Report on the First Homework (Convolutions, Fourier Transforms, and Image Pyramids)

Negar Nejatishahidin
Computer Science
George Mason University
Virginia, United states
nnejatis@gmu.edu

I. P2.1 Harris Corners

A. *P2.1.1 Computing Harris Corners*

In this part we asked for implementing harries corner detection. I have shown the following plots for 2 images(cube figure 3 and chess figure 4).

- The original image in figures 3(a) and 4(a).
- The x and y derivatives of the image in figure 3(b), 4(b), 3(c) and 4(c).
- The scoring function f in figures 3(d) and 4(d).
- The original image with dots showing the location of the detected corners in figures 3(e) and 4(e).

QUESTION A: Are the features where you expected? For the chess image fig. 4, the features' are the ones that I have expected, since the corners are obvious. However, with increasing the threshold we may loose some corners or vice-versa. But for the Cube image fig. 3, some of the corners did not detected according to my threshold. However, if I decrease the threshold to detect the missed corners, lots of wrong corners will be detected. So, I choose an optimal threshold.

QUESTION B (IMAGE REQUIRED): Are there any features in the image that you are surprised are not present? Yes, as you can see in fig. I-A there are some features that are not detected ,which are shown with white arrows. If I reduce the threshold, the missed features still not be detected instead some wrong features will be detected. The wrong features are shown with black arrow in fig. I-A.

QUESTION C (IMAGES REQUIRED): What would happen if you used a scoring function f=tr(H)=A+C ? The score function and the detected features are shown in figure **??** for both the cube and chess image.

QUESTION D: What does this scoring function detect? This scoring function detects the edges of the image, since It is the sum of the x and y derivative of the image, so the edge of the image will be detected.

QUESTION E: If we want to detect corners, why might we not want to use this scoring function? Since It detects both the edges and the corners, but we only want to detect the corners.

Identify applicable funding agency here. If none, delete this.

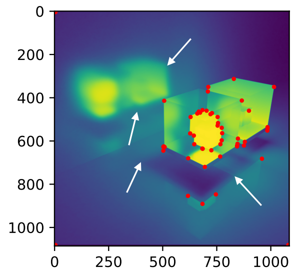


Fig. 1. QUESTION B with threshold 0.009.

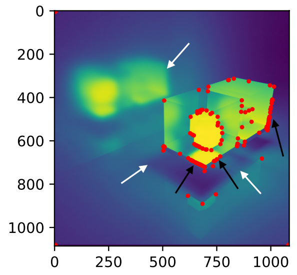
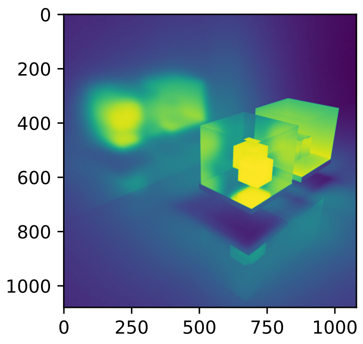


Fig. 2. QUESTION B with threshold 0.003.

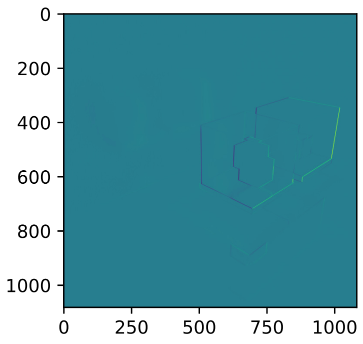
B. *P2.1.2 Varying the Weight Matrix*

I have plotted the score function and the corners of image for the following wight matrices :

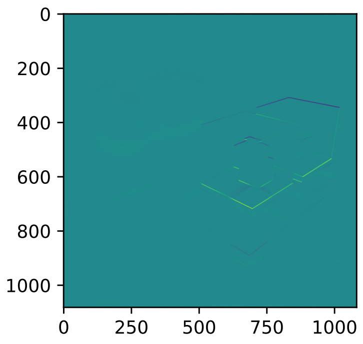
- A uniform weight matrix of size 5x5 weights in figures 3(d) and 3(e).
- A uniform weight matrix of size 25x25 weights in figures 6(a) and 6(d).
- A Gaussian weight matrix with =5 in figures 6(b) and 6(e).
- A Gaussian weight matrix with =50 in figures 6(c) and 6(f).



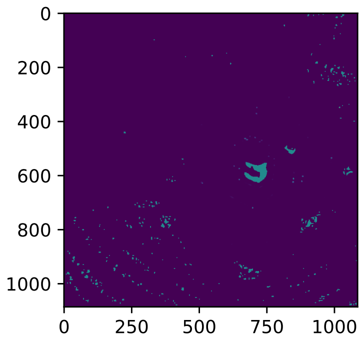
(a) The original image.



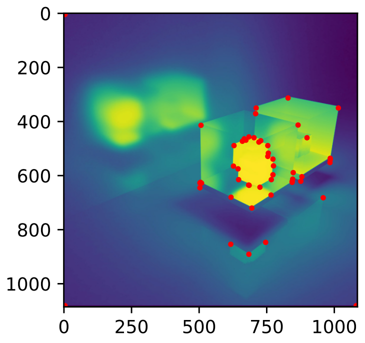
(b) The x derivatives of the image.



(c) The y derivatives of the image.

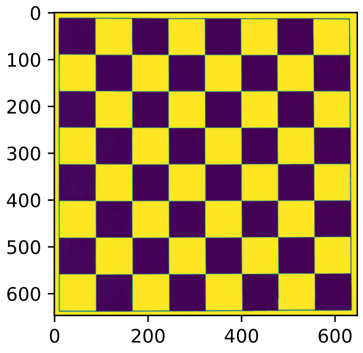


(d) The scoring function f.

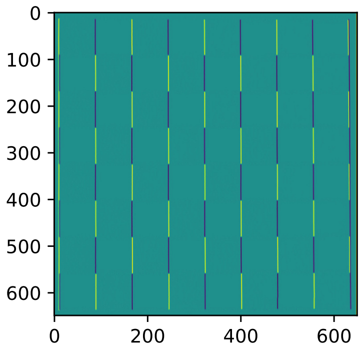


(e) The original image with dots showing the location of the detected corners.

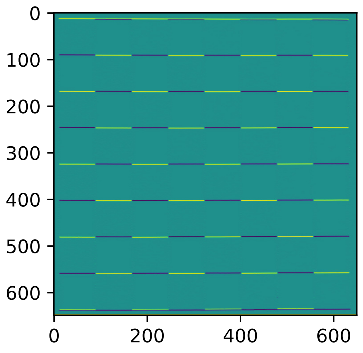
Fig. 3. Plots of figure cube.



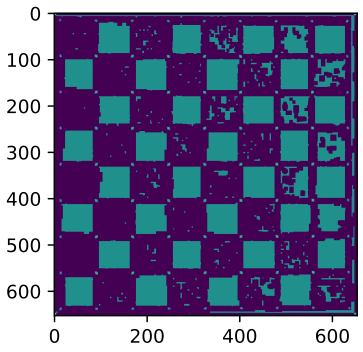
(a) The original image.



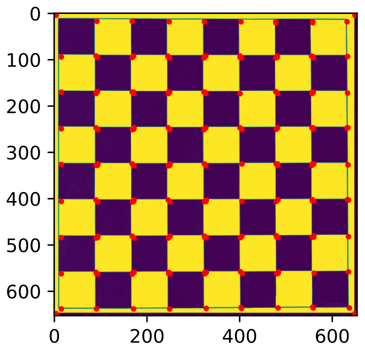
(b) The x derivatives of the image.



(c) The y derivatives of the image.

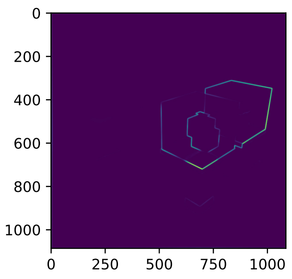


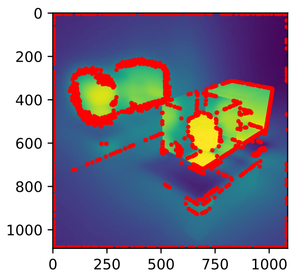
(d) The scoring function f.

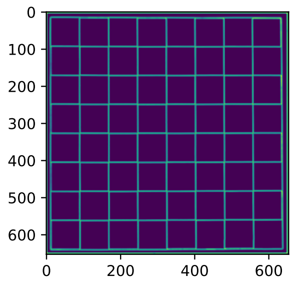


(e) The original image with dots showing the location of the detected corners.

Fig. 4. Plots of figure chess.


(a) The scoring function trace.


(b) The original image with dots showing the location of the detected corners.


(c) The scoring function trace.

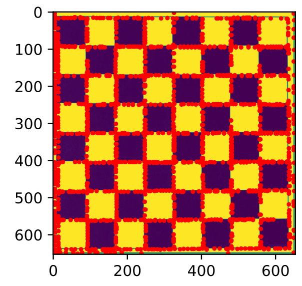
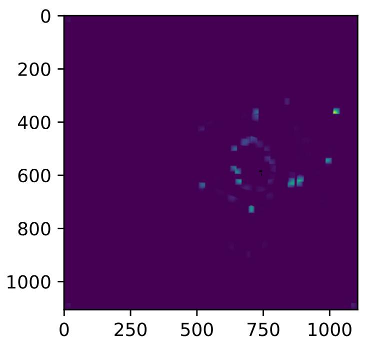
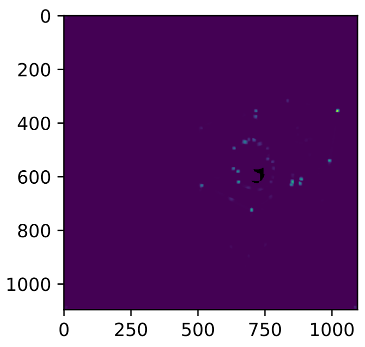
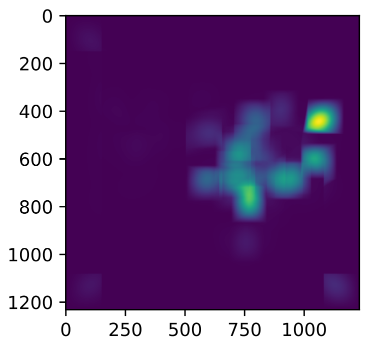
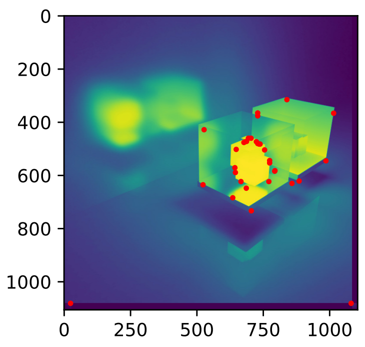

(d) The scoring function f.

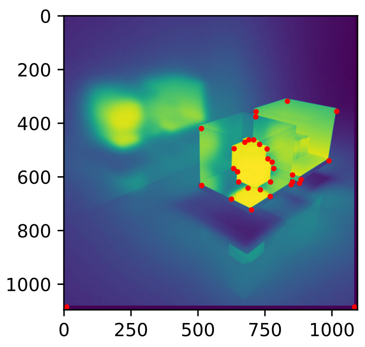
Fig. 5. Trace as a score function on cube and chess images.


(a) Score function with a uniform weight matrix of size 25x25.


(b) Score function with a Gaussian weight matrix with $\sigma = 5$.


(c) Score function with a Gaussian weight matrix with $\sigma = 50$.


(d) Corners with a uniform weight matrix of size 25x25.


(e) Corners with a Gaussian weight matrix with $\sigma = 5$.

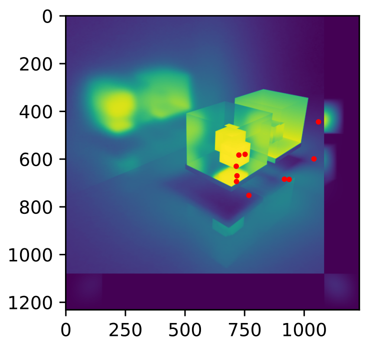

(f) Corners with a Gaussian weight matrix with $\sigma = 50$.

Fig. 6. Score functions and corners using different wight matrices.

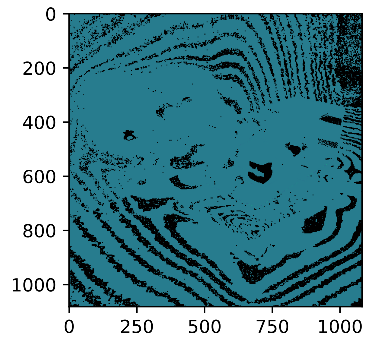
QUESTION A: Discuss the differences between these four weight functions. In particular, what happens when the filter width is very large? As the kernel get bigger the less points detected as corners. The bigger wight function causes that image get smoother. Therefore, only the obvious corners still remain in corner detection.

QUESTION B: What happens if we were to use a 1x1 weight matrix w=1? Why does this occur? If we choose a 1*1 wight matrix, the range of the score function will be reduce since know It is the difference between 2 pixel instead of a 5*5 kernel. Therefore, for 1*1 kernel with a really small threshold ”0.00000000000000001” many pixels are miss classified as a corner but they are edge (with bigger than this threshold no corner will be detected). In fact, the score function for pixels are too small and the value of them are close to each other. Shown in figure **??** and **??**.

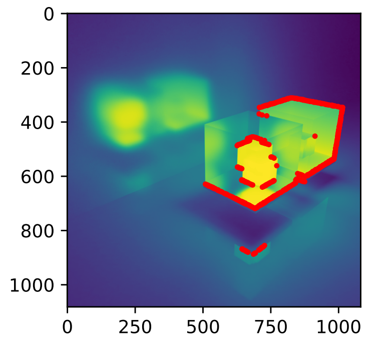
II. P2.2 Multi-scale Blob Detection

A. *P2.2.1 Scale-Normalized Filter Response*

Six different of the filtered images are plotted in figure II-A. Also the graph of the response with respect to the sigma is shown in figure II-A.As you can see in the image the response



(a) Score function with a uniform weight matrix of size 1x1.



(b) Corners with a uniform weight matrix of size 1x1.

Fig. 7. Score functions and corners using different wight matrices.

for the sigma = 14 is the max. QUESTION A : the relation ship between sigma and radius is as follow :

$$\sigma = \frac{R}{\sqrt{2}}$$

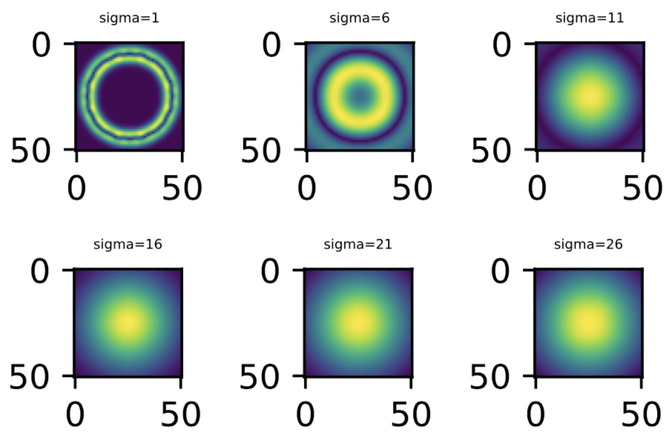


Fig. 8. Six different filtered images.

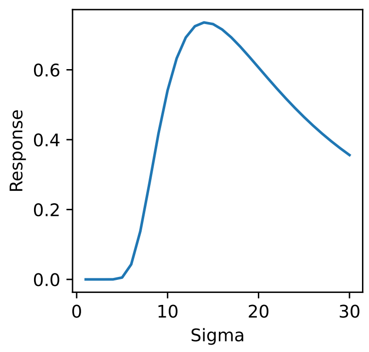


Fig. 9. filter response

B. P2.2.2 Annotating an Image with Multi-Scale Detections

The result is in figure 10. Also you can see the the algorithm applied on sunflower image in figure II-B. The threshold is 0.09 and the sigma is between 6 and 40, each time added by 4. The result on other image is in figure II-B.

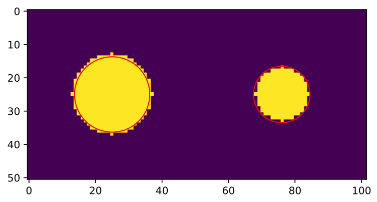


Fig. 10. The location of the extrema plotted on top of the original image.

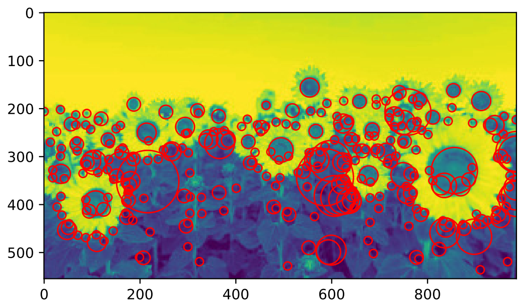


Fig. 11. finding circles on Sunflower image.

III. P2.3 Image Warping

The transformation matrices are as follow:

- The identity (figure III):

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- A rotation by 30 degrees (figure III):

$$\begin{bmatrix} \frac{\pi}{6} & -\sin\frac{\pi}{6} & 0 \\ \sin\frac{\pi}{6} & \frac{\pi}{6} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

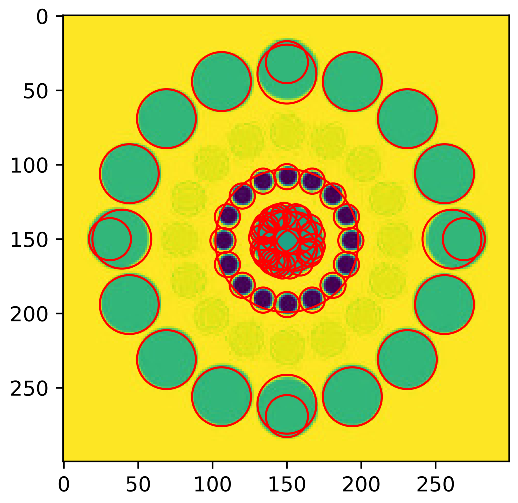


Fig. 12. finding circles on Sunflower image.

- A rotation by 30 degrees and translated to the center (figure III) :

$$\begin{bmatrix} \frac{\pi}{6} & -\sin\frac{\pi}{6} & \frac{img_{length}}{2} \\ \sin\frac{\pi}{6} & \frac{\pi}{6} & \frac{img_{length}}{2} \\ 0 & 0 & 1 \end{bmatrix}$$

- Scale by a factor of 2 along the x-axis (figure III):

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- This kernel (figure III): This kernel rotate the image with $\frac{\pi}{4}$ and then scale it with factor $\sqrt{2}$.

$$\begin{bmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

QUESTION A: The Affine transformation in general will be written as follow:

$$\begin{bmatrix} r_1 & r_2 & t_x \\ r_3 & r_4 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

The r_1 through r_4 are causes the image to be rotated. The t_x translate the image along x and t_y translate it along y axis.

The results of 2 different Homographies kernels are shown in figure III and III. QUESTION B: As you can see in figure III, the multiplication of the image with this transformation matrix, sets a depth value for the image pixels. Therefore, when we divide the output with the Z value again to transform it to the $[x, y, 1]$ form (to projected into image plane) again, the value of x and y are divided by the depth on that pixel.

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} \frac{ax+by+c}{gx+hy+1} \\ \frac{dx+ey+f}{gx+hy+1} \\ 1 \end{bmatrix}$$

Fig. 13. Homographies.

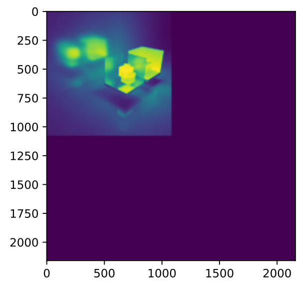


Fig. 14. Homographies.

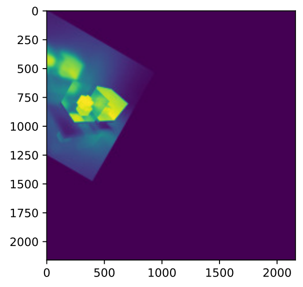


Fig. 15. Homographies.

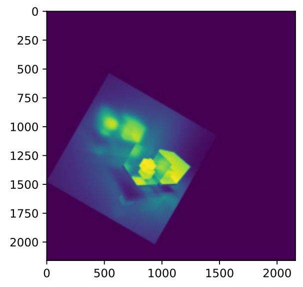


Fig. 16. Homographies.

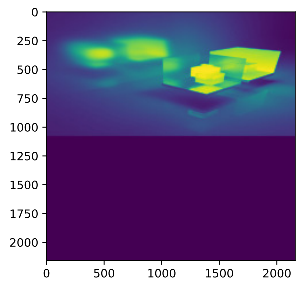


Fig. 17. Homographies.

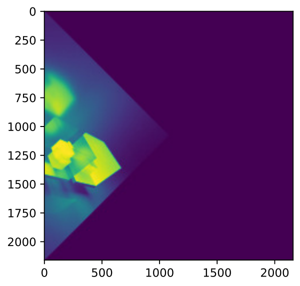


Fig. 18. Homographies.

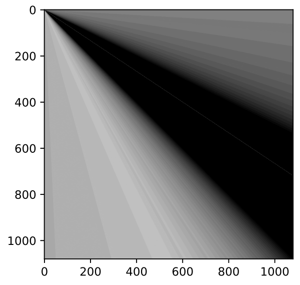


Fig. 19. Homographies 1.

IV. P2.4 Implementing Some Simple Feature Descriptors

The results are shown in figure IV, IV , and IV.

QUESTION A : The Histogram feature descriptor perform worst here, however, this was expected since it shows the image color intensity. Therefore when the image color change, the value of Histogram feature descriptor also change.

QUESTION B : Image Histogram feature descriptor perform the best for the rotation, since the rotation does not have effect on histogram of features. Therefore, it can perfectly match the features.

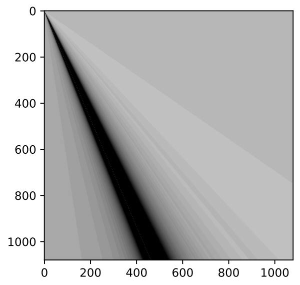


Fig. 20. Homographies 2.

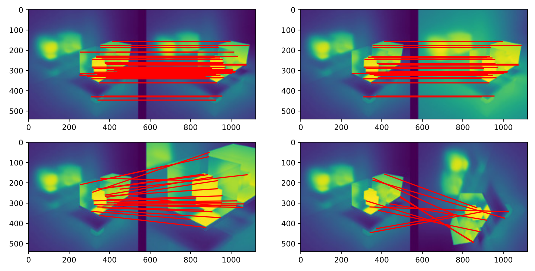


Fig. 21. Match

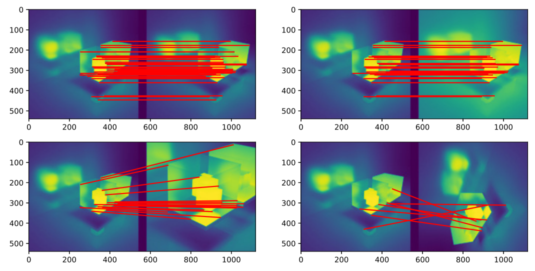


Fig. 22. Binary.

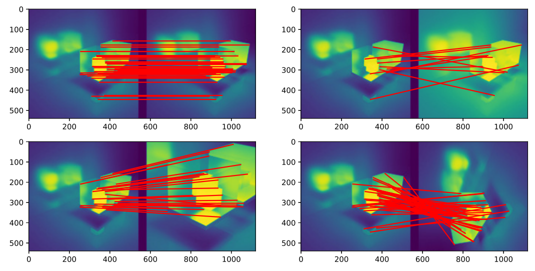


Fig. 23. Histogram