CSC2503 A3

Camera Obscura & Photometric Stereo

Wuqi Li
1000292033

Part A

Q1

Scene:

Setup

Image formed

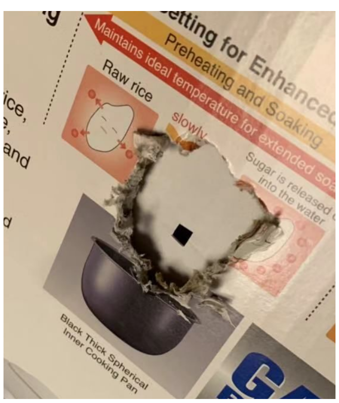
Q2

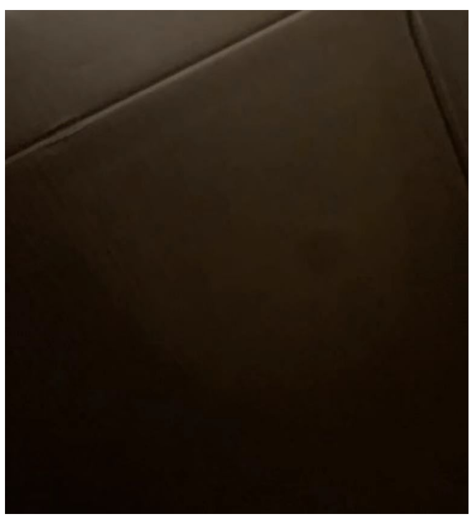
Blurry image

Scene

Pinhole

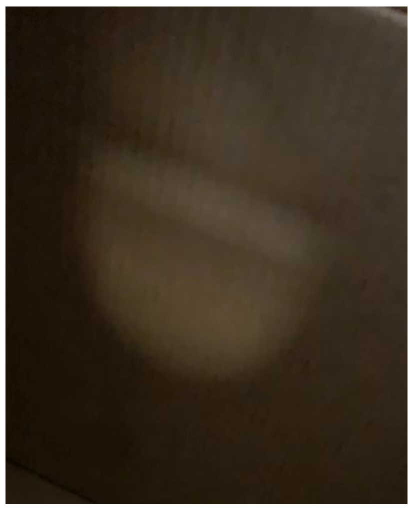
Squared-shaped (small)



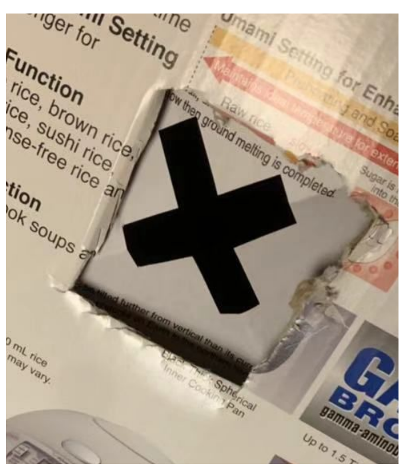


Squared-shaped (large)

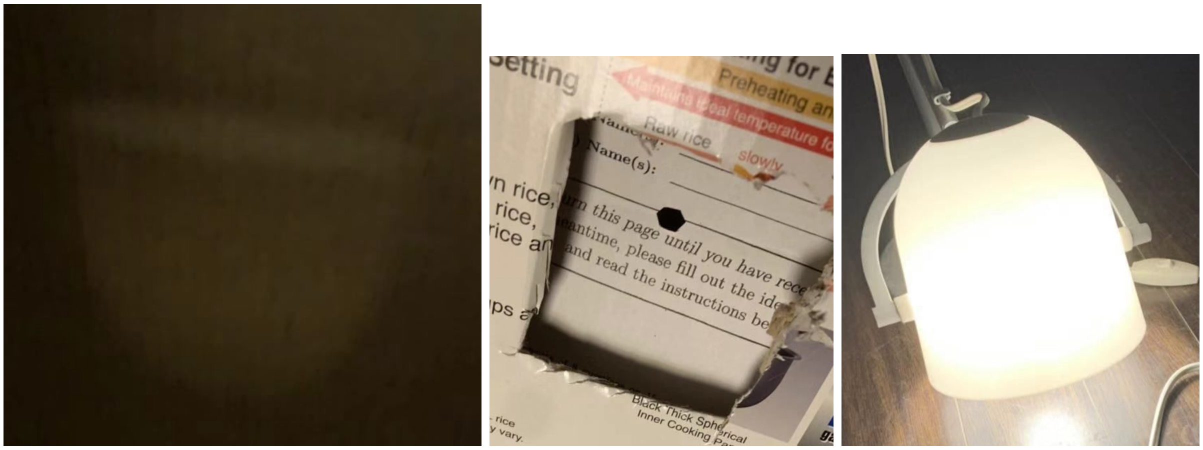
Cross-shaped (small)



Cross-shaped (large)



Hexagonal-shaped (small)



Hexagonal-shaped (large)

Q3

For the ideally extremely small pinhole, the formed image is in back plane should be exactly equal the shape of the scene. For each point in the scene, there should be only point point matches in the back plane. However, for a pinhole with certain shape, every point in the scene, should have a group of points corresponding to.

Therefore, we can come up with the equation $I_S(x, y) = \oint_{S \text{ center at } (x, y)} I(a, b)\, da\, db$, for each point in the image that formed by a shaped pinhole. The $I_s(x,y)$ is the sum of irradiance of all the points that mapping from the shaped pinhole.

The lambertian assumption helps make sure that the brightness of the formed image not relative to the relative location to the light source.
The assumptions about the parallelism make sure that the x-y coordinate of the light source is equivalent to the the x-y coordinate of the backplane. The planarity keeps the shape of the formed image can only be affected by the pinhole shape.

Part B
Q1
Under the assumption that the the camera direction relative to the object is (0,0,-1). The x-y coordinate in the image is equal to the real object. Therefore, doing operations about x-y in the image is equivalent of doing so on the real object.

<u>Algorithm:</u>

Step 1: Find the x-y coordinate of the center of sphere with the mask image.. Also, we can get the radius easily.

Find all points whose mask flag > 0.

Center-x = mean of Xs of those points
Center-y = mean of Ys of those points

Radius = [(max of y) - (min of y) + (max of x) - (min of x)] / 4

(For each image repeat step 2 to step 4)

Step 2: Find the brightest point in the image.

Given the brightness of all the points in a image, we are able to find a group of points which brightness equals to the maximum brightness of all points.

Find the center point's x-y location by averaging all points with maximum brightness.

Step 3: Find the normal vector of the brightest point.

With the x-y coordinates of center point and the brightest point, we are able to find the distance (d) between them. With the radius, we can get the tan(Θ) = d/radius where Θ is the angle between the normal vector of the center and the normal vector of the brightest point.

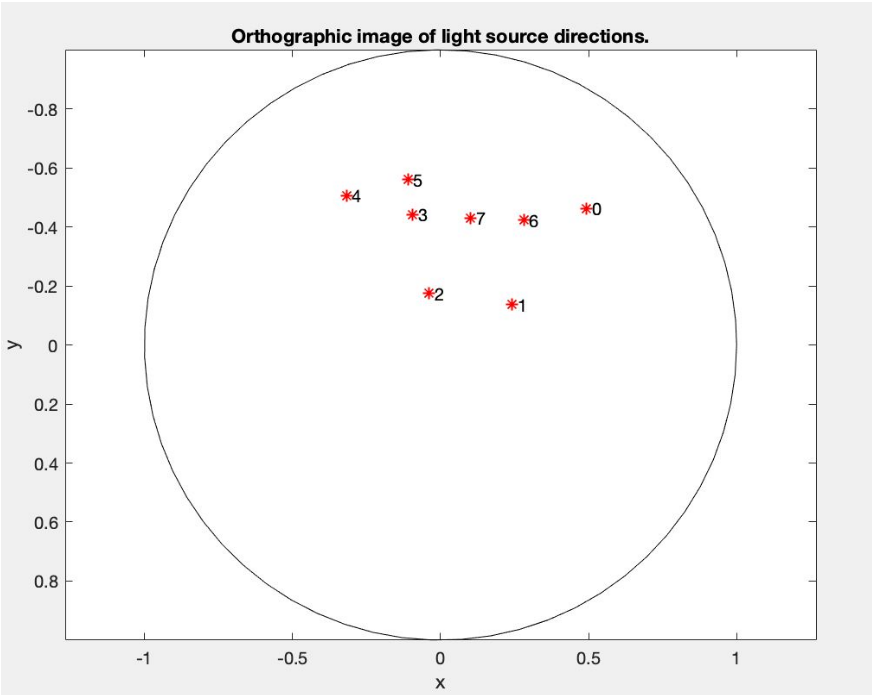
With the Θ, we can get the distance between the brightness point and the center in z-coordinate, which is radius * cos(Θ).

After all, we know the distance between the brightest point and the center in xyz-coordinate, therefore, we know the normal vector.

Step 4: Find light direction by the law of reflection (1)

$$\hat{\mathbf{d}}_{\mathrm{s}} = 2\left(\hat{\mathbf{d}}_{\mathrm{n}} \cdot \hat{\mathbf{d}}_{\mathrm{i}}\right)\hat{\mathbf{d}}_{\mathrm{n}} - \hat{\mathbf{d}}_{\mathrm{i}}$$

With the normal vector $\mathbf{d_n}$ and the camera vector $\mathbf{d_i}$ = [0,0,-1], we are able to find the light direction $\mathbf{d_s}$.



Q2 As shown in the question text, for each image.

$$E = (I - gL)^2$$

$$\frac{\partial E}{\partial g} = 2(I - gL) * -(L)$$

In order to let the derivative to be zero. We have

$$L^T I = g L^T L$$

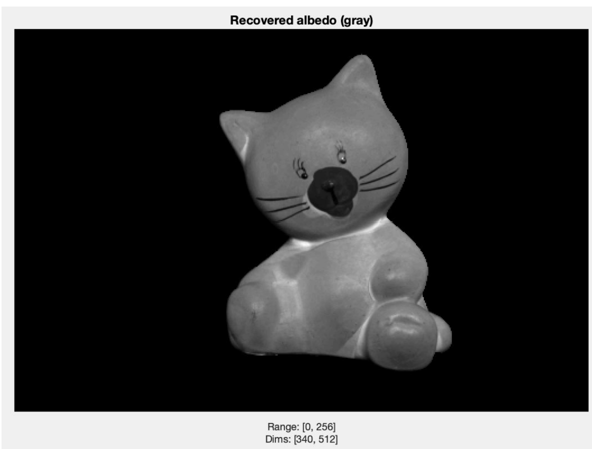
$$g = \frac{L^T I}{L^T L}$$

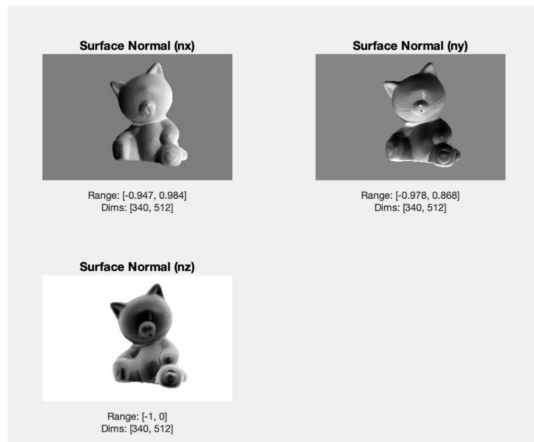
By solving above equation, we are able to get the g for Minimum E. After that, by getting the magnitude and the direction of g, we are able to get albedo and normal vector.

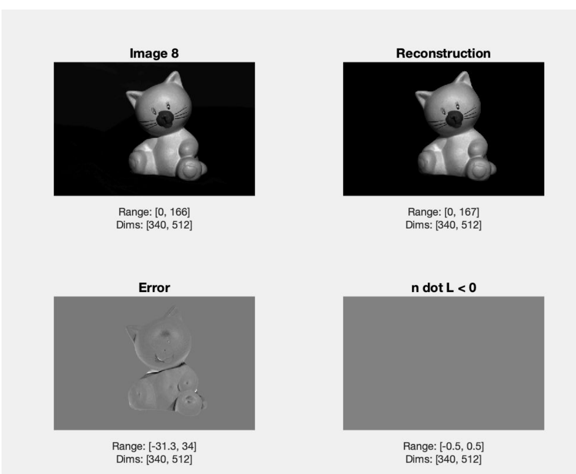
$$a = ||g||$$

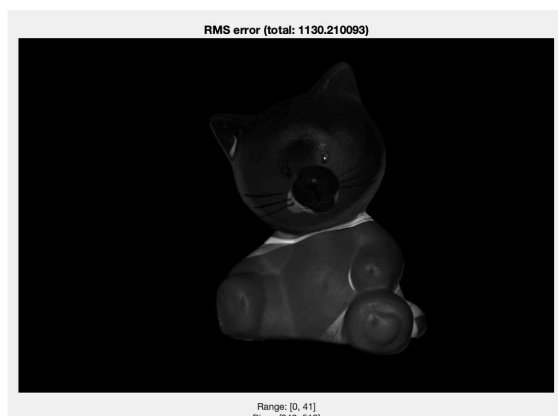
$$n = \frac{g}{||g||}$$

By running the solutemplate script on image 3, the following images for cat are generated. (Appendix A contains results of other objects) As it can be seen, the image is recovered very well. The overall shape of the cat and even many details are shown in the grey level image reconstruction. However, errors can still be found. The main location of the errors are the location of the cat's neck. Those errors are caused by the shadow of the cat. As the light coming to the cat, some parts of the body will block the light and shadow will occurs. We are doing the reconstruction with only the light direction and we did not simulate the shadow. Therefore, the errors will occurs in places where shadows exist in the original picture.









Q3

As shown in the question text. For each color, in each image we have
$E = (I - anL)^2$

$$\frac{\partial E}{\partial a} = 2(I - anL) * (-nL)$$

In order to have the derivative to be zero

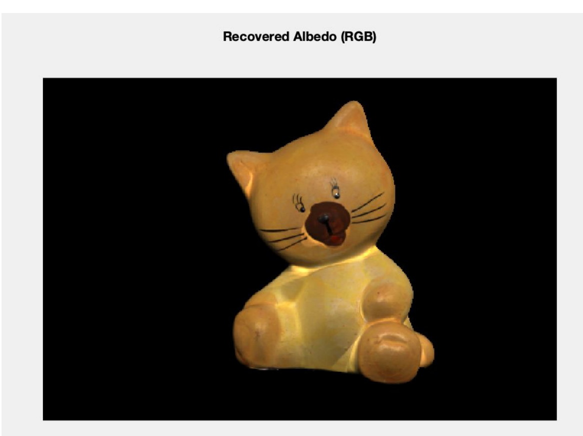
$$(nL)^T I = (nL)^T anL$$

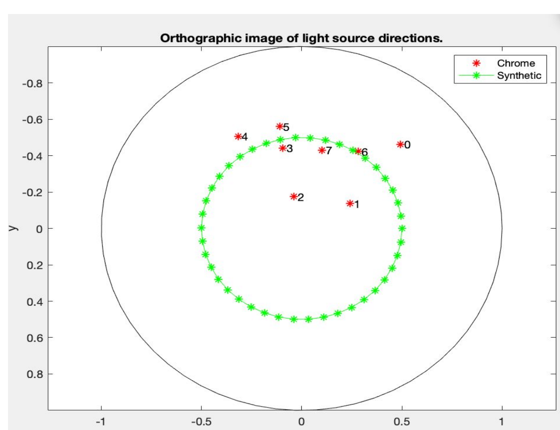
$$a = \frac{(nL)^T I}{(nL)^T nL}$$

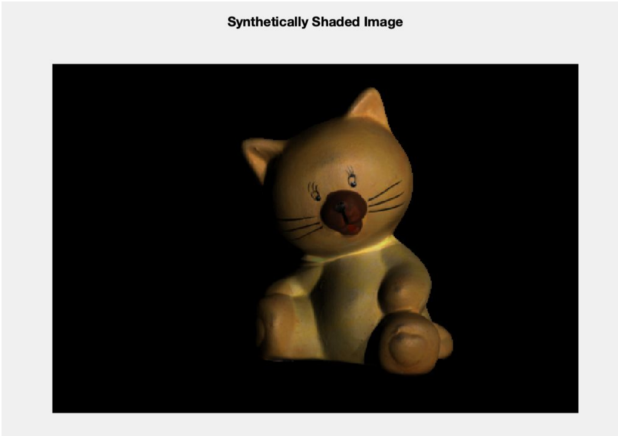
The recovered RGB Albedo and the synthetically shaded image are shown below. As it can be seen, the overall shape is correct in both images, however, the shadow errors are still present in both pictures.

In the recovered Albedo image, the color difference between pixels are recovered very well, however the shadow errors are causing some places too bright.

In the synthetically shaded image, it seems realistic. The shadow and the bright points are all seems real to us. However, some parts of the darkness and brightness seems not follow common logic in our life. For example, due to the impact the single light source, the top part of cat's head are in dark, bottom part of the cat's ear should also in dark. However, in the synthetically shaded image, the ears are quite bright which is not real in our life. In conclude, the synthetically shaded images is well presenting the object, the brightness and darkness are all seems realistic. However, some parts of it are not follow the real world logic, and it may cause the image not make sense.





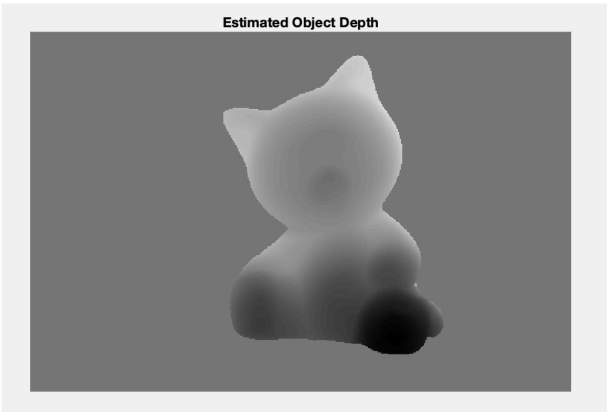


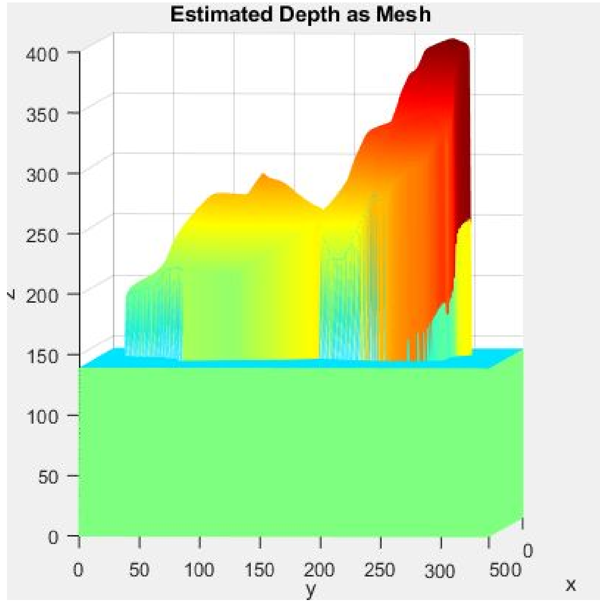
Q4

The main task for this part is constructing matrix A and v.

As shown in the question text, we have $Az = v$. In order to construct A and v, we need to go through each pixel, and construct two rows for each pixel. One row is for $\vec{\tau}_X(x, y) \cdot \vec{n}(x, y) = 0$, and another row is for $\vec{\tau}_Y(x, y) \cdot \vec{n}(x, y) = 0$. Therefore, there are 2 * (number of points) rows in both A and v.

The estimated object depth is shown below. As it shown, it got good optimation. For example the right feet got the darkest color while it should be the thickest part since it is closest part to the camera. Also, for other images which are shown in the appendix C, the object depth are estimated pretty well for most parts of the objects. Based on the estimated depth, we are able to recover the overall structure of the object.





From the Estimated Object depth, we are clearly to see the z-coordinate relationship between different parts in a object. We know which part is closer and which part is further from the camera. However, the actual number of depth seems not realistic. For example, the cat, from the mesh we can see the distance from the right leg to the head is way larger than it should be.

The main cause for this problem might be that the least-square estimator is not good enough for optimizing this issue. In order to getting a small overall error, least-square might lead to wrong direction in some local area.
There are different ways to improve optimization problems. One way we can try is altering estimator, we can try robust estimation for this issue to minimize the effect noise points and points from different local areas.

Q5

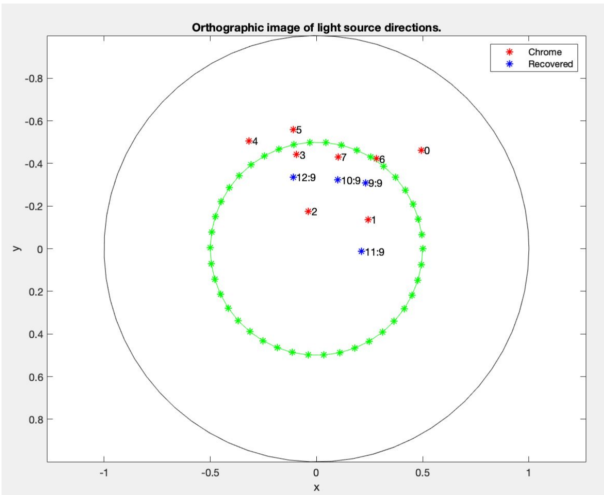
As we know, for each image, we have

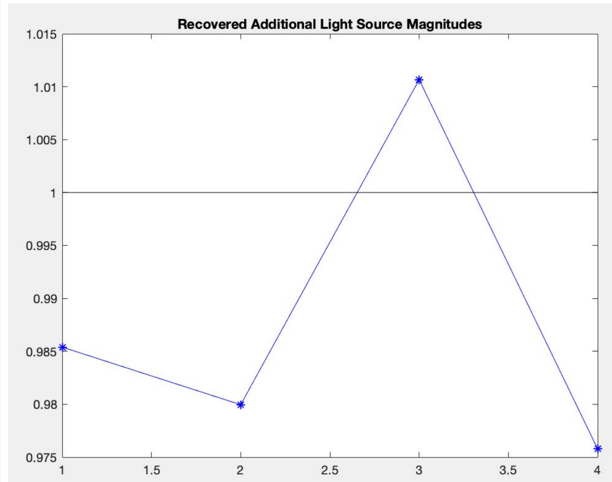
$$E = (I - gL)^2$$

$$\frac{\partial E}{\partial L} = 2(I - gL) * (-g)$$

In order to have the derivative to be zero

$$g * L = I$$



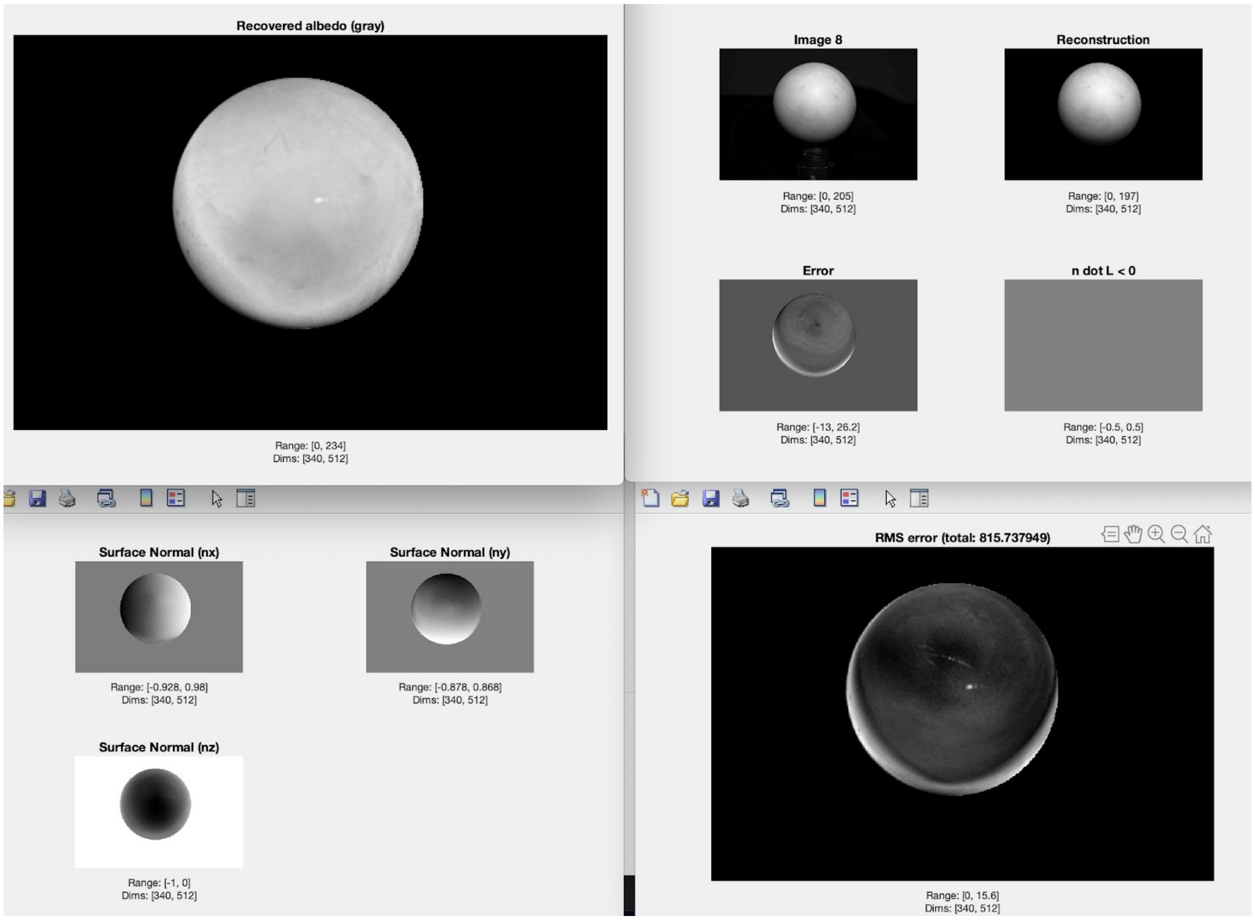


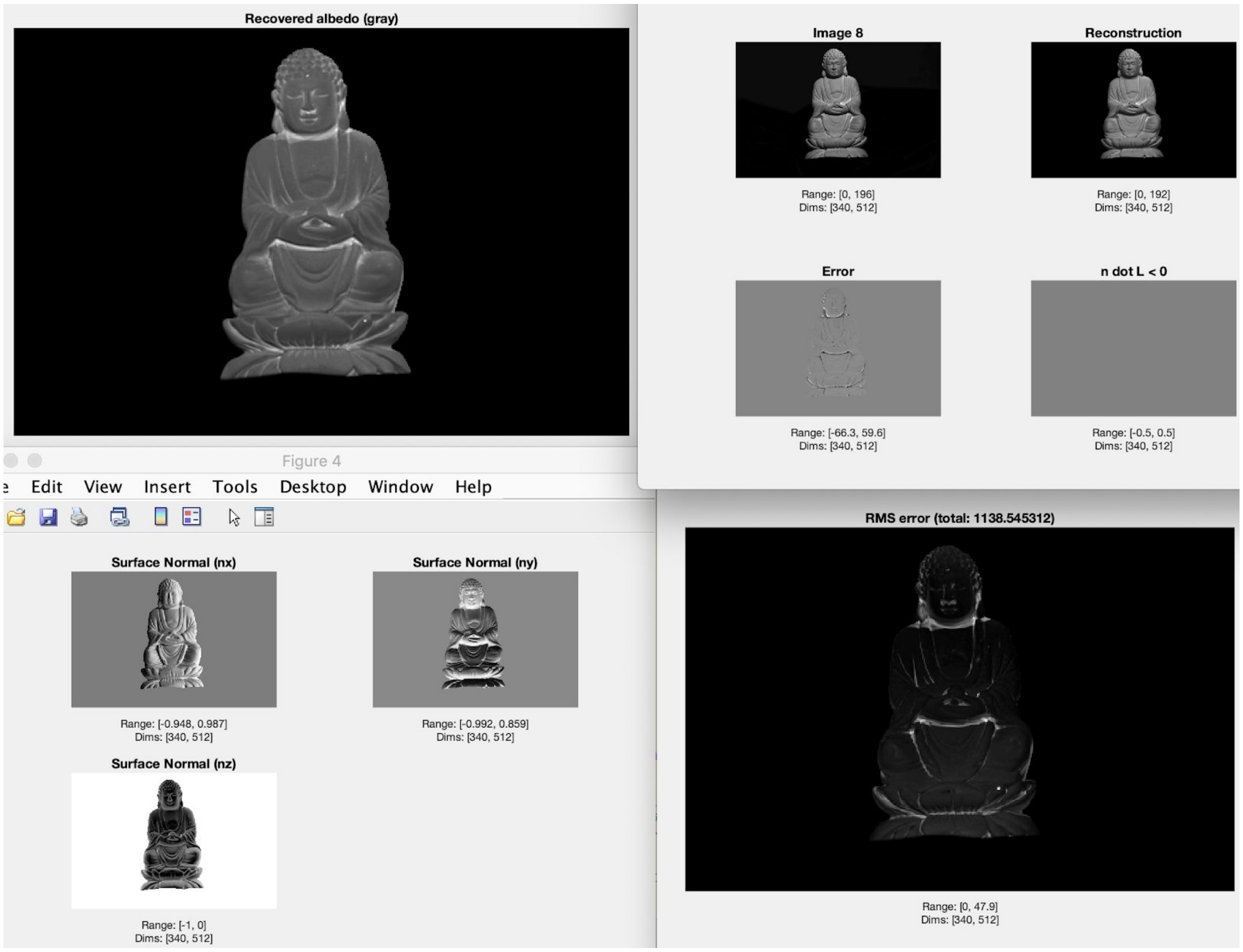
We are not able to find those two light source in all the scenario. As we do not know the brightness and the distance difference between those two lights. One can be extremely bright, and another one can be very low. Therefore, the brightness of the darkest point as a result of the first light can be the brightest point that the second light can bring. Therefore, the little difference that caused by the second light can be very small and unable to catch. Therefore, we are not able to detect two light sources.

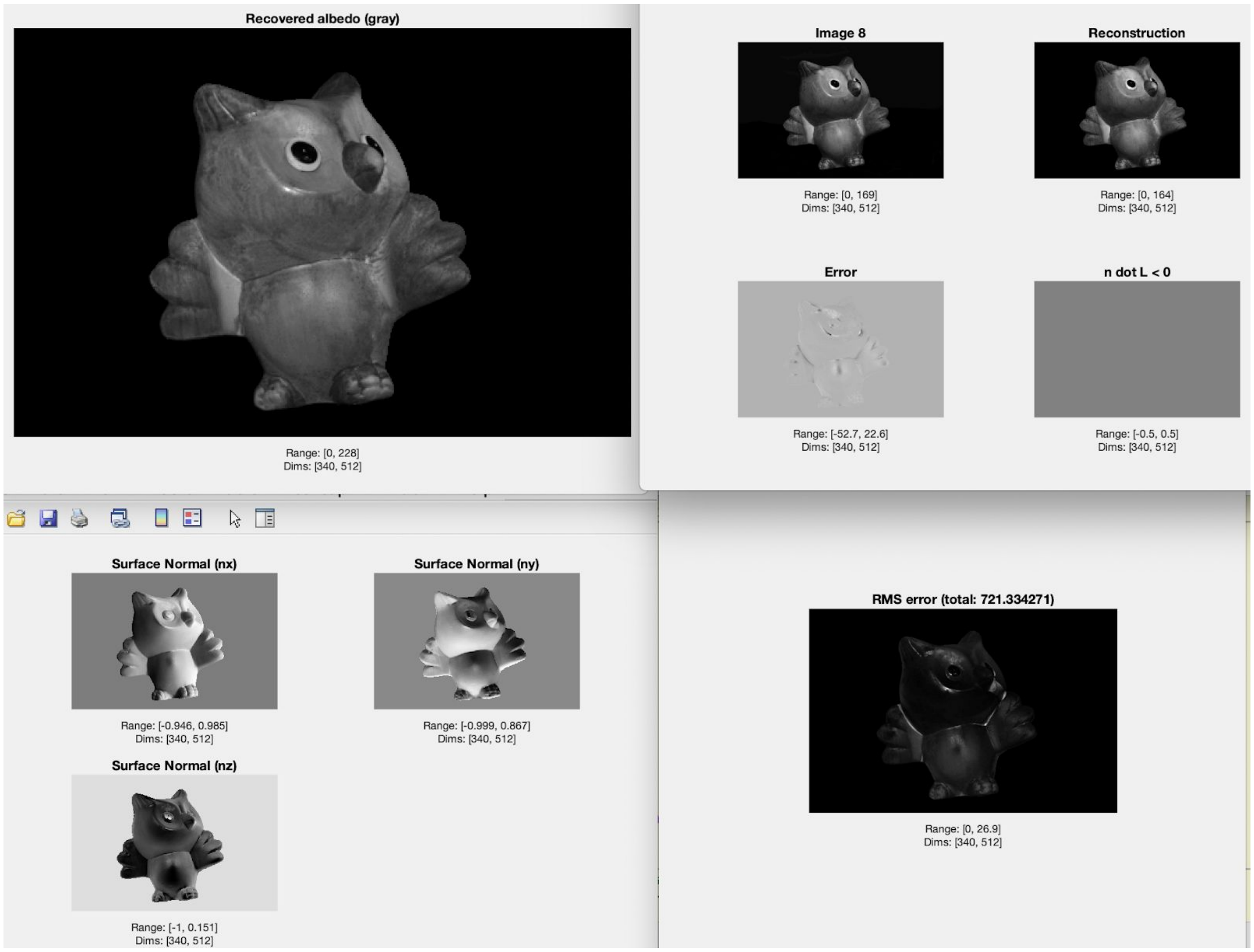
References

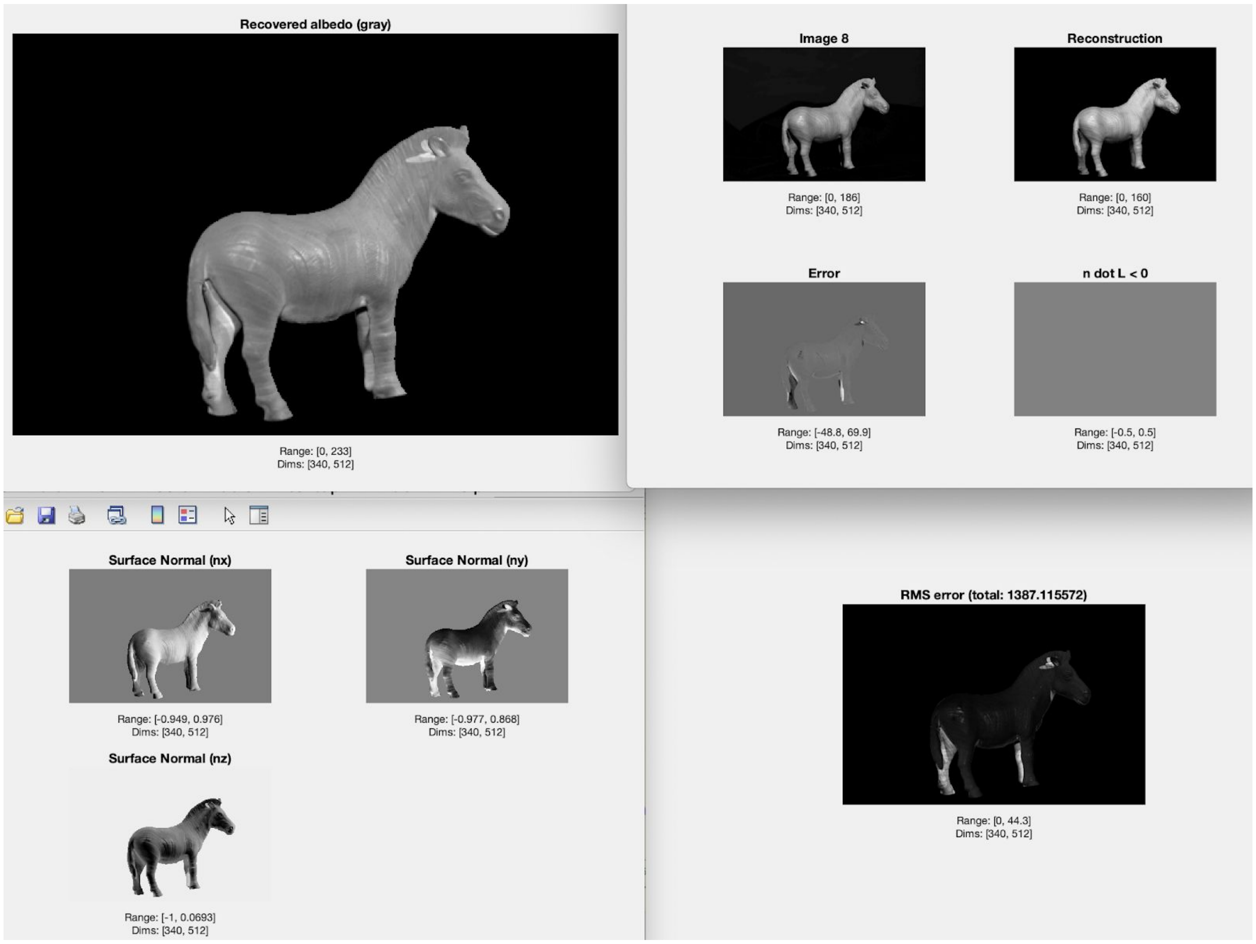
(1) https://en.wikipedia.org/wiki/Specular_reflection

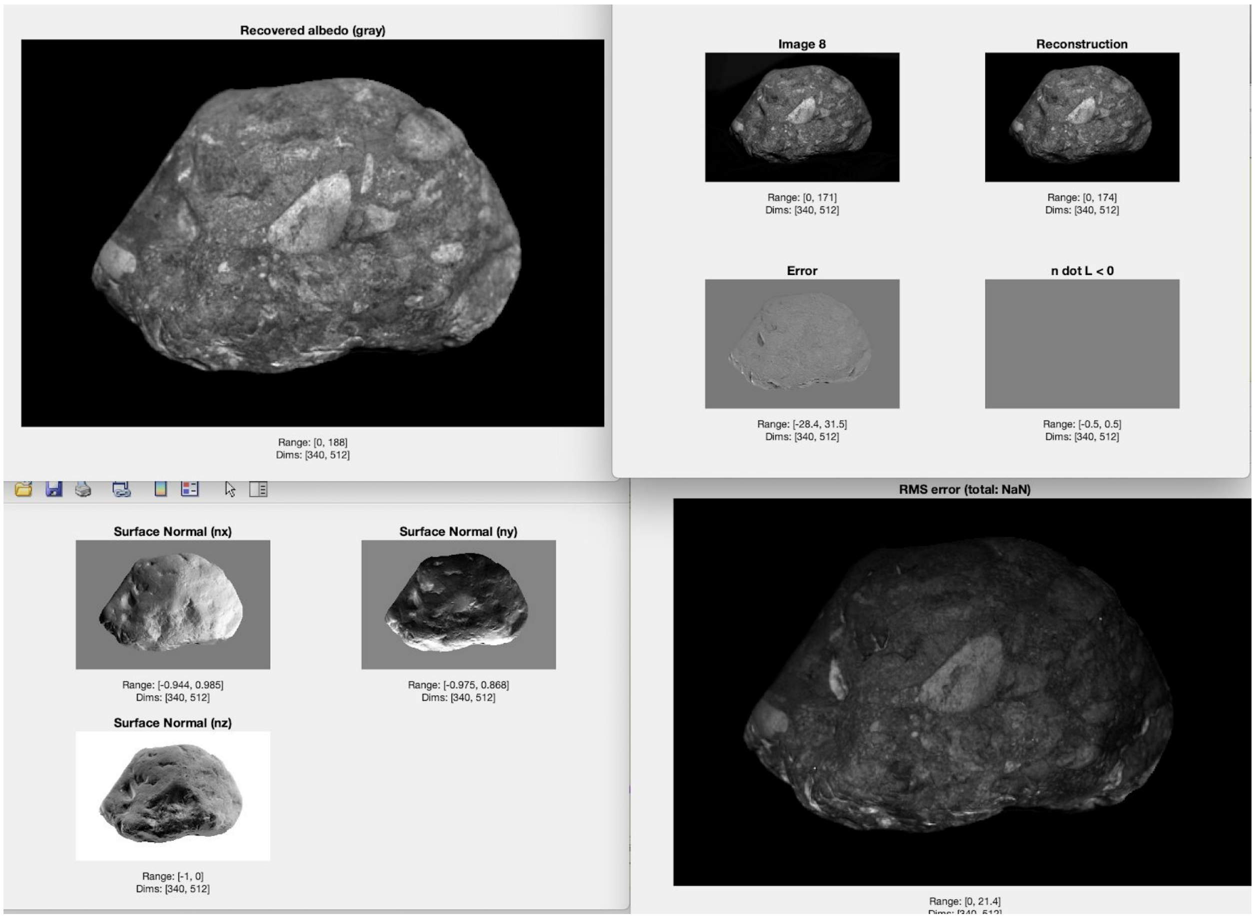
Appendix A Grey Albedo reconstruction image


Recovered albedo (gray)
Range: [0, 234]
Dims: [340, 512]
Image 8
Range: [0, 205]
Dims: [340, 512]
Reconstruction
Range: [0, 197]
Dims: [340, 512]
Error
Range: [-13, 26.2]
Dims: [340, 512]
n dot L < 0
Range: [-0.5, 0.5]
Dims: [340, 512]
Surface Normal (nx)
Range: [-0.928, 0.98]
Dims: [340, 512]
Surface Normal (ny)
Range: [-0.878, 0.868]
Dims: [340, 512]
Surface Normal (nz)
Range: [-1, 0]
Dims: [340, 512]
RMS error (total: 815.737949)
Range: [0, 15.6]
Dims: [340, 512]

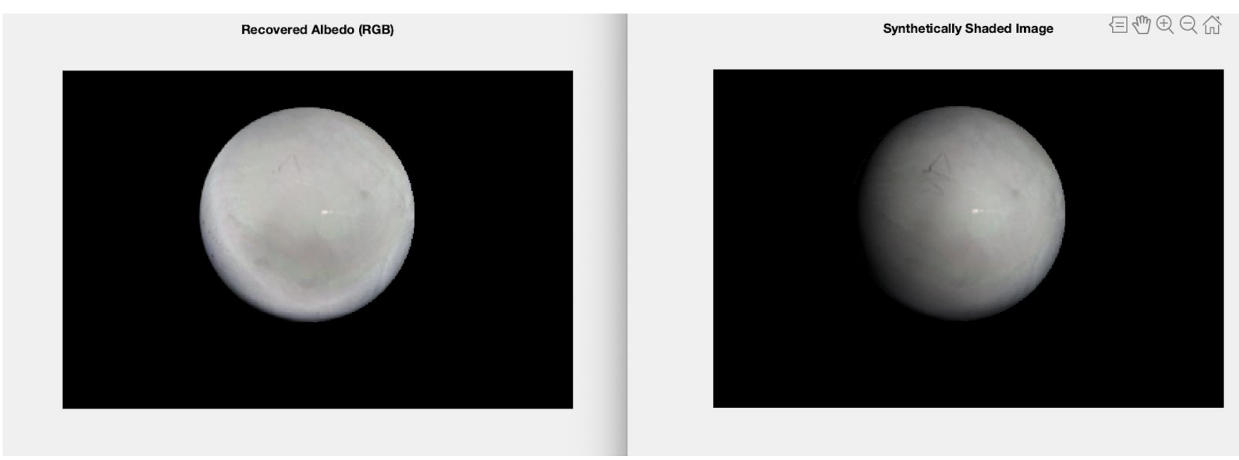

Recovered albedo (gray)
Image 8
Range: [0, 196]
Dims: [340, 512]
Reconstruction
Range: [0, 192]
Dims: [340, 512]
Error
Range: [-66.3, 59.6]
Dims: [340, 512]
n dot L < 0
Range: [-0.5, 0.5]
Dims: [340, 512]
Figure 4
e Edit View Insert Tools Desktop Window Help
Surface Normal (nx)
Range: [-0.948, 0.987]
Dims: [340, 512]
Surface Normal (ny)
Range: [-0.992, 0.859]
Dims: [340, 512]
Surface Normal (nz)
Range: [-1, 0]
Dims: [340, 512]
RMS error (total: 1138.545312)
Range: [0, 47.9]
Dims: [340, 512]


Recovered albedo (gray)
Range: [0, 228]
Dims: [340, 512]
Image 8
Range: [0, 169]
Dims: [340, 512]
Reconstruction
Range: [0, 164]
Dims: [340, 512]
Error
Range: [-52.7, 22.6]
Dims: [340, 512]
n dot L < 0
Range: [-0.5, 0.5]
Dims: [340, 512]
Surface Normal (nx)
Range: [-0.946, 0.985]
Dims: [340, 512]
Surface Normal (ny)
Range: [-0.999, 0.867]
Dims: [340, 512]
Surface Normal (nz)
Range: [-1, 0.151]
Dims: [340, 512]
RMS error (total: 721.334271)
Range: [0, 26.9]
Dims: [340, 512]

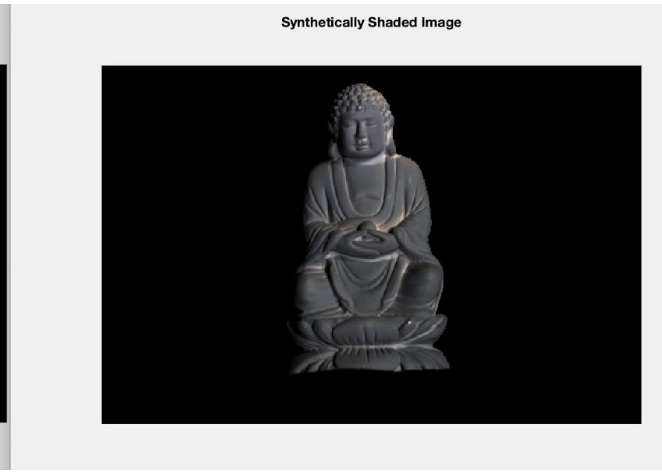

Recovered albedo (gray)
Range: [0, 233]
Dims: [340, 512]
Image 8
Range: [0, 186]
Dims: [340, 512]
Reconstruction
Range: [0, 160]
Dims: [340, 512]
Error
Range: [-48.8, 69.9]
Dims: [340, 512]
n dot L < 0
Range: [-0.5, 0.5]
Dims: [340, 512]
Surface Normal (nx)
Range: [-0.949, 0.976]
Dims: [340, 512]
Surface Normal (ny)
Range: [-0.977, 0.868]
Dims: [340, 512]
Surface Normal (nz)
Range: [-1, 0.0693]
Dims: [340, 512]
RMS error (total: 1387.115572)
Range: [0, 44.3]
Dims: [340, 512]

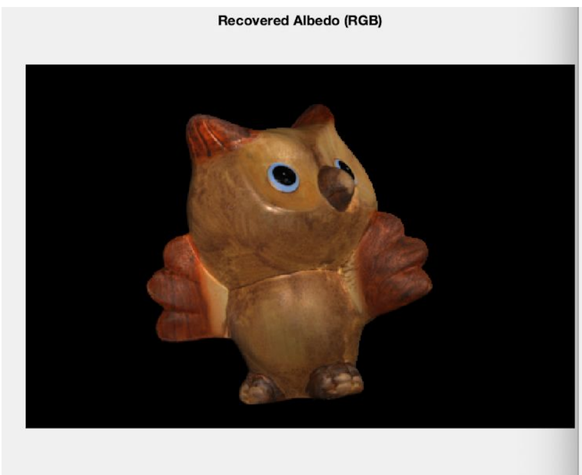


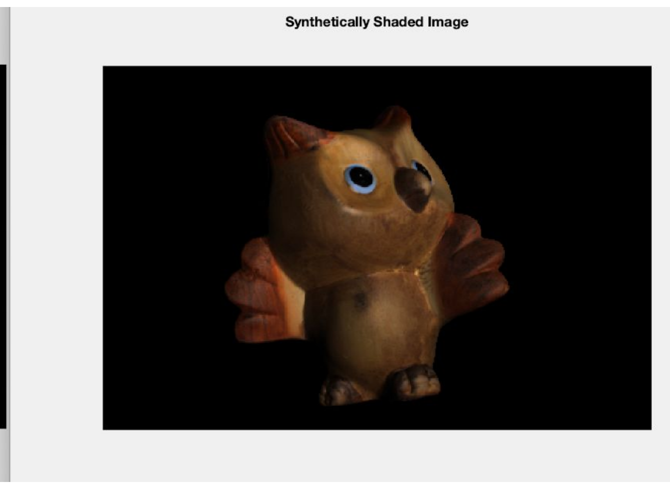
Appendix B RGB Albedo reconstruction image and synthetically shaded image

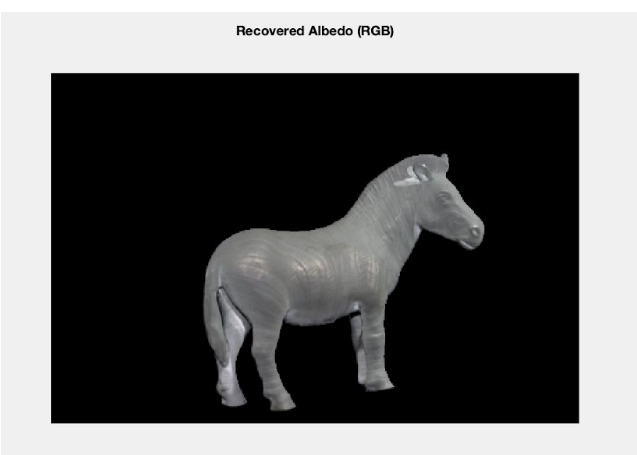


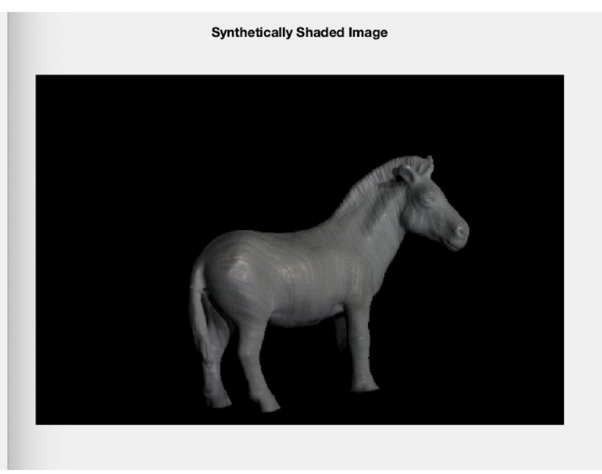

Recovered Albedo (RGB)

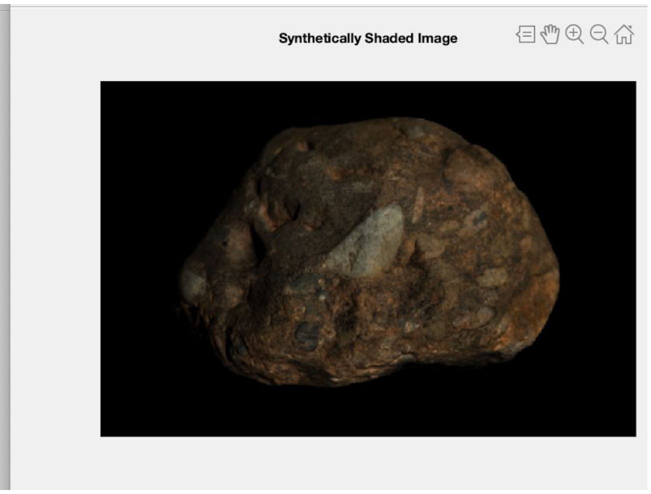

Synthetically Shaded Image


Recovered Albedo (RGB)

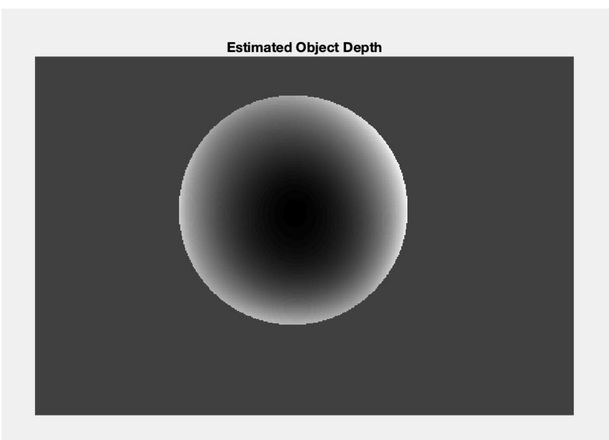

Synthetically Shaded Image

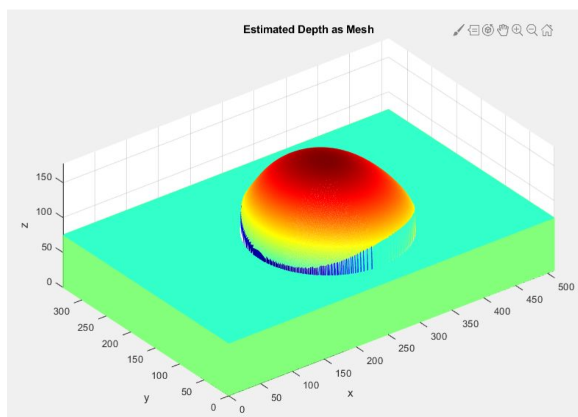

Recovered Albedo (RGB)

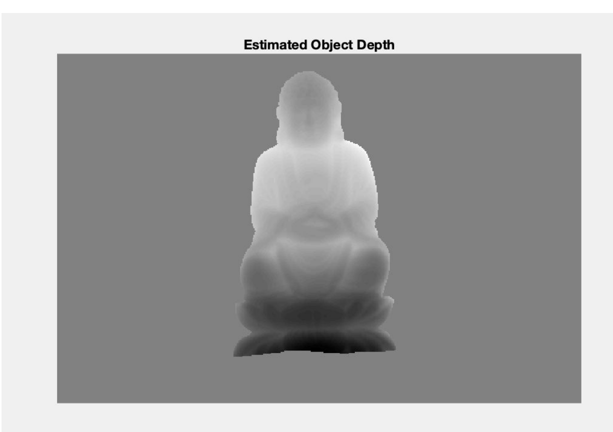

Synthetically Shaded Image

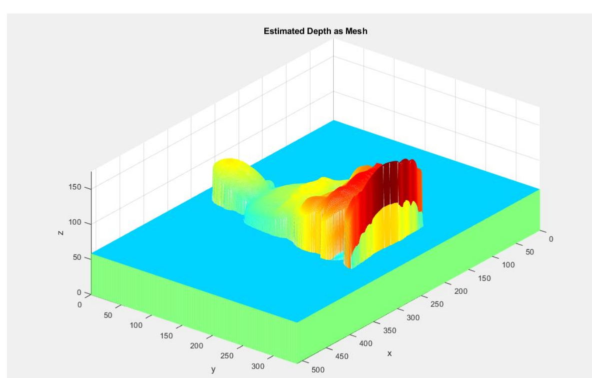


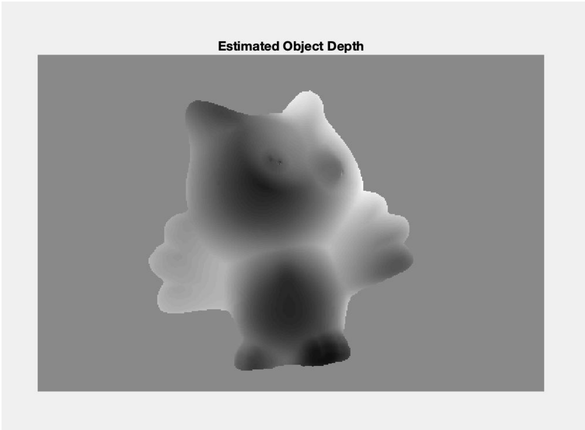
Appendix C Estimated Object Depth

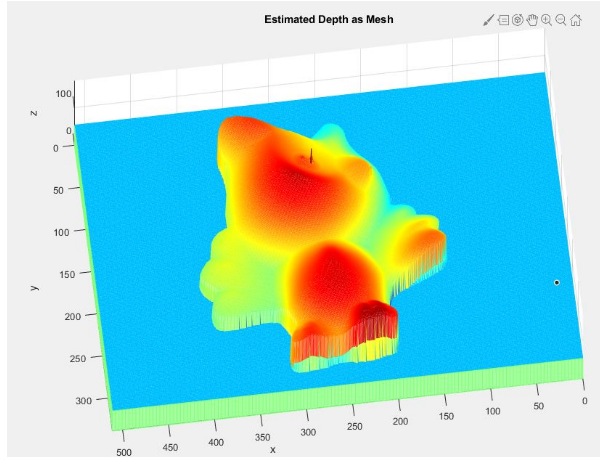


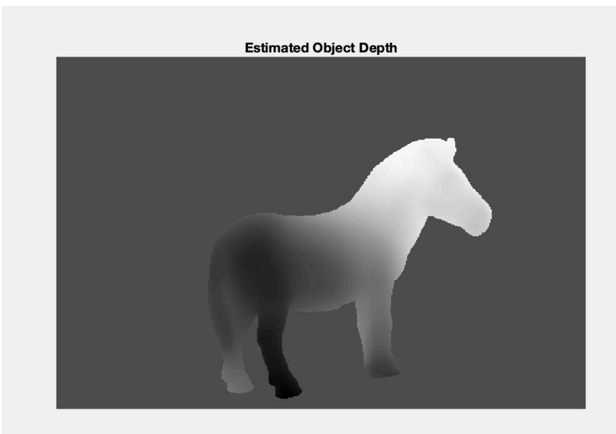


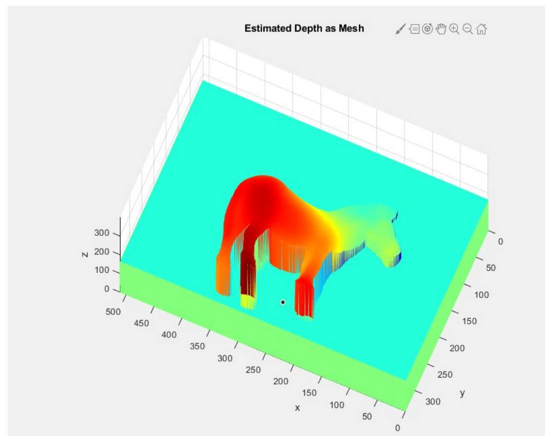


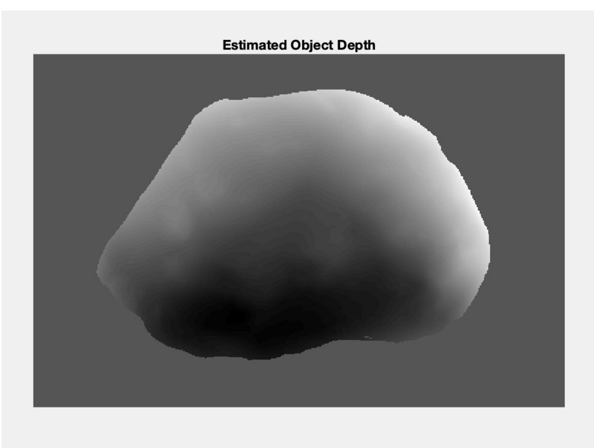


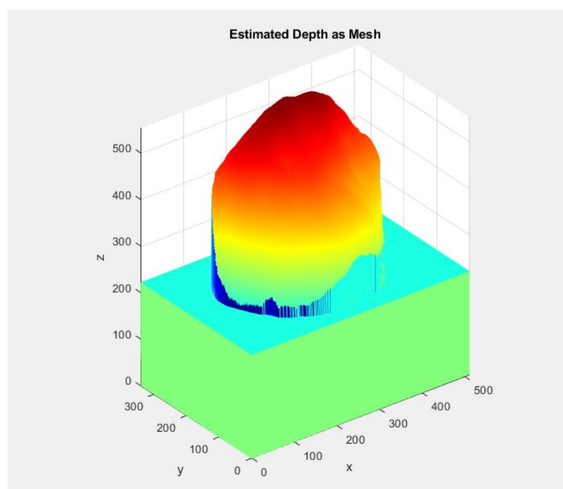

Estimated Object Depth


Estimated Depth as Mesh
x
y
z


Estimated Object Depth


Estimated Depth as Mesh
x
y
z


Estimated Object Depth


Estimated Depth as Mesh
x
y
z