

LArSoft - Bug #24706

Opposite channel numbering sorting for OpDets

08/03/2020 06:42 PM - Iker de Icaza Astiz

Status:	Feedback	Start date:	08/03/2020
Priority:	Low	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour
Occurs In:		Co-Assignees:	
Experiment:	-		

Description

The convention used to assign channels numbers to OpDets is (mostly) opposite to any other.

Function in question is `sortorderOpDets()`

<https://github.com/LArSoft/larcoreal/blob/develop/larcoreal/Geometry/GeoObjectSorter.cxx>

the block:

```
if (xyz1[2] != xyz2[2])
    return xyz1[2] > xyz2[2];
else if (xyz1[1] != xyz2[1])
    return xyz1[1] > xyz2[1];
else
    return xyz1[0] > xyz2[0];
```

means that **the channel number increases with decreasing Z, Y or X; in that order of priority**

To better coincide with the other sorter algorithms and a more natural convention it should be:

```
if (xyz1[2] != xyz2[2])
    return xyz1[2] < xyz2[2];
else if (xyz1[1] != xyz2[1])
    return xyz1[1] < xyz2[1];
else
    return xyz1[0] < xyz2[0];
```

Note that I changed the inequality sign.

With that change it will become: **the channel number increases with increasing Z, Y or X; in that order of priority**

Obviously I understand the implications of making this change, and that it will break more things than the potential issues it might prevent. I'm mostly highlighting this to make this discrepancy evident for others.

History

#1 - 08/03/2020 11:32 PM - Lynn Garren

- Status changed from New to Feedback

Iker, would you like to present this possible change and its implications at a coordination meeting?

#2 - 08/03/2020 11:32 PM - Gianluca Petrillo

One additional information: the channel number is assigned by LArSoft in contiguous blocks, first to the first cryostat (C:0, then to the second C:1 and so on). Within the cryostat, channels are assigned in the same order as the optical detectors are sorted within the cryostat, which is the reason why `geo::GeoObjectSorter::SortOpDets()` affects also the channel mapping.

Also note that both the original and the suggested sorters have a strange effect on cathode-in-the-middle TPC: optical detectors on one drift volume are assigned only odd channel numbers, while the ones on the opposite drift volume will be assigned only even ones. ICARUS has moved away from that (at the cost of unspeakable pain).

I add as an annotation that I am surprised that such a rounding-error-sensitive algorithm has not created weird jumps.