

Instructions for creating remap filters for ffmpeg from fish2persp.

Paul Bourke

March 2021

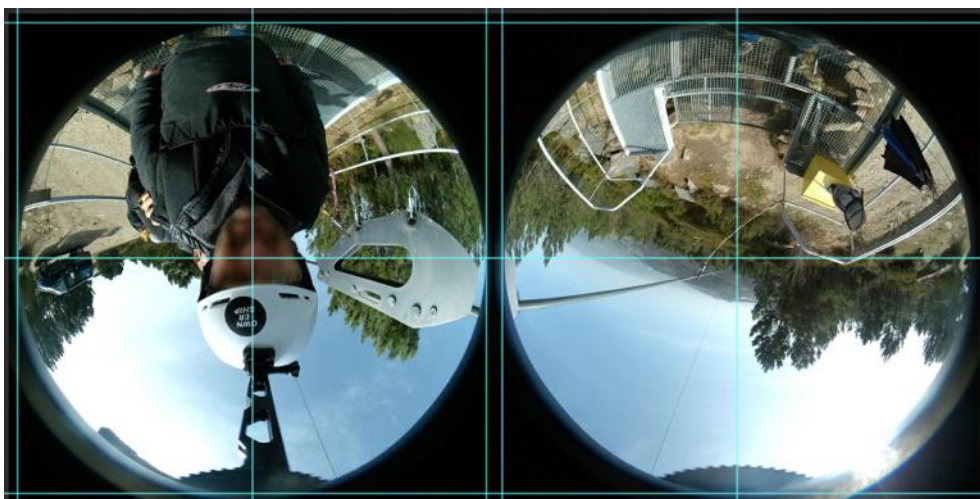
The following describes the step-by-step procedure for creating remap filters for ffmpeg from the authors fish2persp software. It is important to realise that the remap filters encapsulate all aspect of the resulting perspective image extraction. In particular, a different set of filters is required if the fisheye images are located in different locations in the frame or are different resolutions. Additionally, different remap filters are required if a different output perspective resolution or aspect ratio is required.

To create the remap filter one first creates the desired perspective projection using fish2persp, once that is done an additional flag is added to the command line to create the two remap pgm filters.

1. Extract a single representative frame from the video. This will serve to set up all the transformation settings in fish2persp, noting that fish2persp only operates on a single image.



2. Determine the center and radius of each fisheye circle in units of pixels. The author uses PhotoShop and a combination of the circular lasso tool and snapping guides.



In this case the left fisheye center is located at (867,867) with radius of 835 pixels. The right fisheye circle is located at (2593,867) also with a radius of 835. Using fish2persp to extract a perspective view from the left image might be done as follows, assuming the fisheye has a field of view of 190 degrees (-s option)

```
fish2persp -w 1920 -h 1080 -r 835 -c 867 867 -s 190 sampleframe.jpg
```

Resulting in the following



The coordinate system for fish2persp has the z axis up, the y axis into the page and the x axis to the right. One can tilt with the command line -x option, roll with the command line -y option, and pan with the command line -z option. Multiple rotation can be applied, they are executed in the reverse order specified. The rotational command line options are the only ones that are order dependent, all the other options can be any order. So, one can make this upright by a roll (rotate about y axis) of 180 degrees with

```
fish2persp -w 1920 -h 1080 -r 835 -c 867 867 -s 190 -y 180 sampleframe.jpg
```

Giving the following



The default perspective field of view is 100 degrees, one can change this with the -t option. For example the following will extend the FOV to 120 degrees.

```
fish2persp -w 1920 -h 1080 -r 835 -c 867 867 -s 190 -y 180 -t 120 sampleframe.jpg
```



If one wanted to tilt the camera up a little

```
fish2persp -w 1920 -h 1080 -r 835 -c 867 867 -s 190 -x -10 -y 180 -t 120 sampleframe.jpg
```

As pointed out before, the order is reversed so the roll about the y axis is performed first followed by the tilt about the x axis. Resulting in the following



The full list of command line options is displayed by typing `fish2persp` without any arguments.

Usage: `fish2persp [options] fisheyeimage`

Options

```
-w n      perspective image width, default = 1600
-h n      perspective image height, default = 1200
-t n      field of view of perspective (degrees), default = 100
-s n      field of view of fisheye (degrees), default = 180
-c x y    center of the fisheye image, default is center of image
-r n      fisheye radius (horizontal), default is half width of fisheye image
-ry n     fisheye radius (vertical) for anamorphic lens, default is circular fisheye
-x n      tilt angle (degrees), default: 0
-y n      roll angle (degrees), default: 0
-z n      pan angle (degrees), default: 0
-a n      antialiasing level, default = 2
-f        create PGM files for ffmpeg remap filter, default: off
-p n n n n 4th order lens correction, default: off
-d        verbose mode, default: off
```

3. When the desired perspective view is achieved, run the same command with the `-f` option, for example

```
fish2persp -w 1920 -h 1080 -r 835 -c 867 867 -s 190 -x -10 -y 180 -t 120 -f sampleframe.jpg
```

This will create two files called `fish2persp_x.pgm` and `fish2persp_y.pgm`, these should be renamed to something informative, for example `leftcam_120deg_x.pgm` and `leftcam_120deg_y.pgm`.

4. These can then be applied to the original movie using `ffmpeg` as follows

```
ffmpeg -i samplemovie.mp4 -i leftcam_120deg_x.pgm -i leftcam_120deg_y.pgm -lavfi remap -
pix_fmt yuv420p leftcam_persp.mp4
```

and

```
ffmpeg -i samplemovie.mp4 -i rightcam_120deg_x.pgm -i leftcam_120deg_y.pgm -lavfi remap -
pix_fmt yuv420p rightcam_persp.mp4
```