

GEM Clustering Algorithm

James Shirk

Intended Workflow

A. Identify events/channel with hits per event; format into TTree

B. Using ADC, pulse time, channels with hits per event, Perform 1D clustering

C. Combinatorially associate all 1D U clusters with all 1D V clusters

D. Decide likely pairings, associate into hit values positions

E. Perform coordinate transform into Cartesian

F. Hit map

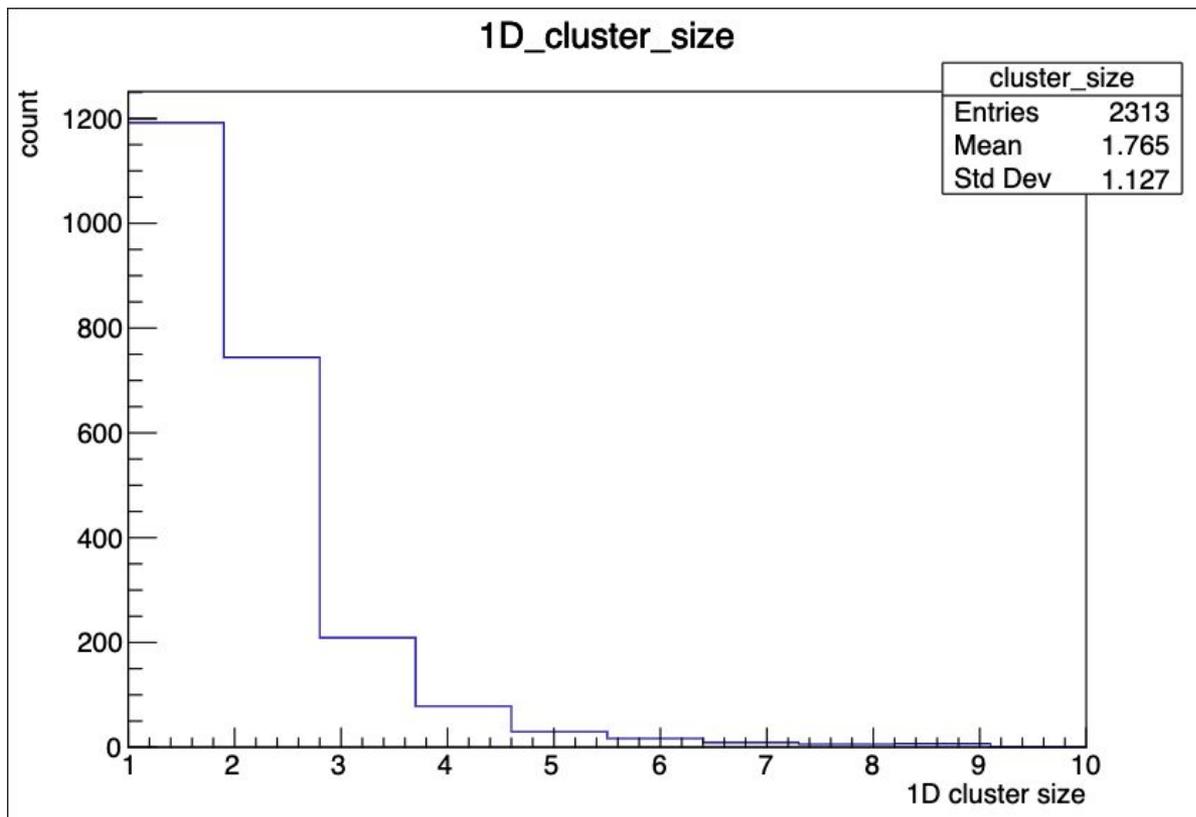
A. Identifying hits

- Jaydeep wrote an algorithm to do some baseline noise correction and locate hits
- Formatted into TTree
- Probably some further noise corrections to be done, will improve quality of next steps

B. 1D clustering: Problem statement with example

- Counts belonging to one cluster do not necessarily show up in subsequent channels
 - We occasionally miss channels due to high noise, etc.
- [16, 497, 498, 499, 630, 1082, 1083, 1085, 1086, 1198, 1201, 1268, 1269, 1270]
- By eye, would cluster as:
 - [[16], [497, 498, 499], [1082, 1083, 1085, 1086], [1198, 1201], [1268, 1269, 1270]]
 - If we just checked subsequent channels, we would have two extra clusters

B. 1D Cluster size



Intended Workflow

A. Identify events/channel with hits per event; format into TTree

B. Using ADC, pulse time, channels with hits per event, Perform 1D clustering

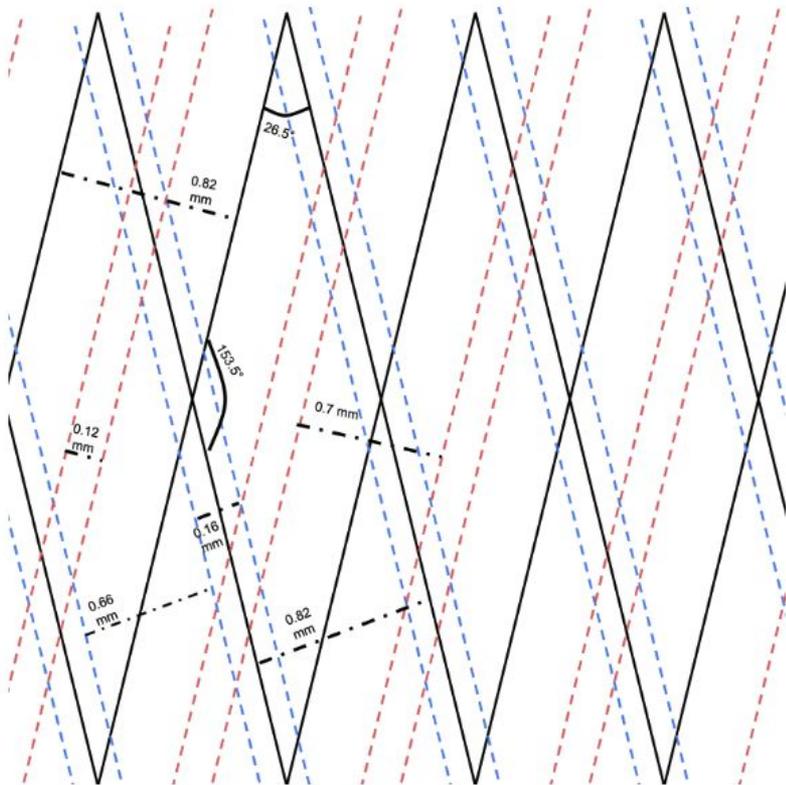
C. Combinatorially associate all 1D U clusters with all 1D V clusters

D. Decide likely pairings, associate into hit values positions

E. Perform coordinate transform into Cartesian

F. Hit map

E. Coordinate transformation readout geometry

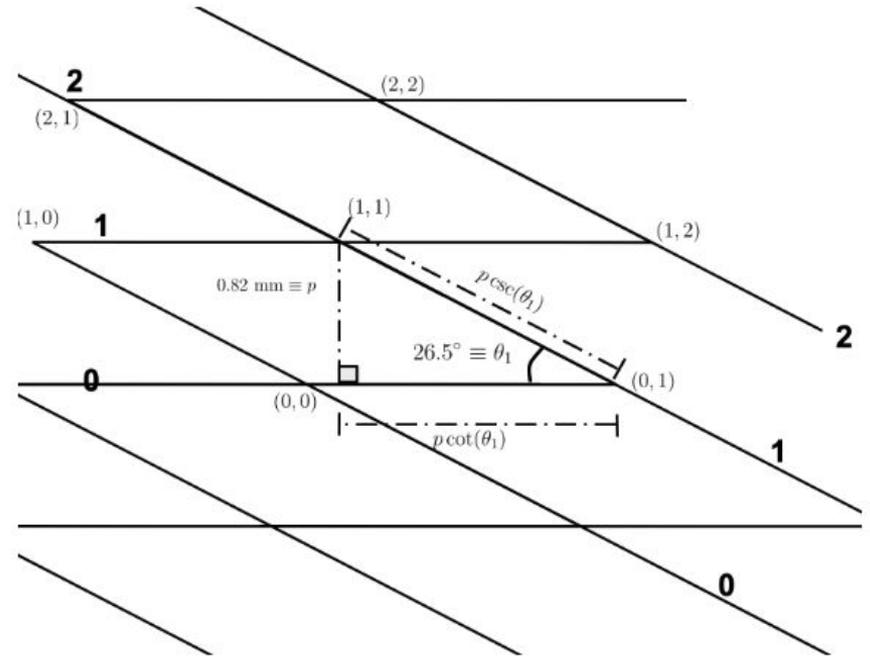


E. Coordinate transformation

- Number coordinates based on strip numbers
- Can generate a transformation matrix from strip coordinates to cartesian coordinates

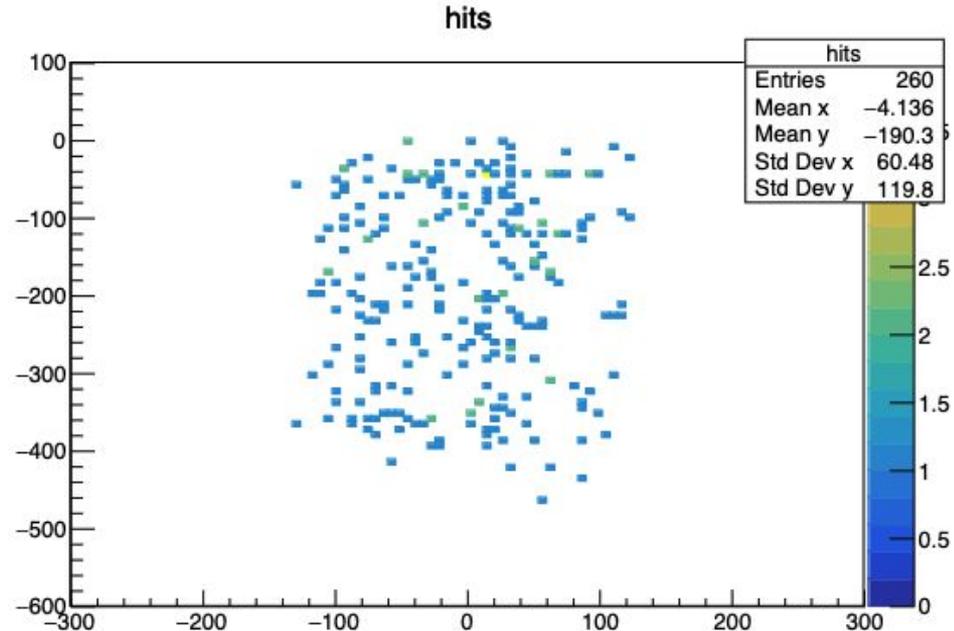
$$\begin{bmatrix} r_x \\ r_y \end{bmatrix} = \begin{bmatrix} a_{1x} & a_{1y} \\ a_{2x} & a_{2y} \end{bmatrix} \begin{bmatrix} \rho_x \\ \rho_y \end{bmatrix}$$

$$= \begin{bmatrix} p \csc(\theta_1) & 0 \\ -p \cot(\theta_1) & p \end{bmatrix} \begin{bmatrix} \rho_x \\ \rho_y \end{bmatrix}$$



E. Coordinate transformation proof of concept

- For multiplicity 2-events (to prove concept) can skip 1D clustering
- Shaped like a rectangle rather than our GEM because we used rectangular scintillators for trigger



Intended Workflow

A. Identify events/channel with hits per event; format into TTree

B. Using ADC, pulse time, channels with hits per event, Perform 1D clustering

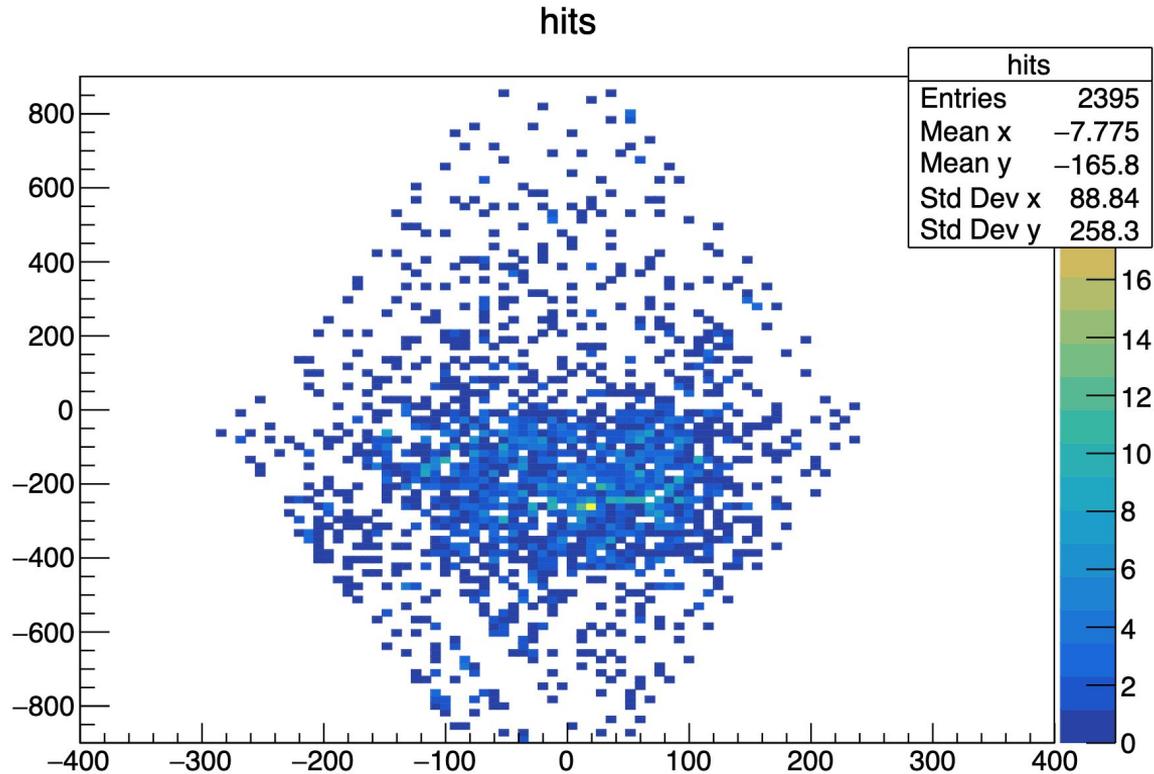
C. Combinatorially associate all 1D U clusters with all 1D V clusters

D. Decide **likely** pairings, associate into hit values positions

E. Perform coordinate transform into Cartesian

F. Hit map

Results for all numbers of clusters



Hits for events with two clusters

