

RecoBase package discussion

(Shower.h)

LArSoft Meeting
02 Sep 2010

Biagio Rossi

Albert Einstein Center for Fundamental Physics
Laboratory for High Energy Physics
University of Bern

ShowerFinder

LArSoft has a new package called
ShowerFinder

its goal is to check for electromagnetic showers in the 2D views
and perform the spatial and the calorimetric reconstruction in 3D

The output is stored in to an object of **Shower**,
that is a sub-class of **Prong** (package **RecoBase**).

Shower.h

```
double fVertex[4]; // Position of the Shower Vertex in wire, ticks
                // Vertex[0] = Wire Induction
                // Vertex[1]= Wire Collection
                // Vertex[2]=Time Induction
                // Vertex[3]=Time Collection

double fLastWire[2];
        // fLastWire[0] = Wire of Induction plane where the shower ends
        // fLastWire[1] = Wire of Collection plane where the shower ends

double fLastTime[2];
        // fLastTime[0] = Tick of Induction plane where the shower ends
        // fLastTime[1] = Tick of Induction plane where the shower ends

double fDir2D[4]; // 2D Slopes and Intercepts of the Shower axis in (wire, ticks)
                // Dir3DV[0] = Intercept of the shower axis for Induction plane
                // Dir3DV[1] = Slope of the shower axis for Induction plane
                // Dir3DV[0] = Intercept of the shower axis for Collection plane
                // Dir3DV[1] = Slope of the shower axis for Collection plane

double fDir3D[4]; // 2D Slopes and Intercepts of the Shower in cm
                // Dir3DV[0-1] = Induction, Dir3D[2-3]=Collection
```

only 2D info
are stored

Shower.h and Prong.h

In Shower.h is still missing:

- Vectors for Longitudinal and Transverse Energy histos of the shower → Shower.h
 - Total MIPs released for each plane
 - Pitch for each plane
- Common to tracks and showers } Prong.h

Prong.h

```
public:
void
double      SetPitch(double pitch, geo::View_t view);
double      Pitch(geo::View_t view) const;

private:
double  fPitch[3];    ///length of the prong segment seen by a wire (for each view)
```

Should we include the total energy released in **Prong.h** ?

Calibration factors (MIPs to Energy) in **Prong** or somewhere else ?

3D directions reco

Is **Prong.h** the place to store the 3D info of tracks and showers ?

Prong.h

public:

```
void          SetDirection(double *dcosStart, double *dcosEnd);  
void          Direction(double *dcosStart, double *dcosEnd) const;
```

private:

```
double  fDCosStart[3];  ///direction cosines at start of prong  
double  fDCosEnd[3];   ///direction cosines at end of prong
```

Should we add the StartPoint and the EndPoint of the tracks and showers reconstructed in 3D?

Can we use the same function for the reconstruction of the 3D directions and pitches of tracks and showers ?