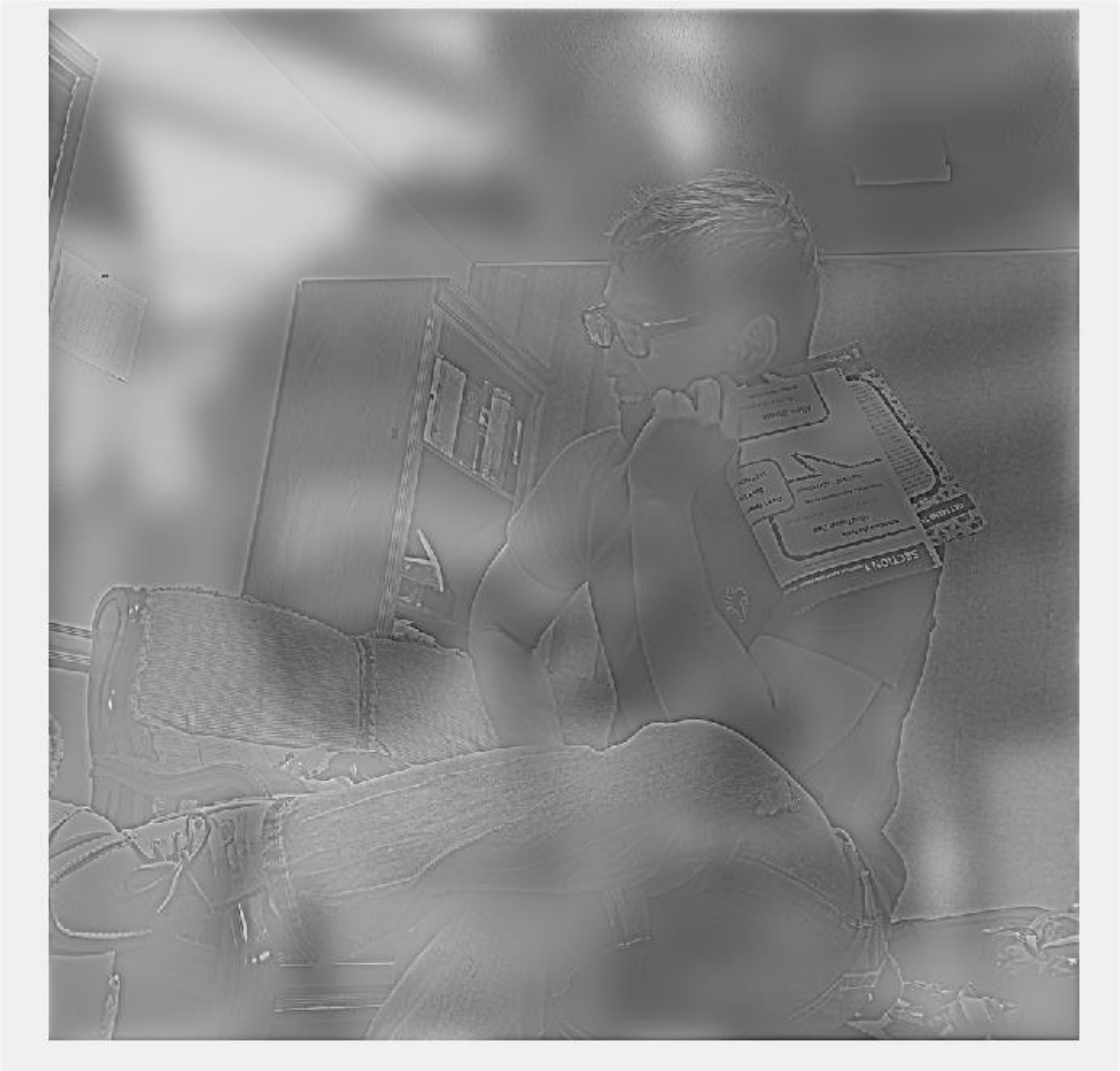
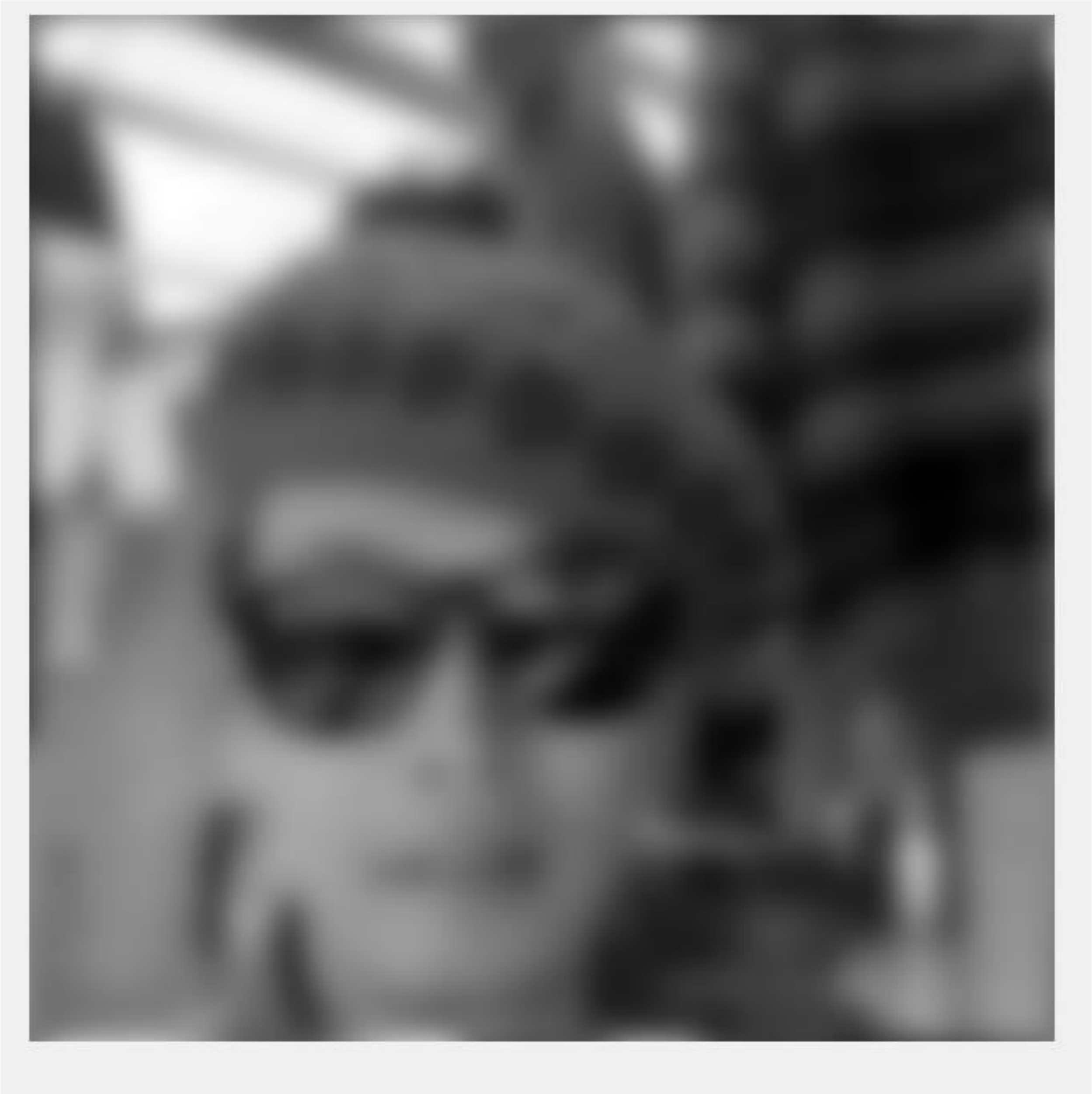
Hybrid image



HPF image:



LPF Image:

Hybrid Image Source Codes:

```
% load in my and friend's images
myImageDir='me3.png';
friendImageDir='friend.png';

% read it in and gray scale it
myImage=im2double( rgb2gray(  imread(myImageDir) ) );
friendImage=im2double( rgb2gray(  imread(friendImageDir) )
);

% setting gaussian matrix multiplication
sigma=6;
gaussianDim=3*sigma*2+1;

myImageHPFed=HPF(myImage,gaussianDim);

figure('Name','high pass filtered my image'),
imshow(myImageHPFed,[]);

sigma=1;
gaussianDim=3*sigma*2+1;
friendImageLPF=LPF(friendImage,gaussianDim);

figure('Name','low pass filtered friend image'),
imshow(friendImageLPF,[]);

hybridedImage=friendImageLPF+myImageHPFed;
figure('Name','Hybrided Image'), imshow(hybridedImage, []);
```

high pass filter function source code:

```
function HPF_Image=HPF(image,filter_size)

% fouier and shift the image
fourierImage = fft2(image);
fourierShiftedImage=fftshift(fourierImage);

% alloting size of image
[i ,j]=size(fourierImage);

% looping through the iteration
```

```
X=0:j-1;
Y=0:i-1;
% mesh grid
[X, Y]=meshgrid(X,Y);
Center_of_x=0.5*j;
Center_of_y=0.5*i;
% h(t) for fourier transformation
G=1-exp(-((X-Center_of_x).^2+(Y-
Center_of_y).^2)./(2*filter_size).^2);

% fourer transform and shift for High Pass Filtered Images
fourierFilterImage=fourierShiftedImage.*G;
fourierFilteredShiftedImage=ifftshift(fourierFilterImage);
HPF_Image=ifft2(fourierFilteredShiftedImage);

end
```

LPF function source codes:

```
% function for images undergoing low-pass filter
function LPF_Image=LPF(image,filter_size)

% fourier and shifted
fourierImage = fft2(image);
fourierShiftedImage=fftshift(fourierImage);

% size of the image
[i, j]=size(fourierImage);

% iterating through matrix to add values
X=0:j-1;
Y=0:i-1;
% mesh grid
[X, Y]=meshgrid(X,Y);
Center_of_x=0.5*j;
Center_of_y=0.5*i;
% h(x) to be used
G=exp(-((X-Center_of_x).^2+(Y-
Center_of_y).^2)./(2*filter_size).^2);

% fourier transform and shift for low pass filter process
image_FFT_filtered=fourierShiftedImage.*G;
```

```
image_FFT_filtered_shifted=ifftshift(image_FFT_filtered);
LPF_Image=ifft2(image_FFT_filtered_shifted);

end
```