Correcting bias on the edges of particle tracking detectors

Peter Voerman

Nikhef

May 24, 2024

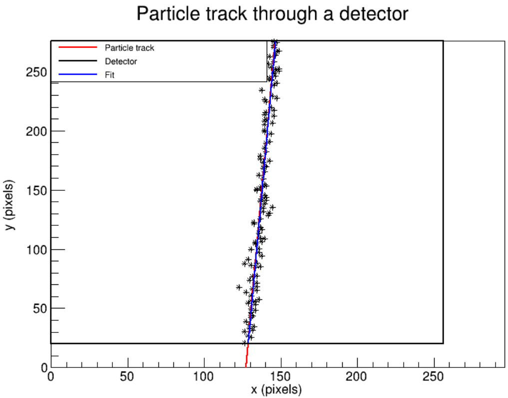
Outline

In this presentation, I will discuss the following questions:

- How does bias on detector edges occur?
- Does the bias show up in simulations?
- Is it possible to find a function that fits the bias?
- Can this function be used to correct the bias?

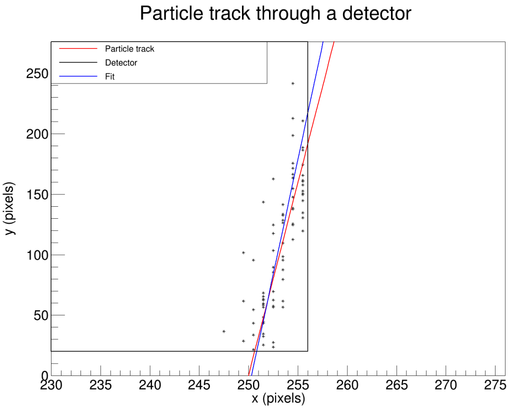
Simulation setup

- Based on a Timepix3 detector
- Consists of 256x256 pixels with a size of 55x55 μm^2
- Each track is defined by a random angle and starting position on the x-axis
- Diffusion is modeled by adding a random number from $\mathcal{N}(0, \sigma^2_{\text{diffusion}})$ to the hit position



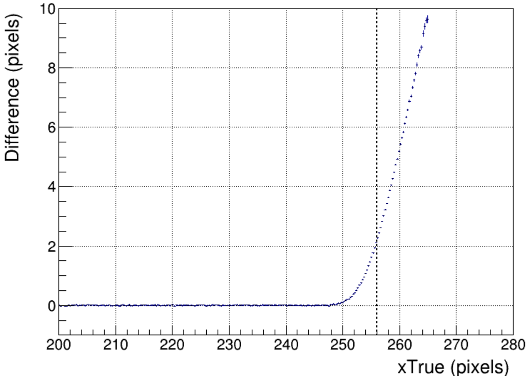
Bias

- Close to the edge of the detector, more hits are registered on one side of the track
- This leads to biased track parameters when fitting the data



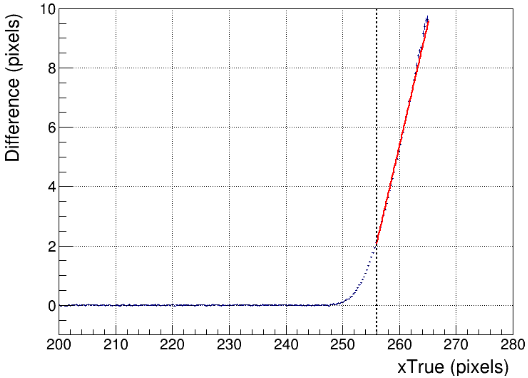
Measurement bias

- Close to the edge of the detector, the average difference between the true and measured x-coordinates grows



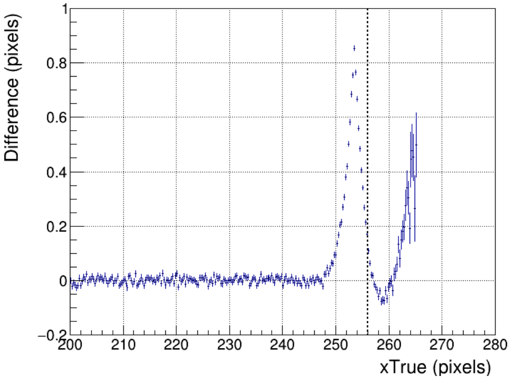
Fitting the bias

- If we can find a function that fits the bias, we might be able to correct it
- Since the last part of the data looks linear, one option could be:
- $f_{bias}(x) = \max(0, p_0(x - p_1))$



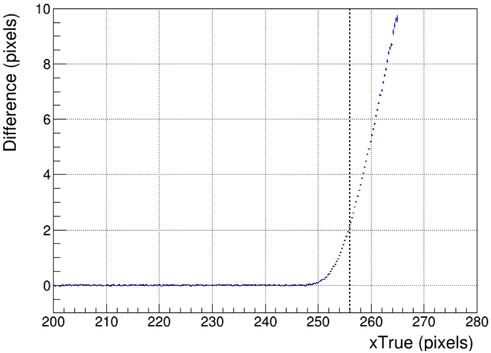
Fitting the bias

- This plot shows the same data, but with $f_{bias}(x)$ subtracted from it
- Most of the bias is removed, but some still remains



Fitting the bias

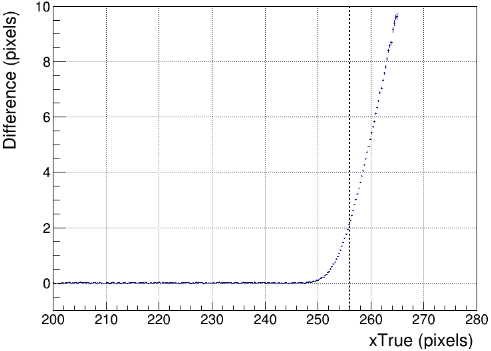
- Solution: add a quadratic part before the linear part
- $f_{bias}(x) = \begin{cases} 0 & \text{if } x < p_1 \\ \frac{(x-p_1)^2}{p_0} & \text{if } p_1 < x < p_2 \\ \frac{2(p_2-p_1)}{p_0}(x-p_2) + \frac{(p_2-p_1)^2}{p_0} & \text{if } p_2 < x \end{cases}$



Nikhef

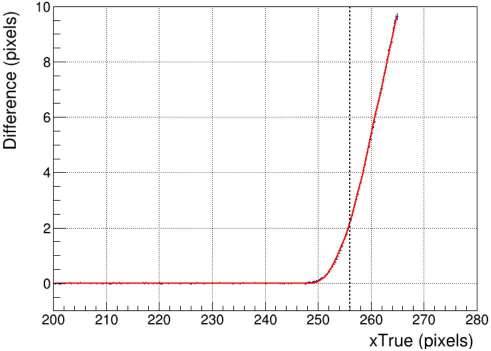
Fitting the bias

- Solution: add a quadratic part before the linear part
- $$f_{bias}(x) = \begin{cases} 0 & \text{if } x < p_1 \\ \frac{(x-p_1)^2}{p_0} & \text{if } p_1 < x < p_2 \\ \frac{2(p_2-p_1)}{p_0}(x-p_2) + \frac{(p_2-p_1)^2}{p_0} & \text{if } p_2 < x \end{cases}$$



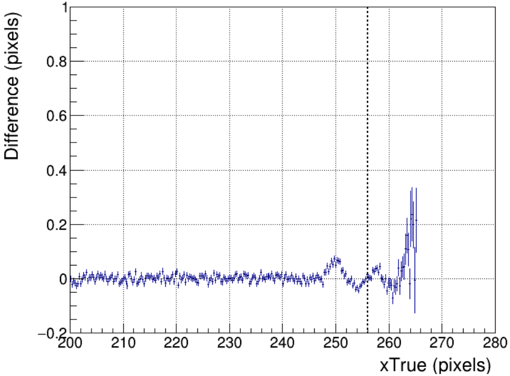
Fitting the bias

- Solution: add a quadratic part before the linear part
- $f_{bias}(x) = \begin{cases} 0 & \text{if } x < p_1 \\ \frac{(x-p_1)^2}{p_0} & \text{if } p_1 < x < p_2 \\ \frac{2(p_2-p_1)}{p_0}(x-p_2) + \frac{(p_2-p_1)^2}{p_0} & \text{if } p_2 < x \end{cases}$



Fitting the bias

- Now, the bias is almost completely removed
- We can use this function to correct the bias found in the track parameters

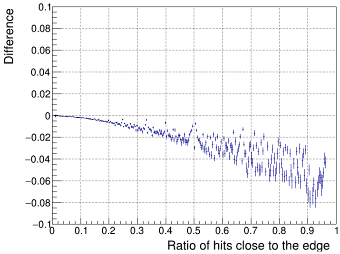


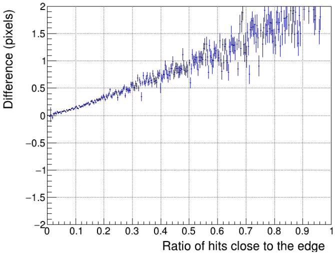
Track fitting

- Find the track parameters that best fit the data by minimizing χ^2
- $\chi^2 = \sum_{i=1}^{N} \frac{(a+by_{t,i}-x_{m,i})^2}{\sigma_i^2}$
- To start with, the uncertainty in the y-dimension is assumed to be zero

Track parameter bias

- As more hits are close to the edge, the bias on the track parameters grows



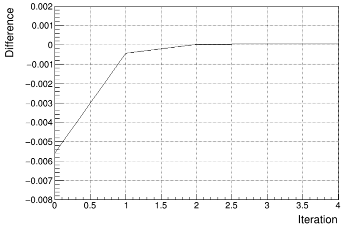


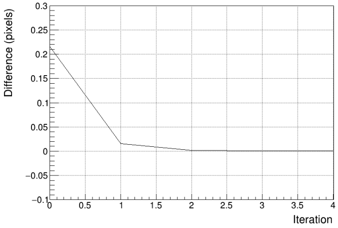
Correcting the bias

- To correct the track parameter bias, $f_{bias}(x)$ is added to the χ^2 formula
- $\chi^2 = \sum_{i=1}^{N} \frac{(a+by_{t,i}-x_{m,i}-f_{bias}(x_{t_i}))^2}{\sigma_i^2}$

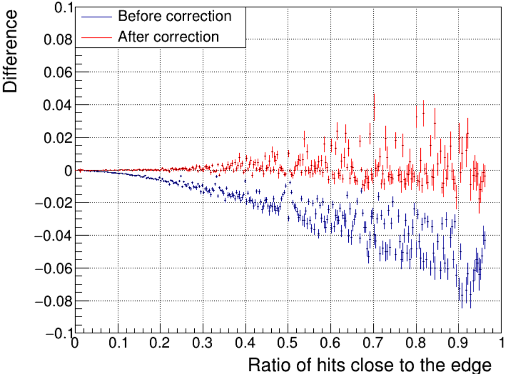
Correcting the bias

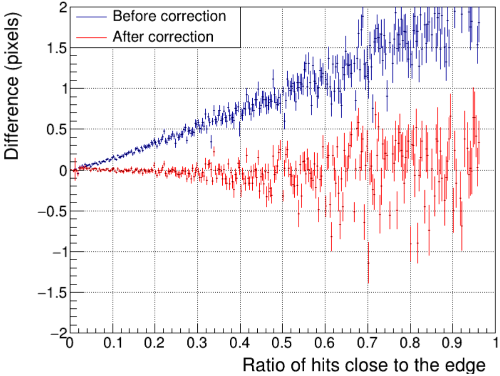
- The true values of the x-coordinates are not known during the fit
- Extra iterations are needed to converge to the correct track parameters
- The fit converges after two extra iterations





Correcting the bias





Correcting the bias

- With measurement uncertainty in only one dimension, most of the bias can be corrected
- On average, the bias is reduced by 99.2% for the a parameter and by 99.5% for the b parameter

Uncertainty in both dimensions

- In real data, there is uncertainty in both dimensions
- To take this into account, a different χ^2 formula is needed

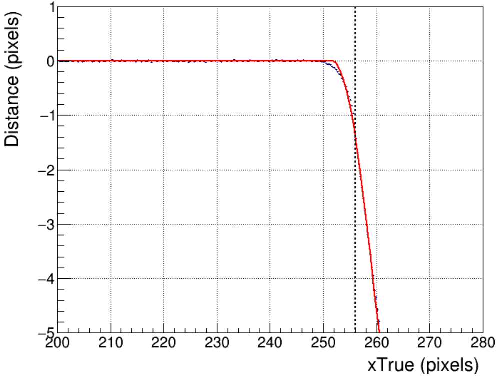
Uncertainty in both dimensions

- In real data, there is uncertainty in both dimensions
- To take this into account, a different χ^2 formula is needed
- $\chi^2 = \sum_{i=1}^{N} \frac{(\sin(\phi)x_{m,i} - \cos(\phi)y_{m,i} - d_0)^2}{\sigma_i^2}$
- The parameters ϕ and d_0 denote the angle and offset of the track

Nikhef

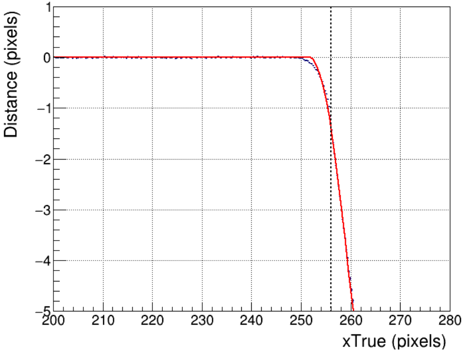
Uncertainty in both dimensions

- This figure shows the average distance between the measurements and the true track as a function of the true x-coordinates
- $d = \sin(\phi_t)x_m - \cos(\phi_t)y_m - d_{0_t}$



Uncertainty in both dimensions

- The same function can be used to fit the bias
- $$f_{bias}(x) = \begin{cases} 0 & \text{if } x < p_1 \\ \frac{(x-p_1)^2}{p_0} & \text{if } p_1 < x < p_2 \\ \frac{2(p_2-p_1)}{p_0}(x-p_2) + \frac{(p_2-p_1)^2}{p_0} & \text{if } p_2 < x \end{cases}$$

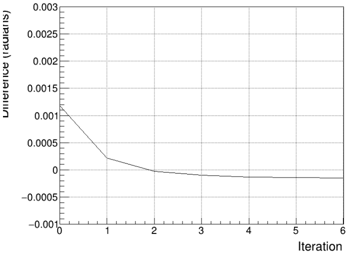


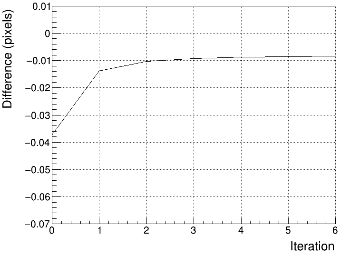
Correcting the bias

- Bias can now occur on the top/bottom edges as well as the left/right edges
- To correct the bias, two terms need to be added to the χ^2 formula
- $\chi^2 = \sum_{i=1}^{N} \frac{(\sin(\phi)(x_{m,i} - \frac{f_{bias}(x_{t_i})}{\sin(\phi_t)}) - \cos(\phi)(y_{m,i} - \frac{f_{bias}(y_{t_i})}{\cos(\phi_t)}) - d_0)^2}{\sigma_i^2}$

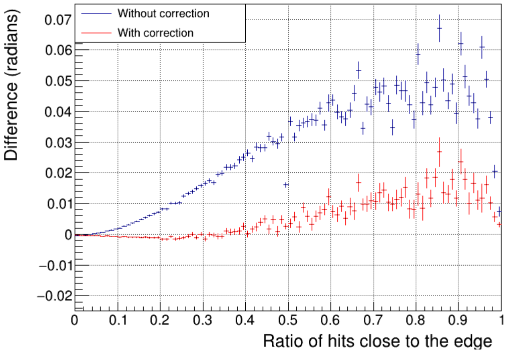
Correcting the bias

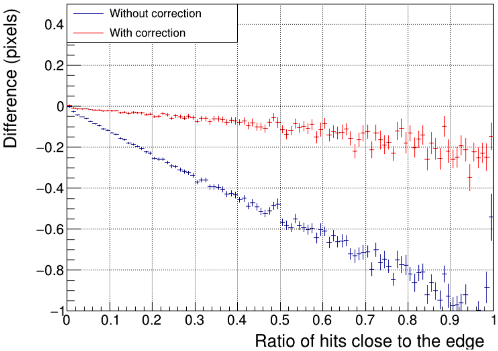
- Again, extra iterations are needed to converge to the correct track parameters
- In this case, four iterations seem to be enough



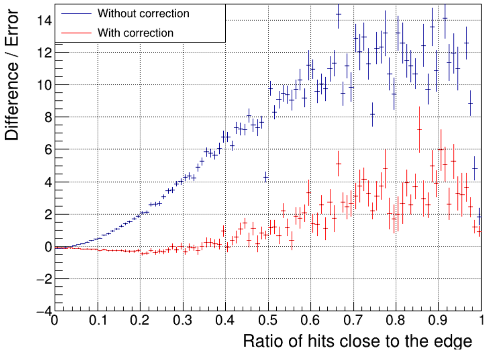


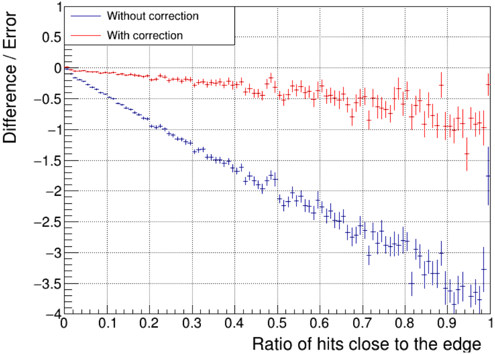
Correcting the bias





Correcting the bias



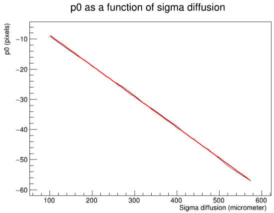


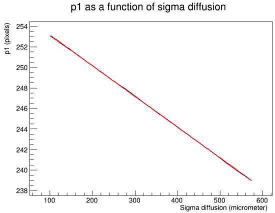
Computing the correction parameters

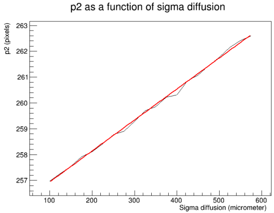
- $$f_{bias}(x) = \begin{cases} 0 & \text{if } x < p_1 \\ \frac{(x-p_1)^2}{p_0} & \text{if } p_1 < x < p_2 \\ \frac{2(p_2-p_1)}{p_0}(x - p_2) + \frac{(p_2-p_1)^2}{p_0} & \text{if } p_2 < x \end{cases}$$

Computing the correction parameters

- $$f_{bias}(x) = \begin{cases} 0 & \text{if } x < p_1 \\ \frac{(x-p_1)^2}{p_0} & \text{if } p_1 < x < p_2 \\ \frac{2(p_2-p_1)}{p_0}(x-p_2) + \frac{(p_2-p_1)^2}{p_0} & \text{if } p_2 < x \end{cases}$$

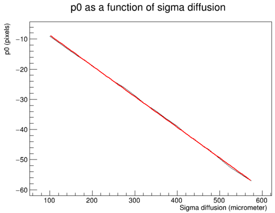


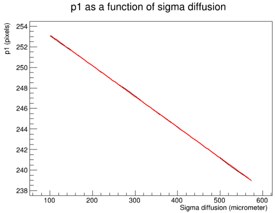


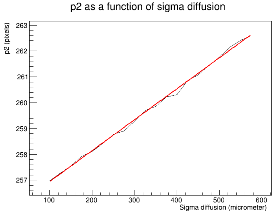


Computing the correction parameters

- $$f_{bias}(x) = \begin{cases} 0 & \text{if } x < p_1 \\ \frac{(x-p_1)^2}{p_0} & \text{if } p_1 < x < p_2 \\ \frac{2(p_2-p_1)}{p_0}(x-p_2) + \frac{(p_2-p_1)^2}{p_0} & \text{if } p_2 < x \end{cases}$$
- $p_0(\sigma_d) = -0.103\sigma_d + 0.5$
- $p_1(\sigma_d) = -0.029\sigma_d + 256$
- $p_2(\sigma_d) = 0.012\sigma_d + 256$







Conclusions

- Most of the bias can be corrected by adding an extra term to the χ^2 formula
- The average reduction is 92.3% for the angle and 84.3% for the offset
- The correction parameters are linearly dependent on the diffusion constant

Outlook

- In future research, the bias correction can be applied to different types of tracks:
  - Curved tracks
  - 3-dimensional tracks
  - Tracks that move through multiple detectors
- These methods can then be applied to real data
- Code is available at www.nikhef.nl/~s01/tpc_bias.tar.gz

Questions?