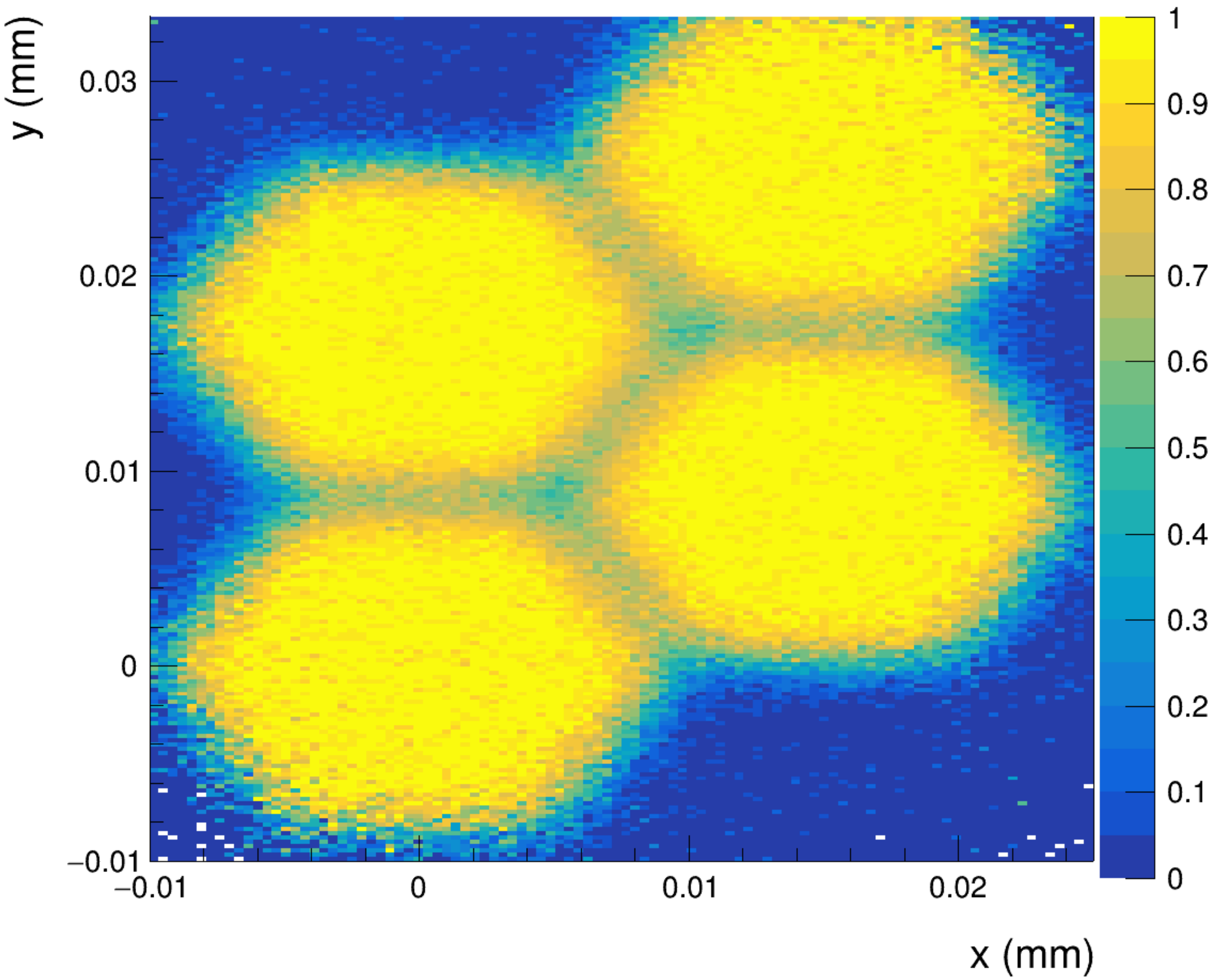
Hexagonal pixel simulations in Allpix Squared

H. Wennlöf, with much input from Larissa Mendes

23/5 -23

Outline

- Introduction
  - Hexagonal geometry
- Implementation in Allpix Squared
- Usage how-to
- Example studies
  - Both fast and detailed simulations
- Conclusions and outlook

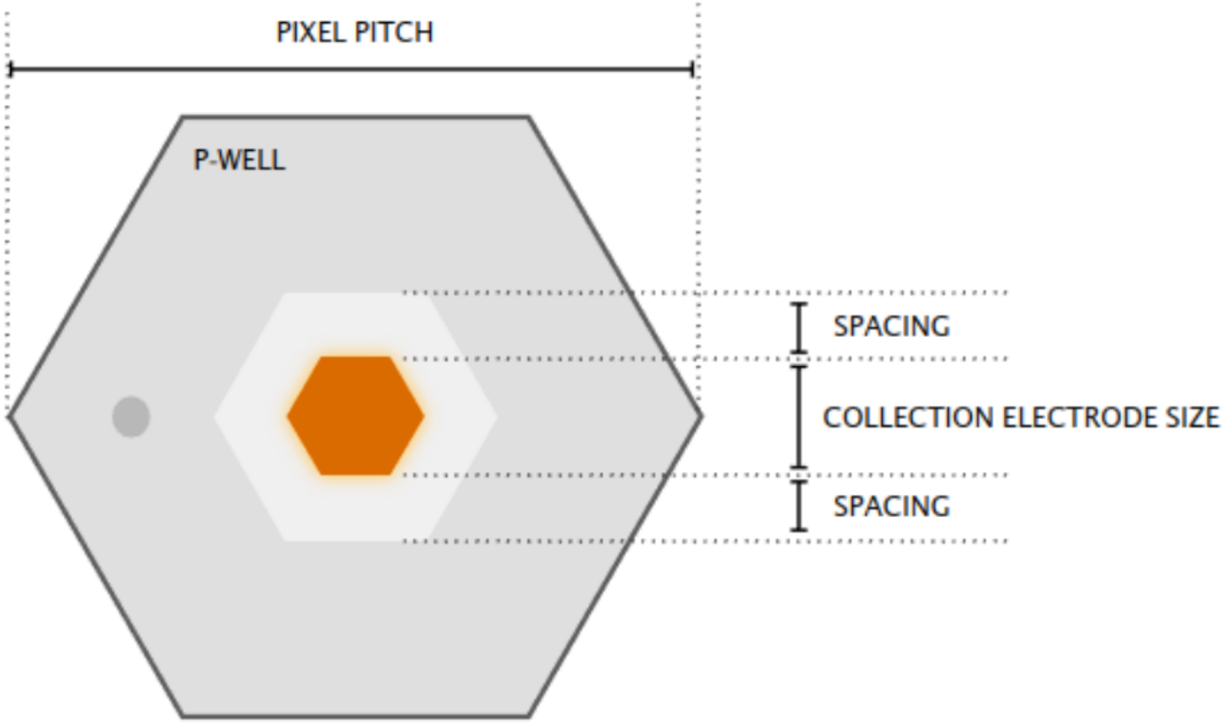


Hexagonal efficiency map in Allpix Squared (MR 539)

Hexagonal pixel geometry

Benefits of hexagons

- Hexagonal pixels used in more and more developments
  - E.g. FASTPIX, MONOLITH, PicoAD
- Several **benefits** compared to rectangular pixels
  - Charge sharing only between 3 pixels
    - Maintains **efficiency** for small signals
  - Reduced **pixel perimeter**
    - 7% less than for a square pixel with the same area
  - **Reduced distance** between edges and collection electrode
    - Potentially **faster** charge collection
  - More obtuse corners - 120° instead of 90°
  - Same distance between all adjacent pixel centres

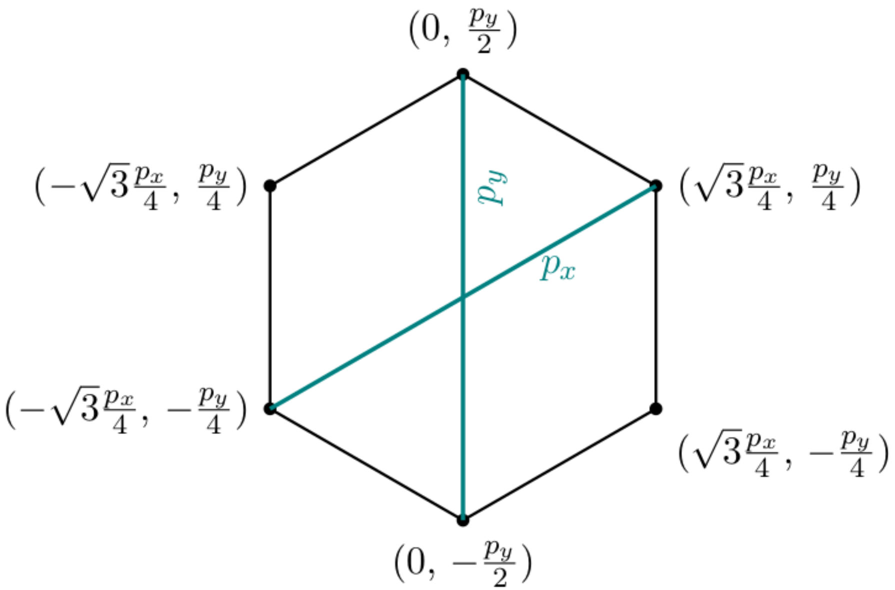


L. Mendes

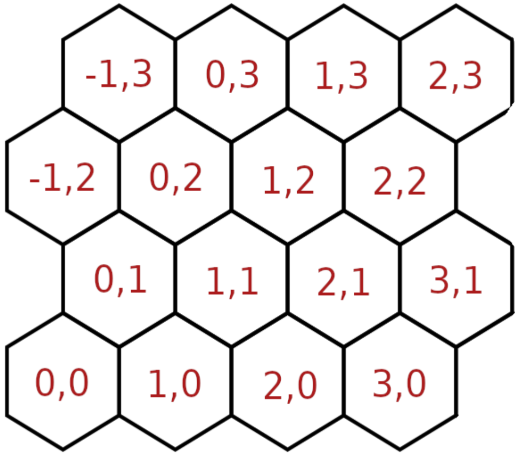
Hexagonal pixel geometry

Definitions

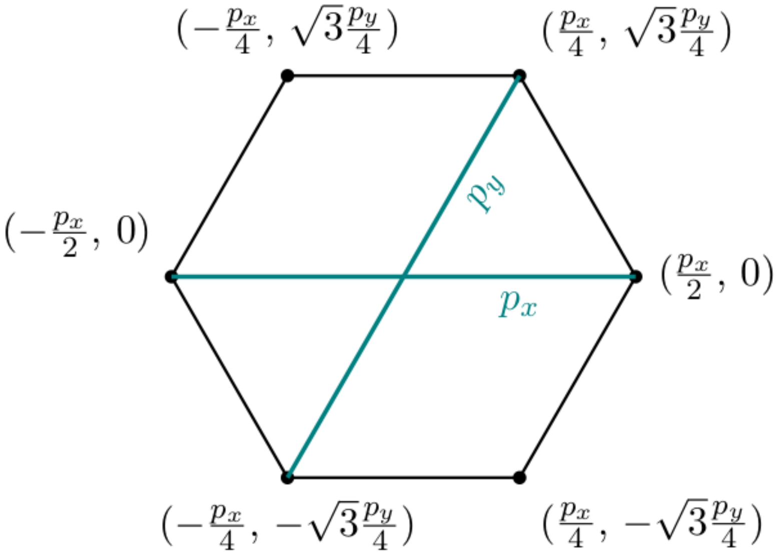
- An **axial** coordinate system is used for hexagonal pixel matrices in Allpix Squared
  - Good resource on hexagonal systems can be found here
- Pixel pitch needs to be defined differently to rectangles
  - In Allpix Squared, pitch is defined by the "**outer radius**" of the hexagon (i.e. as "**corner to corner**")
    - p_x, p_y in the figures on the right
  - Pitches align with axial coordinate system
- 2 hexagon orientations; "pointy" and "flat"
  - "Pointy" have sides parallel to Cartesian y
  - "Flat" have sides parallel to Cartesian x



"**Pointy**" hexagon geometry



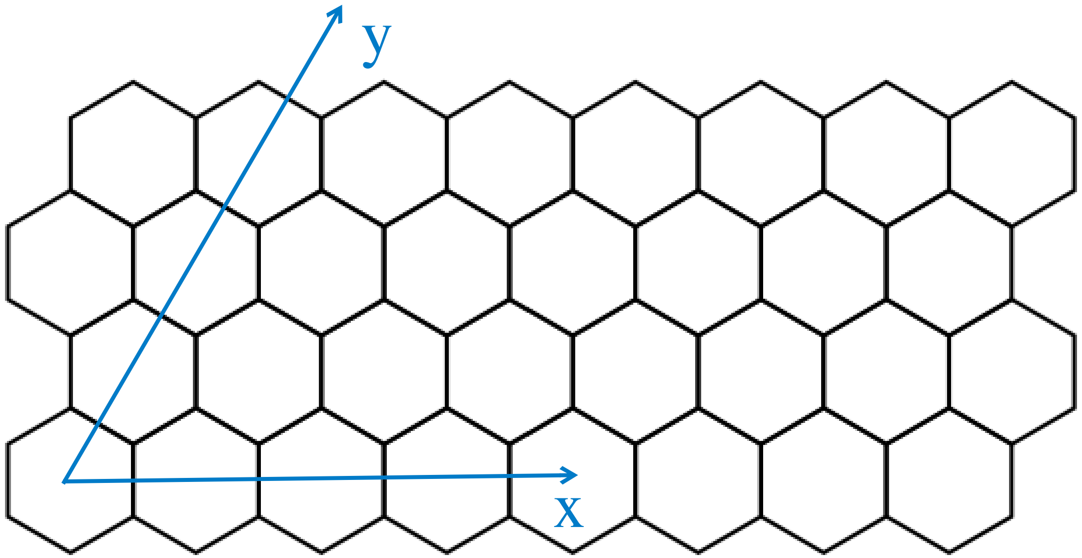
Figures from MR 539



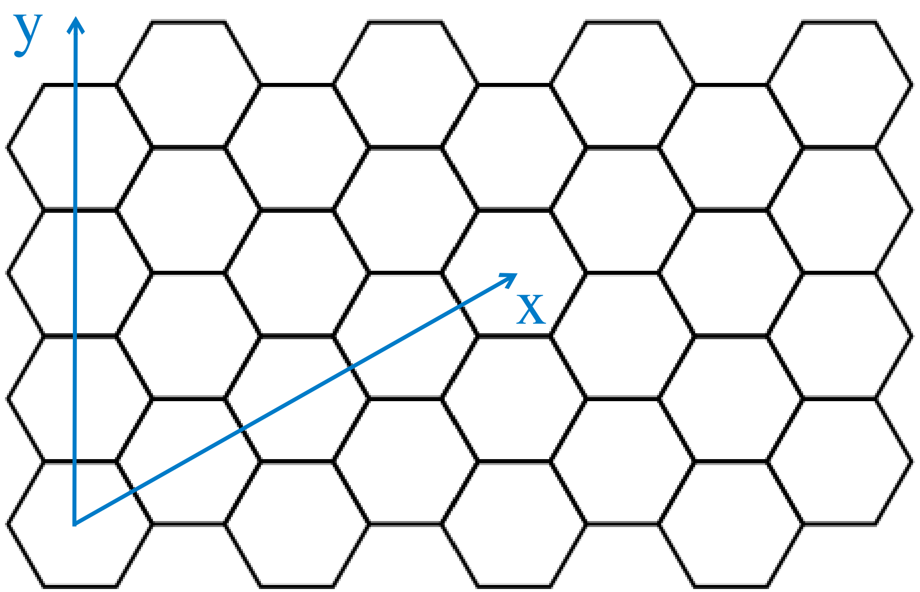
"**Flat**" hexagon geometry

Hexagonal pixel geometry

Axial coordinate system



(a) 8x4 grid with *pointy* hexagons



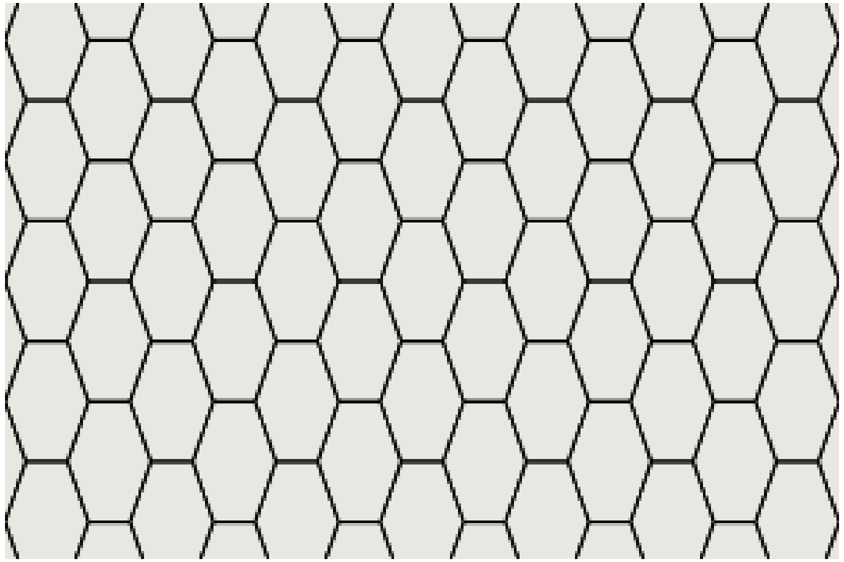
(b) 8x4 grid with *flat* hexagons

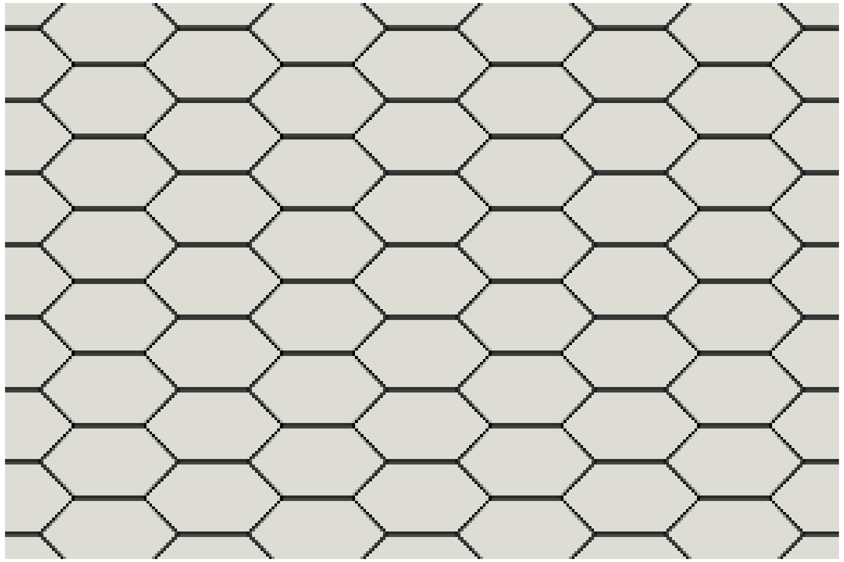
Figures from documentation

- Number of pixels counted along **Cartesian axes**, taking offset into account
- Results in different matrix shapes for pointy and flat orientations

Hexagonal pixel geometry

Irregular shapes





Figures from MR 539

- The pixel size can be **different in the two directions**, resulting in a stretched/squished hexagons

How-to in Allpix Squared

Usage

- What is needed in the configurations?
  - Changes in the **detector model** file
  - Set the **geometry** parameter to “hexagonal”
  - Select the hexagon orientation via the **pixel type** parameter
- Set number of pixels, see slide 5 on the effect of hexagon orientation

```
geometry = "hexagonal"

pixel_type = "hexagon_pointy"

pixel_type = "hexagon_flat"
```

- **Note:** things to keep in mind for analysis
  - **Negative pixel indices** are possible (difference to rectangular pixels)
  - Coordinate system is different (**angle between axes is not 90**°)

Example studies

Simple fast simulations

- Allows **rapid prototyping** of algorithms, and testing of geometries
  - Example: hexagonal eta correction development
- Detector model (on the right) set up for a monolithic sensor with flat-topped hexagons

```
type = monolithic
geometry = "hexagonal"

#pixel_type = "hexagon_pointy"
pixel_type = "hexagon_flat"

number_of_pixels = 20 20
pixel_size = 25um 25um

sensor_thickness = 100um
```

Simple fast simulations

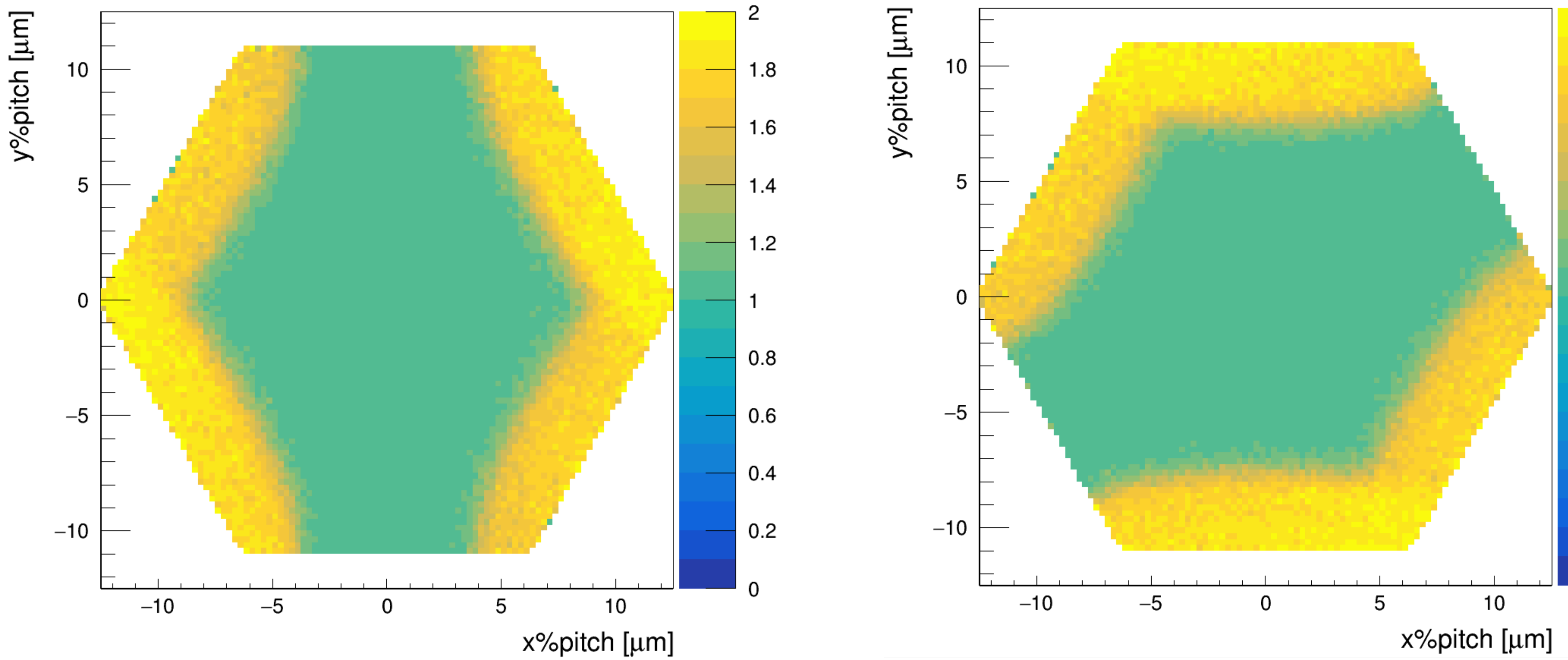
- Pion beam
- **Linear** electric field
- **Projection** propagation

- Simple and fast, 500 000 events takes ~7 minutes to run with 7 cores on my laptop

```
[Allpix]
number_of_events = 500000
detectors_file = "detector.conf"
multithreading = true

[GeometryBuilderGeant4]

[DepositionGeant4]
particle_type = "Pi+"
source_energy = 120GeV
source_type = "beam"
beam_size = 20um
source_position = 0um 0um -200mm
beam_direction = 0 0 1

[ElectricFieldReader]
model="linear"
bias_voltage=-50V
depletion_voltage=-30V
output_plots = 1

[ProjectionPropagation]
charge_per_step = 10
output_plots = 1

[SimpleTransfer]
output_plots = 1

[DefaultDigitizer]
output_plots = 1
threshold = 500e

[DetectorHistogrammer]
output_plots = 1
track_resolution = 0 0
granularity = 100, 100
```

Simple fast simulations - results

Cluster size



Cluster size in (axial) x

Cluster size in (axial) y

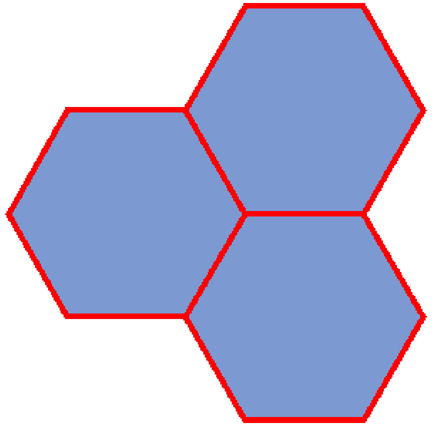
- Cluster sizes in x- and y-directions, in the hexagonal coordinate system for a 25x25 μm pixel size

Simple fast simulations - results

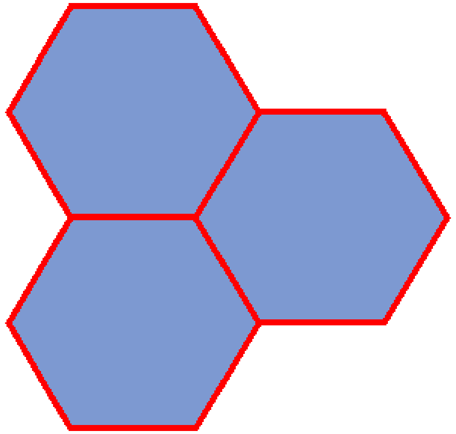
Full cluster size

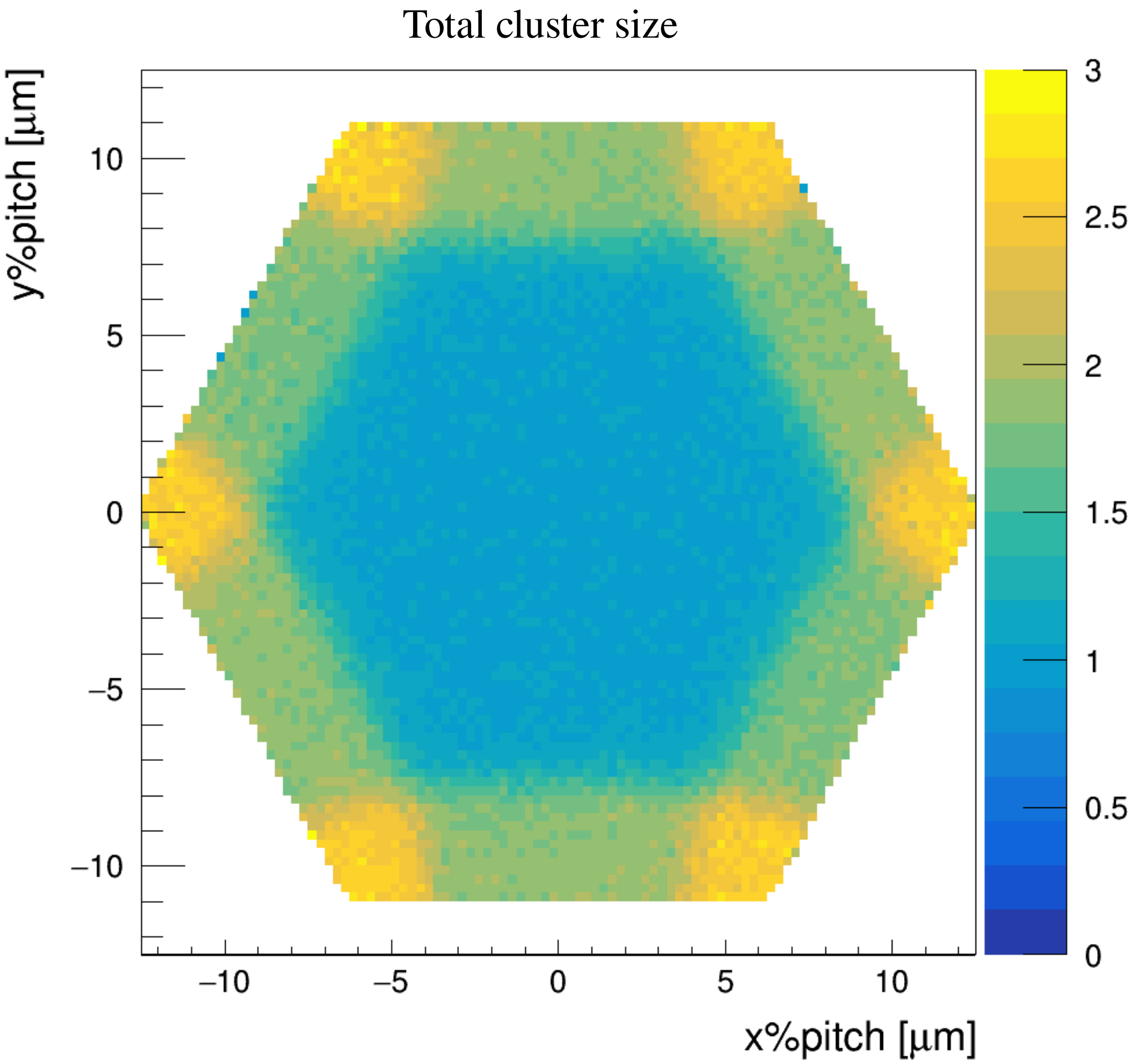
- Cluster size and shape looks as expected
  - The tested sensor is relatively **thick** in order to reach higher cluster sizes in places, to allow testing of eta correction calculations for both 2- and 3-pixel clusters
- “Bottom-left” pixel in a cluster used as reference for cluster shape; leads to two most-common cluster shapes for 3-pixel clusters

48.7% of 3-pixel clusters



48.5% of 3-pixel clusters

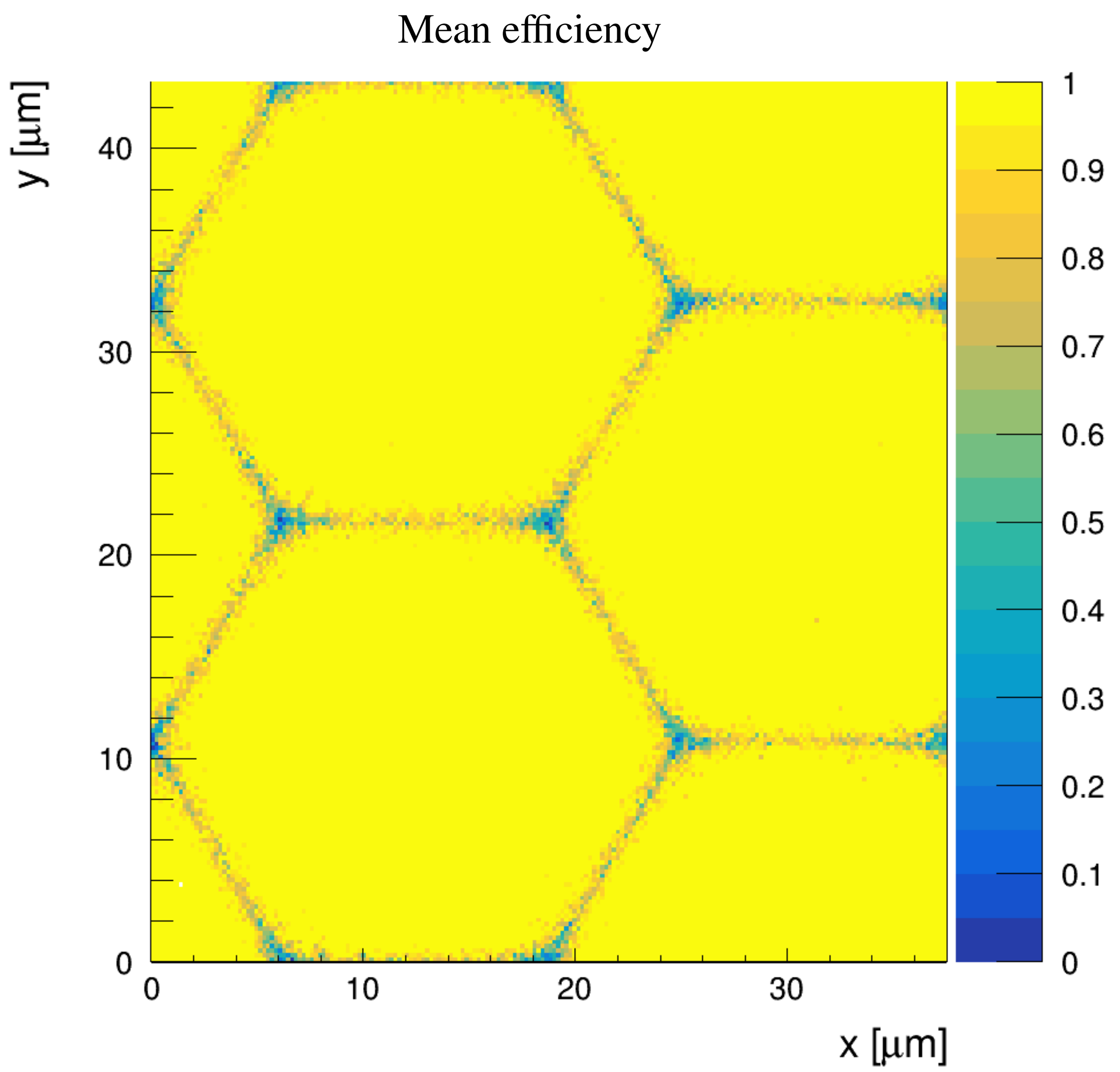




Simple fast simulations - results

Mean efficiency

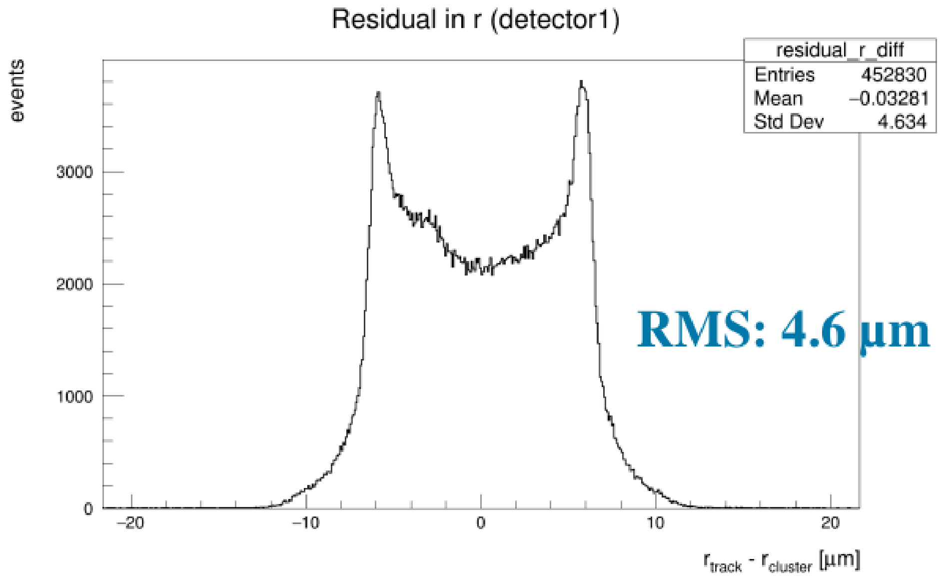
- Efficiency looks as expected
  - Threshold is relatively high here, just to demonstrate a drop occurring at pixel edges first
- Flat hexagonal shape shows up well
  - Bins are square, but many



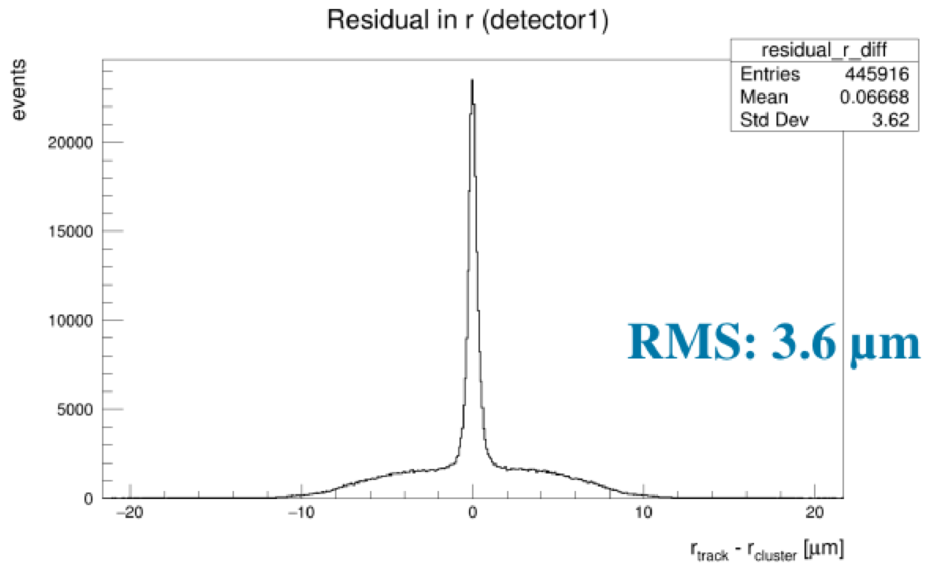
Example: hexagonal eta correction

- Correcting for **nonlinear charge sharing** in hexagonal pixels
- Needs to be different compared to rectangular pixels
  - Working in an r-φ coordinate system
  - Allows correction also of 3-pixel clusters
- Won't go into details, but development was simplified by the fast simulations, and the **algorithm has been applied** both to data (FASTPIX, by J. Braach) and simulations in the Tangerine project

Full residual in r, before



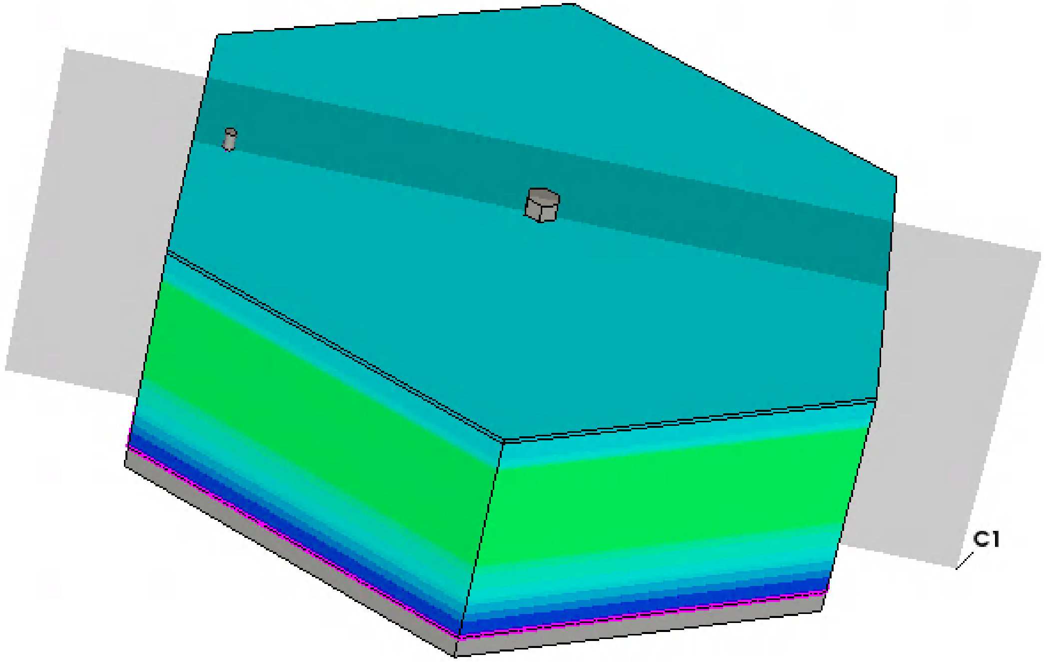
Full residual in r, after both 2- and 3-cluster corrections

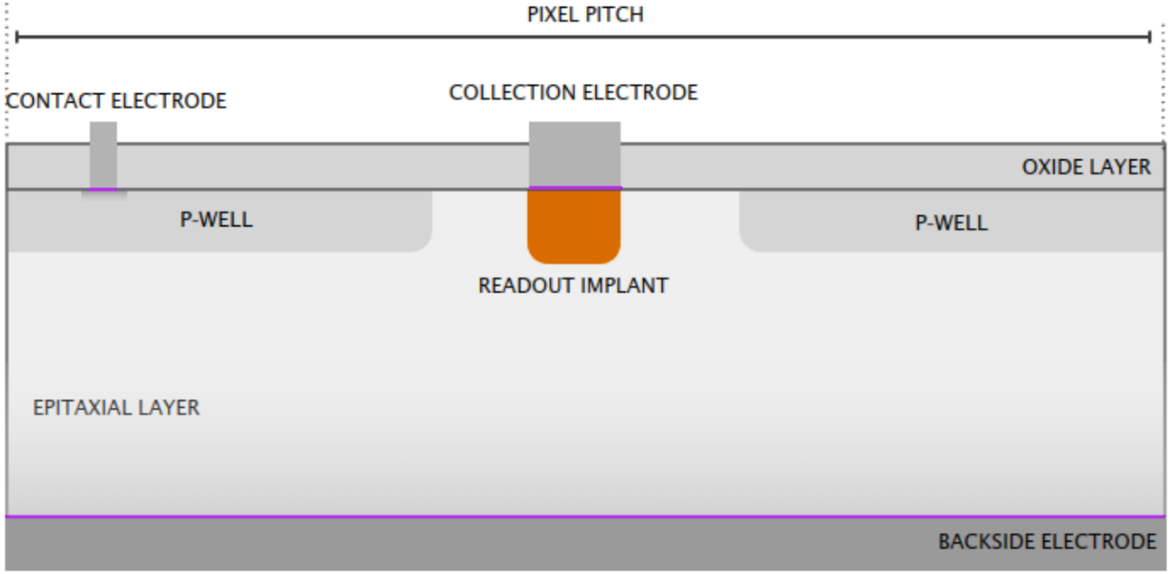


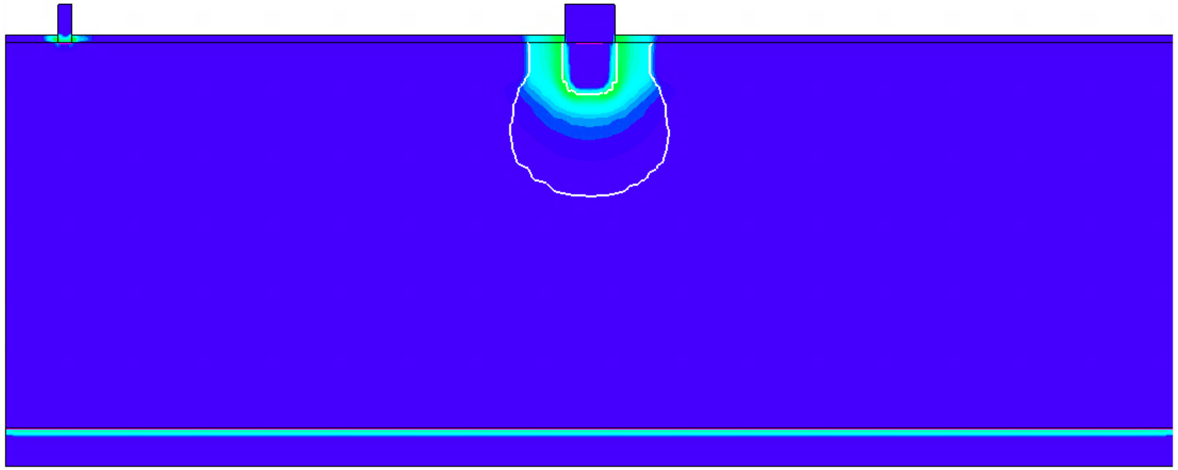
Hexagonal simulations in the Tangerine project

All results by L. Mendes

- Generating TCAD fields with the methodology shown in Adriana's presentation earlier
- Full 3D simulations of a single full hexagonal pixel
- Electric field and doping concentration exported





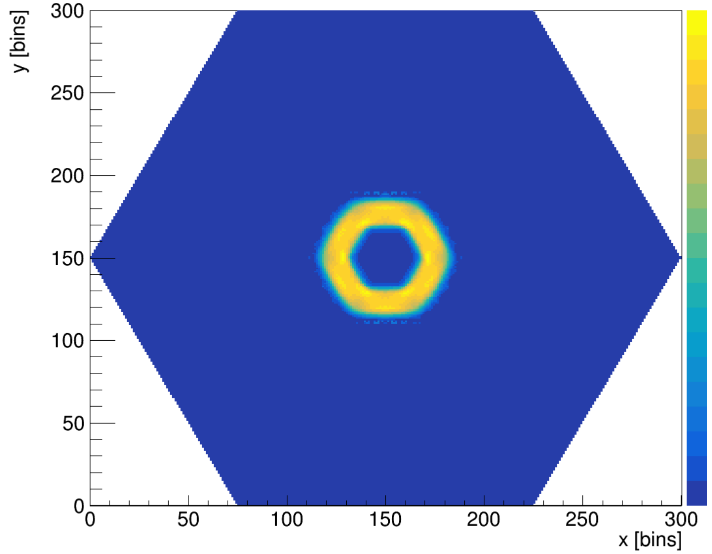


Importing a TCAD field

- Imported using the **mesh converter** tool
  - Important: tool uses interpolation, and expects points to interpolate between
  - If none found, the search radius is increased, and interpolation is aborted when max_radius is reached
- When importing non-rectangular shapes: use the **allow_failure** keyword in the mesh converter configuration
  - Sets **mesh element to zero** if no neighbouring points are found in the input mesh
  - This way, the regularly-spaced mesh used in Allpix Squared will have the correct shape
- Loaded into the simulation using the PIXEL_FULL field mapping (see documentation for other variants)

```
model = "APF"
region = "si-bulk"
observable = ElectricField
observable_units = "V/cm"
divisions = 300 300 100
initial_radius = 0.01
xyz = x y -z
max_radius = 1
allow_failure = true

workers = 20
```

Converted electric field, at a cut in z



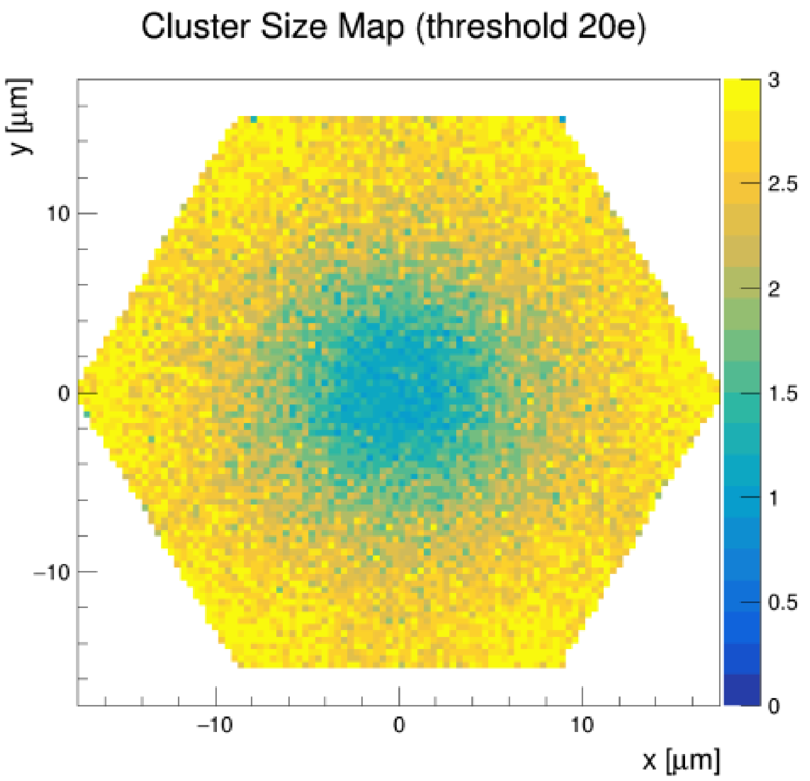
Simulation setup

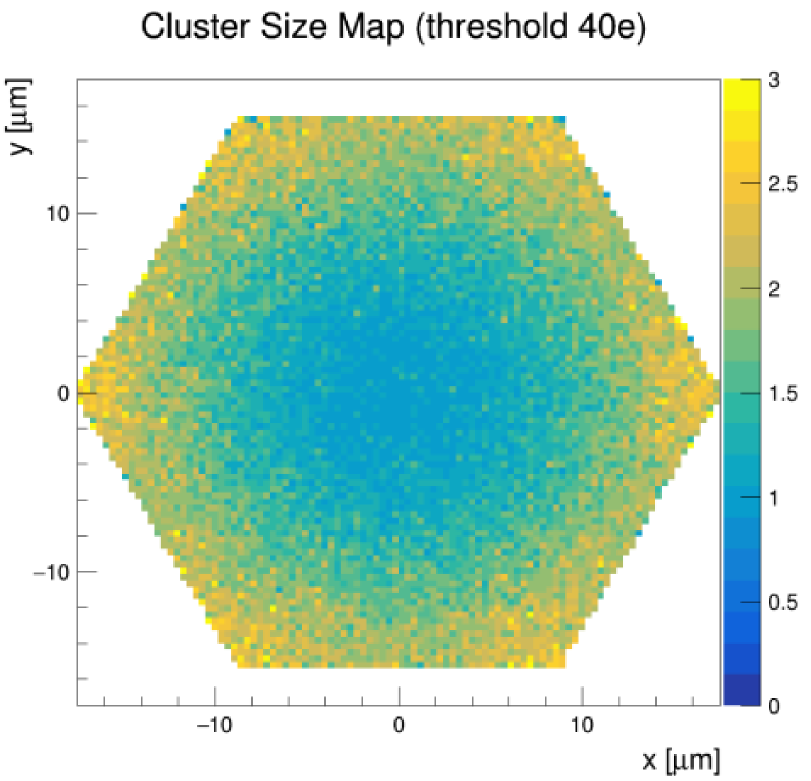
- Using **TCAD** field, and energy deposition via **Geant4**
- Comparing the performance of **hexagonal pixels** to that of **square pixels of equivalent area**
  - The pixel area is a key parameter for fitting the same in-pixel electronics in
- Beam of 5 GeV electrons
  - 1 electron per event
  - 500 000 events
  - Sensitive area much larger than beamspot (100x100 pixels)
- Extended Canali mobility model (accurate over a wide range of field strengths, and doping-dependent)
- Shockley-Read-Hall-Auger recombination model (valid over a wide range of doping concentrations)
- Digitisation stage is run several times with different thresholds

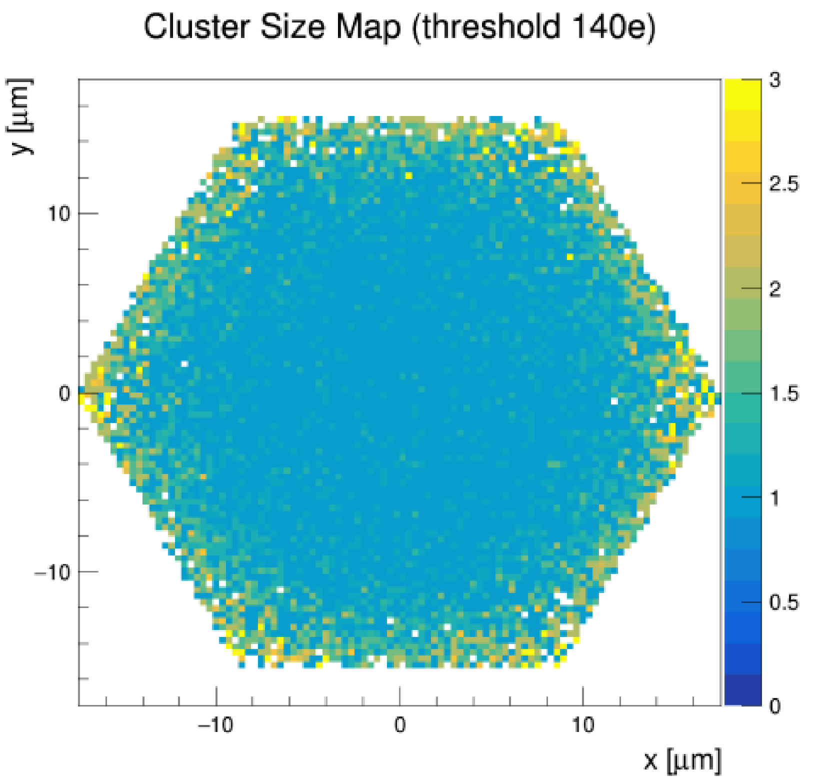
```
[AllPix]
number_of_events = 500000
detectors_file = "../Detector.conf"

[GeometryBuilderGeant4]
world_material = "air"

[DepositionGeant4]
physics_list = QGSP_BERT_EMZ
enable_pai = 1
particle_type = "e-"
number_of_particles = 1
source_energy = 5GeV
source_position = 0um 0um -10mm
source_type = "beam"
beam_size = 100um
beam_divergence = 0mrad 0mrad
beam_direction = 0 0 1
max_step_length = 0.5um

[ElectricFieldReader]
model = "mesh"
file_name = "hex-cmos-field_ElectricField.apf"
field_depth = 10um
field_mapping = PIXEL_FULL

[DopingProfileReader]
model = "mesh"
file_name = "hex-cmos-field_DopingConcentration.apf"
doping_depth = 10um
field_mapping = PIXEL_FULL

[GenericPropagation]
mobility_model="masetti_canali"
recombination_model = "srh_auger"
charge_per_step = 5
timestep_min = 0.5ps
timestep_max = 0.05ns
integration_time = 25ns
```

Results - cluster size

- Comparing standard layout and n-gap layout, for a 35x35 µm hexagonal pixel size
- In-pixel cluster size for different thresholds, for the **standard layout**
- Efficiency loss visible already at a threshold of 140 electrons

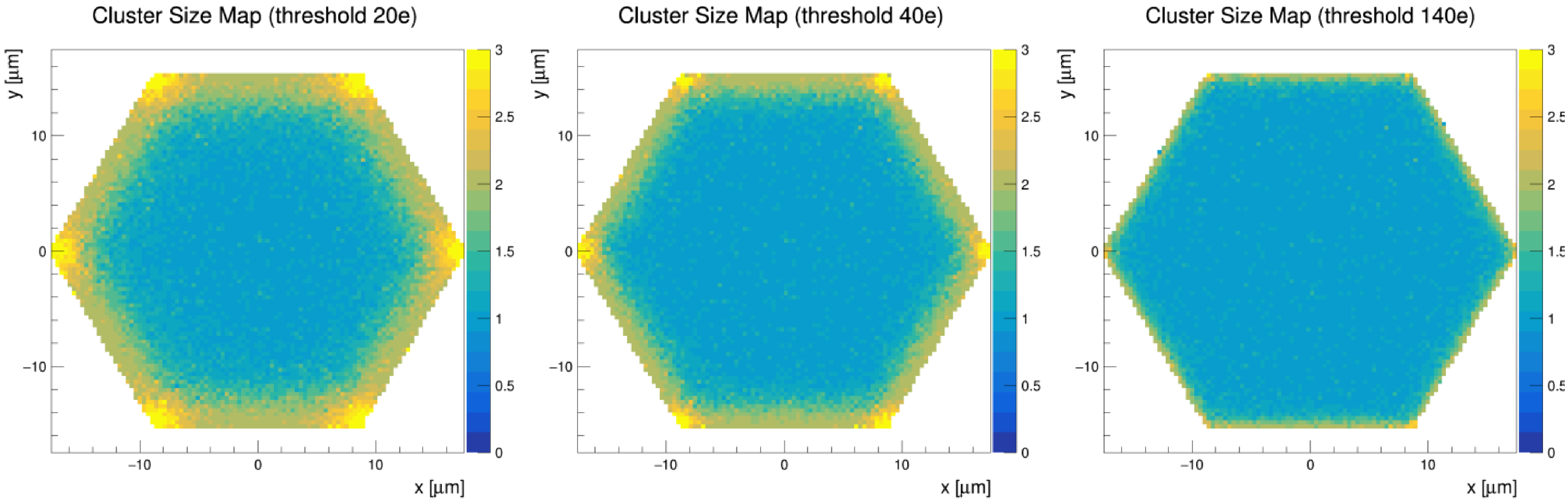






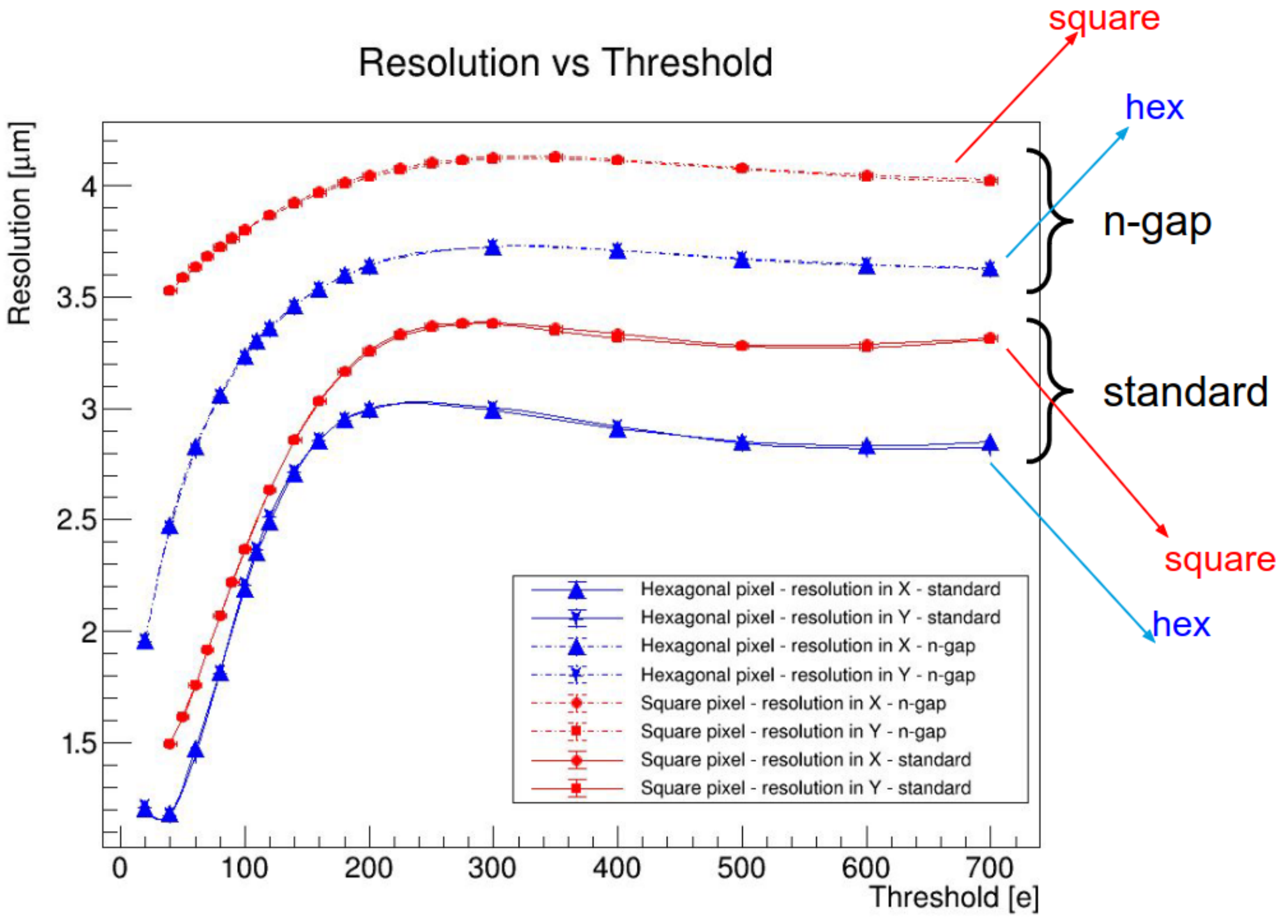
Results - cluster size

- Comparing standard layout and n-gap layout, for a 35x35 µm hexagonal pixel size
- In-pixel cluster size for different thresholds, for the **n-gap layout**
- Smaller than for standard, as expected (n-gap depleted, and funnels charge more towards collection electrode)



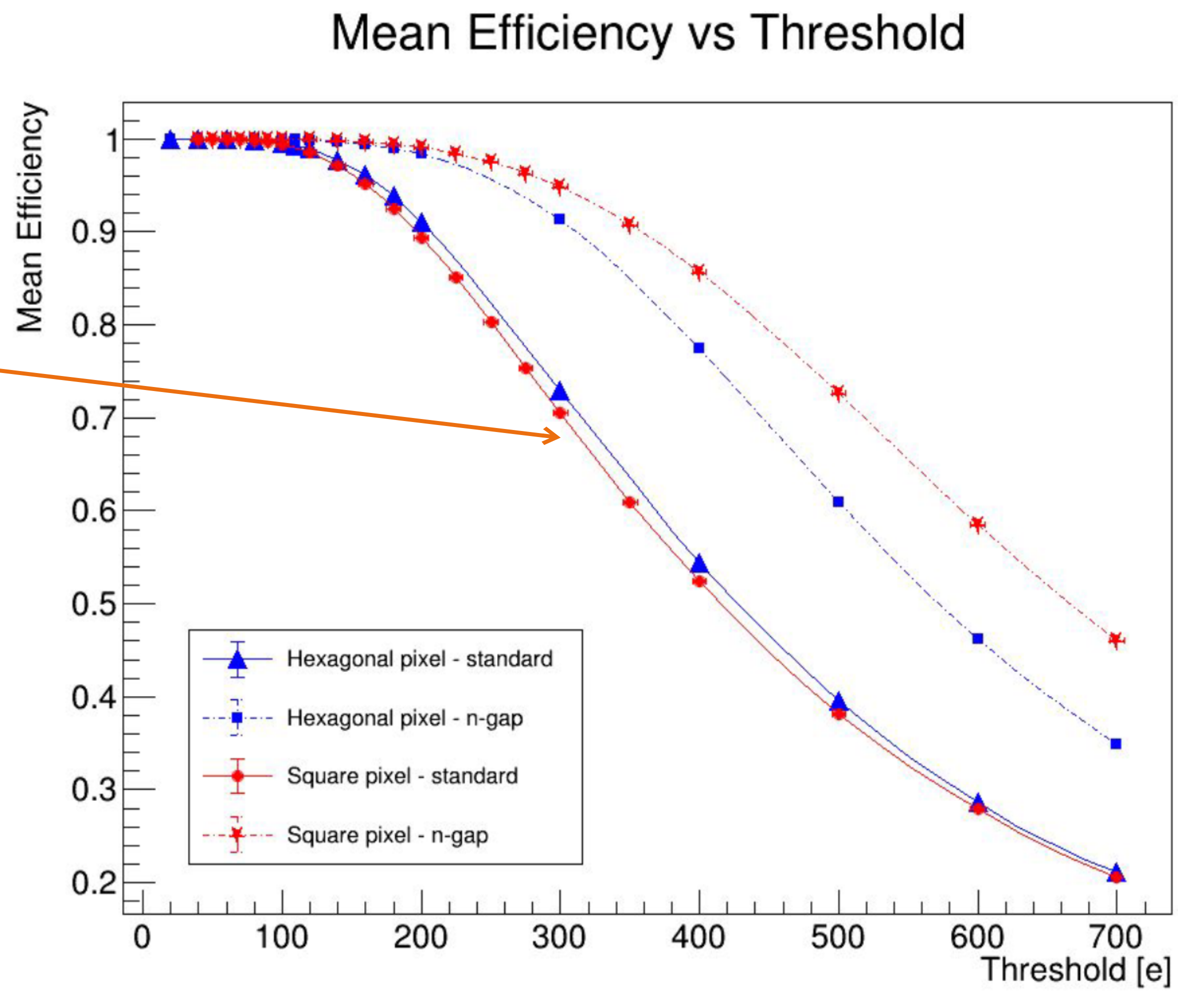
Results - resolution

- Defined as the RMS of the central 3σ of the (MC hit position - reconstructed position) distribution
  - Reconstructed position is charge-weighted mean position for a cluster, using the full collected charge information
- Comparing a square pixel with a size of 14.5x14.5 µm and a hexagonal pixel with a size of 18x18 µm (**same pixel area**)
- Standard has **better resolution than n-gap** in both geometries, due to increased cluster size and thus better position interpolation
- Hexagonal geometry **significantly improves resolution** compared to square



Results - efficiency

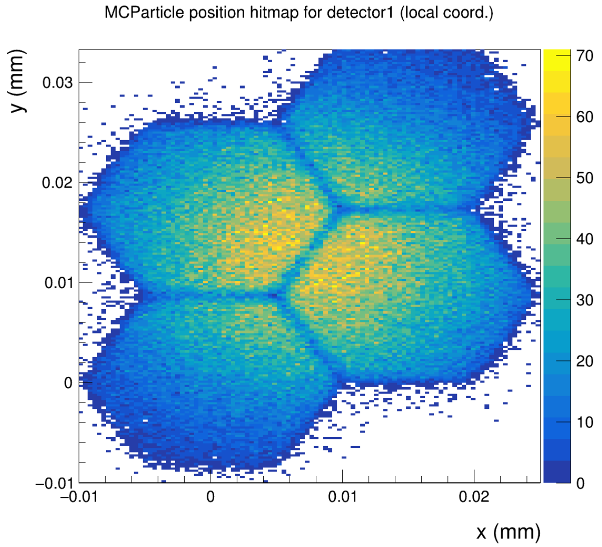
- Mean efficiency for different thresholds
- Comparing a square pixel with a size of 14.5x14.5 µm and a hexagonal pixel with a size of 18x18 µm (**same pixel area**)
- Standard layout is **less efficient** than n-gap layout, for both geometries
- Hexagonal geometry (blue) shows **decrease in efficiency** compared to square
  - This is **unexpected**
  - Less shared charge is expected to lead to higher efficiency
  - Investigations are ongoing into the cause of this...

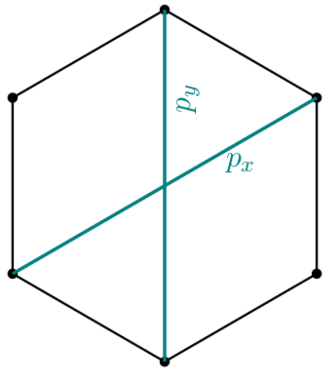


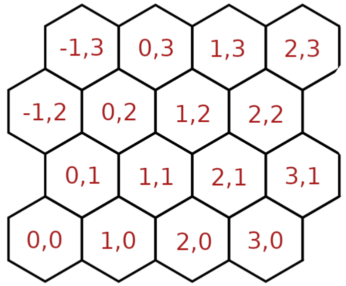
Conclusions and outlook

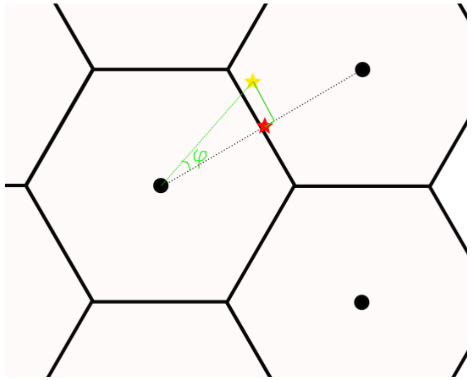
Conclusions and outlook

- **Hexagons work well in Allpix Squared**
  - Can perform both simple and complex simulations using them
  - Only **small alterations** are needed in the configuration files to use hexagons instead of rectangles
  - Results (mostly) match expectations
- Some details of results of the Tangerine studies **need to be understood** - studies are ongoing
  - Larissa (who has done **most of the hexagon work** in Tangerine) starts as a PhD student with us in June - we'll figure things out properly then
  - Hexagons may be a viable geometry for future sensor submissions in the used CMOS imaging technology
  - **Simulations will guide the way!**





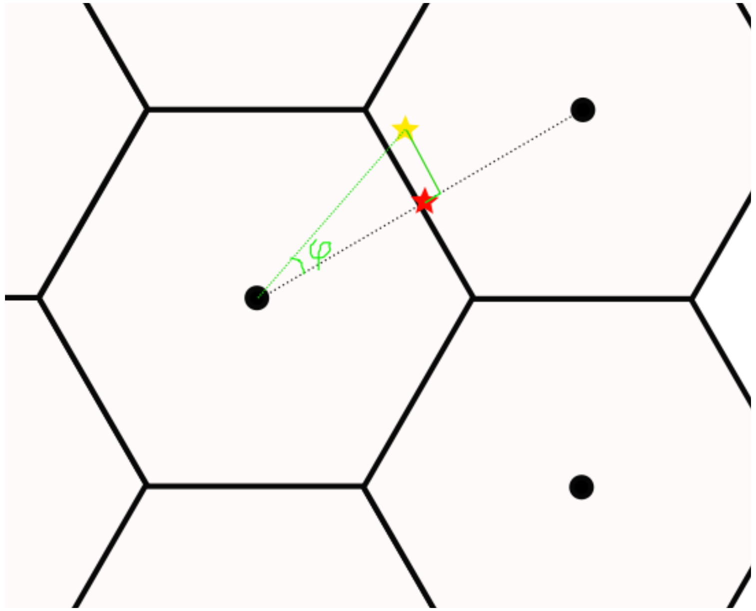




Backup slides

Algorithm for 2-pixel clusters

- Find the pixel in the cluster with the **lowest index,** i.e. the bottom-left pixel of the cluster
  - This is just a choice of what to use as reference position
- The centre of this pixel is the reference position; calculate r and φ of reconstructed position and track/truth position relative to this
- Taking the difference in φ, project the track radial position onto the cluster position radial direction
- Plot the projected track radial position vs the cluster position, and do a fit to get the correction in r
  - There is nothing to be done to correct the φ direction difference for 2-pixel clusters
- To reconstruct, find the reference pixel in the same way, find the cluster position in the r direction, and evaluate the fitted function there and take that value instead



Yellow is track position, red is reconstructed

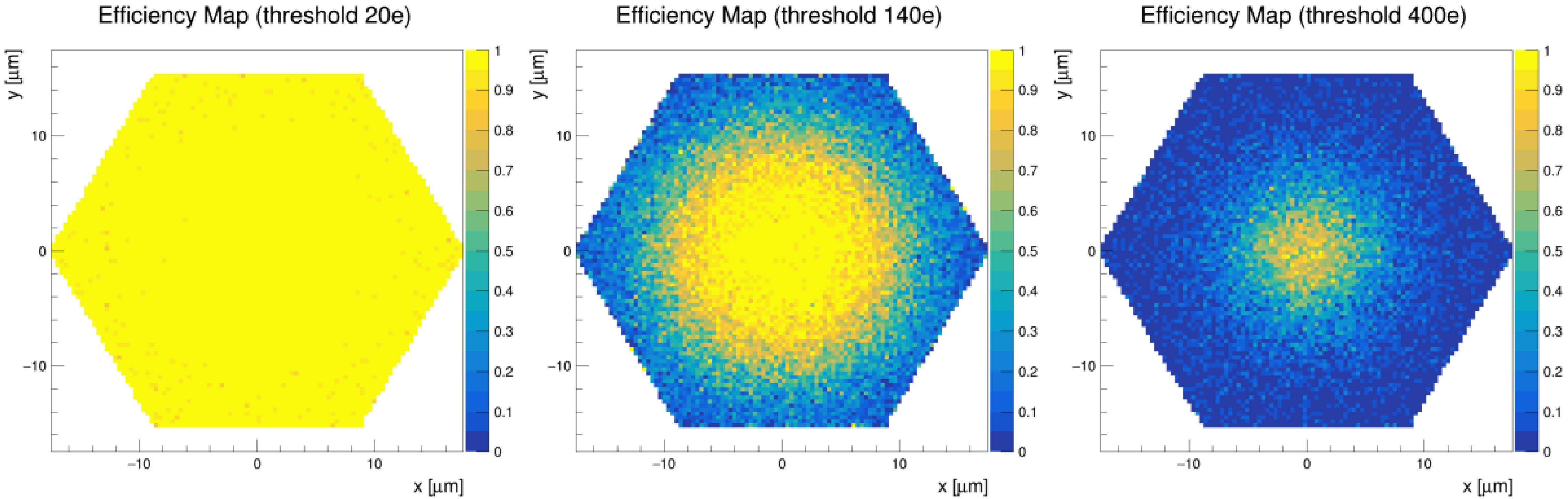
$$\Delta\varphi = \varphi_{\text{track}} - \varphi_{\text{cluster}}$$

$$r_{\text{track, projected}} = r_{\text{track}} \cdot \cos(\Delta\varphi)$$

$$phiDist_{\text{track, projected}} = r_{\text{track}} \cdot \sin(\Delta\varphi)$$

Results - efficiency

- Comparing standard layout and n-gap layout, for a 35x35 μm hexagonal pixel size
- In-pixel efficiency for different thresholds, for the **standard layout**
- Significant efficiency loss already at a threshold of 140 electrons



Results - efficiency

- Comparing standard layout and n-gap layout, for a 35x35 µm hexagonal pixel size
- In-pixel efficiency for different thresholds, for the **n-gap layout**
- Efficiency is maintained even at a threshold of 400 electrons