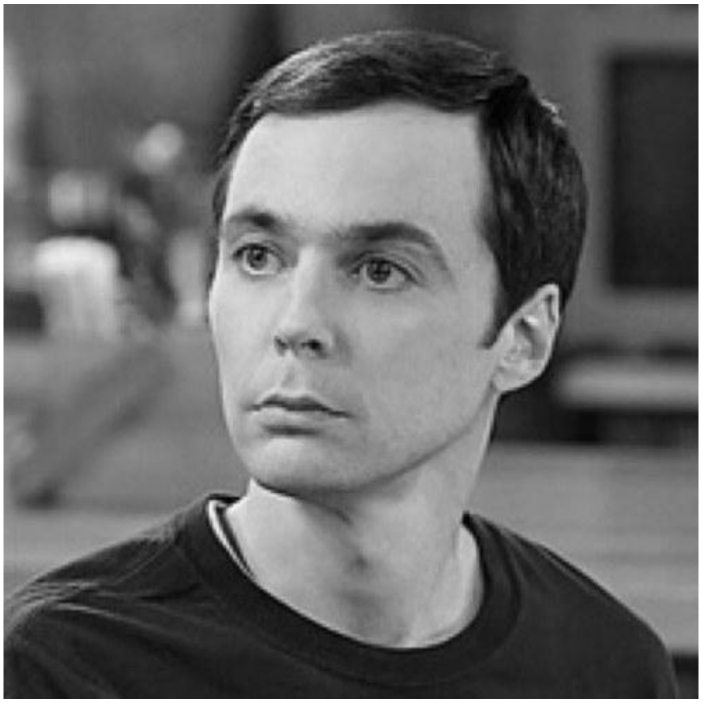
CHANDRASEKAR SWAMINATHAN (swaminathan.42)

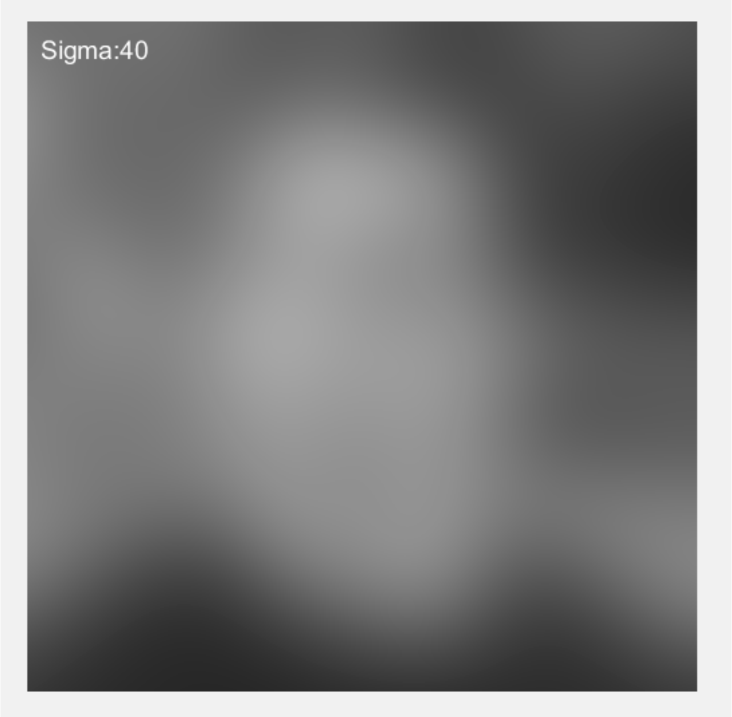
Computer Vision for HCI

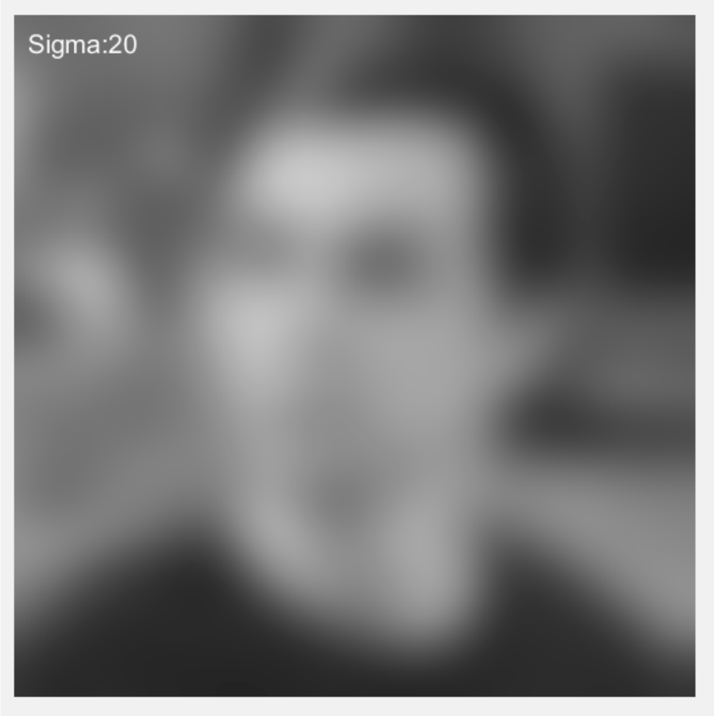
CSE 5524

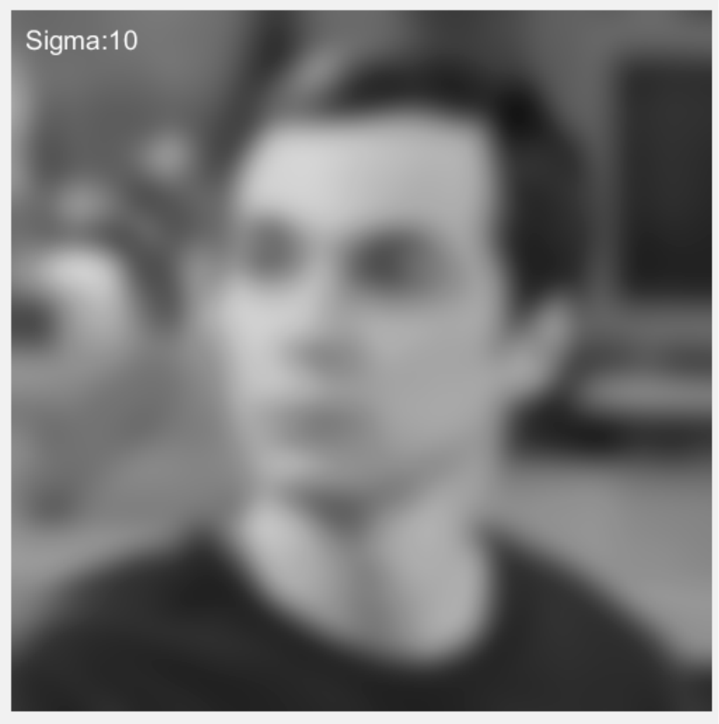
Homework Assignment #2

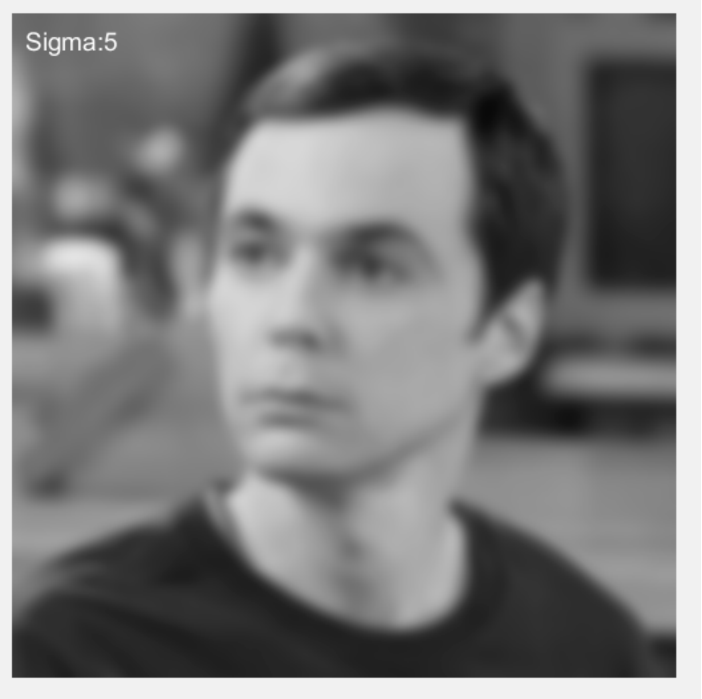
1) Perform Gaussian smoothing on face.jpeg. Try with multiple sigma values, starting with larger values (e.g., from 20 to 0.5). When does the face become recognizable?

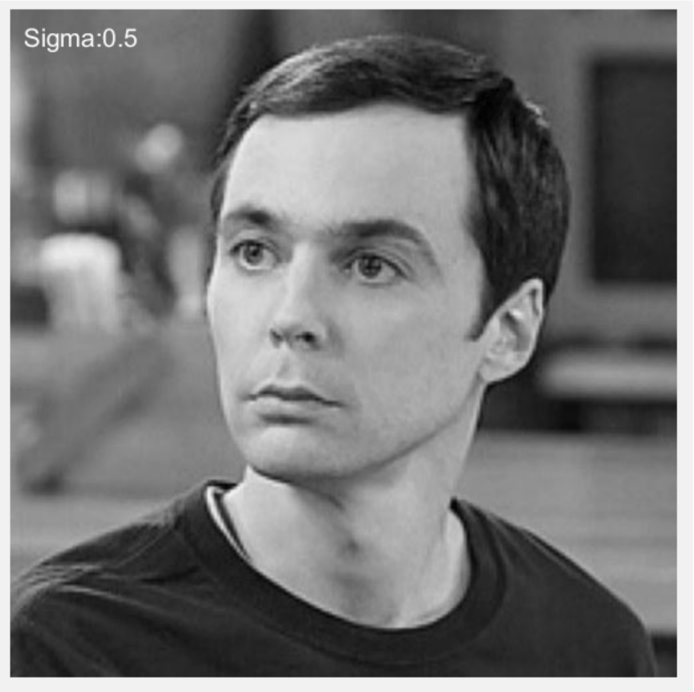






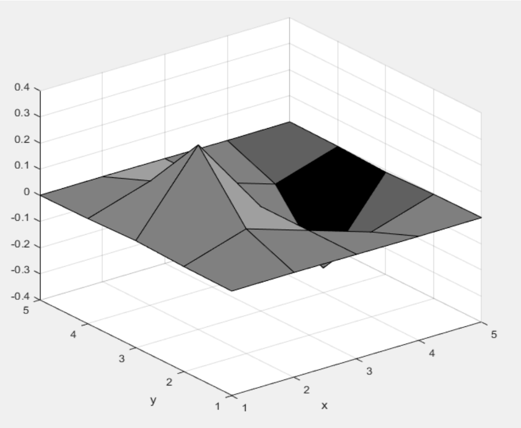




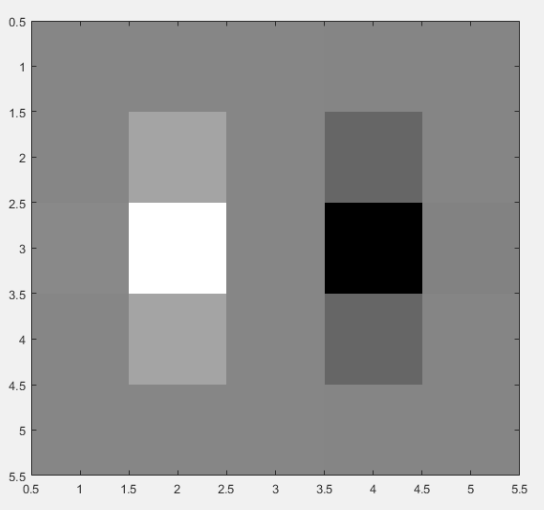


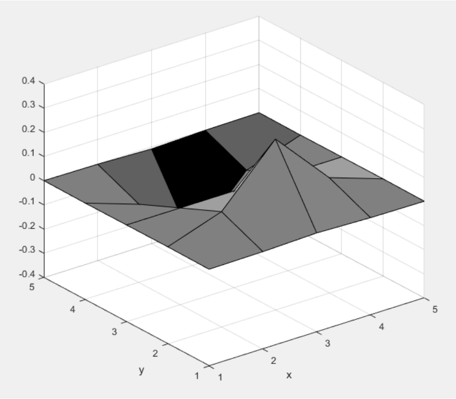
We can see from the above images that as sigma increases the image is blurred to an extent that it is hard to recognize the face in the image. The face became recognizable to one of my friends at the sigma value of 20 and for another friend at the sigma value of 15 (not shown above).

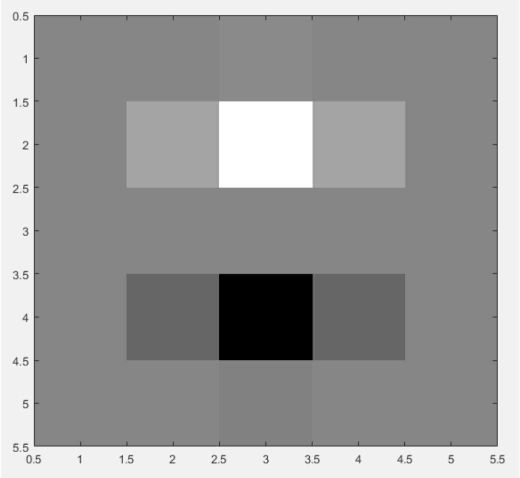
2) Write a MATLAB function to compute and display the 2D Gaussian derivative masks Gx and Gy for a given sigma (see class notes). Note: each mask is a square 2D matrix.



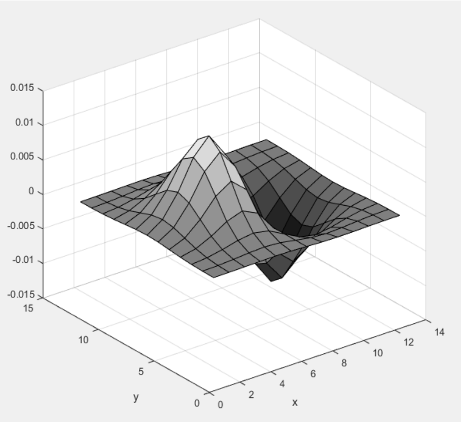
Gx, sigma = 0.6

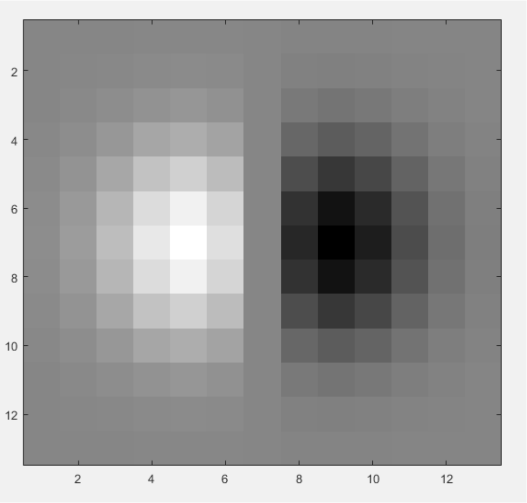




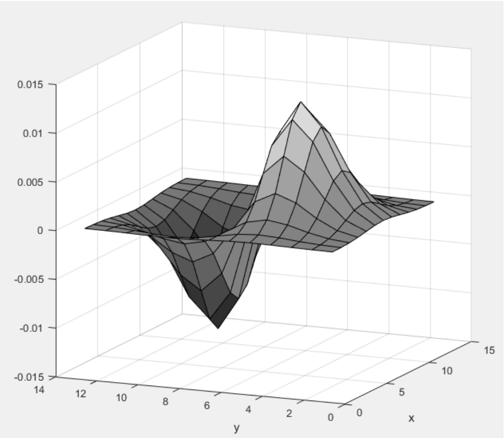


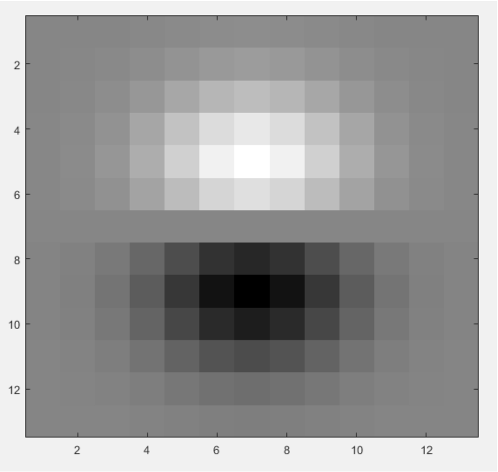
Gy sigma = 0.6





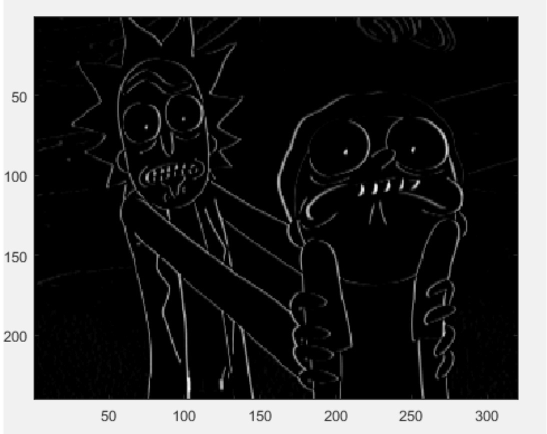
Gx sigma = 2



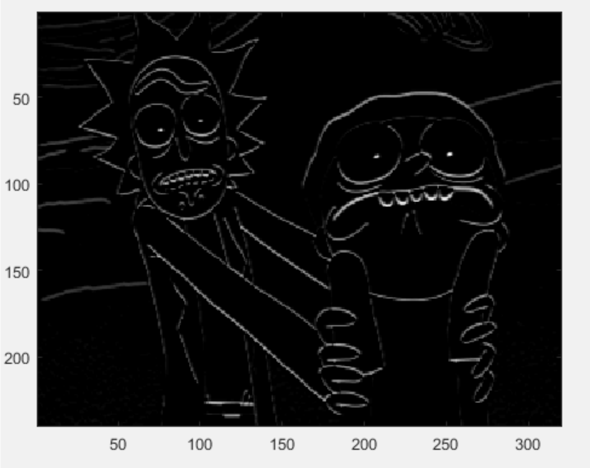


Gy sigma = 2

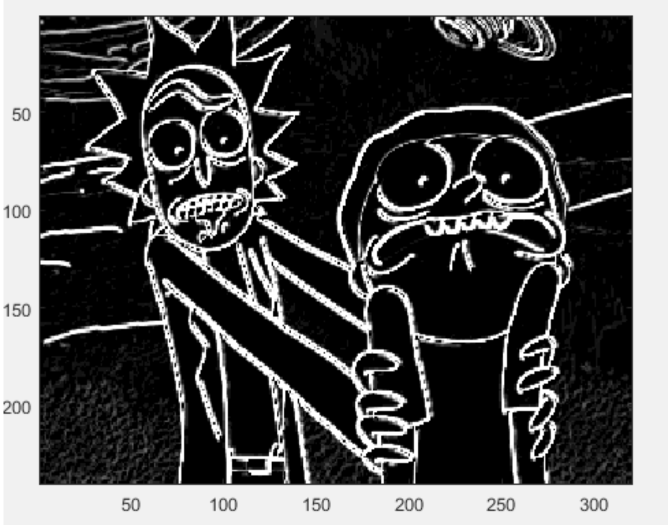
3) Compute and display the gradient magnitude of an image (search the web for an interesting image; convert to grayscale if necessary; make sure to upload the image with code in your submission)



Gx filter, sigma = 0.6, vertical edges are highlighted more than horizontal edges

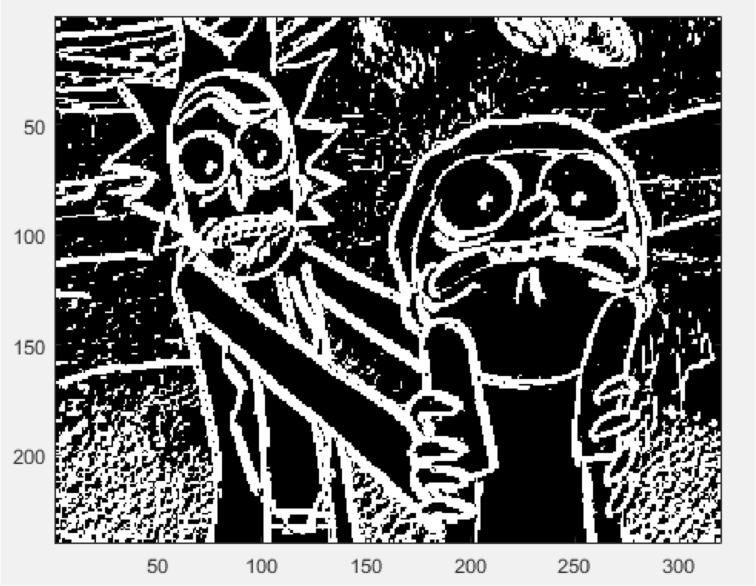


Gy filter, sigma = 0.6 horizontal edges are highlighted more than the vertical edges

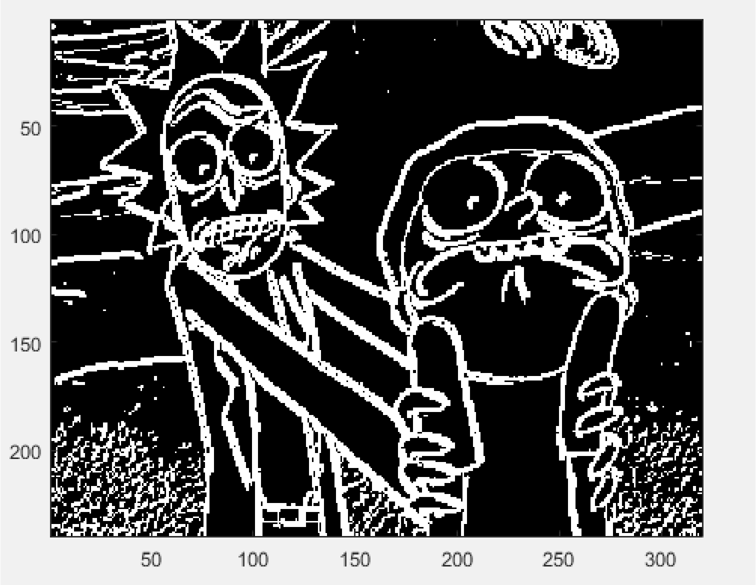


Gradient magnitude combining both the above images

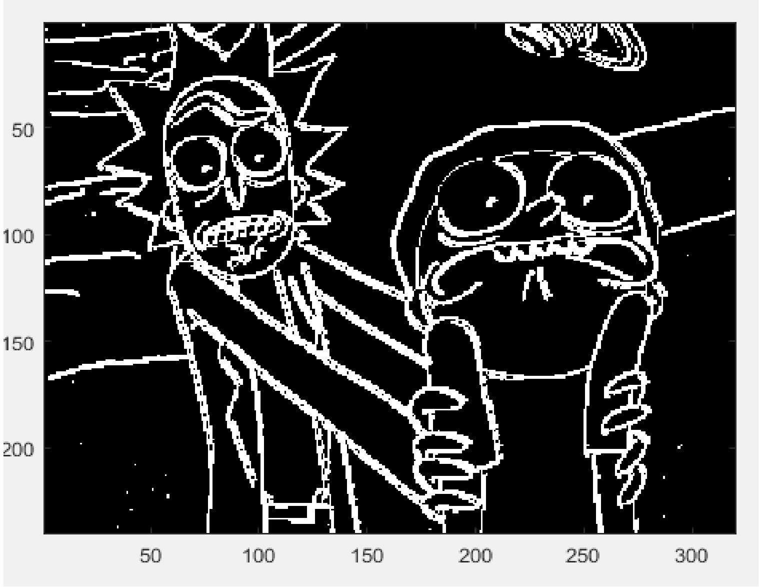
4) Threshold and display the magnitude image with different threshold T levels.



Threshold = 0.5

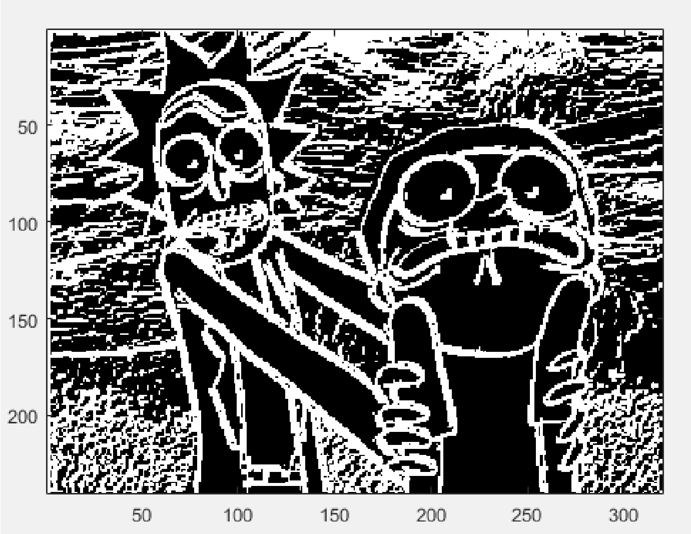


Threshold = 1.5

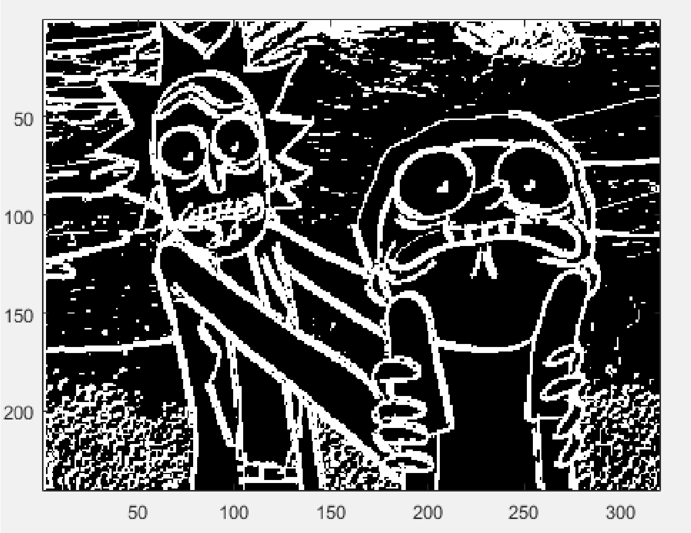


Threshold = 5

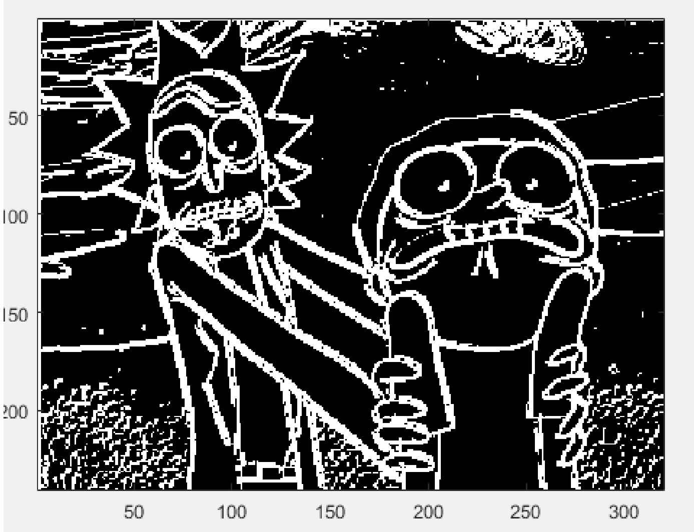
5) Compare the above results with the Sobel masks.



Sobel Threshold = 5

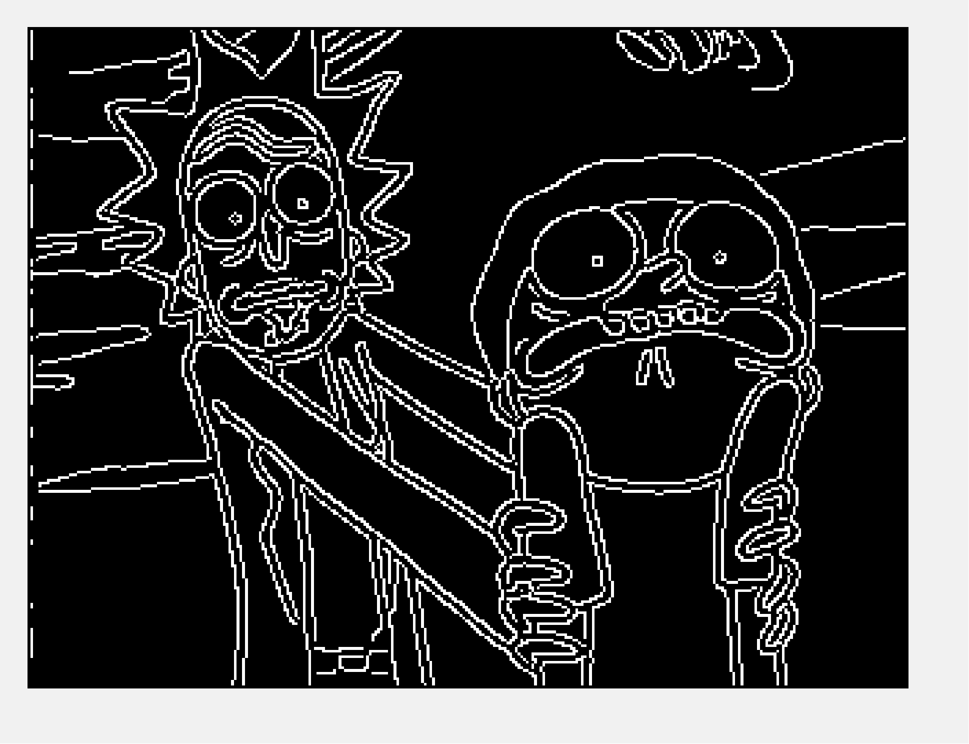


Sobel Threshold = 10



Sobel Threshold = 15

6) Run the MATLAB canny edge detector, edge(Im,'canny'), on your image and display the default results. How does it compare?



Code

HW2.m

```
% Chandrasekar Swaminathan
% swaminathan.42@osu.edu
% CSE5524 - HW2
% 9/6/2016

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Problem 1
% applies gaussian filter with sigma values from 40 to 0.5
face_gaussian_blur;
pause;

% Problem 2
% create gaussian second derivative mask with sigma as 0.6 and 2
[Gx, Gy] = gaussDeriv2D(0.6);
[Gx2, Gy2] = gaussDeriv2D(2);
surf(Gx);xlabel('x');ylabel('y');
pause;
imagesc(Gx);
colormap('gray');
pause;
surf(Gy);xlabel('x');ylabel('y');
pause;
imagesc(Gy);
colormap('gray');
pause;
surf(Gx2);xlabel('x');ylabel('y');
pause;
imagesc(Gx2);
colormap('gray');
pause;
surf(Gy2);xlabel('x');ylabel('y');
pause;
imagesc(Gy2);
colormap('gray');
pause;


% Problem 3
% Computing gradient magnitude
img = imread('rickandmorty.jpg');
img = rgb2gray(img);
gxIm = imfilter(img, Gx, 'replicate');
gyIm = imfilter(img, Gy, 'replicate');
magIm = sqrt(double(gxIm.^2 + gyIm.^2));
```

```
imagesc(gxIm);
colormap('gray');
pause;
imagesc(gyIm);
colormap('gray');
pause;
imagesc(magIm);
colormap('gray');
pause;

% Problem 4
% applying a threshold on the gradient magnitude
T = 0.5;
tIm = magIm > T;
imagesc(tIm);
pause; % very low threshold
T = 1.5;
tIm = magIm > T;
imagesc(tIm); %slightly better
T = 5;
tIm = magIm > T;
imagesc(tIm); %a lot better


% Problem 5
% Using sobel mask to detect edges
sobel_masks(img, 5);
pause;
sobel_masks(img, 10);
pause;
sobel_masks(img, 15.95);
pause;

% Problem 6
cannyImg = edge(img, 'canny');
imshow(cannyImg);
```

sobel_masks.m

```
function [] = sobel_masks (Im, T)
    Fx = -fspecial('sobel')';
    fxIm = imfilter(Im, Fx);
    Fy = -fspecial('sobel');
    fyIm = imfilter(Im, Fy);
    magIm = sqrt(double(fxIm.^2 + fyIm.^2));
    tIm = magIm > T;
    imagesc(tIm);
    colormap('gray');
    axis('image');
end
```

gaussDeriv2D.m

```
function [Gx, Gy] = gaussDeriv2D (sigma)
    maskSize = 2*ceil(3*sigma) +1;
    twoSigmaSquared = 2 * sigma^2;
    twoPiSigmaQuad = 2 * pi * sigma^4 ;
    Gx = zeros (maskSize,maskSize);
    Gy = zeros (maskSize,maskSize);
    x0 = floor(maskSize/2) + 1;
    y0 = floor(maskSize/2) + 1;
    for n=1:maskSize
        for m=1:maskSize
            exponentTerm = exp(-((n-x0)*(n-x0) + (m-y0)*(m-y0))/twoSigmaSquared);
            Gx(n,m)= -(m-y0) * exponentTerm/twoPiSigmaQuad;
            Gy(n,m)= -(n-x0) * exponentTerm/twoPiSigmaQuad;
        end
    end
end
```

face_gaussian_blur.m

```
function [] = face_gaussian_blur()
    sigma = 40;
    decrement_step =5;
    faceIm = double (imread('face.jpeg'));
    gIm = faceIm;
    while sigma >= 0.5
        G = fspecial ('gaussian', 2 * ceil(3*sigma) + 1, sigma);
        gIm = imfilter(faceIm, G, 'replicate');
        imshow(gIm/255);
        text(10, 20, strcat('Sigma: ', num2str(sigma)), 'Color', 'white', 'FontSize', 14);
        pause;
        if sigma <= decrement_step
            sigma=sigma-0.5;
        else
            sigma=sigma-decrement_step;
        end
    end
end
```