# SIMULATION STUDIES AT SANTA CRUZ

Bruce Schumm

University of California at Santa Cruz

ALCPG Workshop, Vancouver

July 19-22, 2006

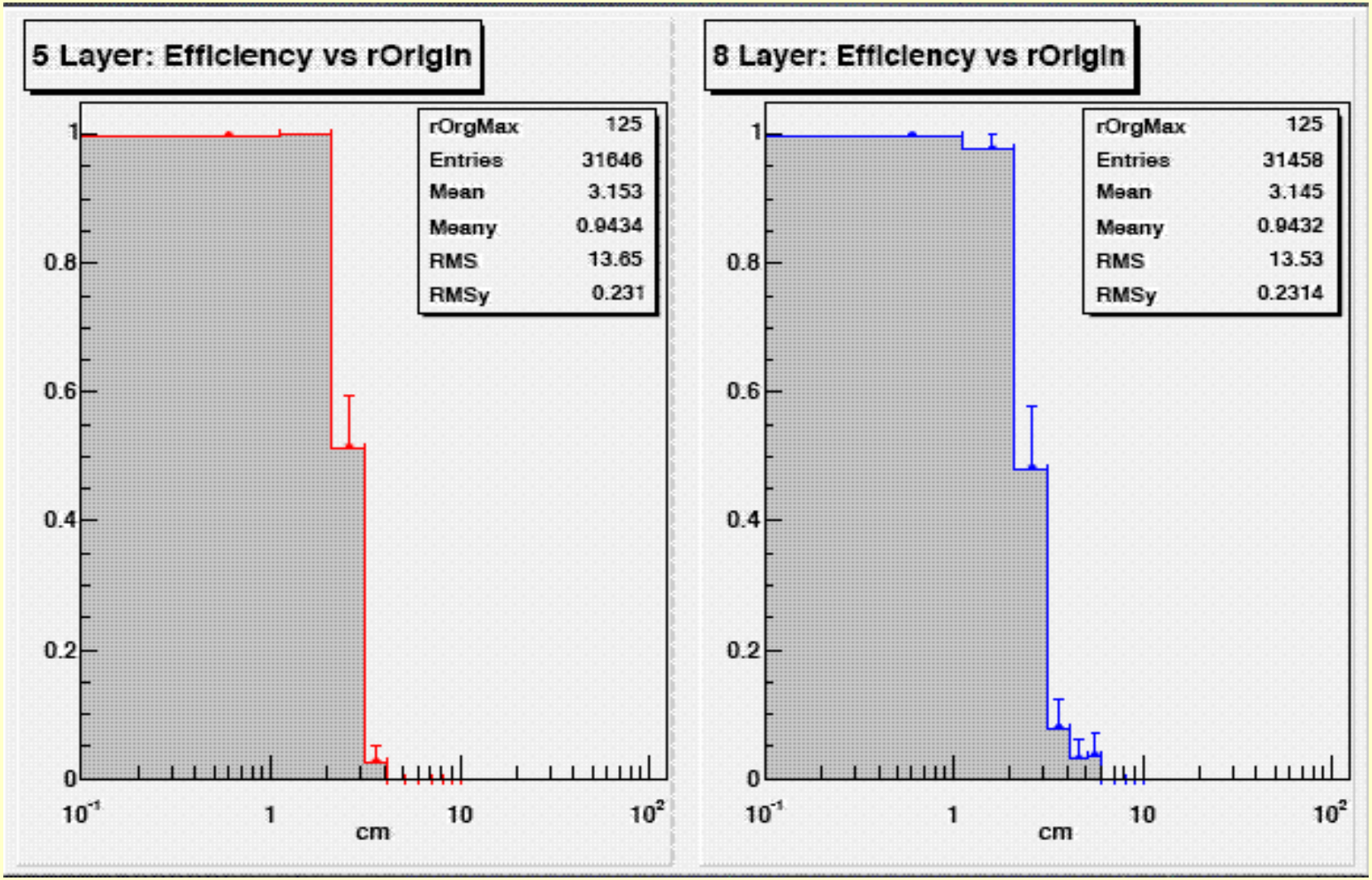
**Special Recognition: Eric Wallace, UCSC senior thesis student (to U. Washington Physics in the fall)**

# Latest Focus: SiD Tracking Efficiency

- Snowmass: Tim Nelson wrote axial-only algorithm to reconstruct tracks in absence of Vertex Detector
- UCSC idea: use this to “clean up” after vertex-stub based reconstruction (VXDBasedReco)
- About 5% of tracks originate beyond the VXD inner layers

# Efficiency vs. rOrigin with VXDBasedReco

Michael Young, UCSC



# Cheater

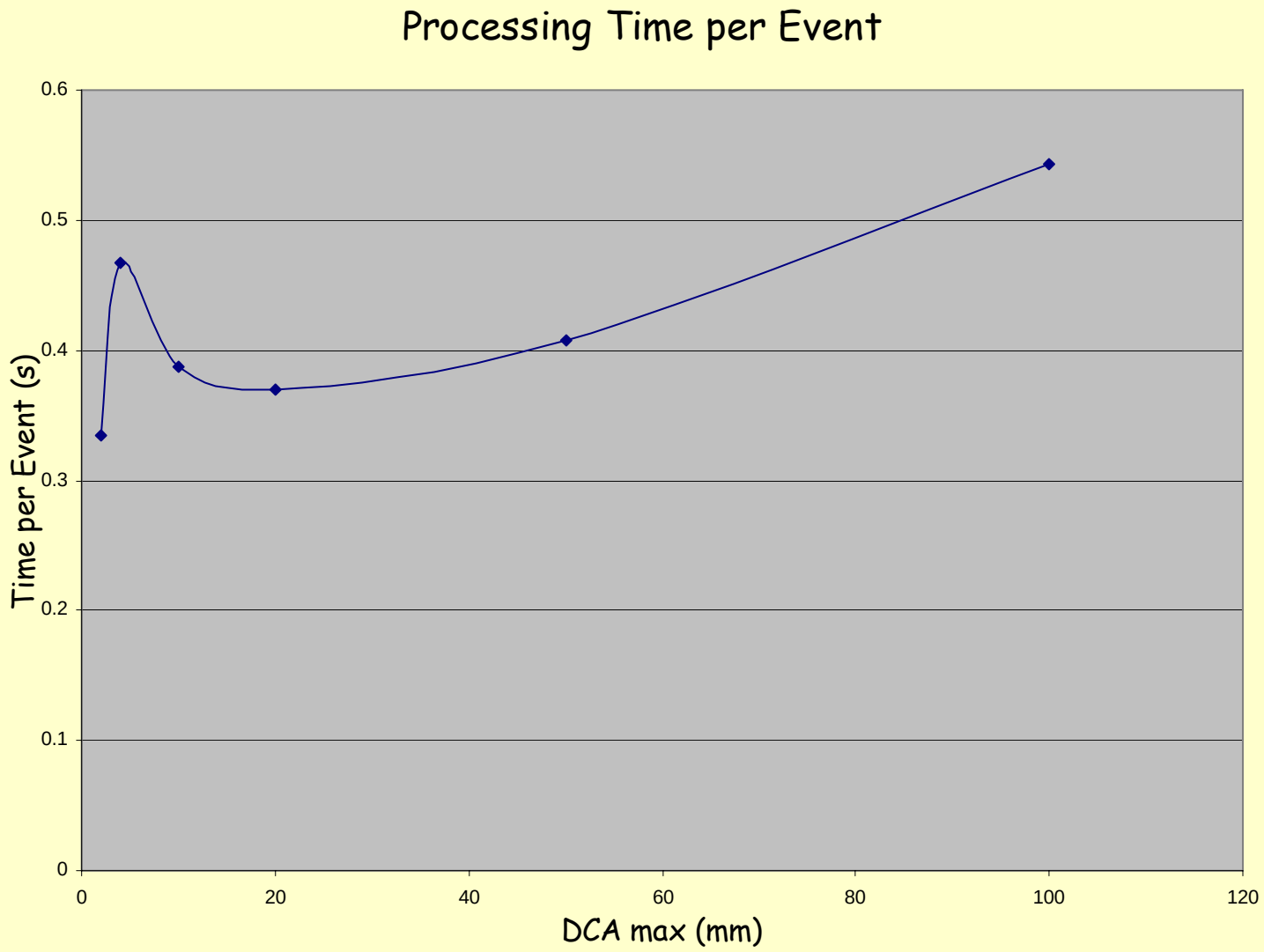
- VXDBasedReco had not yet been ported to org.lcsim framework, so…
- Wrote “cheater” to emulate perfectly efficient VXDBasedReco; assume anything that can be found by VXDBasedReco is found and the hits flagged as used
- Loops over TkrBarrHits and MCParticles, finds particles with rOrigin < 20mm and hits from those particles, removes them from collections
- rOrigin defined as sqrt(particle.getOriginX()^2 + particle.getOriginY()^2)

# AxialBarrelTrackFinder (Tim Nelson, SLAC)

- Loops over all hits in each layer, from the outside in, and finds 3 “seed” hits, one per layer
- Performs CircleFit (alogrithm provided by Norman Graf) to seed hits
- Looks for hits on the remaining layers that can be added to seed fit, refitting after each hit added.
- If at least 4 hits on track, and Chi^2 of fit reasonable, creates track object and adds to collection

# DCA cut

- Original code placed tight limitation on dca of seed fit to IP, in order to make combinatorics manageable. Limiting the number of hits that need to be considered allows for easing this restriction without drastically increasing computation time.
- Easing the max dca from 2mm to 100mm almost doubled efficiency, with computation time staying under 1 sec per event. However, also produces significant increase in reconstruction of “fake tracks”.


Processing Time per Event
Time per Event (s)
0.6
0.5
0.4
0.3
0.2
0.1
0
0
20
40
60
80
100
120
DCA max (mm)

# MC Track Association

- Added a constructor to StandaloneAxialBarrelTrack to include association with majority MC particle and track purity.
- MCParticle associated with a given track is just majority particle - particle associated with largest number of hits used to create that track
- To reduce garbage tracks, they must have a purity (# hits from majority particle/#hits in track) of at least 75%
- Only counts one found track per MCParticle. If there are several tracks associated with the same particle, only counts the first one.
- Tracks with purity less than 75% and “successful fit” counted as fake tracks

# Z segmentation

- Z segmentation logic as written requires each new hit to be within half a module length of a line from the origin to the first hit used.
- Requires a certain stiffness of tracks, excludes a low momentum regime that we wanted to include in our analysis, so commented out segmentation logic.
- Did maintain requirement of same sign in z

# Other Modifications

- Code as written was associating same set of hits with many tracks. Turns out this was mainly superficial (wasn't affecting the properties of the tracks themselves), problem seems to be with scope of variable storing set of hits. Fixed.
- Algorithm creates multiple tracks with same first 1 or 2 seed hits; once it picks 2 hits, it will make a track with every workable combination of those 2 with a third. Not sure if this is intentional or not; could add a few lines to adjudicate between multiple tracks using same hits, e.g. pick one with best chi^2. Added temporary fix to only allow single use of each hit.

# Fiducial Volume

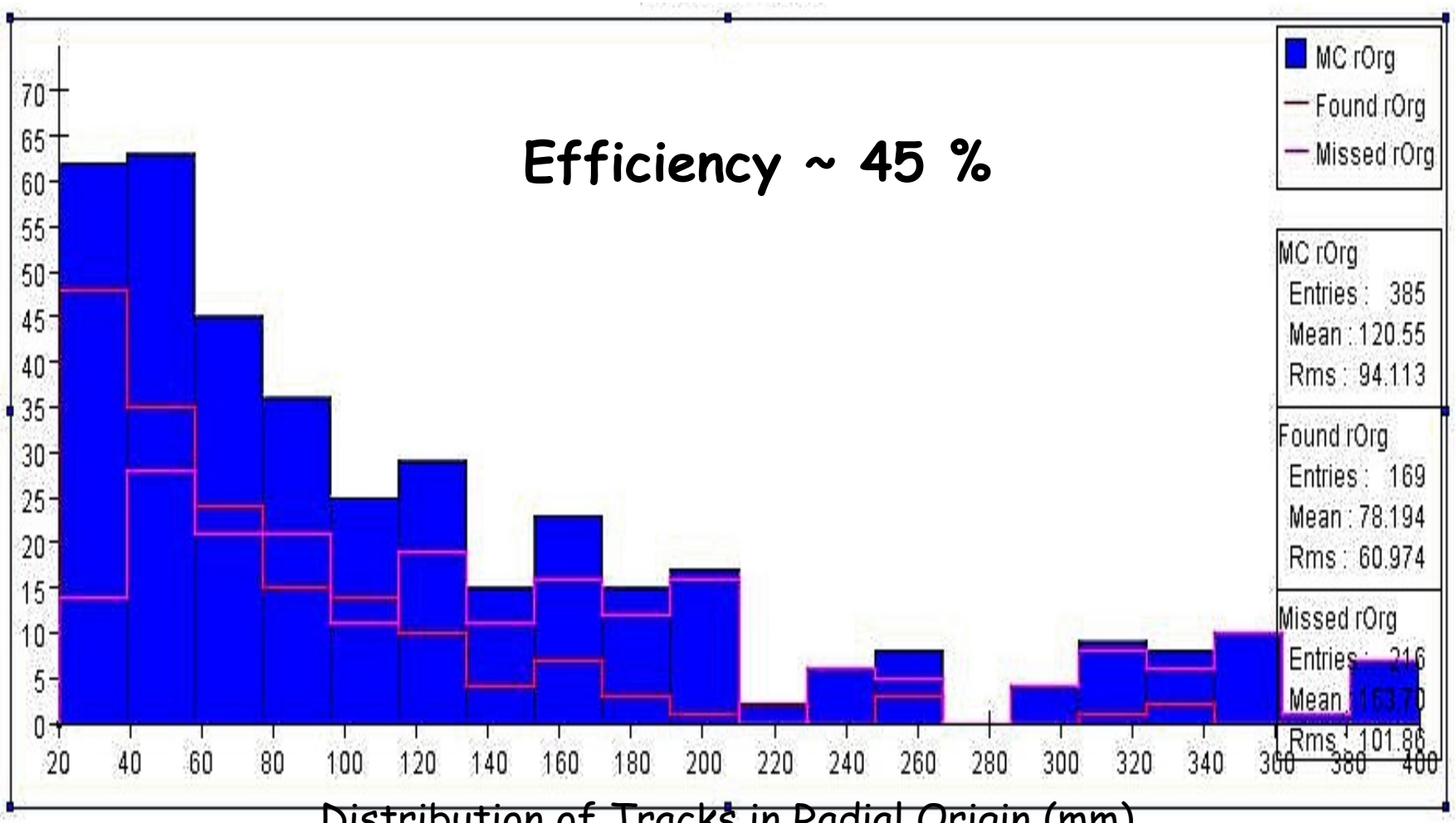
For found tracks/MC denominator

- 20mm < Rorg < 400mm (>3 hit layers)
- Pt > .75 GeV (>3 hit layers)
- Cos(theta) < 0.5
- Final state, not backscatter

For Fake Tracks

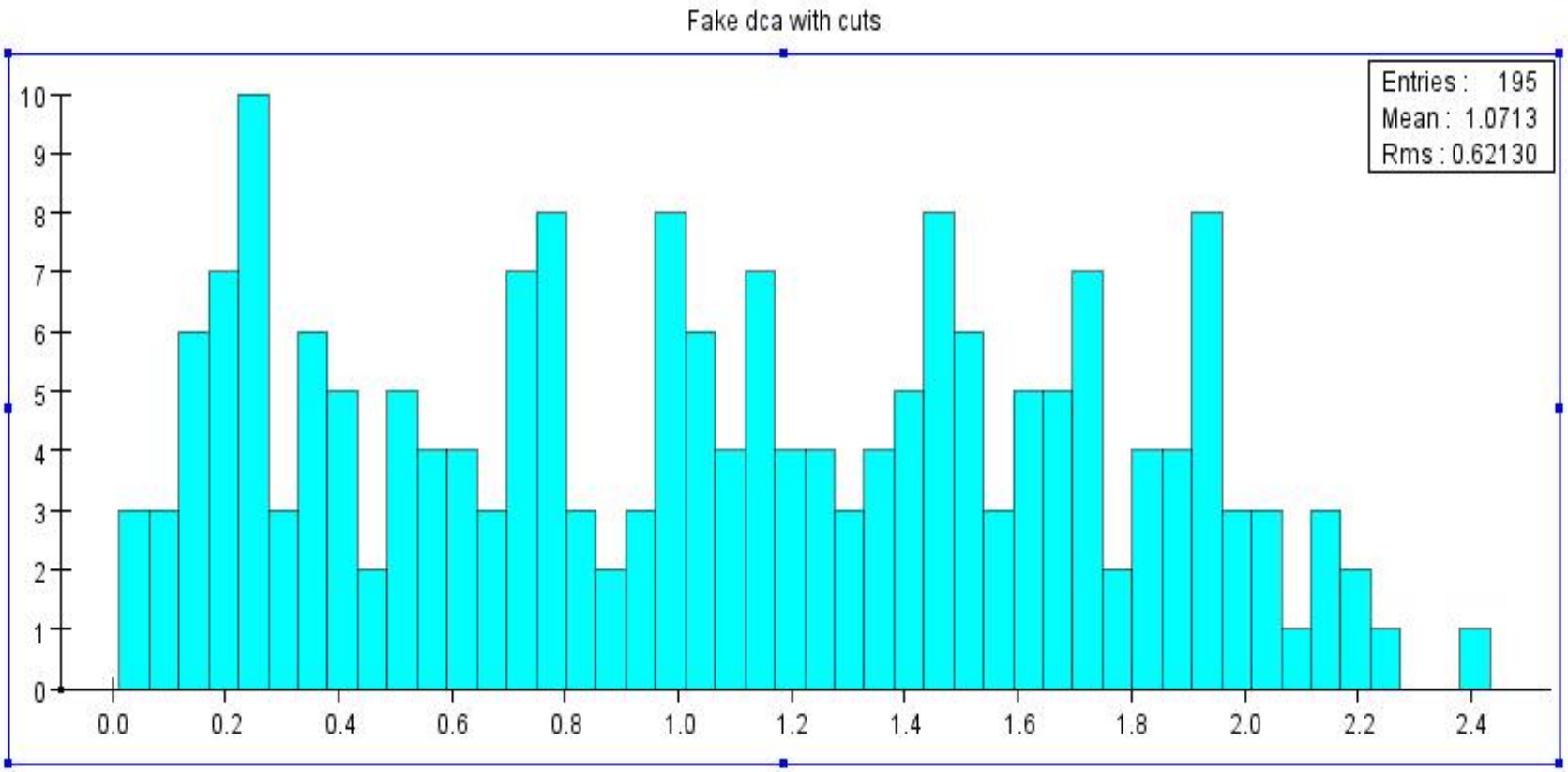
- Dca < 100mm
- “Successful” fit

# Distance of Closest Approach < 2.0 mm




**Eric Wallace**

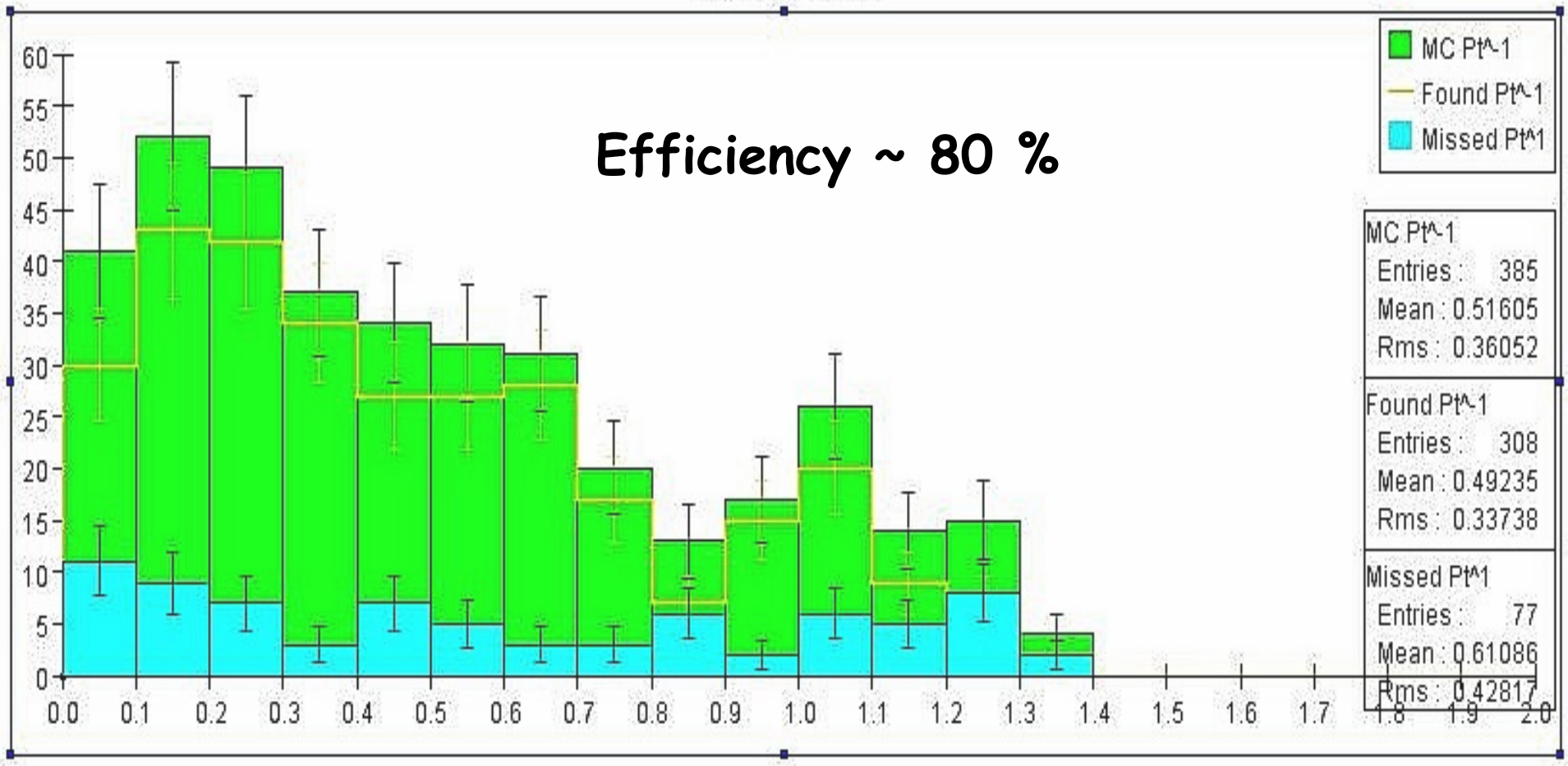
# Distance of Closest Approach < 2.0 mm




Distribution of Fake Tracks in DCA (mm)

**Eric Wallace**

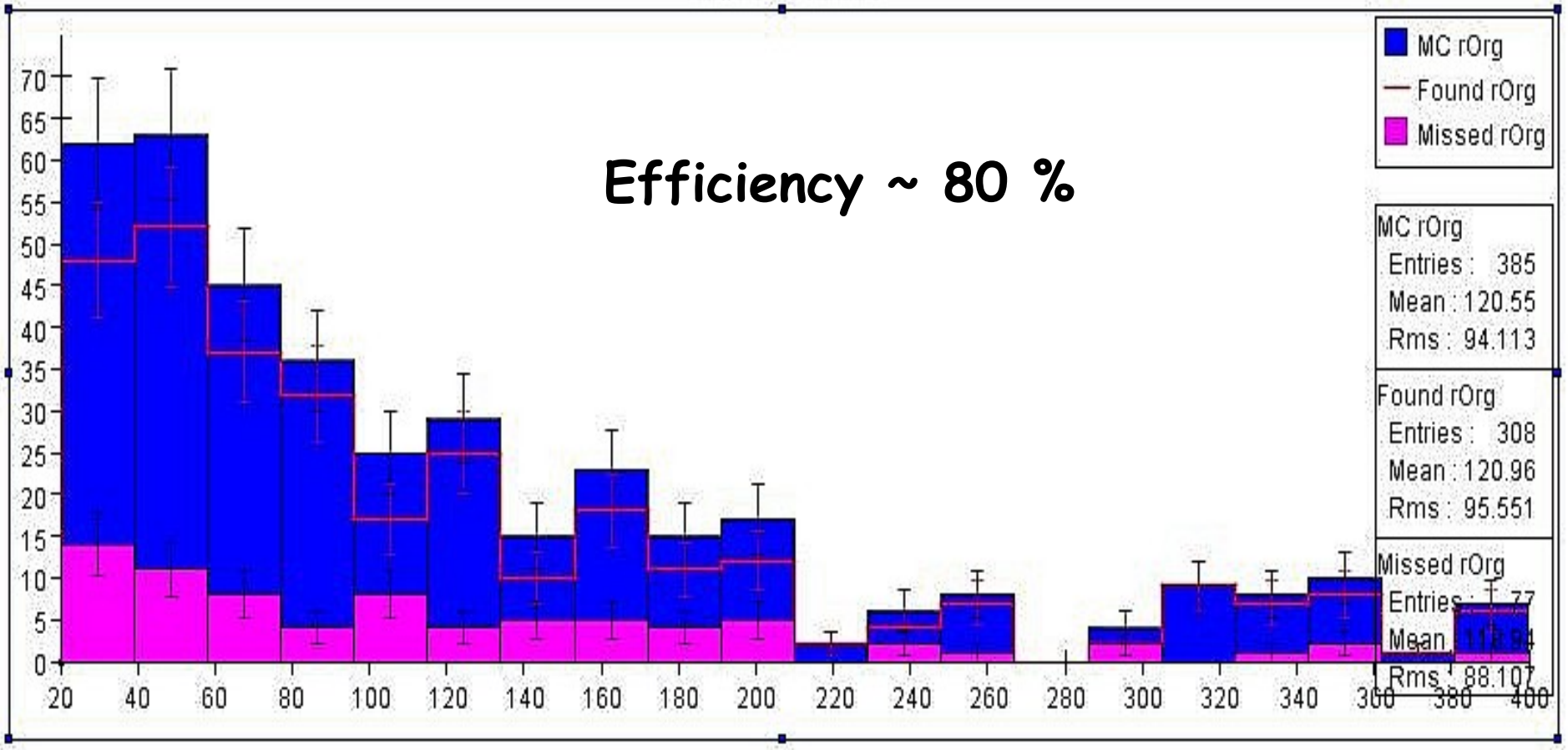
# Distance of Closest Approach < 100 mm




Distribution of Tracks in Pt^-1 (GeV^-1)

**Eric Wallace**

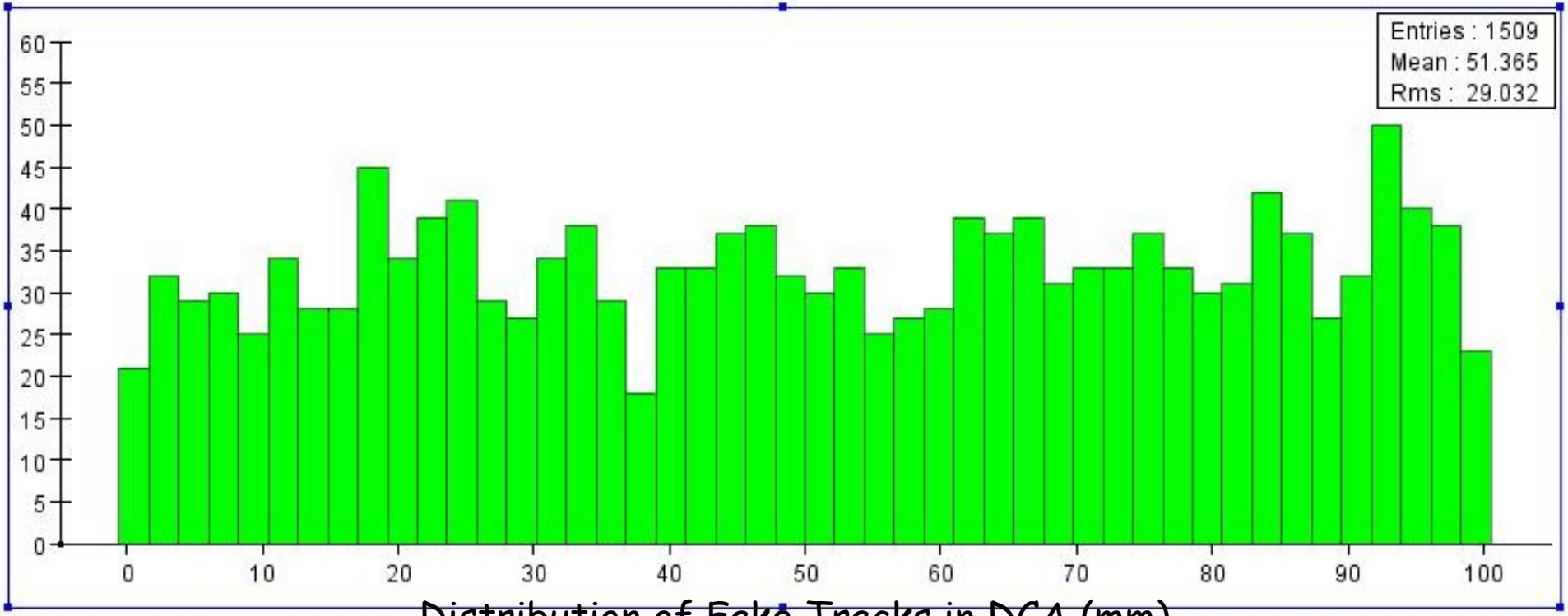
# Distance of Closest Approach < 2.0 mm




Distribution of Tracks in Radial Origin (mm)

**Eric Wallace**

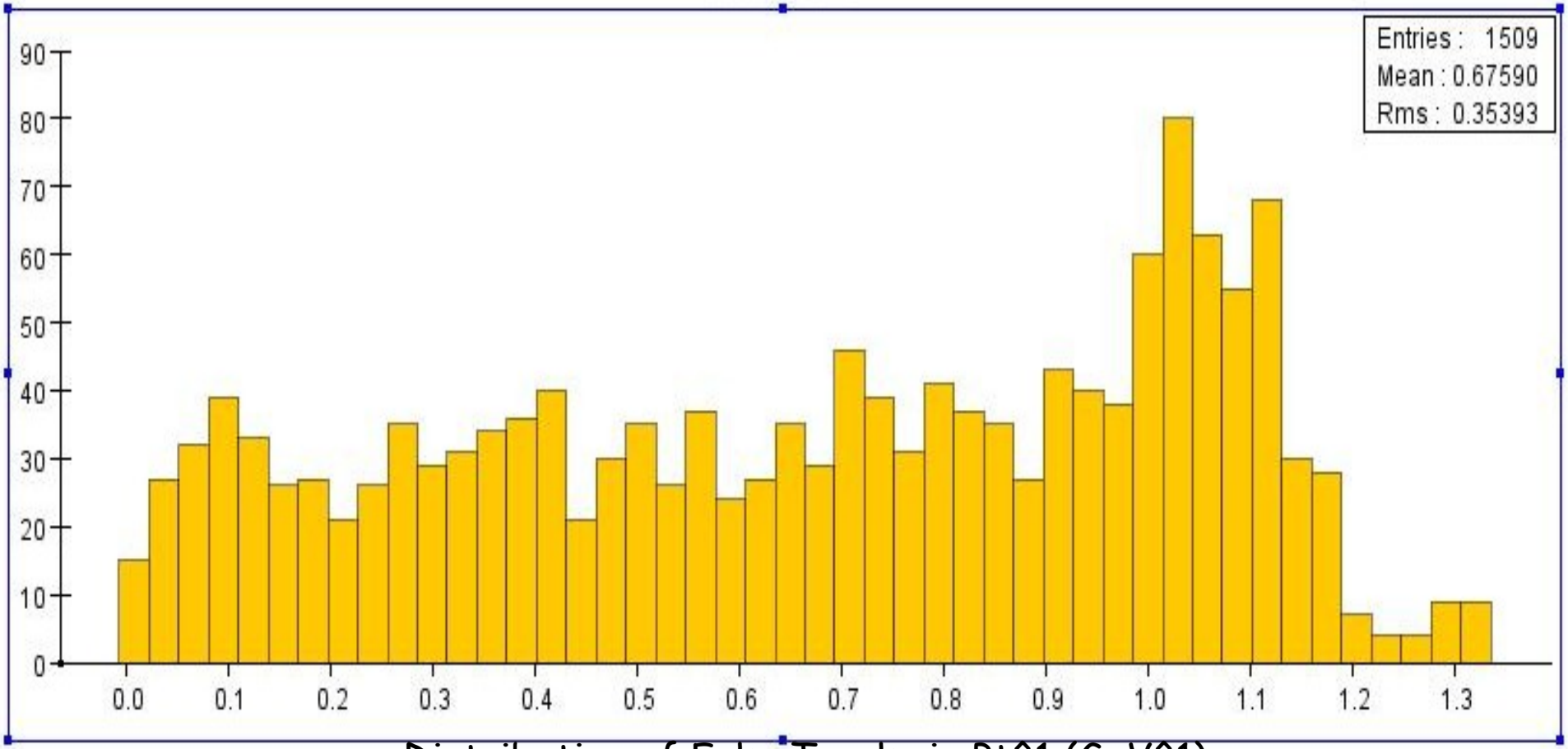
# Distance of Closest Approach < 100 mm




Distribution of Fake Tracks in DCA (mm)

**Eric Wallace**

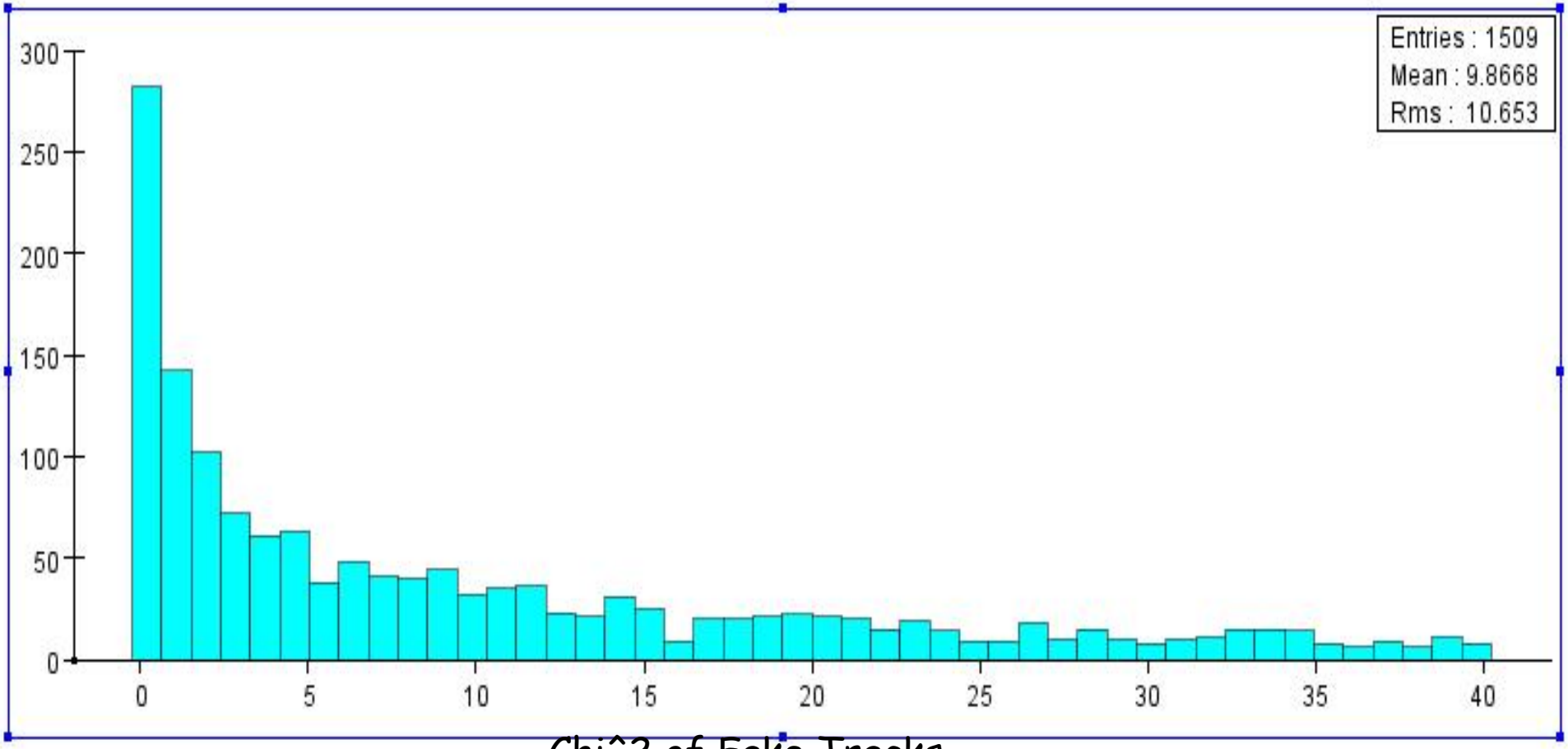
# Distance of Closest Approach < 100 mm




Distribution of Fake Tracks in Pt^1 (GeV^1)

**Eric Wallace**

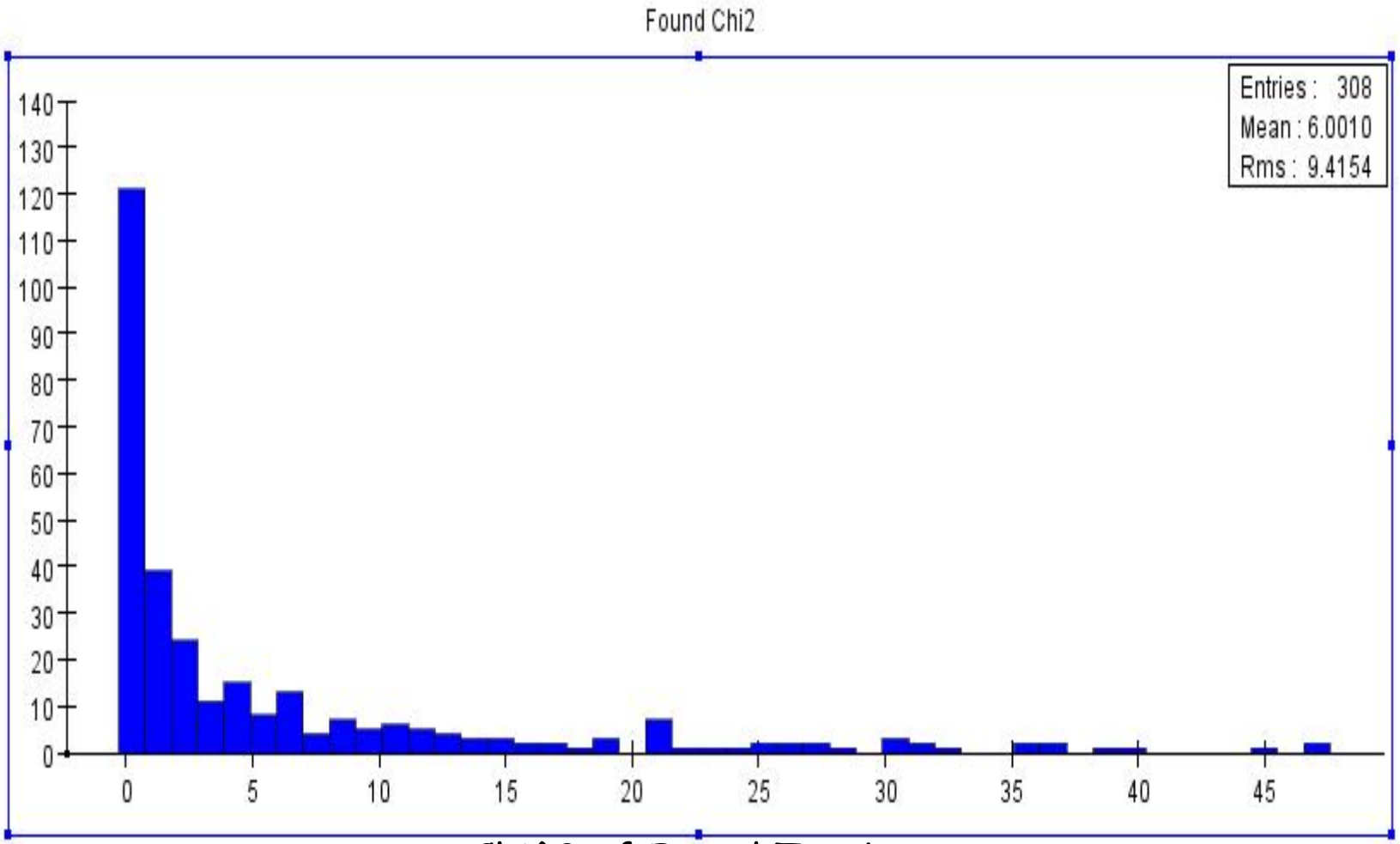
# Distance of Closest Approach < 100 mm




Chi^2 of Fake Tracks

**Eric Wallace**

# Distance of Closest Approach < 100 mm




Chi^2 of Found Tracks

**Eric Wallace**

# Fake Tracks

- Large number of fake tracks compared with found tracks is problematic
- Still trying to find reason for so many fake tracks; they don’t all seem to be coming from loopers.
- Working on developing “true” ChisSq code that should help eliminate them

# Other Issues

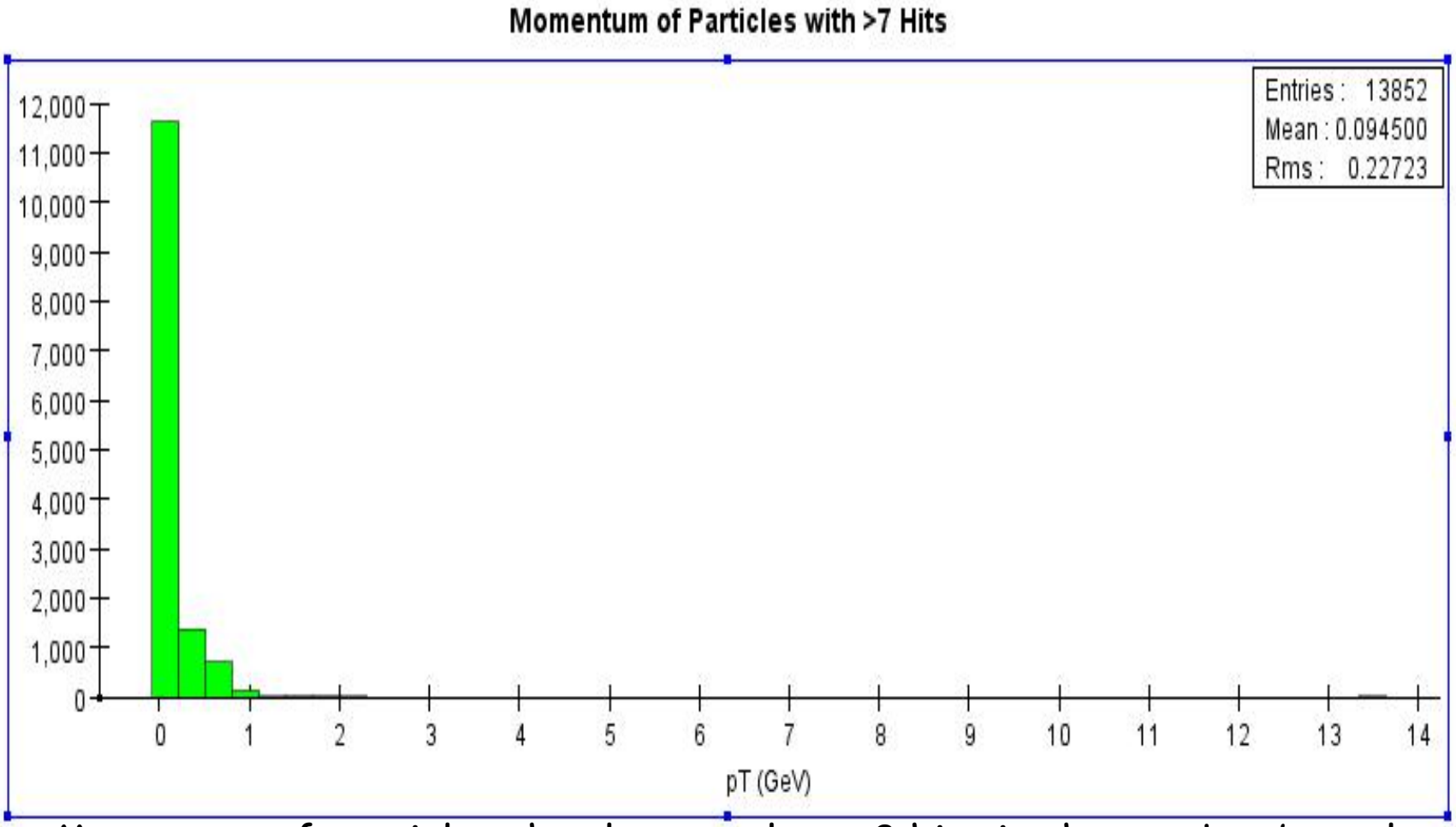
- Total number of “reconstructable” tracks (denominator) seems smaller than 5%; need to understand why (“FinalState” designation?)
- So, results should be considered tentative
- Are developing team of 4 students to continue these studies; hope to move forward again by end of summer

# Conclusions…

Tentatively, using AxialBarrelTrackFinder to clean up non-prompt tracks looks promising, but some questions remain to be answered.

Some thoughts for the future:

- Replace cheater with VXDBasedReco, or similar algorithm
- Implement z segmentation. Is there available code for performing helical fits?
- Try running barrel tracking code in two steps, once with restricted dca cut, again with it opened up
- Modify standalone barrel tracking to add leftover VXD hits to tracks
- Examine potential use of calorimeter-based stubs
- Compare efficiency and fake rates with 5 layers and 8 layers




Momentum of particles that leave at least 8 hits in the tracker barrel

**Eric Wallace**