Table of Contents

Mayank Vanjani

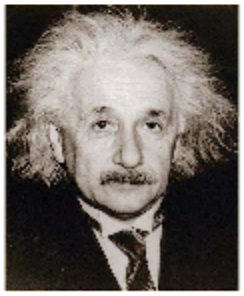
Signals, Systems, & Transform, Hybrid Images Lab 10/19/18

```
clc
clear
```

Load images:

```
%pair = 'cat&dog';
%Img1 = imread('1_dog.bmp');
%Img2 = imread('1_cat.bmp');

%Img1 = imread('2_bicycle.bmp');
%Img2 = imread('2_motorcycle.bmp');

%Img1 = imread('3_bird.bmp');
%Img2 = imread('3_plane.bmp');

%Img1 = imread('4_fish.bmp');
%Img2 = imread('4_submarine.bmp');

Img1 = imread('5_marilyn.jpeg');
Img2 = imread('5_einstein.jpeg');

%Img1 = imread('6_reddragon.jpg');
%Img2 = imread('6_whitedragon.jpg');

% To process images, change data type
% Signal ranges from 0 - 1 with double precision
Img1 = double(Img1)./255;
Img2 = double(Img2)./255;

[m, n, ~] = size(Img1); % Don't need third dimension

figure(1)
clf
subplot(2,1,1)
imshow(uint8(Img1.*255)) % Recover division from before and double

subplot(2,1,2)
```

```
imshow(uint8(Img2.*255))
```

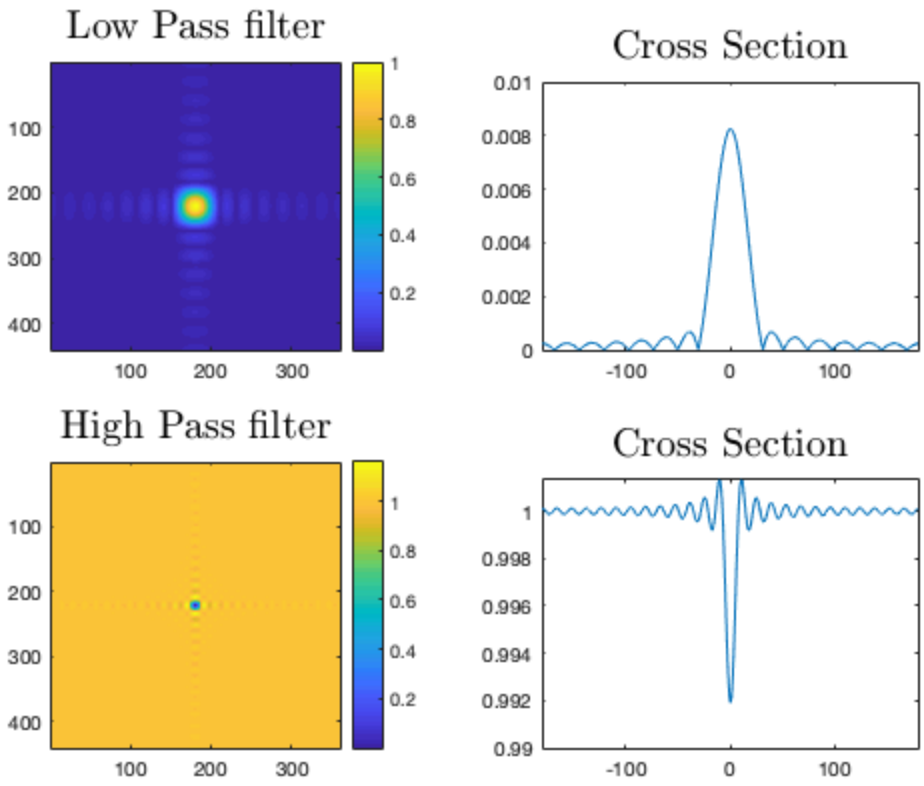


Prepare a filter

Odd Length filter because we can have a center

```
% Low-pass filter
sigma_lp = 5;         % standard deviation, changes Gaussian cutoff
hsize_lp = 15;        % filter size
% h_lp     = fspecial('average',hsize_lp);         % Average filter
h_lp     = fspecial('gaussian',hsize_lp,sigma_lp); % Gaussian filter
 (better)
H_lp     = fft2(h_lp,m,n);              % Frequency response (2D,
 generally complex)

% High-pass filter
sigma_hp = 30;
hsize_hp = 51;
h_I = zeros(hsize_hp);
xc  = (hsize_hp+1)/2;
h_I(xc, xc) = 1;

% Continue the process of making a high-pass filter
h_lph = fspecial('gaussian',hsize_hp,sigma_hp);
h_hp = h_I - h_lph;             % High Pass = All Pass - Low Pass
H_hp = fft2(h_hp,m,n);
```

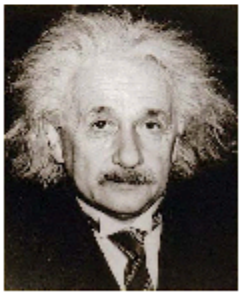
```
figure(2)
clf
subplot(2,2,1)
% abs because magnitude of frequency response of complex number
imagesc(abs(fftshift(H_lp)))
colorbar
title('Low Pass filter','FontSize',20,'Interpreter','latex')

subplot(2,2,2)
% Cross section: 2D resolves to 1D filter
plot(-(n-1)/2:(n-1)/2,abs(fftshift(H_lp(175,:))))
xlim([-(n-1)/2, (n-1)/2])
title('Cross Section','FontSize',20,'Interpreter','latex')
% xlim([-(n-1)/2, (n-1)/2])

subplot(2,2,3)
imagesc(abs(fftshift(H_hp)))
colorbar
title('High Pass filter','FontSize',20,'Interpreter','latex')

subplot(2,2,4)
plot(-(n-1)/2:(n-1)/2,abs(fftshift(H_hp(175,:))))
xlim([-(n-1)/2, (n-1)/2])
title('Cross Section','FontSize',20,'Interpreter','latex')
% xlim([-(n-1)/2, (n-1)/2])
```

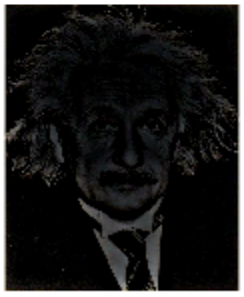


Filtered Image

Apply the filters on images

```
Img_lp = imfilter(Img1, h_lp);    % Low frequencies
Img_hp = imfilter(Img2, h_hp);    % High frequencies

figure(3)
clf
subplot(2,2,1)

% Show the original images and the filtered images
imshow(uint8(Img1.*255));
subplot(2,2,2)
imshow(uint8(Img_lp.*255));
subplot(2,2,3)
imshow(uint8(Img2.*255));
subplot(2,2,4)
imshow(uint8(Img_hp.*255));
```

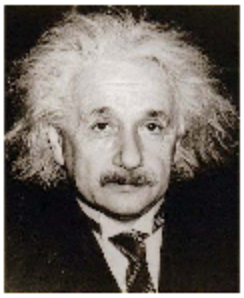


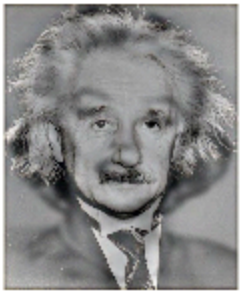


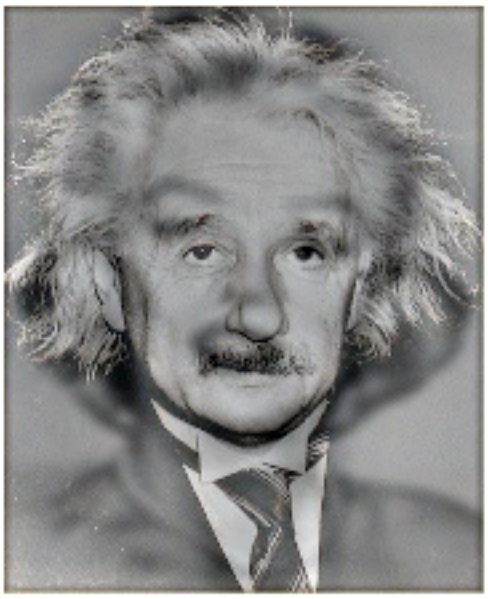
Hybrid Image

```
% Make a Hybrid image with a linear combination
% of filtered images
```

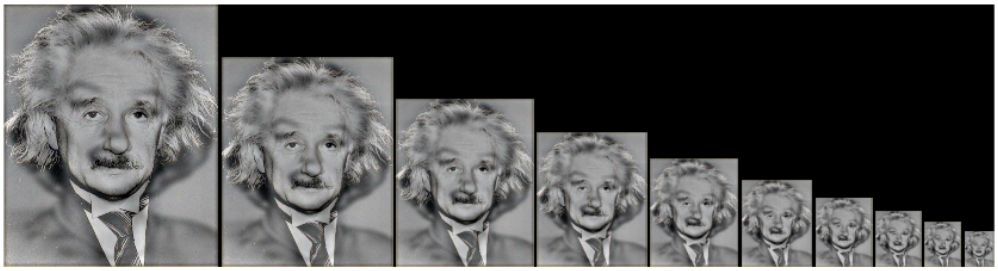
```
scale = 1.5;
alpha = 0.5;

% Hybrid_image = scale*(alpha*low_freq_image + (1-
alpha)*high_freq_image)

Img_H = scale*(alpha*Img_lp + (1-alpha)*Img_hp);

figure(4)
clf
subplot(2,2,1)
% First image
imshow(uint8(Img1.*255));

subplot(2,2,2)
% Second image
imshow(uint8(Img2.*255));

subplot(2,2,[3 4])
% Hybrid image
imshow(uint8(Img_H.*255));

figure(5)
clf
% Hybrid image
imshow(uint8(Img_H.*255));

vis = vis_hybrid_image(uint8(Img_H.*255));

figure(6)
clf
imshow(vis)
title('Hybrid Image','FontSize',20,'Interpreter','latex')
```







Hybrid Image



Frequency Perspective (Fourier Transform)

Complete section, Nothing to be coded ! Focus on understanding and interpretations

```
channel = 1;

F_img1  = fftshift(log(abs(fft2(Img1(:,:,channel).*255))));
F_img2  = fftshift(log(abs(fft2(Img2(:,:,channel).*255))));

F_imglp = fftshift(log(abs(fft2(Img_lp(:,:,channel).*255))));
F_imghp = fftshift(log(abs(fft2(Img_hp(:,:,channel).*255))));
F_imgH  = fftshift(log(abs(fft2(Img_H(:,:,channel).*255))));


figure(7)
clf
subplot(2,4,1)
imagesc(F_img1)
colorbar
title('Image 1','FontSize',20,'Interpreter','latex')

subplot(2,4,2)
imagesc(abs(fftshift(H_lp)))
colorbar
title('Low Pass filter','FontSize',20,'Interpreter','latex')

subplot(2,4,3)
imagesc(F_imglp)
colorbar
title('Low-Freq Image 1','FontSize',20,'Interpreter','latex')

subplot(2,4,4)
imagesc(F_img1 - F_imglp)
colorbar
title('Diff Image 1','FontSize',20,'Interpreter','latex')


subplot(2,4,5)
imagesc(F_img2)
colorbar
title('Image 2','FontSize',20,'Interpreter','latex')
```

```
subplot(2,4,6)
imagesc(abs(fftshift(H_hp)))
colorbar
title('High Pass filter','FontSize',20,'Interpreter','latex')


subplot(2,4,7)
imagesc(F_imghp)
colorbar
title('High-Freq Image 2','FontSize',20,'Interpreter','latex')

subplot(2,4,8)
imagesc(F_img2 - F_imghp)
colorbar
title('Diff Image 2','FontSize',20,'Interpreter','latex')


figure(8)
clf
subplot(2,2,1)
imagesc(F_imglp)
colorbar
title('Low-Freq Image 1','FontSize',20,'Interpreter','latex')

subplot(2,2,2)
imagesc(F_imghp)
colorbar
title('High-Freq Image 2','FontSize',20,'Interpreter','latex')

subplot(2,2,[3 4])
imagesc(F_imgH)
colorbar
title('Hybrid Image','FontSize',20,'Interpreter','latex')
```

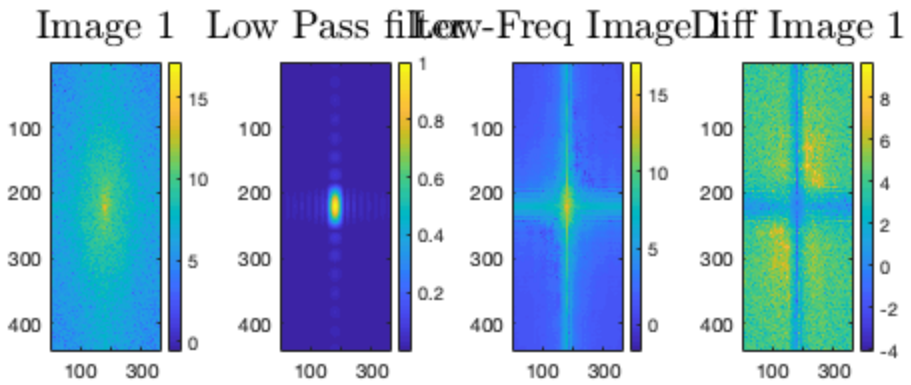


Image 1
Low Pass filter
Low-Freq Image
Diff Image 1

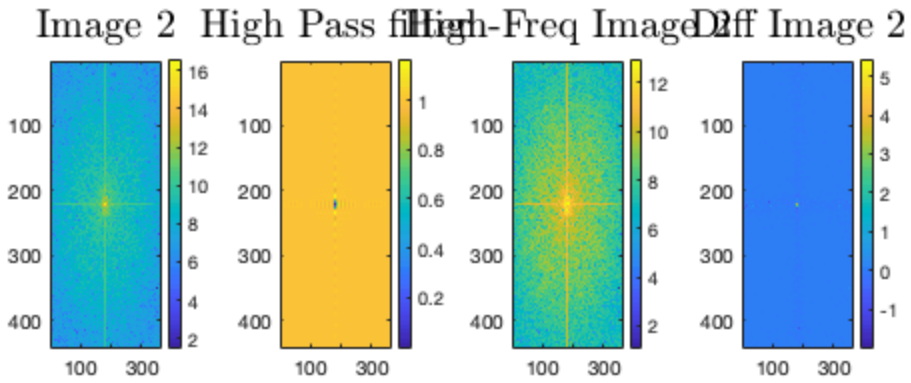
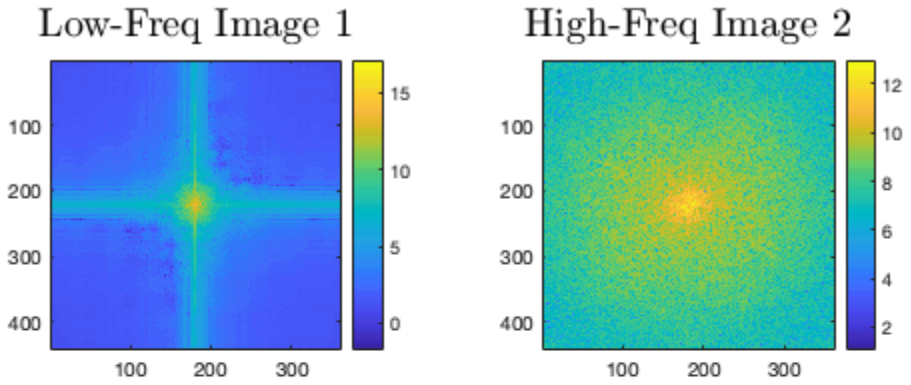
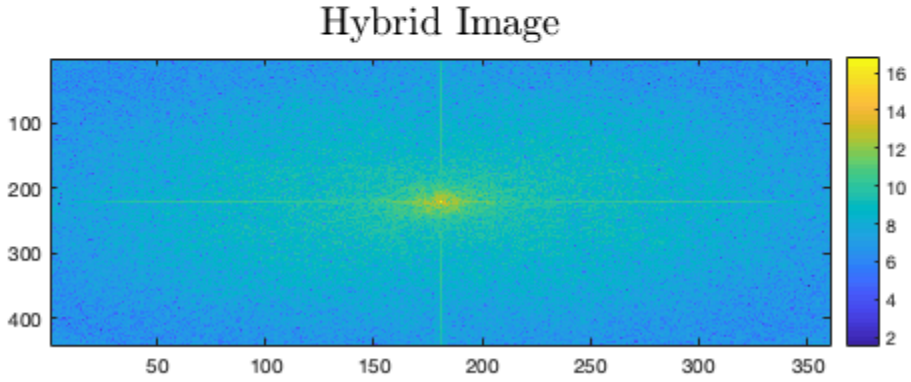


Image 2
High Pass filter
High-Freq Image
Diff Image 2


Low-Freq Image 1
High-Freq Image 2


Hybrid Image

Answers

```
%{
1: Load Images and Create Filter
The standard deviation from the frequency response of the filter
 determines
the Gaussian cutoff frequency point of the filter (lpf in this case).
 The
filter size determines how many samples are taken for the filter.

2: Filter Images and Create a Hybrid Image
Created own hybrid image as image 5 (Marilyn Monroe and Albert
 Einstein)
and image 6 (Red Dragon and White Dragon on a Cliff).

3: Frequency Perspective
    1) The dual frequency operation to the spatial 2D convolution is
        multiplication
    2) The difference between the original and filtered images are the
        blurring factor. The images that are put under the low pass
 filter
        act as averages of the nearby pixels resulting in a blur of
 the
        image overall. The image put through the high pass filter only
        records the sudden changes of color resulting in only outlines
 and
        sudden color changes present. The low pass image is better
        perceived from farther distances because the signal travels
 farther
        whereas the high pass image is easily visible near the image
 (like
        on the laptop we work on).
    3) A new way of computing convolution would be to convert the time
 into
        frequency and performing multiplication; the result should
 then be
        put back to time using the inverse Fourier transform.
 Generally,
        this is better for more complex signals since simple signals
 can
        have convolution done manually.

I don't know the grades of my lab reports and quiz yet so I'm not sure
where to apply the extra credit. Probably the MATLAB quiz.
%}
```

Published with MATLAB® R2018a