HW 2 Writeup
COS 429

Alexander Gaura

October 10, 2018

1 Algorithms

1.1 Preprocessing

For the preprocessing, I subtracted the ambient image from every image, normalized it so that the maximum value in the entirety of the image set would be 1, and eliminated all negatives that may have resulted from the subtraction and turned them into zeros.

1.2 Albedo

A linear system was solved finding the g such that when g was multiplied by the light directions, it resulted in the pixel values, i.e. for every pixel we had to solve

$$\begin{pmatrix} I_1(x,y) \\ ... \\ I_n(x,y) \end{pmatrix} = \begin{pmatrix} V_1(x,y) \\ ... \\ V_n(x,y) \end{pmatrix} g(x,y)$$

where n is the number of images, I are the intensities and V are the light directions. The magnitude of these g vectors for every pixel made up the albedo image.

1.3 Surface Normal

The surface norms were found by taking the g vectors and dividing them by their magnitude. Essentially, we solved $N_i(x,y) = g_i(x,y)/||g||$.

1.4 Integration

Integration was performed by finding the ratio between the x and z coordinates and between the y and z coordinates of the surface normals multiplied by -1 (i.e. $f_x(x,y) = -N_1(x,y)/N_3(x,y)$ and $f_y(x,y) = -N_2(x,y)/N_3(x,y)$). Using the top-left corner as our starting position, the expected height of a pixel would be the sum of the x/z and y/z ratios corresponding to the pixels traversed on some path from point (1,1).

2 Results

Figure 1: Albedo images are of subjects 1, 2, 5, and 7, respectively.

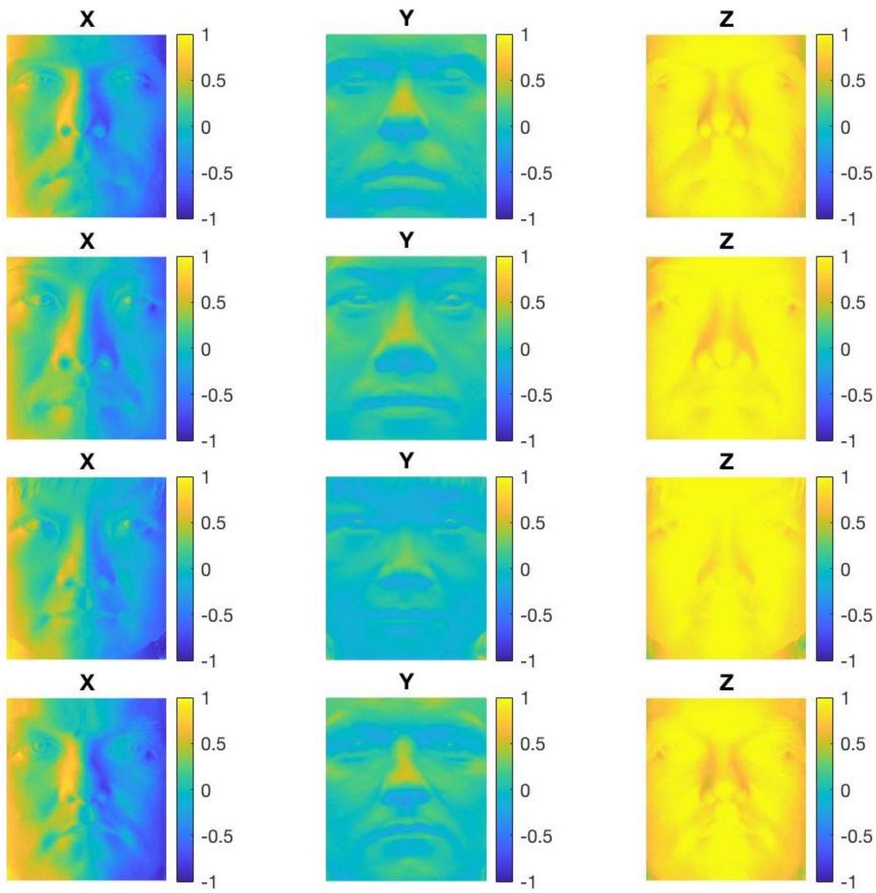

Figure 2: Rows of surface normal images are of subjects 1, 2, 5, and 7, respectively.

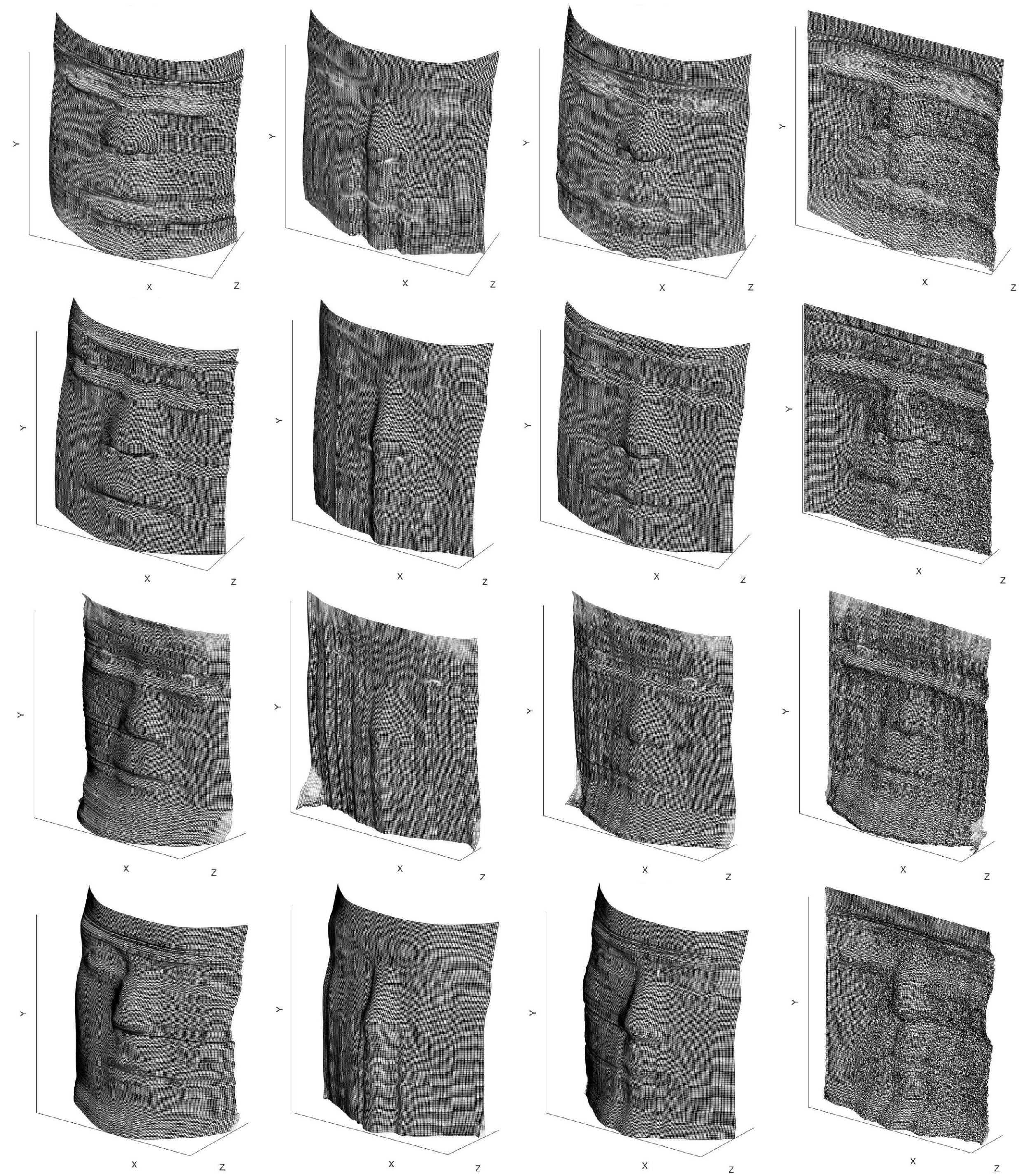

Figure 3: Rows depict, from top to bottom, subjects 1, 2, 5, and 7. From left to right the columns are the row, column, average, and random integration methods.

3 Discussion of Subject 7

3.1 Results

Subject 7, whose height maps are shown on the bottom of page 3, seems to have the best results when taking the average method. Two major flaws with the row and column methods are their tendency to cause ridges along rows and columns respectively, and this is particularly prone to happen after crossing a major 3d feature. Notice how the right side of the face has many horizontal ridges after passing the nose and lips for the row image, or notice how the column image has two major ridges flowing down from the sides of the nose. These issues are made much less pronounced in the image produced by the average method. Additionally, features are given a more well-rounded effect, like how the nose does not look as blunt as in the row image or as a giant ridge as in the column image. The image produced by the random method, while giving much shape to the nose, tends to flatten the borders of the image giving a concavity that human faces do not naturally have. This may be from an issue in the implementation, and I would regard the flattening of the edges as far less natural than the subtle ridges still found on the image produced by the average method. Thus, subject 7 looks best with the average method.

3.2 Timing

The row, column, and average methods were all roughly similar in timing, which during one recording were found to take 1.016389, 1.014804, and 1.020031 seconds long to complete respectively. The random method, however, required a random path to be found for every pixel, which was much more time consuming. Additionally, the amount of variation produced by taking a random path had led to some particularly bad results which needed to be averaged out. A single run was found to take 23.089086 seconds in one instance, but 30 images were averaged together to produce the results above. So, the entire run of the random method was found to have taken 658.619087 seconds in one of these instances.

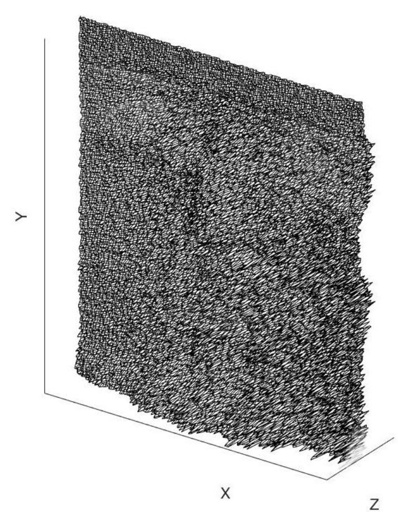

Figure 4: Result of the random method on subject 7 without averaging.

4 Shape-from-Shading Violation

Clearly, all images produced have some issues with how the faces are represented. Subject 7 arguably comes closest after having used the average method, but it still shows ridges. As the data is discrete, the actual differences needing to be integrated often will not accurately reflect the entire nature of what is going on surrounding a pixel, which is what is often causing these ridges. This issue can best be seen in subject 5 as they had some hair floating above their forehead. It can be seen that all images with the exception of that of the row method were left with many vertical ridges as a result of the hair.