VIP :: Assignment 3: Photometric Stereo

Olga Iarygina (hwk263), Ioannis Manasidis (jdv382), Marina Pinzhakova (ptr273)

In this assignment, we used the Photometric Stereo method to obtain 3D shapes of objects.

Part 1. Theory

Question 1: *Write for a generic pixel, light source and normal, Lambert's law in its simplest form.*

Lambert's law in its linearised form:

$$I(p) = \rho(x)\, s(x) \cdot n(x)$$

where:

- *I(p)* is the light intensity *I* (amount of light emitted) at pixel *p*, usually ranging 0-255 or 0-1, with the right bound indicating the brightest pixel
- *ρ(x)* - the albedo at x
- *s(x) · n(x)* is the dot product between the light direction vector and the vector normal to the surface at x.

For a generic pixel *(u,v)* we have the following system of equations:

$$\begin{cases} I^1(u,v) = \rho(u,v)s^1 \cdot n(u,v) \\ I^2(u,v) = \rho(u,v)s^2 \cdot n(u,v) \\ \dots \\ I^k(u,v) = \rho(u,v)s^k \cdot n(u,v) \end{cases}$$

where:

- I^k *(u,v)* is the light intensity at pixel (u, v) of the *k*-th image
- *ρ(u,v)* is the albedo extracted from the images at pixel (u,v)
- s^k is the light source matrix for the *k*-th image
- *n (u,v)* is the surface normal at pixel (u, v)

Question 2: *How is Lambert's law modified to deal with self shadows?*

Self-shadow area of an object is the back of the surface (light is blocked by the object). The self-shadowed area is described by the angle between the normal vector to the surface and a light direction (angle of incidence) that is larger than 90 degrees.

In such cases, the normal to the shadowed area would be antialigned to the direction of the light source. When applying Lambert's law for an occluded pixel (u,v), the light intensity

$I(u,v)$ becomes negative, which doesn't make sense. Consequently, we only consider the angle of incidence from 0 to 90 degrees and take zero instead of a negative value $s \cdot n$.

Question 3: *What about cast shadows? Comment on the difference between the two.*

When a part of an object is occluded by another object or the object itself, it is considered a cast shadow. Lambert's law can not account for this occasion. Thus we take the following equation for the light intensity at the occluded point:

$$I(u, v) = \rho\ max(s \cdot n(u, v),\ 0)$$

To distinguish between the two kinds of shadows:
A cast shadow is the one that occurs when the light is blocked by something. A self shadow on the other hand is the one found on the flip side of a surface that receives light.

Question 4: *Comment on the modelling limits of Lambert's law.*

Handling shadows is one of the main limitations of Lambert's law. Self and cast shadows are essentially ignored. Cast shadows are not taken into account at all, as the model is centered only on objects that do not occlude each other, while self shadows require a hack to get the light intensities to be larger or equal to 0 *(question #2)*. In addition, the Lambertian model makes an assumption that the surface of the object is diffuse/matte and cannot describe specular/mirror-like reflections. It starts to fall short when the object in the scene happens to be glossy and possesses, albeit minimally, some light-reflective properties.

Question 5: *How can we obtain an estimate of albedo and normals in Woodham's approach to Photometric Stereo? Write the equation.*

The albedo is the measure of the amount of light that gets reflected from an object in the scene without being absorbed. A high albedo value for a pixel (u, v) indicates that throughout all the *k* images of a scene that were taken, the material at that pixel reflected most of the light back, and vice versa, a low value means that the material absorbed most of the light (and appears therefore darker).

Woodham suggested taking the product of the albedo and the normal, instead of considering them separately. This new unknown is **m** in the following equation:

$$m = \rho n, \rho = ||m|| = \sqrt{\sum_{i=1}^{k} m_i}\ ,\ n\ = \frac{m}{\rho} = \frac{m}{||m||}$$

where:
- *m* is the albedo-modulated normal, in fact - the product of albedo and the normal
- *k* is the number of total light sources (images)

- ρ is the albedo, the modulated norm over different light sources $m_1, m_2, \ldots, m_k$
- n is the normalized vector, which is the estimation of the surface normal

Using this approach, we can update Lambert's law for a generic pixel:

$$\begin{bmatrix} I_1 \\ I_2 \\ \cdot \\ \cdot \\ I_k \end{bmatrix} = \begin{bmatrix} s_1^1 & s_1^2 & s_1^3 \\ s_2^1 & s_2^2 & s_2^3 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ s_k^1 & s_k^2 & s_k^3 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}$$

where: I is the vector of light intensities, S is the matrix of light directions, and m is the albedo-modulated normal, which is computed by using the Moore-Penrose pseudoinverse of S: $m = S^{-1}I$

Question 6: *What should be done if one uses RANSAC. Please describe. It will help you when implementing the RANSAC based estimation.*

Our data consists of observations, which include k light source vectors **S** and k intensity values **I**. The intensity value needs to be per pixel, not per image. According to Lambert's law, each intensity value observed in the image is explained by the model **m(p)** given the light source:

$$I_i(p) \approx s_i^\top \mathbf{m}(p)$$

RANSAC gives an estimate of the value **m** (albedo-modulated normal) by randomly sampling the intensity values **I(p)** and light vectors **S**. The minimum number of intensities and light sources is 3.

Part 2. Photometric Stereo

For all datasets we have several pictures of an object made with a fixed camera, fixed object and different lighting. With one lighting angle, some details are clearly visible, and when the angle changes, they may become indistinguishable, while the others clarify. So, we stack all the pictures to get the details of all parts of the object. For each of the datasets we create an array J with shape 3 * number of pixels in the non-zero part of the mask. Having that, we compute the albedo with the following formula: M = S^1 * J. Then we normalize it and extract the normal field, with its 3 components, and get a 3D image.

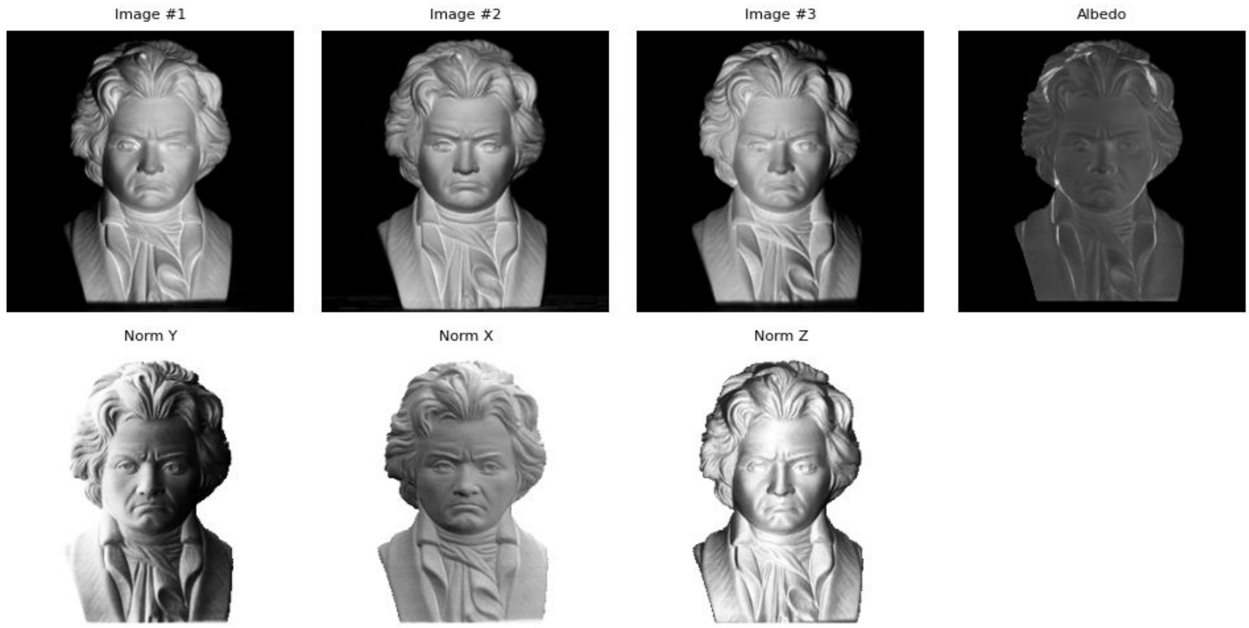
When the surface is diffuse, light falls and reflects not at the same angle at which it fell, but practically in any direction. For shiny elements we have to use smoothing in order to get rid of peaks on 3D visualisations. When the data becomes more complex and impure, but at the same time we have a lot of inliers in the data, we can perform random sampling to get a coherent image. This method is called RANSAC. So, we did random sampling and then repeated all the previous procedures for the data we got.

It should be noted that we ran into a mayavi bug (version 4.7.2), where the existence of a NaN value in the depth map would cause the mlab.mesh() method to not display anything (the scalars= parameter did not work as specified in the documentation on Linux and Mac systems, but worked on Windows). We "fixed" this by assigning the max/min value of the depth map (or 0, depending on the dataset, so that it looks better) to any NaN value present in the map. This creates a black plane in the resulting 3D surface, but at least it makes mayavi able to plot it on our systems.

2.1 Beethoven

Starting with the first dataset, the three provided pictures of Beethoven are quasi-monochromatic and matte, which eliminates any specular reflections. It is worth noting that the 3D surface reconstruction appears to be slightly "slanted" to the right. This means that we cannot obtain all the needed information from the light intensities to fully reconstruct the 3D surface. This can be fixed by adding more pictures from different light sources into the dataset, which will make the resulting surface more round and natural.

Images, albedo and normals plot:

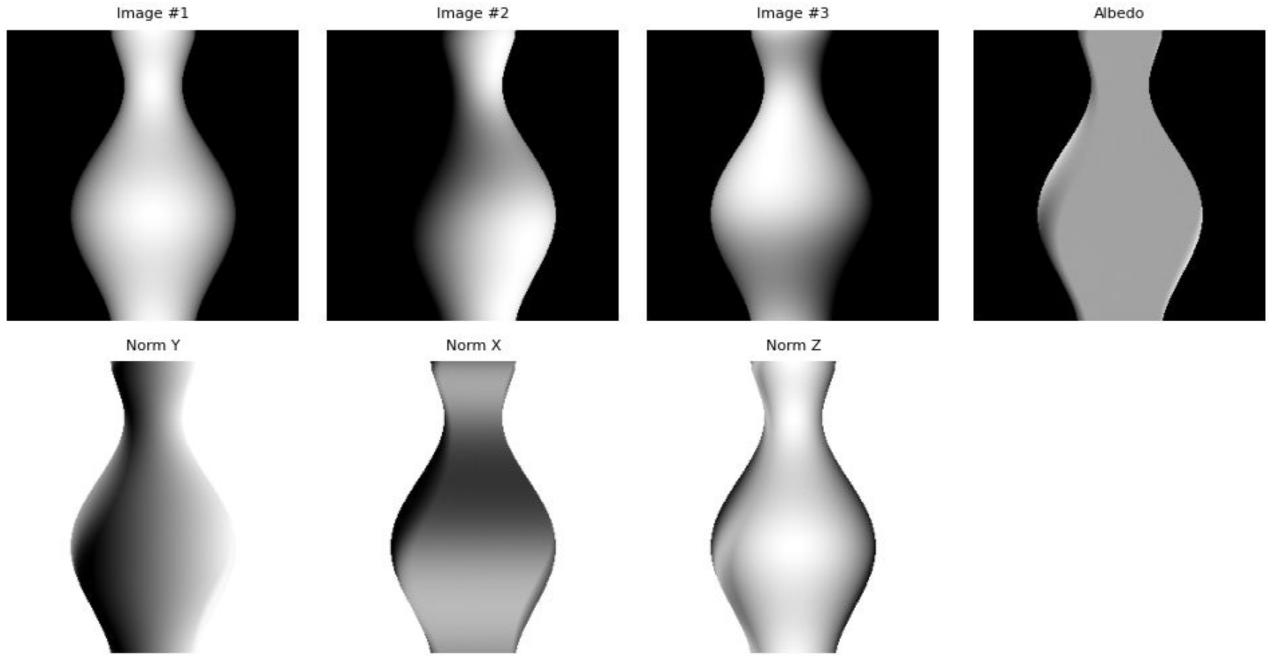


3D surface plot from 3 viewpoints:

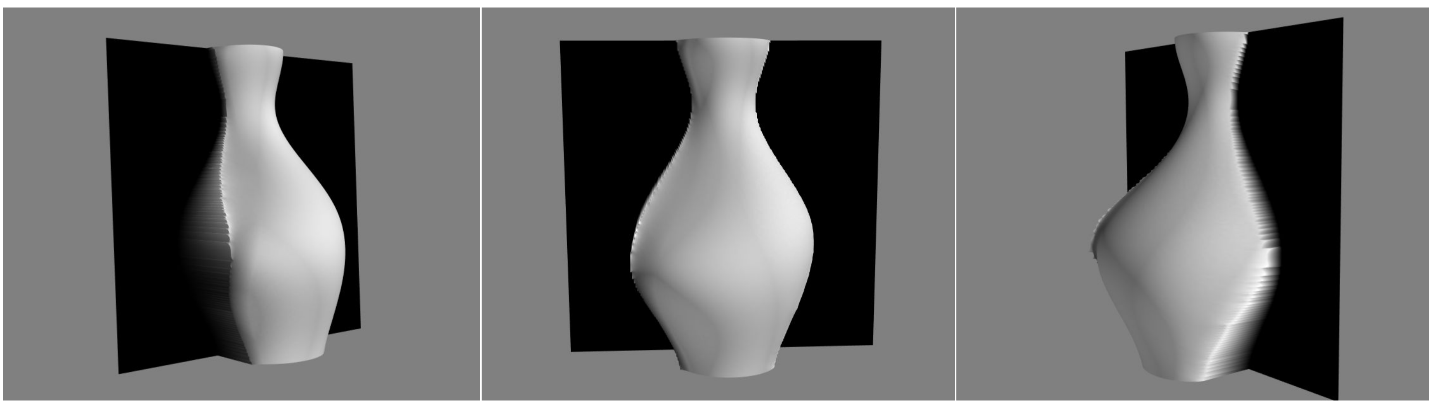
2.2 mat_vase

The second dataset is very similar to the first one. We use the inverse method to calculate the albedo and the normals. This dataset has the same problem as the previous one. Because there are only three images available, the vase's surface is not as plump and smooth as it should be. More shots with different light sources must be taken in order to reconstruct the object in 3D better.

Images, albedo and normals plot:

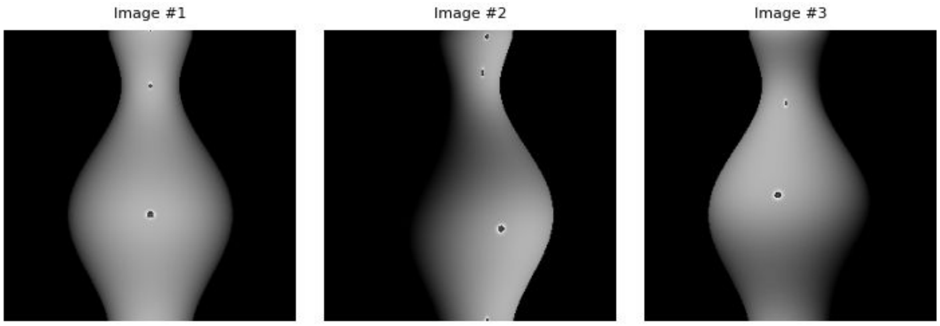


3D surface plot from 3 viewpoints:



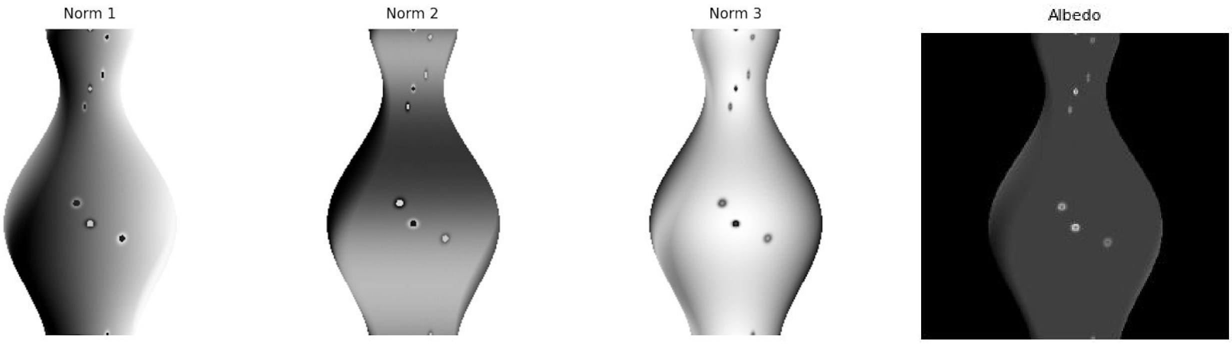
2.3 shiny_vase

This dataset shows the limitations of Lambert's law. It consists of several images of a vase with a reflective surface. The albedo and the norms seem to consist of rings around the spots where light was reflected. On the 3D image reconstruction, there are very noticeable spikes in the reflection points, courtesy of the light not being absorbed. That creates a spike in the albedo, which in turn creates a spike in the normalized normals. We conduct 4 experiments: two with using the inverse method with and without smoothing, and two using RANSAC with and without smoothing. There are no noticeable differences between the inverse method and RANSAC, but applying smoothing makes a visible change, removing the spikes that were caused by the reflections. Unfortunately, even though the spikes are gone, the white rings are still present in the albedo, normals and 3D surface. Moreover, the dataset consists of only three images, bringing forth the problem discussed in the previous two datasets.

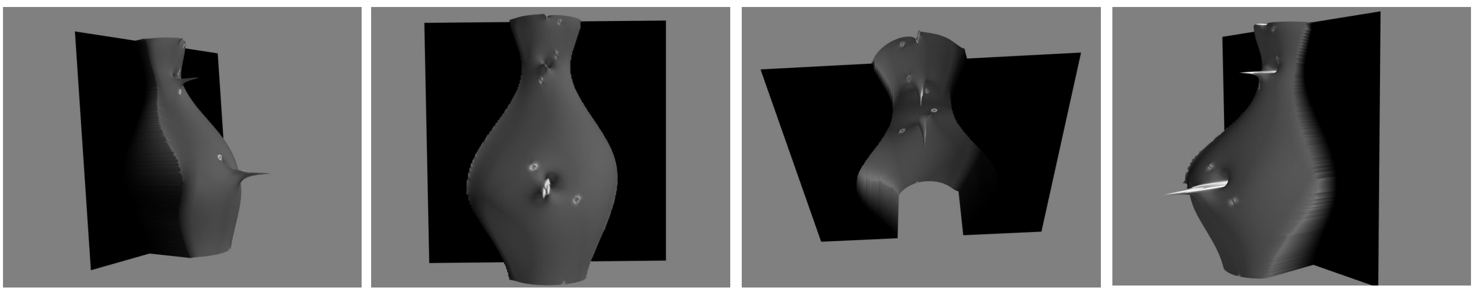


[Inverse method, no smoothing]

Images, albedo and normals plot:

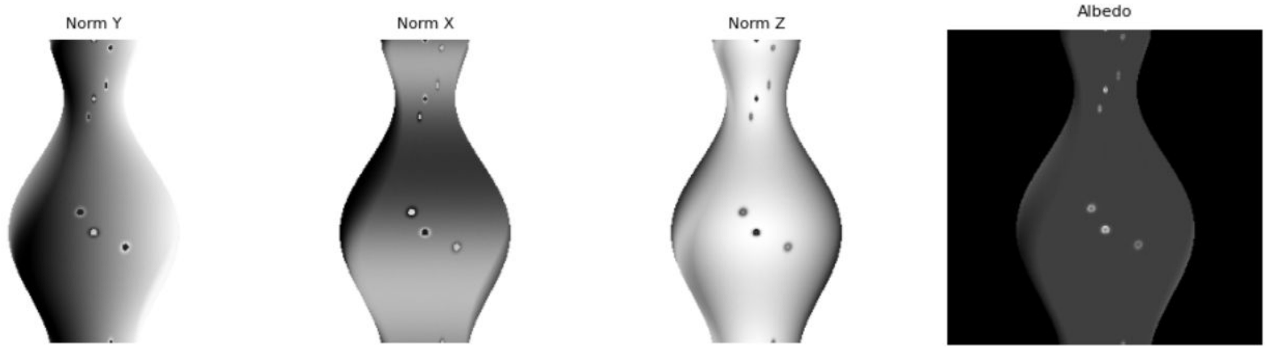


3D surface plot from 4 viewpoints:

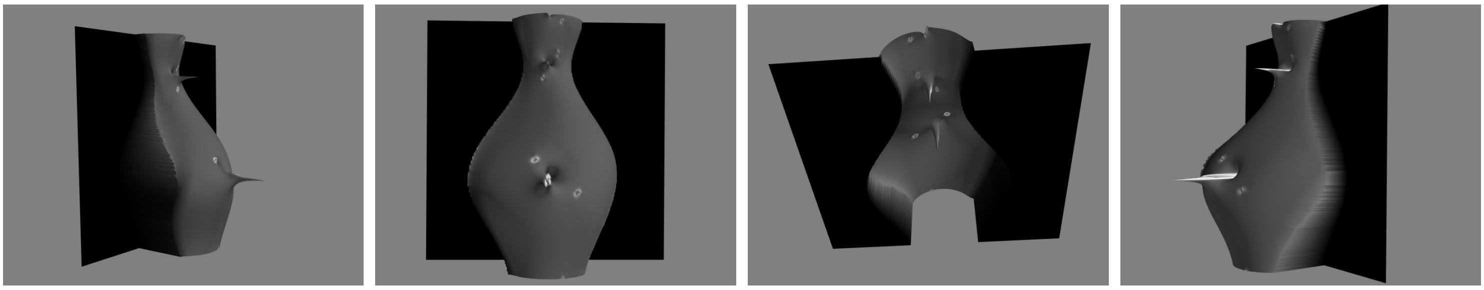


[RANSAC method, no smoothing]

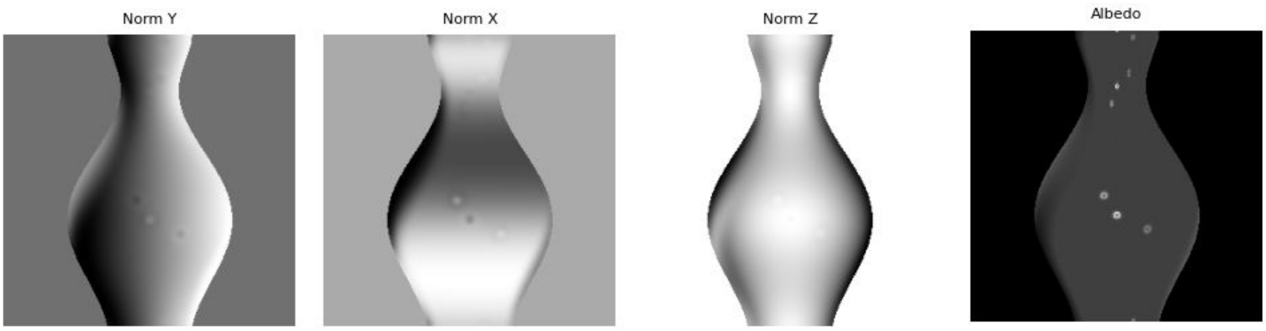
Albedo and normals:



3D surface plot from 4 viewpoints:

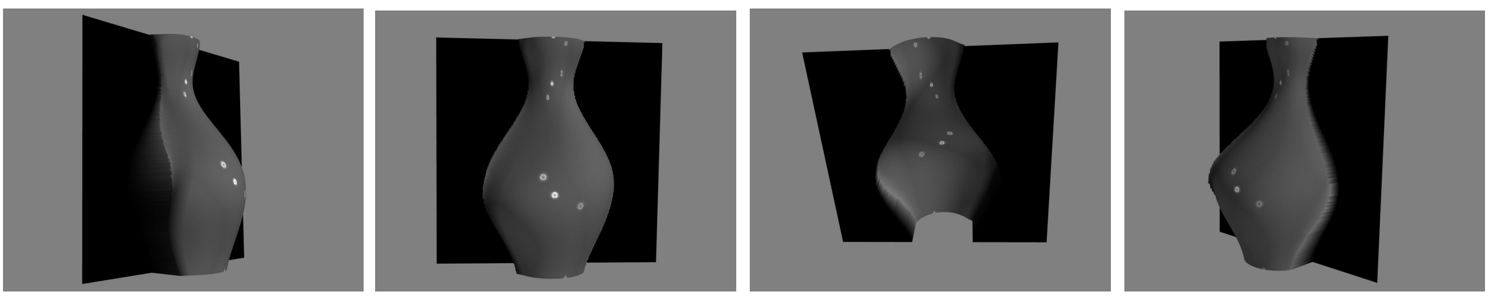


[Inverse method with smoothing]



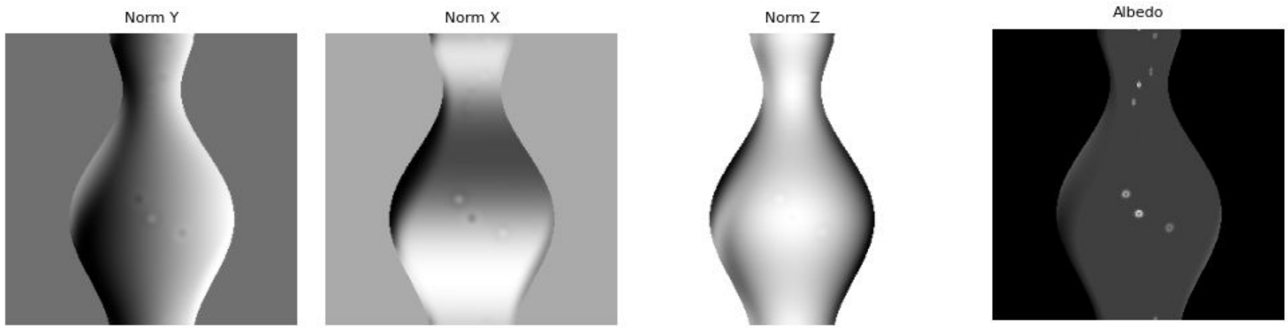
Image, albedo and normals plot:

3D surface plot from 4 viewpoints:

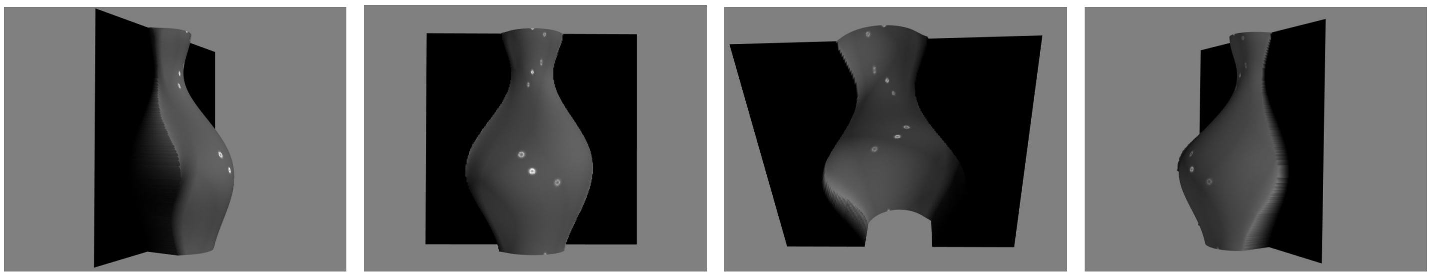


[RANSAC method with smoothing]

Image, albedo and normals plot:



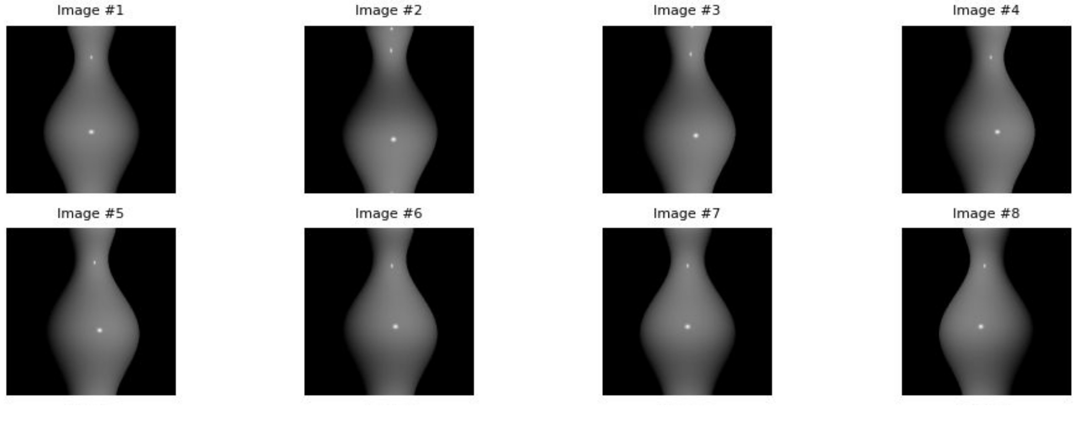
3D surface plot from 4 viewpoints:



2.4 shiny_vase2

Starting with this dataset, we use the Moore-Penrose pseudoinverse to compute the *m* vector, as the object's surface is reflective. This means that Lambert's law in its simplest form is not applicable anymore. As with the previous dataset, we run 4 experiments: pseudoinverse, RANSAC, with and without smoothing. This dataset includes 22 images, which made the resulting 3D surface much more symmetric and smooth, compared to the previous experiment. That is because we have much more information from 22 pictures to extract the shape. Also, as can be seen in the pictures, since we have a lot of lighting positions, the position of the reflection changes as well. So, for the norms and albedo we have a kind of a group of point reflections. With an inverse experiment without smoothing, we get a "bumpy" surface at that area. While with a smoothing, the reflection takes the form of a spot, not an extrusion.

First 8 original images:



[Inverse method, no smoothing]

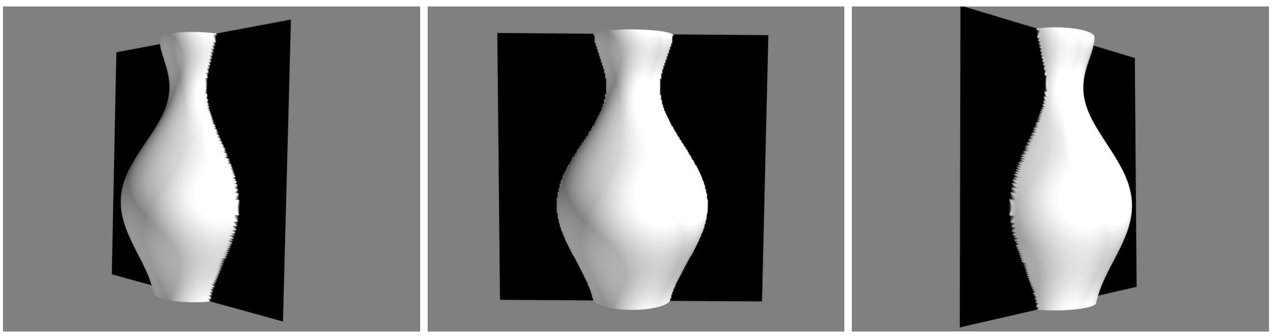
Images, albedo and normals plot:

3D surface plot from 3 viewpoints:

[RANSAC method, no smoothing]

Images, albedo and normals plot:

3D surface plot from 3 viewpoints:



[Inverse method with smoothing]

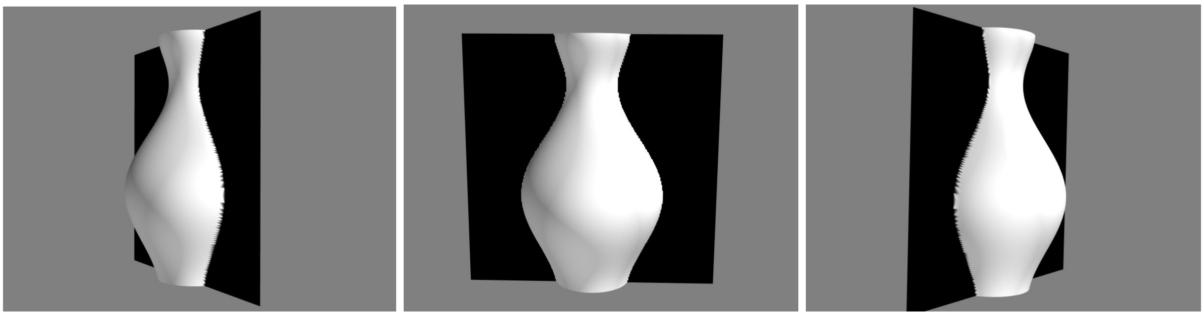
Images, albedo and normals plot:

3D surface plot from 3 viewpoints:

[RANSAC method with smoothing]

Images, albedo and normals plot:

3D surface plot from 3 viewpoints:



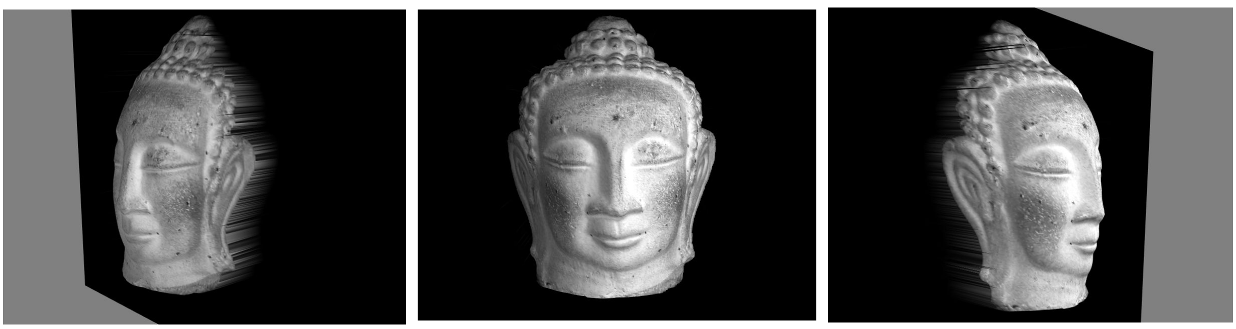
2.5 Buddha

Buddha dataset is composed of 10 images. When getting a 3D image with inverse method and smoothing, the results are quite satisfactory. The same is for the smoothed RANSAC. However, when RANSAC was applied for the first time without smoothing there was a big spike above the buddha's head, but the second time it was run any large spikes were absent. That is due to the fact that RANSAC performs random sampling, and it can start randomly from a non-optimal appropriate point. Furthermore, we need to mention that in this dataset there are zero values present inside the mask area, across all images. That causes the *m* (albedo-modulated normal) and the albedo to contain zeros (about 170 pixels). This makes the n array have NaN values due to zero division, which, in turn, leads to *z* having NaN values. Coupled with the mayavi bug, the image did not plot at all because of those 0 pixels. That was fixed by replacing the zeros in the albedo and modulated normals to ones before calculating the n array (n = m/albedo). The result was some barely visible spikes in the 3D surface around the top-left/top-right edges of the head.

[Inverse method, no smoothing]

Image, albedo and normals plot:

3D surface plot from 3 viewpoints:

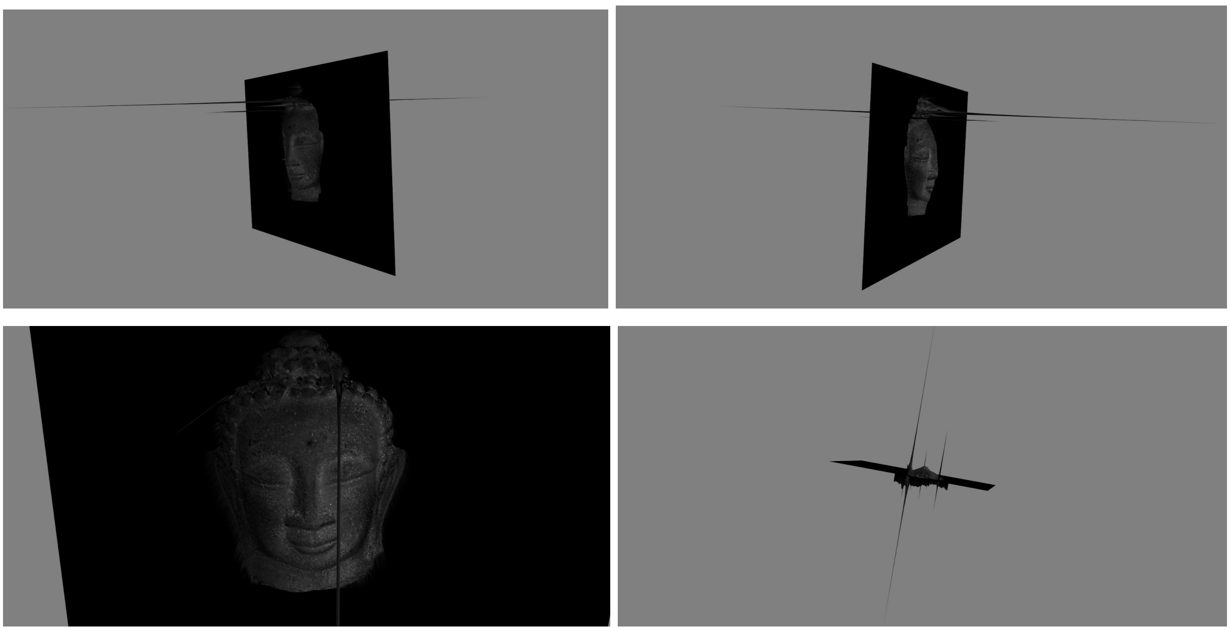


[RANSAC method, no smoothing]

Image, albedo and normals plot:

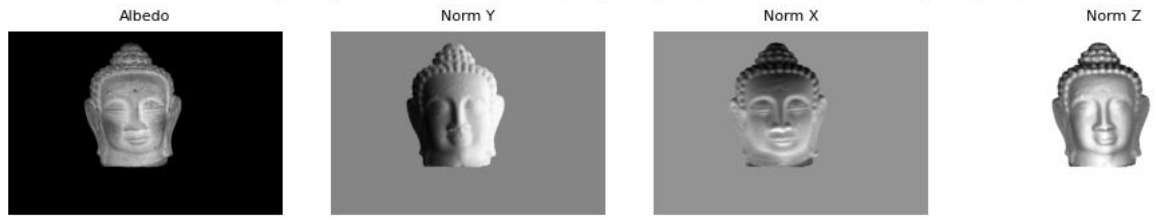
3D surface plot from 3 viewpoints *without peaks:*

3D surface plot from 4 viewpoints *with peaks:*

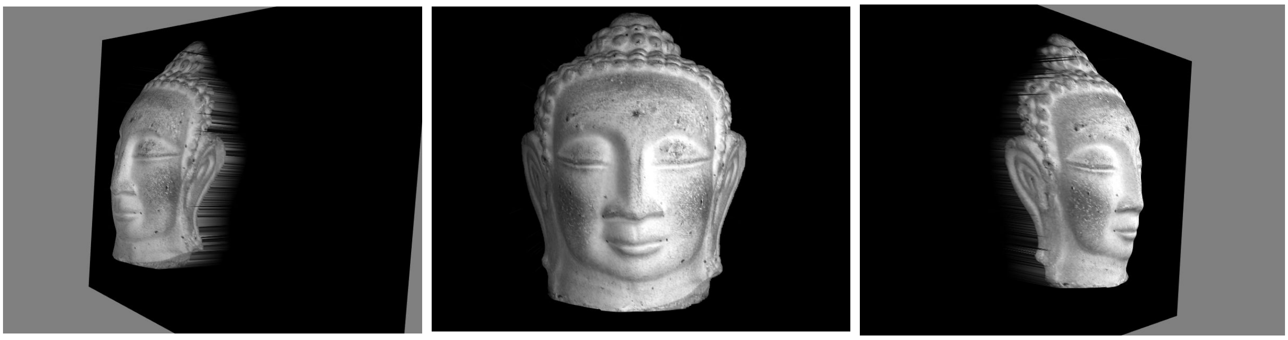


[Inverse method with smoothing]

Image, albedo and normals plot:



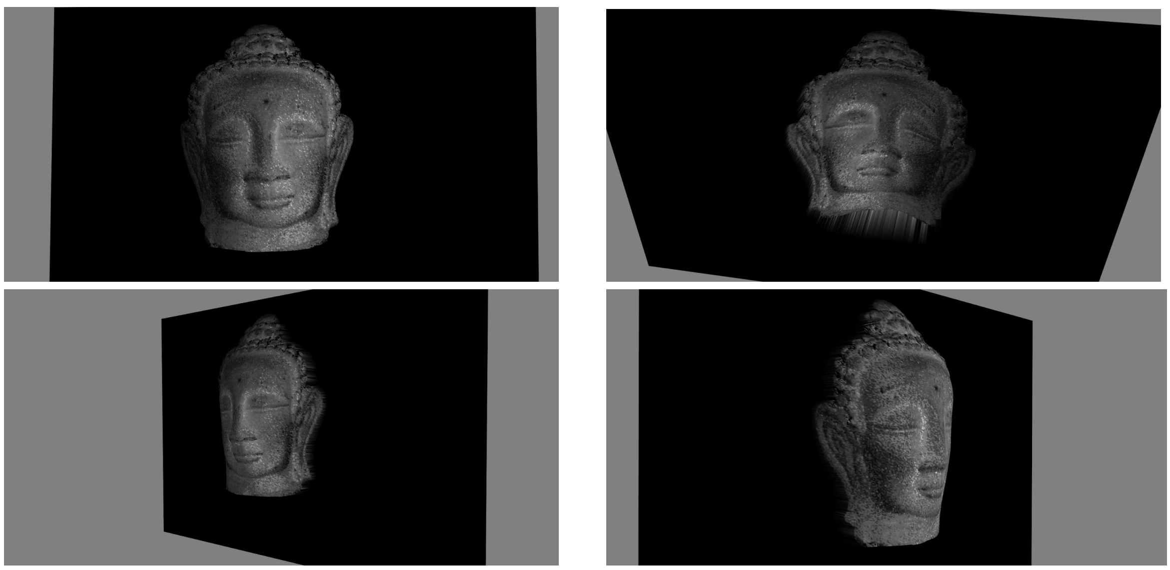
3D surface plot from 3 viewpoints:



[RANSAC method with smoothing]

Image, albedo and normals plot:

3D surface plot from 4 viewpoints:

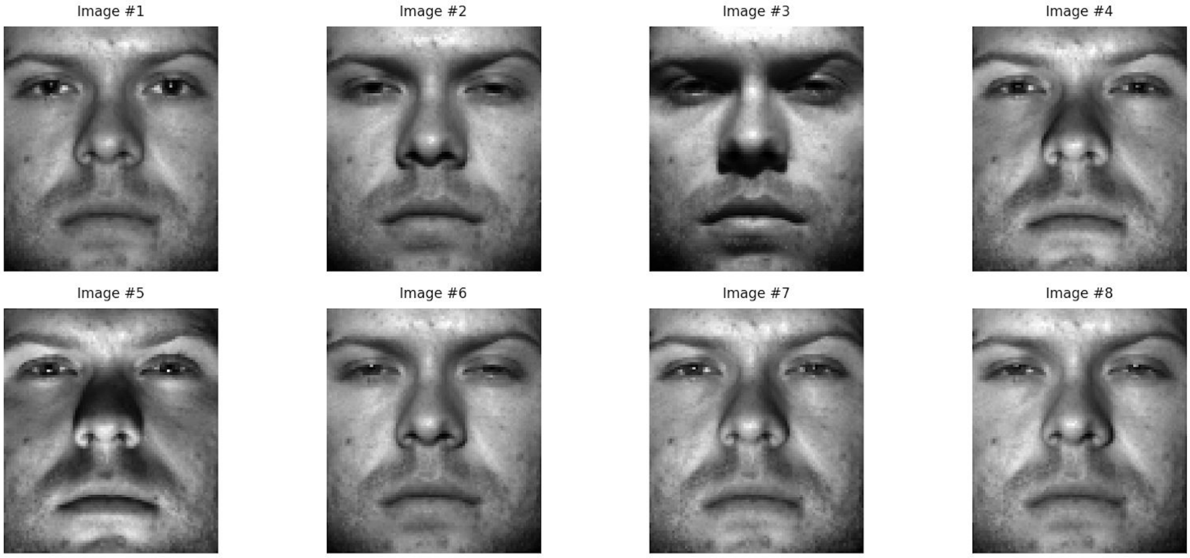


2.6 Face

This dataset consists of 27 photographs, with more critical changes between them. For example, in some pictures the eyes are squinted, and in some harsh shadows are cast by the nose. Thus, we have to sample some data in order to get a realistic picture, so we use RANSAC. We performed 2 experiments: RANSAC with barely any smoothing (2 iterations) and RANSAC with thorough smoothing (200 iterations). A threshold of 10 was used. Both render quite good results, and provide well-distinguishable features and shape.

However, it seems that the picture with light smoothing gives a more detailed texture and a realistic shape, which is probably better for displaying a human face.

The first 8 original images:



[RANSAC method with light smoothing (2 iterations)]
Image, albedo and normals plot:

3D surface plot from 3 viewpoints:

[RANSAC method with thorough smoothing (200 iterations)]
Image, albedo and normals plot:

3D surface plot from 3 viewpoints:

References:

1. Forsyth, D., & Ponce, J. (2011). Computer Vision - A Modern Approach, Second Edition.