

Clupatra

Topological TPC pattern recognition

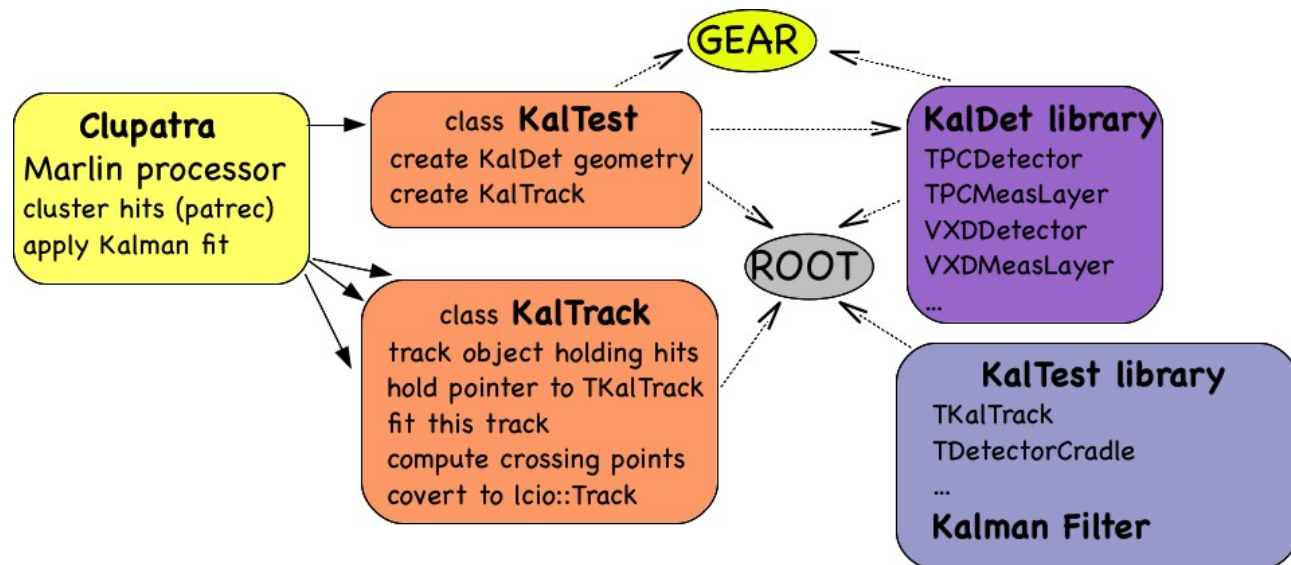
Frank Gaede, DESY

ILDWS Orsay, SW pre-meeting

May 22, 2011

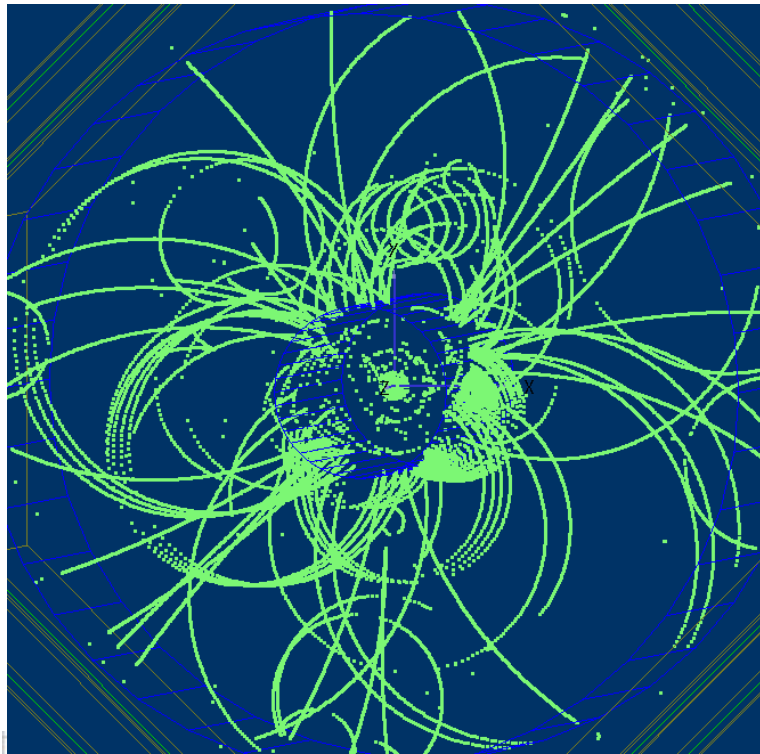
Introduction

- identified the need to replace old f77 tracking code in order to improve the sw-maintenance and the performance (background studies, 1 TeV, CLIC,...)
- need new TPC pattern recognition
- need new Kalman filter tool
 - chose KalTest (K.Fujii et al)
 - also used by LCTPC/MarlinTPC

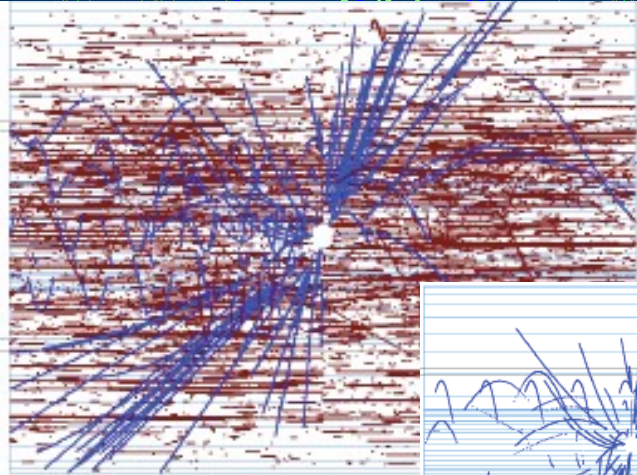


- developed interface to KalTest (decouple from concrete fitter)
- under evolution
-> see Steve's talk

TPC Pattern recognition



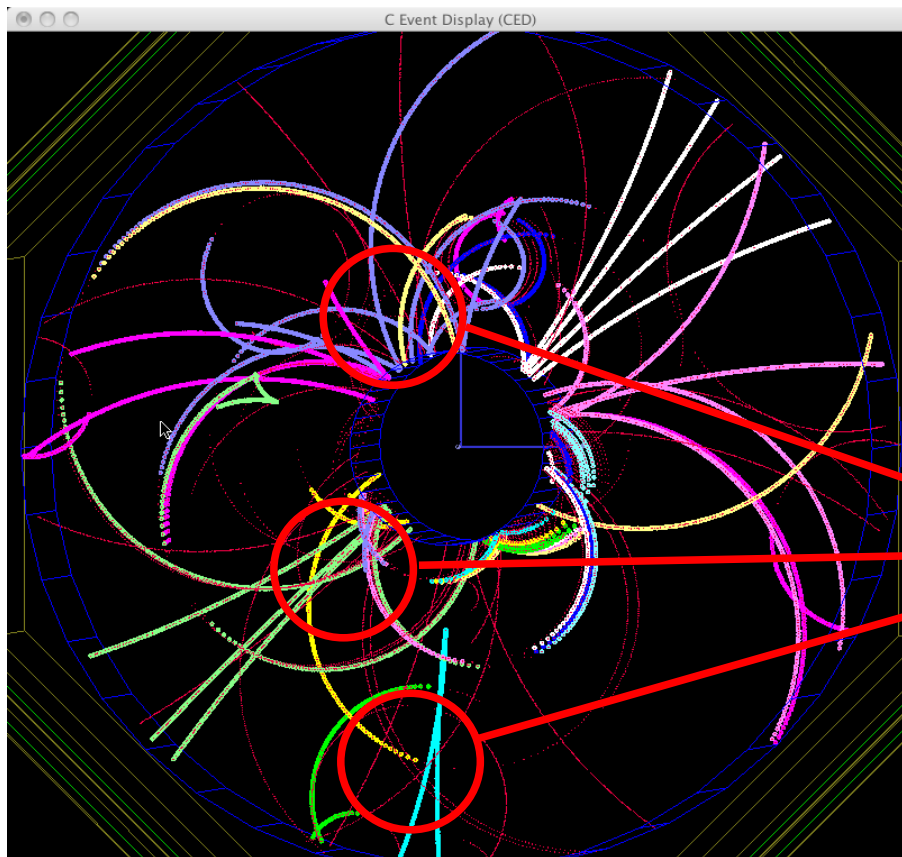
- patrec in a TPC should be rather easy
 - tracks immediately visible
 - “could be done by a kid with crayons”
- ILD TPC has a huge number of voxels >200 hits on many tracks
- classic triplet search and combinatorial Kalman filter probably overkill (CPU & coding intensive)
- mean distance between hits on track is mostly much smaller than distance between tracks
- => can use **NN-Clustering**
- **micro curlers from pair bg should be removed beforehand**



Clupatra step 1

- nearest neighbor clustering
- use simple euclidian distance
 - $d = \sqrt{dx^2 + dy^2 + dz^2} < C ; C = 40\text{mm}$
- use z-index + sliding window to speed up processing

```
inline bool mergeHits( GenericHit<HitClass>* h0,  
                      GenericHit<HitClass>* h1) {  
  
    const PosType* pos0 = h0->first->getPosition() ;  
    const PosType* pos1 = h1->first->getPosition() ;  
  
    return  
        ( pos0[0] - pos1[0] ) * ( pos0[0] - pos1[0] ) +  
        ( pos0[1] - pos1[1] ) * ( pos0[1] - pos1[1] ) +  
        ( pos0[2] - pos1[2] ) * ( pos0[2] - pos1[2] )  
        < _dCutSquared ;  
}
```



example:

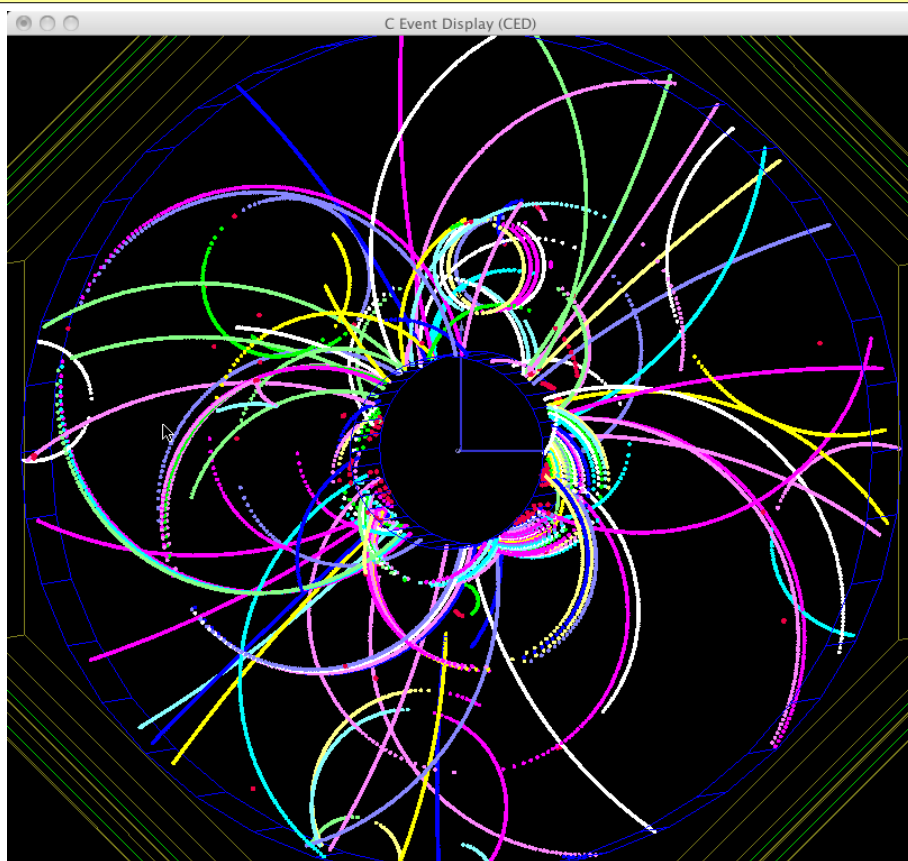
- ttbar event @ 500 GeV

obvious issue:

- close by tracks are merged into one cluster

Clupatra step 2

- for merged clusters (duplicate pad row fraction):
 - **cluster in pad row ranges** (e.g. 15 rows) – **going inwards**
 - identify **clean track stubs**
 - **extend clean stubs forward & backward using Kaltest fitter**
 - add Hit if $\Delta(\chi^2) < 35$.
 - update track state !

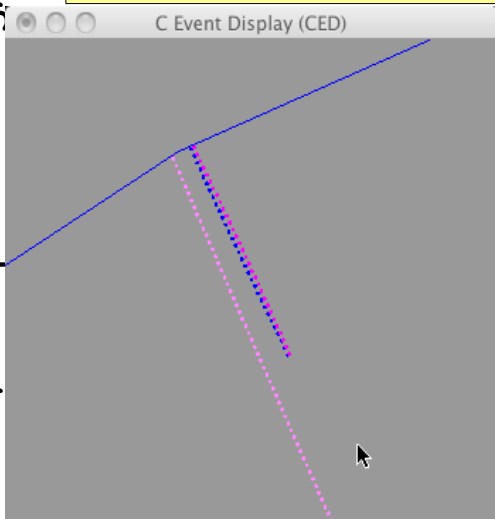


example:

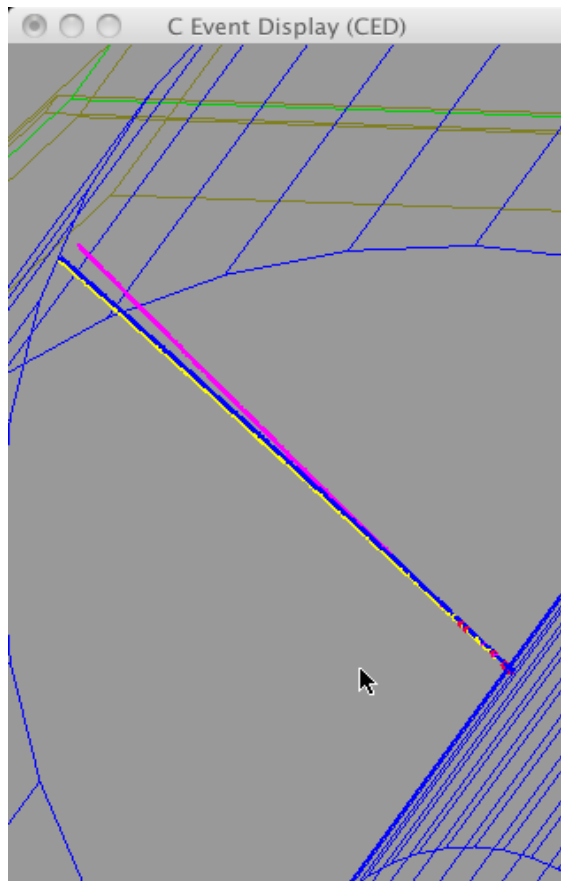
- ttbar event @ 500 GeV
- results in clean tracks and segments for curlers
- little leftover hits (red)
- some very close by tracks lost (taus/conversions)

Clupatra step 3

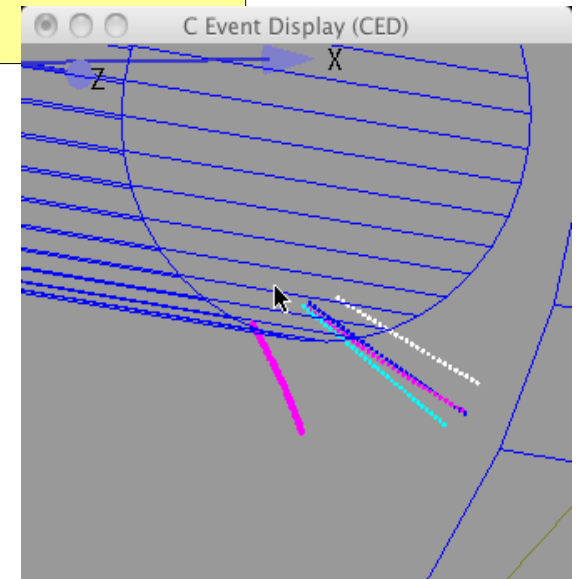
- re-cluster in leftover hits (NN clustering)
- compute pad row multiplicity
- force into one, two or three clusters
- apply KalTest fit to throw out falsely merged hits (rare)
 - higher multiplicity to be done (very rare)



- gamma conversion in barrel
- forced into two tracks



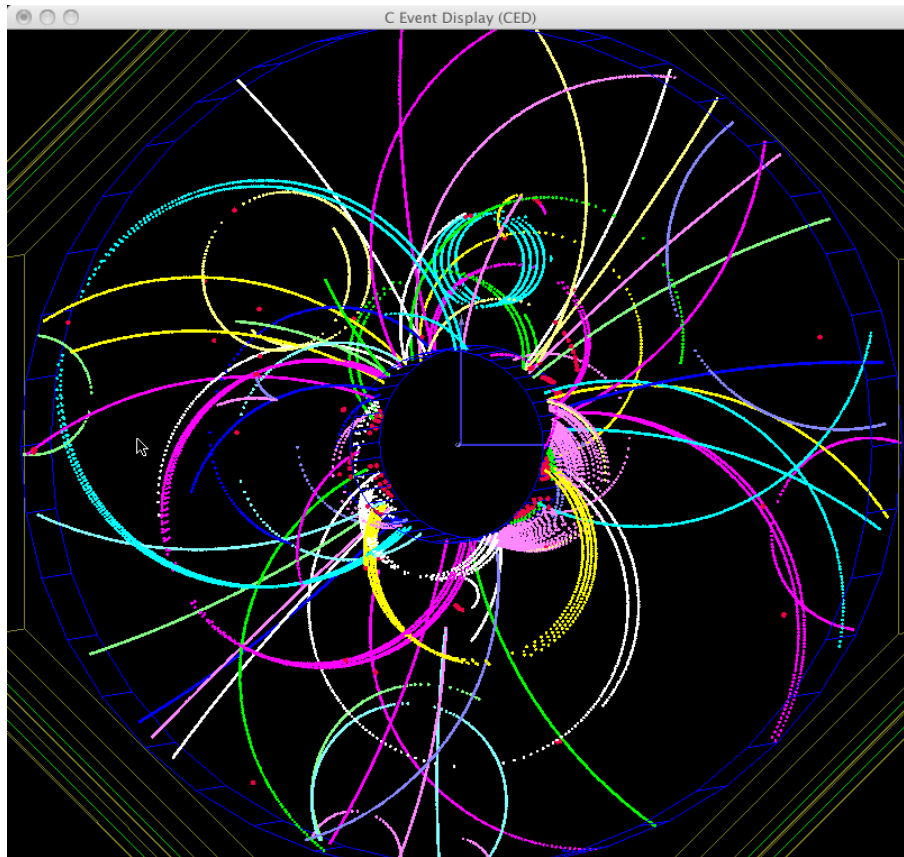
- three prong tau - barrel
- two close-by tracks forced into two tracks



- five prong tau - forward
- three close-by tracks forced into three tracks

Clupatra step 4

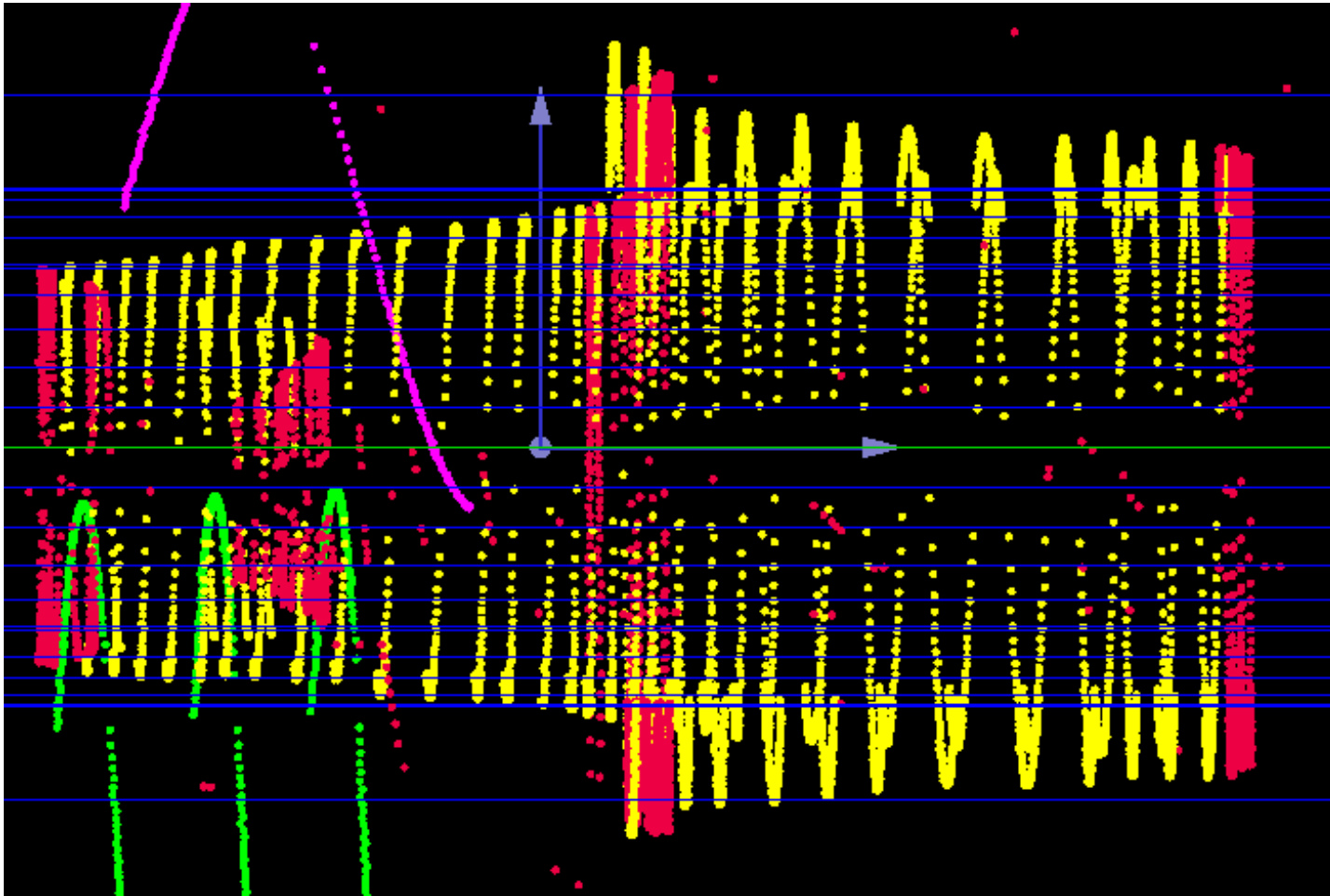
- merge track segments (from curlers)
- based on rough ($O(10\%)$) criterion for R , $\Delta(x_c, y_c)$, $\tan(\lambda)$
- disallow z-overlaps



example:

- $t\bar{t}$ event @ 500 GeV
- works nicely
- few segments are not merged
- most of these curler segments were lost in on old pat rec

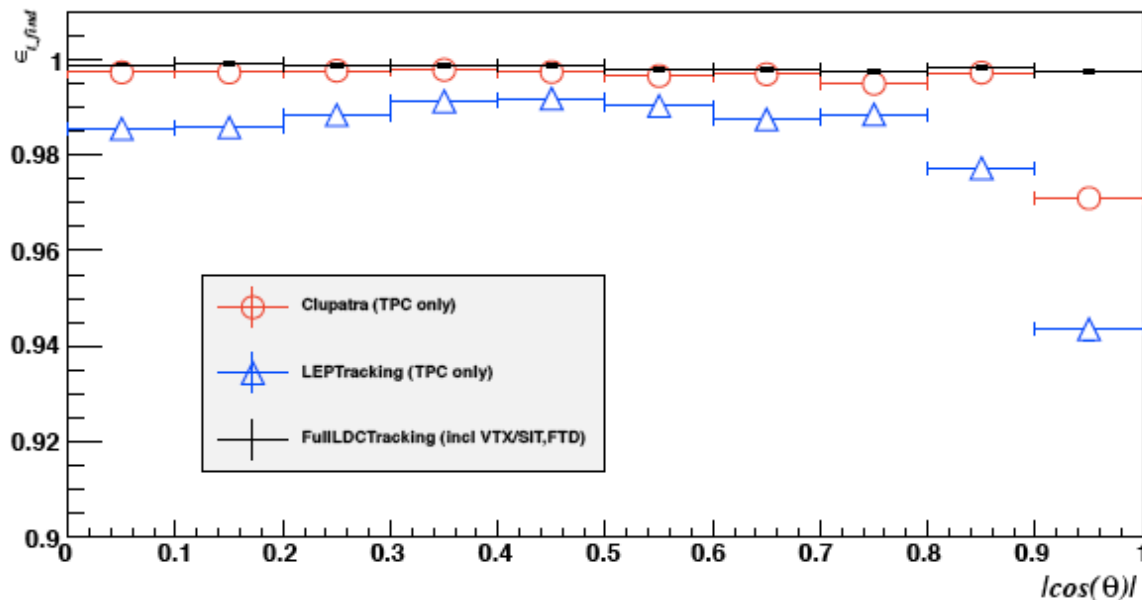
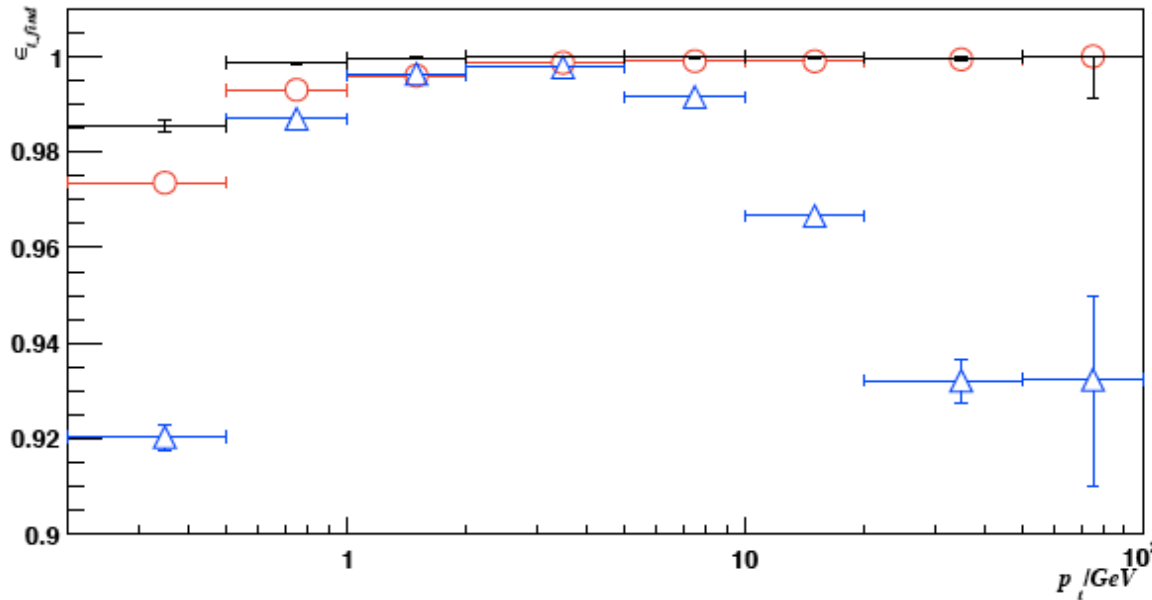
'The crazy muon'



- Clupatra finds interleaved curlers to large extend
 - yellow: clupatra track - red: hits have been missed
- this muon curls back into itself five times !
- don't need to deal with this often :-)

track finding efficiency I

TPC track finding efficiency - $t\bar{t}$ @ 500 GeV



- prompt tracks $\text{PCA(IP)} < 10\text{cm}$
- > 5 TPC Hits
 - ($p_t > 100\text{ MeV}$)
 - ($|\cos(\theta)| > .99$)

• comparison to LEPTracking pattern recognition

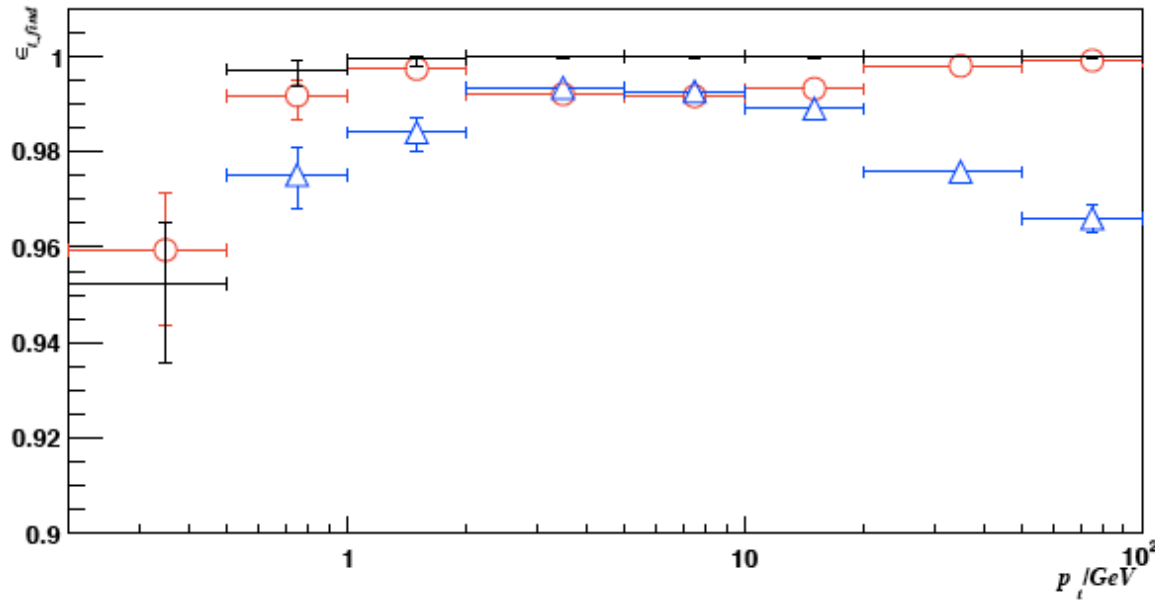
• NB: Clupatra has no reconstructed tracks yet

• and no quality cuts are applied

• high efficiency demonstrates that algorithm works and could replace old f77 code sometime soon

track finding efficiency II

TPC track finding efficiency - tau pairs @ 500 GeV



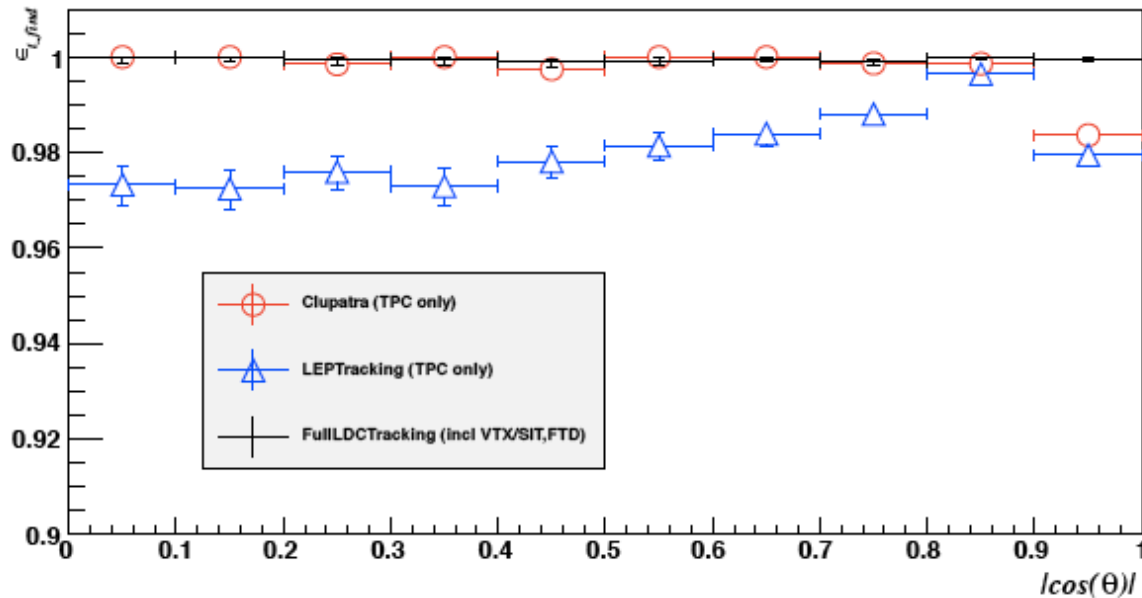
prompt tracks PCA(IP)<10cm
> 5 TPC Hits

- ($p_t > 100$ MeV)
- ($|\cos(\theta)| > .99$)

comparison to LEPTracking
pattern recognition

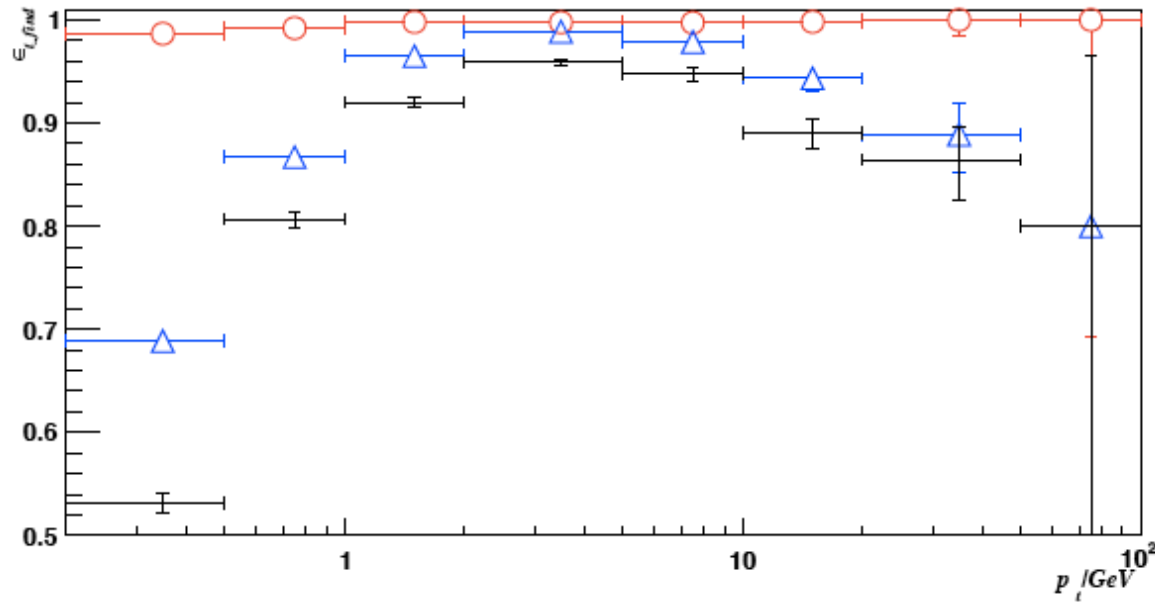
NB: Clupatra has no
reconstructed tracks yet
and no quality cuts are
applied

high efficiency demonstrates
that algorithm works and
could replace old f77 code
sometime soon



track finding efficiency III

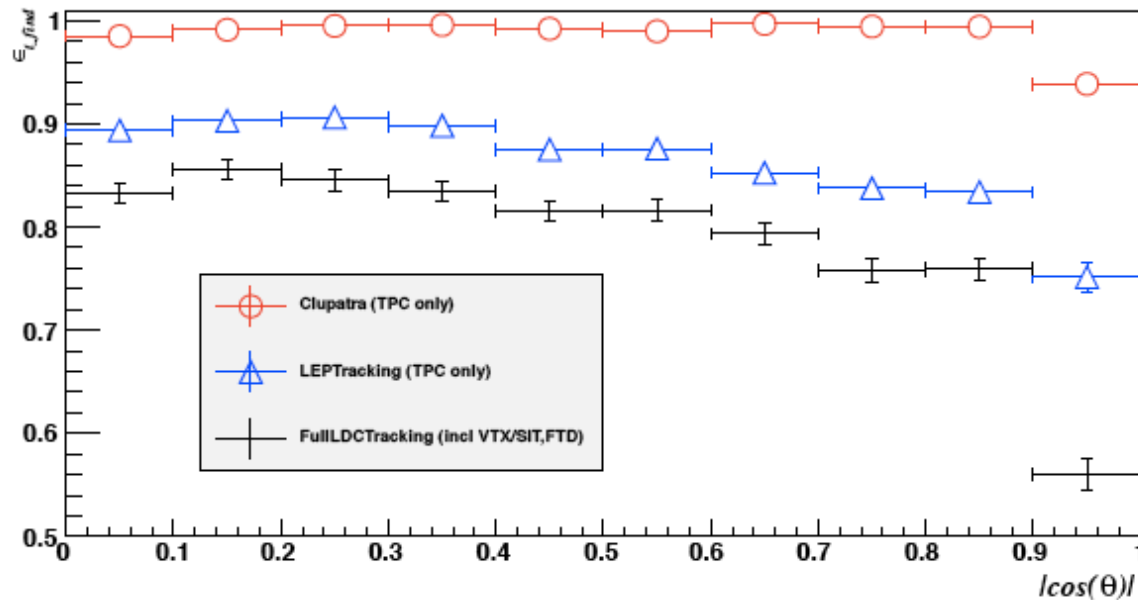
TPC track finding efficiency - tracks from v-zeros (In ttbar @ 500 GeV)



- non-prompt tracks
- $\text{rho_vtx} > 10\text{cm}$
 - parent charge==0
 - mostly vzeros and conversions
- > 5 TPC Hits
- ($p_t > 100 \text{ MeV}$)
 - ($|\cos(\theta)| > .99$)

comparison to LEPTracking
 pattern recognition
 NB: Clupatra has no reconstructed tracks yet and no quality cuts are applied

need to study if we can gain for v0/conversion reconstruction



Summary & Outlook

- a new topological TPC pat-rec has been developed combining clustering and combinatorical Kalman filter methods
- first results show improvements compared to existing LEPTracking pat-rec code
- To Do
 - code cleanup and documentation
 - proper track fitting (propagation through inner material)
 - track collections with quality cuts
 - study effects of background
 - pair-bg and gamma-gamma
 - combine with Si-Tracking
 - pick up hits for TPC segments
 - merge track segments from independent Si-Tracking