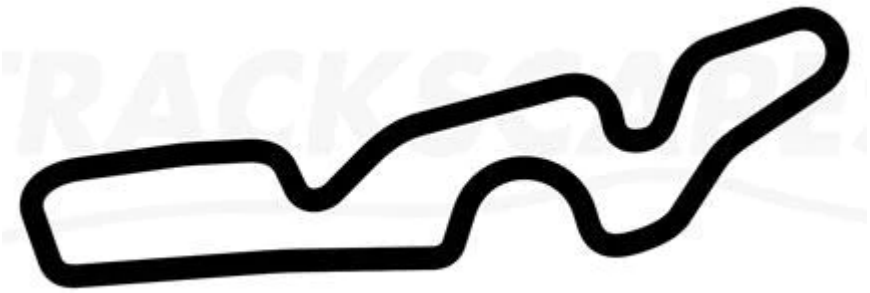
Lydd GoKart Simulation documentation



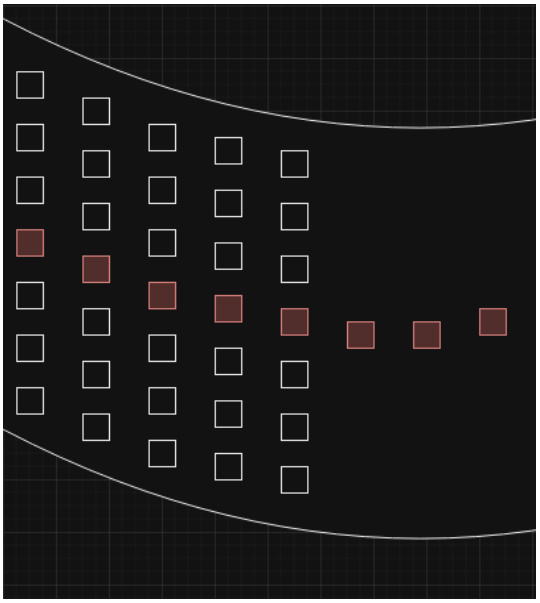
Step 1 - Process Image

The first step was to read the Lydd track png image into some data format to then process. This is achieved by reading in the pixel brightness data and filtering out all the ‘light’ pixels. Also due to the convention of images starting at the top left this data is also transformed to have the origin at the bottom left to match a cartesian coordinate frame.

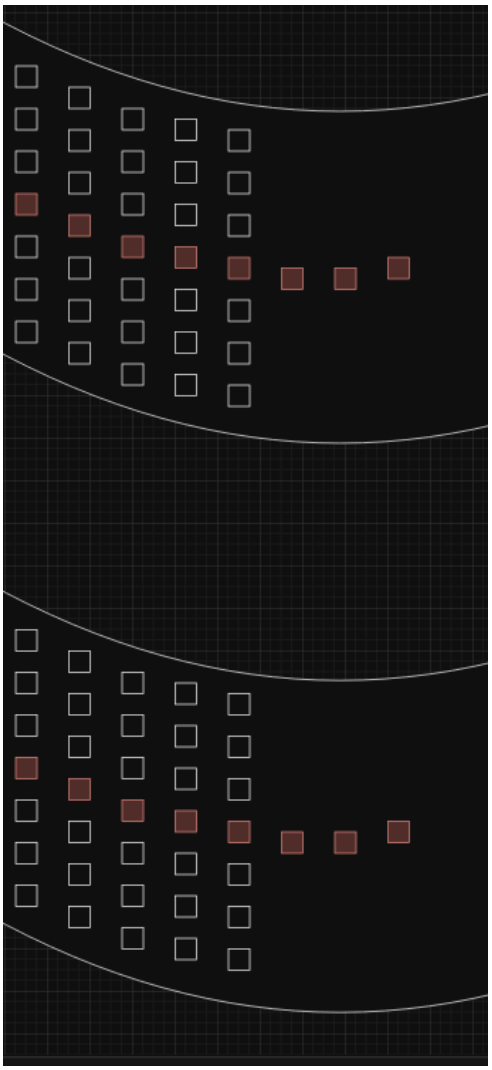
The imageProcess function takes the .png file and returns x_raw, y_raw and the RGB image. This has split the pixel data into slices where each X value has many Y value. The X array can be thought of as the X axis for the image. The index of the Y array corresponds to the X axis of the image. (See image in next section)

Step 2 - Calculate ‘race-line’

The method implemented here takes the midpoint of the Y values. This can be seen below where each square is a pixel on the track and in red is the midpoint that is being dubbed the race line.



However the is a loop so there will be both an upper and a lower track. This results in two distinct sections in the Y values for each X value. The sections can be distinguished by checking where the Y values are no longer consecutive then writing the upper and lower midpoints to separate values for upper and lower track.



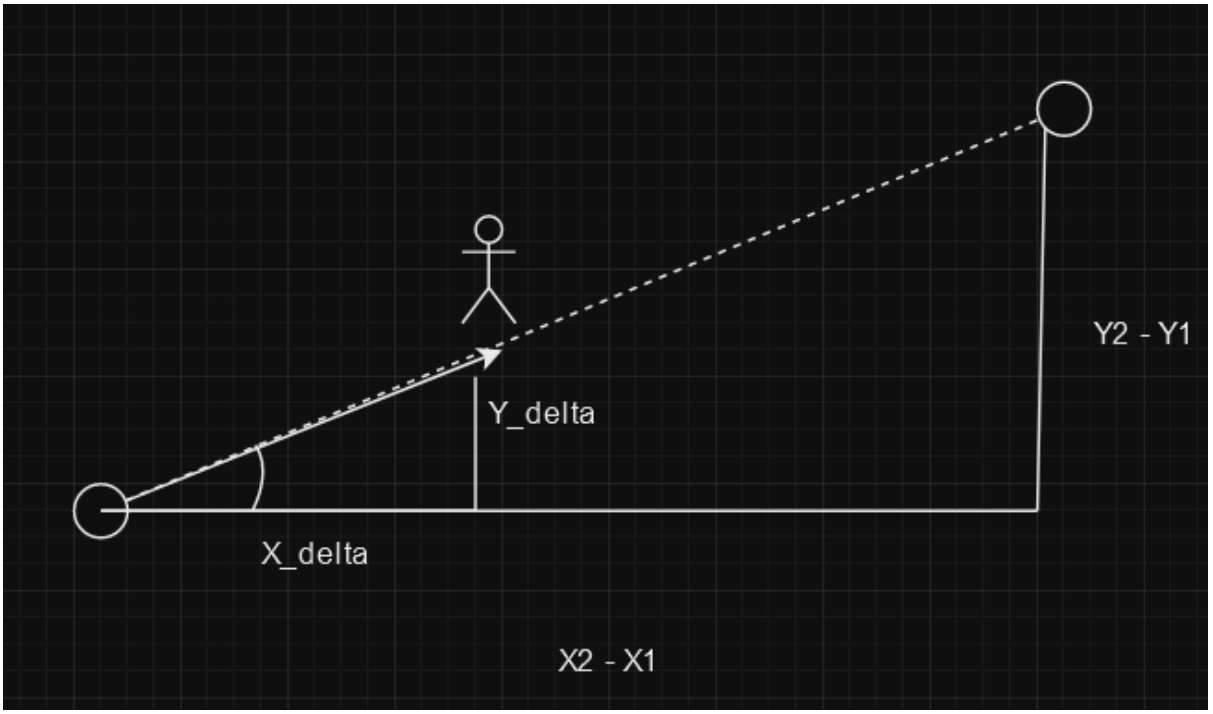
This method breaks down at the far left and far right side of the track as only one consecutive section of Y values will be present but an upper and lower race-line point is still required. Here a method of taking the upper and lower quartile is used but some ad hoc tweaking was performed to create a neater looking race-line. The data is then all appended together.

Next, the data is filtered to make the simulation faster. There is no loss of accuracy here due to the interpolation that happens later.

Step 3 - Simulation

The lap times are read in from the .csv file. Using the lap time and average speed for that lap is calculated and this is then used to determine the drivers distance from the start line at each time step t. Now using the race-line the cumulative distance between each point is calculated. The driver's distance will fall between two of these points, assuming a linear interpolation the exact X and Y coordinates can be computed.

The ‘extra’ driver distance is assumed to be in the direction of the next point. With this assumption a triangle can now be drawn.



The larger triangle is used to find the angle. The drive distance up the slope and the angle is then used to calculate the ‘extra’ X and Y values. These then are added onto X1 and Y1 to find the drivers coordinates.