COMP27112 lab 4

1. Canny has two thresholds that control the edge thresholding process. What is their purpose?

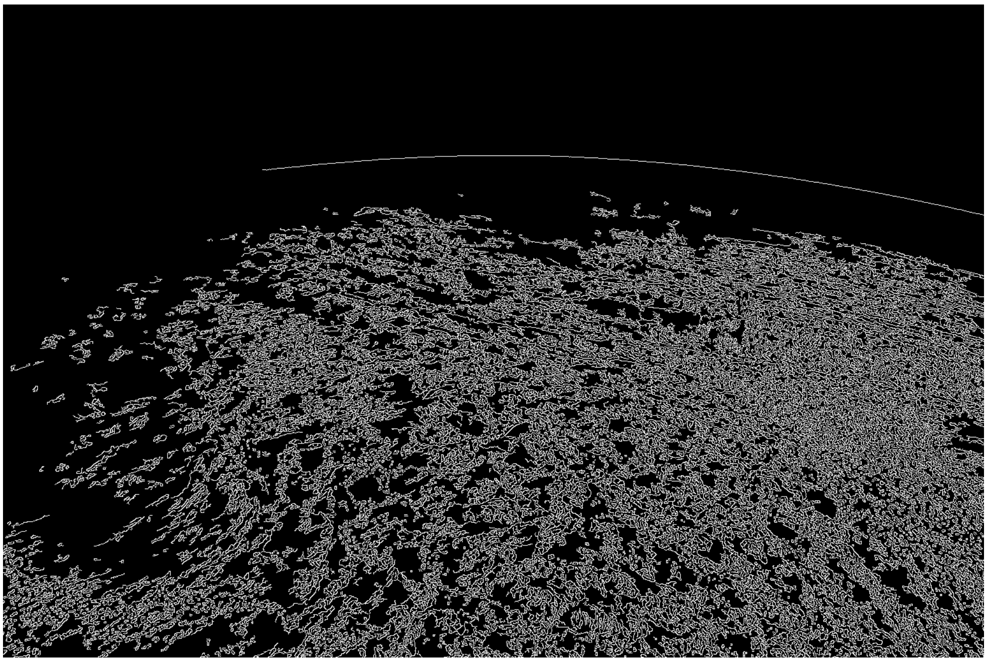
For the edge thresholding process, there is `threshold1` and `threshold2`. `threshold1` is also known as the lower threshold, and any edges with a gradient magnitude lower than this threshold will be discarded in general, and no longer considered an edge. `threshold2` aka the higher threshold is used to identify weak and strong edges as such:

- gradient magnitude > higher treshold → strong edge
- lower threshold < gradient magnitude < higher threshold → weak edge
  finally, any pixels which are part of the strong edges are final edges - additionally, any neighbouring pixels which belong to a weak edge will also be included in the final edge map

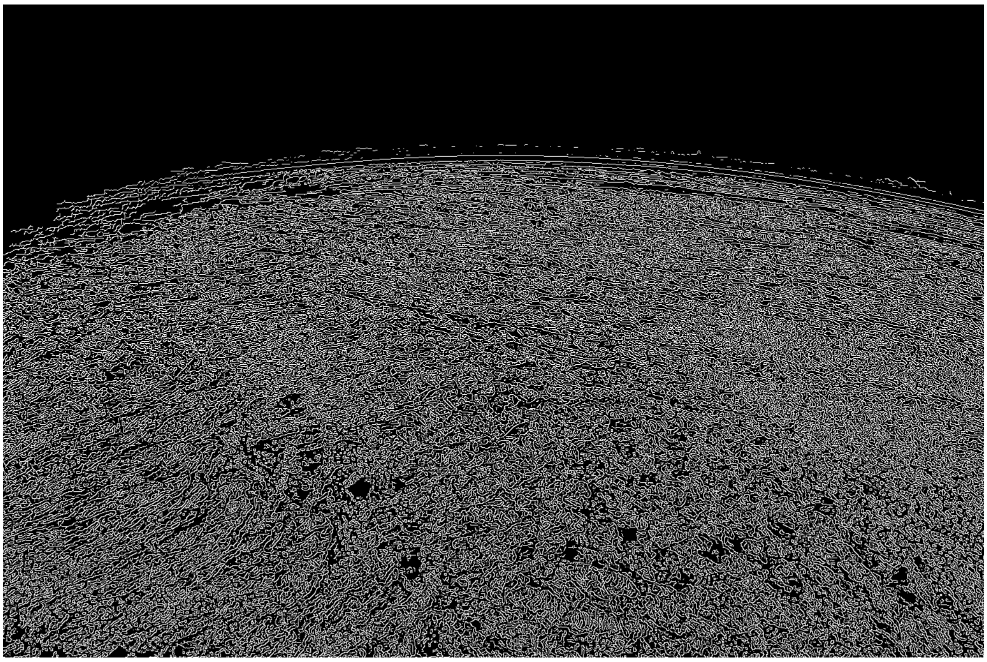
2. What is the purpose of the aperture parameter? What is the result of changing it from 3 to 5, 7, 9 or greater?

`aperture` defines the size of the Sobel kernel, which passes over the original image to calculate the gradient magnitudes - for instance, `aperture` being set to 3 results in a 3×3 Sobel kernel being used.
As shown in an example below, larger filters i.e. a larger aperture value results in more edges/noise being picked up by the canny filter, which may not always be favourable in the context of horizon detection

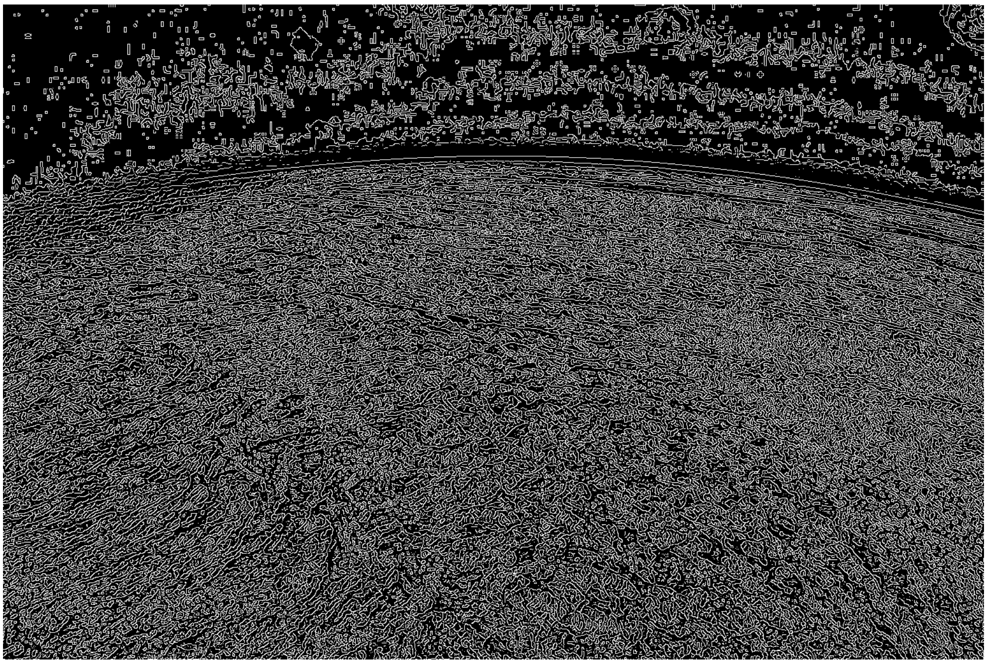
`aperture = 3`



`aperture = 5`



`aperture = 7`

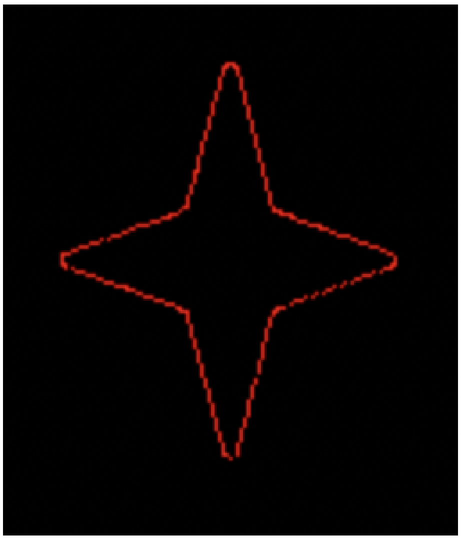


openCV does not permit a sobel filter size above 7, so effects on the canny filter above 7 cannot be displayed, but one would assume this pattern would continue

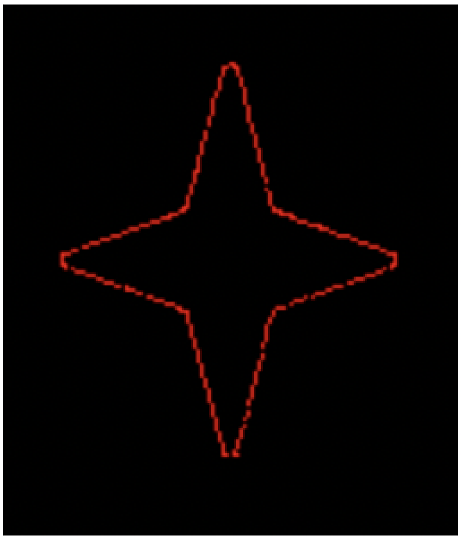
3. The Hough transform has two parameters that specify the resolution of the accumulator. Their default values are 1 and π/180. What is the effect of increasing the first and reducing the second?

The first parameter with a default of 1 is `rho` (ρ), and the second parameter with a default of $\pi/180$ is `theta` (θ). `rho` defines the distance resolution of the accumulator array, and `theta` defines the angle resolution of the accumulator array. The accumulator array is used in the hough transformation, in order to detect the presence of any line (i.e. y = mx + b), and its dimensions are equal to `rho` and `theta`. Therefore, by increasing the first parameter `rho`, the accumulator array's distance resolution increases, and the hough transformation becomes more sensitive to detecting lines within the image - this means its

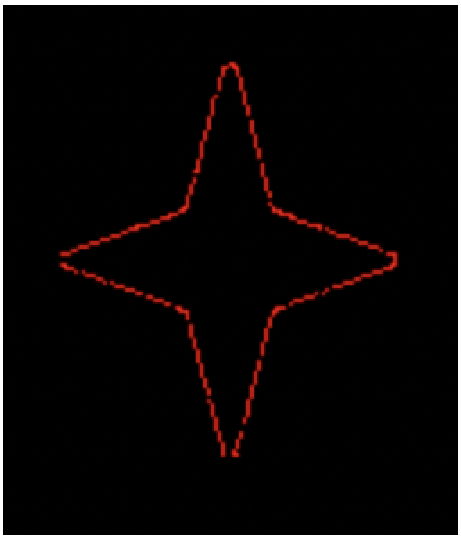
possible that more lines are drawn in its result.
By decreasing the second parameter `theta`, the accumulator array's angle resolution actually increases, as it is possible to detect more lines at precise angles - this means more lines may also be detected
When increasing rho and decreasing theta simultaneously, the lines appear to become slightly more visually segmented, for example:
rho = 1, theta = $\pi/180$



*rho = 2, theta = $0.5*\pi/180$



rho = 5, theta = $0.1*\pi/180$



Note that by increasing rho and decreasing theta, the time taken for the hough transformation may also increase, as more calculations must be carried out with higher accumulator resolutions

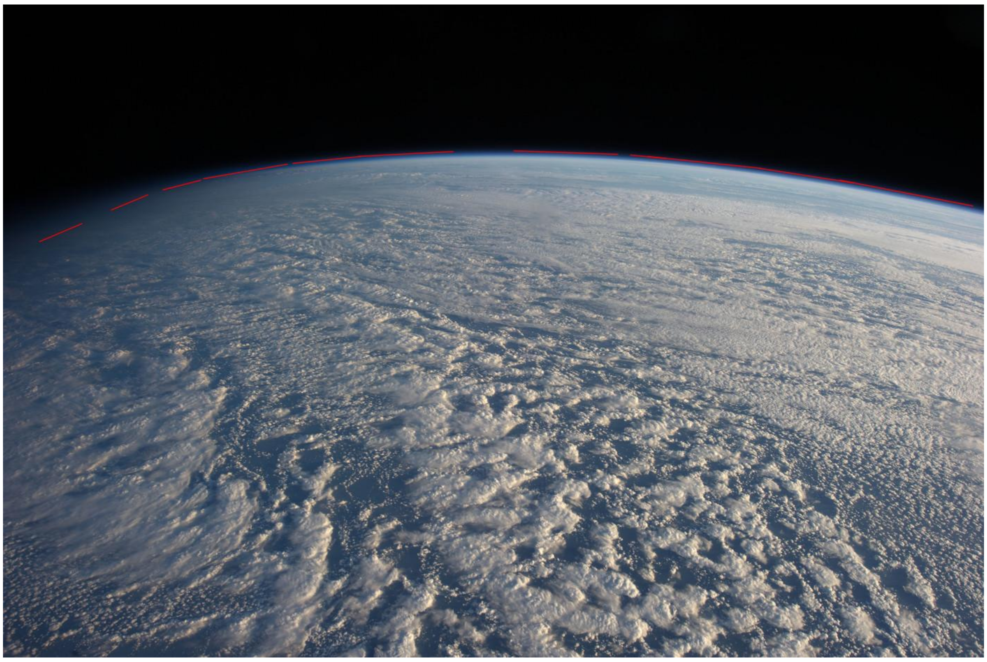
---

4. The Hough transform has a pair of parameters that determine the minimum length of a line that can be accepted, and the maximum gap between two segments if they are to be considered part of the same line. What is the effect of changing these values?

The `minLineLength` parameter determines the minimum length of a line that can be accepted - increasing this will result shorter lines will be discarded, while decreasing this value will allow more short lines to be detected.
The `maxLineGap` parameter determines the maximum gap the Hough transform will allow between two line segments, and still bridge them into a singular line. By increasing this value, longer lines may be detected but additionally multiple lines may wrongly be merged into a single line if the max gap is very large. By decreasing `maxLineGap`, detected lines will likely be shorter, and fragmented

---

5. How close are the computed horizons to where you think the horizon should be? What might cause any discrepancy?
The computed horizons are very close, if not on top of where the horizons 'should' be, if judging by eye - in some cases (particularly horizon1.jpg), the computed horizon line will be slightly above the earth, or towards the left side of horizon1.jpg, fewer/shorter lines are picked up within the hough transformation:



As can be seen in the image, the separation between horizon and space (black) is clear on the right side, but becomes less clear and like more of a gradient on the left side - across the entire horizon line, there is a small gradient separation in general, which could be one of the contributors to the difference between computed line and actual line