EECS442 Homework2 Ningqi Zhu

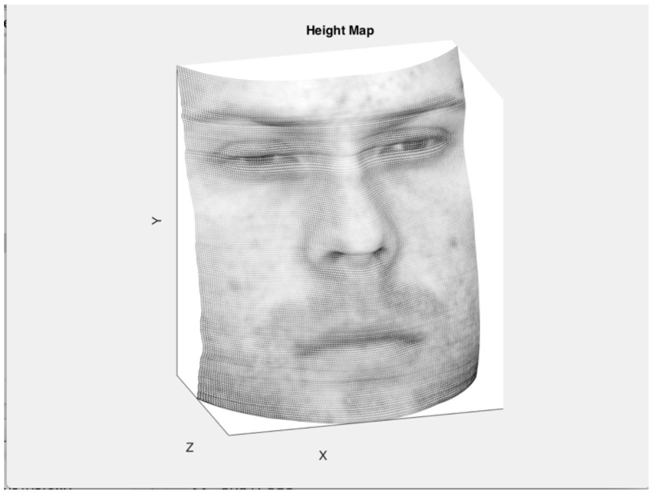
Part1

1. prepareData.m:
    a. Used bsxfun(@minus) to subtract the ambientImage from imArray.
    b. Used find to seek for the max element in imArray and divided imArray by it to scale imArray into 0~1.
2. photomericStereo.m
    a. Used permute and reshape to make A into （n,h*w） matrix
    b. Used “\” to get the g (x,y)
    c. Used permute and reshape to make g into 3 dimensional matrix
    d. Calculated the magnitude of g(x,y) to get the albedoImage
    e. Used bsxfun to get the surfaceNormals
3. getSurface.m
    a. Column method:
        i. Since every points need the integration of the first column of fy(x,y), I used cumsum to calculate the integration of first column and duplicated it into h*w matrix
        ii. Used cumsum to get the sum down the fx (x,y)
        iii. Added two matrix together to get the heightMap
    b. Row method
        i. Instead of do the integration of the first column of fy(x,y), I calculated the integration of the first row of fx(x,y) and duplicated it into h*w matrix
        ii. Used cumsum to get the sum along each row of the fy(x,y)
        iii. Added twp matrix together to get the heightMap
    c. Average
        i. Just calculated the average of the column method and row method

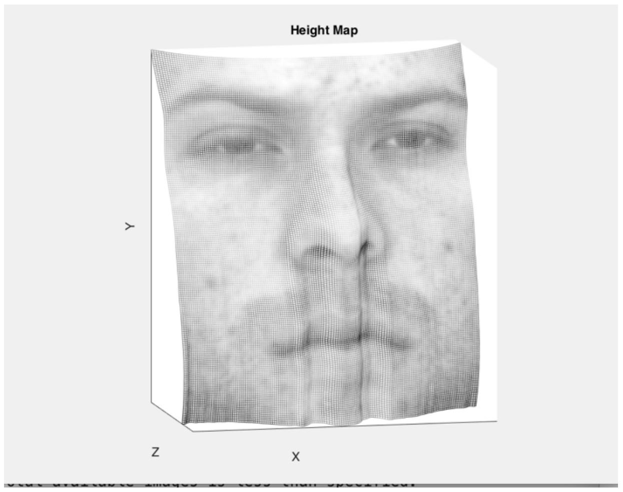
    d. Random
        i. Start at (1,1), generate a random number of 1 or 0.
        ii. If 1, move down the column, else move along the row
        iii. If move from(m,n) to (m+1,n) add fy(m+1,n) to the integration
        iv. If move from (m,n) to (m,n+1) add fx(m,n+1) to the integration
        v. Loop this process unitil (m,n) get to the point(x,y)
        vi. Repeat several times(100 i.e.) and calculates the average number

Part2

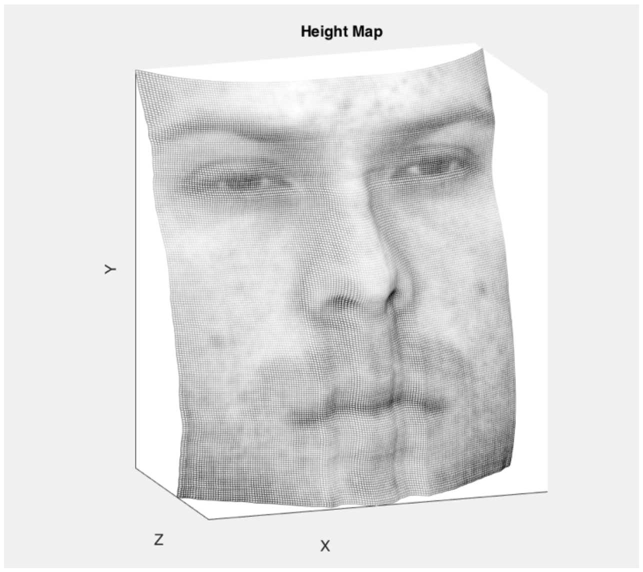
I will use yaleB001 below to illustrate my work:



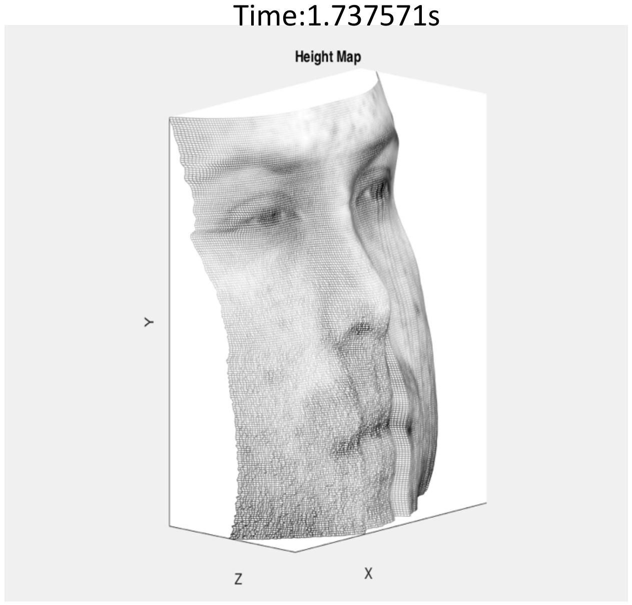
Column Method
Time:1.915766s



Row Method
Time:1.737571s



Average Method
Time:1.967338s



Random Method
Time:28.083671

Here I display 4 3D screenshots of the heightMap of the yaleB01.

As you can see, in the Column Method, the heightMap is more smooth along the x-axis, which is the direction of the column. Because there is only little difference between every adjacent pair of points with the same y's height, say the fx(x,y), which is extremely small compared to the whole integration.

In the Row Method, the heightMap is more smooth along the y-axis, which is the direction of the row. The reason is similar to Column Method, the difference between two adjacent points along y are only a small number compared to the whole integration.

The time of these two methods are comparable.

As for Average Method, at the top of the figure(head) the heightMap is more smooth along x-axis, while at the bottom it is more smooth along y-axis. Since at the top and especially top right part of the heightMap, the integration along x-axis are much larger than the integration along y-axis, so it's looks like the Row Method. And vice versa in the bottom part of the picture which looks like the Column Method.

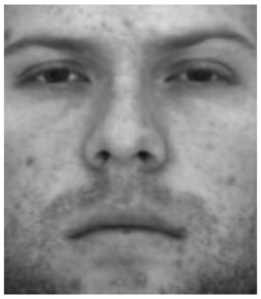
The time of Average Method is smaller than the sum of Row and Column Method. I speculate that the calculation of fx(x,y) and fy(x,y) takes the main part of the time which do not need to be calculated twice in Average Method.

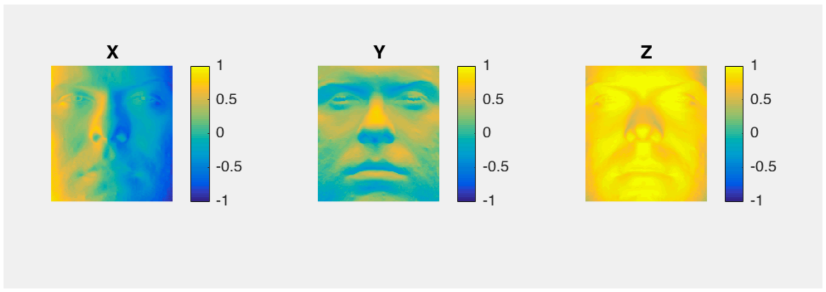
The result of Random Method is not smooth in any direction since every points may be calculated in totally different path even though I calculated 100 times and got the average number. The area around the mouth are extremely rugged because this man has his beard around it. Also, since my probability are 50% and 50% to go down or right, in average case, it will get to the right side of the figure first and walk down the figure along y-axis. So you can find in the right part of the picture it is more smooth along y-axis.

The time cost in Random Method is much more longer than the other 3methods. I looped 100 times in my program but it is not 100 times longer in time. That also tells us the calculation of the fx and fy is more slow than loop the matrix once.

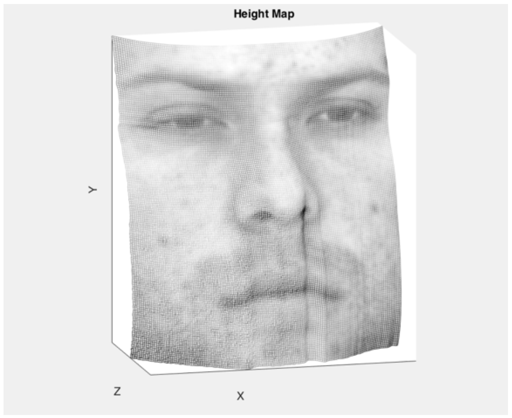
Part3 Display

YaleB01:


albedo

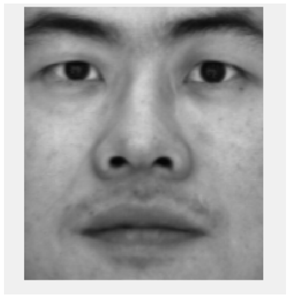


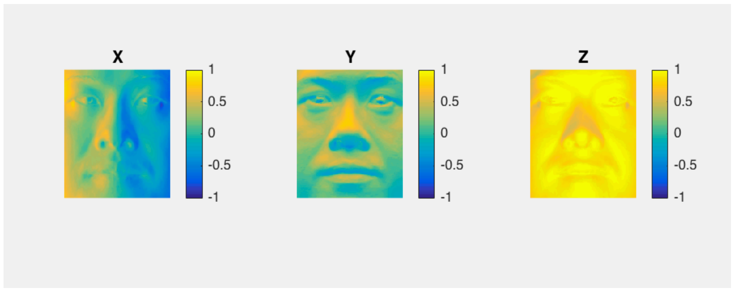
surface normal



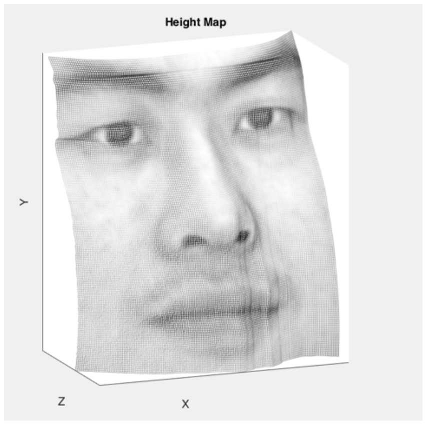
Height Map: Random Method

YaleB02:


albedo

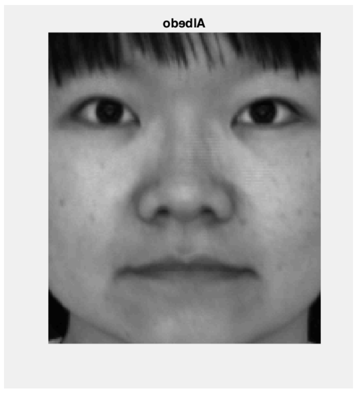


surface normal

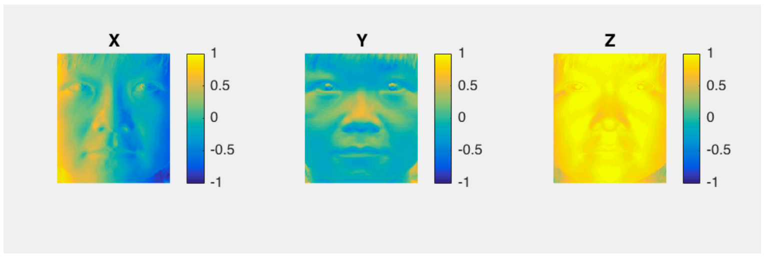


Height Map: Random Method

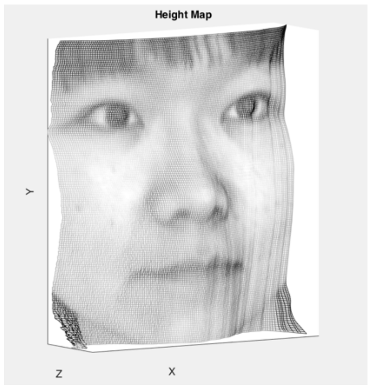
YaleB05:



albedo

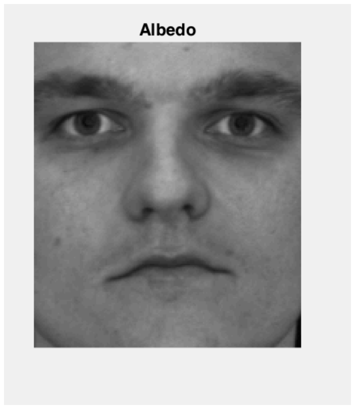


surface normal

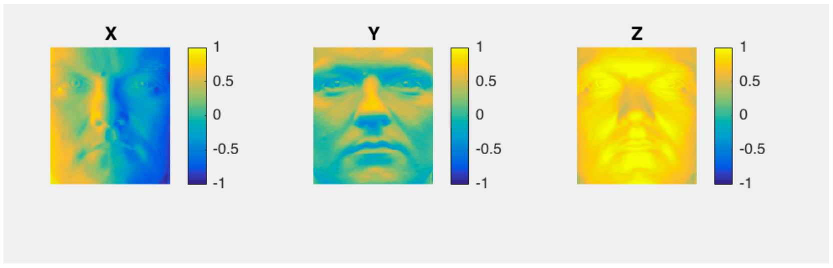


Height Map: Random Method

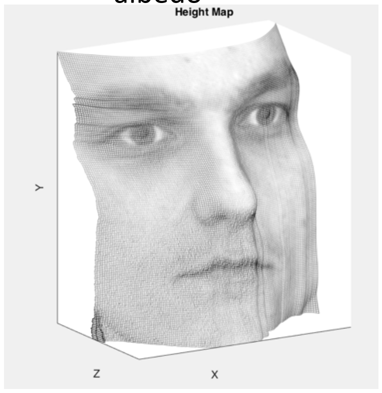
YaleB07:



albedo



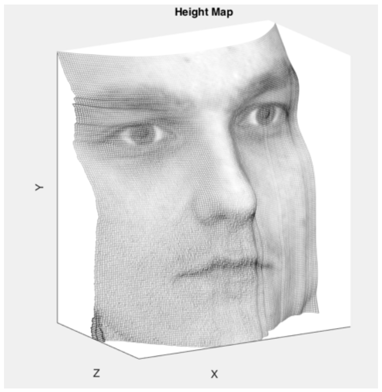
surface normal

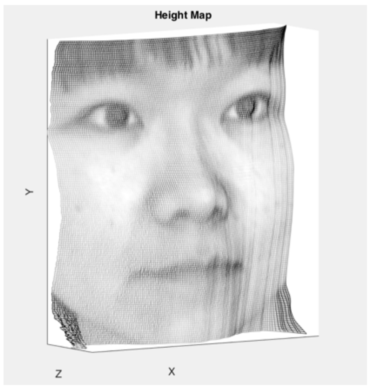


Height Map: Random Method

Part 4

1. First of all, the pictures are not aligned perfectly. As you can see in the below figures, the edge of the figures are extremely unsmooth due to the misalignment.

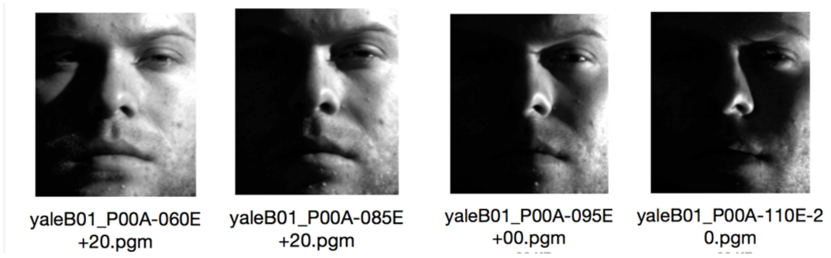




YaleB01 and YaleB02 perform better than YaleB05 and YaleB07 although they are processed in the same algorithm.

2. The integration of fx, fy is consecutive in the lecture slides. However here we use the discretion of the sum to substitute the integration. This is where the errors are mainly generated from.'

3. The shadow on the face may be another property that affects the result.



Since people have noses and lips, these will generate shadow for the picture which will affect the I(x,y)'s accuracy.