

pano_align tutorial

Written by Paul Bourke

October 2020

Introduction

The RoundShot stereoscopic panorama camera records a left and right eye image on two rolls of film. This film is then scanned by some means, the panoramas will be located at different positions in the resulting images, they may be slightly stretched and almost always slightly rotated with respect to each other. The final pair of panoramas need to be perfectly aligned within their respective image, be identical in terms of scale and angle. Assuming the pair have been captured correctly the alignment must not add any vertical parallax. Additionally, the left and right edges need to wrap seamlessly and the panoramas horizontally aligned to define a zero parallax distance.

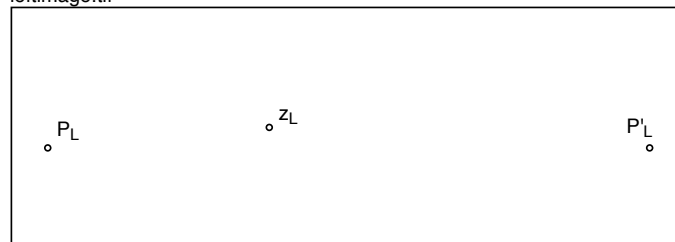
In theory one could achieve the alignment required in an image editing package such as PhotoShop, but in reality it would be highly problematic due to the many degrees of freedom involved, and it would be quite time consuming. The pano_align software is designed to perform the alignment automatically once an operator has defined 3 points in each of the image.

Parameter file

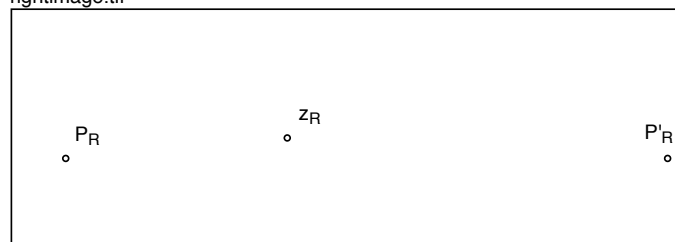
The parameter file contains the information required to perform the alignment. pano_align reads the parameter file which includes the name of the two images and the coordinates of three points in each. It aligns the images and along with settings chosen in the command line parameters. It creates two output images which are the aligned panoramas. The parameter file is a simple text file consisting of 8 lines, it has the following format

```
left file name
x and y pixel values in the left image corresponding to a feature to have zero parallax
x and y pixel values in the left image of a feature in the left overlap zone
x and y pixel values in the left image of the same feature in the right overlap zone
right file name
x,y pixel values in the right image corresponding to a feature to have zero parallax
x and y pixel values in the right image of the same feature in the left overlap zone
x and y pixel values in the right image of the same feature in the right overlap zone
```

leftimage.tif



rightimage.tif



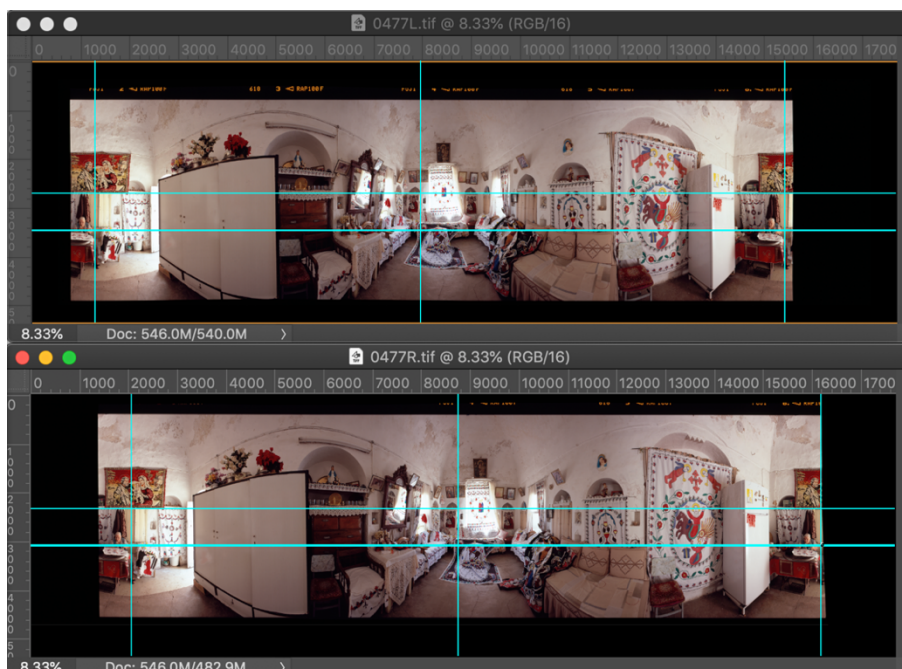
An example of a parameter file might be

```
0477L.tif
7957 2710
1284 3462
15414 3473
0477R.tif
8741 2344
2050 3088
16176 3109
```

In what follows two sample images are supplied, 0477L.tif and 0477R.tif, the parameter file above applies to these two image pairs. Note that it isn't a unique parameter file, one might choose different feature points or an object at a different zero parallax distance.

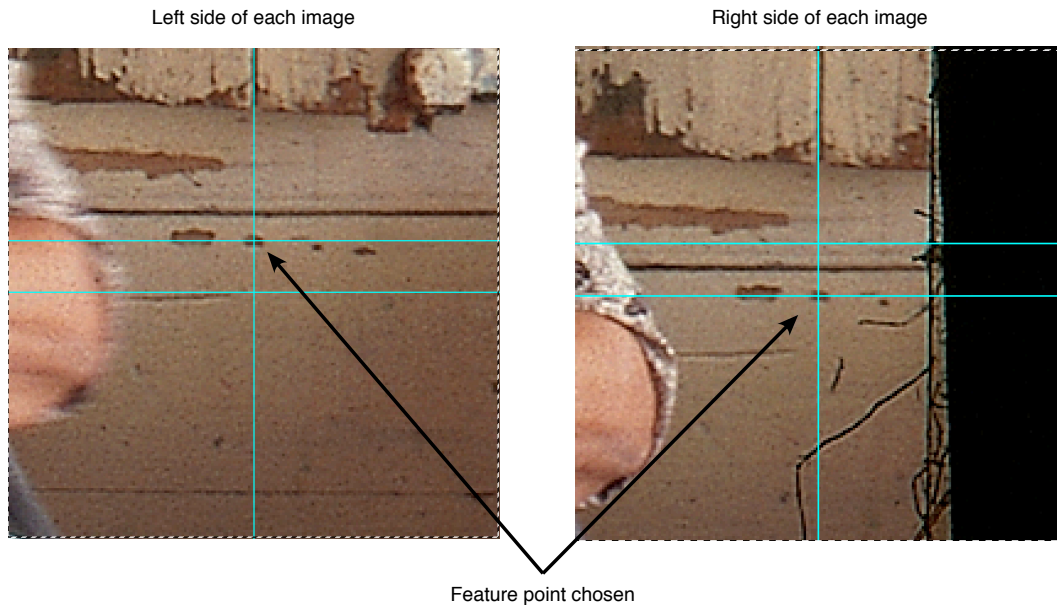
Procedure

1. Both images must be orientated horizontally. For example, some scanning processes align the images vertically.
2. Both images must be the same horizontal and vertical pixel dimension. I suggest finding the largest and adjusting the canvas of the smaller. Note, this is just for the pair, different pairs can be different dimensions.
3. Create the parameter file for this left and right eye pair. This is just a plain text file and it should be kept with the image pairs for possible future reprocessing, for example, to create different sizes, vertical cropping or perhaps higher level of antialiasing for final versions. An example of the two images in PhotoShop with guides on the three points, are shown below, of course one zooms in to accurately locate the feature points. The approach taken by the author is to place guides through each of the chosen points, then using the rectangular select tool and ensuring snapping is enabled, read off the coordinates for each of the 6 points in the info pane.



4. Save images and parameter file. The images should be save in a lossless format, in this case 16 bit tiff and have consistent colour profiles. Saving the images will include the guides for the three points in the parameter file, makes it easier to identify mistakes, or to adjust zero parallax point (if necessary).

Important: once the parameter file is created the input images should not have their dimensions changed or have the panorama strip shifted within the image. If this is honoured then one can change the colours, do spot editing, adjust levels and so on and then simply run the pano_align software again.



The guides have been saved in the sample images provided so the reader can inspect the exact left/right point chosen and the zero parallax point. The zero parallax point for screen based viewing corresponds to the distance that should appear the screen distance. For HMD viewing the zero parallax point should be located at infinity, or at least the most distant object in the scene. It is proposed in this exercise that the zero parallax point is located on a distant object, the exact rotational wrapping shift between the two images will occur in the playback software.

Description of command line options

The usage string, achieved by typing "pano_align" without any parameter file, is as follows.

```
Usage: pano_align [options] parameterfile
Options:
  -a n    antialias level, default: 2
  -e n    width of edge blend zone, default: 100
  -w n    width of final image, default: autodetermined
  -h n    height of the final image, default: autodetermined
  -z      adjust zero parallax, default: on
  -d      debug mode, default: off
```

- a This utility uses simple supersampling antialiasing. For testing purposes one might set this to 1, for final production a value of 3 is about as high as one needs to go. Note that this affects the processing time as the square of the value. So 2 times antialiasing takes 4 times longer than 1 times antialiasing. And 3 times antialiasing takes 2.25 times (9/4) longer than 2 times antialiasing.
- e The width of the edge blend zone. No real need has been found to date to change from the default. It does mean that the feature points used in the parameter file should be at least this value way from the edge of

the panorama scan.

- w The width of the output image. If not specified then the "natural" width will be used which will approximately be the distance between the left-right feature points.
- h The height of the output image. If not specified then the "natural" height will be used which will approximately the height of the input images, and in proportion to the output image width.
- z Whether or not to align the zero parallax point. Not many situations when this should not be done.
- d Simply reports more information during processing. Only really required for debugging in the case of unexpected behaviour.

For the example image and parameter file supplied one might use simply type the following.

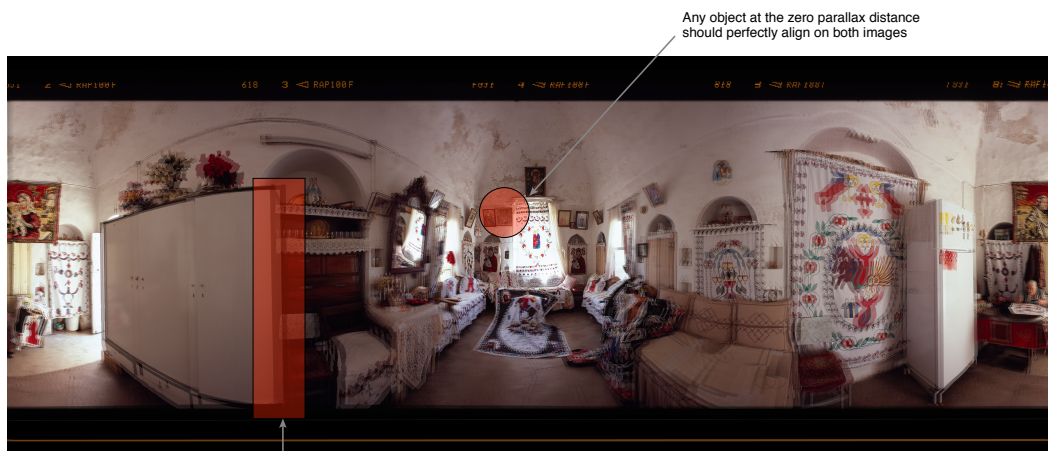
```
pano_align 0477.dat
```

This will result in an image with the same name as the input files but with a "_align" appended (before the .tif extension). It will be about the same width as the distance between the chosen left and right selected feature points, in this example approximately 14,000 pixels.

There are some tests one might do. For example, open one image, select it all, extend the canvas to the right by 200%, paste the image into the right half and check the seam that is now in the center of the image is perfect. If not then it suggests an error in the left and right feature point of the corresponding image.



Another test might be to copy one image and paste it onto the other and set the transparency of the top layer to 50%. One should verify that the parallax is as expected given the left and right pair labelling. One can also slide one layer with respect to the other to investigate the effect of locating the zero parallax distance on different parts of the scene. For example, when presented on a screen based system the zero parallax point will be close to the closest object in the scene.



For objects closer than the zero parallax object, then the right eye projection of the object should be to the left of same object in the left eye.

If there are objects further away than the zero parallax object, then the right eye projection of the object should be to the right of the same object in the left eye.

And lastly, ideally there should be no significant vertical parallax. If there is then this is most likely due to the tripod not being level. Correcting this is an "advanced topic and is a discussion to be had once one is comfortable with the above.



In all cases the projection of the right eye and left eye only results in a horizontal displacement.

Another example might be

```
pano_align -w 16384 -a 3 0477.dat
```

This will produce a high quality conversion and set the output resolution to a power of 2, in case the playback software expects that.

And finally, while the panoramas can easily be cropped in other packages, if one knows in advance exactly what cropping is required then the following will do that.

```
pano_align -w 16384 -h 4096 -a 3 0477.dat
```

Notes

- Avoid choosing points for the parameter file where the vertical of the point may pass through something that might move, like a person or animal. You can see in this example that the woman has moved between the start and end of the capture.
- Have a thought about a consistent file naming scheme so that parameter files are easily matched to their left and right image pairs. Ideally appear in order in an alphabetical listing of the directory.
- The file names cannot contain spaces or other "white" characters.
- Processing time is proportional to the number of pixels in the output images, so a 4K output will take 1/4 the time of an 8K output, and 1/16 the time of an 16K output image.
- It is suggested that a script is written to mass convert files which exposes parameters likely to change (eg: antialiasing, width, height), one then simply reruns this as parameter files are updated, or command line options changed.
- This utility does not preserve the colour profile in the tif files generated. If the colour profile is important then it should be reassigned to the output images, in PhotoShop say.

Source code

The source code as supplied should build out of the box on Mac (xcode installed) and Linux systems. The only dependencies are the standard jpeg and tiff libraries. If these are not in the standard include and lib paths then the -I and -L lines in the simple Makefile supplied will need to be adjusted.