Question 1:

varibles used:

I = original image (calendar.png)

G = grayscale image obtained from I

Gn = G interpolated to the next powers of 2 on both axes

Gn1 = image obtained after applying 5 * 5 gaussian kernel on Gn

Gn1d = Gn1 downsampled by a factor of 2 (or scaled by a factor of 0.5)

Gn12 = image obtained after applying 5 * 5 gaussian kernel on Gn1d

Gn2d = Gn2 downsampled by a factor of 2 (or scaled by a factor of 0.5)

Gn3 = image obtained after applying 5 * 5 gaussian kernel on Gn2d

```
% load original image
I = imread('HW2_images/multires/calendar.png');
imshow(I), title('calendar.png (original image)', 'FontSize', 10)
```


```
% convert original image I to grayscale
G = rgb2gray(I);
imshow(G), title('grayscale image of calendar.png', 'FontSize', 10)
```

grayscale image of calendar.png


```
% upscaling the image by increase the dimensions to the next powers of 2
Gn = imresize(G,[2^nextpow2(size(G,1)) 2^nextpow2(size(G,2))]);
```

```
% gaussian kernel of size = 5 and standard deviation = 0.5
kernel = fspecial('gaussian', 5, 0.5);
```

```
% convolving on the image using a 5 * 5 circularly symmetric gaussian kernel
% with std = 0.5 to obtain the first level of the gaussian pyramid
Gn1 = conv2(Gn, kernel, 'same');

% downsampling level 1 image by a scale factor of 0.5
Gn1d = imresize(Gn1, 0.5);
```

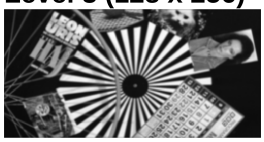
```
% convolving on the downsampled level 1 image Gn1d to obtain the
% second level of gaussian pyramid
Gn2 = conv2(Gn1d, kernel, 'same');

% downsampling level 2 image by a scale factor of 0.5
Gn2d = imresize(Gn2, 0.5);
```

```
% convolving on the downsampled level 2 image Gn2d to obtain the
```

```
% second level of gaussian pyramid
Gn3 = conv2(Gn2d, kernel, 'same');
```

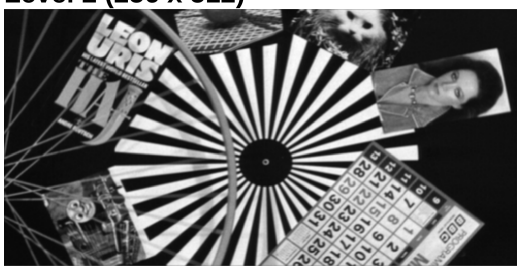
```
% Gaussian pyramid with gaussian filter smoothing
figure;

subplot(3,1,1),imshow(Gn3, []),title('Level 3 (128 x 256)', 'FontSize', 10)
set(gca, 'TitleHorizontalAlignment', 'left')
truesize([64, 128])
```

Level 3 (128 x 256)



```
subplot(3,1,2),imshow(Gn2, []),title('Level 2 (256 x 512)', 'FontSize', 10)
set(gca, 'TitleHorizontalAlignment', 'left')
truesize([128, 256])
```

Level 2 (256 x 512)



```
subplot(3,1,3),imshow(Gn1, []),title('Level 1 (512 x 1024)', 'FontSize', 10)
set(gca, 'TitleHorizontalAlignment', 'left')
truesize([256, 512])
```

Level 1 (512 x 1024)


```
Gn1 = imboxfilt(Gn, 5);
Gn1d = imresize(Gn1, 0.5);
```

```
Gn2 = imboxfilt(Gn1d, 5);
Gn2d = imresize(Gn2, 0.5);
```

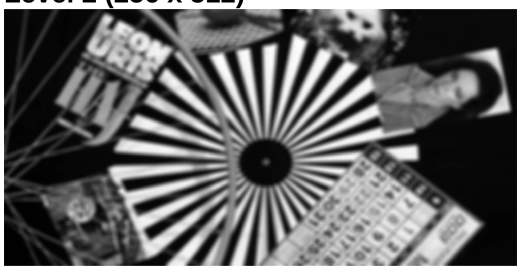
```
Gn3 = imboxfilt(Gn2d, 5);
```

```
% Gaussian pyramid with box filter filter smoothing
figure;

subplot(3,1,1),imshow(Gn3, []),title('Level 3 (128 x 256)', 'FontSize', 10)
set(gca, 'TitleHorizontalAlignment', 'left')
truesize([64, 128])
```

Level 3 (128 x 256)

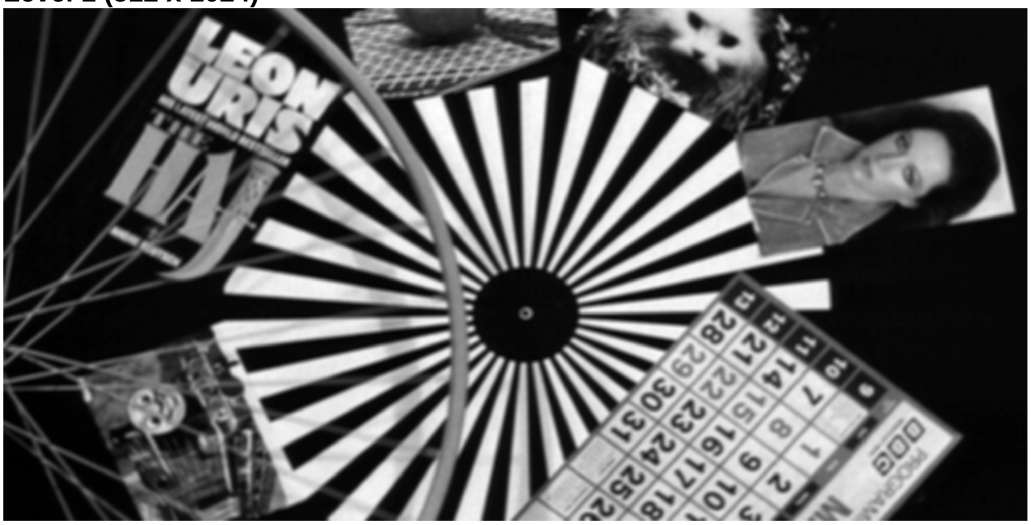
```

subplot(3,1,2),imshow(Gn2, []),title('Level 2 (256 x 512)', 'FontSize', 10)
set(gca, 'TitleHorizontalAlignment', 'left')
truesize([128, 256])
```

Level 2 (256 x 512)



```
subplot(3,1,3),imshow(Gn1, []),title('Level 1 (512 x 1024)', 'FontSize', 10)
set(gca, 'TitleHorizontalAlignment', 'left')
truesize([256, 512])
```

Level 1 (512 x 1024)



Gaussian filter does a better job of smoothing the image compared to box filter. To understand why, we need to look at the frequency response of gaussian and box filter. After performing the Fourier transform to convert the functions to freqency domain, we see gaussian transforms to a gaussian and box transforms to a sinc function (= sin(x) / x). This means that although both gaussian and box filters are both low pass filters, box filters allows some of the high frequencies to pass through, while gaussian filter is a true low pass filter. Hence, gaussian filter does a better job of smoothing compared to the box filter.

Question 2:

variables used:

I = original image (calendar.png)

G = grayscale image obtained from I

Gn = G interpolated to the next powers of 2 on both axes

Gn1 = image obtained after applying 5 * 5 gaussian kernel on Gn

Hn0 = absoulute difference of Gn1 and G

Gn1d = Gn1 downsampled by a factor of 2 (or scaled by a factor of 0.5)

Gn12 = image obtained after applying 5 * 5 gaussian kernel on Gn1d

Hn1 = absoulute difference of Gn2 and Gn1d

Gn2d = Gn2 downsampled by a factor of 2 (or scaled by a factor of 0.5)

Gn3 = image obtained after applying 5 * 5 gaussian kernel on Gn2d

Hn2 = absoulute difference of Gn3 and Gn2d

```
% convert original image I to grayscale
G = rgb2gray(I);
```

```
% convolving on the image using a 5 * 5 circularly symmetric gaussian kernel
% with std = 0.5 to obtain the first level of the gaussian pyramid
Gn1 = imgaussfilt(G, 3, 'FilterSize', 5);

% absolute difference of level 1 image and original image
Hn0 = imabsdiff(Gn1, G);

% downsampling level 1 image by a scale factor of 0.5
Gn1d = imresize(Gn1, 0.5);
```

```
% convolving on the downsampled level 1 image Gn1d to obtain the
% second level of gaussian pyramid
Gn2 = imgaussfilt(Gn1d, 3, 'FilterSize', 5);

% absolute difference of level 2 image and downsampled level 1 image
Hn1 = imabsdiff(Gn2, Gn1d);

% downsampling level 2 image by a scale factor of 0.5
Gn2d = imresize(Gn2, 0.5);
```

```
% convolving on the downsampled level 2 image Gn1d to obtain the
% third level of gaussian pyramid
Gn3 = imgaussfilt(Gn2d, 3, 'FilterSize', 5);

% absolute difference of level 3 image and downsampled level 1 image
Hn2 = imabsdiff(Gn3, Gn2d);

% downsampling level 3 image by a scale factor of 0.5
Gn3d = imresize(Gn3, 0.5);
```

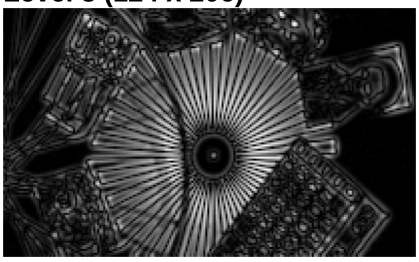
```
% Laplacian pyramid
```

```
figure;

subplot(4,1,1),imshow(Hn2, []),title('Level 4 (62 x 103)', 'FontSize', 10)
set(gca, 'TitleHorizontalAlignment', 'left')
truesize([62, 103])
```

Level 4 (62 x 103)

```
subplot(4,1,2),imshow(Hn2, []),title('Level 3 (124 x 206)', 'FontSize', 10)
set(gca, 'TitleHorizontalAlignment', 'left')
truesize([124, 206])
```

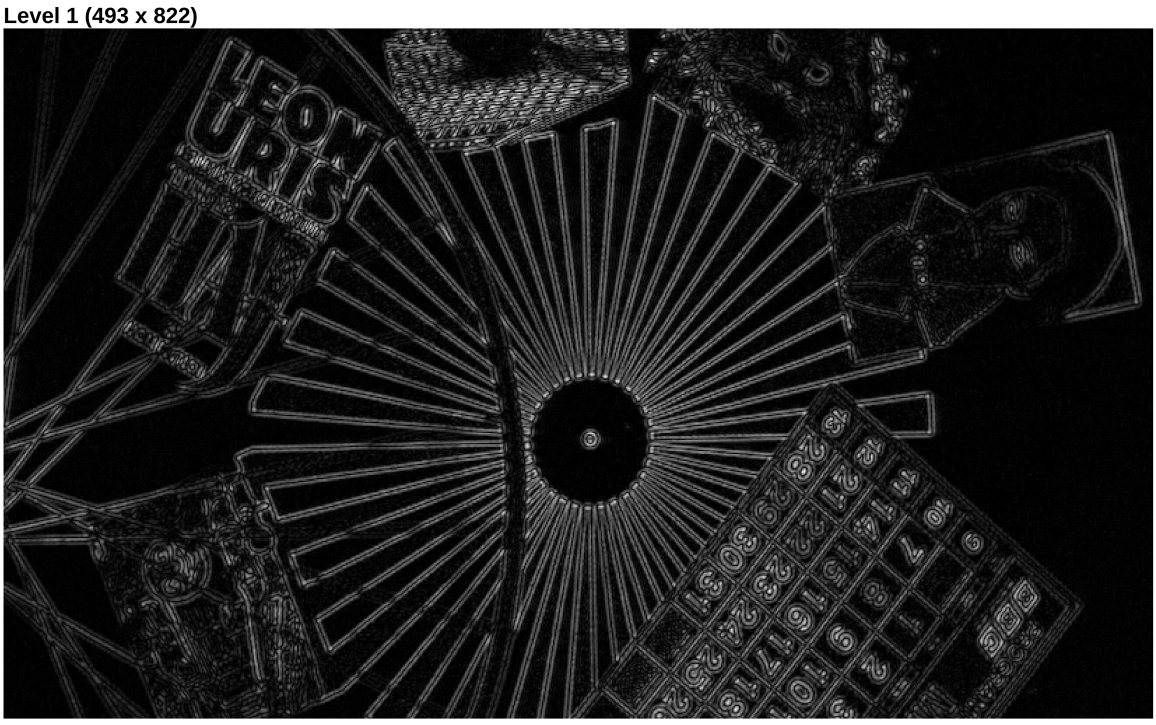
Level 3 (124 x 206)



```
subplot(4,1,3),imshow(Hn1, []),title('Level 2 (247 x 411)', 'FontSize', 10)
set(gca, 'TitleHorizontalAlignment', 'left')
truesize([247, 411])
```

Level 2 (247 x 411)

```
subplot(4,1,4),imshow(Hn0, []),title('Level 1 (493 x 822)', 'FontSize', 10)
set(gca, 'TitleHorizontalAlignment', 'left')
truesize([493, 822])
```



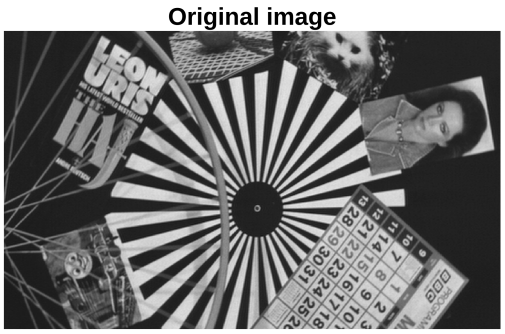

```
% interpolating downsampled level 3 image
Gn3I = imresize(Gn3d, 2);

% reconstructing level 2 image from interpolated level 3 image and level 2
% difference image
Gn2R = imadd(Gn3I, Hn2);
```

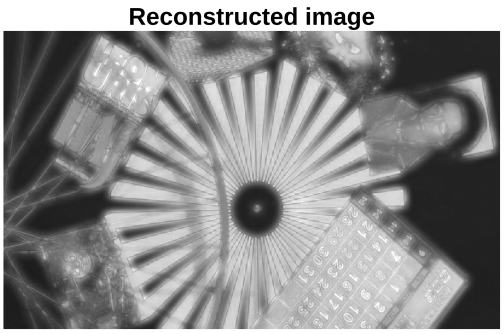
```
% interpolating downsampled level 2 image
Gn2I = imresize(Gn2R, 2);
Gn2I(248, :) = [];
Gn2I(:, 412) = [];

% reconstructing level 2 image from interpolated level 3 image and level 2
% difference image
Gn1R = imadd(Gn2I, Hn1);
```

```
% interpolating downsampled level 1 image
Gn1I = imresize(Gn1R, 2);
Gn1I(494, :) = [];
```

```
% reconstructing level 1 image from interpolated level 2 image and level 2
% difference image
GR = imadd(Gn1I, Hn0);
```

```
% comparison between level 1 reconstructed image and original image
f = figure('Position', [100, 100, 1000, 256]);
subplot(1,2,1),imshow(G),title('Original image')
subplot(1,2,2),imshow(GR),title('Reconstructed image')
```





Question 3:

variables used:

I = original image

G = normalized double precision grayscale image

Gb = image after applying bilateral filter on G

Gycbcr = image after converting I to ycbcr colorspace

Gbn = Gb scaled to 255

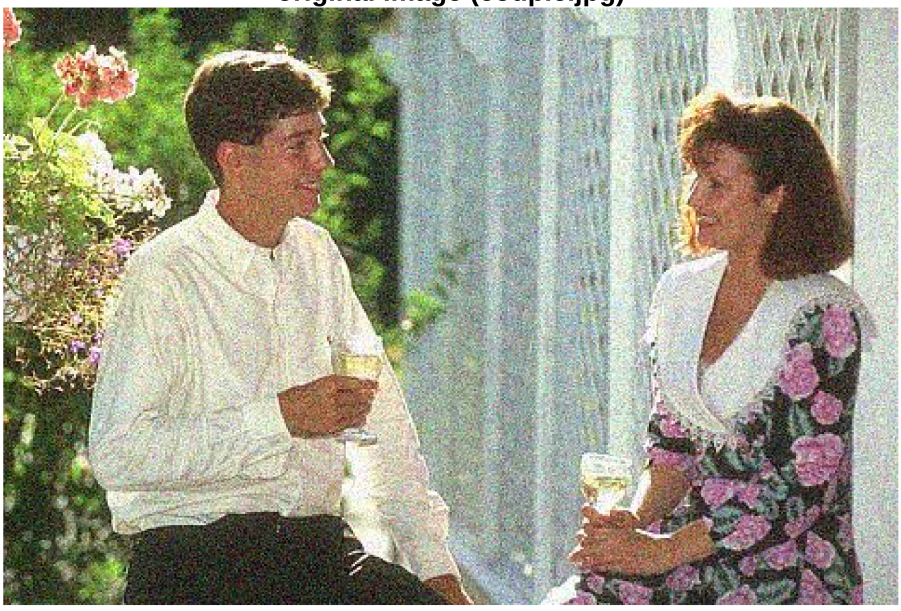
Grgb = image Grcbcr in RGB colorspace

ss = spatial sigma

dos = degree of smoothness

```
figure;
I = imread('HW2_images/bilateral/couple.jpg');
imshow(I);
title("original image (couple.jpg)", FontSize=10)
```

original image (couple.jpg)



```
G = normalize(rgb2gray(im2double(I)), 'range');
```

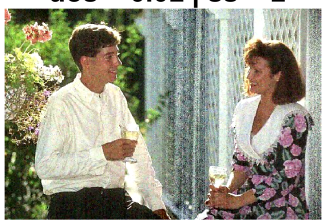
```
Gb = imbilatfilt(G);
Gycbcr = rgb2ycbcr(I);
Gbn = uint8(Gb * 255);
Gycbcr(:,:,1) = Gbn;
imshow(ycbcr2rgb(Gycbcr))
title("image after applying bilateral filter with default values", FontSize=10)
```

image after applying bilateral filter with default values

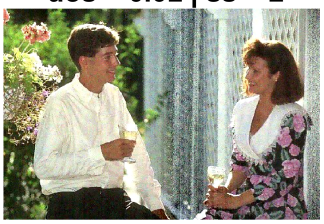
```
% trying different combination of values for spatial sigma and degree of
% smoothness to land on an optimal set of values
ss = [0.5, 1, 2];
dos = [0.01, 0.1, 1, 10];

figure('Position', [100, 100, 1024, 1024])
for i = 1:length(dos)
    for j = 1:length(ss)
        Gb = imbilatfilt(G, dos(i), ss(j));
        Gbn = uint8(Gb * 255);
        Gycbcr(:,:,1) = Gbn;
        Grgb = ycbcr2rgb(Gycbcr);
        pos = (i - 1) * 3 + j;
        subplot(4, 3, pos);
        imshow(Grgb);
        title(" dos = " + dos(i) + " | ss = " + ss(j));
    end
end
```

dos = 0.01 | ss = 0.5

dos = 0.01 | ss = 1

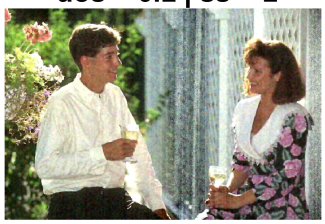


dos = 0.01 | ss = 2

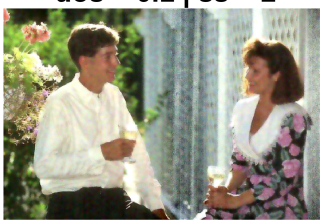


dos = 0.1 | ss = 0.5

dos = 0.1 | ss = 1



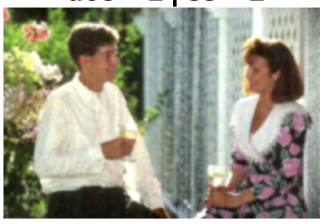
dos = 0.1 | ss = 2



dos = 1 | ss = 0.5

dos = 1 | ss = 1

dos = 1 | ss = 2



dos = 10 | ss = 0.5

dos = 10 | ss = 1

dos = 10 | ss = 2

```
figure;
Gb = imbilatfilt(G, 10, 1);
Gycbcr = rgb2ycbcr(I);
Gbn = uint8(Gb * 255);
Gycbcr(:,:,1) = Gbn;
imshow(ycbcr2rgb(Gycbcr))
title("best results found for values dos = 10 and ss = 1", FontSize=10)
```

best results found for values dos = 10 and ss = 1