SampaSRS data reconstruction tool

RD51 collaboration meeting
06/12/2023

Geovane G. A. de Souza[1], Thiago B. Saramela[1], Hugo Natal da Luz[2], Marco Bregant[1].

1-High Energy Physics and Instrumentation Center @ IF-USP (Brazil)
2-Institute of Experimental and Applied Physics (ÚTEF) @ CTU in Prague (Czech Rep.)

Summary:

Introduction

- SAMPA overview
- The first test setup
- The first version of the software

Non-zero suppression acquisition

- Non-zero suppression
- The clustering algorithm
- First results without common-mode correction

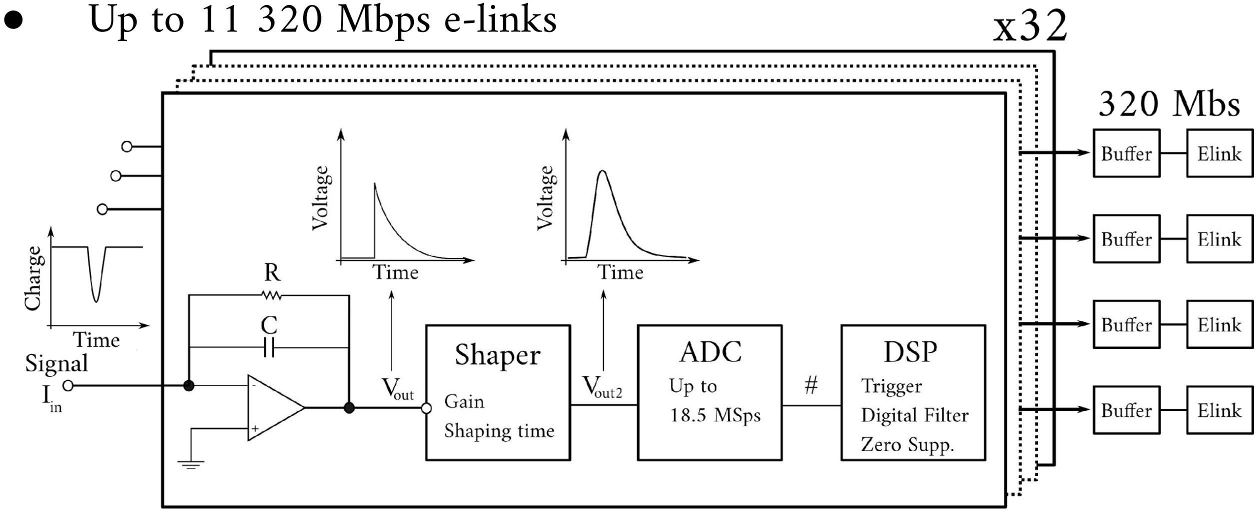
The common-mode correction algorithm

- Noise reduction
- Results for energy resolution
- Results for position resolution

The operation on Zero suppression

SAMPA overview

- TSMC CMOS 130nm, 1.25V technology.
- 32 Channels, Front-end + ADC + DSP.
- Positive and negative polarities with 2 analog front-end modes:
  - 20 or 30 mV/fC with 160 ns shaping time.(Sensor Cap: 12 - 25 pF)
  - 4 mV/fC with 300 ns shaping time. (Sensor Cap: 40 - 80 pF)
- ADC: 10-bit resolution, up to 18.5 MSPS.
- Up to 11 320 Mbps e-links



SAMPA Block Diagram

The SAMPA chip is an ASIC (Application Specific Integrated Circuits) developed in collaboration with the *Laboratório de Sistemas Integráveis* (LSI) at POLI, to be used in the ALICE TPC and Muon Chamber during the Run3.

We are developing a software for control, acquisition, decoding of data, and event reconstruction.

Version 1.0

SampaSRS

How to build

The code has the following build dependencies

- C++17 compiler
- Git
- CMake (3.14 or later)
- ROOT (6.26 or later)
- libpcap
- SDL2 (for sampa_gui)

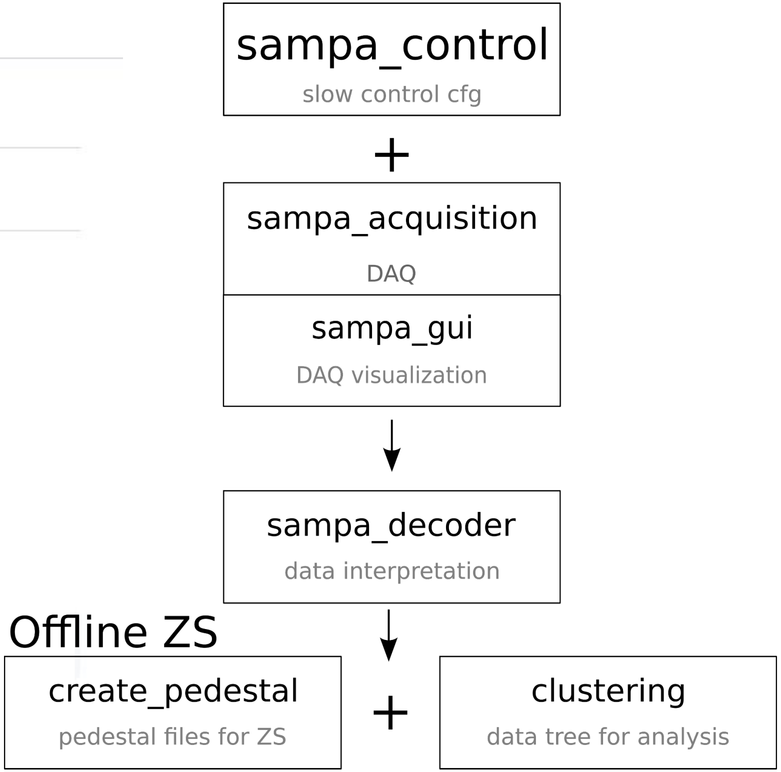
Ubuntu dependencies

```
sudo apt install build-essential git cmake libpcap-dev libsdl2-dev
```

Windows dependencies

Download the precompiled binaries from WinPcap.

To build you also need to pass the location of the extracted binaries to CMake:



https://github.com/SampaSRS/SampaSRS

We are developing a software for control, acquisition, decoding of data, and event reconstruction.

Version 2.0

SampaSRS

How to build

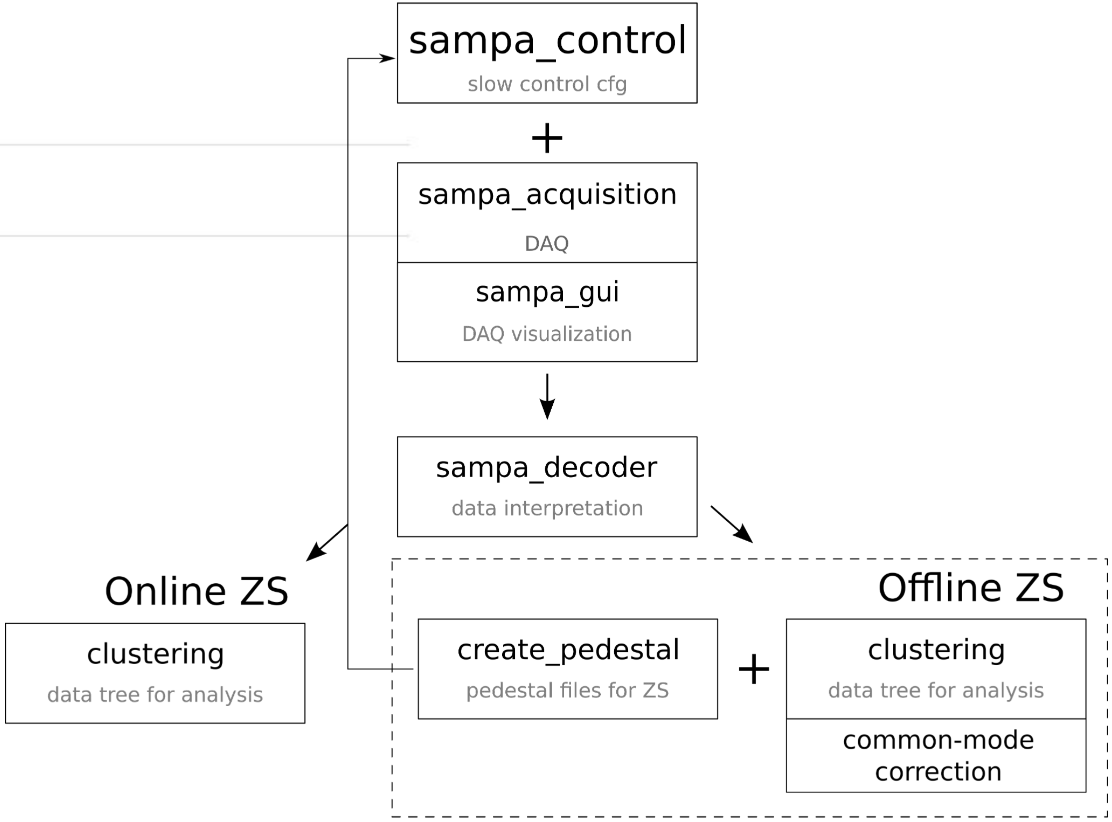
The code has the following build dependencies

- C++17 compiler
- Git
- CMake (3.14 or later)
- ROOT (6.26 or later)
- libpcap
- Latest compatible firmware version - 14

Ubuntu dependencies

```
sudo apt install build-essential git cmake libpcap-dev
```

Windows dependencies



https://github.com/SampaSRS/SampaSRS

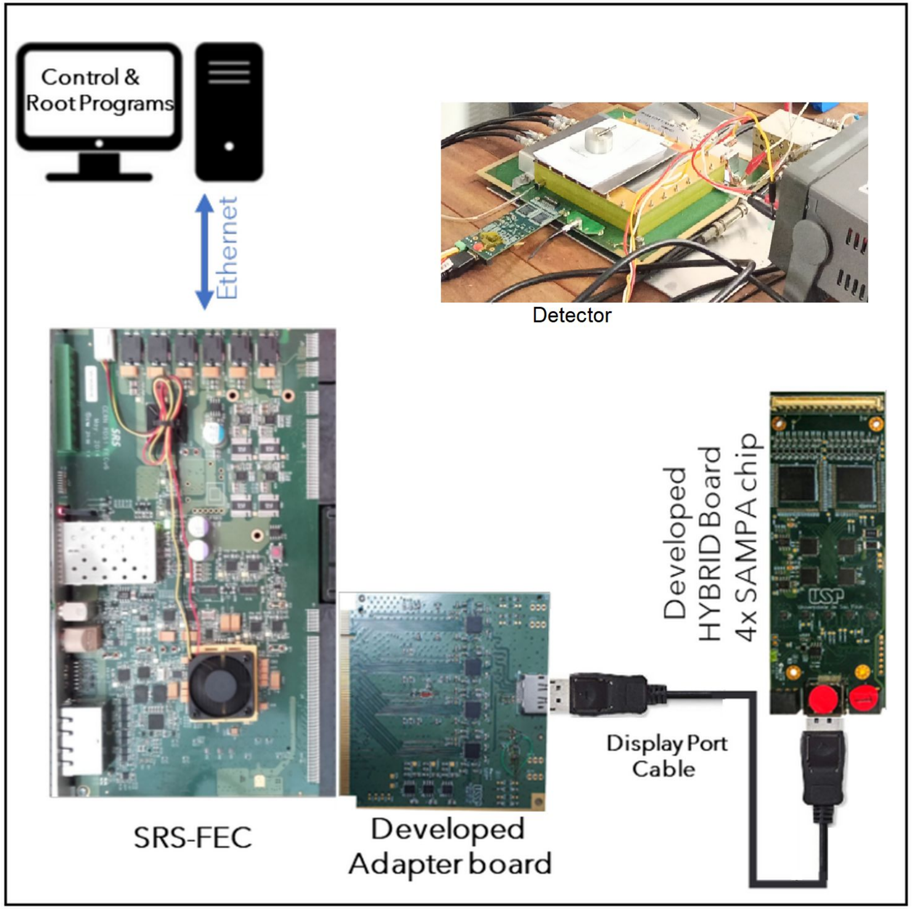
SampaSRS software

https://github.com/SampaSRS/SampaSRS

Acquisition details:
1. Total number of events
2. Event rate
3. Ratio of valid events
4. Network speed/usage
5. Write speed

SampaGUI window (left to right):
1. Energy spectrum
2. Channel with higher ADC value
3. Waveform of the channel with higher ADC value

The complete setup



Hybrid board overview:

Each hybrid provides 128 channels
The physical dimensions (width of the hybrid) are compatible with the readout plane
(10 x 10 cm²) developed by the RD51 collaboration

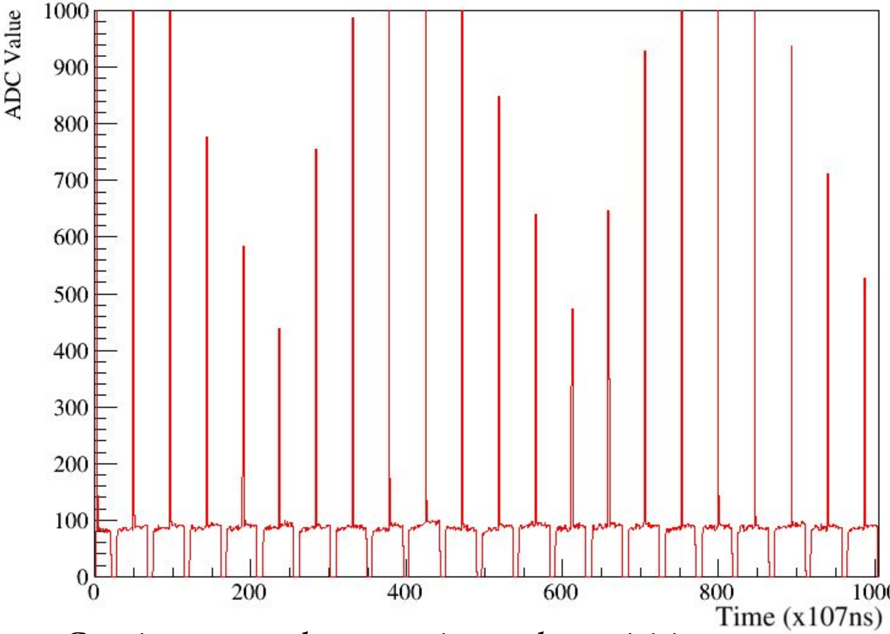
Adapter board overview:

Each SAMPA chip is connected to one high speed serializer
A single Display-Port cable is used to connect the hybrid and the adapter board
The adapter board has four deserializers and a PCIx16 standard to connect a Front-End Card (FEC).

FEC:
Ethernet communication limit to 1Gbit/s

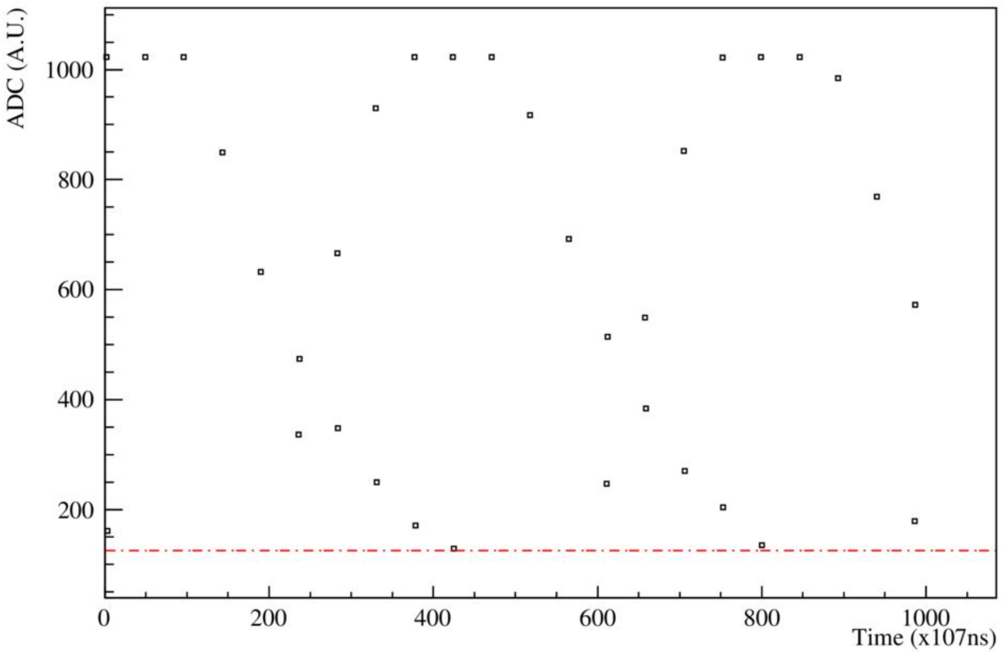
Acquisition rate: Depends on the type of operation

Non-zero suppression



Continuous readout or triggered acquisition
Capability to recover the raw event → full information
Apply common-mode correction → reduce noise

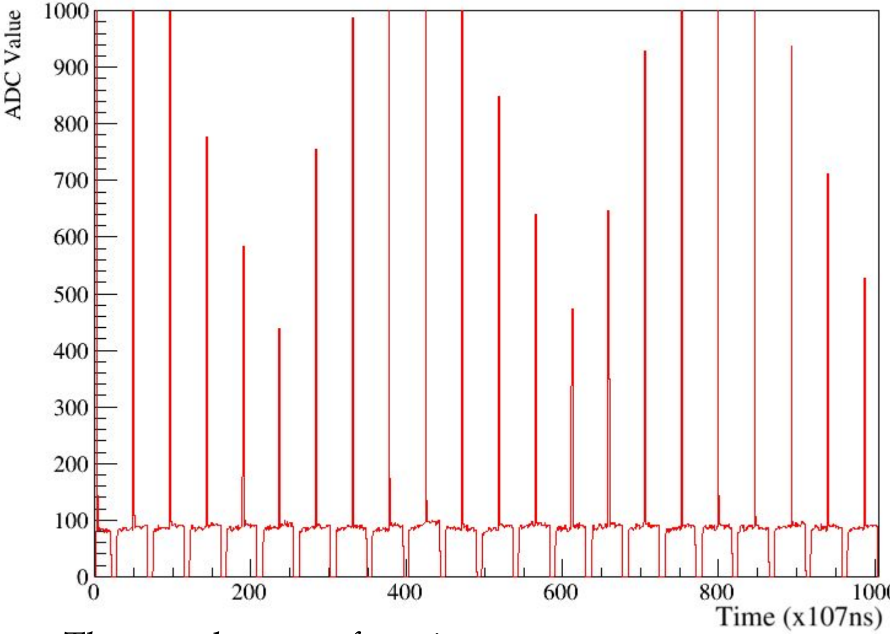
Zero suppression



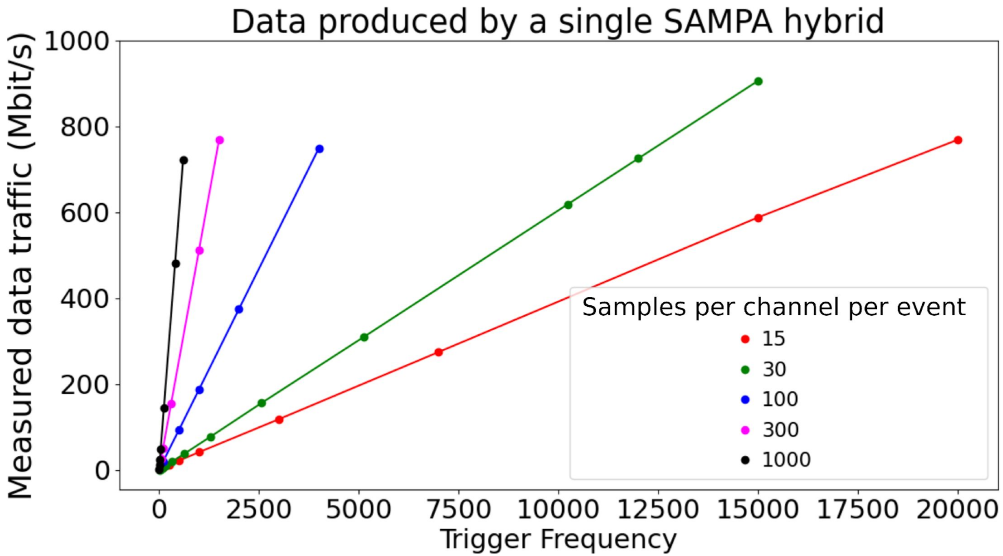
The cut is made by SAMPA
Operate at higher counting rates
Less disk space required

Acquisition rate: Depends on the type of operation

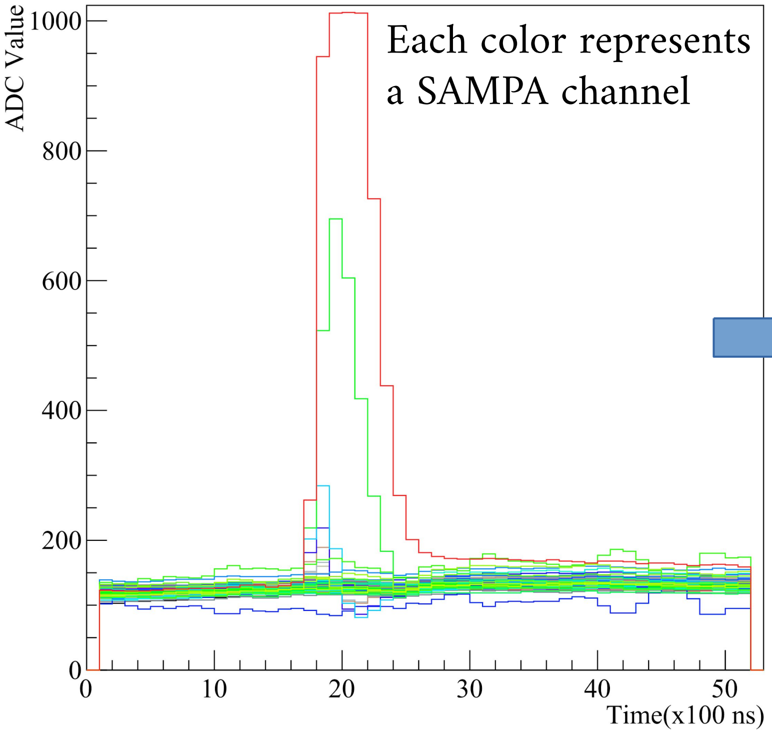
Non-zero suppression



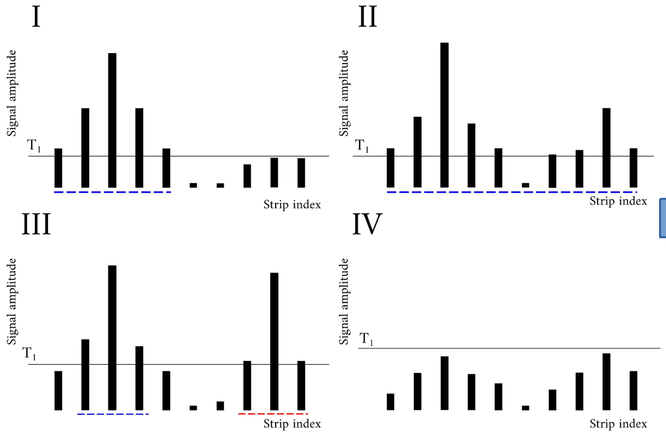
The complete waveform is sent
Capability to recover the raw event → full information
Apply common-mode correction → reduce noise



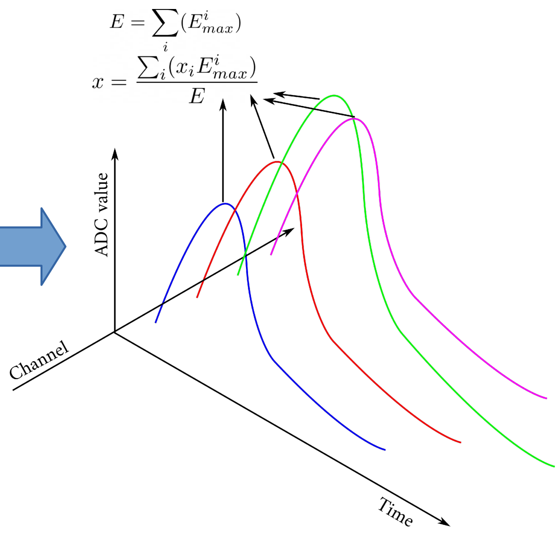
We apply a reconstruction algorithm to data: Offline ZS →Time cluster→ position cluster → CM.



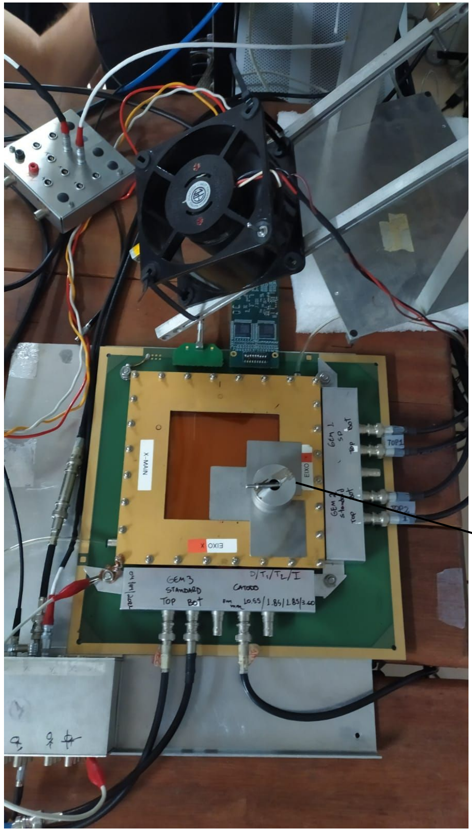
Clusters



Center of mass



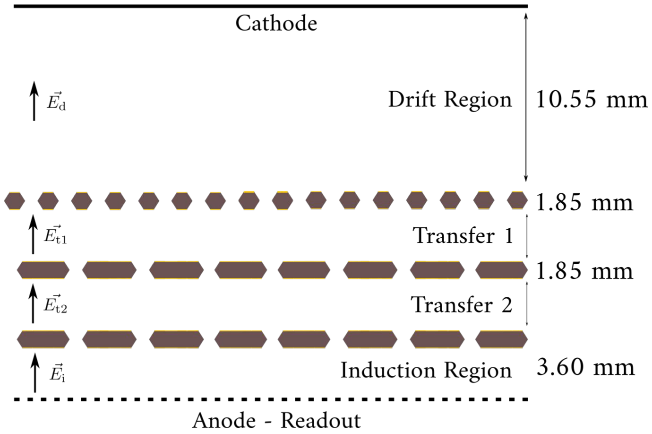
We have set an X-ray position sensitive detector to test the integration with SAMPA and the SRS

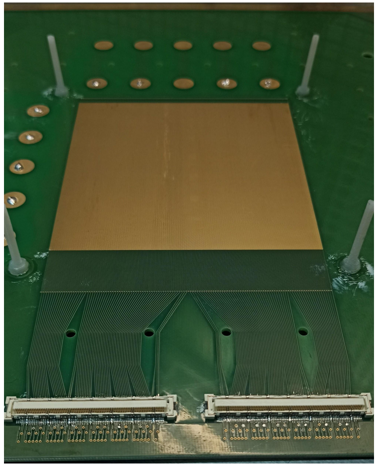


Triple-GEM

$Ar/CO_2(70/30)$

^{55}Fe source



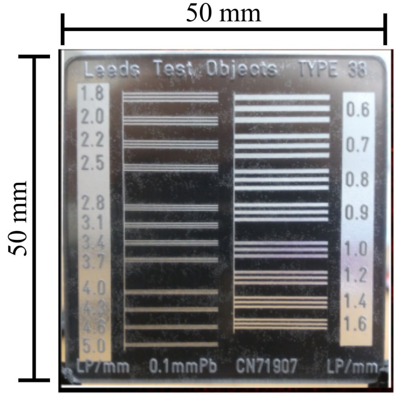


1D strip readout (0.4 mm pitch)

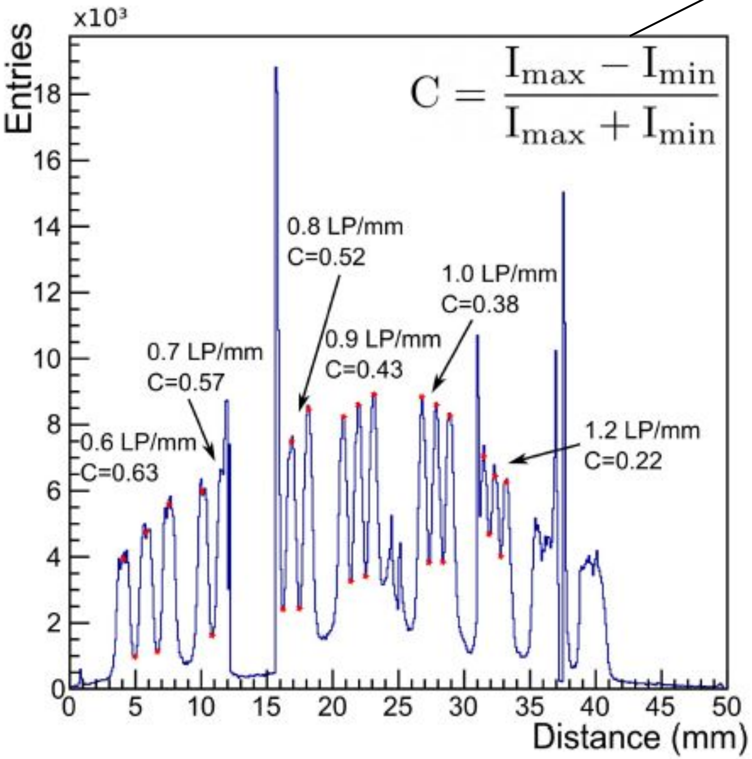
Designed by T. Abelha HEPIC@IFUSP

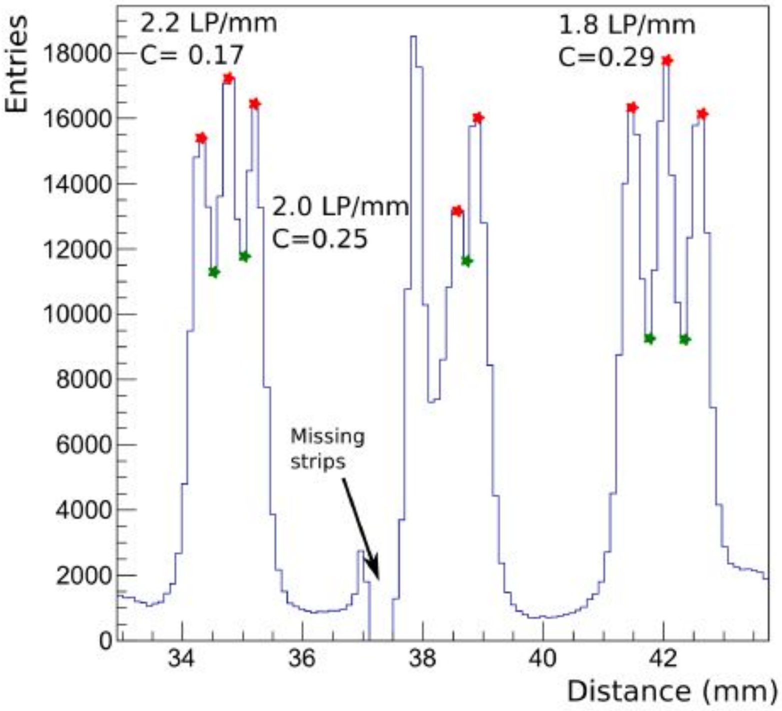
Results without Using a lead mask + X-ray tube we can calculate the position resolution of the detector

Michelson contrast

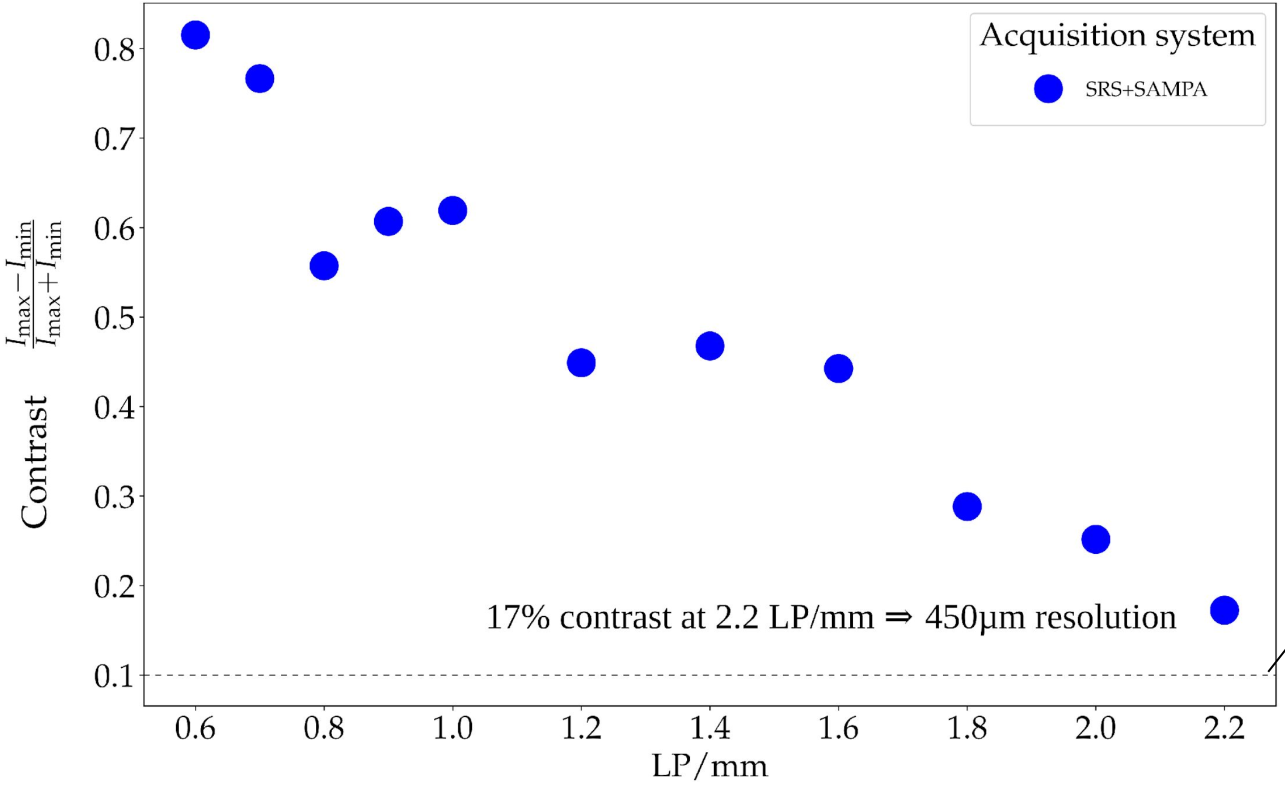


Resolution pattern *Type* 38



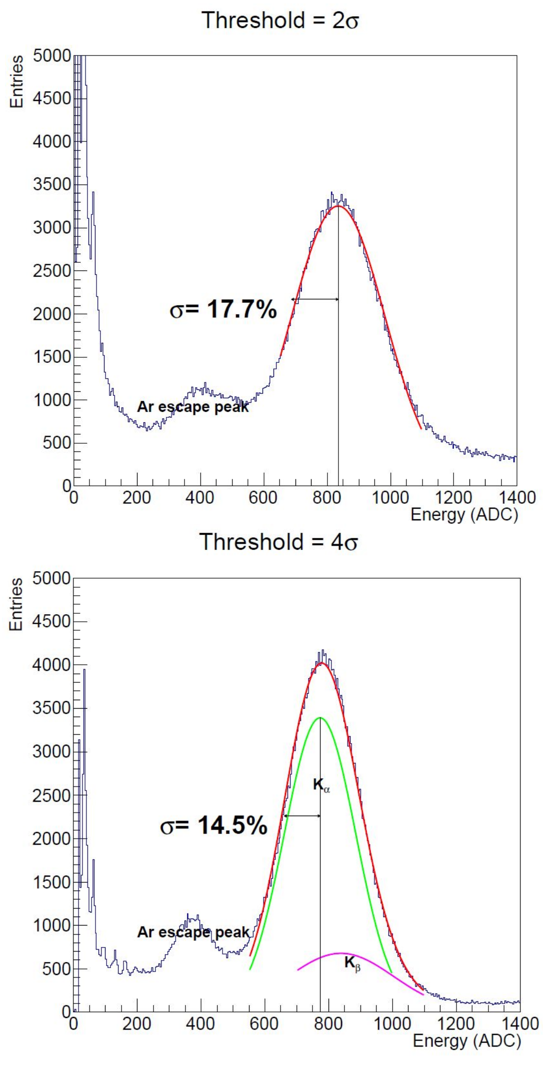


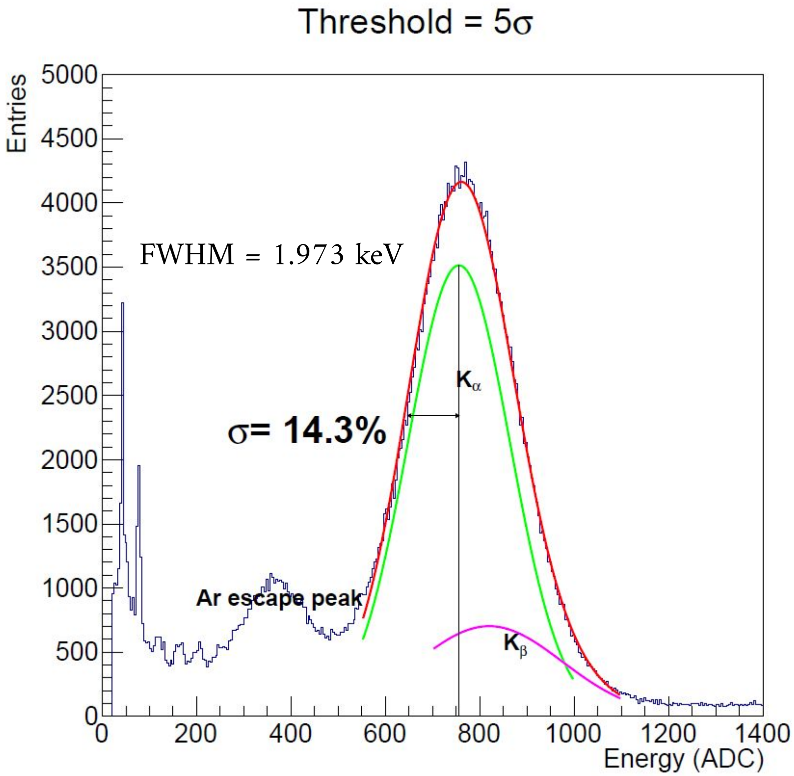
The position resolution is normally given by the contrast at 10%



For this run, better than 450μm

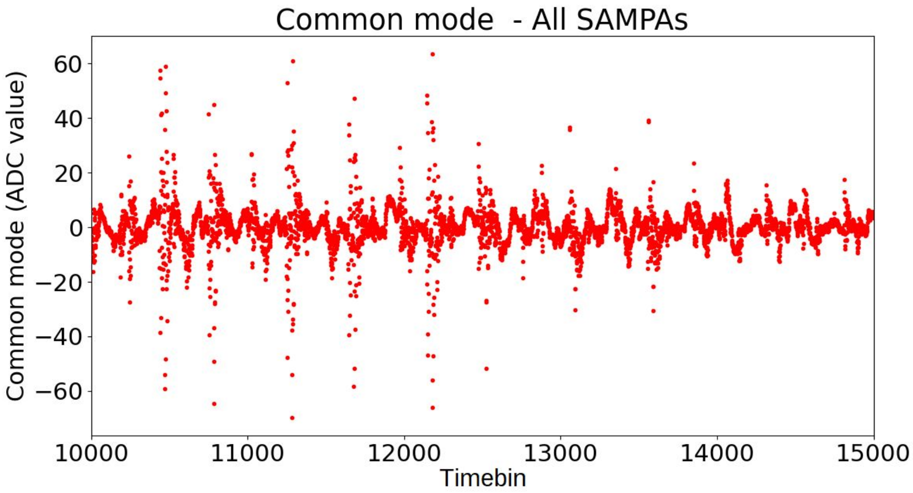
Energy spectrum for different choices of ZS threshold

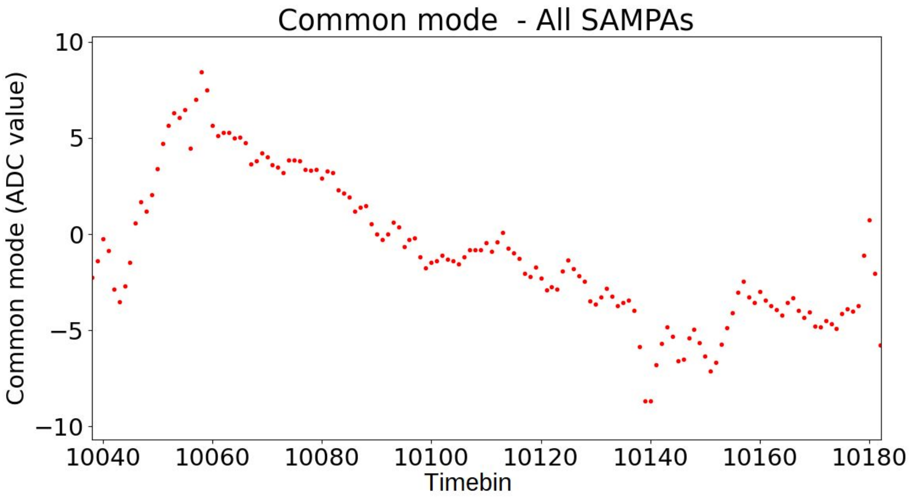




Common-mode correction

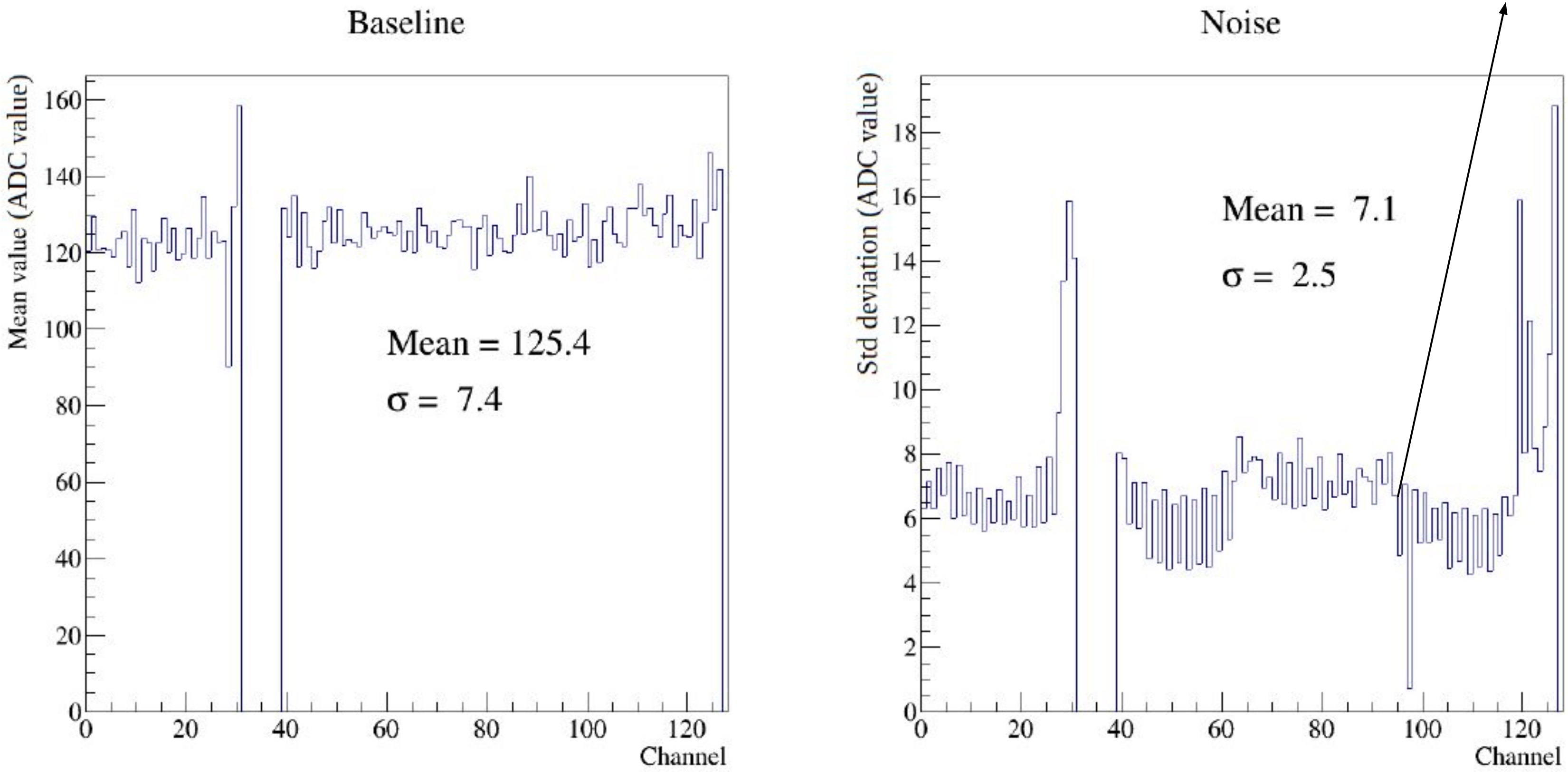
$$C(t) = \frac{\sum_{i=0}^{N} B_i(t) - \bar{B}(t)}{N}$$





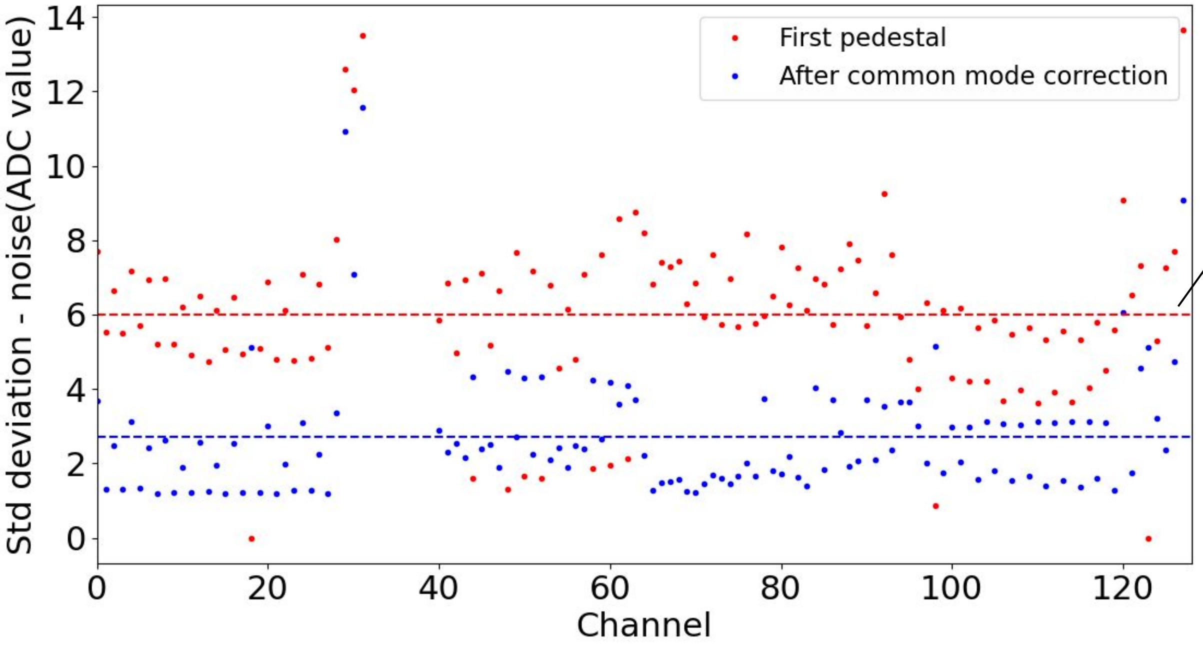
Baseline and noise before common-mode correction

≈5000 e-



Grounding issues in our lab

Common-mode correction

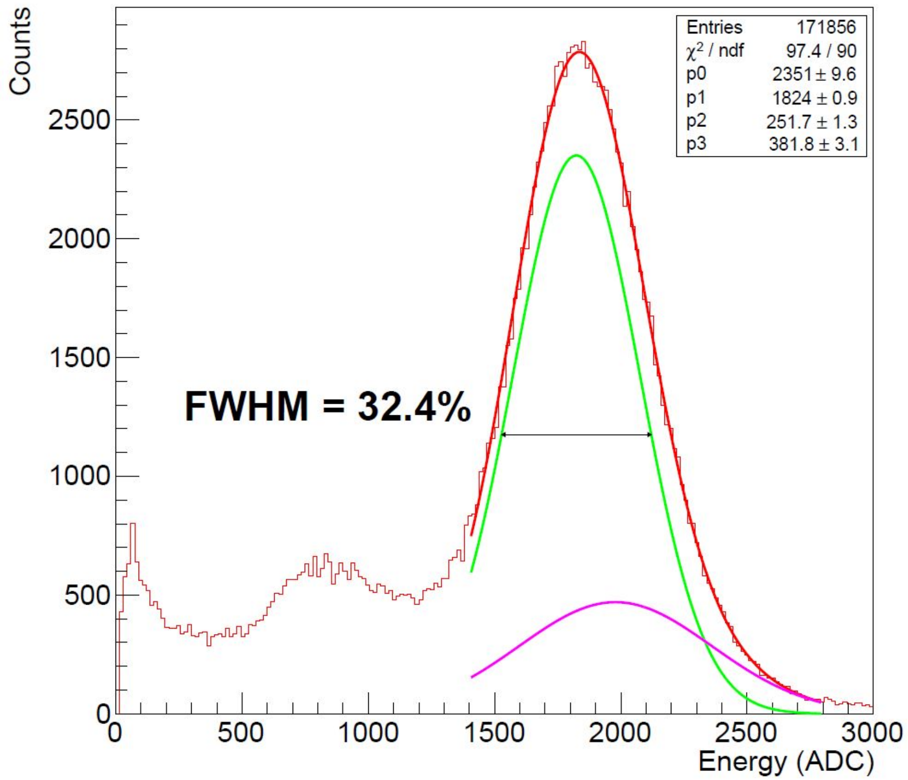


≈5000 e-

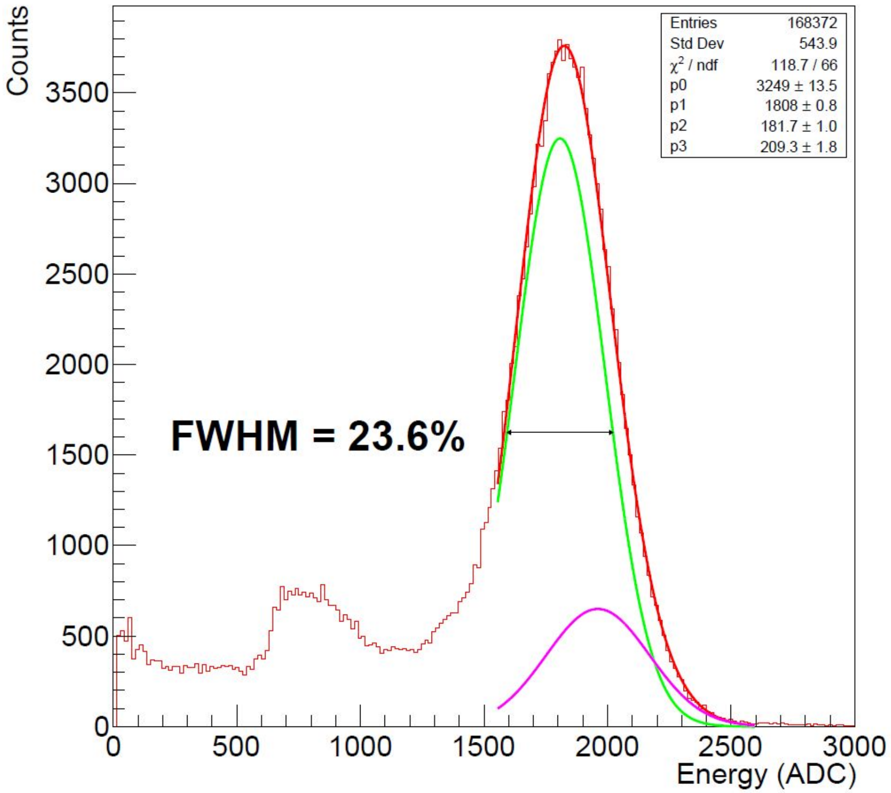
Reduction to half of the noise (works for offline ZS runs.)

Common-mode correction (Energy spectrum ^{55}Fe)

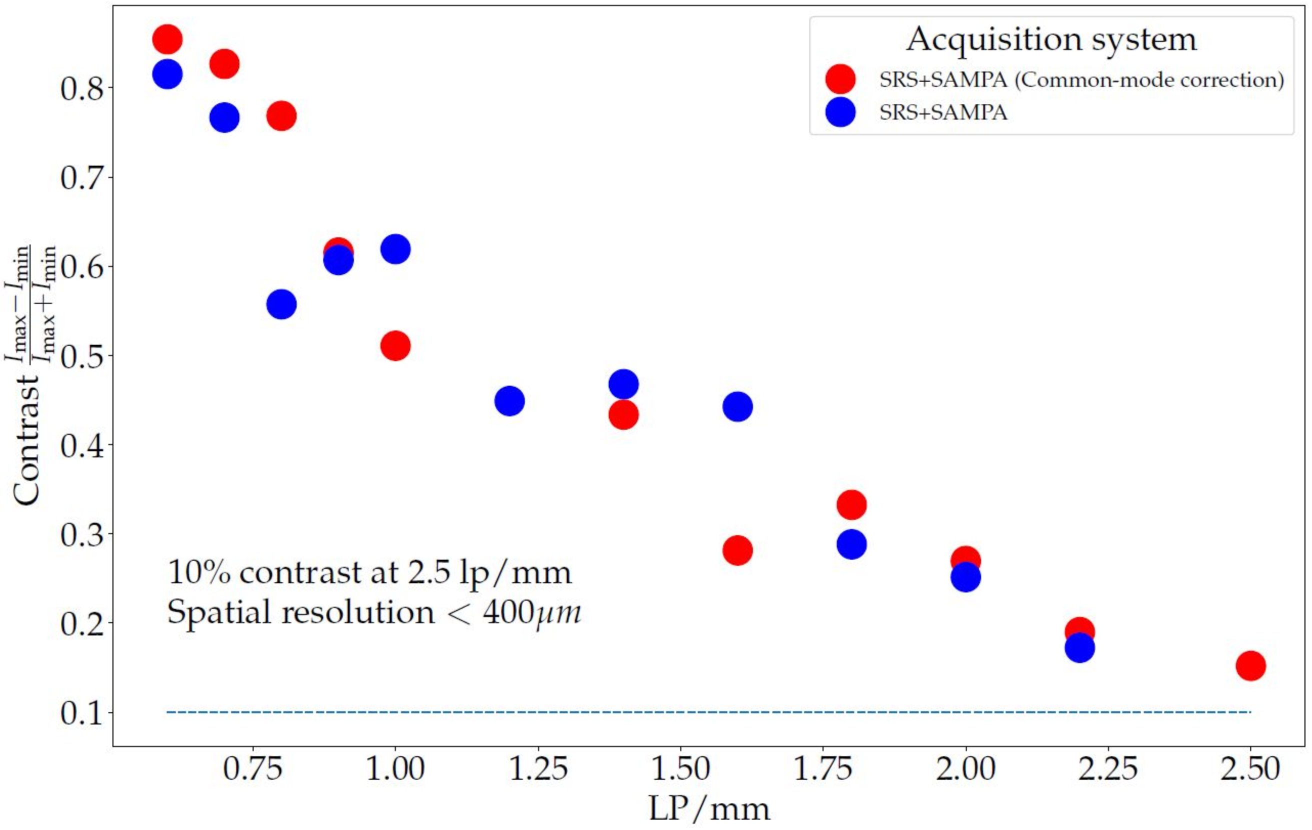
Before



After

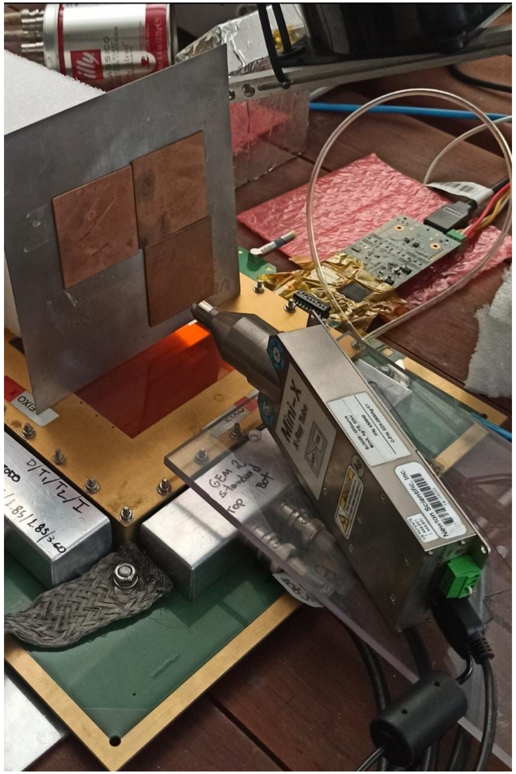


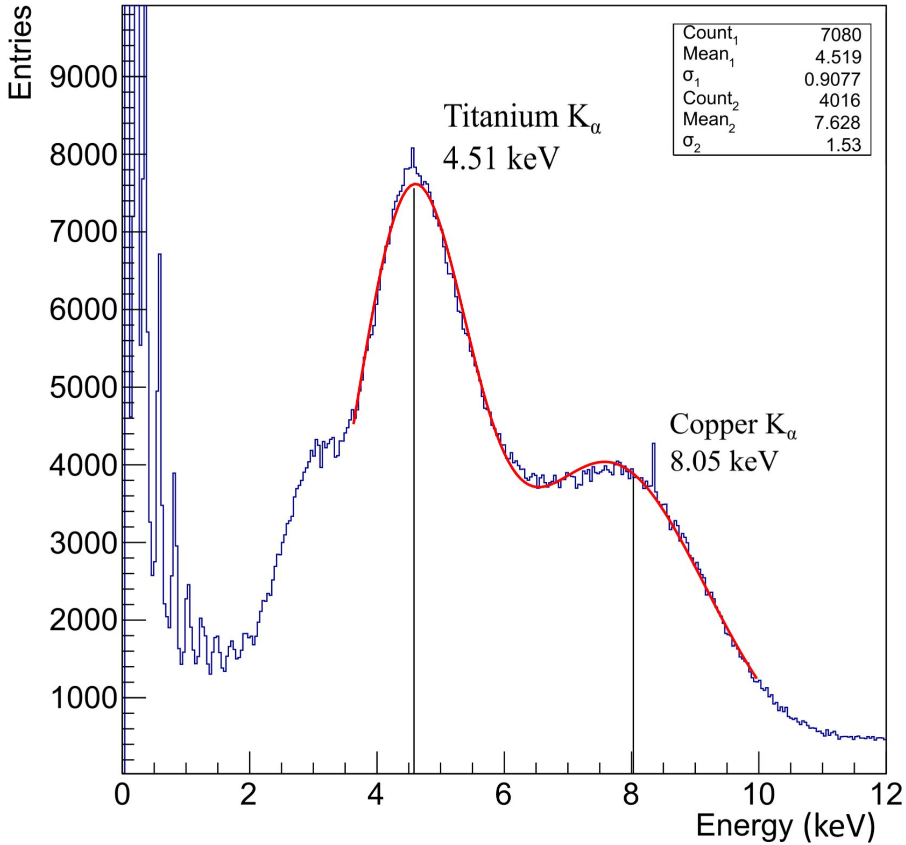
The position resolution is normally given by the contrast at 10%



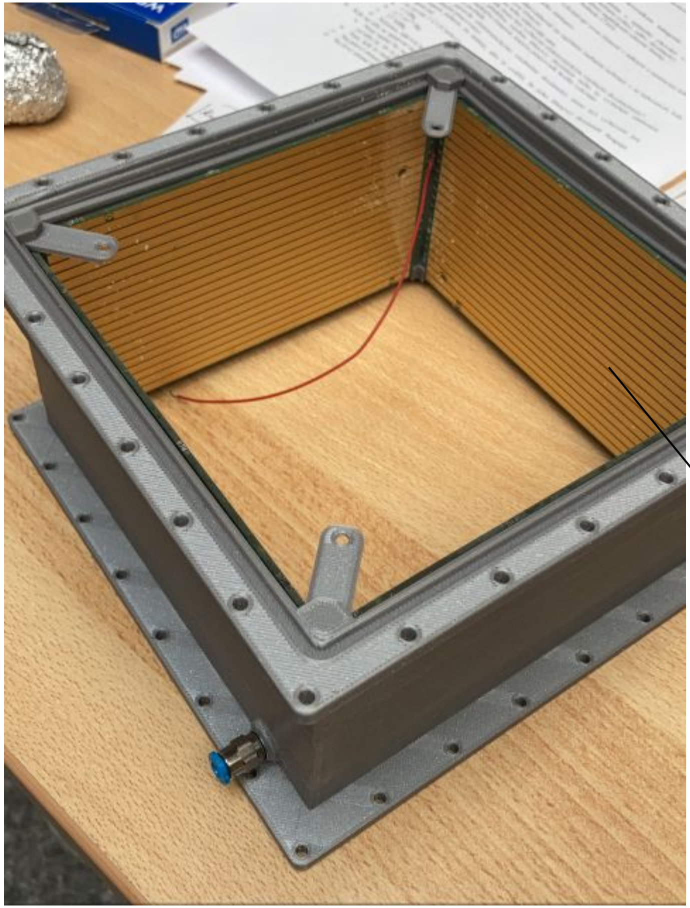
We can see additional contrast points (unfortunately one was on a dead region of the hybrid - missing SAMPA chip)

X-ray fluorescence using SAMPA and SRS

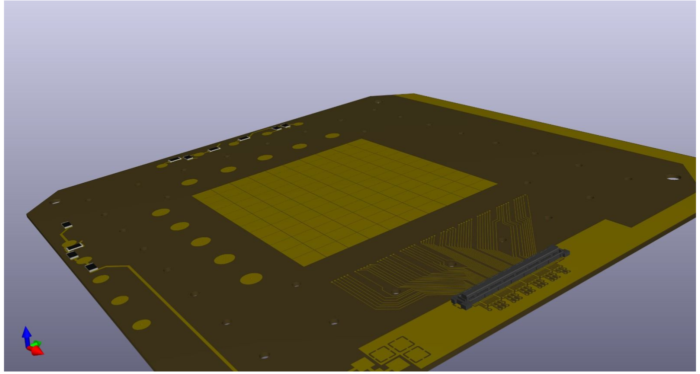




The Time Projection Chamber prototype

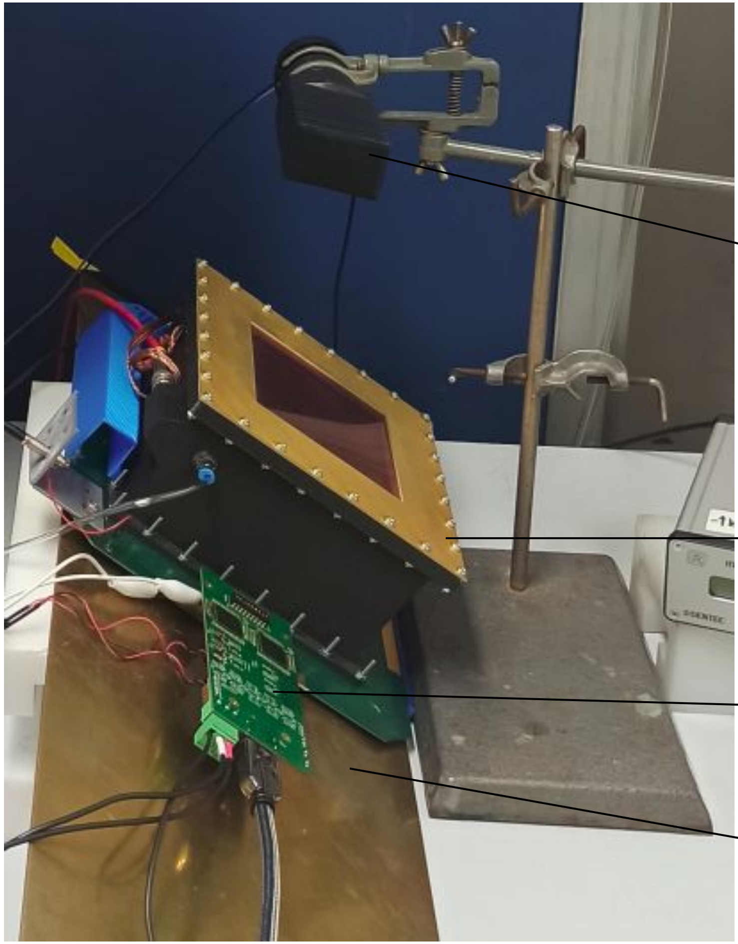


- 80 mm drift region Ar/CO_2(70/30)
-
- 3D printed frame- PLA
- Field cage made of PCB strips and SMD resistors



- Triple-GEM and pad readout (10x10 cm²)
- 10 x 12 pads

Detector tilted to detect longer tracks (event rate ≈ 1Hz)



Telescope setup (coincidence trigger for 3 detectors)

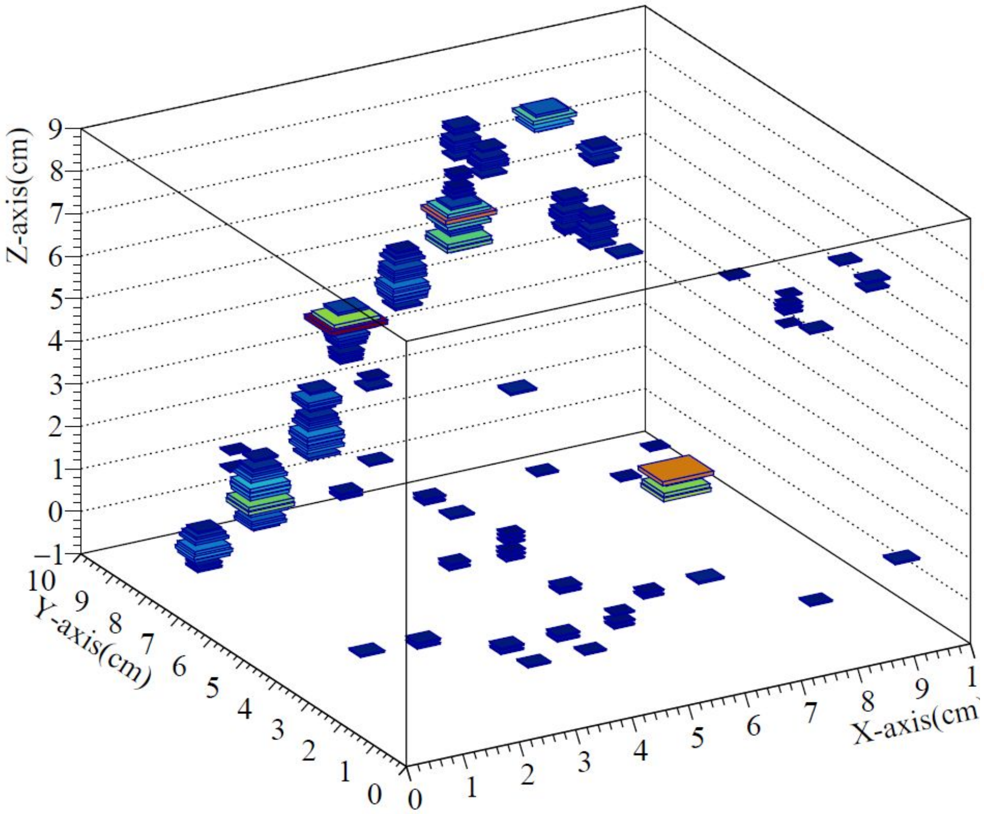
1° Plastic scintillator+PMT

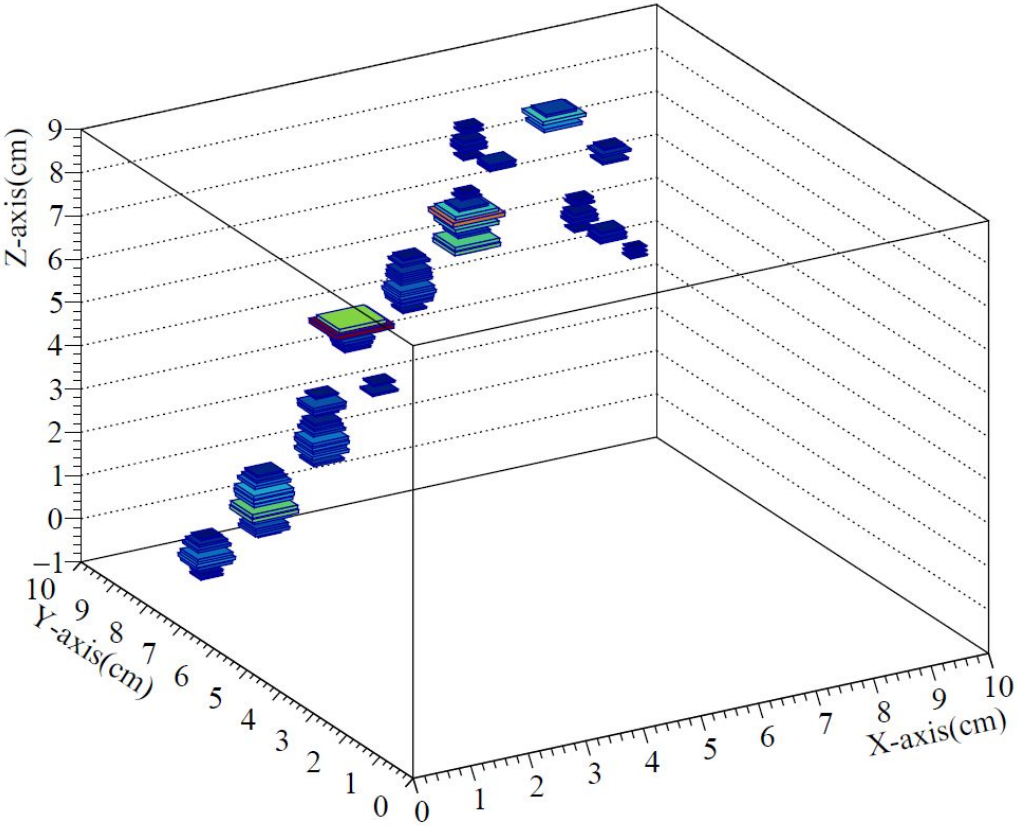
TPC

SAMPA hybrid

2° Plastic scintillator+PMT (hidden)

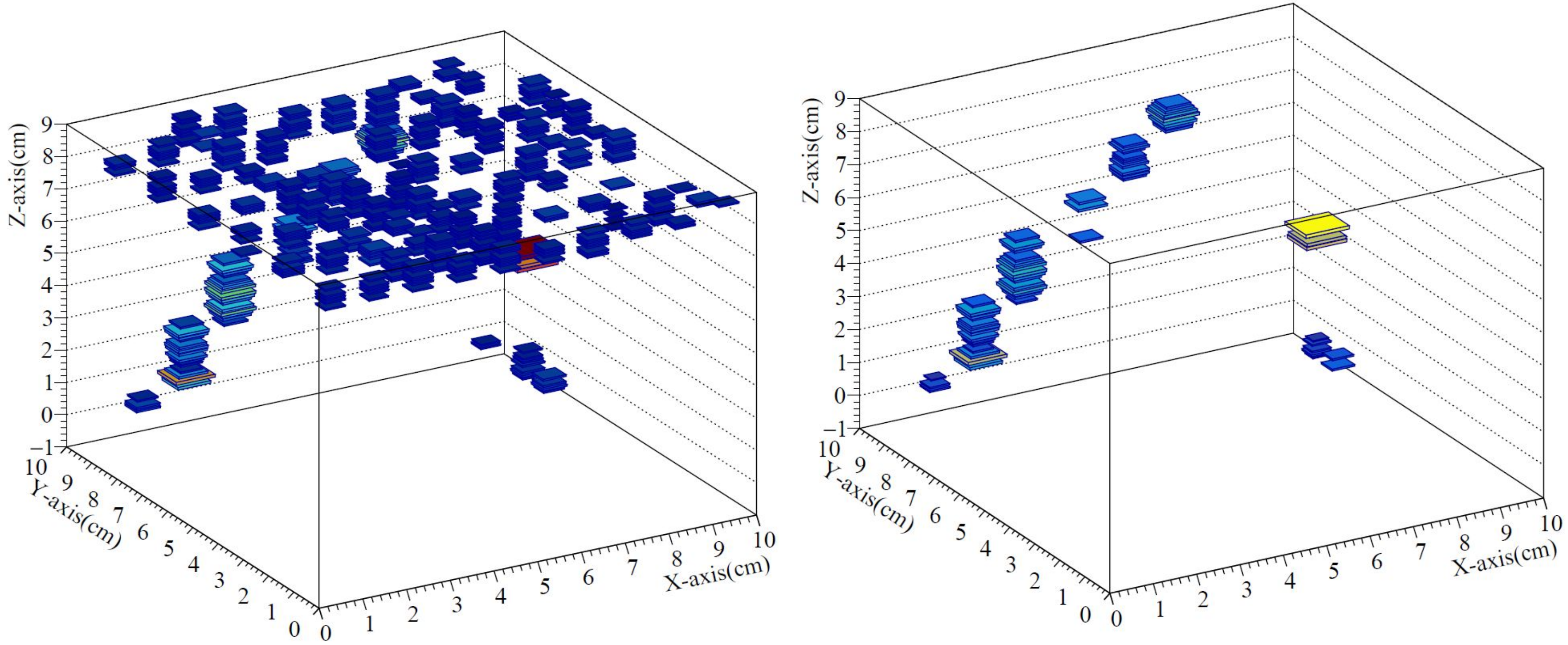
Example 1:





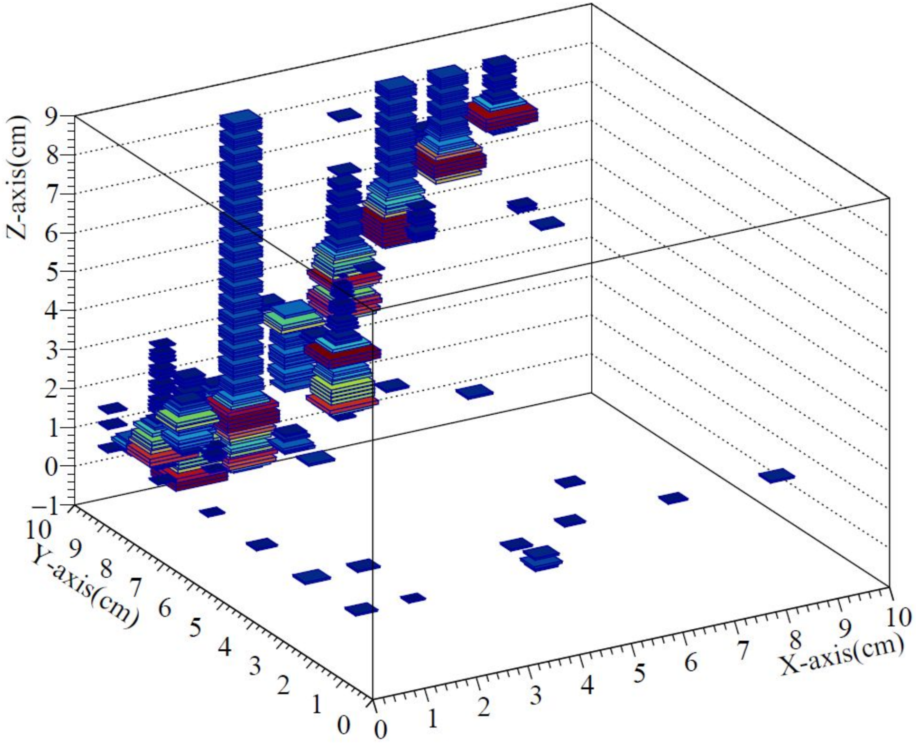
Secondary events/noise that eventually cross the threshold can be removed by applying a algorithm to remove non-neighboring signals.

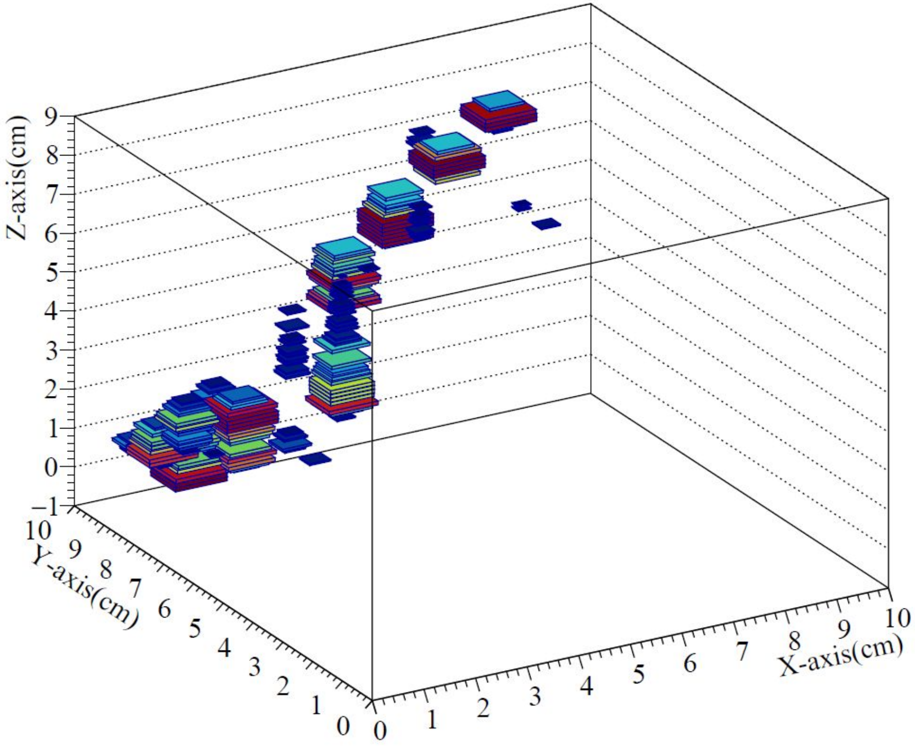
Example 2:



Artifacts yet somewhere unknown in the electronics (maybe a ground/baseline shift?) generated a small amplitude signal that affects many channels at the same time. They can be removed during data processing.

Example 3:





Whenever the charge saturates a channel, the baseline increases its mean. In these cases, the zero suppression cut is changed to 20σ.

Conclusions:

- The integration between SAMPA and the SRS was a success and we are capable to provide the tools (hardware/software) for other groups.

- We were able to achieve up to 20 kHz acquisition rate in non-zero suppression mode and we calculate that we can work up to 2 MHz in zero suppression mode (we will start testing and working on the reconstruction with this feature)

Prospects:

- Work with more than 1 hybrid (for 2D images)

- Think on ways to apply common-mode for zero suppressed data

Thank you!

Contact: geovane.souza@usp.br