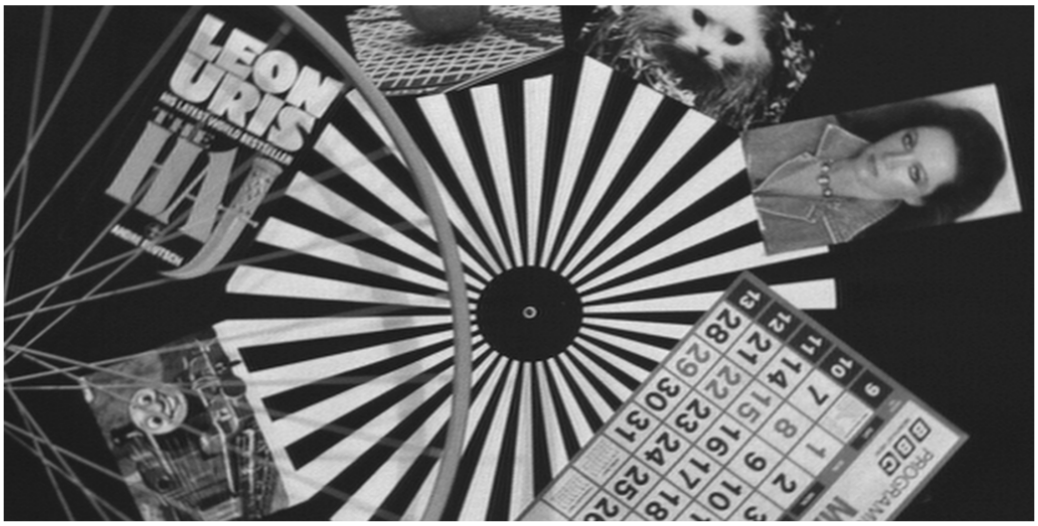
1. **Gaussian Pyramid (Filtering and Decimation)**

a) Image resizing to the next power of 2 and extract gray level image.

```
I = imread("~/Documents/HW2_images/multires/calendar.png");
G = rgb2gray(I);
figure;imshow(G);
G = double(G)/255;
Gs = imresize(G,[2^nextpow2(size(I,1)) 2^nextpow2(size(I,2))]);
figure;imshow(Gs);
```



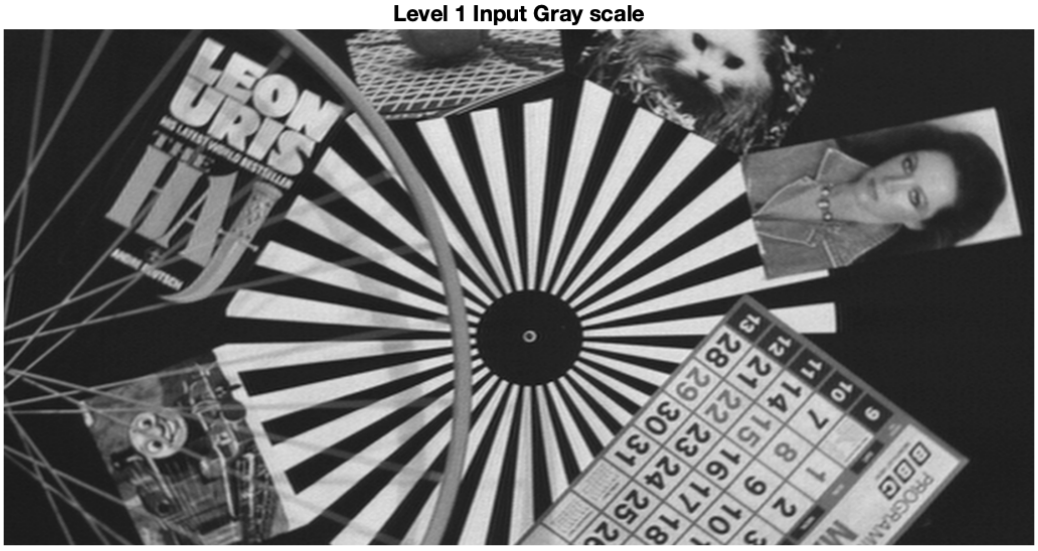
Gray Scale Image 512x1024

Original Image 493x822x3

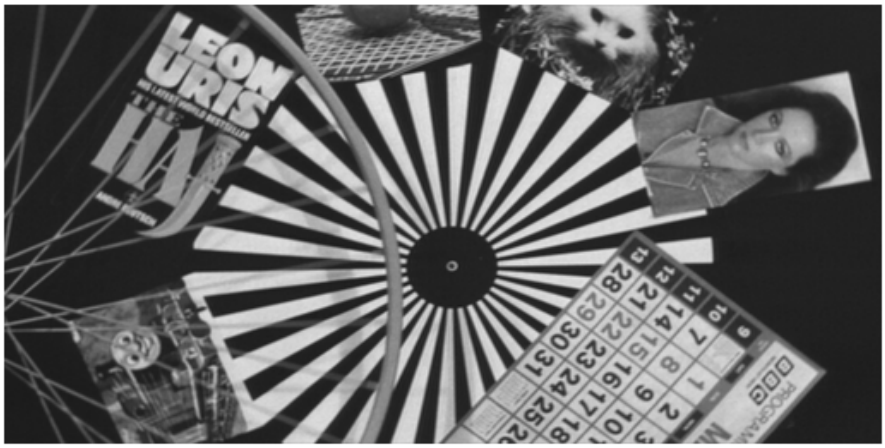
b) 3 Level Gaussian Pyramid with 5x5 circular symmetric gaussian filter

```
g0 = Gs;
filter = fspecial('gaussian',5);
figure;imshow(g0);
g1 = conv2(g0, filter, "same");
g1s = imresize(g1, 0.5);
figure;imshow(g1s);
g2 = conv2(g1s, filter, "same");
g2s = imresize(g2, 0.5);
figure;imshow(g2s);
```

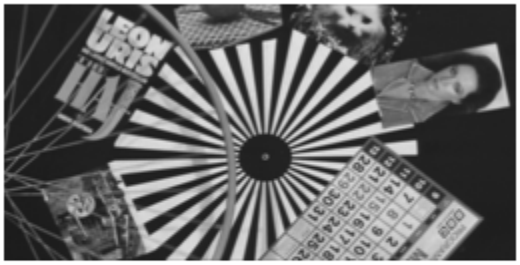
Level 1 Input Gray scale



Level 2 Gaussian Filter + Downscale



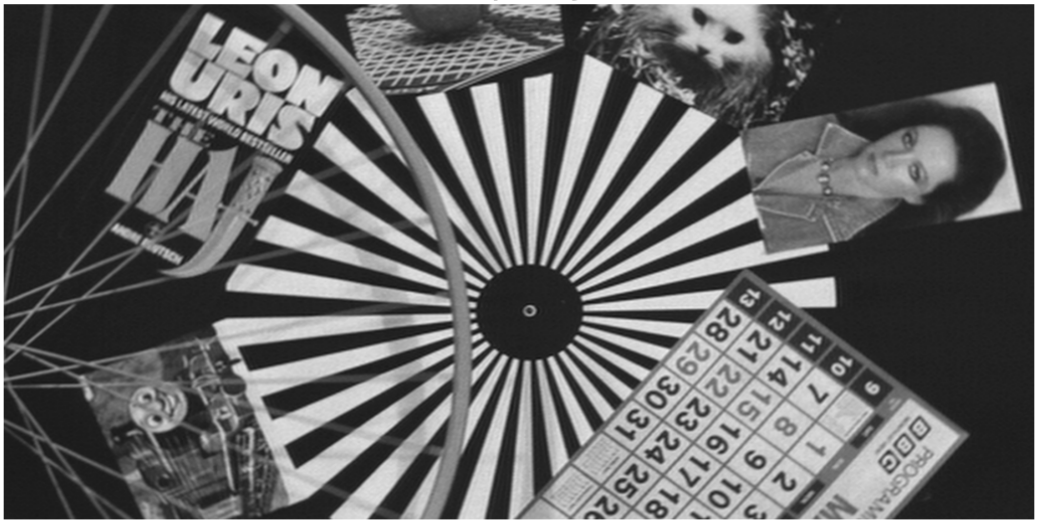
Level 3 Gaussian Filter + downscale



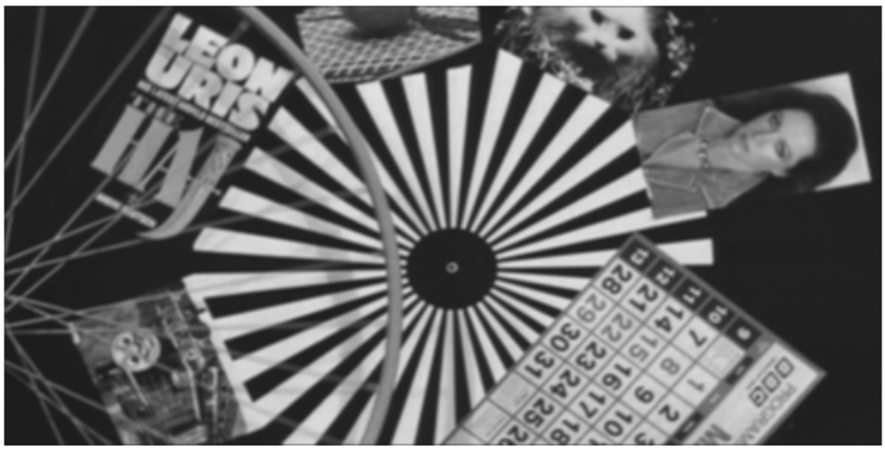
c) 3 Level Gaussian Pyramid with 5x5 box-car filter

```
filter2 = 1/25 * ones(5);
bc1 = conv2(g0, filter2, "same");
bc1s = imresize(bc1, 0.5);
figure;imshow(bc1s);
bc2 = conv2(bc1s, filter2, "same");
bc2s = imresize(bc2, 0.5);
figure;imshow(bc2s);
```

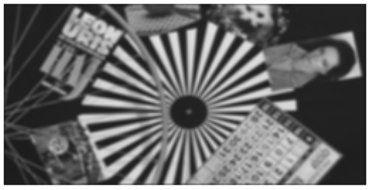
Level 1 Input Gray scale



Level 2 Box-car filter + downscale

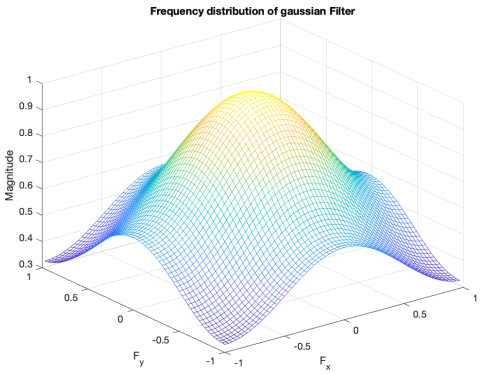


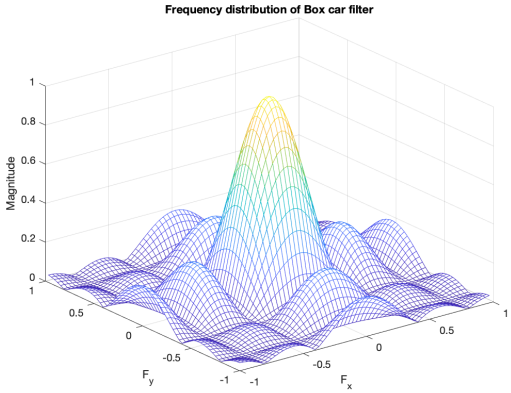
Level 3 Box-car filter + downscale



It is a 128x256 image.

The output image of the Gaussian filter is more blurry than that of the Box-car filter.





The frequency distribution of the filters shows the boxcar filter has sharp cut offs, this causes it to give more noisier output than Gaussian filter.

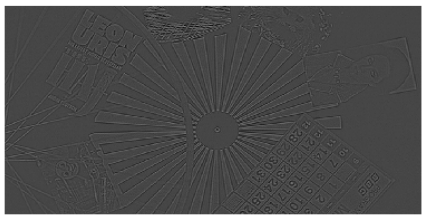
2. Laplacian Pyramid (Image differencing)

a) Laplacian Pyramid construction

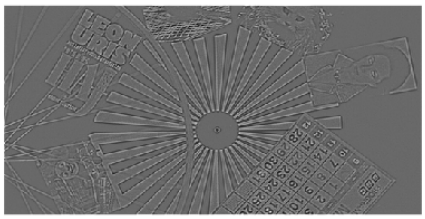
```
I = imread("~/Documents/HW2_images/multires/calendar.png");
G = rgb2gray(I);
figure;imshow(G);
G = double(G)/255;
Gs = imresize(G,[2^nextpow2(size(I,1)) 2^nextpow2(size(I,2))]);
figure;imshow(Gs);
filter = fspecial('gaussian');
f1 = conv2(Gs, filter, 'same');
%level 1
h0 = Gs - f1;
f1s = imresize(f1, 0.5);
%level 2
f2 = conv2(g1s, filter, 'same');
h1 = f1s - f2;
f2s = imresize(f2, 0.5);
%level 3
f3 = conv2(f2s, filter, 'same');
h2 = g2s - f3;
%level 4
f3s = imresize(f3, 0.5);

figure;
subplot(2,2,1),imshow(h0, []);
subplot(2,2,2),imshow(h1, []);
subplot(2,2,3),imshow(h2, []);
subplot(2,2,4),imshow(f3s);
```

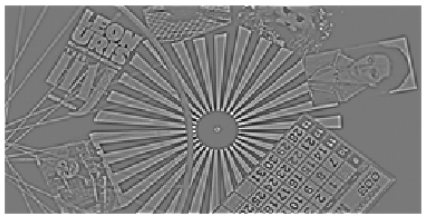
h0 512x1024



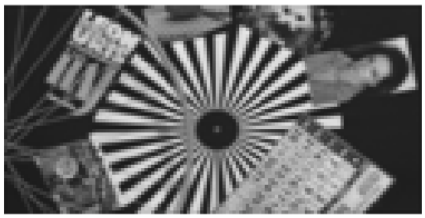
h1 256x512



h2 128x256



f3s 64x128



b) Recover Image

```
%Restore Image
p1 = imresize(f3s,2);
r1 = p1 + h2;
figure;imshow(r1);
p2 = imresize(r1,2);
r2 = p2 + h1;
figure;imshow(r2);
p3 = imresize(r2,2);
r3 = p3 + h0;

figure;
subplot(1,2,1), imshow(Gs);
subplot(1,2,2), imshow(r3);
```

Original Image

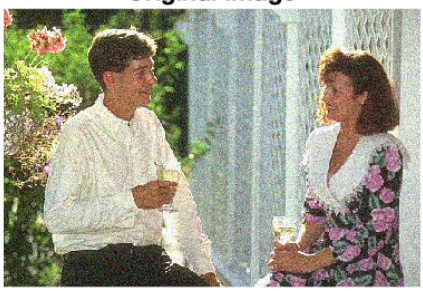
Recovered Image

3. Bilateral filter tune up

a) Recover RGB with default values of degreeOfSmoothing and spatialSigma

```
I = imread("~/Documents/HW2_images/bilateral/couple.jpg");
Ig = im2gray(I);
I2 = im2double(Ig);
If = imbilatfilt(I2);
figure;imshow(If);
%Recover Filtered RGB
Iy = rgb2ycbcr(I);
figure;imshow(Iy);
I8B = uint8(255 * If);
Iy(:,:,1) = I8B;
RGB = ycbcr2rgb(Iy);
figure;imshow(RGB);
```

Original Image



Recovered Image

b) Initially tried random values as DOS but the resulting Image has much noise. So, tried picking a patch of light color in the image and setting DOS as the standard variance of that patch (Process specified in the MATLAB documentation of imbilatfilt). The resulting image has less noise compared to the image of default params.

```
If1 = imbilatfilt(I2, 100);
Iy1 = rgb2ycbcr(I);
Iy1(:,:,1) = uint8(255 * If1);
RGB2 = ycbcr2rgb(Iy1);

patch = imcrop(I,[270, 35, 50 50]);
patchVar = std2(patch)^2;
DoS = 2*patchVar;
If2 = imbilatfilt(I2, DoS);
Iy3 = rgb2ycbcr(I);
Iy3(:,:,1) = uint8(255 * If2);
RGB3 = ycbcr2rgb(Iy3);
figure;
subplot(2,2,1), imshow(I);
subplot(2,2,2), imshow(RGB);
subplot(2,2,3), imshow(RGB2);
subplot(2,2,4), imshow(RGB3);
```

Original

Default Params (better edge less noise)

Standard Variance for White Patch (Much less noise)

100 DOS