
Subject: Neighbour-finding at large angles
Posted by [mcharles](#) on Mon, 26 Nov 2007 18:36:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

One routine that's used quite a lot in calorimeter clustering is `getNeighbourIDs()`. For example, the nearest-neighbour clustering routine in `org.lcsim.recon.cluster.nn.NearestNeighborCluster` uses a snippet like:

```
Quote:IDDecoder decoder = hit.getIDDecoder();  
CalorimeterHit c = hits.get(i);  
decoder.setID(c.getCellID());  
long[] neighbors = decoder.getNeighbourIDs(dLayer, dU, dV);
```

For a non-projective detector the neighbour-finding across layers is tricky. Guilherme implemented these routines in `org.lcsim.geometry.segmentation` a while back and the code works pretty nicely. There's one case where I'm not quite sure what the right thing to do is, though. (Maybe Guilherme has thought this through already and can explain it?)

Right now, finding neighbours in layer *m* given a hit in layer *n* in the barrel looks roughly like this:

- 1) Find theta, phi of original hit in layer *n*
- 2) Get the radius *r_m* of layer *m*
- 3) Compute the point in 3D with (*r_m*, theta, phi), i.e. project from layer *n* to layer *m*.
- 4) Find the hit containing that point.
- 5) Find that hit's grid neighbours in layer *m*.

I was looking at a case of a shower in the HCAL with $\cos(\theta) \gg 0$ and was initially confused when I saw hits that were not flagged as neighbours despite being close (e.g. 3D separation 30mm) even though there were neighbours with larger 3D separations (e.g. 40mm). But after thinking about step 3 a bit more, I realized this was working as intended: the hits were spatially close but not so close in (theta, phi).

I'm not actually sure what the ideal thing to do here is. The projective approach is simple and works well in a lot of cases (most primaries are more or less projective), and the places where it's non-intuitive are usually when clustering very tightly (e.g. 1,1,1). On the other hand, secondaries lose a lot of the initial direction, so projective clustering isn't really the best for them -- I can see this biting from time to time.

Any thoughts?

Mat.
