EN2550 Exercise 6

Name - Ekanayake E.M.S.S.N.

Index no - 190164M

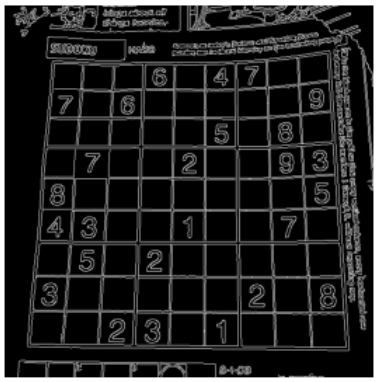
01

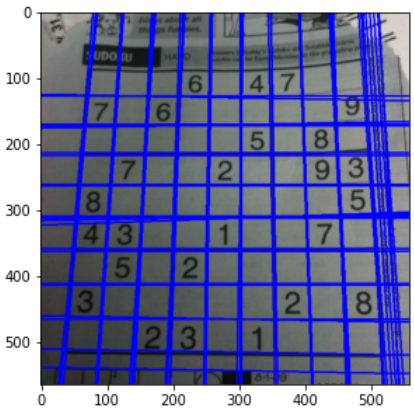
In [ ]:
```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt
```

In [ ]:
```
im = cv.imread(r'Images/sudoku.png', cv.IMREAD_COLOR)
assert im is not None

gray = cv.cvtColor(im, cv.COLOR_BGR2GRAY)
edges = cv.Canny(gray, 20, 120, apertureSize=3)
lines = cv.HoughLines(edges, 1, np.pi/180, 175)
for line in lines:
    rho, theta = line[0]
    a = np.cos(theta)
    b = np.sin(theta)
    x0, y0 = a*rho, b*rho
    x1, y1 = int(x0 + 1000*(-b)), int(y0 + 1000*(a))
    x2, y2 = int(x0 - 1000*(-b)), int(y0 - 1000*(a))
    cv.line(im, (x1,y1), (x2,y2), (0,0,255), 2)

fig, ax = plt.subplots(1,3,figsize = (16,8))
ax[0].imshow(gray, cmap = 'gray')
ax[1].imshow(edges, cmap = 'gray')
ax[2].imshow(im, cmap = 'gray')
ax[0].axis('off')
ax[1].axis('off')
plt.show()
```



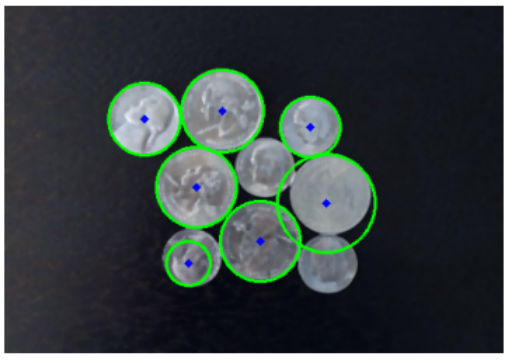


02

In [ ]:
```
img = cv.imread(r'Images/coins.jpg', cv.IMREAD_COLOR)

img = cv.medianBlur(img,5)
gray = cv.cvtColor(img,cv.COLOR_BGR2GRAY)
circles = cv.HoughCircles(gray,cv.HOUGH_GRADIENT,1,75,
                          param1=150,param2=20,minRadius=20,maxRadius=50)
circles = np.uint16(np.around(circles))
for i in circles[0,:]:
    # draw the outer circle
    cv.circle(img,(i[0],i[1]),i[2],(0,255,0),2)
    # draw the center of the circle
    cv.circle(img,(i[0],i[1]),2,(0,0,255),3)

fig, ax = plt.subplots(1,2, figsize = (14,8))
ax[0].imshow(gray , cmap='gray')
ax[1].imshow(img, cmap='gray')
ax[0].axis('off')
ax[1].axis('off')
plt.show()
```



03

In [ ]:
```
img = cv.imread(r"Images/pic1.png", cv.IMREAD_REDUCED_GRAYSCALE_2)
assert img is not None
temp_img = cv.imread(r"Images/templ.png", cv.IMREAD_REDUCED_GRAYSCALE_2)
assert img is not None

im_edges = cv.Canny(img, 50, 250)
temp_edges = cv.Canny(temp_img, 50, 250)
alg = cv.createGeneralizedHoughGuil()
alg.setTemplate(temp_edges)
alg.setAngleThresh(100000)
alg.setScaleThresh(40000)
alg.setPosThresh(1000)
alg.setAngleStep(1)
alg.setScaleStep(0.1)
alg.setMinScale(0.9)
alg.setMaxScale(1.1)
positions, votes = alg.detect(im_edges)

out = cv.cvtColor(img, cv.COLOR_BAYER_BG2BGR)
for x, y, scale, orientation in positions[0]:
    halfHeight = temp_img.shape[0] / 2.* scale
    halfWidth = temp_img.shape[1] / 2.* scale
    p1 = (int(x - halfWidth), int(y - halfHeight))
    p2 = (int(x + halfWidth), int(y + halfHeight))
    print("x = {}, y = {}, scale = {}, orientation = {}, p1 = {}, p2 = {}".format(x, y, scale, orientation, p1, p2))
    cv.rectangle(out, p1, p2, (0, 0, 255))

fig, ax = plt.subplots(1,3, figsize=(20,10))
ax[0].imshow(img , cmap='gray')
ax[1].imshow(temp_img , cmap='gray')
ax[2].imshow(out, cmap='gray')
for i in range(3):
    ax[i].axis('off')
plt.show()
```

```
x = 29.0, y = 109.0, scale = 1.0, orientation = 0.0, p1 = (4, 76), p2 = (54, 141)
```

04

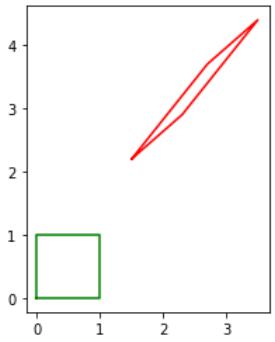
In [ ]:
```
a, b, c, d = [0, 0, 1], [0, 1, 1], [1,1,1], [1,0,1]
X = np.array([a,b,c,d]).T

theta = np.pi*30/180
s = 1
tx, ty = 1.5, 2.2
# H = np.array([[s*np.cos(theta), -s*np.sin(theta), tx], [s*np.sin(theta), s*np.cos(theta), ty], [0,0,1]])
# Y = H @ X

a11, a12, a21, a22 = 0.8, 1.2, 0.7, 1.5 #Should be a non-singular matrix here
A = np.array([[a11,a12,tx], [a21, a22, ty], [0,0,1]])
Y = A @ X

x = np.append(X[0, :], X[0, 0])
y = np.append(X[1, :], X[1, 0])
fig, ax = plt.subplots(1,1)
ax.plot(x, y, color='g')
ax.set_aspect('equal')

x = np.append(Y[0, :], Y[0, 0])
y = np.append(Y[1, :], Y[1, 0])
ax.plot(x, y, color='r')
ax.set_aspect('equal')
plt.show()
```



05

In [ ]:
```
im1 = cv.imread(r"Images/graf/img1.ppm", cv.IMREAD_ANYCOLOR)
im5 = cv.imread(r"Images/graf/img5.ppm", cv.IMREAD_ANYCOLOR)

im1 = cv.cvtColor(im1, cv.COLOR_BGR2RGB)
im5 = cv.cvtColor(im5, cv.COLOR_BGR2RGB)

H = np.array([[6.6378505e-01,6.8003334e-01,-3.1230335e+01],[-1.4495500e-01,9.7128304e-01,1.4877420e+02],[4.2518504e-04,-1.3930359e-05,1.00000
with open(r'Images/graf/H1to5p') as f:
    H = np.array([[float(h) for h in line.split()] for line in f])

im1to5 = cv.warpPerspective(im4, np.linalg.inv(H), (2000,2000))

fig, ax = plt.subplots(1,3, figsize=(20,10))
ax[0].imshow(im1)
ax[1].imshow(im5)
ax[2].imshow(im1to5)
for i in range(3):
    ax[i].axis('off')
plt.show()
```