Submitted by: Shahram Najam Syed **Dated: 8th December, 2023**
Andrew ID: snsyed **HW6 (16-720)**

Q1:

a.

Ans: In the context of the n-dot-l lighting model, as represented in figure 2a, the illumination of a surface is determined by the scalar product of the surface normal ($\vec{n}$) and the light vector ($\vec{l}$). The mathematical formulation for the radiance L perceived froma Lambertian surface is given by the equation:

$$L = \frac{\rho_d I}{\pi} \cos\theta_i$$

Here, ρ_d denotes the diffuse reflectance of the surface, I is the intensity of the incident light, and $\cos\theta_i$ is equivalent to the dot product $\vec{n}.\vec{l}$, assuming both vectors are normalized.

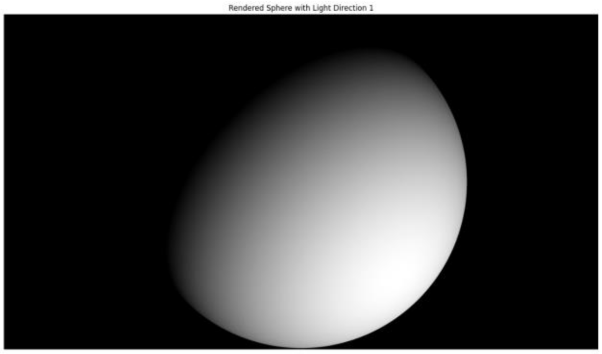
The dot product arises in the context as it compactly represents the cosine of the angle between the two vectors. Since the magnitude of the dot product decreases with increasing angle, this relationship captures the physical principle that surfaces appear less illuminated as the angle between the incident light and the normal to the surface increases.

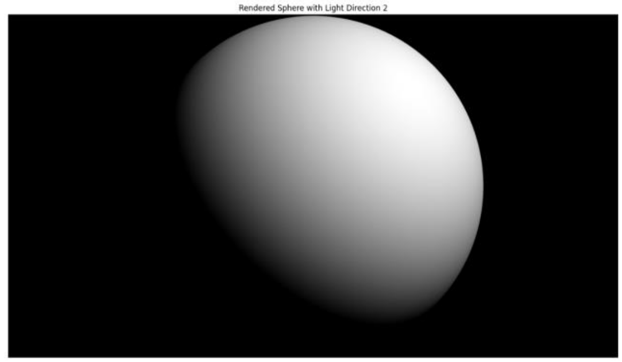
The irradiance received by a surface is proportional to the cosine of the angle of incidence, which is geometrically represented by the ratio of the projected area to the original area. As light strikes a surface obliquely, the illumination spreads over a larger area compared to when it strikes perpendicularly, reducing the irradiance and hence the apparent brightness.

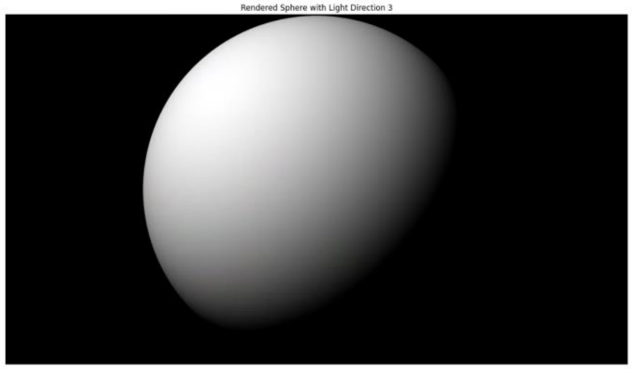
In the n-dot-l model, the viewing direction represented by vector $\vec{v}$ does not influence radiance L because the model is based on the assumption of Lambertian reflectance, where the surface reflects light equally in all directions. This isotropic scattering ensures that the perceived brightness of the surface is the same regardless of the observer's viewpoint. Therefore, only the angle between the incident light and the normal to the surface i.e. θ_i is responsible for the radiance observed, making the viewing direction irrelevant in this model.

b.

Ans: Following are the renditions of the n-dot-lighting:


Rendered Sphere with Light Direction 1


Rendered Sphere with Light Direction 2


Rendered Sphere with Light Direction 3

c.

Ans: Implemented as code (and submitted in the zip folder).

d.

Ans: In photometric stereo, the intensity matrix $\boldsymbol{I}$ is formulated as $I = L^T B$, where L is the lighting direction matrix of size 3X7, and B is the matrix of psuedonormals of size 3XP, which P is the number of pixels. Theoretically, the rank of I should be 3 as it is primarily determined by the three dimensional nature of the pseudonormals and the lighting directions.

However, when performing Singular Value Decomposition (SVD) on I, we often observe three significant singular values, suggesting a practical rank higher than 3. This discrepancy can be attributed to various factors in the image capture and processing steps:

i. Real-world scenarios often involve complex lighting conditions, including inter-reflections, that add additional independent components to the image data. The complexity can effectively increase the rank of the matrix I.
ii. The presence of noise and artifacts in the captured images can also contribute to the higher rank. Noise in the images introduces additional variance, which is reflected in the SVD as non-negligible singular values.
iii. The precision limitations of computational methods can sometimes lead to small but non-zero singular values, which contribute to an apparent increase in rank.
iv. Blurring effects, either due to camera optics or motion, can introduce subtle variations in pixel values images. Moreover, the resolution of the images can impact how these variations are captured and represented in the matrix I.

In the context of our SVD results, where the singular values are [72.41, 12.01, 8.43, 2.23, 1.51, 1.18, 0.84], it is evident that the rank is practically 7. The first three values are significantly higher, aligning with the expected theoretical rank. The presence of additional, smaller singular values indicate the influence of non-ideal factors.

e.

Ans: In the problem, we are given the equation $I = L^T B$ where I represents the intensity measyrements, L s the light source direction matrix, and B is the vector of pseudonormals that we aim to estimate. Given that we have more measurements (7 per pixels) than variable (3 per pixel), a least square solution is appropriate:

i. Step-1: Constructing Matrix A and Vector y:
From the equation $I = L^T B$, we identify our matrix A and vectors x and y in the $Ax = y$ for:

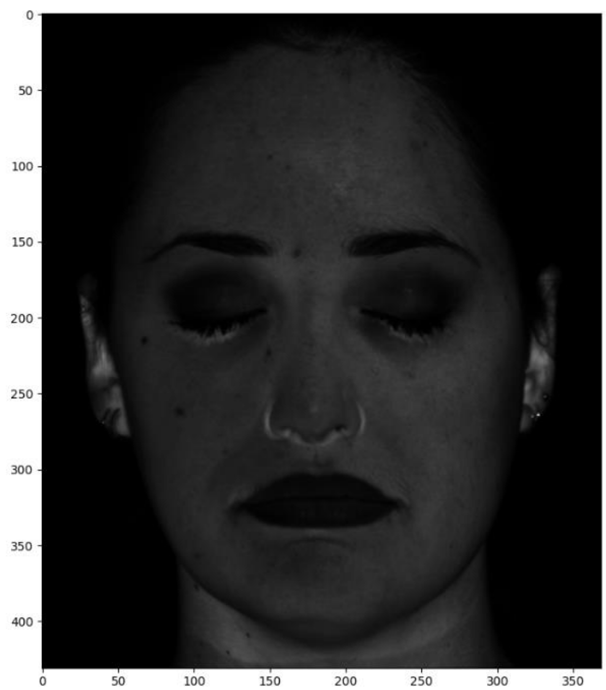
$$A = L^T$$
$$x = B \ (the\ unknown\ pseudonormals)$$
$$y = I \ (the\ intensity\ measurements)$$

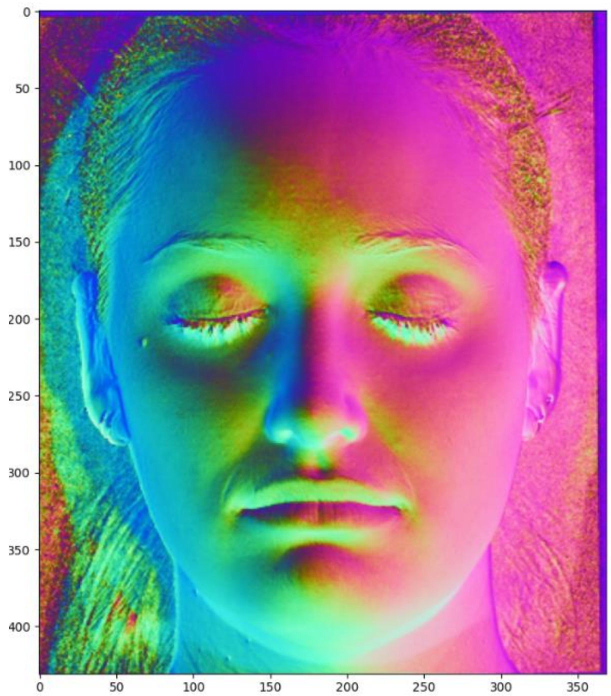
ii. Step-2: Solving for B using Least Squares:
Since $A(L^T)$ is not necessarily a square matrix, we cannot use direct matrix inversion. Instead we use the least squares method which minimizes the sum of the squares of the residuals.

iii. Step-3: Estimate per-pixel albedos and normal:
After solving for B, we can separate the albedo and normal information for each pixel from B. This involves normalizing the vectors in B to unit length (to get normal) and scaling them to the original vectors in B (to get albedos).

f.

Ans: Following are the Albedo and Normal generated after executing displayAlbedosNormals function:





In the observed image above (to the left), the anomalously brighter appearance of regions typically darker in real life (ear, nose, and neck), results from two primary factors. Firstly, the limitations of the n-dot-l model, with its assumption of singular, directional lighting, lead to inadequate handling of shadows and consequently inaccurate normal and albedo estimation in these areas. Secondly, this effect is compounded by inter-reflections, where light bouncing off surrounding surfaces add to the direct illumination, causing these typically shadowed regions to appear brighter in the image, deviating from ideal Lambertian reflection.

Similarly, the image above (on the right) captures the image normal, and accurately matches the curvature of the face, providing a fundamental linkage between the geometry of the face, showcased by varying color intensities in the rainbow cmap.

g.

Ans: To estimate the actual shape of the face represented as a 3D depth map, given by $z = f(x, y)$, and to understand the relation between the normal at any point (x, y) and the partial derivative of f, we proceed as follows:

i. The face's shape is modeled by the depth map function $z = (x, y)$, where z is the depth at coordinates (x, y). For any point on this surface, the normal vector $n = (n_1, n_2, n_3)$ can be related to the surface gradients, which are the partial derivatives of f with respect to x and y.

ii. The components of the surface gradient at a point (x, y) are given by $p = \frac{\partial z}{\partial x}$ and $q = \frac{\partial z}{\partial y}$. These gradients represent the slope of the surface in the x and y directions, respectively. Therefore, a small movement along x or y results in a change in height (Δz), which can be expressed in terms of p and q.

iii. The normal vector to the surface is perpendicular to all tangents on the surface. Since $(1, 0, p)$ and $(0, 1, q)$ are tangential vectors in the x and y directions, respectively, the normal vector n is parallel to their cross-product, resulting in n being proportional to $(-p, -q, 1)$.

iv. Normalizing the aforementioned vector, we get the unit normal $n = \frac{(-p,-q,1)}{\sqrt{1+p^2+q^2}}$, which translates to:

$$n_1 = \frac{-p}{\sqrt{1 + p^2 + q^2}}$$
$$n_2 = \frac{-q}{\sqrt{1 + p^2 + q^2}}$$
$$n_3 = \frac{1}{\sqrt{1 + p^2 + q^2}}$$

Thus relating the components of the normal vector n to the partial derivatives of z, we have:

$$\frac{\partial z}{\partial x} = p = -\frac{n_1}{n_3}$$
$$\frac{\partial z}{\partial y} = q = -\frac{n_2}{n_3}$$

In this way, the normal at any point (x,y) on the face's surface are directly related to the partial derivatives of the depth map function f(x,y), providing a fundamental linkage between the geometry of the face and its surface normal.

h.

Ans: Given the matrix g and the task of reconstructing it using its x and y gradients, g_x and g_y, we first of all start by defining the x-gradient g_x and y-gradient g_y. The x-gradient g_x is defined as $g_x(xi, yj) = g(xi+1, yj) - g(xi, yj)$ and similarly, the y-gradient g_y as $g_y(xi, yj) = g(xi, yj+1) - g(xi, yj)$. For the provided matrix g:

$$g = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

As defined above, for g_x each element is the difference between the right and left element in g:

$$g_x = \begin{bmatrix} 2-1 & 3-2 & 4-3 \\ 6-5 & 7-6 & 8-7 \\ 10-9 & 11-10 & 12-11 \\ 14-13 & 15-14 & 16-15 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

For g_y, each element is the difference between the bottom and top element in g:

$$g_y = \begin{bmatrix} 5-1 & 6-2 & 7-3 & 8-4 \\ 9-5 & 10-6 & 11-7 & 12-8 \\ 13-9 & 14-10 & 15-11 & 16-12 \end{bmatrix} = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \end{bmatrix}$$

1. Using g_x for the first row of g and g_y to reconstruct the rest of matrix g:
   a. First row using g_x

$$g(0,0) = 1$$
$$g(0,1) = g(0,0) + g_x(0,0) = 1 + 1 = 2$$
$$g(0,2) = g(0,1) + g_x(0,1) = 2 + 1 = 3$$
$$g(0,3) = g(0,2) + g_x(0,2) = 3 + 1 = 4$$

   b. Remaining rows using g_y:

$$g(1,0) = g(0,0) + g_y(0,0) = 1 + 4 = 5$$
$$g(1,1) = g(0,1) + g_y(0,1) = 2 + 4 = 6$$
$$g(1,2) = g(0,2) + g_y(0,2) = 3 + 4 = 7$$
$$g(1,3) = g(0,3) + g_y(0,3) = 4 + 4 = 8$$
$$g(2,0) = g(1,0) + g_y(1,0) = 5 + 4 = 9$$
$$g(2,1) = g(1,1) + g_y(1,1) = 6 + 4 = 10$$
$$g(2,2) = g(1,2) + g_y(1,2) = 7 + 4 = 11$$
$$g(2,3) = g(1,3) + g_y(1,3) = 8 + 4 = 12$$
$$g(3,0) = g(2,0) + g_y(2,0) = 9 + 4 = 13$$
$$g(3,1) = g(2,1) + g_y(2,1) = 10 + 4 = 14$$
$$g(3,2) = g(2,2) + g_y(2,2) = 11 + 4 = 15$$
$$g(3,3) = g(2,3) + g_y(2,1) = 12 + 4 = 16$$

   So,

$$g = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

2. Using g_y for the first column of g and g_x to reconstruct the rest of matrix g:
   a. First column using g_y

$$g(0,0) = 1$$
$$g(1,0) = g(0,0) + g_y(0,0) = 1 + 4 = 5$$
$$g(2,0) = g(1,0) + g_y(1,0) = 5 + 4 = 9$$
$$g(3,0) = g(2,0) + g_y(2,0) = 9 + 4 = 13$$

   b. Remaining columns using g_y:

$$g(0,1) = g(0,0) + g_x(0,0) = 1 + 1 = 2$$
$$g(0,2) = g(0,1) + g_x(0,1) = 2 + 1 = 3$$
$$g(0,3) = g(0,2) + g_x(0,2) = 3 + 1 = 4$$
$$g(1,1) = g(1,0) + g_x(1,0) = 5 + 1 = 6$$
$$g(1,2) = g(1,1) + g_x(1,1) = 6 + 1 = 7$$
$$g(1,3) = g(1,2) + g_x(1,2) = 7 + 1 = 8$$
$$g(2,1) = g(2,0) + g_x(2,0) = 9 + 1 = 10$$
$$g(2,2) = g(2,1) + g_x(2,1) = 10 + 1 = 11$$
$$g(2,3) = g(2,2) + g_x(2,2) = 11 + 1 = 12$$
$$g(3,1) = g(3,0) + g_x(3,0) = 13 + 1 = 14$$
$$g(3,2) = g(3,1) + g_x(3,1) = 14 + 1 = 15$$
$$g(3,3) = g(3,2) + g_x(3,2) = 15 + 1 = 16$$

   So,

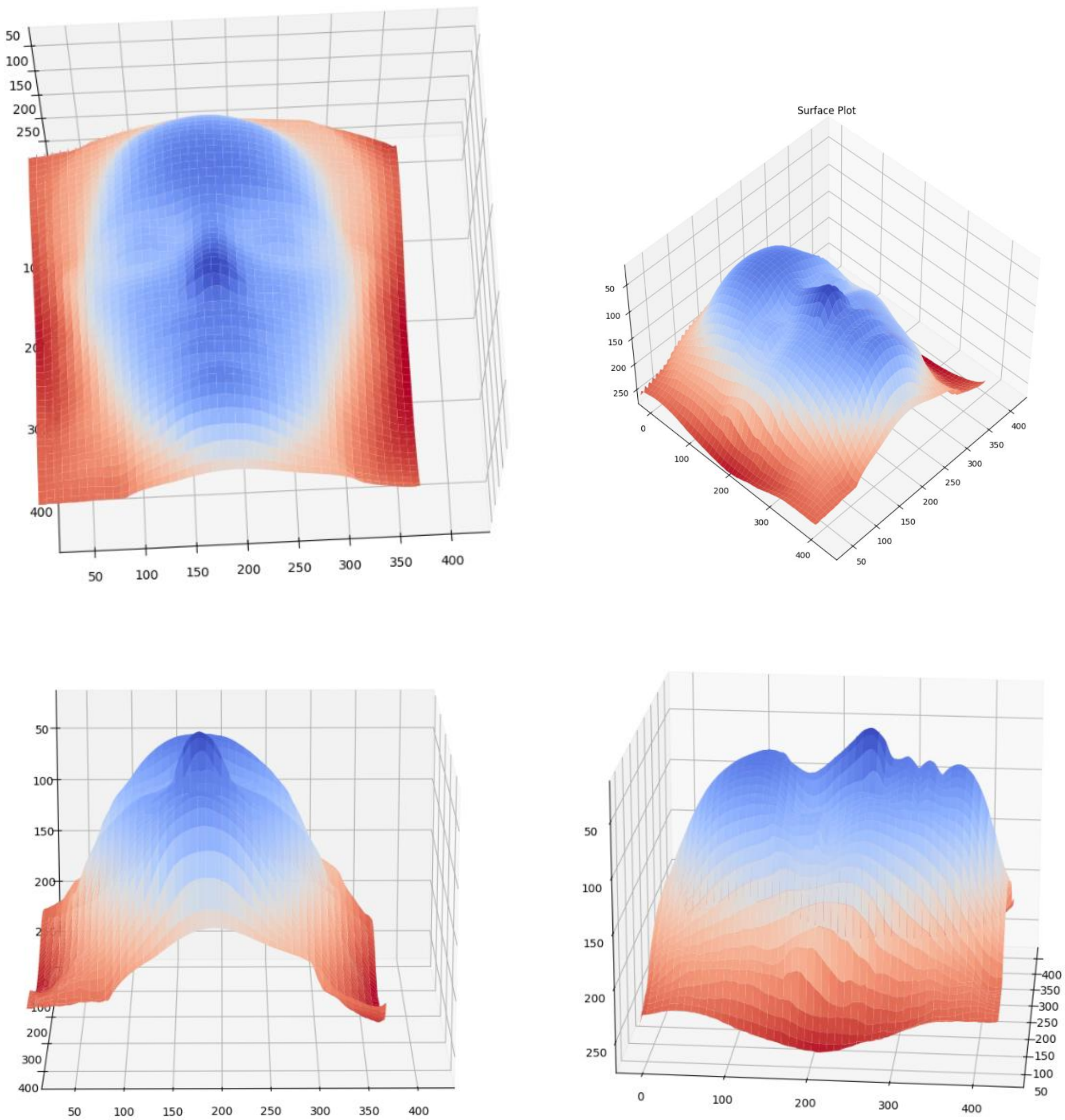
$$g = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

In both the methods, the reconstructed matrix g is the same.

This confirms that the gradients g_x and g_y are consistent and integrable, allowing for a faithful reconstruction of the original matrix g regardless of the starting direction of reconstruction.

i.

Ans: The function *estimateShape* was implemented and following is the plot of the surface area from varying viewpoints:



Q2:

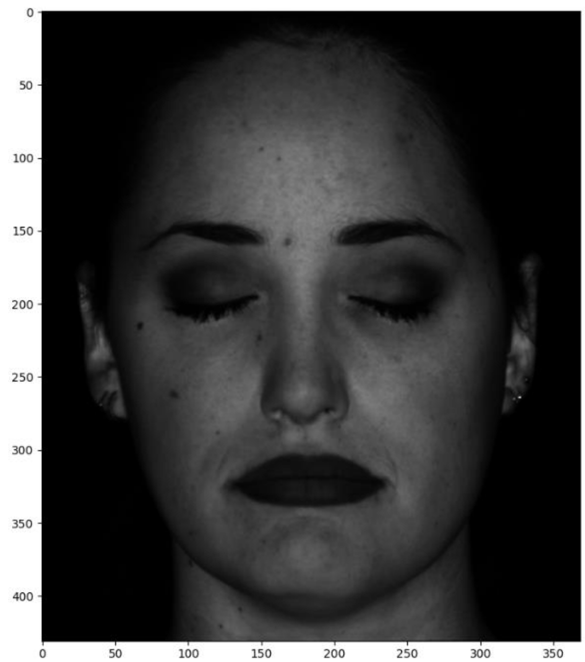
a.

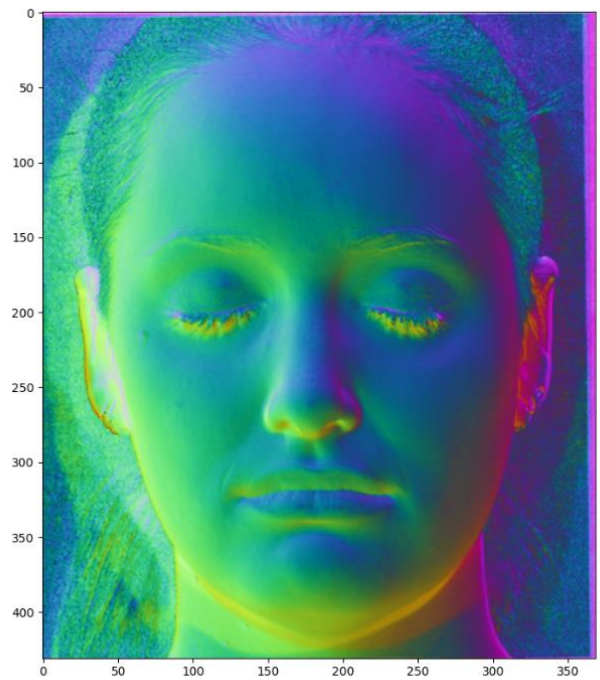
Ans: To factorize the matrix I into L^T and B with constraint that the rank of $I = L^T B$ is 3, we employ Singular Value Decomposition (SVD). The process is as follows:

- Perform SVD on the matrix I yielding $I = U\sum V^T$. Here, U and V are orthogonal matrices, and $\sum$ is a diagonal matrix of singular valyes.
- To enforce the rank-3 constraint, modify $\sum$ to $\sum_3$ by retaining only top three singular values and setting all others to zero. The truncation reduces the rank of I to 3.
- Form L^T by taking the first three columns of U and multiplying by the square root of $\sum_3$, i.e. $L^T = U_3\sqrt{\sum 3}$
- Similarly, constuct B by taking the first three columns of V and multiplying by the square root of $\sum_3$, followed by transposition, i.e. $B = \sqrt{\sum 3}\, {V_3}^T$

b.

Ans: The visualization of the albedo and normal are:





As compared to calibrated photometric stereo (q1), no significant or anomalous bright regions are observed in the albedos, while for the normal the color map identifying the geometric curvatures of the face distinctly.

c.

Ans: Following is the L_0 for ground truth lighting:

```
(sam_env) shahram95@shahram95:~/Desktop/shahram/HW6/hw6/src$ python q1.py
/home/shahram95/Desktop/shahram/HW6/hw6/src/q1.py:60: RuntimeWarning: invalid value encountered in sqrt
  z = np.sqrt(rad**2 - (x - center[0])**2 - (y - center[1])**2) + center[2]
[[-0.1418  0.1215 -0.069   0.067  -0.1627  0.      0.1478]
 [-0.1804 -0.2026 -0.0345 -0.0402  0.122   0.1194  0.1209]
 [-0.9267 -0.9717 -0.838  -0.9772 -0.979  -0.9648 -0.9713]]
```

While following is the L_{hat} (estimated):

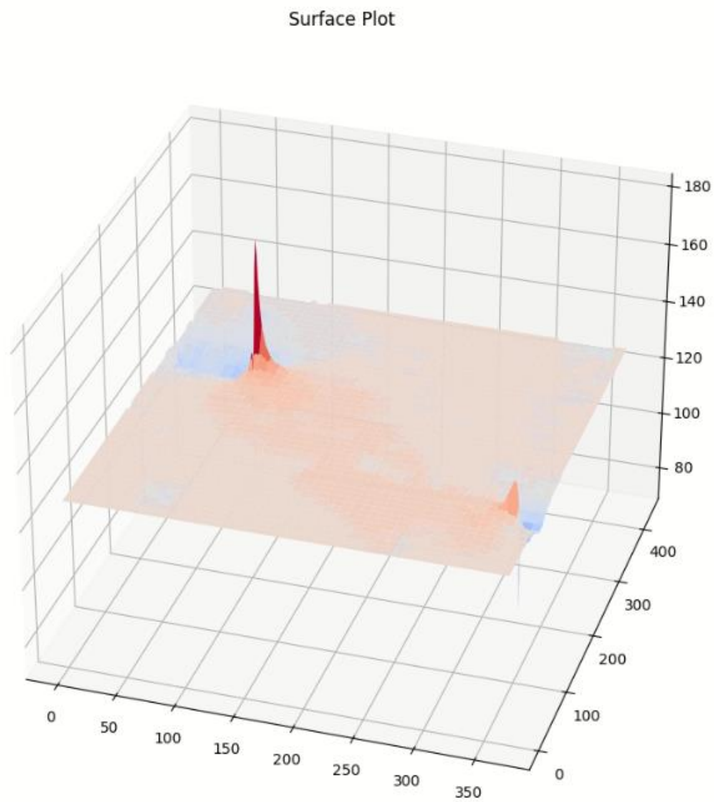
```
(sam_env) shahram95@shahram95:~/Desktop/shahram/HW6/hw6/src$ python q2.py
[[-2.85201979 -3.70224968 -2.29991163 -3.58159342 -3.42762044 -3.23473591
  -3.19937559]
 [ 0.89993439 -2.2191511   0.47629996 -0.59379374  2.22099759  0.45043184
  -0.74679075]
 [ 1.79851522  0.96528268  0.410125   -0.0210451  -0.28691153 -0.87230035
  -1.80219843]]
(sam_env) shahram95@shahram95:~/Desktop/shahram/HW6/hw6/src$
```

It is evident that both are distinctly different, which is primarily because of the L_{hat} determined by SVD factorization, which decomposes the image matrix without specific constraints on lighting.

To align L_{hat} with L_0while maintaining the same rendered image, we can introduce an orthogonal transformation such as a rotation matrix Q. Applying this matrix to both L_{hat} and B_{hat} such that $L_{hat}Q$ and $Q^T B_{hat}$ still yield the same product as $L_{hat}B_{hat}$. This adjustment changes the orientations and intensities in L_{hat} and B_{hat} without affecting the overall rendering, potentially bringing L_{hat} closer to the ground truth L_0.

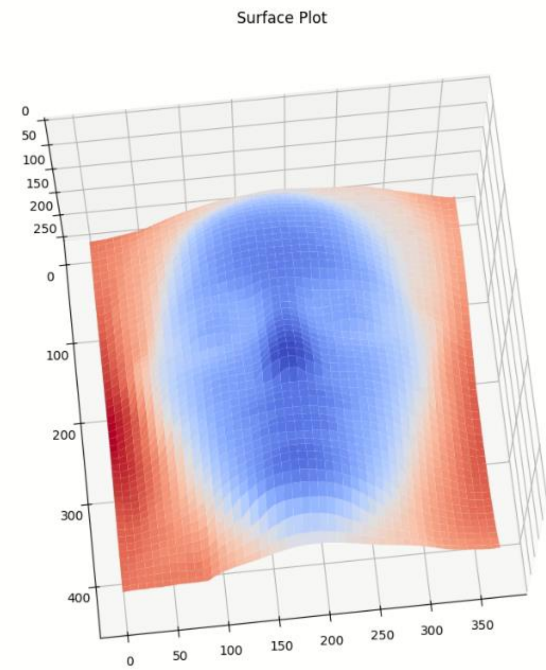
d.

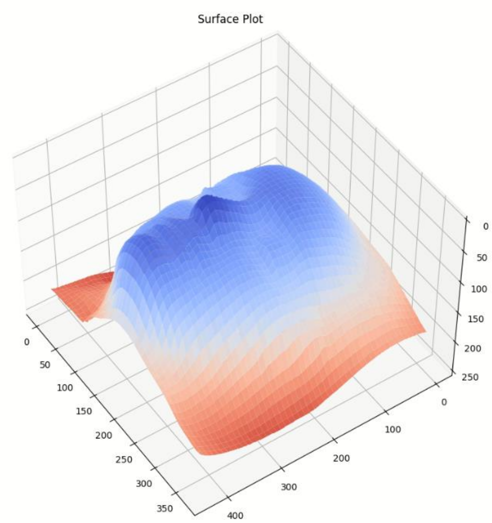
Ans: Following is the result of using the Frankot-Chellappa algorithm to reconstruct the 3D depth map, which does not look like a face compared to the results from the previous set of questions in calibrated photometric stereo:

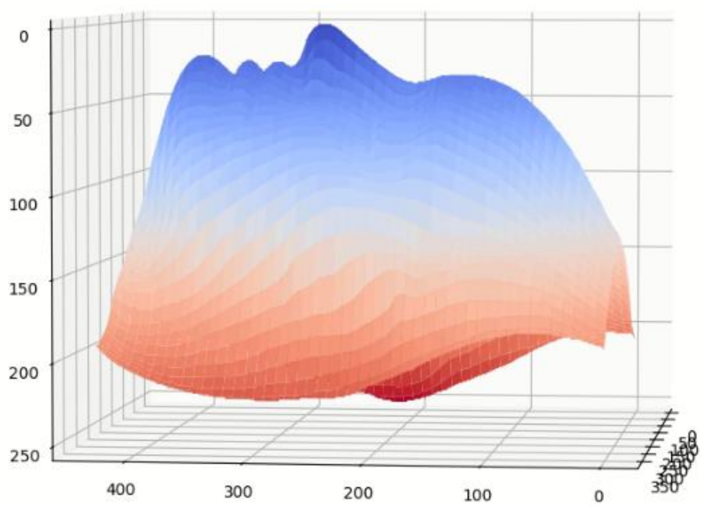


e.

Ans: Using *enforceIntegrability* by passing in psedonormals to estimate the shape with Frankot-Chellappa algorithm, we get the following surface plots, which are very similar (if not identical) to the output of photometric stereo in the previous question:

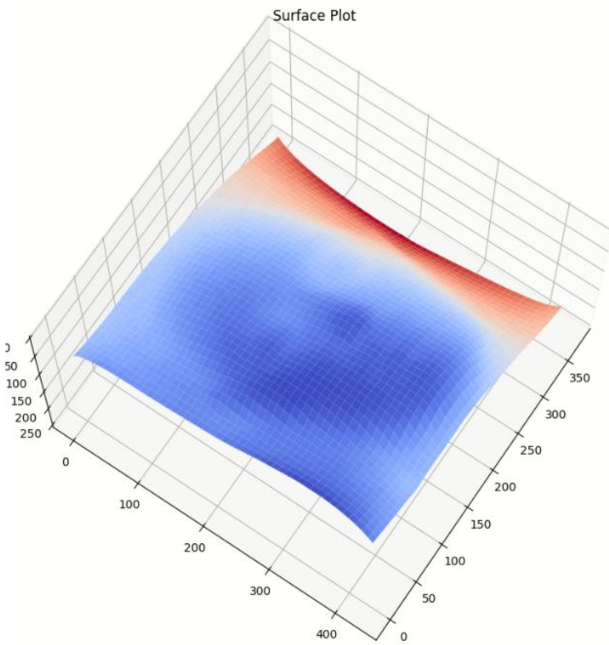
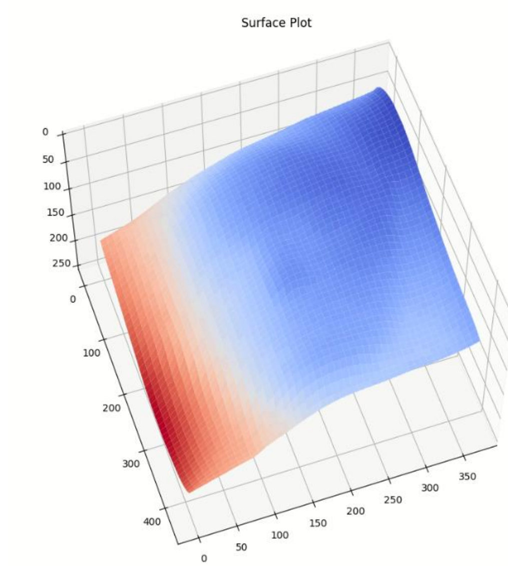


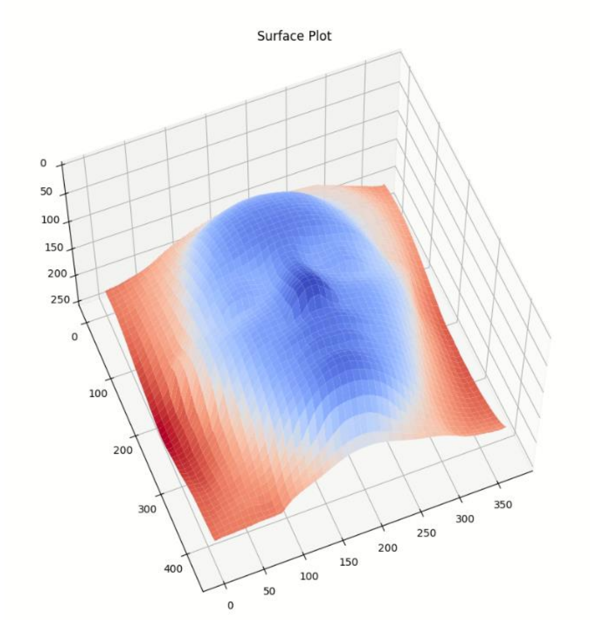
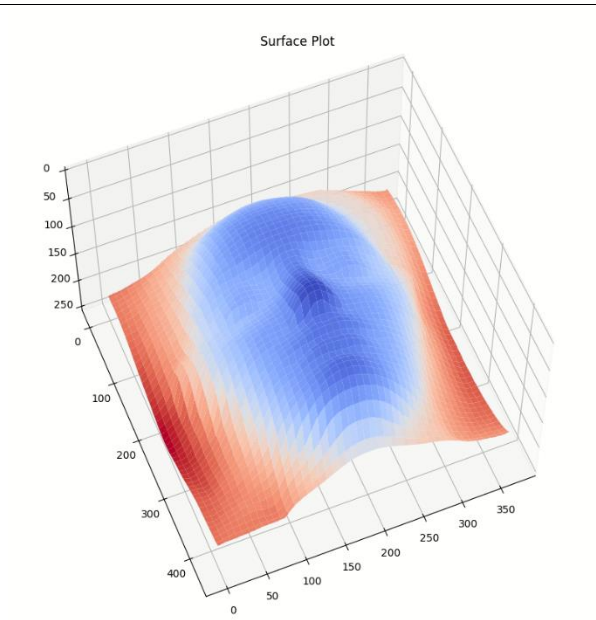


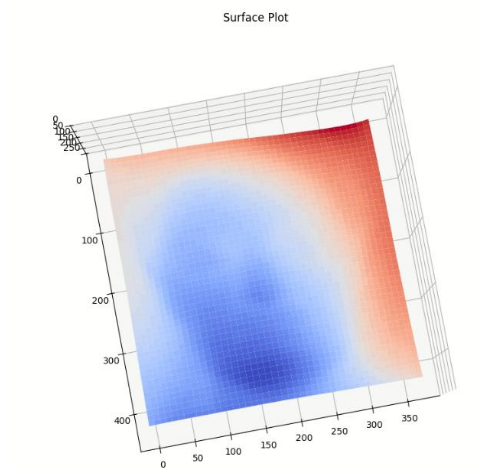
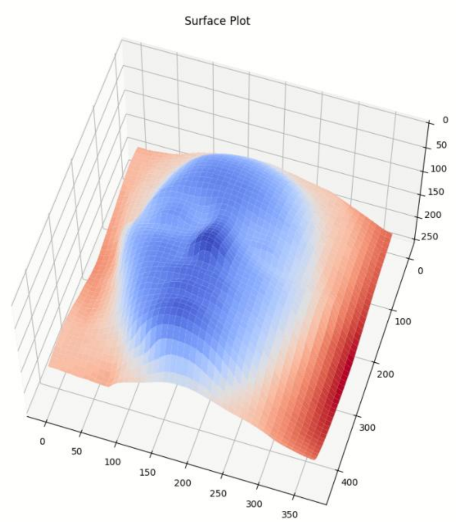
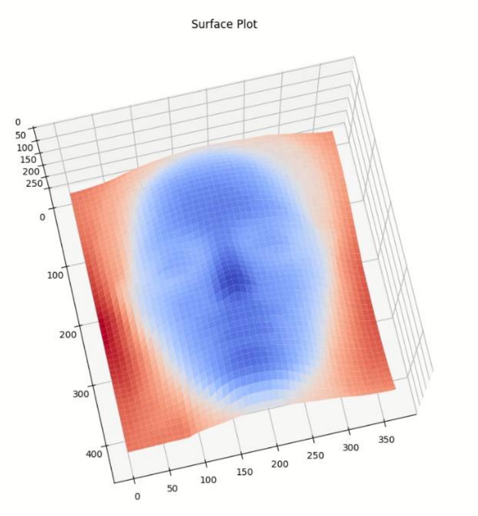


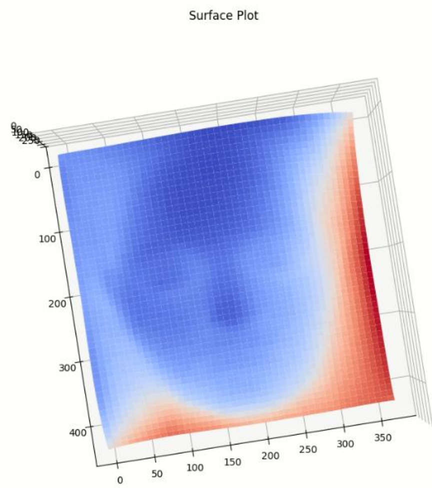
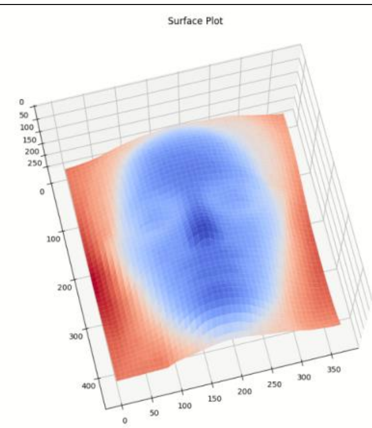
f.

Ans: Following is an ablation table for varying the parameters:

μ	v	λ	Surface plot
1	0.1	0.1	
-1	0.1	0.1	

0	0.1	0.1	
0.1	1	0.1	

0.1	-1	0.1	
0.1	0	0.1	
0.1	0.1	1	

0.1	0.1	-1	
0.1	0.1	5	

Observing the effects on the graphs above following conclusions about the effects were reached:

- **μ:** Varying **μ** primarily is affecting the horizontal stretching of the surface along the x-axis, where the visualization for different values of **μ** displayed how the surface appears to elongate (when **μ** is increased) or shrink horizontally (when **μ** is decreased).
- **v:** Changing v changes the vertical dimension of the surface along the y-axis. Increasing the **v** amplifies the vertical stretching of the surface along the y-axis, while decreasing the **v** shrinks the surface vertically.
- **λ:** Finally, **λ** intensifies the depth component of the surface. By increasing **λ** the overall relief of the surface becomes more pronounced, leading to a stronger impression of three-dimensionality, enhancing the surface's perceived depth, and vice versa for decreasing **λ**.

The name bas-relief ambiguity is named so because, upon visualizing the results of the transformations, the surfaces appear to exhibit varied depth and three dimensionality despite minimal actual changes in depth, similar to the effect of bas-relief in art. In bas-relief sculptures, subtle depth variations create a pronounced perception of three-dimensionality. Similarly, slightly varying the parameters can significantly change the perceived depth and orientation of the surface, creating an illusion of more profound relief than what actually exists.

g.

Ans: The bas-relief ambiguity involves adjusting the perceived depth of a surface. To design a transformation that makes the estimated surfaces as flat as possible i.e. appear flatter, we can:

- Reduce the value of **λ** significantly, closer to zero. This will minimize the z-component of the normal, effectively reducing perceived depth, thus flattening the surface in the visual representation.
- We can adjust **μ** and **ν** to control the horizontal and vertical stretching of the surface, to give a semblance of “flattening” in the depth.

h.

Ans: Acquiring more images from various lighting directions can indeed aid in resolving the ambiguity in 3D face reconstruction, primarily base:

- Additional images increase the coverage of lighting angles, illuminating the face differently and revealing more details about its surface geometry.
- Varied lighting conditions help distinguish between surface reflectance properties and shadows, thereby reducing ambiguity in interpreting surface normal and depth.

However, with this approach we would encounter increased computational complexity and diminishing returns as the number of images grows.