#tutorial_event_display

Pandora event display

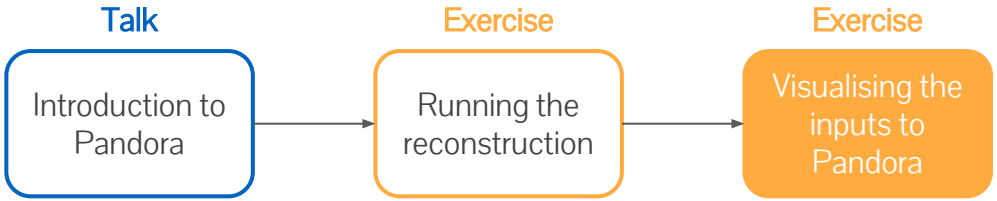
Part 1: Inputs to Pandora

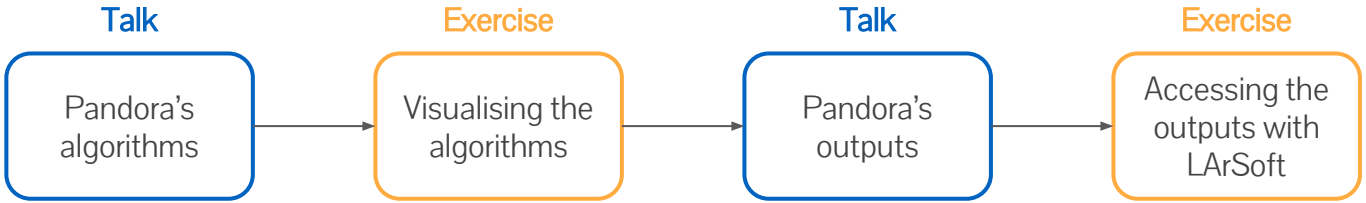
2nd November 2021
LArTPC Software Analysis Workshop - Edinburgh

Steve Dennis - For the Pandora team

Tutorial and slides developed by Andrew Smith

Reconstruction session





Goals

- **This session scheduled for 40 mins**

- **Main goal** - Visualise the input hits in Pandora
  - Enable visual monitoring in the Pandora configuration XML file
  - Re-run Pandora to start the EVE GUI and see the input hits
  - Get to grips with the GUI

Main goal

Visualise the input hits in Pandora

Modifying the Pandora XML

- Add our config directory to the `FW_SEARCH_PATH` so pandora knows where to look for it

```
$ mkdir $MRB_TOP/reco/config # This could already exist
$ export FW_SEARCH_PATH=$MRB_TOP/reco/config:$FW_SEARCH_PATH
$ export FHICL_FILE_PATH=$MRB_TOP/reco/config:$FHICL_FILE_PATH
```

- Go to our config directory and make a copy of the Pandora XML settings file

```
$ cd $MRB_TOP/reco/config
$ cp $LARPANDORA_DIR/scripts/PandoraSettings_Master_Standard.xml MyPandoraSettings_Master_Standard.xml
$ vim MyPandoraSettings_Master_Standard.xml
```

- Enable Pandora Monitoring by modifying the file, then save and close:

```
<pandora>
    <!-- GLOBAL SETTINGS -->
    <IsMonitoringEnabled>true</IsMonitoringEnabled>
    ...
```

If you closed your terminal since the last session, don't forget to set everything up again! You will also need to export your `FHICL_FILE_PATH` again!

Writing a FHiCL file to run the event display

- The event display runs within Pandora. To avoid having to run all of the reconstruction steps again, let's make a new FHiCL file that just runs Pandora using our custom XML configuration

```
$ cd $MRB_TOP/reco/config          # You're probably already here
$ vim event_display_driver.fcl
```

- Create `event_display_driver.fcl`, and add the following lines, save and close:

```
#include "standard_reco1reco2_sbnd.fcl"

process_name: EventDisplay


# Use our custom settings file
physics.producers.pandora.ConfigFile: "MyPandoraSettings_Master_Standard.xml"

# Only run pandora
physics.eventDisplay: [ pandora ]
physics.trigger_paths: [ eventDisplay ]

# Don't produce any output ART root files
physics.end_paths: []
```

Use our modified settings for reco

Rename the process

Point to our new XML settings file

Case Matters!

Only run the Pandora stage

Don't produce output root files, we only want to see the events

What are we going to visualize?

MyPandoraSettings_Master_Standard.xml

```
<pandora>
    <!-- GLOBAL SETTINGS -->
    <IsMonitoringEnabled>true</IsMonitoringEnabled>
    <ShouldDisplayAlgorithmInfo>false</ShouldDisplayAlgorithmInfo>
    <SingleHitTypeClusteringMode>true</SingleHitTypeClusteringMode>

    <!-- ALGORITHM SETTINGS -->
    <algorithm type = "LArPreProcessing">
        <OutputCaloHitListNameU>CaloHitListU</OutputCaloHitListNameU>
        <OutputCaloHitListNameV>CaloHitListV</OutputCaloHitListNameV>
        <OutputCaloHitListNameW>CaloHitListW</OutputCaloHitListNameW>
        <FilteredCaloHitListName>CaloHitList2D</FilteredCaloHitListName>
        <CurrentCaloHitListReplacement>CaloHitList2D</CurrentCaloHitListReplacement>
    </algorithm>
    <algorithm type = "LArVisualMonitoring">
        <CaloHitListNames>CaloHitListU CaloHitListV CaloHitListW</CaloHitListNames>
        <ShowDetector>true</ShowDetector>
    </algorithm>

    <algorithm type = "LArMaster">
        <CRSettingsFile>PandoraSettings_Cosmic_Standard.xml</CRSettingsFile>
        <NuSettingsFile>PandoraSettings_Neutrino_Standard.xml</NuSettingsFile>
        <SlicingSettingsFile>PandoraSettings_Slicing_Standard.xml</SlicingSettingsFile>
        <StitchingTools>
            <tool type = "LArStitchingCosmicRayMerging"><ThreeDStitchingMode>true</ThreeDStitchingMode></tool>
            <tool type = "LArStitchingCosmicRayMerging"><ThreeDStitchingMode>false</ThreeDStitchingMode></tool>
        </StitchingTools>
        <CosmicRayTaggingTools>
            <tool type = "LArCosmicRayTagging"/>
        </CosmicRayTaggingTools>
        <SliceIdTools>
            <tool type = "LArSimpleNeutrinoId"/>
        </SliceIdTools>
        <InputHitListName>Input</InputHitListName>
        <RecreatedPfoListName>RecreatedPfos</RecreatedPfoListName>
        <RecreatedClusterListName>RecreatedClusters</RecreatedClusterListName>
        <RecreatedVertexListName>RecreatedVertices</RecreatedVertexListName>
        <VisualizeOverallRecoStatus>false</VisualizeOverallRecoStatus>
    </algorithm>

    <algorithm type = "LArVisualMonitoring">
        <ShowCurrentPfos>true</ShowCurrentPfos>
        <ShowDetector>true</ShowDetector>
    </algorithm>
</pandora>
```

Open your custom Pandora settings file

The line we just changed to enable visualisations

The visual monitoring algorithm starts up the event display - first we'll look at the input hit collections in the U, V, and W views

The master algorithm is in charge of running the different steps of the Pandora's pattern recognition - recall we configured Pandora to only to run the neutrino algorithm chain, which is defined in: `PandoraSettings_Neutrino_Standard.xml`

After the pattern-recognition is finished, we run the visual monitoring algorithm again to update the event display to now show the reconstructed particles = PFOs

Running the event display

- Now just run your FHiCL file to launch the event display. You need to point to our new root files with reconstruction information so we have access to the hits

```
$ cd $MRB_TOP/reco/work
$ lar -c event_display_driver.fcl -s reco2_events.root -n 2
```

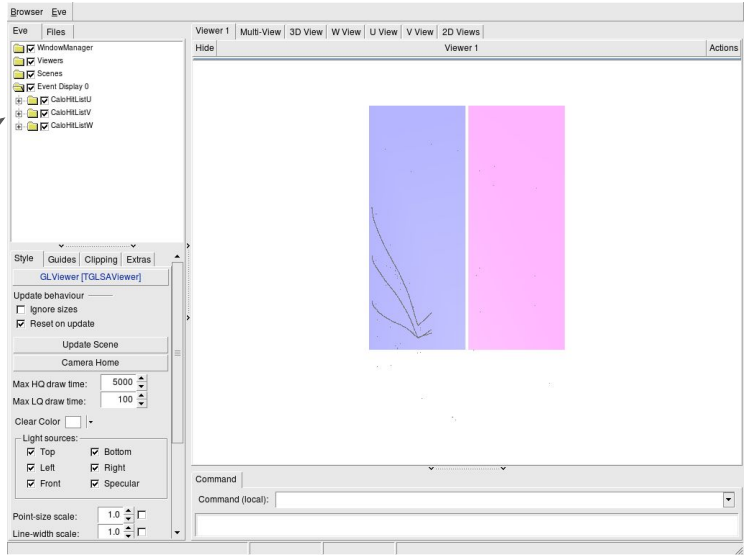
For now, let's just look at 2 events. If this command fails, check that you used the `-X` option with `ssh` (or using VNC). If you still have problems, ask us.

- After a few seconds, the event display will pop-up

MyPandoraSettings_Master_Standard.xml

```
<pandora>

 ... Get the input lists of hits ...

   <algorithm type = "LArVisualMonitoring">
       <CaloHitListNames>CaloHitListU CaloHitListV CaloHitListW</CaloHitListNames>
       <ShowDetector>true</ShowDetector>
   </algorithm>

 ... Run the pattern recognition ...

   <algorithm type = "LArVisualMonitoring">
       <ShowCurrentPfos>true</ShowCurrentPfos>
       <ShowDetector>true</ShowDetector>
   </algorithm>
</pandora>
```

Event 1
Visualization 1



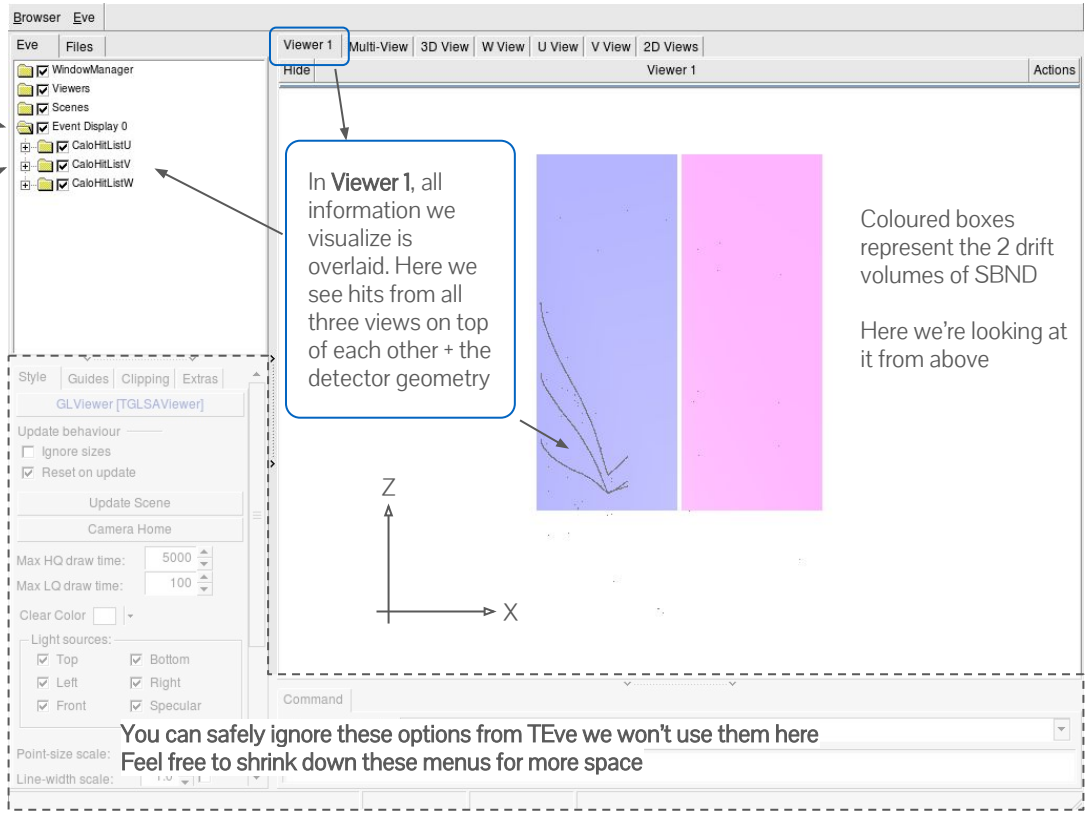
Looking at the input hits - Viewer 1

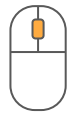
Every time the visual monitoring algorithm runs, we get a new event display (enumerated from zero)

Try checking and unchecking the boxes to turn on and off the hits from each of the views
- `CaloHitListU`
- `CaloHitListV`
- `CaloHitListW`

The 2D hit coordinates are stored in Pandora as 3D coordinates (X, Y, Z)

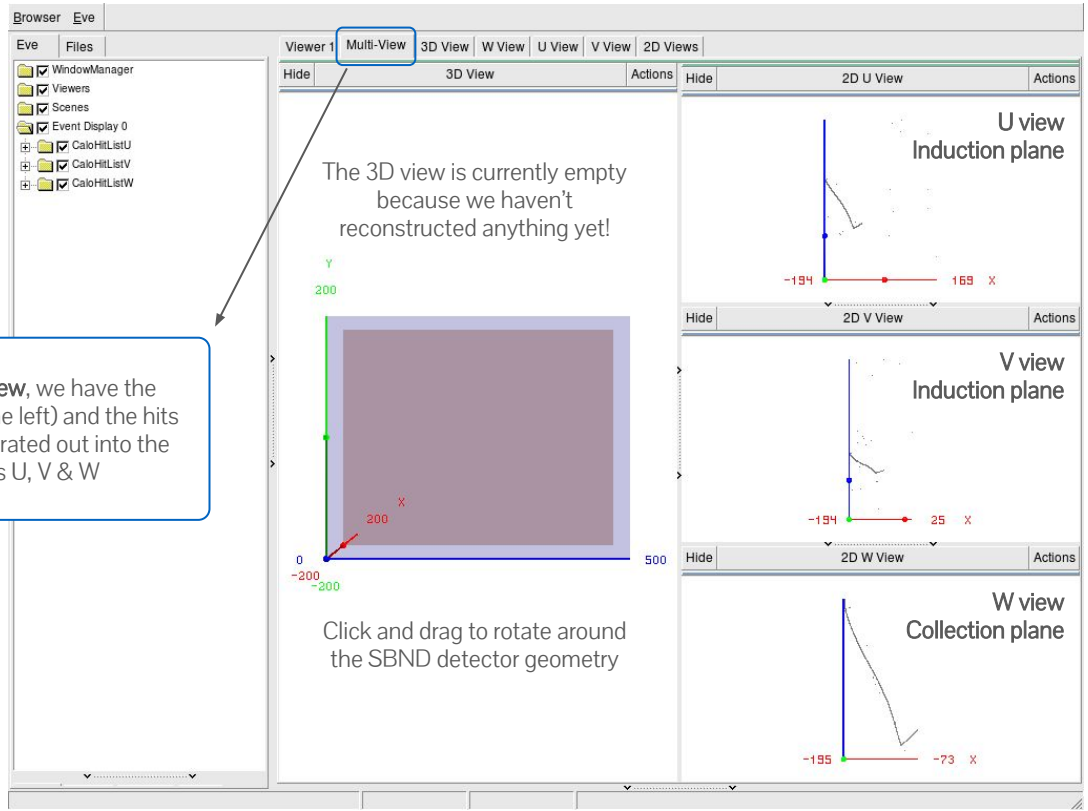
X = drift time coordinate
Y = 0
Z = wire number coordinate



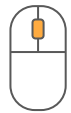


Wheel up - zoom out
Wheel down - zoom in
Wheel press + drag - pan viewport

Looking at the input hits - Multi-View



In the **Multi-View**, we have the 3D view (on the left) and the hits (on right) separated out into the three 2D views U, V & W

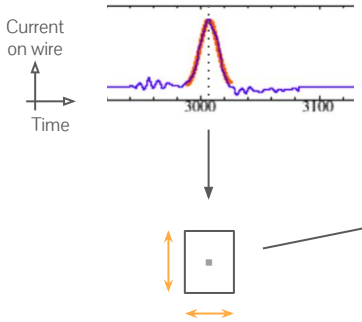


Wheel up - zoom out
Wheel down - zoom in
Wheel press + drag - pan viewport

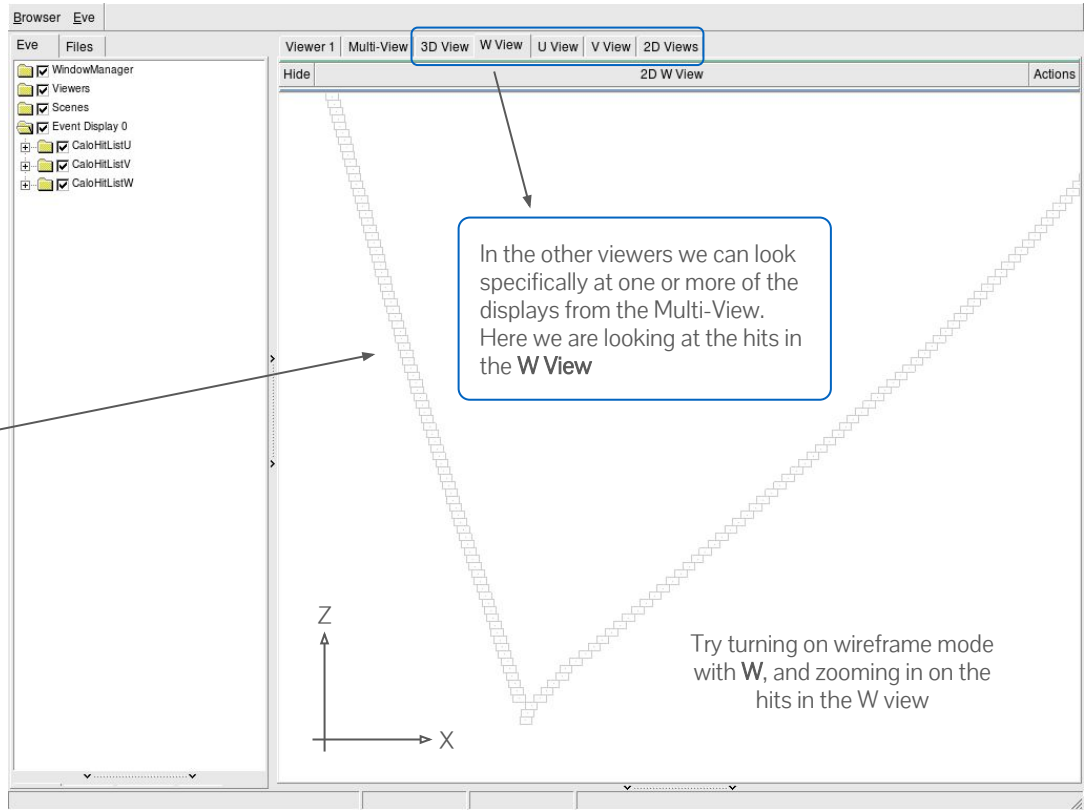
Left press + drag - rotate 3D view

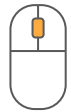
Looking at the input hits - W View

Hits are drawn as a rectangle. The X-coordinate is calculated from the time of the hit, and the Z-coordinate is from the wire number



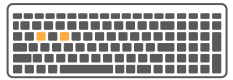
The X-width of the hit is from the Gaussian fit to the waveform, and the Z-width is the wire-spacing distance





Wheel up - zoom out
Wheel down - zoom in
Wheel press + drag - pan viewport

Left press + drag - rotate 3D view



W - wireframe mode
R - return from wireframe mode

Looking at the final output of the pattern-recognition

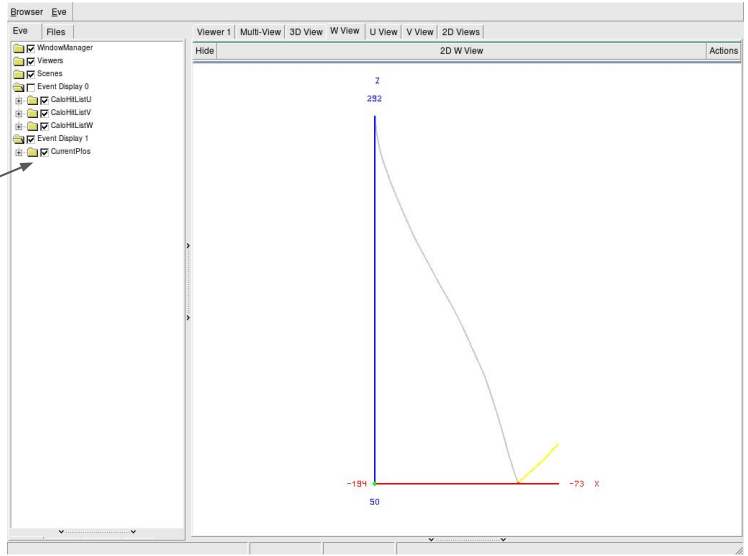
- Click in the terminal window and press `Return` ↵
- This will exit from the current visual monitoring algorithm and continue running through our settings file
- After the pattern-recognition is finished, we reach the second visual monitoring algorithm - go back to the event display window to see what we are visualizing

MyPandoraSettings_Master_Standard.xml

```
<pandora>

 ... Get the input lists of hits ...

   <algorithm type = "LArVisualMonitoring">
       <CaloHitListNames>CaloHitListU CaloHitListV CaloHitListW</CaloHitListNames>
       <ShowDetector>true</ShowDetector>
   </algorithm>

 ... Run the pattern recognition ...

   <algorithm type = "LArVisualMonitoring">
       <ShowCurrentPfos>true</ShowCurrentPfos>
       <ShowDetector>true</ShowDetector>
   </algorithm>
</pandora>
```

Event 1

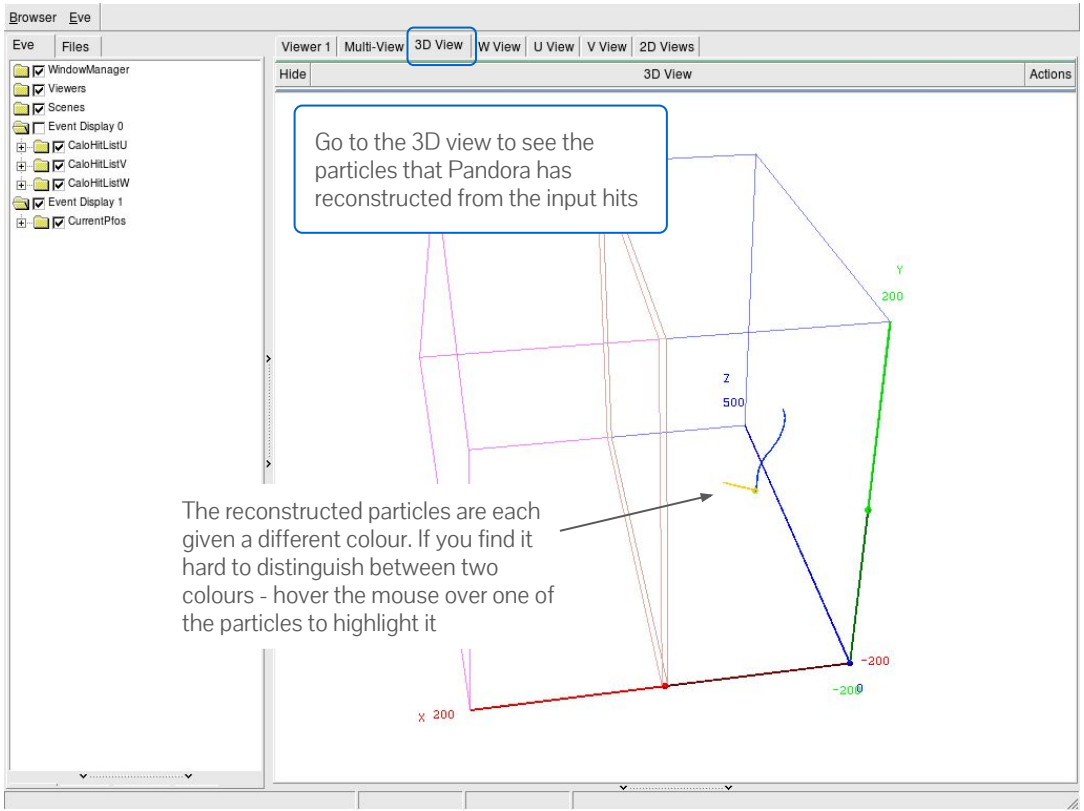
Visualization 2

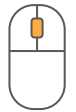


Looking at the reconstructed particles - 3D view

We've now moved on to the next visualisation

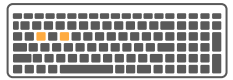
Unfortunately at the moment these checkboxes only work in Viewer 1





Wheel up - zoom out
Wheel down - zoom in
Wheel press + drag - pan viewport

Left press + drag - rotate 3D view



W - wireframe mode
R - return from wireframe mode

Moving through events

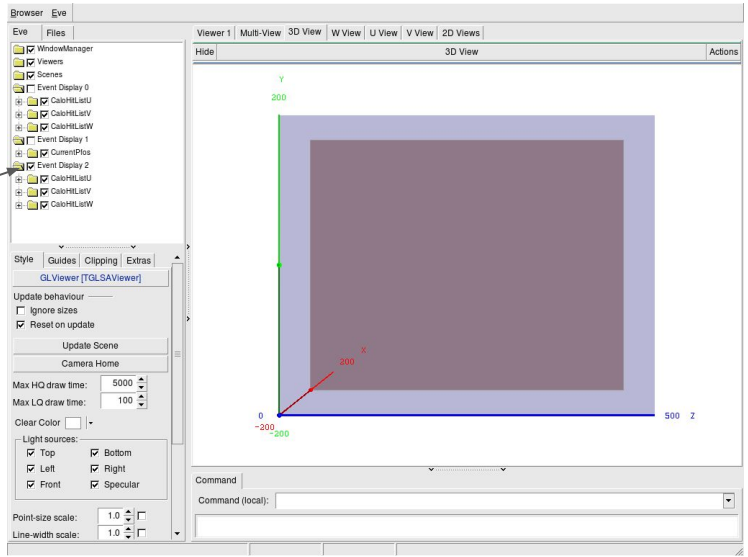
- Click in the terminal window and press **`Return`** ↵ again
- As before, this will exit from the current visual monitoring algorithm and continue through our settings file
- Now we reached the end, Pandora will run again from the top with the next event - check the visualisation
- Click in the terminal window and press **`Return`** ↵ once again to show the second visualization for event 2
- Press **`Return`** ↵ a final time to close the display

MyPandoraSettings_Master_Standard.xml

```
<pandora>

 ... Get the input lists of hits ...

   <algorithm type = "LArVisualMonitoring">
       <CaloHitListNames>CaloHitListU CaloHitListV CaloHitListW</CaloHitListNames>
       <ShowDetector>true</ShowDetector>
   </algorithm>

 ... Run the pattern recognition ...

   <algorithm type = "LArVisualMonitoring">
       <ShowCurrentPfos>true</ShowCurrentPfos>
       <ShowDetector>true</ShowDetector>
   </algorithm>
</pandora>
```

Event 2

Visualization 1



Got spare time?

Try scanning through more events to get a feel for our input sample

Zoom in on the final reconstructed particles, is this what you expect?

Pandora development team

Slack: #tutorial_event_display

Pandora is an open project and new contributors would be extremely welcome.
We'd love to hear from you and we will always try to answer your questions.

Pandora Liaisons

Pandora lead	John Marshall	john.marshall@warwick.ac.uk
	Andy Blake	a.blake@lancaster.ac.uk
DUNEFD single phase	Dom Brailsford	d.brailsford@lancaster.ac.uk
	Andy Chappell	andrew.chappell@warwick.ac.uk
ProtoDUNE single phase	Leigh Whitehead	leigh.howard.whitehead@cern.ch
	Steve Dennis	sdennis@hep.phy.cam.ac.uk
ProtoDUNE dual phase & vertical drift	Dom Brailsford	d.brailsford@lancaster.ac.uk
	Maria Brigida Brunetti	Maria.Brunetti@warwick.ac.uk
DUNE NDLAr	Melissa Uchida	mauchida@hep.phy.cam.ac.uk
	Alex Moor	afm67@cam.ac.uk
	John Back	j.j.back@warwick.ac.uk
MicroBooNE	Alex Moor	afm67@cam.ac.uk
SBND	Dom Brailsford	d.brailsford@lancaster.ac.uk
	Henry Lay	h.lay@lancaster.ac.uk
	Ed Tyley	e.tyley@sheffield.ac.uk
ICARUS	Bruce Howard	bruhowar@indiana.edu

Graduate students

Ryan Cross
Henry Lay
Isobel Mawby
Alex Moor
Mousam Rai
Natsumi Taniuchi
Ed Tyley
Karolina Wresilo

github.com/PandoraPFA

PandoraPFA.slack.com

For reference

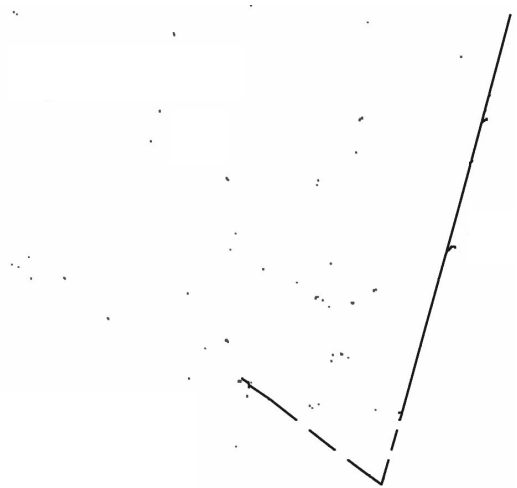
Making visualisations within Pandora

- Event displays are invaluable tools & a number of different options exist
- Today we will be focussing on the event display provided by Pandora


```
PANDORA_MONITORING_API(
  VisualizeCaloHits(myHits, ...
);
```

A snippet from a Pandora algorithm, that visualises the input hits via the monitoring API macro

The Pandora Monitoring API is the "middleman" that allows Pandora algorithms to easily make displays using ROOT's event visualisation environment, EVE



Pandora Monitoring API & Visual Monitoring Algo

- Many different visualisation options are available through the API to make bespoke displays, e.g.

```
/**
 *  @brief  Add CaloHits to the Eve event-display
 *
 *  @param  pandora the calling pandora instance
 *  @param  pCaloHitList list of calohits to be added to the event display
 *  @param  name of the calohit list
 *  @param  color The color the cluster elements are drawn with
 */
static void VisualizeCaloHits(const pandora::Pandora &pandora, const pandora::CaloHitList *const pCaloHitList,
                              const std::string &name, const Color color);
```

- Bespoke displays can be very useful to understand the specifics of a given algorithm
- Quite often though, all we need is to see the hits, clusters, etc. to understand the state of the pattern-recognition at a specific point
- The visual monitoring algorithm exists to do just that! All we need to do is add a snippet to our Pandora XML settings file, and re-run Pandora - no C++ necessary

Visual Monitoring Algorithm options reference

- These are the most useful options for this workshop - see the header for an exhaustive list

`<ShowCurrentCaloHits>`	Whether to show current calohitlist
`<CaloHitListNames>`	Names of calo hit lists to show
`<ShowCurrentClusters>`	Whether to show current clusters
`<ClusterListNames>`	Names of cluster lists to show
`<ShowCurrentPfos>`	Whether to show current particle flow object list
`<PfoListNames>`	Names of pfo lists to show
`<ShowCurrentVertices>`	Whether to show current vertex list
`<VertexListNames>`	Names of vertex lists to show
`<ShowDetector>`	Whether to display the detector geometry