How can I extract internal contours (holes) with python opencv?

Asked 4 years, 3 months ago Active 11 months ago Viewed 5k times

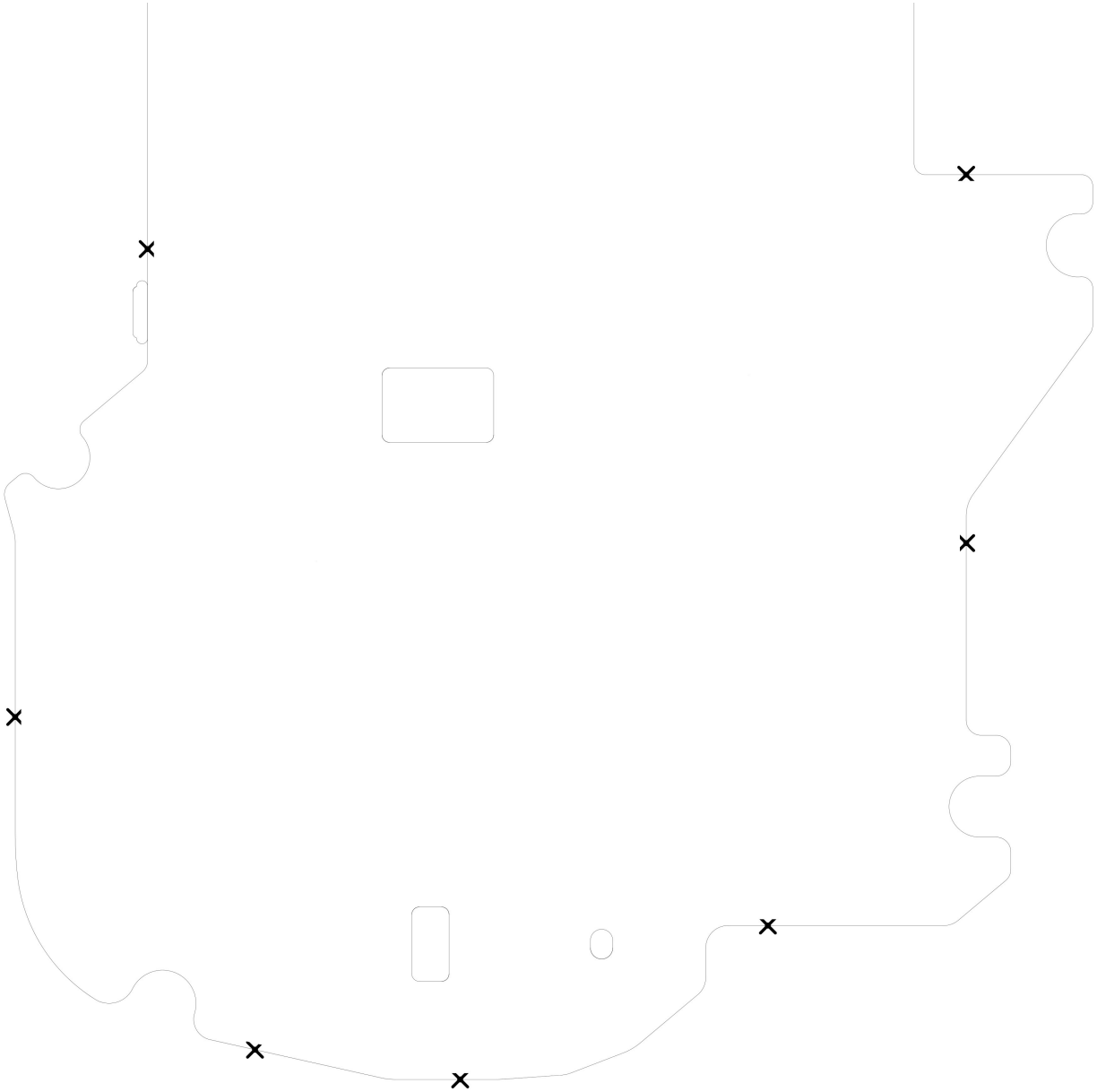
2

3

is there an easy and direct way to extract the internal contours (holes) from an image using opencv 3.1 python ?

I know that I can use "area" as a condition. However, if I change the image resolution, the "areas" are not the same.

For instance, with this image:



How can I extract the internal holes?

```
_, contours, hier_ =
cv2.findContours(img,cv2.RETR_CCOMP,cv2.CHAIN_APPROX_SIMPLE)
areas = [cv2.contourArea(c) for c in millCnts]
max_area = np.max(areas)
Mask = np.ones(img.shape[:2], dtype="uint8") * 255

   # I can do something like this (currently not working, just to show an
example)
for c in contours:
    if(( cv2.contourArea(c) > 8) and (cv2.contourArea(c)< 100000)):
        cv2.drawContours(Mask ,[c],-1,0,1)
```

python opencv

asked May 11 '16 at 10:37

marco

677 2 13 29

You can always use the hierarchy values to know if they are nested (holes) or not (outer one)... not sure if already find contours gets all of the contours correctly (you should show that output) but if it

does, you can check if the countour ´c´ is nested by checking the corresponding hierarchy ´h´, the 4th value (`h[3]`) if it is non negative. – api55 May 11 '16 at 11:14

I think my main problem is understanding how hierarchy works. The ouput array is huge :\ At this particular case it won't work . The limits should be between 8000 and 9000 probably ! – marco May 11 '16 at 11:19

where is `millCnts` defined? can you show an image of all the contours found? – api55 May 11 '16 at 13:29

1 Answer

Active | Oldest | Votes

5

As I explained in my comment, you have to check the hierarchy return variable. After find contours you will get the contours (List of List of Points) and hierarchy (List of List).

The documentation is very clear in this:

> hierarchy – Optional output vector, containing information about the image topology. It has as many elements as the number of contours. For each i-th contour contours[i] , the elements `hierarchy[i][0]` , `hiearchy[i][1]` , `hiearchy[i][2]` , and `hiearchy[i][3]` are set to 0-based indices in contours of the next and previous contours at the same hierarchical level, the first child contour and the parent contour, respectively. If for the contour i there are no next, previous, parent, or nested contours, the corresponding elements of hierarchy[i] will be negative.

So, this means that for each `countour[i]` you should get a `hierarchy[i]` that contains a List with 4 variables:

- `hierarchy[i][0]` : the index of the next contour of the same level
- `hierarchy[i][1]` : the index of the previous contour of the same level
- `hierarchy[i][2]` : the index of the first child
- `hierarchy[i][3]` : the index of the parent

So, saying that, in your case, there should be one without a parent, and you can check which one by checking the `hierarchy[i][3]` if it is negative.

It should be something like (untested code):

```
holes = [contours[i] for i in range(len(contours)) if hierarchy[i][3] >= 0]
```

*** UPDATE:***

To summarize what we discussed in the chat,

- The image was too big, and the contours had small holes: solved with dilate and erode with a kernel of size 75
- The image needed to be inverted since OpenCV expects for dilate a black background
- The algorithm was giving 2 big contours, one outside (as expected) and one inside (almost the same as the outside one), this is probably due to the image having some external (and closed) bumps. This was solved by removing the contour without a parent and its first child.

edited May 12 '16 at 15:33

answered May 11 '16 at 11:53

api55

9,245 4 34 50

I think that I'm understading how it works, however It is not clear how can it be done comparasions. I know that you didn't test code, but the output is : ValueError: The truth value of an array with more than one element is ambiguous. Use a.any() or a.all() – marco May 11 '16 at 13:11

@marco why would it be ambiguous, it should be a List of Lists as far as I know.... Can you print `len(contours)` and `len(hier_)` it should be the same, or something is wrong there, and also you can try to do `len(_hier[0])` it should be 4 – api55 May 11 '16 at 13:27

The contours ' len is 1335 and hier it's 1, and hier[0] it's 1335 – marco May 11 '16 at 13:33

@marco, wow, 1335 is too many contours for that image.... can you print `len(_hier[0][0])`, hopefully this is 4 and then you will have to change to `hierarchy[0][i][3]`. However, I will be kind of weird if you think of the output explained in the documentation.... – api55 May 11 '16 at 13:39

I find it wierd. But I think it works. The code is : holes = [millCnts[i] for i in range(len(millCnts)) if hier[0][i][3] >= 0] However, I don't know how to draw "holes", because ther are a list, and the drawCountours only accepts countours or numpy – marco May 11 '16 at 13:51