

Forum:[Hack Forums](#)

Thread: [Counting circles?](#)

XRiBE - November 16, 2015, 6:56 am

Hi,

I'm interested in writing some python code which accepts an image, and just counts all the circles in the image. It's more of a "I want to try, and achieve this so I can feel like I'm doing something productive with my life rather than playing games my entire uni holidays" type of things.

So:

1. Load image
2. Do something to count all circles
3. Output number of circles, number of half circles, number of overlapped circles

How would I achieve this? I do understand that there will be massive hurdles, and obstetrical such as; a circle might be a solid color, have just a border, the border might be 0.01px etc. So yeah.

I'm assuming HF has a few coders that are really OP, and might be able to share some advice or tips?

WojeT - November 16, 2015, 9:07 am

This post on SO answers most of your questions:

<http://stackoverflow.com/questions/11424002/how-to-detect-simple-geometric-shapes-using-opencv>

You might want to take some computer vision classes or read some material online:

<http://cs.brown.edu/courses/cs143/>

<https://www.udacity.com/course/introduction-to-computer-vision--ud810>

https://www.youtube.com/watch?v=715uLCHt4jE&list=PLd3hlSJjX_lmk_BPmB_H3AQjFKZS9XgZm

Pentahedron - November 19, 2015, 3:55 pm

this is my first ever post on HF, so sorry in advance for any conventions I'm not following.

definitely as WojeT stated, you should first start with downloading opencv, and also, you will need numpy. "pip install cv2" and "pip install numpy" (if I remember correctly).

opencv is an open source library with all the tools you will need for this, and more :)

numpy "is the fundamental package for scientific computing" (numpy site)

As you said, it is first needed to load an image. do this by creating an opencv image object, and read the file into it

```
Code >>> import cv2
>>> import numpy as np
>>> img = cv2.imread("IMAGE PATH")
```

YAY! done with part one :). Next, we need to "Do something to count all circles". How to execute this step is dependant on how nitty gritty you want to get. Theoretically you could make your own algorithm and scan the image for circle(s), pixel by pixel. However, for simplicity I recommend you use the standard pre included function `cv2.HoughCircles(<parameters>)`. Saying anything beyond would take the challenge out of it ;p

This does have limitations. What are they? (hint: you have a valid concern regarding over-lapping circles)

I just made a simple py program to find simply full circles. I'm not sure if to share it with you (obviously if you didn't want to write it yourself you would search on github or something ?).. PM me if you would like me to post it :). I'm here for any questions.

Useful tut's (remove all \$, I'm not allowed to post links):

http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_houghcircles/py_houghcircles.html

<http://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/>

<http://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/>

*lastly, a small note:

a pixel is the smallest unit in a picture. There is no such thing as "half a pixel" or anything similar. Pixel counts are in whole numbers.

so "the border might be 0.01px" can't happen :p, but it can very well be 1 pixel.

hope I helped :)