Making The StDev_Min Method More Robust

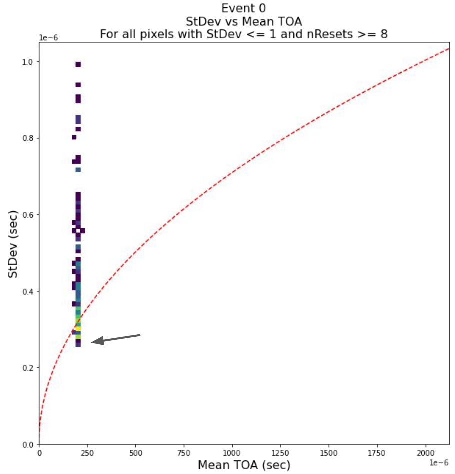
Dave Elofson, Dave Toback
Texas A&M University
Q-Pix Simulation Meeting
March 4th, 2022

Outline

- Recap: the StDev_Min Method and Known Pathologies
- Addressing Type 1 Pathologies: Delta_StDev Method
    - Simple attempt w/ t0=0 assumption
    - Iterative process to remove t0=0 assumption
- Addressing Type 2 Pathologies: Decreasing Reset Threshold
- Wrapping it up: Next Steps, Thoughts and Conclusions

Recap

The StDev_Min Method



Previously introduced the StDev_Min method:

For every event:

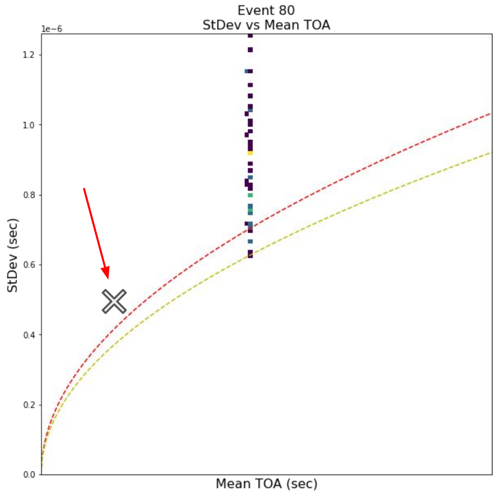
- Look at all pixels in an event with nResets ≥ 8
- Find the pixel with the minimum standard deviation of the reset times (since StDev only fluctuates low in a systematic way)
- Use the StDev_Min pixel from each event to get the functional form of Z vs StDev_Min
- Use the functional form to calculate the Z and t_0 of other pixels
- Use the StDev_Min pixel as the basis for the rest of the event reconstruction

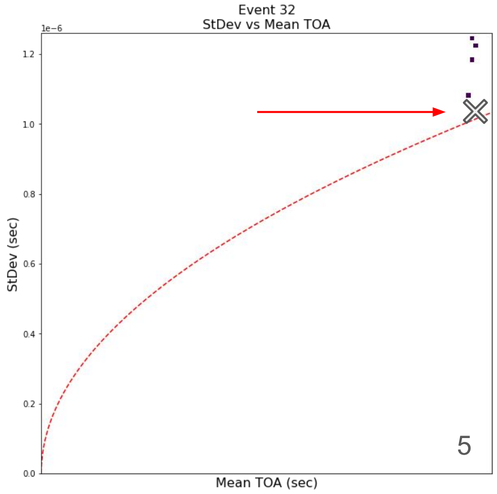
Two issues with the StDev_Min Method

The method is a very simple first try and there are two known reasons as to why the smallest StDev may not lead to a good measure of t0:

Type 1 Pathologies) Algorithm is susceptible to outlier pixels (***see top right***). If a particle branches off and deposits its energy at a much smaller Z than the rest of the event, there is a good chance that the algorithm will pick that pixel regardless of whether there is a cleaner pixel readout.

Type 2 Pathologies) Algorithm is susceptible to low statistics (***see bottom right***). Because of the dE/dx nature of the physics, there are very few resets that occur from a single clean track. 8 resets in a pixel readout usually only happens from multiple interactions. Therefore the minimum StDev of 8+ resets tends to be higher than the predicted StDev for the corresponding Z. When there are very few 8+ reset pixels, the chance that one fluctuated low (clean) decreases.





Addressing Type 1 Pathologies

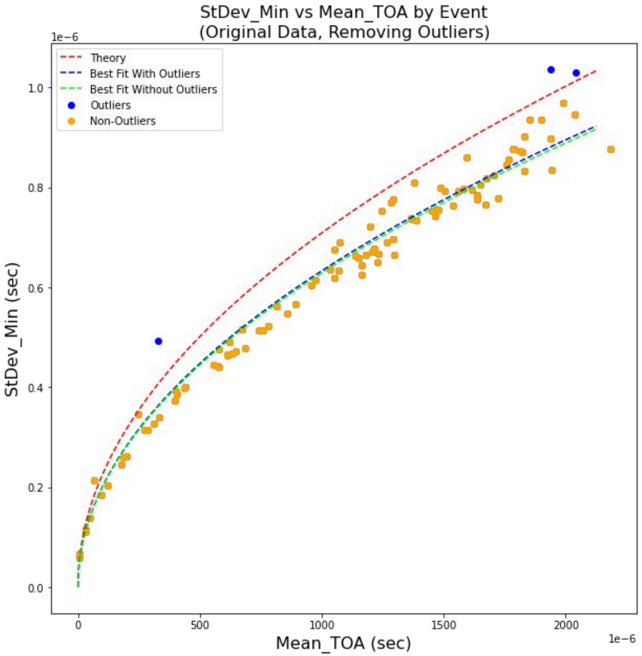
Tweak the results of StDev vs TOA functional form before using it

Remove outliers contributing more than 3 sigma to chi squared that biased the functional form and refit. Doesn't change much.

Three plots shown here:

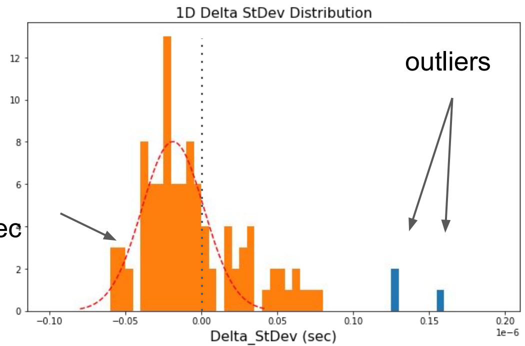
- Red is the theoretical relation
- Blue is the original fit (with all 100 points)
- Green is new fit (excluding the solid blue outliers)

StDev_Min = 1.9872e-05*sqrt(Mean_TOA)



mu = -1.873e-08 sec
sigma = 2.042e-08 sec

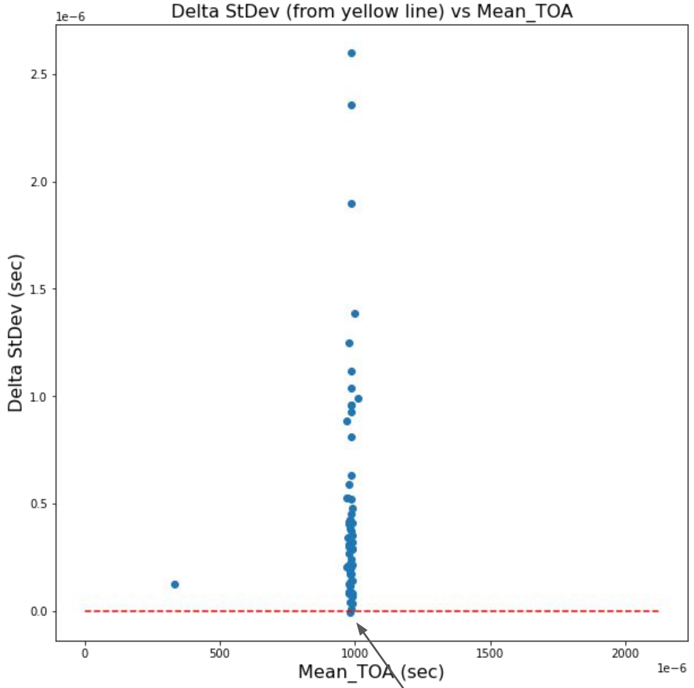
We will come back to the offset from zero later,
as this is caused by biases as function of Z



More robust method of picking the correct pixel

Since we want fluctuations low, we have to be more thoughtful about picking the pixel with the smallest value.

- Move to using the variable which is the difference between the *expected Stdev for this time*, and the observed Stdev
  - Use the “fluctuated_low” functional form for comparison
- Pick the pixel with DeltaStDev_Min
- Will fix the bad assumption that t_0=0 soon

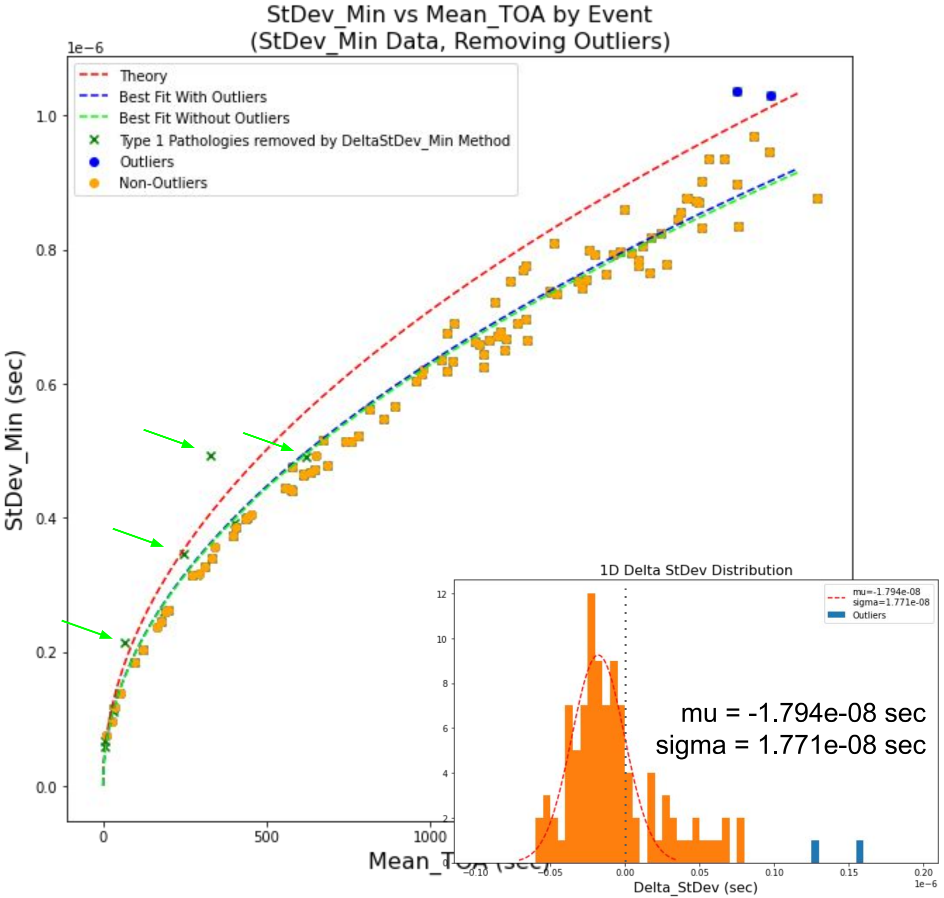


Select this pixel as it has the smallest Delta

Results after moving to the new method

Determine the *true* StDev_Min to be the pixel from each event that has the most negative (smallest) Delta_StDev compared to the predicted value given by the functional form at the given Mean_TOA.

- **Green x's** (highlighted by green arrows) show type 1 pathologies that were removed with the DeltaStDev_Min method
  - As expected, due to the nature of Type 1 Pathologies, these only occur in the smaller TOA section of the plot
- **Blue** are the two blue outlier events shown in the bottom right hand plot
- **Orange** are non-outlier points (contributing less than 3 sigma to chi squared)

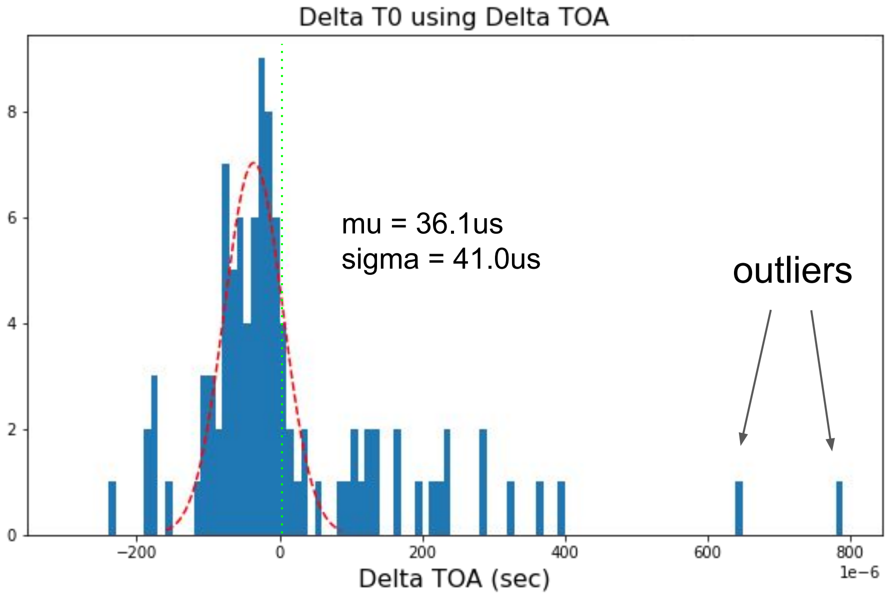


Revised fit is now:

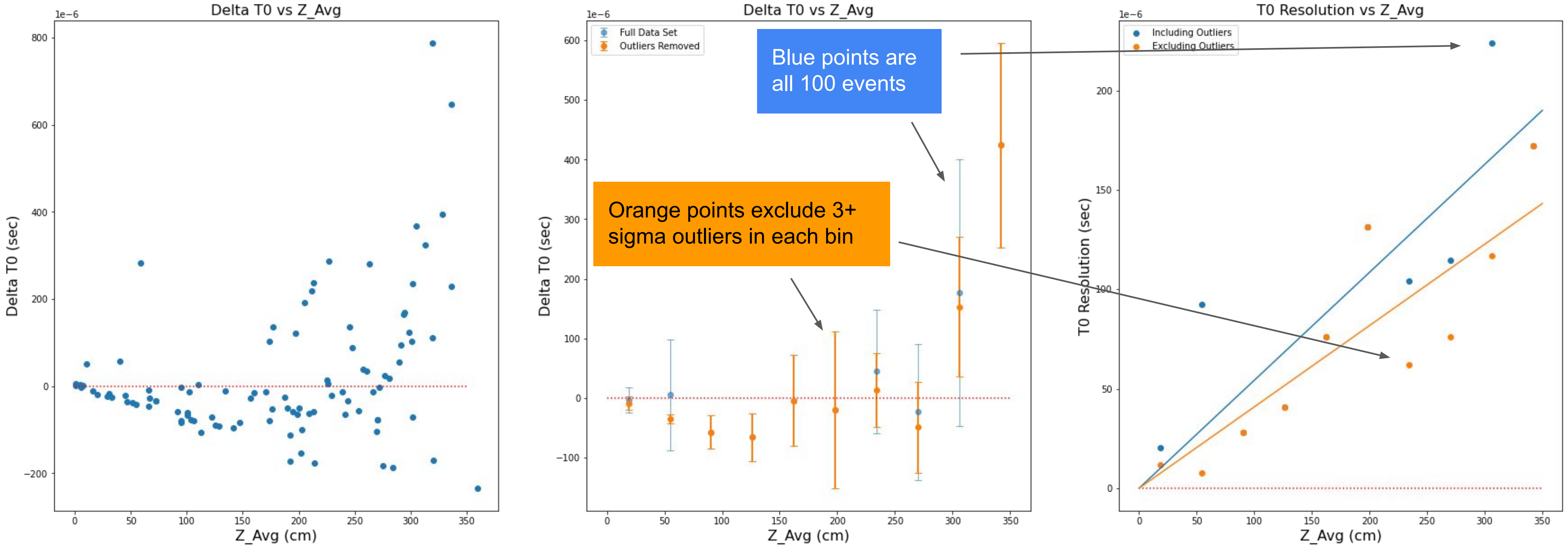
StDev_Min = 1.9852e-05*sqrt(Mean_TOA)

Calculating t_0 from TOA

- Distribution is Gaussian, but is shifted from zero as well as has tails and outliers
- Most of the variation in this plot comes from the variation in Delta_StDev
  - If we could find a way to reduce the variation in Delta_StDev, we could clean this up
  - Still biased t0 as a function of Z, but we'll deal with this after we finish the next section
- The tails come from the fact that the functional form has small biases as a function of Z, and the resolution is also rising as a function of Z.



Plotting Delta_TOA vs StDev_Min



- Still has a Z dependence
- Has a resolution of about 100-150us at Z=300cm

Correcting the t0=0 Assumption

The previous algorithm only works if we have the right x-axis, and in simulation we have the unfair advantage of t0=0 for all events

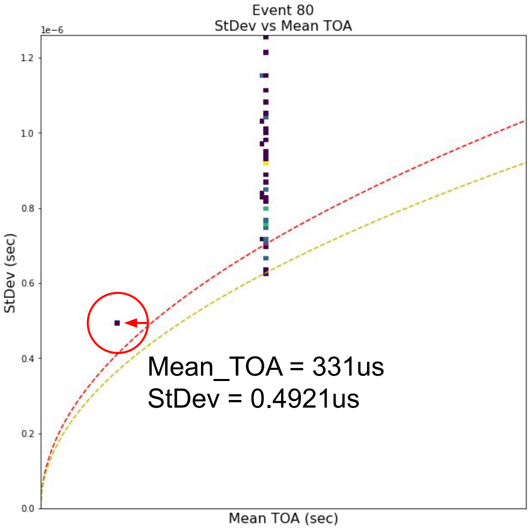
Can iterate:

- Start with the pixel with the smallest StDev
- Assume that the StDev corresponds to the yellow line fit
- Then, re-look at all StDev's and find the minimum DeltaStDev and assume that is on the yellow line. This is a direct best-estimate of the t0

See back up slides for a step by step using E80

Step-by-Step Iteration of Minimizing DeltaStDev Using E80

Step 1: Pick the pixel with smallest StDev

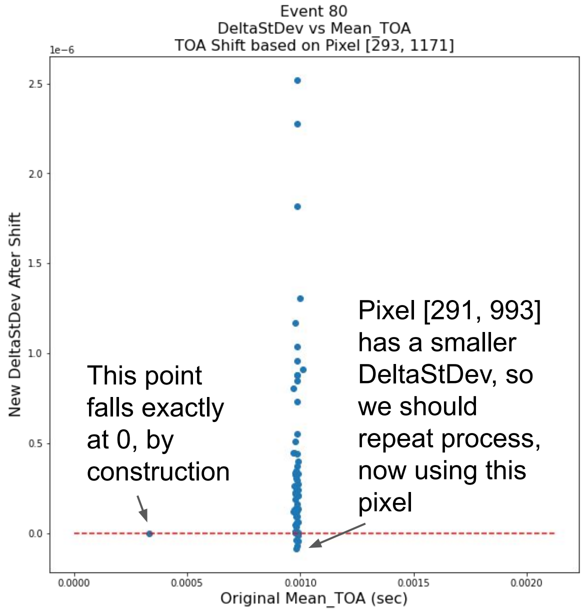


Step 2: Assume the pixel corresponds to the yellow fit line

A pixel that falls on the fit line with a StDev of 0.492us corresponds to a Mean_TOA of 614.5us

Since it actually has a Mean_TOA of 331us, we need to adjust all Mean_TOAs by adding 614.5-331 = 283.5us and calculate DeltaStDev based off their shifted positions and the yellow fit line

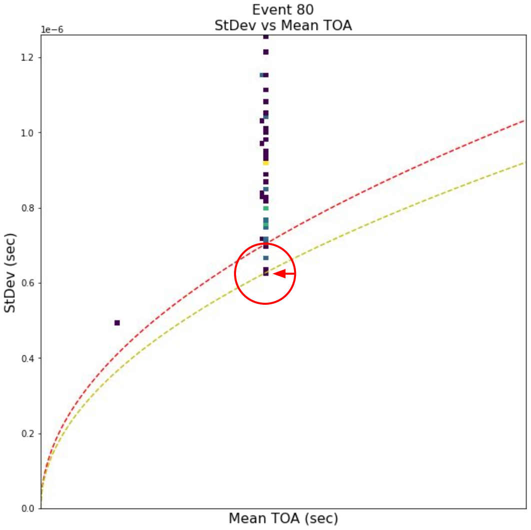
Step 3: Calculate DeltaStDev's and re-evaluate. Put new DeltaStDev_Min on yellow line



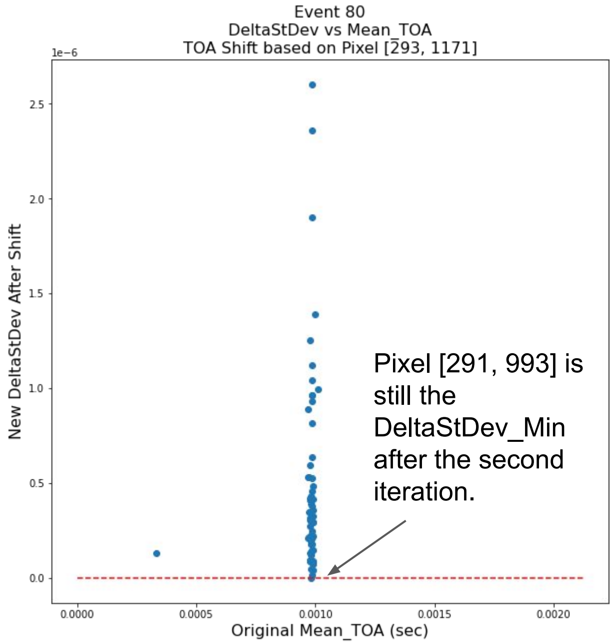
Step-by-Step Iteration of Minimizing DeltaStDev Using E80

Repeating the process, except this time, using Pixel [291, 993]:

Mean_TOA = 980us
StDev = 0.621us



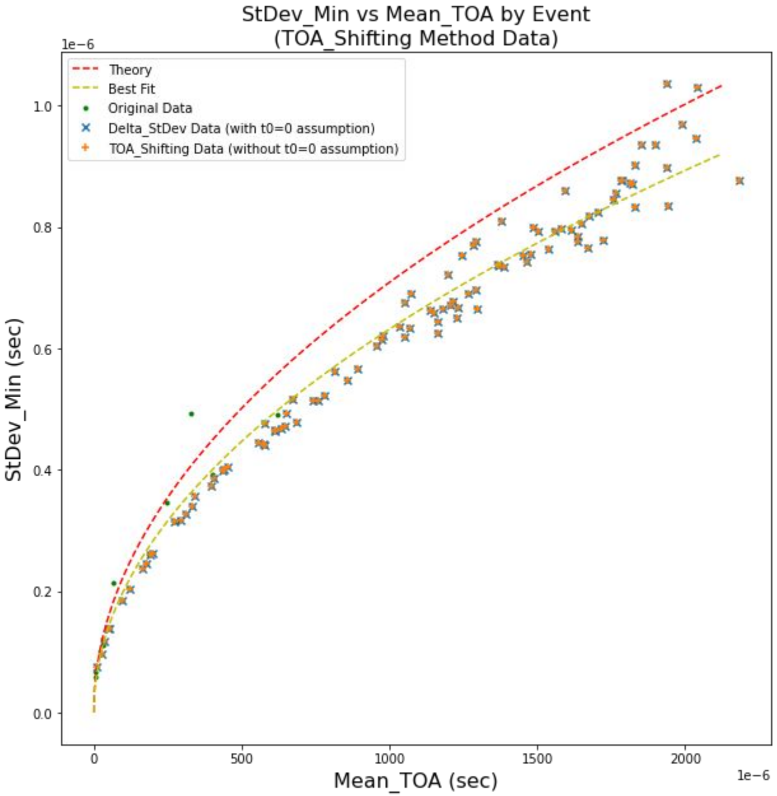
A StDev of 0.621us corresponds to a TOA of 980.1us → All Mean_TOAs will be shifted 0.1us.



Pixel [291, 993] is still the DeltaStDev_Min after the second iteration.

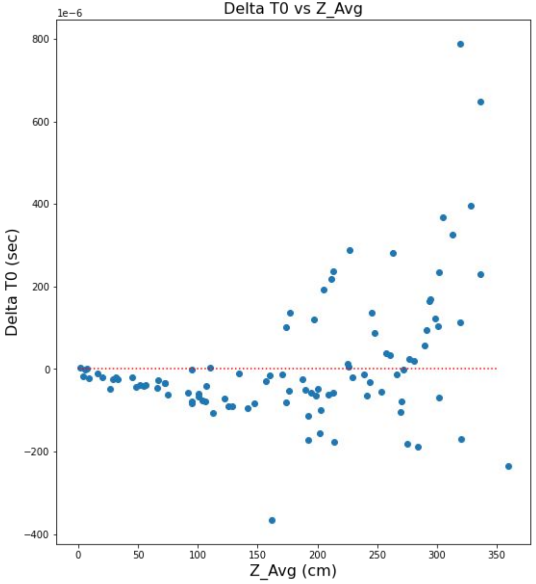
…Therefore, this confirms we should use Pixel [291, 993] in t_0 calculations, and it successfully measures t0 to 0.1us.

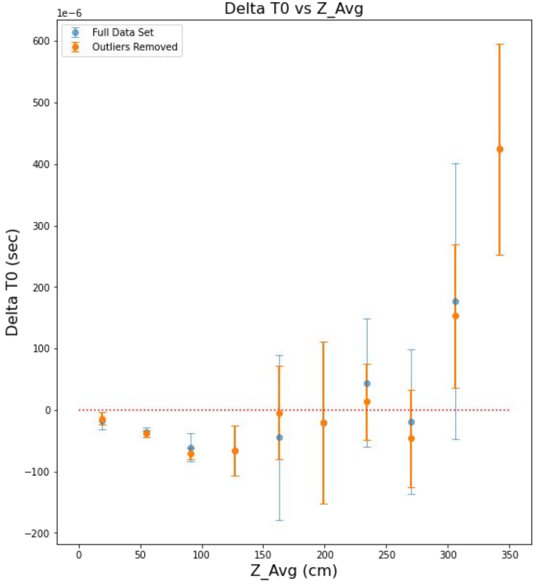
Results of TOA_Shifting (without t0=0 assumption)

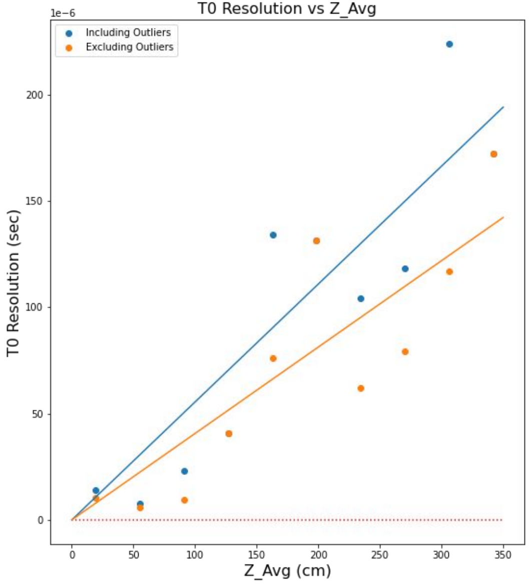


- All TOA_Shifting data matches up with the DeltaStDev data, as expected
- Now, we are getting the same correct answer, but without working on the assumption that we are starting with the correct Mean_TOA

Results of TOA_Shifting (without t0=0 assumption)







We can see that the results are almost identical, and we arrived at these results WITHOUT assuming t0 = 0. This means that we should be able to get a good measure on t0 using this method on any arbitrary t0 and TOA

Addressing Type 2 Pathologies

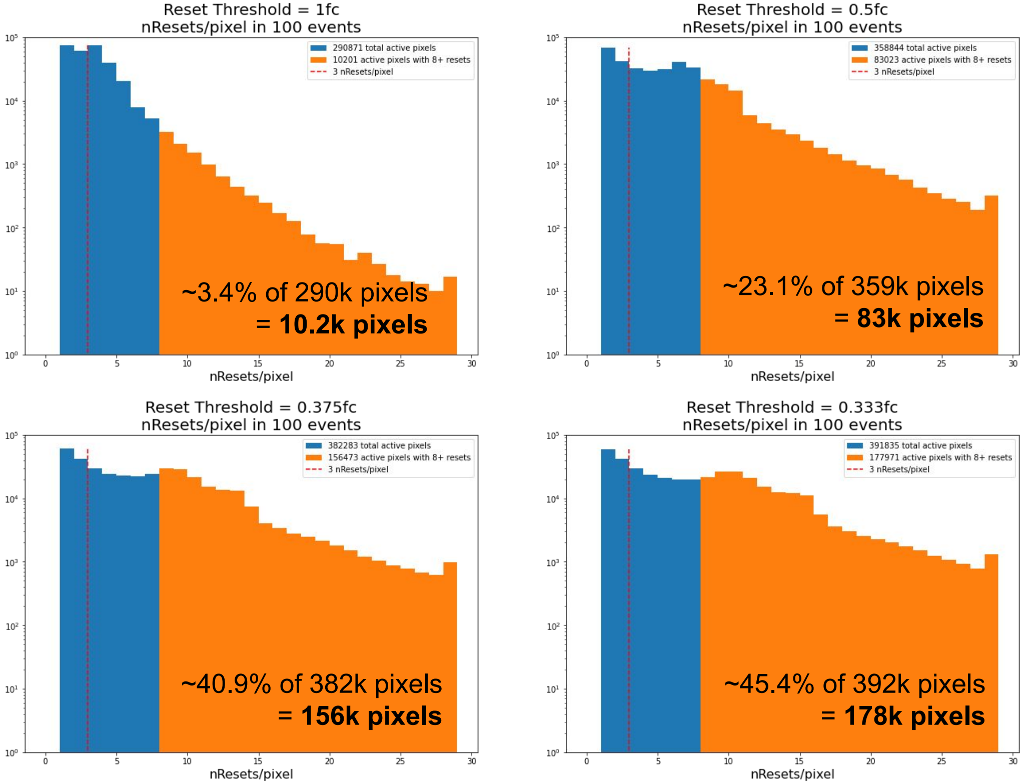
Change 2: Decreasing Reset Threshold

Problem to solve: In order to get enough resets to dependably measure the StDev, we are limited to looking at mostly multi-deposit pixel readouts, keeping us from ever getting many clean measurement

Solution Concept: Lowering the reset threshold will produce more resets per pixel. For example, if we lower the reset threshold by 50% (From 1 reset = 6250e^- to 1 reset = 3125e^-), we can get twice the resets out of a deposit and start considering more pixels in each event.

Expected Benefit: More resets per pixel raises the probability that a clean deposit has 8 resets. This should improve the resolution as well as reduce events that fluctuate to small numbers of pixels (and t0 mismeasurements)

Decreasing the reset threshold increases the number of pixels with 8+ resets per event

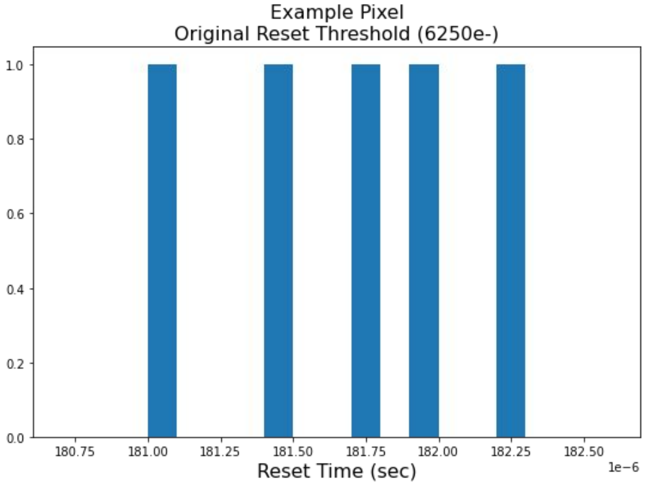


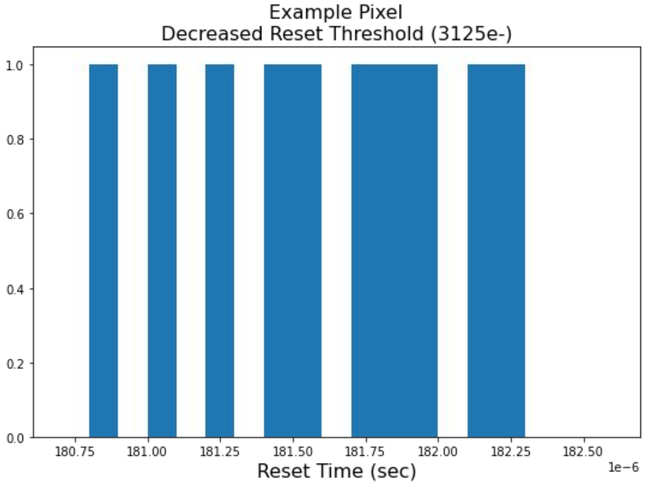
	Factor decrease of reset threshold	Factor increase of 8+ reset pixels	Percentage of new pixels added
1fc	–	–	–
0.5fc	2	8.3	25%
0.375fc	2.67	15.6	32%
0.33fc	3	17.8	35%

- With the same sample, we can drastically increase the number of 8+ reset pixels while only adding a modest number of new pixels.
- This also allows us to measure pixels that were formerly only 3+ reset pixels. This means that the chances of finding a clean pixel are much higher.

Reset times for the same pixel with two different reset thresholds

5 resets





10 resets

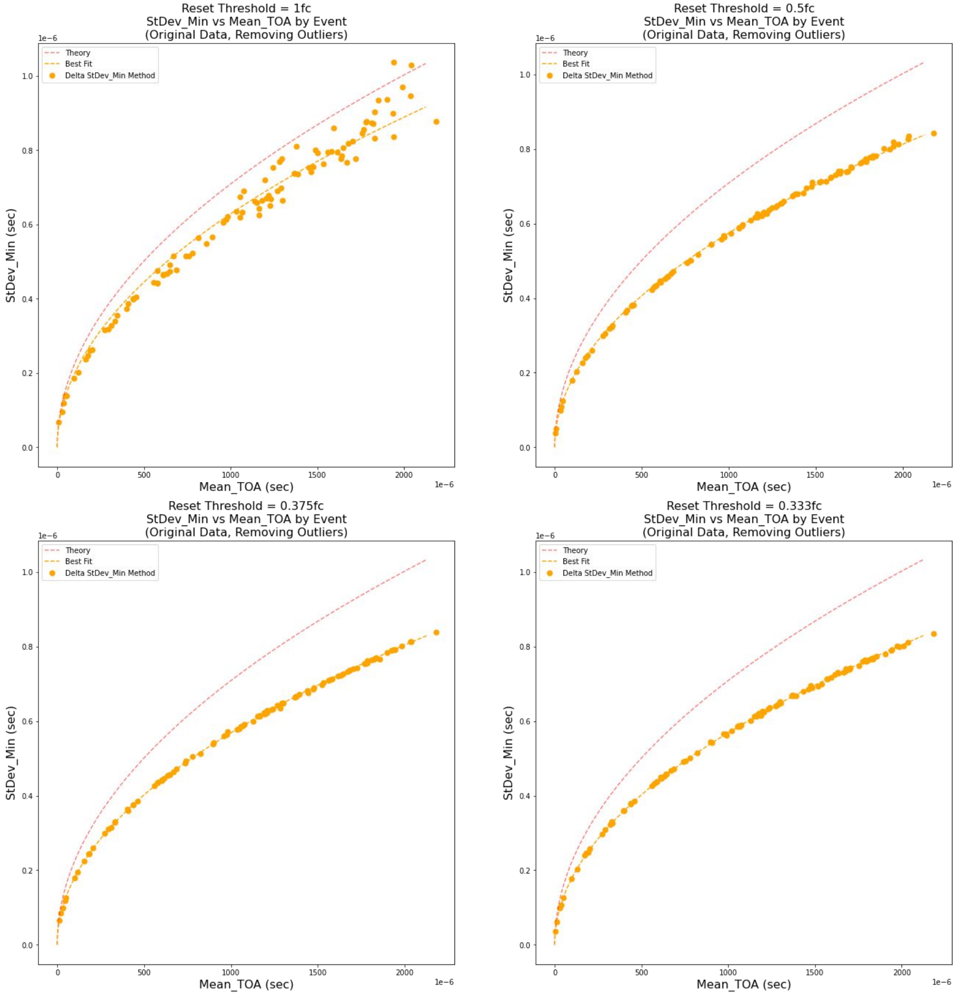
In this event, by decreasing the reset threshold by 50%, it increases the number of resets by 100%. The pixel shown above would not have been considered under the original reset threshold, but it (along with ~700 more pixels per event) can now be used to find a good t_0 measurement. Pixels that were originally able to be measured can now be measured with better resolution.

Results with New Reset Thresholds

Orange points are the points using DeltaStDev_Min method.

Can see clearly that the decreasing thresholds pull the points closer to the fit line.

We also note that the fit line moves slightly lower, as expected.



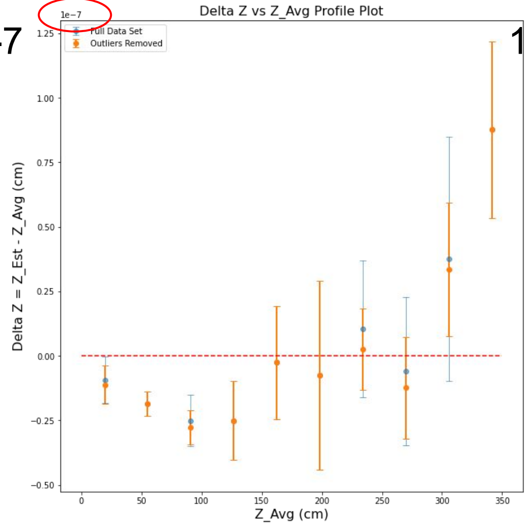
Showing Resolution and Systematic Bias

The resolution is improving as a function of reset threshold.

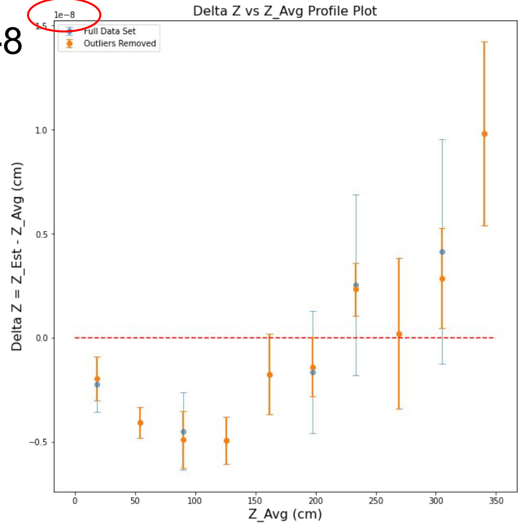
The systematic bias as a function of Z also goes away and becomes flat within statistics.

(Will compare all on one plot in the next slide)

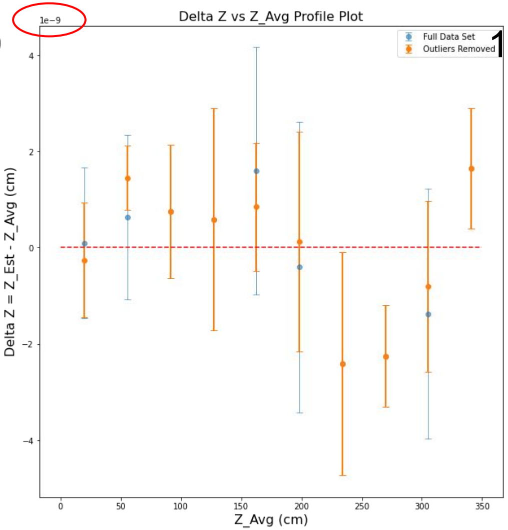
1e-7



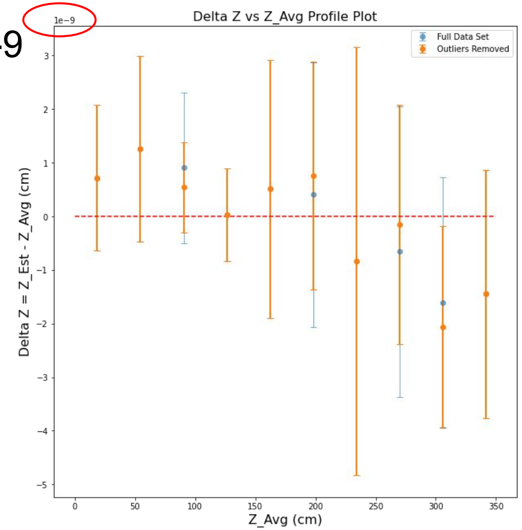
1e-8



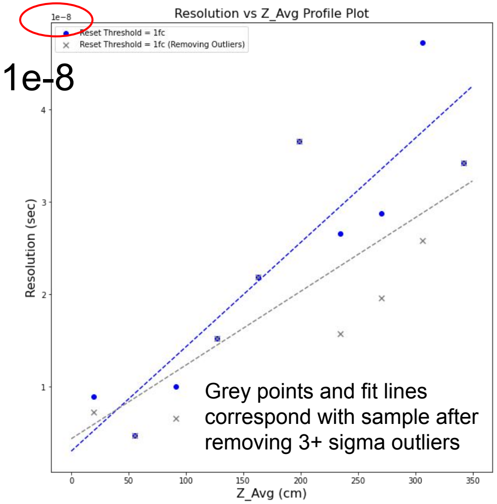
1e-9

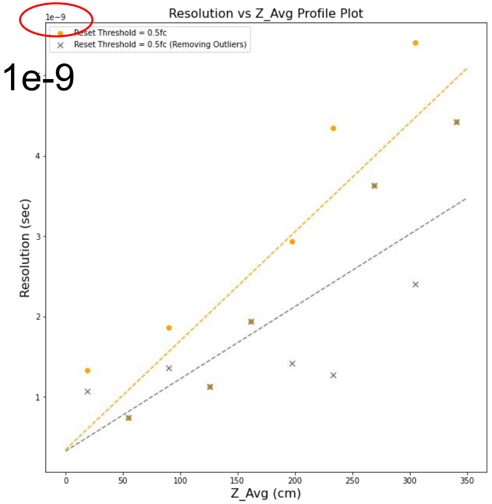


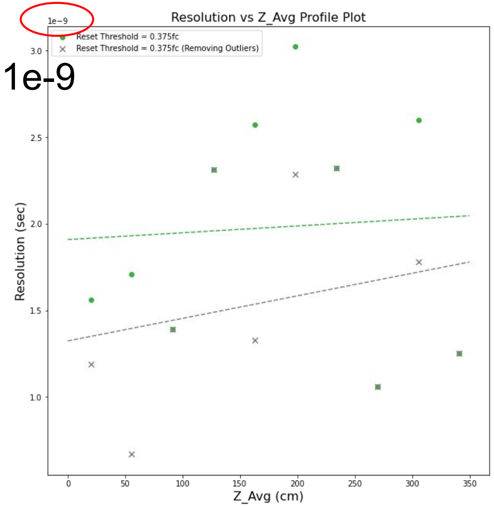
1e-9

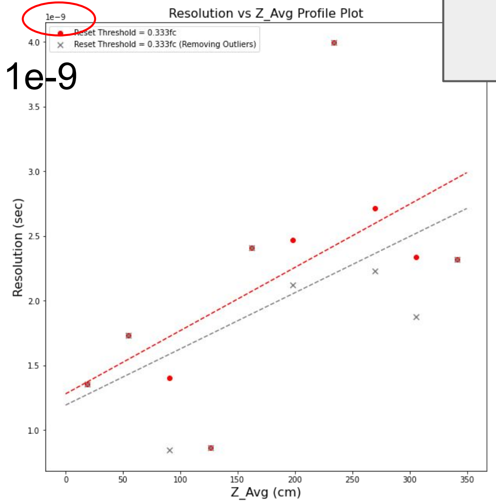


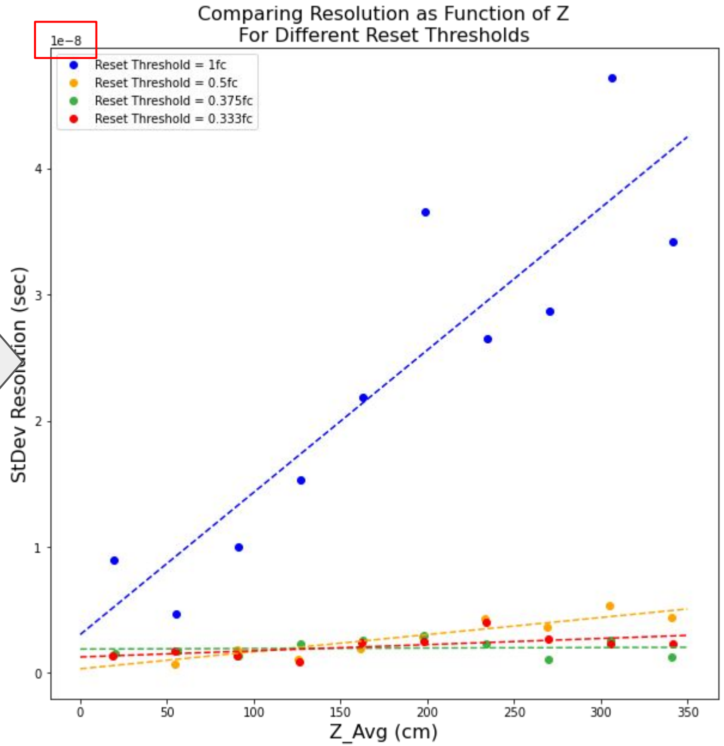
Results: RMS resolution as a function of Z



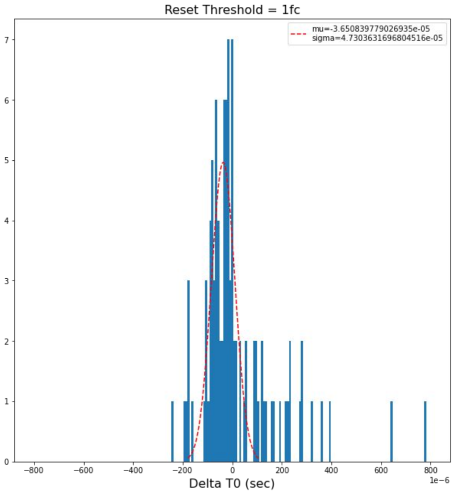




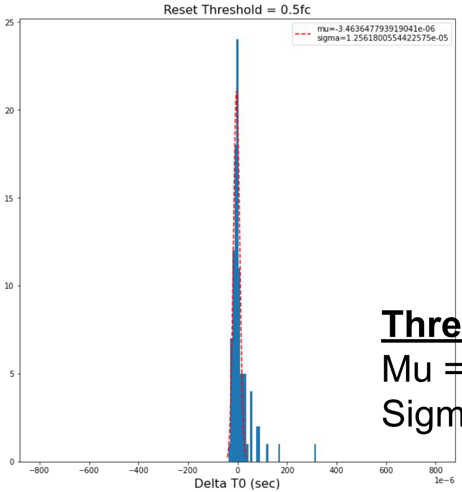




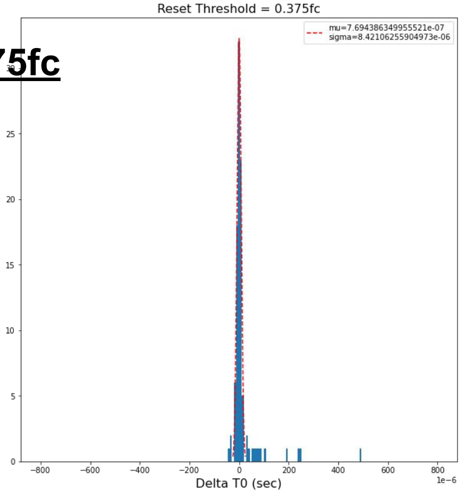
T0 resolution for different reset thresholds



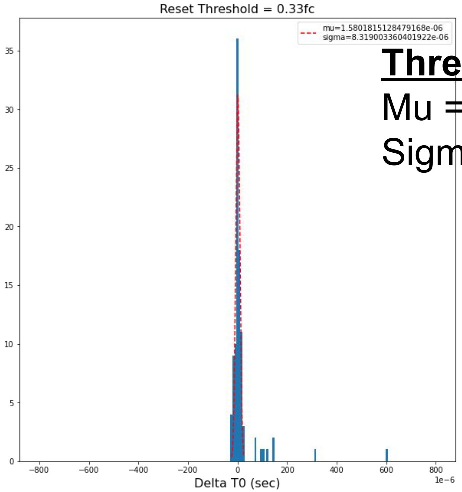
Threshold = 1fc
Mu = -3.6us
Sigma = 47.3us



Threshold = 0.5fc
Mu = -3.5us
Sigma = 12.6us

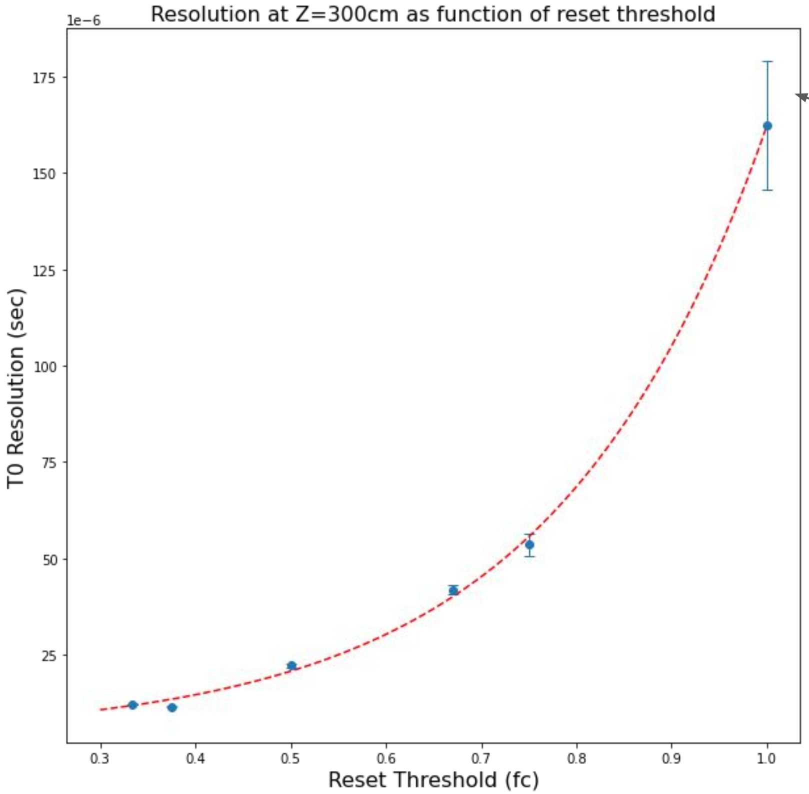


Threshold = 0.375fc
Mu = 0.8us
Sigma = 8.4us



Threshold = 0.33fc
Mu = 1.6us
Sigma = 8.3us

Comparing T0 Resolution as function of reset threshold New Sample: 100 events at Z=300cm (for better statistics)



Error bars come from the variance of the rms

Res = 1.81e-06*exp{4.47*Threshold} +3.77e-06

Note that it is 14.8us at 0.33fc, Asymptotes to ~6us (in theory)

Consistent with the full sample being ~8us as most of it has lower Z (better resolution)

Wrapping It Up: Next Steps, and Conclusions

Conclusions

- The simple StDev_Min solution was a good start, but was susceptible to 2 types of pathologies leading to misleading results
- Using the DeltaStDev_Min method remedies the type 1 pathologies, while still remaining open to clean outlier pixels, and also removes the t0=0 assumption
- Decreasing the reset threshold has big impact on the resolution of t0 calculations.
  - Decreasing the threshold greatly increases the number of pixels with nResets ≥ 8
- While there are many more checks that could be done, using both methods together allows for a dependable measurement of t0 on the order of 10us or less
- Next step is to turn this into a simple reconstruction algorithm and release