Detecting April Tags with Opencv and python

Apriltags is a visual fiducial system, useful for a wide variety of tasks including augmented reality, robotics, and camera calibration.The tags can be created by software and a printer.Detecting apriltags is the goal of this writeup.

1) Script:load the necessary packages,here i am using google collab,open cv already comes with collab but pip install apriltags

```
# import the necessary packages
import argparse
import cv2
```

2) Script:specifying image path,reading image from disk and converting to gray scale

```
# load the input image and convert it to grayscale
img_path = "./example_02.png"
print("[INFO] loading image...")
image = cv2.imread(img_path)
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

3) Script:To use the apriltag detector ,there is need to define the options which is what family of apriltags we wish to detect.there are 6 family types.Define the detector and detect using detector.detect function.

```
# define the AprilTags detector options and then detect the
AprilTags
# in the input image
print("[INFO] detecting AprilTags...")
options = apriltag.DetectorOptions(families="tag36h11")
detector = apriltag.Detector(options)
results = detector.detect(gray)
print("[INFO] {} total AprilTags detected".format(len(results)))
```

4) Script:The following loops through depending on the number of april tags detected

```
# loop over the AprilTag detection results
for r in results:
```

5) Script: we get the corner point of the tags and convert to integers

```
  # extract the bounding box (x, y)-coordinates for the AprilTag
  # and convert each of the (x, y)-coordinate pairs to integers
  (ptA, ptB, ptC, ptD) = r.corners
  ptB = (int(ptB[0]), int(ptB[1]))
  ptC = (int(ptC[0]), int(ptC[1]))
  ptD = (int(ptD[0]), int(ptD[1]))
```

```
  ptA = (int(ptA[0]), int(ptA[1]))
```
6)Script:then we draw the green lines seen around the result tags
```
        # draw the bounding box of the AprilTag detection
  cv2.line(image, ptA, ptB, (0, 255, 0), 2)
  cv2.line(image, ptB, ptC, (0, 255, 0), 2)
  cv2.line(image, ptC, ptD, (0, 255, 0), 2)
  cv2.line(image, ptD, ptA, (0, 255, 0), 2)
```
7) Script: also draw the red circle in the middle of the tags detected

```
  # draw the center (x, y)-coordinates of the AprilTag
  (cX, cY) = (int(r.center[0]), int(r.center[1]))
  cv2.circle(image, (cX, cY), 5, (0, 0, 255), -1)
```
8) Script:We then draw the tags family on the image with the green texts shown in the right corner of the result
```
  # draw the tag family on the image
  tagFamily = r.tag_family.decode("utf-8")
  cv2.putText(image, tagFamily, (ptA[0], ptA[1] - 15),
    cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
  print("[INFO] tag family: {}".format(tagFamily))
```
9) Script:Outside the loop,We display the result,because i am using collab we need to import cv2 patch for imshow

```
from google.colab.patches import cv2_imshow
# show the output image after AprilTag detection
cv2_imshow(image)
cv2.waitKey(0)
```

Original image

Result: