A Simulation of the Tiny Triplet Finder Algorithm for the BTeV Level 1 Trigger

By: Henry Garcia

Advisor: Jin Yuan Wu

Outline

- BTeV overview,
- BTeV trigger overview,
- Level 1 trigger overview,
- TTF algorithm,
- Preliminary results,
- Possible future improvements.

The Goals of BTeV

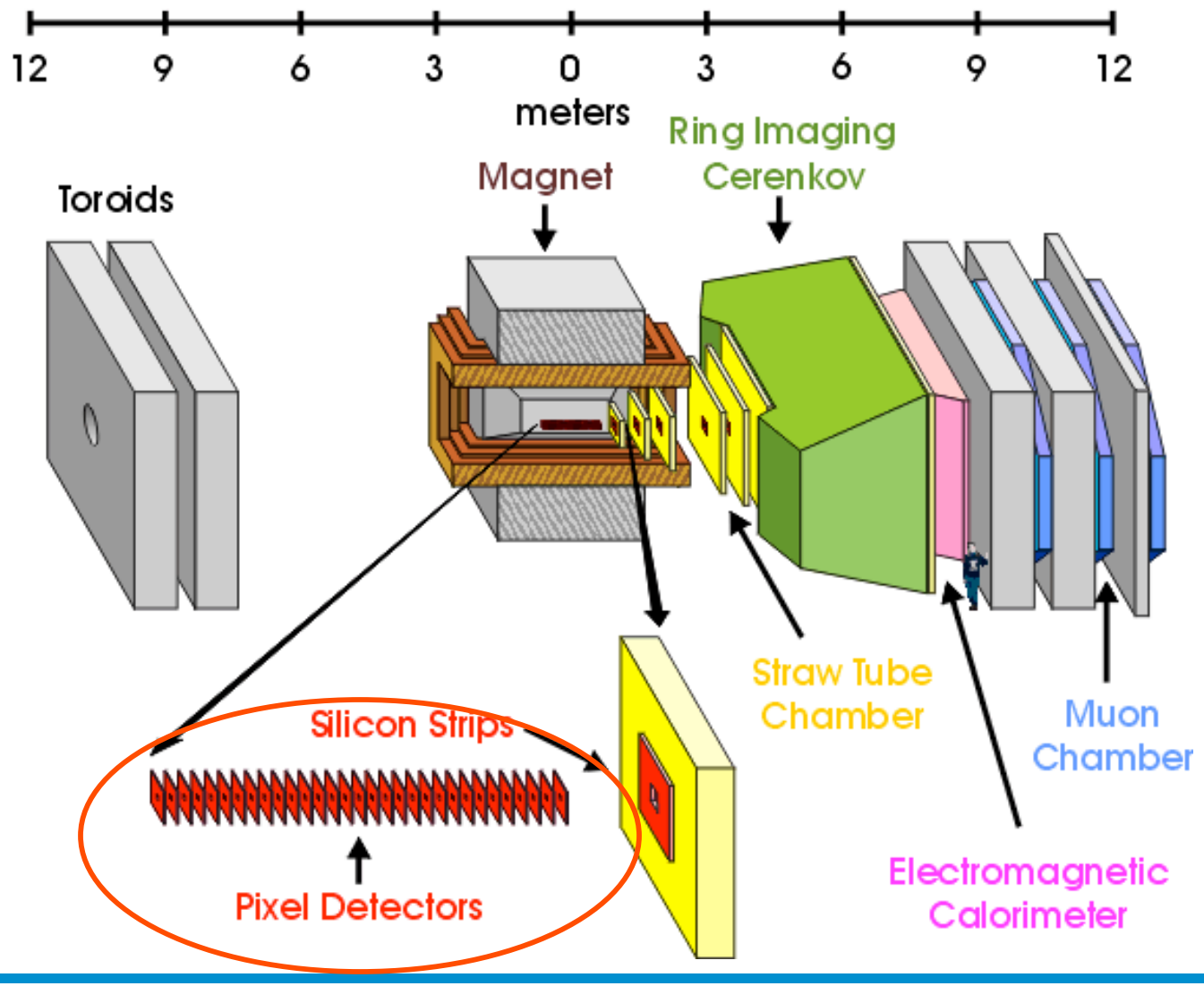
The BTeV detector is designed to challenge the standard model (SM) explanation of CP violation, mixing and rare decays of bottom and charm quark states by making precise measurements of SM parameters and an exhaustive search for physics beyond the SM.

Installation is scheduled to start in 2007, followed by commissioning in 2009, and data-taking in 2010.

Why BTeV?

- BTeV excels in several crucial areas including:
  - Triggering on decays with purely hadronic final states (vetex trigger),
  - Charged particle identification (RICH, electromagnetic calorimeter, & muon system),
  - Single photon and π_0 reconstruction (electromagnetic calorimeter),
  - Excellent proper time resolution (pixel detector),
  - Good mass resolution (forward silicon & straws).

BTeV Detector Layout



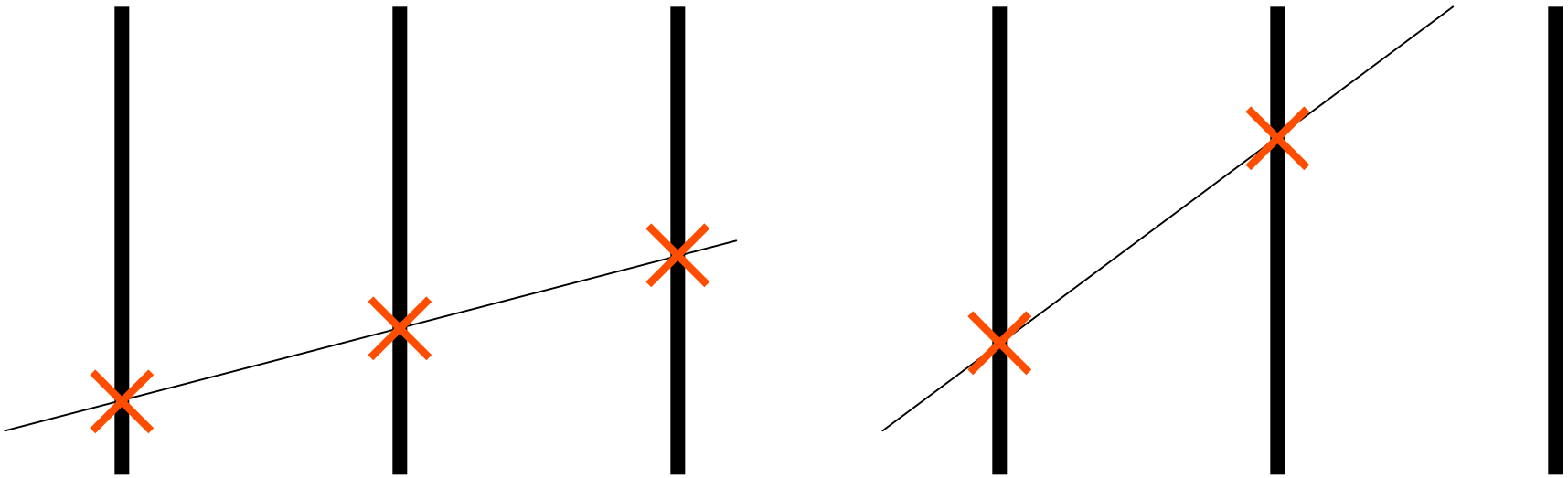
The BTeV Trigger

Goal of each level: *to select events with detached vertices typical of heavy-quark decays.*

- Level 1: initial vertex finding algorithms applied to pixel data only. (132 ns)
- Level 2: redoes reconstruction finding each hit in each track and extending the data to the first three stations of the forward tracker. (13.2 _s)
- Level 3: full pattern recognition using pixel and all station data. (132 _s)

Triplet Finding

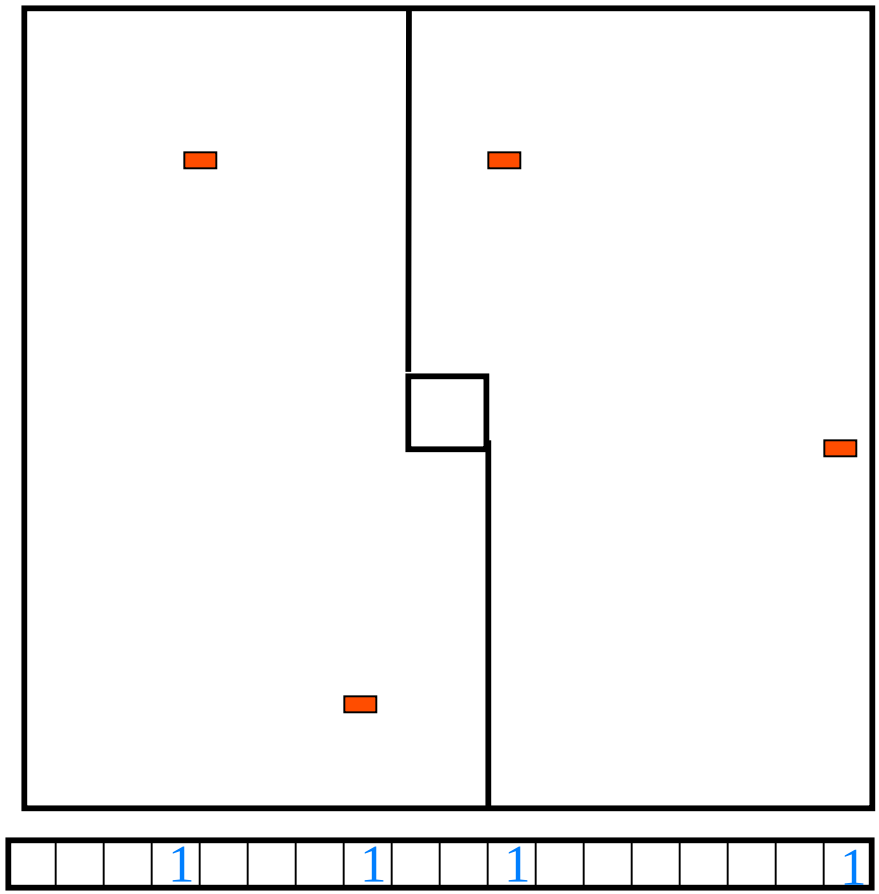
- Level 1 simply finds small segments of the tracks called triplets.
  - Triplets are composed of three hits from three adjacent stations forming a track segment.



Evaluating The Level 1 Trigger

- Simulated data file gives each track a unique track id.
- Need to find at least 1 triplet for each track.
- 47 events evaluated separately.

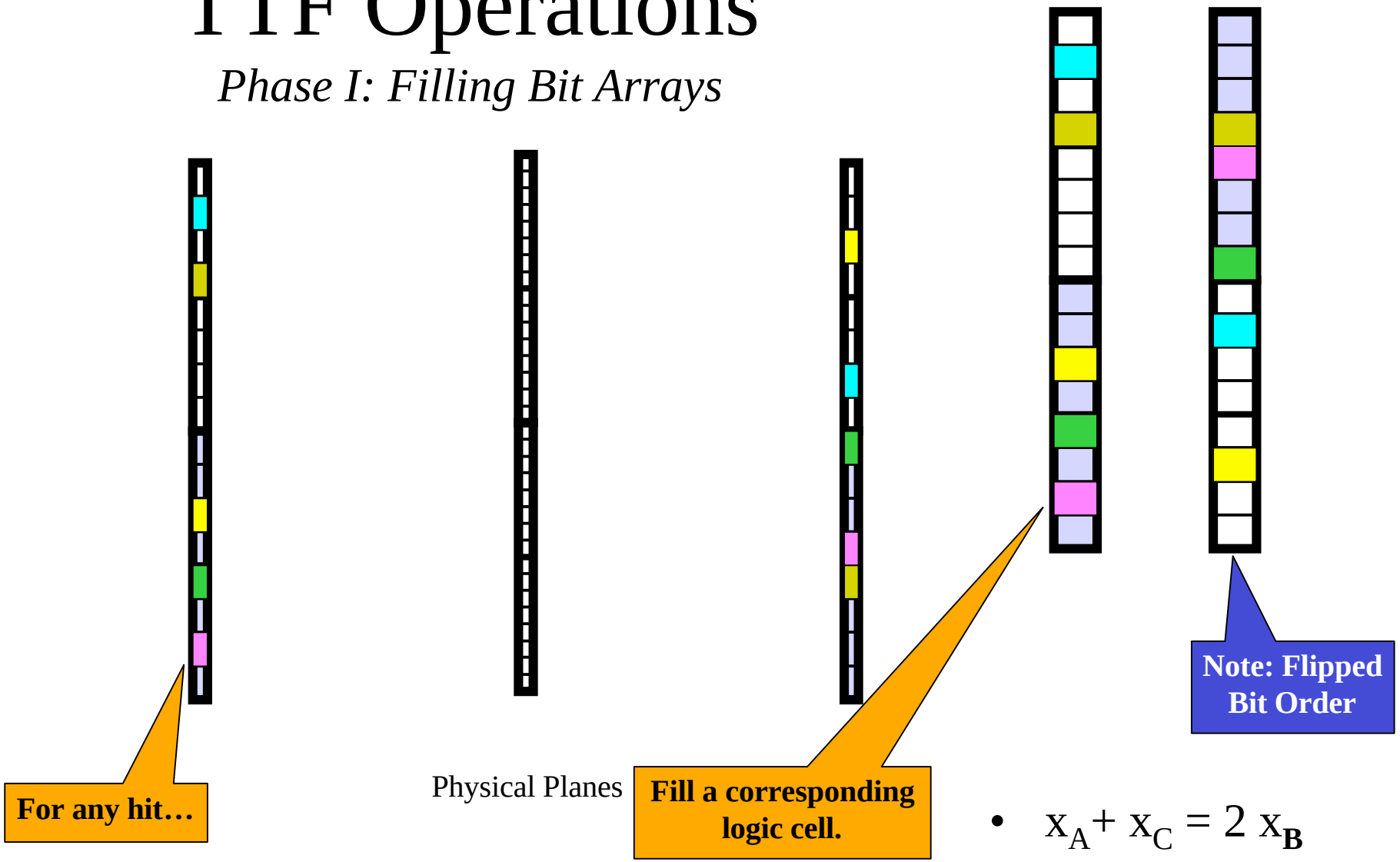
Storing the Hit Data



- Take a 2-d station and collapse it to a 1-d array.
- Pixels are 50 _m x 400 _m.
  - This results in a needed 270 bit array.

TTF Operations

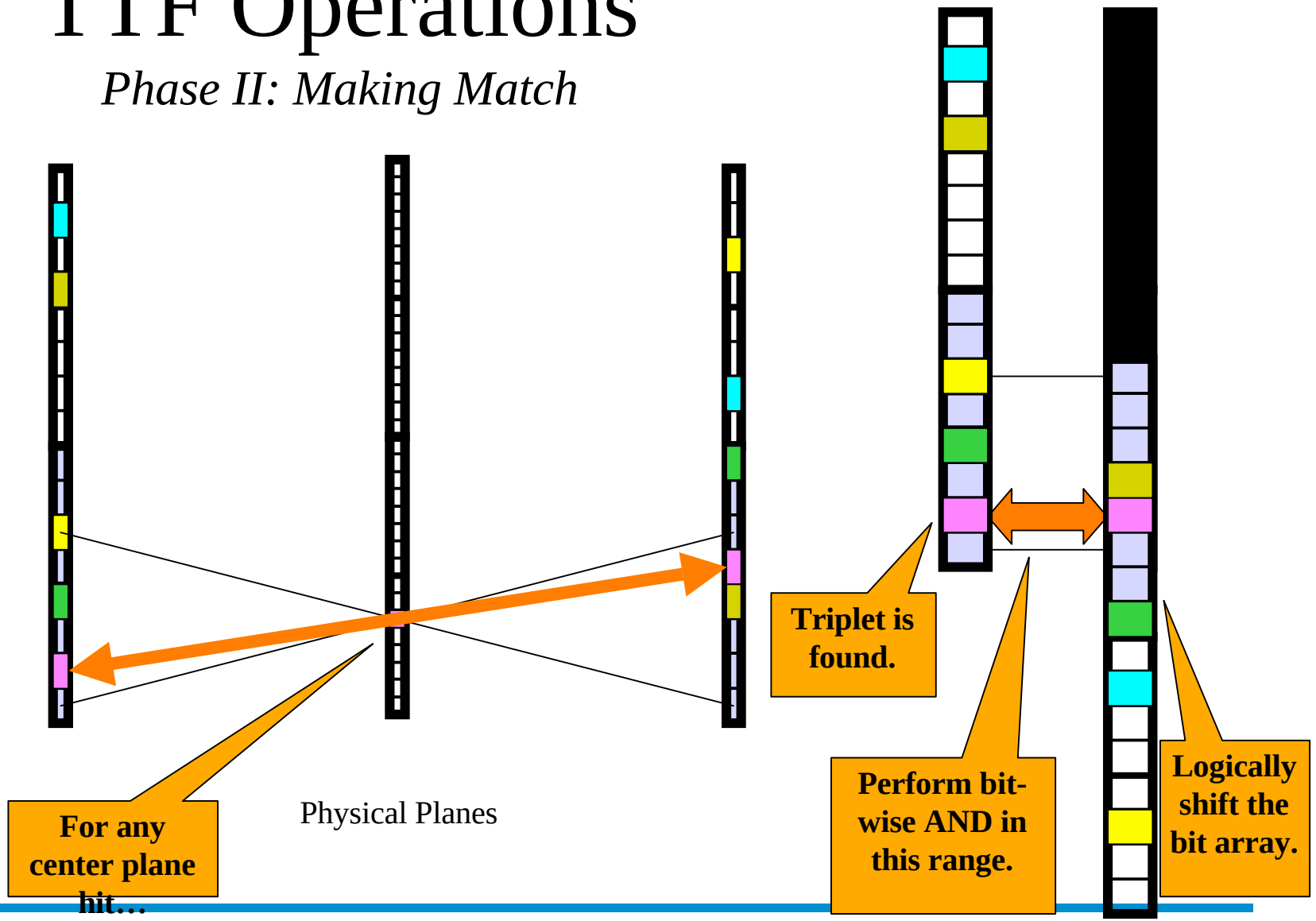
Phase I: Filling Bit Arrays



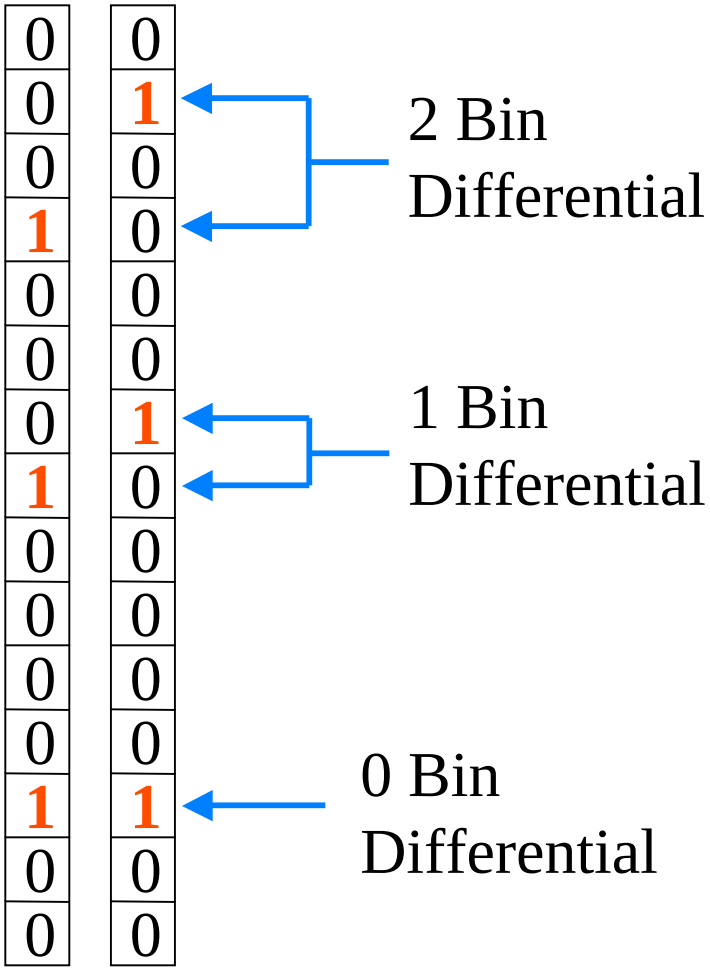
- $x_A + x_C = 2\, x_B$

TTF Operations

Phase II: Making Match

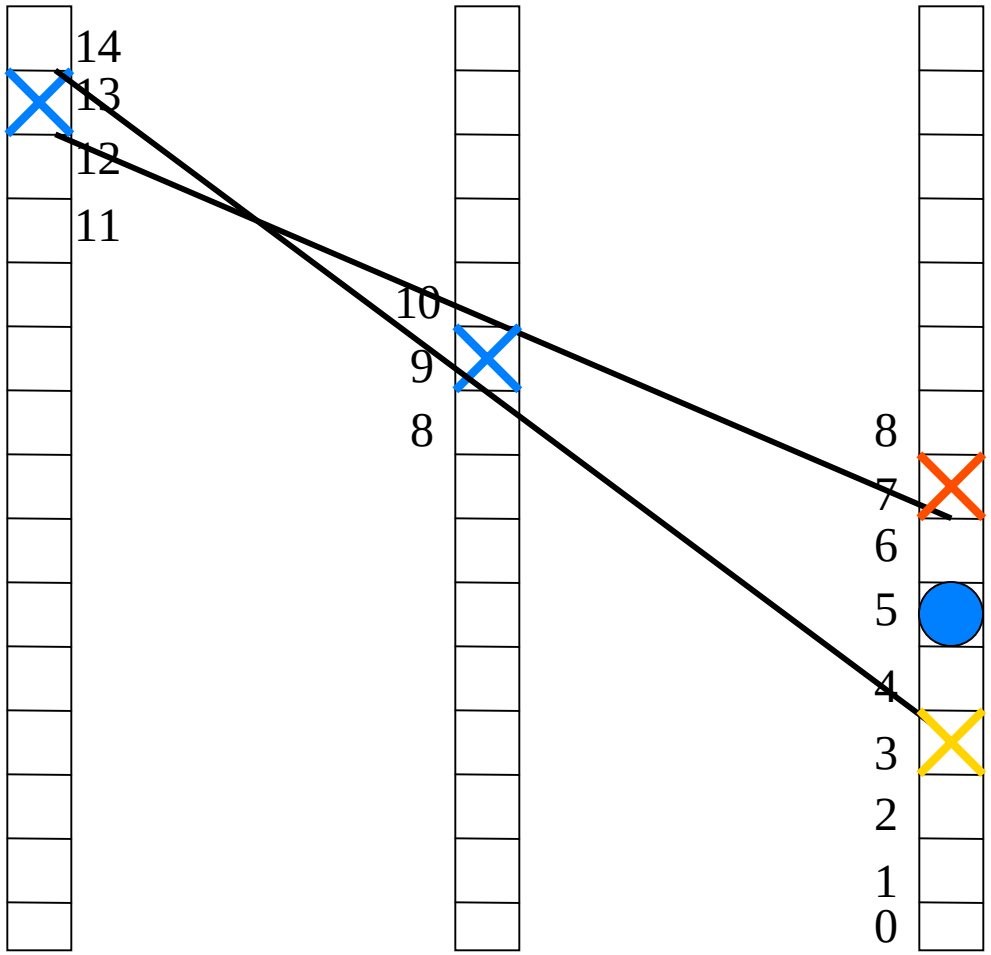


The AND Operation



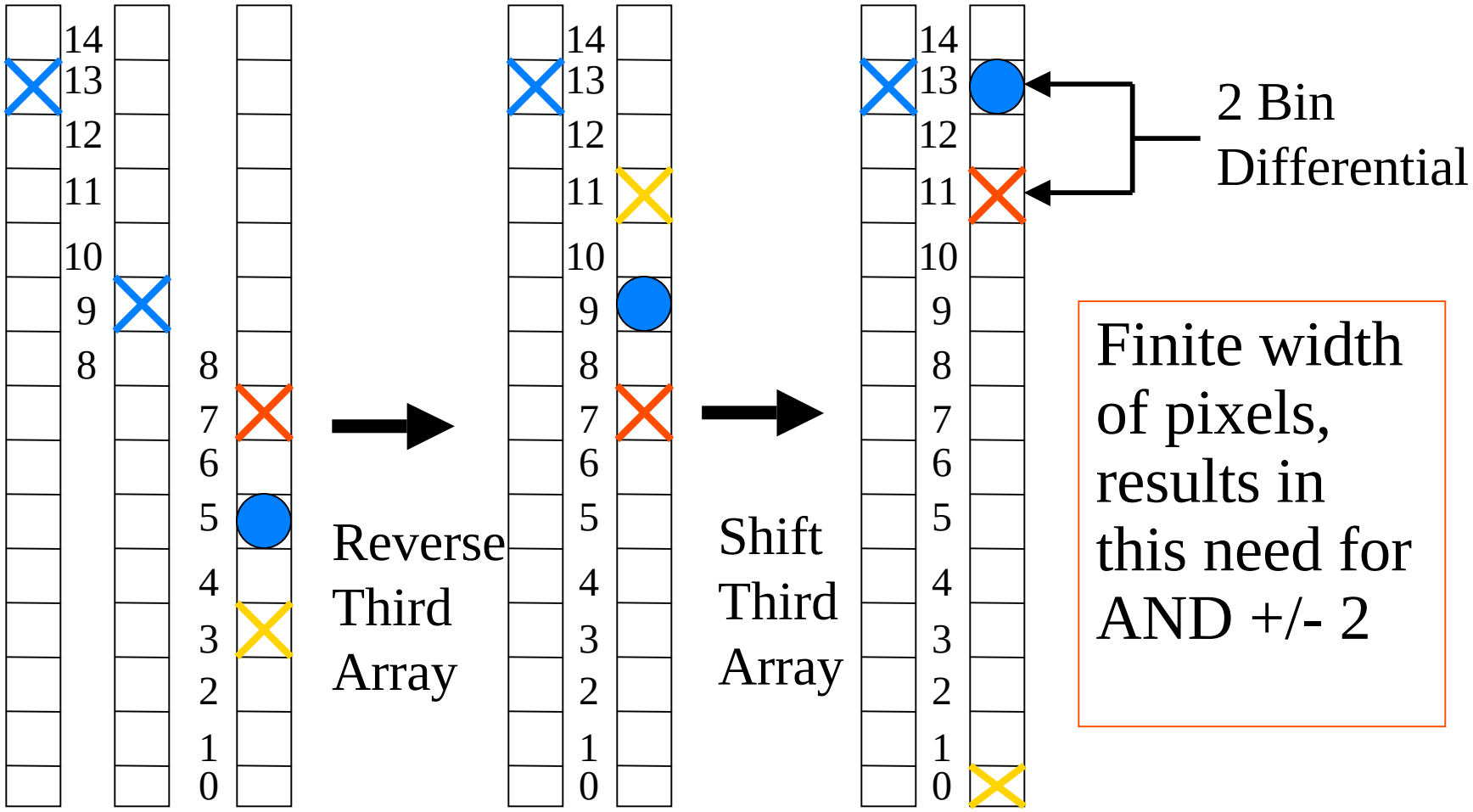
- Finite width of pixels results in a possible shift in bin #.
- Triplets can have a differential of 0, 1, 2 or 3.
- Check AND bin +/- 1 or +/- 2?

The Rounding Problem



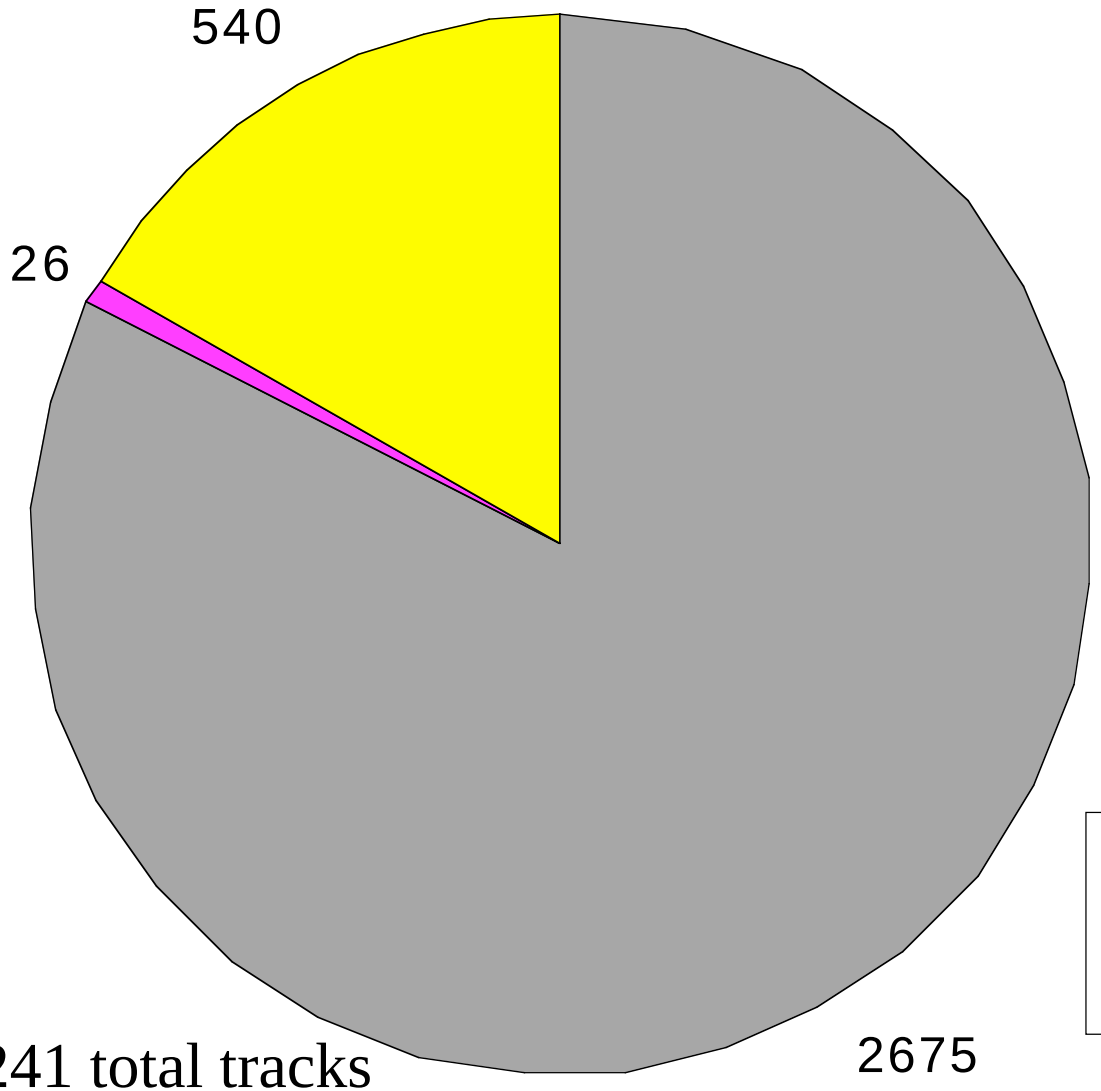
- Designate a bin corresponding to a pixel hit.
- Notice borderline hits.

The Rounding Problem



Finite width of pixels, results in this need for AND +/- 2

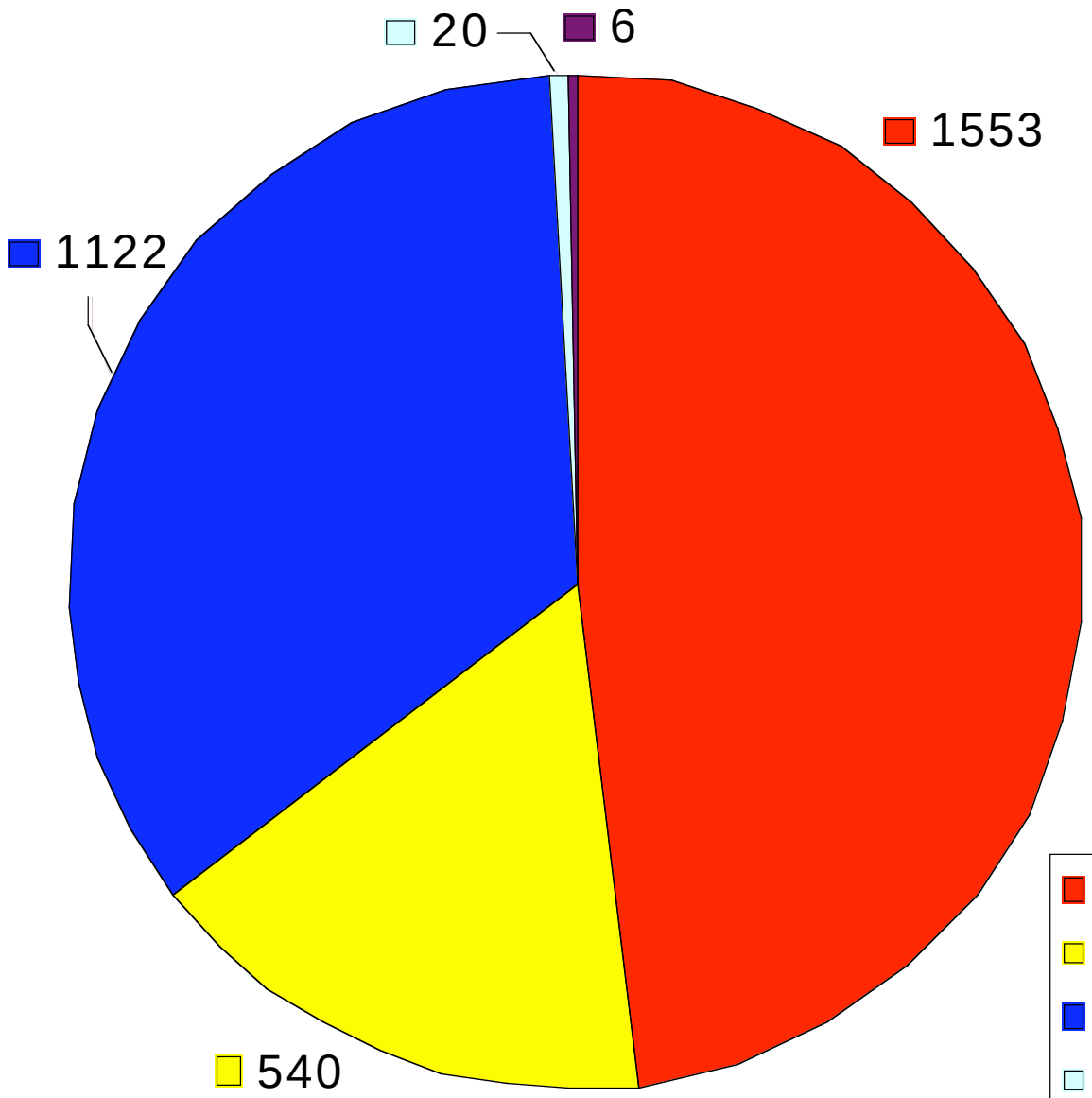
Results of AND +/- 1



3241 total tracks

- 99.0% of findable tracks found.
- Only 26 findable tracks are missed.
- 17% of the total tracks are not findable.

Results of AND +/- 1

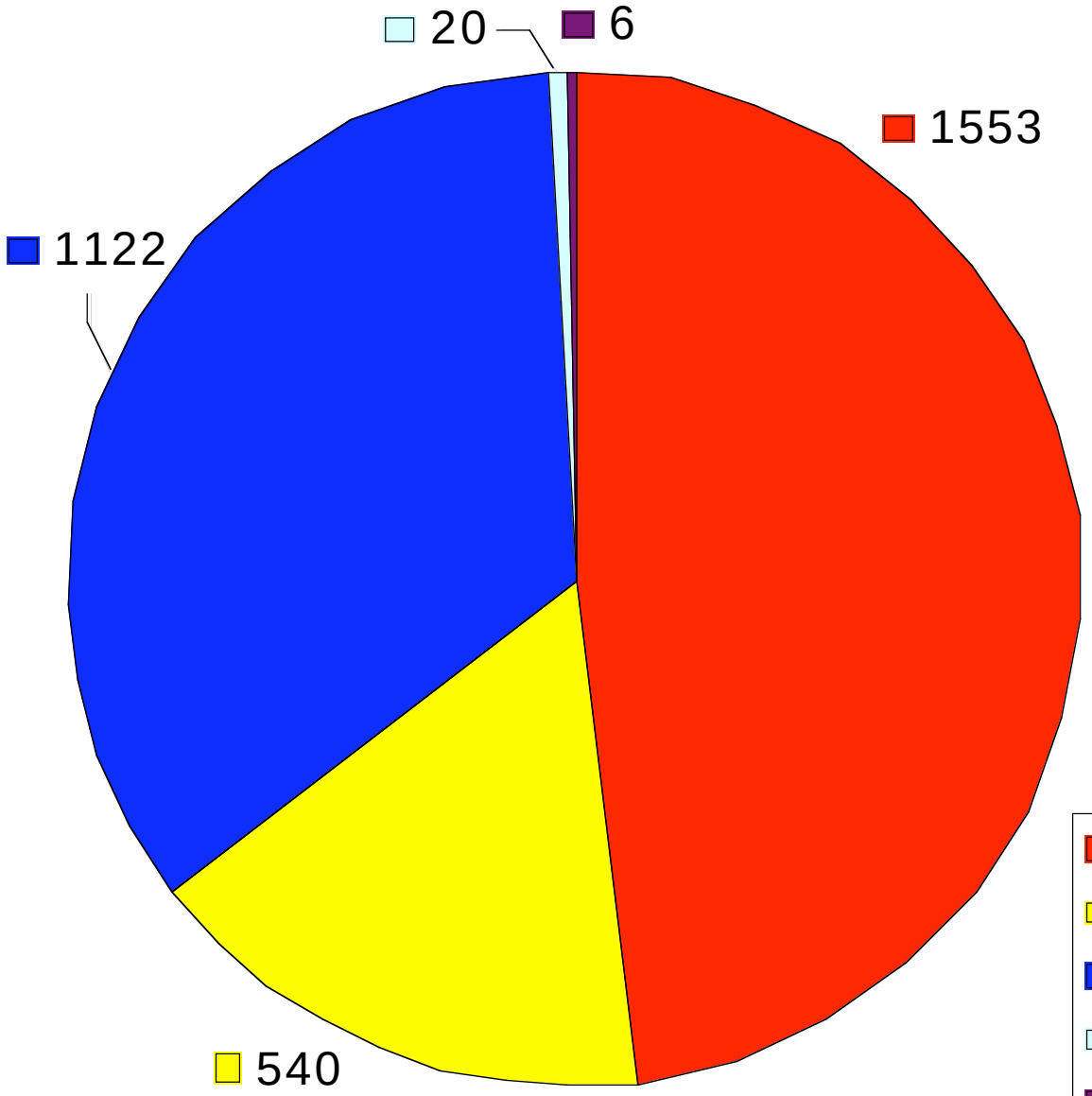


3241 total tracks

- 99.0% of findable tracks found.
- For tracks > 2GeV, 99.6% of findable tracks found.
- 48% of tracks are < 2 GeV.

> 2 GeV Tracks Found
< 3 Hits
< 2 GeV Tracks Found
< 2 GeV Tracks Missed
> 2 GeV Tracks Missed

Results of AND +/- 2



3241 total tracks

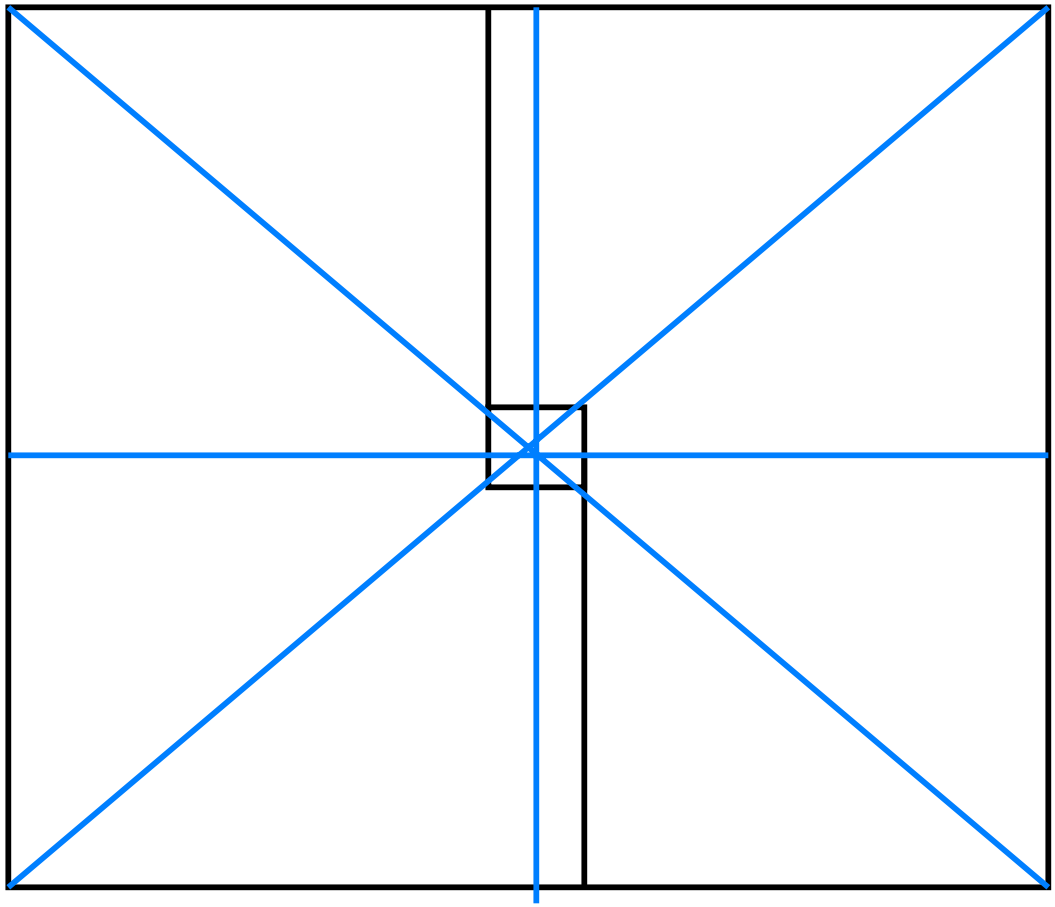
- 100% of findable tracks are found.
- Invalid triplets found greatly increases.

> 2 GeV Tracks Found
< 3 Hits
< 2 GeV Tracks Found
< 2 GeV Tracks Missed
> 2 GeV Tracks Missed

Comparative Results

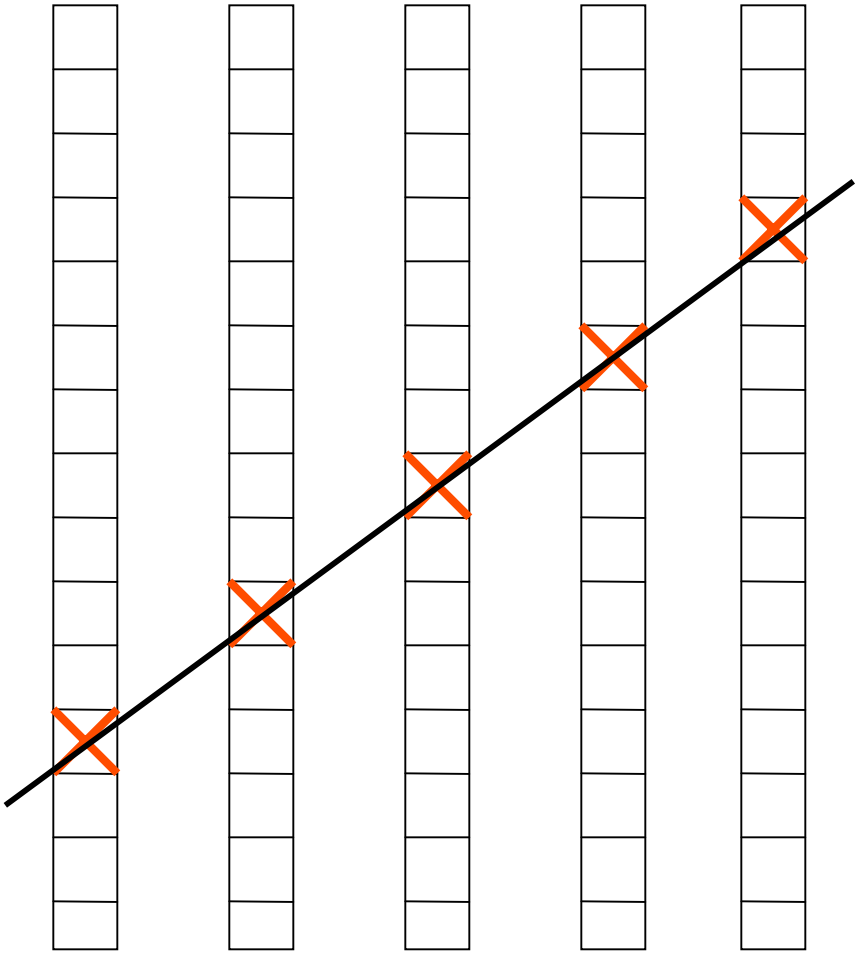
	# Valid Triplets Found	# Invalid Triplets Found	% Findable Tracks Found
AND +/-1 All P	20,152	150,339	99.0%
AND +/-1 >2 GeV P	20,152 +12%	150,339 +63%!	99.6%
AND +/-2 All P	22,652	245,253	100%

Reducing Invalid Triplets (1)



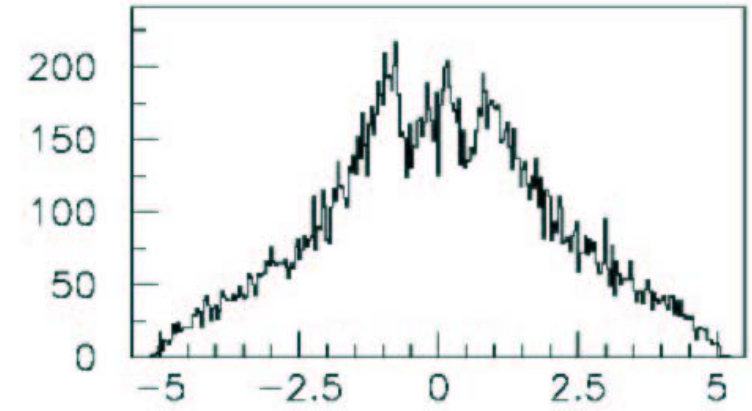
- Slice the stations into octets.
- Exploits the fact that the triplets must project into the beamline.

Reducing Invalid Triplets (2)

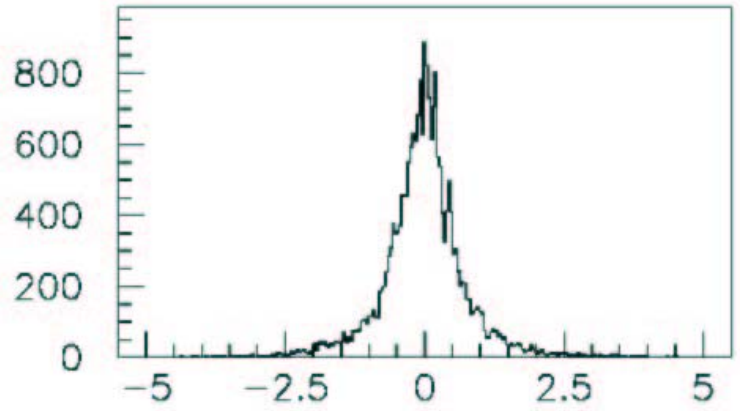


- Check 5 adjacent stations vs. 3 adjacent stations.
- Will increase the number of unfindable tracks from 540 to 1171.
- Low momentum will most likely need to be cut.

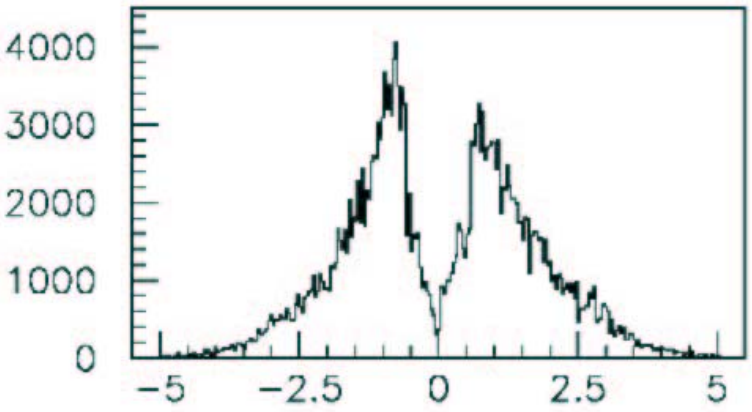
Reducing Invalid Triplets (3)



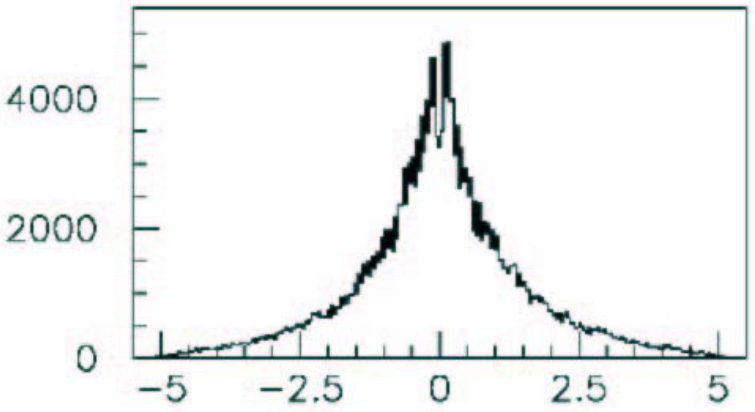
valid triplets x2



valid triplets x3−x1

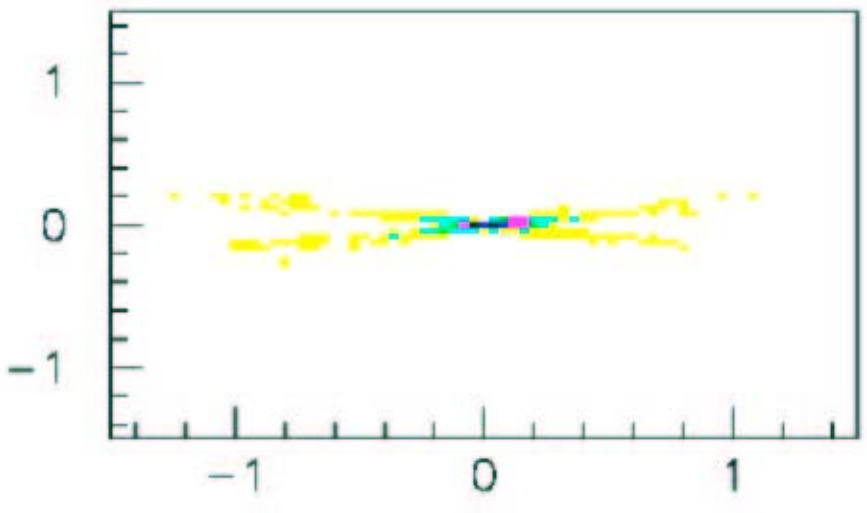


invalid triplets x2

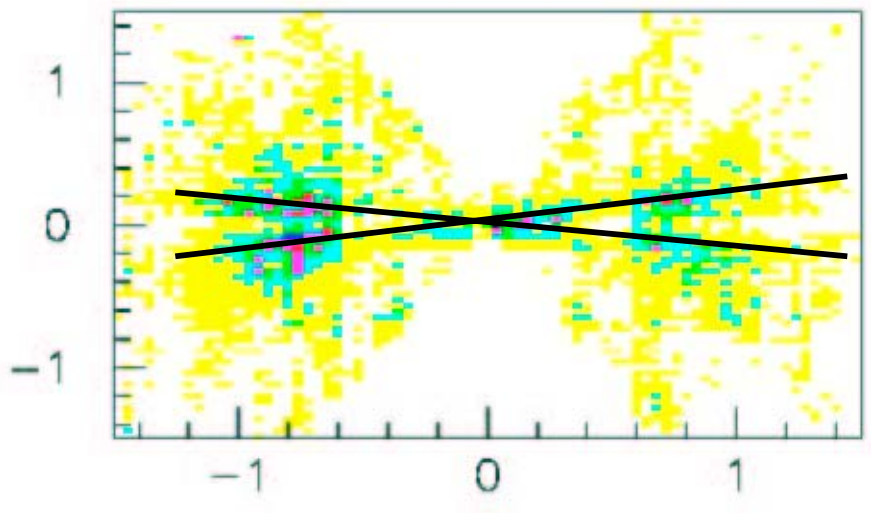


invalid triplets x3−x1

Reducing Invalid Triplets (4)



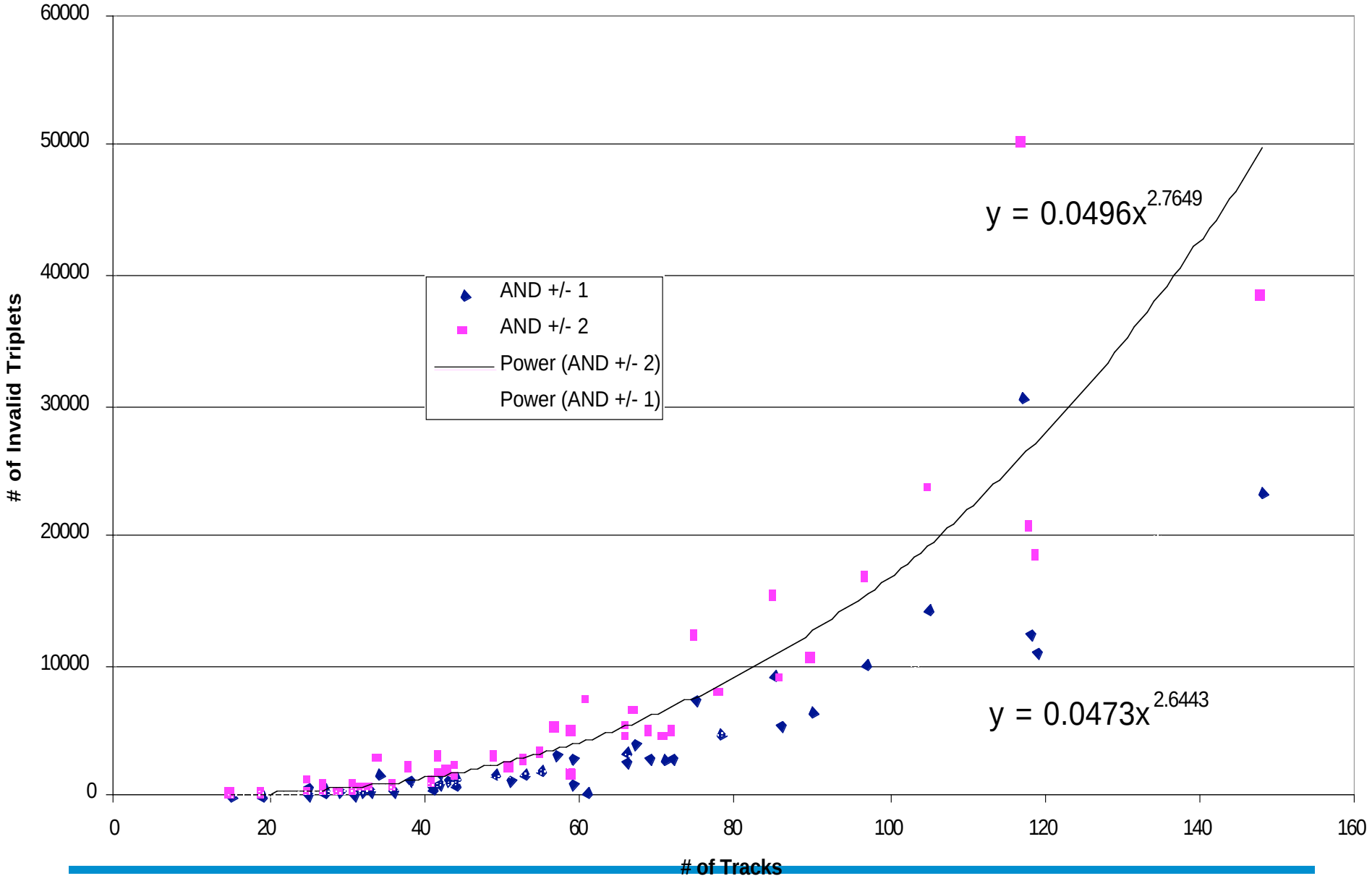
Valid Triplets: X_2 vs X_{31}



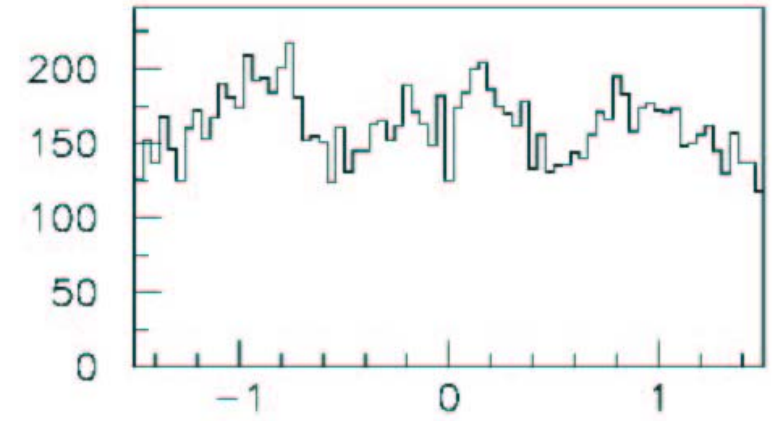
Invalid Triplets: X_2 vs X_{31}

Questions/suggestions?

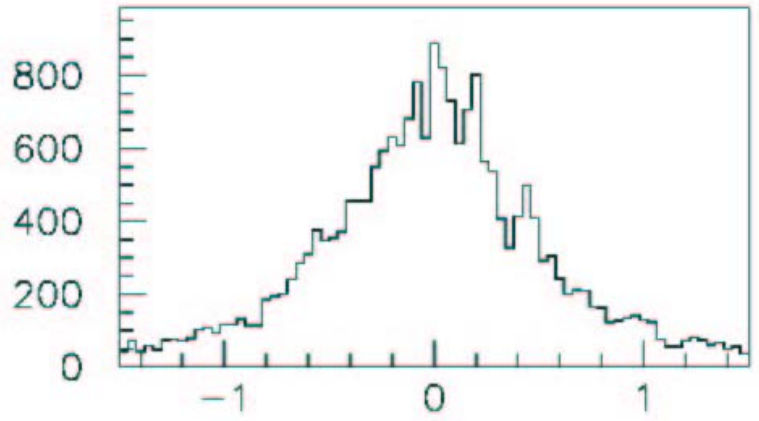
Tracks vs. Invalid Triplets



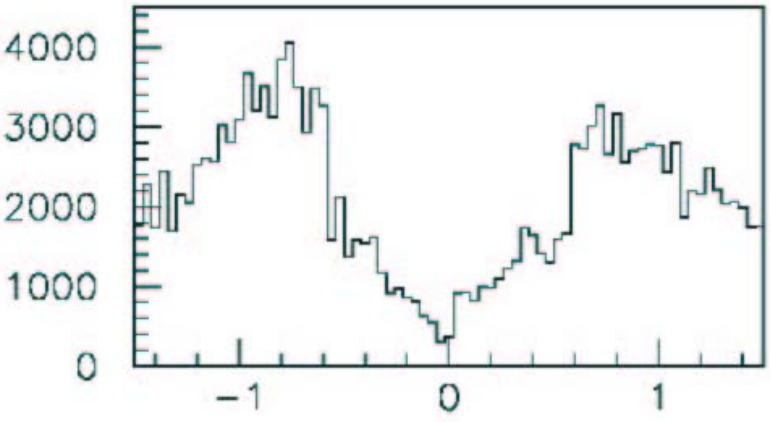
Reducing Invalid Triplets (3)



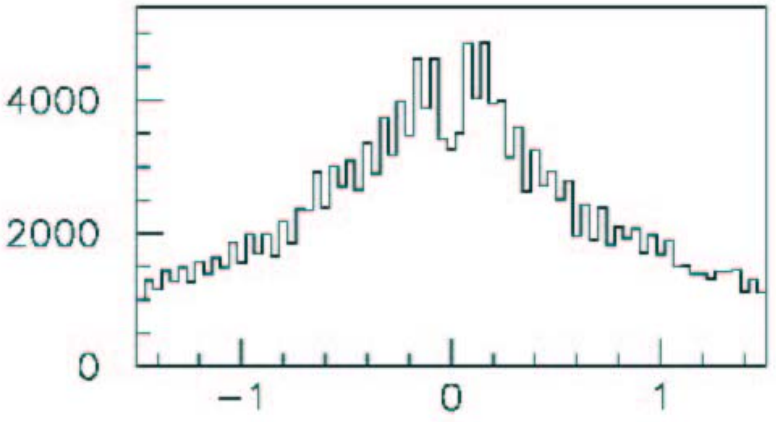
valid triplets x2



valid triplets x3−x1

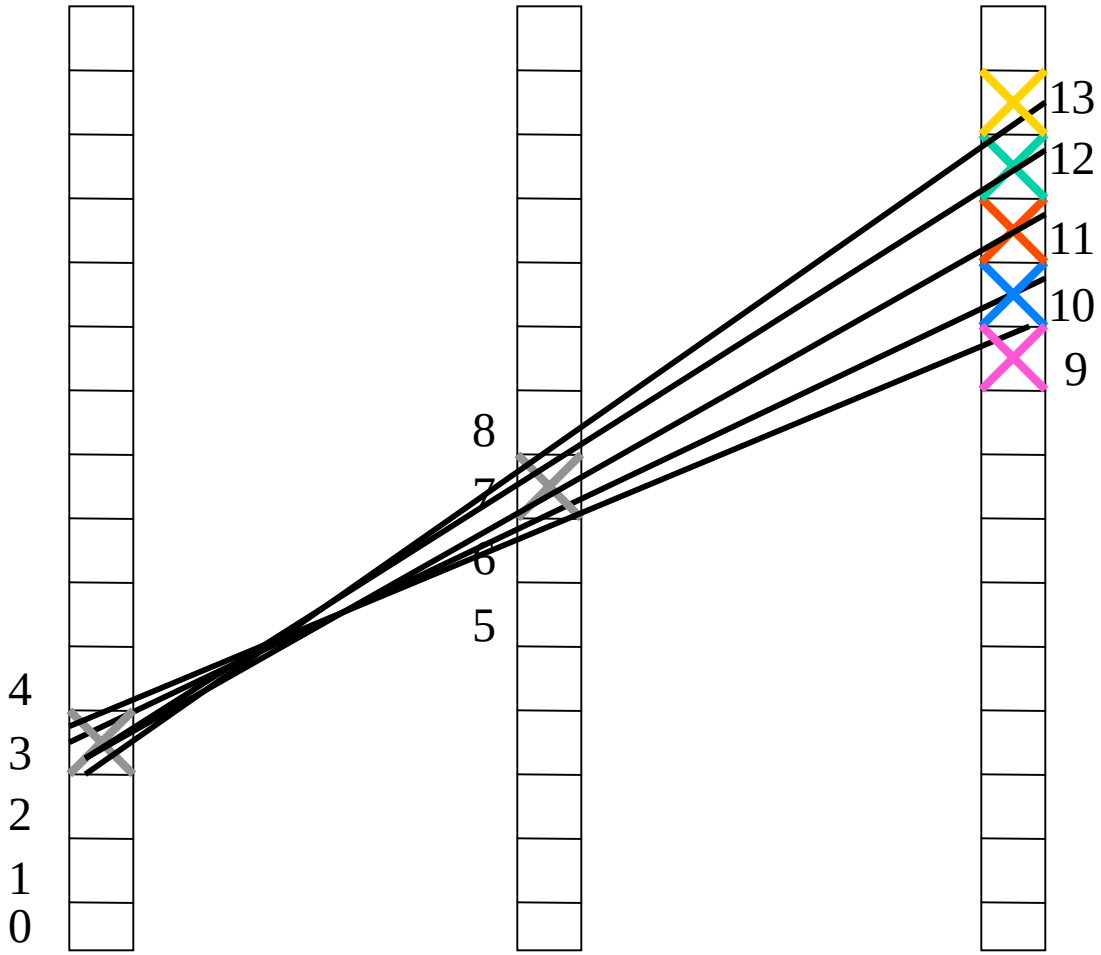


invalid triplets x2



invalid triplets x3−x1

The Rounding Problem (2)



- ✕ Designates a bin corresponding to a pixel hit.
- Notice borderline hits.

The Rounding Problem (2)