how to count non-zero pixels inside a contour opencv

Asked 3 years, 1 month ago Active 1 year ago Viewed 2k times

1

I am developing `OMR` scanner android application using `opencv` library. I have detected my circles inside the sheet as contours and now I want to get filled circle contours from all the obtains contours Since java support for opencv is very less I couldnt figure out anything, please suggest some method for the same.

```
    //paramview is my image
    Utils.bitmapToMat(paramView, localMat1);
    Mat localMat2 = new Mat();
    double[] lo;
    Imgproc.GaussianBlur(localMat1, localMat2, new Size(5.0D, 5.0D), 7.0D,
6.5D);
    Object localObject = new Mat();
    Imgproc.cvtColor(localMat2, (Mat)localObject, COLOR_RGB2GRAY);
    Mat cloneMat= ((Mat) localObject).clone();
    localMat2 = localMat1.clone();
    bitwise_not(cloneMat,cloneMat);
    Imgproc.threshold(cloneMat,localMat2,127,255,Imgproc.THRESH_OTSU);
    Mat thresh=localMat2.clone();

    List<MatOfPoint> contours = new ArrayList<MatOfPoint>();
    List<MatOfPoint> questions = new ArrayList<MatOfPoint>();
    List<MatOfPoint> sorted = new ArrayList<MatOfPoint>();

    //All contours detected
    Mat hierarchy = new Mat();
    Imgproc.findContours(localMat2, contours, hierarchy,
    Imgproc.RETR_EXTERNAL, Imgproc.CHAIN_APPROX_SIMPLE);
```

[Image of Detected circles here]

java android opencv

edited Jul 15 '17 at 5:16

asked Jul 12 '17 at 6:23

Abhishek

41 9

Can you add your code to find the circles' contours. Also, add your current image output and the desired result. Finally, please have a look at the Help Center. – Elouarn Laine Jul 12 '17 at 10:01

2 Answers

Active | Oldest | Votes

1

I reworked my own code and found this solution. Hope it might help.

```
for (int contourIdx = 0; contourIdx < questionSortedR.size(); contourIdx++) {
        //creating rectangle around identified contour
        Rect rectCrop = boundingRect(questionSortedR.get(contourIdx));
        //creating crop of that contour from actual image
        Mat imageROI= thresh.submat(rectCrop);
        //apply countnonzero method to that crop
        int total = countNonZero(imageROI);
```

```
            double pixel =total/contourArea(questionSortedR.get(contourIdx))*100;
            //pixel is in percentage of area that is filled
            if(pixel>=100 && pixel<=130){
                //counting filled circles
                count++;
            }

        }
```

answered Jul 31 '17 at 9:30

Abhishek
41 9

0

I propose an alternative to the accepted answer: instead of counting pixels inside a bounding rectangle, paint the contour into a mask, then mask the original image and count the pixels inside it. I was counting black pixels on a white background, where the contour kept several pixels on the edge, so your mileage may vary. Here is my code in Python:

```
mask = np.zeros(bw_image.shape, np.uint8)
cv.drawContours(mask, [contour], 0, 255, -1)
inverted = cv.bitwise_not(bw_image)
masked = cv.bitwise_not(cv.bitwise_and(inverted, inverted, mask = mask))

# Grab masked image inside contour
x, y, w, h = cv.boundingRect(contour)
pixels = masked[y:y+h, x:x+w]

# Check if black is only a line, in which case whiteness is 1
kernel = np.ones((3, 3), np.uint8)
dilated = cv.dilate(pixels, kernel, iterations = 1)
whiteness = np.sum(dilated) / (255 * w * h)
```

answered Aug 2 '19 at 8:17

miguelmorin
2,641 1 13 33