

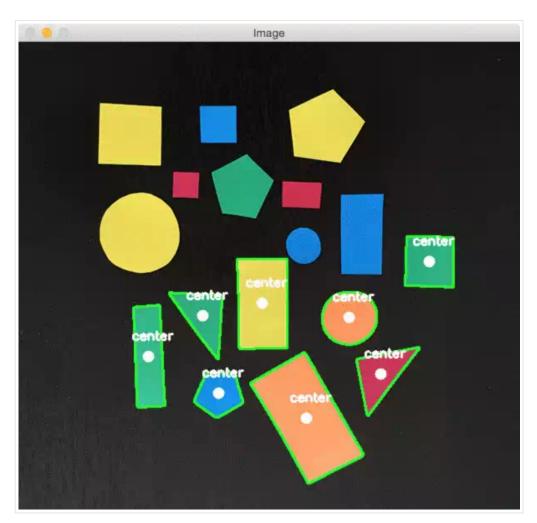
Navigation



OpenCV center of contour

by Adrian Rosebrock on February 1, 2016 in Image Processing, OpenCV 3, Tutorials





Today, we are going to start a new 3-part series of tutorials on **shape detection and analysis**.

Throughout this series, we'll learn how to:

- 1. Compute the center of a contour/shape region.
- 2. Recognize various shapes, such as circles, squares, rectangles, triangles, and pentagons using only contour properties.
- 3. Label the color of a shape.

While today's post is a bit basic (at least in context or some or the more advanced concepts on the PylmageSearch blog recently), it still addresses a question that I get asked a lot:



"How do I compute the center of a cont

In today's post, I'll answer that question.

And in later posts in this series, we'll build upon ou images.

OpenCV center of contour

Looking for the source code to this Jump right to the downloads section

Jump right to the downloads sec

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

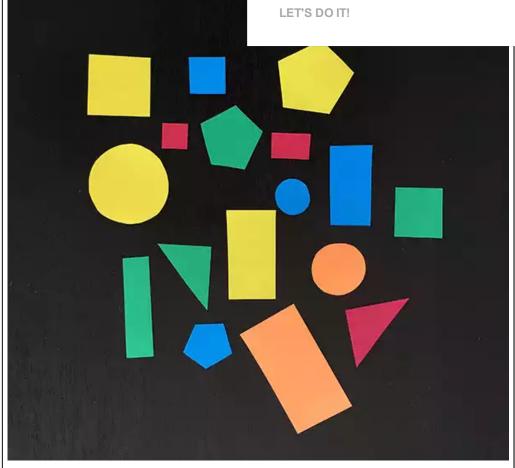


Figure 1: An example image containing a set of shapes that we are going to compute the center of the contour for.

In above image, you can see a variety of shapes cut out from pieces of construction paper. Notice how these shapes are not *perfect*. The rectangles aren't quite rectangular — and the circles are not entirely circular either. These are *human drawn and human cut out shapes*. implying there is *variation* in each shape type.

Free 21-day crash course on

computer vision & image search engines

With this in mind, the goal of today's tutorial is to (1) detect the outline or each shape in the image, followed by (2) computing the center of the contour — also called the centroid of the region.

In order to accomplish these goals, we'll need to pe

- Conversion to grayscale.
- Blurring to reduce high frequency noise to male
- Binarization of the image. Typically edge detection this post, we'll be applying thresholding.

Before we start coding, make sure you have the in

```
OpenCV center of contour
1 $ pip install imutils
```

From there, we can go ahead and get started.

Open up a new file, name it center_of_shape.py

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

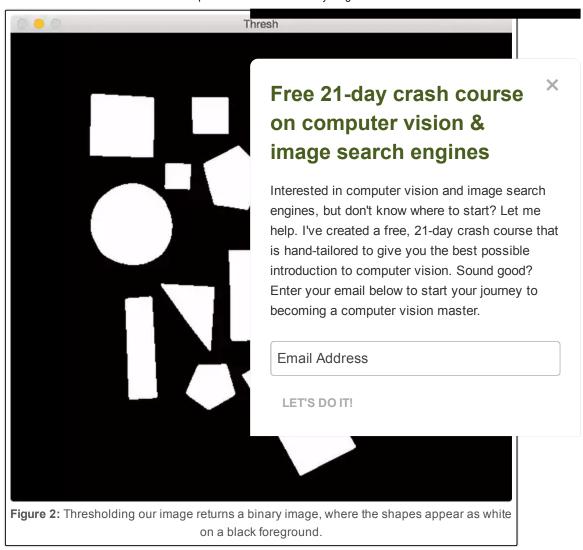
Email Address

```
OpenCV center of contour
                                                                                           n
                                                 LET'S DO IT!
   # import the necessary packages
  import argparse
3
   import imutils
4 import cv2
5
6 # construct the argument parse and parse the arguments
7
  ap = argparse.ArgumentParser()
  ap.add_argument("-i", "--image", required=True,
       help="path to the input image")
9
10 args = vars(ap.parse_args())
11
12 # load the image, convert it to grayscale, blur it slightly,
13 # and threshold it
14 image = cv2.imread(args["image"])
15 gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
16 blurred = cv2.GaussianBlur(gray, (5, 5), 0)
17 thresh = cv2.threshold(blurred, 60, 255, cv2.THRESH_BINARY)[1]
```

We start off on **Lines 2-4** by importing our necessary packages, followed by parsing our command line arguments. We only need a single switch here, --image, which is the path to where the image we want to process resides on disk.

We then take this image, load it from disk, and pre-process it by applying grayscale conversion, Gaussian smoothing using a 5×5 kernel, and finally thresholding (**Lines 14-17**).

The output of the thresholding operation can be seen below:



Notice how after applying thresholding the shapes are represented as a *white foreground* on a *black background*.

The next step is to find the location of these white regions using contour detection:

A call to cv2.findContours on Lines 20 and 21 returns the set of outlines (i.e., contours) that correspond to each of the white blobs on the image. Line 22 then grabs the appropriate tuple value based on whether we are using OpenCV 2.4 or OpenCV 3. You can read more about how the return signature of cv2.findContours changed between OpenCV versions in this post.

We are now ready to process each of the contours:

```
OpenCV center of contour
                                                                                       Python
24 # loop over the contours
25 for c in cnts:
26
      # compute the center of the contour
27
       M = cv2.moments(c)
       cX = int(M["m10"] / M["m00"])
28
                                                Free 21-day crash course on
29
       cY = int(M["m01"] / M["m00"])
                                                computer vision & image search
30
                                                engines
31
       # draw the contour and center of the
```

```
32
            cv2.drawContours(image, [c], -1, (0,
            cv2.circle(image, (cX, cY), 7, (255, 255, 255), -1)
cv2.putText(image, "center", (cX - 20, cY - 20),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), -1)
33
34
35
36
37
            # show the image
            cv2.imshow("Image", image)
38
39
            cv2.waitKey(0)
```

On **Line 25** we start looping over each of the indivi moments for the contour region on Line 27.

In computer vision and image processing, image m an object in an image. These moments capture bas the area of the object, the centroid (i.e., the center with other desirable properties.

Here we are only interested in the center of the col

From there, Lines 32-34 handle:

- Drawing the outline of the contour surrounding cv2.drawContours .
- Placing a white circle at the *center* (cx, cy) -coordinates of the shape.

Free 21-day crash course on computer vision & image search engines

X

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

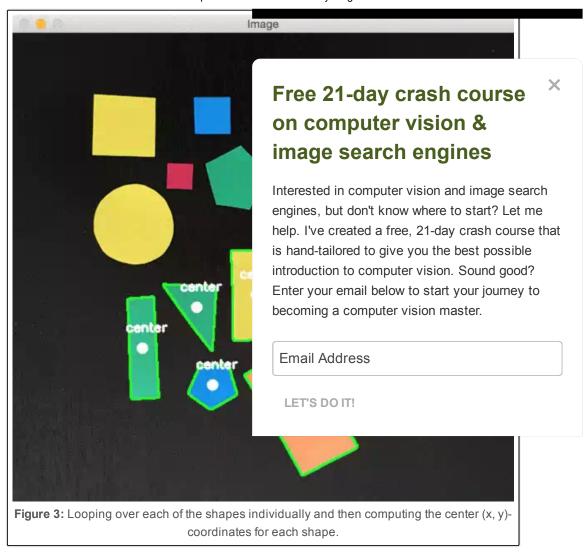
LET'S DO IT!

Writing the text center near the white circle.

To execute our script, just open up a terminal and execute the following command:

```
OpenCV center of contour
                                                                                       Shell
1 $ python center_of_shape.py --image shapes_and_colors.png
```

Your results should look something like this:



Notice how each of the shapes are successfully detected, followed by the center of the contour being computed and drawn on the image.

Summary

In this lesson, we learned how to compute the center of a contour using OpenCV and Python.

This post is the *first* in a *three part series* on **shape analysis**.

In next week's post, we'll learn how to identify shapes in an image.

Then, two weeks from now, we'll learn how to *analyze the color of each shape* and *label the shape with a specific color* (i.e., "red", "green", "blue", etc.).

To be notified when these posts go live, be sure to enter your email address using the form below!

Downloads:

If you would like to download the code and images used in this post, please enter your email address in

the form below. Not only will you get a .zip of the co **Guide** on Computer Vision and Image Search Eng on this blog! Sound good? If so, enter your email a

Free 21-day crash course on computer vision & image search engines

ost



Email address:

Your email address

DOWNLOAD THE CODE!

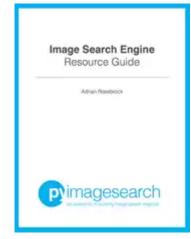
Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!

Resource Guide (it's totally free).



Enter your email ad Engine Resource (don't publish on this your own!

Your email address

DOWNLOAD THE

approximate contours, contour properties, contours, find contours, shapes

Real-time panorama and image stitching with OpenCV

OpenCV shape detection >

X

23 Responses to OpenCV center of contour



Luis Jose February 1, 2016 at 1:41 pm #

REPLY 🦴

Hi Adrian!

Great job. I'm looking forward to completing this series of posts.

Thanks for sharing your knowledge with the world!!!



Adrian Rosebrock February 2, 2016 at 10:32 am #

REPLY 🦴

Thanks Luis!



Marco February 1, 2016 at 2:28 pm #

Hi Adrian, another great Tutorial, but ho

Adrian Rosebrock February 2, 2016

You would have to copy and paste e using IDLE, you should also look into using I friendly.

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!

Harley Mackenzie February 1, 2016 at 6:5

There seems to be good support for opequivalent routines for line detection. I have four discriminating between lots of small noise lines a

Adrian Rosebrock February 2, 2016

Line detection is much, much more

method to perform line detection is to use the Hough Lines transform. But for noisy images, you'll often get mixed results.



Boško February 2, 2016 at 8:06 am #

REPLY

REPLY 🦴

Hello Andrian,

I got "ZeroDivisionError: float division by zero", because all "m" values are 0. Why? Where I wrong? I trying solve it but do not have luck. I use Python 2.7 and openCV 3.1.

Thanks,

Best Regards.

Boško

Adrian Rosebrock February 2, 2016 at 10:24 am #

REPLY 🦴

It seems like both you and Ruttunenn are getting the same error message. It seems like the segmentation may not be perfect and there is some noise left over in the thresholding. A simple check would be to use:

1 if cv2.contourArea(c) > MIN_THRESH:

process the contour

Where you can set MIN_THRESH to be a suitable value to filter out small contour regions.



Boško February 2, 2016 at 3:17 pm #

REPLY +



Thanks! It's work now



Adrian Rosebrock Februar

Awesome, I'm happy it work



leena February 9, 2016 at 3:49 am #

I also got the same and resolved wit

if (M["m00"] == 0): M["m00"]=1

regards

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!



ESPLondon February 26, 2016 at 9:33 am #

Thanks leena, that worked







Ruttunenn February 2, 2016 at 10:21 am #

REPLY

Hi,

Just run to a minor glitch in the example as I was getting zeros on the M = cv2.moments(c) on the first iteration, leading to float division by zero. A simple work around was to implement a check for 0.0 results.

Cheers for awesome tutorials anyway.



David February 3, 2016 at 10:20 pm #

REPLY

Hi, excellent post Adrian!!!

Could you please explain a bit more why on the pre-processing stage you slightly blur the image???

Thanks.

David Darias

Adrian Rosebrock February 4, 2016



Blurring (also called "smoothing") is used to smooth night frequency not Simply put, this allows us to ignore the details in the image and focus on what matters

— the shapes. So by blurring, we sm allowing the thresholding and contour extrac



Free 21-day crash course on computer vision & image search engines Ken Doman February 9, 2016 at 6:42 pm



Great post, Adrian!

It may be a little off topic, but I'm curious how the shaped features. Would the centroid be inside the

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!

Adrian Rosebrock February 10, 2010 Great question. It would still be insid

example can be found here. Keep in mind th calculation of the centroid.



gary February 11, 2016 at 4:43 pm #

REPLY

Hi Adrian.

I have a question about the value of cX and cY. As i want to know what is the pixel value at the point (cX, cY), i tried to print it by image[cX,cY]. However, I got error like:

IndexError: index 1040 is out of bounds for axis 0 with size 1024

which means that cX and cY is outside of range of the image size. Therefore, I want to ask how can i find out the pixel coordinate at point (cX, cY)?

Thanks

Adrian Rosebrock February 12, 2016 at 3:19 pm #

REPLY

When accessing pixel values in OpenCV + NumPy, you actually specify them in (y, x) order rather than (x, y) order. Thus, you need to use: image[cY, cX]



thecanadiran October 12, 2016 at 4:15 pm #

REPLY 5

Thanks for a great tutorial Adrian.

Could you please explain here or in another tutorial how to use image moments to characterize the other shape and statistical properties of an object?

Adrian Rosebrock October 13, 2016 at 9:12 am #

REPLY **←**



In fact, I have already done that! Ple Zernike Moments to characterize the shape inside the PylmageSearch Gurus course.

Trackbacks/Pingbacks

OpenCV shape detection - PyImageSearch - Februal [...] Last week we learned how to compute the center Determining object color with OpenCV - PyImageSe [...] Compute the center of a contour [...]

Leave a Reply

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!

Name (required)
Email (will not be published) (required
Website

SUBMIT COMMENT

Resource Guide (it's totally free).



Click the button below to get my free 11-page Image Search Engine Resource Guide PDF. Uncover exclusive techniques that I don't publish on this blog and start building image search engines of your own.

Download for Free!

You can detect faces in images & video.



Are you interested in **detecting faces in images & vide**Then let me help! I guarantee that my new book will turn
weekend. Click here to give it a shot yourself.

CLICK HERE TO MASTER FACE DETECTION

PylmageSearch Gurus: NOW ENROLLING!

Free 21-day crash course on computer vision & image search engines

X

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!

The PylmageSearch Gurus course is now enrolling! Inside the course you'll learn how to perform:

- Automatic License Plate Recognition (ANPR)
- Deep Learning
- Face Recognition
- and much more!

Click the button below to learn more about the course, take a tour, and get 10 (FREE) sample lessons.

TAKE A TOUR & GET 10 (FREE) LESSONS

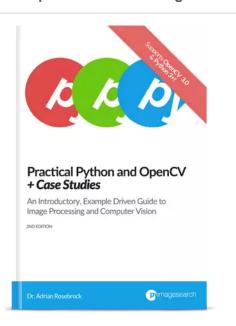
Hello! I'm Adrian Rosebrock.

I'm an entrepreneur and Ph.D who has launched two successful image search engines. ID My Pill and Chic Engine.

I'm here to share my tips, tricks, and hacks I've learned



Learn computer vision in a single weekend.



Free 21-day crash course on computer vision & image search engines

X

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to

becoming a computer vision master.

Email Address

LET'S DO IT!

Want to learn computer vision & OpenCV? I can teach you in a **single weekend**. I know. It sounds crazy, but it's no joke. My new book is your **guaranteed**, **quick-start guide** to becoming an OpenCV Ninja. So why not give it a try? Click here to become a computer vision ninja.

CLICK HERE TO BECOME AN OPENCY NINJA

Subscribe via RSS



Never miss a post! Subscribe to the PylmageSearch RSS Feed and keep up to date with my image search engine tutorials, tips, and tricks

POPULAR

Install OpenCV and Python on your Raspberry Pi 2 and B+

FEBRUARY 23, 2015

Home surveillance and motion detection with the Raspberry Pi, Python, OpenCV, and Dropbox

JUNE 1, 2015

How to install OpenCV 3 on Raspbian Jessie

OCTOBER 26, 2015

Install OpenCV 3.0 and Python 2.7+ on OSX

JUNE 15, 2015

Install OpenCV 3.0 and Python 2.7+ on Ubuntu JUNE 22, 2015

Accessing the Raspberry Pi Camera with OpenCV and Python MARCH 30, 2015

Basic motion detection and tracking with Python and Or MAY 25,2015

Search

Search...

Find me on **Twitter**, **Facebook**, **Google+**, and **LinkedIn** © 2016 PyImageSearch. All Rights Reserved.

Free 21-day crash course on computer vision & image search engines



Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!