|  |  |
| --- | --- |
| **EXERCISE NO:7**  **DATE:06/03/2025** | **Hough Transform, Harris Corner Detection, and Object Segmentation** |

**Aim:**

To implement and analyze Hough Transform, Harris Corner Detection, and Object Segmentation techniques in image processing.

**1. Introduction**

Hough Transform is a feature extraction technique used in image processing to detect simple shapes such as lines and circles. Harris Corner Detection is used to detect corners within an image, and Object Segmentation is used to separate objects from the background.

**2. Algorithm:**

**Hough Transform:**

1. Convert the image to grayscale.
2. Apply edge detection (e.g., Canny edge detection) to find prominent edges.
3. Use the Hough Transform algorithm to detect lines or circles based on edge data.
4. Display the detected lines/circles on the original image.

**Harris Corner Detection:**

1. Convert the image to grayscale.
2. Compute the gradient of the image.
3. Calculate the Harris response matrix.
4. Threshold the response to identify corners.
5. Mark the detected corners on the image.

**Object Segmentation:**

1. Convert the image to grayscale.
2. Apply thresholding or edge detection to highlight object boundaries.
3. Use contour detection to extract object shapes.
4. Display the segmented objects.

**3. Code Implementation**

**Hough Transform Function:**

og = cv2.imread(r"Downloads/OIP.jpeg")

gray = cv2.cvtColor(og, cv2.COLOR\_BGR2GRAY)

blurred = cv2.GaussianBlur(gray, (5,5), 0)

edges = cv2.Canny(blurred, 50, 100)

lines = cv2.HoughLinesP(edges, 1, np.pi/180, threshold=90,minLineLength=30,maxLineGap=20)

output = og.copy()

if lines is not None:

for line in lines:

x1,y1,x2,y2 = line[0]

cv2.line(output, (x1, y1), (x2, y2), (0, 255, 0), 2)

plt.figure(figsize=(20,10))

plt.subplot(1,2,1)

plt.imshow(edges, cmap='gray')

plt.title("Edge Detection")

plt.axis("off")

plt.subplot(1,2,2)

plt.imshow(cv2.cvtColor(output, cv2.COLOR\_BGR2RGB))

plt.title("Filtered Line Detection")

plt.axis("off")

plt.tight\_layout()

plt.show()

**Harris Corner Detection Function:**

image = cv2.imread(r"Downloads/OIP.jpeg")

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

gray = cv2.GaussianBlur(gray, (5,5), 0)

dst = cv2.cornerHarris(gray, blockSize=2, ksize=3, k=0.05)

dst = cv2.dilate(dst, None)

threshold = 0.02 \* dst.max()

corner\_points = np.where(dst > threshold)

for y, x in zip(\*corner\_points):

cv2.circle(image, (x, y), 2, (255, 0, 255), -1)

plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)

plt.imshow(gray, cmap='gray')

plt.title("Grayscale Image")

plt.axis("off")

plt.subplot(1, 2, 2)

plt.imshow(cv2.cvtColor(image, cv2.COLOR\_BGR2RGB))

plt.title("Harris Corner Detection")

plt.axis("off")

plt.tight\_layout()

plt.show()

**Object Segmentation Function:**

image = cv2.imread(r"Downloads/CARDS.jpeg")

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

blurred = cv2.GaussianBlur(gray, (5,5), 0)

edges = cv2.Canny(blurred, 50, 150)

circles = cv2.HoughCircles(blurred, cv2.HOUGH\_GRADIENT, dp=1.2, minDist=50, param1=50, param2=40, minRadius=10, maxRadius=100)

output = image.copy()

if circles is not None:

circles = np.uint16(np.around(circles))

for i in circles[0, :]:

cv2.circle(output, (i[0], i[1]), i[2], (0, 255, 0), 3)

cv2.circle(output, (i[0], i[1]), 2, (0, 0, 255), 3)

corners = cv2.goodFeaturesToTrack(gray, maxCorners=7, qualityLevel=0.01, minDistance=10)

corners = corners.astype(int)

for i in corners:

x, y = i.ravel()

cv2.circle(output, (x, y), 5, (255, 0, 255), -1)

plt.figure(figsize=(12,6))

plt.subplot(1,2,1)

plt.imshow(edges, cmap='gray')

plt.title("Edge Detection")

plt.axis("off")

plt.subplot(1,2,2)

plt.imshow(cv2.cvtColor(output, cv2.COLOR\_BGR2RGB))

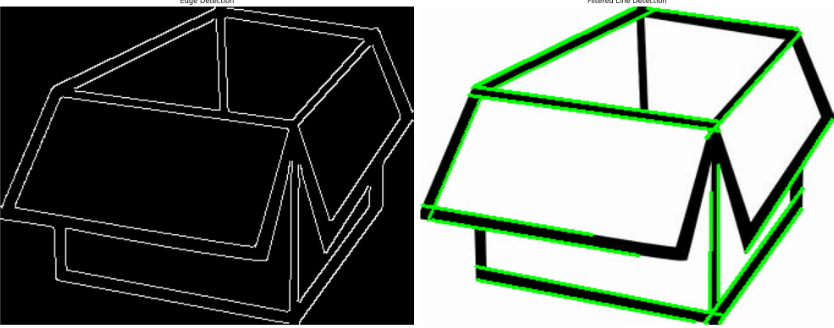
plt.title("Fixed Object Segmentation")

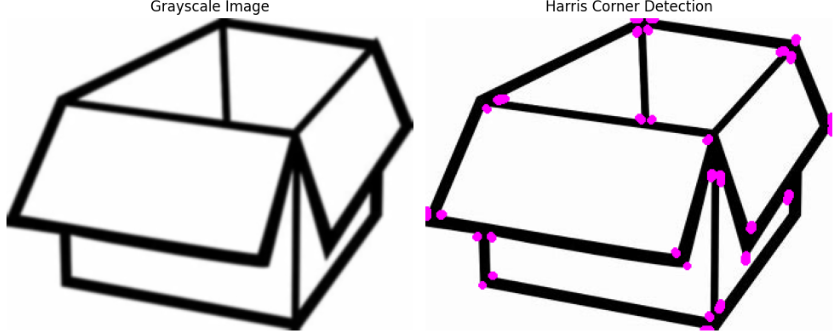
plt.axis("off")

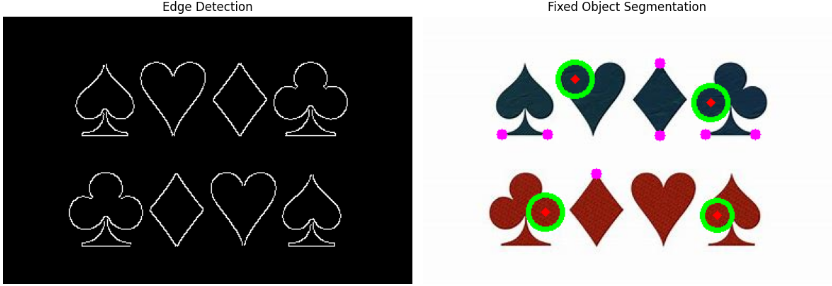
plt.tight\_layout()

plt.show()

**OUTPUT:**

****

****

****

**4. Function**

* **Hough Transform:** Detected lines are drawn in green on the original image.
* **Harris Corner Detection:** Corners are marked in red.
* **Object Segmentation:** Object boundaries are outlined in green.

**5. Results:**

The implemented techniques demonstrate fundamental image processing operations. Hough Transform effectively detects lines, Harris Corner Detection identifies key points, and Object Segmentation separates objects for further analysis. These methods are widely used in computer vision applications.