
```

clear all;
clc;
clf
close all;

filetype='.jpg';
%-----%
% problem 1 %
%-----%

%-----%
% part 1 %
%-----%
%%construct 3x3 normalized gauss mask
%default response to mimic matlab's built in function
dimensions = 3;
sigma = 0.5;
h = gaussfilter(dimensions,sigma)

%-----%
% part 2 %
%-----%
%%built in matlab function, compare to my function
H = fspecial('gaussian',dimensions,sigma)
sprintf('comapring my gauss filter vs builtin: h-H:')
gauss_difference = H-h %proves my mask and built in masks are identical
                     %because it = 0

%-----%
% part 3 %
%-----%
%%construct a 300x300 gauss mask
h2=fspecial('gaussian',300,sigma);

%-----%
% part 4 %
%-----%
%%plot frequency respsns of said mask
figure('name', 'frequency response of gaauss mask');
freqz2(h2)
title('frequency response of gauss mask');
%-----%
% part 5 %
%-----%
%%display magnitude of FFT of gauss mask with DC component in the center
h2 = fft2(h2);
h2=fftshift(h2);
%h2 = sqrt(shifted);
figure('name','shifted fourier');
imshow(h2);
title('shiftedfft');

```

```

%-----%
% problem 2 %
%-----%

%-----%
% part 1 %
%-----%
%%read image (i develop on both windows and linux so this command
%%automatically determines what filepath to use)
image = './Carl_Friedrich_Gauss.jpg';
baseimage = imread(image);
I = imresize(baseimage, [400,400]);
I = rgb2gray(I);

figure('name', 'original image')
imshow(I);
title('original image');
%-----%
% part 2 %
%-----%
%%add gauss noise to image, call it image2
image2 = imnoise(I,'gaussian',0,.05);
figure('name', 'gauss noise added to img');
imshow(image2);
title('gauss noise added to img');

%-----%
% part 3 %
%-----%
%%add salt/pepper noise to image, call it image3
image3 = imnoise(I, 'salt & pepper',.05);
figure('name', 'salt & pepper added to img');
imshow(image3);
title('salt & pepper added to img');

%-----%
% part 4 %
%-----%
%%apply filtering in space domain (convolution)
%see convolution.m

%-----%
% part 5 %
%-----%
%%apply median filtering operation
%see medianfilter.m

%-----%
% part 6 %
%-----%
%%use convolution code to perform gauss filter on image2
FI = convolution(image2, h);
figure('name', 'gauss filtered image2');

```

```

imshow(FI);
title('gauss filtered image2');

% figure('name', 'gauss filter using builtin convolv');
% builtin = conv2(double(I),double(h));
% imshow(builtin);

%-----%
% part 7 %
%-----%
%%use convolution code to perform gauss filter on image3
G3 = convolution(image3, h);
figure('name', 'gauss filtered img3');
imshow(G3);
title('gauss filtered image3');

%-----%
% part 8 %
%-----%
%%use code to perform median filter on image2
med1= medianfilter(image2);
figure('name','median filtered image2');
imshow(med1);
title('median filtered image3');

%-----%
% part 9 %
%-----%
%%use code to perform median filter on image3
med2 = medianfilter(image3);
figure('name','median filtered image3');
imshow(med2);
title('median filtered image3');

%-----%
% part 10 %
%-----%
%%show and discuss results

```

$h =$

0.0113	0.0838	0.0113
0.0838	0.6193	0.0838
0.0113	0.0838	0.0113

$H =$

0.0113	0.0838	0.0113
0.0838	0.6193	0.0838
0.0113	0.0838	0.0113

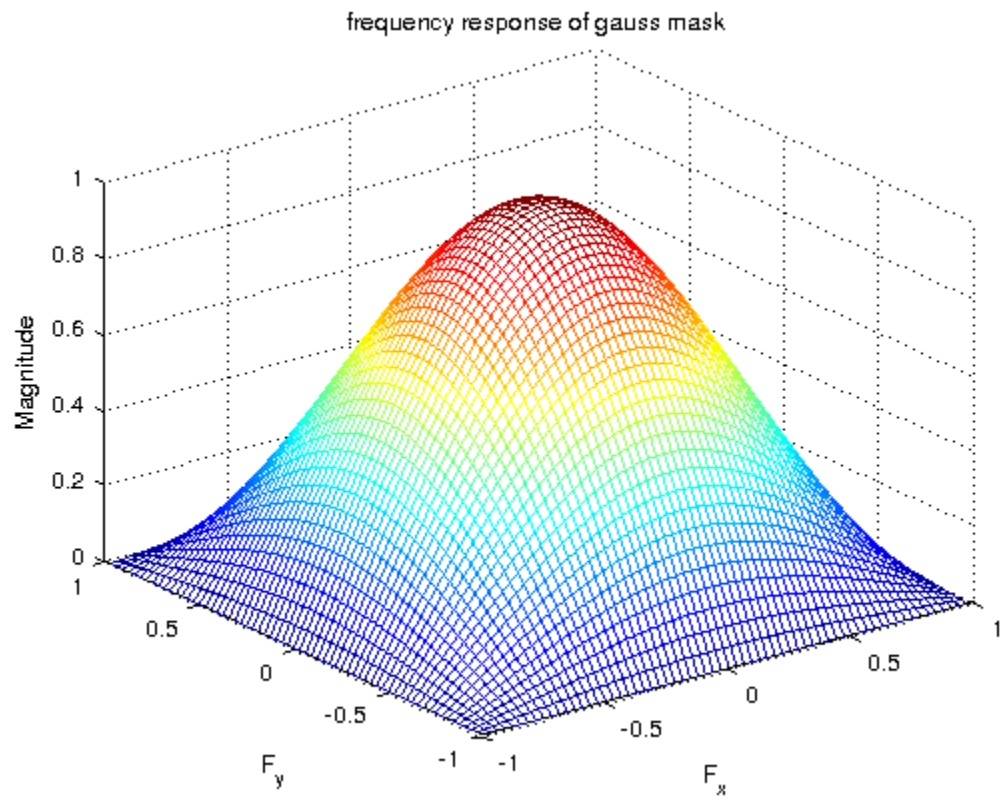
```
ans =
```

```
comapring my gauss filter vs builtin: h-H:
```

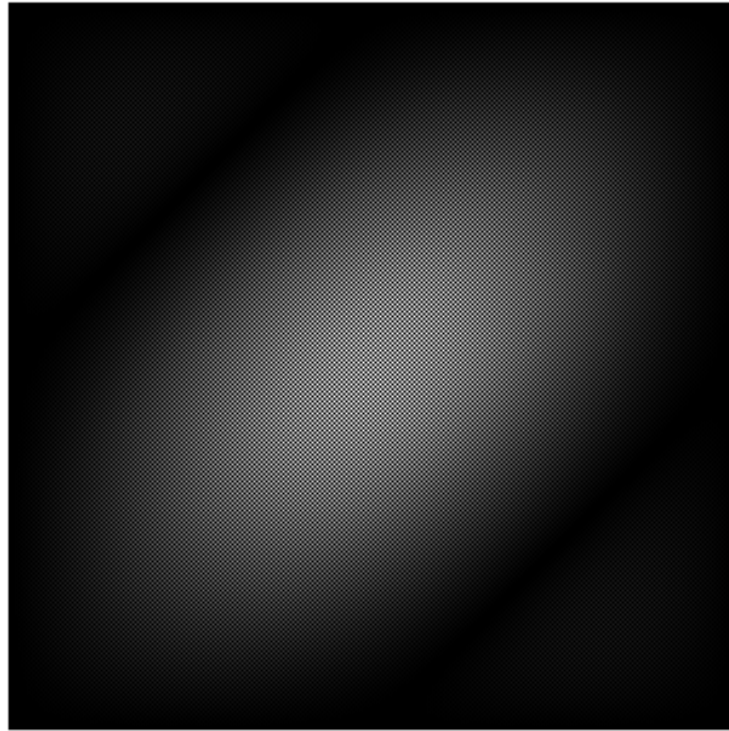
```
gauss_difference =
```

```
0    0    0
0    0    0
0    0    0
```

```
Warning: Displaying real part of complex input.
```



shiftedfft



original image



gauss noise added to img



salt & pepper added to img



gauss filtered image2



gauss filtered image3



median filtered image3



median filtered image3



Published with MATLAB® R2013b