Track Finding in the STC Drift Chamber

Christopher Brown[1]
High Energy Physics Group, Imperial College London.

(Dated: 31 October 2019)

I. INTRODUCTION

The aim of this program is to find tracks in a multi wire drift chamber based on a set of drift times outputted by layers of detector wires. In each event a charged particle travels through the detector, at a number of hit points along the track the particle causes ionization which in turn causes electrons to move towards the detector wires due to an electric field in the detector. Each wire records the time taken from the start of the event to the electrons hitting the wire. The tracking algorithm takes these times, finds a straight line fit and a drift velocity that relates the electron drift time and distance between the wire coordinates and the fitted line.

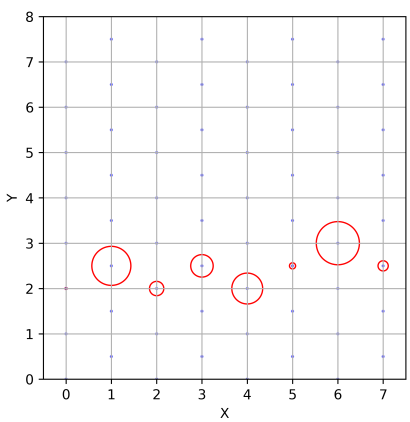

FIG. 1. Visualisation of the raw data, the red circles represent an arbitrary drift distance proportional to the drift time

II. ALGORITHM

The line fitting algorithm relies on different variations on a weighted least squares fit outlined below.

The equation of a straight line is:

$$y = mx + c \tag{1}$$

Where m is the gradient and c is the y intercept.

A weighted least squares estimate of these parameters on a set of x and y data x_i, y_i with a weight on each data point w_i is given by:

$$m = \frac{\sum w_i(x_i - \bar{x})(y_i - \bar{y})}{\sum w_i(x_i - \bar{x})^2} \tag{2}$$

$$c = \bar{y} - m\bar{x} \tag{3}$$

with:

$$\bar{x} = \frac{\sum w_i x_i}{\sum w_i} \quad \bar{y} = \frac{\sum w_i y_i}{\sum w_i} \tag{4}$$

The overall fitting algorithm is broken down into a number of steps

A. Initial Fit

The first fit has a weighting function of 1 for every data point and simply fits a line to the x and y wire coordinates. Any wire coordinates 1 cell width away from this line are disregarded. This initial cut is used to quickly remove any hits that could be erroneous, see figure 2.

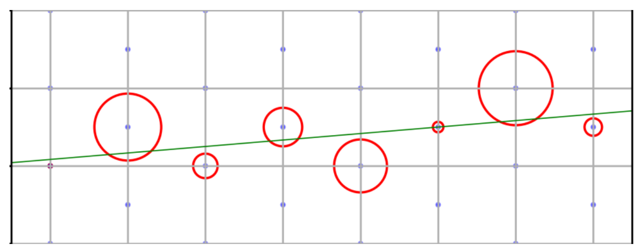

FIG. 2. First fit visualisation

B. Second Fit

The second fit is weighted with the following function:

$$w_i = \frac{1}{T_i^2 + 1} \tag{5}$$

Where T_i are the drift times for each wire i. This fit promotes those with smaller times, this ensures that the fit line occurs on the correct side of these wire points as all the wires with the smallest times will most heavily effect the fitting function. A +1 is included in the denominator to ensure any $T_i = 0$ don't cause infinities, see figure 4.

C. Velocity Determination

The first guess for the velocity is performed at this point by finding the closest distance, D, for each wire coordinate to the fit line with the following:

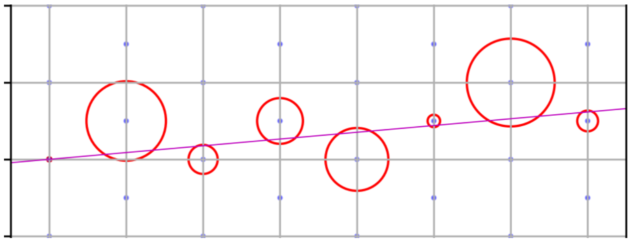

FIG. 3. Second fit visualisation

$$D = \frac{|-mx_i + y_i - c|}{m^2+1} \tag{6}$$

This distance is divided by the time for each drift wire and averaged to give the drift velocity.

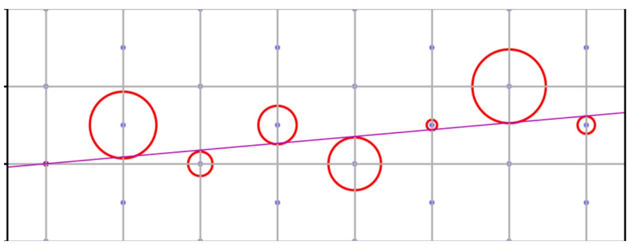

FIG. 4. Drift Velocity calculation, the red circles have now been updated to better represent the new drift distances

Using this velocity a new set of hit positions are found. These hit positions are the drift distance away from the wire coordinates and the closest point to the fit line. These hit positions represent the physical points at which the electrons began moving to the hit wires and so a best fit line to these points is the track of the charged particle through the detector.

D. Final cuts

Another weighted fit is performed on the new hit positions but with a weighting $\propto T_i^2$, this fit favours the more accurate longer times. More accurate as the relative error from the absolute timing error is smaller. After this fit any hit points with a distance > 0.3cm away from the fit line are removed.

This fit, while not strictly necessary, reduces the final time as it removes any far away points that will affect the next set of iterated fits.

E. Iterative Fitting

After this point the algorithm alternates finding the best drift velocity and finding the best fit line to the data.

The best fit is a non-weighted least squares fitting. At this point the error on the best fit line is found using the error in the velocity, strictly the error on the best fit line should also take into account the timing error and not just the velocity error but as this timing error has already been included in the velocity error. Also this error calculation assumes the error is fully in the y direction and not a mixture between x and y as the timing error should be however there is no analytical way of solving a linear fit with errors in both x and y and with such shallow gradients the difference was deemed small enough to not include.

The fit and velocity calculation are repeated iteratively with each new fit being based on the updated hit positions which change with the new velocity and the velocity changing with each new fit. This process continues unless there are fewer than 4 hit points left, the iteration has passed 10 iterations or the velocity has not changed between iterations by more than a given tolerance. In the last case the fit is deemed as good and the fitting process stops, otherwise the fit is bad and discarded.

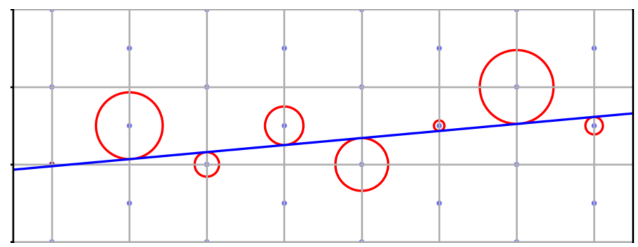

FIG. 5. Final fit, with final drift distances

III. RESULTS

The fitting algorithm described above was performed on the single track and 1 million track data files.

TABLE I. Summary of final results. Many tracks results are quoted as mean and sigma from fitted Gaussians to weighted histograms based on individual event errors of the full dataset.

One Track Drift Velocity	250.45 ± 0.17 μm/ns
One Track Track Angle	5.2082 ± 0.0057 degrees
Many Tracks Drift Velocity Mean	52.92 μm/ns
Many Track Drift Velocity Sigma	0.14 μm/ns
Many Tracks Track Angle Mean	8.964 degrees
Many Tracks Track Angle Sigma	2.302 degrees

Figures 6 and 7 clearly show good fits to the weighted histogrammed data for both drift velocity, their parameters are shown in table I. The weighting was performed based on the error of each individual event, smaller errors were weighted larger. Overall the algorithm performed effectively producing a fits quickly with a clear convergence across all events to a single drift velocity. The program reads in the binary, calculates the tracks and outputs to a datafile in around 10 seconds.

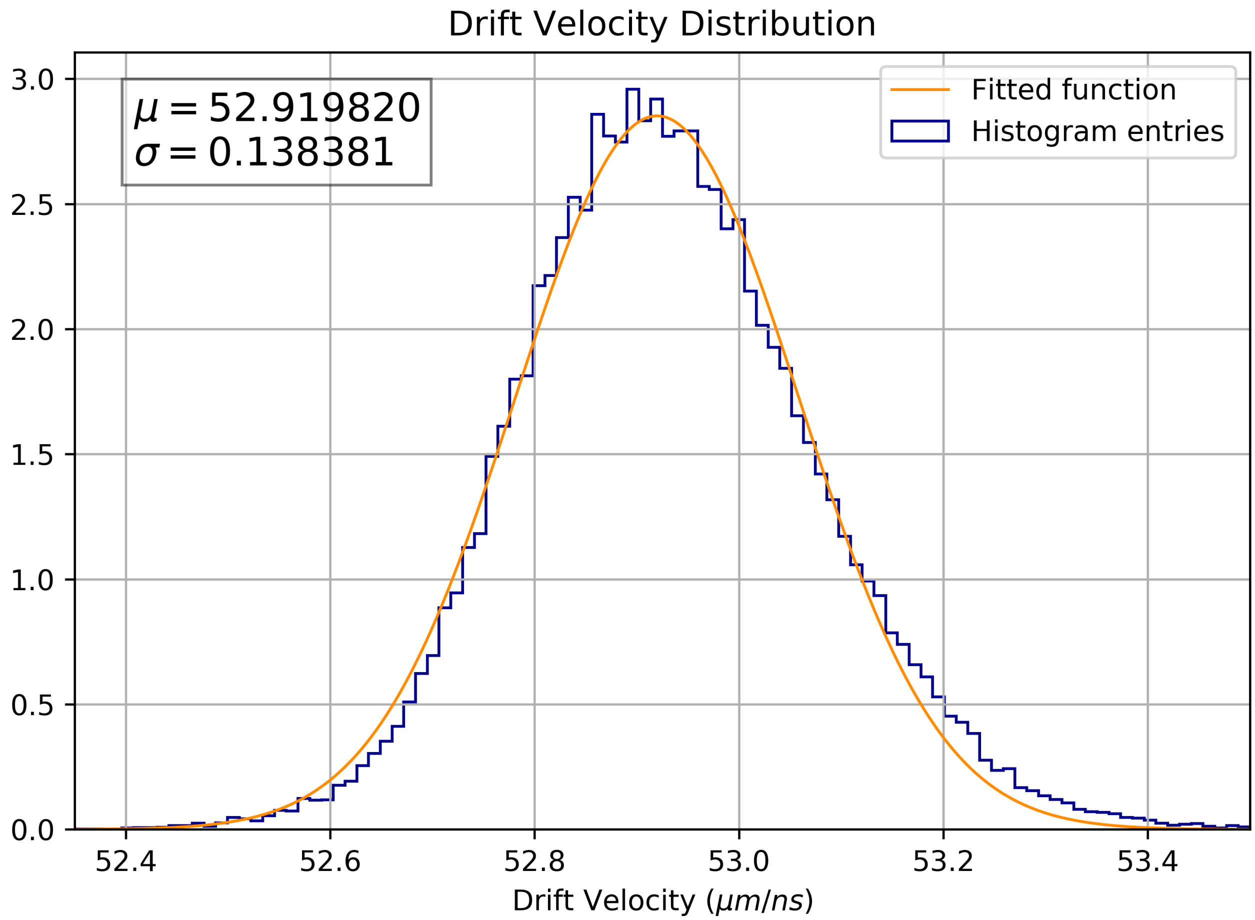

FIG. 6. Weighted histogram of the final velocity for 1,000,000 events, 100 bins, with the fitted line shown in orange. The mean and standard deviation are also shown.

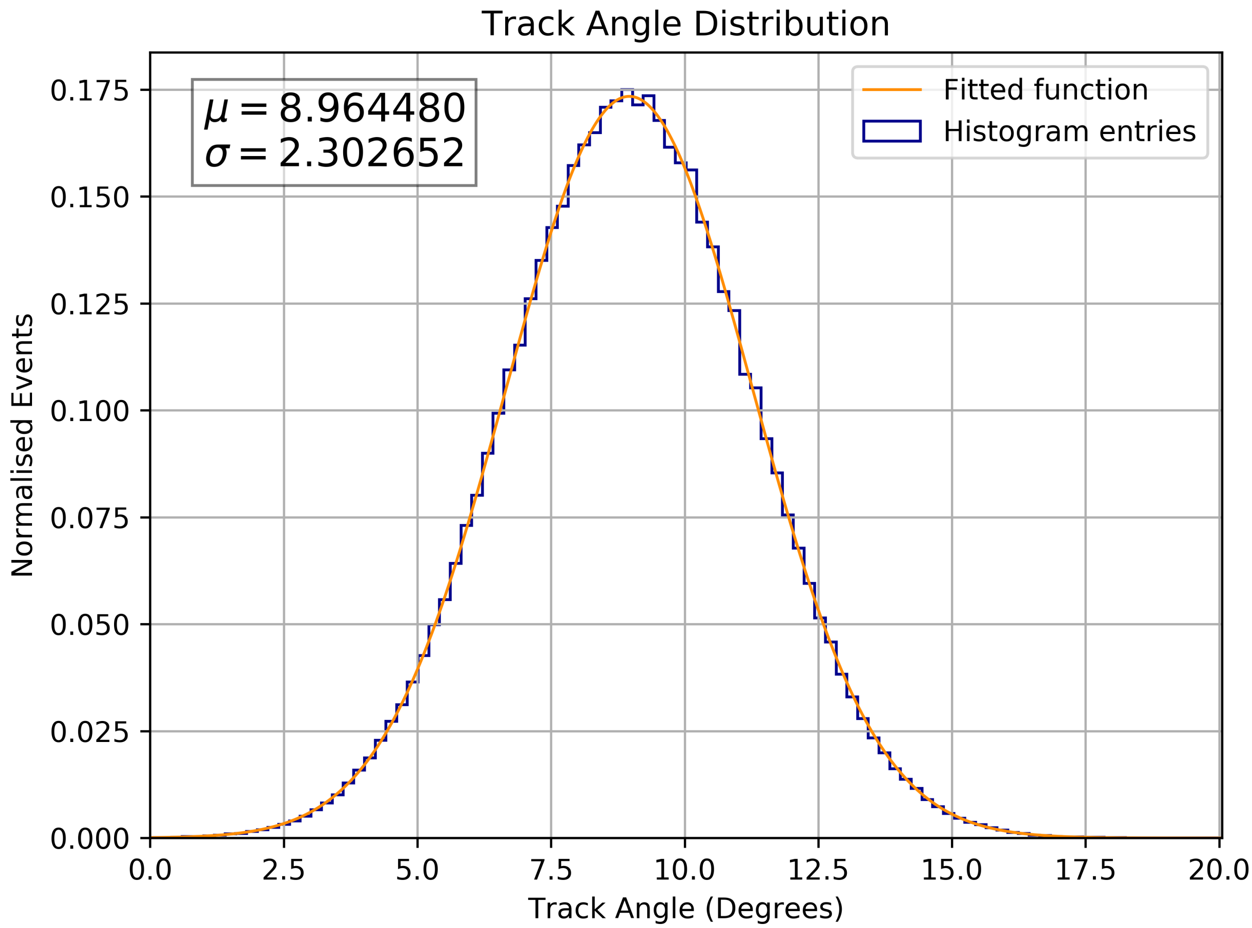

FIG. 7. Weighted histogram of the final angular distribution for 1,000,000 events, 100 bins, with the fitted line shown in orange. The mean and standard deviation are also shown.