

PS-5 REPORT

24-678

SUBMITTED BY: AKSHAY ANTONY
akshayan@andrew.cmu.edu

Pseudo Code:

- 1. Input the image
- 2. Do the required number of erosions and dilations to obtain **wall-blobs.png**
- 3. Use cv2.findContour to find all the contours
- 4. Use random colors to draw around different blobs and save them as **wall-contours.png**
- 5. Threshold out required contours based on the **arc length**, using **cv2.arcLength** function
- 6. Draw the thresholded contours on a separate white screen
- 7. Thin the contour to find the central axis and save it as **wall-cracks.png**

Wall-1.png

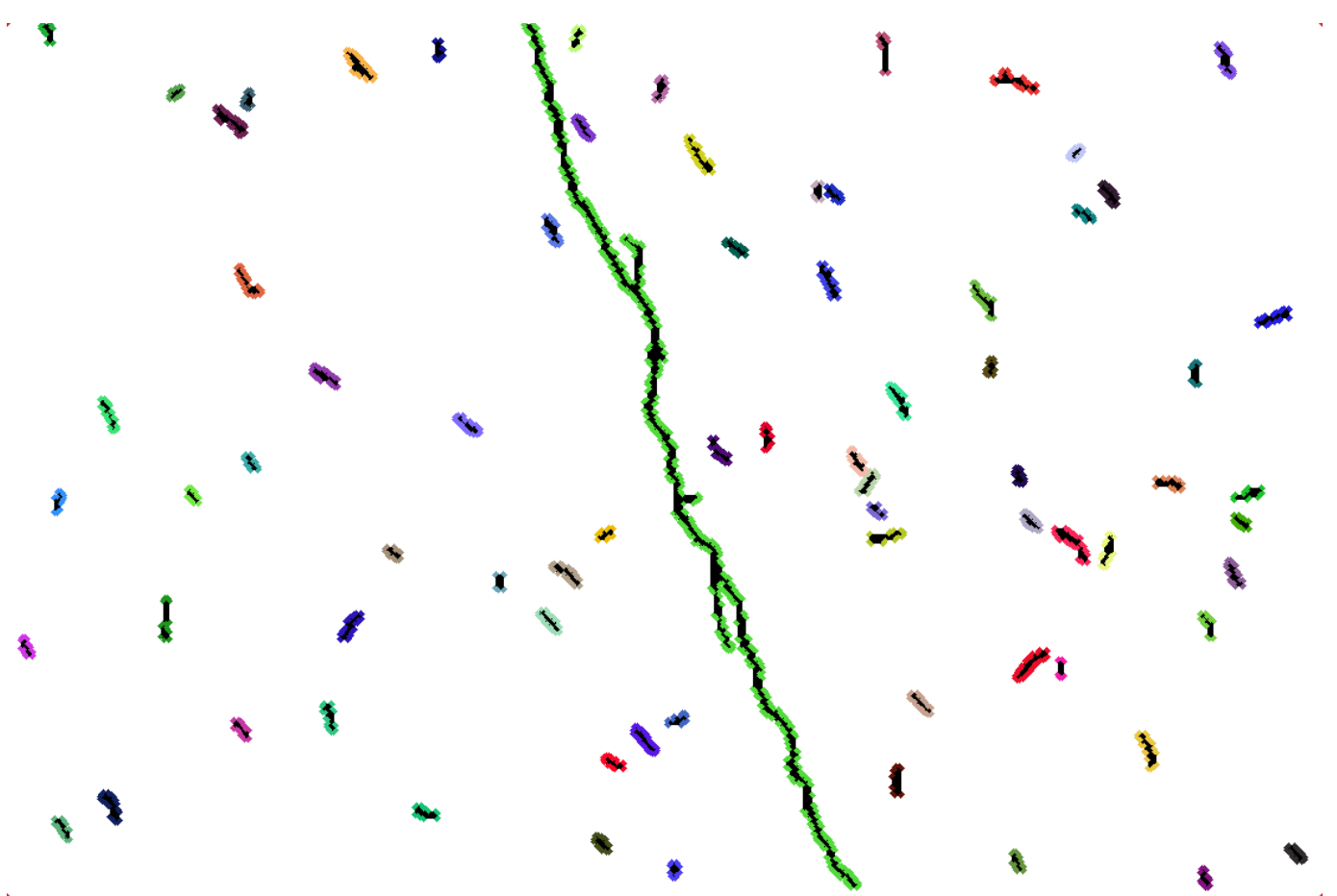
Thresholds used:

Filtered out the contours whose length is greater than 130, the actual length of the crack is around 5000 pixels

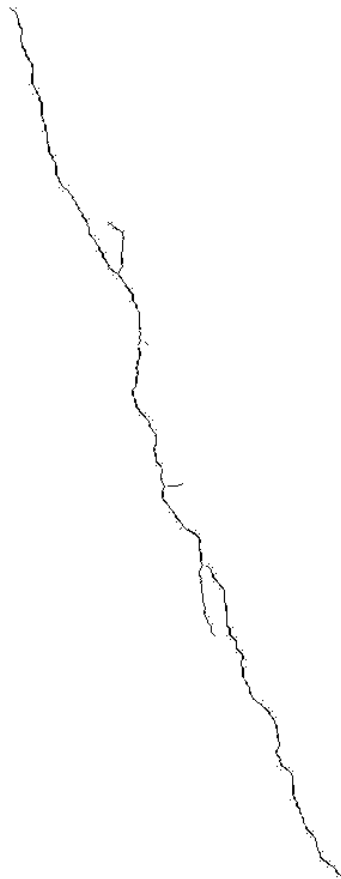
Applied two erosions and 1 dilation.



wall1-blobs.png



Walls1-contours.png



walls1-crack.png

Wall-2.png

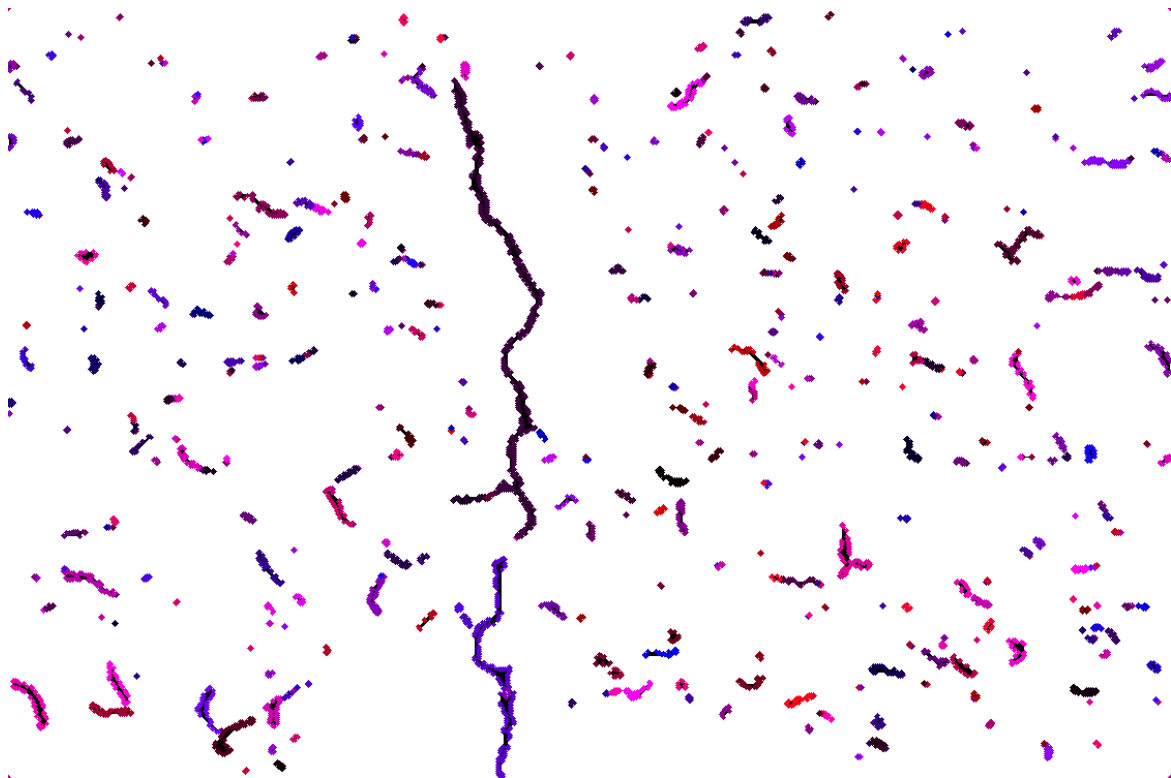
Thresholds used:

Filtered out the contours whose length is greater than 500, the actual total length of the crack is around 5000 pixels

Applied 1 erosion and 2 dilations, using MORPH_ELIPSE



Wall1-blobs.png



walls2-contours.png



Walls2-crack.png

Screenshots of code files

Readme.txt

```
1 Python version: Python 3.9.6
2 OpenCV Version: 4.5.2
3 Operating System: Linux 20.02
4 IDE: Sublime text, run via terminal
5 Almost spend 6 hours for this problem
```

```

1 import cv2
2 import numpy as np
3 import random
4
5
6 def thin(img1):
7
8     k_e = cv2.getStructuringElement(cv2.MORPH_CROSS, (3,3))
9     thin = np.zeros(img1.shape, dtype=np.uint8)
10
11     while cv2.countNonZero(img1) != 0:
12         er = cv2.erode(img1, k_e)
13         op = cv2.morphologyEx(er, cv2.MORPH_OPEN, k_e)
14         subset = er - op
15         thin = cv2.bitwise_or(subset, thin)
16         img1 = er.copy()
17
18     return thin
19
20 if __name__ == '__main__':
21
22     choice = input("Please select wall-1 or wall-2: ")
23     #choice = 'wall-2'
24     if choice == 'wall-1':
25         #inputting the images
26         image_input = cv2.imread("/home/akshay/Downloads/CV/ps-5/ps5-images/wall1.png")
27         #creating an image with full white pixels, to store only the thresholded contours
28         white_screen = np.full(image_input.shape, 255, dtype=np.uint8)
29         image_contours = image_input
30         img1 = image_input
31
32         #applying erosion twice
33         for i in range(2):
34             img1 = cv2.erode(img1, None)
35
36         img1 = cv2.dilate(img1, None)
37
38         img2 = img1.copy()
39
40         img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2GRAY)
41         image_dil_eroded = img2
42
43         img2[:,img2.shape[1]-1] = 255
44         img2[:,0] = 255
45         img2[img2.shape[0]-1,:] = 255
46         img2[0,:] = 255

```

```

45 img2[img2.shape[0]-1,:] = 255
46 img2[0,:] = 255
47
48 #detecting contours
49 contours, hierarchy = cv2.findContours(img2, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
50
51 #showing the images
52 cv2.imshow("original", image_input)
53 cv2.imshow("eroded_image", image_dil_eroded)
54
55 img2 = cv2.cvtColor(img2, cv2.COLOR_GRAY2BGR)
56
57 for contour in contours:
58     cv2.drawContours(img2, contour, -1, (random.randint(0,255),random.randint(0,255),random.randint(0,255)), 4)
59
60 cv2.imshow("contours", img2)
61
62 #finding all the contours that have a length greater than 130, which is the long central crack
63 threshold_contours = []
64 for contour in contours:
65     if (cv2.arcLength(contour, True) > 130):
66         print(cv2.arcLength(contour, True))
67         threshold_contours.append(contour)
68
69 #drawing the thresholded contour on the white screen
70 cv2.drawContours(white_screen, threshold_contours, -1, (0,0,0), cv2.FILLED)
71 white_screen = cv2.cvtColor(white_screen, cv2.COLOR_BGR2GRAY)
72
73 #thinning the image according to the given function
74 thinned = thin(white_screen)
75 thinned = cv2.bitwise_not(thinned)
76
77 #saving all the images
78 cv2.imwrite("wall1-blobs.png", image_dil_eroded)
79 cv2.imwrite("wall1-contours.png", img2)
80 cv2.imwrite("wall1-cracks.png", thinned)
81 cv2.imshow("final_crack", thinned)
82
83 else:
84     #inputting the images
85     image_input = cv2.imread("/home/akshay/Downloads/CV/ps-5/ps5-images/wall2.png")
86
87     #creating an image with full white pixels, to store only the thresholded contours
88     white_screen = np.full(image_input.shape, 255, dtype=np.uint8)
89     image_contours = image_input
90     img1 = image_input

```

```

90 img1 = image_input
91
92 #applying erosion and dialtion
93 ke = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3,3))
94 img1 = cv2.erode(img1, ke)
95 img1 = cv2.dilate(img1, ke)
96 img1 = cv2.dilate(img1, ke)
97
98 img2 = img1.copy()
99
100 img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2GRAY)
101 image_dil_eroded = img2
102
103 #making border pixels white else border will be joined with contours and produce weong results
104 img2[:,1123] = 255
105 img2[:,0] = 255
106 img2[744,:] = 255
107 img2[0,:] = 255
108
109 #finding contours
110 contours, hierarchy = cv2.findContours(img2, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
111
112 #showing the images
113 cv2.imshow("original", image_input)
114 cv2.imshow("eroded_image", image_dil_eroded)
115
116 img2 = cv2.cvtColor(img2, cv2.COLOR_GRAY2BGR)
117
118 #displaying countours with random colors
119 for contour in contours:
120     color = (random.randint(0,255), 0, random.randint(0,255))
121     cv2.drawContours(img2, contour, -1, color, 4)
122
123 cv2.imshow("contours", img2)
124
125 #finding all the contours that have a length greater than 500, which is the long central crack
126 threshold_contours = []
127 for contour in contours:
128     if (cv2.arcLength(contour, True) > 500):
129         print(cv2.arcLength(contour, True))
130         threshold_contours.append(contour)
131
132 #drawing the thresholded contour on the white scree
133 cv2.drawContours(white_screen, threshold_contours, -1, (0,0,0), cv2.FILLED)
134 white_screen = cv2.cvtColor(white_screen, cv2.COLOR_BGR2GRAY)
135

```

kite: Indexing, © tabnine, Line 36, Column 26

```

135
136 #thinning the image according to the given function
137 thinned = thin(white_screen)
138 thinned = cv2.bitwise_not(thinned)
139
140 #saving all the images
141 cv2.imwrite("wall2-blobs.png", image_dil_eroded)
142 cv2.imwrite("wall2-contours.png", img2)
143 cv2.imwrite("wall2-cracks.png", thinned)
144 cv2.imshow("final_crack", thinned)
145
146 cv2.waitKey(0)

```