
BeamCal electron reconstruction

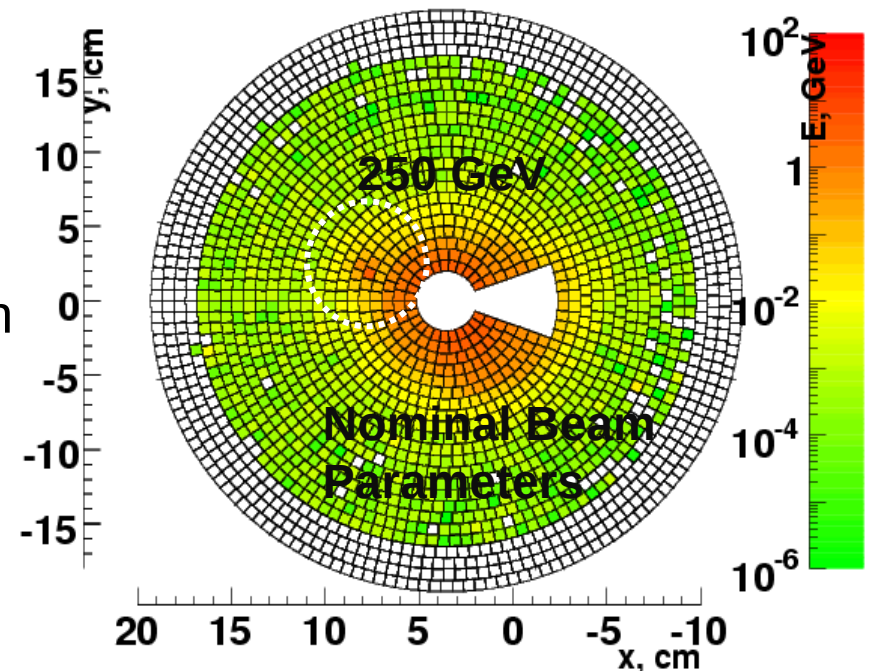
Aura Rosca
22th August 2011

Short reminder: Cluster reconstruction algorithm in BeamCal

(Developed by Olga N. and Wolfgang L.)

1. Background subtraction procedure

- Calculate average and rms of the energy deposition of the background in each pad of the BeamCal, from 10 BX;
- Superimpose 1 BX background + 1 high energy electron;
- Subtract the value of the background average from the superposition;

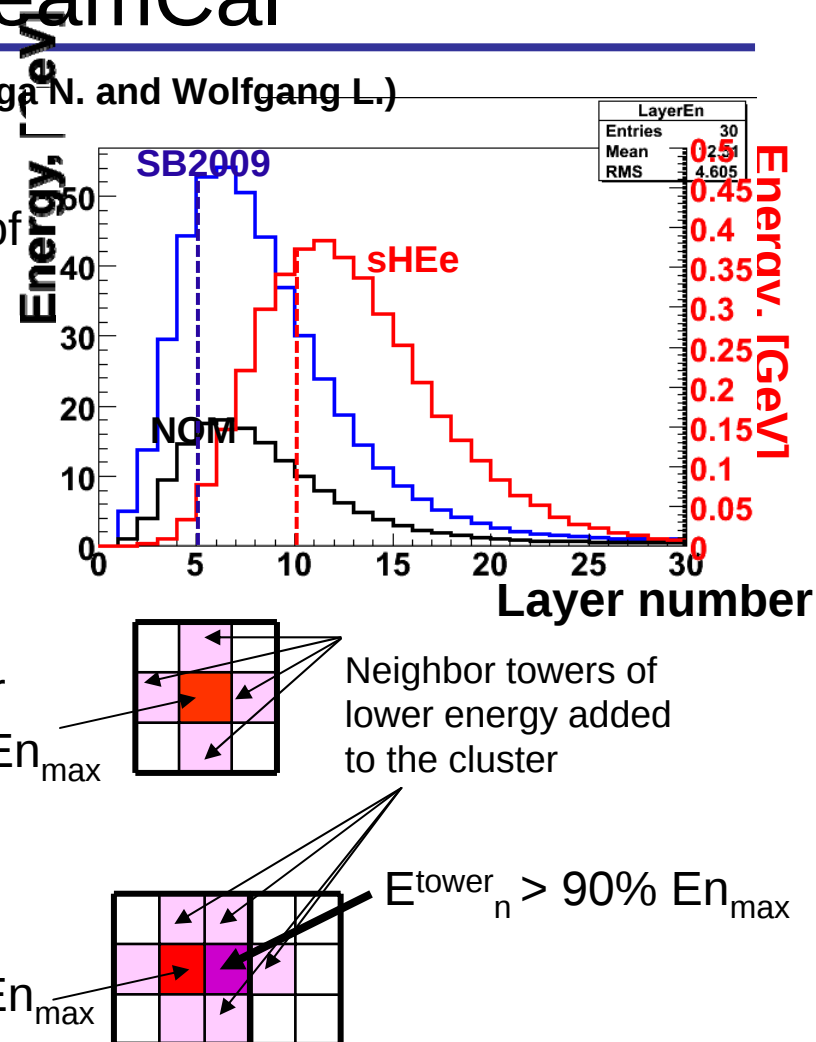


Short reminder: Cluster reconstruction algorithm in BeamCal

2. Cluster search

- Identify towers after the 5-th layer as chains of 10 consecutive fired pads;
- Search for the tower with the maximum deposited energy;
- Add neighbor towers, in a 3×3 matrix around the tower with the highest energy;
- If such a neighbor tower has an energy larger than 90% of the energy of the central tower, $E_{n_{\max}}$, add this tower neighbors as well;
- Output: energy of the cluster, corresponding ring number, pad number.
- SW used for efficiency studies in FW BeamCal

(Developed by Olga N. and Wolfgang L.)



Standalone SW reconstructed clusters

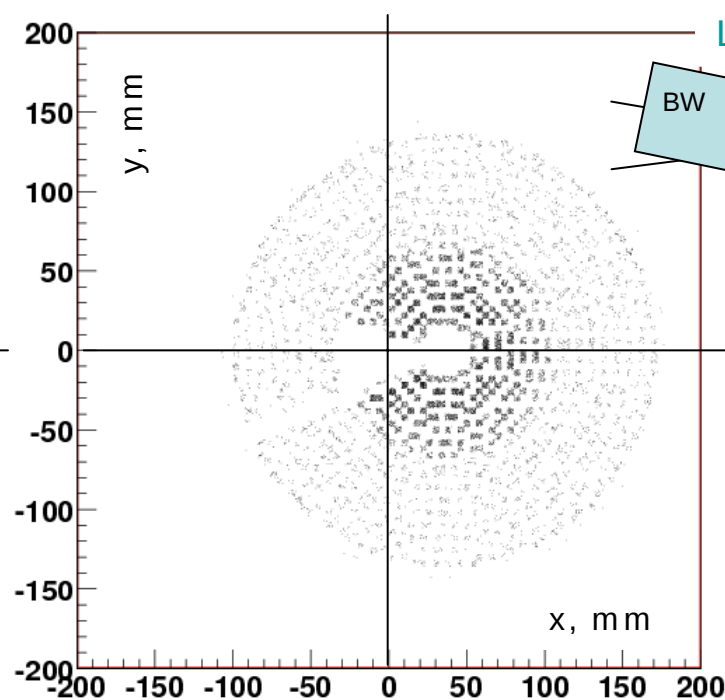
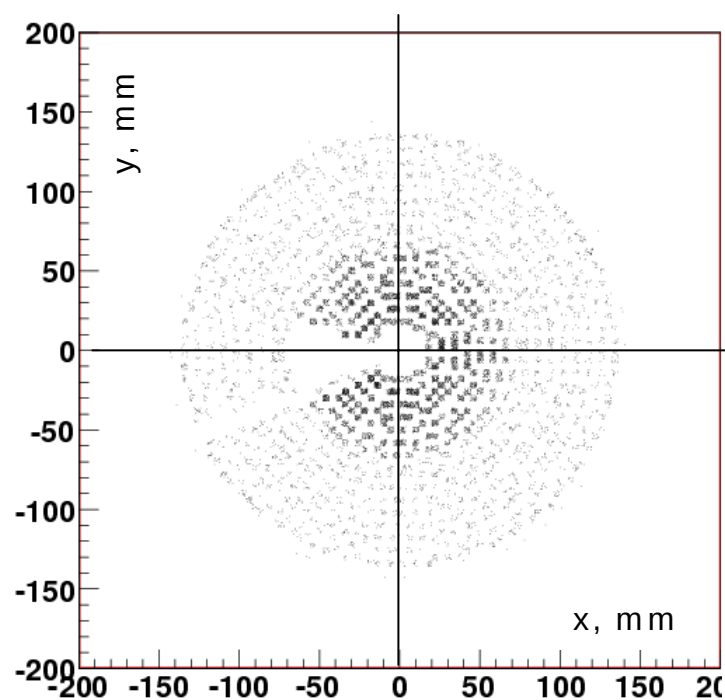
X-Y Coordinates of Clusters in the FW BeamCal

BeamCal Local CS (LCS)

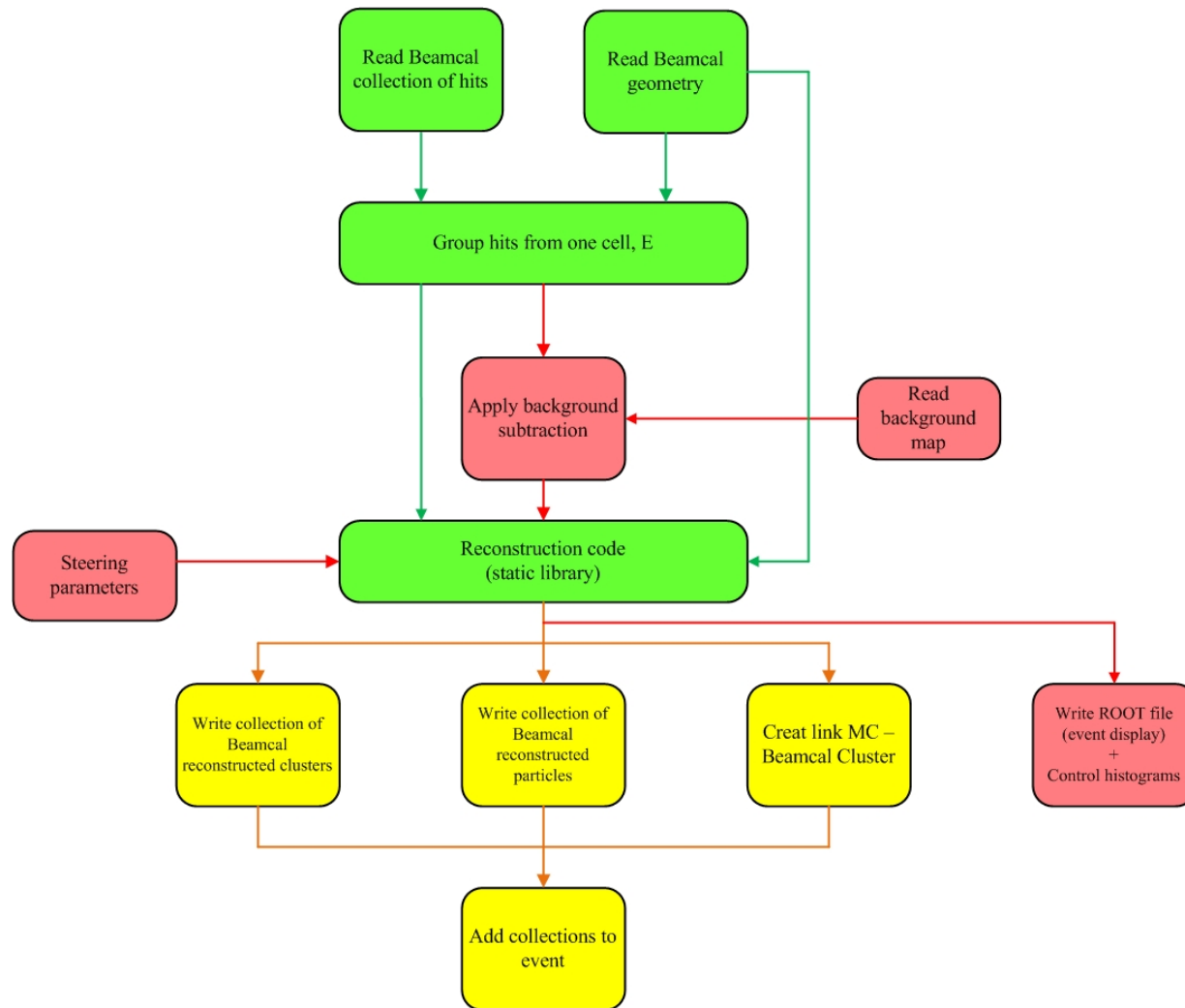
Global ILD CS (GCS)

Clusters X-Y Coord

Clusters X-Y Coord



Implementation into central reconstruction



Redesign of the reconstruction code

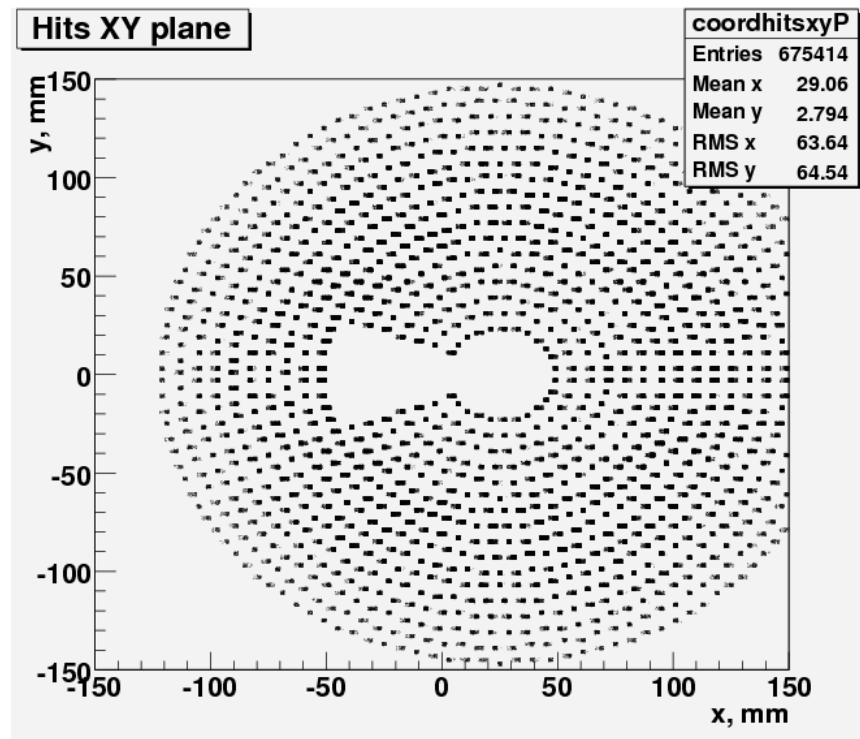
- Develop a BeamCal reconstruction class, BCalReconstruction:
 - Destructor: ~BCalReconstruction()
 - Functions:
 - RecCorr GetReconstrCoordinates (int number_layers, int number_rings, int number_pads[], CellType ***info_detector);
 - typedef struct {
 int side; // 0,1,-1 -> no, FW, BW reconstruction
 double RecEne, ErrEne, CoordX, CoordY, CoordZ, RecRad, RecPhi;
} RecCorr;
 - Protected member functions:
 - vector SearchTowers (int the_Chains[maxrings][maxphis][maxlayers]);
 - RecCorr SearchClustersFW (CellType ***info_detector);
 - RecCorr SearchClustersBW (CellType ***info_detector);
 - double GetEnergyCalib (double energy);
 - double GetEnergyErr (int ring, int pad);
 - double GetCoordRotX (int ring, int pad, float IP, float angle);
 - double GetCoordY (int ring, int pad);
 - double GetCoordRotZ (int ring, int pad, float IP, float angle);
 - void Free2DArray (int **p2DArray), void Free3DArray (CellType ***p3DArray);

New Marlin processor

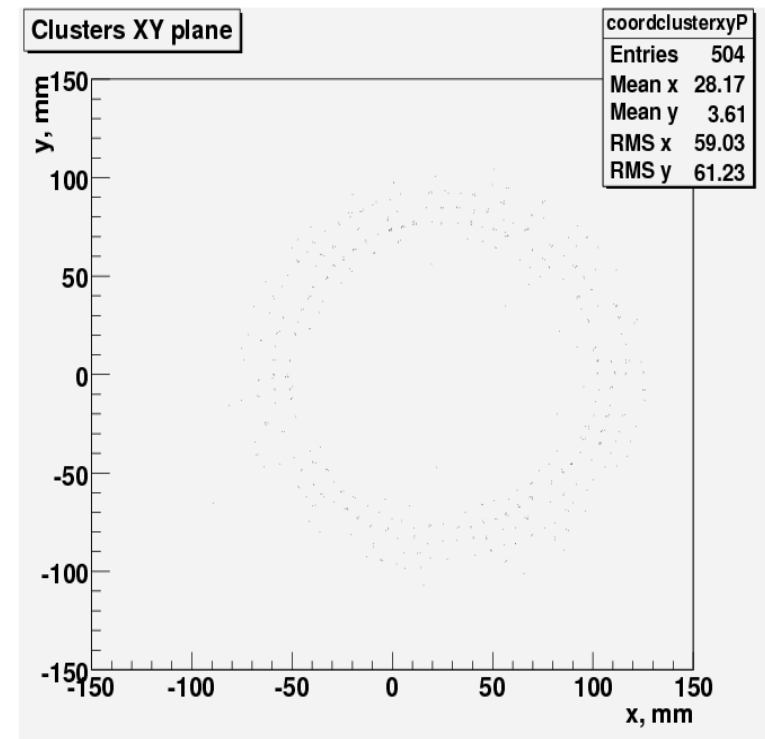
- Develop a Marlin processor, BCalReco, to perform the following tasks:
 - read the BeamCal collection of hits and put the information into a 3D dynamic array of structures, CellType ***info_detector:
 - `typedef struct {
 double sRin,sRout,sZstart,sZend,sSphi,sDphi,sEdepNeg,sEdepPos;
 int sPos[3];
} CellType;`
 - call the reconstruction code (linked to the processor as a static library, libbcreco.a):
 - `bcal_reco = new BCalReconstruction();`
 - `bcal_electron = bcal_reco->GetReconstrCoordinates(nLayers,nRings,nbPhis,cells);`
 - output the relevant collections (clusters, reconstructed particles) – **work in progress.**

Hits and Clusters in BeamCal (Bhabha events)

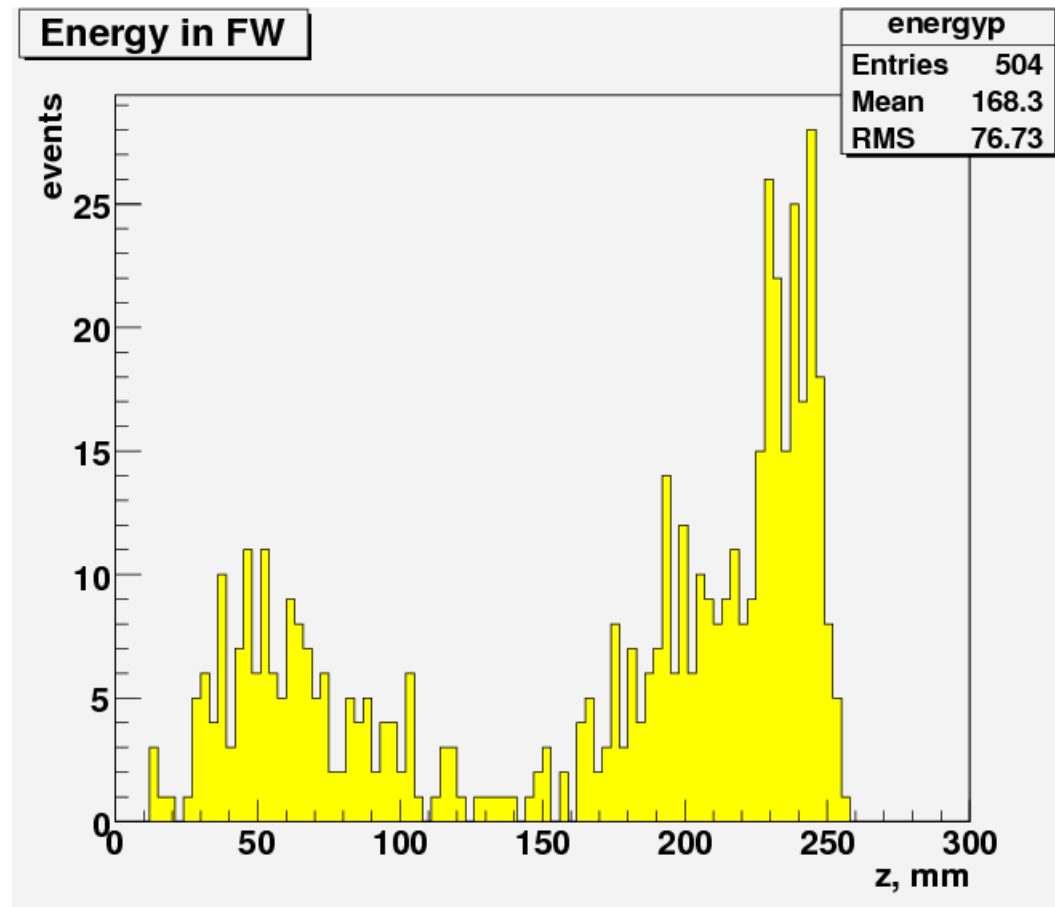
X-Y Coordinates of hits in BeamCal



X-Y Coordinates of clusters in BeamCal



Electrons in the BeamCal



Next steps to be done

- Finalize the conversion of the output into the relevant collections, inside the processor (until August 26th).
- Write a new processor to read and test the Beamcal collections (one week).
- Apply the background subtraction procedure (1/2 week).
- Currently the reconstruction code finds one cluster the FW BeamCal and one in the BW BeamCal respectively; the case when more than one single cluster in each BeamCal is found should be implemented next (2-3 weeks work).
- Generate documentation from the source code (LC Note ?).

Summary

- Electron reconstruction algorithm for BeamCal was redesigned to be included into the central reconstruction software
- New Marlin processor exists, BCalReco
- Several issues to be addressed in the near future: simulation of the background, production of the relevant output collections.