BrightnESS

Building a research infrastructure and synergies for highest scientific impact on ESS

H2020-INFRADEV-1-2015-1

Grant Agreement Number: 676548

Deliverable Report: D5.2 Processing Choices for Detector Types

1 Project Deliverable Information Sheet

BrightnESS Project	Project Ref. No. 676548	
	Project Title: BrightnESS - Building a research infrastructure and synergies for highest scientific impact on ESS	
	Project Website: https://brightness.esss.se	
	Deliverable No.: 5.2	
	Deliverable Type: Report	
	Dissemination Level: Public	Contractual Delivery Date: 2017-05-31
		Actual Delivery Date:
	EC Project Officer: Anna Maria Johansson	

2 Document Control Sheet

Document	Title: BrightnESS_Deliverable_5.2	
	Version: 0.1	
	Available at: https://brightness.esss.se	
	Files: 1	
Authorship	Written by	Martin Shetty, Morten Jagd Christensen (WP5)
	Contributors	Stig Skelboe (WP5)
	Reviewed by	Tobias Richter (WP5 leader)
	Approved by	Steering Board

3 List of Abbreviations

ADC	Analog-to-Digital Converter
API	Application Programming Interface
CNCS	Cold Neutron Chopper Spectrometer
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
DMSC	Data Management and Software Centre
DPDK	Data Plane Development Kit
EPICS	Experimental Physics and Industrial Control System
EF	Event Formation
EFU	Event Formation Unit
ESS	European Spallation Source ERIC
FPGA	Field Programmable Gate Array
GEM	Gas Electron Multiplier
GPU	Graphics Processing Unit
HDF5	Hierarchical Data Format version 5
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
VLAN	Virtual Local Area Network
MAC	Media Access Controller
MTU	Maximum Transmission Unit
NIC	Network Interface Controller
RAM	Random Access Memory

SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

4 List of Figures

5 Table of Contents

6 Executive Summary

Task 5.1 “Creating a standard neutron event data stream for different detector types” focuses on software event processing for the expected ESS detector suite. This will deliver generic neutron event information required for scientific experiments. Deliverable 5.2, which presents the results of our investigations in this task, shows that we are in a good position to cope with the processing needs of the different detector types once their configuration is final. For the future of the BrightnESS task, we see no unusual or high impact risks.

For the majority of cases, specifications of upcoming detectors and their raw output format are not yet in a state that allows software prototyping. However, during the course of the project thus far, we have developed a good working relationship with the detector group and their partner institutes across Europe. That resulted in a good understanding of the domain and, not least, to two working prototypes of the event formation system for NMX and the Multi-Grid detector ahead of schedule. These two detector systems are a good template for future customized implementations. Most systems are quite similar to the Multi-Grid detector, and with NMX we are close to covering the most complex computational needs. To drive these working prototypes forward, we have put in place a common and modular framework. It hosts the detector-type specific processing algorithms whereby future tasks are divided into manageable modular chunks.

With the tools developed, we are confident that we can adapt our processing algorithms and parameters in sync with any new hardware prototypes that become available, any updates to prototype detectors and production versions. We will be able to review quality indicators for individual events, statistics within and across datasets and interactively examine the effect of different filters or code changes. The software will also enable the specialist from ESS' Data Management and Software Centre (DMSC) or the ESS detector group to fine-tune the event processing pipeline, change detector architecture and varying user needs, and will serve as commissioning and calibration tools.

7 Purpose of Software Event Formation

Neutron detectors work by converting neutrons through nuclear processes into charged particles that can be captured using electronics. These secondary particles are either photons or they carry a charge and kinetic energy. For traditional detector technologies (for example ^{3}He-based detectors) research facilities have developed a full data chain in electronics, from the amplification of the secondary particle's data all the way to forming neutron events from that.

Software Event Formation is the process that starts by taking digitized raw detector information provided by the readout hardware and ends with the transmission of event-tuples to the data aggregator in a standardized format. An event tuple is a timestamped identifier of a pixel of the logical detector image. It contains a timestamp ***t*** and a ***pixel ID***. There is a one-to-one correspondence between the {***t, pixel ID***} tuple and the pixelated spatial and temporal location of a detected neutron. By reducing the information down to those two numbers, the data volume is reduced by up to three orders of magnitude, depending on the detector technology. Storing all the raw data therefore is not feasible for a facility that aims to have the highest flux of neutrons in the world.

The input from the detector readout system is dependent on the detector technology. Therefore, the processing is also detector technology dependent. Since most ESS detector types are novel, no proven electronics exist to convert the raw detection data into standard neutron events. BrightnESS will provide the means to develop the processing in software in parallel to the hardware development, which reduces the development time over a sequential approach and hence reduces the technical risk to the overall ESS project.

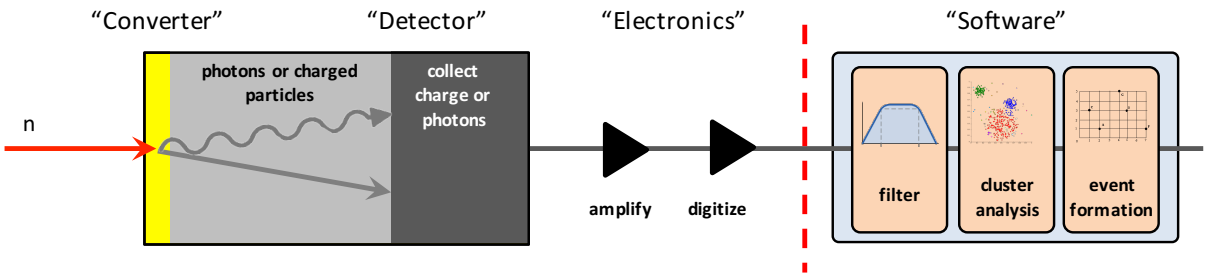


Figure 1: Schematic of the relationship of detector hardware and software event formation

The two main interfaces of the Event Formation system are the detector data interface on the ingress side and the aggregator data interface on the egress side (see Figure 1). These will be described in the following sections.

7.1 Detector Data Interface

The physical detector data readout interface is a standard fibre Ethernet with a link speed of at least 10 Gb/s. The data transport is partly standardized by the ESS detector group, which will cover a large fraction of the ESS Instruments. In addition, several unique readout systems must also be supported. The specific data format will be customized to the individual detector types. Regardless of these differences, the data essentially contains geometric and timing-related information. This information is used to a) identify clusters in the data stream and b) calculate the position of the detected neutron.

7.1.1 Detector Geometry

A detector will typically have either a 2D detection area or a 3D detection volume. Discrete units of these spaces are referred to as pixels and voxels respectively. Each individual pixel (or voxel) has a unique representation – either as a logical coordinate such as (x, y, z) or simply a pixel ID (voxel ID). The 2D or 3D information originates from the physical design and electric layout of the detector. For example, the Multi-Grid detectors use wires and grids, Multi-Blade detectors wires and strips and the NMX detector x-strips and y-strips (see Figure 2 for an illustration).

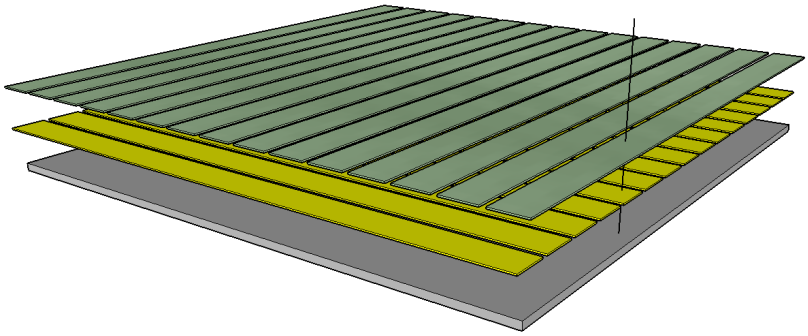


Figure 2: Example of a coincidence detector using strips for charge collection. Electrons deposit charges on both x-strips (top layer) and y-strips (middle layer) giving rise to two (or more) readouts with different timestamps.

Information about the physical dimensions of the detector is needed in the data analysis phase, but for the purpose of event formation it is not needed. We therefore operate with logical definitions for describing detector geometry. These definitions must be unambiguous for correct data processing. Examples of choices are definitions of x, y and z directions for determining the 3D position of a voxel, or the mapping between pixel ID and (x, y, z) position. Such a mapping will have to be defined for each unique detector type of every instrument. See Figure 3 for the definition of the geometry of a Multi-Grid detector demonstrator panel consisting of two modules each constructed of 48 grids and 64 wires.

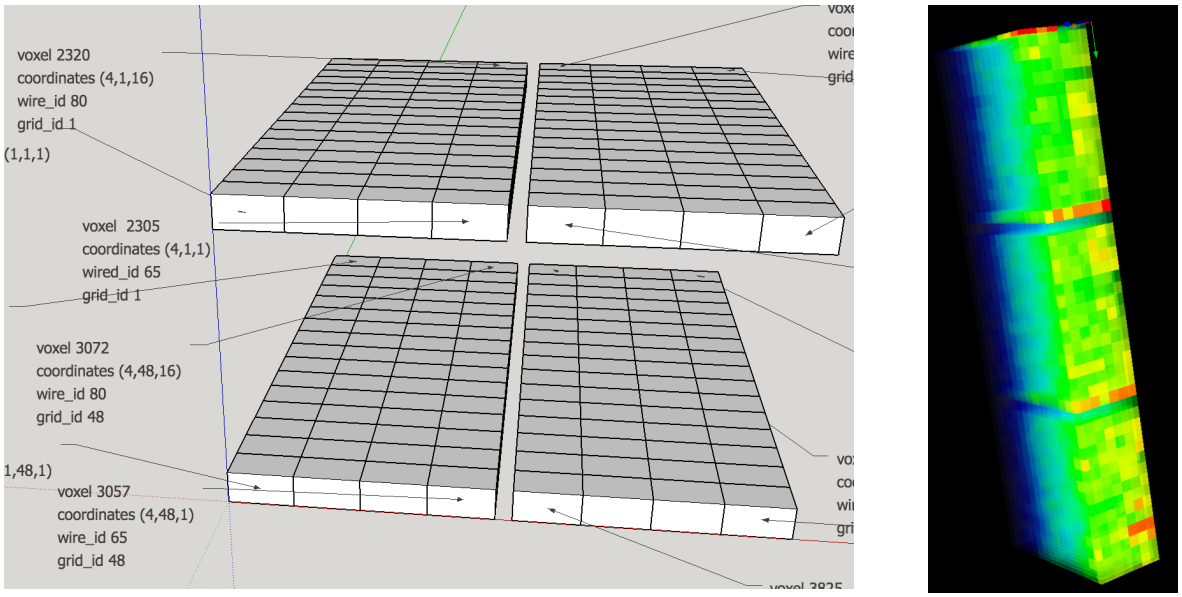


Figure 3: Definition of the logical geometry for Multi-Grid detectors. Wire and grid definitions are given, but the (x, y, z) coordinates and the voxel IDs are an agreed convention (left). 3D Rendering of Multi-Grid detector voxel intensity based on 10000 readout events (right).

7.1.2 Clustering

This step consists of correlating multiple coincident readout events from spatially adjacent detector segments (pixels, voxels or strips). The purpose of this step is to prevent over- or under-reporting of neutron events and, in some cases, it is essential for determining the neutron position with acceptable uncertainty.

Some detectors will produce several readout values for a single detected neutron. These readout events should be grouped into clusters that contain values that are close in space and in time. The event formation will then refine the location and timestamp of the original neutron conversion from the cluster.

Figure 4 shows an example of how this might look in a detector that has wires and strips onto which charge is deposited when a neutron has been captured. In addition to [time, strip] and [time, wire], the readout system also provides the intensity of the deposited charge, but this information is not shown in the figure. Various methods can be used to partition the readout data into clusters, ranging from simple time-boxed grouping to unsupervised learning algorithms such as k-means clustering.

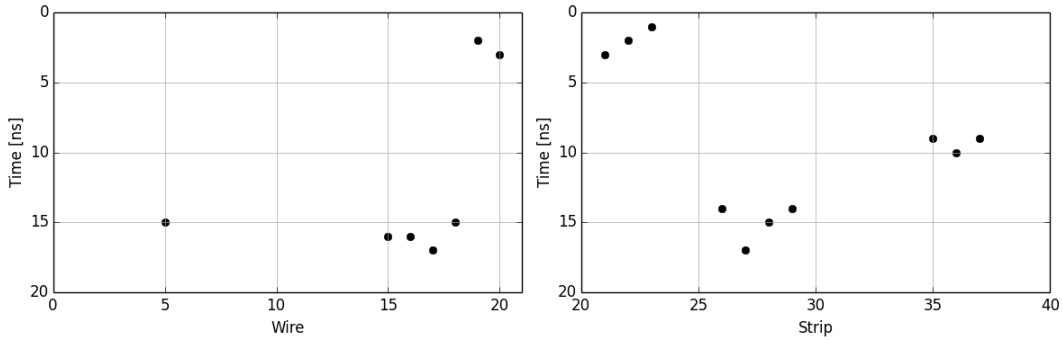


Figure 4: Example of data from a coincidence detector that requires clustering. Valid clusters are found at time 2 ns, an invalid readout with no coincidence is found at 10 ns and a potentially valid cluster at time 16 ns with a possible invalid readout on strip 5.

7.2 Aggregator Interface

The output of the Event Formation system is streamed to a scalable data aggregator: Apache Kafka. A separate report on this choice and architecture was presented in deliverable D5.1 [8]. Data streams are addressed using topics, and topic namespaces has been defined for detectors and monitors. Kafka uses TCP for the reliable delivery of (arrays of) event tuples serialized with Google Flatbuffers. The definition of the Flatbuffer data format is done using user defined schemas. A standard set of Flatbuffer schemas have been developed to ensure that data is unambiguously interpreted by both the event formation producer and downstream consumers.

```
// Schema for neutron detection event data

include "is84_isis_events.fbs";

file_identifier "ev42";

union FacilityData { ISISData }

table EventMessage {
  source_name : string; // optional field identifying the producer type, for example detector type
  message_id : ulong;   // consecutive numbers, to detect missing or unordered messages
  pulse_time : ulong;   // time of source pulse …  nanoseconds since Unix epoch (1 Jan 1970)
  time_of_flight : [uint]; // nanoseconds measured from pulse time
  detector_id : [uint];   // detector on which the event was recorded
  facility_specific_data : FacilityData;
}
root_type EventMessage;
```

Figure 5: Common schema used for all detectors at ESS operating in event mode.

7.3 Summary

Generally speaking, Event Formation consists of the following steps:

1. Upon receiving a packet of readout events, parse the readout format and for each readout event;

   resolve detector segment information (pixel, voxel, strip) via predefined mapping from hardware data (panel, chip, channel) to a geometrically meaningful location

if needed by the clustering step, submit the parsed readout event to the clustering queue.

2. Check cluster queue for any clusters ready for processing. If any are available, perform detector-specific event formation processing;

   if an acceptable event is produced by the detector-specific processing step, perform any additional geometrical mapping to conform event position to conventions expected by downstream consumers.

8 General Considerations for Choosing Processing Options

This section discusses the main considerations for event processing. The two main concerns are correctness and performance. By correctness we mean how accurate the determination of the neutron position and timestamp is and how well we can distinguish real neutron events from background signals such as gammas or electronic noise. This will mainly be covered in the detector technology specific section. Performance is about the ability to cope with the expected large data rates. As there is no significant storage buffer for the raw detector data, processing needs to keep up with the experiments in real time or the software processing will need to drop events, thus limiting the performance of the instrument.

8.1 Readout electronics output

The data delivery methods and data encodings must be unambiguously defined. There is ongoing collaboration with the ESS Detector Group to match the needs and capabilities of the readout system and the software event formation system. It has already been agreed that the data stream from the readout system will contain all information necessary to perform the event formation calculations. This means that no sideband communication should be required. Readout data will be formatted to be compatible with the selected CPU architecture, for example: alignment on n-bit boundaries (n typically 8, 32, 64), and support for little endian representation of multi byte integer values. The readout system will also provide data integrity measures, such as sequence numbers and heart beat signals which are valuable during both development and operations.

8.2 Latency limits

Latency is the delay caused by performing some operations on data. Uncontrolled latency can lead to large memory requirements or data loss. The overall latency from the time a detector was read out until an event has been produced as such, is not critical for the subsequent analysis. However, there is also an expected maximum latency of the whole data chain from the user's perspective. The user will expect feedback from the experiment without undue delay, in the order of a second.

In the Event Formation system, latency is mainly caused by three factors

- data transmission;
- data processing;
- data producer queueing.

The data transmission latency is determined by the link speed of the transmission medium and the amount of data to be transmitted. This type of latency is limited to a maximum of 7 us (10G Ethernet, 9K data).

The clustering step in the event formation system needs to check readout events for coincidence and geometrical adjacency. The architecture of the readout system cannot guarantee that events arrive in chronological order. However, the clustering process may only submit clusters for further processing once it is guaranteed that no more adjacent event information will arrive. To keep the buffering memory for “unfinished” clusters to a reasonable size, the readout electronics needs to have a typical maximum latency, i.e. time it takes to send out events from the detection system. Currently no specific numbers are available for this latency contribution. This can, to some extent, be countered by planning for sufficient RAM and CPU cache sizes on the EFU machines.

The data producer will queue up event tuples until a suitable chunk of data can be delivered. The size of the buffer is 1 MB. The latency will be determined by the event generation rate. If the rate is very low, the buffer will be flushed once per second.

8.3 Performance considerations

The overall performance gauge of the event formation system is the capability to process all detector data at whatever rates it is produced. There are several contributions to achieving good performance: efficient algorithms and data processing, fast computational hardware and transmission of readout data. Each of these will be discussed below.

8.3.1 Algorithms

Because of the high data rates that will be generated at ESS, the implemented algorithms must perform well. The clustering step may be more sensitive to performance costs, as events may “pile up” at high rates and each new readout event may need multiple comparison operations with pending clusters.

We expect our performance requirements can be satisfied by using standard processing modules with individual performance guarantees and doing all accumulation and averaging operations over a dataset by looping over all the data points only once. All evaluated algorithms can be implemented with this assumption.

It may also be necessary to ensure certain properties in the data stream to achieve such performance, such as latency limits or a particular ordering of data, as mentioned in the previous sections. Certain configurations of readout electronics and/or FPGA code may have a role to play in optimizing performance, but we have tried to minimize such assumptions.

8.3.2 CPU

In the design philosophy, all data processing for event formation should be done in software based on general purpose CPUs. Only simple processing, such as zero suppression and threshold based discards, will be done in FPGAs. This means that the data processing will take advantage of all the features of modern operating systems, including efficient network I/O, parallel processing, use of third party software etc. Using standard commercial hardware will also guarantee the availability of replacement parts for the expected lifetime of a detector system. Modern CPU architectures are running at around 2.4 billion instructions/s and CPUs with more than 30 cores and 60 hyper-threads are available.

We anticipate the use of the Intel Xeon processor family for event formation data processing. The current Xeon CPU hardware, which has already shown good performance, is in the middle of the range of currently offered CPUs from Intel. There are solutions that will allow more cores, larger caches and support for aggregating up to four CPU sockets on a single motherboard to achieve staggering computational power.

Tuning a good performance to a great one requires measuring where the processing bottlenecks are in order understand where to make suitable modifications. The advantage of using CPUs is that there are a number of existing tools to perform these measurements. Examples are: CPU usage profilers, cache profilers, memory profilers, etc.

8.3.3 Network

Data transport from the readout system will be based on at least 10 Gb/s Ethernet. Raw link speed is not a concern: 40G and 100G Ethernet is readily available today and 400G is currently being developed by IEEE and will be available around 2018. UDP is a candidate for the transport protocol, although this has not been fully decided. However, more advanced protocols such as TCP have been decided against. This is partly due to the added complexity to the FPGA-based readout system this would cause, and partly because there is no obvious advantage of using TCP due to the tightly controlled network infrastructure and topology.

We have demonstrated the saturation of a 10 Gb/s Ethernet link using UDP packets of size 9000 B with zero packet loss. Depending on the type of detector and on the data packing done by the readout system, this corresponds to 78M 128 bit readouts per second.

9 Detector Specific Choices

This section describes the different detector technologies for which we now have detailed technical information. The main focus of this work has been the NMX Gd-GEM (Gas Electron Multiplier) detector, because as a novel detector technology, it presents additional processing challenges.

Most ESS instruments will use coincidence detectors, where a captured neutron will result in simultaneous signals in multiple independent channels. In the case of Multi-Grid, for example, signals are induced on both wires and grids. These are orthogonal to each other and the combined (coincidence) signal determines the position of the detected neutron. Often more than two signals are detected, which requires a clustering step in the data processing as discussed in 7.1.2. Once an efficient clustering algorithm has been created, it can be adapted and reused in for other instruments.

The Multi-Grid, Multi-Blade, Jalousie, and Gd-GEM use coincidence position determination. This covers the following ESS instruments[1]:

Detector type	Instruments		
Multi-Grid	C-SPEC	T-REX	VOR
Multi-Blade	FREIA	ESTIA	
Jalousie	DREAM	HEIMDAL	MAGIC
Gd-GEM	NMX		

[1] The Jalousie detectors are being developed for ESS by CDT.

9.1 Multi-Grid

The Multi-Grid detector is based on a neutron conversion event in ^{10}B for neutron detection [12]. It is based on a design of orthogonal wires and grids. The grid element is shown in Figure 6. The simultaneous occurrence of a signal in a grid and on a wire determines a coincidence.

Data processing for Multi-Grid detectors consist of the following steps:

- apply threshold values for grid and wire readout signals. Discard signals with too low a value;
- perform clustering analysis to obtain concurrent readouts of grids and wires;
- discard invalid data such as double neutron events and orphaned readouts;
- calculate grid position and wire position from grid IDs and wire IDs;
- convert grid and wire positions to pixel IDs.

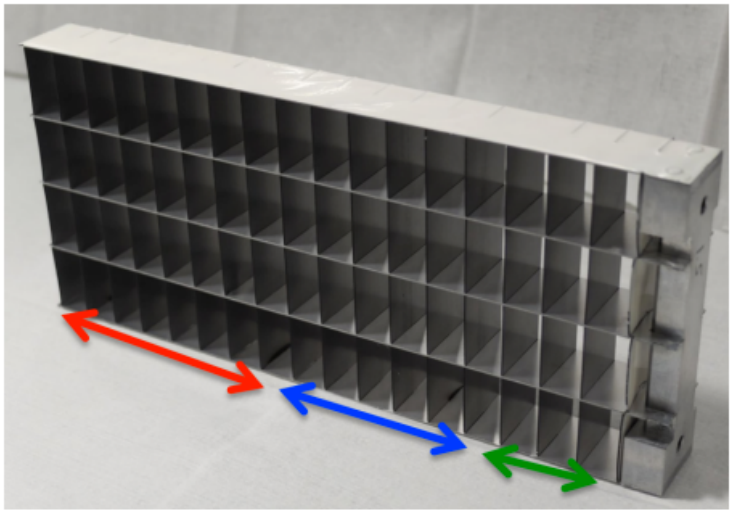


Figure 6: The Multi-Grid detector building block, the Grid element. Multiple grid elements are stacked to create a module. Multiple modules are assembled side-by-side to create the detector. Wires run through each cell of the grids (The arrows are not relevant in this context).

Most of these processing steps require very little computational effort and no buffering. However, the clustering step does need to queue up readouts until there is sufficient data available for the subsequent steps. The maximum amount of buffering depends on the time resolution of the readout system and the number of readout values produced per conversion event. Neither of these values have ben definitively been defined, but the minimum for the latter is two readout events (one for the grids and one for wires). For the detection of a single neutron, we expect less than 5[2] readouts for wires and grids for a total maximum of 10 readout events. This is currently not considered to be a challenge.

9.1.1 Multi-Grid demonstrator

We have collaborated with the Multi-Grid detector development team for delayed streaming of live Multi-Grid data. With this, a full processing pipeline prototype for event formation was demonstrated based on previously recorded data from CNCS. The experimental setup is shown in Figure 7.

[2] This estimate is based on discussions with the Multi-Blade detector team. This technology is very similar. The numbers will be adjusted when the Multi-Grid detector has been subject to more tests.

The prototype implemented the processing of the data from the experimental readout system (which is not the final one) and demonstrated monitor images of the detector and also an 'expert view' of application counters. In this experiment, there was no clustering analysis as the data was already clustered by the electronic readout system but the remaining processing steps were performed as described above.

The experiment showed that it was possible to process Multi-Grid data at rates of the order of millions of events per second with very little overall utilization of the available computational resources.

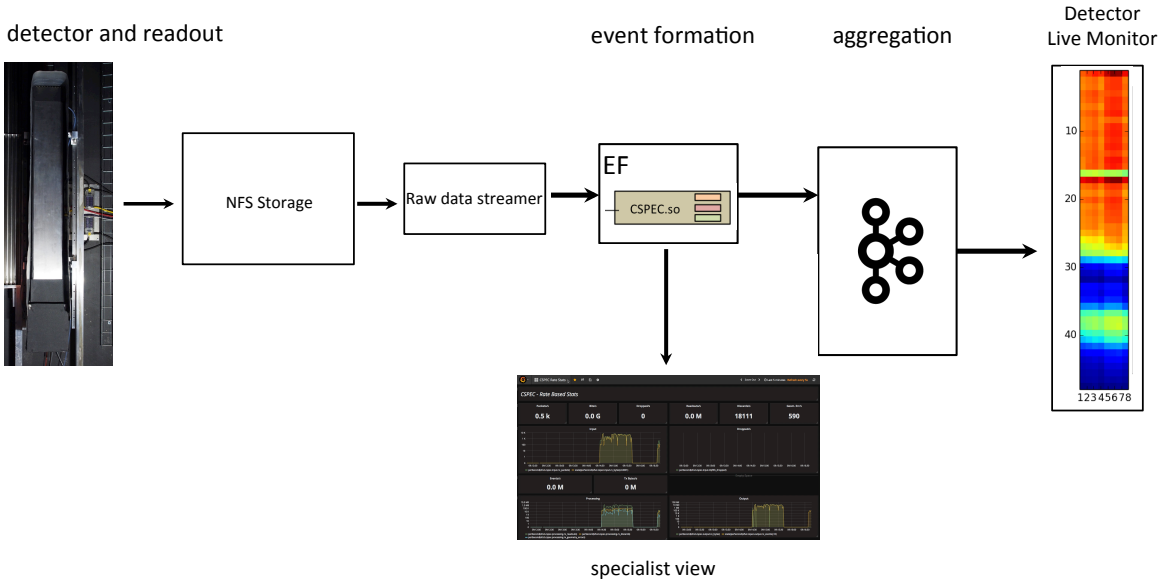


Figure 7: Delayed live streaming of Multi-Grid detector data

9.2 Multi-Blade

The ESS Multi-Blade detectors consist of several cassettes, each arranged with the wires representing the x-directions and the strips – the y-direction [9]. This is shown in Figure 8. The Multi-Blade detectors are similar to the Multi-Grid detectors in terms of their processing requirements. This assumption will be verified by developing a working prototype of the software event formation system in the period right after this report.

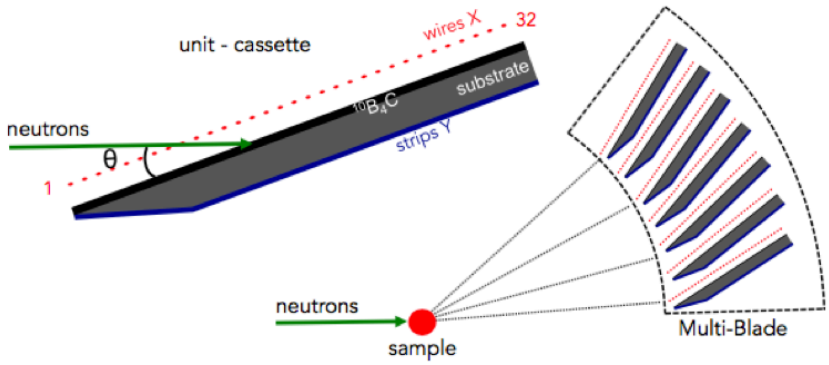


Figure 8: Illustration of the Multi-Blade detector design [9].

9.3 Jalousie

Three instruments will be based on commercial detectors developed by CDT. These Jalousie named detectors also detect neutrons using coincidence. This clustering could in principle be carried out by the event formation system. However, the CDT detectors are expected to implement their own clustering algorithm and deliver either clustered data or even event tuples. This reduces processing requirements for the event formation system. The event formation system will simply collect events, convert them to global pixel coordinates and forward them to the Kafka aggregator in the standard data transport format (Flatbuffer).

9.4 Gd-GEM

Because of its high spatial resolution and detection efficiency requirements, the Neutron Macromolecular Diffractometer (NMX) to be installed at ESS [1] is geared up to use novel detector technology that will require more complex data processing. A considerable effort has therefore been directed at ensuring the correctness of data processing for this detector type.

NMX will eventually have three independent detector panels of about 60 cm x 60 cm. Each panel consists of 4 detector modules like the one illustrated in Figure 9. The figure illustrates a Gas Electron Multiplier (GEM) based micro Time Projection Chamber (μTPC) detector. The cathode, bounding the drift gas at the top of the module, consists of a 250 um thick Gadolinium (Gd) foil, which has a high cross section for interactions with thermal neutrons. The interaction results in the production of a so-called conversion electron with energies ranging from 29 keV to 250 keV, with a peak around 70 keV. These electrons will, with some probability, enter the drift volume of the GEM, hereby ionizing the gas and liberating electrons from the gas atoms. The liberated electrons will then drift towards the amplification plates, which massively increase the number of electrons. Finally, the electrons arrive in the 2mm induction gap. The movement of the charges in the electrical field of the induction gap then induces a signal in the x and y strips of the readout.

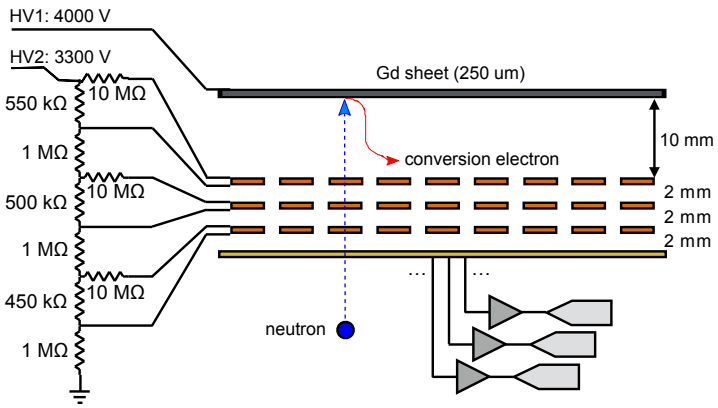


Figure 9: Schematic cross section of the GEM based μTCP detector [2]. Charge from the conversion electrons is collected on the x and y anode strips having a 0.4 mm pitch. The VMM3 chip amplifies and detects peak charge and time for every strip.

The readout strips of the GEM detector are connected to the read-out electronics, with a channel for each strip. Most prototypes so far have been using the APV25 chip, which has 128 channels that are triggered using the signal from the GEM foil closest to the anode of the detector.

The APV25 chip has a charge amplifier followed by a 50 ns CR-RC-type shaping amplifier followed by an analogue memory implemented as a pipeline buffer with 4 μsec latency. The

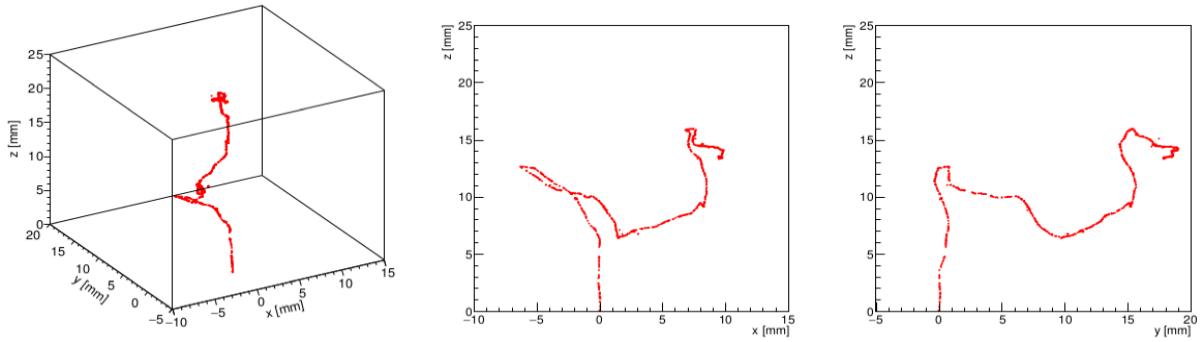
output of the shaping amplifier is sampled every 25 ns and stored in a buffer with a maximum length of n=30. That means that waveforms of 750 ns can be recorded. The time length of the waveform depends in addition to the shaping time on the drift length of the detector and the drift speed of the electrons, hence on the electrical fields.

The VMM3 chip which will be used for the final design for ESS operations, is a digital chip. It will not record the wave forms but only peak values and corresponding time. The pulse width of the shaper can be adjusted down to τ = 25 ns. After the recording of a maximum pulse, A/D conversion and buffering take place. From measurements done by the ATLAS experiment at CERN, the total dead time is expected to be about 200 ns. This results in an additional loss of information, thus reducing the required bandwidth, but that may make it difficult to identify certain traces correctly, for example C-type events (see Figure 13 for further details).

The entry point of the neutron must be determined preferably to 250 μm accuracy (Task 4.1). The entry point of the conversion electron, hence the start of the trace, can be taken to be the point where the neutron interacted with the Gd foil. The resulting algorithm and associated code, must be efficient enough to cope with the expected data rates.

9.4.1 Event classification

The conversion electrons from the neutron interaction with the Gd foil have relatively small kinetic energies. They will lose some of their energy as they traverse the Gd foil and will enter the drift volume with a mean energy of 70 keV. Due to this small energy level, the electrons may change their trajectory significantly as they traverse the gas in the drift volume. This is demonstrated in Figure 10 by Pfeiffer et al [2], which displays the result of a simulation of a low energy electron in a drift gas.



(a) 3D (b) Projection zx (c) Projection zy

Figure 10: A simulated trajectory of a 70 keV electron in an Ar/Co2 70/30 gas. Neutron conversion layer is at the bottom [2].

As can be seen, the electron may change directions significantly along its path. For this reason, we aim to characterize the events into a few representative types which are called event classes. The event formation algorithm, which locates the entry points of the conversion electrons, must be able to process all the different event classes or discard them, in order to achieve the high resolution required by the experiment.

Regular events are events where the electron has followed a path which is more or less linear. An example of such an event is presented in Figure 11. It shows the detected trace in x and y on the left and right panel respectively. The spatial axis is horizontal and time is plotted on the vertical axis, with later events at the top, corresponding to charge closer to the Gd conversion layer. For each strip, we see the waveform resulting from electrons hitting the strip. The dots indicate the location of the maximum charge recorded. It takes approximately 50 ns

from the first charge is recorded until the maximum is reached and then the output signal tapers off.

In Figure 11 no significant changes to the path direction occurred as it traversed the drift volume. Locating the entry point of electrons following such paths is straight forward, since the entry point can be taken to be the strip where the signal arrived at the strips the latest.

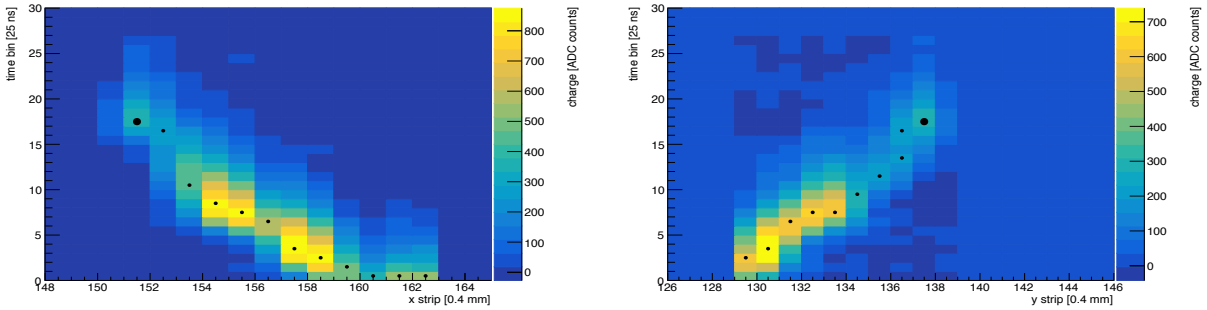


Figure 11: An example of a regular event (APV25 data). The charge is recorded in 25 ns time bins. The last arriving charge signal represents the neutron entry, (x,y)=(151,137).

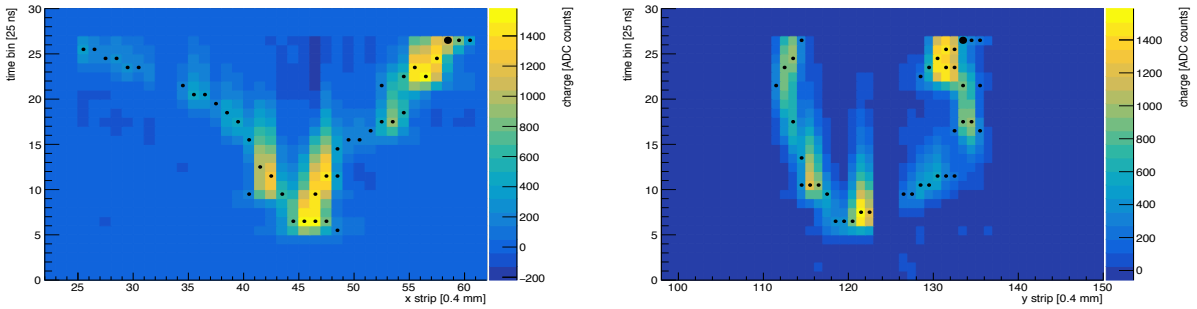


Figure 12: An example of a U-type event where the y-value of the neutron entry is difficult to determine (APV25 data).

The situation in Figure 12 is more complicated. The electron starts by moving downward, but changes its direction to move upward again towards the cathode. This will give two possibilities as the entry point for the electron. The signals are *U-shaped* and the y-strips in Figure 12 have two possible entry locations, either y=114 or y=133, 134 or 135. Choosing the wrong branch will lead to an error in the reconstruction. The entry location of the x-strip is also ambiguous.

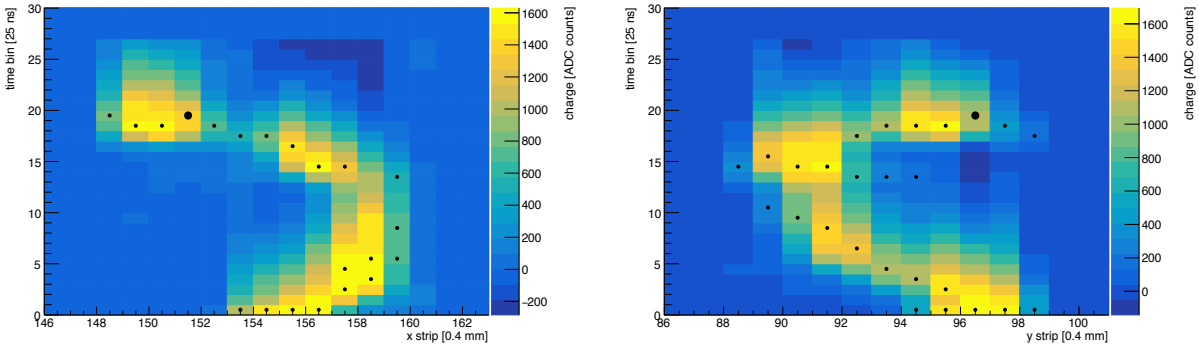


Figure 13: C-type event.

In the case of a *C-type* event, several strips are hit twice or even more times during the neutron event resulting in a C-shaped trace. Locating the entry point from C-type events, is straight forward when the GEM has been read out by the APV25 chip, since the entry point again can be taken to be the strip, where the signal arrived the latest. However, these events may be problematic when recorded with the VMM3 chip, which has a certain dead time after recording a charge maximum. Figure 13 shows an example of a C-type event.

In the general case charge traces cannot be classified unambiguously and will show different characteristics simultaneously. We will introduce a number of additional classification measures later when discussing filtering of events by the prototype algorithm.

9.4.2 Simulations

A challenge analysing the experimental data to develop an algorithm is that the actual neutron entry point is not known thus quality of any given event formation algorithm cannot be quantified by ab initio methods. For this purpose, simulated neutron events are extremely valuable. The simulated traces of a neutron event start at a known point. Therefore, the quality of the reconstruction is straightforward to evaluate. A simulation system based on Geant4 and Garfield has reached production grade recently, so the results are preliminary.
The simulation system interfaces GEANT4 [3] with Garfield [4]. GEANT4 simulates the interaction between the incoming neutron with the Gadolinium, thereby producing the conversion electron. In a subset of these simulated events, the conversion electron will enter the gas volume. GEANT4 then propagates the electron through the gas. While the electron traverses the gas, GEANT4 simulates the ionization and the production of conduction and delta electrons. Electrons with a kinetic energy below the threshold (usually set at 1keV) is stopped in GEANT4 and Garfield continues the simulation. Garfield then simulates the drift of the electrons due to the simulated electric field towards the induction zone where Garfield calculates the induced signal on the readout strips located opposite to the Gadolinium converter in the GEM [5]. The induced signals are then convoluted with a response function to mimic the shaper circuitry. The amplification region of the detector, and the diffusion caused by the three GEM foils has not yet been considered in the simulations. Noise from the electronics is also currently not simulated. However, plans do exist to include the simulation of electronics noise, should further studies show that it will provide a better understanding of the NMX GEM signals.

The simulation in the configuration described above will produce and propagate conversion electrons with different energies up to about 250keV. The simulation can be configured to by-pass the neutron interaction. Instead the simulation initiates with electrons with a pre-set energy at the boundary between the Gadolinium and the gas volume. This has the advantage of studying the dependence of the electron energy with the precision of which the entry point can be determined.

Figure 14 shows an example of a simulated event with multiple entry candidates. The correct entry point for both x and y-strips is 125. Simple algorithms get the y-strip wrong.

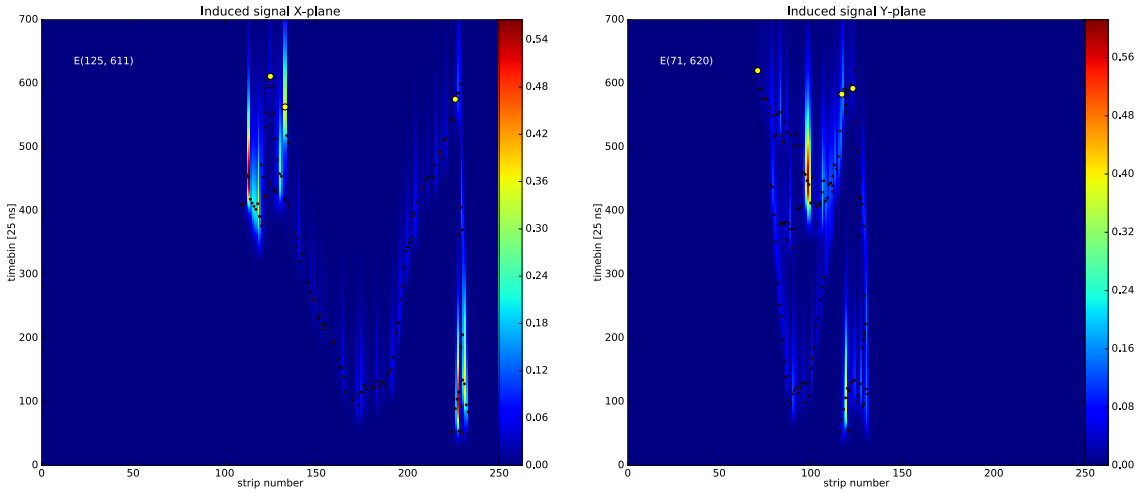


Figure 14: Simulated U-type event.

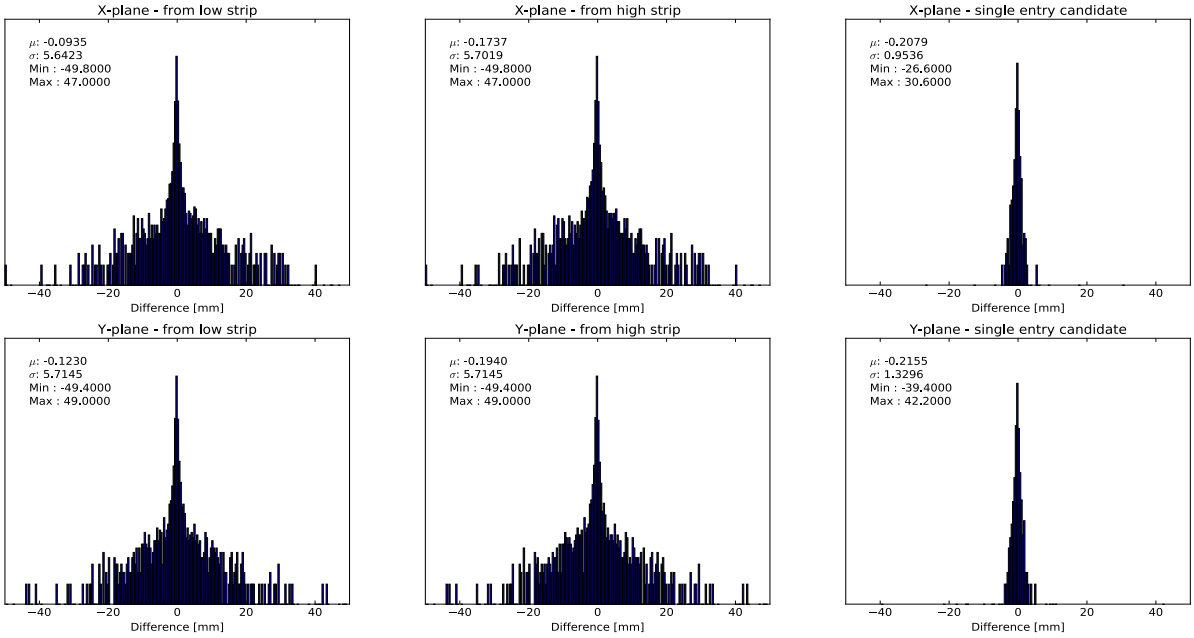


Figure 15: Simulated errors of event formation.

10,000 traces were simulated and the entry point reconstructed using two slightly different strategies (low and high). The first 2x2 panels of Figure 15 show the histograms in log scale of the differences between the neutrons interaction points and the found entry points. While the average is close to zero, as expected, there is a significant spread in the distribution. This spread will blur diffraction patterns for NMX users.

Initial studies indicate that the larger differences have two main causes, namely choice of the wrong branch for a U-type event and electrons generated by gamma rays. This can be seen in the last two panels of the figure which is the same measure as before, but where events with multiple entry candidates and electrons from gamma rays are excluded.

So far, no algorithm exists which can properly detect the correct entry candidate for U-type events so a robust event formation algorithm should exclude these events entirely. The electrons from gamma rays are difficult to identify and reject and will result in a certain number of misidentified neutron entries. However, very recent results show that it is possible to reduce

the number of gamma generated electrons considerably at the expense of a comparable number of conversion electrons. Thus, sensitivity is traded for accuracy.

One would expect that the relatively rich data from the APV25 would allow for more nuanced analysis and qualitative evaluation of the events. The influence of the VMM3 latency time was evaluated by assuming the recording of only the maxima per strip permitted by the dead time. A study was done using almost 100,000 simulated traces to enable a precise evaluation of the different data types. Three different reconstructions were performed. The first is based on APV25 like data, i.e. all extremes are used. The next two are based on VMM3 like data with dead times of 200 ns and 400 ns, respectively. The quality of the reconstructions was also evaluated by histogramming the error in millimeters of the event formation. Standard deviation is evaluated for the x-strips and y-strips separately. For APV25 like data we get (σx, σy)=(7.20, 7.14). For VMM3 like data and a dead time of 200 ns we get (σx, σy)=(7.23, 7.19) and with dead time 400 ns (σx, σy)=(7.34, 7.31). It shows that the difference between event formation based on APV25 data and VMM3 data for a latency of 200 ns is insignificant. In all cases approximately 18% of the events were discarded due to entry point ambiguity. These results indicate that the expected data reduction performed by the VMM3 would be useful rather than detrimental with regards to the uTPC approach.

It should be noted that the simulation has been implemented into the general ESS computing framework, enabling usage for other detector development projects. For this reason, the simulation has been constructed for detailed configuration with minimal effort.

9.4.3 Analysis of Experimental Data from Prototypes

Raw data from recent experiments carried out with GEM detectors as part of Work Package 4 [7] has come in various formats depending on the readout electronics used. To facilitate analysis with existing tools and simplify the development of new ones, a tools has ben developed to convert all data to HDF5, which is a standard in many physics applications for complex structured data. We have adopted the applicable best practices from the NeXus standard for neutron scattering data in designing an appropriate data format, including descriptive metadata to facilitate extraction and interpretation of data.

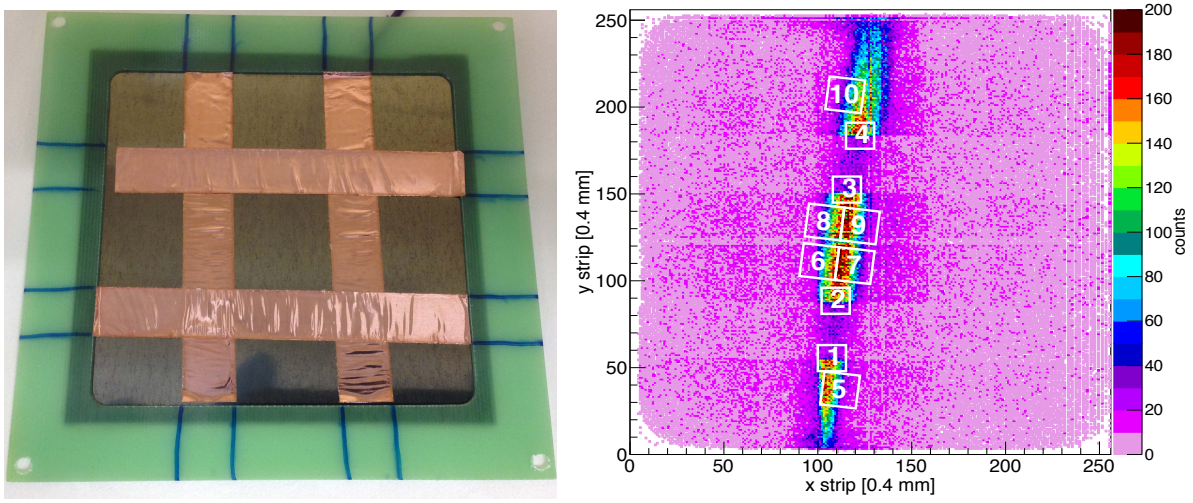


Figure 16: The Gd converter with copper tape (left) to give a distinct picture (right) [2].

Figure 16 (left) shows a Gd converter where part of the surface has been masked off with copper tape to prevent conversion electrons from escaping, thus creating some distinct features in the reconstructed image. The data has been produced by the event formation

algorithm of [2], which gives good results. Despite the presence of the copper tape, electrons are detected everywhere and filtering events may improve the quality.

To date, about 110 experimental runs have been processed and analyzed datasets, which includes around 110 million conversion electron tracks (250 GB of raw data). Most experiments have been performed using the APV25 chip. As compared to the VMM3 chips to be used in the final setup, this chip is slower but provides more detailed data, and the traces are conveniently triggered. This precludes the need for a solution of the clustering problem for NMX detectors at this point. We also emulated the behavior of the VMM3 chip in software by reducing experimental APV25 data using certain preset thresholds. All data from recent runs has been converted to emulated VMM data, resulting in a large volume of data to test prototype algorithms offline and in the Event Formation chain. In the following sections, we will refer to datasets produced with the APV25 chip and the reduced VMM-like data. Though it may be a good approximation, we make no claim that the data is representative of the data that will be received in the final electronics setup.

The EFU will receive conversion electron traces from the detector electronics, akin to those shown in Figure 11. The uTPC (micro time projection chamber) approach may be applied to each projection to determine the x- and y-coordinates of the start of the track. There may be some benefit in evaluating both x- and y-coordinates in combination. In practice that has not shown easy, especially for (emulated) VMM data. For now, both dimensions are treated independently and the discussion here is reduced to only one projection at a time.

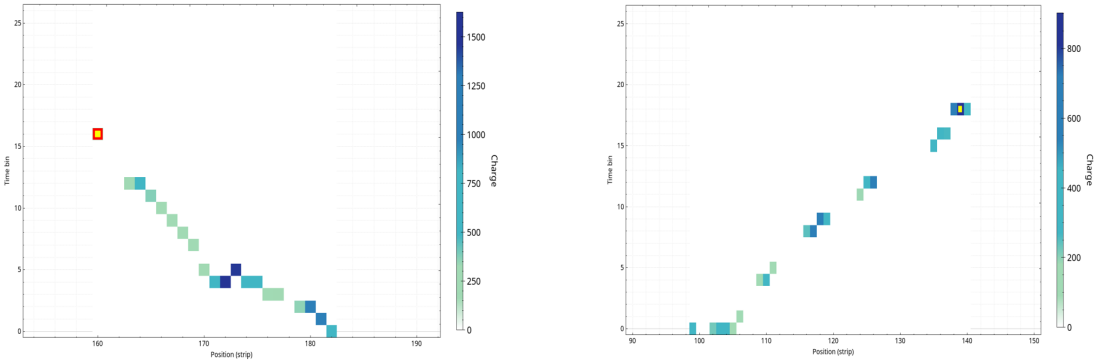


Figure 17: Linear event with one and only one latest data point (reduced VMM-like data). The latest point is in red (left). Linear event with multiple latest data points (emulated VMM3 data). The weighted average of latest points is in yellow (right).

As previously implemented, the uTPC approach is quite reliable when particle traces are linear. It has already been shown that a simple implementation of the algorithm performs better than center of gravity for the entire particle track [2]. The correctness of this approach can be confirmed for events that have one and only one latest point in a track.

As shown by the simulations, any further attempts to remove uncertainty in the event formation hinge upon how particle tracks that have ambiguous entry points (U shape) can be filtered out. As a first step to filtering, a number of scalar quality indicators were developed for readout clusters. This allows us to have a quantitative classification of events and get a handle on the quality of all events in any dataset by building histograms of these metrics.

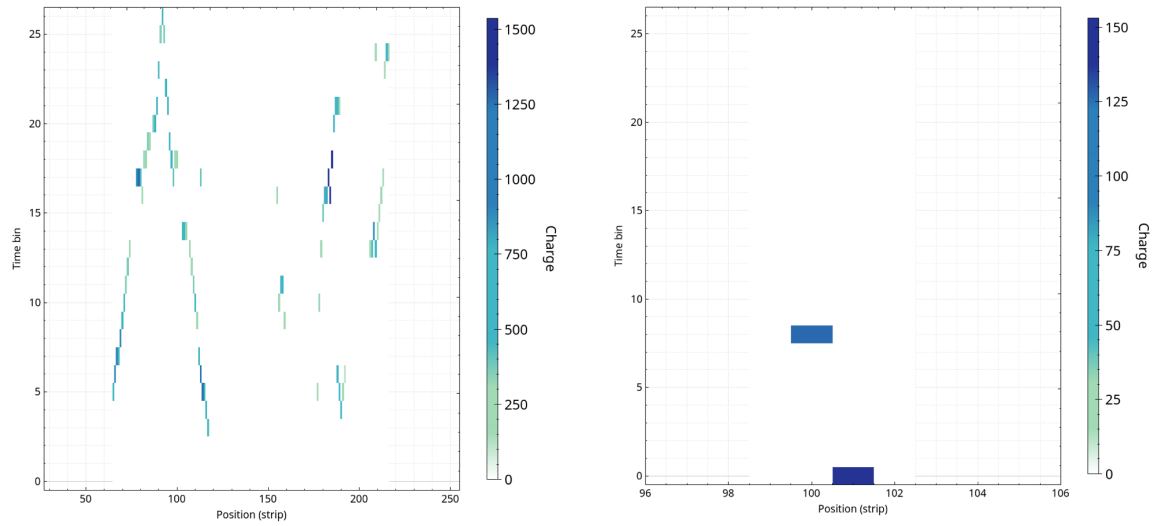


Figure 18: The "integral" metric indicates total charge collected in a particle track. Value = 41990 (left). Value = 127 (right).

To elaborate this on a simple example, one may consider the integral of all ADC values in a cluster as a metric. With some assumptions about detector efficiency and reliability of clustering, this should - as a first approximation - be proportional to the charge deposited by the conversion electron and hence its initial kinetic energy. Figure 18 shows two examples with very different values. On the left it appears multiple particles have ionized the same gas volume. As a result, we may want to reject events with very high integral. Conversely, some clusters with very low integral (right panel) could be indicative of electronics noise or electrons produced by gamma rays, so they may also be rejected. Rejecting such events could reduce noise, which is something we will examine in the following sections.

Figure 19 shows a histogram of the metric for all events in an experimental dataset (or across multiple datasets) to examine the distribution of a particular event quality. The interactive analysis tools allow us to select a particular range of metric values and view only events with those particular qualities. This helps to isolate "irregular" events and evaluate the correctness of the intended metric's implementation.

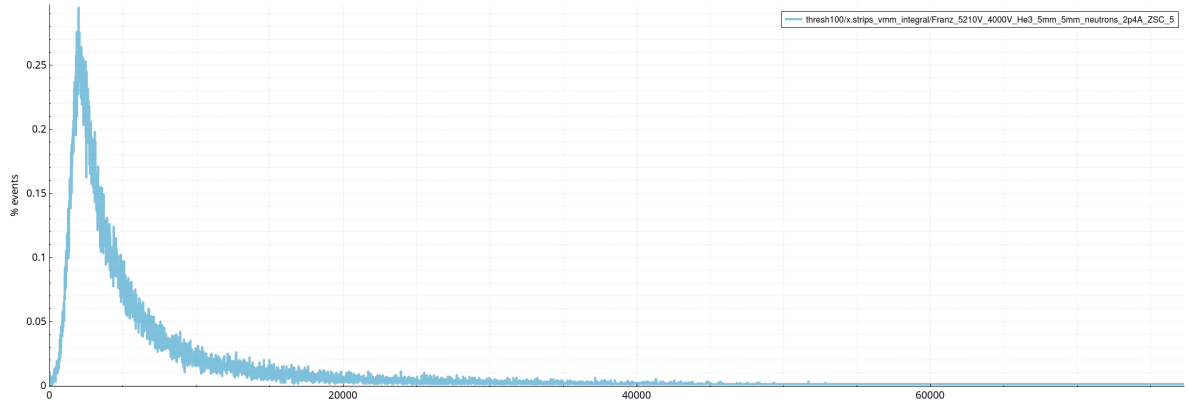


Figure 19: Normalized histogram of the "integral" metric for 200k events from a collimated neutron beam. The x-axis is the value of the metric.

We have developed a variety of metrics, most of which are constructed from the following simple and predictable values:

valid	number of strips with valid data
valid_points	number of valid readout events with ADC > 0
span	span of strip values, i.e. max – min + 1
density	fraction of span that has valid data, reported as percent, i.e. valid/span*100
integral	sum of all ADC values in cluster
integral_density	normalized integral, i.e. integral / valid
integral_norm	alternative normalized integral, i.e. integral / valid_points
cuness	cness or uness, depending on chosen perspective (space or time)
cuness2	variation of above with additional threshold
average_c	mean (unweighted) average strip number
center_c	average of all points weighted by ADC
center_ortho_c	average of all points weighted by time bin

These are calculated for both dimensions of the readout events available (x and y) as well as of the difference between the two. Appendix A presents the upwards of 300 metrics generated for a conversion track, as well as detailed descriptions of the implementation of these metrics.

While no ab initio knowledge is available about the correct neutron position, we have examined the effect of using the metrics as filters on three criteria of interest: spatial resolution, count rate and signal-to-noise ratio. This is done by applying the same method as described in [2]: data is used from an experiment with a sharp edge between high and low neutron intensity regions (Figure 20 left), a region of interest is selected that isolates such an edge (Figure 20 right), a projection of the data (onto the y-axis in this case) is used to examine the sharpness of the edge by fitting an error function (Figure 21) using ROOT's Minuit optimizer. The error function is a smooth step function. The parameters of the fit function reveal the criteria of interest: resolution corresponds to slope of the step, background to vertical offset, signal to height. We follow the code's loose definition of signal-to-noise ratio (SNR) as signal height divided by vertical offset. While the numbers are different from traditional understanding of signal-to-noise, this measure follows the same trend and purpose. In future we will change to equation. These three criteria are of consequence for the quality of scientific data obtained from the instrument.

The following detailed data analysis is based on data from a neutron beam experiment performed in February 2015, with copper masking in a 2cm x 2cm pattern (similar to Figure 16 left) and lead shielding to minimize gamma interference. This dataset was chosen because it has a well-defined edge between regions of high and low neutron intensity. The edge is not perfectly aligned with the readout strips. For the reason conclusions about the intrinsic resolution limits of the physical detector would require additional processing steps. This is not the purpose of this exercise and hence our resolution values are only valid to judge the relative quality of our processing options.

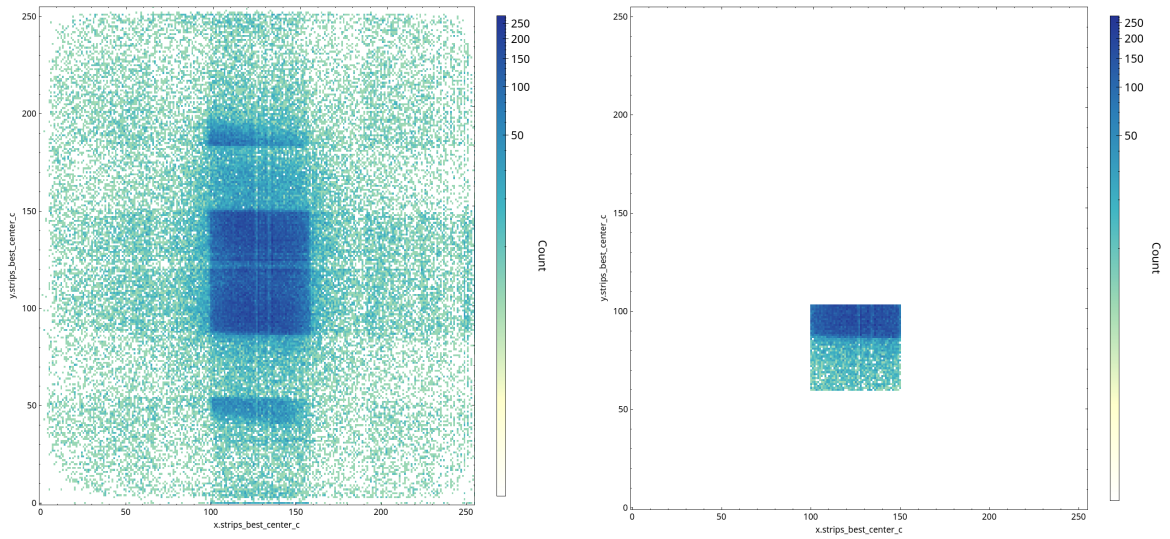


Figure 20: Reconstructed images from experimental dataset of prototype NMX detector. Full image from dataset (left). Image of selected edge region (right).

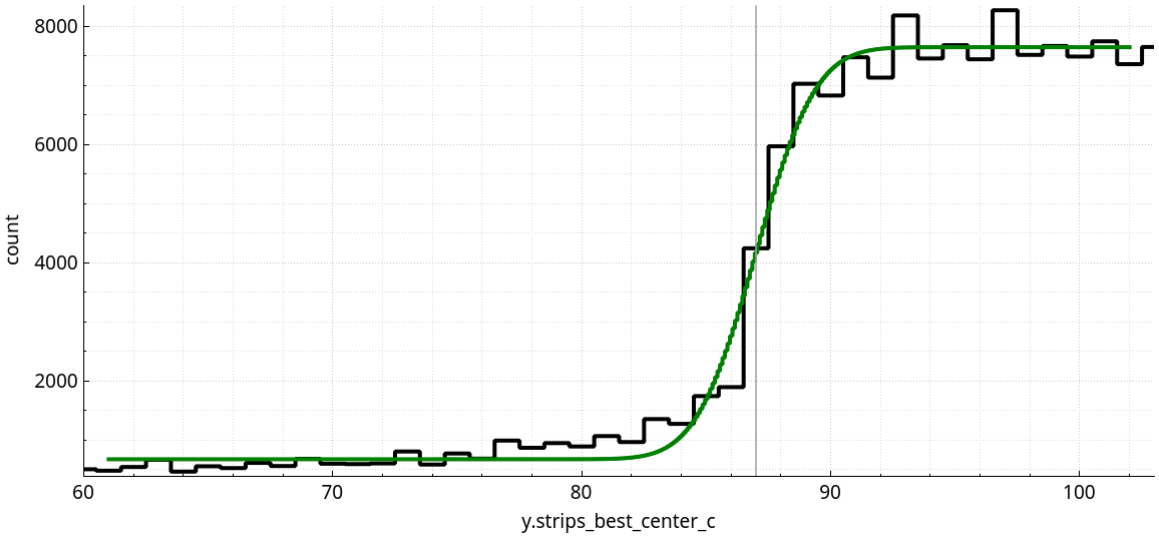


Figure 21: Projection of edge region onto the y-plane with fitted error function.

9.4.3.1 Example 1: Later span

The metric “later span” provides a reasonable uncertainty for the entry coordinate found by the uTPC baseline algorithm. The metric takes the readout events within the latest timeframes of the cluster and reports their maximum distance in strips. A higher the number correlates with a less reliable reported position. Consider the U-shaped event (Figure 22 left). There are two “branches” of the track towards the conversion plane and therefore ambiguity as to which of these represent the start of the particle path. The uncertainty reported by “later span” is the span of the highlighted points, in this case 57. In another case (Figure 22 center), the particle path is linear and does not curve back. The span of the highlighted points is only 7, thus indicating a much more reliable result. Consider another case (Figure 22 right), where the particle path may be curved, but these trajectory changes occur further from the conversion plane. The metric again reports a low number (5), as the latest points are clustered rather tightly in the critical area. Though this particle path may seem less “perfect” or typical than

that of the linear one, the reported uncertainty is lower. This is because the linear track (Figure 22 center) has a greater initial horizontal component, making the start of the track more spread out.

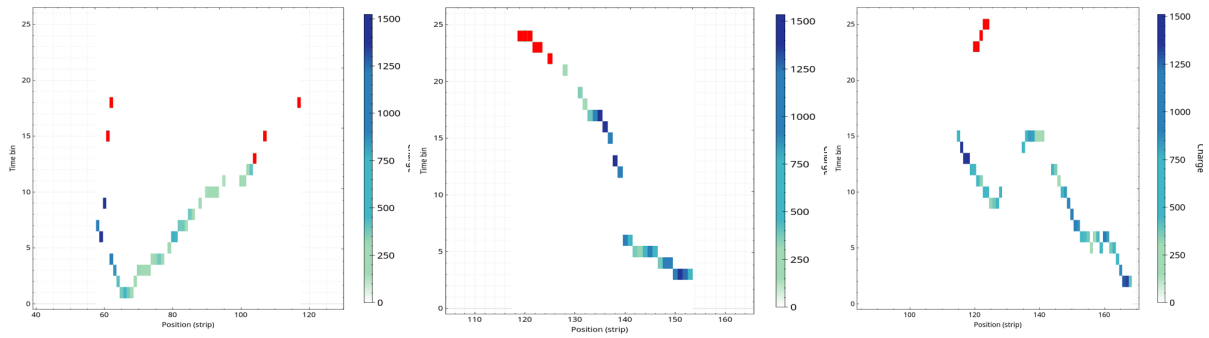


Figure 22: Events with "later" points highlighted in red. The span of these points constitutes an estimated uncertainty for the calculated entry coordinate. The values for these figures from left to right are 57, 7 and 5.

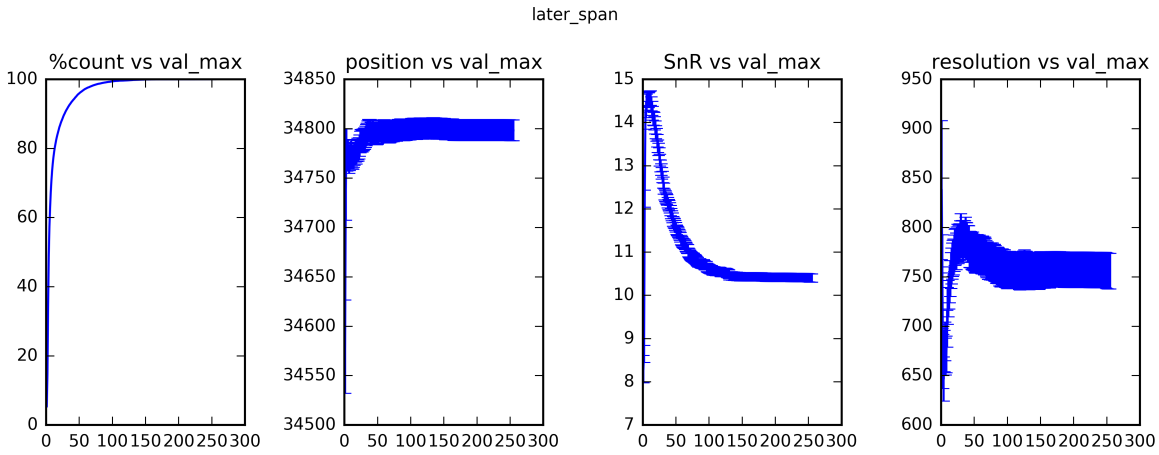


Figure 23: Evaluated criteria for the edge region as a function of maximum "later span"

We now examine the effect of filtering by this "later span" uncertainty quality indicator on the criteria of interest in Figure 23. The independent variable is the maximum allowed value (val_max in the plot) of the "later span" quality indicator for an event to be accepted into the projection and fit. We observe that as we relax the filter a higher fraction of the events is accepted. Signal-to-noise ratio peaks at rather strict filter settings – i.e. with values of "later span" at 12 or lower. Resolution also shows a global optimum in the same range of filter settings. The key observation is that efficiency is traded for signal clarity and/or resolution. Since the fraction of accepted events is always a monotonous function of the filter selectivity, it is reasonable to evaluate these costs by viewing fit-derived criteria in terms of the fraction of events accepted (Figure 24). Although different quality indicators may have different value ranges, efficiency is the currency of event filtering. This change in axis will help us compare the various metrics with each other, as we shall see in later sections.

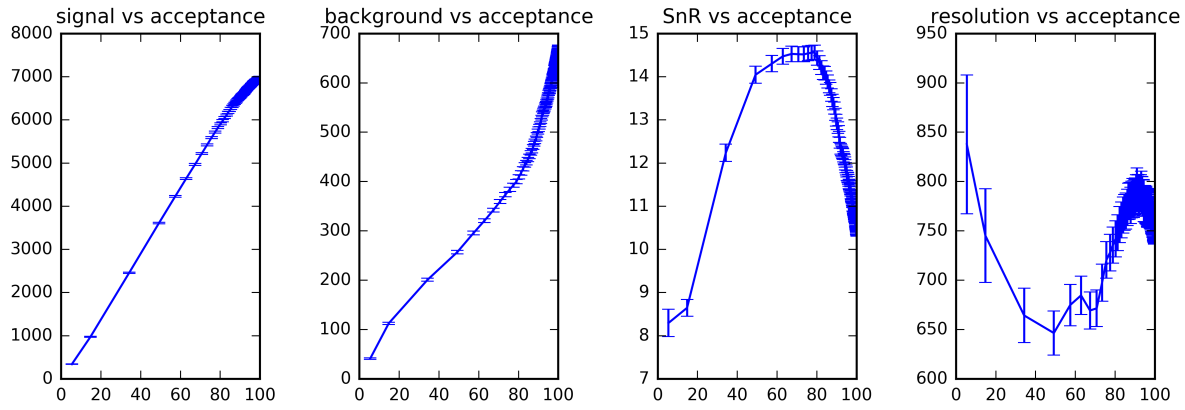


Figure 24: Evaluated criteria as a function of the % of events in the edge region accepted using the "later span" filter

Take note of the background count in comparison to signal. At first it grows slower as the filter bounds are relaxed, before it starts growing faster. These inflections underlie the presence of the optimal region for both signal-to-noise (plateau) and the resolution (valley). Good values for both can be ensured with as high as 80% relative efficiency.

9.4.3.2 Example 2: Spatial density

Another approach was to evaluate events where charge deposition is not spatially contiguous. To find such events, a "density" indicator is defined as percentage of strips with valid data out of the total strip span (see Figure 25). The ADC value of each point is ignored, and only their spatial density on the x axis is taken into account. Higher values are better, as they indicate a more continuous event, whereas lower values are worse. We should therefore apply a minimum threshold filter for this metric.

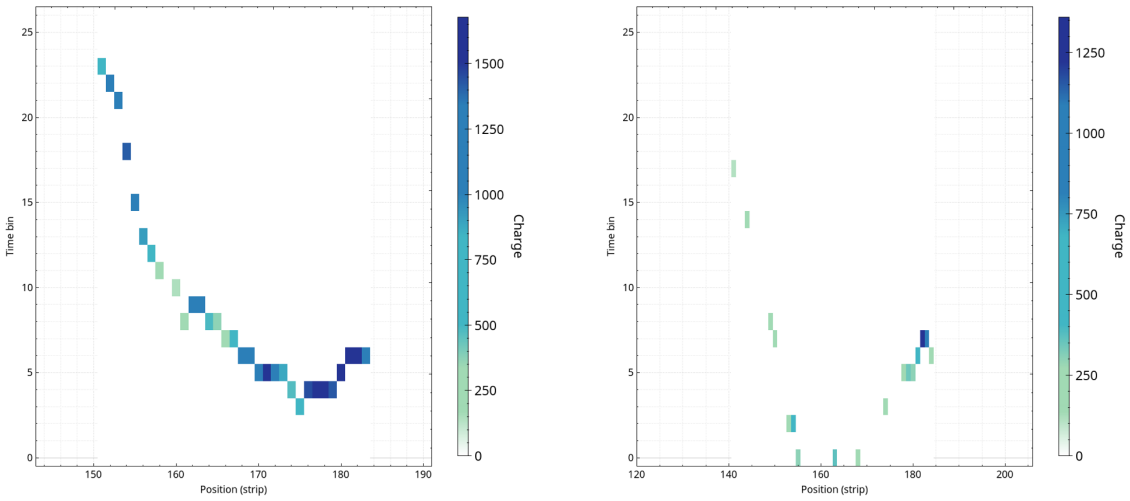


Figure 25: Density. Left: 100, right: 50

We can vary this filter and observe its effects, as was done above for “later span” (Figure 26). Effect on the resolution is inconclusive. The signal to noise shows a nearly linear inverse trend, but the gains are expensive.

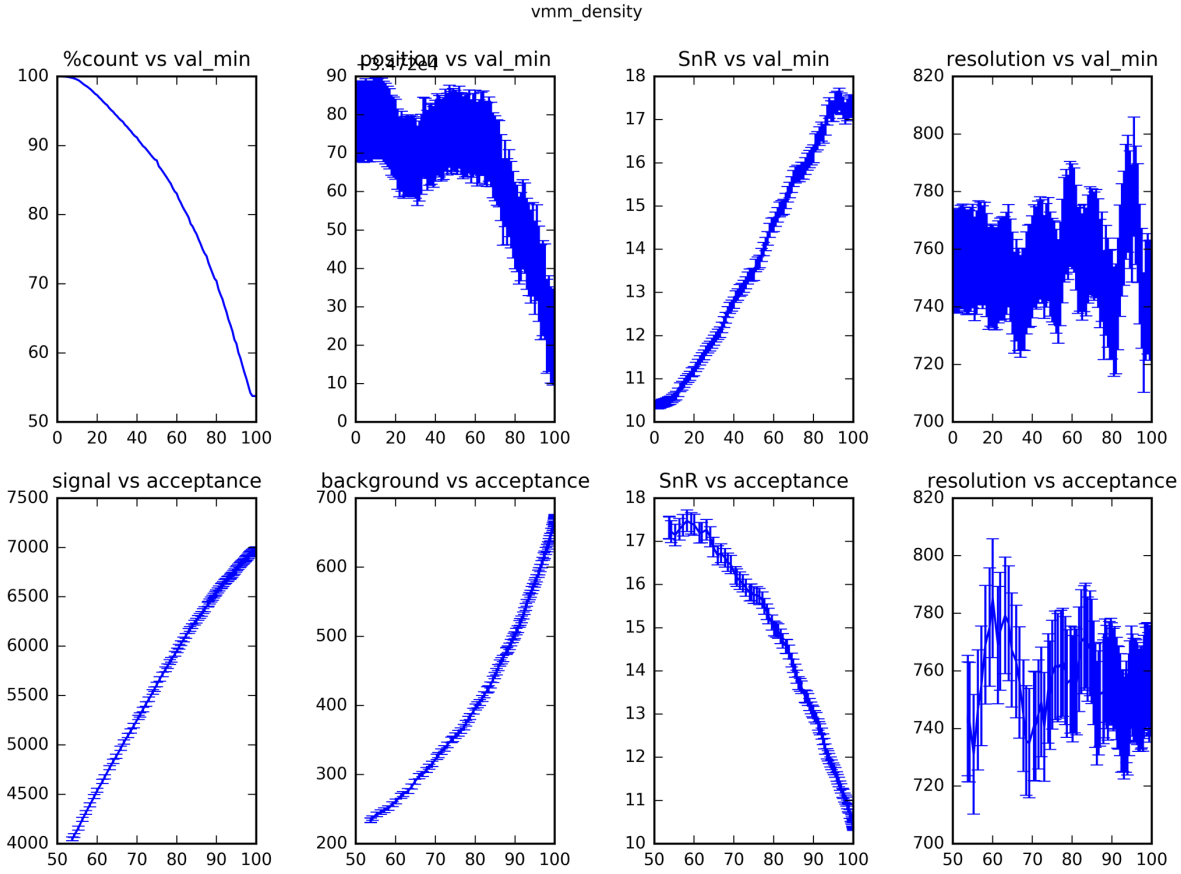


Figure 26: Evaluated criteria for filtering events using the density of the reduced VMM-like points.

Figure 27 shows a similar metric. The same definition of “spatial density” is used, but rather than use all points in the cluster as above, we use only the “later” points (highlighted in red in Figure 22). The plots on the bottom row are used to compare this filter to the previously discussed "later span". Note that it offers similar trade-offs with the least expensive gains possible at around 80% efficiency. Note also the inflections in the signal and background curves that dictate the optimum range. The exact values for obtaining such signal quality can be found by looking at the top row of plots.

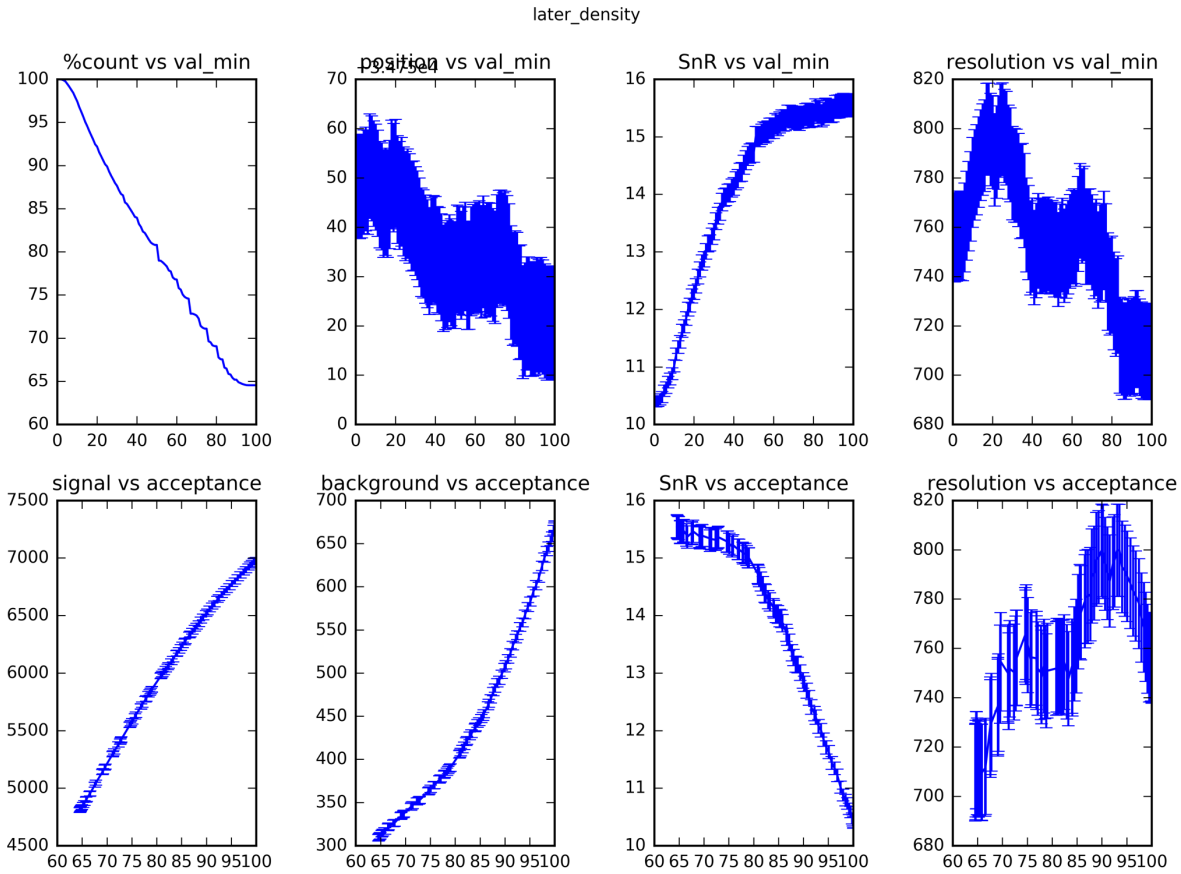


Figure 27 Evaluated criteria for filtering events using the density of the “later” points.

9.4.3.3 Example 3: c-ness

In extension to the quantitative description of C-type events earlier, we also define a "c-ness" metric. This quality indicator reflects the number of strips that have more than one readout event at different points in time (precise implementation details in Appendix A). Higher values indicate less reliability; therefore, we apply the maximum threshold filter and generate the same set of plots (Figure 27). We also observe the steep growth in background signal in the upper acceptance percentile. In this case, we can apply a very loose filter (low efficiency cost, above 95%) for noticeable gains in signal clarity. As previously noted, VMM3 data may not reveal these features in the same way, so the metric will have to be reevaluated with data from later detector prototypes.

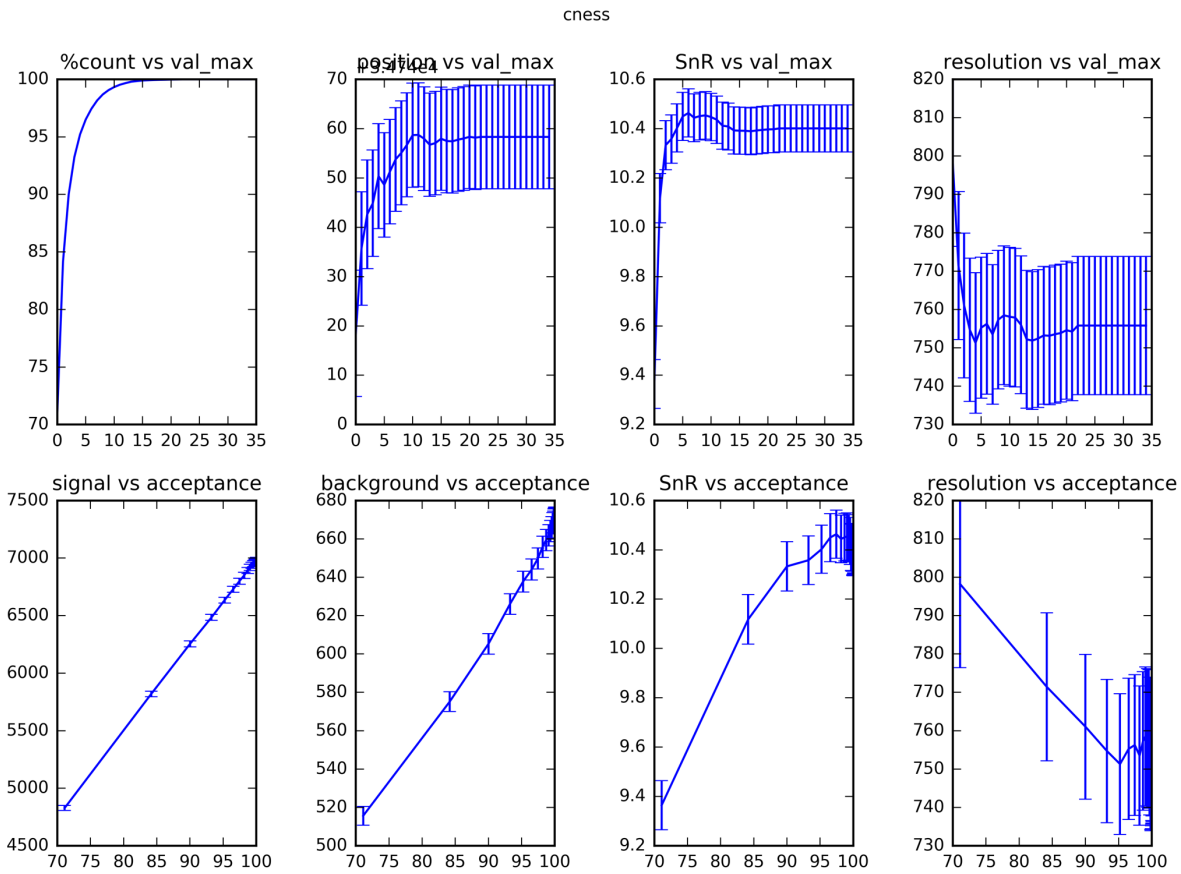


Figure 28: Evaluated criteria for filtering events using the "c-ness" metric.

9.4.3.4 Example 4: u-ness

U-ness can also be quantified. The approach is the same as for C-ness, but with change of perspective, finding coincident readout events in one time bin but multiple strips (implementation details in Appendix A). Again, higher values should indicate less reliable events, assuming that more curved events are likely to be more ambiguous. After applying the maximum filter, we see that the case is not so simple (Figure 29). SNR appears to have an optimum in a similar range to previously evaluated metrics, but resolution does not respond in a likewise manner. This may be due to poor implementation choices for determining U-ness. A different reason for its limited effectiveness may be the previously observed fact that not all curved events with coincident points are necessarily ambiguous, particularly if trajectory changes do not invade the later timeframe. This may simply not be a relevant metric. There is also reason to believe that data from the VMM3 may not be dense enough to easily reveal such features.

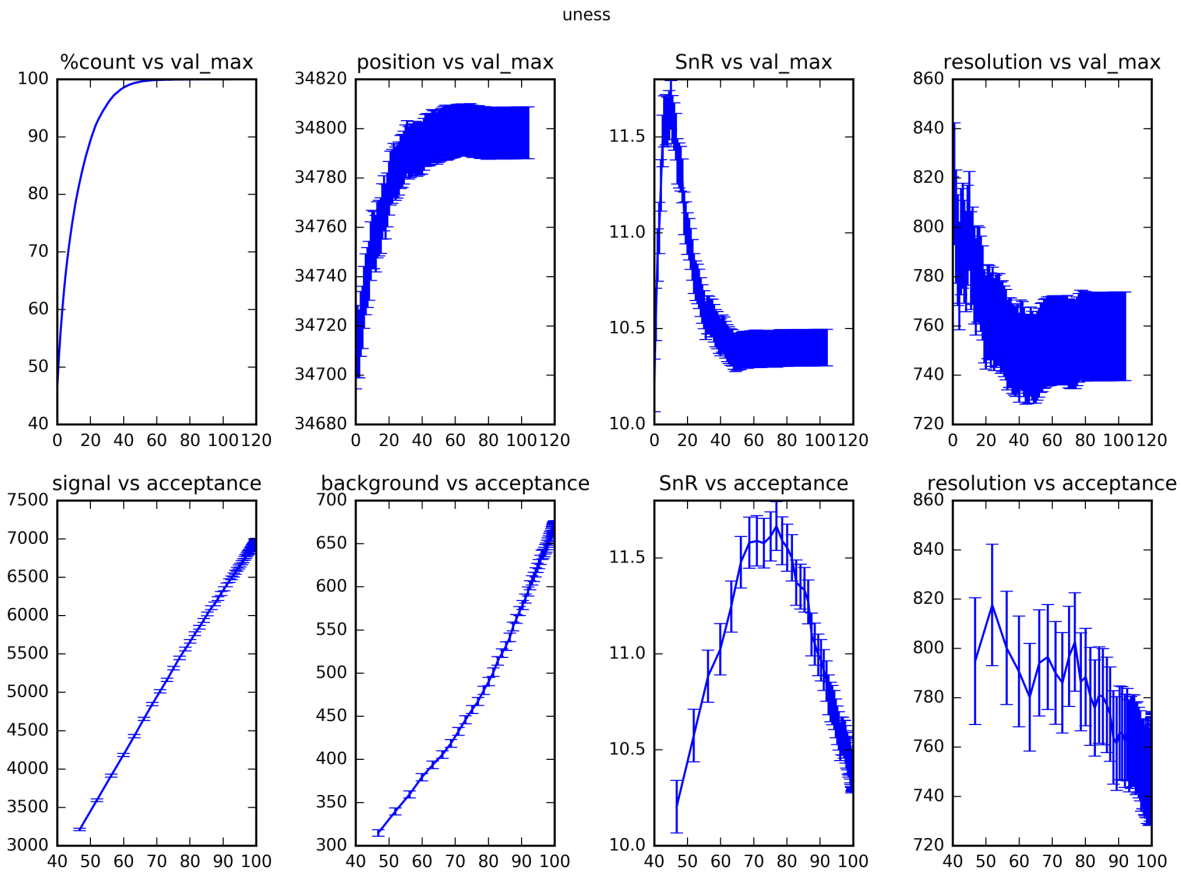


Figure 29: Evaluated criteria for filtering events using the "u-ness" metric.

9.4.3.5 Other filters

Some filters have proven not to be useful. For example, the sum of all ADC values reflects the total charge deposited, which is approximately proportional to the initial kinetic energy of the conversion electron.

Other filters are still under development and will be evaluated continuously throughout the detector development process, particularly as new prototype detector data is acquired using the intended VMM3-based electronics. See Appendix A for a full list of metrics currently available for event filtering.

9.4.3.6 Combining filters

To take full advantage of this event filtering approach, multiple filters can be applied in sequence. With this approach, the aim is for every individual filter to only take out a small targeted fraction of events in order to improve one of the quality metrics. In combination, this should result in a small loss in efficiency while optimizing the positive effect on the other figures of merit.

For example, in addition to our reasonable uncertainty value based on the "later" points, we may also define a lower uncertainty bound defined by the span of the very latest coincident points in a cluster. We call this the "latest span" filter, and we will similarly vary the upper bound as the independent variable. In this case, we have already filtered for events that have "later span" as 11 or less, and on top of this, we apply a second filter as described above.

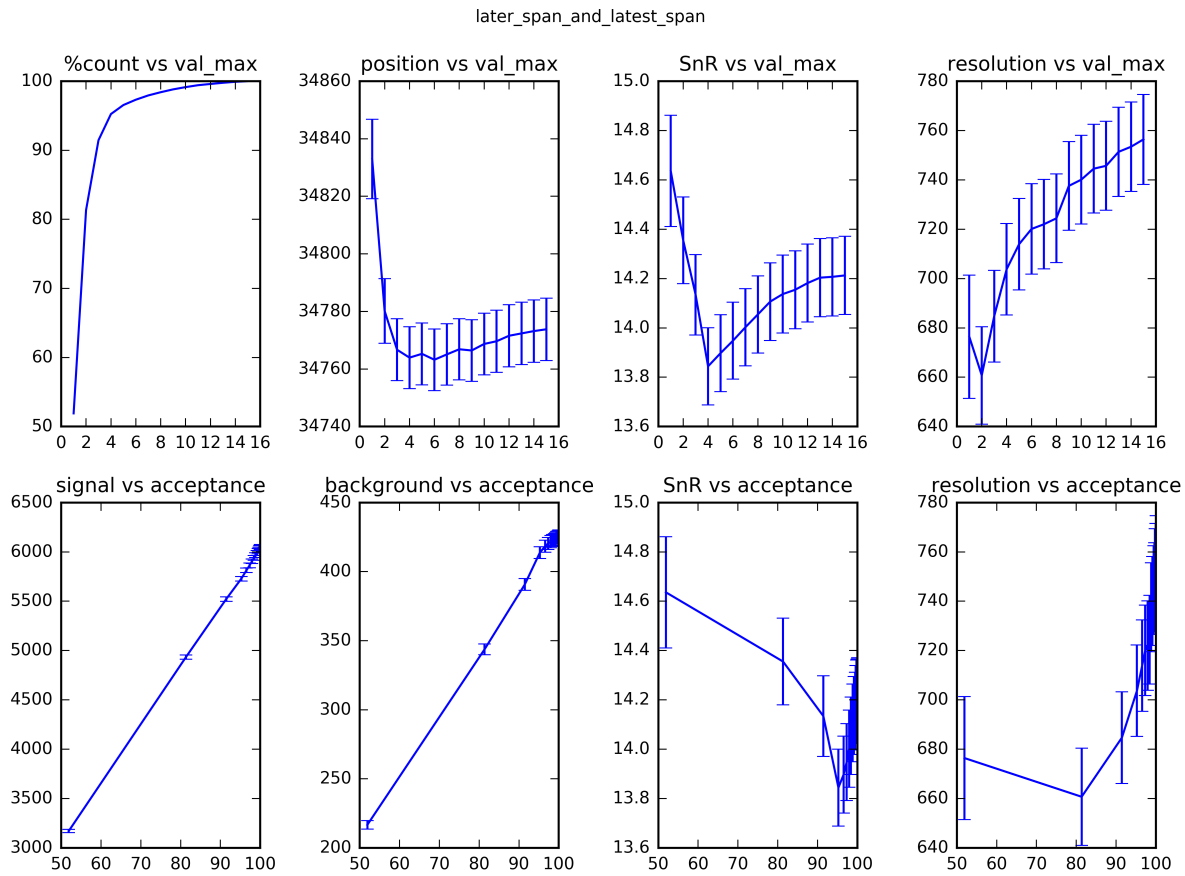


Figure 30: Evaluated criteria for filtering events using the "latest span" metric on top of a preset "later span" < 12 filter.

In Figure 30 we see different optima for SNR and resolution. The effect on signal to noise is not very pronounced. So, while this is not an impressive example, it shows that any number of filters can be applied in such a way, at a relatively small computational cost per filter.

9.4.4 Clustering

The clustering algorithm is still under development. The current approach is to dynamically join incoming readout events into spatially and chronologically adjacent clusters. A cluster is considered complete and released to the event formation step once readout events have been received that are newer than the timestamp of the current cluster minus the maximum latency on the readout event. That means we cannot expect any events to arrive that would be close in time to the cluster to be released.

An implementation of this approach has been tested with synthetic VMM3 data generated out of the same datasets as above. For data collected at (simulated) high neutron flux eventually we expect of chains of time-overlapping clusters. When there is no way to disentangle events, they have to be rejected, which has an equivalent signal reducing effect to the dead time of electronics. There are a few of open questions for example how to deal with potential time stamp overflow. On this we have recently entered fruitful discussions with the ESS Detector Group and its in kind partners developing the readout system for the detectors. A good understanding of common requirements was reached. It looks likely that timestamps can be high resolution 64bit absolute values, which would solve the problem on the hardware (FPGA) side. So we currently expect no serious problems preparing this processing step for operations.

9.4.5 Algorithm Summary

The above exploration of data has provided a set of components to incorporate into the event formation software for NMX detectors. We may summarize the above sections with an outline of the event formation algorithm for NMX neutron detectors as follows:

1) Perform clustering in x- and y-planes separately
2) Correlate coincident x- and y-clusters
3) Perform the following for clustered points in x- and y-planes separately:
   a) calculate entry point with baseline algorithm
   b) apply a optimized set of event filters
4) Return success and event coordinates if both planes are accepted

The above examined filters are included in the event formation algorithm, to be selectively enabled as filters to fine-tune data quality. Different filters may be preferred depending on experimental needs. If signal is quite weak or beam time is limited, in principle a user might prefer higher efficiency in favor of spatial resolution. Or, in some cases, higher special resolution may be preferred at all costs. While the focus has been on correctness and data quality, some computational costs of various processing choices will have to be considered.

It is also important to note that the final selection of filters and their settings will depend on the physical properties, environment and operations parameters of the detector. The properties of the VMM3 data may be different from what we currently have emulated. This will require the recalibration of optimal thresholds for these filters, a different selection of filters, or even the development of new filters.

We are confident that we can find optimal parameters because of the flexible software we have developed in arriving at the above results. We can review quality indicators for individual events, statistics within and across datasets, and to interactively examine the effect of applying different filters. This software will enable the instrument operators to fine-tune the event processing pipeline to changing detector architecture and varied user needs, and will serve as commissioning and calibration tools.

10 Event Formation System Architecture

The ESS Detector data path is illustrated in Figure 31: Experimental stations generate detector data which is transmitted over high speed Ethernet links. The detector data is collected at the event formation system. Here the collected readout data is converted into event tuples.

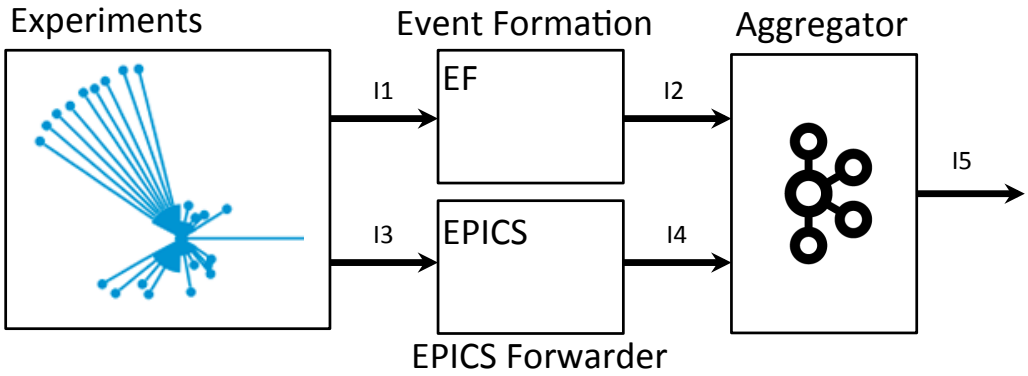


Figure 31: The four major components of the system (experiments, forwarder, event formation, aggregator) and their interfaces. The arrow indicates the main direction of data flow.

10.1 Event Formation System

The core system design is illustrated in Figure 32 and consists of an Event Formation Unit (EFU) for each detector panel. The EFU is capable of processing bulk data encapsulated in network packets at a high rate.

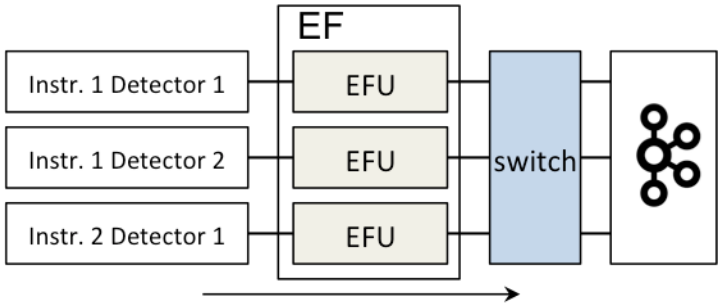


Figure 32: A single Event Formation Unit (EFU) receives bulk data from a specific detector panel, and forwards event tuples to the aggregator.

10.2 EFU Software

A single EFU will have the capability to process data from all instruments, thus all EFUs run identical software. However, it will only run a single event formation pipeline per instance.

The software comprises the Linux operating system and support processes as well as the event formation application itself. The Linux operating system will likely be based on CentOS. The event formation application could be a multi-threaded C++ Linux application using standard libraries. Performance critical parts may be written in a different language, like C.

The event formation application, once launched, will not require management. Its sole responsibility is to read, process and forward data. Each detector type could have a different

processing pipeline. The following section describes a possible pipeline for NMX detector data.

10.2.1 Event Formation Pipeline

The design of the current prototype pipeline for event formation is shown in Figure 33.

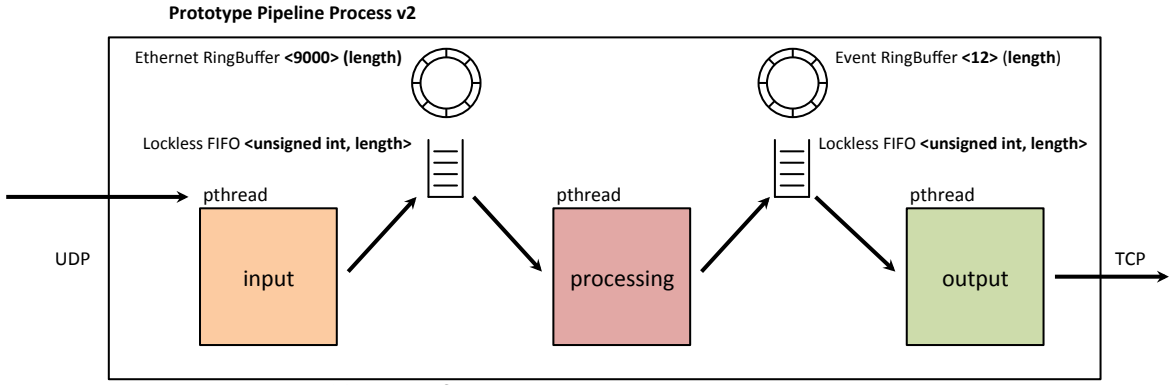


Figure 33: Event formation processing pipeline architecture.

As the event formation code matures, focus will shift to ensuring optimal performance. One obvious route to improve performance is parallelization of some independent processing steps. The more complex algorithms involved in NMX event formation are quite amenable to parallelization. Each clustered event is independent of the others and so at higher events rates this work can be delegated to multiple threads.

10.2.2 NMX Processing Pipeline

As the name implies, the pipeline consists of one or more steps connected in succession. Progressing to the next step in the pipeline will only occur when the previous step has completed. Completing successfully provides data to the next step, or in case of failure (insufficient data or timeout) by an error indication.

The processing pipeline is a dynamic object meaning that it is not part of the main event formation application. Rather the application can be instructed to load a specific pipeline during deployment.

The different elements of the NMX pipeline are shown in Figure 34. The pipeline is segmented into three logical groups: input, processing and output.

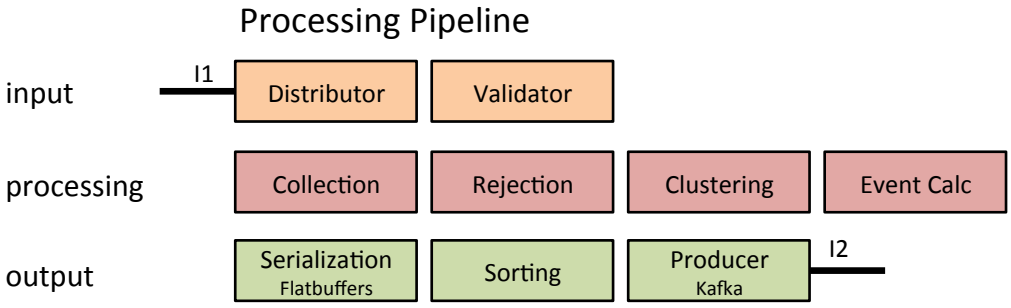


Figure 34: Processing pipeline (example).

Distributor
The distributor is responsible for receiving instrument data from the Ethernet interface and dispatching it for processing on a specific CPU.

There are several ways to achieve this with a delicate balance between performance, programming effort and maintainability. Three possibilities, presented in increasing order of complexity, are

- Using Linux native network stack and drivers
- Using Linux with an optimized network stack
- Custom execution environment and device drivers

An example of the second choice is Fastsocket, and for the third example, Intel Data Plane Development Kit (DPDK).

No choice of technology has been made. But it is preferred to use the Linux native protocol stack with fallback to some of the other methods if performance cannot be achieved.

Validator
The validator performs various sanity checks on the received data and rejects invalid packets. Checks could be valid Ethernet CRC, IP header checksums, data parity, etc. On modern NICs the Ethernet, IP and UDP checksums are typically calculated in hardware. However additional validations may have to be done in software.

Data Collection

The data collection step keeps reading from its input until there is sufficient data to proceed. This is only necessary if data is not already clustered when it is sent from the readout system.

Data rejection

This step could reject false positives in the detector data such as gamma ray induced ionization trails, electronic spikes and other phenomena.

Clustering analysis

This step identifies detector coincidences and groups readouts into clusters belonging to a single conversion event. The clustering step of event formation is a potential bottleneck. All data from a contiguous detector area or volume must be considered before the decision is made that clusters are independent of each other. A way to minimize this problem is to make sure that individual EFUs are responsible only for data from physically contiguous detector areas.

Event Calculation

This step calculates the detector position/detector id for the neutron based on the validated detector data.

Serialization

At the end of the processing pipeline, the result is typically contained in a C struct or a C++ object. The serialization step converts this data into a buffer used for network transmission. The serialization method chosen for this is Google FlatBuffers.

Sorting

It is possible that the pipeline will consist of a sorting step where event tuples are sorted based on detector id. Performance will be critical. Preliminary experiments indicate that a Priority Queue algorithm is a good candidate for this.

Kafka producer

Kafka has been chosen as the technology for data aggregation and the Kafka component is described elsewhere. The data producer is responsible for transmitting data to the Kafka cluster using Kafka's API. The producer must know the (IP) address of one of the Kafka brokers and the name and format of the topic to be published.

10.2.3 NMX Demonstrator Deployment

The WP5.1 partners have collaborated with the detector group on demonstrating the use of the event formation system software as a data acquisition system. This was done during a visit to GEM detector group at CERN. The setup was similar to the one used for the Multi-Grid demonstrator. The purpose was to integrate the event processing software prototype with the CERN developed Scalable Readout System and to provide monitor images and 'expert views' for use in the development of the Gd-GEM detector. For example, particle tracks were sampled once per second and displayed together with histograms of x- and y- strip hits and detector image.

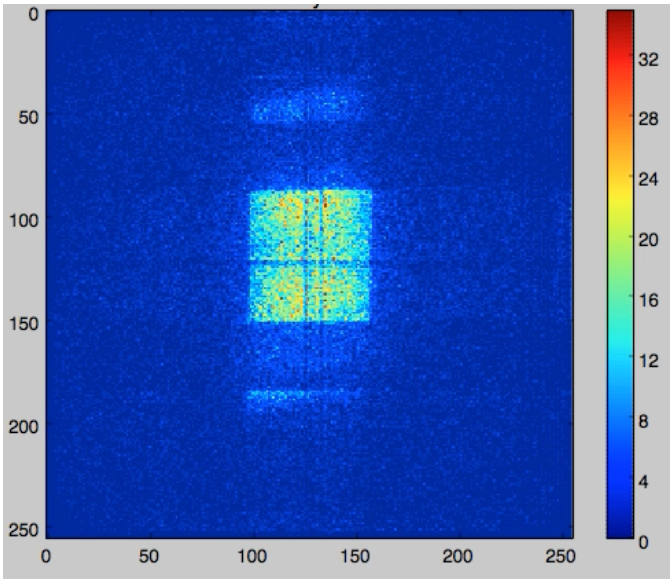


Figure 35: Screenshot of a visualisation prototype for the NMX detector

That means that we are already able to run the prototype event formation software in real time. While this lab test environment produces at count rates far below what ESS will generate at full accelerator power, this is still an impressive demonstration of the outcome of this task. Especially considering that in this demonstrator runs the entire architecture that is designed to be scaled out over multiple compute nodes on a single workstation.

10.3 Interface Specifications

Interface protocols and packet formats are specified in the following sub sections. There is one aspect, however, that can influence design considerations on data formats that must be addressed here. The Ethernet link has some fundamental limits in terms of packet sizes and inter-packet times that affect achievable data rates.

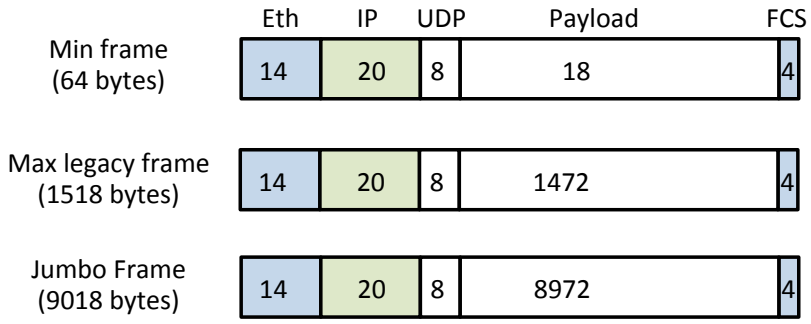


Figure 36: Increasing the MTU reduces the per-packet processing overhead.

A previous report on using Ethernet for detector data [10] uses standard, maximum sized, Ethernet frames of 1518 bytes, where [11] used jumbo frames of 7000 bytes and reported a 50% performance improvement.

Table 1 shows some calculations on packet rates, link efficiency and data rates for a 10 Gb/s Ethernet link. The assumption is that a bulk data tuple is 64 bit.

Table 1: Packet rate calculations for 10Gb Ethernet. An event size of 128 bits is assumed. The use of jumbo frames (> 1518 bytes) gives the highest throughput and the lowest per-packet processing overhead.

Frame Size (bytes)	Data Size (bytes)	Events (#)	Efficiency (%)	Pkt. Rate (M/s)	Data rate (Gbit/s)	Event Rate (M/s)
64	18	2	28	14,88	2,1	33
128	82	10	64	8,45	5,5	87
256	210	26	82	4,53	7,6	119
512	466	58	91	2,35	8,8	137
1518	1472	184	97	0,81	9,6	150
9018	8972	1122	99	0,14	9,9	155

From the calculations it can be seen that any data production (data tuples of size 64 bit) cannot exceed 155M tuples per second on a single 10Gb Ethernet link (last column and row). But this requires the full utilization of jumbo frames.

10.4 Network architecture

The proposed network topology and architecture is shown in Figure 37.

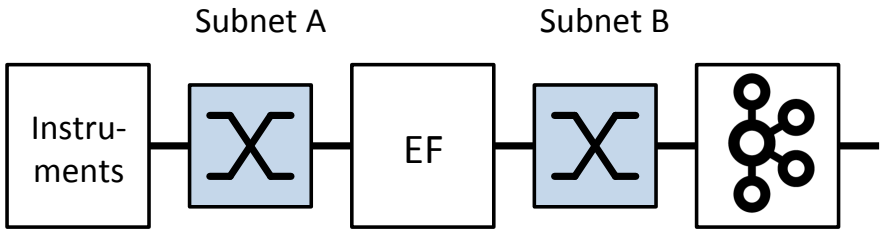


Figure 37: Network topology and architecture.

There is no need for routing protocols since packets are not forwarded. Rather they are produced at the output interface and consumed at the input interface.

On the detector interface side (see Figure 37), we expect a single IP subnet. The IP address space should be chosen liberally for example a ‘class B’ or /16 subnet with room for ~ 65.000 nodes. It is highly unlikely that there will be a need for even a tenth of this, so there should be room for future growth. The aggregator interface will probably have the same properties but with a different IP subnet address range.

All switches should be managed, support jumbo frames of up to 9000 bytes to minimize the need for packet reassembly, and have support for VLAN and SNMP (or some other monitoring tool). Considering the supported number of nodes, the switch should eventually have room for a substantial number of MAC addresses. It should be at least twice the number of active nodes to prevent flooding. However, this is a part of the infrastructure that can start small and be upgraded later.

11 Risks and Mitigation Strategies

With the current rate of the progress, WP5 will be able to deliver the project goals of BrightnESS Task 5.1 as planned. In order to do so we need to keep a close eye on the following risks that are considered to have the highest potential impact. The risks can be grouped into three main areas: (1) Staff, (2) communications and collaboration, and (3) technical risks. These are covered in sequence in the following paragraphs.

Software event formation is a very specialized subject and not something you can specifically hire staff for. If we lost one or more key people in the project, the resulting delays to re-hire and re-train staff would pose a major problem and make full success unlikely. We aim to keep the tasks interesting and the work environment attractive, but we also need to keep an eye on a fair distribution of workloads to avoid imbalance as well as sharing of knowledge to avoid single failure points.

With respect to communication and collaboration, we currently rely on the established fruitful dialogue with the ESS detector group on the other side of the interface. Keeping this relationship stable will allow us to continue to synchronously deliver our respective tasks. For this we have set up a number of recurring meetings and other joint activities. The actual delivery of some components will involve further in kind partners from ESS instrument projects. Those need to be aware of interface requirements and be involved in ongoing dialogues. A meeting to kick-start this, is planned for the next IKON Meeting in Lund in September 2017. Internally, activities also need to be follow common goals and converge according to plans. For this the WP5 partners agreed to make better use of the task tracking in the Altassian tool suite (Jira).

On the technical side, we have mastered some important issues. There is always potential for unexpected problems. To allow some contingency for any resulting delays it is important to stick to an ambitious schedule whenever possible. In the second half of the year we will start to look into performance bottlenecks. Our first idea is to attempt parallelization of processing should this not scale as well as expected, for example due to increasing communication needs, acceleration using GPUs or other more hardware specific means might be a fallback option.

12 Conclusion

We have presented our current understanding of the processing needs for the expected ESS detector suite. In the majority of the cases specifications of upcoming detectors and their raw output format are not in a state that would allow software prototyping. However, during the course of the project so far, we have developed a good working relationship with the detector group and their partners. That led to a good understanding of the domain and not least to two working prototypes of the event formation system for NMX and the Multi-Grid detector ahead of schedule. These two detector systems are a good template for future customized implementations. Most systems are quite similar to the Multi-Grid detector and with NMX we are close to having covered the most complex computational needs. Giving where we are in the development lifecycle of ESS detectors, we have as good an understanding of the processing needs for the different detector types as possible.

With a common and modular framework in place to drive the detector type specific processing algorithms the project has future tasks divided into manageable modular chunks. Partners can focus on implementing individual algorithms, improving their performance, testing the correctness, etc without unnecessary coordination. This report hardly mentioned the output side of the event processing pipelines. This is an indication that the choice of Kafka as a

commercial grade (but open source) message broker in task 5.3 was good. We arrived at a working prototype quickly and could focus on the domain specific problems.

At the moment, none of the prototypes use any processing optimized to a particular platform or hardware, like GPUs or accelerator cards. Should the performance requirements be difficult to meet, this is still an option. However, if possible we would like to avoid 'lock-in' to specific hardware components that may become unavailable or prohibitively expensive during the expected lifetime of the ESS facility. Removing this maintenance stumbling block is one of the reasons behind running event formation in software.

With the tools developed, we are confident that we can adapt processing algorithms and parameters in sync with any new hardware prototypes that become available, any updates to prototype detectors and production versions. We will review quality indicators for individual events, statistics within and across datasets, and to interactively examine the effect of different filters or code changes. The software will also enable the specialist from DMSC or the detector group to fine-tune the event processing pipeline to changing detector architecture and varied user needs, and will serve as commissioning and calibration tools.

13 List of Publications

[1] M. Shetty, C. Søgaard, et al., *Event classification and performance diagnostics software for GEM neutron detectors*, 11th Conference on New Opportunities for Better User Group Software (NOBUGS 2016), https://indico.esss.lu.se/event/357/session/4/contribution/83

[2] T. Richter, M. Shetty et al., *Replacing Complex Detector Electronics with Scalable Software Solutions*, The 22nd Meeting of the International Collaboration on Advanced Neutron Sources (ICANS XXII 2017), http://icansxxii.iopconfs.org/IOP/media/uploaded/EVIOP/event_955/ICANSXXII_Abstracts.pdf

14 Appendix A

The following list represents the event quality metrics (henceforth "metrics" in this section) generated for a conversion track recorded on the x plane. The same metrics may be repeated for the y plane, as well as differences in the value of each metric between the two planes. In total we have been generating three times the metrics in the following table.

Event quality metric	Description
x.not_gamma	higher numbers indicate event less likely to be a gamma
x.strips_all_average_c	strip average (no weights) using valid ADC values
x.strips_all_center_c	strip center of gravity using valid ADC values
x.strips_all_center_ortho_c	time bin-weighted strip center of gravity using valid ADC values
x.strips_all_cuness	number of points above 1 in strips using valid ADC values
x.strips_all_cuness2	number of points above 1 in strips with span > cuness_min_span using valid ADC values
x.strips_all_density	% of strips in span with valid ADC values
x.strips_all_integral	integral of strips with valid ADC values
x.strips_all_integral_density	integral / valid for strip with valid ADC values
x.strips_all_integral_norm	integral / valid_points for strip with valid ADC values
x.strips_all_span	span of strips with valid ADC values
x.strips_all_valid	number of strips with valid ADC values

x.strips_all_valid_points	number of valid points in strips with valid ADC values
x.strips_latest_average_c	strip average (no weights) using latest VMM maxima
x.strips_latest_center_c	strip center of gravity using latest VMM maxima
x.strips_latest_center_ortho_c	time bin-weighted strip center of gravity using latest VMM maxima
x.strips_latest_cuness	number of points above 1 in strips using latest VMM maxima
x.strips_latest_cuness2	number of points above 1 in strips with span > cuness_min_span using latest VMM maxima
x.strips_latest_density	% of strips in span with latest VMM maxima
x.strips_latest_integral	integral of strips with latest VMM maxima
x.strips_latest_integral_density	integral / valid for strip with latest VMM maxima
x.strips_latest_integral_norm	integral / valid_points for strip with latest VMM maxima
x.strips_latest_span	span of strips with latest VMM maxima
x.strips_latest_valid	number of strips with latest VMM maxima
x.strips_latest_valid_points	number of valid points in strips with latest VMM maxima
x.strips_later_average_c	strip average (no weights) using later VMM maxima
x.strips_later_center_c	strip center of gravity using later VMM maxima
x.strips_later_center_ortho_c	time bin-weighted strip center of gravity using later VMM maxima
x.strips_later_cuness	number of points above 1 in strips using later VMM maxima
x.strips_later_cuness2	number of points above 1 in strips with span > cuness_min_span using later VMM maxima
x.strips_later_density	% of strips in span with later VMM maxima
x.strips_later_integral	integral of strips with later VMM maxima
x.strips_later_integral_density	integral / valid for strip with later VMM maxima
x.strips_later_integral_norm	integral / valid_points for strip with later VMM maxima
x.strips_later_span	span of strips with later VMM maxima
x.strips_later_valid	number of strips with later VMM maxima
x.strips_later_valid_points	number of valid points in strips with later VMM maxima
x.strips_vmm_average_c	strip average (no weights) using VMM maxima
x.strips_vmm_center_c	strip center of gravity using VMM maxima
x.strips_vmm_center_ortho_c	time bin-weighted strip center of gravity using VMM maxima
x.strips_vmm_cuness	number of points above 1 in strips using VMM maxima
x.strips_vmm_cuness2	number of points above 1 in strips with span > cuness_min_span using VMM maxima
x.strips_vmm_density	% of strips in span with VMM maxima
x.strips_vmm_integral	integral of strips with VMM maxima
x.strips_vmm_integral_density	integral / valid for strip with VMM maxima
x.strips_vmm_integral_norm	integral / valid_points for strip with VMM maxima
x.strips_vmm_span	span of strips with VMM maxima
x.strips_vmm_valid	number of strips with VMM maxima
x.strips_vmm_valid_points	number of valid points in strips with VMM maxima
x.timebins_all_average_c	time bin average (no weights) using valid ADC values
x.timebins_all_center_c	time bin center of gravity using valid ADC values
x.timebins_all_center_ortho_c	strip-weighted time bin center of gravity using valid ADC values
x.timebins_all_cuness	number of points above 1 in time bins using valid ADC values
x.timebins_all_cuness2	number of points above 1 in time bins with span > cuness_min_span using valid ADC values
x.timebins_all_density	% of time bins in span with valid ADC values

x.timebins_all_integral	integral of time bins with valid ADC values
x.timebins_all_integral_density	integral / valid for time bin with valid ADC values
x.timebins_all_integral_norm	integral / valid_points for time bin with valid ADC values
x.timebins_all_span	span of time bins with valid ADC values
x.timebins_all_valid	number of time bins with valid ADC values
x.timebins_all_valid_points	number of valid points in time bins with valid ADC values
x.timebins_entry_c	Latest time bin of VMM maxima
x.timebins_vmm_average_c	time bin average (no weights) using VMM maxima
x.timebins_vmm_center_c	time bin center of gravity using VMM maxima
x.timebins_vmm_center_ortho_c	strip-weighted time bin center of gravity using VMM maxima
x.timebins_vmm_cuness	number of points above 1 in time bins using VMM maxima
x.timebins_vmm_cuness2	number of points above 1 in time bins with span > cuness_min_span using VMM maxima
x.timebins_vmm_density	% of time bins in span with VMM maxima
x.timebins_vmm_integral	integral of time bins with VMM maxima
x.timebins_vmm_integral_density	integral / valid for time bin with VMM maxima
x.timebins_vmm_integral_norm	integral / valid_points for time bin with VMM maxima
x.timebins_vmm_span	span of time bins with VMM maxima
x.timebins_vmm_valid	number of time bins with VMM maxima
x.timebins_vmm_valid_points	number of valid points in time bins with VMM maxima

You may note some repetitive patterns in the naming and descriptions. There are fundamentally only about a dozen metrics:

Valid	number of (strips) with valid data
valid_points	number of valid readout events with ADC > 0
span	span of (strip) values, i.e. max – min + 1
density	fraction of span that has valid data, reported as percent, i.e. valid/span*100
integral	sum of all ADC values in subset
integral_density	normalized integral, i.e. integral / valid
integral_norm	alternative normalized, i.e. integral / valid_points
cuness	cness or uness, depending on domain (see below)
cuness2	variation of above (see below)
average_c	mean average (strip) number of all points
center_c	centre of gravity of all points weighted by ADC
center_ortho_c	centre of all points weighted by (time)

These are calculated for various subsets of the readout events available. These subsets are selected from the total points by selecting a domain and applying certain threshold criteria. The first component of the metric naming convention reflects this selection of domain and subset. If a parameter plays a part in the construction of some metric, said [parameter] will be in square brackets.

The following are subsets with readout event strip number as domain:
strip_all – all readout events available in the APV25 dataset
strip_vmm – emulated VMM readout events: in each strip, only events with ADC maxima in contiguous regions where ADC values are above [threshold] for some minimum

[over_threshold] number of time bins. Furthermore, to emulate digitizer dead-time, any maximum less than [min_peak_separation] time bins after the first will be rejected.
strip_later – a subset of strip_vmm, with points only from the latest [best_max_bincount] time bins, no more than [best_max_binspan] time bins before the latest time bin.
strip_latest – an even smaller subset of strip_vmm, using only the readout events in the very latest time bin. This is a degenerate case of the above strip_later, with both parameters set to 1.

The following are subsets with readout event time bin as domain:
timebins_all – all readout events available in the APV25 dataset
timebins_vmm – nothing to do with VMM as such, only that it applies the same filtering but with strips and time bins interchanged, i.e. scanning each time bin across all strips, looking for when ADC goes above threshold from one strip to the next. The purpose of was to find "u-ness". This perspective will be impossible with actual VMM data.

For each of the above subsets, the following elementary metrics are calculated. For clarity of illustration, it is here assumed that the chosen domain is (strips) and the codomain is (timebins), but as per chosen subset, these may be reversed.

valid_points – number of valid readout events: (strip, time bin) points with non-zero ADC value. For the superset of all readout events, this is perhaps some indication of the spread of the track. For the vmm subset, this can be used as some estimate of data volumes that could be expected in the final event pipeline.

valid – number of (strips) with valid readout events. This is also some indication of diffuseness of the track, but only in the one dimension (the domain). Intermediate step for normalization of integral and for the density/sparseness of event in domain space.

span – difference between maximum and minimum strip indices. This is the spread of the particle trace in one dimension. This is an uncertainty of event location. The "strictness" of this uncertainty depends on what subset of readout events is chosen. Intermediate step in determination of event density/sparseness. Looking at this metric for the superset will show a lot of broad spans as a result of readout artefacts. The more selective "vmm" data is thus more indicative of the true span of events.

density – percentage of valid (strips) in span, i.e. valid / span * 100.0; In another way, the lower the number, the more "empty" strips appear to be interspersed in the middle of the track, indicating discontinuity in the particle track, multiple tracks in the same cluster, or problems with the readout.

integral – sum of ADC values of all readout events, or total charge deposited in the track. This metric can be histogrammed for a number of events to generate a spectrum of the conversion electrons. The usefulness of this, however, is limited by the energy resolution of the detector system. Comparing the integral of all points and that of emulated VMM3 points, for example, shows correlation, so the smaller sample may be a good indication of total energy deposition. However, total energy deposited does not appear to be a good filter for event validity.

integral_norm – same as above, divided by valid_points. Results in a smoother spectrum.

integral_density – somewhat of a misnomer, it is also just a normalization, but this time divided by valid (number of valid domain values).

average_c – arithmetic mean of strip number. This is not of much use when considering all the points, but makes sense for a small cluster of (latest) points. The _c suffix is added to indicate that this is one of the contender entry coordinates.

center_c – center of gravity, i.e. average strip number, weighted by their ADC values. This appears to be better than just the simple unweighted average. Even if we end up choosing the less intense branch in a U-type event, we would still want the centre of gravity of the end of one of the branches.

center_ortho_c – also an average of strip number, but in this case weighted by time bin number (the orthogonal coordinate), giving more weight to later events. This has not turned out to be useful.

cuness – meant to be a measure of either u-ness or c-ness depending on chosen domain. In the case of domain as strips, this counts the number of points above one found in any of the strips. For each strip, if it has more than one readout event, we increment cuness by number of points minus one. Thus, strips with 3 points increase cuness by 2, strips with 1 point do not affect it, and so forth. This does not strictly identify events that look like the letter C but rather more generally those that exhibit some lateral "wrap-around" or change in direction in the path of the conversion electron. In both domains, it helps identify events that exhibit such features to a high degree. By itself the metric does not appear to be very useful for selecting reliable events. As long as there is little ambiguity about the particle's position closer to the conversion plane (i.e. later time bins) the curved path of the particle in other parts of the detector space do not affect the uncertainty too much. It is this observation that prompted the creation of the later/better subset of points.

cuness2 – a variation of the above, with the added criterion that the distance between maxima on one strip be greater than [cuness_min_span]. This ensures that closely-spaced maxima do not inflate the value, instead the wrap-around having to be rather broad for it to count. This is useful in particular for U-ness more than C-ness, because APV data appears to come in vertical streaks, sometimes rather tightly packed. We want to focus on the "broader" features. This metric has been useful in quantifying U-like events, but it will likely be unavailable in the final data gathered with the VMM3 chip, due to digitizer dead-time. The issue is sidestepped, however, by considering uncertainty associated with “later” readout events, as discussed in section XXX.

not_gamma – an attempt was to discriminate gamma rays from neutrons. In datasets from experiments that contained only a gamma source, most electron tracks appear to be broad but compact. Usefulness of the current implementation is inconclusive. The issue may have to be re-examined in later experiments.

15 References

[1] S. Peggs et al., *ESS Technical Design Report*, ESS-2013-0001 (2013).

[2] D. Pfeiffer et al., *First measurements with new high-resolution gadolinium-GEM neutron detectors*, J. Inst., **11**, (2016).

[3] GEANT4 website: http://geant4.web.cern.ch/geant4/

[4] Garfield++ website: http://garfieldpp.web.cern.ch/garfieldpp/

[5] Interfacing Geant4 and Garfield++,
https://garfieldpp.web.cern.ch/garfieldpp/examples/geant4-interface/
[5] M.J. French et al., *Design and results from the APV25, a deep sub-micron CMOS front-end chip for the CMS tracker,* Nucl. Instrum. Meth. A **466** (2001) 359.
[6] G. Lakovidis et al., *VMM3 Status Update*, https://indico.cern.ch/event/607147/contributions/2482014/attachments/1415234/2166424/2017_02_20_RD51MiniWeek_VMM3_Update_Iakovidis.pdf
[7] D.Pfeiffer, P. Thurnier, et al. *BrightnESS Deliverable Report: D4.3 − Natural and enriched Gadolinium converters design*
[8] A. Mukai, T. Richter, et al. *BrightnESS Deliverable Report: D5.1 – Design report data aggregator software*
[9] F. Piscitelli, F. Messi, et al. *BrightnESS Deliverable Report: D4.6 – Detector Characterization*
[10] K. Hu, H. Lu, et al., *A DAQ prototype for the ATLAS small-strip Thin Gap Chamber Phase-I trigger upgrade*, IEEE-NPSS Real Time Conference, (2016)
[11] G. Bauer, V. Boyer, et al., *CMS DAQ Event Builder Based on Gigabit Ethernet*, IEEE Trans. On Nucl. Sci., **55** (2008)
[12] M. Anastasopolus, R. Bebb et al., *Multi-Grid Detector for Neutron Spectroscopy: Results Obtained on Time-of-Flight Spectrometer CNCS*, JINST **12** (2017)