PaNaRuf

User Guideline

Rufat Eyvazli
stu213430@mail.uni-kiel.de
MSc Computer Science
CAU Kiel
June 4, 2020

This user guideline has been provided for the readers/users who may intend to use the software. The user, via this guideline, is informed how he/she can run the software with several properties such as spherical panorama, cylindrical panorama and perspective warping.

Tools

The software has been developed in programming language C++ and the Visual Studio 2019 has been used as an IDE. The user should ensure that the OpenCV library 4.1.0 or any of the greater versions have been installed. Moreover, he/she should enable OpenMP 2.0 since it has also been used in the implementation of the software to parallelize program to reduce the overall run time.

Run Steps

Below, one can step-by-step find how to run the software, depending on chosen warping mode: spherical, cylindrical or perspective.

1. **Same Directory:**

   Firstly, the images that are stitched to create panorama should be in the same folder:

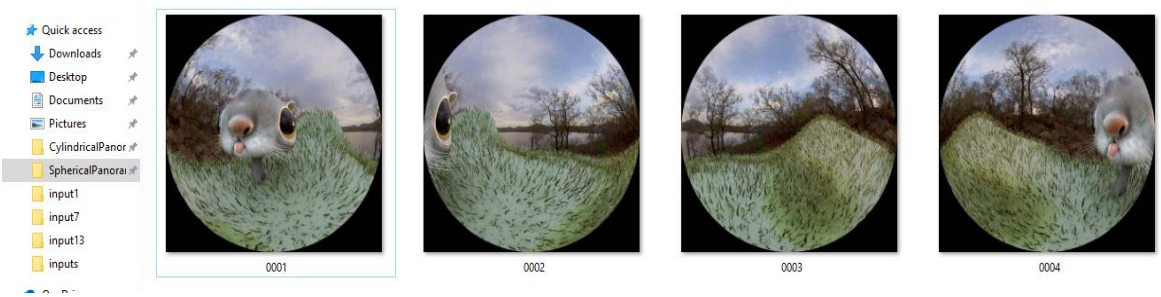

Figure 1. Four fisheye images have been moved to the same folder.

2. Input Text File:

The image names (along with their paths) should be written to a **.txt** file.

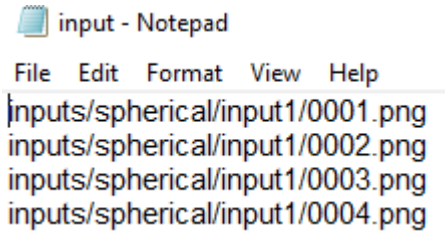

Figure 2 image names with path written to input.txt

As it is shown in Figure 2, the image names along with their paths are written in **input.txt** file. The only thing that the user should know is the location of the **input.txt** file.

Before running the program, the user should consider which mode he/she wants to choose. Below, the list of the modes and the appropriate commands that are used for running them separately, are presented:

- Spherical panorama: **/path/to/input.txt** -s
- Cylindrical panorama: **/path/to/input.txt** -c
- Perspective warping: **/path/to/input.txt** -p

Only in spherical panorama, one should also indicate additional parameters: **hfov** and **vfov,** which are horizontal field of view and vertical field of view respectively.

Then, to run spherical panorama, one should write the following commands:

/pathTo/.txt -s hfov vfov

3. Output:

The **result image** is stored in **/outputs/spherical/result.png**, **/outputs/cylindrical/result.png** or **/outputs/perspective/result.png** depending on the mode chosen by the user.

Below, the user can see the inputs and appropriate outputs for each mode separately.

Spherical Panorama:

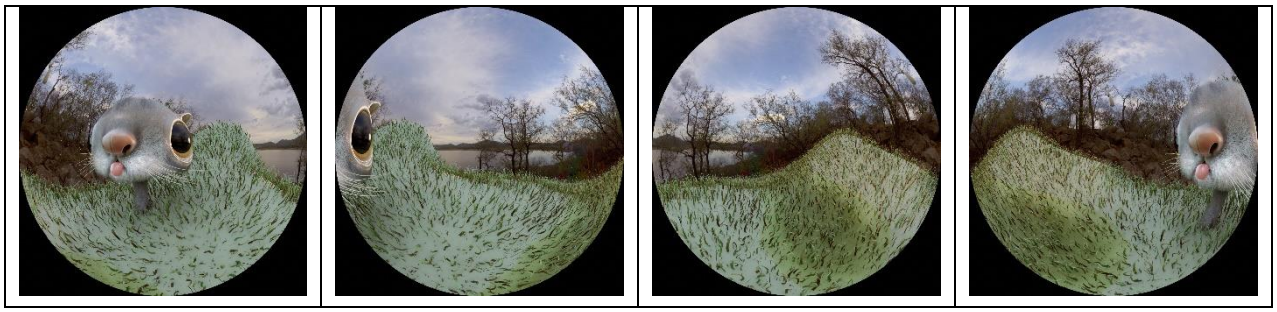

Figure 3. inputs\spherical\input1\input.txt -s 190 190

Result:

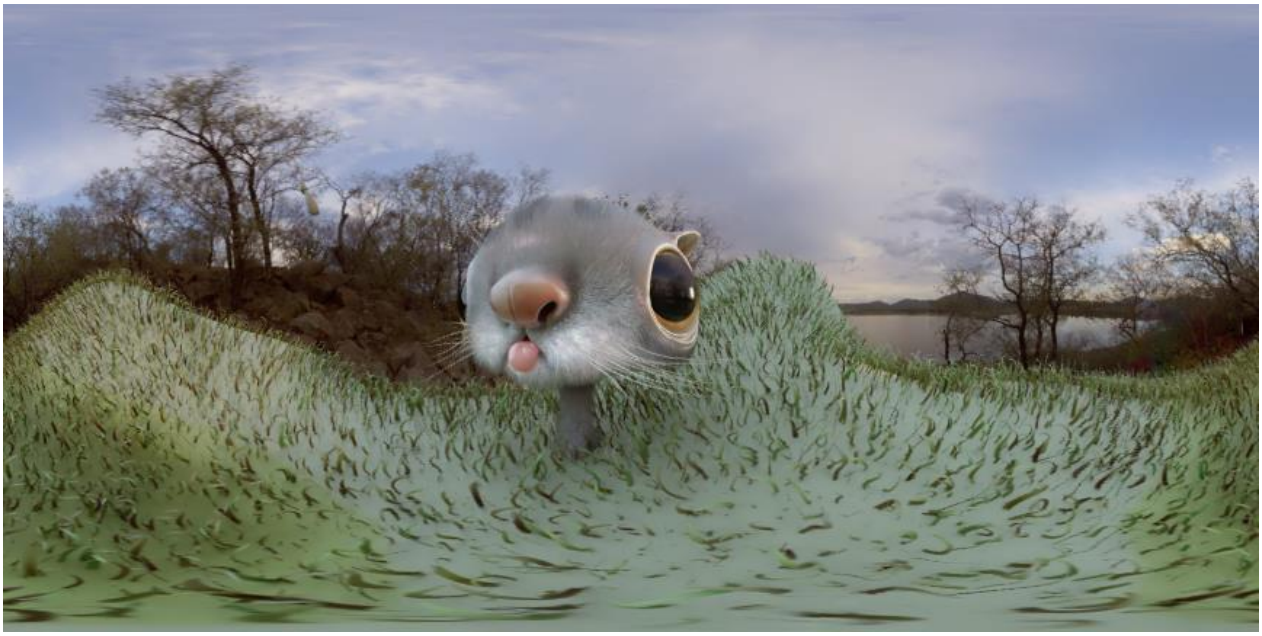

Figure 4. Resolution of the result image is 864x432 which holds property of 360x180 degree equirectangular panorama.

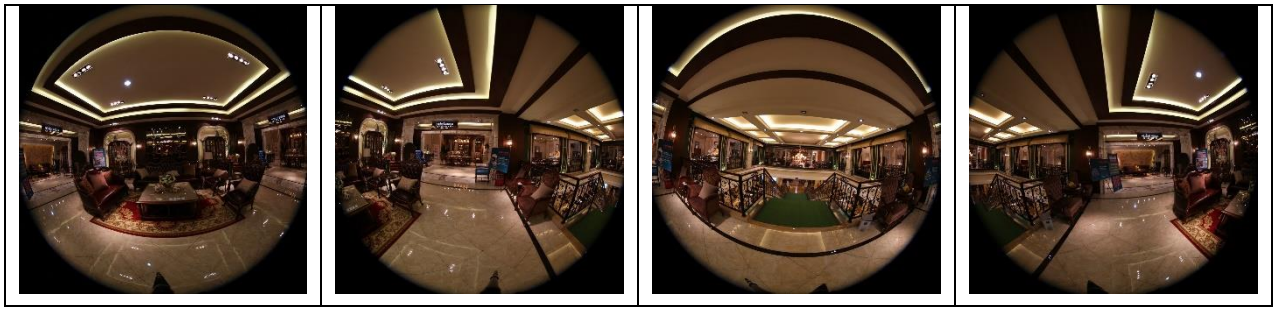

Figure 5. inputs\spherical\input1\input.txt -s 180 180

Result:

Figure 6. Resolution of the result image is 942x471 which holds property of 360x180 degree equirectangular panorama.

After running the program, as mentioned above, it reads multiple fisheye images with greater field of view. In the result, the program gives an equirectangular (spherical) panoramic images with resolution 864x432 which ensures that the software truly holds the aspect ratio (2:1) property of spherical panorama (360x180 degrees).

Cylindrical Panorama

As indicated above, the two important input arguments are required to switch to cylindrical panorama: setting -c and writing **input.txt** file that covers normal images (taken by a pinhole camera model).

Figure 7. inputs\input1\input.txt -c

Result

Generally, it is always better to have camera parameters in order to have better panoramic results. In the implementation of the cylindrical panorama, the only given inputs are the multiple images. Therefore, thanks to Szeliski (p.57), the camera focals are estimated from the homography matrices among the overlapping images. The results look quite acceptable.

Brief Explanation:

- f0 and f1 are derived from H
- for each H, sqrt(f0*f1) is added to focals list
- the list is sorted, and the median value is chosen as the final focal. (it is assumed that all images are taken by the same camera)

Figure 8. inputs\input4\input.txt -c

Result

Perspective Warping

If the user would like to apply perspective mapping to align one image on another image, he/she should simply change -c to -p. The image alignment is handled based on the homography matrices among the images. The input arguments and results for perspective warping are demonstrated below:

Using the same images in Figure 8 as inputs with the following parameter:

inputs\input4\input.txt -p

Result:

It is obviously seen that the image in the left is gone. This is because its edge points are too far (approaches to +,- infinity). Therefore, it is automatically detected by the program itself and removed from the view.

Figure 9. inputs\input4\input.txt -p

Result:

Conclusion

In this guideline, the users/readers are informed about the functionalities of the software: which tools they need and which steps they should follow to run the program. Firstly, visual explanations are provided. Then, the users are equipped with the corresponding input arguments to choose, depending on the modes: spherical, cylindrical or perspective. Additionally, they are informed about some technical details in brief. Finally, inputs and outputs are demonstrated respectively so that the users can see the quality of our software.