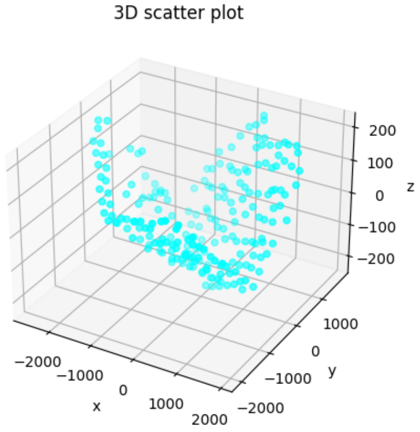
CS543/ECE549 Assignment 5

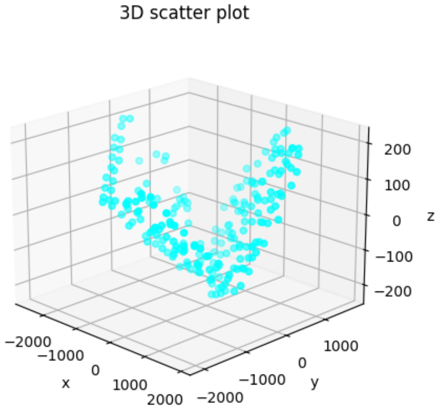
Name: Hongbo Zheng
NetId: hongboz2

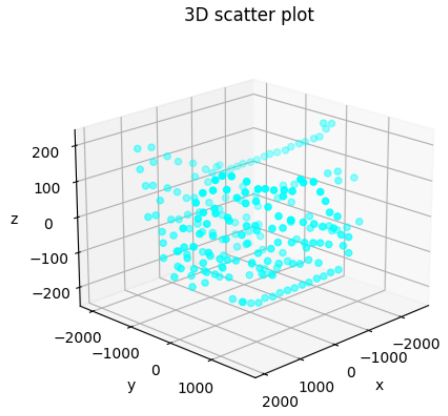
Part 1: Affine factorization

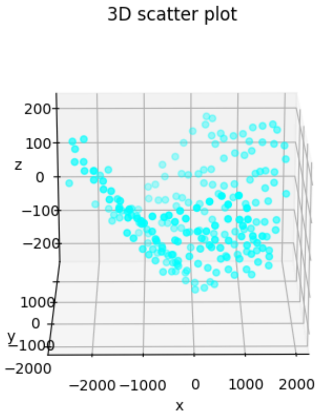
A: Display the 3D structure (you may want to include snapshots from several viewpoints to show the structure clearly). Report the Q matrix you found to eliminate the affine ambiguity. Discuss whether or not the reconstruction has an ambiguity.

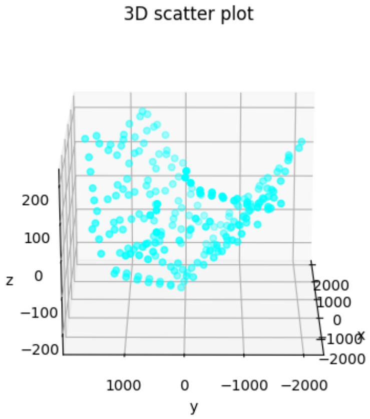
According to the 3D structure plots and Q matrix shown below, the Q matrix is (almost) diagonal, it implies that there is a consistent scaling factor along each axis which indicates that the reconstruction is not affected by ambiguity in terms of scaling.







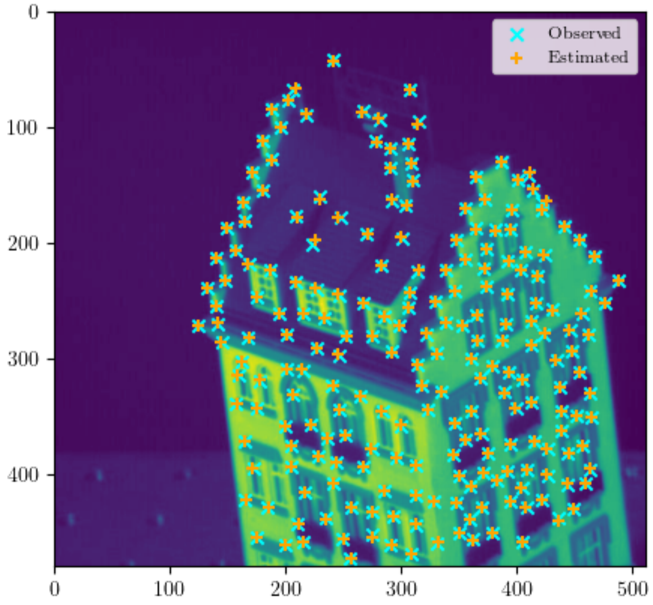




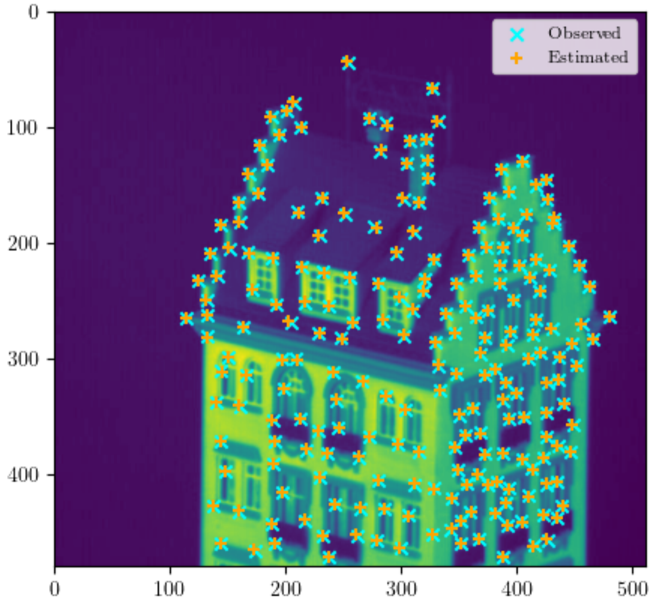
```
Q matrix
[[ 7.94778741e-03  1.19575961e-18  2.73207678e-19]
 [ 0.00000000e+00  8.53955215e-03 -3.98386123e-18]
 [ 0.00000000e+00  0.00000000e+00  2.53757864e-02]]
```

B: Display three frames with both the observed feature points and the estimated projected 3D points overlayed.

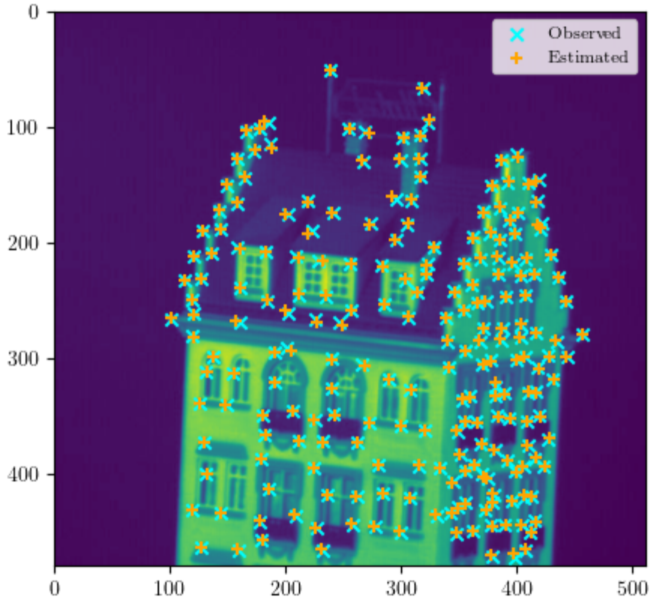
frame 35



frame 70



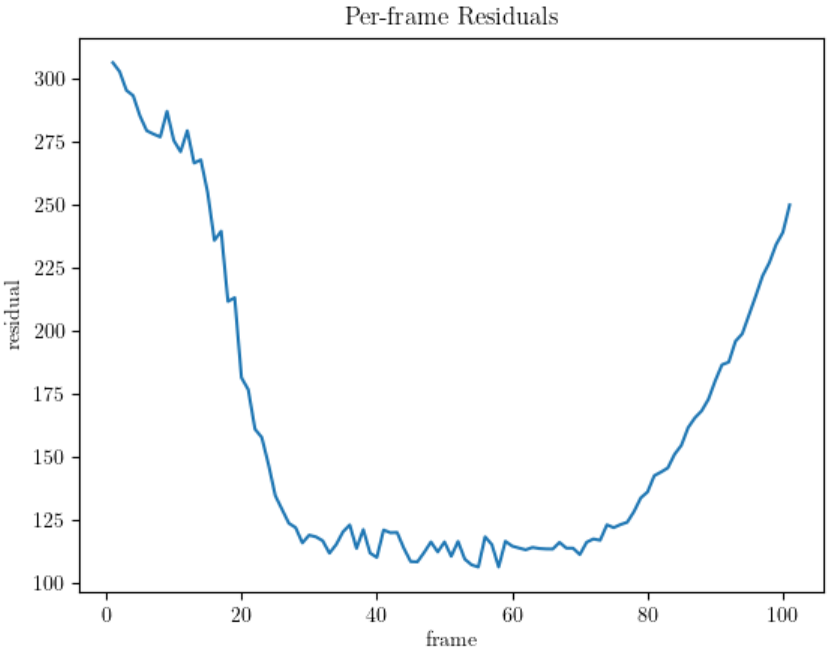
frame 100



C: Report your total residual (sum of squared Euclidean distances, in pixels, between the observed and the reprojected features) over all the frames and plot the per-frame residual as a function of the frame number.

Total residual: 16428.332063

Per-frame Residual



Part 2: Binocular stereo

A: Display best output disparity maps for both pairs.

image: tsukubai

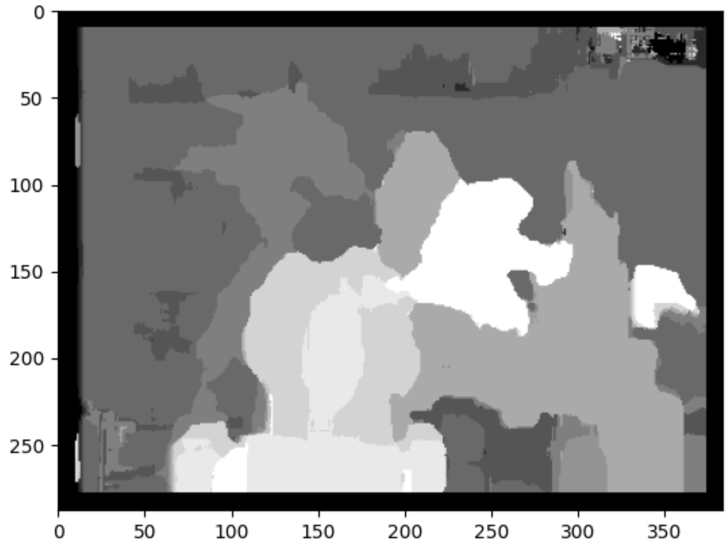
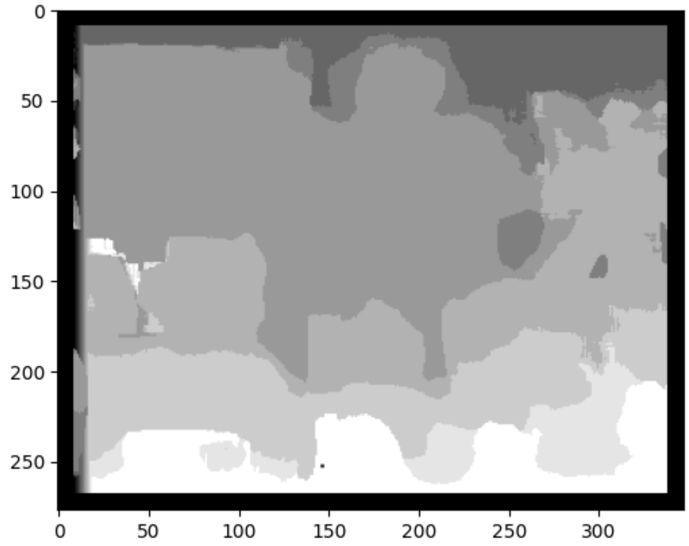


image: moebius

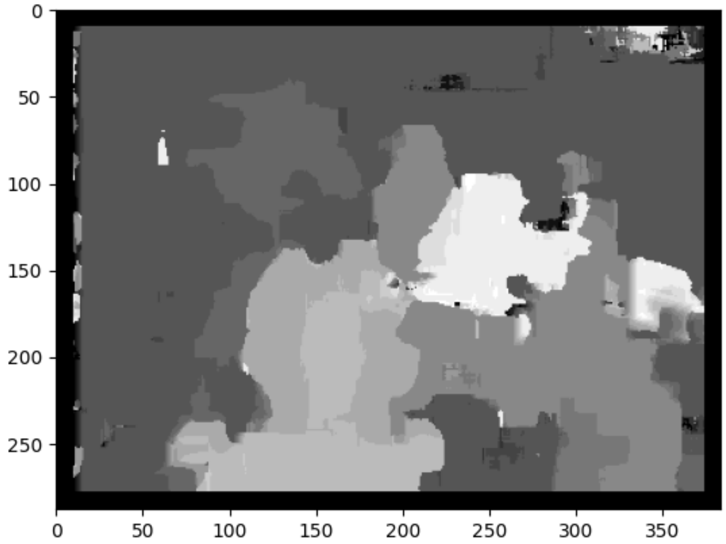


B: Study of implementation parameters:

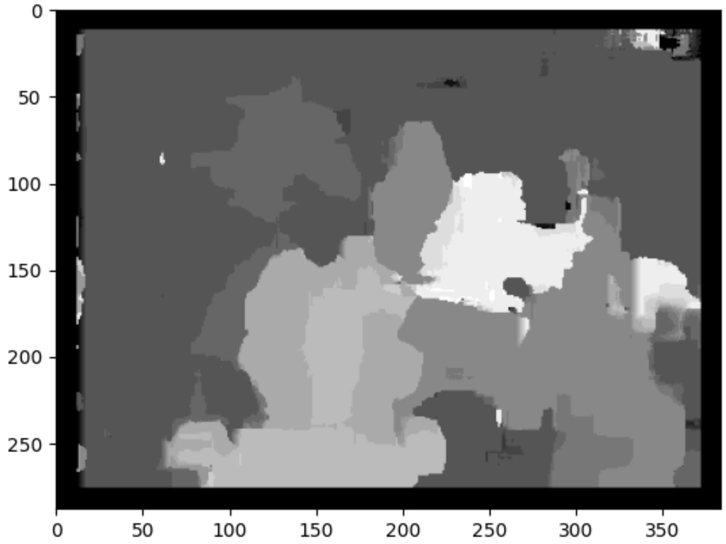
1. **Search window size:** show disparity maps for several window sizes and discuss which window size works the best (or what are the tradeoffs between using different window sizes). How does the running time depend on window size?

   image: tsukuba, algorithm: NCC (since SAD can't tell the run-time difference), offset = 12

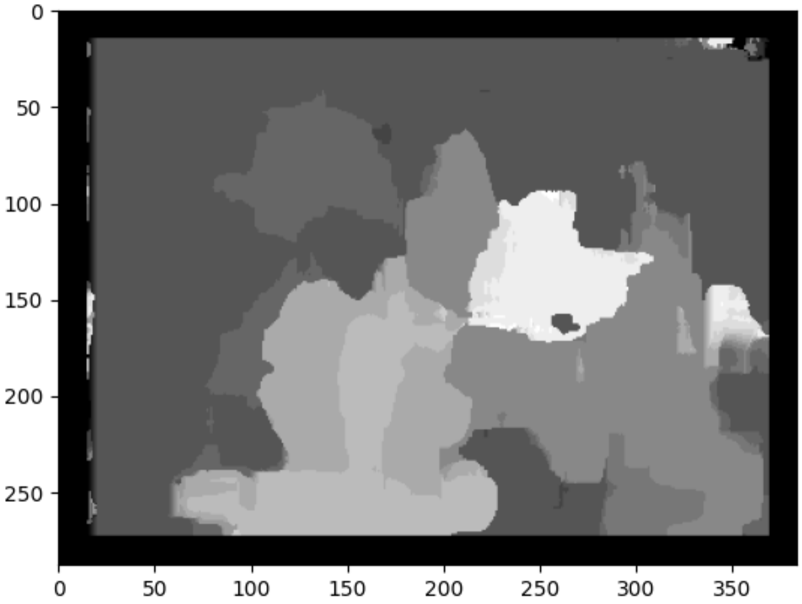
   window size = 20, run-time = 89.68371987342834s



   window size = 25, run-time = 86.82891201972961s



window size = 30, run-time = 83.08237385749817s



I initially employed the SAD algorithm with window sizes of 15, 20, and 25, with an offset of 12. The run-times for each of them are: 18.40703511238098s, 18.47301697731018s, 19.911337852478027s. However, the SAD algorithm's rapid computation speed made it challenging to find the trend in processing time as I increased the window size. Therefore, I switched to the NCC algorithm.

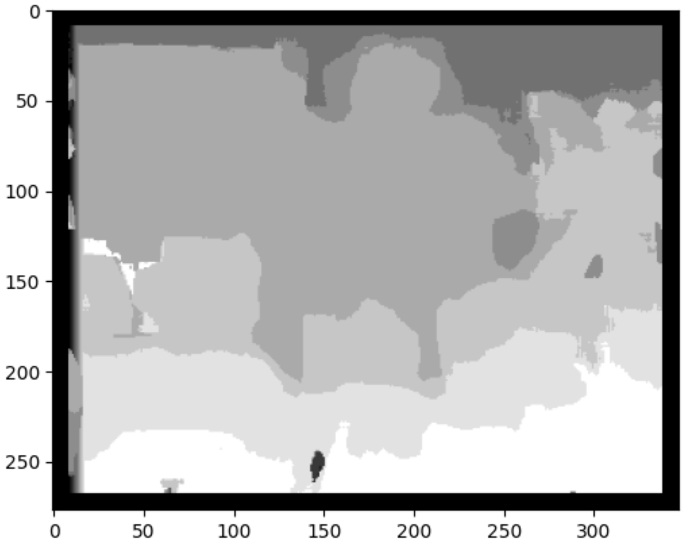
The disparity map obtained with a window size of 25 and an offset of 15 using NCC yielded best results, particularly in enhancing boundaries between distinct objects. Additionally, the run-time exhibits an increase as the window size decreases.

2. **Disparity range:** what is the range of the scanline in the second image that should be traversed in order to find a match for a given location in the first image? Examine the stereo pair to determine what is the maximum disparity value that makes sense, where to start the search on the scanline, and which direction to search in. Report which settings you ended up using.

   image: tsukubai, algorithm: SAD, window size = 18

   10 ≤ offset ≤ 18

   offset = 9



   The setting I ended up using is window size = 18 and offset = 12. The scan range should be within the window size. The result is getting worse, objects start to blend together, when offset is less than 9 and getting dark when offset is larger than window size, therefore the range should be 9 to 18. The direction to search in is from left to right.

3. **Matching function:** try sum of squared differences (SSD), sum of absolute differences (SAD), and normalized correlation. Show the output disparity maps for each. Discuss whether there is any difference between using these functions, both in terms of quality of the results and in terms of running time.

   image: tsukubai, window size = 18, offset = 12

   algorithm: SSD, run-time = 18.157890796661377s

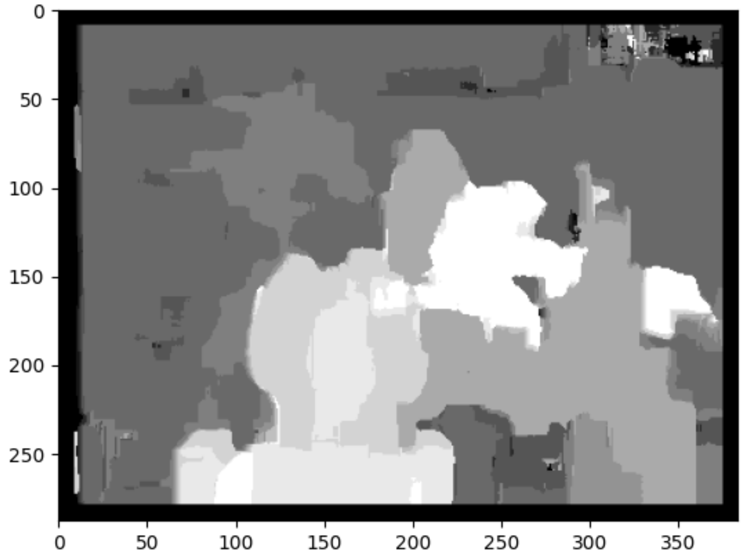


   image: tsukubai, algorithm: SAD, run-time = 18.40703511238098s

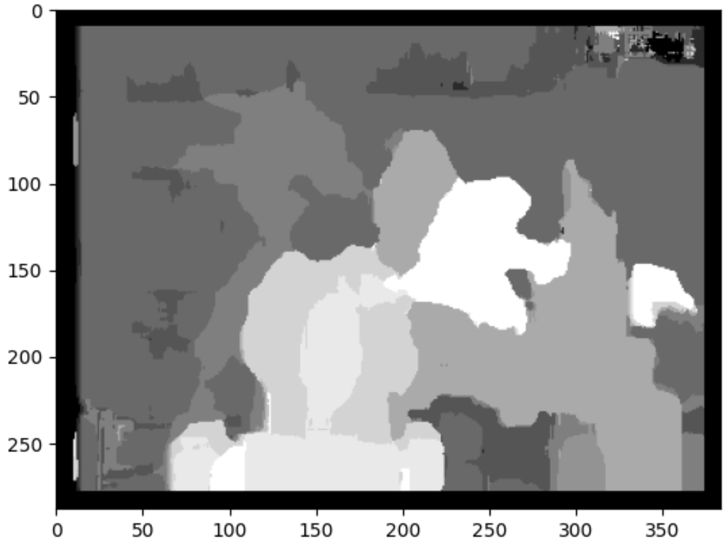


image: tsukubai, algorithm: NCC, run-time = 75.72297215461731s

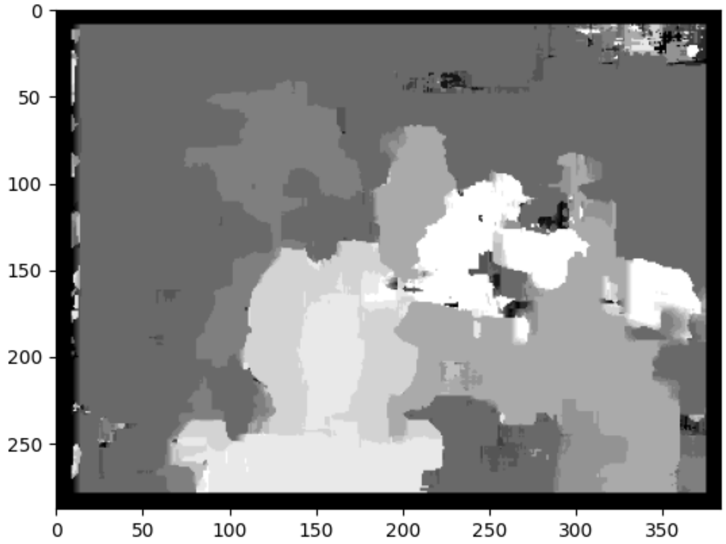
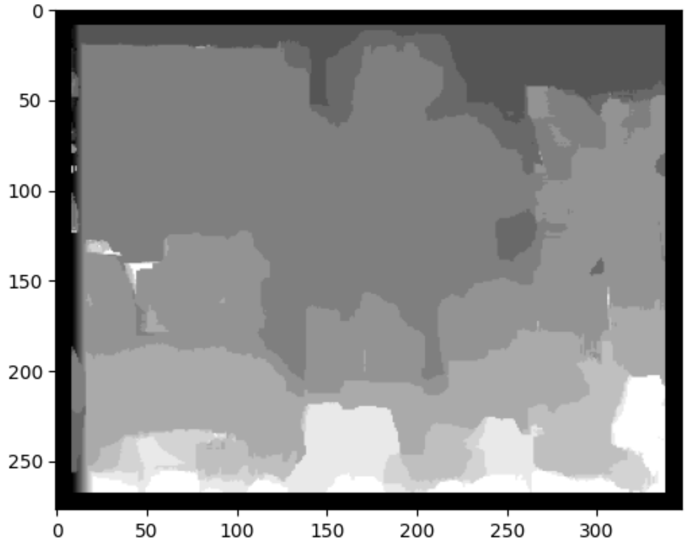
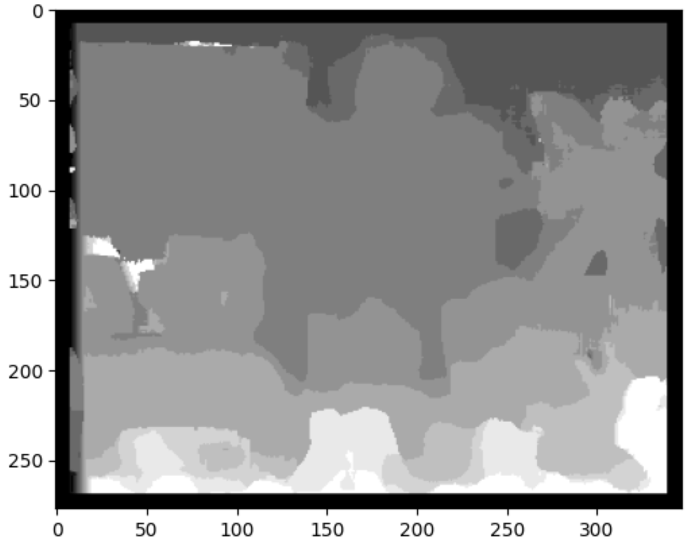


image: tsukubai, window size = 18, offset = 12

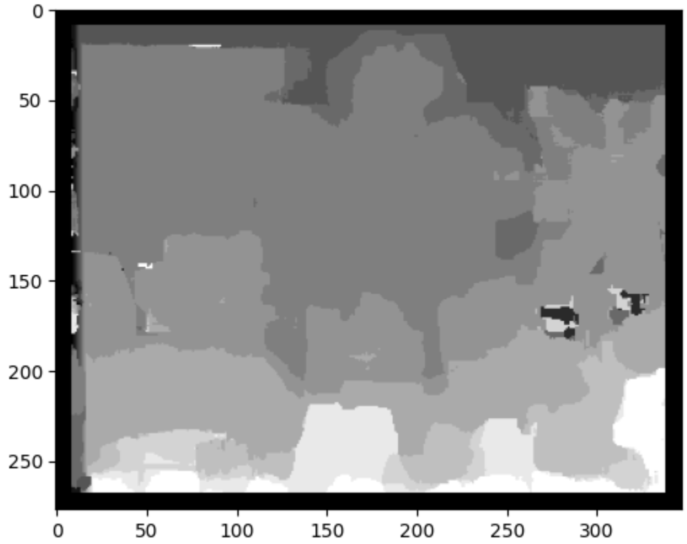
algorithm: SSD, run-time = 16.91584300994873s



algorithm: SAD, run-time = 16.38619089126587s



algorithm: NCC, run-time = 64.34889602661133s



Overall, SAD generates the best results for tsukubai image, and has a very short run-time for both images. However, for mobius image, SAD only generates slightly better result than SSD and NCC. The run-time of NCC is several times slower than SSD and SAD, and the run-time of SAD is similar to that of SSD.

C: Discuss the shortcomings of your algorithm. Where do the estimated disparity maps look good, and where do they look bad? What would be required to produce better results? Also discuss the running time of your approach and what might be needed to make stereo run faster.

The algorithm performs well in regions with consistent textures and clear disparities, yielding accurate and reliable disparity maps. At the areas where are textureless or occluded, the correspondences are ambiguous. The disparity maps may exhibit inaccuracies around object boundaries or in low-texture regions. SSD and SAD are algorithms that may struggle in the presence of noise or variations in lightning.

To produce better results, we can use interpolation, segmentation, or more advanced matching algorithms. Additionally, the window size can also be dynamically adjusted base on different areas of the image. Moreover, we can also implement postprocessing techniques such as filtering to refine the disparity maps.

The main problem of the algorithm is it contains nested for loop. We can perform vectorization and parallelization to speed up computations. Furthermore, utilizing GPUs can also improve the computing speed.

Part 3: Extra Credit

Post any extra credit for parts 1 or 2 here. Don’t forget to include references, an explanation, and outputs to receive credit. Refer to the assignment for suggested outputs.