**Python code using OpenCV python library to find matching DRC1 pattern**

import cv2

import numpy as np

import math

# read chessboard image

img = cv2.imread('chessboard.png')

# read drc1 image template

template = cv2.imread('drc1.png', cv2.IMREAD\_UNCHANGED)

hh, ww = template.shape[:2]

# extract drc1 base image and alpha channel and make alpha 3 channels

drc1 = template[:,:,0:3]

alpha = template[:,:,3]

alpha = cv2.merge([alpha,alpha,alpha])

# set threshold

threshold = 0.89

# do masked template matching and save correlation image

corr\_img = cv2.matchTemplate(img, drc1, cv2.TM\_CCORR\_NORMED, mask=alpha)

# search for max score

result = img.copy()

max\_val = 1

rad = int(math.sqrt(hh\*hh+ww\*ww)/4)

while max\_val > threshold:

# find max value of correlation image

min\_val, max\_val, min\_loc, max\_loc = cv2.minMaxLoc(corr\_img)

print(max\_val, max\_loc)

if max\_val > threshold:

# draw match on copy of input

cv2.rectangle(result, max\_loc, (max\_loc[0]+ww, max\_loc[1]+hh), (0,0,255), 2)

# write black circle at max\_loc in corr\_img

cv2.circle(corr\_img, (max\_loc), radius=rad, color=0, thickness=cv2.FILLED)

else:

break

# save results

cv2.imwrite('chessboard\_drc1.png', drc1)

cv2.imwrite('chessboard\_alpha.png', alpha)

cv2.imwrite('chessboard\_correlation.png', (255\*corr\_img).clip(0,255).astype(np.uint8))

cv2.imwrite('chessboard\_matches2.jpg', result)

cv2.imshow('drc1',drc1)

cv2.imshow('alpha',alpha)

cv2.imshow('result',result)

cv2.waitKey(0)

cv2.destroyAllWindows()