Status of fake trigger background simulation

- Blahoslav Pastirčák, Pavol Bobík,
- Franceso Fenu, Kenji Shinozaki

JEM-EUSO General meeting, Moscow Russia,
30 May 2012

Outline

- Motivation
- SW and configuration
- Accumulated information
- Analysis, pattern recognition attempt

Motivation

- Aim of trigger is to detect the signal from real event among extremely high background (~10^{11} counts/s/FS)
- Signal filtered on several levels reducing the trigger rate
- Organized in two main levels, basing on positions and time correlations of physical events compared to background

Outline of noise reduction capability.

Level		**Rate of signals/triggers at PDM level**	**Rate of signals/triggers at FS level**
PDM level trigger	Photon trigger	~9.2×10^{8} Hz	~1.4×10^{11} Hz
	Counting trigger	~7.1×10^{5} Hz	~1.1×10^{8} Hz
	Persistency track trigger (PTT)	~7 Hz	~10^{3} Hz
PDM cluster level trigger (FS=144 PDM's)	Linear track trigger (LTT)	~6.7×10^{-4} Hz	~0.1 Hz
Expected rate of cosmic ray events		~6.7×10^{-6} Hz	~10^{-3} Hz

Motivation

- Very high statistics of simulated background needed
  → 10^5 events → 10^{14} GTU's
- Impossible to simulate by ESAF:
  → 10^3 slower then used code
  → cannot be computed parallely (mem. share)
- Fast and standalone code written in C++ developed by Francseco Fenu

The code

- Trigger algorithm implemented (as in ESAF)
- One PDM simulated
- Persistency Track Trigger algorithm → 1^{st} level → 1Hz/PDM
- Linear Track Trigger algorithm → 2^{nd} level → 1 mHZ/PDM
- Background source → Poisson distribution of average 500 photons (m^{-2} s^{-1} sr^{-1}) = 2.1 photons/pixel/GTU
- Code fast but since to produce huge statistics has to be run in parallel (on Kosice cluster)
- Minimal needed statistics obtained by a year of continuos computing

JEM-EUSO Kosice cluster

- Actually used and available for also for collaboration
  7*32 cores @ 2.3 Ghz; 25 TB
  upgrade and RAID configuration in progress right now
- 1.2.5-2.fc14 kernel 2.6.35.13-91.fc14.x86_64, gcc 4.5.1-4
- ROOT v32.00, ESAF trunk, GEANT4 9.4

M36 Configuration

M36

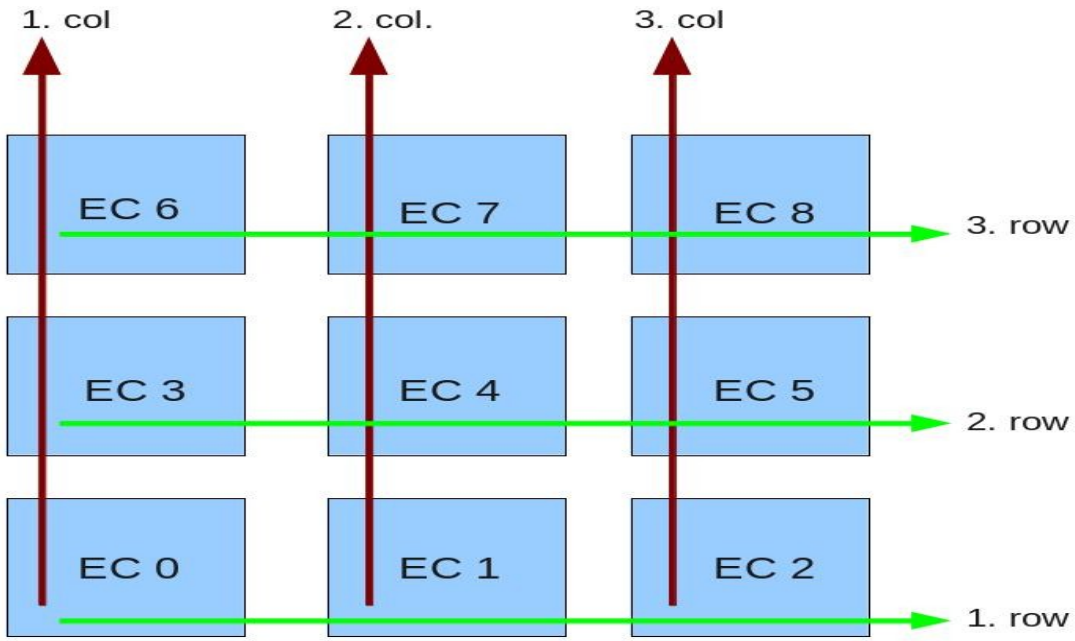
BG = 2.1 ph/pix/GTU

PTT_integr = 43

LTT integration = 145

Consecutive GTU = 5

Yellow pixel th = 4



1 PDM = 9 EC = 1296 pixel

1 EC = 4 x PMT = 144 pixel

1 PMT = M 36 = 36 pixel (6 x 6)

M36 Stored information

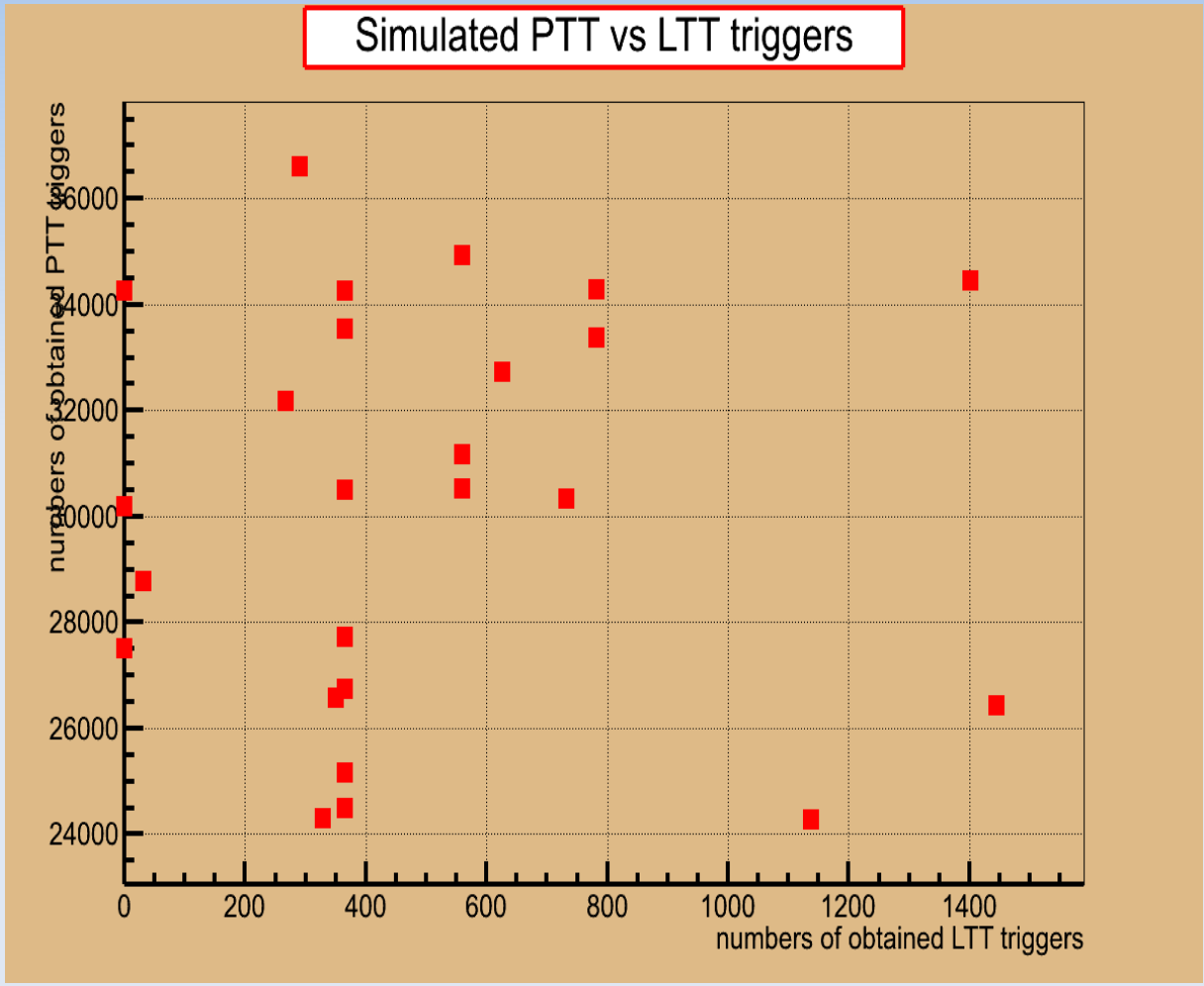
- Store data filtered on PTT and LTT levels
- Two files written when thresholds reached:
- PTT_SECOND_OUT → (x,y,pers, ecid,counts)

  12x12x5 =720 lines/pdm
- LTT_SECOND_OUT → (x,y,time,counts)

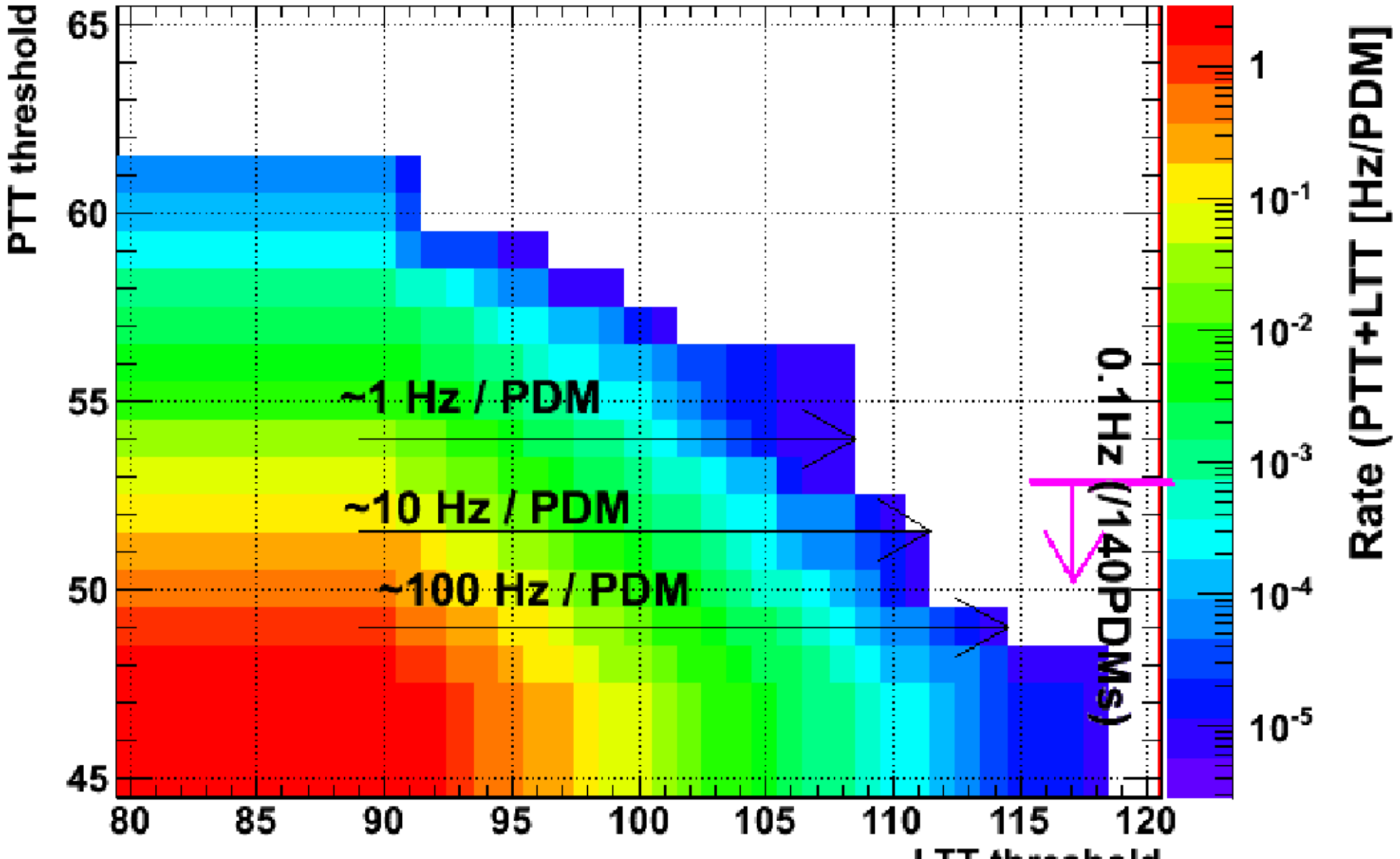
  36x36x31 = 40196 lines/pdm
- Information for which PPT, LTT dumped (reached threshold)
- Analysis of only pixels contributed to LTT

M36 Present statistics

- 10^{12} GTU's
  6 weeks simulation
- 12000 LTT triggers
  → 1mHz/PDM
- 750000 PTT triggers
  → 0.1 Hz/PDM



M64 rates


PTT threshold
65
60
55
50
45
80
85
90
95
100
105
110
115
120
LTT threshold
~1 Hz / PDM
~10 Hz / PDM
~100 Hz / PDM
0.1 Hz (1/140PDMs)
Rate (PTT+LTT [Hz/PDM]
1
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}

M64 Configuration

- Modification of BG according $(36/64)^2$ → scaled
- PTT and LTT integration thresholds modification following obtained background rates for M64

M64

BG = 0.4 cts/ms

PTT_integr = 52

LTT integration = 115

Consecutive GTU = 5

Yellow pixel th = 2

M64 Stored information

- Store data filtered on PTT and LTT levels
- Two files written when thresholds reached:
- PTT_SECOND_OUT → (x,y,pers, ecid,counts)

  16x16x5 =1280 lines/pdm
- LTT_SECOND_OUT → (x,y,time,counts)

  48x48x31 = 71424 lines/pdm
- Information for which PPT, LTT dumped (reached threshold)
- Analysis of only pixels contributed to LTT

M64 Present statistics

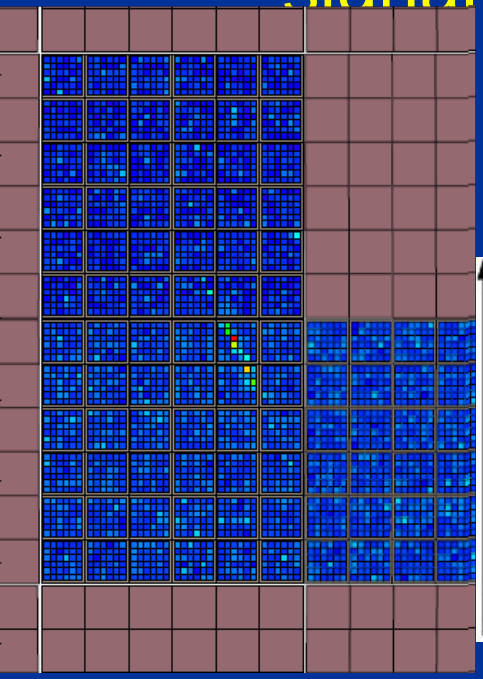
- still running
- at present 2 weeks simulation
- → 10^{11} GTU's
- 60000 PTT triggers → 0.1 Hz/PDM

Pattern recognition

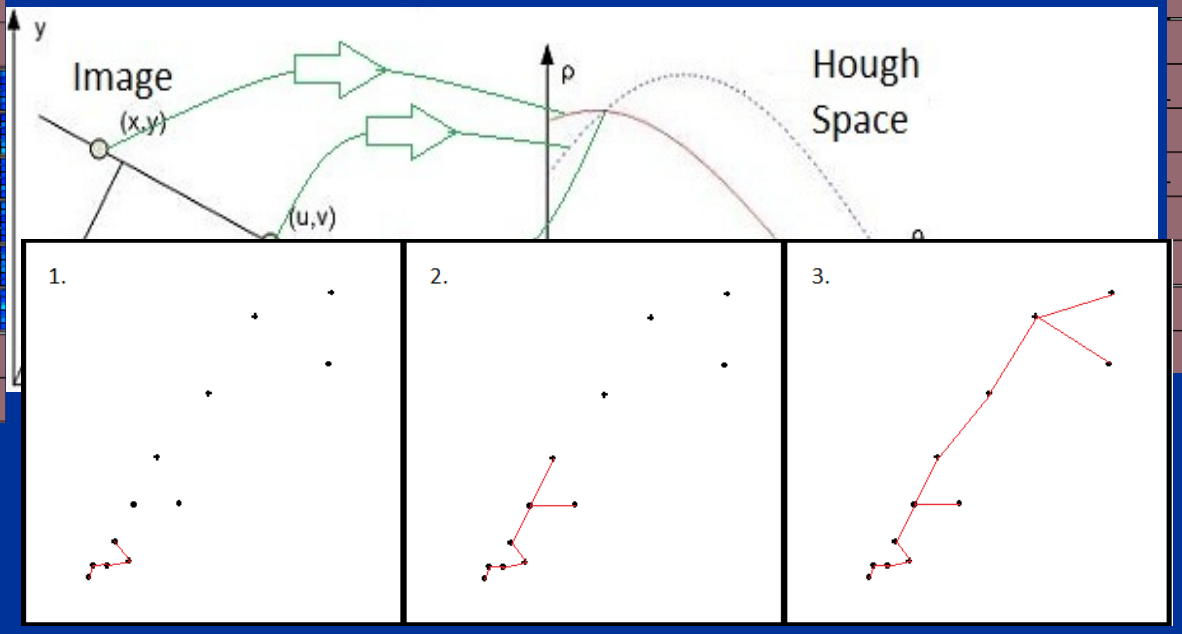
- We have started to study pattern recognition on obtained result
- 1) Adaption of RobustModule.cc code prepared by Svetlana for ESAF
- 2) Finding fake patterens in only randomly produced noise in 8x8 matrices

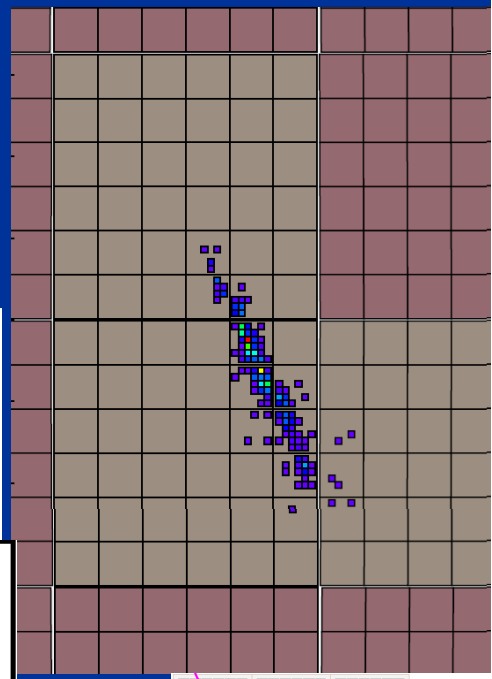
Pattern Recognition

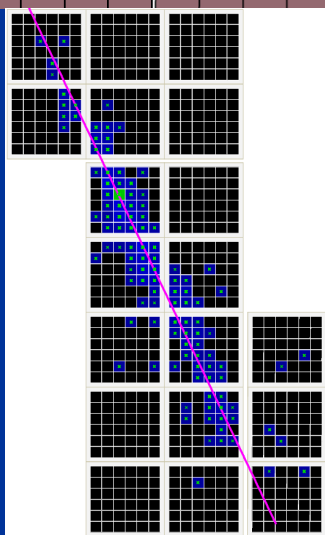
Hough transformation or clustering to disentangle signal



from background…

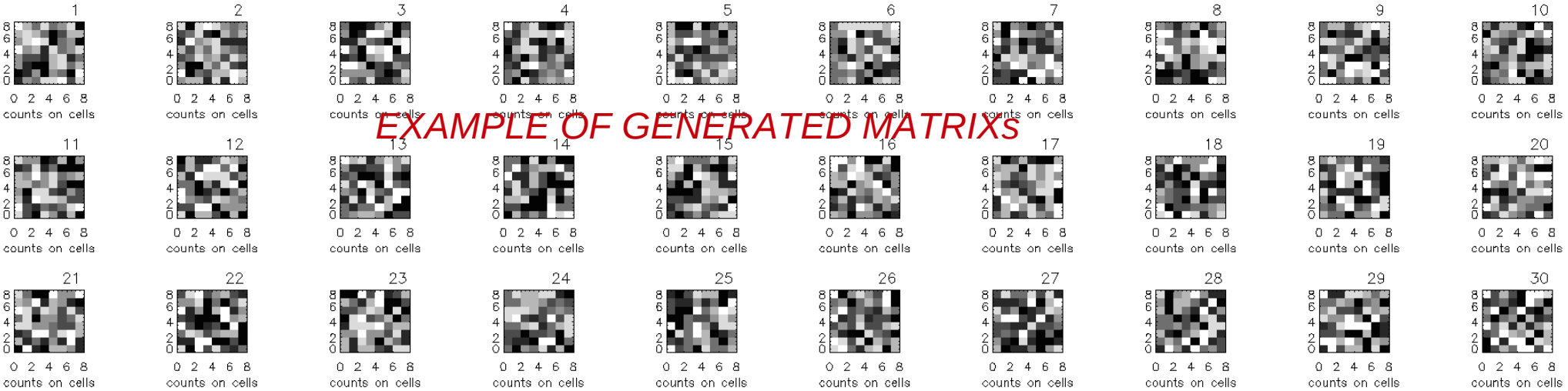


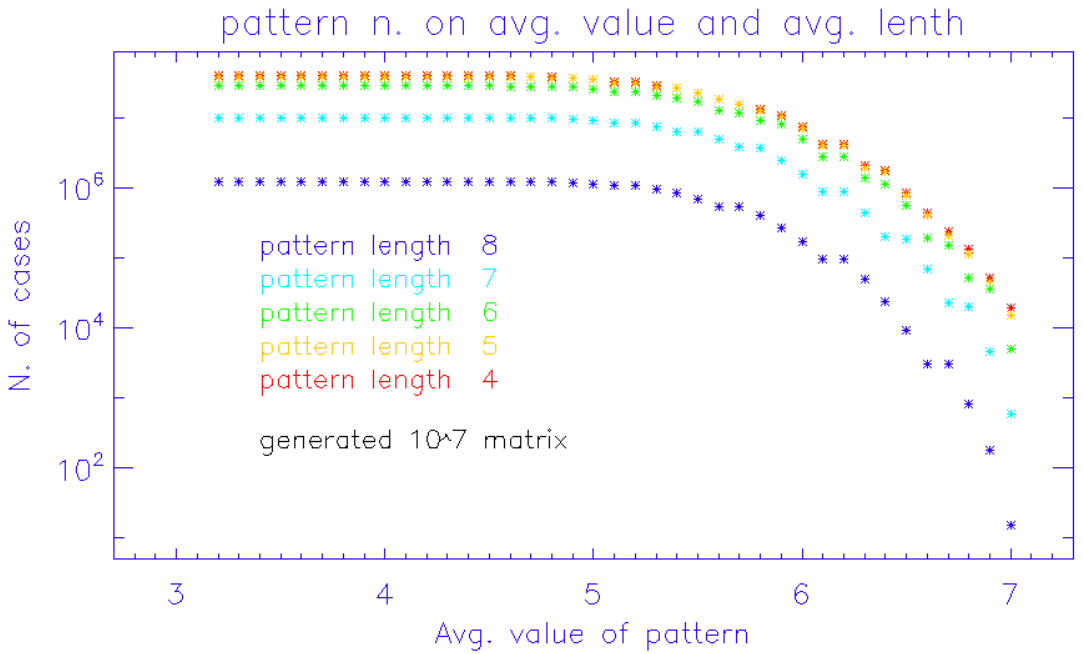




…and to determine the geometrical parameters of the track.

Pattern recognition - Hough transform





- M. Staroň diploma work - started work

- generated N matrix 8 x 8 pixels (like PMT) with values on pixels 0-7 (uniformly distributed random values)

- **Hough transform** applied
  - static pattern - work well
  - moving pattern - work well

- fake patterns in generated matrix
  - seems in range predicted by theory

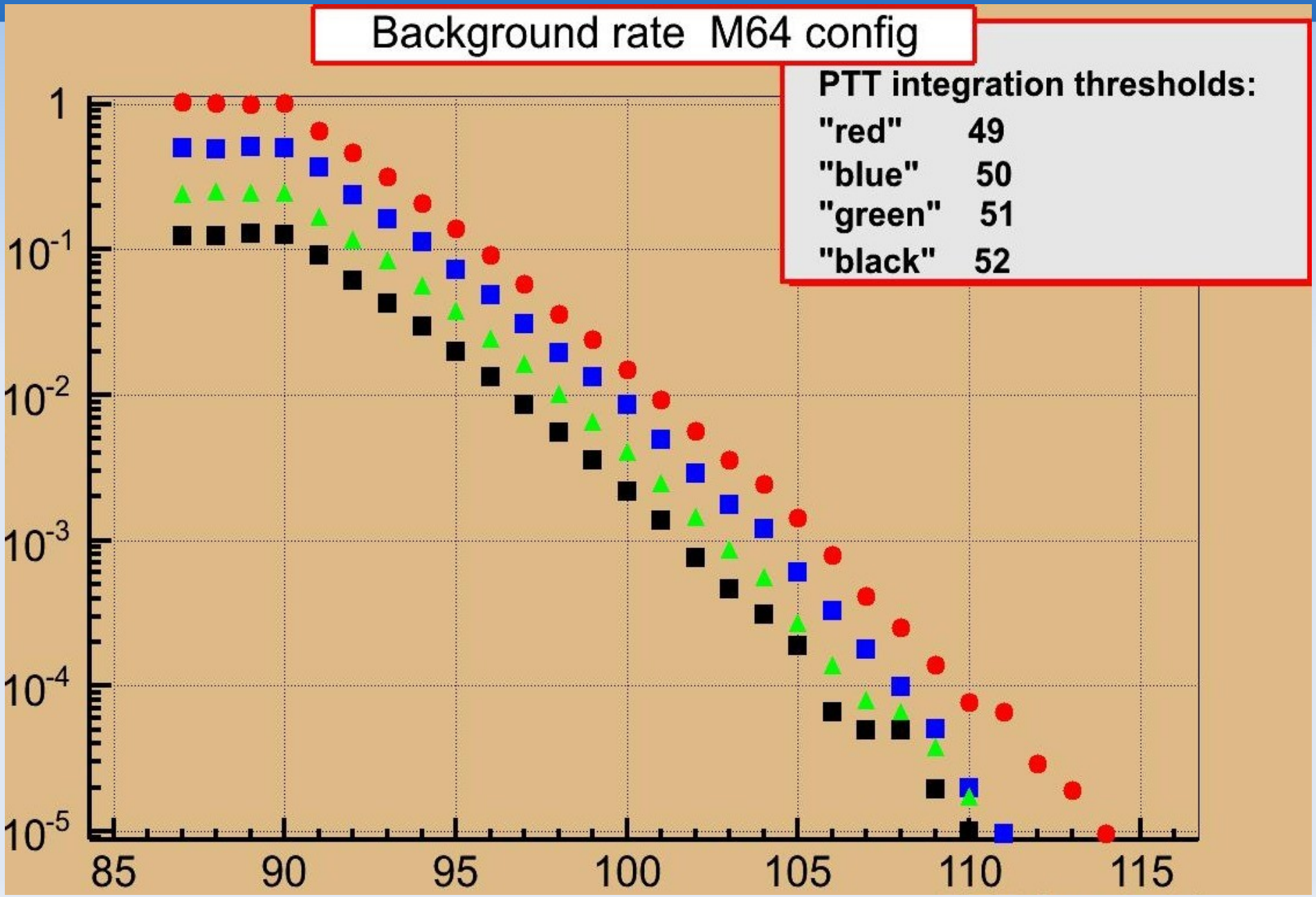
- pattern characteristics
  – average patter value, pattern length

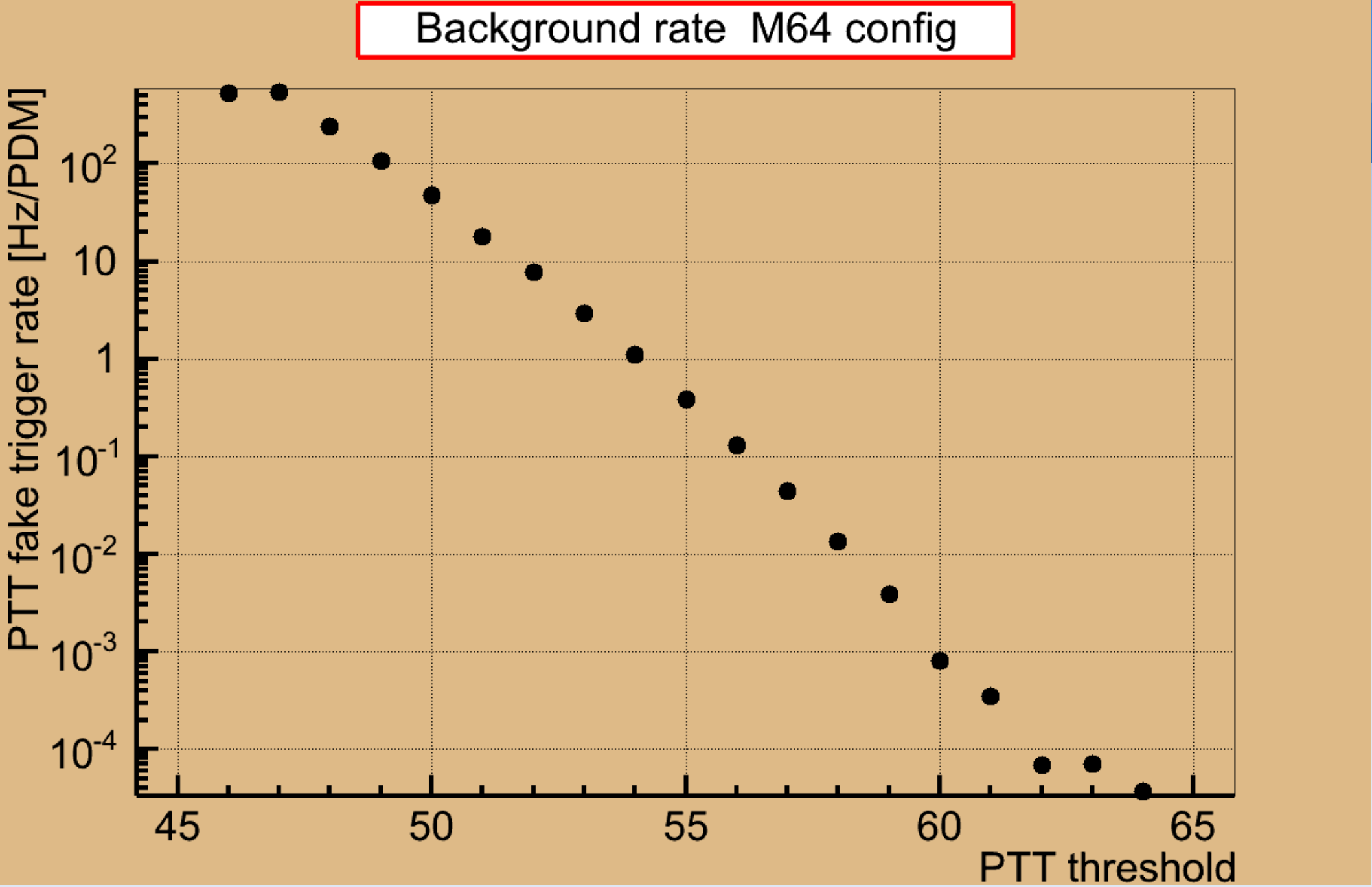
- direction of work – Hough transform verification for JEM-EUSO fake trigger simulations

Summary, todo

- Checked trigger rates obtained from the code compatible with expectation
- Optimization for M64 configuration and massive simulation started
- Pattern recognition of the obtained data continuing study

LTT threshold for M64




Background rate M64 config
PTT fake trigger rate [Hz/PDM]
10^{2}
10
1
10^{-1}
10^{-2}
10^{-3}
10^{-4}
45
50
55
60
65
PTT threshold