**Practical No: 10**

Page | 29

**Aim:** Write the program to apply segmentation for detecting lines, circles, and other shapes/objects. Also, implement edge-based and region-based segmentation.

**A:** Hough transform – detecting lines and circles

The Hough transform (HT) can be used to detect lines circles or • The Hough transform (HT) can be used to detect lines, circles or other parametric curves.

**Program code:**

image = rgb2gray(imread('../images/triangle\_circle.png'))

fig, axes = plt.subplots(2, 2,figsize=(20, 20))

ax = axes.ravel()

ax[0].imshow(image, cmap=cm.gray)

ax[0].set\_title('Input image', size=20)

ax[0].set\_axis\_off()

ax[1].imshow(np.log(1 + h),

extent=[10\*np.rad2deg(theta[-1]), np.rad2deg(theta[0]), d[-1], d[0]],

cmap=cm.hot, aspect=1/1.5)

ax[1].set\_title('Hough transform', size=20)

ax[1].set\_xlabel('Angles (degrees)', size=20)

ax[1].set\_ylabel('Distance (pixels)', size=20)

ax[1].axis('image')

ax[2].imshow(image, cmap=cm.gray)

for \_, angle, dist in zip(\*hough\_line\_peaks(h, theta, d)):

y0 = (dist - 0 \* np.cos(angle)) / np.sin(angle)

y1 = (dist - image.shape[1] \* np.cos(angle)) / np.sin(angle)

ax[2].plot((0, image.shape[1]), (y0, y1), '-r')

ax[2].set\_xlim((0, image.shape[1]))

ax[2].set\_ylim((image.shape[0], 0))

ax[2].set\_axis\_off()

ax[2].set\_title('Detected lines', size=20)

hough\_radii = np.arange(50, 100, 2)

hough\_res = hough\_circle(image, hough\_radii)

accums, cx, cy, radii = hough\_circle\_peaks(hough\_res, hough\_radii, total\_num\_peaks=6)

image = gray2rgb(image)

for center\_y, center\_x, radius in zip(cy, cx, radii):

circy, circx = circle\_perimeter(center\_y, center\_x, radius)

image[circy, circx] = (0.9, 0.2, 0.2)

ax[3].imshow(image, cmap=plt.cm.gray)

ax[3].set\_axis\_off()

ax[3].set\_title('Detected Circles', size=20)

plt.tight\_layout()

Page | 30

plt.show()

image = rgb2gray(imread('../images/coins.png'))

fig, axes = plt.subplots(1, 2,figsize=(20, 10),sharex=True, sharey=True)

ax = axes.ravel()

ax[0].imshow(image, cmap=plt.cm.gray)

ax[0].set\_axis\_off()

ax[0].set\_title('Original Image', size=20)

hough\_radii = np.arange(65, 75, 1)

hough\_res = hough\_circle(image, hough\_radii)

accums, cx, cy, radii = hough\_circle\_peaks(hough\_res, hough\_radii, total\_num\_peaks=4)

image = gray2rgb(image)

for center\_y, center\_x, radius in zip(cy, cx, radii):

circy, circx = circle\_perimeter(center\_y, center\_x, radius)

image[circy, circx] = (1, 0, 0)

ax[1].imshow(image, cmap=plt.cm.gray)

ax[1].set\_axis\_off()

ax[1].set\_title('Detected Circles', size=20)

plt.tight\_layout()

plt.show()

**Output:**



**B:** edge-based and region-based segmentation

Page | 31

**Program Code:**

coins **=** data**.**coins()

hist **=** np**.**histogram(coins, bins**=**np**.**arange(0, 256), normed**=True**)

fig, axes **=** plt**.**subplots(1, 2,figsize**=**(20, 10))

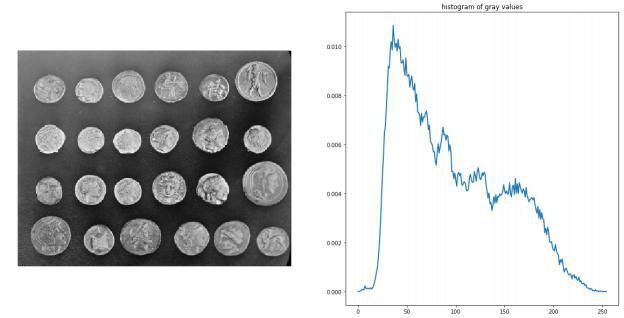
axes[0]**.**imshow(coins, cmap**=**plt**.**cm**.**gray, interpolation**=**'nearest')

axes[0]**.**axis('off')

axes[1]**.**plot(hist[1][:**-**1], hist[0], lw**=**2)

axes[1]**.**set\_title('histogram of gray values')

plt**.**show()



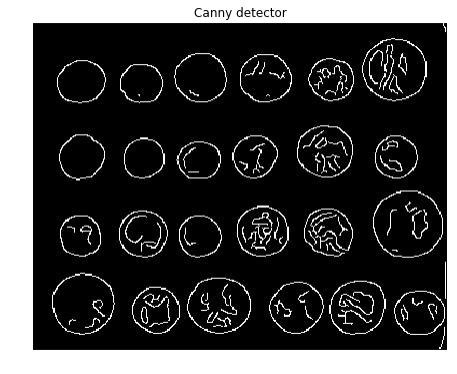
edges **=** canny(coins, sigma**=**2)

fig, ax **=** plt**.**subplots(figsize**=**( 10, 6))

ax**.**imshow(edges, cmap**=**plt**.**cm**.**gray, interpolation**=**'nearest')

ax**.**set\_title('Canny detector')

ax**.**axis('off')

plt**.**show()

Page | 32

**from** scipy **import** ndimage **as** ndi

fill\_coins **=** ndi**.**binary\_fill\_holes(edges)

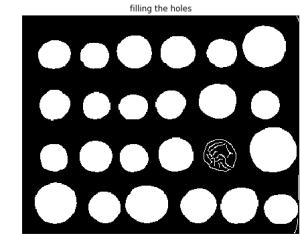
fig, ax **=** plt**.**subplots(figsize**=**( 10, 6))

ax**.**imshow(fill\_coins, cmap**=**plt**.**cm**.**gray, interpolation**=**'nearest')

ax**.**set\_title('filling the holes')

ax**.**axis('off')

plt**.**show()



ax**.**set\_title('segmentation')

Page | 33

ax**.**axis('off')

plt**.**sho