ECE549 Computer Vision: Assignment 1

Yutong Xie
NetID: yutongx6
Instructor: Prof. Saurabh Gupta

February 20, 2020

1 Question 3: Shape from Shading

1.1 3c: Estimate the albedo and surface normals

For four subjects, I present the albedo and surface normal images here.

1.1.1 YaleB01

For YaleB01 the residual of least square fitting is 1.7196826076250108. I got residuals for all N images and took the mean value.

```
npix = h*w
imarray = imarray.reshape((npix,N)).transpose()
g, re, ra, sv = np.linalg.lstsq(light_dirs,imarray, None)
re = np.mean(re)
```

Figure 1: Albedo and x-y-z surface normals of YaleB01

1.1.2 YaleB02

For YaleB02, mean residual is 1.925507003212805.

Figure 2: Albedo and x-y-z surface normals of YaleB02

1.1.3 YaleB05

For YaleB05, mean residual is 1.6902902287341741.

Figure 3: Albedo and x-y-z surface normals of YaleB05

1.1.4 YaleB07

For YaleB07, mean residual is 1.4576228512311087.

Figure 4: Albedo and x-y-z surface normals of YaleB07

1.2 3d: Surface height map

In this part, I used four different integration methods to generate the height map. I choose YaleB02 to analyze the difference between different methods here. For each method, I display the result from four different view points.

1.2.1 Row integration

Integrating first the rows, then the columns.

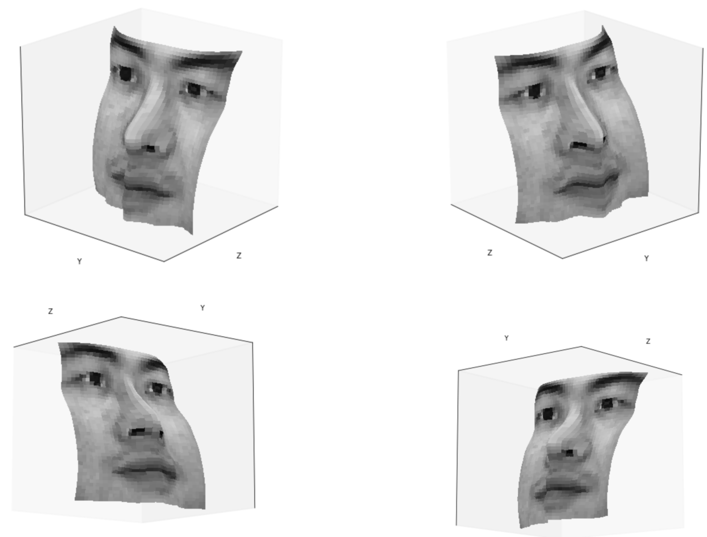


Figure 5: Height map for row integration from different view points

1.2.2 Column Integration

Integrating first along the columns, then the rows.

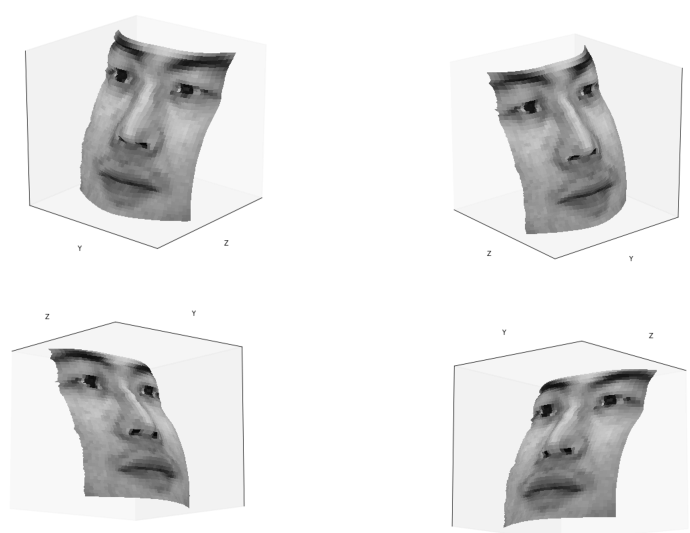


Figure 6: Height map for column integration from different view points

1.2.3 Average integration

Average the row integration and column integration.

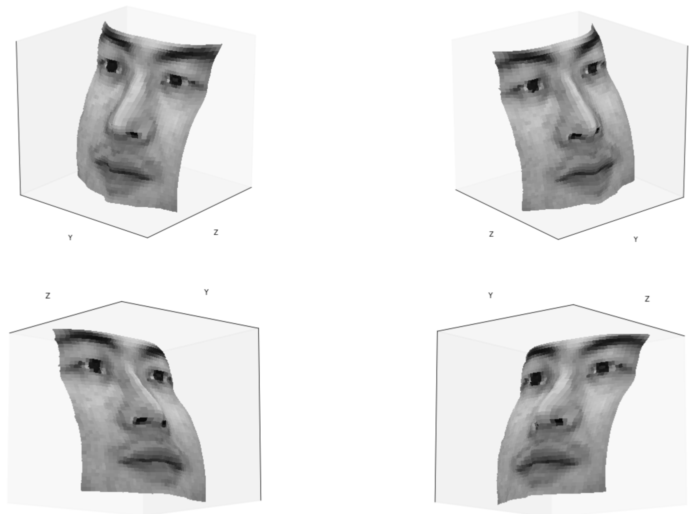


Figure 7: Height map for average integration from different view points

1.2.4 Random integration

Average of multiple random paths. I set the number of paths as 10 here.

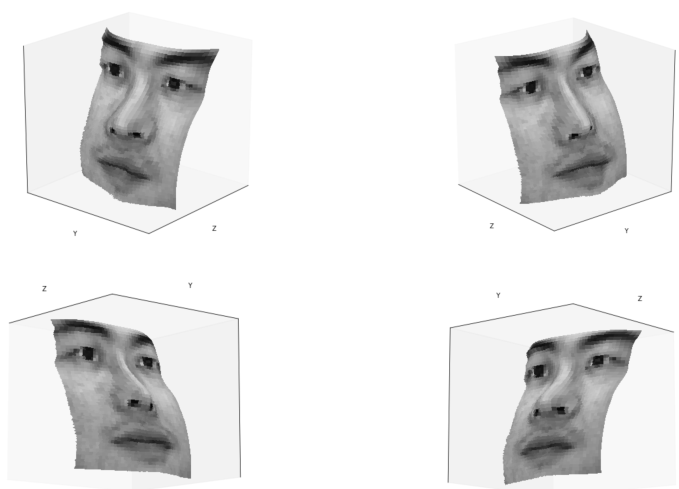


Figure 8: Height map for random integration from different view points

From the generated height map, I found that the random integration method produces the best results. The random integration generate n paths to calculate the height value of each pixel and take the average of them. Compared to other integration methods, random integration can achieve less bias for pixels due to randomness and the result does not mostly depend on the path itself.

The running time is shown below. We can see that random integration uses the longest time, because we calculate the height value pixel by pixel rather than using cumsum function. And the time is proportional to the number of paths. The time for row and column integration has little difference and the time for average integration is longer because it includes both row and column integration.

Integration Method	Running Time(s)
Row integration	0.0002839565277
Column integration	0.0002820491791
Average integration	0.0006852149963
Random integration	36.10919213

Figure 9: Running time for different methods

For the remaining three subjects, the random integration method is the best also and I show the results below.

1.2.5 YaleB01

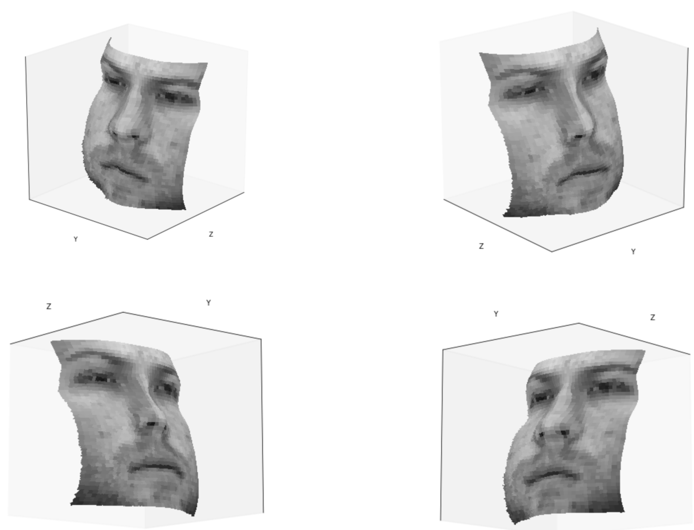


Figure 10: Height map for random integration from different view points

1.2.6 YaleB05

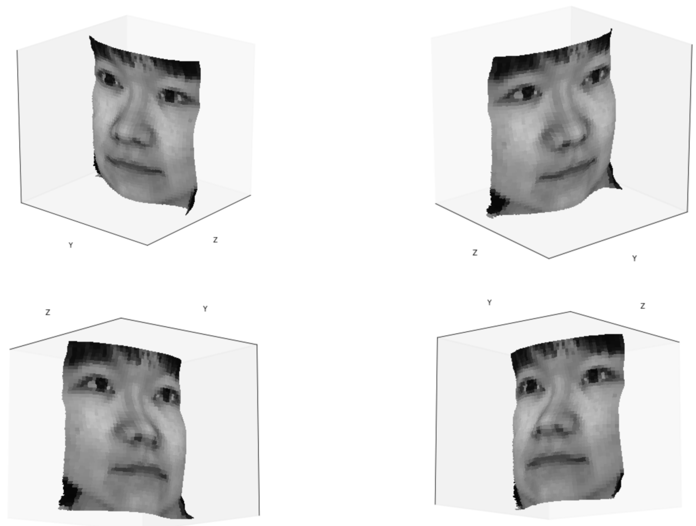


Figure 11: Height map for random integration from different view points

1.2.7 YaleB07

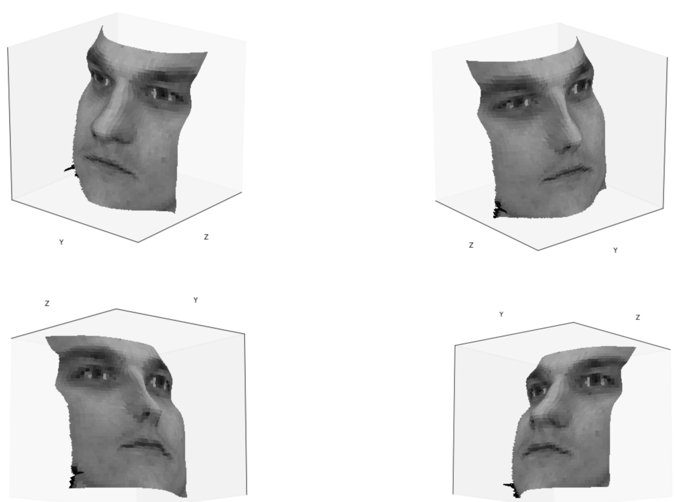


Figure 12: Height map for random integration from different view points

1.3 2e: Violation discussion

1.3.1 How the Yale Face data violate the assumptions of the shape-from-shading method covered in the slides. What features of the data can contribute to errors in the results?

In the slides, there are several assumptions, such as a lambertian object, l local shading model, a set of known light source directions, and so on. However, human face is not lambertian object because the light is not scattered equally in all directions. Meanwhile, I found that some images have heavy shading which would make the output not perfect.

1.3.2 Choose one subject and attempt to select a subset of all viewpoints that better match the assumptions of the method.

I choose the YaleB02 and selected the subset with less shadow to generate the height map. The result is shown below.

Figure 13: Original height map (left) and improved height map (right)

1.3.3 Show your results for that subset and discuss whether you were able to get any improvement over a reconstruction computed from all the viewpoints.

From the image above, we can easily see that the output image is improved by using subset. The face becomes more stereo, especially the nose, mouth and eyebrow.

1.4 Extra Credit

As a mentioned before, some images contain heavy shadows. Hence, I tried to find these images and use the remained appropriate images to do the reconstruction. The code is shown below.

```
improve_list = []
for fname in im_sub_list:
    im_arr = load_image(fname)
    count = 0
    for x in range(im_arr.shape[0]):
        for y in range(im_arr.shape[1]):
            if im_arr[x,y] < 40:
                count += 1
    shadow = count / im_arr.size
    if shadow < 0.4:
        improve_list.append(fname)
im_sub_list = improve_list
```

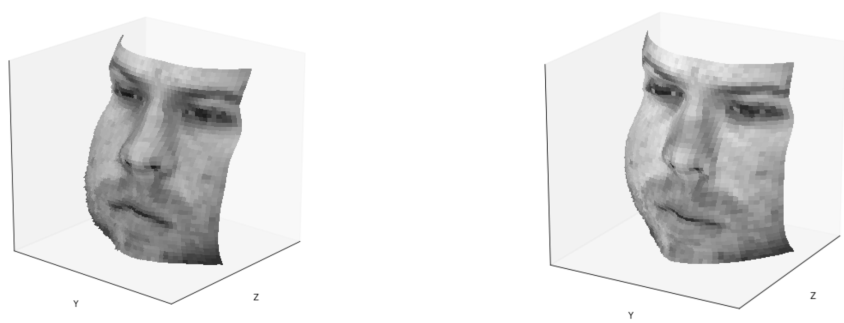


Figure 14: Original height map (left) and improved height map (right)

1. **Vanishing Points and Vanishing Lines [10 pts]**[1]**.** Consider a plane defined by $\mathbf{N}^T\mathbf{X} = d$, that is undergoing perspective projection with focal length f. Show that the vanishing points of lines on this plane lie on the vanishing line of this plane.

plane: $N^T X = d \Rightarrow N_1 X + N_2 y + N_3 z = d$.

$$\frac{N_1 f x}{z} + \frac{N_2 f y}{z} + f = \frac{fd}{z}$$

$$N_1 x + N_2 y + f N_3 = \frac{fd}{z}$$

As $x_3 \to \infty$

$$N_1 X + N_2 y + f N_3 = 0$$

Any line in the plane can be defined by two points $A = (x_1, y_1, z_1)$ and $B = (x_2, y_2, z_2)$ in the plane as.

$$L(\lambda) = A + \lambda B = (A_1 + \lambda B_1, A_2 + \lambda B_2, A_3 + \lambda B_3)$$
$$= \left(f \frac{A_1 + \lambda B_1}{A_3 + \lambda B_3}, f \frac{A_2 + \lambda B_2}{A_3 + \lambda B_3}\right)$$

As $A_3 + \lambda B_3 \to \infty \iff \lambda \to \infty$

$$\lim_{\lambda \to \infty} L(\lambda) = \left(f \frac{B_1}{B_3}, f \frac{B_2}{B_3}\right) = \left(f \frac{x_2}{z_2}, f \frac{y_2}{z_2}\right)$$

Put the point back to the vanishing line function $N_1 x + N_2 y + f N_3$, we can get:

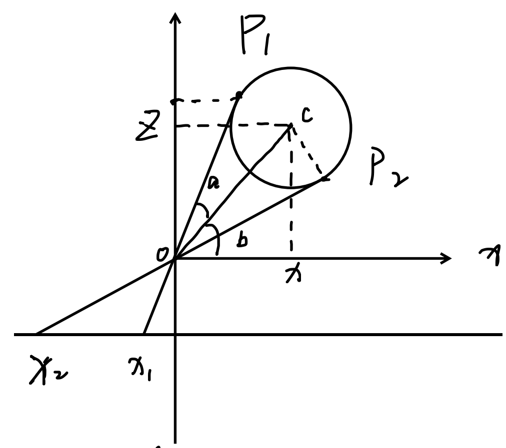
$$N_1 f \frac{x_2}{z_2} + N_2 f \frac{y_2}{z_2} + f N_3$$
$$= \frac{f}{z_2}(N_1 x_2 + N_2 y_2 + N_3 z_2) = \frac{fd}{z_2} \overset{z_2 \to \infty}{=} 0$$

[1] Adapted from Jitendra Malik.

Hence, the vanishing points of lines on this plane lies on the vanishing line of the plane.

2. **Sphere Under Perspective Projection [30 pts].**[2]

(a) **[20 pts]** Under typical conditions, the silhouette of a sphere of radius r with center $(X, 0, Z)$ under planar perspective projection is an ellipse. Show that the eccentricity of this ellipse is $\frac{X}{\sqrt{X^2+Z^2-r^2}}$. Recall that, under perspective projection a point (X, Y, Z) in 3D space maps to $\left(f\frac{X}{Z}, f\frac{Y}{Z}\right)$ in the image, where f is the distance of the image plane from the pinhole.
Hint: There are different ways you can solve this. One line of attack would be to compute the lengths of major and minor axes of the projected ellipse, and compute eccentricity via $e = \sqrt{1-\left(\frac{\text{length of minor axis}}{\text{length of major axis}}\right)^2}$, but there could be other simpler alternatives as well.



P1 project to B: $x_1 = \frac{X_{P_1}}{Z_{P_1}} = f\tan(\frac{\pi}{2}-a-b)$

P2 project to x_2: $x_2 = \frac{X_{P_2}}{Z_{P_2}} = f\tan(\frac{\pi}{2}-(b-a))$

$x_1 = f\cot(a+b) = f\frac{\cot a\cot b - 1}{\cot a + \cot b}$

$x_2 = f\cot(b-a) = f\frac{\cot b\cot a + 1}{\cot a - \cot b}$

$\cot a = \frac{\sqrt{x^2+z^2-r^2}}{r}$ $\quad \cot b = \frac{x}{z}$.

$x_1 = f\frac{\frac{x\sqrt{x^2+z^2-r^2}}{zr} - 1}{\frac{z\sqrt{x^2+z^2-r^2}+xr}{zr}} = \frac{fx\sqrt{x^2+z^2-r^2} - zr}{z\sqrt{x^2+z^2-r^2}+xr}$

$x_2 = f\frac{\frac{x\sqrt{x^2+z^2-r^2}}{zr} + 1}{\frac{z\sqrt{x^2+z^2-r^2}-xr}{zr}} = f\frac{x\sqrt{x^2+z^2-r^2} + zr}{z\sqrt{x^2+z^2-r^2}-xr}$

major axis $l_1 = x_1 - x_2$

$l_1 = \frac{2fr\sqrt{x^2+z^2-r^2}}{z^2-r^2}$

when sphere projects to plane y-z, we can get minor axis, where $x=0$.

$l_2 = \frac{2fr\sqrt{z^2-r^2}}{z^2-r^2}$

$\therefore e = \sqrt{1-(\frac{l_2}{l_1})^2}$

$= \sqrt{1-(\frac{\sqrt{z^2-r^2}}{\sqrt{x^2+z^2-r^2}})^2}$

$= \frac{x}{\sqrt{x^2+z^2-r^2}}$

[2] Adapted from Jitendra Malik.

(b) **[10 pts]** Are there circumstances under which the projection could be a parabola or hyperbola? If yes, write down the conditions on X, Z, r and f, for parabola and hyperbola respectively; if no, explain why.

According to previous question, we calculated the eccentricity.

Hence, if $e > 1$, the projection is hyperbola.

$$e = \frac{x}{\sqrt{x^2+z^2-r^2}} > 1 \Rightarrow z^2 - r^2 < 0$$

$$\Rightarrow -r < z < r$$

if $e = 1$, the projection is parabola.

$$e = \frac{x}{\sqrt{x^2+z^2-r^2}} = 1 \Rightarrow z^2 - r^2 = 0$$

$$\Rightarrow z = \pm r$$