Tracking in ILCroot with different nozzles

Muon Collider Physics
And
Detector Working Group

Anna Mazzacane
Fermilab

Outline

- Tracking systems for MuX studies
- Nozzles geometries
- Digitization and reconstruction algorithms
- Performance studies: reconstruction efficiency and resolution
- Conclusions

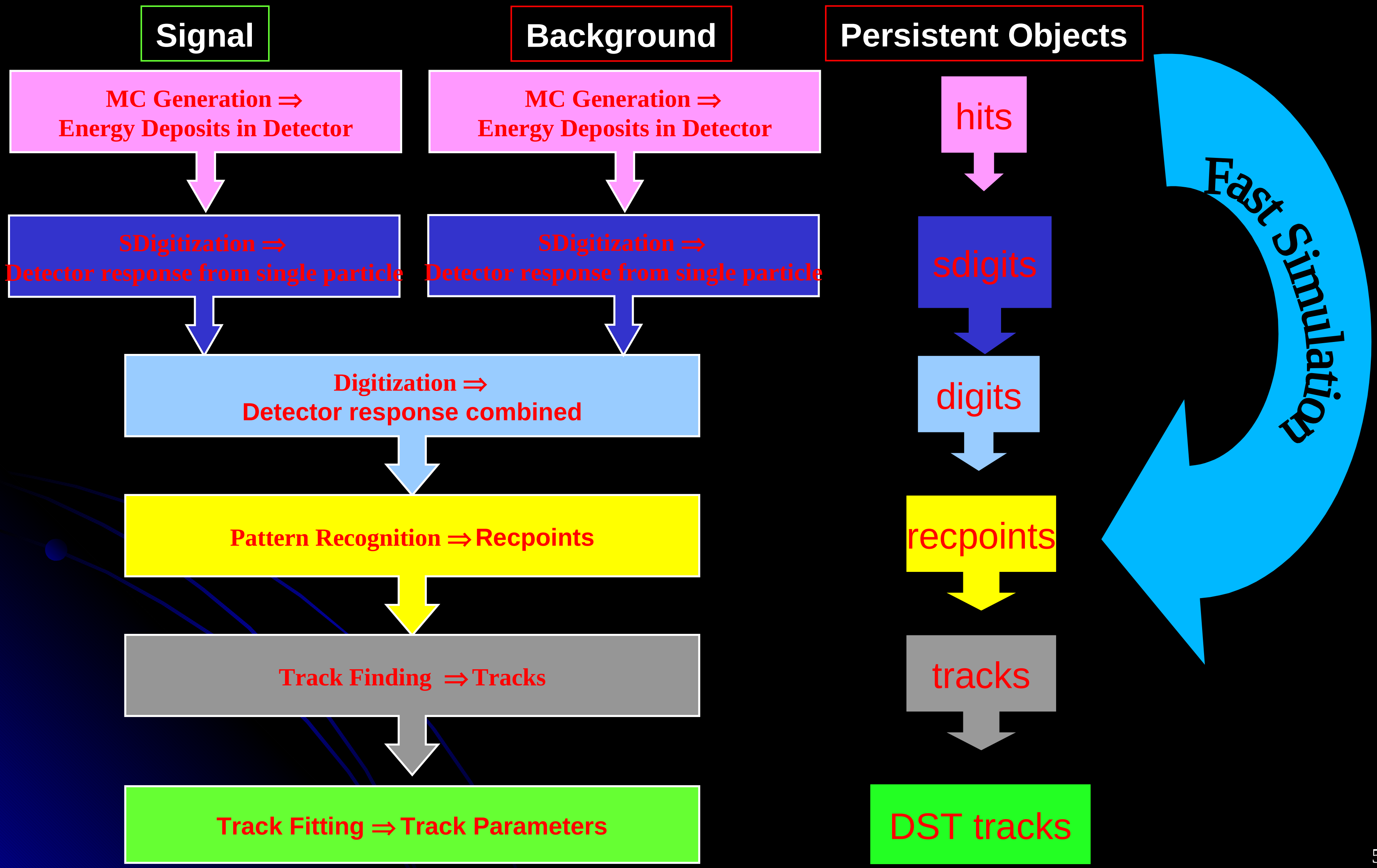
Rationale

- Large background is expected in the tracking detectors at a MuX experiment
  - Pepper-like bkg (mostly from photons)
  - Real tracks trough the detector: (beware of muons from outside)
- What matters is **NOT** the total amount of background but, rather, the ability to reconstruct tracks in a dense environnment of spurious hits
- Two strategies have been implemented in ILCroot:
  - Detector layout with extra redundancy in forward region (7 disks)
  - Full parallel Kalman Filter
- Both have been implemented in ILCroot for CLIC-related studies

ILCroot: root Infrastructure for Large Colliders

- **Software architecture based on root, VMC & Aliroot**
  - All ROOT tools are available (I/O, graphics, PROOF, data structure, etc)
  - Extremely large community of users/developers
- **Re-allignement with latest Aliroot version every 1-2 years (v4.17 release)**
- **It is a simulation framework and an Offline Systems:**
  - **Single framework, from generation to reconstruction through simulation. Don't forget analysis!!!**
  - It is immediatly usable for test beams
  - Six MDC have proven robustness, reliability and portability
- **Main add-ons Aliroot:**
  - Interface to external files in various format (STDHEP, text, etc.)
  - Standalone VTX track fitter
  - Pattern recognition from VTX (for si central trackers)
  - Parametric beam background (# integrated bunch crossing chosen at run time
- Growing number of experiments have adopted it: Alice (LHC), Opera (LNGS), (Meg), CMB (GSI), Panda(GSI), 4th Concept, (SiLC ?) and LHeC
- **It is Publicly available at FNAL on ILCSIM since 2006**
- **Used for ILC, CLIC and Muon Collider studies**

Simulation steps in ILCroot: Tracking system



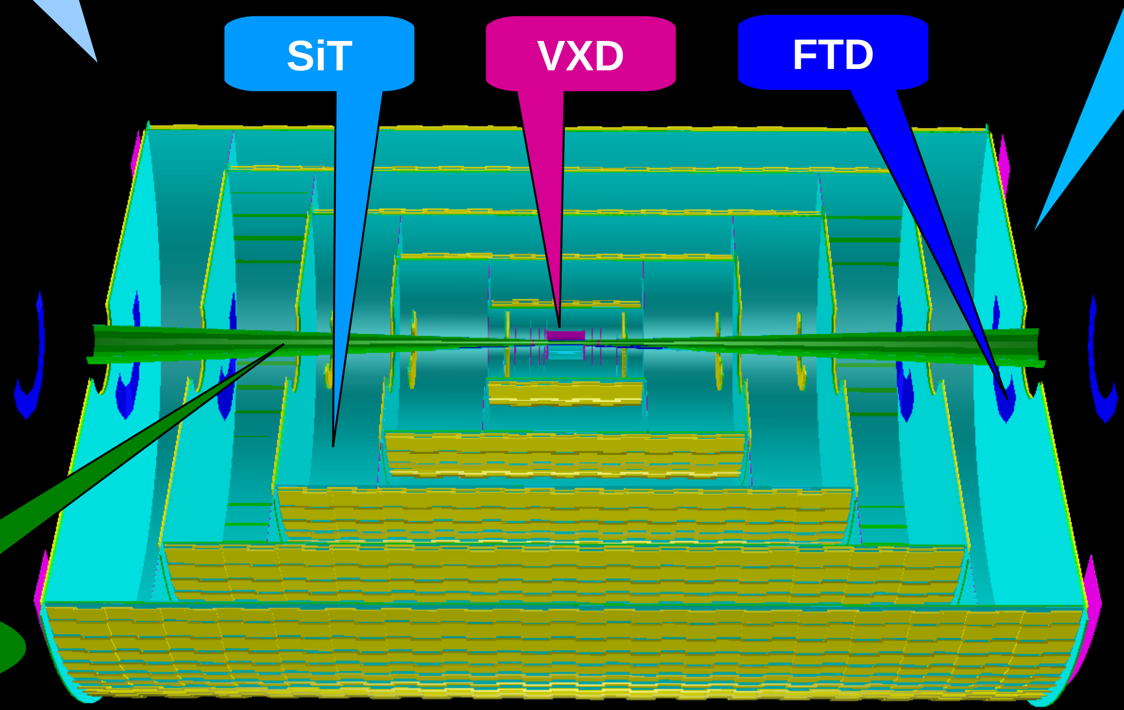
Fast simulation and/or fast digitization also available in ILCroot for tracking system

- Fast Simulation = hit smearing
- Fast Digitization = full digitization with fast algorithms
- Do we need fast simulation in tracking studies?

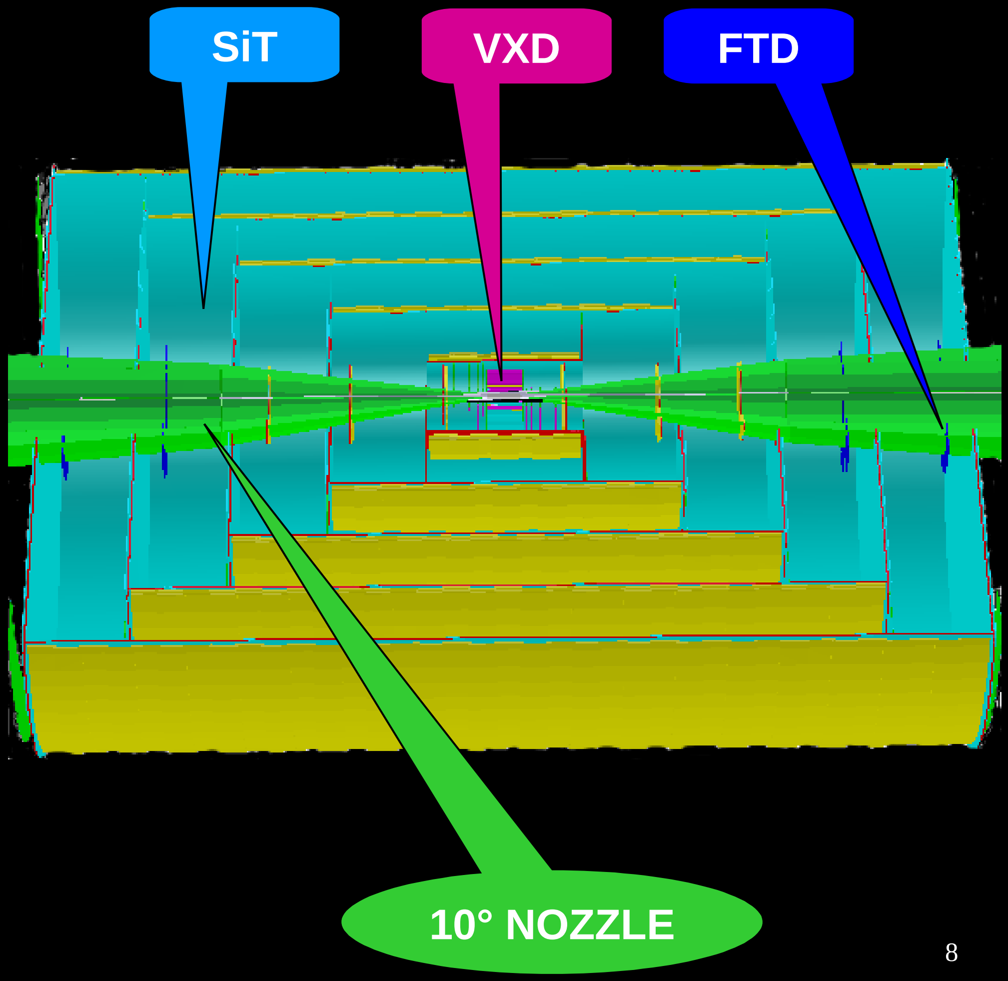
  Yes!
  - Calorimetry related studies do not need full simulation/digitization for tracking
  - Faster computation for quick answer to response of several detector layouts/shielding
- Do we need full simulation in tracking studies?

  Yes!
  - Fancy detector and reconstruction needed to be able to separate hits from signal and background

Tracking detectors for MuX
VXD + SiT + FTD
+ 6° nozzle



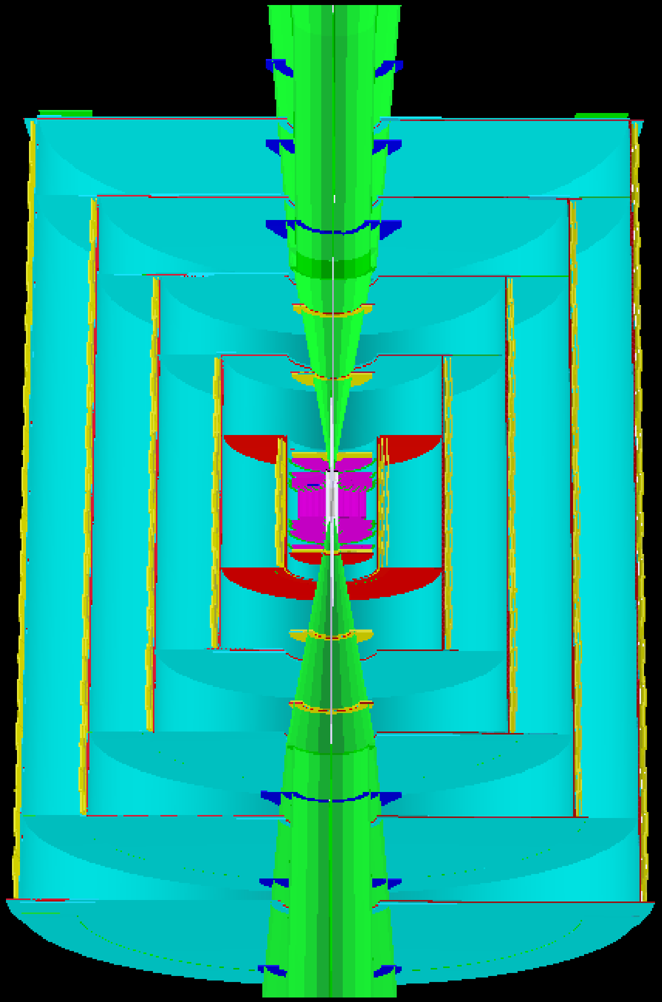
Tracking detectors for MC
VXD + SiT + FTD
+ 10° nozzle



Version	SiD01-Polyhedra + SiD01
Guard ring: mm	0.07
Barrel Layers:	5
Total Tiles Barrel	7312
Wafer layout	
Si wafer	300 mm
Carbonfiber in	0.228 mm
Rohacell tickness	3.175 mm
Carbonfiber out	0.228 mm
Si support	300 mm x 6.667 mm x 63.8 mm
Kapton Layer	0.1 mm
Support layout	
Carbon Fiber	500 mm
Rohacell	8.075 mm
Carbon Fiber	500 mm

Silicon Tracker (SiT) and Forward Tracker Detector (FTD)

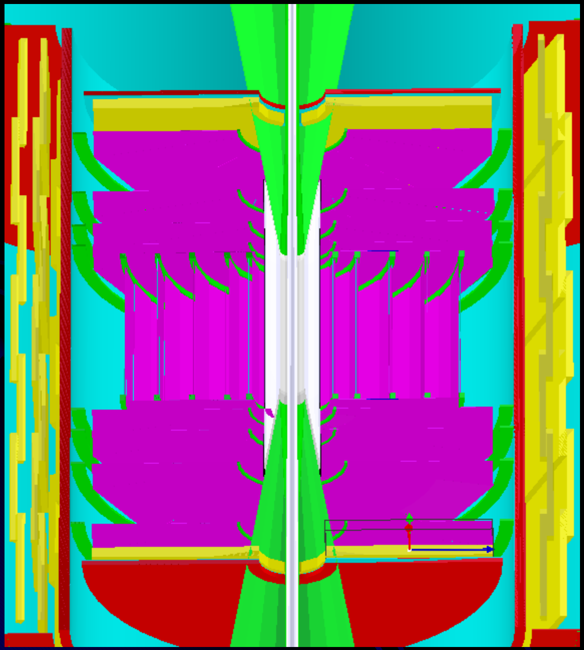
- 50 µm x 50 µm Si pixel (or Si strips or double Si strips available)
- Barrel : 5 layers subdivided in staggered ladders
- Endcap : (4+2) + (4+2) disks Si pixel
- FTD: 3 + 3 disks Si pixel

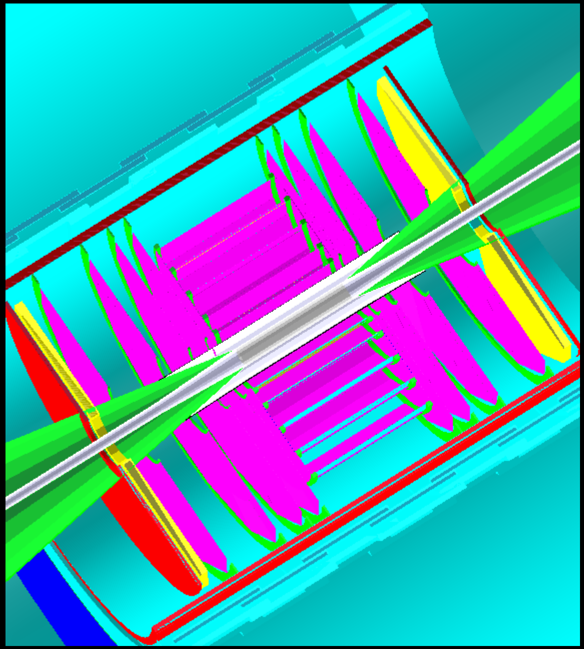


- Mostly SiD layout + FTD
- Not parametrized geometry yet

Vertex Detector (VXD) Nozzle and Beam Pipe

- 20 µm x 20 µm Si pixel
- Barrel : 5 layers subdivided in 12- 30 ladders
- Endcap : 4 + 4 disks subdivided in 12 ladders

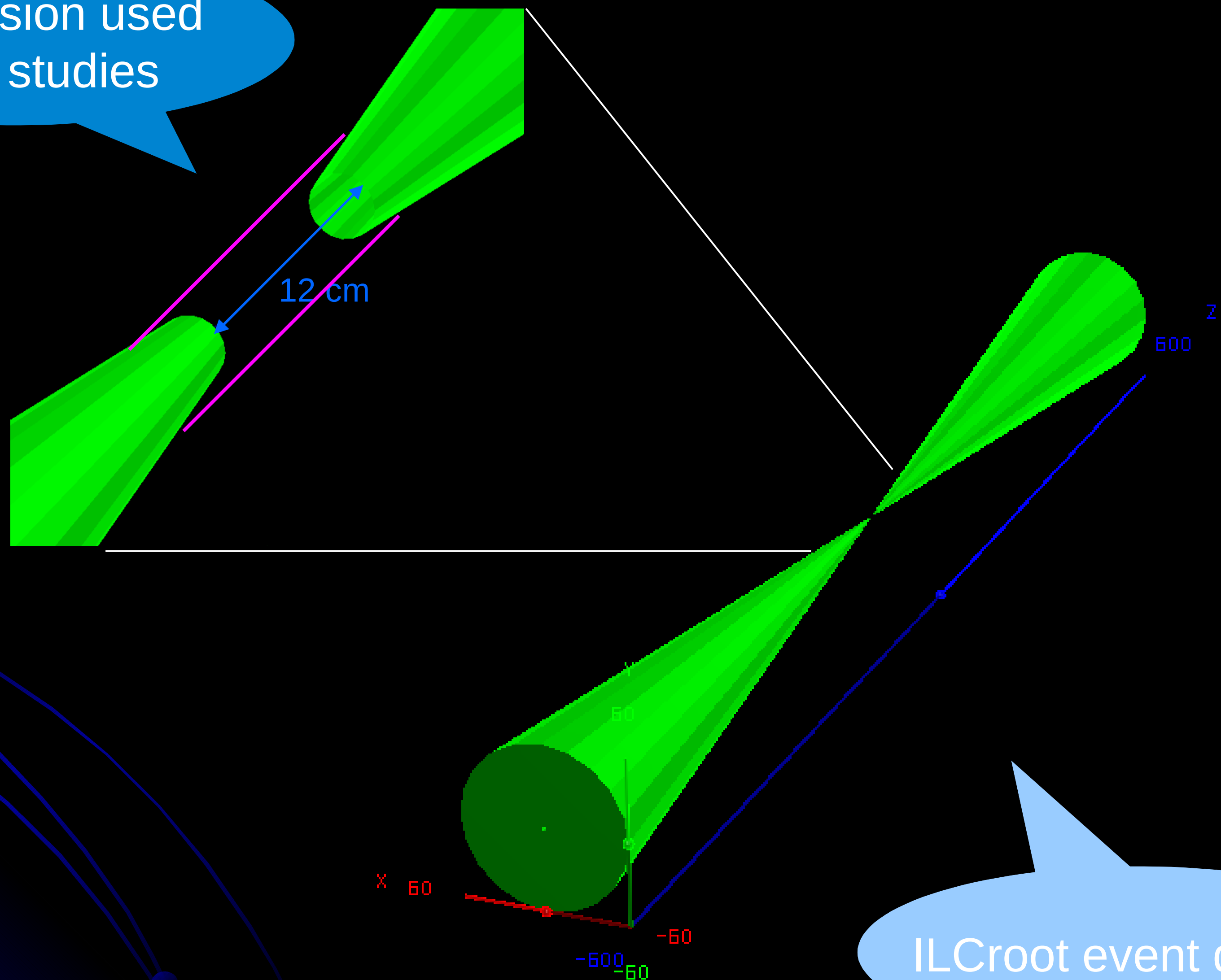




- Mostly SiD layout
- Different dimensions (different B field = 3.5 T)
- Full parametrized geometry

6° Nozzle

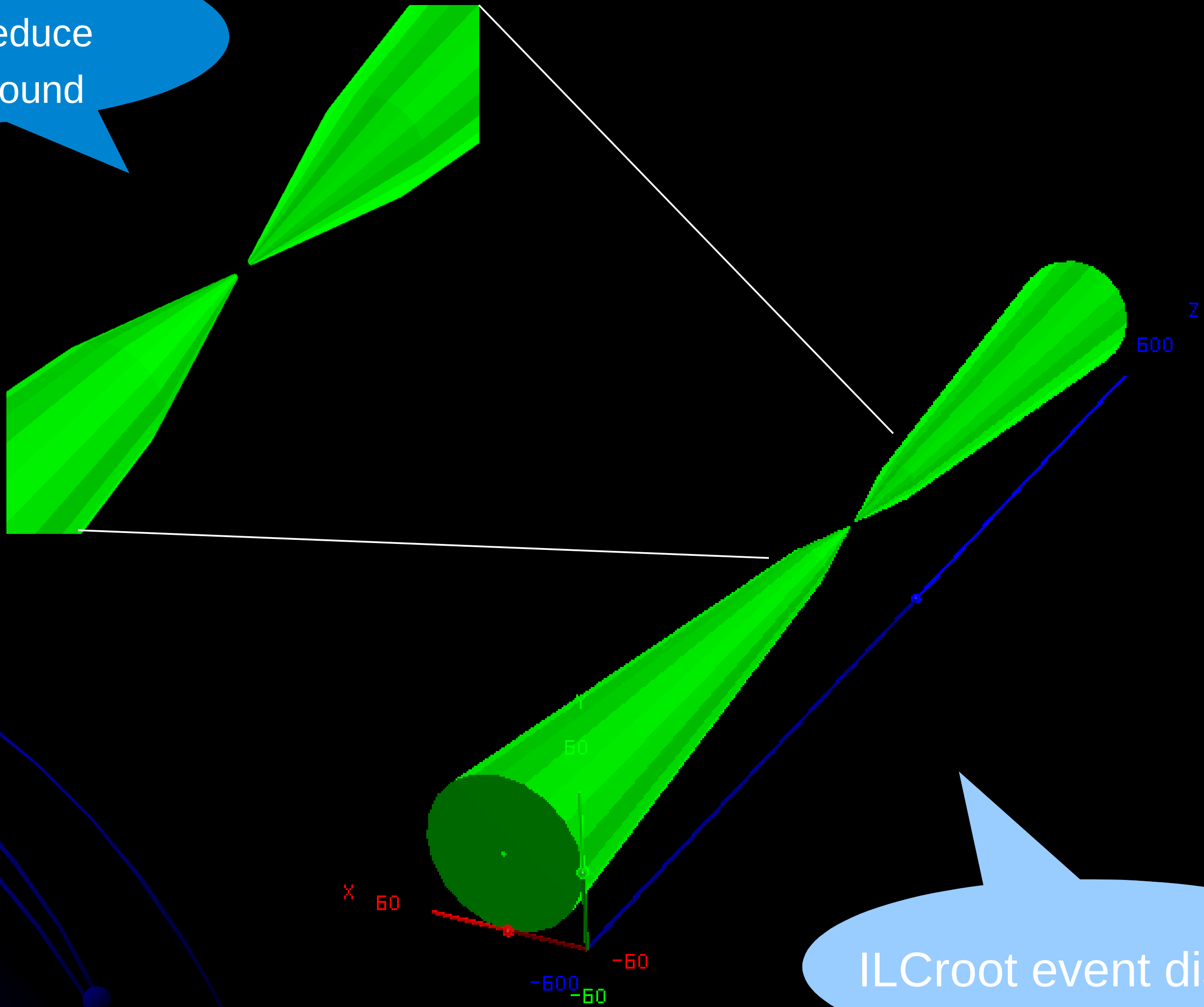
First version used
In MuX studies



ILCroot event display

10° Nozzle

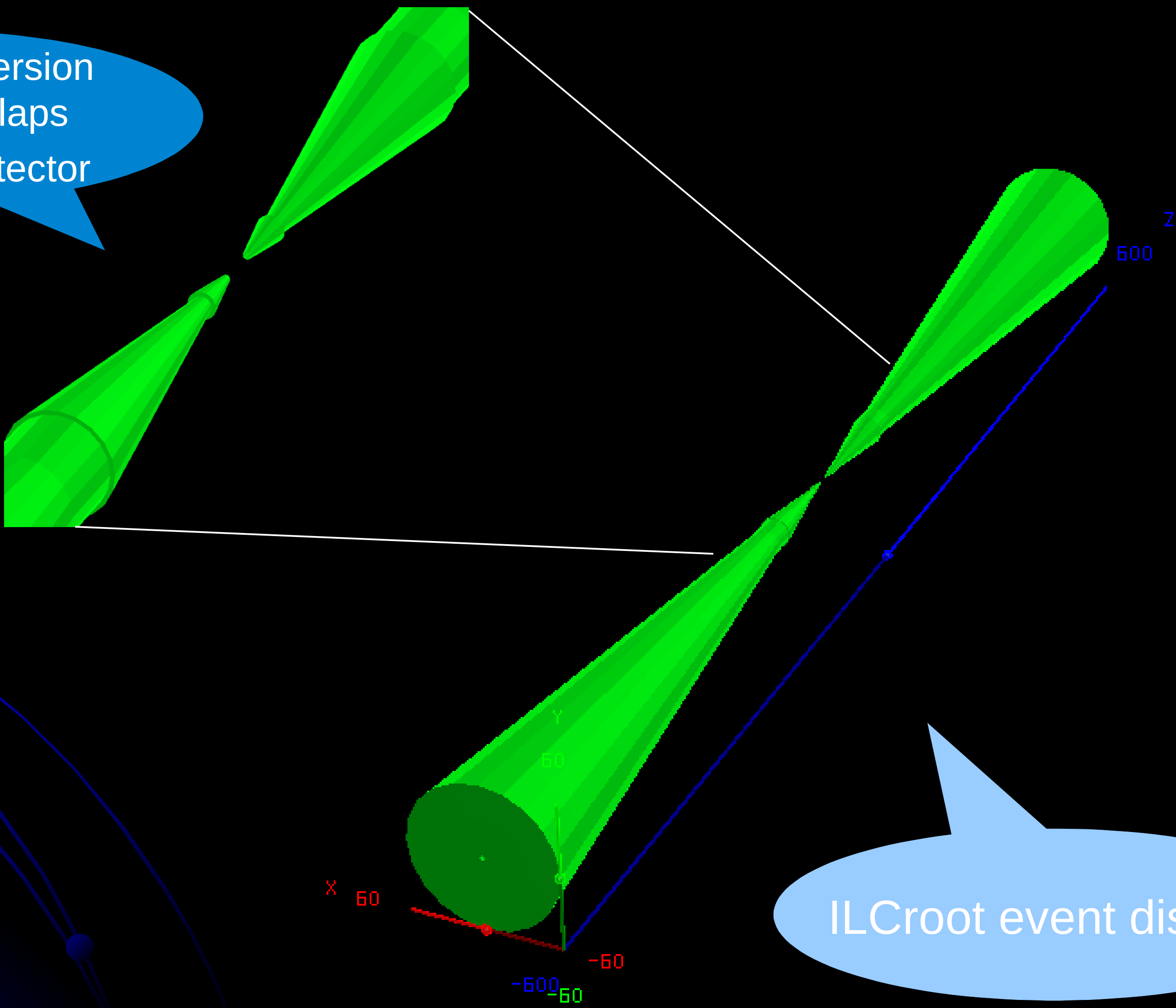
Newer version
To further reduce
MuX background



ILCroot event display

10° Skinned Nozzle

N. Tierentev version
To avoid overlaps
with tracking detector



ILCroot event display

Digitization and Clusterization of Si Detectors in Ilcroot: a description of the algorithms available for detailed tracking simulation and studies

Technologies Implemented

- 3 detector species:
  - Silicon pixels
  - Silicon Strips
  - Silicon Drift

Used for VXD SiT and FTD in present studies

- Pixel can have non constant size in different layers
- Strips can also be stereo and on both sides
- Dead regions are taken into account
- Algorithms are parametric: almost all available technologies are easily accomodated (MAPS, 3D, DEPFET, etc.)

SDigitization in Pixel Detector (production of summable digits)

- Summable digit = signal produced by each individual track in a pixel
- Loop over the hits produced in the layer and create a segment in Si in 3D
  - Step (from MC) along the line >1 µm increments
    - Convert GeV to charge and get bias voltage:

      q = dE*dt/3.6e-9   dV= thick/bias voltage
    - Compute charge spreading:

      σ_{xy}=sqrt(2k/e*T°*dV*L), σ_z=fda*σ_{xy}
    - Spread charge across pixels using Erfc(xy,z,σ_{xy},σ_z)
  - Charge pile-up is automatically taken into account

SDigitization in Pixels (2)

- Add couplig effect between nearby pixels row-wise and column-wise (constant probability)
- Remove dead pixels (use signal map)

Digitization in Pixels

Digit = sum of all sdigit corresponding to the same pixel

- Load SDigits from several files (signal or multiple background)
- Merge signals belonging to the same pixel
  - Non-linearity effects
  - Saturation
- Add electronic noise
- Save Digits over threshold

Clusterization in Pixel Detector

Cluster = a collection of nearby digit

Create a initial cluster from adjacent pixels (no for diagonal)

Subdivide the previous cluster in smaller NxN clusters

Reconstruct cluster and error matrix from coordinate average of the cluster

Kalman filter picks up the best cluster

Parameters used for the pixel tracking detectors in current MuX studies

Size Pixel X = 20 μm (VXD and FTD), 50 μm (SiT)

Size Pixel Z = 20 μm (VXD and FTD), 50 μm (SiT)

Eccentricity = 0.85 (fda)

Bias voltage = 18 V

cr = 0% (coupling probability for row)

cc = 4.7% (coupling probability for column)

threshold = 3000 electrons

electronics noise = 0 electrons

T° = 300 °K

Track Fitting in ILCRoot

Track finding and fitting is a global tasks: individual detector collaborate

It is performed after each detector has completed its local tasks (simulation, digitization, clusterization)

It occurs in three phases:

1. Seeding in SiT and fitting in VXD+SiT+MUD
2. Standalone seeding and fitting in VXD
3. Standalone seeding and fitting in MUD

Two different seedings:

A. Primary seeding with vertex constraint
B. Secondary seeding without vertex constraint

Not yet implemented

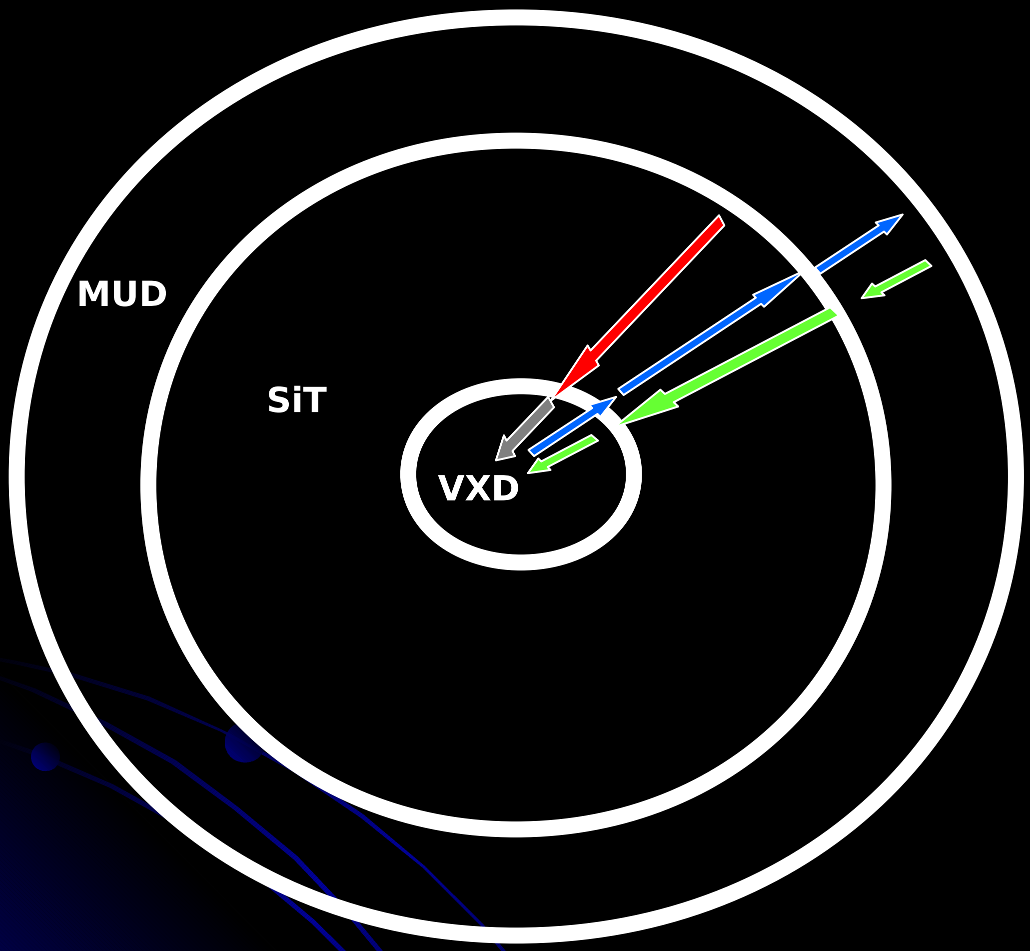
Kalman Filter (classic)

- Recursive least-squares estimation.
- Equivalent to global least-squares method including all correlations between measurements due to multiple scattering.
- Suitable for combined track finding and fitting
- Provides a natural way:
  - to take into account multiple scattering, magnetic field inhomogeneity
  - possibility to take into account mean energy losses
  - to extrapolate tracks from one sub-detector to another

Parallel Kalman Filter

- Seedings with constraint + seedings without constraint at different radii (necessary for kinks and V0) from outer to inner
- Tracking
  - Find for each track the prolongation to the next layer
  - Estimate the errors
  - Update track according current cluster parameters
  - (Possible refine clusters parameters with current track)
- Track several track-hypothesis in parallel
  - Allow cluster sharing between different track
- Remove-Overlap
- **Kinks and V0** fitted during the Kalman filtering

Tracking Strategy – Primary Tracks



- Iterative process
  - Seeding in SiT
  - Forward propagation towards to the vertex
    SiT → VXD
  - Back propagation towards to the MUD
    VXD → SiT → MUD
  - Refit inward
    MUD → SiT → VXD
- Continuous seeding –track segment finding in all detectors

VXD Standalone Tracking

- Uses Clusters leftover in the VXD by Parallel Kalman Filter
- **Requires at least 4 hits to build a track**
- Seeding in VXD in two steps
  - Step 1: look for 3 Clusters in a narrow row or 2 Clusters + IP constraint
  - Step 2: prolongate to next layers each helix constructed from a seed
- After finding Clusters, all different combination of clusters are refitted with the Kalman Filter and the tracks with lowest χ^2 are selected
- Finally, the process is repeated attempting to find tracks on an enlarged row constructed looping on the first point on different layers and all the subsequent layers
- In 3.5 Tesla B-field $P_t > 20$ MeV tracks reconstructable

Performance studies

Resolution
and
Total Reconstruction Efficiency:
Tracking and Geometrical efficiency

$$\epsilon_{tot} = \frac{reconstructed\ tracks}{generated\ tracks} = \epsilon_{geom} * \epsilon_{track}$$

$$\epsilon_{geom} = \frac{good\ tracks}{generated\ tracks} \qquad \epsilon_{track} = \frac{reconstructed\ tracks}{good\ tracks}$$

Defining "good tracks" (candidate for reconstruction)

DCA(true) < 3.5 cm

AND

at least 4 hits in the detector

Performance studies

20000 events of 10muons single tracks

P: [0,200] GeV

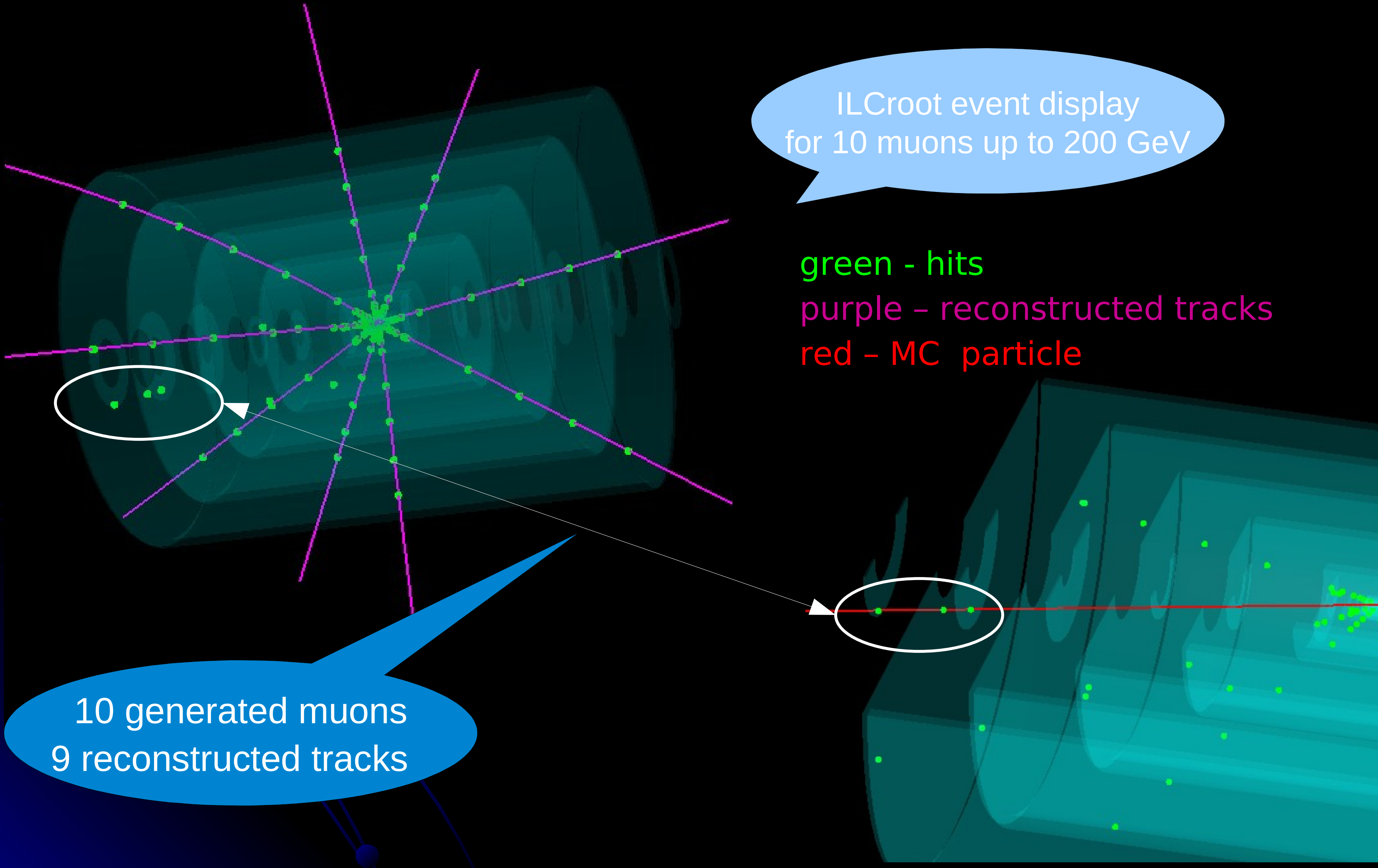
θ: [0,180] Degrees

Φ: [0,360] Degrees

B: 3.5 Tesla

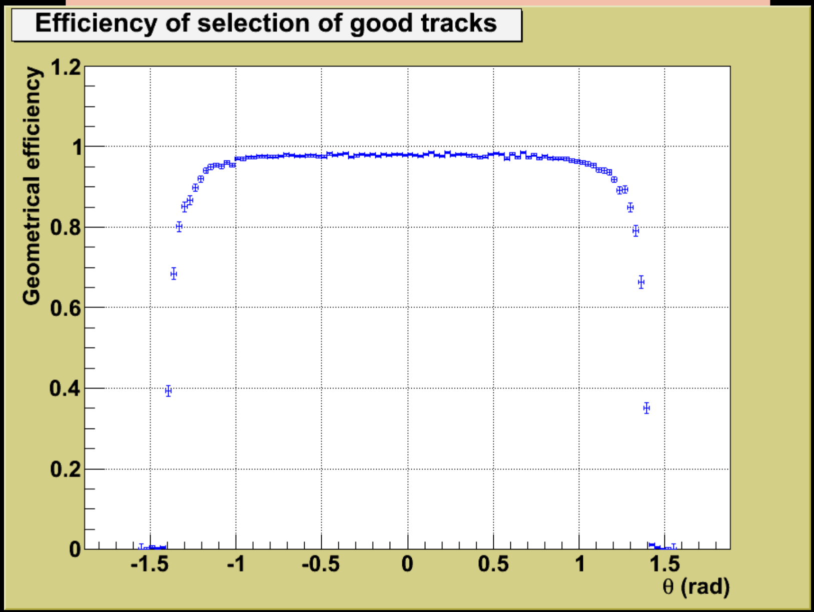
Compare:
6° Nose (March 2010)
vs
10° Nose (Oct. 2010)
vs
skinned 10° Nose
(temporary version by N. Terentiev)

Event Display

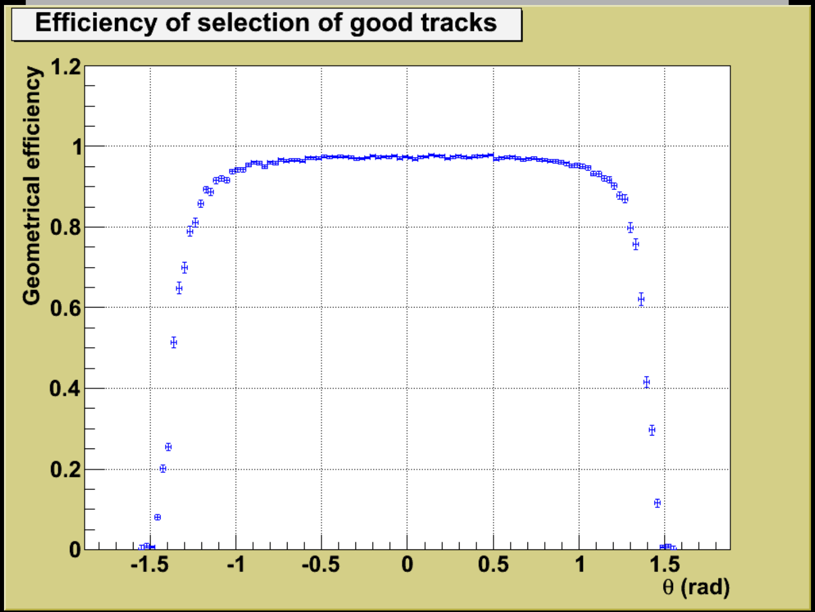


Geometrical Efficiency vs Theta

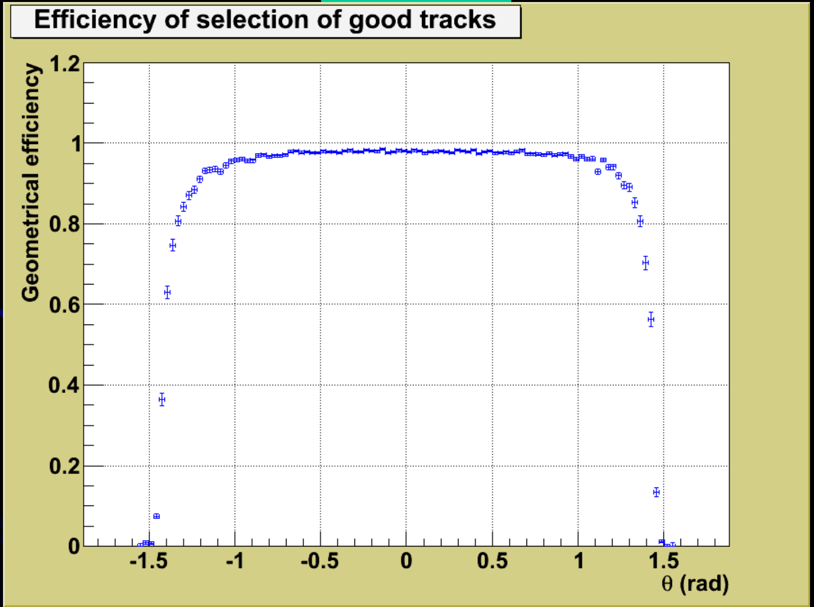
10° Nose with detector interference



10° Nose with no detector interference



6° Nose



$$\epsilon_{geom} = \frac{good\ tracks}{generated\ tracks}$$

Single muons

Defining "good tracks"

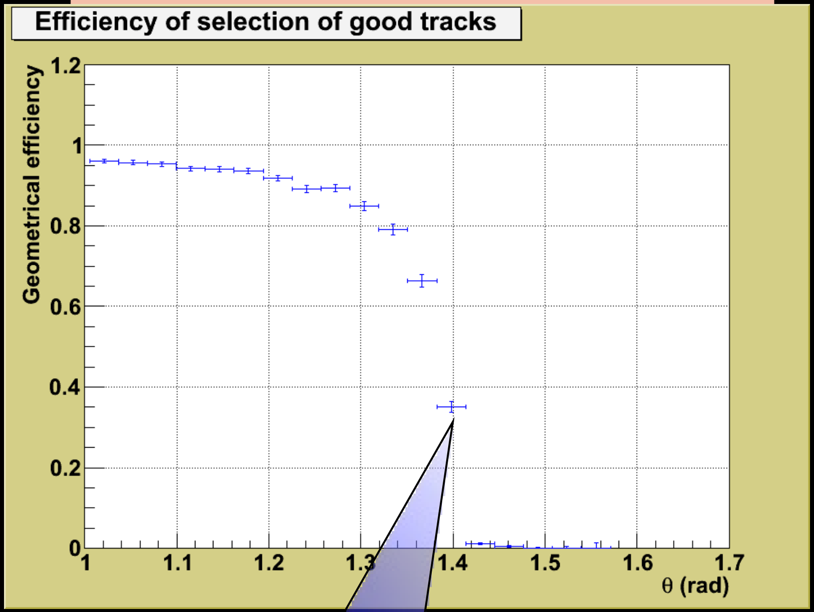
DCA(true) < 3.5 cm

AND

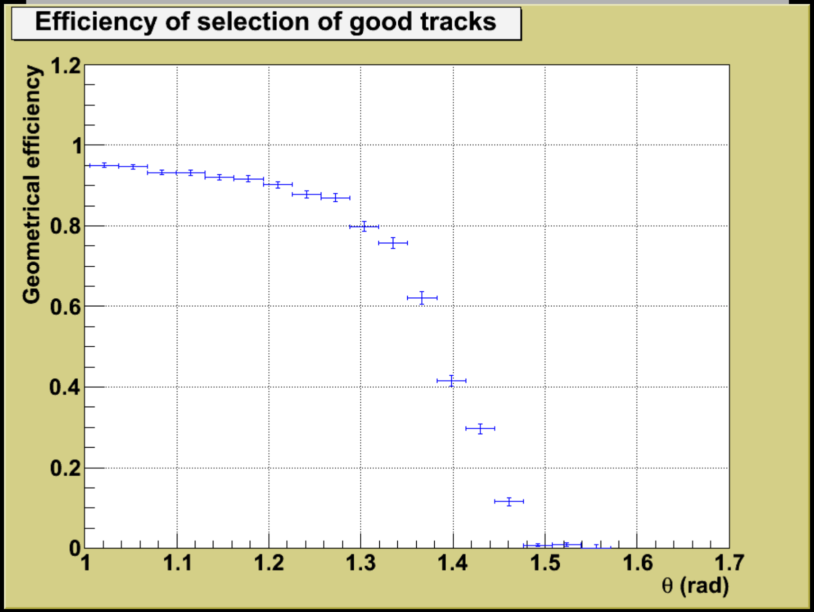
at least 4 hits in detector

Geometrical Efficiency vs Theta (zoom)

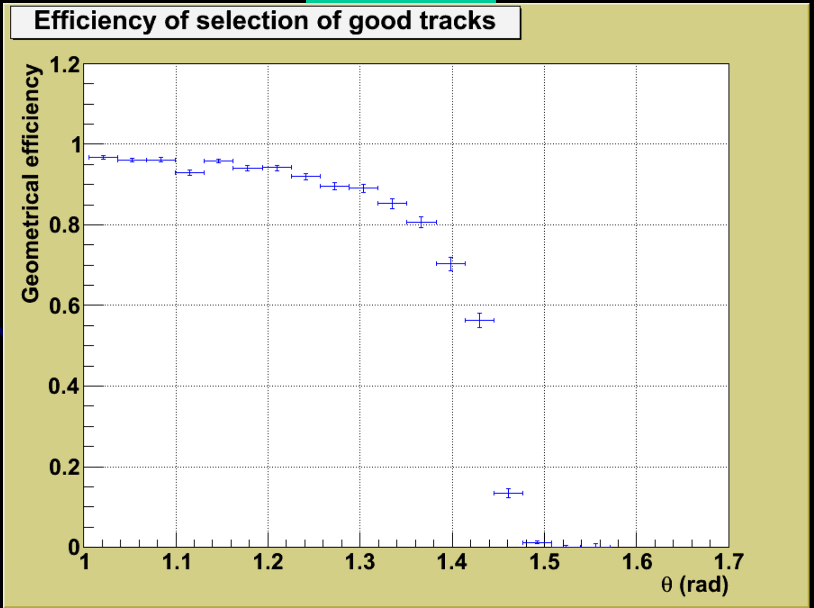
10° Nose with detector interference



10° Nose with no detector interference



6° Nose



Slightly lower geometrical efficiency

$$\epsilon_{geom} = \frac{good\ tracks}{generated\ tracks}$$

Defining "good tracks"

DCA(true) < 3.5 cm

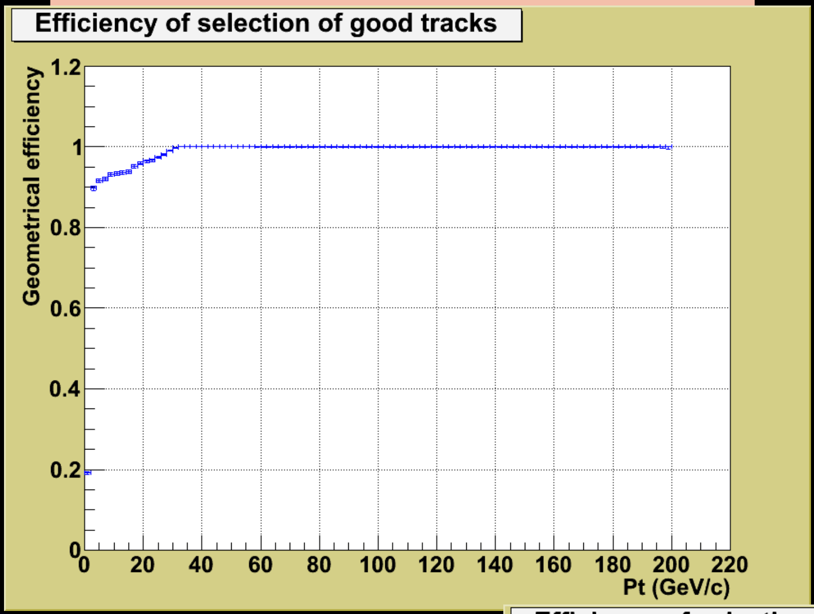
AND

at least 4 hits in detector

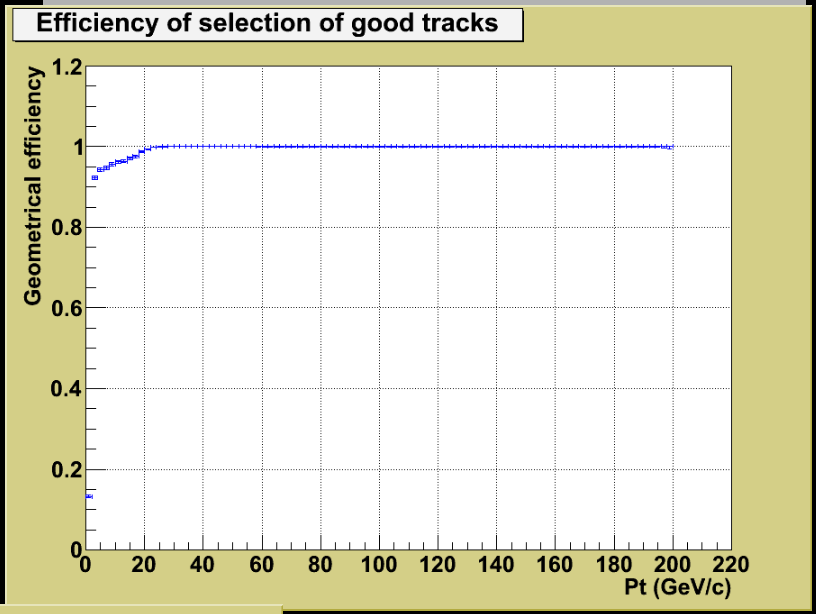
Single muons

Geometrical Efficiency vs Pt

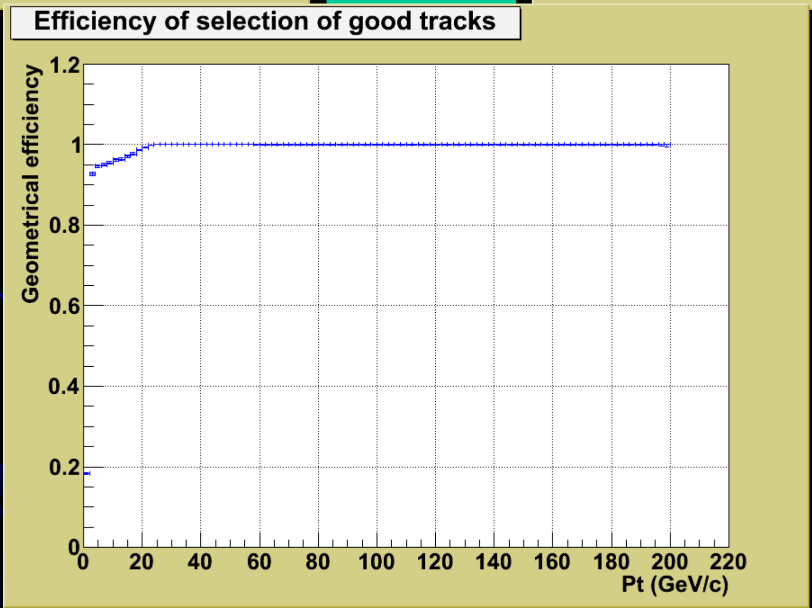
10° Nose with detector interference



10° Nose with no detector interference



6° Nose



$$\epsilon_{geom} = \frac{good\ tracks}{generated\ tracks}$$

Single muons

Defining "good tracks"

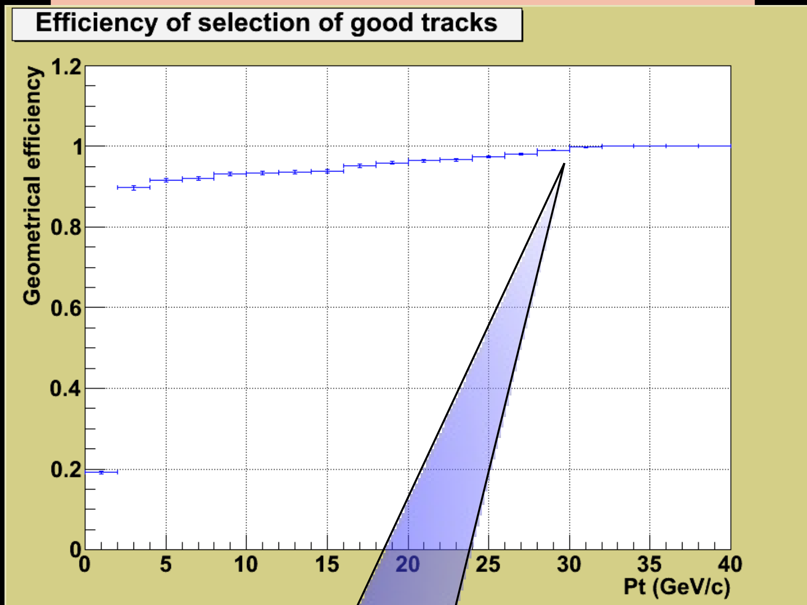
DCA(true) < 3.5 cm

AND

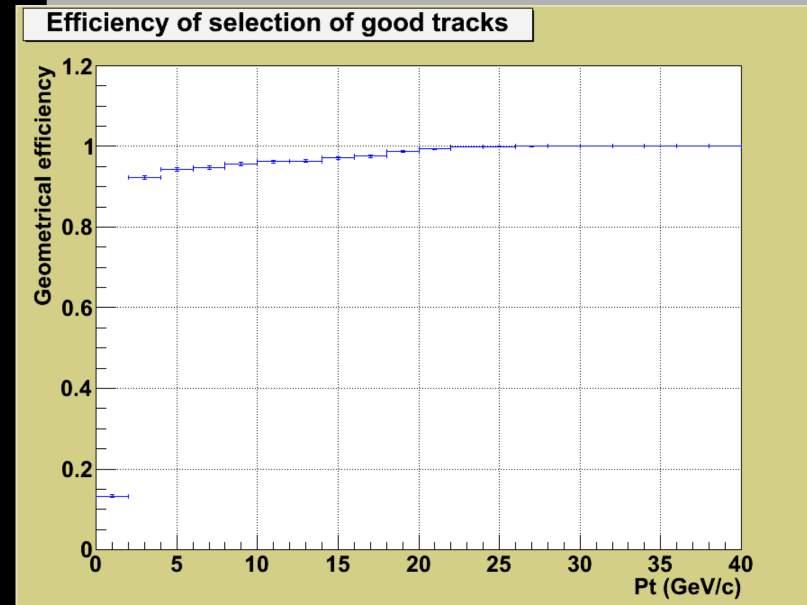
at least 4 hits in detector

Geometrical Efficiency vs Pt (zoom)

10° Nose with detector interference

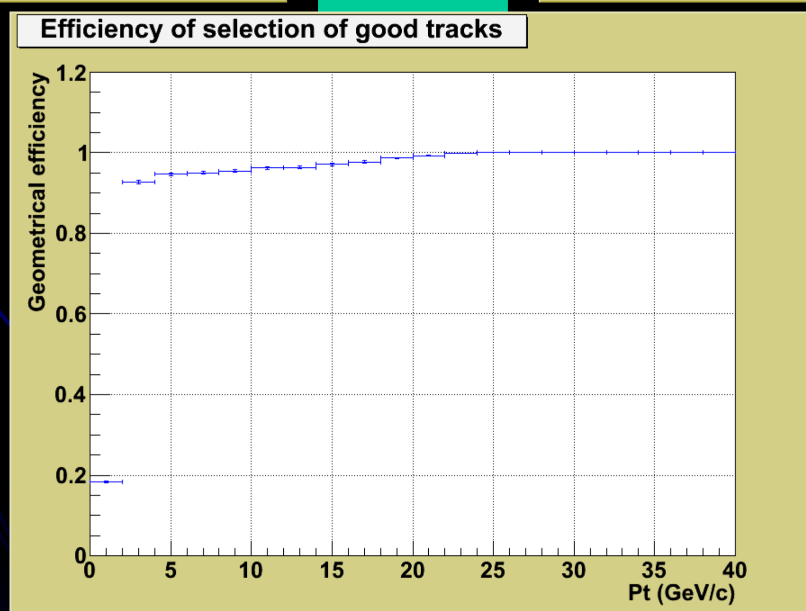


10° Nose with no detector interference



Slightly lower geometrical efficiency

6° Nose



Single muons

$$\epsilon_{geom} = \frac{good\ tracks}{generated\ tracks}$$

Defining "good tracks"

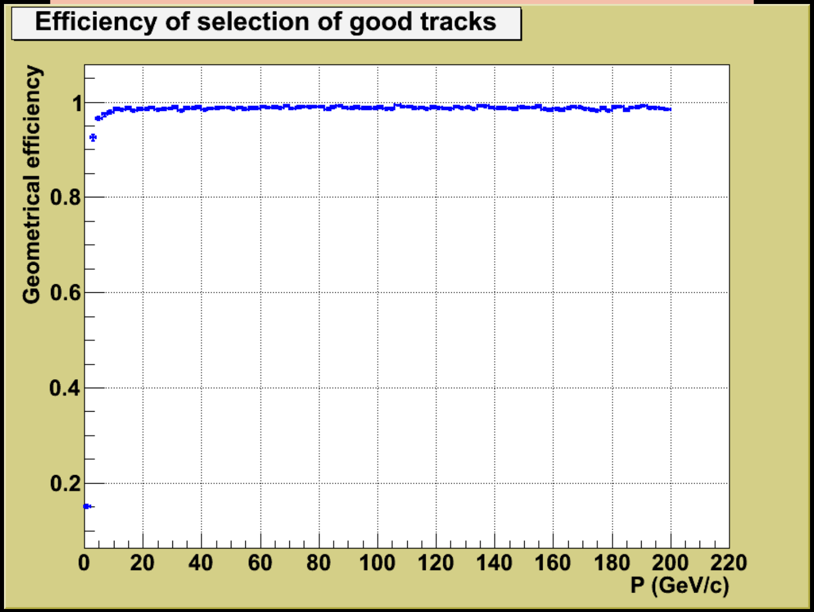
DCA(true) < 3.5 cm

AND

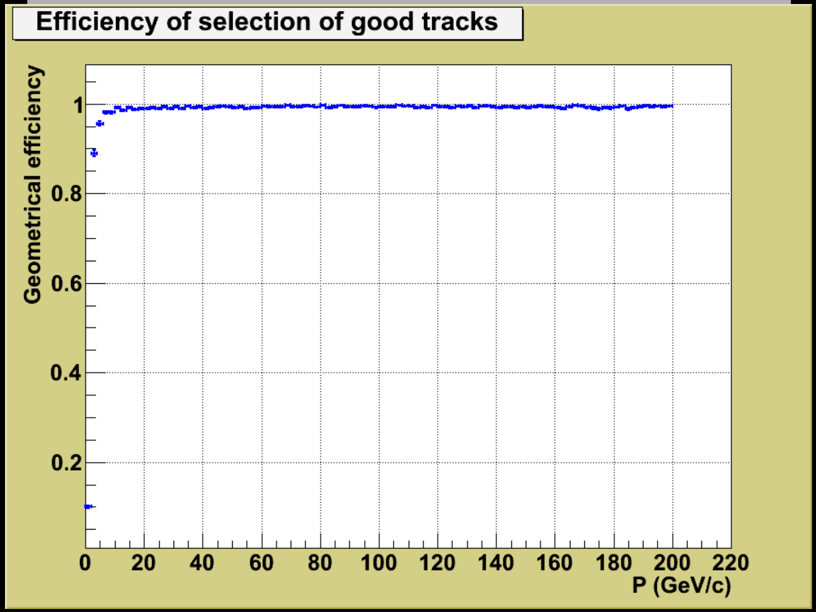
at least 4 hits in detector

Geometrical Efficiency vs P

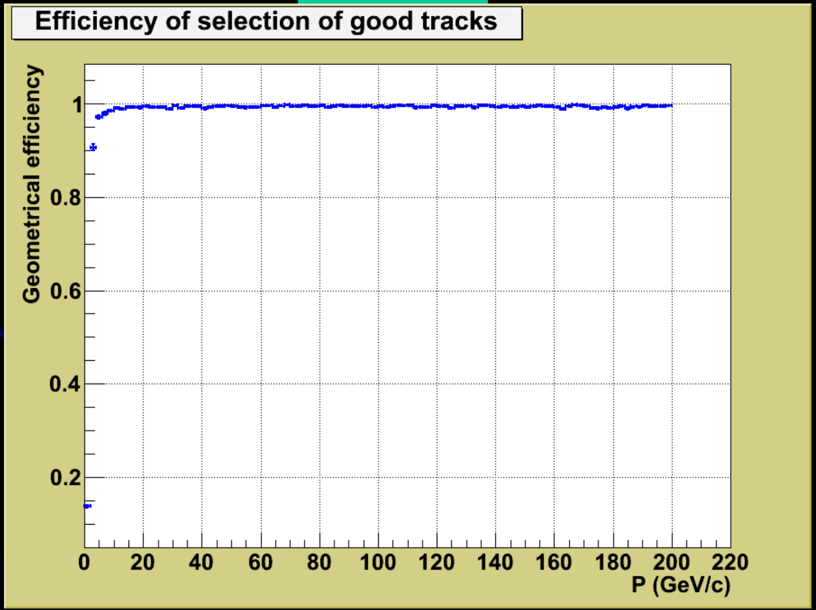
10° Nose with detector interference



10° Nose with no detector interference



6° Nose



$$\epsilon_{geom} = \frac{good\ tracks}{generated\ tracks}$$

Single muons

Defining "good tracks"

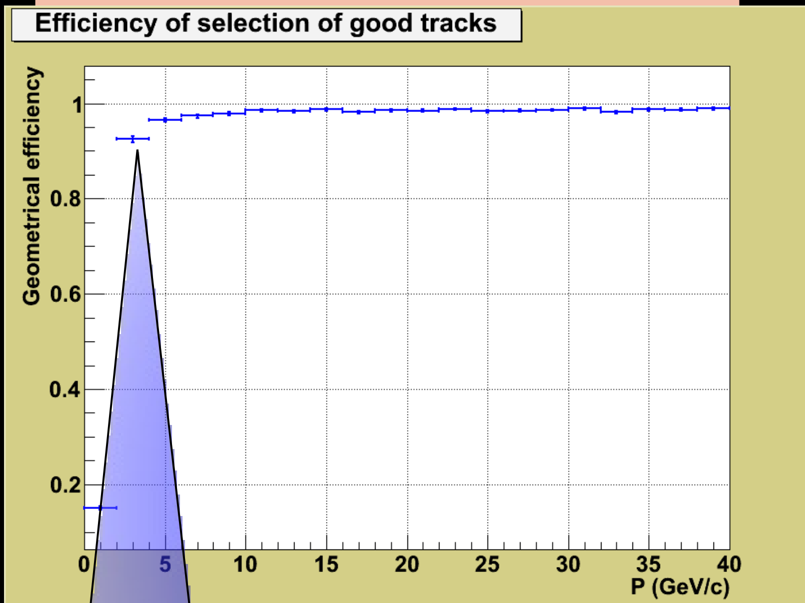
DCA(true) < 3.5 cm

AND

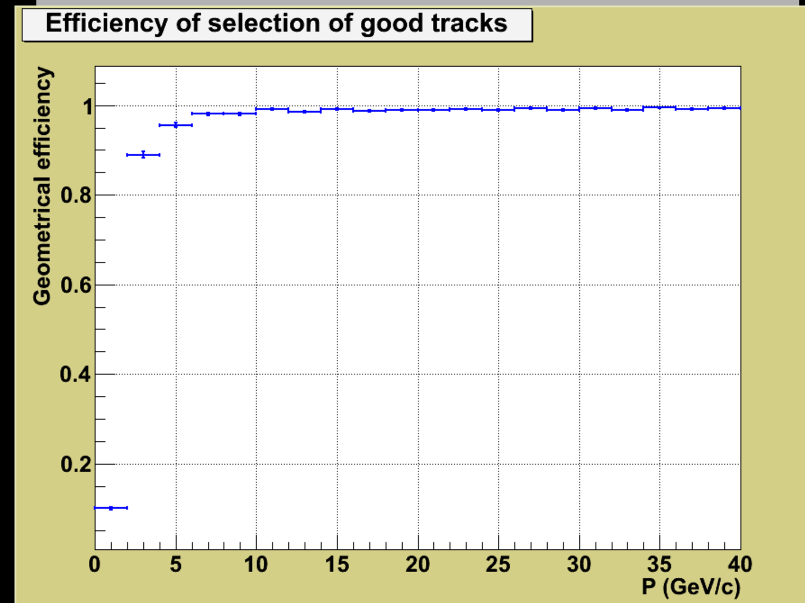
at least 4 hits in detector

Geometrical Efficiency vs P (zoom)

10° Nose with detector interference

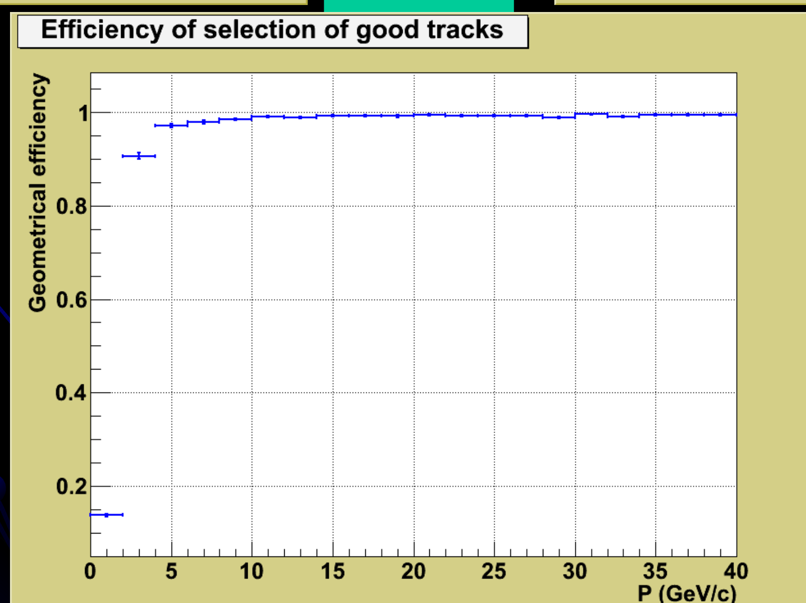


10° Nose with no detector interference



Slightly lower geometrical efficiency

6° Nose



$$\epsilon_{geom} = \frac{good\ tracks}{generated\ tracks}$$

Single muons

Defining "good tracks"

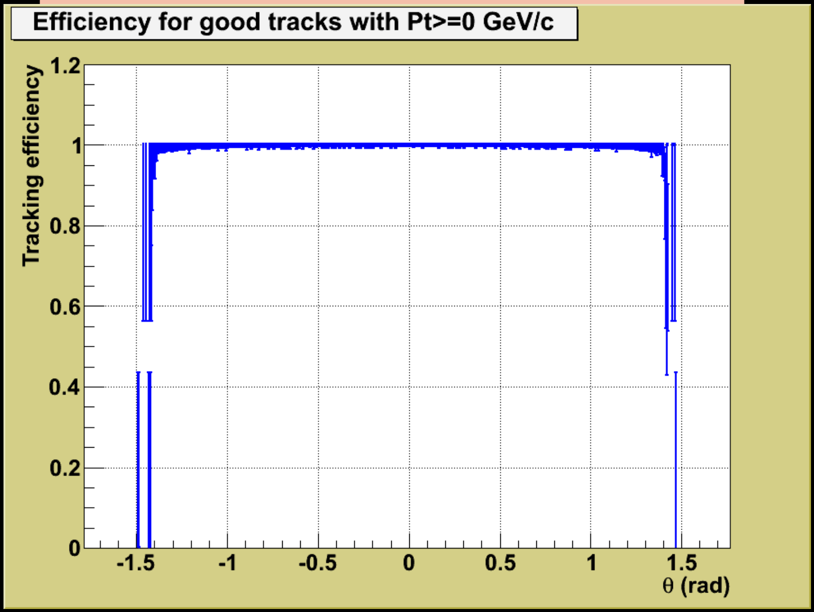
DCA(true) < 3.5 cm

AND

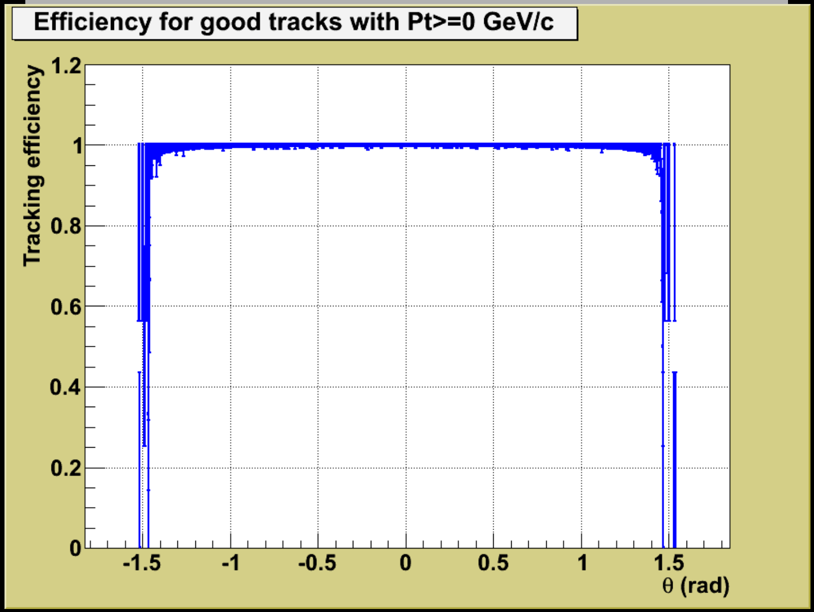
at least 4 hits in detector

Tracking Efficiency vs Theta

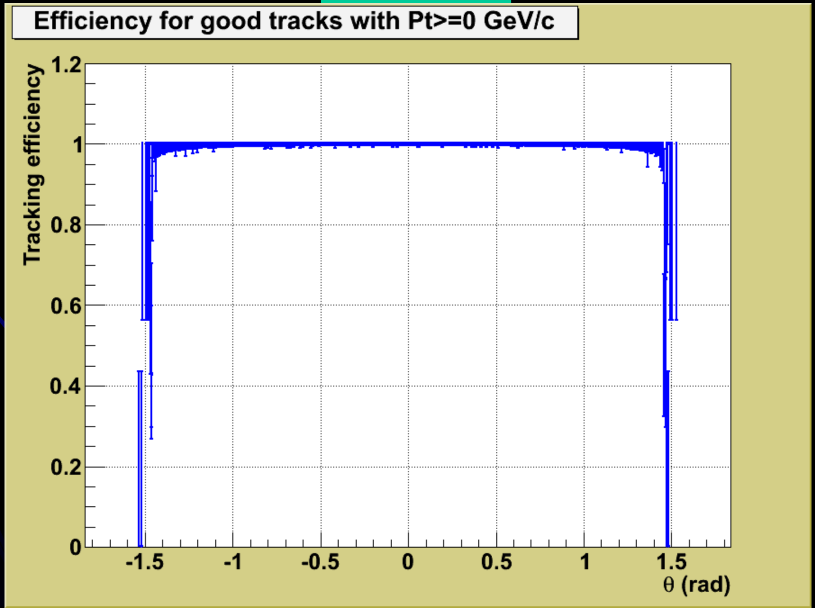
10° Nose with detector interference



10° Nose with no detector interference



6° Nose



$$\epsilon_{track} = \frac{reconstructed\ tracks}{good\ tracks}$$

Defining "good tracks"

DCA(true) < 3.5 cm

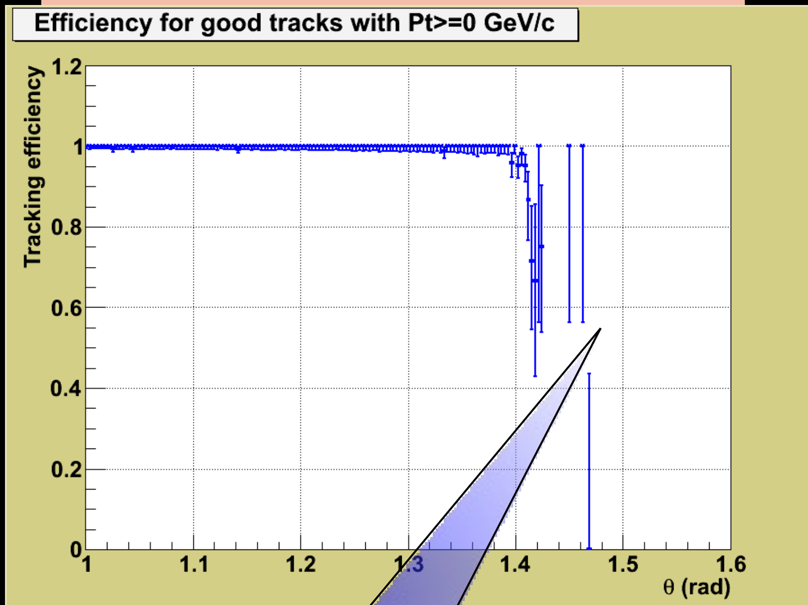
AND

at least 4 hits in detector

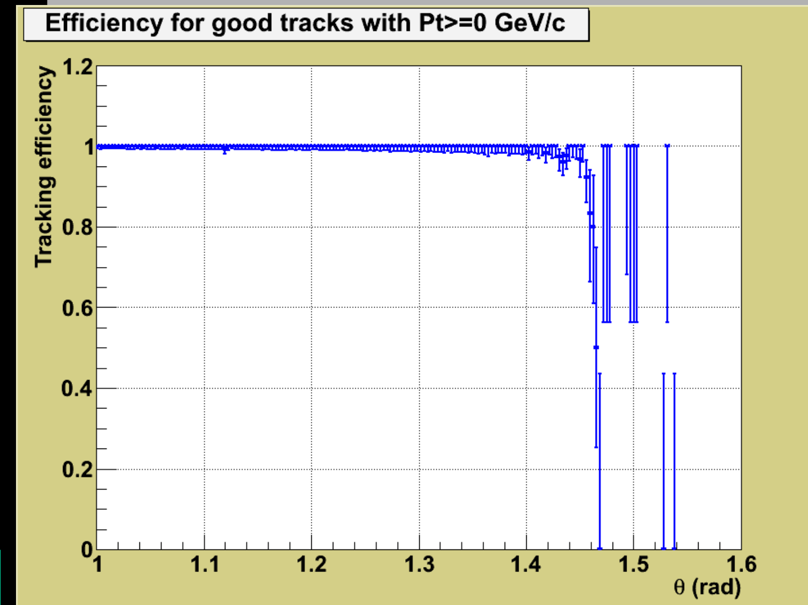
Single muons

Tracking Efficiency vs Theta (zoom)

10° Nose with detector interference

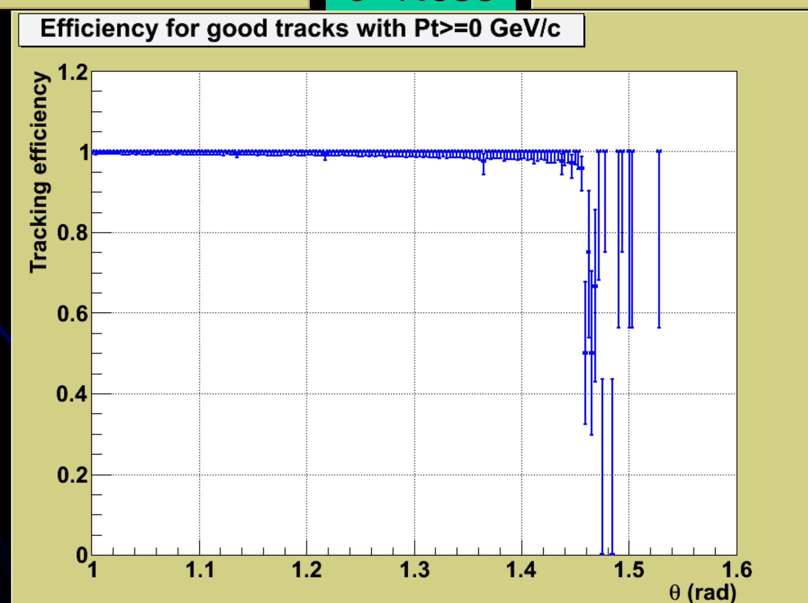


10° Nose with no detector interference



No reconstructed tracks at θ < 8.4°

6° Nose



$$\epsilon_{track} = \frac{reconstructed\ tracks}{good\ tracks}$$

Defining “good tracks”

DCA(true) < 3.5 cm

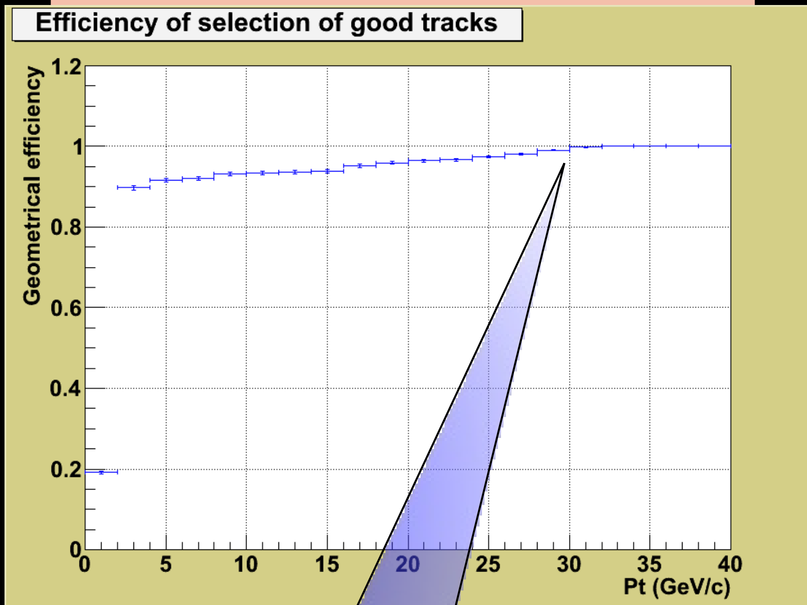
AND

at least 4 hits in detector

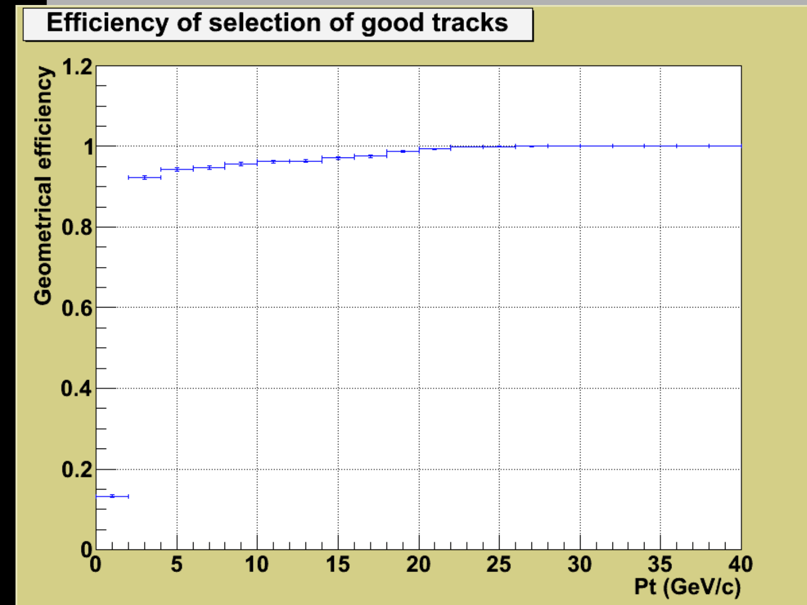
Single muons

Geometrical Efficiency vs Pt (zoom)

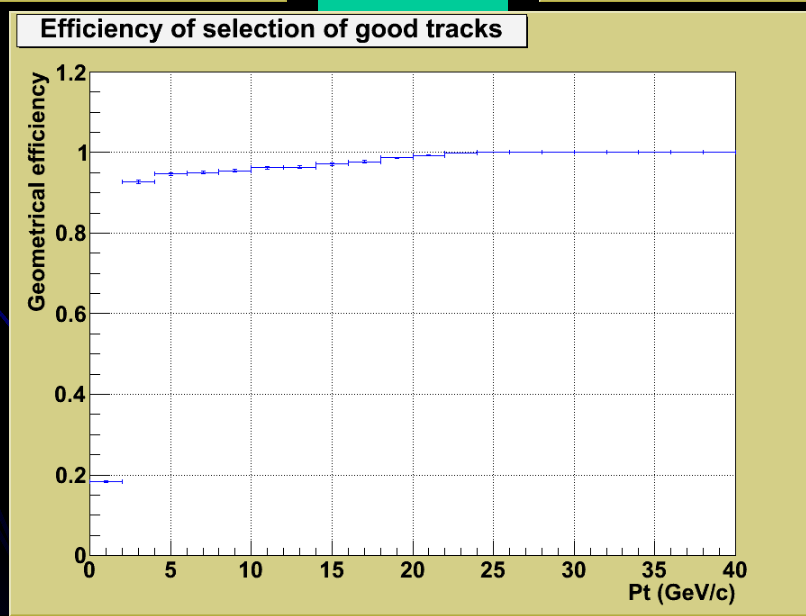
10° Nose with detector interference



10° Nose with no detector interference



6° Nose



Slightly lower geometrical efficiency

Single muons

$$\epsilon_{geom} = \frac{good\ tracks}{generated\ tracks}$$

Defining "good tracks"

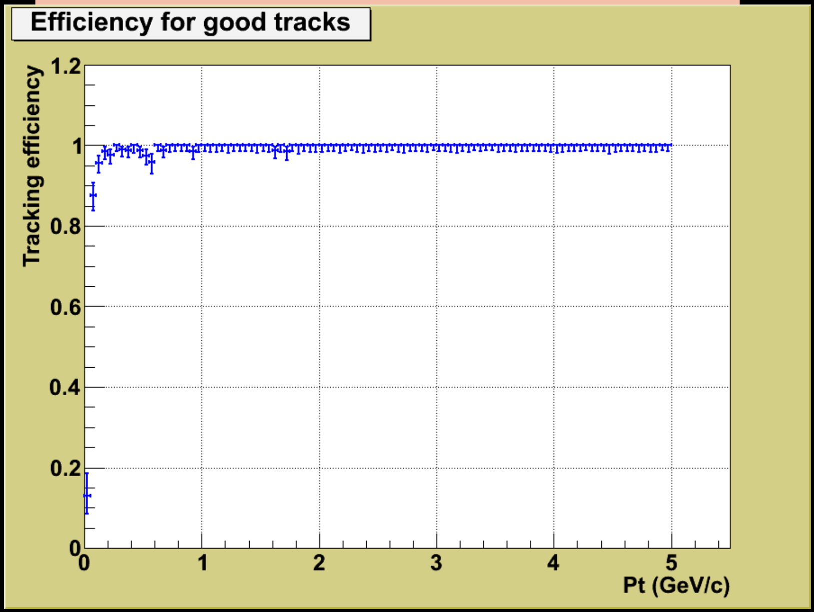
DCA(true) < 3.5 cm

AND

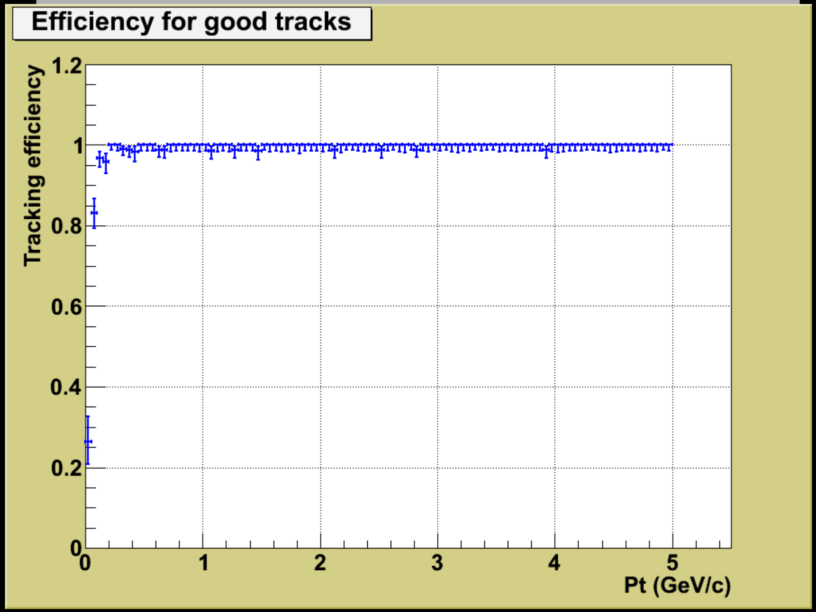
at least 4 hits in detector

Tracking Efficiency vs Pt

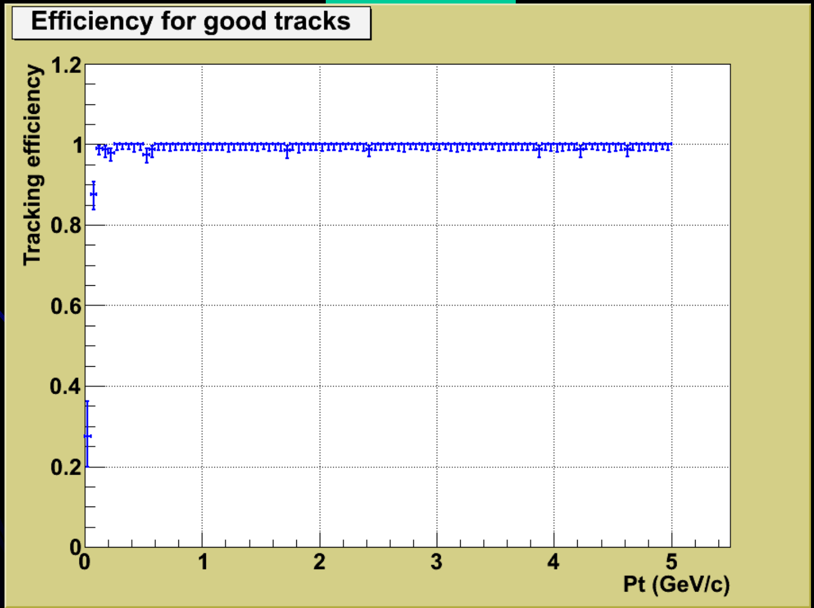
10° Nose with detector interference



10° Nose with no detector interference



6° Nose



$$\epsilon_{track} = \frac{reconstructed\ tracks}{good\ tracks}$$

Single muons

Defining "good tracks"

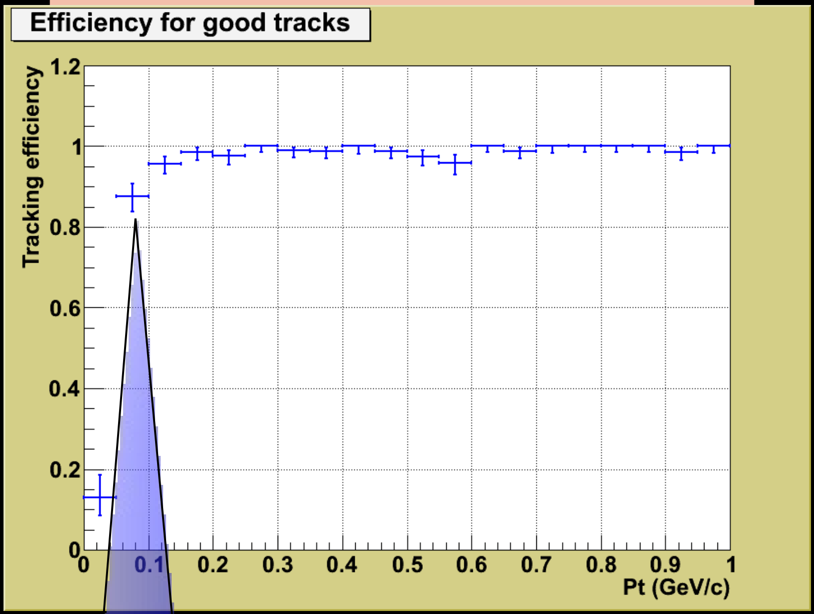
DCA(true) < 3.5 cm

AND

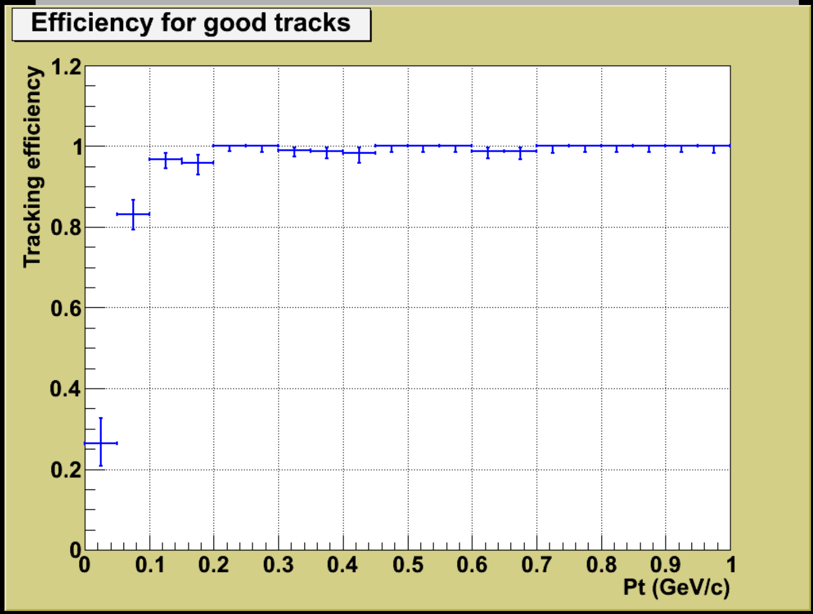
at least 4 hits in detector

Tracking Efficiency vs Pt (zoom)

10° Nose with detector interference

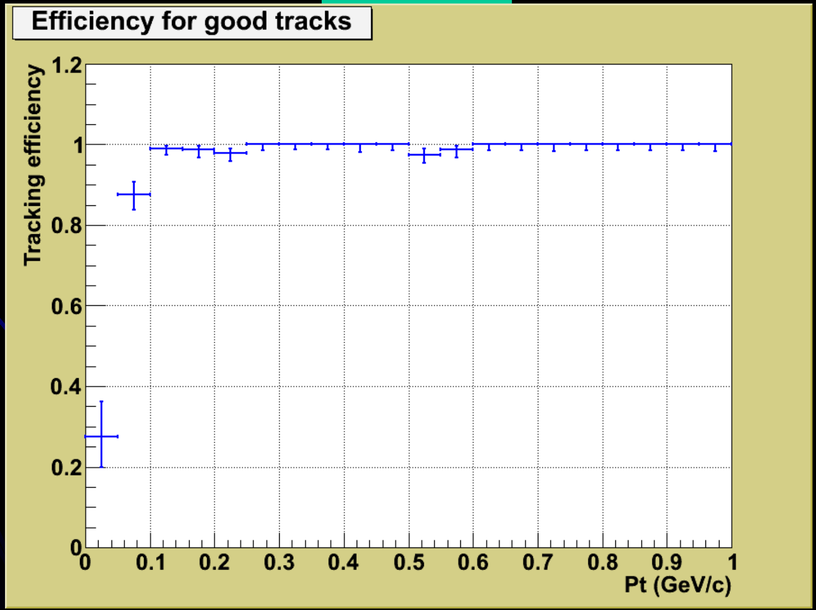


10° Nose with no detector interference



Minimal effect seen here

6° Nose



$$\epsilon_{track} = \frac{reconstructed\ tracks}{good\ tracks}$$

Defining "good tracks"

DCA(true) < 3.5 cm

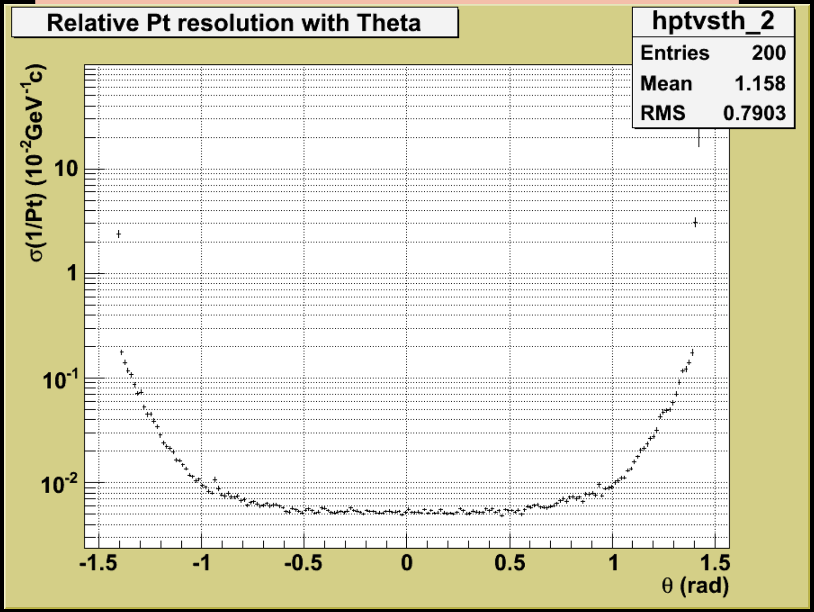
AND

at least 4 hits in detector

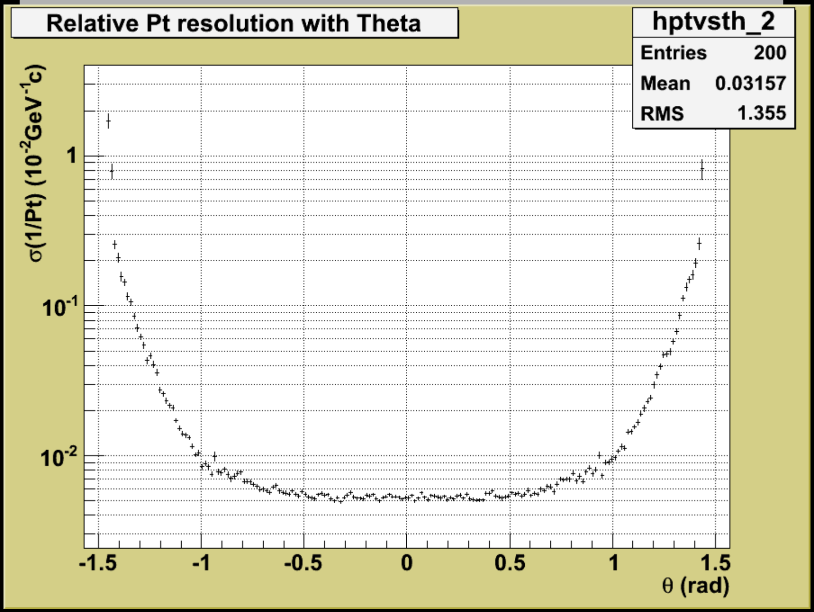
Single muons

1/Pt Resolution vs Theta

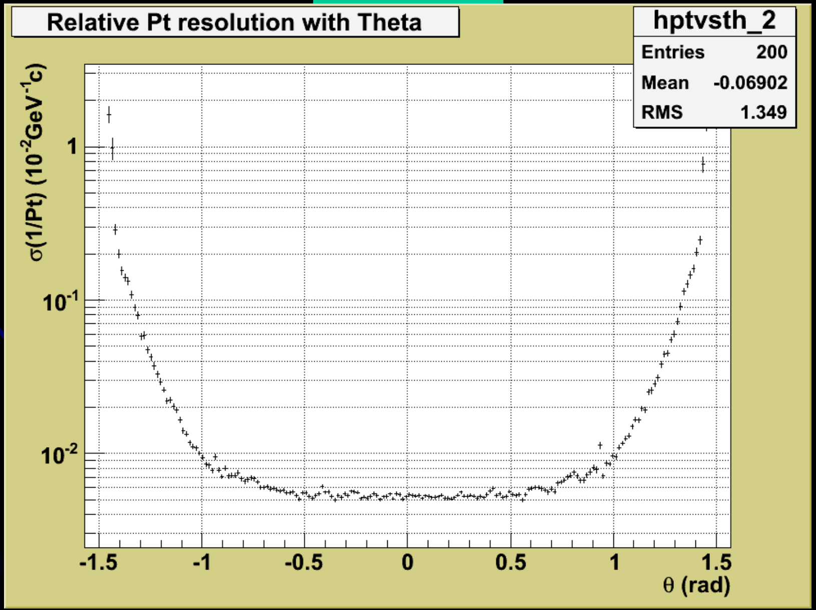
10° Nose with detector interference



10° Nose with no detector interference



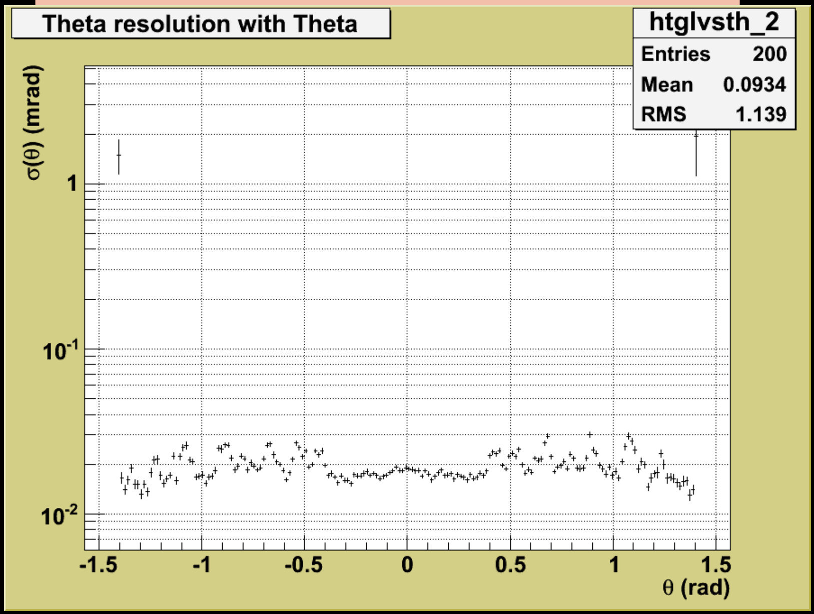
6° Nose



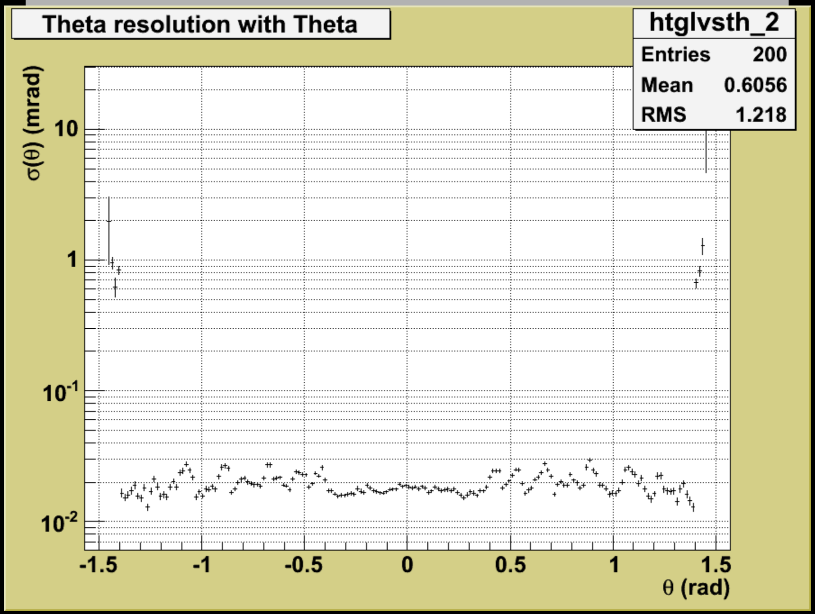
Single muons

Theta Resolution vs Theta

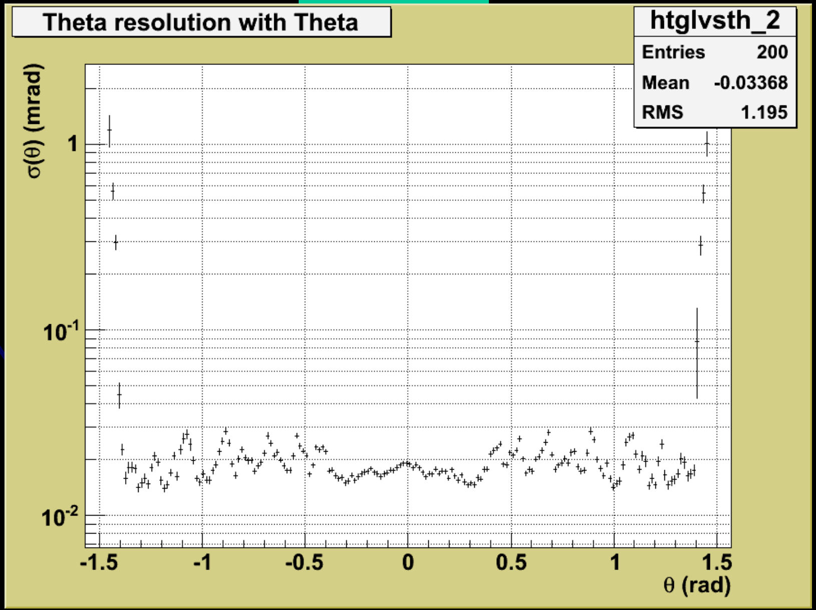
10° Nose with detector interference



10° Nose with no detector interference



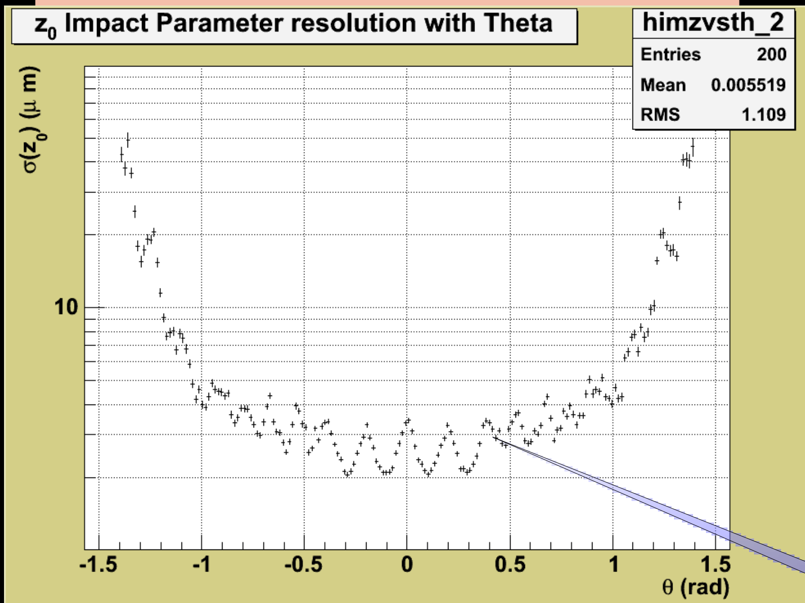
6° Nose



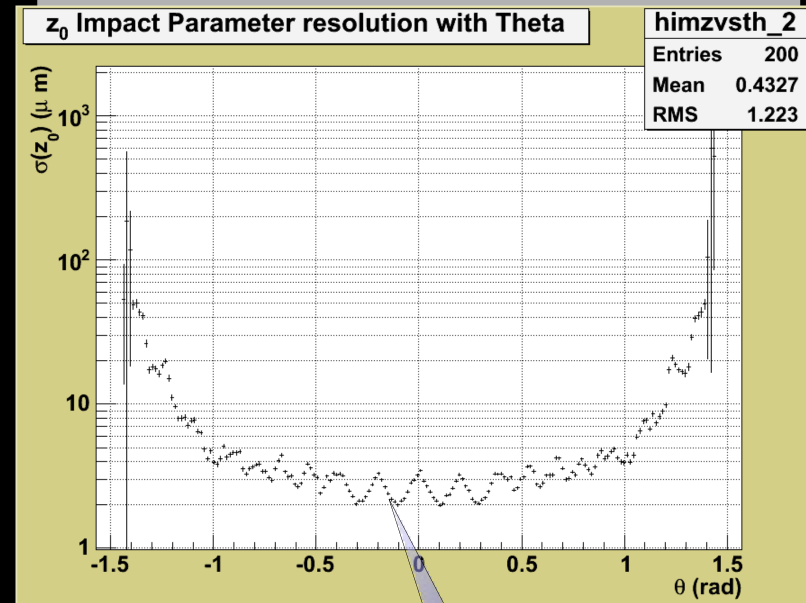
Single muons

Z_0 Resolution vs Theta

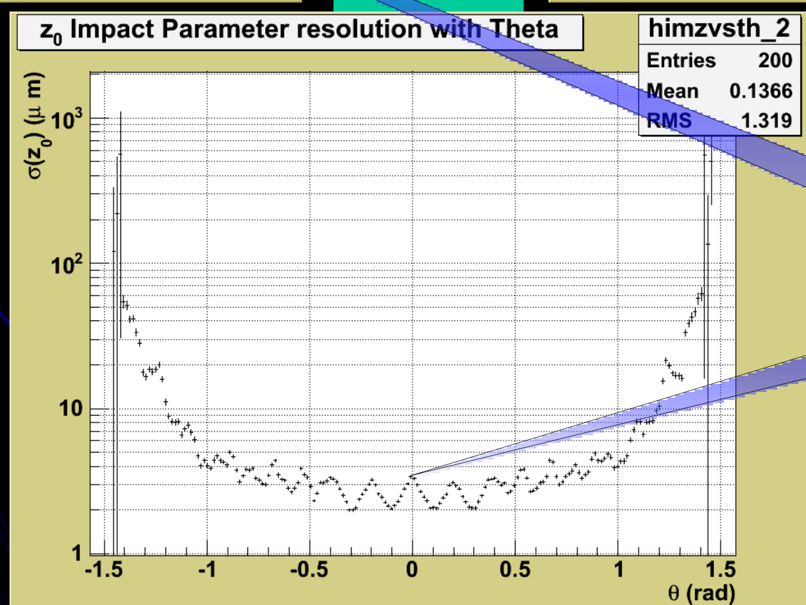
10° Nose with detector interference



10° Nose with no detector interference



6° Nose

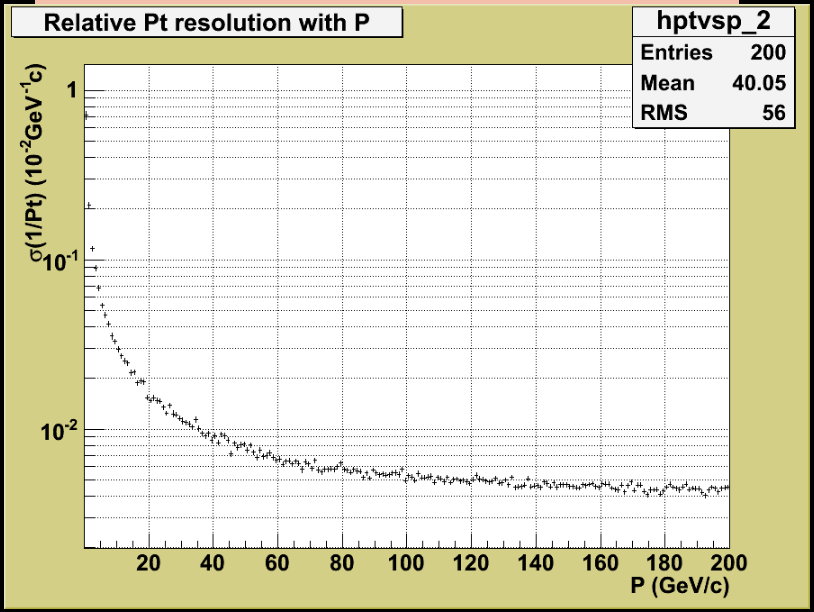


Single sensor in the barrel

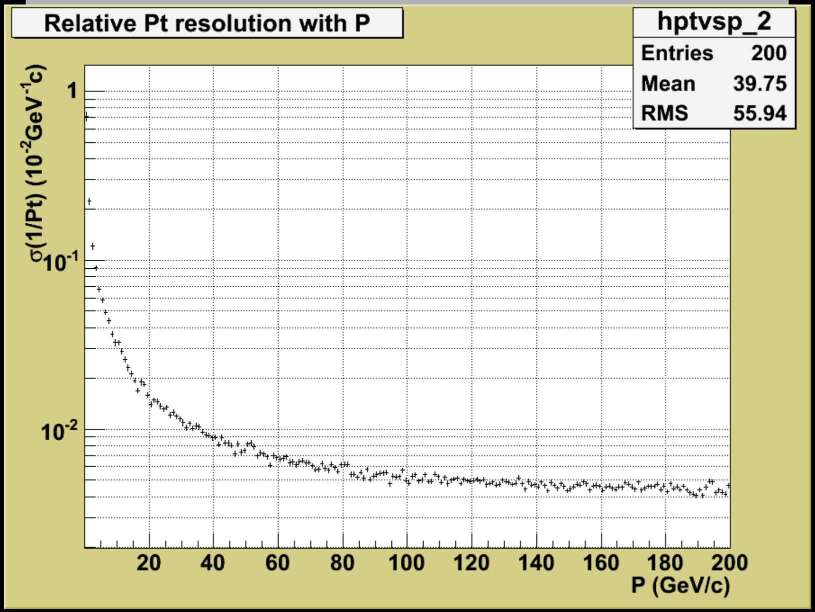
Single muons

1/Pt Resolution vs P

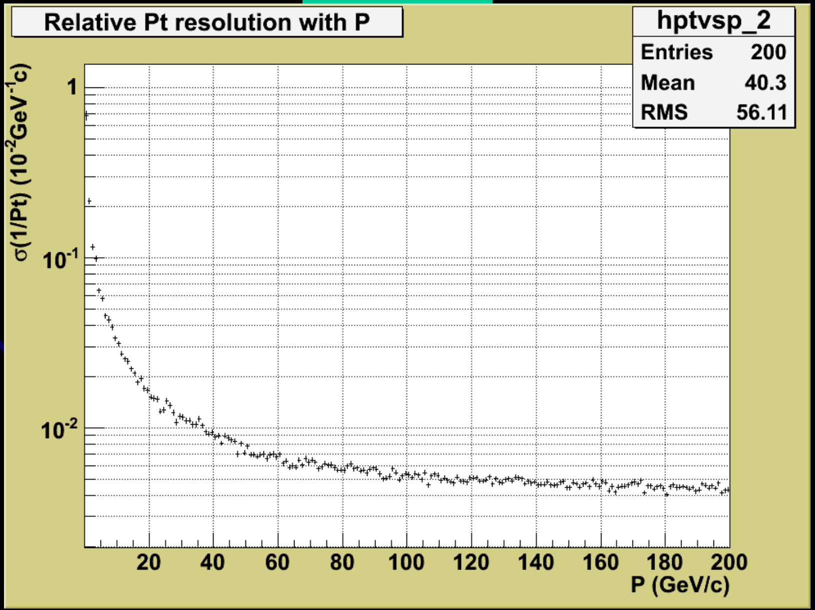
10° Nose with detector interference



10° Nose with no detector interference



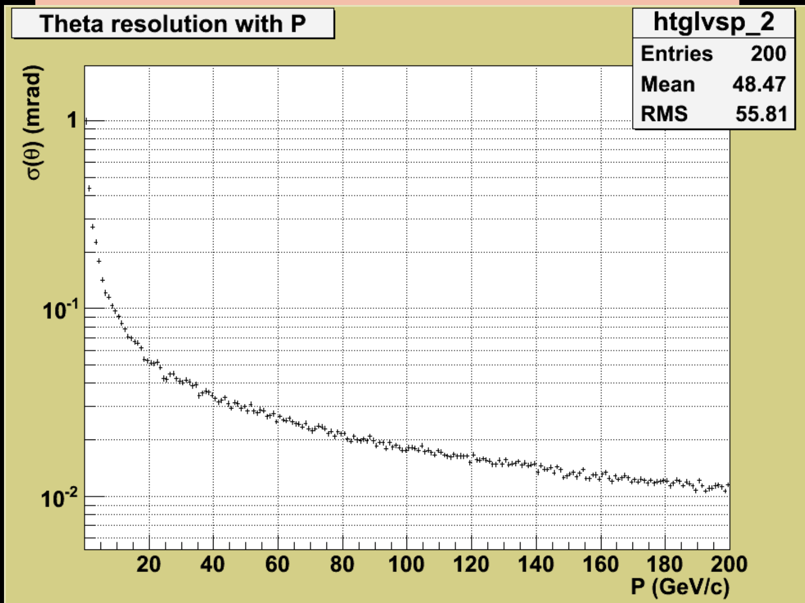
6° Nose



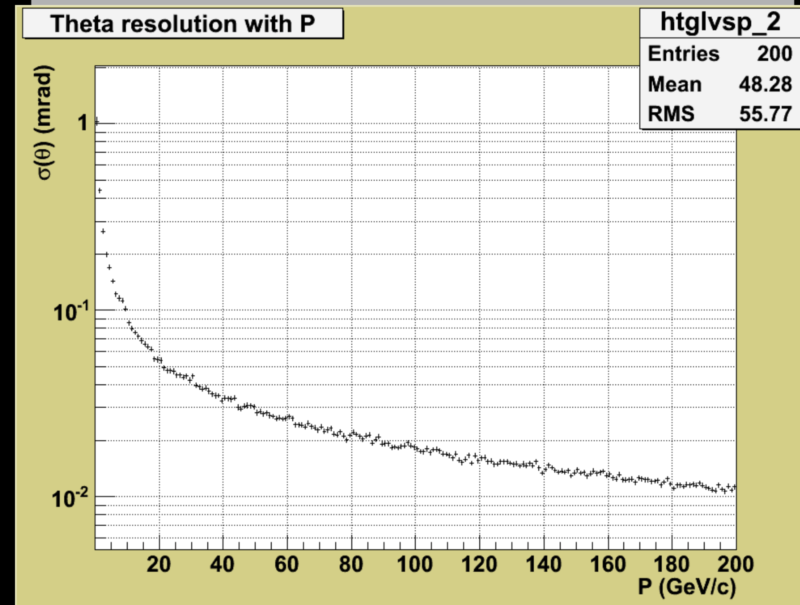
Single muons

Theta Resolution vs P

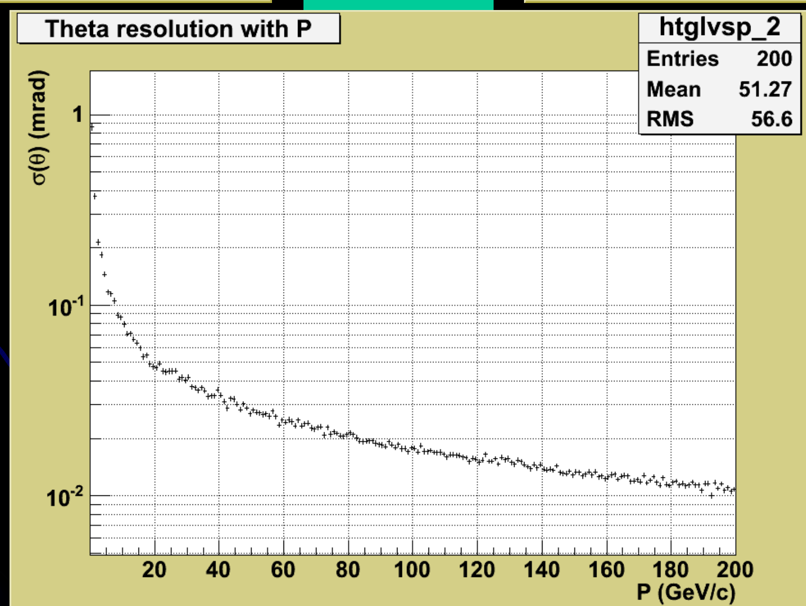
10° Nose with detector interference



10° Nose with no detector interference



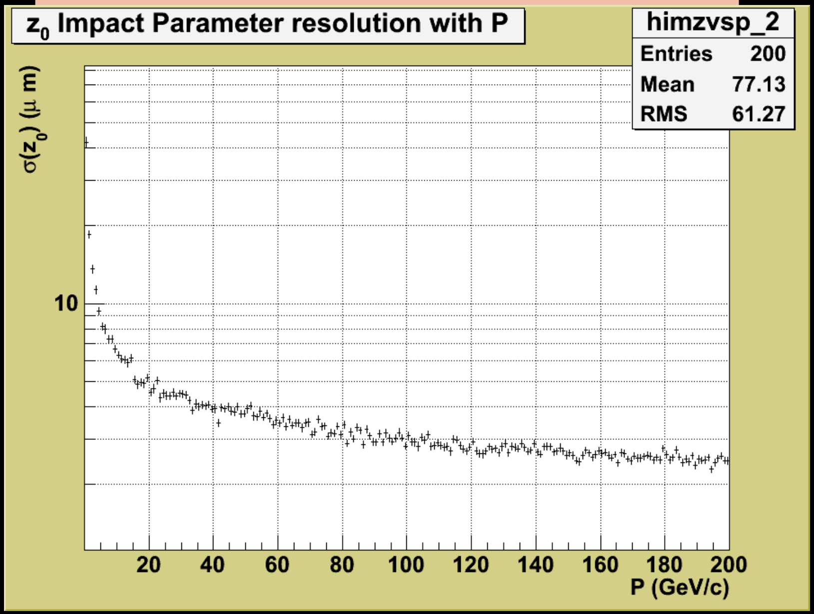
6° Nose



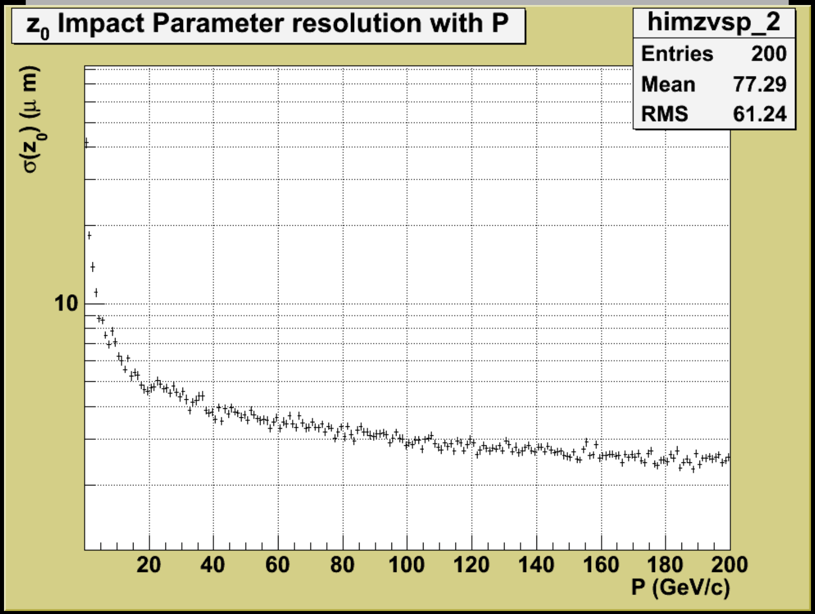
Single muons

Z_0 Resolution vs P

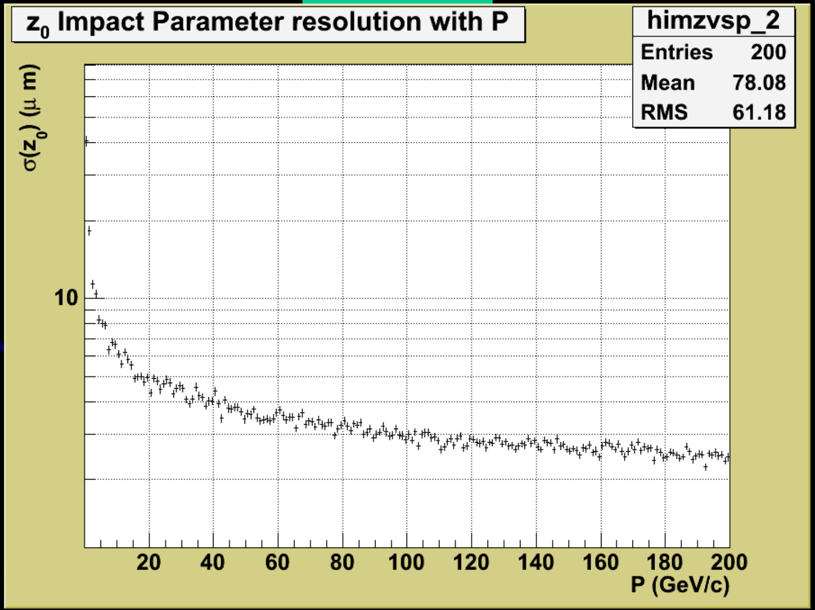
10° Nose with detector interference



10° Nose with no detector interference



6° Nose



Single muons

Conclusions

- A full simulation and reconstruction of Si-tracking detectors is implemented in ILCroot
- Pattern recognition and Kalman Filter have been continuously improved over the past 4 years for ILC and CLIC studies (F. Ignatov)
- Preliminary studies indicate that current nose (September 2010, 10°) has slightly lower geometrical efficiency than skinned nose
  - More material in the game
  - Geant4 interferance between detector and nozzle volumes in the game (see N. Terentiev talk on Nov. 10th)
- Implementation of new Si-detector compatible with 10° nozzle will start immediatly
- Assistence by F. Ignatov is of paramount importance to fine-tune track reconstrucion
- Repeat current studies with background (next talk)

Backup slides

SDigitization in Strips Detector

- Get the Segmentation Model for each detector (from IlcVXDSegmentationSSD class)
- Get Calibration parameters (from IlcVXDCalibrationSSD class)
- Load background hits from file (if any)
- Loop on the hits and create a segment in Si in 3D
  - Step along the line in equal size increments
    - Compute Drift time to p-side and n-side:

```
tdrift[0] = (y+(seg->Dy()*1.0E-4)/2)/GetDriftVelocity(0);
tdrift[1] = ((seg->Dy()*1.0E-4)/2-y)/GetDriftVelocity(1);
```

    - Compute diffusion constant:

```
sigma[k] = TMath::Sqrt(2*GetDiffConst(k)*tdrift[k]);
```

    - integrate the diffusion gaussian from -3σ to 3σ
  - Charge pile-up is automatically taken into account

SDigitization in Strips (2)

- Add electronic noise per each side separately

```
// noise is gaussian
   noise  = (Double_t) gRandom->Gaus(0,res->GetNoiseP().At(ix));

   // need to calibrate noise
                                        noise *= (Double_t) res->GetGainP(ix);

   // noise comes in ADC channels from the calibration database
   // It needs to be converted back to electronVolts
   noise /= res->GetDEvToADC(1.);
```

- Add coupling effect between nearby strips
  - different contribution from left and right neighbours
  - Proportional to nearby signals

- Remove dead pixels (use signal map)

- Convert total charge into signal (ADC count)

```
     if(k==0) signal /= res->GetGainP(ix);
     else signal /= res->GetGainN(ix);
// signal is converted in unit of ADC
     signal = res->GetDEvToADC(fMapA2->GetSignal(k,ix));
```

Clusterization in Strip Detector

- Create a initial cluster from adjacent strips (no for diagonal)
- Separate into Overlapped Clusters
  - Look for through in the analog signal shape
  - Split signal of parent clusters among daugheter clusters
- Intersect stereo strips to get Recpoints from CoG of signals (and error matrix)
- Kalman filter picks up the best Clusters

The Parameters for the Strips

- Strip size (p, n)
- Stereo angle (p-> 7.5 mrad, n->25.5 mrad)
- Ionization Energy in Si = 3.62E-09
- Hole diffusion constant (= 11 cm^2/sec)
- Electron diffusion constant (= 30 cm^2/sec)
- v^P_{drift}(=0.86E+06 cm/sec) , v^N_{drift}(=2.28E+06 cm/sec)
- Calibration constants
  - Gain
  - ADC conversion (1 ADC unit = 2.16 KeV)
- Coupling probabilities between strips (p and n)
- σ of gaussian noise (p AND n)
- threshold