Try to find a reasonable stapler
* missing pp. 18-19

B-

v c r

B

code: * not very good glass-box corners
* gradient doesn't work correct (square)
* no local_max function
* LoG_interest not correct
* LoG_patches does not work !

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.

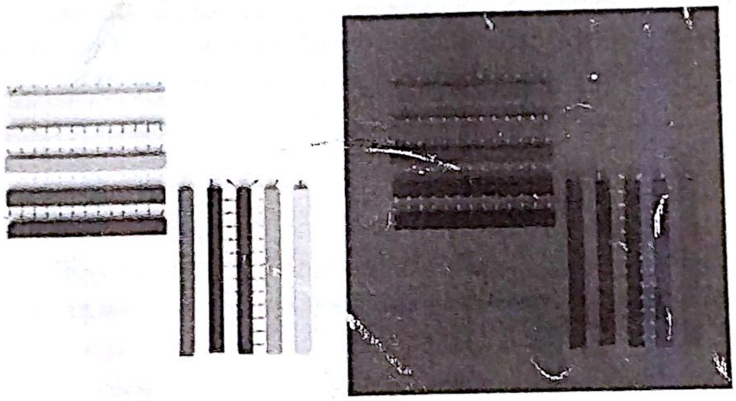


FIGURE 5.7: The magnitude of the image gradient changes when one increases or decreases the intensity. The orientation of the image gradient does not change; we have plotted every 10th orientation arrow, to make the figure easier to read. Note how the directions of the gradient arrows are fixed, whereas 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 is 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?

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;*

? missing computation of R

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));

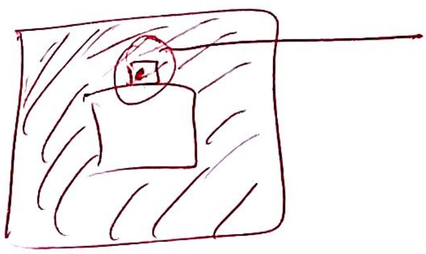
```
radius = A_scale(r,c);
call   H = CS5320_gradient_histogram(im,Xc,Yc,radius*k,0,0);
[thetaP,index] = max(H);
allThetaP = find(H==thetaP);
loop  for i = 1:length(allThetaP)
        tp(1) = Xc;
        tp(2) = Yc;
        tp(3) = r;
        tp(4) = 20 * allThetaP(i);
        patches = [patches;tp];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

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

Not really; give an image with simple gradients, e.g., a square



histogram has all 80-450 θ's

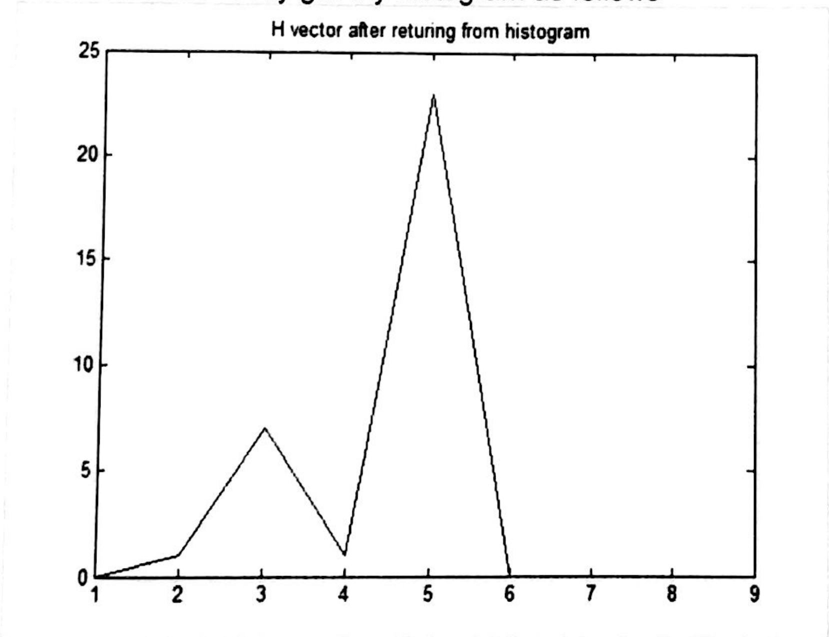
- From the above two images we can see that R(2,2) is 1.23. I am getting same in the matlab window:

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

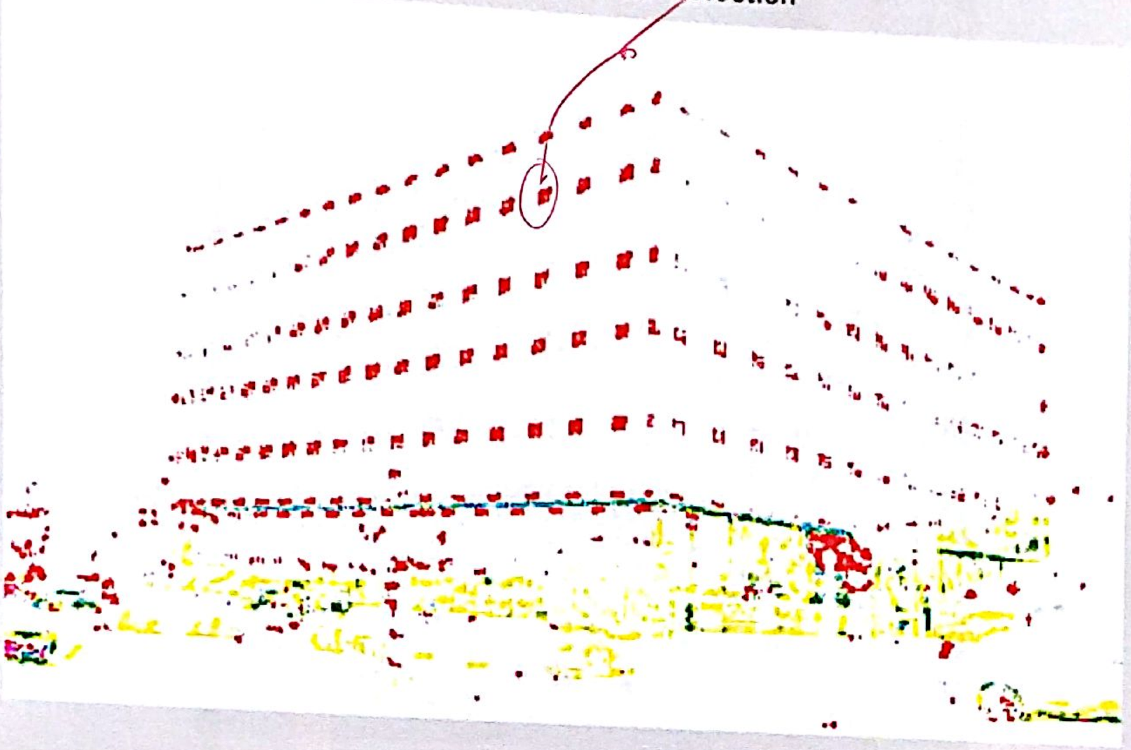


does not verify anything since we don't know what the input was nor what the output is

label axes
Put Figure captions

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.



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');
```

should use mat2gray(imoriginal)