CS543/ECE549 Assignment 3

Name: Huey-Chii Liang

NetID: hcliang2

Part 1: Homography Estimation:

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

I used SIFT detector to extract keypoints and compute descriptors of the images. The results are shown in Figure 1 and 2. Since we only focused on keypoints in the overlapping area, I selected the top 200 smallest pairwise distances between the descriptors in the two images. The new results are shown in Figure 3 and 4. In RANSAC, I ran the algorithm 10,000 times with a threshold of 5 to obtain satisfactory inliers. I constructed a matrix A using selected points and solved the least squares system AX=0 through SVD. For image warping, I first created a large enough black image and pasted the right image onto it. Then I set the pixel value to 0 in the area where the two images overlap. Finally, I added the left image to it to get the final stitched image.

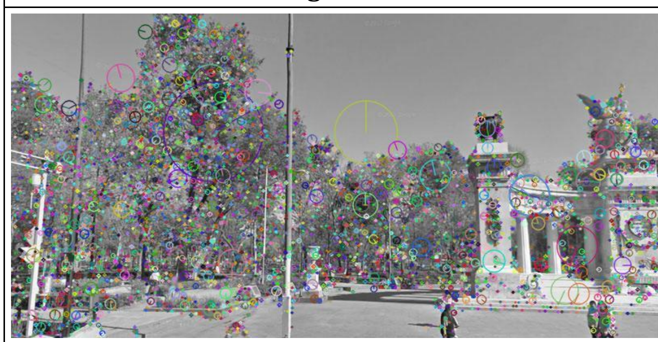
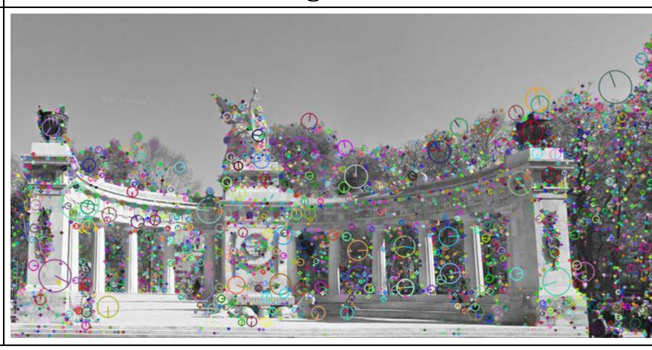
Figure 1	Figure 2
	

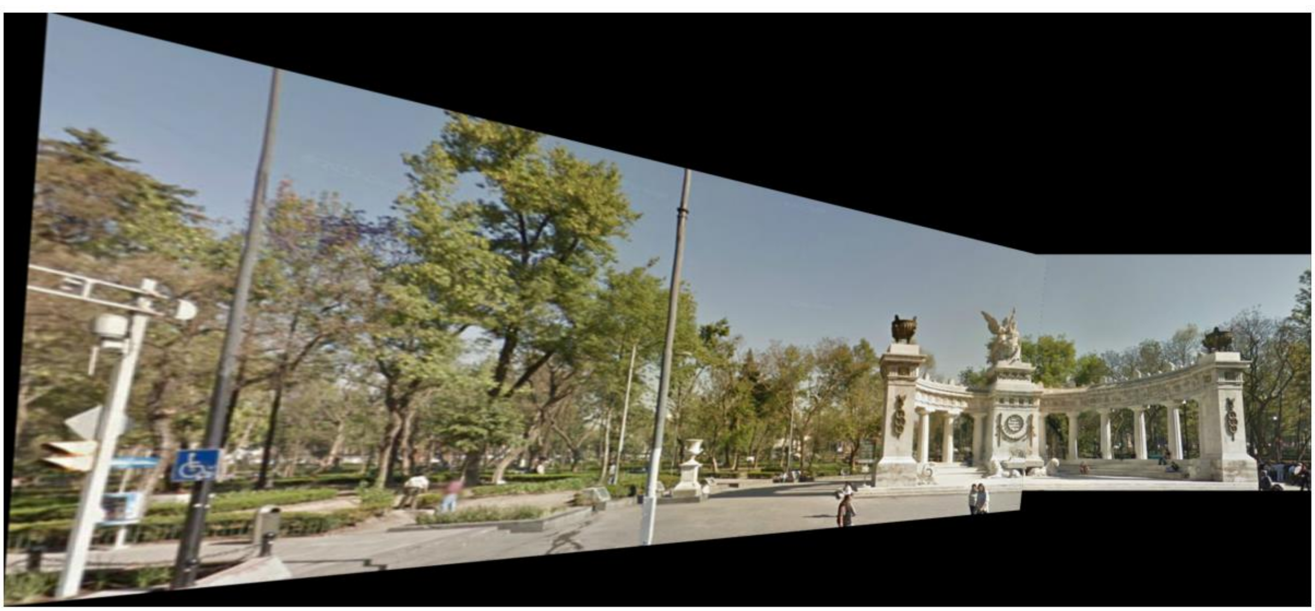
Figure 3	Figure 4
	

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

The number of homography inliers: 72.

The average residual for the inliers: 1.2365.

C: Display the final result of your stitching.



Part 2: Shape from Shading:

A. Estimation the albedo and surface normal

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

			
yaleB01	yaleB02	yaleB05	yaleB07

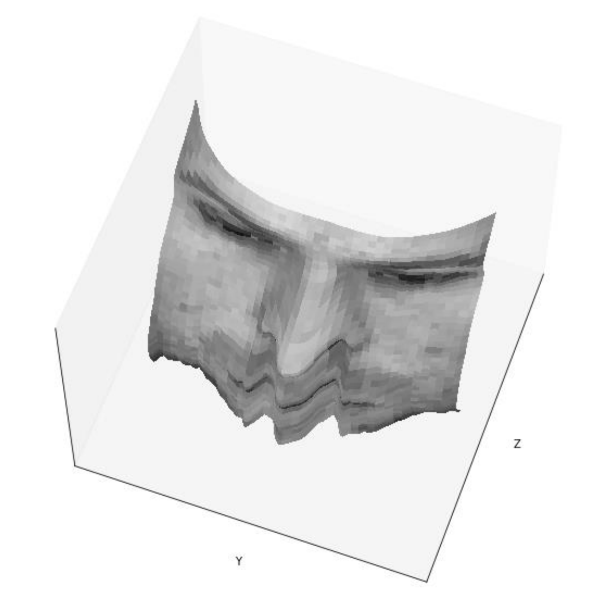
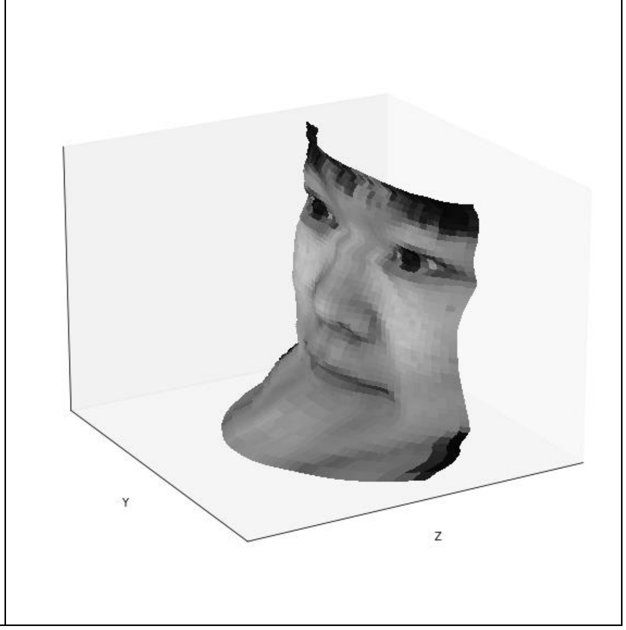
2) What implementation choices did you make? How did it affect the quality and speed of your solution?

In the function “photometric_stereo”, I changed the shape of the image array from (h, w, N) to (N, h×w), so I could directly use the “np.linalg.lstsq” function to get all the solutions at once. This step can be much faster than solving linear systems for all pixel. For the random integration, I used “np.random.randint(2)” to randomly select the integration path. If the value is 0, the path will move forward one pixel in the positive Y direction. If the value is 1, the path will move forward one pixel in the positive X direction. When the integration path reaches the maximum value of one direction, it will only proceed in the other direction until it reaches the bottom right pixel of ‘surface_normals”.

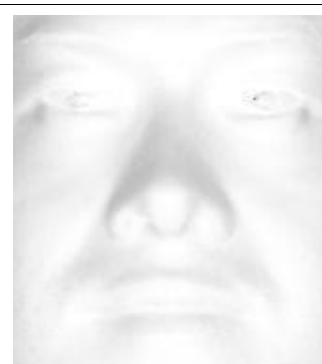
3) What are some artifacts and/or limitations of your implementation, and what are possible reasons for them?

The height map of yaleB01 using the “row” method has a wave-shaped mouth, as shown in the left image below. I think it is because the "row" method does not consider the relationship between two adjacent columns except row=0, which leads to an increasing integration error and ultimately results in the row at the relative bottom not being smooth enough.

The image on the lower right shows that the height map of yaleB05 using the “column” method has an abnormal chin. I think the reason is that there is dark hair around the chin in the lower corner of the photo, just like shadow, so integrating over the column for that part would create a huge value.

Height map using “row” method	Height map using “column” method
	

4) Display the surface normal estimation images below:

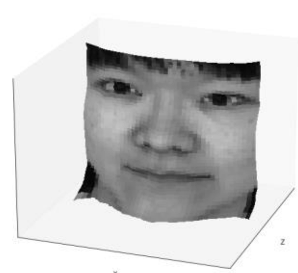
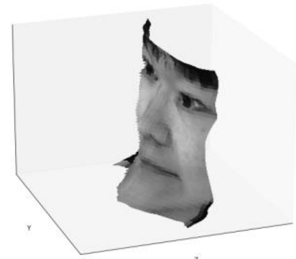
	normals_x	normals_y	normals_z
yaleB01			
yaleB02			

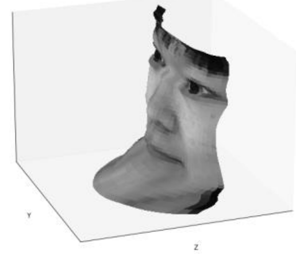
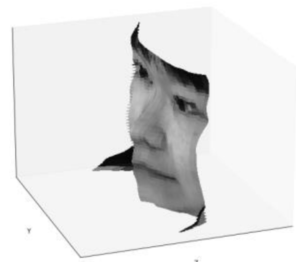
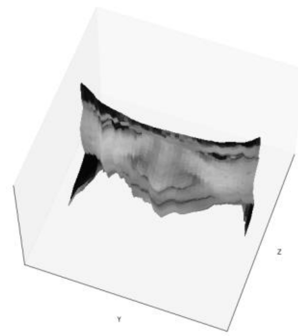
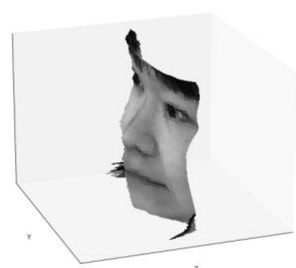
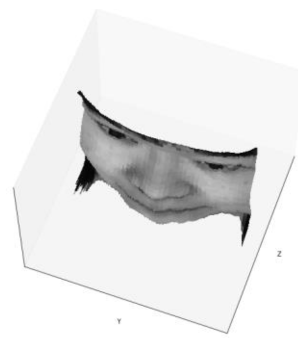
yaleB05			
yaleB07			

B. Compute Height Map

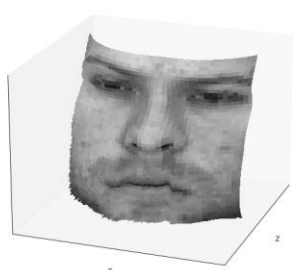
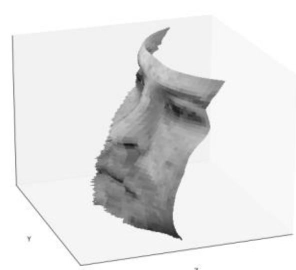
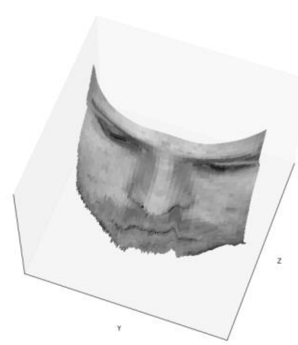
5) For every subject, display the surface height map by integration. Select one subject, list height map images computed using different integration method and from different views; for other subjects, only from different views, using the method that you think performs best. When inserting results images 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.

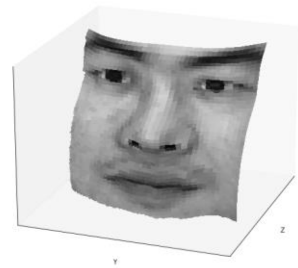
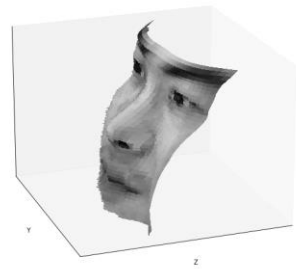
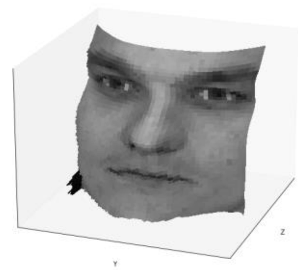
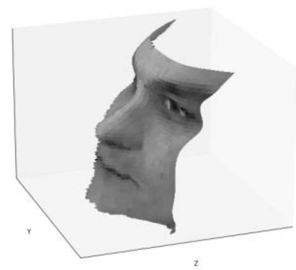
For yaleB05:

	front	side	up
average			

column			
row			
random			

For integration method “random”:

	front	side	up
yaleB01			

yaleB02			
yaleB07			

6) Which integration method produces the best result and why?

The "random" approach always gives the best results. This is because the random method averages many paths with different integration values, so it is more robust to noise. Even if there is noise on some paths, averaging can smooth and reduce the impact of the noise.

7) 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
Random	180.49233055114746 sec
Average	0.01855912208557129 sec
Row	0.01768522262573242 sec
Column	0.017689776420593262 sec

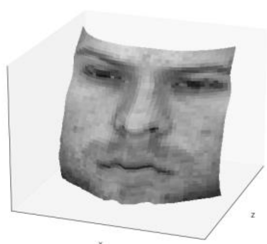
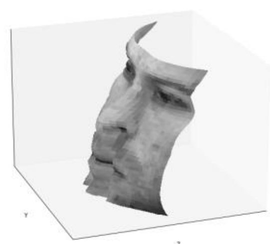
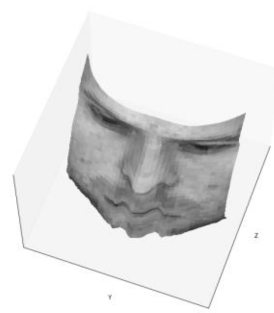
The "row" and "column" methods have similar running times, with the "average" method slightly longer. The "random" method takes much longer than other methods. The reason is that the "row" and "column" methods only integrate once. The Average method performs two integrations. But the “random” method performs 10 iterations, and each iteration is equivalent to an integration.

C. Violation of the assumptions

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

When capturing faces, the images do not all look the same, as shown in the image below. There are some subtle differences between faces, such as eye size, lip shape, etc.

9) 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.

Albedo	Height map (front)	Height map (side)	Height map (up)
			

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

I think the image recovered using the selected subset is not better than before. The reason is that the selected subset contains less information than the original one, even though this subset may contain less noise and errors. As a result, the 3D view image is not better.