# CS543/ECE549 Assignment 3

**Name:** Hanliang Jiang

**NetId:** hj33

**Part 1: Homography estimation**

**A: Describe your solution, including any interesting parameters or implementation choices for feature extraction, putative matching, RANSAC, etc.**

For the first part, which is choosing keypoints, I implemented both SIFT and harris corner detector provided by start code. It turned out that when using SIFT, I can get a very good fitting after ransac; however, using the given start code can only work well with a threshold near threshold = 70. When experimenting with RANSAC, the effect doesn’t improve much after increasing iteration numbers of random sampling.

**B: For the image pair provided, report the number of homography inliers and the average residual for the inliers. Also, display the locations of inlier matches in both images.**

number of inliers: 13

average residual: 0.027222486063155603

**C: Display the final result of your stitching.**

电脑萤幕画面

中度可信度

**Part 2: Shape from shading**

**A: Estimate the albedo and surface normals**

1. Insert the albedo image of your test image here:



1. What implementation choices did you make? How did it affect the quality and speed of your solution?

I choose to use the method in the lecture, which is utilizing the first return value given by np.linalg.lstsq as the sum of (albedo\*[normals]), and calculate albedo with Euclid norm, the divide sum by albedo to get normals.

1. What are some artifacts and/or limitations of your implementation, and what are possible reasons for them?

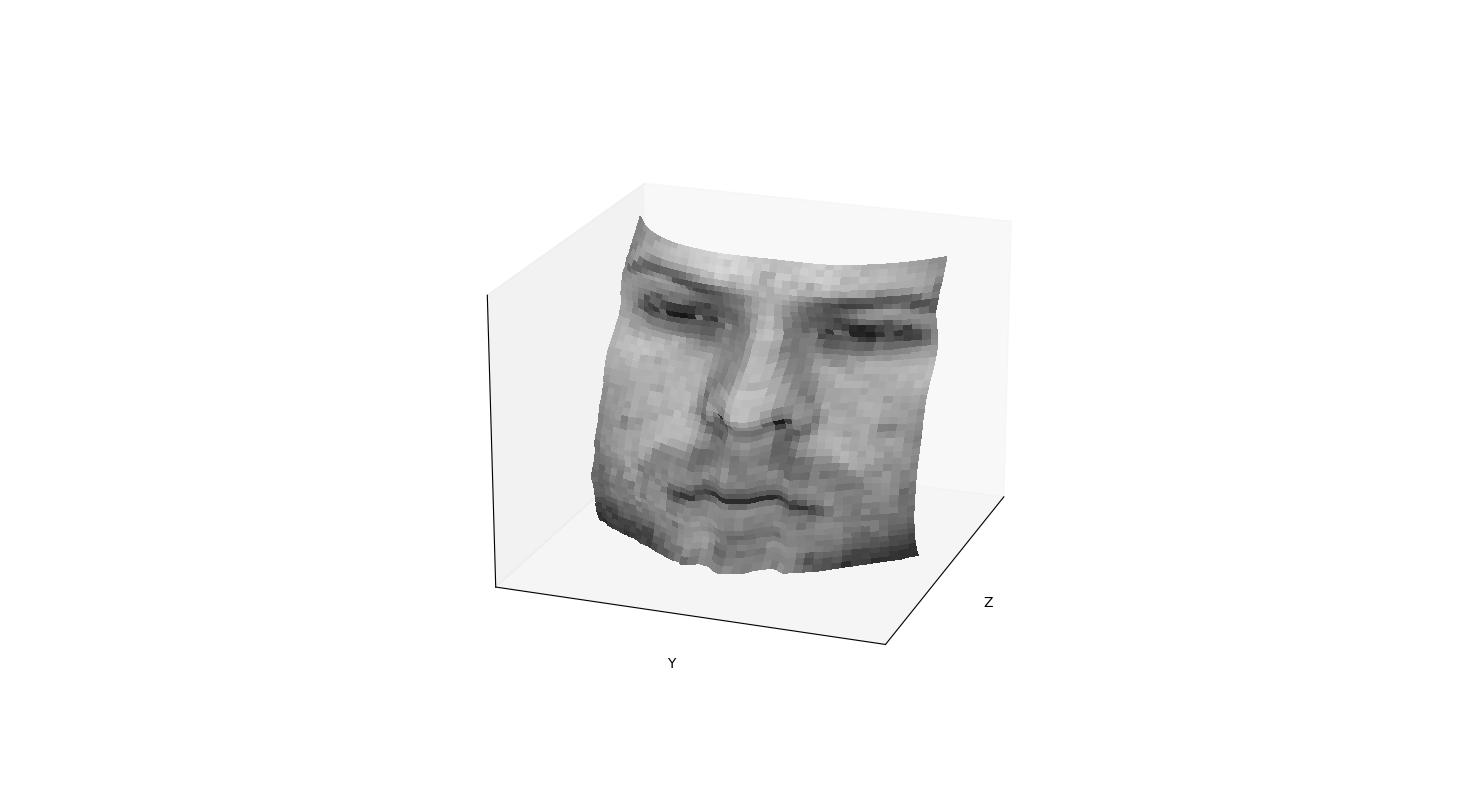
For simplicity, I chose np.linalg.lstsq instead of implementing my own linear system, which means the result could possibly be inaccurate.

1. Display the surface normal estimation images below:

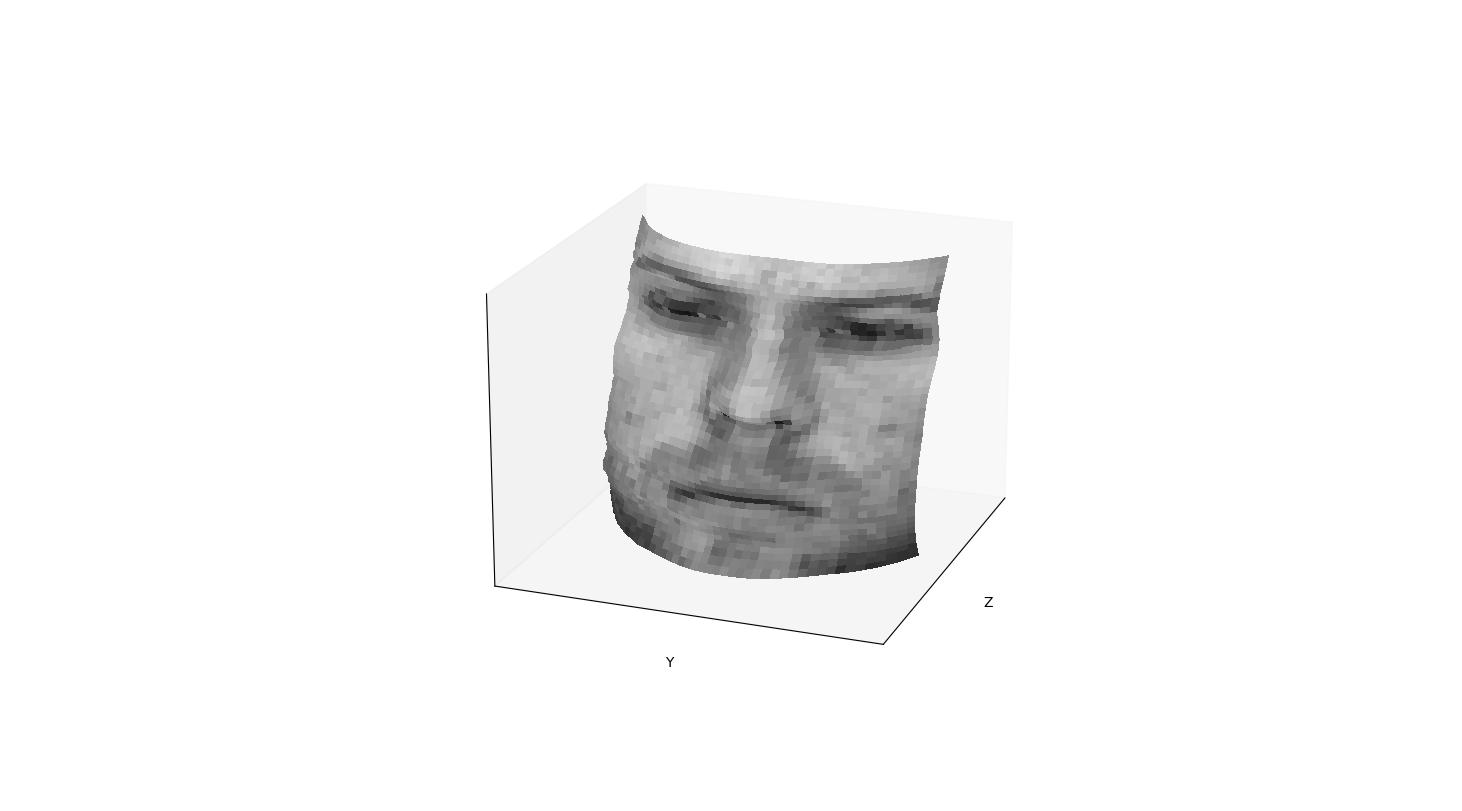


**B: Compute Height Map**

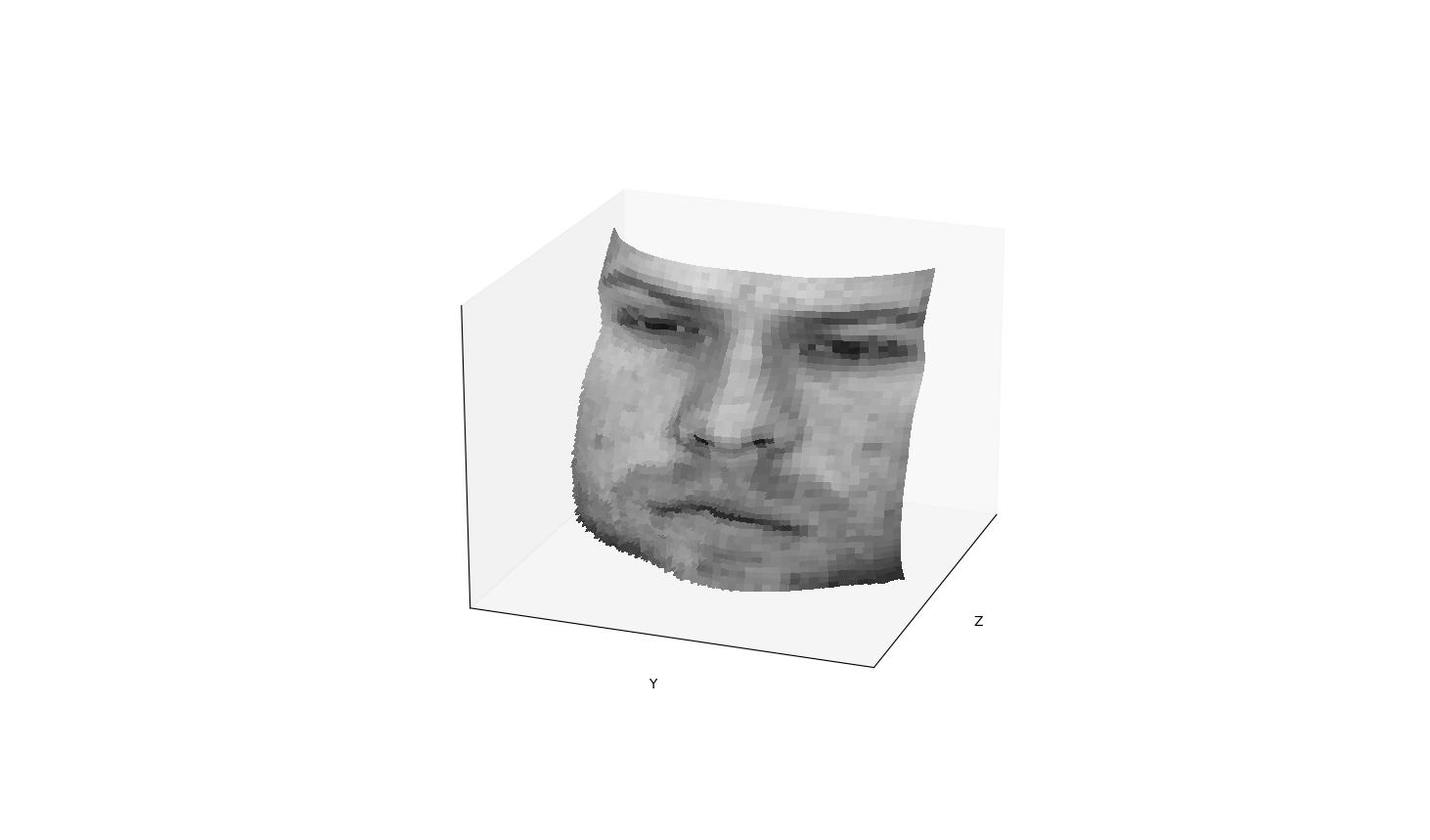
1. For every subject, display the surface height map by integration. Select one subject, list height map images computed using different integration method and from different views; for other subjects, only from different views, using the method that you think performs best. When inserting results images into your report, you should resize/compress them appropriately to keep the file size manageable -- but make sure that the correctness and quality of your output can be clearly and easily judged.



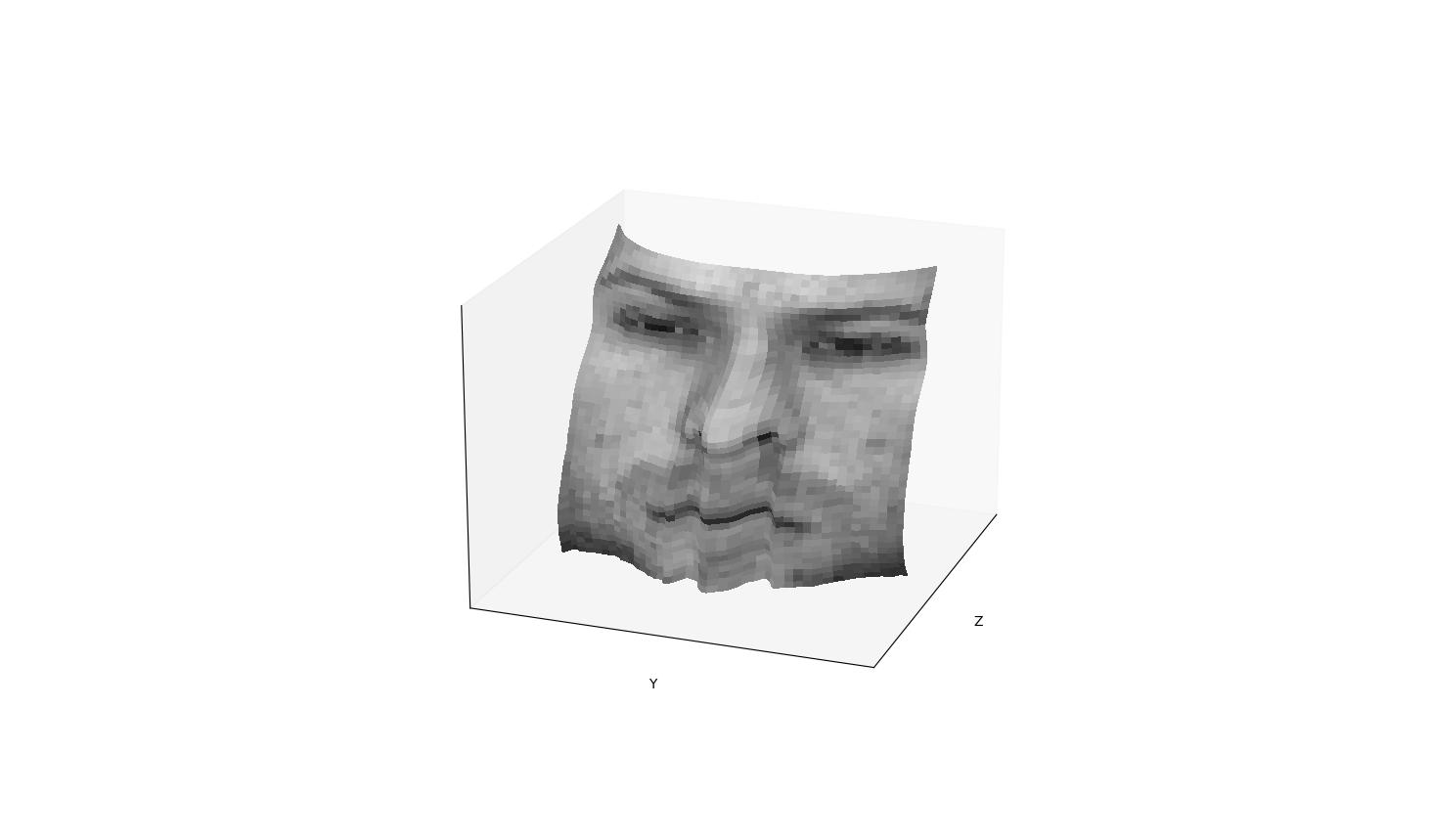
average



column

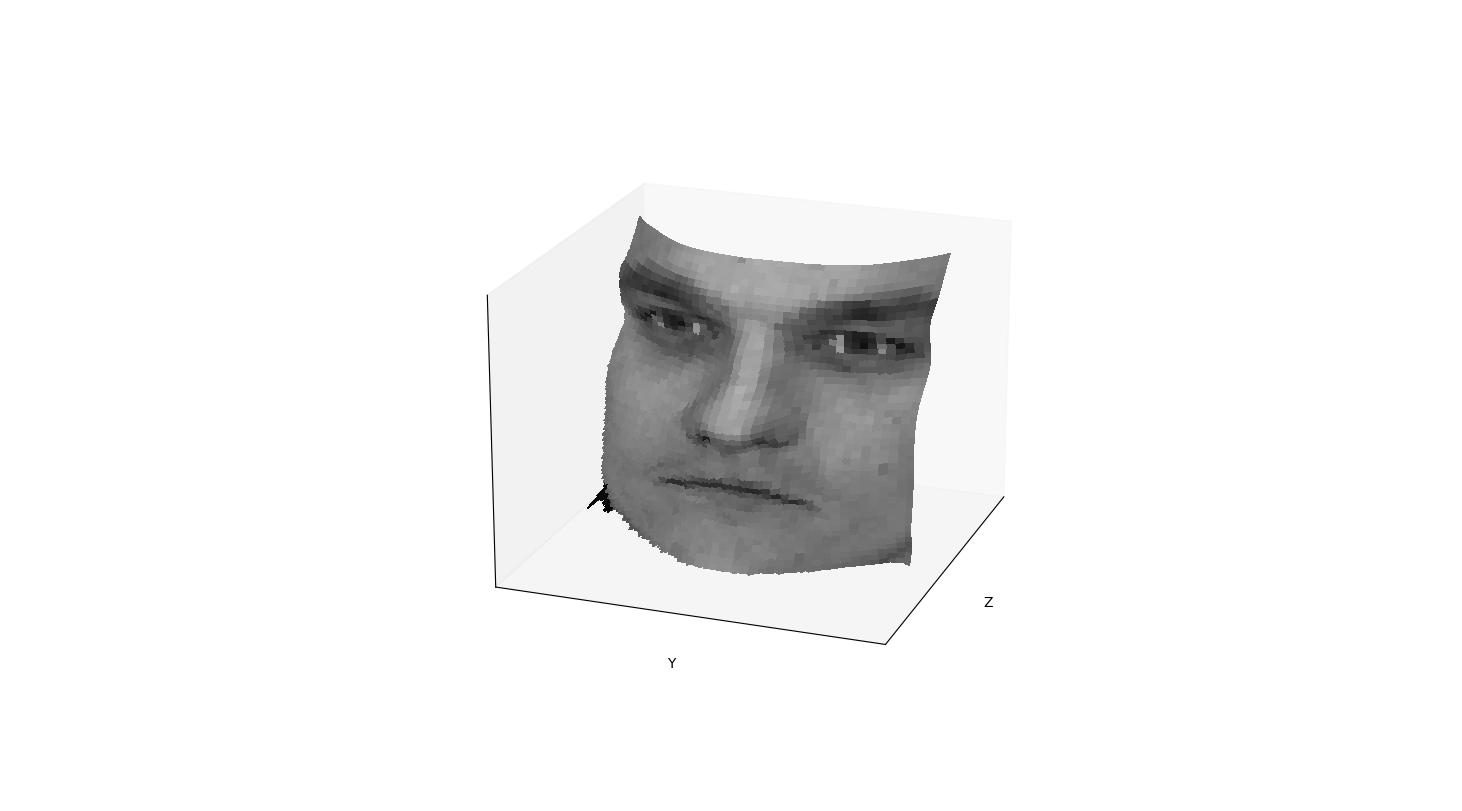
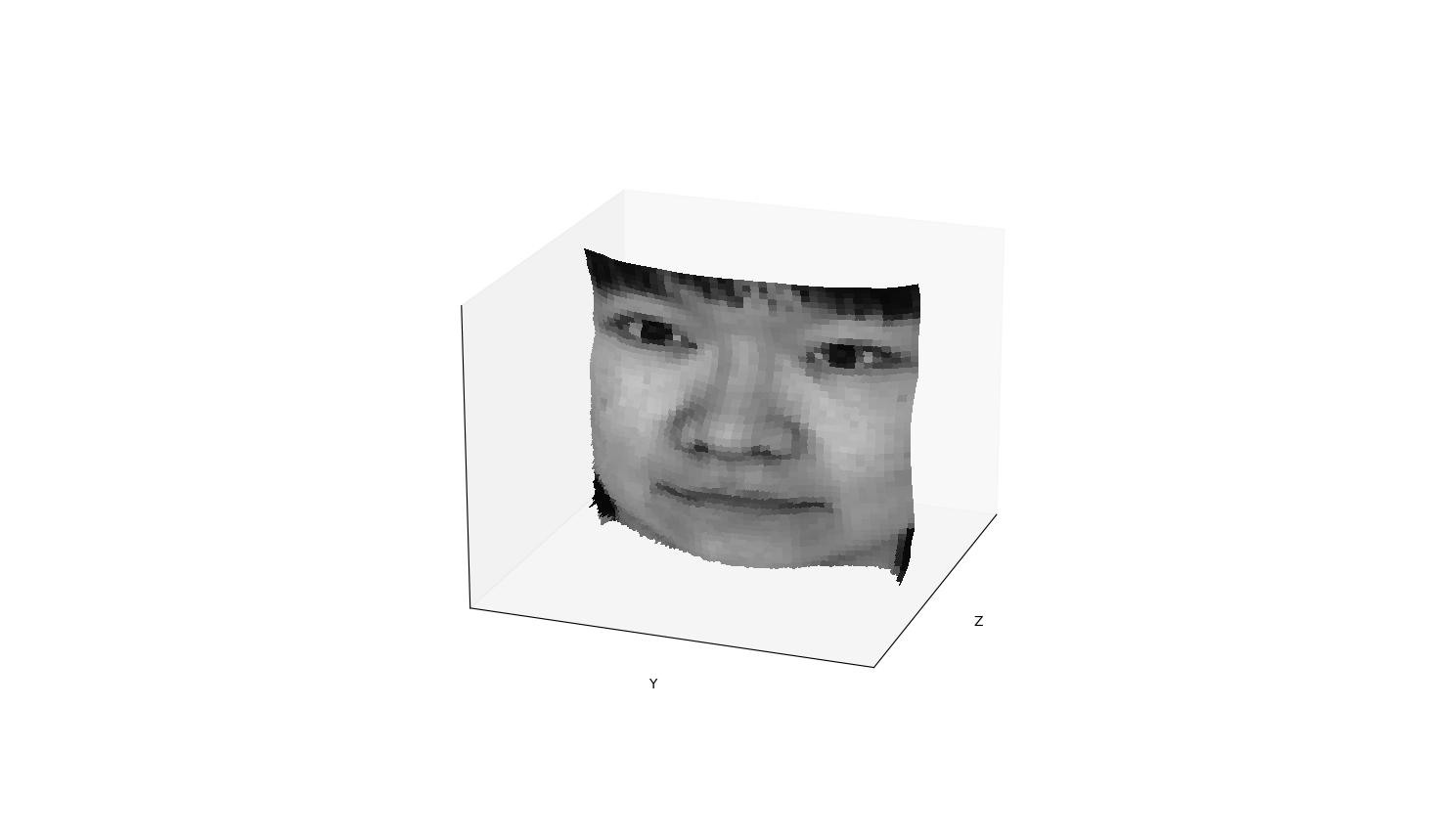
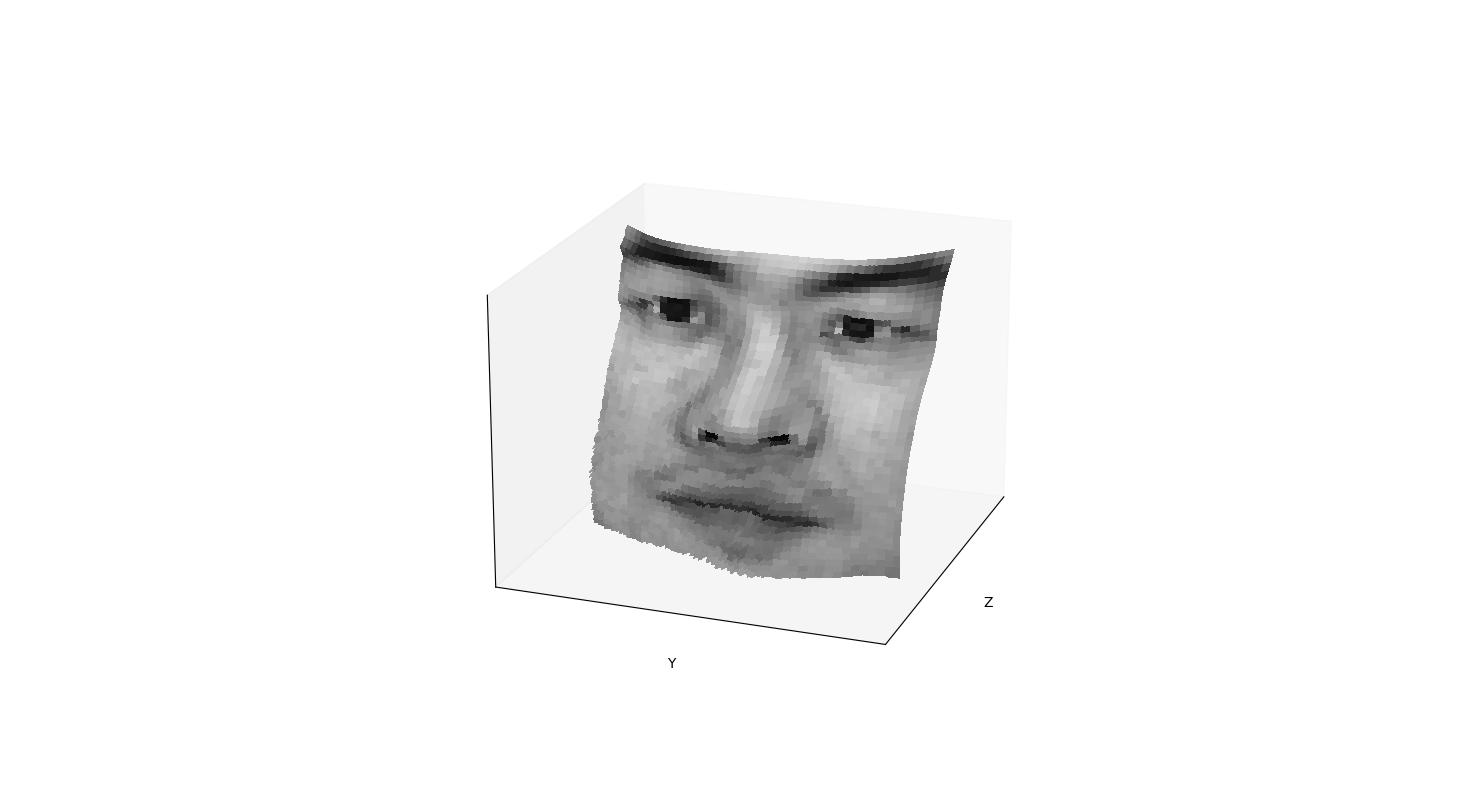


random



Row

Images listed below use random method



1. Which integration method produces the best result and why?

The random method produces the best. Because each pixel take use of many paths, and the result for each pixel could be more accurate

1. Compare the average execution time (only on your selected subject, “average” here means you should repeat the execution for several times to reduce random error) with each integration method, and analyze the cause of what you’ve observed:

|  |  |
| --- | --- |
| Integration method | Execution time(seconds) |
| random | 9.52788233757019 |
| average | 0.04766035079956055 |
| row | 0.04363751411437988 |
| column | 0.046926259994506836 |

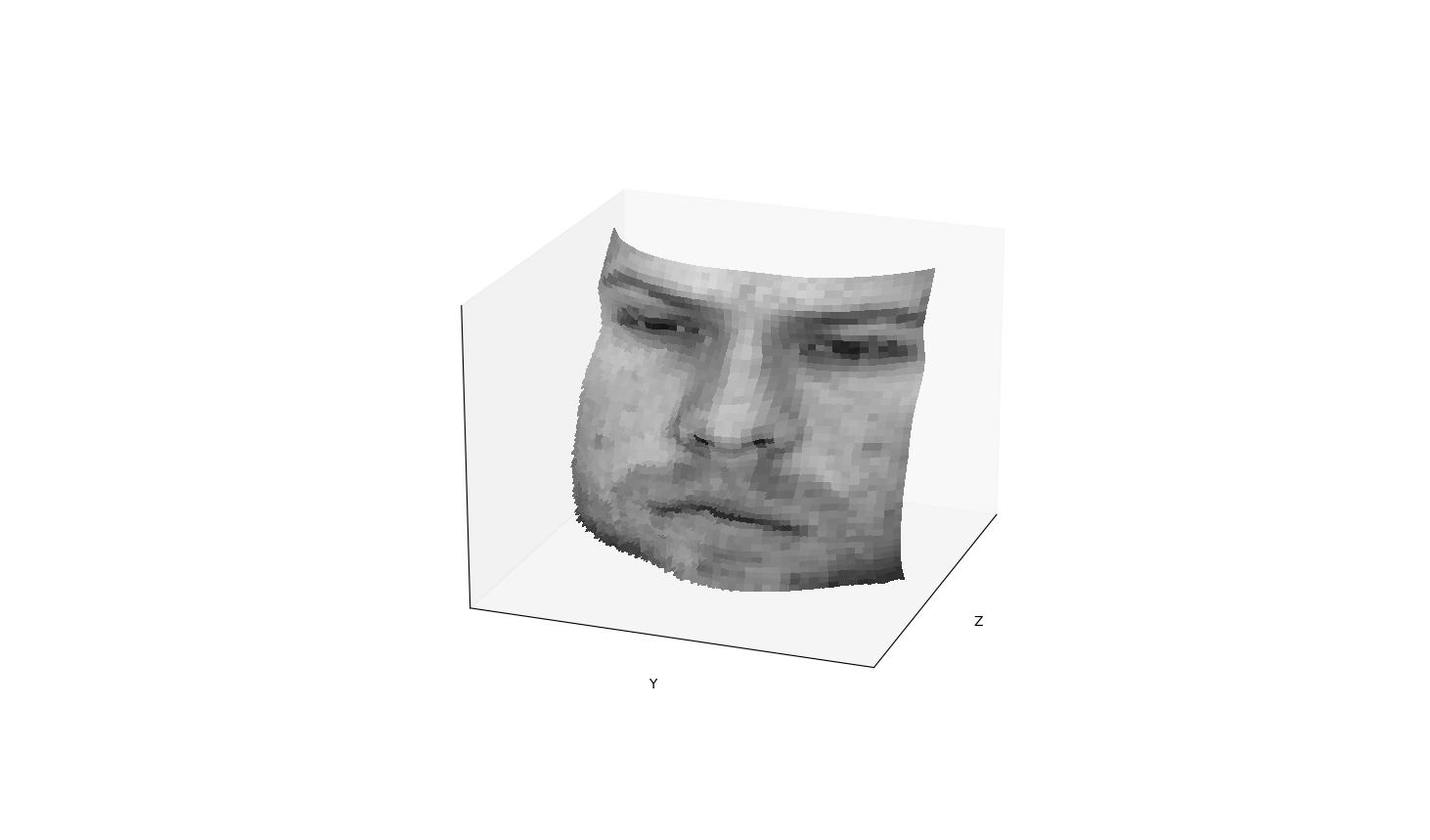
**C: Violation of the assumptions**

1. Discuss how the Yale Face data violate the assumptions of the shape-from-shading method covered in the slides.

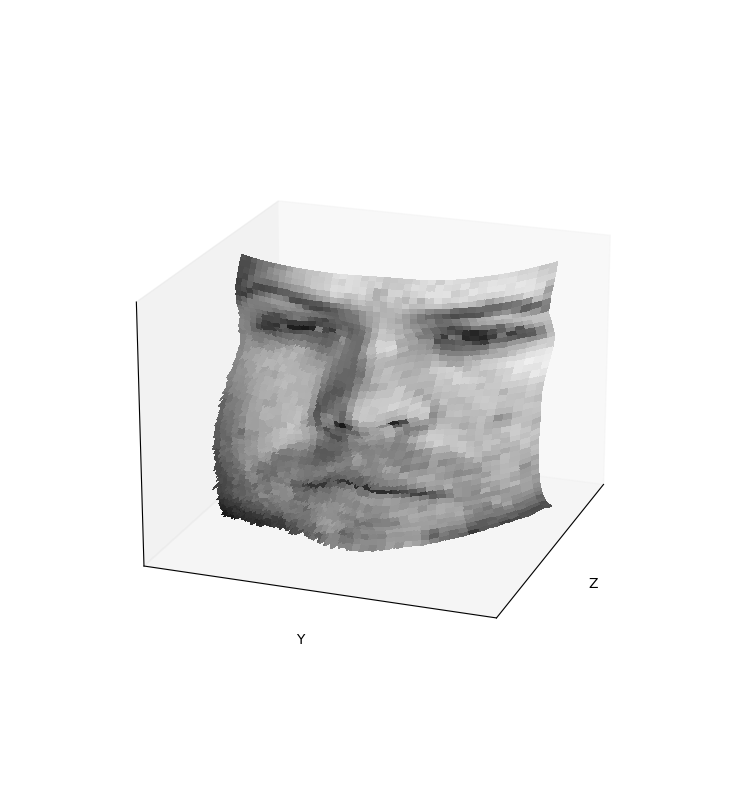
The data source may not be completely Lambertian, and it contains only one picture for each object, while it should have contained a set of pictures instead of only one.

1. Choose one subject and attempt to select a subset of all viewpoints that better match the assumptions of the method. Show your results for that subset.

Original:



Choosing a subset:



1. Discuss whether you were able to get any improvement over a reconstruction computed from all the viewpoints.

Yes, I did get a improvement after selecting a few bright view points.