Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit

Variational Perspective Photometric Stereo

Andreas Triebl

Course of Study: Informatik

Examiner: Prof. Dr. Andrés Bruhn

Supervisor: Daniel Maurer

Commenced: 2017-03-20

Completed: 2017-09-20

CR-Classification: I.4.8, I.4.9

Abstract

In this thesis we present a method to estimate the surface orientation of a 3D object. The general technique is called Photometric Stereo (PS) since we use several 2D images taken from the same location while the illumination changes for each image. Therefore, we use the varying intensities for each pixel to estimate the surface normal vector. In order to compute the estimation of the surface normals we used a variational approach and derived an energy functional depending on the Cartesian depth z. This energy functional is like a cost function that we want to minimise to obtain a good estimation of the shape of the test object. For the minimisation technique we used the method of Maurer et al. [MJBB15] to overcome the difficulties in the minimisation of the energy functional and efficiently reach a global minimum. Further, we present three variants of this model that use different illumination models and two model extensions. Finally, we compare the performances of all the variants of the PS model in different experiments.

Contents

List of Figures

List of Tables

1 Introduction

In the field of computer vision one of the earliest and most studied problems is the so called shape reconstruction. As the name implies one tries to reconstruct the shape of an three-dimensional object given only two-dimensional images. Over the time many different methods have been developed to compute the missing depth component for the pixels that represent the said object in order to compute the corresponding real world coordinates. The probably most intuitive way to obtain the depth is by using triangulation. But for this stereo methods two images from different known viewpoints are needed and the corresponding points in both images have to be identified. Therefore, a lot of methods have been presented that only require one camera to infer the shape by using different cues, for instance shading, texture, or focus. The technique we are using in this thesis is called *Photometric Stereo (PS)*, which is basically a more reliable way of the *Shape from Shading (SfS)* method. In general, both techniques use the intensity variation on the object to recover the normal vector of the surface for every pixel. But PS uses multiple images taken from the same location under a varying location of the light source, instead of using only one image where the illumination source is located in the camera centre. This is also the reason why the PS approach is called *photometric*, since it uses the pixel intensities in consecutive images to estimate the surface orientation. Hence, the correspondence problem is non-existent.

In this work we present a variational PS approach where a perspective camera model and a Lambertian reflectance model is assumed. Additionally, we compare three variants of this approach where we use a different illumination model in each variant. Namely, a single distant point light source, a near point light source model with an integrated light attenuation over distance, and a general lighting model based on spherical harmonics.

1.1 Related Work

The first one who introduced the PS method was Robert J. Woodham in 1980 [Woo80]. In this original PS model an orthographic camera model, a single distant light source and a Lambertian reflectance model were assumed.

The first one who introduced the perspective viewing geometry in a Shape-from-Shading approach was Horn [Hor75] in the 1970s. Later in 2005 this perspective method was adapted to a Photometric Stereo approach by Tankus and Kiryati [TK05]. Regarding the variational approach in our model we focused on the work of Maurer et al. [MJBB15] where they presented an efficient linearisation approach, based on the warping strategy by Brox et al. [BBPW04], for the variational SfS model by Ju et al [JMBB16]. This model includes a Cartesian depth parameterisation of the surface, a perspective camera model, and also a Lambertian reflectance model. In this work, we adapted this approach to our variational perspective PS model.

1.2 Organisation

Let us briefly mention how this thesis is structured. In the next Chapter 2 we cover the mathematical foundations that are necessary to understand the proposed PS model in this work. This contains the camera geometry, the reflectance model, the different illumination models, and some basics of the calculus of variations. Chapter 3 addresses the setup of our variational PS model and the efficient minimisation technique used. After that we present the results of the evaluation in Chapter 4 where we compare the performance of this model under different illumination models in different experiments. The last Chapter 5 gives a conclusion of this thesis.

2 Foundations of Photometric Stereo

In this chapter we provide the mathematical foundations that are utilised to build our variational PS model. To back up the information given in this chapter we refer to the books [HZ04] for the background related to camera geometry and [Sze10] for the section covering the reflectance model.

2.1 Perspective Camera Model

The perspective camera model, which is also known as the pinhole camera model, is a simple but realistic model of a camera system. This camera model uses a perspective projection of the 3D space onto a 2D image plane to generate 2D images from a three-dimensional world. Figure Figure 2.1 shows how 3D camera coordinates of a scene point $\mathbf{M} = (X, Y, Z)^\top$ are mapped to the corresponding image point in 2D image coordinates

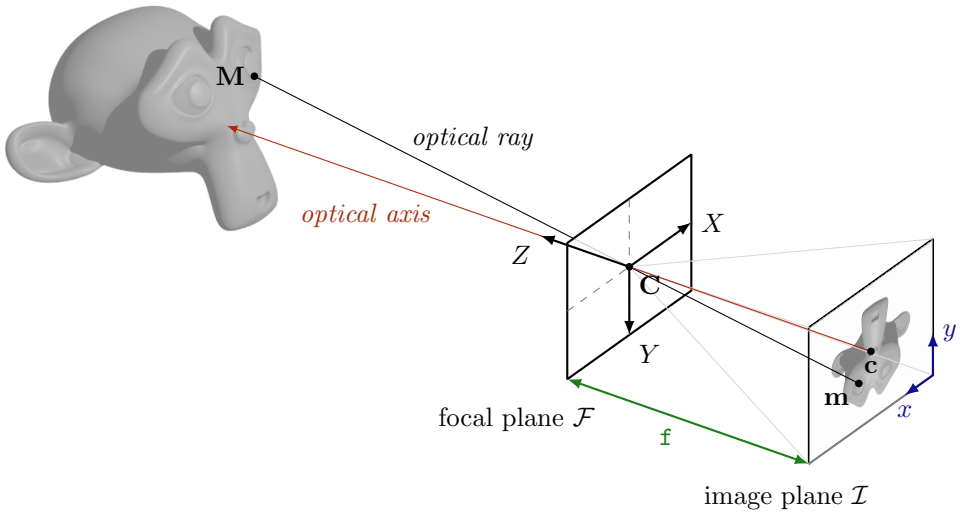


Figure 2.1: Perspective camera model.

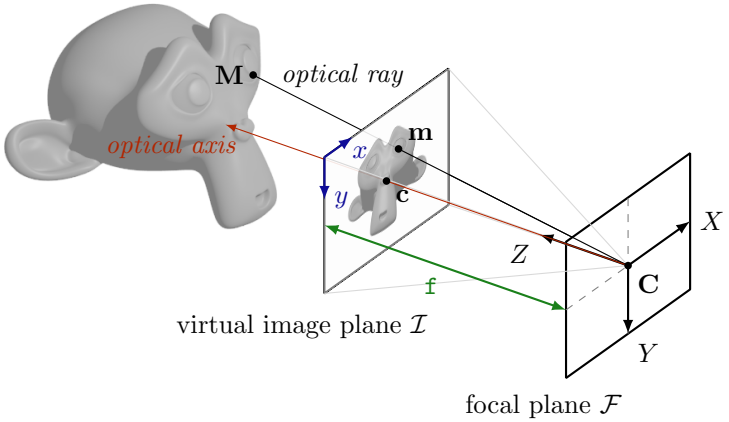


Figure 2.2: Perspective camera model with virtual image plane in front of focal plane.

$\mathbf{m} = (x, y)^\top$ on the image plane $\mathcal{I}$. The optical centre of the camera, also known as focal point, is denoted by $\mathbf{C}$ and lies on the focal plane $\mathcal{F}$. The optical axis is orthogonal to the image plane and passes through $\mathbf{C}$, the point where the optical axis intersects the image plane is called principal point $\mathbf{c}$ and $\mathtt{f}$ stands for the focal length which denotes the distance between the image plane and the optical centre $\mathbf{C}$.

2.1.1 Simplified Camera Model

The drawback of the perspective camera model is that the projective mapping inverts the orientation which means that objects that are oriented upwards in the real world appear downwards in the image. To counter that we use a simplified model instead where a virtual image plane is considered that is placed in front of the focal plane with distance $\mathtt{f}$. Therefore, the orientation of real world objects remains the same in the image. This simplification of the perspective camera model is shown in Figure 2.2. With a closer look at Figure 2.2, one can notice that the *theorem of intersecting lines* can describe the relation between the 3D point $\mathbf{M}$ and its projection $\mathbf{m}$ as shown in Figure 2.3 for the y-direction, since the optical ray and the optical axis are both intersected by the image plane and a parallel plane where $\mathbf{M}$ lies on.

With that said, the theorem of intersecting lines yields the following basic equation

$$\frac{x}{X} = \frac{y}{Y} = \frac{\mathtt{f}}{Z}\,, \tag{2.1}$$

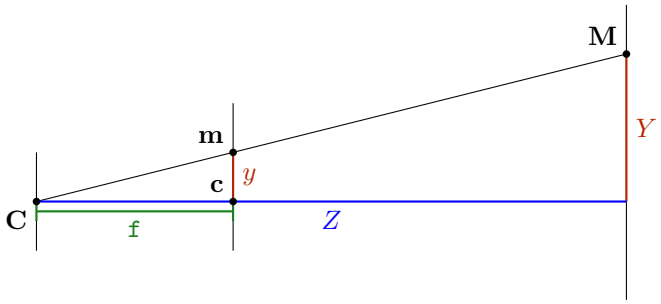


Figure 2.3: Theorem of intersecting lines describes ratio between $\mathbf{M}$ and $\mathbf{m}$.

hence, we know that the equations for the coordinates of $\mathbf{m}$ are

$$x = \frac{\mathtt{f}X}{Z} \quad \text{and} \quad y = \frac{\mathtt{f}Y}{Z} \; . \tag{2.2}$$

Consequently, all the points on the corresponding optical ray are projected on the same image point $\mathbf{m}$. This loss of depth information is called depth ambiguity.

Homogeneous Coordinates Now we introduce homogeneous coordinates, which allows us to describe the nonlinear projective transformation in terms of simple matrix operations. For our case, the homogeneous coordinates for the image point $\mathbf{m}$ and the scene point $\mathbf{M}$ are defined as follows:

$$\tilde{\mathbf{m}} := \begin{pmatrix} \mathbf{m} \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \; , \quad \tilde{\mathbf{M}} := \begin{pmatrix} \mathbf{M} \\ 1 \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \; , \tag{2.3}$$

and for the back transformation, we have to divide the first two coordinates by the third

$$\begin{pmatrix} \tilde{x}/\tilde{z} \\ \tilde{y}/\tilde{z} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} = \mathbf{m} \; . \tag{2.4}$$

Further, we have to mention, that homogeneous coordinates can represent points that have no Euclidean equivalent. This is the case, when the last entry of a point in homogeneous coordinates equals zero. By looking at the back transformation in Equation (2.4), we see that this would lead to a division by zero. Such points can be interpreted as points at infinity.

Now we can use homogeneous coordinates with the scaling factor Z and the equations for the coordinates of $\mathbf{m}$ (2.2) to obtain a linear presentation of the nonlinear transformation:

$$Z\tilde{\mathbf{m}} = Z\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} Zx \\ Zy \\ Z \end{pmatrix} = \underbrace{\begin{pmatrix} \mathtt{f} & 0 & 0 & 0 \\ 0 & \mathtt{f} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\text{projection matrix}} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = P\tilde{\mathbf{M}} \,, \tag{2.5}$$

where the 3×4 matrix P is called projection matrix and the tilde stands for vectors in homogeneous coordinates that use one additional dimension.

2.1.2 Extrinsic and Intrinsic Parameters

The transitions between the world coordinate system, camera coordinate system, image coordinate system, and pixel coordinate system depend on the position and orientation of the camera and the internal camera characteristics. The extrinsic and intrinsic camera parameters characterise these transformations.

Extrinsic camera parameters. The extrinsic parameters denote the position and orientation of the camera coordinate system relative to the world coordinate system. This transformation can be expressed in homogeneous coordinates through a 4×4 matrix T. In this transformation matrix information on the position is given by the translation vector $t \in \mathbb{R}^3$ that reads $(t_1, t_2, t_3)^\top$ in Euclidean coordinates. The rotation matrix contains the orientational information and can be represented in homogeneous coordinates as a 4×4 matrix $R \in SO(3)$. The translation vector and the rotation matrix look as follows:

$$t = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \,, \quad R = \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & 0 \\ r_{2,1} & r_{2,2} & r_{2,3} & 0 \\ r_{3,1} & r_{3,2} & r_{3,3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \,. \tag{2.6}$$

The translation and rotation can be concatenated to obtain the extrinsic matrix that transforms world coordinates to camera coordinates. In homogeneous coordinates this yields the 4×4 matrix T as mentioned above:

$$T = \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} . \tag{2.7}$$

The extrinsic matrix offers six degrees of freedom, that result from the three degrees of freedom given by the translation vector t and the three degrees of freedom that are offered by the three 1D rotation angles around each axis (ϕ_X, phi_Y, phi_Z). All in all, the transformation from world coordinates to image coordinates via camera coordinates is given by the following projective transformation:

$$Z \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} Zx \\ Zy \\ Z \end{pmatrix} = \underbrace{\begin{pmatrix} \mathtt{f} & 0 & 0 & 0 \\ 0 & \mathtt{f} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\text{projection matrix}} \underbrace{\begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\text{extrinsic matrix}} \begin{pmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{pmatrix} . \tag{2.8}$$

Intrinsic camera parameters. The intrinsic parameters characterise the geometry of the image plane inside the camera with respect to the image coordinate system. These parameters include the location of the principal point in this coordinate system $(o_x, o_y)^\top$, a spacing factor of the pixel in x-direction h_x and in y-direction h_y, and the skew s between the coordinate axis due to the sensor not being mounted perpendicular to the optical axis.

These five parameters lead to the intrinsic matrix K that therefore offers five degrees of freedom and describes the transformation from the image coordinates to the pixel coordinates:

$$K = \begin{pmatrix} 1/h_x & s/\mathtt{f} & o_x \\ 0 & 1/h_y & o_y \\ 0 & 0 & 1 \end{pmatrix} . \tag{2.9}$$

Projective Mapping. To get the full projective mapping from a 3D point in world coordinates $(X_W, Y_W, Z_W, 1)^\top$ to a 2D image point in pixel coordinates $(Zu, Zv, Z)^\top$, we have to concatenate the extrinsic, projection and intrinsic matrices:

$$\begin{pmatrix} Zu \\ Zv \\ Z \end{pmatrix} = \underbrace{\begin{pmatrix} 1/h_x & s/\mathtt{f} & o_x \\ 0 & 1/h_y & o_y \\ 0 & 0 & 1 \end{pmatrix}}_{\text{intrinsic matrix}} \underbrace{\begin{pmatrix} \mathtt{f} & 0 & 0 & 0 \\ 0 & \mathtt{f} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\text{projection matrix}} \underbrace{\begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\text{extrinsic matrix}} \begin{pmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{pmatrix} . \tag{2.10}$$

Thus, the full projection matrix is a 3×4 matrix, which offers 11 degrees of freedom resulting from the six extrinsic parameters and five intrinsic parameters without the focal length.

2.2 Reflectance Model

Generally speaking, two factors determine the radiation properties of a surface as explained in [Sze10]. The first one is the illumination direction which is described through the position of the surface relative to the light source. The second factor is the reflectance property of the surface, that depends on the optical properties of the surface material. Since, we use the Lambertian reflectance in our model, in the following section we will have a further look into the Lambertian reflectance model and describe its origin.

2.2.1 Bidirectional Reflectance Distribution Function

The *bidirectional Reflectance Distribution Function (BRDF)* is a general model of light scattering and describes the relation between the two factors mentioned above mathematically, namely the incoming light direction and the corresponding brightness reflected from the surface.

Let us consider a surface element that is illuminated by a single light source with a given light direction in polar coordinates (θ_i, ϕ_i). The corresponding energy to the incoming light is denoted by $E(\theta_i, \phi_i)$. $L(\theta_e, \phi_e)$ stands for the brightness that is emitted by the surface in the direction (θ_e, ϕ_e). The BRDF $f_r(\theta_i, \phi_i, \theta_e, \phi_e)$ describes the ratio between the emitted brightness $L(\theta_e, \phi_e)$ and the incoming energy $E(\theta_i, \phi_i)$:

$$f(\theta_i, \phi_i, \theta_e, \phi_e) := \frac{L(\theta_e, \phi_e)}{E(\theta_i, \phi_i)} , \tag{2.11}$$

usually it only depends on the difference angles:

$$f(\theta_i, \phi_i, \theta_e, \phi_e) = f(\theta_i - \theta_e, \phi_i - \phi_e) \ . \tag{2.12}$$

2.2.2 Lambertian Reflectance Model

Here we assume a Lambertian surface that reflects light in all directions. This property is also known as *diffuse reflection*. Since the light is scattered uniformly the BRDF is constant in this case:

$$f_{Lambert}(\theta_i, \phi_i, \theta_e, \phi_e) = \rho \ , \tag{2.13}$$

where $\rho \in [0, 1]$ is a material dependent coefficient, called *albedo* that determines the nonabsorbed fraction of the energy by the surface. The Lambertian reflectance is often used as a model for rough, unsmooth surfaces.

The reflected radiance R of a Lambertian surface in a point (X_W, Y_W, Z_W) depends on the orientation of the light source relative to the surface. More precisely, R is proportional to the cosine of the angle θ between the incoming light direction $\mathbf{L}$ and the surface normal vector $\mathbf{N}$. This can also be expressed as the dot product of the normalised surface normal vector and the normalised light direction vector:

$$R_{\rho,\mathbf{L}}(X_W, Y_W, Z_W) = \rho \cos\theta = \rho \, (\mathbf{L} \cdot \mathbf{N}) \ , \tag{2.14}$$

Thus, the reflected radiance is the largest, if the light source direction $\mathbf{L}$ is in the normal direction $\mathbf{N}$.

The corresponding grey value i in the image point $\mathbf{x}$ for the reflected radiance $R_{\rho,\mathbf{L}}$ in a scene point (X_W, Y_W, Z_W) is then given by the *image irradiance function*

$$i(\mathbf{x}) = R_{\rho,\mathbf{L}}(X_W, Y_W, Z_W) \ . \tag{2.15}$$

2.3 Illumination Models

In our work compare three different illumination models: a *distant light source*, a *near point light source,* and a general lighting model based on *Spherical Harmonics*. In the following section we introduce these models.

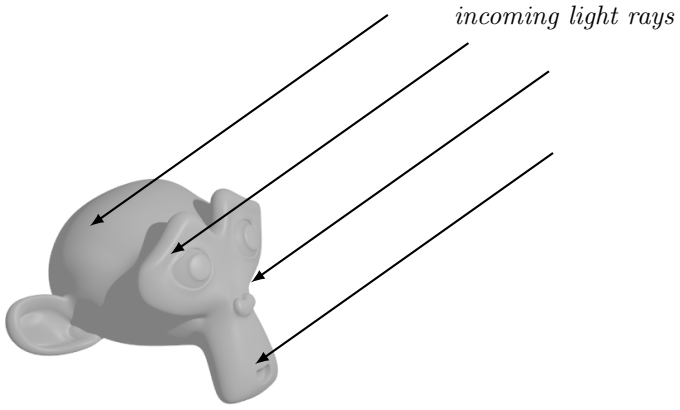


Figure 2.4: Distant illumination model with parallel rays.

2.3.1 Distant Light Source

In the first case, we consider a distant point light source that is located infinitely far away from the observed object. Thus, all the emitted light vectors can be assumed to be parallel to each other. With this said, the light direction vector L remains the same for every point on the object surface. This simple distant light source model is shown in Figure 2.4. Further, the intensity of the light does not vary with the distance, since the magnitude of each light direction vector is infinite. In contrast, the next illumination model we present considers this light attenuation over distance.

2.3.2 Near Point Light Source

In the second model, a point light source is assumed on the position $\mathbf{P}$. The point light source emits the light equally in all directions, hence, each point on the object surface has a different light vector. The light direction vector L, shown in Figure 2.5, can then be computed by subtracting the position of the current point on the surface $\mathbf{M}$ from the light position as shown in equation (2.16). How to derive the position of the current point on the surface is explained later in Section 3.1.

$$L(\mathbf{x}) = \mathbf{P} - \mathbf{M}\ ,\ \mathbf{L}(\mathbf{x}) = \frac{L(\mathbf{x})}{|L(\mathbf{x})|}\ . \tag{2.16}$$

As mentioned above, this illumination model takes the attenuation of the light intensity into account, therefore we give a brief introduction to the origin of this *light attenuation term* and how it is implemented in the near point light source model.

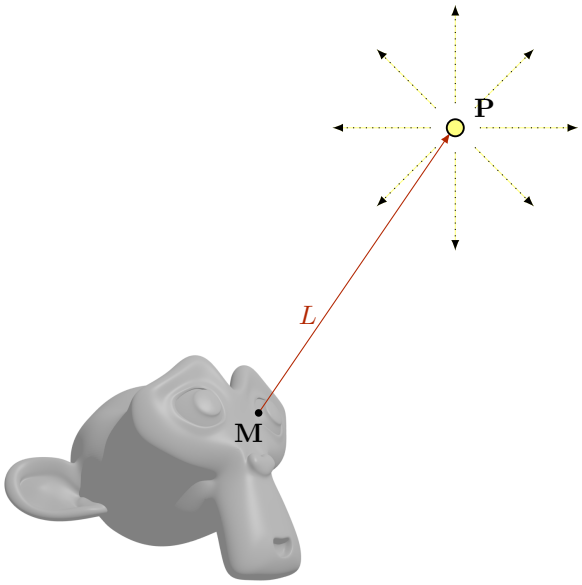


Figure 2.5: Near point illumination model.

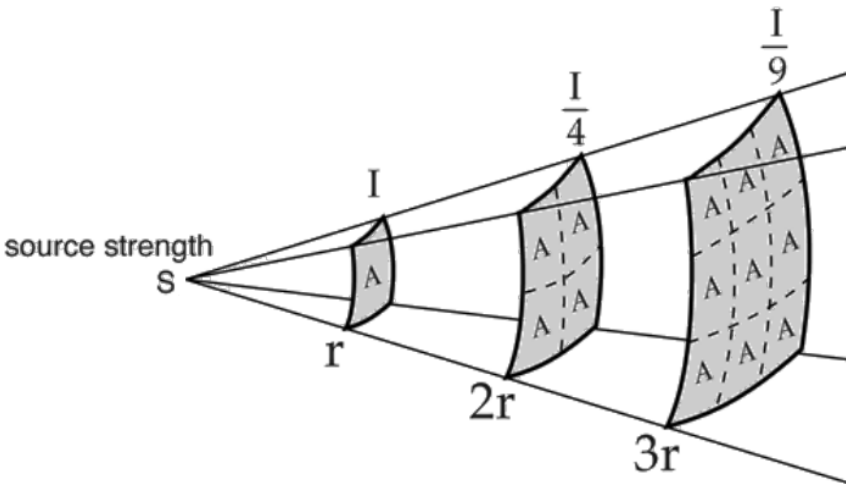
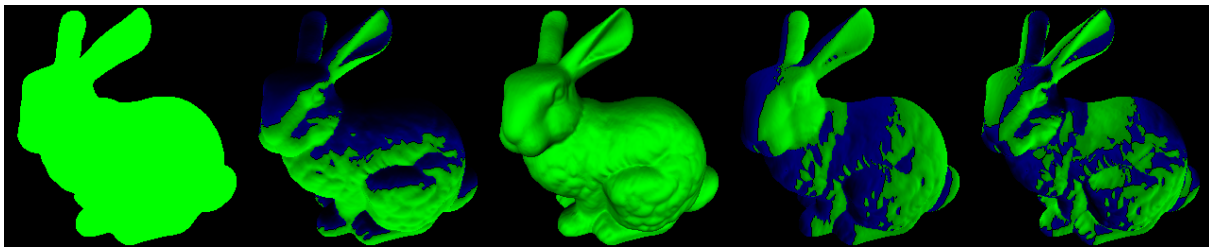


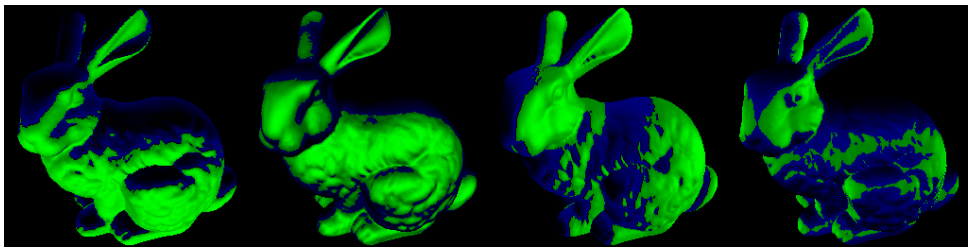
Figure 2.6: Inverse square law. The energy twice as far from the source is spread over four times the area, hence quarter the intensity [Nav16].

Light Attenuation. The *inverse square law* from physics states that the intensity of a physical quantity is inversely proportional to the square of the distance from the source of the physical quantity. This law can be expressed as follows [Nav16]:

$$I \propto \frac{s}{r^2}\,, \tag{2.17}$$



(a) $k = 1$ **(b)** $k = 2$ **(c)** $k = 3$ **(d)** $k = 4$ **(e)** $k = 5$



(f) $k = 6$ **(g)** $k = 7$ **(h)** $k = 8$ **(i)** $k = 9$

Figure 2.7: Influence of the Spherical Harmonics coefficients. The images above show the areas that are influenced by the different SH coefficients l_k. Blue denotes negative values and green positive contributions.

where I is the intensity, s is the *strength* of the source, which in the case of light could be characterised by power in watts, and r denotes the distance to the source. An illustration of the *inverse square law* is shown in Figure 2.6. Here we can see that the same light intensity is spread over four times the area when the distance to the source doubles.

This law applies to any point source which spreads its influence equally in all directions without a limit to its range. Therefore, the near point light source model should also obey the *inverse square law*. In order to achieve this, we additionally use the simple *light attenuation term* $1/r^2$ in our Photometric Stereo model with a near point light source as also done in earlier works, e.g. [PF05], [JMBB16].

2.3.3 General Lighting with Spherical Harmonics

In the case of scenes with complex lighting, that include different point or area lights, usually an integration over the upper hemisphere has to be done for each pixel. In contrast to that, the general lighting approach based on *spherical harmonics (SH)* uses a simple irradiance formula for general lighting distribution where the irradiance environment map is approximated analyticly as presented in [RH01]. Further, Ramamoorthi and Hanrahan [RH01] state that the Lambertian BRDF is well approximated using only

the first nine spherical harmonics. That is the case, because the irradiance is insensitive to high frequencies in the lighting.

When SH are used to represent general lighting, the irradiance i at the image point $\mathbf{x}$ is described as

$$i(\mathbf{x}) = \rho(\mathbf{M}) \sum_{k=1}^{n} l_k H_k(\mathbf{N}) \ , \tag{2.18}$$

where $\rho(\mathbf{M})$ is the albedo at the surface point, n denotes the number of SH frequency bands considered, and l_k are SH coefficients for the lighting at $\mathbf{M}$. As mentioned above, in our model we need $n = 9$ SH frequency bands for Lambertian reflectance. The influenced object areas by these coefficients is shown in Figure 2.7. $H(\mathbf{N})$ stands for the spherical harmonics basis function that depends on the unit normal vector $\mathbf{N} = (N_x, N_y, N_z)$ of the surface point to compute the following values according to [WZN+14]:

$$\begin{aligned} &H_0 = 1.0, H_1 = Ny, H_2 = N_z, H_3 = N_x, H_4 = N_x N_y, H_5 = N_y N_z, \\ &H_6 = -N_x N_x - N_y N_y, H_7 = N_z N_x, H_8 = N_x N_x - N_y N_y \ . \end{aligned} \tag{2.19}$$

2.4 Variational Calculus

Since we present a variational PS approach, we cover the calculus of variations in this section. This allows us to frame our model as an energy functional which must be minimised to solve the photometric stereo problem.

One-Dimensional Case. Equation (2.20) shows such a functional for the simple one-dimensional case. It maps an input function to an output value, like an ordinary function is mapping an input value to an output value. To put it simply, a functional is a function of functions. What we need now is a function $u(x)$ so that the energy functional $E(u)$ gets minimised. Also, $u(x)$ shall be continuously differentiable on the interval $[a, b] \in \mathbb{R}$.

$$E(u) = \int_a^b \mathbf{F}(x, u(x), u_x(x)) \, \mathrm{d}x \tag{2.20}$$

Here, $\mathbf{F}$ is called the Lagrange function that is continuous and continuously differentiable with respect to $u(x)$ and $u_x(x)$.

In order to find a minimiser, a similar strategy is used as for ordinary functions: we need to derive necessary conditions. These necessary conditions are called Euler-Lagrange equations. They state that the first derivative of the energy functional, which is called first variation, must vanish. This is a necessary condition for the existence of an extremum. Thus, in the one-dimensional case with first-order derivatives all the local minimisers $u(x)$ have to satisfy the following Euler-Lagrange equation:

$$0 = \mathbf{F}_u(x, u(x), u_x(x)) - \frac{\mathrm{d}}{\mathrm{d}x}\mathbf{F}_{u_x}(x, u(x), u_x(x)) \tag{2.21}$$

with the natural boundary conditions

$$\begin{aligned} \mathbf{F}_{u_x}(a, u(a), u_x(a)) &= 0 \\ \mathbf{F}_{u_x}(b, u(b), u_x(b)) &= 0 \ . \end{aligned} \tag{2.22}$$

Two-Dimensional Case. Since the input function in our model originates from a 2D image, we also need the energy functional for the two-dimensional case and the corresponding Euler-Lagrange equations. But this is a straightforward extension, now we integrate over a given region Ω:

$$E(u) = \int_\Omega \mathbf{F}(x, y, u(x, y), u_x(x, y), u_y(x, y))\,\mathrm{d}x\mathrm{d}y \ . \tag{2.23}$$

The partial derivations in both dimensions lead us to the Euler-Lagrange equation

$$0 = \mathbf{F}_u - \frac{\partial}{\partial x}\mathbf{F}_{u_x} - \frac{\partial}{\partial y}\mathbf{F}_{u_y} \tag{2.24}$$

with the natural boundary conditions

$$0 = \mathbf{n}^\top \begin{pmatrix} \mathbf{F}_{u_x} \\ \mathbf{F}_{u_y} \end{pmatrix} \ , \tag{2.25}$$

where $\mathbf{n}$ is the outer normal vector to the boundary $\partial\Omega$.

Second-Order Derivatives. The energy functional can also depend on higher-order derivatives. In fact, in our PS model we use a second-order regulariser and therefore need to further extend the 2D energy functional:

$$E(u) = \int_\Omega \mathbf{F}(x, y, u(x,y), u_x(x,y), u_y(x,y), u_{xx}(x,y), u_{xy}(x,y), u_{yx}(x,y), u_{yy}(x,y))\,\mathrm{d}x\mathrm{d}y \tag{2.26}$$

When the Lagrange function $\mathbf{F}$ contains second-order derivatives the resulting Euler-Lagrange equation reads as follows:

$$0 = \mathbf{F}_u - \frac{\partial}{\partial x}\mathbf{F}_{u_x} - \frac{\partial}{\partial y}\mathbf{F}_{u_y} + \frac{\partial^2}{\partial xx}\mathbf{F}_{u_{xx}} + \frac{\partial^2}{\partial x \partial y}\mathbf{F}_{u_{xy}} + \frac{\partial^2}{\partial y \partial x}\mathbf{F}_{u_{yx}} + \frac{\partial^2}{\partial yy}\mathbf{F}_{u_{yy}} \tag{2.27}$$

with the natural boundary conditions

$$\begin{aligned} 0 &= \mathbf{n}^\top \begin{pmatrix} \mathbf{F}_{u_x} - \frac{\partial}{\partial x}\mathbf{F}_{u_{xx}} - \frac{\partial}{\partial y}\mathbf{F}_{u_{xy}} \\ \mathbf{F}_{u_y} - \frac{\partial}{\partial x}\mathbf{F}_{u_{yx}} - \frac{\partial}{\partial y}\mathbf{F}_{u_{yy}} \end{pmatrix} \\ 0 &= \mathbf{n}^\top \begin{pmatrix} \mathbf{F}_{u_{xx}} \\ \mathbf{F}_{u_{xy}} \end{pmatrix}, \quad 0 = \mathbf{n}^\top \begin{pmatrix} \mathbf{F}_{u_{yx}} \\ \mathbf{F}_{u_{yy}} \end{pmatrix}, \end{aligned} \tag{2.28}$$

where $\mathbf{n}$ is again the outer normal vector to the boundary $\partial\Omega$.

3 Model

The following chapter covers how the variational PS approach is modelled. We start with the parameterisation, then we give a brief introduction to the variational SfS approach [MJBB15] since it serves as basis for our variational model. Its introduction follows naturally. Afterwards, we address the efficient minimisation of the resulting energy functional. At the end of this chapter we present an extension to our PS model that is based on the use of image ratios.

3.1 Surface Parameterisation

In the following section, our goal is to present a parameterisation of the surface, so that a 3D surface point is parameterised via a single unknown z for every pixel. Before we can start with the parameterisation, we need to look again at the projection by the perspective camera model described in Section 2.1.

For our model we concatenate the intrinsic matrix K and the projection matrix P, further we assume that the sensor is perpendicular to the optical axis, thus the skew $s = 0$. This leads to a simplified intrinsic matrix that we denote as K_s:

$$K_s = KP = \begin{pmatrix} 1/h_x & s/\mathtt{f} & o_x \\ 0 & 1/h_y & o_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathtt{f} & 0 & 0 & 0 \\ 0 & \mathtt{f} & 0 & 0 \end{pmatrix} = \begin{pmatrix} \mathtt{f}/h_x & 0 & o_x \\ 0 & \mathtt{f}/h_y & o_y \\ 0 & 0 & 1 \end{pmatrix} . \tag{3.1}$$

We are interrested in presenting our model in the so called *Cartesian depth parameterisation*, this means we parameterise the depth z by backprojecting along the optical axis. Therefore the surface $\mathcal{S} : \mathbf{\Omega_x} \to \mathbb{R}^3$ can be parameterised as

$$\begin{aligned} \mathcal{S}(\mathbf{x}, z(\mathbf{x})) &= K_s^{-1}\tilde{\mathbf{x}} \cdot z(\mathbf{x}) \\ &= \begin{pmatrix} h_x/\mathtt{f} & 0 & -o_x h_x/\mathtt{f} \\ 0 & h_y/\mathtt{f} & -o_y h_y/\mathtt{f} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} z(\mathbf{x}) = \begin{pmatrix} (x - o_x)h_x/\mathtt{f} \\ (y - o_y)h_y/\mathtt{f} \\ 1 \end{pmatrix} z(\mathbf{x}) , \end{aligned} \tag{3.2}$$

as described by Schroers et al. [SHW15]. Here $\mathbf{x} := (x, y)^\top \in \mathbf{\Omega_x}$ is the position of the pixel in the image plane $\mathbf{\Omega_x} \subset \mathbb{R}^2$, $\tilde{\mathbf{x}}$ is the corresponding point in homogeneous coordinates, and K_s^{-1} is the inverse of the simplified intrinsic matrix.

3.2 Surface Normal

Before we continue with the parameterisation of our model components we briefly define a mathematical operator based on the well-known operator *nabla* represented by ∇ to to achieve a more compact notation in the following sections.

In vector calculus ∇ is a vector differential operator that contains the partial derivative operators

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right) . \tag{3.3}$$

Accordingly, we introduce a second order *nabla* defined as

$$\nabla_2 = \left(\frac{\partial^2}{\partial x^2}, \frac{\partial^2}{\partial x \partial y}, \frac{\partial^2}{\partial y \partial x}, \frac{\partial^2}{\partial y^2} \right) . \tag{3.4}$$

Since we need the surface normal to use the Lambertian reflectance model as explained in Section 2.2, the next step is to parameterise the surface normal in terms of the Cartesian depth. To accomplish this we first have to compute the partial derivatives of the surface in Equation (3.2) in x-direction

$$\mathcal{S}_x(\mathbf{x}, z(\mathbf{x})) = K_s^{-1} \begin{pmatrix} z(\mathbf{x}) + z(\mathbf{x})_x x \\ z(\mathbf{x})_x y \\ z(\mathbf{x})_x \end{pmatrix} = \begin{pmatrix} \frac{h_x}{\mathtt{f}} (z(\mathbf{x}) + z(\mathbf{x})_x (x - o_x)) \\ z(\mathbf{x})_x \frac{h_y}{\mathtt{f}} (y - o_y) \\ z(\mathbf{x})_x \end{pmatrix} \tag{3.5}$$

and y-direction

$$\mathcal{S}_y(\mathbf{x}, z(\mathbf{x})) = K_s^{-1} \begin{pmatrix} z(\mathbf{x})_y x \\ z(\mathbf{x}) + z(\mathbf{x})_y y \\ z(\mathbf{x})_y \end{pmatrix} = \begin{pmatrix} z(\mathbf{x})_y \frac{h_x}{\mathtt{f}} (x - o_x) \\ \frac{h_y}{\mathtt{f}} (z(\mathbf{x}) + z(\mathbf{x})_y (y - o_y)) \\ z(\mathbf{x})_y \end{pmatrix} . \tag{3.6}$$

The normalised cross-product of these partial derivatives then yields the normal vector of the surface

Illumination Model	Brightness Equation
Distant Light	$I_{DL}(\mathbf{x}) = \rho(\mathbf{x})\mathbf{N}(\mathbf{x}) \cdot \mathbf{L}(\mathbf{x})$
Near Point Light	$I_{NL}(\mathbf{x}) = \rho(\mathbf{x})\frac{1}{r(\mathbf{x})^2}\left(\mathbf{N}(\mathbf{x}) \cdot \mathbf{L}(\mathbf{x})\right)$
Spherical Harmonics	$I_{SH}(\mathbf{x}) = \rho(\mathbf{x})\sum_{k=1}^{9} l_k H_k(\mathbf{N}(\mathbf{x}))$

Table 3.1: The different brightness equations used in this work.

$$\mathbf{N}(\mathbf{x}, z(\mathbf{x})) = \frac{\mathcal{S}_x(\mathbf{x}, z(\mathbf{x})) \times \mathcal{S}_y(\mathbf{x}, z(\mathbf{x}))}{|\mathcal{S}_x(\mathbf{x}, z(\mathbf{x})) \times \mathcal{S}_y(\mathbf{x}, z(\mathbf{x}))|}\,, \tag{3.7}$$

where

$$\begin{aligned}\mathcal{S}_x(\mathbf{x}, z(\mathbf{x})) \times \mathcal{S}_y(\mathbf{x}, z(\mathbf{x})) &= \begin{bmatrix} s_{x2}s_{y3} - s_{x3}s_{y2} \\ s_{x3}s_{y1} - s_{x1}s_{y3} \\ s_{x1}s_{y2} - s_{x2}s_{y1} \end{bmatrix} \\ &= \frac{z(\mathbf{x})}{\mathtt{f}} \begin{bmatrix} -z(\mathbf{x})_x h_y \\ -z(\mathbf{x})_y h_x \\ \frac{h_x h_y}{\mathtt{f}}(z(\mathbf{x}) + \nabla z(\mathbf{x})^\top(\mathbf{x} - \mathbf{o})) \end{bmatrix}.\end{aligned} \tag{3.8}$$

and

$$\begin{aligned}|\mathcal{S}_x(\mathbf{x}, z(\mathbf{x})) \times \mathcal{S}_y(\mathbf{x}, z(\mathbf{x}))| \;=\; &\left(\frac{z(\mathbf{x})^2 z(\mathbf{x})_x^2 h_y^2 + z(\mathbf{x})^2 z(\mathbf{x})_y^2 h_x^2}{\mathtt{f}^2}\right. \\ &\left. + \frac{z(\mathbf{x})^2 h_x^2 h_y^2 (z(\mathbf{x}) + \nabla z(\mathbf{x})^\top(\mathbf{x} - \mathbf{o}))^2}{\mathtt{f}^4}\right)^{\frac{1}{2}}.\end{aligned} \tag{3.9}$$

3.3 Brightness Equation

Now that we have seen how to parameterise the surface and compute the surface normal with respect to the Cartesian depth, we can move on to the brightness equation. In this work, we present our model with three different brightness equations, which are shown in Table 3.1, and compare the corresponding outcome. Each illumination model presented in Section 2.3 leads to a different brightness equation that is later used in the data term of our variational model as follows:

$$I_{cam}(\mathbf{x}) - I_{model}(\mathbf{x}) = 0 \ , \tag{3.10}$$

where I_{cam} is the pixel value of the recorded image and I_{model} is the corresponding computed brightness using one of the presented illumination models, i.e. the corresponding brightness equation given in Table 3.1.

In Table 3.1 $\mathbf{N}$ stands for the normalised surface normal vector, the normalised light direction vector is denoted by $\mathbf{L}$, and r is the distance from the light source to the surface. As stated in Section 2.3, the general lighting model based on spherical harmonics uses the specific coefficients l_k and the basis functions $H_k(\mathbf{N}(\mathbf{x}))$. In this work we assume the light direction vector for the distant light source model, the light position for the near point light model, and the SH coefficients for the general lighting model to be given. Additionally, for all illumination models we assume a constant albedo $\rho = 1$.

3.4 Variational Perspective Shape from Shading Model

We use the work of Maurer et al. [MJBB15] as a foundation in this thesis and extend their *Shape from Shading* model. Thus, we regard this variational SfS model with Cartesian depth parameterisation in this section.

The key assumptions here are that a point light source is located in the optical center of the perspective camera, the reflectance properties of the surface are described through the Lambertian reflectance model, and the albedo is uniformly fixed to one. So, the main difference to our approach is that in the PS model with near point lights we have multiple images taken with a single fixed camera under different lightning conditions.

Surface parameterisation. Since their parameterisation of the surface is slightly different to ours, we briefly regard it. Our parameterisation shown in Section 3.1 uses the pixel coordinate system where the basis SfS model is parameterised in terms of image coordinates. The final Cartesian depth parameterisation of the surface is shown in Equation (3.11).

$$\mathcal{S}_{SfS}(\mathbf{m}, z(\mathbf{m})) = Q(\mathbf{m}) \left(\frac{z(\mathbf{m})}{\mathtt{f} Q(\mathbf{m})} \right) \begin{bmatrix} x \\ y \\ -\mathtt{f} \end{bmatrix} = \begin{bmatrix} \frac{z(\mathbf{m})x}{\mathtt{f}} \\ \frac{z(\mathbf{m})y}{\mathtt{f}} \\ -z(\mathbf{m}) \end{bmatrix} . \tag{3.11}$$

with the conversion factor $Q(\mathbf{m}) = \mathtt{f}/\sqrt{|\mathbf{m}|^2 + \mathtt{f}^2}$ between the radial and Cartesian depth. Then the surface normal is computed using the partial derivatives of the surface

as shown in Section 3.2. Note, that the z-axis of the coordinate system used in this model points in the opposite direction compared to ours. Therefore, the z-component in Equation (3.11) is negative.

Brightness Equation. Now that we have seen how Maurer et al. parameterised the surface with respect to the Cartesian depth, we can move on to the derivation of the brightness equation. They assume a Lambertian reflectance model and a quadratic light attenuation term that follows the inverse square law. From this results the brightness equation that looks exactly like our brightness equation for near point light sources given in Table 3.1:

$$I(\mathbf{m}) = \frac{1}{r(\mathbf{m})^2} \left(\mathbf{N}(\mathbf{m}) \cdot \mathbf{L}(\mathbf{m})\right) \ . \tag{3.12}$$

Light Direction. What now remains left is the computation of the light direction vector $\mathbf{L}$. We already mentioned the model assumption that the point light source is located in the camera center which coincides with the origin of the coordinate system. Thus, the direction of light rays and the direction of optical rays have the exact opposite direction. This means the light direction can simply be derived from equation (3.11)

$$\mathbf{L}(\mathbf{m}) = \frac{1}{\sqrt{|\mathbf{m}|^2 + \mathtt{f}^2}} \begin{bmatrix} -x \\ -y \\ \mathtt{f} \end{bmatrix} = \frac{Q}{\mathtt{f}} \begin{bmatrix} -x \\ -y \\ \mathtt{f} \end{bmatrix} \ . \tag{3.13}$$

Finally, this leads to the constraint for perspective SfS in the Lambertian case:

$$I(\mathbf{m}) - \frac{Q(\mathbf{m})^3}{zW(\mathbf{m}, z, \nabla z)} = 0 \ . \tag{3.14}$$

where

$$W(\mathbf{m}, z, \nabla z) = \sqrt{\mathtt{f}^2 |\nabla z(\mathbf{m})|^2 + [(\nabla z \cdot \mathbf{m}) + z(\mathbf{m})]^2} \ . \tag{3.15}$$

Variational Model. To complete this section we now discuss how Maurer et al. integrated their model into a variational framework with smoothness term. They use the constraint for perspective SfS shown in (3.14) in a quadratic error term as data term and complement it with a suitable second-order regulariser. The Cartesian depth z is then computed as minimiser of the resulting energy functional

$$E_{SfS}(z) = \int_{\Omega_{\mathbf{m}}} c(\mathbf{m}) \underbrace{D(\mathbf{m}, z, \nabla z)^2}_{\text{data term}} + \alpha \underbrace{\Psi\left(S(\text{Hess}(z)(\mathbf{m}))^2\right)}_{\text{smoothness term}} \mathrm{d}\mathbf{m} \,. \tag{3.16}$$

The confidence function $c(\mathbf{m}) : \mathbf{m} \in \Omega_{\mathbf{m}} \subset \mathbb{R}^2 \to \{0, 1\}$ in equation (3.16) can be used to exclude unreliable image regions from the computation. However, these image regions have to be identified as such a priori, for example by texture detection or background segmentation algorithms. This can be very useful for solving tasks that contain real-world images which often suffer from noise or missing data. The parameter $\alpha \in \mathbb{R}^+$ in the energy functional denotes a regularisation parameter with which the degree of smoothness of the surface can be steered. In the energy functional 3.16 D in the data term and S in the smoothness term $\Psi(S^2)$ are defined as

$$D(\mathbf{m}, z, \nabla z) = \left(I_{cam}(\mathbf{m}) - \frac{Q(\mathbf{m})^3}{zW(\mathbf{m}, z, \nabla z)}\right) \,, \tag{3.17}$$

$$S(\text{Hess}(z)(\mathbf{m})) = \sqrt{z_{xx}(\mathbf{m})^2 + 2z_{xy}(\mathbf{m})^2 + z_{yy}(\mathbf{m})^2} \,. \tag{3.18}$$

Here Ψ denotes the Charbonnier function [CBAB97]

$$\Psi(S^2) = 2\lambda^2\sqrt{1 + \frac{S^2}{\lambda^2}} \tag{3.19}$$

with the contrast parameter λ. Furthermore, it is to mention that first order smoothness terms are not advisable for SfS from a theoretical viewpoint since the data term already contains first order derivatives. Hence, a edge-preserving second-order smoothness term based on the Frobenius norm of the Hessian is used here.

The minimisation of the energy functional is explained in detail in the following section, where we consider our PS model.

3.5 Variational Photometric Stereo Model

We also use a quadratic error term as data term and complement it with a suitable second-order regulariser. The Cartesian depth z is then computed as minimiser of the resulting energy functional

$$E(z) = \int_{\Omega_{\mathbf{x}}} \underbrace{D(\mathbf{x}, z, \nabla z)^2}_{\text{data term}} + \alpha \underbrace{\Psi\left(S(\text{Hess}(z)(\mathbf{x}))^2\right)}_{\text{smoothness term}} \mathrm{d}\mathbf{x} \,. \tag{3.20}$$

Data Term. Since we have multiple input images in Photometric Stereo where each corresponds to a different incoming light direction, we have to add up the brightness equations (2.16) for multiple light direction vectors.

Finally, for m recorded input images $I_i \subset \Omega_{\mathbf{x}}$ and therefore m corresponding light directions $\mathbf{L}_i$ the constraint for perspective Photometric Stereo with a Lambertian reflectance model and the distant light source model reads

$$D_{DL}(\mathbf{x}, z, \nabla z) = \sum_{i=1}^{m} \left(I_i(\mathbf{x}) - (\mathbf{N}(\mathbf{x}) \cdot \mathbf{L}_i(\mathbf{x}))\right) \;=\; 0 \,. \tag{3.21}$$

Respectively, for the near point light source model the PS constraint is given by

$$D_{NL}(\mathbf{x}, z, \nabla z) = \sum_{i=1}^{m} \left(I_i(\mathbf{x}) - \frac{1}{r_i(\mathbf{x})^2} (\mathbf{N}(\mathbf{x}) \cdot \mathbf{L}_i(\mathbf{x}))\right) \;=\; 0 \,, \tag{3.22}$$

and for the general lighting model based on spherical harmonics we obtain

$$D_{SH}(\mathbf{x}, z, \nabla z) = \sum_{i=1}^{m} \left(I_i(\mathbf{x}) - \sum_{k=1}^{9} l_k H_k(\mathbf{N}(\mathbf{x}))\right) \;=\; 0 \,. \tag{3.23}$$

In the energy functional (3.20) D stands for one of the constraints D_{NL}, D_{DL} and D_{SH}, depending on the chosen brightness equation.

Smoothness Term. Let us now consider the smoothness term $\Psi(S^2)$ where S is defined as in the variational SfS model

$$S(\text{Hess}(z)(\mathbf{x})) = \sqrt{z_{xx}(\mathbf{x})^2 + 2z_{xy}(\mathbf{x})^2 + z_{yy}(\mathbf{x})^2}\ . \tag{3.24}$$

Again, Ψ denotes the Charbonnier function

$$\Psi(S^2) = 2\lambda^2\sqrt{1 + \frac{S^2}{\lambda^2}} \tag{3.25}$$

with the contrast parameter λ. As in the SfS model, $\alpha \in \mathbb{R}^+$ denotes a regularisation parameter with which the degree of smoothness of the surface can be controlled.

3.5.1 Minimisation

In this section we will cover the minimisation of the energy functional (3.20). In order to compute the minimisation efficiently we use the linearisation technique proposed by Maurer et. al in [MJBB15] called the *Hyperbolic Warping Scheme* as stated in [Ju17]. Therefore, we introduce an approximation for the partial derivatives in the data term using the upwind scheme presented in [RT92]. In order to solve the resulting Euler-Lagrange equation we derive the same numerical scheme as Maurer et al. in [MJBB15] which is based on the work of Brox et. al [BBPW04]. This technique consists of two fixed point iterations, an incremental computation, and a coarse-to-fine strategy. With this technique we can obtain an efficient computation and we can overcome the non-convexity of the energy functional induced by the dependency on ∇z in the data term [Ju17].

Euler-Lagrange Equation. Firstly, let us have a look at the resulting Euler-Lagrange equation for our energy functional in Equation (3.20) according to the variational calculus 2.4. For our model we have to consider the 2D case with second-order derivatives:

$$\begin{aligned} 0 =& [D^2 + \alpha\Psi(S^2)]_{z(\mathbf{x})} - \frac{\partial}{\partial x}[D^2 + \alpha\Psi(S^2)]_{z_x(\mathbf{x})} - \frac{\partial}{\partial y}[D^2 + \alpha\Psi(S^2)]_{z_y(\mathbf{x})} \\ &+ \frac{\partial^2}{\partial x^2}[D^2 + \alpha\Psi(S^2)]_{z_{xx}(\mathbf{x})} + 2\frac{\partial^2}{\partial x \partial y}[D^2 + \alpha\Psi(S^2)]_{z_{xy}(\mathbf{x})} + \frac{\partial^2}{\partial y^2}[D^2 + \alpha\Psi(S^2)]_{z_{yy}(\mathbf{x})}\ , \end{aligned} \tag{3.26}$$

with the natural boundary conditions on $\partial\Omega$

$$\begin{aligned} 0 &= \mathbf{n}^\top \begin{pmatrix} [D^2 + \alpha\Psi(S^2)]_{z_x(\mathbf{x})} - \frac{\partial}{\partial x}[D^2 + \alpha\Psi(S^2)]_{z_{xx}(\mathbf{x})} - \frac{\partial}{\partial y}[D^2 + \alpha\Psi(S^2)]_{z_{xy}(\mathbf{x})} \\ [D^2 + \alpha\Psi(S^2)]_{z_y(\mathbf{x})} - \frac{\partial}{\partial x}[D^2 + \alpha\Psi(S^2)]_{z_{yx}(\mathbf{x})} - \frac{\partial}{\partial y}[D^2 + \alpha\Psi(S^2)]_{z_{yy}(\mathbf{x})} \end{pmatrix} \\ 0 &= \mathbf{n}^\top \begin{pmatrix} [D^2 + \alpha\Psi(S^2)]_{z_{xx}(\mathbf{x})} \\ [D^2 + \alpha\Psi(S^2)]_{z_{xy}(\mathbf{x})} \end{pmatrix}, \quad 0 = \mathbf{n}^\top \begin{pmatrix} [D^2 + \alpha\Psi(S^2)]_{z_{yx}(\mathbf{x})} \\ [D^2 + \alpha\Psi(S^2)]_{z_{yy}(\mathbf{x})} \end{pmatrix}. \end{aligned} \tag{3.27}$$

Since D in the data term depends on z and ∇z and S in the smoothness terms depends on the second-order derivatives of z, we can further simplify the Euler-Lagrange equation by dropping the zero contribution terms. This leads to:

$$\begin{aligned} 0 =& [D^2]_{z(\mathbf{x})} - \frac{\partial}{\partial x}[D^2]_{z_x(\mathbf{x})} - \frac{\partial}{\partial y}[D^2]_{z_y(\mathbf{x})} \\ &+ \frac{\partial^2}{\partial x^2}[\alpha\Psi(S^2)]_{z_{xx}(\mathbf{x})} + 2\frac{\partial^2}{\partial x \partial y}[\alpha\Psi(S^2)]_{z_{xy}(\mathbf{x})} + \frac{\partial^2}{\partial y^2}[\alpha\Psi(S^2)]_{z_{yy}(\mathbf{x})}. \end{aligned} \tag{3.28}$$

with the simplified boundary conditions

$$\begin{aligned} 0 &= \mathbf{n}^\top \begin{pmatrix} [D^2]_{z_x(\mathbf{x})} - \frac{\partial}{\partial x}[\alpha\Psi(S^2)]_{z_{xx}(\mathbf{x})} - \frac{\partial}{\partial y}[\alpha\Psi(S^2)]_{z_{xy}(\mathbf{x})} \\ [D^2]_{z_y(\mathbf{x})} - \frac{\partial}{\partial x}[\alpha\Psi(S^2)]_{z_{yx}(\mathbf{x})} - \frac{\partial}{\partial y}[\alpha\Psi(S^2)]_{z_{yy}(\mathbf{x})} \end{pmatrix} \\ 0 &= \mathbf{n}^\top \begin{pmatrix} [\alpha\Psi(S^2)]_{z_{xx}(\mathbf{x})} \\ [\alpha\Psi(S^2)]_{z_{xy}(\mathbf{x})} \end{pmatrix}, \quad 0 = \mathbf{n}^\top \begin{pmatrix} [\alpha\Psi(S^2)]_{z_{yx}(\mathbf{x})} \\ [\alpha\Psi(S^2)]_{z_{yy}(\mathbf{x})} \end{pmatrix}. \end{aligned} \tag{3.29}$$

By computing the derivatives of the smoothness term and plugging the results into Equation (3.28) we obtain the corresponding Euler-Lagrange equation

$$\begin{aligned} 0 =& [D^2]_{z(\mathbf{x})} - \frac{\partial}{\partial x}[D^2]_{z_x(\mathbf{x})} - \frac{\partial}{\partial y}[D^2]_{z_y(\mathbf{x})} \\ &+ 2\alpha \left(\frac{\partial^2}{\partial x^2}[\Psi'(S^2) z_{xx}(\mathbf{x})] + 2\frac{\partial^2}{\partial x \partial y}[\Psi'(S^2) z_{xy}(\mathbf{x})] + \frac{\partial^2}{\partial y^2}[\Psi'(S^2) z_{yy}(\mathbf{x})] \right), \end{aligned} \tag{3.30}$$

where the derivative of the penaliser function $\Psi(S^2)$ is given by

$$\Psi'(S^2) = \frac{1}{\sqrt{1 + S^2/\lambda^2}} \, . \tag{3.31}$$

With the use of the ∇_2 operator defined in Equation (3.4) we can bring Equation (3.30) in a more compact notation:

$$0 = [D^2]_{z(\mathbf{x})} - \frac{\partial}{\partial x}[D^2]_{z_x(\mathbf{x})} - \frac{\partial}{\partial y}[D^2]_{z_y(\mathbf{x})} + 2\alpha \left(\nabla_2^\top [\Psi'(S^2) \nabla_2^\top z(\mathbf{x})] \right) \, , \tag{3.32}$$

Approximation of the Spatial Derivatives. Now we discuss the approximation of z_x and z_y in the data term using the upwind scheme. Standard finite differences schemes, such as central differences are not appropriate for this model because of the hyperbolic nature of the data term in our energy functional [MJBB15]. With that said, the approximations read

$$\tilde{z}_x = \max(\mathcal{D}^{-x} z, -\mathcal{D}^{+x} z, 0) \, , \tag{3.33}$$

where

$$\mathcal{D}^{-x} z = \frac{z(x,y) - z(x - h_x, y)}{h_x} \, , \; \mathcal{D}^{+x} z = \frac{z(x + h_x, y) - z(x,y)}{h_x} \, . \tag{3.34}$$

Since the sign matters in this case, we have to restore the correct sign as follows:

$$z_x \approx \begin{cases} -\tilde{z}_x & \text{if} \quad \tilde{z}_x = -\mathcal{D}^{+x} z \, , \\ \tilde{z}_x & \text{else} \, . \end{cases} \tag{3.35}$$

And respectively for z_y:

$$\tilde{z}_y = \max(\mathcal{D}^{-y} z, -\mathcal{D}^{+y} z, 0) \, , \tag{3.36}$$

where

$$\mathcal{D}^{-y} z = \frac{z(x,y) - z(x, y - h_y)}{h_y} \, , \; \mathcal{D}^{+y} z = \frac{z(x, y + h_y) - z(x,y)}{h_y} \, . \tag{3.37}$$

and the restoration of the correct sign is given by

$$z_y \approx \begin{cases} -\tilde{z}_y & \text{if} \quad \tilde{z}_y = -\mathcal{D}^{+y} z \ , \\ \tilde{z}_y & \text{else} \ . \end{cases} \tag{3.38}$$

Here h_x and h_y denote the grid spacings in x- and y-direction. Now that we have the approximations of the spatial derivatives in the data term, we are able to express D in a way that it only depends on z. To this end, we have to include values at neighbouring locations and obtain the following approximation $D(\mathbf{x})$ for $D(\mathbf{x}, z(\mathbf{x}), \nabla z(\mathbf{x}))$

$$D(\mathbf{x}) := D(x, y, z(x, y), z(x - h_x, y), z(x + h_x, y), z(x, y - h_y), z(x, y + h_y)) \ , \tag{3.39}$$

Euler-Lagrange Equation with Approximation of Derivatives. In what follows, we present the Euler-Lagrange equation that any stationary point of the approximated energy functional has to fulfil according to the calculus of variations. Let us have a look how the contributions from the data term E_D change with the use of the aforementioned approximations for the spatial derivatives.

Since the first order derivatives in the Euler-Lagrange equation are no longer present in the data term due to the approximations for the spatial derivatives, the terms related to the gradient $\frac{\partial}{\partial x}[D^2]_{z_x(\mathbf{x})}$ and $\frac{\partial}{\partial y}[D^2]_{z_y(\mathbf{x})}$ are substituted by the corresponding terms that contain the neighbourhood for the upwind scheme. Hence, we obtain:

$$E_D = \sum_{\mathbf{h} \in \mathcal{H}} [D^2(\mathbf{x} + \mathbf{h})]_{z(\mathbf{x})} = \sum_{\mathbf{h} \in \mathcal{H}} 2\, D(\mathbf{x} + \mathbf{h}) [D(\mathbf{x} + \mathbf{h})]_{z(\mathbf{x})} \ . \tag{3.40}$$

where $\mathbf{h}$ stands for the elements in the set

$$\mathcal{H} = \{-\mathbf{h}_y, -\mathbf{h}_x, \mathbf{0}, +\mathbf{h}_x, +\mathbf{h}_y\} \ , \tag{3.41}$$

which is the neighbourhood used by the upwind scheme for the approximation as explained previously. $\mathbf{h}_x = (h_x, 0)^\top$, $\mathbf{h}_y = (0, h_y)^\top$ are the grid spacings in x- and y-direction.

Now can put (3.40) into Equation (3.32) and leave the common factor 2 out of the equation, this leads to the linearised Euler-Lagrange equation

$$0 = \left(\sum_{\mathbf{h} \in \mathcal{H}} D(\mathbf{x} + \mathbf{h}) [D(\mathbf{x} + \mathbf{h})]_{z(\mathbf{x})} \right) + \alpha \left(\nabla_2^\top (\Psi'(S^2(\mathbf{x})) \nabla_2 z(\mathbf{x})) \right) \ . \tag{3.42}$$

Linearisation. Now we follow the work of Maurer et al. [MJBB15] and introduce a numerical scheme to solve Equation (3.42). According to [MJBB15] we start with a first fixed point iteration on z with the iteration index k. In detail, a semi-implicit scheme and a implicit scheme is used where we apply the semi-implicit scheme to terms that a related to the data term and the implicit scheme to terms that are derived from the smoothness term. This leads to the following expression:

$$0 = \sum_{\mathbf{h}\in\mathcal{H}} D^{k+1}(\mathbf{x}+\mathbf{h})[D^k(\mathbf{x}+\mathbf{h})]_{z^k(\mathbf{x})} + \alpha\left(\nabla_2^\top(\Psi'((S^{k+1}(\mathbf{x}))^2)\nabla_2 z^{k+1}(\mathbf{x}))\right) . \tag{3.43}$$

In Equation (3.43) the terms containing the z at time step $k+1$ are still non-linear. Namely, the terms related to the data term containing D^{k+1} and the argument ofΨ' in the smoothness term containing S^{k+1}. To resolve the non-linearity of the data term, Maurer et al. use a first order Taylor expansion to linearise D^{k+1} around D^k and introduce the incremental computation which leads to

$$D^{k+1}(\mathbf{x}+\mathbf{h}) = D^k(\mathbf{x}+\mathbf{h}) + \sum_{\mathbf{h}'\in\mathcal{H}} [D^k(\mathbf{x}+\mathbf{h})]_{z^k(\mathbf{x}+\mathbf{h}+\mathbf{h}')} dz^k(\mathbf{x}+\mathbf{h}+\mathbf{h}') . \tag{3.44}$$

Here dz^k describes the unknown increment $dz^k = z^{k+1} - z^k$ that is computed in each iteration. This incremental computation, where only the increment dz^k is computed was introduced by Brox et al. in [BBPW04]. By plugging these expressions from the first order Taylor expansion into the part corresponding to the data term, we obtain the new contributions from the data term:

$$E_D = \sum_{\mathbf{h}\in\mathcal{H}} \left(D^k(\mathbf{x}+\mathbf{h}) + \sum_{\mathbf{h}'\in\mathcal{H}} [D^k(\mathbf{x}+\mathbf{h})]_{z^k(\mathbf{x}+\mathbf{h}+\mathbf{h}')} dz^k(\mathbf{x}+\mathbf{h}+\mathbf{h}') \right) [D^k(\mathbf{x}+\mathbf{h})]_{z^k(\mathbf{x})} . \tag{3.45}$$

Let us now regard the contributions from the smoothness term and bring it in incremental formulation. Hence, the part of the smoothness term in Equation (3.43) looks as follows:

$$E_S = \alpha\left(\nabla_2^\top(\Psi'((S^{k+1}(\mathbf{x}))^2)\nabla_2(z^k(\mathbf{x}) + dz^k(\mathbf{x})))\right) , \tag{3.46}$$

where

$$\begin{aligned}(S^{k+1}(\mathbf{x}))^2 &= (z^k_{xx}(\mathbf{x}) + dz^k_{xx}(\mathbf{x}))^2 + 2(z^k_{xy}(\mathbf{x}) + dz^k_{xy}(\mathbf{x}))^2 + (z^k_{yy}(\mathbf{x}) + dz^k_{yy}(\mathbf{x}))^2 \\ &= \nabla_2(z^k(\mathbf{x}) + dz^k(\mathbf{x}))^2 \ .\end{aligned} \tag{3.47}$$

As mentioned above, the last remaining nonlinearity now lies in the derivative of the penaliser function Ψ'. To overcome this nonlinearity Maurer et al. in [MJBB15] introduced a second fixed point iteration on dz with the iteration index l and the initialisation $dz^{k,0} = 0$. Then, we use the increment $dz^{k,l}$ at the iteration k, l for all arguments of the Ψ'-function and the increment $dz^{k,l+1}$ at the iteration $k, l+1$ for the remaining terms. This method is also known as the *lagged nonlinearity* method, where the basic idea of this method is to solve a nonlinear problem as a series of linear problems [Ju17]. From this follows the expression:

$$\begin{aligned}0 = &\\ &\left(\sum_{\mathbf{h}\in\mathcal{H}}\left(D^k(\mathbf{x}+\mathbf{h}) + \sum_{\mathbf{h}'\in\mathcal{H}}[D^k(\mathbf{x}+\mathbf{h})]_{z^k(\mathbf{x}+\mathbf{h}+\mathbf{h}')}dz^{k,l+1}(\mathbf{x}+\mathbf{h}+\mathbf{h}')\right)[D^k(\mathbf{x}+\mathbf{h})]_{z^k(\mathbf{x})}\right)\\ &+ \alpha\left(\nabla_2^\top(\Psi'((S^{k,l}(\mathbf{x}))^2)\nabla_2(z^{k,l+1}(\mathbf{x}) + dz^{k,l+1}(\mathbf{x})))\right) \ ,\end{aligned} \tag{3.48}$$

where

$$(S^{k,l}(\mathbf{x}))^2 = \nabla_2(z^{k,l}(\mathbf{x}) + dz^{k,l}(\mathbf{x}))^2 \ . \tag{3.49}$$

Thereby, we finally possess all the linearised contributions from the data and the smoothness term and now hold a linear system of equations with respect to $dz^{k,l+1}$.

Coarse-to-Fine Approach. In order to complete the hyperbolic warping scheme we now explain the coarse-to-fine approach. Since our energy functional is non-convex the minimisation may get stuck in local minima. To counter that we embed the first fixed point iteration into a coarse-to-fine scheme to improve the approximation of the global minimiser. Such coarse-to-fine strategies have also been used in [BBPW04] [JMBB16] [MJBB15].

We start the computation on the coarsest resolution of the input image and reconstruct the surface while refining the image successively up to the original image size. Thus, we get a computed increment dz for every resolution level where the result of the last scale of resolution is used as initialisation for the next finer scale. More precisely, the

initial $z_{m=0}$ for the original resolution can be computed by adding up the increments from each level which is described by

$$z_{m=0} = z_{m_{\max}} + \sum_{m=0}^{m_{\max}} dz_{m_{\max}-m} \ . \tag{3.50}$$

Here, $m = 0$ denotes the original resolution of the input image and $m = m_{\max}$ is the coarsest resolution level. Regarding the initialisation of the depth $z_{m_{\max}}$ on the coarsest level, we choose $z_{m_{\max}}$ to be constant which results in a plane that lies parallel to the image plane.

Now we will have a closer look on the details of the coarse-to-fine approach. In order to compute the coarser scales of the input image Maurer et al. introduced the parameter κ to specify after how many iterations k the resolution level is changed and the downsampling factor η. The downsampling parameter denotes the factor between two consecutive resolution levels and lies in the interval $(0, 1)$. This reduction of the image resolution also affects the intrinsic parameters of the camera. Therefore, we briefly show how the downsampling factor η changes the simplified intrinsic matrix given in Equation (3.1) which results in the scaled intrinsic matrix K_η for the next coarser level:

$$K_\eta = \begin{pmatrix} \eta\, \mathtt{f}/h_x & 0 & \eta\, o_x \\ 0 & \eta\, \mathtt{f}/h_y & \eta\, o_y \\ 0 & 0 & 1 \end{pmatrix} \ . \tag{3.51}$$

Keep in mind that the size of the image plane remains constant and thus the grid size increases while the number of pixels decreases at coarser resolution scales.

Additionally, to the improved quality of the surface reconstruction by avoiding local minima this coarse-to-fine scheme has also the effect that the computation is more robust with regards to the initialisation of z. In fact, Ju et al. mentioned in [JMBB16] that they could hardly observe any impact of the initialisation on the quality of the final reconstructed surface, if sufficiently many resolution levels were computed.

Computation. Finally, in order to solve for $dz^{k,l+1}$ numerically we have to bring Equation (3.48) in discrete form by considering each function dependent on $\mathbf{x}$ on a $nx \times ny$ pixel grid with the grid spacings h_x and h_y. Note that we drop the superscripts for the iteration steps in the next approximations to obtain a clearer notation.

To compute the derivatives of D with respect to z numerically, we again follow the strategy proposed in [MJBB15] by Maurer et al. They vary the current estimated depth

z by a small margin $\pm h_z$ and recompute the terms related to the data term D after that. This allows to estimate the derivatives of D with respect to z using standard central differences at each pixel (i, j) as follows:

$$[D_{i,j}]_{z_{i,j}} \approx \frac{D_{i,j}(z_{i,j} + h_z) - D_{i,j}(z_{i,j} - h_z)}{2h_z} \,, \tag{3.52}$$

where $D_{i,j}(z_{i,j} \pm h_z)$ means that we compute $D_{i,j}$ with the varied z.

In order to discretise the second-order derivatives in the contributions from the smoothness term in Equation (3.48) we can use a central difference scheme. The discretisation of $\nabla_2^\top(\Psi'(S^2)\nabla_2(z + dz))$ is then given by

$$
\begin{aligned}
&\nabla_2^\top(\Psi'(S^2_{i,j})\nabla_2(z_{i,j}+dz_{i,j}))\\
&\approx\frac{1}{h_x^2}\left([\Psi'(S^2)(z+dz)_{xx}]_{i+1,j}-2[\Psi'(S^2)(z+dz)_{xx}]_{i,j}+[\Psi'(S^2)(z+dz)_{xx}]_{i-1,j}\right)\\
&\quad+\frac{2}{4h_xh_y}\Big([\Psi'(S^2)(z+dz)_{xy}]_{i+1,j+1}-[\Psi'(S^2)(z+dz)_{xy}]_{i+1,j-1}\\
&\quad-[\Psi'(S^2)(z+dz)_{xy}]_{i-1,j+1}+[\Psi'(S^2)(z+dz)_{xy}]_{i-1,j-1}\Big)\\
&\quad+\frac{1}{h_y^2}\left([\Psi'(S^2)(z+dz)_{yy}]_{i,j+1}-2[\Psi'(S^2)(z+dz)_{yy}]_{i,j}+[\Psi'(S^2)(z+dz)_{yy}]_{i-1,j-1}\right)\\
&=\frac{1}{h_x^4}\\
&\quad\Big((\Psi'(S^2)_{i+1,j}(z+dz)_{i+2,j}-2\Psi'(S^2)_{i+1,j}(z+dz)_{i+1,j}+\Psi'(S^2)_{i+1,j}(z+dz)_{i,j})\\
&\quad-2\left(\Psi'(S^2)_{i,j}(z+dz)_{i+1,j}-2\Psi'(S^2)_{i,j}(z+dz)_{i,j}+\Psi'(S^2)_{i,j}(z+dz)_{i-1,j}\right)\\
&\quad+(\Psi'(S^2)_{i-1,j}(z+dz)_{i,j}-2\Psi'(S^2)_{i-1,j}(z+dz)_{i-1,j}+\Psi'(S^2)_{i-1,j}(z+dz)_{i-2,j})\Big)\\
&\quad+\frac{2}{16h_xh_y}\\
&\quad\Big((\Psi'(S^2)_{i+1,j+1}(z+dz)_{i+2,j+2}-\Psi'(S^2)_{i+1,j+1}(z+dz)_{i+2,j}\\
&\quad-\Psi'(S^2)_{i+1,j+1}(z+dz)_{i,j+2}+\Psi'(S^2)_{i+1,j+1}(z+dz)_{i,j})\\
&\quad-(\Psi'(S^2)_{i+1,j-1}(z+dz)_{i+2,j}-\Psi'(S^2)_{i+1,j-1}(z+dz)_{i+2,j-2}\\
&\quad-\Psi'(S^2)_{i+1,j-1}(z+dz)_{i,j}+\Psi'(S^2)_{i+1,j-1}(z+dz)_{i,j-2})\\
&\quad-(\Psi'(S^2)_{i-1,j+1}(z+dz)_{i,j+2}-\Psi'(S^2)_{i-1,j+1}(z+dz)_{i,j}\\
&\quad-\Psi'(S^2)_{i-1,j+1}(z+dz)_{i-2,j+2}+\Psi'(S^2)_{i-1,j+1}(z+dz)_{i-2,j})\\
&\quad+(\Psi'(S^2)_{i-1,j-1}(z+dz)_{i,j}-\Psi'(S^2)_{i-1,j-1}(z+dz)_{i,j-2}\\
&\quad-\Psi'(S^2)_{i-1,j-1}(z+dz)_{i-2,j}+\Psi'(S^2)_{i-1,j-1}(z+dz)_{i-2,j-2})\Big)\\
&\quad+\frac{1}{h_y^4}\\
&\quad\Big((\Psi'(S^2)_{i,j+1}(z+dz)_{i,j+2}-2\Psi'(S^2)_{i,j+1}(z+dz)_{i,j+1}+\Psi'(S^2)_{i,j+1}(z+dz)_{i,j})\\
&\quad-2\left(\Psi'(S^2)_{i,j}(z+dz)_{i,j+1}-2\Psi'(S^2)_{i,j}(z+dz)_{i,j}+\Psi'(S^2)_{i,j}(z+dz)_{i,j-1}\right)\\
&\quad+(\Psi'(S^2)_{i,j-1}(z+dz)_{i,j}-2\Psi'(S^2)_{i,j-1}(z+dz)_{i,j-1}+\Psi'(S^2)_{i,j-1}(z+dz)_{i,j-2})\Big)
\end{aligned}
\tag{3.53}
$$

Now that we have all the discretised parts, we obtain the fully discretised version of Equation (3.48):

$$\begin{aligned} 0 =& \Bigg(D_{i,j}^k [D_{i,j}^k]_{z_{i,j}^k} + [D_{i,j}^k]_{z_{i,j}^k} [D_{i,j}^k]_{z_{i,j}^k} dz_{i,j}^{k,l+1} \\ &+ \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_H} [D_{\tilde{i},\tilde{j}}^k]_{z_{i,j}^k} \\ &+ \sum_{(\hat{i},\hat{j}) \in \mathcal{N}_D} [D_{\hat{i},\hat{j}}^k]_{z_{\hat{i},\hat{j}}^k} [D_{\hat{i},\hat{j}}^k]_{z_{i,j}^k} dz_{\hat{i},\hat{j}}^{k,l+1} \Bigg) \\ &+ \alpha \left(w_{i,j} (z_{i,j}^{k,l+1} + dz_{i,j}^{k,l+1}) + \sum_{(\bar{i},\bar{j}) \in \mathcal{N}_S} w_{\bar{i},\bar{j}} (z_{\bar{i},\bar{j}}^{k,l+1} + dz_{\bar{i},\bar{j}}^{k,l+1}) \right) . \end{aligned} \tag{3.54}$$

The involved neighbourhoods $\mathcal{N}_H$, $\mathcal{N}_D$, and $\mathcal{N}_S$ in Equation (3.54) contain the following neighbouring positions:

$$\begin{aligned} \mathcal{N}_H &= \{(i-1,j), (i+1,j), (i,j-1), (i,j+1)\} , \\ \mathcal{N}_D &= \{(i-2,j), (i-1,j), (i-1,j-1), (i-1,j+1), (i,j-1), (i,j-2), \\ &\qquad (i,j+1), (i,j+2), (i+1,j-1), (i+1,j+1), (i+1,j), (i+2,j)\} , \\ \mathcal{N}_S &= \{(i-2,j-2), (i-2,j), (i-2,j+2), (i-1,j), (i,j-2), (i,j-1), \\ &\qquad (i,j+1), (i,j+2), (i+1,j), (i+2,j-2), (i+2,j), (i+2,j+2)\} . \end{aligned} \tag{3.55}$$

Further, $w_{i,j}$ stands for the weight for the smoothness term corresponding to the location (i,j). These weights for the whole neighbourhood are given in the stencil (3.2):

After that, we have to solve the sparse linear system of equations for $dz^{k,l+1}$. To this end, Maurer et al. applied an iterative solver called the *successive over-relaxation method* (SOR) [You54] to solve the derived linear equation system efficiently. This solver is executed at each inner loop l where an iteration step s looks as follows:

	$i-2$	$i-1$	i	$i+1$	$i+2$
$j-2$	$\frac{1}{8h_x^2h_y^2}(\Psi'(S^2)^{k,l}_{i-1,j-1})$		$\frac{1}{h_y^4}(\Psi'(S^2)^{k,l}_{i,j-1})$ $-\frac{1}{8h_x^2h_y^2}(\Psi'(S^2)^{k,l}_{i-1,j-1}$ $+\Psi'(S^2)^{k,l}_{i+1,j-1})$		$\frac{1}{8h_x^2h_y^2}(\Psi'(S^2)^{k,l}_{i+1,j-1})$
$j-1$			$\frac{-2}{h_y^4}(\Psi'(S^2)^{k,l}_{i,j-1}+(S^2)^{k,l}_{i,j})$		
j	$\frac{1}{h_x^4}(\Psi'(S^2)^{k,l}_{i-1,j})$ $-\frac{1}{8h_x^2h_y^2}$ $(\Psi'(S^2)^{k,l}_{i-1,j-1}$ $+\Psi'(S^2)^{k,l}_{i-1,j+1})$	$\frac{-2}{h_x^4}(\Psi'(S^2)^{k,l}_{i-1,j}$ $+\Psi'(S^2)^{k,l}_{i,j})$	$\frac{1}{h_x^4}(\Psi'(S^2)^{k,l}_{i-1,j}$ $+4\Psi'(S^2)^{k,l}_{i,j}+\Psi'(S^2)^{k,l}_{i+1,j})$ $+\frac{1}{h_y^4}(\Psi'(S^2)^{k,l}_{i,j-1}$ $+4\Psi'(S^2)^{k,l}_{i,j}+\Psi'(S^2)^{k,l}_{i,j+1})$ $+\frac{1}{8h_x^2h_y^2}$ $(\Psi'(S^2)^{k,l}_{i-1,j-1}+\Psi'(S^2)^{k,l}_{i-1,j+1}$ $+\Psi'(S^2)^{k,l}_{i+1,j-1}+\Psi'(S^2)^{k,l}_{i+1,j+1})$	$\frac{-2}{h_x^4}(\Psi'(S^2)^{k,l}_{i,j}$ $+\Psi'(S^2)^{k,l}_{i+1,j})$	$\frac{1}{h_x^4}(\Psi'(S^2)^{k,l}_{i+1,j})$ $-\frac{1}{8h_x^2h_y^2}$ $(\Psi'(S^2)^{k,l}_{i+1,j-1}$ $+\Psi'(S^2)^{k,l}_{i+1,j+1})$
$j+1$			$\frac{-2}{h_y^4}(\Psi'(S^2)^{k,l}_{i,j}+\Psi'(S^2)^{k,l}_{i,j+1})$		
$j+2$	$\frac{1}{8h_x^2h_y^2}(\Psi'(S^2)^{k,l}_{i-1,j+1})$		$\frac{1}{h_y^4}(\Psi'(S^2)^{k,l}_{i,j+1})$ $-\frac{1}{8h_x^2h_y^2}$ $(\Psi'(S^2)^{k,l}_{i-1,j+1}+\Psi'(S^2)^{k,l}_{i+1,j+1})$		$\frac{1}{8h_x^2h_y^2}(\Psi'(S^2)^{k,l}_{i+1,j+1})$

Table 3.2: Stencil for the smoothness term.

$$
\begin{aligned}
dz_{i,j}^{k,l+1,s+1} =& (1-\omega) dz_{i,j}^{k,l+1,s} \\
&+ \omega \Bigg(- D_{i,j}^{k} [D_{i,j}^{k}]_{z_{i,j}^{k}} - \alpha w_{i,j} (z_{i,j}^{k,l+1} \\
&- \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_H} [D_{\tilde{i},\tilde{j}}^{k}]_{z_{i,j}^{k}} \\
&- \sum_{(\hat{i},\hat{j}) \in \mathcal{N}_D^-} [D_{\tilde{i},\tilde{j}}^{k}]_{z_{\hat{i},\hat{j}}^{k}} [D_{\tilde{i},\tilde{j}}^{k}]_{z_{i,j}^{k}} dz_{\hat{i},\hat{j}}^{k,l+1,s+1}) \\
&- \sum_{(\tilde{i},\tilde{j}) \in \mathcal{N}_D^+} [D_{\tilde{i},\tilde{j}}^{k}]_{z_{\hat{i},\hat{j}}^{k}} [D_{\tilde{i},\tilde{j}}^{k}]_{z_{i,j}^{k}} dz_{\hat{i},\hat{j}}^{k,l+1,s} \\
&- \alpha \sum_{(\bar{i},\bar{j}) \in \mathcal{N}_S^-} w_{\bar{i},\bar{j}} (z_{\bar{i},\bar{j}}^{k,l+1} + dz_{\bar{i},\bar{j}}^{k,l+1,s+1}) \\
&- \alpha \sum_{(\bar{i},\bar{j}) \in \mathcal{N}_S^+} w_{\bar{i},\bar{j}} (z_{\bar{i},\bar{j}}^{k,l+1} + dz_{\bar{i},\bar{j}}^{k,l+1,s}) \Bigg) \\
&\left(\alpha w_{i,j} + [D_{i,j}^{k}]_{z_{i,j}^{k}} [D_{i,j}^{k}]_{z_{i,j}^{k}} \right)^{-1} .
\end{aligned}
\tag{3.56}
$$

Here, $\mathcal{N}_D^-$ and $\mathcal{N}_S^-$ denote the set of neighbouring pixels that have already been updated. Accordingly, $\mathcal{N}_D^+$ and $\mathcal{N}_S^+$ refer to the pixels in the neighbourhood that have not been updated yet.

Note that the SOR method is a variant of the Gauss-Seidel method with faster convergence elaborated by David Young [You54]. The SOR method can be reduced to the Gauss-Seidel method by using the over-relaxation parameter $\omega = 1$.

Another advantage of this linearisation technique is that it allows to speed up the computation on the same synthetic test images by more than three orders of magnitude compared to their basis variational SfS model [JMBB16] without reducing the quality of the reconstruction significantly [MJBB15]. But it also entails a problem which Maurer and colleagues had to tackle. In fact, incorrect increments can misdirect the computation. Therefore, they had to control the size of the updates by cropping the computed increments $dz^{k,l_{\max}}$, so that

$$
|dz^{k,l_{\max}}| \leq dz_{\text{limit}} \, . \tag{3.57}
$$

This is done after sufficient solver iterations and sufficient fix point iterations l. Then, the depth is updated via

$$z^{k+1} = z^k + dz \ , \tag{3.58}$$

where dz is defined as

$$dz := \begin{cases} -dz_{\text{limit}} & \text{if} \quad dz^{k,l_{max}} > dz_{\text{limit}} \ , \\ dz_{\text{limit}} & \text{if} \quad dz^{k,l_{max}} < -dz_{\text{limit}} \ , \\ dz^{k,l_{max}} & \text{else} \ . \end{cases} \tag{3.59}$$

3.6 Model Extensions

In this section we further extend our model to a PS approach that is based on the image ratios and therefore is independent from the albedo ρ. Hence, we will refer to this PS model as *Unbiased Photometric Stereo*.

In addition, we present an extension called *light estimation* that is applied to the PS approach with the general lighting model where we try to estimate the SH coefficients for every resolution level.

3.6.1 Unbiased Photometric Stereo

Since the images used in PS models are taken from the same camera location we obtain a lot of overlapping information. Hence, one can simplify the PS problem by using the image ratios which makes the task independent from the albedo at each pixel. The PS approach where the ratios of images is considered was introduced by Davis and Soderblom [AA84]. After that, this approach has been used in many different works, e.g. [MK96] [HVC11] [QMD16]. Another advantage of this technique is that it can easily be implemented in our PS model since it can be applied with the perspective camera model and the different illumination models. Consequently, we integrate this extension in all the different variants of our model mentioned in the sections above and compare the results. With this, we try to adopt the feature of being independent of the albedo to our technique.

As an example we show the resulting constraint for the data term by means of the distant point light source model. In order to derive the data term for the unbiased PS approach, we take a brief look at the brightness equation for the distant illumination model in Table 3.1:

$$I(\mathbf{x}) = \rho(\mathbf{x})\mathbf{N}(\mathbf{x}) \cdot \mathbf{L}(\mathbf{x}) \ . \tag{3.60}$$

Now we can rearrange the equation above to solve for ρ which leads to

$$\rho(\mathbf{x}) = \frac{\mathbf{I}(\mathbf{x})}{\mathbf{N}(\mathbf{x}) \cdot \mathbf{L}(\mathbf{x})} \ . \tag{3.61}$$

In the next step we take two images i and j into the equation and since the albedo depends solely on the reflectance of the surface, we obtain the following equation:

$$\frac{\mathbf{I}_i(\mathbf{x})}{\mathbf{N}(\mathbf{x}) \cdot \mathbf{L}_i(\mathbf{x})} = \rho(\mathbf{x}) = \frac{\mathbf{I}_j(\mathbf{x})}{\mathbf{N}(\mathbf{x}) \cdot \mathbf{L}_j(\mathbf{x})} \ . \tag{3.62}$$

With this, we can derive the constraint for unbiased PS that is independent from surface color and considers all combinations of the m given pictures. The resulting constraint for the data term looks as follows:

$$\sum_{i=1}^{m} \sum_{j=i+1}^{m} \frac{\mathbf{I}_i(\mathbf{x})}{\mathbf{N}(\mathbf{x}) \cdot \mathbf{L}_i(\mathbf{x})} - \frac{\mathbf{I}_j(\mathbf{x})}{\mathbf{N}(\mathbf{x}) \cdot \mathbf{L}_j(\mathbf{x})} \ = \ 0 \ . \tag{3.63}$$

Theoretically, Equation 3.63 could be used as D in our data term. But the computation of the fraction can lead to very high errors when the difference between the computed and the actual image value is high. Consequently, this leads to false estimations in the darker areas, especially when we use the near point light source model with the attenuation term that further decreases the computed illumination. Therefore, we dissolved the fraction by multiplying with the denominators and introduced the natural logarithm in order to obtain the following equation:

$$\sum_{i=1}^{m} \sum_{j=i+1}^{m} \ln(\mathbf{L}_i \cdot \mathbf{N}) - \ln(\mathbf{L}_j \cdot \mathbf{N}) + \ln(\mathbf{I}_j) - \ln(\mathbf{I}_i) = 0 \ . \tag{3.64}$$

Equation (3.64) can then be integrated as the data term in our energy functional (3.20) and minimised as stated in Section 3.5.1. In order to improve the results with this approach we additionally used a pixel-wise similarity measure that induces to not use the information from the pixels if the values provided by the two images are too different. To be precise, we only use the current pixel information when the following holds:

$$\frac{\mathbf{I}_i(\mathbf{x})}{\mathbf{I}_i(\mathbf{x}) + \mathbf{I}_j(\mathbf{x})} < 0.9 \quad \wedge \quad \frac{\mathbf{I}_i(\mathbf{x})}{\mathbf{I}_i(\mathbf{x}) + \mathbf{I}_j(\mathbf{x})} > 0.1 \ . \tag{3.65}$$

3.6.2 Light Estimation

As a second extension we implemented a technique to estimate the SH coefficients l_k based on the estimated z. In detail, this is done by solving the linear system of equations, which we derived from the brightness equation for the general lighting model based on spherical harmonics. The linear system of equations $A\mathbf{c} = \mathbf{b}$ looks as follows:

$$\underbrace{\begin{pmatrix} H_1H_1 & H_2H_1 & \cdots & H_9H_1 \\ H_1H_2 & H_2H_2 & \cdots & H_9H_2 \\ \vdots & \vdots & \ddots & \vdots \\ H_1H_9 & H_2H_9 & \cdots & H_9H_9 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} l_1 \\ l_2 \\ \vdots \\ l_9 \end{pmatrix}}_{\mathbf{c}} = \underbrace{\begin{pmatrix} H_1I \\ H_2I \\ \vdots \\ H_9I \end{pmatrix}}_{\mathbf{b}} \tag{3.66}$$

where, H_k denotes the values for the SH basis function, I stand for the image values, and l_k are the missing light coefficients. Note that every entry in A and b is added up over all pixels in the image. We then solve Equation (3.66) with a Gaussian elimination solver to obtain the estimated coefficients l_k.

Since we need a sufficiently well estimated z in order to estimate the light coefficients, we start the *light estimation* after a few coarse-to-fine iterations where the light coefficients are given. In order to compute the needed basis function $H(\mathbf{N})$ we use the latest computed depth z. This technique could still be used to improve the computation when the given coefficients are not fully correct, e.g. in an uncalibrated environment.

4 Evaluation

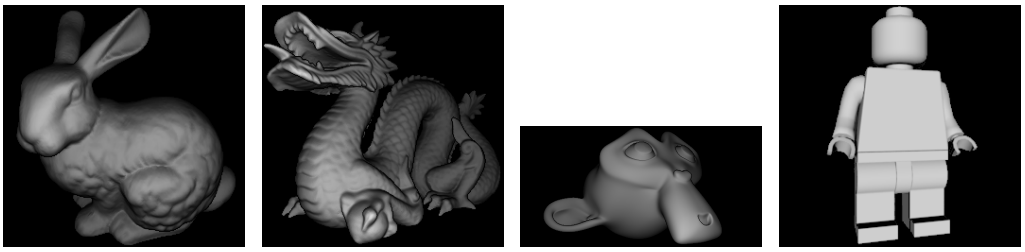
In this chapter we present the evaluation of performance tests under different testing environments for all the variants of our model mentioned in Chapter 3. Before we start looking at the results our variational PS approach produces, we give background information about how our synthetic test images are constructed and the different test cases we use.

4.1 Test Images

For the synthetic test images we use four different objects. Two of them are well-known models from the *Stanford 3D Scanning Repository*, namely the *Stanford Bunny* and the *Dragon*. Another object is the also popular *Suzanne*, a model of a chimpanzee head known from the 3D graphics software *Blender*. The last test object is a simple model of a LEGO figure. All these test models are shown in Figure 4.1.

In order to create the images we loaded the test model files in Blender, extracted a depth file and the intrinsic camera parameters with the help of a script. Furthermore, we used the file that holds the depth for every pixel to compute the normal vector of the surface. With that, the pixel values can be computed by using the different brightness equations mentioned in the previous chapter.

In detail, for the near point light source model we generated the light position for each image in such a way that they are Gaussian distributed on a half sphere in front of the object. The same method was used to generate the light vectors in the distant light source case. For the general lighting model based on Spherical Harmonics we used the method in [RH01] to extract the missing light coefficients out of different environment maps. All of these synthetic images are generated as 8-bit greyscale images.



(a) Stanford Bunny **(b)** Dragon **(c)** Suzanne **(d)** Lego

Figure 4.1: The test objects used in the following experiments.

4.2 Quality Measures

Since we have the ground truth depth map for each object, we can introduce the following metrics to measure the quality of the reconstructed surface.

Relative Surface Error. The first error metric we use is called *Relative Surface Error* (RSE). As the name implies, this metric describes the relative closeness from the reconstructed surface to the ground truth surface by means of the Euclidean distance between the corresponding points. The denominator causes the computed distance between the surfaces to be relative to the ground truth surface. With this said, the RSE reads

$$\text{RSE} = \frac{\sqrt{\sum_{\Omega_{i,j}} |\mathcal{S}_{i,j} - \mathcal{S}_{i,j}^{\text{GT}}|}}{\sqrt{\sum_{\Omega_{i,j}} |\mathcal{S}_{i,j}^{\text{GT}}|}} \,. \tag{4.1}$$

Average Angular Error. The second error metrics we use for the evaluation is called *Average Angular Error* (AAE). The AAE describes the average angle between the computed normalised normal vector and ground truth normalised normal vector. These angles are then added up for every pixel and divided by the number of pixels in the image to compute the average. As already mentioned the dot product between two normalised vectors corresponds to the cosine of the angle between the vectors. In order to compute the angle between the normal vectors in degrees instead of radians we added the factor $\frac{180}{\pi}$ to the equation. Therefore, the AAE is defined as follows:

$$\text{AAE} = \frac{1}{nxny} \sum_{\Omega_{i,j}} \arccos\left(\mathbf{N}_{i,j} \cdot \mathbf{N}_{i,j}^{\text{GT}})\right) \frac{180}{\pi} \,. \tag{4.2}$$

4.3 Experiments

In this section we present the different experiments we performed on our model. With the first experiment we want to test the influence of the number of input images on the quality of the reconstructed surface.

The second one is similar to the first experiment with the difference that we use noisy test images and again survey the influence of more input images on the outcome. Further, we test the quality of the reconstruction computed with the unbiased PS approach in the third experiment. For this, we multiplied the test images with a texture image to obtain images where with a non-uniform albedo.

In our last test we want to try to estimate the light information for the SH illumination model. In detail, these are the SH coefficients for the general lighting model where we provide the coefficients for the first resolution levels of the computation and continue with fully estimated SH coefficients for the following levels. Furthermore, we present a comparison between the different variants of our PS model for each of the mentioned experiments.

In order to compare the computed surface normals to the ground truth we use a colour scheme where the colours are mapped to the direction where the surface normal is pointing. In addition, we present images of the computed 3D point cloud by using the program *MeshLab* which allows us to reconstruct the surface with the estimated 3D points. The outcomes are then shown in the colour scheme of the surface normals or as a depth map of the model. The depth map illustrates the estimated depth where dark areas are closer to the observer and brighter areas are further away.

Parameters In all our experiments we keep the following parameters fixed in order to provide a suitable comparison between the model variants. The weight of the smoothness term $alpha = 0.08$, the downsampling factor between consecutive resolution levels $\eta = 0.9$, the iterations per resolution level $\kappa = 5$, the number of fix point iterations $l_{\text{max}} = 10$, 10 iterations for the SOR solver with $\omega = 1.8$, the parameter to crop the computed increment $dz_{\text{limit}} = 0.01$, the parameter $\lambda = 0.01$ in the Charbonnier function, and $h_z = 10^{-12}$ for the numerical computation of the derivatives with respect to z. Furthermore, we start the computation with an initial depth plane parallel to the image plane located at $z = 1.5$.

DL	3 images		6 images		10 images		15 images		20 images	
	RSE	AAE	RSE	AAE	RSE	AAE	RSE	AAE	RSE	AAE
Stanford Bunny	0.39075	4.20985	0.31011	1.46851	0.15756	1.13615	0.54028	1.10505	0.44787	0.74537
Dragon	0.52088	24.83787	0.25481	5.65841	0.21259	4.29755	0.36885	3.82705	0.32957	3.20775
Suzanne	0.48203	7.03492	0.31795	0.63646	0.35614	0.55957	0.45411	0.42056	0.44383	0.29933
Lego	0.12789	6.66228	0.13373	4.65844	0.22320	3.69162	0.05078	1.91113	0.16049	2.95470

Table 4.1: Results for the distant light source model.

4.3.1 Influence of Number of Images

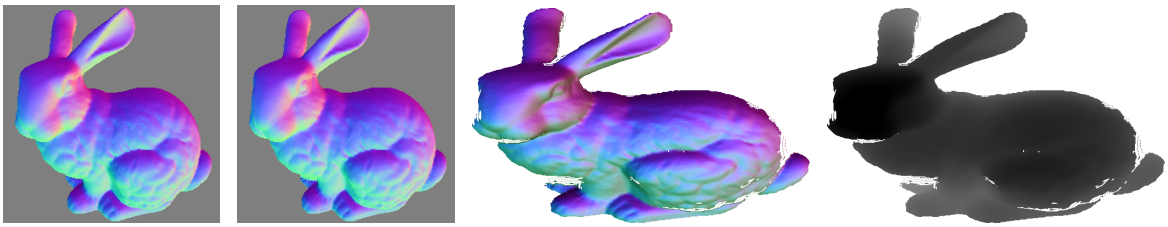
In this first experiment we want to evaluate the influence of the number of input images on the quality of the reconstructed surface. Therefore, we compared the outcomes of the different model variants with 3, 6, 10, 15, and 20 input images. The relative surface error and the average angular error for each object are then presented for each illumination model.

Distant Light Source Model. Let us first have a look at the results for the distant light source model. Here, the results show that for all test objects the relative surface error does not improve when we use more input images. Moreover, we can see that the RSE even fluctuates over the number of input images. This is the case, since the distant light source model just uses the light direction and the position of the light source is assumed to be infinitely far away from the object. Hence, the light attenuation over the travelled distance cannot be computed. Consequently, the distance to the ground truth surface depends highly on the initialisation of z and an even distribution of light directions over the input images.

Nevertheless, the average angular error does decrease significantly with more input images and the surface normals are reconstructed pretty well with this illumination model. With the distant light source PS model we even get an AAE $< 1^\circ$ for the Stanford Bunny and the Suzanne test model when we use 20 input images. For the most complex test object, the Dragon, this illumination model at least gives us an AAE of about 3.2°.

The surface normals computed with 20 input images for the Stanford Bunny compared to its ground truth normals are shown in Figure 4.2. Additionally, we present the corresponding reconstructed surface with the same color mapping in the same figure.

Table 4.1 presents the resulting RSE and AAE for the evaluated number of input images in detail.



(a) GT N. **(b)** Comp. N. **(c)** Surface. **(d)** Depth map.

Figure 4.2: Result images for the distant light source model with 20 input images. Object: Stanford Bunny.

Near Light Source Model. After we have shown how the distant light source PS model performs with more input images, we want to know, whether the same holds for the near point light illumination model. The setup for this test is exactly the same as for the distant light source model above. Again, all the results are presented in Table 4.2 in terms of the computed errors.

As expected, the near point light source model is able to achieve a much lower relative surface error than the PS model using the distant light source model. This is the case due to the additional distance information given by the light attenuation term and the exact light direction that gets computed for each pixel. Therefore, the near point light source PS model is much more precise in the estimation of the actual depth z which shows in a RSE below 0.06 for all test objects even when only 3 input images are given. Overall, the RSE is much more stable for this PS model and even decrease when we use more input images for all objects.

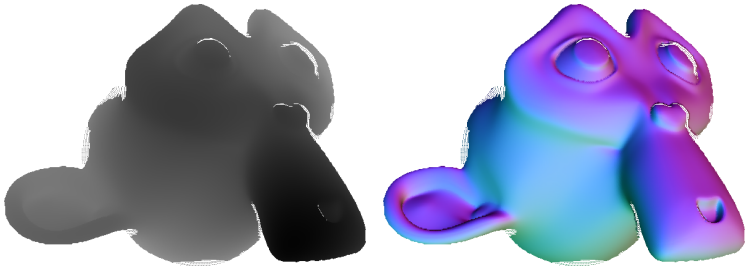
Considering the AAE when using 3 images for this illumination model the results are in average slightly better than for the former light source model. But when we look at the progression of this error over the number of given input images, we see that the it decreases not as much when we add more images compared to the distant light source PS model. This leads to the fact, that the distant illumination model beats the near illumination model regarding the AAE when we use 20 input images.

As before, the best results are achieved with the Suzanne model due to its simple surface, the computed surface and the depth map is shown in Figure 4.3. In comparison, we present the resulting images for the dragon model in Figure 4.4, which produced the highest error.

General Lighting Model. We complete this first experiment with the results for the SH illumination model. Again, the conditions and parameters are the same as for the previously covered illumination models.

NL	3 images		6 images		10 images		15 images		20 images	
	RSE	AAE	RSE	AAE	RSE	AAE	RSE	AAE	RSE	AAE
Stanford Bunny	0.02354	4.21871	0.01786	3.07364	0.01473	2.19313	0.01309	1.79815	0.01217	1.57330
Dragon	0.05767	11.03400	0.04567	7.80622	0.03970	6.01537	0.03675	5.20891	0.03545	4.62976
Suzanne	0.01281	2.64974	0.01112	1.84713	0.00926	1.29241	0.00853	0.96189	0.00800	0.81613
Lego	0.02527	5.33301	0.01586	2.99302	0.01465	2.30348	0.01396	1.77948	0.01384	1.56066

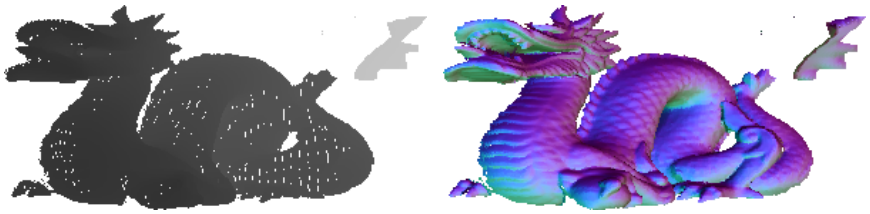
Table 4.2: Results for the near light source model.



(a) Depth map. **(b)** Surface.

Figure 4.3: Result images for the near light source model with 20 input images. Object: Suzanne.

The outcomes for the general lighting model based on SH are more inconsistent over the number of input images than the result from the other illumination models as presented in Table 4.3. This means that although the results on average get better with more input images, the quality of the reconstruction depends strongly on the quality of the input images and 20 input images are not enough to fully balance out this dependency. But in general we should see a further improvement of the reconstructed surface normal when we use more input images and consequently a lower AAE. Figure 4.5 shows this effect where the reconstruction for the Stanford Bunny with 10 input images is compared to the result with 20 input images.

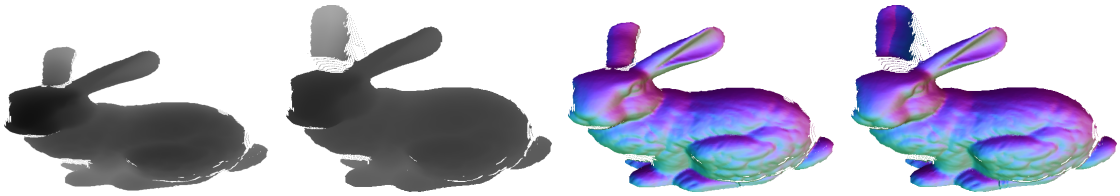


(a) Depth map. **(b)** Surface.

Figure 4.4: Result images for the near light source model with 20 input images. Object: Dragon.

SH	3 images		6 images		10 images		15 images		20 images	
	RSE	AAE	RSE	AAE	RSE	AAE	RSE	AAE	RSE	AAE
Stanford Bunny	0.51832	5.70269	0.50510	3.35017	0.49643	0.99578	0.46239	2.76553	0.44789	2.87659
Dragon	0.44455	16.78433	0.52040	8.79879	0.52861	4.79331	0.53227	3.94948	0.52200	4.51251
Suzanne	0.46946	4.67310	0.42471	1.47409	0.32254	0.62607	0.34539	0.66061	0.33893	0.81204
Lego	0.57845	4.86854	0.56119	16.22645	0.48979	3.16317	0.54368	3.49392	0.50616	1.22515

Table 4.3: Results for the general lighting model.



(a) Depth, 10 imgs. **(b)** Depth, 20 imgs. **(c)** Surface, 10 imgs. **(d)** Surface, 20 imgs.

Figure 4.5: Results for the SH illumination model. Comparison between 10 and 20 input images. Model: Stanford Bunny.

Similar to the distant light source model the relative surface error for the SH illumination model stays relatively the same once a stable reconstruction is reached. This is again due to the fact that the SH illumination model is a general lighting model that only depends on the surface normal and can not provide further distance information to the PS model.

4.3.2 Noisy Test Images

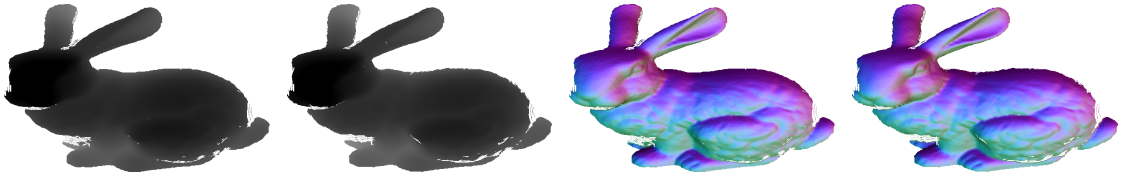
For our second experiment we used the same test images as before but now we added a Gaussian noise with $\sigma = 20$ to the input images. Then, we followed the same test procedure where we want to evaluate the influence of the number of input images on the outcome of the computation.

Distant Light Source Model. As in the first experiment, we start with the distant light source PS model. Compared to the results with the standard test images we notice a drop in the performance when only 3 input images are given. In detail, the AAE is higher for all test models and the reconstructed surface looks more ruffled as shown in Figure 4.6. However, the relative surface error is close to the error computed in the first experiment.

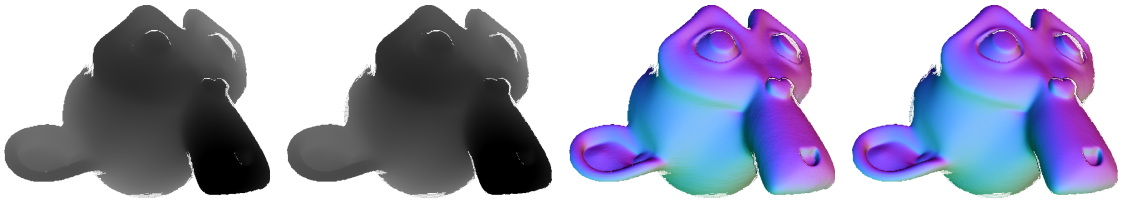
When we observe the change of these errors when more input images are used, shown in Table 4.4, we can tell that the RSE behaves like explained in the first test. But the

DL	3 images		6 images		10 images		15 images		20 images	
	RSE	AAE	RSE	AAE	RSE	AAE	RSE	AAE	RSE	AAE
Stanford Bunny	0.39932	7.71790	0.35124	3.08081	0.06751	2.40990	0.53013	2.05700	0.45265	1.78180
Dragon	0.51247	26.86592	0.31066	7.71108	0.29465	5.54470	0.38405	4.84847	0.34054	3.95813
Suzanne	0.47345	7.88373	0.34670	1.93042	0.33629	1.56972	0.46171	1.32904	0.44889	1.26547
Lego	0.09855	11.99019	0.23585	4.38992	0.22349	3.45652	0.13558	3.20089	0.14662	3.86978

Table 4.4: Results for the distant light source model under noise.



(a) Depth, 6 imgs. **(b)** Depth, 20 imgs. **(c)** Surface, 6 imgs. **(d)** Surface, 20 imgs.



(e) Depth, 6 imgs. **(f)** Depth, 20 imgs. **(g)** Surface, 6 imgs. **(h)** Surface, 20 imgs.

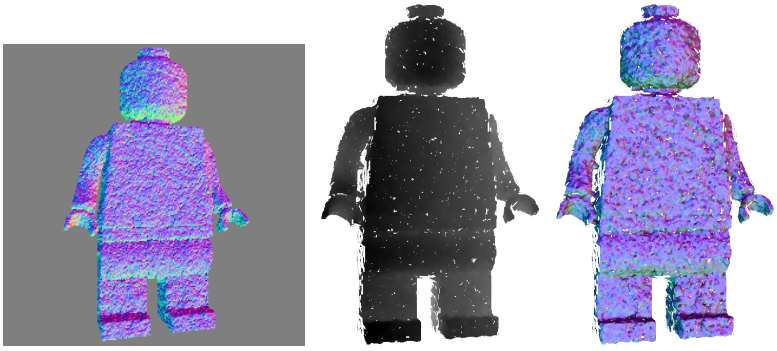
Figure 4.6: Comparison of outcomes for the distant light source model under noise. Objects: Stanford Bunny (top row), Suzanne (bottom row).

computation of the surface normals again profits from more input images. The AAE decreases when more images with different light directions are given. This even leads to a smooth reconstructed surface illustrated in Figure 4.6, although the impact of the smoothness term, given by the weight α, has not been changed. Logically, the results are still not as good as for the experiment without noise but we can say that the distant light source PS model can handle the noisy images pretty well, if we use sufficient input images.

Near Light Source Model. As we already mentioned, the near point light source model is the most precise one out of the three variants of our PS approach because of the additional information with regards to the distance between the object and the light source and the computed light direction per pixel. But in this test setup with noisy input images this property becomes a drawback. The evaluation shows that the average angular error is much higher compared to the distant light source model. And further, this effect can not be countered by using more input images. The reason behind this is that the noise in the images leads to an erroneous computation of the depth component.

NL	3 images		6 images		10 images		15 images		20 images	
	RSE	AAE	RSE	AAE	RSE	AAE	RSE	AAE	RSE	AAE
Stanford Bunny	0.02159	18.21794	0.01322	19.28376	0.01052	19.81654	0.00995	20.14651	0.00921	20.33531
Dragon	0.05961	22.15925	0.04557	22.63644	0.04073	22.57236	0.03815	22.96672	0.03724	22.95617
Suzanne	0.00746	20.77004	0.00570	21.69885	0.00452	22.32122	0.00432	22.66794	0.00449	22.86964
Lego	0.03296	30.06756	0.04628	30.81433	0.04738	30.70240	0.04450	30.84894	0.04196	31.36786

Table 4.5: Results for the near light source model under noise.



(a) Computed Normals. **(b)** Depth map. **(c)** Surface.

Figure 4.7: Result images for the near light source model under noise with 20 input images. Model: Lego.

In other words, the noise creates small peaks and vales on the surface due to the precision of this approach.

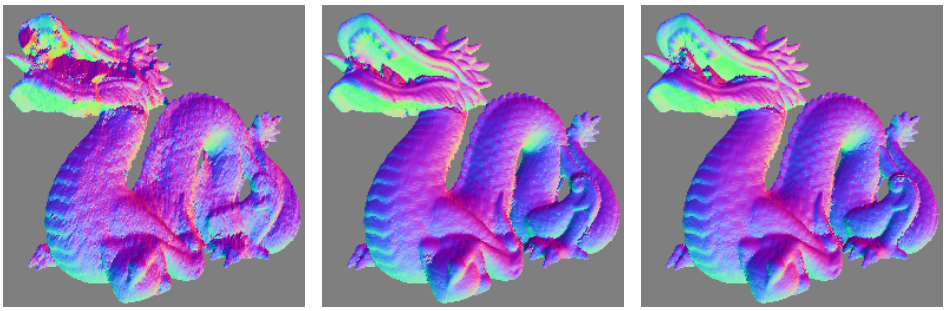
The euclidean distance from the computed surface to the ground truth surface however is still small. It is the same effect we observed with the distant light source model. The overall fitting of the initial z-plane can be done well, but the noise highly disturbs the reconstruction of the fine details. The computed normals and the computed depth map with 20 input images for the Lego object is shown in Figure 4.7. The exact computed errors for this experiment are presented in Table 4.2.

General Lighting Model. What now remains left is the evaluation of the SH illumination model under additive Gaussian noise. As we expected, this PS model performs quite similar with noisy images as it did without noise with respect to the relative surface error. The strong dependency on the quality of the input images is still visible and over the range of 20 input images there is no noticeable improvement regarding the RSE as shown in Table 4.6.

When we look at the AAE we see that it is heavily increased for the lower number of images due to the noise. But since the SH illumination model is an approach where the general lighting of the scene is modelled the errors resulting through the noise can be

SH	3 images		6 images		10 images		15 images		20 images	
	RSE	AAE	RSE	AAE	RSE	AAE	RSE	AAE	RSE	AAE
Stanford Bunny	0.50636	15.50376	0.50611	11.09803	0.48625	6.70451	0.45989	8.98995	0.44896	8.44254
Dragon	0.46723	26.80570	0.52516	15.77552	0.52888	8.76309	0.53144	9.28297	0.52873	8.63447
Suzanne	0.45910	16.51037	0.42785	10.10753	0.32495	6.83623	0.35428	7.46936	0.34957	7.07622
Lego	0.57977	23.54003	0.55982	24.70058	0.49164	8.59140	0.53991	11.02450	0.52439	8.98018

Table 4.6: Results for the SH illumination model under noise.



(a) 3 images. **(b)** 10 images. **(c)** 20 images.

Figure 4.8: Result images for the SH illumination model under noise. Computed surface normals. Model: Dragon.

smoothened out by using more images. Therefore, we see the same behaviour as for the distant light source model regarding the angle error. Figure 4.8 illustrates the effect of more input images on the computed surface normals.

As an extension to the previous experiment, we want to test whether an increased weighting parameter α can further improve the estimation of the surface normals under the influence of noise. Therefore, we set the $\alpha = 3.0$ and have a look at the outcome for the three PS model variants with 20 input images. The RSE and AAE for each model are presented in Table 4.7.

For the distant light source model the RSE is distinctly lower when we give the smoothness term a higher impact. We even obtain a better RSE with the noisy images than in our first experiment where we used the parameter $\alpha = 0.08$. But as a downside the blur induced by the smoothness term also makes the estimation of the surface normal slightly worse. However, the near point light source model, which performance got dropped the most by the noise, obviously profits from more smoothness to counter the aforementioned peaks and vales on the surface. The AAE is almost $10°$ lower for the simpler test objects but overall the reconstructed surface still contains a lot of errors, the irregularities just got slightly smaller. Some examples are shown in Figure 4.9. As a downside, we lose precision regarding the relative surface error when we use a higher α due to the distribution of information by the smoothness term. The effect of the increased α is not as strong regarding the PS approach with the SH illumination. The

noisy	DL		NL		SH	
$\alpha = 3$	RSE	AAE	RSE	AAE	RSE	AAE
Stanford Bunny	0.13571	2.96653	0.06862	10.71432	0.55200	5.84787
Dragon	0.15034	6.70752	0.11913	18.41285	0.57415	9.95840
Suzanne	0.13772	2.21046	0.03832	10.73685	0.04174	5.81829
Lego	0.15725	2.11303	0.04664	20.22211	0.60727	7.82319

Table 4.7: Comparison of the PS models under noise with $\alpha = 3.0$.

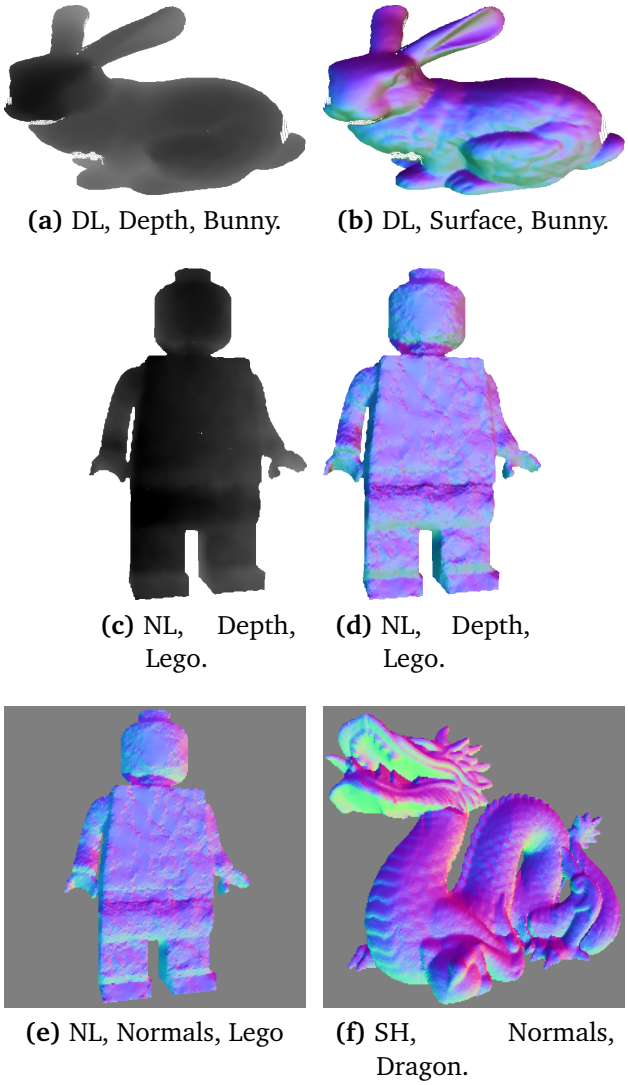
most improvement can be noticed with the simpler test models, Stanford Bunny and Suzanne. Here the AAE got further reduced and the computed surface looks smoother compared to the test run with the lower α.

Figure 4.9 shows the outcomes with the increased α on the same test object used above. With that said, we can compare them to the results with $\alpha = 0.08$ for each model shown in Figure 4.6, Figure 4.7, and Figure 4.8.

4.3.3 Non-uniform Albedo Images

Now we want to investigate how our extended PS model based on image ratios performs in comparison to our standard PS model. As mentioned, the theoretical advantage of the unbiased PS approach is that it is independent from the albedo ρ. In order to test this characteristic we multiplied a texture map, shown in Figure 4.10, on the known test images to obtain a change in the albedo. After that we continue the experiment under the usual testing environment. Since we are not interested in the influence that more input images have in this test, we only compare the performances of the PS models when 10 input images are provided.

Distant Light Source Model. As before we first have a look at the results when the distant light source model is used. As expected, for all test objects the albedo images make the reconstruction with the standard PS model worse with regards to the RSE and the AAE. For the Stanford Bunny and the Lego model the unbiased PS approach performs with a lower AAE but a higher RSE, but as already mentioned the distant light source model can be inconsistent regarding the RSE. The best results are again achieved on the simplest model, Suzanne. Here, we obtain the best reconstruction overall with the lowest RSE and AAE as shown in Figure 4.11. However, the reconstruction of the Dragon is again the hardest to do and the unbiased PS approach even performs worse on this test model than our standard approach. The computed errors for the two distant light source approaches are presented in Table 4.8.



(a) DL, Depth, Bunny. **(b)** DL, Surface, Bunny.

(c) NL, Depth, Lego. **(d)** NL, Depth, Lego.

(e) NL, Normals, Lego **(f)** SH, Normals, Dragon.

Figure 4.9: Result images for the PS models under noise with $\alpha = 3.0$.

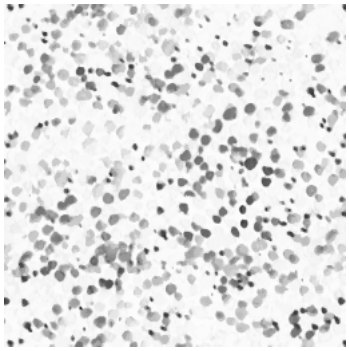
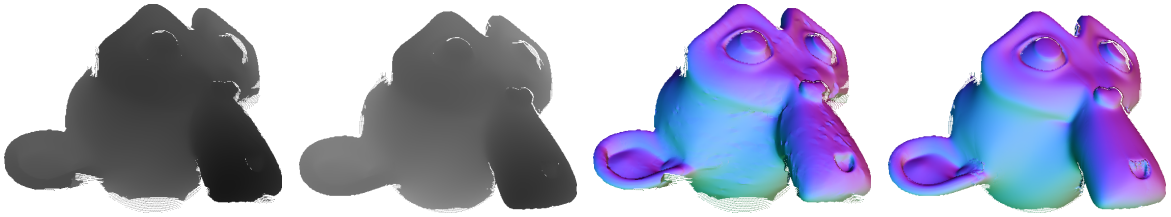


Figure 4.10: Texture used to modify the albedo of the test images.

DL	standard PS		unbiased PS	
	RSE	AAE	RSE	AAE
Stanford Bunny	0.21406	4.64634	0.43568	2.97402
Dragon	0.33203	8.86611	0.17788	10.66890
Suzanne	0.38083	4.52128	0.22497	1.92287
Lego	0.16630	7.58404	0.35900	4.02851

Table 4.8: Comparison of results for the distant light source models with non-uniform ρ.



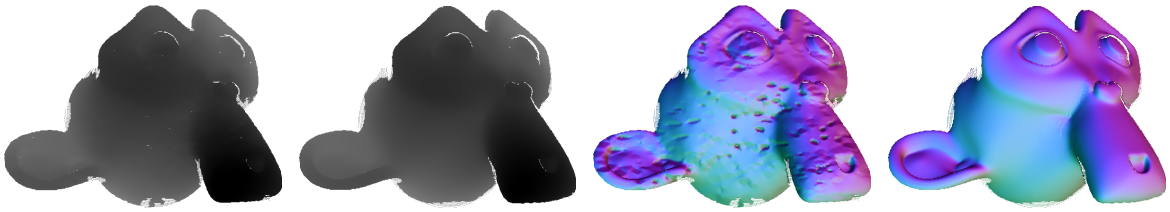
(a) Standard PS. **(b)** Unbiased PS. **(c)** Standard PS. **(d)** Unbiased PS.

Figure 4.11: Result images for the distant light source models with non-uniform ρ. Objects: Suzanne.

Near Light Source Model. Let us move on to our best performing approach so far, the near light source model. Here, the results are much more consistent over the different test objects. The unbiased PS model is able to reduce the AAE significantly compared to the standard approach. On top of that, the RSE is also lower for all test models. A side-by-side comparison between the two approaches is given in Figure 4.12 in terms of the reconstructed surface of Suzanne and the computed errors are given in Table 4.9. However, the unbiased PS approach does not perform well enough to come close to the errors computed in the first experiment.

NL	standard PS		unbiased PS	
	RSE	AAE	RSE	AAE
Stanford Bunny	0.08766	8.92199	0.03302	2.00486
Dragon	0.12491	12.36228	0.08687	7.85636
Suzanne	0.09365	10.34390	0.04954	2.11723
Lego	0.06821	11.85704	0.05551	2.22445

Table 4.9: Comparison of results for the near light source models with non-uniform ρ.



(a) Standard PS. **(b)** Unbiased PS. **(c)** Standard PS. **(d)** Unbiased PS.

Figure 4.12: Result images for the near light source models with non-uniform ρ. Objects: Suzanne.

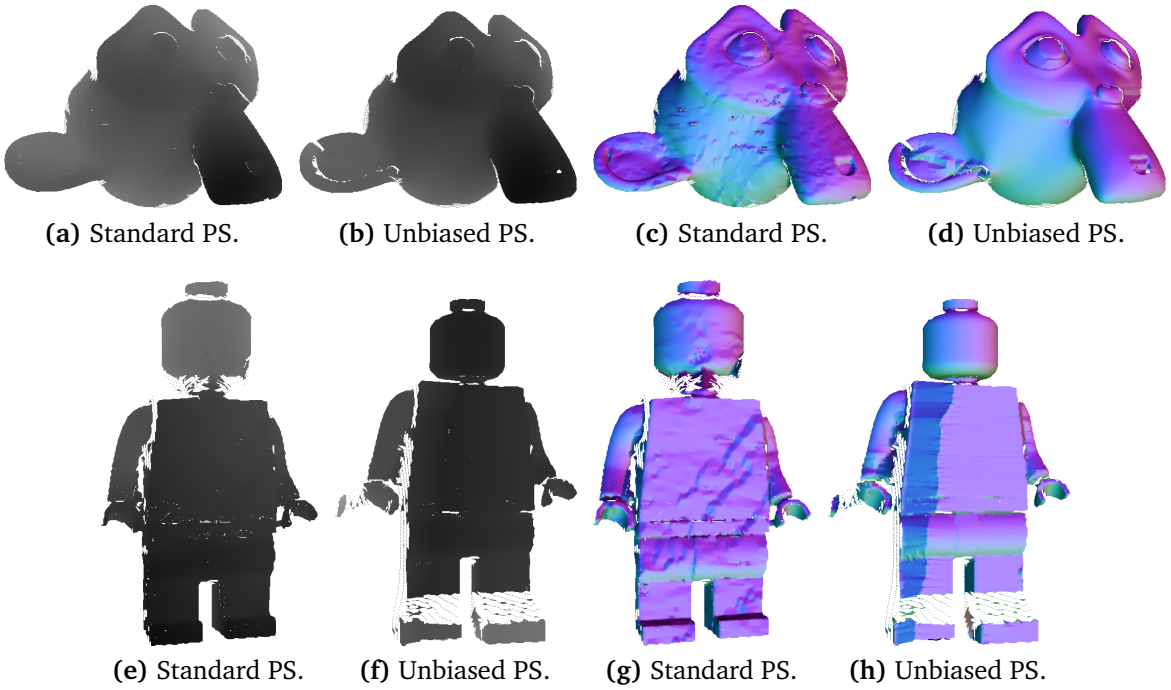
SH	standard PS		unbiased PS	
	RSE	AAE	RSE	AAE
Stanford Bunny	0.44063	9.12594	0.33271	5.58609
Dragon	0.33692	18.23377	0.45524	13.50232
Suzanne	0.26631	8.93981	0.29554	4.69646
Lego	0.33523	14.76533	0.52590	16.67052

Table 4.10: Comparison of results for the SH illumination models with non-uniform ρ.

General Lighting Model. To complete this experiment we compare the standard PS approach to the unbiased PS model under the SH illumination model.

On the Stanford Bunny we can see a clear improvement of the surface normal estimation when using the PS approach based on image ratios. Regarding the Dragon and Suzanne models the results are not really surprising. On Suzanne we obtain the lowest error values for the estimation of the surface and almost half of the AAE compared to the standard PS model with non-albedo images, which is also shown in Figure 4.13. The performance on the Dragon is once again not as good as for the other objects but still we can observe an error reduction for the computed surface normals. Unfortunately, there are some erroneous estimated areas on the Lego model that caused the unbiased PS model to produce a higher AAE and RSE than the standard model with the albedo images. The computed surface normals for the Lego object is shown in the bottom row of Figure 4.13. As for the other test the error comparison is presented in Table 4.10 for all the test models.

Overall we can say that the unbiased PS approach is able to handle the albedo images much better than the standard approach. Especially in combination with the near point light source model we obtained reconstructed surfaces with pretty low errors. But still, this extension to our model is not capable of completely cancel out the error induced by the non-uniform albedo.



(a) Standard PS. **(b)** Unbiased PS. **(c)** Standard PS. **(d)** Unbiased PS.

(e) Standard PS. **(f)** Unbiased PS. **(g)** Standard PS. **(h)** Unbiased PS.

Figure 4.13: Result images for the SH illumination models with non-uniform ρ. Objects: Suzanne (top). Lego (bottom).

4.3.4 Light Estimation

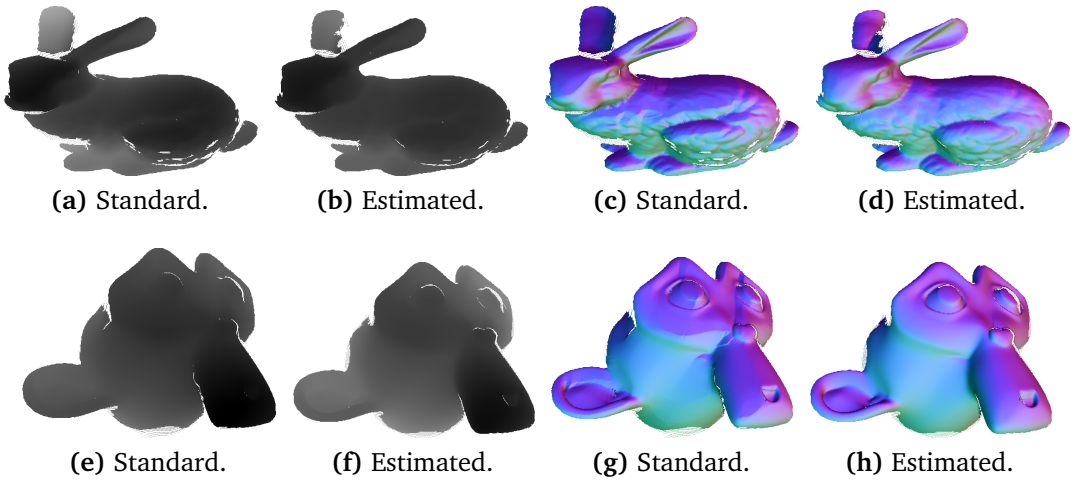
For our last experiment we evaluate the estimation of the SH coefficients l based on the estimated z. We start the *light estimation* with an estimated z after 17 resolution levels. This corresponds to a resolution of 22×11 for the Suzanne images and 11×11 for the other models.

In order to simulate a realistic environment where this approach could be applied we added a Gaussian noise on the extracted SH coefficients with $\sigma = 0.1$. The idea behind this is that the we try to improve the performance of the SH approach if the SH coefficients are not exactly known. We compare the outcomes of the standard SH approach to the results of the SH approach with estimated coefficients l in this *uncalibrated* environment and under the standard conditions. The corresponding results with 20 input images are presented in Table 4.11.

The results are listed in terms of the known error metrics AAE and RSE in Table 4.11. Considering the RSE the performance of the SH approach with light estimation is pretty similar to the standard SH approach. This holds when we use the *noisy* coefficients and the exact ones. But the computed errors also show that the outcome of our light estimation highly depends on the test object. For the Dragon and Lego model this

estimation of coefficients	SH		SH + estimation		SH noisy		SH noisy + est.	
	RSE	AAE	RSE	AAE	RSE	AAE	RSE	AAE
Stanford Bunny	0.44789	2.87659	0.46118	6.64115	0.44532	10.64780	0.44489	7.94682
Dragon	0.52200	4.51251	0.49896	19.13113	0.54191	11.06421	0.54383	22.24419
Suzanne	0.33894	0.81204	0.34104	2.53216	0.23019	8.74499	0.24435	2.49427
Lego	0.50616	1.22515	0.51902	44.29616	0.43810	17.30491	0.40805	45.42152

Table 4.11: Comparison of the SH models with standard and *noisy* coefficients.

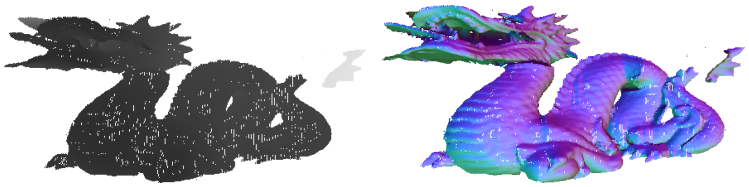


(a) Standard. **(b)** Estimated. **(c)** Standard. **(d)** Estimated.

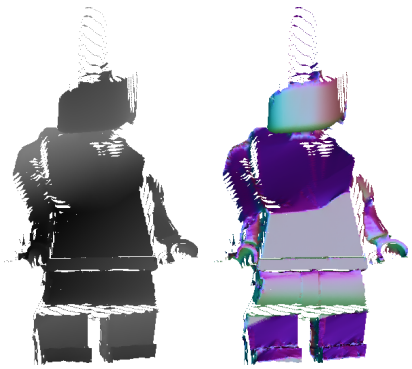
(e) Standard. **(f)** Estimated. **(g)** Standard. **(h)** Estimated.

Figure 4.14: Comparison of the SH illumination models with noisy coefficients. Objects: Stanford Bunny (top), Suzanne (bottom).

approach is not capable of providing reasonable results. Some areas on the surface of these objects cannot be reconstructed successfully as illustrated in Figure 4.14. However, for the test models with a simpler surface, namely the Stanford Bunny and Suzanne, the estimation of the surface normal works as intended. Under the uncalibrated test environment our light estimation approach is able to outperform the standard SH model. The outcomes of these reconstructions are juxtaposed in Figure 4.15.



(a) Depth. **(b)** Surface.



(c) Depth. **(d)** Surface.

Figure 4.15: Result Images for the SH illumination model with light estimation and noisy coefficients. Objects: Dragon (top), Lego (bottom)

5 Conclusion

In this thesis we presented a variational photometric stereo approach that we used with three different illumination models. Furthermore, we implemented an extension to our model based on image ratios of which we can make use of with the different illumination models. Besides, we provided an additional extension for the PS approach with the general lighting model where we tried to estimate the SH coefficients in order to obtain reasonable results if the light coefficients are not known exactly.

We started this work by giving an introduction into the topic of photometric stereo and explaining the basic idea behind this approach. In the following chapter we provided the mathematical background information for our PS model. This included the perspective camera model where the transformation between the used pixel coordinate system and the world coordinate system was explained. Further, we mentioned the basics of the Lambertian reflectance model and introduced the three illumination models that are embedded in the variational PS approach. The last point in this chapter was a brief look into the calculus of variations where the Euler-Lagrange equations were introduced which build the starting point of the minimisation of the resulting energy functional. The subsequent chapter covers the details of the variational PS models. Firstly, we addressed the parameterisation of the model components. Then, we introduced the variational SfS approach by Maurer et al. [MJBB15] since this model serves as a basis for our PS model. After we mentioned the details of the variational SfS approach we outlined the differences to our model, introduced two model extensions, and explained the minimisation technique which gets utilised in the basis SfS approach and in our PS approach. Afterwards in Chapter 4, we evaluated the different variants of our variational PS approach and compared their performances under different testing environments.

The evaluation has shown that the best performing variant overall is the PS model with the near point light source model. In the first experiment, where we tested the influence of the number of input images, it was able to outperform the models with the distant light source and the general lighting model in terms of the closeness to the ground truth surface. Furthermore, we saw that in general more input images lead to better results for all variants, with the exception that the general lighting PS model relies more on the quality of the given input images once a sufficient number of images is reached. This effect could be further investigated by using even more images. By doing this, we should

see results that are similar to the outcomes of the distant light PS model. Furthermore, we have seen that the performance on complex objects like the Dragon are overall worse but could be improved with more input images although we used a small smoothness weight α. These results imply that the importance of the smoothness term decreases when more information is provided by the higher amount of input images.

Under the influence of noise we have shown that more input images can highly improve the outcomes for the distant light PS model and the general lighting PS model. However, the PS model with near point light sources has suffered the most under this condition. Overall, the distance of the estimated surface to the ground truth surface was still close to the results from the first experiment but the estimation of the surface normal was more difficult under noise.

In the next experiment we have compared our standard PS models to the unbiased PS approach with the non-uniform albedo test images. In general, these test images were harder to handle and consequently the results under this condition were poorer for all variants. But as we have hoped, the unbiased PS approach was able to decrease the errors and produced surfaces that were much smoother than the reconstructions with the standard models. But we have also seen that the unbiased approach still struggles on the more complex test objects, especially when the SH illumination model was used which gave us the poorest results compared to the other illumination models.

The light estimation extension also has worked quite well on the simpler test objects like the Stanford Bunny and Suzanne. When we added noise to the coefficients that we want to estimate to simulate an uncalibrated testing environment, the SH model with light estimation was capable of outperforming the standard SH model. But we have to mention that this was not the case for the Dragon and Lego test objects. For a future work we could try to make this model more consistent so that it also performs well on more complex test objects. We could also implement a similar light estimation technique for the distant and near point light source models in order to improve the light directions when they are not exactly given. Furthermore, a combination of the unbiased PS approach with the light estimation technique could also be done.

Bibliography

[AA84] P. A. Davis, L. A. Soderblom. “Modeling Crater Topography and Albedo From Monoscopic Viking Orbiter Images 1. Methodology.” In: 89 (Jan. 1984), pp. 9449–9457 (cit. on p. 46).

[BBPW04] T. Brox, A. Bruhn, N. Papenberg, J. Weickert. “High Accuracy Optical Flow Estimation Based on a Theory for Warping.” In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2004, pp. 25–36. DOI: 10.1007/978-3-540-24673-2_3. URL: https://doi.org/10.1007%2F978-3-540-24673-2_3 (cit. on pp. 12, 34, 38, 39).

[CBAB97] P. Charbonnier, L. Blanc-Feraud, G. Aubert, M. Barlaud. “Deterministic edge-preserving regularization in computed imaging.” In: *IEEE Transactions on Image Processing* 6.2 (1997), pp. 298–311. DOI: 10.1109/83.551699. URL: https://doi.org/10.1109%2F83.551699 (cit. on p. 32).

[Hor75] B. K. P. Horn. “Obtaining shape from shading information.” In: *The Psychology of Computer Vision*. Ed. by P. Winston. New York: McGraw-Hill, 1975, pp. 266–290 (cit. on p. 12).

[HVC11] C. Hernandez, G. Vogiatzis, R. Cipolla. “Overcoming Shadows in 3-Source Photometric Stereo.” In: 33 (Mar. 2011), pp. 419–426 (cit. on p. 46).

[HZ04] R. I. Hartley, A. Zisserman. *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, ISBN: 0521540518, 2004 (cit. on p. 13).

[JMBB16] Y. C. Ju, D. Maurer, M. Breuß, A. Bruhn. “Direct Variational Perspective Shape from Shading with Cartesian Depth Parametrisation.” In: *Perspectives in Shape Analysis* (2016), pp. 43–72. ISSN: 2197-666X. DOI: 10.1007/978-3-319-24726-7_3. URL: http://dx.doi.org/10.1007/978-3-319-24726-7_3 (cit. on pp. 12, 22, 39, 40, 45).

[Ju17] Y. C. Ju. *PDE-based vs. variational methods for perspective shape from shading*. en. 2017. DOI: 10.18419/opus-9096. URL: http://elib.uni-stuttgart.de/handle/11682/9113 (cit. on pp. 34, 39).

[MJBB15] D. Maurer, Y. C. Ju, M. Breuß, A. Bruhn. “An Efficient Linearisation Approach for Variational Perspective Shape from Shading.” In: *Pattern Recognition* (2015), pp. 249–261. ISSN: 1611-3349. DOI: 10.1007/978-3-319-24947-6_20. URL: http://dx.doi.org/10.1007/978-3-319-24947-6_20 (cit. on pp. 3, 12, 27, 30, 34, 36, 38–40, 45, 67).

[MK96] K. Mu Lee, C.-C. J. Kuo. “Shape from Photometric Ratio and Stereo.” In: 7 (June 1996), pp. 155–162 (cit. on p. 46).

[Nav16] R. Nave. *Inverse Square Law, Light*. 2016. URL: http://hyperphysics.phy-astr.gsu.edu/hbase/Forces/isq.html#isq (cit. on p. 21).

[PF05] E. Prados, O. Faugeras. “Shape from shading: a well-posed problem?” In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 2. June 2005, 870–877 vol. 2. DOI: 10.1109/CVPR.2005.319 (cit. on p. 22).

[QMD16] Y. Quéau, R. Mecca, J.-D. Durou. “Unbiased Photometric Stereo for Colored Surfaces: A Variational Approach.” In: (Jan. 2016) (cit. on p. 46).

[RH01] R. Ramamoorthi, P. Hanrahan. “An Efficient Representation for Irradiance Environment Maps.” In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '01. New York, NY, USA: ACM, 2001, pp. 497–500. ISBN: 1-58113-374-X. DOI: 10.1145/383259.383317. URL: http://doi.acm.org/10.1145/383259.383317 (cit. on pp. 22, 49).

[RT92] E. Rouy, A. Tourin. “A Viscosity Solutions Approach to Shape-From-Shading.” In: *SIAM Journal on Numerical Analysis* 29.3 (June 1992), pp. 867–884. DOI: 10.1137/0729053. URL: https://doi.org/10.1137%2F0729053 (cit. on p. 34).

[SHW15] C. Schroers, D. Hafner, J. Weickert. “Multiview Depth Parameterisation with Second Order Regularisation.” In: *Lecture Notes in Computer Science*. Springer International Publishing, 2015, pp. 551–562. DOI: 10.1007/978-3-319-18461-6_44. URL: https://doi.org/10.1007%2F978-3-319-18461-6_44 (cit. on p. 28).

[Sze10] R. Szeliski. *Computer Vision: Algorithms and Applications*. 1st. New York, NY, USA: Springer-Verlag New York, Inc., 2010. ISBN: 1848829345, 9781848829343 (cit. on pp. 13, 18).

[TK05] A. Tankus, N. Kiryati. “Photometric stereo under perspective projection.” In: *In Proc. 10 th IEEE International Conference on Computer Vision*. 2005, pp. 611–616 (cit. on p. 12).

[Woo80] R. J. Woodham. “Photometric Method For Determining Surface Orientation From Multiple Images.” In: *Optical Engineering* 19.1 (Feb. 1980). ISSN: 0091-3286. DOI: 10.1117/12.7972479. URL: http://dx.doi.org/10.1117/12.7972479 (cit. on p. 11).

[WZN+14] C. Wu, M. Zollhöfer, M. Nießner, M. Stamminger, S. Izadi, C. Theobalt. “Real-time Shading-based Refinement for Consumer Depth Cameras.” In: *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2014)*. Vol. 33. 6. Dec. 2014 (cit. on p. 23).

[You54] D. Young. “Iterative Methods for Solving Partial Difference Equations of Elliptic Type.” In: *Transactions of the American Mathematical Society* 76.1 (Jan. 1954), p. 92. DOI: 10.2307/1990745. URL: https://doi.org/10.2307%2F1990745 (cit. on pp. 43, 45).

All links were last followed on September 18, 2017.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

---

place, date, signature