VTX Digitizer

Ch. Finck

Introduction

Number of pixels

Pixel distributions

Pileup events

Conclusions

Introduction

MC hits ➡ pixel hits (seed) ➡ kFilter

- MC hits (x, y, eloss) fired only one pixel

Introduction

MC hits ➡ pixel hits (seed) ➡ kFilter

- MC hits (x, y, eloss) fired only one pixel

MC hits ➡ pixel hits ➡ clusters ➡ kFilter

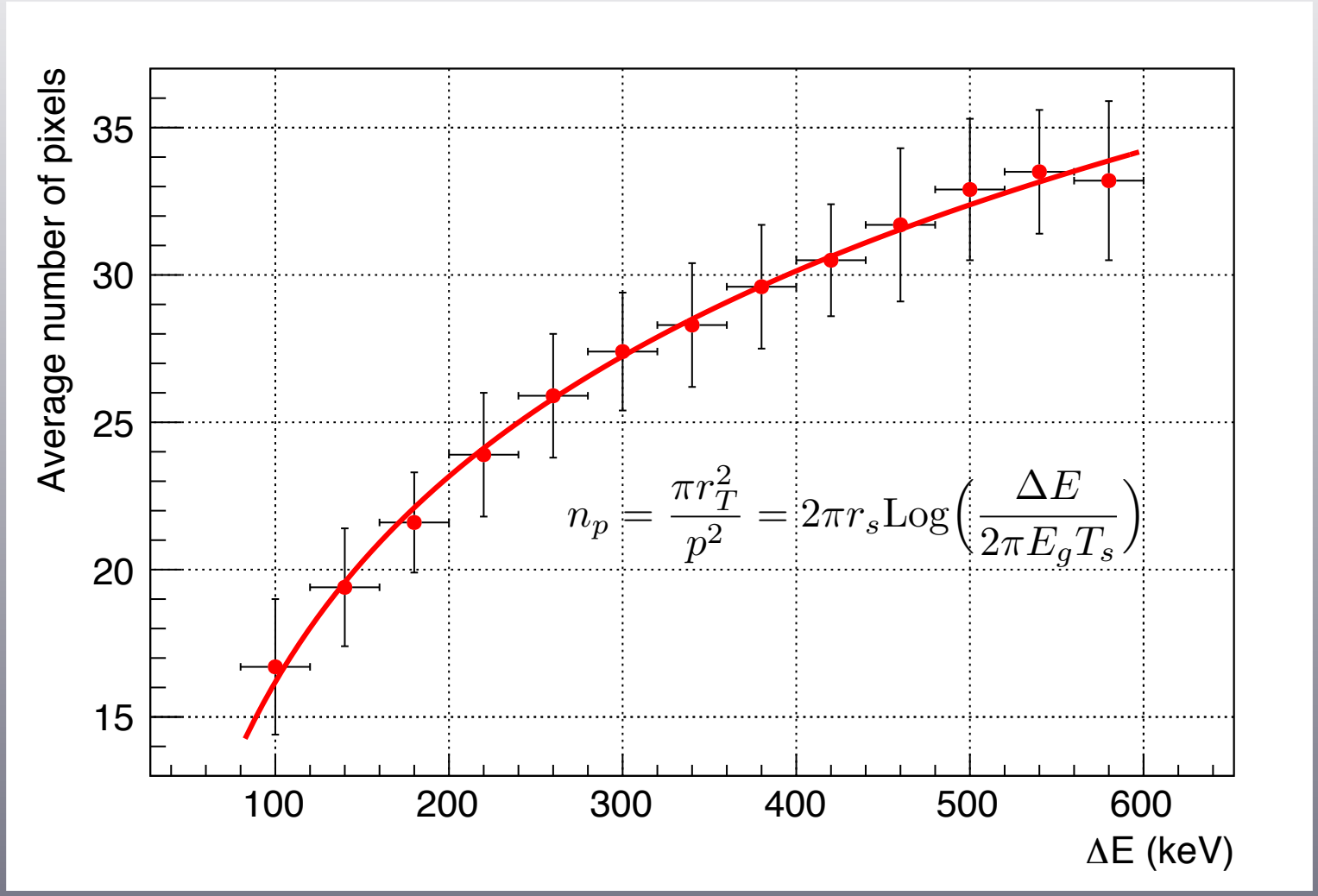
- MC hits (x, y, eloss) fired n pixels of a cluster

➡ Need a digitizer from (x, y, eloss) to cluster size.

➡Inverse mapping (x,y) to (col, line)

Number of pixels (i)

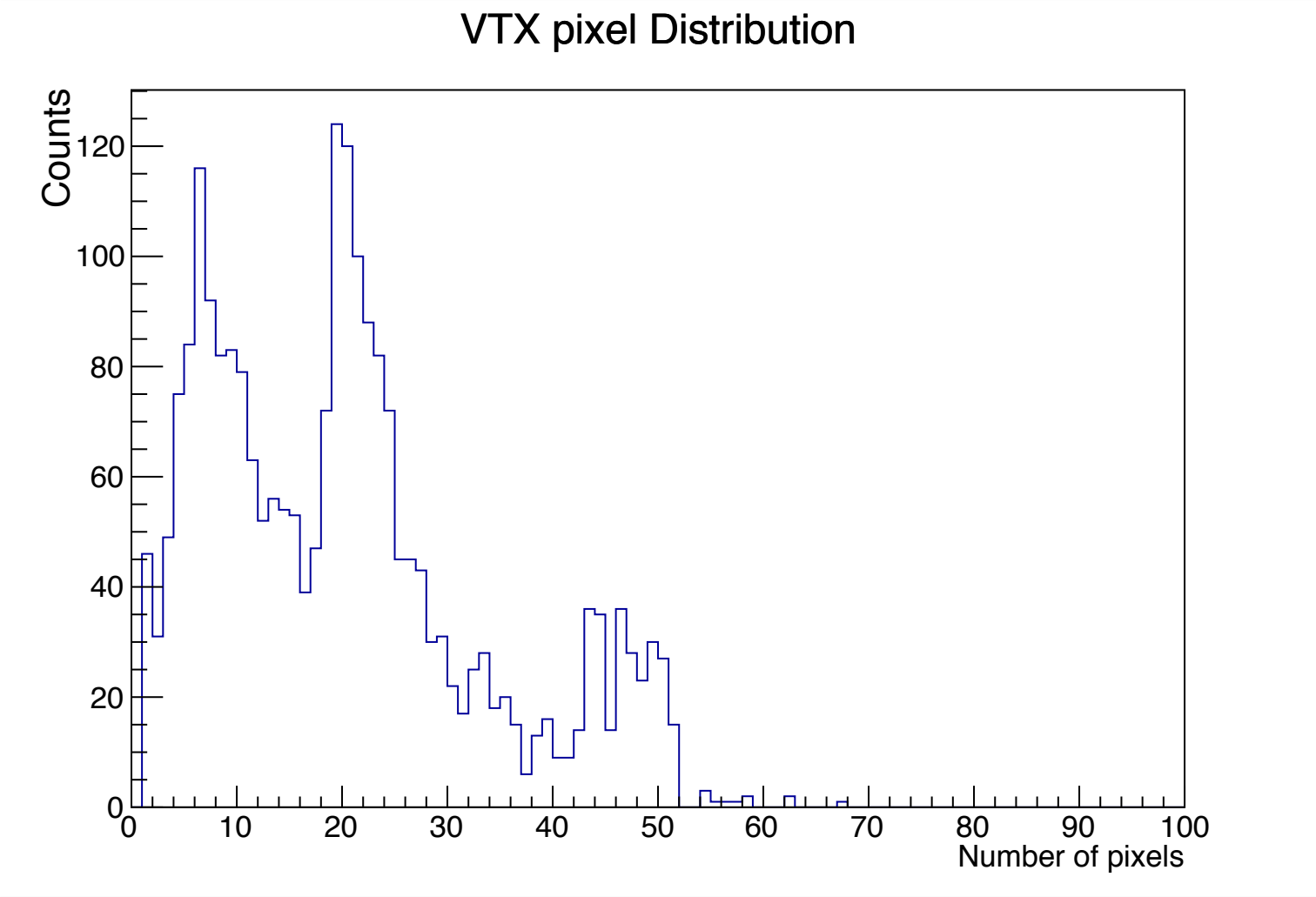
- Empirical model (accepted in MIM)



- Two free parameters r_s and T_s.

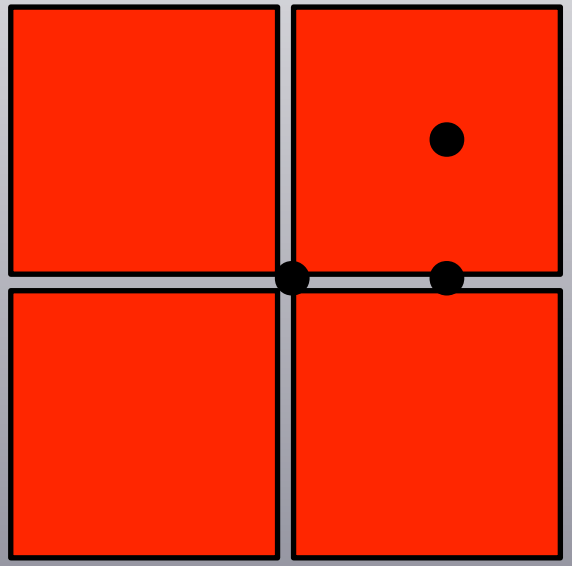
Number of pixels (ii)

- Example for ^{16}O beam on C_2H_4 target @ 200 MeV



Pixel distributions (i)

- Arrange the number of pixels n_p over the 2D plane



- hit impact can have 3 configurations
  - btw 4 pixels
  - btw 2 pixels
  - center of a pixels
- All other configurations by symmetry

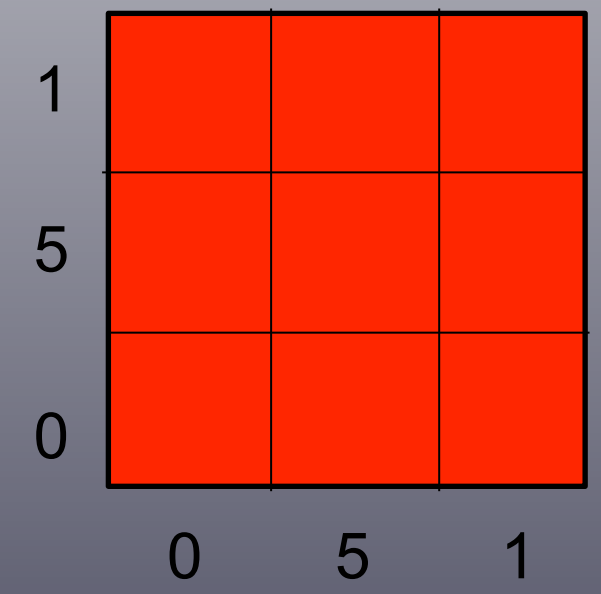
Pixel distributions (ii)

- Methods

  - Randomly draw a 2D Gaussian shape
    - Works well
    - But becomes extremely slow for n_p >>1
    - Under progress…

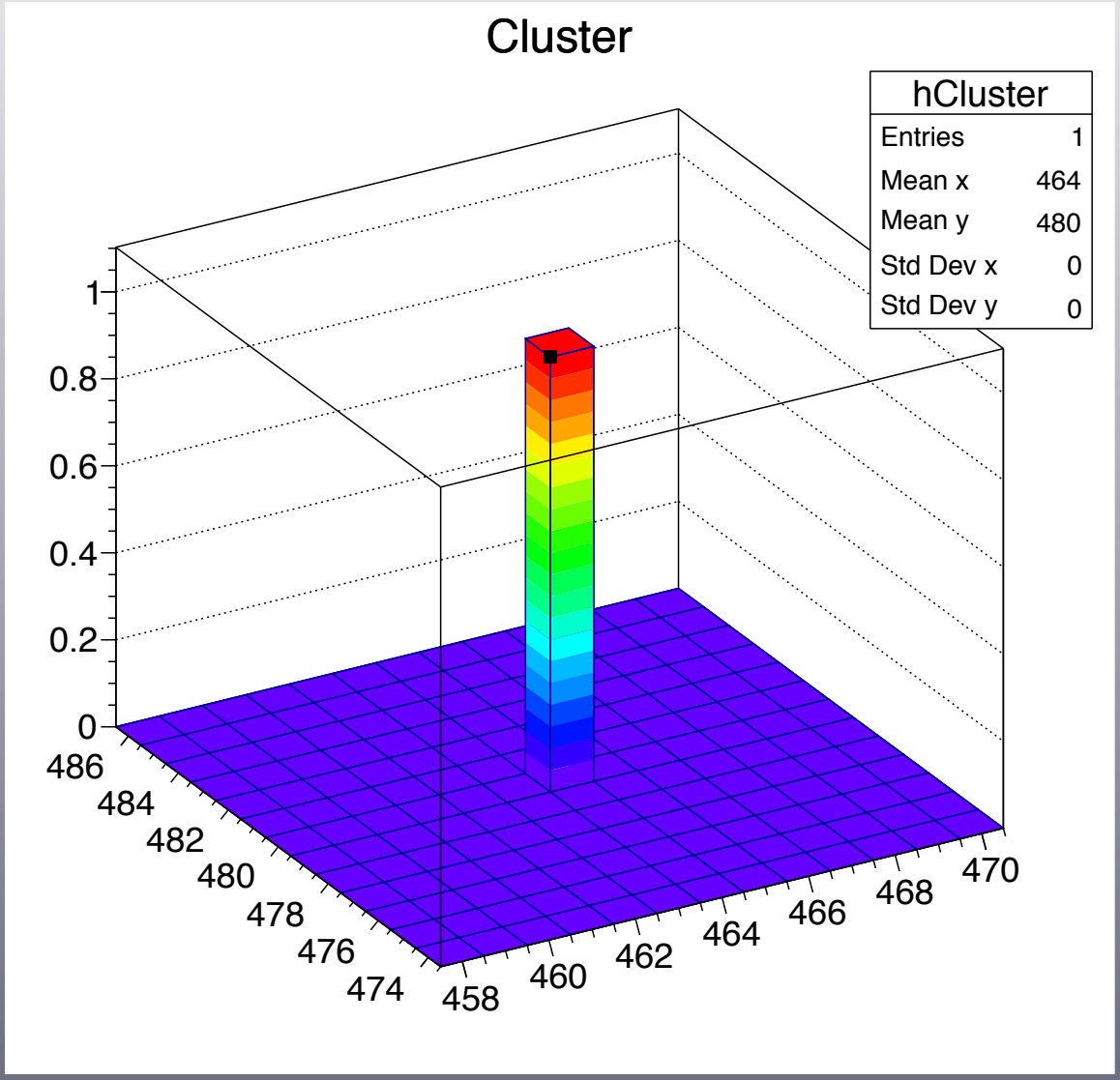
  - Draw concentrical circle with a growing radius with n_p.
    - Works well
    - More static behavior
    - Cannot take into account tilted tracks
    - Very fast.

    - Subdivided pixels in 9 sub-sectors



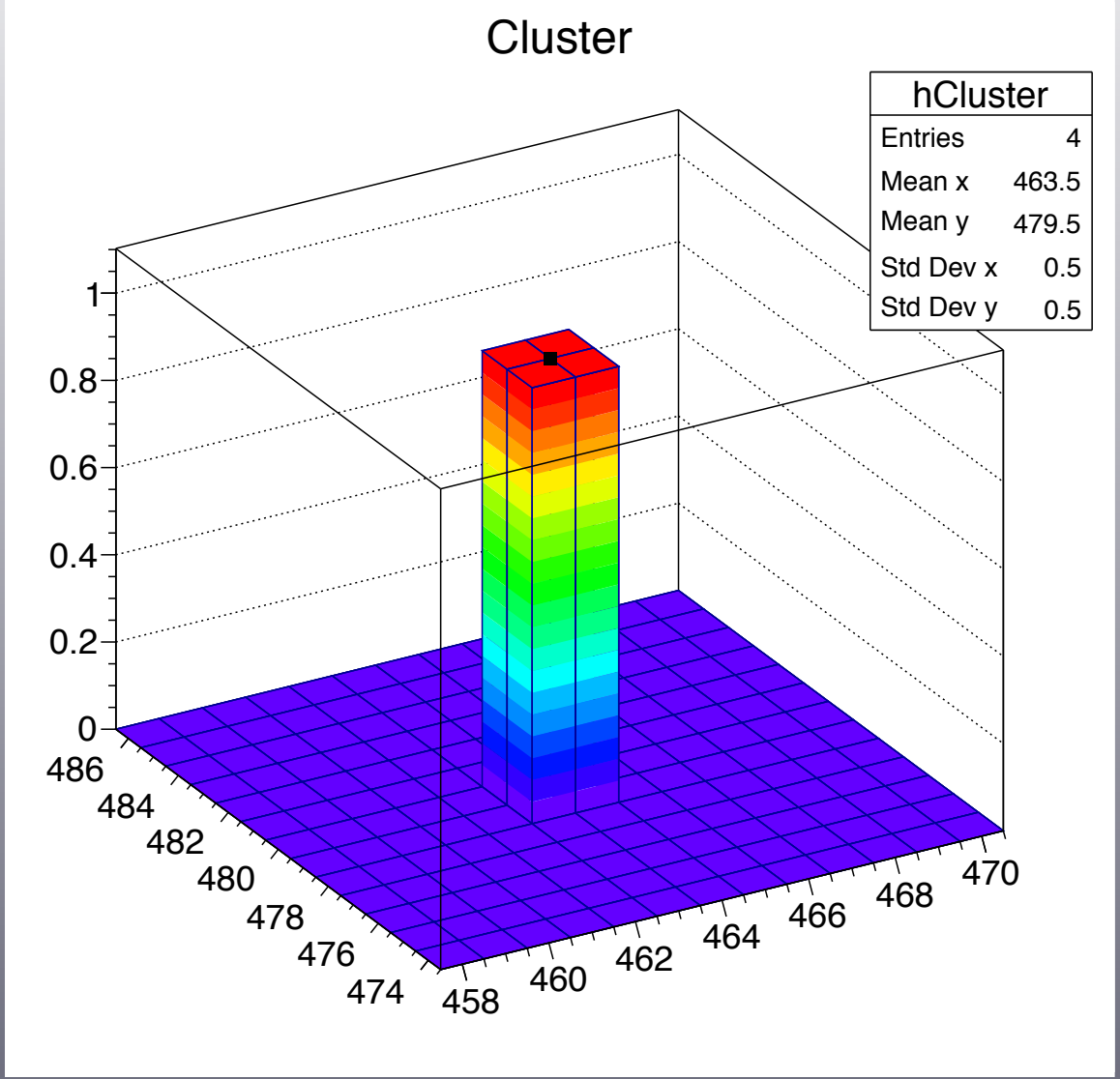
Radius Pixel distributions (i)

- Pixels distribution for configuration (0,0)



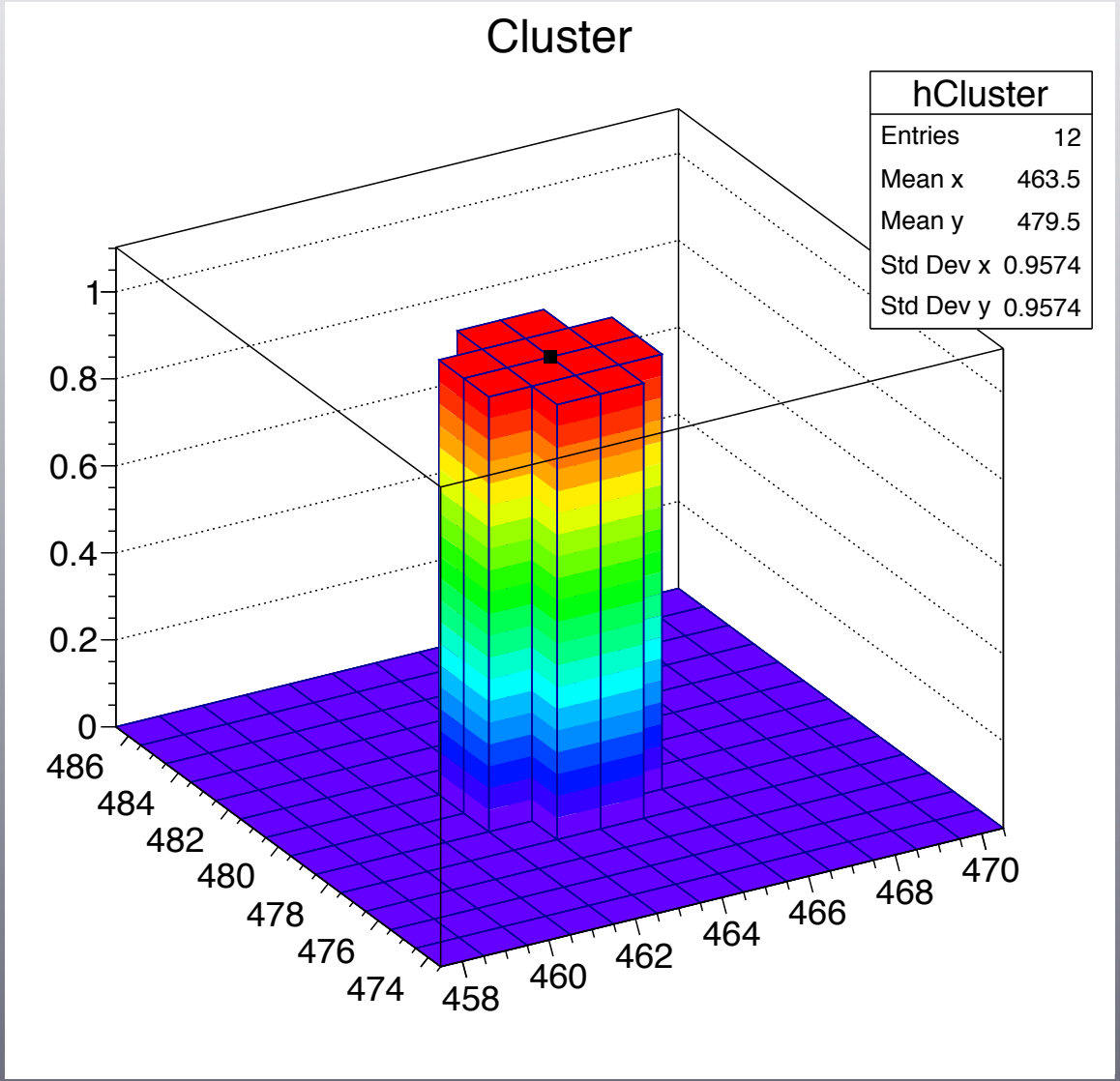
Radius Pixel distributions (i)

- Pixels distribution for configuration (0,0)



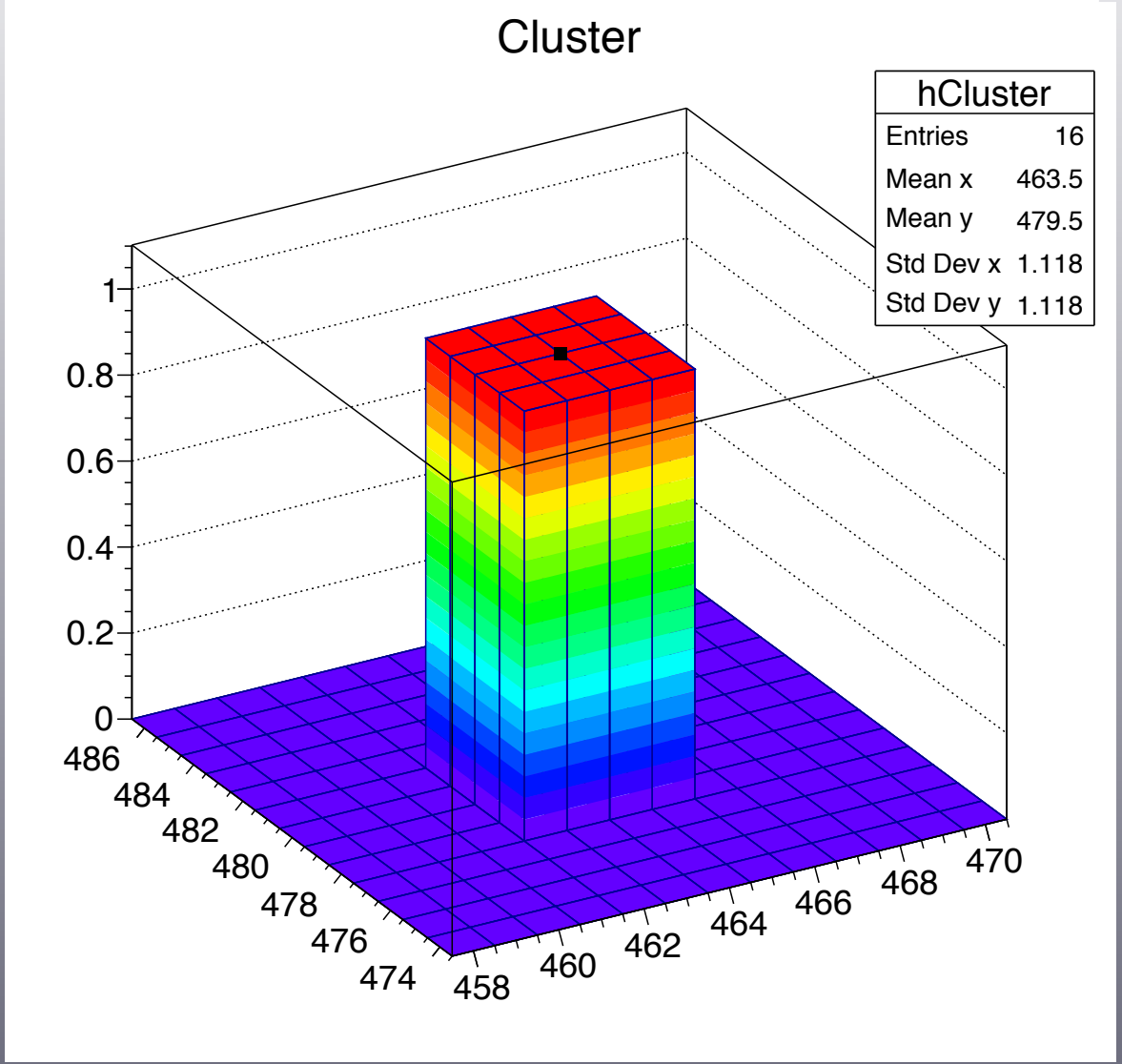
Radius Pixel distributions (i)

- Pixels distribution for configuration (0,0)



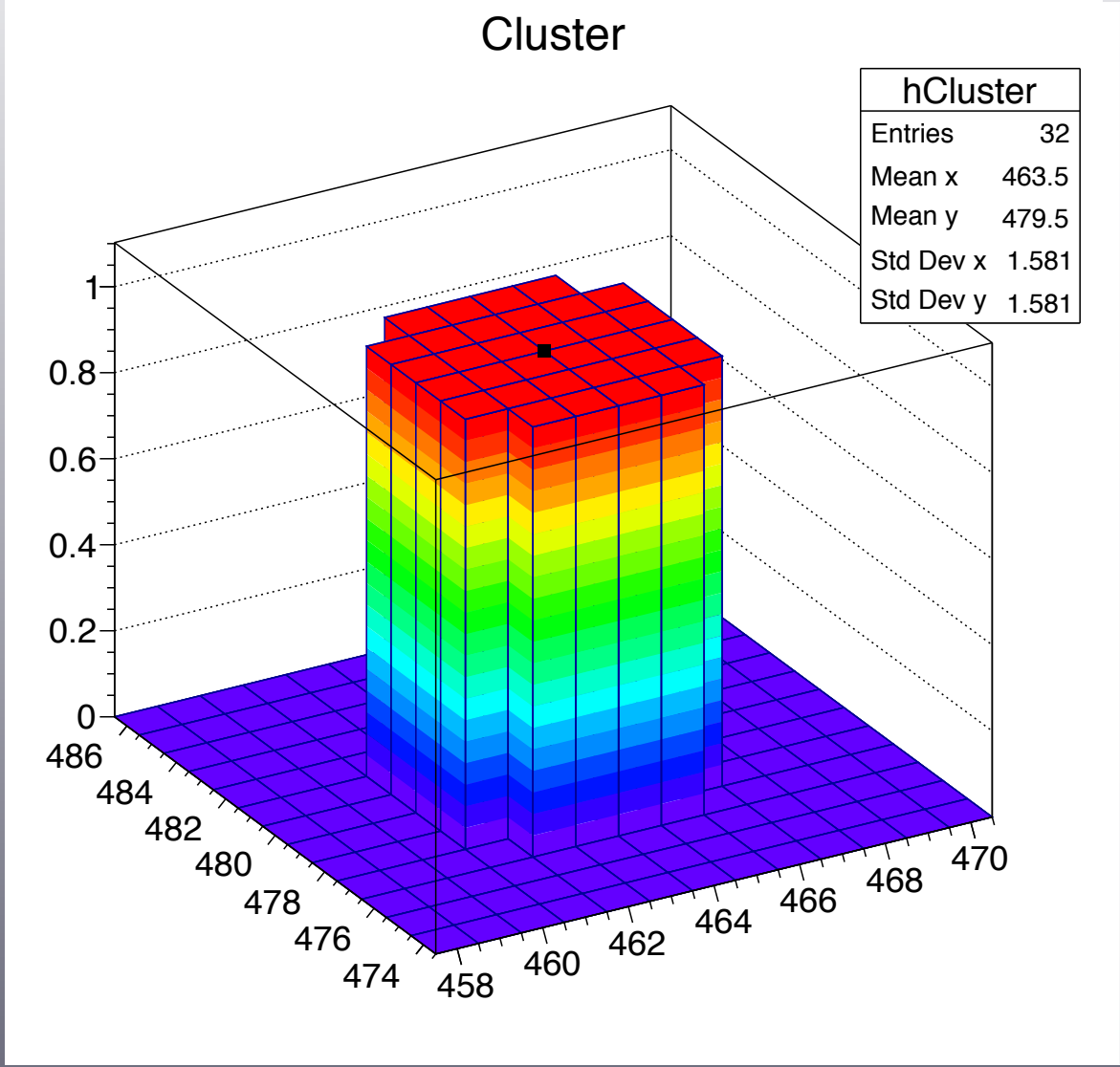
Radius Pixel distributions (i)

- Pixels distribution for configuration (0,0)



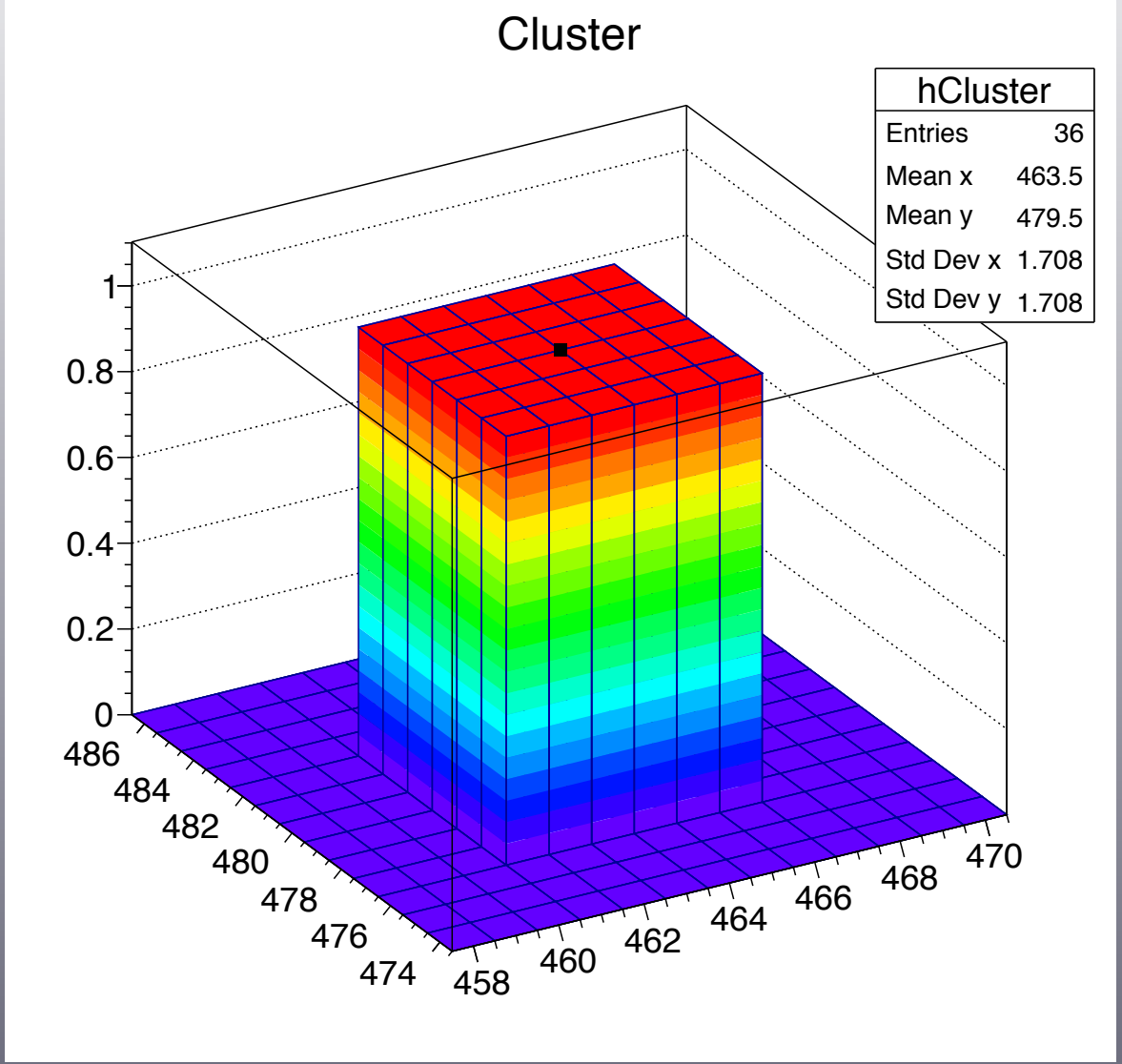
Radius Pixel distributions (i)

- Pixels distribution for configuration (0,0)



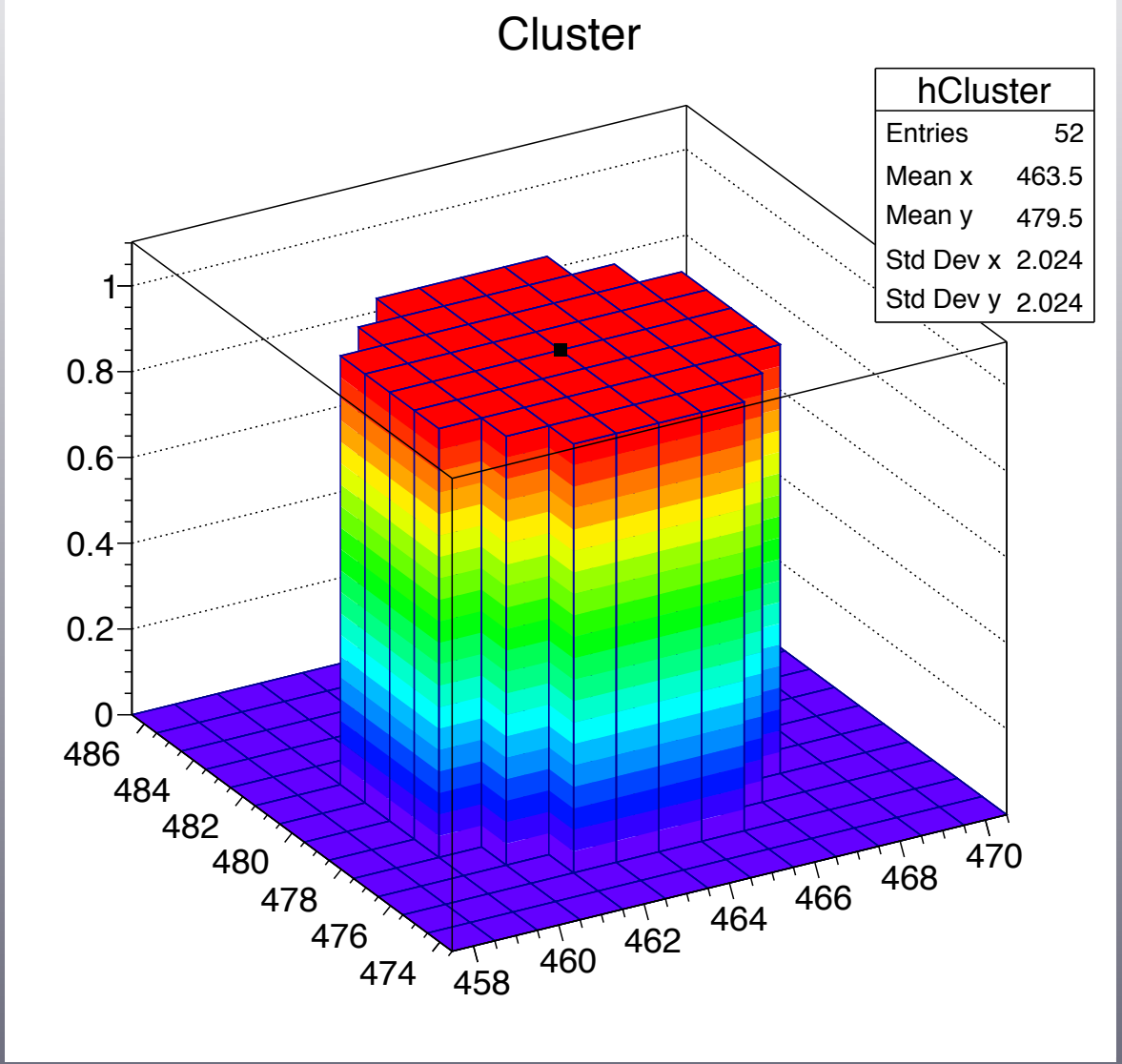
Radius Pixel distributions (i)

- Pixels distribution for configuration (0,0)



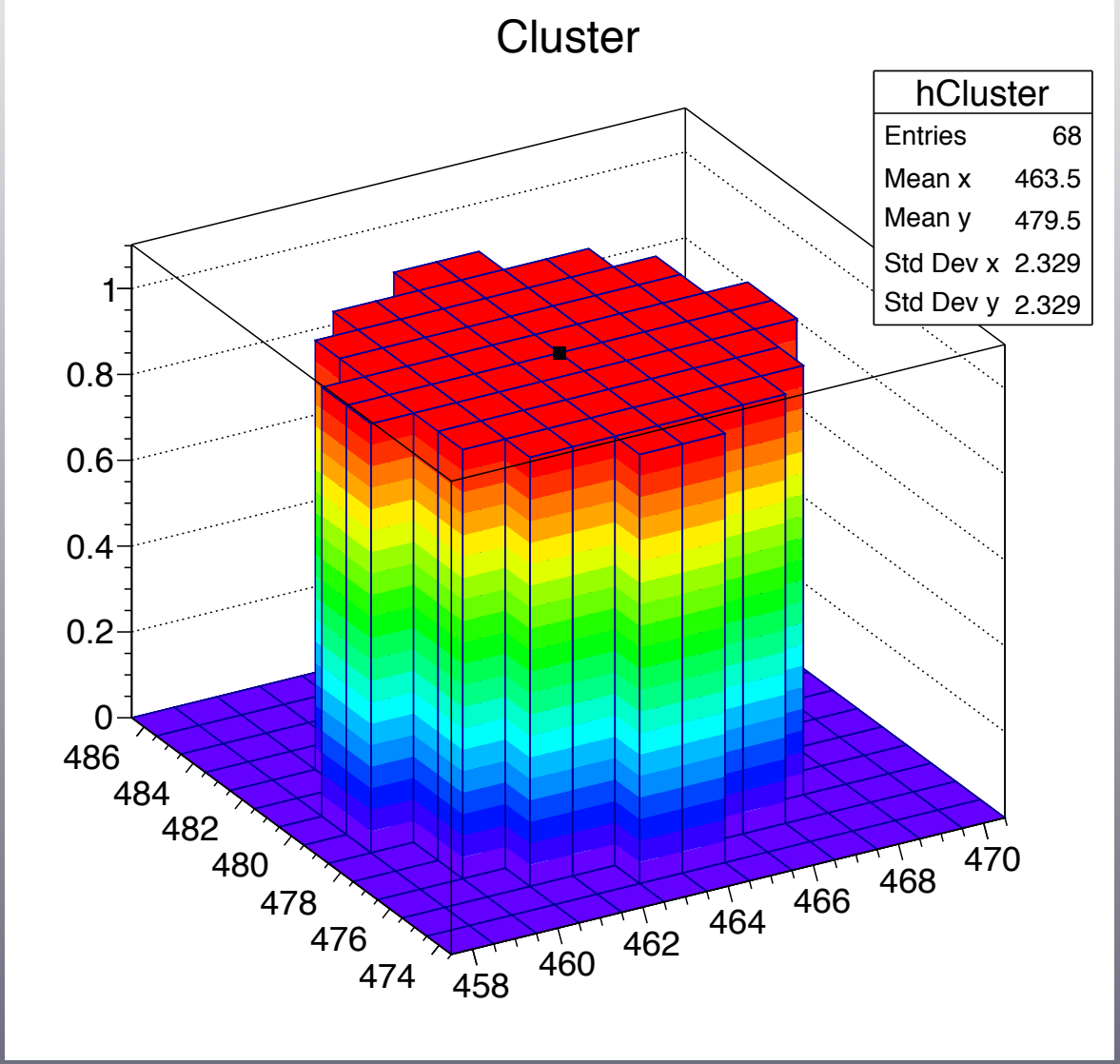
Radius Pixel distributions (i)

- Pixels distribution for configuration (0,0)



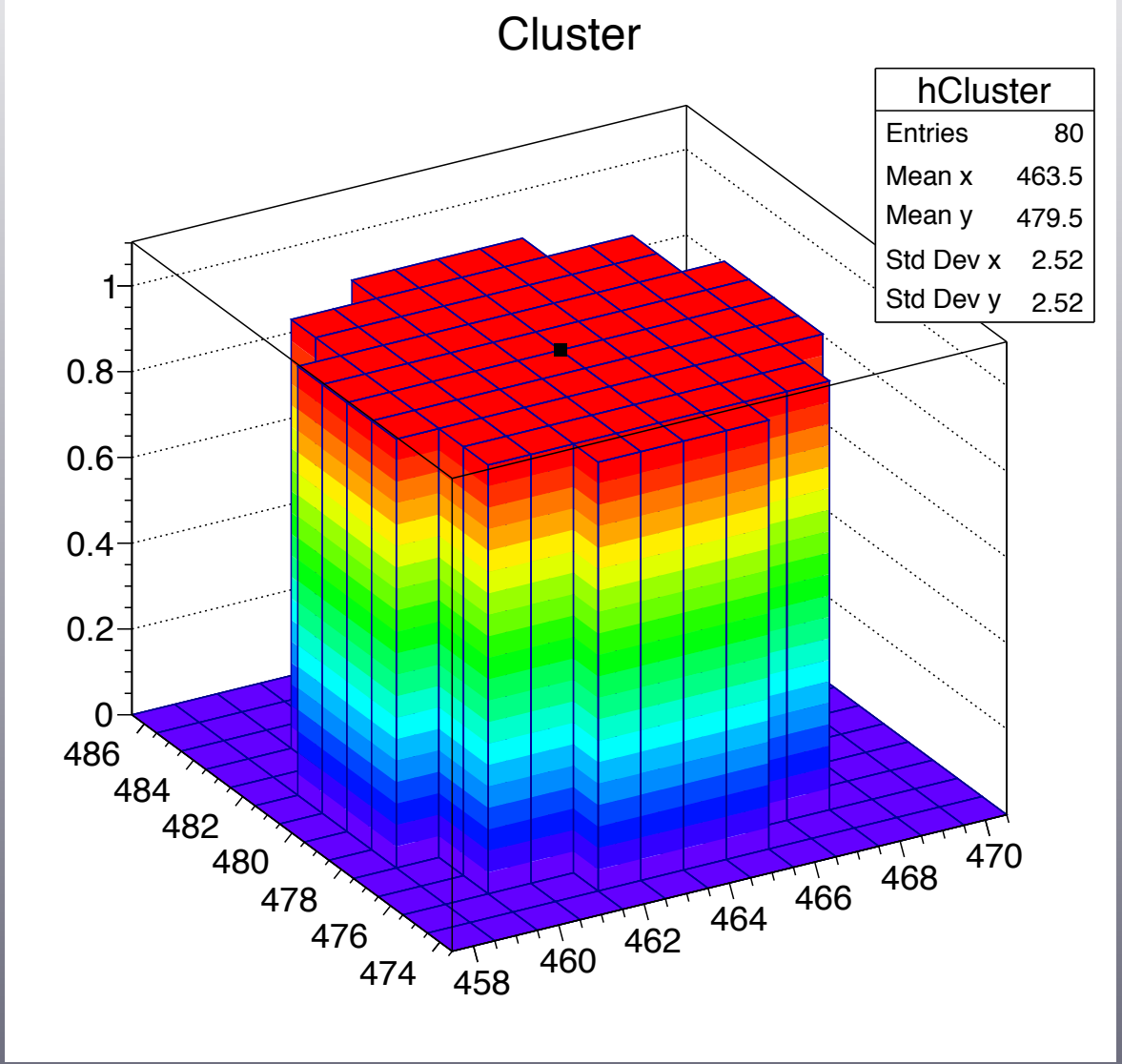
Radius Pixel distributions (i)

- Pixels distribution for configuration (0,0)



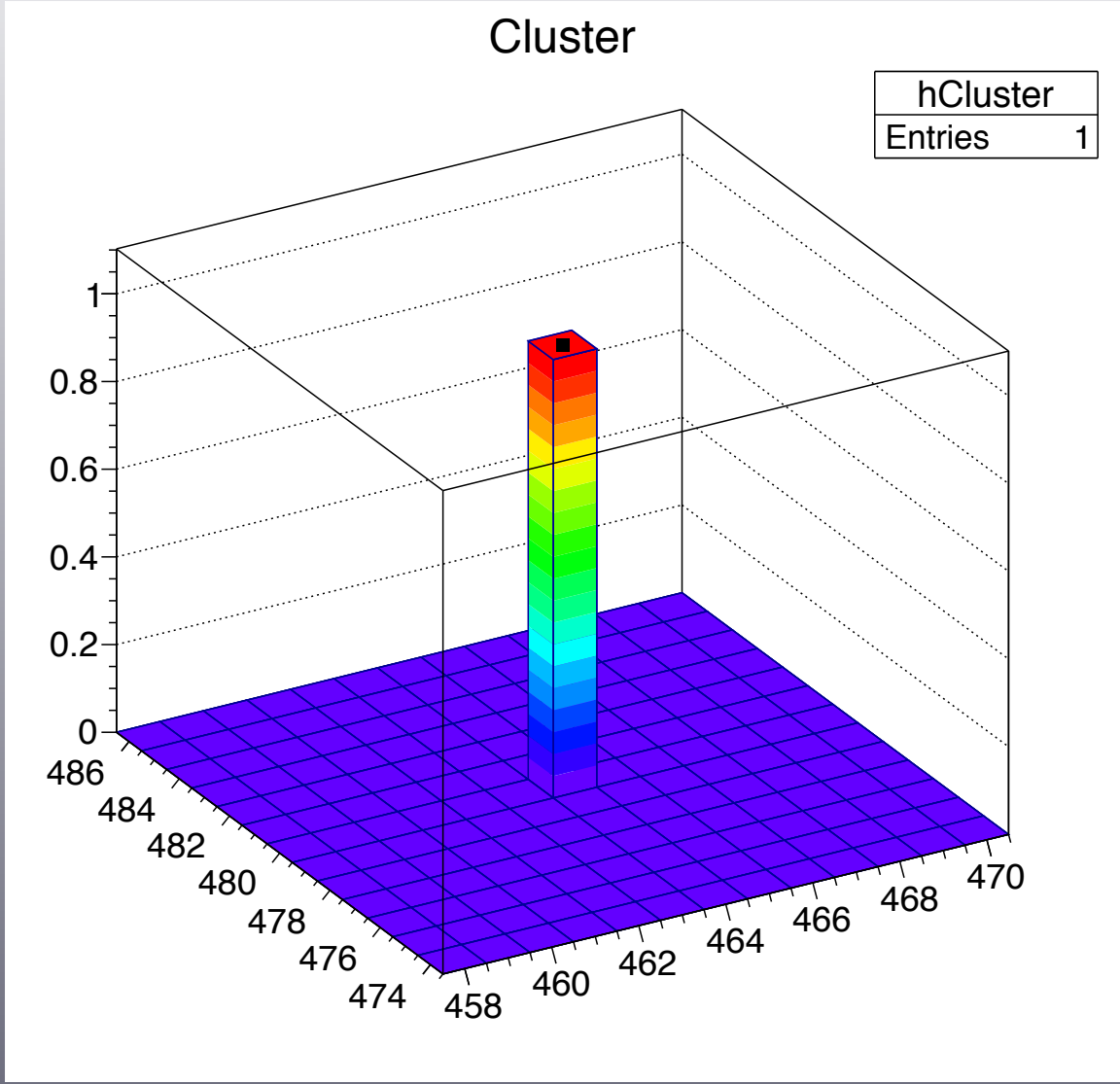
Radius Pixel distributions (i)

- Pixels distribution for configuration (0,0)



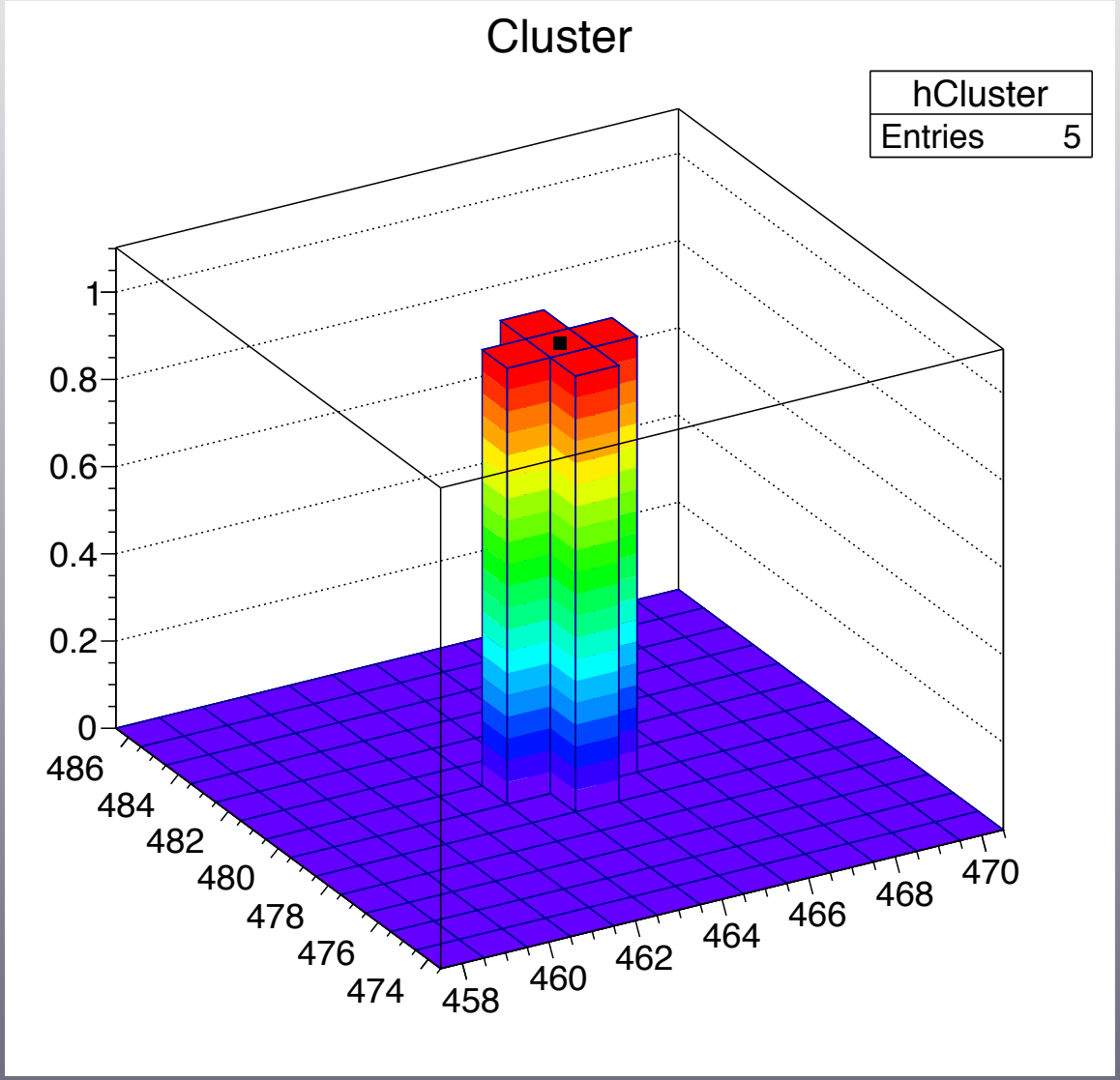
Radius Pixel distributions (ii)

- Pixels distribution for configuration (5,5)



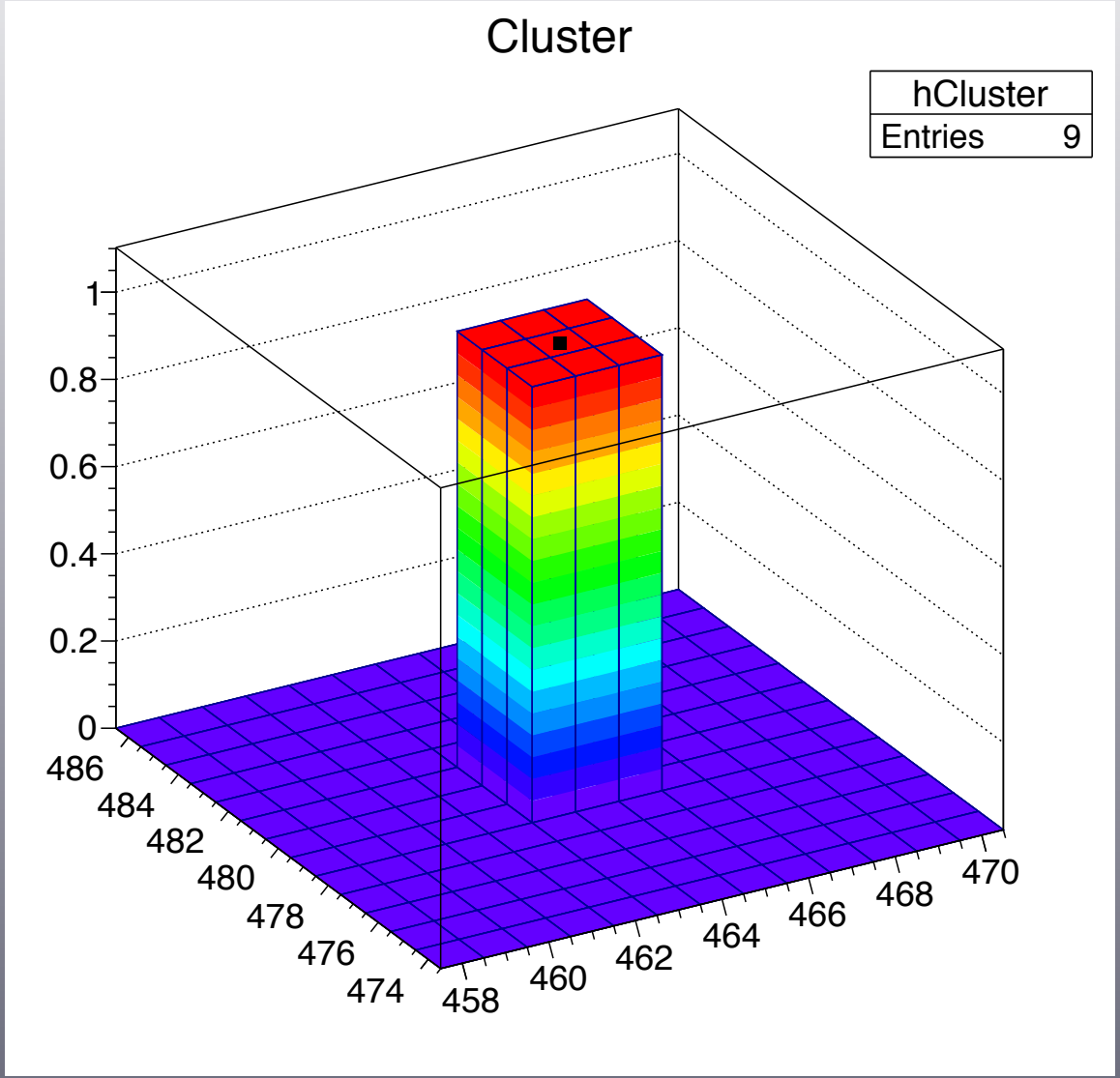
Radius Pixel distributions (ii)

- Pixels distribution for configuration (5,5)



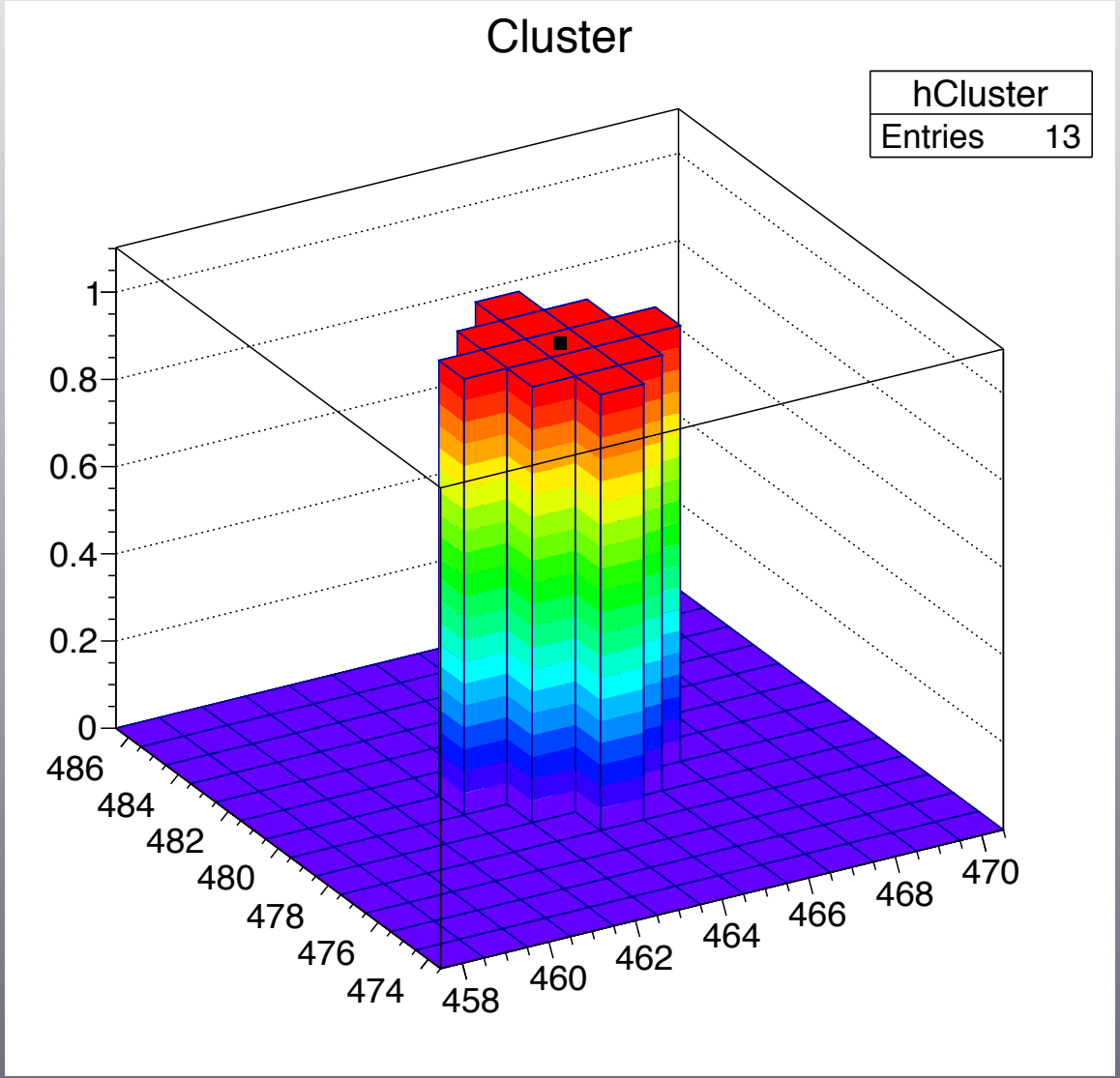
Radius Pixel distributions (ii)

- Pixels distribution for configuration (5,5)



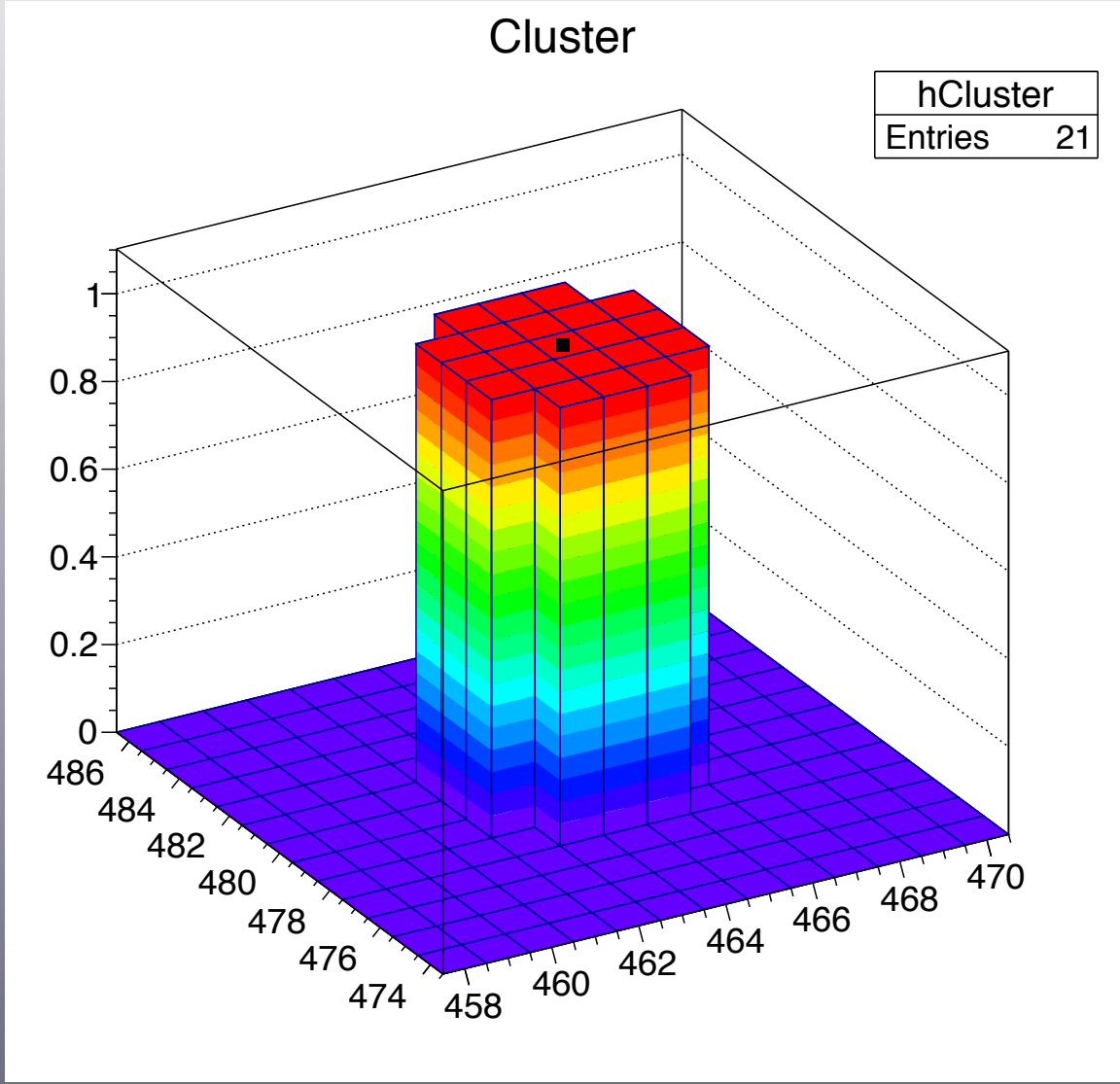
Radius Pixel distributions (ii)

- Pixels distribution for configuration (5,5)



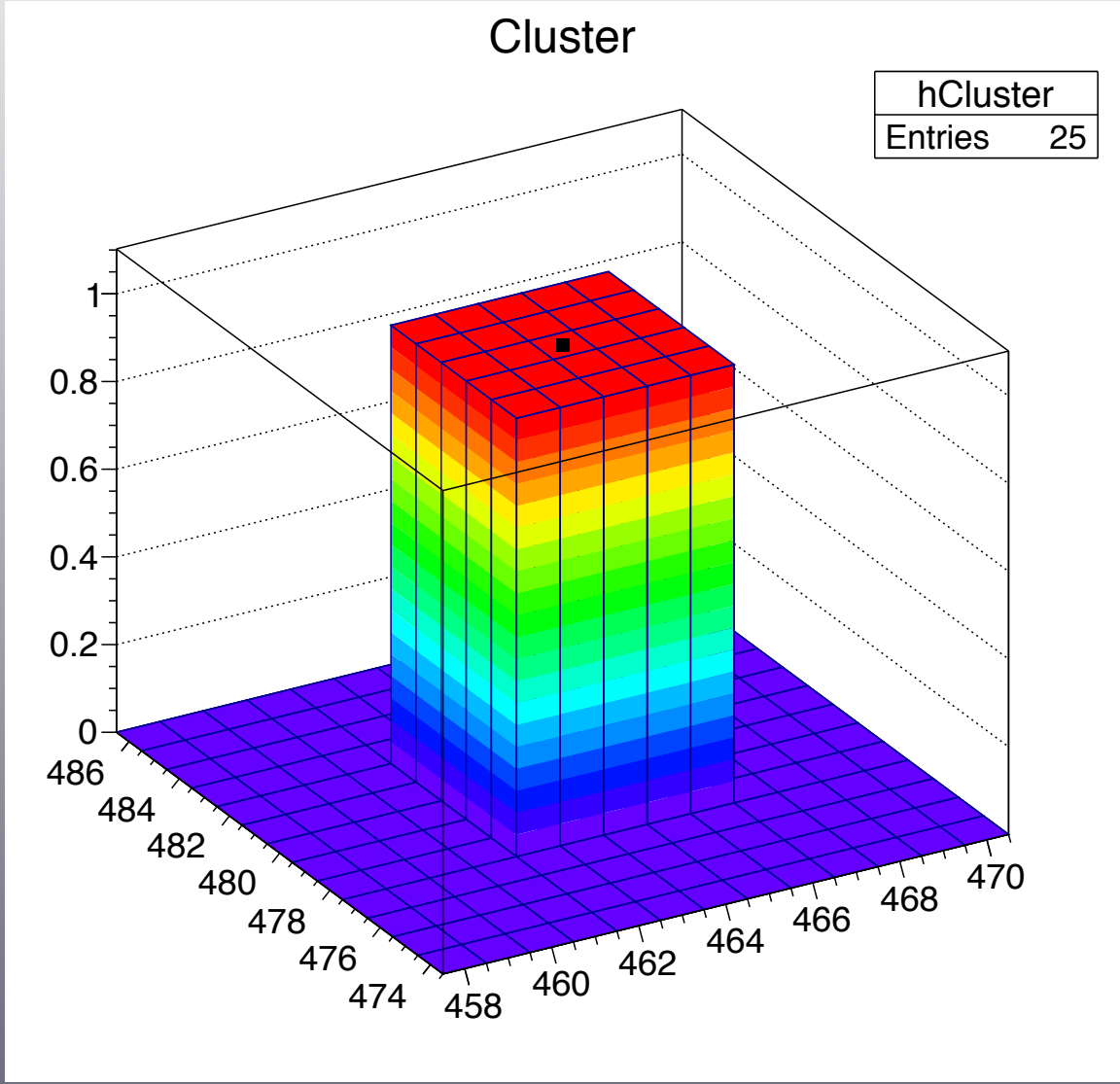
Radius Pixel distributions (ii)

- Pixels distribution for configuration (5,5)



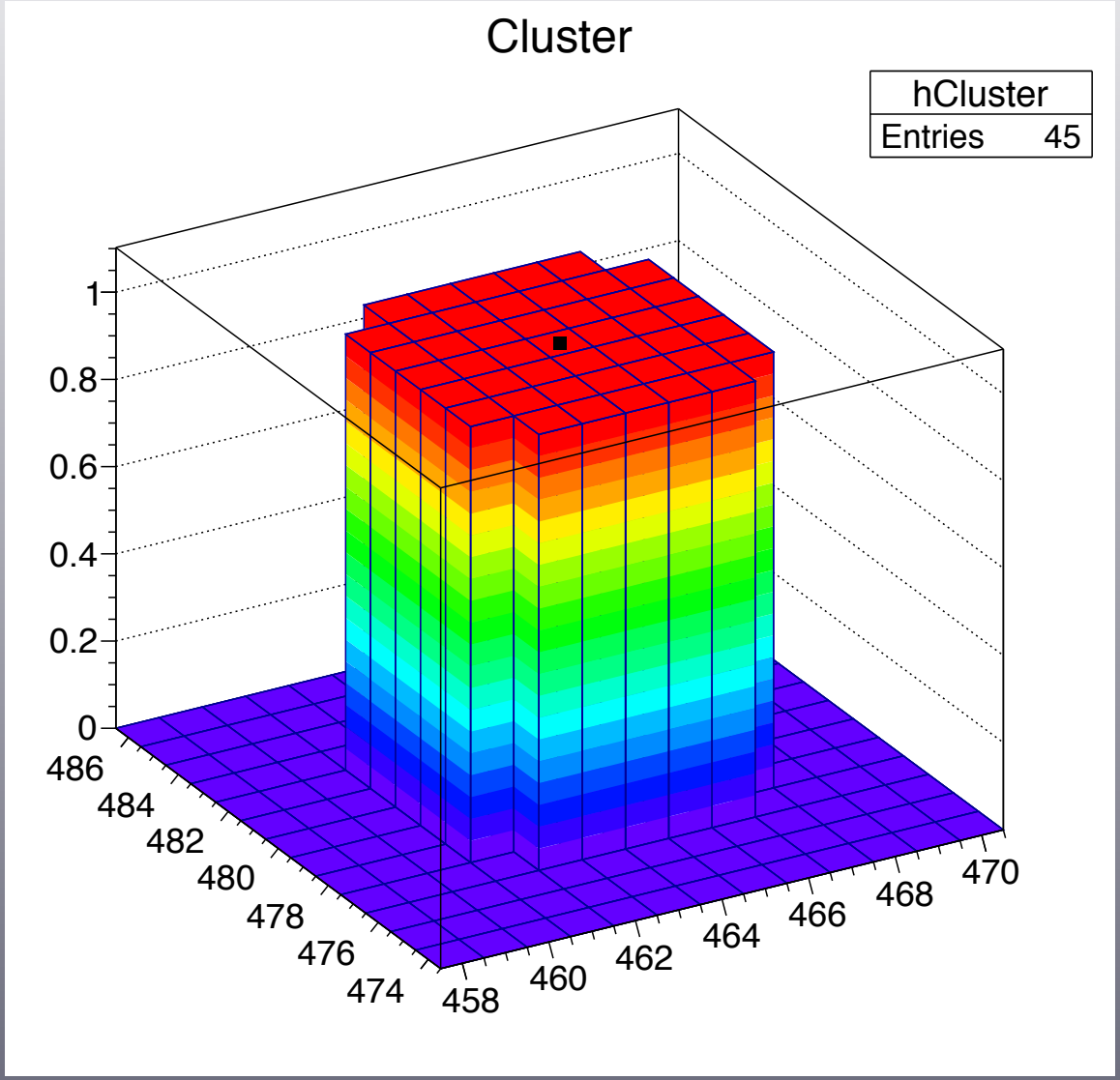
Radius Pixel distributions (ii)

- Pixels distribution for configuration (5,5)



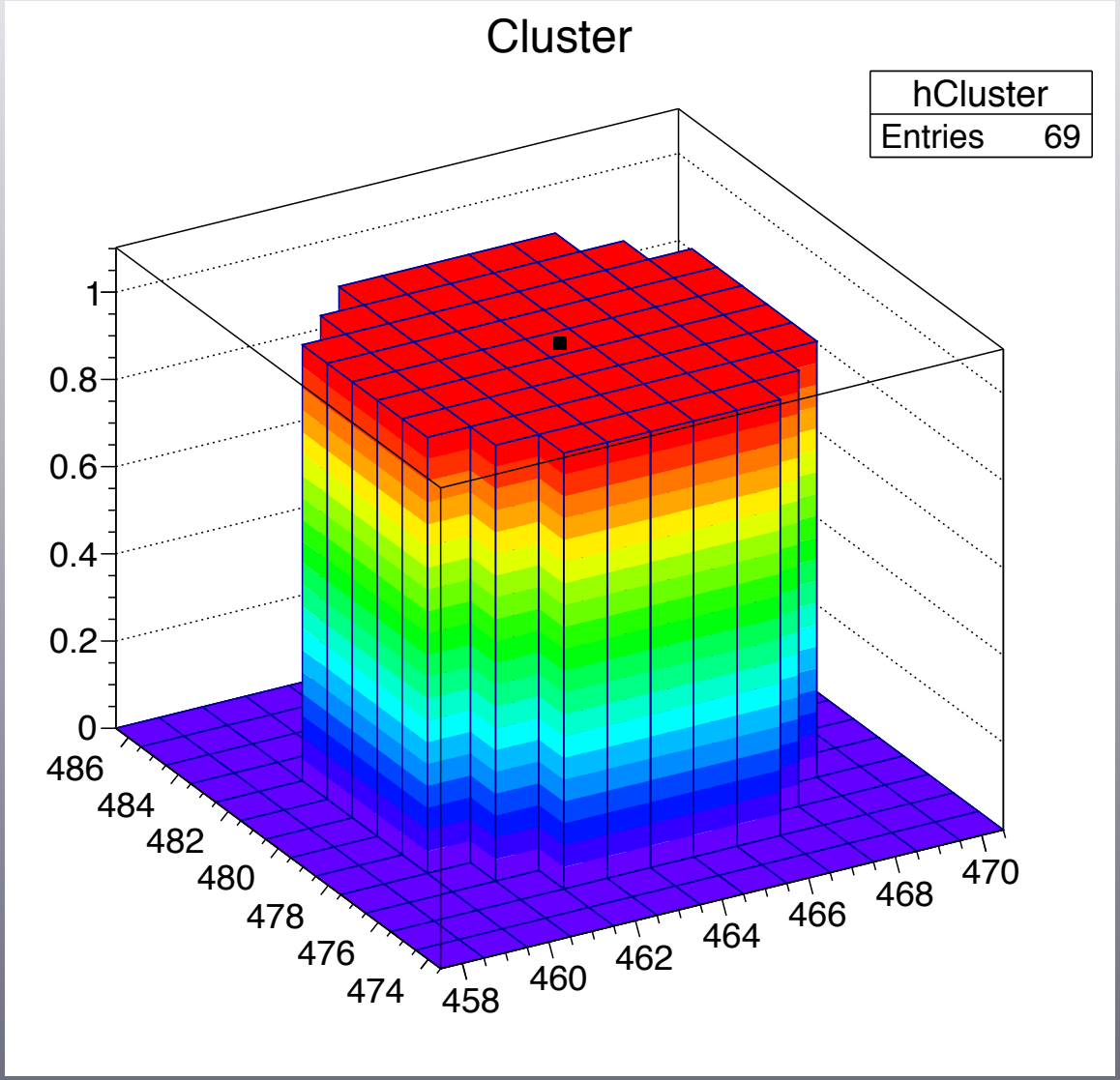
Radius Pixel distributions (ii)

- Pixels distribution for configuration (5,5)



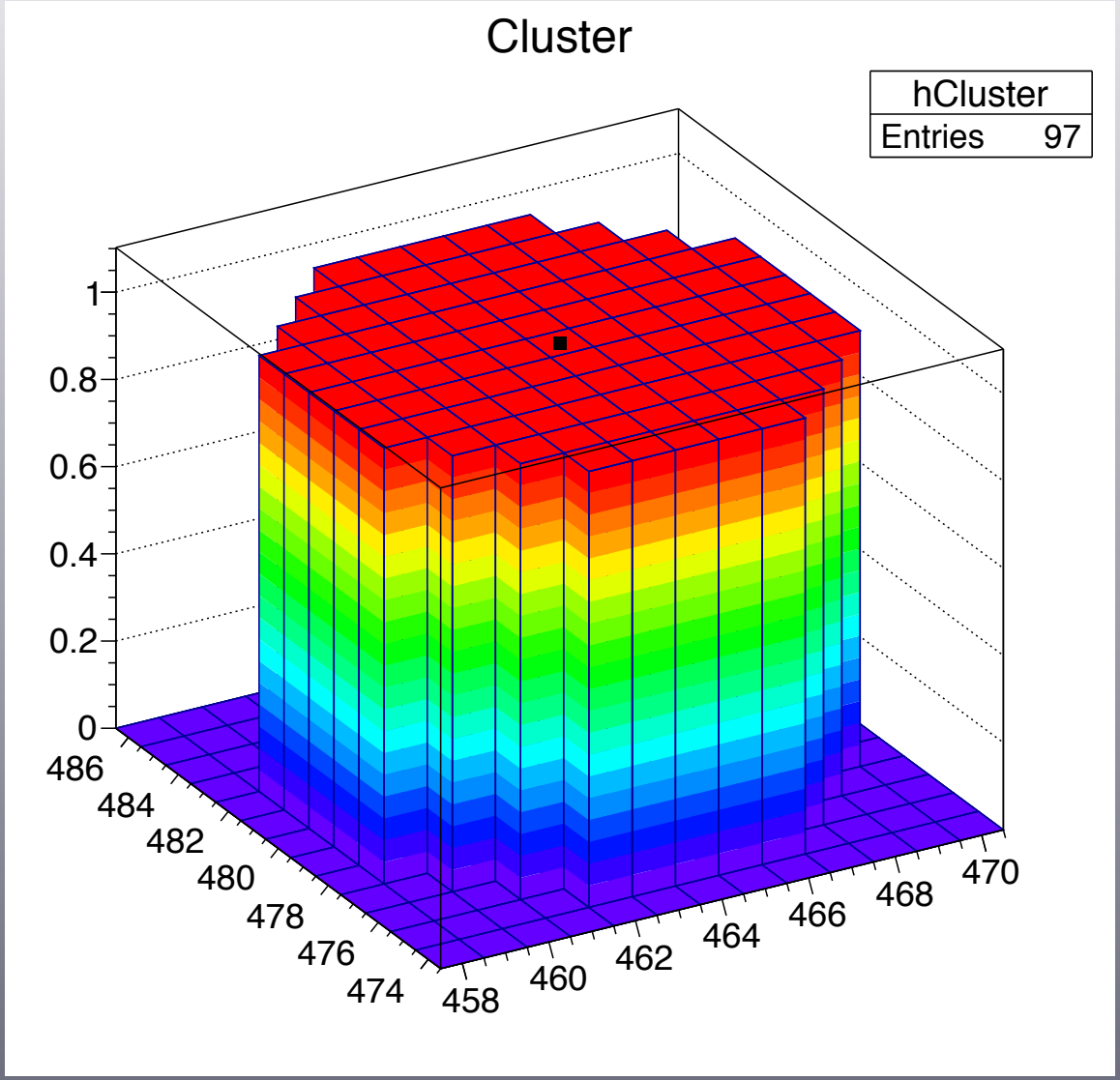
Radius Pixel distributions (ii)

- Pixels distribution for configuration (5,5)



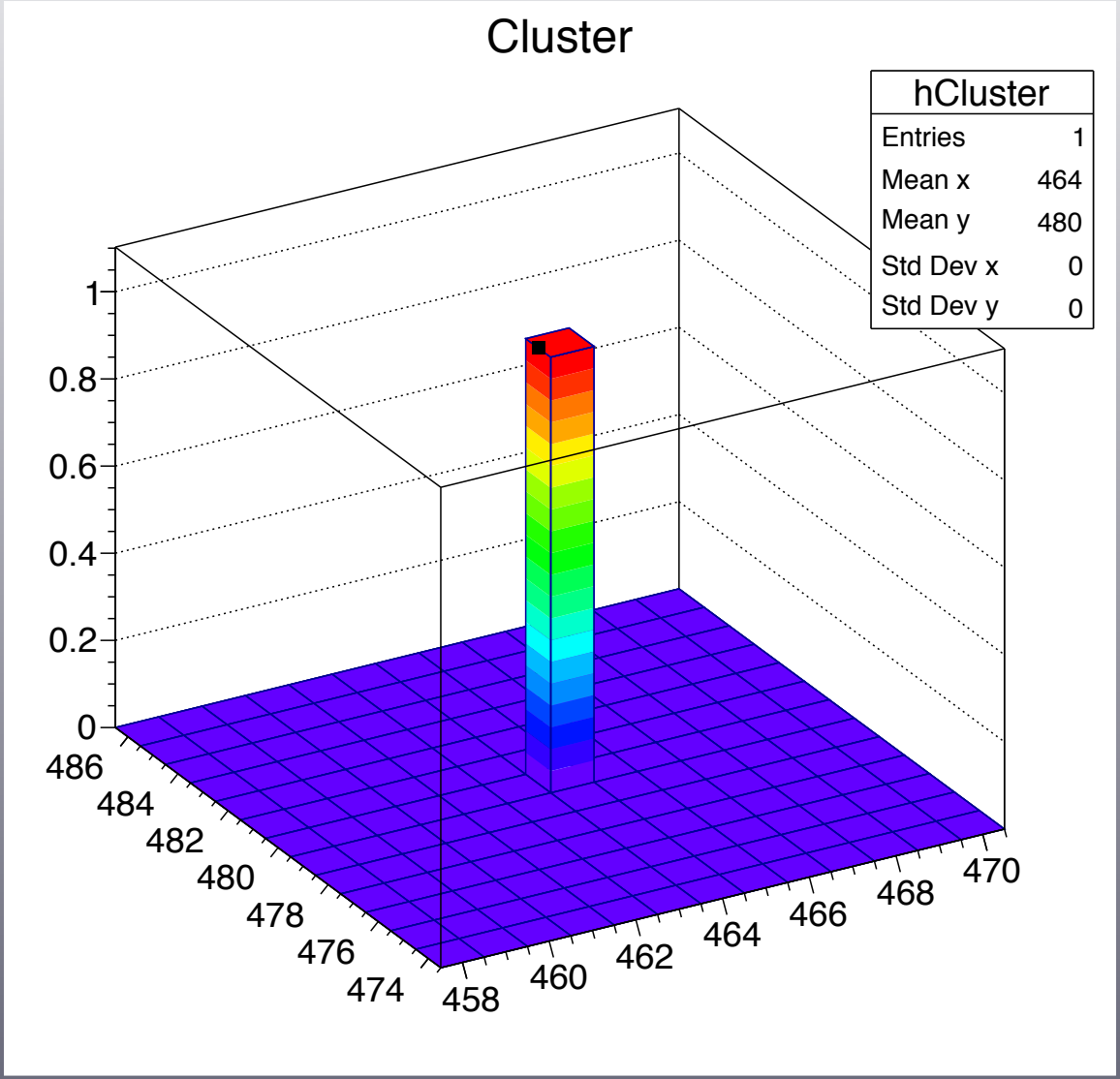
Radius Pixel distributions (ii)

- Pixels distribution for configuration (5,5)



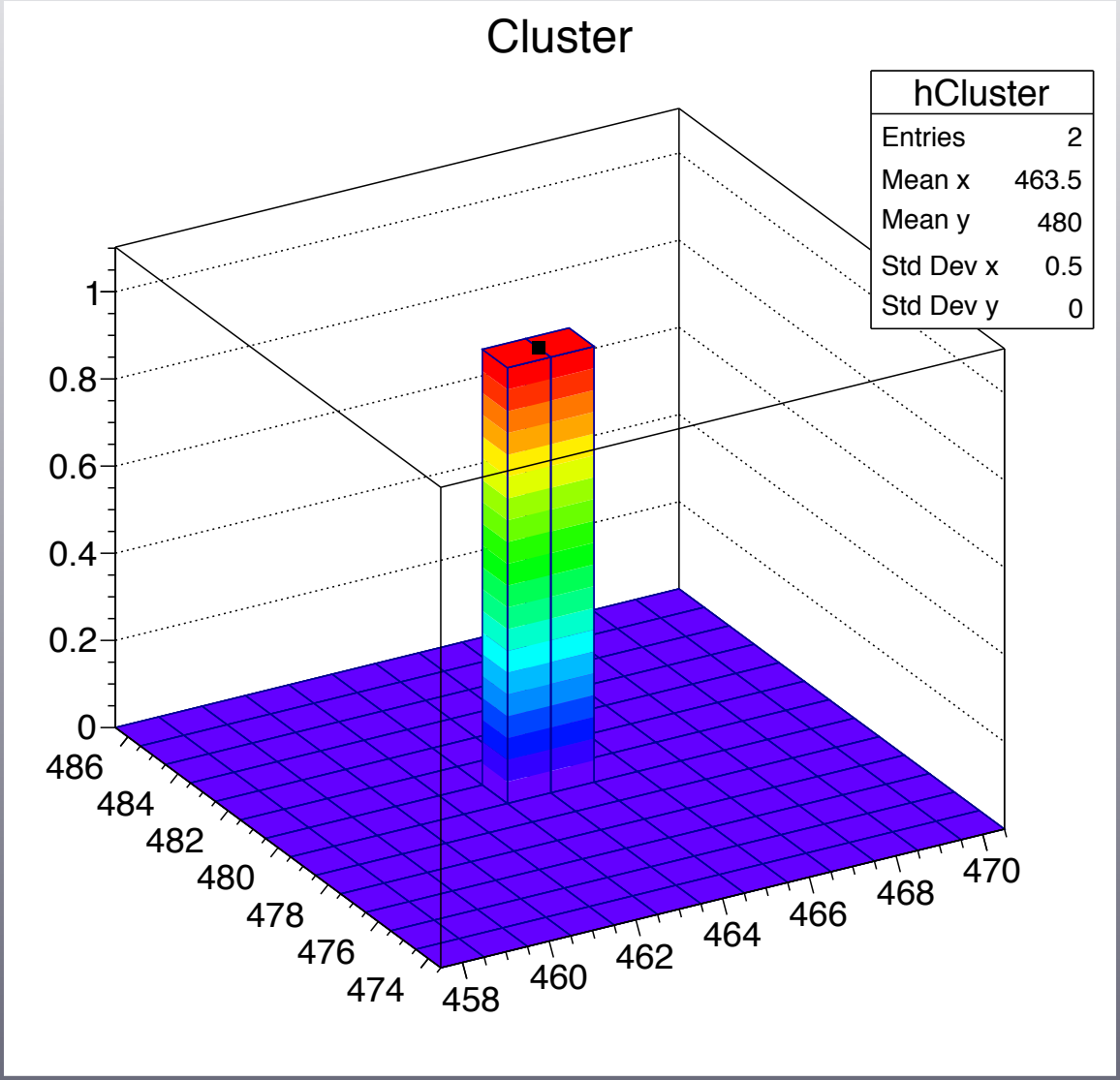
Radius Pixel distributions (iii)

- Pixels distribution for configuration (5,0)



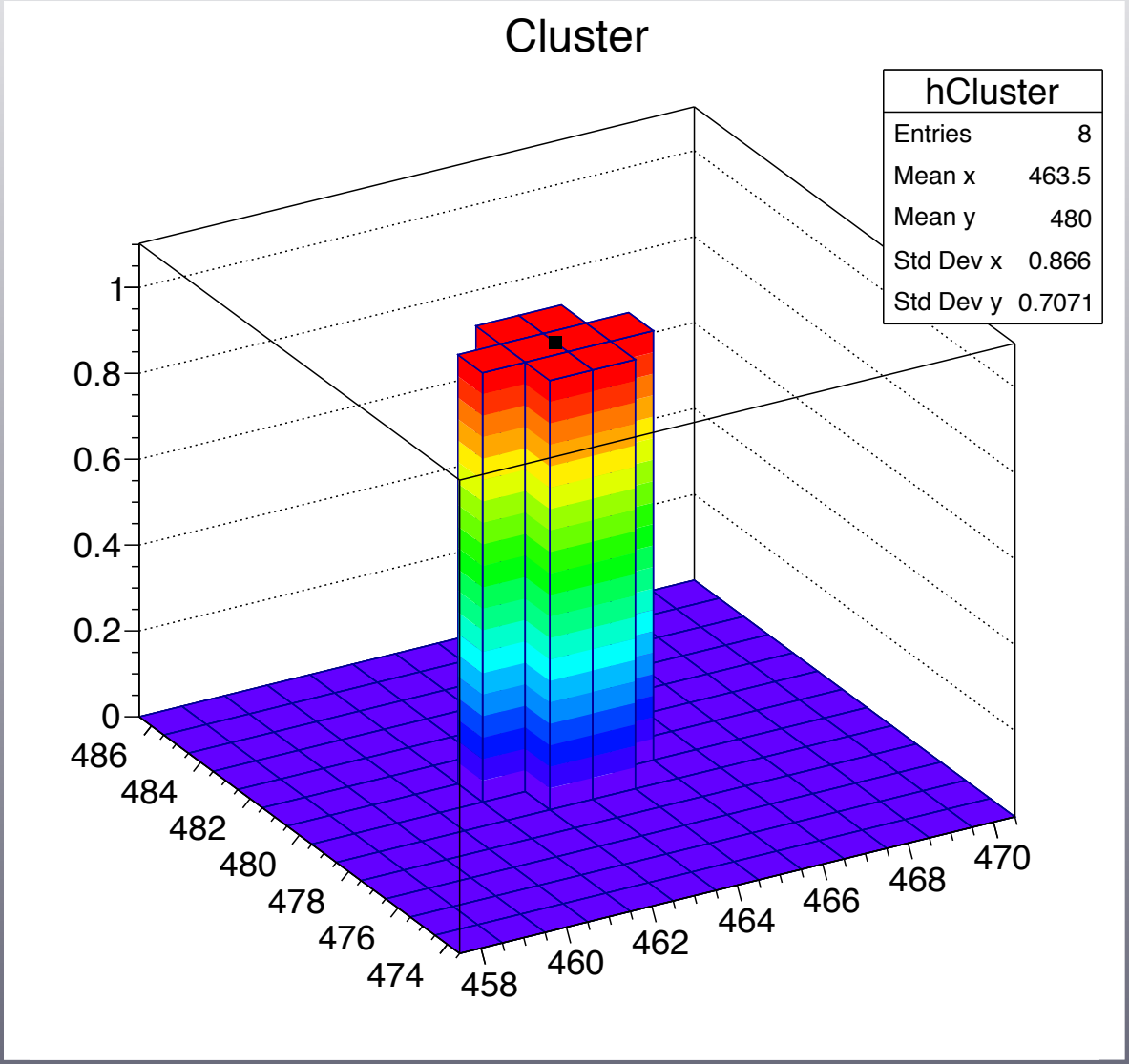
Radius Pixel distributions (iii)

- Pixels distribution for configuration (5,0)



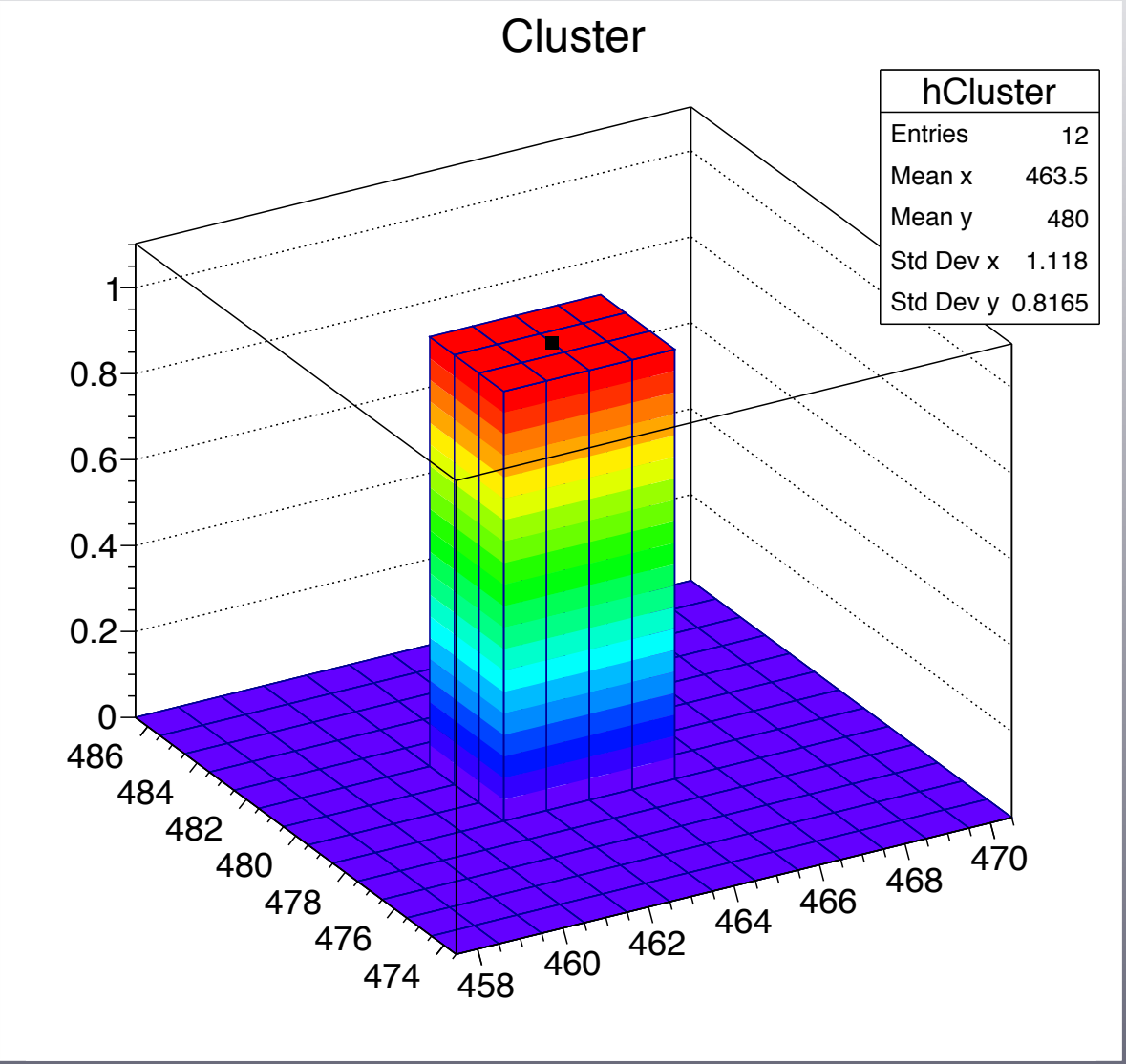
Radius Pixel distributions (iii)

- Pixels distribution for configuration (5,0)



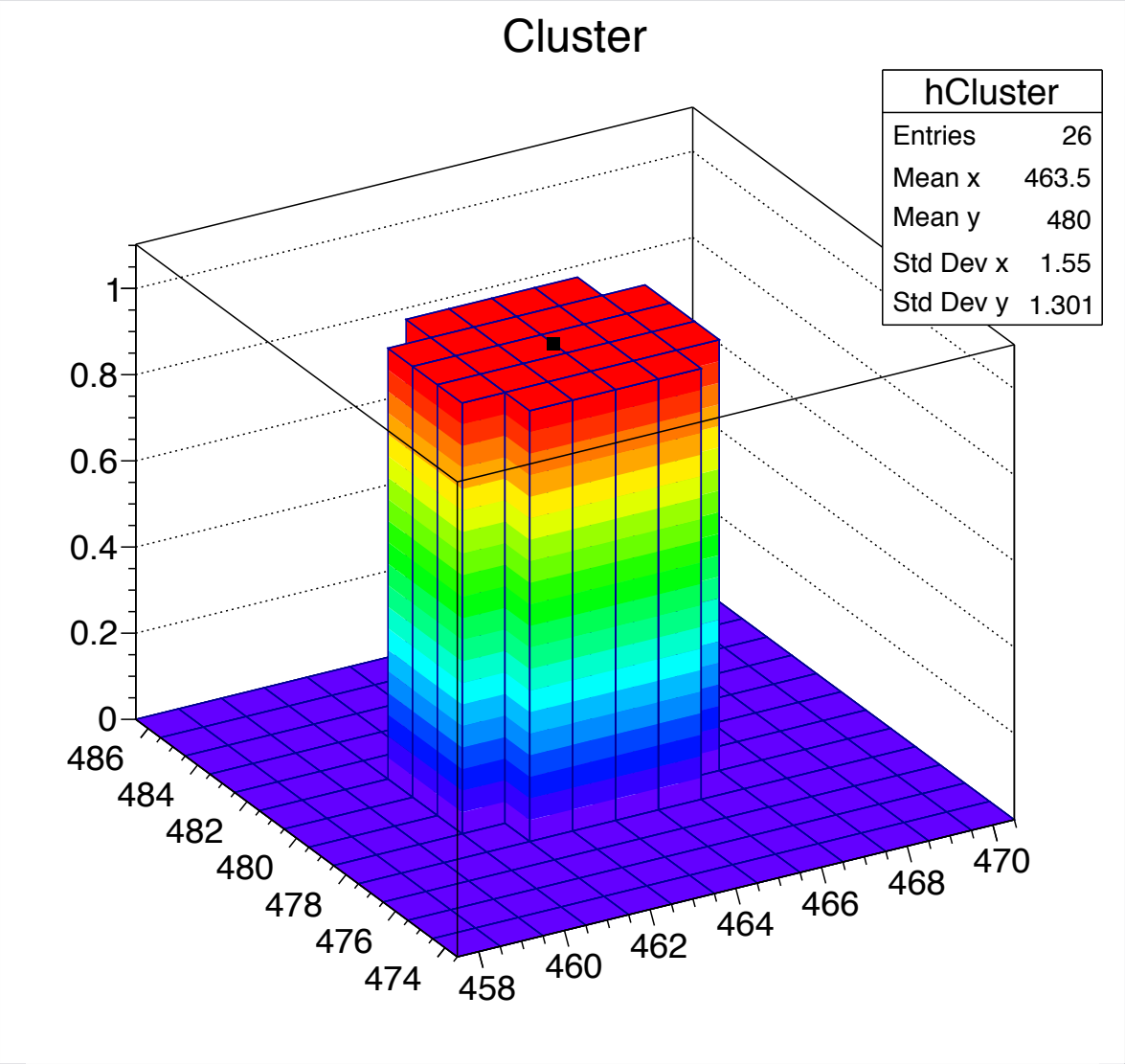
Radius Pixel distributions (iii)

- Pixels distribution for configuration (5,0)



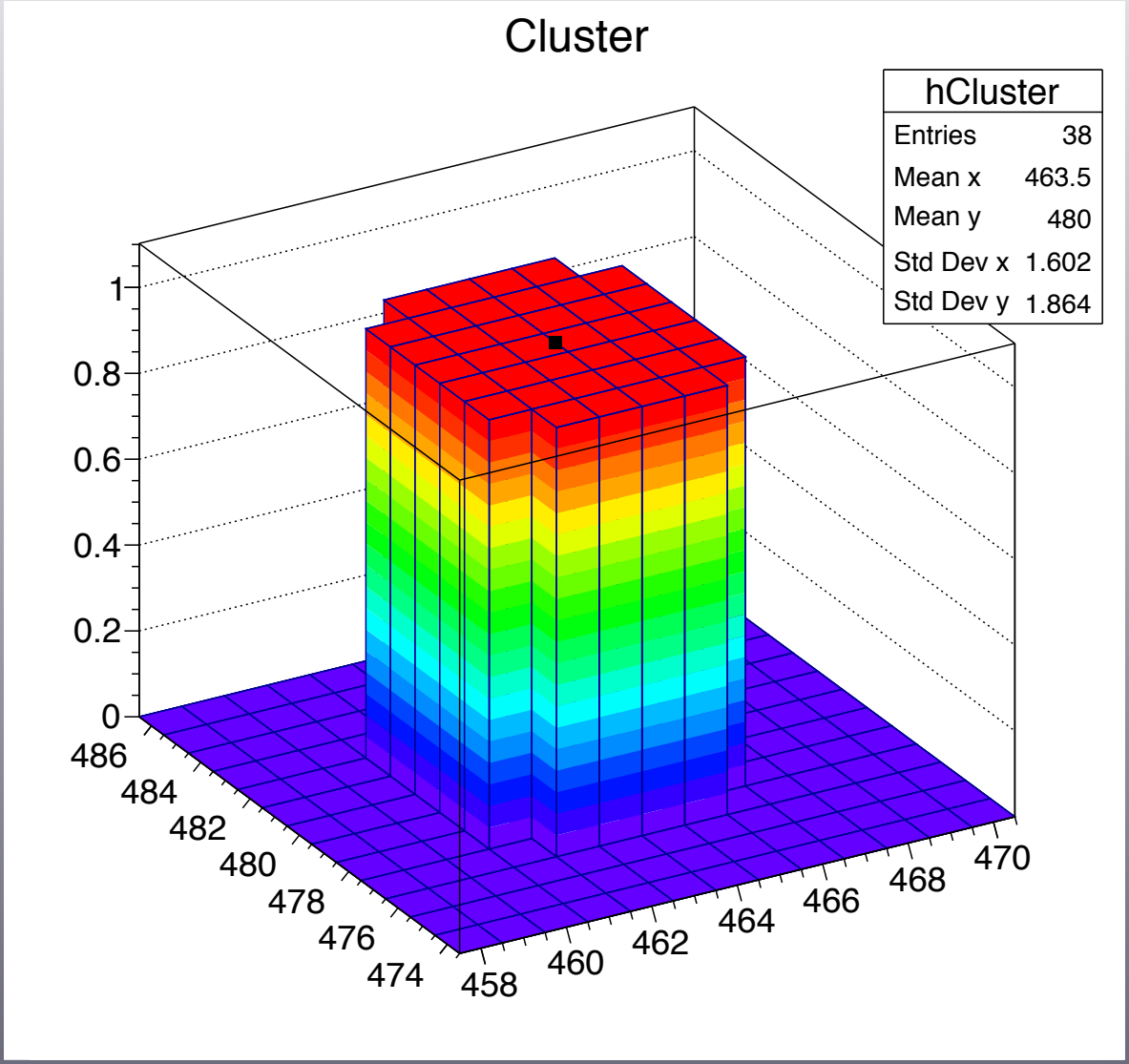
Radius Pixel distributions (iii)

- Pixels distribution for configuration (5,0)



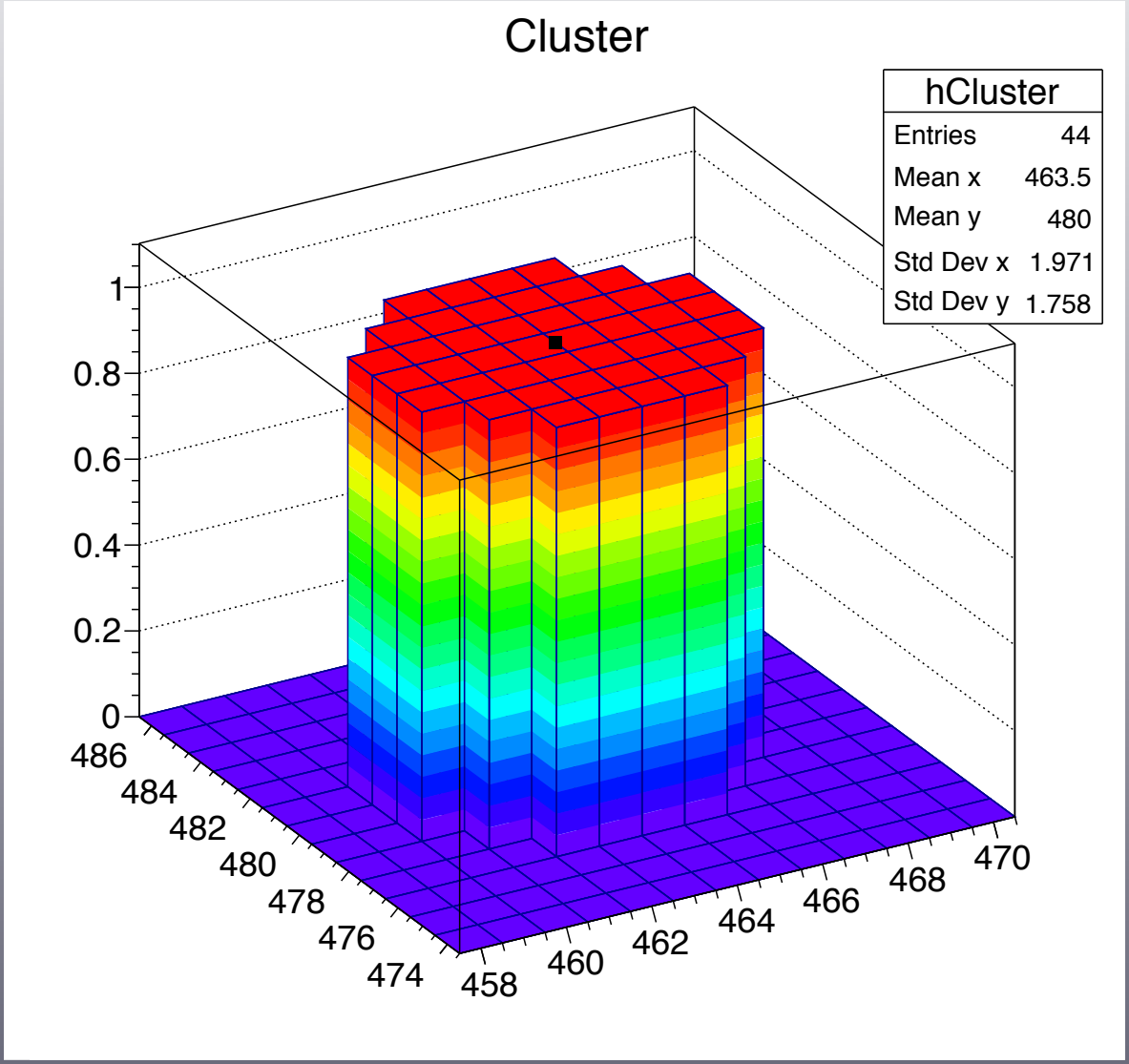
Radius Pixel distributions (iii)

- Pixels distribution for configuration (5,0)



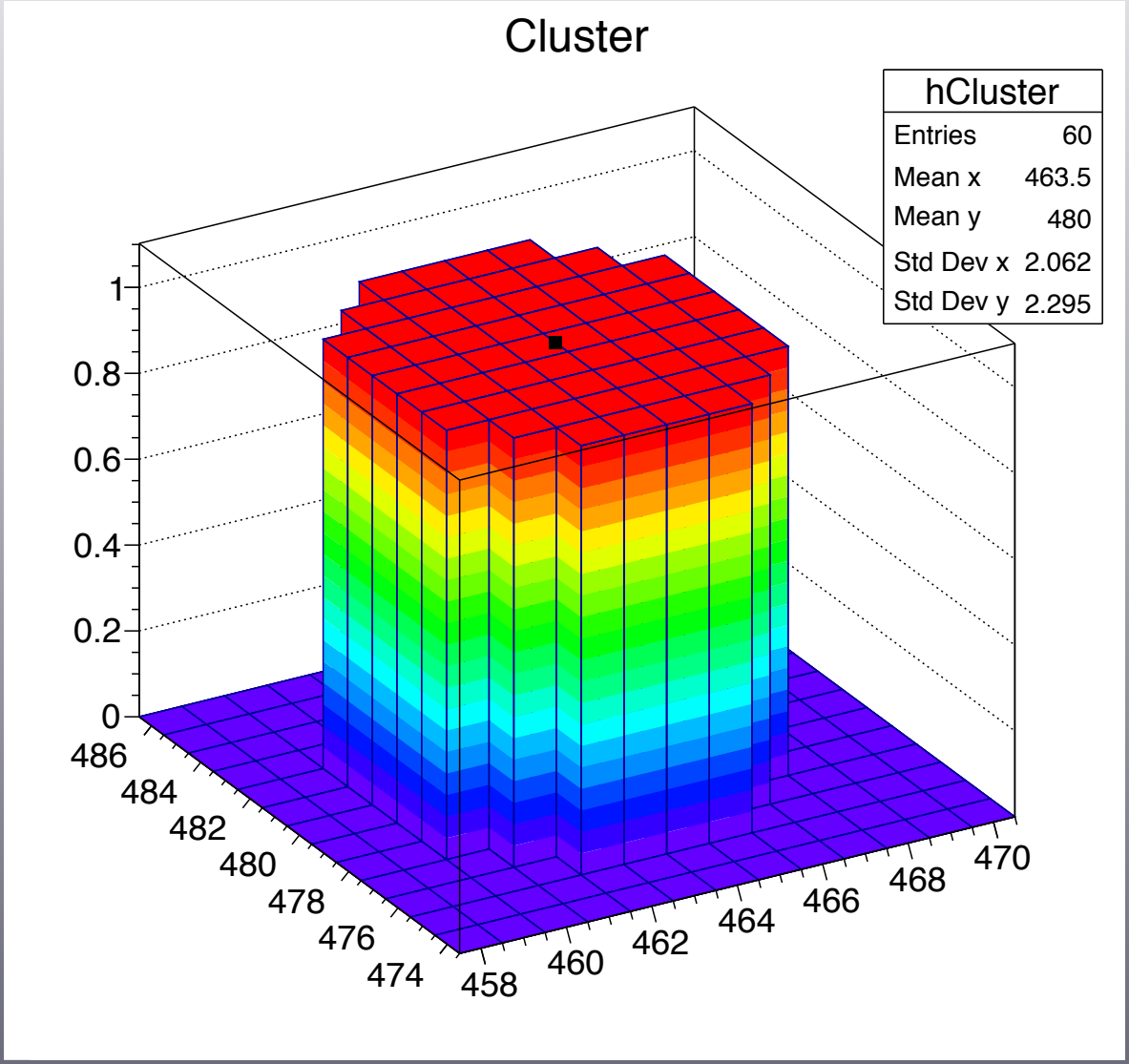
Radius Pixel distributions (iii)

- Pixels distribution for configuration (5,0)



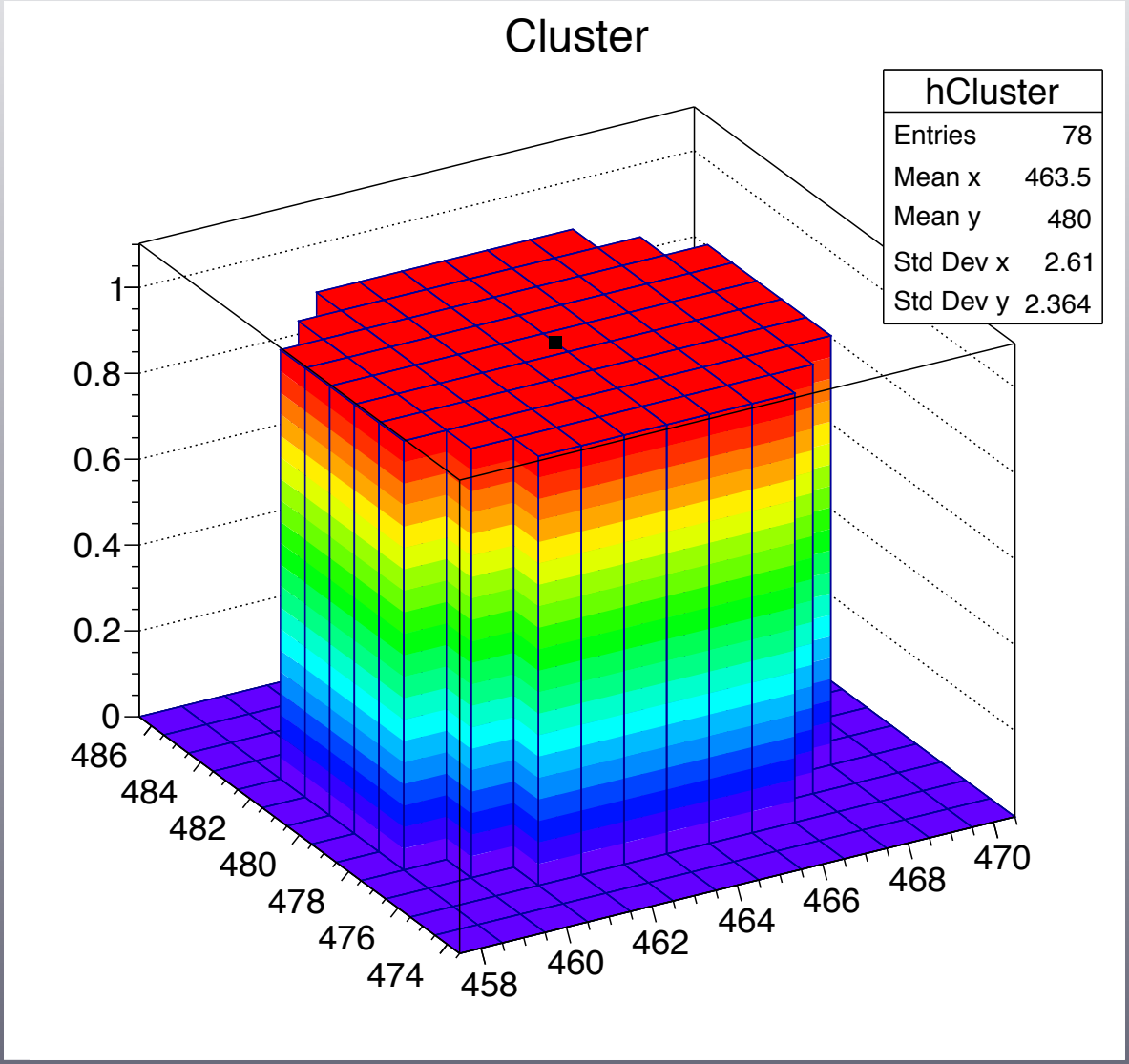
Radius Pixel distributions (iii)

- Pixels distribution for configuration (5,0)



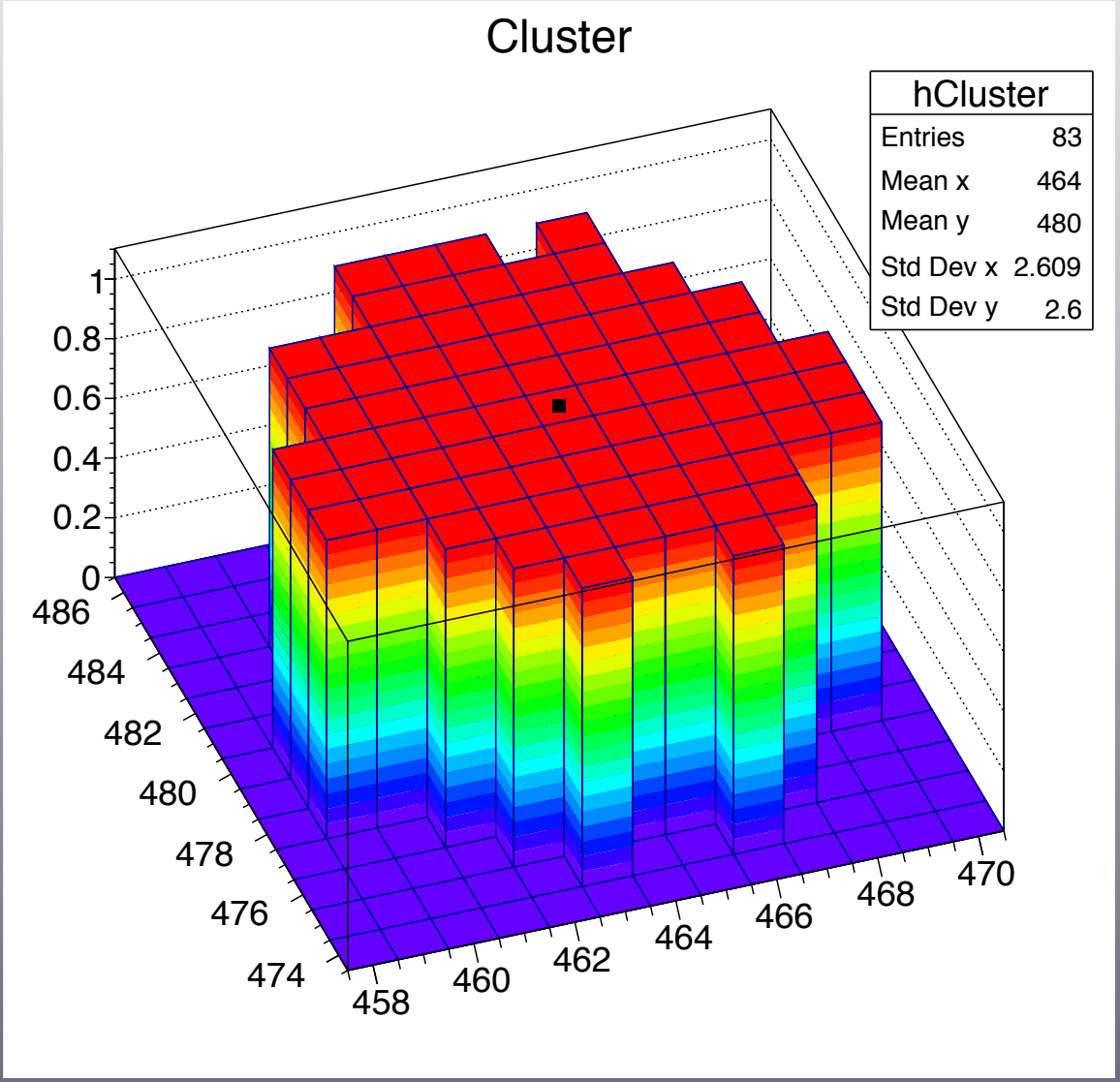
Radius Pixel distributions (iii)

- Pixels distribution for configuration (5,0)



Radius Pixel distributions (iii)

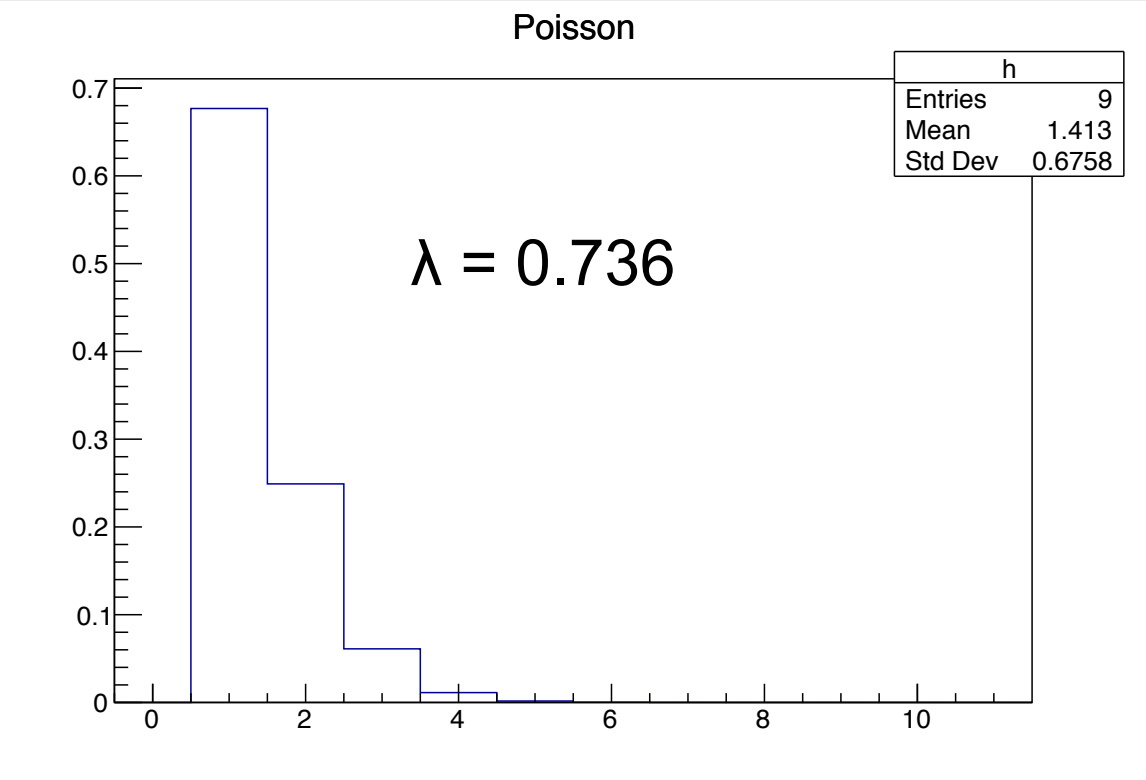
- Last incomplete “shell”, pixels are randomly draw



Pileup events (i)

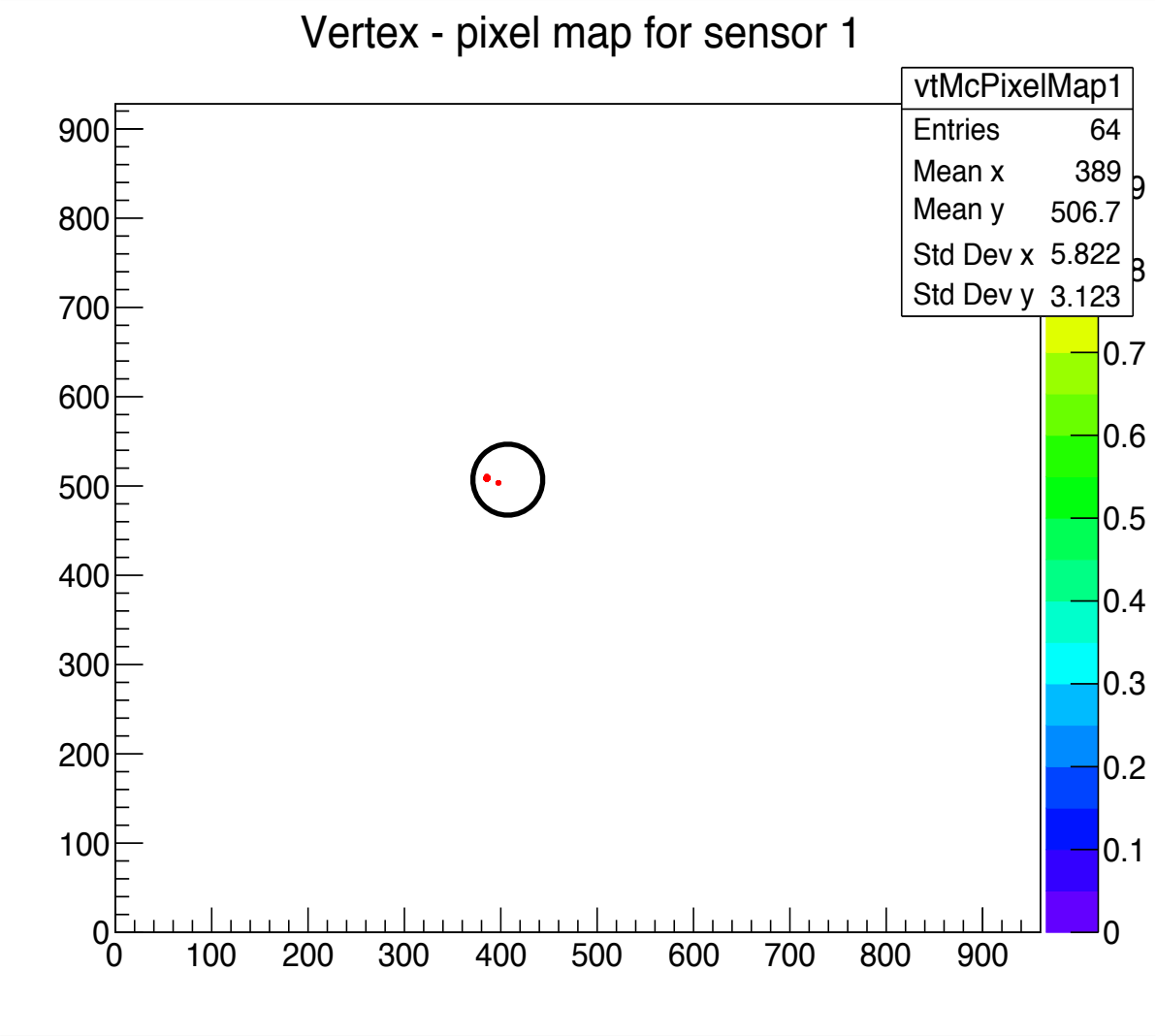
- Due to the integration time of M28 (~200 µs), probability to pileup

  - Method:
    - Store N events in a list
    - Compute pileup probability n from Poisson law: $p(n) = \frac{\lambda^n}{n!} e^{-n}$
    - Mixed n events out of N stored



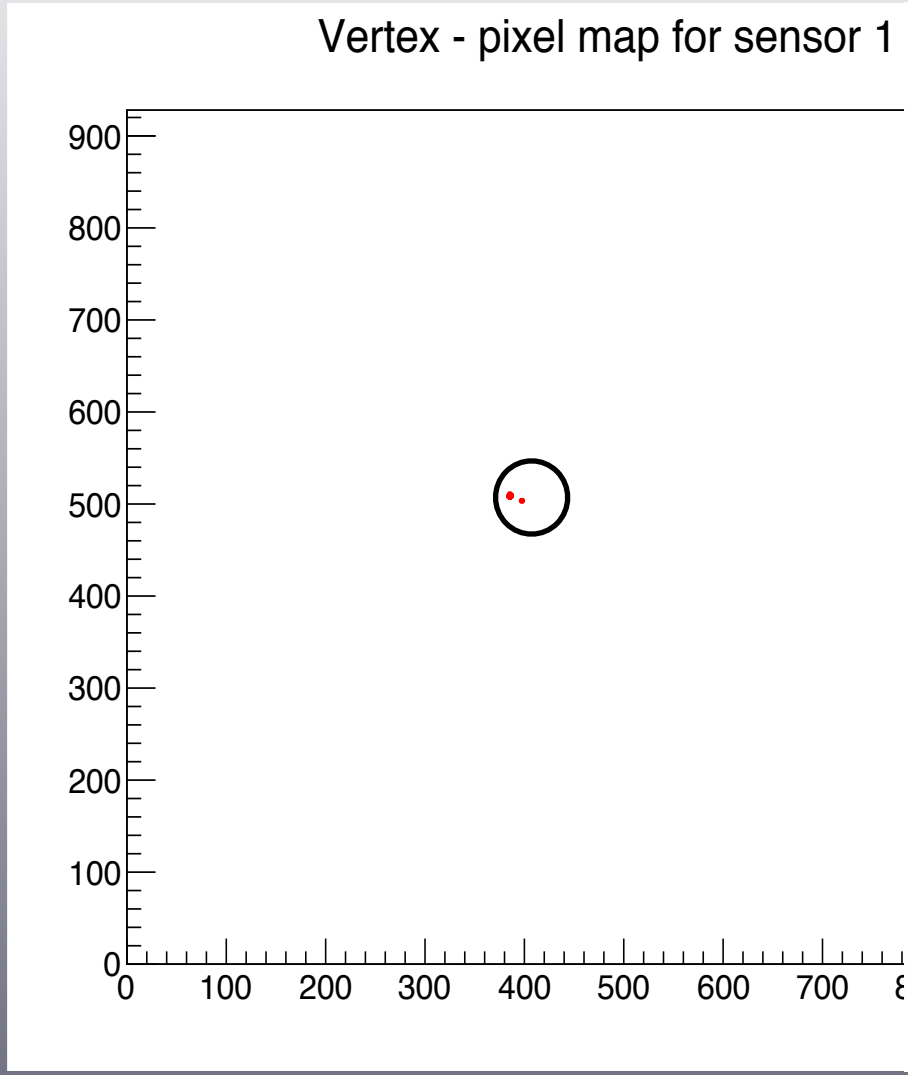
Pileup events (ii)

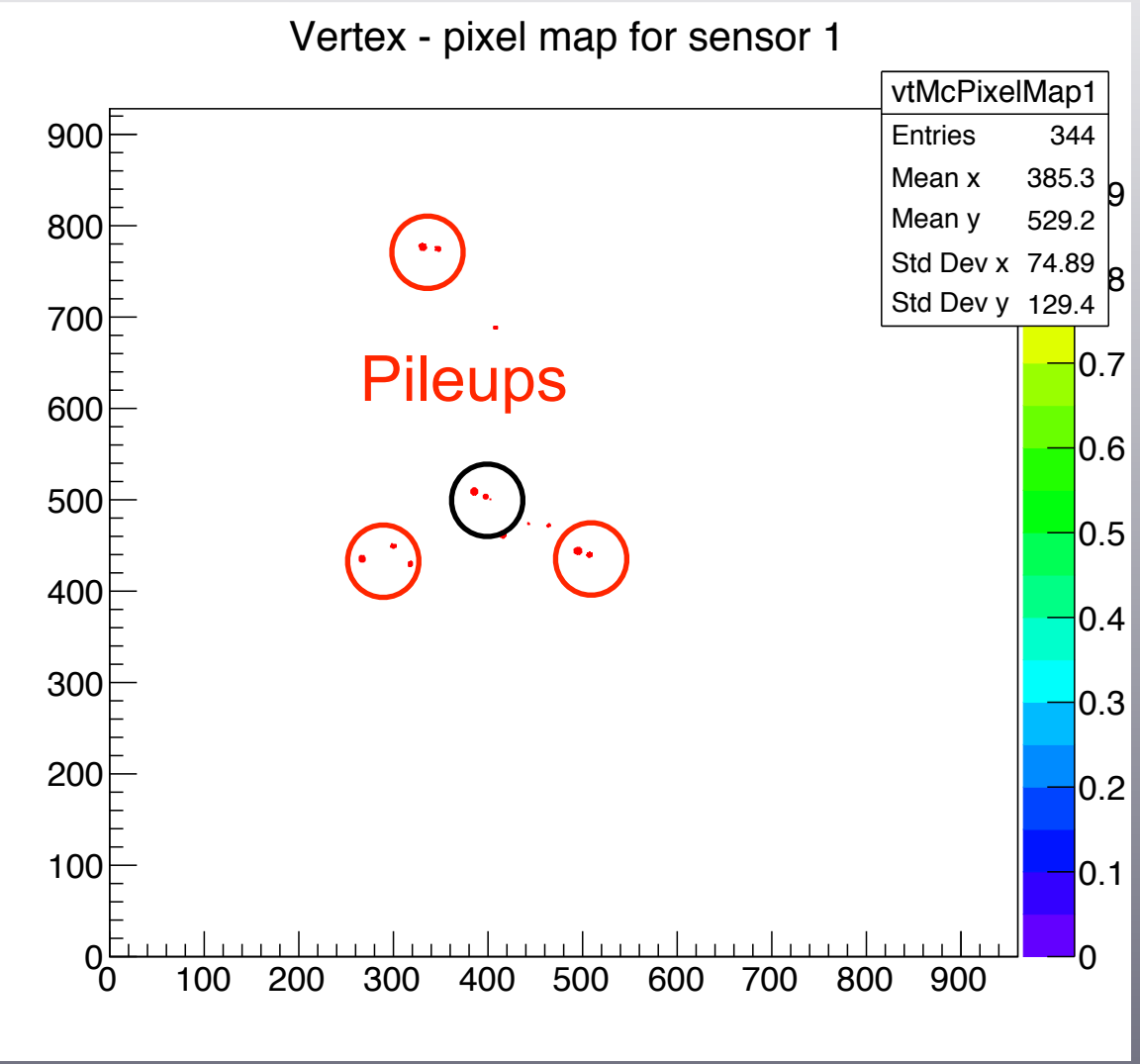
- Event + 3 pileup events



Pileup events (ii)

- Event + 3 pileup events





Conclusions

- Digitizer works
  - Generates clusters
  - Provides pileup events
  - ➡ Interface with the whole procedure