Q1

a) Define $x' = xcos\theta + ysin\theta$ and $y' = -xsin\theta + ycos\theta$ to be the new coordinates formed by rotating (x, y) by an arbitrary angle θ.

$$f_x = \frac{\partial f}{\partial x} = \frac{\partial f}{\partial x'} \times \frac{\partial x'}{\partial x} + \frac{\partial f}{\partial y'} \times \frac{\partial y'}{\partial x} = cos\theta f_{x'} - sin\theta f_{y'}$$
$$f_y = \frac{\partial f}{\partial y} = \frac{\partial f}{\partial x'} \times \frac{\partial x'}{\partial y} + \frac{\partial f}{\partial y'} \times \frac{\partial y'}{\partial y} = sin\theta f_{x'} + cos\theta f_{y'}$$

$$f_{xx} = \frac{\partial f_x}{\partial x} = \frac{\partial}{\partial x}\left(cos\theta f_{x'} - sin\theta f_{y'}\right)$$
$$= \frac{\partial\left(cos\theta f_{x'} - sin\theta f_{y'}\right)}{\partial x'} \times \frac{\partial x'}{\partial x} + \frac{\partial\left(cos\theta f_{x'} - sin\theta f_{y'}\right)}{\partial y'} \times \frac{\partial y'}{\partial x}$$
$$= cos\theta(cos\theta f_{x'x'} - sin\theta f_{y'x'}) - sin\theta(cos\theta f_{x'y'} - sin\theta f_{y'y'})$$
$$f_{yy} = \frac{\partial f_y}{\partial y} = \frac{\partial}{\partial y}\left(sin\theta f_{x'} + cos\theta f_{y'}\right)$$
$$= \frac{\partial\left(sin\theta f_{x'} + cos\theta f_{y'}\right)}{\partial x'} \times \frac{\partial x'}{\partial y} + \frac{\partial\left(sin\theta f_{x'} + cos\theta f_{y'}\right)}{\partial y'} \times \frac{\partial y'}{\partial y}$$
$$= sin\theta(sin\theta f_{x'x'} + cos\theta f_{y'x'}) + cos\theta(sin\theta f_{x'y'} + cos\theta f_{y'y'})$$

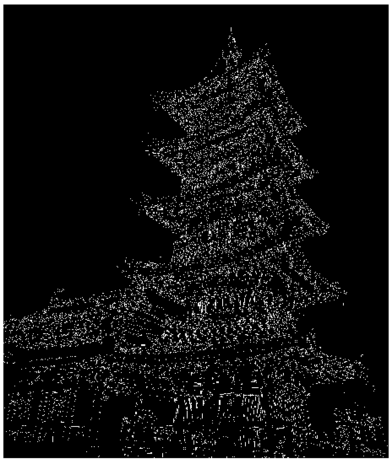
$$f_{xx} + f_{yy} = cos\theta(cos\theta f_{x'x'} - sin\theta f_{y'x'}) - sin\theta(cos\theta f_{x'y'} - sin\theta f_{y'y'})$$
$$+ sin\theta(sin\theta f_{x'x'} + cos\theta f_{y'x'}) + cos\theta(sin\theta f_{x'y'} + cos\theta f_{y'y'})$$
$$= f_{x'x'} - 2sin\theta cos\theta f_{y'x'} + 2sin\theta cos\theta f_{y'x'} + f_{y'y'} = f_{x'x'} + f_{y'y'}$$

This means $\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \frac{\partial^2 f}{\partial x'^2} + \frac{\partial^2 f}{\partial y'^2}$ for two arbitrary coordinates (x, y) and (x', y').

Therefore Laplacian is rotation invariant.

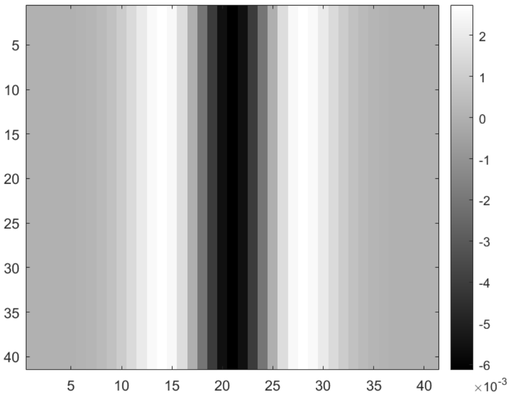
b) The assumption is called the condition of linear variation: the intensity variation near and parallel to the line of zero-crossings should locally be linear.
We want the orientation of the directional derivative to coincide with the local orientation of the underlying line of zero-crossings. If the linear variation holds, we have the results 1) the orientation of the line of zero-crossings is perpendicular to the orientation at which the zero-crossings have maximum slope. 2) the lines of zero-crossings are precisely the zero-crossings of the orientation-independent differential operator, the Laplacian. Then the zero-crossings can be detected and accurately located by the zero values of the Laplacian.

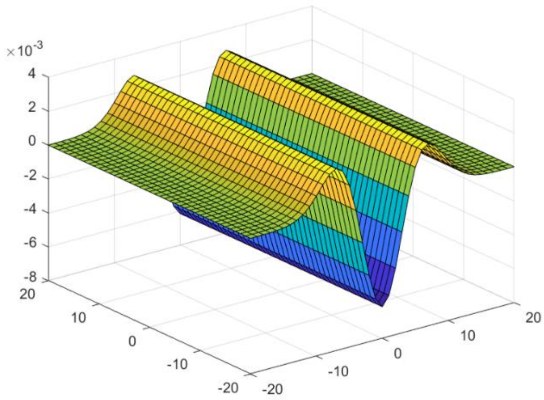
c) The Marr-Hildreth may work well with images which have continuous and smooth edges, and images which don’t have so many details. It tends to fail with sharp shape and object with too many details on it, since intensities don’t change linearly. It perceives many edges, but locality is not good and the edges are too spotty or thick to really identify the features.

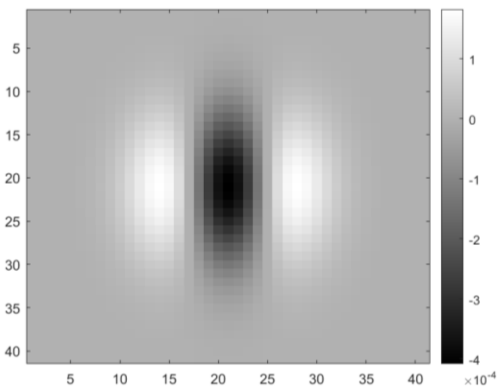

Q4

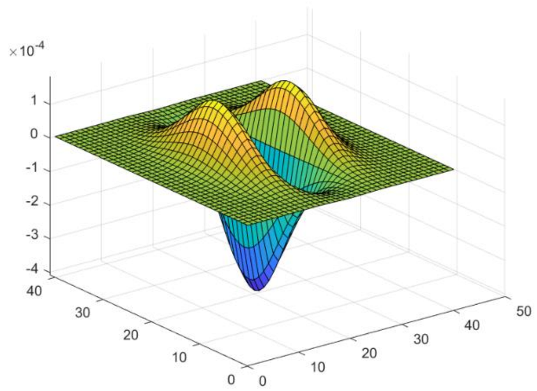
a) I choose sigma=4 and filter size = [6sig+1,6sig+1]. Since we need to rotate the filter in part c, I generate a filter of larger size [10sig+1,10sig+1]. This ensures that the rotated filters in c) are of size [6sig+1,6sig+1] and have valid values everywhere.



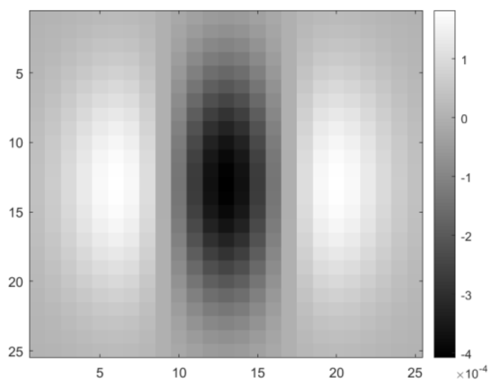


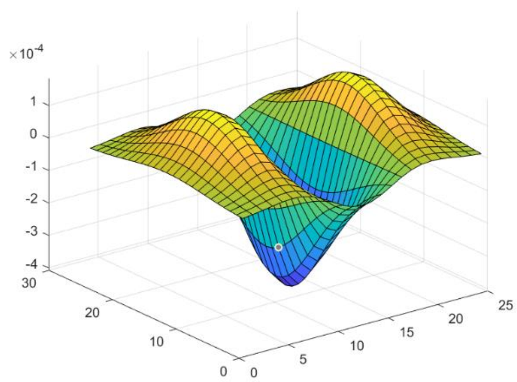
b) Choose sigma_y=6 (>4). Multiply entries of the above filter by a normalized 1D Gaussian in the y-direction with sigma_y



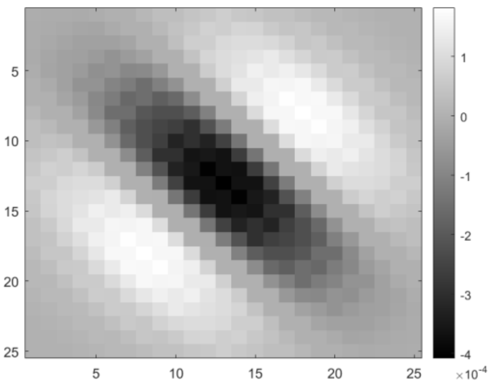


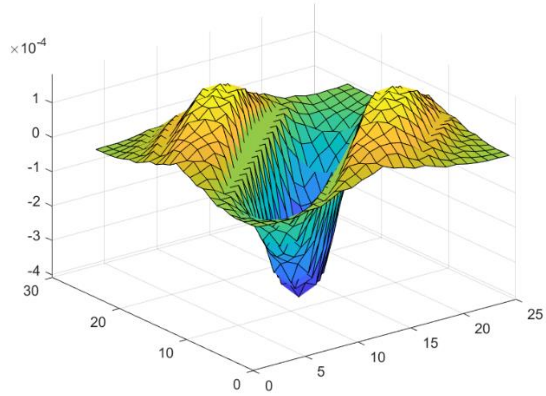
c) In this part all filters have size [6sig+1,6sig+1].
Rotation by 0 degree:



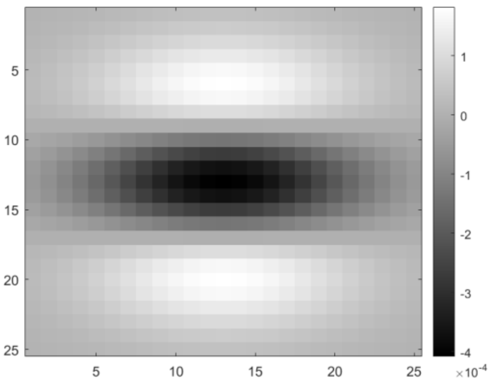


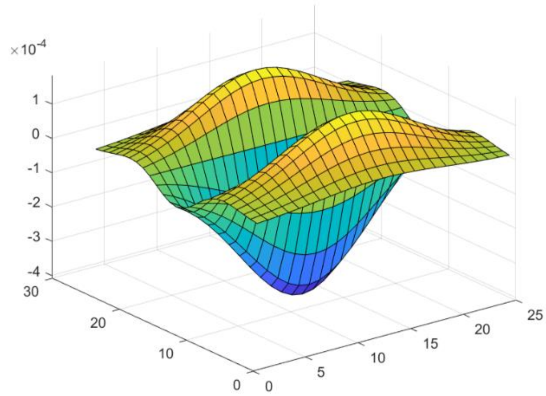
Rotation by 45 degrees:



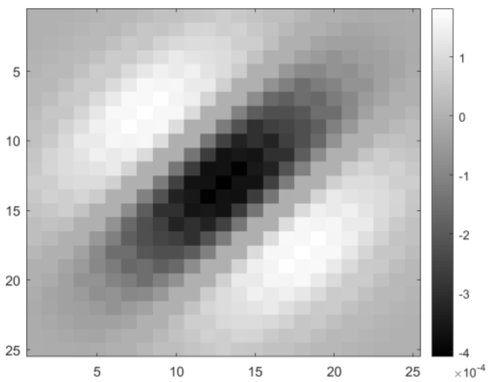


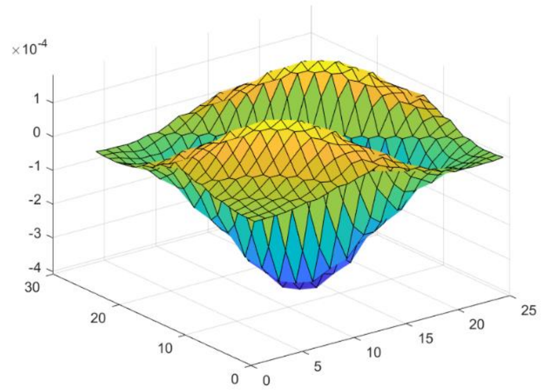
Rotation by 90 degrees:





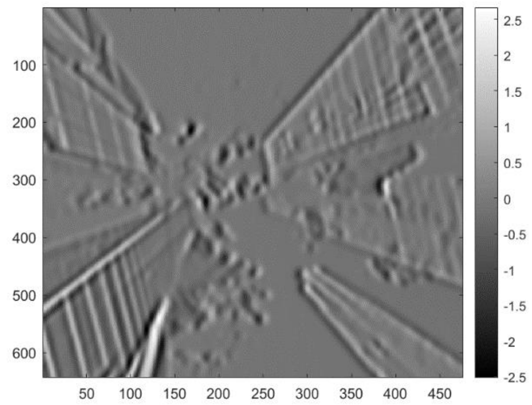
Rotation by 135 degrees:

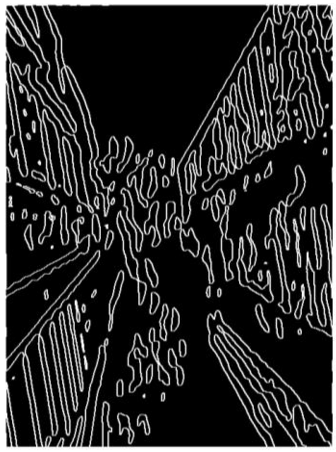




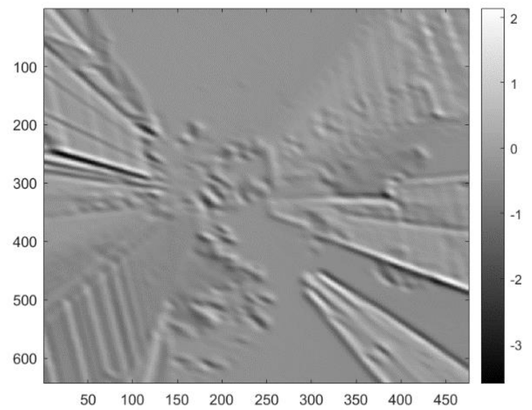
d) Filter the image with 4 rotated filters(0,45,90,135-degrees rotation), and detect edges by finding zero crossings respectively.

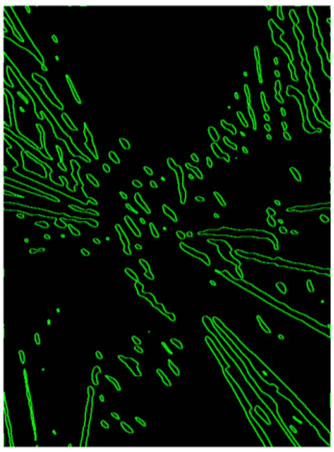
Rotation by 0 degree:



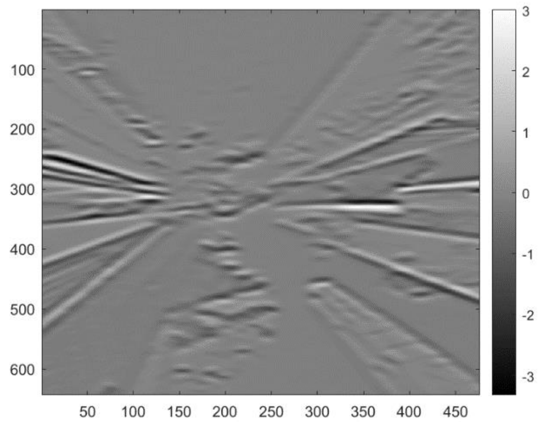


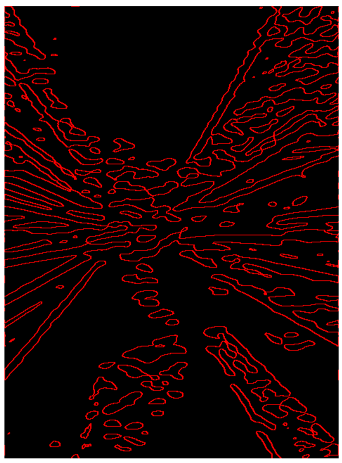
Rotation by 45 degrees:



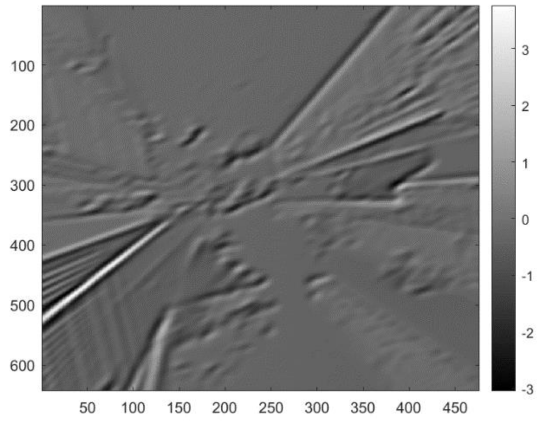


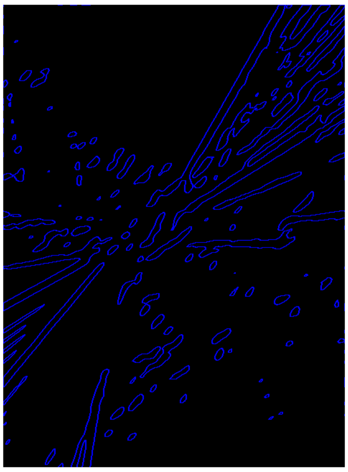
Rotation by 90 degrees:



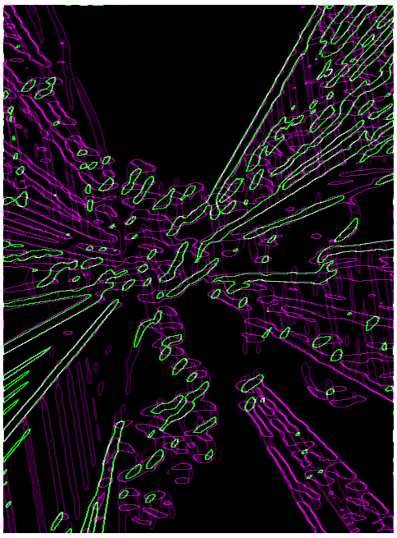


Rotation by 135 degrees:





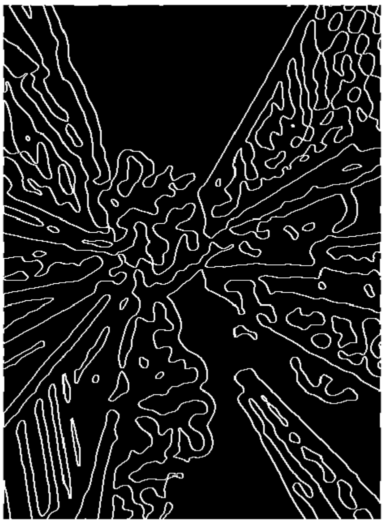
At last, I compose the 4 sets of edges into one image.



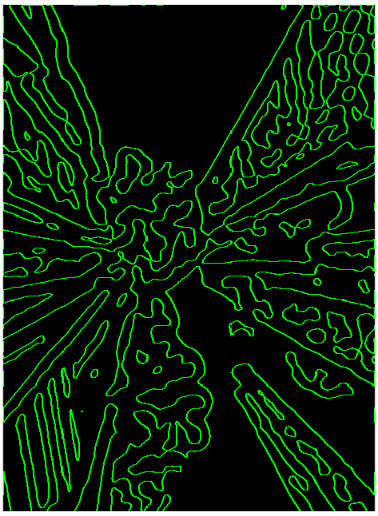
e) We need a Laplacian of a Gaussian filter with sigma=6 and width=40 (40 is the width after rotation, so I generate a larger filter with width 57). Rotate the filter by degree of 0,45,90 and 135, filter the image with the rotated filters, and find zero-crossings respectively.

As the edge images shown, results in d) are quite different, edges in the direction of the filter rotation are detected most since the filter we designed is oriented. In comparation, results in e) are quite similar. This somehow proves that the Laplacian is rotation invariant.

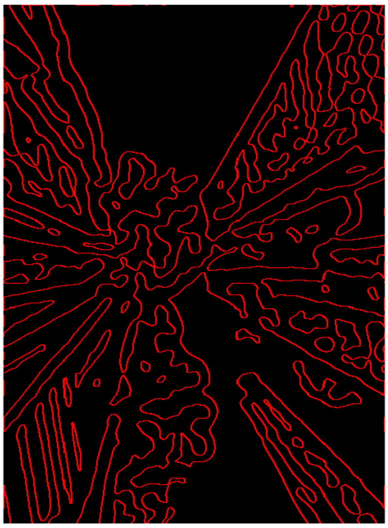
0-degree rotation



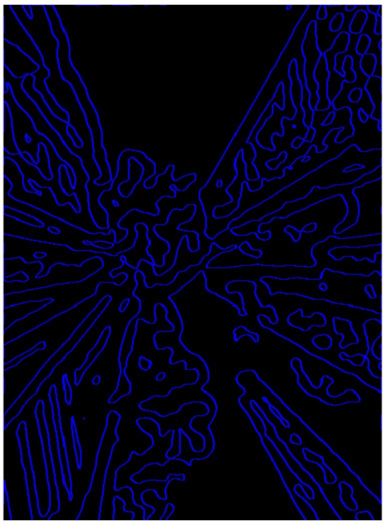
45-degree rotation



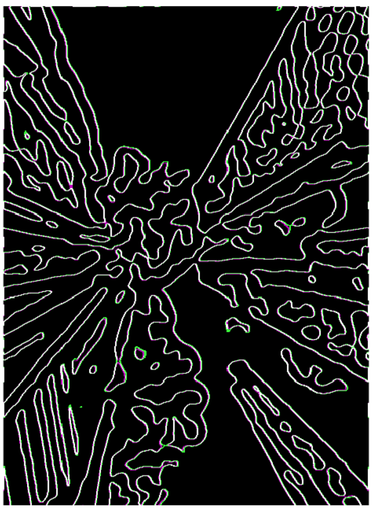
90-degree rotation



135-degree rotation



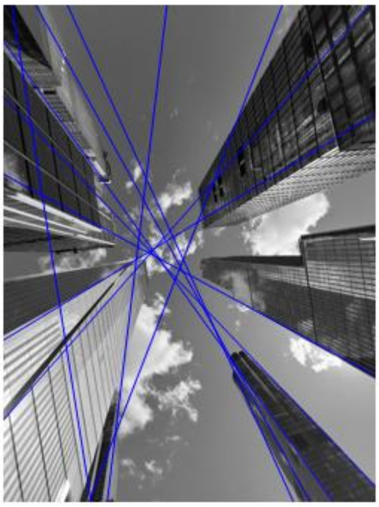
Composite of 4 images above

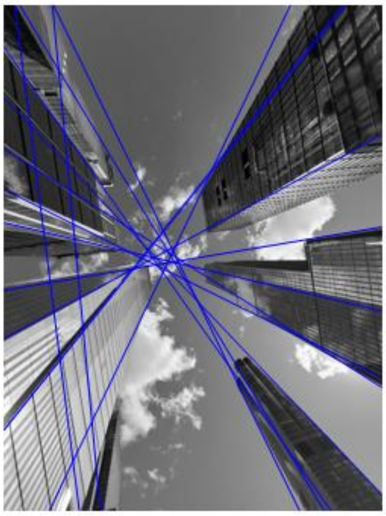
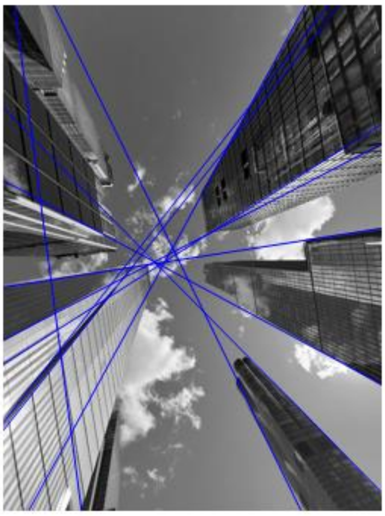


Q3

a) Values of the parameters of my RANSAC are set as follows:
M=500, threshold =3, delta_theta=pi/20, Cmin=150
M=200, threshold =3, delta_theta=pi/20, Cmin=100
M=2000, threshold =2, delta_theta=pi/20, Cmin=100
M=2000, threshold =2, delta_theta=pi/90 Cmin=50
(Outputs are shown from right to left, up to down.)

We can see that basically decreasing distance threshold or delta_theta makes the edges in the consensus more likely to have similar orientation and on the same straight line. And increasing Cmin also makes the conditions for a good line model more strict. This therefore returns less line models, but the lines fit the linear structure in the image better. Increasing M (times of iteration) let us find more line segments.



b) Parameters are set the same as part a. It is obvious that the least square fitting method finds better line models when threshold and delta_theta are bigger. However, if they are set small enough, the two method both work quite well.