Computer Vision: 16720-A

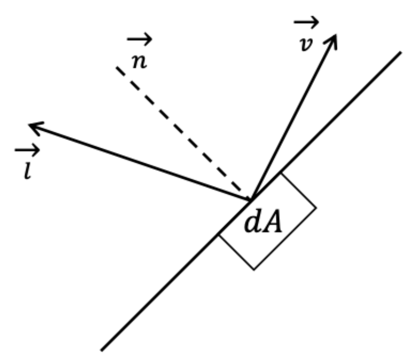
Kinjal Jain

Homework 6: Photometric Stereo

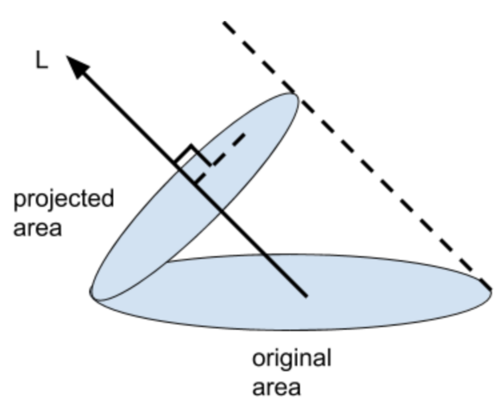
May 6, 2020

1 Calibrated photometric stereo

Q1.a Understanding n-dot-l lighting



Here, $\vec{l}$ is the vector corresponding to light source. $\vec{n}$ is the surface normal and $\vec{v}$ denotes the viewing direction. Based on physics, the amount of diffused light is maximum when the light source is at surface normal (head on illumination) because then dA is small and it gets all the light directly. However, as the angle θ is increased between $\vec{l}$ and $\vec{n}$, the amount of light is scattered among more area.



As can be seen in the image above (posted on Piazza by a TA), as $\vec{l}$ moves away from $\vec{n}$, θ increases, and the original area (dA) increases. Let I represent the source intensity and L denote the surface radiance.If we were to calculate L now, we need to get the projection of I from $\vec{l}$ onto $\vec{n}$ which will be I.cosθ We can see that cos$\theta = \vec{n}.\vec{l}$ when both $\vec{n}$ and $\vec{l}$ are unit vectors. Some constant is added to the equation of n-dot-l lighting model based on surface properties (ρ). So, the model is given by:

$$L = \rho I cos\theta/\pi \tag{1}$$

The viewing direction doesn't matter, as can be seen from the equation. The amount of surface radiance viewed from any direction $\vec{v}$ is same due to diffused reflection equally in all directions. The surface radiance value L only depends on surface properties, surface normal and the direction of light source.

Q1.b Rendering n-dot-l lighting

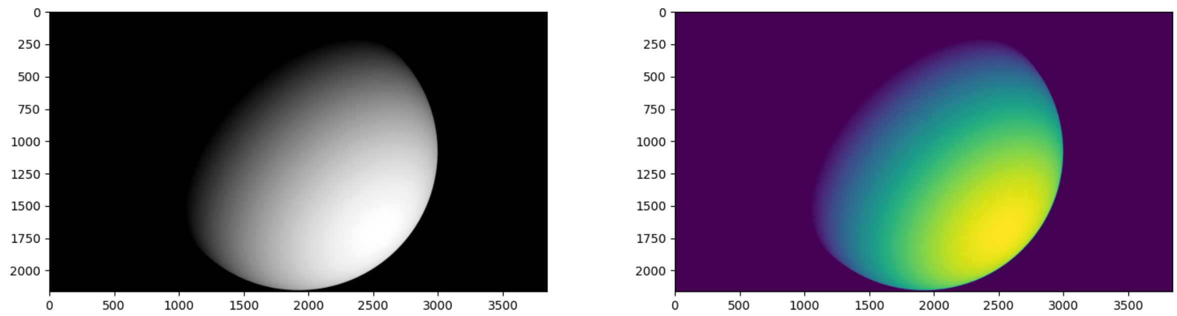


Figure 1: Light source direction: $(1,1,1)/\sqrt{3}$

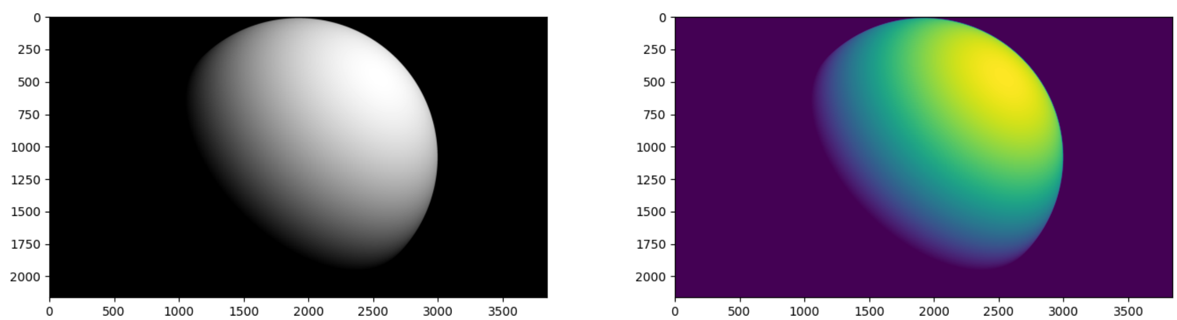


Figure 2: Light source direction: $(1,-1,1)/\sqrt{3}$

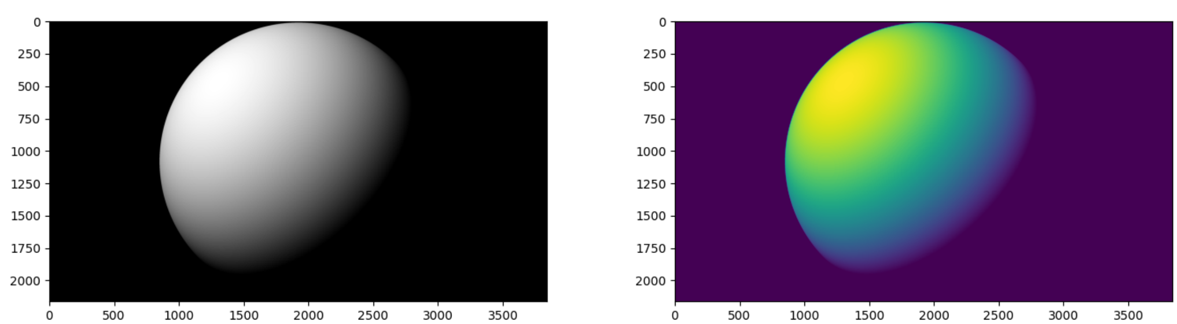


Figure 3: Light source direction: $(-1,-1,1)/\sqrt{3}$

Q1.d Initials

$$I = L^T.B \tag{2}$$

Here B is the set of pseudonormals in the image given by dimension ($3 \times$ P), and L is given by dimension (3×7). For both of them individually rank is 3. So, ideally rank of I should be 3 as well. Moreover, the space is spread in 3 dimensions too.
When we perform SVD on I, the eigen values are all greater than 0. The eigen values from the decompostion are [**79.36348099, 13.16260675, 9.22148403, 2.414729, 1.61659626, 1.26289066, 0.89368302**]
Here, based on 7 light source directions, we get rank of I to be 7 since all eigen values are greater than 0. So, the rank is not 3. This is because of the various non-idealities in the image capturing process like radiation propagation through the medium or surface, camera motion.

Q1.e Estimating pseudonormals

I have used pseudo inverse method directly to compute pseudonormals by using pseudo inverse of L^T. Or since, L^TB = I. We can say that it applies to Ax = y by keeping A= L^T and y=I and solve for B using least squares method.

Q1.f Albedos and normals

In the albedos image, the pixels around the ears, nostrils, eyelashes and neck seems to have wrong values ie. they have high intensities. This is because these areas obstruct the light path on reflection and it keeps bouncing back and forth in such corner regions, and camera might get the reflected light from various rays for the same pixel because of multiple bouncing of the ray in corner.

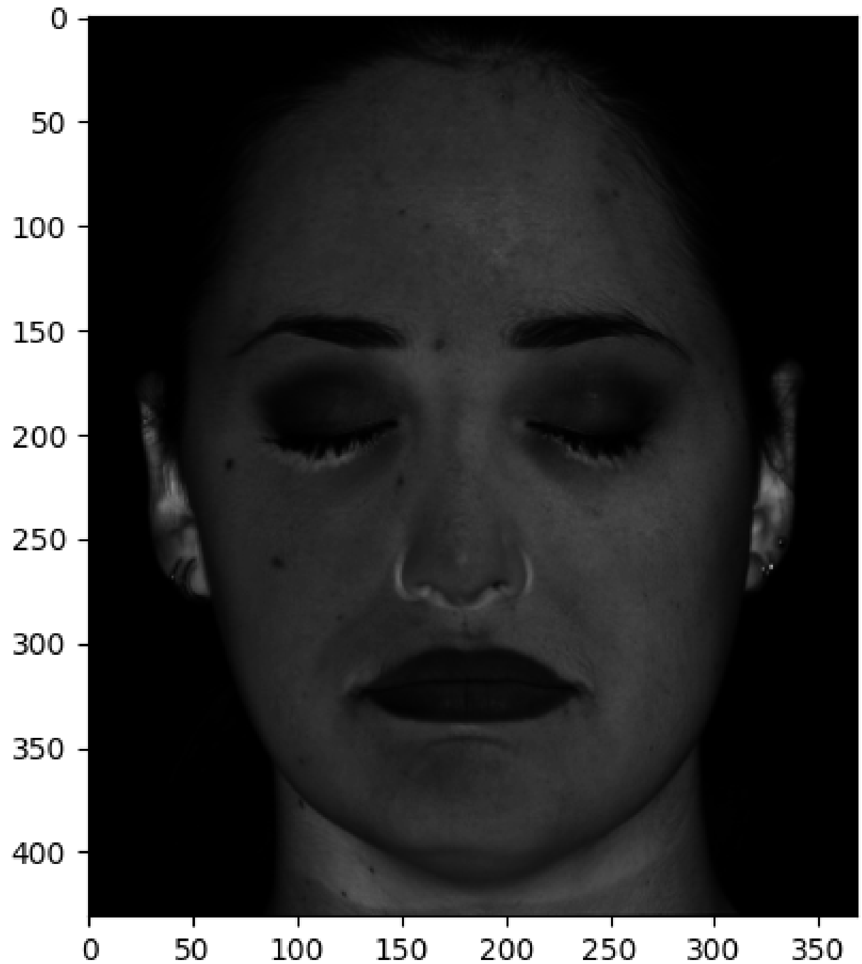


Figure 4: Albedos

The normals are pretty much nice, they convey the curvature of the face right at most places except the ones which are really bright.

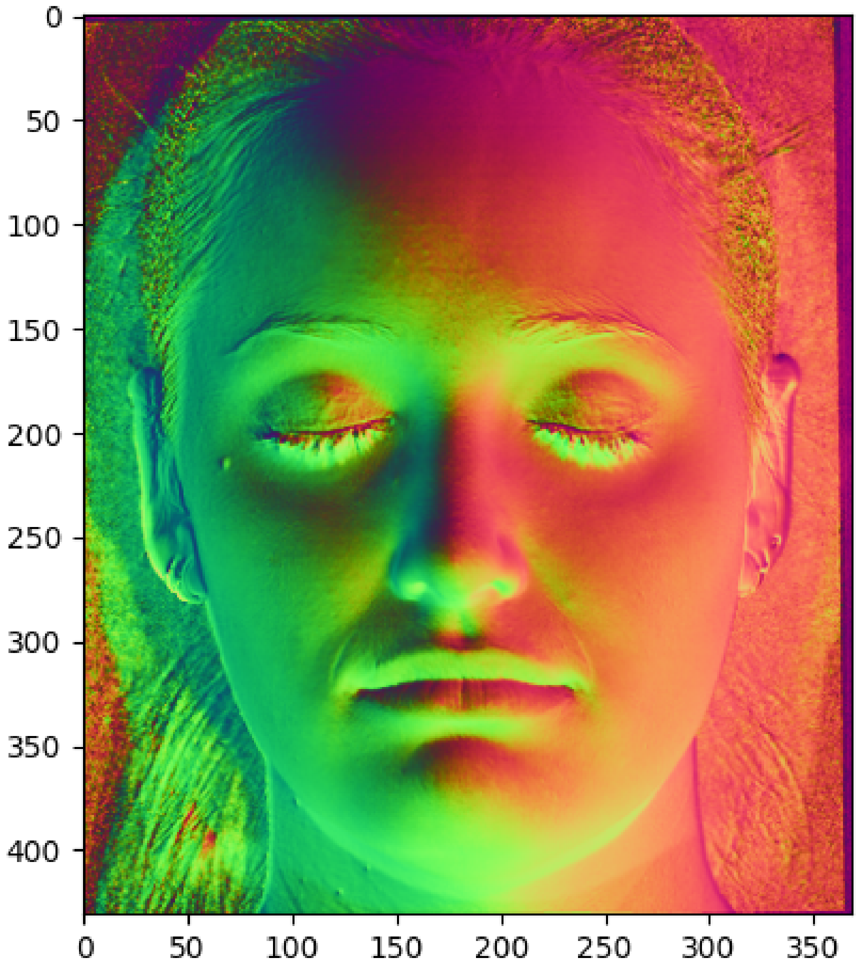


Figure 5: Normals

Q1.g Normals and depth

$$z = f(x, y) \tag{3}$$

Let the normal at the point (x, y) be n = (n1, n2, n3). Then zero level set of the surface (implicit form) is give by,

$$F(x, y, z) = z - f(x, y) = 0 \tag{4}$$

The surface normal (n) is given by the gradient of the implicit form of surface. So,

$$n = \delta F(x, y, z) = [-\frac{\partial f(x, y)}{\partial x}, -\frac{\partial f(x, y)}{\partial y}, 1] \tag{5}$$

We can convert n = (n1, n2, n3) to the form similar to previous equation by dividing first 2 components by the last one like:

$$n = (\frac{n1}{n3}, \frac{n2}{n3}, 1) \tag{6}$$

Then we can combine the two forms of n for each component and get:

$$\frac{n1}{n3} = -\frac{\partial f(x, y)}{\partial x} \tag{7}$$

$$\frac{n2}{n3} = -\frac{\partial f(x, y)}{\partial y} \tag{8}$$

which can be rewritten to get:

$$\frac{\partial f(x, y)}{\partial x} = -\frac{n1}{n3} \tag{9}$$

$$\frac{\partial f(x, y)}{\partial y} = -\frac{n2}{n3} \tag{10}$$

Q1.h Understanding integrability of gradients

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

Given that,

$$g_x(x_i, y_j) = g(x_{i+1}, y_j)g(x_i, y_j) \tag{12}$$
$$g_y(x_i, y_j) = g(x_i, y_{j+1})g((x_i, y_j) \tag{13}$$

Using these equations and padding an extra row and a column with 0s, we can write g_x and g_y as:

$$g_x = \begin{pmatrix} 1 & 1 & 1 & -4 \\ 1 & 1 & 1 & -8 \\ 1 & 1 & 1 & -12 \\ 1 & 1 & 1 & -16 \end{pmatrix} \tag{14}$$

$$g_y = \begin{pmatrix} 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \\ -13 & -14 & -15 & -16 \end{pmatrix} \tag{15}$$

Now, to generate g back. 1. When g_x is used to construct the first row of g, then g_y is used to construct the rest of g, we get:

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

2. When g_y is used to construct the first column of g, then g_x is used to construct the rest of g, we get:

$$g = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} \tag{17}$$

So, these two ways result in same value of g. Integrability implies that the value of g estimated in both these ways (or any other way you can think of) is the same. So, these g_x and g_y are integrable.

However, these gradients g_x and g_y can be made non-integrable. For instance, if

$$g_x = \begin{pmatrix} 1 & 2 & 1 & -4 \\ 2 & 2 & 1 & -8 \\ 1 & 2 & 1 & -12 \\ 1 & 2 & 1 & -16 \end{pmatrix} \tag{18}$$

$$g_y = \begin{pmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 4 & 4 & 4 & 4 \\ -13 & -14 & -15 & -16 \end{pmatrix} \tag{19}$$

Then, 1. When g_x is used to construct the first row of g, then g_y is used to construct the rest of g, we get:

$$g = \begin{pmatrix} 1 & 2 & 4 & 5 \\ 5 & 6 & 8 & 9 \\ 7 & 7 & 9 & 10 \\ 8 & 11 & 13 & 14 \end{pmatrix} \tag{20}$$

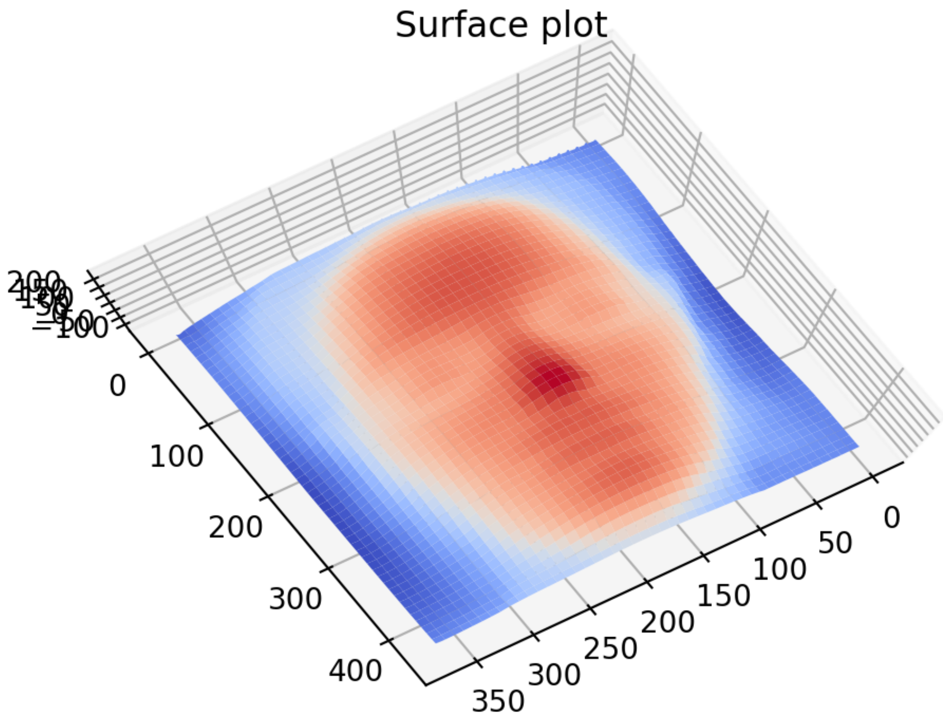
2. When g_y is used to construct the first column of g, then g_x is used to construct the rest of g, we get:

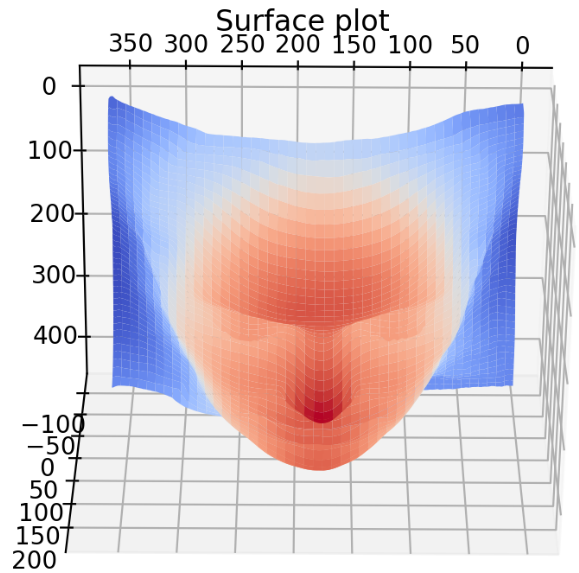
$$g = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 7 & 9 & 10 \\ 6 & 7 & 11 & 12 \\ 10 & 11 & 15 & 16 \end{pmatrix} \tag{21}$$

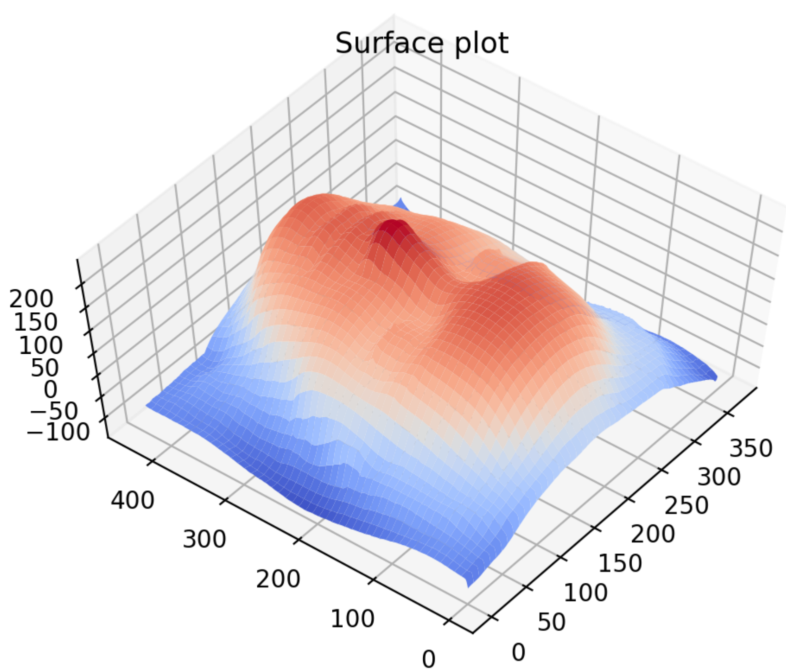
The two g matrices obtained now are different, which means these gradients were non-integrable.

The gradients estimated in the way of previous part may be non-integrable because a surface might not be smooth. Other things include non-idealities of image capturing process like camera movement and vibrations, other noises captured by pixels, camera reading out photons from sensors, etc.

Q1.i Shape estimation




Surface plot
350
300
250
200
150
100
50
0
0
100
200
300
400
−100
−50
0
50
100
150
200


Surface plot
200
150
100
50
0
−50
−100
400
300
200
100
0
350
300
250
200
150
100
50
0

2 Uncalibrated photometric stereo

Q2.a Uncalibrated normal estimation

$$I = L^T.B \tag{22}$$

but we don't know either L or B. However, we know that rank of I should be 3.
So, we can perform SVD on I with constraint that with the estimated L and B, the rank of I will be 3.
SVD for M is given as M = U $\sum$ V^T , and all singular values except the top k from $\sum$ can be set to 0 to get the matrix $\sum$ with desired rank and reconstitute M = U $\sum$ V^T.

So, in our case the decomposition will be done as I = U $\sum$ V^T and then all singular values of $\sum$ apart from first 3 are set to 0. Once, this is done some part of sigma is combined with U and some with V^T to get L and B.

Q2.b Calculation and visualization

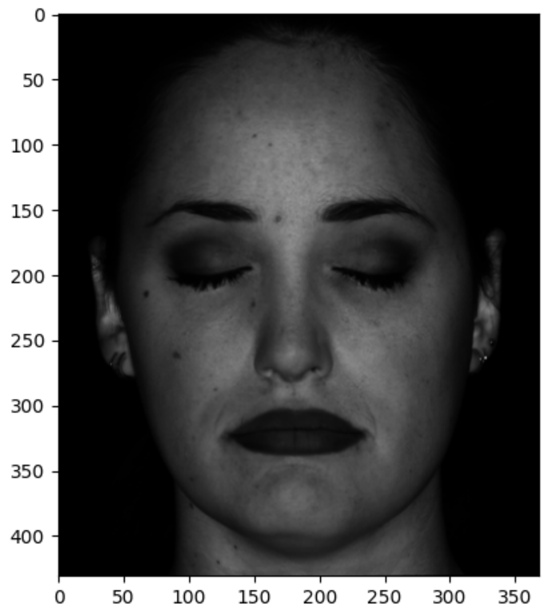


Figure 6: Albedos

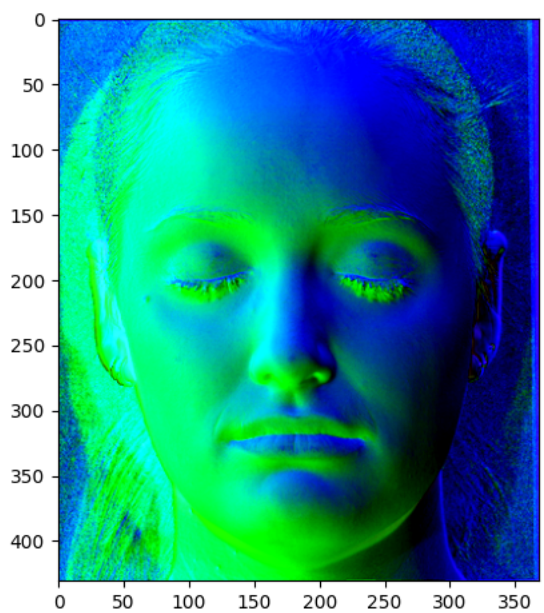


Figure 7: Normals

Q2.c Comparing to ground truth lighting

By changing the way in which we combine $\sum$ with U and V^T, the estimate of L will change but resulting albedos and normals look the same. However, these Ls are not the same as L_0.

By combing $\sqrt{\sum}$ with U and B by combining $\sqrt{\sum}$ with V^T, L^T = U$\sqrt{\sum}$ and B = $\sqrt{\sum}V^T$, L =

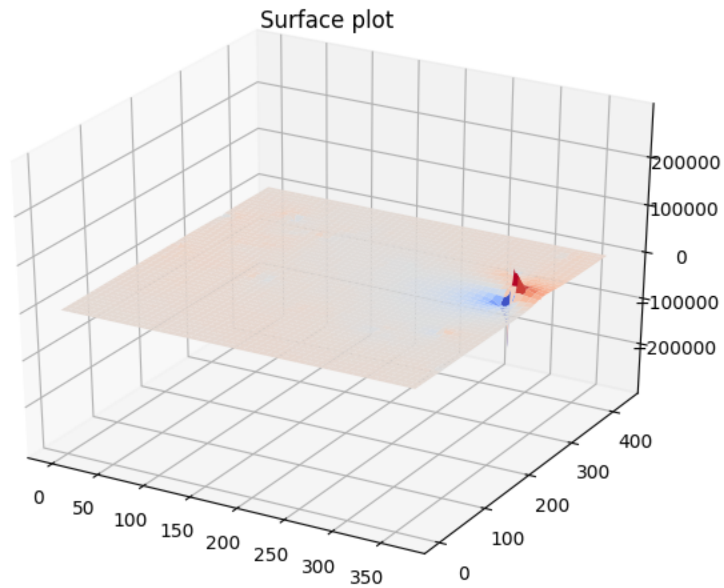
```
[[-2.99267472 -3.86998525 -2.40803005 -3.74500806 -3.59135539 -3.38666635 -3.3525448 ]
[ 0.94780484 -2.31708946  0.49911094 -0.62599426  2.32568155  0.46605103 -0.79271078]
[ 1.87934697  1.01461663  0.42942606 -0.01730299 -0.3107729  -0.91273581 -1.8830081 ]]
```

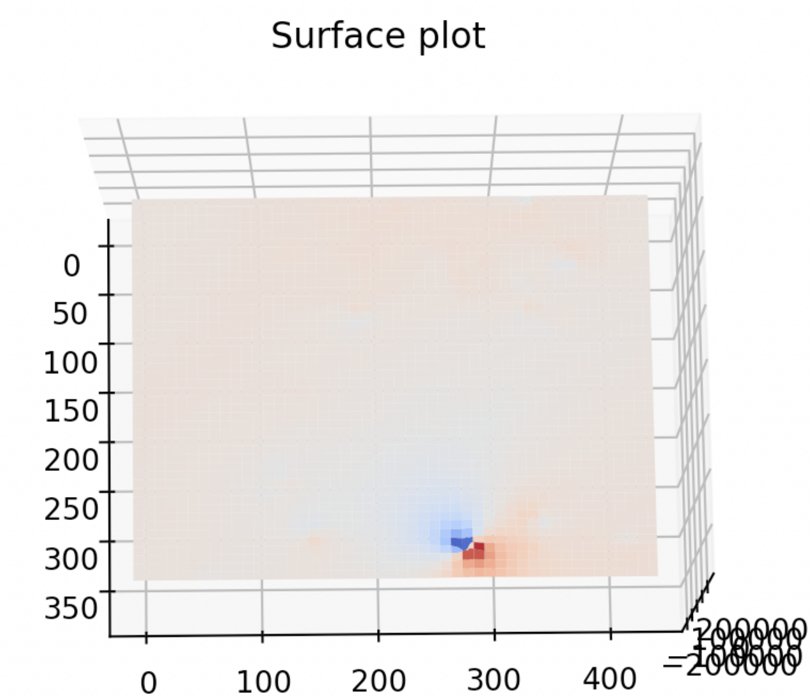
L_0 =

```
[[-0.1418 -0.1804 -0.9267  0.1215 -0.2026 -0.9717 -0.069 ]
 [-0.0345 -0.838   0.067  -0.0402 -0.9772 -0.1627  0.122 ]
 [-0.979   0.      0.1194 -0.9648  0.1478  0.1209 -0.9713]]
```

I can also combine the whole $\sum$ with U to get L and keep B as just V^T. This redistribution of $\sum$ doesn't change the rendered image even though in this way L^T = U$\sum$ and B = V^T.

Q2.d Reconstructing the shape, attempt 1





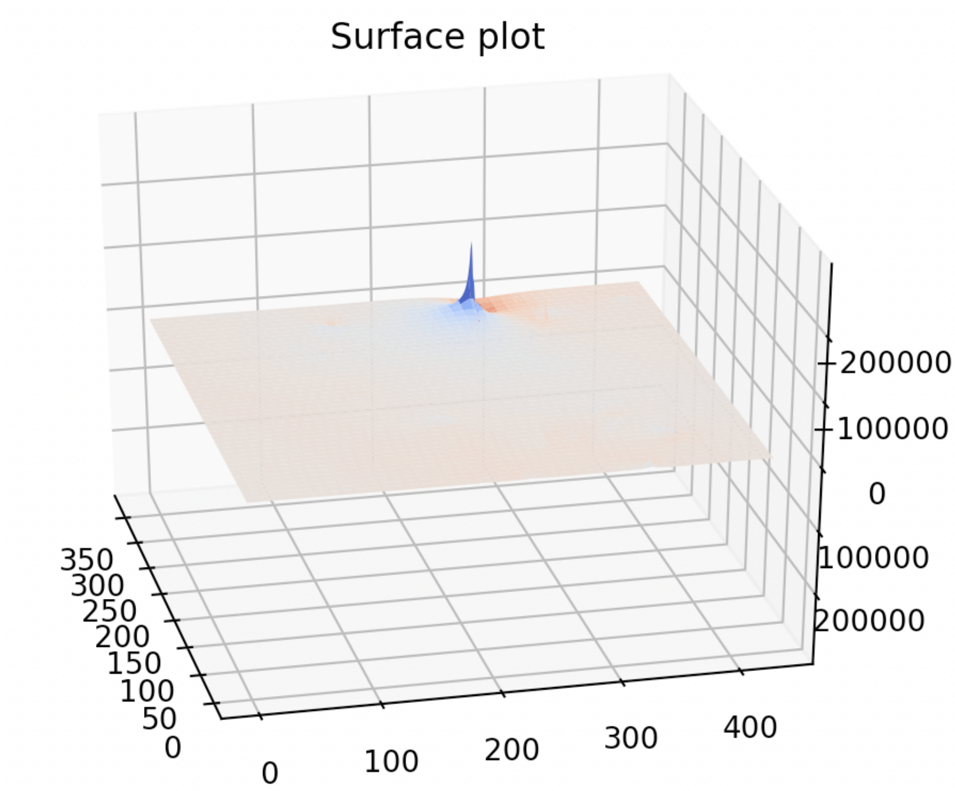
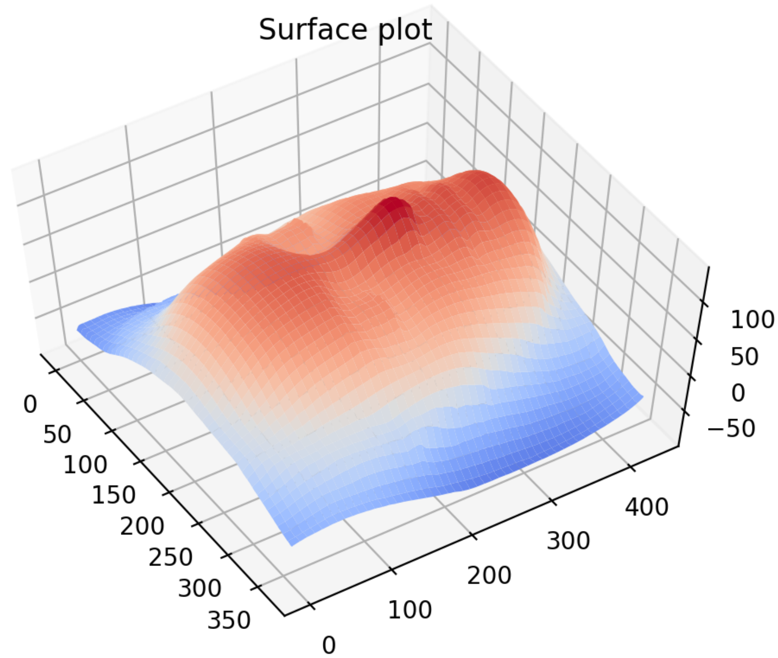
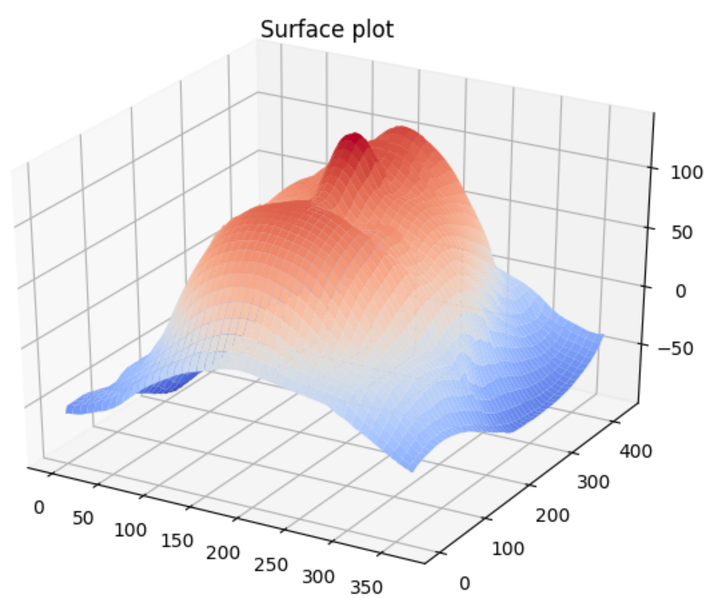


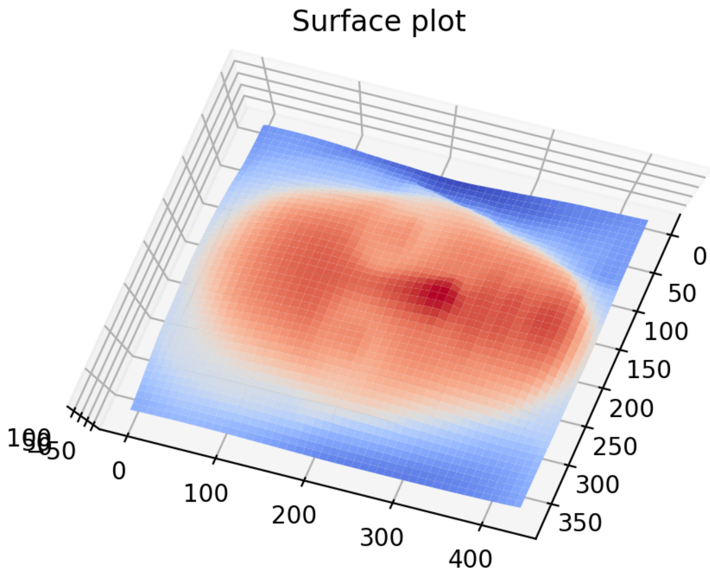
Figure 8: 3D depth map using Frankot-Chellappa algorithm

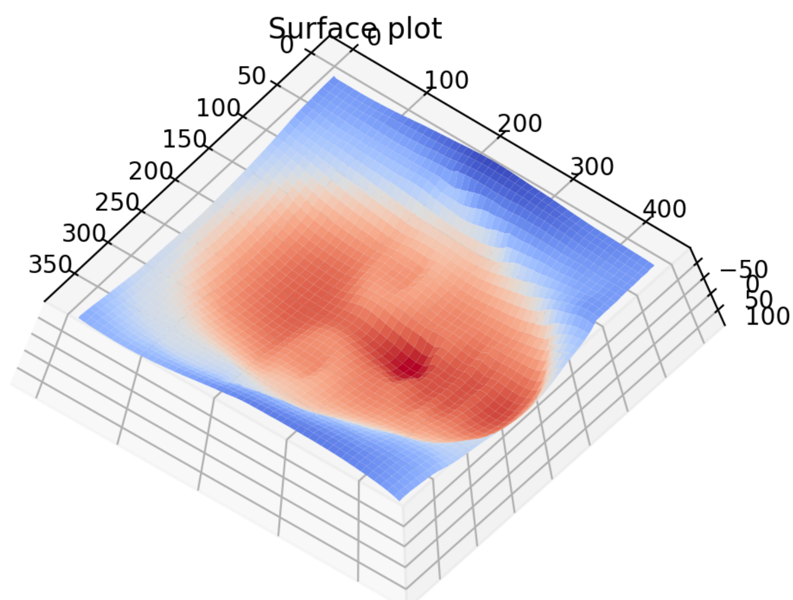
Q2.e Reconstructing the shape, attempt 2

Some views of the 3D depth map after enforcing integretibility are shown below. Yes, they do look like the map from calibrated photometric stereo.






Surface plot
0
50
100
150
200
250
300
350
0
100
200
300
400


Surface plot
0
50
100
150
200
250
300
350
0
100
200
300
400
−50
0
50
100

Even the normals and albedo look similar to the ones obtained from calibrated photometric stereo.

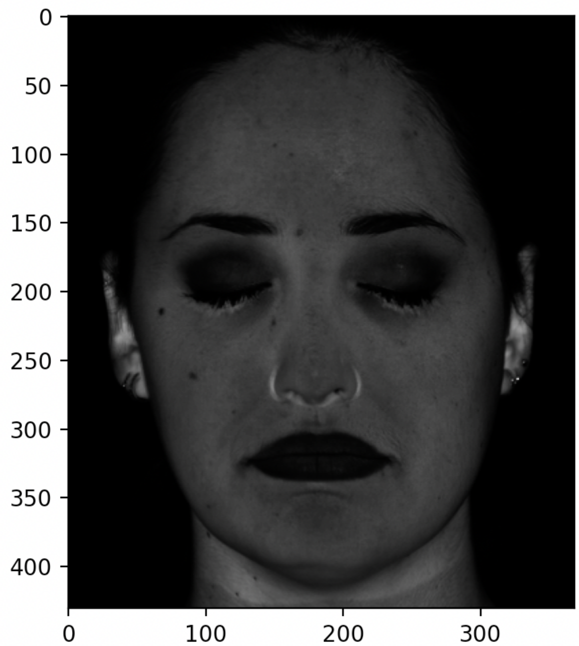


Figure 9: Albedos

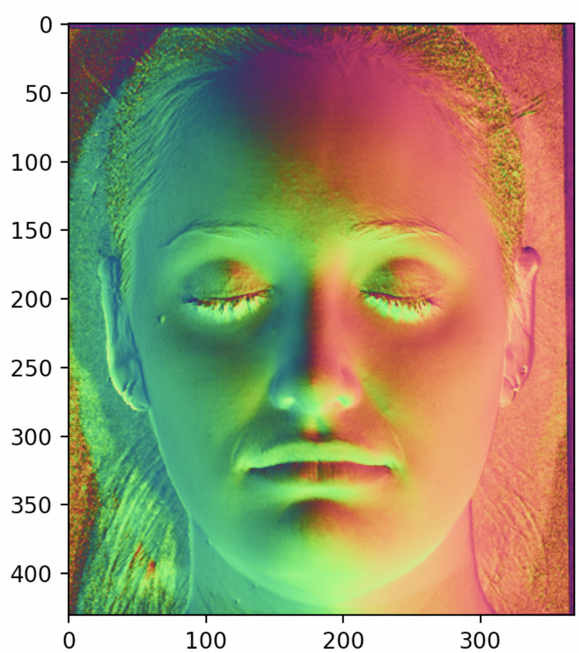
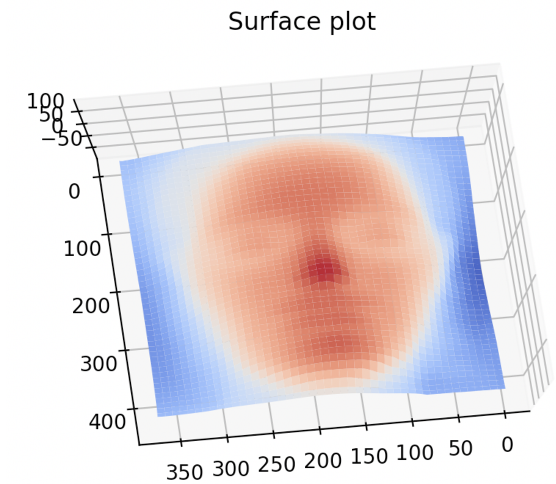
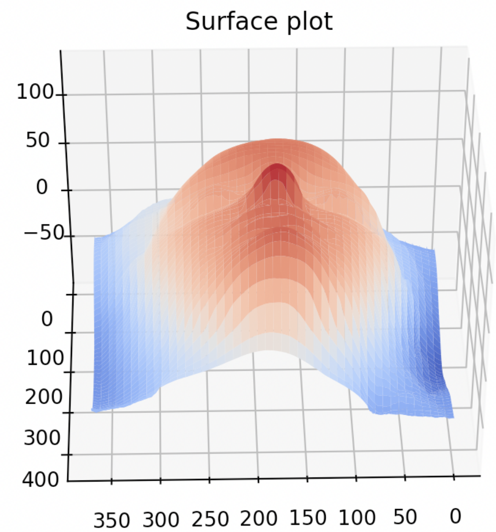


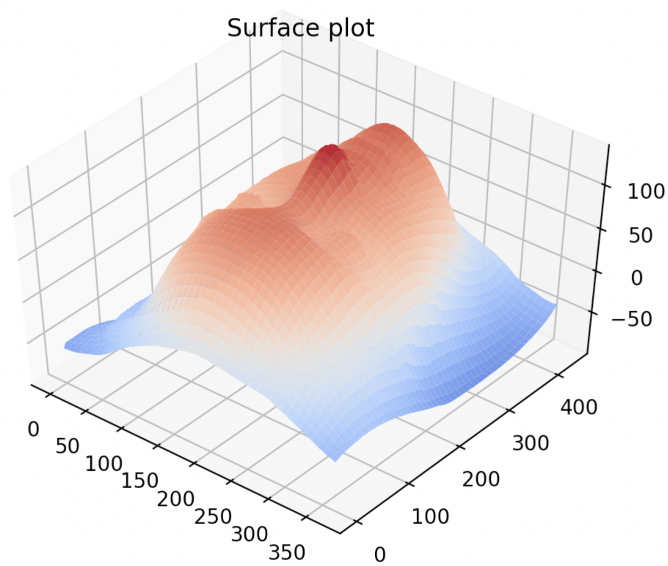
Figure 10: Normals

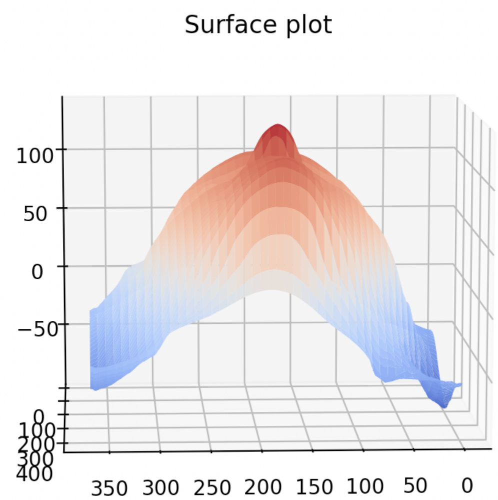
Q2.f Why low relief?

The surface pictures without bas relief transformations (i.e.when μ=0, ν=0 and λ=1)is as shown below:



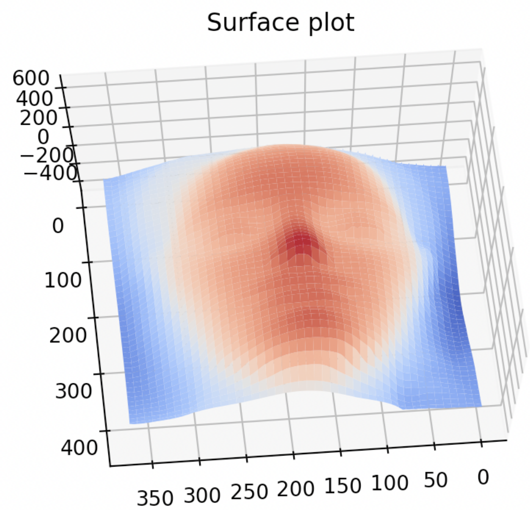


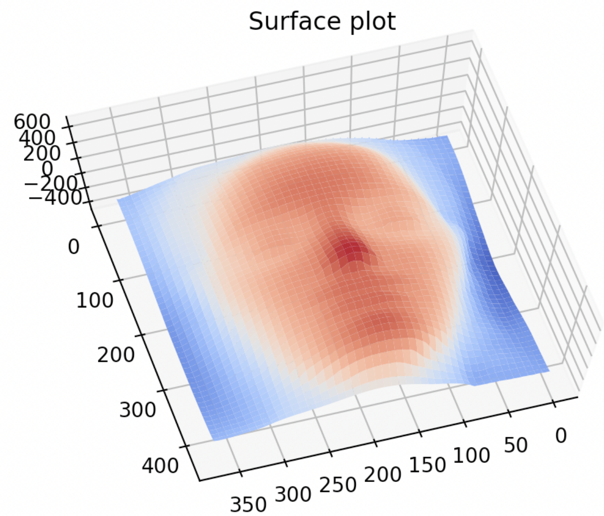


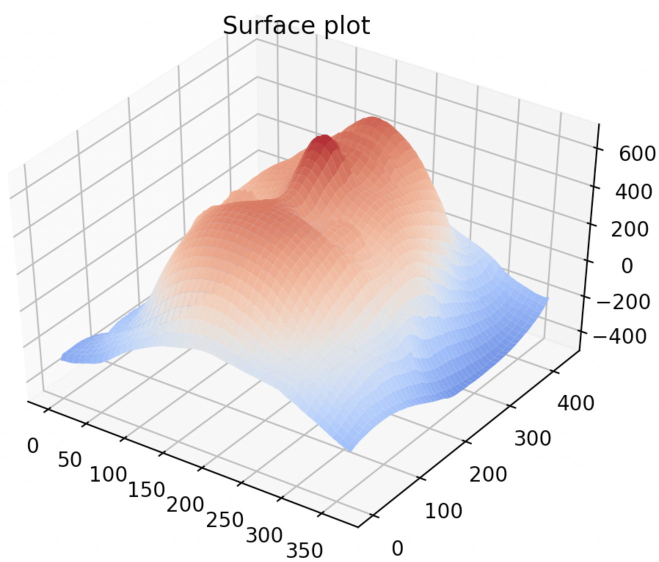


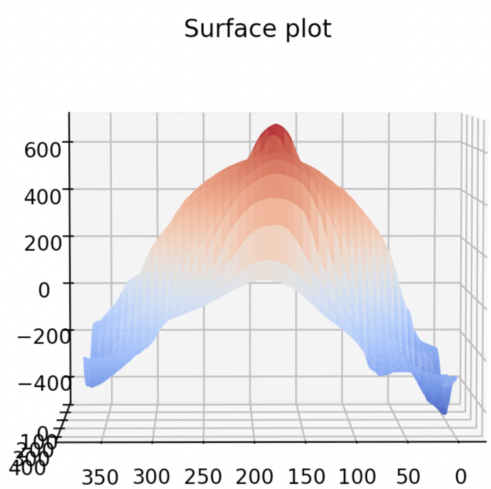
μ, ν and λ affect the surface in following ways:

1. When λ=5 keeping μ=0 and ν=0, it can be seen below that the image is stretched, as depth increased from 100 to 600 on the positive side and -50 to -400 in the negative side.

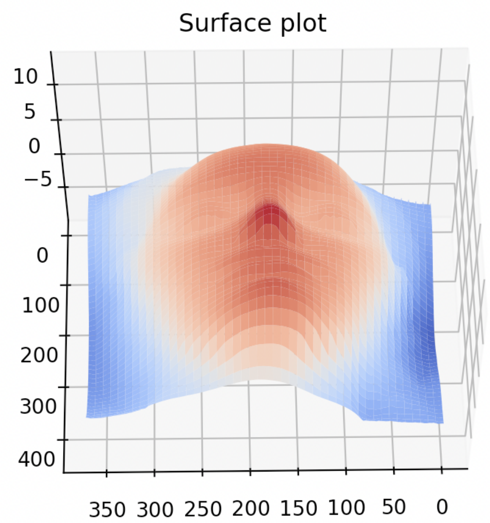


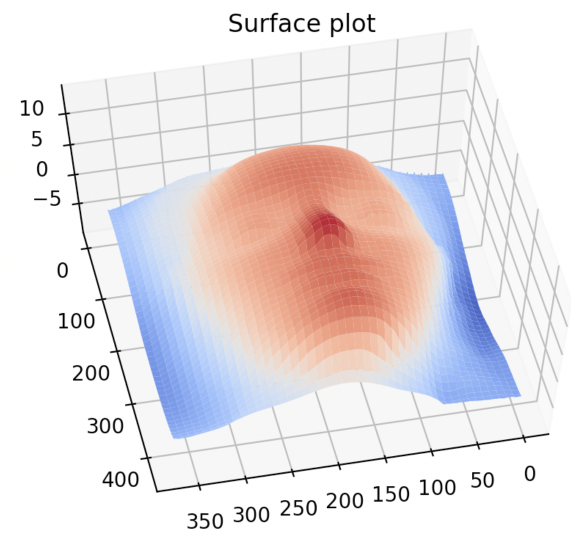


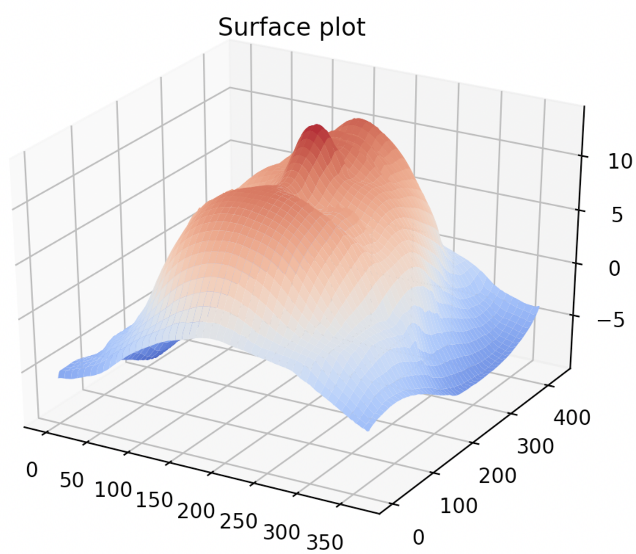


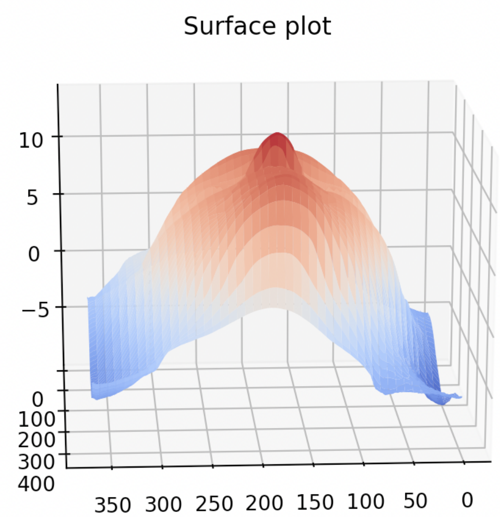


2. When λ=0.1 keeping μ=0 and ν=0, it can be seen below that the image is squished, as depth decresed from 100 to 10 on the positive side and -50 to -5 in the negative side.

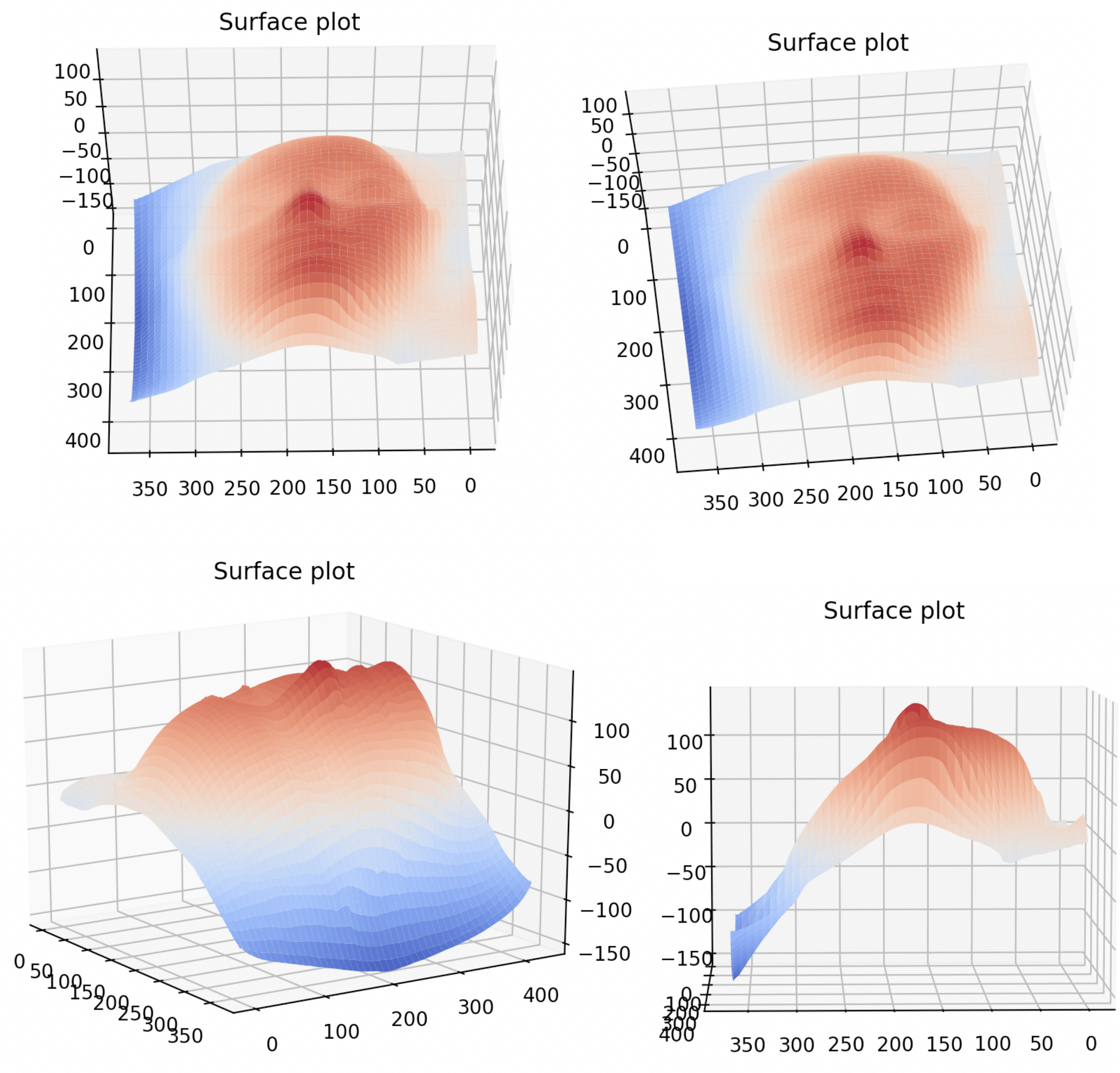




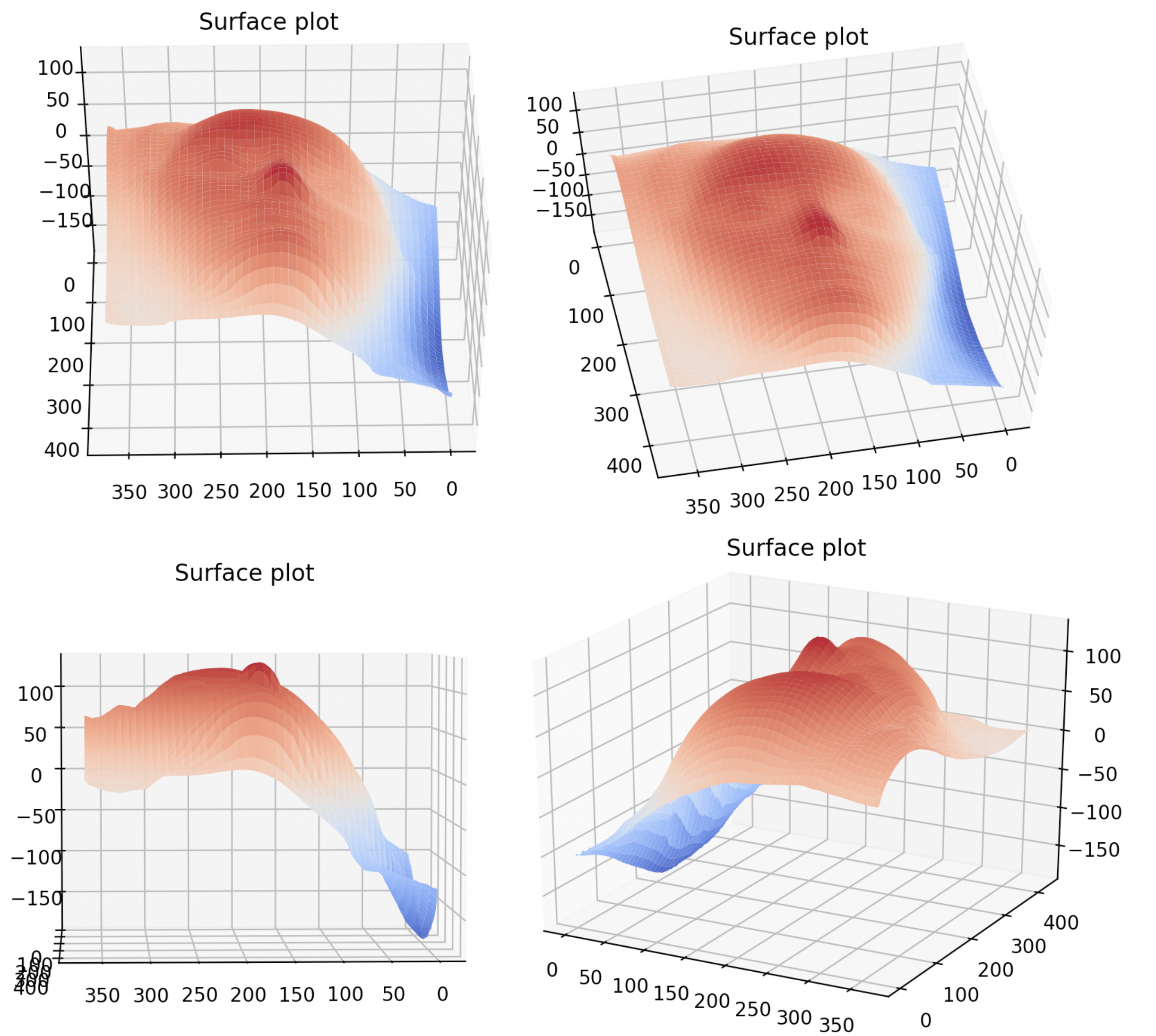




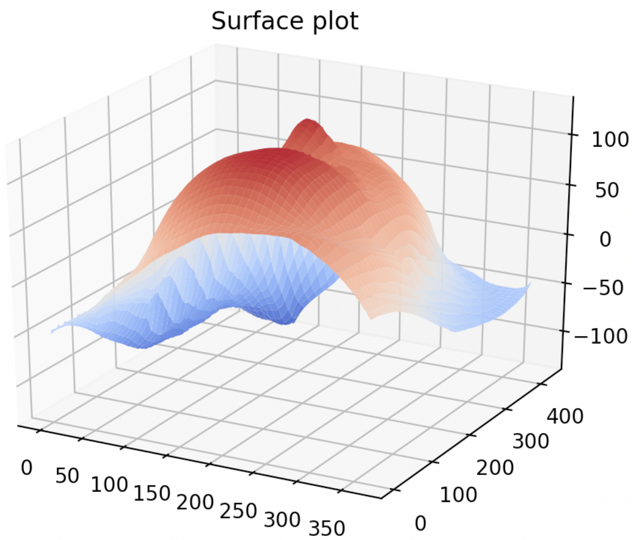
3. When μ=2 keeping λ=1 and ν=0, it can be seen below that the image is stretched on one side and depth has increased on that side.

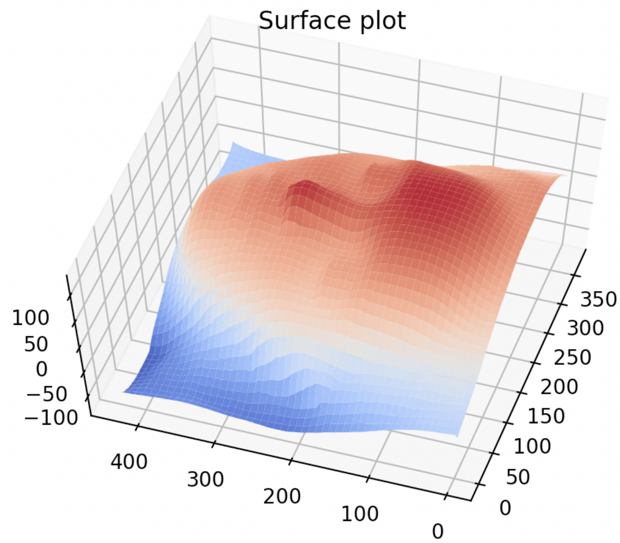


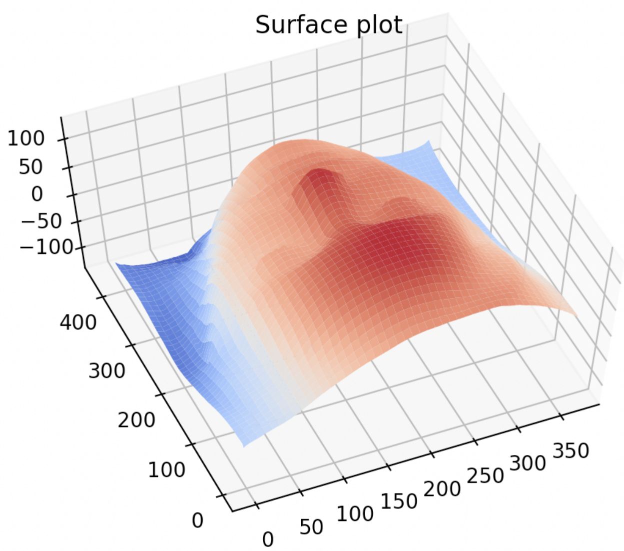
4. When μ=-2 keeping λ=1 and ν=0, it can be seen below that the image is stretched on the opposite side compared to previous way and depth has increased on that side.

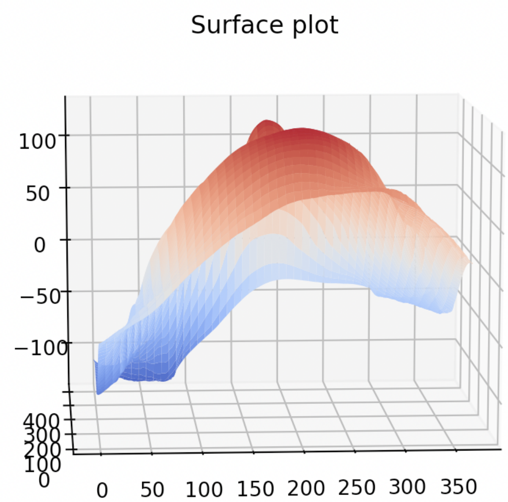


5. When ν=2 keeping λ=1 and μ=0, it can be seen below that the image is stretched on the orthogonal side compared to when just μ was changed to 2.

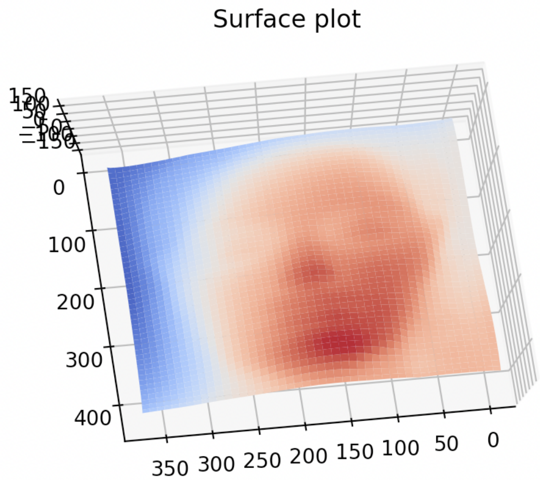


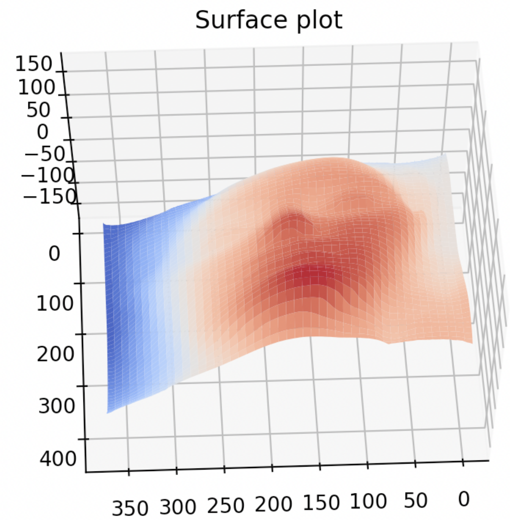


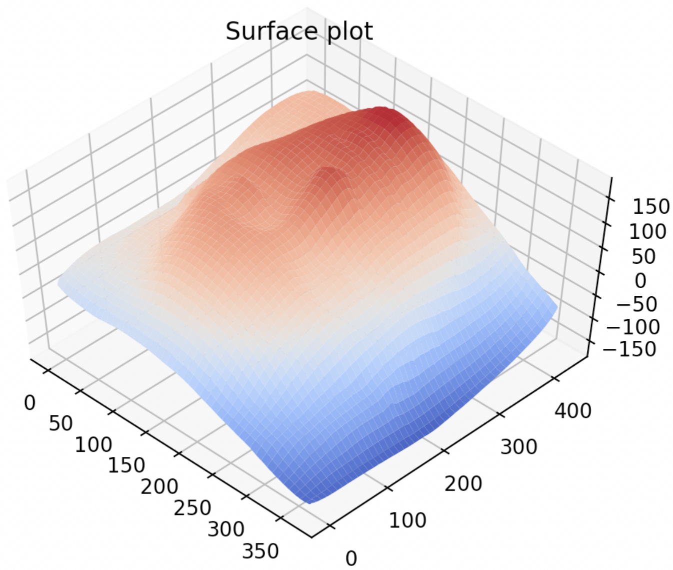


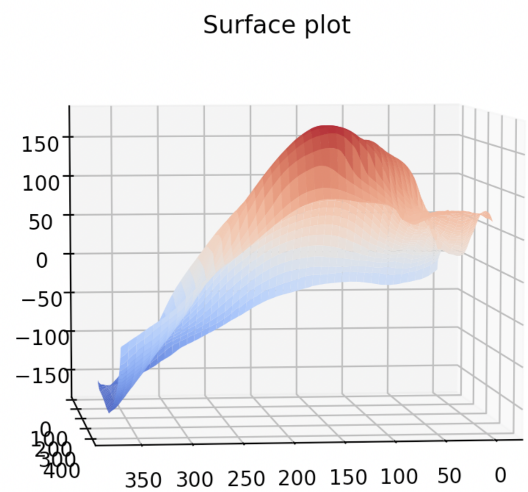


6. When μ=2, ν=-2 and λ=1 , it can be seen below that the image is become comparatively flatter as it is stretched in both x and y plane in opposite directions.

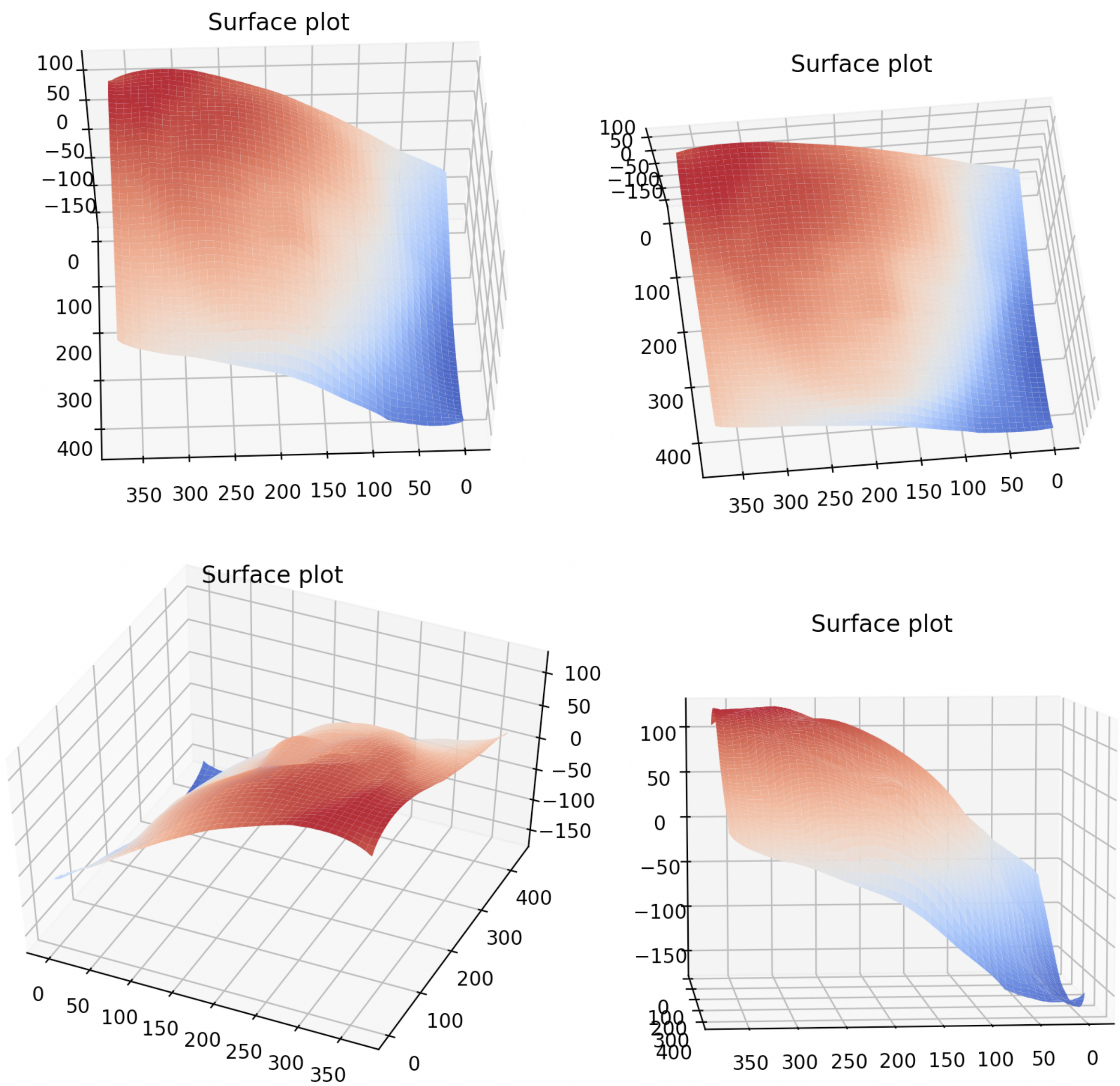








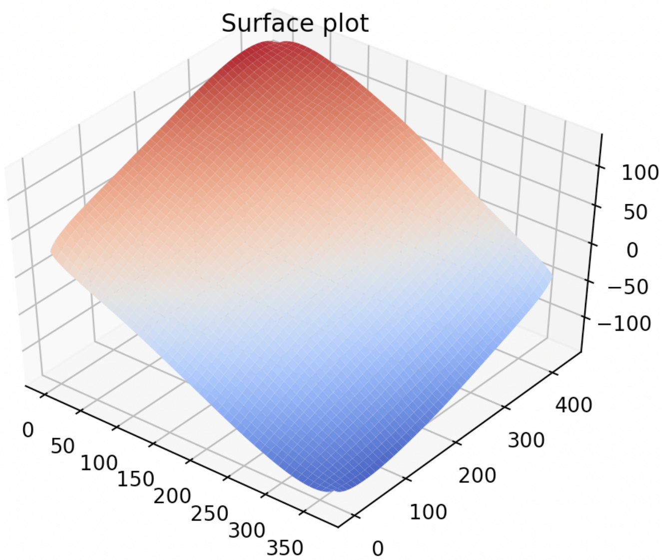
7. When all parameters are changed i.e. μ=-2, ν=2 and λ=5 , it can be seen below that the image has become almost flat.

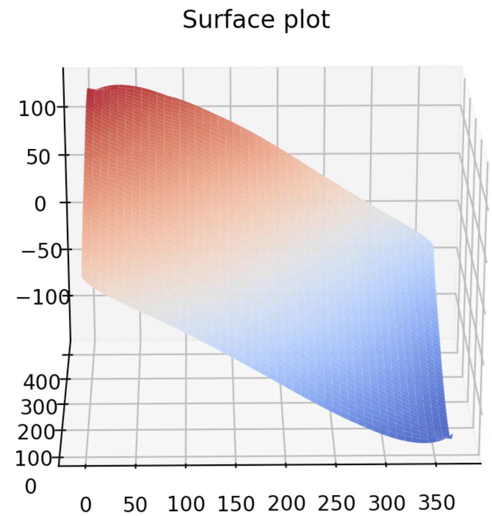


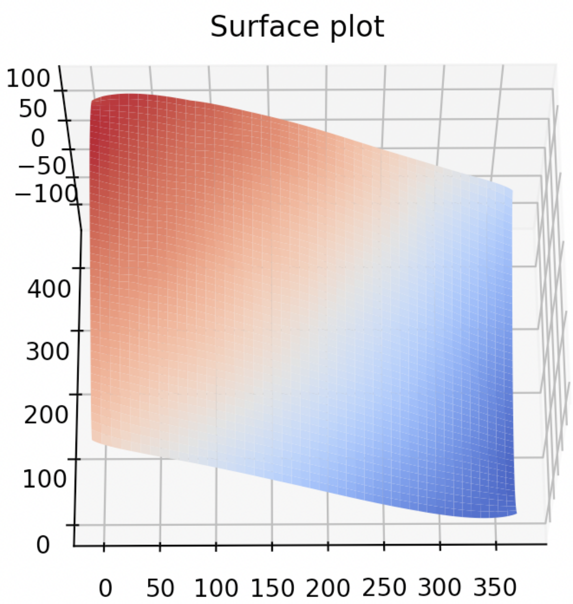
It is named bas relief translating to low relief in the sense that it reduces the ambiguity to only 3 degrees of freedom which here as can be seen above can be varied to stretch, squish, rotate the plane of the image. The bas-relief ambiguity gives you surfaces of varying relief, one of which is the correct, high-relief surface. However, the same appearance as created by the high-relief surface can be created by another, low-relief (less depth) surface and a change in lighting due to the bas-relief ambiguity.

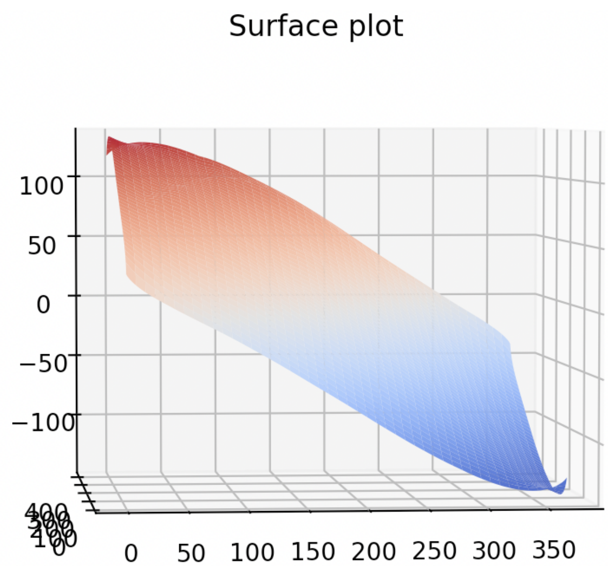
Q2.g Flattest surface possible

Giving μ and ν opposite values like 2 and -2, and combining it with a low value of λ, around 5 results in a flat image as below:









Q2.h More measurements

We solved the problem here with 7 pictures of the face. Acquiring more pictures from more lighting directions will not help resolve the ambiguity, as these images obtained have a rank 3 and there will still exist an ambiguity with three degrees of freedom. The matrix I can still be decomposed into L^T and B with $\sum$ being distributed in a bunch of ways among them.

3 Homework Feedback, Extra Credit

Q3.a Feedback

The homework is very well designed and is a good culmination for the course as it involved maths and physics as well. Personally, I really enjoyed doing the q1(b) which involved visualising the sphere and camera location. Maybe more such questions could be added in future. Also, there was no gold image provided for albedos and normals for both q1 and q2. It would've saved a lot of time if they were provided to use beforehand. I would also like to point out a few mistakes in code:

1. In utils.py, def integrateFrankot(zx, zy, pad = 512), this function has zy : tuple in the docstring when it is actually of type numpy.ndarray

2. In q1.py, def displayAlbedosNormals(albedos, normals, s), this function asks to use the 'coolwarm' colormap for the albedo image and the 'rainbow' colormap for the normals, whereas the handout asks to use 'gray' colormap for albedo image.