ECSE 4530 Image Processing

HW 4

Feb 20, 2018
Daniel Southwick
661542908

```
%% ==================================4.1====================================
```

Main Function

```
%% 1a
function fftshow(im)
figure
imshow(fftshift(log(abs(fft2(im)))),[]);
end
```

Main Function

```
%% 1b
function imfilt = boxfilt(im,w,h)
[a,b] = size(im);
filter = zeros(a,b);
filter(((a-h)/2+1):((a+h)/2),((b-w)/2+1):((b+w)/2))=ones(h,w);
imfilt = fftshift(filter);
end
```

Main Function

```
%% 1c
function imfilt = freqflt(im, filt)
mult = fft2(im) .* (filt);
imfilt = real(ifft2(mult));
end
```

Main Function

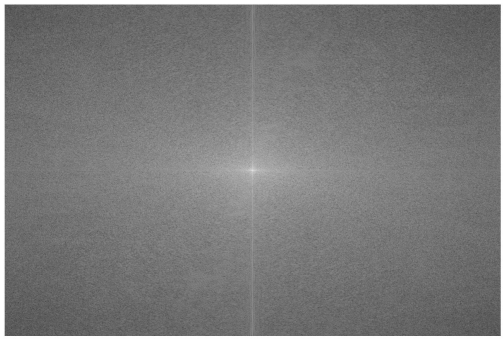
```
%% 1d
function imfilt = guassflt(im, filt)
[a,b] = size(im);
[x,y] = meshgrid(1:b,1:a);
imfilt = exp(-0.5*sum([x(:)-(b+1)/2 y(:)-(a+1)/2].^2 *inv(filt),2));
imfilt = reshape(imfilt, [a b]);
maxi = max(imfilt(:));
imfilt = fftshift(imfilt/maxi);
end
```

```
%% ==================================4.2==================================
```

Main Function

```
%% 2a
tiger=imread('tiger.png');
fftshow(tiger);
```

Matlab Result

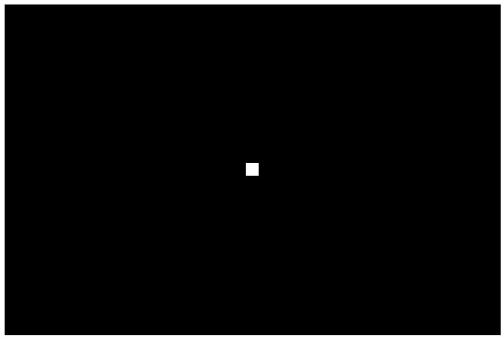


(2D DFT of the image)

Main Function

```
%% 2b
boxfilt_b = boxfilt(tiger,31,31);
imshow(boxfilt_b,[]);
```

Matlab Result



(Frequency domain of the filter)

Main Function

```
%% 2c
tiger_2c = freqflt(tiger,boxfilt_b);
imshow(tiger_2c,[]);
```

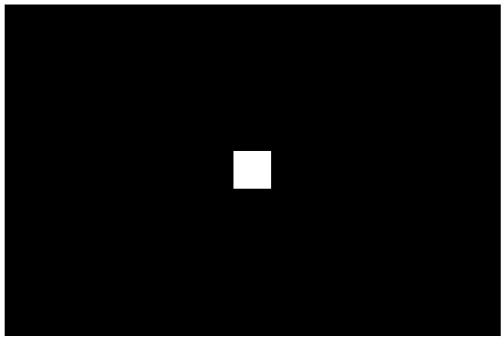
Matlab Result

After applying the box filter [31, 31] to the Tiger picture, we can see it is blurred evenly on both the horizontal and vertical axis.

Main Function

```
%% 2d
boxfilt_d = boxfilt(tiger,91,91);
imshow(uint8(boxfilt_d),[]);
tiger_2d = freqflt(tiger,boxfilt_d);
imshow(tiger_2d,[]);
```

Matlab Result



(Frequency domain of the filter)

Result is shown above after the box filter [91,91] has been applied to the picture, compare with the [31,31] box filter in part a,b,c, we can see that the image also have been blurred on both direction, but the blurring effect is much less than [31,31], we can still somehow see the tiger clearly.

Main Function

```
%% 2e
boxfilt_e = boxfilt(tiger,91,31);
tiger_2e = freqflt(tiger,boxfilt_e);
imshow(tiger_2e,[]);
```

Matlab Result

Result is shown above after the box filter [91,31] has been applied to the picture. The higher the number that box filter has, more original image shall pass the filter, and thus the image should be less blurred. So [91,31], The image should be blurred on the "height" a lot more than the "width", this is shown in the result from the almost visual visible horizontal lines across the picture.

Main Function

```
%% 2f
boxfilt_f = boxfilt(tiger,91,399);
tiger_2f = freqflt(tiger,boxfilt_f);
imshow(tiger_2f,[]);
```

Matlab Result

Result is shown above after the box filter [91,399] has been applied to the picture. Since higher number will make the filtered picture clear, so the result is expected, the image should be blurred on the "width" a lot more than the "height", this is shown in the result from the almost visible vertical blurry lines coming out of the tiger.

Main Function

```
%% 2g
boxfilt_g = boxfilt(tiger,599,31);
tiger_2g = freqflt(tiger,boxfilt_g);
imshow(tiger_2g,[]);
```

Matlab Result

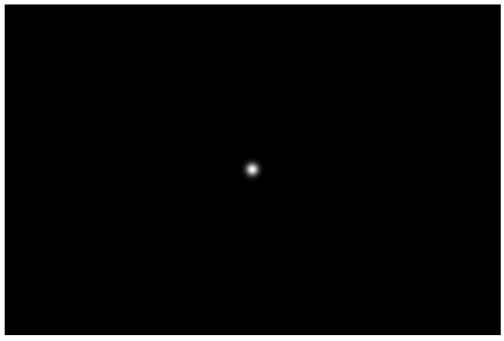
Result is shown above after the box filter [599, 31] has been applied to the picture. Though the “width” number is really high on this one, which means the vertical information’s should be really close or look exactly the same as the original. However, the “height” number is really small, image is blurred a lot more on the horizontal, and is blurred to a point that our visual system thinks the picture is more blurry than 2f as a whole.

```
%% ==================================4.3==================================
```

Main Function

```
%% 3a
C1 = [100,0; 0,100];
guassfilt_a = guassflt(tiger,C1);
imshow(fftshift(guassfilt_a));
```

Matlab Result



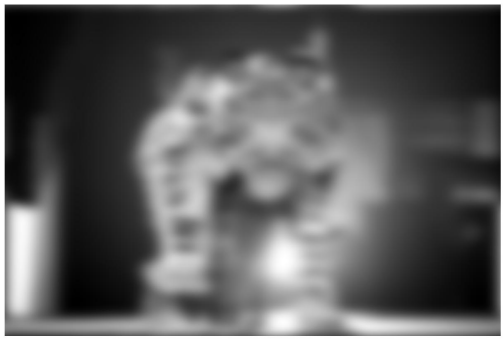
(Frequency domain of the filter)

The Gaussian filter in the frequency domain is shown above, comparing to the box filter in Problem 2b, this filter is a lot smoother around the edges, around the DC component, so we should see a smooth blurry across the image overall comparing to the box filters.

Main Function

```
%% 3b
tiger_3b = freqflt(tiger,guassfilt_a);
imshow(tiger_3b,[]);
```

Matlab Result



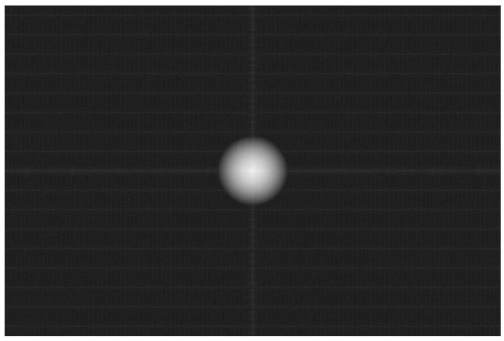
(Spatial domain of the filtered Tiger)

The filtered picture using our Gaussian filter with covariance `[100,0; 0,100]` in the spatial domain is shown above, as predicted, the image is blurred evenly across the whole image and also there's no visible vertical or horizontal lines like the box filter. Since all Gaussian filter in the frequency domain have smooth edges around DC, thus the spatial filtered result is evenly blurred.

Main Function

```
%% 3c
fftshow(tiger_3b);
```

Matlab Result

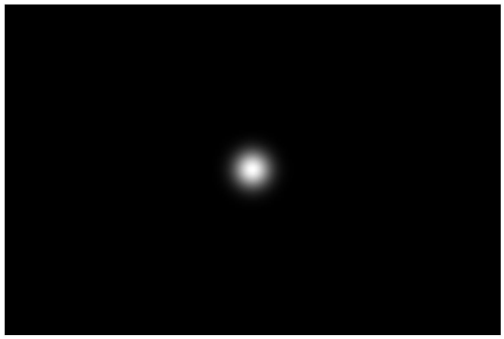


(Frequency domain of the filtered Tiger)

Main Function

```
%% 3d
C2 = [900,0; 0,900];
guassfilt_d = guassflt(tiger,C2);
imshow(fftshift(guassfilt_d));
tiger_3d = freqflt(tiger,guassfilt_d);
imshow(tiger_3d,[]);
fftshow(tiger_3d);
```

Matlab Result



(Frequency domain of the filter)

(Spatial domain of the filtered Tiger)

The frequency domain of the filter and the spatial domain of the filtered Tiger using our Gaussian filter with covariance `[900,0; 0,900]` are shown above. Comparing to 3a, the Gaussian filter in the frequency domain has a larger white circle centered around DC, this makes sense since the covariance of the Gaussian is large, thus the Gaussian width is large and more information from the original picture can pass through the filter. Thus this is a lot less filtered, a lot more clearer than part a,b,c.

Main Function

```
%% 3e
C3 = [900,0; 0,100];
guassfilt_e = guassflt(tiger,C3);
imshow(fftshift(guassfilt_e));
tiger_3e = freqflt(tiger,guassfilt_e);
imshow(tiger_3e,[]);
fftshow(tiger_3e);
```

Matlab Result

(Frequency domain of the filter)

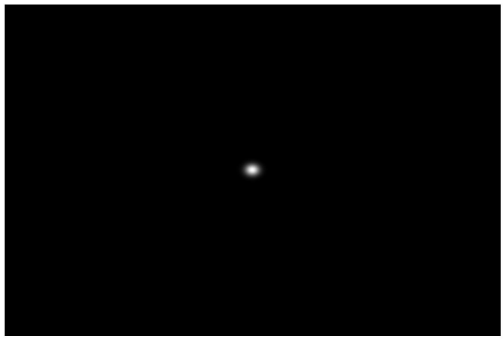
(Spatial domain of the filtered Tiger)

The frequency domain of the filter and the spatial domain of the filtered Tiger using our Gaussian filter with covariance `[900,0; 0,100]` are shown above. Different from the previous two filters, this Gaussian has covariance of 900 on the "width" and 100 on the "height"; this is shown from the frequency domain of the filter, the whole filter is in the shape of an ellipse instead of a circle. Thus the Tiger should be filtered a lot more on the horizontal than the vertical. We can clearly see that from the spatial domain result above, a smooth filter with horizontal blurred a lot more than the vertical.

Main Function

```
%% 3f
R = [cos(pi/6), -sin(pi/6); sin(pi/6), cos(pi/6)];
D = [900, 0; 0 100];
C4 = transpose(R)*D*R;
guassfilt_f = guassflt(tiger,C4);
imshow(fftshift(guassfilt_f));
```

Matlab Result



The frequency domain of the picture is shown above, it looks like a tilted ellipse. Vector R normalized D with an angle, the center ellipse appeared to be smaller and almost evenly distributed.

Main Function

```
%% 3g
tiger_3f = freqflt(tiger,guassfilt_f);
imshow(tiger_3f,[]);
```

Matlab Result

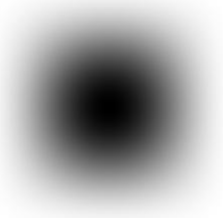
The filtered Tiger in the spatial domain is shown above, since the filter in the frequency domain is tilted, so the blurry direction is also tilted in the result. We can see the difference if we compare this to problem 3e, The image became more blurry but it has evenly fuzziness across the whole picture.

```
%% ==================================4.4==================================
```

Main Function

```
%% 4a
H = [-1,-2,-1;-2,28,-2;-1,-2,-1]./16;
Y = abs(fft2(H,256,256));
imshow(fftshift(Y),[]);
```

Matlab Result



The result of the magnitude 2D 256x256 DFT of the given filter is shown below. While a low-pass filter has a white center a black outer region, this high-pass filter is nearly the opposite.

Main Function

```
%% 4b
Y2 = filter2(H,tiger);
imshow(uint8(Y2));
```

Matlab Result

Quite clearly the filtered image is much sharper compare to the original picture, specifically around the edges of the tiger. Some of the natural ‘blur’ surrounding the figure have been increased and it is visually easier to recognize the tiger and more appealing to look at it. Thus, the given filter is edge-boosting.

4c)

$$h(x,y) = \frac{1}{16}\begin{bmatrix} -1 & -2 & -1 \\ -2 & 28 & -2 \\ -1 & -2 & -1 \end{bmatrix}$$

$$H(u,v) = \sum_{x=1}^{3}\sum_{y=1}^{3} h(x,y)\, e^{-j\frac{2\pi}{3}(ux+vy)}$$

$$= \frac{28}{16} - \frac{2}{16}\left(e^{-j\frac{2\pi}{3}v} + e^{j\frac{2\pi}{3}v} + e^{-j\frac{2\pi}{3}u} + e^{j\frac{2\pi}{3}u}\right)$$
$$- \frac{1}{16}\left(e^{-j\frac{2\pi}{3}(u+v)} + e^{j\frac{2\pi}{3}(u+v)} + e^{-j\frac{2\pi}{3}(u-v)} + e^{j\frac{2\pi}{3}(u-v)}\right)$$

$$= \frac{28}{16} - \frac{2}{16}\left(2\cos\left(\frac{2\pi}{3}v\right) + 2\cos\left(\frac{2\pi}{3}u\right)\right) - \frac{1}{16}\left(2\cos\left(\frac{2\pi}{3}(u+v)\right) + 2\cos\left(\frac{2\pi}{3}(u+v)\right)\right)$$

$$= \frac{7}{4} - \frac{1}{4}\left(\cos\left(\frac{2\pi}{3}u\right) + \cos\left(\frac{2\pi}{3}v\right)\right) - \frac{1}{4}\left(\cos\left(\frac{2\pi}{3}u + \frac{2\pi}{3}v\right)\right)$$

$$= \frac{1}{4}\left(7 - \cos\left(\frac{2\pi}{3}u\right) - \cos\left(\frac{2\pi}{3}v\right) - \cos\left(\frac{2\pi}{3}u\right)\cdot\cos\left(\frac{2\pi}{3}v\right)\right)$$

Plug in (u,v)

$$H(u,v) = \frac{1}{16}\begin{bmatrix} 31 & 28 & 31 \\ 28 & 16 & 28 \\ 31 & 28 & 31 \end{bmatrix}$$