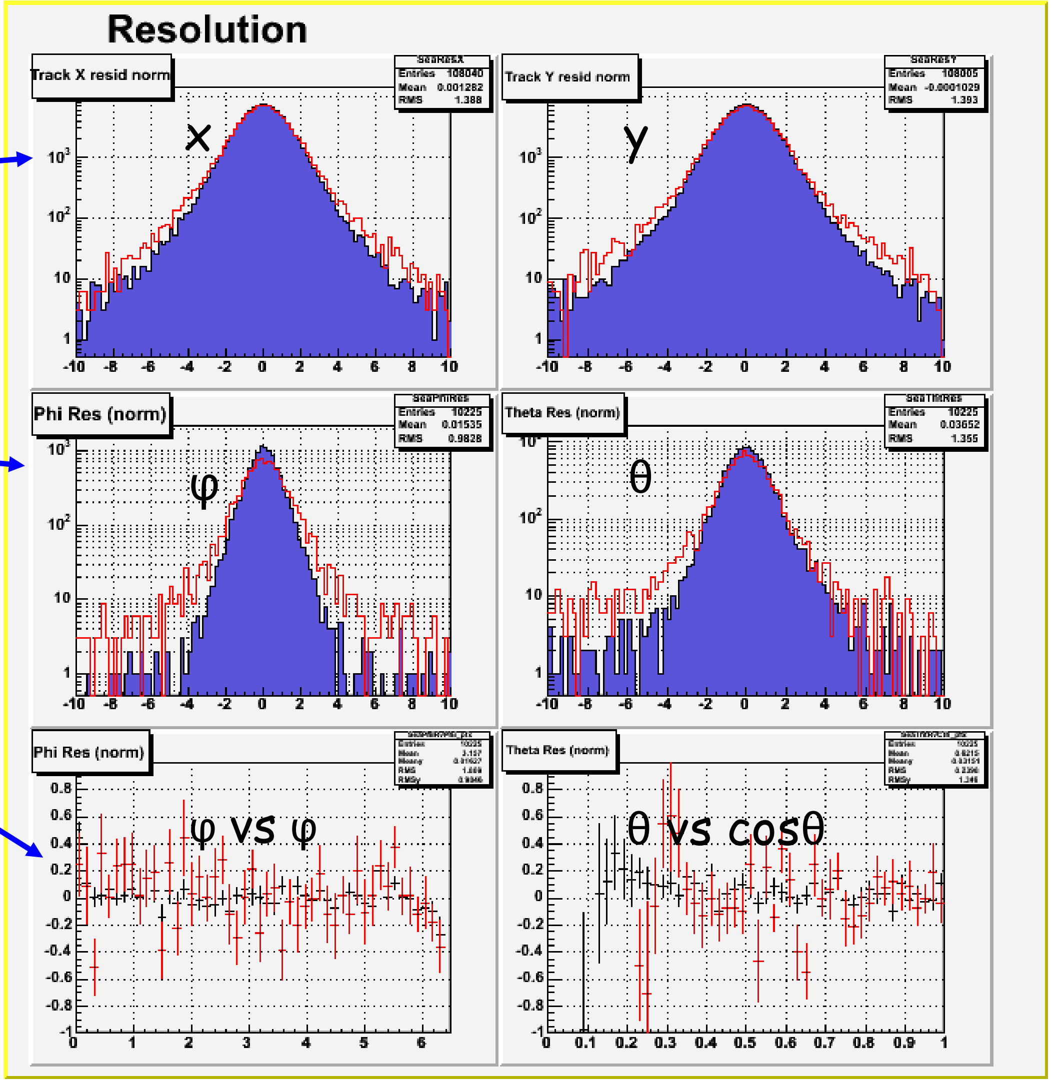


GLAST

Track Pointing

100 MeV Muons – upper hemisphere

- At 100 MeV multiple scattering effects dominate over the measurement errors
- Normalized Hit Residuals
  - Take the difference between the Fit predicted position and measured position
  - Normalize by measured error
- Normalized Fit Angles
  - Difference between track fit φ, θ and MC angles at the same point
  - Normalize by fit errors on φ and θ
- Normalized Angles Vs φ, cos θ
- Plots contain
  - Blue fill/Black lines: MC Hits
  - Red lines: Combo Pat Rec Hits

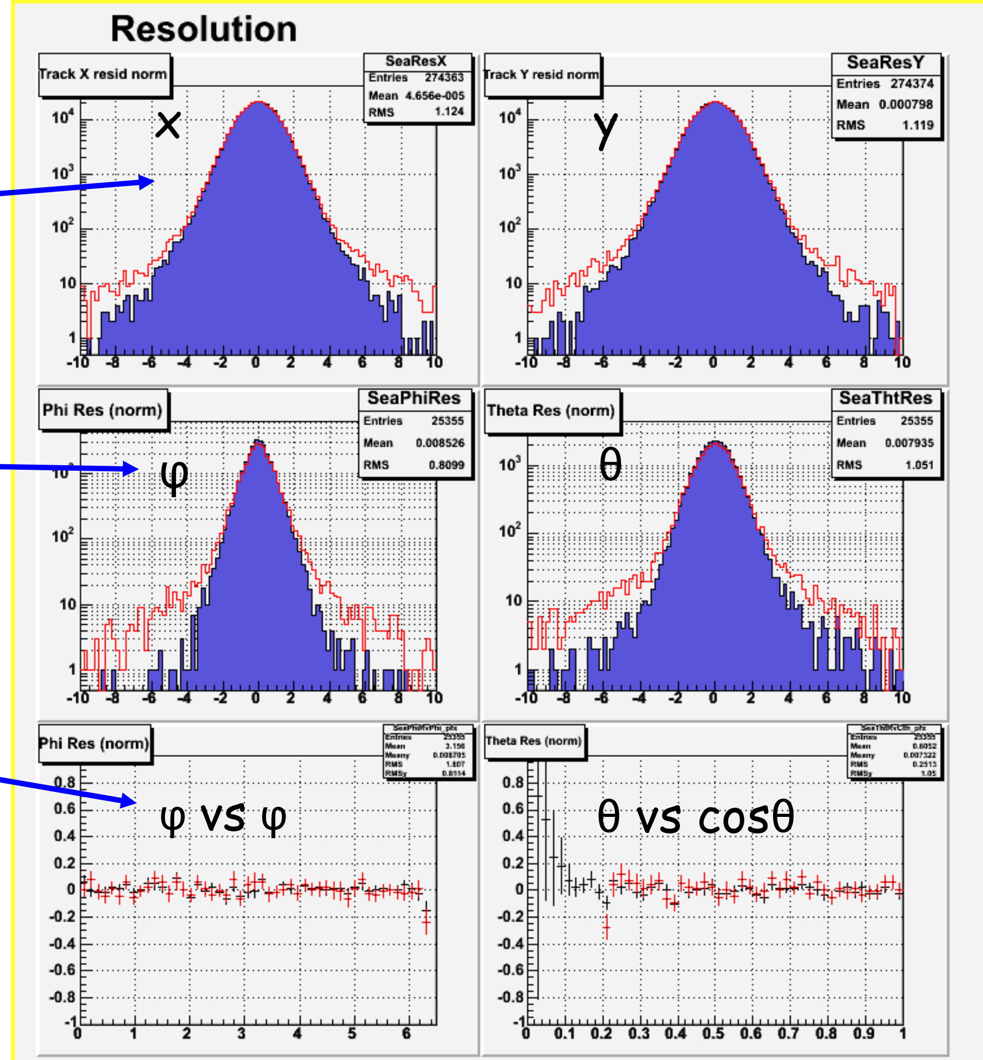


GLAST

Track Pointing

500 MeV Muons – upper hemisphere

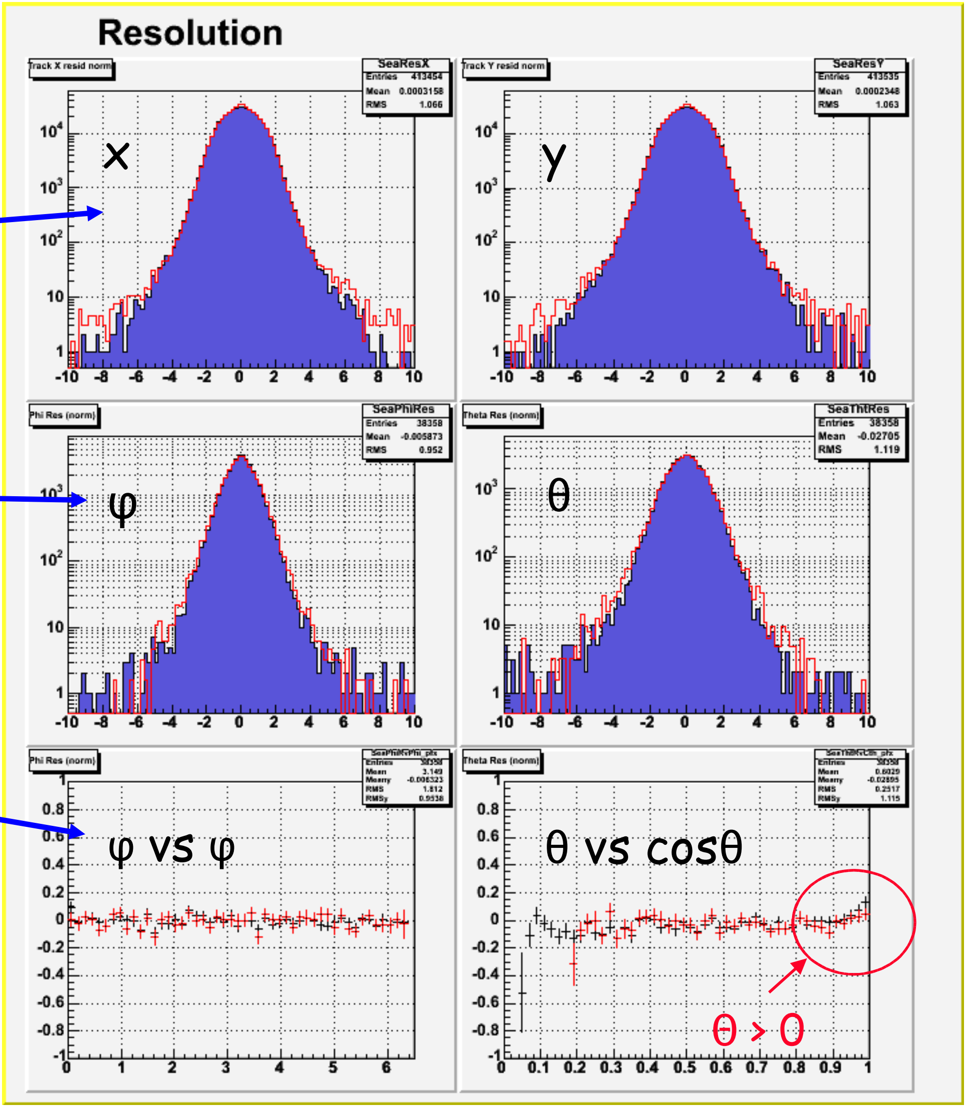
- At 500 MeV multiple scattering effects are commensurate with measurement errors
- Normalized Hit Residuals
  - Take the difference between the Fit predicted position and measured position
  - Normalize by measured error
- Normalized Fit Angles
  - Difference between track fit φ, θ and MC angles at the same point
  - Normalize by fit errors on φ and θ
- Normalized Angles Vs φ, cos θ
- Plots contain
  - Blue fill/Black lines: MC Hits
  - Red lines: Combo Pat Rec Hits



Track Pointing

10 GeV Muons – upper hemisphere

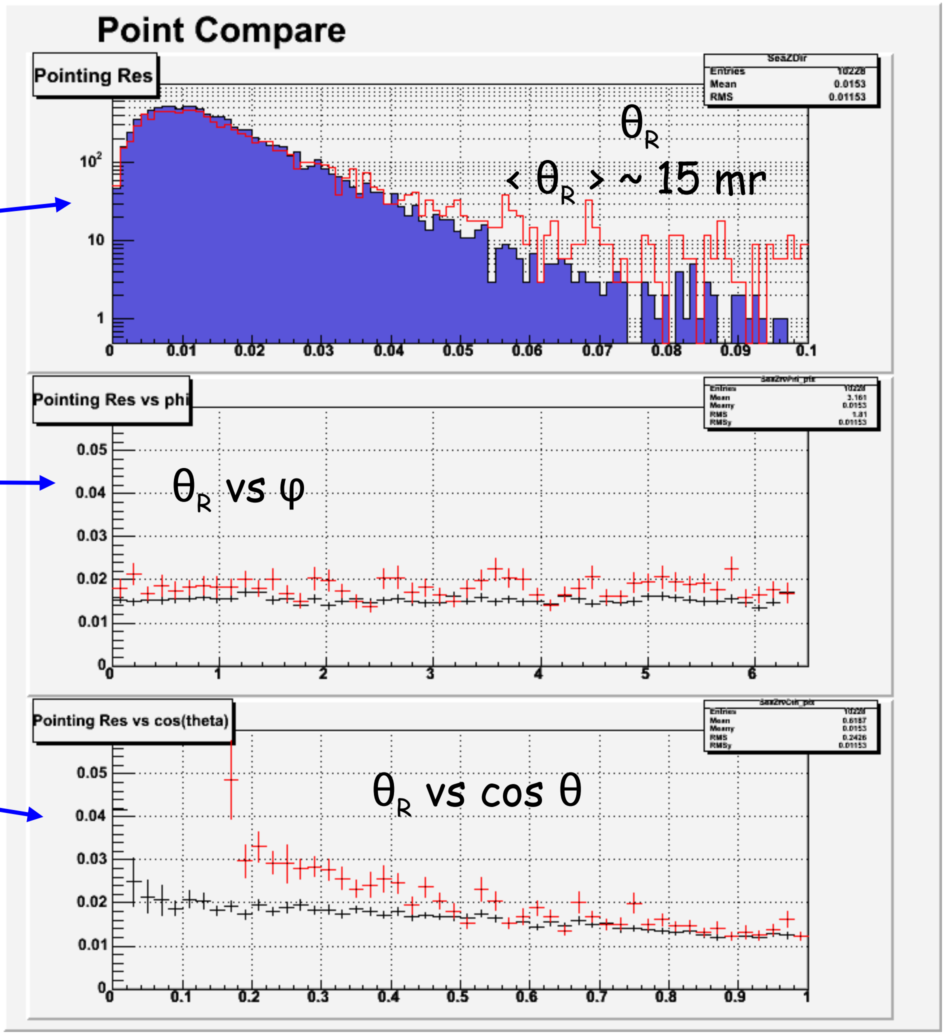
- At 10 GeV multiple scattering effects are minimal and we should be testing our sensitivity to measurement errors only
- Normalized Hit Residuals
  - Take the difference between the Fit predicted position and measured position
  - Normalize by measured error
- Normalized Fit Angles
  - Difference between track fit φ, θ and MC angles at the same point
  - Normalize by fit errors on φ and θ
- Normalized Angles Vs φ, cos θ
- Plots contain
  - Blue fill/Black lines: MC Hits
  - Red lines: Combo Pat Rec Hits



Track Pointing

100 MeV Muons – upper hemisphere

- At 100 MeV:
  - multiple scattering effects dominate over measurement errors
- Plot angle track makes to Monte Carlo trajectory at the first hit on the track
  - θ_R = arccos($\mathbf{u}_{fit} \cdot \mathbf{u}_{MC}$)
- Plot versus φ
- Plot versus cos θ
- Plots contain
  - Blue fill/Black lines: MC Pat Rec
  - Red lines: Combo Pat Rec

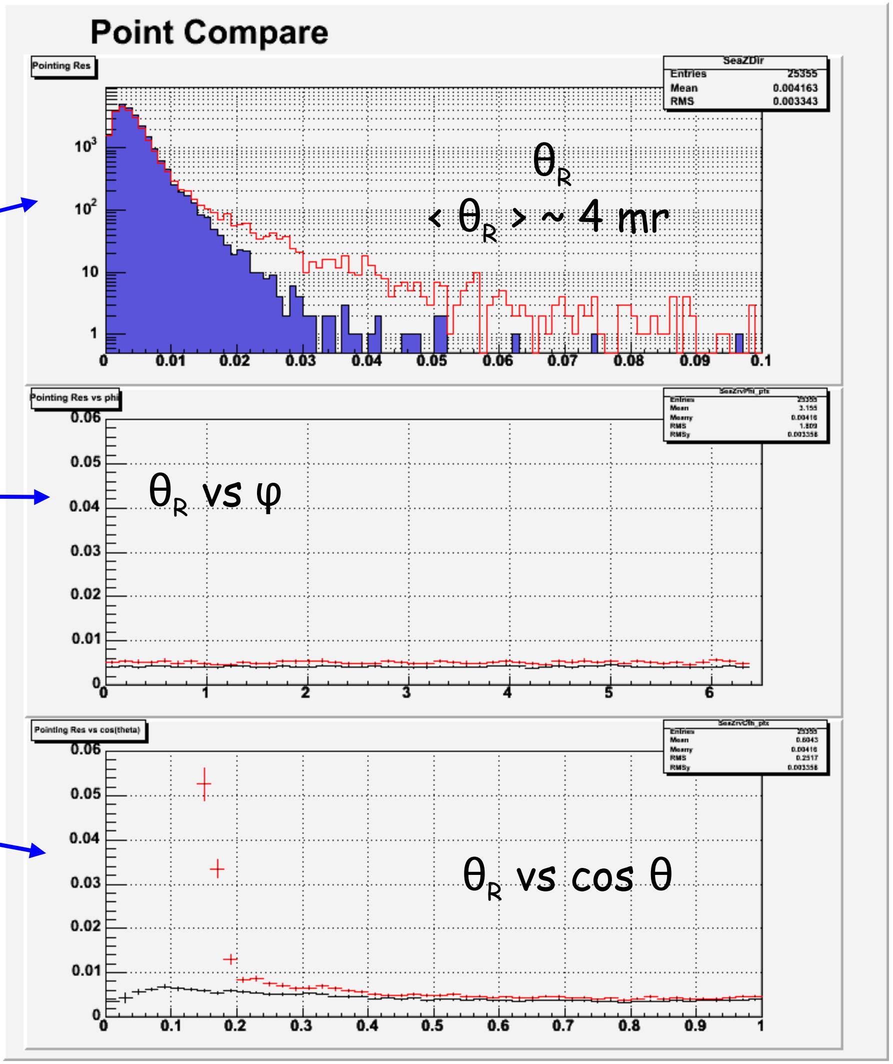


GLAST

Track Pointing

500 MeV Muons – upper hemisphere

- At 500 MeV:
  - multiple scattering effects are commensurate with measurement errors
- Plot angle track makes to Monte Carlo trajectory at the first hit on the track
  - $\theta_R = \arccos(\mathbf{u}_{fit} \cdot \mathbf{u}_{MC})$
- Plot versus φ
- Plot versus cos θ
- Plots contain
  - Blue fill/Black lines: MC Pat Rec
  - Red lines: Combo Pat Rec

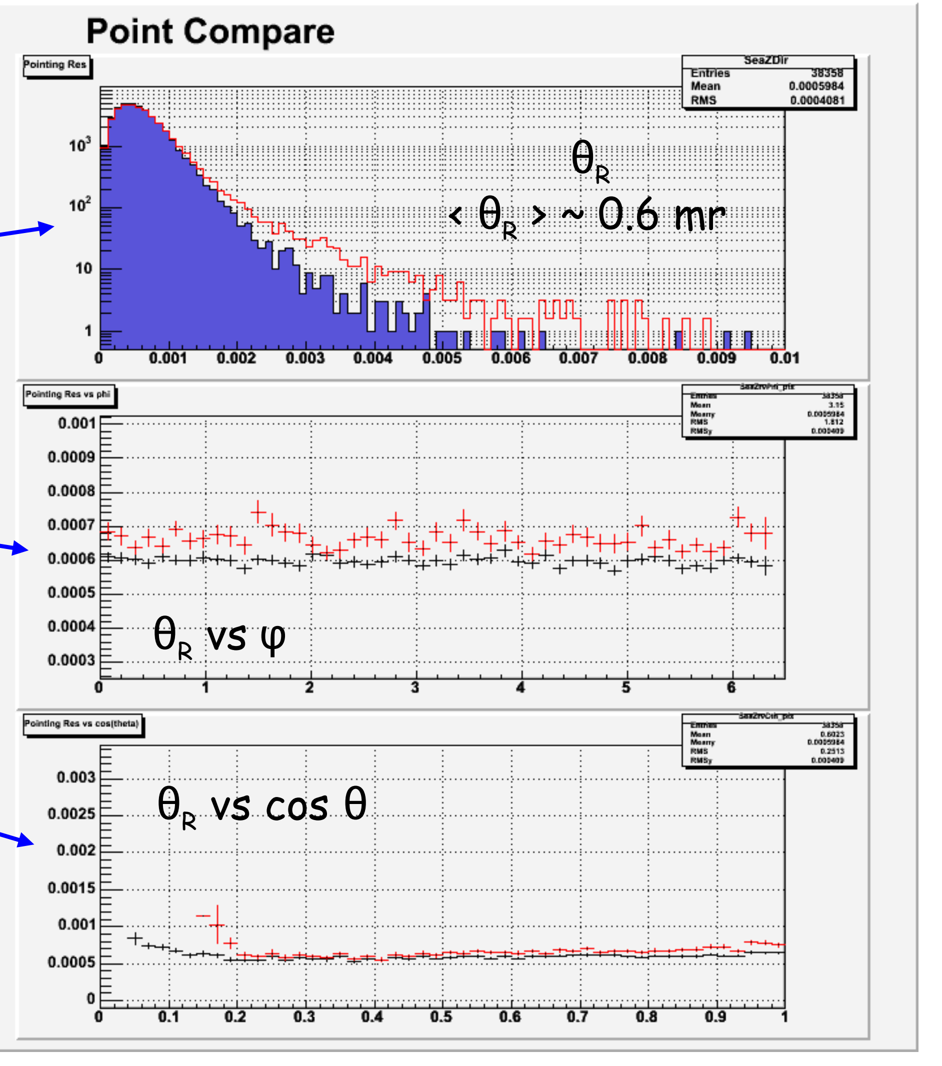


GLAST

Track Pointing

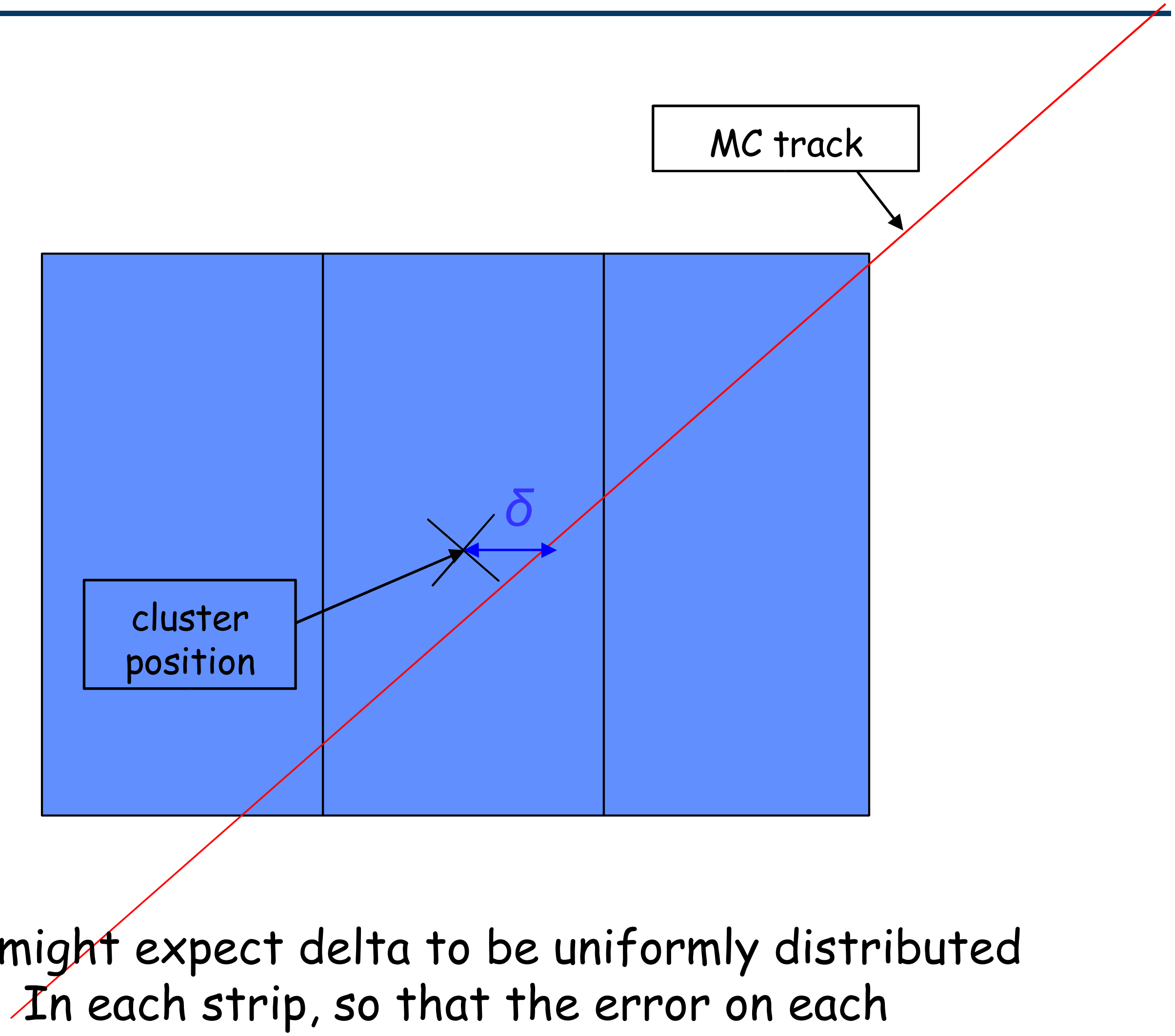
10 GeV Muons – upper hemisphere

- At 10 GeV:
  - multiple scattering effects are minimal
  - Should achieve ultimate resolution of tracker
- Plot angle track makes to Monte Carlo trajectory at the first hit on the track
  - $\theta_R = \arccos(\mathbf{u}_{fit} \cdot \mathbf{u}_{MC})$
- Plot versus φ
- Plot versus cos θ
- Plots contain
  - Blue fill/Black lines: MC Pat Rec
  - Red lines: Combo Pat Rec



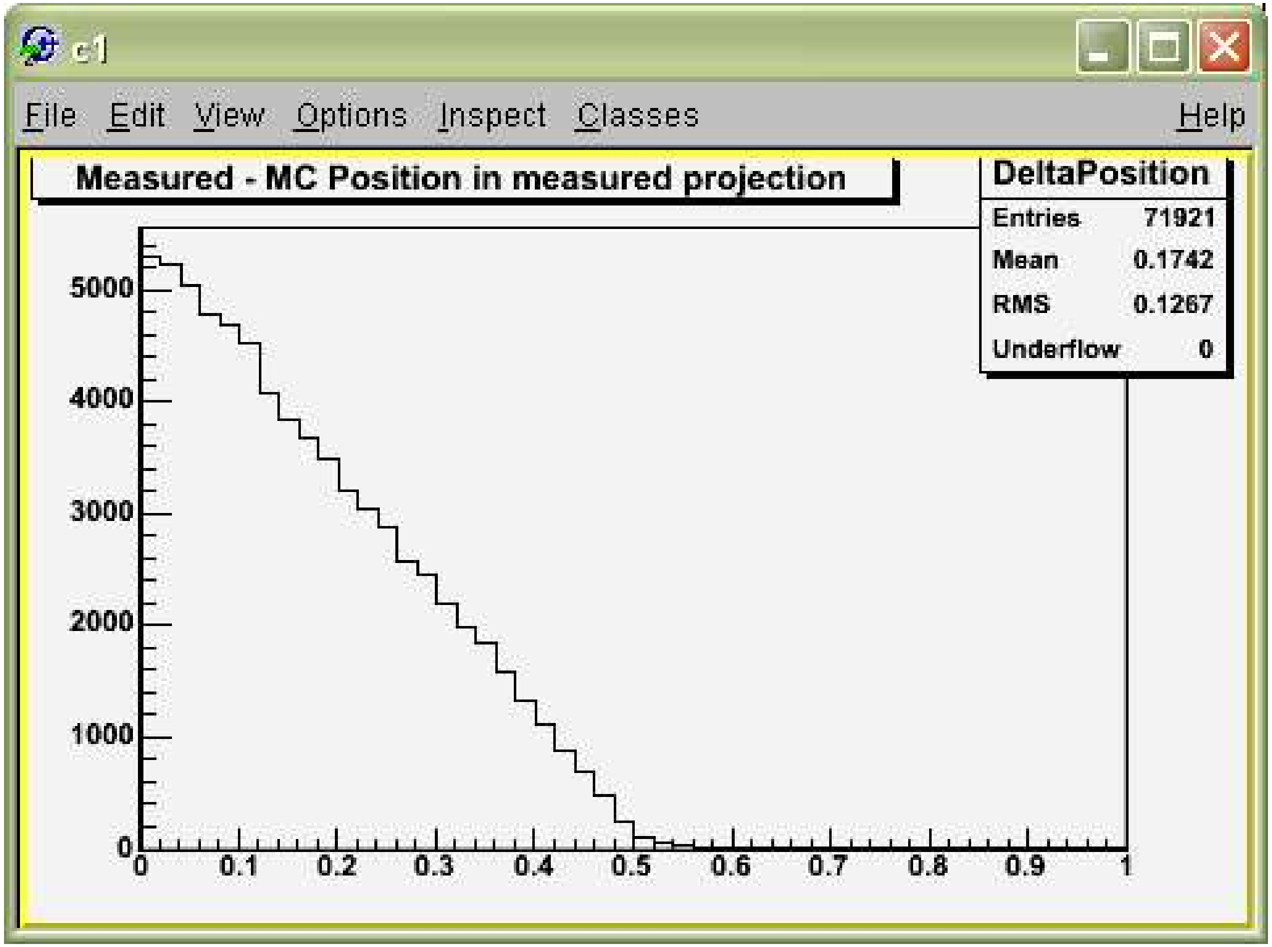
Extra slides

Measurement Errors



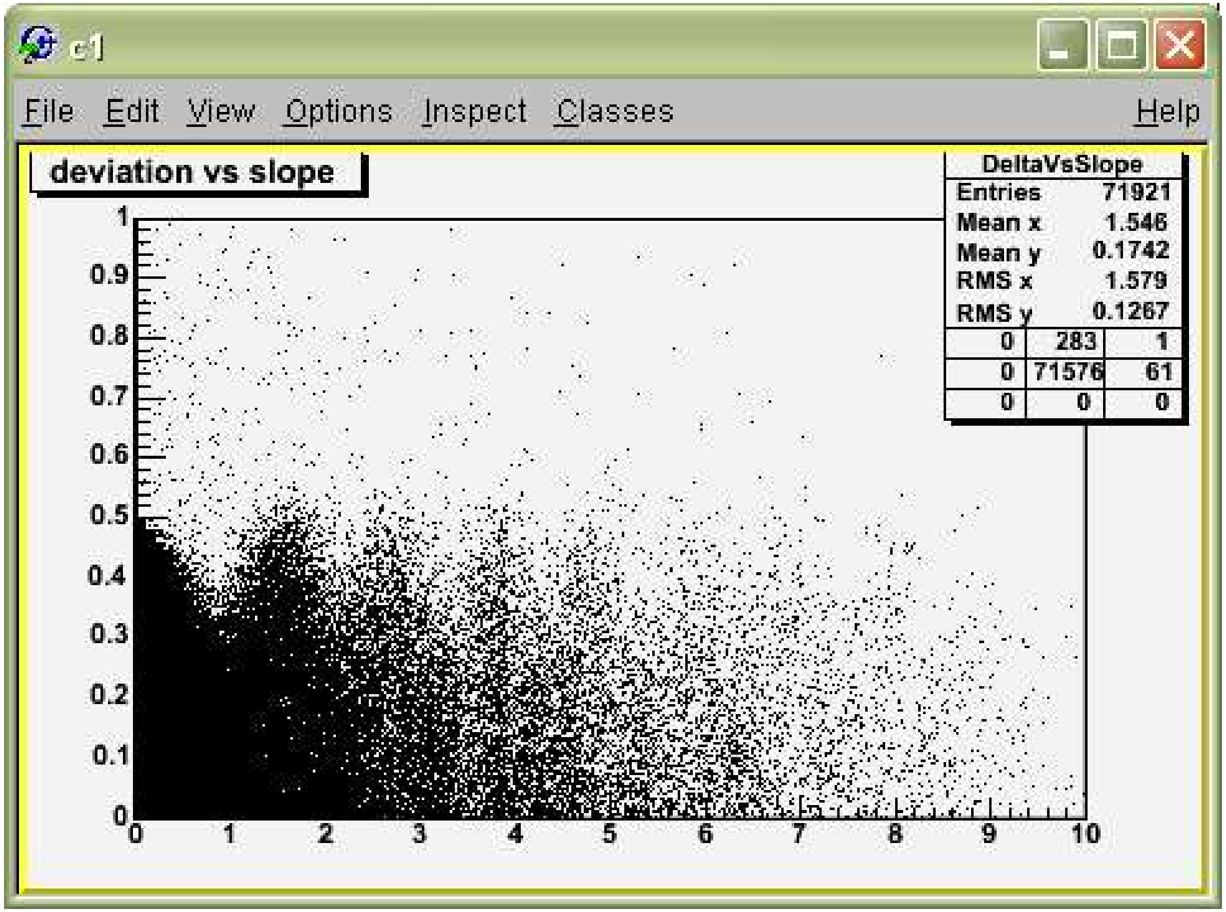
You might expect delta to be uniformly distributed
In each strip, so that the error on each
measurement would be stripPitch/√12

But…distribution of delta (all clusters)



fraction of strip width

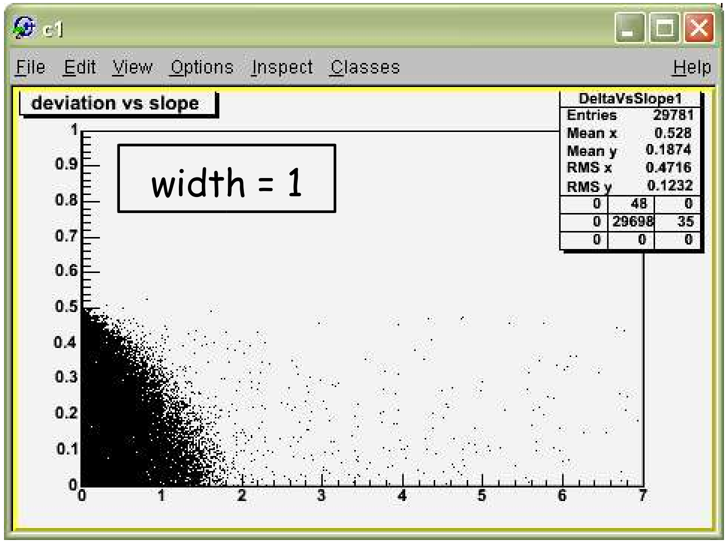
Delta vs. Slope

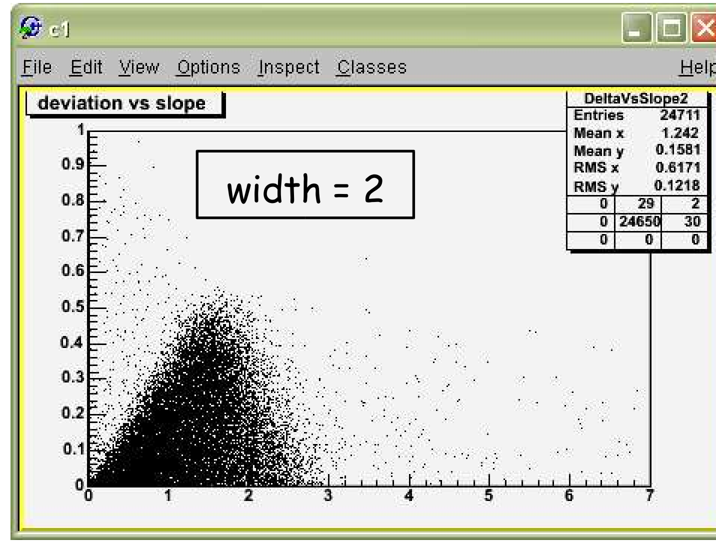


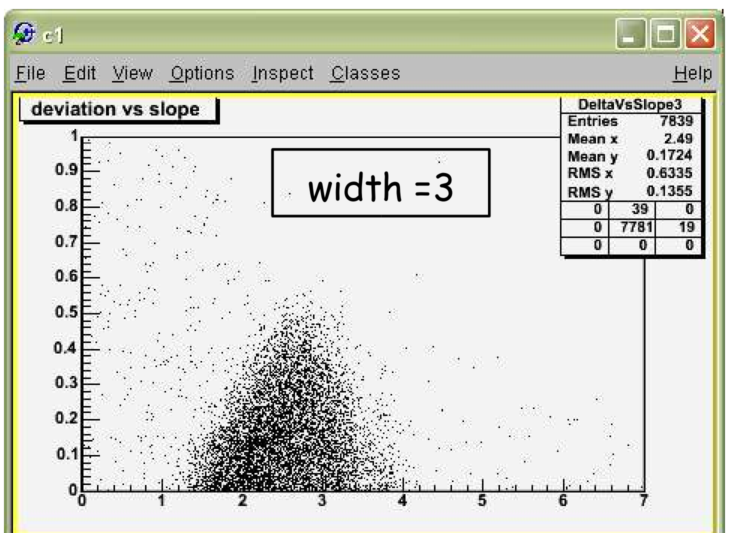
Slope is in units of stripPitch/siliconHeight

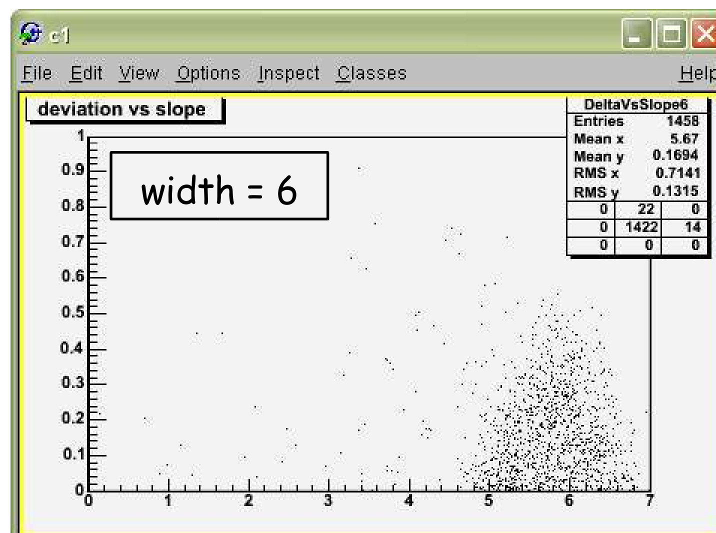
(Both slopes and deltas are folded around zero.)

Plots separated by cluster width

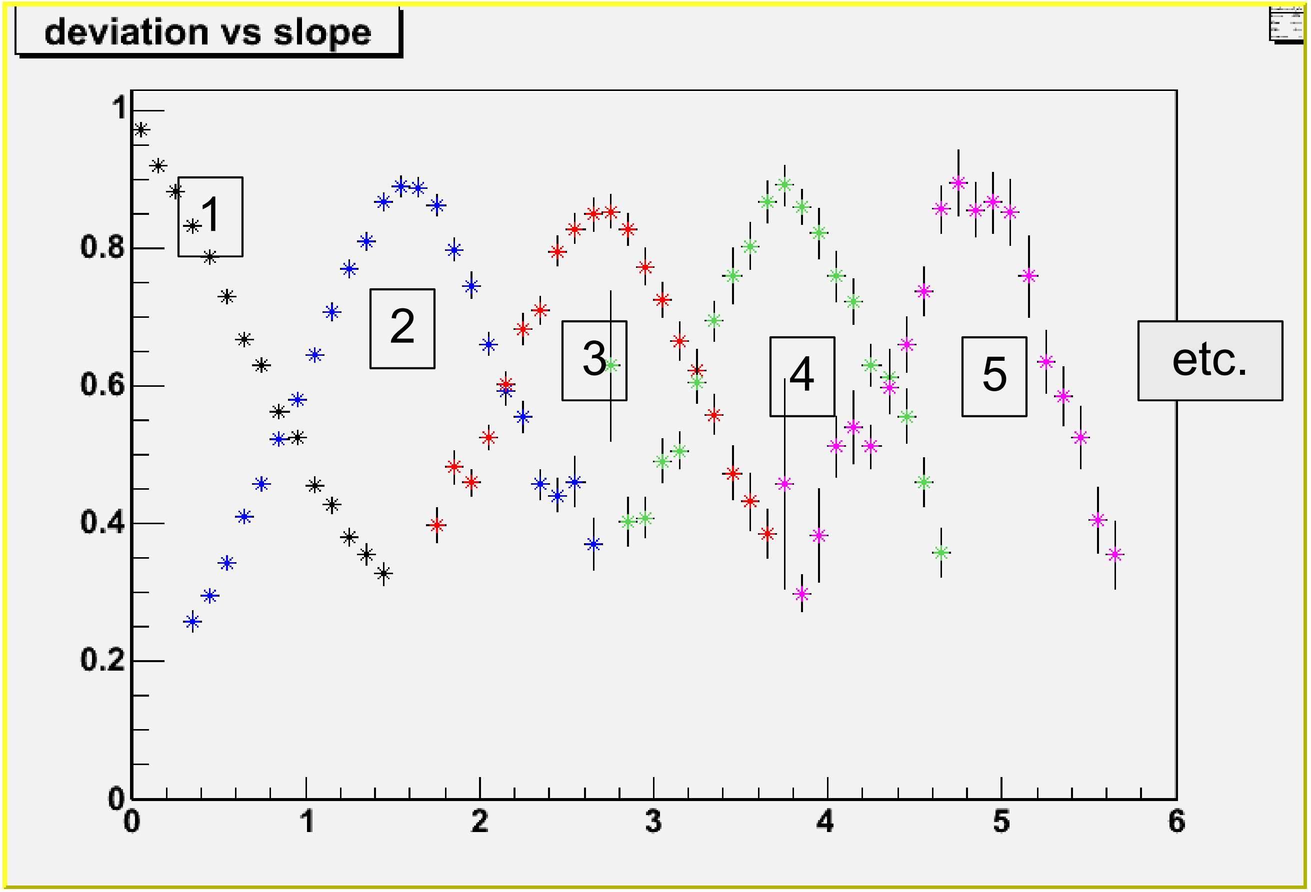








Error Factor



This is the factor by which the error is less than stripPitch/√12