cern.ch/allpix-squared

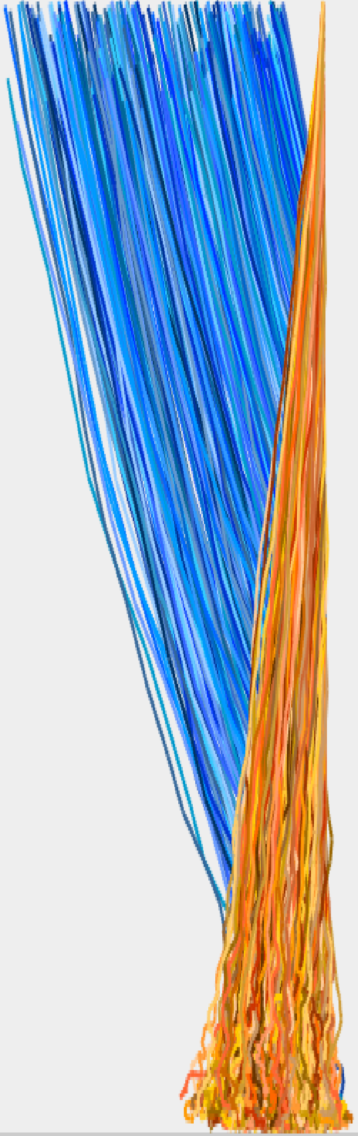
Investigating the resolution of a semiconductor pixel detector, using Allpix Squared

Manuel A. Del Rio Viera
Håkan Wennlöf

25/8 -23

Silicon sensors

- Silicon sensor operation presented this morning
- Signal generation recap:
  - An incident particle **deposits energy** in the sensitive volume, creating electron-hole pairs
  - The electron-hole pairs are **propagated** through the sensor, by diffusion and drift
  - The signal formed by movement of charge carriers is **assigned to a readout channel**
  - The **front-end electronics response** (amplification, threshold, digitisation, ...) finalises the signal for output

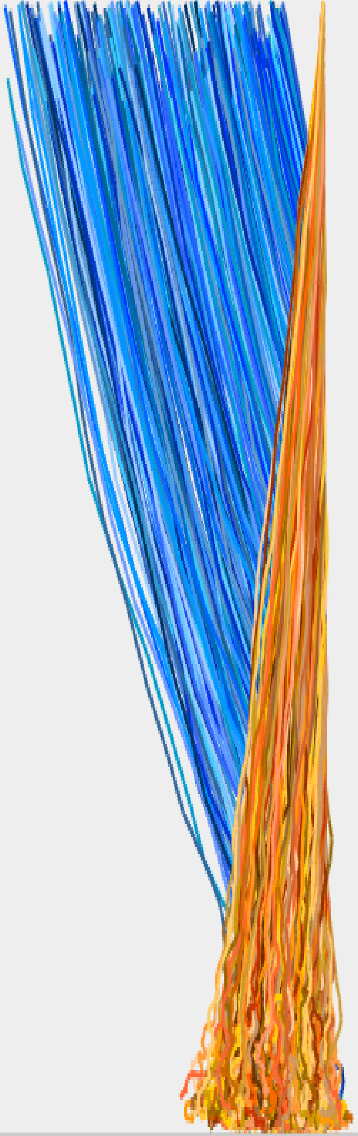


Signal formation

- Sensor operated as diode in reverse bias → depleted volume
- Signal formed by motion of e/h pairs in electric field

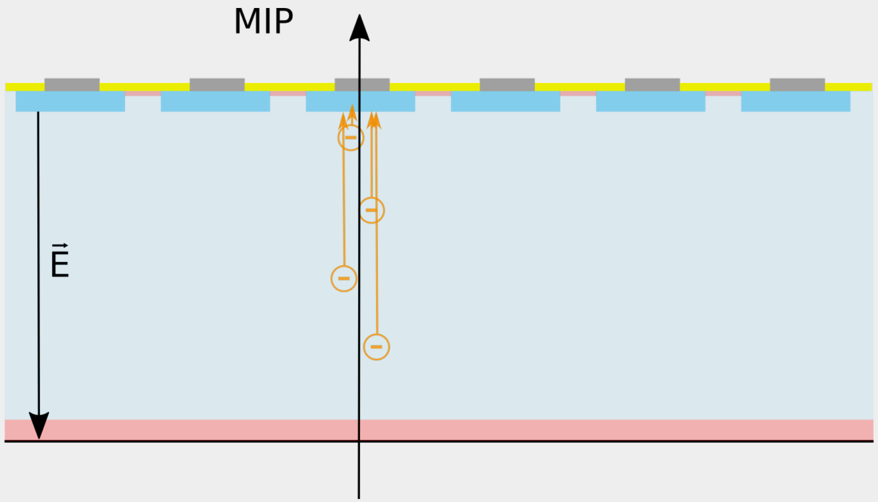
- Contribution to motion:
  - **Diffusion** – Temperature-driven random motion, mean free path ~ 0.1 µm, mean 0
  - **Drift** – Directed motion, depending on electric field and charge carrier mobility,
    different parametrizations for mobility available, depending on temperature, silicon, …

- Motion stops when...
  - Charge carriers reach readout electrode (conductor)
  - Charge carriers recombine/get trapped (depends on purity, doping, lattice defects, …)



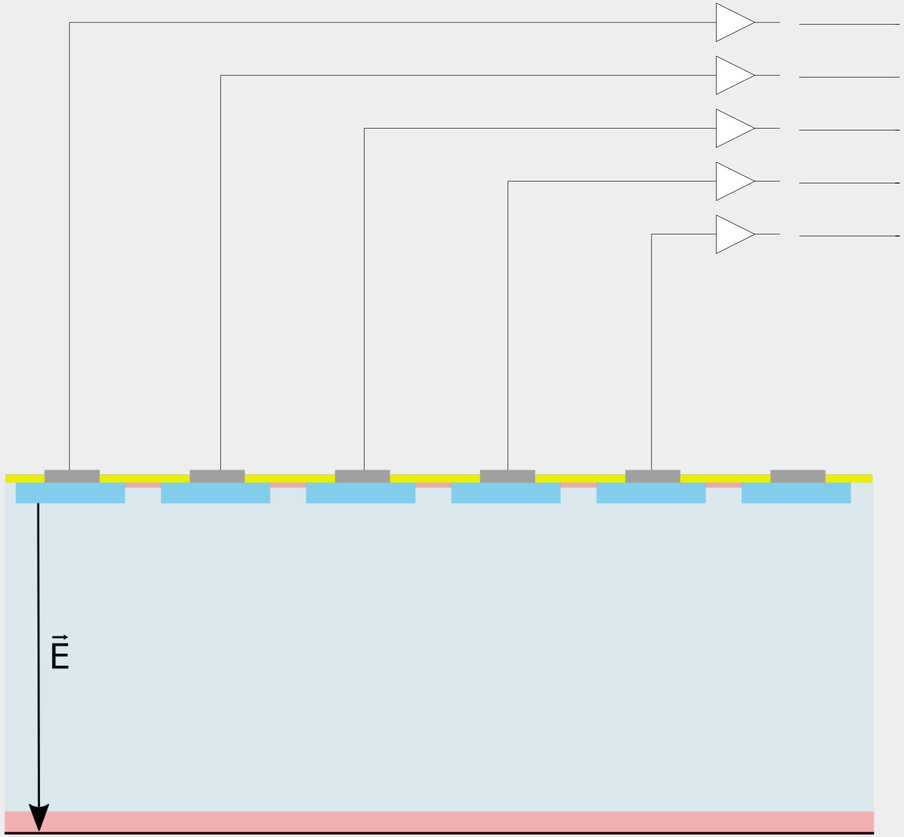
Resolution

- How well can my detector reconstruct the lateral position of a traversing particle?
- **Spatial resolution** ≡ Width of residual
- **Residual** ≡ Distance between reconstructed position and true position



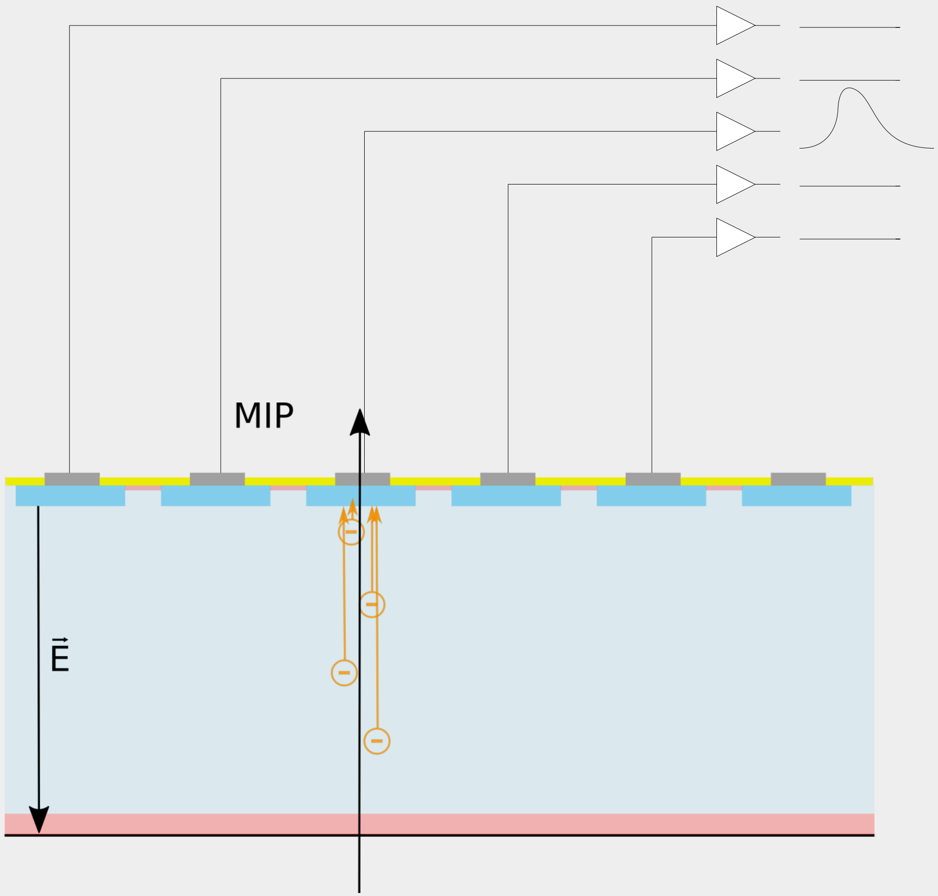
Particle position reconstruction

- Estimation of the lateral position of the particle traversal
- Use information of signal per strip (pixel)



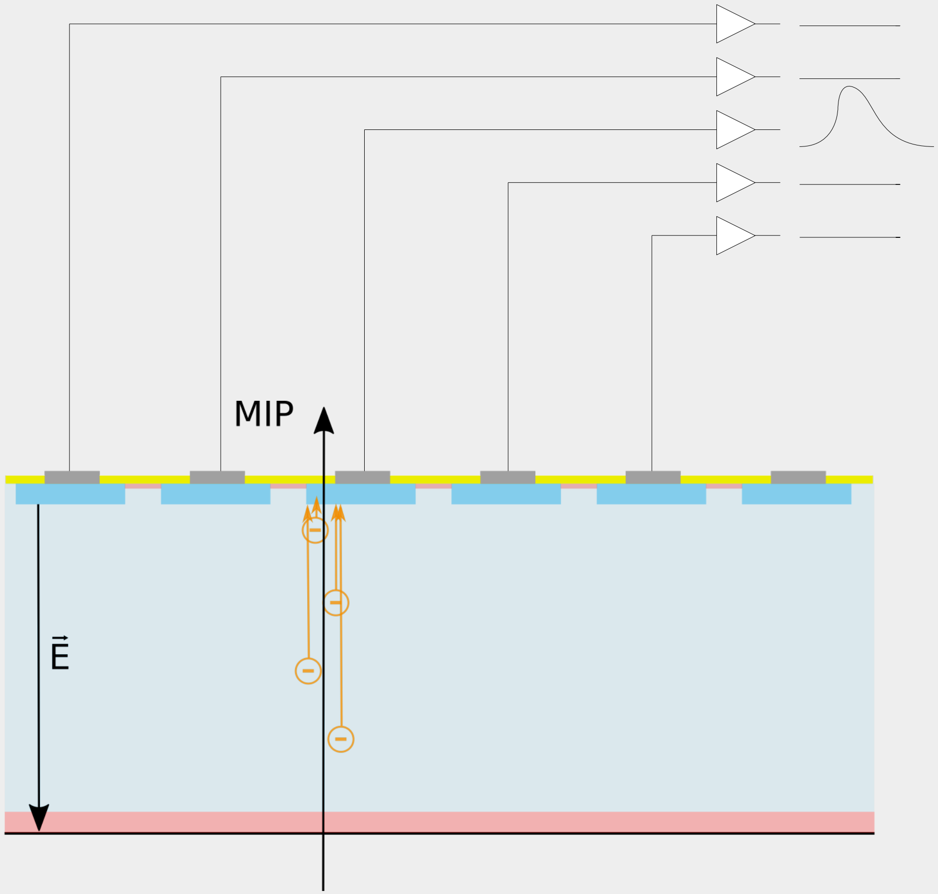
Particle Position Reconstruction

- Single responding pixel:
  - “This pixel was hit”
  - No information of where inside the pixel the particle was located



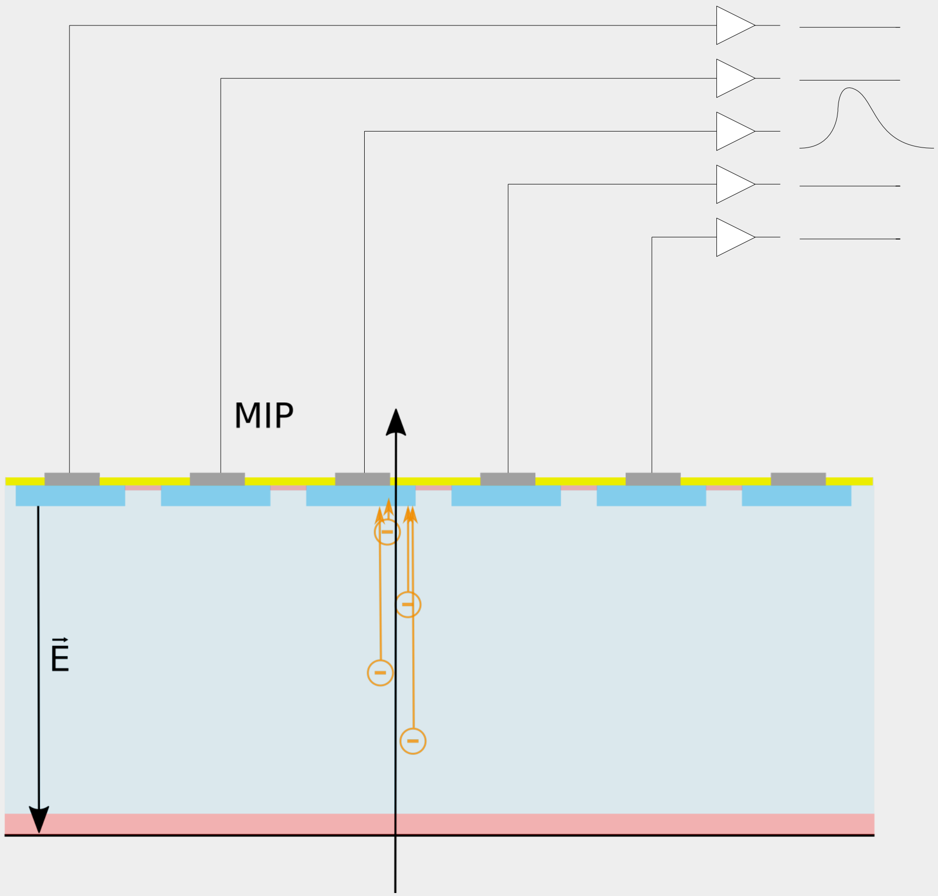
Particle Position Reconstruction

- Single responding pixel:
  - “This pixel was hit”
  - No information of where inside the pixel the particle was located



Particle Position Reconstruction

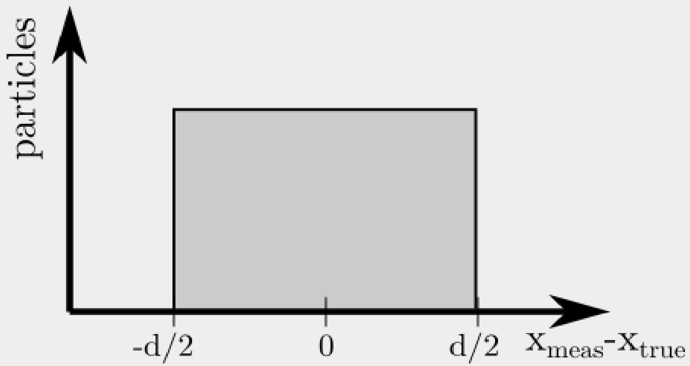
- Single responding pixel:
  - “This pixel was hit”
  - No information of where inside the pixel the particle was located

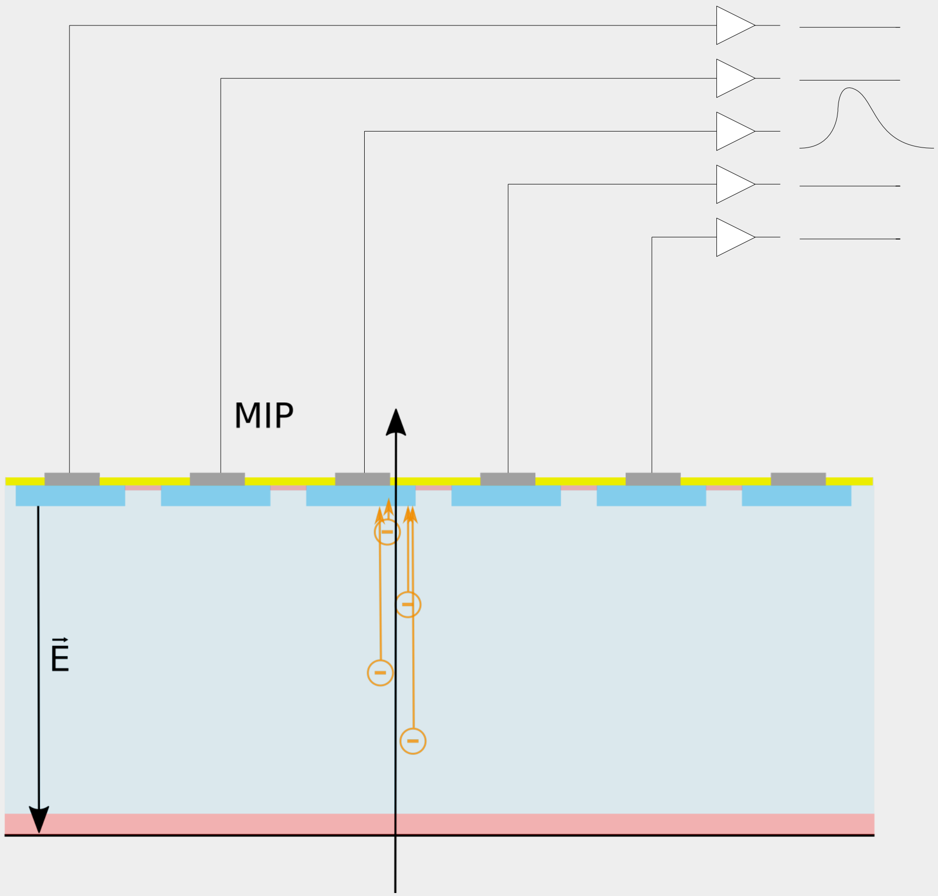


Particle Position Reconstruction

- Single responding pixel:
  - “This pixel was hit”
  - No information of where inside the pixel the particle was located

➔ Resolution: $\sigma = \frac{d}{\sqrt{12}}$

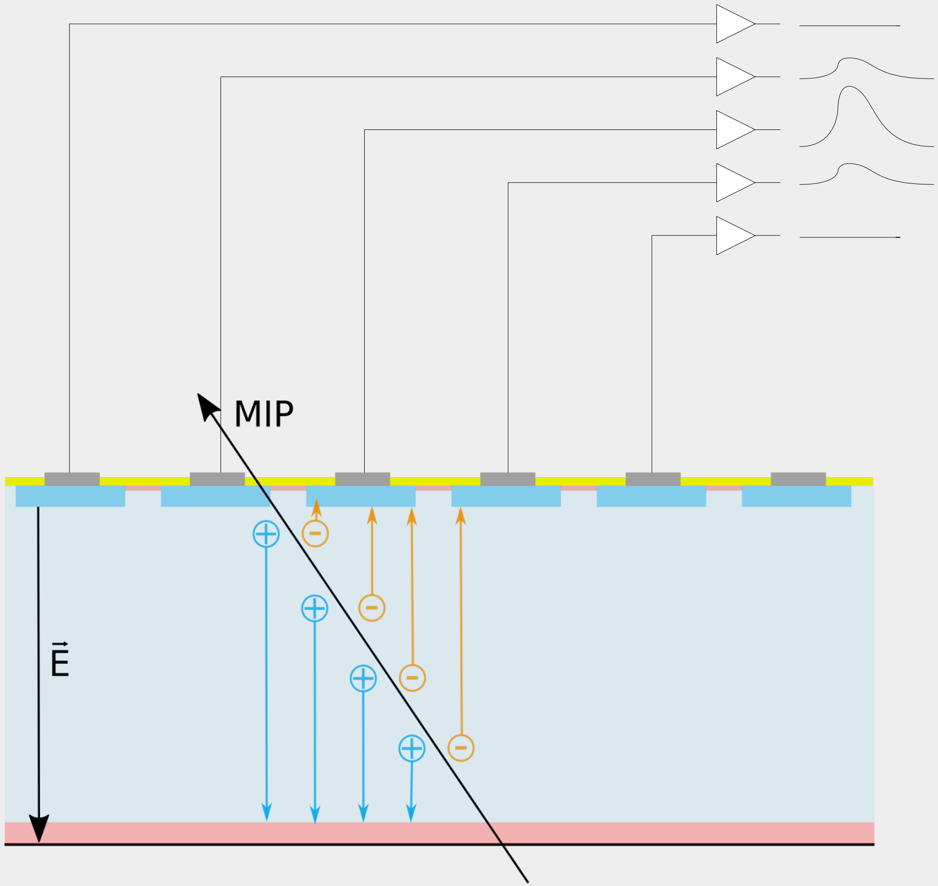




Particle Position Reconstruction

- Several responding pixels:
  a) Calculate center of hit pixels
  b) Calculate **center of gravity** using signal amplitudes of individual pixels

$$x = \frac{\sum_{i=1}^{N} q_i x_i}{\sum_{i=1}^{N} q_i}$$

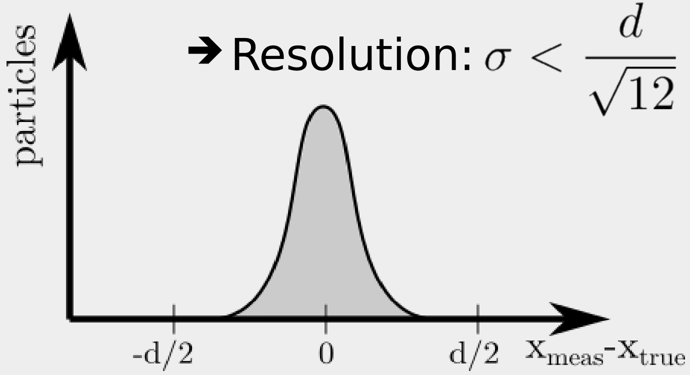


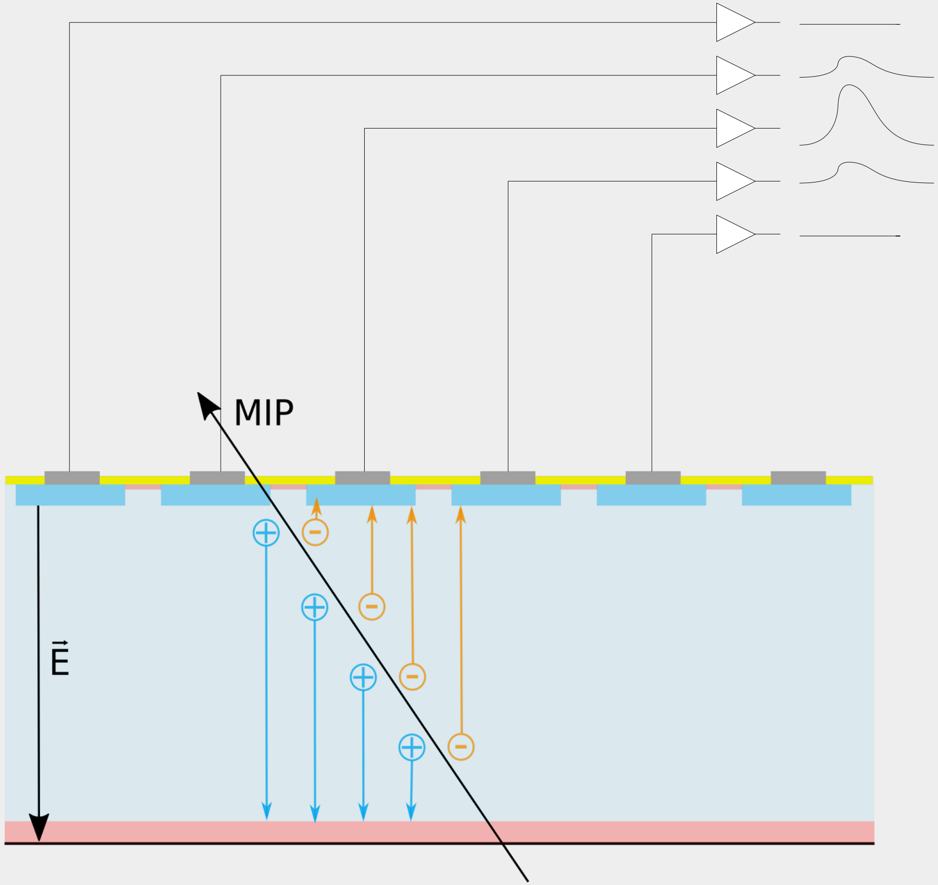
Particle Position Reconstruction

- Several responding pixels:
  a) Calculate center of hit pixels
  b) Calculate **center of gravity** using signal amplitudes of individual pixels

$$x = \frac{\sum_{i=1}^{N} q_i x_i}{\sum_{i=1}^{N} q_i}$$

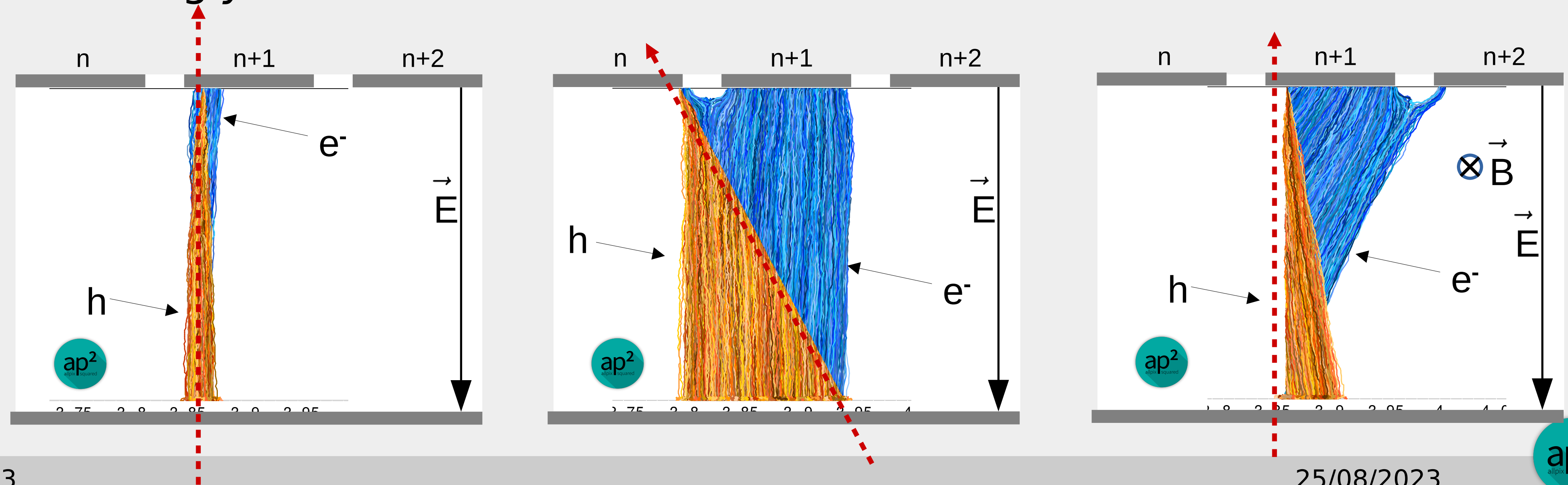
➔ Resolution: $\sigma < \frac{d}{\sqrt{12}}$



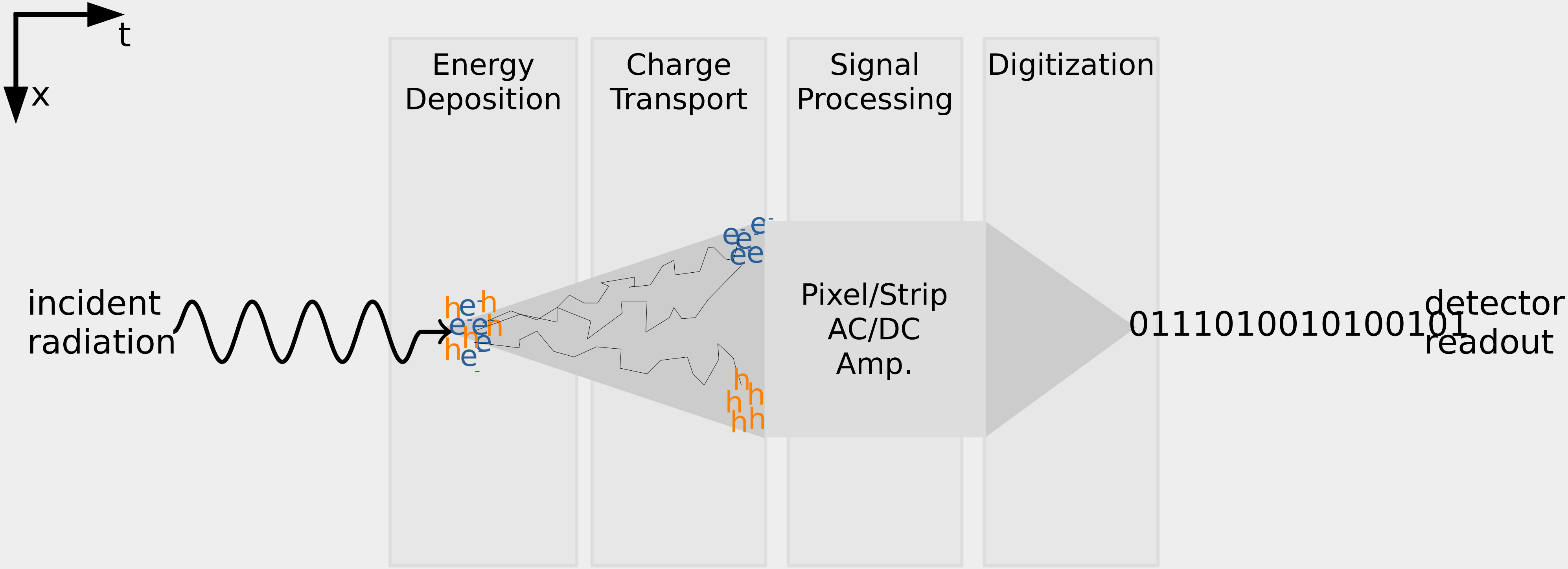


Increasing Charge Sharing – Inclined Tracks & Lorentz Drift – computer lab exercise

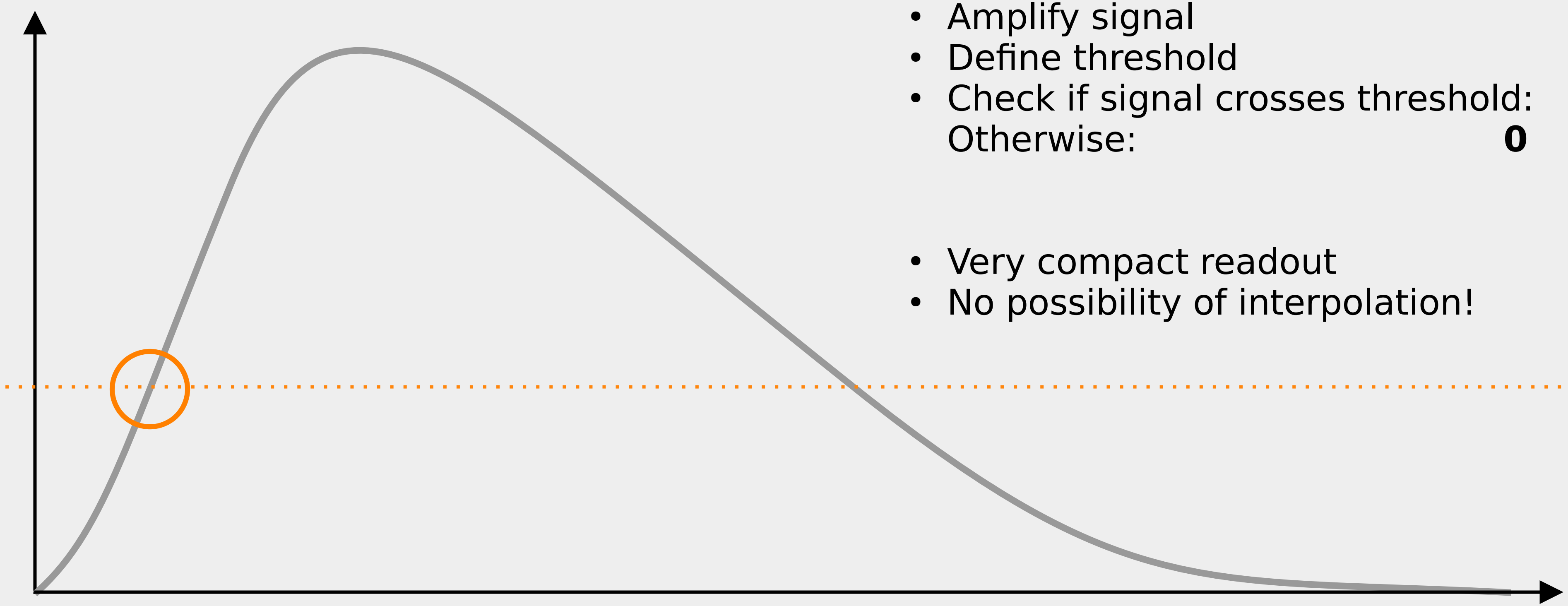
- Charge sharing: distribution of charge carriers / signal over several strips (pixels)
- Can significantly improve the spatial resolution
- Often used: Inclined particle incidence along *x* & Lorentz drift along *y*



Particle Detection with Silicon Detectors



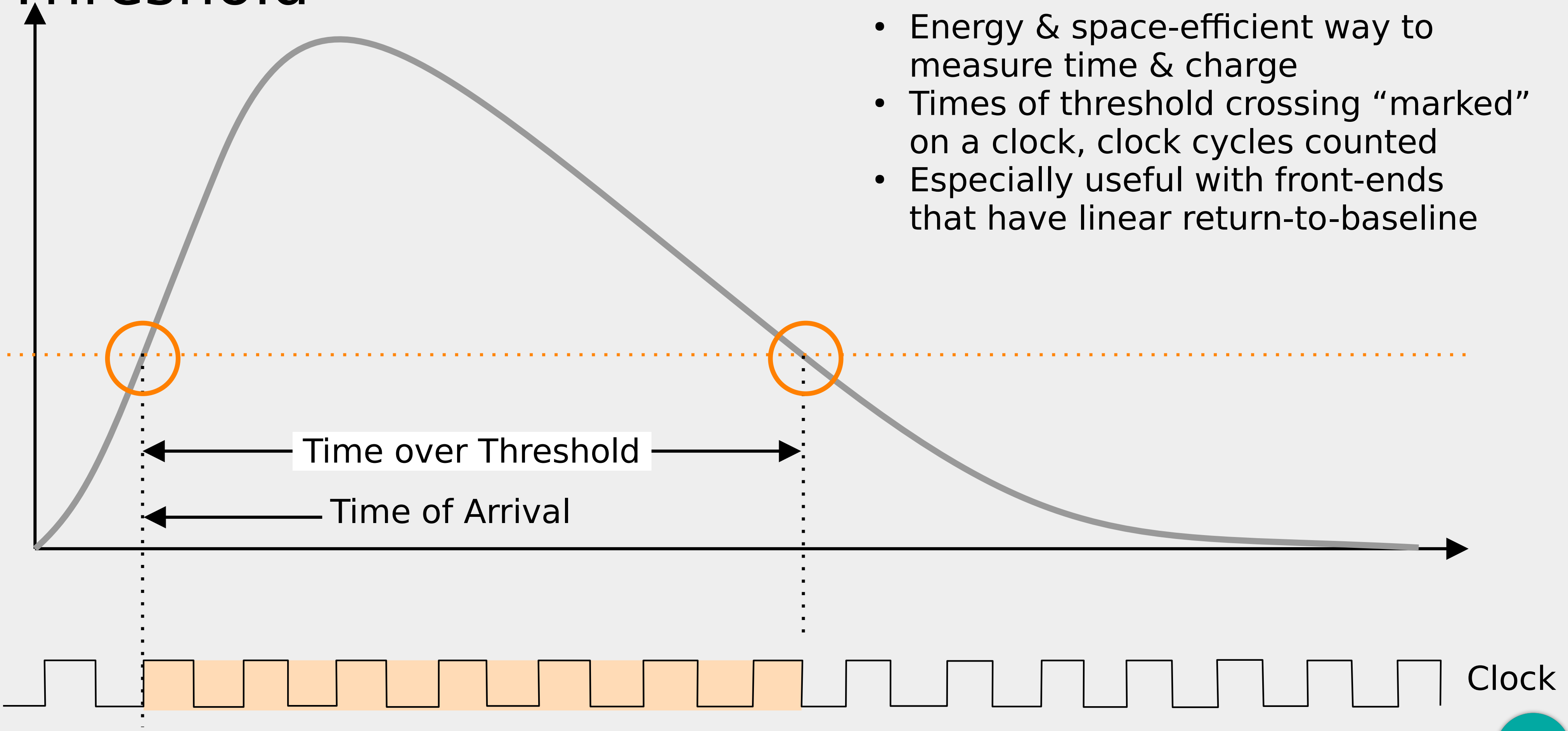
Digitization: Threshold



- Simplest possible measurement: **hit or no hit**
- Amplify signal
- Define threshold
- Check if signal crosses threshold: **1**
  Otherwise: **0**

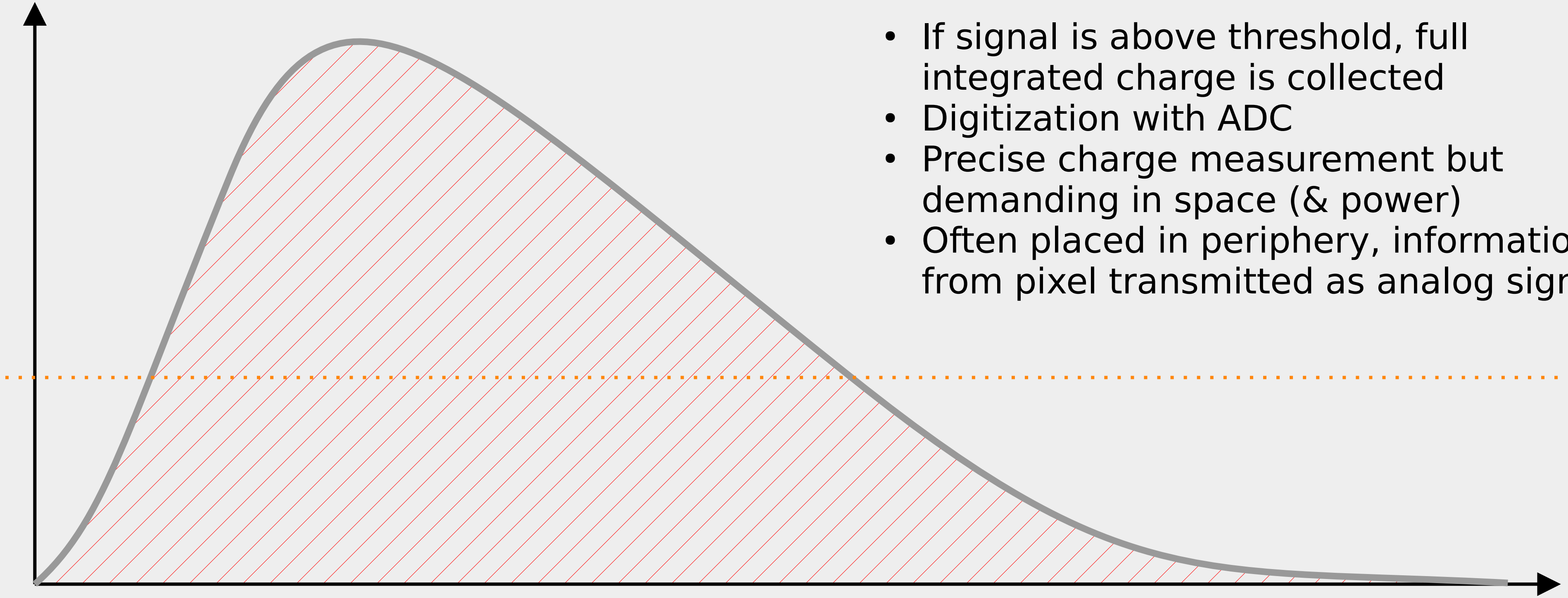
- Very compact readout
- No possibility of interpolation!

Digitization: Time of Arrival & Time over Threshold



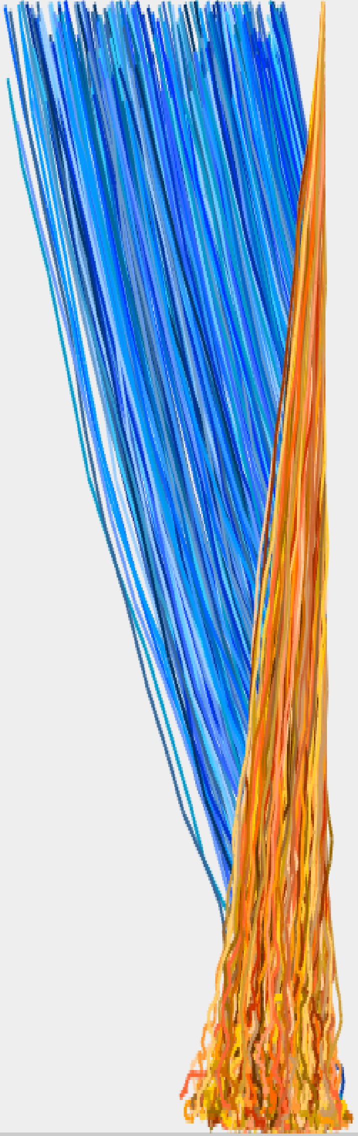
- Energy & space-efficient way to measure time & charge
- Times of threshold crossing “marked” on a clock, clock cycles counted
- Especially useful with front-ends that have linear return-to-baseline

Digitization: Analog-to-Digital Converter



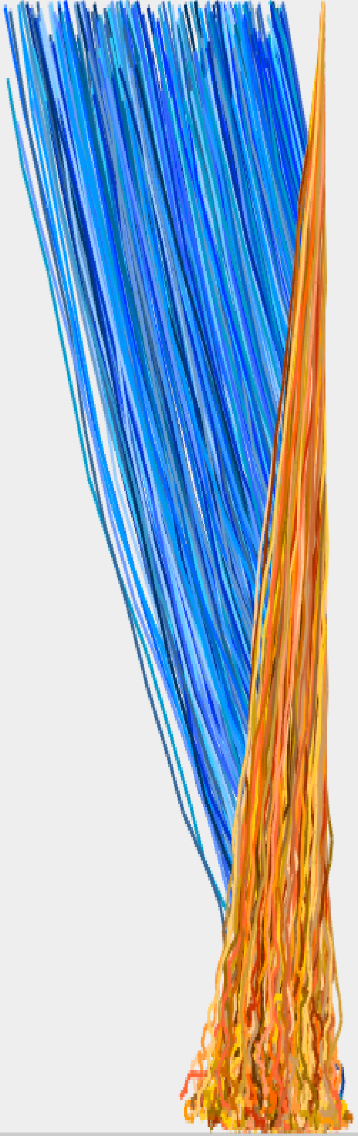
- If signal is above threshold, full integrated charge is collected
- Digitization with ADC
- Precise charge measurement but demanding in space (& power)
- Often placed in periphery, information from pixel transmitted as analog signals

The Allpix Squared simulation framework



Allpix Squared

- **Open-source** simulation framework, based on a **modular** design, written in **modern C++**
- **Complementary** to detailed device simulation; fast, allows for high-statistics samples accounting for stochastic effects in the physics
- Can interface to **Geant4** for energy deposition simulation, **TCAD** for electric fields, and **ROOT** for I/O
- Allows for **Monte Carlo simulations** of pixellated detectors
  - Can simulate the **full detector hit chain;** energy deposition, charge carrier propagation and transfer, and digitisation
  - Most modules can be run **multithreaded**
- Multiple-detector setups can be simulated, giving realistic simulations of test beam applications
  - Can extract efficiency, cluster size, resolutions, ...



Modular?

- Each stage in the simulation is performed by **exchangeable modules**
- Example simulation flow between modules:

Geometry definition → **Electric field and doping definition** → **Energy deposition** → **Charge propagation** → **Charge transfer** → **Digitisation**

Geometry definition
Build the required geometry in the framework

Electric field and doping definition
Describe or import the electric field map and doping of sensors

Energy deposition
A simulated particle deposits energy in sensors, creating electron-hole pairs

Charge propagation
The electron-hole pairs are propagated through the sensor bulk

Charge transfer
Charge is transferred from bulk to collection electrode

Digitisation
Describe the digitisation of charge by the readout electronics

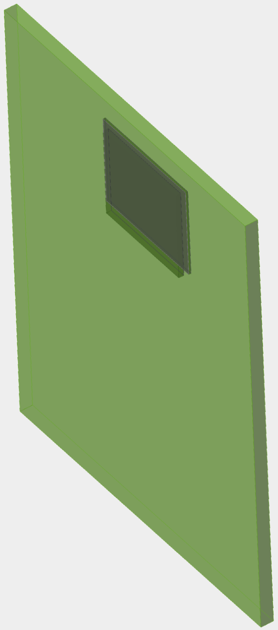
Modular!

- Modules are loaded in order in a plain-text configuration file
- No advanced coding needed to set up, run, and do basic analysis of a simulation
- Module parameters can be **easily tweaked**, and modules **exchanged**
- Simulation data can also be output in a number of formats
  - LCIO, RCE, Corryvreckan, plain text, Allpix² ROOTObjects (allowing simulations to be performed in parts)

```
[Allpix]
detectors_file = "detector.conf"
multithreading = true
number_of_events = 100000

[GeometryBuilderGeant4]

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

[ElectricFieldReader]
model = "mesh"
file_name = "../fields/Gap_20_ElectricField.apf"
field_offset = 0.5,0.5

[DopingProfileReader]
model = "mesh"
file_name = "../fields/Gap_20_DopingConcentration.apf"
doping_depth = 50um

[GenericPropagation]
temperature = 293K
mobility_model = "jacoboni"
recombination_model = "srh_auger"
propagate_electrons = true

[PulseTransfer]

[DefaultDigitizer]
electronics_noise = 10e
threshold = 200e
threshold_smearing = 5e

[DetectorHistogrammer]

[ROOTObjectWriter]
file_name = "output_ngap_20x20_500000.root"
```

Geometry definition

- Geometry is defined via a flexible and human-readable format
- **Hybrid** and **monolithic** sensors are supported, as well as **passive materials**
- Several sensors can be easily combined with different orientations and positions, forming **detector systems**
- Passive material can also be imported via GDML files created in other software

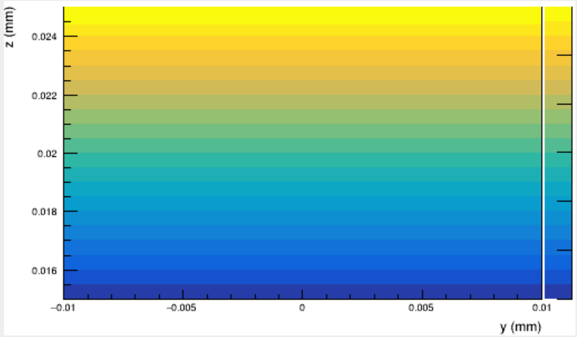


Example: Timepix3 assembly

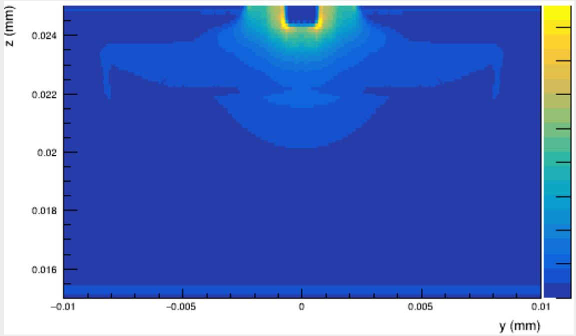
```
type = "hybrid"

number_of_pixels = 256 256
pixel_size = 55um 55um

sensor_thickness = 300um
sensor_excess = 1mm

bump_sphere_radius = 9.0um
bump_cylinder_radius = 7.0um
bump_height = 20.0um

chip_thickness = 700um
chip_excess_left = 15um
chip_excess_right = 15um
chip_excess_bottom = 2040um

[support]
thickness = 1.76mm
size = 47mm 79mm
offset = 0 -22.25mm
```

Electric field and doping profile definition

- Electric fields and doping concentrations can be defined within Allpix², or imported from TCAD simulations
  - Electric fields defined as constant, linear, parabolic, or an arbitrary function
  - Doping concentrations defined as constant or as different regions
- By using the built-in methods of defining fields, **quick approximative simulations** can be made
- By using the import from TCAD, **high-statistics simulations can be made with very accurate models** even for highly non-linear configurations



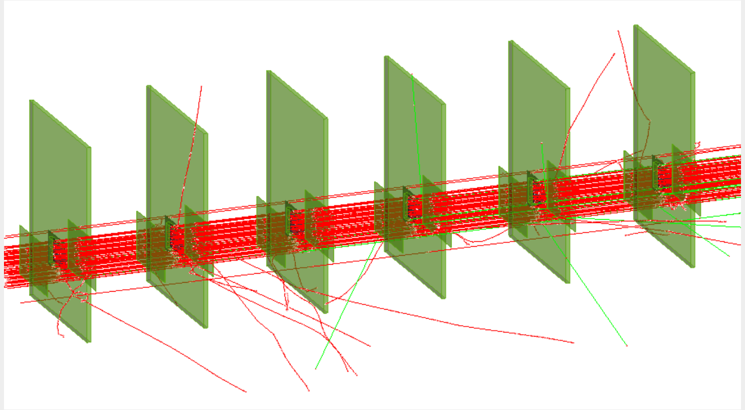
Magnitude of linear field model



Magnitude of TCAD field in Allpix²

Energy deposition

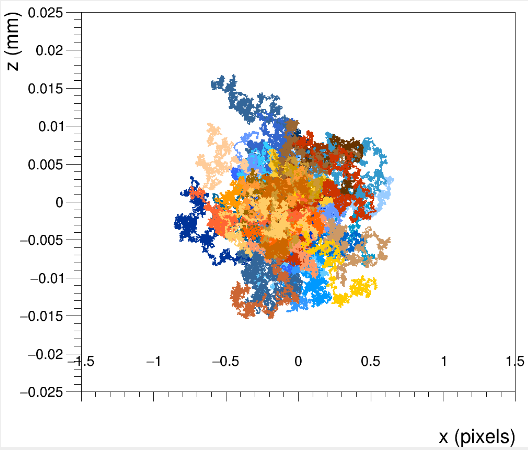
- Energy deposition can be made at a point, along a line, via an interface to **Geant4**, or via a cosmic ray model
- Deposited energy can also be read in from an external file created by other software
- The user can select particle sources, physics lists to be used, and different physics cuts
  - Particle type, energy, direction, ...
  - Common source types are directly available, complex sources can be defined via Geant4 macros
- Energy deposit and scattering is simulated in all material present, and deposited energy in the sensitive volume creates **electron-hole pairs**



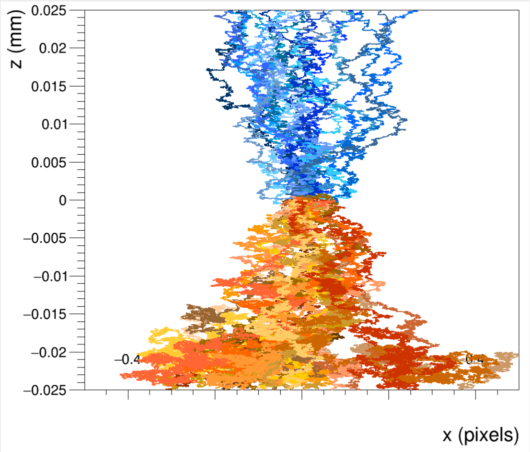
Simulation of a 5 GeV electron beam shot through six EUDET telescope planes

Charge propagation

- Electron-hole pairs get **individually propagated** through the sensor, and collected at the backside or collection electrodes
- Propagation simulation includes both **diffusion** and **drift in electric fields**
- Physics models adjustable; Carrier mobility, lifetime, trapping time
- Using weighting field information, the **transient** charge information on collection electrodes can also be available
    - Accurate **charge pulses** can thus be simulated



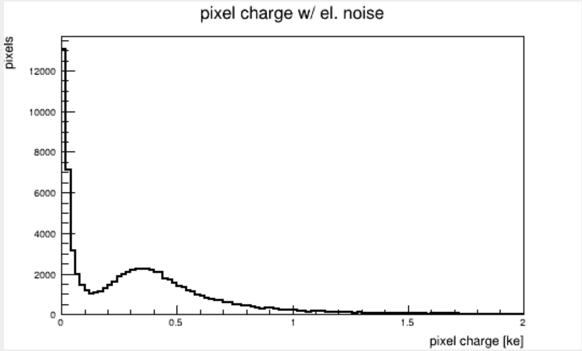
Zero electric field – diffusion dominates

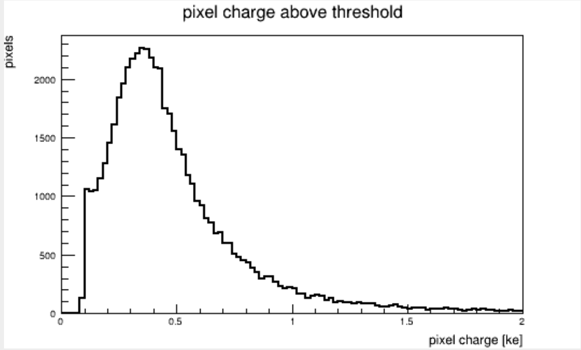


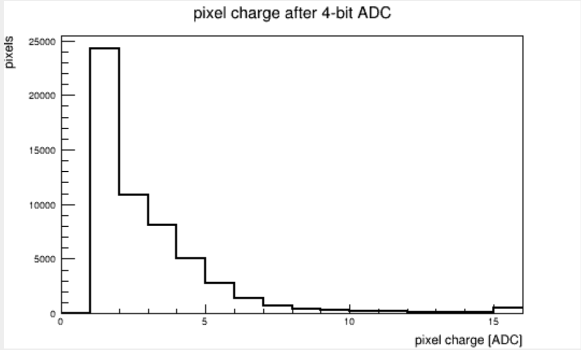
Linear electric field – drift dominates, e-h separated

Digitisation

- Readout electronics behaviour can be simulated by different parts:
  - Electronics noise
  - Amplifier gain and smearing
  - Thresholds and threshold dispersion
  - ADC and TDC resolution and smearing
  - Saturation effects
- An implementation of a charge-sensitive amplifier with Krummenacher type feedback is available
- Sensor-specific front-end modules can be developed (ongoing for example for the Timepix3 chip)

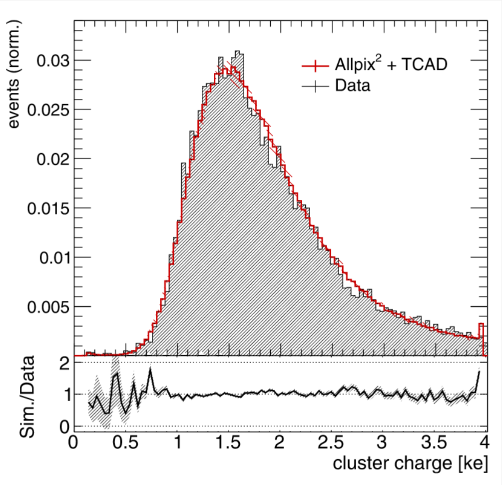


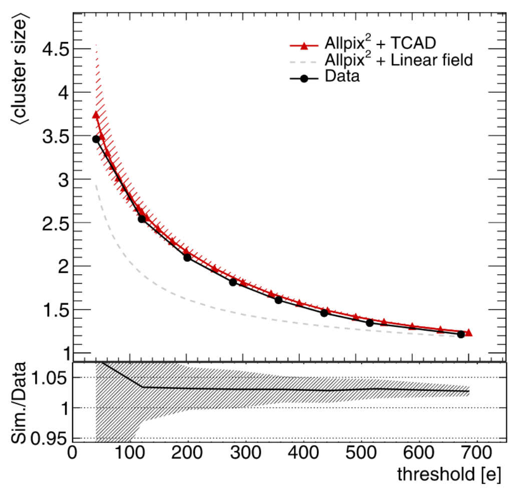




Validations against data

- Combining detailed TCAD electric field simulations with Allpix Squared Monte Carlo simulations provides a realistic high-statistics data sample that can be directly compared to **experiments** — rather: data sample that can be **directly compared to experiments**
- Can show whether the models used are good or need tweaking, thus improving the simulation accuracy
- Validation is made whenever possible, e.g. in these papers;
  - https://doi.org/10.1016/j.nima.2018.06.020
  - https://doi.org/10.1016/j.nima.2020.163784
  - http://cds.cern.ch/record/2801189
- Simulations show a **good agreement** with data





Images from https://doi.org/10.1016/j.nima.2020.163784

Lab tasks using Allpix Squared

Lab tasks using Allpix Squared

- Through the tasks, you will:
  - Learn how to use the framework (it's relatively straightforward)
  - Learn to extract detector resolution from data
  - Investigate how different **digitisation**, sensor **rotation**, and external **magnetic fields** affect single-sensor spatial resolution
  - Observe charge carrier transport in a silicon sensor
- Exercises run in parallel sessions next week

Allpix Squared resources

Website
https://cern.ch/allpix-squared

Repository
https://gitlab.cern.ch/allpix-squared/allpix-squared

Mattermost channel

https://mattermost.web.cern.ch/allpix2

User Forum:

https://cern.ch/allpix-squared-forum/

Mailing Lists:

allpix-squared-users https://e-groups.cern.ch/e-groups/Egroup.do?egroupId=10262858

allpix-squared-developers https://e-groups.cern.ch/e-groups/Egroup.do?egroupId=10273730

User Manual:
https://cern.ch/allpix-squared/usermanual/allpix-manual.pdf

Testbeam data analysis using the Corryvreckan framework

Manuel A. Del Rio Viera
Håkan Wennlöf

25/8 -23

Motivation

- New sensor prototypes need to be **characterised** and **understood** in detail
  - Submitted design may not match what really comes out, in the end (errors, quirks, fabrication issues, ...)
- To use a sensor in an experiment, it must first be thoroughly tested
- Prototypes thus undergo large **characterisation campaigns**

Sensor characterisation

- Typical sensor lab tests include:
  - biasing tests (current vs. high-voltage characteristics)
  - configuration tests
  - optimisation of sensor settings (amplifier, voltages, etc.)
  - measurement of the power consumption
  - measurement of the noise rate
  - energy calibration with radioactive sources or X-ray tubes (photons with well-known energies)
- Not everything can be determined by lab tests, however!

Sensor characterisation method – test beams

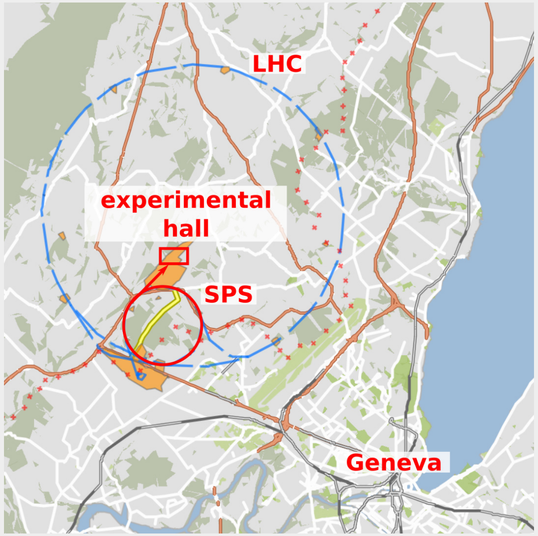
- Providing a **reference measurement** to determine further sensor characteristics
  - Tracking the passage of a particle through the sensors
- Allows for
  - spatial and time resolution
  - hit detection efficiency
  - detailed studies, such as hit detection efficiency across pixel matrix or within pixels
  - comparison and optimisation of different operation parameters
  - system integration tests, such as simultaneous operation and readout of many sensors in parallel or multiple subsystems

Test beam facilities

- Large-scale scientific user facilities, which allow **relativistic particle beams** to traverse a sample
- Examples:
  - DESY in Hamburg, Germany
  - CERN near Geneva, Switzerland
  - PSI near Zurich, Switzerland

Example: the SPS at CERN

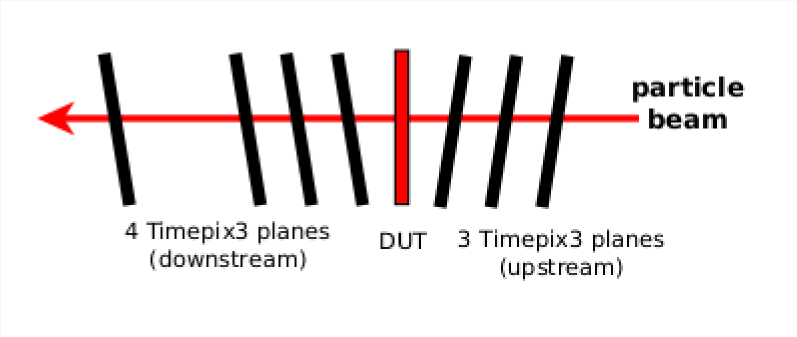
- Super Proton Synchrotron at CERN
- ~7 km circumference
- Beam of 120 GeV hadrons (mainly pions)
- Data for this exercise taken here

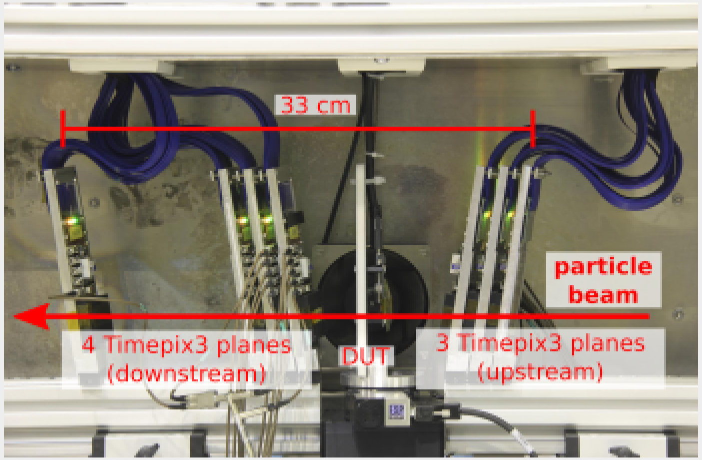


Beam, and reference telescope

Reference telescope

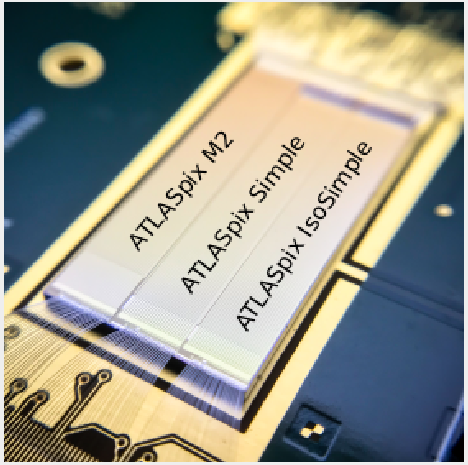
- Used to reconstruct a particle track
- Commonly silicon sensors
- This exercise: Timepix3-based telescope
  - Pixellated
  - Data recorded: pixel address, timestamp, and time-over-threshold
  - Tracking resolution of 1-2 µm
- Device-under-test (**DUT**) in the middle
  - This is the investigated sensor

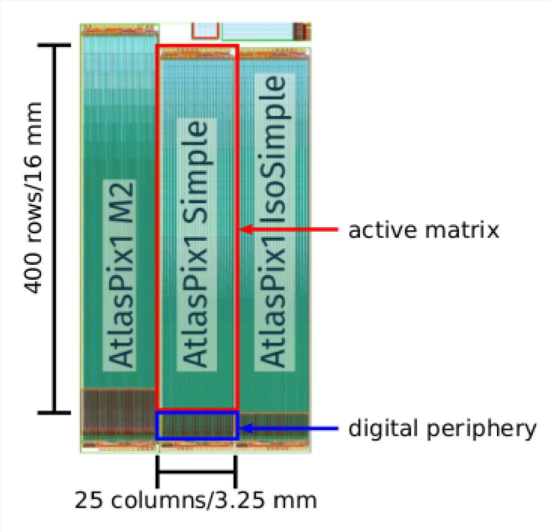




The device-under-test

- In this exercise, looking at an *ATLASpix_Simple*
  - High-voltage monolithic active pixel sensor (HV-MAPS)
  - 25 columns and 400 rows of pixels
  - Pixel size of 130x40 µm
  - Data recorded: pixel address, timestamp, and time-over-threshold





Test beam analysis flow (the exercise)

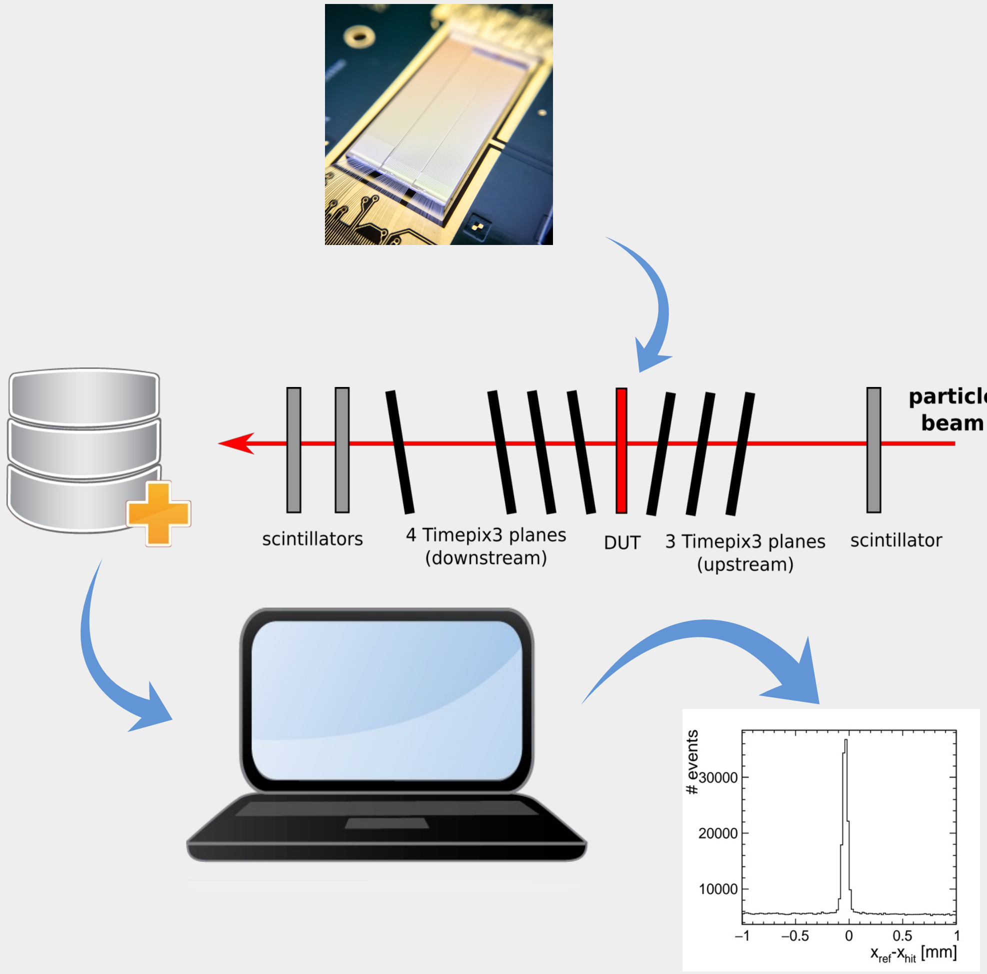
- Using Corryvreckan to extract observables from real test beam data, and ROOT to plot things
- Works through several steps:
  - **Reading in** test beam data
  - Performing **clustering**
  - Checking **correlations**
  - Performing detector **alignment**
  - Performing **tracking** and cluster **association**
  - **Analysing** the detector performance

The exercise

- This lab course is a pure analysis project, with real data recorded at the Super Proton Synchrotron (SPS) at CERN, Switzerland.

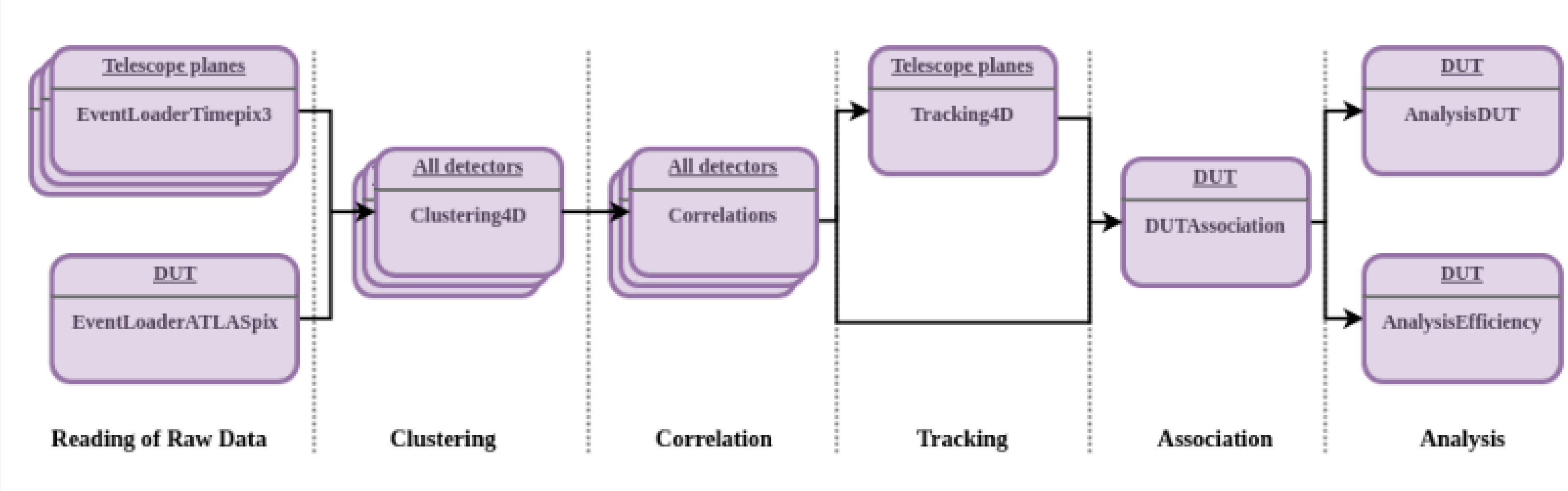
  The main focus of the tutorial:

  - Understand the working principle of silicon pixel detectors
  - Analyse a set of test-beam data in order to characterise a pixel sensor prototype and investigate its performance



Corryvreckan

- A **reconstruction and analysis tool** for pixel sensor testbeam data
- Modular structure (similar to Allpix Squared)
- Highly flexible and configurable
- Relatively clean code in modern C++



Configuration of Corry

- Two files needed: main **configuration** and **geometry**
- Configuration:
  - Global parameters
  - Specifies modules
  - Sets input and output paths
- Geometry:
  - Defines detectors and their positions
  - Characteristics and roles

```
[Corryvreckan]
log_level = "WARNING"
detectors_file = "geometry.conf"
number_of_tracks = 900000

[EventLoaderEUDAQ2]
file_name = "data/run0000456.raw"
inclusive = false
buffer_depth = 1000
shift_triggers = -1
```

```
[W0013_D04]
number_of_pixels = 256, 256
orientation = 10.9deg, 17.2deg, -1.3deg
orientation_mode = "xyz"
pixel_pitch = 55um, 55um
position = 886.5um, 270um, 0
spatial_resolution = 4.8um,4.8um
time_resolution = 1.56ns
type = "Timepix3"
role = "reference"
```

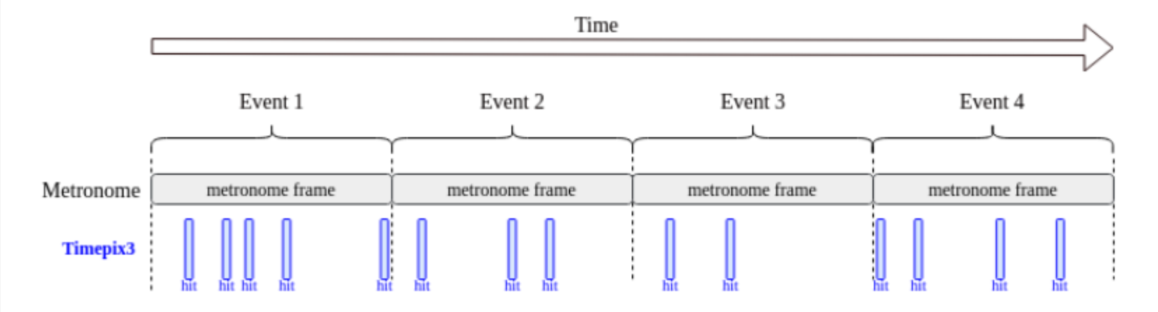
Setting up geometry

- Defining all detectors in the geometry, with unique names within square brackets […]
- Need one detector to have the role of “reference”, and the DUT should have the role “DUT”
- Each sensor should have its number of pixels, pixel size, resolution, and spatial position defined

```
...
[Timepix3_2]
type = "Timepix3"
number_of_pixels = 256,256
pixel_pitch = 55um,55um
position = 0mm,0mm,43.5mm
orientation = 9deg,189deg,0deg
orientation_mode = "xyz"
spatial_resolution = 4um, 4um
time_resolution = 1.56ns
role = "reference"

[ATLASpix_0]
type = "ATLASpix"
number_of_pixels = 25,400
pixel_pitch = 130um,40um
position = 0mm,0mm,105mm
orientation = 0deg,0deg,0deg
orientation_mode = "xyz"
spatial_resolution = 37.5um, 11.5um
time_resolution = 16ns
role="DUT"
...
```

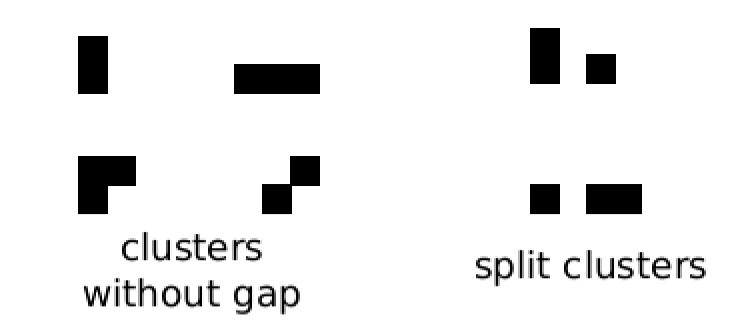
Reading of data

- Done by [EventLoader...] modules (special for each detector)
- Reads in raw data into Corryvreckan, for different sensor types
- In this exercise, each sensor provides an address (column and row of the pixel hit), a timestamp, and the time-over-threshold value
  - ToT is proportional to the signal amplitude
- Events are defined using “time-slices” in these data
  - Each slice has a length of 20 µs, and hits with timestamps within the slice are used
  - Defined using the [Metronome] module



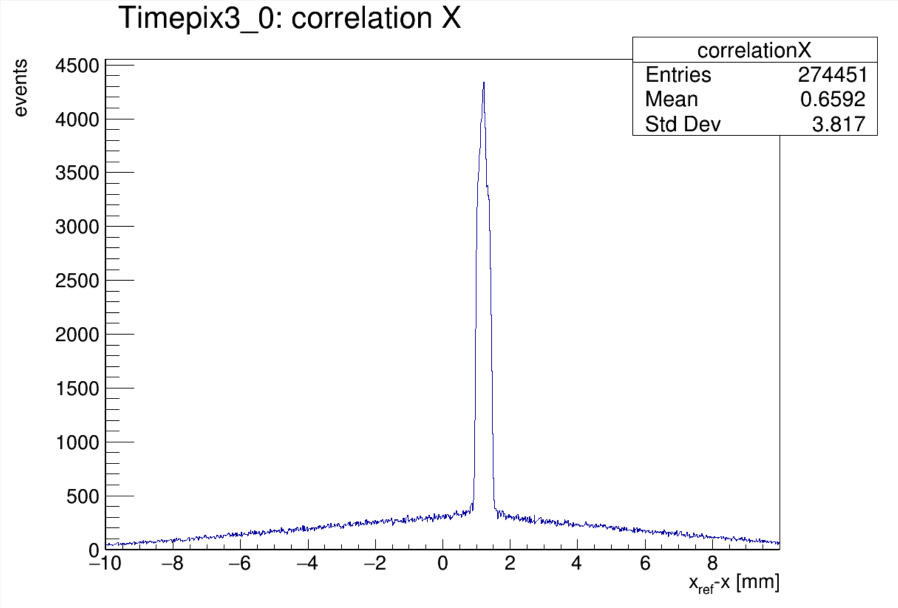
Clustering

- Handled by the [Clustering4D] module
- Creates clusters from **adjacent pixel hits**
- Creates histograms of cluster properties
  - Size
  - Charge
  - Seed charge (charge of the pixel with the largest signal in the cluster)
    - Note: charges in units of ToT: uncalibrated



Correlations

- Normally one of the first things checked at a test beam
- Gives an idea of data quality
- Checks cluster positions on sensors, compared to a reference detector (i.e. detector with role = “reference”)
- Gives an idea of detector positions in relation to the reference



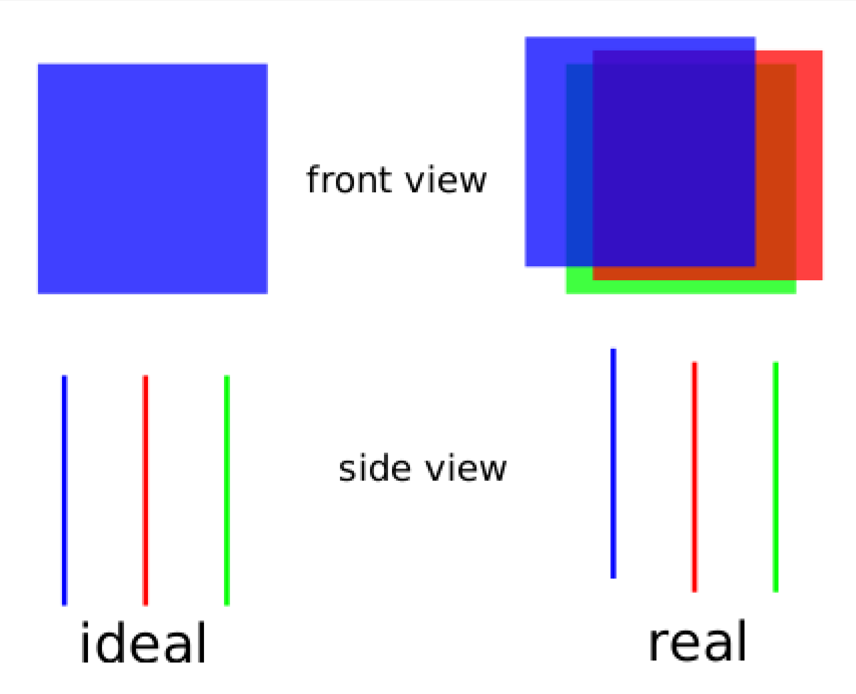
If centered on zero: aligned
Width of peak: gives idea of rotations

$$x_{\text{correlation}} = x_{\text{cluster on \textbf{reference} detector}} - x_{\text{cluster on \textbf{this} detector}}$$

$$y_{\text{correlation}} = y_{\text{cluster on \textbf{reference} detector}} - y_{\text{cluster on \textbf{this} detector}}$$

Alignment

- Impossible to perfectly place detectors (reminder: pixel size is of the order of **tens of µm**)
- Need to measure it in the z-direction (along the beam), but **using correlations** Corryvreckan can do a first alignment in the other directions and rotation
- Finer alignment is done using **tracks**, once available (minimising the error on the tracks)



Tracking

- Done using the [Tracking4D] module
- Uses the **cluster information to fit tracks** through the detectors
    - Normally using only the telescope, excluding the DUT so as not to bias the DUT analysis
- For this exercise (thin sensors in the SPS, straight lines are used to fit the tracks)
- After tracks have been fit, the tracks are associated to hits on the DUT via the [DUTAssociation] module

DUT analysis

- When we have tracks, we can determine spatial and temporal resolutions, and efficiency of the DUT
- Spatial resolution:

$$x_{\text{residual}} = x_{\text{track intercept}} - x_{\text{associated cluster}}$$

$$y_{\text{residual}} = y_{\text{track intercept}} - y_{\text{associated cluster}}$$

  - The resolution is taken as the root mean square (RMS) of these distributions, for many events
  - Note: the result contains the **combined resolution of the telescope and the DUT**. However, telescope resolution is relatively small in this case
- Time resolution taken in a similar way, using track and DUT associated cluster timestamps

DUT analysis

- Efficiency is the **probability of detecting a particle traversing the sensor**
- Defined as the number of tracks through the sensor that produce a signal divided by the total number of tracks traversing the sensor
  - Ideally, we detect them all, i.e. efficiency = 100%
- We can find **global efficiency**, and (if tracking is good) **in-pixel efficiency**

$$\epsilon = \frac{\#\ \text{tracks}\ \textbf{with}\ \text{an associated cluster}}{\#\ \text{tracks}\ \textbf{with} + \textbf{without}\ \text{an associated cluster}}$$

Summary

- We will work through a **full testbeam analysis chain** in the exercise
  - Starting from raw data taken at the CERN SPS, and extracting DUT final observables
- The goal is to learn how testbeam analysis is usually carried out, and how to use the Corryvreckan framework
  - Note: please do not infer actual sensor performance from this exercise, as it is just meant to be illustrative
- This exercise takes place **this afternoon** in the computer room
  - Please pair up, groups of two is best

Backup slides

Development of Allpix Squared

- Constantly ongoing – current release version is **2.2.0**
- Based on GitLab and its tools
  - Allows for issue tracking, merge requests, and automatic code tests
- Continuous integration and deployment (CI/CD)
  - Every commit is automatically checked and corrected for formatting, and tested and reviewed before merging
  - Several compilers and platforms are checked
  - Simulations with known outcomes are automatically run, to check for differences and performance
  - This keeps the code clean, coherent, readable, and **functional**
- Contributions are very welcome! Instructions are available at https://gitlab.cern.ch/allpix-squared/allpix-squared