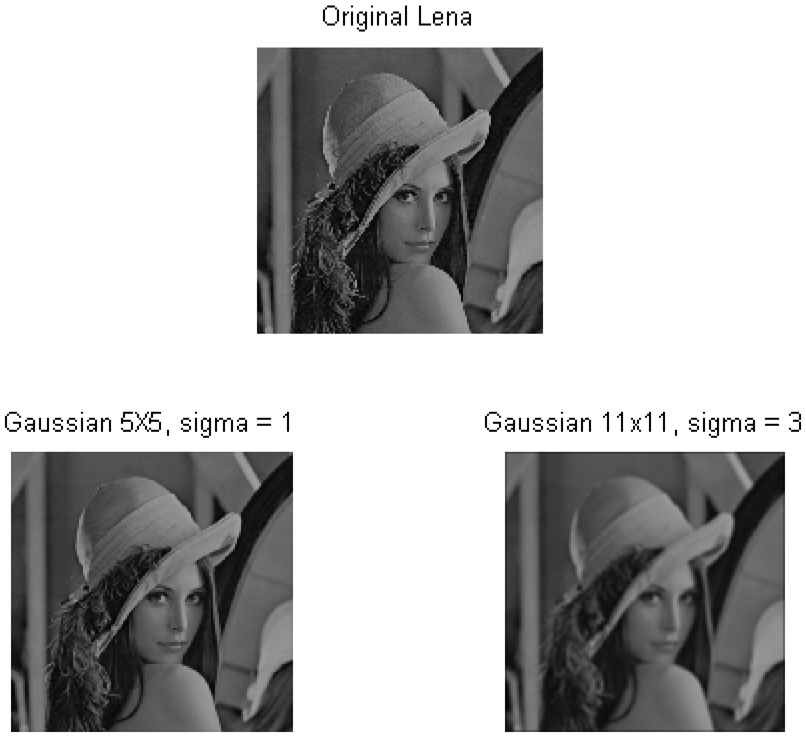
Table of Contents

```
%%Problem 1

M5 = gaussian_kernel(5,1);
M11 = gaussian_kernel(11,3);

lena = im2double(imread('../hw1_images/lena.bmp'));
lena_noise = im2double(imread('../hw1_images/lena_noise.bmp'));

barbara = im2double(imread('../hw1_images/barbara.bmp'));
barbara_noise = im2double(imread('../hw1_images/barbara_noise.bmp'));

mandrill = im2double(imread('../hw1_images/mandrill.bmp'));
mandrill_noise = im2double(imread('../hw1_images/mandrill_noise.bmp'));
```

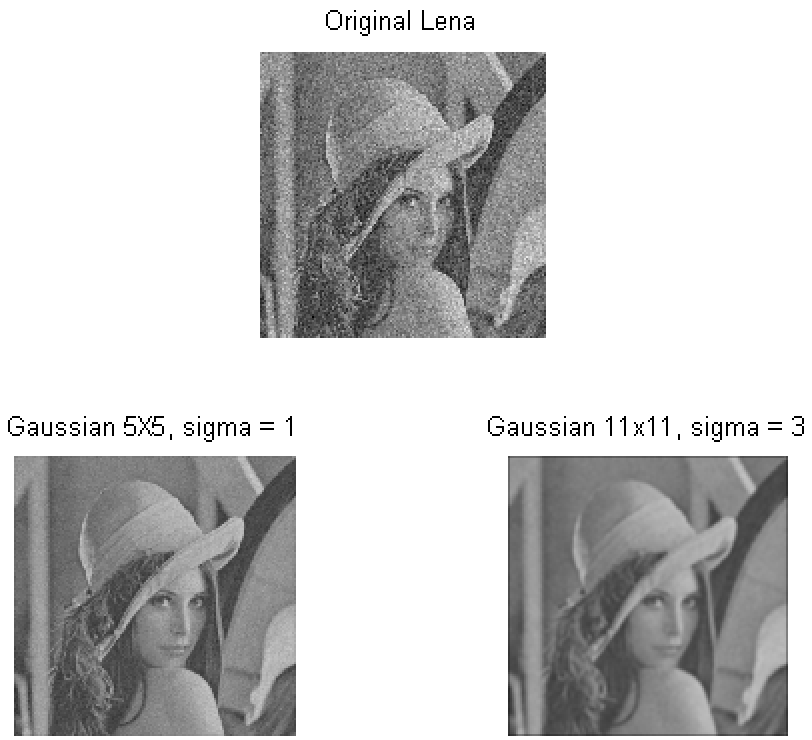
Lena

```
lena_conv_sigma1 = conv2(lena,M5,'same');
lena_conv_sigma3 = conv2(lena,M11,'same');

figure;
subplot(2,2,1.5), subimage(lena);
title('Original Lena')
axis off
subplot(2,2,3), subimage(lena_conv_sigma1)
title('Gaussian 5X5, sigma = 1')
axis off

subplot(2,2,4), subimage(lena_conv_sigma3)
title('Gaussian 11x11, sigma = 3')
axis off
```



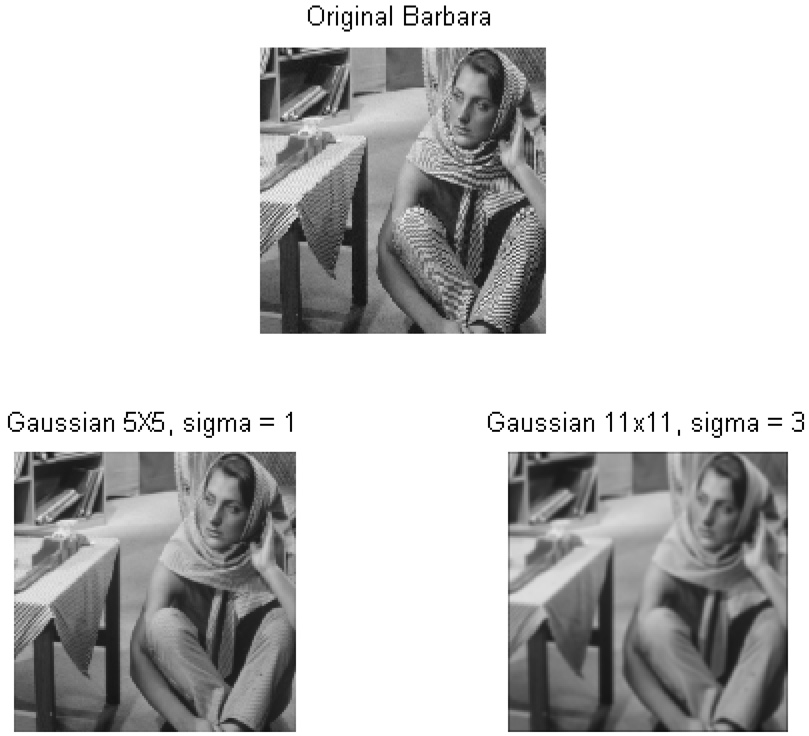
Lena Noise

```
lena_noise_conv_sigma1 = conv2(lena_noise,M5, 'same');
lena_noise_conv_sigma3 = conv2(lena_noise,M11, 'same');

figure;
subplot(2,2,1.5), subimage(lena_noise);
title('Original Lena')
axis off
subplot(2,2,3), subimage(lena_noise_conv_sigma1)
title('Gaussian 5X5, sigma = 1')
axis off
imwrite(lena_noise_conv_sigma1, 'lena_5x5_sigma_1.bmp')

subplot(2,2,4), subimage(lena_noise_conv_sigma3)
title('Gaussian 11x11, sigma = 3')
axis off
imwrite(lena_noise_conv_sigma3, 'lena_11x11_sigma_3.bmp')
```



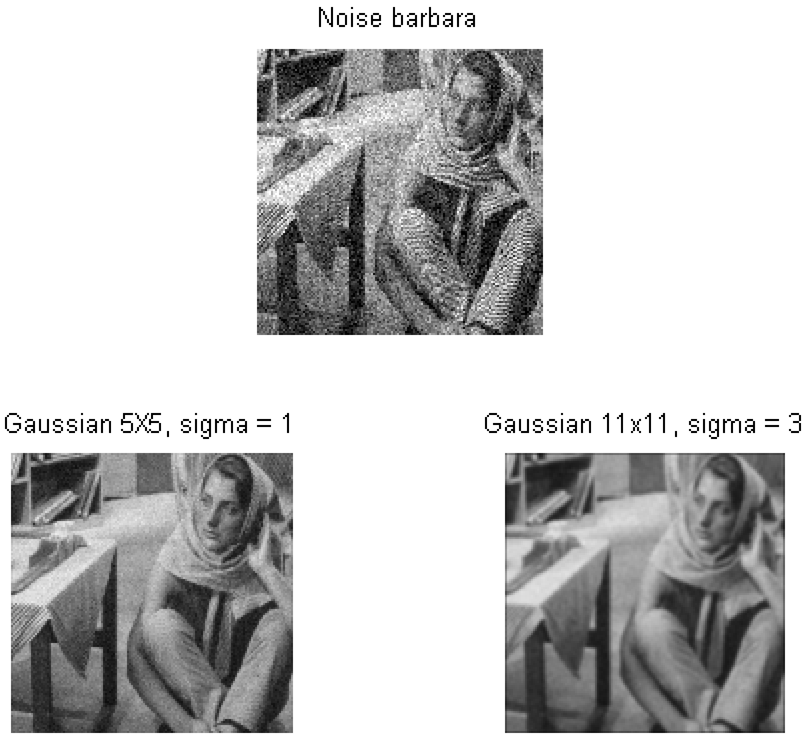
Barbara

```
barbara_conv_sigma1 = conv2(barbara,M5,'same');
barbara_conv_sigma3 = conv2(barbara,M11,'same');

figure;
subplot(2,2,1.5), subimage(barbara);
title('Original Barbara')
axis off
subplot(2,2,3), subimage(barbara_conv_sigma1)
title('Gaussian 5X5, sigma = 1')
axis off

subplot(2,2,4), subimage(barbara_conv_sigma3)
title('Gaussian 11x11, sigma = 3')
axis off
```



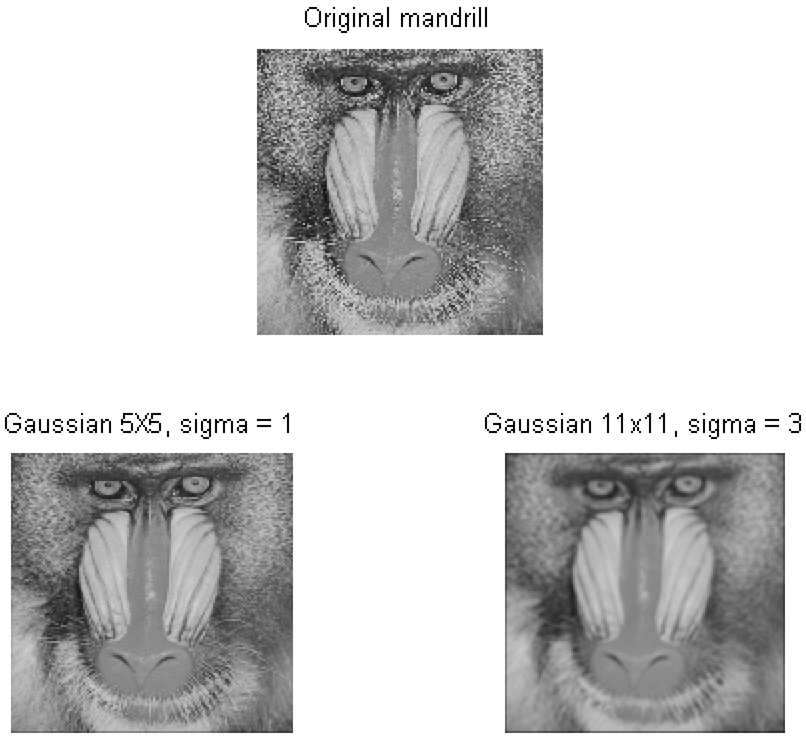
Barbara Noise

```
barbara_noise_conv_sigma1 = conv2(barbara_noise,M5,'same');
barbara_noise_conv_sigma3 = conv2(barbara_noise,M11,'same');

figure;
subplot(2,2,1.5), subimage(barbara_noise);
title('Noise barbara')
axis off
subplot(2,2,3), subimage(barbara_noise_conv_sigma1)
title('Gaussian 5X5, sigma = 1')
axis off
imwrite(barbara_noise_conv_sigma1, 'barbara_5x5_sigma_1.bmp')

subplot(2,2,4), subimage(barbara_noise_conv_sigma3)
title('Gaussian 11x11, sigma = 3')
axis off
imwrite(barbara_noise_conv_sigma3, 'barbara_11x11_sigma_3.bmp')
```



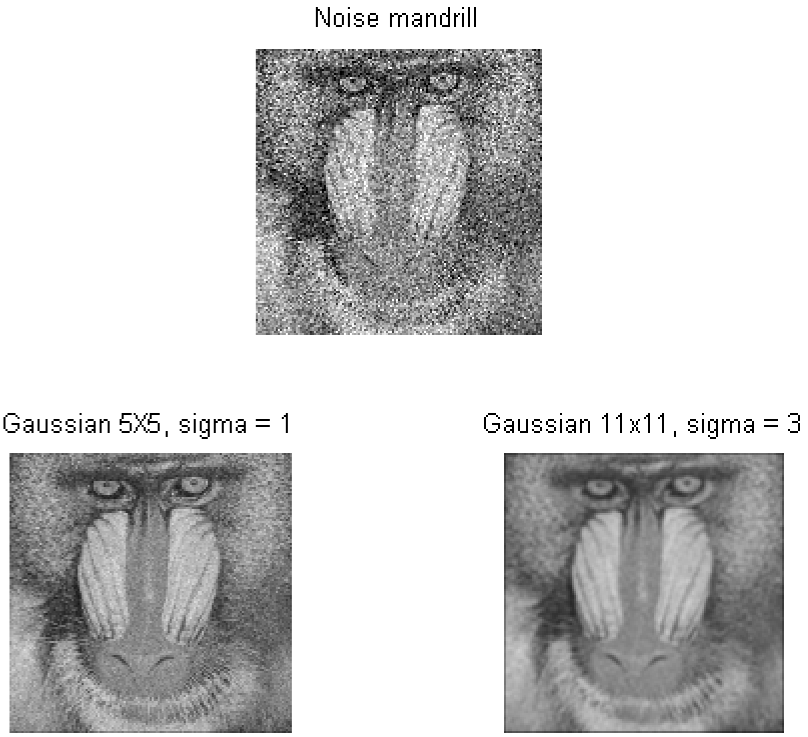
Mandrill

```
mandrill_conv_sigma1 = conv2(mandrill,M5,'same');
mandrill_conv_sigma3 = conv2(mandrill,M11,'same');

figure;
subplot(2,2,1.5), subimage(mandrill);
title('Original mandrill')
axis off
subplot(2,2,3), subimage(mandrill_conv_sigma1)
title('Gaussian 5X5, sigma = 1')
axis off

subplot(2,2,4), subimage(mandrill_conv_sigma3)
title('Gaussian 11x11, sigma = 3')
axis off
```



Mandrill Noise

```
mandrill_noise_conv_sigma1 = conv2(mandrill_noise,M5,'same');
mandrill_noise_conv_sigma3 = conv2(mandrill_noise,M11,'same');

figure;
subplot(2,2,1.5), subimage(mandrill_noise);
title('Noise mandrill')
axis off

subplot(2,2,3), subimage(mandrill_noise_conv_sigma1)
title('Gaussian 5X5, sigma = 1')
axis off
imwrite(mandrill_noise_conv_sigma1, 'mandrill_5x5_sigma_1.bmp')

subplot(2,2,4), subimage(mandrill_noise_conv_sigma3)
title('Gaussian 11x11, sigma = 3')
axis off
imwrite(mandrill_noise_conv_sigma3, 'mandrill_11x11_sigma_3.bmp')
```



Published with MATLAB® R2014a

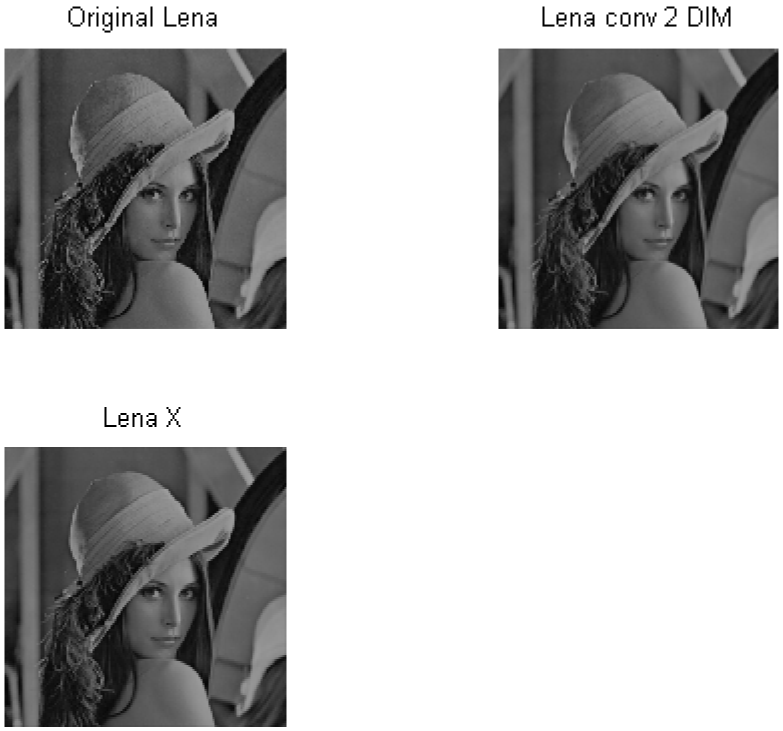
Problem2

Table of Contents

```
M2 = gaussian_kernel_two_dimension(5,1);
M_X = gaussian_kernel_single_dimension(5,1);
M_Y = M_X';

lena = im2double(imread('../hw1_images/lena.bmp'));
```

Lena

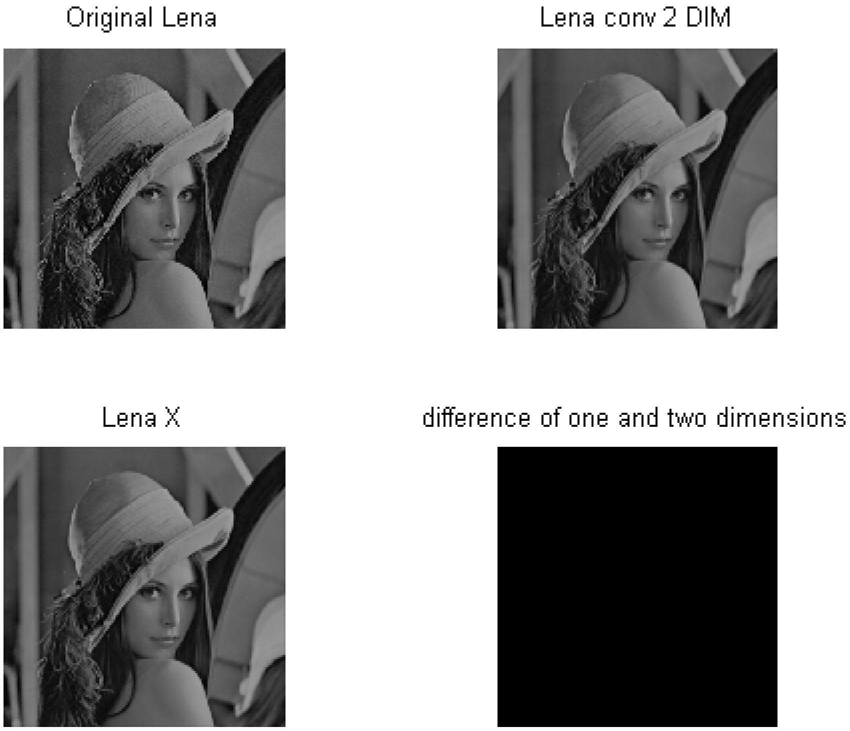
```
lena_conv2 = conv2(lena,M2, 'same');
lena_conv_X = conv2(lena,M_X, 'same');
lena_conv_Y = conv2(lena_conv_X, M_Y, 'same');

figure;
subplot(2,2,1), subimage(lena);
title('Original Lena')
axis off

subplot(2,2,2), subimage(lena_conv2);
title('Lena conv 2 DIM')
axis off


subplot(2,2,3), subimage(lena_conv_Y)
title('Lena X')
axis off
```



Difference of images

```
diff = abs(lena_conv2-lena_conv_Y);

subplot(2,2,4), subimage(diff)
title('difference of one and two dimensions')
axis off
%sum is Zero
output = sum(sum(diff))
```

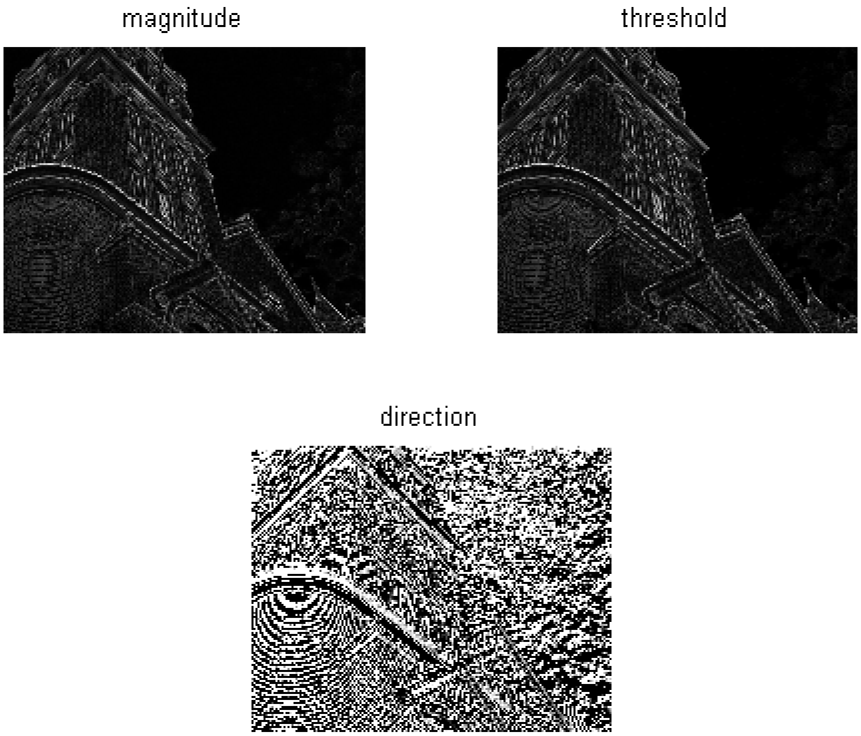
```
output =

   1.0783e-11
```



Published with MATLAB® R2014a

Problem 3

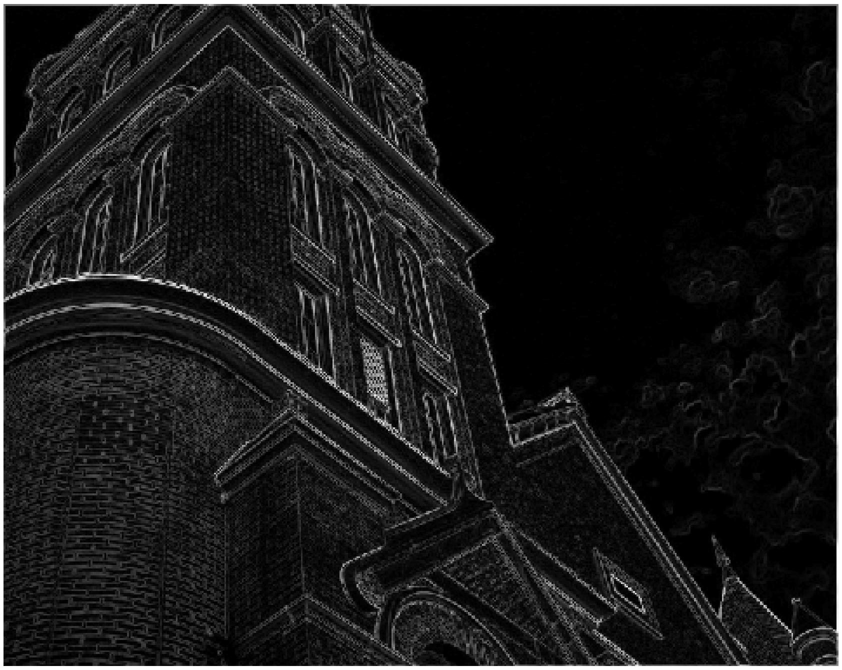
```
% function [magnitude, direction] = sobel(I, th)
 th = 0.01;
 I = imread('../hw1_images/building.bmp');

I = im2double(I);
kernel_x = [-1, 0, +1; -2, 0, +2; -1, 0, +1];
kernel_y = [+1, +2, +1; 0, 0, 0; -1, -2, -1];

sobel_x = conv2(I, kernel_x, 'same');
sobel_y = conv2(I, kernel_y, 'same');

magnitude = sqrt(sobel_x.^2 + sobel_y.^2);
%Normalize
magnitude = magnitude/max(max(magnitude));

direction = atan2(sobel_y, sobel_x);

threshold = (magnitude > th);
threshold = threshold.* magnitude;

result = im2uint8(threshold);
figure;
imwrite(result, 'sobel_th_1.5.bmp');

figure;
subplot(2,2,1), imshow(abs(magnitude));
title('magnitude')

subplot(2,2,2),imshow(threshold);
title('threshold')

subplot(2,2,3.5),imshow(direction);
title('direction')
```


magnitude
threshold
direction

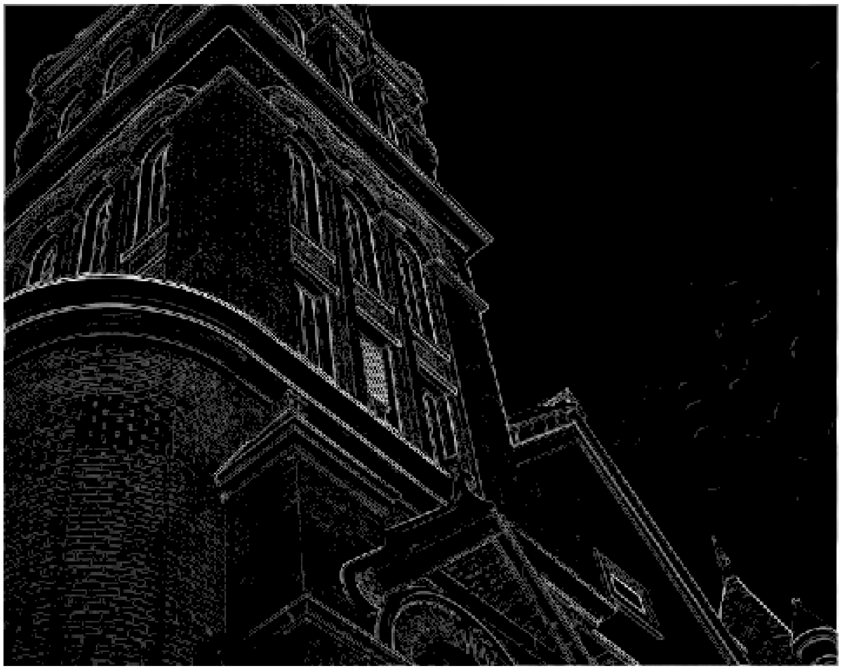
At different thresholds we can get different results it upon us what edges are important for us

```
figure;
imshow('sobel_th_0.01.bmp');
title('Th = 0.01');

figure;
imshow('sobel_th_0.1.bmp');
title('Th = 0.1');
figure;

imshow('sobel_th_0.1.bmp');
title('Th = 0.15');

figure;
imshow('sobel_th_0.2.bmp');
title('Th = 0.12');


%  end
```

```
        Warning: Image is too big to fit on screen; displaying at 67%
        Warning: Image is too big to fit on screen; displaying at 67%
        Warning: Image is too big to fit on screen; displaying at 67%
        Warning: Image is too big to fit on screen; displaying at 67%
```

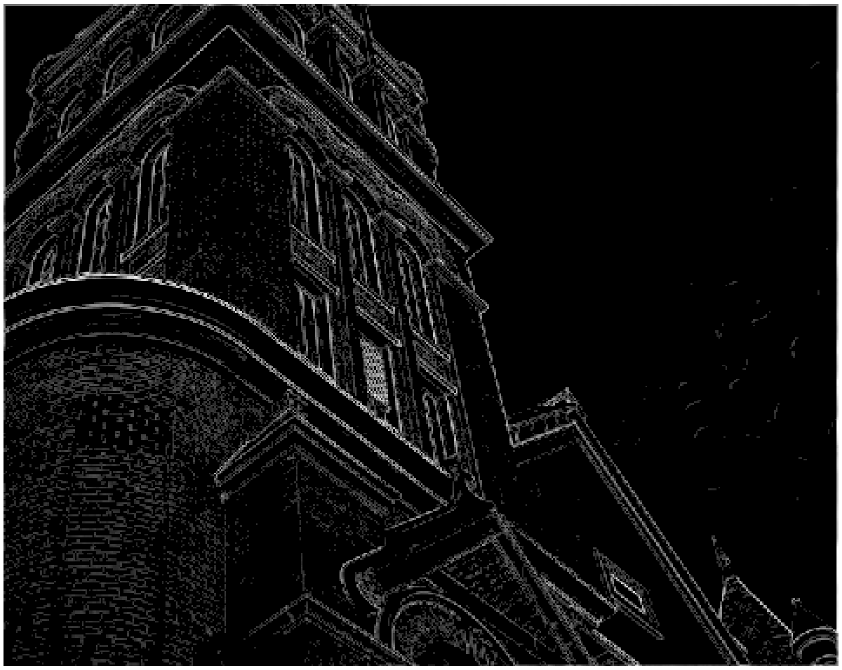


Th = 0.01

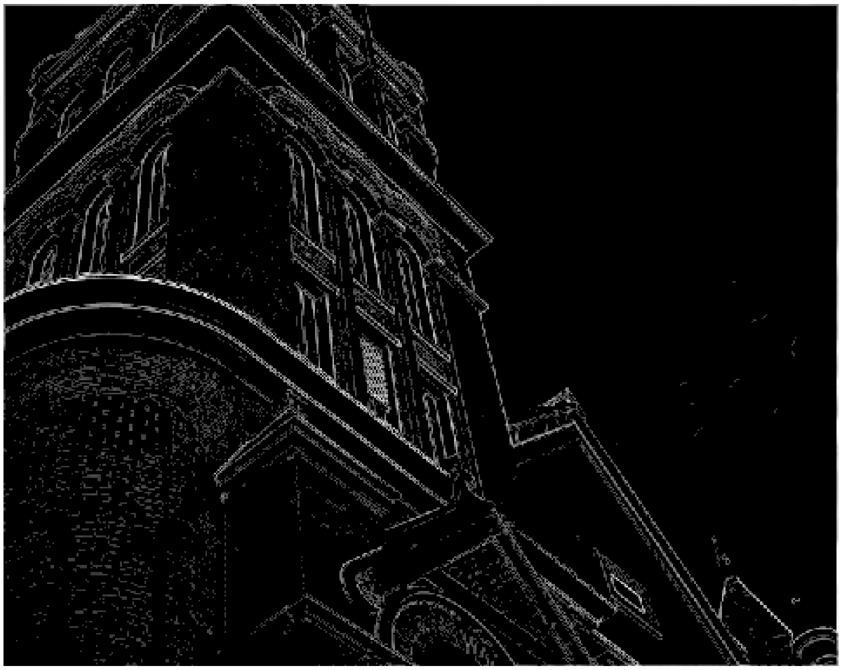


Th = 0.1

Th = 0.15



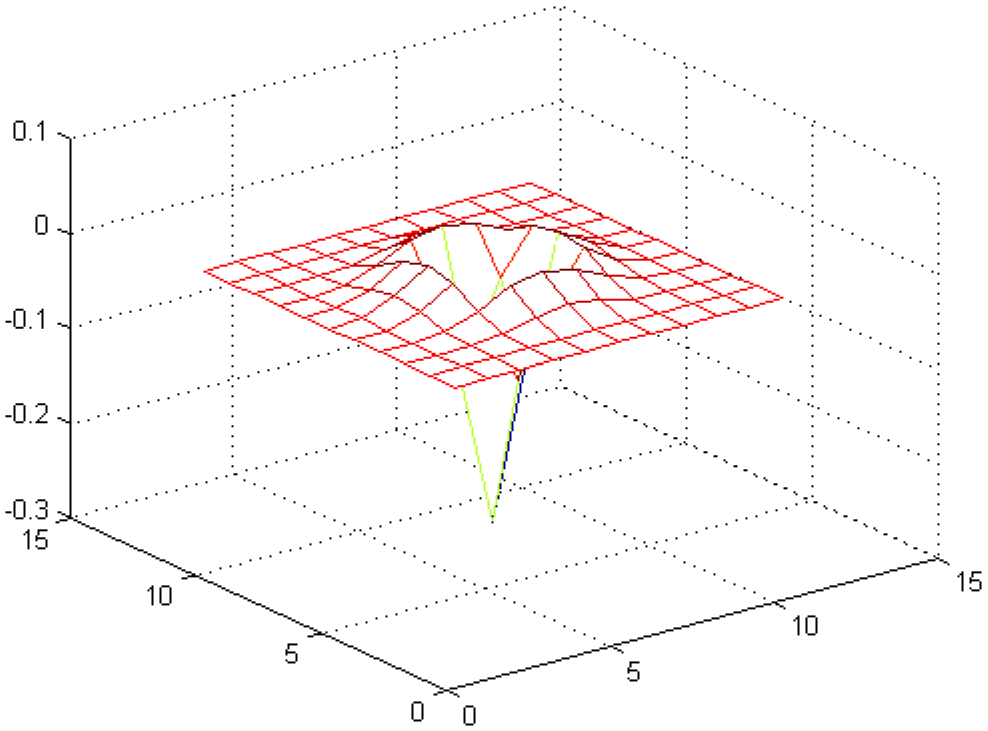
Th = 0.12



Published with MATLAB® R2014a

Problem 4

```
gauss = gaussian_kernel(11,1);
laplacian_kernel = [0,1,0;1,-4,1;0,1,0];
conv  = conv2(gauss, laplacian_kernel,'same');
mesh(conv);
% x = -15 : 0.1 : 15;
% y = -15 : 0.1 : 15;
% plot3(x,y, conv);
```

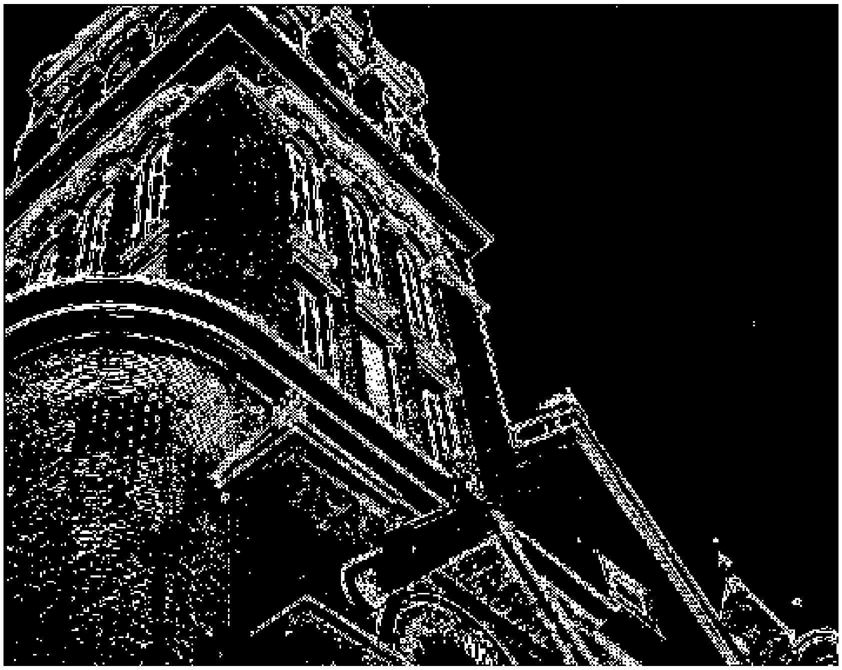


Published with MATLAB® R2014a

Problem 5

```
% function [zcd1] = main(I,th)
I = imread('../hw1_images/building.bmp');
th = 0.1;
I = im2double(I);
log = log_kernel(11,1);
zcd = conv2(I, log,'same');

zcd1 = zeros(512,512);
% row
for i = 2:size(zcd,1)-1
% col
    for j = 2:size(zcd,2)-1
        if(( (zcd(i-1,j)* zcd(i+1,j)<0) &&  abs(zcd(i-1,j)-zcd(i+1,j)) > th) || ((
            zcd1(i,j) = 1;
        end
    end
end

% zcd1 = im2uint8(zcd1);
figure;
imshow(zcd1);
imwrite(zcd1, 'zcd_th0.08.bmp');


% end
```

```
Warning: Image is too big to fit on screen; displaying at 67%
```



Published with MATLAB® R2014a

Problem 6 Canny with non Maximal Supression

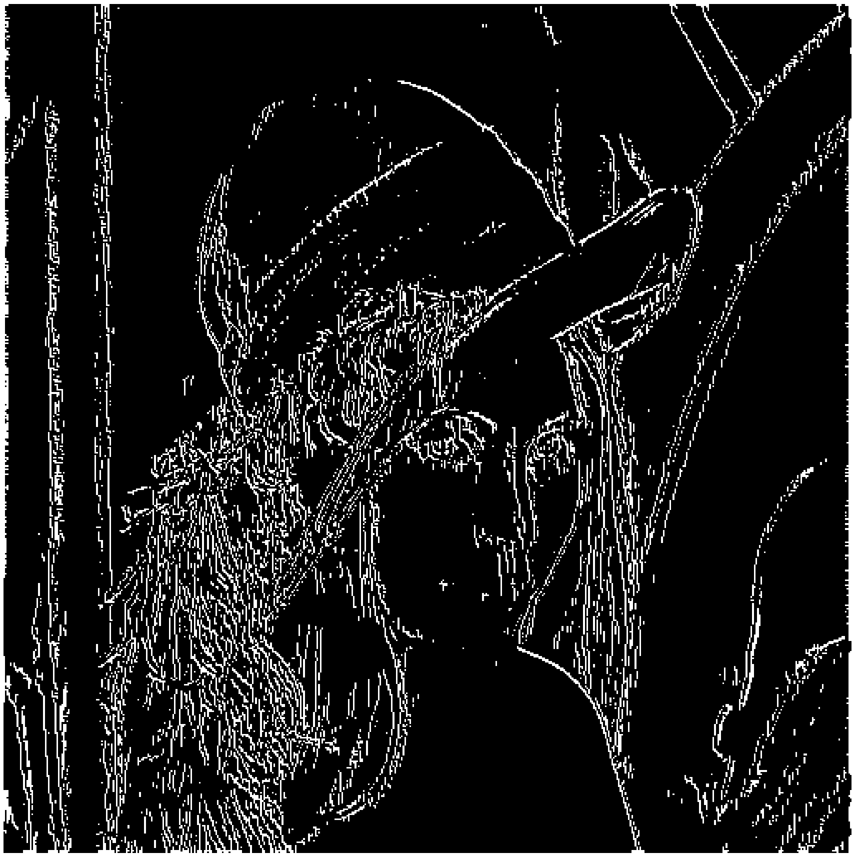
```
% function [magnitude] = canny(I, w, var, th_low)
I = imread('../hw1_images/lena.bmp');
w = 23;
var = 2;
th_low = 0.07;
I = im2double(I);
gauss = gaussian_kernel_single_dimension(w,var);
IX = conv2(I, gauss, 'same');
IY = conv2(I, gauss', 'same');
% kernel_x = [-2, 0, +2];
% kernel_y = [-2, 0, +2]';

% grad_x = conv2(IX, kernel_x, 'same');
% grad_y = conv2(IY, kernel_y, 'same');
grad_x = gradient(IX, w);
grad_y = gradient(IY, w);


magnitude = sqrt(grad_x.^2 + grad_y.^2);
magnitude = magnitude/max(max(magnitude));

% figure;
% subplot(2,2,1), imshow(abs(grad_x));
% title('grad_x')
%
% subplot(2,2,2),imshow(abs(grad_y));
% title('grad_y')
%
% subplot(2,2,3),imshow(magnitude);
% title('magnitute')

non_max = magnitude;
```

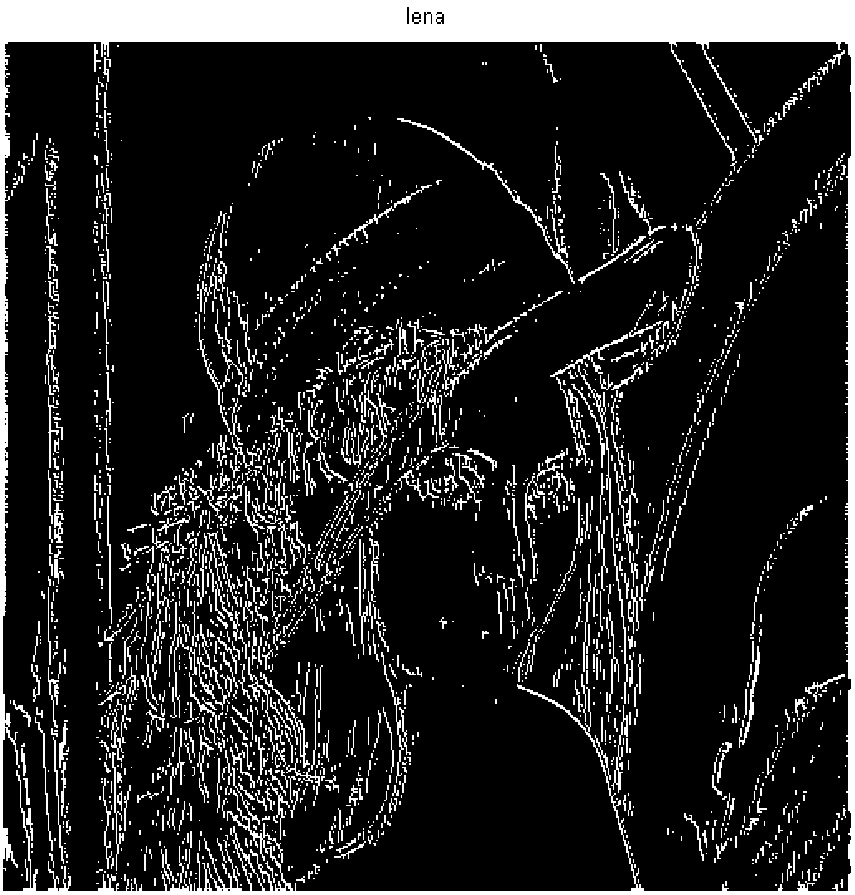
Non maximal Supression

```
for i = 2 : size(magnitude,1)-2
    for j = 2 : size(magnitude,2)-2
%90 degree

        tangent = grad_y(i,j)/grad_x(i,j);

        if(grad_x(i,j) == 0)
            if((magnitude(i,j) < magnitude(i-1,j) || magnitude(i,j) < magnitude(i+
                non_max(i,j)=0.0;
            end

        elseif(tangent == 0)
            if((magnitude(i,j) < magnitude(i,j-1) || magnitude(i,j) < magnitude(i,
                non_max(i,j) = 0.0;
            end


        elseif(tangent > 0 && tangent <= 1)
            inter1 = tangent * magnitude(i-1,j+1) + (1-tangent)*magnitude(i,j+1);
            inter2 = tangent * magnitude(i+1,j-1) + (1-tangent)*magnitude(i,j-1);

            if(magnitude(i,j) < inter1 || magnitude(i,j) < inter2)
                non_max(i,j) = 0.0;
%                 non_max(i,j)
            end

        elseif(tangent > 0 && tangent >= 1)
            inter1 = (1/tangent) * magnitude(i-1,j+1) + (1-(1/tangent))*magnitude(
            inter2 = (1/tangent) * magnitude(i+1,j-1) + (1-tangent)*magnitude(i+1,

            if(magnitude(i,j) < inter1 || magnitude(i,j) < inter2)
                non_max(i,j) = 0.0;
%                 non_max(i,j)
            end

        elseif(tangent < 0 && abs(tangent) < 1)
            inter1 = tangent * magnitude(i+1,j+1) + (1-tangent)*magnitude(i,j+1);
            inter2 = tangent * magnitude(i-1,j-1) + (1-tangent)*magnitude(i-1,j);

            if(magnitude(i,j) < inter1 || magnitude(i,j) < inter2)
                non_max(i,j) = 0.0;
%                 non_max(i,j)
            end

        elseif(tangent < 0 && abs(tangent) > 1)
            inter1 = (1/tangent) * magnitude(i+1,j+1) + (1-(1/tangent))*magnitude(
            inter2 = (1/tangent) * magnitude(i-1,j-1) + (1-(1/tangent))*magnitude(

            if(magnitude(i,j) < inter1 || magnitude(i,j) < inter2)
                non_max(i,j) = 0.0;
            end
        end

    end
end
% non_max = im2uint8(non_max);

% subplot(2,2,4),imshow(non_max);
% title('non max')

result1 = (non_max > th_low);
result1 = im2uint8(result1);
imwrite(result1, 'non_max_23_1_0.07.bmp');
figure;
imshow(result1);
figure;
imshow('barbara_23_2_0.07.bmp');
figure;
```

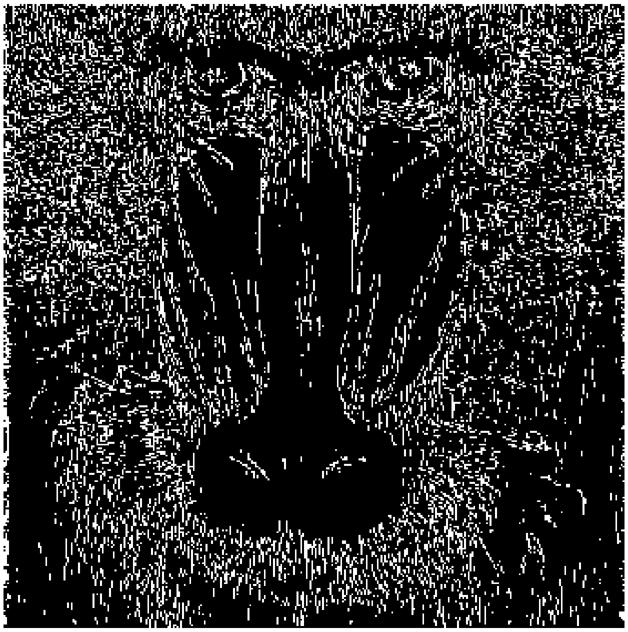
```
title('barbara')
imshow('lena_23_2_0.07.bmp');
title('lena')

figure;
imshow('baboon.bmp');
title('mandrill')

figure;
imshow('building_23_2_0.1.bmp');
title('building')

figure;
imshow('barbara_noise_23_1_0.07.bmp');
title('barbara noise')

figure;
imshow('mandrill_50_1.5_0.3.bmp');
title('mandrill noise')

figure;
imshow('lena_noise_23_2.5_0.2.bmp');
title('mandrill noise')

% end
```

```
        Warning: Image is too big to fit on screen; displaying at 67%
        Warning: Image is too big to fit on screen; displaying at 67%
```

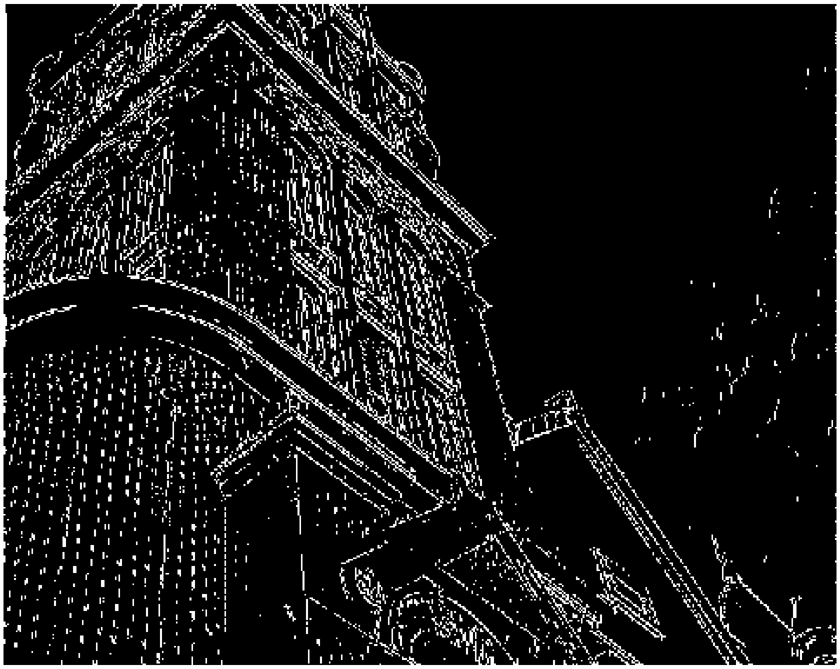


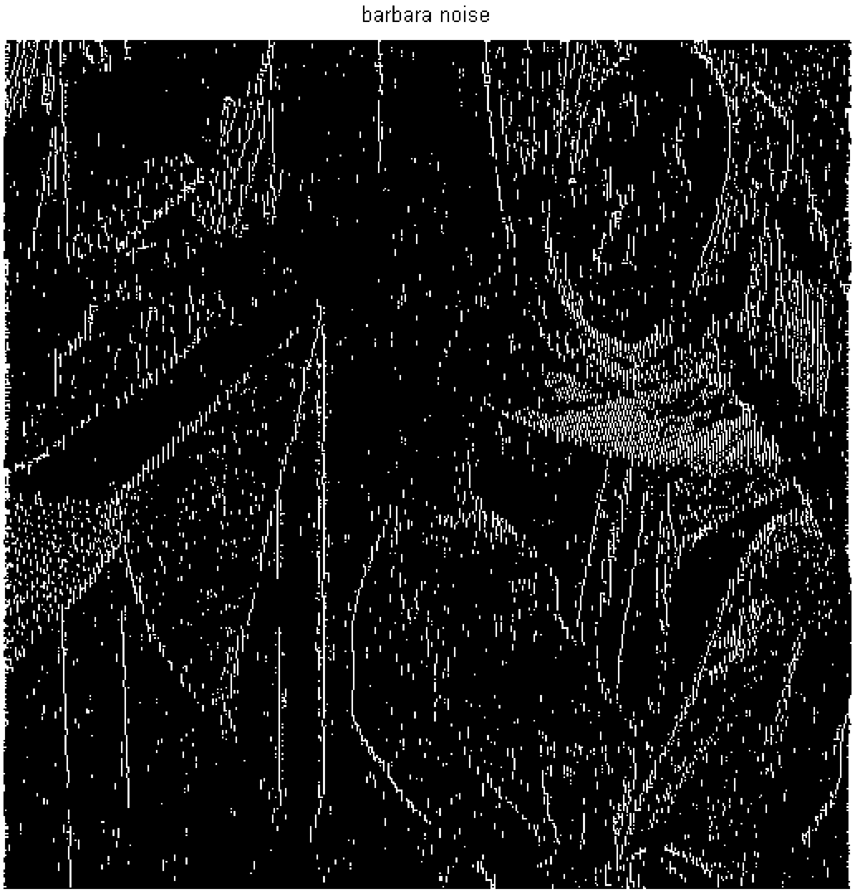

lena

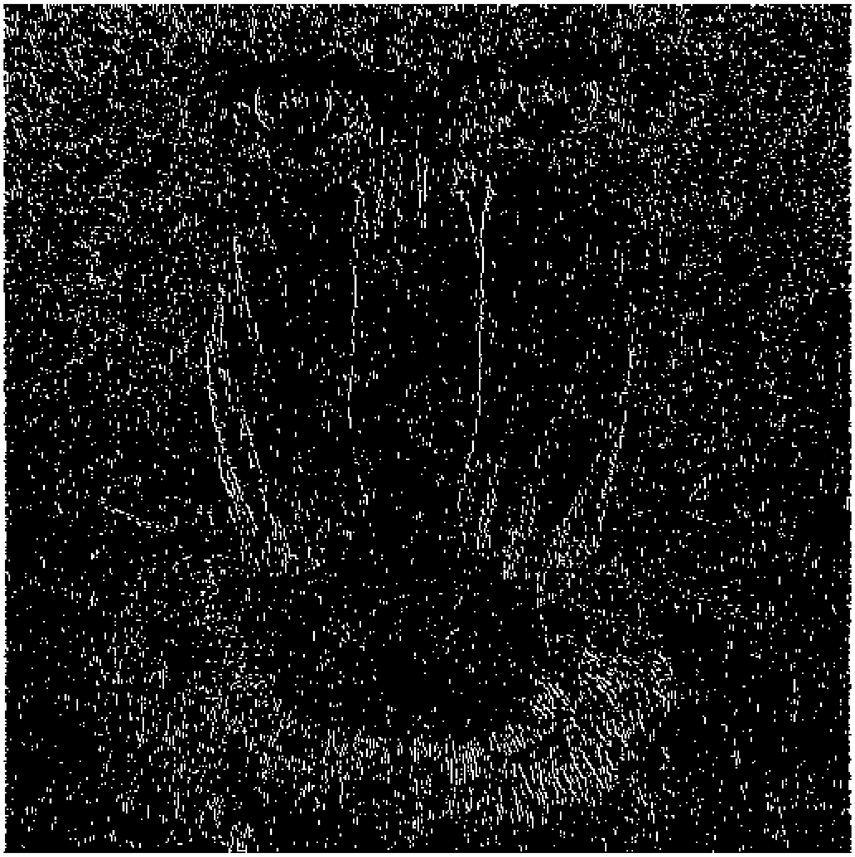
mandrill



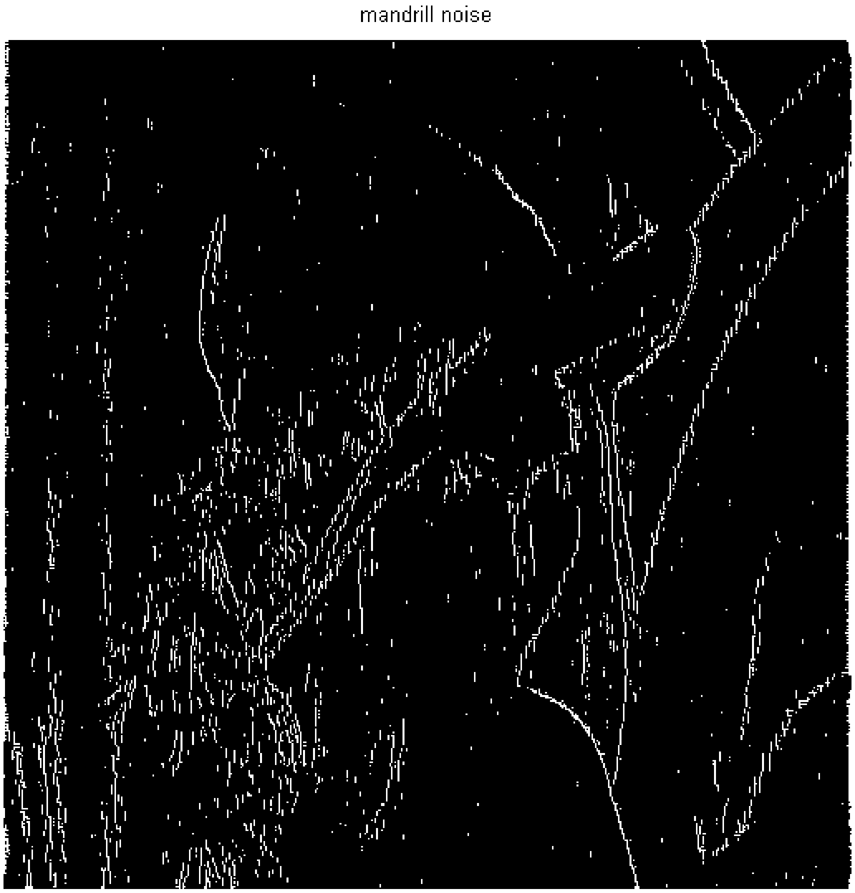
building




barbara noise



mandrill noise



Published with MATLAB® R2014a

```
% Problem 7
```

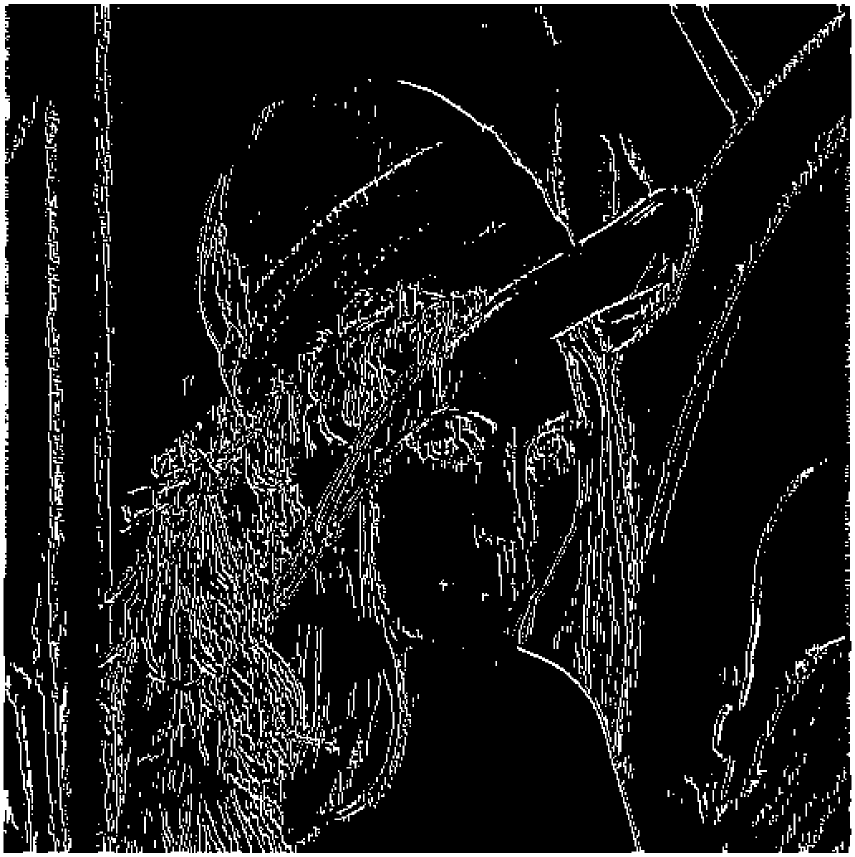
After Hysterisis and non maximal supression

```
I = imread('../hw1_images/lena.bmp');
w = 23;
var = 2;
th_low = 0.07;
th_high = 0.2;

%function [magnitude] = canny(I, w, var, th_low, th_high)
I = im2double(I);
gauss = gaussian_kernel_single_dimension(w,var);
IX = conv2(I, gauss, 'same');
IY = conv2(I, gauss', 'same');
% kernel_x = [-2, 0, +2];
% kernel_y = [-2, 0, +2]';

% grad_x = conv2(IX, kernel_x, 'same');
% grad_y = conv2(IY, kernel_y, 'same');
grad_x = gradient(IX);
grad_y = gradient(IY);


magnitude = sqrt(grad_x.^2 + grad_y.^2);
magnitude = magnitude/max(max(magnitude));

% figure;
% subplot(2,2,1), imshow(abs(grad_x));
% title('grad_x')
%
% subplot(2,2,2),imshow(abs(grad_y));
% title('grad_y')
%
% subplot(2,2,3),imshow(magnitude);
% title('magnitute')

non_max = magnitude;
```

Non maximal Supression

```
for i = 2 : size(magnitude,1)-1
    for j = 2 : size(magnitude,2)-1
%90 degree

        tangent = grad_y(i,j)/grad_x(i,j);

        if(grad_x(i,j) == 0)
            if((magnitude(i,j) < magnitude(i-1,j) || magnitude(i,j) < magnitude(i+
                non_max(i,j)=0.0;
```

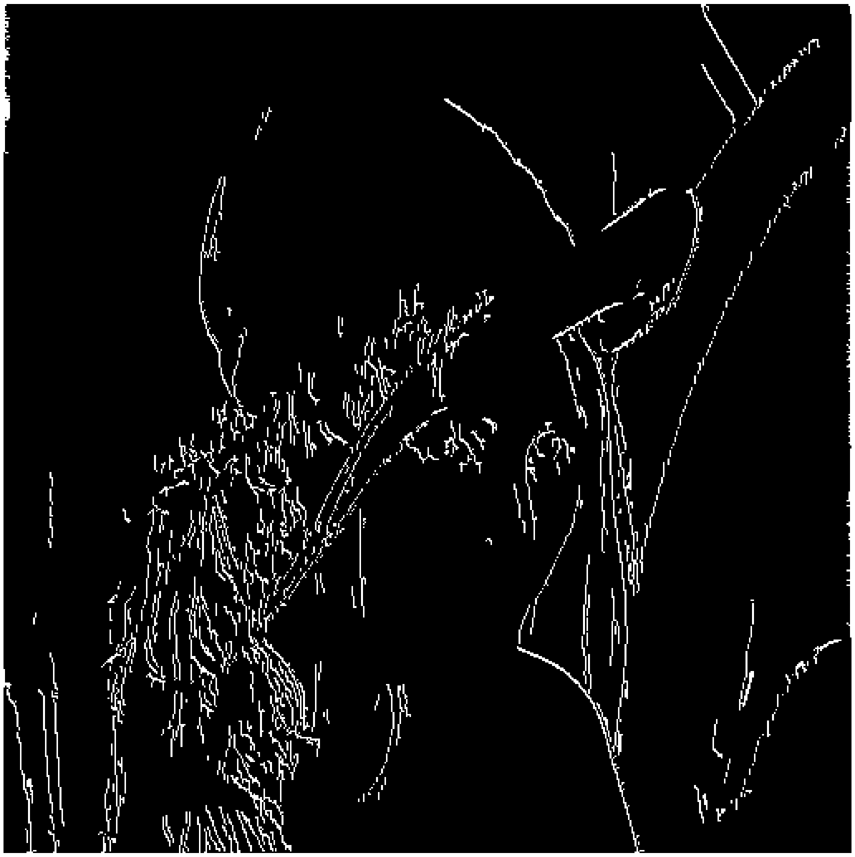
```
      end

    elseif(tangent == 0)
      if((magnitude(i,j) < magnitude(i,j-1) || magnitude(i,j) < magnitude(i,
        non_max(i,j) = 0.0;
      end


    elseif(tangent > 0 && tangent <= 1)
      inter1 = tangent * magnitude(i-1,j+1) + (1-tangent)*magnitude(i,j+1);
      inter2 = tangent * magnitude(i+1,j-1) + (1-tangent)*magnitude(i,j-1);

      if(magnitude(i,j) < inter1 || magnitude(i,j) < inter2)
        non_max(i,j) = 0.0;
%        non_max(i,j)
      end

    elseif(tangent > 0 && tangent >= 1)
      inter1 = (1/tangent) * magnitude(i-1,j+1) + (1-(1/tangent))*magnitude(
      inter2 = (1/tangent) * magnitude(i+1,j-1) + (1-tangent)*magnitude(i+1,

      if(magnitude(i,j) < inter1 || magnitude(i,j) < inter2)
        non_max(i,j) = 0.0;
%        non_max(i,j)
      end

    elseif(tangent < 0 && abs(tangent) < 1)
      inter1 = tangent * magnitude(i+1,j+1) + (1-tangent)*magnitude(i,j+1);
      inter2 = tangent * magnitude(i-1,j-1) + (1-tangent)*magnitude(i-1,j);

      if(magnitude(i,j) < inter1 || magnitude(i,j) < inter2)
        non_max(i,j) = 0.0;
%        non_max(i,j)
      end

    elseif(tangent < 0 && abs(tangent) > 1)
      inter1 = (1/tangent) * magnitude(i+1,j+1) + (1-(1/tangent))*magnitude(
      inter2 = (1/tangent) * magnitude(i-1,j-1) + (1-(1/tangent))*magnitude(

      if(magnitude(i,j) < inter1 || magnitude(i,j) < inter2)
        non_max(i,j) = 0.0;
      end
    end

  end
end
% non_max = im2uint8(non_max);

% subplot(2,2,4),imshow(non_max);
% title('non max')

result1 = (non_max > th_low);
figure;
imshow(result1);
```

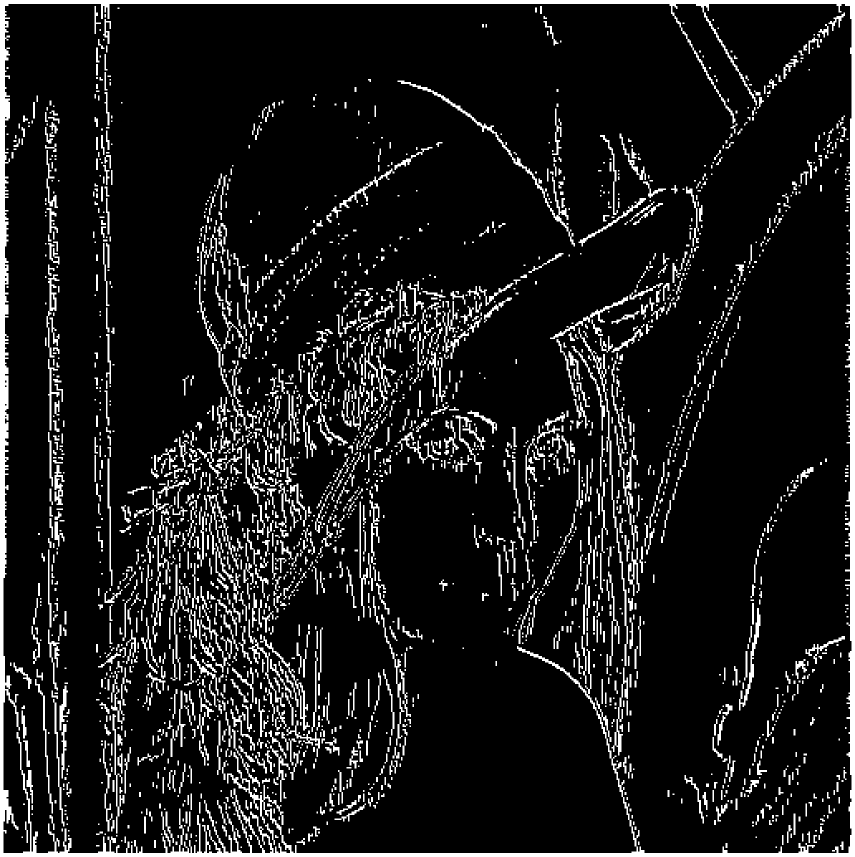


```
hesterisis = non_max;

hesterisis_low = (hesterisis > th_low);
hesterisis_high = (hesterisis > th_high);
% % figure;
% % subplot(2,2,1), imshow(hesterisis_low)
% % subplot(2,2,2), imshow(hesterisis_high)
%
final = hesterisis_high;

for r = 2: size(hesterisis_high,1)-1
    for c = 2: size(hesterisis_high,2)-1
        if(hesterisis_high(r,c))
            final(r,c) = 1;

        elseif(final(r-1,c) || final(r,c-1) || final(r-1,c+1)|| final(r-1,c-1)|| f
             if(hesterisis_low(r,c)~=0)
                final(r,c)=1;
             end
        end
    end
```

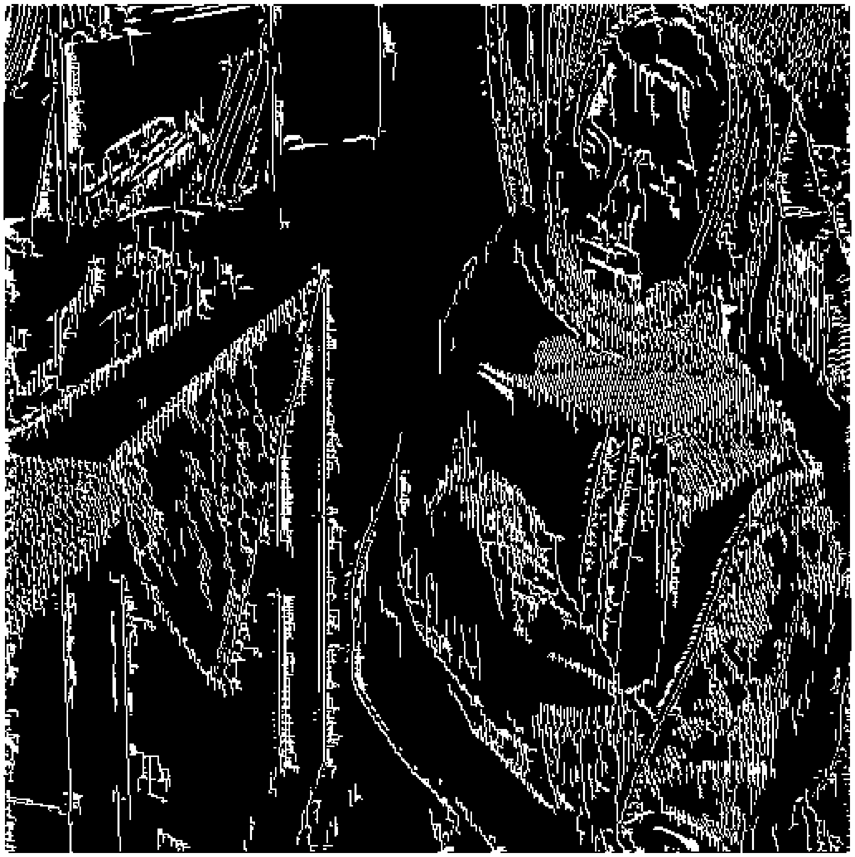
```
end
% % % subplot(2,2,3),imshow(non_max);
% % % subplot(2,2,4), imshow(final)
% %
figure;
imshow(final);
final = im2uint8(final);
imwrite(final, 'hesterisis.bmp');

figure;
imshow(result1);
figure;
imshow('barbara.bmp');
title('barbara')

figure;
imshow('lena_23_2_0.07.bmp');
title('lena')

figure;
imshow('baboon.bmp');
title('mandrill')

figure;
imshow('building.bmp');
title('building')

figure;
imshow('barbara_noise_23_1_0.07.bmp');
title('barbara noise')

figure;
imshow('mandrill_50_1.5_0.3.bmp');
title('mandrill noise')

figure;
imshow('lena_noise_23_2.5_0.2.bmp');
title('mandrill noise')

% imshow(post_hysteresis)
% end
```

```
        Warning: Image is too big to fit on screen; displaying at 67%
```

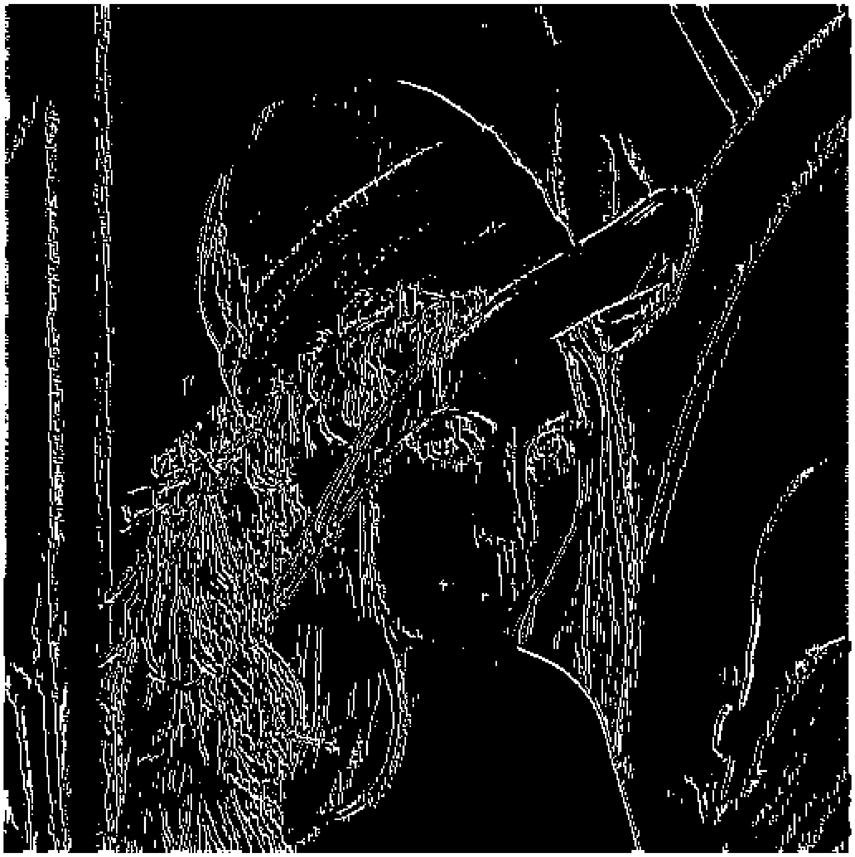




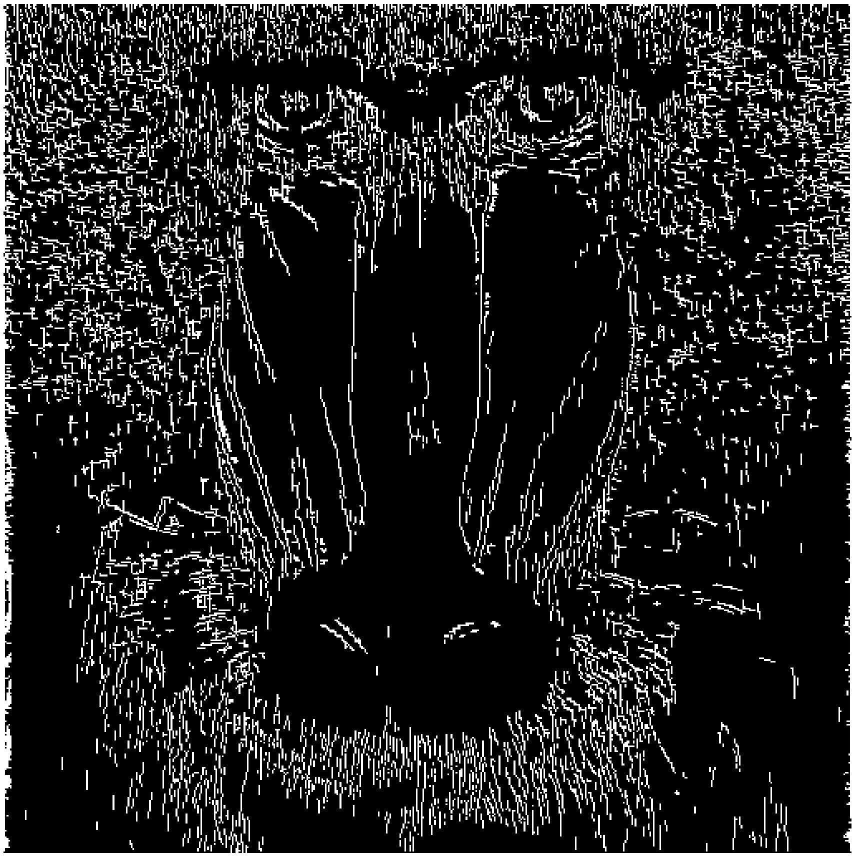
barbara

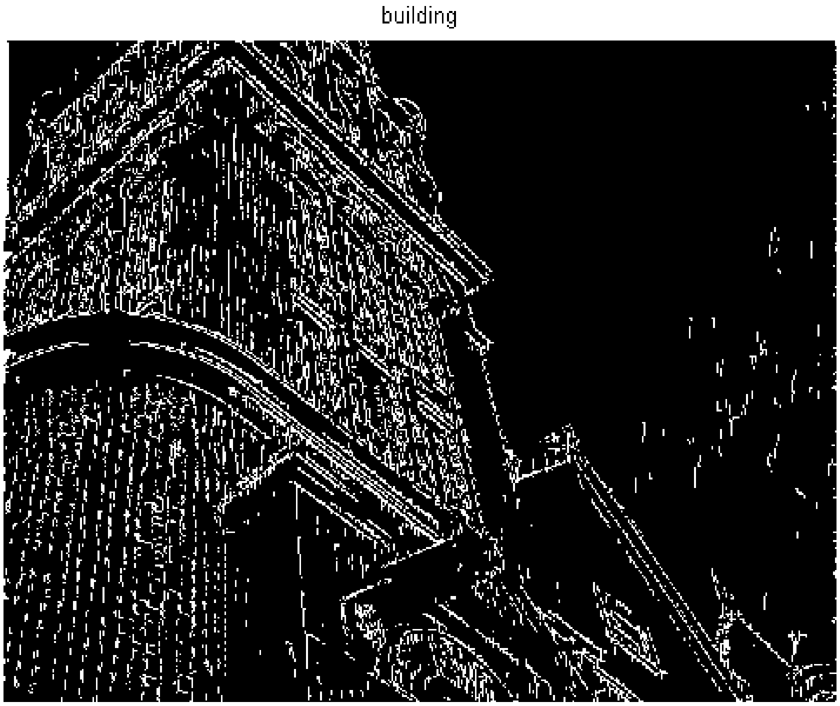


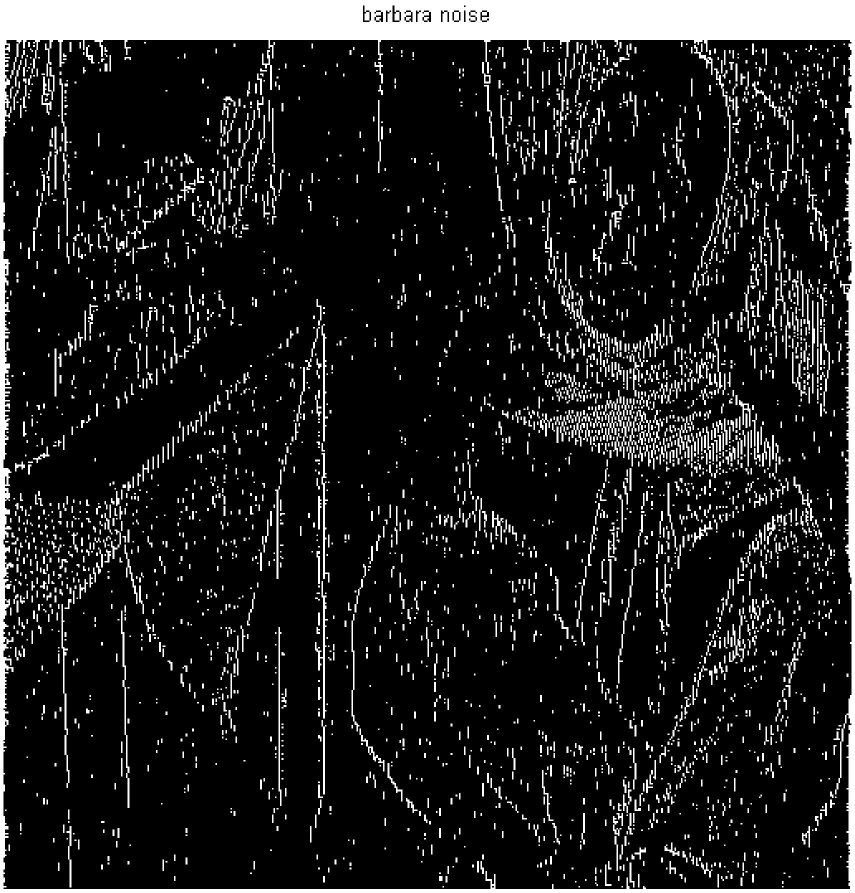
lena



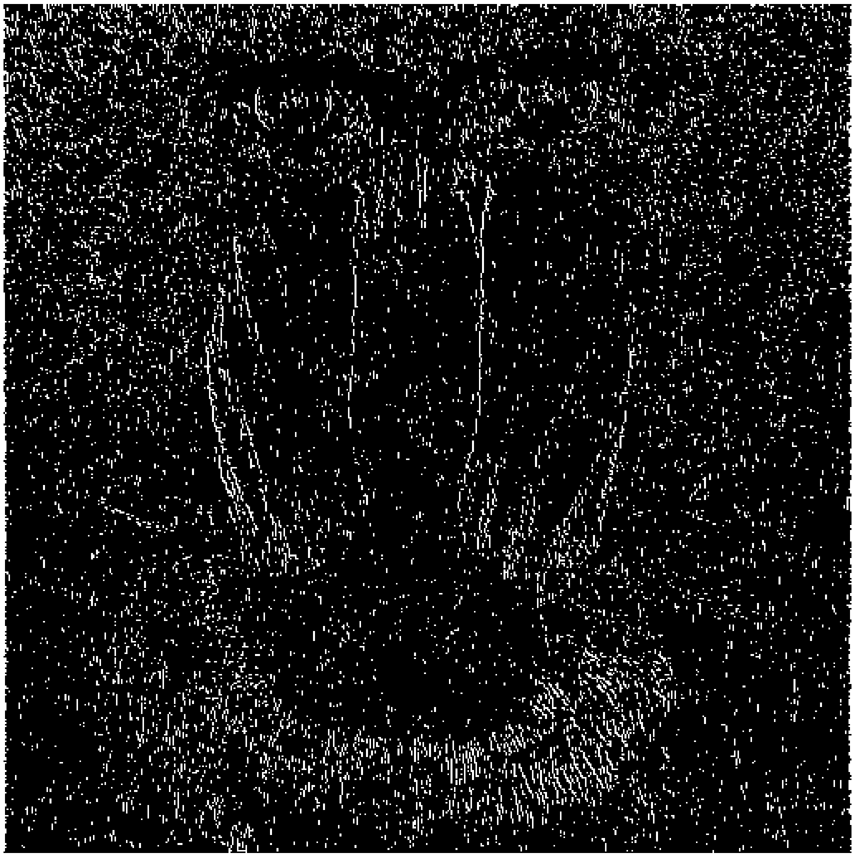
mandrill

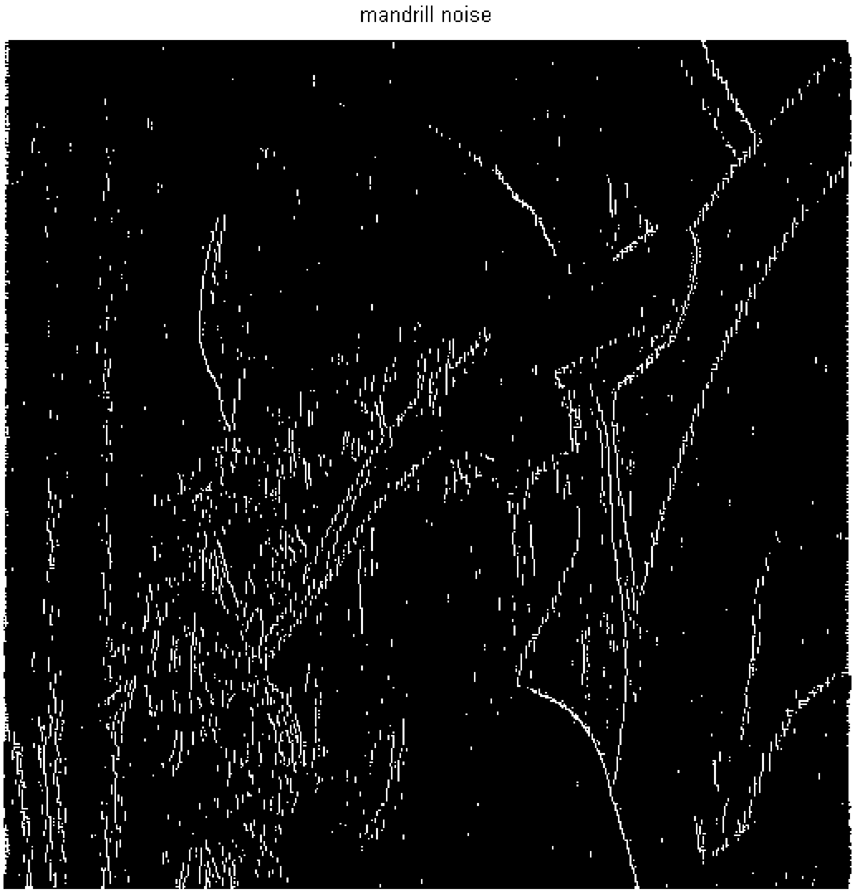



building


barbara noise

mandrill noise





Published with MATLAB® R2014a