

## LArSoft - Task #14281

Milestone # 17338 (Accepted): Phase 3 Tracking Data Product

### Provide an interface for access of reconstructed track information

10/26/2016 10:28 AM - Gianluca Petrillo

<b>Status:</b>	Accepted	<b>Start date:</b>	10/26/2016
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	60%
<b>Category:</b>	Architecture	<b>Estimated time:</b>	100.00 hours
<b>Target version:</b>	2019-1-quarter	<b>Spent time:</b>	60.00 hours

#### Description

Similarly as for clusters (issue [#14265](#)), add an interface providing access to track information:

- providing access to all the pre-computed information currently available (currently `recob::Track`)
- no less-than-trivial computation is performed
- access to lower level associated objects; possibly incomplete list:
  - clusters (ideally via `view::Cluster`)
  - hits
  - vertices
  - trajectory points ("space points")
  - track seeds
- access to truth information needs to be considered both in principle and in practice

This interface object, with the candidate name of `view::Track`, is bound to a main reconstruction action (that is, a track making module).

Careful consideration is needed as how to include vertex information: many tracking algorithms provide vertex information, but vertices can be as well produced by specific algorithms and then associated to tracks, or used as input to reconstruct tracks.

A simple approach is to stick to the same policy as for `view::Cluster` (issue [#14265](#)) and navigate only elements created with the main tracking module. Then, the other connections would be provided by a higher level abstraction (see issue [#14061](#)).

This task is related to the redefinition of the track data products, but it's not fully dependent on it since that one mainly deals with content and storage model, while this one mainly deals with access interface.

#### Related issues:

Related to LArSoft - Necessary Maintenance #15453: Remove legacy data members...	<b>Assigned</b>	<b>02/03/2017</b>
Related to LArSoft - Necessary Maintenance #15446: BezierTrack should be remo...	<b>Closed</b>	<b>02/02/2017</b>
Blocked by LArSoft - Feature #14786: Rediscuss the content of <code>recob::Track</code>	<b>Rejected</b>	<b>12/08/2016</b>

#### Associated revisions

##### Revision [f75cbd73](#) - 02/03/2017 05:02 PM - Gianluca Petrillo

Changed deprecation warning to be less intrusive (see issue [#14281](#) )

#### History

##### #1 - 10/27/2016 12:53 PM - Katherine Lato

- Status changed from New to Assigned

##### #2 - 12/08/2016 11:23 AM - Gianluca Petrillo

- Estimated time set to 60.00 h

This step is on hold until the new content of the tracking system is defined.

##### #3 - 12/08/2016 11:30 AM - Gianluca Petrillo

- Blocked by Feature [#14786](#): Rediscuss the content of `recob::Track` added

##### #4 - 02/02/2017 11:37 AM - Gianluca Petrillo

The first step is the redefinition of the content of `recob::Track`.

The design and implementation of it is staged:

1. the new data structures are plugged into LArSoft with an interface as backward-compatible as possible, to minimise the impact on current code
2. the code is transitioned into the new interface; utilities are provided for writing the new objects in a compliant way and for basic navigation
3. a complete navigation interface is provided

The delivery time of first stage is constrained by MicroBooNE production, whose official deadline is end of January 2017.

#### #5 - 02/02/2017 11:42 AM - Gianluca Petrillo

The material for stage 1 (see [note 4](#)) has been delivered in branches `feature/cerati_TrackTrajectory` (including `lardataobj` and others).

This is scheduled for merge into LArSoft v06\_23\_00 (which was basically delayed to merge it).

#### #6 - 02/03/2017 12:01 PM - Gianluca Petrillo

The new `recob::Track` class interface is as backward-compatible as possible to the old one.

Some data members are been left in there until there is agreement on how to life without them (issue [#15453](#)).

### Interface change

Some data members underwent an interface change. These were the only cases where a change was unavoidable.

Return value changed from reference to value, because of internal changes:

- `DirectionAtPoint()`, `LocationAtPoint()`, `Vertex()`, `End()`, `VertexDirection()`, `EndDirection()` return a `TVector3` instead than a `TVector3 const&`
- `CovarianceAtPoint()`, `VertexCovariance()` and `EndCovariance()` return a `TMatrixD` instead of a `TMatrixD const&`

Typically this is a problem when storing the return into a reference variable, e.g.

```
TVector3 const& vertex = track.Vertex();
```

The most durable, but less readable, way to fix this is to let the compiler do what it takes:

```
decltype(auto) vertex = track.Vertex();
```

will have `vertex` of whatever exact type `recob::Track::Vertex()` returns.

Be aware that this backward compatibility is going to be removed, and all these methods will return a `recob::Track::Point_t` or `recob::Track::Vector_t` as appropriate. Since these vectors have small differences in the interface with respect to `TVector3`, the following code (e.g., the one using `vertex`) may have to be fixed.

The same holds for the covariance matrices.

### Deprecated methods

A few methods have been *formally* deprecated. This means that every time you use one of them in your code, the compiler will give you a warning. **You can and should fix your code to avoid that.** Please also report cases where LArSoft still uses them, in case we missed (m)any.

- `Extent(std::vector<double>&, std::vector<double>&) const`: use `Extent()` instead; it returns a pair of `recob::Track::Point_t`. If your use requires `std::vector<double>`, please contact us so that the downstream interface can be extended to support `Point_t` as well.
- `NumberFitMomentum()`: the new protocol requires the presence of as many momenta as there are points in a track trajectory, or none at all. Therefore, use `NumberTrajectoryPoints()` to know how many of both there are in the track. Another use in the code was to determine whether there was momentum information available in the track or not. In that case, the usual check if `(track.NumberFitMomentum() > 0)` should be replaced by `if (track.HasMomentum())`.

Other data members and methods have been nominally deprecated:

- `TrajectoryAtPoint(size_t, TVector3&, TVector3&)`: the new method `TrajectoryPoint(size_t)` (no more `At` in the name!) returns a `TrajectoryPoint_t` object with copies of the direction and momentum. Note that this is different from before, where the returned vector was a direction rather than a momentum. Also, the results are not `TVector3` any more, but rather `Point_t` and `Vector_t`.
- `Direction(double*, double*)` has gone through a treatment similar to `Extent(TVector3&, TVector3&)` (see above); the return type of the replacement, `Direction()`, is a pair of `Vector_t`.
- `GlobalToLocalRotationAtPoint(size_t, TMatrixD&)`, `LocalToGlobalRotationAtPoint(size_t, TMatrixD&)` should be replaced by `GlobalToLocalRotationAtPoint(size_t)` and `LocalToGlobalRotationAtPoint(size_t)` respectively, which return a rotation matrix (`Rotation_t`) that can be used directly on `Point_t` and `Vector_t` objects.

These members have not *formally* deprecated yet, and the compiler will not issue any warning when used.

An example of dealing with the `Extent()` change and tips for upgrading `TVector3` to `GenVector` vectors (used by the new `recob::Track` interface) can be found in the [From ROOT vectors \(TVector3\) to ROOT GenVector](#) wiki page.

#### #7 - 02/06/2017 10:30 AM - Gianluca Petrillo

- Related to Necessary Maintenance [#15453](#): Remove legacy data members from `recob::Track` added

**#8 - 02/06/2017 10:32 AM - Lynn Garren**

- Related to Necessary Maintenance #15446: BezierTrack should be removed from LArSoft added

**#9 - 04/27/2017 10:54 AM - Katherine Lato**

- % Done changed from 0 to 60

- Estimated time changed from 60.00 h to 100.00 h

**#10 - 07/27/2017 11:53 AM - Katherine Lato**

as part of a discussion with Erica, Gianluca and Giuseppe, decided that to proceed on this will provide an interface for writing tracks and one for reading tracks that hides connections between tracks in other products.

Giuseppe will take care of the writing interface, Gianluca will take care of the reading interface.

**#11 - 08/02/2017 04:26 PM - Katherine Lato**

- Status changed from Assigned to Accepted

- Assignee deleted (Gianluca Petrillo)

- Parent task changed from #14060 to #17338

Moving this task to phase 3 of track work on data products.

**#12 - 01/09/2018 02:53 PM - Katherine Lato**

- Target version set to 2019-1-quarter