# Report for assignment 1

*Moser Stefan*
*09-277-013*

## 1 Photometric Stereo (Due on 29/10/2013)

In this assignment, we first retrieve the light directions of twelve images of a chrome sphere. Then we use these light directions to compute the normals, albedo and depth of a scene using twelve images that were taken under the same lighting conditions as measured before. We assume, the chrome sphere does specular reflection, while the other scenes only do diffuse reflection (as lambertian materials would).

### 1.1 Calculating the light direction

We know, that every point on the sphere must satisfy

$$r^2 = x^2 + y^2 + z^2 \tag{1}$$

for the cartesian representation the surface of a sphere as $(x, y, z)^T$ with the radius $r$ and origin at the center of the sphere. This enables us to express the point of specular reflection in sphere coordinates. For the z coordinate, eq. **??** can be solved for z. The negative solution is the correct choice, since it is the one pointing towards the camera. Because the shape is a sphere, we also know that $n_i$ is perpendicular to the surface, ie. the normal. Putting all this information into one equation, we end up with

$$n_i = \begin{bmatrix} (p_i - c)_x \\ (p_i - c)_y \\ -\sqrt{r^2 - (p_i - c)_x^2 - (p_i - c)_y^2} \end{bmatrix} \tag{2}$$

with

- $p_i$: The xy coordinates of the light source $i$'s highlight on the sphere. It is computed as centroid of all points above a certain brightness threshold
- $c$: The xy coordinates of the centre of the sphere. computed as the centroid of the mask.

Once $n_i$ is retrieved, it can be used with ray reflection to get the direction towards the light

$$L_i = d - 2(d \cdot \hat{n}_i)\hat{n}_i \tag{3}$$

with $\hat{n}_i$ being the normalized vector of $n_i$ and $d$ the direction of a light ray from the camera towards the sphere. The assumption given in the assignment is, that every ray originating from the camera has direction $d = (0, 0, 1)$ for every point of the image.

All light directions $L_i$ can be assembled into a matrix $\mathbf{L}$. Here the transposed version is shown, since it is easier to fit on a page.

$$\mathbf{L}^T = \begin{bmatrix} 0.4507 & -0.4230 & -0.7861 \\ 0.2180 & -0.1229 & -0.9682 \\ -0.0345 & -0.1577 & -0.9869 \\ -0.0854 & -0.3983 & -0.9133 \\ -0.2897 & -0.4593 & -0.8397 \\ -0.1012 & -0.5080 & -0.8554 \\ 0.2540 & -0.3823 & -0.8885 \\ 0.0908 & -0.3885 & -0.9170 \\ 0.1856 & -0.3014 & -0.9353 \\ 0.0794 & -0.2990 & -0.9510 \\ 0.1158 & -0.0415 & -0.9924 \\ -0.1295 & -0.3261 & -0.9364 \end{bmatrix}$$

They are fairly close to the defaults given. The mean squared error over all normalized directions is 0.0217.

## 1.2 Computing Surface Normals and Grey Albedo

For computing the grayscale albedo, I first reduced the color information to a single channel by computing the luminance $\ell$

$$\ell = c_{\mathrm{r}} \cdot 0.299 + c_{\mathrm{g}} \cdot 0.587 + c_{\mathrm{b}} \cdot 0.114 \tag{4}$$

for each image.

In the lecture it was shown that we can compute the normals by solving

$$\begin{aligned} I &= S\tilde{n} \\ S^T I &= S^T \tilde{n} \\ \tilde{n} &= (S^T S)^{-1} S^T I \end{aligned}$$

with

- $I$ a vector of the luminance values of the inspected pixel for all $n$ images
- $S$ a matrix of all light directions, with each row having the light direction of the corresponding image.
- $\tilde{n}$ the normal direction (not unit size)

We know that we can obtain the grayscale albedo $\rho$ by taking the length of $\tilde{n}$

$$n = \frac{\tilde{n}}{\rho} \tag{5}$$

Once we know the normals, we can use them to compute the color albedo $a$, using

$$\begin{aligned} I &= a(n \cdot L) \\ (n \cdot L)^{-1} I &= a \end{aligned}$$

on every pixel.

## 1.3 Surface Fitting

We are looking for a way, to express the surface depth as function of the normal. The surface $S$ can be defined as with the depth as function of the other coordinates, as in $S(X,Y) = (X, Y, Z(X,Y))^T$. Next, we define the surface tangent vectors in $X, Y$ direction at any point as partial derivatives

$$\begin{aligned} \mathbf{t}_X &= \frac{\partial \mathbf{S}}{\partial X} = (1, 0, Z_X)^T \\ \mathbf{t}_Y &= \frac{\partial \mathbf{S}}{\partial Y} = (1, 0, Z_Y)^T \end{aligned} \tag{6}$$

Since the normal can be expressed as normalized cross product of two tangential vectors, we know that

$$\begin{aligned} n(X,Y) &= \frac{\mathbf{t}_X \times \mathbf{t}_Y}{||\mathbf{t}_X \times \mathbf{t}_Y||} \\ &= \frac{(Z_X, Z_Y, -1)^T}{||(Z_X, Z_Y, -1)||} \end{aligned} \tag{7}$$

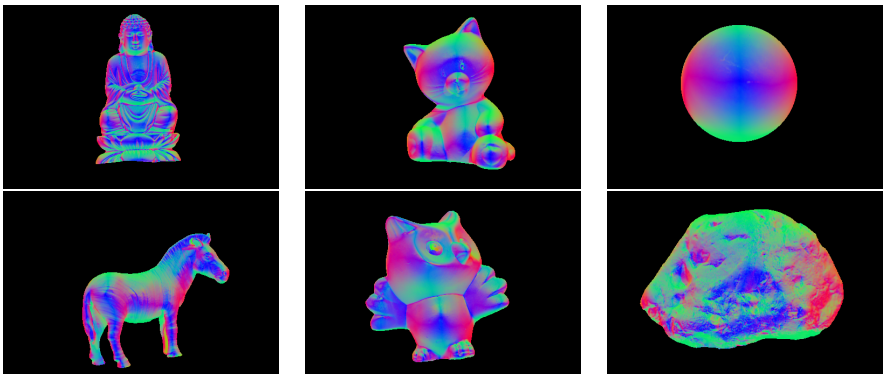

Fig. 1: The normals visualized for every scene. For transforming them into rgb range, I took the absolute value. The x-axis is pointing to the right, the y-axis up and the z-axis towards the reader. The background (where the mask is zero) is shown black.

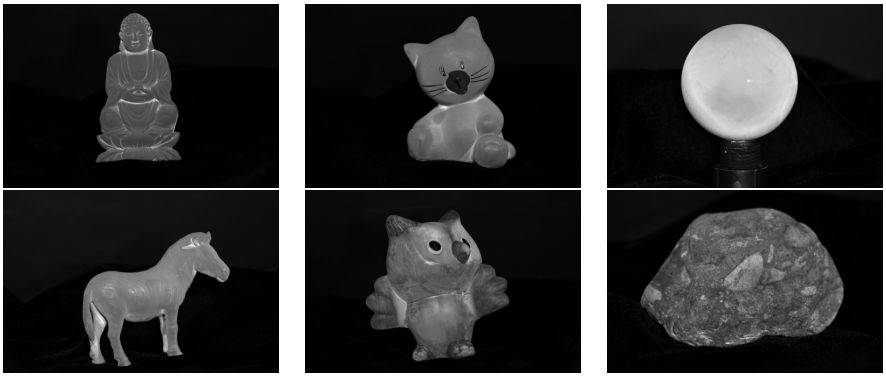

Fig. 2: The grayscale albedo (luminance)

Further, the derivatives of the unknown depths can be approximated by

$$\begin{aligned} Z_X &\approx Z(x+1,y) - Z(x,y) \\ Z_Y &\approx Z(x,y+1) - Z(x,y) \end{aligned} \tag{8}$$

These approximations can be used to express the tangent vectors defined in eq. **??**. Since we know, that the normal and the tangent vector are perpendicular to each other, we can formulate the constraint

$$\begin{aligned} \mathbf{t}_X(x,y) \cdot n(x,y) &= 0 \\ \mathbf{t}_Y(x,y) \cdot n(x,y) &= 0 \end{aligned} \tag{9}$$

This can be transformed to

$$\begin{aligned} (Z(x+1,y) - Z(x,y)) \cdot n(x,y)_z &= -n(x,y)_x \\ (Z(x,y+1) - Z(x,y)) \cdot n(x,y)_z &= -n(x,y)_y \end{aligned} \tag{10}$$

Fig. 3: The color albedo. The buddha, sphere and horse do in fact have a gray albedo, so there is not too much of a difference to the grayscale albedo.

This gives us a linear system we can solve for. The constraints introduced so far reduce the space of solutions to $Z(x, y) + c$ for any constant $c$. To reduce it to one unique solution, an additional constraint in the form of

$$Z(x_0, y_0) = 0 \tag{11}$$

must be defined for a single pixel $(x_0, y_0)$. I choose to set the pixel in the top left to $Z(0, 0) = 0$. An alternative I explored would be to set the depth of all pixels just outside the object boarder to 0.

### 1.3.1 Results

A successful reconstruction of the surface is never guaranteed, but having more images with more diverse light directions is one way to optimize the result. With more images, the impact of self occlusion and noise is reduced. The depth map of the scenes given for this assignment can be seen in figure **??**.

### 1.3.2 Failure cases

- Materials are assumed to reflect light as a lambertian material would. If the material does have specular highlights, this method will fail.
- Self occlusion renders this method difficult, since this methods does not handle shadows.
- Inter reflection is not handled.

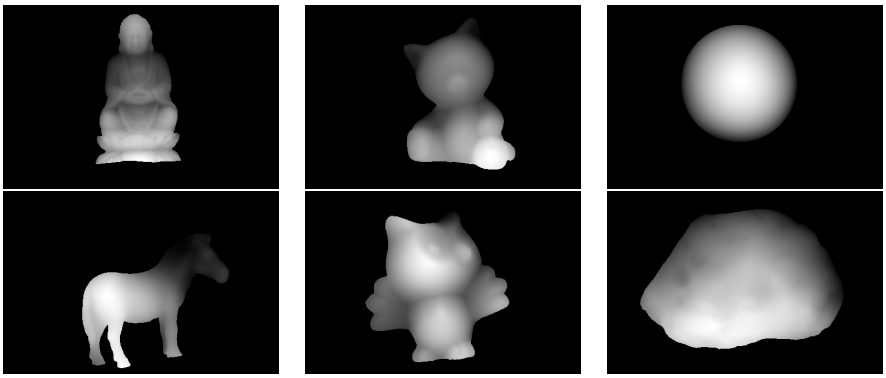

Fig. 4: The depth. Brighter means closer to the camera. The background is set to black.

- The surface must be sufficiently smooth. Objects with many sharp edges would not give a good result.