PWISE analysis of fake events

Blahoslav Pastirčák & KE group

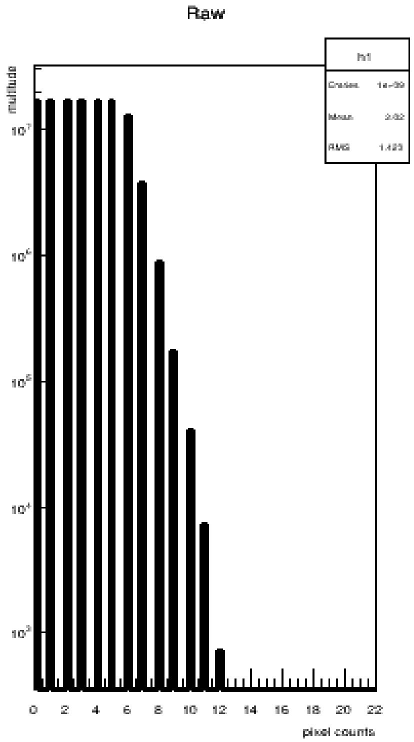
Simulation meeting Palermo June 2014

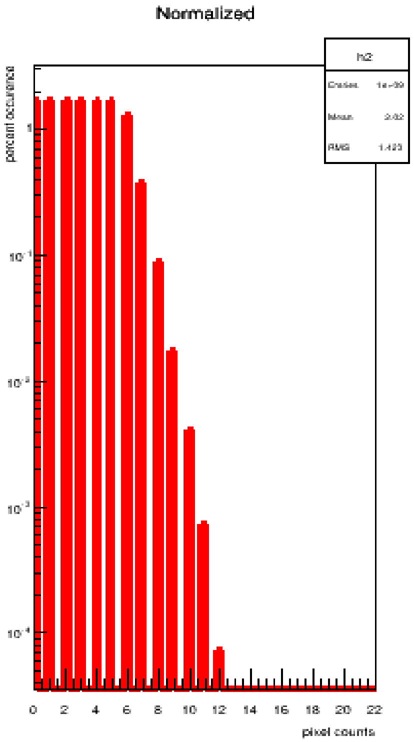
Items

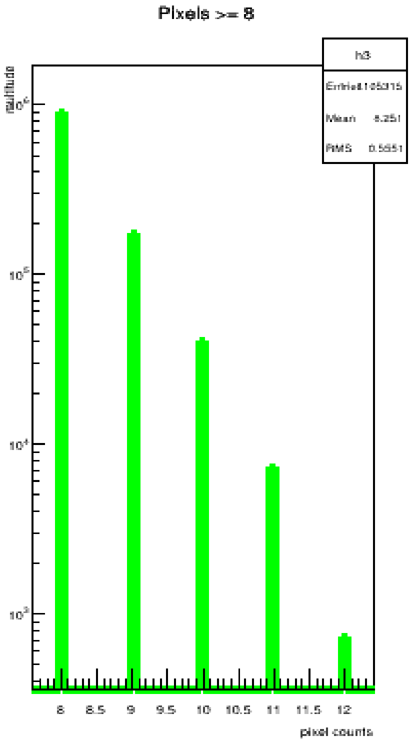
- Status of massive fake triggers bckg simulations
- PWISE approch to simulated results
- Bckg simulations for various bckg

Statistics M36

- 4e12 GTU's = 1e7s
- 10372 LTT triggers => 0.14 Hz/FS



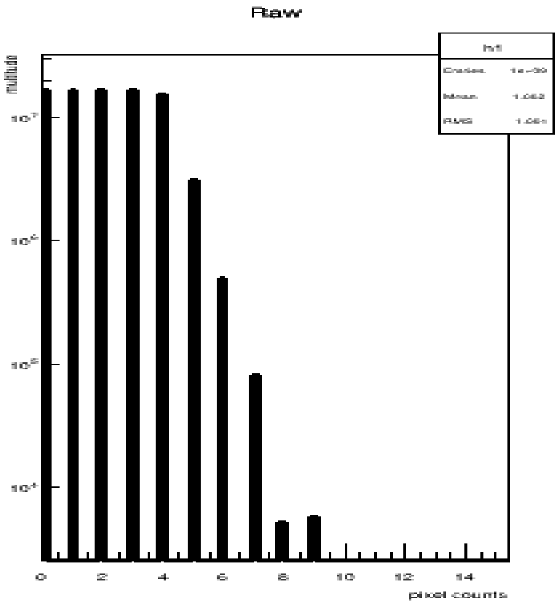


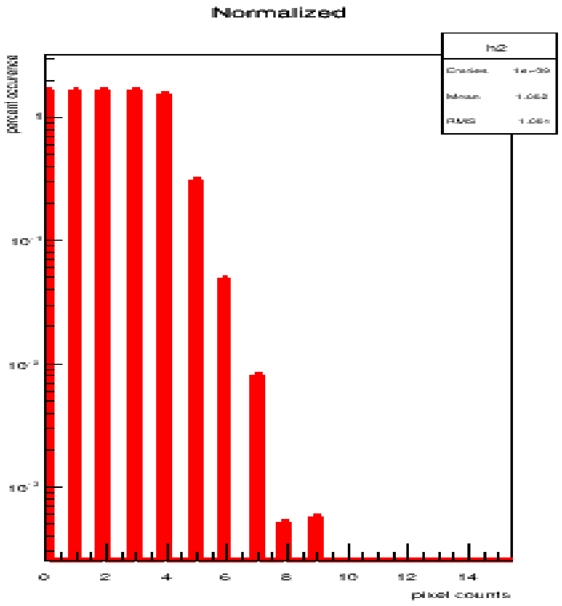


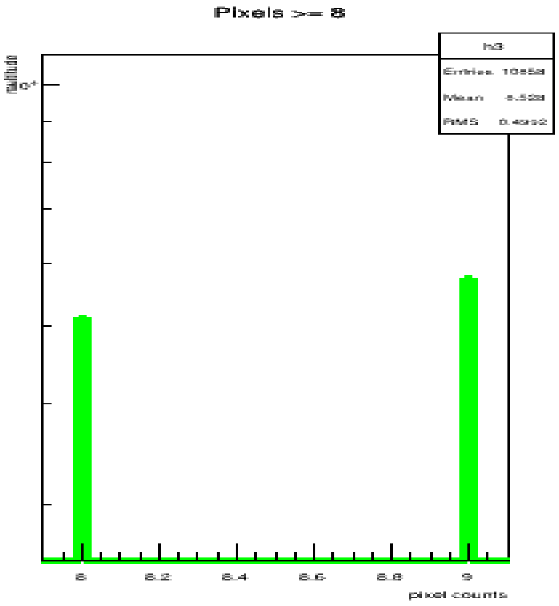
Statistics M64

- 2e13GTU's = 5e7 s
- 36500 LTT triggers => 0.1 Hz/FS

- runs with also different time windows (128, 256 GTU)
- 1.2T disk space for M64, 300 GB for M36
- Conversion to ROOT trees reduced 50-100x







Access to data

- user: upjs /passwd: +Wv#uU:"@F
- IP address: 147.213.198.208/ dir: /mnt/data

Storage:

^^^^^^^^

Data are basically stored in directories as they were simulated in runs, each run simulating 1.e9 GTU = 2500 s, so one directory means 1.e9 GTU

We have finished siumulations for M36 configuration (and all our analysis was performed only for this configuration)

The data for M36 conf are in subdirectory M36, and then in 3 subdirectories:

M36-1.e9.runs-116

M36-1.e9.runs-128

M36-1.e9.runs-576

corresponding to 3 periods during which they were obtained.The last number in

subdirectory name is number of finished runs.

We are simulating M64 configuration at present, simulation is still going on, but I have tried to collect some results on one place for you:

They are stored in M64 subdirectory.

As they were simulated on several nodes ande several times on each node, thae names of subdirectories in M64 subdirectory corresponds to this.

In these subdirectory there are then results from particular runs as described above.

Format of data:

^^^^^^^^^^^^^^^

PTT_SECOND_OUT:

(x,y,pers, ecid,counts)

1col = x of pix

2col = y of pix

3col = persistency (1-5) in GTU

4col = which EC number was counted in PDM (1-9)

5col = number counts in pix (pixel value)

(6x2)x(6x2)x5 = 720 lines/PDM for M36 configuration

(8x2)x(8x2)x5 = 1280 lines/PDM for M64 configuration

LTT:

LTT_SECOND_OUT → (x,y,time,counts)

1col = x

2col = y

3col = time in GTU (1-31), 1 present + 15 before + 15 after

4col = counts

(6x6)x(6x6)x31 = 40196 lines/PDM for M36 configuration

(8x6)x(8x6)x31 = 71424 lines/PDM for M64 configuration

PWISE

1) For each pixel only pixels whose highest photon-count is above a certain threshold – peak-threshold **pix_TH**

2) Next PWISE looks for the time window that has the highest signal-to-noise ratio (SNR) defined by:

$$SNR = \left(\frac{1}{\Delta\tau \cdot RMS} \right) \sum_{\Delta\tau} pc(t)$$

where delta Tau' s around pixels are 1, 3, 5, 7 GTU's:
t ± 0*GTU, t ± 1*GTU, t ± 2*GTU,t ± 3*GTU

3) Check if maximum SNR is above given SNR-threshold **SNRmax_TH**
Only if the SNR is above the threshold we select the photon-counts within the time window that maximizes SNR.

pix_TH = 8
SNRmax_TH =5
setuped in ESAF

PWISE run

- 3 code versions - check+comparison
- Free pascal = 1
- C++ = 1.5 FP
- ROOT (CINT) = 2.5 FP

PWISE check

- Full M36/M64 statistics analyzed
- M36: for pix_TH = 8, SNRmax_TH =5
  only pattern lengths of 2 and 3 appeared
- M64: no PWISE for these thresholds
- Moving thresholds down provide PWISE with patterns upto 8 (M36) and shows much more “sensitivity” to SNRMax_TH then to pix_TH

Summary

- **M64/M36 simulations finished**

- **For M64 PWISE on fake triggers confirmed on full statistics that are practically all removed from further analysis**

Simulations for BG_level-vs-Trigger_thresholds for various bckg

for K-EUSO needs starts/continue