BigBite Analysis

Transversity BB Shower class

Matthew Posik

[1] Temple University
Philadelphia, PA 19122

10/14/2010

Outline

1. BB Transversity Shower Class
2. Summary
3. What's Next

Shower Cluster Reconstruction

- Shower reconstruction is used to determine energy and position of a particle that generates electromagnetic shower in the detector
- A cluster is a group of continuous shower blocks where energy loss due to em shower is detected
- Maximum block in the cluster is know as the central block
- There can be more than one cluster per event in the shower and preshower
- Transversity cluster reconstruction algorithm identifies all clusters and saves them for analysis

Cluster Reconstruction Algorithm Overview (1)

- Search for block with largest energy in shower (central block)
- Sum over 8 blocks surrounding the central block to get cluster energy in the shower
- Repeat procedure to find other clusters in order of decreasing energy and saved
- Corresponding to maximum cluster in the shower find matching preshower cluster within a certain distance. If not found then cluster is invalid and next shower cluster is considered.

Cluster Reconstruction Algorithm Overview (2)

- If found, then sum over 6 blocks in the preshower. This is the cluster energy in the preshower
- For further cluster validity, track projection on to shower X,Y coordinates is matched with reconstructed position of the cluster within a certain distance. If no match is found then above procedure is repeated until all conditions are met.
- If all conditions are satisfied, then the energy and position is saved for later analysis.
- Photon tracks are constructed in a similar fashion as above, but they do not leave a track in the MWDCs, the cluster position can not be matched to the reconstructed track

Shower variables (1)

- With the new shower class, the energy clusters are indexed on track number

```
BB.ts.ps_e[], BB.ts.sh_e[] indexed on track number
```

- When using the new energy clusters, the shower cluster array is different from the tracking array. To account for this another variable is used to match the cluster array to the tracking array

```
BB.ts.sh_flag_tr (shower track number)
Example:
T->Draw("BB.ts.ps_e[]","BB.tr.p[BB.ts.sh_flag_tr]>0.6")
```

Shower variables (2)

- There are also cluster energy corrections

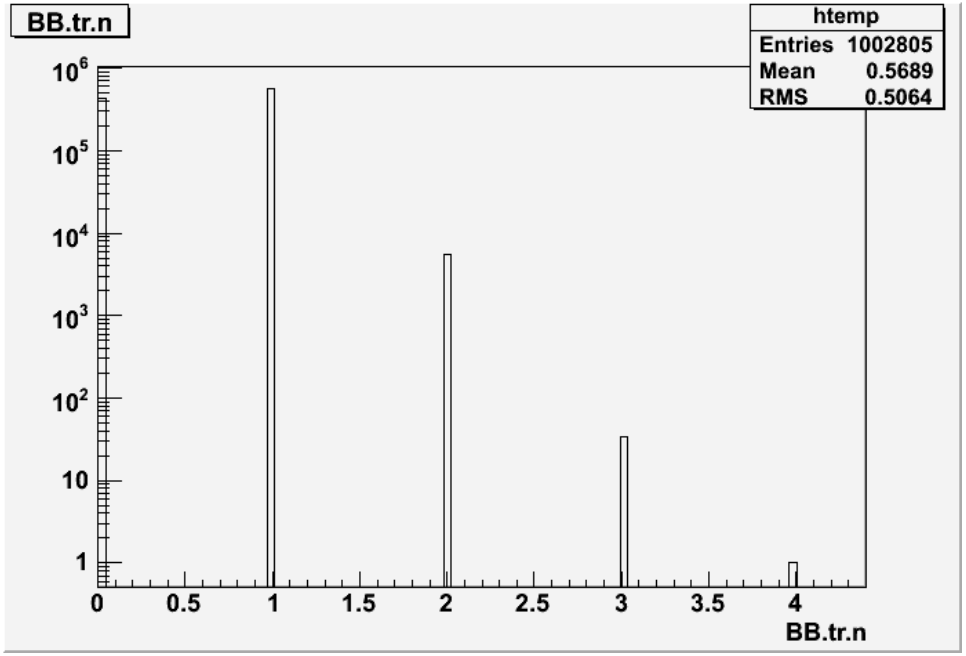
```
BB.ts.ps_e_c[], BB.ts.sh_e_c[] indexed on track number
```

- These were used to correct for color change in the Shower/Preshower Pb glass
- Corrections were made to the slope of a linear fit to Preshower peak vs accumulated charge
- Similar correction is done to the Shower by fitting second order polynomial to the E/p vs accumulated charge
- The results of the corrections are found in the BB.ts.dat file
- The track free variables, used for photon cluster reconstruction are:

```
BB.ts.clus_ps_e[], BB.ts.clus_sh_e[]
BB.ts.clus_ps_e_c[],BB.ts.clus_sh_e_c[]
BB.ts.clus_x[], BB.ts.clus_y[]
```

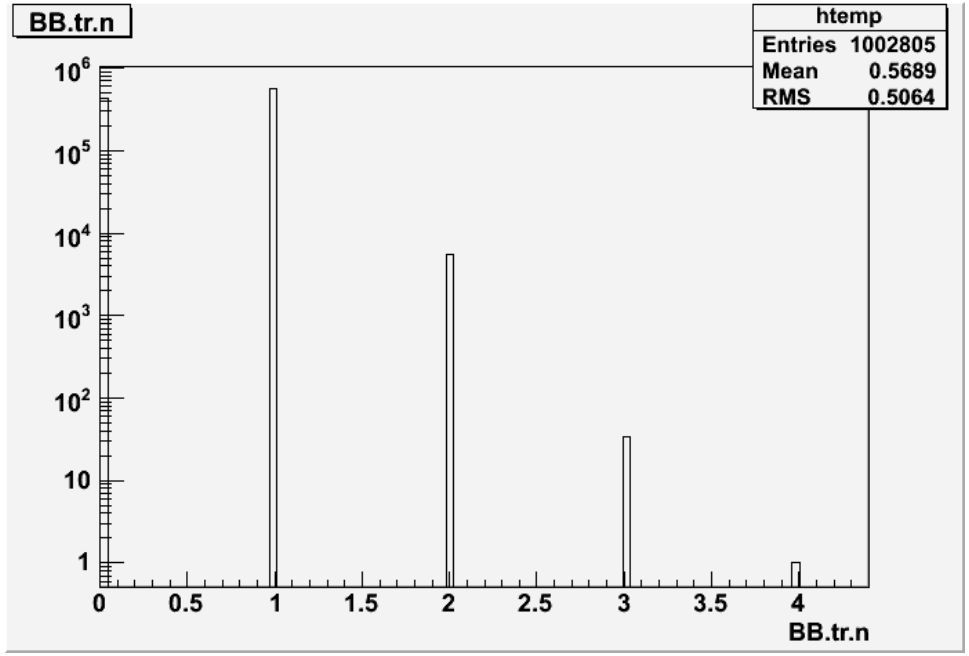
d2n/Transversity Total Track Comparison

d2n Shower Class



Events: 1002805

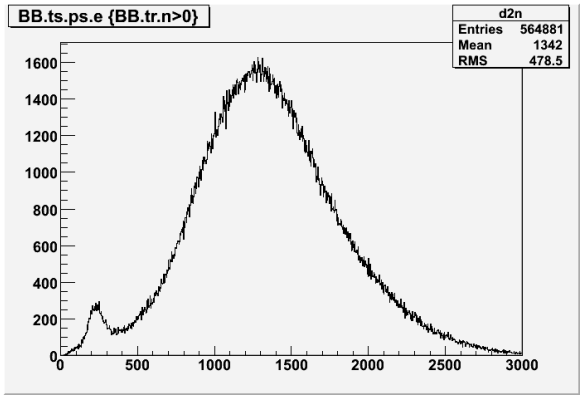
Transversity Shower Class



Events: 1002805

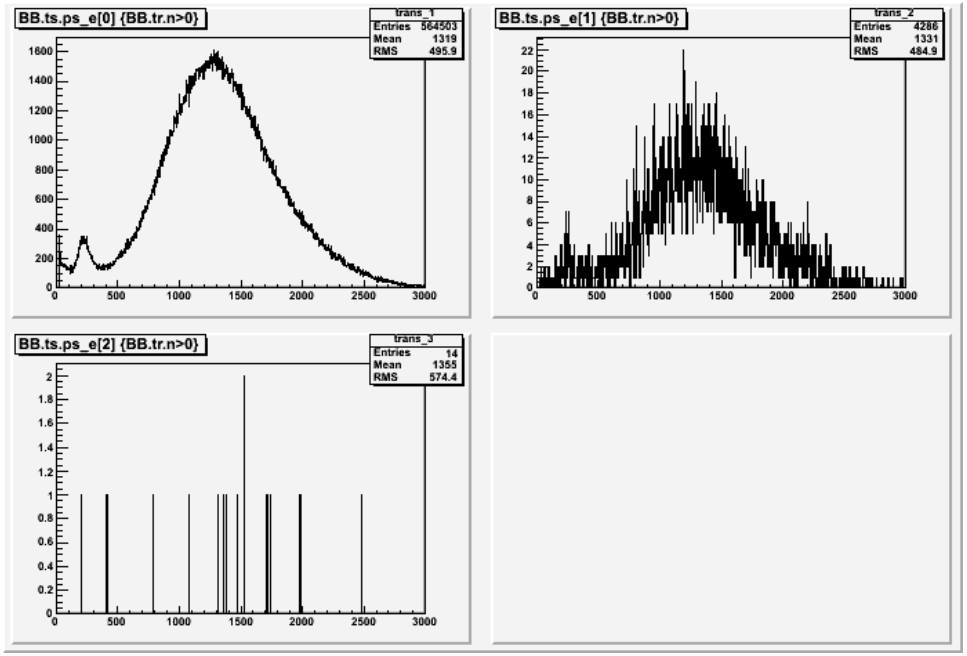
d2n/Transversity PreShower Comparison

d2n Shower Class



Events: 564,881

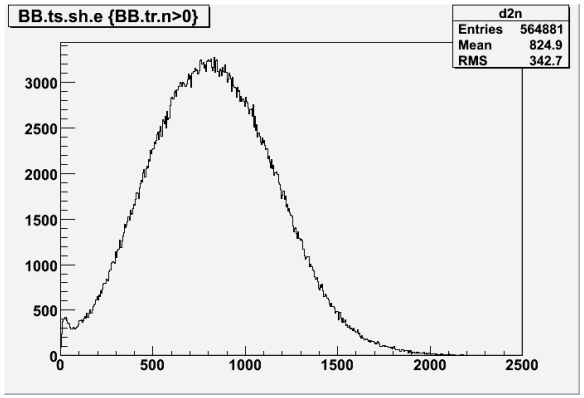
Transversity Shower Class



1 track events: 564,503
2 track events: 4,286
3 track events: 14

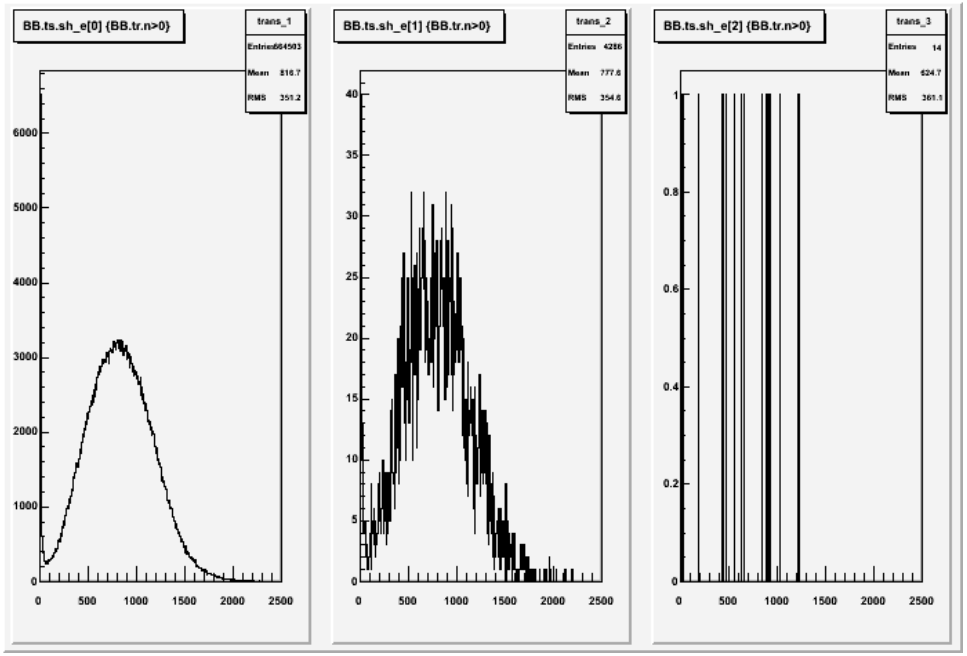
d2n/Transversity Shower Comparison

d2n Shower Class



Events: 564,881

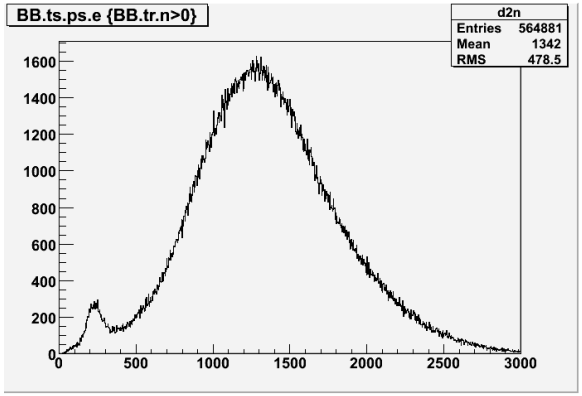
Transversity Shower Class



1 track events: 564,503
2 track events: 4,286
3 track events: 14

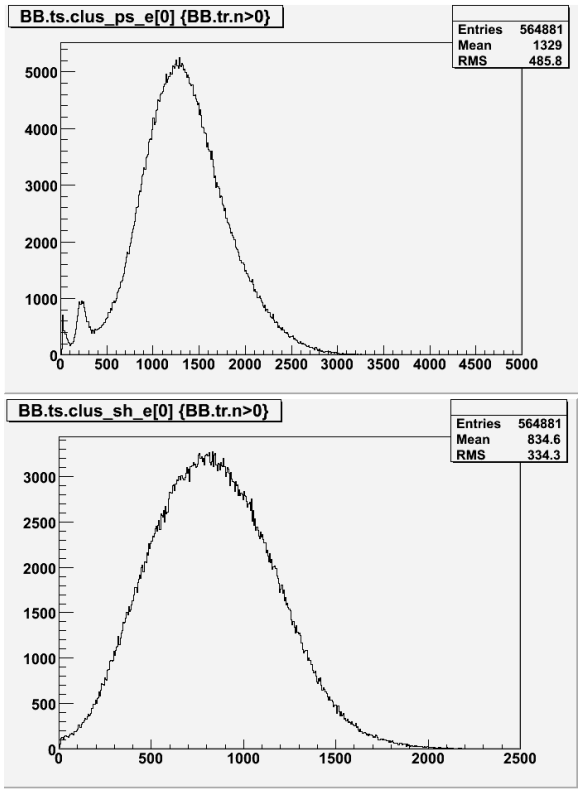
d2n/Transversity Cluster Comparison

d2n Shower Class



Events: 564,881

Transversity Shower Class



Events: 564,881

Summary

Transversity shower class...

- Matches shower and preshower clusters
- Compares to reconstructed tracks to cluster position
- Considers multiple tracks

For Next week

- Look at shower calibration
- Begin finer treatment of BB production run list
- Look at defining BB Čerenkov mirror cuts using Čerenkov cones