CSE 5524 HW2 Report

Author: Yi Zhao, zhao.2175@osu.edu

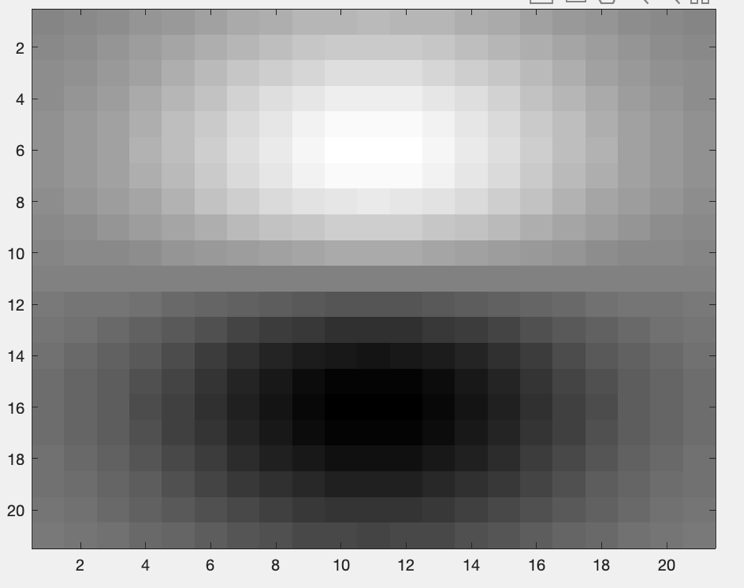
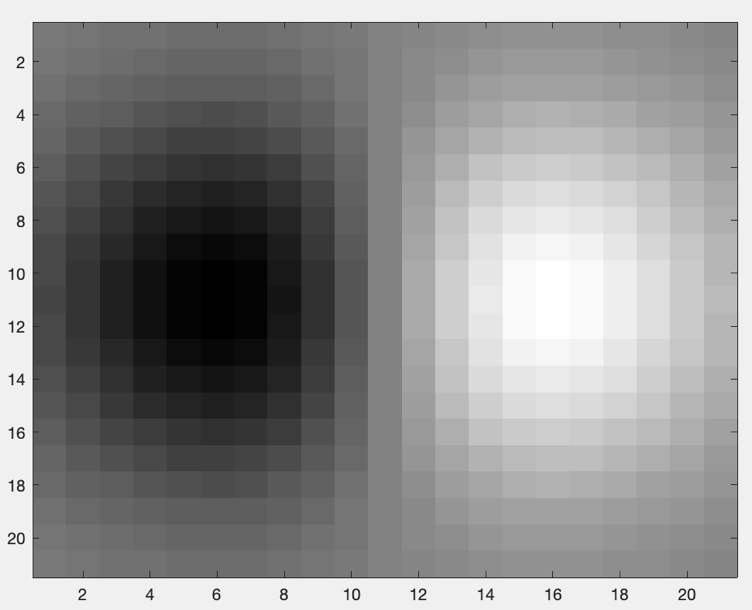
1. The face becomes recognizable when sigma changed to 12. When sigma= 20, the image is blurred significantly, and people usually can tell that “this is a human”. As image becomes clear, people can start guessing who the person in image is, as they can see more details of this person’s face. The smaller sigma is, the clear the image is.

Original: sigma=12: sigma=20



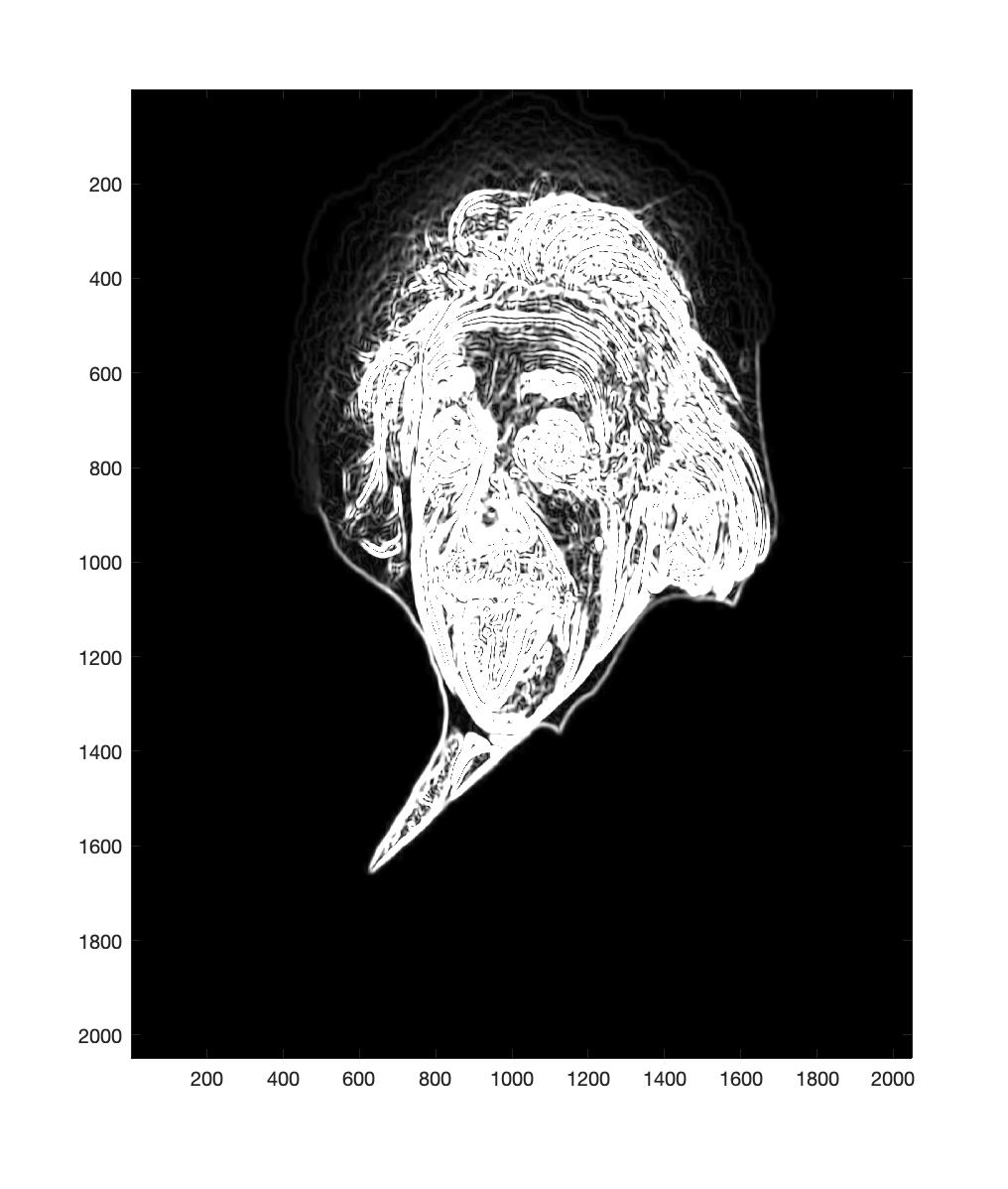
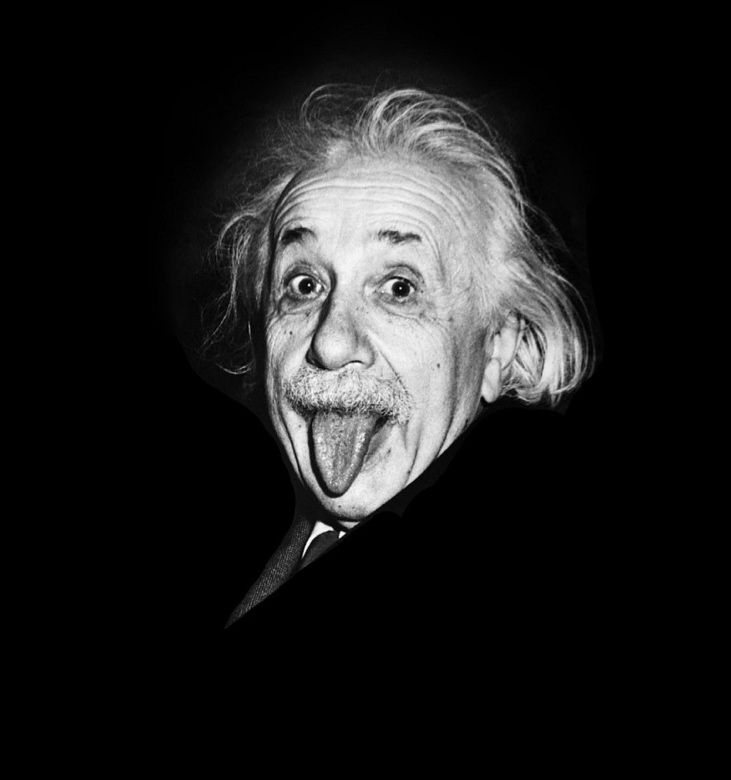
1. Just follow the 2D gaussian formula for x and y.

Gx filter: Gy filter:

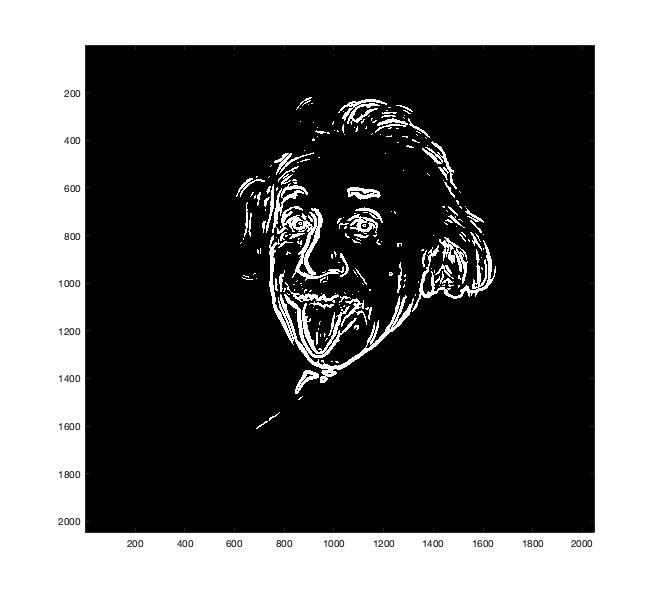


1. The image is a portrait of a famous scientist, Einstein. By using the mask created above using sigma = 5, the edges are obtained line by line. The smaller sigma tends to contain more details of edges, while larger sigma will deprecate some edges and only the essential ones remain. In the result below, the edges of face and hairs remain, and other details on face are missing.

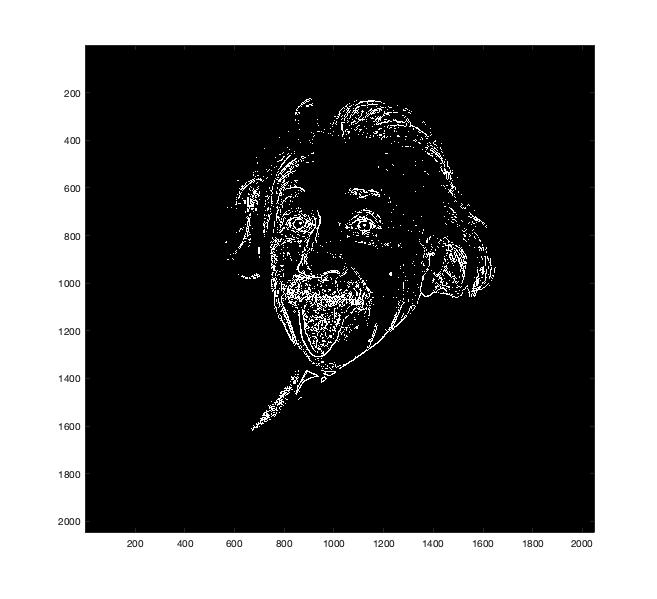
Origin(left) vs. Gaussian 2D with sigma=5(right):



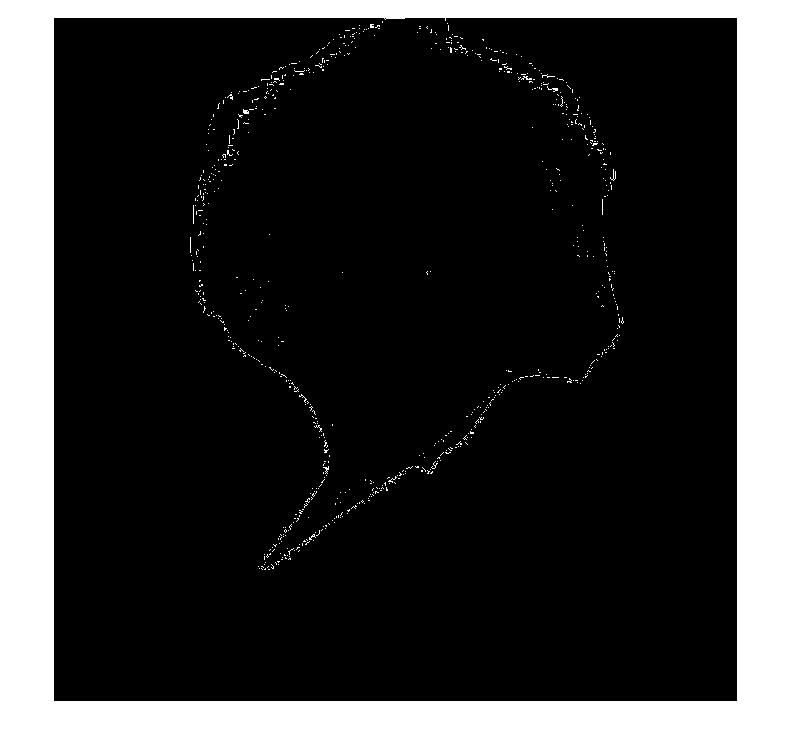
1. The threshold filters the edges after image was filtered through gaussian 2d filter. As T becomes larger, more details (edges) are removed from picture. If T goes very large, then very little edges remain on the picture. If T goes very small, then too many edges show up. Here I choose T=3 because I think the edges are captured as expected, and the outline is representative enough.



1. The sobel masks contains too much edges whatever it’s important or not, when using the same T value as used in previous questions. I increased T value to 100 and then the edges become clear, and small unnecessary edges are eliminated. Comparing to gaussian2d filter, its edges are skinny, thinner (because I used a large sigma value).



1. The Canny detector did pretty well on edge selection. Only the outlines remain and all inside small edges are not shown in the result. The hair is the most difficult part in this image, and Canny precisely obtained the outlines of the head, without being lost in edges of hairs or other facial details. Two threshold value helps to extract most obvious edges and also deprecate insignificant details.



% Author: Yi Zhao, zhao.2175@osu.edu

% Professor: Jim Davis

% Class: CSE 5524, MW 12:45PM

% Date: 09/06/2019

%%%%%%%%%%%%%%%%%%%% HW 2 %%%%%%%%%%%%%%%%%%%%

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Problem 1

sigma=20; % use different values

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

faceIm=double(imread('./data/affleck\_gray.png'));

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

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

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

pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Problem 2 (Function definition locates at the end)

[a,b] = gaussDeriv2D(5); % change sigma when needed

img\_gx = imagesc(a);

colormap('gray');

pause;

img\_gy = imagesc(b);

colormap('gray');

pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Problem 3

[Gx,Gy] = gaussDeriv2D(5);

Im = double(imread('./data/img.jpg'));

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

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

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

imagesc(magIm);

pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Problem 4

T = 3;

tIm = magIm > T;

imagesc(tIm);

pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Problem 5

T = 100;

Fx = -fspecial('sobel')';

fxIm = imfilter(Im,Fx);

Fy = -fspecial('sobel');

fyIm = imfilter(Im,Fy);

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

tIm = magIm > T;

imagesc(tIm);

pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Problem 6

Im = rgb2gray(Im);

edge(Im, 'canny')

pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Problem 2 Function definition

function [Gx, Gy] = gaussDeriv2D(sigma)

length = 2 \* ceil(sigma \* 2) + 1;

for r = 1:length

for c = 1:length

y = -r + ceil(2\*sigma) + 1;

x = c - ceil(2\*sigma) - 1;

Gx(r,c) = x \* exp(-1 \* (x^2 + y^2)/(2 \* sigma.^2)) / (2 \* pi \* sigma^4);

Gy(r,c) = y \* exp(-1 \* (x^2 + y^2)/(2 \* sigma.^2)) / (2 \* pi \* sigma^4);

end

end

end