

LArSoft - Task #17427

Feature # 11522 (Assigned): 'Global wire' coordinate within LArSoft geometry

Add geometry infrastructure for support of global wire coordinates

08/10/2017 11:17 AM - Katherine Lato

Status:	Feedback	Start date:	08/10/2017
Priority:	Normal	Due date:	
Assignee:	Tingjun Yang	% Done:	0%
Category:	Geometry	Estimated time:	28.00 hours
Target version:		Spent time:	4.00 hours
Description			
The specifics depend on the architecture of the global wire coordinate. Can't finish this task until it has been decided how it will be used, but some prototype work needs to be done before the decision can be made on how to design it.			

History

#1 - 11/17/2017 02:50 PM - Gianluca Petrillo

I have talked with Robert Sulej, who has illustrated me the use case already implemented in [dunetpc:source:dune/CVN/adcutils/SPMultiTpcDump_module.cc](https://dunetpc.sourceforge.net/dune/CVN/adcutils/SPMultiTpcDump_module.cc):

- an image in the wire/drift space is produced for each view of each TPC; in this case, the height (charge) of the calibrated signal on each channel and time is used
- the images from all TPC planes sharing the same projection direction are combined into a collage image covering the whole detector

Complications specific to DUNE arise from channels spanning different views, and from the fact that for the same projection direction (which can be expressed as

```
Error executing the latex macro (Missing template wiki_external_filter/macro_image with {:locale=>[:en], :formats=>[:pdf], :variants=>[], :handlers=>[:erb, :builder, :raw, :ruby, :haml, :rsb]}). Searched in: *
"/fnal/packages/redmine/3.4.11/linux-scientific7-x86_64-gcc-4.8.5-qs5cpueulghvvh7gnl5h52cbww7nw4j/plugins/wiki_external_filter/app/views" *
"/fnal/packages/redmine/3.4.11/linux-scientific7-x86_64-gcc-4.8.5-qs5cpueulghvvh7gnl5h52cbww7nw4j/plugins/simple_ci/app/views" *
"/fnal/packages/redmine/3.4.11/linux-scientific7-x86_64-gcc-4.8.5-qs5cpueulghvvh7gnl5h52cbww7nw4j/plugins/redmine_redcarpet_viewer/app/views" *
"/fnal/packages/redmine/3.4.11/linux-scientific7-x86_64-gcc-4.8.5-qs5cpueulghvvh7gnl5h52cbww7nw4j/plugins/redmine_jenkins/app/views" *
"/fnal/packages/redmine/3.4.11/linux-scientific7-x86_64-gcc-4.8.5-qs5cpueulghvvh7gnl5h52cbww7nw4j/plugins/redmine_group_assignee/app/views" *
"/fnal/packages/redmine/3.4.11/linux-scientific7-x86_64-gcc-4.8.5-qs5cpueulghvvh7gnl5h52cbww7nw4j/plugins/redmine_better_gantt_chart/app/view
s" * "/fnal/packages/redmine/3.4.11/linux-scientific7-x86_64-gcc-4.8.5-qs5cpueulghvvh7gnl5h52cbww7nw4j/plugins/embedded/app/views" *
"/fnal/packages/redmine/3.4.11/linux-scientific7-x86_64-gcc-4.8.5-qs5cpueulghvvh7gnl5h52cbww7nw4j/app/views" )
) may be labelled with different view names in different TPCs (that is,  $u$  and  $v$  views are mixed). The essential elements needed by this use case,
which have been implemented already with the specific knowledge of the detector geometry they are applied to, are, for each TPC wire plane:
```

1. the identification of which wire plane has a given wire coordinate direction, since the view label can't be trusted
2. whether the wire coordinates increase or decrease along a given direction, necessary to flip the wire axis direction and have all the images consistent
3. the identification of which is the ID of the "first" wire on the plane, defined as the one with the lowest "global" wire coordinate
4. a wire offset, defined as the "global" coordinate of the first wire on the plane, necessary to translate the TPC image into the right place of the global image on the wire axis
5. whether the TDC tick values increase or decrease along a given direction, necessary to flip the drift axis direction and have all the images consistent
6. a depth offset, defined as the "global" coordinate of the plane itself, necessary to translate the TPC image into the right place of the global image on the drift axis

Existing facilities have been identified to facilitate all of these quantities in a detector-agnostic way.

The basic idea is to choose an arbitrary TPC and wire plane as a *reference* (TPCref as a geo::TPCGeo and planeRef as a geo::PlaneGeo), and to use the coordinates from all the other TPCs converted to be relative to this common reference. These will be referred to as *global coordinates*. After these definitions, for each given TPC (TPC, also a geo::TPCGeo):

1. the chosen wire coordinate direction is planeRef.GetIncreasingWireDirection(); its comparison with all the planes from each TPC should allow the discovery of the matching plane in them (*see low level tools below*)
2. **the global wire coordinate of a point is obtained as planeRef.WireCoordinate(point)**, and the one of a wire (geo::WireGeo) can be obtained as planeRef.WireCoordinate(wire.GetCenter())
whether the wire coordinates increase or decrease along a given direction: boolconst increasingWireCoord = (planeRef.WireCoordinate(plane.FirstWire().GetCenter()) < planeRef.WireCoordinate(plane.LastWire().GetCenter()));, which is the flag for flipping wire coordinate direction;
the lowest wire is geo::WireGeo const& lowestWire = increasingWireCoord? plane.FirstWire(): plane.LastWire();
3. the wire offset is planeRef.WireCoordinate(lowestWire.GetCenter())
4. whether the TDC tick values increase in the same direction as in the reference TPC: geo::vect::Dot(TPC.DriftDir(), TPCref.DriftDir()) > 0

5. depth offset: `planeRef.DistanceFromPlane(plane.GetCenter())`

Low level tools that may further help this use case:

- matching of planes with the same wire direction; these methods are expected to be "slow" and their result should be cached by the user if needed often:
 - `bool geo::PlaneGeo::isAbsWireCoordinateDirection(geo::vect::Vector_t const& dir, double tol = 1e-5) const` returning whether `dir` is the same direction (least a tolerance `tol`) as the one of the wire coordinate, barred the verse; an equivalent `bool geo::PlaneGeo::isWireCoordinateDirection(geo::vect::Vector_t const& dir, double tol = 1e-5) const` may be provided to include also the verse in the comparison
 - complimentary `bool geo::PlaneGeo::hasSameAbsWireCoordinateDirection(geo::PlaneGeo const& plane, double tol = 1e-5) const` and `bool geo::PlaneGeo::hasSameWireCoordinateDirection(geo::PlaneGeo const& plane, double tol = 1e-5) const` directly comparing the wire coordinate direction of a plane
 - also complimentary `geo::PlaneGeo const* geo::TPCGeo::PlaneWithAbsWireCoordinateDirection(geo::vect::Vector_t const& dir, double tol = 1e-5) const`, `geo::PlaneGeo const* geo::TPCGeo::PlaneWithAbsWireCoordinateDirectionAs(geo::PlaneGeo const& plane, double tol = 1e-5) const`, and `geo::PlaneGeo const* geo::TPCGeo::PlaneWithWireCoordinateDirection(geo::vect::Vector_t const& dir, double tol = 1e-5) const` and `geo::PlaneGeo const* geo::TPCGeo::PlaneWithWireCoordinateDirectionAs(geo::PlaneGeo const& plane, double tol = 1e-5) const` returning the plane in the TPC with the wire coordinate direction same as the one directly specified or as the one from the specified plane
- complimentary `double geo::PlaneGeo::WireCoordinate(geo::WireGeo const&) const` to get the wire coordinate of a wire directly
- generalisation of `geo::PlaneGeo::WireIDincreasesWithZ` method: `bool geo::PlaneGeo::WireIDincreasesAlong(geo::vect::Vector_t const& dir) const` returning whether the wire coordinates increase when going toward the positive verse of `dir`
- tagging of planes and TPCs with parallel or antiparallel drift directions:
 - complimentary `bool geo::TPCGeo::isDriftDirection(geo::vect::Vector_t const& dir, double tol = 1e-5) const` and `bool geo::TPCGeo::isAbsDriftDirection(geo::vect::Vector_t const& dir, double tol = 1e-5) const` to assess if TPC has drift direction exactly along `dir`, or along the same direction as `dir` but either verse
 - complimentary `bool geo::TPCGeo::hasSameDriftDirection(geo::PlaneGeo const& dir, double tol = 1e-5) const` and `bool geo::TPCGeo::hasSameAbsDriftDirection(geo::PlaneGeo const& dir, double tol = 1e-5) const` and `bool geo::TPCGeo::hasSameDriftDirection(geo::TPCGeo const& dir, double tol = 1e-5) const` and `bool geo::TPCGeo::hasSameAbsDriftDirection(geo::TPCGeo const& dir, double tol = 1e-5) const` for the same effect as in the previous item, but using the drift direction from geometry objects
 - complimentary `bool geo::PlaneGeo::isDriftDirection(geo::vect::Vector_t const& dir, double tol = 1e-5) const` and `bool geo::PlaneGeo::isAbsDriftDirection(geo::vect::Vector_t const& dir, double tol = 1e-5) const` to assess if the plane has drift direction exactly along `dir`, or along the same direction as `dir` but either verse
 - complimentary `bool geo::PlaneGeo::hasSameDriftDirection(geo::PlaneGeo const& dir, double tol = 1e-5) const` and `bool geo::PlaneGeo::hasSameAbsDriftDirection(geo::PlaneGeo const& dir, double tol = 1e-5) const` and for the same effect as in the previous item, but using the drift direction from the specified wire plane
- complimentary `double geo::PlaneGeo::DistanceFromPlane(geo::PlaneGeo const&) const` to get the drift distance of another wire plane directly

Note The image obtained this way is not homogeneous; the use `geo::PlaneGeo::PlaneCoordinate()` instead of `geo::PlaneGeo::WireCoordinate()` will fix that by returning a standard length (centimetres).

#2 - 11/17/2017 03:18 PM - Gianluca Petrillo

- Status changed from Assigned to Work in progress

#3 - 04/06/2018 02:52 AM - Leigh Whitehead

The specifics of the global wire coordinate needs to be fully defined before this work can be completed.

The code committed by Robert gives purely a global wire number and is not, as such, a coordinate direction. It provides the ability to make a single image of (wire number, tdc) for the entire detector. Due to the relative orientation of the APAs either side of a CPA, this means that the U and V views must be mixed.

#4 - 04/10/2018 12:37 PM - Gianluca Petrillo

I am not married to the "U" and "V" names, because they don't have a specific definition (in fact, I struggle to have those labels deprived of meaning).

In general, I foresee one global coordinate axis per wire orientation (defined for example as the angle from the z axis). I don't remember whether the U and V views together define two of them¹, or four².

If the latter, then a "global" image of the cryostat from a chosen orientation must contain gaps from the drift volumes with no wires with that orientation. I am not sure whether this is the issue you are talking about. Such image would still be useful to put together images from different APA's on the same geometry plane (that is, aligned and at different z coordinates).

In the former case of just two orientations, I don't see a problem in the definition of the image.

¹ For example, in SBND the orientations are defined to be $\pm 60^\circ$, symmetric.

² For example, if one APA has angles at 44° and -46° on one side, and -44° and 46° on the other side.

#5 - 05/08/2018 10:07 AM - Katherine Lato

- Assignee changed from Gianluca Petrillo to Erica Snider

#6 - 08/03/2018 10:02 AM - Katherine Lato

- *Status changed from Work in progress to Feedback*

- *Assignee changed from Erica Snider to Tingjun Yang*

Need feedback from DUNE to understand the answers to some of the questions raised earlier. We understand that this may need to wait until after the initial ProtoDUNE runs are finished, and that LArSoft may need to be involved in feedback discussions.