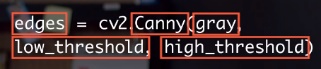
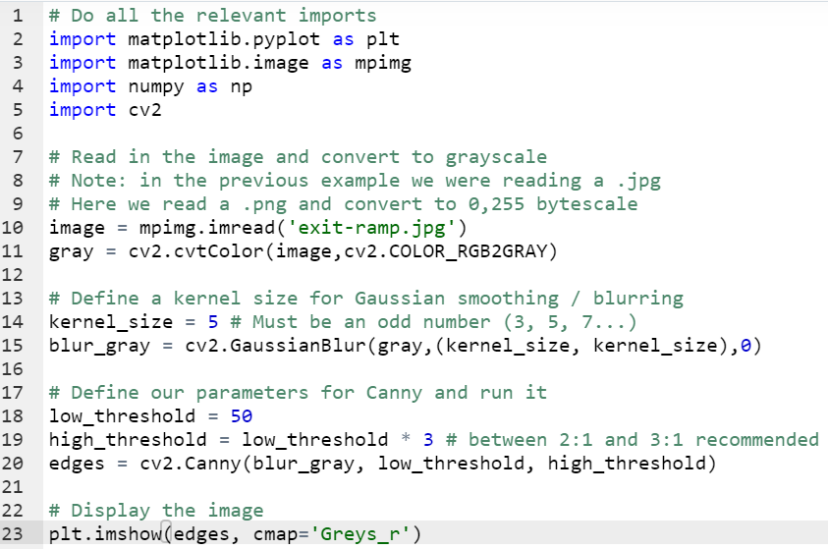
# Color selection



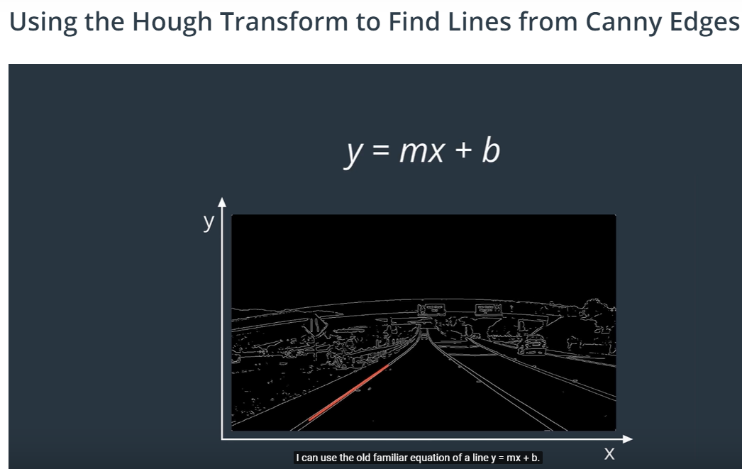
# Canny edge detection

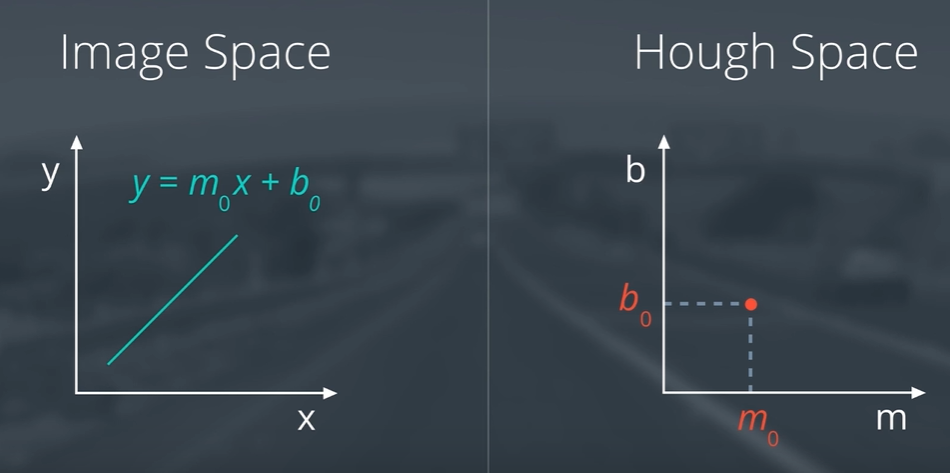
It is an improvement of the previous version. Color and light conditions do not affect.  
The algorithm maps the grey version of the original image into a one black and white (representing edges) by taking the highest gradients in the image. Then it shrinks the output into thin lines.

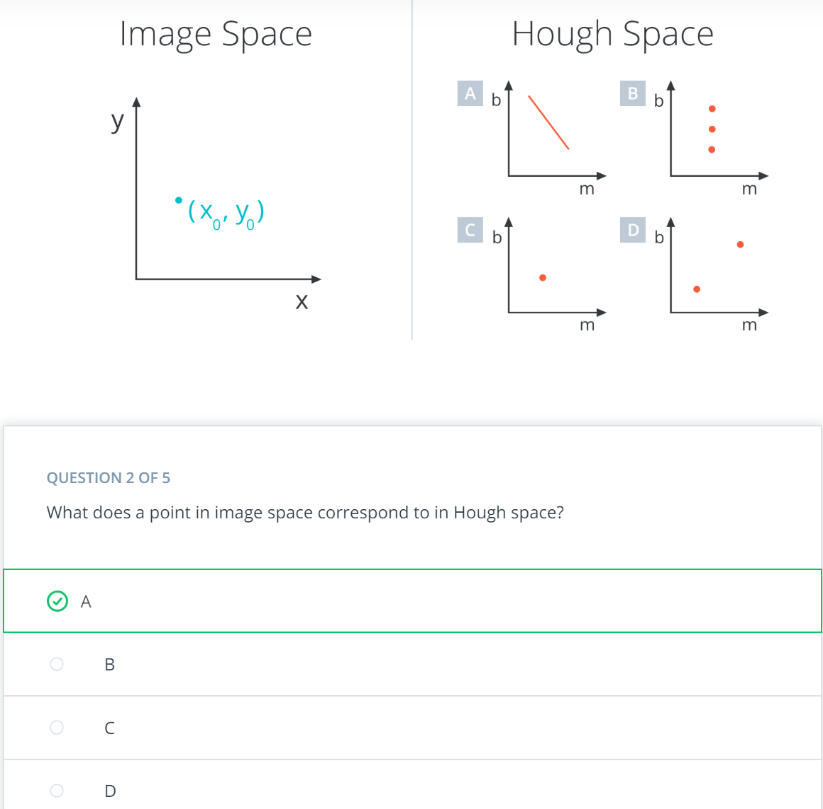
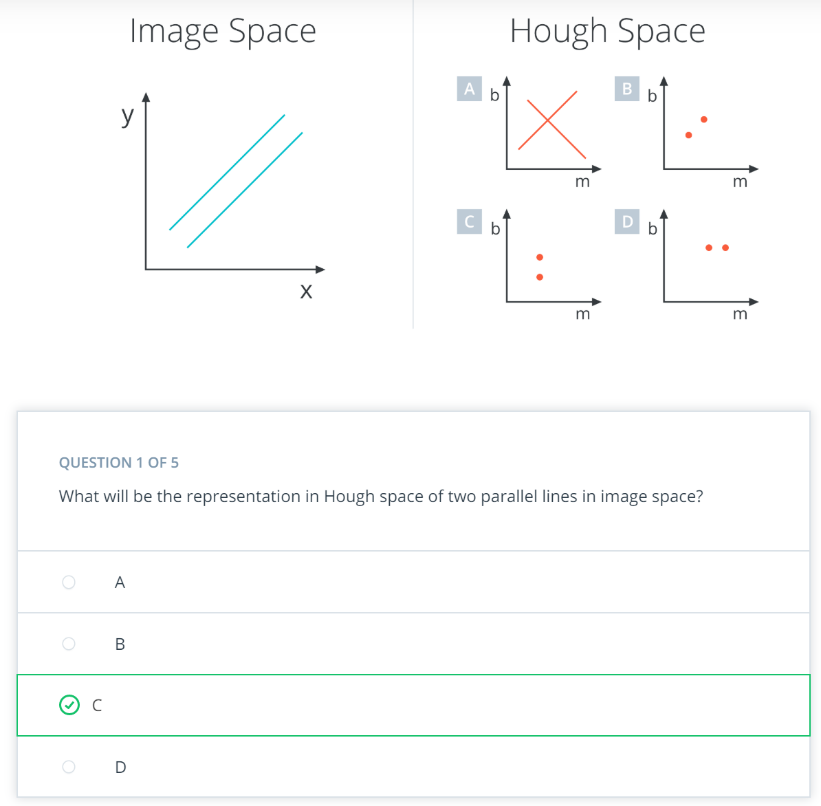




# Hough Transformation

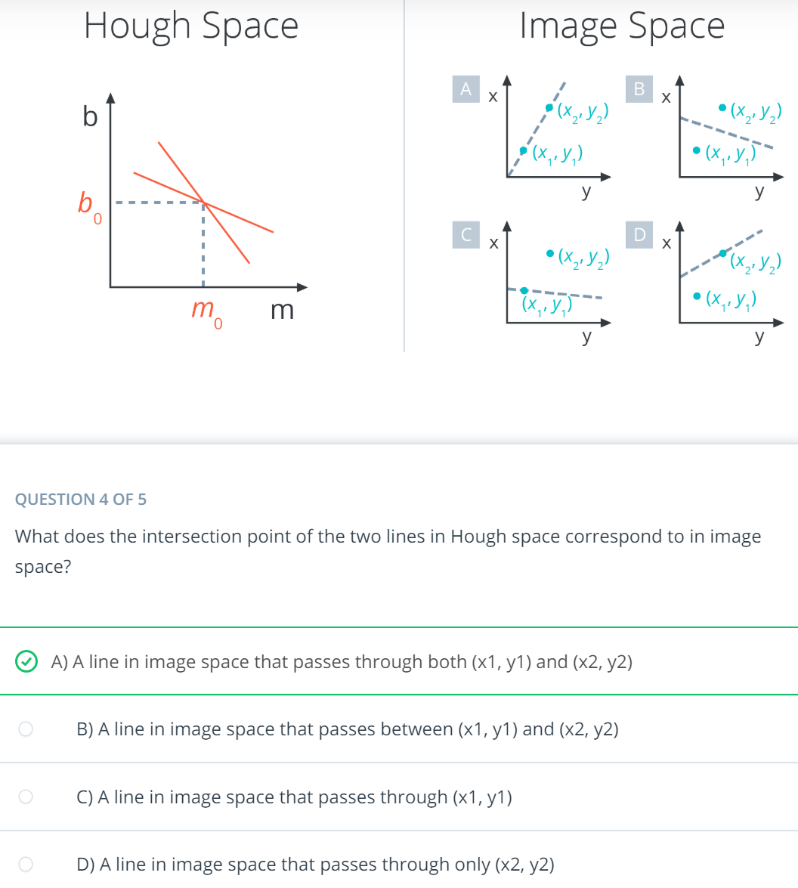
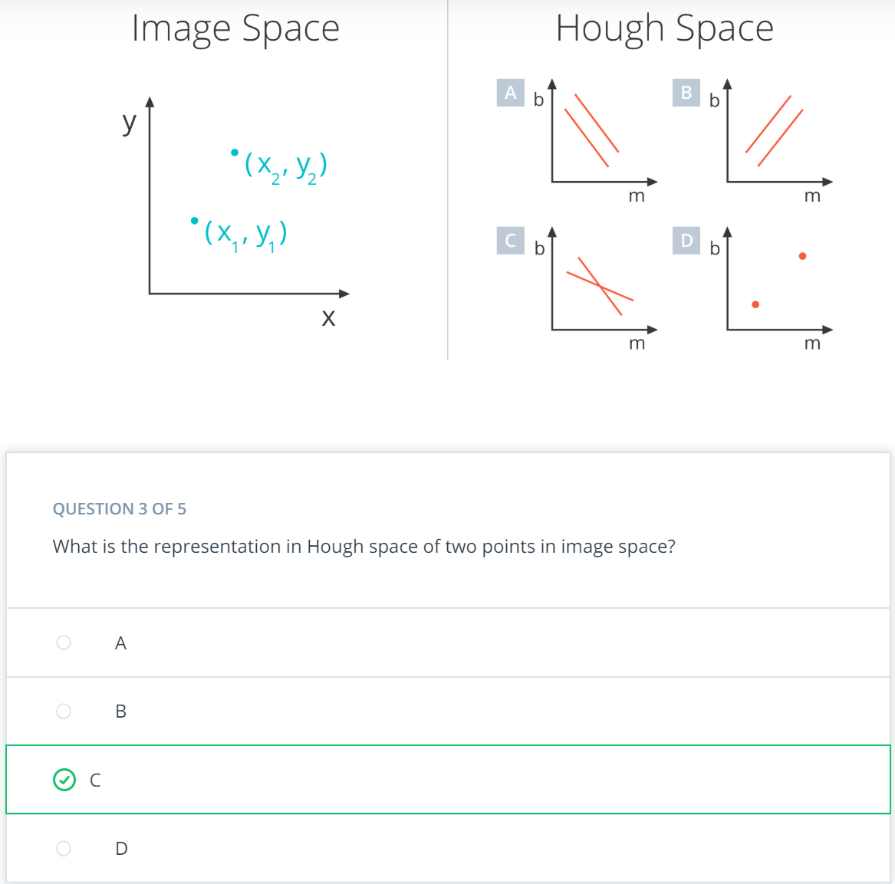




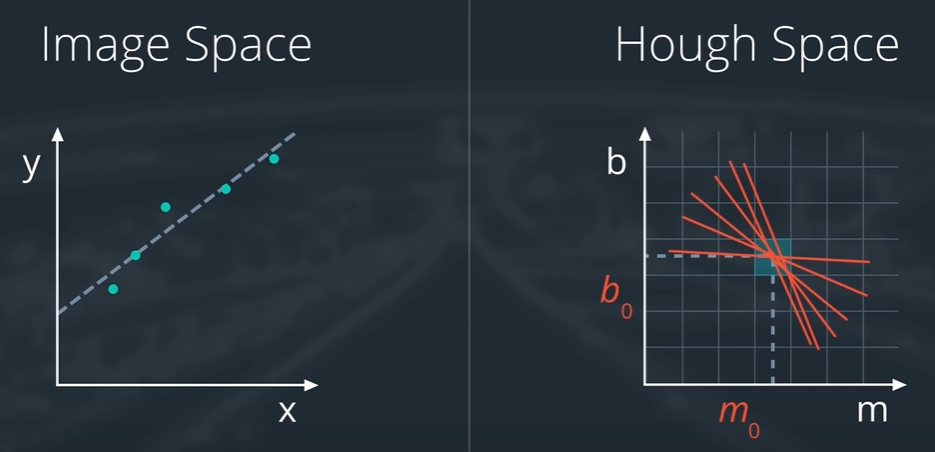


A point in image space describes a line in Hough space. So a line in an image is a point in Hough space and a point in an image is a line in Hough space… cool!

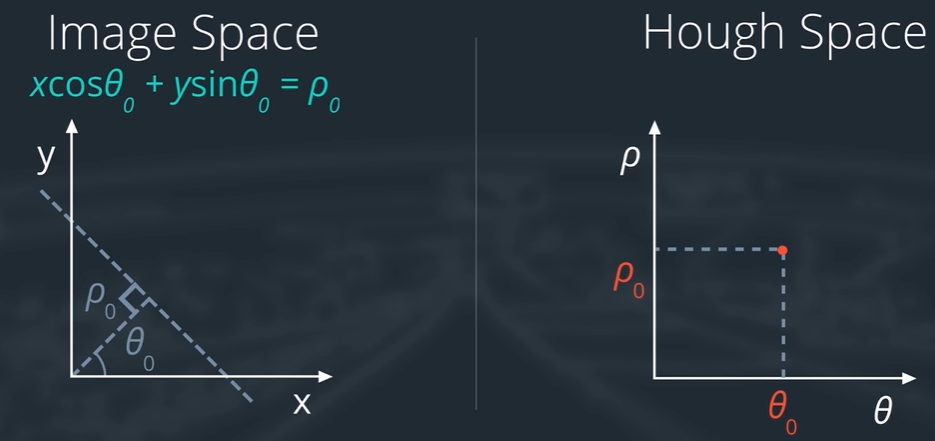
Rearranging the equation of a line, we find that a single point (x,y) corresponds to the line b = y - xm

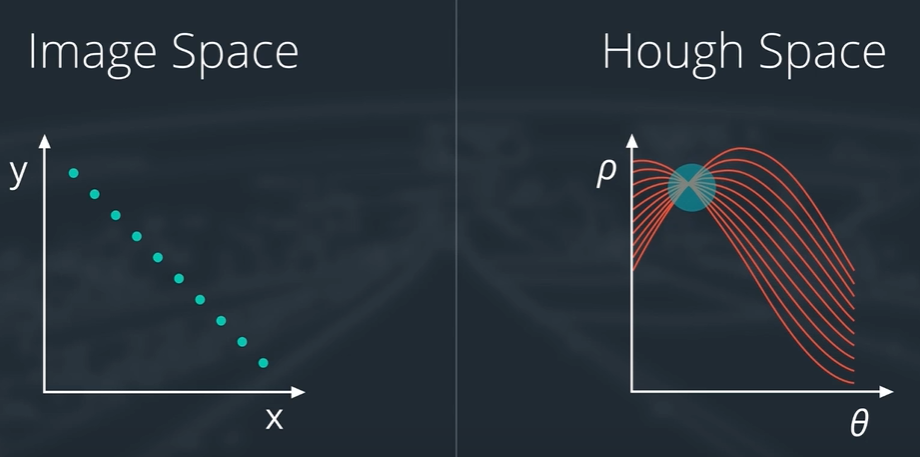


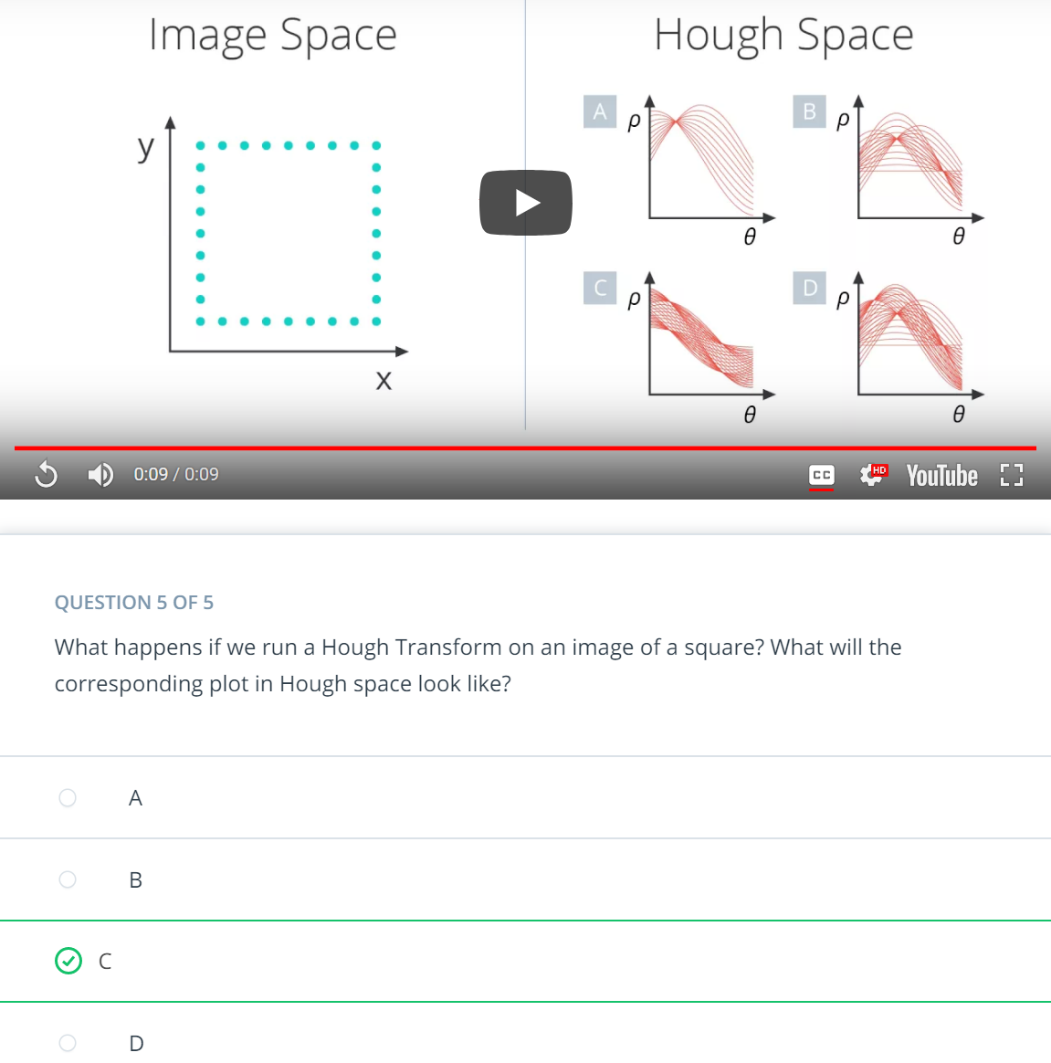
The intersection point at (m0, b0) represents the line y = m0x + b0 in image space and it must be the line that passes through both points!



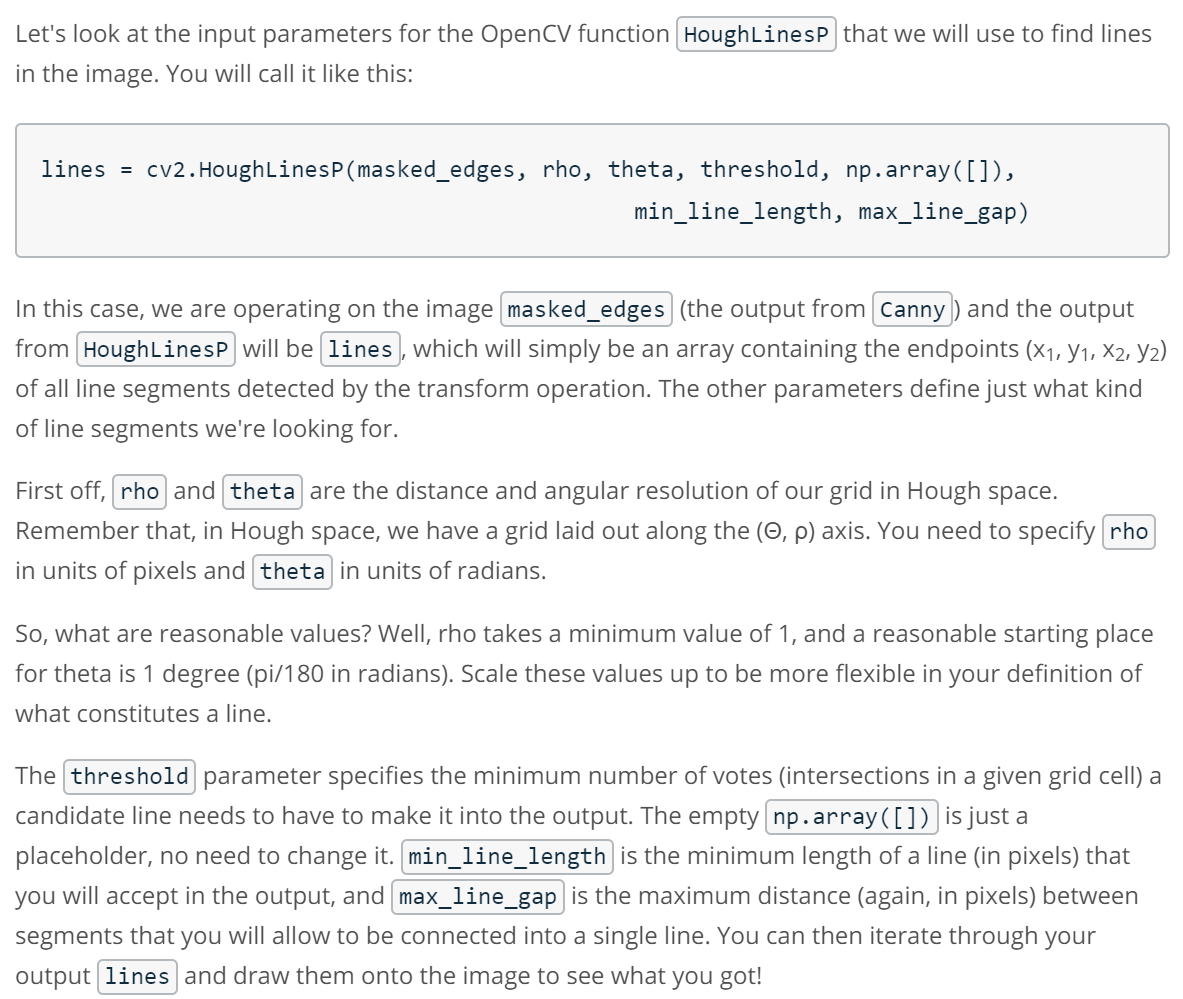
The problem is vertical lines have infinite slope. So we transform to polar coordinates

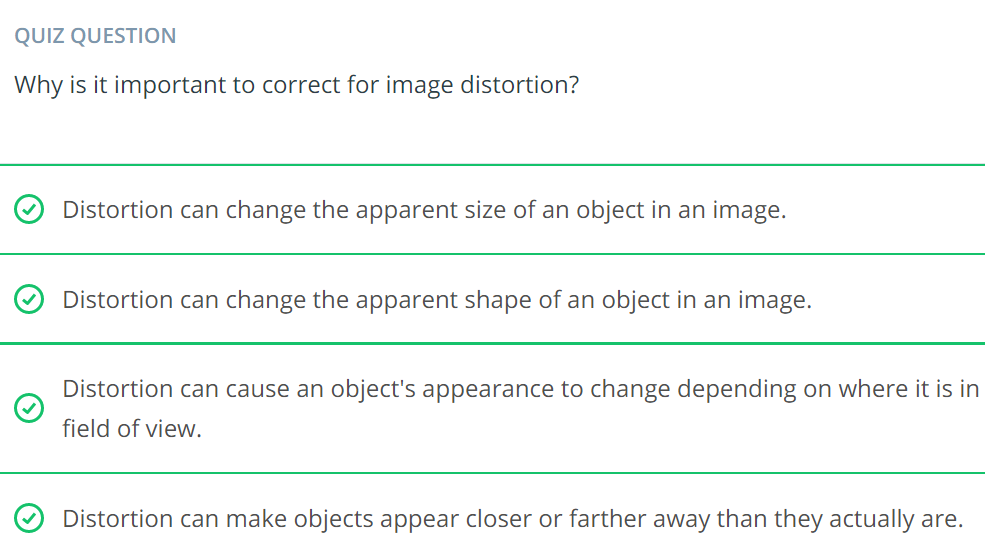


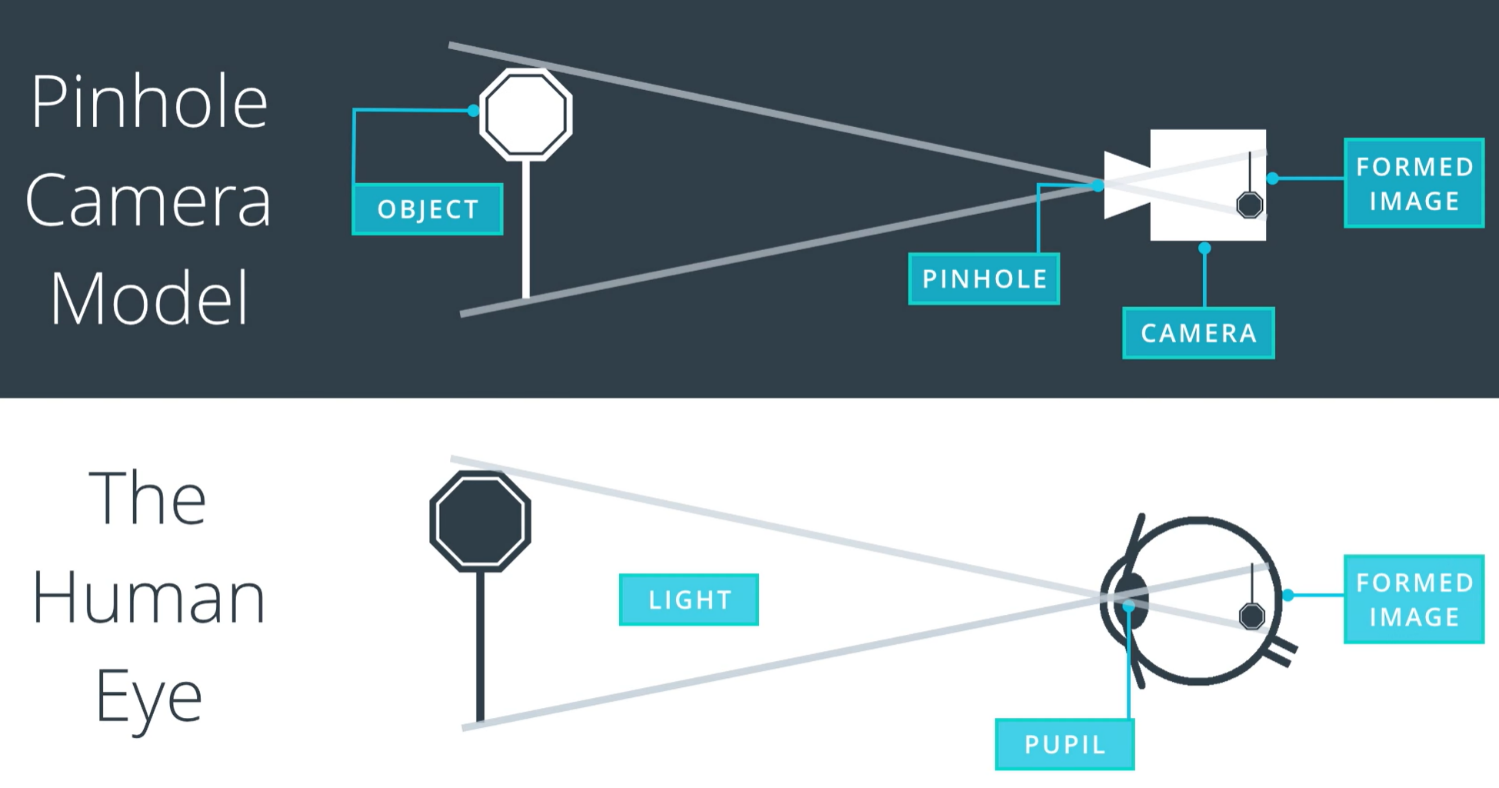


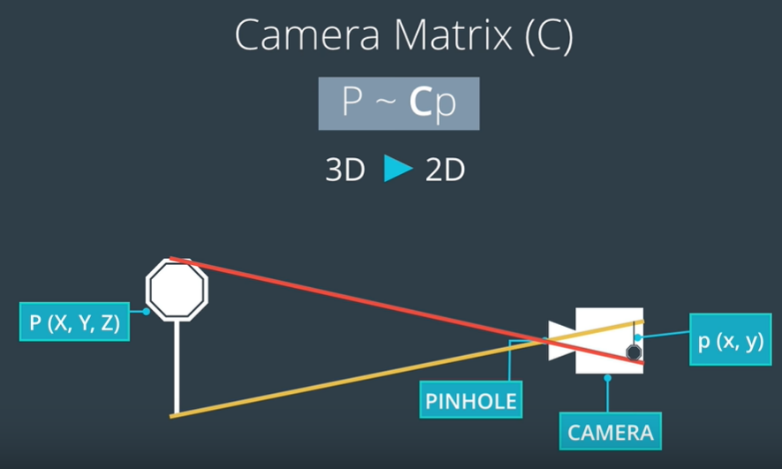


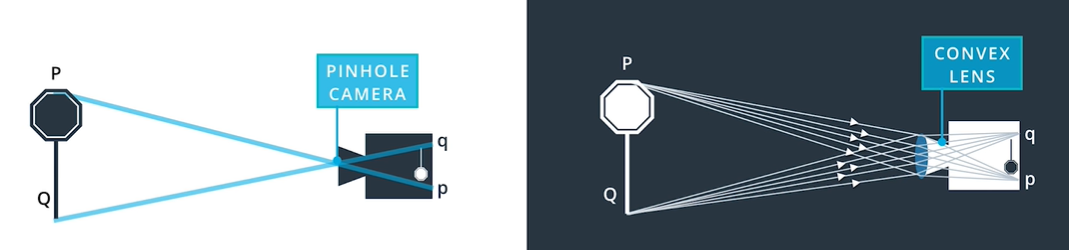
The four major intersections between curves in Hough space correspond to the four sides of the square.





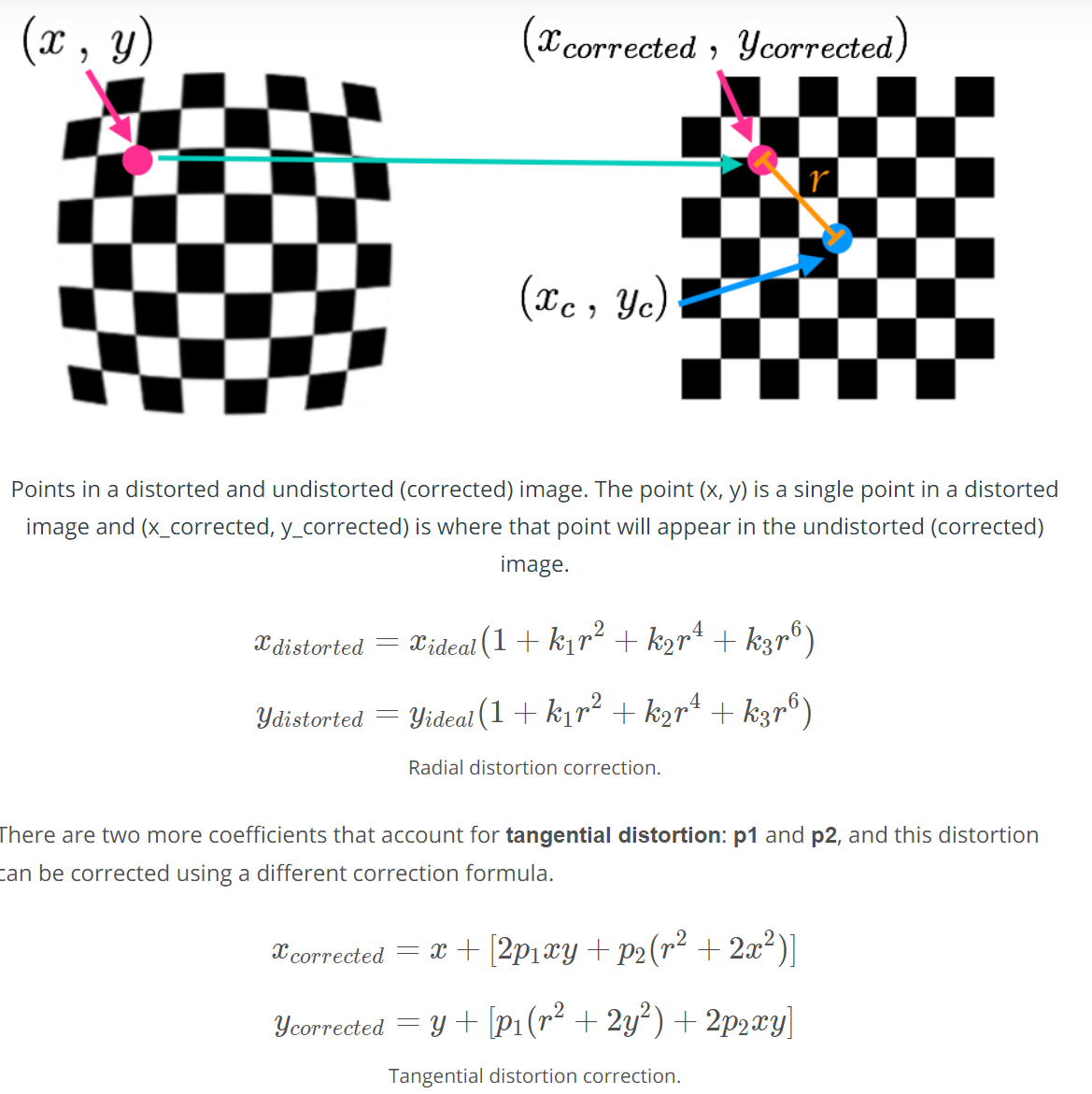


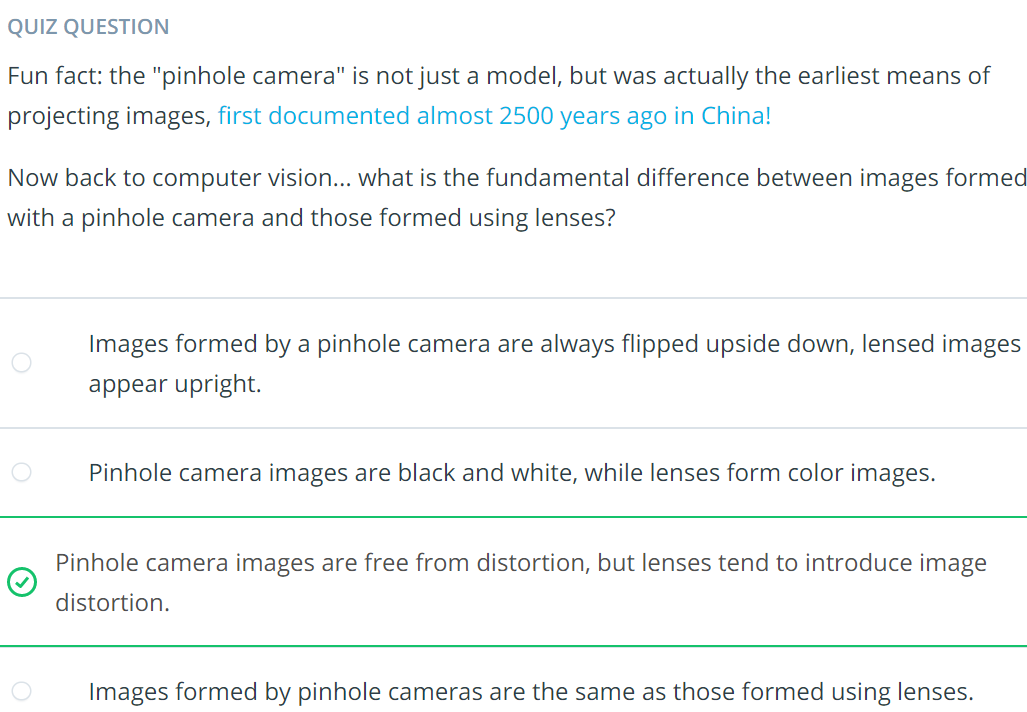


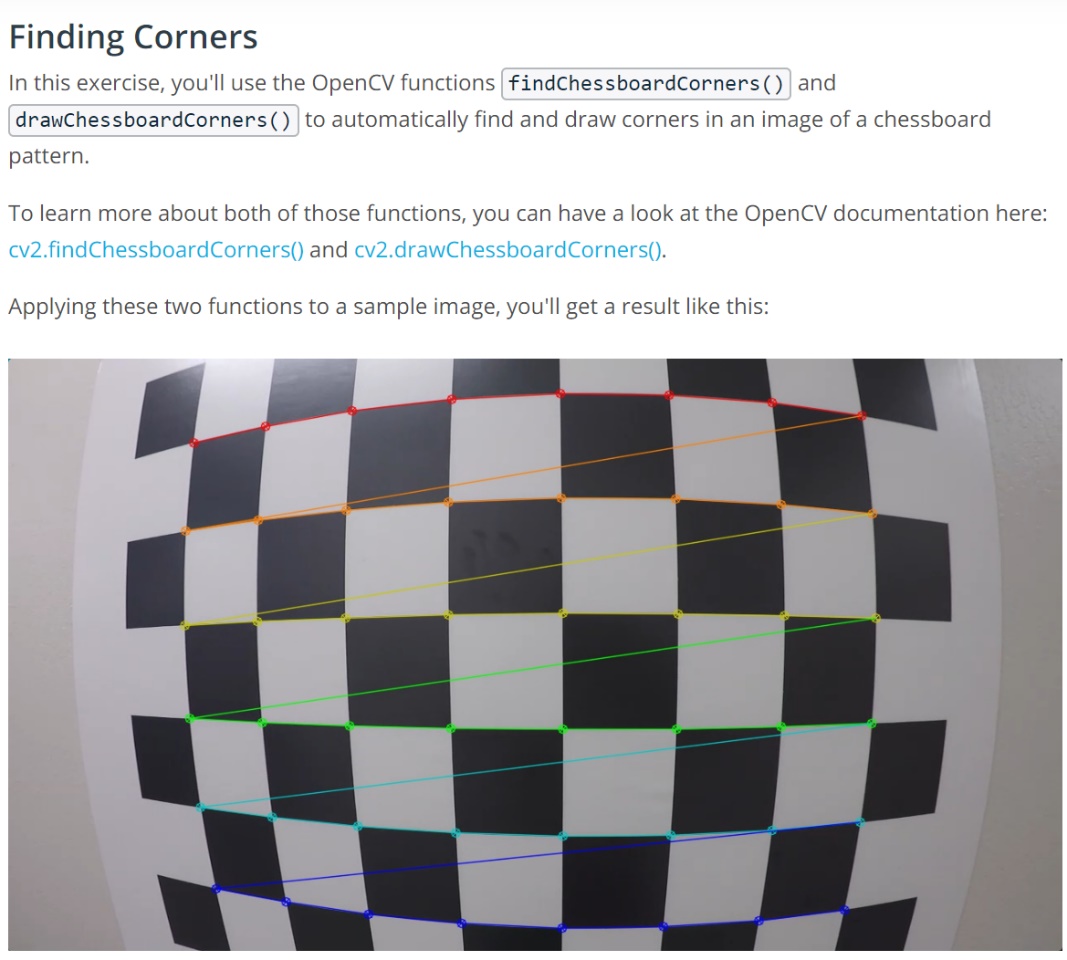


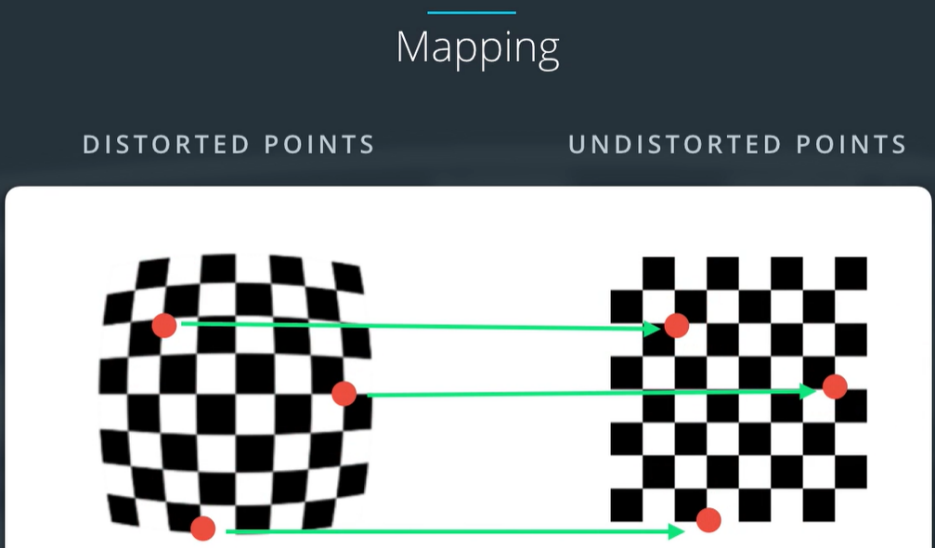


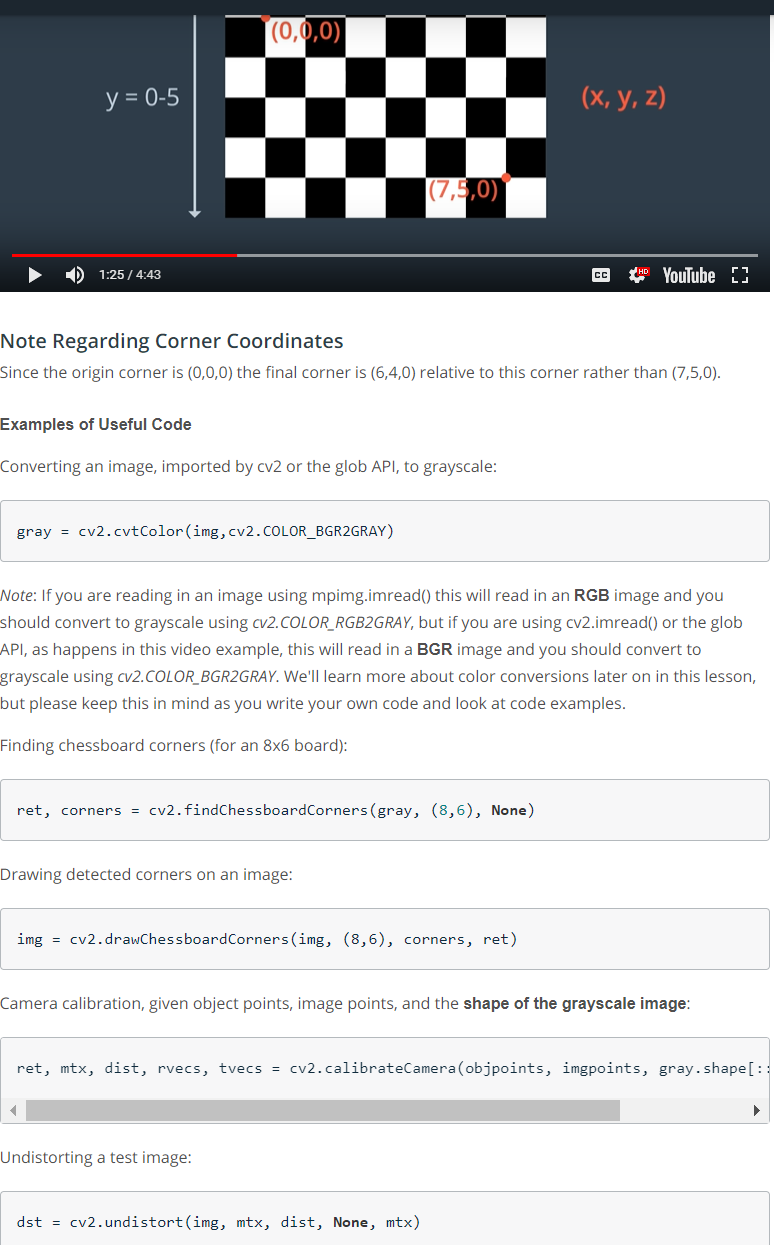


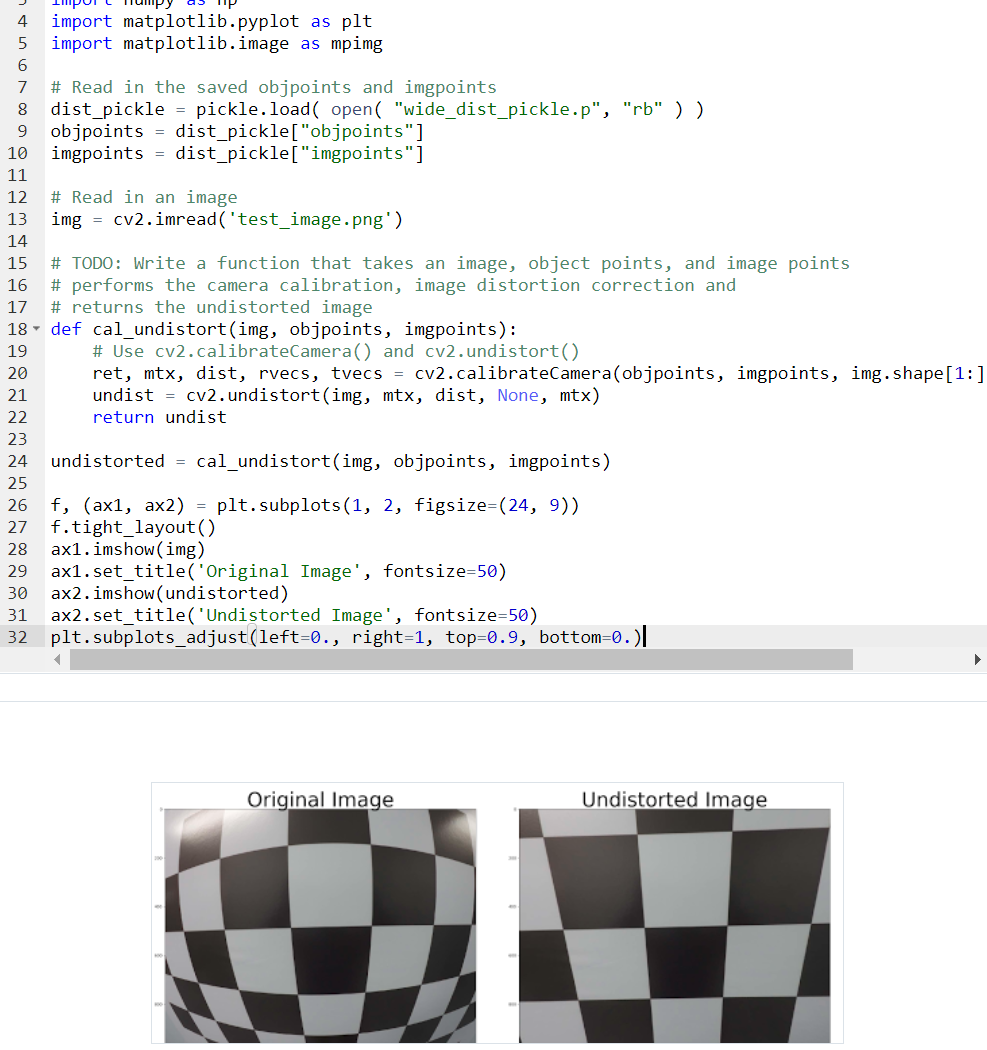












Lane curvature is detected through this pipeline:

Image 🡪 Thresholding to get the lanes 🡪 Masking ROI 🡪 Perspective Transformation 🡪 Detect Lines and detect curvature

