

VTX reco update

Yujiao Chen, Dave Winter

16 June 2009

Basic Idea

- Fit helices to vtx hits
- Pattern recognition performed via Hough Transform
 - Assumes circular trajectories in xy plane
 - Through user-supplied reference point
- Vertex finding by refitting helices with fixed, common point
 - Tracks accepted if vertex constraint improves χ^2
 - Tracks rejected if vertex constraint worsens χ^2
 - Algorithm applied to both “primary” and “secondary” tracks
- Track fitting based on Karimaki’s methods
- Vertex finding based on E910 techniques

Procedure

- Multiple iterations:
 - Iterating on unused hits with wider theta bins and number of layer thresholds
 - Hough transform on hits assuming ref $(x,y)=(0,0)$
 - In theta bins
 - Should find primaries
 - Should reject secondaries far from vertex
 - Find vertex using current list of found tracks
- Road finding on remaining unused hits
 - Start from outside layer, working inward
 - Hough transform to accept/reject candidates
- Secondary vertex search
 - Find secondary tracks with intersecting helices
 - Apply vertex finding on them
 - No constraints on helicity or number of input tracks

Implementation

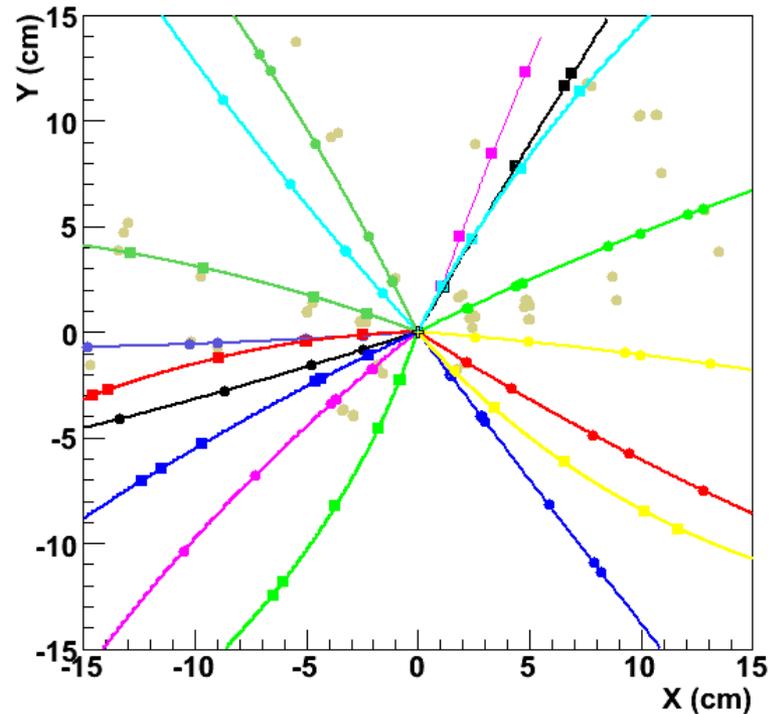
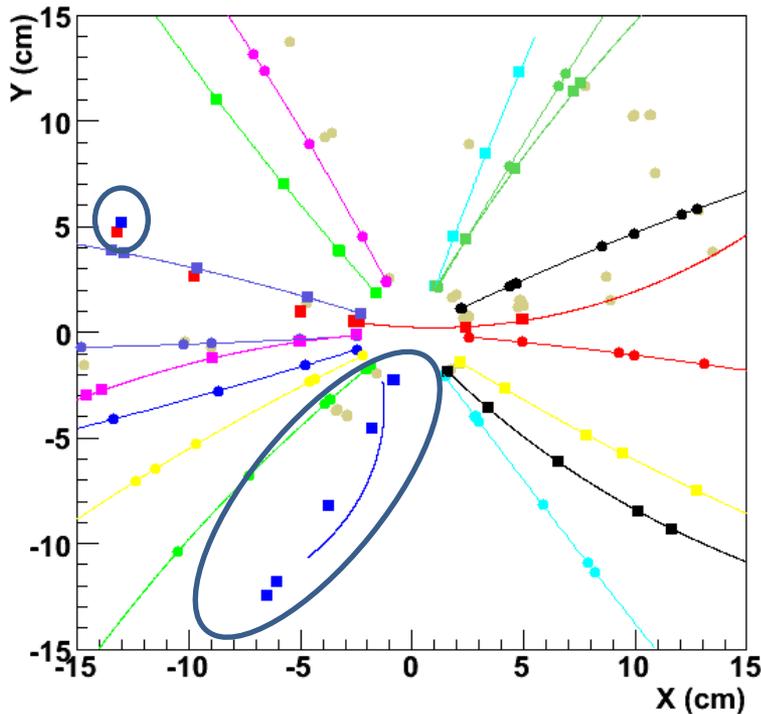
- Fun4All module + mutoo framework
 - One module for fitting
 - Module for evaluation of MC matching
 - Reads SvxFisaHit, SvxFusterList
 - Outputs helix tracks, vertices (both are mutoo maps) and adds to VtxOut
- Performs reconstruction three times:
 - Fits collection of PISA hits (no pattern recognition needed) [Reference for fitting]
 - Finds tracks and reconstructs vertices for PISA-derived SvxFusters [Reference for patt recog + reco, ie. no digitization or clustering]
 - Finds tracks and reconstructs vertices for SvxFusters [the whole shebang]
- Average running time (pythia) < 220 ms / event
 - Includes all three steps
 - Includes filling of evaluation trees
 - No serious optimization performed yet

Initial studies

- 1M pythia events reconstructed at Nevis
 - Simulated with phuniverse current as of ~Jan 1
- Minbias pythia configuration
- All events thrown from (0,0,0)
- Matching based only on hits
 - Unbelievably efficient?
 - For tracks with 3 or more layers:
 - 10M MC tracks
 - 200k MC Reco tracks with no MC track matching
 - 3k MC Reco tracks matching 2 MC tracks
 - Need to reverse the sense of this test (i.e. ask how often an MC track has no reco track matching it)
- We can currently look at:
 - Effect of fitting tracks w/o pattern recognition and w/o spatial resolution
 - Primary vertex position resolution

Event display example

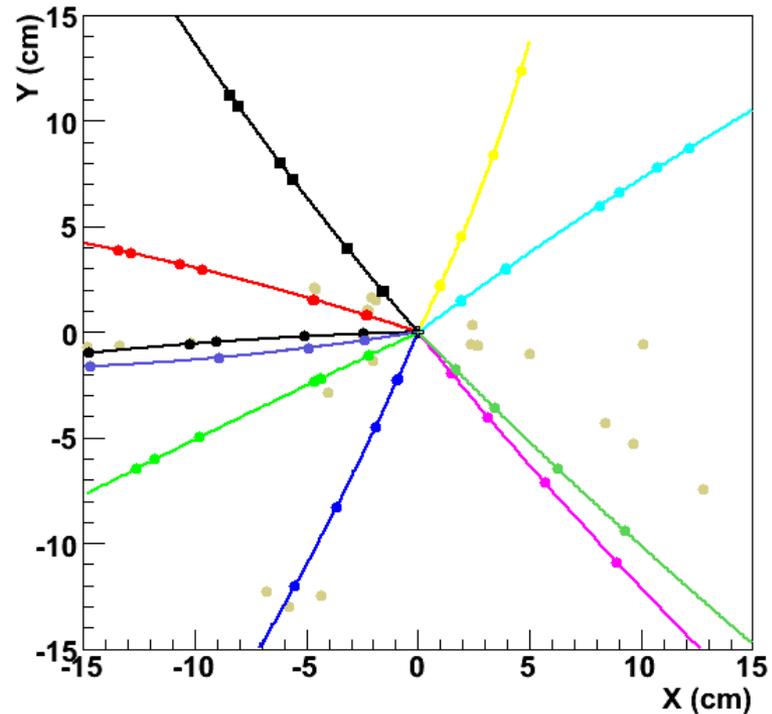
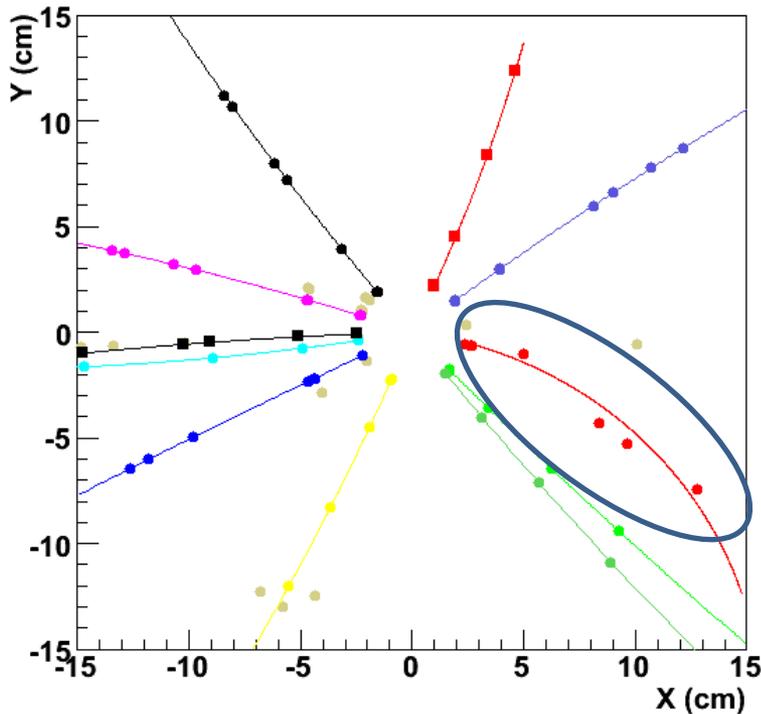
VTX Reconstruction Event Display
File \$WORKAREA/build/vtx_reco/vtxfit_mc_eval.root -- Event Number 3



- Odd track with “extra” hit? Well reconstructed...

Event display example

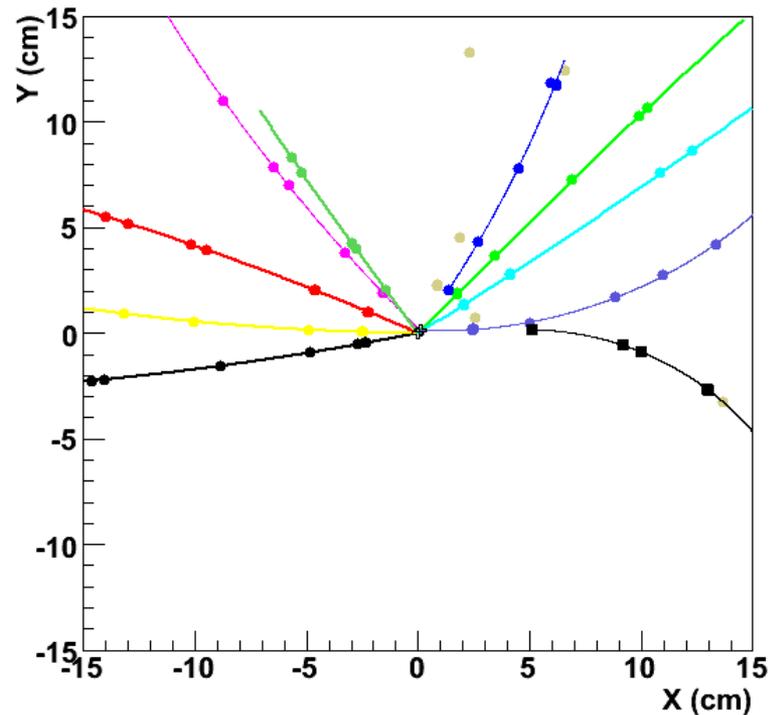
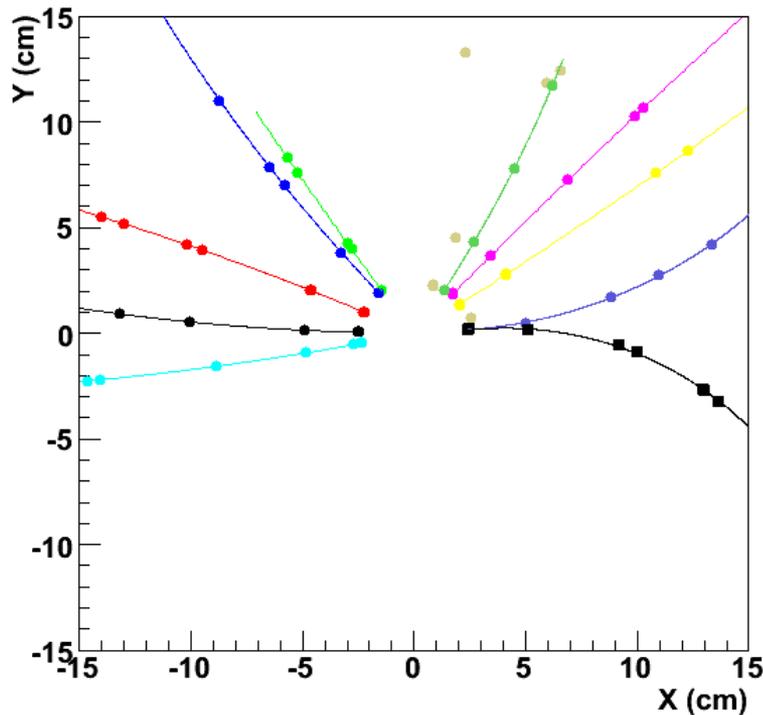
VTX Reconstruction Event Display
File 100/vtxfit_mc_eval.root -- Event Number 3



- Some tracks will just be hard to reconstruct.

Event display example

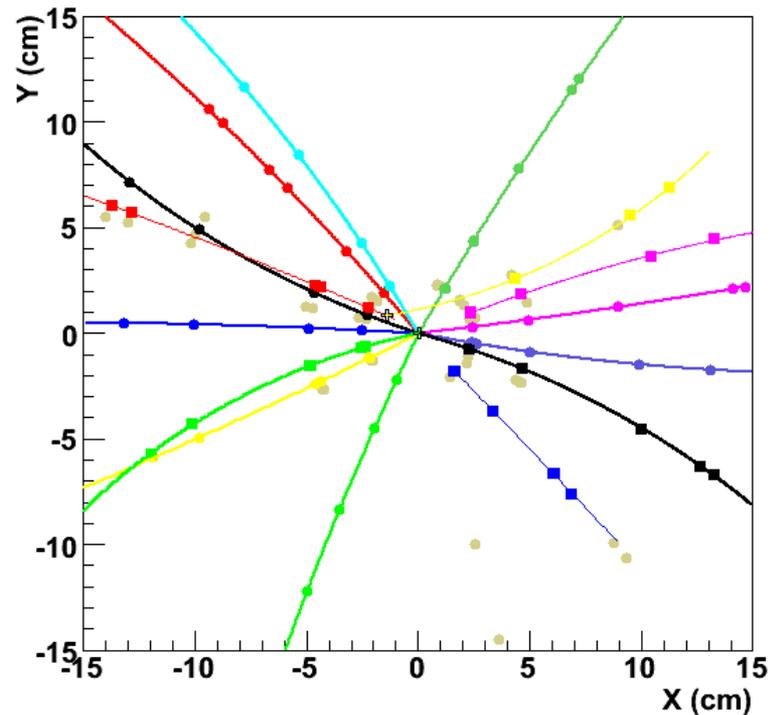
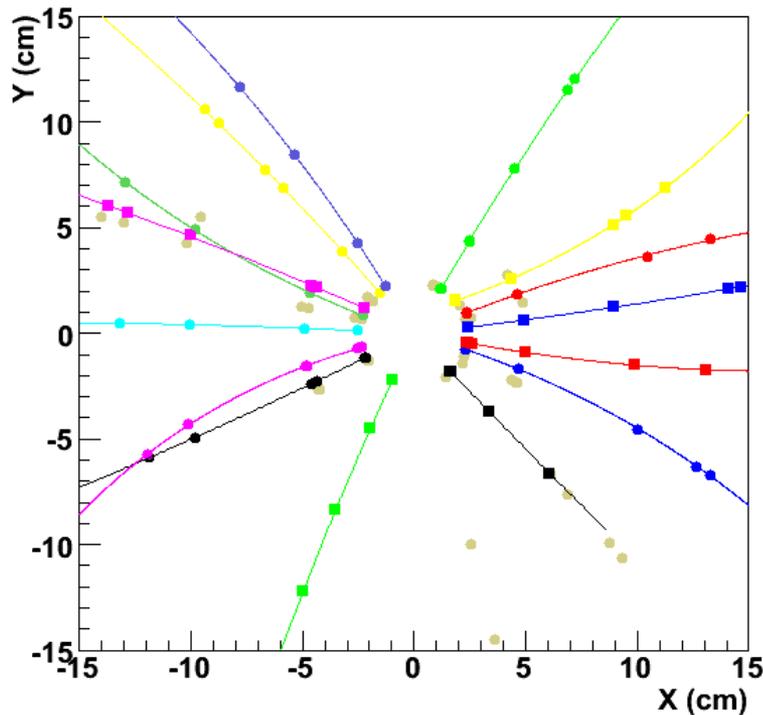
VTX Reconstruction Event Display
File 100/vtxfit_mc_eval.root -- Event Number 37



- Finding conversions will take a bit more effort

Event display example

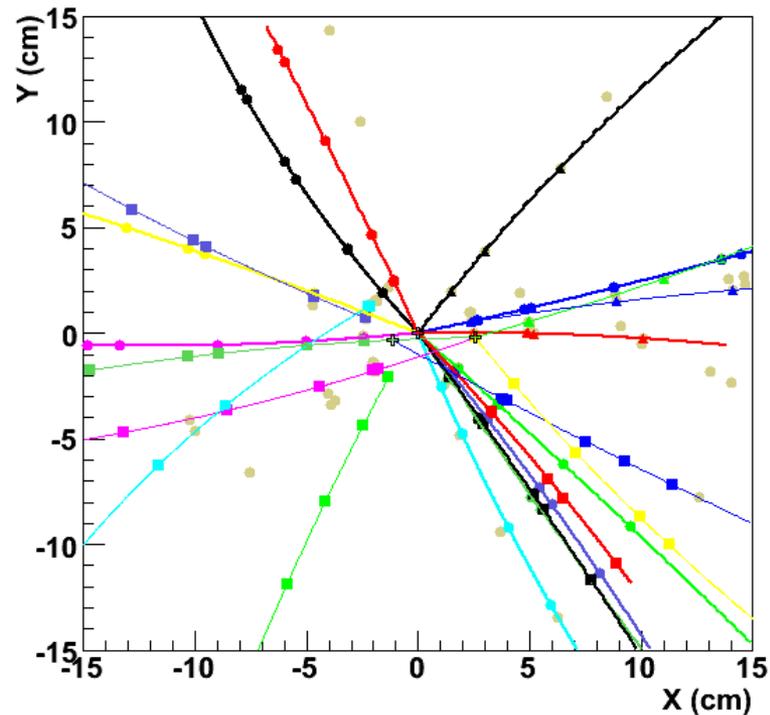
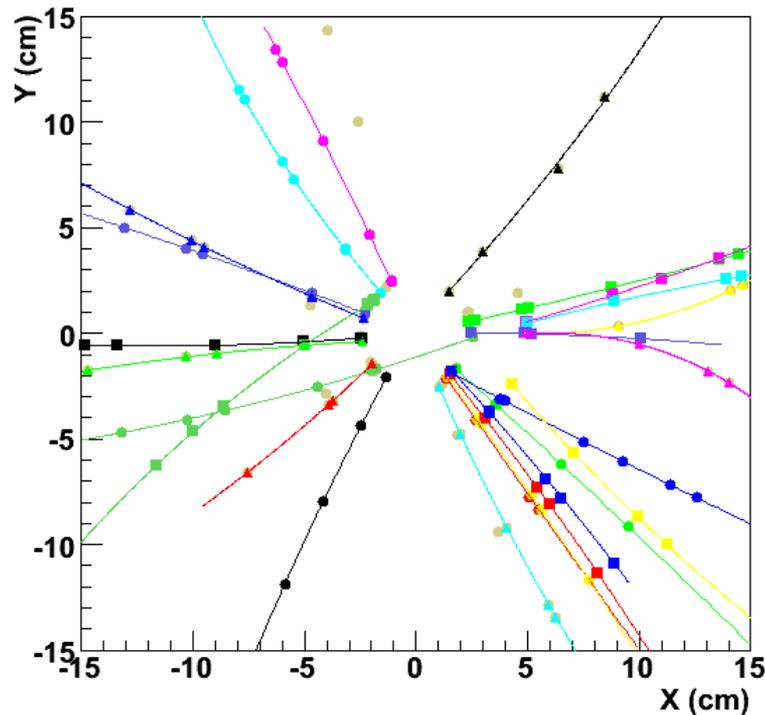
VTX Reconstruction Event Display
File 100/vtxfit_mc_eval.root -- Event Number 73



- Example of secondary vertex

Event display example

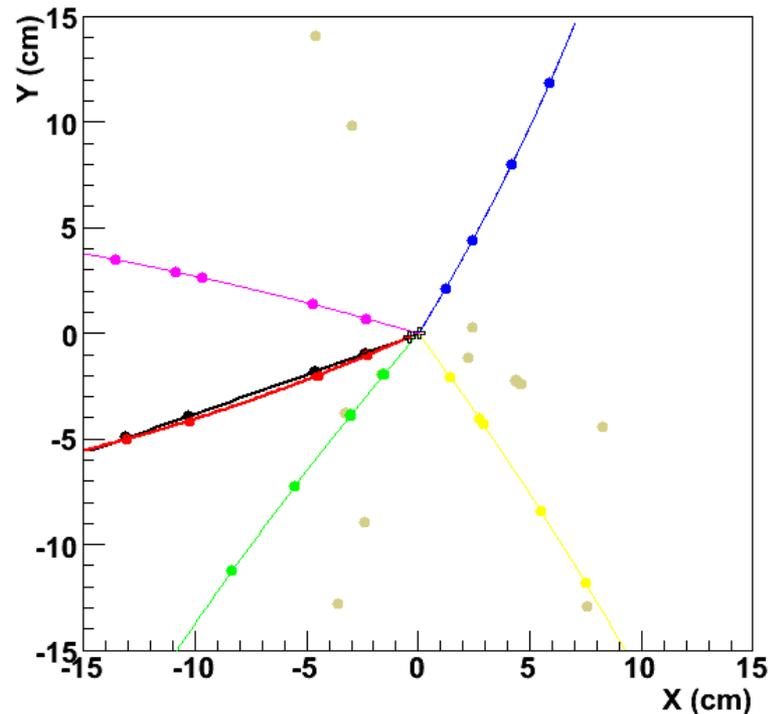
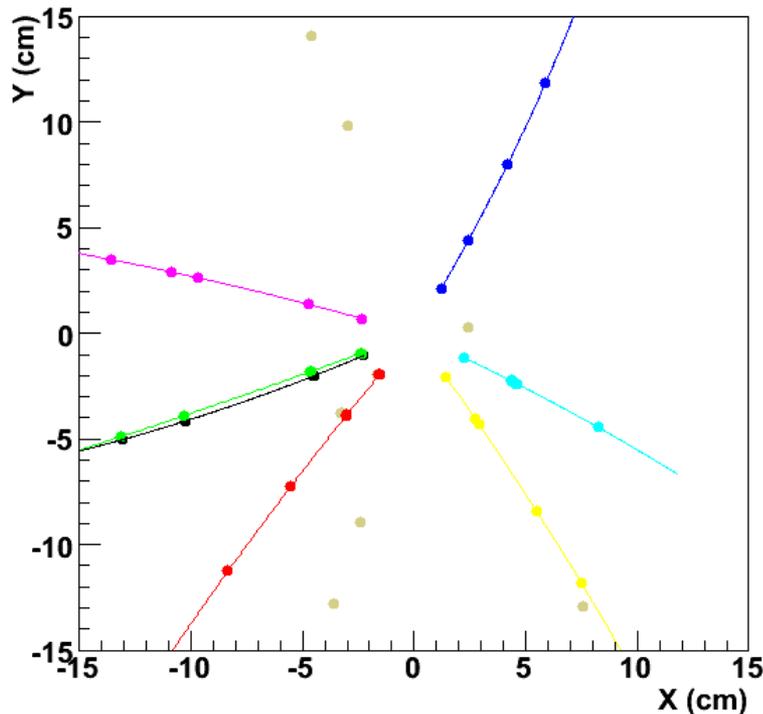
VTX Reconstruction Event Display
File 100/vtxfit_mc_eval.root -- Event Number 3189



- Multiple vertices (one is a hadronic interaction?)

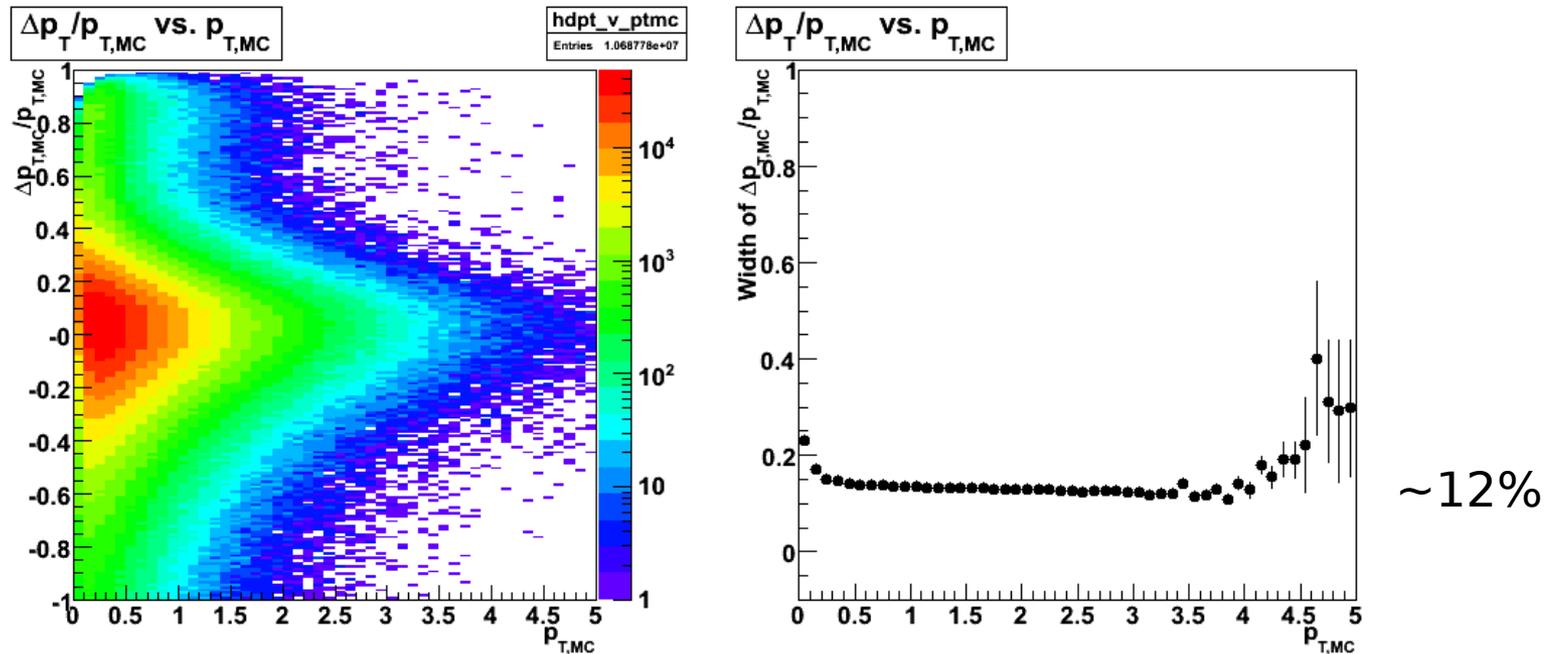
Event display example

VTX Reconstruction Event Display
File 100/vtxfit_mc_eval.root -- Event Number 7359



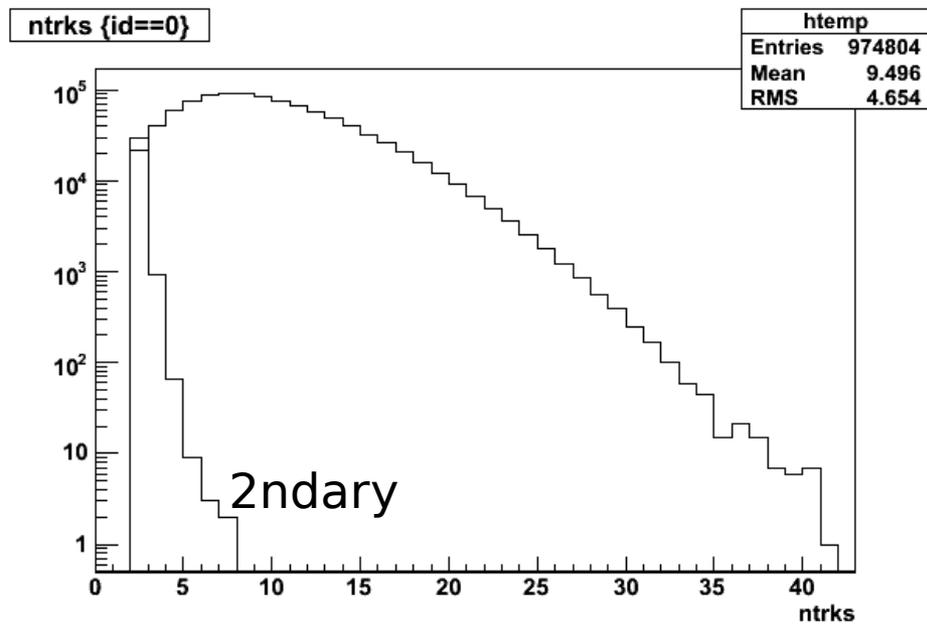
- **BOLD**: primary vertex & tracks, Others: secondary tracks and vertices
- Example of ambiguity between primary and secondary vertices

Momentum resolution

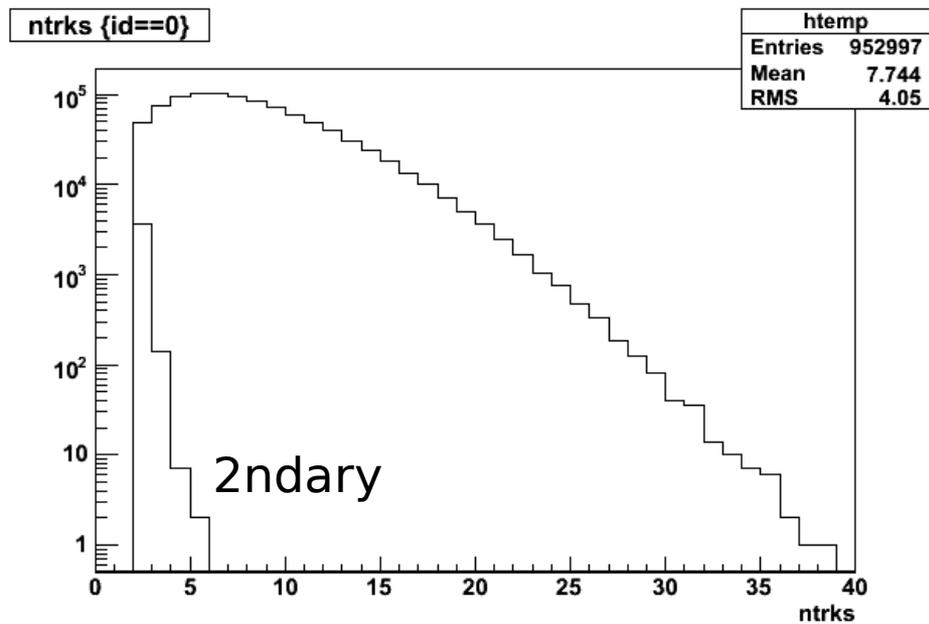


- No quality cuts placed on tracks for these plots
- Only compares SvxSnglPisaHit results (no spatial resolution, no pattern recognition) with momentum in FKIN node

Number of tracks used in vertex fits

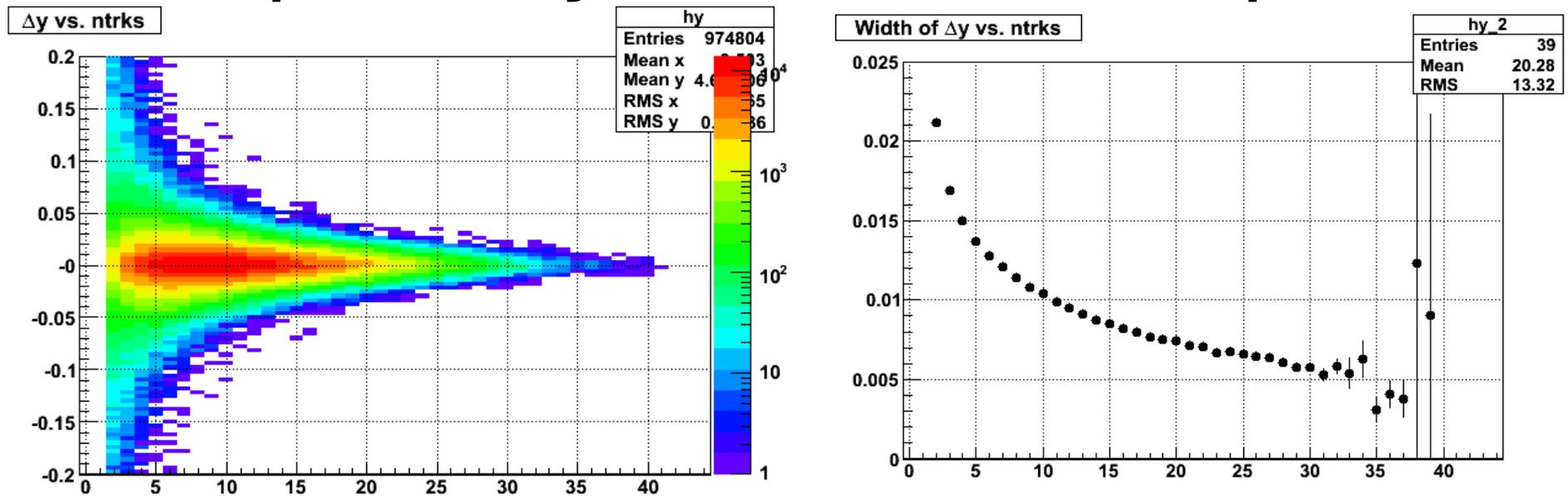


PISA Hits as Clusters
+ Pattern Recognition



Detector Clusters
+ Pattern Recognition

Δy vs. number of tracks (primary vertex) example

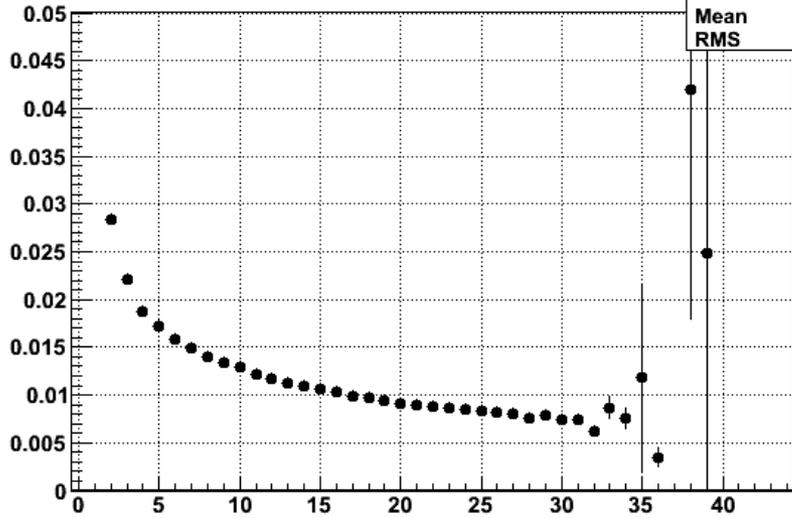


PISA Hits as Clusters
+ Pattern Recognition

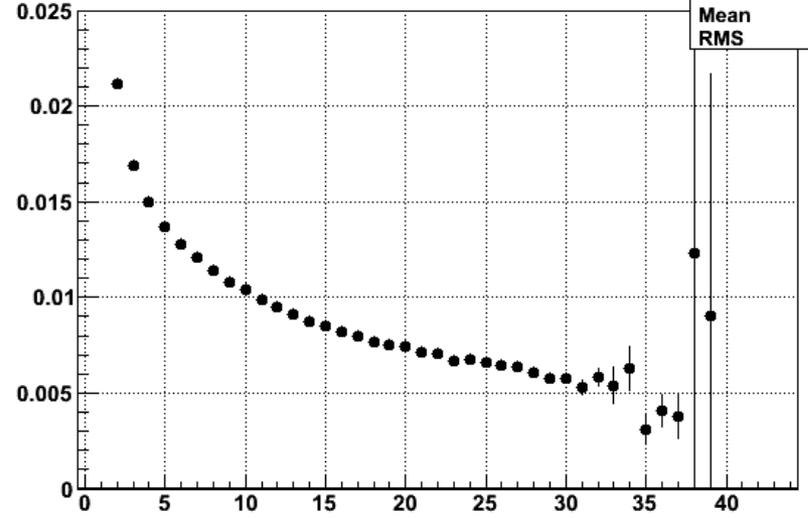
Sigma of difference (prim)

X . y . z

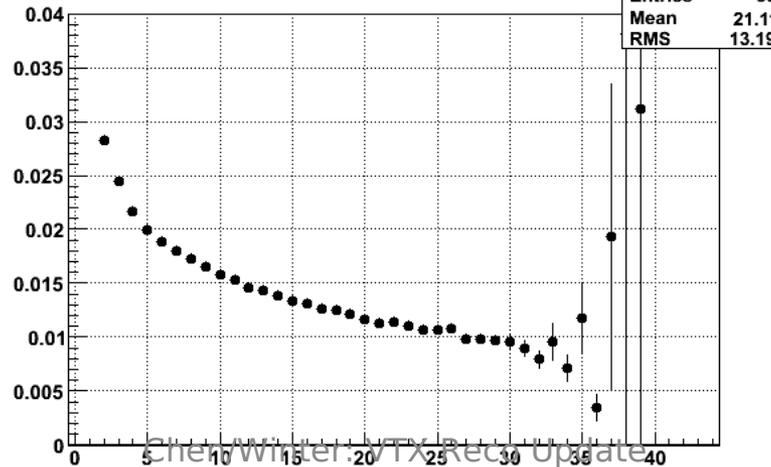
Width of Δx vs. ntrks



Width of Δy vs. ntrks

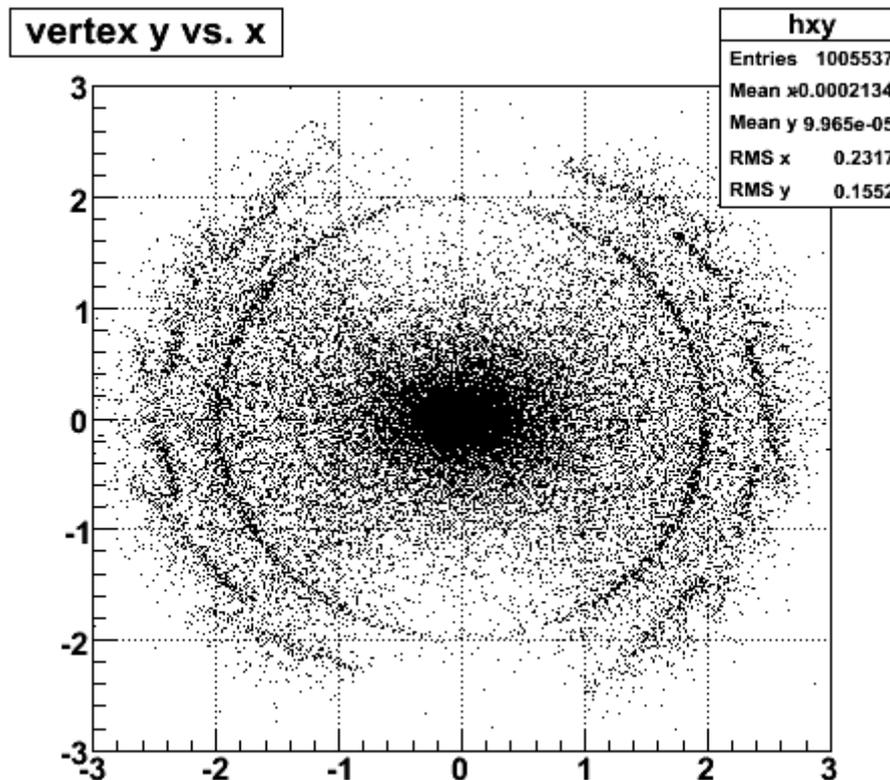


Width of Δz vs. ntrks



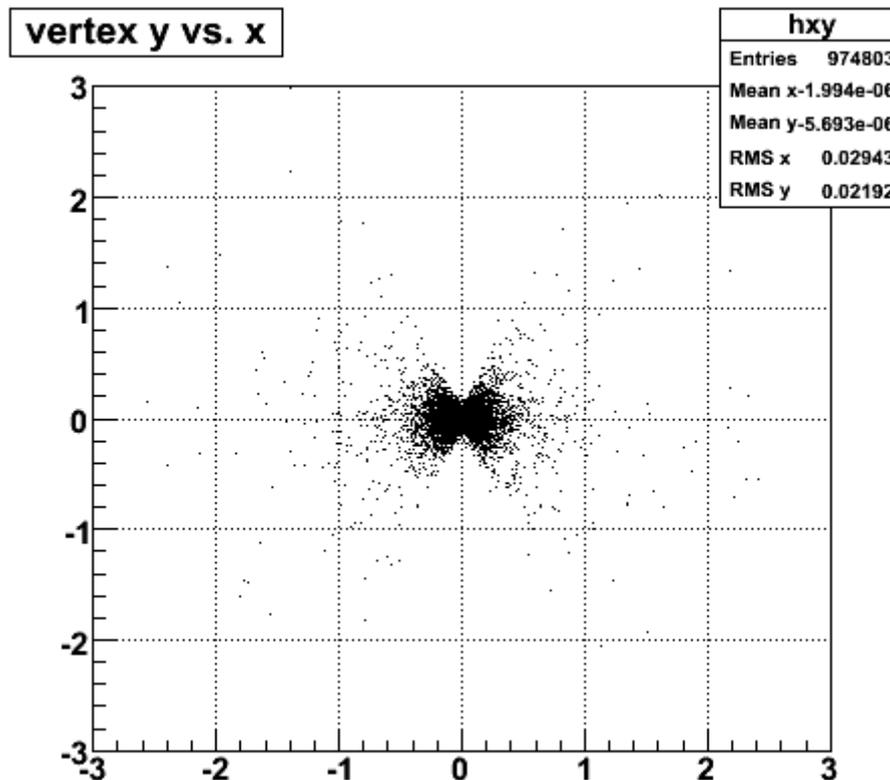
PISA Hits as Clusters
+ Pattern Recognition

Detector Tomography



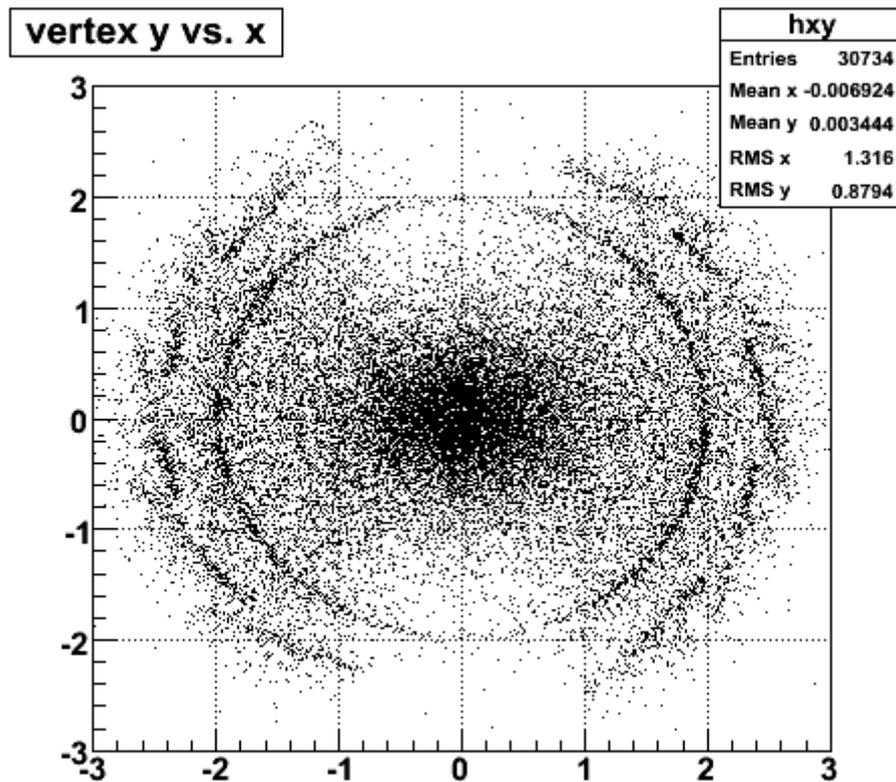
All Vertices

Detector Tomography



Primary Vertices

Detector Tomography



Secondary Vertices

Some ToDos

- In the process of separating our evaluation trees from main module
- Also translating Yujiao's matching into fun4all module
- Need associations between clusters and geant hits (lots of info from Alan and Kenichi is helping)
- Evaluation of efficiencies (tracks, vertices)
- More complete evaluation of resolutions and errors
- Understand finer issues of secondary vertex reconstruction
 - Conversions?
 - Fake vertices?
 - Optimization
- Add FVTX hits to the mix?