Improvements to digitization

F.Petrucci

Notes on how to improve in digitization:
- Sensor noise
- Gain & fluctuations
- Diffusion

(inputs needed to finalize the code)

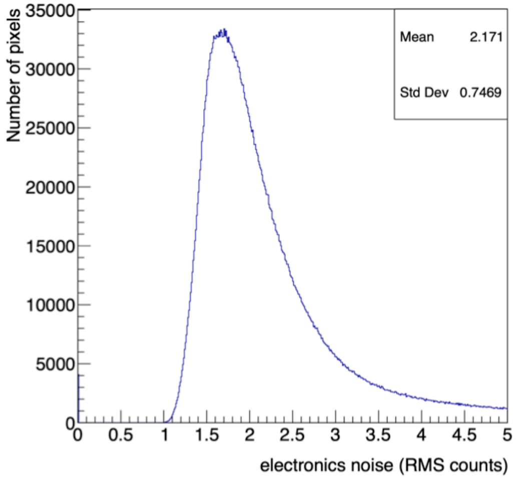
30/06/20

Sensor noise

- Current implementation: Add to each image pixel a random value from a Gaussian distribution with mean=99 photons and σ=2 photons.

BUT:

- data shows a pixel noise RMS distribution with non Gaussian tails well above the mean value of 2.



"Easy fix":

- Assign each pixel a different Gaussian noise RMS:
  1. sampling the measured distribution or extracting from a pdf fit to the data;
  2. picking for each pixel its own RMS from a 2D map (obtained from data) and extract accordingly.

Option 2 would consider any "structure" in noise distribution (for example, is the noise RMS higher in specific regions of the sensor?) but it is relevant only when simulated events are distributed on the whole surface.

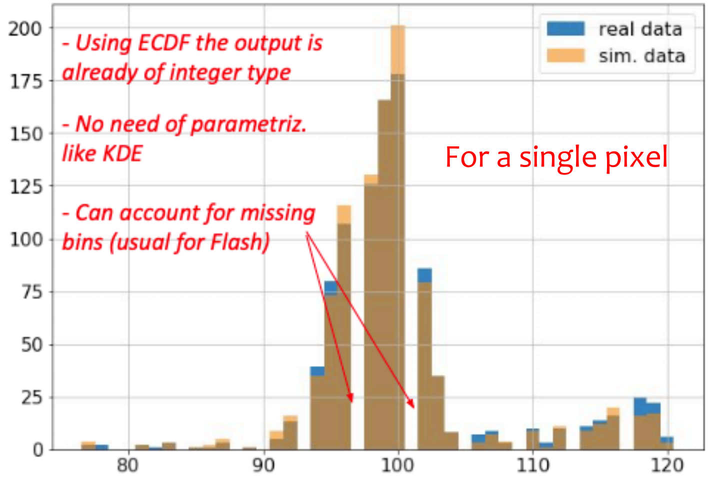
Sensor noise

A more detailed look at the pixel noise:

- The noise distribution of a pixel is not Gaussian;
  - *Is this true for all (most of) the pixels or is it a feature of only a few of them?*

- anyway, only considering a different RMS for each pixel, the "easy-fix" solution will not accurately describe the noise…
- An upgrade would be to extract the noise according to a different (not Gaussian) distribution describing the tails, but a study of several pixels noise distributions would be required…

From the Brazilian Team, as an example



An alternative proposal for noise implementation: see talk from Rafael

Diffusion – (inputs from Davide)

- So far, a constant value of σ_T=500 μm is used
- We should use:

$$\sigma_T = \sqrt{\sigma_{T_0} + D_T^2 \cdot z}$$

where:

- $\sigma_{T_0} = 300\mu$m (measured from data)
- D_T = 141 μm/√cm (from simulation?)

- Currently, it's like if we are looking at events at a distance z=8cm from the read-out.

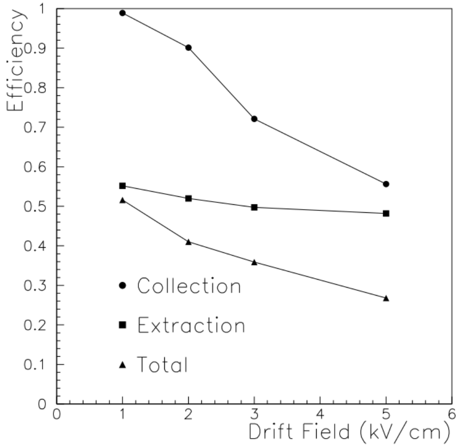
- Problem: up to now all the simulated events (NR from SRIM and ER from G4) are generated at the same z.
- Workaround: in digitization stage a random z value can be extracted and used in the formula. What to use to have a reasonable comparison with data?
  - uniform distribution?
  - In the full gas volume?

Gain & fluctuations – (inputs from Davide, again)

- So far, a constant value of 1ɣ/2eV deposit energy is used as conversion factor (for LEMON)

→ we have to describe gain fluctuations

a Maxwell simulation from Veenhof



From simulations we can assume:

- $\varepsilon \sim 0.5 \div 0.6$

 (low drift field, ε_{Coll} saturates at 1 and ε_{Extr} smoothly increasing)

- the number of secondaries is described by an exponential (leaving the bin at 0 that is the inefficiency)

- combine 3 efficiency and 3 exponentials giving an overall mean value of 1ɣ/2eV

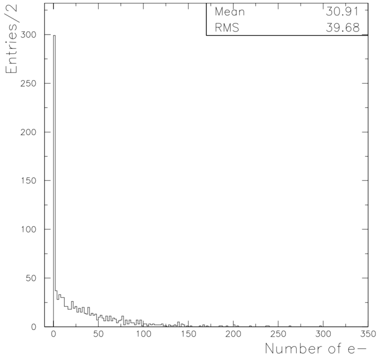


Figure 3.16: Distribution of the number of secondaries leaving the GEM

a simulation from Pinci?

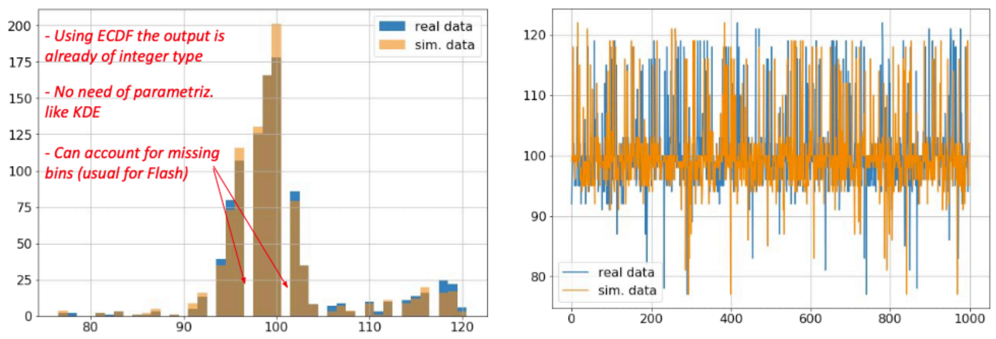
Sensor noise – the Brazilian study (as I understood it…)

- Use about 1000 images from a real run to have, for each pixel, a distribution function of the noise;
- extract the noise for each pixel according to this distribution using the Empirical Cumulative Density Function.

- At the moment, the generation of the distribution from data is done at the start of the digitization run, nothing saved/stored (time consuming?).
- Would be good to store the pixels' ECDFs as a map file (but it's 4M distributions… .)

From the Brazilian Team

Pixel (1,6) Flash sensor run 2054



Time consuming? Not so much… :

- Estimated time for 100 full images(2048x2048) ~45 minutes
- For 1000 images it is not 45 x 10 min….it is faster (not proportional)
- present code is only slightly better (but on a different machine… should compare performances directly)

CYGNO Experiment

CYGNO