

Filling holes in an image using OpenCV (Python / C++)

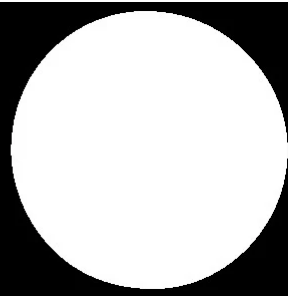
(<https://learnopencv.com/>)



Satya Mallick (<https://learnopencv.com/author/spmallick/>).

NOVEMBER 23, 2015

[how-to](https://learnopencv.com/category/how-to/) (<https://learnopencv.com/category/how-to/>).



(</wp-content/uploads/2015/11/imfill.jpg>)

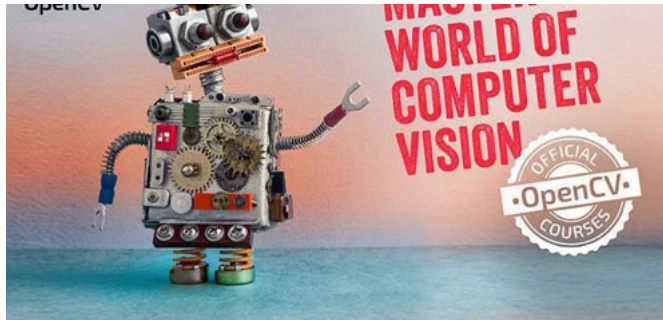
Figure 1. Left : Image of a nickel. Center : Thresholded and inverted. Right : Holes filled.

In this tutorial we will learn how to fill holes in a binary image. Consider the image on the left in Figure 1. Let's say we want to find a binary mask that separates the coin from the background as shown in the right image. In this tutorial the circular region that contains the coin will also be referred to as the foreground.

Notice that the boundary of the coin is dark and distinct from its white background. So, we use simple image [thresholding](#) ([opencv-threshold-python-cpp](#)) to separate the boundary from the background. In other words, we say pixels with intensities above a certain value (threshold) are the background and the rest are the foreground. The center image shows this thresholded image (black represents background, and white represents foreground). Unfortunately, even though the boundary has been nicely extracted (it is solid white), the interior of the coin has intensities similar to the background. Therefore, the thresholding operation cannot distinguish it from the background. How do we fill all pixels inside the circular boundary with white ?

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it. [Privacy policy](https://learnopencv.com/privacy-policy/) (<https://learnopencv.com/privacy-policy/>)

Accept



(<http://opencv.org/courses/>)

Start your exciting journey from an absolute Beginner to Mastery in AI,
Computer Vision & Deep Learning!

Learn More
(<https://opencv.org/courses/>)

MATLAB has a function called `imfill` (<http://www.mathworks.com/help/images/ref/imfill.html>) that allows you to fill holes, and you can use it in the following way.

```
1 % MATLAB code for filling holes in a binary image.  
2 im = imfill(im, 'holes');
```

imfill in OpenCV

There is no `imfill` function in OpenCV, but we can surely write one! The idea is rather simple. We know the pixel (0,0) is connected to the background. So we can extract the background, by simply doing a floodfill operation from pixel (0, 0). Pixels that are not affected by the floodfill operation are necessarily inside the boundary. The flood-filled image when inverted and combined with the thresholded image gives the foreground mask!

Steps for implementing imfill in OpenCV

The image and corresponding steps are given below.

Read in the image.

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it. [Privacy policy](https://learnopencv.com/privacy-policy/) (<https://learnopencv.com/privacy-policy/>)

Accept



(/wp-content/uploads/2015/11/imfill-opencv-steps.jpg).

Figure 2.

Threshold the input image to obtain a binary image.

Flood fill from pixel (0, 0). Notice the difference between the outputs of step 2 and step 3 is that the background in step 3 is now white.

Invert the flood filled image (i.e. black becomes white and white becomes black).

Combine the thresholded image with the inverted flood filled image using bitwise OR operation to obtain the final foreground mask with holes filled in. The image in Step 4 has some black areas inside the boundary. By design the image in Step 2 has those holes filled in. So we combine the two to get the mask.

in a binary image

C++ and Python code for filling holes

Download Code To easily follow along this tutorial, please download code by clicking on the button below. It's FREE!

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it. [Privacy policy](https://learnopencv.com/privacy-policy/) (https://learnopencv.com/privacy-policy/)

Accept

And here is how it is done in code

C++

```
1 #include "opencv2/opencv.hpp"
2
3 using namespace cv;
4
5 int main(int argc, char **argv)
6 {
7     // Read image
8     Mat im_in = imread("nickel.jpg", IMREAD_GRAYSCALE);
9
10    // Threshold.
11    // Set values equal to or above 220 to 0.
12    // Set values below 220 to 255.
13    Mat im_th;
14    threshold(im_in, im_th, 220, 255, THRESH_BINARY_INV);
15
16    // Floodfill from point (0, 0)
17    Mat im_floodfill = im_th.clone();
18    floodFill(im_floodfill, cv::Point(0,0), Scalar(255));
19
20    // Invert floodfilled image
21    Mat im_floodfill_inv;
22    bitwise_not(im_floodfill, im_floodfill_inv);
23
24    // Combine the two images to get the foreground.
25    Mat im_out = (im_th | im_floodfill_inv);
26
27    // Display images
28    imshow("Thresholded Image", im_th);
29    imshow("Floodfilled Image", im_floodfill);
30    imshow("Inverted Floodfilled Image", im_floodfill_inv);
31    imshow("Foreground", im_out);
32    waitKey(0);
33 }
```

Python

```
1 import cv2;
2 import numpy as np;
3
4 # Read image
5 im_in = cv2.imread("nickel.jpg", cv2.IMREAD_GRAYSCALE);
6
7 # Threshold.
8 # Set values equal to or above 220 to 0.
9 # Set values below 220 to 255
```

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it. [Privacy policy \(https://learnopencv.com/privacy-policy/\)](https://learnopencv.com/privacy-policy/)

Accept

```
15
16 # Mask used to flood filling.
17 # Notice the size needs to be 2 pixels than the image.
18 h, w = im_th.shape[:2]
19 mask = np.zeros((h+2, w+2), np.uint8)
20
21 # Floodfill from point (0, 0)
22 cv2.floodFill(im_floodfill, mask, (0,0), 255);
23
24 # Invert floodfilled image
25 im_floodfill_inv = cv2.bitwise_not(im_floodfill)
26
27 # Combine the two images to get the foreground.
28 im_out = im_th | im_floodfill_inv
29
30 # Display images.
31 cv2.imshow("Thresholded Image", im_th)
32 cv2.imshow("Floodfilled Image", im_floodfill)
33 cv2.imshow("Inverted Floodfilled Image", im_floodfill_inv)
34 cv2.imshow("Foreground", im_out)
35 cv2.waitKey(0)
```

Other techniques

There are other ways to solve the same problem. One way is to use **morphological close** operation. However, for morphological operations to work you will need to know the maximum size of the hole. Another way is to use **findContours** to find the contours and then fill it in using **drawContours**. I prefer the simplicity and speed of the technique described in this post.

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it. [Privacy policy \(https://learnopencv.com/privacy-policy/\)](https://learnopencv.com/privacy-policy/)

Accept

Subscribe & Download Code

If you liked this article and would like to download code (C++ and Python) and example images used in this post, please [click here](#). Alternately, sign up to receive a free [Computer Vision Resource](#) Guide. In our newsletter, we share OpenCV tutorials and examples written in C++/Python, and Computer Vision and Machine Learning algorithms and news.

[Download Example Code](#)

Subscribe Now

Disclaimer

All views expressed on this site are my own and do not represent the opinions of OpenCV.org or any entity whatsoever with which I have been, am now, or will be affiliated.

Getting Started

[Installation](#)

[PyTorch](#)

Course

[Opencv Courses](#)

[CV4Faces \(Old\)](#)

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it. [Privacy policy \(https://learnopencv.com/privacy-policy/\)](https://learnopencv.com/privacy-policy/)

Accept

Resource Guide

Information

[Privacy Policy](#)[Terms and Conditions](#)

About LearnOpenCV

In 2007, right after finishing my Ph.D., I co-founded TAAZ Inc. with my advisor Dr. David Kriegman and Kevin Barnes. The scalability, and robustness of our computer vision and machine learning algorithms have been put to rigorous test by more than 100M users who have tried our products.

[Read More
\(https://learnopencv.com/about/\)](https://learnopencv.com/about/)

Copyright © 2021 – BIG VISION LLC

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it. [Privacy policy \(https://learnopencv.com/privacy-policy/\)](https://learnopencv.com/privacy-policy/)

Accept