

PYIMAGESEA RCH

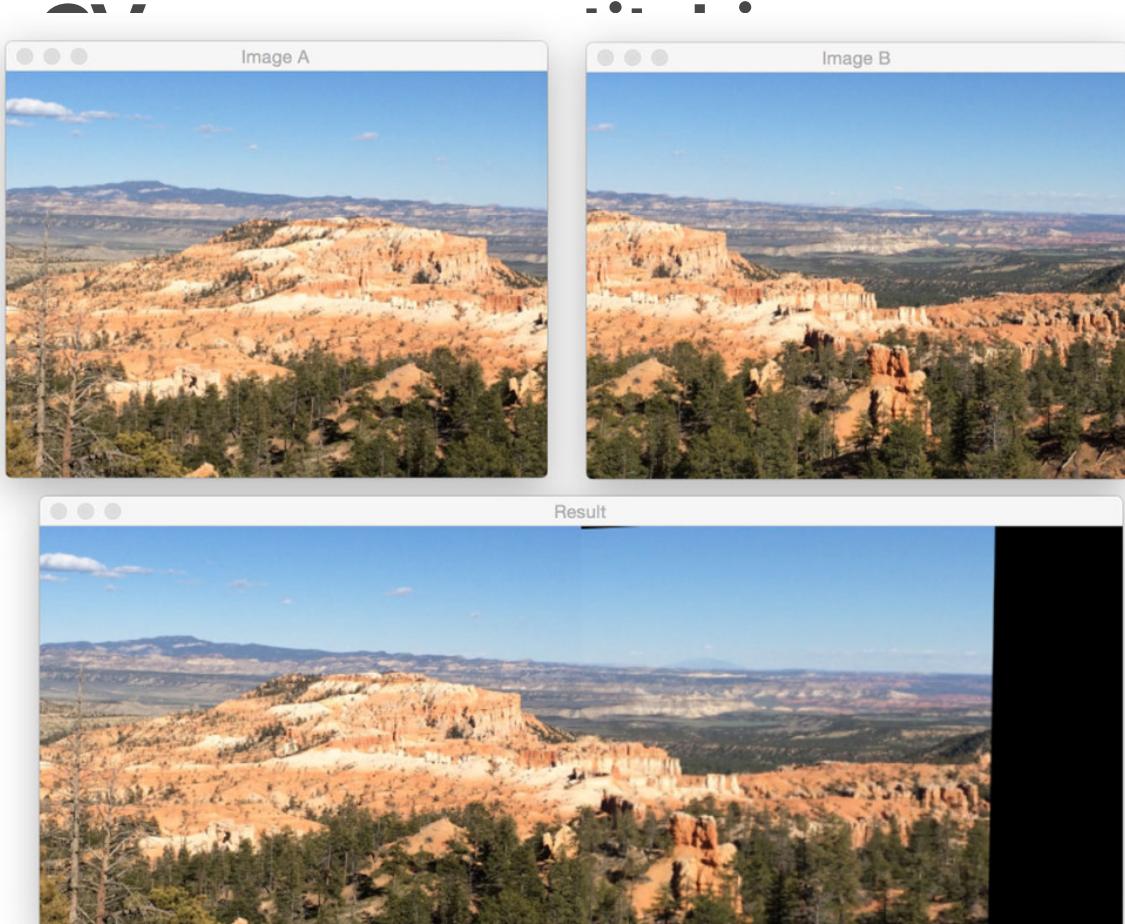
[IMAGE DESCRIPTORS \(HTTPS://WWW.PYIMAGESEARCH.COM/CATEGORY/IMAGE-DESCRIPTORS/\).](https://www.pyimagesearch.com/category/image-descriptors/)

[OPENCV TUTORIALS \(HTTPS://WWW.PYIMAGESEARCH.COM/CATEGORY/OPENCV/\).](https://www.pyimagesearch.com/category/opencv/)

[TUTORIALS \(HTTPS://WWW.PYIMAGESEARCH.COM/CATEGORY/TUTORIALS/\).](https://www.pyimagesearch.com/category/tutorials/)

Open

by [Adrian](#)



(https://pyimagesearch.com/wp-content/uploads/2016/01/bryce_result_02.jpg)

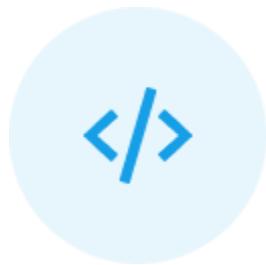
In today's blog post, I'll demonstrate how to perform ***image stitching and panorama construction*** using Python and OpenCV. Given two images, we'll "stitch" them together to create a simple panorama, as seen in the example above.

To construct our image panorama, we'll utilize computer vision and image processing techniques such as: keypoint detection and local invariant descriptors; keypoint matching; RANSAC; and perspective warping.

Since there are [major differences in how OpenCV 2.4.X and OpenCV 3.X handle keypoint detection and local invariant descriptors](#) (<https://pyimagesearch.com/2015/07/16/where-did-sift-and-surf-go-in-opencv-3/>) (such as SIFT and SURF), I've taken special care to provide code that is compatible with **both** versions (provided that you compiled OpenCV 3 with `opencv_contrib` support, of course).

In future blog posts we'll extend our panorama stitching code to work with multiple images rather than just two.

Read on to find out how panorama stitching with OpenCV is done.



Looking for the source code to this post?

[JUMP RIGHT TO THE DOWNLOADS SECTION →](#)

OpenCV panorama stitching

Our panorama stitching algorithm consists of four steps:

- **Step #1:** Detect keypoints (DoG, Harris, etc.) and extract local invariant descriptors (SIFT, SURF, etc.) from the two input images.
- **Step #2:** Match the descriptors between the two images.
- **Step #3:** Use the [RANSAC algorithm](#) (<https://en.wikipedia.org/wiki/RANSAC>) to estimate a [homography matrix](#) ([https://en.wikipedia.org/wiki/Homography_\(computer_vision\)](https://en.wikipedia.org/wiki/Homography_(computer_vision))) using our matched feature vectors.
- **Step #4:** Apply a warping transformation using the homography matrix obtained from **Step #3**.

We'll encapsulate all four of these steps inside `panorama.py`, where we'll define a `Stitcher` class used to construct our panoramas.

The `Stitcher` class will rely on the [imutils \(<https://github.com/jrosebr1/imutils>\)](https://github.com/jrosebr1/imutils) Python package, so if you don't already have it installed on your system, you'll want to go ahead and do that now:

```
OpenCV panorama stitching
1. | $ pip install imutils
```

Let's go ahead and get started by reviewing `panorama.py` :

```
OpenCV panorama stitching
1. | # import the necessary packages
2. | import numpy as np
3. | import imutils
4. | import cv2
5.
6. | class Stitcher:
7. |     def __init__(self):
8. |         # determine if we are using OpenCV v3.X
9. |         self.isv3 = imutils.is_cv3(or_better=True)
```

We start off on **Lines 2-4** by importing our necessary packages. We'll be using NumPy for matrix/array operations, `imutils` for a set of OpenCV convenience methods, and finally `cv2` for our OpenCV bindings.

From there, we define the `Stitcher` class on **Line 6**. The constructor to `Stitcher` simply checks which version of OpenCV we are using by making a call to the `is_cv3` method. Since there are [major differences in how OpenCV 2.4 and OpenCV 3 handle keypoint detection and local invariant descriptors \(<https://pyimagesearch.com/2015/07/16/where-did-sift-and-surf-go-in-opencv-3/>\)](https://pyimagesearch.com/2015/07/16/where-did-sift-and-surf-go-in-opencv-3/), it's important that we determine the version of OpenCV that we are using.

Next up, let's start working on the `stitch` method:

```
OpenCV panorama stitching
11. |     def stitch(self, images, ratio=0.75, reprojThresh=4.0,
12. |                 showMatches=False):
13. |         # unpack the images, then detect keypoints and extract
14. |         # local invariant descriptors from them
15. |         (imageB, imageA) = images
16. |         (kpsA, featuresA) = self.detectAndDescribe(imageA)
17. |         (kpsB, featuresB) = self.detectAndDescribe(imageB)
18.
19. |         # match features between the two images
20. |         M = self.matchKeypoints(kpsA, kpsB,
21.
22. |             featuresA, featuresB, ratio, reprojThresh)
23.
24. |         # if the match is None, then there aren't enough matched
25. |         # keypoints to create a panorama
26. |         if M is None:
27. |             return None
```

The `stitch` method requires only a single parameter, `images`, which is the list of (two) images that we are going to stitch together to form the panorama.

We can also optionally supply `ratio`, used for David Lowe's ratio test when matching features (more on this ratio test later in the tutorial), `reprojThresh` which is the maximum pixel "wiggle room" allowed by the RANSAC algorithm, and finally `showMatches`, a boolean used to indicate if the keypoint matches should be visualized or not.

Line 15 unpacks the `images` list (which again, we presume to contain only two images). The ordering to the `images` list is important: **we expect images to be supplied in left-to-right order**. If images are *not* supplied in this order, then our code will still run — but our output panorama will only contain one image, not both.

Once we have unpacked the `images` list, we make a call to the `detectAndDescribe` method on **Lines 16 and 17**. This method simply detects keypoints and extracts local invariant descriptors (i.e., SIFT) from the two images.

Given the keypoints and features, we use `matchKeypoints` (**Lines 20 and 21**) to match the features in the two images. We'll define this method later in the lesson.

If the returned matches `M` are `None`, then not enough keypoints were matched to create a panorama, so we simply return to the calling function (**Lines 25 and 26**).

Otherwise, we are now ready to apply the perspective transform:

```
OpenCV panorama stitching
28.     # otherwise, apply a perspective warp to stitch the images
29.     # together
30.     (matches, H, status) = M
31.     result = cv2.warpPerspective(imageA, H,
32.         (imageA.shape[1] + imageB.shape[1], imageA.shape[0]))
33.     result[0:imageB.shape[0], 0:imageB.shape[1]] = imageB
34.
35.     # check to see if the keypoint matches should be visualized
36.     if showMatches:
37.         vis = self.drawMatches(imageA, imageB, kpsA, kpsB, matches,
38.             status)
39.
40.         # return a tuple of the stitched image and the
41.         # visualization
42.         return (result, vis)
43.
44.     # return the stitched image
45.     return result
```

Provided that `M` is not `None`, we unpack the tuple on **Line 30**, giving us a list of keypoint matches, the homography matrix `H` derived from the RANSAC algorithm, and finally

`status`, a list of indexes to indicate which keypoints in `matches` were successfully spatially verified using RANSAC.

Given our homography matrix `H`, we are now ready to stitch the two images together. First, we make a call to `cv2.warpPerspective` which requires three arguments: the image we want to warp (in this case, the *right* image), the 3×3 transformation matrix (`H`), and finally the shape out of the output image. We derive the shape out of the output image by taking the sum of the widths of both images and then using the height of the second image.

Line 30 makes a check to see if we should visualize the keypoint matches, and if so, we make a call to `drawMatches` and return a tuple of both the panorama and visualization to the calling method (**Lines 37-42**).

Otherwise, we simply returned the stitched image (**Line 45**).

Now that the `stitch` method has been defined, let's look into some of the helper methods that it calls. We'll start with `detectAndDescribe`:

```
OpenCV panorama stitching
47.     def detectAndDescribe(self, image):
48.         # convert the image to grayscale
49.         gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
50.
51.         # check to see if we are using OpenCV 3.X
52.         if self.isv3:
53.             # detect and extract features from the image
54.             descriptor = cv2.xfeatures2d.SIFT_create()
55.             (kps, features) = descriptor.detectAndCompute(image, None)
56.
57.         # otherwise, we are using OpenCV 2.4.X
58.         else:
59.             # detect keypoints in the image
60.             detector = cv2.FeatureDetector_create("SIFT")
61.             kps = detector.detect(gray)
62.
63.             # extract features from the image
64.             extractor = cv2.DescriptorExtractor_create("SIFT")
65.             (kps, features) = extractor.compute(gray, kps)
66.
67.             # convert the keypoints from KeyPoint objects to NumPy
68.             # arrays
69.             kps = np.float32([kp.pt for kp in kps])
70.
71.             # return a tuple of keypoints and features
72.             return (kps, features)
```

As the name suggests, the `detectAndDescribe` method accepts an image, then detects keypoints and extracts local invariant descriptors. In our implementation we use the

Difference of Gaussian

([http://www.cs.utexas.edu/~grauman/courses/fall2009/papers/local features synthesis draft.pdf](http://www.cs.utexas.edu/~grauman/courses/fall2009/papers/local_features_synthesis_draft.pdf)).

[pdf](#)) (DoG) keypoint detector and the [SIFT feature extractor \(\[https://en.wikipedia.org/wiki/Scale-invariant_feature_transform\]\(https://en.wikipedia.org/wiki/Scale-invariant_feature_transform\)\)](#).

On **Line 52** we check to see if we are using OpenCV 3.X. If we are, then we use the `cv2.xfeatures2d.SIFT_create` function to instantiate both our DoG keypoint detector and SIFT feature extractor. A call to `detectAndCompute` handles extracting the keypoints and features (**Lines 54 and 55**).

It's important to note that you **must** have compiled OpenCV 3.X with [opencv_contrib \(\[https://github.com/itseez/opencv_contrib\]\(https://github.com/itseez/opencv_contrib\)\)](#) support enabled. If you did not, you'll get an error such as `AttributeError: 'module' object has no attribute 'xfeatures2d'`. If that's the case, head over to my [OpenCV 3 tutorials page \(<https://pyimagesearch.com/opencv-tutorials-resources-guides/>\)](#) where I detail how to install OpenCV 3 with `opencv_contrib` support enabled for a variety of operating systems and Python versions.

Lines 58-65 handle if we are using OpenCV 2.4. The `cv2.FeatureDetector_create` function instantiates our keypoint detector (DoG). A call to `detect` returns our set of keypoints.

From there, we need to initialize `cv2.DescriptorExtractor_create` using the SIFT keyword to setup our SIFT feature extractor . Calling the `compute` method of the `extractor` returns a set of feature vectors which quantify the region surrounding each of the detected keypoints in the image.

Finally, our keypoints are converted from `KeyPoint` objects to a NumPy array (**Line 69**) and returned to the calling method (**Line 72**).

Next up, let's look at the `matchKeypoints` method:

```
OpenCV panorama stitching
74.     def matchKeypoints(self, kpsA, kpsB, featuresA, featuresB,
75.                         ratio, reprojThresh):
76.                             # compute the raw matches and initialize the list of actual
77.                             # matches
78.                             matcher = cv2.DescriptorMatcher_create("BruteForce")
79.                             rawMatches = matcher.knnMatch(featuresA, featuresB, 2)
80.                             matches = []
81.
82.                             # loop over the raw matches
83.                             for m in rawMatches:
84.                                 # ensure the distance is within a certain ratio of each
85.                                 # other (i.e. Lowe's ratio test)
86.                                 if len(m) == 2 and m[0].distance < m[1].distance * ratio:
87.                                     matches.append((m[0].trainIdx, m[0].queryIdx))
```

The `matchKeypoints` function requires four arguments: the keypoints and feature vectors associated with the first image, followed by the keypoints and feature vectors associated with the second image. David Lowe's `ratio` test variable and RANSAC re-projection threshold are also

SECOND IMAGE. DAVID LOWE'S RATIO TEST VARIABLE AND RANSAC RE-PROJECTION THRESHOLD ARE ALSO
be supplied.

Matching features together is actually a fairly straightforward process. We simply loop over the descriptors from both images, compute the distances, and find the smallest distance for each pair of descriptors. Since this is a very common practice in computer vision, OpenCV has a built-in function called `cv2.DescriptorMatcher_create` that constructs the feature matcher for us. The `BruteForce` value indicates that we are going to *exhaustively* compute the Euclidean distance between *all feature vectors* from both images and find the pairs of descriptors that have the smallest distance.

A call to `knnMatch` on **Line 79** performs [k-NN matching \(https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm\)](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm) between the two feature vector sets using $k=2$ (indicating the top two matches for each feature vector are returned).

The reason we want the top *two* matches rather than just the top one match is because we need to apply David Lowe's ratio test for false-positive match pruning.

Again, **Line 79** computes the `rawMatches` for each pair of descriptors — but there is a chance that some of these pairs are false positives, meaning that the image patches are not actually true matches. In an attempt to prune these false-positive matches, we can loop over each of the `rawMatches` individually (**Line 83**) and apply Lowe's ratio test, which is used to determine high-quality feature matches. Typical values for Lowe's ratio are normally in the range $[0.7, 0.8]$.

Once we have obtained the `matches` using Lowe's ratio test, we can compute the homography between the two sets of keypoints:

```
OpenCV panorama stitching
89.     # computing a homography requires at least 4 matches
90.     if len(matches) > 4:
91.         # construct the two sets of points
92.         ptsA = np.float32([kpA[i] for (_, i) in matches])
93.         ptsB = np.float32([kpB[i] for (i, _) in matches])
94.
95.         # compute the homography between the two sets of points
96.         (H, status) = cv2.findHomography(ptsA, ptsB, cv2.RANSAC,
97.                                         reprojThresh)
98.
99.         # return the matches along with the homography matrix
100.        # and status of each matched point
101.        return (matches, H, status)
102.
103.    # otherwise, no homography could be computed
104.    return None
```

Computing a homography between two sets of points requires *at a bare minimum* an initial set of four matches. For a more reliable homography estimation, we should have substantially more ~~than just four matched points~~

train just your matched points.

Finally, the last method in our `Stitcher` method, `drawMatches` is used to visualize keypoint correspondences between two images:

OpenCV panorama stitching

```
106.     def drawMatches(self, imageA, imageB, kpsA, kpsB, matches, status):
107.         # initialize the output visualization image
108.         (hA, wA) = imageA.shape[:2]
109.         (hB, wB) = imageB.shape[:2]
110.         vis = np.zeros((max(hA, hB), wA + wB, 3), dtype="uint8")
111.         vis[0:hA, 0:wA] = imageA
112.         vis[0:hB, wA:] = imageB
113.
114.         # loop over the matches
115.         for ((trainIdx, queryIdx), s) in zip(matches, status):
116.             # only process the match if the keypoint was successfully
117.             # matched
118.             if s == 1:
119.                 # draw the match
120.                 ptA = (int(kpsA[queryIdx][0]), int(kpsA[queryIdx][1]))
121.                 ptB = (int(kpsB[trainIdx][0]) + wA, int(kpsB[trainIdx][1]))
122.                 cv2.line(vis, ptA, ptB, (0, 255, 0), 1)
123.
124.         # return the visualization
125.         return vis
```

This method requires that we pass in the two original images, the set of keypoints associated with each image, the initial matches after applying Lowe's ratio test, and finally the `status` list provided by the homography calculation. Using these variables, we can visualize the "inlier" keypoints by drawing a straight line from keypoint *N* in the first image to keypoint *M* in the second image.

Now that we have our `Stitcher` class defined, let's move on to creating the `stitch.py` driver script:

OpenCV panorama stitching

```
1.  # import the necessary packages
2.  from pyimagesearch.panorama import Stitcher
3.  import argparse
4.  import imutils
5.  import cv2
6.
7.  # construct the argument parse and parse the arguments
8.  ap = argparse.ArgumentParser()
9.  ap.add_argument("-f", "--first", required=True,
10.    help="path to the first image")
11.  ap.add_argument("-s", "--second", required=True,
12.    help="path to the second image")
13.  args = vars(ap.parse_args())
```

We start off by importing our required packages on **Lines 2-5**. Notice how we've placed the `panorama.py` and `Stitcher` class into the `pyimagesearch` module just to keep our

code tidy.

Note: If you are following along with this post and having trouble organizing your code, please be sure to download the source code using the form at the bottom of this post. The .zip of the code download will run out of the box without any errors.

From there, **Lines 8-14** parse our command line arguments: `--first`, which is the path to the first image in our panorama (the *left-most* image), and `--second`, the path to the second image in the panorama (the *right-most* image).

Remember, these image paths need to be supplied in *left-to-right* order!

The rest of the `stitch.py` driver script simply handles loading our images, resizing them (so they can fit on our screen), and constructing our panorama:

```
OpenCV panorama stitching
15.  # load the two images and resize them to have a width of 400 pixels
16.  # (for faster processing)
17.  imageA = cv2.imread(args["first"])
18.  imageB = cv2.imread(args["second"])
19.  imageA = imutils.resize(imageA, width=400)
20.  imageB = imutils.resize(imageB, width=400)
21.
22.  # stitch the images together to create a panorama
23.  stitcher = Stitcher()
24.  (result, vis) = stitcher.stitch([imageA, imageB], showMatches=True)
25.
26.  # show the images
27.  cv2.imshow("Image A", imageA)
28.  cv2.imshow("Image B", imageB)
29.  cv2.imshow("Keypoint Matches", vis)
30.  cv2.imshow("Result", result)
31.  cv2.waitKey(0)
```

Once our images are loaded and resized, we initialize our `Stitcher` class on **Line 23**. We then call the `stitch` method, passing in our two images (**again, in left-to-right order**) and indicate that we would like to visualize the keypoint matches between the two images.

Finally, **Lines 27-31** display our output images to our screen.

Panorama stitching results

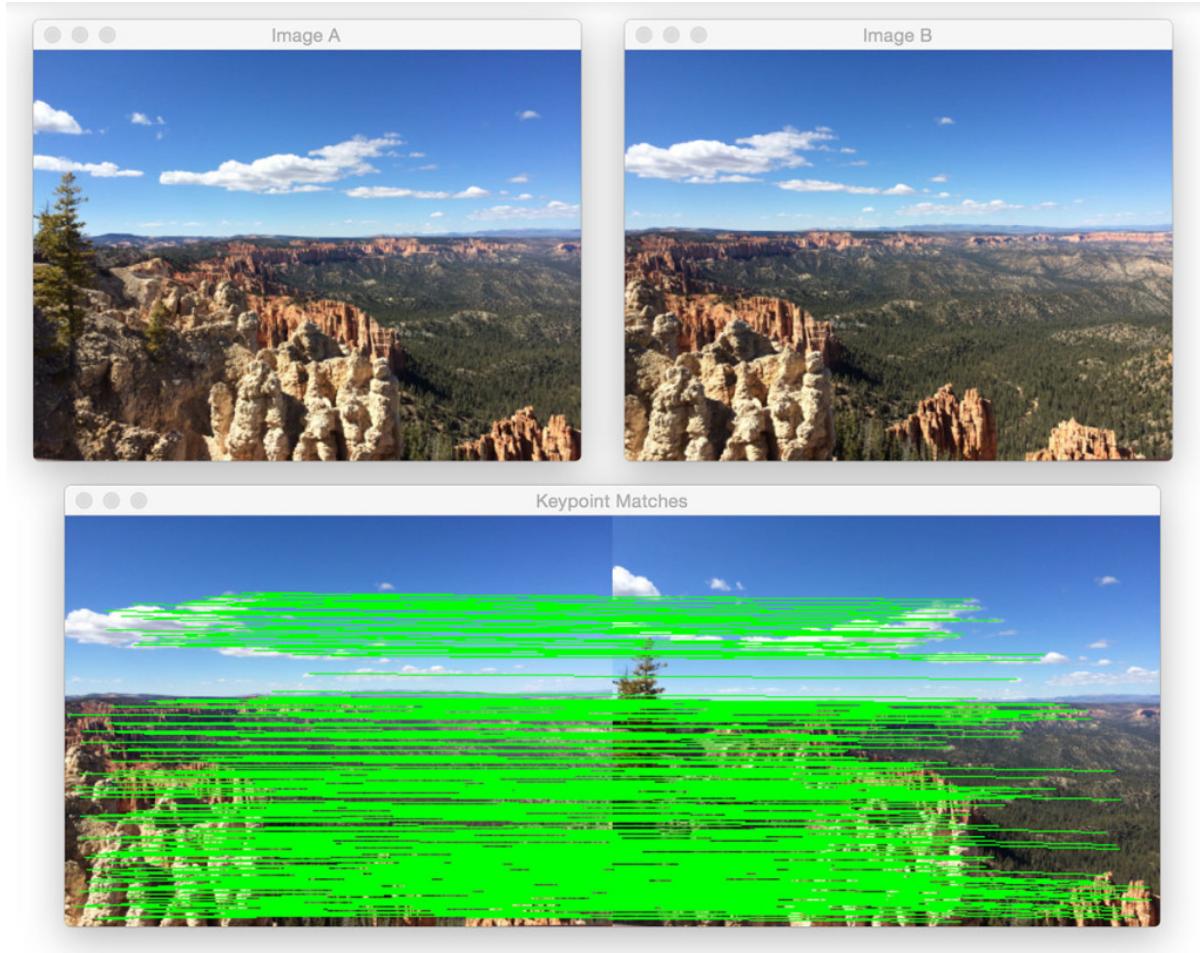
In mid-2014 I took a trip out to Arizona and Utah to enjoy the national parks. Along the way I stopped at many locations, including Bryce Canyon, Grand Canyon, and Sedona. Given that these areas contain beautiful scenic views, I naturally took a bunch of photos — some of which are perfect for constructing panoramas. I've included a sample of these images in today's blog to

demonstrate panorama stitching.

All that said, let's give our OpenCV panorama stitcher a try. Open up a terminal and issue the following command:

OpenCV panorama stitching

```
1. | $ python stitch.py --first images/bryce_left_01.png \
2. |   --second images/bryce_right_01.png
```



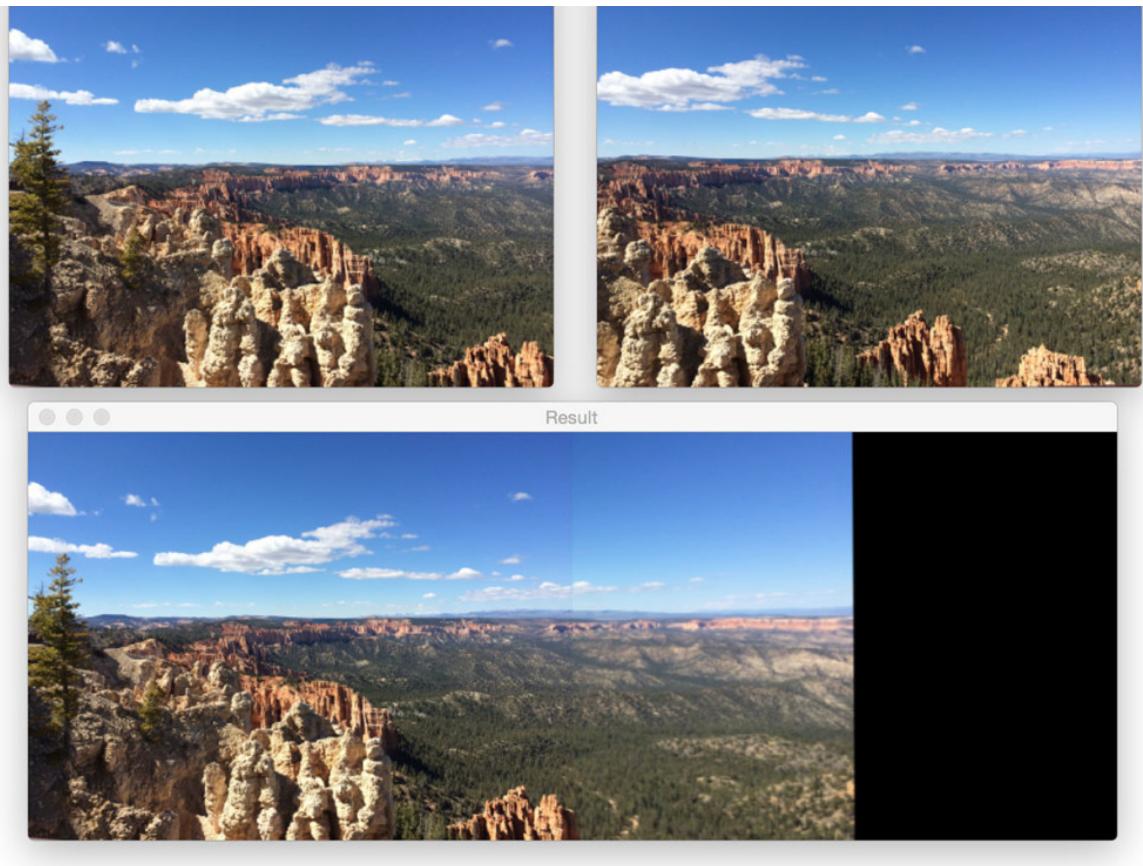
(https://pyimagesearch.com/wp-content/uploads/2016/01/bryce_match_01.jpg)

Figure 1: (Top) The two input images from Bryce canyon (in left-to-right order). (Bottom) The matched keypoint correspondences between the two images.

At the *top* of this figure, we can see two input images (resized to fit on my screen, the raw .jpg files are a much higher resolution). And on the *bottom*, we can see the matched keypoints between the two images.

Using these matched keypoints, we can apply a perspective transform and obtain the final panorama:





(https://pyimagesearch.com/wp-content/uploads/2016/01/bryce_result_01.jpg)

Figure 2: Constructing a panorama from our two input images.

As we can see, the two images have been successfully stitched together!

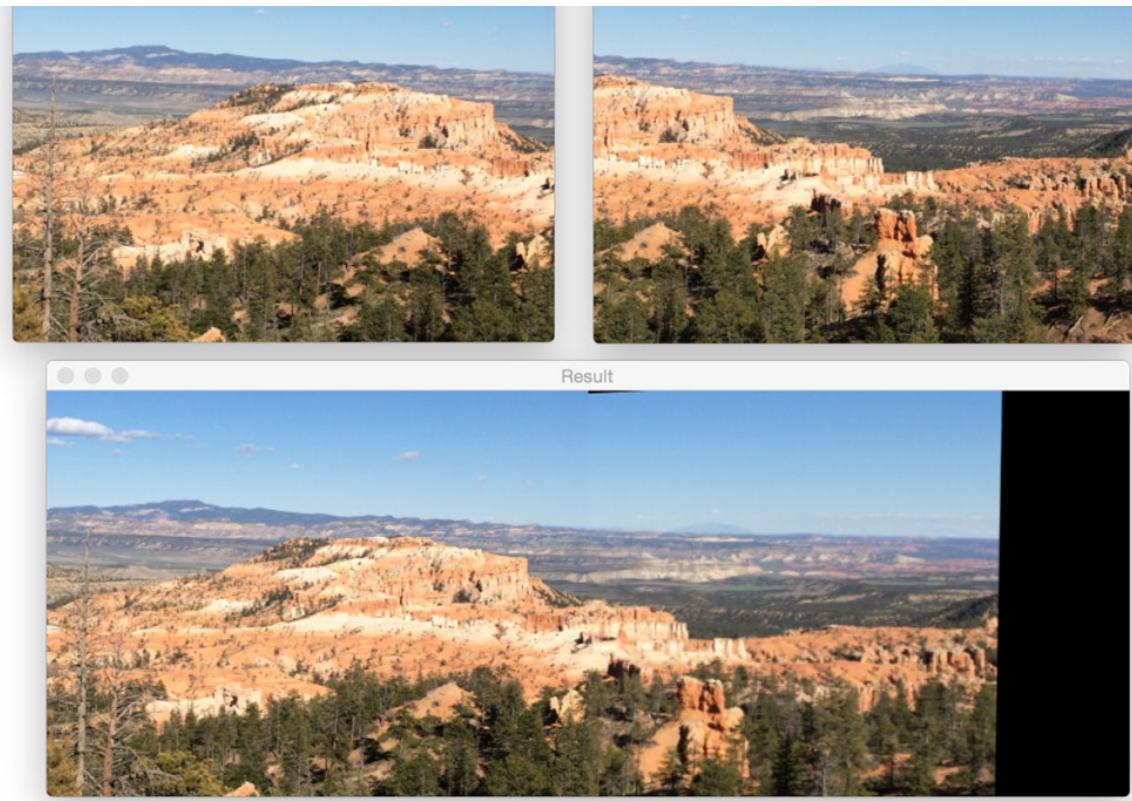
Note: On many of these example images, you'll often see a visible "seam" running through the center of the stitched images. This is because I shot many of photos using either my iPhone or a digital camera with autofocus turned on, thus the focus is slightly different between each shot. Image stitching and panorama construction work best when you use the same focus for every photo. I never intended to use these vacation photos for image stitching, otherwise I would have taken care to adjust the camera sensors. In either case, just keep in mind the seam is due to varying sensor properties at the time I took the photo and was not intentional.

Let's give another set of images a try:

OpenCV panorama stitching

1. | \$ python stitch.py --first images/bryce_left_02.png \
2. | --second images/bryce_right_02.png





(https://pyimagesearch.com/wp-content/uploads/2016/01/bryce_result_02.jpg)

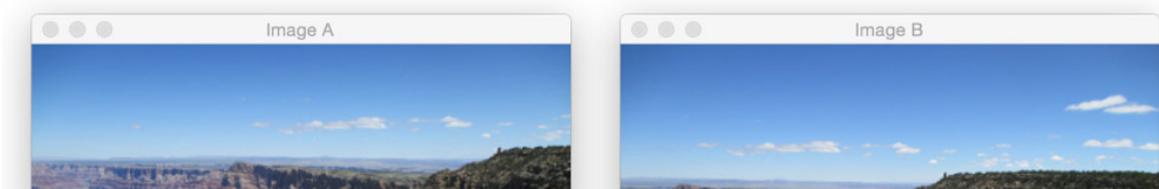
Figure 3: Another successful application of image stitching with OpenCV.

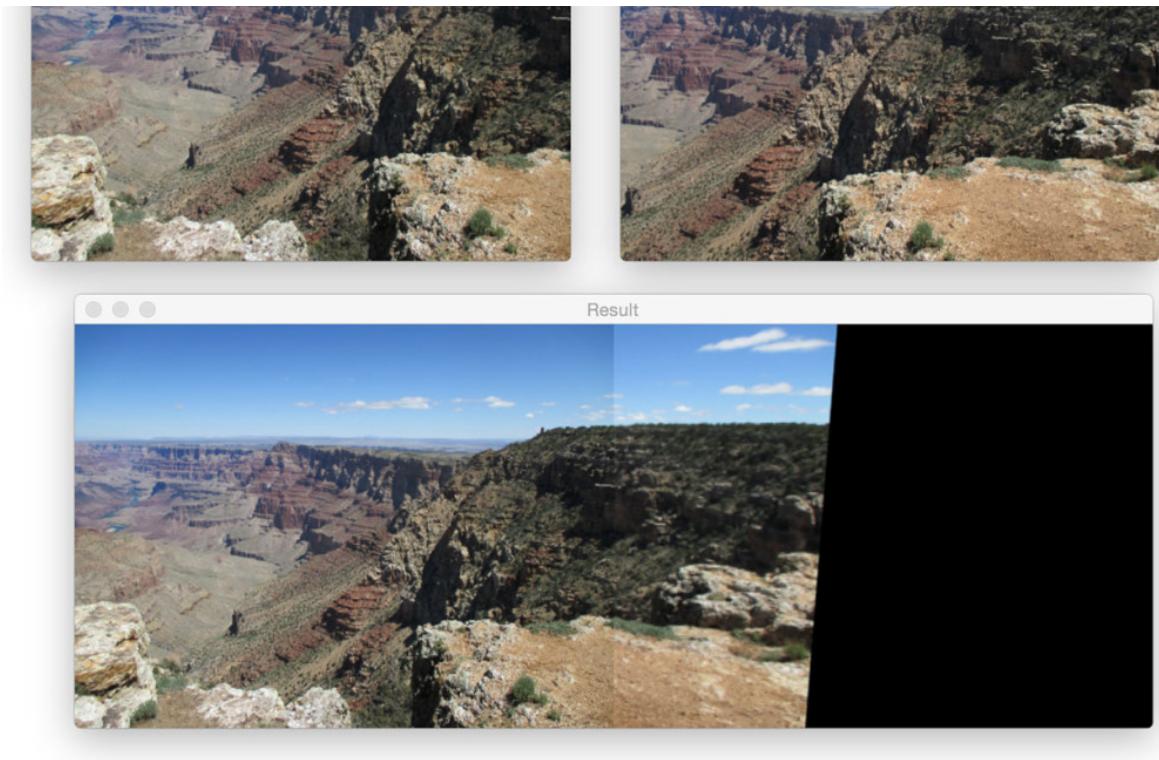
Again, our `Stitcher` class was able to construct a panorama from the two input images.

Now, let's move on to the Grand Canyon:

OpenCV panorama stitching

```
1. | $ python stitch.py --first images/grand_canyon_left_01.png \
2. |   --second images/grand_canyon_right_01.png
```





(https://pyimagesearch.com/wp-content/uploads/2016/01/grand_canyon_result_01.jpg)

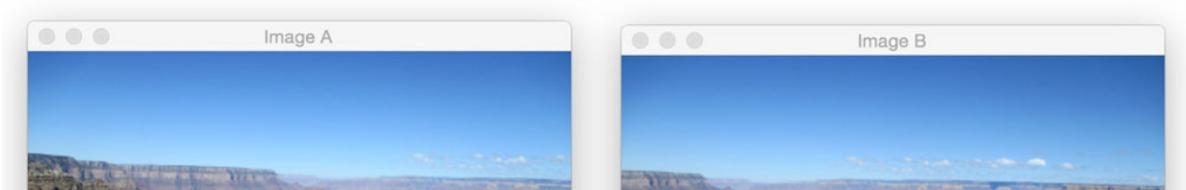
Figure 4: Applying image stitching and panorama construction using OpenCV.

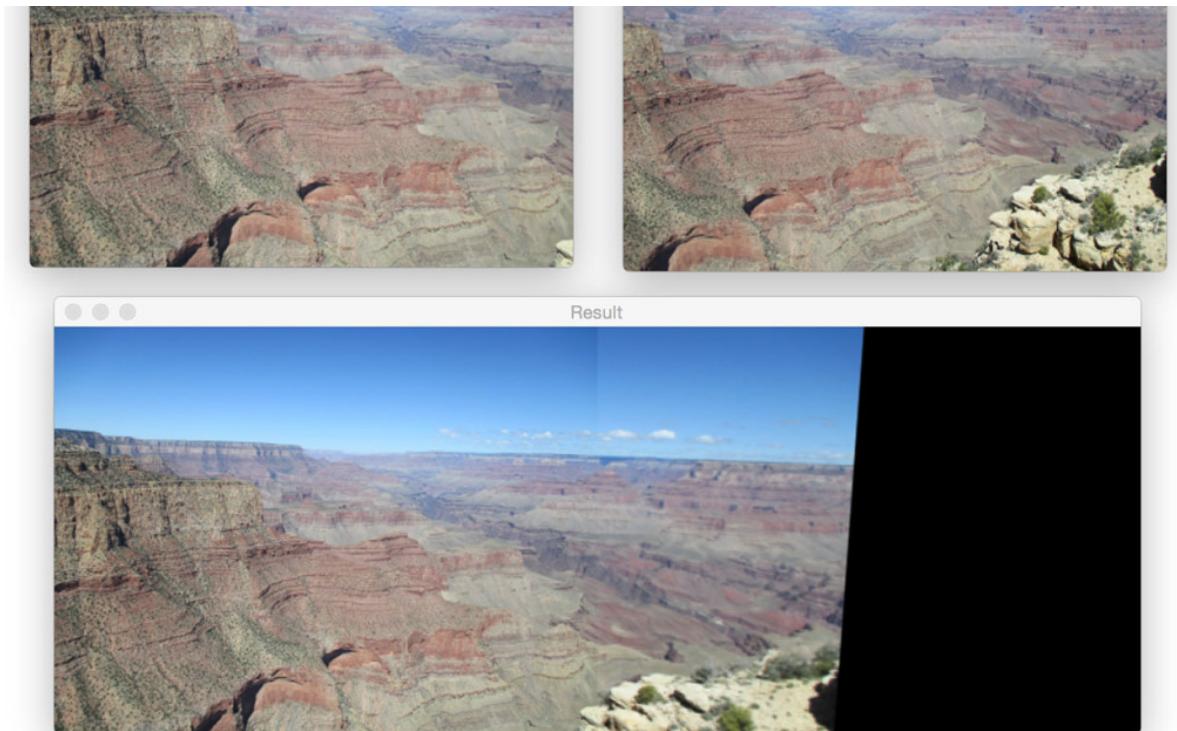
In the above input images we can see heavy overlap between the two input images. The main addition to the panorama is towards the *right side* of the stitched images where we can see more of the “ledge” is added to the output.

Here's another example from the Grand Canyon:

OpenCV panorama stitching

```
1. | $ python stitch.py --first images/grand_canyon_left_02.png \
2. |   --second images/grand_canyon_right_02.png
```





[\(https://pyimagesearch.com/wp-content/uploads/2016/01/grand_canyon_result_02.jpg\)](https://pyimagesearch.com/wp-content/uploads/2016/01/grand_canyon_result_02.jpg)

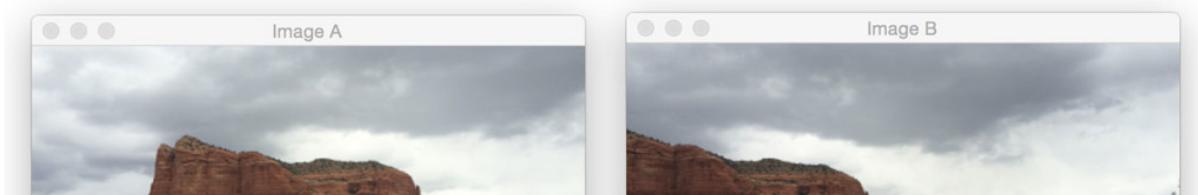
Figure 5: Using image stitching to build a panorama using OpenCV and Python.

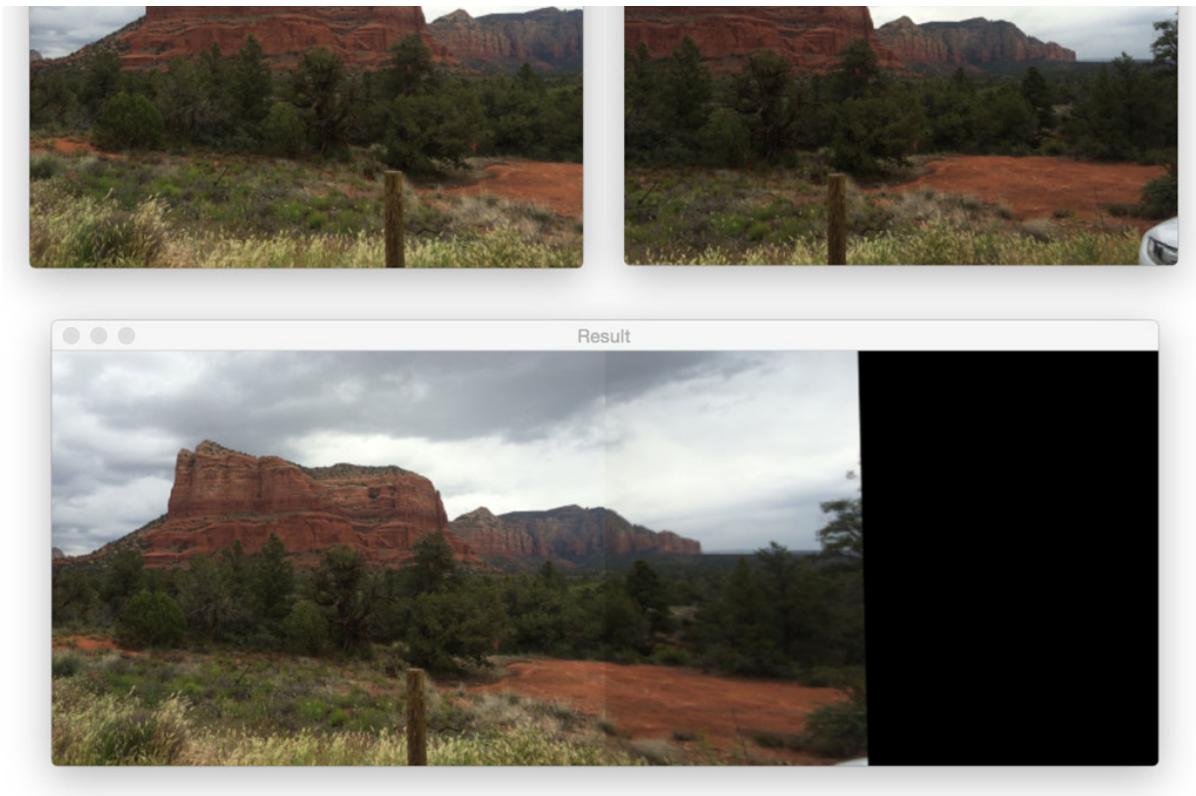
From this example, we can see that more of the *huge* expanse of the Grand Canyon has been added to the panorama.

Finally, let's wrap up this blog post with an example image stitching from Sedona, AZ:

OpenCV panorama stitching

```
1. | $ python stitch.py --first images/sedona_left_01.png \
2. |   --second images/sedona_right_01.png
```





(https://pyimagesearch.com/wp-content/uploads/2016/01/sedona_result_01.jpg)

Figure 6: One final example of applying image stitching.

Personally, I find the red rock country of Sedona to be one of the most beautiful areas I've ever visited. If you ever have a chance, definitely stop by — you won't be disappointed.

So there you have it, image stitching and panorama construction using Python and OpenCV!

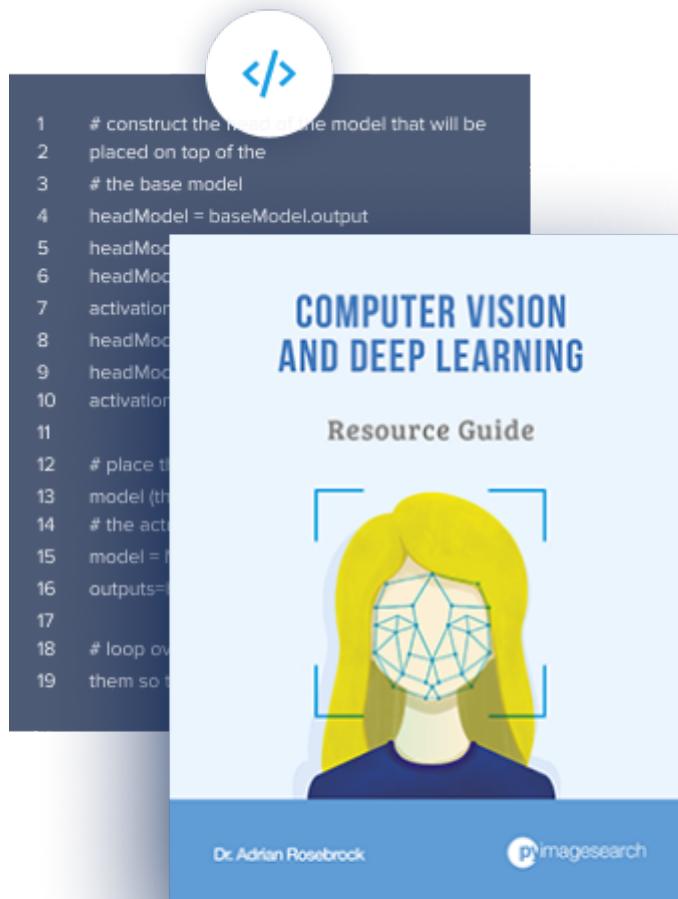
Summary

In this blog post we learned how to perform image stitching and panorama construction using OpenCV. Source code was provided for image stitching for *both* OpenCV 2.4 and OpenCV 3.

Our image stitching algorithm requires four steps: (1) detecting keypoints and extracting local invariant descriptors; (2) matching descriptors between images; (3) applying RANSAC to estimate the homography matrix; and (4) applying a warping transformation using the homography matrix.

While simple, this algorithm works well in practice when constructing panoramas for two images. In a future blog post, we'll review how to construct panoramas and perform image stitching *for more than two images*.

Anyway, I hope you enjoyed this post! **Be sure to use the form below to download the source code and give it a try.**



Download the Source Code and FREE 17-page Resource Guide

Enter your email address below to get a .zip of the code and a **FREE 17-page Resource Guide on Computer Vision, OpenCV, and Deep Learning**. Inside you'll find my hand-picked tutorials, books, courses, and libraries to help you master CV and DL!



About the Author

Hi there, I'm Adrian Rosebrock, PhD. All too often I see developers, students, and researchers wasting their time, studying the wrong things, and generally struggling to get started with Computer Vision, Deep Learning, and OpenCV. I created this website to show you what I believe is the best possible way to get your start.

[Previous Article:](#)

[Unifying picamera and cv2.VideoCapture into a single class with OpenCV](#)

[\(https://www.pyimagesearch.com/2016/01/04/unifying-picamera-and-cv2-videocapture-into-a-single-class-with-opencv/\)](https://www.pyimagesearch.com/2016/01/04/unifying-picamera-and-cv2-videocapture-into-a-single-class-with-opencv/)

[Next Article:](#)

[Multiple cameras with the Raspberry Pi and OpenCV](#)

[\(https://www.pyimagesearch.com/2016/01/18/multiple-cameras-with-the-raspberry-pi-and-opencv/\)](https://www.pyimagesearch.com/2016/01/18/multiple-cameras-with-the-raspberry-pi-and-opencv/)

242 responses to: OpenCV panorama stitching



Tito Luyo Murata

[January 11, 2016 at 1:25 pm \(https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-386440\)](https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-386440)

Still don't finish to read but looks amazing.

One question: I suppose it can be used to complete a map using different pics of aerial photos

[Reply](#)



Adrian Rosebrock

January 11, 2016 at 3:04 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-386452>)

Certainly! Provided that there are enough keypoints matched between each photos, you can absolutely use it for aerial images.

[Reply](#)



Waladi (<http://www.aerialview.co.id>)

January 25, 2016 at 11:42 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-388026>)

Great topic Adrian.

As for stitching Aerial View imagery – mainly for mapping purpose, can it be used to stitch second image that “overlap” on below part of the first image (not the left-to-right, but bottom-to-upper part of image) ?

[Reply](#)



Adrian Rosebrock

January 26, 2016 at 5:57 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-388123>)

Yes, you can use it to stitch bottom-to-top images as well, but you'll need to change **Lines 31-33** to handle allocating an image that is tall rather than wide and then update the array slices to stack the images on top of each other. But again, yes, it's totally possible.

[Reply](#)



Jakob Kirkegaard (<http://percepto.dk>)

February 11, 2016 at 4:09 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-389822>)

Unfortunately, the stitcher functionality in OpenCV 3.1 or more precisely the HomographyBasedEstimator and the BunderAdjuster used by the Stitcher class will only estimate camera rotations (no camera translations), i.e. it assumes that all camera centers are approximately equal.

If the camera experience translations (like aerial shots) or translations in general, the obtained results are usually not that great – even though the images can be matched given good keypoints.

[Reply](#)



Manuel

August 5, 2016 at 1:16 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-403587>)

Hi Jakob, could you please point me out how what approach could I follow to handle the no-camera-translations problem?

Thanks



Rein van den Boomgaard

December 17, 2018 at 12:06 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-492702>)

Not always! Image stitching works in two situations:

1/ Camera is fixed in position and only allowed to rotate around the optical center, or

2/ Camera is looking at a flat plane in 3D space. Then the position and orientation of the camera are not important.

For aerial photographs the second situation is approximately true in case the distance away

from the camera is large compared to the sizes of the objects on the ground.

(see

<https://staff.fnwi.uva.nl/r.vandenboomgaard/PCV20172018/LectureNotes/CV/PinholeCamera/Projectivities.html>

(<https://staff.fnwi.uva.nl/r.vandenboomgaard/PCV20172018/LectureNotes/CV/PinholeCamera/Projectivities.html>)

BTW this tutorial makes one of the exercises in a course i am teaching dead easy...

[Reply](#)



Ankit Dixit

[January 12, 2016 at 1:00 am](#) (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-386520>)

Hi adrian,

A very good topic you have covered in this post, thanks for the description, i have a question regarding an OCR problem, i have first version of your book where you have described digit recognition using HOG features, that algorithm works on contour detection (blob based), my question is what may be the other way to approach the problem where i cant get individual contours for each digit or character (Segmentation is not possible), thanks for your suggestion in advance. and a big thank you for writing a very easy to understand book.

[Reply](#)



Adrian Rosebrock

[January 12, 2016 at 6:25 am](#) (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-386553>)

Are you referring to cursive handwriting where the characters are not individually segmentable? If so, I think R-CNNs are a likely candidate if you have enough training data. Otherwise, you might want to look at the [**Street View House Numbers Dataset**](#)
[\(http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html\)](http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html)

and the relevant papers associated with high accuracy results.

[Reply](#)

**Ankit Dixit**

January 13, 2016 at 5:12 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-386671>)

Hi Adrian thanks for your reply, actually i am working something like vehicle registration data extraction through registration card into a json file, where there are some fixed fields like name and address and their respective variables. i have captured a few images of registration card from a mobile camera so scale varies a lot and in some cases minor orientation changes also there, a big advantage here is there is no hand written letters or digits so variability of data is less, and all alphabets are in upper case, but at the time of segmentation(image Thresholding) some letters got merged in a single blob (or contour) so i cant extract each letter individually. so i cant apply blob based analysis, i have tried few pre-processing steps to separate the blobs but it results in some useful structural information loss, what should i do here.

[Reply](#)

**Adrian Rosebrock**

January 13, 2016 at 6:29 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-386679>)

I would suggest sending me an email so we can chat more offline about it. This post is about panorama construction, not digit extraction, so it would be great if we could keep the comments thread on topic.

[Reply](#)

**Sean McLeod**

January 12, 2016 at 4:31 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-386538>)

Great post. I've been wanting to try use OpenCV for orthophoto stitching of aerial photos from

~~Great post! I've been wanting to try out OpenCV for panoramic stitching or aerial photos from drones.~~

In terms of the seam that's surely to do with different exposures and not focusing. Typically for landscape photos the focus will be on infinity anyway. Some of the panoramic software I've used in the past has a feature to try and equalize the exposures so that the seam isn't visible. Sounds like potentially a topic for another OpenCV blog post 😊

[Reply](#)



Adrian Rosebrock

[January 12, 2016 at 6:22 am](https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-386551) (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-386551>)

Great point Sean! I ran a few tests just using images around the apartment captured with my iPhone with a fixed exposure. This got rid of the seam. But of course, it requires that your environment doesn't change dramatically through the Panorama.

[Reply](#)



Bruno

[January 17, 2016 at 6:00 pm](https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-387128) (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-387128>)

An alternative is to apply a blending mask along the seam, this way you avoid having to adjust exposure between images. If the difference in exposure is small between the neighbouring images, it hides the seam nicely.

[Reply](#)



Adrian Rosebrock

[January 18, 2016 at 3:22 pm](https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-387239) (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-387239>)

Great point Bruno! Perhaps that will be a subject for a future blog post 😊

[Reply](#)



tonyv

January 20, 2016 at 8:56 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-387424>)

Why, in stitcher.stitch(), line 15 is (imageB, imageA) = images
when it is called with stitcher.stitch([imageA, imageB]). i.e reversed?

I note that showMatches displays the left and right images exchanged, and correcting line 15 fixes that, but renders the main panorama incorrectly.

Any comments?

[Reply](#)



Adrian Rosebrock

January 20, 2016 at 1:41 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-387448>)

It's more intuitive for us to think of a panorama being rendered from left-to-right (which is an assumption this code makes). We reverse the unpacking in the `stitch` method for the actual matching of the keypoint detection and local invariant descriptors. Again, order *does* matter when it comes to the stitching of the images.

[Reply](#)



VIJAYA KUMAR

February 5, 2016 at 10:21 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-389155>)

hi adrian i'm vijay i've configured the opencv and python in windows and i've some doubt....!
still to do some project does we need the any ide like eclipses? i've just stucked in here
will you please help me.....
thanks in advance.....

[Reply](#)



Adrian Rosebrock

[February 6, 2016 at 10:00 am \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-389249>\)](#)

Hey Vijay, I honestly haven't used a Windows system in 9+ years and I don't do development with Eclipse, so I'm not the right person to ask about this. Best of luck with your project!

[Reply](#)



Fadil

[February 23, 2016 at 9:54 am \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-390727>\)](#)

Hey Adrian,

First thanks for your blogpost, really well explained! As for my problem, I am trying to make your code work for three images and I can't seem to obtain a good result (the right side of the image gets deformed and pixelized).

I did not get to creative I just conduct the exact same operation you did on 2 pictures. First I stitch picture A and B(call the result R1), then picture B and C (R2)and finally I stitch R1 and R2. You said you'll post how to do panorama with +2 images, do you have a better algorithm in mind ?

[Reply](#)



Adrian Rosebrock

[February 23, 2016 at 3:20 pm \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-390750>\)](#)

Constructing panoramas > 2 images is a substantially harder problem. I would suggest [**giving this thread a read \(<http://stackoverflow.com/questions/24563173/stitch-multiple-images-using-opencv-python>\)**](#) for a more detailed discussion on the problem.

[Reply](#)



Massimo

March 9, 2016 at 9:40 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-391694>)

Hi Adrian,

great post!

I was wondering if you can help me with this:

<http://nbviewer.jupyter.org/gist/anonymous/443728eef41cca0648f1>
[\(http://nbviewer.jupyter.org/gist/anonymous/443728eef41cca0648f1\)](http://nbviewer.jupyter.org/gist/anonymous/443728eef41cca0648f1)

My images has a black border (results from camera calibration)

The mosaic works great, but I'm struggling in understanding how to get rid of the black borders.

Have you any clue?

Thanks!!!

Massimo.

[Reply](#)



Adrian Rosebrock

March 10, 2016 at 12:04 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-391747>)

When you say "get rid of" the black borders, do you mean simply setting the border to white? If so, first create the result image using `np.ones` and fill them with (255, 255, 255) (i.e., white) rather than 0 (black).

[Reply](#)



Massimo



MASSIMO

March 13, 2016 at 4:17 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-391889>)

Hi Adrian,

both original images (A,B) have black pixels at the border (before stitching them together)

See: <http://epinux.com/panorama.png> (<http://epinux.com/panorama.png>)

The resulting mosaic has the black border (from image A) overlaying pixels with data from image B.

What I'm looking for is a way to set some sort of transparency
(or perhaps set to null? the black borders)
see the image labeled "desired result" at the link above.

[Reply](#)



Adrian Rosebrock

March 13, 2016 at 10:17 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-391908>)

Keep in mind that an image is always rectangular. You can't set the pixels to "null" since they are part of the image matrix. You can change their actual color (such as making them black or white), but you can't "remove" the pixels from the image.

[Reply](#)



Massimo

March 13, 2016 at 12:56 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-391928>)

Sorry, what I meant is: do you know a way to add an alpha channel to the resulting image ?

If possible, I can then try to set the transparency to 100% where the black borders are.



Adrian Rosebrock

March 14, 2016 at 3:31 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-392013>)

I would suggest creating an alpha channel that has the same shape as the other channels and then merging all four together:

```
alpha = np.zeros(image.shape[:2], dtype="uint8")
# ... set background pixels to 100% transparent
image = cv2.merge([B, G, R, alpha])
```

I haven't tested this myself, but it should work.



Koshy George

March 11, 2016 at 4:47 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-391801>)

Dear Adrian,

A brilliant tutorial explained with such simplicity. It really helped me understand and appreciate the Python-OpenCV combo. Thank you

[Reply](#)



Adrian Rosebrock

March 13, 2016 at 10:27 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-391916>)

No problem, I'm happy the tutorial could help Koshy! 😊

[Reply](#)



Vanessa

April 17, 2016 at 5:58 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-395108>)

Hi Adrian,

Is it possible to save the output image without the black excess? I know it's because it computed based on the width of the two images, is there a way to save the image without it? If so, how do i do this? Thank You^

[Reply](#)



Adrian Rosebrock

April 17, 2016 at 3:28 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-395123>)

I personally haven't done this, but yes, it is possible. You basically need to find where the black excess is surrounding the image. A hacky way to do this would be to apply thresholding and find the contour of the image itself. A better approach would be to examine the homography/warping matrix and figure out the coordinates of where the valid stitched image is. This will result in some cropping of actual image content, but will remove the black excess. I'll try to do a blog post on this topic in the future.

[Reply](#)



Anoop Hallur

October 13, 2016 at 9:15 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-408148>)

Another approach is to use Content Aware Fill with Patch Match Algorithm.

<https://research.adobe.com/project/content-aware-fill/>
[\(https://research.adobe.com/project/content-aware-fill/\)](https://research.adobe.com/project/content-aware-fill/)

This is not a trivial solution though.

[Reply](#)



Jonjo

May 3, 2016 at 2:21 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-396549>)

When are you going to release a blog about stitching multiple images? You mentioned in this article that you would do it soon.

[Reply](#)



Adrian Rosebrock

May 3, 2016 at 5:41 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-396557>)

I would love to do it, but I simply haven't had enough time as I've been working on other projects. I honestly cannot commit to a timeframe for the stitching multiple images.

[Reply](#)



Kyle

May 6, 2016 at 4:07 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-396895>)

This is awesome. If it's trivial, how could I change the code in order to only take the intersection of the two images?

Thanks

[Reply](#)



Adrian Rosebrock

May 6, 2016 at 4:29 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-396899>)

Hey Kyle — what do you mean by "intersection of the two images"? Could you please clarify?

[Reply](#)



Kyle

May 6, 2016 at 5:20 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-396906>)

By intersection I mean that I only want the parts image portion that is present in both images. I tried using bitwise_and (with the result of your cv2.warpPerspective in the stitch function and with the other image from the panorama), but the image it outputs has its colors all messed up. I don't really know how to create a mask correctly in order to achieve what I want.

[Reply](#)



Adrian Rosebrock

May 7, 2016 at 12:38 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-396979>)

Instead of creating a mask, the best option is to explore the (x, y) -coordinates of the matched feature vectors. Find the (x, y) -coordinates of the **matched** keypoints that correspond to the top-left, top-right, bottom-right, and bottom-left corners. From there, you can crop out the overlapping ROI.

[Reply](#)



Wayne Anderson

May 9, 2016 at 4:27 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-397094>)

Hello, really enjoying your tutorials but I've run into a little snag. I have been trying to get this one to work but I keep getting a typeerror: Nonetype for this line: (result, vis) = stitcher.stitch([imageA, imageB], showMatches=True). I even downloaded the source files too

and they don't seem to work. Pretty new to python so I am not sure why this is happening. Any ideas?

[Reply](#)

**Adrian Rosebrock**

May 9, 2016 at 6:58 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-397173>)

The `stitch` method will return `None` if not enough keypoints are matched to compute the homography. What images are you using? Are they the same images as in this post or ones of your own? If they are ones of your own, then there are not enough matched keypoints to stitch the images together. In that case, I would try different combinations of keypoint detectors and feature descriptors. It could also be the case that the images simply cannot be stitched together.

[Reply](#)

**Wayne Anderson**

May 10, 2016 at 11:06 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-397273>)

Ah okay, that's good to know! I guess that the images don't have enough keypoints then. And I did get the posts pictures to work, I was using two of the left images instead of a left and right... Looks like it was mainly human error. Thanks for the help and information though!

[Reply](#)

**Maggie**

August 24, 2018 at 2:35 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-475436>)

Hi Adrian,

I am having the same problem as Wayne, where nothing is being displayed, however I am not even getting an error message. I tried running with photos from this post and still nothing showed up. Using print statements I've realized the problem originates in line 61. It seems like the code just stops after that line because any print statement after line 61 is not displayed.

I'd really appreciate your help.

[Reply](#)

**Adrian Rosebrock**

August 30, 2018 at 9:40 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-475998>)

Hey Maggie — see my original reply to Wayne. The `stitch` method is returning `None`. When this happens not enough keypoints are matched between the two images. If not enough keypoints are matched then you cannot stitch the images together.

[Reply](#)

**Mr. Pink**

May 9, 2016 at 3:21 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-397155>)

What do you think would have to be done to make the stitching not care whether you gave the left photo first or the right photo first?

[Reply](#)

**Adrian Rosebrock**

May 9, 2016 at 6:46 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-397168>)

This becomes very problematic because at least one of the images needs to act as a “reference point”. This is why panorama apps normally tend to make sure you move from left-to-right or down-to-up. Without setting an initial reference point, you have to resort to heuristics, which often fail.

[Reply](#)

**Mr. Pink**

May 10, 2016 at 11:09 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-397274>)

So I've been looking at this for a bit now, and have managed to find something that you may be interested in.

I plotted all the matched points (called "matches" from line 30 in your code) of when images are inputted left to right and when inputted right to left. I found out that these points make roughly a line, and it is possible to calculate the slope of such a line.

The slope of the left to right instance should always be smaller than the right to left instance. Therefore it's possible to throw in a simple if statement and make a swap of variables.

I have tested this theory with many images, and it seems to work very well. I have a couple others in my code for special cases (like being equal for example). Just thought you or some reader would be interested to know.

[Reply](#)



Adrian Rosebrock

May 10, 2016 at 6:20 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-397294>)

Thanks for sharing, and great investigative work! This is certainly an interesting heuristic — however, it's a heuristic that is easily broken. For certain situations though, this might work well enough.

[Reply](#)



najib

June 8, 2016 at 4:30 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-399883>)

you can do the same program in c ++

[Reply](#)



Amey (<http://none>)

June 15, 2016 at 6:23 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-400273>)

Hello Adrian

Thanks for a wonderful post on image stitching. This is kind of faster than C++ example I had tried earlier.

I had a query though on the topic. How can you stitch 3 or 4 images to get one panorama image? What changes are needed in this code?

Regards

Amey

[Reply](#)



Adrian Rosebrock

[June 15, 2016 at 12:27 pm \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-400293>\)](#)

Stitching > 2 images together is *substantially harder* than stitching 2 images together. The code has to change dramatically. I will have to write a separate blog post on this, but I'm honestly not sure when I'll be able to.

[Reply](#)



thenuka

[June 22, 2016 at 4:00 am \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-400751>\)](#)

How to remove the black portion in stitching if you are going to stitch about 10 images together when kinda creating an aerial map

[Reply](#)



Abdulkader



Adrian Rosebrock

July 23, 2016 at 4:59 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-402844>)

Hi Adrian, I'm so grateful for that brilliant Tutorial ... but I have a problem !
when I try to run the code from Terminal nothing will be shown on screen although it gives NO
error and –first/-second parameters are set perfectly .. I have the the latest version of OpenCV,
is there any suggestion to help me, please my project should be finished in less than a week ...
wainting for ur reply ... thanks again 😊

[Reply](#)



Adrian Rosebrock

July 27, 2016 at 2:48 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-403121>)

If nothing is getting displayed to your screen, then it sounds like the homography matrix isn't being computed (in which case there are not enough keypoint matches). I would suggest inserting some print statements to help you debug where the code is existing.

[Reply](#)



Sri

July 25, 2016 at 4:29 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-402980>)

Awesome Mr. Adrian. Love you blog.

I have a question. I am trying to run this program. Getting this error line 33
result[0:imageB.shape[0],0:imageB.shape[1]] = imageB

ValueError: could not broadcast input array from shape (302,400,3) into shape (300,400,3)

[Reply](#)



Adrian Rosebrock

July 27, 2016 at 2:30 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-403107>)

Hey Sri — based on the error message, it looks like `imageB` has a different height than the result image. Try changing the code to this and see if it helps:

```
result = cv2.warpPerspective(imageA, H,
    (imageA.shape[1] + imageB.shape[1], max(imageA.shape[0], imageB.shape[0]))
```

[Reply](#)



Manuel

August 4, 2016 at 8:25 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-403581>)

Hi Adrian,

You are calling `stitch.py` with the left image as first argument, that becomes `imageA`, and you warp it with `warpPerspective`. But what you say in the text is that `warpPerspective` gets as input the image we want to warp: the right image.

So I don't know what I'm missing here.

Thanks.

PD: I'm going send you my previous comment tomorrow.

[Reply](#)



Ahmad

August 18, 2016 at 8:06 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-404538>)

Very informative post. Thanks for it. I just wanted to know can you direct me to some post which is about spherical stitching. Actually i have 7 images from gopros (Including top and bottom) I want to create a spherical panorama from it. Any idea how that can be done.

[Reply](#)

**Adrian Rosebrock**

[August 18, 2016 at 9:24 am \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-404543>\)](#)

I don't know of any tutorials related to spherical/fish eye image stitching, but if I come across any, I'll be sure to post them here.

[Reply](#)

**Ahmad**

[August 29, 2016 at 11:48 am \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-405113>\)](#)

Thanks Adrian. That would be of great help.

[Reply](#)

**Tirumaleswararao Gudivada**

[March 1, 2017 at 3:12 pm \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-418721>\)](#)

Hi Adrian,

Great blog. Thank you for your valuable time. I learned a lot.

I am also looking for algos to stitch fisheye images.

It would be great if you can share any info you find relevant to this.

Thanks,

Tiru

[Reply](#)

**Felipe**



September 24, 2016 at 1:01 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-406639>)

Hi Adrian, excellent work with this example.

I'm trying to implement image stitching from a video using this example , but I cannot make it work, none of images retrieves from video have successfully stitched together. I've searched a lot of examples using OpenCV with Java, C# and Python. Please if you can send me some highlights to accomplish image stitching from a video it would be great!! thanks in advance 😊

[Reply](#)



Adrian Rosebrock

September 27, 2016 at 8:51 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-406821>)

Have you tried using [my example on image stitching from video](#)

(<https://www.pyimagesearch.com/2016/01/25/real-time-panorama-and-image-stitching-with-opencv/>)?

[Reply](#)



Felipe

October 1, 2016 at 12:04 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-407244>)

Thank for your quickly response! 😊 I'm trying to do something like this:

<https://www.youtube.com/watch?v=93jOLIObfuE> (<https://www.youtube.com/watch?v=93jOLIObfuE>)

[Reply](#)



Adrian Rosebrock

October 2, 2016 at 8:59 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-407311>)

Thanks for sharing. I'll certainly use that video as inspiration when I create the future tutorial on > 2 image/frame stitching.

[Reply](#)

**Felipe**

[October 3, 2016 at 11:25 am](#) (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-407401>)

Awesome! I'll be waiting for your tutorial to test it! thanks again! 😊

**Mezher**

[April 10, 2018 at 10:53 am](#) (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-455919>)

helle Felipe , i am working on a project where i shoud have something like the video you shared.

Do you have some helpfull information ?

[Reply](#)

**Devin Willis**

[October 6, 2016 at 9:59 am](#) (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-407621>)

Hey Adrian,

I was wondering if it would be possible to take multiple images of a slide and stitch these together to create a very high resolution image of the whole slide. If so I was wondering why when I run the code it says segmentation fault (core dumped)

[Reply](#)

**Adrian Rosebrock**

[October 7, 2016 at 7:38 am](#) (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-407705>)

You're likely trying to stitch together very large images and thus detecting TON of keypoints

and local invariant descriptors (hence the memory error). Instead, load your large images into memory, but then resize them to be a maximum of 600 pixels along their largest dimension. Perform keypoint detection and matching on the smaller images (we rarely process images larger than 600 pixels along their maximum dimension).

Once you have your homography matrix, apply it to your original high resolution images.

[Reply](#)



Glenn

[January 7, 2017 at 9:42 pm](https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-415370) (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-415370>)

I am getting this error as well. It happens even when I use smaller images. It seems to be related to this line:

kps = detector.detect(gray)

[Reply](#)



Adrian Rosebrock

[January 9, 2017 at 9:15 am](https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-415475) (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-415475>)

How small are your “smaller” images in terms of width and height? Also, which version of OpenCV are you using?

[Reply](#)



Lorena

[March 31, 2017 at 3:49 am](https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-421587) (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-421587>)

hello felipe you can see your code.
thank you

[Reply](#)

**Chris**

October 6, 2016 at 3:04 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-407644>)

Great Tutorial this is awesome! I had an interesting idea. I was wondering if it was possible to combine photos of different sizes based on your code. As a project I want to use a video to create a panorama. Using OpenCV to parse through the frames I would stitch one photo to the combined strip. This would mean that the left/first photo would be a lot wider than the right/second photo. Just looking for your opinion. Also your book is great I have been using it for my research.

[Reply](#)

**Adrian Rosebrock**

October 7, 2016 at 7:30 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-407700>)

Your images certainly don't have to be the same size or from the same camera sensor to be stitched together provided enough keypoints match; however, I think you're going to get mixed results that may not be aesthetically pleasing. If one image has a different exposure than the other then you'll need to correct the final image by applying image blending.

[Reply](#)

**sam (<http://sachinckhai>)**

October 21, 2016 at 4:47 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-408645>)

Hay Andrian,

Please post the multiple image stitching it would be great please...

[Reply](#)



Metalim

November 1, 2016 at 12:16 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-409596>)

hi Adrian,

How can I evaluate quantitatively different feature descriptors and extractors performance on the same image pair? Which parameters can tell me about the accuracy of feature matching using different descriptors and extractors? Is Lowe ratio or repError doing that also?

[Reply](#)



Adrian Rosebrock

November 3, 2016 at 9:56 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-409760>)

If I understand your question right you are trying to determine which set of keypoint detectors and local invariant descriptors perform best for a given set of image? The Lowe ratio test is used to avoid false positive matches. On the other hand, the RANSAC method is used to actually spatially verify these matches.

[Reply](#)



Chris

November 17, 2016 at 3:40 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-411001>)

Hi Adrian,

I was wondering if you happened to do the other blog post where you stitched multiple images together? I modified ur code from this example to linearly stitch images but am struggling to

find a way to stitch images regardless of orientation. Any help is appreciated.

Also If you know of any tutorials on how to build the opencv_conrib modules for opencv 3 on windows that would be a god send.

Thanks for your help,

Cheers.

[Reply](#)



Adrian Rosebrock

November 18, 2016 at 8:53 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-411031>)

Hey Chris, I haven't had a chance to write up another blog post on image stitching. Most of my current posts have been related to deep learning and updated install tutorials for Ubuntu and Mac (I don't support Windows on this blog). I'll try to circle back to the image stitching post, but I honestly can't say when that might be.

[Reply](#)



Chris

November 20, 2016 at 1:31 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-411230>)

Thanks for getting back to me:)

Cheers,

[Reply](#)



Chris

November 20, 2016 at 2:05 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-411235>)

Hi Adrian,

Sorry for all the questions. I was curious as to why you used the imutils package to resize your images instead of the built in resize function in OpenCV? are there any advantages or tradeoffs?

Thanks in advance.

Cheers,

[Reply](#)



Adrian Rosebrock

November 21, 2016 at 12:30 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-411300>)

My `imutils.resize` function automatically takes into account the aspect ratio of the image whereas the `cv2.resize` function does not.

[Reply](#)



Hugo

November 30, 2016 at 12:42 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-412286>)

Hi Adrian!

First of all, thanks A LOT for that and all the other contents I've been using a lot since I got an interest in image processing with Python!

I enjoy a lot both the quality and the pedagogy of your guides & solutions 😊

I have a question about this one I can't find the answer of on my own: I'm trying to get a "reduction factor" of the homographies I compute.

Which would mean something like: if a segment measures 10 pixels before warping, how long is it after warping. Because of deformation, there's no unique value but I guess it could be

possible to have the value range?

I was thinking maybe calculate the total distance between all matching key points on image A then the total distance between all matching key points on image B and calculate the ratio of those 2 values? If there's enough and well-reparted matching points that should give me an average reduction factor shouldn't it?

Unless there's something built-in and more reliable with a cv2 function?

Thanks again for everything 😊

[Reply](#)



Adrian Rosebrock

[December 1, 2016 at 7:30 am \(https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-412371\)](https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-412371)

Hmmm, this is a good question. I'm not sure there is a built-in OpenCV function to do this. By definition applying a perspective transform is going to cause the images to be warped. I guess my question is what are you trying to accomplish by computing this value?

[Reply](#)



Julio

[December 14, 2016 at 12:35 pm \(https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-413522\)](https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-413522)

Hello, could you explain how to use SURF with RANSAC but without using the cv2.findHomography () function because I want to use the cv2.getaffinetransform

Thanks

[Reply](#)



Christopher Daniel

[December 20, 2016 at 11:14 am \(https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-413522\)](https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-413522)

[stitching/#comment-413929](#)

hello,

I think we don't have to use imutils I executed code without imutils and it works fine and quality is also good compared to imutils substituted input image

[Reply](#)



Adrian Rosebrock

[December 21, 2016 at 10:23 am \(\)](#)

In this example `imutils` is used to (1) determine your OpenCV version and (2) resize your image. You can modify this code to *not* use `imutils`, but it is highly recommended.

[Reply](#)



Christopher Daniel

[December 25, 2016 at 2:41 am \(\)](#)

thank you if images are resized only by width I got broadcast error. So I removed the lines
imageA = imutils.resize(imageA, width=400)
imageB = imutils.resize(imageB, width=400)
and worked fine

then I tried

imageA = imutils.resize(imageA, width=400,height=350)
imageB = imutils.resize(imageB, width=400,height=350) and worked fine

it worked fine. May be you can add this change to your tutorial

[Reply](#)



Chris

January 15, 2017 at 1:36 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-415842>)

Thanks again for all your effort into building this OpenCV community. I had a question regarding the photos themselves. Your implementation requires a certain order for the images to be piped into your program. To my understanding this is due to the warp perspective and the way you link the images from line 31-33 in panorama.py. I was wondering if there is a way to modify that section of code to allow for any order of images to be passed through (I've been trying my own thing to no avail)?

Thanks for any help in advance.

Chris

[Reply](#)



Adrian Rosebrock

January 16, 2017 at 8:13 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-415876>)

You are correct, we assume a specific order passed to the function — this is by far the easiest method. For multiple images you actually do keypoint matching on all the frames and then define a “path finding” algorithm that can order the frames. But for two images, I think it’s much easier to define a function that expects the images in a specific order.

[Reply](#)



Chris

January 16, 2017 at 2:34 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-415897>)

Do you know where I might be able to find out more information regarding these “path finding” algorithms to allow me to input an unordered image set?

Again thank you for all your help.

[Reply](#)



Adrian Rosebrock

[January 17, 2017 at 8:50 am \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-415945>\)](#)

Take a look at Dijkstra's algorithm and dynamic programming to start. I don't know of any examples off the top of my head that implement this explicitly for image stitching. But again, this will (ideally) be a topic that I'll cover in a future PyImageSearch post, I'm just not sure when.

[Reply](#)



Saswati

[January 30, 2017 at 7:02 am \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-416739>\)](#)

Hi,

I am new to opencv and image processing. Please anybody help me to solve this error. I am even unable to figure out its meaning.

usage: stitch.py [-h] -f FIRST -s SECOND

stitch.py: error: argument -f/-first is required

[Reply](#)



Adrian Rosebrock

[January 30, 2017 at 4:19 pm \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-416788>\)](#)

You are not supplying the image paths via command line argument. Please take the time to [brush up on command line arguments](#)

(<https://www.pyimagesearch.com/2018/03/12/python-argparse-command-line-arguments/>) before continuing.

[Reply](#)



Judy

February 1, 2017 at 6:45 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-416881>)

Would it be possible to create a solution with this that stitches stereo video? Working on a project for school and looking for some insight. Thanks.

[Reply](#)



Adrian Rosebrock

February 1, 2017 at 12:45 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-416895>)

Yes, absolutely. In fact, I've already done a [blog post](#) (<https://www.pyimagesearch.com/2016/01/25/real-time-panorama-and-image-stitching-with-opencv/>) on the topic.

[Reply](#)



Judy

February 3, 2017 at 3:14 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-417031>)

Thank you for your response! I actually did find that blog post later, and you'll notice that I made a comment on there as well, I'm curious about how one might go about streaming the stitched video to something like a VR headset, or just streaming it in general. Just looking for some ideas.

Thanks again!

[Reply](#)



Adrian Rosebrock

February 4, 2017 at 9:26 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-417060>)

I don't do any work with VR headsets; however, there are a number of different streaming protocols. I would suggest doing research on your particular headset and see if it's possible to stream the frames from the headset itself. Regarding streaming in general, I plan to cover that in a future blog post, although I'm not entirely sure when that will be.

[Reply](#)



JJolles (<http://jollejolles.com>)

February 6, 2017 at 9:56 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-417120>)

Thanks Adrian for a very clear tutorial. I was wondering if you could help me a bit further how to stitch four images together (2x2 grid), or guide me in the right direction. Basically I have four cameras in a grid filming a large space and which overlap by 1/5th of the space on the sides and top/bottom. I got your script working on two side-by-side images, but how could I adapt your script to stitch all four images together? Is there an easy way to adapt your script by first stitching the top 2 and then the bottom 2 and then stitching those new top and bottom images together? Any help is appreciated. I will then actually try and use the stiching parameters to use in stitching frames from the video together.

Thanks again!

[Reply](#)



Adrian Rosebrock

February 7, 2017 at 9:11 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-417199>)

Are these cameras fixed and non-moving? If so it might be easier to simply calibrate the cameras and perform projection that way.

[Reply](#)



gwhiz

February 14, 2017 at 2:03 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-417676>)

Hi Adrian,

We are also looking for a way to stitch video from multiple cameras together. IN our arrangement, the cameras are pointing from the edge of the space towards the center of the space, the opposite from most rigs today. This arrangement is basically bullet time. If we align our cameras to have the same centers (for example put one on each of four walls in a room at the same x/y on each wall) can we stitch the video together?

[Reply](#)



JJolles

February 20, 2017 at 12:02 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-418129>)

Thanks, I hadn't seen you reply hence the delay. Yes the cameras are fixed and non-moving. I calibrated them to account for any distortion of the lenses. But then using your script on just the top two cameras it does warp the right camera based on the left. I would like all four to be stitched with all having the ability to be warped. Do you have any further tips, also what you mean regarding calibrating the cameras? I appreciate it!

[Reply](#)



Adrian Rosebrock

February 22, 2017 at 1:45 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-418255>)

I don't have any tutorials on camera calibration here on the PyImageSearch blog, but I would suggest [starting here \(\[https://opencv-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_table_of_contents_calib3d/py_table_of_contents_calib3d.html#py-table-of-content-calib\]\(https://opencv-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_table_of_contents_calib3d/py_table_of_contents_calib3d.html#py-table-of-content-calib\)\)](https://opencv-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_table_of_contents_calib3d/py_table_of_contents_calib3d.html#py-table-of-content-calib).

[Reply](#)

**Nilavro**

February 10, 2017 at 5:34 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-417494>)

Hi Adrain, thanks for the code and guide.

I was wondering how may I perform a cylindrical/inverse cylindrical projection before of the candidate images to be stitched together. This will help me to stitch multiple images taken from a rotating base.

Thanks again.

[Reply](#)

**Adrian Rosebrock**

February 13, 2017 at 1:53 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-417658>)

Absolutely. I will try to cover cylindrical projections in a future blog post.

[Reply](#)

**Daniel**

March 8, 2017 at 10:20 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-419784>)

Hi Adrian,

Thanks for your sharing! Now the output panorama is left photo overlaps right photo. I am wondering which part of code should I change to make right photo overlaps left photo?

Thanks!

[Reply](#)

**Adrian Rosebrock**

March 10, 2017 at 3:55 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-419915>)

You would want to swap the left and right images on Lines 31-33.

[Reply](#)

**Mathew Orman (<http://innovation.ag>)**

March 16, 2017 at 10:19 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-420484>)

Does not work on Raspberry Pi 3 but it works on PC windows 10

Any ideas?

[Reply](#)

**Mathew Orman (<http://innovation.ag>)**

March 16, 2017 at 10:58 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-420486>)

AT first it complained about an extra argument showMatches in stitch call.

Then I removed it and now it say “Segmentation fault” on 320 x 240 images...

I use python 2.7...

[Reply](#)

**Adrian Rosebrock**

March 17, 2017 at 6:43 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-420549>)

The segmentation fault is helpful, it's likely that this is a local invariant descriptor issue. I would start to debug the problem by inserting `print` statements throughout your code until you can narrow down *exactly* which line is causing the seg-fault.

Also, make sure you use the “Downloads” section of the post to download the code (if you haven’t done so already) rather than copying and pasting. This will ensure you are using the same codebase and project structure.

[Reply](#)



Mathew Orman (<http://innovation.ag>)

March 16, 2017 at 11:13 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-420488>)

I am using Python v. 2.7 and cv2 v. 2.4.9.1

[Reply](#)



Carol

April 8, 2017 at 12:02 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-422247>)

Hey Adrian,

I've see your tutorial page that shows how to install OpenCV 3 with `opencv_contrib` support enabled, but I didn't see the way for windows, can you please upload one?

Thank you so much!

[Reply](#)



Adrian Rosebrock

April 8, 2017 at 12:33 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-422248>)

Hi Carol — I actually don't have any Windows install tutorials. Sorry about that! When it comes to computer vision and OpenCV, I *highly* recommend that you use a Unix-based environment.

[Reply](#)



Enkhbold Munkhbold

April 11, 2017 at 4:25 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-422512>)

Hello Adrian,

I'm having this error message:

```
usage: stitch.py [-h] -f FIRST -s SECOND
stitch.py: error: argument -f/-first is required
```

When i try and run the stitch.py

I extracted the files to a folder and just ran it on Python2.7.13

Did i do something wrong ? it's not working.

[Reply](#)



Adrian Rosebrock

April 12, 2017 at 1:08 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-422625>)

You need to read up on [command line arguments](#)

(<https://www.pyimagesearch.com/2018/03/12/python-argparse-command-line-arguments/>) before you continue.

[Reply](#)



Enkhbold Munkhbold

April 21, 2017 at 2:14 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-423329>)

I still don't quite get it.
Can you tell me what to do step by step after downloading the zip file?
When and if you get a chance. It would be really helpful.
I personally do not know much of python since i was taught on Java and a little bit of C++.

I got on the command line and went to the file (i put it on my desktop and did "cd desktop => panorama-stitching" and tried to run stitch.py) and had that trouble.
and doing "stitch.py bryce_left_02.png bryce_right_02.png" had the same result.
Thanks.

[Reply](#)



Adrian Rosebrock

April 21, 2017 at 10:48 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-423368>)

Hi Enkhbold — while I'm happy to help point readers in the right direction, the PyImageSearch blog does assume you've had some basic programming experience and are comfortable working with the command line. I can't provide step-by-step instructions on how to execute a Python script. I would recommend you go through [my list of recommended Python resources \(<https://simpleprogrammer.com/2017/02/15/get-started-learning-python/>\)](#) to help you learn the basics first.

[Reply](#)



Enkhbold Munkhbold

April 28, 2017 at 1:37 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-423950>)

Was able to make it work (found the comment on top of the stitch.py) i was trying "python stitch.py –first images/bryce_left_01.png \ –second images/bryce_right_01.png" from the picture above and what made it not work was that backslash.
Now i want to stitch multiple images can you provide the page you talked about that if

you made it ?

Me and my teacher are trying to make a program that constantly reads images from a video and saves it into a big panoramic picture to make it easier for watchers to see what was on a few secs or way back whenever they want.(the idea came from MIT lectures

and because the video can only show so much, or it would make things impossible to read since its too small)

So again, Is there a multiple image stitching method you made that i can take a look at ?

**Adrian Rosebrock**

April 28, 2017 at 9:23 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-423987>)

Hi Enkhbold — I have not written a tutorial on stitching multiple images together yet. I'm not sure when I will get to it, but I will try to cover it in the future.

**Daniella Solomon**

April 17, 2017 at 7:13 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-423011>)

Hi,

There is a reason that you sending in stitcher line 55 the RGB image and not the gray one?
(OpenCV3 option)

attached:

(kps, features) = descriptor.detectAndCompute(image, None)

instead of

(kps, features) = descriptor.detectAndCompute(gray, None)

I'm trying to do something similar with videos, I have two cameras, one PTZ and one wide and I'm drawing a rectangular on the wide one of what the PTZ is showing and it's really slow, I tried to use threads but still not close to real time. any suggestion?

[Reply](#)

**Adrian Rosebrock**

April 17, 2017 at 10:50 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-423020>)

Thank you for pointing out this typo — we typically detect keypoints and extract local invariant descriptors from `grayscale` rather than `multi-channel` images. The `gray` image should be used

~~descriptors from gray-scale rather than multi-channel images. The gray image should be used instead, but in most cases you won't notice any changes in performance.~~

[Reply](#)



Esraa Elbasha

April 28, 2017 at 11:20 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-424027>)

I want to control a servo depending on the overlap between two pictures .. is no overlap I will rotate the servo till there is .. I tried to push two pictures of different places and it still finds keypoints .. how can I prevent that or at least increase the accuracy

[Reply](#)



Esraa Elbasha

April 29, 2017 at 1:51 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-424080>)

is there anyway to know the overlap % between two pictures !?

[Reply](#)



Adrian Rosebrock

May 1, 2017 at 1:43 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-424217>)

Yes, absolutely. A good way to approximate the overlap is to match the keypoints. Once you have the (x, y)-coordinates of the matched keypoints in both images you can determine how much of image #1 overlaps with image #2.

[Reply](#)

**Mohsen khoshnazar**

May 21, 2017 at 1:34 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-425683>)

Hello Adrian and thanks for your post.

Is that necessary using Homography? Or it just use for better results?

Thanks.

[Reply](#)

**Adrian Rosebrock**

May 21, 2017 at 5:06 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-425694>)

It is necessary to compute the homography as this allows us to estimate the affine transformation.

[Reply](#)

**Mohsen khoshnazar**

May 21, 2017 at 5:18 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-425707>)

Thanks for your answer. Yes its right, because we have more than 4 points. Thanks again Adrian.

[Reply](#)

**Cristian**

June 12, 2017 at 8:55 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-427206>)

Hi Adrian, very good job. I want to know how to make the mosaic but without reducing the quality of the images and the resulting mosaic. Greetings.

[Reply](#)



Adrian Rosebrock

June 13, 2017 at 10:55 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-427245>)

Apply keypoint detection, local invariant descriptors, and RANSAC to the resized images, but then apply the homography transfer to the *original, not resized* images.

[Reply](#)



Cristian

June 13, 2017 at 11:47 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-427265>)

Thank you very much, it worked for me but half of the mosaic is presented (I work with four aerial images). In other words, I do not work with a horizontal panorama, rather a square panorama.

[Reply](#)



Daniel

June 17, 2017 at 10:55 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-427555>)

Hi Adrian, Thank you for the tutorial. I wonder how to warp the left image to match the right image instead of warping the right one to match the left one. Thank you.

[Reply](#)



Daniel

June 17, 2017 at 10:59 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-427556>)

I tried to change the parameter of warpPerspective from imageA to imageB and cover the columns 400:800 with imageA, but the left image does not stitch to the right one

[Reply](#)



Adrian Rosebrock

June 20, 2017 at 11:12 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-427770>)

Make sure you update Line 33 where the images are actually stitched together.

[Reply](#)



Lorena

June 18, 2017 at 11:16 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-427619>)

Line 33 what does it represent me ??

[Reply](#)



Adrian Rosebrock

June 20, 2017 at 11:06 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-427765>)

Line 33 is responsible for stitching the actual images together. It accomplishes this via NumPy array slicing.

[Reply](#)



Evan Hutomo (<http://setsu.me>)



June 28, 2017 at 3:05 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-428377>)

Hi Adrian, Great post as usual.

Btw, I wonder about stitching not in panorama way, but match and stack two image with different size, for example: image1 size = 500×500, image2 size = 100×100. image2 is part of spatial area on image1 so it should have high matches keypoint. But when I try with your code, It was said : “could not broadcast input array from shape (...) into shape (...).

I knew this error was already asked before but how about this specific problem?

fyi, I try to make image2 size same as image1 with framing it with black pixel, but image2 always behind on image1.

I try to change the black frame into alpha channel too, but after I try to match it, the alpha channel frame become bright red.

Thanks in advance

[Reply](#)



Adrian Rosebrock

June 30, 2017 at 8:23 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-428582>)

Hi Evan — this is a bit more challenging, but you would need to compute the size of the new, resulting image manually. Take the coordinates of the matched region, compute the overlap, and subtract from image2. This will give you the new dimensions that you need to “pad” the output image by in order to apply the array slicing.

[Reply](#)



Evan Hutomo (<http://setsu.me>)

July 3, 2017 at 12:42 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-428821>)

Hi Adrian, thanks for replying – The compute the overlap part I still didn’t know. Is overlap calculation is a kind of image registration method? It seems overlap calculation is draconian

reference image into sensed image.

Thank you

[Reply](#)



Adrian Rosebrock

July 5, 2017 at 6:17 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-429062>)

You can actually compute the overlap percentage by examining the (x, y)-coordinates of the *matched keypoints*. Loop over the coordinates of the matched keypoints and examine their coordinates.

[Reply](#)



Esraa

June 30, 2017 at 7:36 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-428683>)

How does the stitching happen, I mean how does it define the overlapped area or the key features between two images ? does it compare the RGP of each pixel ?!

[Reply](#)



Adrian Rosebrock

July 5, 2017 at 6:34 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-429073>)

The area of overlap is determined by detecting keypoints, extracting local invariant descriptors, and applying the RANSAC algorithm to determine keypoint correspondences. If you are interested in learning more about this technique, I cover it in both **Practical Python and OpenCV** (<https://www.pyimagesearch.com/practical-python-opencv/>) and inside the

PyImageSearch Gurus course (<https://www.pyimagesearch.com/deep-learning-computer-vision-python-book/>).

[Reply](#)



Cynric

July 1, 2017 at 4:03 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-428722>)

Hi Adrian, your post really helps me a lot.

I'm working on a dual fisheye camera stitching project, that is to stitch the two equirectangular projections of two fisheye images.

So I tried to apply your solution here as the stitching method. But the result I got was unsatisfactory. The "right" image was usually warped too much.

It's probably because the distortion made by equirectangular projection affects the homography matrix.

Do you have any idea to deal with this?

Or, are there any stitching method having better performance on fisheye stitching?

Thank you!

[Reply](#)



Adrian Rosebrock

July 5, 2017 at 6:32 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-429072>)

Hi Cynric — I have not worked with fisheye lenses before, so unfortunately I do not have any guidance in this specific instance. Best of luck with the project!

[Reply](#)



Cynric

July 12, 2017 at 4:57 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-429616>)

Thanks for your reply

[Reply](#)



Shreyash

July 3, 2017 at 12:18 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-428867>)

Hey Adrian,

Great post and a great blog overall! It was very easy to understand, even as a beginner. I tried to modify this code to stitch multiple images (not the best way to do it, but it kinda works). I'm cropping out all the black regions that are left out after the stitching is done and then go on and stitch the next image alongside. However, after stitching a few images, it starts going out of the plane, resulting in a completely stretched out image.

I've gone thru' all the comments and the topic of 'multiple image stitching' keeps coming up again and again. It would really help all of us if you could do a tutorial on that. I did find some other tutorials on this topic but they're nowhere as close to simplicity as this one. I hope you respond to this request soon!

Thanks.

[Reply](#)



Adrian Rosebrock

July 5, 2017 at 6:07 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-429052>)

Hi Shreyash — thank you for the request, I will certainly consider it for the future.

[Reply](#)



Shreyash Gater



Shreyash Gotee

July 6, 2017 at 2:46 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-429200>)

Adrian,

I somehow managed to stitch multiple images using this code but after stitching a few images I get the following error:

ValueError: could not broadcast input array from shape (320,480,3) into shape (297,480,3)

I tried the solution you gave above, but that didn't work.

Also, can you please explain this line of the code:

```
result[0:imageB.shape[0], 0:imageB.shape[1]] = imageB
```

Thanks!

[Reply](#)



Adrian Rosebrock

July 7, 2017 at 9:52 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-429289>)

The code:

```
result[0:imageB.shape[0], 0:imageB.shape[1]] = imageB
```

Is simply a NumPy array slicing. You can read more about NumPy array slicing [here](https://stackoverflow.com/questions/4257394/slicing-of-a-numpy-2d-array-or-how-do-i-extract-an-mxm-submatrix-from-an-nxn-ar) (<https://stackoverflow.com/questions/4257394/slicing-of-a-numpy-2d-array-or-how-do-i-extract-an-mxm-submatrix-from-an-nxn-ar>), as well as inside [Practical Python and OpenCV](https://www.pyimagesearch.com/practical-python-opencv/) (<https://www.pyimagesearch.com/practical-python-opencv/>).

[Reply](#)



Shreyash Gotee

July 7, 2017 at 3:02 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-429311>)

Thanks for the reply Adrian!

I understand that this line is slicing the ‘result’ array or in simpler terms, we are cropping the image. But what I don’t understand is the flow of the code in this line.

Are we taking a slice of that array and equating that to `imageB`, thereby superimposing `imageB` on top of the result, which completes the stitching? Please correct me if I’m wrong.

Also, can you please explain why does slicing a numpy array like this result in broadcasting errors?

Thanks again!



Adrian Rosebrock

July 11, 2017 at 6:54 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-429561>)

Instead of trying to detail every aspect of “broadcasting” and how it turns up in error messages, I think it’s best that you review the [**NumPy documentation on broadcasting**](#)(). In this case, it seems that the output dimensions of the image cannot hold the slice.

The result of the code `0:imageB.shape[0]` starts from $y=0$ to $y=imageB$ (height). The second slice, `0:imageB.shape[1]` starts from $x=0$ to $x=imageB$ (width). Then, `imageB` is stored in this slice of the result.



Adly Harith Abul Khair

November 9, 2017 at 12:04 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-440124>)

Hey Shreyash, could you share with me your modified code to stitch multiple images? Or give me some tips on how to do it from Adrian’s code?

Thanks!

[Reply](#)



Ayush Jindal



[July 6, 2017 at 6:25 am \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-429162>\)](https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-429162)

Hi Adrian,

When i ran the above code i got this:

Traceback (most recent call last):

File "stitch.py", line 16, in

(result, vis) = stitcher.stitch([img1, img2], showMatches=True)

TypeError: 'NoneType' object is not iterable

I am actually initialising 2 webcameras and taking input from them rather than using argparse. Rest of the steps are as you have mentioned above. I have checked and both img1 and img2 are initialised(I used imshow). Can you please help me out

[Reply](#)



Adrian Rosebrock

[July 7, 2017 at 9:59 am \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-429294>\)](https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-429294)

I know you mentioned validating the images via `cv2.imshow`, but I would double and triple check this. It really sounds like that one or both of the images you are trying to stitch are not being properly read from the webcam(s). I discuss these types of NoneType errors in [this blog post](https://www.pyimagesearch.com/2016/12/26/opencv-resolving-nonetyperrors/) (<https://www.pyimagesearch.com/2016/12/26/opencv-resolving-nonetyperrors/>).

[Reply](#)



Ayush Jindal

[July 7, 2017 at 3:27 pm \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-429313>\)](https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-429313)

Hi,

I went through the link and as suggested, have verified both `cv2.VideoCapture` and `cv2.imread`. I am not missing any codecs and rest of the codes are running just fine. What else can I check??

[Reply](#)



Ayush Jindal

July 8, 2017 at 2:51 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-429331>)

No I am sure that the image is loading. I have used imshow as well as first wrote the frames from videotream then loaded through imread. Still giving the same error

[Reply](#)



Adrian Rosebrock

July 11, 2017 at 6:46 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-429555>)

That is indeed quite strange behavior. I'm not sure why this may happen off the top of my head. I would start inserting more print and cv2.imshow statements into the code until you can see exactly where the error happens.

[Reply](#)



Ayush Jindal

July 14, 2017 at 4:12 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-429823>)

Hi Adrian,

The error was not with the image but actually with the M. For some reason it always returned None.

The error it shows while i tried to debug is

```
OpenCV Error: Bad argument (The input arrays should be 2D or 3D point sets) in
findHomography, file /home/ayush/opencv/opencv-
3.2.0/modules/calib3d/src/fundam.cpp, line 341
.. (H, status) = cv2.findHomography(ptsA, ptsB, cv2.RANSAC,reprojThresh) error: (-5) The
input arrays should be 2D or 3D point sets in function findHomography
```

Any insights on why this is happening??

**Adrian Rosebrock**

July 14, 2017 at 7:18 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-429836>)

It sounds like not enough keypoints and local invariant descriptors are being detected and matched. How many keypoints are being computed for each image?

**Binks**

July 23, 2017 at 7:38 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-430605>)

Hi Adrian,

I'm just curious, if I would like to move the seam to the left – say, stitch it a little bit to the left, maybe more central – does that mean I have to restrict region of matching points on the right-hand-side image? How else could I achieve that...?

Thanks!

[Reply](#)

**Adrian Rosebrock**

July 24, 2017 at 3:32 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-430693>)

I'm not sure what you can by "move the seam to the left". Can you elaborate on your question?

[Reply](#)

**Binks**

July 24, 2017 at 8:19 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-430736>)

At this moment. on the stitched panorama. there is 100% of the imageA (left). I meant to only

include 70% of the left image.

[Reply](#)



Adrian Rosebrock

July 28, 2017 at 10:12 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-431060>)

If you want to only include 70% of the left image you would either (1) crop the left portion of the image you don't need via NumPy array slicing or (2) after detecting keypoints, remove any keypoints from the list that fall into the 30% range that you do not want to stitch. This can be accomplished by examining the (x, y)-coordinates of the keypoints.

[Reply](#)



Binks

July 23, 2017 at 10:30 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-430618>)

Hi Adrian,

I would like to move the seam a bit to the left, say in the middle of the left-hand-side photo? Is it possible by restriction of keypoints on the right-hand-side photo? Or, is there another more efficient way ?

Thanks!

[Reply](#)



Binks

July 23, 2017 at 10:31 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-430619>)

sorry for two questions, my browser must had played tricks on me... 😊

[Reply](#)



Ali

August 28, 2017 at 10:36 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-433587>)

Hi Adrian,

Is it possible to change panorama projection from Equirectangular to rectilinear in python?

[Reply](#)



Mika

August 31, 2017 at 10:52 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-433774>)

Hi Adrian,

Thank you so much for this post!

Is there any parameter I could use to neutralize the rotation of the pictures during the stitching?

[Reply](#)



Adrian Rosebrock

September 1, 2017 at 12:01 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-433852>)

Hi Mika — I'm not sure what you mean by "neutralize" the rotation.

[Reply](#)



RAEBEN

**Mika**

September 3, 2017 at 2:34 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-434017>)

Hi,

Using a microscope, I took pictures of a sample from above line by line:

I “split” the sample into squares, took a photo of each square in the line, and then moved to the next line.

Now I am trying to stitch them all together to get a picture of the whole sample.

Therefore, I don’t want the algorithm to rotate the pictures more than 1-2 degrees in trying to match the keypoints, and if it is possible, I had like to create an adjusted homography (or change another parameter), to optimize the algorithm.

Reply

**Adrian Rosebrock**

September 5, 2017 at 9:25 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-434131>)

I’m not an expert in microscope-captured images and it’s also a bit hard to provide a suggestion without seeing example images. However, if you’re worried about a 1-2 degree rotation than I’d be concerned about the quality of the matched keypoints you are receiving in the first place. Cellular structures can look very similar and aren’t exactly the intended use case of keypoint detectors + local invariant descriptors. It may be the case that this panorama stitching method isn’t appropriate for your images.

Reply

**Mika**

September 5, 2017 at 10:15 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-434157>)

I used a few filters and I have managed to find the right keypoints. Now I am trying to calculate the translation and stitch the images according this parameter only (instead of using the knnMatch function). Thanks a lot any way!

[Reply](#)



Dan Pollock (<http://none>)

[September 26, 2017 at 10:04 pm \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-435776>\)](#)

I need to stitch 4 images from a 4 camera array. Can the stitch procedure be extended to do 4 or would I need to do the two pairs then stitch those? Can the opencv routine handle more than 2 images?

[Reply](#)



Adrian Rosebrock

[September 27, 2017 at 6:44 am \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-435810>\)](#)

The method I presented in this post is only intended for two images. I'll try to cover image stitching with more than two images in the future.

[Reply](#)



Mostafa

[October 8, 2017 at 3:29 pm \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-436814>\)](#)

What do you think about large images with high resolution? images with size near GB.

[Reply](#)



Adrian Rosebrock

[October 9, 2017 at 12:21 pm \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-436947>\)](#)

You could try resizing your input images to 500-600px along the maximum dimension, obtaining the transformation matrix M, and then applying the stitching to the original large images using the matrix M.

[Reply](#)



Deniz

October 30, 2017 at 8:12 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-439145>)

Hello Adrian!

Thank you so much for the tutorial. It helped me a lot in the process of understanding the stitching pipeline. Currently, I'm trying to convert the code to C++, in order to use OpenCV CUDA functions while warping images. However, I didn't understand what happens in lines 92 & 93:

```
ptsA = np.float32([kpsA[i] for (_, i) in matches])
ptsB = np.float32([kpsB[i] for (i, _) in matches])
```

Could you please explain how ptsA & ptsB are obtained from kpsA and kpsB? There is a similar notation in line 69:

```
kps = np.float32([kp.pt for kp in kps])
```

In this one, at every iteration of the for loop, we take an object "kp" from kps, and append its "pt" property to the kps on the left hand side of the equation. However, the notation used in 92 & 93 looks a bit different and I couldn't understand what $(_, i)$ and $(i, _)$ mean, and what is the correlation between the elements of the matches vector, and $kpsA[i]$ and $kpsB[i]$ elements. I would be glad if you could help me with this part.

Thank you for your time.

[Reply](#)



Adrian Paschbach



Adrian Rosebrock

October 30, 2017 at 1:29 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-439167>)

Hi Deniz, thank you for your comment. Converting this code to C++ is quite the task but you will achieve some speed. It seems you have 2 questions so I'll answer them both.

1) **List comprehension in Python**

(<https://docs.python.org/3/tutorial/datastructures.html#list-comprehensions>) is just a concise notation for building a list.

2) (i, _) is tuple notation. In general the _ means that you don't care about the value and you are ignoring it. It's a Python convention.

[Reply](#)



Deniz

October 31, 2017 at 11:19 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-439302>)

Thank you so much for your answer Adrian. I will take look at the examples to understand the notation.

[Reply](#)



Adly Harith Abul Khair

November 9, 2017 at 12:08 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-440125>)

Hi Adrian,

I have ran your image-stitching code. I now would like to modify it to implement an image-stitching algorithm for more than 2 images. Could you give me some hints/tips/references on how to do that?

I thank you.

[Reply](#)



Adrian Rosebrock

November 13, 2017 at 2:24 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-440490>)

Working with more than two images dramatically changes the algorithm. It's too much for me to cover in a comment — I will try to do a detailed tutorial on multi-image stitching.

[Reply](#)



james

November 27, 2017 at 6:13 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-441743>)

hi bro your site is very good for me thanks for every think i want to use this source real time i mean i want to make this : my drone is flying and take pictures number 1 and save it and in a few later in certain location using gps take another shot so i want to stitch image 2 and 1 and 3 or many more image and make map or mapping so can you help me to do that ?

[Reply](#)



Adrian Rosebrock

November 27, 2017 at 12:58 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-441774>)

I cover real-time image stitching (<https://www.pyimagesearch.com/2016/01/25/real-time-panorama-and-image-stitching-with-opencv/>) in this post. I will try to cover more in-depth image stitching (including panoramas with more than two photos) in a future blog post.

[Reply](#)





james

November 29, 2017 at 1:25 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-441967>)

Awesome! I'll be waiting for your tutorial to test it! thanks again! ?

Reply



james

November 30, 2017 at 4:08 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-442025>)

how can i stitch more than two image? i mean stitch 50 image and create air map can you give me a link or something and if you can post in futures please tell me when? thank you for every think

Reply



Adrian Rosebrock

November 30, 2017 at 3:34 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-442047>)

Hi James — I do not have any resources directly for putting together an aerial map. I cannot say when I will cover a tutorial on that in the future, but I will certainly try to.

Reply



Brendon

November 29, 2017 at 8:25 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-441984>)

Adrian,

Thanks for the post! While I have this working for the most part one quick question. I'd like to hard mount my cameras to a fixture. If I know the amount of overlap, is there anyway to hard code the points for homography with a number of pixels from an edge or something similar?

I thanks!

[Reply](#)



Adrian Rosebrock

[November 30, 2017 at 3:39 pm \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-442049>\)](#)

If you want you can compute the homography once and serialize the weights to disk and then re-load the weights each time the script runs. This will work since the camera is fixed and non-moving. However, it might be good to periodically recompute the homography matrix just in case the camera shifts slightly.

[Reply](#)



Kelsier

[December 4, 2017 at 7:13 am \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-442315>\)](#)

Thanks for the post!!

Traceback (most recent call last):

File "stitch.py", line 22, in

imageA = imutils.resize(imageA, width=400)

File "C:\Users\Lisbon\Anaconda3\lib\site-packages\imutils\convenience.py", line 69, in resize

(h, w) = image.shape[:2]

AttributeError: 'NoneType' object has no attribute 'shape'

I'm not sure what's going on imutils here?

Did install incorrectly?

Thanks for your reply

[Reply](#)



Adrian Rosebrock

**Adrian Rosebrock**

December 5, 2017 at 7:34 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-442425>)

It sounds like your path to your input images are incorrect. Your imutils install is fine, but your image paths passed into cv2.imread do not exist. I would also suggest on reading up on **NoneType errors in OpenCV** (<https://www.pyimagesearch.com/2016/12/26/opencv-resolving-nontype-errors/>). I hope that helps!

[Reply](#)

**Jane**

April 16, 2018 at 1:23 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-456940>)

Hi Kelsier!

It's pretty late but I'm wondering if you managed to solve this?

[Reply](#)

**Sameer**

February 8, 2018 at 8:33 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-449520>)

I want to stitch 115 images , but your approach goes on increasing the width of the image and i want aa efficient approach to stitch all those images. It's very urgent . atleast an approach to be followed will be appreciated

[Reply](#)

**Mezher**

April 10, 2018 at 10:42 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-455917>)

Can we use BRISK detector instead of SIFT ?

[Reply](#)



Adrian Rosebrock

April 10, 2018 at 11:52 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-455928>)

Yes. Provided you have enough keypoint correspondences you can use a different combination of keypoint detector and local invariant descriptor. Just make sure if you are using binary features to update the distance function to use the Hamming distance.

[Reply](#)



kamal

April 12, 2018 at 9:45 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-456321>)

How can I modify this to stitch whole of two images i.e combine whole of two images into single frame?

[Reply](#)



Adrian Rosebrock

April 13, 2018 at 6:48 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-456466>)

You mean something like [this](https://www.pyimagesearch.com/2016/01/25/real-time-panorama-and-image-stitching-with-opencv/) (<https://www.pyimagesearch.com/2016/01/25/real-time-panorama-and-image-stitching-with-opencv/>)?

[Reply](#)



ramy

April 15, 2018 at 5:55 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-456466>)

[April 13, 2016 at 9:55 pm \(https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-456793\)](https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-456793)

Hi,

I get imageB only as an output while performing the above code.

can you help me with this issue?

Thanks,

[Reply](#)



Adrian Rosebrock

[April 16, 2018 at 2:24 pm \(https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-456961\)](https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-456961)

It sounds like your input images may be ordered differently. This code assumes left-to-right ordering but you may have a different ordering.

[Reply](#)



Mezher

[May 3, 2018 at 9:25 am \(https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-459738\)](https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-459738)

hello adrian ,

can you please tell us how to implement this method to stitch multiple images !

Thanks in advance

[Reply](#)



Adrian Rosebrock

[May 3, 2018 at 9:27 am \(https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-459740\)](https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-459740)

Hey Mezher — I haven't had a chance to write a multi-image stitching post yet. I'm not sure

when I will be able to do that (I'm very busy) but I hope to be able to do it eventually.

[Reply](#)



Deb

May 6, 2018 at 10:22 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-460202>)

Hello Adrian,

Can this algorithm be adapted to make a 3D model from an adequate number of images? I was just looking at the research paper “Building Rome in a Day”. what do you suggest?

[Reply](#)



Adrian Rosebrock

May 9, 2018 at 10:08 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-460670>)

The paper you are referring to actually refers to building a 3D reconstruction based on keypoint matching. The paper also introduced a number of novel parallel optimizations. Keypoint matching and panorama stitching are two different computer vision topics.

[Reply](#)



Leif

May 21, 2018 at 9:17 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-463125>)

Thank you, great code! Is this approach also working when there is a translation of the camera between the images? Say I image half of a long painting, move a meter or so to the side and image the other half (with some overlap)?

[Reply](#)



Adrian Rosebrock

May 22, 2018 at 6:01 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-463324>)

As long as there is some overlap this method should work. By “some overlap” I’m referring to enough valid keypoint matches to construct the homography matrix.

[Reply](#)



Saad

May 26, 2018 at 8:39 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-464425>)

Hi Adrian,
Can you help me in implementing this code for multiple images?

[Reply](#)



Danish

June 7, 2018 at 3:44 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-467319>)

Hi Adrian,

How do I obtain result without having image B warped/distorted after stitching. I would like to have an output where the keypoint matches and overlap over each other without any perspective warps.

[Reply](#)



Adrian Rosebrock

**Adrian Rosebrock**

June 7, 2018 at 3:01 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-467382>)

Hey Danish — I'm not sure I understand the goal here. Are you specifically asking about drawing/visualizing the keypoints? Keep in mind that that it may be impossible to have the keypoints lineup and match without a perspective transform.

[Reply](#)

**Danish**

June 8, 2018 at 12:15 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-467435>)

Let say I have a really long horizontal pole and I have a drone scanning from left to right(the images having a certain % of overlap) maintaining constant distance. the pole could either become thinner or larger along its length.

I understand this is another problem as its unlike a normal panorama where there is a center point. But what I'm trying to achieve is just to detect the features between two consecutive images and put them over each other without a perspective warp.

"Keep in mind that that it may be impossible to have the keypoints lineup and match without a perspective transform." Since this is the case, would you happen to know a work around? I'm still relatively new to python so theres a lot I still don't know.

[Reply](#)

**Adrian Rosebrock**

June 8, 2018 at 6:51 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-467478>)

Depending on how the pole looks it may be impossible to detect enough keypoints to stitch the images together in the first place. Keypoints require enough edges, corners, and "blobs" in an image to create the correspondence. Without seeing enough images of this "pole" I wouldn't be able to provide any specific recommendations.

As far as scanning from left-to-right, have you tried any basic image processing operations like thresholding or edge detection just to see if you can segment the pole easily? A quick hack would be to segment the pole and then fit a line through the center of the region.

If you're new to Python and OpenCV I would recommend that you read through [**Practical Python and OpenCV**](#) (<https://www.pyimagesearch.com/practical-python-opencv/>) to help you get up to speed. The book teaches you the core fundamentals and would better prepare you for your project.

[Reply](#)



crescent

June 22, 2018 at 3:25 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-468894>)

"In future blog posts we'll extend our panorama stitching code to work with multiple images rather than just two." you said. Did not you post? Or Am I missing the post on website?

[Reply](#)



Adrian Rosebrock

June 25, 2018 at 2:04 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-469166>)

I simply have not written the post. I've been too busy with other topics. I'd like to circle back to this but I'm not sure if/when that may be.

[Reply](#)



Danielle

July 14, 2018 at 5:19 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-471038>)

Hi Adrian,

Thank you for your very helpful post. I am a high school student trying to learn OpenCV and your posts have helped me tremendously!!

I am trying to stitch 4 images together. Say i1, i2, i3, and i4 and they are same focus and there is overlap in the images between i1 and i2, between i2 and i3, and i3 and i4. They have same focus.

When I stitch i1 with i2, it returns an image with a large black border on the right. I am struggling to “crop” that black portion. Would you kindly put a code snippet to remove the black right border from the result.

This is what I am trying to do: (hopefully I can stitch all 4 images or more this way)

Stitch i1 and i2, get result1.

Stitch i3 and i4, get result2.

Then stitch result1 with result2 (there should be some “overlap” in image as i2 and i3 have overlaps.

Thank you very much for your help.

[Reply](#)



Adrian Rosebrock

July 17, 2018 at 7:32 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-471266>)

Hi Danielle — congrats on studying computer vision at such a young age, that's very impressive. I unfortunately do not have any tutorials on stitching images together with more than 2 images. It's been a topic I've wanted to cover but never been able to get to. I don't have any code snippets for removing the black border either but I do hope that another PylImageSearch reader may be able to help out with the project.

[Reply](#)



[July 29, 2018 at 4:05 pm \(https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-472367\)](https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-472367)

Hello Adrian, I am following your posts and I really appreciate you. Your articles really solve my problems and encourage me. I have two questions about your panorama stitching code

1-) When inputs are grayscaled, it gives an error. Could we apply stitching to grayscaled images?

2-) Is there a solution for the images that has different light intensity and different focus. I want to apply it to a moveable car, but since car is mobile the photos have different light averages. Is there any method for equalizing the light in both picture?

Thank you very much for your help.

[Reply](#)



Adrian Rosebrock

[July 31, 2018 at 9:55 am \(https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-472585\)](https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-472585)

1. What is the error you are getting? Without knowing the error I'm not sure what the problem may be.

2. When you say "solution" what are you referring to? Obtaining the actual output stitched images? Or "blending" the two together to make it look like the output was taken from a "single" camera sensor with the same lighting conditions?

[Reply](#)



Andre

[August 2, 2018 at 7:39 pm \(https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-472863\)](https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-472863)

Hello Adrian, thanks for the post very helpful when walking through it step by step.

Whenever I try to stitch images that should have a ton of keypoint matches I'm only getting like 2-4 matches max when it should obviously be a lot more. I'm literally taking a complete image and splitting it in 2 then feeding them in so I don't see why it won't work.

I'm trying to use this without imutils because the system I need to run it on only has opencv on it natively so I took out the resize function in stitcher.py. Other than that, I didn't change much else.

Any ideas why this is happening?

[Reply](#)



Adrian Rosebrock

[August 7, 2018 at 7:42 am](#) (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-473398>)

How large, in terms of pixels, are your input images? I think the issue may be that you're trying to stitch very large images but not resizing when computing the homography matrix. Secondly, it may be the case that you aren't detecting a ton of keypoints — you should verify that as well.

[Reply](#)



Shuyi

[August 20, 2018 at 4:01 pm](#) (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-474973>)

Hi Adrian, here you said:

"Line 15 unpacks the images list (which again, we presume to contain only two images). The ordering to the images list is important: we expect images to be supplied in left-to-right order. If images are not supplied in this order, then our code will still run — but our output panorama will only contain one image, not both."

When I'm applying the algorithm, I have two troubles:

(1) What if I am processing a stream of photos and there is no chance for me to know the

sequence of image in advance? Is there anyway I can tell the sequence in Python/ during the process of stitching?

(2) Trying to answer the first question myself, I swapped the two images in the sequence. I want to detect the sequence by looking at the ptA and ptB in the Stitcher.drawMatches(). Applying image A and B in the right order, it works fine, but when I apply B and A (in the opposite sequence), it produces wrong matches between points. The output contains more than one image, not following what you described.

So my biggest problem is how to use this great method without knowing the sequence of the two images and if possible, how should I detect the sequence?

Thanks

[Reply](#)



Adrian Rosebrock

August 22, 2018 at 9:50 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-475167>)

The method in this post assumes you have a priori knowledge regarding image ordering. Without this knowledge, the method will not work. You need to know the image ordering.

[Reply](#)



Priyanka

October 14, 2018 at 12:11 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-482489>)

how to go for stitching more than two images

[Reply](#)



Kamil Hassaan

October 30, 2018 at 8:03 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-482489>)

[stitching/#comment-484625\)](#)

I am getting these errors:

...

descriptor = cv2.xfeatures2d.SIFT_create()

AttributeError: module 'cv2.cv2' has no attribute 'xfeatures2d'

How can this be fixed?

[Reply](#)



Adrian Rosebrock

[November 2, 2018 at 8:23 am \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-485038>\)](#)

You need to install OpenCV with the “opencv_contrib” module enabled. You can [follow one of my OpenCV install tutorials \(<https://www.pyimagesearch.com/opencv-tutorials-resources-guides/>\)](#) to help you install OpenCV correctly.

[Reply](#)



Ubaid

[November 10, 2018 at 9:19 am \(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-486424>\)](#)

I have two images of the same scene just slightly translated(not stereo). I want them perfectly aligned right on top of each other to perform image differencing. So far I've been able to perform the feature detection using which I computed the fundamental matrix but i don't know what to do next. I want to know how to exploit that fundamental matrix and transform either of the two images so that they can be subtracted perfectly. Any help in this regard would be very much appreciated.

[Reply](#)



[Adrian Rosebrock](#)



Adrian Rosebrock

November 10, 2018 at 9:51 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-486431>)

Hi Ubaid — you mentioned using the “fundamental matrix” which makes me think you are performing stereo vision, is that correct? Either way, is there a particular reason you do not want to use the technique covered in this tutorial?

[Reply](#)



Ubaid

November 10, 2018 at 12:07 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-486475>)

I have two aerial shots of a road(same location); one with a car and the other without. I just want to subtract the two and be able to get the car as a difference but that is not the case since i have to transform them first. I couldn't find anything related to transformation of images that i could use for my application. Kindly help me.

[Reply](#)



Teodor

December 4, 2018 at 6:00 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-490616>)

Hi Adrian,

Thank You for your opencv panorama stitching tutorial, it is a great starting point. I am testing it with 4 IP cameras and it is working but I have the issue that the image that is warpPerspective it is stretched.

I have the same result also with 2 Images, the image that I add to the stitching is stretched and kind of distorted. Is there a way to stitch two images without distorting the warpPerspective one. Can you please give me a hint what flags or values I can start twerking?

I thank You!

Teo

[Reply](#)



Adrian Rosebrock

December 4, 2018 at 9:36 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-490654>)

The issue isn't with the warpPerspective per se. The issue is not enough keypoints are being correctly detected and matched between frames. Try looking into your keypoint matching procedure.

[Reply](#)



Teodor

December 12, 2018 at 5:05 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-491843>)

Hi Adrian,

Thank You very much for your reply. Since your reply I was trying to look into the keypoint matching procedure, I have used the same procedure from your example and other online procedures. From what I have seen there are enough matching keypoints between the 2 images, the stitching is perfect, very nice, no difference between the images but the second image that is warp-ed it is kind of 'adjusted' to the first image trying to fit it in but stretching it.

I cannot attach the images here but I have analyzed your stitching and if you look also at your stitching example the second image on the right it is not the same with the original one, it is 'stretched' (maybe not the right wording), the ground are where the trees are on the right is no longer there in the stitched image. Is there a way that you are aware of to point me on the direction to adjust this?

Thank You very much in advanced.

Kind Regards,
Teo

[Reply](#)



Adrian Rosebrock

December 13, 2018 at 9:05 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-492063>)

I'm actually publishing a brand new tutorial on image stitching this coming Monday. Keep an eye on the PylImageSearch blog for it, I think it will help you.

[Reply](#)



Teodor

December 14, 2018 at 8:48 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-492223>)

☺ Thank You! I am waiting with interest the new tutorial. Teo

[Reply](#)



Adrian Rosebrock

December 18, 2018 at 9:24 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-492843>)

The new tutorial can be found [here. \(https://www.pyimagesearch.com/2018/12/17/image-stitching-with-opencv-and-python/\)](https://www.pyimagesearch.com/2018/12/17/image-stitching-with-opencv-and-python/)

[Reply](#)



Emmanuel

December 14, 2018 at 10:31 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-492235>)

Hey Adrian, ive read all the post an comments. Great post. I now looking for stitching >2

images not only horizontal. Is there anywhere to start? or some code or blog you have provided before for this?

thanks

[Reply](#)



Adrian Rosebrock

December 18, 2018 at 9:20 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-492838>)

Be sure to refer to my [latest guide on image stitching](#).

(<https://www.pyimagesearch.com/2018/12/17/image-stitching-with-opencv-and-python/>)

[Reply](#)



dan

December 28, 2018 at 5:32 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-494034>)

Greetings Adrian and a wonderful opencv stitching project. Do you have Python or C++ stitching information on LDA “linear-diode-array” LINE cameras? They seem to be popular in document scanners, astronomy “NASA images”, and X-ray machines that scan objects on conveyor belts for defects. I have seen two x-ray images stitched together to produce a very nice panoramic picture. But I do not know if it was done with opencv? The LINE_cameras have great contrast and resolution detail as the image is put together line-by-line. There are some hobby LDA line camera software examples but not much on opencv code. Adrian, do you have any suggestions? Thank you.

[Reply](#)



Adrian Rosebrock

January 2, 2019 at 9:37 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-494509>)

Sorry, I do not have any LDA/LINE related code for image stitching.

[Reply](#)



Aniket Mallick

March 18, 2019 at 12:46 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-507402>)

Hi Adrian,

A great tutorial overall, I had a query as to which IDE or environment you're running your programs in?

[Reply](#)



Adrian Rosebrock

March 19, 2019 at 10:03 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-507826>)

I like to use Sublime Text or PyCharm to write code. I always execute my code via the command line.

[Reply](#)



Katarina (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/>)

March 19, 2019 at 6:53 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-507738>)

Hi Adrian. I downloaded your code from this website

(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/>)

(<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/>) but I want to run it, there is error:

ImportError: No module named imutils

I work in Python3 on OS Raspbian. Where is problem? Thank you

[Reply](#)



Adrian Rosebrock

March 19, 2019 at 9:50 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-507809>)

You need to install “imutils” on your system:

```
$ pip install imutils
```

[Reply](#)



captainst

April 9, 2019 at 9:28 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-512072>)

Another excellent tutorial !

I found that increasing the value of parameter “ratio” in function stitch could increase the accuracy of stitching and reduce the displacement.

[Reply](#)



Adrian Rosebrock

April 12, 2019 at 12:14 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-512501>)

Thanks for sharing!

[Reply](#)



Frank

May 9, 2019 at 8:39 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-516830>)

what about video stitching? would this be possible by just swapping the variable corresponding to video feed rather than the images?

Reply



Adrian Rosebrock

May 15, 2019 at 3:15 pm (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-517832>)

It's possible. See **this tutorial.** (<https://www.pyimagesearch.com/2016/01/25/real-time-panorama-and-image-stitching-with-opencv/>)

Reply



Mark

October 16, 2019 at 1:35 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-564527>)

Hi Adrian

Thanks your sharing!

I have a task need someone to help:

use bilinear interpolation to stitch 2 picture's 3 same points become a panorama picture

I think the result is very like your example, but the 'way' to completed is different

wish you can give me some tips, thanks!!

[Reply](#)



Muhammad Shahzaib

January 27, 2020 at 5:45 am (<https://www.pyimagesearch.com/2016/01/11/opencv-panorama-stitching/#comment-678893>)

Hi Adrian,

Thank You for your opencv panorama stitching tutorial, it is a great starting point. I am testing it with single camera video and it is working but I have the issue that when it read images from directory it distort the original order and while cropping it only fives the small images region and cropping most of the region. So please guide me how can I implement real time panorama of single camera video frames just like mobile camera panorama frame

Thank You!

Shahxoo

[Reply](#)

Before you leave a comment...

Hey, Adrian here, author of the PyImageSearch blog. I'd love to hear from you, but before you submit a comment, **please follow these guidelines:**

- **If you have a question, read the comments first.** You should also search this page (i.e., **ctrl + f**) for keywords related to your question. It's likely that I have already addressed your question in the comments.
- **If you are copying and pasting code/terminal output, please don't.** Reviewing another programmers' code is a very time consuming and tedious task, and due to the volume of emails and contact requests I receive, I simply cannot do it.

- **Be respectful of the space.** I put a lot of my own personal time into creating these free weekly tutorials. On average, each tutorial takes me 15-20 hours to put together. I love offering these guides to you and I take pride in the content I create. Therefore, I will not approve comments that include large code blocks/terminal output as it destroys the formatting of the page. Kindly be respectful of this space.
- **Be patient.** I receive 200+ comments and emails per day. Due to spam, and my desire to personally answer as many questions as I can, I hand moderate all new comments (typically once per week). I try to answer as many questions as I can, but I'm only one person. Please don't be offended if I cannot get to your question
- **Do you need priority support? Consider purchasing one of my books and courses.** I place customer questions and emails in a separate, special priority queue and answer them first. **If you are a customer of mine you will receive a guaranteed response from me.** If there's any time left over, I focus on the community at large and attempt to answer as many of those questions as I possibly can.

Thank you for keeping these guidelines in mind before submitting your comment.

Similar articles

I was just interviewed on the Talk Python to Me Podcast.

June 9, 2015

(<https://www.pyimagesearch.com/2015/06/09/i-was-just-interviewed-on-the-talk-python-to-me-podcast/>) 

IMAGE PROCESSING TUTORIALS

Long exposure with OpenCV and Python

August 14, 2017

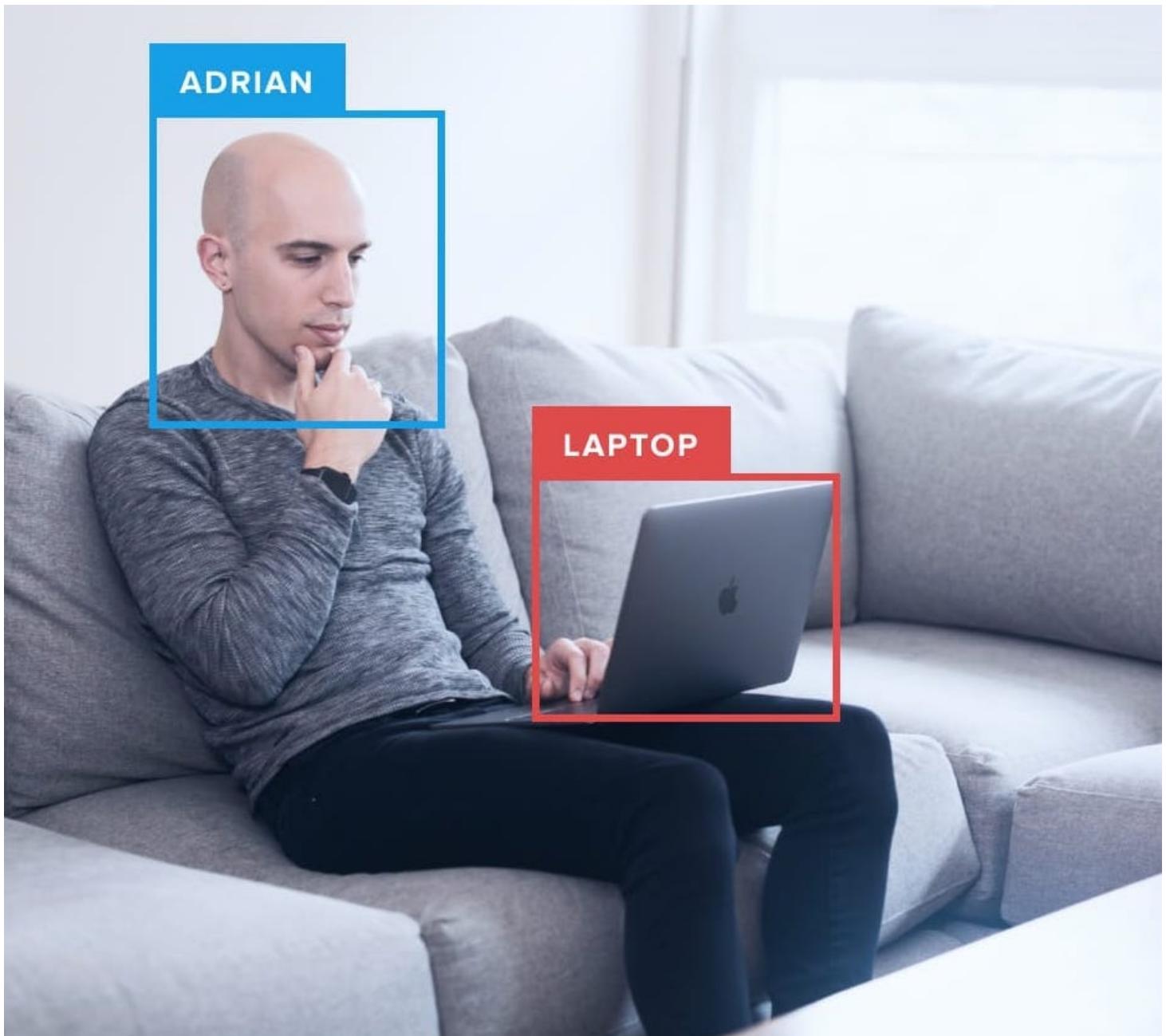
(<https://www.pyimagesearch.com/2017/08/14/long-exposure-with-opencv-and-python/>) 

KICKSTARTER RASPBERRY PI

Table of Contents – Raspberry Pi for Computer Vision

April 5, 2019

(<https://www.pyimagesearch.com/2019/04/05/table-of-contents-raspberry-pi-for-computer-vision/>) 



You can learn Computer Vision, Deep Learning, and OpenCV.

Get your FREE 17 page Computer Vision, OpenCV, and Deep Learning Resource Guide PDF. Inside you'll find my hand-picked tutorials, books, courses, and libraries to help you master CV and DL.

Topics

[Machine Learning and Computer Vision](#)
[\(https://www.pyimagesearch.com/category/machin](https://www.pyimagesearch.com/category/machin)

Deep Learning

(<https://www.pyimagesearch.com/category/deep-learning-2/>)

Dlib Library

(<https://www.pyimagesearch.com/category/dlib/>)

Embedded/IoT and Computer Vision

(<https://www.pyimagesearch.com/category/embedded/>)

Face Applications

(<https://www.pyimagesearch.com/category/faces/>)

Image Processing

(<https://www.pyimagesearch.com/category/image-processing/>)

Interviews

(<https://www.pyimagesearch.com/category/interviews/>)

Keras

(<https://www.pyimagesearch.com/category/keras/>)

Books & Courses

FREE CV, DL, and OpenCV Crash Course

(<https://www.pyimagesearch.com/free-opencv-computer-vision-deep-learning-crash-course/>)

Practical Python and OpenCV

(<https://www.pyimagesearch.com/practical-python-opencv/>)

Deep Learning for Computer Vision with Python

(<https://www.pyimagesearch.com/deep-learning-computer-vision-python-book/>)

PylImageSearch Gurus Course

(<https://www.pyimagesearch.com/pyimagesearch-gurus/>)

Raspberry Pi for Computer Vision

(<https://www.pyimagesearch.com/raspberry-pi-for-computer-vision/>)

e-learning-2/

Medical Computer Vision

(<https://www.pyimagesearch.com/category/medical/>)

Optical Character Recognition (OCR)

(<https://www.pyimagesearch.com/category/optical-character-recognition-ocr/>)

Object Detection

(<https://www.pyimagesearch.com/category/object-detection/>)

Object Tracking

(<https://www.pyimagesearch.com/category/object-tracking/>)

OpenCV Tutorials

(<https://www.pyimagesearch.com/category/opencv/>)

Raspberry Pi

(<https://www.pyimagesearch.com/category/raspberry-pi/>)

PylImageSearch

Get Started (<https://www.pyimagesearch.com/start-here/>)

OpenCV Install Guides

(<https://www.pyimagesearch.com/opencv-tutorials-resources-guides/>)

About (<https://www.pyimagesearch.com/about/>)

FAQ (<https://www.pyimagesearch.com/faqs/>)

Blog (<https://www.pyimagesearch.com/topics/>)

Contact (<https://www.pyimagesearch.com/contact/>)

Privacy Policy

(<https://www.pyimagesearch.com/privacy-policy/>)

© 2020 PylImageSearch (<https://www.pyimagesearch.com>). All Rights Reserved.