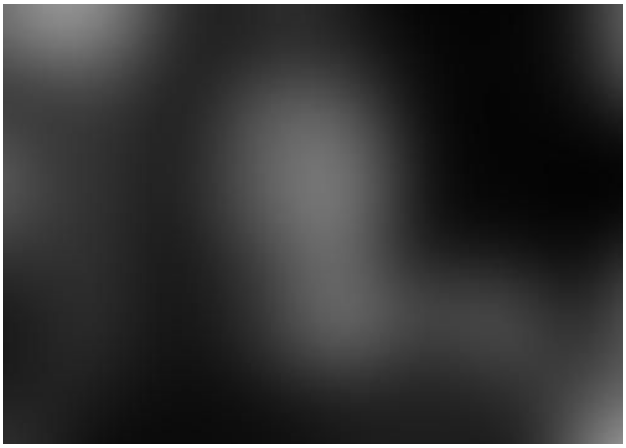
Computer Vision for HCI — Abhinav Mahalingam (.15)

CSE 5524

Homework #2

1. Perform Gaussian smoothing on stark.jpg . Try with multiple sigma values, starting with larger values (e.g., from 20 to .5). When does the face become recognizable to your friends?

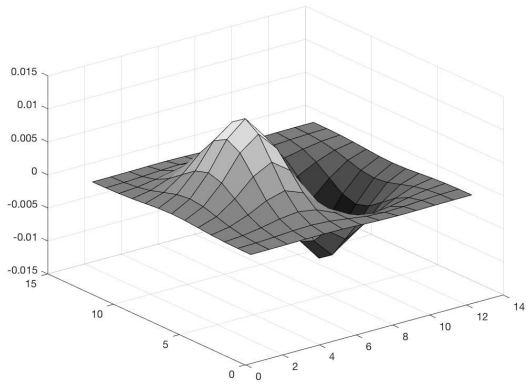


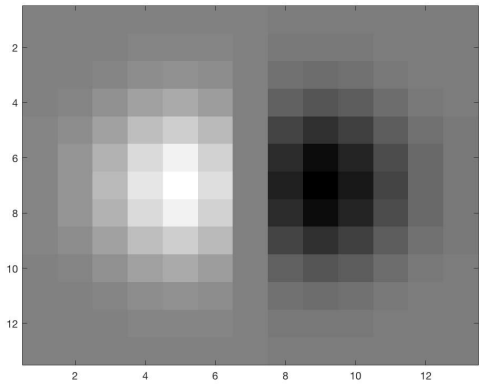
Sigma : 30

Sigma: 7.5

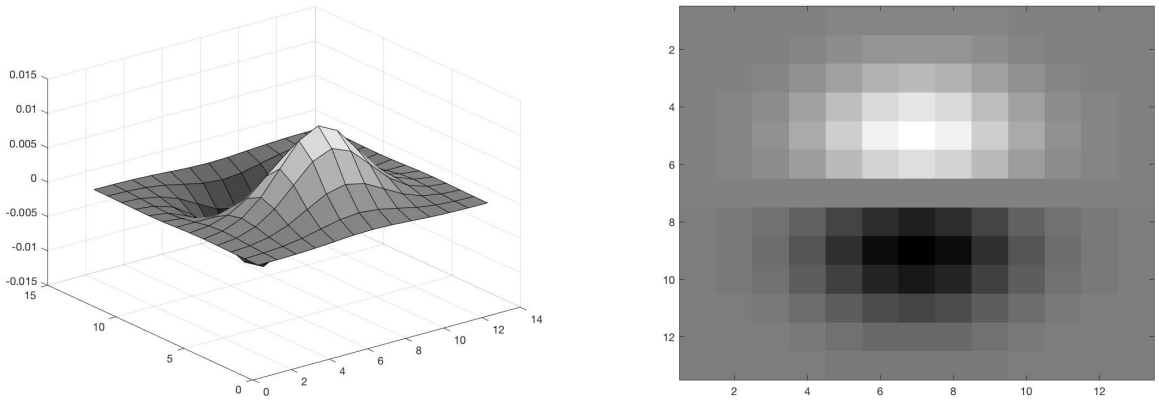
From the images above, it can be seen that the smoothing/blurring is proportional to the value of sigma. The face became recognizable to one of my friend at 17.5 and the other at 10.

2. Write a MATLAB function to compute and display the 2D Gaussian derivative masks Gx and Gy for a given sigma. Sample the Gaussian derivative/gradient (2D) equation directly (see class notes) at the appropriate x,y locations. Note: each mask is a square 2D matrix. [3 pts]





Gx at sigma = 2

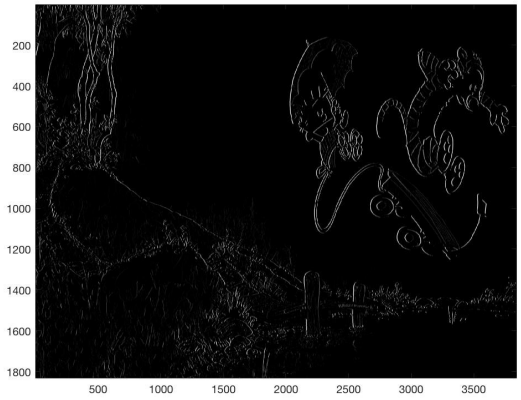


Gy at sigma = 2

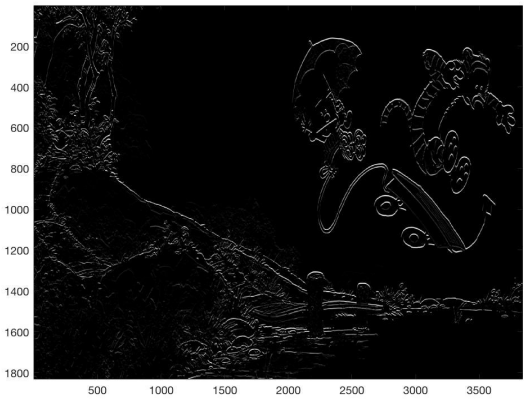
The surface plot and the 2-D plot of the Gaussian derivative masks Gx and Gy for sigma 2 is shown above.

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 your code in the Carmen submission). [2 pts]

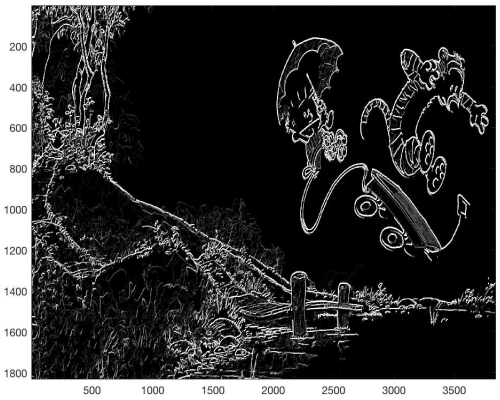
Original Image



Gradient along X, sigma = 2



Gradient along Y, sigma = 2



Gradient Magnitude, sigma = 2

4. Threshold and display the magnitude image with different threshold T levels. [2 pts]

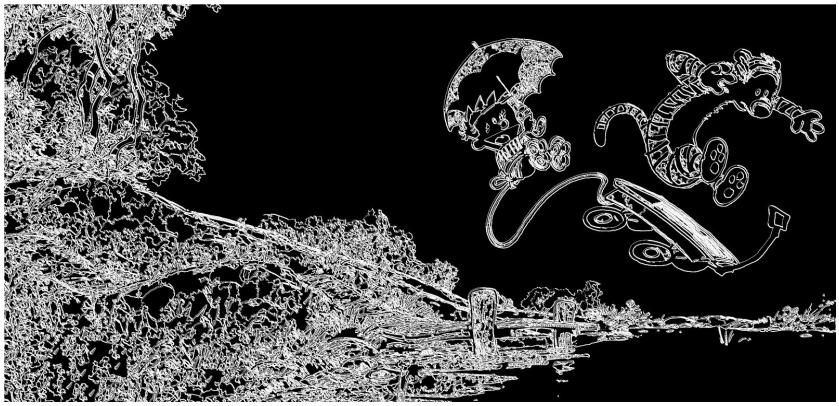
Threshold = 1

Threshold = 7

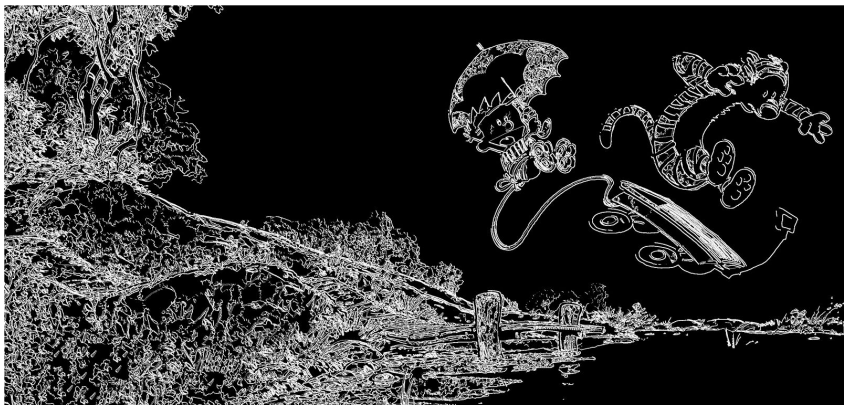
Threshold = 15

The thresholded image is completely black for threshold values of 16 and above. For values from 15 to 10, it provides good results with just the significant edges shown. As it goes lower, it picks up a lot of noise and minor details and changes.

5. Compare the above results with the Sobel masks. [2 pts]



Threshold = 1

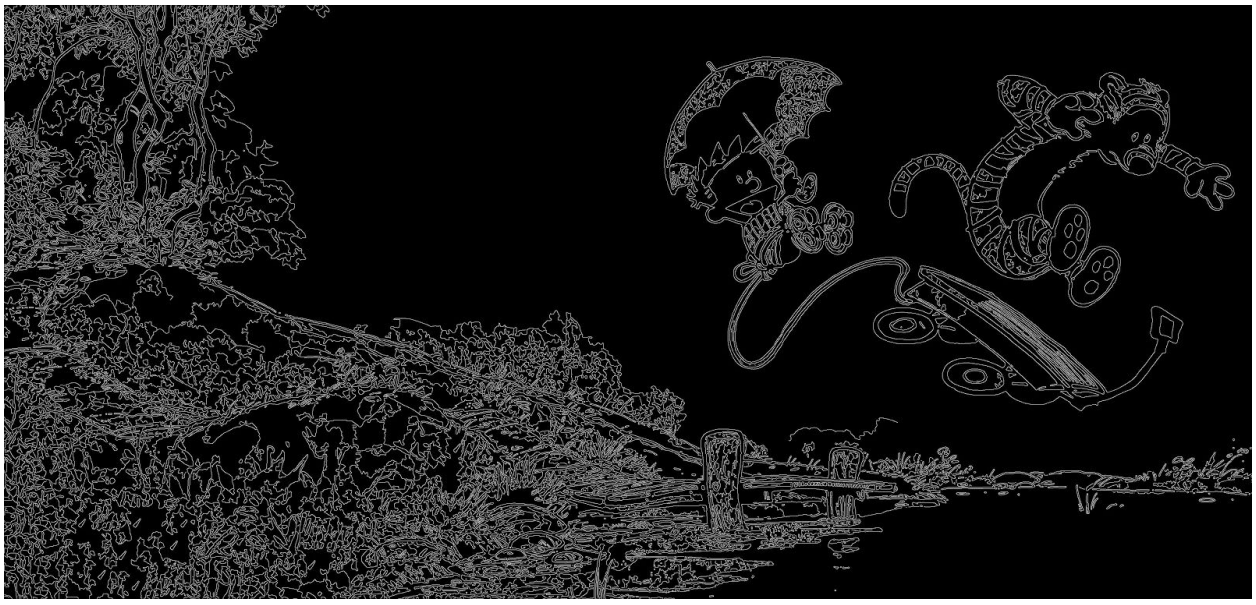


Threshold = 7

Threshold = 15

We can see that the pattern observed in the previous results can be seen here too. The thresholded image is completely black for threshold values greater than 16 and as the value decreases, it picks up increasing amount of details. Compared to the gaussian filter, we can also see that the final image with Sobel masks is more cluttered and doesn't provide clear edges.

6. Run the MATLAB canny edge detector, edge(Im,'canny'), on your image and display the default results. How does it compare? [2 pts]



The canny edge detector performs poorer than the gaussian filter and on par with the Sobel mask.

```
% Abhinav Mahalingam
% CSE5524 - HW1
% 8/28/2017
HW2.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Problem 1
sigma_vals=[40.0,30.0,20.0,17.5,15.0,12.5,10.0,7.5,5.0,3.0,2.0];
faceIm=double(imread('stark.jpg'));
for i=1:length(sigma_vals);
   sigma=sigma_vals(i);
   G = fspecial('gaussian', 2*ceil(3*sigma)+1, sigma);
   gIm = imfilter(faceIm, G, 'replicate');
   imshow(gIm/255);  % double images need range of 0-1
   fname = sprintf('results/starkFilt%d.jpg', i);
   imwrite(uint8(gIm), fname);
   pause;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Problem 2
sigma=2;
colormap('gray');
[Gx, Gy] = gaussDeriv2D(sigma);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Problem 3
Im=(imread('CandH.png'));
grayIm = rgb2gray(Im);
gxIm = imfilter(grayIm, Gx, 'replicate');
imagesc(gxIm);
pause;
gyIm = imfilter(grayIm, Gy, 'replicate');
imagesc(gyIm);
pause;
magIm = sqrt(double(gxIm.^2 + gyIm.^2));
imagesc(magIm);
pause;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Problem 4
thres_vals=[16,15,10,7,4,2,1];
for i=1:length(thres_vals);
   T=thres_vals(i);
   tIm = magIm > T;
   imagesc(tIm);
   fname = sprintf('results/threshold%d.jpg', T);
   imwrite(tIm, fname);
   pause;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Problem 5
Fx = -fspecial('sobel')';
fxIm = imfilter(grayIm,Fx);
Fy = -fspecial('sobel');
fyIm = imfilter(grayIm,Fy);
magIm = sqrt(double(fxIm.^2 + fyIm.^2));
for i=1:length(thres_vals);
   T=thres_vals(i);
   tIm = magIm > T;
   imagesc(tIm);
   fname = sprintf('results/sobelThreshold%d.jpg', T);
   imwrite(tIm, fname);
   pause;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Problem 6
cannyImg = edge(grayIm,'canny');
imagesc(cannyImg);
imwrite(cannyImg, 'results/canny.jpg');
```

gaussDeriv2D.m

```
function [Gx, Gy] = gaussDeriv2D(sigma)
   maskSize = 2*ceil(3*sigma)+1;
   Gx = zeros(maskSize,maskSize);
   xo = 1 + ceil(3*sigma);yo = xo;
   const1 = (2*pi*sigma^4);
   const2 = (2*sigma^2);
   for x=1:maskSize;
      for y=1:maskSize;
         Gx(y,x) = -((x - xo)/const1)*exp(-((x-xo)^2 + (y-yo)^2)/const2);
      end
   end
   Gy = Gx';
   surf(Gx);
   pause;
   imagesc(Gx);
   pause;
   surf(Gy);
   pause;
   imagesc(Gy);
   pause;
end
```