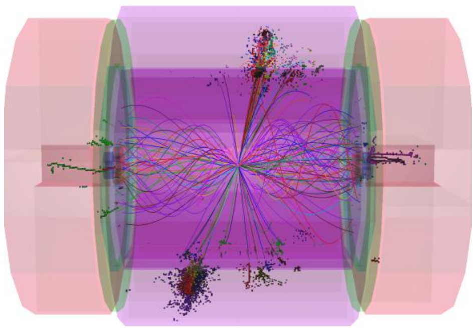


PandoraPFA

ILD Update

John Marshall,
University of Cambridge

ILD Workshop, LAL Orsay, May 22 2011

Overview

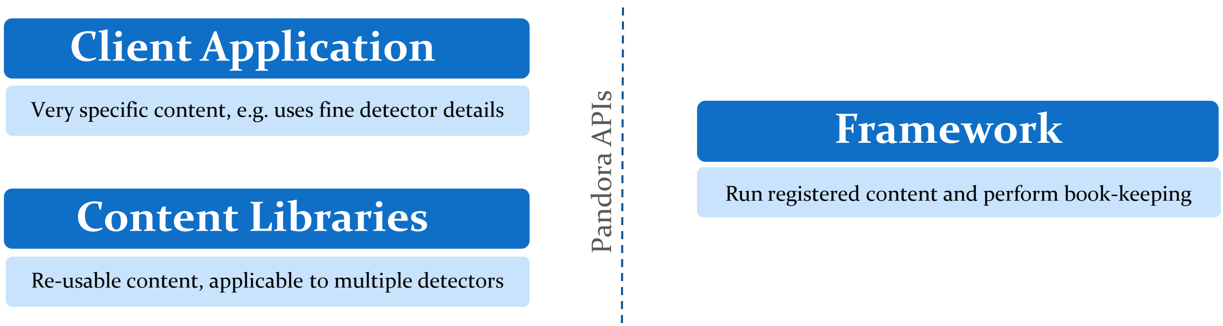
- Since the last presentation of PandoraPFA in an ILD meeting, many important changes have been made. Some changes were driven by requirements for the CLIC CDR.
- Some changes were to make Pandora algorithms more generic, and so applicable to other pattern-recognition problems. ChangeLogs list all modifications made since first public release.

- Today, will discuss a selection of the most interesting changes:
  - Restructuring of Pandora and opportunities for customising Pandora reconstruction,
  - Forced clustering algorithm and the transition to “energy flow” calorimetry,
  - Monitoring of energy changes during reclustering, to assess reconstruction quality,
  - Efficiency improvements to reduce CPU time in presence of overlaid background events,
  - Photon reconstruction algorithm, to improve efficiency and accuracy of photon identification and reduce confusion for subsequent Pandora reconstruction,
  - Current jet energy reconstruction performance for ILD.

- Will start with some of the more technical changes…

Pandora Structure

- A powerful feature of Pandora framework is the ability to register content (algorithms, particle id functions, etc.) from different libraries and combine their functionality in the reconstruction.
- Have used this ability to help re-structure Pandora:



- The idea is that a Pandora client application registers the content it needs to perform its specific reconstruction within the framework.
- Content can often be re-used for different detector models, so can be bundled together. ILD-applicable content lives in the FineGranularityContent library, which offers 60+ algorithms, particle id functions, a pseudolayer calculator and a shower-profile calculator.
- FineGranularity assumes only the presence of inner tracker, fine ECAL, coarser HCAL and yoke.

Pandora Development Kit

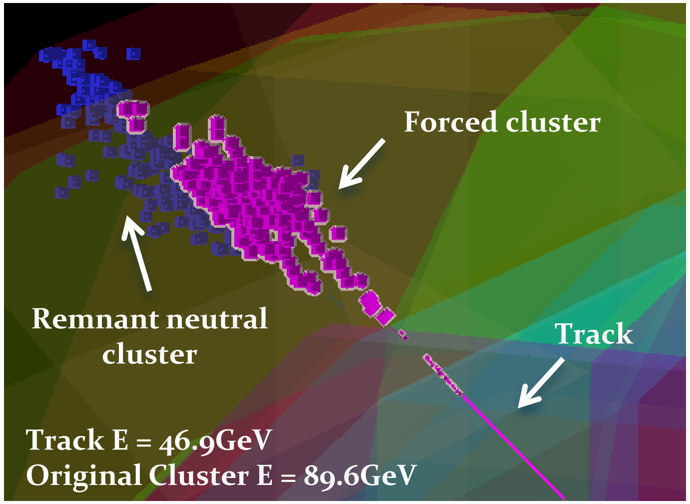
The following content can be customised and registered with Pandora, then configured via xml:

Content	Description
Algorithms	Responsible for performing reconstruction; make use of all information provided by objects, helper functions, calculators, etc. to make decisions and create PFOs.
PseudoLayer calculator	Responsible for dividing hits into layers that broadly follow structure of detector; helps to isolate algorithms from need to know specific geometry.
B-field calculator	Responsible for providing signed B-field value for given Cartesian coordinates; often a wrapper for a full field map in client software framework.
Shower-profile calculator	Responsible for examining longitudinal and transverse profile of cluster energy deposits and performing comparison with expectation for EM shower.
Particle id functions	Responsible for providing ("fast" or "full") particle id information to algorithms, which may want to avoid certain particle-types, or simply apply results to PFOs.
Energy corrections	Responsible for applying corrections, improvements or custom calibrations to reported hadronic or electromagnetic cluster energy values.
Geometry	Optional detector description, which can be used by an algorithm if necessary.
Objects	Self-describing properties for tracks, hits and optional MC particles.

Forced Clustering

- In the Pandora framework, a transition from particle flow to "energy flow" calorimetry can be made when it is evident there is a problem that cannot be fixed by reclustering.
- Simply add new "forced clustering" algorithm to the list of algorithms to be used in reclustering.
- For a poorly matched track and cluster, the algorithm will select the relevant hits to force compatible track cluster energies, as follows:

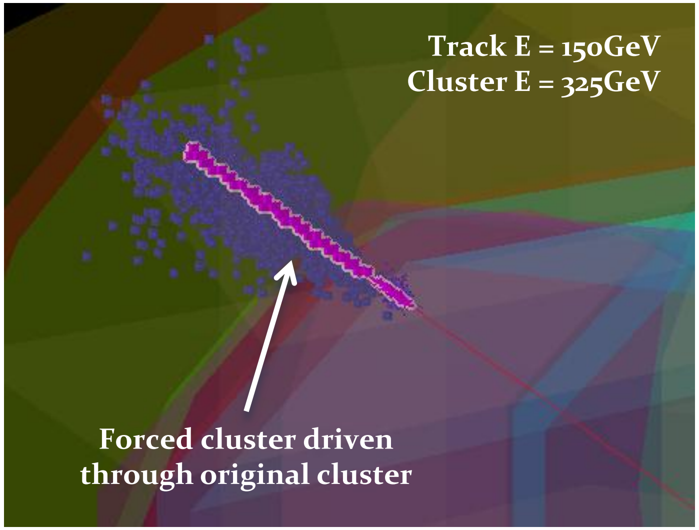
  - The track helix fit is extrapolated into the calorimeters and the distance between the helix and each available hit is calculated.
  - A new track-seeded cluster is created (the "forced cluster") and the distance-ordered hits are added to the cluster until the cluster energy matches the track energy.
  - Any remnant hits are clustered using an instance of the standard clustering algorithm.



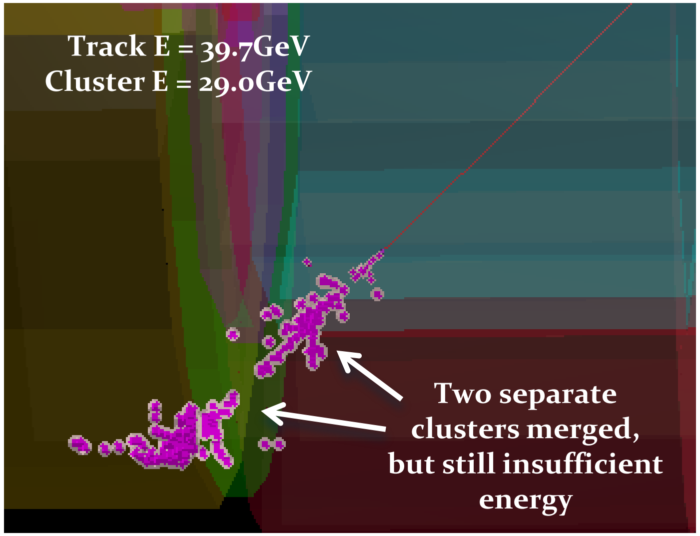
Forced Clustering

There are two use-cases for forced clustering:

1. **ClusterE > TrackE.** In this case, hits remaining after the creation of the forced cluster are grouped using the standard Pandora clustering algorithm.
2. **TrackE > ClusterE.** In this case, the parent reclustering algorithm must identify likely fragments near the original cluster. These are typically clusters lying in a cone along the track direction, but without track associations. The additional clusters are added to the reclustering list, so their constituent hits are available to the forced clustering algorithm.



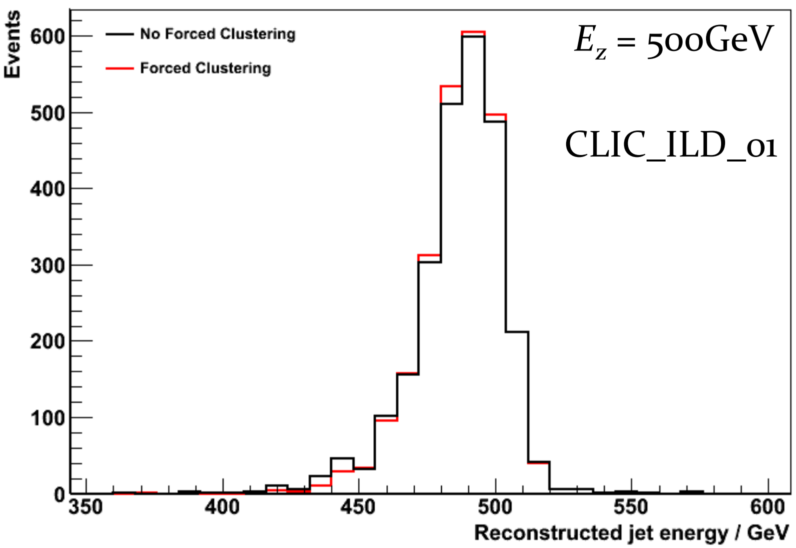
1 . ClusterE > TrackE

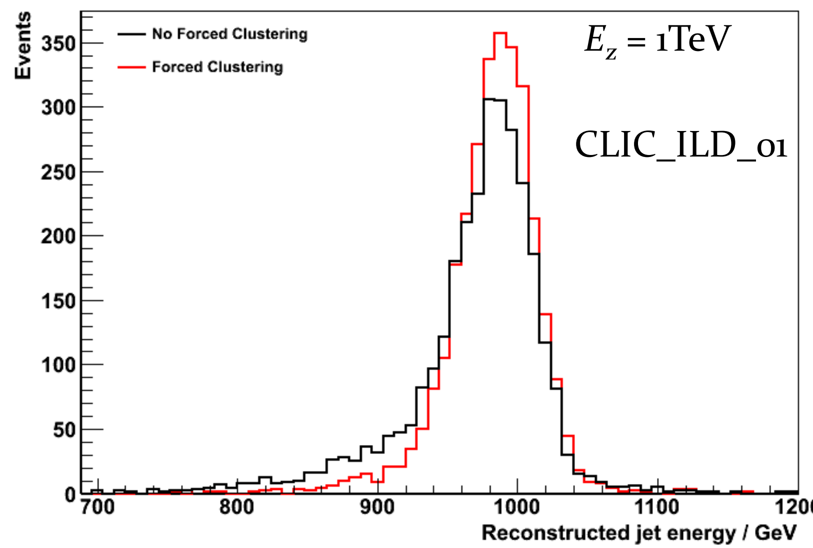


2. TrackE > ClusterE

Forced Clustering

Performance tests with $Z \rightarrow uds$ events show that forced clustering is important at high energies.



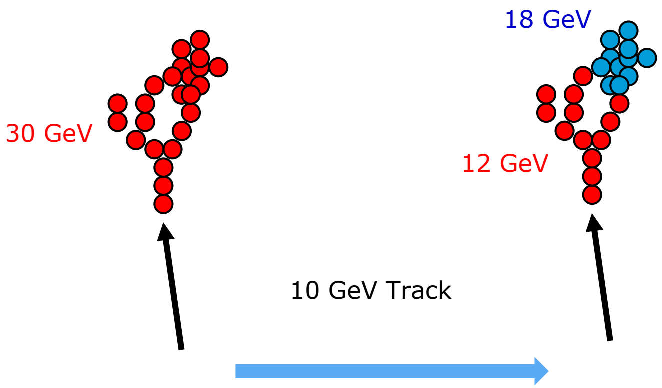


CLIC_ILD_01, E_z (= 2 * E_j)	91GeV	200GeV	500GeV	1TeV
PandoraPFANew, No Forced Clustering, $rms_{90}(E_j)$ / E_j	3.56 ± 0.11	2.94 ± 0.08	**3.00 ± 0.07**	**3.97 ± 0.09**
PandoraPFANew, Forced Clustering, $rms_{90}(E_j)$ / E_j	3.56 ± 0.11	2.94 ± 0.08	**2.88 ± 0.07**	**3.01 ± 0.07**

Performance is quoted in terms of rms_{90}, the rms in the smallest range of reconstructed energy containing 90% of the events. The total energy is reconstructed and the jet energy resolution obtained by dividing the total energy resolution by √2. A cut on the polar angle is applied to avoid the barrel/endcap overlap region: $|\cos\theta| < 0.7$

Recluster Monitoring

- To help assess the quality of the Pandora reconstruction, we can monitor how much work the reclustering needs to do in order to enforce consistency between the energies of associated tracks and clusters.
- The Pandora framework now does this automatically. For each track, the framework records the changes in associated cluster energy that occur between the API calls to initialize and end the reclustering.
- An API is available to access the results, in the form of the net energy change, sum of moduli of energy changes and square root of the sum of squared energy changes.
- MarlinPandora currently writes out collections of LCIO generic objects to store the energy changes and LCIO relations to link the changes to the relevant tracks.

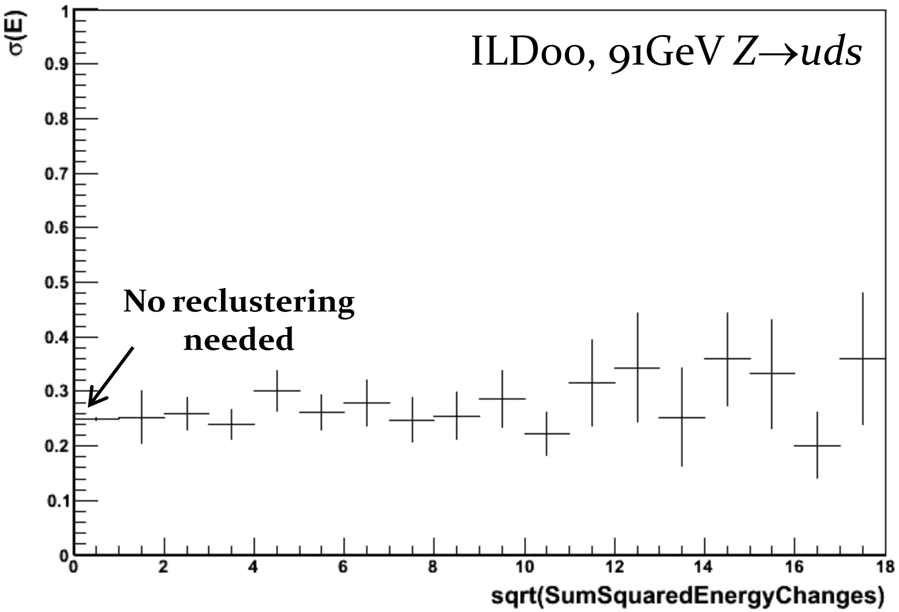


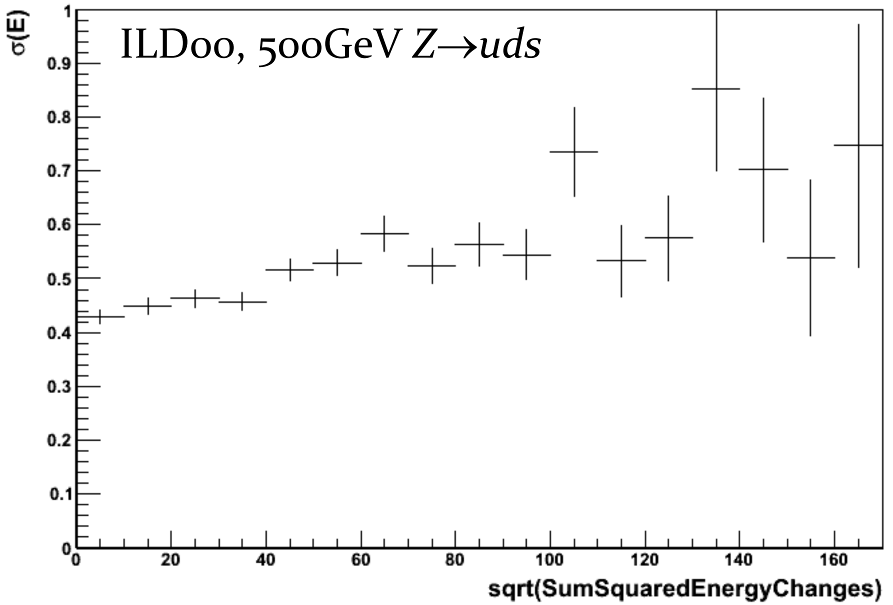
Change clustering until obtain sensible track-cluster match

18GeV change recorded for this track if/when new cluster candidates are accepted

Recluster Monitoring

- Recluster monitoring results can be quickly examined using the simple PandoraAnalysis package.

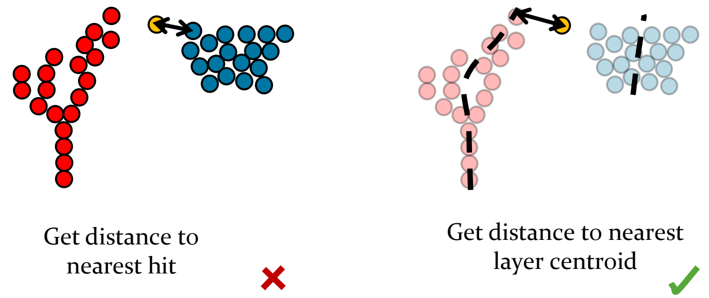




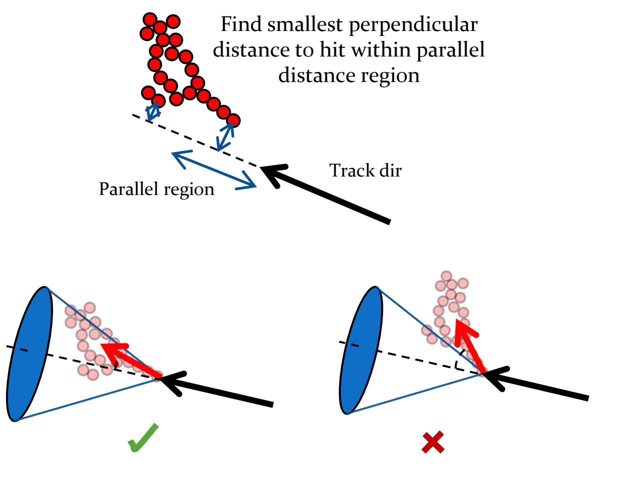
- Results for $Z \to uds$ events in ILDoo show that there is some correlation between the width of the reconstructed jet energy distribution and the energy changes that occurred during reclustering.
- Size of the energy changes should be related to the level of confusion posed by an event.
- Energy change values are confirmed correct, but results/correlation not yet fully understood.

Efficiency Improvements

CPU time required by Pandora has been reduced, with an emphasis on improving performance in presence of $\gamma\gamma \rightarrow$hadrons background.



2. Avoid nested loops over all hits in cluster layers



1. Avoid comparison of tracks/clusters with very different directions

```
CartesianVector::CartesianVector() :
    m_x(0.f),
    m_y(0.f),
    m_z(0.f),
    m_isInitialized(false)
{}

CartesianVector::CartesianVector(float x, float y, float z) :
    m_x(x),
    m_y(y),
    m_z(z)
{}
```

3. Remove default constructor, so no longer need to check initialization state in member functions

Efficiency Improvements

Function	Revision 1100	Revision 1137
ConeClusteringAlgorithm::GetGenericDistanceToHit	29.802s	28.010s
IsolatedHitMergingAlgorithm::GetDistanceToHit	32.070s	15.310s
ClusterHelper::GetTrackClusterDistance	42.839s	10.450s
ClusterHelper::GetDistanceToClosestHit	15.620s	10.150s
ConeClusteringAlgorithm::GetDistanceToHitInSameLayer	20.070s	9.742s
ClusterContact::HitDistanceComparison	11.077s	9.089s
Cluster::GetCentroid	8.950s	8.730s
CaloHitHelper::GetDensityWeightContribution	6.739s	(6.230s)
CartesianVector::GetCosOpeningAngle	0.190s	5.540s
ConeClusteringAlgorithm::FindHitsInSameLayer	5.399s	5.158s
CaloHitHelper::IsolationCountNearbyHits	4.139s	4.040s
ClusterHelper::GetDistanceToClosestCentroid	3.661s	3.019s
CartesianVector::GetUnitVector	0.070s	2.430s
ConeClusteringAlgorithm::GetConeApproachDistanceToHit	3.870s	2.160s
FragmentRemovalHelper::GetClusterContactDetails	2.591s	1.870s
ConeClusteringAlgorithm::FindHitsInPreviousLayers	2.360s	1.691s
TrackClusterAssociationAlgorithm::Run	4.279s	1.600s
CaloHitHelper::GetSurroundingEnergyContribution	1.781s	(1.529s)

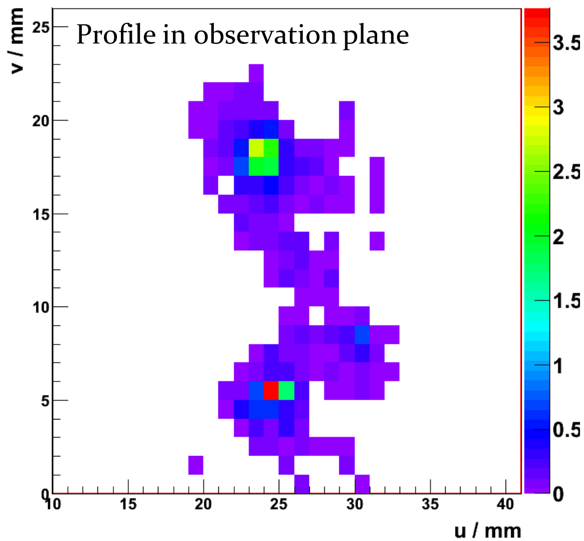
CLIC_ILD_CDR, overlaid background, NBackground=3.2

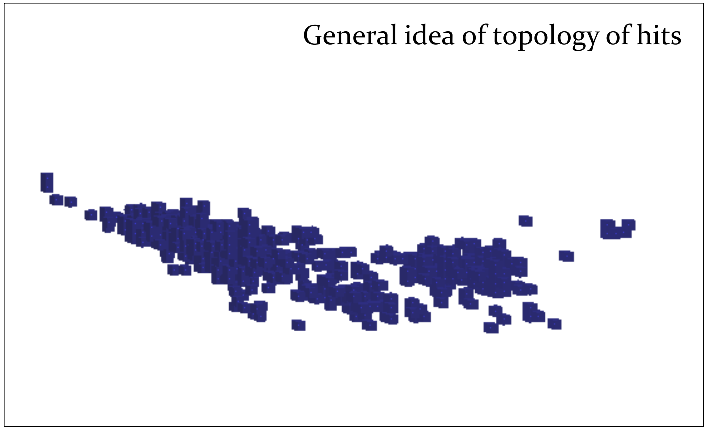
Analysis of `PandoraPFANew` before and after efficiency improvements: observe how load has been redistributed. There is an large overall decrease in CPU time.

Photon Reconstruction Algorithm

The photon reconstruction aims to reconstruct, tag and remove all photons before the standard Pandora reconstruction, reducing confusion and improving the jet energy reconstruction.

1. The cone-based clustering algorithm is applied to the ECAL hits, with all of its track-seeding options disabled. The transverse shower profiles of the clusters are then examined in detail. Any peaks in the profile are identified and characterised.

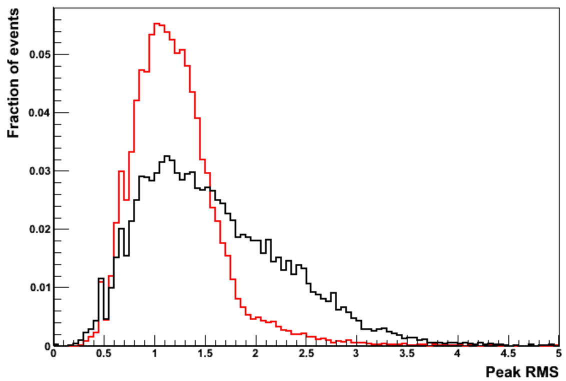


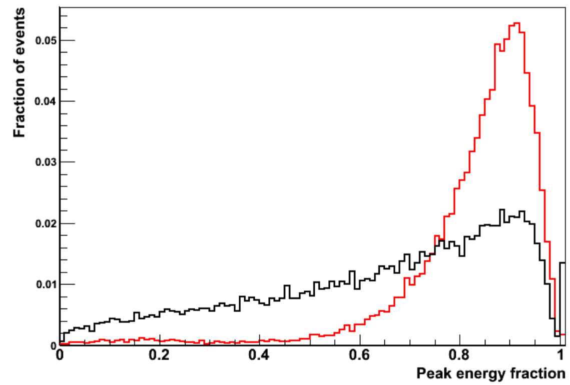


Not originally identified as a photon, but 17GeV from the 30GeV cluster is actually from a true photon. This is evident from the profile – split cluster up.

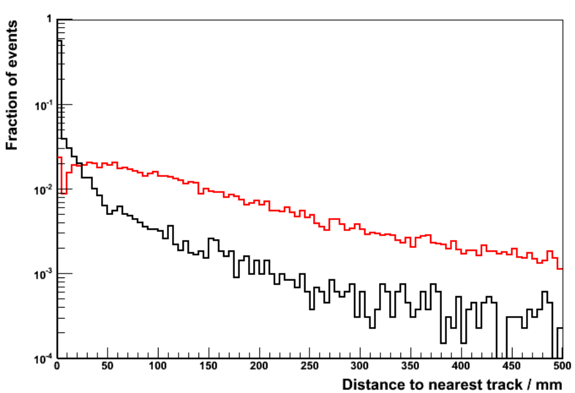
Photon Reconstruction Algorithm

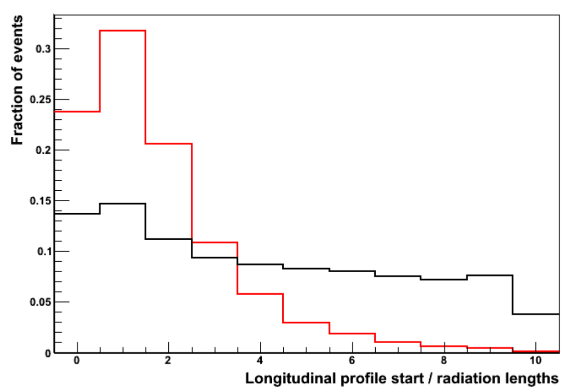
2. For each peak, a new photon cluster candidate is created and examined. Cuts are placed on the longitudinal shower profile of the new cluster and a multivariate/PID analysis is used to decide whether to accept the cluster as a photon.

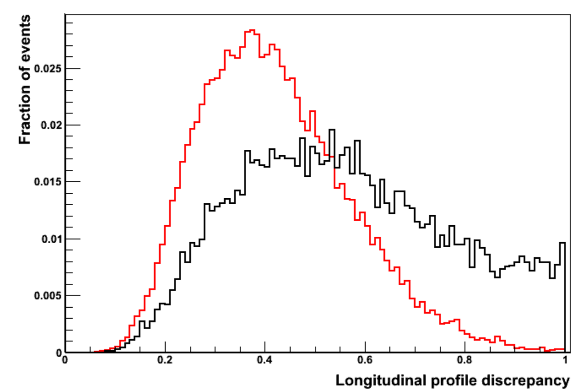




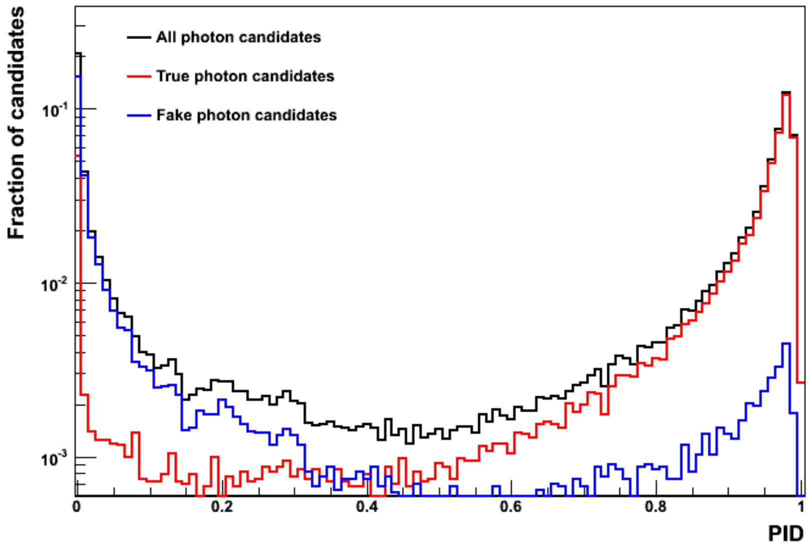
PDFs used for photon identification, constructed using 500GeV ILD00 Z→uds events

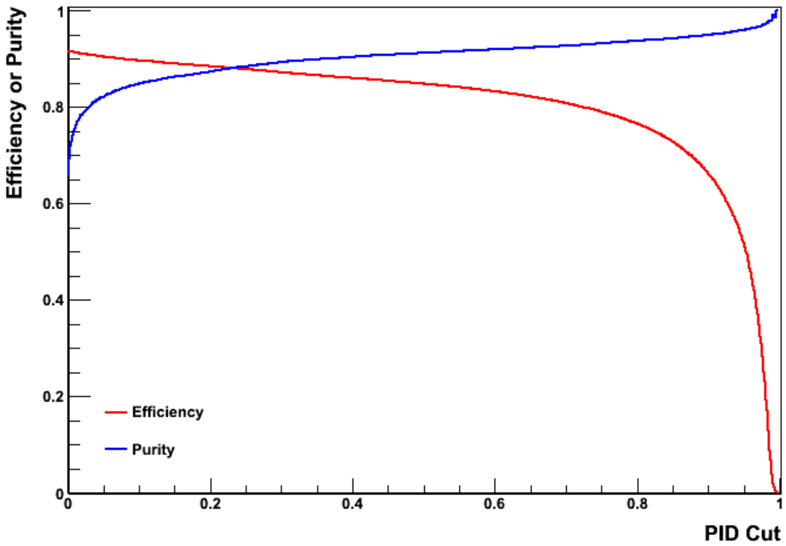






Photon Reconstruction Algorithm



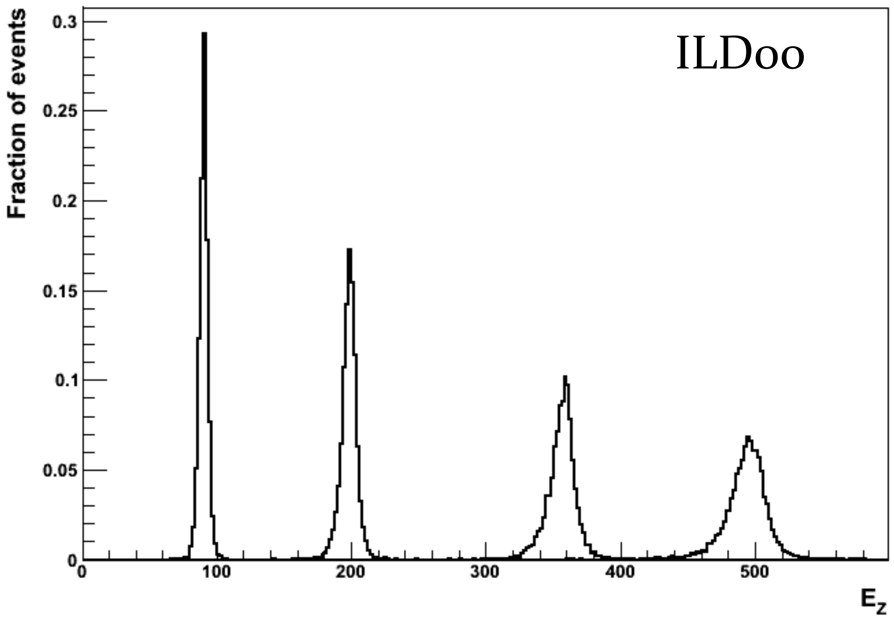


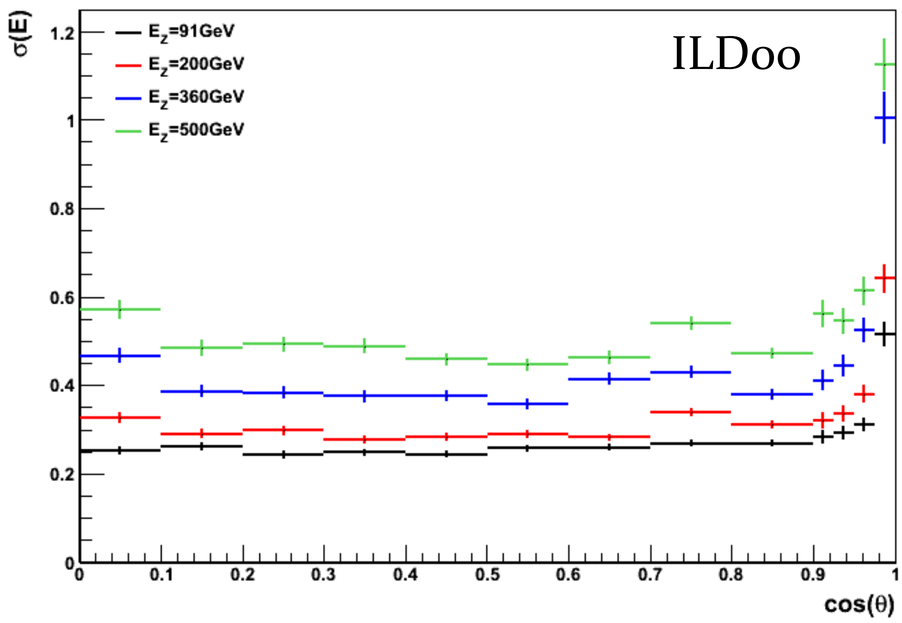
3. If a peak cluster is accepted, it is tagged as a photon and saved; the original cluster is deleted. If the peak represents the majority of the energy in original cluster, original may be used instead.

   With the exception of the addition of isolated hits, the photon clusters can remain unchanged and can be used to form photon particle flow objects in the PfoCreation algorithm.

Can now examine the impact of the photon reconstruction algorithm on jet energy reconstruction.

Jet Energy Performance





ILDoo, E_z (= 2 * E_j)	91GeV	200GeV	360GeV	500GeV
Standard Pandora, $rms_{90}(E_j) / E_j$	3.64 ± 0.05	2.93 ± 0.04	3.00 ± 0.04	3.19 ± 0.04
Photon Clustering, $rms_{90}(E_j) / E_j$	**3.69 ± 0.05**	**2.88 ± 0.04**	**2.91 ± 0.04**	**3.03 ± 0.05**

Best ever Pandora performance for ILDoo

Summary

Development of Pandora continues at a rather rapid pace:

- Pandora has been restructured so that it can be easily applied to a number of different pattern-recognition problems. The reconstruction can be customised in a number of ways.
- A forced clustering algorithm has been developed to carefully manage the transition from particle flow to "energy flow" calorimetry when there is no other alternative.

- Monitoring of the energy changes that occur during Pandora reclustering algorithms has been implemented and the data provided are under investigation.
- The efficiency of the FineGranularityContent algorithms and framework has been improved, reducing the CPU time required to reconstruct events in ILD with overlaid background.

- A photon reconstruction algorithm has been implemented, which aims to fully reconstruct and tag photons and remove them from the subsequent Pandora reconstruction.
- The photon reconstruction algorithm proves beneficial at high energies and the current jet energy performance figures are the best ever obtained for the ILD00 detector model.