

CS 543 - Assignment 1: Shape from Shading

Yiwen Xu

February 16, 2016

Abstract

This project mainly accomplished the task to reconstruct faces from a series of pictures of taken under different illuminants. Based on **Photo-metric stereo**, surface normals are reconstructed after basic per-processing of image series and integration is applied on surface normals to form the height map.

1 Description of Implementation

1.1 Albedo and Surface Normals Estimation

When solving least square problems for albedo and surface normal, instead of directly implement the method introduced in the lecture, some adjustments have been made to improve the performance. In order to avoid loops in Matlab, the least square problem is arranged in following way so that all can be solved at one backslash operation.

$$\begin{bmatrix} I_1(x_1, y_1) & I_1(x_2, y_2) & \cdots & I_1(x_{lm}, y_{lm}) \\ I_2(x_1, y_1) & I_2(x_2, y_2) & \cdots & I_2(x_{lm}, y_{lm}) \\ \vdots & \vdots & \ddots & \vdots \\ I_n(x_1, y_1) & I_n(x_2, y_2) & \cdots & I_n(x_{lm}, y_{lm}) \end{bmatrix}_{n \times (l \times m)}$$
$$= \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix}_{n \times 3} \begin{bmatrix} g(x_1, y_1) & g(x_2, y_2) & \cdots & g(x_{lm}, y_{lm}) \end{bmatrix}_{3 \times lm}$$

1.2 Height Map Recover

Five implementation of integration methods to recover height map is implemented, namely 'Column', 'Row', 'Average', 'Random' and 'Onedim-Random'.

Column, Row and Average: Take 'Column' as an example, integration is accomplished by first generate integration along the first column and than take the integrated first row as a start to accumulate row-wisely. 'Row' is implemented likewise and 'Average' is simply the mean of 'Row' and 'Column'.

Random Path: Random path implementation start from the leftup corner and randomly select path to go vertically and horizontally towards destination (x, y) . Direction could be changed at each pixel.

One Dimensional Random Path: Randomly pick one row as start and divide the image into two parts, namely upper one and lower one, then integrate vertically in different direction. This method is plus method to eliminate distortions due to long accumulation path.

1.3 Artifacts and Limitations

One artifacts with my implementation of random path would be the non-smooth surface in the rightdown corner. The reason for this might be the directions turn too frequently. For a small area, if neighboring pixels accumulated from dramatically different paths which carry dramatically different errors. these error cannot be eliminated by simply averaging. Details explained in latter section.

2 Integration Methods

2.1 Output Comparison

Given to the experience I have conducted, I would say 'Column' works best among four basic types of integration methods. Output height map from different methods are shown in Figure 1.

Reason of Difference:

In my understanding, the quality of height map largely depend on the travel distance that carrying errors. The longer distance as it travels the poorer quality that area is going to get. The most significant error in the face comes from parts under nose when going vertically down. Therefore, 'Row' start with initial row at top and takes large accumulated error as it travel downwards especially under nose part. In contrast to 'Column', where the vertical big error is avoided by traveling horizontal rightwards across that part and thus make quality height map. The 'Average' takes the wavy lip from 'Row' and mix with 'Column' thus has quality in between.

Another issue affects quality of height map comes from the travel path coherency in neighboring area. When using 'Random', neighboring pixels is going to be integrated from significant different path. As it is seen in comparison, errors at same area can be different when travel in different directions. Therefore frequent turning direction makes huge difference between neighboring pixels. Averaging somehow doesn't make up for the accumulated error. Assume we are traveling from (m, n) to $(m, n + k)$, even if the latter point coming from subpath of former one, there is going to be 2^k methods of afterwards steps. As the averaging effects grows linearly, the error grows exponentially. Therefore in random right-down corner has more profound artifacts.

2.2 Running Time

The running time of four basic methods as well as my additional implementation of 'One-dimensional Random' is shown in Table 1.

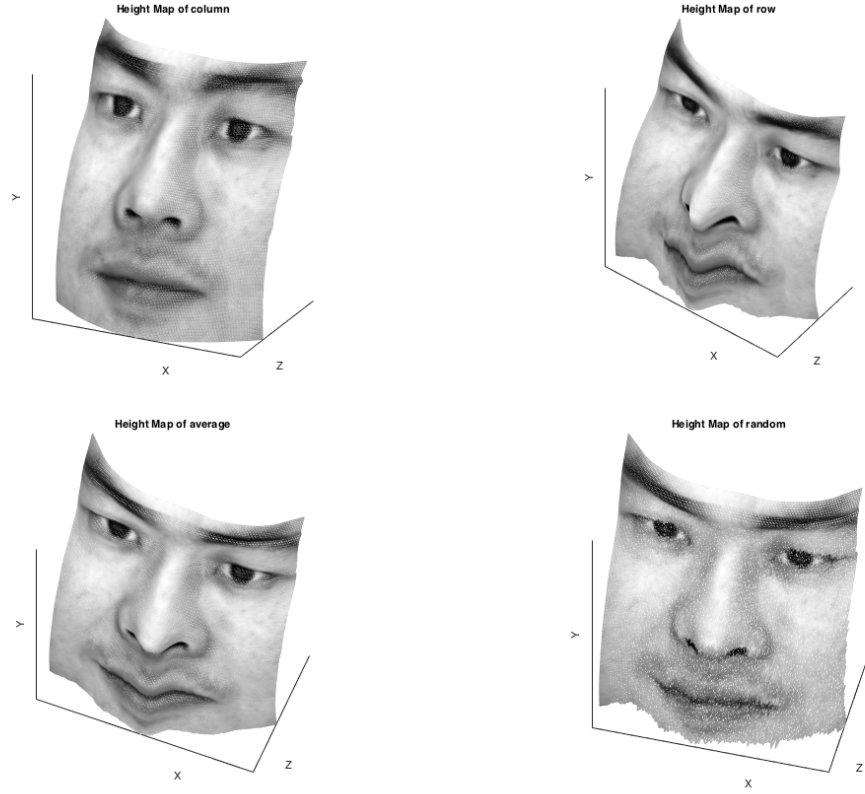


Figure 1: Output height maps from 4 basic methods, namely 'Column', 'Row', 'Average' and 'Random'.

| Implementation | Column | Row | Average | Random | One-dim Random |
|----------------|------------|------------|---------|--------|----------------|
| Running Time | 5.8349e-04 | 6.7812e-04 | 0.0011 | 0.7571 | 0.0058 |

Table 1: The running time comparison for different implementation.

2.3 Additional Illustration

For other subjects, generally speaking column is still the best but sometimes chins could wrongly go upwards. The best output of rest of subject shown as Figure 2.

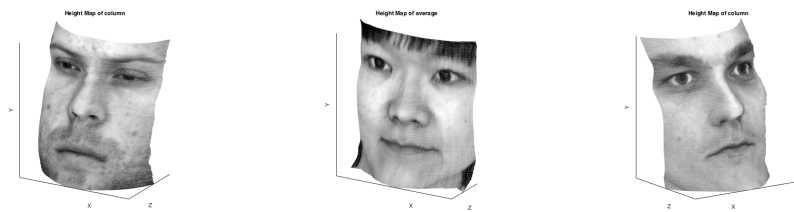


Figure 2: Best output height maps from rest of subjects.

3 Albedo and Height Maps

The estimated albedo maps, surface normals and height maps for each subject is shown in Figure 3. The 3D height maps is constructed all in 'Column' method for comparison. As one can see, 'Column' is not necessarily the best integration method for subject 5 where creates an obvious up-pointed chin. Improvement will be introduce in next section.

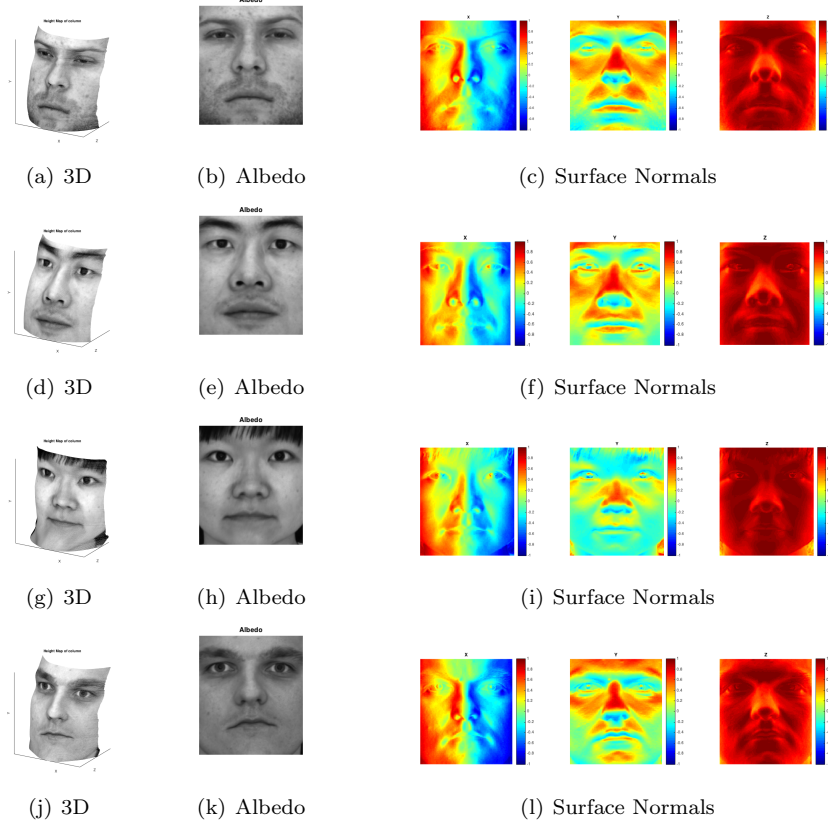


Figure 3: 3D, albedo and surface normal maps for subject 1,2,5,7

4 Improvement for Extra Credit

As it is mentioned in former section, the estimated error of height maps is accumulative thus its quality highly depend on traveling distance. Common problems for subject 5 and 7 is that they probably contain up-pointing chins. The reason for that is the long distance accumulation of error area, most errors happens in under nose part.

4.1 One dimensional random

In this section, One dimensional random method is introduced to eliminate the error by decrease accumulating distance. To do that, a random horizontal line is

picked as initial constant. In this line each pixel is integrated from left to right. For the vertical direction, integration happens from the initial line and go up or down towards target position. When go upwards, the integration happens reversely travel from down to up with negative value for the middle random line is considered as a hill. When go downwards, the integration happens normally.

4.2 Output Improvements

Comparison between 'Column' and 'One dimensional Random' shown in Figure 4. As a matter of face, one dimensional random method generally overperforms other basic method in creating overall good estimation but still contains some artifacts like wavy surface.

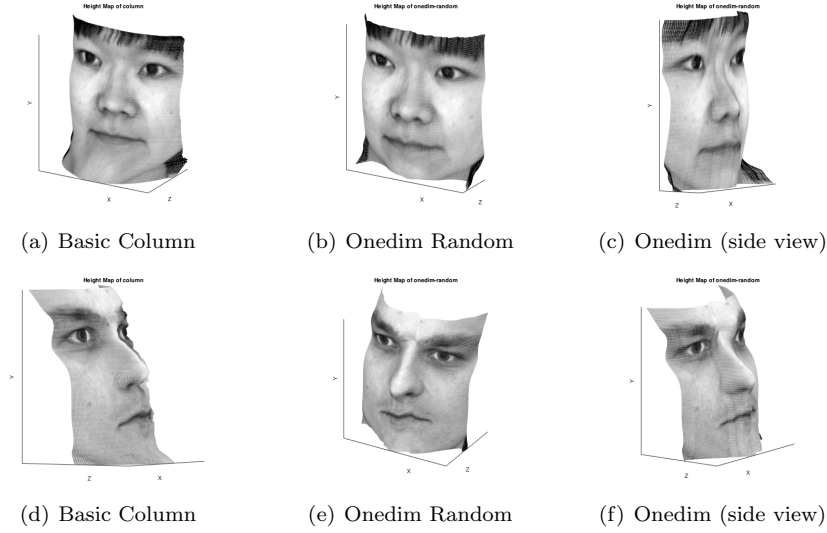


Figure 4: Output improvement for subject 5,7 using one dimensional random

5 Discussion

5.1 Violate of Assumption

Shade from shading method based on several basic assumptions, which will cast problems to reconstructed map if those assumptions are not fulfilled. Also some practical issue during facial data collection process might also has issues with poor mesh quality. Those model assumptions violation and practical issues might include:

1. *Assumption: Face can be seen as a lambertian object.* In the diffusive reflection, light should be reflected equally in all directions and independent of viewing direction. Face contains areas that cannot always be simplified as only diffuse reflection happens, such as oily nose tip, front head and eyes, like 5(a). The reflection model in those part are most likely to be a specular reflection. On the other hand, furry parts also create a issue

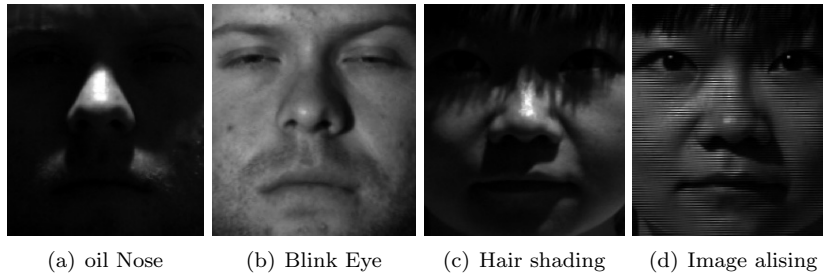


Figure 5: Sample images of violation

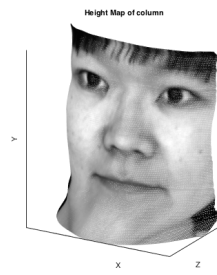
because they block out the reflective lights and seems dark in all directions which is the same as a hole. In the model, therefore eyebrow more likely to be concave parts. And if some hair is floating in front of face, that might also cause problems, like 5(c).

2. *Assumption: Exact same camera configuration and object settings.* Camera settings are usually very controllable but object is not that easy to be overall stable. It is obvious people's face various over time, like blinking eyes, shift position and so on, like 5(b).
3. *Assumption: Image collected is accurate.* It is actually a common sense that a camera cannot capture facial details very well when the amount of light coming in is extremely limited like 5(a). For shady parts it is more likely to be noise in height map. And actually, when looking into the datasets I found not all the pictures are perfectly shot but some imaging artifacts exist such as alising in 5(d). Those might be contributes to noise as well.

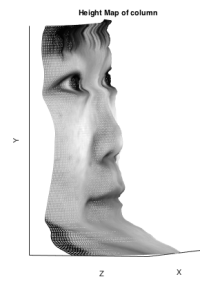
In hope of having better results by eliminating photos seem to have artifacts on final results on Subject 5. The original dataset is size of 64 and was reduced to size of 35, 40 or 50. I tried with several subset of images getting rid of problematic images but no significant improvement is found, though. Except for the chin problem is better which probably means noise is less. See below Figure 6.

References

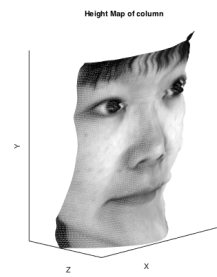
- [1] David A. Forsyth, Jean Ponce *Computer Vision: A Modern Approach, 2nd Edition*.
- [2] Yale Face dataset <http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html>.
- [3] Lecture Notes, Svetlana Lazebnik <http://slazebni.cs.illinois.edu/spring16/lec04-light.pdf>.



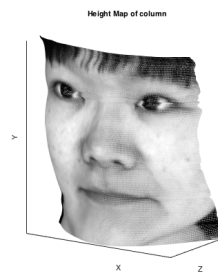
(a) Original Set



(b) Side View: Original



(c) Subset



(d) Side View: Subset

Figure 6: Subset compare