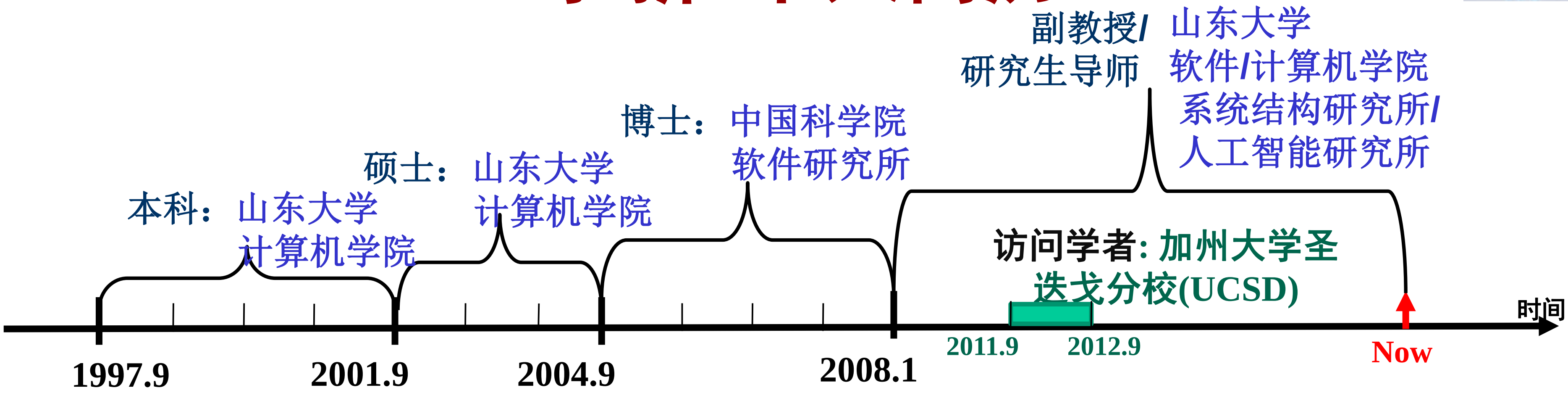
工业人工智能技术

基于OpenCV的形状检测

李新

山东大学 软件学院

李新 个人简历



本科：山东大学
计算机学院

硕士：山东大学
计算机学院

博士：中国科学院
软件研究所

副教授/
研究生导师

山东大学
软件/计算机学院
系统结构研究所/
人工智能研究所

访问学者: 加州大学圣迭戈分校(UCSD)

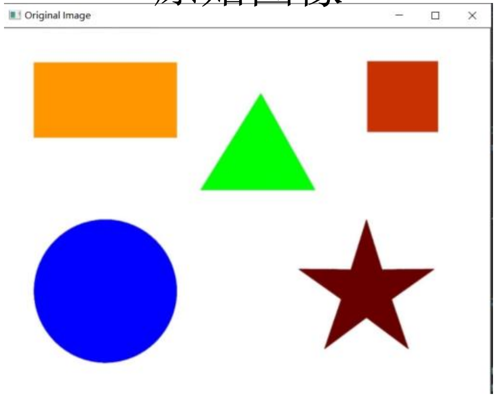
- ➢ 研究方向
  - 目标检测与跟踪
  - 无人机智能巡检
  - 大数据处理

邮箱：lx@sdu.edu.cn

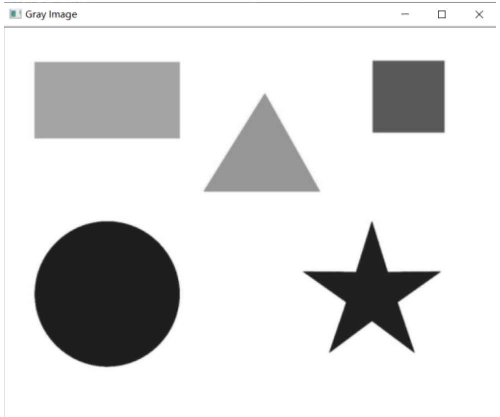
电话：138-531-23559

形状检测

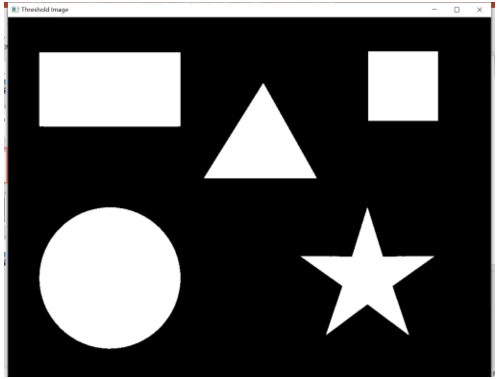
原始图像



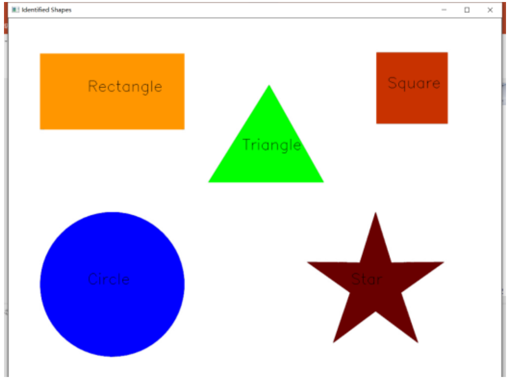
灰度图像



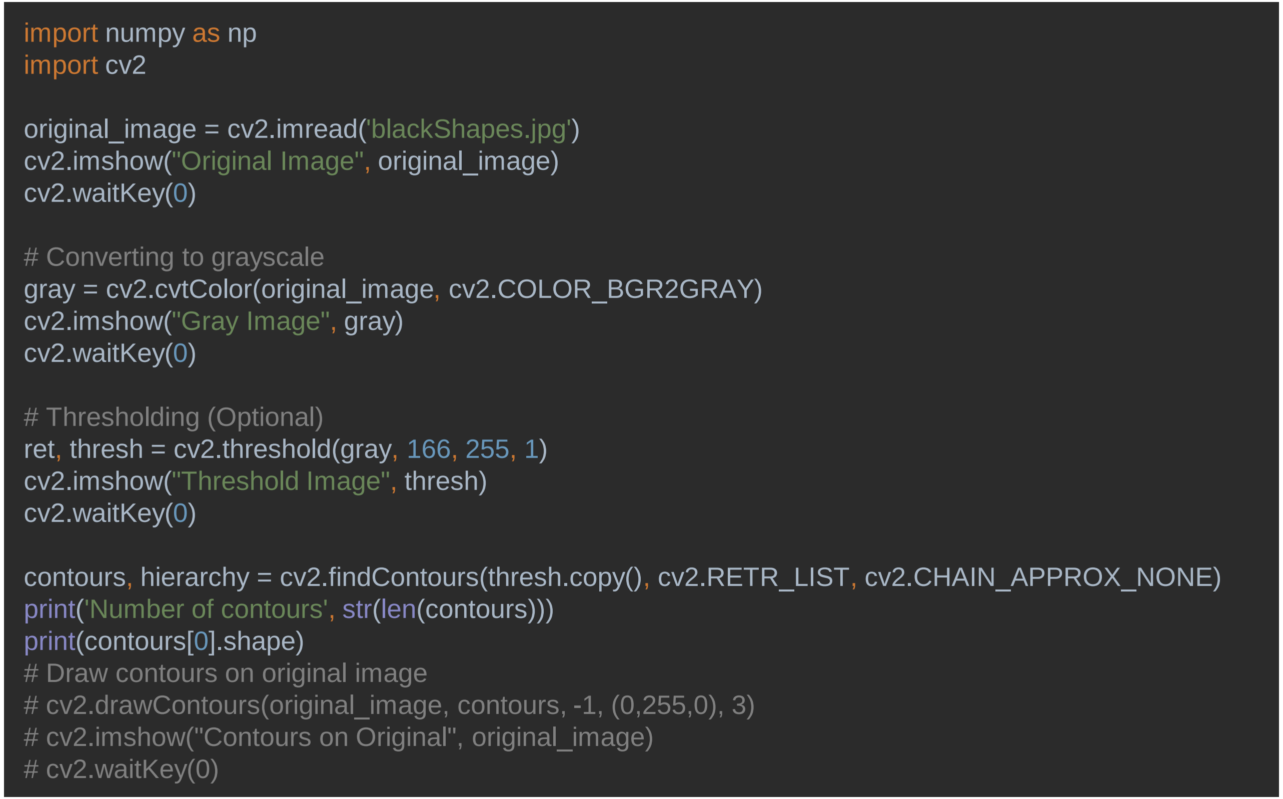
二值图像



提取边框、顶点数，进行形状判断

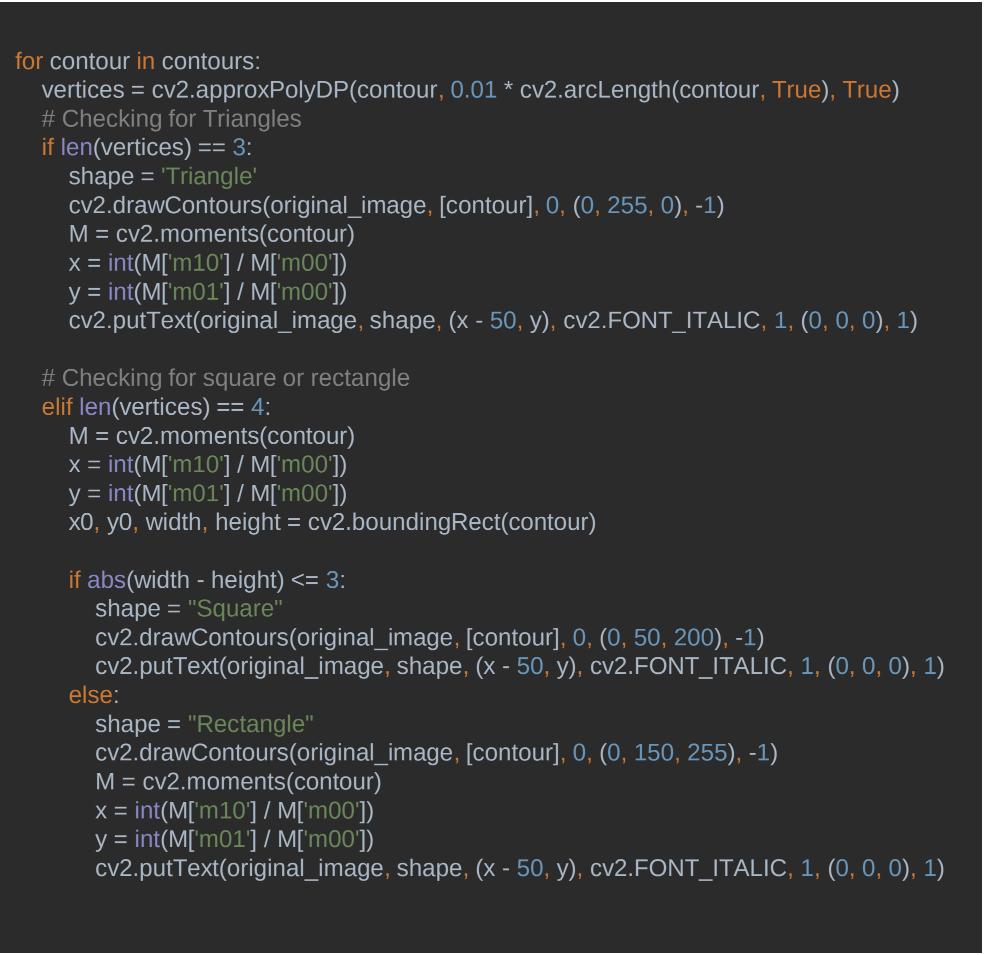


实验代码

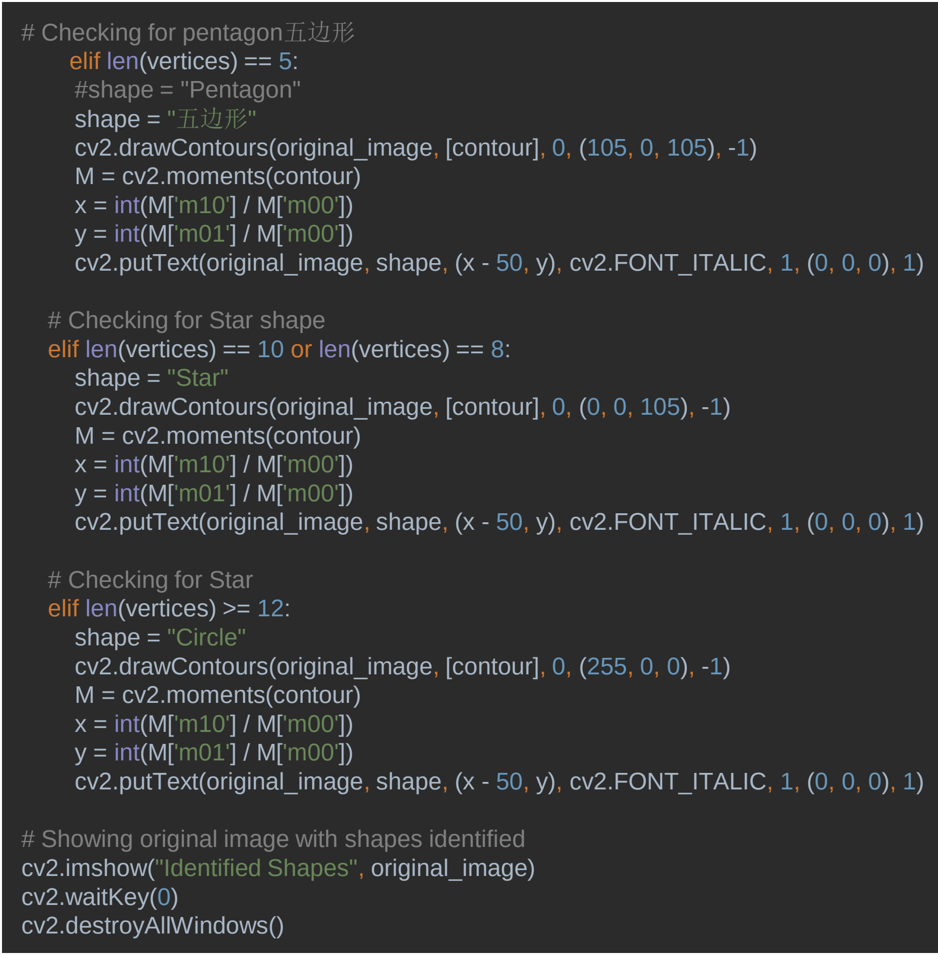


```
import numpy as np
import cv2

original_image = cv2.imread('blackShapes.jpg')
cv2.imshow("Original Image", original_image)
cv2.waitKey(0)

# Converting to grayscale
gray = cv2.cvtColor(original_image, cv2.COLOR_BGR2GRAY)
cv2.imshow("Gray Image", gray)
cv2.waitKey(0)

# Thresholding (Optional)
ret, thresh = cv2.threshold(gray, 166, 255, 1)
cv2.imshow("Threshold Image", thresh)
cv2.waitKey(0)

contours, hierarchy = cv2.findContours(thresh.copy(), cv2.RETR_LIST, cv2.CHAIN_APPROX_NONE)
print('Number of contours', str(len(contours)))
print(contours[0].shape)
# Draw contours on original image
# cv2.drawContours(original_image, contours, -1, (0,255,0), 3)
# cv2.imshow("Contours on Original", original_image)
# cv2.waitKey(0)
```

实验代码



```
for contour in contours:
    vertices = cv2.approxPolyDP(contour, 0.01 * cv2.arcLength(contour, True), True)
    # Checking for Triangles
    if len(vertices) == 3:
        shape = 'Triangle'
        cv2.drawContours(original_image, [contour], 0, (0, 255, 0), -1)
        M = cv2.moments(contour)
        x = int(M['m10'] / M['m00'])
        y = int(M['m01'] / M['m00'])
        cv2.putText(original_image, shape, (x - 50, y), cv2.FONT_ITALIC, 1, (0, 0, 0), 1)

    # Checking for square or rectangle
    elif len(vertices) == 4:
        M = cv2.moments(contour)
        x = int(M['m10'] / M['m00'])
        y = int(M['m01'] / M['m00'])
        x0, y0, width, height = cv2.boundingRect(contour)

        if abs(width - height) <= 3:
            shape = "Square"
            cv2.drawContours(original_image, [contour], 0, (0, 50, 200), -1)
            cv2.putText(original_image, shape, (x - 50, y), cv2.FONT_ITALIC, 1, (0, 0, 0), 1)
        else:
            shape = "Rectangle"
            cv2.drawContours(original_image, [contour], 0, (0, 150, 255), -1)
            M = cv2.moments(contour)
            x = int(M['m10'] / M['m00'])
            y = int(M['m01'] / M['m00'])
            cv2.putText(original_image, shape, (x - 50, y), cv2.FONT_ITALIC, 1, (0, 0, 0), 1)
```



```
# Checking for pentagon五边形
       elif len(vertices) == 5:
        #shape = "Pentagon"
        shape = "五边形"
        cv2.drawContours(original_image, [contour], 0, (105, 0, 105), -1)
        M = cv2.moments(contour)
        x = int(M['m10'] / M['m00'])
        y = int(M['m01'] / M['m00'])
        cv2.putText(original_image, shape, (x - 50, y), cv2.FONT_ITALIC, 1, (0, 0, 0), 1)

    # Checking for Star shape
    elif len(vertices) == 10 or len(vertices) == 8:
        shape = "Star"
        cv2.drawContours(original_image, [contour], 0, (0, 0, 105), -1)
        M = cv2.moments(contour)
        x = int(M['m10'] / M['m00'])
        y = int(M['m01'] / M['m00'])
        cv2.putText(original_image, shape, (x - 50, y), cv2.FONT_ITALIC, 1, (0, 0, 0), 1)

    # Checking for Star
    elif len(vertices) >= 12:
        shape = "Circle"
        cv2.drawContours(original_image, [contour], 0, (255, 0, 0), -1)
        M = cv2.moments(contour)
        x = int(M['m10'] / M['m00'])
        y = int(M['m01'] / M['m00'])
        cv2.putText(original_image, shape, (x - 50, y), cv2.FONT_ITALIC, 1, (0, 0, 0), 1)

# Showing original image with shapes identified
cv2.imshow("Identified Shapes", original_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

谢 谢！