ASSIGNMENT A6

Shantnu Kakkar

CS 6320, Spring 2016

March 09, 2016

Section 1: Intro:

This assignment is based on the edge tracking and feature detection (corners) techniques learnt in the lecture. I will be implementing various method, for example **Harris corner detector, orientation histogram, algorithm 5.2 and 5.3 from text**. The basic idea behind my approach is that as the light gets brighter or darker, the image will get brighter or darker. This means that the gradient scales with the image. This creates a problem for edge detectors since they rely on image gradient magnitude. To overcome that problem, we use orientation of the image gradient, which is unaffected by scaling. The following figure 1 from the text proves this. Orientation histogram is a good technique that will be used for this.

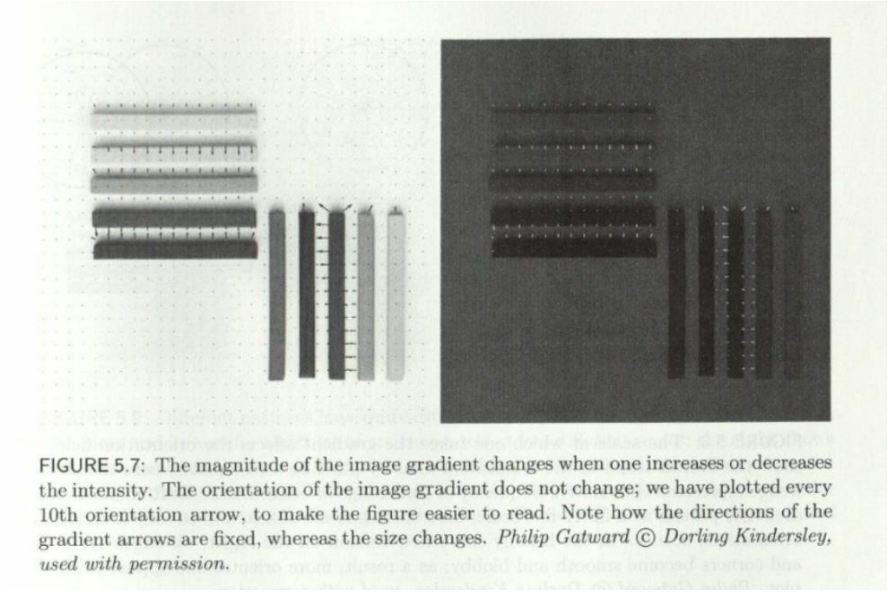


FIGURE 5.7: The magnitude of the image gradient changes when one increases or decreases the intensity. The orientation of the image gradient does not change; we have plotted every 10th orientation arrow, to make the figure easier to read. Note how the directions of the gradient arrows are fixed, whereas the size changes. *Philip Gatward © Dorling Kindersley, used with permission.*

Figure 1 This figure forms the motivation for me to using image orientation for corner detection.

Another motivation is to use Laplacian of Gaussian since we need to estimate the radius (scale) of the circular patch. The radius estimate should get larger proportionally when the image gets bigger. We could center a blob of fixed appearance on the corner and then choose the scale to be the radius of the best fitting blob. An efficient way to do that us Laplacian of Gaussian.

For this problem, we are given two images and we need to find corners in them. We are required to implement algorithm 5.2 and 5.3 and compare them as feature detectors by implementing on the two given images. After the implementation, I will be answering the following question

- Are the two algorithms stable with respect to image rotation, translation and scaling?

Section 2: Method:

- Matlab is used to carry out the experiments.
- For the interest points I will try to shoot between 20-40 points.
- I will be doing the translation, rotation and scaling part as follows:
  Translation:

  If the image is mxn, thenshift it right (with wraparound):

  ```
  imt = [im(:,51:n),im(:,1:50)];
  ```

  Rotation:

  ```
  im45 = imrotate(im,45);
  ```

  Scaling:

  ```
  im2 = imresize(im,2);
  ```
- I will be using the combo function provided by professor to analyse my results for Harris.
- I will be using imregionalmax function instead of defining my own function since it worked better for me compared to my function. I was able to get stable rotation and translation by using this function. There were some problems when I tried with my own local_max function. My own local max function will go into the critique section.

Following functions are implemented

- CS5320_Harris – This function detects the corners(interest points). It takes input as the gray image and the desired window size, and the output is the response matrix defined as:
  R (mxn array): corner response <=0: homogeneous , >0 and small: edge, large: corner
- CS5320_LoG_interest – It finds the interest points as well as the radius (maxima sigma scale) at those points. Its input is the gray image and a value p which helps us to set the threshold to reduce the number of threshold points. It outputs the interest points as 1 and other points as 0. It also outputs the corresponding max sigma scale at that location.
- CS5320_gradient_histogram – It gets the histogram of gradient orientations. Its input is the gray image, the row of center of patch, column of center of patch, radius of pixels to consider, minimum gradient magnitude to consider, and a Boolean which tells which tells whether to use magnitude of gradient as weight in histogram. It outputs orientation counts in 20-degree bins
- CS5320_corner_patches – It produces patches using Harris (Alg. 5.2 from text). See below for detailed description if this function.
- CS5320_LoG_patches – It produces patches using LogInterest (Alg. 5.3 text).). See below for detailed description if this function. It takes 30-40 seconds to run, but could be somewhat faster sometimes.

In addition to the above functions there is a script named verification which contains how I call every function. Please note that I have divided this script into various cells (sections). Please run individual section for every function. I have commented out imshow in this script everywhere. Please uncomment that to see the answers. For translation, rotation and scaling, uncomment from the corner patches and loG patches section to see the stability analysis. **Press ctrl + enter to run section wise.**

The following algorithm has been used for Harris:

- *Initialize R = zeros(size(im,1),size(im,2));*
- *Find gradient[dx,dy] = gradient(double(im));*
- *Loop for r = 1+k:size(im,1)-k*
  - *Loop for c = 1+k:size(im,2)-k*
    - *a = dx(r-k:r+k,c-k:c+k);*
    - *b = dy(r-k:r+k,c-k:c+k);*
    - *a = a(:);*
    - *b = b(:);*
    - *pts = [a,b];*
    - *M = pts'*pts;*

The following algorithm has been used for log interest

- *Make sigma vector sigma = 0.3:0.01:6;*
- *Initializations*

  scale = zeros(nr,nc);

  C = zeros(nr,nc,num_sigmas);

  interest_pts = zeros(nr,nc);

  maxResponses = zeros(nr,nc);

  temp1_scale = zeros(nr,nc);

  temp2_scale = zeros(nr,nc);

- *Loop for s_index = 1:num_sigmas*

  Make template T = fspecial('log', 21, sigma(s_index));

  Find response C(:,:,s_index) = abs(filter2(T,im));

- *Loop for r = 1:nr*
  - *loop for c = 1:nc*

```
temp1 = C(r,c,:);

temp1 = temp1(:);

[temp2(r,c),idx] = max(temp1);

if idx ~=1 && idx ~= num_sigmas

    temp1_scale(r,c) = sigma(idx);

    maxResponses(r,c) = temp2(r,c);
```

- *Find points above threshold*

```
maxvalues = p*max(max(maxResponses));

for r = 1:nr

    for c = 1:nc

    if maxResponses(r,c)<maxvalues

        maxResponses(r,c) = 0;
```

- *Find local maxima interest_pts = imregionalmax(maxResponses);*
- *Update scale = interest_pts.* temp1_scale;*

The following method is used for finding hands in CS5320_gradient_histogram function**:**

- Initialize *H = [0;0;0;0;0;0;0;0;0];*
- *Find window/patch = im(r-ceil(radius):r+ceil(radius),c-ceil(radius):c+ceil(radius));*
- *Gradient [dx,dy] = gradient(double(patch));*
- *Loop for r = 1:size(patch,1)*
  - *for c = 1:size(patch,2)*

```
magnitude = sqrt( dx(r,c)^2 + dy(r,c)^2);
if magnitude  > thresh
   orientation = (atan2d(dy(r,c),dx(r,c)));
   if orientation <0
      orientation = orientation + 180;
   end
   bin = ceil(orientation/20);
   if bin > 0
      if w==0
         H(bin) = H(bin) + 1;
      else
```

H(bin) = H(bin) + magnitude;

The following method is used in CS5320_corner_patches:

```
Initialize sigma = 0.3:0.01:6;
num_sigmas = length(sigma);
call R = CS5320_Harris(im,w);
call RregionalMax = imregionalmax(R);
R = RregionalMax.*R;
[nr,nc] = size(im);
Find threshold = max(max(R/2));
Make interest point vector x = R > threshold;
initialize patches = [];
loop for s_index  = 1:num_sigmas
   T = fspecial('log', 7, sigma(s_index));
   C(:,:,s_index) = abs(filter2(T,im));

loop for r = 1:nr
 loop   for c = 1:nc
      if x(r,c) ==1
         Xc = r;
         Yc = c;
         temp1 = C(r,c,:);
         temp1 = temp1(:);
         [temp2(r,c),idx] = max(temp1);
         radius = sigma(idx);
         H = CS5320_gradient_histogram(im,Xc,Yc,radius*k,0,0);
         [thetaP,index] = max(H);
         allThetaP = find(H==thetaP);
         for i = 1:length(allThetaP)
            tp(1) = Xc;
            tp(2) = Yc;
            tp(3) = r;
            tp(4) = 20 * allThetaP(i);
            patches = [patches;tp];
```

The following method is used in CS5320_loG_patches:

```
Initialize patches = [];

Find interest point and radius [x,A_scale] = CS5320_LoG_interest(im,0.7);

[nr,nc] = size(im);

Loop for r = 1:nr

 Loop   for c = 1:nc

      if x(r,c) ==1

         row of corner point Xc = r;

         column of corner point Yc = c;
```

```
radius = A_scale(r,c);
call   H = CS5320_gradient_histogram(im,Xc,Yc,radius*k,0,0);
[thetaP,index] = max(H);
allThetaP = find(H==thetaP);
loop  for i = 1:length(allThetaP)
        tp(1) = Xc;
        tp(2) = Yc;
        tp(3) = r;
        tp(4) = 20 * allThetaP(i);
        patches = [patches;tp];
```

Section 3: Verification:

Testing CS5320_Harris

- In testing for Haris, there IS A PROBLEM THAT gradient needs to be computed, which is difficult by hand. Hence, to find the gradient, I will be using matlab. Other calculations will be done by hand. Following figure shows hand calculations

For pixel (2,2),

The window is

$$w = im(1:3, 1:3) = \begin{bmatrix} 151 & 151 & 151 \\ 151 & 151 & 151 \\ 152 & 152 & 152 \end{bmatrix}$$

Gradient at these points in x

$$dx = \begin{bmatrix} 0 & 0 & 0.5 \\ 0 & 0.5 & 0.5 \\ 0 & 0 & 0 \end{bmatrix}$$

Gradient in y

$$dy = \begin{bmatrix} 0 & 0 & 1 \\ 0.5 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 \end{bmatrix}$$

$$\therefore \quad pts = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.5 & 1 \\ 0 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0 & 0.5 \\ 0 & 1 \\ 0 & 0.5 \end{bmatrix}$$

$$M = pt' * pt = \begin{bmatrix} 0.75 & 1 \\ 1 & 3.5 \end{bmatrix}$$

$$R(2,2) = det(M) - 0.05 * (trace(M)/2)^2$$

$$= (2.65 - 1) - 0.05 * \left(\frac{4.25}{2}\right)^2$$

$$= 1.65 - 0.05 * (2.125)^2$$

$$= 1.23$$

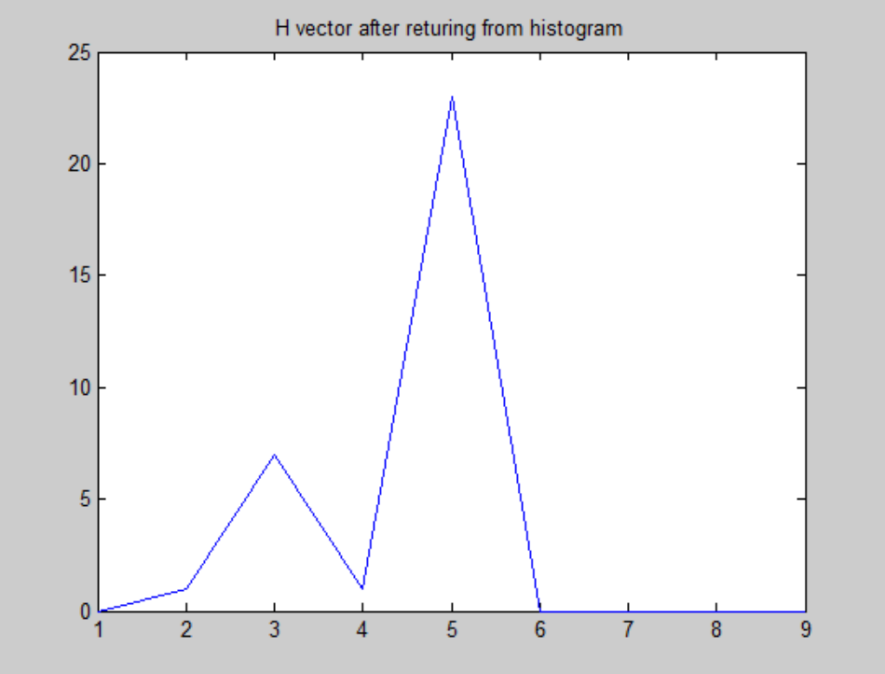
- From the above two images we can see that R(2,2) is 1,23. I am getting same in the matlab window:

R <288x466 double>

	1	2	3
1	0	0	0
2	0	1.2375	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0

Testing **CS5320_gradient_histogram**

I am able to correctly get my histogram as follows



Testing **log_interest function**

- This function should output 20-40 interest points.The following command window shows that I am able to get interest points in this range.

```
Command Window
K>> clear
K>> im = imread('glass-box.jpg');
im = rgb2gray(im);
 [A_IP,A_scale] = CS5320_LoG_interest(im,0.7);
K>> NumofInterestpts = sum(sum(A_IP))

NumofInterestpts =

   38
```

Testing corner patches function

I am getting good results for this function as follows:

```
Command Window
K>>  im = imread('glass-box.jpg');
im = rgb2gray(im);
p = CS5320_corner_patches(im,2,2);
imshow(im);
hold on;
quiver(p(:,2),p(:,1),-1*cosd(p(:,4)),-1*sind(p(:,4)));
```

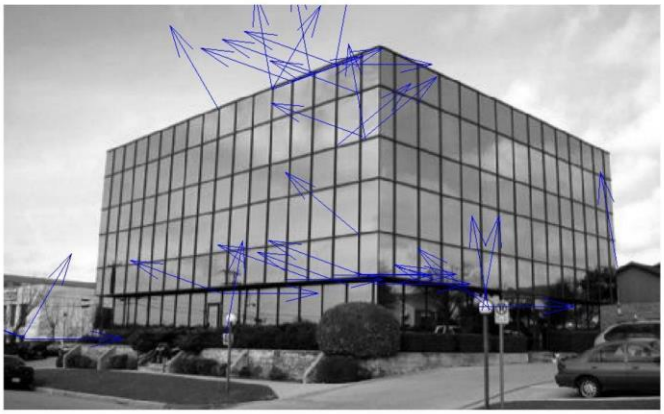


Figure 2 result for algo5.2

Testing loG patches function

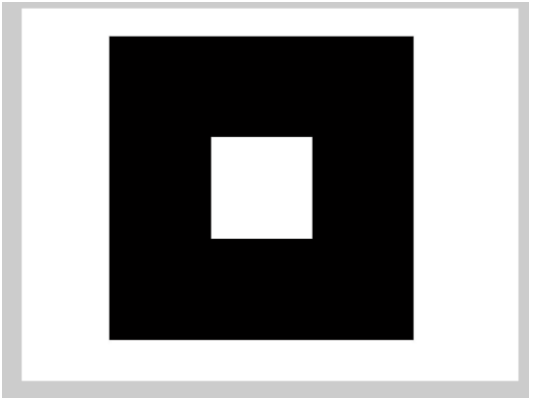
I am getting good results for this function as follows:

```
Command Window
K>> clear
K>> im = imread('glass-box.jpg');
im = rgb2gray(im);
K>> p = CS5320_LoG_patches(im,2,2);
imshow(im);
hold on;
quiver(p(:,2),p(:,1),-1*cosd(p(:,4)),-1*sind(p(:,4)));
```

Figure 3 result for algo 5.3

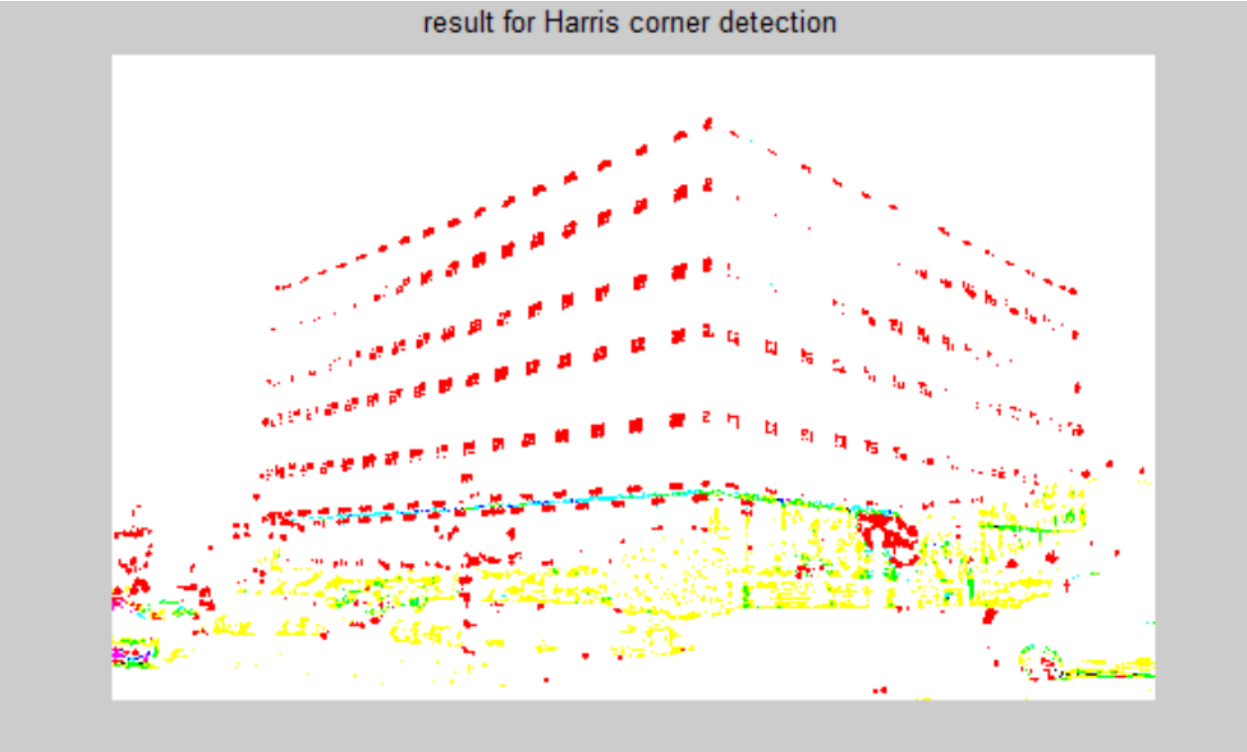
Section 4: Data:

The following two images are being used:



The histogram being formed has been shown in the verificiation dection

The following images show the result for Harris:



Please note that the combo function is not working fine for me since my matlab version is old. I ran this in matlab version 2015 and got the building correctly. Please run my Harris and you will know what I am talking about. I am calling it like this:

```
imoriginal = imread('glass-box.jpg');
im = imread('glass-box.jpg');
% imshow(im);
im = rgb2gray(im);
R = CS5320_Harris(im,1);
combo(double(imoriginal), R>max(max(R/25)));
title('result for Harris corner detection');
```

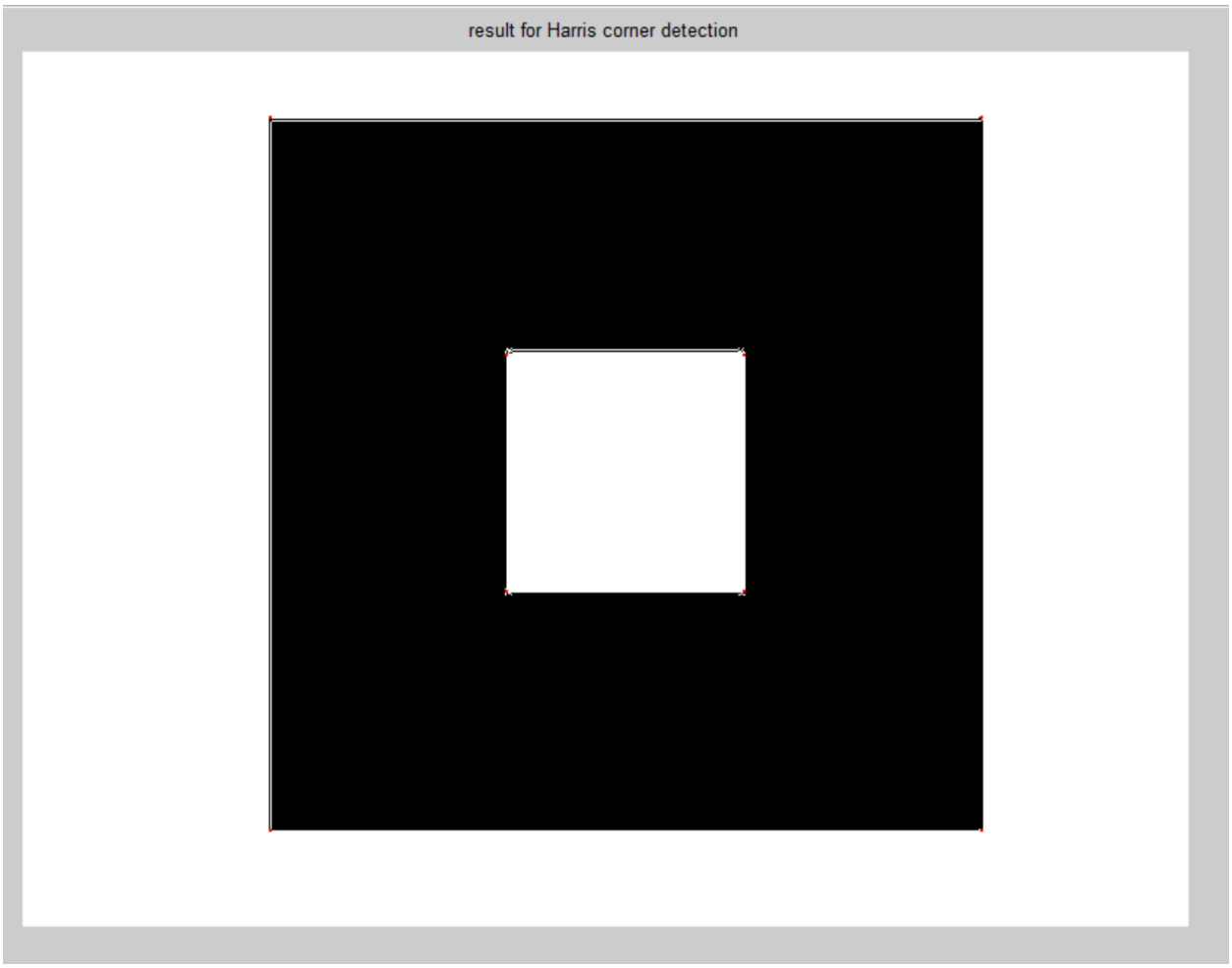
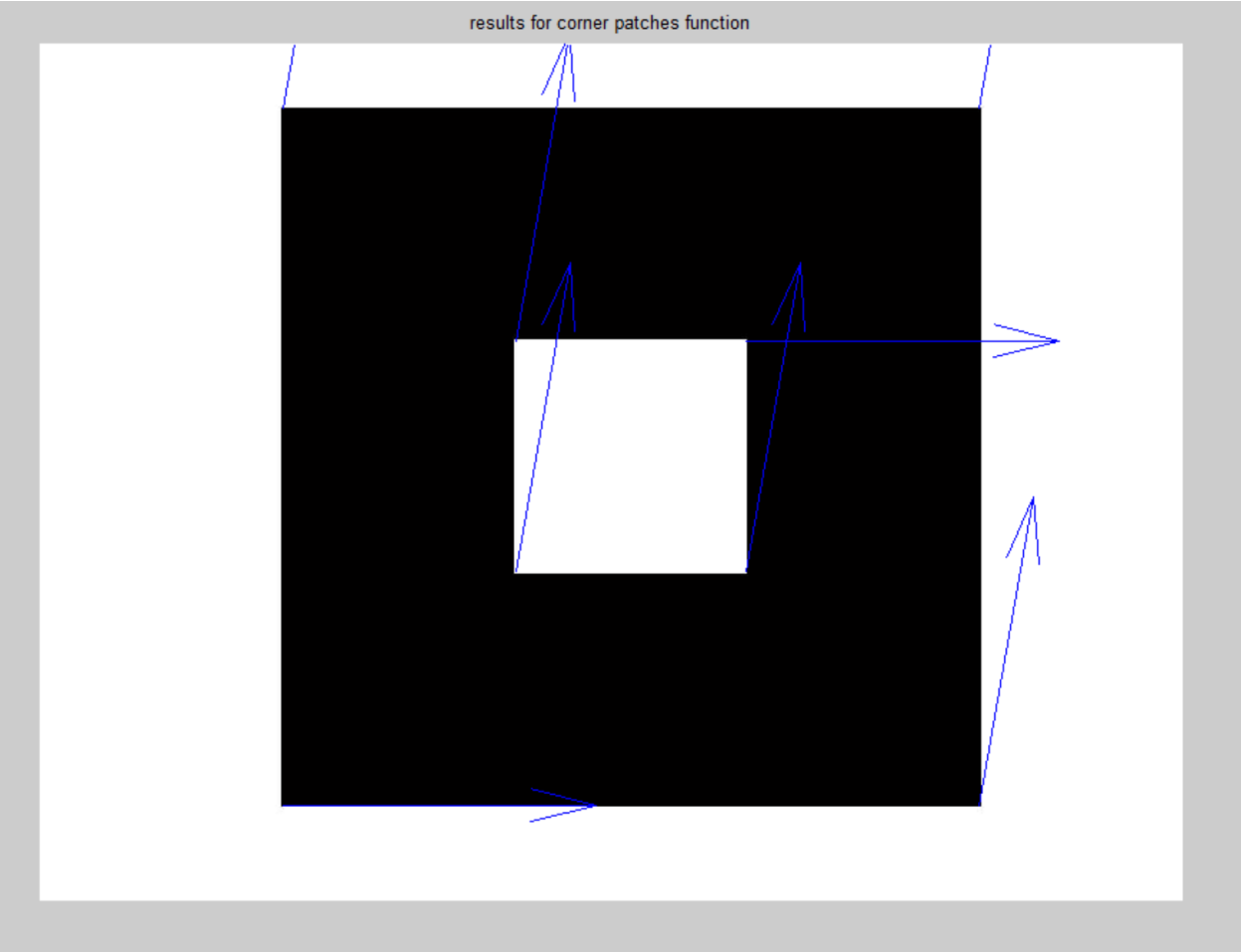


Figure 4 Please try to see the red dots on the corner. They there

Following is the result for CS5320_corner_patches:



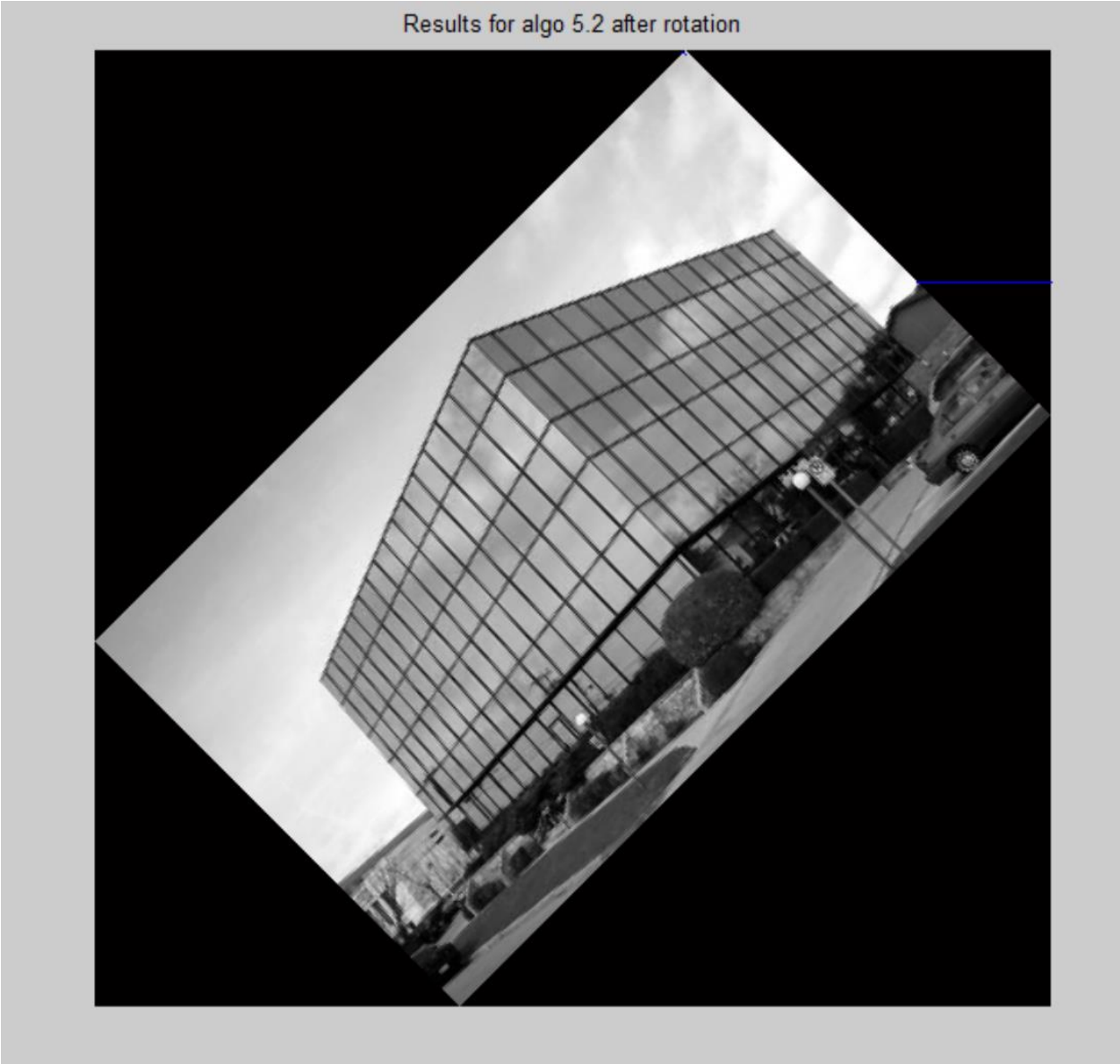
Section 5: Analysis:

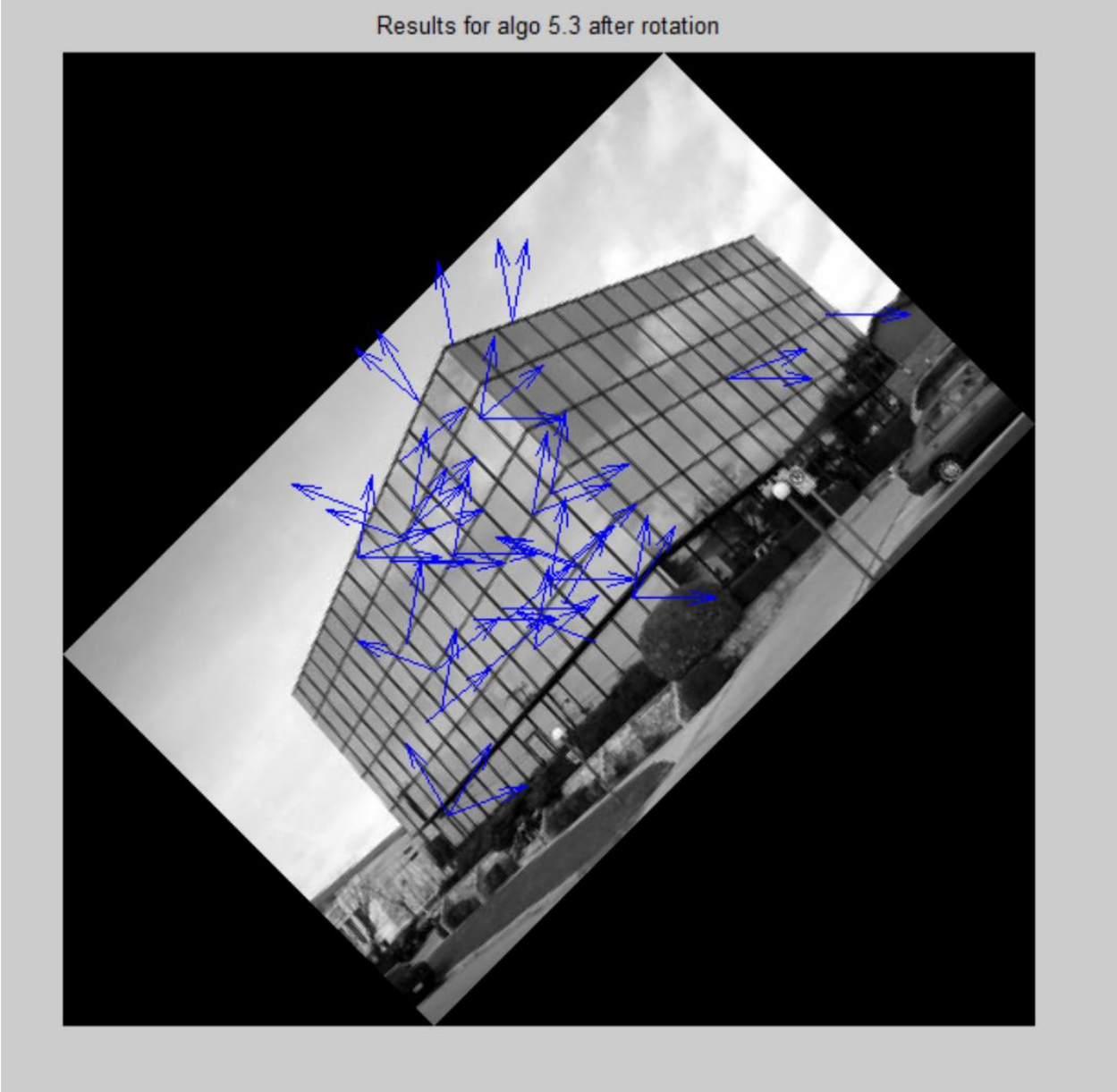
Now we will see the results for translation, rotation and scaling for CS5320_corner_patches function

1) Translation

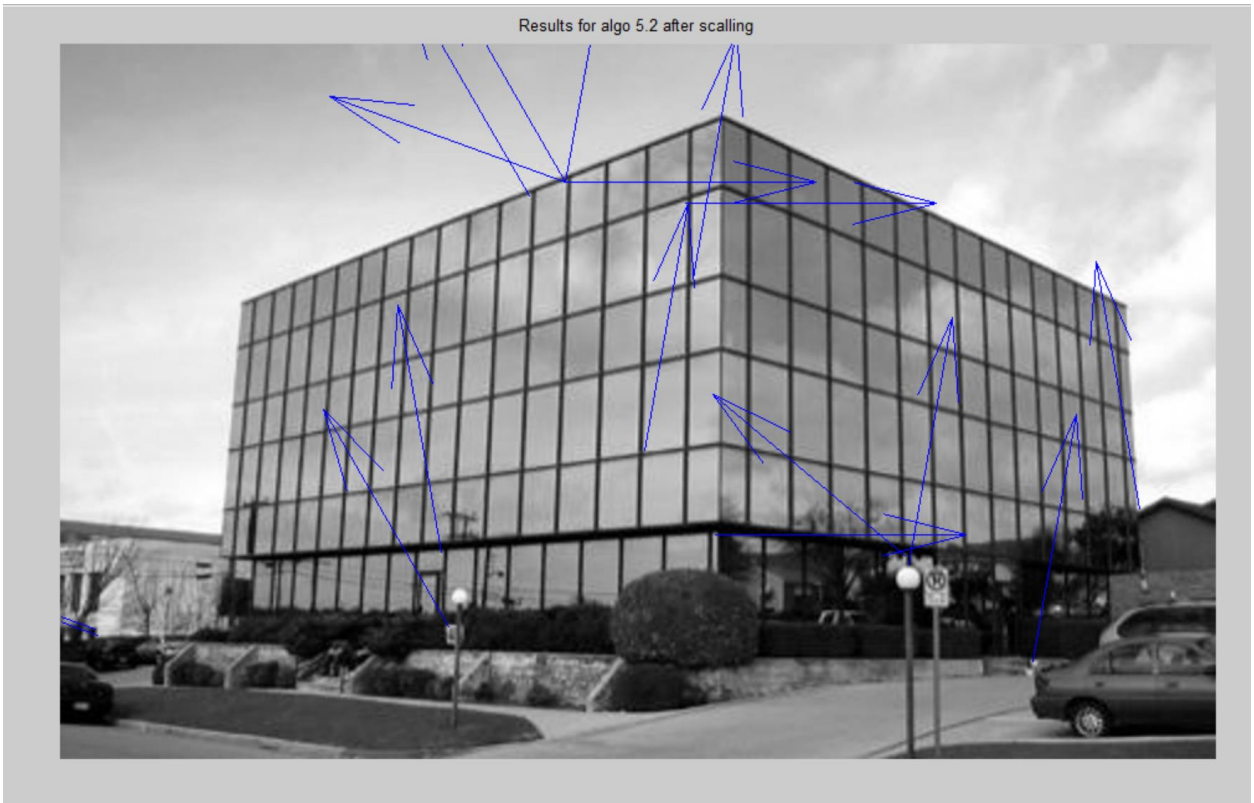

Results for algo 5.3 after translation

2) Rotation




Results for algo 5.3 after rotation

3) Scaling



Section 6: Interpretation:

The following are my observations:

- Orientation proved to be a powerful method for corner recognition
- Harris detection, although powerful, solely cannot be used for corner detection and getting orientation
- Both algo 5.2 and 5.3 are able to give me good number of corner points. But what I observed is that in algo 5.3, I can more flexibly choose the number of points that I want by playing around with p.
- **However, Algo 5.3 took more time to run.**
- Square image takes more time to run.
- **Translation and rotation are stable with algorithm 5.3**
- **Scaling is stable with algo 5.2**. **Scaling for algo5.3 is giving not so good results**(infact it is causing dynamic runtime problems). This makes me sceptical whether algo 5.3 actually is stable with a scaled image or not. This question is still unanswered from my observations.

Section 7: Critique:

The experiment could be improved by following ways:

- Implement my own local maxima function
- Couldn't get my algo 5.3 to properly work for a scaled image.

Section 8: log

18 hours total