Pandora event display

Part 1: Inputs to Pandora

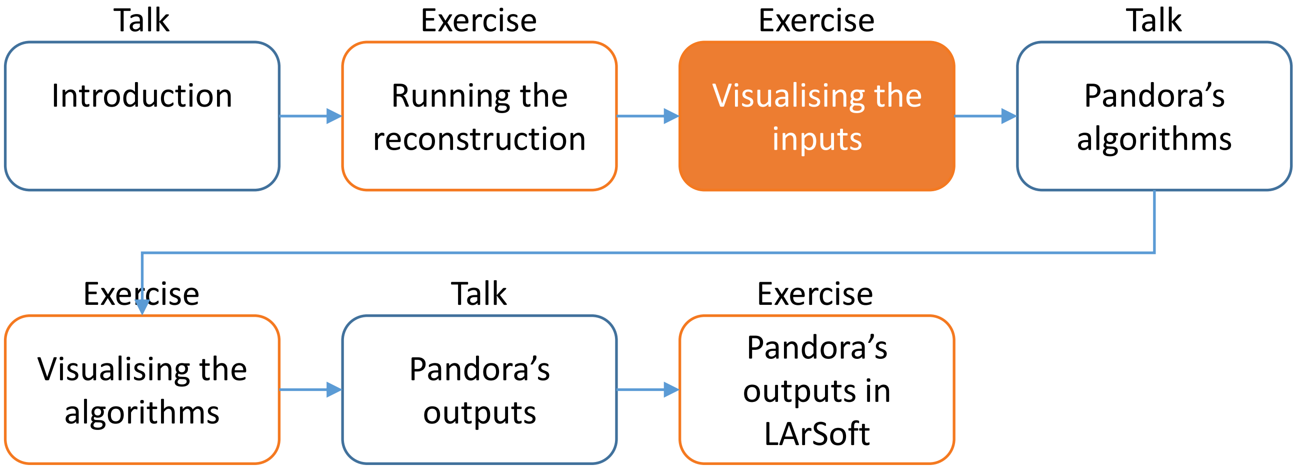
Dom Brailsford for the Pandora team

08/09/2022

UK-Latin America LArSoft Workshop

WARWICK

Reconstruction session



Credit: These slides are based on previous LArSoft workshop slides by Andrew Smith

Key references: Pandora ProtoDUNE paper
Pandora MicroBooNE paper

Goals

- This session scheduled for 40 minutes
- Main goal - Visualize 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

Visualize the input hits in Pandora

Modifying the Pandora XML

- Copy the standard Pandora reconstruction configuration file to your config directory

```
$ 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!

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

```
$ export FW_SEARCH_PATH=$MRB_TOP/reco/config:$FW_SEARCH_PATH
```

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
```

- Add the lines below to event_display_driver.fcl, save and close:

```
#include “standard_reco_dune10kt_1x2x6.fcl"

# We’ll run over the reco files, so we can’t reuse Reco as the process name
process_name: EventDisplay

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

# Run up to pandora
physics.event_display: [ rns, caldata, gaushit, hitfd, linecluster, pandora ]
physics.trigger_paths: [ event_display ]

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

Use our modified settings for reco

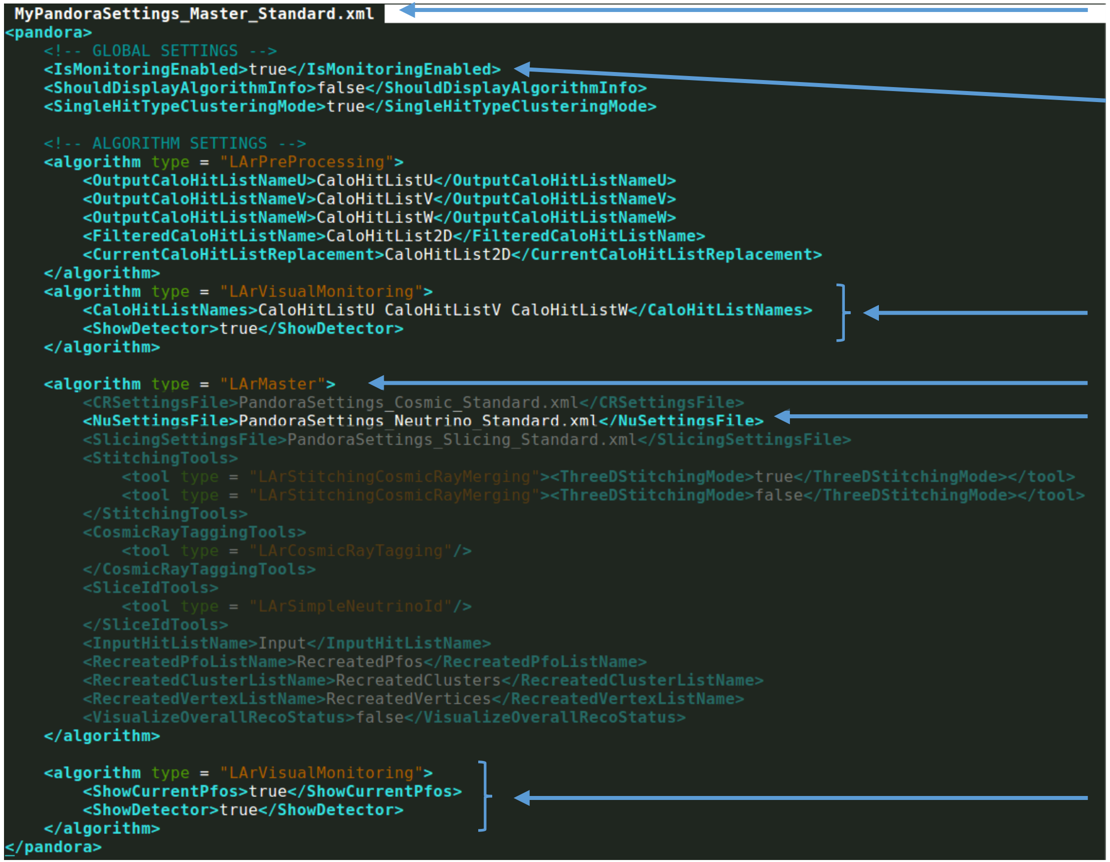
Rename the process

Point to our new XML settings file

Run up to 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

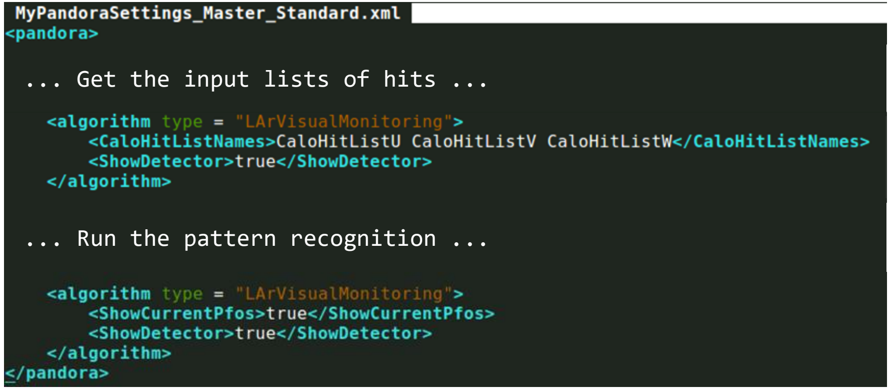
Running the event display

- 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

For now, let's just look at 2 events.

```
$ cd $MRB_TOP/reco/work
$ lar -c event_display_driver.fcl -s reco_1mu1p.root -n 2
```

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



MyPandoraSettings_Master_Standard.xml

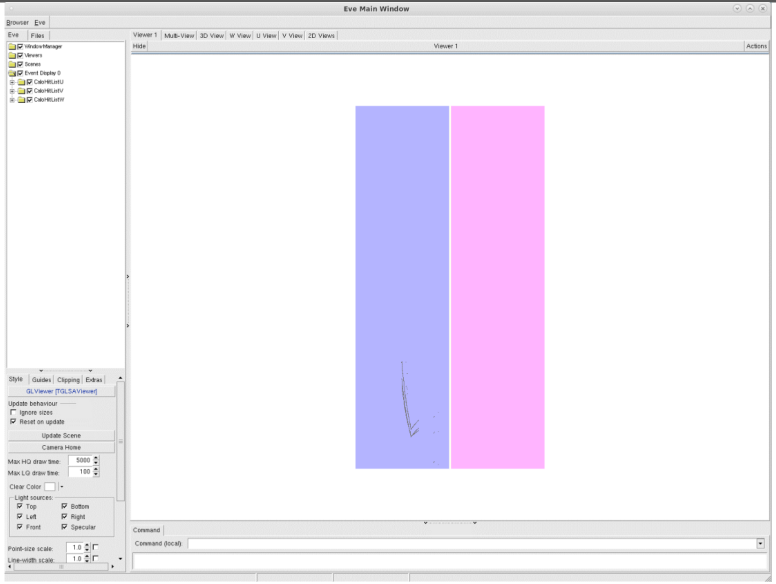
```
<pandora>

 ... 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>
```



Update behaviour
□ Ignore sizes
☑ Reset on update

Max HQ draw time: 5000
Max LQ draw time: 100
Clear Color

Light sources:
☑ Top ☑ Bottom
☑ Left ☑ Right
☑ Front ☑ Specular

Point-size scale: 1.0 □
Line-width scale: 1.0 □

Looking at the input hits - Viewer 1

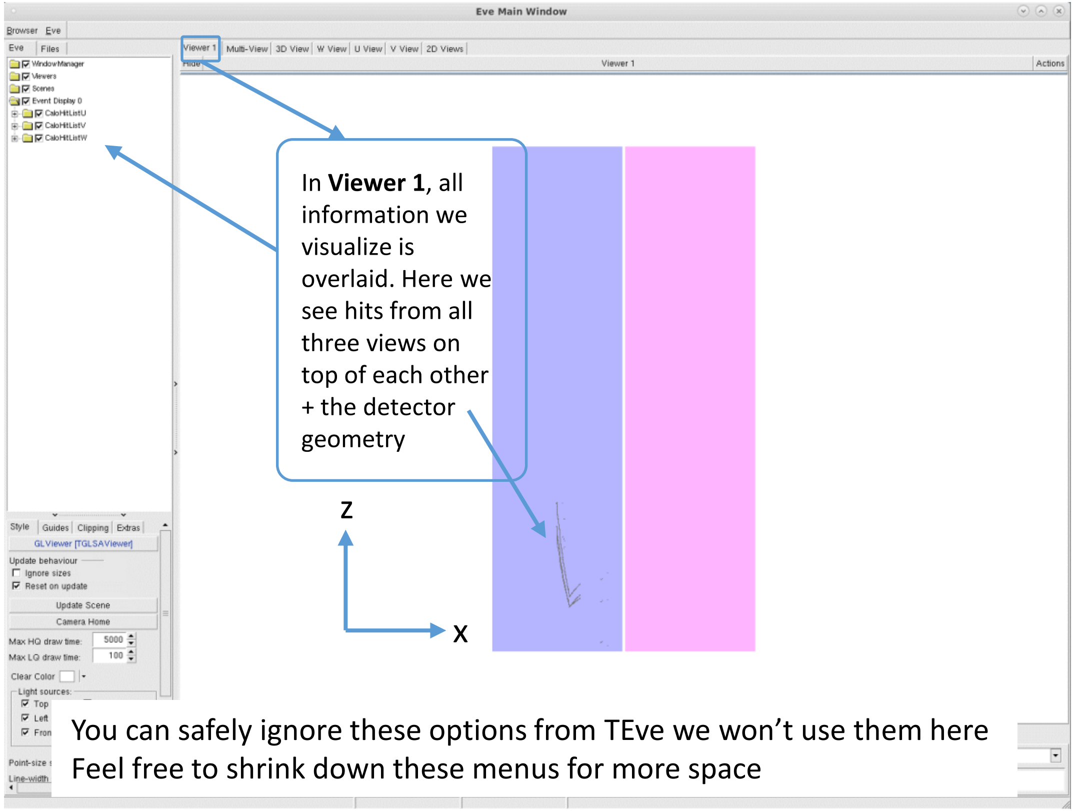
Every time the visual monitoring algorithm runs, we get a new event display (enumerated from zero) →

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



☑ WindowManager
☑ Viewers
☑ Scenes
☑ Event Display 0
☑ CaloHitListU
☑ CaloHitListV
☑ CaloHitListW

Update behaviour
□ Ignore sizes
☑ Reset on update

Max HQ draw time: 5000
Max LQ draw time: 100
Clear Color

Light sources:
☑ Top
☑ Left
☑ Fron

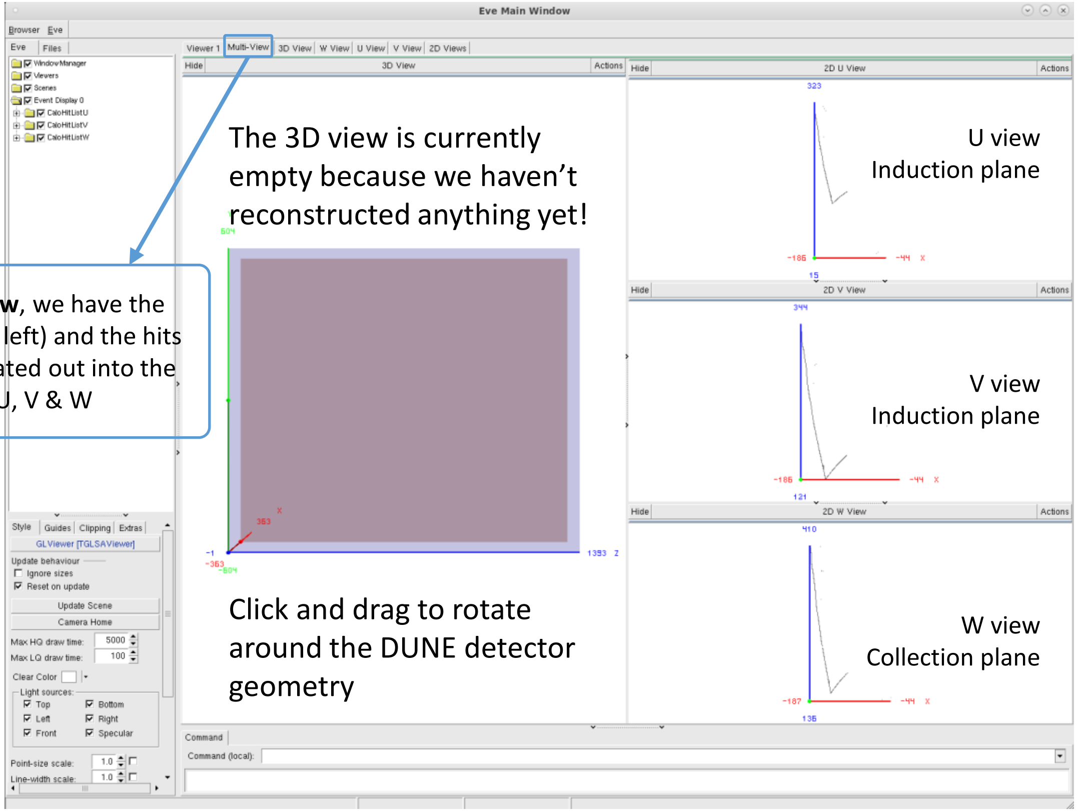
Point-size
Line-width

In **Viewer 1**, all information we visualize is overlaid. Here we see hits from all three views on top of each other + the detector geometry

You can safely ignore these options from TEve we won't use them here
Feel free to shrink down these menus for more space

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

Looking at the input hits – Multi-View



☐ Ignore sizes
☑ Reset on update

Max HQ draw time: 5000
Max LQ draw time: 100
Clear Color

☑ Top ☑ Bottom
☑ Left ☑ Right
☑ Front ☑ Specular

Point-size scale: 1.0
Line-width scale: 1.0

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

The 3D view is currently empty because we haven't reconstructed anything yet!

Click and drag to rotate around the DUNE detector geometry

U view
Induction plane

V view
Induction plane

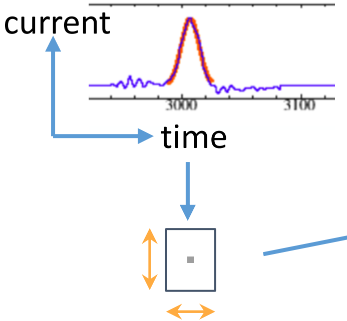
W view
Collection plane

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

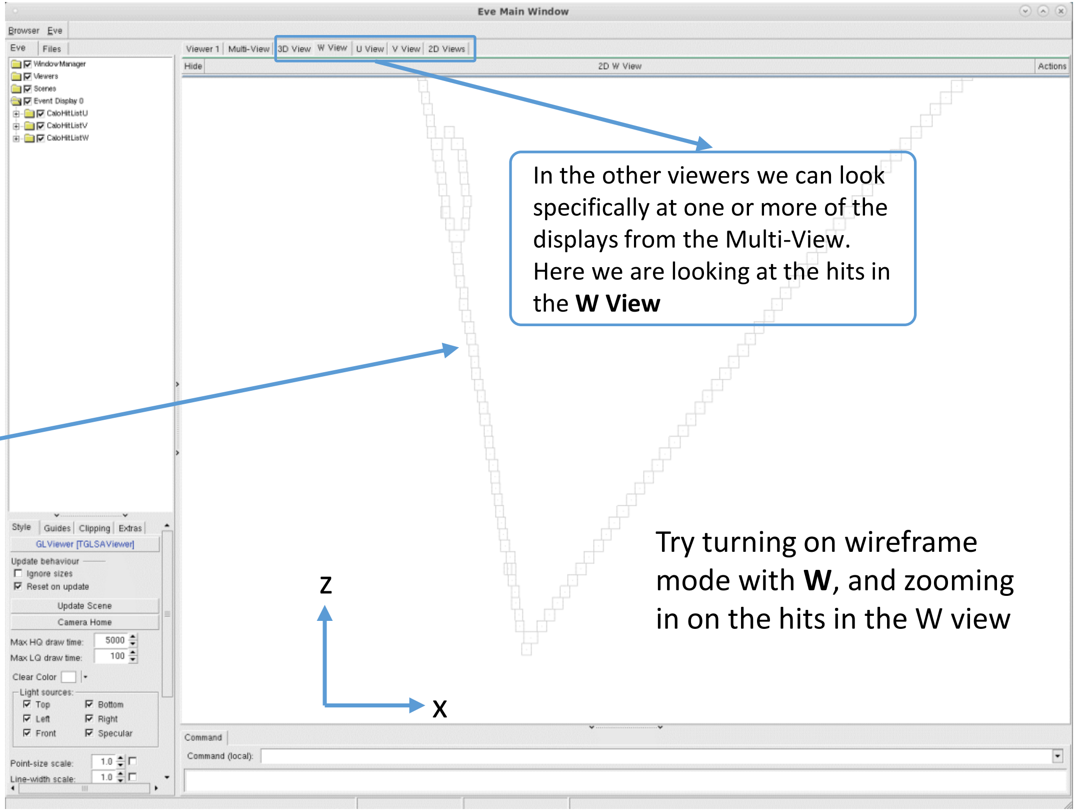
Left press + drag - rotate 3D view

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



Update behaviour
☐ Ignore sizes
☑ Reset on update

Max HQ draw time: 5000
Max LQ draw time: 100
Clear Color

Light sources:
☑ Top ☑ Bottom
☑ Left ☑ Right
☑ Front ☑ Specular

Point-size scale: 1.0 ☐
Line-width scale: 1.0 ☐

Command (local):

In the other viewers we can look specifically at one or more of the displays from the Multi-View. Here we are looking at the hits in the **W View**

Try turning on wireframe mode with **W**, and zooming in on the hits in the W view

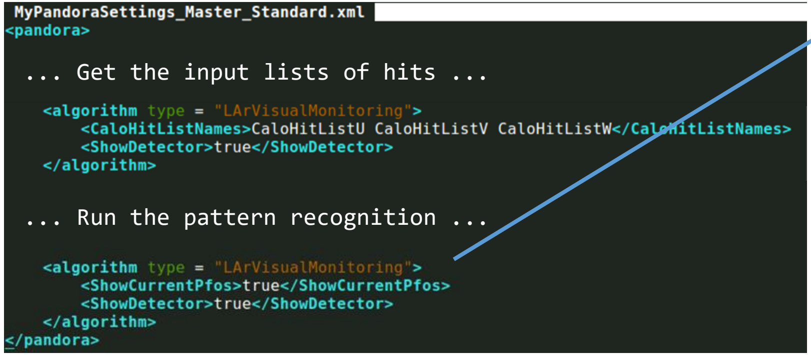
Wheel up - zoom out
Wheel down - zoom in
Wheel press + drag - pan viewport

Left press + drag - rotate 3D view

W - wireframe mode
R - return from wireframe mode

Looking at the final output of 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



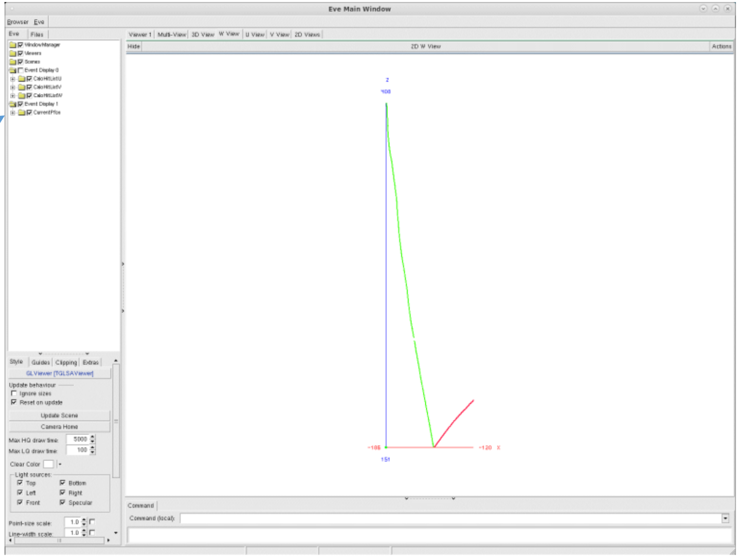
```
MyPandoraSettings_Master_Standard.xml
<pandora>

 ... 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>
```

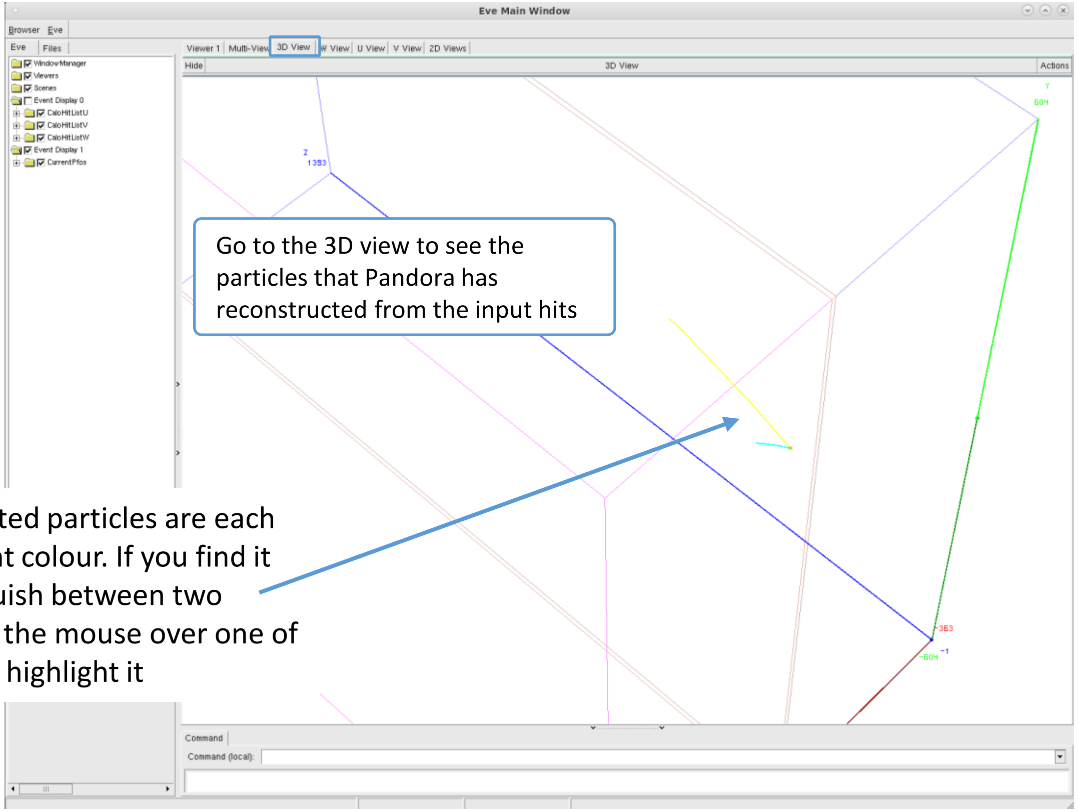


Looking at the reconstructed particles – 3D View

WARWICK

We’ve now moved on to the next visualization

Unfortunately, these checkboxes only work in Viewer 1



Go to the 3D view to see the particles that Pandora has reconstructed from the input hits

The reconstructed particles are each given a different colour. If you find it hard to distinguish between two colours - hover the mouse over one of the particles to highlight it

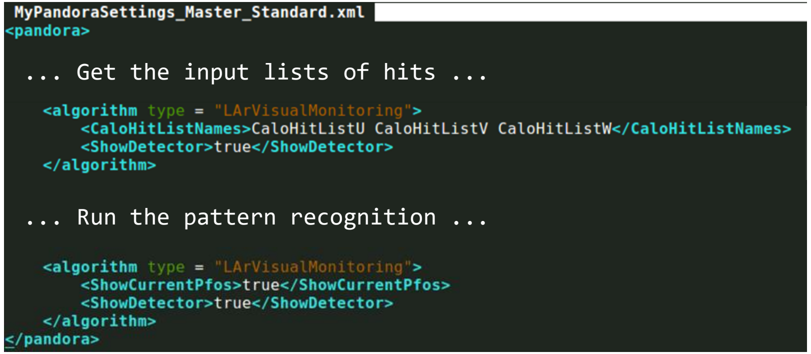
Wheel up - zoom out
Wheel down - zoom in
Wheel press + drag - pan viewport

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



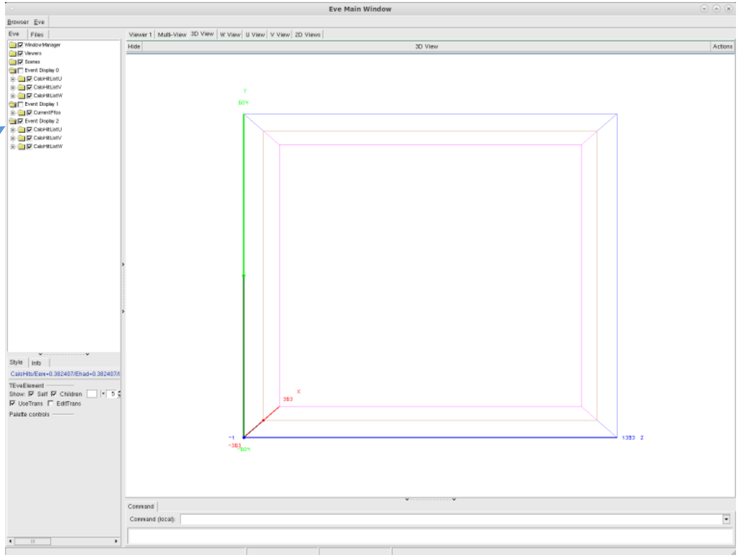
```
MyPandoraSettings_Master_Standard.xml
<pandora>

 ... 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>
```



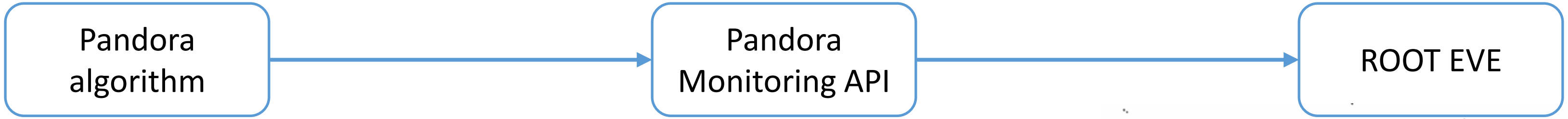
Got spare time?

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?

Additional information

Making visualizations within Pandora

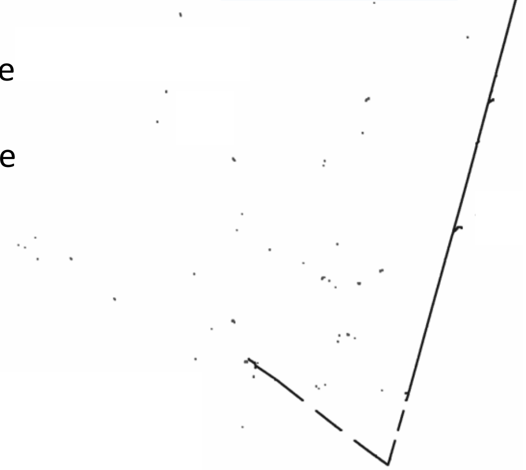
- Event displays are invaluable tools & a number of different options exist
- Today we will be focussing on the event display provided by Pandora


Pandora algorithm → Pandora Monitoring API → ROOT EVE

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

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

The Pandora Monitoring API is the provides the services that allow Pandora algorithms to easily make displays using ROOT's event visualization environment, EVE



Pandora Monitoring API & Visual Monitoring Alg

- 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

- 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