CSE 5524 HW2

**Problem 1**

%% Problem 1

clear;

clc;

sigma=20.0; % use different values

G = fspecial('gaussian', 2\*ceil(3\*sigma)+1, sigma);

faceIm=double(imread('harry.png'));

gIm = imfilter(faceIm, G, 'replicate');

imshow(gIm/255); % double images need range of 0-1

imwrite(uint8(gIm), 'gIm.bmp');

A close up of a screen

Description automatically generatedA screenshot of a person

Description automatically generatedA screenshot of a person

Description automatically generated

Sigma = 20 sigma = 10 sigma = 5

A screenshot of a person

Description automatically generatedA screenshot of a person

Description automatically generatedA screenshot of a person

Description automatically generated

Sigma = 2.5 sigma = 1.25 sigma = 0.625

As you can see from the six pictures above, the face is very vague with large value of sigma and becomes more and more clear as the value of sigma goes down. The picture becomes recognizable when sigma reaches 2.5.

**Problem 2**

%% Problem 2

clc;

clear;

sigma = 3;

[Gx,Gy] = gaussDeriv2D(sigma);

[X,Y] = meshgrid(-ceil(3\*sigma):ceil(3\*sigma));

surf(X,Y,Gx);

xlabel('x');

ylabel('y');

pause();

surf(X,Y,Gy);

xlabel('x');

ylabel('y');

function [Gx, Gy] = gaussDeriv2D(sigma)

% this function samples 3 std from the Gaussian

Gx = ones(2\*ceil(3\*sigma) + 1,2\*ceil(3\*sigma) + 1); % mask size

Gy = ones(2\*ceil(3\*sigma) + 1,2\*ceil(3\*sigma) + 1); % mask size

range = [-ceil(3\*sigma): ceil(3\*sigma)];

for x = 1:2\*ceil(3\*sigma)+1

for y = 1:2\*ceil(3\*sigma)+1

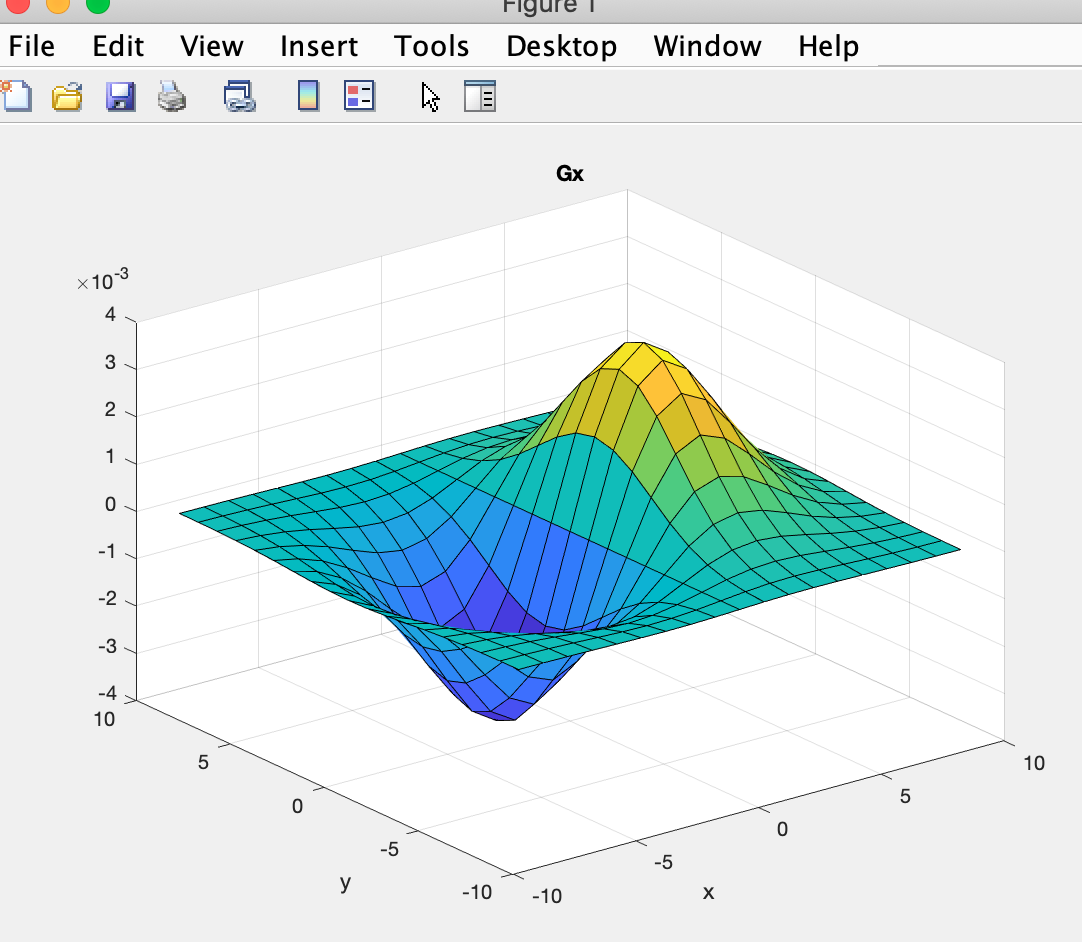
Gx(x,y) = range(y)\*exp(-(range(x)^2+range(y)^2)/(2\*sigma^2))/(2\*pi\*sigma^4);

Gy(x,y) = range(x)\*exp(-(range(x)^2+range(y)^2)/(2\*sigma^2))/(2\*pi\*sigma^4);

end

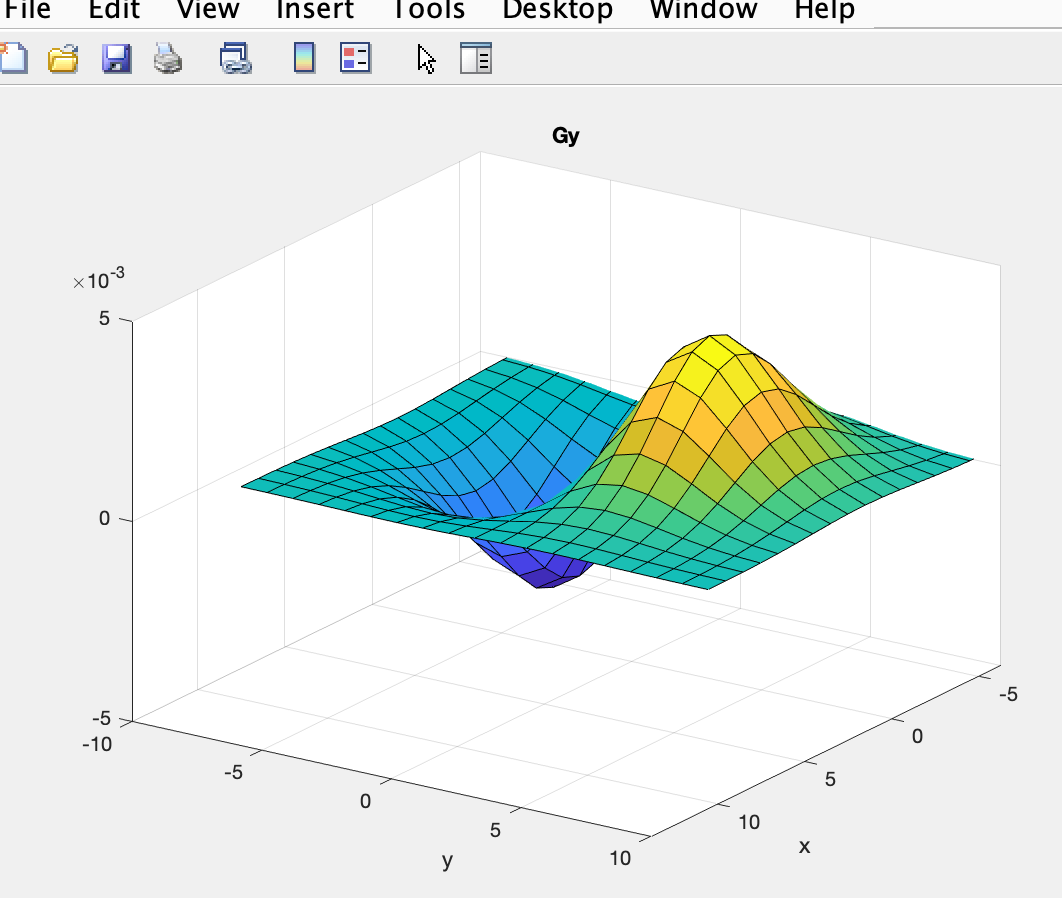
end

end

A screenshot of a cell phone

Description automatically generated

Surfaces for Gx

A screenshot of a cell phone

Description automatically generated

Surface for Gy

**Problem 3**

%% Problem 3

clear;

clc;

sigma = 3;

[Gx,Gy] = gaussDeriv2D(sigma);

I = im2double(imread('touma.jpg'));

imshow(I);

pause;

gxIm = imfilter(I, Gx, 'replicate');

gyIm = imfilter(I ,Gy, 'replicate');

magIm = sqrt(gxIm.^2 + gyIm.^2);

imagesc(gxIm);

pause;

imagesc(gyIm);

pause;

imagesc(magIm);

A picture containing building, computer, desk, sitting

Description automatically generatedA picture containing dark, standing, sitting, door

Description automatically generated

Original image filtered by Gx

A picture containing dark, sitting, night, bed

Description automatically generatedA person in a dark room

Description automatically generated

Filtered by Gy combination of Gx and Gy

**Problem 4**

%% Problem 4

T = 2;

tIm = magIm > T;

imagesc(tIm);

A close up of a logo

Description automatically generated A picture containing food

Description automatically generated

T = 2 t = 0.2

T = 0.02 t = 0.002

**Problem 5**

%% Problem 5

T = 2;

Fx = -fspecial('sobel')';

fxIm = imfilter(I,Fx);

Fy = -fspecial('sobel');

fyIm = imfilter(I,Fy);

magIm = sqrt(fxIm.^2 + fyIm.^2);

tIm = magIm > T;

imagesc(tIm);

A picture containing dark, lit, table, holding

Description automatically generated 

T = 2 t = 0.2

A picture containing rain

Description automatically generated A picture containing computer, rain, large, many

Description automatically generated

T = 0.02 t = 0.002

As you can see from the images above, Sobel mask did a better job at a relatively high threshold value. However, under the low threshold values, Sobel mask start to lose counter compared with Gaussian’s method.

**Problem 6**

%% Problem 6

grayI = rgb2gray(I);

edge(grayI,'canny');

A close up of a logo

Description automatically generated

It is obvious that canny edge detection did the best job compared with the two methods above.

**Discussion:** The result of problem 2 shows the surface of the partial derivative of 2D Gaussian. Here we add an extra positive sign to both to make slope increase in the axis direction.

For problem 3, the Gx filter catches the edge in the x-direction, where you can see the boundary between face and hair; the Gy filter catches the edge in the y-direction, where you can see eyes and shoulder are captured. When combined together, we can see a rough counter of Touma Kazusa.

For problem 4, I tried 4 different T values. At first the image is black since no pixel value is higher than that threshold. When the value decreases, we are able to see some white pixels and the threshold becomes too small, most part of the images become white and it is hard to tell the detail of the images.

For problem 5, after applying Sobel mask to the image, it shows a better result at high value of T, which means the pixel value after applying the Sobel mask is higher compared with Gaussian mask.

For problem 6, canny edge detection did the best job. The reason for this dues to no-maximal suppression where we eliminate the pixel that is not dominant in the neighbor and hysteresis thresholding helps to preserve the weak candidate linked to the strong candidate.