Homework 2 Solutions

Question 1

```
img = imread('slope.tif');

bitplane8 = mod(img,2);
temp = bitshift(img, -1);
bitplane7 = mod(temp,2);
temp = bitshift(temp, -1);
bitplane6 = mod(temp,2);
temp = bitshift(temp, -1);
bitplane5 = mod(temp,2);
temp = bitshift(temp, -1);
bitplane4 = mod(temp,2);
temp = bitshift(temp, -1);
bitplane3 = mod(temp,2);
temp = bitshift(temp, -1);
bitplane2 = mod(temp,2);
bitplane1 = bitshift(temp, -1);

figure;
subplot(2,2,1), imshow(bitplane1.*255); title('Bit Plane 1');
subplot(2,2,2), imshow(bitplane2.*255); title('Bit Plane 2');
subplot(2,2,3), imshow(bitplane3.*255); title('Bit Plane 3');
subplot(2,2,4), imshow(bitplane4.*255); title('Bit Plane 4');

figure;
subplot(2,2,1), imshow(bitplane5.*255); title('Bit Plane 5');
subplot(2,2,2), imshow(bitplane6.*255); title('Bit Plane 6');
subplot(2,2,3), imshow(bitplane7.*255); title('Bit Plane 7');
subplot(2,2,4), imshow(bitplane8.*255); title('Bit Plane 8');

rimg1 = bitplane1.*bitshift(1, 7);
rimg2 = rimg1 + bitplane2.*bitshift(1, 6);
rimg3 = rimg2 + bitplane3.*bitshift(1, 5);
rimg4 = rimg3 + bitplane4.*bitshift(1, 4);
rimg5 = rimg4 + bitplane5.*bitshift(1, 3);
rimg6 = rimg5 + bitplane6.*bitshift(1, 2);
rimg7 = rimg6 + bitplane7.*bitshift(1, 1);
rimg8 = rimg7 + bitplane8;

figure;
subplot(2,2,1), imshow(rimg1); title('reconstructed upper 1 bit plane');
subplot(2,2,2), imshow(rimg2); title('reconstructed upper 2 bit planes');
subplot(2,2,3), imshow(rimg3); title('reconstructed upper 3 bit planes');
subplot(2,2,4), imshow(rimg4); title('reconstructed upper 4 bit planes');

figure;
subplot(2,2,1), imshow(rimg5); title('reconstructed upper 5 bit planes');
subplot(2,2,2), imshow(rimg6); title('reconstructed upper 6 bit planes');
subplot(2,2,3), imshow(rimg7); title('reconstructed upper 7 bit planes');
subplot(2,2,4), imshow(rimg8); title('reconstructed upper 8 bit planes');
```

Observations: The upper bitplanes contain more important structural and visual information, and the lower bitplanes are more likely to contain noise. Reconstructed images using 5 to 6 upper bitplanes are visually close to the full reconstruction, suggesting that the lowest bitplanes contribute little visual information.

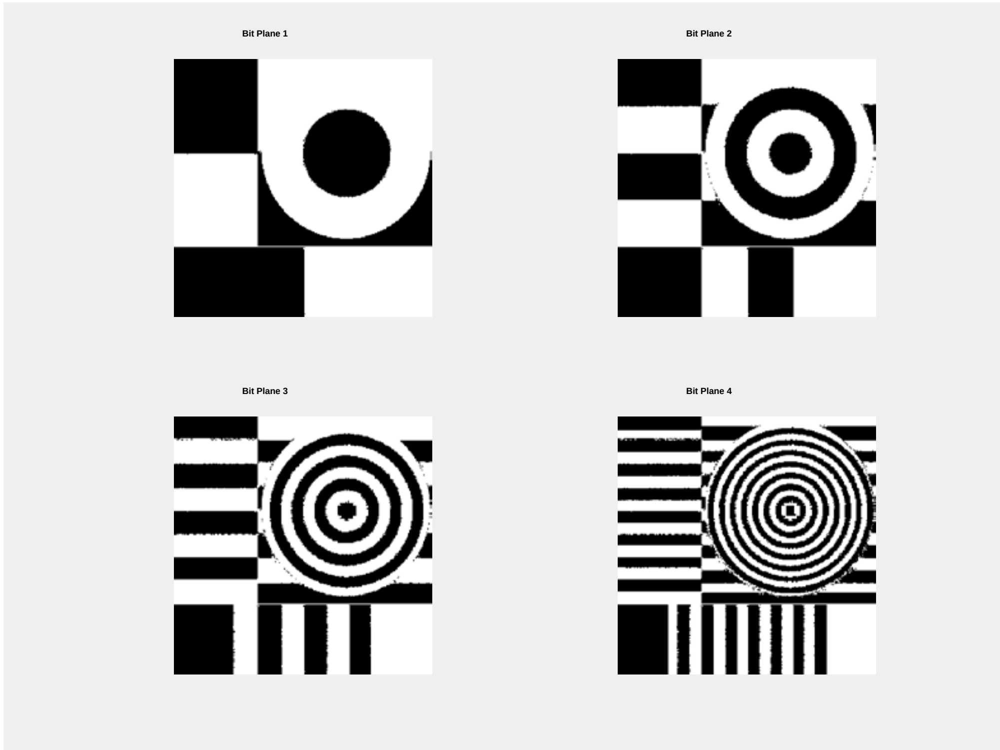

Figure 1.1

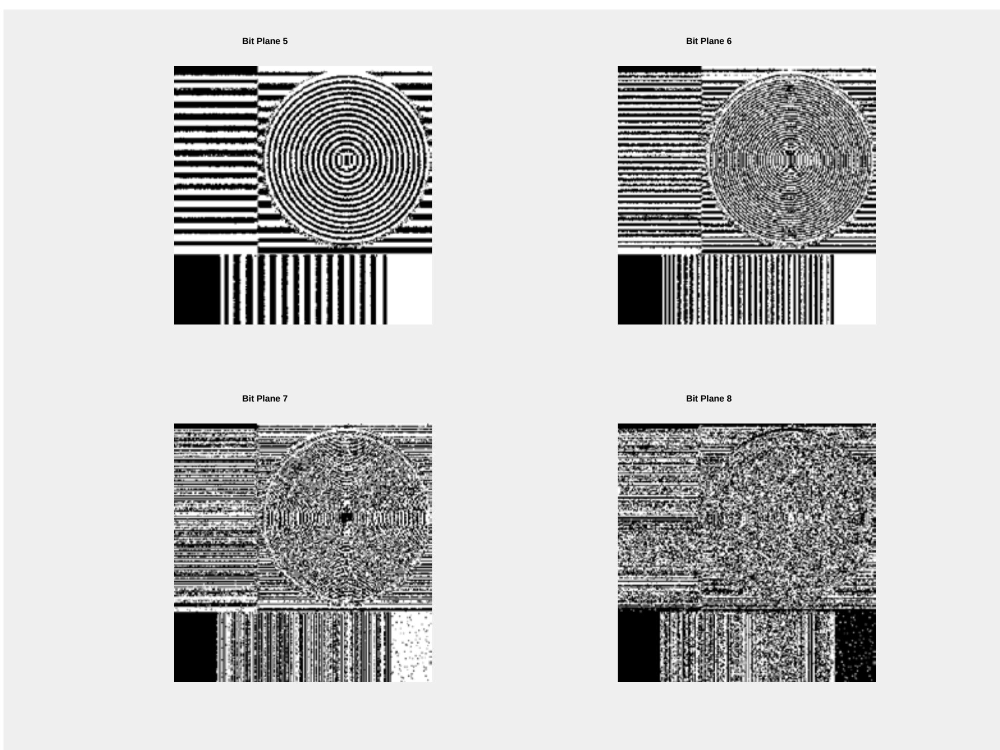

Figure 1.2

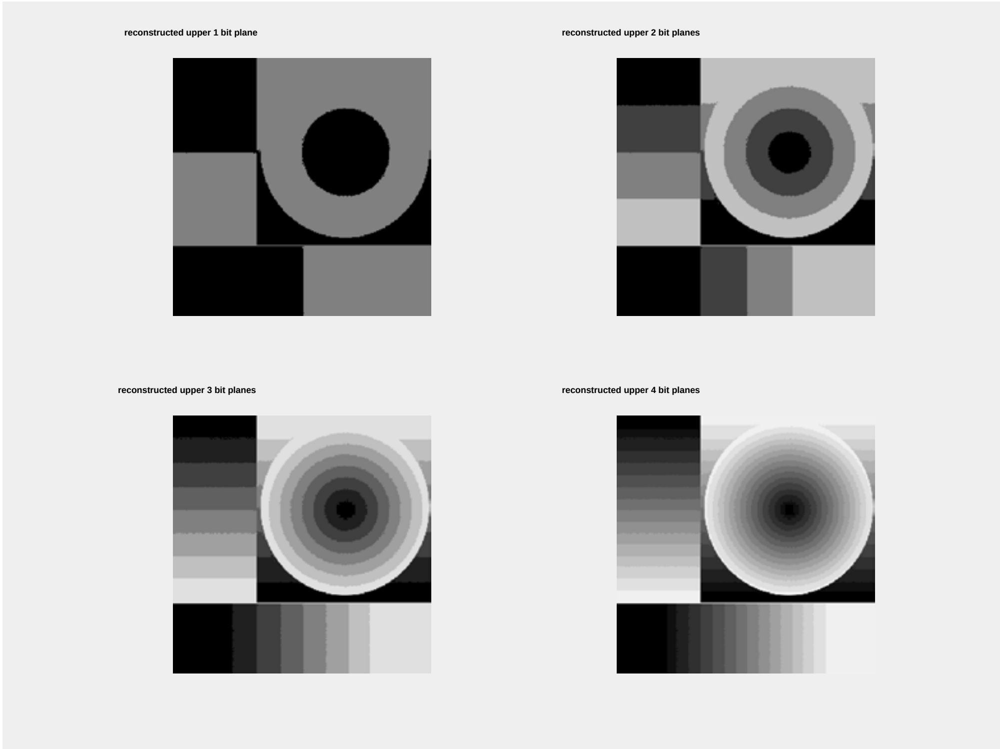

Figure 1.3

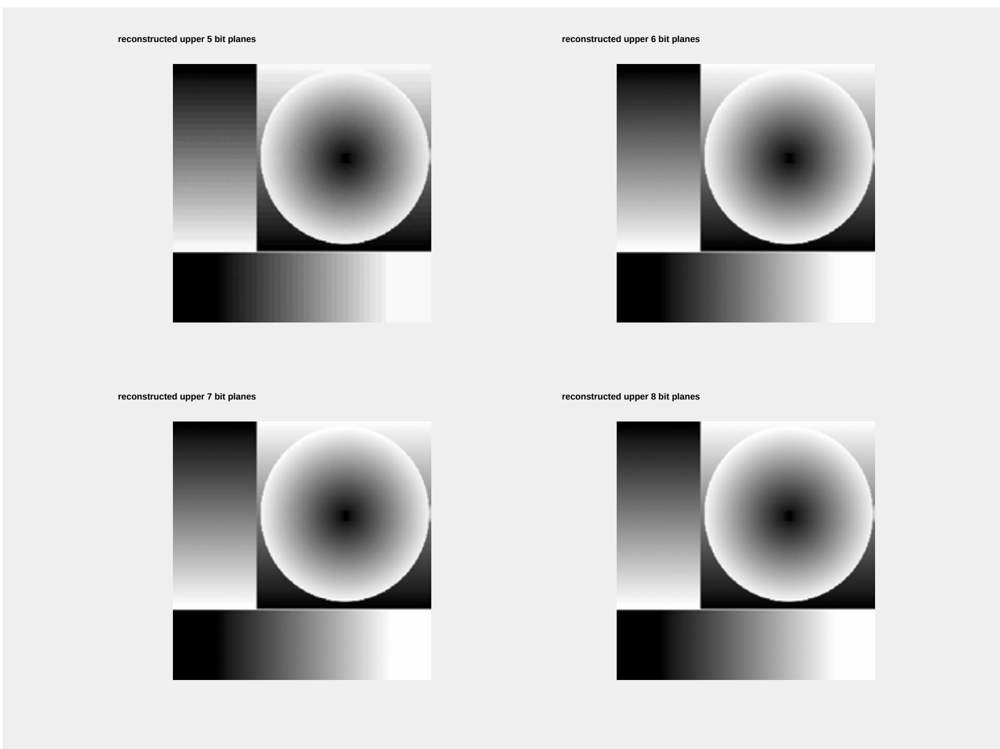

Figure 1.4

Question 2

```
figure;
img = double(imread('books.tif'));
subplot(2,2,1), imshow(img/255);
title('books image');
hist_img = hist(img(:), 0:255);

gamma = 0.5;
gamma_img = ((img/255).^gamma)*255;
subplot(2,2,2), imshow(gamma_img/255);
title('gamma mapping');
hist_gamma_img = hist(gamma_img(:), 0:255);

fscs_img = 255.*(img-min_value)./(max_value-min_value);
subplot(2,2,3), imshow(fscs_img/255);
title('full-scale contrast stretch');
hist_fscs_img = hist(fscs_img(:), 0:255);

cum_hist_img = cumsum(hist_img)/(prod(size(img)));
interM_img = cum_hist_img(img+1);
hequal_img = round(255*interM_img);
subplot(2,2,4), imshow(hequal_img/255);
title('histogram equalization');
hist_hequal_img = hist(hequal_img(:), 0:255);

figure;
subplot(2,2,1), bar(hist_img); axis([0,256,0,4000]);
title('books image'); xlabel('pixel value'); ylabel('umber of pixels');
subplot(2,2,2), bar(hist_gamma_img); axis([0,256,0,4000]);
title('gamma mapping'); xlabel('pixel value'); ylabel('number of pixels');
subplot(2,2,3), bar(hist_fscs_img); axis([0,256,0,4000]);
title('full-scale contrast stretch'); xlabel('pixel value'); ylabel('number of
pixels');
subplot(2,2,4), bar(hist_hequal_img); axis([0,256,0,4000]);
title('histogram equalizatioin'); xlabel('pixel value'); ylabel('number of pixels');
```

Observations: The original “books” image looks dark and low-contrast, and its histogram concentrates mostly at the half of lower intensity values. The Gamma mapping transformed image improves the overall brightness but does not help much improve the contrast. The histogram shifts to the middle intensity range, but does not occupy the low and high intensity ranges. The full-scale contrast stretched image significantly improves both brightness and contrast, and better spreads the intensity values across the histogram. The histogram-equalized image appears to have the best brightness and contrast, and has most evenly distributed intensity in its histogram.

Figure 2.1

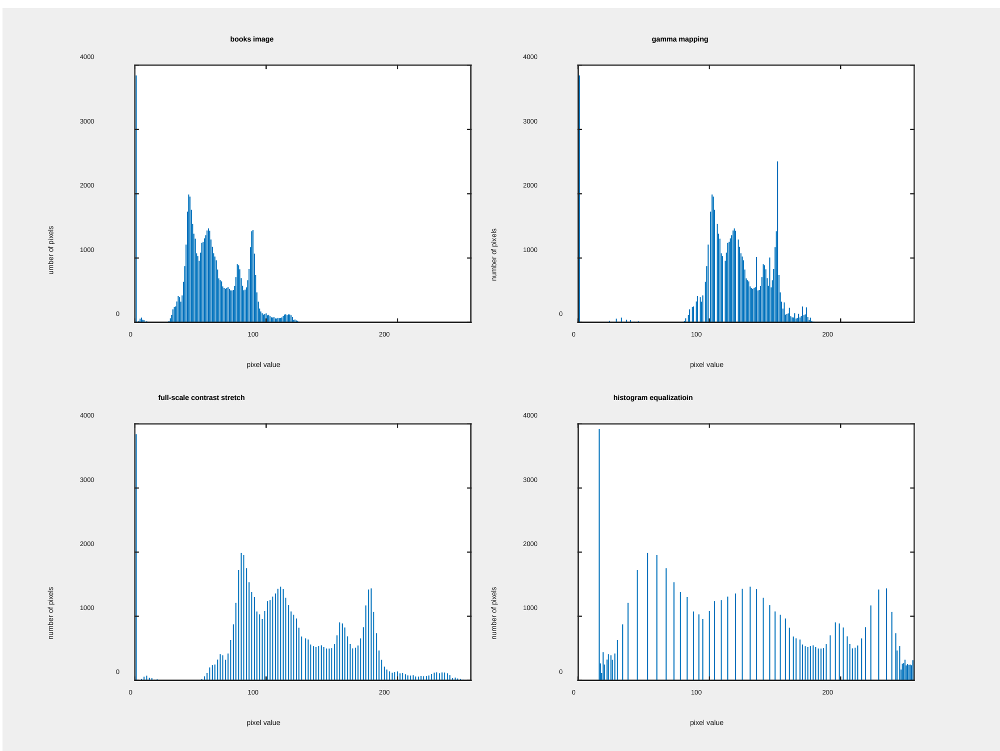

Figure 2.2

Question 3

```
img = double(imread('bridge.tif'));
figure(1);
subplot(2, 2, 1), imshow(img/255);
title('spatial domain image');

freq_img = fftshift(fft2(img));
view_img = log(1 + abs(freq_img));
white = max(view_img(:));
figure(2);
subplot(2, 2, 1), imshow(view_img/white);
title('frequency domain image');

img_size = size(img);
center = img_size/2;
[x y] = meshgrid(1:img_size(1), 1:img_size(2));
lowpass_filter = sqrt((x-center(1)).^2 + (y-center(2)).^2) <= (center(1)/8);
highpass_filter = sqrt((x-center(1)).^2 + (y-center(2)).^2) > (center(1)/2);
bandpass_filter = (~lowpass_filter) & (~highpass_filter);

freq_lowpass_img = freq_img.*lowpass_filter;
view_img = log(1 + abs(freq_lowpass_img));
figure(2);
subplot(2, 2, 2), imshow(view_img/white);
title('low-pass filtered image');

lowpass_img = real(ifft2(fftshift(freq_lowpass_img)));
figure(1);
subplot(2, 2, 2), imshow(lowpass_img/255);
title('low-pass filtered image');

freq_bandpass_img = freq_img.*bandpass_filter;
view_img = log(1 + abs(freq_bandpass_img));
figure(2);
subplot(2, 2, 3), imshow(view_img/white);
title('band-pass filtered image');

bandpass_img = real(ifft2(fftshift(freq_bandpass_img)));
figure(1);
subplot(2, 2, 3), imshow((bandpass_img+128)/255);
title('band-pass filtered image');

freq_highpass_img = freq_img.*highpass_filter;
view_img = log(1 + abs(freq_highpass_img));
figure(2);
subplot(2, 2, 4), imshow(view_img/white);
title('high-pass filtered image');

highpass_img = real(ifft2(fftshift(freq_highpass_img)));
figure(1);
subplot(2, 2, 4), imshow((highpass_img+128)/255);
title('high-pass filtered image');
```

Observations: The lowpass filter only takes a small proportion of all the frequency components of the original image. The lowpass filtered image appears to be a smoothed version of the original image. The bandpass and highpass filtered images have zero-means. The values are close to zero in most locations, and are significant mostly in edge and texture regions. In the Fourier transform domain, the energy is mostly concentrated in the lowpass regions. The bandpass and highpass regions are close to zero and are visible after logarithmic mapping.

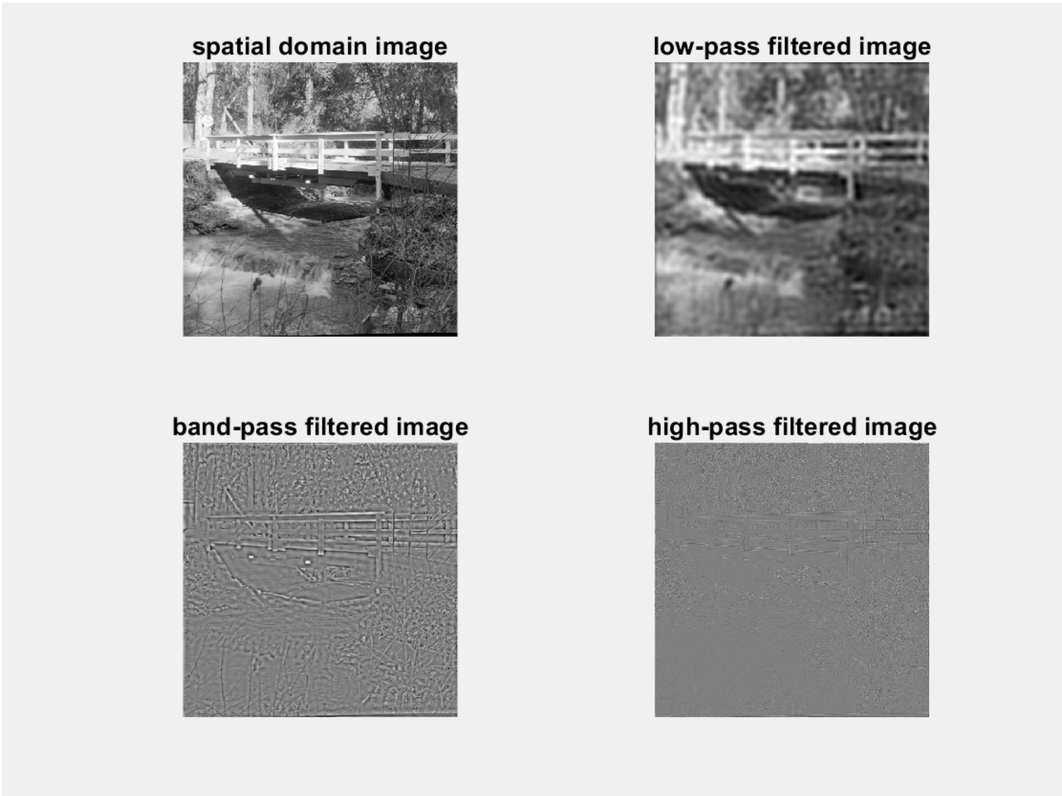

Figure 3.1

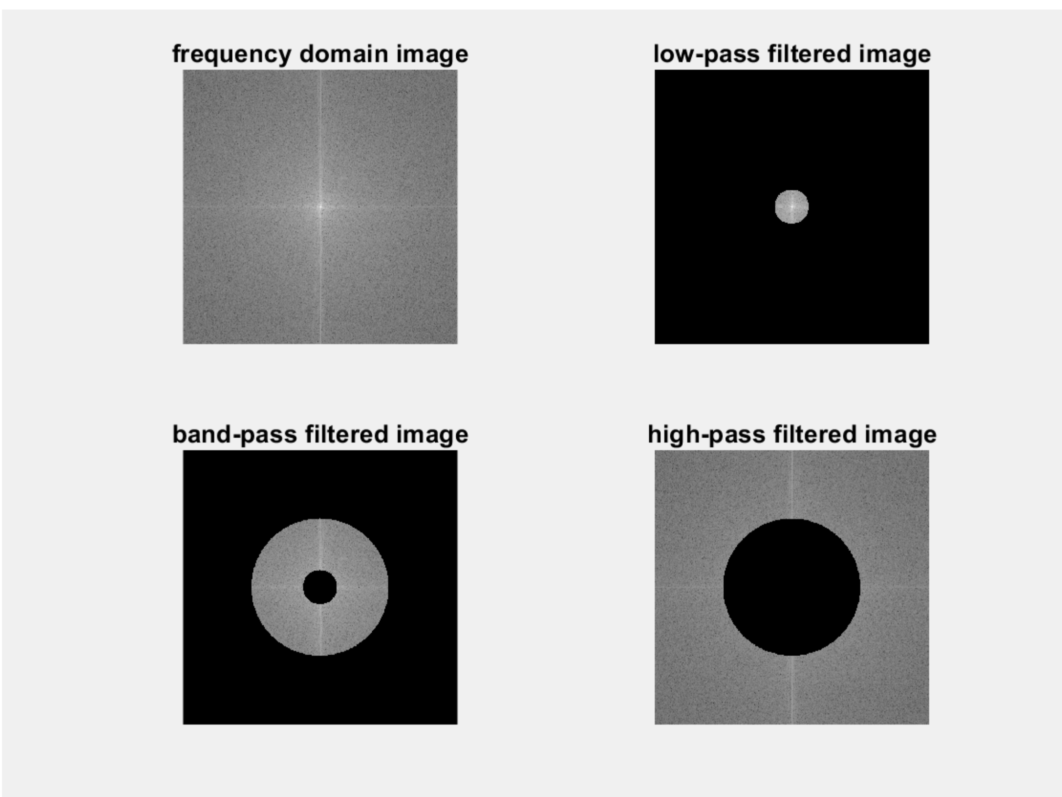

Figure 3.2

Question 4

```
img = double(imread('text.tif'));

sigma = 1;
N = 10;
[x, y] = meshgrid(-N:N, -N:N);
Gauss_filter = exp(-(x.*x + y.*y)/(2.*(sigma.*sigma)));
Gauss_filter = Gauss_filter/sum(Gauss_filter(:));

blur_img = conv2(img, Gauss_filter, 'same');

freq_Gauss_filter = fft2(Gauss_filter);
inv_freq_Gauss_filter = 1./freq_Gauss_filter;
inv_Gauss_filter = real(ifft2(inv_freq_Gauss_filter));

deblur_img = conv2(blur_img, inv_Gauss_filter, 'same');

figure;
subplot(2, 3, 1), imshow(img/255);
title('original image');
subplot(2, 3, 2), imshow(blur_img/255);
title('blurred image');
subplot(2, 3, 3), imshow(deblur_img/255);
title('deblurred image');
freq_img = log(1+abs(fftshift(fft2(img))));
subplot(2, 3, 4), imshow(freq_img./max(freq_img(:)));
title('frequency image');
freq_blur_img = log(1+abs(fftshift(fft2(blur_img))));
subplot(2, 3, 5), imshow(freq_blur_img./max(freq_blur_img(:)));
title('frequency blurred image');
freq_deblur_img = log(1+abs(fftshift(fft2(deblur_img))));
subplot(2, 3, 6), imshow(freq_deblur_img./max(freq_deblur_img(:)));
title('frequency deblurred image');

figure;
subplot(2, 2, 1), mesh(Gauss_filter);
title('spatial blur filter');
subplot(2, 2, 2), mesh(abs(fftshift(freq_Gauss_filter)));
title('frequency blur filter');
subplot(2, 2, 3), mesh(inv_Gauss_filter);
title('spatial deblur filter');
subplot(2, 2, 4), mesh(abs(fftshift(inv_freq_Gauss_filter)));
title('frequency deblur filter');
```

Observations: The original sharp text image becomes blurry after Gaussian filtering. The Gaussian filter is a lowpass filter that can be seen in the Fourier transforms of the images, where the high frequency regions become dark after filtering. The inverse Gaussain filter boost those frequency components and recover the high frequency energy as shown in the Fourier transform of the deblurred image. This also results in much sharper picture of the deblurred image. The inverse Gaussian filter is a highpass filter as shown in the frequency domain. In the spatial domain, the Gaussian lowpass filter does not take any negative values, but the inverse Gaussian filter takes both positive and negative values. In addition, there is strong artifacts at the image boundaries, which is an aliasing effect.

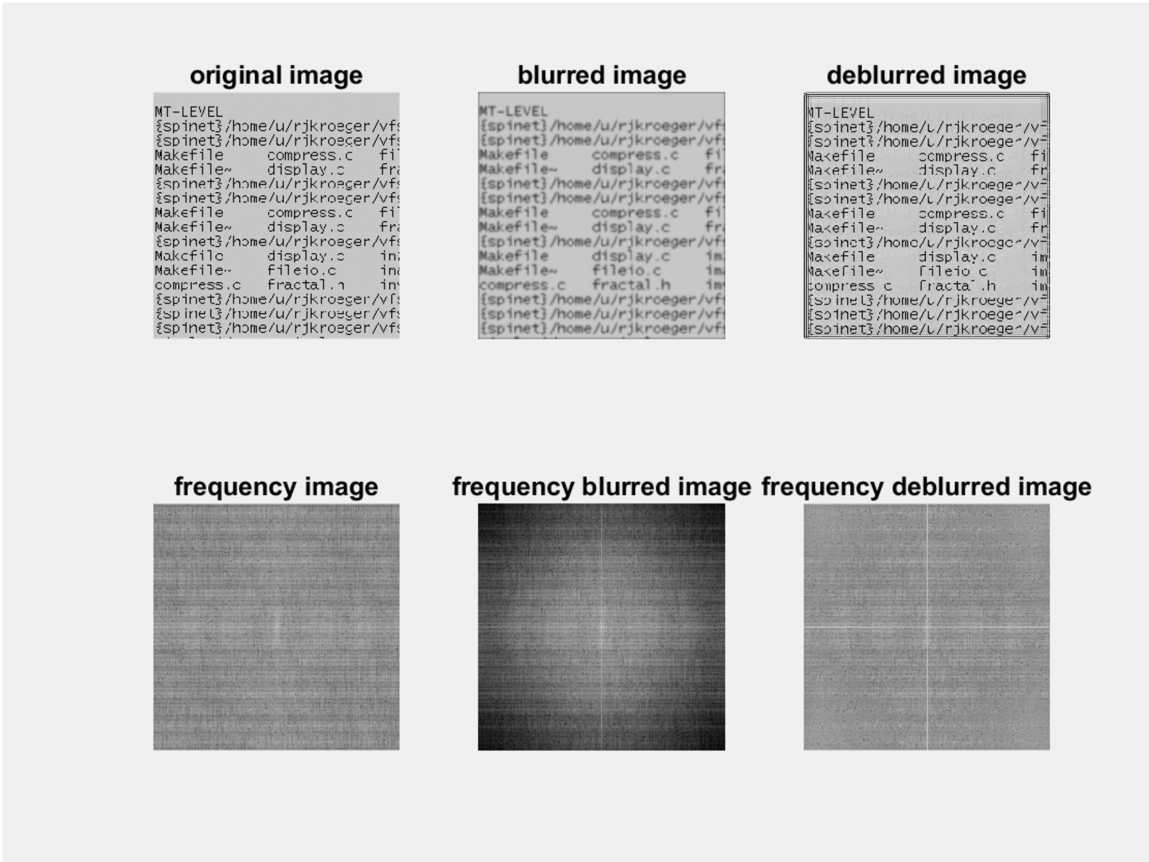

Figure 4.1

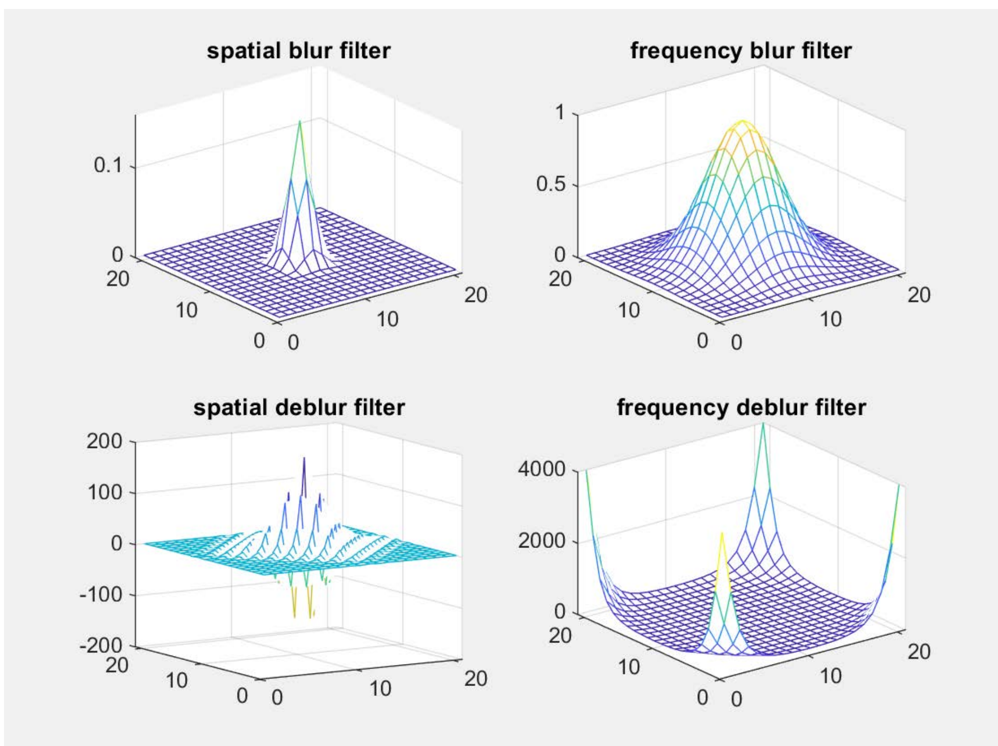

Figure 4.2

Question 5

```
img = double(imread('einstein.tif'));
[M N] = size(img);

for D = [3 7]
    downsamp_img = img(((D+1)/2):D:M, ((D+1)/2):D:N);
    upsamp_img = zeros(size(img));
    upsamp_img(((D+1)/2):D:end, ((D+1)/2):D:end) = downsamp_img;

    figure;
    subplot(2,2,1), imshow(img/255);
    subplot(2,2,2), imshow(upsamp_img/255);

    filter1 = zeros(2*D-1, 1)';
    filter1((D+1)/2:(3*D-1)/2) = 1;
    for n=1:N
        flt_line = conv(filter1, upsamp_img(:,n));
        inter1_img(:, n) = flt_line(D:(M+(D-1)));
    end
    for m=1:N
        flt_line = conv(filter1, inter1_img(m, :));
        inter1_img(m, :) = flt_line(D:(N+(D-1)));
    end
    subplot(2,2,3), imshow(inter1_img/255);

    filter2 = 1-(1/D)*abs([-(D-1):(D-1)]);
    for n=1:N
        flt_line = conv(filter2, upsamp_img(:,n));
        inter2_img(:, n) = flt_line(D:(M+(D-1)));
    end
    for m=1:N
        flt_line = conv(filter2, inter2_img(m, :));
        inter2_img(m, :) = flt_line(D:(N+(D-1)));
    end
    subplot(2,2,4), imshow(inter2_img/255);
end
```

Observations: Both “nearest neighbor” and “bilinear” methods recover the missing pixels from the downsampled images without changing the pixel values at the sampling points. The interpolated images by the nearest neighbor method appear to be blocky, and those by bilinear interpolation are much smoother and more blurry. The effect is stronger when the downsampling factor is larger, as shown in the D=3 and D=7 cases, respectively.

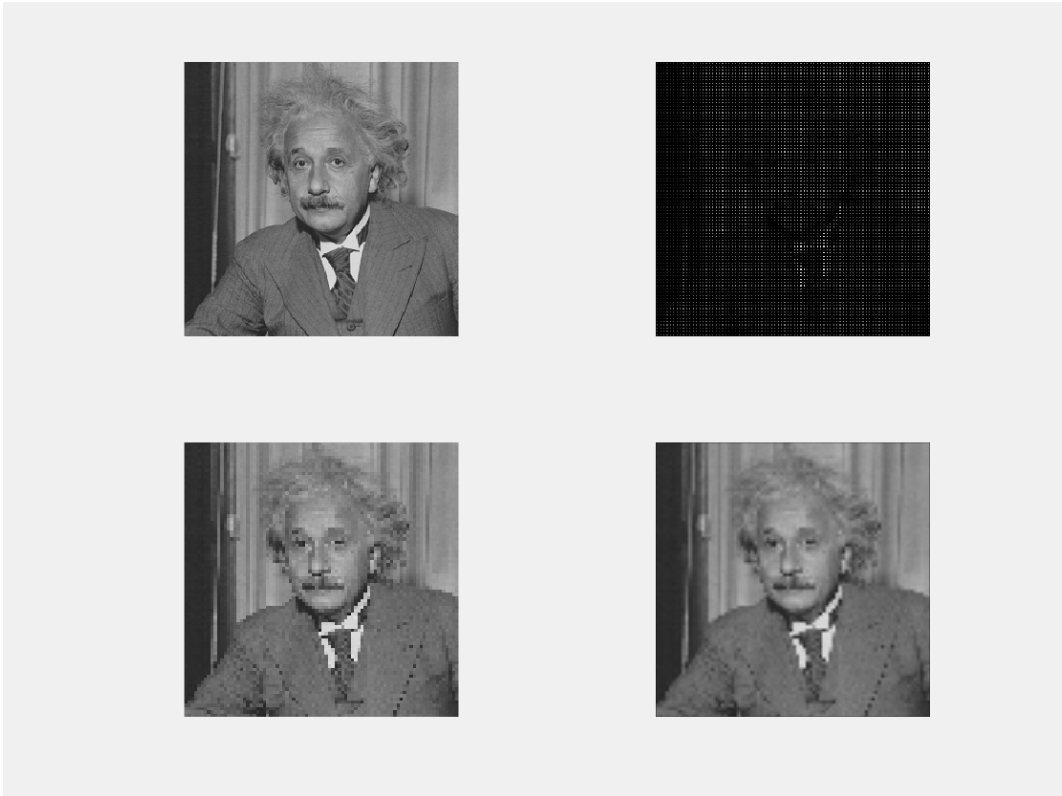

Figure 5.1

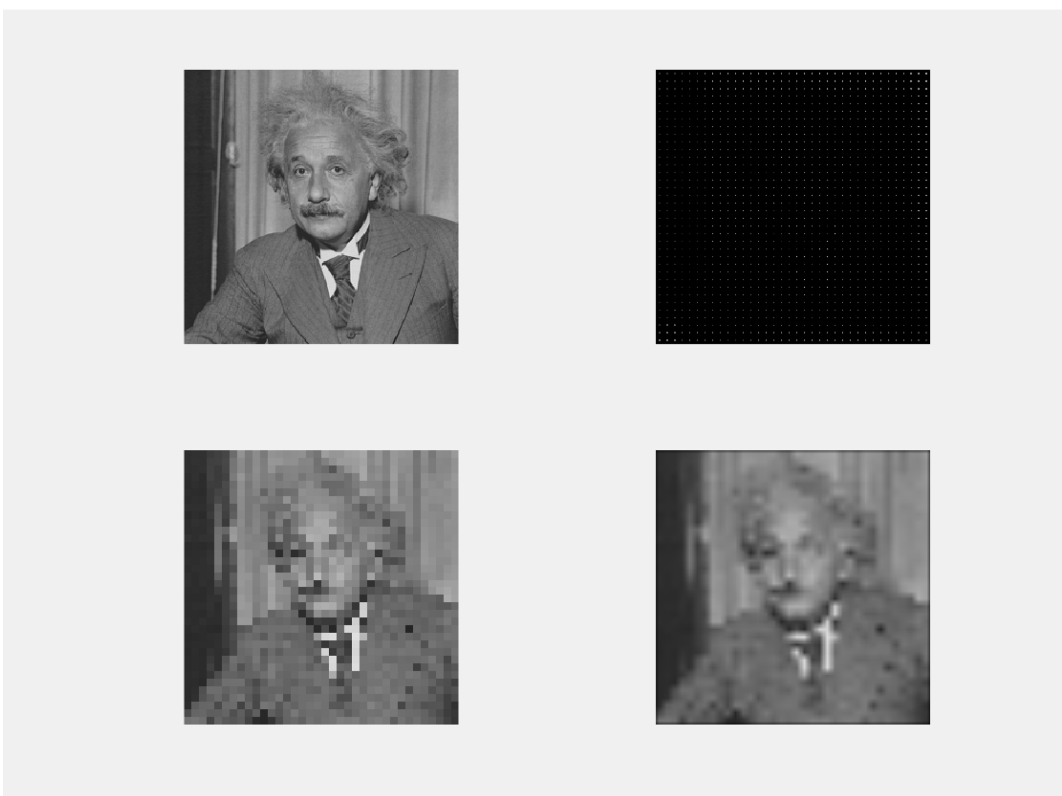

Figure 5.2