

## LArSoft - Feature #9264

### Support reconstruction of objects from channel/time space

06/22/2015 10:20 AM - Gianluca Petrillo

<b>Status:</b>	Closed	<b>Start date:</b>	06/22/2015
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Gianluca Petrillo	<b>% Done:</b>	100%
<b>Category:</b>	Geometry	<b>Estimated time:</b>	40.00 hours
<b>Target version:</b>		<b>Spent time:</b>	40.00 hours
<b>Experiment:</b>	LArSoft	<b>Co-Assignees:</b>	
<b>Description</b>			
<p>In detectors where TPC readout channels span multiple TPCs or appear in more than one straight wire, simple reconstruction in the physical wire/time space bound to a single TPC yields to potentially ambiguous results, where the location of the reconstructed objects can't be uniquely determined. This is the case for the DUNE 35t prototype and possibly for the final DUNE 10kt detector.</p> <p>The reconstruction in the virtual space where channel numbers are used in place of wire numbers yields instead to unambiguous results. The objects reconstructed in such a way will likely require a conversion to the physical space to complete the physics analysis, and this conversion has the same ambiguity potential as the one described above. Nevertheless, delaying the "disambiguation" phase may allow to collect more detailed information to help resolving the ambiguity.</p> <p>LArSoft should provides ways to facilitate the reconstruction in this virtual space.</p> <p>This request originates from David Adams (DUNE collaboration). A model satisfying his specific needs can be found in the <a href="#">GeoHelper class</a> in his <a href="#">personal GitHub repository</a></p>			
<b>Related issues:</b>			
Blocks dunetpc - Feature #12527: Add option to keep neighboring channels in z...		<b>Closed</b>	<b>05/05/2016</b>
Blocks dunetpc - Feature #13877: Provide dataprep module that handles one APA...		<b>Closed</b>	<b>09/19/2016</b>

#### Associated revisions

##### Revision 7fcc2fd5 - 10/12/2016 05:27 PM - Gianluca Petrillo

Added readout-level geometry IDs.

This is implementation of LArSoft issue #9264.

This code has been moved from a namesake feature branch in larcore.

#### History

##### #1 - 06/22/2015 10:33 AM - Gianluca Petrillo

- Assignee set to Gianluca Petrillo

- Estimated time set to 40.00 h

The [LArSoft Architecture Review committee convened on June Wednesday 17th, 2015](#) to discuss the details of the issue.

The outcome is summarized in the [meeting minutes](#).

A short summary follows:

- introduce ID classes for the new "readout" entities: APA (TPCs sharing readout channels) and ROP
- the new entities will not have any "geometry" counterpart (e.g., no geo::ROPGeo class), and their role is completely described in the code: no modification to GDML detector description is necessary
- extend the geometry interface to allow for mapping between these and the geometry IDs
- extend the reconstructed objects to allow for the identification in terms of the new readout entities

##### #2 - 06/22/2015 10:33 AM - Gianluca Petrillo

- Status changed from New to Assigned

##### #3 - 06/22/2015 11:00 AM - Gianluca Petrillo

- % Done changed from 0 to 10

Work has started on feature branch `feature/gp_WrappedGeometry` (so far only in `larcore`, in the future also in `lbnecode`).

Added ID classes for the new entities.

**#4 - 08/13/2015 02:09 PM - Gianluca Petrillo**

- Status changed from *Assigned* to *Feedback*

- % Done changed from 10 to 90

Work has been delivered in feature branch `feature/gp_WrappedGeometry` for the experiments (DUNE in particular) to test.

**#5 - 05/05/2016 08:59 AM - David Adams**

Gianluca:

Sorry for overlooking this. I looked at the branch and it has what I need. There is an iterator over ROPs, means to get the begin and end of that iteration, the first channel in a ROP and the number of channels in a ROP.

I will also need to know if a ROP is circular, i.e. if the first channel is "next" to the last. I didn't see that in the geometry but the class is very complex and I could have overlooked it. I suppose I can get the information for the first channel and assume it is the same for all and so this is not a showstopper.

Please go ahead and put this mod into the head of `larcore` and we will start developing against it.

Thank you.

**#6 - 05/05/2016 11:09 AM - David Adams**

- Blocks Feature #12527: Add option to keep neighboring channels in zero suppression added

**#7 - 06/17/2016 04:14 PM - Gianluca Petrillo**

- Status changed from *Feedback* to *Assigned*

- % Done changed from 90 to 80

I don't know how hard it is to update after one year, so I can't give an estimate of the delivery time.

About your question: the ROP by itself is not circular, but I think I know what you mean, that is whether iterating through the wires in a wire plane (not a ROP), I go after the last channel of the ROP and land back to the first one.

That information is not present.

I suggest that the feature be added after the current changes are integrated.

We will also need a detailed description of the meaning of this information.

**#8 - 07/07/2016 03:02 PM - Gianluca Petrillo**

- % Done changed from 80 to 90

I have updated the branch `feature/gp_WrappedGeometry` in `larcore` to `develop` (based on `v05_14_01`).

I have also updated the one in `dunetpc`, **but** more work is needed.

The changes to channel mapping require a new set of methods, and each channel mapping object must define them.

I had implemented them for the existing channel mappings one year ago, based on David's implementation in his own code.

Since then, a new mapping has been added, apparently for dual phase TPC. That mapping must be updated.

Given the nature of dual phase TPC, I think that by necessity the detector will not have matching TPCs, and I hope it does not even have wrapped wires. In that case, the mapping is trivial and can be "copied", *mutata mutandis*, from the standard mapping (`larcore/Geometry/ChannelMapStandardAlg.h` and `.cxx`).

**#9 - 07/12/2016 05:58 AM - David Adams**

Gianluca:

Can we implement this so that no changes are required for detectors without wire wrapping?

If not, it sounds like you are asking for changes in the DUNE dual-phase geometry. I have added Tom and Tingjun who can tell us who (me??) should make changes to that code.

We have many other geometries, e.g. 35t and options for FD and SP protoDUNE. Are those all covered?

Can we move your changes to the `larcore develop` branch, i.e. can we release them as they are now. I think it OK if the ROP-specific methods fail or give bad answers where mods have not yet been made.

Thanks.

**#10 - 07/12/2016 04:58 PM - Gianluca Petrillo**

Note: the following is my own view, that may be overruled by the project lead.

Let me clarify one point first: this feature is becoming part of LArSoft and it will be supported as such.

The choice of not providing a fallback implementation was deliberate, because that fallback implementation could be wrong depending on the detector.

I don't want to leave around implementations that are silently wrong.

What *is* acceptable for LArSoft is that an Experiment decides, by their own will, not to support a feature. That also means that the Experiment accepts that they won't benefit from LArSoft content which depends on that feature, that LArSoft may add in the future.

For this specific case, will there be any such a content in the future? I very much hope so (in particular, reconstruction in readout space may be viable). Will it be relevant for ProtoDUNE Dual Phase? probably not.

Suggestion: you (or I) can replace the static\_assert that constitute the current implementation of the new methods in Dual Phase channel mapping, with:

```
throw cet::exception("ChannelMapCRMAlg") << __func__ << " not implemented for ChannelMapCRMAlg\n";
```

First, though, the authors of that mapping should be given the choice to implement the functions for real, or to bless the change above.

**#11 - 09/20/2016 07:14 AM - David Adams**

Gianluca:

Is this feature in the head of larsoft?

da

**#12 - 09/20/2016 07:15 AM - David Adams**

- *Blocks Feature #13877: Provide dataprep module that handles one APA at a time added*

**#13 - 10/05/2016 01:31 PM - Gianluca Petrillo**

I missed one question from your comment:

We have many other geometries, e.g. 35t and options for FD and SP protoDUNE. Are those all covered?

I copied the mappings from your implementation. I think that covered everything except protoDUNE's.

Incidentally, I need to port the branches to LArSoft 6.

**#14 - 10/13/2016 06:57 AM - David Adams**

From Gianluca:

Hello David,

let me summarise the status of issue 9264 about readout-level IDs (<https://cdcv.fnal.gov/redmine/issues/9264>).

- a feature branch is implemented based on LArSoft v5
- it contains an interface change that intentionally demands the experiments to take a decision on how to deal with the extended interface
- I created an implementation for the "standard" channel mapping
- I copied the implementation for DUNE from your code
- this leaves out protoDUNE and SBND \* I will take care of SBND when it will update to a current LArSoft \* I don't know the status for protoDUNE; I think single phase shared channel mapping implementation with the far detector (?) and as such it should be already covered by your code, which I copied; double phase instead is not covered
- in the current conditions, my understanding is that dunetpc would not compile because protoDUNE DP mapping class does not implement the new pure virtual functions

**#15 - 10/13/2016 07:09 AM - David Adams**

Gianluca:

Thanks for you work on this. We should move as quickly as possible to get this into develop and then releases of larsoft and dunetpc. By its nature, this is an extension that should not break any existing clients and so I very much prefer to work in the develop branch of dunetpc and not have create and reconcile feature branches.

I don't know a lot about the DP detector but I believe it is like most non-DUNE detector in that there is a one-to-one mapping between TPCs and APAs and so should be handled like those detectors. Can you add such code? If not, just move your code to larsoft develop and Tingjun or I will fill in

the missing methods.

It would be good to do this soon--the ticket is well over a year old. If this is not going to converge, I will take a different approach and add an APA/ROP mapping service that is distinct from the geometry service

Thank you.

David

**#16 - 10/13/2016 11:26 AM - Gianluca Petrillo**

I have implemented a mapping that is a copy of the "standard" one for DUNE "CRM" mapping, that is used in protoDUNE DualPhase 10kt geometry. This implementation should be checked by experts. Also, DUNE should consider to use directly the standard mapping for it; if there are technical questions on that, Fermilab LArSoft team can provide support as usual. If the difference is just a different sorter, the standard mapping can be templated on the sorter, or the sorter can be dynamically passed in.

The implementation **needs** to go in as a feature branch, because it affects different repositories.

The implementation is pushed as branch feature/gp\_WrappedGeometry in larcobj, larcobj, lariatsoft and dunetpc. SBND will need to update its mappings when they update to a recent LArSoft version; ArgoNeuT and MicroBooNE use the standard mapping in LArSoft and do not need explicit update.

**#17 - 10/13/2016 11:30 AM - Gianluca Petrillo**

- % Done changed from 90 to 100

The branches are ready to go in.

The changes are non-breaking in that with the new branches no code will need to change. Yet, failing to adopt the new branches will cause a compilation failure.

**#18 - 10/13/2016 01:30 PM - Gianluca Petrillo**

For reference: this implementation covers part of what was [presented at LArSoft Coordination meeting on July 14, 2015](#).

The changes to data products are *not* covered by this ticket.

**#19 - 10/24/2016 04:47 PM - Gianluca Petrillo**

- Status changed from Assigned to Resolved

Changes were merged in LArSoft v06\_12\_00.

**#20 - 11/02/2016 02:22 PM - Gianluca Petrillo**

- Status changed from Resolved to Closed