# Tesi di Dottorato

Silvia Tozza

## Analysis and Approximation of Non-Lambertian Shape-from-Shading Models

*Dottorato in Matematica*, Roma «La Sapienza» (2014).

<http://www.bdim.eu/item?id=tesi_2014_TozzaSilvia_1>

# Analysis and Approximation of Non-Lambertian Shape-from-Shading Models

**Advisor:**
**Chiar.mo Prof.**
**Maurizio Falcone**

**Candidate:**
**Silvia Tozza**
matricola 1276585

# Acknowledgments

# Contents

# Introduction

## Context and Motivation

The three dimensional reconstruction of an object is a topic of great interest in many different fields of application: from the biomedical application of 3D reconstruction of human tissues, to the archeology for the digitization of artistic works. Other examples come from engineering for the construction of an endoscope 3D for surgical operations in real time and from astronomy for the characterization of properties of planets or other astronomical entities. More recently the application to security have emerged for the facial recognition of individuals. This problem has always attracted a great attention because although its formulation is simple, there is still no global method for its resolution under realistic assumptions but it finds applications in various fields of application as listed just above.

The firsts being interested in the problem were some opticians [33, 96] in the Fifties-Sixties. Afterwards, Horn [50, 51, 53, 54, 52] first formulated this problem, known in the literature as the Shape from Shading (SfS) problem, via a partial differential equation (PDE), giving rise to an expansion in the field of mathematics as well, upon which others researchers gave theoretical contributions. In fact, starting e.g. from the first works of Lions, Rouy and Tourin [97, 75], the SfS problem was inserted in the context of the viscosity solutions frameworks, hence in a much more theoretical area [81].

This classical problem of image processing is a difficult inverse problem where the goal is to reconstruct the surface from a single image, that is, given a two-dimensional gray scale digital image $I(\mathbf{x})$, where $\mathbf{x} := (x, y)$, the goal is to reconstruct the surface $z = u(\mathbf{x})$ that corresponds to it, using the information contained in the image (hence the name *Shape from Shading*). This problem is described in general by the irradiance equation

$$R(\mathbf{N}(\mathbf{x})) = I(\mathbf{x}) \tag{0.1}$$

where the normalized brightness of the given grey-value image $I(\mathbf{x})$ is put in relation with the function $R(\mathbf{N}(\mathbf{x}))$ that represents the reflectance map giving the value of the light reflection on the surface as a function of its orientation (i.e., of the normal $\mathbf{N}(\mathbf{x})$) at each point $(\mathbf{x}, u(\mathbf{x}))$. Depending on how we describe the function $R$ different reflection models are determined. In the literature, the most common representation of $R$ takes into account only the angle between the outgoing normal to the surface $\mathbf{N}$ and the light source $\boldsymbol{\omega}$, that is

$$I(\mathbf{x}) = \gamma_D(\mathbf{x})\mathbf{N}(\mathbf{x}) \cdot \boldsymbol{\omega}, \tag{0.2}$$

where $\gamma_D(\mathbf{x})$ indicates the albedo of the diffuse part, i.e. the diffuse reflectivity or reflecting power of a surface. It is the ratio of reflected radiation from the surface to incident

radiance upon it. Its dimensionless nature is expressed as a percentage and is measured on a scale from zero for no reflection of a perfectly black surface to 1 for perfect reflection of a white surface. The data are the grey-value image $I(\mathbf{x})$, the direction of the light source $\boldsymbol{\omega}$ and the albedo $\gamma_D(\mathbf{x})$. The light source $\boldsymbol{\omega}$ is a unit vector, hence $|\boldsymbol{\omega}| = 1$.

Several approaches have been proposed to solve this problem that can be collect in two big classes (see the survey [34]): methods of resolution of partial differential equations (PDEs), in particular the method of characteristics [33, 96, 51, 80, 79, 14, 71] and the approximation of viscosity solutions (see [57, 18, 20, 41, 19, 75, 92, 8, 98, 60, 42, 89]), and optimization methods based on the variational approach (see [53, 72, 31, 105, 44, 106, 55, 15, 121]). In this thesis we will follow the differential approach, that leads to a nonlinear partial differential equation of the first order (Hamilton-Jacobi equation). It is well known that, in general, this problem is ill-posed since there can be many viscosity solutions (no matter which regularity is required for the solutions) unless additional conditions/informations are added to the problem [124, 34]. This explains the growing importance of a generalization of this classical problem in order to obtain uniqueness of the solution while reducing the assumptions on the physical reflectance properties of the objects. A continuous effort has been made by the scientific community to take into account more realistic scenarios including perspective deformations [78, 109, 90, 3, 76] and/or multiple images of the same object [122, 123], taken from the same point of view but with different light sources (photometric stereo technique, see [107, 70]) or from different points of view but with the same light source (stereoscopy, see [21]). We can consider in addition other supplementary issues, as the estimation of the albedo or of the direction of the light source (the so-called calibrated/uncalibrated problem), that are known quantities for the model but not provided in the real images. Moreover, the boundary conditions that we need to know in real applications consists to know a priori the height of the surface on the boundary of the image. This is a supplementary data that it is possible to approximate using the classical (not differential) approach of photometric stereo, following the seminal paper by Woodham [120], aims at reconstructing the surface through its normal field approximation and integration.

Our initial effort goes in another direction: we want to take into account more realistic models considering generic surfaces with not uniform reflection properties, in such a way that the light intensity of the image does not depend only on the angle between the outgoing normal to the surface and the light source (such a model is said *the Lambertian model*). We will see how this problem does not admit a unique solution due to the so-called concave/convex ambiguity. This ambiguity is common to the two approaches, i.e. methods based on partial differential equation or minimization methods (see [34]).

## Contributions of this thesis

The research work is focused on the analysis and the approximation of a problem related to image processing, namely the Shape-from-Shading (SfS) problem. Our goal is to overcome the classical Lambertian model, which reflects the light uniformly over the entire surface supposed smooth, without taking into account the position of the observer or

the physical properties of the material of the object to be reconstructed. We also would like to overcome the concave/convex ambiguity typical of the SfS problem. In the course of this research work we have proposed a unified formulation of a general model for the orthographic case, that is

$$I(\mathbf{x}) = k_A I_A(\mathbf{x}) + k_D I_D(\mathbf{x}) + k_S I_S(\mathbf{x}), \tag{0.3}$$

where $I_A(\mathbf{x})$, $I_D(\mathbf{x})$ and $I_S(\mathbf{x})$ are respectively the ambient, diffuse reflected and specular reflected light components and $k_A$, $k_D$ and $k_S$ indicate the percentages of these components such that their sum is equal to 1 (we do not consider absorption phenomena). Note that the diffuse or specular albedo is inside the definition of $I_D(\mathbf{x})$ or $I_S(\mathbf{x})$, respectively. This formulation can describe different models of reflection, specular or diffuse, in a single notation and mathematical formulation that, using the various components such as a switch on-off, finds the different models. We analyzed with particular attention two reflectance models alternative to the classical Lambertian one: the first was the diffuse model introduced by Oren and Nayar [83, 84, 82, 85], which irradiance equation in the simplified version suggested by the authors themselves in the Section 4.4 of [83] is

$$E_D(\mathbf{x}) = \gamma_D(\mathbf{x})\cos(\theta_i)(A + B\sin(\alpha)\tan(\beta)\max[0,\cos(\varphi_r - \varphi_i)]). \tag{0.4}$$

Assuming that there is a linear relation between the irradiance of the image and the image intensity, the $I_D$ *brightness equation* for the ON–model is given by

$$I_D(\mathbf{x}) = \gamma_D(\mathbf{x})\cos(\theta_i)(A + B\sin(\alpha)\tan(\beta)\max[0,\cos(\varphi_r - \varphi_i)]) \tag{0.5}$$

where

$$A = 1 - 0.5\,\sigma^2(\sigma^2 + 0.33)^{-1} \tag{0.6}$$
$$B = 0.45\sigma^2(\sigma^2 + 0.09)^{-1}. \tag{0.7}$$

Note that $A$ and $B$ are two non negative constants depending on the statistics of the cavities via the roughness parameter $\sigma$ that we can imagine that takes values between 0 and $\pi/2$, as representative of the slope of the roughness of the surface considered. In this model $\theta_i$ represents the angle between the unit normal to the surface $\mathbf{N}(\mathbf{x})$ and the light source direction $\boldsymbol{\omega}$, $\theta_r$ stands for the angle between $\mathbf{N}(\mathbf{x})$ and the observer direction $\mathbf{V}$, $\varphi_i$ is the angle between the projection of the light source direction $\boldsymbol{\omega}$ and the $x_1$ axis onto the $(x_1, x_2)$-plane, $\varphi_r$ denotes the angle between the projection of the observer direction $\mathbf{V}$ and the $x_1$ axis onto the $(x_1, x_2)$-plane and the two variables $\alpha$ and $\beta$ are given by

$$\alpha = \max[\theta_i, \theta_r] \text{ and } \beta = \min[\theta_i, \theta_r]. \tag{0.8}$$

The second model analyzed was the specular model proposed by Phong [86], which brightness equation is

$$I(\mathbf{x}) = k_A I_A(\mathbf{x}) + k_D \gamma_D(\mathbf{x})(\mathbf{N}(\mathbf{x}) \cdot \boldsymbol{\omega}) + k_S \gamma_S(\mathbf{x})(\mathbf{R}(\mathbf{x}) \cdot \mathbf{V})^\alpha, \tag{0.9}$$

where the diffuse component is represented by the Lambertian model and the specular light component $I_S(\mathbf{x})$ is described as a power of the cosine of the angle between the unit vectors $\mathbf{V}$ and $\mathbf{R}(\mathbf{x})$ (it is the vector representing the reflection of the light $\boldsymbol{\omega}$ on the surface). The exponent $\alpha$ expresses the specular reflection characteristics of a material. A careful analysis of the parameters involved in these models was carried out.

Using a differential approach it is possible to associate partial differential equations (PDEs) to the equations of these models: in the vertical light case, we noted that it is possible to arrive to an eikonal equation form for each model

$$|\nabla u(\mathbf{x})| = f(\mathbf{x}) \quad \text{for} \quad \mathbf{x} \in \Omega, \tag{0.10}$$

where $\Omega$ is an open subset of $\mathbb{R}^2$ and the expression of $f(\mathbf{x})$ changes for each model. With respect to the oblique light case, after a Kruzkov transform, we were able to write the problem in the new variable $v$ in a similar way for all the models as follows:

$$\mu v(\mathbf{x}) + \max_{a \in \partial B_3} \{-b^M(\mathbf{x}, a) \cdot \nabla v(\mathbf{x}) + P^M(\mathbf{x}, z)(1 - \mu v(\mathbf{x}))\} = 1, \qquad \mathbf{x} \in \Omega, \tag{0.11}$$

where the vector field $b^M(\mathbf{x}, a)$ and the function $P^M(\mathbf{x}, z)$ have a different expression for each model ($M$ indicates the acronym of the specific model considered).

Rewriting this problem in a fixed point form $V = T^M(V)$, we discretized it by constructing a semi-Lagrangian scheme associated to it. We were able to write the discrete operator associated to the different models in a similar manner as

$$T_i^M(W) = \min_{a \in \partial B_3} \{e^{-\mu h} w(x_i + hb^M(x_i, a)) - \tau P^M(x_i, z)a_3(1 - \mu w(x_i))\} + \tau, \tag{0.12}$$

where

$$\tau = 1 - e^{-\mu h}/\mu \tag{0.13}$$

$$P^M : \Omega \times \mathbb{R} \to \mathbb{R} \tag{0.14}$$

and $w(x_i + hb^M(x_i, a))$ is obtained by interpolating on $W$, that denotes the vector solution giving the approximation of the height of $u$ at every node $x_i$ of the grid.

Note that $\mu$ is a free positive parameter without a specific physical meaning, but it is important because varying its value it is possible to modify the slope. In fact, the slope increases for increasing values of $\mu$.

Thanks to this general form of the discrete operator, we came to the formulation of a single general convergence theorem, then checked for individual models that the properties assumed by the general theorem were actually respected by individuals and different operators.

Numerical tests to compare the performances of the two models with the classical Lambertian model were done. In addition, we applied acceleration techniques based on a finite difference solver (Fast Marching and Fast Sweeping methods) and compared the performances of them too.

We have also studied the perspective case based on the reflectance model of Oren-

Nayar. In the literature it is always assumed that the light source is located exactly at the optical center of the camera, a restrictive and unrealistic assumption (think for example to the flash of a camera or to an endoscope), see e.g. Ahmed and Farag [3, 2] or Prados and Faugeras [90, 89] using this restrictive assumption. Starting from a perspective model, we have reached a formulation of a generalized model, using a parameterization of the surface in spherical coordinates, which allows us to locate the light source at *any point in the scene*. The switching from Cartesian coordinates to spherical coordinates is useful and crucial to ensure that the points at maximum brightness coincide with the critical points, an essential condition for the application of the Fast Marching method. Because of the *min* and *max* operators involved in the Oren-Nayar model, it is necessary to distinguish several cases that have been analyzed and implemented in [59].

It is still in progress the analysis and approximation of non-linear systems corresponding to the application of the photometric stereo technique to non-Lambertian models (Cf. [23, 128, 47]). This technique considers more than one input image taken from the same point of view but from different light sources, an approach which has been successfully applied for Lambertian model (see [76]). Note that this approach is able to overcome the concave/convex ambiguity typical of the SfS problem: in fact, it ensures uniqueness of the solution with only two images, that give rise to two partial differential equations, with boundary conditions. If we consider three images the uniqueness is guarantee without the use of boundary conditions (see details in [77]). On the heels of that success in the Lambertian case, we hope to achieve similar results to more realistic models, i.e. non-Lambertian models.

## Organization of this thesis

In Chapter 1 we give a description of some non-Lambertian reflectance models, after a very short explanation of the classical Lambertian model. In particular, we focus our attention on the diffuse Oren-Nayar model, the specular Phong model and the Blinn-Phong model. We begin the chapter by formulating the problem, listing the assumptions that we will follow in the orthographic case and we illustrate a new general mathematical formulation that includes the different models in a unique notation. In this way, we can switch on-off the different components (ambient, diffuse and specular components) and we can retrieve the Lambertian model in some special case. We observe that in the particular case of vertical position of the light source, it is possible to rewrite the PDEs associated to each model in an eikonal form. After that, we recall some definitions and concepts related to the viscosity solutions and their theory (existence and uniqueness results). Then, we focus our attention on two methods of resolutions of PDE: characteristic strips expansion and approximation of viscosity solutions. Finally, a subsection on the issue of boundary conditions is given.

In Chapter 2 we begin by describing two class of numerical schemes for the solution of the PDEs illustrated in Chapter 1. We focus our attention on the Finite Difference (FD) approximation and on the semi-Lagrangian (SL) approach, describing the main characteristics of each technique. After that, a new general convergence theorem is stated

for a general discrete fixed point operator. We list and prove the properties of this general operator. In this way, we show that the general operator is a contraction mapping under some assumptions and we will see in the next Chapters 4 and 5 that the discrete operators associated to the Oren-Nayar model and to the Phong model described in Chapter 1 satisfy the properties listed here.

In Chapter 3 we move on the description of acceleration methods, as the Fast Marching (FM) and the Fast Sweeping (FS) methods, describing how they work in a section devoted to each of them, showing the algorithms in pseudocode too.

Chapter 4 is devoted to the Oren-Nayar model. An analysis of the parameter $\sigma$ involved in the model is given. Then, we show how the discrete operator $T^{ON}$ of the SL scheme for this model has the properties stated in the general convergence theorem of the Chapter 2. Finally, some numerical tests are shown on synthetic and real input images. We give tables of errors and images of the results obtained. We conclude the chapter with some final comments.

Chapter 5 deals with the analysis and the approximation of the Phong model. The structure of this chapter is similar to that of Chapter 4: we give an analysis of the parameters $\alpha$ and $k_S$ involved in the model, we show how the discrete operator $T^{PH}$ has the properties stated in the general convergence theorem of the Chapter 2, we show some numerical tests on synthetic and real input images, accompanied by tables of errors too. Finally, a section on the comparison between the Oren-Nayar model, the Lambertian model and the Phong model is given, with some final remarks.

In Chapter 6 we move to consider the perspective case formulated via PDEs, by combining a spherical surface parametrization with the non-Lambertian Oren-Nayar reflectance model. We present a new general model that allows to deal with an arbitrary position of the light source in the scene, not located at the optical center of the camera that is the common assumption in the literature. We explain why we choose a spherical coordinate system, we parametrize the surface in spherical coordinates and then we describe the normal vector in spherical coordinates. We present the perspective Oren-Nayar SfS brightness equation and we derive its formulation in spherical coordinates for the several cases typical of the Oren-Nayar model. Afterwards, we describe the Fast Marching method that we use to solve the resulting set of Hamilton-Jacobi equations (HJEs). Finally, we show some numerical results obtained by using endoscopic images provided in [1].

Appendix A contains supplementary materials related to Chapter 6, showing in details intermediate steps that are helpful to understand the derivation of the normal vector and the four cases of the HJEs that correspond to the generalized perspective SfS model with Oren-Nayar reflectance described in Chapter 6.

## Original material for this thesis

Part of the Chapter 1, Chapter 4 and 5 are an extended version of the paper [TF14b]. The complete proof of the general convergence theorem and a version similar in extension to the material present in the thesis will appear in [TF15].

Chapter 4 is based on the refereed paper [TF14a], presented at the 9th International Conference on Computer Vision Theory and Applications (VISAPP).

Chapter 6 is based on the refereed paper [JTBBK13], presented at the 24th British Machine Vision Conference (BMVC).

## List of publications

[MT13]  R. Mecca, S. Tozza, *Shape Reconstruction of Symmetric Surfaces using Photometric Stereo*, In: Michael Breuss, Alfred Bruckstein, Petros Maragos (Editors), Innovations for Shape Analysis: Models and Algorithms, pages 219–243, Springer Edition, 2013 (ISBN: 978-3-642-34140-3).

[JTBBK13]  Y.C. Ju, S. Tozza, M. Breuß, A. Bruhn, A. Kleefeld, *Generalised Perspective Shape from Shading with Oren-Nayar Reflectance*. In Proc. 24th British Machine Vision Conference (BMVC 2013, Bristol, UK, September 2013), BMVA Press 2013, pages 42.1–42.11 (http://dx.doi.org/10.5244/C.27.42).

[TF14a]  S. Tozza, M. Falcone, *A Semi-Lagrangian Approximation of the Oren-Nayar PDE for the Orthographic Shape–from–Shading Problem*, In: Sebastiano Battiato and José Braz, editors, Proceedings of the 9th International Conference on Computer Vision Theory and Applications (VISAPP), pages 711–716, Vol. 3, SciTePress, 2014.

[BT13]  L.T.T. Bui, S. Tozza, *On the Cahn-Hilliard regularization of forward-backward parabolic equations*, Dipartimento di Matematica, Sapienza - Università di Roma, preprint 2013 submitted to the International Journal "Advances in Mathematical Sciences and Applications".

[TF14b]  S. Tozza, M. Falcone, *A comparison of non-Lambertian models for the Shape–from–Shading problem*, submitted to Michael Breuss, Alfred Bruckstein, Petros Maragos, Stefanie Wuhrer (Editors), New Perspectives in Shape Analysis, Springer Edition.

[MTDDb14]  R. Mecca, S. Tozza, M. Duocastella, A. Del Bue, *Three Lights Photometric Stereo from Diffuse and Specular Components*, submitted to the 28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015).

[TMDDb14]  S. Tozza, R. Mecca, M. Duocastella, A. Del Bue, *Direct differential Photometric-Stereo shape recovery of diffuse and specular surfaces*, in preparation.

[TF15]  S. Tozza, M. Falcone, *Analysis and approximation of some Shape–from-Shading models for non-Lambertian surfaces*, in preparation.

# List of Figures

# List of Tables

# 1. A review of some non-Lambertian models

We want to take into account more realistic models considering generic surfaces with reflection properties not uniform, in such a way that the light intensity of the image does not depend only on the angle between the outgoing normal to the surface and the light source (such a model is said *the Lambertian model*). With this purpose, in this chapter, after a formulation of the problem and its assumptions, and a description of the classical Lambertian model, of course, we will describe briefly several non-Lambertian reflectance models. Our initial effort was to describe these different models in a single formulation and notations unified which will help in their analysis and approximation.

## 1.1. Formulation of the general model

We fix a camera in a three-dimensional coordinate system ($Oxyz$) in such a way that $Oxy$ coincides with the image plane and $Oz$ with the optical axis. Let $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3) = (\tilde{\boldsymbol{\omega}}, \omega_3) \in \mathbb{R}^3$ (with $\omega_3 > 0$) be the unit vector that represents the direction of the light source; let $I(\mathbf{x})$ be the function that measures the gray tones of the image at the point $\mathbf{x} := (x, y)$. $I(\mathbf{x})$ is the datum in the model since it is measured at each pixel of the image, for example in terms of a greylevel (from 0 to 255). In order to construct a continuous model, we will assume that $I(\mathbf{x})$ takes real values in the interval $[0, 1]$, defined in a compact domain $\overline{\Omega}$ called "reconstruction domain" (with $\Omega \subset \mathbb{R}^2$ open set), $I : \overline{\Omega} \to [0, 1]$, where the points with a value of 0 are the dark point (blacks), while those with a value of 1 correspond to a completely reflection of the light (white dots, with a maximum reflection).

We consider the following *assumptions*:

**A1.** there is a single light source placed at infinity in the direction $\boldsymbol{\omega}$ (the light rays are, therefore, parallel to each other);

**A2.** the observer's eye is placed at an infinite distance from the object you are looking at (i.e. there is no perspective deformation);

**A3.** there are no autoreflections on the surface.

In addition to these assumptions, there are other hypothesis that depend on the different reflectance models (we will see them in the description of the individual models).
Being valid the assumption ($A2$) of orthographic projection, the visible part of the scene

is a graph $z = u(\mathbf{x})$ and the unit normal to the regular surface at the point corresponding to $\mathbf{x}$ is given by:

$$\mathbf{N}(\mathbf{x}) = \frac{\mathbf{n}(\mathbf{x})}{|\mathbf{n}(\mathbf{x})|} = \frac{(-\nabla u(\mathbf{x}), 1)}{\sqrt{1 + |\nabla u(\mathbf{x})|^2}}, \qquad (1.1)$$

where $\mathbf{n}(\mathbf{x})$ is the outgoing normal vector.

We assume that the height function, which is the unknown of the problem, is $u(\mathbf{x}) \geq 0$ and the surface is standing on a flat background, we will denote by $\Omega$ the region inside the silhouette and we will assume (just for technical reasons) that $\Omega$ is an open and bounded subset of $\mathbb{R}^2$ (see Fig. 1.1). It is well known that the Shape-from-Shading (SfS)
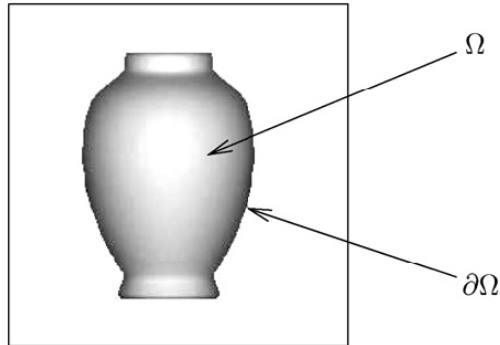


Figure 1.1.: An object on a flat background: $\Omega$ indicates the region inside the silhouette, $\partial\Omega$ the boundary of it.

problem is described by the image irradiance equation

$$I(\mathbf{x}) = R(\mathbf{N}(\mathbf{x})), \qquad (1.2)$$

where $I(\mathbf{x})$ is the normalized brightness of the given grey-value image, $\mathbf{N}(\mathbf{x})$ is the unit normal to the surface at the point $(\mathbf{x}, u(\mathbf{x}))$ and $R(\mathbf{N}(\mathbf{x}))$ is the reflectance map giving the value of the light reflection on the surface as a function of its orientation (i.e., of the normal) at each point. Depending on how we describe the function $R$ different reflection models are determined. We describe below some of them. Note that a more general formulation of the reflectance function $R$ present in the irradiance equation (1.2) consists of adding a dependence on $\mathbf{x}$ too, in order to include several features like e.g. non uniform ambient light depending on some diffuse lights in the ambient (that can be generated by other light sources at finite distance). We will not consider this generalization in this thesis. For the analysis of the different models, it would be useful to introduce a representation of the brightness function $I(\mathbf{x})$ where we can distinguish different terms representing the contribution of ambient, diffuse reflected and specular reflected light. We will write then

$$I(\mathbf{x}) = k_A I_A(\mathbf{x}) + k_D I_D(\mathbf{x}) + k_S I_S(\mathbf{x}), \qquad (1.3)$$

where $I_A(\mathbf{x})$, $I_D(\mathbf{x})$ and $I_S(\mathbf{x})$ are respectively the above mentioned components and $k_A$, $k_D$ and $k_S$ indicate the percentages of these components such that their sum is equal to 1 (we do not consider absorption phenomena). Note that the diffuse or specular albedo is inside the definition of $I_D(\mathbf{x})$ or $I_S(\mathbf{x})$, respectively. In the sequel, we will always consider $I(\mathbf{x})$ normalized in $[0, 1]$. This will allow to switch on and off the different contributions depending on the model. Let us note that the ambient light term $I_A(\mathbf{x})$ represents light everywhere in a given scene. As we will see in the following sections, the intensity of diffusely reflected light in each direction is proportional to the cosine of the angle $\theta_i$ between surface normal and light source direction, without taking into account the point of view of the observer, but another diffuse model (the Oren–Nayar model) will consider it in addition. The amount of specular reflected light towards the viewer is proportional to $(\cos\theta_s)^\alpha$, where $\theta_s$ is the angle between the ideal (mirror) reflection direction of the incoming light and the viewer direction, $\alpha$ being a constant modelling the specularity of the material. But another specular model will consider an intermediate vector between the light source and the viewer, representing the intensity of specular reflected light in proportion to the cosine of the angle between surface normal and this intermediate vector. In this way we have a more general model and, dropping the ambient and specular component, we retrieve the Lambertian reflection as a special case.

## 1.2. The Lambertian model (L–model)

For a *Lambertian surface*, which generates a purely diffuse model, the specular component does not exist. So, the general equation (1.3) becomes

$$I(\mathbf{x}) = k_A I_A(\mathbf{x}) + k_D I_D(\mathbf{x}), \tag{1.4}$$

whose diffuse component $I_D(\mathbf{x})$ is

$$I_D(\mathbf{x}) = \gamma_D(\mathbf{x})\,\mathbf{N}(\mathbf{x})\cdot\boldsymbol{\omega}, \tag{1.5}$$

where $\gamma_D(\mathbf{x})$ is the diffuse albedo. Neglecting the ambient component that can be considered as a constant (i.e. setting $k_A = 0$), recalling that the sum $k_A + k_D + k_S$ must be equal to 1, we obtain that necessarily $k_D = 1$ and we can omit it in the following. Then, for a Lambertian surface the irradiance equation (1.2) becomes

$$I(\mathbf{x}) = \gamma_D(\mathbf{x})\,\mathbf{N}(\mathbf{x})\cdot\boldsymbol{\omega}, \tag{1.6}$$

where we assume to know $\gamma_D(\mathbf{x})$ (in the sequel we suppose uniform albedo and we put $\gamma_D(\mathbf{x}) = 1$, that is all the points of the surface reflect completely the light that hits them).

Under these assumptions, the orthogonal SfS problem consists in determining the function $u : \overline{\Omega} \to \mathbb{R}$ that satisfies the equation (1.6). The unit vector $\boldsymbol{\omega}$ and the function $I(\mathbf{x})$ are the only quantity known.

In this model, we can note that the measured light in the image depends only on the scalar product between $\mathbf{N}(\mathbf{x})$ and $\boldsymbol{\omega}$ and the parameter $\gamma_D(\mathbf{x})$, which describes the physical properties of the surface reflection. So in this case the surface of the object has uniform properties of light reflection (see Fig. 1.2). We will see in the next sections that
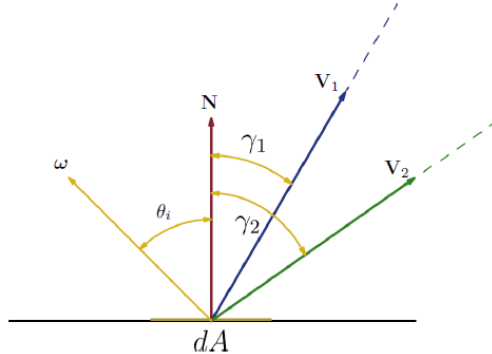


Figure 1.2.: A Lambertian surface diffuses the incident radiation independently from the angle between the surface normal $\mathbf{N}$ and the direction of the observer. Two different observers $\mathbf{V}_1$ and $\mathbf{V}_2$ do not detect any difference in radiance. The radiance is a function only of the angle $\theta_i$ between $\boldsymbol{\omega}$ and $\mathbf{N}$.

for other models it is no longer true because other quantities are taken into account.
For Lambertian surfaces [53, 54], just considering an orthographic projection of the scene, it is possible to model the SfS problem via a nonlinear PDE of the first order which describes the relation between the surface $u(\mathbf{x})$ (our unknown) and the brightness function $I(\mathbf{x})$. In fact, recalling the definition of the unit normal to a graph given in (1.1), we can write (1.6) as

$$I(\mathbf{x})\sqrt{1 + |\nabla u(\mathbf{x})|^2} + \widetilde{\boldsymbol{\omega}} \cdot \nabla u(\mathbf{x}) - \omega_3 = 0, \text{ in } \Omega \tag{1.7}$$

where $\widetilde{\boldsymbol{\omega}} = (\omega_1, \omega_2)$. This is an Hamilton-Jacobi type equation which does not admit in general regular solution. It is known that the mathematical framework to describe its weak solutions is the theory of viscosity solutions as in [75].

**The vertical light case.**
If we choose $\boldsymbol{\omega} = (0, 0, 1)$, the equation (1.7) becomes the so-called "eikonal equation":

$$|\nabla u(\mathbf{x})| = f(\mathbf{x}) \quad \text{for} \quad \mathbf{x} \in \Omega, \tag{1.8}$$

where

$$f(\mathbf{x}) = \sqrt{\frac{1}{I(\mathbf{x})^2} - 1}. \tag{1.9}$$

Points $\mathbf{x} \in \Omega$ such that $I(\mathbf{x})$ assume maximum value correspond to the case in which $\boldsymbol{\omega}$ and $\mathbf{N}(\mathbf{x})$ have the same direction: these points are usually called "singular points".

In order to make the problem well-posed, we need to add boundary conditions to the equations (1.7) or (1.8): they can require the value of the solution $u$ (Dirichlet boundary conditions type), or the value of its normal derivative (Neumann boundary conditions), or an equation that must be satisfied on the boundary (the so-called boundary conditions "state constraint"). The current choice in our work is to consider the Dirichlet boundary conditions equal to zero, assuming the surface on a flat background

$$u(\mathbf{x}) = 0, \quad \text{for} \quad \mathbf{x} \in \partial\Omega, \tag{1.10}$$

but we will see in the numerical tests that a second possibility of the same type occurs when it is known the value of $u$ on the boundary, which leads to the more general condition

$$u(\mathbf{x}) = g(\mathbf{x}), \quad \text{for} \quad \mathbf{x} \in \partial\Omega. \tag{1.11}$$

We will focus on the analysis of boundary conditions in the Section 1.7.4 of this chapter. But adding a boundary condition to the PDE that describes the SfS model is not enough to obtain a unique solution because of the concave/convex ambiguity. We will explore better this issue in the Section 1.7.

For analytical and numerical reasons it is useful to introduce the exponential transform $\mu v(\mathbf{x}) = 1 - e^{-\mu u(\mathbf{x})}$ and change the variable. Note that here $\mu$ is a free positive parameter without a physical meaning, but it is important because varying its value it is possible to modify the slope. In fact, the slope increases for increasing values of $\mu$. Assuming that the surface is standing on a flat background and following [42], we can write (1.7)+(1.10) in a fixed point form in the new variable $v$ obtaining the

---

*Lambertian Model*

$$\begin{cases} \mu v(\mathbf{x}) = \min_{a \in \partial B_3} \{b^L(\mathbf{x}, a) \cdot \nabla v(\mathbf{x}) + f^L(\mathbf{x}, a, v(\mathbf{x}))\} & \text{for } \mathbf{x} \in \Omega, \\ v(\mathbf{x}) = 0 & \text{for } \mathbf{x} \in \partial\Omega, \end{cases} \tag{1.12}$$

---

where $b^L : \Omega \times \partial B_3(0,1) \to \mathbb{R}^2$ and $f^L : \Omega \times \partial B_3(0,1) \times [0,1] \to \mathbb{R}$ are defined as

$$b^L(\mathbf{x}, a) = \frac{1}{\omega_3} \left( I(\mathbf{x})a_1 - \omega_1, I(\mathbf{x})a_2 - \omega_2 \right), \tag{1.13}$$

$$f^L(\mathbf{x}, a, v(\mathbf{x})) = -\frac{I(\mathbf{x})a_3}{\omega_3}(1 - \mu v(\mathbf{x})) + 1 \tag{1.14}$$

and $B_3$ is the unit ball in $\mathbb{R}^3$.

## 1.3. The Oren-Nayar model (ON–model)

The diffuse reflectance ON–model [83, 84, 82, 85] is an extension of the previous L–model which explicitly allows to handle *rough* surfaces. The idea of this model is to represent a rough surface as an aggregation of V-shaped cavities, each with Lambertian reflectance properties (see Fig. 1.3).

In [83] and, with more details, in [85], Oren and Nayar derive a reflectance model for three types of surfaces with different slope-area distributions:

- **Uni-directional Single-Slope Distribution:** This distribution results in a non-isotropic surface where all facets have the same slope and all cavities are aligned in the same direction.

- **Isotropic Single-Slope Distribution:** Here, all facets have the same slope but they are uniformly distributed in orientation on the surface plane.

- **Gaussian Distribution:** This is the most general case examined where the slope-area distribution is assumed to be normal with zero mean. The roughness of the surface is determined by the standard deviation of the normal distribution.

The reflectance model obtained for each of the above surface types is used to derive the succeeding one, called by the authors the "Qualitative Model", a simpler version obtained by discarding the coefficient $C_3$ present in their formulation and ignoring interreflections (see Section 4.4 of [83] for more details).
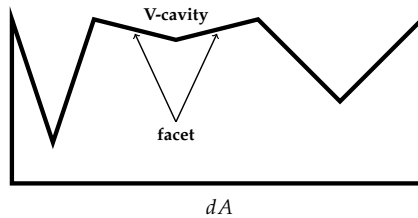


Figure 1.3.: Facet model for surface patch $dA$ consisting of many V-shaped Lambertian cavities.

The irradiance equation of this simpler model is then

$$E_D(\mathbf{x}) = \gamma_D(\mathbf{x})\cos(\theta_i)(A + B\sin(\alpha)\tan(\beta)\max[0,\cos(\varphi_r - \varphi_i)]). \qquad (1.15)$$

Assuming that there is a linear relation between the irradiance of the image and the image intensity, the $I_D$ *brightness equation* for the ON–model is given by

$$I_D(\mathbf{x}) = \gamma_D(\mathbf{x})\cos(\theta_i)(A + B\sin(\alpha)\tan(\beta)\max[0,\cos(\varphi_r - \varphi_i)]) \qquad (1.16)$$

where

$$A = 1 - 0.5\,\sigma^2(\sigma^2 + 0.33)^{-1} \qquad (1.17)$$
$$B = 0.45\sigma^2(\sigma^2 + 0.09)^{-1}. \qquad (1.18)$$

Note that $A$ and $B$ are two non negative constants depending on the statistics of the cavities via the roughness parameter $\sigma$ that we can imagine that takes values between 0 and $\pi/2$, as representative of the slope of the roughness of the surface considered. In this

model (see Fig. 1.4), $\theta_i$ represents the angle between the unit normal to the surface $\mathbf{N}(\mathbf{x})$ and the light source direction $\boldsymbol{\omega}$, $\theta_r$ stands for the angle between $\mathbf{N}(\mathbf{x})$ and the observer direction $\mathbf{V}$, $\varphi_i$ is the angle between the projection of the light source direction $\boldsymbol{\omega}$ and the $x_1$ axis onto the $(x_1, x_2)$-plane tangent to the surface, $\varphi_r$ denotes the angle between the projection of the observer direction $\mathbf{V}$ and the $x_1$ axis onto the $(x_1, x_2)$-plane tangent to the surface and the two variables $\alpha$ and $\beta$ are given by

$$\alpha = \max\left[\theta_i, \theta_r\right] \text{ and } \beta = \min\left[\theta_i, \theta_r\right]. \tag{1.19}$$

Note that in order to make all the quantities present in this model computable for the implementation, we consider a modified version of the ON–model in which the difference $\varphi_r - \varphi_i$ is constant and depends on the data $\boldsymbol{\omega}$ and $\mathbf{V}$ given. That simplification allows for the numerical tests to compute $\max[0, \cos(\varphi_r - \varphi_i)]$ only once for a whole image.
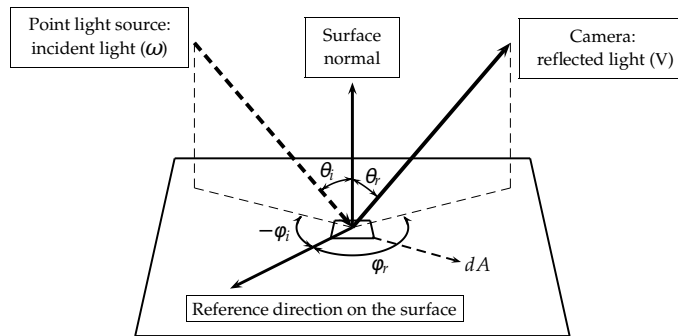


Figure 1.4.: Diffuse reflectance for the ON–model.

We define (see Fig. 1.4):

$$\cos(\theta_i) = \mathbf{N} \cdot \boldsymbol{\omega} = \frac{-\widetilde{\boldsymbol{\omega}} \cdot \nabla u(\mathbf{x}) + \omega_3}{\sqrt{1 + |\nabla u(\mathbf{x})|^2}} \tag{1.20}$$

$$\cos(\theta_r) = \mathbf{N} \cdot \mathbf{V} = \frac{-\widetilde{\mathbf{v}} \cdot \nabla u(\mathbf{x}) + v_3}{\sqrt{1 + |\nabla u(\mathbf{x})|^2}} \tag{1.21}$$

$$\cos(\varphi_r - \varphi_i) = (\omega_1, \omega_2) \cdot (v_1, v_2) = \widetilde{\boldsymbol{\omega}} \cdot \widetilde{\mathbf{v}} \tag{1.22}$$

$$\sin(\theta_i) = \sqrt{1 - (\cos(\theta_i))^2} = \frac{g_\omega(\nabla u(\mathbf{x}))}{\sqrt{1 + |\nabla u(\mathbf{x})|^2}} \tag{1.23}$$

$$\sin(\theta_r) = \sqrt{1 - (\cos(\theta_r))^2} = \frac{g_v(\nabla u(\mathbf{x}))}{\sqrt{1 + |\nabla u(\mathbf{x})|^2}} \tag{1.24}$$

where

$$g_\omega(\nabla u(\mathbf{x})) = \sqrt{1 + |\nabla u(\mathbf{x})|^2 - (-\widetilde{\boldsymbol{\omega}} \cdot \nabla u(\mathbf{x}) + \omega_3)^2} \tag{1.25}$$

$$g_v(\nabla u(\mathbf{x})) = \sqrt{1 + |\nabla u(\mathbf{x})|^2 - (-\widetilde{\mathbf{v}} \cdot \nabla u(\mathbf{x}) + v_3)^2} \tag{1.26}$$

For smooth surfaces, we have $\sigma = 0$ and the ON–model brings back to the L–model. In the particular case $\boldsymbol{\omega} = \mathbf{V} = (0,0,1)$, or, more precisely, when $\cos(\varphi_r - \varphi_i) \leq 0$ (e.g. the case when the unit vectors $\boldsymbol{\omega}$ and $\mathbf{V}$ are perpendicular we get $\cos(\varphi_r - \varphi_i) = -1$) the equation (1.16) simplifies and reduces to a L–model scaled by the coefficient $A$. This means that the model is more general and flexible than the L–model. This happens when only one of the two unit vectors is zero or, more in general, when the dot product between the normalized projections onto the $(x_1, x_2)$-plane of $\boldsymbol{\omega}$ and $\mathbf{V}$ is equal to zero.

Also for this diffuse model we neglect the ambient component. Then, we get $k_D = 1$ and, as a consequence, in the general equation (1.3) the total light intensity $I(\mathbf{x})$ is equal to the only diffuse component $I_D(\mathbf{x})$, in this case described by the equation (1.16). Hence, for what follow, we will write $I(\mathbf{x})$ instead of $I_D(\mathbf{x})$.

To deal with this equation one has to resolve the *min* and *max* operators which appear in (1.16), (1.19). Hence, we must consider several cases (and for simplicity we set the albedo $\gamma_D(\mathbf{x}) = 1$):

**Case 1:** $\theta_i \geq \theta_r$ and $(\varphi_r - \varphi_i) \in [0, \frac{\pi}{2}) \cup (\frac{3}{2}\pi, 2\pi]$

The brightness equation (1.16) becomes

$$I(\mathbf{x}) = \cos(\theta_i)\ \left(A + B\sin(\theta_i)\frac{\sin(\theta_r)}{\cos(\theta_r)}\cos(\varphi_r - \varphi_i)\right) \tag{1.27}$$

and by using the formulas (1.20)-(1.24) we arrive to the following first order nonlinear Hamilton-Jacobi equation

$$I(\mathbf{x})(\sqrt{1 + |\nabla u(\mathbf{x})|^2}) + A(\widetilde{\boldsymbol{\omega}} \cdot \nabla u(\mathbf{x}) - \omega_3) - B\frac{(\widetilde{\boldsymbol{\omega}} \cdot \widetilde{\mathbf{v}})\, g_\omega(\nabla u(\mathbf{x}))\, g_v(\nabla u(\mathbf{x}))}{\sqrt{1 + |\nabla u(\mathbf{x})|^2}\ (-\widetilde{\mathbf{v}} \cdot \nabla u(\mathbf{x}) + v_3)} = 0,$$
$$\tag{1.28}$$

where $\widetilde{\boldsymbol{\omega}} = (\omega_1, \omega_2)$ and $\widetilde{\mathbf{v}} = (v_1, v_2)$.

**Case 2:** $\theta_i < \theta_r$ and $(\varphi_r - \varphi_i) \in [0, \frac{\pi}{2}) \cup (\frac{3}{2}\pi, 2\pi]$

In this case the brightness equation (1.16) becomes

$$I(\mathbf{x}) = \cos(\theta_i)\ \left(A + B\sin(\theta_r)\frac{\sin(\theta_i)}{\cos(\theta_i)}\cos(\varphi_r - \varphi_i)\right) \tag{1.29}$$

and by using again the formulas (1.20)-(1.24) we get

$$I(\mathbf{x})(1 + |\nabla u(\mathbf{x})|^2) + A(\widetilde{\boldsymbol{\omega}} \cdot \nabla u(\mathbf{x}) - \omega_3)\sqrt{1 + |\nabla u(\mathbf{x})|^2} - B(\widetilde{\boldsymbol{\omega}} \cdot \widetilde{\mathbf{v}})\, g_\omega(\nabla u(\mathbf{x}))\, g_v(\nabla u(\mathbf{x})) = 0,$$
$$\tag{1.30}$$

that is again a first order nonlinear Hamilton-Jacobi equation.

**Case 3:** $\forall\, \theta_i, \theta_r$ and $(\varphi_r - \varphi_i) \in [\frac{\pi}{2}, \frac{3}{2}\pi]$

In this case we have the implication $\max(0, \cos(\varphi_r - \varphi_i)) = 0$. The brightness equation

(1.16) simplifies in

$$I(\mathbf{x}) = A \cos(\theta_i) \tag{1.31}$$

and the Hamilton-Jacobi equation associated to it becomes consequentially

$$I(\mathbf{x})(\sqrt{1 + |\nabla u(\mathbf{x})|^2}) + A(\widetilde{\boldsymbol{\omega}} \cdot \nabla u(\mathbf{x}) - \omega_3) = 0, \tag{1.32}$$

that is equal to the L–model scaled by the coefficient $A$.

**Case 4:** $\theta_i = \theta_r$ and $\varphi_r = \varphi_i$

This is a particular case when the position of the light source $\boldsymbol{\omega}$ coincides with the observer direction $\mathbf{V}$ but there are not on the vertical axis. This choice implies $\max[0, \cos(\varphi_i - \varphi_r)] = 1$, then defining $\theta := \theta_i = \theta_r = \alpha = \beta$, the equation (1.16) simplifies to

$$I(\mathbf{x}) = \cos(\theta) \ \left(A + B \sin(\theta)^2 \cos(\theta)^{-1}\right) \tag{1.33}$$

and we arrive to a first order nonlinear Hamilton-Jacobi equation

$$(I(\mathbf{x}) - B)(\sqrt{1 + |\nabla u(\mathbf{x})|^2}) + A(\widetilde{\boldsymbol{\omega}} \cdot \nabla u(\mathbf{x}) - \omega_3) + B \frac{(-\widetilde{\boldsymbol{\omega}} \cdot \nabla u(\mathbf{x}) + \omega_3)^2}{\sqrt{1 + |\nabla u(\mathbf{x})|^2}} = 0. \tag{1.34}$$

**The vertical light case.**

If $\boldsymbol{\omega} = (0, 0, 1)$, regardless of the position of $\mathbf{V}$, the first three cases of the previous four cases are reduced to only one to which we can associate the following PDE to the brightness equation (1.16)

$$I(\mathbf{x}) = \frac{A}{\sqrt{1 + |\nabla u(\mathbf{x})|^2}}. \tag{1.35}$$

In this way we can put it in the following eikonal type equation, analogous to the Lambertian eikonal equation (1.8):

$$|\nabla u(\mathbf{x})| = f(\mathbf{x}) \quad \text{for} \quad \mathbf{x} \in \Omega, \tag{1.36}$$

where

$$f(\mathbf{x}) = \sqrt{\frac{A^2}{I(\mathbf{x})^2} - 1}. \tag{1.37}$$

Following [112], we write the surface as $S(\mathbf{x}, z) = z - u(\mathbf{x}) = 0$, for $\mathbf{x} \in \Omega$, $z \in \mathbb{R}$, and $\nabla S(\mathbf{x}, z) = (-\nabla u(\mathbf{x}), 1)$, so (1.34) becomes

$$(I(\mathbf{x}) - B)|\nabla S(\mathbf{x}, z)| + A(-\nabla S(\mathbf{x}, z) \cdot \boldsymbol{\omega}) + B \left(\frac{\nabla S(\mathbf{x}, z)}{|\nabla S(\mathbf{x}, z)|} \cdot \boldsymbol{\omega}\right)^2 |\nabla S(\mathbf{x}, z)| = 0. \tag{1.38}$$

Defining $d(\mathbf{x}, z) = \nabla S(\mathbf{x}, z)/|\nabla S(\mathbf{x}, z)|$ and $c(\mathbf{x}, z) = I(\mathbf{x}) - B + B(d(\mathbf{x}, z) \cdot \boldsymbol{\omega})^2$, using the equivalence $|\nabla S(\mathbf{x}, z)| \equiv \max\limits_{a \in \partial B_3} \{a \cdot \nabla S(\mathbf{x}, z)\}$ we get

$$\max_{a \in \partial B_3} \{c(\mathbf{x}, z)\, a \cdot \nabla S(\mathbf{x}, z) - A\boldsymbol{\omega} \cdot \nabla S(\mathbf{x}, z)\} = 0. \tag{1.39}$$

Defining the vector field for the ON-model

$$b^{ON}(\mathbf{x}, a) = \frac{1}{A\omega_3} \left( c(\mathbf{x}, z)a_1 - A\omega_1, c(\mathbf{x}, z)a_2 - A\omega_2 \right), \tag{1.40}$$

introducing the exponential transform $\mu v(\mathbf{x}) = 1 - e^{-\mu u(\mathbf{x})}$ as already done for the L–model, we can finally write the Dirichlet problem in the new variable $v$ obtaining the

*Oren-Nayar Model*

$$\begin{cases} \mu v(\mathbf{x}) + \max\limits_{a \in \partial B_3} \{-b^{ON}(\mathbf{x}, a) \cdot \nabla v(\mathbf{x}) + \frac{c(\mathbf{x},z)a_3}{A\omega_3}(1 - \mu v(\mathbf{x}))\} = 1, & \mathbf{x} \in \Omega, \\ v(\mathbf{x}) = 0, & \mathbf{x} \in \partial\Omega. \end{cases} \tag{1.41}$$

Note that the simple homogeneous Dirichlet boundary condition is due to the flat background behind the object but a condition like $u(x) = g(x)$ can also be considered if necessary.

In the particular case when $\cos(\varphi_r - \varphi_i) = 0$, the equation (1.16) simply reduces to

$$I(\mathbf{x}) = A \cos(\theta) \tag{1.42}$$

and, as a consequence, the Dirichlet problem in the variable $v$ is equal to (1.41) with $c(\mathbf{x}, z) = I(\mathbf{x})$.

## 1.4. The Phong model (PH–model)

The PH–model introduce a specular component to the brightness function $I(\mathbf{x})$. As we said in the first section of this chapter, this can be described in general as the sum $I(\mathbf{x}) = k_A I_A(\mathbf{x}) + k_D I_D(\mathbf{x}) + k_S I_S(\mathbf{x})$, where $I_A(\mathbf{x})$, $I_D(\mathbf{x})$ and $I_S(\mathbf{x})$ are the ambient, diffuse and specular light component, respectively. We will set for simplicity $k_A = 0$ and represent the diffuse component $I_D(\mathbf{x})$ as the Lambertian reflectance model.

The most simple specular model is obtained putting the incidence angle equal to the reflection one and $\boldsymbol{\omega}$, $\mathbf{N}(\mathbf{x})$ and $\mathbf{R}(\mathbf{x})$ belong to the same plane.

The PH–model is an empirical model that was developed by Phong [86] in 1975. This model describes the specular light component $I_S(\mathbf{x})$ as a power of the cosine of the angle between the unit vectors $\mathbf{V}$ and $\mathbf{R}(\mathbf{x})$ (it is the vector representing the reflection of the light $\boldsymbol{\omega}$ on the surface), then for the Phong model

$$I_S^{PH}(\mathbf{x}) = \gamma_S(\mathbf{x})(\mathbf{R}(\mathbf{x}) \cdot \mathbf{V})^\alpha \tag{1.43}$$

where $\alpha$ expresses the specular reflection characteristics of a material.

Hence, the brightness equation for the PH–model is

$$I(\mathbf{x}) = k_D\gamma_D(\mathbf{x})(\mathbf{N}(\mathbf{x}) \cdot \boldsymbol{\omega}) + k_S\gamma_S(\mathbf{x})(\mathbf{R}(\mathbf{x}) \cdot \mathbf{V})^\alpha, \tag{1.44}$$

where $\gamma_D(\mathbf{x})$ and $\gamma_S(\mathbf{x})$ represent the diffuse and specular albedo, respectively.

Starting to see in details the PH–model in the case of oblique light source $\boldsymbol{\omega}$ and oblique observer $\mathbf{V}$.

Assuming that $\mathbf{N}(\mathbf{x})$ is the bisector of the angle between $\boldsymbol{\omega}$ and $\mathbf{R}(\mathbf{x})$ (see Fig. 1.5), we obtain

$$\mathbf{N}(\mathbf{x}) = \frac{\boldsymbol{\omega} + \mathbf{R}(\mathbf{x})}{||\boldsymbol{\omega} + \mathbf{R}(\mathbf{x})||} \text{ which implies } \mathbf{R}(\mathbf{x}) = ||\boldsymbol{\omega} + \mathbf{R}(\mathbf{x})||\mathbf{N}(\mathbf{x}) - \boldsymbol{\omega}. \tag{1.45}$$
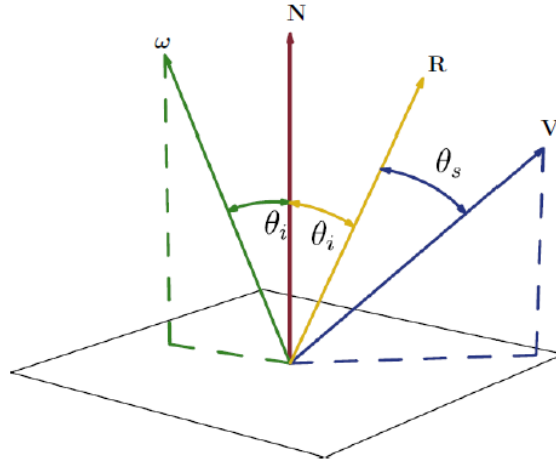


Figure 1.5.: Geometry of the Phong reflection model.

From the parallelogram law, taking into account that $\boldsymbol{\omega}, \mathbf{R}(\mathbf{x})$ and $\mathbf{N}(\mathbf{x})$ are unit vectors, we can write $||\boldsymbol{\omega} + \mathbf{R}(\mathbf{x})|| = 2(\mathbf{N}(\mathbf{x}) \cdot \boldsymbol{\omega})$, then we can derive the unit vector $\mathbf{R}(\mathbf{x})$ as follow:

$$\begin{aligned}
\mathbf{R}(\mathbf{x}) &= 2(\mathbf{N}(\mathbf{x}) \cdot \boldsymbol{\omega})\mathbf{N}(\mathbf{x}) - \boldsymbol{\omega} = 2\left(\frac{-\widetilde{\boldsymbol{\omega}} \cdot \nabla u(\mathbf{x}) + \omega_3}{\sqrt{1 + |\nabla u(\mathbf{x})|^2}}\right)\mathbf{N}(\mathbf{x}) - (\omega_1, \omega_2, \omega_3) \\
&= \left(\frac{-2\widetilde{\boldsymbol{\omega}} \cdot \nabla u(\mathbf{x}) + 2\omega_3}{1 + |\nabla u(\mathbf{x})|^2}\right)(-\nabla u(\mathbf{x}), 1) - (\omega_1, \omega_2, \omega_3). \tag{1.46}
\end{aligned}$$

With this definition of the unit vector $\mathbf{R}(\mathbf{x})$ we have

$$\mathbf{R}(\mathbf{x}) \cdot \mathbf{V} = \left(\frac{-2\widetilde{\boldsymbol{\omega}} \cdot \nabla u(\mathbf{x}) + 2\omega_3}{1 + |\nabla u(\mathbf{x})|^2}\right)(-\nabla u(\mathbf{x}) \cdot \widetilde{\mathbf{v}} + v_3) - \boldsymbol{\omega} \cdot \mathbf{V}. \tag{1.47}$$

Then, putting $\alpha = 1$, equation (1.44) becomes

$$I(\mathbf{x})(1 + |\nabla u(\mathbf{x})|^2) \quad -k_D \gamma_D(\mathbf{x})(-\nabla u(\mathbf{x}) \cdot \widetilde{\boldsymbol{\omega}} + \omega_3)(\sqrt{1 + |\nabla u(\mathbf{x})|^2})$$

$$-2k_S \gamma_S(\mathbf{x})(-\nabla u(\mathbf{x}) \cdot \widetilde{\boldsymbol{\omega}} + \omega_3)(-\widetilde{\mathbf{v}} \cdot \nabla u(\mathbf{x}) + v_3) \qquad (1.48)$$

$$+k_S \gamma_S(\mathbf{x})(\boldsymbol{\omega} \cdot \mathbf{V})(1 + |\nabla u(\mathbf{x})|^2) = 0,$$

to which we add a Dirichlet boundary condition equal to zero, assuming that the surface is standind on a flat blackground. Note that the cosine in the specular term is usually replaced by zero if $\mathbf{R}(\mathbf{x}) \cdot \mathbf{V} < 0$ (and in that case we get back to the L–model).
As we have done for the previous models, we write the surface as $S(\mathbf{x}, z) = z - u(\mathbf{x}) = 0$, for $\mathbf{x} \in \Omega$, $z \in \mathbb{R}$, and $\nabla S(\mathbf{x}, z) = (-\nabla u(\mathbf{x}), 1)$, so (1.48) will be written as

$$(I(\mathbf{x}) + k_S \gamma_S(\mathbf{x})(\boldsymbol{\omega} \cdot \mathbf{V}))|\nabla S(\mathbf{x}, z)|^2 \quad -k_D \gamma_D(\mathbf{x})(\nabla S(\mathbf{x}, z) \cdot \boldsymbol{\omega})(|\nabla S(\mathbf{x}, z)|)$$

$$-2k_S \gamma_S(\mathbf{x})(\nabla S(\mathbf{x}, z) \cdot \boldsymbol{\omega})(\nabla S(\mathbf{x}, z) \cdot \mathbf{V}) = 0. \tag{1.49}$$

Dividing by $|\nabla S(\mathbf{x}, z)|$, defining $d(\mathbf{x}, z) = \nabla S(\mathbf{x}, z)/|\nabla S(\mathbf{x}, z)|$ as in the ON–model and $c(\mathbf{x}) = I(\mathbf{x}) + k_S \gamma_S(\mathbf{x})(\boldsymbol{\omega} \cdot \mathbf{V})$, we get

$$c(\mathbf{x})|\nabla S(\mathbf{x}, z)| - k_D \gamma_D(\mathbf{x})(\nabla S(\mathbf{x}, z) \cdot \boldsymbol{\omega}) - 2k_S \gamma_S(\mathbf{x})(\nabla S(\mathbf{x}, z) \cdot \boldsymbol{\omega})(d(\mathbf{x}, z) \cdot \mathbf{V}) = 0. \tag{1.50}$$

By the equivalence $|\nabla S(\mathbf{x}, z)| \equiv \max\limits_{a \in \partial B_3} \{a \cdot \nabla S(\mathbf{x}, z)\}$ we obtain

$$\max\limits_{a \in \partial B_3} \{c(\mathbf{x}) \, a \cdot \nabla S(\mathbf{x}, z) - k_D \gamma_D(\mathbf{x})(\nabla S(\mathbf{x}, z) \cdot \boldsymbol{\omega}) - 2k_S \gamma_S(\mathbf{x})(\nabla S(\mathbf{x}, z) \cdot \boldsymbol{\omega})(d(\mathbf{x}, z) \cdot \mathbf{V})\} = 0. \tag{1.51}$$

Defining the vector field

$$b^{PH}(\mathbf{x}, a) = $$
$$\frac{(c(\mathbf{x})a_1 - k_D \gamma_D(\mathbf{x})\omega_1 - 2k_S \gamma_S(\mathbf{x})\,\omega_1(d(\mathbf{x}, z) \cdot \mathbf{V}), c(\mathbf{x})a_2 - k_D \gamma_D(\mathbf{x})\omega_2 - 2k_S \gamma_S(\mathbf{x})\,\omega_2(d(\mathbf{x}, z) \cdot \mathbf{V}))}{Q^{PH}(\mathbf{x}, z)}$$

$$\tag{1.52}$$

where

$$Q^{PH}(\mathbf{x}, z) := 2k_S \gamma_S(\mathbf{x})\omega_3(d(\mathbf{x}, z) \cdot \mathbf{V}) + k_D \gamma_D(\mathbf{x})\omega_3, \tag{1.53}$$

and using the exponential transform $\mu v(\mathbf{x}) = 1 - e^{-\mu u(\mathbf{x})}$ as done for the previous models, we can finally write the nonlinear problem corresponding to the

---

*Phong Model*

$$\begin{cases} \mu v(\mathbf{x}) + \max\limits_{a \in \partial B_3} \{-b^{PH}(\mathbf{x}, a) \cdot \nabla v(\mathbf{x}) + \frac{c(\mathbf{x})a_3}{Q^{PH}(\mathbf{x}, z)}(1 - \mu v(\mathbf{x}))\} = 1, & \mathbf{x} \in \Omega, \\ v(\mathbf{x}) = 0, & \mathbf{x} \in \partial\Omega. \end{cases} \tag{1.54}$$

Again, note that the simple homogeneous Dirichlet boundary condition considered is due to the flat background behind the object but a different boundary condition can also be considered.

**A. Oblique light source and vertical position of the observer.**
In the case of oblique light source $\boldsymbol{\omega}$ and vertical observer $\mathbf{V} = (0,0,1)$, the dot product $\mathbf{R}(\mathbf{x}) \cdot \mathbf{V}$ becomes

$$\mathbf{R}(\mathbf{x}) \cdot \mathbf{V} = \frac{-2\widetilde{\boldsymbol{\omega}} \cdot \nabla u(\mathbf{x}) + 2\omega_3}{1 + |\nabla u(\mathbf{x})|^2} - \omega_3 = \frac{-2\widetilde{\boldsymbol{\omega}} \cdot \nabla u(\mathbf{x}) + \omega_3(1 - |\nabla u(\mathbf{x})|^2)}{1 + |\nabla u(\mathbf{x})|^2}. \tag{1.55}$$

The Dirichlet problem in the variable $v$ will be equal to (1.54) with

$$
\begin{aligned}
c(\mathbf{x}) &= I(\mathbf{x}) + \omega_3 k_S \gamma_S(\mathbf{x}), \\
Q^{PH}(\mathbf{x}, z) &= 2k_S \gamma_S(\mathbf{x})(d(\mathbf{x}, z) \cdot \boldsymbol{\omega}) + k_D \gamma_D(\mathbf{x})\omega_3, \\
b^{PH}(\mathbf{x}, a) &= \tfrac{1}{Q^{PH}(\mathbf{x},z)} \left( c(\mathbf{x})a_1 - k_D \gamma_D(\mathbf{x})\omega_1, c(\mathbf{x})a_2 - k_D \gamma_D(\mathbf{x})\omega_2 \right).
\end{aligned}
\tag{1.56}
$$

**B. Vertical light source and oblique position of the observer.**
When $\boldsymbol{\omega} = (0,0,1)$ the definition of the vector $\mathbf{R}(\mathbf{x})$ reported in (1.46) becomes

$$\mathbf{R}(\mathbf{x}) = \left( \frac{-2u_x}{1 + |\nabla u(\mathbf{x})|^2}, \frac{-2u_y}{1 + |\nabla u(\mathbf{x})|^2}, \frac{2}{1 + |\nabla u(\mathbf{x})|^2} - 1 \right) \tag{1.57}$$

and, as a consequence, the dot product $\mathbf{R}(\mathbf{x}) \cdot \mathbf{V}$ with general $\mathbf{V}$ is

$$\mathbf{R}(\mathbf{x}) \cdot \mathbf{V} = \frac{-2\widetilde{\mathbf{v}} \cdot \nabla u(\mathbf{x}) + v_3(1 - |\nabla u(\mathbf{x})|^2)}{1 + |\nabla u(\mathbf{x})|^2}. \tag{1.58}$$

Hence, the Dirichlet problem in the variable $v$ is equal to (1.54) with

$$
\begin{aligned}
c(\mathbf{x}) &= I(\mathbf{x}) + v_3 k_S \gamma_S(\mathbf{x}), \\
Q^{PH}(\mathbf{x}, z) &= 2k_S \gamma_S(\mathbf{x})(d(\mathbf{x}, z) \cdot \mathbf{V}) + k_D \gamma_D(\mathbf{x}), \\
b^{PH}(\mathbf{x}, a) &= \tfrac{1}{Q^{PH}(\mathbf{x},z)} \left( c(\mathbf{x})a_1, c(\mathbf{x})a_2 \right).
\end{aligned}
\tag{1.59}
$$

**C. Vertical light source and vertical position of the observer.**
If we choose $\boldsymbol{\omega} \equiv \mathbf{V} = (0,0,1)$ the equation (1.48) simplify in

$$I(\mathbf{x})(1 + |\nabla u(\mathbf{x})|^2) - k_S \gamma_S(\mathbf{x})(1 - |\nabla u(\mathbf{x})|^2) = k_D \gamma_D(\mathbf{x})(\sqrt{1 + |\nabla u(\mathbf{x})|^2}). \tag{1.60}$$

After some manipulations we can write it in the following eikonal type equation, analogous to the Lambertian eikonal equation (1.8):

$$|\nabla u(\mathbf{x})| = f(\mathbf{x}) \quad \text{for} \quad \mathbf{x} \in \Omega, \tag{1.61}$$

where

$$f(\mathbf{x}) = \sqrt{\frac{k_D^2 \gamma_D(\mathbf{x})^2 - 2I_+(\mathbf{x})I_-(\mathbf{x}) + k_D^2 \gamma_D(\mathbf{x})^2 \sqrt{Q(\mathbf{x})}}{2\left(I(\mathbf{x}) + k_S \gamma_S(\mathbf{x})\right)^2}}, \tag{1.62}$$

with

$$
\begin{aligned}
I_+(\mathbf{x}) &:= I(\mathbf{x}) + k_S \gamma_S(\mathbf{x}), & \text{(1.63)} \\
I_-(\mathbf{x}) &:= I(\mathbf{x}) - k_S \gamma_S(\mathbf{x}), & \text{(1.64)} \\
Q(\mathbf{x}) &:= k_D^2 \gamma_D(\mathbf{x})^2 + 8k_S^2 \gamma_S(\mathbf{x})^2 + 8\, I(\mathbf{x})\, k_S \gamma_S(\mathbf{x}). & \text{(1.65)}
\end{aligned}
$$

## 1.5. The Blinn-Phong model (BP–model)

A modification of the PH–model has been proposed by Blinn [10] in 1977 taking into account an intermediate vector $\mathbf{H}$, which bisects the angle between the unit vectors $\boldsymbol{\omega}$ and $\mathbf{V}$ (see Fig. 1.6).
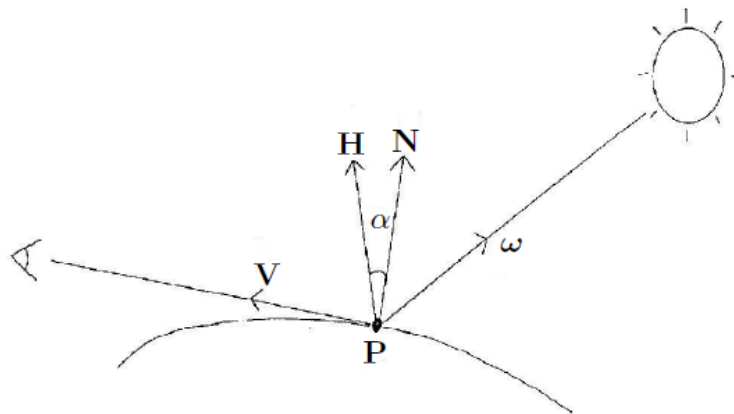


Figure 1.6.: Geometry of the Blinn-Phong reflection model.

For this model, the specular component is defined as follow:

$$I_S^{BP}(\mathbf{x}) = \gamma_S(\mathbf{x})(\mathbf{N}(\mathbf{x}) \cdot \mathbf{H})^c \tag{1.66}$$

where the parameter $c$ is a measure of shininess of the surface and $\mathbf{H} = \dfrac{\boldsymbol{\omega} + \mathbf{V}}{||\boldsymbol{\omega} + \mathbf{V}||}$.

Note that the BP–model will produce a faster algorithm in terms of CPU time in the case in which the observer and the light source are placed at infinity because $\mathbf{H}$ is in this case independent of the position and the orientation of the surface. This implies that $\mathbf{H}$ can be computed only once for each light whereas this is not possible in the original model of Phong where $\mathbf{R}(\mathbf{x})$ depends on the curvature of the surface and must be computed again at every pixel of the image.

Setting $k_A = 0$ and choosing the diffuse component $I_D(\mathbf{x})$ as the L–model as done for the PH–model, the *brightness equation* for the BP–model is

$$I(\mathbf{x}) = k_D \gamma_D(\mathbf{x})(\mathbf{N}(\mathbf{x}) \cdot \boldsymbol{\omega}) + k_S \gamma_S(\mathbf{x})(\mathbf{N}(\mathbf{x}) \cdot \mathbf{H})^c, \tag{1.67}$$

where $\gamma_D(\mathbf{x})$ and $\gamma_S(\mathbf{x})$ represent the diffuse and specular albedo, respectively.
We suppose $v_3 > 0$. This assumption is not restrictive because it is coherent with $\omega_3 > 0$. Recalling the definition of the unit normal given in (1.1), we can write the first order nonlinear Hamilton-Jacobi equation associated to (1.67) as

$$I(\mathbf{x}) = k_D \gamma_D(\mathbf{x}) \left( \frac{-\nabla u(\mathbf{x}) \cdot \widetilde{\boldsymbol{\omega}} + \omega_3}{\sqrt{1 + |\nabla u(\mathbf{x})|^2}} \right) + k_S \gamma_S(\mathbf{x}) \left( \frac{-\widetilde{\mathbf{H}} \cdot \nabla u(\mathbf{x}) + H_3}{\sqrt{1 + |\nabla u(\mathbf{x})|^2}} \right)^c, \tag{1.68}$$

where $\widetilde{\mathbf{H}} = (H_1, H_2)$.
We start by analyzing the equation for $c = 1$ in the case of oblique light source and generic position of the observer (vertical or oblique). Equation (1.68) can be written as

$$I(\mathbf{x})(\sqrt{1 + |\nabla u(\mathbf{x})|^2}) - k_D \gamma_D(\mathbf{x})(-\nabla u(\mathbf{x}) \cdot \widetilde{\boldsymbol{\omega}} + \omega_3) - k_S \gamma_S(\mathbf{x}) \left( -\widetilde{\mathbf{H}} \cdot \nabla u(\mathbf{x}) + H_3 \right) = 0. \tag{1.69}$$

As already done for the previous models, we write the equation (1.69) in terms of $\nabla S(\mathbf{x}, z) = (-\nabla u(\mathbf{x}), 1)$, having defined the surface as $S(\mathbf{x}, z) = z - u(\mathbf{x}) = 0$, for $\mathbf{x} \in \Omega, z \in \mathbb{R}$

$$I(\mathbf{x})|\nabla S(\mathbf{x}, z)| - k_D \gamma_D(\mathbf{x})(\nabla S(\mathbf{x}, z) \cdot \boldsymbol{\omega}) - k_S \gamma_S(\mathbf{x})(\nabla S(\mathbf{x}, z) \cdot \mathbf{H}) = 0. \tag{1.70}$$

By using the equivalence $|\nabla S(\mathbf{x}, z)| \equiv \max_{a \in \partial B_3} \{a \cdot \nabla S(\mathbf{x}, z)\}$ we obtain

$$\max_{a \in \partial B_3} \{I(\mathbf{x}) \, a \cdot \nabla S(\mathbf{x}, z) - k_D \gamma_D(\mathbf{x})(\boldsymbol{\omega} \cdot \nabla S(\mathbf{x}, z)) - k_S \gamma_S(\mathbf{x})(\nabla S(\mathbf{x}, z) \cdot \mathbf{H})\} = 0. \tag{1.71}$$

Defining the vector field

$$b^{BP}(\mathbf{x}, a) = \frac{1}{Q^{BP}} \left( I(\mathbf{x})a_1 - k_D \gamma_D(\mathbf{x})\omega_1 - k_S \gamma_S(\mathbf{x})H_1, I(\mathbf{x})a_2 - k_D \gamma_D(\mathbf{x})\omega_2 - k_S \gamma_S(\mathbf{x})H_2 \right) \tag{1.72}$$

where

$$Q^{BP} := k_D \gamma_D(\mathbf{x})\omega_3 + k_S \gamma_S(\mathbf{x})H_3, \tag{1.73}$$

and using the exponential transform $\mu v(\mathbf{x}) = 1 - e^{-\mu u(\mathbf{x})}$ we can finally write the nonlinear problem corresponding to the BP–model in the new variable $v$ obtaining

*Blinn-Phong Model*

$$\begin{cases} \mu v(\mathbf{x}) + \max_{a \in \partial B_3} \{ -b^{BP}(\mathbf{x}, a) \cdot \nabla v(\mathbf{x}) + \frac{I(\mathbf{x})a_3}{Q^{BP}}(1 - \mu v(\mathbf{x})) \} = 1, & \mathbf{x} \in \Omega, \\ v(\mathbf{x}) = 0, & \mathbf{x} \in \partial\Omega. \end{cases} \quad (1.74)$$

We add Dirichlet boundary condition supposing the object is standing on a flat background.

**A. Vertical light source and oblique position of the observer.**
When $\boldsymbol{\omega} = (0, 0, 1)$, equation (1.69) becomes

$$I(\mathbf{x})(\sqrt{1 + |\nabla u(\mathbf{x})|^2}) - k_D \gamma_D(\mathbf{x}) - k_S \gamma_S(\mathbf{x}) \left( -\widetilde{\mathbf{H}} \cdot \nabla u(\mathbf{x}) + H_3 \right) = 0. \quad (1.75)$$

Hence, the Dirichlet problem in the variable $v$ is equal to (1.74) with

$$\begin{aligned} Q^{BP} &= k_D \gamma_D(\mathbf{x}) + k_S \gamma_S(\mathbf{x}) H_3, \\ b^{BP}(\mathbf{x}, a) &= \tfrac{1}{Q^{BP}} \left( I(\mathbf{x})a_1 - k_S \gamma_S(\mathbf{x}) H_1, I(\mathbf{x})a_2 - k_S \gamma_S(\mathbf{x}) H_2 \right). \end{aligned} \quad (1.76)$$

**B. Vertical light source and vertical position of the observer.**
If we choose $\boldsymbol{\omega} \equiv \mathbf{V} = (0, 0, 1)$ then $\mathbf{H} = (0, 0, 1)$ and equation (1.69) simplify in

$$I(\mathbf{x})(\sqrt{1 + |\nabla u(\mathbf{x})|^2}) - k_D \gamma_D(\mathbf{x}) - k_S \gamma_S(\mathbf{x}) = 0 \quad (1.77)$$

and we can write it in the following eikonal type equation, analogous to the Lambertian eikonal equation (1.8):

$$|\nabla u(\mathbf{x})| = f(\mathbf{x}) \quad \text{for} \quad \mathbf{x} \in \Omega, \quad (1.78)$$

where

$$f(\mathbf{x}) = \sqrt{\frac{(k_D \gamma_D(\mathbf{x}) + k_S \gamma_S(\mathbf{x}))^2}{I(\mathbf{x})^2} - 1}. \quad (1.79)$$

## 1.6. A short survey of other non-Lambertian models

Several other non-Lambertian specular models exist. In this section we briefly describe some of them with specular component, in which the main feature is that the reflected radiance is concentrated along a particular direction, the one for which the reflected ray and the incident one lie on the same plane and the angle of reflection equals the incidence angle (this is the behavior of a perfect mirror).

### 1.6.1. The Torrance-Sparrow model

A more sophisticated model was proposed by Torrance and Sparrow [111]. This model assumes that the surface is composed of small-faces mirror randomly oriented. Only the faces with a normal in the direction of $\mathbf{H}$ contribute to $I_S$. The specular model that results is

$$I_S = FDA \tag{1.80}$$

where

**F** is the Fresnel reflection coefficient, computed as the ratio between the reflected and incident radiation ($\frac{I_r}{I_i}$);

**D** is a distribution function of the orientation of the face;

**A** is a geometric factor of attenuation improved by inclination.

The Fresnel coefficient models the amount of light that is reflected from the individual faces. In general, it depends on the angle of incidence and the refractive index of the reflective material (which is a complex number in the case of a material which exhibits absorption).

The distribution function $D$ describes the orientation of the faces relative to the normal $\mathbf{N}$. In agreement with Torrance and Sparrow, Healey and Blinford [48], which Tsai and Shah refer in [113], use the Gaussian distribution function

$$D = Ke^{-(\frac{\alpha}{m})^2} \tag{1.81}$$

where

$K$ is a normalization constant;

$m$ is the constant that indicates the roughness of the surface and is proportional to the standard deviation of the Gaussian.

Small values of $m$ describe smooth surfaces for which most of the specular reflection is concentrated in one direction. Large values of $m$, instead, describe rough surfaces with large differences in the orientation between neighboring faces.

The factor $A$ quantifies the effects of a geometrical attenuation factor $G$ corrected for inclination dividing by $(\mathbf{N} \cdot \mathbf{V})$

$$A = \frac{G}{(\mathbf{N} \cdot \mathbf{V})} \,. \tag{1.82}$$

**Remark 1.1** *Since the intensity is proportional to the number of faces pointing in the direction of* $\mathbf{H}$*, we must take into account that the observer sees a greater surface area when this is inclined. The increase of the visible area is inversely proportional to the cosine of the angle of inclination, which is the angle between the vectors* $\mathbf{N}$ *and* $\mathbf{V}$*. This explains the division by* $(\mathbf{N} \cdot \mathbf{V})$ *in* (1.82).

The $G$ that appears in (1.82) is the one used by Torrance and Sparrow [111], who, by this hypothesis, examine the possible configurations of the different faces that you have in correspondence with the phenomena of *shading* or *masking*; for different positions of the light and the observer, it may have the three different cases shown in the Figs. 1.7, 1.8, 1.9.
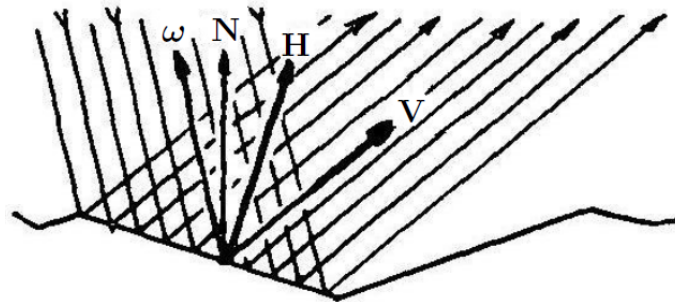
Figure 1.7.: Case (a) without interference.
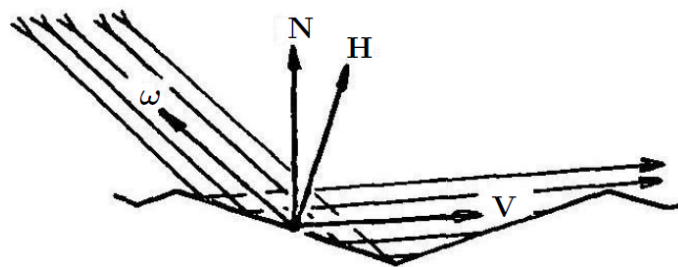
Figure 1.8.: Case (b): A part of the reflected light is intercepted (*shading*).
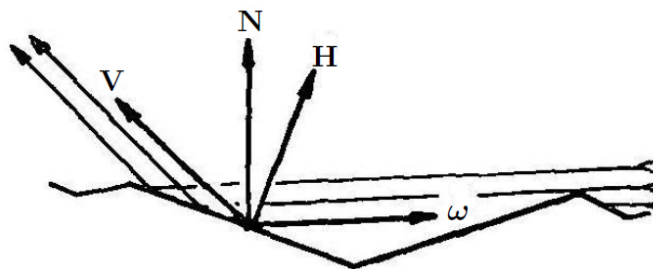
Figure 1.9.: Case (c): Part of the incident light is hidden (*masking*).

**Remark 1.2** *Note that the vectors $\omega$ and $\mathbf{V}$ do not necessarily belong to the plane containing $\mathbf{N}$ and $\mathbf{H}$. This is better visible with a top view presented in Fig. 1.10.*
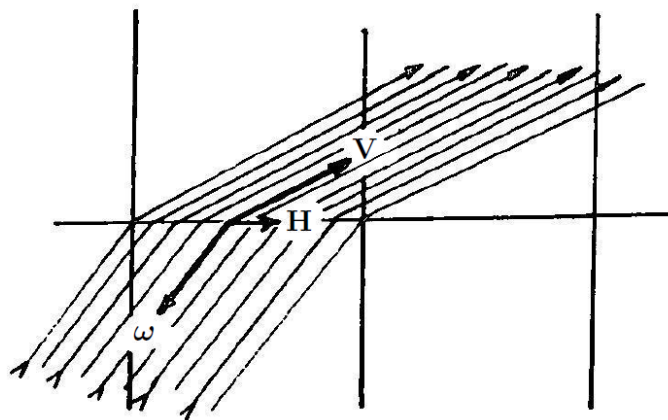
Figure 1.10.: Top view of the reflection of a micro face.

In the case of Fig. 1.7 the value of $G$ is 1.0, which means that we are not in the presence of attenuation.

In order to compute $G$ for the case (b) visible in Fig. 1.8, we need to compute the ratio $1 - \frac{m}{l}$ that indicates the amount of face that contributes to the reflected light (see Fig. 1.11).
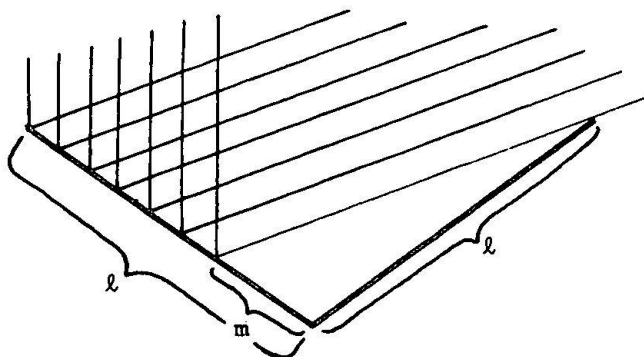
Figure 1.11.: Light that comes out is $1 - \frac{m}{l}$.

We can reduce the problem to two dimensions if we project $\mathbf{V}$ on the plane that contains $\mathbf{N}$ and $\mathbf{H}$. By calling this projection $\mathbf{V}_P$ and denoting the interested angles we obtain the following Fig. 1.12.

Recalling that in any triangle the ratio between the measure of one side and the sine of the opposite angle is constant, we obtain that

$$\frac{m}{l} = \frac{\sin f}{\sin b}.$$
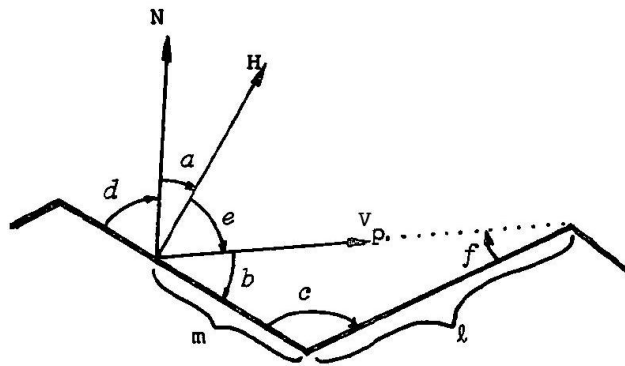
Figure 1.12.: Measurement of $\frac{m}{l}$.

We note that

$$\sin b = \cos e \qquad\qquad \cos b = \sin e$$

being known the relations

$$\sin \alpha = \cos(\frac{\pi}{2} - \alpha) \qquad\qquad \cos \alpha = \sin(\frac{\pi}{2} - \alpha).$$

Since the sum of the angles of a triangle must be equal to $\pi$ we obtain that

$$\sin f = \sin(b + c) = \sin b \cos c + \cos b \sin c. \tag{1.83}$$

For the symmetry of the cavity and the complementarity of the angles $d$ and $a$

$$c = 2d$$
$$\cos c = 1 - 2\sin^2 d = 1 - 2\cos^2 \alpha$$
$$\sin c = 2\cos d \sin d = 2\sin \alpha \cos \alpha.$$

By using these into (1.83) we get

$$\sin f = \cos e(1 - 2\cos^2 \alpha) + 2\sin e \sin \alpha \cos \alpha$$
$$= \cos e - 2\cos \alpha(\cos e \cos \alpha - \sin e \sin \alpha)$$
$$= \cos e - 2cos\alpha \cos(e + \alpha)$$
$$= (\mathbf{H} \cdot \mathbf{V}_P) - 2(\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \mathbf{V}_P).$$

Since $\mathbf{V}_P$ is the projection of $\mathbf{V}$ on the plane which $\mathbf{N}$ and $\mathbf{H}$ belong, then $\mathbf{N} \cdot \mathbf{V}_P = \mathbf{N} - \mathbf{V}$ and $\mathbf{H} - \mathbf{V}_P = \mathbf{H} \cdot \mathbf{V}$ such that

$$G_b = 1 - \tfrac{m}{l} = 2(\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \mathbf{V})/(\mathbf{V} \cdot \mathbf{H}).$$

Examining the case (c) we see that is the same as case (b), but with the roles of $\boldsymbol{\omega}$ and $\mathbf{V}$ exchanged. Thus

$$G_c = 2(\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \boldsymbol{\omega})/(\mathbf{H} \cdot \boldsymbol{\omega}) = 2(\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \boldsymbol{\omega})/(\mathbf{V} \cdot \mathbf{H}).$$

For a particular situation, the actual value of $G$ is the minimum of $G_a$, $G_b$ and $G_c$

$$G = \min\{1, \frac{2(\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \mathbf{V})}{(\mathbf{V} \cdot \mathbf{H})}, \frac{2(\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \boldsymbol{\omega})}{(\mathbf{V} \cdot \mathbf{H})}\}. \tag{1.84}$$

**Remark 1.3** *The attenuation factor $G$ is a value between $0$ and $1$ and represents the proportionate quantity of light that remains after the effect of* shading *or* masking.

Focusing for a moment on the Fresnel coefficient $F$, which is a function of the incidence angle of micro-faces and refractive index, it is given by the following expression:

$$F = \frac{1}{2}\left(\frac{\sin^2(\phi - \theta)}{\sin^2(\phi + \theta)} + \frac{\tan^2(\phi - \theta)}{\tan^2(\phi + \theta)}\right) \tag{1.85}$$

where

$\sin\theta = \frac{\sin\phi}{n}$

$\phi$ is the incidence angle

$n$ is the refractive index.

In our case, the incidence angle is $\phi = \arccos(\boldsymbol{\omega} \cdot \mathbf{H}) = \arccos(\mathbf{V} \cdot \mathbf{H})$. The interesting thing about this function is that it has a substantially different shape for metallic and non-metallic substances. For metals, which correspond to large values of $n$, $F$ is almost constant at 1; for non-metals, which correspond to small values of $n$, has more exponential form, starting from 0 per $\phi = 0$ and reaching 1 in $\phi = \pi/2$.

Back to the issue of the distribution functions mentioned, Blinn [10] compares the distribution function of Blinn-Phong (Cf. (1.66))

$$D_1(\alpha) = \cos^{c_1}\alpha \tag{1.86}$$

with the function proposed by Torrance and Sparrow (Cf. (1.81))

$$D_2(\alpha) = e^{-(\alpha c_2)^2} \tag{1.87}$$

and with another introduced by Trowbridge and Reitz

$$D_3(\alpha) = \left(\frac{c_3^2}{\cos^2\alpha(c_3^2 - 1) + 1}\right)^2 \tag{1.88}$$

where $\alpha$ is the angle between $\mathbf{N}$ and $\mathbf{H}$ (see Fig. 1.6) and $c_3$ is the eccentricity of the ellipsoid and it is zero for very bright surfaces and is one for those very diffuse.

Each of these distribution functions takes on a maximum value of 1 for $\alpha = 0$. To compare these three functions is necessary to specify the values of $c_1$, $c_2$ and $c_3$ with respect to the same variable. If we compute these values as a function of the angle $\beta$ in

which the distribution falls in the middle we obtaine

$$c_1 = -\frac{\ln 2}{\ln \cos \beta}$$

$$c_2 = \frac{\sqrt{\ln 2}}{\beta}$$

$$c_3 = \left( \frac{\cos^2 \beta - 1}{\cos^2 \beta - \sqrt{2}} \right)^{1/2}.$$

According to Blinn, the distribution function that we must choose is the $D_3(\alpha)$.

Thus, we have to distinguish three cases following (1.84) in order to define the PDE associated to the (1.80).

**Case 1:** $G = 1$

$$I(x) = \frac{-u_x \omega_1 - u_y \omega_2 + \omega_3}{\sqrt{|\nabla u|^2 + 1}} + FD \frac{\sqrt{|\nabla u|^2 + 1}}{-u_x v_1 - u_y v_2 + v_3}$$

**Case 2:** $G = 2(\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \mathbf{V})/(\mathbf{V} \cdot \mathbf{H})$

$$I(x) = \frac{-u_x \omega_1 - u_y \omega_2 + \omega_3}{\sqrt{|\nabla u|^2 + 1}} + FD \frac{2}{\sqrt{|\nabla u|^2 + 1}} \frac{-u_x(\omega_1 + v_1) - u_y(\omega_2 + v_2) + \omega_3 + v_3}{\omega_1(\omega_1 + v_1) + \omega_2(\omega_2 + v_2) + \omega_3(\omega_3 + v_3)}$$

**Case 3:** $G = 2(\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \boldsymbol{\omega})/(\mathbf{V} \cdot \mathbf{H})$

$$I(x) = \frac{-u_x \omega_1 - u_y \omega_2 + \omega_3}{\sqrt{|\nabla u|^2 + 1}} +$$

$$FD \frac{2}{\sqrt{|\nabla u|^2 + 1}} \frac{(-u_x(\omega_1 + v_1) - u_y(\omega_2 + v_2) + \omega_3 + v_3)(-u_x \omega_1 - u_y \omega_2 + \omega_3)}{(\omega_1(\omega_1 + v_1) + \omega_2(\omega_2 + v_2) + \omega_3(\omega_3 + v_3))(-u_x v_1 - u_y v_2 + v_3)}$$

### 1.6.2. The Healey-Binford model

Healey and Binford [48] simplify (1.80) considering $F$ as a constant with respect to the geometry of the visual; this is a good approximation for metals and many other materials. Furthermore, noting that the exponential factor in (1.81) changes much more rapidly than any term of $A$, they also consider $A$ constant through mirroring. So the form of (1.80) becomes

$$I_S = K' e^{-(\frac{\alpha}{m})^2} \tag{1.89}$$

where $K'$ is a constant. It is assumed that the observer and the light source is placed at a distance relative to the size of the surface, therefore, $\mathbf{V}$ and $\boldsymbol{\omega}$ can be regarded as constant; consequently, also the vector $\mathbf{H}$ which bisects the angle between the two vectors is constant. In addition, the position of the light source and the observer are supposed known. Finally, since the distance of the observer from the surface is large, one can approximate the perspective projection with an orthographic projection. The equation (1.89) is solved in a geometric way by Healey and Benford, while the same expression of $I_S$ is solved by Tsai and Shah [113] using a linear approximation, basing

the iterative algorithm on the form of the method of Newton-Raphson that converges quadratically if we start from an initial approximation sufficiently accurate.

## 1.7. Viscosity solutions and concave/convex ambiguity

We have seen in the previous sections the description of several reflectance models. Now, let's see which kind of solutions we are looking for. In this section we present a brief introduction to the notion of continuous viscosity solutions of first order Hamilton-Jacobi equations. The aim is first to present the fundamental definitions and theorems which are used to try to solve the SfS problem.

The notion of viscosity solution of Hamilton-Jacobi equations has been introduced by Crandall and Lions [26, 27, 28, 74] in the 80s. It is a very nice way of making quantitative and operational the intuitive idea of weak solutions of first-order (and for that matter, second-order) PDEs. In the context of the SfS problem we are only concerned with first-order PDEs.

The following definitions and results can be found in Barles's, Bardi and Capuzzo Dolcetta's or Lions' book [5, 8, 74].

### 1.7.1. A notion of weak solutions

Let $u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ be a $C^1$ function ($\Omega$ is an open subset of $\mathbb{R}^2$) and consider a Hamilton-Jacobi equation of the form:

$$H(x, y, u(x, y), \nabla u(x, y)) = 0 \quad \text{on } (x, y) \in \Omega, \tag{1.90}$$

where $H$ is a continuous scalar function on $\Omega \times \mathbb{R} \times \mathbb{R}^n$ and it is called Hamiltonian.

**Definition 1.1** *Viscosity subsolution:*
*$u \in BUC(\Omega)$ (set of bounded and uniformly continuous functions) is a viscosity subsolution of equation (1.90) if:*

$$\forall \phi \in C^1(\Omega), \forall (x_0, y_0) \in \Omega \text{ local maximum of } (u - \phi),$$

$$H(x_0, y_0, u(x_0, y_0), \nabla \phi(x_0, y_0)) \leq 0$$

**Definition 1.2** *Viscosity supersolution:*
*$u \in BUC(\Omega)$ is a viscosity supersolution of equation (1.90) if:*

$$\forall \phi \in C^1(\Omega), \forall (x_0, y_0) \in \Omega \text{ local minimum of } (u - \phi),$$

$$H(x_0, y_0, u(x_0, y_0), \nabla \phi(x_0, y_0)) \geq 0$$

**Definition 1.3** *Viscosity solution:*
*$u$ is a viscosity solution of equation (1.90) if it is both subsolution and supersolution of (1.90).*

### 1.7.2. Existence and uniqueness of continuous viscosity solutions

As we have seen, the classical modelization of Lambertian SfS problem results in an Hamilton-Jacobi equation where the Hamiltonian does not depend on $u$.

We present here an uniqueness result due to Ishii [56]. This result has been proved later in a different manner by Lions [74]. Rouy and Tourin have also given this uniqueness result for Hamiltonian $H$ which do not depend upon $u$ (see [97]). For more general condition, see [75].

**Theorem 1.7.1** *Let $u, v \in BUC(\overline{\Omega})$ respectively subsolution and supersolution of the equation:*

$$H(x, y, \nabla u(x, y)) = 0 \quad \text{on the bounded open set } \Omega \subset \mathbb{R}^2. \tag{1.91}$$

*If the following hypotheses are verified:*

- *$\forall (x_1, y_1), (x_2, y_2) \in \Omega, \ \forall p \in \mathbb{R}^2, \ |H(x_1, y_1, p) - H(x_2, y_2, p)| \leq \rho(|(x_1 - x_2, y_1 - y_2)|(1 + |p|))$, where $\rho$ is a continuous nondecreasing function such that $\rho(0) = 0$;*

- *$H$ is continuous in $\overline{\Omega} \times \mathbb{R}^2$ and convex with respect to $\nabla u$;*

- *there exists a strict viscosity subsolution $\underline{u} \in C^1(\Omega) \cap C(\overline{\Omega})$ of (1.91) (i.e. such that $H(x, y, \nabla u(x, y)) < 0$ for all $(x, y) \in \Omega$);*

*then there exists at most one continuous viscosity solution of (1.91) verifying $u = g$ on $\partial\Omega$ where $g \in C(\partial\Omega)$.*

In the eikonal equation case, it is easy to show that the previous theorem is sharp: there exist many solutions if there are singular points in $\Omega$. Recall that in this context, a singular point is a point $(x, y) \in \Omega$ such that $I(x, y) = 1$.

### 1.7.3. Methods of resolution of PDEs

Equations (1.7) and (1.8) have attracted much attention in the research community in PDEs for their wide range of applications. In the framework of the SfS problem, several methods of resolution have been tested. In the following we focus our attention on two methods: characteristic strips expansion and approximation of viscosity solutions.

**Characteristic strip expansion.**
The characteristic method is an important tool useful to solve a particular class of PDEs. The one more suitable are the partial differential equations where the information is propagated with finite velocity: hyperbolic equation which belong the wave equation and the transport equation are examples.

The simpler way to introduce this method, that makes understandable its intrinsic meaning in particular for our problem, is to introduce the characteristics: let us consider a first order nonlinear PDE

$$F(x, y, u(x, y), \nabla u(x, y)) = 0, \qquad \text{in } \Omega, \tag{1.92}$$

under the boundary condition

$$u(x, y) = g(x, y), \qquad \text{on } \Gamma, \tag{1.93}$$

where $\Gamma \subset \partial\Omega$ and $g : \Gamma \to \mathbb{R}$ are known. Furthermore we suppose that $F$ and $g$ are regular functions. The characteristics method solve (1.92)-(1.93) converting the partial differential equation in a opportune ordinary differential equations system. We suppose that $u$ solves (1.92)-(1.93) and we fix a point $(x, y) \in \Omega$. We want to compute $u(x, y)$ finding a curve that lies in $\Omega$, which connects $(x, y)$ to a point $(x_0, y_0) \in \Gamma$ and along which is possible to compute $u$. Since the boundary condition, we know the value of $u$ in $(x_0, y_0)$. We hope to be able to compute $u$ along all the curve and so in particular also in $(x, y)$. But, how we have to choose this curve? Supposing that it is parametrically described by the function $(x_c(s), y_c(s))$; the parameter $s$ belongs to an interval of $\mathbb{R}$. We suppose $u \in C^2$ solution of the equation (1.92). We define also

$$z(s) = u(x_c(s), y_c(s)). \tag{1.94}$$

In addition, we set:

$$\boldsymbol{p}(s) = \nabla u(x_c(s), y_c(s)), \tag{1.95}$$

that is, $\boldsymbol{p}(s) = (p^1(s), p^2(s))$ where

$$p^1(s) = \frac{\partial u}{\partial x}(x_c(s), y_c(s)), \quad p^2(s) = \frac{\partial u}{\partial y}(x_c(s), y_c(s)). \tag{1.96}$$

In this way $z(\cdot)$ gives the value of $u$ along the curve and $\boldsymbol{p}(\cdot)$ records the values of the gradient $\nabla u$. We must choose the function $(x_c(\cdot), y_c(\cdot))$ in such a way that we can compute $z(\cdot)$ and $\boldsymbol{p}(\cdot)$.

If we differentiate (1.96) with respect to $s$, and the equation (1.92) with respect to $x$ and $y$ and (1.94), making opportune substitutions, we arrive to the following equations:

$$\begin{cases} \text{(a)} & (\dot{p}^1(s), \dot{p}^2(s)) = -D_z F(p(s), z(s), x_c(s), y_c(s)) p(s) \\ & \quad -(D_{x_c} F(p(s), z(s), x_c(s), y_c(s)), D_{y_c} F(p(s), z(s), x_c(s), y_c(s))) \\ \text{(b)} & \dot{z}(s) = D_p F(p(s), z(s), x_c(s), y_c(s)) \cdot p(s) \\ \text{(c)} & (\dot{x}_c(s), \dot{y}_c(s)) = D_p F(p(s), z(s), x_c(s), y_c(s)). \end{cases} \tag{1.97}$$

This important system of $2n + 1$ ODEs of the first order includes the *characteristic* equations of the first order nonlinear PDE (1.92). The functions $\boldsymbol{p}(\cdot) = (p^1(\cdot), p^2(\cdot))$, $z(\cdot)$, $(x_c(\cdot), y_c(\cdot))$ are called characteristics. We will refer to $(x_c(\cdot), y_c(\cdot))$ as projected characteristic: it is the projection of the characteristic $(\boldsymbol{p}(\cdot), z(\cdot), x_c(\cdot), y_c(\cdot)) \subset \mathbb{R}^5$ on the domain $\Omega \subset \mathbb{R}^2$.

The first mention of 3D reconstruction using photometric cues is due to the Dutch astronomer Van Diggelen [33]. The first resolution was suggested by Rindfleisch [96] who demonstrated that if the photometric behavior of a surface follows certain properties, then the shape can be expressed as an integral along a set of convergent straight lines. He implemented this computation on images of the Moon, claiming that its surface

verifies the necessary photometric properties reasonably well. Later, Horn suggested calling this problem "Shape–form–Shading", and showed that the resolution proposed by Rindfleisch in a particular case could be generalized, while still using the characteristic strip expansion [51], under the following two conditions:

- the function $u$ has to be of class $C^2$

- the 5-tuple $(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y})$ has to be known at every point of a curve called the initial curve, which means in fact that two boundary conditions are needed simultaneously, one on $u$ (Dirichlet boundary condition) and other on $(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y})$ (Neumann boundary condition).

The characteristic lines (which are the lines along which the integration has to be performed) can be of any form in the image domain, and this differs from the case studied by Rindfleisch.

Besides the inherent defect of error accumulation, which is typical of every method of resolution using integration, the determination of these characteristic lines is a new problem itself, since they also are defined through integration. Therefore, the accuracy of boundary conditions is much more crucial than for other methods. It follows that a certain number of obstacles must be overcome (i.e. the crossing of a characteristic lines) which should normally occur only at a singular points, or at the presence of holes in $\Omega$, which must be filled using secondary lines [51].

Finally, this method has been essentially used for the theoretical study of the number of solutions of class $C^2$ of the eikonal equation: a number of uniqueness results have been provided (see [80, 79, 14, 71] and the reference therein).

**Approximation of viscosity solutions.**
Starting from the paper by Rouy and Tourin [97], the most recent approach to the resolution of SfS uses the notion of viscosity solutions to first order PDEs. These are almost-everywhere solution (a.e. solutions) which can be obtained as the limit in a family of solutions of regularized second order problems (the so-called "vanishing viscosity" method). These solutions are typical Lipschitz continuous solutions (but discontinuous viscosity solutions have also been considered in the literature [8]). The development of the theory of viscosity solutions for Hamilton-Jacobi type equations provides a good framework for the analysis of the SfS problem.

Moreover, several algorithms have been proposed to compute viscosity solutions. Finite difference numerical methods have been used in [97, 75] for the resolution of (1.8) and generalized to the resolution of (1.7) in [92]. Similar results have been obtained by Oliensis and Dupuis [81] with an algorithm based on the Markow Chain approximation. Unfortunately, the Dirichlet problem (1.7)-(1.11) can have several weak solutions in the viscosity sense and also several classical solutions (due to the so-called concave/convex ambiguity, see [51]). As an example, all the surfaces represented in Fig. 1.13 are viscosity solutions of the same equations (1.8)-(1.10) which is a particular case of (1.7)-(1.11) (in fact the equation is $|u'| = -2x$ with homogenous Dirichlet boundary condition). The

solution represented in Fig. 1.13-a is the maximal solution and is smooth. All the non-smooth a.e. solutions, which can be obtained by a reflection with respect to a horizontal axis, are still admissible weak solutions (see Fig. 1.13-b). In this example, the lack of uniqueness of the viscosity solutions is due to the existence of a singular point where the right hand side of (1.8) vanishes. An additional effort is then needed to define which is the preferable solution since the lack of uniqueness is also a big drawback when trying to compute a numerical solution. In order to circumvent these difficulties, the problem is usually solved by adding some information such as height at each singular point [75].
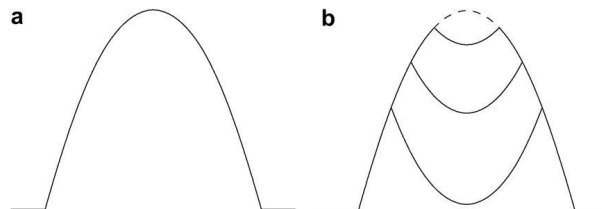


Figure 1.13.: Illustration of the concave/convex ambiguity: (a) maximal solution and (b) a.e. solutions giving the same image.

More recently, an attempt has been made to eliminate the need for a priori additional information. In recent results in the theory of viscosity solutions, the "maximal solution" without additional information apart from the equation was characterized, as was the construction of an algorithm which converges to that solution. A result by Ishii and Ramaswamy [57] applied to SfS guarantees that if the function $I$ is continuous and the number of singular points is finite, then a unique maximal solution (in the viscosity sense) of (1.7)-(1.11) exists. It should be noted that their result on the characterization of the maximal solution does not apply to the general situation when the set of singular points has a positive measure. More general uniqueness results for maximal solution have been recently obtained by Camilli et al. [18, 20]. Several papers have followed this approach providing different algorithms to compute the maximal solution, which has been shown to be unique, see [41, 19, 98] and the references therein.

**Remark 1.4** *All the theoretical results mentioned above use the regularity of the grey level function $I$, which is supposed to be (at least) continuous. Naturally, real images do not fit that assumption, even in the case of Lambertian objects. The continuity assumption for $I$ has been removed in paper by Kain and Ostrov [60] and by Prados and Faugeras [89]. Both these papers also contain a scheme and some numerical examples.*

Another approach which produces a global solution to SfS consists in the search for equal-height contours, originally proposed by Bruckstein [16] and later re-introduced by Kimmel and Bruckstein [64, 65]. The method consists of two major steps: the computation of weighted distance functions from all the singular points, using a level set method, and the merging of these surfaces. The algorithm can compute a global solution (which is an a.e. solution) of eikonal equation (1.8) in the reconstruction domain, only combining the local solutions obtained during the first step. Interestingly, this method has been extended to the case of near light source [78].

Finally, the idea of solving the eikonal equation using a power series expansion at a regular point, in the case of a grey level function of class $C^\infty$, has been introduced by Bruss [17] and has been extended to the analytical grey level functions by Durou and Piau [35], which could exhibit a "non-visible deformation" (i.e., a continuous family of analytical shapes giving the same image). This is an important theoretical result but, nevertheless, no algorithm has been derived from this method of resolution.

### 1.7.4. Boundary conditions

We have seen in the previous sections that, although with several differences, all the reflectance models described above are represented by a first order nonlinear Hamilton-Jacobi equation that is described in the more general form as follow

$$H(\mathbf{x}, u(\mathbf{x}), \nabla u(\mathbf{x})) = 0, \qquad \mathbf{x} \in \Omega. \tag{1.98}$$

The use of PDE methods for the resolution of the SfS problem leads necessarily to the definition of some sort of boundary conditions. This is one of the differences with respect to the methods using optimization, since for those methods boundary conditions can be imposed but are not compulsory (see the surveys [34, 124] for more informations concerning the principal properties of these classes). A detailed analysis of the well-posedness of the boundary value problem for nonlinear PDEs in the framework of the weak solutions (in the viscosity sense) can be found in Barles' book [8] and in the references therein. It is important to note that the addition of boundary conditions does not solve the concave/convex ambiguity and that in practical applications boundary conditions are seldom known.

The choice between the different types of boundary conditions is a crucial question: it depends on appropriateness and simplicity, or depends on the additional information available on the object (if any). The Dirichlet boundary condition is typically used when the object is standing on a flat background and the surfaces meets the background at $\partial\Omega$, or if the height on $\partial\Omega$ is known (or assumed, for example by symmetry). Neumann boundary conditions correspond to $\frac{\partial u}{\partial \nu}(\mathbf{x}) = m(\mathbf{x})$, where $\nu(\cdot)$ represents the outward normal to domain $\Omega$. A typical use of it is when we know (or we presume) that the level curves of the surface are orthogonal to the boundary $\partial\Omega$ or to a subset of it where we simply choose $m(\mathbf{x}) = 0$. The Neumann boundary condition gives more freedom in the computation since it only imposes the value of a derivative and does not fix the height of the surface at the boundary. Naturally, also this condition modifies the surface. State constraint boundary conditions differ from the previous ones since they do not impose a value either for the height or for its normal derivative. In this respect, it has been interpreted as a "no boundary condition" choice [88], although this interpretation is rather superficial. In fact, a real function $u$ bounded and uniformly continuous is said to be a "state constraint viscosity solution" if and only if it is a subsolution (in the viscosity sense) in $\Omega$ and a supersolution in $\overline{\Omega}$ (i.e. up to the boundary). It can also be stated as a Dirichlet boundary condition, simply setting $g(\mathbf{x}) = g_c$ on $\partial\Omega$, where $g_c$ is a constant,

provided:

$$g_c > \max_{\mathbf{x} \in \Omega} \{u(\mathbf{x})\}\,.$$

Note that in our problem, this is a mild assumption since we can easily fix an upper bound for the height of the object. The effect of the state constraint boundary condition is to produce solutions that grow inwards from the boundary $\partial \Omega$. This choice can be appropriate in some situations and wrong in other situations, in any case also this boundary condition affects the computation.

# 2. Numerical techniques and a new general convergence theorem

We start this chapter by describing the numerical schemes that we are going to use for our numerical tests on the different models illustrated in Chapter 1. These schemes can be divided into two class of numerical techniques: the methods based on finite difference (FD) approximation and the semi-Lagrangian (SL) method. We will sketch the main characteristics of each technique. After that, we will state and prove a general convergence theorem suitable for a general operator written in a fixed point form.

## 2.1. Finite difference schemes

Let consider the following nonlinear problem in $d$-dimensional space

$$\begin{cases} u_t(x,t) + H(Du(x,t)) = 0, & (x,t) \in \mathbb{R}^d \times [0,T], \\ u(x,0) = u_0(x). \end{cases} \tag{2.1}$$

Once out of the framework of linear equations (and linear schemes), no convergence result of the same generality of Lax-Richtmeyer theorem exists. In practice, while consistency of a scheme can be defined in essentially the same way for a nonlinear scheme, stability becomes a more subtle topic, since uniform boundedness of numerical solutions does not suffice in general for convergence.

Therefore, different nonlinear stability concepts have been developed for nonlinear equations and schemes. In the framework of Hamilton-Jacobi equations (HJEs) two main concepts have been proposed: the more classical idea of monotone stability (which leads to the convergence theorems of Crandall-Lions [28, 29] and Barles-Souganidis [104, 9]) and the idea of uniform semiconcavity used in the Lin-Tadmor convergence theorem [73]. We just recall here the first one. For further details on this issue see [39].

### 2.1.1. Crandall-Lions theorem

The Crandall-Lions theorem is inspired by the result of convergence of monotone conservative schemes for conservation laws, therefore it assumes the scheme to have a structure which parallels the structure of conservative schemes. On the contrary, Barles-Souganidis theorem does not assume any particular structure for the scheme, and is suitable for more general situation (including second-order HJEs), provided a comparison principle holds for the exact equation. It also requires a more technical definition of consistency. We will describe it in the next section regarding SL schemes.

We will present here the Crandall-Lions theorem in the case of two space dimensions. The extension to an arbitrary number of dimensions being straightforward. Hence, we will rewrite (2.1) as

$$u_t + H(u_{x_1}, u_{x_2}) = 0. \tag{2.2}$$

Crandall-Lions theorem works in the framework of difference schemes, so we assume that the space grid is uniform, $\Delta x_i$ being the space step along the $i$-th direction. We define an approximation of the partial derivative $u_{x_i}$ at the point $x_j$ by the right (partial) incremental ratio, that is

$$D_{i,j}[V] = \frac{v_{j+e_i} - v_j}{\Delta x} \qquad (i = 1, 2). \tag{2.3}$$

We give now the definition of a scheme in *differenced form.*

**Definition 2.1** *A scheme $S$ is said to be in differenced form if it has the form*

$$v_j^{n+1} = v_j^n - \Delta t \mathcal{H} \left( D_{1,j-p}[V^n], \ldots, D_{1,j+q}[V^n]; D_{2,j-p}[V^n], \ldots, D_{2,j+q}[V^n] \right) \tag{2.4}$$

*for two multi-indices $p$ and $q$ with positive components and for a Lipschitz continuous function $\mathcal{H}$ (called the numerical Hamiltonian).*

In practice, (2.4) defines schemes in which the dependence on $V^n$ appears only through its finite differences, computed on a rectangular stencil of points around $x_j$. The differenced form of a scheme lends itself to an easier formulation of the consistency condition, which is given in the following

**Definition 2.2** *A scheme in differenced form is consistent if, for any $a, b \in \mathbb{R}$,*

$$\mathcal{H}(a, \ldots, a; b, \ldots, b) = H(a, b). \tag{2.5}$$

This last definition matches the usual one (refer to [39] for details on it).

Note that, in the nonlinear case, we expect that monotonicity may or may not hold depending on the speed of propagation of the solution, this speed being related to the Lipschitz constant of $u_0$ (in fact, in general monotonicity does depend on the speed of propagation, but in the linear case this speed is given and unrelated to $u_0$). We will say that the scheme is monotone on $[-R, R]$ if the usual stability condition is satisfied for any $V$ and $W$ such that $|D_{i,j}[V]|, |D_{i,j}[W]| \leq R$.

Now, we have all elements to state the Crandall-Lions convergence theorem.

**Theorem 2.1.1 (Crandall-Lions)**
*Let $H : \mathbb{R}^2 \to \mathbb{R}$ be continuous, let $u_0$ in (2.1) be bounded and Lipschitz continuous (with Lipschitz constant L) on $\mathbb{R}^2$, and let $v_j^0 = u_0(x_j)$. Let the scheme (2.4) be monotone on $[-(L+1), L+1]$ and consistent for a locally Lipschitz continuous numerical Hamiltonian $\mathcal{H}$. Then, there exists a constant $C$ such that, for any $n \leq T/\Delta t$ and $j \in \mathbb{Z}$,*

$$\left| v_j^n - u(x_j, t_n) \right| \leq C \Delta t^{1/2} \tag{2.6}$$

*as $\Delta t \to 0$, $\Delta x_i = \lambda_i \Delta t$ $(i = 1, 2)$.*

**Remark 2.1** *In what follows, we will give an example of scheme in one dimensional. Hence, we will apply this convergence result to first-order, three-point stencil schemes in one space dimension. In this particular setting, a differenced scheme takes the simpler form (in which the index $i$ of the variable is clearly omitted)*

$$v_j^{n+1} = v_j^n - \Delta t \mathcal{H} \left( D_{j-1}[V^n], D_j[V^n] \right), \tag{2.7}$$

*involving only the points $v_{j-1}^n$, $v_j^n$ and $v_{j+1}^n$ in the computation of $v_j^{n+1}$.*

### 2.1.2. An example: the Lax-Friedrichs scheme

Let consider the following one-dimensional problem

$$\begin{cases} u_t(x,t) + H(u_x(x,t)) = 0, & (x,t) \in \mathbb{R} \times [0,T], \\ u(x,0) = u_0(x). \end{cases} \tag{2.8}$$

We will make the standing assumptions that $H$ is convex and that there exists $\alpha_0 \in \mathbb{R}$ such that

$$\begin{cases} H'(\alpha) \leq 0 & \text{if } \alpha \leq \alpha_0, \\ H'(\alpha) \geq 0 & \text{if } \alpha \geq \alpha_0. \end{cases} \tag{2.9}$$

We also define

$$M_{H'}(L) = \max_{[-L,L]} |H'|, \tag{2.10}$$

which corresponds to the maximum speed of propagation of a solution with Lipschitz constant $L$.

We will see now the Lax-Friedrichs scheme adapted to the nonlinear case, listing the definitions and the convergence theorem for it.

**Construction of the Lax-Friedrichs scheme.**
The simplest way to recast Lax-Friedrichs scheme for the HJE is to define it in the form

$$v_j^{n+1} = \frac{v_{j-1}^n + v_{j+1}^n}{2} - \Delta t H(D_j^c[V^n]), \tag{2.11}$$

where $D_j^c[V^n]$ is the centered difference at $x_j$ defined by

$$D_j^c[V^n] = \frac{v_{j+1}^n - v_{j-1}^n}{2\Delta x} = \frac{D_{j-1}[V^n] + D_j[V^n]}{2}. \tag{2.12}$$

This definition of the LF scheme completely parallels the linear case, and is also suitable to be treated in the framework of the Crandall-Lions theorem. In fact, once we recall that

$$\frac{v_{j-1}^n + v_{j+1}^n}{2} = v_j^n + \frac{\Delta x}{2} \left( D_j[V^n] - D_{j-1}[V^n] \right), \tag{2.13}$$

(2.11) can be written in the differenced form

$$v_j^{n+1} = v_j^n - \Delta t \mathcal{H}^{LF}(D_{j-1}[V^n], D_j[V^n]), \tag{2.14}$$

by setting

$$\mathcal{H}^{LF}(\alpha, \beta) = H(\frac{\alpha + \beta}{2}) - \frac{\Delta x}{2\Delta t}(\beta - \alpha). \tag{2.15}$$

**Consistency of the Lax-Friedrichs scheme.**
The LF scheme (2.11) satisfies the consistency condition (2.5), and in fact

$$\mathcal{H}^{LF}(a, a) = H\left(\frac{a + a}{2}\right) = H(a). \tag{2.16}$$

Consistency is therefore satisfied.

**Stability of the Lax-Friedrichs scheme.**
We examine the issues of CFL condition and monotonicity, which in this case give the same restriction on the discretization steps.

**CFL condition for the Lax-Friedrichs scheme**   Taking into account that the maximum speed of propagation is $M_{H'}(L)$, the CFL condition reads as

$$\frac{M_{H'}(L_V)\Delta t}{\Delta x} \leq 1. \tag{2.17}$$

This condition is necessary and sufficient, since it also ensures monotonicity (as we will soon show).

**Monotonicity of the Lax-Friedrichs scheme.**
Considering the LF scheme in the form (2.11). Clearly, the $j$-th component $S_j^{LF}(\Delta; V)$ depends only on the values $v_{j\pm 1}$, so that

$$\frac{\partial}{\partial v_i} S_j^{LF}(\Delta; V) = 0 \qquad (i \neq j \pm 1). \tag{2.18}$$

On the other hand, if $i = j \pm 1$, we have

$$\begin{aligned}
\frac{\partial}{\partial v_{j\pm 1}} S_j^{LF}(\Delta; V) &= \frac{1}{2} - \Delta t H'\left(D_j^c[V]\right) \frac{\partial D_j^c[V]}{\partial v_{j\pm 1}} \\
&= \frac{1}{2} \mp \frac{\Delta t}{2\Delta x} H'\left(D_j^c[V]\right),
\end{aligned} \tag{2.19}$$

where we have used the fact that

$$\frac{\partial D_j^c[V]}{\partial v_{j\pm 1}} = \pm \frac{1}{2\Delta x}. \tag{2.20}$$

Therefore, if $L_V$ is the Lipschitz constant of the sequence $V$, the scheme is monotone, provided

$$\frac{\Delta t}{\Delta x} \leq \frac{1}{M_{H'}(L_V)}. \tag{2.21}$$

**Convergence result for the Lax-Friedrichs scheme.**
The convergence result is obtained from consistency (2.16) and monotonicity (2.21) and applying Theorem 2.1.1.

**Theorem 2.1.2** *Let $H$ satisfy the basic assumptions, let $u_0 \in W^{1,\infty}(\mathbb{R})$, let $u$ be the solution of (2.8) with $L$ as its Lipschitz constant, and let $v_j^n$ be defined by (2.11) with $v_j^0 = u_0(x_j)$.*
*Then, for any $j \in \mathbb{Z}$ and $n \in [1, T/\Delta t]$,*

$$\left| v_j^n - u(x_j, t_n) \right| \leq C \Delta t^{1/2} \tag{2.22}$$

*as $\Delta \to 0$, with $M_{H'}(L+1)\Delta t \leq \Delta x$.*

## 2.2. Semi-Lagrangian schemes

In this section we will focus our attention on semi-Lagrangian (SL) schemes which have shown to be very effective for first order problems since it tries to mimic at the discrete level the method of characteristics (see [39] for more details). Other approaches based on finite difference or finite volumes are feasible. As we have seen there are basically two main problems related to the vertical light case and the oblique light case. In the vertical case, we have to solve an eikonal-type equation for each model (Cf. (1.8), (1.36), (1.61), (1.78)). In the oblique case, we get more general first-order Hamilton-Jacobi equations (HJEs) (1.7), (1.34), (1.48), (1.69) where the nonlinear term is also coupled with linear terms. The general framework for these type of problems is the theory of viscosity solutions which guarantees (under appropriate assumptions) existence and uniqueness results for the vertical light case. A similar approach can also be applied to the case of an oblique light source when the surface is not smooth and black shadows are present in the image. It should be noted that to have uniqueness one has to add additional assumptions or to add more informations, like the height at maximum brightness points or the fact that we select to approximate the maximal solution (as introduced in [18]). General convergence results for the approximation scheme to the maximal solution of the degenerate eikonal equation can be found in [34, 19]. There are two types of algorithms based on the semi-Lagrangian approach. The first type of algorithm is global and gives an approximation of the fixed point problem on the whole grid at every iteration till the stopping rule is satisfied. Some acceleration methods, like the Fast Sweeping method ([62, 61]) can be introduced to speed up convergence. The second type of methods is local and tries to concentrate the numerical effort only in a neighborhood of a region which is considered to be already exact (the so called Accepted region). The Fast Marching method (extensively described in [102, 30]) is a typical example of this class of methods. We will briefly describe them in the next chapter.

The algorithms corresponding to the models presented in the Chapter 1 compute the maximal solution in the domain without additional information of the surface and with a single boundary condition which can be either homogeneous $u = 0$ or not (but to set $u = g$ on the boundary of the mask one has to know or guess the right solution there).

As already stated in Section 1.1, we suppose a surface given as a graph. In the case of vertical light, for such a surface we do not have shadows covering an open domain (i.e. the points where $I(\mathbf{x}) = 0$ are either isolated or curves in the plane). If the light is oblique, we usually have shadows so that we can divide the support of the surface (the domain of $u$) into two regions, $\Omega_l \equiv \{\mathbf{x} : I(\mathbf{x}) > 0\}$ and $\Omega_s \equiv \{\mathbf{x} : I(\mathbf{x}) = 0\}$, which represent respectively the "light" and the "shadow" regions. Typically they have both nonempty interior and, naturally, $\Omega = \Omega_l \cup \Omega_s$. Note that $\Omega$ now represents the new mask which also includes black regions. Moreover, we assume that $\Omega \subset \overline{Q}$, where $Q$ is the rectangular domain corresponding to the image.

In $\Omega_l$ the equation is always the same, whereas in the "shadow" region the surface can have any shape since the model is naturally not able to describe the real surface there. This is why other authors have included boundary conditions (e.g., Neumann boundary conditions) on $\partial\Omega_l$ to treat the problem in $\Omega_l$ just ignoring the region $\Omega_s$. This can in turn create difficulties in the construction of the numerical algorithm since the boundary of $\Omega_l$ can be nonsmooth and somewhat difficult to follow. We include the region $\Omega_s$ in the computation just defining there a conventional surface to replace the unknown surface. We will substitute for the surface the "separation surface" (or "shadow surface") i.e. the surface separating light from shadow. That surface will depend on $\boldsymbol{\omega}$ and on the normal direction to the original surface. This means that in $\Omega_s$, for example for the L–model, we have to solve the equation

$$\widetilde{\boldsymbol{\omega}} \cdot \nabla u(\mathbf{x}) - \omega_3 = 0, \qquad \mathbf{x} \in \Omega_s, \tag{2.23}$$

that coincides with (1.7) since $I = 0$ in $\Omega_s$. Then, we can use the same equation everywhere in $Q$ and we do not need to introduce any boundary condition on $\partial\Omega_l$.

Now, let us see in detail how the scheme works. As illustrative case we will describe the process for a specific model, for example the L–model.

We can rewrite the Lambertian eikonal equation (1.8) as

$$\begin{cases} \max_{\boldsymbol{a}\in B_2(0,1)} \{-\boldsymbol{a} \cdot \nabla u\} = f(\mathbf{x}) & \text{for } \mathbf{x} \in \Omega \subset \mathbb{R}^2, \\ u(\mathbf{x}) = g(\mathbf{x}) & \text{for } \mathbf{x} \in \partial\Omega, \end{cases} \tag{2.24}$$

where $B_2(0,1)$ is the $\mathbb{R}^2$ unit ball and $f(\mathbf{x})$ is given by (1.9). In fact, it is sufficient to note that the maximum is achieved for $\bar{\boldsymbol{a}} \equiv \nabla u/|\nabla u|$. Without loss of generality, we can suppose that $u \geq 0$ and, with the appropriate change of variable $\mu v(\mathbf{x}) = \frac{1-\exp(-\mu u(\mathbf{x}))}{\mu}$, the equation (2.24) can be rewrited as follow

$$\begin{cases} \mu v(\mathbf{x}) + \max_{\boldsymbol{a}\in B_2(0,1)} \left\{-\frac{\boldsymbol{a}}{f(\mathbf{x})} \cdot \nabla v(\mathbf{x}) - 1\right\} = 0 & \text{for } \mathbf{x} \in \Omega, \\ v(\mathbf{x}) = 1 - \exp(g(\mathbf{x})) & \text{for } \mathbf{x} \in \partial\Omega. \end{cases} \tag{2.25}$$

Here $\mu$ is a free positive parameter without a specific physical meaning, but it is important because varying its value it is possible to modify the slope. In fact, the slope increases for increasing values of $\mu$. Note that by definition $0 \leq v(\mathbf{x}) \leq 1/\mu$. For the vertical case we will set $\mu = 1$.

This equation admits a unique solution provided that $f(\mathbf{x})$ is non-zero and bounded in $\Omega$ [6].

Let us consider the equation (2.25), we suppose that the set $\Omega$ is entirely contained in the rectangular domain $Q$ corresponding to the image, and that on this image is defined a discrete grid $G$ (size $n \times m$) of points $\{x_{ij}\}$ with $i, j \in \mathbb{Z}$. We denote by $G_{in}$ the set of grid points belonging to $\Omega$ and by $G_{out}$ the set $G \backslash G_{in}$. We can write the scheme corresponding to the problem (2.25) with $\mu = 1$ as follow:

$$\begin{cases} w(x_{ij}) = \min\limits_{\mathbf{a} \in \mathrm{B}_2(0,1)} \left\{ e^{-h} w(x_{ij} + h \frac{\mathbf{a}}{f(\mathbf{x})}) \right\} + 1 - e^{-h} & \text{for } i, j \in G_{in}, \\ w(x_{ij}) = 1 - \exp(-g(x_{ij})) & \text{for } i, j \in G_{out}, \end{cases} \quad (2.26)$$

where $h$ is a small parameter.

### Fixed point Scheme.

Denoting by $W$ the matrix containing the values of $w(x_{ij})$, we can rewrite the equation as

$$W = T(W), \quad T \colon \mathbb{R}^{n \times m} \to \mathbb{R}^{n \times m}.$$

In [7, 40] it has been proved that the numerical solution to (2.26) exists and it is unique by a fixed point argument on the iteration

$$W^{k+1} = T(W^k).$$

Moreover, we can describe in general the interpolated value as $\Lambda(W)$ and the operator $T$ as the operator

$$T(W) = \min\{e^{-h} \Lambda(W)\} + 1 - e^{-h}.$$

Such a scheme is a fixed point scheme during which we search the matrix $W^\star$ such that

$$W^\star = T(W^\star).$$

It means that the solution is computed for successive iterations, computing a new matrix $W^{k+1}$ by applying the operator $T$ to the previous matrix $W^k$. In fact, the monotony of $T$ implies that if we start from a sub-solution $W^0$ the sequence will monotonically converge to the fixed point. It should be noted that the presence of singular points makes the vectorfield $\mathbf{a}/f(\mathbf{x})$ unbounded since $f$ vanishes. In order to overcome this difficulty, the minimum of $f(\mathbf{x})$ is truncated to a value $\varepsilon$, that is, $f(x_{ij})$ is approximated with

$$f_\varepsilon(x_{ij}) = \max\{f(x_{ij}), \varepsilon\}.$$

Even with the introduction of this perturbation the solution converges to the original not perturbed problem in the limit of $\varepsilon$ going to 0 (see [20]). The condition for the convergence to the maximal solution [34, 98] is given by

$$h\left|\frac{1}{f_\varepsilon}\right|_\infty \leq \delta. \tag{2.27}$$

**The stopping rule.**
In iterative methods, the method stops when you have reached the required tolerance or when you have exceeded the maximum number of iterations allowed. In an iterative method of fixed point, the point is reached when

$$\|W^{n+1} - W^n\| \leq \eta,$$

for some appropriate norm and a tolerance factor $\eta$.

With the same presuppositions of the vertical light case, regarding the problem (1.12) in the variable $v$, assuming that the set $\Omega$ is entirely contained in the rectangular image, and by using the same definition of $G$, $G_{in}$ and $G_{out}$ introduced just above, also in the case of oblique light source we seek solutions $w(x_{ij})$ in the space of the linear approximations on the grid points. With these definitions, the fully discrete scheme of (1.12) is

$$\begin{cases} w(x_{ij}) = \min\limits_{\boldsymbol{a}\in\partial B_3} \left\{ e^{-\mu h} w\left(x_{ij} + h\,b^L(x_{ij}, \boldsymbol{a})\right) - \tau\frac{I(x_{ij})a_3}{\omega_3}\left(1 - \mu w(x_{ij})\right) \right\} + \tau, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{for } i,j \in G_{in}, \\ w(x_{ij}) = 0, \qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{for } i,j \in G_{out}, \end{cases} \tag{2.28}$$

where $x_{ij}$ are the grid nodes, $\tau = (1 - e^{-\mu h})/\mu$ and

$$b^L(x_{ij}, \boldsymbol{a}) = \frac{1}{\omega_3}\left(I_D(x_{ij})a_1 - \omega_1, I_D(x_{ij})a_2 - \omega_2\right). \tag{2.29}$$

The operator

$$T_i^L(W) = \min\limits_{\boldsymbol{a}\in\partial B_3} \left\{ e^{-\mu h} w(x_{ij} + h\,b^L(x_{ij}, \boldsymbol{a})) - \tau\frac{I(x_{ij})a_3}{\omega_3}(1 - \mu w(x_{ij})) \right\} + \tau \tag{2.30}$$

is a contraction mapping [42] in $[0, 1/\mu)^G$ if

$$I(x_{ij})\bar{a}_3 < \mu\omega_3, \tag{2.31}$$

where $\bar{a}_3 \equiv arg\min\limits_{a\in\partial B_3}\left\{ e^{-\mu h} w(x_{ij} + hb^L(x_{ij}, a)) - \tau\frac{I(x_{ij})a_3}{\omega_3}(1 - \mu w(x_{ij})) \right\}$.

As done for the equation (2.26), we can rewrite the equation (2.28) in a fixed point form as

$$W = T(W), \quad T\colon \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m}.$$

In addition to a stopping rule for the iterative process, the ingredients of a SL scheme are: a grid, an interpolation, a procedure for the search of the minimum value. Let's see in detail the possible choices for each ingredient.

### 2.2.1. Types of grids: structured or unstructured

A grid is called *structured* when the numbering of the nodes belonging to each cell can be obtained through simple algebraic operations. The advantages of this kind of grid is that you can write more efficient algorithms using direct addressing the memory variables. The disadvantage is the difficulty to generate grids of good quality on very complex geometric configurations. Not are suitable for local thickening of grid nodes (grid adaptivity). In order to remedy in part to the geometric limitations of structured grids is used sometimes block-structured grids or multiblock, obtained by partitioning a priori the domain in subdomains (blocks) of simpler geometry and using a structured grid in each block. The disadvantage in this case is the request of complex data structures in order to obtain an efficient management of subdomains and their interconnections. Moreover, the division into blocks is hardly automatable.

The unstructured mesh can be or not hybrid (flat, composed of triangular or quadrilateral elements, or three-dimensional, consisting of tetrahedral or hexahedral). In the unstructured meshes is not easy to carry out a numbering of the nodes but we proceed to the numbering following a predetermined order. Despite the need "to store" nodes, these meshes can reproduce complex geometries and are very flexible. For this reason they are currently the mesh most used for the calculations structural.

The most popular algorithms used for unstructured grid composed of triangular elements are the triangulation of Delaunay and the technique of advancement of the front. For more details on these algorithms see [46, 110].

In this thesis we will use a structured grid with space steps fixed.

### 2.2.2. Types of interpolation

Interpolation is a method of constructing new data points within the range of a discrete set of known data points. The scheme described above is not fully discrete. In fact, $x_{ij} + h\,\boldsymbol{a}/f(\mathbf{x})$ is not in general a point belonging to the grid. The values of $w(x_{ij} + h\,\boldsymbol{a}/f(\mathbf{x}))$ must be find through an interpolation process.

There exist several types of interpolation:

- the *Piecewise constant interpolation*, the simplest interpolation method that consists into locate the nearest data value, and assign the same value. In simple problems, this method is unlikely to be used, as linear interpolation (see below) is almost as easy, but in higher-dimensional multivariate interpolation, this could be a favourable choice for its speed and simplicity, but not for accuracy.

- the *Linear interpolation*, that takes two data points, say $(x_a, y_a)$ and $(x_b, y_b)$, and

the interpolant at the point $(x, y)$ is given by:

$$y = y_a + (y_b - y_a)\frac{x - x_a}{x_b - x_a}. \tag{2.32}$$

This kind of interpolation is quick and easy, but it is not very precise. In fact, the error is proportional to the square of the distance between the data points.

- the *Bilinear interpolation*, that is an extension of linear interpolation for interpolating functions of two variables (e.g., $x$ and $y$) on a regular $2D$ grid. The key idea is to perform linear interpolation first in one direction, and then again in the other direction. Although each step is linear in the sampled values and in the position, the interpolation as a whole is not linear but rather quadratic in the sample locations (details below). The result of bilinear interpolation is independent of which axis is interpolated first and which second. If we had first performed the linear interpolation in the y-direction and then in the x-direction, the resulting approximation would be the same.

  We will consider the bilinear interpolation which needs of the coefficients $(a, b, c, d) \in \mathbb{R}^4$ to express the function value $w(x_{ss}) = w(x_s, y_s) = w(x_{ij} + h\,\mathbf{a}/f(\mathbf{x}))$ in the bilinear combination $ax_s y_s + bx_s + cy_s + d$. We chose this last kind of interpolation because each point $(x_s, y_s) \in G_{in}$ falls inside a cell of four nodes and we assumed as the reference vertex the one in the bottom on the left that we will call $(x_k, y_k) \in G_{in}$ (see Fig. 2.1). If we impose the interpolation condition for the
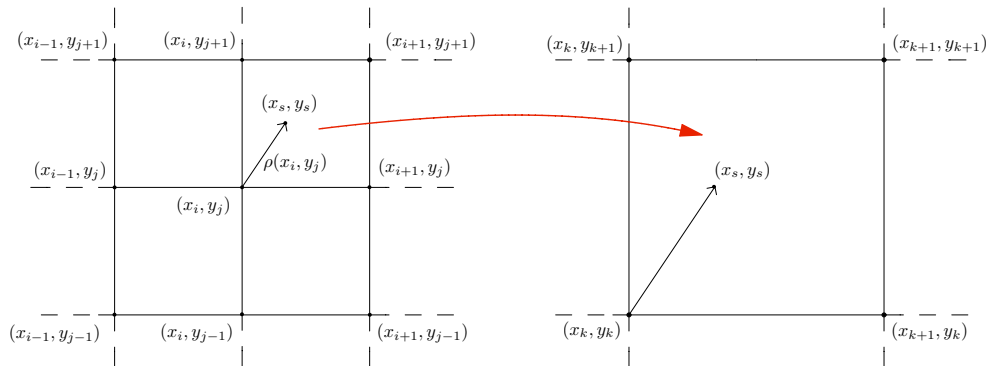


Figure 2.1.: The picture explains the position of the different points for the bilinear interpolation to determine the value $w(x_s, y_s)$ in the case where the two components of the vector field $\rho(x_i, y_j)$ are both positive. On the left, the part of the lattice centered on $(x_i, y_j)$. On the right, the zooming of the interesting square with the reference coordinate.

nearest four points of $(x_s, y_s)$ inside this cell is thus possible to obtain the following

system:

$$\begin{cases} ax_k y_k + bx_k + cy_k + d = w(x_k, y_k) \\ ax_{k+1}y_k + bx_{k+1} + cy_k + d = w(x_{k+1}, y_k) \\ ax_{k+1}y_{k+1} + bx_{k+1} + cy_{k+1} + d = w(x_{k+1}, y_{k+1}) \\ ax_k y_{k+1} + bx_k + cy_{k+1} + d = w(x_k, y_{k+1}) \end{cases} \qquad (2.33)$$

that is

$$\begin{pmatrix} x_k y_k & x_k & y_k & 1 \\ x_{k+1}y_k & x_{k+1} & y_k & 1 \\ x_{k+1}y_{k+1} & x_{k+1} & y_{k+1} & 1 \\ x_k y_{k+1} & x_k & y_{k+1} & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} w(x_k, y_k) \\ w(x_{k+1}, y_k) \\ w(x_{k+1}, y_{k+1}) \\ w(x_k, y_{k+1}) \end{pmatrix}. \qquad (2.34)$$

Let us emphasize an interpolation results based on the uniqueness of bilinear interpolation on this kind of data:

**Lemma 2.2.1** *The system (2.34) admits a unique solution.*

*Proof.*
The proof is simply based on the computation of the determinant of the matrix. We want to prove that the necessary condition to make it be zero is not possible if we consider a discretization like our.

$$\det \begin{pmatrix} x_k y_k & x_k & y_k & 1 \\ x_{k+1}y_k & x_{k+1} & y_k & 1 \\ x_{k+1}y_{k+1} & x_{k+1} & y_{k+1} & 1 \\ x_k y_{k+1} & x_k & y_{k+1} & 1 \end{pmatrix} =$$

$$= -(x_{k+1}y_k x_{k+1}y_{k+1} + x_{k+1}y_{k+1}x_k y_{k+1} + x_{k+1}y_{k+1}x_k y_k -$$
$$- x_k y_{k+1}x_{k+1}y_k - x_{k+1}y_{k+1}x_{k+1}y_{k+1} - x_k y_{k+1}x_{k+1}y_k) +$$
$$+ x_k y_k x_{k+1}y_{k+1} + x_k y_{k+1}x_k y_{k+1} + x_{k+1}y_{k+1}x_k y_k$$
$$- x_k y_{k+1}x_{k+1}y_k - x_{k+1}y_{k+1}x_k y_{k+1} - x_k y_{k+1}x_k y_k -$$
$$- (x_k y_k x_{k+1}y_{k+1} + x_k y_k x_k y_{k+1} + x_{k+1}y_k x_k y_k -$$
$$- x_k y_{k+1}x_{k+1}y_k - x_{k+1}y_k x_k y_{k+1} - x_k y_k x_k y_k) +$$
$$+ x_k y_k x_{k+1}y_{k+1} + x_k y_k x_{k+1}y_{k+1} + x_{k+1}y_k x_{k+1}y_k -$$
$$- x_{k+1}y_{k+1}x_{k+1}y_k - x_{k+1}y_k x_k y_{k+1} - x_{k+1}y_k x_k y_k =$$
$$= -2x_{k+1}^2 y_k y_{k+1} - 2x_k x_{k+1}y_{k+1}^2 + x_{k+1}^2 y_{k+1}^2 + 4x_k x_{k+1}y_k y_{k+1} +$$
$$+ x_k^2 y_{k+1}^2 - 2x_k^2 y_k y_{k+1} - 2x_k x_{k+1}y_k^2 + x_k^2 y_k^2 + x_{k+1}^2 y_k^2 =$$
$$= x_{k+1}^2(y_k - y_{k+1})^2 + x_k^2(y_k - y_{k+1})^2 - 2x_k x_{k+1}(y_k - y_{k+1})^2 =$$
$$= (y_k - y_{k+1})^2(x_k - x_{k+1})^2.$$

Therefore the matrix has the determinant equal to zero if and only if

$$y_k = y_{k+1} \quad \text{or} \quad x_k = x_{k+1},$$

that is not true. Then it concludes the proof.

□

Let us call the unique solution of the previous system $(a^*, b^*, c^*, d^*)$ for every point $(x_i, y_j) \in G_{in}$. Now it is possible to calculate the bilinear form in the evaluation point $(x_s, y_s)$, that is:

$$a^* x_s y_s + b^* x_s + c^* y_s + d^* = w(x_s, y_s) = w(x_{ij} + h \, \rho_{ij}), \qquad (2.35)$$

where $\rho_{ij} := \rho(x_i, y_j) = \boldsymbol{a}/f(\mathbf{x})$.

### 2.2.3. Search of the minimum

The search of the minimum is a very important building block of the SL schemes for nonlinear problems such as our problem. There exist several methods of minimization suitable for function to be minimized that has no explicit form, so we will briefly sketch here methods which minimize a function $f$ of $N$ variables without making use of the analytic expressions of the gradient that is unknown.

*Direct search methods.*
This class of methods generally collects methods of heterogeneous nature, which share the feature of minimizing a function without using, either in exact or approximate form, its gradient. Direct search methods rather proceed by computing the function at a discrete, countable set of points. The most common schemes of this class are based on the *simplex* technique (not to be confused with the algorithm used in Linear Programming), whose basic idea dates back to the '60s. Let a simplex of vertices $V_1, \ldots, V_m$ be constructed in $\mathbb{R}^N$. Once computed the corresponding values $y_1, \ldots, y_m$ of $f$ at the vertices, the highest value $y_h$ is selected, and the corresponding vertex $V_h$ is reflected with respect to the centroid $C$ of the simplex made by the remaining vertices. Then, the new simplex obtained by replacing the node $V_h$ with $V_r$ is accepted if the value of the function at $V_r$ is not the highest value in the new simplex. Otherwise, the second highest value of the original simplex is selected, and the procedure is repeated. If all possible updates of the simplex lead to finding the highest value on the reflected node, the algorithm stops. The size of the simplex should then be considered as a tolerance in locating the minimum, and to refine the search the algorithm should be restarted with a simplex of smaller size. For further details see [69].

*Descent methods (without derivatives).*
In these methods, an initial estimate of the minimum point $x_0$ is given, and the updating of the approximation $x_k$ is performed via the formula $x_{k+1} = x_k + \beta_k d_k$, that is, the scheme moves from $x_k$ to $x_{k+1}$ along the direction $d_k$ with step $k$. As a rule, the direction $d_k$ should be a descent direction, i.e. $(d_k, \nabla f(x_k)) < 0$, $\beta_k$ should be positive, and $f(x_{k+1}) < f(x_k)$. Neglecting for the moment the problem of avoiding the computation of derivatives, the most typical scheme of this class is the *method of steepest descent* (or

*gradient method*), in which $d_k = -\nabla f(x_k)$ and $\beta_k$ is chosen on the basis of an *exact line search*, that is, $f(x_{k+1}) = \min_{\beta}[x_k - \beta\nabla f(x_k)]$. Such a choice of the direction $d_k$ gives locally the largest negative directional derivative, and the original formulation of this method requires a high complexity to determine $\beta_k$. A first improvement consists in accepting a certain $\beta$ if it ensures that the function "decreases enough" passing from $x_k$ to $x_{k+1}$. A further improvement is to choose more efficient search directions $d_k$, e.g. via the Newton's method or the conjugate direction methods (see [39] for more details).

To turn back to the problem of implementing schemes of this class without computing derivatives, we remark that in principle it is possible to replace gradient of $f$ by their finite difference approximations, although the complexity of this operation is $N+1$ computations of $f$. Whenever the computation of $f$ has a critical complexity (and this is definitely the case when using minimization algorithms within a SL scheme), the minimization is better accomplished by cheaper techniques.

*Powell's method.*

This method is genuinely derivative-free, but search directions are rather computed exploiting a geometric property of quadratic functions, which allows to construct conjugate directions without computing the gradient, nor the hessian. The basic idea in the case of two space dimensions is the following: let a quadratic function $f$ (shown through its elliptical level curves) and a search direction $d_1$ be given. Consider the minima of $f$ along two parallel lines with direction $d_1$ (we recall that at a line-constrained minimum the search direction is tangent to the level curve). Then, the direction $d_2$ through the two minima is conjugate with $d_1$. The technique can be extended to generate conjugate directions for a generic dimension $N$, and produces a scheme in the form of a descent method. Being a conjugate direction method, Powell's method converges in $N$ iterations for quadratic functions of $N$ variables, and is expected to be superlinear for smooth nonquadratic functions. For technical details see [87].

*Trust-region methods based on quadratic models.*

In this class of schemes, the iterative minimization of the function $f$ is carried out by working at each step on an approximate (typically, quadratic) model of the function, which is associated to a so-called *trust region*, which is a set in which the model is considered as a good approximation of the function. Let $x_k$ denote the current iterate. We write the quadratic model for the function $f$ at $x_k$ as $m_k(x_k + s) = f(x_k) + (g_k, s) + 1/2(H_k s, s)$ and the trust region as the ball $\mathbb{B}_k = \{x \in \mathbb{R}^N : ||x - x_k|| \leq \Delta_k\}$. Although the vector $g_k$ and the (symmetric) matrix $H_k$ play the role of respectively the gradient and the hessian of $f$, their construction should not use any information on derivatives. In practice, they are typically constructed by interpolating a certain set of previous iterates where the function $f$ has already been computed. Once the model has been constructed at the iterate $x_k$, a new candidate point $x_k^+$ is computed as the minimum point for $m_k$ in the trust region. If the value $f(x_k^+)$ represent a sufficient improvement of $f(x_k)$, then the iterate is updated as $x_{k+1} = x_k^+$, and typically a former point is removed from the set of points used in the interpolation. If not, the trust region is reduced by decreasing

$\Delta_k$, and/or new points are added to improve the quality of the approximation. The robustness of the algorithm is crucially related to the geometric properties of the set of points used to construct the model, so various recipes have been proposed to generate this set, to iteratively update it, and to improve it whenever the algorithm is unable to proceed. See [24] for more details.

### 2.2.4. Strategy of search for the minimum with less controls

In the equation (2.26) and in the other similar equations, the search of the minimum consists in search for the minimum value obtained by interpolating on a grid of discrete points. The general method is to use the control $\boldsymbol{a}$ to approximate the values of $w(x_{ij} + h\,\boldsymbol{a})$. The value of these controls is in the domain of minimization and can be limited to an appropriate choice in the set of unit vectors, as in our case. Obviously, the total number of controls to be used must be finite, and the most suitable choice depends, apart from the need to maintain limited the time required to compute, also by the choice of the interpolation type and by the need to obtain a better approximation of the minimum. The general method consists to generate all the possible controls $\boldsymbol{a}$, excluding the controls that bring the point off the grid, or those that do not satisfy other criteria such as the contraction condition (2.31) for the model with oblique light, and then choose the minimum between all the values generated by the controls. For the equations like (2.26), in the case of linear approximation [30], it is possible to compute a lower number of controls on the border of the unit ball $\partial B_2(0, 1)$.

**Vertical light case**

Let us consider, for simplicity, a Cartesian grid with step $\Delta x = \Delta y = 1$. In such a grid, for each point $x_{ij}$ we consider the four triangles formed by the itself point and its four neighbors $\{x_{i,\pm j}, x_{\pm i,j}\}$ (see Fig. 2.2). By choosing an appropriate coordinate system, such that the point $(i, j, w_{ij})$ coincides with the origin $O$, we consider the three vertices of the triangle $T_1$, that is $\mathbf{O} = (0, 0, 0)$, $\mathbf{A} = (1, 0, A)$, $\mathbf{B} = (0, 1, B)$. The equation of the plane passing through these three points is $z = -(Ax + By)$. The minimum of $w$ on such a plane is in the intersection of the base cylinder $h\boldsymbol{a}$ and the plan itself. The equation of the cylinder with axis $z$ and radius $h$ is

$$\begin{cases} x & = h \cos \theta, \\ y & = h \sin \theta, \\ z & = \zeta. \end{cases}$$

Its intersection with the plane passing for $\mathbf{OAB}$ is

$$z = -h\,(A \cos \theta + B \sin \theta),$$
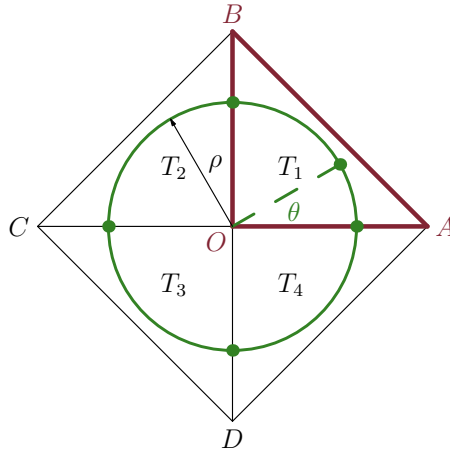
Figure 2.2.: By choosing an appropriate coordinate system, such that the point $(i, j, w_{ij})$ coincides with the origin **O**, the minimum of $w(x_{ij} + h\,\boldsymbol{a})$ is located on the intersection of the planes defined by the triangles $T_i$ with the cylinder of radius $h$ and center **O**. The 5 points where it can be located the minimum are marked with thicker points.

whose stationary points in $\theta$ are given by

$$\frac{d}{d\theta}\{-h\,(A\,\cos\theta + B\,\sin\theta)\} = 0,$$

that is

$$\theta = \arctan B/A,$$

from which, considering only the triangle $T_1$, being

$$\sin(\arctan B/A) = \frac{B}{\sqrt{A^2 + B^2}}, \quad \cos(\arctan B/A) = \frac{A}{\sqrt{A^2 + B^2}},$$

we obtain the stationary point

$$z = -h\,\sqrt{A^2 + B^2}, \tag{2.36}$$

value that we have to compare with the values at the extremes

$$z = h\,A, \quad z = h\,B. \tag{2.37}$$

The stationary point $z = -h\,\sqrt{A^2 + B^2}$ is actually a minimum point in $T_1$ if $B$ and $A$ are both less to 0, i.e. $\theta = \arctan B/A$ is in $(0, \pi/2)$.

**Remark 2.2** *Interpreting in a more restrictive way the condition* (2.27)*, we can further simplify the problem by requiring that* $h|1/f_\varepsilon|_\infty \le \delta/\sqrt{2}$*, so that the interpolation is always carried out between the next neighbors of the point* $x_{ij}$*, i.e. so that the circumfer-*

*ence $\partial B_2(0, h)$ is always contained in the rhombus having as vertices the four neighbors of the point $x_{ij}$.*

Summarizing, for each of the four triangles $T_i$ we compute the points $z_1 = -h\sqrt{A^2 + B^2}$, $z_2 = hA$, $z_3 = hB$ and finally we choose the minimum between all the values computed. In this way, we have reduced the needed number of controls to 5 restricting at the only linear interpolation. Note that being only interested in the value of the minimum and not to where it is carried on the grid, we can change the problem transforming appropriately each of the remaining triangles, so that it is equivalent to that just solved for the triangle $T_1$. Furthermore, noting that the minimum given by (2.36) is achieved when $A$ and $B$ are both negative, we can in advance search the triangle with the minimum values of the vertices and perform the computation of the minimum only on this triangle by using equation (2.36) and on the two intersections with the axes by using (2.37), thus cutting the number of the controls to a maximum of 3. Reintroducing the steps of the grid, the solution of the equation (2.26) for each grid point is the minimum value between

$$ w_{ij} = \beta(-h\sqrt{A^2 + B^2}\,\sqrt{2}/\delta + w_{ij}) + 1 - \beta, $$

$$ w_{ij} = \beta(h\,A\,\sqrt{2}/\delta + w_{ij}) + 1 - \beta, $$

and

$$ w_{ij} = \beta(h\,B\,\sqrt{2}/\delta + w_{ij}) + 1 - \beta, $$

where $\beta = e^{-h}$ and $\delta = \min\{\Delta x, \Delta y\}$. The scheme in pseudocode is illustrated in details in Algorithm 1.

**Oblique light case**

Also for the equation (2.28) we need to compute the value of $w$ in the points which concide with the grid nodes. The general method is to use the control $\boldsymbol{a}$ to approximate the values of $w(x_{ij} + h\,\boldsymbol{a})$. The value of these controls is in the domain of minimization and can be limited to an appropriate choice in the set of unit vectors, as in our case. For example, we can discretize the ball $B_3(0, 1)$ and use the controls $\boldsymbol{a}$ in order to approximate the values $w$. As already done for the case of vertical light source (see equation (2.26)), also in this case it is possible to limit the computation only to a finite (lower) number of controls. Note that $h$ should not be chosen globally, and that for a suitable choice of $h$, $x_{ij} + h\,b(x_{ij}, \boldsymbol{a})$ can be entirely contained in the square (or in the rhombus in the case of a cartesian grid with different step in $x$ and $y$) having as vertices the next four neighbors of each grid point $x_{ij}$. For each point $x_{ij}$, the points $x_{ij} + h\,b(x_{ij}, \boldsymbol{a})$ are the locus of the circle with center $(-k\omega_1, -k\omega_2)$ and radius $I(x_{ij})k\,\rho$ where

$$ \rho = \sqrt{a_1^2 + a_2^2} $$

and $k = h/\omega_3$. The condition, required in order to contain this circumference in the rhombus formed by the four grid points close to $x_{ij}$, is satisfied for each radius less or equal to the radius of the circumference tangent to the nearest side. Without loss of

---

**Algorithm 1:** Search for the minimum with less controls - vertical light case

$x_{ij} \in G_{in}$
$P_1 = w_{i\,j+1} - w_{ij},\ P_2 = w_{i+1\,j} - w_{ij}$
$P_3 = w_{i\,j-1} - w_{ij},\ P_4 = w_{i-1\,j} - w_{ij}$
**if** $P_1 < P_3$ **then**
   $A \leftarrow P_1$
**else**
   $A \leftarrow P_3$
**end if**
**if** $P_2 < P_4$ **then**
   $B \leftarrow P_2$
**else**
   $B \leftarrow P_4$
**end if**
**if** $B < 0$ **and** $A < 0$ **then**
   $w_{ij} \leftarrow \beta(-h\,\sqrt{A^2 + B^2}\,\sqrt{2}/\delta + w_{ij}) + 1 - \beta$
**else**
   $w_{ij} \leftarrow \beta(h\,\min\{A, B\}\,\sqrt{2}/\delta + w_{ij}) + 1 - \beta$
**end if**

---

generality we can perform the computation in the first quarter, where for symmetry the center of the circumference is $(k\,|\omega_1|, k\,|\omega_2|)$ and the equation of the line containing the side of the square is

$$\frac{x}{\Delta x} + \frac{y}{\Delta y} - 1 = 0.$$

Thanks to these definition (see Fig. 2.3), the tangent condition with the circumference of maximum radius $I(x_{ij})k_M$ is

$$\frac{\Delta x \Delta y - k_M|\omega_1|\Delta y - k_M|\omega_2|\Delta x}{\sqrt{\Delta x^2 + \Delta y^2}} = I(x_{ij})k_M,$$

hence

$$k_M = \frac{\Delta x\,\Delta y}{\sqrt{\Delta y^2 + \Delta x^2}\,I(x_{ij}) + \Delta x\,|\omega_2| + \Delta y\,|\omega_1|}$$

and, as a consequence, the radius is

$$r_M = I(x_{ij})\frac{\Delta x\,\Delta y}{\sqrt{\Delta y^2 + \Delta x^2}\,I(x_{ij}) + \Delta x\,|\omega_2| + \Delta y\,|\omega_1|}.$$

The contraction condition (2.31) adds a lower limit to $r$. Defining $a_3 = \sqrt{1 - \rho^2}$ we can write

$$\rho_m = \sqrt{1 - \mu^2\omega_3^2/I(x_{ij})^2},$$
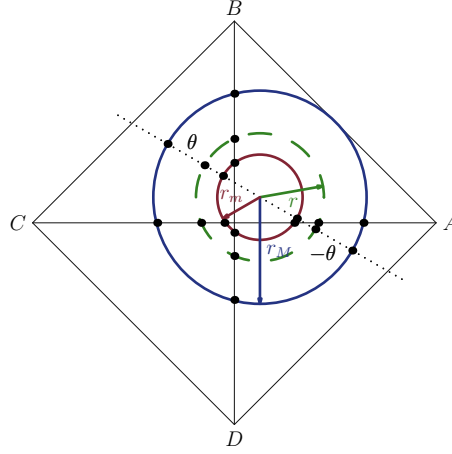
Figure 2.3.: Controls for oblique light case. For each point of a Cartesian grid, we can define the maximum value of $h$, such that $h\,b$ lies inside the rhombus whose vertices are the next four neighbors of the point. For each of the four triangles of the grid we compute $r_m$, $r_M$, $r$, $\theta$ and $\theta + \pi$ and the correspondents interpolated points. Between these we choose the minimum. The points where it can be located the minimum are marked with thicker points.

hence

$$r_m = \frac{\sqrt{I(x_{ij})^2 - \mu^2\omega_3^2}\,\Delta x\,\Delta y}{\sqrt{\Delta y^2 + \Delta x^2}\,I(x_{ij}) + \Delta x\,|\omega_2| + \Delta y\,|\omega_1|}$$

and $r \in [r_m, r_M]$. Note that $\mu\omega_3 > I(x_{ij})$ is a possible condition: in the contraction condition it means $\sqrt{1-\rho^2}$ less to a quantity greater than 1, that is always satisfied, hence $\mu\omega_3 > I(x_{ij})$ is equivalent to require $r_m = 0$.

Finally,

$$w(x_{ij}) = \min_{\substack{r \in [r_m, r_M] \\ \theta \in [0, 2\pi]}} \left\{ e^{-\mu h} P(x_{ij}, r, \theta) - \tau Q(x_{ij}, r, \theta) \right\} + \tau,$$

where $P(x_{ij}, r, \theta)$ is the intersection between the plane $z = -Ax - By + w_{ij}$ (defined in each quarter in a way similar to what has been done in §2.2.4 p. 43), and the cylinder of radius $r$ and center $(-k\omega_1, -k\omega_2)$, that is

$$P(x_{ij}, r, \theta) = -A\,(r\,\cos\theta - k\omega_1) + w_{ij} - B\,(r\,\sin\theta - k\omega_2) + w_{ij}.$$

From the condition

$$\frac{I(x_{ij})}{\omega_3}a_3 = \frac{I(x_{ij})}{\omega_3}\sqrt{1-\rho^2},$$

we get

$$Q(x_{ij}, r, \theta) = \frac{1}{\omega_3 k_M}\sqrt{(k_M\,I(x_{ij}))^2 - r^2}(1 - \mu\,w_{ij}).$$

Then, we can define

$$F(x_{ij}, r, \theta) = \beta\, P(x_{ij}, r, \theta) - \tau\, Q(x_{ij}, r, \theta)$$

whose stationary points are

$$\theta_1 = \arctan(B/A), \quad \theta_2 = \arctan(B/A) + \pi$$

and

$$r = \frac{\beta\sqrt{A^2 + B^2}k_M\, I(x_{ij})}{\sqrt{\omega_3^2\, k_M^2 \beta^2 (A^2 + B^2) + 4\tau^2}}\omega_3\, k_M.$$

Obviously, the minimum should be sought even to extremes, i.e. also for $r = r_m$ and $r = r_M$, and these values should be compared with the values with which the three cylinders intersect the diagonals of the rhombus formed by the point $x_{ij}$ and its neighbors. In conclusion: for each of the four triangles $T_i$ we compute $r_m$, $r_M$, $r$, $\theta_1$ and $\theta_2$. We obtain the interpolated points corresponding and finally we choose the minimum. It should be noted that in this way, at the cost of limited to only linear interpolation, we have reduced the number of necessary controls to a maximum of 36 (limit value not actually always achieved, taking into account that some controls are not to be computed in case you are out of the quarter). If $I(x_{ij}) = 0$, the minimum coincides with the value interpolated $w$ in the point $(-k\omega_1, -k\omega_2) = (b_1, b_2)$. A description of the method in pseudocode can be found in Algorithm 2.

### 2.2.5. Convergence results

The SL schemes are based on the idea which is to discretize directly the "directional derivative" which is hidden behind the nonlinearity $\nabla u(\mathbf{x})$. The goal is to mimic the method of characteristics by constructing the solution at each grid point integrating back along the characteristics passing through it and reconstructing the value at the root of the characteristic line by interpolation. In this way the schemes have always an upwind correction which is crucial for stability allowing for larger time steps with respect to finite differences (FD). In fact, FD schemes first discretize in space then discretize in time using a fixed stencil. Instead SL schemes first discretize in time then discretize in space, moving the stencil. We begin by writing the equation

$$u_t + c(\mathbf{x})|\nabla u(\mathbf{x})| \tag{2.38}$$

as

$$u_t + \max_{a \in B(0,1)} [-c(\mathbf{x})a \cdot \nabla u(\mathbf{x})] = 0. \tag{2.39}$$

Naturally, this is equivalent to our equation since plugging $a^* = \frac{-\nabla u(\mathbf{x})}{|\nabla u(\mathbf{x})|}$ we get back to the first equation. In order to simplify notations we will consider $N = 2$. We define the

---

**Algorithm 2:** Search for the minimum with less controls - oblique light case

---

$x_{ij} \in G_{in}$, $I(x_{ij}) > 0$

$f_\gamma \leftarrow \infty$

computation of $r_M$, $r_m$, $h$

$b_1 \leftarrow -h\omega_1/\omega_3$

$b_2 \leftarrow -h\omega_2/\omega_3$

**for all** quarters $Q$ **do**

    computation of the coefficients $A$ and $B$ of the plane

    computation of $r$, $\theta$

    **if** $r \in [r_m, r_M]$ **then**

        **if** $(\pm r \cos\theta, \pm r \sin\theta) \in Q$ **then**

            $f_\gamma \leftarrow \min\{f_\gamma, F(x_{i,j}, \pm r, \theta)\}$

        **end if**

        **if** $|b_1| < r$ **or** $|b_2| < r$ **then**

            computation of $q_r$ intersections with diagonal of rhombus, $f_\gamma \leftarrow \min(f_\gamma, q))$

        **end if**

    **end if**

    **if** $(\pm r_m \cos\theta, \pm r_m \sin\theta) \in Q$ **then**

        $f_\gamma \leftarrow \min(f_\gamma, F(x_{ij}, \pm r_m, \theta))$

    **end if**

    **if** $|b_1| < r_m$ **or** $|b_2| < r_m$ **then**

        computation of $q_{rm}$ intersections with diagonal of rhombus, $f_\gamma \leftarrow \min(f_\gamma, q_{rm}))$

    **end if**

    **if** $(\pm r_M \cos\theta, \pm r_M \sin\theta) \in Q$ **then**

        $f_\gamma \leftarrow \min(f_\gamma, F(x_{ij}, \pm r_M, \theta))$

    **end if**

    **if** $|b_1| < r_M$ **or** $|b_2| < r_M$ **then**

        computation of $q_{rM}$ intersections with diagonal of rhombus, $f_\gamma \leftarrow \min(f_\gamma, q_{rM}))$

    **end if**

**end for**

$w(x_{ij}) \leftarrow f_\gamma + \tau$

---

lattice $L(x, y, z)$ by

$$L = \{(x_i, y_j, t_n) : x_i = i\Delta x, y_j = j\Delta y, t_n = n\Delta t\} \tag{2.40}$$

where $i, j, \in \mathbb{Z}$ and $n \in \mathbb{N}$, $(x_i, y_j, t_n) \in \mathbb{R}^2 \times \mathbb{R}^+$, $x$ and $y$ are the space steps (we assume they are constant) and $t$ is the time step.

In order to obtain the SL-scheme let us consider the following approximation

$$- \nabla v(x_i, y_j, t_n) \cdot a = \frac{v(x_i - a_1\Delta t, y_j - a_2\Delta t, t_n) - v(x_i, y_j, t_n)}{\Delta t} + O(t). \tag{2.41}$$

We will use the standard notation $v^n(i, j)$ for an approximation of $v(x_i, y_j, t_n)$, with $i, j, \in \mathbb{Z}$, $n \in \mathbb{N}$ and $v^n : \mathbb{R}^2 \to \mathbb{R}$ for its reconstruction, i.e. its extension to any triple $(x, y, t_n)$ by interpolation of the values on the grid. Replacing the term $v_t$ by forward finite differences and the discrete directional derivative by the SL discretization, we get

$$\frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta t} = \min_{a \in B(0,1)} \left[ \frac{v^n(x_i - a_1\Delta t c_{ij}, y_j - a_2\Delta t c_{ij}) - v_{ij}^n}{\Delta t} \right]. \tag{2.42}$$

For a general Hamiltonian $H(\nabla u)$ we have

$$\frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta t} = \min_{a \in \mathbb{R}^2} \left[ \frac{v^n(x_i - a_1\Delta t c_{ij}, y_j - a_2\Delta t c_{ij}) - v_{ij}^n}{\Delta t} + H^*(a) \right], \tag{2.43}$$

where $H^*(a)$ is the Legendre transform defined by

$$H^*(a) = \sup_{q \in \mathbb{R}^n} \{a \cdot q - H(q)\}. \tag{2.44}$$

Finally, we can write the time explicit scheme

$$v_{i,j}^{n+1} = v_{i,j}^n + \min_{a \in \mathbb{R}^2} \left[ v^n(x_i - a_1\Delta t, y_j - a_2\Delta t) + \Delta t H^*(a) \right]. \tag{2.45}$$

By construction, the numeric dependence domain contains the continuous dependence domain without any additional conditions on $t$ and $x$. This allows for larger time steps than FD where a CFL condition of the type $\frac{\Delta t}{\Delta x} C \leq 1$ has to be imposed to guarantee stability.

In order to compute the solution we first compute the value of $v$ on the right-hand side by an interpolation procedure based on the values of the nodes of the lattice $L$. Then, one has to determine $H^*(a)$ so that we can finally compute the minimum for $a \in \mathbb{R}^2$.

A major difficulty when applying SL schemes is to compute $H^*$. Sometimes it is possible to determine its explicit expression (as e.g. for the front propagation model). In general one has to rely on its approximation by the Fast Legendre Transform (FLT) developed by Brenier [12] and Corrias [25]. This solution is feasible as far as the state space has 2 or 3 dimensions. Now, we can give a convergence theorem in $\mathbb{R}$.

Consider the SL-scheme

$$\begin{cases} v_j^{n+1} = \min_{a \in \mathbb{R}} \left[ I_r[V^n](x_j - a\Delta t) + \Delta t H^*(a) \right], \\ v_j^0 = u_0(x_j), \end{cases} \tag{2.46}$$

where $I[\cdot]$ is a generic (high order) interpolation operator on the grid with a reconstruction of degree $r > 1$.

**Theorem 2.2.2 (Ferretti, 2003)**
*Let $V^n$ be defined by the scheme* (2.46). *Assume that*

*A1. $H^{*''}(a) \leq \frac{1}{m_H}$*

*A2. $|I_r[V](x) - I_1[V](x)| \leq C_r \max\limits_{x_{j-1},x_j,x_{j+1} \in \mathbb{S}} |v_{j+1} - 2v_j + v_{j-1}|, \quad (C_r < 1),$ where $\mathbb{S}$ is
the stencil of the reconstruction defined as $\mathbb{S}(x) = (x - h_-\Delta x, x + h_+\Delta x)$.*

*Then, for any $j \in \mathbb{Z}$ and $n \geq 1$,*

$$v_{j+1}^n - 2v_j^n + v_{j-1}^n \leq \frac{\Delta x^2}{m_H \Delta t}. \tag{2.47}$$

*Moreover, since A2 holds, then*

$$\max_{x_{j-1},x_j,x_{j+1} \in \overline{U}(x_j) + \overline{a_j}\Delta t} |v_{j+1}^n - 2v_j^n + v_{j-1}^n| \leq C \frac{\Delta x^2}{\Delta t} \tag{2.48}$$

*for some positive constant $C$ depending on $C_r$, $h$ and $m_H$ and with $\overline{U}(x) = (x - h\Delta x, x + h\Delta x)$, with a fixed $h > \max(h_-, h_+)$ so that $\mathbb{S}(x) \subset \overline{U}(x)$.*

Hence, the scheme converges in $L^\infty \times (0, T)$ to the unique viscosity solution. See [43] for the proof and more details.

Another important convergence theorem that we want to recall here is the Barles-Souganidis theorem [9]. This convergence theory gives a more abstract and general framework for convergence of schemes, including the possibility of treating second-order, degenerate and singular equations. Roughly speaking, this theory states that any monotone, stable and consistent scheme converges to the exact solution provided there exists a comparison principle for the limiting equation.
They consider approximation schemes of the form

$$S(\rho, x, u^\rho(x), u^\rho) = 0, \qquad \text{in} \quad \overline{\Omega}, \tag{2.49}$$

where $\Omega$ is an open subset of $\mathbb{R}^N$, $\overline{\Omega}$ its closure. Here $S : \mathbb{R}^+ \times \overline{\Omega} \times \mathbb{R} \times B(\overline{\Omega}) \to \mathbb{R}$ is locally bounded.

**Theorem 2.2.3 (Barles-Souganidis, 1991)**
*Let $u(x)$ be the unique viscosity solution in $\mathbb{R}^N$ of the equation*

$$H(x, u, Du, D^2u) = 0. \tag{2.50}$$

*Assume that the scheme $S$ defined in (2.49) verifies the following assumptions:*

**A1.** *The scheme is monotone, i.e.*

$$S(\rho, x, t, u) \leq S(\rho, x, t, v) \tag{2.51}$$

*if $u \geq v$ for all $\rho \geq 0$, $x \in \overline{\Omega}$, $t \in \mathbb{R}$, and $u, v \in B(\overline{\Omega})$.*

**A2.** *The scheme is stable, i.e. for all $\rho > 0$, there exists a solution $u^{\rho} \in B(\overline{\Omega})$ of (2.49), with a bound independent of $\rho$.*

**A3.** *The scheme is consistent, i.e. for all $x \in \overline{\Omega}$ and $\varphi \in C_b^{\infty}(\overline{\Omega})$*

$$\liminf_{\rho \to 0, \, y \to x, \xi \to 0} \frac{S\left(\rho, y, \varphi\left(y\right) + \xi, \varphi + \xi\right)}{\rho} \geq H_*\left(x, \varphi\left(x\right), D\varphi\left(x\right), D^2\varphi\left(x\right)\right) \tag{2.52a}$$

*and*

$$\limsup_{\rho \to 0, \, y \to x, \xi \to 0} \frac{S\left(\rho, y, \varphi\left(y\right) + \xi, \varphi + \xi\right)}{\rho} \leq H^*\left(x, \varphi\left(x\right), D\varphi\left(x\right), D^2\varphi\left(x\right)\right), \tag{2.52b}$$

*where $H_*$ and $H^*$ denote viscosity sub- and supersolution of (2.50) respectively.*

*Finally, assume that (2.50) has the following strong uniqueness property:*

**A4.** *If $u \in B(\overline{\Omega})$ is an upper semicontinuous solution of (2.50) and $v \in B(\overline{\Omega})$ is a lower semicontinuous solution of (2.50), then $u \leq v$ in $\overline{\Omega}$.*

*Then, as $\rho \to 0$, the solution $u^{\rho}$ of (2.49) converges locally uniformly to the unique continuous viscosity solution of (2.50).*

There exists also an extended version of this theorem for the time-marching form. For more details on convergence theory, on the proofs of the theorems state here and for more details on SL schemes in general, see the book [39].

## 2.3. A new general convergence theorem

Now, let us state a general convergence theorem suitable for a general operator. We will see in the next chapters that the discrete operators of the ON–model and the PH–model described in Chapter 1 satisfy the properties listed here.

Let $W_i = w(x_i)$ so that $W$ will be the vector solution giving the approximation of the height $u$ at every node $x_i$ of the grid. Note that in one dimension the index $i$ is an integer number, in two dimensions $i$ denotes a multi-index, $i = (i_1, i_2)$. We consider a semi-Lagrangian scheme written in a fixed point form, so we will write the fully discrete scheme as

$$W_i = T_i(W). \tag{2.53}$$

Denoting by $G$ the global number of nodes in the grid, the operator corresponding to the oblique light source is $T : \mathbb{R}^G \to \mathbb{R}^G$ that is defined componentwise by

$$T_i(W) = \min_{a \in \partial B_3} \{ e^{-\mu h} w(x_i + hb(x_i, a)) - \tau P(x_i, z) a_3 (1 - \mu w(x_i)) \} + \tau, \qquad (2.54)$$

where

$$\tau = 1 - e^{-\mu h}/\mu \qquad (2.55)$$
$$P : \Omega \times \mathbb{R} \to \mathbb{R} \qquad (2.56)$$

and $w(x_i + hb(x_i, a))$ is obtained by interpolating on $W$.

Note that here $\mu$ is a free positive parameter without a specific physical meaning, but it is important because varying its value it is possible to modify the slope. In fact, the slope increases for increasing values of $\mu$.

Comparing (2.54) with its analogue for the vertical light case we can immediately note that the former has the additional term $\tau P(x_i, z) a_3 (1 - \mu w(x_i))$ which requires analysis.

**Theorem 2.3.1** *Let $T_i(W)$ the operator defined as in (2.54). Then $T_i(W)$ has the following properties:*

1. *Let*

$$P(x_i, z)\overline{a}_3 \leq 1, \qquad (2.57)$$

   *where* $\quad \overline{a}_3 \equiv arg \min_{a \in \partial B_3} \{ e^{-\mu h} w(x_i + hb(x_i, a)) - \tau P(x_i, z) a_3 (1 - \mu w(x_i)) \}$.

   *Then $0 \leq W \leq \frac{1}{\mu}$ implies $0 \leq T(W) \leq \frac{1}{\mu}$.*

2. *$T$ is a monotone operator, i.e., $v \leq u$ implies $T(v) \leq T(u)$.*

3. *$T$ is a contraction mapping in $[0, 1/\mu)^G$ if $P(x_i, z)\,\overline{a}_3 < \mu$.*

*Proof.*

1. To prove that $W \leq \frac{1}{\mu}$ implies $T(W) \leq \frac{1}{\mu}$ we just note that

$$T(W) \leq \frac{e^{-\mu h}}{\mu} + \tau = \frac{1}{\mu}. \qquad (2.58)$$

   Let $W \geq 0$; then

$$T(W) \geq -\tau P(x_i, z)\,\overline{a}_3 (1 - \mu w(x_i)) + \tau = \tau \left( 1 - P(x_i, z)\,\overline{a}_3 (1 - \mu w(x_i)) \right). \quad (2.59)$$

   This implies that $T(W) \geq 0$ if $P(x_i, z)\,\overline{a}_3 \leq 1$ since $0 \leq 1 - \mu w(x_i) \leq 1$.

2. In order to prove that $T$ is monotone, assume that $v \leq u$ and let $a^*$ and $\overline{a}$ be the arguments of the minima. Then

$$
\begin{aligned}
T(v) - T(u) &\leq e^{-\mu h}[v(x + hb(x, a^*)) - u(x + hb(x, \overline{a}))] \\
&\quad - \tau P(x, z)(a_3^*(1 - \mu v(x)) - \overline{a}_3(1 - \mu u(x))) \\
&\leq e^{-\mu h}[v(x + hb(x, \overline{a})) - u(x + hb(x, \overline{a}))] \\
&\quad + \tau P(x, z)\overline{a}_3(v(x) - u(x)).
\end{aligned}
\tag{2.60}
$$

Hence,

$$
\begin{aligned}
T(v) - T(u) &\leq e^{-\mu h}[v(x + hb(x, \overline{a})) - u(x + hb(x, \overline{a}))] \\
&\quad + \tau P(x, z)\overline{a}_3(v(x) - u(x)) \leq 0.
\end{aligned}
\tag{2.61}
$$

Note that this property does not require condition (2.57) to be satisfied.

3. Let

$$
P(x_i, z)\overline{a}_3 < \mu.
\tag{2.62}
$$

To prove that $T$ is a contraction mapping note that (2.61) yields

$$
T(v) - T(u) \leq \left(e^{-\mu h} + \tau P(x, z)\overline{a}_3\right) ||v - u||_\infty.
\tag{2.63}
$$

Then, $T$ is a contraction mapping if and only if

$$
e^{-\mu h} + \tau P(x, z)\overline{a}_3 < 1
\tag{2.64}
$$

and this holds true if the bound (2.62) is satisfied.

$\square$

The algorithm is rather simple since it is based on the fixed-point iteration

$$
\begin{cases}
W^n &= T(W^{n-1}), \\
W^0 &\text{ given.}
\end{cases}
$$

However, to improve the speed of convergence, the iteration must be accelerated (see [36] for an accelerator method based on the monotonicity of the $T$ operator, as the FM method that will be described in the next chapter). On the boundary of $\Omega$ we just impose the homogeneous Dirichlet boundary condition. This condition implies that the shadows must not cross the boundary of $\Omega$, so the choice $\omega_3 = 0$ corresponding to an infinite shadow behind the surface is not admissible. Note that the interfaces between light and black shadows are ignored, since the algorithm computes with the same rule on every internal node.

# 3. Numerical acceleration techniques

Since in the numerical tests we will also compare results obtained with Fast Marching and Fast Sweeping methods, in this chapter we will briefly sketch how they works and their properties.

## 3.1. Fast Marching Method

All the literature on this subject mainly stems from a paper by Tsitsiklis [116] in which the author introduced a Dijkstra-like algorithm to solve efficiently the discretized Hamilton-Jacobi equation associated to a particular minimum time problem. One year later, Sethian [100] refined Tsitsiklis' arguments introducing the so-called Fast Marching (FM) method for the eikonal equation modelling the monotone front propagation. Later on, some authors improved the FM method in order to make it more efficient. For example, Kim [63] introduced the Group Marching (GM) method and Chopp [22] extended the FM method to high order approximation. Sethian himself proposed a wide class of applications for FM method (see, among others, [66, 67]). More recently, some authors tried to modify the FM method in order to deal with more general equations. Sethian and Vladimirsky [103] introduced an extension which covers a wider class of equations including the monotone anisotropic front propagation and Vladimirsky [117] introduced an extension which covers the front evolution with time-depending velocity. Prados and Soatto [93] suggested a modification which allows to deal with quite general Hamilton-Jacobi equations provided a subsolution of the equation is given. In parallel, other authors competed with FM method introducing the so-called Fast Sweeping (FS) method which provides a very efficient technique in order to speed up the classical iterative algorithm. We will focus our attention on the FS method in the next section. Now, let us explain how FM works.

The Fast Marching [102] is a local method based on a finite difference approximation, which produces a very fast algorithm for the numerical solution of eikonal equations also on reconstruction domains that are not connected. We suppose that the set $\Omega$ is entirely contained in the rectangular domain corresponding to the image, and that on this image is defined a discrete grid $G$ (size $n \times m$) of points $\{x_{ij}\}$ with $i, j \in \mathbb{Z}$. The basis of the method is the division of the grid into three separate sets at every iteration $n$:
the set of Accepted nodes ACCEPTED$(n)$, whose value has been already computed and accepted;
the set of Considered nodes CONSIDERED$(n)$, or Narrow Band NARROW BAND$(n)$, for which the value has to be computed at the present iteration;
the set of Far nodes FAR$(n)$, on which has not been made any computation and the nodes will be computed in future iterations.

This method was first introduced for solving problems of evolution of the fronts, therefore, the initial value known is usually a given value and coincides with the initial front. In our case, ACCEPTED*(0)* at the first iteration is the set of nodes where we have to apply boundary conditions (which are known). Then, at iteration $n$, the set CONSIDERED*(n)* contains the neighboring nodes of ACCEPTED*(n)*, and FAR*(n)* are the remaining nodes where we do nothing at that iteration. The algorithm computes the value in CONSIDERED*(n)*. The node $x_{ij}$ where the minimum is achieved is marked ACCEPTED (i.e. ACCEPTED*(n+1)*=ACCEPTED*(n)*$\cup\{x_{ij}\}$); the set CONSIDERED*(n)* is updated adding the neighboring nodes to $x_{ij}$ and we compute the solution in CONSIDERED*(n+1)*. The algorithms accepts only one node for each iteration and ends only when the FAR region is empty. The method converges in a finite number of iteration but unfortunately its application is limited to eikonal type equations. Start to see more in details how it works.

In the SfS problem the image data are defined on a discrete grid. It is necessary, therefore, to find a suitable convention for the definition of approximations for the spatial derivatives. The central finite difference approximation is not stable for the equations of hyperbolic type, so the choice falls on finite difference forward or backward. However, the only forward or backward differences may introduce instability when the direction of the derivative is not followed. The right solution for this problem is to use an upwind-type discretization.

### 3.1.1. Approximation of the eikonal equation

We consider an equation of the type

$$|\nabla u(\mathbf{x})| = f(\mathbf{x}), \qquad \mathbf{x} \in \Omega \tag{3.1}$$

and we denote by $G_{in}$ the set of grid points belonging to $\Omega$, by $G_{out}$ the set $G\backslash G_{in}$ and by $G_b \subset G_{out}$ the set of grid points belonging to $\partial\Omega$. We define neighboring nodes of node $w_{i,j}$ the nodes

$$\text{NEAR}(w_{i,j}) = \{w_{i+1,j}, w_{i-1,j}, w_{i,j+1}, w_{i,j-1}\}.$$

For simplicity, we also denote by $\delta$ the grid step (in the case of Cartesian grid with various steps $\delta = \sqrt{\Delta x^2 + \Delta y^2}/\sqrt{2}$).

The discrete version of (3.1) is

$$[(w_{i,j} - w_{xm})^+]^2 + [(w_{i,j} - w_{ym})^+]^2 = f_{i,j}^2\,\delta^2 \tag{3.2}$$

where $w_{xm} = \min\{w_{i-1,j}, w_{i+1,j}\}$, $w_{ym} = \min\{w_{i,j-1}, w_{i,j+1}\}$ and

$$(x)^+ = \begin{cases} x, & x > 0, \\ 0, & x \leq 0. \end{cases}$$

At the extremes of the domain of definition of the grid can be used appropriate finite

difference, for example, for $i = 0$ is possible to use

$$[(w_{0,j} - w_{1,j})^+]^2 + [(w_{0,j} - w_{ym})^+]^2 = f_{0,j}^2 \, \delta^2. \tag{3.3}$$

In the initialization phase the exact (or interpolated) values of $u(\mathbf{x}) = 0$ or $u(\mathbf{x}) = g(\mathbf{x})$ are assigned to the grid points corresponding to the boundary condition, and these nodes are marked as Accepted. The next closest nodes are marked as Considered, or as Accepted if they belong to $G_{out}$. All other nodes are marked as Far and a very high value is assigned to them.

As already mentioned, the method is used initializing the grid nodes. An initial front of nodes is defined, whose value has been definitively accepted. Subsequently, the neighboring nodes are computed again and compared with the value previously assumed. The algorithm proceeds locally to search for the minimum, hence assigning to each node a very high initial value simplifies the comparisons in subsequent computations. In detail, the equations (3.2) and (3.3) can be rewritten more compactly as

$$[(x - a)^+]^2 + [(x - b)^+]^2 = f^2 \delta^2$$

where $a = w_{xm}$ and $b = w_{ym}$. This equation has a unique solution $\bar{x}$ that is

$$\bar{x} = \begin{cases} \min\{a, b\} + f\delta, & |a - b| \geq f\delta, \\ \frac{1}{2}\left(a + b + \sqrt{2f^2\delta^2 - (a - b)^2}\right), & |a - b| < f\delta. \end{cases} \tag{3.4}$$

The method assigns a value of $g(\mathbf{x})$ to the nodes belonging to $G_b$ and the conventional value 0 to the other nodes belonging to $G_{out}$. To all the other grid nodes is assigned the conventional value of $\infty$, that is the highest positive possible value. The method is described in detail in Algorithm 3 and in Fig. 3.1.

A scheme of the procedure is summarized below:

- Initialization. The grid points are marked as Far, Accepted e Considered.
  - The nodes that belong to $G_{out}$ are marked as Accepted and we assign them the value $w_{ij} = 0$; to the nodes of $G_b \subset G_{out}$ that constitute the border of the reconstruction domain is given the value of $w_{ij} = g(x_{ij})$ and are marked as Accepted. The next closest of these nodes marked as Accepted, belonging to $G_{in}$, are marked as Considered and we assign them a temporary value computed with the equation (3.4).
  - For the remaining nodes, marked as Far, is assigned a value of $w_{ij} = \infty$.

- Core of the algorithm.
  As long as there are points marked as Considered:
  - Among all the nodes marked as Considered is chosen the node with the minimum value of $w$, i.e. $w_{hk}$, and it is marked as Accepted.

– Among all the nodes next neighboring of $(h, k)$ the nodes FAR are marked as CONSIDERED and computed again according to equation (3.4).
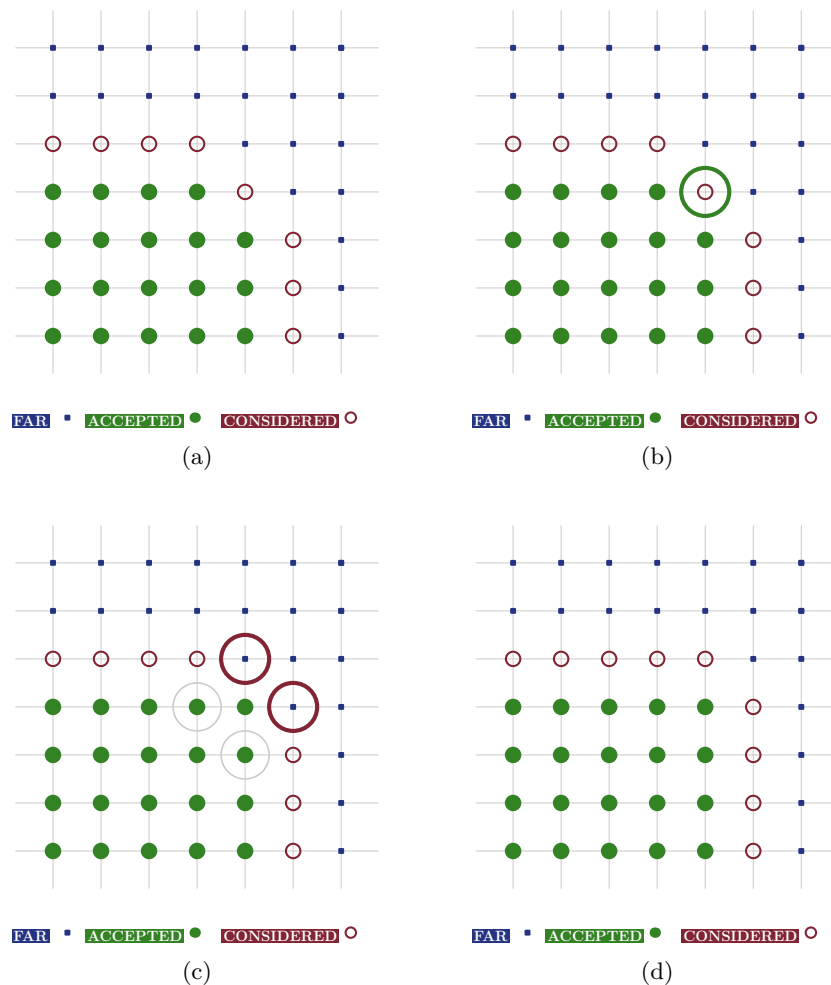


Figure 3.1.: Fast Marching Method: (a) Initialization. The grid points are marked as FAR, ACCEPTED and CONSIDERED. (b) The minimum of the CONSIDERED is marked as ACCEPTED. (c) We are looking for the neighbors of the minimum. The neighbors that are FAR are marked as CONSIDERED. All the neighbors that are marked as CONSIDERED are computed again according to the numerical scheme in (3.4). (d) The resulting new grid. The method continues until exhaustion of the list CONSIDERED.

**Remark 3.1** *The algorithm computes the value in each node only when one of its neighbors becomes* ACCEPTED. *A node can become* ACCEPTED *only once, after which it is no longer taken into account. In a Cartesian grid each node has four neighbors, then each*

*node can be computed to a maximum of four times. It appears that the complexity of the algorithm is $O(N)$, where $N$ is the number of nodes of the grid belonging to $G_{in}$. This value must be added to the cost of the search for the minimum in the list CONSIDERED. Using an algorithm of type Heapsort to sort the list CONSIDERED, such that the minimum is at the top of the list and the complete ordering is performed during the insertion phase, we can estimate this cost in $O(\log(N))$, bringing the total value of the computational cost to $O(N \log(N))$.*

---

**Algorithm 3:** Fast Marching Method

---

    Initialization
    **for all** $x_{ij} \in G$ **do**
      **if** $x_{ij} \in G_{out}$ **then**
        $x_{ij} \rightarrow$ ACCEPTED
        $W_{ij} \leftarrow 0$
        **if** $x_{ij} \in G_b$ **then**
          $W_{ij} \leftarrow g(x_{ij})$
        **end if**
      **else**
        $x_{ij} \rightarrow$ FAR
        $W_{ij} \leftarrow \infty$
      **end if**
    **end for**
    **for all** $x_{ij} \in$ ACCEPTED **do**
      **for all** $x_{hk} \in$ NEAR$(x_{ij})$ **do**
        **if** $x_{hk} \in$ FAR **then**
          $x_{hk} \rightarrow$ CONSIDERED
          $W_{hk} \leftarrow \min\{W_{hk}, \text{upwind}(x_{hk})\}$
        **end if**
      **end for**
    **end for**

    Execution
    **while** CONSIDERED **do**
      $x_{ij} = \text{argmin }\{W(x_{lm})|x_{lm} \in$ CONSIDERED$\}$
      $x_{ij} \rightarrow$ ACCEPTED
      **for all** $x_{hk} \in NEAR(x_{ij})$ and $x_{hk} \notin$ ACCEPTED **do**
        $W_{hk} \leftarrow \min\{W_{hk}, \text{upwind}(x_{hk})\}$
        **if** $x_{hk} \in$ FAR **then**
          $x_{hk} \rightarrow$ CONSIDERED
        **end if**
      **end for**
    **end while**

---

## 3.2. Fast Sweeping Method

There are mainly two types of approaches for solving the Eikonal equation. One approach is to transform it to a time dependent problem. For example, if we have $u(\mathbf{x}) = 0$, $\mathbf{x} \in \Gamma \subset \partial\Omega$, then $u(\mathbf{x})$ is the first arrival time at $\mathbf{x}$ for a wave front starting at $\Gamma$ with a normal velocity that is equal to $\frac{1}{f(\mathbf{x})}$. This can be solved by the level set method. In the control framework, a semi-Lagrangian scheme is obtained for Hamilton-Jacobi equations by discretizing in time the dynamic programming principle [37, 38]. However, many time steps may be needed for the convergence of the solution in the entire domain due to finite speed of propagation and CFL condition for time stability. The other approach is to treat the problem as a stationary boundary value problem and to design an efficient numerical algorithm to solve the system of nonlinear equations after discretization. For example, the Fast Marching method [100, 116] is of this type. In the FM method, the update of the solution follows the causality in a sequential way; i.e., the solution is updated one grid point by one grid point in the order that the solution is strictly increasing (decreasing). Hence an upwind difference scheme and a heapsort algorithm are needed. The complexity is of order $O(N \log N)$ for $N$ grid points, where the $\log N$ factor comes from the heapsort algorithm.

Here we present and analyze an iterative algorithm, called the Fast Sweeping (FS) method, for computing the numerical solution for the Eikonal equation on a rectangular grid in any number of space dimensions. The Fast Sweeping method is motivated by the work in [11] and was first used in [127] for computing the distance function. The main idea of this method is to use nonlinear upwind difference and Gauss-Seidel iterations with alternating sweeping ordering. In contrast to the Fast Marching method, the FS method follows the causality along characteristics in a parallel way; i.e., all characteristics are divided into a finite number of groups according to their directions and each Gauss-Seidel iteration with a specific sweeping ordering covers a group of characteristics simultaneously. The Fast Sweeping method is extremely simple to implement and for this reason is a popular method for solving HJEs [32, 114, 94]. The algorithm is optimal in the sense that a finite number of iterations is needed. So the complexity of the algorithm is $O(N)$ for a total of $N$ grid points. The number of iterations is independent of grid size. The accuracy is the same as any other method which solves the same system of discretized equations. The FS method has been extended to more general Hamilton-Jacobi equations [62, 114]. The idea of alternating sweeping ordering was also used in Danielsson's algorithm [32]. The algorithm computes the distance mapping, i.e., the relative $(x, y)$ coordinate of a grid point to its closest point using an iterative procedure. Danielsson's algorithm is based on a strict dimension by dimension discrete formulation which in general does not follow the real characteristics of the distance function in two and higher dimensions and hence results in low accuracy and twice as many iterations compared to the Fast Sweeping method we present here. Danielsson's algorithm does not work for distance functions to more general data sets such as the distance to a curve or a surface. Neither does it extend to general Eikonal equations. Later, another discrete approach that uses the idea of FS method was proposed in [115]. It can compute the distance function more accurately but does not apply to general Eikonal equations either.

Other related methods include a dynamic programming approach and post sweeping idea in [99] and a group marching method in [63].

The Fast Sweeping (FS) is another popular method for solving HJEs [32, 114, 94]. The main advantage of this method is its implementation, which is extremely easy (easier than that of Fast Marching). FS method is basically the classical iterative (fixed-point) method, since each node is visited in a predefined order, until convergence is reached. Here, the visiting directions (sweeps) are alternated in order to follow all possible characteristic directions, trying to exploit causality. In two-dimensional problems, the grid is visited sweeping in four directions: $S \to N$ & $W \to E$, $S \to N$ & $E \to W$, $N \to S$ & $E \to W$ and $N \to S$ & $W \to E$.

The basic idea of the Fast Sweeping method [126, 125] is to use an upwind discretization and iterations of the Gauss-Seidel type with an alternating order (sweep). Unlike the Fast Marching method, FS follows the characteristics by computing along alternate directions, which favor specific groups of characteristics. All the characteristics are divided into a finite number of groups based on the direction and each iteration of Gauss-Seidel runs along a specific group, performed with a specific order. It is well known that in the case of eikonal equations FS converges in only four sweeps.

The method is appropriate to obtain the viscosity solutions of the equation (3.1). Also in this case we suppose that the reconstruction domain $\Omega$ is entirely contained in the rectangular image, and that on this image is defined a discrete grid $G$ (size $n \times m$) of points $\{x_{ij}\}$ with $i, j \in \mathbb{Z}$. We use the same sets $G_{in}$, $G_{out}$ and $G_b$ defined before. We use again the discrete version of the equation (3.1), that is the (3.2).

In the initialization phase the exact (or interpolated) values of $u(\mathbf{x})$ are assigned to the grid points corresponding to the boundary condition ($u(\mathbf{x}) = 0$ or $u(\mathbf{x}) = g(\mathbf{x})$). At all other points of the grid is assigned a very high value. Subsequently, the values of $w_{ij}$ are computed again using the equation (3.2), by alternating the order of computation (sweep). The computation is repeated until the convergence condition is satisfied.

### 3.2.1. Approximation of the eikonal equation

As for the FM method, the equation (3.2) can be written in a more compact form as follow:

$$[(x - a)^+]^2 + [(x - b)^+]^2 = f^2 \delta^2$$

where $a = w_{xm}$ and $b = w_{ym}$, whose unique solution $\bar{x}$ is given by equation (3.4). The method assigns a value of $g(\mathbf{x})$ to the nodes belonging to $G_b$ and the conventional value 0 to the other nodes belonging to $G_{out}$. To all the other grid nodes is assigned the conventional value of $\infty$, that is the highest positive possible value. As mentioned before, at each sweep the new value obtained thanks to the equation (3.4) is compared with the previous value, and accepted only if it is smaller than the value previously computed. Then, assigning a positive value higher than any result of equation (3.4) allows us to accept the value computed in the next sweep. The order of computation of the grid points occurs according to a different sweep alternatively, iteration after iteration (assuming to be at iteration number $d$, the order is given by the method described in Algorithm 4).

---

**Algorithm 4:** Sweeping Order

$n \leftarrow d \pmod 4$

**if** $n == 0$ **then**

$\quad (i_0, i_1) \leftarrow (0, i_M), (j_0, j_1) \leftarrow (0, j_M)$

**end if**

**if** $n == 1$ **then**

$\quad (i_0, i_1) \leftarrow (i_M, 0), (j_0, j_1) \leftarrow (0, j_M)$

**end if**

**if** $n == 2$ **then**

$\quad (i_0, i_1) \leftarrow (i_M, 0), (j_0, j_1) \leftarrow (j_M, 0)$

**end if**

**if** $n == 3$ **then**

$\quad (i_0, i_1) \leftarrow (0, i_M), (j_0, j_1) \leftarrow (j_M, 0)$

**end if**

---

Let us suppose for simplicity that the grid is defined with values of $i$ and $j$ between 0 and $i_M$ and between 0 and $j_M$. Then the four sweeps that we have to use are

1. $i = 0 \ldots i_M$, $j = 0 \ldots j_M$,

2. $i = i_M \ldots 0$, $j = 0 \ldots j_M$,

3. $i = i_M \ldots 0$, $j = j_M \ldots 0$,

4. $i = 0 \ldots i_M$, $j = j_M \ldots 0$.

A complete description of the method is described in pseudocode in Algorithm 5.

### 3.2.2. Motivation

In the Fast Sweeping algorithm the upwind difference scheme used in the discretization enforces the causality; i.e., the solution at a grid point is determined by its neighboring values that are smaller. The one sided difference scheme at the boundary enforces the propagation of information to be from inside to outside since the data set $\Gamma$ is contained in the computational domain. If all grid points can be ordered according to the causality along characteristics, one iteration of the Gauss-Seidel iteration is enough for convergence. For example, the heapsort algorithm is used in the Fast Marching method to sort out this order every time a grid point is updated. The key point behind Gauss-Seidel iterations with different sweeping ordering is that each sweep will follow the causality of a group of characteristics in certain directions simultaneously and all characteristics can be divided into a finite number of such groups according to their directions. The value at each grid point is always nonincreasing during the iterations due to the updating rule. Whenever a grid point obtains the minimal value it can reach, the value is the correct value and the value will not be changed in later iterations.

---

**Algorithm 5:** Fast Sweeping Method

---

Initialization
**for all** grid points $x_{ij} \in G$ **do**
  **if** $x_{ij} \in G_{out}$ **then**
    **if** $x_{ij} \in G_b$ **then**
      $w_{ij} \leftarrow g(x_{ij})$
    **end if**
    $w_{ij} \leftarrow 0$
  **else**
    $w_{ij} \leftarrow \infty$
  **end if**
**end for**
End of initialization
Execution
$d \leftarrow 0$
**repeat**
  sweeped $\leftarrow$ FALSE
  $(i_0, i_1)(j_0, j_1) \leftarrow$ Sweeping Order$(d)$
  **for** $i = i_0 \ldots i_1$ **do**
    **for** $j = j_0 \ldots j_1$ **do**
      **if** $x_{ij} \in G_{in}$ **then**
        compute the value $(w_{ij})_u$ using equation (3.4)
        **if** $(w_{ij})_u < w_{ij}$ **then**
          $w_{ij} \leftarrow (w_{ij})_u$
          sweeped $\leftarrow$ TRUE
        **end if**
      **end if**
    **end for**
  **end for**
  $d \leftarrow d + 1$
**until** sweeped == TRUE

---

We use the distance function as an example to illustrate the motivation. The distance function $d(\mathbf{x})$ to a set $\Gamma$ satisfies the Eikonal equation

$$|\nabla d(\mathbf{x})| = 1, \qquad d(\mathbf{x}) = 0, \quad \mathbf{x} \in \Gamma. \tag{3.5}$$

All characteristics of this equation are straight lines that radiate from the set $\Gamma$. In one dimension, the upwind differencing at the interior grid point $i$ is

$$[(w_i - \min(w_{i-1}, w_{i+1}))^+]^2 = \delta^2, \qquad 1 \leq i \leq i_M - 1. \tag{3.6}$$

We use two Gauss-Seidel iterations with sweeping orderings, $i = 0 : i_M$ and $i = i_M : 0$

successively, to solve the above system. The update of the distance value at grid $i$ simply becomes

$$w_i^{new} = \min(\min(w_{i-1}, w_{i+1}) + \delta, w_i). \tag{3.7}$$

Figure 3.2 shows how one sweep from left to right followed by one more sweep from right to left is enough to finish the calculation of the distance function. This follows because there are only two directions for the characteristics in one dimension, left to right or vice versa. In another word, the distance value at any grid point can be computed from either its left neighbor or right neighbor by exactly $d_i = \min(d_{i-1}, d_{i+1}) + \delta$. The first sweep will cover those characteristics that go from left to right; i.e., those grid points whose values are determined by their left neighbors are computed correctly. Similarly, in the second sweep all those grid points whose values are determined by their right neighbors are computed correctly. Since we only update the current value if the newly computed value is smaller, those values that have been calculated correctly in the first sweep have achieved their minimal possible values and will not be changed in the second sweep. Convergence in two sweeps is true for arbitrary Eikonal equations in one dimension. In the special case of the distance function, it is easy to see that the Fast Sweeping method finds the exact distance function in two sweeps.



(a) t e computed distance unction a ter t e rst e t to rig t auss eide s eeping

( ) t e computed distance unction a ter t e second rig t to e t auss eide s eeping
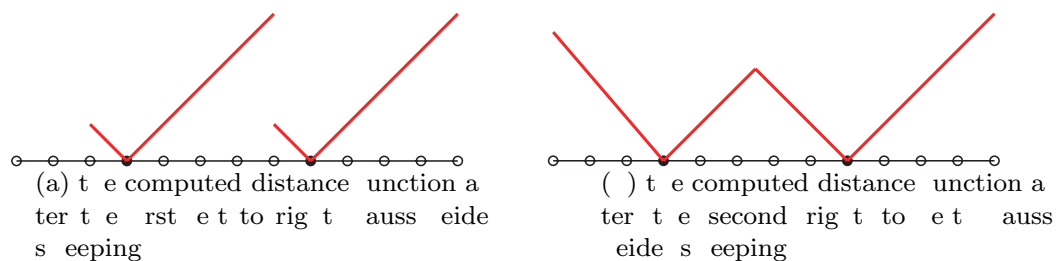
Figure 3.2.: The Fast Sweeping algorithm in one dimension.

In higher dimensions characteristics have infinitely many directions which cannot be followed exactly by the Cartesian grid lines.

In the case of one data point, the distance function is smooth except at the data point. For a more general data set, interactions of characteristics at their intersections can cause more than $2^n$ sweeps for the iteration to converge in $n$ dimensions. It is impossible to track the exact number of sweeps for the highly nonlinear discretized system in general. However, it is possible to show that in $n$ dimensions, after $2^n$ sweeps the Fast Sweeping method can compute a numerical solution to the distance function that is as accurate as the numerical solution after the iteration converges. This means $2^n$ sweeps is good enough in practice for computing the distance function to an arbitrary data set. For general Eikonal equations, the characteristics are curves instead of straight lines. So more than one sweep may be needed to cover one characteristic curve. Anyway, it is possible to prove that given a fixed domain and the right-hand side $f(\mathbf{x})$, the number of sweeps needed is still finite and is independent of grid size. For the proof of this last result, for the basic properties or more details in general on this method see [126].

# 4. Analysis and approximation of the Oren-Nayar model

In this Chapter, we will focus our attention on the Oren-Nayar model. We will analyze the parameters of the model, we will prove that the discrete operator of the corresponding SL scheme has the properties required by Theorem 2.3.1 and we will show some numerical tests carried out on synthetic and real images, in order to compare the performances of the ON–model with those of the classical L–model.

## 4.1. Analysis of the parameter $\sigma$

In the vertical light case, $\boldsymbol{\omega} = (0, 0, 1)$, we have obtained the expression (1.35), that is

$$I(\mathbf{x}) = \frac{A}{\sqrt{1 + |\nabla u(\mathbf{x})|^2}},$$

where

$$A = 1 - 0.5\,\sigma^2(\sigma^2 + 0.33)^{-1}.$$

In this expression we can see that there is an explicit dependence of the total light intensity $I(\mathbf{x})$ at the point $\mathbf{x} = (x, y)$ with respect to the euclidean norm of the gradient $|\nabla u(\mathbf{x})|$. This relation is also present in the L–model, in the PH–model or in the BP–model (see eqs. (1.8), (1.61), (1.78)).

In order to use these models, for which we have required $I(\mathbf{x}) \in [0, 1]$, we need to check that it is true with respect to the different values that the parameters can assume. Hence, in the specific case of the ON–model, we have to check this hypothesis with respect to the parameter $\sigma$. In this analysis we neglect the contribution of the ambient light ($k_A \equiv 0$), that can be considered as a constant as already said in Chapter 1.

**Remark 4.1** *Denoting by $p$ the euclidean norm of the gradient of the surface, i.e. $p = |\nabla u|$, in the L–model results*

$$\frac{\partial I}{\partial p} = -\frac{p}{(p^2 + 1)^{\frac{3}{2}}}.$$

*Then $I$ is monotone decreasing with respect to $|\nabla u|$, its maximum value is $I = 1$ in $p = 0$ and $I$ goes to zero in the limit of $p$ that goes to infinity. (see Fig. 4.1). Hence, the model is suitable to be used without variations for the reconstruction of real and synthetic images with values in $[0, 1]$.*
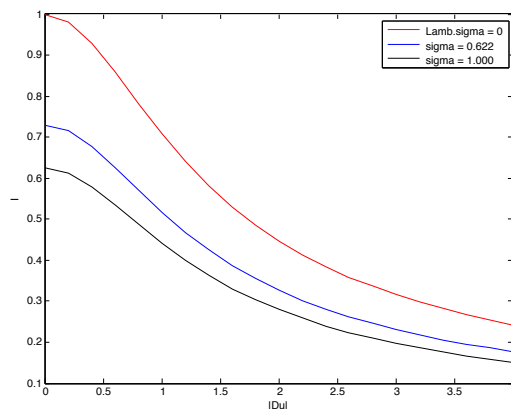
Figure 4.1.: In the ON–model, that coincides with the L–model for $\sigma = 0$, the value of the parameter $\sigma$ does not affects the monotony of $I$ with respect to $|\nabla u|$. $I$ is monotone non-increasing for values of $\sigma$ in the range $[0, \frac{\pi}{2}]$.

The dependence on $p$ for the ON–model can be shown as follows

$$\frac{\partial I}{\partial p} = -\frac{Ap}{(p^2 + 1)^{\frac{3}{2}}}.$$

Then $I$ is monotone decreasing with respect to $|\nabla u|$, its maximum value is $I = A$ in $p = 0$ and $I$ goes to zero in the limit of $p$ that goes to infinity, as seen for the L–model. Since $A$ depends on the parameter $\sigma$, we can derive the function $I$ with respect to $\sigma$ and we obtain

$$\frac{\partial I}{\partial \sigma} = -\frac{0.33\,\sigma}{\sqrt{1 + p^2(\sigma^2 + 0.33)^2}}. \tag{4.1}$$

This is a negative quantity for each value of $\sigma \in [0, \frac{\pi}{2}]$, hence $I$ is a monotone decreasing function with respect to this parameter. Therefore, also the ON–model is adequate for the reconstruction of real and synthetic images with values in $[0, 1]$.

## 4.2. Properties of the discrete operator $T^{ON}$

Let $W_i = w(x_i)$ so that $W$ will be the vector solution giving the approximation of the height $u$ at every node $x_i$ of the grid. Note that in one dimension $i$ is an integer number, in two dimensions $i$ denotes a multi-index, $i = (i_1, i_2)$. We consider a semi-Lagrangian discretization of (1.41) written in a fixed point form, so we will write the *SL fully discrete scheme for the ON–model* as

$$W_i = T_i^{ON}(W), \tag{4.2}$$

where $ON$ is the acronym identifying the ON–model. Denoting by $G$ the global number of nodes in the grid, the operator corresponding to the oblique light source is $T^{ON}$ :

$\mathbb{R}^G \to \mathbb{R}^G$ that is defined componentwise by

$$T_i^{ON}(W) = \min_{a \in \partial B_3} \{e^{-\mu h} w(x_i + h b^{ON}(x_i, a)) - \tau P^{ON}(x_i, z) a_3 (1 - \mu w(x_i))\} + \tau, \quad (4.3)$$

where

$$\tau = \frac{1 - e^{-\mu h}}{\mu},$$
$$b^{ON}(x_i, a) = \frac{1}{A\omega_3} \left(c(x_i, z) a_1 - A\omega_1, c(x_i, z) a_2 - A\omega_2\right),$$
$$c(x_i, z) = I(x_i) - B + B(d(x_i, z) \cdot \boldsymbol{\omega})^2, \quad (4.4)$$
$$d(x_i, z) = \nabla S(x_i, z) / |\nabla S(x_i, z)|,$$
$$P^{ON}(x_i, z) = \frac{c(x_i, z)}{A\omega_3},$$

and $w(x_i + h b^{ON}(x_i, a))$ is obtained by interpolating on $W$.

Now, let us check if the discrete operator $T^{ON}$ for the ON–model satisfies the three properties of the general Theorem 2.3.1 described in Section 2.3 on page 52, whose statement in this case is the following:

**Theorem 4.2.1** *Let $T_i^{ON}(W)$ the operator defined as in (4.3). Then $T_i^{ON}(W)$ has the following properties:*

1. *Let*
$$P^{ON}(x_i, z)\bar{a}_3 \le 1, \quad (4.5)$$
   *where* $\bar{a}_3 \equiv arg \min_{a \in \partial B_3} \{e^{-\mu h} w(x_i + h b^{ON}(x_i, a)) - \tau P^{ON}(x_i, z) a_3 (1 - \mu w(x_i))\}.$

   *Then $0 \le W \le \dfrac{1}{\mu}$ implies $0 \le T^{ON}(W) \le \dfrac{1}{\mu}$.*

2. *$T^{ON}$ is a monotone operator, i.e., $v \le u$ implies $T^{ON}(v) \le T(u)$.*

3. *$T^{ON}$ is a contraction mapping in $[0, 1/\mu)^G$ if $P^{ON}(x_i, z)\bar{a}_3 < \mu$.*

*Proof.*

1. To prove that $W \le \dfrac{1}{\mu}$ implies $T^{ON}(W) \le \dfrac{1}{\mu}$ we just note that

$$T^{ON}(W) \le \frac{e^{-\mu h}}{\mu} + \tau = \frac{1}{\mu}. \quad (4.6)$$

   Let $W \ge 0$; then

$$T^{ON}(W) \ge -\tau P^{ON}(x_i, z)\,\bar{a}_3(1 - \mu w(x_i)) + \tau = \tau \left(1 - P^{ON}(x_i, z)\,\bar{a}_3(1 - \mu w(x_i))\right). \quad (4.7)$$

This implies that $T^{ON}(W) \geq 0$ if $P^{ON}(x_i, z)\,\overline{a}_3 \leq 1$ since $0 \leq 1 - \mu w(x_i) \leq 1$.

2. In order to prove that $T^{ON}$ is monotone, assume that $v \leq u$ and let $a^*$ and $\overline{a}$ be the arguments of the minima. Then

$$
\begin{aligned}
T^{ON}(v) - T^{ON}(u) &\leq e^{-\mu h}[v(x + hb^{ON}(x, a^*)) - u(x + hb^{ON}(x, \overline{a}))] \\
&\quad -\tau P^{ON}(x, z)(a_3^*(1 - \mu v(x)) - \overline{a}_3(1 - \mu u(x))) \\
&\leq e^{-\mu h}[v(x + hb^{ON}(x, \overline{a})) - u(x + hb^{ON}(x, \overline{a}))] \\
&\quad +\tau P^{ON}(x, z)\overline{a}_3(v(x) - u(x)).
\end{aligned}
\tag{4.8}
$$

Hence,

$$
\begin{aligned}
T^{ON}(v) - T^{ON}(u) &\leq e^{-\mu h}[v(x + hb^{ON}(x, \overline{a})) - u(x + hb^{ON}(x, \overline{a}))] \\
&\quad +\tau P^{ON}(x, z)\overline{a}_3(v(x) - u(x)) \leq 0.
\end{aligned}
\tag{4.9}
$$

Note that this property does not require condition (4.5) to be satisfied.

3. Let
$$
P^{ON}(x_i, z)\overline{a}_3 < \mu.
\tag{4.10}
$$
To prove that $T^{ON}$ is a contraction mapping note that (4.9) yields

$$
T^{ON}(v) - T^{ON}(u) \leq \left(e^{-\mu h} + \tau P^{ON}(x, z)\overline{a}_3\right) ||v - u||_\infty.
\tag{4.11}
$$

Then, $T^{ON}$ is a contraction mapping if and only if

$$
e^{-\mu h} + \tau P^{ON}(x, z)\overline{a}_3 < 1
\tag{4.12}
$$

and this holds true if the bound (4.10) is satisfied.

$\square$

## 4.3. Hints on the implementation

The ON–model, as the L–model, chooses always to approximate the maximal viscosity solution, which sometimes can not coincide with the original surface that we want to reconstruct (see e.g. Test 2 in the next Section). In these cases, in order to obtain a reconstruction closer to the original desired surface, we will fix the value in the origin at zero. In this way we force the scheme to converge to a solution different from the maximal one. Moreover, the discrete operator defined in (4.3) shows that there are many contributions that can cause some problems in the implementation like for example $d(x_{ij}, z)$ defined in (4.4) as $d(x_{ij}, z) = \nabla S(x_{ij}, z)/|\nabla S(x_{ij}, z)|$, where we recall that $\nabla S(x_{ij}, z) = \left(\dfrac{-\nabla w(x_{ij})}{1 - \mu w(x_{ij})}, 1\right)$. In fact, we will see in the experiments presented in the

next Chapter 5 in order to compare the three models that the ON–model results highly sensitive to the change of parameter values in the oblique light cases since roundoff errors occur and limit the accuracy that can be achieved. In order to improve the results by guaranteeing the stability of the numerical scheme, we use a fixed time step $h$ for the implementation of the scheme in the oblique case.

## 4.4. Numerical tests

In this section we show some numerical tests to compare the two SL schemes related to the L–model and the ON–model. All the tests are contained in the paper [112]. Other numerical experiments will be presented in the next Chapter 5 in order to compare the performances with those of the PH–model too.

The algorithm for both the schemes is based on the fixed-point iteration

$$\begin{cases} V^n = T^M(V^{n-1}), \\ V^0 \text{given}. \end{cases} \tag{4.13}$$

where $M$ is the acronym identifying the model, then $M = L$ or $M = ON$.
All the numerical tests presented in this Chapter have been implemented in MATLAB R2012a. The computer used is a MacBook Pro 13" Intel Core 2 Duo with speed of 2.66 GHz and 4GB of RAM.

### 4.4.1. Synthetic images

For the synthetic images, we discretize the domain $Q$ with 151×151 nodes. The fixed point has been computed with an accuracy of $\eta = 10^{-4}$ and the stopping rule used is $\max(|V^{n+1} - V^n|) \leq \eta$.

**Test 1**

The first experiment is related to the paraboloid in $[-1.5, 1.5]$x$[-1.5, 1.5]$ described by the function

$$u(x, y) = \begin{cases} 1 - (x^2 + y^2) & \text{if } (x^2 + y^2) < 1, \\ 0 & \text{otherwise}, \end{cases} \tag{4.14}$$

with light direction $\omega = (0, 0, 1)$ and visible in Fig. 4.2.

As one can see in Fig. 4.3, there are no significant differences between the two surfaces reconstructed, but we can note from Tables 4.1, 4.2 and 4.3 that increasing the value of the roughness parameter $\sigma$ the error generated by the ON–model decreases and it is lower than the error for the L–model. Note that for $\sigma = 0$ we get exactly the same results (since the two models coincide). In Fig. 4.4 the background gray level is different because it has been computed via the model.
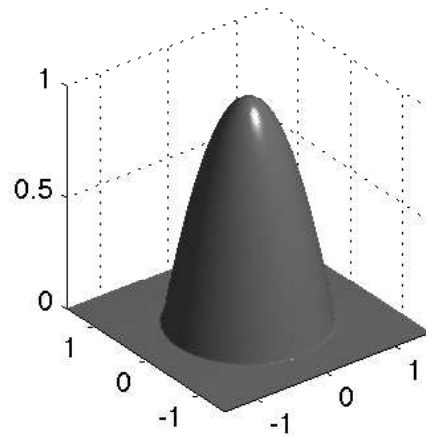
Figure 4.2.: Test 1: Original surface $u(x, y)$.
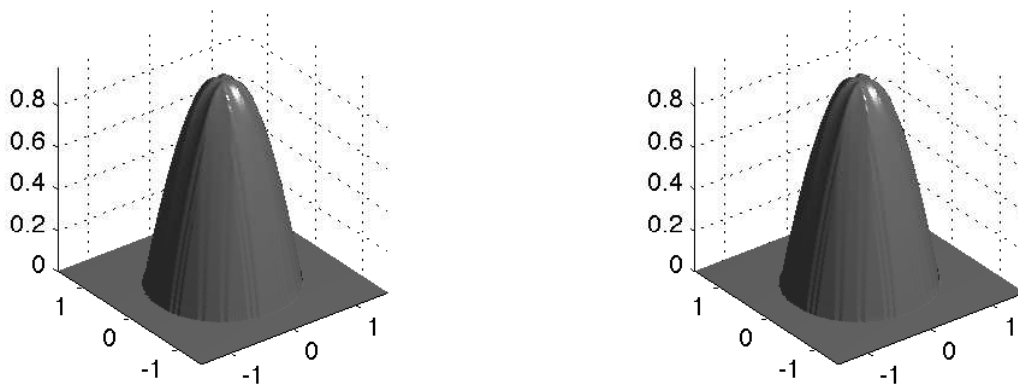


Figure 4.3.: Test 1: Surface reconstruction, L–model (left) and ON–model with $\sigma = 0.8$ (right).

Table 4.1.: Test 1: $L^\infty$ Error on the image for different values of $\sigma$.

| $\sigma$ | $L^\infty$ Error Lamb | $L^\infty$ Error ON |
|---|---|---|
| 0 | 0.074826 | 0.074826 |
| 0.3 | 0.074826 | 0.066809 |
| 0.5 | 0.074826 | 0.058700 |
| 0.8 | 0.074826 | 0.050141 |
| $\pi/2$ | 0.074826 | 0.041826 |

Table 4.2.: Test 1: $L^1$ Error on the image for different values of $\sigma$.

| $\sigma$ | $L^1$ Error Lamb | $L^1$ Error ON |
|----------|------------------|----------------|
| 0 | 0.028256 | 0.028256 |
| 0.3 | 0.028256 | 0.025228 |
| 0.5 | 0.028256 | 0.022166 |
| 0.8 | 0.028256 | 0.018934 |
| $\pi/2$ | 0.028256 | 0.015795 |

Table 4.3.: Test 1: Standard Deviation on the image for different values of $\sigma$.

| $\sigma$ | Std Dev. Lamb | Std Dev. ON |
|----------|---------------|-------------|
| 0 | 0.007426 | 0.007426 |
| 0.3 | 0.007426 | 0.006631 |
| 0.5 | 0.007426 | 0.005826 |
| 0.8 | 0.007426 | 0.004976 |
| $\pi/2$ | 0.007426 | 0.004151 |

Input I(x)　　　　　　　　　　　　　　Computed I(x)



Input I(x)　　　　　　　　　　　　　　Computed I(x)
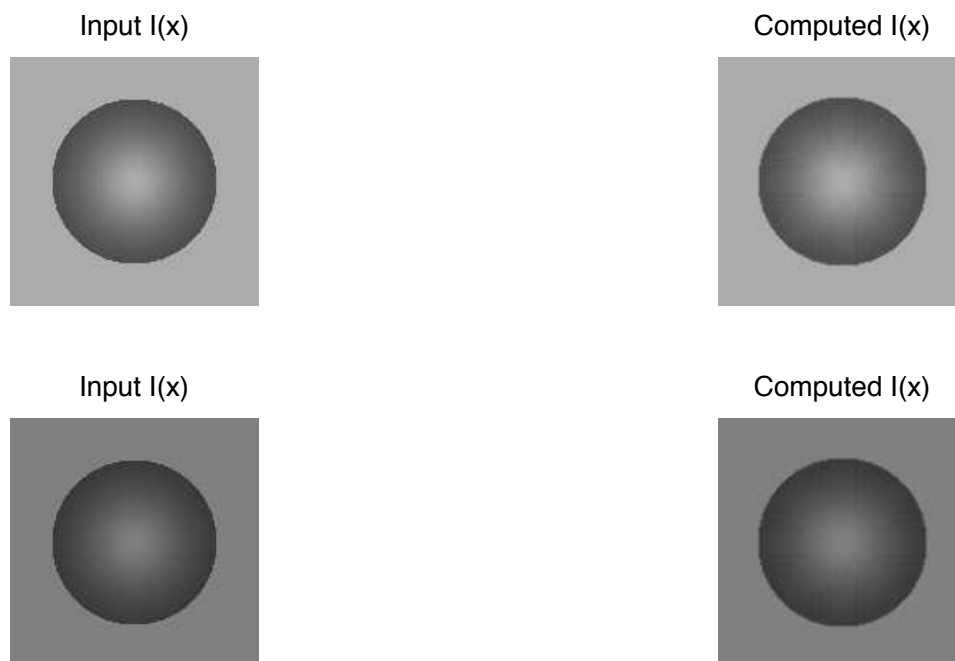


Figure 4.4.: Test 1: Images, ON–model with $\sigma = 0.8$ (up) and L–model (down).

**Test 2**

The second numerical test is related to the surface described by the function

$$u(x,y) = \begin{cases} -(1 - (x^2 - y^2))^2 + 1 & \text{if } (x^2 + y^2) < 2, \\ 0 & \text{otherwise,} \end{cases} \tag{4.15}$$

with light direction $\omega = (0, 0, 1)$ in the same domain of the previous test (See Fig. 4.5 for the input surface).

Looking at Fig. 4.6 we can note that both the schemes choose the maximal viscosity
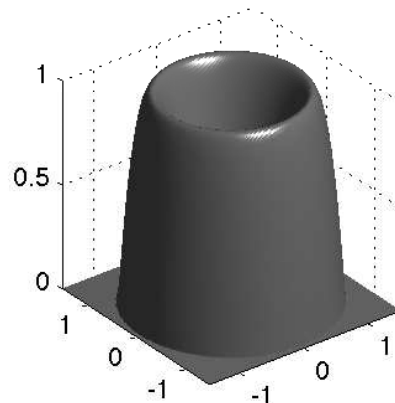


Figure 4.5.: Test 2: Original surface $u(x, y)$.

solution, which does not coincide with the original surface. In order to obtain a reconstruction closer to the original surface, we fix the value in the origin at zero. In this way we forced schemes to converge to a solution different from the maximal one (see Fig. 4.7). Also in this case we can see that the reconstruction of the surface is very similar for both the schemes. In Fig. 4.8 note that the background gray level is different for the same reason of the Fig. 4.4.

### 4.4.2. Real images

#### Test real 1: Beethoven bust

The next test is on a real-world image: the bust of Beethoven (see Fig. 4.9). The light direction is $\omega = (-0.19798, -0.01680, 0.98006)$ and the size of the input image is $77 \times 210$. Obviously, in the case of real image, the achievable accuracy is more sensitive to the values of $\sigma$ because the input image is given. After finding a correct value for the parameter $\sigma$, we can see again in Fig. 4.9 that the approximations generated by the two schemes are more o less the same, but the values in Tables 4.4, 4.5 and 4.6 show that the different errors on the image with the ON–model are lower than the errors obtained with the L–model. Note that the improvement is more evident in Table 4.6.

#### Test real 2: Real vase

The last test concerns the reconstruction of a vase enlightened by a vertical light source. The size of the input image is $128 \times 128$. We can see in Fig. 4.10 the approximated images with the two schemes on the right, starting from the same input image on the left. The reconstructed surface computed by both methods is shown in Fig. 4.11. As in
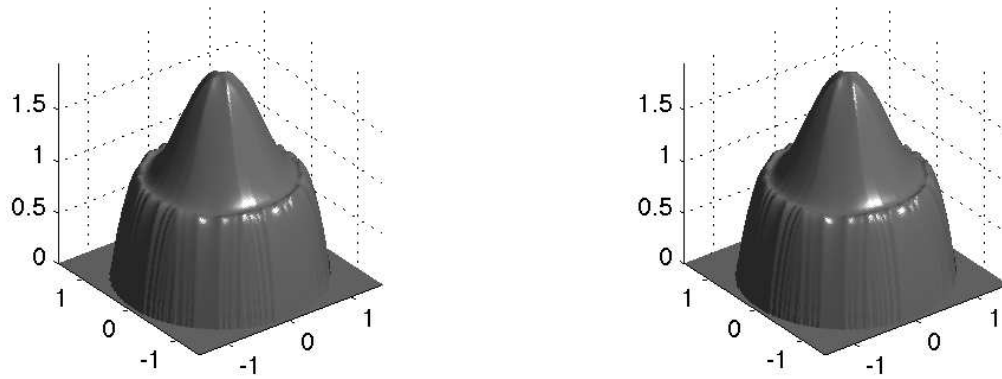
Figure 4.6.: Test 2: Approximated surface $u(x, y)$ by the two schemes that compute the maximal viscosity solution.
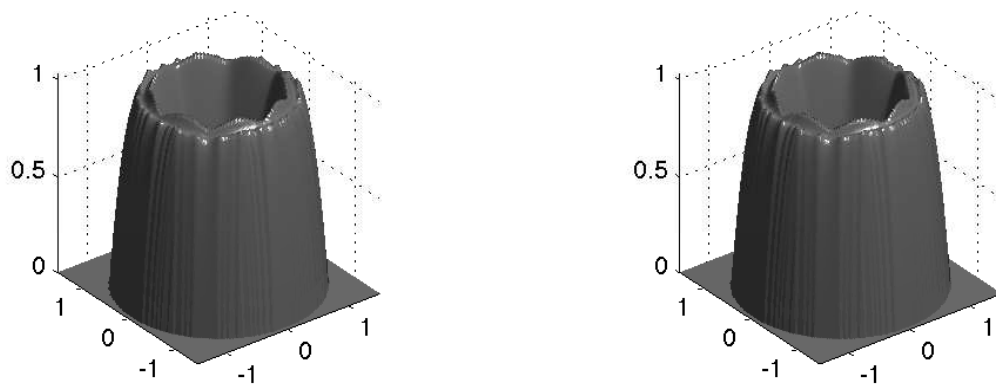


Figure 4.7.: Test 2: Approximated surface $u(x, y)$ with the setting $u(x, y) = 0$, L–model (left) and ON–model (right).

Table 4.4.: Beethoven Test: $L^\infty$ Error on the image for different values of $\sigma$.

| $\sigma$ | $L^\infty$ Error Lamb | $L^\infty$ Error ON |
|---|---|---|
| 0 | 0.635977 | 0.635977 |
| 0.2 | 0.635977 | 0.567406 |
| 0.4 | 0.635977 | 0.515963 |
| 0.5 | 0.635977 | 0.419684 |

the previous real test, the $L^\infty$ and the $L^1$ errors obtained with the Oren–Nayar approach are always lower than the Lambertian errors how we can note looking at Tables 4.7 and 4.8.

Other numerical tests will be presented in the next chapter 5, in order to compare the

Input I(x)

Computed I(x)

Input I(x)
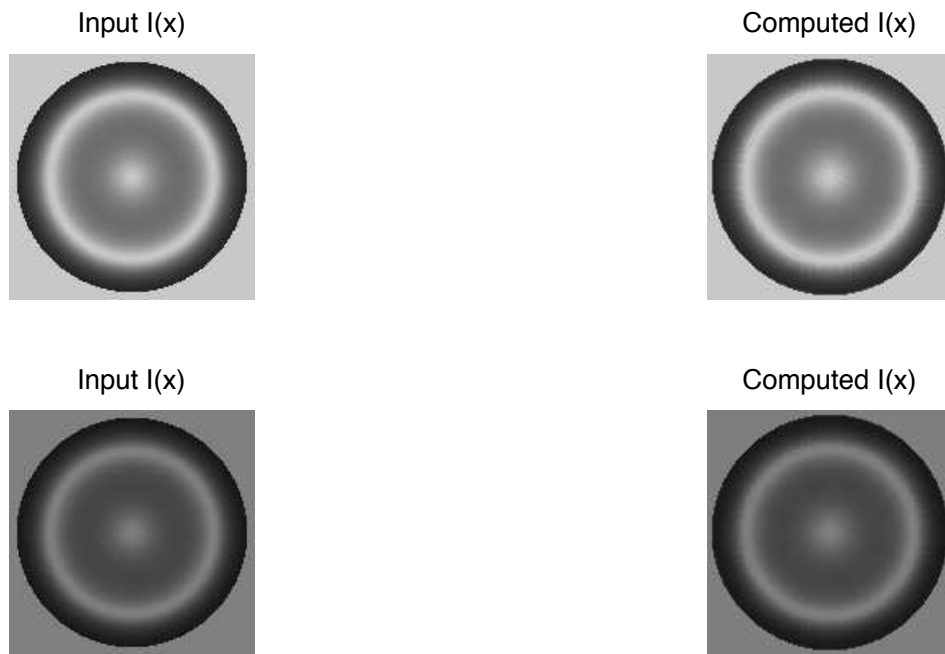
Computed I(x)

Figure 4.8.: Test 2: Images, ON–model with $\sigma = 0.5$ (up) and L–model (down).

Input I(x)

Computed I(x)

Computed I(x)

a)

b)

c)
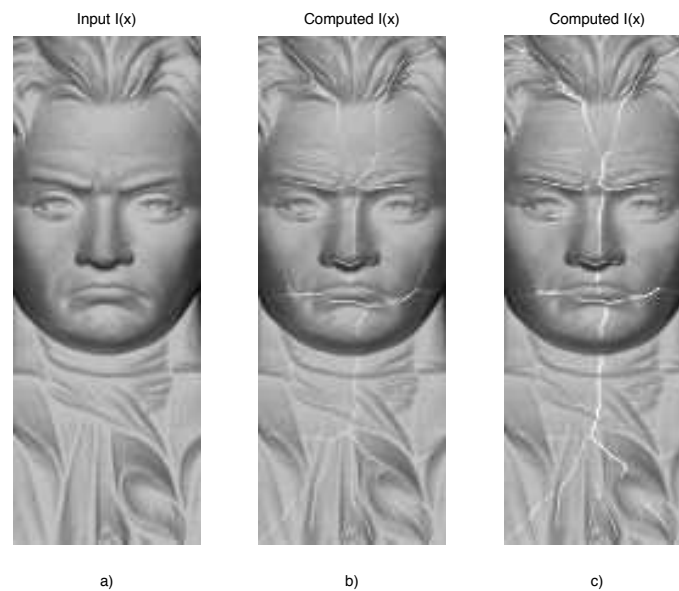
Figure 4.9.: a) Beethoven input image. b) Oren–Nayar computed image with $\sigma = 0.4$. c) Lambertian computed image.

Table 4.5.: Beethoven Test: $L^1$ Error on the image for different values of $\sigma$.

| $\sigma$ | $L^1$ Error Lamb | $L^1$ Error ON |
|---|---|---|
| 0 | 0.047027 | 0.047027 |
| 0.2 | 0.047027 | 0.045838 |
| 0.4 | 0.047027 | 0.043205 |
| 0.5 | 0.047027 | 0.042169 |

Table 4.6.: Beethoven Test: Standard deviation on the image for different values of $\sigma$.

| $\sigma$ | Std Dev. Lamb | Std Dev. ON |
|---|---|---|
| 0 | 0.056253 | 0.056253 |
| 0.2 | 0.056253 | 0.054361 |
| 0.4 | 0.056253 | 0.050138 |
| 0.5 | 0.056253 | 0.048308 |
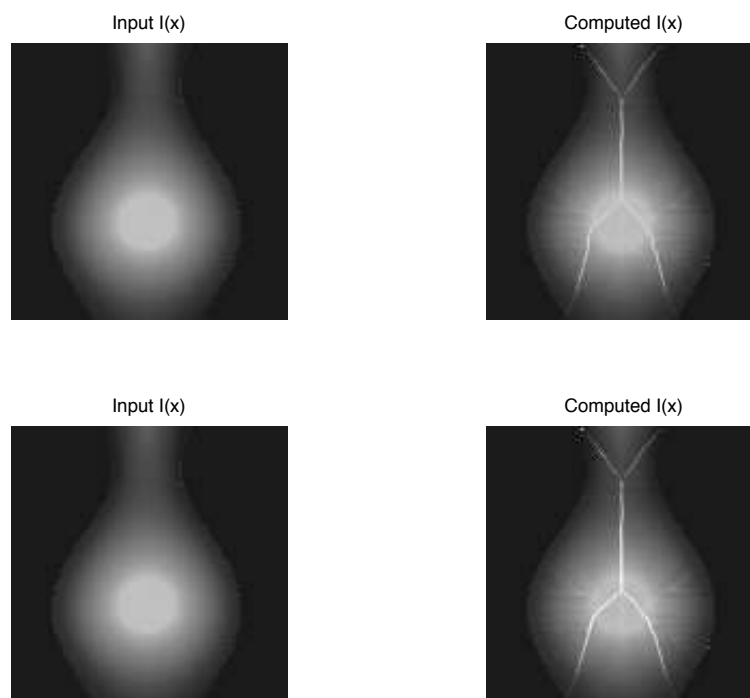


Figure 4.10.: Vase images: ON–model with $\sigma = 0.4$ (up) and L–model (down).

performances of the ON–model with those of the PH–model.
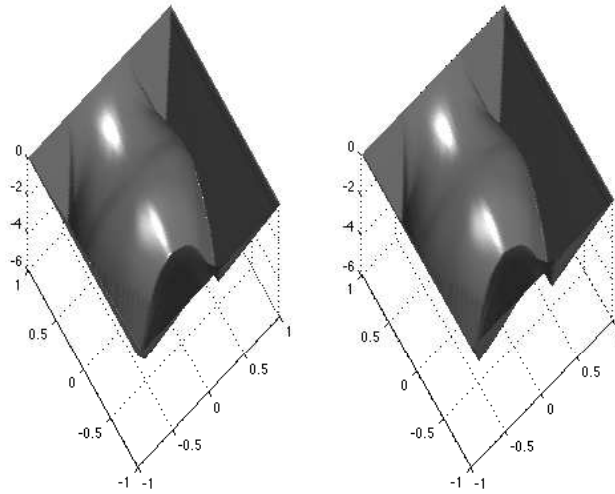
Figure 4.11.: Vase reconstruction: L–model (left) and ON–model (right).

Table 4.7.: Vase Test: $L^\infty$ Error on the image for different values of $\sigma$.

| $\sigma$ | $L^\infty$ Error Lamb | $L^\infty$ Error ON |
|---|---|---|
| 0 | 0.808202 | 0.808202 |
| 0.2 | 0.808202 | 0.766265 |
| 0.4 | 0.808202 | 0.678274 |
| 0.5 | 0.808202 | 0.634672 |

Table 4.8.: Vase Test: $L^1$ Error on the image for different values of $\sigma$.

| $\sigma$ | $L^1$ Error Lamb | $L^1$ Error ON |
|---|---|---|
| 0 | 0.028919 | 0.028919 |
| 0.2 | 0.028919 | 0.027292 |
| 0.4 | 0.028919 | 0.023764 |
| 0.5 | 0.028919 | 0.022190 |

Table 4.9.: Vase Test: Standard deviation on the image for different values of $\sigma$.

| $\sigma$ | Std Dev. Lamb | Std Dev. ON |
|---|---|---|
| 0 | 0.028751 | 0.028751 |
| 0.2 | 0.028751 | 0.026846 |
| 0.4 | 0.028751 | 0.022844 |
| 0.5 | 0.028751 | 0.021026 |

## 4.5. Conclusions

The non-Lambertian ON–model leads to rather complex nonlinear PDEs of the first order which can be treated in the framework of weak (viscosity) solutions. The analysis of this model shows that it is not able to solve the well known concave/convex ambiguity, what we expected since it is not possible to obtain uniqueness with a single image, despite the fact that it can deal with more general surfaces. From the numerical point of view, these equations can be approximated via semi-Lagrangian techniques in a rather effective way. The role of the roughness parameter $\sigma$ is crucial to obtain accurate results, playing with this parameter can in fact improve the approximation with respect to the classical L–model. In fact, the numerical tests carried out show that by increasing of the value of the parameter $\sigma$ the errors in the different norms decrease, resulting always less to those obtained using the classical L–model and equal to them in the case $\sigma = 0$ since in this case the two model coincide. In this respect, the non-Lambertian framework is more flexible and effective.

# 5. Analysis and approximation of the Phong model

In this Chapter we will focus our attention on the Phong model. We will analyze the parameters of the model, we will prove that the discrete operator of the corresponding SL scheme has the properties required by Theorem 2.3.1 and we will compare its performances with the classical L–model in terms of some error indicators on series of benchmarks images which come from synthetic and real images. Additionally, we will show other numerical experiments in order to compare the performances of the ON–model with those of the PH–model.

## 5.1. Analysis of the parameters $\alpha$ and $k_S$

In the case of vertical light source $\boldsymbol{\omega} = (0,0,1)$ and an oblique position of the observer $\mathbf{V}$, we associate the following PDE to the brightness equation (1.44)

$$I(\mathbf{x}) = k_D \gamma_D(\mathbf{x}) \frac{1}{\sqrt{1 + |\nabla u(\mathbf{x})|^2}} + k_S \gamma_S(\mathbf{x}) \left( \frac{1 - |\nabla u(\mathbf{x})|^2}{1 + |\nabla u(\mathbf{x})|^2} \right)^\alpha. \tag{5.1}$$

In this expression we have an explicit dependence of the total light intensity $I(\mathbf{x})$ in the point $\mathbf{x} = (x,y)$ with respect to the norm of the gradient $|\nabla u(\mathbf{x})|$.

As already stated for the ON–model, in order to use this model, for which we have required $I(\mathbf{x}) \in [0,1]$, we need to check if it is true with respect to the different values that the parameters can assume. Hence, we have to check this hypothesis with respect to the parameters $\alpha$ and $k_S$. In this analysis we neglect the contribution of the ambient light ($k_A \equiv 0$) and we suppose that the diffuse and specular albedo are uniform, i.e. $\gamma_D(\mathbf{x}) = \gamma_S(\mathbf{x}) = 1$. Since the ambient component is neglected and recalling that we must have $k_A + k_D + k_S = 1$, we can rewrite the diffuse $k_D$ as $k_D = 1 - k_S$ and then we rewrite the equation (5.1) as follow:

$$I(\mathbf{x}) = (1 - k_S) \frac{1}{\sqrt{1 + p^2}} + k_S \left( \frac{1 - p^2}{1 + p^2} \right)^\alpha, \tag{5.2}$$

where we have used the notation $p := |\nabla u(\mathbf{x})|$ to ease the writing.

In the case of the PH–model (see Fig. 5.1), the dependence on $\alpha$ is defined only for $p \le 1$ because the cosine in the specular term is replaced by zero if $\mathbf{R}(\mathbf{x}) \cdot \mathbf{V} < 0$ (and in that case we get back to the L–model, see Chapter 1 on page 12). For such values we
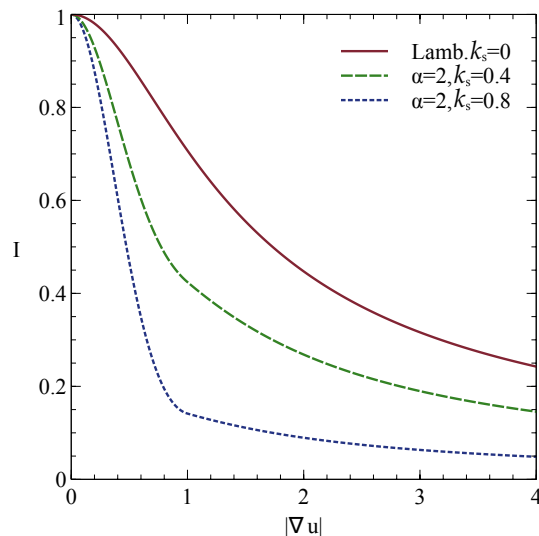
Figure 5.1.: In the PH–model, which coincides with the L–model when $k_S = 0$, the value of the parameter $k_S$ does not affect the monotony of $I$ with respect to $|\nabla u|$ and for any value of $k_S$ and $\alpha$, $I$ has always values in $[0, 1]$.

get

$$\frac{\partial I(\mathbf{x})}{\partial \alpha} = k_S \left(\frac{1-p^2}{1+p^2}\right)^{\alpha} \ln\left(\frac{1-p^2}{1+p^2}\right), \tag{5.3}$$

that is a negative quantity. The grey level intensity of the image, as function of the parameter $\alpha$, is monotone for $p \leq 1$ and it is constant for $p > 1$. If we compute the derivative with respect to $k_S$ we obtain

$$\frac{\partial I(\mathbf{x})}{\partial k_S} = \begin{cases} -\dfrac{1}{\sqrt{p^2+1}} & p > 1, \\[2mm] \left(\dfrac{1-p^2}{1+p^2}\right)^{\alpha} - \dfrac{1}{\sqrt{p^2+1}} & p \leq 1, \end{cases} \tag{5.4}$$

from which we deduce that the intensity is obviously monotone for $p > 1$. Instead, for $p \leq 1$, since the term in parenthesis is a quantity less than 1, we deduce that it decreases with the increase of $\alpha$. Finally, noting that for $\alpha = 1$

$$\left(\frac{1-p^2}{1+p^2}\right) - \frac{1}{\sqrt{p^2+1}} \leq 0, \tag{5.5}$$

we conclude that the intensity is monotone non-increasing also as a function of the

parameter $k_S$. Moreover, being

$$
\frac{\partial I(\mathbf{x})}{\partial p} = \begin{cases} -\frac{(1-k_S)\,p}{(p^2+1)^{\frac{3}{2}}} & p > 1 \\ -\frac{(p^2+1)^{-\alpha-\frac{1}{2}}\left(4\,\alpha\,k_S\,p\,(1-p^2)^\alpha\,\sqrt{p^2+1}+(p^2+1)^\alpha\left((k_S-1)\,p^3+(1-k_S)\,p\right)\right)}{1-p^4} & p \le 1, \end{cases}
\tag{5.6}
$$

we can verify that $I(\mathbf{x})$ is monotone also in $p$. In fact, the quantity $(k_S-1)\,p^3+(1-k_S)\,p$ is always nonnegative for $p \le 1$ and, hence, the numerator is always not positive.

Regarding the eikonal form of this model, that we obtain in the case of vertical light source and vertical position of the observer, we consider the definition of $Q(\mathbf{x})$ given in (1.65) with $k_D = 1 - k_S$ and so we write it as

$$
Q(\mathbf{x}) = (1 - k_S)^2 + 8k_S^2 + 8k_S I(\mathbf{x}).
\tag{5.7}
$$

Noting that $1 - k_S \ge I(\mathbf{x}) - k_S\ \forall \mathbf{x} \in R$, where $R$ is a rectangle domain containing the support of the image $\Omega$, omitting the dependence on the point $\mathbf{x}$ to facilitate the reading of what follows, we can write

$$
Q \ge (I - k_S)^2 + 8k_S^2 + 8k_S I,
$$

$$
Q \ge I^2 - 2k_S I + k_S^2 + 8k_S^2 + 8k_S I,
$$

$$
Q \ge I^2 + 6k_S I + 9k_S^2,
$$

arriving finally at the following inequality

$$
Q \ge (I + 3k_S)^2.
\tag{5.8}
$$

Thanks to this result and recalling again that $1 - k_S \ge I(\mathbf{x}) - k_S\ \forall \mathbf{x} \in R$, we can write for the numerator of the equation (1.62),

$$
\begin{aligned}
k_D^2 - 2I_+ I_- + k_D\sqrt{Q} &\ge \\
(1 - k_S)^2 - 2(I - k_S)(I + k_S) + (1 - k_S)(I + 3\,k_S) &\ge \\
(I - k_S)^2 - 2(I - k_S)(I + k_S) + (I - k_S)(I + 3\,k_S) &= \\
(I - k_S)\left[I - k_S - 2I - 2k_S + I + 3k_S\right] &= 0,
\end{aligned}
\tag{5.9}
$$

deducing that the expression of $f$ present in the eikonal equation corresponding to the PH–model (1.61) is always defined and real-valued. Thus, even for the PH–model, we conclude that it is appropriate to be used without changes for the reconstruction of real and synthetic images with values in $[0, 1]$.

## 5.2. Properties of the discrete operator $T^{PH}$

Let $W_i = w(x_i)$ so that $W$ will be the vector solution giving the approximation of the height $u$ at every node $x_i$ of the grid. Note that in one dimension $i$ is an integer number,

in two dimensions $i$ denotes a multi-index, $i = (i_1, i_2)$. We consider a semi-Lagrangian discretization of (1.54) written in a fixed point form, so we will write the *SL fully discrete scheme for the PH–model* as

$$W_i = T_i^{PH}(W), \tag{5.10}$$

where $PH$ is the acronym identifying the PH–model. Denoting by $G$ the global number of nodes in the grid, the operator corresponding to the oblique light source is $T^{PH} : \mathbb{R}^G \to \mathbb{R}^G$ that is defined componentwise by

$$T_i^{PH}(W) = \min_{a \in \partial B_3} \{e^{-\mu h} w(x_i + h b^{PH}(x_i, a)) - \tau P^{PH}(x_i, z) a_3 (1 - \mu w(x_i))\} + \tau, \tag{5.11}$$

where, in the case of oblique light source and vertical position of the observer,

$$\begin{aligned}
\tau &= \frac{1 - e^{-\mu h}}{\mu}, \\
b^{PH}(x_i, a) &= \frac{1}{Q^{PH}(x_i, z)} \left(c(x_i) a_1 - k_D \gamma_D(\mathbf{x}) \omega_1, c(x_i) a_2 - k_D \gamma_D(\mathbf{x}) \omega_2\right), \\
c(x_i) &= I(x_i) + \omega_3 k_S \gamma_S(\mathbf{x}), \\
d(x_i, z) &= \nabla S(x_i, z) / |\nabla S(x_i, z)|, \\
Q^{PH}(x_i, z) &= 2 k_S \gamma_S(\mathbf{x})(d(x_i, z) \cdot \boldsymbol{\omega}) + k_D \gamma_D(\mathbf{x}) \omega_3, \\
P^{PH}(x_i, z) &= \frac{c(x_i)}{Q^{PH}(x_i, z)},
\end{aligned} \tag{5.12}$$

and $w(x_i + h b^{PH}(x_i, a))$ is obtained by interpolating on $W$.

Now, as already done for the ON–model in Section 4.2 on page 66, we will check if the discrete operator $T^{PH}$ for the PH–model satisfies the three properties of the general Theorem 2.3.1 described in Section 2.3 on page 52, whose statement in this case is the following:

**Theorem 5.2.1** *Let $T_i^{PH}(W)$ the operator defined as in* (5.11). *Then $T_i^{PH}(W)$ has the following properties:*

1. *Let*

$$P^{PH}(x_i, z)\overline{a}_3 \leq 1, \tag{5.13}$$

   *where* $\quad \overline{a}_3 \equiv arg \min_{a \in \partial B_3} \{e^{-\mu h} w(x_i + h b^{PH}(x_i, a)) - \tau P^{PH}(x_i, z) a_3 (1 - \mu w(x_i))\}.$

   *Then $0 \leq W \leq \dfrac{1}{\mu}$ implies $0 \leq T^{PH}(W) \leq \dfrac{1}{\mu}$.*

2. *$T^{PH}$ is a monotone operator, i.e., $v \leq u$ implies $T^{PH}(v) \leq T(u)$.*

3. *$T^{PH}$ is a contraction mapping in $[0, 1/\mu)^G$ if $P^{PH}(x_i, z)\overline{a}_3 < \mu$.*

*Proof.*

1. To prove that $W \leq \dfrac{1}{\mu}$ implies $T^{PH}(W) \leq \dfrac{1}{\mu}$ we just note that

$$T^{PH}(W) \leq \frac{e^{-\mu h}}{\mu} + \tau = \frac{1}{\mu}. \qquad (5.14)$$

Let $W \geq 0$; then

$$T^{PH}(W) \geq -\tau P^{PH}(x_i, z)\,\overline{a}_3(1 - \mu w(x_i)) + \tau = \tau \left(1 - P^{PH}(x_i, z)\,\overline{a}_3(1 - \mu w(x_i))\right). \qquad (5.15)$$

This implies that $T^{PH}(W) \geq 0$ if $P^{PH}(x_i, z)\,\overline{a}_3 \leq 1$ since $0 \leq 1 - \mu w(x_i) \leq 1$.

2. In order to prove that $T^{PH}$ is monotone, assume that $v \leq u$ and let $a^*$ and $\overline{a}$ be the arguments of the minima. Then

$$
\begin{aligned}
T^{PH}(v) - T^{PH}(u) \quad \leq \quad & e^{-\mu h}[v(x + h b^{PH}(x, a^*)) - u(x + h b^{PH}(x, \overline{a}))] \\
& -\tau P^{PH}(x, z)(a_3^*(1 - \mu v(x)) - \overline{a}_3(1 - \mu u(x))) \\
\leq \quad & e^{-\mu h}[v(x + h b^{PH}(x, \overline{a})) - u(x + h b^{PH}(x, \overline{a}))] \\
& +\tau P^{PH}(x, z)\overline{a}_3(v(x) - u(x)).
\end{aligned}
\qquad (5.16)
$$

Hence,

$$
\begin{aligned}
T^{PH}(v) - T^{PH}(u) \quad \leq \quad & e^{-\mu h}[v(x + h b^{PH}(x, \overline{a})) - u(x + h b^{PH}(x, \overline{a}))] \\
& +\tau P^{PH}(x, z)\overline{a}_3(v(x) - u(x)) \leq 0.
\end{aligned}
\qquad (5.17)
$$

Note that this property does not require condition (5.13) to be satisfied.

3. Let

$$P^{PH}(x_i, z)\overline{a}_3 < \mu. \qquad (5.18)$$

To prove that $T^{PH}$ is a contraction mapping note that (5.17) yields

$$T^{PH}(v) - T^{PH}(u) \leq \left(e^{-\mu h} + \tau P^{PH}(x_i, z)\overline{a}_3\right) ||v - u||_\infty. \qquad (5.19)$$

Then, $T^{PH}$ is a contraction mapping if and only if

$$e^{-\mu h} + \tau P^{PH}(x_i, z)\overline{a}_3 < 1 \qquad (5.20)$$

and this holds true if the bound (5.18) is satisfied.

$\square$

## 5.3. Hints on the implementation

The discrete operator defined in (5.11) shows that there are many contributions that can cause some problems in the implementation like for example $d(x_{ij}, z)$ defined in (5.12) as $d(x_{ij}, z) = \nabla S(x_{ij}, z)/|\nabla S(x_{ij}, z)|$, where we recall that $\nabla S(x_{ij}, z) = \left( \dfrac{-\nabla w(x_{ij})}{1 - \mu w(x_{ij})}, 1 \right)$. In fact, we will see in the next section that the model results highly sensitive to the change of parameter values in the oblique light cases since roundoff errors occur and limit the accuracy that can be achieved. In order to improve the results of the approximation we adopt the following formula for the approximation of $d(x_{ij}, z)$: because the quantity $d(x_{ij}, z)$ have to be compute at each grid point for each iteration since depends on the iterative variable $W$ of the fixed point scheme, we first tried to approximate it by the "minmod" technique, i.e. choosing the less in absolute value between the right and the left finite differences on each direction (on the axis $x$ and $y$) of $w(x_{ij})$ at the point $(i, j)$ of the grid (see [102] for more details on this technique). But then we noted that considering the average of the four neighboring points of $(i, j)$, i.e. the value $w$ at $(i + 1, j), (i - 1, j), (i, j + 1), (i, j - 1)$ computed by the minmod finite derivative, plus the itself point $w(i, j)$ computed in the same way, we can obtain in some cases one order of magnitude more for accuracy because in this way we regularize a bit the gradient in question, making the function less discontinuous. Hence, in the numerical tests presented here we will use this last approximation. Of course, this suggestion cannot solve the difficulties that we will meet in the oblique light case to set manually the parameters involved in the model.

## 5.4. Numerical tests

In this section we show some numerical experiments on synthetic images in order to analyze the behavior of the parameters involved in the PH–model and to compare the performances of the PH–model with respect to the L–model. Other numerical experiments on synthetic and real images will be done in the next section in order to compare the performances of the PH–model with those of the ON–model too. All the numerical tests inserted in this chapter have been implemented in language C++. The computer used to compute the CPU times is a MacBook Pro 13" Intel Core 2 Duo with speed of 2.66 GHz and 4GB of RAM.

We call $G$ the discrete grid of points $x_{ij}$, with size $card(G) = n \times m$. We define $G_{in} := \{x_{ij} : x_{ij} \in \Omega\}$ as the set of grid points inside $\Omega$; $G_{out} := G \setminus G_{in}$. The boundary $\partial\Omega$ is defined as the set $G_b \subset G_{out}$ such that at least one of the neighboring points belongs to $G_{in}$. For each image we define a map, called *mask* or *silhouette*, where the pixels $x_{ij} \in G_{in}$ are white and the pixel $x_{ij} \in G_{out}$ are black. In this way it is easy to distinguish the nodes that we have to use for the reconstruction (the nodes inside $\Omega$) and the nodes on the boundary $\partial\Omega$ (see e.g. Fig. 5.2(b)).

All the synthetic images are defined on the same domain $G$, that is a rectangle containing the support of the image $\Omega$, $G \equiv [-1, 1] \times [-1, 1]$. We can easily modify the number of the pixels choosing different values for the steps in space $\Delta x$ and $\Delta y$. All the

synthetic images have $256 \times 256$ pixels, unless otherwise specified. For all it is possible to define, if useful, the function $g(\mathbf{x})$, that is the border of the object. $X$ and $Y$ represent the real size (e.g. for $G \equiv [-1, 1] \times [-1, 1]$, $X = 2, Y = 2$).

**Test 1: Paraboloid**

The paraboloid is defined on $G_{in} \colon \{x^2 + y^2 < r^2\}$ in the following way

$$\begin{cases} u(x, y) = r^2 - x^2 - y^2, & (x, y) \in G_{in}, \\ u(x, y) = 0, & (x, y) \in G_{out}, \end{cases} \tag{5.21}$$

where

$$r = \frac{\min\{X, Y\}}{2} - 2\,\tilde{\delta},$$

with $\tilde{\delta} = \max\{\Delta x, \Delta y\}$. As example, we can see in Fig. 5.2 the input image, the corresponding mask and the surface reconstructed by the L–model.



(a) Paraboloid Input    (b) Paraboloid Mask    (c) Paraboloid surface
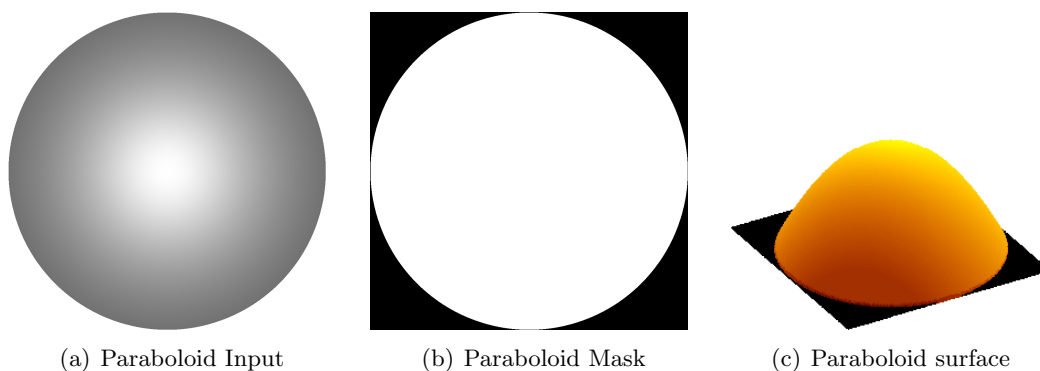
Figure 5.2.: Paraboloid via the L–model: (a) Input image; (b) Mask; (c) surface.

By using the values of the parameters reported in Table 5.1, we can note in Table 5.2 that when the specular component is zero, so we just have the contribution of the diffuse component, we retrieve exactly the errors values and the number of iterations of the L–model as expected. Then, by increasing the value of the coefficient $k_S$ and, as a consequence, decreasing the value of $k_D$, we can note that for the PH–model the $L^1(I)$, $L^2(I)$ and $L^\infty(I)$ errors on the image grow albeit slightly and remaining of the same order of magnitude, while the errors on the surface decrease until the use of intermediate values of the parameters (with $k_D = 0.4$ and $k_S = 0.6$) and then increase.

Table 5.1.: Synthetic paraboloid: parameter values used in the models.

| Model | $k_A$ | $k_D$ | $k_S$ | $\alpha$ |
|---|---|---|---|---|
| LAM | 0 | | | |
| PH-s00 | 0 | 1 | 0 | 1 |
| PH-s02 | 0 | 0.8 | 0.2 | 1 |
| PH-s04 | 0 | 0.6 | 0.4 | 1 |
| PH-s06 | 0 | 0.4 | 0.6 | 1 |
| PH-s08 | 0 | 0.2 | 0.8 | 1 |
| PH-s10 | 0 | 0 | 1 | 1 |

Table 5.2.: Synthetic paraboloid: vertical light source $\boldsymbol{\omega} = (0, 0, 1)$.

| SL–Schemes | Iter. | $[s]$ | $L^1(I)$ | $L^2(I)$ | $L^\infty(I)$ | $L^1(S)$ | $L^2(S)$ | $L^\infty(S)$ |
|---|---|---|---|---|---|---|---|---|
| LAM | 1756 | 1.84 | 0.0001 | 0.0013 | 0.0235 | 0.0030 | 0.0035 | 0.0068 |
| PH-s00 | 1756 | 2.31 | 0.0001 | 0.0013 | 0.0235 | 0.0030 | 0.0035 | 0.0068 |
| PH-s02 | 1756 | 1.91 | 0.0002 | 0.0020 | 0.0353 | 0.0025 | 0.0030 | 0.0067 |
| PH-s04 | 1758 | 1.96 | 0.0002 | 0.0027 | 0.0471 | 0.0015 | 0.0020 | 0.0055 |
| PH-s06 | 1762 | 1.93 | 0.0003 | 0.0034 | 0.0588 | 0.0008 | 0.0009 | 0.0069 |
| PH-s08 | 1784 | 2.01 | 0.0005 | 0.0041 | 0.0706 | 0.0063 | 0.0070 | 0.0161 |
| PH-s10 | 1314 | 1.58 | 0.0004 | 0.0048 | 0.0824 | 0.2513 | 0.2513 | 0.2547 |

**Test 2: Vase**

The synthetic vase is defined as

$$
\begin{cases}
u(x, y) = \sqrt{P(\bar{y})^2 - x^2} & (x, y) \in G_{in}, \\
u(x, y) = g(x, y) & (x, y) \in G_{out},
\end{cases}
\tag{5.22}
$$

where

$$
\bar{y} = y/Y,
$$
$$
P(\bar{y}) = (-10.8\,\bar{y}^6 + 7.2\,\bar{y}^5 + 6.6\,\bar{y}^4 - 3.8\,\bar{y}^3 - 1.375\,\bar{y}^2 + 0.5\,\bar{y} + 0.25)\,X
$$

and

$$
G_{in} = \{(x, y) | P(\bar{y})^2 > x^2\}.
$$

We utilize this test to study the behavior of the PH–model changing the value of the exponent $\alpha$ and comparing the performances of the FM and FS schemes.

We start to note looking at Fig. 5.3 that increasing the value of $\alpha$ (from $\alpha = 2$ to $\alpha = 20$) the spotlight areas becomes smaller and smaller and the surface appears more lucid and dark.

Looking at Table 5.3 we can note firstly that the errors are of the same order for the two schemes. Analyzing more in details the error values, the FS schemes gets a smaller
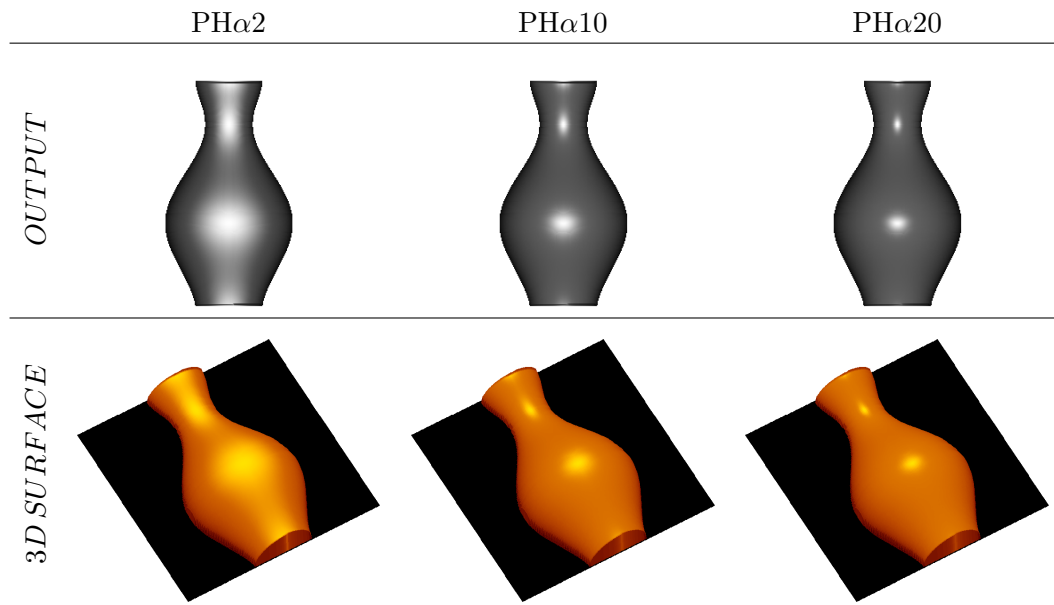
Figure 5.3.: Synthetic vase: output images and 3D reconstructions varying $\alpha$.

$L^1(I)$ and $L^2(I)$ error, but a bigger $L^\infty(I)$ error with respect to the FM method; instead it has a bigger $L^1(S)$ and $L^2(S)$ errors on the surface but same $L^\infty(S)$ error. The CPU time is very small, always less than 2 seconds, for both the schemes, but FM is always the fastest as expected.

Table 5.3.: Synthetic vase: FM and FS schemes with $k_S = 0.6$ and different values of $\alpha$.

| Model | Schemes | [s] | $L^1(I)$ | $L^2(I)$ | $L^\infty(I)$ | $L^1(S)$ | $L^2(S)$ | $L^\infty(S)$ |
|---|---|---|---|---|---|---|---|---|
| PH$\alpha$20 | FM | 0.37 | 0.0068 | 0.0275 | 0.7922 | 0.0195 | 0.0210 | 0.0425 |
| PH$\alpha$20 | FS | 1.80 | 0.0050 | 0.0236 | 0.8588 | 0.0205 | 0.0219 | 0.0425 |
| PH$\alpha$10 | FM | 0.29 | 0.0075 | 0.0320 | 0.8157 | 0.0197 | 0.0212 | 0.0425 |
| PH$\alpha$10 | FS | 1.64 | 0.0055 | 0.0271 | 0.8784 | 0.0208 | 0.0221 | 0.0425 |
| PH$\alpha$5 | FM | 0.14 | 0.0083 | 0.0377 | 0.8196 | 0.0201 | 0.0216 | 0.0428 |
| PH$\alpha$5 | FS | 1.66 | 0.0061 | 0.0317 | 0.8863 | 0.0211 | 0.0225 | 0.0428 |
| PH$\alpha$2 | FM | 0.25 | 0.0093 | 0.0463 | 0.8235 | 0.0205 | 0.0220 | 0.0434 |
| PH$\alpha$2 | FS | 0.76 | 0.0068 | 0.0386 | 0.8941 | 0.0215 | 0.0229 | 0.0434 |

**Test 3: A sinusoidal function**

The third synthetic numerical experiment is related to the sinusoidal function defined
as follow

$$\begin{cases} u(x,y) = 0.5 + 0.5 \, \sin(\dfrac{\pi \, x}{\Delta x}) \, \sin(\dfrac{\pi \, y}{\Delta y}), & (x,y) \in G_{in}, \\ u(x,y) = 0, & (x,y) \in G_{out}. \end{cases} \tag{5.23}$$

With this test we want to show that also the PH–model, as already seen for the ON–
model, not overcomes the concave/convex ambiguity typical of the SfS problem. In fact,
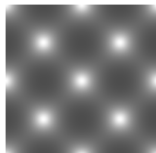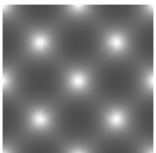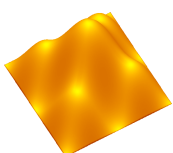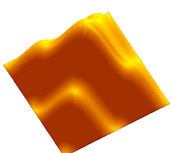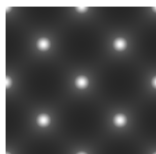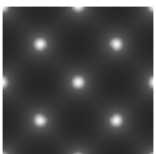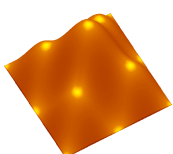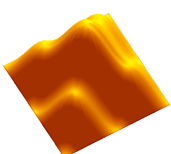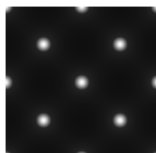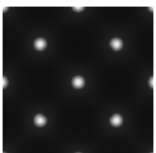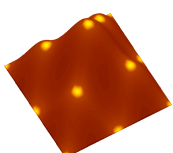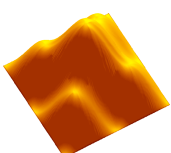
| Model | in | out | vertical source | oblique source |
|-------|----|----|-----------------|----------------|
| PH-s00 | | | | |
| PH-s05 | | | | |
| PH-s08 | | | | |



Figure 5.4.: Synthetic sinusoidal function: example of concave/convex ambiguity.

looking at Fig. 5.4 we can see that also varying the parameters $k_D$ and $k_S$ we can observe
the same phenomenon: the SL method always chooses the maximal solution.

## 5.5. Comparison between PH–model and ON–model

In this section we give other numerical experiments on synthetic and real images too in
order to compare the performances of the PH–model with those of the ON–model and
the L–model.

### 5.5.1. Synthetic images

As said in the previous section, all the synthetic images are defined on the same domain
$G$, that is a rectangle containing the support of the image $\Omega$, $G \equiv [-1,1] \times [-1,1]$. We
can easily modify the number of the pixels choosing different values for the steps in space
$\Delta x$ and $\Delta y$. All the synthetic images have $256 \times 256$ pixels, unless otherwise specified.

For all it is possible to define, if useful, the function $g(\mathbf{x})$, that is the border of the object. $X$ and $Y$ represent the real size (e.g. for $G \equiv [-1, 1] \times [-1, 1]$, $X = 2, Y = 2$).

**Test 1: Synthetic vase**

We start to compare the performances of the models on a synthetic test: the synthetic vase defined before via the equation (5.22). The input images generated by L–model, ON–model and PH–model are visible in Fig. 5.5. We show in Table 5.4 the values of



(a) L–model        (b) ON–model        (c) PH–model
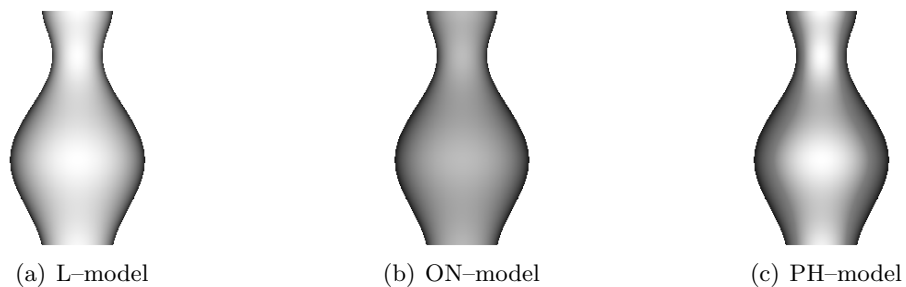
Figure 5.5.: Input vase images by L–model, ON–model ($\sigma = 0.6$), PH–model ($k_S = 0.3$).

the parameters related to some numerical tests performed. It is possible to compute error estimates in $L^1$, $L^2$, $L^\infty$ norm on the image ($L^p(I)$) and on the surface ($L^p(S)$) too because for synthetic images we know the real surface (for the vase this is given by (5.22)). The reconstruction of the surface and the output image obtained with the three

Table 5.4.: Synthetic vase: parameter values used in the models.

| Model | $k_A$ | $k_D$ | $k_S$ | $\alpha$ | $\sigma$ |
|-------|-------|-------|-------|----------|----------|
| LAM   | 0     |       |       |          |          |
| ON-00 | 0     |       |       |          | 0        |
| ON-04 | 0     |       |       |          | 0.4      |
| ON-06 | 0     |       |       |          | 0.6      |
| ON-10 | 0     |       |       |          | 1        |
| PH-00 | 0     | 1     | 0     | 1        |          |
| PH-03 | 0     | 0.7   | 0.3   | 1        |          |
| PH-07 | 0     | 0.3   | 0.7   | 1        |          |
| PH-10 | 0     | 0     | 1     | 1        |          |

models, starting from the input images in Fig. 5.5, are visible in Fig. 5.6.
Looking at the errors in Table 5.5 we note that the ON–model performs better increasing the parameter $\sigma$ both on the image $I$ and on the surface, with the same error order than the L–model. Instead, for the PH–model we can see that the errors on the surface decrease

Table 5.5.: Synthetic vase: vertical light source $\boldsymbol{\omega} = (0, 0, 1)$.

| SL–Schemes | Iter. | [s] | $L^1(I)$ | $L^2(I)$ | $L^\infty(I)$ | $L^1(S)$ | $L^2(S)$ | $L^\infty(S)$ |
|---|---|---|---|---|---|---|---|---|
| LAM | 1337 | 0.73 | 0.0063 | 0.0380 | 0.7333 | 0.0267 | 0.0286 | 0.0569 |
| ON-00 | 1337 | 0.72 | 0.0063 | 0.0380 | 0.7333 | 0.0267 | 0.0286 | 0.0569 |
| ON-04 | 1341 | 0.73 | 0.0054 | 0.0316 | 0.6118 | 0.0263 | 0.0282 | 0.0562 |
| ON-06 | 1344 | 0.75 | 0.0049 | 0.0277 | 0.5373 | 0.0259 | 0.0277 | 0.0553 |
| ON-10 | 1334 | 0.74 | 0.0044 | 0.0229 | 0.4510 | 0.0254 | 0.0274 | 0.0547 |
| PH-00 | 1337 | 0.76 | 0.0063 | 0.0380 | 0.7333 | 0.0267 | 0.0286 | 0.0569 |
| PH-03 | 1331 | 0.73 | 0.0068 | 0.0396 | 0.8078 | 0.0264 | 0.0283 | 0.0561 |
| PH-07 | 1356 | 3.81 | 0.0075 | 0.0419 | 0.9098 | 0.0235 | 0.0252 | 0.0496 |
| PH-10 | 737 | 0.40 | 0.0081 | 0.0472 | 0.9961 | 0.1496 | 0.1590 | 0.2309 |
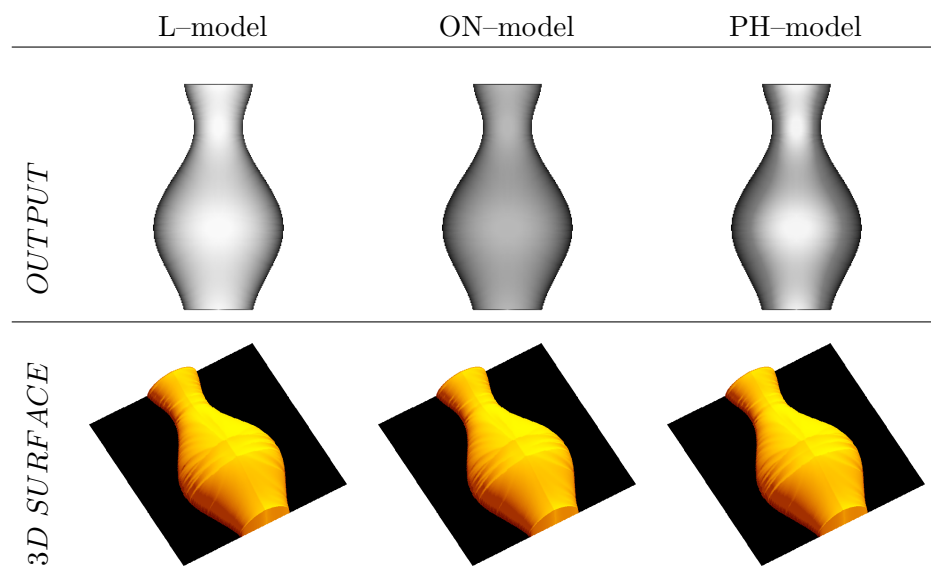


Figure 5.6.: Synthetic vase: output images and 3D reconstructions for the three models.

increasing the parameter $k_S$, except for the specular case ($k_S = 1$), but it is not true that the errors on the image decrease for increasing values of $k_S$. The best model for this test seems to be the ON–model with $\sigma = 1$.

### 5.5.2. Real images

Now, we will consider two real input images: the bust of Beethoven (size ($256 \times 256$)) and the black horse (size ($184 \times 256$)), both visible in Fig. 5.7(a), 5.7(c).
Unless otherwise specified, the value of $\eta$ for the stopping rule of the iterative method is fixed to $10^{-8}$ and the maximum number of allowed iterations is 9000. If a scheme arrives

| (a) Beethoven Input | (b) Beethoven Mask | (c) Horse Input | (d) Horse Mask |
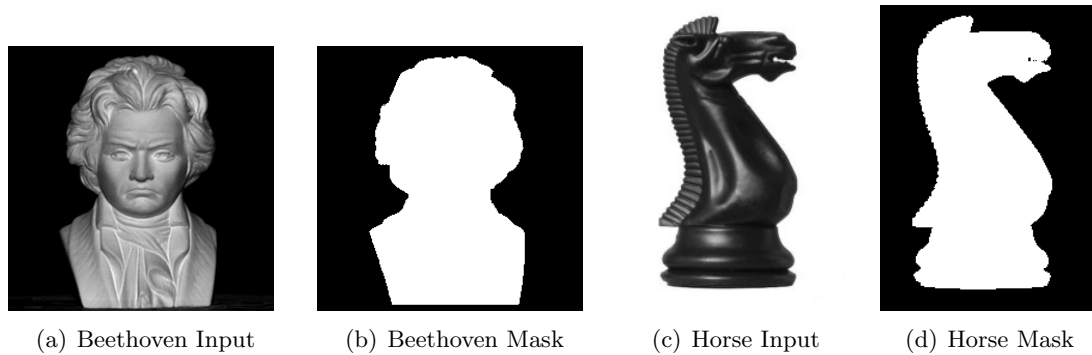
Figure 5.7.: Real input images and masks.

to the maximum of 9000 iterations, we put a $*$ before it in the table.

**Test 1: Beethoven**

In this case, we have compared the performances of the SL–schemes applied to the three models with the parameters reported in Table 5.6 with two different cases for the light source: the vertical case ($\boldsymbol{\omega}_{vert} = (0,0,1)$) and the oblique case ($\boldsymbol{\omega}_{obl} = (0.0168, 0.198, 0.9801)$). As we can see in Table 5.7, all the cases converge in less than

Table 5.6.: Beethoven: parameter values used in the models.

| Model | $k_A$ | $k_D$ | $k_S$ | $\alpha$ | $\sigma$ |
|-------|-------|-------|-------|----------|----------|
| LAM   | 0     |       |       |          |          |
| ON-00 | 0     |       |       |          | 0        |
| ON-01 | 0     |       |       |          | 0.1      |
| ON-02 | 0     |       |       |          | 0.2      |
| ON-03 | 0     |       |       |          | 0.3      |
| ON-04 | 0     |       |       |          | 0.4      |
| ON-06 | 0     |       |       |          | 0.6      |
| PH-00 | 0     | 1.0   | 0     | 1        |          |
| PH-01 | 0     | 0.9   | 0.1   | 1        |          |
| PH-02 | 0     | 0.8   | 0.2   | 1        |          |
| PH-03 | 0     | 0.7   | 0.3   | 1        |          |
| PH-04 | 0     | 0.6   | 0.4   | 1        |          |

three seconds with the same order of iteration. Looking at the errors on the images, they are of the same order for all the cases, $L^{\infty}(I)$ is a little bit higher for the ON–model with $\sigma = 0.6$. We can note that in the case of $\sigma = 0$ for the ON–model and $k_S = 0$ for the PH–model we obtain the same errors and number of iterations too because the three

models coincide as illustrated in Chapter 1. With respect to the ON–model, by increasing the value of $\sigma$ the errors grow slightly or remain unchanged. The same behavior has the PH–model with respect to the parameter $k_S$. In fact, by increasing the value of $k_S$ the errors tend to increase, remaining of the same order.

Table 5.7.: Beethoven: numerical results for $\boldsymbol{\omega}_{vert} = (0, 0, 1)$.

| SL–schemes | Iter. | [s] | $L^1(I)$ | $L^2(I)$ | $L^\infty(I)$ |
|---|---|---|---|---|---|
| LAM   vertical | 2920 | 1.68 | 0.0325 | 0.0605 | 0.4118 |
| ON-00 vertical | 2920 | 2.24 | 0.0325 | 0.0605 | 0.4118 |
| ON-01 vertical | 2885 | 2.89 | 0.0325 | 0.0605 | 0.4118 |
| ON-02 vertical | 2790 | 2.23 | 0.0326 | 0.0605 | 0.4118 |
| ON-06 vertical | 2264 | 1.94 | 0.0355 | 0.0628 | 0.4157 |
| PH-00 vertical | 2920 | 2.29 | 0.0325 | 0.0605 | 0.4118 |
| PH-01 vertical | 2676 | 2.12 | 0.0329 | 0.0609 | 0.4118 |
| PH-02 vertical | 2423 | 1.92 | 0.0333 | 0.0613 | 0.4118 |
| PH-03 vertical | 2160 | 1.92 | 0.0337 | 0.0617 | 0.4118 |
| PH-04 vertical | 1887 | 1.72 | 0.0337 | 0.0619 | 0.4118 |

Looking at Table 5.8, we can note that the oblique cases require higher CPU time with respect to the corresponding vertical cases due to the fact that the equations are more complex because of additional terms involved. Analyzing the errors on the images, as noted just before, the case of $\sigma = 0$ for the ON–model and $k_S = 0$ for the PH–model coincide with the L–model in terms of number of iterations and errors. With respect to the ON–model, the errors increase by increasing the parameter $\sigma$. The same holds for the PH–model with respect to $k_S$.

Because of additional terms involved in the oblique case, in Table 5.8 we have reported the results obtained using a value of the tolerance $\eta$ for the stopping rule of the iterative method equal to $10^{-3}$. This is the maximum accuracy achieved by the non-Lambertian models since roundoff errors coming from several terms occur and limit the accuracy. Only for the ON–model with $\sigma = 0.4$ we have reported the result also for $\eta = 10^{-4}$ and for the L–model with $\eta = 10^{-8}$. Lastly, we can note that choosing $k_S = 0.4$ the PH–model does not converge in the maximum number of allowed iterations, i.e. in 9000 iterations.

The $3D$ reconstructions and the output images obtained for the three models are visible in Fig. 5.8. The first two rows refer to the vertical case, the others to the oblique case. The reconstructions in the vertical cases are more accurate than the corresponding in the oblique case also because obtained with a tolerance $\eta = 10^{-3}$ instead of $\eta = 10^{-8}$ as in the vertical case. We can note that there is a concave/convex inversion in the surface due to classical ambiguity of the SfS model.

Table 5.8.: Beethoven: numerical results for $\boldsymbol{\omega}_{obl} = (0.0168, 0.198, 0.9801)$.

| SL–schemes | Iter. | [s] | $L^1(I)$ | $L^2(I)$ | $L^\infty(I)$ | $\eta$ |
|---|---|---|---|---|---|---|
| LAM   oblique | 3129 | 234.60 | 0.0397 | 0.0659 | 0.4039 | $10^{-8}$ |
| LAM   oblique | 236 | 40.85 | 0.0464 | 0.0696 | 0.4039 | $10^{-3}$ |
| ON-00 oblique | 236 | 46.85 | 0.0464 | 0.0696 | 0.4039 | $10^{-3}$ |
| ON-01 oblique | 242 | 50.90 | 0.0439 | 0.0656 | 0.4118 | $10^{-3}$ |
| ON-02 oblique | 262 | 53.43 | 0.0484 | 0.0699 | 0.4196 | $10^{-3}$ |
| ON-03 oblique | 270 | 53.76 | 0.0550 | 0.0763 | 0.4039 | $10^{-3}$ |
| ON-04 oblique | 314 | 65.63 | 0.0604 | 0.0830 | 0.4314 | $10^{-3}$ |
| ON-04 oblique | 3598 | 709.80 | 0.0672 | 0.0890 | 0.4314 | $10^{-4}$ |
| ON-06 oblique | 362 | 75.91 | 0.0722 | 0.0989 | 0.5647 | $10^{-3}$ |
| PH-00 oblique | 236 | 47.42 | 0.0464 | 0.0696 | 0.4039 | $10^{-3}$ |
| PH-01 oblique | 237 | 44.59 | 0.0712 | 0.0917 | 0.4510 | $10^{-3}$ |
| PH-02 oblique | 303 | 58.04 | 0.1095 | 0.1291 | 0.4784 | $10^{-3}$ |
| PH-03 oblique | 513 | 97.09 | 0.1506 | 0.1743 | 0.5333 | $10^{-3}$ |
| PH-04 oblique | 9000* | 1149.00 | 0.1701 | 0.2041 | 0.5765 | $10^{-3}$ |

∗ indicates the maximum number of iterations.

**Test 2: Black horse**

We use this test to compare the performances of the global SL–scheme with the acceleration schemes based on a finite difference solver (FM-FD, FS-FD). The comparison will be made for all the models (L–model, ON–model, PH–model) with the parameter values reported in the second and third column of Table 5.9. Note that the SL–scheme, that is slower than FM-FD and FS-FD methods as expected, however it is more accurate with respect to FD schemes. We can also note that the parameters play an important role in these models. For example, in the PH–model passing from $k_S = 0.4$ to $k_S = 0.8$ the errors change significantly in $L^1$ and $L^2$ norm for the FM-FD and FS-FD methods. In Fig. 5.9 one can see the output images and the 3D reconstructions of the surface obtained by the SL–scheme applied to the three models. The reconstruction obtained by the PH–model recognizes better the silhouette of the object considered so it seems to be a more realistic model, more suitable to deal with real images.
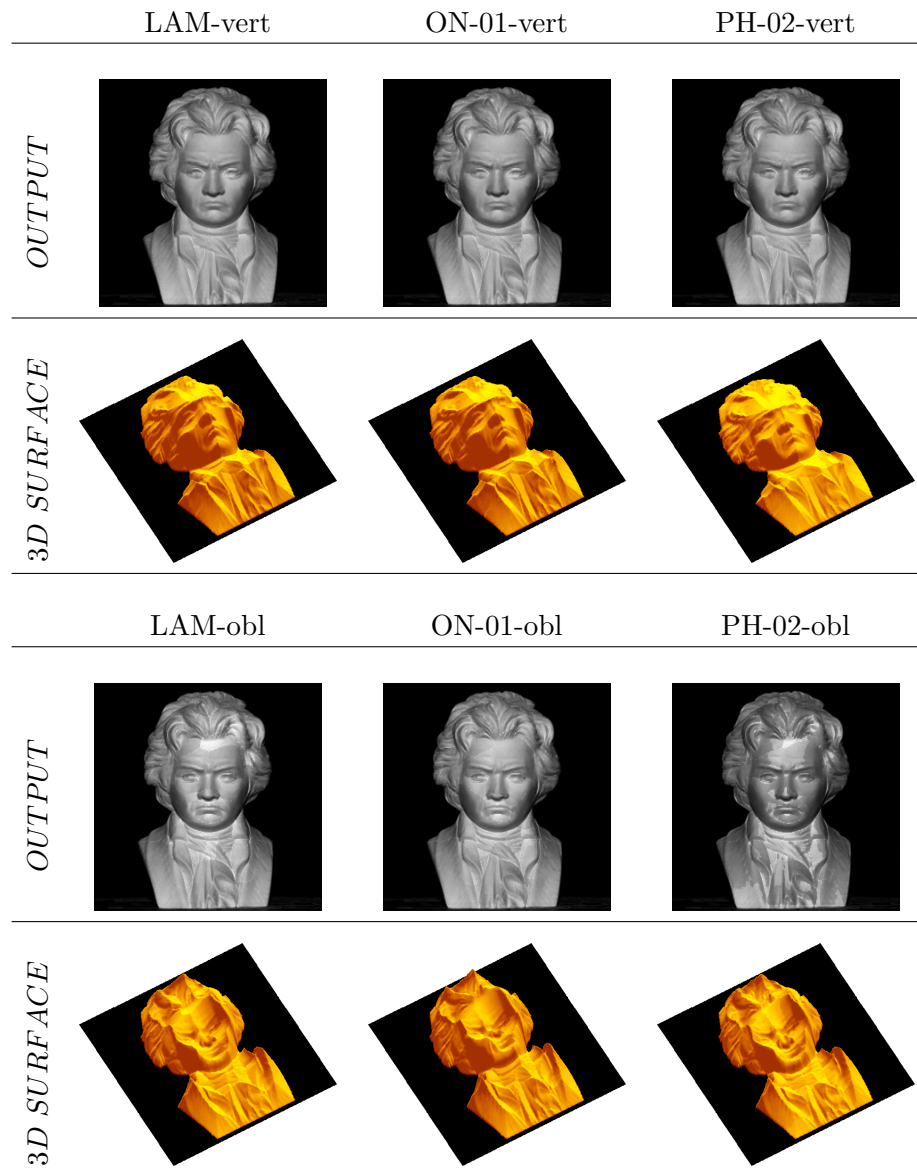
Figure 5.8.: Beethoven: output images and 3D reconstructions for the three models.

## 5.6. Conclusions

We have seen in this chapter as the PH–model leads to rather complex nonlinear PDE of the first order in which the parameters involved in the model play a crucial role to obtain accurate results. In fact, varying the value of the parameters it is possible to improve the approximation with respect to the classical L–model. The analysis performed on the model showed that it is not able to solve the well known concave/convex ambiguity

Table 5.9.: Black horse: parameters and errors of the models with vertical light source.

| Model | $k_S$ | $\sigma$ | $[s]$ | $L^1(I)$ | $L^2(I)$ | $L^\infty(I)$ |
|---|---|---|---|---|---|---|
| LAM-FM | | | 0.08 | 0.0363 | 0.0610 | 0.6902 |
| LAM-FS | | | 0.08 | 0.0362 | 0.0607 | 0.6902 |
| LAM-SL | | | 2.62 | 0.0346 | 0.0590 | 0.6863 |
| ON-02-FM | | 0.2 | 0.07 | 0.0363 | 0.0611 | 0.6902 |
| ON-02-FS | | 0.2 | 0.02 | 0.0362 | 0.0608 | 0.6902 |
| ON-02-SL | | 0.2 | 2.49 | 0.0347 | 0.0591 | 0.6902 |
| ON-03-FM | | 0.3 | 0.14 | 0.0364 | 0.0611 | 0.6941 |
| ON-03-FS | | 0.3 | 0.14 | 0.0363 | 0.0609 | 0.6941 |
| ON-03-SL | | 0.3 | 2.39 | 0.0348 | 0.0592 | 0.6902 |
| PH-04-FM | 0.4 | | 0.28 | 0.0441 | 0.0677 | 0.6902 |
| PH-04-FS | 0.4 | | 0.77 | 0.0439 | 0.0674 | 0.6902 |
| PH-04-SL | 0.4 | | 1.06 | 0.0358 | 0.0606 | 0.6863 |
| PH-08-FM | 0.8 | | 0.16 | 0.0788 | 0.1132 | 0.7098 |
| PH-08-FS | 0.8 | | 0.63 | 0.0788 | 0.1132 | 0.7098 |
| PH-08-SL | 0.8 | | 0.53 | 0.0463 | 0.0736 | 0.7059 |

despite the fact that it can deal with more general surfaces. Regarding the analysis of the parameters involved, we have seen that with respect to the parameter $k_S$, the errors on the surface decrease increasing the values of $k_S$, but it is not more true for the errors on the image. Moreover, the best results on the surface seem to be for values of $k_S$ intermediate between the extreme values 0 and 1, around $k_S = 0.6$.

With respect to the exponent $\alpha$, we have seen that increasing the value of $\alpha$ the spotlight areas becomes smaller and smaller and the surface appears more lucid and dark. For both the schemes used, FM and FS methods, the performances are better increasing the value of $\alpha$ and the errors on the image and on the surface too for the synthetic cases become lower increasing the exponent $\alpha$. This holds for both the schemes. From the numerical point of view, comparing the different techniques we noted that the SL–scheme approximates in a rather effective way the equation that describes the model and it is more accurate with respect to FD schemes although it is slower than FM-FD and FS-FD methods as expected. Looking at the numerical experiments made using the three models, the PH–model seems to approximate better the surface with respect to the L–model and the ON–model choosing an opportune value for $k_S$, although the errors computed on the images are higher than those obtained by the others two models. In this respect, the PH–model is more flexible and effective.

Focusing the attention on the tests performed with an oblique light source, we have to do some comments that are common to the PH–model and the ON–model. Several terms appear in these models and each of them gives a contribution to the roundoff error. Note that the accumulation of these roundoff errors makes difficult in the oblique case

| LAM-SL | ON-02-SL | PH-08-SL |
|--------|----------|----------|

*OUTPUT*

*3D SURFACE*
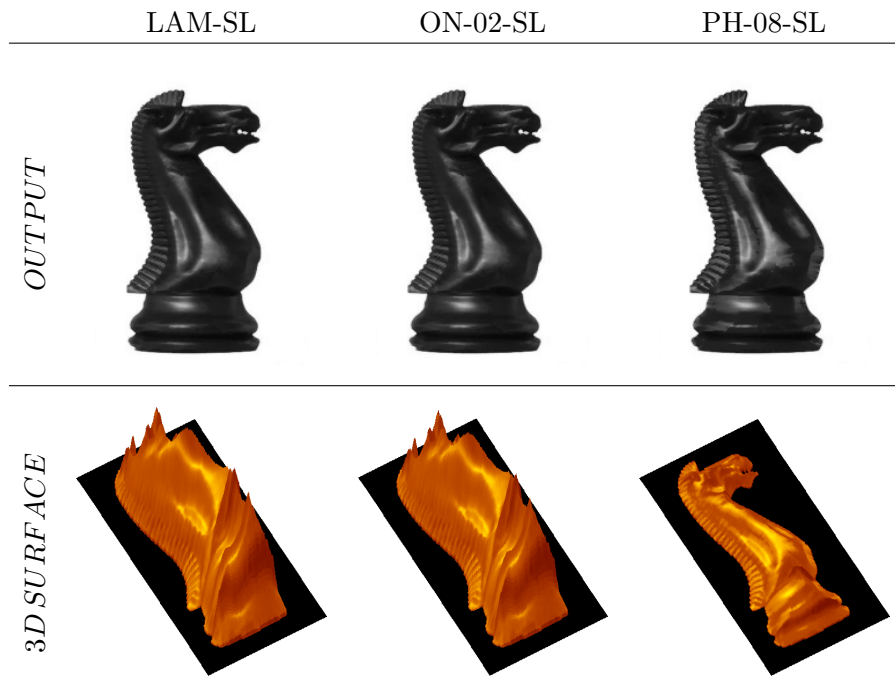
Figure 5.9.: Black horse: output images and 3D reconstructions for the three models.

to obtain a great accuracy. A possible improvement could be the use of second order schemes, that release the link between the space and the time steps which characterizes and blocks the accuracy for first order schemes.

# 6. Generalized Perspective Shape from Shading with Oren-Nayar Reflectance

In this Chapter we consider the class of modern perspective SfS models formulated via partial differential equations (PDEs). By combining a recent spherical surface parametrization with the advanced non-Lambertian Oren-Nayar reflectance model, we obtain a robust approach that allows to deal with an arbitrary position of the light source while being able to handle rough surfaces and thus more realistic objects. To our knowledge, the resulting model is currently the most advanced and most flexible approach in the literature on PDE-based perspective SfS. Apart from deriving our model, we also show how the corresponding set of sophisticated Hamilton-Jacobi equations can be efficiently solved by a specifically tailored Fast Marching scheme. Experiments with medical real-world data show that our model works in practice offering the desired flexibility.

Starting from the formulation of the SfS problem realized to the pioneering work of Horn [50, 54] using *partial differential equations* (PDEs), there has been enormous progress in the area of SfS. In particular, the appropriate modeling of light reflectance and scene illumination has played a key role in the design of successful SfS approaches.

With respect to the light reflectance, the classical orthographic model of Horn [50] based on Lambertian surfaces was soon replaced by advanced non-Lambertian reflectance models that offer an improved capability to handle realistic scenarios (see *e.g.* [4, 95]). Relying on *perspective* instead of orthographic projection, a different class of modern SfS approaches has emerged in the last years [78], most of them formulated again in terms of PDEs [108, 109, 90]. By assuming a Lambertian surface and the illumination given by a point light source located at the optical center of the camera, they feature many well-posedness properties in contrast to their orthographic counterparts. In particular, the quadratic light fall-off due to inverse square law turned out to be very useful in this context [91, 13]. Recently, such perspective SfS models with the light source located at the optical center of the camera have been extended to non-Lambertian reflectance models [3, 118] such as the Oren-Nayar model [3]. This model has also been investigated in [58] for the use with the highly efficient *Fast Marching* (FM) method [102].

Turning to more general illumination scenarios, there have been recent works on perspective SfS with Lambertian reflectance for a point light source *not* located at the optical center. While Wu *et al.* [122] presented a multi-image optimization framework, Galliani *et al.* [45] proposed a PDE-based approach for a single input image that leads to an efficient FM implementation. However, no advanced non-Lambertian reflectance models have been employed in the context of general illumination settings so far.

Summarizing, in the previous work on perspective SfS, approaches for non-Lambertian surfaces as well as for point light sources not co-located with the projection center have

only been considered *individually*. This shows that it is an extremely challenging task to relax the original SfS setup. In particular, there have been no attempts so far to combine a non-Lambertian reflectance model with a general illumination setup – even though, from an application viewpoint, the framework should be as flexible as possible.

In the current chapter, we advance the field by *successfully merging* the aforementioned two paths of research. (i) As in the work of Galliani *et al.* [45] we consider the use of a spherical coordinate system centered at the point light source. (ii) Instead of employing a simple Lambertian reflectance model, we formulate a new brightness equation based on the advanced reflectance model of Oren and Nayar. The resulting approach is a sophisticated PDE belonging to the class of *Hamilton-Jacobi equations* (HJEs) where the solution has to be understood in a viscosity sense, cf. [28, 97, 68, 91]. Apart from creating a very general and flexible approach, setting up our model in this way offers a third advantage: (iii) since for (advanced) diffuse reflectance models the brightest points correspond approximately to local minima of the depth [95], it allows us to extend the efficient FM algorithm proposed in [45] to our model – although our HJEs are significantly more complex. Experiments with medical real-world input images from endoscopy allow us to verify the properties of our approach, see *e.g.* [119, 109] for a similar application of SfS.

Hence first, we briefly recall the SfS setting in spherical coordinates and the Oren-Nayar reflectance model in the perspective case that serves as basis for our approach. Then, we proceed with the derivation of our new model and comment on its numerical approximation. Then, a presentation of computational results and conclusions will be given.

## 6.1. Why using Spherical Coordinates?

Although SfS methods make use of standard Cartesian coordinates [122, 3, 118], such coordinates have one decisive drawback illustrated in Fig. 6.1(a) and Fig. 6.1(b). When the light source is not located at the optical centre of the camera, the critical points, i.e. the points of the object with largest local height, are not the brightest points, i.e. the points that are closest to the light source.

In short: local intensity maxima do not identify critical points (local surface maxima). This is due to the fact that in SfS with Cartesian coordinates, the depth is measured along the $x_3$-axis (Fig. 6.1: vertical axis). Since identifying critical points is required to apply efficient algorithms of Fast Marching (FM) type [100, 101, 102, 116, 49] to solve the brightness equation in (6.7), we propose to use a spherical coordinate system with the origin located at the position of the light source. We measure the depth and thus the critical points from the viewpoint of the light source. By construction, in such a coordinate system, the locally brightest points in the image coincide again with the critical points. Typical assumption in this context are:

i) the albedo is assumed to be uniform,

ii) surface normals at local maxima are parallel to the direction of incoming light (per

(a) Light source located at optical centre. Brightest and highest point $(B, H)$ coincide.

(b) Light source *not* located at optical centre. Brightest and highest point $(B, H)$ differ(!).
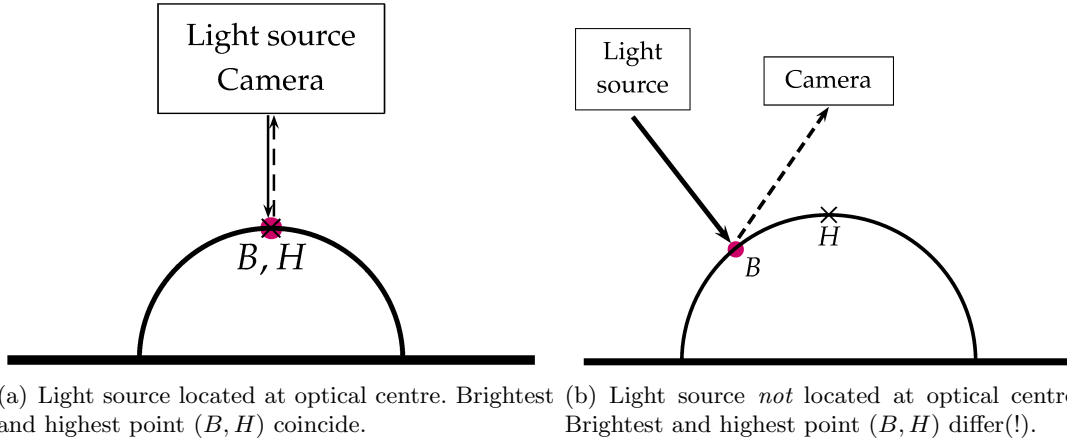
Figure 6.1.: Relationship between the brightest point $B$ (critical point) and the highest point $H$ of the object depending on the scene geometry. The problem of differing $H$ and $B$ is inherent to Cartesian coordinates. In a spherical coordinates with origin at the light source, $B$ is always $H$.

    definition of local maxima in our new coordinate system),

iii) remaining concave-convex ambiguities are resolved by the light fall-off factor $1/r^2$ in the brightness equation [91].

Let us now describe our parametrization.

## 6.2. Perspective SfS Based on Spherical Coordinates

Although most approaches in the SfS literature are still based on the assumption that the light source is located at the optical center of the camera, we follow Galliani *et al.* [45] and consider a general setup in which the light source is allowed to be located anywhere in the scene. The corresponding geometry described in Fig. 6.2 is based on a spherical coordinate system whose origin has been located at the position of the light source.

The main idea is to represent the Cartesian vector $\mathbf{r} = \overrightarrow{LS}$, i.e. the distance from the light source to the surface, via two angles $\theta$ and $\varphi$, respectively, as well as a radius $r$:

$$\mathbf{r} = R_{x_3}(\theta)\, R_{x_2}(\varphi) \begin{bmatrix} 0 \\ 0 \\ r \end{bmatrix} = \begin{bmatrix} \cos\theta\, \cos\varphi & -\sin\theta & \cos\theta\, \sin\varphi \\ \sin\theta\, \cos\varphi & \cos\theta & \sin\theta\, \sin\varphi \\ -\sin\varphi & 0 & \cos\varphi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ r \end{bmatrix}. \quad (6.1)$$

Here, the two matrices $R_{x_3}(\theta)$ and $R_{x_2}(\varphi)$ represent respectively the rotations about
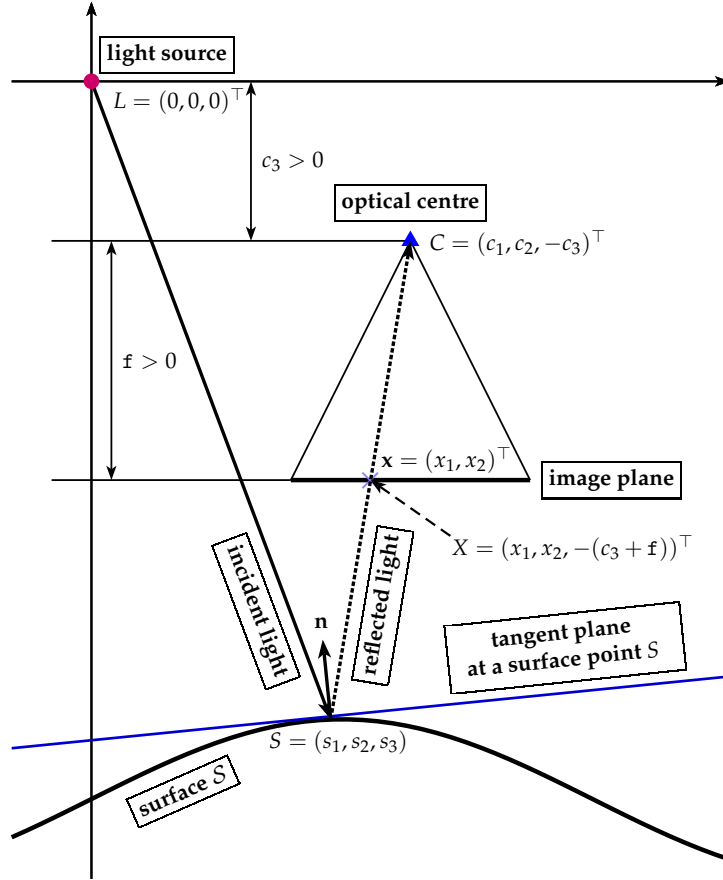
Figure 6.2.: General SfS setup with arbitrary position of the light source.

the $x_3$- and $x_2$-axis. The corresponding orthonormal basis is then given by

$$\mathbf{e}_\varphi = \begin{bmatrix} \cos\varphi\,\cos\theta \\ \cos\varphi\,\sin\theta \\ -\sin\varphi \end{bmatrix}, \quad \mathbf{e}_\theta = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ 0 \end{bmatrix}, \quad \mathbf{e}_r = \begin{bmatrix} \sin\varphi\,\cos\theta \\ \sin\varphi\,\sin\theta \\ \cos\varphi \end{bmatrix}, \qquad (6.2)$$

where

$$\varphi = \arccos\left(r_3/\sqrt{r_1^2 + r_2^2 + r_3^2}\right) \quad \text{and} \quad \theta = \arctan\left(r_2/r_1\right). \qquad (6.3)$$

In this basis defined by spherical coordinates, the position vector from the light source to a surface point can be compactly written as:

$$\begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}^\top =: \mathbf{r} := r\,\mathbf{e}_r \qquad \text{with} \qquad r = \sqrt{r_1^2 + r_2^2 + r_3^2}. \qquad (6.4)$$

Having parametrized the unknown surface in spherical coordinates, we now need to

compute the surface normal for each pixel of the input image. This can be done by first determining the vectors defining the tangent plane – these vectors are given by the derivatives of the surface with respect to the two directions orthogonal to $\mathbf{r}$, namely $\varphi$ and $\theta$ – and then by computing the cross product to obtain the corresponding normal vector. Using the definition of $\mathbf{r}$ from (6.4), the surface normal vector can be determined by

$$\mathbf{n} = \frac{\partial (r\mathbf{e}_r)}{\partial \theta} \times \frac{\partial (r\mathbf{e}_r)}{\partial \varphi} = r \frac{\partial r}{\partial \theta} \left( \mathbf{e}_r \times \frac{\partial \mathbf{e}_r}{\partial \varphi} \right) + r \frac{\partial r}{\partial \varphi} \left( \frac{\partial \mathbf{e}_r}{\partial \theta} \times \mathbf{e}_r \right) + r^2 \left( \frac{\partial \mathbf{e}_r}{\partial \theta} \times \frac{\partial \mathbf{e}_r}{\partial \varphi} \right).$$
$$(6.5)$$

Note that by (6.2), $\frac{\partial \mathbf{e}_r}{\partial \varphi} = \mathbf{e}_\varphi$ and $\frac{\partial \mathbf{e}_r}{\partial \theta} = \sin \varphi \, \mathbf{e}_\theta$, since $(\mathbf{e}_\varphi, \mathbf{e}_\theta, \mathbf{e}_r)$ constitutes a right-handed coordinate system. Therefore, we obtain

$$\mathbf{n} = r \frac{\partial r}{\partial \theta} \mathbf{e}_\theta + r \sin \varphi \frac{\partial r}{\partial \varphi} \mathbf{e}_\varphi - r^2 \sin \varphi \, \mathbf{e}_r \,. \tag{6.6}$$

This information is required later on in the reflectance model to establish the connection between the known image brightness and the unknown local orientation of the surface.

## 6.3. Perspective Oren-Nayar SfS Brightness Equation

Let us now explain the *Oren-Nayar* reflectance model. As already stated in Chapter 1, the advanced Oren-Nayar model [84, 82, 85, 83] explicitly allows to handle general *rough* surfaces. The idea of this model is to represent a rough surface as a collection of V-shaped cavities, each with Lambertian reflectance properties. Assuming the slopes of these cavities to be Gaussian distributed, the roughness of a surface can be characterized by a single parameter, namely the standard deviation $\sigma$ of its slope distribution.

For determining the irradiance in the perspective case, we consider additionally the inverse square law for the light fall-off. Thus, we obtain the following *brightness equation* for the Oren-Nayar model [83, 3]:

$$I(\mathbf{x}) = \frac{1}{r^2} \cos(\theta_i) \ \left( A + B \sin(\alpha) \tan(\beta) \max[0, \cos(\varphi_i - \varphi_r)] \right) \,, \tag{6.7}$$

where the two non-negative coefficients $A$ and $B$, as all the other quantities in the equation, are the same that in the orthographic case (see Chapter 1 on page 6), except for the light source here indicated with the letter $\mathbf{L}$.

## 6.4. Perspective Oren-Nayar SfS in Spherical Coordinates

In order to derive the nonlinear HJEs that describe our new model, we have to formulate the brightness equation of the Oren-Nayar reflectance model in (6.7) using the parametrization in terms of spherical coordinates that we derived in Section 6.2. As a first step we have to determine the direction $\mathbf{L}$ of the light source and the viewing

direction $\mathbf{V}$ of the camera. Following Fig. 6.2, these two directions can be computed as

$$\mathbf{L} = -\mathbf{e}_r \qquad \text{and} \qquad \mathbf{V} = \overrightarrow{SC} = \overrightarrow{LC} - \overrightarrow{LS} = (v_1 - r)\,\mathbf{e}_r + v_2\,\mathbf{e}_\varphi + v_3\,\mathbf{e}_\theta\,, \quad (6.8)$$

where

$$v_1 = \sqrt{c_1^2 + c_2^2 + c_3^2}\,, \;\; v_2 = \arccos\left(-c_3/\sqrt{c_1^2 + c_2^2 + c_3^2}\right)\,, \;\; v_3 = \arctan\left(c_2/c_1\right)\,. \quad (6.9)$$

Knowing the surface normal $\mathbf{n}$ from Eq. (6.6) as well as the light direction $\mathbf{L}$ and the viewing direction $\mathbf{V}$, we can then reformulate all trigonometric expressions occurring in the Oren-Nayar brightness equation (6.7) in terms of spherical coordinates. By the relation

$$\nabla r := \nabla_{(\theta,\varphi)}r = \frac{1}{r}\left(\frac{\partial r}{\partial \varphi}\right)\mathbf{e}_\varphi + \frac{1}{r\sin\varphi}\left(\frac{\partial r}{\partial \theta}\right)\mathbf{e}_\theta\,, \quad (6.10)$$

we get

$$\cos(\theta_i) = \mathbf{N}\cdot\mathbf{L} = \frac{\mathbf{n}}{|\mathbf{n}|}\cdot\mathbf{L} = \left(|\nabla r|^2 + 1\right)^{-1/2}\,, \quad (6.11)$$

$$\cos(\theta_r) = \mathbf{N}\cdot\mathbf{V} = \left(\frac{1}{r\sin\varphi}\frac{\partial r}{\partial \theta}v_3 + \frac{1}{r}\frac{\partial r}{\partial \varphi}v_2 + r - v_1\right)\left(|\nabla r|^2 + 1\right)^{-1/2}\,, \quad (6.12)$$

with $\sin(\theta_i) = \sqrt{1 - \cos^2(\theta_i)}$ and $\sin(\theta_r) = \sqrt{1 - \cos^2(\theta_r)}$ which, in particular, allows to rewrite $\sin(\alpha)$ and $\tan(\beta)$. What still remains to be computed is the expression $\cos(\varphi_i - \varphi_r)$. To this end, we calculate the projections of the directions $\mathbf{L}$ and $\mathbf{V}$ onto the $(x_1, x_2)$-plane. This is realized setting $\varphi = \frac{\pi}{2}$ and, consequently, by reducing $v_1$ and $r$ defined in (6.9) and (6.4) to their first two components. Denoting these projections of $\mathbf{L}$ and $\mathbf{V}$ by $\hat{\mathbf{l}}$ and $\hat{\mathbf{v}}$, respectively, we get

$$\cos(\varphi_i - \varphi_r) = \hat{\mathbf{l}}\cdot\hat{\mathbf{v}} = \sqrt{r_1^2 + r_2^2} - \sqrt{c_1^2 + c_2^2}\,. \quad (6.13)$$

Now, we can rewrite Eq. (6.7) entirely in spherical coordinates. However, since there are several *min* and *max* operators involved, we have to distinguish four different cases. These cases are given by Eqs. (6.14)–(6.17). For each case we have listed the corresponding implications, the resulting brightness equation as well as the HJE that we finally have to solve. Note that Case 3 resembles the single Lambertian PDE in [45], whereas the HJEs for the other cases are significantly more complex. This makes explicit that our general SfS approach with Oren-Nayar reflectance model is substantially more challenging than previous approaches based on the Lambertian assumption. Moreover, Case 4 is actually a special case of Case 1. Since this case exactly describes the standard setting, where the light source is located at the optical center of the camera, we decided to list it separately. Finally, we would like to mention that the angle between the light direction $\mathbf{L}$ and the viewing direction $\mathbf{V}$ cannot be larger or equal to $\pi$. This is evident, since

otherwise we cannot project these vectors onto the $(x_1, x_2)$-plane.

**Case 1:** $\theta_i \geq \theta_r$ and $(\varphi_i - \varphi_r) \in [0, \frac{\pi}{2}) \cup (\frac{3}{2}\pi, 2\pi]$

1. Implication: $\quad \max[0, \cos(\varphi_i - \varphi_r)] = \cos(\varphi_i - \varphi_r)$ \hfill (6.14)

2. Brightness equation: $\quad I(\mathbf{x}) = \dfrac{1}{r^2} \cos(\theta_i) \left( A + B \cos(\varphi_i - \varphi_r) \sin(\theta_i) \dfrac{\sin(\theta_r)}{\cos(\theta_r)} \right)$

3. Hamilton-Jacobi equation:

$$r^2 I - \frac{A}{\sqrt{|\nabla r|^2 + 1}} - \frac{B(\hat{\mathbf{l}} \cdot \hat{\mathbf{v}})|\nabla r|}{|\nabla r|^2 + 1} \frac{\sqrt{|\nabla r|^2 + 1 - \left( \frac{1}{r \sin \varphi} \frac{\partial r}{\partial \theta} v_3 + \frac{1}{r} \frac{\partial r}{\partial \varphi} v_2 + r - v_1 \right)^2}}{\left( \frac{1}{r \sin \varphi} \frac{\partial r}{\partial \theta} v_3 + \frac{1}{r} \frac{\partial r}{\partial \varphi} v_2 + r - v_1 \right)} = 0$$

**Case 2:** $\theta_i < \theta_r$ and $(\varphi_i - \varphi_r) \in [0, \frac{\pi}{2}) \cup (\frac{3}{2}\pi, 2\pi]$

1. Implication: $\quad \max[0, \cos(\varphi_i - \varphi_r)] = \cos(\varphi_i - \varphi_r)$ \hfill (6.15)

2. Brightness equation: $\quad I(\mathbf{x}) = \dfrac{1}{r^2} \cos(\theta_i) \left( A + B \cos(\varphi_i - \varphi_r) \sin(\theta_r) \dfrac{\sin(\theta_i)}{\cos(\theta_i)} \right)$

3. Hamilton-Jacobi equation:

$$r^2 I - \frac{A}{\sqrt{|\nabla r|^2 + 1}} - \frac{B(\hat{\mathbf{l}} \cdot \hat{\mathbf{v}})|\nabla r|}{|\nabla r|^2 + 1} \sqrt{|\nabla r|^2 + 1 - \left( \frac{1}{r \sin \varphi} \frac{\partial r}{\partial \theta} v_3 + \frac{1}{r} \frac{\partial r}{\partial \varphi} v_2 + r - v_1 \right)^2} = 0$$

**Case 3:** For any $\theta_i, \theta_r$, and $(\varphi_i - \varphi_r) \in [\frac{\pi}{2}, \frac{3}{2}\pi]$

1. Implication: $\quad \max[0, \cos(\varphi_i - \varphi_r)] = 0$ \hfill (6.16)

2. Brightness equation: $\quad I(\mathbf{x}) = \dfrac{1}{r^2} A \cos(\theta_i)$

3. Hamilton-Jacobi equation: $\quad I \sqrt{|\nabla r|^2 + 1} - \dfrac{A}{r^2} = 0$

**Case 4:** $\theta_i = \theta_r$ and $\varphi_i = \varphi_r$

1. Implications: $\quad \theta := \theta_i = \theta_r = \alpha = \beta$ and $\max[0, \cos(\varphi_i - \varphi_r)] = 1$ \hfill (6.17)

2. Brightness equation: $\quad I(\mathbf{x}) = \dfrac{1}{r^2} \cos(\theta) \left( A + B \dfrac{\sin(\theta^2)}{\cos(\theta)} \right)$

3. Hamilton-Jacobi equation: $\quad r^2 I(|\nabla r|^2 + 1) - A\sqrt{|\nabla r|^2 + 1} - B|\nabla r|^2 = 0$

## 6.5. Fast Marching Solver: a variant for spherical coordinates

After we have derived the brightness equation for the Oren-Nayar reflectance model with general position of the light source, let us now discuss how we can solve the resulting set of HJEs for the unknown radial distance $r$. In order to allow for an efficient computation, we rely on a variant of the *Fast Marching* (FM) schemes which are among the fastest solvers for PDEs available in the literature [102]. Such schemes start from critical points, i.e. points that are local minima of the depth, and then propagate the solution to the remaining points on the surface. At each grid point, the corresponding nonlinear PDE from Eqs. (6.14)–(6.17) has to be solved iteratively for the unknown radial distance. Since our HJEs are formulated in a spherical coordinate system that is usually not centered at the camera position, the location where the radiance data $I$ has to be evaluated depends on $r$. This, however, prevents the application of standard FM schemes, since they assume $I$ not to change during the iterations.

Recently, Galliani *et al.* [45] proposed a specifically adapted variant of FM for spherical coordinates. Although their SfS model is restricted to the Lambertian case and the corresponding single HJE is significantly simpler than our set of HJEs, we can still make use of the basic strategy of their approach:

(i) In a first step, we identify critical points on the surface based on their brightness values. Since such points denote local minima with respect to their distance to the light source, we know that $\nabla r = 0$. This allows us to solve the set of HJEs at those locations *locally* for the radial depth $r$, i.e. *without* considering information from neighbouring pixels. Solving for $r$ can either be done iteratively (Case 1 and 2) or directly (Case 3 and 4).

(ii) After computing the depth values at critical points, we propagate the information to other points in terms of $\theta$ and $\varphi$ by sequentially updating the depth values at neighboring locations via solving the corresponding HJE there. As proposed in [45], we apply the iterative update strategy from [118] making use of the *regula falsi* method that has been originally employed in the context of nonlinear HJEs in Euclidean coordinates. This requires to discretize our HJEs for which we use a standard upwind scheme as described in [97]. Moreover, since $x_1$ and $x_2$ depend on the radial depth $r$, we have to update the sample location and thus the value of $I(x_1(\theta, \varphi, r), x_2(\theta, \varphi, r))$ at each iteration within our iterative framework. The sampling at subpixel locations is done using bilinear interpolation. The iterations are stopped if the residual of the equation drops below a certain threshold $T$. In our case, we use $T = 10^{-3}$.

## 6.6. Experimental Evaluation

In order to evaluate the quality and robustness of our model, we have used endoscopic images provided in [1]. Since model parameters in SfS are in general unknown, experiments with such real-world data are highly challenging (see *e.g.* [95, 109]). In particular, the position of the camera relative to the position of the light source is typically not provided with such data sets, since standard SfS models are restricted to the assump-

tion that both positions coincide (which is often not true by construction). Since we consider position of the light source *not* located at the optical center of the camera, for our experiments we had to rely on a rough estimation of the correct position by visually inspecting the images. Hence, we assumed that the light source is located at the origin based on Fig. 6.2 and chose the position of the camera nearby but *not too close* to the origin. Since it was already shown in [45] that a model with flexible position of the light source can give significant advantages over a standard SfS approach, we focus in the following on the other two important aspects of our novel method: the visual quality of the reconstruction as well as its robustness with respect to parameter variations.

**Test 1**

In our first experiment, we evaluated the impact of the spherical resolution on the quality of the reconstruction. To this end, we used uniform grids in the $\varphi$-$\theta$-domain with mesh widths $\delta_{\varphi\theta} = \Delta\varphi = \Delta\theta$ and computed the results for different resolutions. As we can see from Fig. 6.3, the reconstruction quality improves significantly when increasing the resolution, i.e. when refining the grid. However, one should keep in mind that the actual quality is limited by the resolution of the input image which is repeatedly evaluated at sub-pixel positions during our FM computation. Moreover, one should note that, due to choice that the light source is at the origin of our coordinate system, the reconstruction is computed from the viewpoint of the light source. This explains the slight shift of the reconstruction with respect to the input image.



(a) Input image $(115 \times 106)$.    (b) $\delta_{\varphi\theta} = 0.025$.    (c) $\delta_{\varphi\theta} = 0.0175$.    (d) $\delta_{\varphi\theta} = 0.0125$.    (e) $\delta_{\varphi\theta} = 0.0075$.
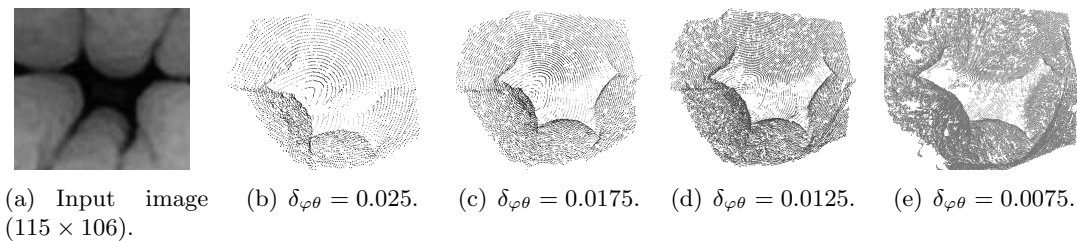
Figure 6.3.: Reconstruction of gastric antrum with $\sigma = \frac{\pi}{6}$.

**Test 2**

In a second experiment, we investigated the robustness of the reconstruction with respect to the choice of the roughness parameter $\sigma$ for the Oren-Nayar reflectance model. To this end, we have used four different input images and computed the reconstructions for different values of $\sigma$ accordingly. The Figs. 6.4–6.7 show the reconstruction of the duodenum, the esophagus, the papilla of Vater, and the stomach of lining, respectively. As we can see, the model gives very reasonable results and behaves in a well-posed manner, i.e. the results are stable and depend continuously on $\sigma$. This is very important for performing SfS computations with real-world input images, since the correct value of $\sigma$ is typically not known.

(a) Input image (211 × 208).    (b) $\sigma = \frac{\pi}{6}$.    (c) $\sigma = \frac{\pi}{4}$.    (d) $\sigma = \frac{\pi}{2}$.

(e) Input image (cropped, 106 × 78).    (f) $\sigma = \frac{\pi}{6}$.    (g) $\sigma = \frac{\pi}{4}$.    (h) $\sigma = \frac{\pi}{2}$.
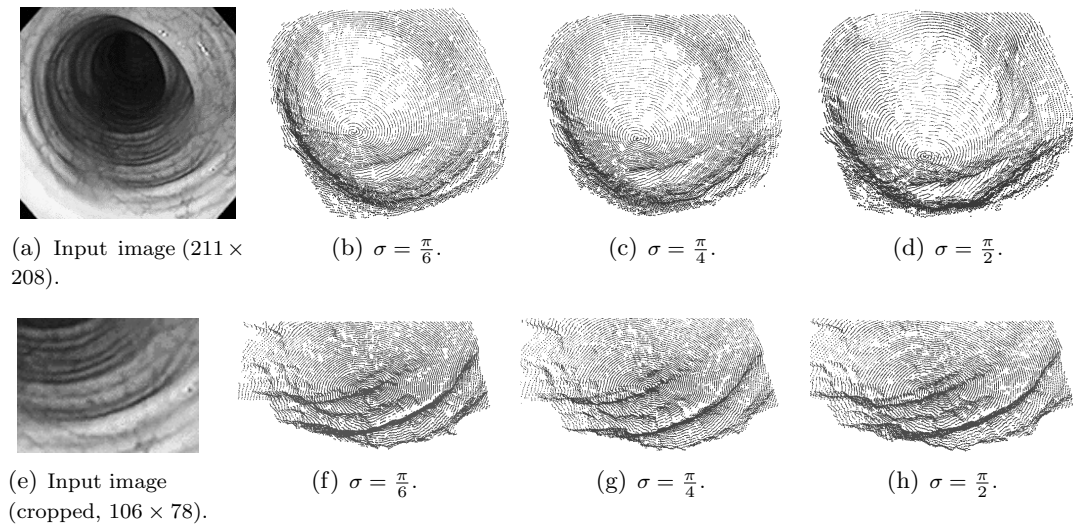
Figure 6.4.: Reconstruction of the duodenum $\delta_{\varphi\theta} = 0.0125$, grid size 504 × 504.



(a) Input image (586 × 502).    (b) $\sigma = \frac{\pi}{6}$.    (c) $\sigma = \frac{\pi}{4}$.    (d) $\sigma = \frac{\pi}{2}$.
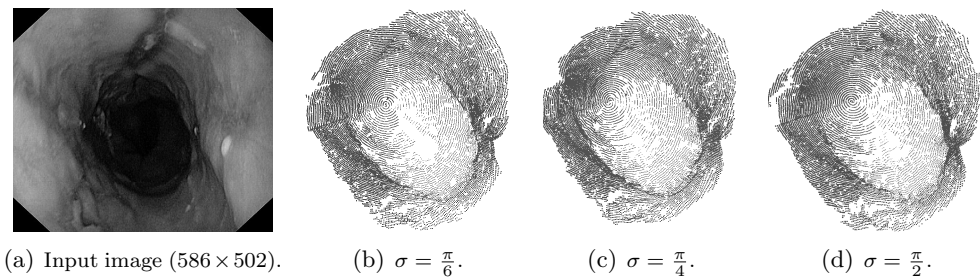
Figure 6.5.: Reconstruction of the esophagus $\delta_{\varphi\theta} = 0.0125$, grid size 504 × 504.



(a) Input image (210 × 160).    (b) $\sigma = \frac{\pi}{6}$.    (c) $\sigma = \frac{\pi}{4}$.    (d) $\sigma = \frac{\pi}{2}$.

Figure 6.6.: Reconstruction of the papilla of Vater $\delta_{\varphi\theta} = 0.0125$, grid size 504 × 504.

## 6.7. Conclusions

We have presented a perspective SfS model that combines the advantage of freely selecting the position of the light source with the robustness of an advanced non-Lambertian

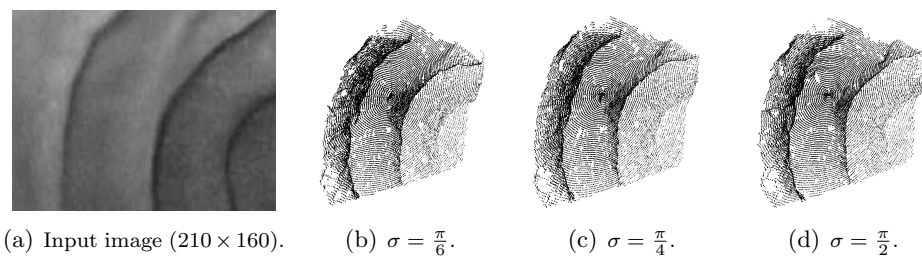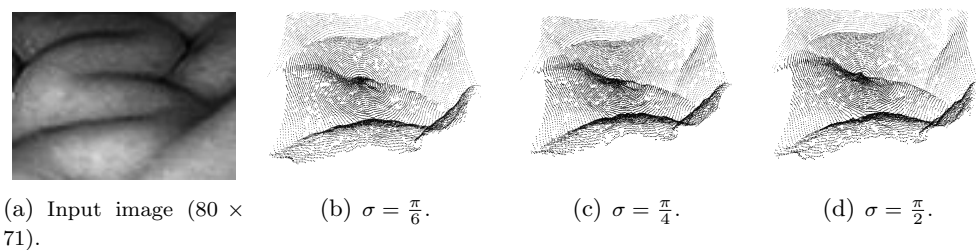(a) Input image (80 × 71).    (b) $\sigma = \frac{\pi}{6}$.    (c) $\sigma = \frac{\pi}{4}$.    (d) $\sigma = \frac{\pi}{2}$.

Figure 6.7.: Reconstruction of the stomach lining $\delta_{\varphi\theta} = 0.0125$, grid size $504 \times 504$.

reflectance model. By considering both a general light source setup and an improved reflectance model, the resulting approach is one of the most advanced and flexible models in this area. This is also validated by our experiments with real-world images coming from medical applications that gave good results. Despite these results, we found that the relatively large number of unknown parameters still makes conducting experiments difficult. There is a gap between the model and the real world: parameters as the camera position, the light position and the roughness have to be roughly estimated since they are required in the formulation of the model as data. Surely, at the end results are more accurate with respect to the Lambertian model, but after many tests carried out by setting manually the various parameters involved. This is the price to pay for taking into account an advanced reflectance model that leads to a set of relatively complex PDEs. Nevertheless, we believe that the combination of advanced models for light source setup and reflectance is necessary to obtain good results. The reflectance model should not be more complicated than the one of Oren and Nayar otherwise a carefully controlled experimental environment is needed to achieve meaningful results (see *e.g.* [95]).

# 7. Conclusions and perspectives

In this thesis we focused our attention on the SfS problem, most under orthographic projection but also under perspective projection. We considered non-Lambertian reflectance models, i.e. models that take into account the position of the observer or physical characteristics of the material of the object to be reconstructed, an extension of the classical Lambertian model, which reflects the light uniformly over the entire surface supposed smooth, without taking into account the physical properties of the material or other points of view.

## 7.1. Orthographic projection

With respect to the orthographic projection, we derived nonlinear partial differential equations of first order, i.e. Hamilton-Jacobi equations (HJEs), associated to the non-Lambertian reflectance models ON–model, PH–model and BP–model. We derived the HJEs in all the possible cases for each models, coupling vertical or oblique light source with vertical or oblique position of the observer, in order to give a complete description of these models, that is not present at the moment in the literature. We have proposed and analyzed a general and unique mathematical formulation of these models. In this general formulation we can switch on and off the different terms related to ambient, diffuse and specular reflection in a very simple way. This elegant general model is very flexible to treat the various situations with vertical and oblique light sources.

Moreover, we try to overcome the typical concave/convex ambiguity known for the classical Lambertian model via these non-Lambertian models, but our numerical simulations carried out on synthetic and real images showed that they all fail on this issue. But our work is able to respond to the need to overcome the Lambertian model that is not suitable to deal with realistic images coming from medical or security application.

We studied several numerical techniques to solve the HJEs rewritten in a fixed point form: the semi-Lagrangian scheme and acceleration methods based on a finite difference approximation, as the Fast Marching and the Fast Sweeping ones. We have obtained a general convergence result (Theorem 2.3.1 illustrated in Chapter 2).

From the numerical point of view, our numerical tests showed that the schemes are consistent and we obtained good results for synthetic and real input images. With a vertical light source, the CPU time of the non-Lambertian models is of the same order of the CPU time employed by the Lambertian model but setting the parameters in a right way the performances are always better. Regarding the oblique light source case, the CPU time necessary for the non-Lambertian models is greater with respect to the Lambertian model. This happens since the Lambertian model is easier to use being devoid of parameters to be set. Moreover, the fixed point scheme corresponding to the

Lambertian model converges using a smaller tolerance $\eta$ for the same reason of before. Choosing a model that is the best of all is not simple, because the two non-Lambertian models analyzed are both sensitive to the choice of the parameters involved in the model and to the choice of the light source. Note that the roughness parameter $\sigma$ for the Oren-Nayar model or the exponent $\alpha$ and the coefficient $k_S$ for the Phong model are known quantities for the model, but they are not provided in real situations.

### Orthographic projection: open problems

As already said, the new models do not solve the classical concave/convex ambiguity of the SfS problem based on a single image. This can be done only adding additional informations on the image or dealing with more than one image as in the photometric stereo SfS problem. In this respect, we are going to develop an interesting model applying the Photometric Stereo technique. This model will consider the Lambertian model as diffuse component and the Blinn-Phong model for the specular one. We will try first to prove that the only specular part is well-posed and there exists a unique solution, as already done for the Lambertian component [76]. After that, we will try to find a way to combine the two components in order to work with both.

Another interesting improvement is a new model that has the diffuse component represented by the more general and complex Oren-Nayar model, and the specular component represented by the model introduced by Phong or the one proposed by Blinn-Phong. Then, verify whether such a model can be written in the same form of the models analyzed here, namely, through a fixed point form; check if the discrete operator of the corresponding semi-Lagrangian scheme respects the properties of the general convergence Theorem 2.3.1 illustrated in Chapter 2 of this thesis, and if this new model gives better results than those obtained so far in the simulations, especially provided on real images. A possible improvement can also be to resort to the use of high order approximation schemes.

## 7.2. Perspective projection

Regarding the perspective case, we described a new general model that combines the advantage of freely selecting the position of the light source with an advanced non-Lambertian reflectance model, i.e. the Oren-Nayar model. We derived the HJEs in spherical coordinates for all the positions of the light source and of the camera. By considering both a general light source setup and an improved reflectance model, the resulting approach is one of the most advanced and flexible models in the field, although there are no comparisons of the performances in terms of some errors estimations with other models since they are not present in the literature. From the numerical point of view, our experiments on medical real-worlds images gave good results. Despite these results, some remarks on the difficulties encountered should be made: there is a gap between the model and the real world. Parameters as the camera position, the light position and the roughness have to be roughly estimated since they are required in the formulation of the

model as data. Surely, at the end results are more accurate with respect to the Lambertian model, but after many tests carried out by setting manually the various parameters involved. This is the price one has to pay for taking into account an advanced reflectance model that leads to a set of relatively complex PDEs. If considering models of reflection increasingly complex is completely convenient in terms of time and accuracy, hence if it is worth to deal with them, is a question still open.

### Perspective projection: open problems

An interesting future direction could be to realize a new general model in spherical coordinates that has the same advantage of locating the position of the light source everywhere in the scene. This model will be based on the Phong or Blinn-Phong model instead of the Oren-Nayar model. Once you have created such a model, observe if it is more easy to handle and the results are more accurate than the model based on the Oren-Nayar reflectance model.

Another improvement could be applying the Photometric Stereo technique to these general non-Lambertian models for the resolution of the Perspective SfS problem in cartesian coordinates by using the usual assumption of the light source located at the optical center of the camera.

# A. Appendix: Supplementary Material for Chapter 6

In what follows, we present intermediate steps that are helpful to understand the derivation of the *normal vector* and the *four cases of the Hamilton-Jacobi equations* that correspond to the generalized perspective SfS model with Oren-Nayar reflectance, described in Chapter 6.

## A.1. Surface Parametrization and Normal Vector

Starting our derivation in Cartesian coordinates, we can note from Fig. 6.2 on page 99 in Chapter 6 that

$$\overrightarrow{CX} = \overrightarrow{LX} - \overrightarrow{LC} = \begin{bmatrix} x_1 \\ x_2 \\ -(c_3 + \mathtt{f}) \end{bmatrix} - \begin{bmatrix} c_1 \\ c_2 \\ -c_3 \end{bmatrix} = \begin{bmatrix} x_1 - c_1 \\ x_2 - c_2 \\ -\mathtt{f} \end{bmatrix}. \tag{A.1}$$

This leads to

$$
\begin{aligned}
\overrightarrow{LS} &= \overrightarrow{LC} + \overrightarrow{CS} = \overrightarrow{LC} + \lambda \overrightarrow{CX} \\
&= \begin{bmatrix} c_1 \\ c_2 \\ -c_3 \end{bmatrix} + \lambda \begin{bmatrix} x_1 - c_1 \\ x_2 - c_2 \\ -\mathtt{f} \end{bmatrix} \\
&= \begin{bmatrix} \lambda x_1 + (1 - \lambda)c_1 \\ \lambda x_2 + (1 - \lambda)c_2 \\ -(c_3 + \lambda \mathtt{f}) \end{bmatrix} =: \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix},
\end{aligned}
\tag{A.2}
$$

where $\overrightarrow{AB}$ stands for a vector notation with a starting point $A$ and an endpoint $B$. In spherical coordinates, we can describe (A.2) as

$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} =: \mathbf{r} := r\,\mathbf{e}_r \tag{A.3}$$

with

$$r = \left(r_1^2 + r_2^2 + r_3^2\right)^{\frac{1}{2}}, \tag{A.4}$$
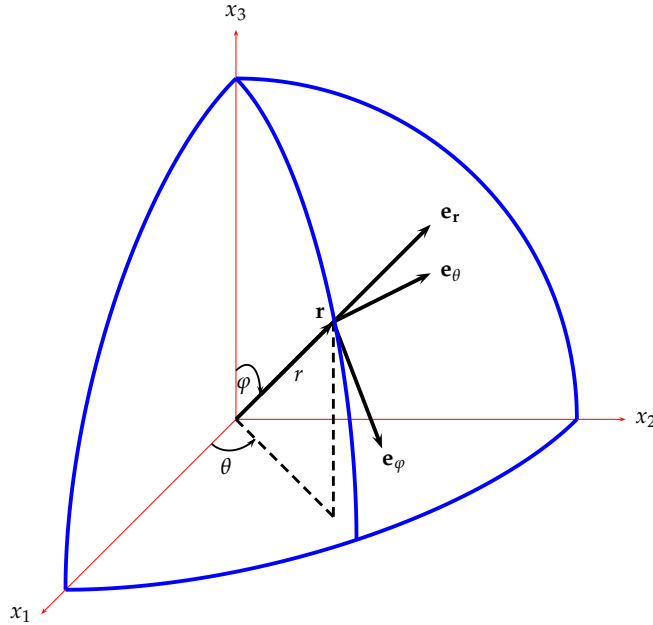
see Fig. A.1 for details.



Figure A.1.: Illustration of parameters in spherical coordinates. $\mathbf{r} = (r_1, r_2, r_3)$ represents a vector on the sphere and $r$ denotes the magnitude of $\mathbf{r}$ given by (A.4).

In order to obtain $\lambda$, we have to solve the following quadratic equation:

$$r^2 = r_1^2 + r_2^2 + r_3^2 \overset{(A.2)}{=} [c_1 + \lambda(x_1 - c_1)]^2 + [c_2 + \lambda(x_2 - c_2)]^2 + (c_3 + \lambda\mathtt{f})^2$$
$$= \left(c_1^2 + c_2^2 + c_3^2\right) + 2\left[c_1(x_1 - c_1) + c_2(x_2 - c_2) + c_3\,\mathtt{f}\right]\lambda + \left[(x_1 - c_1)^2 + (x_2 - c_2)^2 + \mathtt{f}^2\right]\lambda^2 .$$
$$(A.5)$$

Since this requires the radial depth $r$ to be known, we switch from Cartesian coordinates to spherical coordinates and compute $r$ from the input image directly. This can be done via the brightness equation of the Oren-Nayar model that describes the relation between the brightness values of the input image $I$ and the normal $\mathbf{n}$ of the corresponding point on the object surface. However, before we can make use of this brightness equation, we have to express it in spherical coordinates. In particular, this requires to calculate the surface normal $\mathbf{n}$ in terms of the radial depth $r$. Starting from (A.3) and using the

spherical basis described in Section 6.2 on page 98, we obtain

$$
\begin{aligned}
\mathbf{n} &= \frac{\partial\,(r\mathbf{e}_r)}{\partial\theta} \times \frac{\partial\,(r\mathbf{e}_r)}{\partial\varphi} \\[2mm]
&= \left(\frac{\partial\,r}{\partial\theta}\,\mathbf{e}_r + r\,\frac{\partial\,\mathbf{e}_r}{\partial\theta}\right) \times \left(\frac{\partial\,r}{\partial\varphi}\,\mathbf{e}_r + r\,\frac{\partial\,\mathbf{e}_r}{\partial\varphi}\right) \\[2mm]
&= \left(\frac{\partial\,r}{\partial\theta}\,\mathbf{e}_r \times \frac{\partial\,r}{\partial\varphi}\,\mathbf{e}_r\right) + \left(\frac{\partial\,r}{\partial\theta}\,\mathbf{e}_r \times r\,\frac{\partial\,\mathbf{e}_r}{\partial\varphi}\right) + \left(r\,\frac{\partial\,\mathbf{e}_r}{\partial\theta} \times \frac{\partial\,r}{\partial\varphi}\,\mathbf{e}_r\right) + \left(r\,\frac{\partial\,\mathbf{e}_r}{\partial\theta} \times r\,\frac{\partial\,\mathbf{e}_r}{\partial\varphi}\right) \\[2mm]
&= \frac{\partial\,r}{\partial\theta}\,\frac{\partial\,r}{\partial\varphi}\,\underbrace{(\mathbf{e}_r \times \mathbf{e}_r)}_{=0} + r\,\frac{\partial\,r}{\partial\theta}\left(\mathbf{e}_r \times \frac{\partial\,\mathbf{e}_r}{\partial\varphi}\right) + r\,\frac{\partial\,r}{\partial\varphi}\left(\frac{\partial\,\mathbf{e}_r}{\partial\theta} \times \mathbf{e}_r\right) + r^2\left(\frac{\partial\,\mathbf{e}_r}{\partial\theta} \times \frac{\partial\,\mathbf{e}_r}{\partial\varphi}\right) \\[2mm]
&= r\,\frac{\partial\,r}{\partial\theta}\left(\mathbf{e}_r \times \frac{\partial\,\mathbf{e}_r}{\partial\varphi}\right) + r\,\frac{\partial\,r}{\partial\varphi}\left(\frac{\partial\,\mathbf{e}_r}{\partial\theta} \times \mathbf{e}_r\right) + r^2\left(\frac{\partial\,\mathbf{e}_r}{\partial\theta} \times \frac{\partial\,\mathbf{e}_r}{\partial\varphi}\right) \\[2mm]
&\overset{(a)}{=} r\,\frac{\partial\,r}{\partial\theta}\,(\mathbf{e}_r \times \mathbf{e}_\varphi) + r\,\frac{\partial\,r}{\partial\varphi}\,(\sin\varphi\,\mathbf{e}_\theta \times \mathbf{e}_r) + r^2\,(\sin\varphi\,\mathbf{e}_\theta \times \mathbf{e}_\varphi) \\[2mm]
&\overset{(b)}{=} r\,\frac{\partial\,r}{\partial\theta}\,\mathbf{e}_\theta + r\,\sin\varphi\,\frac{\partial\,r}{\partial\varphi}\,\mathbf{e}_\varphi - r^2\,\sin\varphi\,\mathbf{e}_r\,.
\end{aligned}
$$

$$(A.6)$$

The intermediate step (a) in (A.6) can be explained by noting that

$$
\frac{\partial\,\mathbf{e}_r}{\partial\varphi} = \begin{bmatrix} \cos\varphi\,\cos\theta \\ \cos\varphi\,\sin\theta \\ -\sin\varphi \end{bmatrix} = \mathbf{e}_\varphi
\tag{A.7}
$$

and

$$
\frac{\partial\,\mathbf{e}_r}{\partial\theta} = \begin{bmatrix} -\sin\varphi\,\sin\theta \\ \sin\varphi\,\cos\theta \\ 0 \end{bmatrix} = \sin\varphi\,\mathbf{e}_\theta\,.
\tag{A.8}
$$

In addition, we know that

$$
\begin{cases}
\mathbf{e}_r &\times & \mathbf{e}_\varphi &= & \mathbf{e}_\theta \\
\mathbf{e}_\varphi &\times & \mathbf{e}_\theta &= & \mathbf{e}_r \\
\mathbf{e}_\theta &\times & \mathbf{e}_r &= & \mathbf{e}_\varphi
\end{cases}
\tag{A.9}
$$

since $(\mathbf{e}_\varphi, \mathbf{e}_\theta, \mathbf{e}_r)$ constitutes a right-handed coordinate system. This in turn explains step (b).

## A.2. Oren-Nayar Brightness Equation

Starting from Section 6.3 on page 100, we have to calculate the trigonometric quantities that appear in the Oren-Nayar brightness equation (6.7) in order to derive the Hamilton-Jacobi equations in spherical coordinates. Based on Fig. 6.2 from Chapter 6, we know

that the light source is located at the center of the coordinate system and thus the light direction is given by

$$\mathbf{L} = -\,\mathbf{e}_r \,. \tag{A.10}$$

On the other hand, the viewing direction at any surface point reads

$$
\begin{aligned}
\mathbf{V} &= \overrightarrow{SC} \\
&= \overrightarrow{LC} - \overrightarrow{LS} \\
&= (v_1\,\mathbf{e}_r + v_2\,\mathbf{e}_\varphi + v_3\,\mathbf{e}_\theta) - r\,\mathbf{e}_r \\
&= (v_1 - r)\,\mathbf{e}_r + v_2\,\mathbf{e}_\varphi + v_3\,\mathbf{e}_\theta\,,
\end{aligned}
\tag{A.11}
$$

where

$$v_1 = \sqrt{c_1^2 + c_2^2 + c_3^2}\,, \tag{A.12}$$

$$v_2 = \arccos \frac{-c_3}{\sqrt{c_1^2 + c_2^2 + c_3^2}}\,, \tag{A.13}$$

$$v_3 = \arctan \frac{c_2}{c_1}\,. \tag{A.14}$$

Knowing the surface normal $\mathbf{n}$ given by (A.6), the light direction $\mathbf{L}$ and the viewing direction $\mathbf{V}$ given by (A.10) and (A.11), respectively, and by making use of the relation

$$\nabla r := \nabla_{(\theta,\varphi)} r = \frac{1}{r}\left(\frac{\partial r}{\partial \varphi}\right)\mathbf{e}_\varphi + \frac{1}{r\sin\varphi}\left(\frac{\partial r}{\partial \theta}\right)\mathbf{e}_\theta\,, \tag{A.15}$$

we can reformulate all trigonometric expressions of the Oren-Nayar brightness equation (6.7) in terms of spherical coordinates as follows:

$$
\begin{aligned}
\cos(\theta_i) &= \mathbf{N}\cdot\mathbf{L} \\[2mm]
&= \frac{\mathbf{n}}{|\mathbf{n}|}\cdot\mathbf{L} \\[2mm]
&\overset{\substack{(A.6)\\(A.10)}}{=} \frac{\left(r\,\frac{\partial r}{\partial \theta}\,\mathbf{e}_\theta + r\sin\varphi\,\frac{\partial r}{\partial \varphi}\,\mathbf{e}_\varphi - r^2\,\sin\varphi\,\mathbf{e}_r\right)\cdot(-\mathbf{e}_r)}{\sqrt{r^2\left[\left(\frac{\partial r}{\partial \theta}\right)^2 + \sin^2\varphi\left(\frac{\partial r}{\partial \varphi}\right)^2 + r^2\,\sin^2\varphi\right]}} \\[2mm]
&= \frac{r^2\,\sin\varphi}{r\sqrt{\left(\frac{\partial r}{\partial \theta}\right)^2 + \sin^2\varphi\left(\frac{\partial r}{\partial \varphi}\right)^2 + r^2\,\sin^2\varphi}} \\[2mm]
&= \frac{1}{\sqrt{|\nabla r|^2 + 1}}\,,
\end{aligned}
\tag{A.16}
$$

$$
\begin{aligned}
\cos(\theta_r) \quad &= \quad \mathbf{N} \cdot \mathbf{V} \\[2mm]
&= \quad \frac{\mathbf{n}}{|\mathbf{n}|} \cdot \mathbf{V} \\[4mm]
&\overset{\substack{(A.6)\\(A.11)}}{=} \quad \frac{r\frac{\partial r}{\partial \theta}\,v_3 + r\sin\varphi\frac{\partial r}{\partial \varphi}\,v_2 + (v_1 - r)(-r^2\sin\varphi)}{r^2\sin\varphi\,\sqrt{|\nabla r|^2 + 1}} \\[4mm]
&= \quad \frac{\frac{\partial r}{\partial \theta}\,v_3 + \sin\varphi\frac{\partial r}{\partial \varphi}\,v_2 + r^2\sin\varphi - v_1 r\sin\varphi}{r\sin\varphi\sqrt{|\nabla r|^2 + 1}} \\[4mm]
&= \quad \frac{\frac{1}{r\sin\varphi}\frac{\partial r}{\partial \theta}\,v_3 + \frac{1}{r}\frac{\partial r}{\partial \varphi}\,v_2 + r - v_1}{\sqrt{|\nabla r|^2 + 1}}\,,
\end{aligned}
\tag{A.17}
$$

$$
\begin{aligned}
\sin(\theta_i) \quad &= \quad \sqrt{1 - (\mathbf{N} \cdot \mathbf{L})^2} \\[4mm]
&\overset{(A.16)}{=} \quad \sqrt{1 - \left(\frac{1}{\sqrt{|\nabla r|^2 + 1}}\right)^2} \\[4mm]
&= \quad \sqrt{1 - \frac{1}{|\nabla r|^2 + 1}} = \frac{|\nabla r|}{\sqrt{|\nabla r|^2 + 1}}\,,
\end{aligned}
\tag{A.18}
$$

$$
\begin{aligned}
\sin(\theta_r) \quad &= \quad \sqrt{1 - (\mathbf{N} \cdot \mathbf{V})^2} \\[4mm]
&\overset{(A.17)}{=} \quad \sqrt{1 - \left(\frac{\frac{1}{r\sin\varphi}\frac{\partial r}{\partial \theta}\,v_3 + \frac{1}{r}\frac{\partial r}{\partial \varphi}\,v_2 + r - v_1}{\sqrt{|\nabla r|^2 + 1}}\right)^2} \\[4mm]
&= \quad \sqrt{1 - \frac{\left(\frac{1}{r\sin\varphi}\frac{\partial r}{\partial \theta}\,v_3 + \frac{1}{r}\frac{\partial r}{\partial \varphi}\,v_2 + r - v_1\right)^2}{|\nabla r|^2 + 1}}\,.
\end{aligned}
\tag{A.19}
$$

In order to project the vectors $\mathbf{L}$ and $\mathbf{V}$ onto the $(x_1, x_2)$-plane, we have to put $\varphi = \frac{\pi}{2}$ in the orthonormal basis defined in Section 6.2 on page 98, and we have to reduce $v_1$ and $r$ defined in (A.12) and (A.4) to the first two components. In this way, denoting these projections of $\mathbf{L}$ and $\mathbf{V}$ by $\hat{\mathbf{l}}$ and $\hat{\mathbf{v}}$, respectively, we can attain

$$\cos(\varphi_i - \varphi_r) = \hat{\mathbf{l}} \cdot \hat{\mathbf{v}} = \sqrt{r_1^2 + r_2^2} - \sqrt{c_1^2 + c_2^2} \,. \tag{A.20}$$

## A.3. Hamilton-Jacobi Equations in Spherical Coordinates

In order to derive the Hamilton-Jacobi equations (HJEs) corresponding to the Oren-Nayar brightness equation, we will use the formulas of the previous section. Thus, we obtain the following four cases:

**Case 1:** $\theta_i \geq \theta_r$ **and** $(\varphi_i - \varphi_r) \in [0, \frac{\pi}{2}) \cup (\frac{3}{2}\pi, 2\pi]$

In this case, we have the following implication:

$$\max[0, \cos(\varphi_i - \varphi_r)] = \cos(\varphi_i - \varphi_r) \tag{A.21}$$

and hence we attain

$$I(\mathbf{x}) = \frac{1}{r^2} \cos(\theta_i) \left( A + B \cos(\varphi_i - \varphi_r) \sin(\theta_i) \frac{\sin(\theta_r)}{\cos(\theta_r)} \right)$$

$$\Leftrightarrow \quad r^2 I = A \cos(\theta_i) + B \cos(\varphi_i - \varphi_r) \cos(\theta_i) \sin(\theta_i) \frac{\sin(\theta_r)}{\cos(\theta_r)}$$

$$\Leftrightarrow \quad r^2 I = A(\mathbf{N} \cdot \mathbf{L}) + B(\hat{\mathbf{l}} \cdot \hat{\mathbf{v}})(\mathbf{N} \cdot \mathbf{L}) \sqrt{1 - (\mathbf{N} \cdot \mathbf{L})^2} \cdot \frac{\sqrt{1 - (\mathbf{N} \cdot \mathbf{V})^2}}{(\mathbf{N} \cdot \mathbf{V})}$$

$$\Leftrightarrow \quad r^2 I = \frac{A}{\sqrt{|\nabla r|^2 + 1}} + \frac{B(\hat{\mathbf{l}} \cdot \hat{\mathbf{v}})}{\sqrt{|\nabla r|^2 + 1}} \cdot \frac{|\nabla r|}{\sqrt{|\nabla r|^2 + 1}} \cdot$$

$$\sqrt{1 - \frac{\left( \dfrac{1}{r \sin \varphi} \dfrac{\partial r}{\partial \theta} v_3 + \dfrac{1}{r} \dfrac{\partial r}{\partial \varphi} v_2 + r - v_1 \right)^2}{|\nabla r|^2 + 1}} \cdot \frac{\sqrt{|\nabla r|^2 + 1}}{\left( \dfrac{1}{r \sin \varphi} \dfrac{\partial r}{\partial \theta} v_3 + \dfrac{1}{r} \dfrac{\partial r}{\partial \varphi} v_2 + r - v_1 \right)}$$

$$\Leftrightarrow \quad r^2 I - \frac{A}{\sqrt{|\nabla r|^2 + 1}} - \frac{B(\hat{\mathbf{l}} \cdot \hat{\mathbf{v}})|\nabla r|}{|\nabla r|^2 + 1} \cdot \frac{\sqrt{|\nabla r|^2 + 1 - \left( \frac{1}{r \sin \varphi} \frac{\partial r}{\partial \theta} v_3 + \frac{1}{r} \frac{\partial r}{\partial \varphi} v_2 + r - v_1 \right)^2}}{\left( \dfrac{1}{r \sin \varphi} \dfrac{\partial r}{\partial \theta} v_3 + \dfrac{1}{r} \dfrac{\partial r}{\partial \varphi} v_2 + r - v_1 \right)} = 0 \,.$$

$$\tag{A.22}$$

**Case 2:** $\theta_i < \theta_r$ **and** $(\varphi_i - \varphi_r) \in [0, \frac{\pi}{2}) \cup (\frac{3}{2}\pi, 2\pi]$

Since (A.21) is still true, we obtain

$$I(\mathbf{x}) = \frac{1}{r^2}\cos(\theta_i)\left(A + B\cos(\varphi_i - \varphi_r)\sin(\theta_r)\frac{\sin(\theta_i)}{\cos(\theta_i)}\right)$$

$$\Leftrightarrow \quad r^2 I = A\cos(\theta_i) + B\cos(\varphi_i - \varphi_r)\sin(\theta_i)\sin(\theta_r)$$

$$\Leftrightarrow \quad r^2 I = A(\mathbf{N}\cdot\mathbf{L}) + B(\hat{\mathbf{l}}\cdot\hat{\mathbf{v}})\sqrt{1 - (\mathbf{N}\cdot\mathbf{L})^2}\sqrt{1 - (\mathbf{N}\cdot\mathbf{V})^2}$$

$$\Leftrightarrow \quad r^2 I - \frac{A}{\sqrt{|\nabla r|^2 + 1}} - \frac{B(\hat{\mathbf{l}}\cdot\hat{\mathbf{v}})|\nabla r|}{\sqrt{|\nabla r|^2 + 1}}\sqrt{1 - \frac{\left(\frac{1}{r\sin\varphi}\frac{\partial r}{\partial\theta}v_3 + \frac{1}{r}\frac{\partial r}{\partial\varphi}v_2 + r - v_1\right)^2}{|\nabla r|^2 + 1}} = 0$$

$$\Leftrightarrow \quad r^2 I - \frac{A}{\sqrt{|\nabla r|^2 + 1}} - \frac{B(\hat{\mathbf{l}}\cdot\hat{\mathbf{v}})|\nabla r|}{|\nabla r|^2 + 1}\sqrt{|\nabla r|^2 + 1 - \left(\frac{1}{r\sin\varphi}\frac{\partial r}{\partial\theta}v_3 + \frac{1}{r}\frac{\partial r}{\partial\varphi}v_2 + r - v_1\right)^2} = 0 \,.$$

$$(A.23)$$

**Case 3: For any** $\theta_i, \theta_r$, **and** $(\varphi_i - \varphi_r) \in [\frac{\pi}{2}, \frac{3}{2}\pi]$

In this case, we have the trivial implication:

$$\max[0, \cos(\varphi_i - \varphi_r)] = 0 \,. \tag{A.24}$$

Because of (A.24), we end up with the following HJE:

$$I(\mathbf{x}) = \frac{1}{r^2}A\cos\theta_i$$

$$\Leftrightarrow \quad r^2 I = A\cos\theta_i$$

$$\Leftrightarrow \quad r^2 I = A(\mathbf{N}\cdot\mathbf{L})$$

$$\Leftrightarrow \quad r^2 I = \frac{A}{\sqrt{|\nabla r|^2 + 1}} \tag{A.25}$$

$$\Leftrightarrow \quad r^2 I\sqrt{|\nabla r|^2 + 1} - A = 0$$

$$\Leftrightarrow \quad I\sqrt{|\nabla r|^2 + 1} - \frac{A}{r^2} = 0 \,.$$

**Case 4:** $\theta_i = \theta_r$ **and** $\varphi_i = \varphi_r$

In this particular case, we have a different implication: by defining $\theta := \theta_i = \theta_r = \alpha = \beta$, we have

$$\varphi_i = \varphi_r \Rightarrow \max[0, \cos(\varphi_i - \varphi_r)] = \max[0, \cos(0)] = 1 \,. \tag{A.26}$$

So, we obtain the HJE

$$I(\mathbf{x}) = \frac{1}{r^2} \cos(\theta) \left( A + B\frac{\sin(\theta^2)}{\cos(\theta)} \right)$$

$$\Leftrightarrow \quad r^2 I = A \cos(\theta) + B \sin(\theta^2)$$

$$\Leftrightarrow \quad r^2 I = \frac{A}{\sqrt{|\nabla r|^2 + 1}} + \frac{B |\nabla r|^2}{|\nabla r|^2 + 1} \qquad\qquad (A.27)$$

$$\Leftrightarrow \quad r^2 I \left( |\nabla r|^2 + 1 \right) - A\sqrt{|\nabla r|^2 + 1} - B |\nabla r|^2 = 0 \;.$$

# List of Algorithms

# Bibliography

[1] Gastrolab. http://www.gastrolab.net.

[2] A. H. Ahmed and A. A. Farag. Shape from Shading under various imaging conditions. In *IEEE International Conference on Computer Vision and Pattern Recognition CVPR'07*, pages X1–X8, Minneapolis, MN, June 2007.

[3] A.H. Ahmed and A.A. Farag. A new formulation for Shape from Shading for non-Lambertian surfaces. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1817–1824, 2006.

[4] S. Bakshi and Y.-H. Yang. Shape-from-Shading for non-Lambertian surfaces. In *Proc. IEEE International Conference on Image Processing (ICIP)*, pages 130–134, 1994.

[5] M. Bardi and I. Capuzzo Dolcetta. *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations.* Birkhäuser, 1997.

[6] M. Bardi and M. Falcone. An approximation scheme for the minimum time function. *SIAM J. Control Optim.*, 28(4):950–965, June 1990.

[7] M. Bardi and M. Falcone. Discrete approximation of the minimal time function for systems with regular optimal trajectories. In A. Bensoussan and J.L. Lions, editors, *Analysis and Optimization of Systes*, volume 144 of *Lecture Notes in Control and Information Sciences*, pages 103–112. Springer Berlin Heidelberg, 1990.

[8] G. Barles. *Solutions de viscosité des équations de Hamilton-Jacobi.* Springer Verlag, 1994.

[9] G. Barles and P.E. Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asympt. Anal.*, 4:271–283, 1991.

[10] J. F. Blinn. Models of light reflection for computer synthesized pictures. *Computer Graphics*, 11(2):192–198, 1977.

[11] M. Boué and P. Dupuis. Markov chain approximations for deterministic control problems with affine dynamics and quadratic cost in the control. *SIAM Journal on Numerical Analysis*, 36(3):667–695, 1999.

[12] Y. Brenier. Un algorithme rapide pour le calcul de transformées de legendre fenchel discretes. *C.R. Acad, Sci. Paris Ser. I Math.*, 308:587–589, 1989.

[13] M. Breuß, E. Cristiani, J.-D. Durou, M. Falcone, and O. Vogel. Perspective Shape from Shading: Ambiguity analysis and numerical approximations. *SIAM Journal on Imaging Sciences*, 5(1):311–342, 2012.

[14] M.J. Brooks, W. Chojnacki, and R. Kozera. Impossible and ambiguous shading patterns. *International Journal of Computer Vision*, 7(2):119–126, 1992.

[15] M.J. Brooks and B.K.P. Horn. Shape and source from shading. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (vol. II)*, pages 932–936, Los Angeles, CA, USA, 1985.

[16] A.M. Bruckstein. On Shape from Shading. *Computer Vision, Graphics and Image Processing*, 44(2):139–154, 1988.

[17] A. R. Bruss. The eikonal equation: Some results applicable to computer vision. *Journal of Mathematical Physics*, 23(5):890–896, 1982.

[18] F. Camilli and M. Falcone. An approximation scheme for the maximal solution of the Shape from Shading model. In *Proceedings of the IEEE International Conference on Image Processing (vol. I)*, volume 1, pages 49–52, Lausanne, Switzerland, Sep 1996.

[19] F. Camilli and L. Grüne. Numerical approximation of the maximal solutions for a class of degenerate Hamilton-Jacobi equations. *SIAM Journal on Numerical Analysis*, 38(5):1540–1560, 2000.

[20] F. Camilli and A. Siconolfi. Maximal subsolutions of a class of degenerate Hamilton-Jacobi problems. *Indiana University Mathematics Journal*, 48(3):1111–1131, 1999.

[21] A. Chambolle. A uniqueness result in the theory of stereo vision: coupling Shape from Shading and binocular information allows unambiguous depth reconstruction. *Annales de l'Istitute Henri Poincarè*, 11(1):1–16, 1994.

[22] D. Chopp. Some improvements of the fast marching method. *SIAM Journal on Scientific Computing*, 23(1):230–244, 2001.

[23] Hin-Shun Chung and Jiaya Jia. Efficient photometric stereo on glossy surfaces with wide specular lobes. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[24] A.R. Conn, K. Scheinberg, and Ph.L. Toint. Recent progress in unconstrained nonlinear optimization without derivatives. *Math. Programming*, 79:397–414, 1997.

[25] L. Corrias. Fast legendre-fenchel transform and applications to Hamilton-Jacobi equations and conservation laws. *SIAM J. Numer. Anal.*, 33:1534–1558, 1989.

[26] M. G. Crandall. Viscosity solutions of Hamilton-Jacobi equations. In David Campbell Alan Bishop and Basil Nicolaenko, editors, *Nonlinear Problems: Present and Future*, volume 61 of *North-Holland Mathematics Studies*, pages 117–125. North-Holland, 1982.

[27] M. G. Crandall, H. Ishii, and P.L. Lions. User's guide to viscosity solutions of second order partial differential equations. *Bulletin of the American Society*, 27:1–67, 1992.

[28] M.G. Crandall and P.-L. Lions. Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, 277(1):1–42, 1983.

[29] M.G. Crandall and P.-L. Lions. Two approximations of solutions of Hamilton-Jacobi equations. *Math. Comp.*, 43:1–19, 1984.

[30] E. Cristiani and M. Falcone. Fast semi-lagrangian schemes for the eikonal equation and applications. *SIAM J. Numer. Anal.*, 45(5):1979–2011, 2007.

[31] P. Daniel and J.-D. Durou. From deterministic to stochastic methods for Shape from Shading. In *Proceedings of the Fourth Asian Conference on Computer Vision*, pages 187–192, Taipei, Taiwan, 2000.

[32] P. Danielsson. Euclidean distance mapping. *Computer Graphics and Image Processing*, 14:227–248, 1980.

[33] J. Van Diggelen. A photometric investigation of the slopes and heights of the ranges of hills in the maria of the moon. *Bulletin of the Astronomical Institute of the Netherlands*, 11(423):283–290, 1951.

[34] J.-D. Durou, M. Falcone, and M. Sagona. Numerical methods for Shape from Shading: a new survey with benchmarks computer vision and image understanding. *Elsevier*, 109(1):22–43, 2008.

[35] J.-D. Durou and D. Piau. Ambiguous Shape from Shading with critical points. *Journal of Mathematical Imaging and Vision*, 12:99–108, 2000.

[36] M. Falcone. Numerical solution of Dynamic Programming equations. *In [5]*, pages 471–504, 1997.

[37] M. Falcone and R. Ferretti. Discrete time high-order schemes for viscosity solutions of Hamilton-Jacobi-Bellman equations. *Numerische Mathematik*, 67(3):315–344, 1994.

[38] M. Falcone and R. Ferretti. Semi-Lagrangian schemes for Hamilton-Jacobi equations, discrete representation formulae and godunov methods. *Journal of Computational Physics*, 175(2):559–575, 2002.

[39] M. Falcone and R. Ferretti. *Semi-Lagrangian Approximation Schemes for Linear and Hamilton-Jacobi Equations*. SIAM, 2014.

[40] M. Falcone, T. Giorgi, and P. Loreti. Level sets of viscosity solutions: some applications to fronts and rendez-vous problems. *SIAM Journal on Applied Mathematics*, 54(5):1335–1354, 1994.

[41] M. Falcone and M. Sagona. An algorithm for the global solution of the Shape from Shading model. In *Proceedings of the Ninth International Conference on Image Analysis and Processing (vol. I), vol. 1310 of Lecture Notes in Computer Science*, pages 596–603, Florence, Italy, 1997.

[42] M. Falcone, M. Sagona, and A. Seghini. A scheme for the shape–from–shading model with "black shadows". *Numerical Mathematics and Advanced Applications*, pages 503–512, 2003.

[43] R. Ferretti. Convergence of Semi-Lagrangian approximations to convex Hamilton-Jacobi equations under (very) large courant numbers. *SIAM J. Numer. Anal.*, 40(6):2240–2253, 2003.

[44] R.T. Frankot and R. Chellappa. A method for enforcing integrability in Shape from Shading algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):439–451, 1988.

[45] S. Galliani, Y.-C. Ju, M. Breuß, and A. Bruhn. Generalised perspective Shape from Shading in spherical coordinates. In *Proc. Scale Space and Variational Methods in Computer Vision (SSVM)*, pages 222–233, 2013.

[46] P.L. George and H. Borouchaki. *Delaunay Triangulation and Meshing.* Editions Hermes, 1998.

[47] A. S. Georghiades. Incorporating the Torrance and Sparrow Model of Reflectance in Uncalibrated Photometric Stereo. In *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003)*, volume 2, pages 816–823. IEEE, Oct 2003.

[48] G. Healey and T.O. Binford. Local shape from speccularity. *Artificial Intelligence Laboratory, Department of Computer Science, Stanford University*, 1986.

[49] J. Helmsen, E. Puckett, P. Colella, and M. Dorr. Two new methods for simulating photolithography development in 3d. In *Optical Microlithography IX*, volume 2726, pages 253–261. SPIE, 1996.

[50] B.K.P. Horn. *Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object From One View.* PhD thesis, Massachusetts Institute of Technology, 1970.

[51] B.K.P. Horn. Obtaining Shape from Shading information. In P.H. Winston (Ed.), editor, *The Psychology of Computer Vision*, chapter 4, pages 115–155. McGraw-Hill, 1975.

[52] B.K.P. Horn. Height and gradient from shading. *International Journal of Computer Vision*, 5(1):37–75, 1990.

[53] B.K.P. Horn and M.J. Brooks. The variational approach to Shape from Shading. *Computer Vision, Graphics and Image Processing*, 33(2):174–208, 1986.

[54] B.K.P. Horn and M.J. Brooks. *Shape from Shading (Artificial Intelligence)*. The MIT Press, 1989.

[55] Katsushi Ikeuchi and Berthold K.P. Horn. Numerical Shape from Shading and occluding boundaries. *Artificial Intelligence*, 17(1,Äì3):141 – 184, 1981.

[56] A. Ishii. A simple, direct proof of uniqueness for solutions of the Hamilton-Jacobi equations of eikonal type. *Proceedings of the American Mathematical Society*, pages 247–251, 1987.

[57] H. Ishii and M. Ramaswamy. Uniqueness results for a class of Hamilton-Jacobi equations with singular coefficients. *Communications in Partial Differential Equations*, 20:2187–2213, 1995.

[58] Y.-C. Ju, M. Breuß, A. Bruhn, and S. Galliani. Shape from Shading for rough surfaces: Analysis of the Oren-Nayar model. In *Proc. British Machine Vision Conference (BMVC)*, pages 104.1–104.11. BMVA Press, 2012.

[59] Y.-C. Ju, S. Tozza, M. Breuß, A. Bruhn, and A. Kleefeld. Generalised Perspective Shape from Shading with Oren-Nayar Reflectance. In *Proceedings of the 24th British Machine Vision Conference (BMVC 2013)*, pages 42.1–42.11, Bristol, UK, 2013. BMVA Press.

[60] J. Kain and D.N. Ostrov. Numerical Shape from Shading for discontinuous photographic images. *International Journal of Computer Vision*, 44(3):163–173, 2001.

[61] C.-Y. Kao, S. Osher, and Y.-H. Tsai. Fast sweeping methods for static hamilton–jacobi equations. *SIAM Journal on Numerical Analysis*, 42(6):2612–2632, 2005.

[62] Chiu Yen Kao, Stanley Osher, and Jianliang Qian. Lax-Friedrichs sweeping scheme for static Hamilton-Jacobi equations. *Journal of Computational Physics*, 196(1):367 – 391, 2004.

[63] S. Kim. An $\mathcal{O}(N)$ level set method for Eikonal equations. *SIAM Journal on Scientific Computing*, 22(6):2178–2193, 2001.

[64] R. Kimmel and A.M. Bruckstein. Global Shape from Shading. *Computer Vision and Image Understanding*, 62(3):360–369, 1995.

[65] R. Kimmel and A.M. Bruckstein. Tracking level sets by level sets: a method for solving the Shape from Shading problem. *Computer Vision and Image Understanding*, 62(1):47–58, 1995.

[66] R. Kimmel and J.A. Sethian. Computing geodesic paths on manifolds. In *Proc. National Academy of Sciences of the United States of America (PNAS)*, pages 8431–8435, 1998.

[67] R. Kimmel and J.A. Sethian. Optimal algorithm for shape from shading and path planning. *Journal of Mathematical Imaging and Vision*, 14(3):237–244, 2001.

[68] R. Kimmel, K. Siddiqi, B. B. Kimia, and A. M. Bruckstein. Shape from Shading: Level set propagation and viscosity solutions. *International Journal of Computer Vision*, 16:107–133, 1995.

[69] T.G. Kolda, R.M. Lewis, and V. Torczon. Optimization by direct search: new perspectives on some classical and modern methods. *SIAM Review*, 45:385–482, 2003.

[70] R. Kozera. Existence and uniqueness in photometric stereo. *Appl. Math. Comput.*, 44(1):103, 1991.

[71] R. Kozera. Uniqueness in Shape from Shading revisited. *Journal of Mathematical Imaging and Vision*, 7(2):123–138, 1997.

[72] Y.G. Leclerc and A.F. Bobick. The direct computation of height from shading. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 552–558, Maui, HI, USA, 1991.

[73] C.T. Lin and E. Tadmor. $L^1$ stability and error estimates for Hamilton-Jacobi solutions. *Num. Math.*, 87:701–735, 2001.

[74] P.L. Lions. Generalized solutions of Hamilton-Jacobi equations. *Number 69 in Research Notes in Mathematics*, 1982.

[75] P.L. Lions, E. Rouy, and A. Tourin. Shape-from-Shading, viscosity solutions and edges. *Numerische Mathematik*, 64(3):323–353, 1993.

[76] R. Mecca and M. Falcone. Uniqueness and approximation of a photometric Shape-from-Shading model. *SIAM J. Imaging Sciences*, 6(1):616–659, 2013.

[77] R. Mecca and S. Tozza. Shape reconstruction of symmetric surfaces using photometric stereo. In Michael Breuss, Alfred Bruckstein, and Petros Maragos, editors, *Innovations for Shape Analysis: Models and Algorithms*, pages 219–243. Ed. Springer, 2013. ISBN: 978-3-642-34140-3.

[78] T. Okatani and K. Deguchi. Shape reconstruction from an endoscope image by Shape from Shading technique for a point light source at the projection center. *Computer Vision and Image Understanding*, 66(2):119–131, 1997.

[79] J. Oliensis. Shape from Shading as a partially well-constrained problem. *Computer Vision, Graphics and Image Processing: Image Understanding*, 54(2):163–183, 1991.

[80] J. Oliensis. Uniqueness in Shape from Shading. *International Journal of Computer Vision*, 6(2):75–104, 1991.

[81] J. Oliensis and P. Dupuis. A global algorithm for Shape from Shading. In *Computer Vision, 1993. Proceedings., Fourth International Conference on*, pages 692–701. IEEE International Conference on Computer Vision, May 1993.

[82] M. Oren and S.K. Nayar. Diffuse reflectance from rough surfaces. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 763–764, 1993.

[83] M. Oren and S.K. Nayar. Generalization of Lambert's reflectance model. In *Proc. International Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 239–246, 1994.

[84] M. Oren and S.K. Nayar. Seeing beyond Lambert's law. In *Proc. European Conference on Computer Vision (ECCV)*, pages 269–280, 1994.

[85] M. Oren and S.K. Nayar. Generalization of the Lambertian model and implications for machine vision. *International Journal of Computer Vision*, 14(3):227–251, 1995.

[86] B. T. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.

[87] M.J.D. Powell. Developments of NEWUOA for minimization without derivatives. *IMA J. Numer. Anal.*, 28:649–664, 2008.

[88] E. Prados, F. Camilli, and O. Faugeras. A viscosity solution method for Shape-from-Shading without image boundary data. *ESAIM: Mathematical Modelling and Numerical Analysis*, 40(2):393–412, 2006.

[89] E. Prados and O. Faugeras. A mathematical and algorithmic study of the lambertian sfs problem for orthographic and pinhole cameras. Rapport de recherche 5005, Institut National de Recherche en Informatique et en Automatique, Sophia Antipolis, France, November 2003.

[90] E. Prados and O. Faugeras. Perspective Shape from Shading and viscosity solutions. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 826–831, 2003.

[91] E. Prados and O. Faugeras. Shape from Shading: a well-posed problem? In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 870–877, 2005.

[92] E. Prados, O. Faugeras, and E. Rouy. Shape from Shading and viscosity solutions. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *Computer Vision , ECCV 2002*, volume 2351 of *Lecture Notes in Computer Science*, pages 790–804. Springer Berlin Heidelberg, 2002.

[93] E. Prados and S. Soatto. Fast marching method for generic shape from shading. In Nikos Paragios, Olivier Faugeras, Tony Chan, and Christoph Schnörr, editors, *Variational, Geometric, and Level Set Methods in Computer Vision*, volume 3752 of *Lecture Notes in Computer Science*, pages 320–331. Springer Berlin Heidelberg, 2005.

[94] J. Qian, Y.-T. Zhang, and H.-K. Zhao. A fast sweeping method for static convex Hamilton-Jacobi equations. *J. Sci. Comput.*, 31:237–271, 2007.

[95] H. Ragheb and E.R. Hancock. Surface radiance correction for Shape from Shading. *Pattern Recognition*, 38(10):1574–1595, 2005.

[96] T. Rindfleisch. Photometric method for lunar topography. *Photogrammetric Engineering*, 32(2):262–277, 1966.

[97] E. Rouy and A. Tourin. A viscosity solutions approach to Shape-from-Shading. *SIAM Journal on Numerical Analysis*, 29(3):867–884, June 1992.

[98] M. Sagona. *Numerical methods for degenerate Eikonal type equations and applications*. PhD thesis, Dipartimento di Matematica dell'Università di Napoli ""Federico II", Napoli, Italy, November 2001.

[99] W. Schneider, K. Ranzinger, A. Balch, and C. Kruse. A dynamic programming approach to first arrival traveltime computation in media with arbitrarily distributed velocities. *GEOPHYSICS*, 57(1):39–50, 1992.

[100] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[101] J.A. Sethian. Fast-marching level-set methods for three-dimensional photolithography development. In *Optical Microlithography IX*, volume 2726, pages 262–272. SPIE, 1996.

[102] J.A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 2nd edition, 1999.

[103] J.A. Sethian and A. Vladimirsky. Ordered upwind methods for static hamilton–jacobi equations: Theory and algorithms. *SIAM Journal on Numerical Analysis*, 41(1):325–363, 2003.

[104] P.E. Souganidis. Approximation schemes for viscosity solutions of Hamilton-Jacobi equations. *J. Differential Equations*, 57:1–43, 1985.

[105] T.M. Strat. A numerical method for Shape from Shading from a single image. Master's thesis, Department of Electrical Engineering and Computer Science, Massachussetts Institute of Technology, Cambridge, MA, USA, 1979.

[106] R. Szeliski. Fast Shape from Shading. *Computer Vision, Graphics and Image Processing: Image Understanding*, 53(2):129–153, 1991.

[107] A. Tankus and N. Kiryati. Photometric stereo under perspective projection. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 611–616, 2005.

[108] A. Tankus, N. Sochen, and Y. Yeshurun. Perspective Shape-from-Shading by fast marching. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 43–49, 2004.

[109] A. Tankus, N. Sochen, and Y. Yeshurun. Shape-from-Shading Under Perspective Projection. *International Journal of Computer Vision*, 63(1):21–43, 2005.

[110] J. F. Thompson, B.K. Soni, and N.P. Weatherill. *Handook of Grid Generation*. CRC Press, 1998.

[111] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. *Journal of the Optical Society of America*, 57(9):1105–1114, 1967.

[112] S. Tozza and M. Falcone. A Semi-Lagrangian Approximation of the Oren-Nayar PDE for the Orthographic Shape–from–Shading Problem. In Sebastiano Battiato and José Braz, editors, *Proceedings of the 9th International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 3, pages 711–716. SciTePress, 2014.

[113] P.S. Tsai and M. Shah. Shape from Shading using linear approximation. *Image and Vision Computing Journal (Elsevier, Oxford)*, 12(8):487–498, 1994.

[114] Y. Tsai, L. Cheng, S. Osher, and H. Zhao. Fast sweeping algorithms for a class of Hamilton-Jacobi equations. *SIAM J. Numer. Anal.*, 41:673–694, 2003.

[115] Y. R. Tsai. Rapid and accurate computation of the distance function using grids. *Journal of Computational Physics*, 178(1):175–195, 2002.

[116] J. N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538, 1995.

[117] A. Vladimirsky. Static pdes for time-dependent control problems. *Interfaces and Free Boundaries*, 8:281–300, 2006.

[118] O. Vogel, M. Breuß, and J. Weickert. Perspective Shape from Shading with non-Lambertian reflectance. In *Proc. DAGM Symposium on Pattern Recognition*, pages 517–526, 2008.

[119] G.-H. Wang, J.-Q. Han, and X.-M. Zhang. Three-dimensional reconstruction of endoscope images by a fast shape from shading method. *Measurement Science and Technology*, 20(12):125801, 2009.

[120] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1):134–144, 1980.

[121] P.L. Worthington and E.R. Hancock. New constraints on data-closeness and needle map consistency for Shape-from-Shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(12):1250–1267, 1999.

[122] C. Wu, S. Narasimhan, and B. Jaramaz. A multi-image Shape-from-Shading framework for near-lighting perspective endoscopes. *International Journal of Computer Vision*, 86:211–228, 2010.

[123] K.-J. Yoon, E. Prados, and P. Sturm. Joint estimation of shape and reflectance using multiple images with known illumination conditions. *International Journal of Computer Vision*, 86:192–210, 2010.

[124] R. Zhang, P.-S. Tsai, J.E. Cryer, and M. Shah. Shape from Shading: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, 1999.

[125] Y. Zhang, H. Zhao, and S. Chen. Fixed-point iterative sweeping methods for static Hamilton-Jacobi equations. *Methods and Applications of Analysis*, 13(3):299–320, sep 2006.

[126] H. Zhao. A fast sweeping method for eikonal equations. *Math. Comput.*, 74(250):603–627, 2005.

[127] H.-K. Zhao, S. Osher, B. Merriman, and M. Kang. Implicit and non-parametric shape reconstruction from unorganized data using a variational level set method. *Computer Vision and Image Understanding*, 80(3):295–314, 2000.

[128] Z. Zheng, L. Ma, Z. Li, and Z. Chen. An extended photometric stereo algorithm for recovering specular object shape and its reflectance properties. *Computer Science and Information Systems*, 7(1):1–12, 2010.