OpenCV approach

https://towardsdatascience.com/tutorial-build-a-lane-detector-679fd8953132

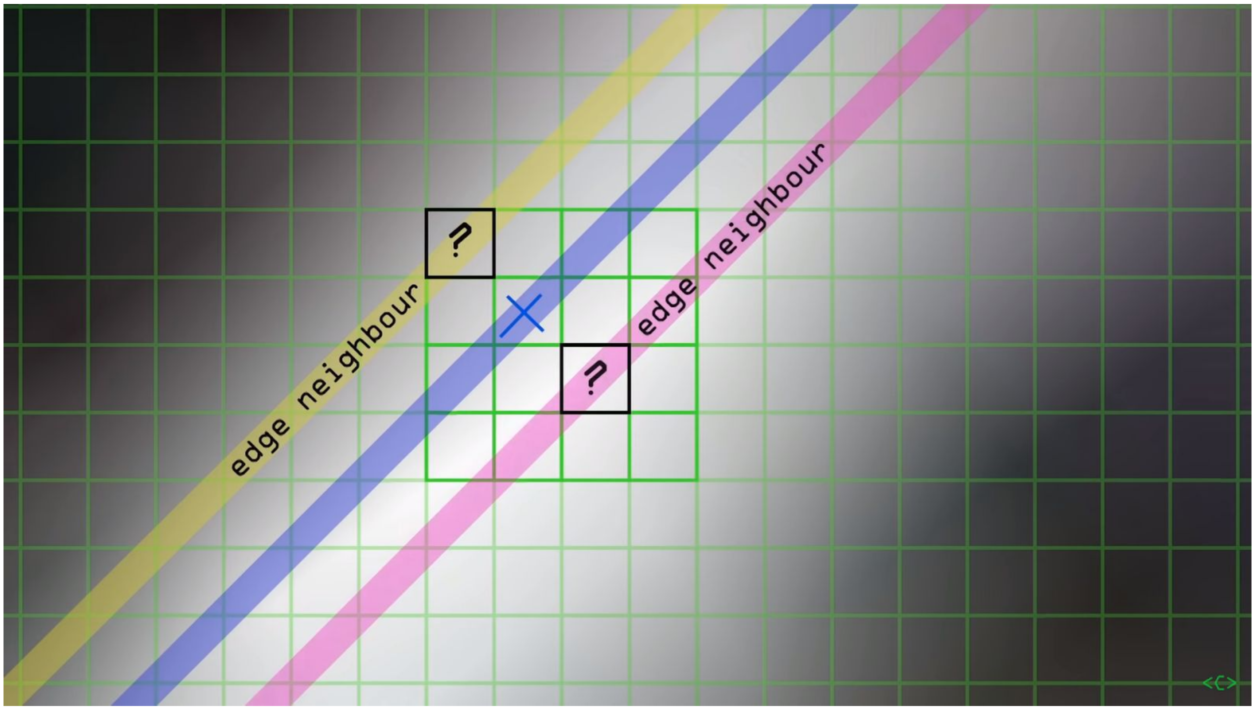
https://www.youtube.com/watch?v=sRFM5IEqR2w

Canny edge detector:

Canny edge detector is better than sobel operator for edge detection

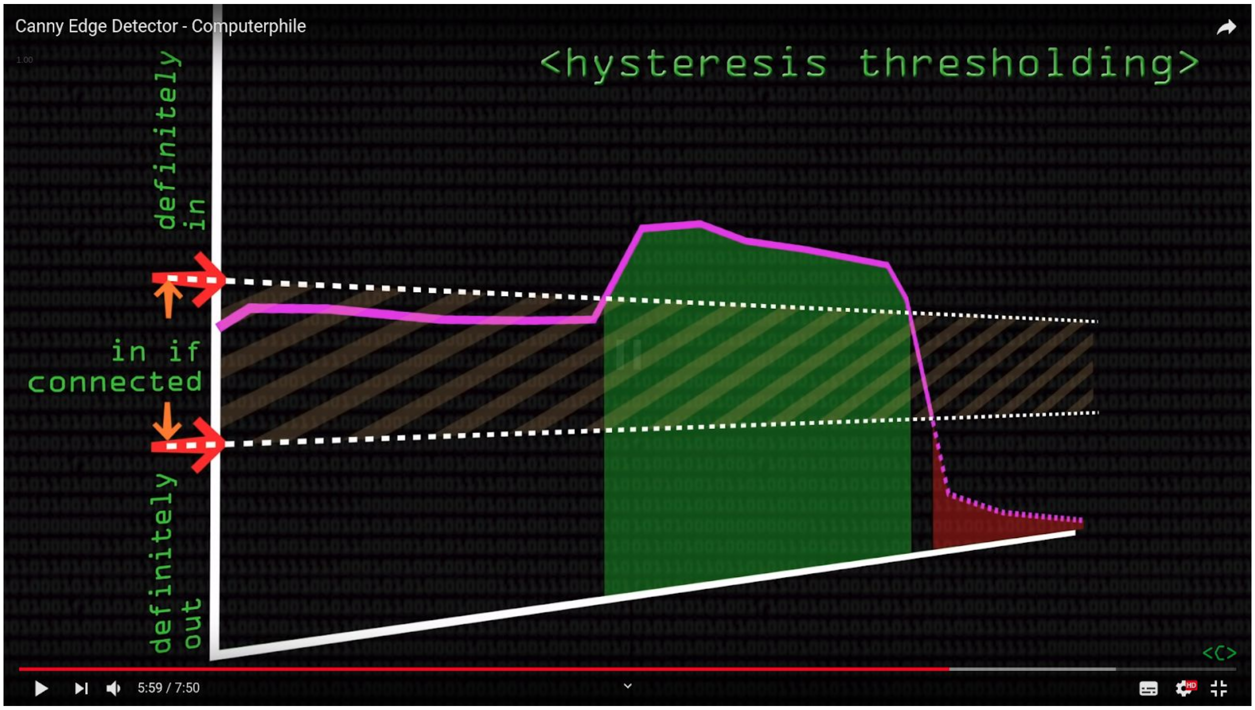
- Input of canny edge is the output of sobel operator
- Canny works by thinning the edges to 1 pixel values -- as thicker pixels are of no use
- In a high resolution of image - gradient will be spread over more pixels, whereas in a low resolution image - gradient will be more narrower
- Sobel operator gives thick and messy edge

First thing to do - is to find out if the pixel is local maximum - and it only compares it with the ones that are perpendicular to the edge. Hence it only takes that particular pixel (with highest gradient - perpendicular to the edge) and creates a thing edge.



→ Hysterisis thresholding - to preserve only the dominating edges. Edge starts with higher threshold but ends at lower threshold - in both the directions.

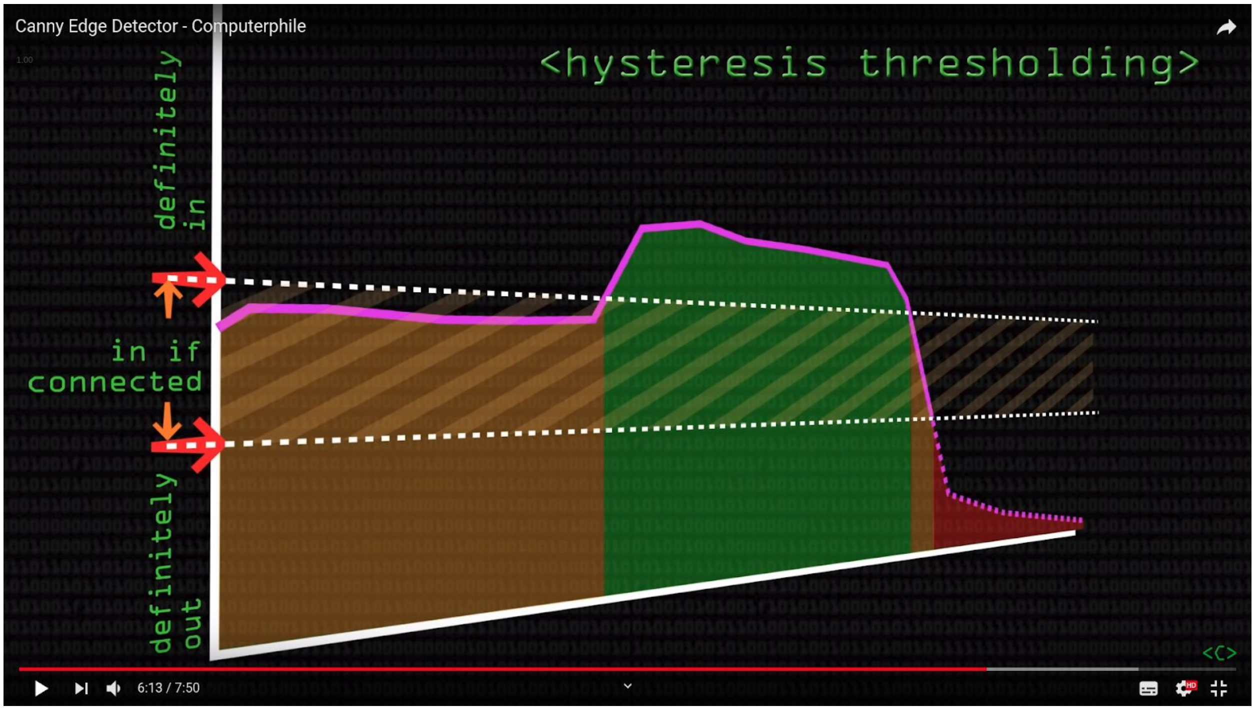
We use thresholds to determine if the edge is strong enough to be preserved


Canny Edge Detector - Computerphile
<hysteresis thresholding>
definitely in
in if connected
definitely out
<C>
5:59 / 7:50

Anything above the top line → definitely in
Anything below the bottom line → definitely out

Anything between the lines → only preserved if it connects to the edges (upward of the top line)

Hence area in brown is selected:


Canny Edge Detector - Computerphile
<hysteresis thresholding>
definitely in
in if connected
definitely out
<C>
6:13 / 7:50

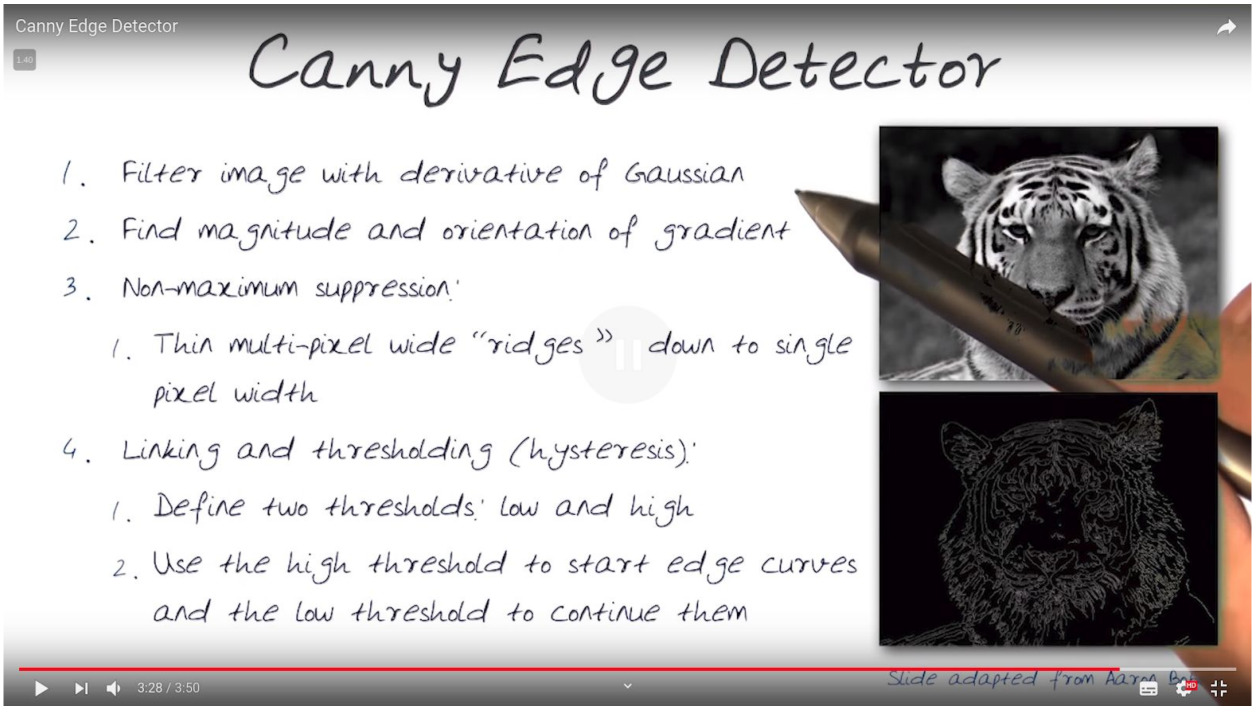
Canny Edge Detector (4)

4. Linking and thresholding (hysteresis):
   1. Define two thresholds: low and high
   2. Use the high threshold to start edge curves and the low threshold to continue them

Note: Actual Output

Edges

Summary:



Canny Edge Detector

1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression:
    1. Thin multi-pixel wide "ridges" down to single pixel width
4. Linking and thresholding (hysteresis):
    1. Define two thresholds: low and high
    2. Use the high threshold to start edge curves and the low threshold to continue them

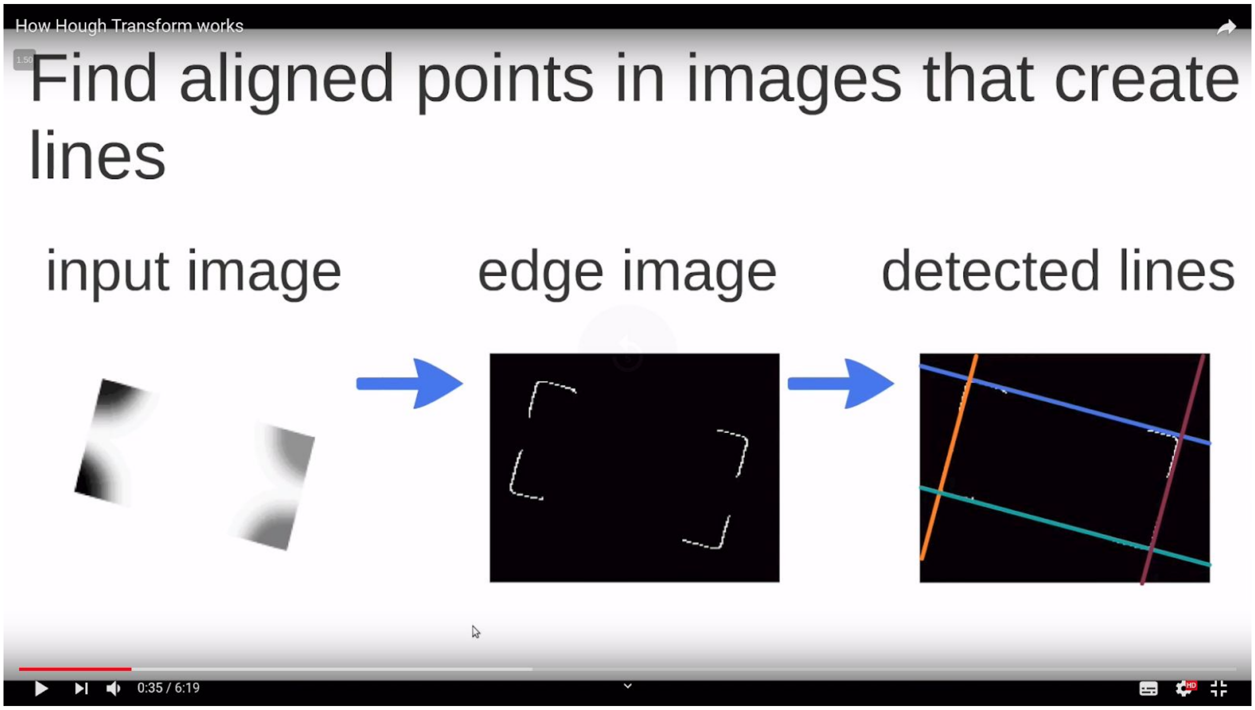
Slide adapted from Aaron Bo

→ Gaussian filter is used in the beginning to remove any unwanted small edges.

Segmenting lane areas:
Create a triangular mask with base as baseline of the image.
This is to black out everything except for that triangle.

Hough transform:
→ to find aligned points in the image that create lines

https://www.youtube.com/watch?v=4zHbI-fFIlI

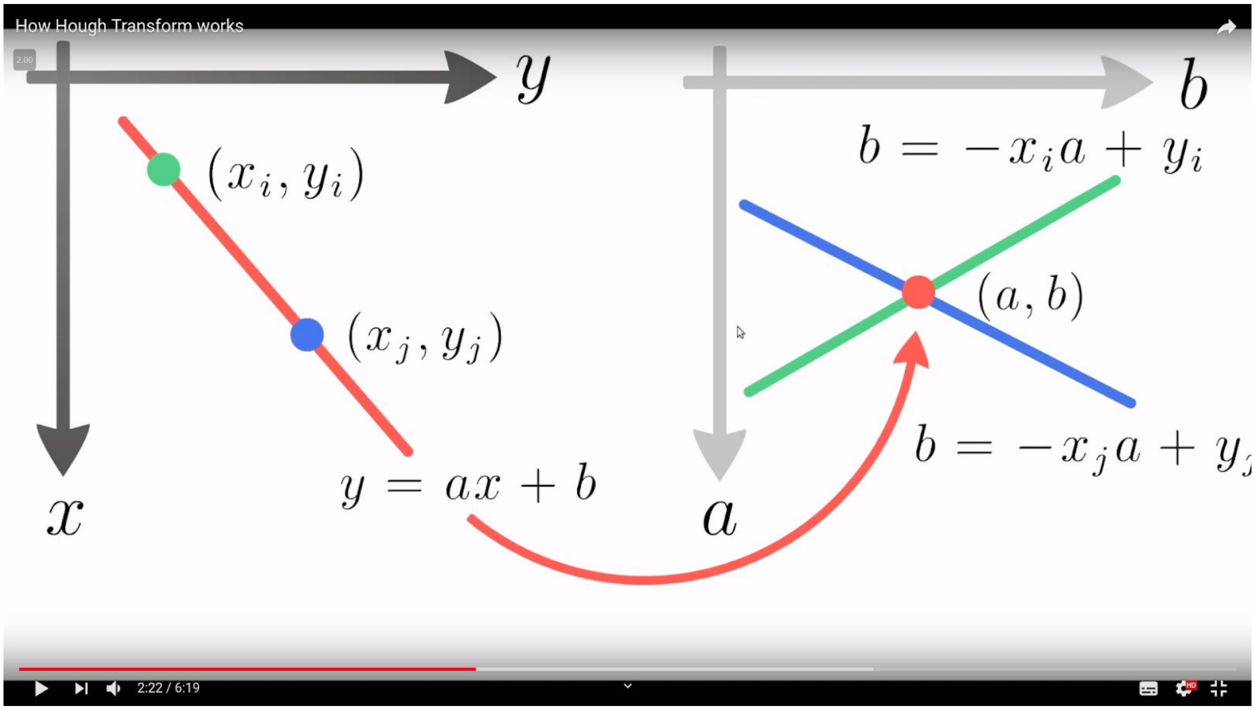


Find aligned points in images that create lines

input image

edge image

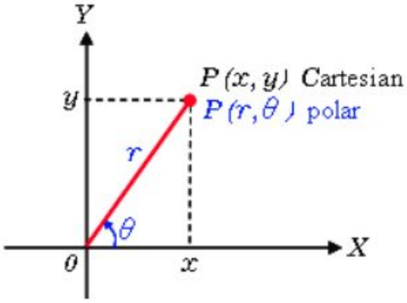
detected lines



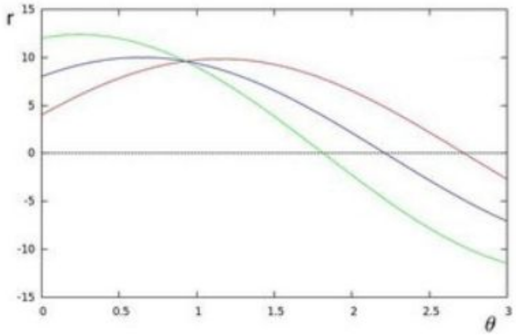
Point in catesian space is line in hough space
And line in cartesian space is point in hough space

<<Important>>

For the simplicity of explanation, we used Cartesian coordinates to correspond to Hough space. However, there is one mathematical flaw with this approach: When the line is vertical, the gradient is infinity and cannot be represented in Hough space. To solve this problem, we will use Polar coordinates instead. The process is still the same just that other than plotting m against b in Hough space, we will be plotting r against θ.



For example, for the points on the Polar coordinate system with `x = 8` and `y = 6`, `x = 4` and `y = 9`, `x = 12` and `y = 3`, we can plot the corresponding Hough space.



We see that the lines in Hough space intersect at `θ = 0.925` and `r = 9.6`. Since a line in the Polar coordinate system is given by `r = xcosθ + ysinθ`, we can induce that a single line crossing through all these points is defined as `9.6 = xcos0.925 + ysin0.925`.

Generally, the more curves intersecting in Hough space means that the line represented by that intersection corresponds to more points. For our implementation, we will define a minimum threshold number of intersections in Hough space to detect a line. Therefore, Hough transform basically keeps track of the Hough space intersections of every point in the frame. If the number of intersections exceeds a defined threshold, we identify a line with the corresponding θ and r parameters.