Question 1

```
%{
Here, for convolution we have used a filter that gives equal weight to all the
pixels in the filter. Thus, the noise is reduced and smoothening occurs when convolution
is performed on the image.
We have used filters of 4 different sizes : 3,7,9 and 11 of which since 3 considers minimum
number of pixels, the image is relatively less smoothened and thus leads to less blurred image.
We see that as the filter size images, the output is more blurred, that is, more smoothening
has occured.
%}
im = imread('house1.jpg');
imgray = rgb2gray(im);
immsall = imresize(imgray, 0.25);
figure; imagesc(immsall); colormap gray; axis image;
```

```
filter = ones(3,3);
imresult3 = conv2(immsall, filter, 'same');
```

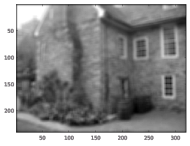
```
Warning: CONV2 on values of class UINT8 is obsolete.
        Use CONV2(DOUBLE(A),DOUBLE(B)) or CONV2(SINGLE(A),SINGLE(B)) instead.
```

```
figure; imagesc(imresult3); colormap gray; axis image;
```

```
filter = ones(7,7);
imresult7 = conv2(immsall, filter, 'same');
```

```
Warning: CONV2 on values of class UINT8 is obsolete.
        Use CONV2(DOUBLE(A),DOUBLE(B)) or CONV2(SINGLE(A),SINGLE(B)) instead.
```

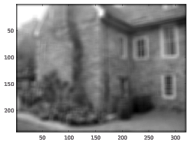
```
figure; imagesc(imresult7); colormap gray; axis image;
```



```
filter = ones(9,9);
imresult9 = conv2(immsall, filter, 'same');
```

```
Warning: CONV2 on values of class UINT8 is obsolete.
        Use CONV2(DOUBLE(A),DOUBLE(B)) or CONV2(SINGLE(A),SINGLE(B)) instead.
```

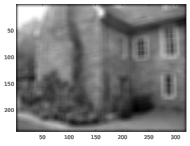
```
figure; imagesc(imresult9); colormap gray; axis image;
```



```
filter = ones(11,11);
imresult11 = conv2(immsall, filter, 'same');
```

```
Warning: CONV2 on values of class UINT8 is obsolete.
        Use CONV2(DOUBLE(A),DOUBLE(B)) or CONV2(SINGLE(A),SINGLE(B)) instead.
```

```
figure; imagesc(imresult11); colormap gray; axis image;
```

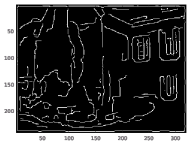


Question 2

```
%{
Here, we are performing edge detection. Initaially, I ran the edge detector with automatic thresholding.

Later, I modified the values that led me to infer that as the absolute difference between the lower and

higher threshold increases, the number of edges considered incrases and vice-versa.

Also, using the bwlabel function, we noticed how every edge that is connected gets the same label and

also every not connected edge gets a different label. Here, we have used the default version of bwlabel

that considers 8 connected components.

%}
```

Automatic Thresholding

```
[canny_edge11, threshold11] = edge(imresult11,'canny');
figure; imagesc(canny_edge11); colormap gray; axis image
```



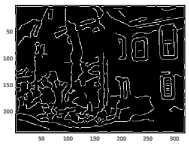
```
bwlabel(canny_edge11)
```

```
ans =

     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
```

```
threshold11
```

```
 threshold11 =

    0.0688    0.1718
```

```
[canny_edge7, threshold7] = edge(imresult7,'canny');
figure; imagesc(canny_edge7); colormap gray; axis image
```



```
bwlabel(canny_edge7)
```

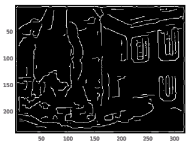
```
ans =

     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     1     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     1     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
```

```
threshold7
```

```
threshold7 =

    0.0750    0.1875
```

Manual Threshholding : [0.1 0.15]

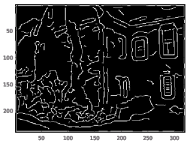
```
canny_edge11 = edge(imresult11,'canny',[0.1 0.15]);
figure; imagesc(canny_edge11); colormap gray; axis image
```



```
bwlabel(canny_edge11)
```

```
ans =

     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
     0     0     1     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
```

```
canny_edge7 = edge(imresult7,'canny',[0.1 0.15]);
figure; imagesc(canny_edge7); colormap gray; axis image
```

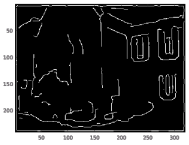


```
bwlabel(canny_edge7)
```

```
ans =

     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     1     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     1     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
```

Manual Threshholding : [0.05 0.25]

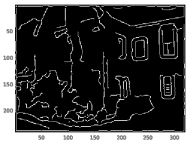
```
canny_edge11 = edge(imresult11,'canny',[0.05 0.25]);
figure; imagesc(canny_edge11); colormap gray; axis image
```



```
bwlabel(canny_edge11)
```

```
ans =

     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     1     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     1     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
```

```
canny_edge7 = edge(imresult7,'canny',[0.05 0.25]);
figure; imagesc(canny_edge7); colormap gray; axis image
```



```
bwlabel(canny_edge7)
```

```
ans =

     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     1     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     1     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
```

Question 3

```
%{
As we have seen, Harris Corner detectors are invariant to rotation and hence (almost) same features must be detected when the image is rotated. We can see from the images below that the same features are detected when the image is rotated.


The harris edge detector is variant to scaling though and thus as we scale the image, the detected features in the image change. As we had discussed in the class, a line may look curved, but as the image is zoomed in, it may resemble close to a straighter line in the harris detector window. But making the image scalled to lower resolution does not seem to produce more features in every case. As the image becomes lesser r
%}

hc = harris_corners(imgray,7,1.5);
imshow(im); hold on;
```

```
 Warning: Image is too big to fit on screen; displaying at 67%
```

```
plot(hc(:,1),hc(:,2),'rs');
hold off
```

```
img_gray_rot = rgb2gray(imread('house1-rotated.jpg'));
hc_rot = harris_corners(img_gray_rot, 7, 1.5);
imshow('house1-rotated.jpg'); hold on;
plot(hc_rot(:,1), hc_rot(:,2), 'rs');
hold off;
```

```
h1 = imread('house1.jpg');
h1g = rgb2gray(h1);
hc1 = harris_corners(h1g,7,1.5);
imshow(h1); hold on;
plot(hc1(:,1),hc1(:,2),'rs');
hold off;
```

```
h12 = imread('house1-2down.jpg');
h12g = rgb2gray(h12);
hc12 = harris_corners(h12g,7,1.5);
imshow(h12); hold on;
plot(hc12(:,1),hc12(:,2),'rs');
hold off;
```

```
h14 = imread('house1-4down.jpg');
h14g = rgb2gray(h14);
hc14 = harris_corners(h14g,7,1.5);
imshow(h14); hold on;
plot(hc14(:,1),hc14(:,2),'rs');
hold off;
```

Question 4

%{

Here, we observe that SIFT is considerably able to find more features as compared to Harris Detector. In the first Image, the harris detector seems to work as good as the SIFT detector, but in the second and third correspondence matching, The harris detector performed not as good as SIFT since the images were scaled and rotated.

```
%{
If we are supposed to find a query.jpg image from the database db*.jpg, we will generate the matches of the query image with each and every image in the database.
```

```
We have noted that, when the image is matched with itself, it generates the most number of matching features. Thus, if we find the image with the maximum number of matches,

we can predict it to be the same (or closest) image to the query image.

%}
```

```
%}
```

```
%% Correspondences using SSD

I1 = imread('house1.jpg');
I2 = imread('house2.jpg');
I3 = imread('house1-rotated.jpg');
I4 = imread('house1-60own.jpg');
Ig1 = rgb2gray(I1);
Ig2 = rgb2gray(I2);
Ig3 = rgb2gray(I3);
Ig4 = rgb2gray(I4);
hc1 = harris_corners(Ig1,7,1.5);
hc2 = harris_corners(Ig2,7,1.5);
hc3 = harris_corners(Ig3,7,1.5);
hc4 = harris_corners(Ig4,7,1.5);
[f1,vp1] = extractFeatures(Ig1,hc1);
[f2,vp2] = extractFeatures(Ig2,hc2);
[f3,vp3] = extractFeatures(Ig3,hc3);
[f4,vp4] = extractFeatures(Ig4,hc4);
pair1 = matchFeatures(f1,f2,'Metric','SSD');
pair2 = matchFeatures(f1,f3,'Metric','SSD');
pair3 = matchFeatures(f4,f2,'Metric','SSD');
mp11 = vp1(pair1(:,1),:);
mp12 = vp2(pair1(:,2),:);
mp21 = vp1(pair2(:,1),:);
mp22 = vp3(pair2(:,2),:);
mp31 = vp4(pair3(:,1),:);
mp32 = vp2(pair3(:,2),:);
figure; showMatchedFeatures(I1,I2,mp11,mp12);
```

```
Warning: Image is too big to fit on screen; displaying at 67%
```

```
figure; showMatchedFeatures(I1,I3,mp21,mp22);
```

```
Warning: Image is too big to fit on screen; displaying at 50%
```

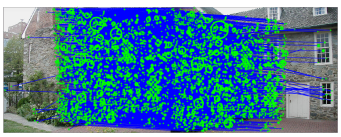
```
figure; showMatchedFeatures(I4,I2,mp31,mp32);
```

```
Warning: Image is too big to fit on screen; displaying at 67%
```

matchCorrespondences.m

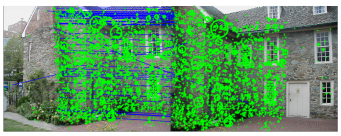
```
%{
This function takes as input 2 images along with the number of lines that need to be plotted on the appended image. It generates the appended image, plots

the feature points and the plots lines connecting the corresponding feature points in each image. This utilizes the vlfeat as described in the question.


Here, I have posted the images displaying lines for all corresponding features followed by lines limited to 50, so as to clearly see the matching of the feature points.



function  matchCorrespondences( Ia, Ib, limit)
Ig1 = single(rgb2gray(Ia));
Ig2 = single(rgb2gray(Ib));
[fa, da] = vl_sift(Ig1);
[fb, db] = vl_sift(Ig2);
[matches,s1] = vl_ubcmatch(da,db);
figure(2) ; clf ;
Ic = appendimages(Ia, Ib);
imshow(Ic); hold on;
xa = fa(1,matches(1,:)) ;
xb = fb(1,matches(2,:)) + size(Ia,2) ;
ya = fa(2,matches(1,:)) ;
yb = fb(2,matches(2,:)) ;
if limit > 0
xa = xa(1:limit);
ya = ya(1:limit);
xb = xb(1:limit);
yb = yb(1:limit);
end
hold on ;
h = line([xa ; xb], [ya ; yb]) ;
set(h,'linewidth', 1, 'color', 'b') ;
vl_plotframe(fa(:,matches(1,:)) ) ;
fb(1,:) = fb(1,:) + size(Ia,2) ;
vl_plotframe(fb(:,matches(2,:)) ) ;
axis image off ;
end
%}
Ia = imread('house1.jpg');
Ib = imread('house2.jpg');
Ic = imread('house1-rotated.jpg');
Id = imread('house1-60own.jpg');
matchCorrespondences(Ia, Ib,0);
```

```
Warning: Image is too big to fit on screen; displaying at 33%
```



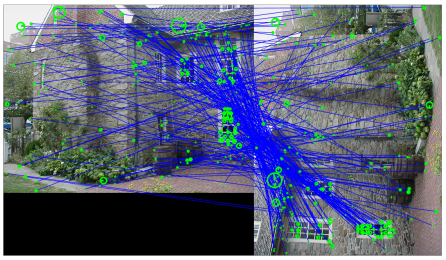
```
matchCorrespondences(Ia, Ib,50);
```

```
Warning: Image is too big to fit on screen; displaying at 33%
```



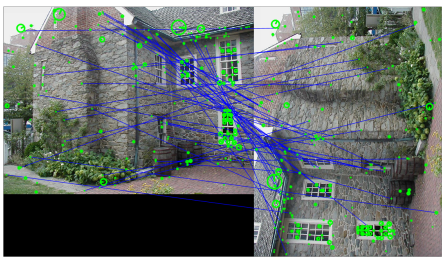
```
matchCorrespondences(Ia, Ic,0);
```

```
Warning: Image is too big to fit on screen; displaying at 50%
```



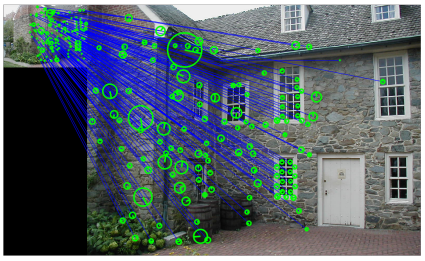
```
matchCorrespondences(Ia, Ic,50);
```

```
Warning: Image is too big to fit on screen; displaying at 50%
```



```
matchCorrespondences(Id, Ib,0);
```

```
Warning: Image is too big to fit on screen; displaying at 67%
```



```
matchCorrespondences(Id, Ib,50);
```

```
Warning: Image is too big to fit on screen; displaying at 67%
```