

CURVE ANALYSIS USING 1D HWT

To Localize the pad in the given binary image (black and white image containing grass pixels), I used the approach as mentioned in the assignment.

Row_bins = 10 rows

Column_bins = 30 columns

Number of iterations of 1D Ordered Fast HWT = 2

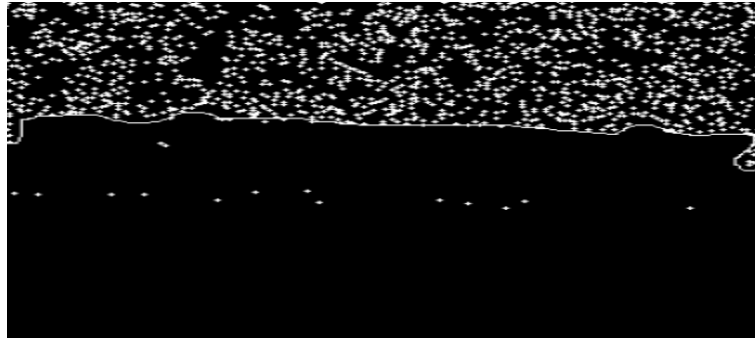


Algorithm:

a) Vertical projection of the image to find the position 'columns' of landing page

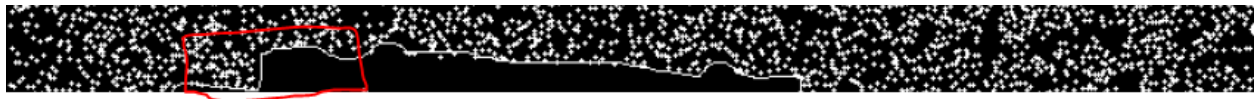
- 1) The above image is divided 'vertically' into bins of 30 columns each. Total number of columns in the image = 720. Number of bins = $720/30 = 24$
- 2) For each vertical bin containing 30 columns, I found the total number of white pixels. A white pixel is any (row, col) with value > 200 .
- 3) Once I got the total values of white pixels for every column bin, I stored these 24 values in an array **double[] array_vertical** and chop off the last 8 values to give me final 16 values which together compute a signal. [16 bins of 30 col each = 480 columns in image]
- 4) I passed this signal containing 16 values, to 2 scales of 1D Ordered Fast HWT and left the last 8 coefficient values from 1st iteration.

- 5) From the 4 coefficient values of 2nd iteration I found the first change in signal giving me the column values where the count of white pixel changes and the remain constant(not exactly) to give me the flat part in the image. (figure below)



b) Horizontal projection of the image to find the position 'columns' of landing page

- 1) The original(binanzed) image is divided 'horizontally' into bins of 10 rows each. Total number of rows in the image = 480. Number of bins = $480/10 = 48$
- 2) For each horizontal bin containing 10 rows, I found the total number of white pixels. A white pixel is any (row, col) with value > 200.
- 3) Once I got the total values of white pixels for every row bin, I stored these 48 values in an array **double[] array_horizontal** and chop off the last 16 values to give me final 32 values which together compute a signal. [32 bins of 10 rows each = 320 rows in image]
- 4) I passed this signal containing 32 values, to 2 scales of 1D Ordered Fast HWT and left the last 16 coefficient values from 1st iteration.
- 5) From the 8 coefficient values of 2nd iteration I found the maximum change in signal giving me the row values where the count of white pixel changes and the remain constant(not exactly) and again the next change which gives me the first ladder.



Finally, I used `drawLineInMat()` method to plot the rectangle in the original image and saved it in the output directory. **I could get the landing pad location for 14 images out of 23.**

Ashwani Chahal

