20508345 zljieong Ieong, Zi Liang Jason

**Show that if you use the line equation ρ = x cos θ + y sin θ each image point (x, y) results in a sinusoid in (ρ, θ) Hough space. Relate the amplitude and phase of the sinusoid to the point**

(x, y).

Consider

$$\begin{aligned} & A \cdot \sin\theta + \mu \\ = & A \cdot \sin\mu\cos\theta + A \cdot \cos\mu\sin\theta \end{aligned}$$

As $A\cot\cos\mu$ and $A\cot\sin\mu$ are constant, then

$$\rho = x\cos\theta + y\sin\theta = A \cdot \sin\theta + \mu \iff \begin{cases} x = A\sin\mu \\ y = A\cos\mu \end{cases}$$

Now $\rho$ is the distance between origin and a line extended from (x,y). As the extended line $\perp \rho$ , then $\sqrt{x^2+y^2} \geq \rho$

As $x$ and $y$ are constant, $\exists\theta \in [0, 2\pi)$ such that $\rho = \sqrt{x^2+y^2}$

Note that $-1 \leq sin\theta + \mu \leq 1$, and the sigmoid function reach maximal at $\rho = \sqrt{x^2+y^2} \to A = \sqrt{x^2+y^2}$

Then $\begin{cases} \mu = \arcsin\frac{x}{\sqrt{x^2+y^2}} \\ \mu = \arccos\frac{y}{\sqrt{x^2+y^2}} \end{cases}$

Then $\mu = \arctan\frac{x}{y}$

**Why do we parametrize the line in terms (ρ, θ) instead of the slope and intercept (m, c)?**

$m$ is unbounded and is a bad design for a voting system

**Express the slope and intercept in terms of (ρ, θ).**

$m = \tan(\theta + \pi/2)$

let $C$ be the intercept and $\vec{\mathrm{P}}$ be the vector between O and the line such that $||\vec{\mathrm{P}}|| = \rho$

Consider $\vec{\mathrm{P}} \cdot (\overrightarrow{OC} - \vec{\mathrm{P}}) = 0$

$\vec{\mathrm{P}} \cdot \overrightarrow{OC} = \rho \times c \cdot \cos\frac{\pi}{2} - \theta = \rho^2$

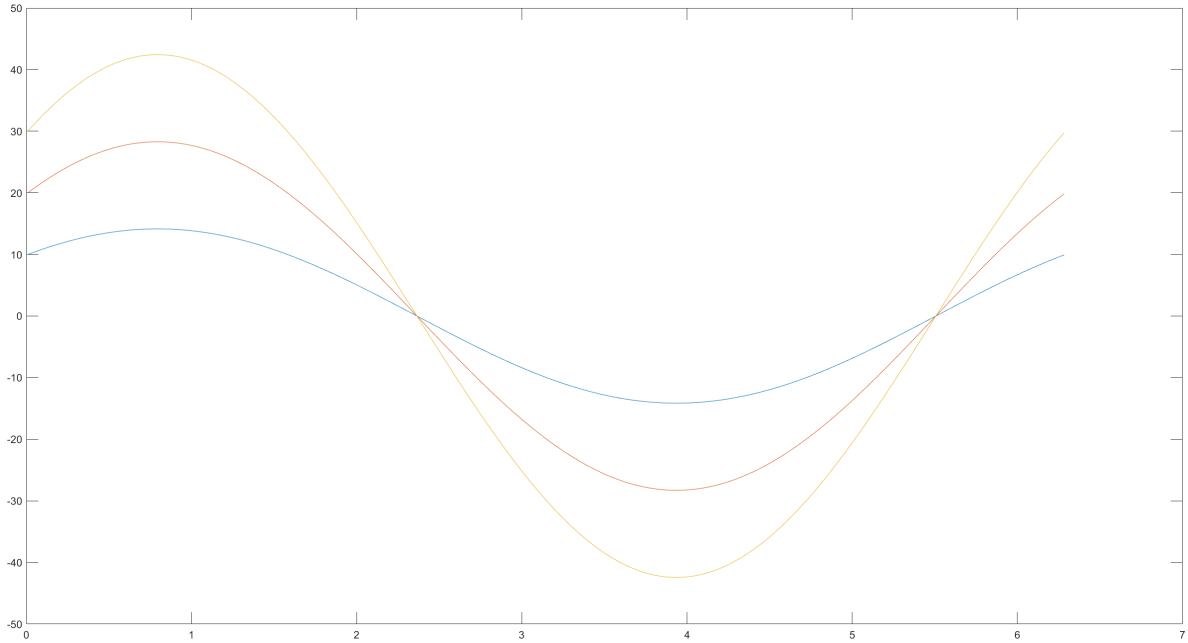
$c = \frac{\rho}{\cos\frac{\pi}{2}-\theta}$

**Assuming that the image points (x, y) are in an image of width W and height H, that is, x ∈ [1, W], y ∈ [1, H], what is the maximum absolute value of ρ, and what is the range for θ?**

$\rho \leq \lceil\sqrt{W^2+H^2}\rceil$(from the sinusoid function)

$\theta \in [0, 2\pi)$ (the domain of the sinusoid function)

**For point (10, 10) and points (20, 20) and (30, 30) in the image, plot the corresponding sinusoid waves in Hough space, and visualize how their intersection point defines the line. What is (m, c) for this line? Please use Matlab to plot the curves and report the result in your write-up.**




The peek occurs at $(\rho, \theta) = \begin{cases} (0, \frac{3\pi}{4}) \\ (0, \frac{7\pi}{4}) \end{cases}$ , then the corresponding (m,c) is

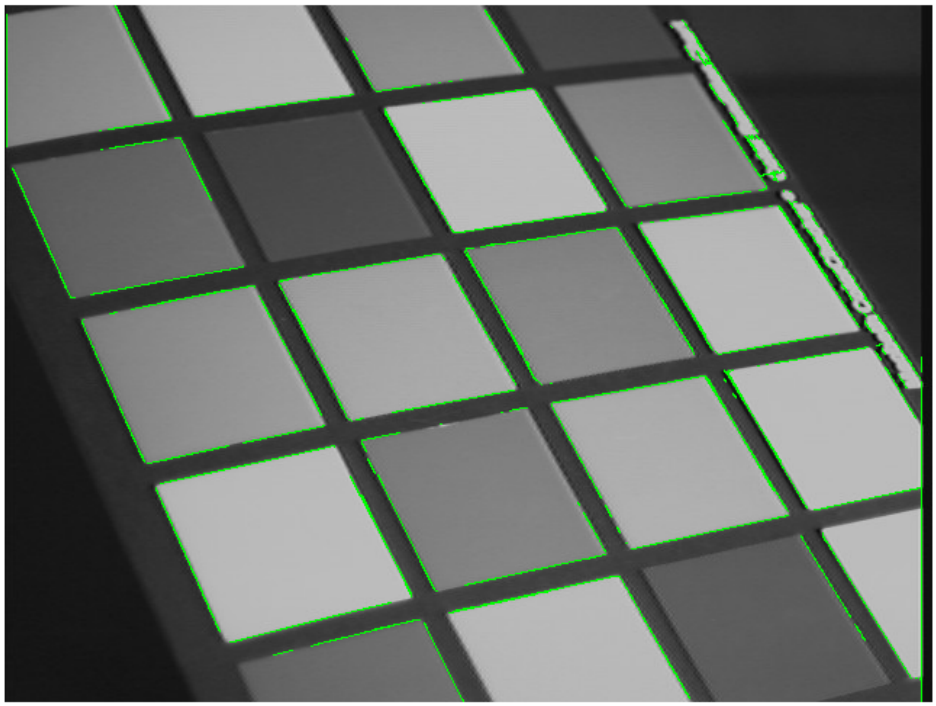
$m = \tan\frac{\pi}{2} + \frac{3\pi}{4} = \tan\frac{\pi}{2} + \frac{3\pi}{4} = 1, c = 0$

# Experiments

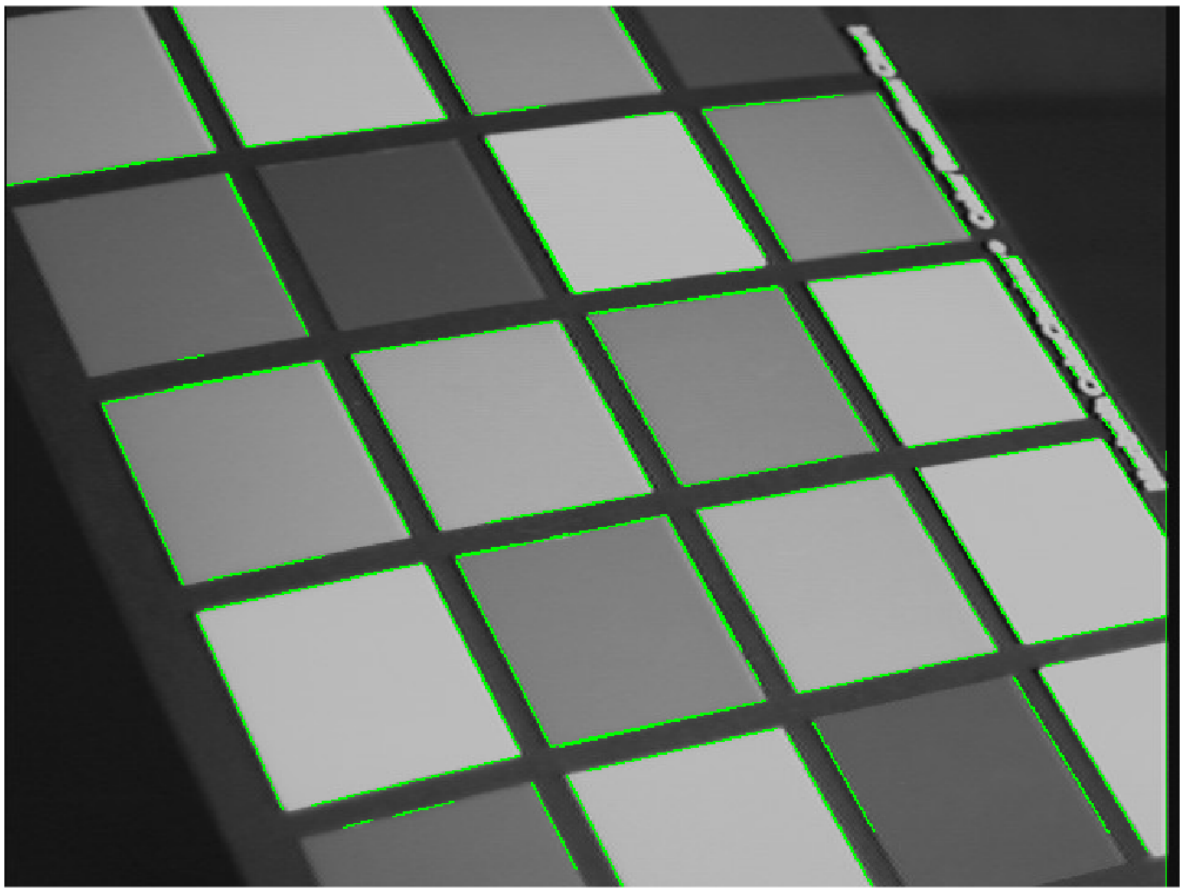
## Q2.5 Implement houghlines yourself

**In Q2.5, we used the Matlab built-in function houghlines to prune the detected lines into line segments that do not extend beyond the objects they belong to. Now, its our turn to implement one ourselves! Please write a function named myHoughLineSegments and then compare your results with the Matlab built-in function in your write-up. Show at least one image for each and briefly describe the differences.**

My implememtation:



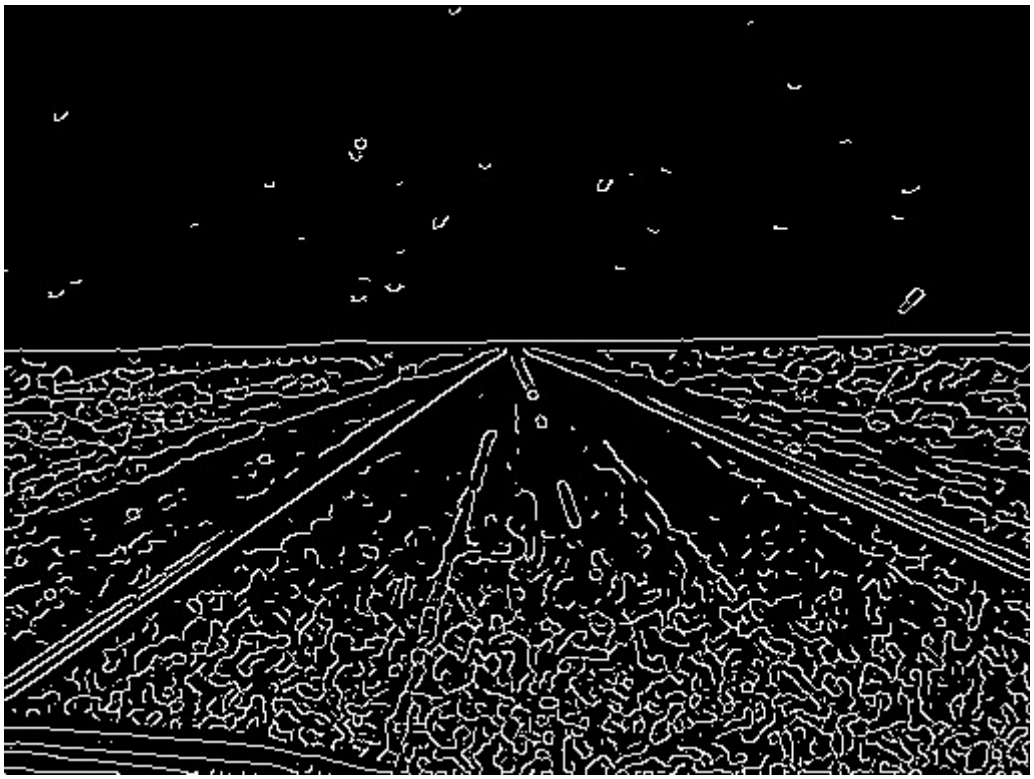
Matlab's built in function:



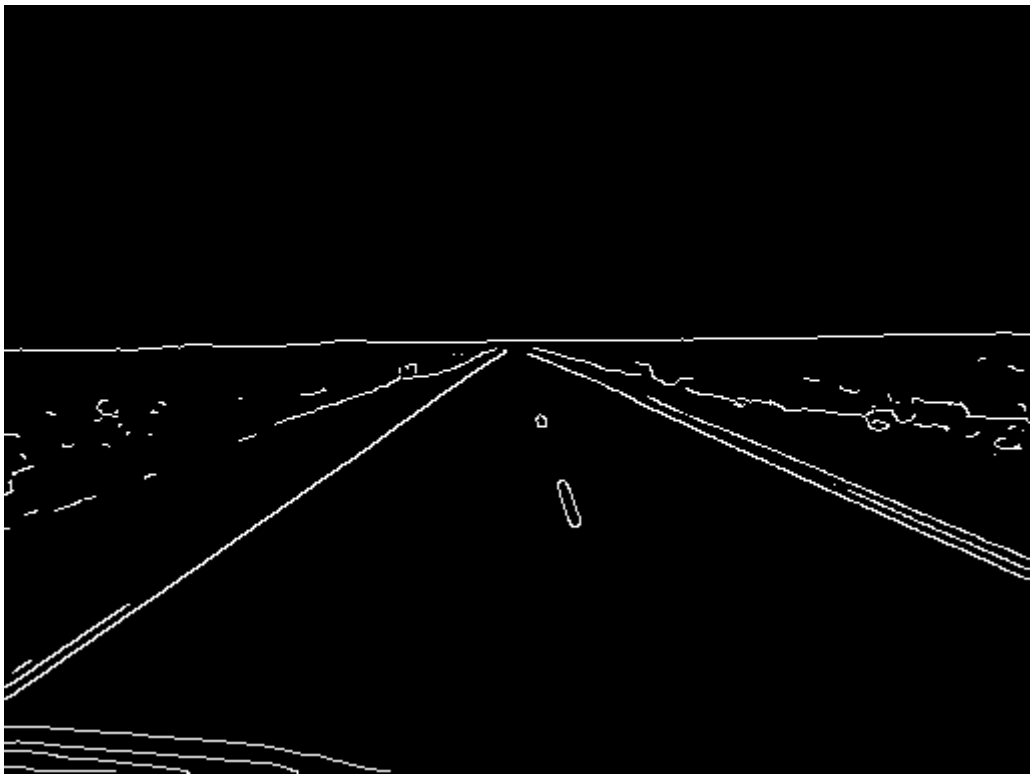
My implementation gives more false positive and does not fill up small gaps. The false positive should come from different parameter of the maximum distance. Not filling up the gaps may increase the number of lines segments and the draw lines function will take more time

## Q3.1 Use the script included to run your Hough detector on the image set and generate intermediate!

**output images. Include the set of intermediate outputs for one image in your write-up. Did your code work well on all the image with a single set of parameters? How did the optimal set of parameters vary with images? Which step of the algorithm causes the most problems? Did you find any changes you could make to your code or algorithm that improved performance? In your write-up, you should describe how well your code worked on different images, what effect do the parameters have and any improvements you made to your code to make it work better.**



My original implementation of edge detection have lots of unwanted edges that are local maximal but not what a human would be interested in.  A single set of parameters won't work for all images and setting any fixed threshold doesn't make much sense as well.



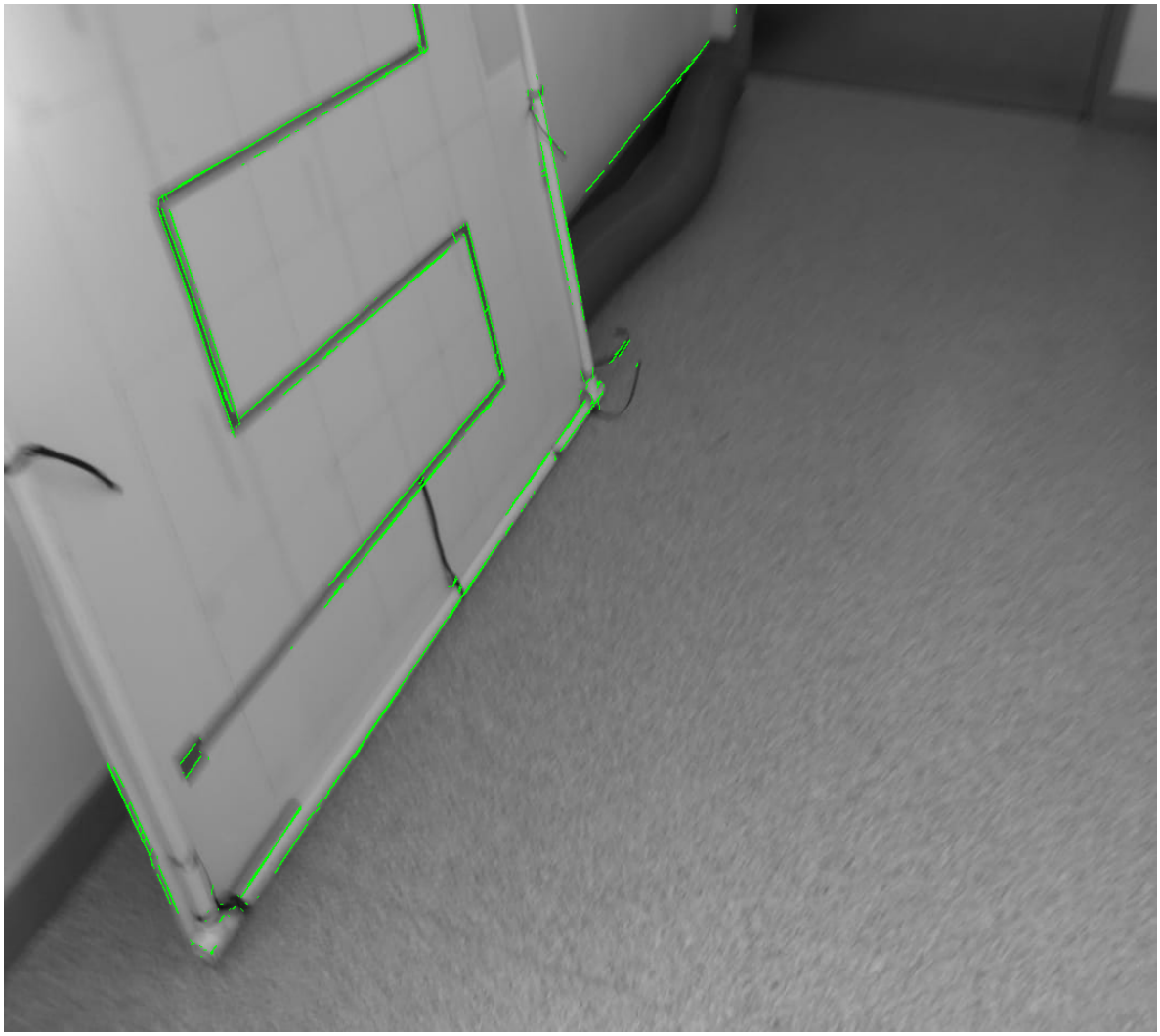
Adaptive thresholding seemes to be working better over different images. The above images is achieved by removing edges with amplitude lower then .15~.25 of the maximum amplitude. However, this may also lead to losing track of some edges. This may also have caused some gaps to appear. I have tried the canny and the result doesn't seems to be much different from sobel.

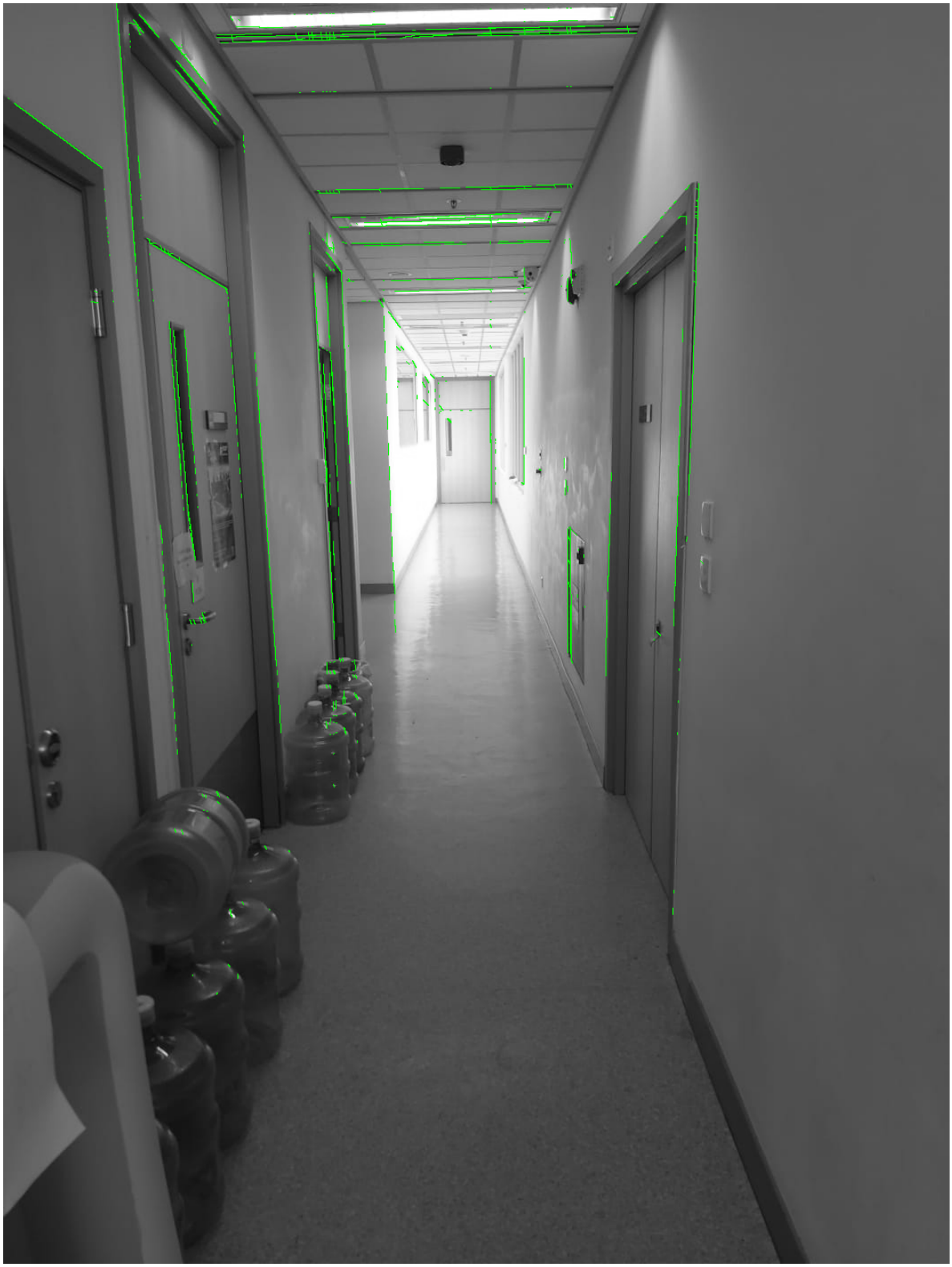
If I have to detect the road mark on the two sides of the road in practice, bluing more and filtering edges that are not white may also be useful

## Try your own images!

## Q4.1x Implement houghlines yourself

**Take five pictures, either with a camera of your own, or from the Internet. Write a script ec.m to take care of reading in your images (use a relative path here, not absolute), making function calls to the various steps of the Hough transform, and generating images showing the output and some of the intermediate steps (like houghScript.m). Submit your own images and ec.m in ec/. Please include resulting images in your write-up.**






3D PRINTING
FILAMENT