



The JRA-1 Online Monitor

Jörg Behr*

November 25, 2008

Abstract

The EUDET pixel telescope data acquisition software consists of an online monitoring system, that performs online data quality checks during telescope operation and allows a quick estimate of the telescope position and performance. The online monitor was successfully tested and used during past data taking periods.

*DESY, Hamburg, Germany

1 Introduction

The EUDET pixel telescope has a flexible data acquisition system (EUDAQ) [2, 3]. One part of this software is an online monitoring system (RootMonitor), which makes use of the object-oriented data analysis framework ROOT [4]. The RootMonitor was successfully tested and used during past test beam periods of the demonstrator telescope. The synchronisation of the telescope can be done at trigger or at DAQ level. If the DAQ level is chosen only one data stream is created. This was done in the case of DEPFET when the DEPFET sensors were simultaneously read out. The decoding of DEPFET sensors was implemented in the RootMonitor. This yields to an easier alignment of the pixel telescope and the DEPFET when running the latter as device under test (DUT). This memo is intended to give an overview about some major developments of the RootMonitor and all necessary informations for telescope users will be given. The source code of the RootMonitor is available on the EUDAQ web page [6].

2 Starting the Online Monitor

The RootMonitor can be used together with the EUDAQ system during data taking as well as a stand-alone application reading and analysing RAW data files.

With the script **STARTRUN** the EUDAQ software is started together with the RootMonitor. In case of a crash, the RootMonitor can be restarted from the command line. The executable can be found in the **eudaq/bin/** subdirectory and it can be connected to the run control with the command **RootMonitor.exe -r eudet.local**. The name of the host where in this example the run control is running is **eudet.local**. In order to start the RootMonitor stand-alone without a connection to the EUDAQ the run number must be given as a command line option using the **-f** switch, e.g. **RootMonitor.exe -f 3456**. The data file must exist in the directory **eudaq/data**.

The RootMonitor is able to consider different sensor types for the various telescope planes. The sensor types must be specified in a configuration file which is called **root-monitor.conf**. Each column in this file represents one telescope plane. So far the following sensor types can be denoted:

DET_MIMOSTAR2, DET_MIMOTEL, DET_MIMOSA18, DET_MIMOSA5

All reconstruction algorithms account then for different sensors automatically.

If a DEPFET sensor is a DUT in the telescope, then an additional pedestal file with name **Pedestals.bdt** must exist. The RootMonitor will refuse the operation and it will quit if this particular file is missing. The file must contain a white-space separated list of five columns in which the first three numbers correspond to the x-, y-pixel positions and the pedestal. However, the last two numbers in each row must exist but they are not used and will be ignored.

3 Event Reconstruction

Several reconstruction algorithms are used to reconstruct each event and allow therefore a rough online data quality monitoring.

First of all, the RAW or zero suppressed (ZS) data in ADCs for each frame are read and then in order to reduce electronic noise the correlated double sampling (CDS) method is applied to each pixel. The differences between data containing three frames (RAW3) and data containing two frames (RAW2) are correctly taken into account.

3.1 Noise Determination

In order to roughly estimate the noise of the sensors, the root mean square of the CDS values for each pixel is determined. This information is used in the cluster reconstruction. In addition, every 50 events a combined CDS distribution for each sensor plane is determined and a fit of a gaussian function is performed. The width of the resulting function is taken as the noise and used to investigate the time dependence of the noise (see below section 4.1.1).

3.2 Cluster Reconstruction

The online monitor provides a reconstruction of clusters in order to measure the position of hits caused by the passing beam particles. The following steps are performed for each sensor plane independently:

- A list of pixel positions is created containing all pixels that pass the signal to noise (S/N) cut for the seed pixel.
- The list of seed pixels is sorted and starting from the pixel with the highest S/N clusters are reconstructed.
- Two cluster sizes are available ($3 \otimes 3$ and $5 \otimes 5$). All pixels around the seed pixel are joined into a cluster if they pass the S/N cut for the neighbour pixels. Overlapping clusters are not allowed and the overlapping region is assigned to the cluster with the highest S/N.
- Clusters are accepted as hit candidates if the whole cluster S/N - the cluster charge - exceeds a certain threshold.

3.2.1 Position Reconstruction

Two different methods for the cluster position reconstruction are available in the Root-Monitor. Either the position of the seed pixel is used or the centre of gravity for each cluster is determined. The latter is a weighted average of the pixel positions belonging

to one cluster. If x is the x position of the pixel, y the y position of the pixel, $n = 1(2)$ for a $3 \otimes 3$ ($5 \otimes 5$) cluster and c the corresponding CDS value, then this yields to

$$r_x = \frac{\sum_{i=-n}^n x_i \cdot \sum_{j=-n}^n c_{ij}}{\sum_{i=-n}^n \sum_{j=-n}^n c_{ij}}, \quad (1)$$

$$r_y = \frac{\sum_{j=-n}^n y_j \cdot \sum_{i=-n}^n c_{ji}}{\sum_{j=-n}^n \sum_{i=-n}^n c_{ji}}, \quad (2)$$

where r_x and r_y are the cluster x and y positions. If a DEPFET sensor is the DUT, then the cluster reconstruction is performed on this sensor in the same way with two limitations: the pixel noise of the DEPFET sensor is fixed to 1 and the cluster position is always the position of the seed pixel.

4 The Graphical User Interface

In this section the graphical user interface (GUI) of the RootMonitor will be described as well as the available plots. These plots are intended to allow a quick users estimate whether the data taking is in a good shape. The way how the GUI and the plots are connected will also be sketched shortly.

4.1 Available Plots

In this section a description of the important provided plots will be given.

4.1.1 Noise as a Function of Time

As described in section 4.1.1 the time dependence of the telescope noise is monitored. This is in particular of interest to see whether the telescope cooling is working and stable. This information is only available if the telescope runs in RAW mode. Figure 1 shows this time dependence of the noise in ADC, which is in this case almost constant. The different sensor planes are indicated with different colours.

4.1.2 X and Y Cluster Correlations

During test beam setup it is difficult to place the telescope inside the beam. In order to allow a better positioning correlation plots were implemented in the RootMonitor.

If a beam particle passes the telescope planes, then it creates clusters in each of them. Furthermore, if the effect of multiple scattering is sufficient small, then this results in a spatial correlation between the reconstructed clusters in the sensor planes. For this reason the reconstructed position in x direction as well as in y direction of each possible cluster pair for neighbouring sensors was filled into a 2-dimensional histogram. If there is a correlation, then it will show up as a straight line in plots of this kind.

In figure 2 the x and y correlation plots are shown for each pair of neighbouring planes as well as the correlation between the first and the last plane. A deviation from a

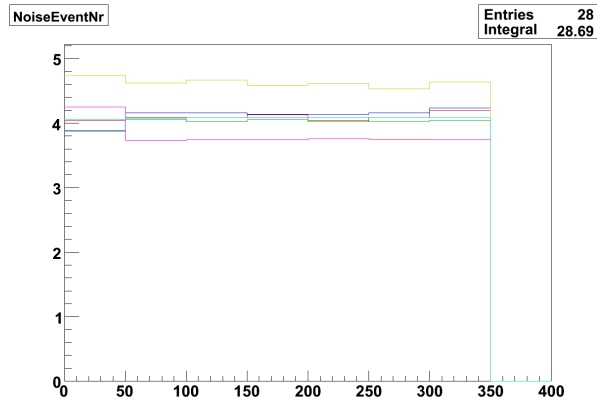


Figure 1: The noise for each sensor plane as a function of time.

diagonal line can be interpreted as telescope mis-alignment. The telescope consists of two arms each equipped with three layers of pixel sensors. The orientation of the sensor mounting is different in the second arm. That's why the orientation of the visible line is not the same in each plot. Entries that do not belong to the solid line are induced by combinatoric background and sensor noise.

In the case of a DEPFET sensor as DUT the RootMonitor is able to provide correlation plots that show the cluster position correlation between each telescope plane and the DEPFET sensor. This can be seen in figure 3 where as an example only some of the available plots are shown

4.1.3 Hit Correlation

In the previous section the correlation of cluster positions in different sensor planes was described. Plots of this manner are only helpful if the influence of multiple scattering of beam particles in the sensor material is sufficiently small. With increasing multiple scattering the cluster position correlation vanishes and therefore the background that enters the correlation plots increases.

In order to deal with data that show large multiple scattering correlation plots between the number of hits in each sensor plane were implemented in the RootMonitor. In figure 4 this correlation can be seen as a cloud that follows a diagonal line. The offset in some of the plots is caused by the different amount of noise in the sensors. The number of bins can be adjusted via the GUI in order to fit better to the data hit multiplicity per event.

4.1.4 CDS Lego

The “CDS Lego” plots were intended to see hits on an event by event basis in the telescope. For this every N events the corresponding plots for each sensor plane are filled with the current CDS values, whereas N can be adjusted in the GUI.

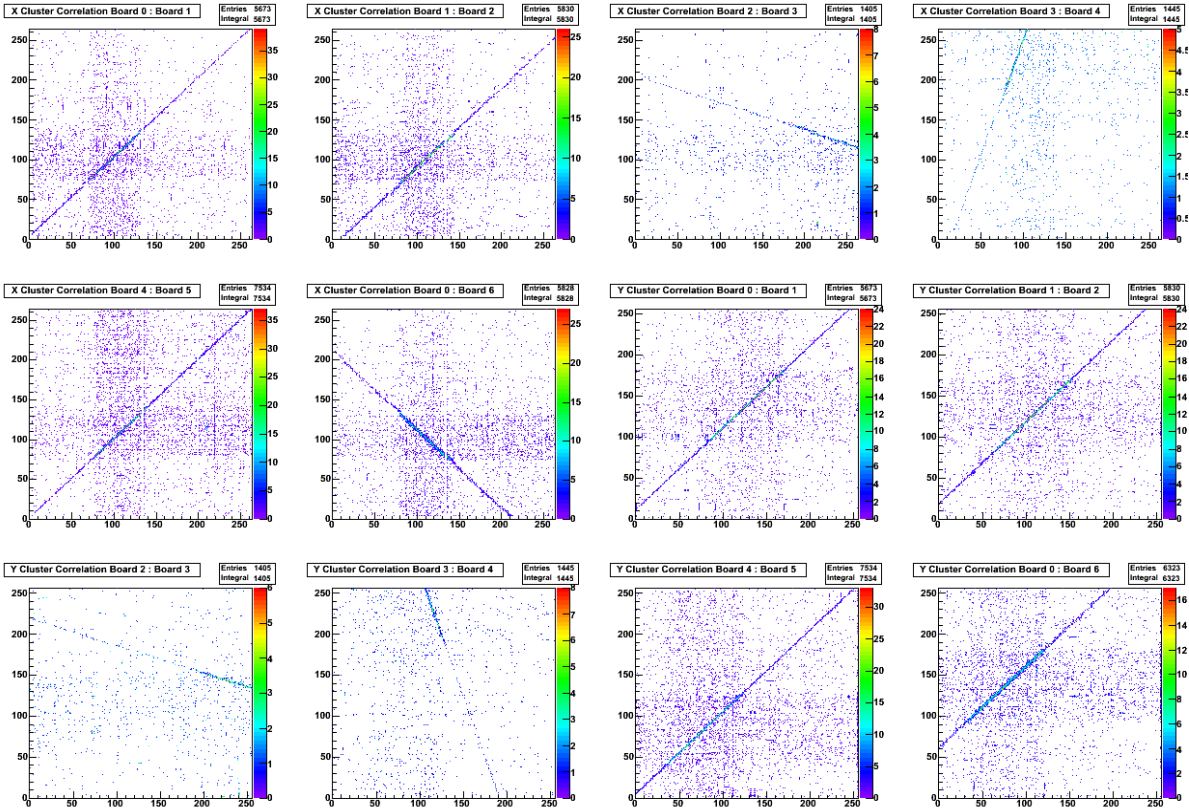


Figure 2: Cluster position correlation between pairs of sensor planes.

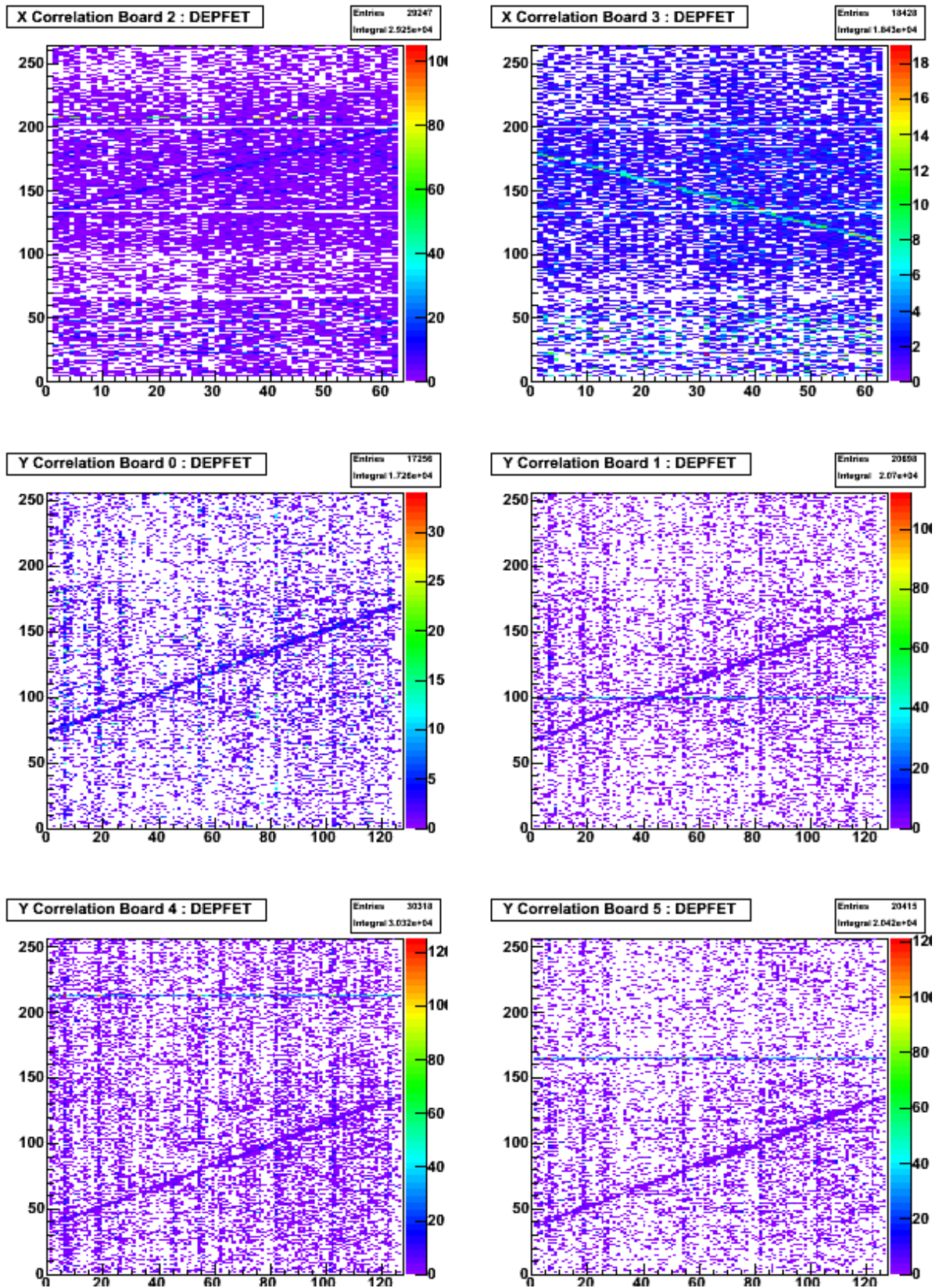


Figure 3: Cluster position correlation between sensor planes and the DEPFET sensor, that was operating as DUT. Only some of the available plots are shown.

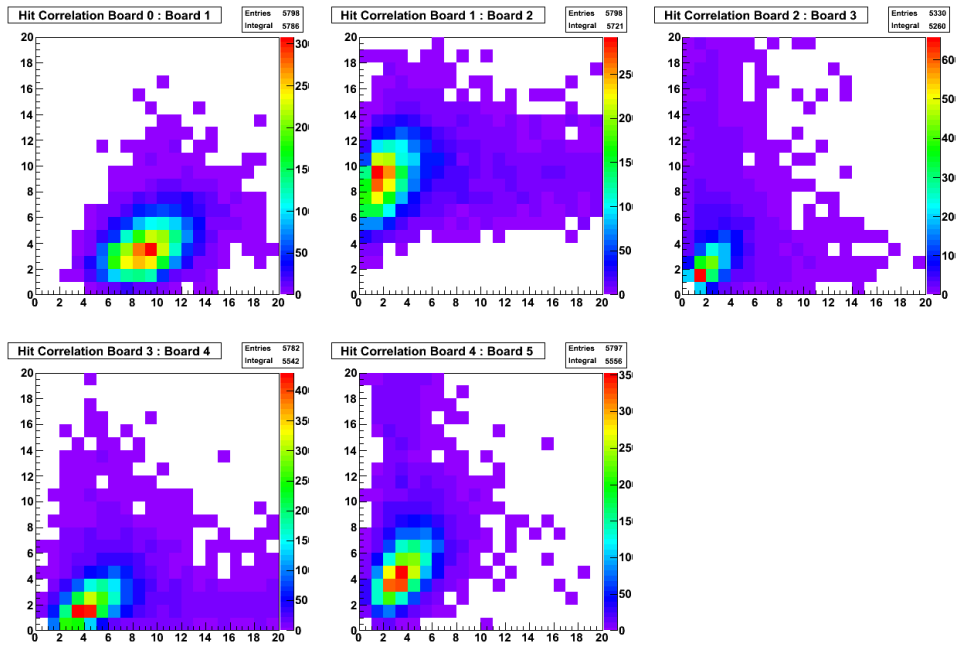


Figure 4: Correlation between the number of hits in neighbouring telescope planes.

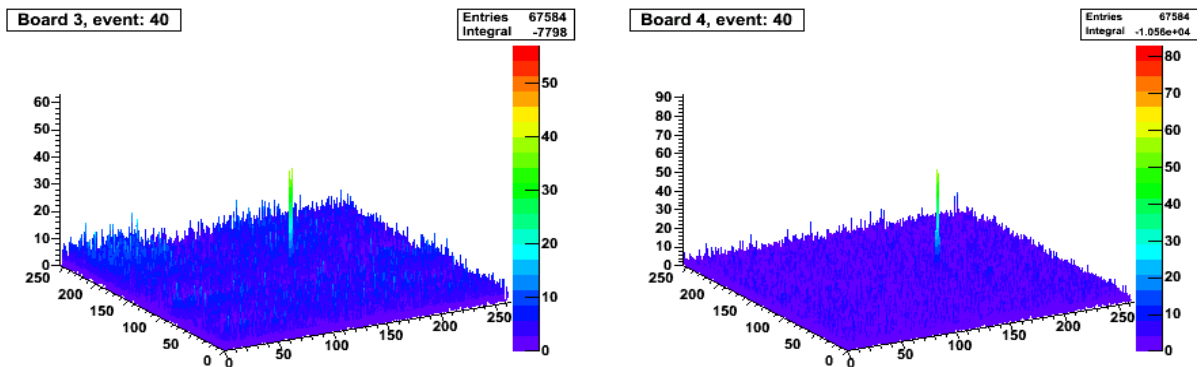


Figure 5: The CDS value for sensor plane number 3 and 4 and event number 40.

Figure 5 shows CDS values for two sensors for event number 40. In each sensor plane one hit can be seen.

4.1.5 Other Plots

The RootMonitor provides additional information about the telescope that was not mentioned in the previous sections. A list of available plots is given in the following:

- ADC and CDS distribution for each sensor plane separately
- 2-dimensional cluster position maps for each sensor
- the number of seed pixels and the number of clusters per sensor
- the noise per sensor
- the cluster charge distribution for each sensor

4.2 Configuration Menu

Sometimes it is important to change reconstruction parameters online without touching the source code of the RootMonitor. In order to ensure a better usability the possibility to change online via the GUI parameters was implemented.

In figure 6 a screen shot of the configuration tab is shown, where one can change parameters. Right-hand side a list of check boxes allow the user to enable and disable pads of histograms. The check boxes are grouped into three parts depending on where the corresponding histograms are shown. After pushing the “apply” button the configuration will be updated and all canvases will be divided dynamically depending on the number of active pads. Histograms belonging to disabled pads are filled in the background. Thus the information is not lost. If one disables all currently not needed pads, then the performance of the RootMonitor GUI increases. It is also possible to refresh the histogram view.

In the window header the current file name, the run and event numbers are displayed. It is possible to reduce the number of processed events and change the time interval after which the RootMonitor is forced to update histogram drawings.

4.3 Technical Implementation

The GUI of the online monitor uses platform independent ROOT libraries, that originally based on the Xclass’95 widget library. The RootMonitor inherits from an monitor class provided by the EUDAQ software as well as from abstract GUI classes.

The changeable reconstruction parameters are stored in an object of the class **ConfigurationClass** that is updated whenever the “apply” button is pushed. The called method **RootMonitor::ProcessMessage** reads the GUI elements and writes their current content to the configuration object.

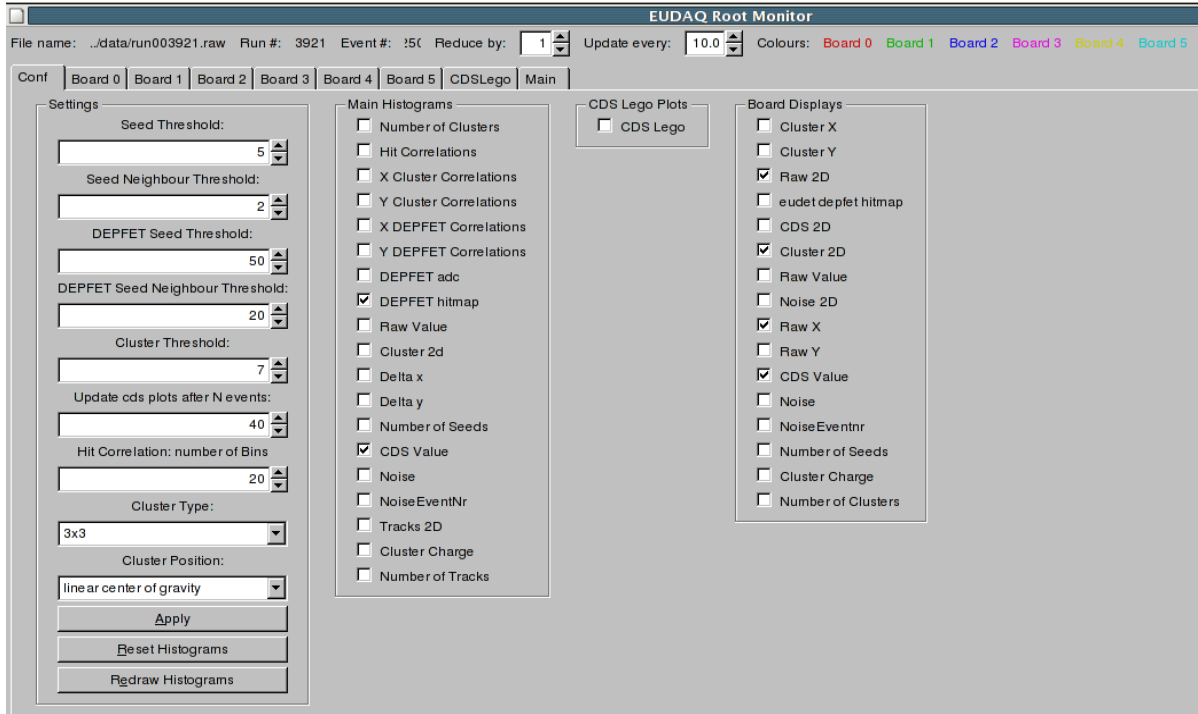


Figure 6: The configuration menu, where one can change online reconstruction parameters. It is also possible to enable or disable pads of histograms.

In order to connect the GUI check boxes with pads of histograms the simple class **histopad** was developed. This class holds a pointer to a check box and arrays of pointers to histograms as well as drawing options for each histogram. This is illustrated in figure 7 with two assigned histograms that are connected with object pointers indicated as arrows. The objects of the class **histopads** are organised and collected in 1- and 2-dimensional arrays. It is therefore possible to assign histograms to different pads. if the “apply” button is used, every displayed canvas is reseted. Furthermore, the number of enabled pads is counted, the canvases are then suitable divided and the stack of histograms is drawn in each pad using the correct drawing options.

The RootMonitor is a multi-thread application and it continuously reads the telescope data. After a certain time period, which can be adjusted with the GUI, the drawing of every histogram is refreshed. In order to gain performance the standard ROOT histogram classes were over-written and a status flag was implemented, that indicates whether the histogram was modified since last update. Only those histogram drawings that were modified are then refreshed. This is in particular useful if the CDS lego plots are activated, because these histograms are only updated every N events. Displaying these histograms reduces the performance due to the complex graphical rendering.

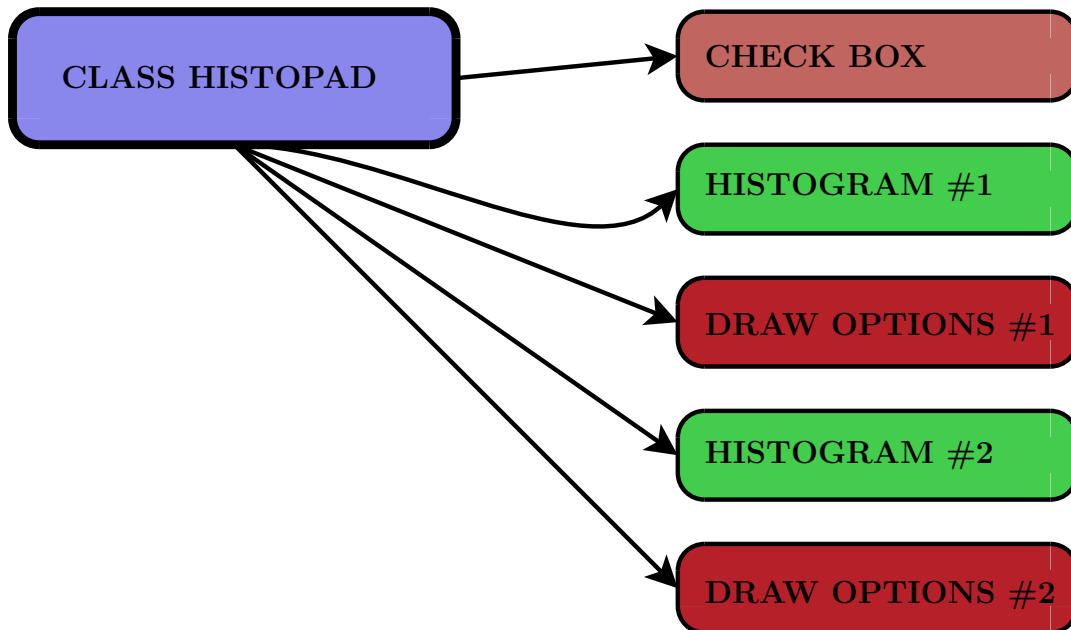


Figure 7: Connection between check boxes, pads and arrays of histograms. In this example two histograms were assigned to one pad. The arrows indicate pointers.

5 Summary

The EUDET pixel telescope data acquisition software consists of a online monitoring system, that provides a event reconstruction and several plots allowing a data quality check during telescope operation. The graphical user interface was improved and it makes the usage of the online monitor easier and more comfortable for the users. In addition the performance and stability was improved.

Acknowledgement

This work is supported by the Commission of the European Communities under the 6th Framework Programme "Structuring the European Research Area", contract number RII3-026126.

References

- [1] R.D. Heuer, D. Miller, F. Richard, P. Zerwas (eds.), "TESLA Technical Design Report, Part IV", DESY 2001-011, ECFA 2001-209, 2001.
- [2] E. Corrin, D. Haas and M. Pohl, EUDET-Memo-2006-07 (2006).
- [3] D. Haas, Proc. of the LCWS'07, Hamburg, EUDET-Report-08 (2007).

- [4] <http://root.cern.ch/>
- [5] L. Reuen, EUDET-Memo-40 (2007).
- [6] <http://projects.hepforge.org/eudaq/>