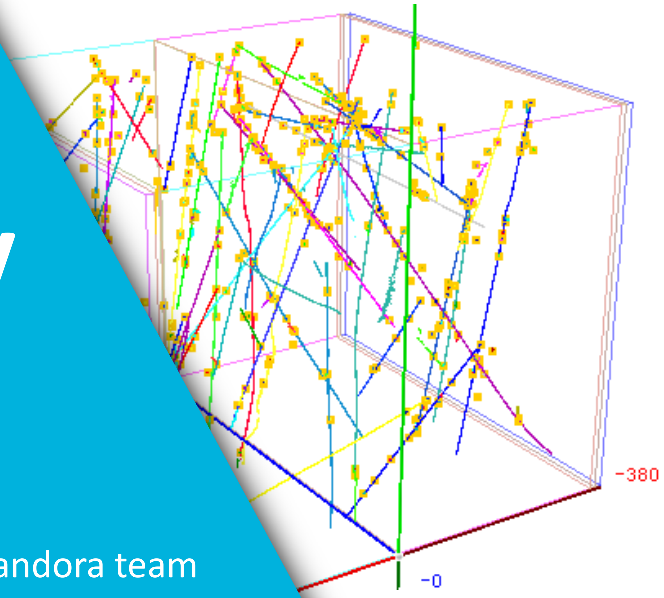
Pandora event display

Visualizing the algorithms

(Exercise)

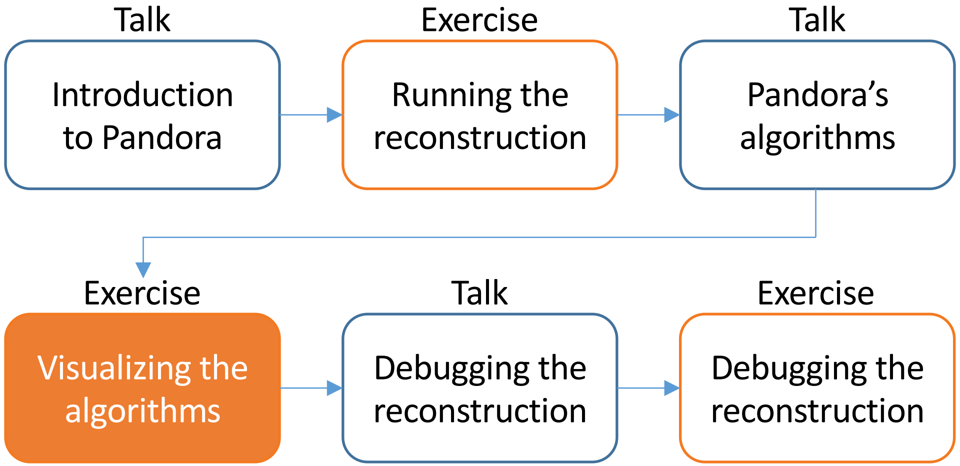


Maria Brigida Brunetti, Andy Chappell and Steve Dennis for the Pandora team

25/10/2023

8th UK LArTPC Software and Analysis Workshop

Reconstruction session



Talk

Introduction to Pandora

Exercise

Running the reconstruction

Talk

Pandora's algorithms

Exercise

Visualizing the algorithms

Talk

Debugging the reconstruction

Exercise

Debugging the reconstruction

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 30 minutes
- Main goal - Visualize the status of the pattern-recognition after each main stage
  - Enable visual monitoring in the Pandora configuration XML file
  - Re-run Pandora to start the GUI and see the input hits
  - Get to grips with the GUI
  - Add the visual monitoring algorithm to the Pandora configuration XML file after running the:
    - 2D reconstruction
    - 3D vertex reconstruction
    - Track & Shower reconstruction & particle refinement
    - 3D hit reconstruction
    - Neutrino hierarchy reconstruction
- Please don’t worry if you don’t get through all of the steps, there are a lot of them
  - This session is just for you to get some intuition for what Pandora’s algorithms do
  - And, importantly, to understand how to look at intermediate reconstruction state

Visualize the input hits in Pandora

Modifying the Pandora XML

- Make a copy of PandoraSettings_Master_Standard.xml. We will edit this to enable monitoring

```
$ mkdir -p $MRB_TOP/reco/config
$ 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 (you might already have this in a setup script) and do the same for the FHICL_FILE_PATH:

```
$ export FW_SEARCH_PATH=$MRB_TOP/reco/config:$FW_SEARCH_PATH
$ export FHICL_FILE_PATH=$MRB_TOP/reco/config:$FHICL_FILE_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_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: []
```

Inherit most reco settings

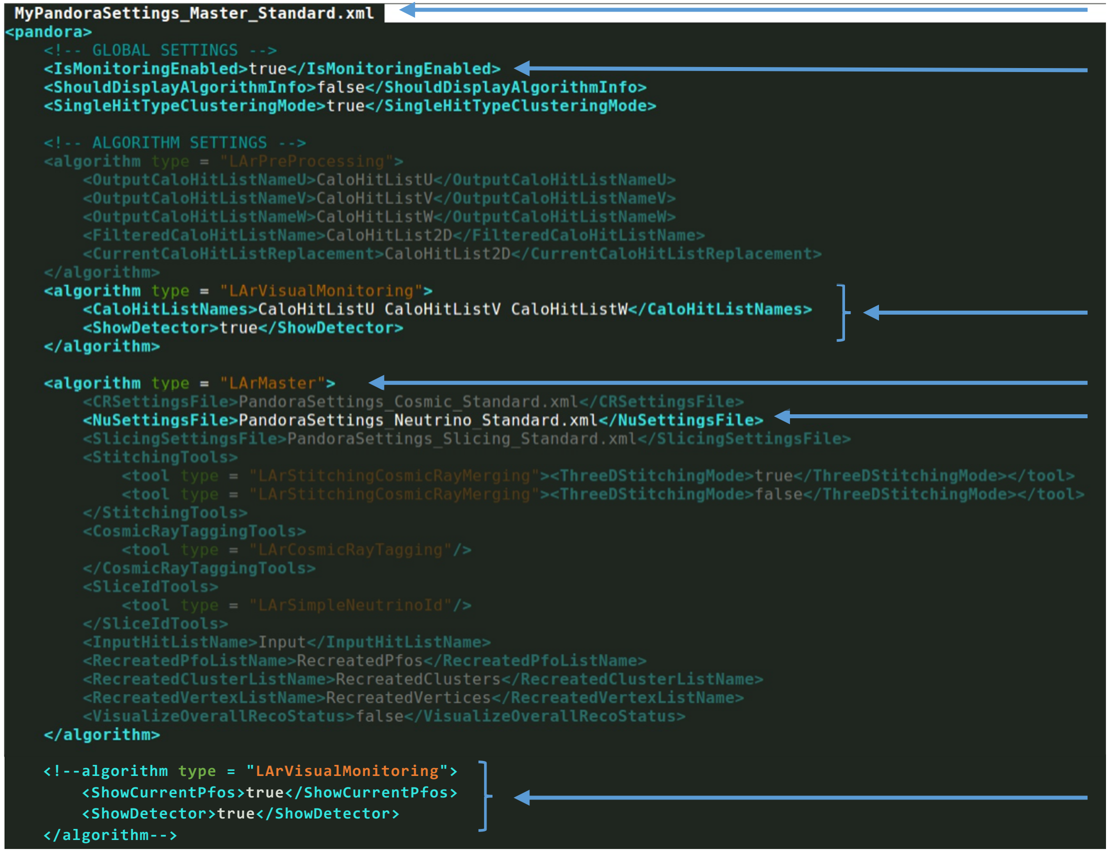
Rename the process

Point to our new XML settings file

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-->
```

Open your custom Pandora settings file

Enable visualizations

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 Pandora's pattern recognition - the neutrino algorithm chain is defined in: PandoraSettings_Neutrino_Standard.xml

Comment out <!-- ... -> the final visual monitoring algorithm for now, which show the reconstructed particles = PFOs

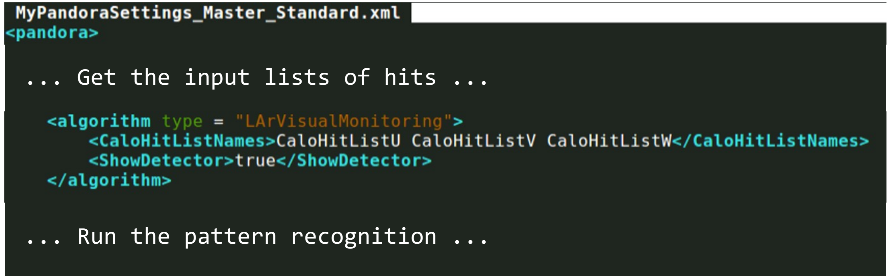
Running the event display

- 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

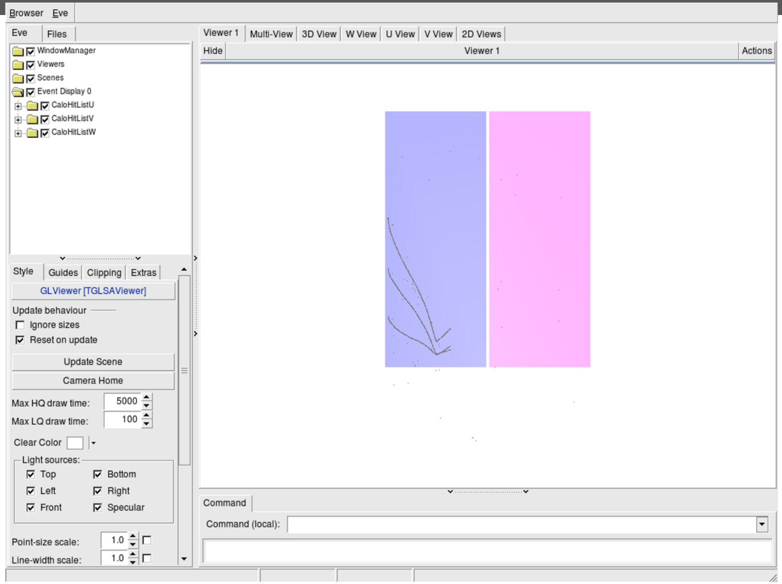
For now, let’s just look at 2 events.

```
$ cd $MRB_TOP/reco/work
$ lar -c event_display_driver.fcl -s reco_events.root -n 2
Can also run on pre-made reco files in /home/share/october2023/reconstruction
output_detsim_numu_like_Reco1Reco2.root
```

- 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 ...
```



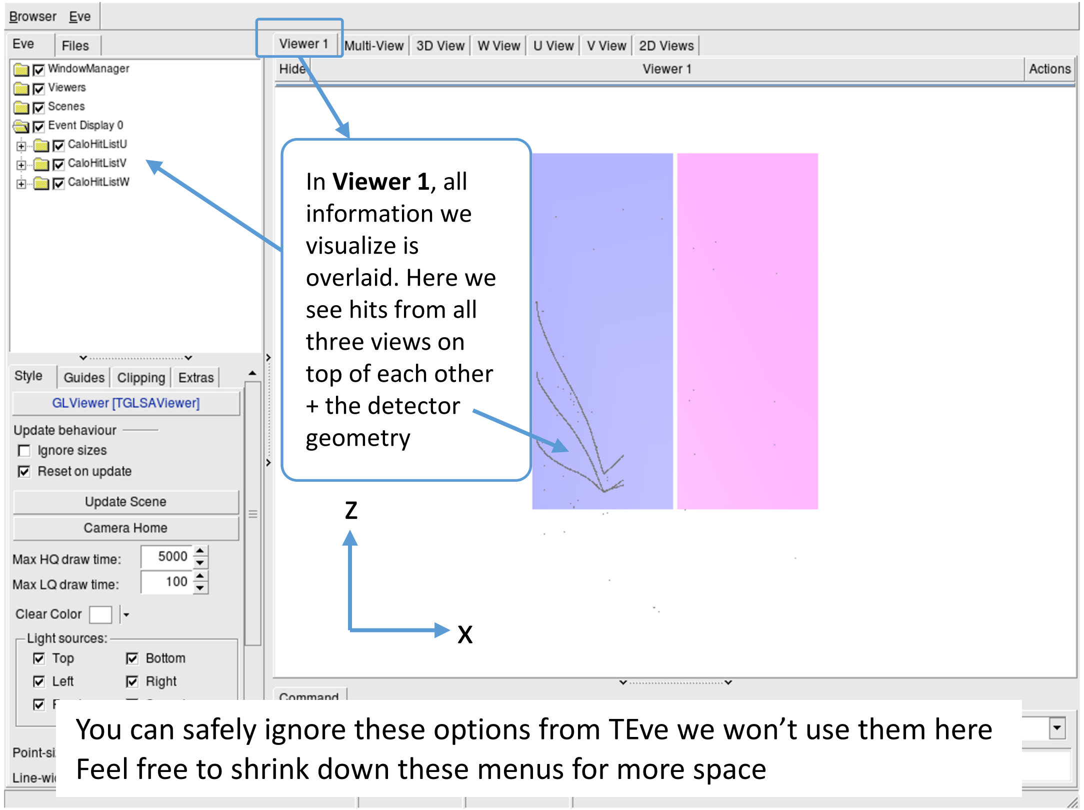
Looking at the input hits - Viewer 1

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 ☑ Bottom
☑ Left ☑ Right

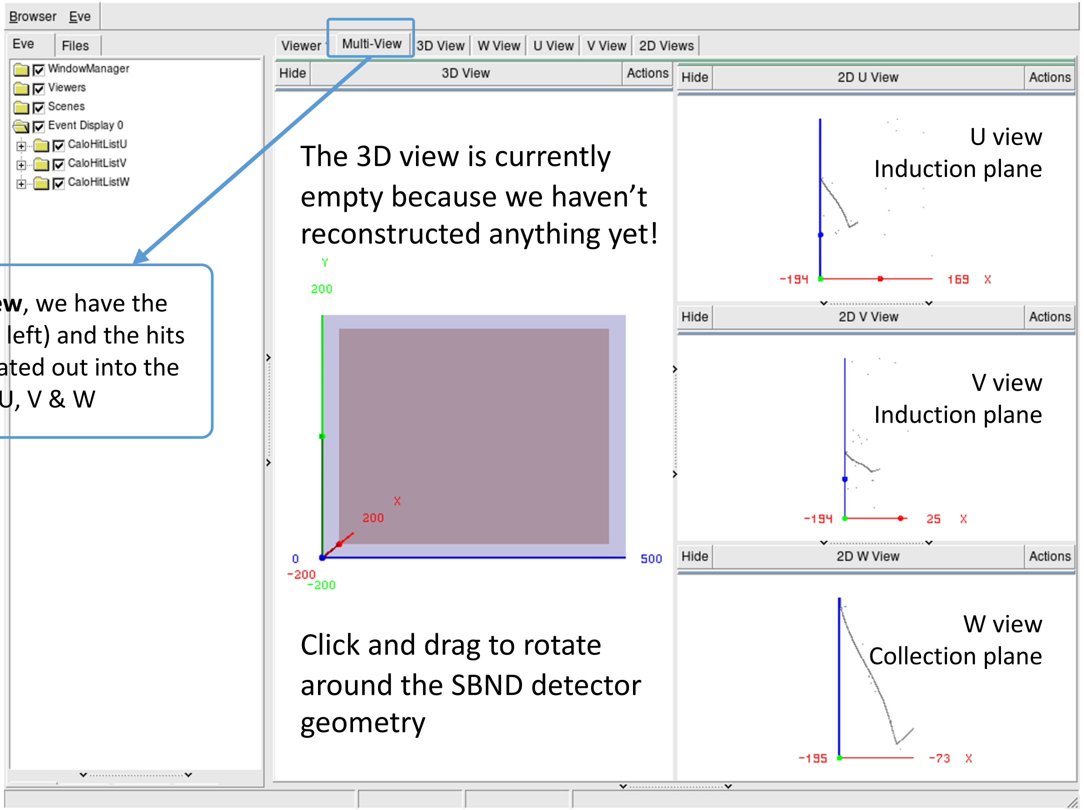
In **Viewer 1**, all information we visualize is overlaid. Here we see hits from all three views on top of each other + the detector geometry

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

If you are using a touchpad, can zoom in and out with two fingers, or use the + and – keys. To move the display you can use the arrow. keys.

Looking at the input hits – Multi-View



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

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 SBND detector geometry

U view
Induction plane

V view
Induction plane

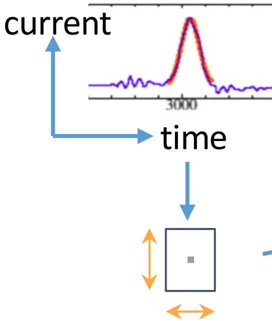
W view
Collection plane

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

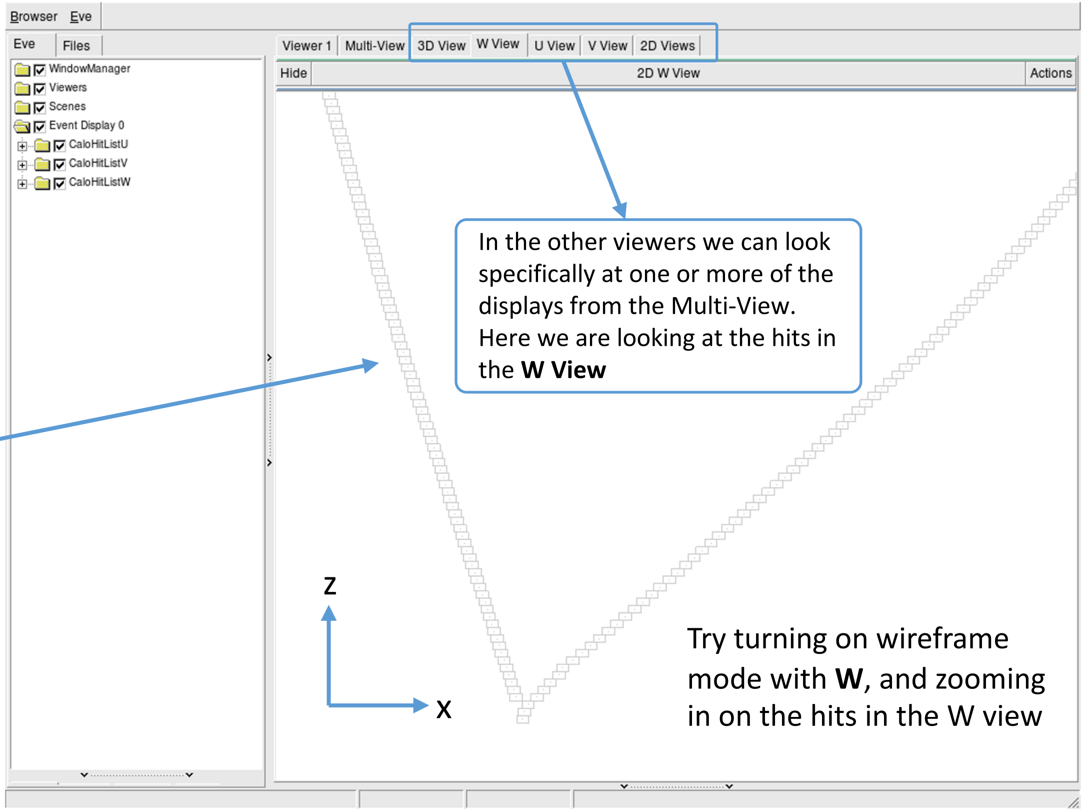
Left press + drag - rotate 3D view

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



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

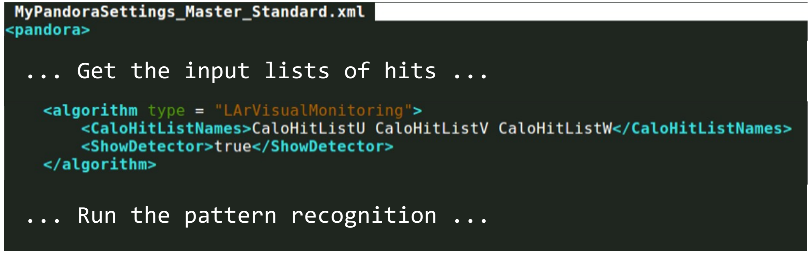
Wheel up - zoom out
Wheel down - zoom in
Wheel press + drag - pan viewport

Left press + drag - rotate 3D view

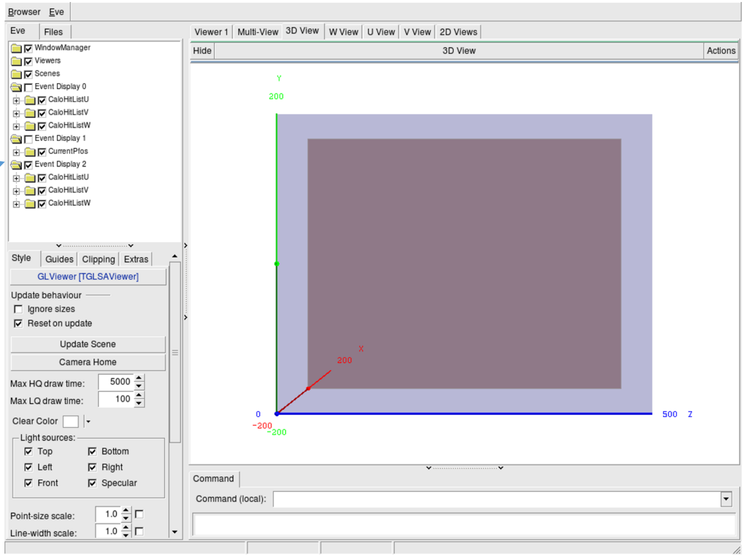
W - wireframe mode
R - return from wireframe mode

Moving on

- If you click in the terminal window and press Return ↲ Pandora will continue running
- This will exit from the current visual monitoring algorithm and either display the next visualisation, or move on to the next event, and begin producing the equivalent set of visualisations
- For now, press Return ↲ until you reach the final event and the display closes



```
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 ...
```



Visualize the algorithms

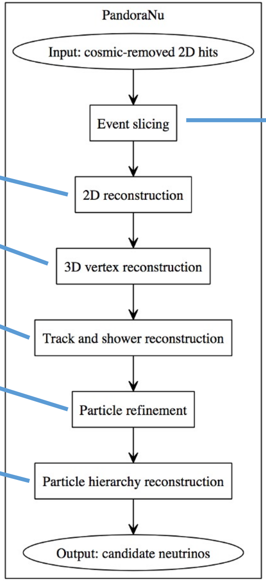
The neutrino algorithm chain

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

```
$ cd $MRB_TOP/reco/config
$ cp $LARPANDORA_DIR/scripts/PandoraSettings_Neutrino_Standard.xml MyPandoraSettings_Neutrino_Standard.xml
$ vim MyPandoraSettings_Neutrino_Standard.xml
```

- Look through the file for the sections listed below:

```
<!-- TwoDReconstruction -->
<!-- VertexAlgorithms -->
<!-- ThreeDTrackAlgorithms -->
<!-- ThreeDShowerAlgorithms -->
<!-- Repeat ThreeDTrackAlgorithms -->
<!-- ThreeDRecoveryAlgorithms -->
<!-- TwoDMopUpAlgorithms -->
<!-- ThreeDHitAlgorithms -->
<!-- ThreeDMopUpAlgorithms -->
<!-- NeutrinoAlgorithms -->
<!-- Track and shower building -->
```


PandoraNu
Input: cosmic-removed 2D hits
Event slicing
2D reconstruction
3D vertex reconstruction
Track and shower reconstruction
Particle refinement
Particle hierarchy reconstruction
Output: candidate neutrinos

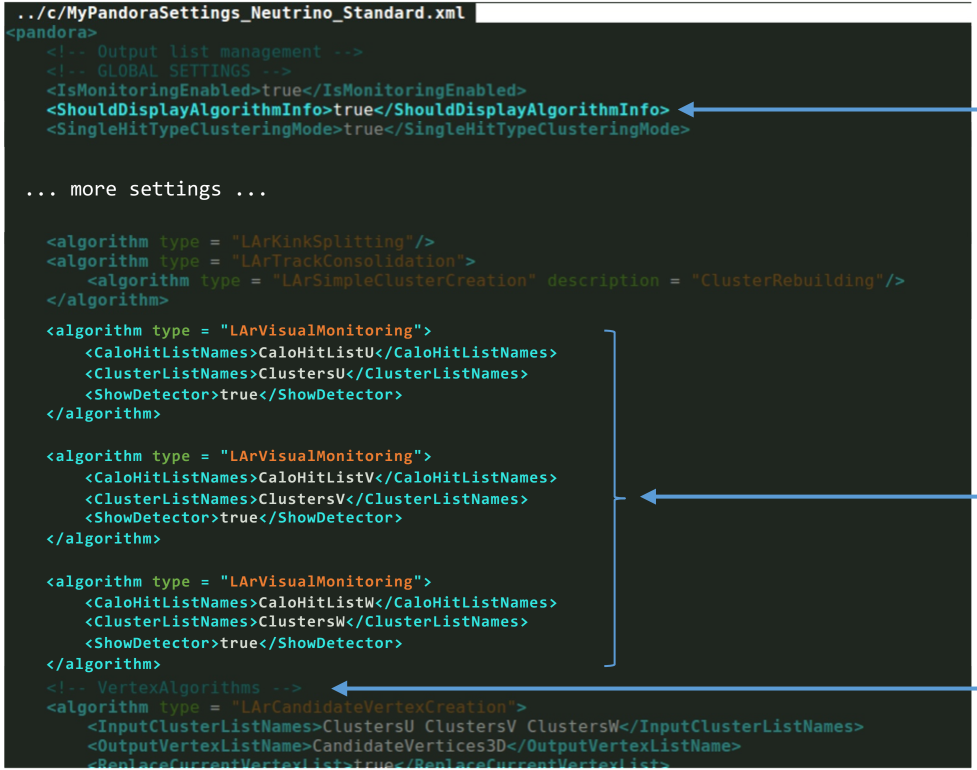
We're not running the event slicing because we don't have cosmics to deal with

- We'll be looking to visualize the reconstruction status at various points in this chain

2D reconstruction

Add in some visualizations

- Add to MyPandoraSettings_Neutrino_Standard.xml at the end of the TwoDReconstruction section



```
 ../c/MyPandoraSettings_Neutrino_Standard.xml
<pandora>
    <!-- Output list management -->
    <!-- GLOBAL SETTINGS -->
    <IsMonitoringEnabled>true</IsMonitoringEnabled>
    <ShouldDisplayAlgorithmInfo>true</ShouldDisplayAlgorithmInfo>
    <SingleHitTypeClusteringMode>true</SingleHitTypeClusteringMode>


 ... more settings ...

    <algorithm type = "LArKinkSplitting"/>
    <algorithm type = "LArTrackConsolidation">
        <algorithm type = "LArSimpleClusterCreation" description = "ClusterRebuilding"/>
    </algorithm>

    <algorithm type = "LArVisualMonitoring">
        <CaloHitListNames>CaloHitListU</CaloHitListNames>
        <ClusterListNames>ClustersU</ClusterListNames>
        <ShowDetector>true</ShowDetector>
    </algorithm>

    <algorithm type = "LArVisualMonitoring">
        <CaloHitListNames>CaloHitListV</CaloHitListNames>
        <ClusterListNames>ClustersV</ClusterListNames>
        <ShowDetector>true</ShowDetector>
    </algorithm>

    <algorithm type = "LArVisualMonitoring">
        <CaloHitListNames>CaloHitListW</CaloHitListNames>
        <ClusterListNames>ClustersW</ClusterListNames>
        <ShowDetector>true</ShowDetector>
    </algorithm>
    <!-- VertexAlgorithms -->
    <algorithm type = "LArCandidateVertexCreation">
        <InputClusterListNames>ClustersU ClustersV ClustersW</InputClusterListNames>
        <OutputVertexListName>CandidateVertices3D</OutputVertexListName>
        <ReplaceCurrentVertexList>true</ReplaceCurrentVertexList>
```

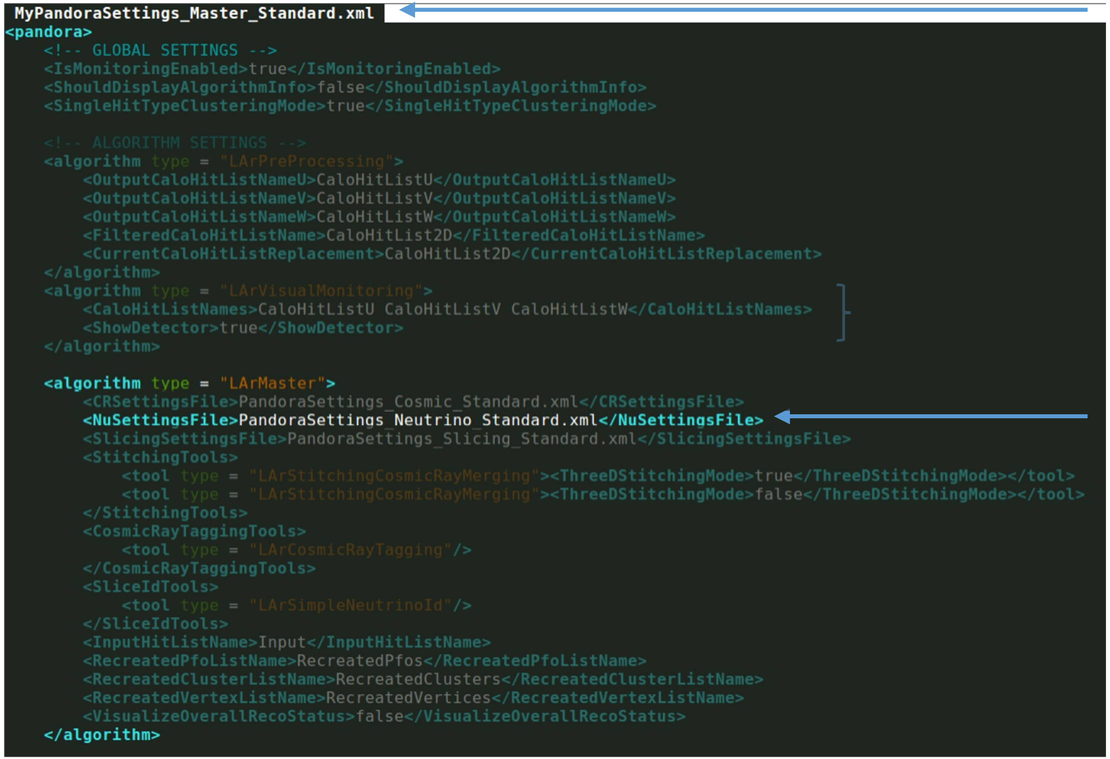
Set this to true - this will print to the terminal all of the algorithms we are running

Modify the **Neutrino** file not the **Master** settings file

Add these visual monitoring blocks. When we run, this will make 3 event displays - each showing the hits and clusters in the U, V and W views respectively

Add the above lines just before the VertexAlgorithms section

Updating the Master configuration



```
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>
```

Open your custom Pandora Master settings file

Now that we also have a custom Neutrino settings file, we need to ensure that we use it

The master algorithm is in charge of running the different steps of the Pandora’s pattern recognition.

Currently it uses the neutrino configuration in:
PandoraSettings_Neutrino_Standard.xml

You should change this to:
MyPandoraSettings_Neutrino_Standard.xml

Visualizing the initial 2D reconstruction

```
$ cd $MRB_TOP/reco/work
$ lar -c event_display_driver.fcl -s reco_events.root -n 1
```

Let’s just look at 1 event for now!

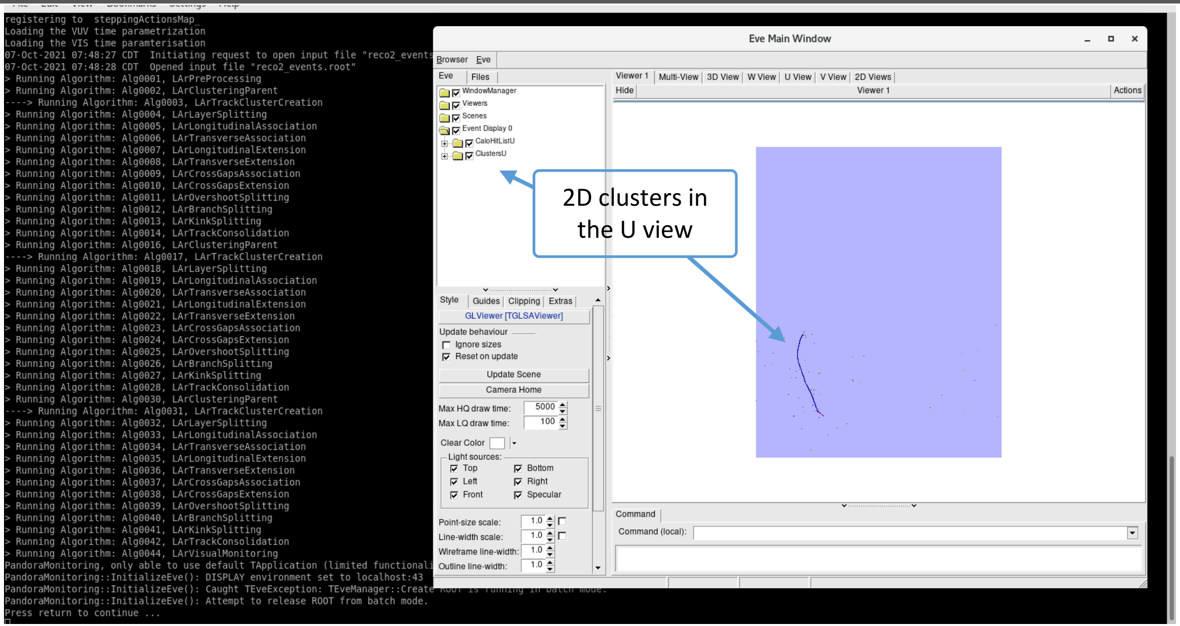
Can also run on pre-made reco file in `/home/share/october2023/reconstruction/output_detsim_numu_like_Reco1Reco2.root`

2D clustering algorithms in the U-view

2D clustering algorithms in the V-view

2D clustering algorithms in the U-view

First visualization



```
registering to  steppingActionsMap_
Loading the VUV time parametrization
Loading the VIS time paramterisation
07-Oct-2021 07:48:27 CDT  Initiating request to open input file "reco2_events
07-Oct-2021 07:48:28 CDT  Opened input file "reco2_events.root"
> Running Algorithm: Alg0001, LArPreProcessing
> Running Algorithm: Alg0002, LArClusteringParent
----> Running Algorithm: Alg0003, LArTrackClusterCreation
> Running Algorithm: Alg0004, LArLayerSplitting
> Running Algorithm: Alg0005, LArLongitudinalAssociation
> Running Algorithm: Alg0006, LArTransverseAssociation
> Running Algorithm: Alg0007, LArLongitudinalExtension
> Running Algorithm: Alg0008, LArTransverseExtension
> Running Algorithm: Alg0009, LArCrossGapsAssociation
> Running Algorithm: Alg0010, LArCrossGapsExtension
> Running Algorithm: Alg0011, LArOvershootSplitting
> Running Algorithm: Alg0012, LArBranchSplitting
> Running Algorithm: Alg0013, LArKinkSplitting
> Running Algorithm: Alg0014, LArTrackConsolidation
> Running Algorithm: Alg0016, LArClusteringParent
----> Running Algorithm: Alg0017, LArTrackClusterCreation
> Running Algorithm: Alg0018, LArLayerSplitting
> Running Algorithm: Alg0019, LArLongitudinalAssociation
> Running Algorithm: Alg0020, LArTransverseAssociation
> Running Algorithm: Alg0021, LArLongitudinalExtension
> Running Algorithm: Alg0022, LArTransverseExtension
> Running Algorithm: Alg0023, LArCrossGapsAssociation
> Running Algorithm: Alg0024, LArCrossGapsExtension
> Running Algorithm: Alg0025, LArOvershootSplitting
> Running Algorithm: Alg0026, LArBranchSplitting
> Running Algorithm: Alg0027, LArKinkSplitting
> Running Algorithm: Alg0028, LArTrackConsolidation
> Running Algorithm: Alg0030, LArClusteringParent
----> Running Algorithm: Alg0031, LArTrackClusterCreation
> Running Algorithm: Alg0032, LArLayerSplitting
> Running Algorithm: Alg0033, LArLongitudinalAssociation
> Running Algorithm: Alg0034, LArTransverseAssociation
> Running Algorithm: Alg0035, LArLongitudinalExtension
> Running Algorithm: Alg0036, LArTransverseExtension
> Running Algorithm: Alg0037, LArCrossGapsAssociation
> Running Algorithm: Alg0038, LArCrossGapsExtension
> Running Algorithm: Alg0039, LArOvershootSplitting
> Running Algorithm: Alg0040, LArBranchSplitting
> Running Algorithm: Alg0041, LArKinkSplitting
> Running Algorithm: Alg0042, LArTrackConsolidation
> Running Algorithm: Alg0044, LArVisualMonitoring
PandoraMonitoring, only able to use default TApplication (limited functionali
PandoraMonitoring::InitializeEve(): DISPLAY environment set to localhost:43
PandoraMonitoring::InitializeEve(): Caught TEveException: TEveManager::Create ROOT is running in batch mode.
PandoraMonitoring::InitializeEve(): Attempt to release ROOT from batch mode.
Press return to continue ...
```

2D clusters in the U view

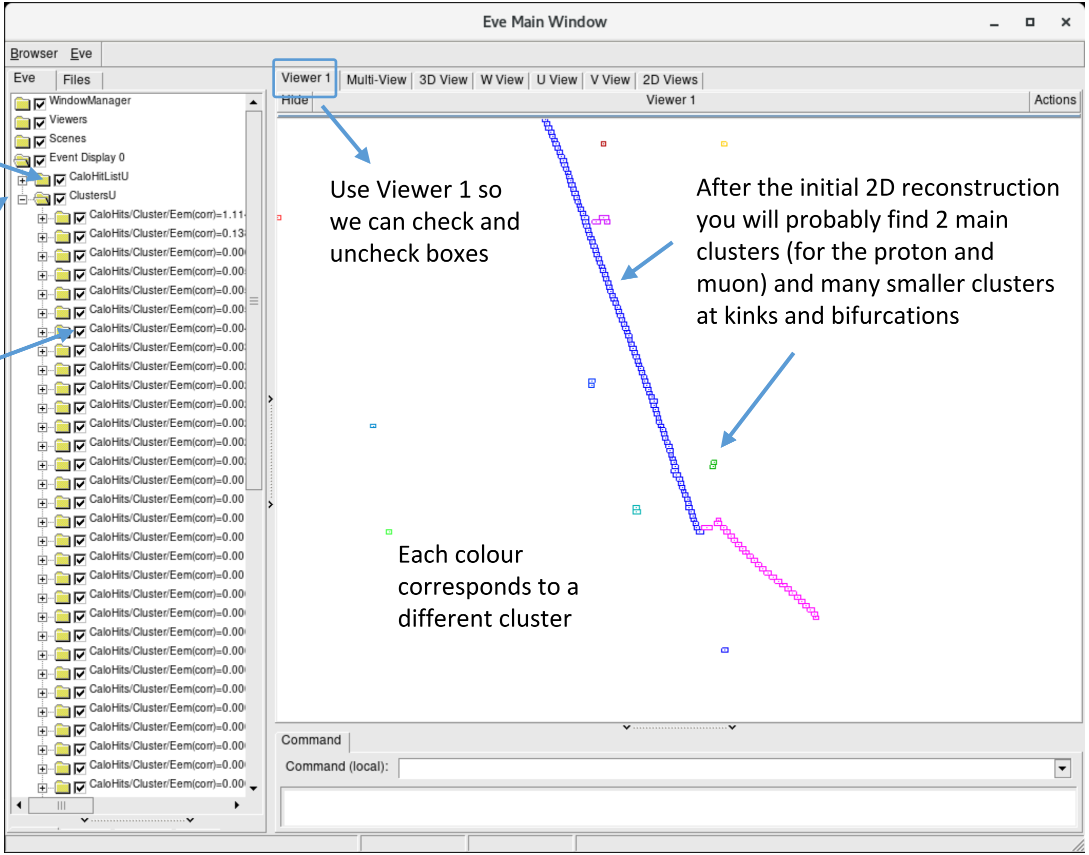
Looking at the reconstructed particles

Turn off the hits, we’ve included them so you can always refer back to the inputs if you like

Expand the list of clusters

Try turning on and off some of the clusters so you can see what they correspond to in the viewer

Clusters are ordered by the total energy deposited


Eve Main Window
Browser
Eve
Eve
Files
WindowManager
Viewers
Scenes
Event Display 0
CaloHitListU
ClustersU
CaloHits/Cluster/Eem(corr)=1.11
CaloHits/Cluster/Eem(corr)=0.13
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
CaloHits/Cluster/Eem(corr)=0.00
Viewer 1
Multi-View
3D View
W View
U View
V View
2D Views
Hide
Viewer 1
Actions
Command
Command (local):

Use Viewer 1 so we can check and uncheck boxes

After the initial 2D reconstruction you will probably find 2 main clusters (for the proton and muon) and many smaller clusters at kinks and bifurcations

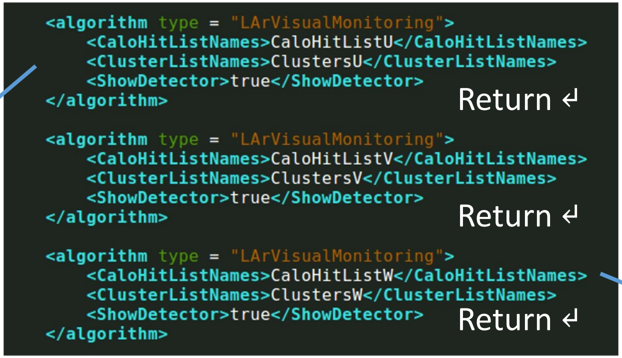
Each colour corresponds to a different cluster

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

W - wireframe mode
R - return from wireframe mode

Looking at the other views

Click in the terminal window and press Return ↵ to visualize the other views

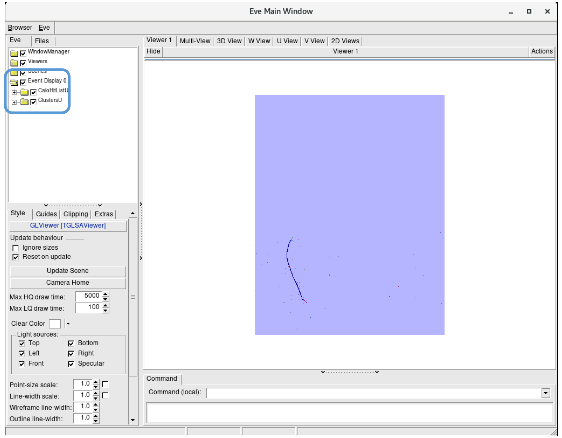


```
<algorithm type = "LArVisualMonitoring">
    <CaloHitListNames>CaloHitListU</CaloHitListNames>
    <ClusterListNames>ClustersU</ClusterListNames>
    <ShowDetector>true</ShowDetector>
</algorithm>

<algorithm type = "LArVisualMonitoring">
    <CaloHitListNames>CaloHitListV</CaloHitListNames>
    <ClusterListNames>ClustersV</ClusterListNames>
    <ShowDetector>true</ShowDetector>
</algorithm>

<algorithm type = "LArVisualMonitoring">
    <CaloHitListNames>CaloHitListW</CaloHitListNames>
    <ClusterListNames>ClustersW</ClusterListNames>
    <ShowDetector>true</ShowDetector>
</algorithm>
```

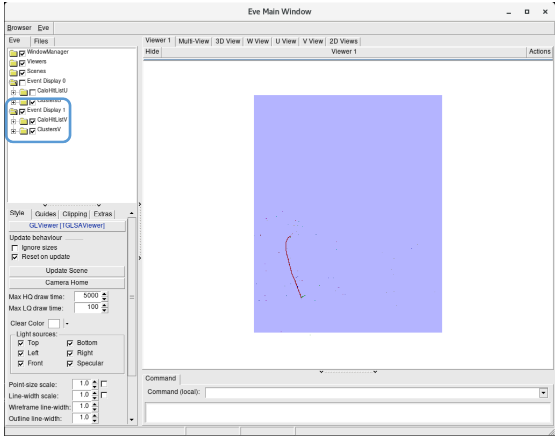
Return ↵

Return ↵

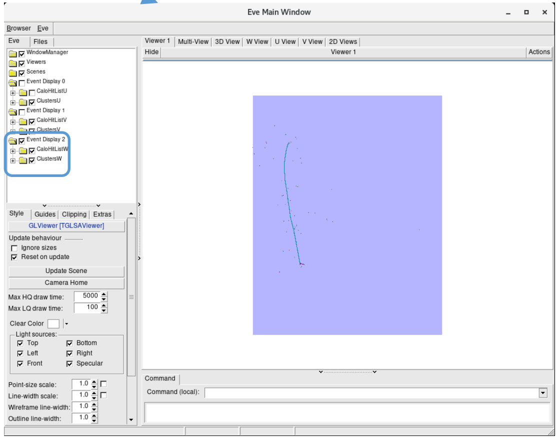
Return ↵



U view clusters



V view clusters

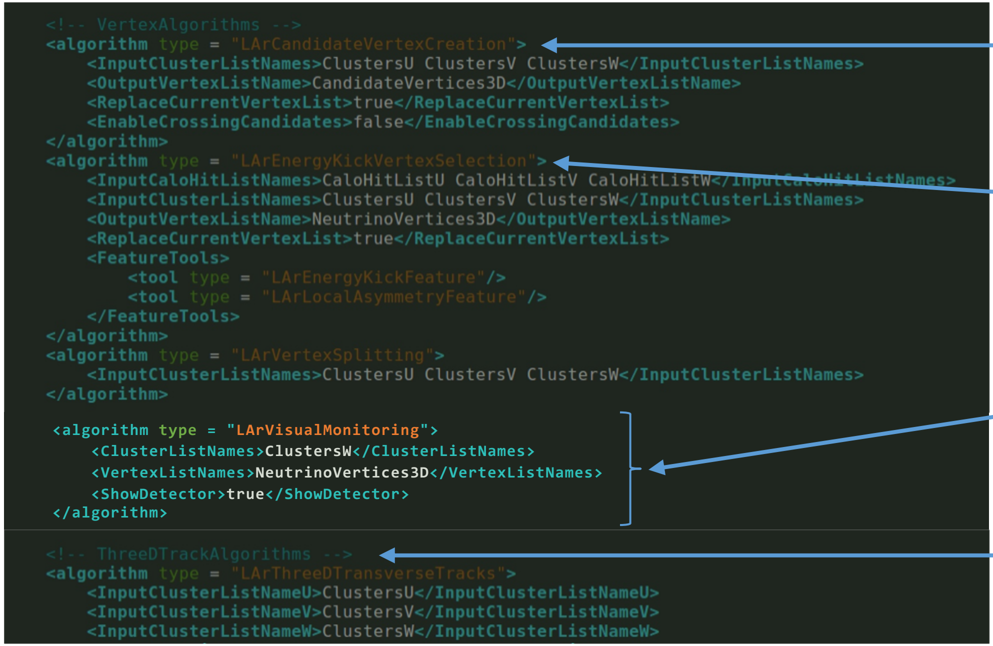


W view clusters

3D vertex reconstruction

Add in some more visualizations

- Add to MyPandoraSettings_Neutrino_Standard.xml at the end of the VertexAlgorithms section



```
<!-- VertexAlgorithms -->
<algorithm type = "LArCandidateVertexCreation">
    <InputClusterListNames>ClustersU ClustersV ClustersW</InputClusterListNames>
    <OutputVertexListName>CandidateVertices3D</OutputVertexListName>
    <ReplaceCurrentVertexList>true</ReplaceCurrentVertexList>
    <EnableCrossingCandidates>false</EnableCrossingCandidates>
</algorithm>
<algorithm type = "LArEnergyKickVertexSelection">
    <InputCaloHitListNames>CaloHitListU CaloHitListV CaloHitListW</InputCaloHitListNames>
    <InputClusterListNames>ClustersU ClustersV ClustersW</InputClusterListNames>
    <OutputVertexListName>NeutrinoVertices3D</OutputVertexListName>
    <ReplaceCurrentVertexList>true</ReplaceCurrentVertexList>
    <FeatureTools>
        <tool type = "LArEnergyKickFeature"/>
        <tool type = "LArLocalAsymmetryFeature"/>
    </FeatureTools>
</algorithm>
<algorithm type = "LArVertexSplitting">
    <InputClusterListNames>ClustersU ClustersV ClustersW</InputClusterListNames>
</algorithm>

 <algorithm type = "LArVisualMonitoring">
     <ClusterListNames>ClustersW</ClusterListNames>
     <VertexListNames>NeutrinoVertices3D</VertexListNames>
     <ShowDetector>true</ShowDetector>
 </algorithm>

<!-- ThreeDTrackAlgorithms -->
<algorithm type = "LArThreeDTransverseTracks">
    <InputClusterListNameU>ClustersU</InputClusterListNameU>
    <InputClusterListNameV>ClustersV</InputClusterListNameV>
    <InputClusterListNameW>ClustersW</InputClusterListNameW>
```

The LArCandidateVertexCreation algorithm creates a list of 3D candidate vertices at positions that project onto the ends of the existing 2D clusters

The LArEnergyKickVertexSelection algorithm selects the neutrino vertex from the candidates

Visualize the selected neutrino vertex along with the W-view clusters for comparison

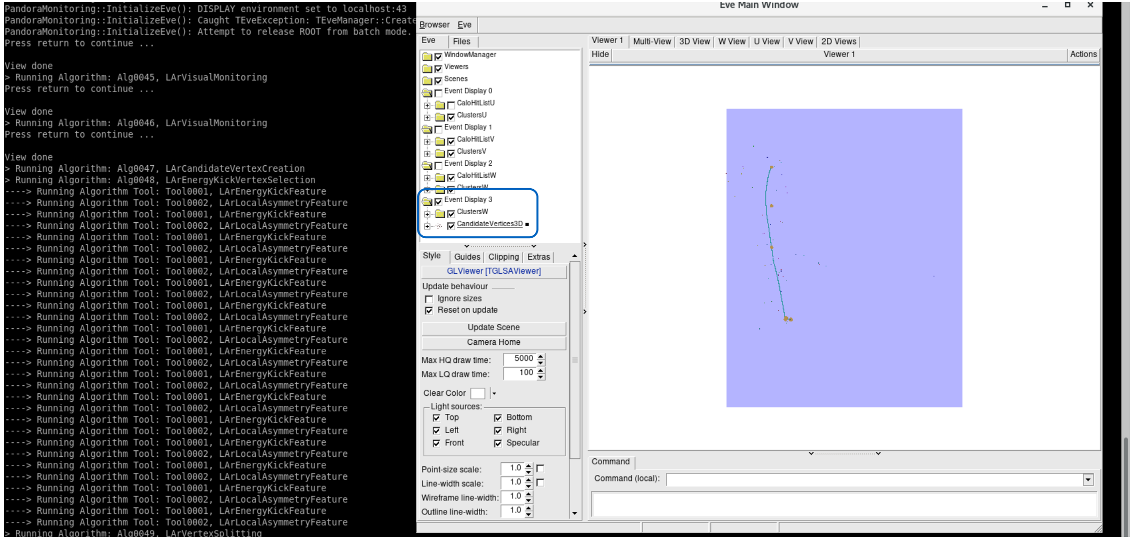
Add the above lines just before the ThreeDTrackAlgorithms section

Run Pandora again!

- Run our FHiCl file again

```
$ cd $MRB_TOP/reco/work
$ lar -c event_display_driver.fcl -s reco_events.root -n 1
```

- After the event display has loaded press Return ↲ three times, to skip through our visualizations from part 1

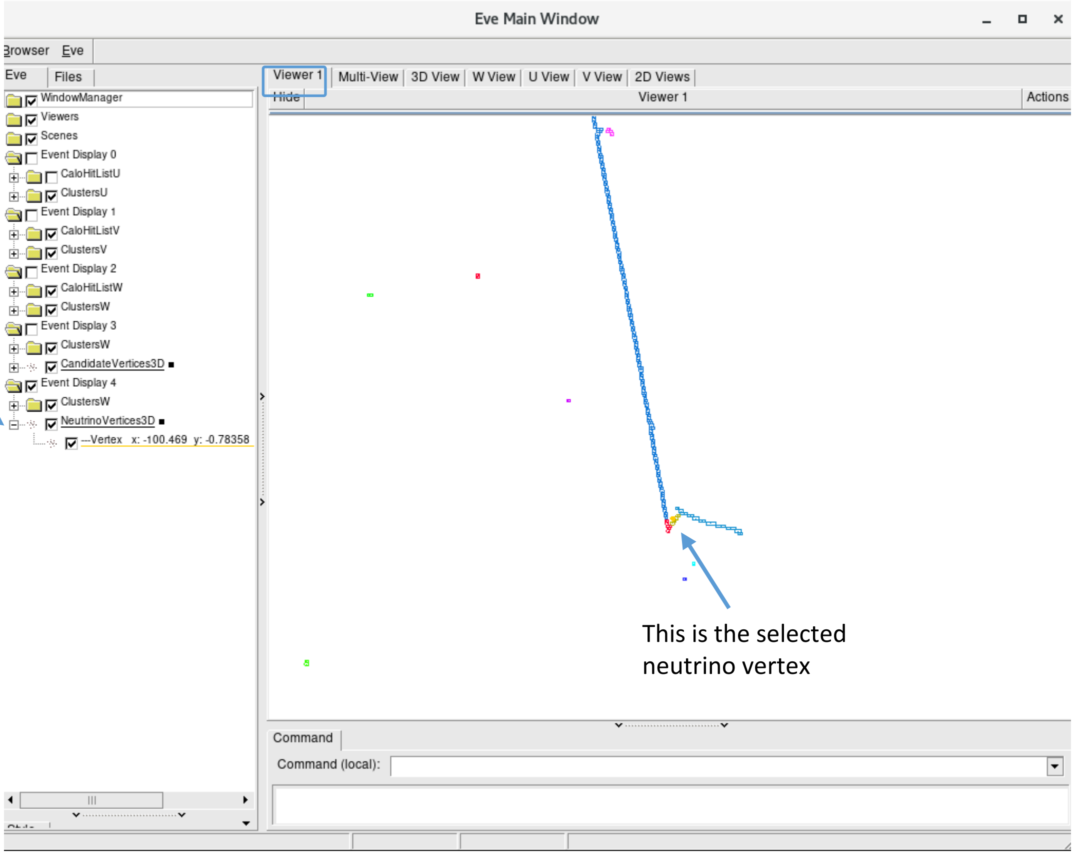


```
PandoraMonitoring::InitializeEve(): DISPLAY environment set to localhost:43
PandoraMonitoring::InitializeEve(): Caught TEveException: TEveManager::Create
PandoraMonitoring::InitializeEve(): Attempt to release ROOT from batch mode.
Press return to continue ...

View done
> Running Algorithm: Alg0045, LArVisualMonitoring
Press return to continue ...

View done
> Running Algorithm: Alg0046, LArVisualMonitoring
Press return to continue ...

View done
> Running Algorithm: Alg0047, LArCandidateVertexCreation
> Running Algorithm: Alg0048, LArEnergyKickVertexSelection
----> Running Algorithm Tool: Tool0001, LArEnergyKickFeature
----> Running Algorithm Tool: Tool0002, LArLocalAsymmetryFeature
----> Running Algorithm Tool: Tool0001, LArEnergyKickFeature
----> Running Algorithm Tool: Tool0002, LArLocalAsymmetryFeature
----> Running Algorithm Tool: Tool0001, LArEnergyKickFeature
----> Running Algorithm Tool: Tool0002, LArLocalAsymmetryFeature
----> Running Algorithm Tool: Tool0001, LArEnergyKickFeature
----> Running Algorithm Tool: Tool0002, LArLocalAsymmetryFeature
----> Running Algorithm Tool: Tool0001, LArEnergyKickFeature
----> Running Algorithm Tool: Tool0002, LArLocalAsymmetryFeature
----> Running Algorithm Tool: Tool0001, LArEnergyKickFeature
----> Running Algorithm Tool: Tool0002, LArLocalAsymmetryFeature
----> Running Algorithm Tool: Tool0001, LArEnergyKickFeature
----> Running Algorithm Tool: Tool0002, LArLocalAsymmetryFeature
----> Running Algorithm Tool: Tool0001, LArEnergyKickFeature
----> Running Algorithm Tool: Tool0002, LArLocalAsymmetryFeature
----> Running Algorithm Tool: Tool0001, LArEnergyKickFeature
----> Running Algorithm Tool: Tool0002, LArLocalAsymmetryFeature
----> Running Algorithm Tool: Tool0001, LArEnergyKickFeature
----> Running Algorithm Tool: Tool0002, LArLocalAsymmetryFeature
----> Running Algorithm Tool: Tool0001, LArEnergyKickFeature
----> Running Algorithm Tool: Tool0002, LArLocalAsymmetryFeature
----> Running Algorithm Tool: Tool0001, LArEnergyKickFeature
----> Running Algorithm Tool: Tool0002, LArLocalAsymmetryFeature
----> Running Algorithm Tool: Tool0001, LArEnergyKickFeature
----> Running Algorithm Tool: Tool0002, LArLocalAsymmetryFeature
----> Running Algorithm Tool: Tool0001, LArEnergyKickFeature
----> Running Algorithm Tool: Tool0002, LArLocalAsymmetryFeature
----> Running Algorithm Tool: Tool0001, LArEnergyKickFeature
----> Running Algorithm Tool: Tool0002, LArLocalAsymmetryFeature
> Running Algorithm: Alg0049, LArVertexSplitting
```

Max HQ draw time: 5000
Max LQ draw time: 100
Clear Color
Point-size scale: 1.0
Line-width scale: 1.0
Wireframe line-width: 1.0
Outline line-width: 1.0

Selected neutrino vertex vs W-view clusters

Expand the list of selected neutrino vertices - there should only be one



- ☑ WindowManager
- ☑ Viewers
- ☑ Scenes
- □ Event Display 0
  - □ CaloHitListU
  - ☑ ClustersU
- □ Event Display 1
  - ☑ CaloHitListV
  - ☑ ClustersV
- □ Event Display 2
  - ☑ CaloHitListW
  - ☑ ClustersW
- □ Event Display 3
  - ☑ ClustersW
  - ☑ CandidateVertices3D ■
- ☑ Event Display 4
  - ☑ ClustersW
  - ☑ NeutrinoVertices3D ■
    - ☑ ---Vertex x: -100.469 y: -0.78358

This is the selected neutrino vertex

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

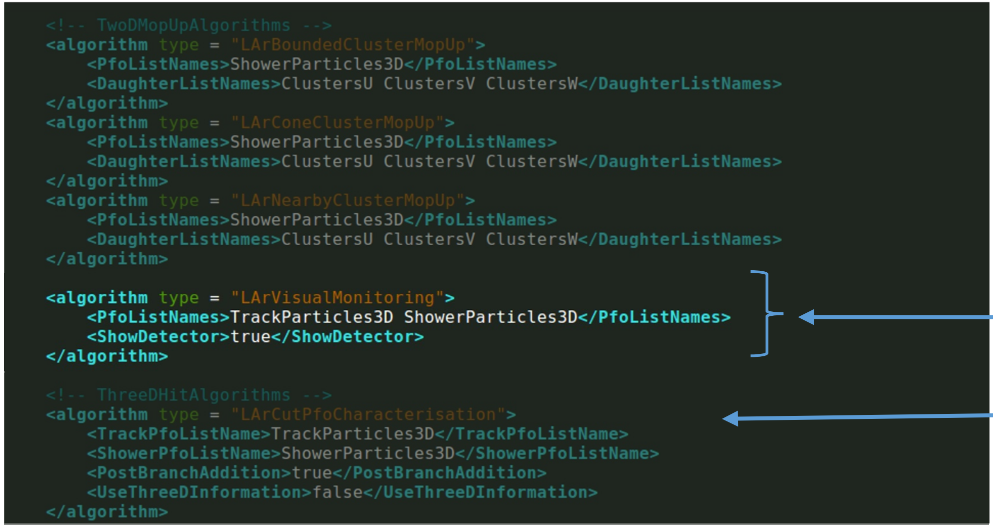
W - wireframe mode
R - return from wireframe mode

When you are finished, press Return ↲ to move to the next display

3D track & shower reconstruction

Add in some more visualizations

- Add to MyPandoraSettings_Neutrino_Standard.xml at the end of the TwoDMopUpAlgorithms section



```
<!-- TwoDMopUpAlgorithms -->
<algorithm type = "LArBoundedClusterMopUp">
    <PfoListNames>ShowerParticles3D</PfoListNames>
    <DaughterListNames>ClustersU ClustersV ClustersW</DaughterListNames>
</algorithm>
<algorithm type = "LArConeClusterMopUp">
    <PfoListNames>ShowerParticles3D</PfoListNames>
    <DaughterListNames>ClustersU ClustersV ClustersW</DaughterListNames>
</algorithm>
<algorithm type = "LArNearbyClusterMopUp">
    <PfoListNames>ShowerParticles3D</PfoListNames>
    <DaughterListNames>ClustersU ClustersV ClustersW</DaughterListNames>
</algorithm>

<algorithm type = "LArVisualMonitoring">
    <PfoListNames>TrackParticles3D ShowerParticles3D</PfoListNames>
    <ShowDetector>true</ShowDetector>
</algorithm>

<!-- ThreeDHitAlgorithms -->
<algorithm type = "LArCutPfoCharacterisation">
    <TrackPfoListName>TrackParticles3D</TrackPfoListName>
    <ShowerPfoListName>ShowerParticles3D</ShowerPfoListName>
    <PostBranchAddition>true</PostBranchAddition>
    <UseThreeDInformation>false</UseThreeDInformation>
</algorithm>
```

Visualize the track-like and shower-like reconstructed particles

Add the above lines just before the ThreeDHitAlgorithms section

Run Pandora once again!

```
$ cd $MRB_TOP/reco/work
$ lar -c event_display_driver.fcl -s reco_events.root -n 1
```

- After the event display has loaded press Return ↲ five times, to skip through our visualizations from parts 1-2

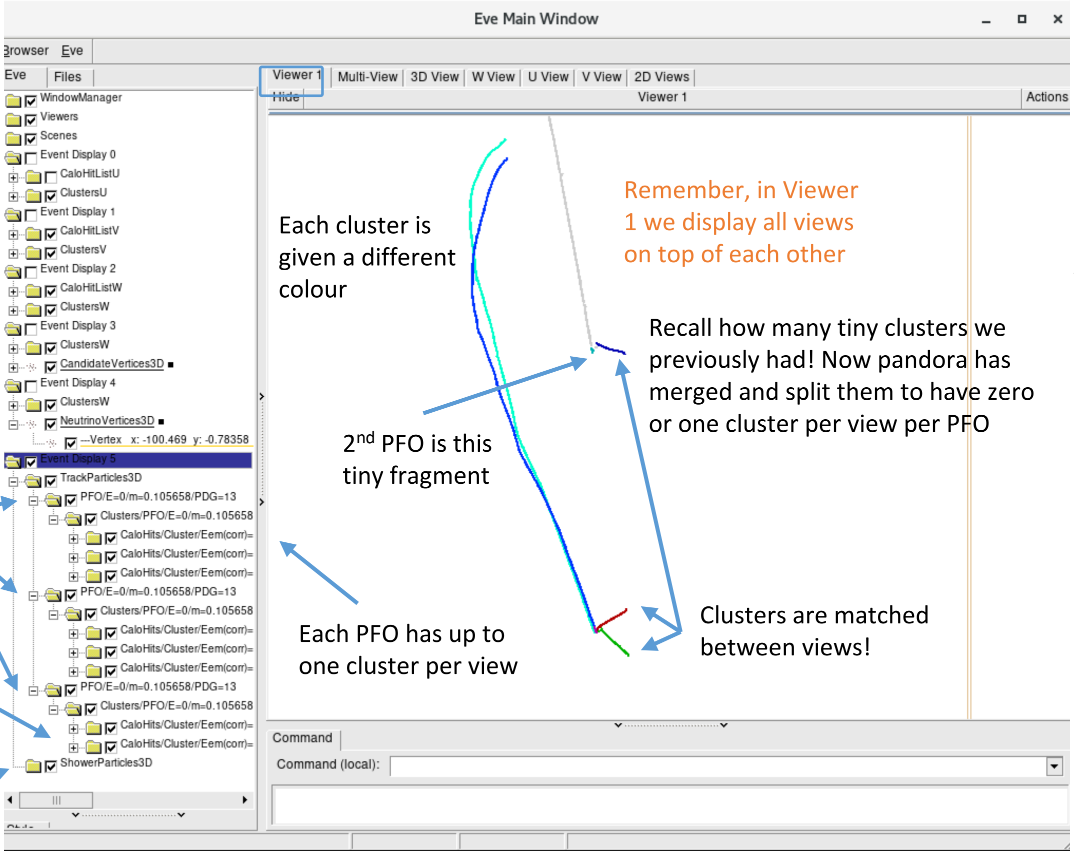
Reconstructed track & shower-like particles

Expand all of the menus to see the clusters at this point and how they have been matched together into reconstructed particles (PFOs)

Here there are 3 track-like PFOs reconstructed

Hover over a cluster to see which view it belongs to - in this case it's the W view

In this event there are no shower-like particles to see


Eve Main Window
Browser Eve
Eve Files
Viewer 1 Multi-View 3D View W View U View V View 2D Views
Hide
Viewer 1
Actions
WindowManager
Viewers
Scenes
Event Display 0
CaloHitListU
ClustersU
Event Display 1
CaloHitListV
ClustersV
Event Display 2
CaloHitListW
ClustersW
Event Display 3
ClustersW
CandidateVertices3D
Event Display 4
ClustersW
NeutrinoVertices3D
--Vertex x: -100.469 y: -0.78358
Event Display 5
TrackParticles3D
PFO/E=0/m=0.105658/PDG=13
Clusters/PFO/E=0/m=0.105658
CaloHits/Cluster/Eem(corr)=
CaloHits/Cluster/Eem(corr)=
CaloHits/Cluster/Eem(corr)=
PFO/E=0/m=0.105658/PDG=13
Clusters/PFO/E=0/m=0.105658
CaloHits/Cluster/Eem(corr)=
CaloHits/Cluster/Eem(corr)=
CaloHits/Cluster/Eem(corr)=
PFO/E=0/m=0.105658/PDG=13
Clusters/PFO/E=0/m=0.105658
CaloHits/Cluster/Eem(corr)=
CaloHits/Cluster/Eem(corr)=
ShowerParticles3D
Command
Command (local):

Each cluster is given a different colour

Remember, in Viewer 1 we display all views on top of each other

Recall how many tiny clusters we previously had! Now pandora has merged and split them to have zero or one cluster per view per PFO

2nd PFO is this tiny fragment

Each PFO has up to one cluster per view

Clusters are matched between views!

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

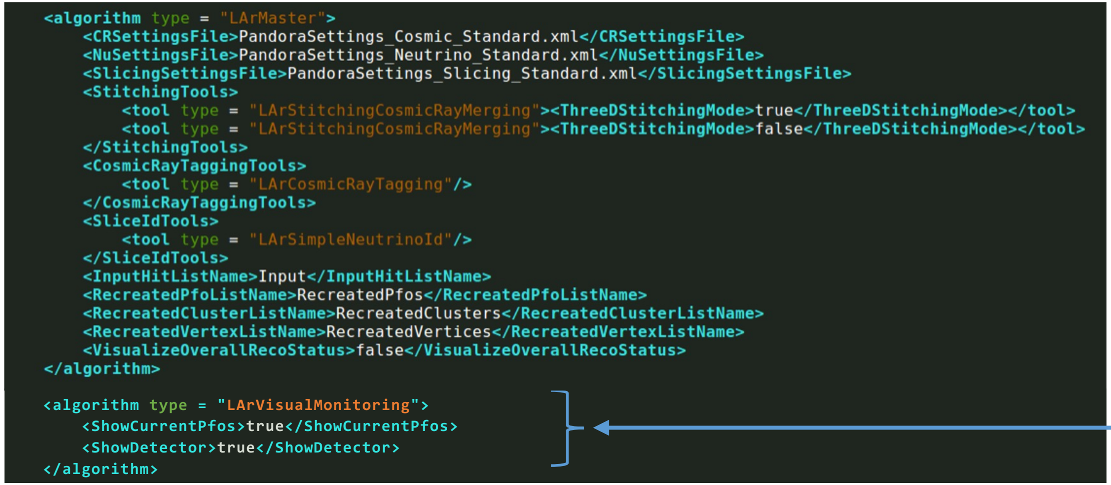
W - wireframe mode
R - return from wireframe mode

When you are finished, press Return ↵ to move to the next display

3D hit and hierarchy reconstruction

Add in some more visualizations

- Uncomment the final visualisation in MyPandoraSettings_Master_Standard.xml



```
<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>
```

Visualize the final reconstructed particles

Run Pandora once again!

```
$ cd $MRB_TOP/reco/work
$ lar -c event_display_driver.fcl -s reco_events.root -n 1
```

- After the event display has loaded press Return ↲ seven times, to skip through parts 1-4

The final outcome

Expand all of the menus again to see what we now have

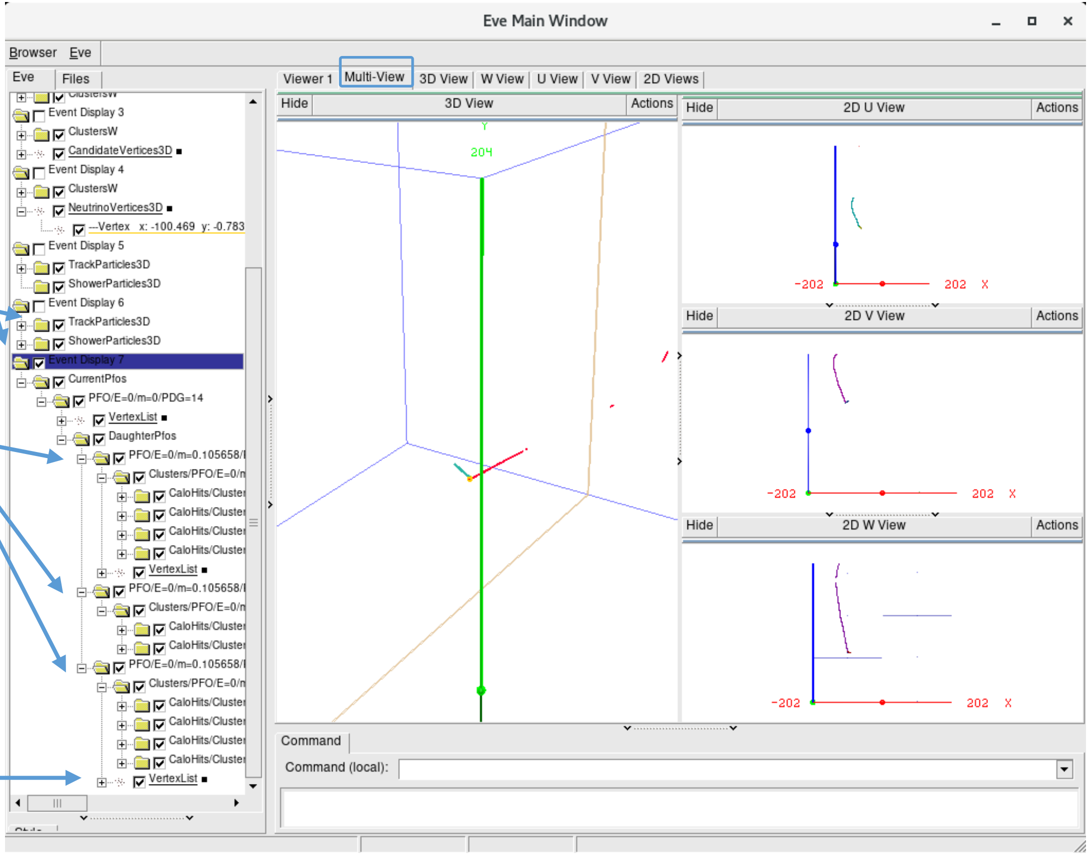
The PFOs are now arranged in a hierarchy! The top-level PFO has PDG code = 14 ⇒ $\nu\mu$

The neutrino PFO has 3 daughter PFOs

Note there are now both 2D & 3D hits

The PFOs have been classified as track-like (assigned PDG 13) or shower-like (assigned PDG 11)

Every PFO has a vertex this is the reconstructed start position



Different 3D hit creation algorithms are used depending on the PFOs track-shower classification

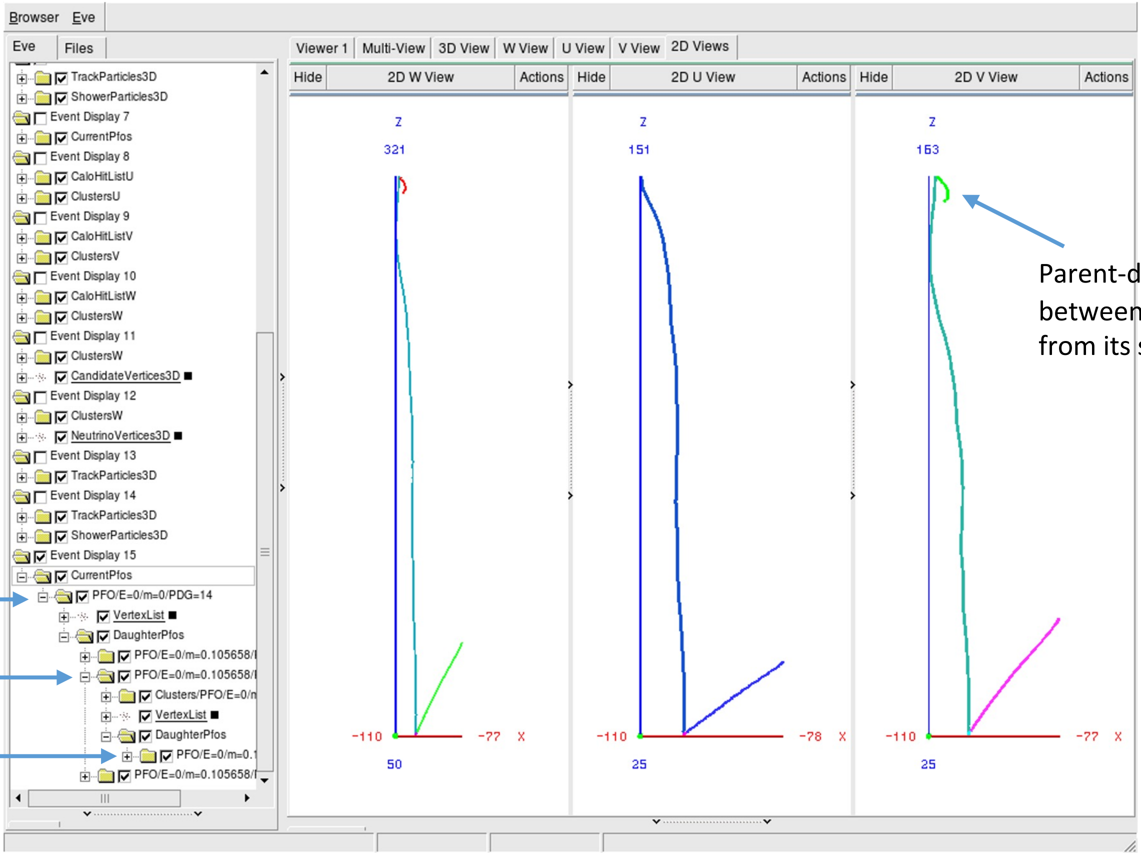
When you are finished, press Return ↵ to close the event display

Secondary particles - a different event

Please note, this is now the final outcome of a different event

In this event, the muon undergoes a secondary interaction

In this event, the muon undergoes a secondary interaction



Parent-daughter link is made between muon and the particle from its secondary interaction

Neutrino PFO

Muon PFO
(Daughter of neutrino)

Electron PFO
(Daughter of muon)

Got spare time?

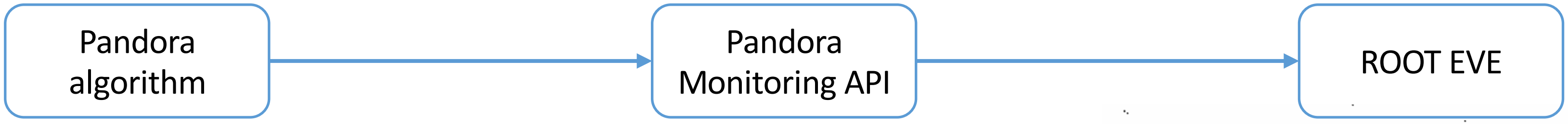
Run your FHiCL file again over multiple events

Do you understand what Pandora is doing in each of the steps?

Additional information

Making visualizations within Pandora

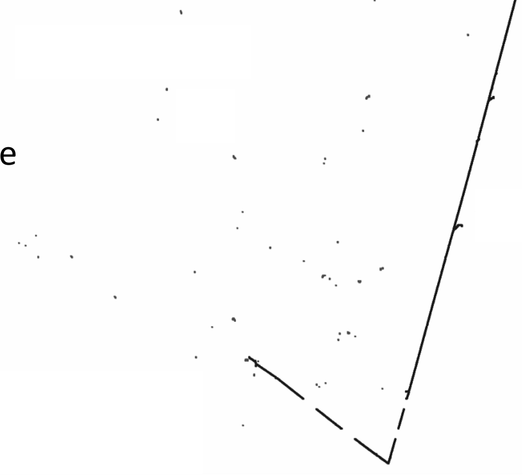
- Event displays are invaluable tools & a number of different options exist
- Today we will be focussing on the event display provided by Pandora



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

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

The Pandora Monitoring API 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 visualization 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