# MAPS – Beam Test: tracking efficiencies, Part III in a Saga

*Revised results – with dead areas implemented*

Jamie Ballin

HEP, Imperial College, London
j.ballin06@imperial.ac.uk

18th March 2008

# Overview

- This is a follow on to 29th February slides, `http://www.hep.ph.ic.ac.uk/calice/mapsMeetings/080229ral/ballin.pdf`
- Here are some revised results...
  - Minor correction to $\chi^2$ error parameter
  - Inclusion of dead areas and global geometry
- Also included,
  - A basic event display

# Fit parameter $e_{x,y}$ used in $\chi^2$ calculation

Last time I quoted,

$$\sigma_0 = 1.25\sigma_{\text{fit}}$$

for $\sigma_{\text{fit}}$ the width of the residual distribution, and $\sigma_0$ the intrinsic error on the sensor. This should in fact be,

$$\sigma_{\text{fit}} = 1.25\sigma_0$$

since the width of the residual error is a convolution of the error in the track and the error intrinsic error of the sensor. Hence $\sigma_0 < \sigma_{\text{fit}}$. The 1.25 factor is a consequence of our particular geometry.
This then implies,

$$e_{x,y} = (0.026, 0.02) \tag{1}$$

in mm, IF you do the fit to pixel coordinates. One then gets a flat $\chi^2$ probability $p_x, p_y$ distribution. This will now be improved on in the next Section.

# Overview

- Sensor includes substantial dead areas
- More dead areas in $x$ than in $y$ due to readout column architecture
- While beam particles have $\theta_z \sim 10\mathrm{mrad} =$ a few pixels, ones may clip dead areas and hence not be confirmed as “fourth hits”.
- This is a small effect, but one to get right!
- Will use official MAPS diagram, `http://www.hep.ph.ic.ac.uk/calice/mapsMeetings/070831ral/mapsCoordinates.pdf`

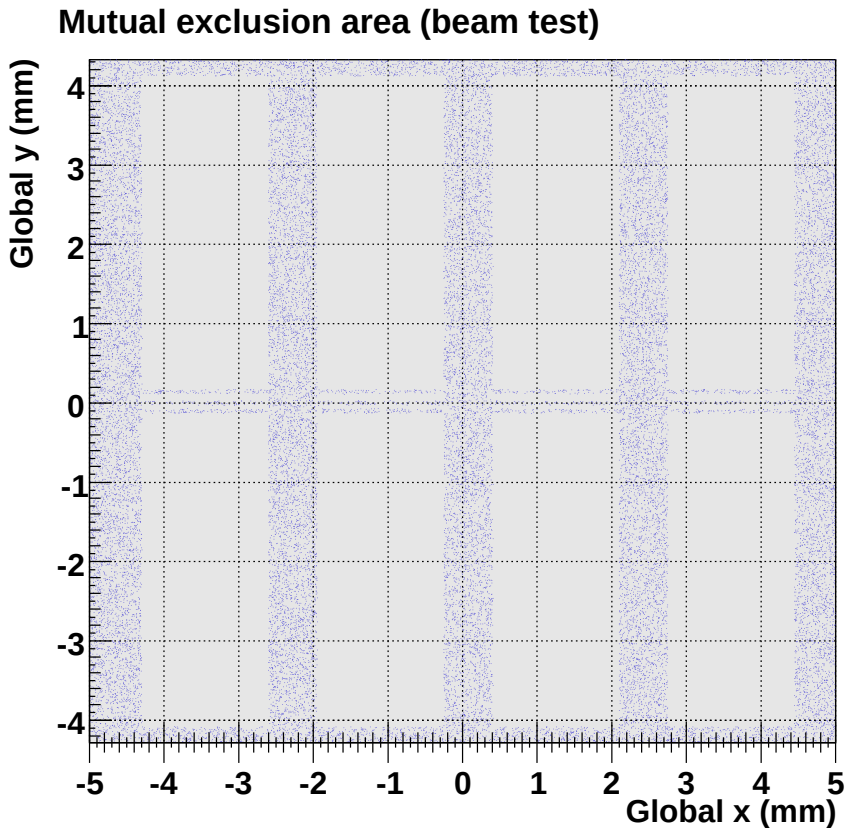
# Implementation

1. Always record raw pixel hits when creating tracks
2. Let each sensor have an angle $\phi$, which represents its clockwise rotation angle w.r.t. global coordiate system
3. Convert a pixel hit to real physical location by,
   - 3.1 Mapping hit to a local $(x, y)$ mm system
   - 3.2 Rotating it by $-\phi$
   - 3.3 Aligning it
4. Methods are provided in `MapsSensor` to...
   - 4.1 Query whether a global position in $(x, y)$ hits a dead area of the sensor
   - 4.2 Convert a global $(x, y)$ to a pixel coordinate in the sensor, where possible
5. This is pretty awkward and tedious stuff! Save yourselves the work of reimplementing it if possible.

CALICE Calorimeter for ILC

# Dead areas as seen by beam

## Shaded areas are to be excluded from the efficiency calculation



# Typical output of a 4 hit track

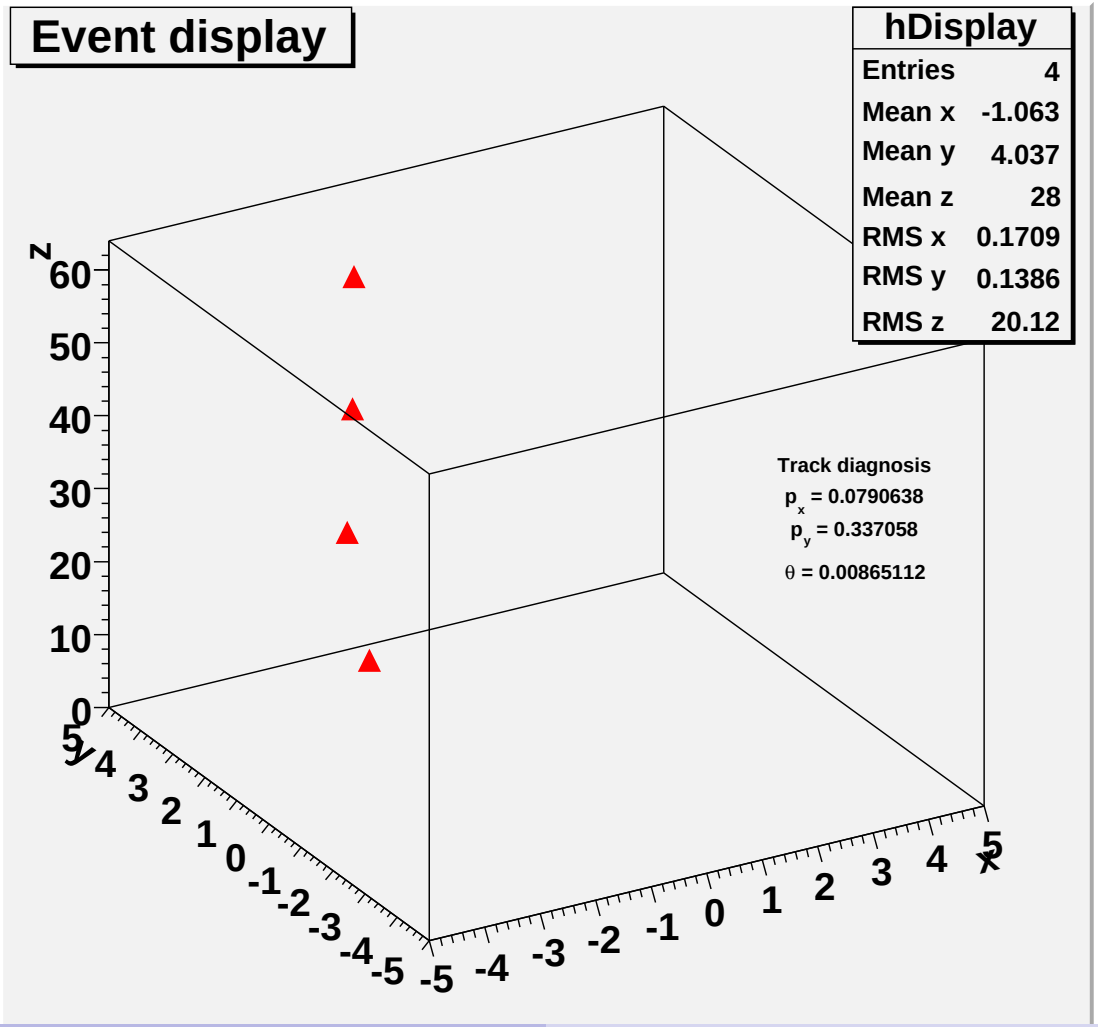
Invoking `diagnose(std::ostream& s, const MapsTrack& t)` shows what happens, Track at BX: 2526, hit pattern:

Sensor id 2 [z=36.000 phi_d=180.000, al=(0.000, 0.000)] : (159, 73) [(9, 95)], (-4.275, 0.550)
Sensor id 6 [z=54.000 phi_d=0.000, al=(0.000, 0.000)] : (5, 98) [(5, 98)], (-4.175, 0.750)
Sensor id 7 [z=18.000 phi_d=0.000, al=(0.000, 0.000)] : (7, 99) [(7, 99)], (-4.075, 0.800)
Sensor id 8 [z=0.000 phi_d=180.000, al=(0.000, 0.000)] : (158, 71) [(10, 97)], (-4.225, 0.650)
chi2X: 2.175 chi2Y: 3.675 p: (-4.180, -0.000) q: (0.680, 0.000)
chi2ProbX: 0.337 chi2ProbY: 0.159
theta: 0.000 meanX: -4.188 meanY: 0.687

So a hit at (159, 73) for $\phi = 180$ goes to (-4.275, 0.550) in the global coordinate system. (The square–bracketed entry is a cross check $[(168 - x, 168 - y)]$ of what the pixel coordinate would be were the sensor not rotated.)

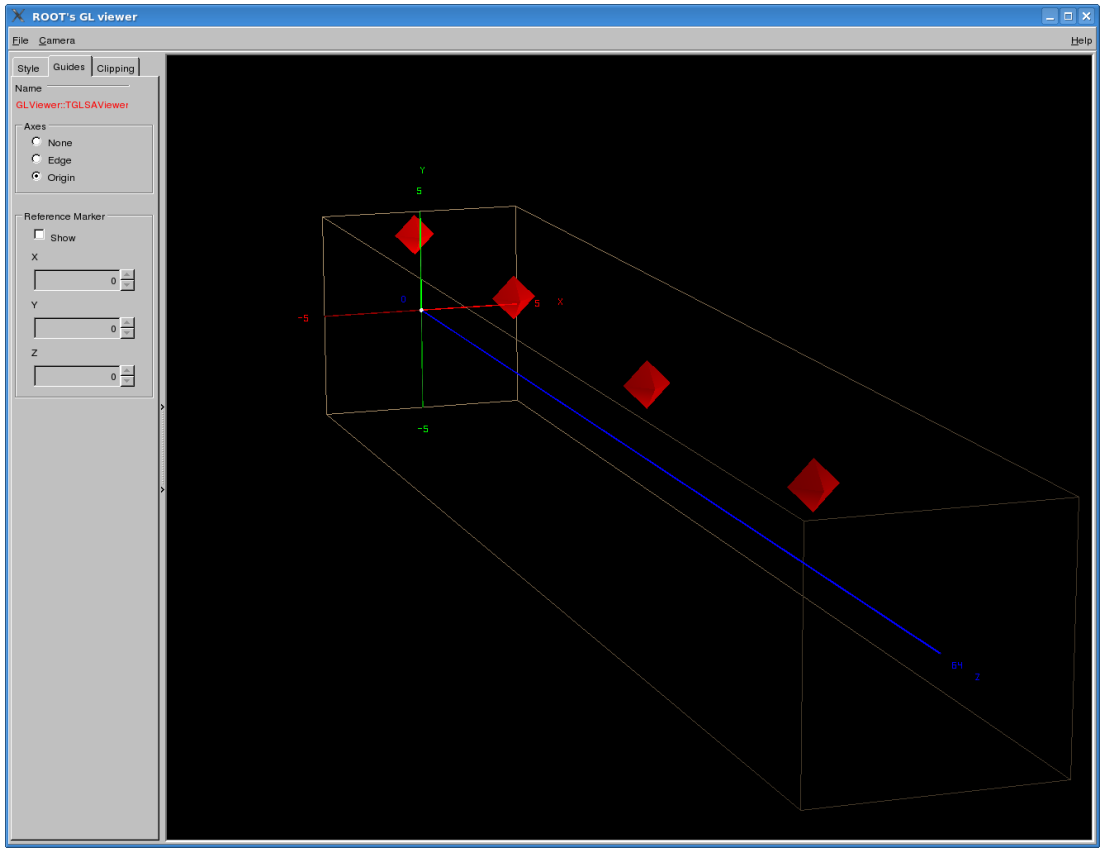
# Prototype event display
## Using a TH3F

Use `DisplayTrack` tool to loop over tracks. . .



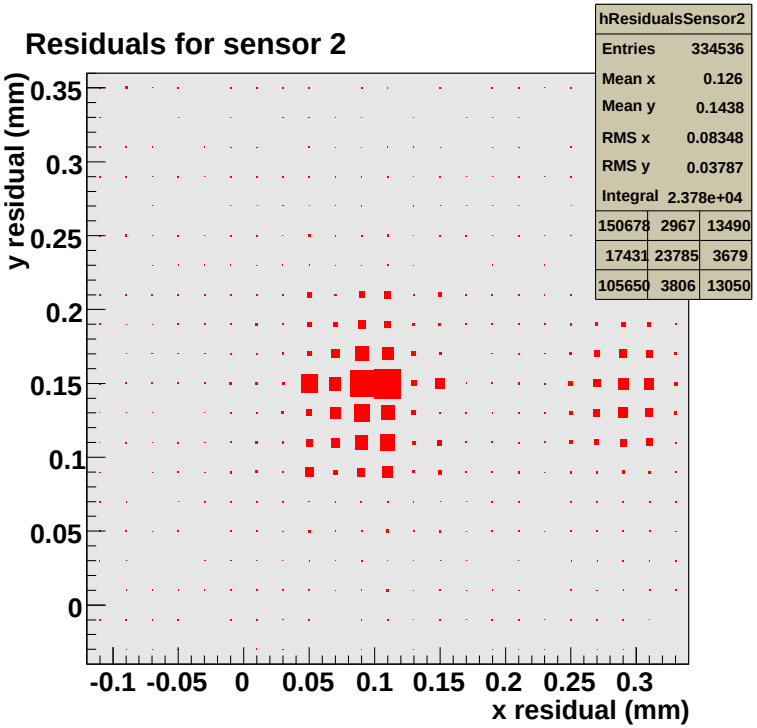
# Prototype event display
## OpenGL display in ROOT

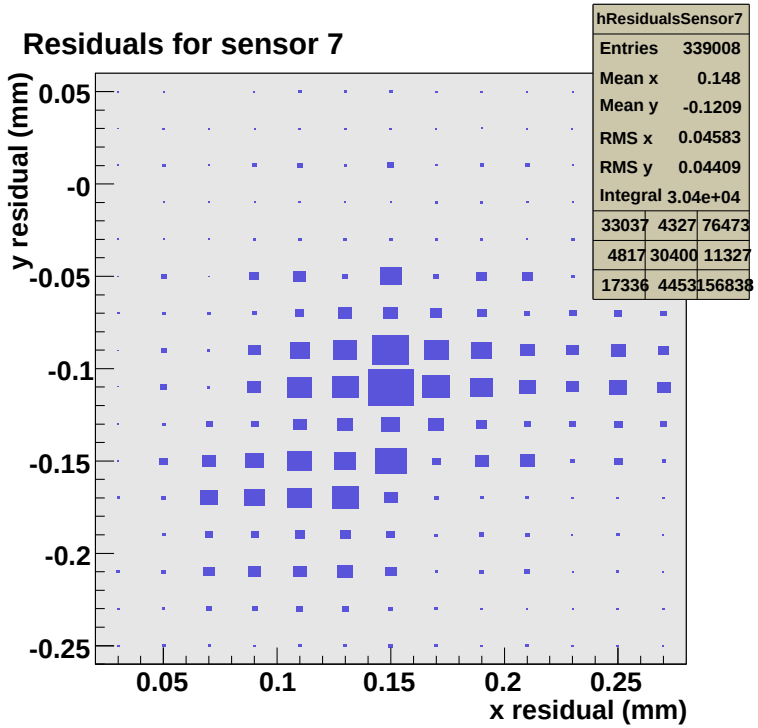
Go to View -> View With. . . -> GLViewer



# Alignment

## Using physical system tracking rather than pixel system yields new residuals




$r_{x,y|2} =$
$(0.092 \pm 0.019, 0.143 \pm 0.026)$

$r_{x,y|7} =$
$(0.151 \pm 0.027, -0.102 \pm 0.019)$

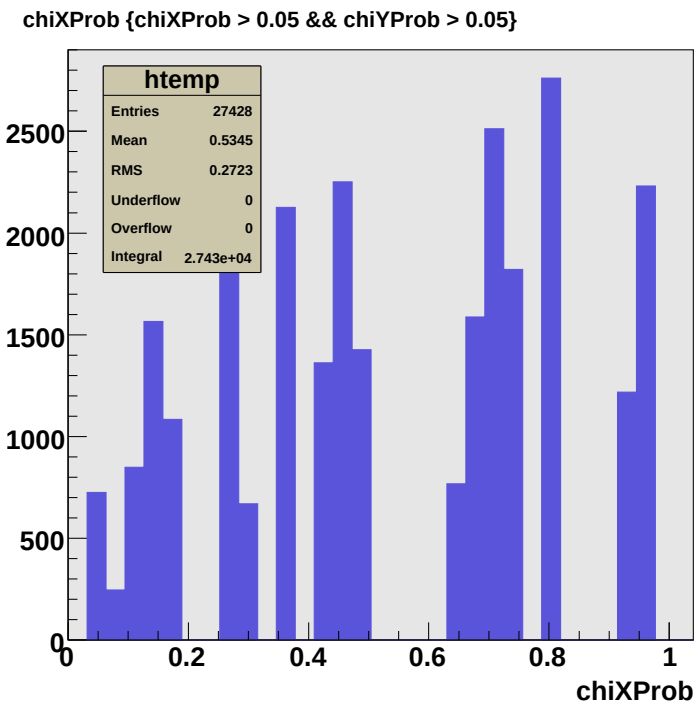
# Alignment

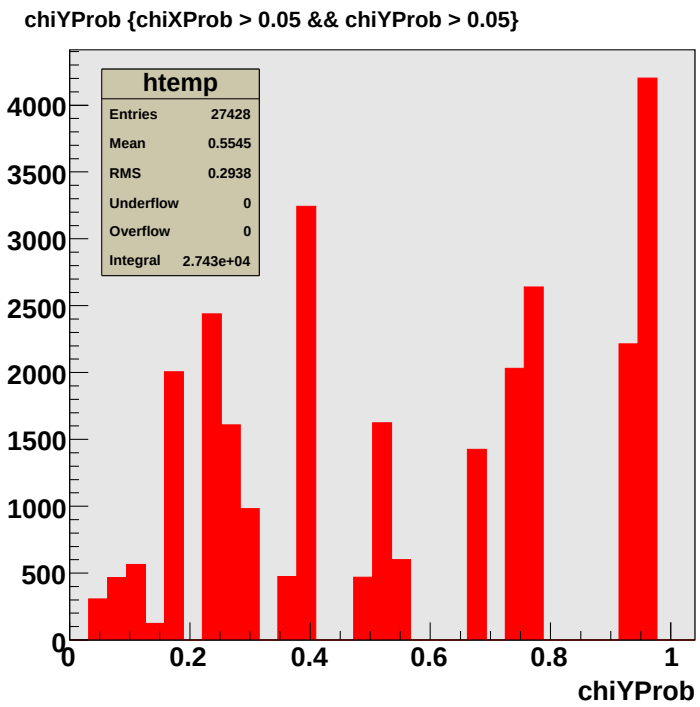
- Ghosting appears reduced
- Absolute alignment appears better than before
- Do we expect this?
- $\Rightarrow$ take new error parameters as,

$$e_{x,y} = (0.019, 0.018) \simeq (0.018, 0.018) \tag{2}$$

- Apparent reduction in $x$ ghosting has improved $e_x$

# $p_x, p_y$ probability distributions



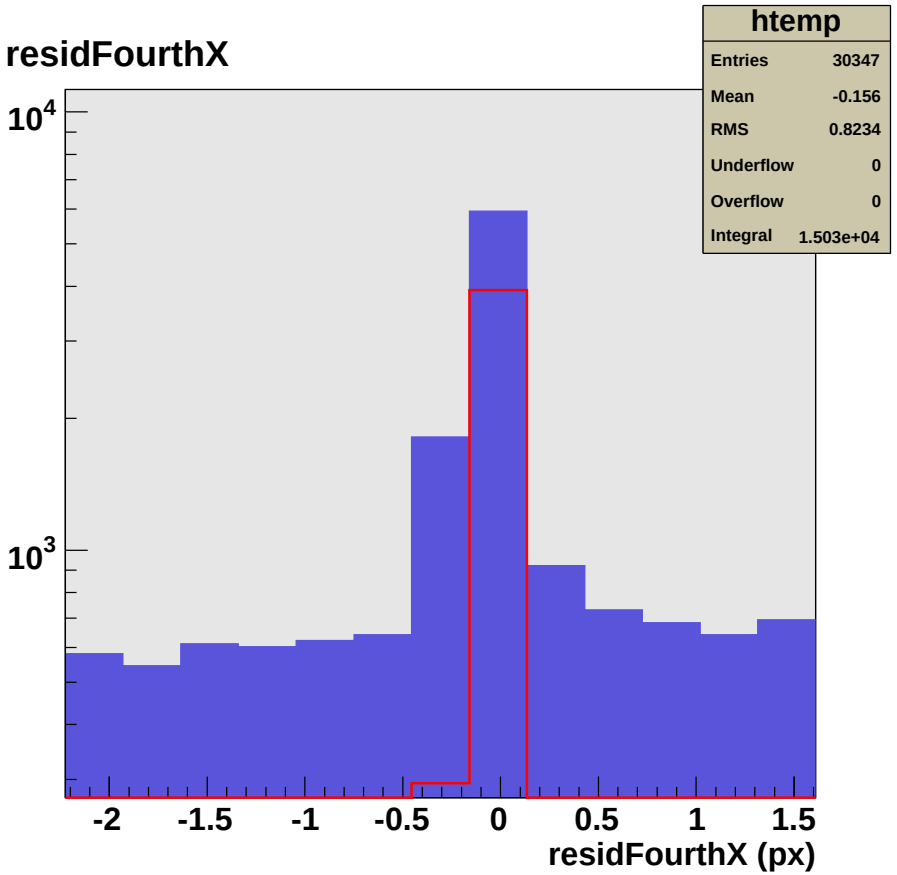




- Less steeply biased to 1 than before
- (Explanation: done with the aligned system, so residuals and errors change yet again)

# $\theta_z$ largely unaffected

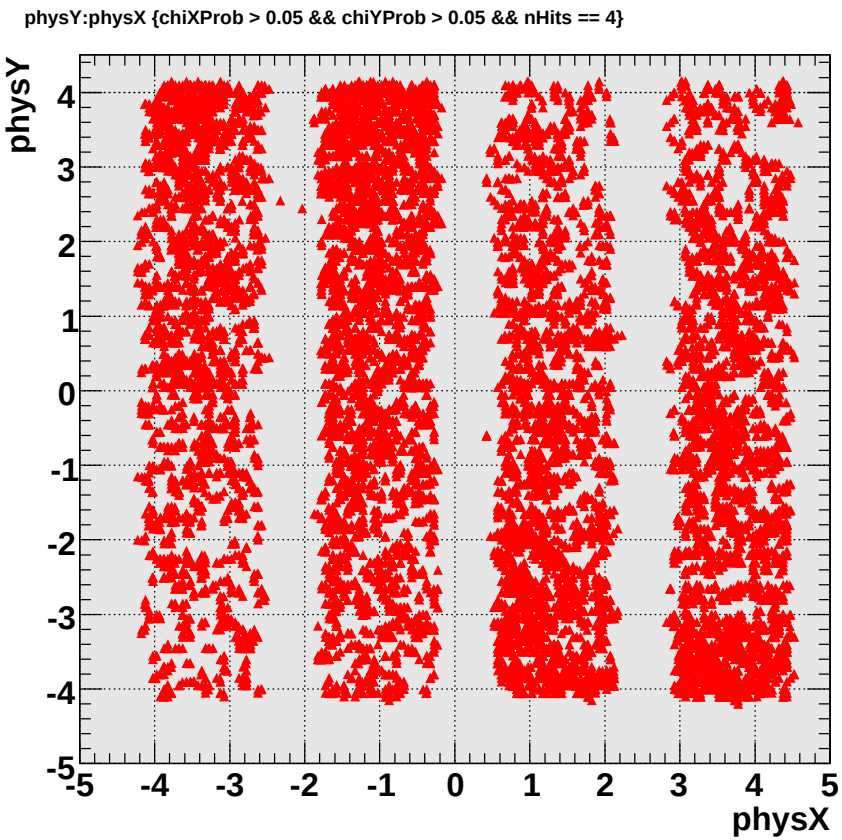
- From this we see than $\bar{\theta}_z \sim 5\mathrm{mrad} \sim 5$ pixels
- We'll revist this shortly

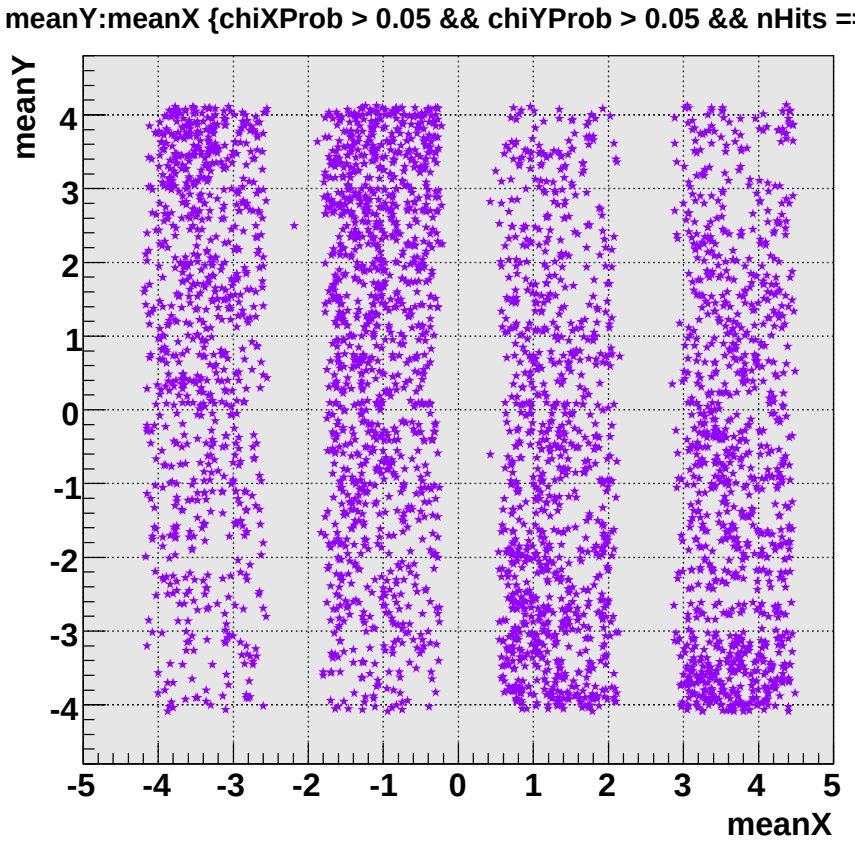
# Alignment consistency




- Good four hit track  All four hit tracks
- Fourth hit residuals are $\sim$ zero!

# Physical $x$, $y$ with good 4 hit tracks




- As seen in global aligned system
- $\Rightarrow$ hits in “dead areas” are from sensors 2 and 7 (they’re not actually in dead areas, it’s a consequence of their physical misalignment)

# Tracks rejected due to dead area intersection

Output of `ExtractEfficiencies ...`

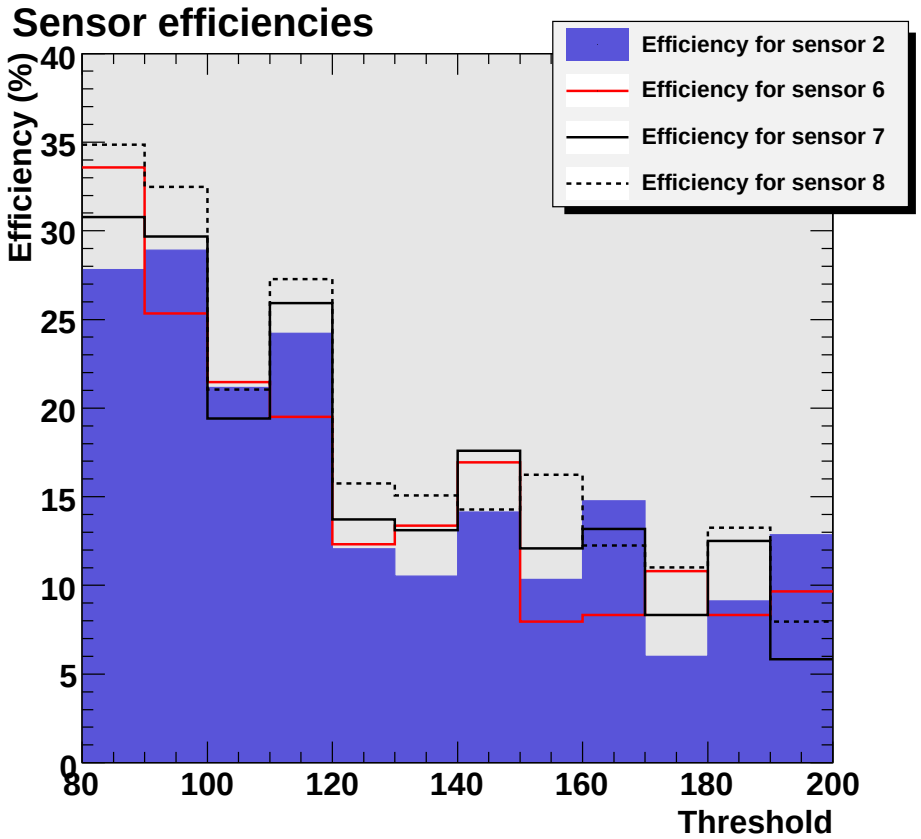
ExtractEfficiencies: summary: Total candidate tracks: 28394
Efficient hits: 3819
Inefficient hits: 23466
Dead area intersections: 1109

- Of the 34,339 candidate tracks 1,405 are zapped: they intersect with the 4th sensor's dead area $\Rightarrow$ unfair test?
- Average sensor efficiency,

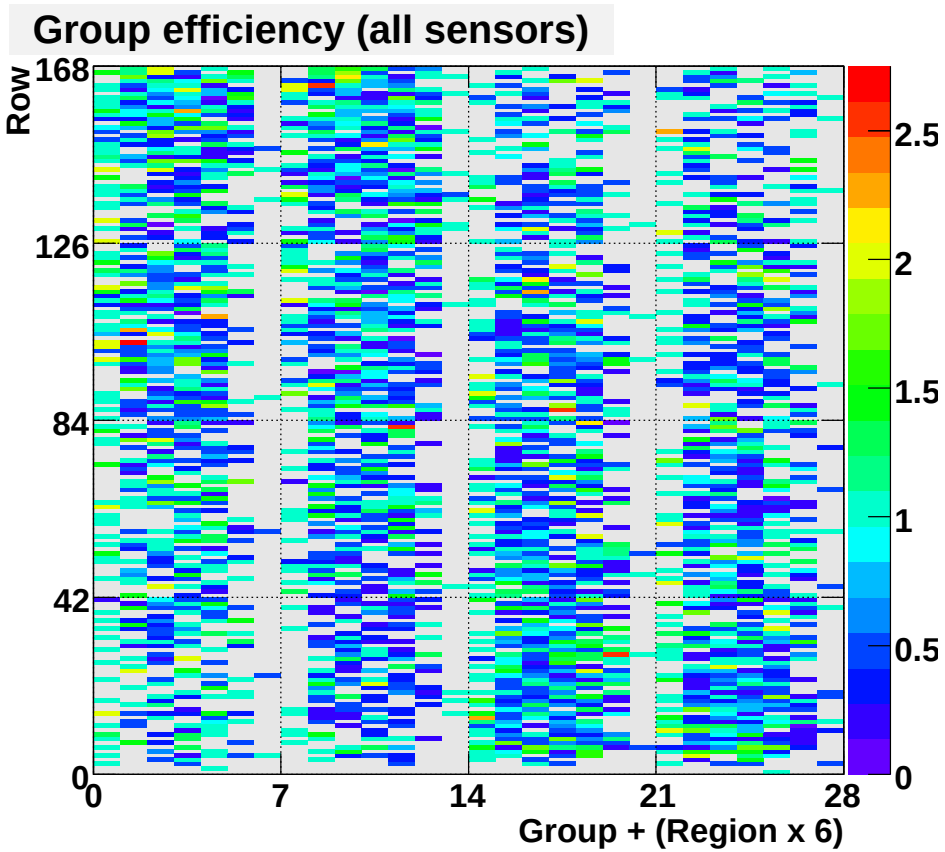
$$\bar{\epsilon} = \frac{\langle n_4 \rangle}{\langle n_3 \rangle + \langle n_4 \rangle} = \frac{3,819}{(3,819 + 23,466)} = 14.0\% \quad (3)$$

- Slight improvement from excluding unfair tests

# Latest sensor efficiencies



# Efficiency by group

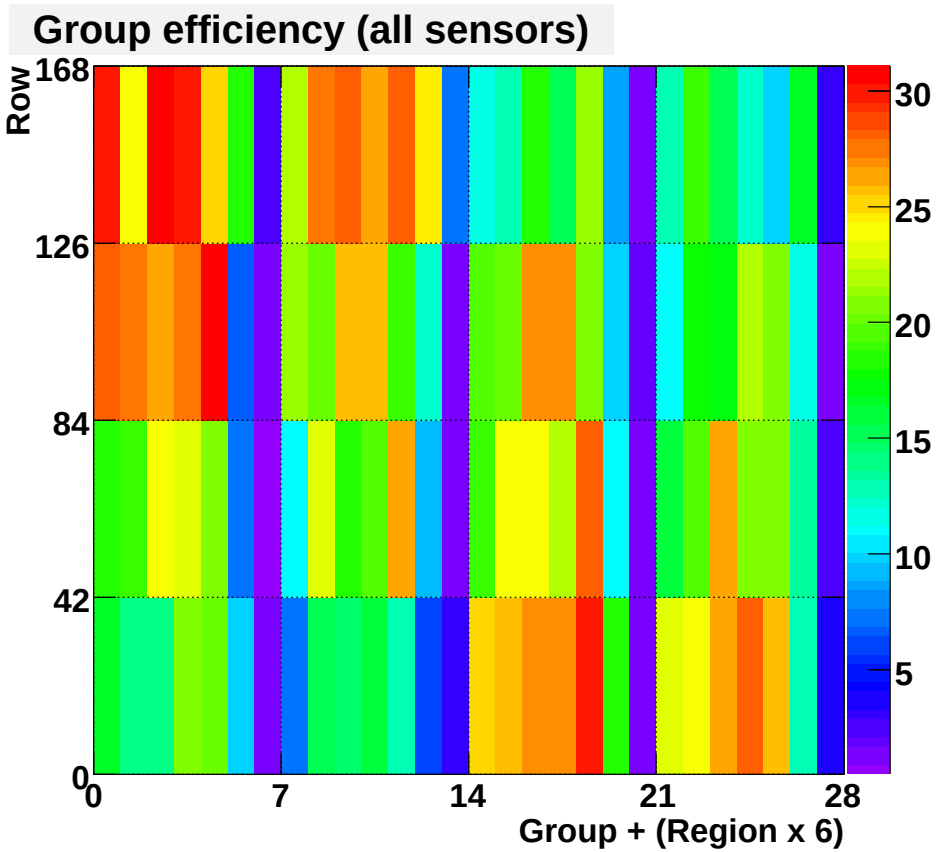



Scale is normalised to 4.0 = 400%; histo is $\sum_{\text{all sensors}}$

- Dotted lines at boundaries between regions $\Rightarrow$ dead areas
- It's weird that quiet areas are seen on the *right* of the regions, in contradiction to expectations[a]
- Consquence of "mutual dead area" exclusion
- Right of boundary Fewer tracks (mutual dead area exclusion)
- Left of boundary Fewer tracks AND lower efficiency

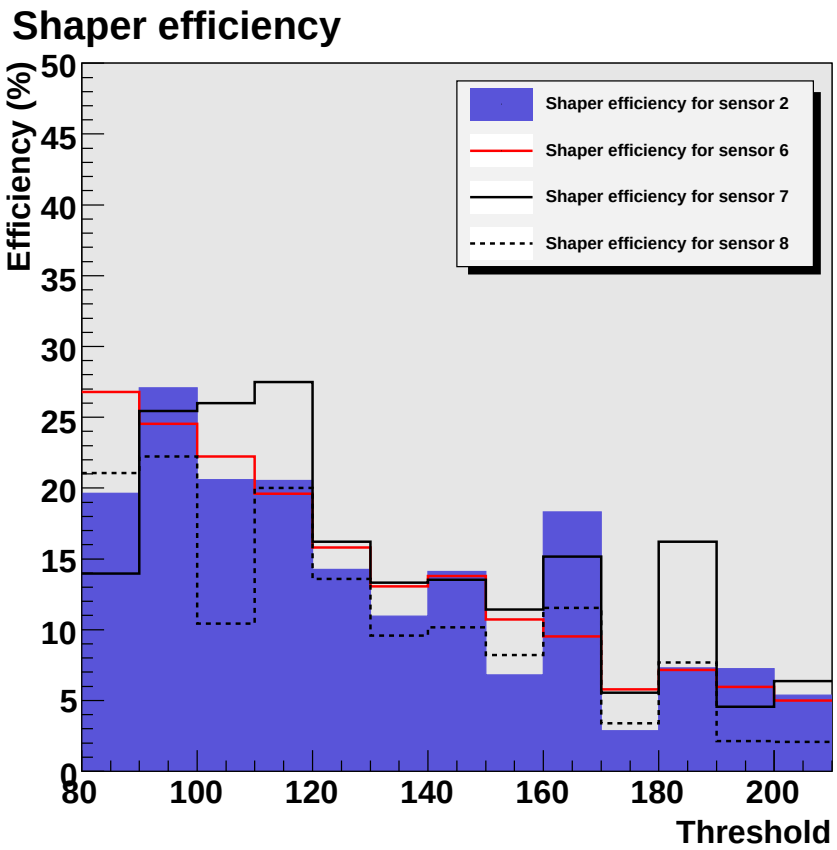
[a] *N.B.* This plot was made with raw pixel hits $\Rightarrow$ no funny rotation business

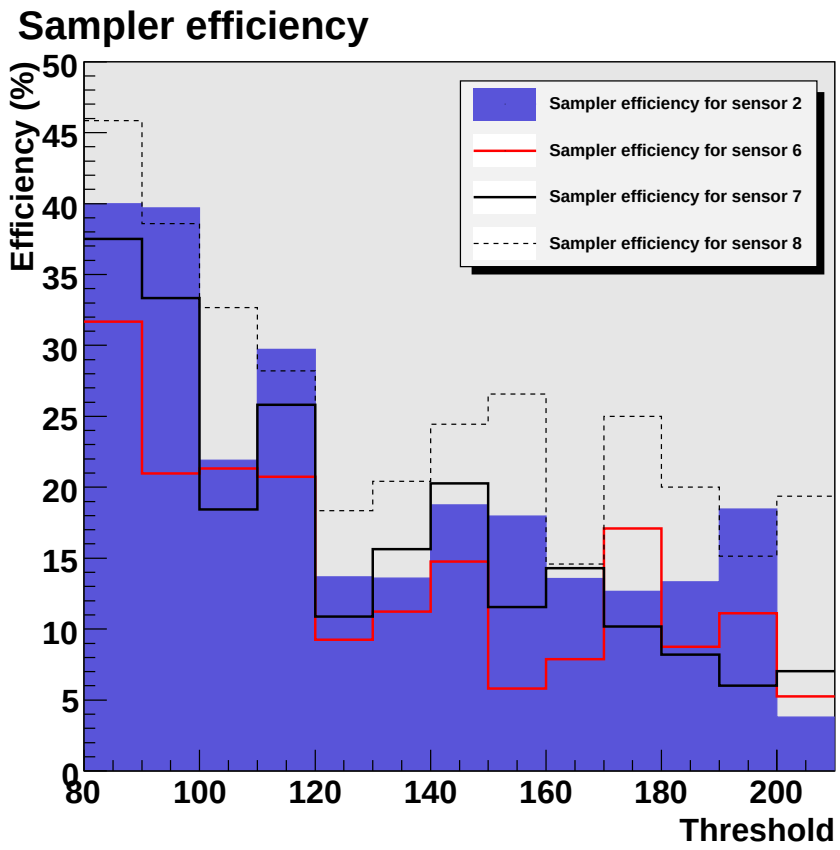
# Effect of averaging over *y* axis




- Partially explainable by $\theta_z \sim 5$ pixels, but not enough:
  - If $5/47$ pixels are dead, then this is at the level of 11% of all possible tracks.
  - But $1,106/(28,394) = 3.9\%$ of tracks are excluded from the efficiency calculation anyway.
  - Drop in efficiency is NOT accounted for by this effect
- Are the memory columns draining charge?
- Are hits on the right edge of the region not getting written into memory?

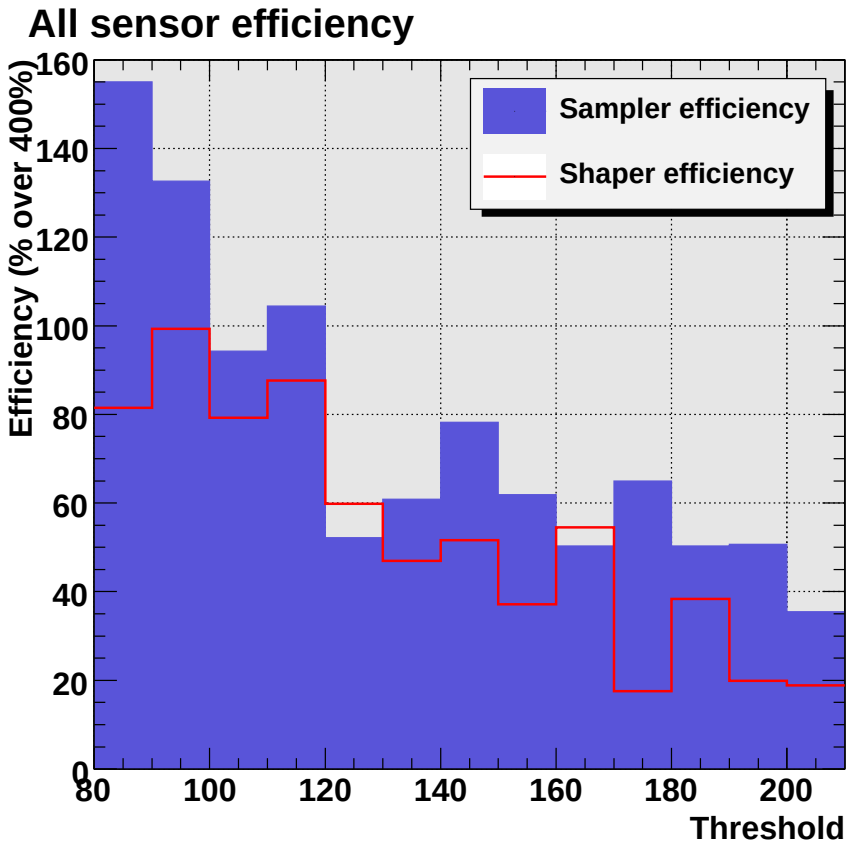
# Shapers and Samplers



# Shapers and Samplers, all sensors

All sensors added together (i.e. normalise histogram to 400%)




Samplers are more efficient than shapers