Subject: Bug in PndEmcMapper class: neighbour lists of PndEmcTwoCoordIndex wrong
Posted by Elwin Dijck on Mon, 08 Jun 2009 20:19:12 GMT
View Forum Message <> Reply to Message

Hello all,

I'm Elwin Dijck, working with PandaRoot for my bachelor's thesis at the KVI in Groningen. I think I found a bug in the PndEmcMapper class, so I'll describe the problem and how it could be fixed.

I wanted to use the neighbour sets of the PndEmcTwoCoordIndex objects stored in the PndEmcMapper class, but found that some of the PndEmcTwoCoordIndex pointers in the set returned by PndEmcTwoCoordIndex::GetNeighbours() are not actually neighbours (though they are close), as the following code would show.

Code
(Note: this needs to be compiled, CINT doesn't seem to like STL containers. The necessary libraries need to be loaded as well. Also it needs a ROOT file with the EMC geometry and the correct mapper version, adjust where needed.)

```
#include <iostream>
#include <map>

#include <TFile.h>

#include <PndEmcMapper.h>
#include <PndEmcNeighbourStore.h>
#include <PndEmcTwoCoordIndex.h>
#include <PndEmcStructure.h>

using namespace std;

void neighbour_test()
{
   TFile file("sim_emc.root");

   PndEmcMapper *mapper = PndEmcMapper::Instance(1);

   cout << "total: " << mapper->GetTciMap().size() << " tci's" << endl;

   map<Int_t, Int_t> errors;  // Indexed by module.

   for (map<Int_t, PndEmcTwoCoordIndex *>::const_iterator
      tci = mapper->GetTciMap().begin(); tci != mapper->GetTciMap().end();
      ++tci)
   {
      // (Need to copy the neighbours as they're returned by value. Return by
      // reference might be better here?)
      PndEmcNeighbourStore neighbours = tci->second->GetNeighbours();

      for (PndEmcNeighbourStore::const_iterator
```

```
        neighbour = neighbours.begin(); neighbour != neighbours.end();
        ++neighbour)
      {
        // Check if the supposed neighbour is actually a neighbour.
        if (!tci->second->IsNeighbour(*neighbour))
          ++errors[tci->first / 100000000];
      }
  }

  for (map<Int_t, Int_t>::const_iterator it = errors.begin();
      it != errors.end(); ++it)
  {
      cout << it->second << " errors in module " << it->first << endl;
  }
}
```

I get the following output for MapperVersion 1 (note that the MapperVersion shouldn't actually matter, the problem only affects the barrel part of the EMC).

total: 19633 tci's
47744 errors in module 1
32064 errors in module 2

So I looked where these neighbour sets are initialized, which is in the constructor of the PndEmcMapper singleton class. What happens there is basically as follows:

First all crystals/coordinates are generated: for every valid detector ID, the corresponding two-coordinate index is generated and a PndEmcTwoCoordIndex object is stored in a std::map, indexed by the detector ID.
Then, for every PndEmcTwoCoordIndex in the map, the neighbours are looked up: for each of the eight possible two-coordinate indices of the neighbours, the private function PndEmcMapper::GetDetId() is called to find the detector ID corresponding to the two-coordinate index and this is used to find the neighbouring PndEmcTwoCoordIndex's in the map.
The problem is that the mapping from a detector ID (module, row, copy, crystal) to a two-coordinate index (theta, phi) and vice versa seems to be done inconsistently for the barrel part of the EMC: at line 76 of emc/EmcTools/PndEmcMapper.cxx is the following mapping, which is used during the first step of generating all PndEmcTwoCoordIndex objects:
iPhi=(11-crystal)+(copy-1)*10;(The same can be found on line 90 (other module of the barrel), line 183 and 197 (MapperVersion 2), line 351 and 365 (MapperVersion 6), line 456 and 470 (MapperVersion 7))

Later in PndEmcMapper::GetDetId() on line 836 and 843, the following conversion back can be found for module 2, which is used during the filling of the neighbour sets of the PndEmcTwoCoordIndex objects:
copy=(iPhi-1)/10+1;
crystal=(iPhi-1)%10+1;
The problem is that in the first case the direction of counting crystal and iPhi are opposite, while in the second case the crystal and iPhi are counted in the same direction. I.e. I think the last line should be crystal = 10 - (iPhi - 1) % 10; to be consistent with the code in the constructor.

This means that the wrong PndEmcTwoCoordIndex's are found as neighbours, but because just the crystal number is inverted which is the innermost index, the found crystals will never be off by more than 10 and only in phi direction. Also, this problem only exists for modules 1 and 2 of the barrel, the indexing of the endcaps is ok as far as I know (haven't checked all MapperVersions).

I don't know which way of indexing the EMC crystals was the intended one, but in both cases it should be easy to fix in either PndEmcMapper::PndEmcMapper() or in PndEmcMapper::GetDetId().

Could someone look into this?

Best regards,
Elwin

## File Attachments

1) neighbour_test.C, downloaded 420 times