Name: C. J. Kurukulasuriya

Index number: 190337X

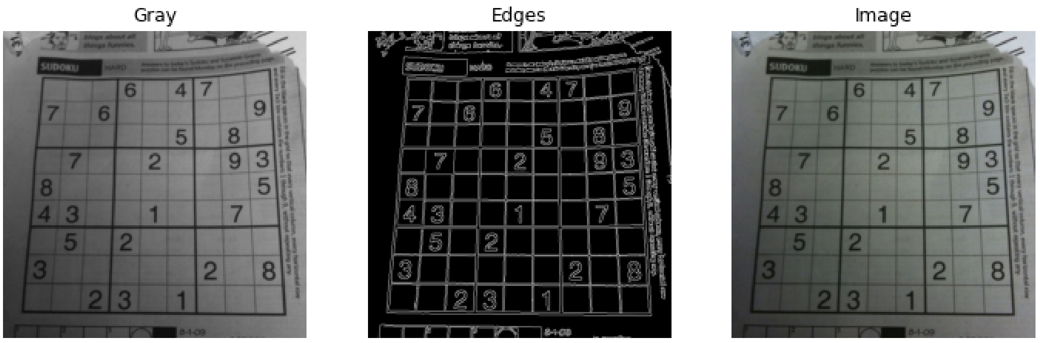
In [ ]:

```
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

Q1

In [ ]:

```
im = cv.imread('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)

# cv.imshow("Image", gray)
# cv.waitKey()
# cv.imshow("Image", edges)
# cv.waitKey()
# cv.destroyAllWindows()

fig, ax = plt.subplots(1,3,figsize=(12,5))
gray  = cv.cvtColor(gray, cv.COLOR_BGR2RGB)
edges = cv.cvtColor(edges, cv.COLOR_BGR2RGB)
im    = cv.cvtColor(im, cv.COLOR_BGR2RGB)

ax[0].imshow(gray)
ax[1].imshow(edges)
ax[2].imshow(im)

ax[0].set_title("Gray")
ax[1].set_title("Edges")
ax[2].set_title("Image")

for i in range(3):
    ax[i].axis('off')

plt.show()
```

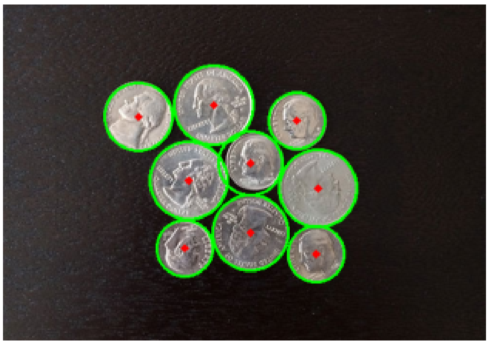


Q2

In [ ]:

```
im = cv.imread('Images/coins.jpg', cv.IMREAD_COLOR)
assert im is not None

gray = cv.cvtColor(im, cv.COLOR_BGR2GRAY)

circles = cv.HoughCircles(gray, cv.HOUGH_GRADIENT, 1, 65, param1 = 155, param2 = 20, minRadius=17, maxRadius=50)
circles = np.uint16(np.around(circles))
```

```
for i in circles[0, :]:
    cv.circle(im, (i[0],i[1]), i[2], (0,255,0), 2) # outer circle
    cv.circle(im, (i[0],i[1]), 2, (0,0,255), 3) # center of circle

# cv.imshow("Detected Circles", im)
# cv.waitKey(0)
# cv.destroyAllWindows()

fig, ax = plt.subplots()
im = cv.cvtColor(im, cv.COLOR_BGR2RGB)
ax.axis('off')
ax.imshow(im)

plt.show()
```



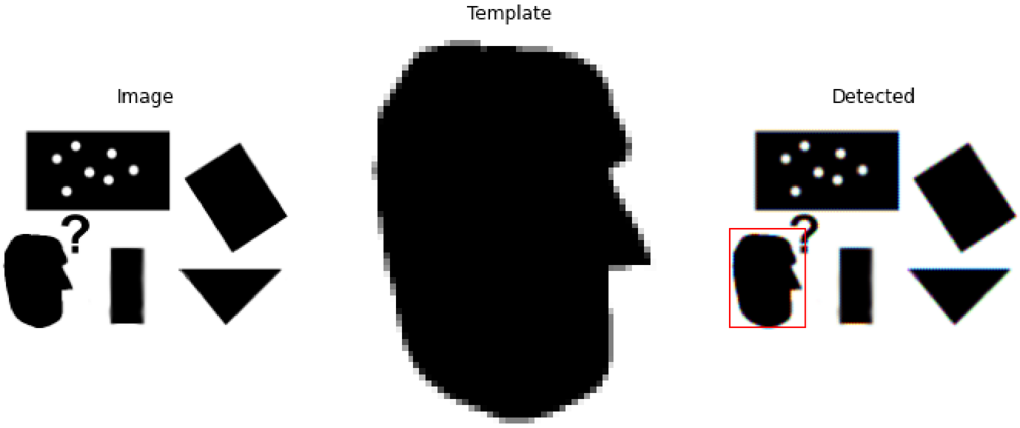
Q3

In [ ]:

```
im = cv.imread('Images/pic1.png', cv.IMREAD_REDUCED_GRAYSCALE_2)
temp1 = cv.imread('Images/templ.png', cv.IMREAD_REDUCED_GRAYSCALE_2)
assert im is not None
assert temp1 is not None

im_edges = cv.Canny(im, 50, 250)
temp1_edges = cv.Canny(temp1, 50, 250)

alg = cv.createGeneralizedHoughGuil()

alg.setTemplate(temp1_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(im,cv.COLOR_BAYER_BG2RGB)
for x,y,scale,orientation in positions[0]:
    halfHeight = temp1.shape[0]/ 2. * scale
    halfWidth = temp1.shape[1]/ 2. * scale
    p1 = (int(x - halfWidth), int(y - halfHeight))
    p2 = (int(x + halfWidth), int(y + halfHeight))
    print("x  = {}\ny= {}\nscale = {}\norientation = {}\np1 = {}\np2 = {}".format(x,y,scale,orientation,p1,p2))
    cv.rectangle(out,p1,p2,(0,0,255))

# cv.namedWindow("Image",cv.WINDOW_NORMAL)
# cv.imshow("Image",im)
# cv.waitKey(0)
# cv.imshow("Image",out)
# cv.waitKey(0)
# cv.destroyAllWindows()

fig,ax = plt.subplots(1,3,figsize=(12,5))

im = cv.cvtColor(im,cv.COLOR_BGR2RGB)
temp1 = cv.cvtColor(temp1,cv.COLOR_BGR2RGB)
out = cv.cvtColor(out,cv.COLOR_BGR2RGB)

ax[0].imshow(im)
ax[1].imshow(temp1)
ax[2].imshow(out)

ax[0].set_title("Image")
ax[1].set_title("Template")
ax[2].set_title("Detected")
```

```
for i in range(3):
    ax[i].axis('off')

plt.show()
```

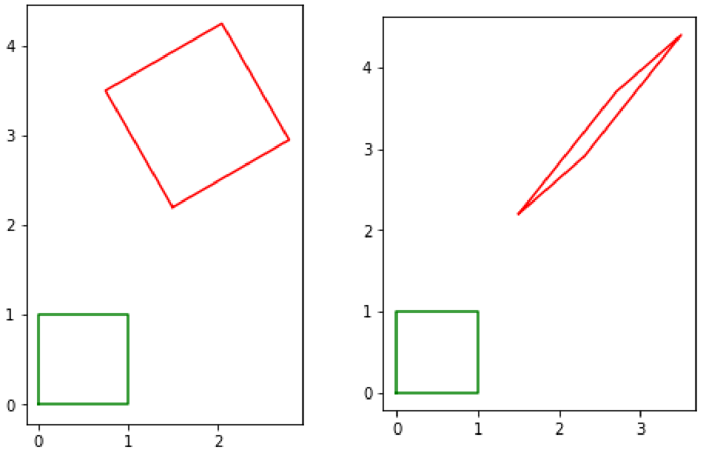
```
x  = 29.0
y= 109.0
scale = 1.0
orientation = 0.0
p1 = (4, 76)
p2 = (54, 141)
```



Q4

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 = 1.5
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]])
Y1 = H @ X

a11, a12, a21, a22 = 0.8, 1.2, 0.7, 1.5 #Should be a non-singular matrix
A = np.array([[a11,a12,tx], [a21, a22, ty], [0,0,1]])
Y2 = A @ X

fig, ax = plt.subplots(1,2, figsize=(8,5))

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

x = np.append(Y1[0, :], Y1[0, 0])
y = np.append(Y1[1, :], Y1[1, 0])
ax[0].plot(x, y, color='r')
ax[0].set_aspect('equal')

x = np.append(Y2[0,:],Y2[0,0])
y = np.append(Y2[1,:],Y2[1,0])
ax[1].plot(x,y,color='r')
ax[1].set_aspect('equal')
```



Q5

In [ ]:

```
im1 = cv.imread("Images/graf/img1.ppm", cv.IMREAD_ANYCOLOR)
im4 = cv.imread("images/graf/img4.ppm", cv.IMREAD_ANYCOLOR)

with open('Images/graf/H1to4p') as f:
    H = np.array([[float(h) for h in line.split()] for line in f])

im4to1 = cv.warpPerspective(im4, np.linalg.inv(H), (2000, 2000))

im1 = cv.cvtColor(im1,cv.COLOR_BGR2RGB)
im4 = cv.cvtColor(im4,cv.COLOR_BGR2RGB)
im4to1 = cv.cvtColor(im4to1,cv.COLOR_BGR2RGB)

fig, ax = plt.subplots(1,3,figsize=(16,5))

ax[0].set_title("im1")
ax[1].set_title("im4")
ax[2].set_title("im4 to im1")
ax[0].imshow(im1)
ax[1].imshow(im4)
ax[2].imshow(im4to1)

for i in range(3):
    ax[i].axis('off')

plt.show()
```