

Computer Vision Homework 4 README

Jiangtian (Joey) Hou, jh4170

Challenge 1a worked fine. I was planning to use `ginput` to get the four corners of two images first, and then hardcode the coordinates into the code. However, finally the `ginput` was remained.

Challenge 1b worked fine. It just worked fine.

Challenge 1c worked fine. I set the error to be 5 by Euclidean distance and the iteration is 100. The result looks just like the one shown in the instruction.

Challenge 1d worked fine. I coded the weighted blending by myself without using `bwdist` (because I didn't figure out the correct usage of it). Since there will be two directions of blending (vertical, horizontal), I further wrote a function called `blendImagePairVertical`, which is included in Challenge 1e (I think).

Challenge 1e worked fine. I used the function in challenge 1b to get the mask of an image, and I wrote a function to get the proper transformation function for shifting (so that the warped image will be remained on canvas). The function also included outputs of the size of the warped image. After that, I coded the algorithm to take the middle image (or the middle two images) to be centered and be warped as little as possible. In this way, the final output looks like the curved screen of a cinema (instead of totally skewed to one side of it by just using a for loop for all images to be warped and blended).

Since there are two directions (vertical and horizontal), I introduced several helper functions as well as two versions of the same implementation with several variables changed.

To get rid of the black sides (useless area), I only implemented a very slow delete-by-row/column method to check and delete the black rows at the bottom/black columns at the right side. Because of this, the program could be a little slow and it took around 40s to produce the output for 3 pictures of mountains.

Challenge 1f worked fine. I took 4 pictures in a school and used the vertical blend mode to blend them together (to show that I have taken odd/even number of photos into mind).

I used some code from `demoMATLABTricksFun.m` and included the function `saveAnnotatedImg()`.

All files, original or generated, are included in the zip file.