Assignment 1
Submitted by
Adarsh Ghimire
Student ID – 100058927

In first part of this assignment a code has been written that takes two images ***low_img*** and ***high_img***, apply lowpass Gaussian filter on ***low_img*** and highpass Gaussian filter on ***high_img*** and combined the output of both filters, the result will be the image ***combined_img*** with the overlook of ***high_img*** and the details of ***low_img*** in such a way if you looked at it from faraway you will see image ***high_img*** and if you looked at it up close you will see image ***low_img***.

Below, the code that has been written for creating the hybrid images will be explained. Then later, different parameters that were tested will be briefed.

The gaussian low pass filter function(**low_pass_filter.m**) has been created in the code base which takes in the input the image, filter size and sigma. And returns the filtered image.

```
function [img_fil] = low_pass_filter(img, hs, sigma)

fil = fspecial('gaussian', 2*hs+1, sigma); %size of the filter is hs*2+1 (odd)

fftsize = 1024; % preferred be be a power of 2 (for speed) and include padding
img_fft = fft2(img, fftsize, fftsize); % 1) fft im with padding; automatically zero-pads
fil_fft = fft2(fil, fftsize, fftsize); % 2) fft fil, pad to same size as image
img_fil_fft = img_fft .* fil_fft; % 3) multiply fft images
img_fil = ifft2(img_fil_fft, 'symmetric'); % 4) inverse fft2
img_fil = img_fil(1 + hs:size(img, 1) + hs, 1 + hs:size(img, 2) + hs); % 5) remove padding
```

The high pass filtering operation using gaussian filter is obtained by passing the image through the low pass filter and then subtracting the result of low pass filter from the original image.

The hybrid image formation takes place in the **combined_code.m** which basically does the following:

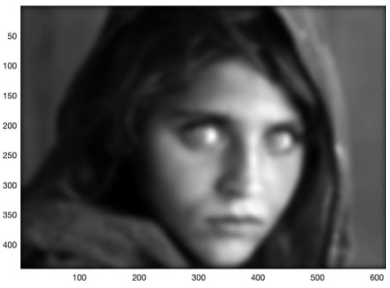
1. First loads the images: Image 1 and Image 2
2. Converts the images from RGB to Gray
3. Does low pass filtering on image 1
4. Does high pass filter on image 2
    A. First apply Low pass filter on image 2
    B. Subtract low pass filtered image 2 from original image 2
5. Combines the result of two filtered image to obtain the hybrid image

```
% Combined low pass and high pass filter

clc;
clear all;
high_img = imread('tiger.jpeg');
low_img = imread('girl.jpeg');

low_img = rgb2gray(low_img);    %Use of the image is in color
low_img = double(low_img);

high_img = rgb2gray(high_img);    %Use of the image is in color
high_img = double(high_img);

[imgW, imgH] = size(low_img);
high_img = imresize(high_img, [imgW, imgH]);
figure(1), imagesc(low_img), axis image, colormap gray
figure(2), imagesc(high_img), axis image, colormap gray

sigma_low = 5;
sigma_high = 10;
low_img_fil = low_pass_filter(low_img, 15, sigma_low);
figure(3), imagesc(low_img_fil), axis image, colormap gray

high_img_fil_low = low_pass_filter(high_img, 15, sigma_high);
figure(4), imagesc(high_img_fil_low), axis image, colormap gray

high_img_fil = high_img - high_img_fil_low;
figure(5), imagesc(high_img_fil), axis image, colormap gray

alpha = 0.5;
combined_image = alpha*low_img_fil + (1-alpha)*high_img_fil;
figure(6), imagesc(combined_image), axis image, colormap gray
```

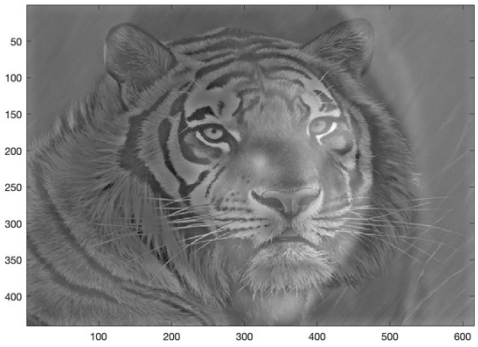
Below is the result of doing the above experiment.

Source Image 1

Low Pass, sigma = 5

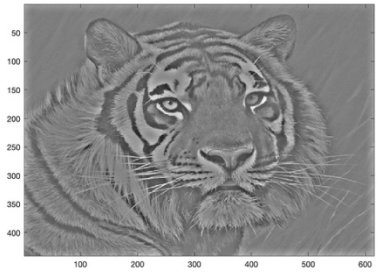


Hybrid

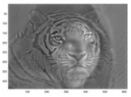


Source Image 2

High Pass, sigma = 10



Hybrid



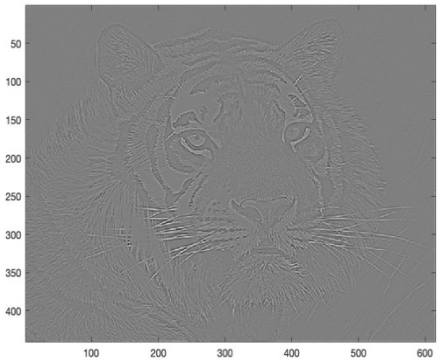
<u>Showing the different parameter settings:</u>

Varying the Sigma

For sigma = 1

Low Pass, sigma = 1

High Pass, sigma = 1

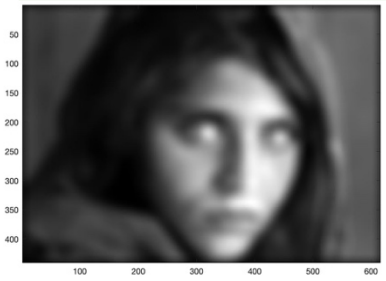


Result

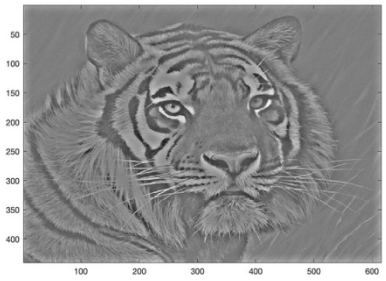
Here, in the result we can see that for low value of sigma the high pass result is bad however the low pass result is good. And the result in combined image also has very less effect of high pass filter thus the effect is not good.

For sigma = 10

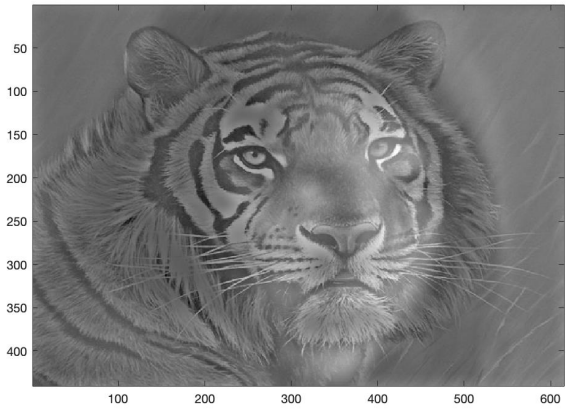
Low Pass, sigma = 10

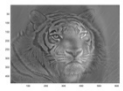


High Pass, sigma = 10



Result





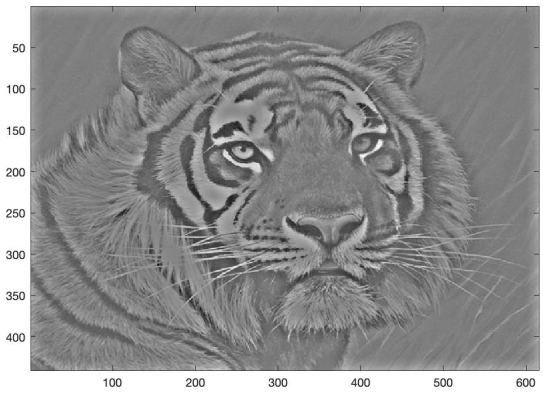
Here, in the result we can see that for high value of sigma the high pass result is good however the low pass result is bad since at lower resolution the girl is not seen properly and more of tiger is seen in low resolution. And the result in combined image also has very less effect of low pass filter thus the effect is not good.

So, looks like low pass with sigma of 1 and high pass of 10 will give a good result. (it is an assumption which is tested below:

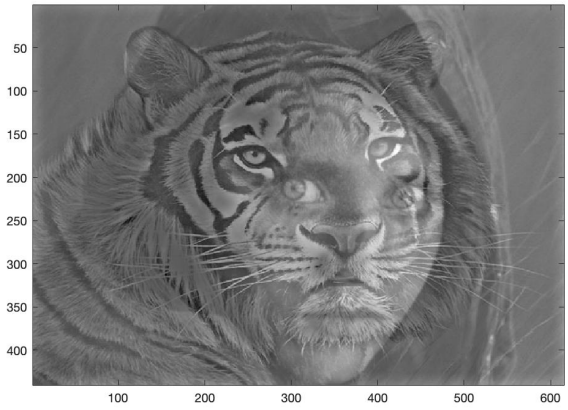
For sigma = 1 for low pass and 10 for high pass

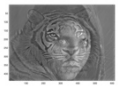
Low Pass, sigma = 1

High Pass, sigma = 10



Result





Here, in the result we can see that for high value of sigma the high pass result is good, and the low pass result is also good. However, the result in combined image also has effect of low pass filter on higher resolution thus the effect is not good.

Thus, from all these experiments we found that optimum value of sigma for low pass filter is 5 and high pass filter is 10 which gave the good result.

Doing the high pass filtering operation using Laplacian.

The combined_gauss_lap.m first filters the image 1 with gaussian low pass filtering. And for high pass operation the Laplacian from fspecial is used. The code snippet is shown below.

*The code below is used for high pass filtering using Laplacian(***high_pass_filter.m***)*

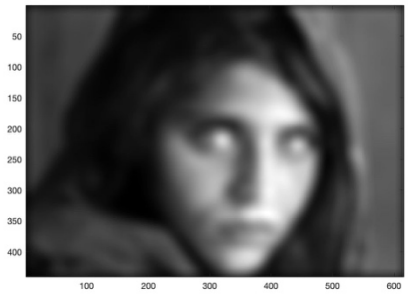
```
% high pass filter using laplacian
function [img_fil] = high_pass_filter(img, hs, alpha)

fil = fspecial('laplacian', alpha); %size of the filter is hs*2+1 (odd) and sigma=1


fftsize = 1024; % preferred be be a power of 2 (for speed) and include padding
img_fft = fft2(img, fftsize, fftsize); % 1) fft im with padding; automatically zero-pads
fil_fft = fft2(fil, fftsize, fftsize); % 2) fft fil, pad to same size as image
img_fil_fft = img_fft .* fil_fft; % 3) multiply fft images
img_fil = ifft2(img_fil_fft, 'symmetric'); % 4) inverse fft2
img_fil = img_fil(1 + hs:size(img, 1) + hs, 1 + hs:size(img, 2) + hs); % 5) remove padding
```

The below code shows the implementation of gaussian low pass filter combined with Laplacian high pass filtering. **(combined_gauss_lap.m***)*

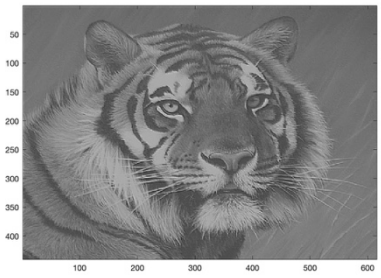
```
% Combined with low pass filter using gaussian and high pass filter with laplacian filter

clc;
clear all;
high_img = imread('tiger.jpeg');
low_img = imread('girl.jpeg');

low_img = rgb2gray(low_img);    %Use of the image is in color
low_img = double(low_img);

high_img = rgb2gray(high_img);    %Use of the image is in color
high_img = double(high_img);

[imgW, imgH] = size(low_img);
high_img = imresize(high_img, [imgW, imgH]);
% figure(1), imagesc(low_img), axis image, colormap gray
figure(2), imagesc(high_img), axis image, colormap gray

low_img_fil = low_pass_filter(low_img, 15, 5);
figure(3), imagesc(low_img_fil), axis image, colormap gray

high_img_fil = high_pass_filter(high_img, 0, 0.1);

high_img_fil = high_img - high_img_fil;
figure(4), imagesc(high_img_fil), axis image, colormap gray

alpha = 0.75;
combined_image = alpha*low_img_fil + (1-alpha)*high_img_fil;
figure(5), imagesc(combined_image), axis image, colormap gray
```

Experiment using low pass with sigma of 5 and high pass filtering using Laplace filter

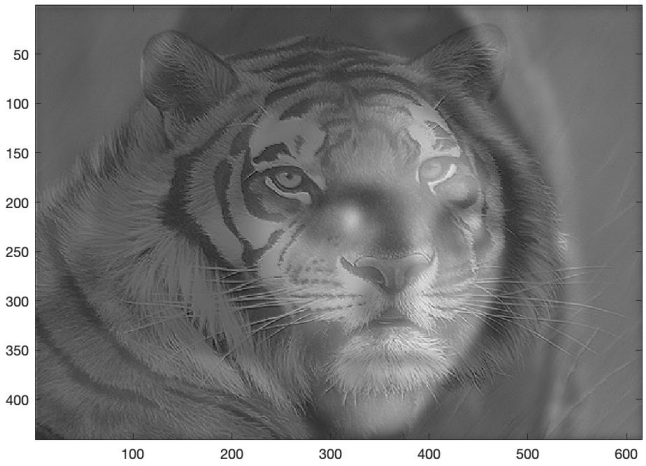
Low Pass, sigma = 5



Laplacian High Pass



Result



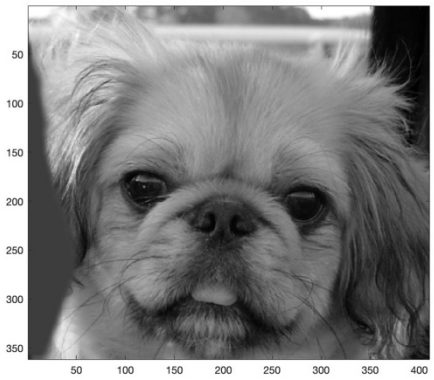
Here, in the result we can see that Laplacian filter gave good high pass result which when combined with gaussian low pass filter (sigma set to 5) resulted in very good optical illusion. Which can be seen in the result above, when at higher resolution the tiger is seen very clearly and when at lower resolution the girl is seen very clearly. The Laplacian filter has been able to capture the finer details resulting in good performance.

Experiment with Different set of images.

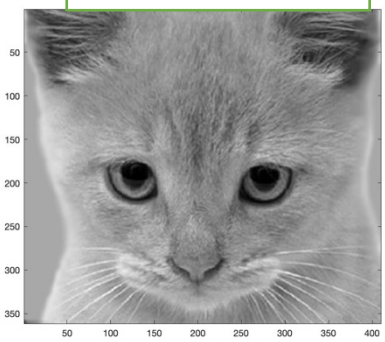
1. Cat image is used with high pass filtering because it has finer details then dog image.
2. Dog image is used with low pass filtering

Using gaussian low and high pass filter.

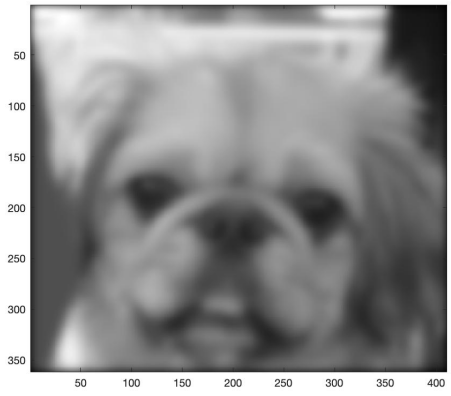
Source Image 1



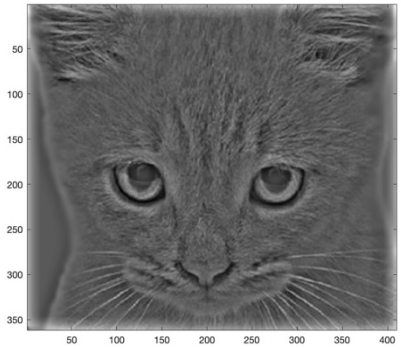
Source Image 2



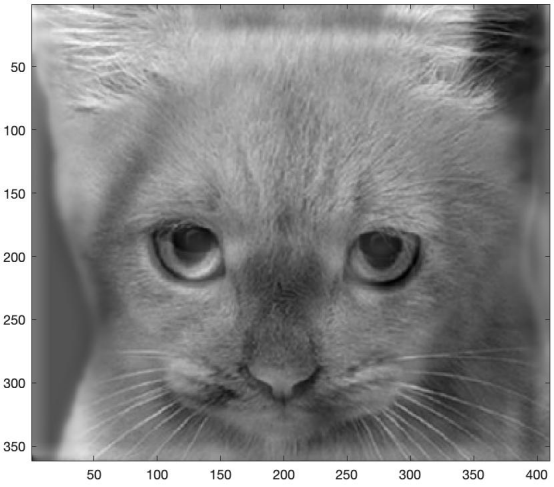
Low Pass, sigma = 5



High Pass, sigma = 10

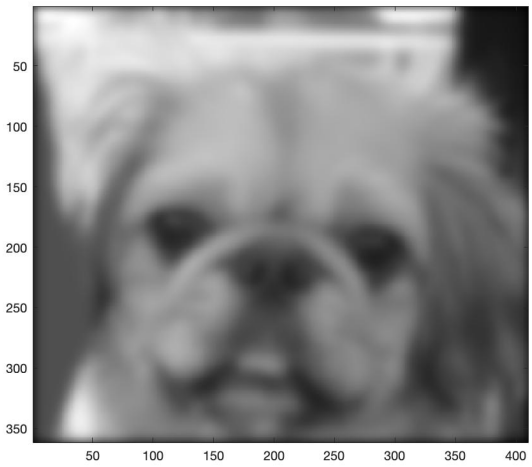


Result

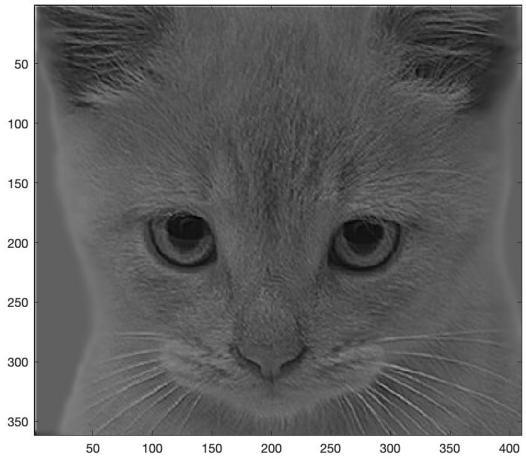


Using Gaussian Low and Laplacian high pass filter.

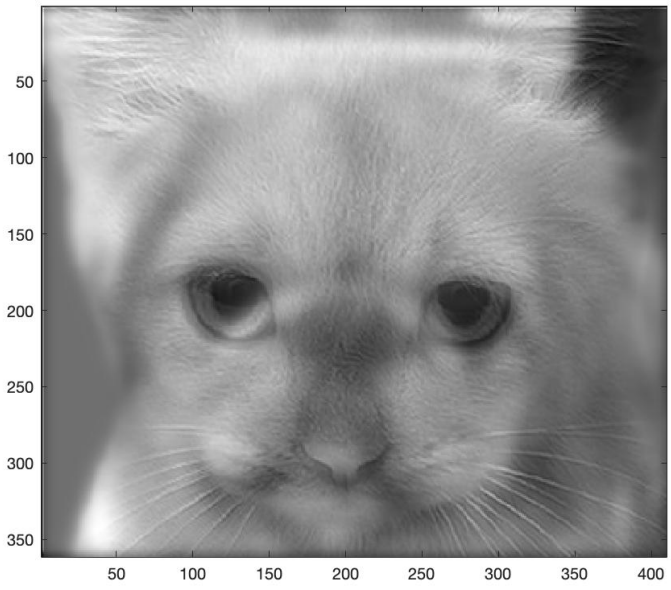
Low Pass, sigma = 5

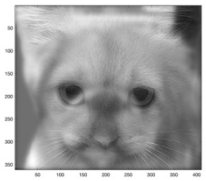


High Pass using Laplace



Result





In this result as well, when we compare the gaussian_gaussian vs the gaussian_laplace result the Laplace result is much good. Because at higher resolution the cat image is very clear in both the results however the low-resolution image with dog the result from Laplacian filtering is much good then gaussian only since we can see the dog at much higher low resolution of the image while the gaussian one required making the image to very very low resolution to obtain the dog image.