Accurate Electron Reconstruction in TPCs with a Neural Network

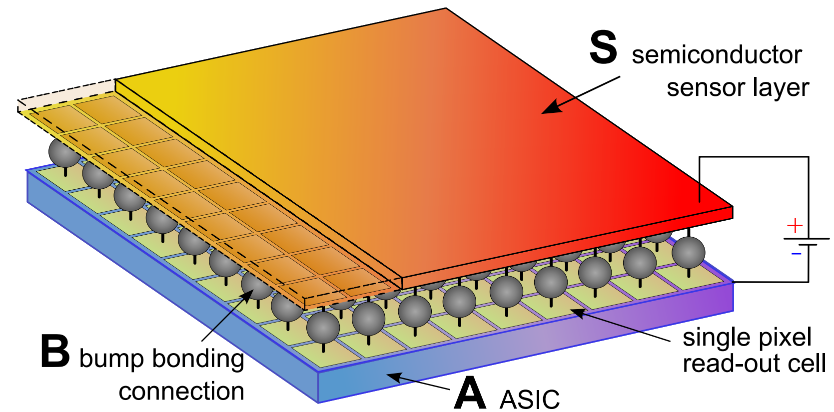
ERLANGEN CENTRE FOR ASTROPARTICLE PHYSICS

Thomas Gleixner*, Mykhaylo Filipenko,
Thilo Michel, Gisela Anton
CERN, 20.3.2015

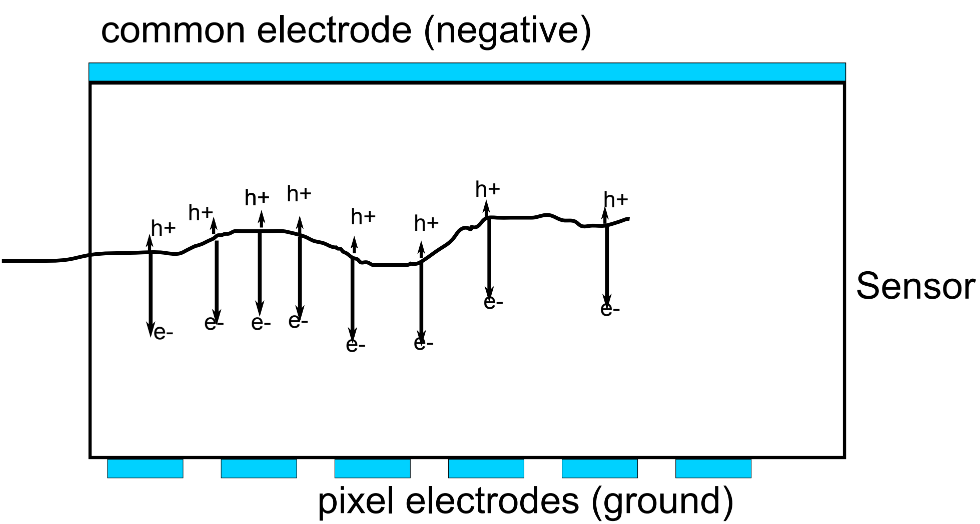
ERLANGEN CENTRE FOR ASTROPARTICLE PHYSICS

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Timepix Detector



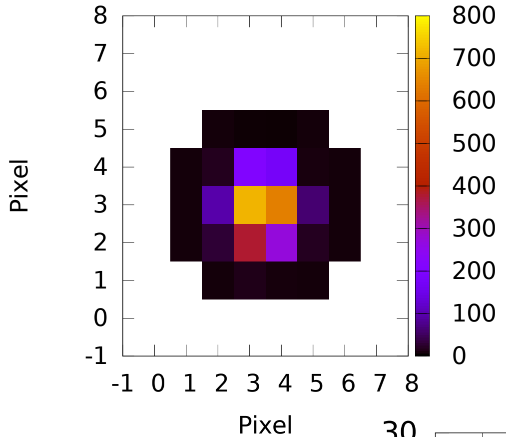
developed at CERN in the Medipix collaboration
in collaboration with EUDET.



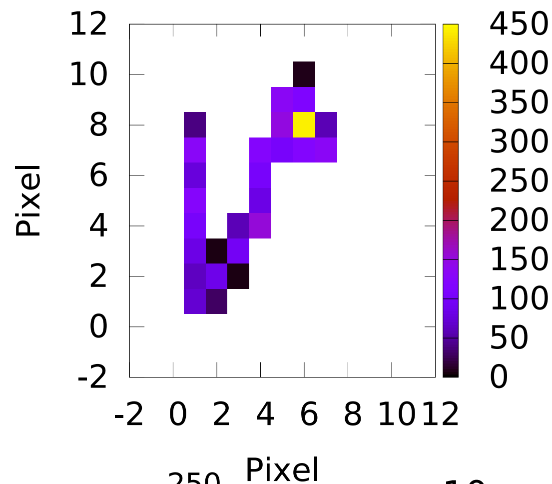
ASIC: 256 x 256 pixel
pixel pitch: 55 µm
0.25 µm CMOS technology

The tracks shown here correspond
to 110 µm pixel pitch and a CdTe sensor

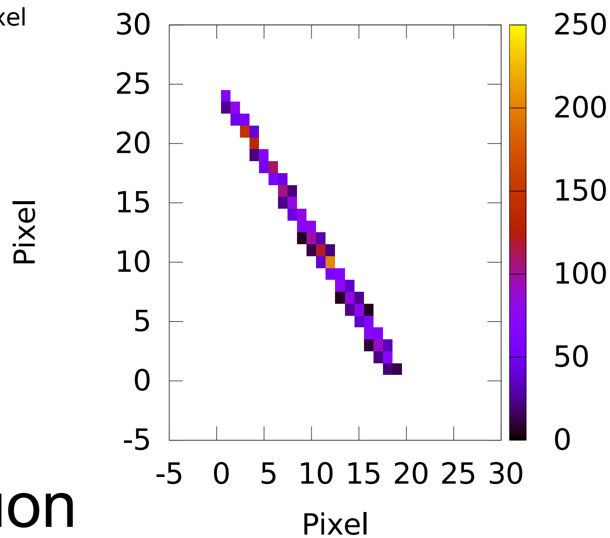
We can measure the particle tracks



Alpha

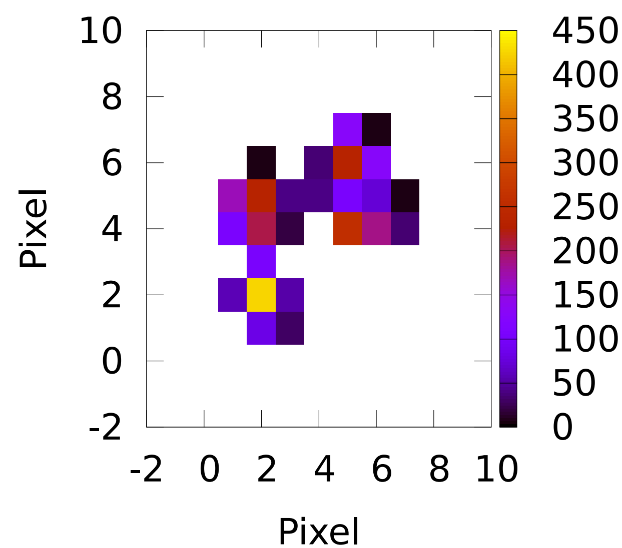


Electron



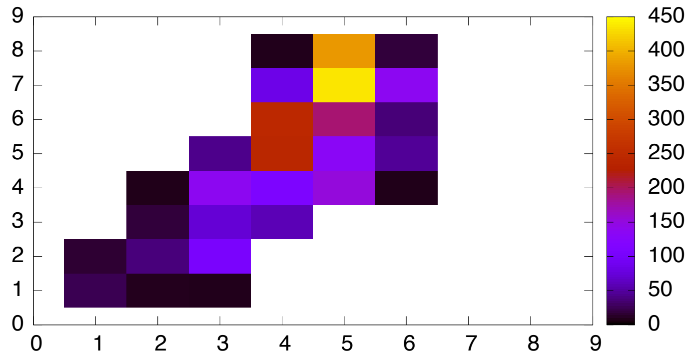
Muon

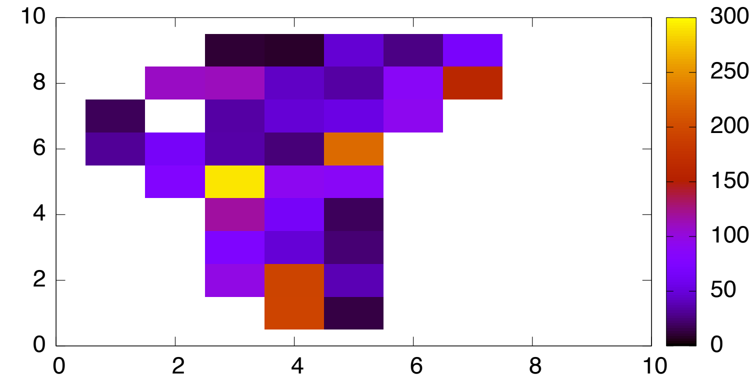
0vbb



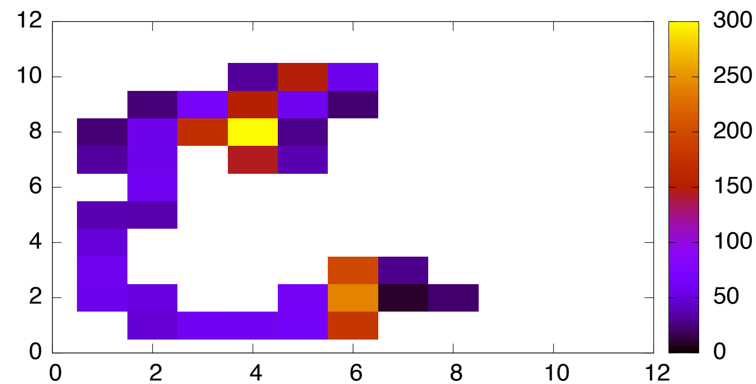
Our goal is to identify 0vbb events

Two electrons with a common point of origin and about 3 MeV kinetic energy



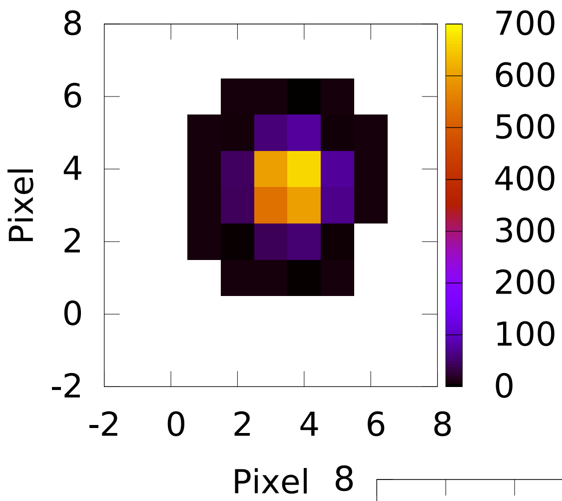


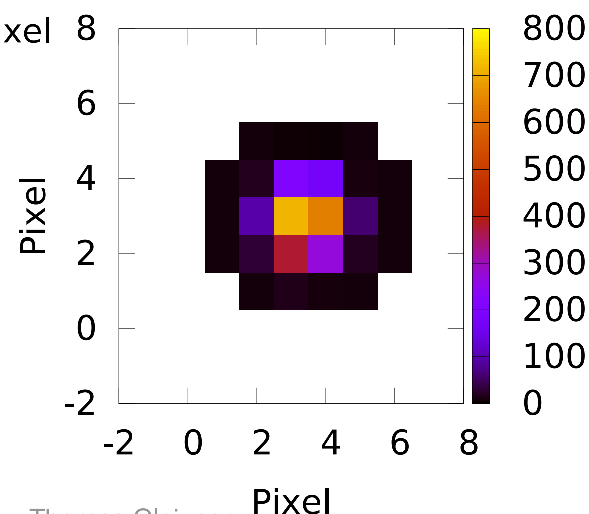
Their form can change a lot
due to random walk of the electrons

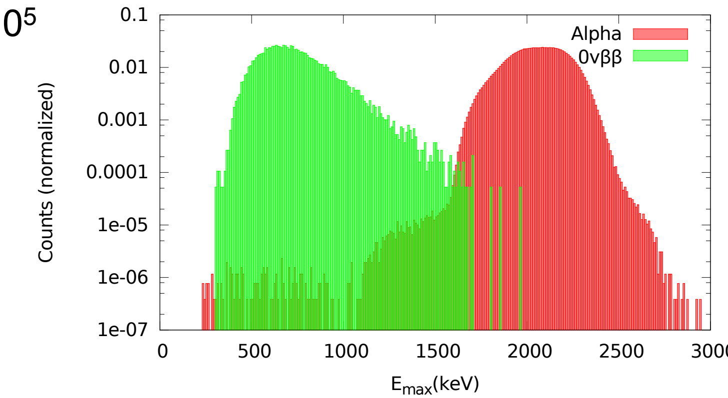


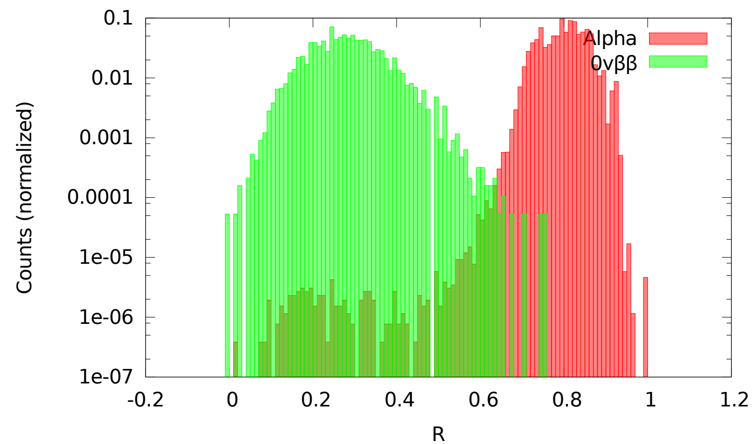
Identifying Alpha particles is easy

We already showed a reduction by a factor > 4.32×10^5









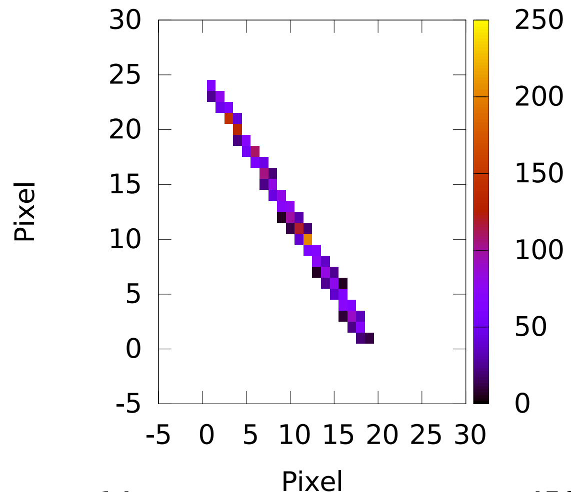
(Alpha in the graph refers to all events from an alpha source, which is mostly alpha events, but can include other types)

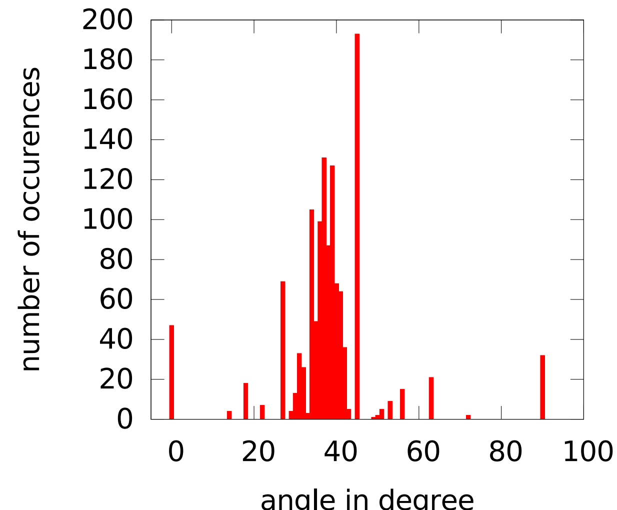
Identifying Muons is also easy

Our estimation is a reduction >10^4, no experimental test yet

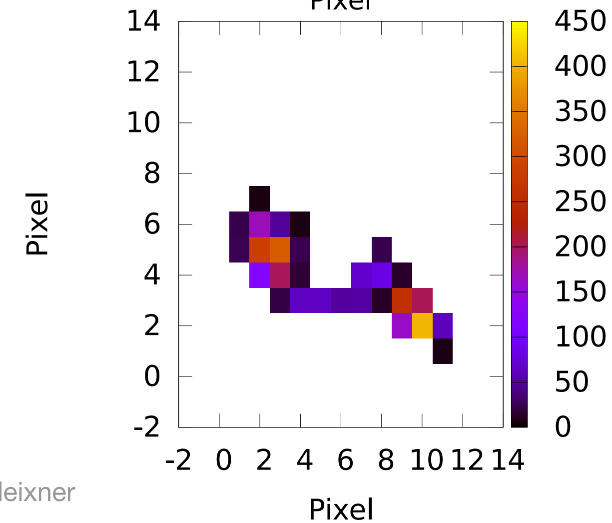
Hough-transform

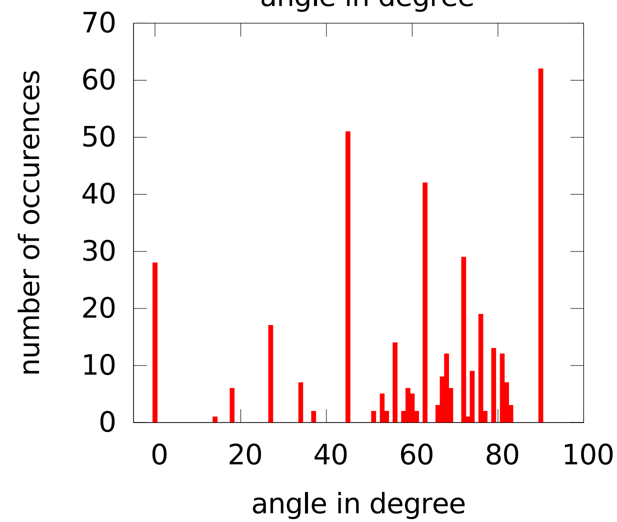
Muon





0vbb





ERLANGEN CENTRE FOR ASTROPARTICLE PHYSICS

The challenge is identifying electrons

Alpha particles “always” form a blob

Muons “always” form a straight line

The electron path is mostly random

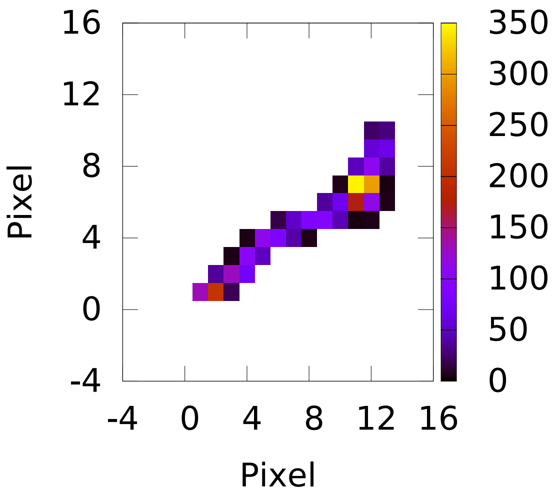
The two electron path of 0vbb is mostly random too

Using neural networks

(or similar classification algorithms)

1. we define n-features that can be calculated from the track

   For example: the number of counting pixel



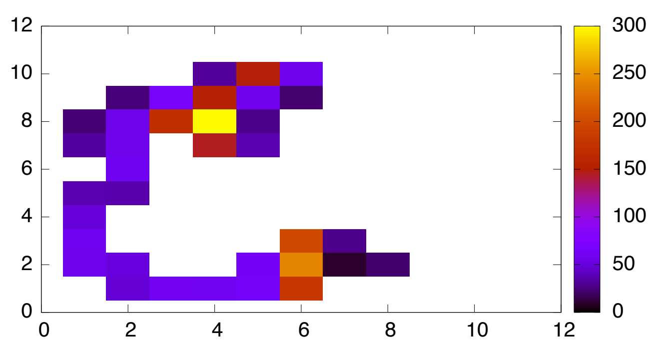
F_1=36

Using neural networks

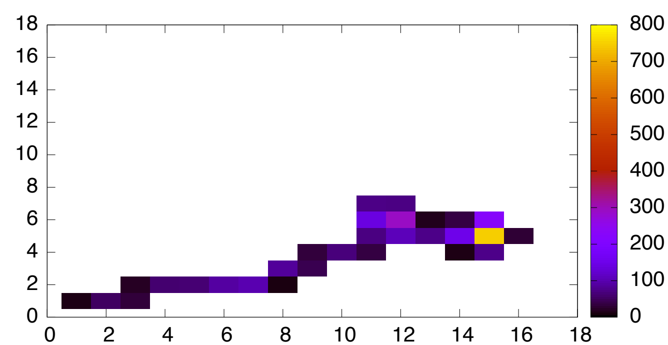
(or similar classification algorithms)

1. we define n-features that can be calculated from the track
2. we simulate a lot of events of the types we want to discriminate

For example: single electrons and 0vbb



simulated 0vbb



simulated electron

Using neural networks

(or similar classification algorithms)

1. we define n-features that can be calculated from the track
2. we simulate a lot of events of the types we want to discriminate
3. We calculate the features for each simulated event
4. We train the ANN with these features (telling the ANN the event type)
5. To identify an event, we calculate its features, and give it to the ANN which calculates a rating for the event type

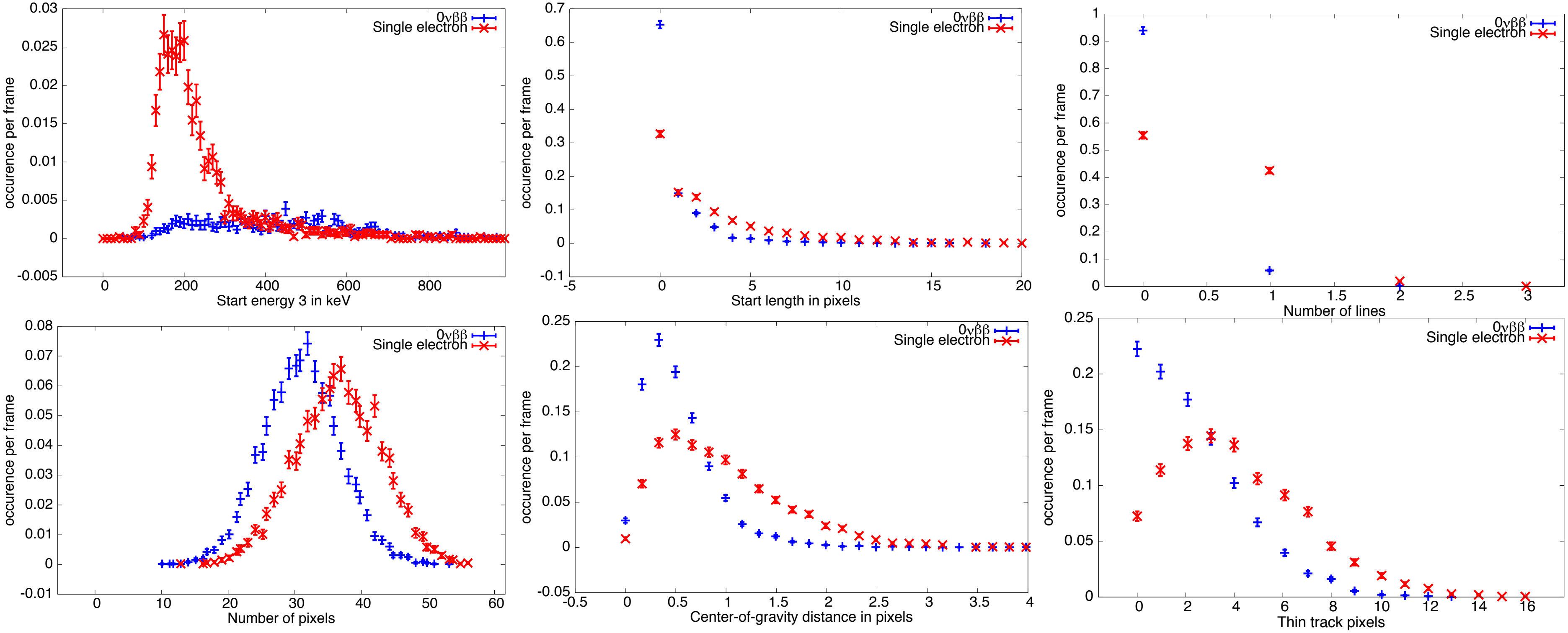
Defining the features

This is the difficult part

The features should describe the characteristic differences
Between the particles that we want to discriminate

- Electron tracks are mostly straight at first
- They become curly at the end
- They show a low energy deposition (per pixel) at first, higher at the end
- In the 0vbb case, the same energy is distributed on two electrons
  - 0vbb tracks should not show a clear start, single electrons should
  - 0vbb events have two “heads” (Bragg peak), electron tracks one
    - The energy weighted centroid is shifted towards the single “head” for single electrons, more towards the not weighted centroid for 0vbb
- Etc.

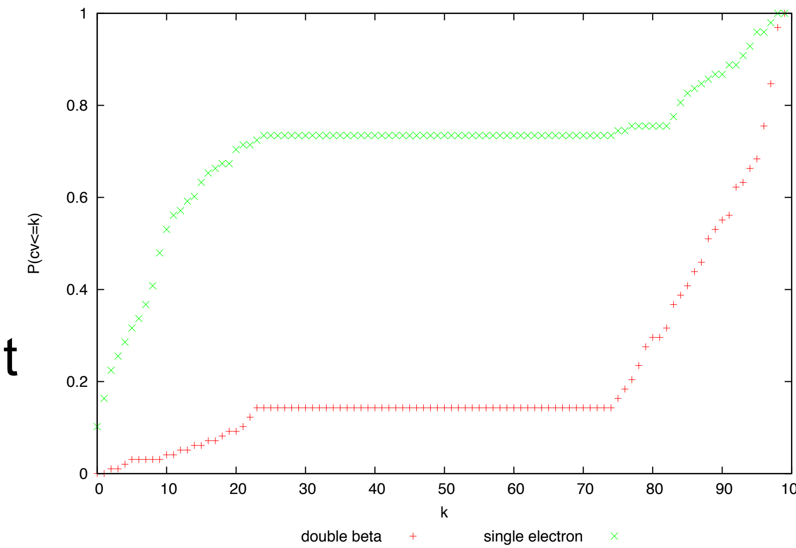
Defining the features



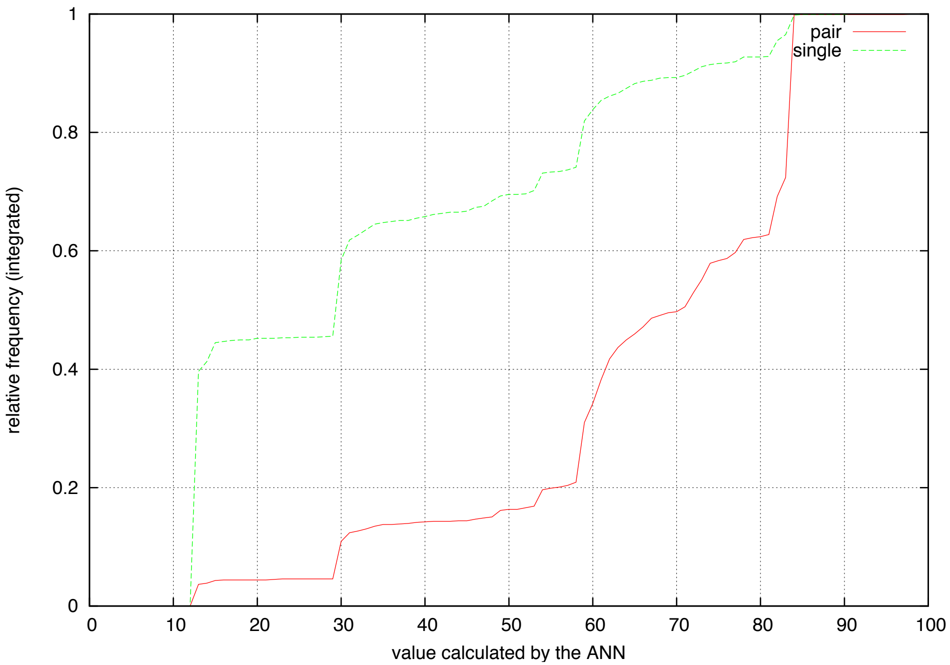
The overlap between the histograms should be small for good features.

Applying the ANN

The classification software can rate the event. If we choose certain rating to be considered a certain event type, there is a certain chance that the software will get it right

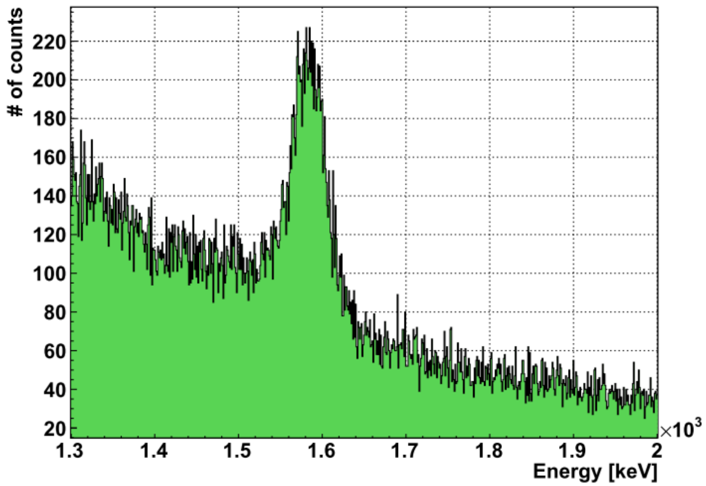


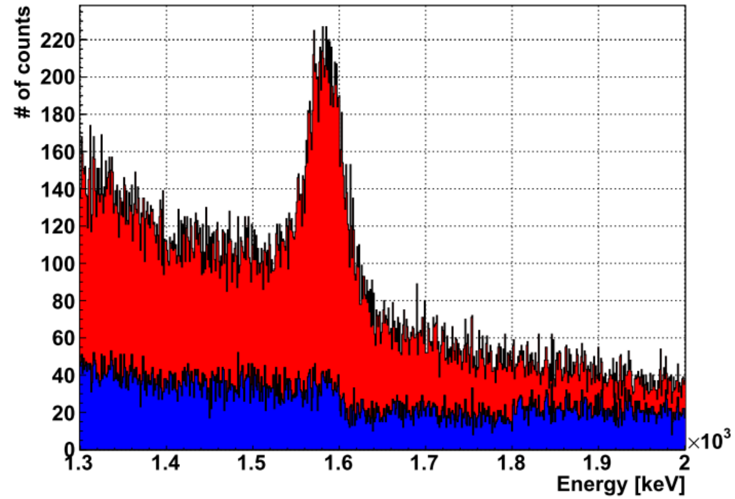
Random decision forest



Artificial neural network

This will lead to missclassification (unless we have perfect features)





green: all events
red: e^-p^+ (pair)
blue: e^- (compton)

Spectrum of a ^{208}Tl γ-emitter (2.6 MeV)

$C_{0vbb} = P_{0vbb} \times n_{0vbb} + (1-P_{electron}) \times n_{electron}$

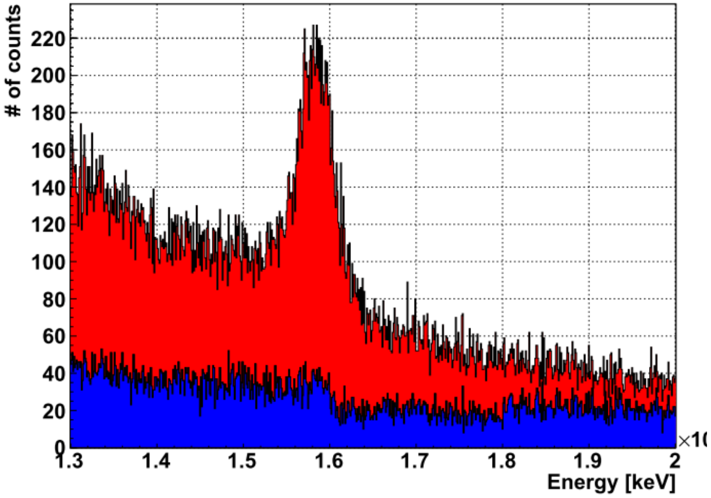
$C_{electron} = P_{electron} \times n_{electron} + (1-P_{0vbb}) \times n_{0vbb}$

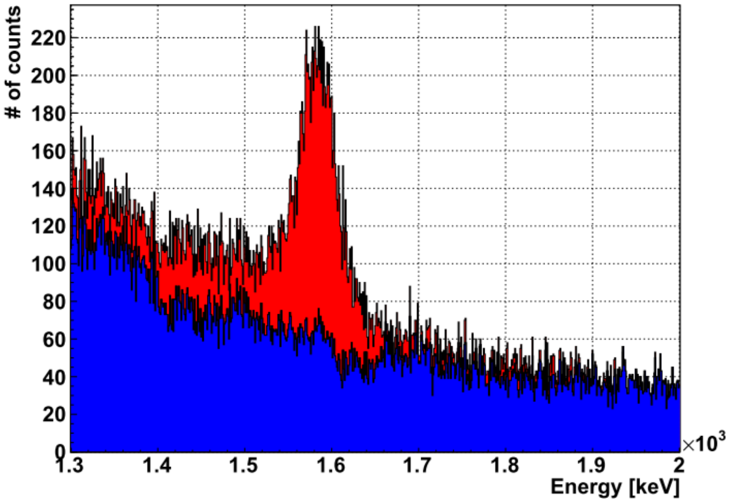
C_k: counts identified as event k; P_k: probability to identify event k as k

N_k: number of events k

Reconstructing the spectrum

left: after classification
right: after reconstruction



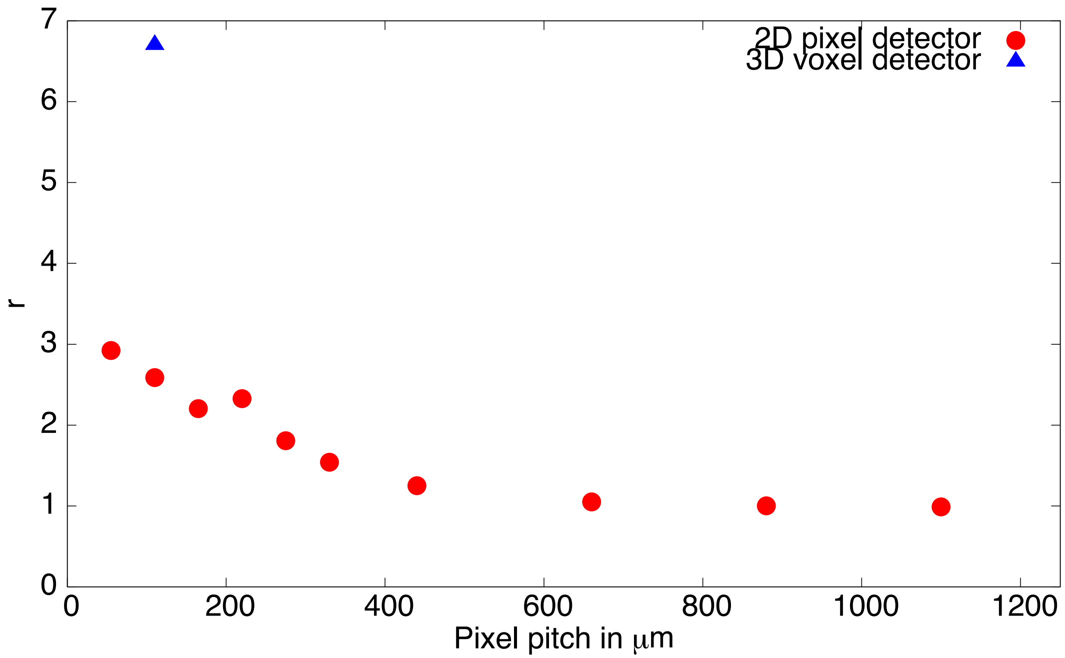


If our simulation was correct we know P and can calculate n, reconstructing the correct spectrum

$$N_{0vbb}=c_{0vbb}+n_{total}(P_{electron}-1)/(P_{0vbb}+P_{electron}-1)$$

Preliminary

This would probably work a lot better with 3D tracks.
The spatial resolution is not even that important.



Simulation of effective background reduction

(the calculation for the 3D detector contains some approximations)

Conclusions

- ANN (or RDF) can be used to discriminate even similar event types
- The efficiency depends a lot on how similar the events are and on the available track information (3D>2D)
- A precise simulation is required for this method

Thank you for your attention

ERLANGEN CENTRE
FOR ASTROPARTICLE
PHYSICS

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG