

Name : Aum M. Dhabalia

Date : 07/10/2024

Roll No. : 21BEC027

Experiment...7 – Edge Detection, Line Detection and Corner Detection

Objective :

- Find the in the image.
- Corner detection with Harris corner detector.
- Line detection.

Import necessary libraries...

'''

Created on 7 October 2024 Mon 2:01:55 pm

'''

```
import numpy as np
```

```
import cv2
```

```
import matplotlib.pyplot as plt
```

```
I = cv2.imread(r"D:\Nirma Files\Computer Vision\Experiments\BO6.jpeg")
```

```
Ig = cv2.cvtColor(I,cv2.COLOR_BGR2GRAY)
```

Task 1. Compute the edge detection using Sobel, Prewitt and Canny operator.

```
x = int(input("Enter first threshold..."))
```

```
y = int(input("Enter second threshold..."))
```

```
edges = cv2.Canny(Ig,x,y)
```

```
plt.subplot(121),plt.imshow(cv2.cvtColor(I,cv2.COLOR_BGR2RGB))
```

```
plt.title('Original Image')
```

```
plt.subplot(122),plt.imshow(edges,cmap = "gray")
```

```
plt.title('Canny Edge Detection')
```

```
plt.show()
```

```
xPrewitt = np.array([[1,0,-1],[1,0,-1],[1,0,-1]])
```

```
yPrewitt = np.array([[1,1,1],[0,0,0],[-1,-1,-1]])
```

```
xEdges = cv2.filter2D(Ig,-1,xPrewitt)
```

```
yEdges = cv2.filter2D(Ig,-1,yPrewitt)
```

```
edgesPrewitt = cv2.addWeighted(xEdges,0.5,yEdges,0.5,0)
```

```
plt.subplot(221),plt.imshow(cv2.cvtColor(I,cv2.COLOR_BGR2RGB)),plt.title('Original Image')  
plt.subplot(222),plt.imshow(edgesPrewitt,cmap = "gray"),plt.title('Prewitt Edge Detection')  
plt.subplot(223),plt.imshow(yEdges,cmap = "gray"),plt.title('Prewitt Edge Detection along y-axis')  
plt.subplot(224),plt.imshow(xEdges,cmap = "gray"),plt.title('Prewitt Edge Detection along x-axis')  
plt.show()
```

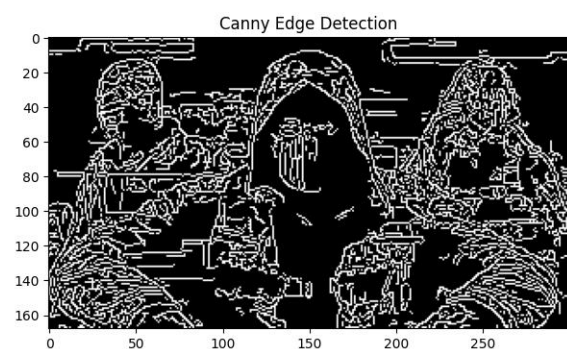
```
xSobel = cv2.Sobel(Ig,cv2.CV_64F,1,0,ksize = 3)  
ySobel = cv2.Sobel(Ig,cv2.CV_64F,0,1,ksize = 3)
```

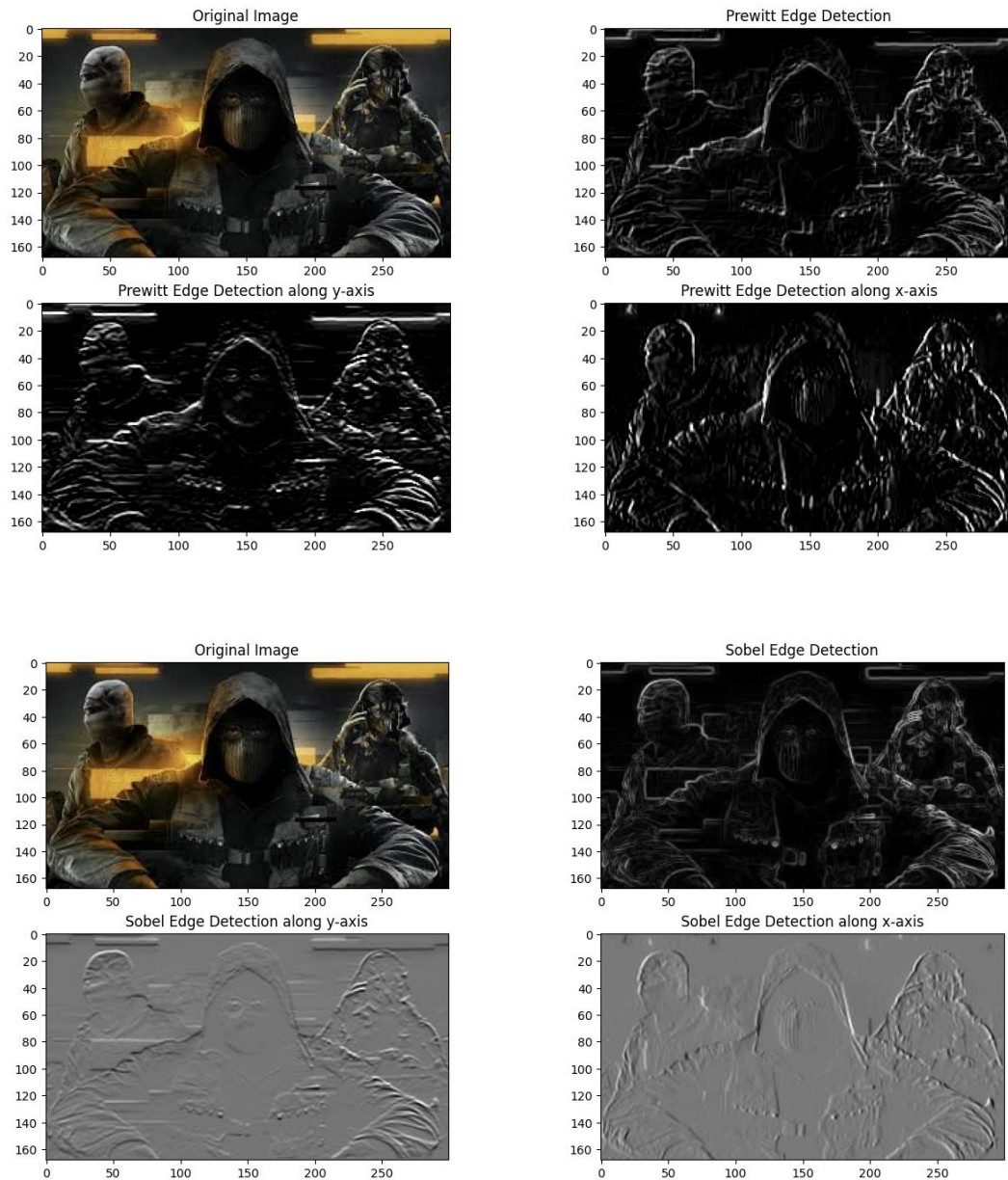
```
edgesSobel = cv2.magnitude(xSobel,ySobel)
```

```
plt.subplot(221),plt.imshow(cv2.cvtColor(I,cv2.COLOR_BGR2RGB)),plt.title('Original Image')  
plt.subplot(222),plt.imshow(edgesSobel,cmap = "gray"),plt.title('Sobel Edge Detection')  
plt.subplot(223),plt.imshow(ySobel,cmap = "gray"),plt.title('Sobel Edge Detection along y-axis')  
plt.subplot(224),plt.imshow(xSobel,cmap = "gray"),plt.title('Sobel Edge Detection along x-axis')  
plt.show()
```

Output

```
Enter first threshold...50  
Enter second threshold...88
```





Observation :

- Edges were detected using Sobel, Prewitt and Canny operators.
- In Sobel operator, horizontal edges are detected when applied in y direction and vertical edges are detected when applied in x direction.
- In Prewitt operator, vertical edges are detected when applied in y direction and same for x direction when applied.

Task 2. Implement Harris corner detector algorithm to determine the corner in the image.

```
I = cv2.imread(r"D:\Nirma Files\Computer Vision\Experiments\3d shapes.png")
```

```
Ig = cv2.cvtColor(I,cv2.COLOR_BGR2GRAY)
```

```
Ig = np.float32(Ig)
```

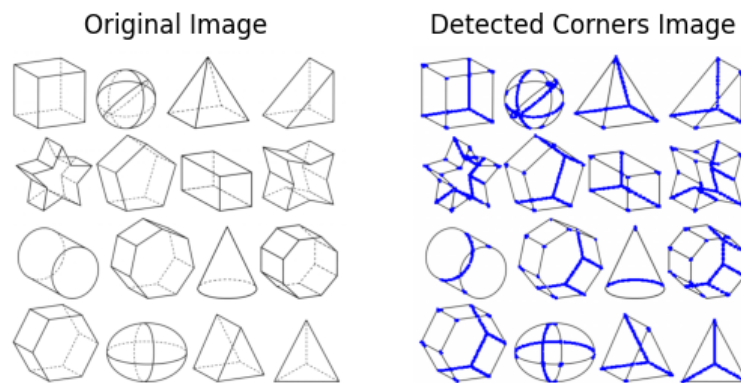
```
dst = cv2.cornerHarris(Ig,2,3,0.04)
```

```

dst = cv2.dilate(dst,None)
I[dst>0.01*dst.max()]=[0,0,255]
plt.subplot(121),plt.imshow(Ig,cmap = 'gray'),plt.title("Original Image")
plt.axis("off")
plt.subplot(122),plt.imshow(I),plt.title("Detected Corners Image")
plt.axis("off")
plt.show()

```

Output



Observation :

- The fundamental matrix was computed using 8 point algorithm.

Task 3. Implement Harris corner detector algorithm without the inbuilt OpenCV() function.

```

def harris_corner_detection(image, window_size=3, k=0.04):
    Ix = cv2.Sobel(image, cv2.CV_64F, 1, 0, ksize=3)
    Iy = cv2.Sobel(image, cv2.CV_64F, 0, 1, ksize=3)
    Ixx = Ix**2
    Iyy = Iy**2
    Ixy = Ix * Iy
    #Apply a Gaussian filter
    G = cv2.GaussianBlur(Ixx, (window_size, window_size), 0)
    H = cv2.GaussianBlur(Iyy, (window_size, window_size), 0)
    Ixy = cv2.GaussianBlur(Ixy, (window_size, window_size), 0)
    #Compute the response of the corner detector
    det_M = (G * H) - Ixy**2
    trace_M = G + H
    R = det_M - k * (trace_M**2)

```

```

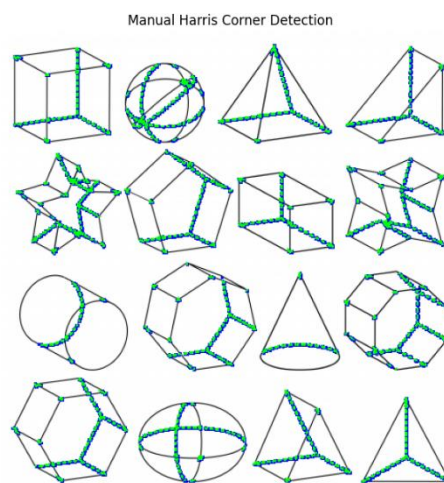
return R

#Apply Manual Harris Corner Detection
corners = harris_corner_detection(Ig)
corners[corners > 0.01 * corners.max()] = 255

# Plot Manual Harris Corner Detection Result
plt.imshow(corners, cmap='gray')
plt.title('Manual Harris Corner Detection')
plt.axis("off")
plt.show()

```

Output



Observation :

- The green marker are the detected corners using manual Harris corner detection.

Task 4. Detect the line using Hough transform.

```

I = cv2.imread(r"D:\Nirma Files\Computer Vision\Experiments\chess.png")
I1 = I
Ig = cv2.cvtColor(I,cv2.COLOR_BGR2GRAY)
edges = cv2.Canny(Ig,50,150,apertureSize = 3)
lines = cv2.HoughLines(edges,1,np.pi/180,200)

for line in lines:
    rho,theta = line[0]
    a = np.cos(theta)
    b = np.sin(theta)
    x0 = a*rho

```

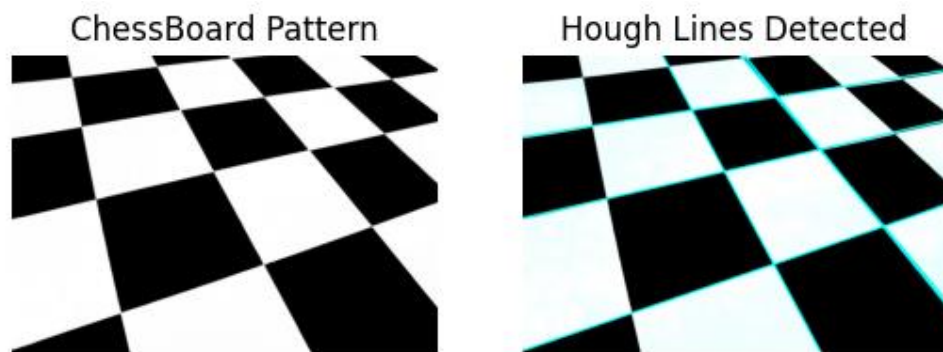
```

y0 = b*rho
x1 = int(x0 + 1000*(-b))
y1 = int(y0 + 1000*(a))
x2 = int(x0 - 1000*(-b))
y2 = int(y0 - 1000*(a))
cv2.line(I,(x1,y1),(x2,y2),(0,255,255),2)

plt.subplot(121),plt.imshow(Ig,cmap = 'gray'),plt.title("ChessBoard Pattern"),plt.axis("off")
plt.subplot(122),plt.imshow(I),plt.title("Hough Lines Detected"),plt.axis("off")
plt.show()

```

Output



Observation :

- The output image has detected chessboard pattern's lines using canny operator and hough line detector algorithm.

Conclusion:-

As the experiment performed,

- Lines, corners and edges were detected using various detectors and operators.
- Sobel, Prewitt and Canny operators were used to detect edges.
- Harris corner detector was used to detect corners in the target image.
- Hough line detection algorithm was used to detect straight lines in the image.

Libraries and functions used are matplotlib, numpy, OpenCV.