Simulating Hexagonal Pixel Cells in Allpix Squared

Ryuji Moriya, CERN Summer Student, University of Glasgow

Supervisor: Katharina Dort, Magdalena Munker

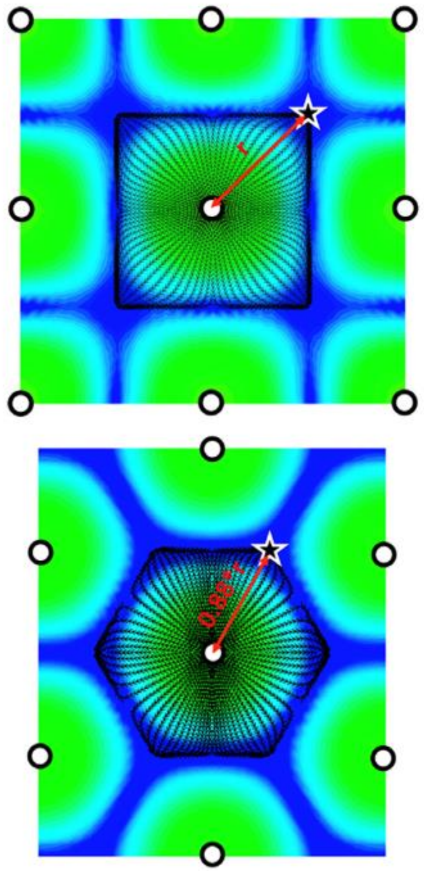
2nd Allpix Squared User Workshop

Outline

- Introduction to Hexagonal Pixels
- Implementation of Hexagonal Pixel Geometry into Allpix Squared
- First simulation with MIMOSA26
- Summary and Outlook

Introduction to Hexagonal Pixels

- Current semiconductor detectors overwhelmingly use rectangular/square pixel design
- Hexagonal pixel design offers multitude of benefits
  - Smaller distance edge region (same area) → reduced charge collection time → improved time resolution
  - Less charge sharing → better detector efficiency
  - More homogeneous response in pixel → more precise timing resolution
  - Fastpix



T. Kugathasan, et al.: https://doi.org/10.1016/j.nima.2020.164461

Implementation of Hexagonal Pixel Geometry

- Current Allpix Squared detector models only compatible with rectangular/square pixel geometry
- A need for hexagonal pixel geometry $\Longrightarrow$ new detector class

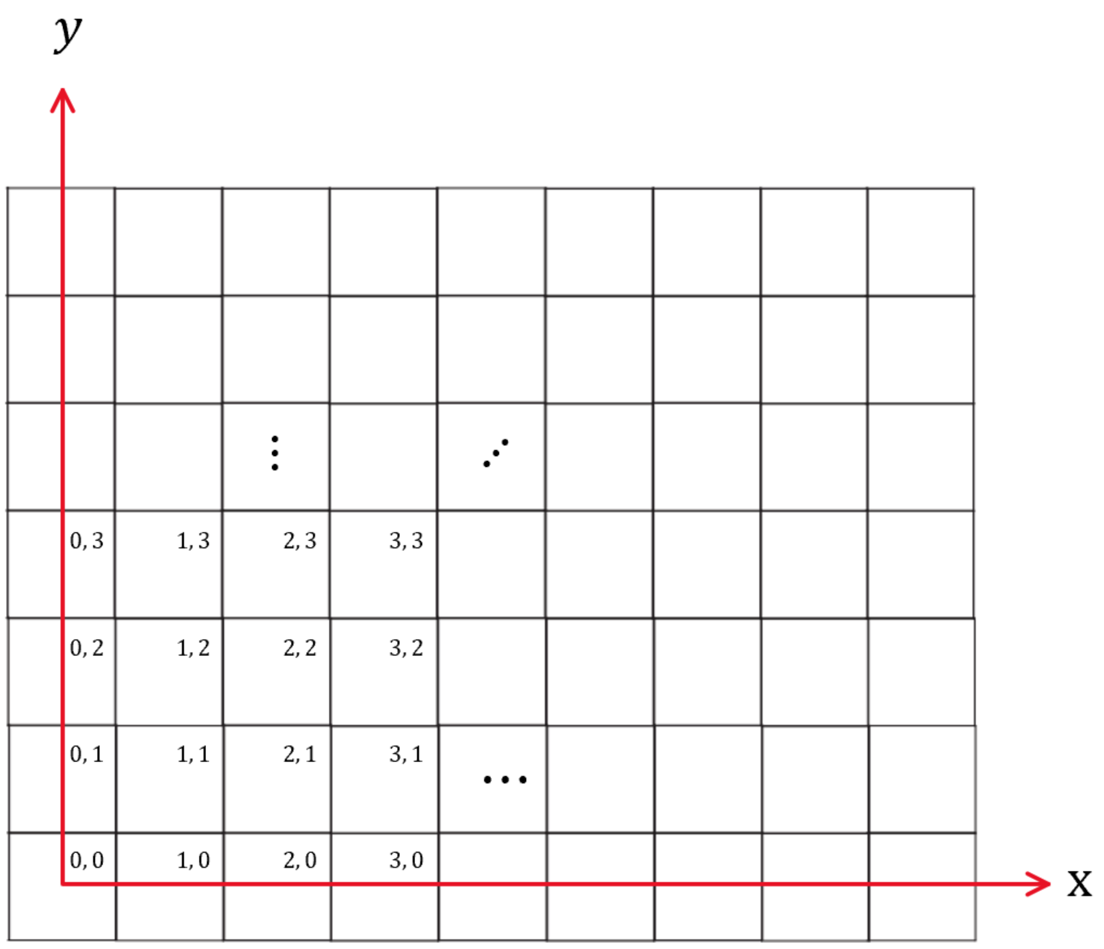
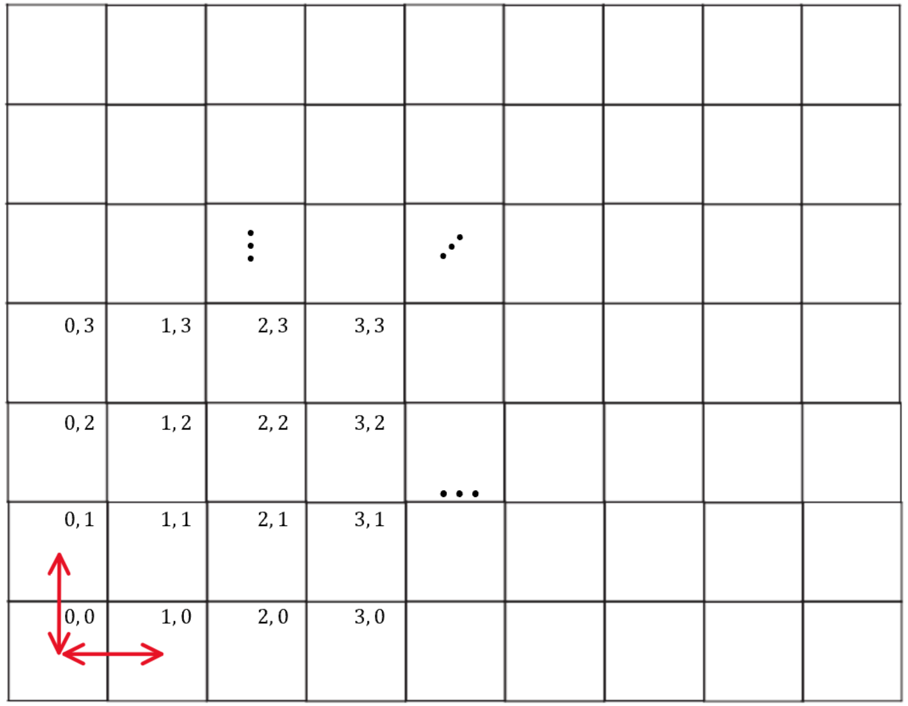


Figure: the rectangular/square pixel grid

Implementation of Hexagonal Pixel Geometry

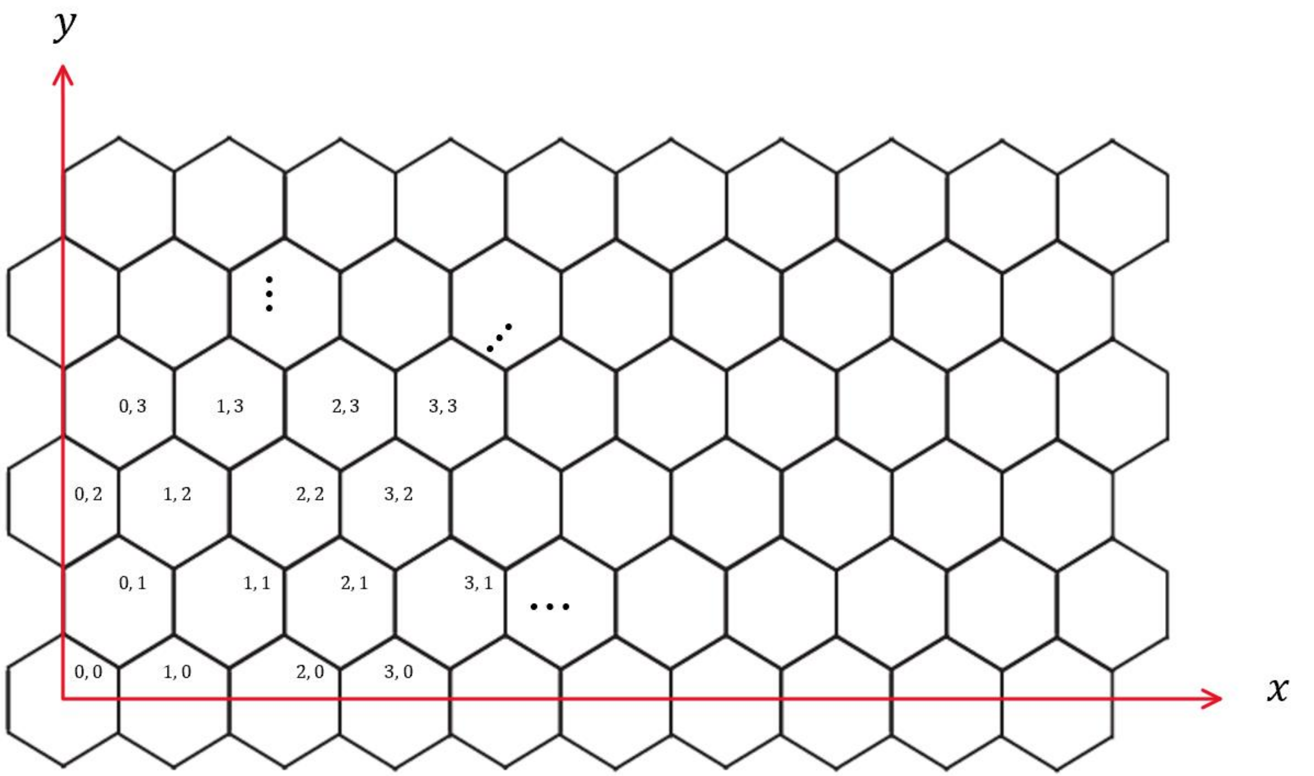
```
std::pair<int, int> DetectorModel::getPixelIndex(const ROOT::Math::XYZPoint& position) const {
    auto pixel_x = static_cast<int>(std::round(position.x() / pixel_size_.x()));
    auto pixel_y = static_cast<int>(std::round(position.y() / pixel_size_.y()));
    return {pixel_x, pixel_y};
}
```

- For a rectangular/square pixel geometry, the calculation of the pixel index for a given position is simple…



Implementation of Hexagonal Pixel Geometry

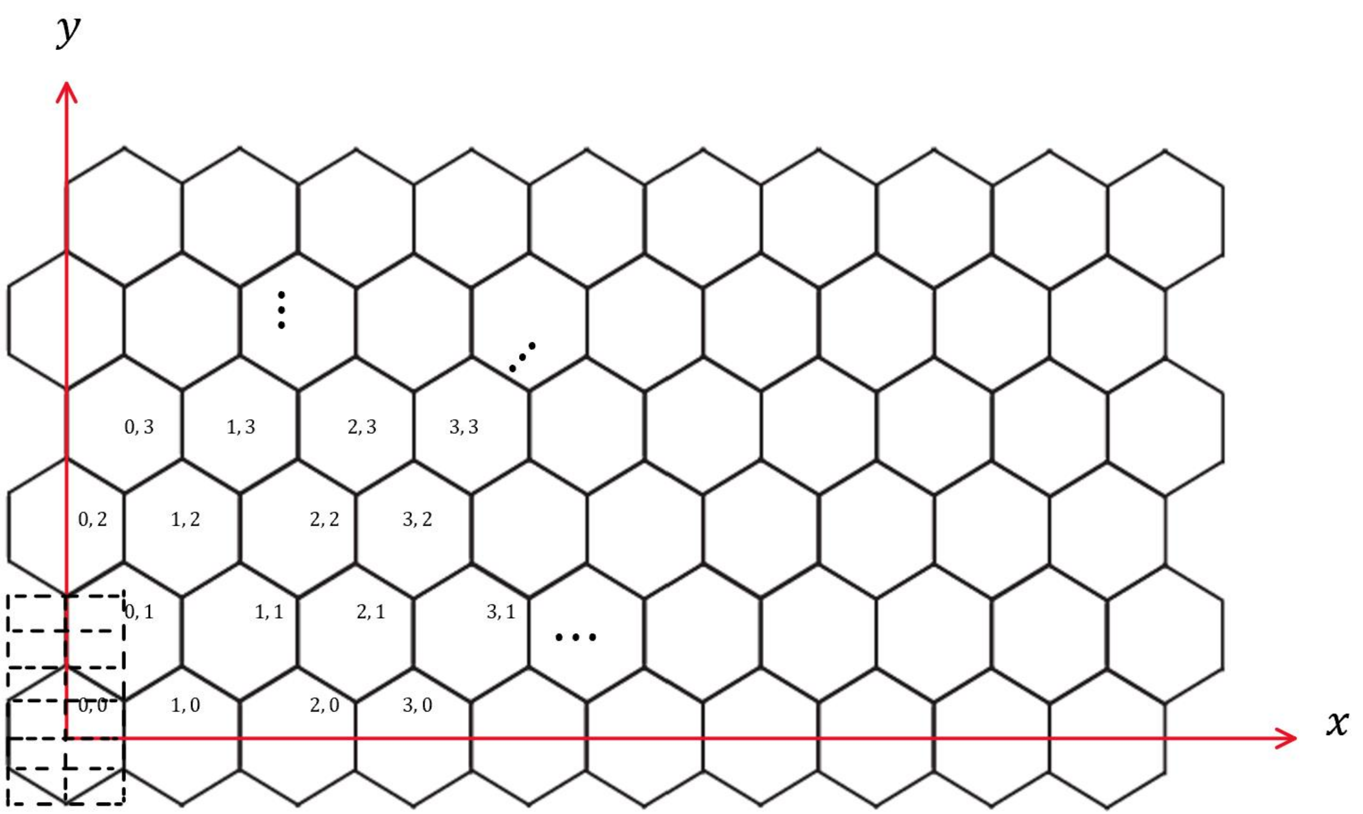
- ...but not so for a grid of hexagonal pixels
- Sides are no longer restricted to being parallel to the x or y-axis



!?

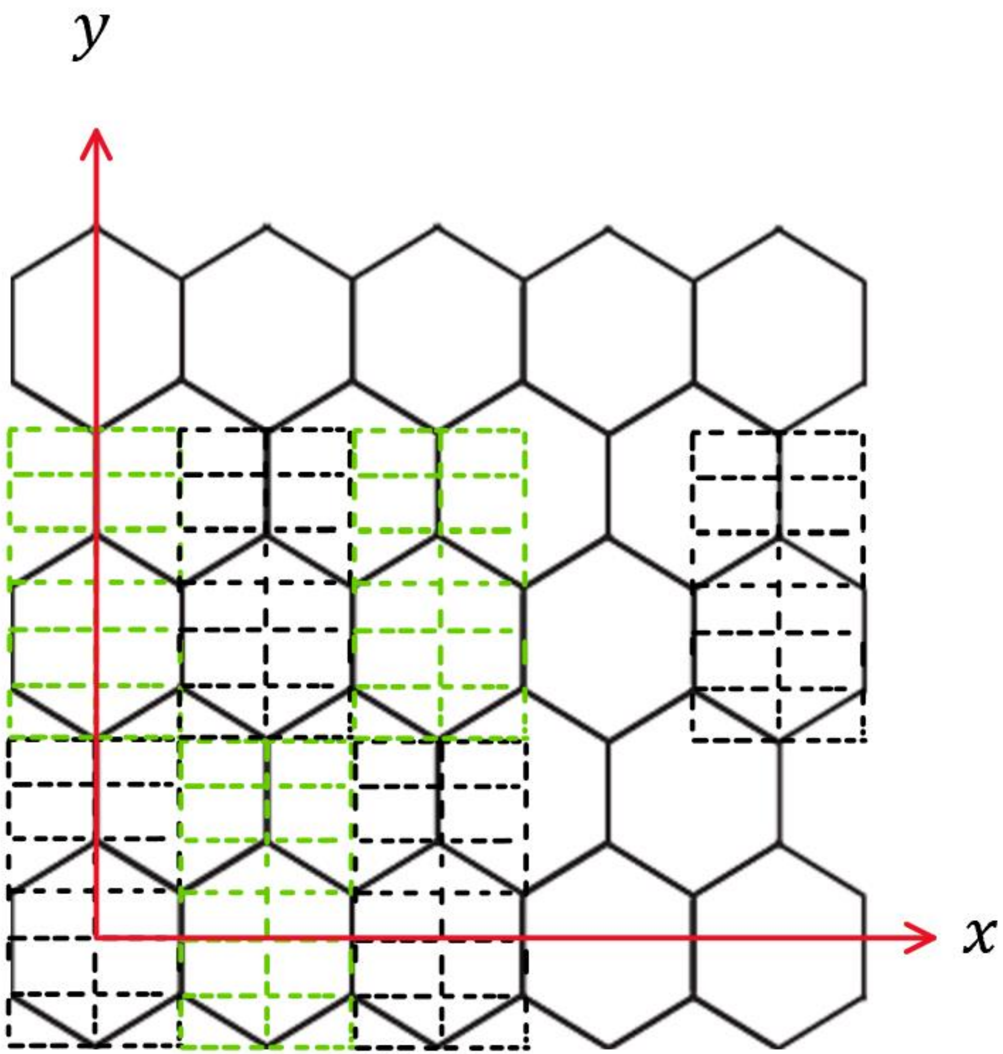
Implementation of Hexagonal Pixel Geometry

- The idea is to look for periodicity where the geometry repeats itself in the x, y directions
- This corresponds to the grid seen at the bottom left of the diagram
- Partially inspired by earlier attempt to implement Hexagonal pixel geometry by Tasneem et al. https://github.com/allpix-squared/allpix-squared/pull/33/files



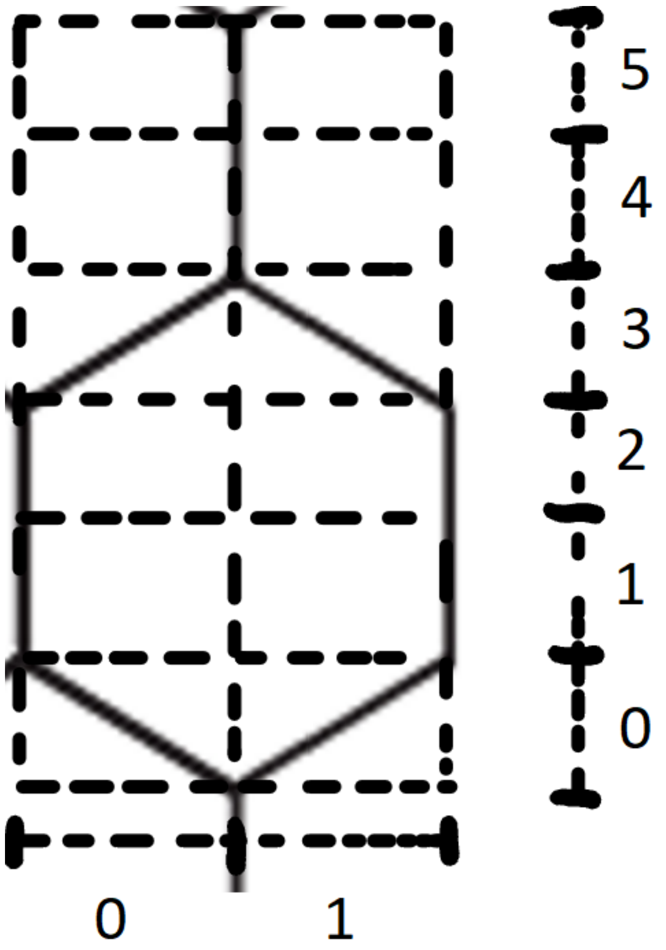
Implementation of Hexagonal Pixel Geometry

- The rectangular ‘unit’ cell is repeated through the entire hexagonal pixel grid
- The pixels are assumed to be normal hexagons
  - Need modification for irregular hexagonal pixels



Implementation of Hexagonal Pixel Geometry

```
switch(y_modulus % 6) {
case 0: {
    if(x_modulus % 2 == 0) {
        offset =
            (3 * side * std::floor(posy / (3 * side))) + ((std::sqrt(3) / 3) * ((x_modulus + 1) * minor_radius));
        if(posy < (-(std::sqrt(3) / 3) * posx + offset)) {
            xpixel -= 1;
            ypixel -= 1;
        }
    } else if(x_modulus % 2 == 1 || x_modulus % 2 == -1) {
        offset = (3 * side * std::floor(posy / (3 * side))) - ((std::sqrt(3) / 3) * (x_modulus * minor_radius));
        if(posy < ((std::sqrt(3) / 3) * posx + offset)) {
            ypixel -= 1;
        }
    }
    break;
}
case 1:
case 2: {
    break;
}
case 3: {
    if(x_modulus % 2 == 0) {
        offset = side * (3 * std::floor(posy / (3 * side)) + 2) -
                 (std::sqrt(3) / 3) * (minor_radius * (x_modulus + 1));
        if(posy > ((std::sqrt(3) / 3) * posx + offset)) {
            xpixel -= 1;
            ypixel += 1;
        }
```



- The unit cell is then divided into 12 different regions
- Pixel index is assigned depending on which region the hit position corresponds to

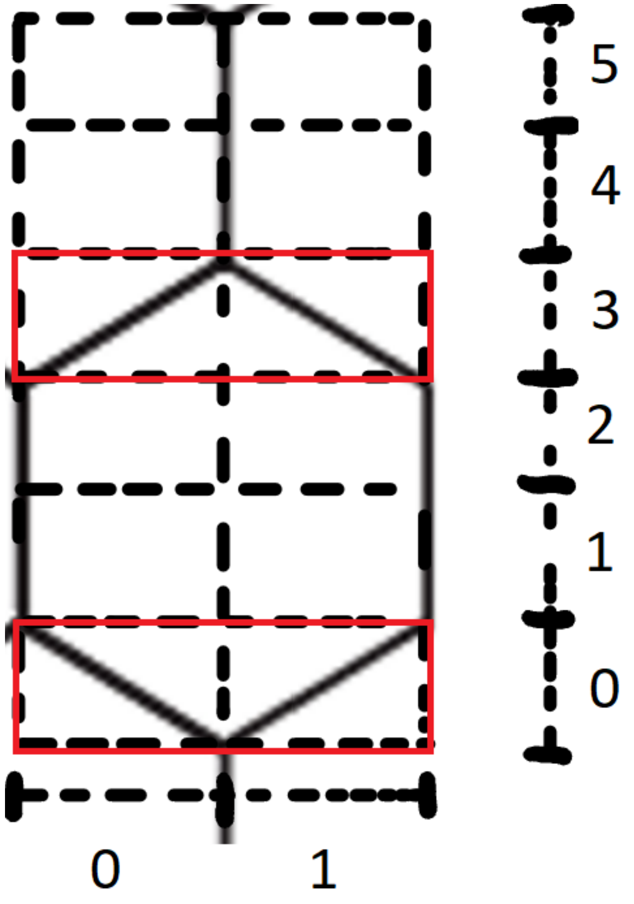
Implementation of Hexagonal Pixel Geometry

- What about regions seen in the diagram (red)?

$\Longrightarrow$ use linear equations!

$$y = mx + b$$

- Assign a different index depending on if it is above or below the side length



Basic Simulation of Hexagonal Pixel Sensor

- A basic comparison between Hexagonal Pixel sensor vs. Rectangular Pixel sensor was performed
- Model used: MIMOSA26
  - Model already implemented within Allpix Squared
- Number of events: 15,000
- Pixel pitch was adjusted for both sensors to ensure same area
- Assume only geometrical effect
  - Custom electric field for hexagonal geometry were not included

Hexagonal Sensor Geometry

Hex_M26.conf

```
[Allpix]
number_of_events = 15000
detectors_file = "Hex_M26_Pos.conf"
log_level = "WARNING"
random_seed_core = 1234
root_file = "Hexagonal.root"
model_paths = ./

[GeometryBuilderGeant4]

[DepositionGeant4]
particle_type = "e-"
source_energy = 5GeV
source_type = "beam"
beam_size = 3mm
source_position = 0 0 -200mm
beam_direction = 0 0 1
physics_list = FTFP_BERT_EMZ
output_plots = 1

[ElectricFieldReader]
model = "linear"
bias_voltage = -4V
depletion_depth = 15um

[GenericPropagation]
name = "detector"
temperature = 293K
charge_per_step = 10
output_plots = 1

[SimpleTransfer]
max_depth_distance = 5um

[DefaultDigitizer]
electronics_noise = 13e
threshold = 60e
threshold_smearing = 4e
qdc_smearing = 0e
output_plots = 1

[DetectorHistogrammer]
max_cluster_charge = 10ke
```

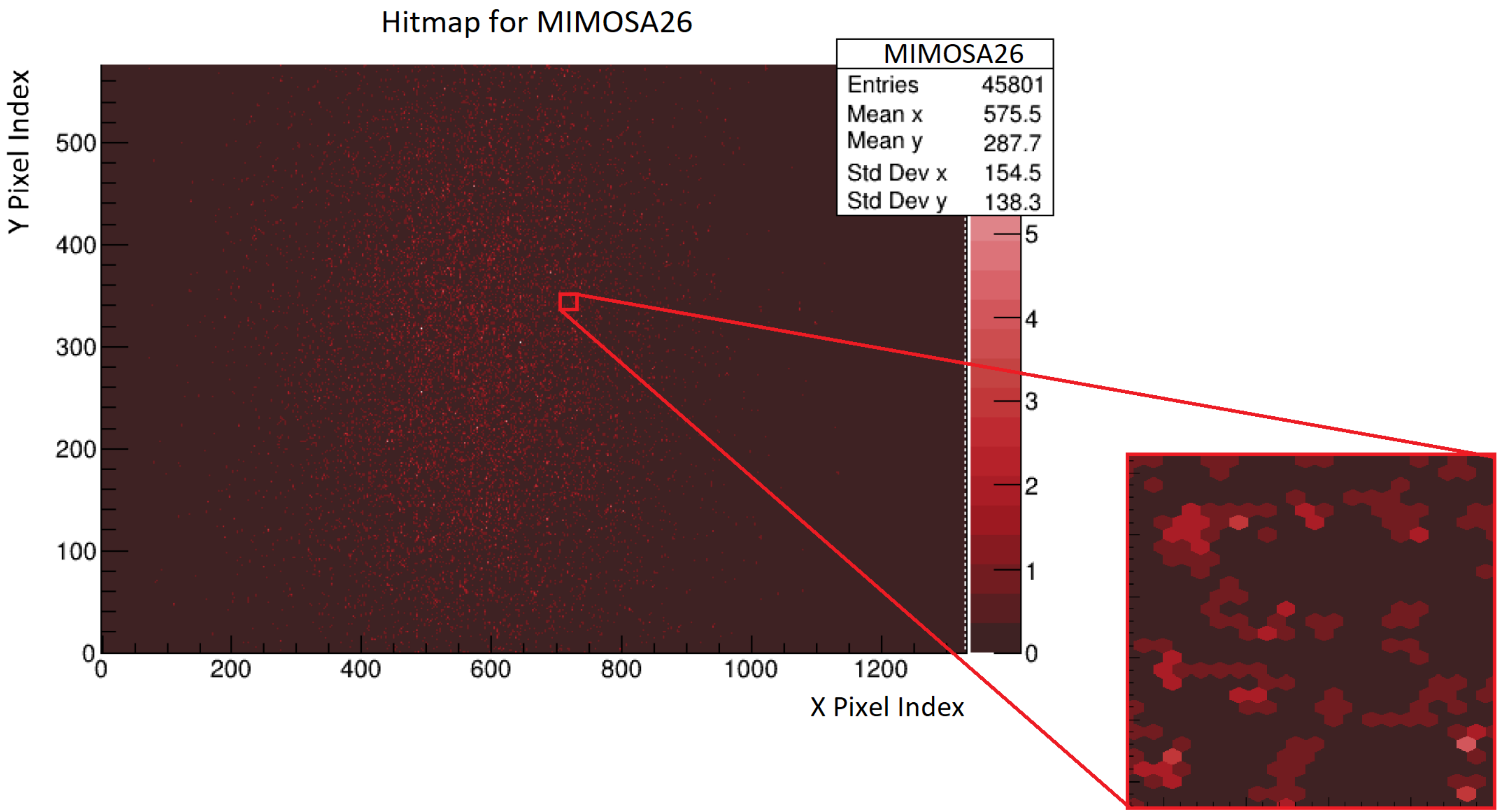
Hex_M26_Pos.conf

```
[detector]
type = "Hex_M26_Geometry"
position = 0mm 0mm 0mm
orientation = 0 0 0
```

Hex_M26_Geometry.conf

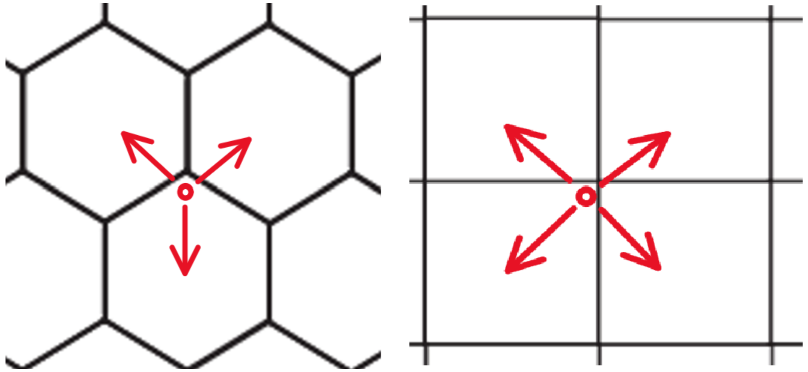
```
# Reference http://www.iphc.cnrs.fr/IMG/baudot_n22-6_nss09proc2.pdf
# See Figure 1 therein
type = "hexagonal"

number_of_pixels = 1152 576
pixel_size = 19.7um 19.7um

sensor_thickness = 45um
sensor_excess_top = 200um
sensor_excess_bottom = 3000um
sensor_excess_left = 350um
sensor_excess_right = 0um

chip_thickness = 5um

# PCB chip board with cut-out for the sensor
[support]
thickness = 1.6mm
size = 61.5mm 79.5mm
offset = 0 -15.0mm
hole_size = 20mm 10mm
hole_offset = 0 14.5mm
material = "g10"

# Kapton foils for shielding the sensor, 25um on both sides
[support]
thickness = 25um
size = 30mm 20mm
location = "absolute"
offset = 0 0 10mm
material = "kapton"
[support]
thickness = 25um
size = 30mm 20mm
location = "absolute"
offset = 0 0mm -10mm
material = "kapton"
```

Hit Map

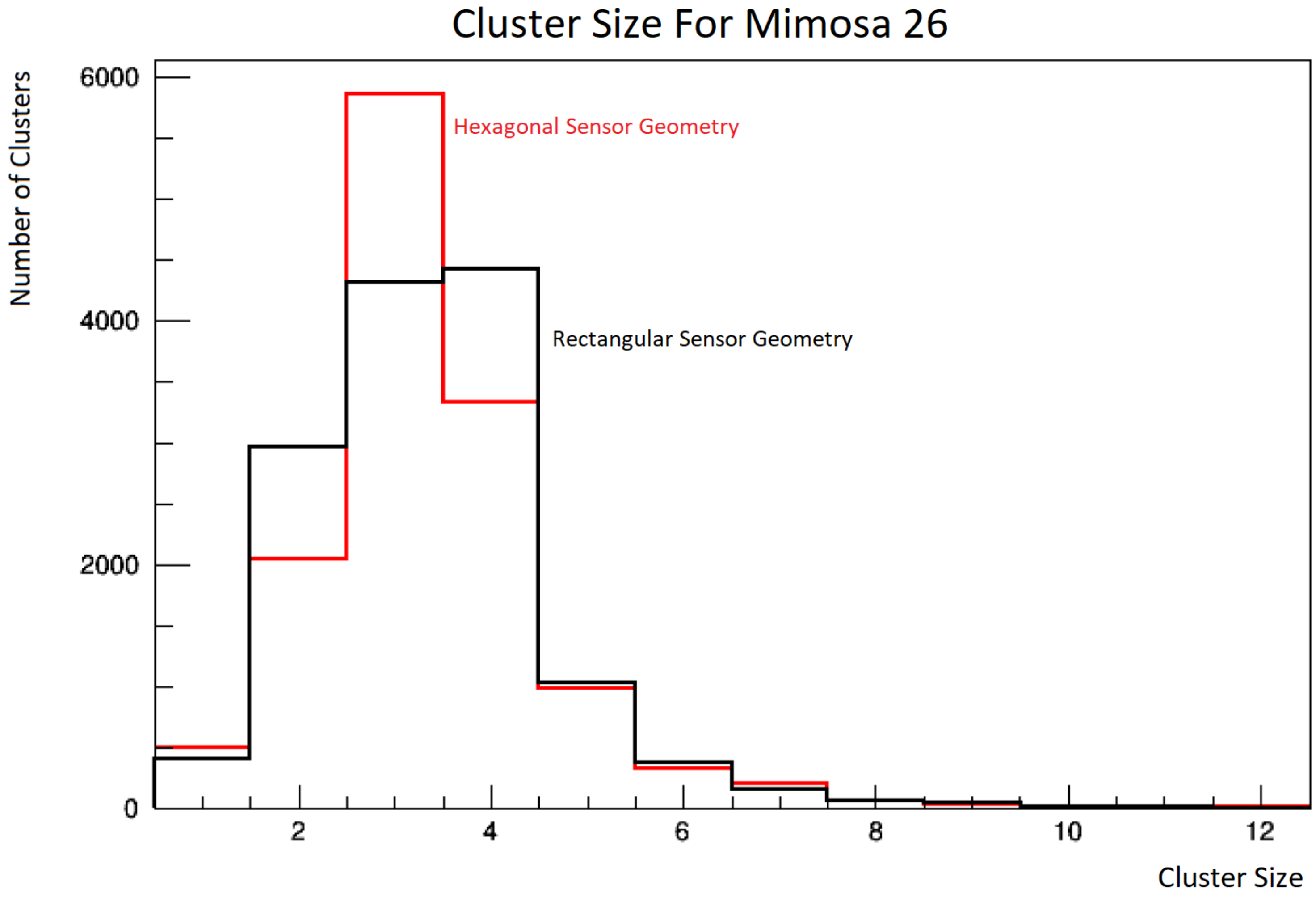


Hitmap for MIMOSA26
MIMOSA26
Entries 45801
Mean x 575.5
Mean y 287.7
Std Dev x 154.5
Std Dev y 138.3
Y Pixel Index
X Pixel Index
0
100
200
300
400
500
0
200
400
600
800
1000
1200
0
1
2
3
4
5

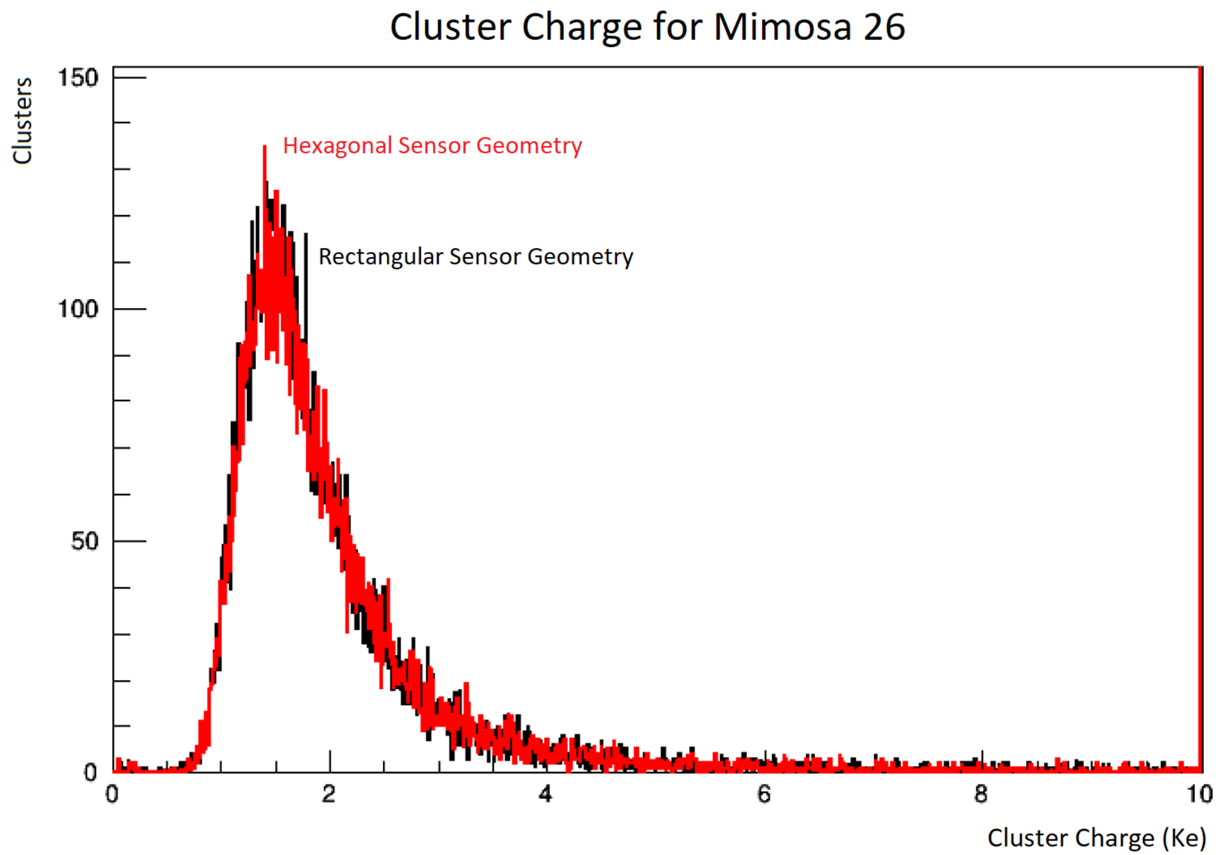
Cluster Size

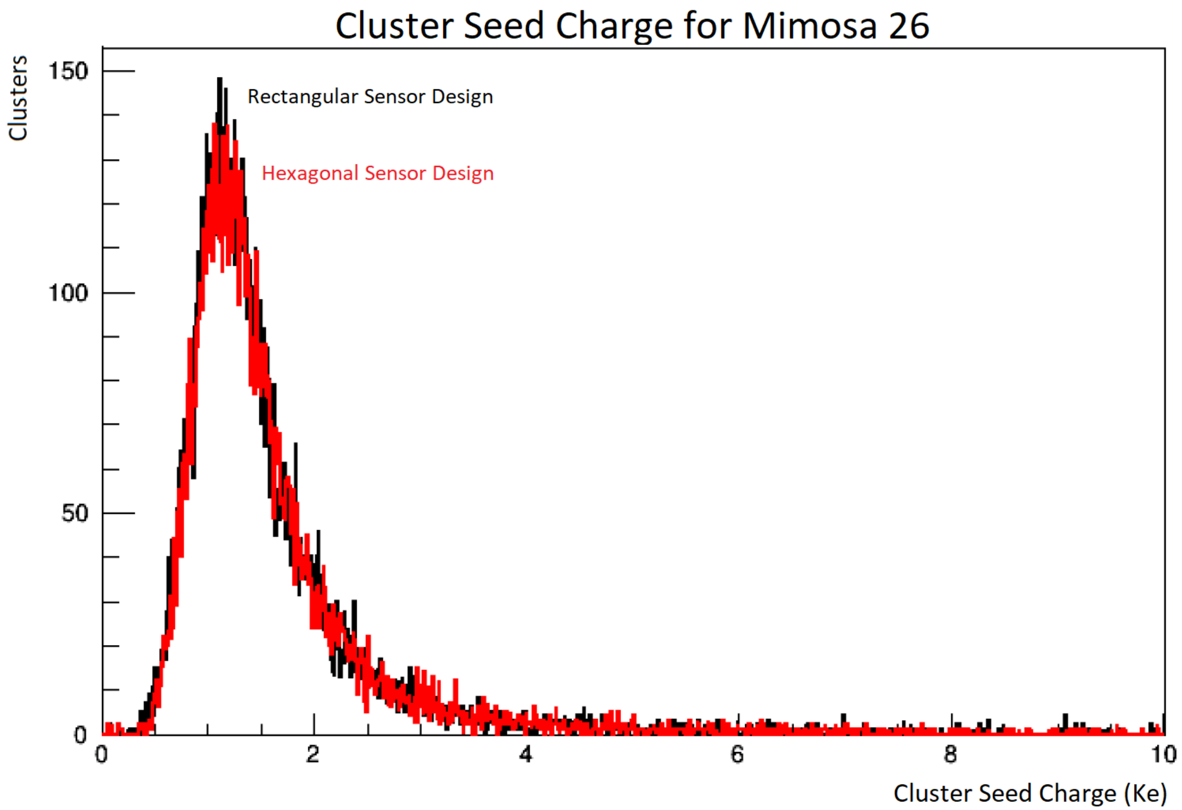


- Cluster size of 3 dominating in hexagonal case
- More likely to see cluster size of 3 due to 3 neighbor pixels at edge



Cluster Charge





- Cluster seed charge: expected higher peak for hexagonal pixels due to less charges shared with neighbors
  - However, the shape is practically the same
  - Still under investigation

Summary and Outlook

- Outlook:
  - Preparation for importing field/potential maps for hexagonal pixel cells is on-going
- Summary:
  - Hexagonal pixel sensors provide several benefits (less charge sharing, better time resolution, etc.)
  - New detector class dedicated to hexagonal pixel sensors was created
  - Basic simulation of MIMOSA26 with hexagonal pixels was performed (hitmap, cluster size, cluster charge, etc.)

Thank you!

Questions?

For further questions/suggestions: Ryuji.Moriya@glasgow.ac.uk