**Computer Vision HW4 Report**

Student ID: R10625016

Name: 許致銓

**Visualize the disparity map of 4 testing images.**

|  |  |
| --- | --- |
| Tsukuba | Venus |
|  |  |
| Teddy | Cones |
|  |  |

**Report the bad pixel ratio of 2 testing images with given ground truth (Tsukuba/Teddy).**

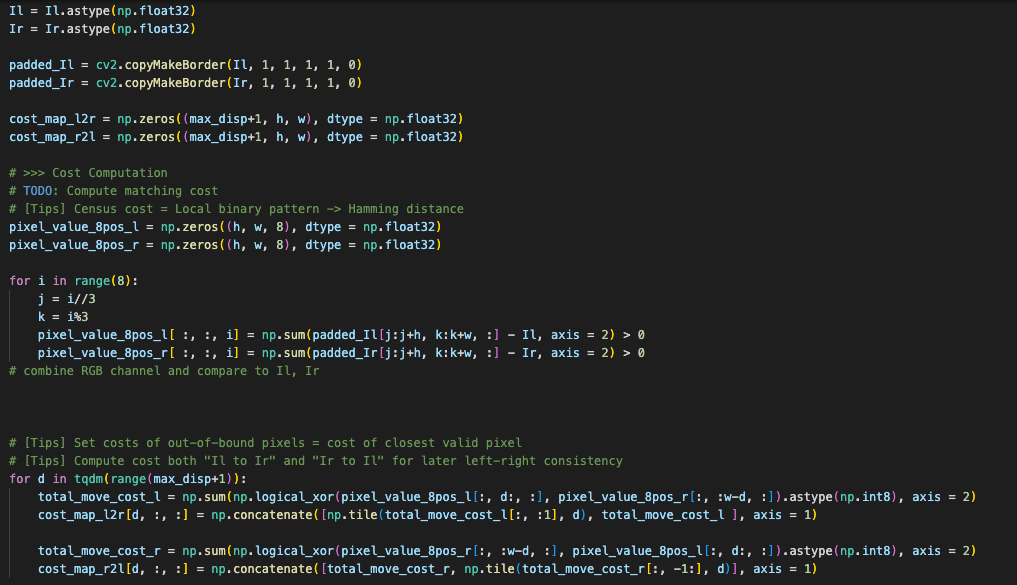
|  |  |
| --- | --- |
|  | bad pixel ratio |
| Tsukuba | 4.99% |
| Teddy | 11.10% |

**Describe your algorithm in terms of 4-step pipeline.**

1. **Cost Computation**

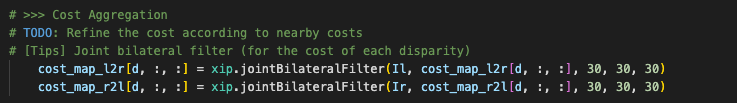
To begin with, I padded the picture with 0 for 4 slides.

For every pixel, I need to calculate the 8 surrounding pixels’ difference of value, so I calculate for three channels, labeling it if the difference is > 0 or < 0, and piled 8 pictures with size h\*w to a h\*w\*8 matrix. Moreover, I did the calculation both for “left to right” and “right to left”. In this way, I can calculate faster in later operations and have a robust disparity optimization. After that, I calculate the cost corresponding to different disparity with xor operations and finish my cost matrix. Due to the disparity, some pixels that didn’t correspond to other pixel in the other picture can’t be calculated. As a result, I just filled the spot by the border cost value.



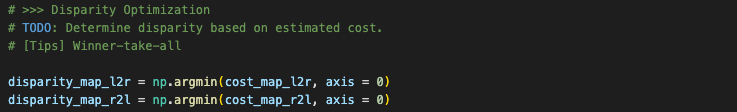
1. **Cost Aggregation**

I did the Joint Bilateral Filter on my cost matrix to refine the cost by neighbors.



1. **Disparity Optimization**

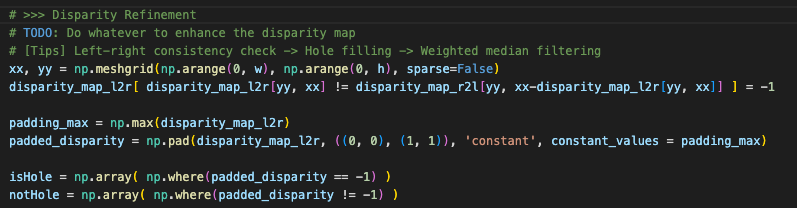
Winner takes all. I find minimum cost and use the corresponding disparity to construct my disparity map.



1. **Disparity Refinement**
   * 1. **Left-right consistency**



Use the given formula to check the consistency. If the corresponding disparities are not the same, I will assign -1 to that pixel.

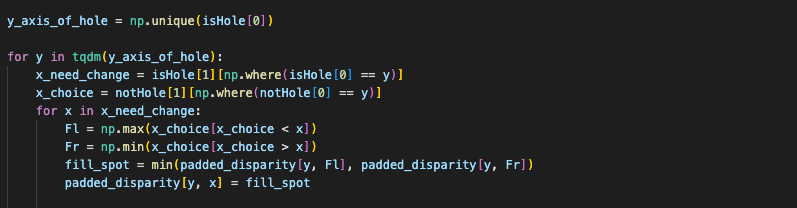


* + 1. **Hole filling**

First, I padded the left and right disparity for 1 pixel with the maximum disparity in this picture.

I use two “for” loops to deal with the “-1 hole” problems, one is for looping over the pixels’ y axis, the other is for the x values whose disparity is -1. In this way, I can trace the nearest pixels that are not -1 and find the value of smaller disparity. Then, fill the hole with this smaller disparity.

After the two for loops, I can extract the non-padded part of h\*w picture.



* + 1. **Weighted median filter**

Do the weighted median filter with “xip” library. The source is the original picture, and the destination is the picture I processed above.

