(Really) Fast Single Vertex Finder

Petr Balek, Tomasz Bold

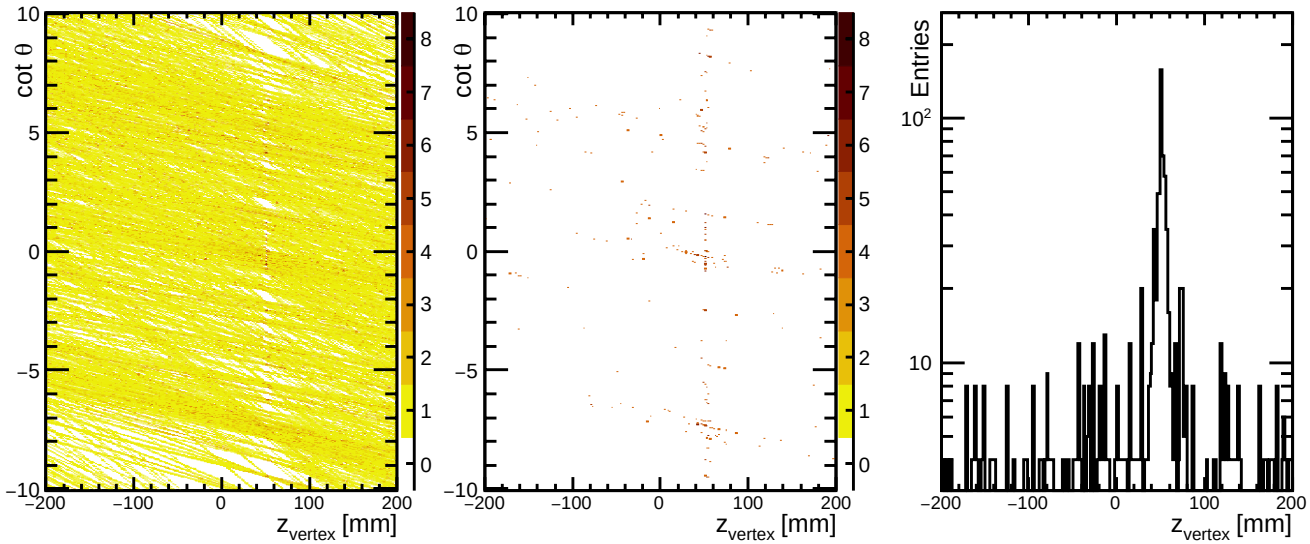
10 September 2024

introduction

- fast single vertex finder presented some time ago
  - link1, link2
  - used triplets of measurements fitted with a line or a plane
  - vertex was the point of minimal $\Sigma\, distance^2$ to the lines/planes
  - **polynomial** in the number of measurements as it had to create all the triplets
  - resolution of a few mm in z

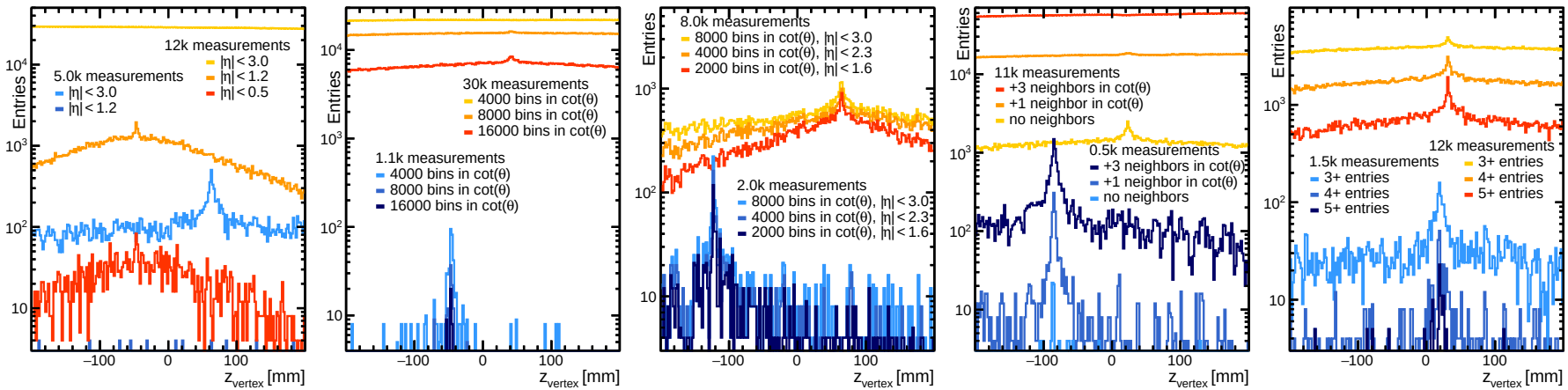
vertex finder with Hough transform

- really fast single vertex finder
  - **linear** in the number of measurements
  - use Hough transform to populate 2D histogram of $z_{vertex} - \cot(\theta)$
    - ★ $\cot(\theta) = (z - z_{vertex})/r$
    - ★ also, $\cot(\theta)$ has reasonable values for any $|\eta|$ range
  - each measurement is a line in the 2D histogram
  - remove the bins without enough entries, i.e. where there is no track
  - make 1D projection to z_{vertex} axis
  - find a peak



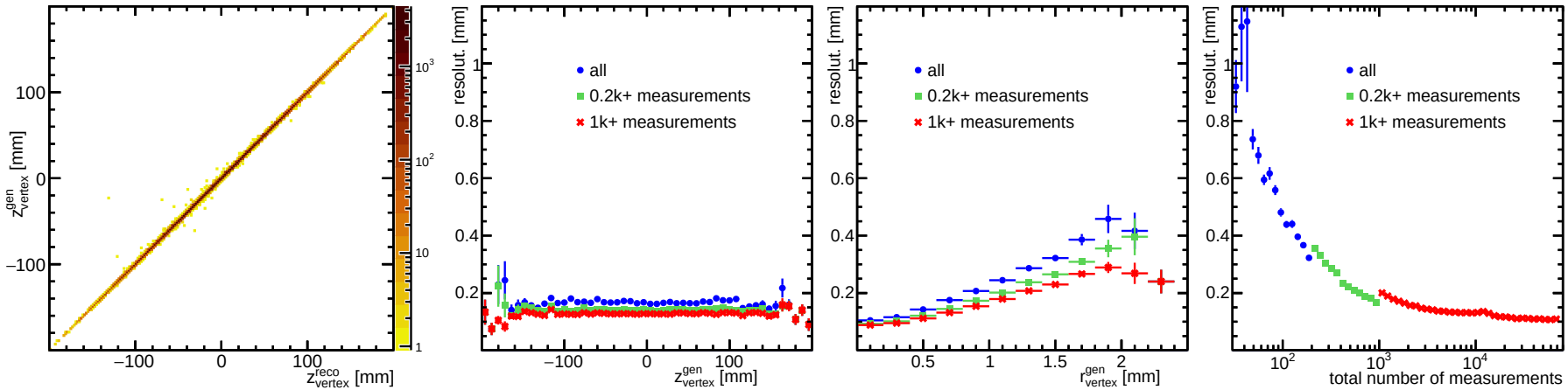
parameters of the vertex finder

- sub-optimal setup may prevent the peak identification, as illustrated below
  - too many/too little measurements
    - ★ number of measurements can be adjusted with $|\eta|$ range
    - ★ $|\eta|$ range also sets limits on $\cot(\theta)$ range
  - too fine/too rough binning in $\cot(\theta)$
  - line is too wide/no enough wide
  - request too little/too much entries for pedestal removal



parameters of the vertex finder

- parameters tuned for ODD
  - limit η to use approx. 10^4 measurements and no more
    - ★ makes the algorithm even better than linear!
  - for less than 10^4, use $|\eta| < 3.0$ but have less bins in $\cot(\theta)$
    - ★ for every $10\times$ less measurements, decrease the number of bins $2\times$
  - make the line in HT wider for less than 1000 measurements; even wider for less than 200 measurements



- off-diagonal entries – events with low multiplicity
- r is calculated from $(x, y) = (0, 0)$

implementation of the vertex finder

- initially wanted to reuse the implementation of the Hough transform as much as possible
- `HoughTransformUtils` is too complex
  - every bin in 2D histogram ("`HoughPlane`") is an object ("`HoughCell`") that keeps track of each hit
    - ★ hits have unique identifiers, those are recorded
  - filling of the 2D histogram is not optimal
    - ★ we can easily calculate at what z_{vertex} value the line actually enters the histogram, skipping all z_{vertex} values before that
    - ★ in `HoughPlane`, we can't skip them because the "line" is represented as a generic function
  - wouldn't help with pedestal removal nor with the projection
  - using only minor function for bin center or for finding bin number
- using `MultiIndexedVector2D` for the 2D histogram
  - still need to implement the pedestal removal and 1D projection
- peak finding is a simple find-a-maximum loop
  - z_{vertex} position is a weighted average around the maximum

summary & outlook

- ultimately, this information will be used to filter seeds
  - only prof of concept implementation in place
- ideal for high-multiplicity events:
  - fast, accurate, potential benefit is large
- with multiplicity low enough, it's not worth to run it:
  - it's inaccurate
  - potential benefit is small

  $\Rightarrow$ there will be some cut-off for the number of measurements

- prepared an advanced paper draft describing this method and it's performance with ODD
- performance with ATLAS ITk should be comparable, assuming we tune the parameters again