import cv2

import numpy as np

# Load RGB fisheye image

img = cv2.imread("fisheye\_rgb.jpg")

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

# Threshold to separate circle from black background

\_, thresh = cv2.threshold(gray, 10, 255, cv2.THRESH\_BINARY)

# Find contours

contours, \_ = cv2.findContours(thresh, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_NONE)

contour = max(contours, key=cv2.contourArea)

# Fit ellipse

if len(contour) >= 5:

ellipse = cv2.fitEllipse(contour)

(cx, cy), (major\_axis, minor\_axis), angle = ellipse

# Generate 12 evenly spaced points along ellipse

angles = np.linspace(0, 2\*np.pi, 12, endpoint=False)

points = []

for a in angles:

x = int(cx + (major\_axis/2) \* np.cos(a) \* np.cos(np.deg2rad(angle))

- (minor\_axis/2) \* np.sin(a) \* np.sin(np.deg2rad(angle)))

y = int(cy + (major\_axis/2) \* np.cos(a) \* np.sin(np.deg2rad(angle))

+ (minor\_axis/2) \* np.sin(a) \* np.cos(np.deg2rad(angle)))

points.append((x, y))

points = np.array(points, dtype=np.int32)

# Draw results

output = img.copy()

for p in points:

cv2.circle(output, tuple(p), 4, (0,0,255), -1)

cv2.polylines(output, [points], isClosed=True, color=(0,0,255), thickness=2)

cv2.imwrite("fisheye\_rgb\_12pts.png", output)

print("12 Ellipse Boundary Points:\n", points)

else:

print("Not enough points to fit an ellipse.")