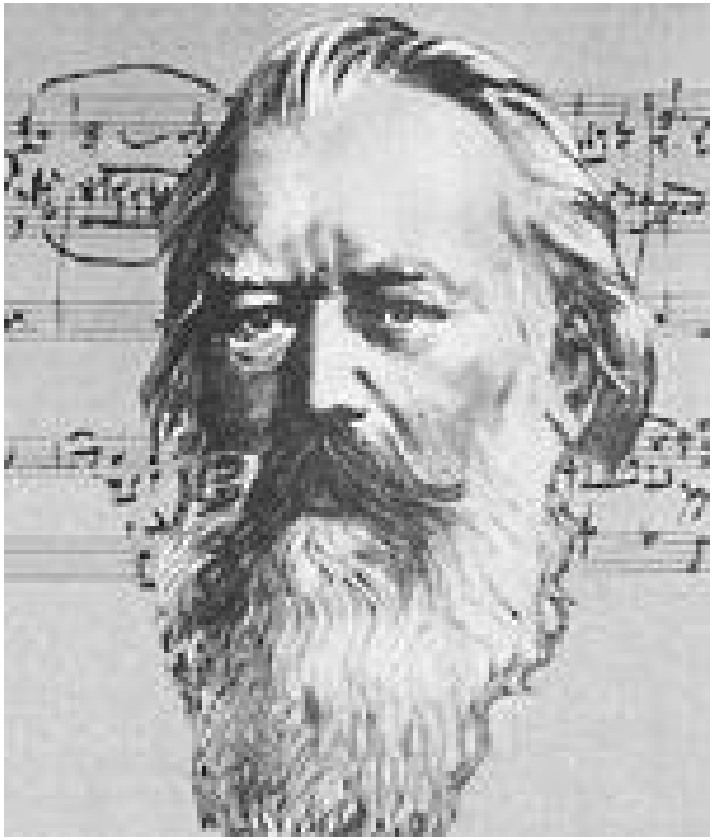
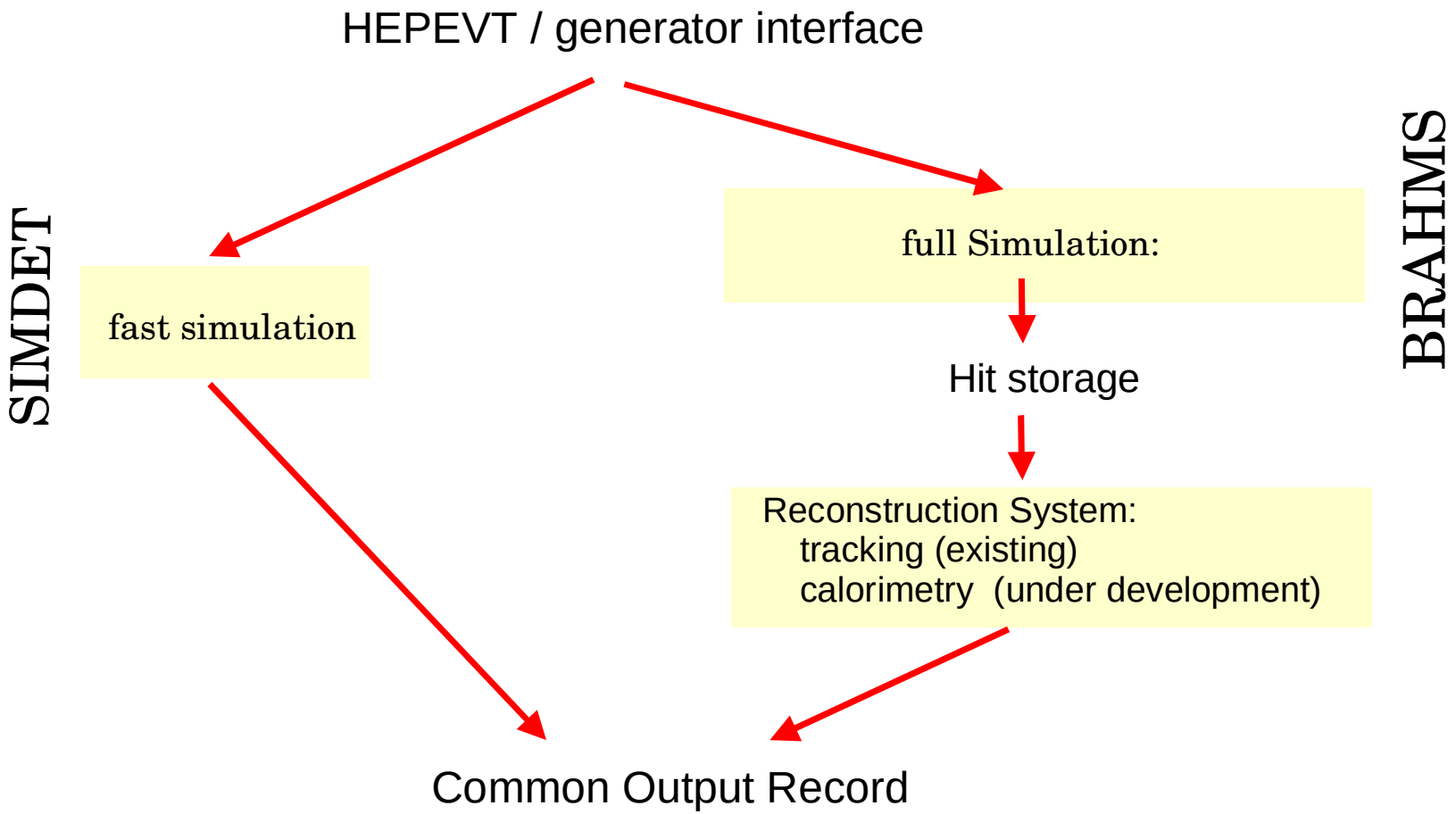
Simulation in Europe

Ties Behnke, SLAC/ DESY

- BRAHMS: our trusted, old full simulation package
  - FORTRAN
  - GEANT321
  - grown, not designed
- SIMDET: the fast simulator
  - FORTRAN
  - fast
  - fairly complete
  - one detector: TESLA TDR
- SGV: fast Simdet alternative
- The new world: see Henri's talk:
  - MOKKA
  - GEANT4
  - everything is better, designed, ...



The simulation framework



We are working on defining a common hit storage format between BRAHMS and MOKKA, and between the European and the US frameworks

BRAHMS

The BRAHMS suite contains two programs:

- GEANT3 based simulation code (“BRAHMS proper”)
- The reconstruction program RERECO

technical detail:
simulation and reco may be run together or separately

BRAHMS: The **simulation** program:

- complete implementation of the TDR tracker
- full implementation of the TDR calorimeter
- full implementation of the forward system
- full implementation of the muon system

communication between simulation and reconstruction via simple serial gzipped, files

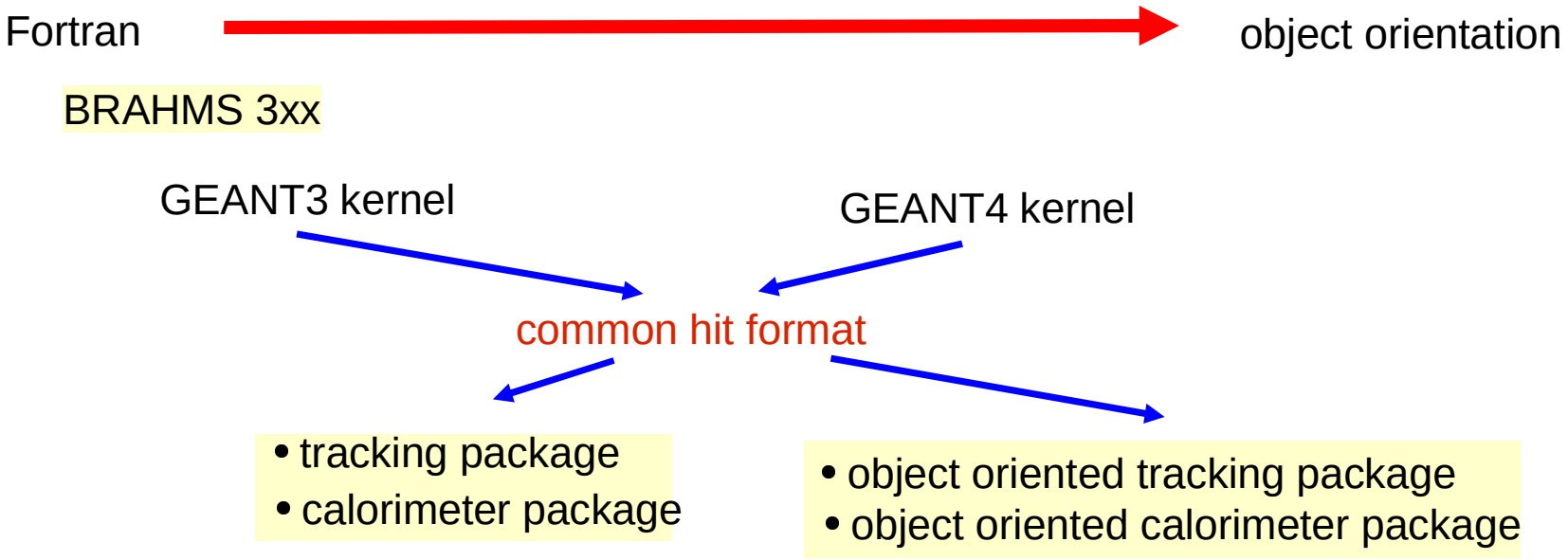
RERECO: The **reconstruction** program

- full track reconstruction and detector merging code
- calorimeter reconstruction code
- full energy flow algorithms (a version of..)

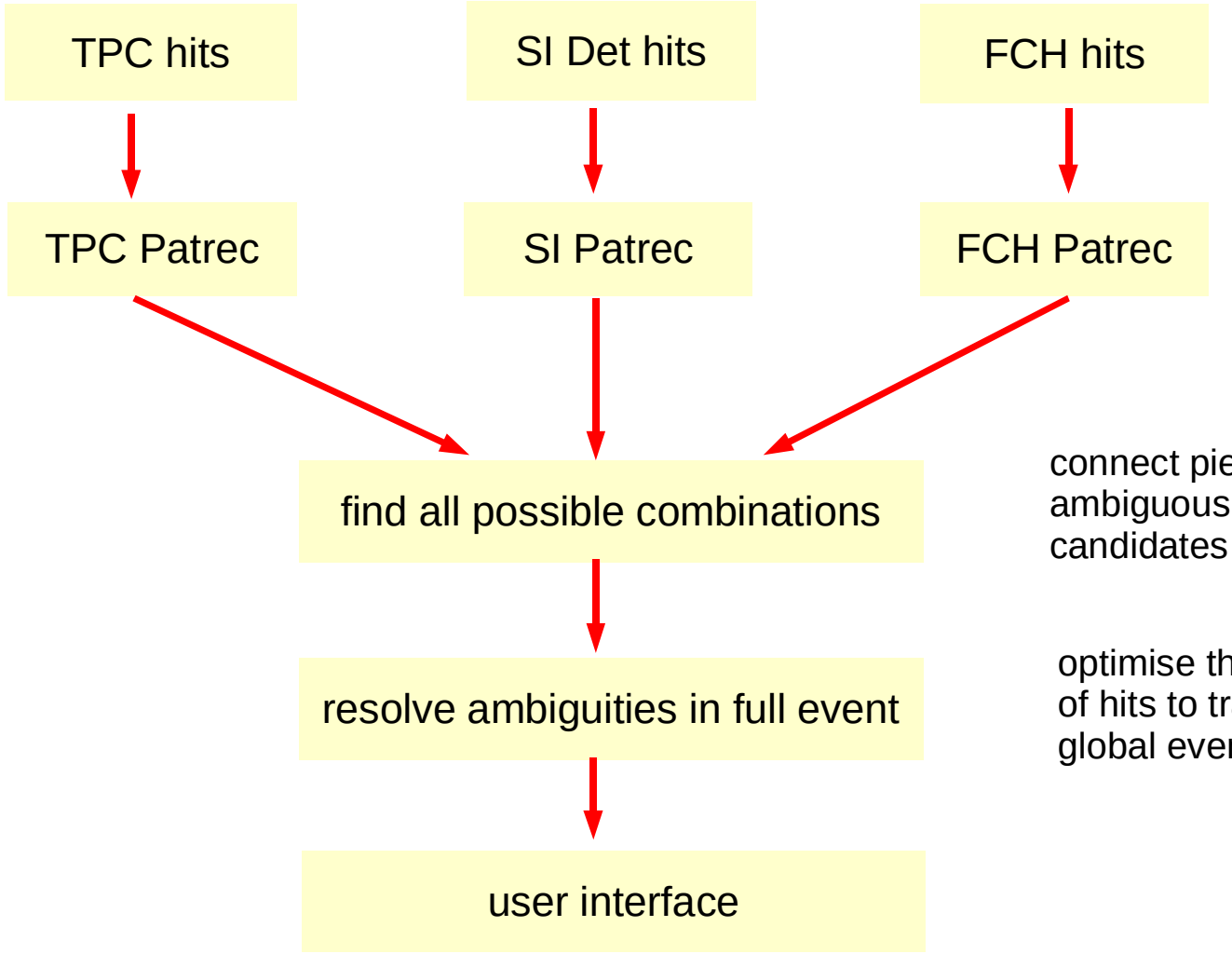
most code is still FORTRAN based

The Goal

- The goal: develop and maintain a modern simulation environment which is
  - flexible
  - maintainable for a long time to come
  - scalable
- At the same time:
  - continue the support for the existing system for still some time to come
- Ideally: maintain a link between the programs to avoid duplication and translation errors



The Tracking Package



ALEPH. OPAL software

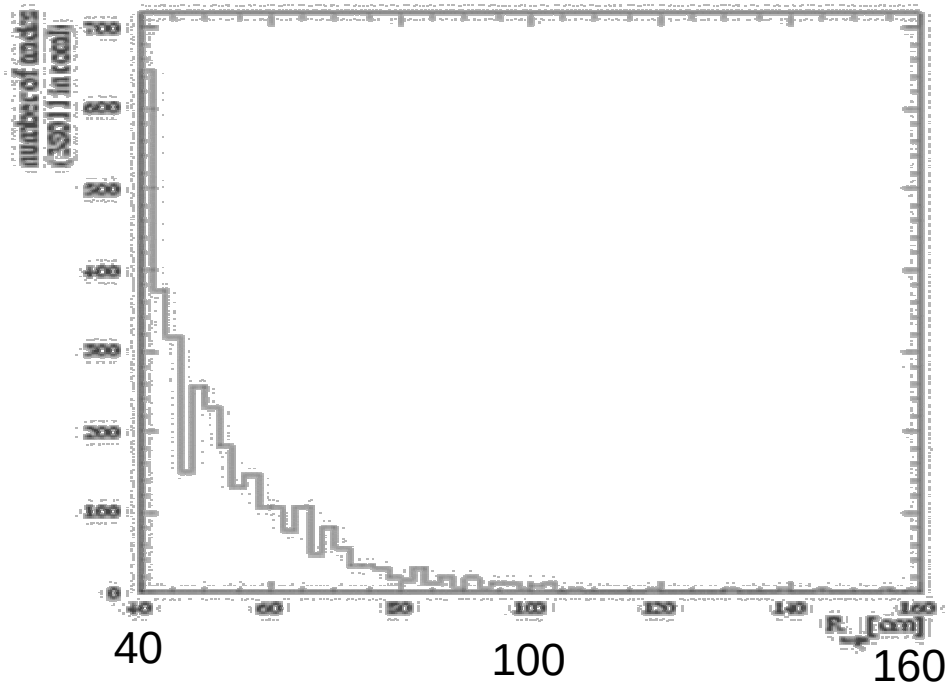
connect pieces to ambiguous track candidates (DELPLHI)

optimise the assignement of hits to tracks based on global event topology

main authors of package: Kristian Harder, Markus Elsing, Daniel Wicke, Richard Hawkings

The TPC pattern recognition

- Program is based on ALEPH pattern recognition
  - start from the outside of the TPC, find tracks.
  - remove “outliers”: drop hits with > n sigma from the track
  - treat close by tracks/ hits: drop any hits from the “confusion” region



number of tracks with “closeby” hits in dd events, vs. radius of last closeby hit found

approx. fraction of hits dropped: <5 %

The track fit

Track fit is based on a KALMAN filter algorithmus

- fast implementation taken from DELPHI software
- transformation into the helix track parameter space is done by a Taylor expansion around a reference trajectory.
- Iteration to obtain good convergance
- Material in the detector is modelled by simple surfaces (cylinders, disks, cones)
- The fit itself removes outliers
  - up to 3 measurement per candidate (chi^2 < 0.1%)
  - use detector ranking (make sure the less precise element is removed)

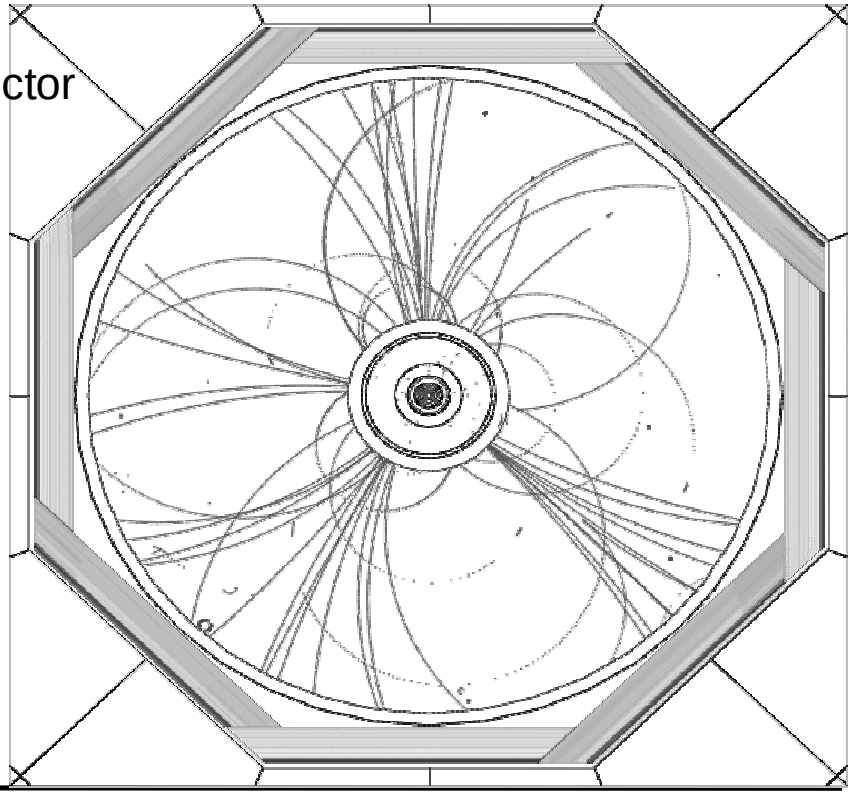
Fit has proven to be
very fast
quite stable and robust

only problem: outlier removal occasionally removes the full TPC segment

What needs to be done:
more systematic evaluation of the performance of the outlier removal logic
influence of the ranking in the tracking has to be studied

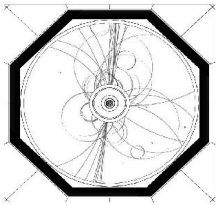
Performance Analysis

- Events used: dd events, 500 GeV
  tau events, 500 GeV, 3 charged particle decay
- Simulation includes backgrounds:
  - simulate random hits according to expected occupancy
  - not included: background tracks (they are tracks as all others...)
- For simulated tracks to be used:
  - p > 1 GeV
  - |cos(theta)| < 0.998
  - tracks from secondaries are rejected
  - should have left at least 3 hits in a detector
- For reconstructed tracks to be used
  - at least 3 hits on a track reconstructed



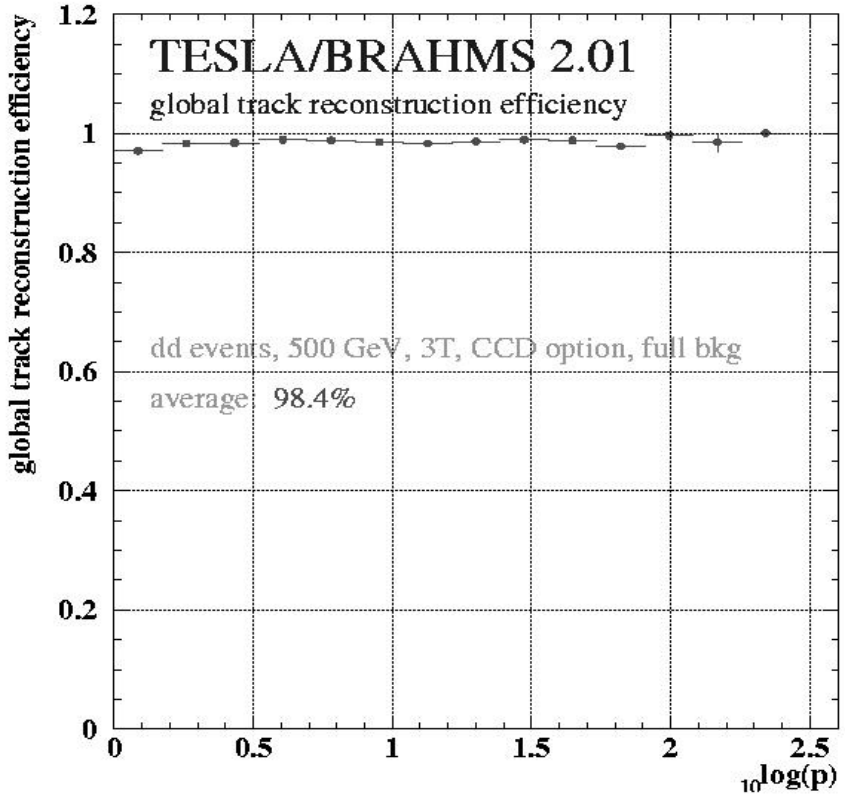
performance analysis done by
Kristian Harder, DESY
Details: LC-DET-2001-029

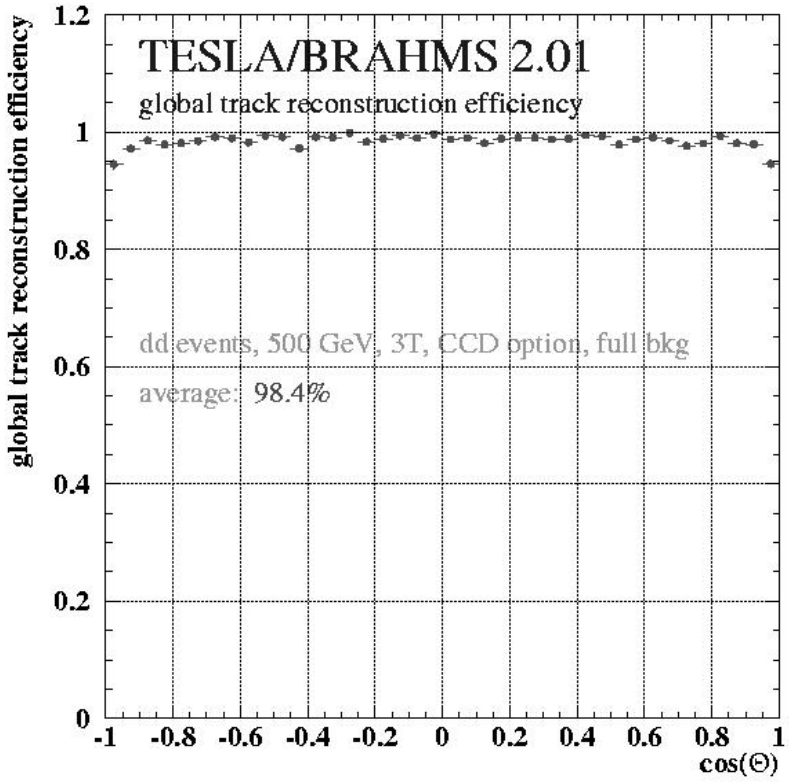
Overall Patrec Performance I



Overall tracking system, **dd events**, 500 GeV, 3T field, “full” background

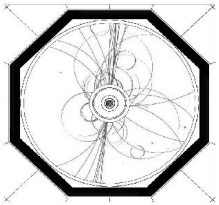
- The merging actually increases the overall efficiency slightly compared to individual sub-detectors





Efficiency: 98.4%

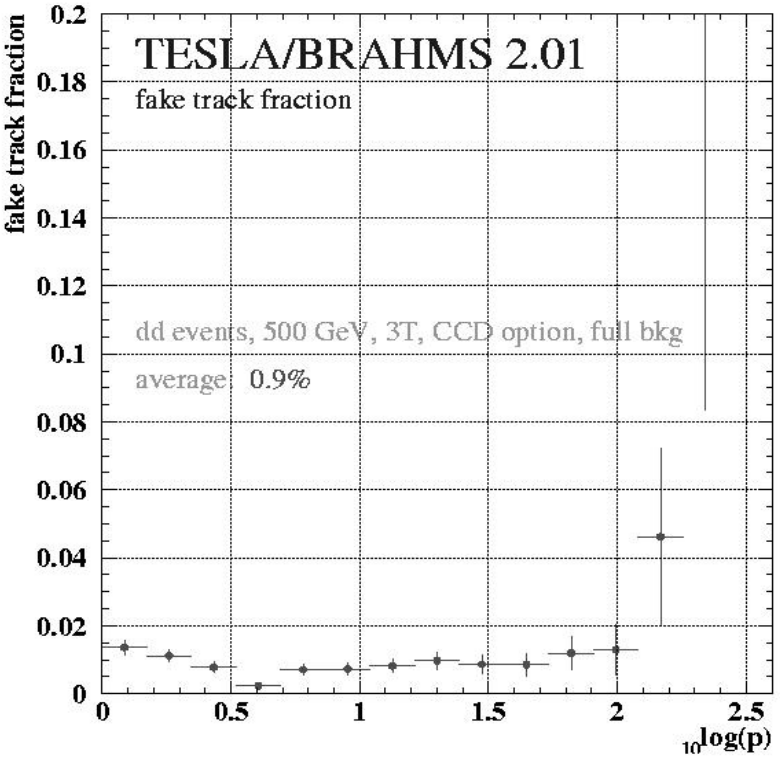
Overall Performance: Fake Rates, dd

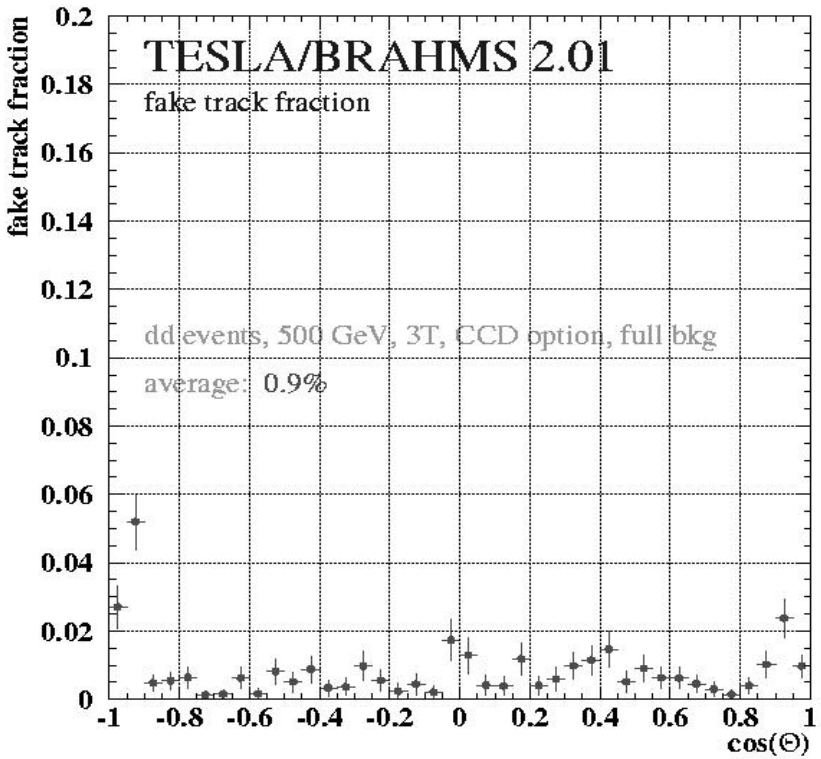


- Fake tracks: produce extra tracks (split tracks) to same parent

  dd events, 500 GeV, full background

  fake rate 0.9%, fairly flat





Tracking Performance Summary

A complete tracking reconstruction has been constructed and tested

Overall performance:

- tracking efficiency > 98% at 500 GeV (dd events, tau events)
- fake track fraction between 0.9 and 5%

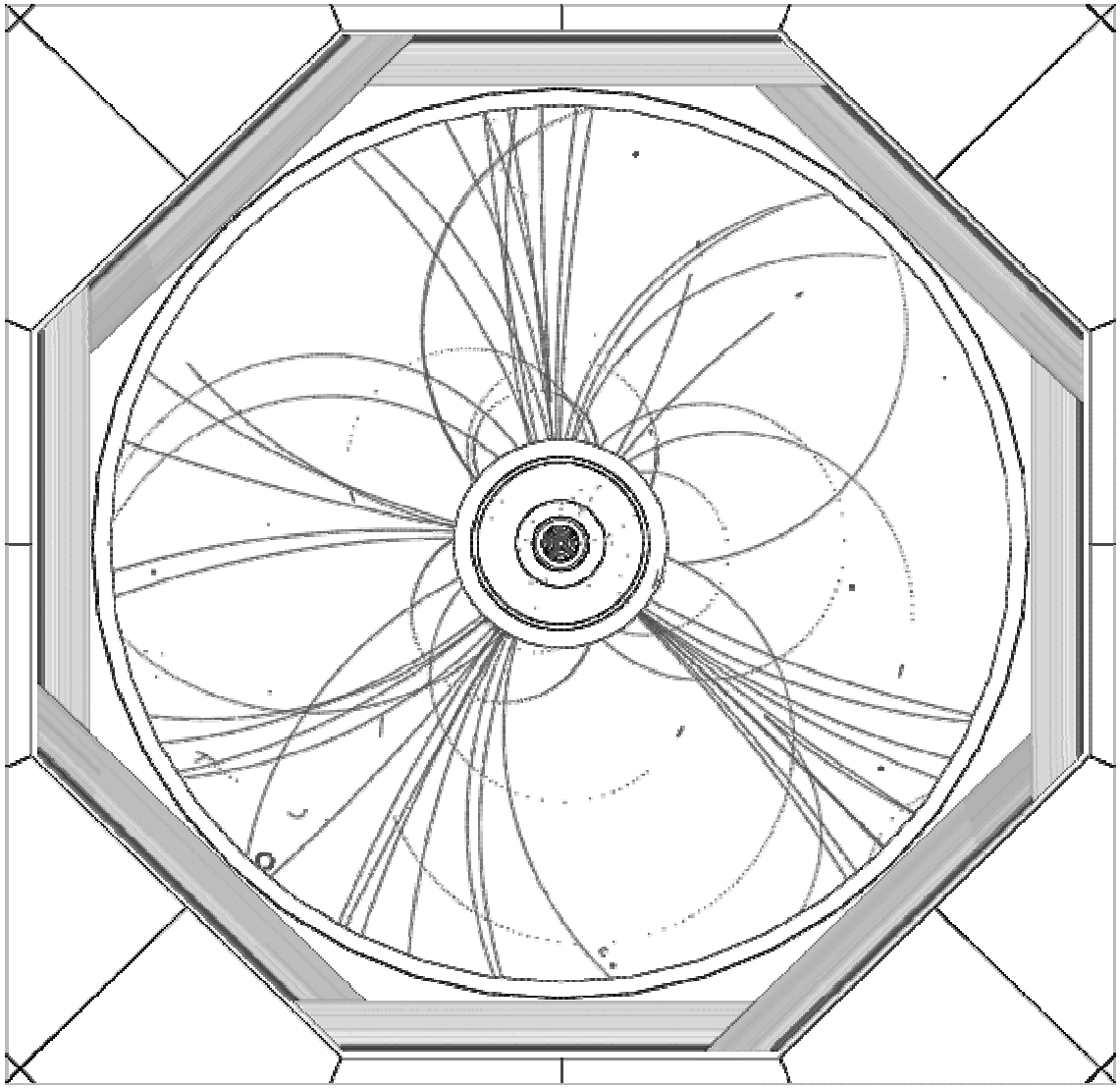
- System has been tested with and without background
- System is stable against reasonable amounts of random hit background

things to be done:

- more realistic background simulation: cluster, curlers, etc
- how about reconstruction of the correct BX?
- how about separation of signal from background?
- need a more thourough investigation of “V0” events
- speed needs to be improved in the presence of background (SI-VTX patrec)

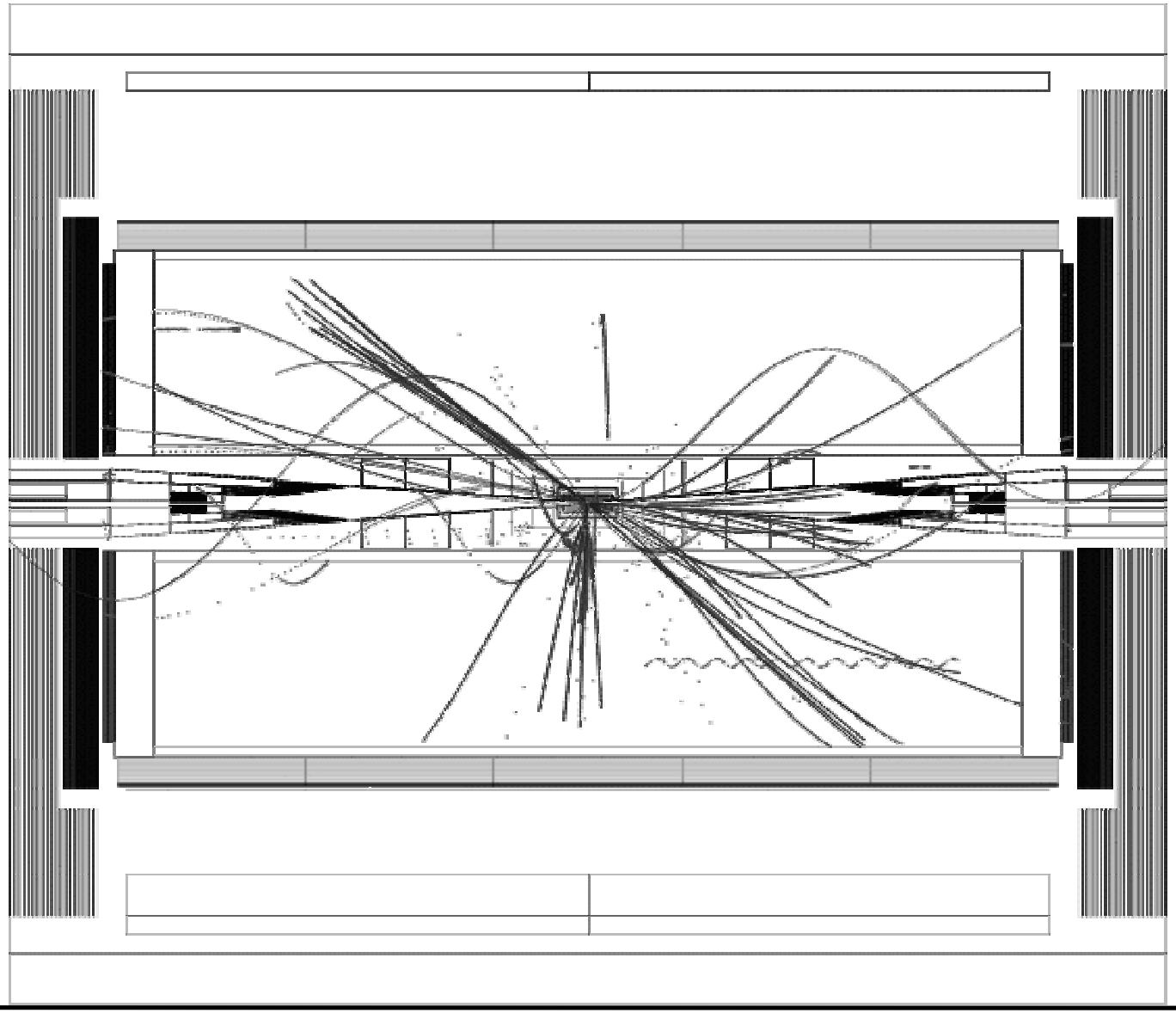
The system is a reasonable starting point and can be used for fairly realistic tracking studies.

“Visual” Performance

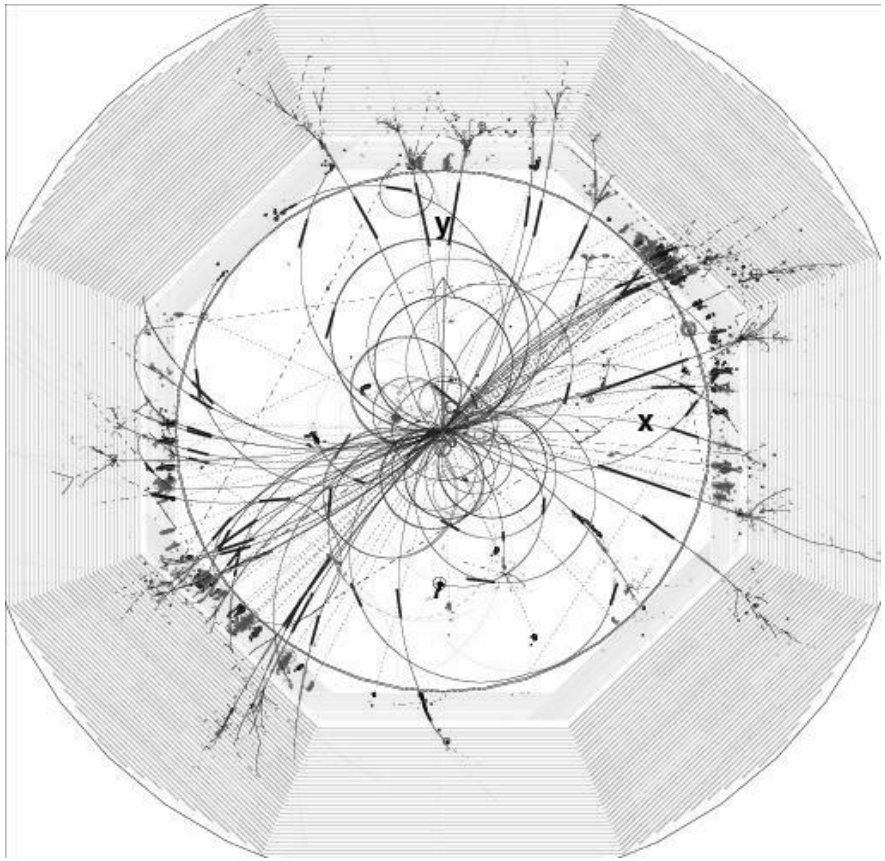


500 Gev,
top pair
no ISR
no Bkg.

Visual Performance



Calorimeter Reconstruction



tt event at 350 GeV, no ISR

The Goal:

Reconstruct the 4-momentum of all particles (charged and neutral) in the event

This is traditionally called "Energy Flow"

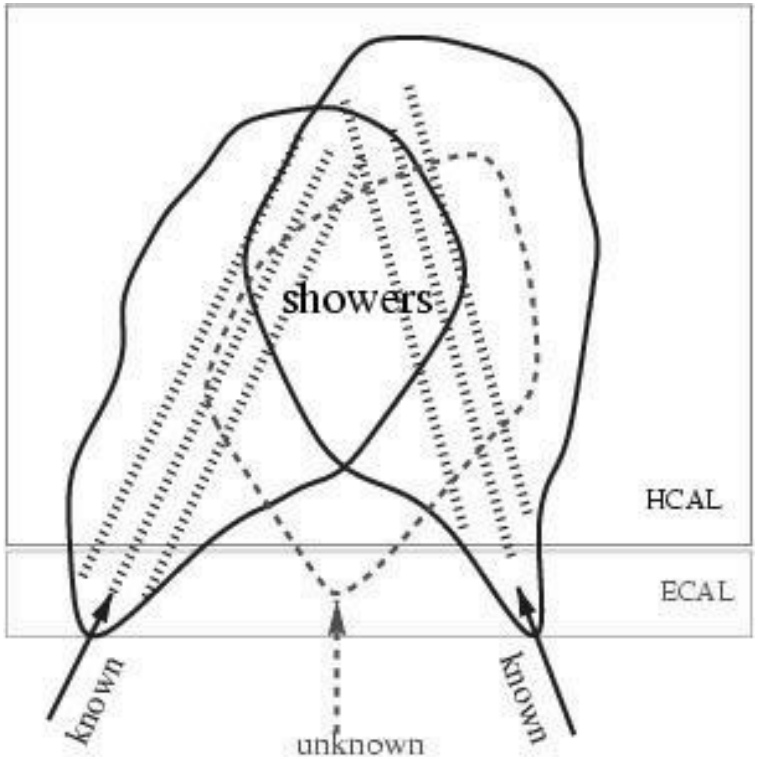
which is misleading.

It should really be called "**Particle Flow**"

Particle / Energy Flow in this context does not deal with event properties

but only with particles
Event properties are part of the analysis

The Algorithm: “SNARK”



A version of the energy flow has been realised in a reconstruction program:

SNARK, Author Vassilly Morgunov

which is part of the BRAHMS suite

- Tracks from charged particles in the tracker are linked to clusters in the calo
  Calo clusters have MAGNITUDE and DIRECTION!
- The associated energy in the calo is **substituted** by the more precise energy from the tracker
- Overlaps of showers are estimated based on magnitude and direction. Charged particle clusters are subtracted, to measure the neutral particles

The Algorithm II

1. Collect hits in the calorimeter along the predicted track (track core) within a distance of +/- one electronic cell.

2. Make a first particle hypothesis (e.g. MIP, ...)

3. Predict the transverse shower profile, collect more hits within the expected road

4. Iterate, until measurement and expectation agree best

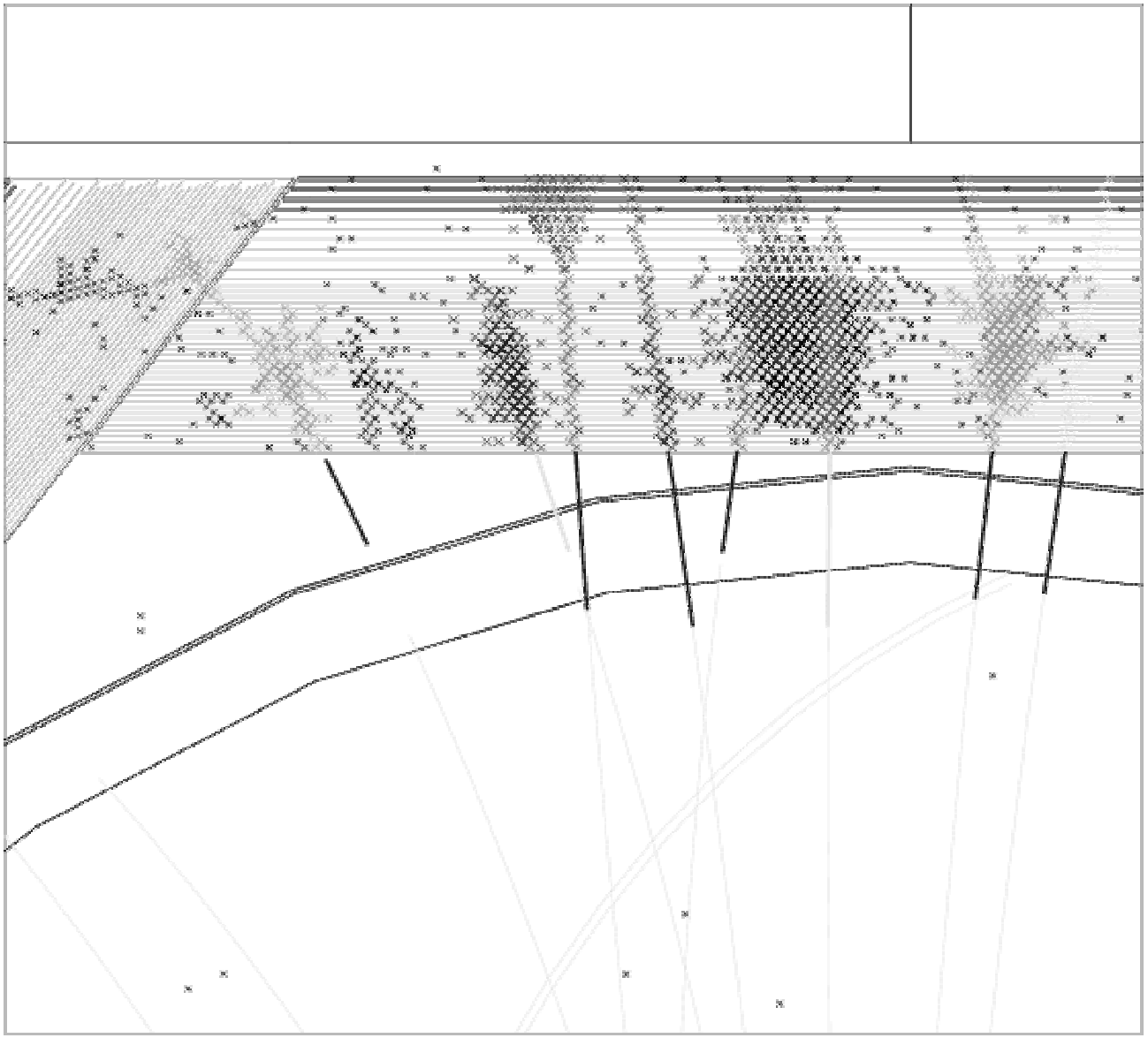
5. Any hits which at the end of the procedure are not associated belong to a neutral particle. Run “conventional” clustering, determine properties of neutral particle

The system depends on

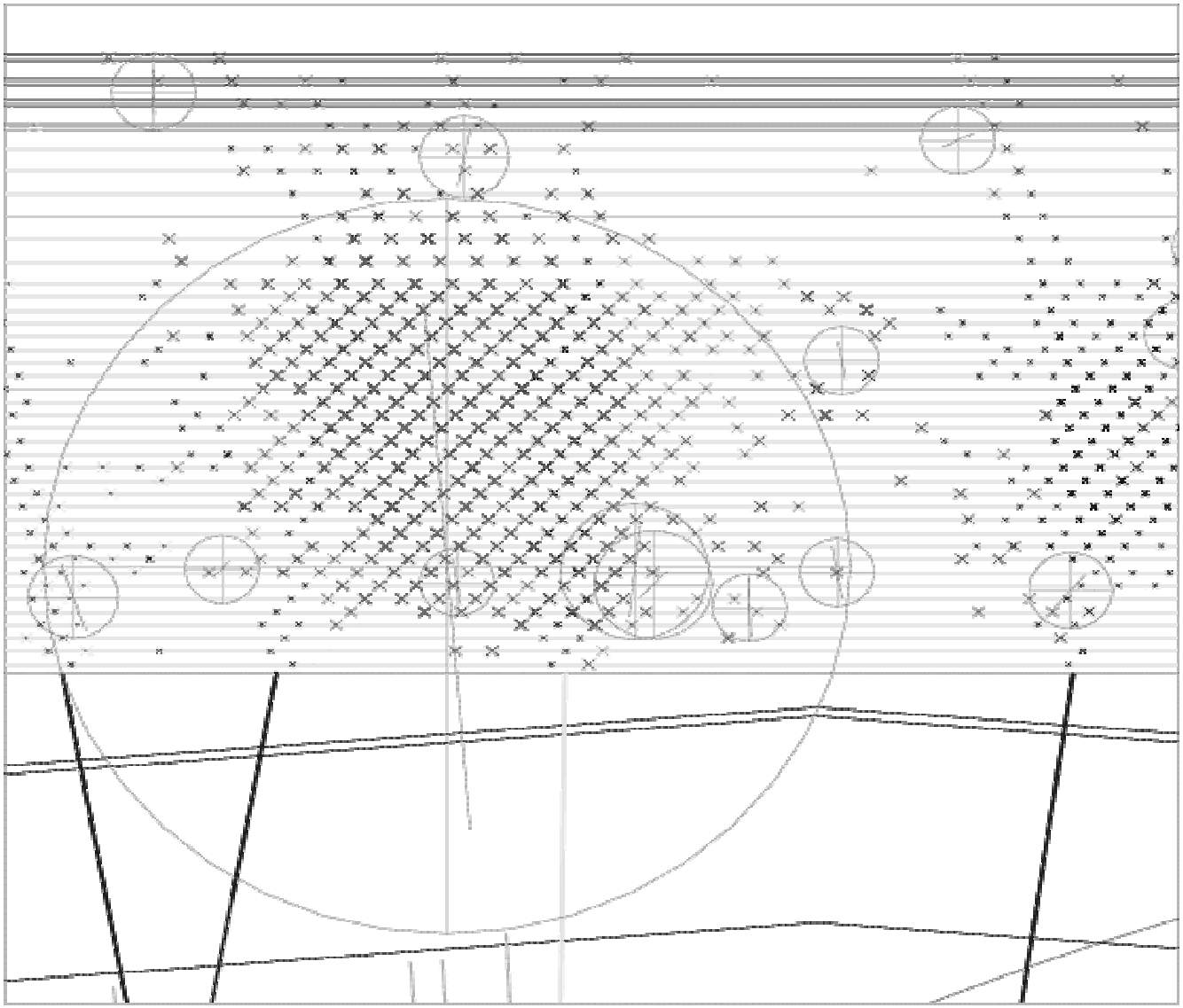
- high granularity both in ECAL and HCAL
- excellent linking between Tracker – ECAL – HCAL
- extensive use of amplitude info (optimised for tile HCAL)

Note: a similar program, but optimised for the digital HCAL, is also under development (Ecole Polytechnic)

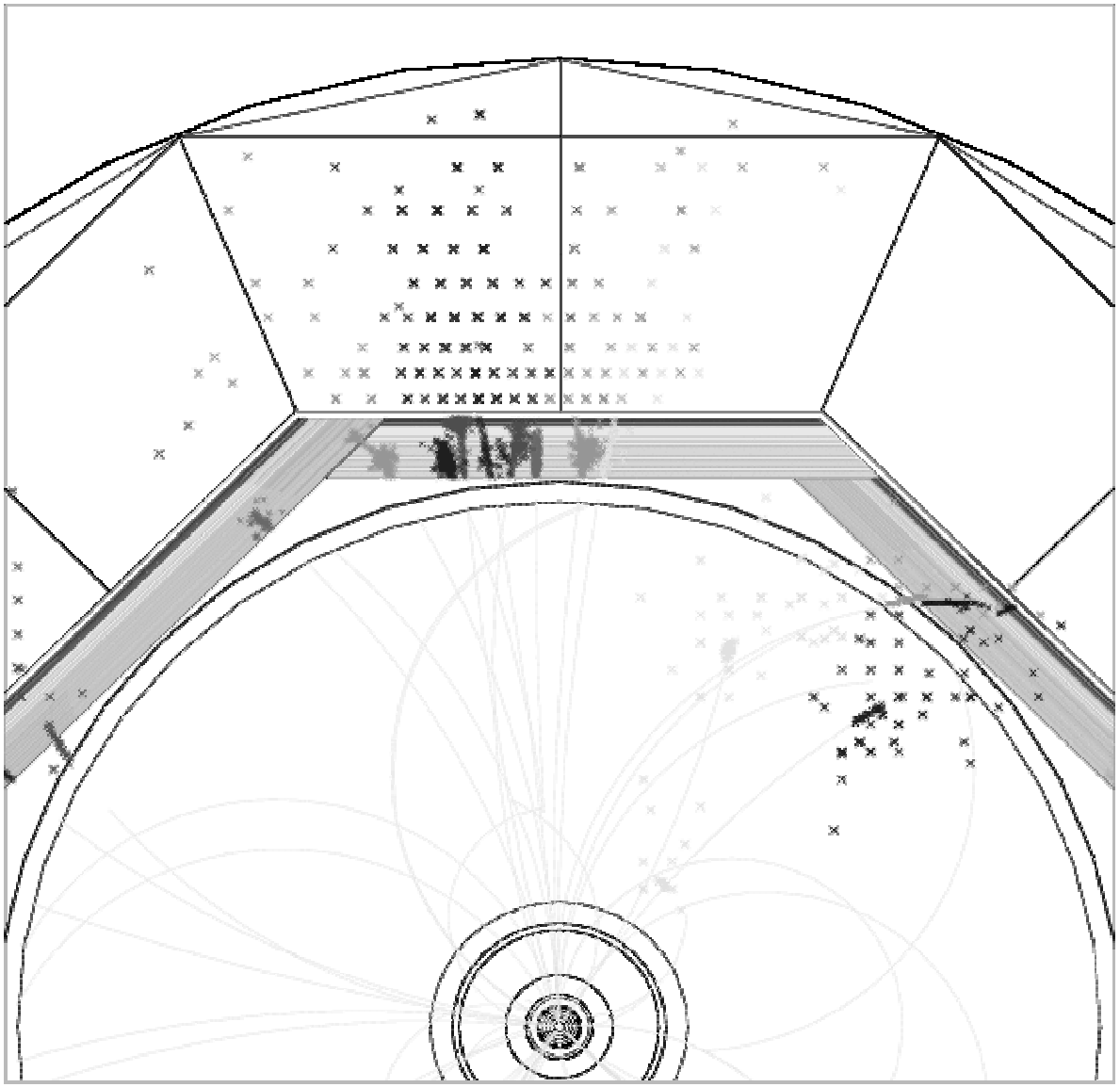
Some details



Some details



The HCAL (tile option)

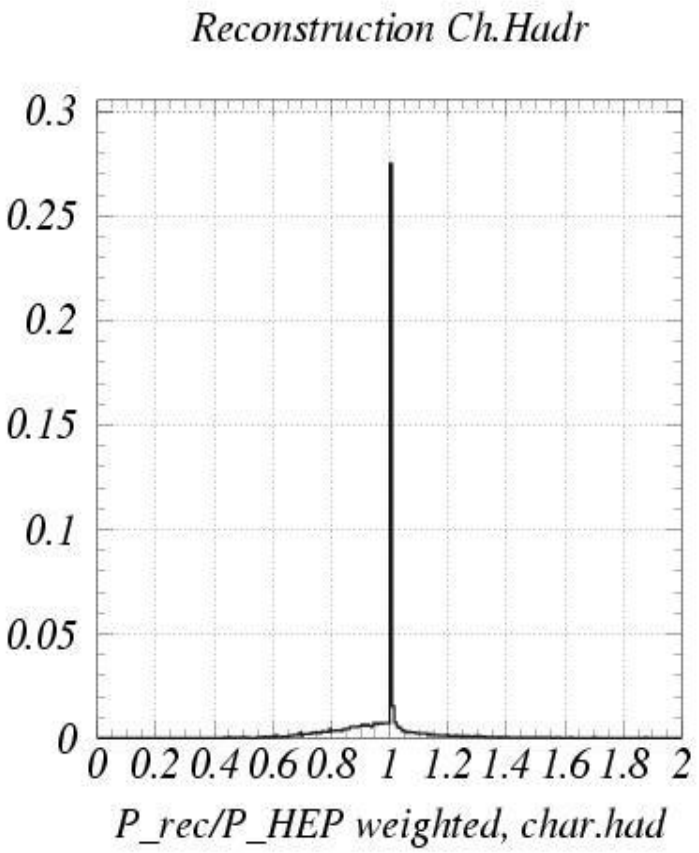


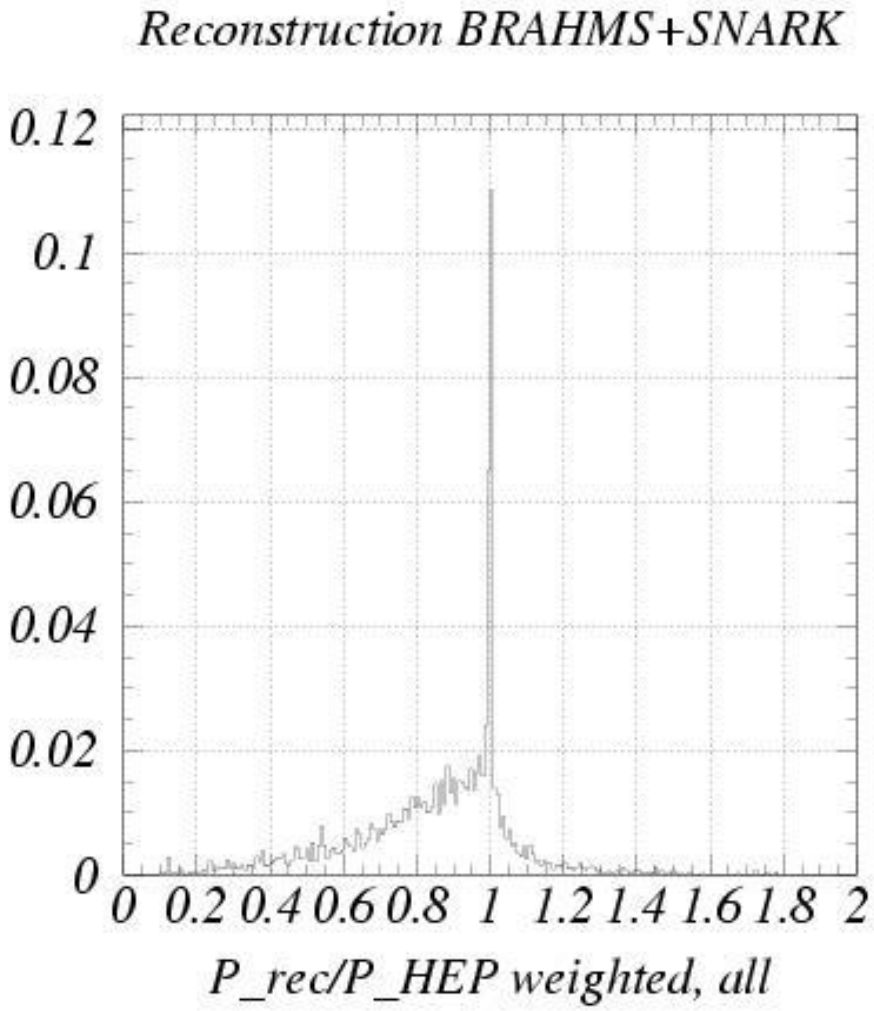
Some results

Compare reconstructed to expected Momentum: P_rec/ P_MC

No tree info on linkage used, only ordered by momentum

ZH -> hadron events at 500 GeV

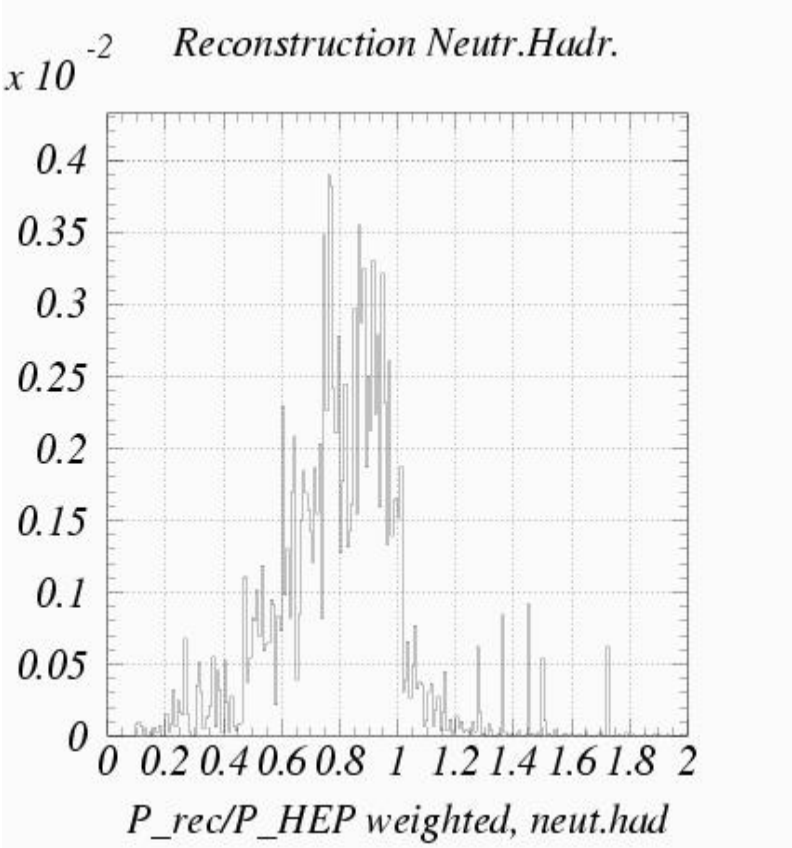




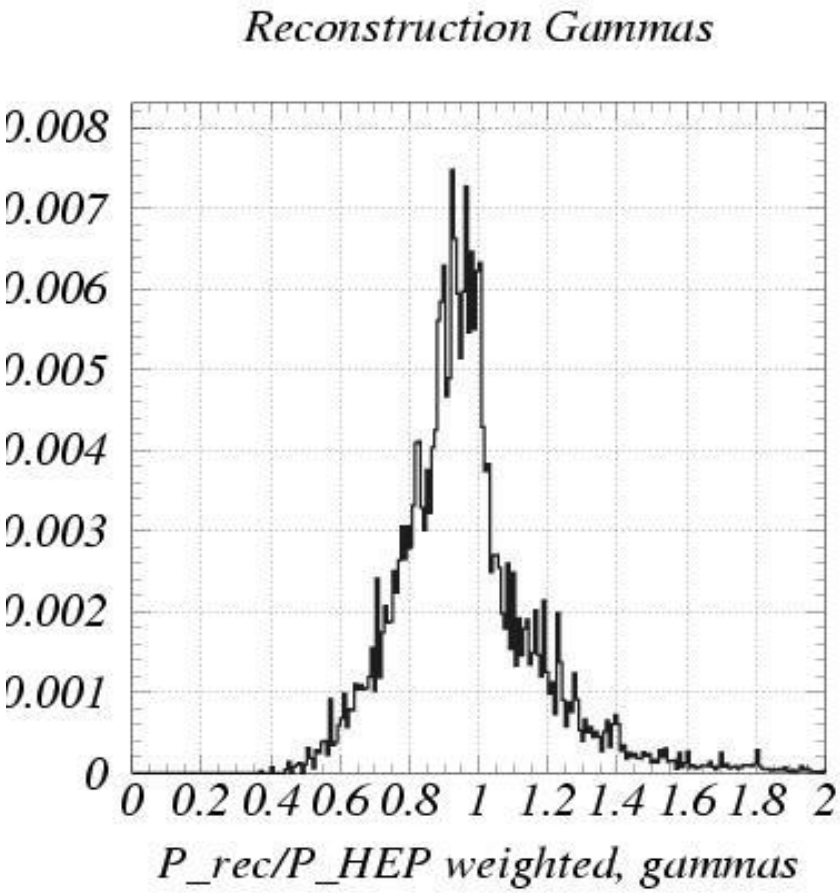
Neutrals/ Photons

Tails are from neutral particles:

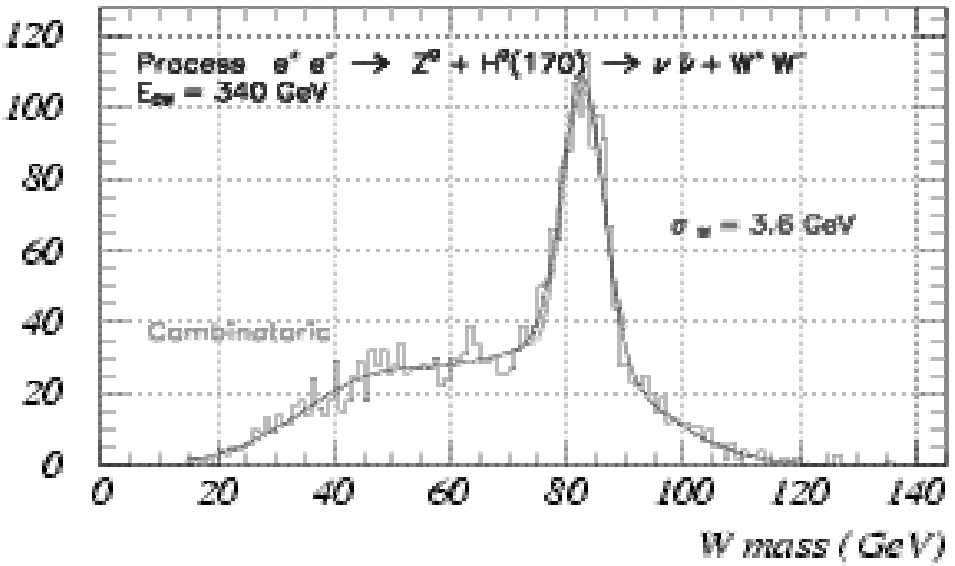
Neutral hadrons

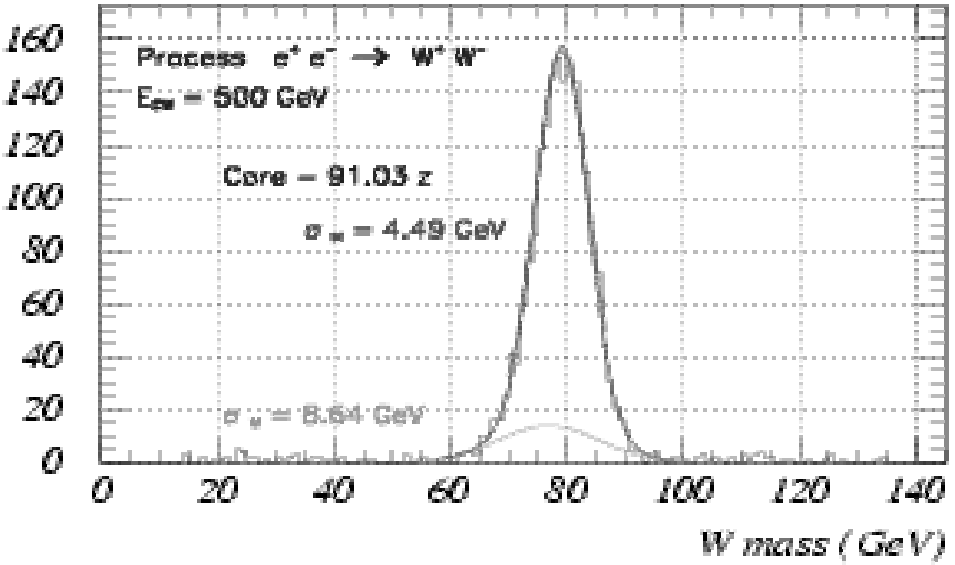


Photons



Some Physics Results





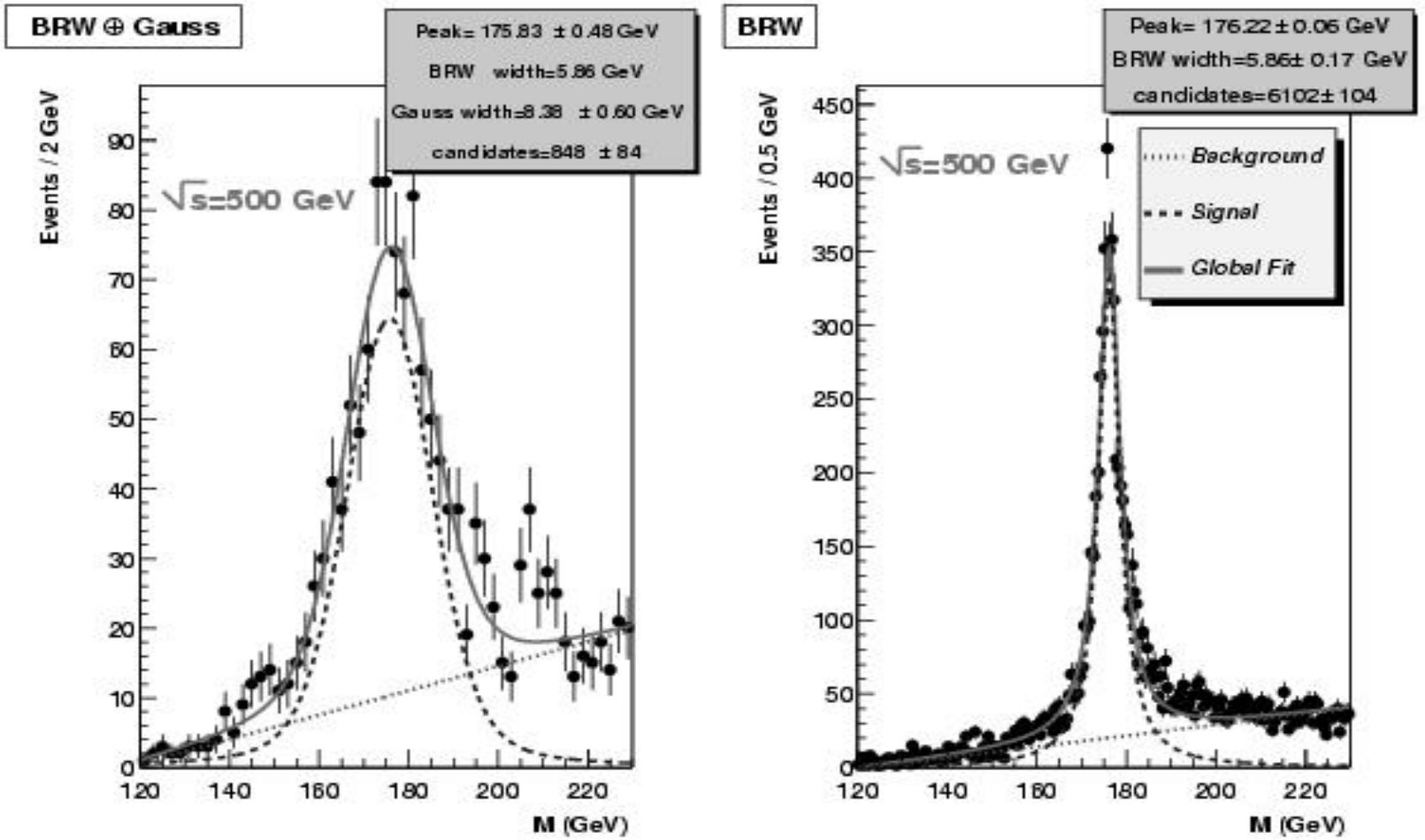
results are preliminary, and are only shown for illustration...

Vassilly Morgunov

Some Physics Results

Including detector and resonstruction effects

Intrinsic mass resolution



S. Chekanov, V. Morgunov

Next Steps

- basic program development done
- user interface now there (needs to be finalised / extended/ discussed)

- need detailed and systematic performance studies

  - ongoing: physics studies (top mass resolution, W mass resolution etc)
    - results (prel) will be shown in Prague next week

  - need much more low-level studies:
    - photon ID
    - hadron ID
    - cluster resolution
    - dependence on internal parameters
    - etc etc .

- need comparison to GEANT4 studies (MOKKA)

Technical Issues: Formats

- Hits after the simulation and digitisation:
  - stored in “HITS” file

BRAHMS: binary output (c-write routines)
gzip on the fly (ZLIB algorithm)

basic file structure:

- Start of Run record: run/ event number
- “constant record”: detector configuration at simulation time

Start of Event Marker

Subdetector Hits: Identifier, number of words/event
- hit record (typically x,y,z,E, but can be more)
- MC particle record (MC only)
- GEANT Volume record (MC only)

●
●
●

End of Event Marker

Record is for the most part self-describing (adding / removing detectors..)
User has to interpret however the information ....

The HITS format

- such a format is
  - simple
  - compact
  - transportable
  - fast
  - available for many languages (only needs ZLIB ...)
- it is not a optimised for object-oriented environments (pointers, ...)
- it does not have direct access (though a simple “fast skip” mechanism would be easy to implement)

The US SIO format is actually very similar, plus
- is has some pointer support
- it has fast skip

BUT: SIO does not exist for FORTRAN systems

The Final Output

- After reconstruction: store “energy flow objects” as main items:
- BRAHMS: adopt scheme already implemented in SIMDET, with some (minor) modifications

MC record

Eflow record

Status	Parameters	Pointers	Tracks	Calorimeter	Muon
type MC ID NGEN NTRK NCAL NMUON	px momentur py pz E Energy of m mass of the object Q charge of the object	MC record num Energy Fractio repeated ngen	Track Param p, theta, phi, l error matrix dedx informa repeated ntrk	Muon track/E info repeated nmuon times pointers to calo hits	

Accessing the information

- The output information is written to file: same format as hit file
  - can read with stand-alone application (FORTRAN, JAVA, C ...)
  - a JAVA interface is under development

- Internally the information is available through FORTRAN statement functions
  - simple array-like syntax, maps onto internal ZEBRA store
  - hides ZEBRA completetly from the user
  - allows the use of “pointers” in FORTRAN
  - fast

```
do neflow=1, nef(1)
  px = ref(1,neflow)
  nehit = ief_ne(neflow)
  if ( ref(8,neflow) .ne. 0. ) then

   ..............

 end if
end do
```

Internally the link to the hit information is maintained:
- track hit info
- calo hit info
- muon hit info ...

Next Steps

Try to standardise on a common hit and output format

- makes interchange of ideas and work easier
- makes reconstruction programs done in one world accessible to another
- makes the life much easier for anyone who wants to work with these things

Try to agree on a common CONTENT of the energy flow records

Try to remain open:

- in europe we do not plan to “force” people into one system (root, JAS, PAW ...)
- we have to be able to move a few years from now

BRAHMS: will continue to maintain, at low level, will be replaced by MOKKA (e.g.) in the forseeable future

RECO: the reconstruction suite will live much longer

BRAHMS: http://www-zeuthen.desy.de/linear_collider

pre-versions (no guarantee): http://www.desy.de/~behnke/brahms

A Comparsion of Showers

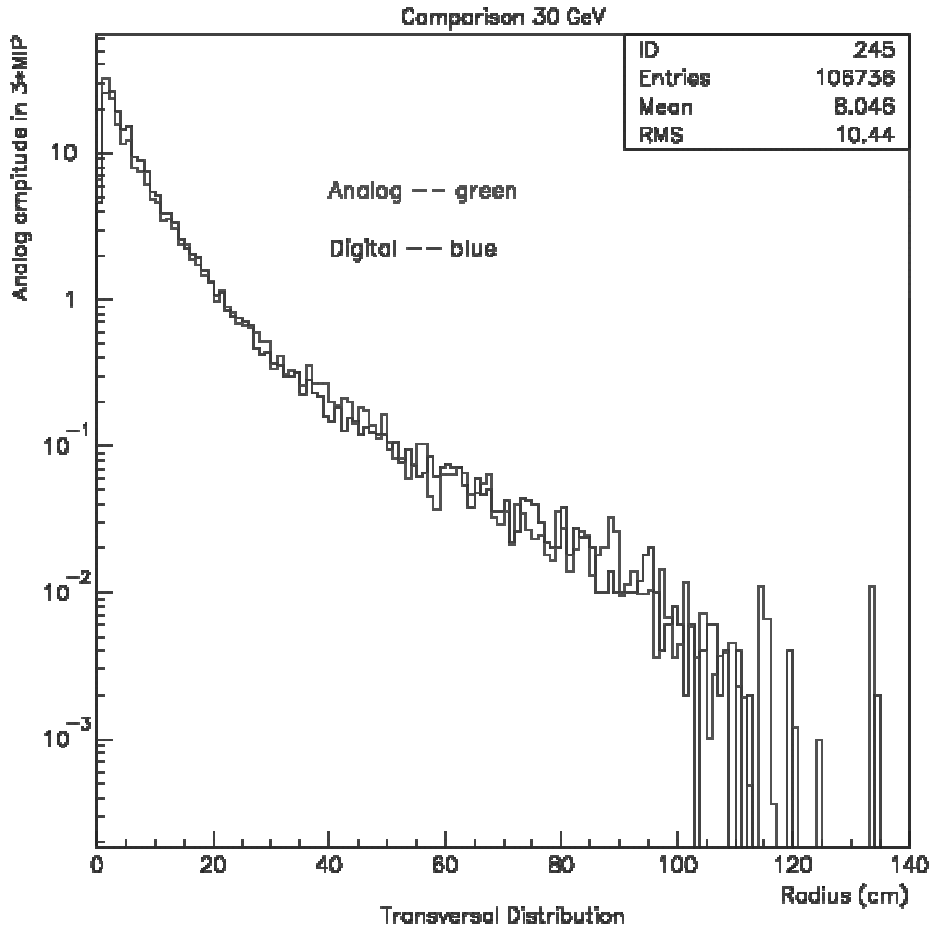
Compare the shower shapes as a function of detector medium:

- Digital HCAL will (probably) use gaseous detectors
- Tile HCAL will use scintillators

We expect some difference:

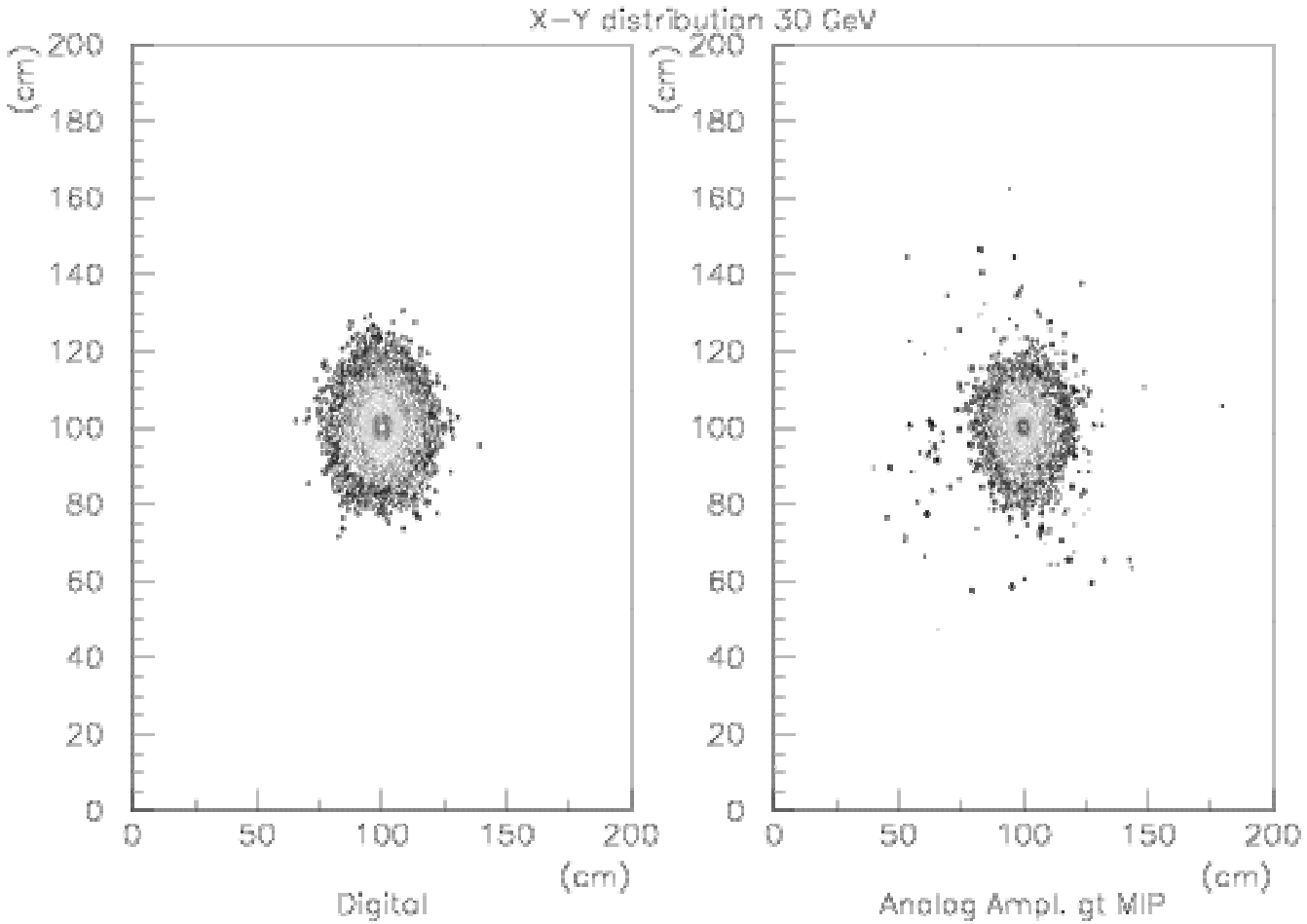
- Neutron cross-section in plastic is different than in gas: neutron component in plastic will be larger
- Scintillator is more sensitive to the (low-energy) photons produced in the shower: expect a “Halo” in the case of the scintillator

Comparsion Gas – Scintillator



Compare the shower shape in Scintillator and in Gas
same sampling structure

Comparsion Gas- Scintillator



Digital

Analog