Perspective Shape from Shading: Models, Algorithms, and Applications

Oliver Vogel

Dissertation zur Erlangung des Grades
des Doktors der Naturwissenschaften
der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

Saarbrücken
2010

Tag des Kolloquiums

31.03.2011

Dekan

Prof. Dr. Holger Hermanns

Prüfungsausschuss

Prof. Dr. Raimund Seidel (Vorsitz)
Prof. Dr. Joachim Weickert (1. Gutachter)
Prof. Dr. Alfred Bruckstein (2. Gutachter)
PD Dr. Michael Breuß

Abstract

Shape from shading (SfS) constitutes the problem of recovering the three-dimensional shape of objects in a scene at hand given a single two-dimensional input image. While the early years of SfS research were dominated by orthographic models, perspective models became popular in the recent years. In this thesis, we extend the state-of-the-art model by Prados and Faugeras to a more general model that also allows the handling of specular highlights in images. We derive a partial differential equation (PDE) corresponding to this novel model.

We introduce a new numerical solver for this PDE. In contrast to other methods used in the field, the presented scheme is much easier to implement. A numerical scale analysis motivates an optimal choice of a discretisation and gives an upper bound for the time step size below which the iterative method is stable.

In a comprehensive experimental evaluation we analyse the different components of the newly introduced model. We confirm the findings of the numerical scale analysis and compare the new iterative scheme to existing numerical methods from the literature, showing a substantial increase in performance. An analysis of the different convergence properties of the numerical methods motivates a hybrid scheme that allows to improve the performance in some cases. Furthermore, we introduce a non-iterative fast marching (FM) solver tailored to the new model. In contrast to other FM methods for previous SfS models, the presented approach does not require any depth information to be provided a priori. Experimentally, we confirm that this FM method is very fast and enables the reconstruction of large scenes.

In addition to introducing these high-performance methods, we also present two ways to parallelise them. The first approach introduces a novel way to FM on a multi-core CPU architecture. In a second algorithm, we combine the new iterative method with the concept of FM. This allows for a massively parallel implementation on graphics hardware which achieves real-time performance.

Finally, we address several general difficulties SfS can have on real-world images. By embedding SfS in a framework with other sophisticated methods from image processing, we introduce a method which allows to apply SfS to a class of real-world images.

Zusammenfassung

Die Arbeit beschäftigt sich mit perspektivischem Shape-from-Shading (SfS). Bei SfS geht es darum, aus einem einzigen, gegebenen zweidimensionalen Bild ein dreidimensionelles Modell der in dem Bild dargestellten Oberfläche zu rekonstruieren. Bei der Entwicklung von SfS-Verfahren ist es notwendig, Annahmen über die Art und Weise zu machen, wie das Ursprungsbild aufgenommen wurde. In den Anfangszeiten der Forschung in diesem Gebiet war die gängiste Annahme, dass das Bild mittels orthographischer Projektion erzeugt wurde. In dieser Arbeit wird ein modernerer Ansatz mit Hilfe perspektivischer Projektion betrachtet. Mit diesen Modellen sind deutlich bessere Rekonstruktionen möglich als mit orthographischen Verfahren. Basierend auf einem modernen perspektivischen Modell leiten wir ein neues Modell her, das es uns ermöglicht, deutlich schwierigere und realistischere Oberflächen zu rekonstruieren als mit den bisher bekannten Verfahren. Insbesondere ist es mit dem neuen Modell möglich, Oberflächen zu rekonstruieren, die helle Lichtreflexe enthalten. Zu diesem Modell leiten wir eine partielle Differentialgleichung her, deren Lösung die gesuchte Oberfläche liefert.

Für die Lösung solcher Differentialgleichungen entwickeln wir ein numerisches Verfahren. Dieses zeichnet sich im Vergleich zu anderen Methoden in diesem Gebiet vor allem dadurch aus, dass es deutlich einfacher zu implementieren ist. Durch eine numerische Skalenanalyse motivieren wir die Wahl einer optimalen Diskretisierung und erhalten ein Stabilitätskriterium für die maximale Zeitschrittweite in unserem Verfahren.

In einer umfassenden experimentellen Studie analysieren wir die verschiedenen Komponenten des neuen Modells. Wir bestätigen die Ergebnisse der Skalenanalyse experimentell und vergleichen das neue iterative Verfahren mit bekannten Verfahren aus der Literatur. Dieser Vergleich ergibt, dass das neue Verfahren deutlich schneller ist. Eine genaue Analyse des Konvergenzverhaltens der verschiedenen numerischen Verfahren motiviert ein hybrides Schema, das Vorteile verschiedener Verfahren kombiniert, wodurch in einigen Fällen ein weiterer Performancegewinn erzielt werden kann.

Weiterhin führen wir eine auf das neue Modell zugeschneiderte Fast-Marching-Methode ein. Im Gegensatz zu Fast-Marching-Methoden zu früheren SfS-Modellen zeichnet sich der vorgestellte Ansatz dadurch aus, dass zu Beginn keinerlei Tiefeninformation bekannt sein muss. Experimentell bestätigen wir, dass diese Fast-Marching-Methode überaus effizient ist und die Rekonstruktion auch größere Szenen ermöglicht.

Um dieses sehr schnelle numerische Verfahren noch weiter beschle-

unigen zu können, präsentieren wir noch zwei Wege, dieses zu parallelisieren. Der erste solche Ansatz ist ein neuer Weg, Fast-Marching-Methoden auf mehrere CPUs zu parallelisieren. In einer zweiten Methode zeigen wir, wie man die Fast-Marching-Methode mit dem iterativen Verfahren kombinieren kann, und wie man diese kombinierte Methode auf einer Grafikkarte einsetzen kann. Mit dieser Methode erhalten wir Echtzeitperformance.

Schließlich untersuchen wir noch die Herausforderungen, die bei SfS auf echten Bildern auftreten. Durch die Kombination von SfS mit Segmentations- und Interpolationssverfahren erhalten wir ein Verfahren, das es uns ermöglicht, SfS erfolgreich auf echte Bildern anzuwenden.

Acknowledgments

I thank Joachim Weickert for giving me the opportunity to work in his group as well as for his constant support and advice.

I thank Michael Breuß for the many discussions we had over the last years, his advice, and his helpful comments.

I thank all my collaborators, Michael Breuß, Emiliano Cristiani (University of Rome), Jean-Denis Durou (Irit, Toulouse), Maurizio Falcone (University of Rome), Pascal Gwosdek, Kai Hagenburg, Thomas Leichtweis, Levi Valgaerts, Martin Welk, and Joachim Weickert and the rest of the MIA Group for helpful discussions and the time and energy they invested into doing research with me.

I thank Michael Breuß, Kai Hagenburg, and Joachim Weickert for reading drafts of my thesis and their helpful feedback.

Furthermore, I would like to thank Joachim Weickert and Alfred Bruckstein (Technion, Israel) for agreeing to review my thesis.

Finally, I would like to thank the German Research Foundation (DFG) for funding large parts of my research, mainly through the Graduiertenkolleg "Leistungsgarantien für Rechnersysteme".

Contents

Chapter 1

Perspective Shape from Shading

In this chapter, we give a short overview over the general concept of perspective shape from shading (SfS) and the model aspects involved. Furthermore, an overview of the contributions made in this thesis as well as the structure of the thesis is given.

Before actually discussing perspective SfS, we should give a short overview of the general concept of SfS itself. SfS deals with the reconstruction of a three-dimensional surface from a single two-dimensional image, in contrast to other methods in computer vision which frequently require two or even more images. This is done by exploiting information present in this single image. For this, several assumptions are made, such as the lighting conditions being known a priori or the depicted surface having certain reflectance properties. In general, however, the underlying concept which is exploited is that surface patches that are illuminated by the light source are brighter than those illuminated by a light source in a flat angle. A good example for this is Earth. In the morning, when the sun rises, it is not quite as bright as around noon, when the sun reaches its highest point and hits Earth in a much steeper angle. If we would take a picture of earth from a sufficient distance, it would appear brighter in areas where the sun is standing high.

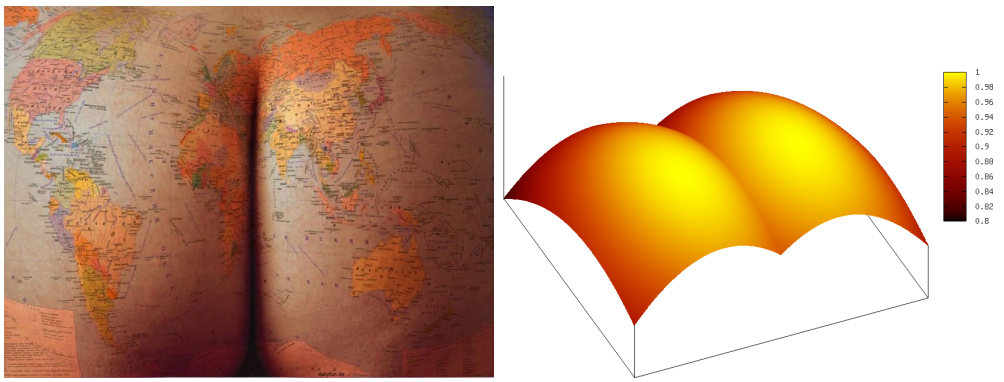

Figure 1.1: Illustration of the concept of 3D-reconstruction. Left: Die "Arschkarte". Right: Fake 3D surface loosely depicting the shape of a buttocks.

Figure 1.1[1] illustrates the general concept of 3D-reconstruction and it also shows a fundamental difficulty of SfS: Texture. Methods using more than one input image frequently rely on sufficient texture information in input images to obtain shape information by using correspondences between the images. SfS, however, relies on brightness information to reflect different exposure to light. Texture, however, violates this concept, as different colours or surfaces reflect light differently. Later in this thesis, we will discuss ways to cope with some forms of texture. By itself, the image in Figure 1.1 is much too difficult for a simple SfS technique out of the box, the "reconstruction" shown here is hand-crafted to illustrate the concept.

1.1 Perspective Shape from Shading

One of the crucial assumptions for SfS is the camera projection model. In the beginning of SfS research (from 1970 towards the end of the 20th century), the commonly used projection model was orthographic projection. Starting in the late 1990s, using perspective projection became more and more popular. Images acquired with perspective projection look considerably different from images acquired using orthographic projection, as Figure 1.2 illustrates. Another aspect that is frequently considered in perspective

[1]The term "Arschkarte" stems from the German idiom "die Arschkarte ziehen" (to get the crap card), which stands for having bad luck, to get into the worst of all possible outcomes of a situation. The word "Arsch" means ass and the word "Karte" both means card and map in German, which forms the basis for the naming of this image.

model is the physically motivated aspect of a decreasing light intensity with increasing distance from the light source. This effect is also visible in Figure 1.2. Generally, perspective projection is much closer to reality, as both the human visual system and standard photographic equipment perform roughly a perspective projection, while orthographic projection is only realistic for objects that are very far away. In addition, as can be observed in Figure 1.2, perspective images exhibit much more variation in grey values, i.e. there is more information contained in them, which allows for a significant improvement of reconstruction quality. In the recent years, all significant contributions to SfS research have been based on perspective projection, as the obtainable results are clearly superior, while orthographic projection suffers from some inherent problems.

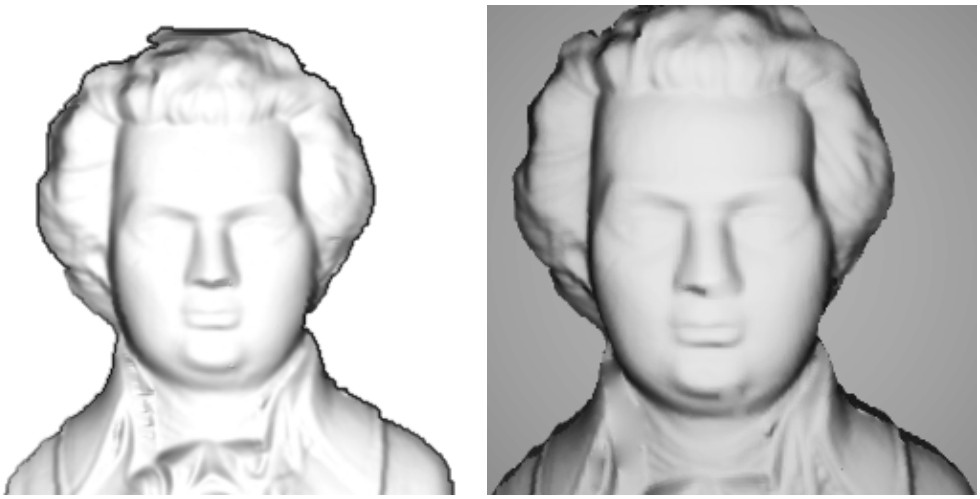

Figure 1.2: Renderings of the synthetic Mozart face. Left: Orthographic projection. Right: Perspective projection.

In this thesis, perspective projection will also be the choice for all novel models presented. The model aspect of orthographic vs perspective projection will be discussed in greater depth in Chapter 2, along with the historic development of SfS models from orthographic models to the most recent models presented in this thesis.

1.2 Structure and Contributions

In Chapter 2 different SfS models are introduced and discussed. The key part of that chapter is the introduction of a novel, accurate model for SfS

and the derivation of the arising partial differential equation (PDE). In this chapter, a state-of-the-art model for SfS is extended by including specular reflectance of surfaces into this model. In an extensive experimental study, the effect of this modelling step is investigated in detail, showing that the model is still applicable even for surfaces with extreme reflectance properties, where existing models fail. In addition, we give a quite detailed survey on SfS models existing in the literature. Key model assumptions like different projection techniques or assumptions on surface or illuminations are introduced, discussed, and an overview is given on the impact of these models in existing SfS models.

Chapter 3 deals with the introduction of an efficient numerical solver for the PDE derived in Chapter 2. We show that the method is stable under certain conditions and that it is more efficient than other solvers used for state-of-the-art SfS algorithms. This study is both done experimentally by comparing run-times of different solvers as well as theoretically by performing a numerical scale analysis of the PDE, which also gives a motivation for the actual numerical realisation. In addition, we show that it is possible to combine advantages of different numerical schemes to obtain even faster convergence. The chapter is mainly restricted to iterative solvers. In a survey part, we give an overview over the many numerical techniques that have been used for SfS methods in the literature. Furthermore, an introduction to the mathematical basics of the components for the proposed scheme is given.

In Chapter 4, an alternative numerical approach for solving the PDE proposed in Chapter 2 is discussed, a fast marching approach. Fast marching (FM) methods are non-iterative methods which allow for solving a large class of hyperbolic PDEs very efficiently. We show that this technique is also applicable at the SfS model presented in this thesis. Usually, FM methods have the undesirable property that the correct solution of the PDE has to be known a priori in certain parts of the image. We show that for the proposed SfS model, it is possible to overcome this problem and to obtain a very fast non-iterative solver which does not need any Dirichlet data to be provided by the user. In an extensive experimental study, we investigate the differences in performance between this non-iterative method and iterative methods. In addition, the numerical effect on the reconstruction is discussed in detail as well. As in the previous chapter, a broad survey part gives overview over existing approaches to use FM methods for SfS.

In Chapter 5, we introduce different ways to parallelise the efficient numerical techniques from the previous chapters. The first such method introduces a way to parallelise the FM algorithm presented in Chapter 4 on a multi-core CPU architecture. Although FM methods are usually of a

sequential nature, we introduce a domain decomposition-free approach for parallelising this numerical solver while using favourable properties of the SfS model to achieve parallelisability. In a second part, we introduce a way to massively parallelise SfS on graphics hardware. While iterative solvers like the Jacobi method are well suited for massively parallel architectures, FM methods are not. However, since a sequential FM implementation on the CPU can in many cases easily outperform a massively parallel, iterative method, with much smaller implementation effort, we show how to combine the two numerical approaches to a method that combines the efficiency of a FM scheme with the favourable parallelisation properties of an iterative solver. By that, we obtain a method that can achieve real-time performance on standard test images and very reasonable computation times even on very large input images.

Finally, in Chapter 6 the applicability of SfS methods to actual real-world images is discussed. We observe that real-world images can be much more challenging than synthetic images. In this chapter, we introduce a method that allows for doing SfS on a large class of real-world images. The method shows how to combine SfS with other techniques from image processing like segmentation or inpainting methods to obtain convincing 3D reconstructions of real-world images. The proposed approach is efficient enough to retain interactive performance for the whole method.

The thesis is concluded by a detailed summary of the results of the thesis and an outlook on possible future improvements both on the model and on the numerical side in Chapter 7.

Chapter 2

Shape from Shading Models

In this chapter, different models for the shape-from-shading problem is discussed. There is a fine line between model and numerical issues for an SfS method.

Any assumptions on the surface, the image acquisition, illumination conditions or similar issues are clearly aspects of the model. Numerical issues are discretisation, boundary conditions, the solving technique, and similar aspects.

The choice of the PDE solved is mainly a decision for the model. However, the choice of the PDE has an impact on numerical issues as well. When assembling the PDE, key decisions on the numerical side are being made already. For example, the choice whether to use a Hamilton-Jacobi-equation or a Hamilton-Jacobi-Bellman-equation makes a choice between different solvers: While the Hamilton-Jacobi suggests the use of the method suggested here, choosing a Hamilton-Jacobi-Bellman-equation suggests the use of an optimal control approach. In this chapter, the numerical consequences of such decisions will be mainly left aside, they will be discussed in the next chapter.

2.1 Contributions

In this chapter, new contributions to different areas of SfS research are presented.

In Section 2.3, a survey on different SfS models from different decades in SfS research will be given. Most survey papers in the field are concentrated on just one branch of SfS research. We attempt to give a good overview on SfS models since the 1960s.

In Section 2.5, an extended model to the one of Prados et al. [67] is proposed. The main goal of this model is to cope with specular highlights being present in the image. As to this date, this is very likely the most advanced SfS model existing. Related publications to this section are [8, 12, 96].

In the experimental Section 2.6 of this chapter, an analysis of the model ingredients will be done in a way that is new in the literature so far, discussing the influence of the focal length and specular highlights on the reconstruction quality. In addition to analysis done in the related publications [12, 97], a detailed investigation on the effects of different parameters on the quality of the reconstruction of the surface and the effects of different state-of-the-art approaches to the modelling of the PDE is done. In these investigations, numerical aspects are not considered. In addition to that, issues like existing ambiguities even in state-of-the-art models, the problem of finding suitable error measures, the rendering of the surfaces, and the dependence of the ground truth on rendering parameters will be discussed. In the literature on SfS, many of these investigations have not been made so far, or only at a very basic level.

2.2 Theoretical Background

Projection

The *projection* describes a relation between the surface and the image showing the surface. Therefore, assumptions on the surface are important for SfS models. It is an assumption on how the image is obtained, on the transition of an actual object to a two-dimensional image depicting this object. While for synthetic images, one can simulate any projection method, the most relevant cases are images acquired by a real cameras. In this section, two simple camera models from computer graphics are discussed. In general, it is possible to group SfS models in two classes: models relying on *orthographic* and on a *perspective* projection, respectively.

Perspective projection is a very simple and very common projection model used in computer graphics. It simulates the behaviour of a *pinhole camera*. A pinhole camera is simply a closed box with a very small hole on one side of the box and film on the other side of the box, the viewing plane of the camera. An ideal pinhole camera, i.e., a pinhole camera where the opening is just a single point, can be described by a perspective projection on a viewing plane between the object and the optical centre of the camera. The distance between viewing plane and optical centre of the projection is the same as the size of the box of the pinhole camera. Figure 2.1 illustrates the relation between perspective projection and a pinhole camera.

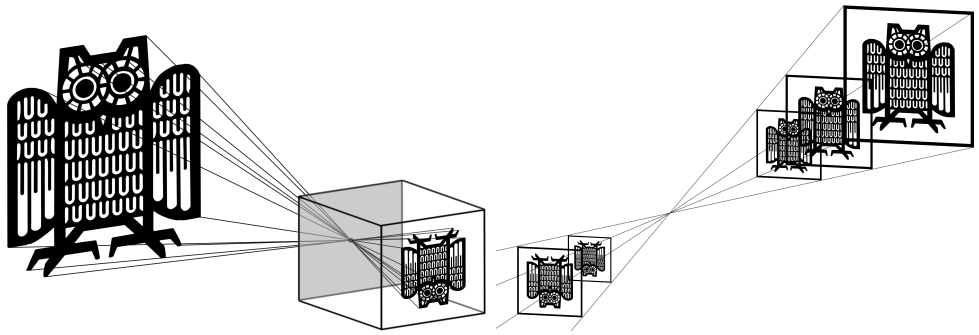

Figure 2.1: Left: Pinhole camera model. Right: Perspective projection.

Important notions for perspective projection are the *focal length* f, which is the distance between the viewing plane and the projection centre, the distance r between projection centre and surface. The distance r is often also referred to as *perspective depth* and is measured in every pixel of the image. When discussing the numerics of perspective SfS methods, however,

one usually defines depth in multiples of the focal length. Pinhole cameras are relatively close to real cameras.

Perspective models for SfS are generally rather complex, which might be the main reason why in the beginning of SfS research, an even simpler projection model has been used: *orthographic projection*. Orthographic projection can be understood as a special case of a pinhole camera: a pinhole camera very far in the distance, infinitely far away from the surface. In this extreme case, the projection lines become parallel, making the viewing plane as wide and high as the surface. Figure 2.2 illustrates this kind of projection.

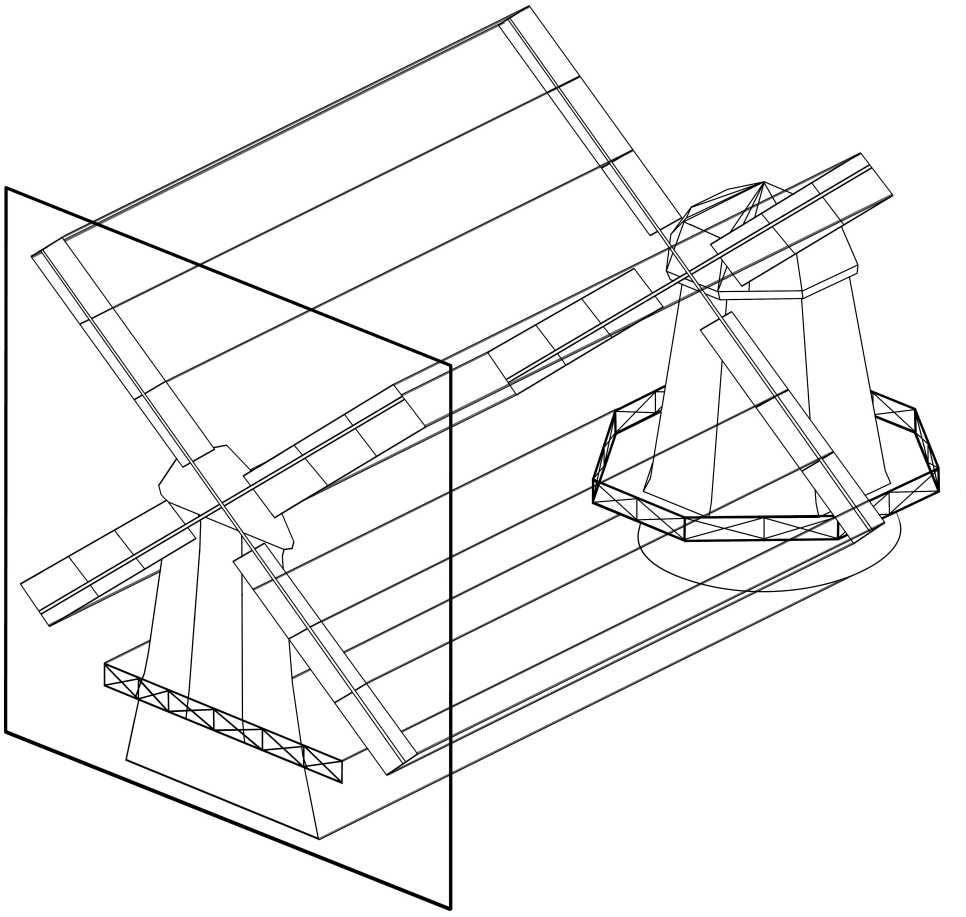

Figure 2.2: Orthographic projection.

Surface Reflectance

Another important concept to be aware of are models describing how energy is reflected on a surface. In computer graphics, there are numerous models for reflectance. Within this thesis, we consider two rather simple ones: *Lambertian* and *specular* reflectance.

A surface is called a *Lambertian* surface, if the incoming light is reflected only diffusely, i.e., the incoming energy is uniformly reflected into all directions. This type of reflectance is illustrated in Figure 2.3.

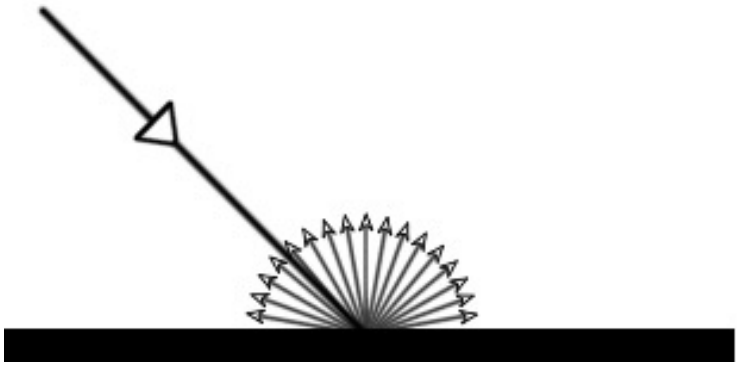

Figure 2.3: Lambertian reflectance: Light is reflected equally in all directions.

If light is reflected only *specularly*, the surface acts like an ideal mirror. Incoming light is reflected in the same angle relative to the surface normal as its incoming angle. This type of reflectance is illustrated in Figure 2.4.

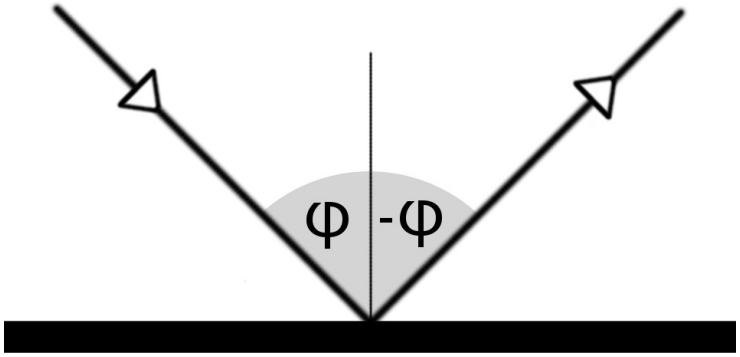

Figure 2.4: Specular reflectance: Light is reflected like on an ideal mirror.

In nature, reflectance properties of surfaces can be far more complex. However, these two models allow for a sufficiently good approximation for SfS applications. In nearly all SfS models discussed in this the reflectance properties of a surface will be a combination of these reflectance types. In most models in the literature, only Lambertian surfaces are considered.

Illumination

In computer graphics, there are many different models for light sources. Within this thesis, only two of them are relevant: point light sources and directional light sources. The concept of these light sources is similar to what has been discussed for projection.

Directional light sources illuminate a surface in a parallel way. For practical applications, the most important information about a light source is the direction of the incoming light, also called the light vector, in a specific point on the surface as well as the intensity of the light. In the case of a directional light, both the light vector and the light intensity are constant throughout the scene.

As the name suggests, a point light source is a point from which light is emitted in all directions. For this type of light source, the light vector points from the point light source on the point of the surface. This results in light vectors that depend on the actual shape of the object. In many SfS models the intensity of the light is assumed to be constant for every surface point as well, but this is not true in reality. In reality, light intensity decreases quadratically with the distance to the light source. This light attenuation factor is considered in more advanced SfS models.

2.3 Previous Work

Before investigating existing models for SfS, it is necessary to discuss which components are necessary for such a model. In general, a model for SfS consists of assumptions on the following components:

1. image acquisition, projection
2. surface reflectance
3. illumination conditions
4. smoothness or continuity of the surface
5. visibility of the surface, existence of shadows

Assumptions on these points are crucial for defining a good model for SfS. A huge variation of different models can be found in the literature. In the next sections, the most important ones of them are reviewed.

Orthographic Models

In this section, the focus lies on a certain class of camera models: Models that use an *orthographic projection*.

The first use of an orthographic model for SfS was also the very first publication on SfS: In 1970, Horn proposed a method for SfS in his Ph.D. thesis [32], which set the basis for many variational SfS methods that have been subsequently developed. Besides orthographic projection, he assumed the surface to have only *Lambertian* reflectance properties.

In addition, he assumed the surface to be illuminated by a single light source placed infinitely far away from the surface, which emits light of a single colour and intensity. This leads to *parallel illumination*. The direction of this light must be known in this model.

He also assumed the surface to be smooth and fully visible without any shadows. He introduced the notion of *singular points*, points of maximum or minimum brightness, representing points of locally minimal distance from the viewer or points with normals orthogonal to the viewing direction. It was stated that for his method to be consistent, the actual shape at these points must be known. Note that the notion of singular points will be of special importance in several parts of this thesis. However, in practice, a slight variation of Horn's definition of singular points is more appropriate. In 1995, Kimmel and Bruckstein [43, 82] pointed out that knowledge on

singular points, in particular their classification in minima and maxima, is essential for uniqueness and quality of orthographic SfS methods.

Horns model influenced SfS research for several decades. Plenty of SfS methods have been based on this model, such as more work by Horn [33–35], Brooks and Horn [14, 37]. In addition to that, Horn and Brooks wrote a book on the state of SfS in 1989 [38].

In essence, this model by Horn has been the only orthographic model. The main variations that occurred have been variations in the reflectance function in the model. Examples for such methods can be found for example in [3, 50].

In 1999, Zhang et al. [105] reviewed the state of the art of orthographic SfS. Their conclusion has been that orthographic SfS methods simply do not work properly. The problems for this lie in the model. Orthographic SfS is very ambiguous and thus extremely ill-posed. In addition to that, many numerical methods for orthographic SfS simply do not work. However, in the recent years there also had been slight progress in the development of classic SfS methods that produce visually more convincing results [2, 98].

Perspective Models

The idea of a perspective SfS method has been brought up in 1991 by Dupuis and Oliensis [23,24,56–58,60]. A closely related model has been discussed by Rouy and Tourin in 1992 [77]. However, the contributions of this paper had a much larger impact on the numerical side of SfS, which will be discussed in the next chapter. Their perspective model was a very basic one. They simply replaced the orthographic projection by a perspective one. By this, their images included perspective distortions, which is generally beneficial for SfS methods.

Much more noteworthy is a slight variation of this method, though. In 2003, Tankus et al. [86–90] proposed a method for perspective SfS with two point-light sources. Again, also for this method there have been ambiguities, such as images being invariant to a multiplicative factor on the depth. However, their model has been the first that could actually be applied to a set of real-world images with reasonable results. Also Prados et al. [70,71,75] published methods based on similar models. These papers showed a trend towards more realistic assumptions in SfS methods. A good review of these methods can be found in survey papers by Falcone et al. [27] and Durou et al. [26].

Perspective Shape from Shading with Light Attenuation

Although the method of Tankus et al. [86–88, 90] produced reasonable results, it was not the rise of perspective models that put SfS in a position where there is hope for actual applications for SfS. This came with the perspective model with light attenuation introduced by Prados et al. [67–70, 72–74] in 2004. Briefly spoken, he included the light attenuation term from physics into his model. In reality, light intensity decreases quadratically with the distance to the light source. Since it is extremely difficult to include such a constraint into a general model, Prados introduced a very elegant simplification into his model: He reduced the model to a single light source located in the projection centre of the perspective projection. By this simplification, the distance to the light source becomes identical to the perspective depth, the sought quantity, and makes it easily possible to include this into the model. Another convenient side effect of this trick was that shadows are impossible within this model and hence are not necessary to be included into the model. Prados showed that his model is well-posed in a sense [74]. His model allowed for impressive results compared to every other SfS method before. Interestingly, Prados has not been the first one to come up with the simplification of a point light source at the projection centre. In 1996, Okatani and Deguchi proposed a perspective method with exactly this simplification [55]. However, their method produced significantly worse results, mainly because they did not incorporate the light attenuation term.

Prados' model also solved several other problems that existed for most SfS models. Practically all SfS models, whether they are orthographic or perspective, always depended on some correct depth data to be provided a priori. This is not surprising, since all previous models have been ambiguous in some sense. Prados' method works without that. Providing state constraint boundary conditions, his method works completely without any depth data being provided. Visually, his method produced spectacular results compared to the whole SfS literature up to this point. However, on real-world images the results have not been perfect. He did a reconstruction of his face [67], but to get the method actually working on this image he had to interpolate the image in the eye region and he painted his face with a matte, white make-up. Nevertheless, the reconstructions looked very appealing in a quality unseen before.

This led several researchers to use variations of this model. The most important one is the model by Ahmed and Farag [2], who used non-Lambertian surfaces, in particular the Oren-Nayar-model, which replaces Lambertian

reflectance for another diffuse-type reflectance model. The Oren-Nayar model is particularly well-suited for the reconstruction of faces, which has been their main application and is very problematic with Prados' model.

2.4 Ambiguities in Shape from Shading Models

Many SfS models contain ambiguities. The simplest one is the shift ambiguity for Lambertian SfS models: in a Lambertian model, the depth can only be determined up to an additive constant, since neither perspective distortion nor any brightness change with the scene being farther away from the camera plane occur.

For perspective models, this ambiguity does not occur. However, a very similar ambiguity is one in perspective models that do not incorporate light attenuation, or even use parallel light. Here, the depth can only be determined up to a multiplicative constant [87, 90]. In perspective models that include light attenuation, this ambiguity is resolved as well..

The most famous ambiguity in SfS models is the so-called *concave-convex* ambiguity. The most simple setting where such an ambiguity can occur is the case of the Lambertian SfS model by Horn. For instance, an orthographic image of the outside of one half of a sphere would be absolutely identical to an image of the inside of the sphere. This is obvious by looking at the brightness equation of this model, where only derivatives of the gradient occur, and always squared. Switching from a concave to a convex surface just introduces a factor -1 to the gradients, which is eliminated by squaring them. It has been shown that perspective models which incorporate no light attenuation also exhibit a variant of this ambiguity [22].

First claimed by Prados [67, 73], it seems like the perspective SfS model with light attenuation does not suffer from a concave-convex ambiguity. It is even implied that this model would be well-posed. At several points of this thesis, it will become clear that this is not true. Certainly the ambiguity is not as obvious as in simpler models, but there is some kind of ambiguity, which is often neglected when discussing related SfS models. At a later point, an example will be presented where two clearly different and continuous surfaces have the same brightness function. Since this is a numerical issue, this can be found in the chapter on the numerics.

Another very interesting point that is closely related to ambiguities is the existence of different valid solutions at image discontinuities. This, however, is not an ambiguity in the model, since solutions of the arising PDEs are

usually assumed to be continuous. However, by imposing boundary conditions at image discontinuities, it is possible to obtain different surfaces that fit to one brightness function. This is closely related to the numerics used in the method and will be discussed in the respective chapter.

2.5 Extending the Model

In this section, a new model based on the Phong reflectance model [65, 66], which is well-known from computer graphics, is proposed. It is not a physically correct model, but it models in a simple way both diffuse and specular reflection. In addition to that, some ambient illumination is introduced. This model allows for a – for SfS standards – simple and easy-to-implement method.

While practically all SfS methods only consider Lambertian surfaces, the goal of this is simple: to include specular highlights in the model. In reality, surfaces are never purely Lambertian, they always exhibit some kind of specular highlights. While for orthographic methods, such models exists, this has not been done in most modern, perspective SfS models that use light attenuation, There is one notable exception to this by Ahmed and Farag [2], who use an entirely different reflectance model, which is focused on recovery of skin surfaces like human faces. They consider realistic, non-Lambertian models, however, also neglect specular highlights.

The *brightness equation* of the Phong reflectance model reads as

$$I(x) \;=\; k_a I_a + \sum_{\text{lights}} \frac{1}{r^2}\big(k_d I_d \cos\phi + k_s I_s (\cos\theta)^\alpha\big) \tag{2.1}$$

where $I(x)$ is the normalised grey value of the image pixel located at x. The cosine in the specular term is replaced by zero if $\cos\theta < 0$.

In this brightness equation, I_a, I_d, and I_s denote the intensities of the ambient, diffuse, and specular components of reflected light, respectively. The constants k_a, k_d, and k_s with $k_a{+}k_d{+}k_s \leq 1$ denote the ratio of ambient, diffuse, and specular reflection. The contributions due to all light sources are added up, i.e. practically one needs to compute the contribution due to each light source separately. For application in SfS, however, the special case of only one light source will be considered.

Concerning the individual reflection contributions, the *ambient light* models light present everywhere in a given scene, i.e. it is a base intensity. In practical applications, where homogeneous reflectance properties will be assumed for the entire scene, this amounts to simply subtracting

$I_a k_a$ in every pixel of the input image and neglect this term afterwards for the actual implementation of the method.

The intensity of *diffusely reflected light* in each direction is proportional to the cosine of the angle ϕ between surface normal and light source direction. This is Lambertian reflectance, as has been discussed in the previous paragraph.

The amount of *specular light* reflected towards the viewer is proportional to $(\cos\theta)^\alpha$, where θ is the angle between the ideal (mirror) reflection direction of the incoming light and the viewer direction, and α is a constant which models the roughness of the material. For $\alpha \to \infty$ this describes an ideal mirror reflection.

Another important component in (2.1) is the so-called *light attenuation factor* $1/r^2$, where r is the distance between light source and surface. Since this term is an effect that actually occurs in reality, it is a very reasonable assumption to make. Note that this model explicitly excludes light sources that are very far away like the sun. Such very far light sources would essentially result in parallel illumination, which would make the model ill-posed [22]. For a discussion of the theoretical implications of using this term see [67], where it is also claimed that this term makes SfS something like well-posed.

In the actual SfS model, a slightly simplified modification of the Phong reflectance model is used. As mentioned before, only a single light source is considered. In addition to that, it is required that this light source is located at the optical centre of the camera, cf. [67, 96]. The fact that view direction and light source direction are the same in this case simplifies the model a lot, since the angle between view direction and direction of specularly reflected light is halved by the surface normal, resulting in $\theta = 2\phi$. Finally, while the Phong model originally contains vector-valued light intensities representing colour information, for SfS applications only the image brightness is relevant. Hence, grey-valued images are assumed.

With these simplifications, equation (2.1) becomes

$$I(x) \;=\; k_a I_a + \frac{1}{r^2}\left(k_d(\vec{\mathrm{N}}\cdot\vec{\mathrm{L}})I_d + k_s(2(\vec{\mathrm{N}}\cdot\vec{\mathrm{L}})^2-1)^\alpha I_s\right), \qquad (2.2)$$

where $\vec{\mathrm{N}} = \frac{\vec{\mathrm{n}}(x)}{|\vec{\mathrm{n}}(x)|}$ denotes the unit normal vector at the surface at point $(x, S(x))^\top$, and where $\vec{L}$ is the unit light vector pointing towards the optical centre of the camera.

In (2.2), the reformulation

$$\cos\theta \;=\; \cos(2\phi) \;=\; (\cos\phi)^2-(\sin\phi)^2 \;=\; 2(\cos\phi)^2-1 \;=\; 2(\vec{\mathrm{N}}\cdot\vec{\mathrm{L}})^2-1 \qquad (2.3)$$

is used.

The Lambertian model, on which most recent SfS methods are based is a special case of formula (2.2), where $k_a = k_s = 0$ and $k_d = 1$.

As the normalised light source direction $\vec{\mathrm{L}}$ is given by

$$\vec{\mathrm{L}}\,(S(x)) \;=\; \frac{1}{\sqrt{|x|^2 + \mathsf{f}^2}}\,(-x, \mathsf{f})^T\;, \tag{2.4}$$

the inner product $\vec{\mathrm{N}} \cdot \vec{\mathrm{L}}$ can be evaluated to

$$\begin{aligned}
&\vec{\mathrm{N}} \cdot \vec{\mathrm{L}}\,(S(x)) \\
&= \frac{\left(\mathsf{f}\nabla u(x) - \dfrac{\mathsf{f}u(x)}{|x|^2 + \mathsf{f}^2}x\,, \nabla u(x)\cdot x + \dfrac{\mathsf{f}u(x)}{|x|^2 + \mathsf{f}^2}\mathsf{f}\right)^T \cdot (-x, f)^T}{|\vec{\mathrm{n}}(x)|\sqrt{|x|^2 + \mathsf{f}^2}} && (2.5)\\
&= \frac{-\mathsf{f}\nabla u(x)\cdot x + \dfrac{\mathsf{f}u(x)}{|x|^2 + \mathsf{f}^2}x\cdot x \;+ (\nabla u(x)\cdot x)\mathsf{f} + \dfrac{\mathsf{f}u(x)}{|x|^2 + \mathsf{f}^2}\mathsf{f}^2}{|\vec{\mathrm{n}}(x)|\sqrt{|x|^2 + \mathsf{f}^2}} && (2.6)\\
&= \frac{\dfrac{\mathsf{f}u(x)}{|x|^2 + \mathsf{f}^2}|x|^2 \;+ \dfrac{\mathsf{f}u(x)}{|x|^2 + \mathsf{f}^2}\mathsf{f}^2}{|\vec{\mathrm{n}}(x)|\sqrt{|x|^2 + \mathsf{f}^2}} = \frac{\mathsf{f}u(x)}{|\vec{\mathrm{n}}(x)|\sqrt{|x|^2 + \mathsf{f}^2}}\,. && (2.7)
\end{aligned}$$

By use of $r = \mathsf{f}u(x)$, from (2.2)-(2.7) follows

$$I(x) = k_a I_a + \frac{1}{\mathsf{f}^2 u(x)^2}\left(k_d \frac{u(x)Q(x)}{|\vec{\mathrm{n}}(x)|} I_d + k_s\left(\frac{2u(x)^2Q(x)^2}{|\vec{\mathrm{n}}(x)|^2} - 1\right)^{\alpha} I_s\right)\,, \tag{2.8}$$

with

$$|\vec{\mathrm{n}}(x)| \;=\; \sqrt{\mathsf{f}^2|\nabla u(x)|^2 + (\nabla u(x)\cdot x)^2 + u(x)^2Q(x)^2} \tag{2.9}$$

and

$$Q(x) \;=\; \sqrt{\mathsf{f}^2/(|x|^2 + \mathsf{f}^2)}\,. \tag{2.10}$$

The PDE (2.8) is hyperbolic, more specifically it is a Hamilton-Jacobi equation. Rewriting (2.8) yields the more convenient formulation

$$(I(x) - k_a I_a)\frac{\mathsf{f}^2|\vec{\mathrm{n}}(x)|}{Q(x)u(x)} - \frac{k_d I_d}{u(x)^2} - \frac{|\vec{\mathrm{n}}(x)|k_s I_s}{u(x)^3 Q(x)}\left(\frac{2u(x)^2Q(x)^2}{|\vec{\mathrm{n}}(x)|^2} - 1\right)^{\alpha} = 0\,. \tag{2.11}$$

Assuming in addition that the surface $\mathcal{S}$ is visible in the front of the optical centre, the depth u is strictly positive.

Then the change of variables $v = \ln(u)$ is used, which implies

$$\frac{|\vec{\mathrm{n}}(x)|}{u(x)} = \frac{\sqrt{\mathsf{f}^2|\nabla u(x)|^2 + (\nabla u(x)\cdot x)^2 + u(x)^2 Q(x)^2}}{u(x)} \tag{2.12}$$

$$= \sqrt{\frac{\mathsf{f}^2|\nabla u(x)|^2 + (\nabla u(x)\cdot x)^2 + u(x)^2 Q(x)^2}{u(x)^2}} \tag{2.13}$$

$$= \sqrt{\mathsf{f}^2|\nabla v(x)|^2 + (\nabla v(x)\cdot x)^2 + Q(x)^2}\,, \tag{2.14}$$

since $\nabla v(x) = \dfrac{1}{u(x)}\nabla u(x)$.

By some further simple computations, one eventually obtains the *PDE of Phong-based perspective Shape from Shading*:

$$\begin{aligned} &J(x)W(x) - k_d I_d \exp\left(-2v(x)\right) \\ &- \frac{W(x)k_s I_s}{Q(x)} \exp\left(-2v(x)\right)\left(\frac{2Q(x)^2}{W(x)^2} - 1\right)^{\alpha} = 0 \end{aligned} \tag{2.15}$$

where

$$J(x) = (I(x) - k_a I_a)\mathsf{f}^2/Q(x) \tag{2.16}$$

and

$$W(x) = \sqrt{\mathsf{f}^2|\nabla v|^2 + (\nabla v\cdot x)^2 + Q(x)^2}\,. \tag{2.17}$$

Equation (2.15) is the basis of the numerical implementation, which will be discussed in the next chapter.

This is a Hamilton-Jacobi-equation, and the solutions to this PDE one is interested in are continuous viscosity solutions [20]. The Lambertian part of the PDE (2.15) is convex, as has been shown in a similar context in [67]. The specular part, however, is not convex. Nevertheless, the algorithm converges in all cases. The investigation on the convexity of the model will be done in Chapter 4, where convexity will be discussed in a numerical context and convergence issues in the non-convex areas will be addressed. Note once again that the purely Lambertian model by Prados et al. is just a special case of this model. By setting $k_d = k_s = 0$, we obtains the same PDE as in [96].

The remainder of this chapter deals with a detailed experimental evaluation of this model. In particular differences between the Phong model and the purely Lambertian models will be investigated. When doing this, however, any numerical considerations are neglected, since these are to be discussed in the next chapter.

2.6 Tools for Experimental Analysis of the Model

In this section, the basics on experimental evaluation will be discussed. One aspect deals with test surfaces. For evaluating the model, synthetic surfaces are the best choice at this point. For dealing with real-world images, some extra effort is necessary. A successful technique for this will be discussed in Chapter 6. However, some images considered in this chapter are motivated by real-world scenes, like surfaces that have specular highlights. The most important difference is that the synthetic world is an ideal world, where a camera can be actually a pinhole camera, surfaces really have properties as described by our model, and no noise or other disturbing artefacts can occur.

In the next sections, the most important aspects of perspective SfS models and their impact on the result will be discussed. These aspects are the focal length and specularity of the surface. Continuity of the surface is more an issue of the numerics and will hence be analysed in the experimental section of the next chapter.

Test Surfaces

For SfS, there are no standard test surfaces the community commonly agrees on. The attempt closest to this has been done in a survey paper on orthographic SfS in 1999, where the following three surfaces have been proposed [105].

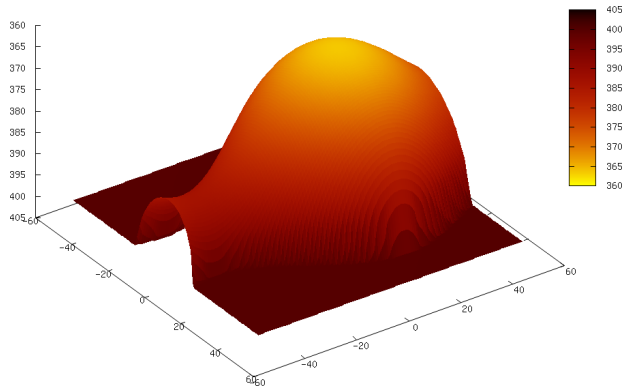

Figure 2.5: The Vase surface.

Figure 2.5 shows the vase surface. This is probably the most simple of the surfaces. However, the large gradient near the boundary of the vase can be difficult for perspective SfS, in particular for small focal lengths. This is irrelevant for its original use in orthographic SfS, though.

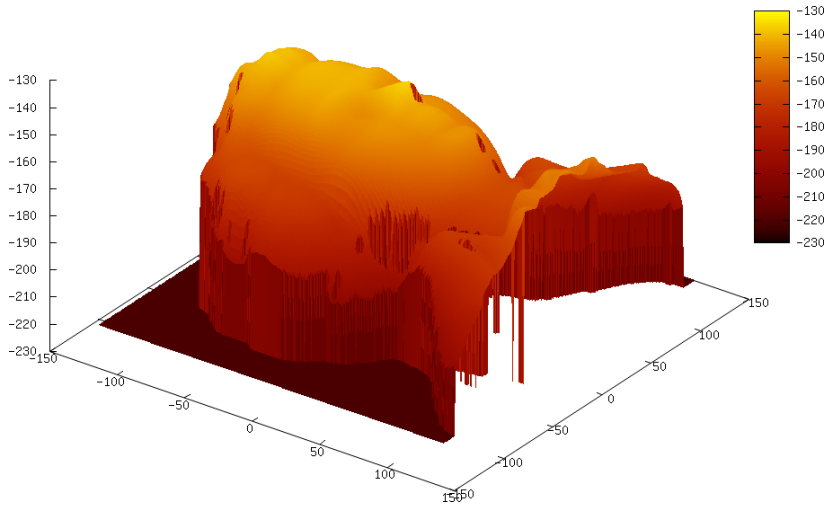

Figure 2.6: The Mozart face surface.

The second surface, the Mozart face, is shown in Figure 2.6. This is probably the most widely-known test surface for SfS. However, the surface, as it has been originally introduced, contains some errors, such as peaks in the surface. In most cases, this is ignored in the discussion of experiments on this surface. However, it is interesting to see how a model can cope with such artefacts.

In most of the synthetic experiments, these surfaces will be used. In Section 3.19, one more synthetic surface will be used for demonstrating a numerical effect at discontinuities.

Rendering Test Surfaces

So far, only test *surfaces* have been mentioned. However, the input to SfS is a test *image*. From a synthetic surface, an input image has to be rendered. Since we have several parameters here, like the focal length, the position of the camera, and the shading parameters, it is not surprising that even for the more prominent test surfaces, no standard test images have become widely accepted in the community. Therefore, a simple self-written ray-

tracing program is used to render images for each test case. For more details on ray tracing, see for instance [85].

Error Measures

Error measures for SfS are a critical issue. For orthographic SfS, this has already been discussed by Horn [35]. However, Horn's favourite error measures are not necessarily the best. Finally, it is suggested simply to consider a depth error. As will be demonstrated in the next chapter, other error measures might be also worth considering. For orthographic SfS, using a depth error is clearly not optimal due to the numerous ambiguities. However, for perspective SfS with light attenuation, a depth error is a valid and reasonable choice.

In this paragraph, three error measures will be discussed. The first two are two types of depth errors that give an impression of how accurately the depth has been reconstructed and on the distribution of the error. These two error measures are:

1. The *average depth error*, which is the sum of all depth errors divided by the corresponding true depth

$$\frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} \left| \frac{u_{i,j} - \overline{u}_{i,j}}{\overline{u}_{i,j}} \right| , \tag{2.18}$$

   where u is the computed depth and $\overline{u}$ is the ground truth depth.

2. The *maximal depth error*, which uses the same error as the previous error, but instead of taking the average, takes the maximum

$$\max_{(i,j)\in\Omega} \left| \frac{u_{i,j} - \overline{u}_{i,j}}{\overline{u}_{i,j}} \right| . \tag{2.19}$$

   This error measure shows how far off the largest outlier is. In most cases, this error will be used together with the average depth error to show both sides.

These depth errors are fairly natural choices for an error measure based on depth. Since depth is the quantity we compute, it is also the most natural choice for an error measure.

However, in a later experiment the following *brightness error* will provide interesting insight as well:

$$\frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} |J_{i,j} - R_{i,j}| \tag{2.20}$$

where R is the right-hand side of equation (2.15). This is an unusual choice for an error measure, since it depends on the discretisation used in the numerical method itself, and might thus be considered unfair for actual applications. In most experiments, however, only depth errors will be considered. This is the common practice of most authors of recent papers on SfS.

Obtaining the Ground Truth

In a previous paragraph, two of the test surfaces from [105] have been presented. On the first glance, the intuition might be that these surfaces constitute the ground truth for all experiments. While this is true for orthographic SfS, one needs to be very careful with that for perspective SfS. Depending on the focal length or how close the camera is to the surface, parts of the surface might be occluded. Also, a perspective depth is reconstructed, not surface points. Since an error in depth is considered, the original surfaces from [105] are no suitable choice for the ground truth.

This is resolved in a very simple way: With the same ray-tracer that has been used for generating the input images, ground truth depth maps can be generated by simply writing the length of the traced ray to a file for every pixel. The length of the traced ray is the ground truth depth in this pixel. In the next section, different ground truth surfaces for different focal lengths will be demonstrated for both the vase surface and the Mozart face surface.

2.7 The Influence of the Focal Length

This section deals with the effect that one parameter in the model has on the quality of the reconstruction. To our best of knowledge, this has not been discussed so far in the literature. However, it will give very interesting insight in the stability of the different approaches. The analysis in this section will only be done on Lambertian images, and all experiments will be performed on both optimal control approaches by Prados et al. and Cristiani et al. as well as a method that solves the PDE (2.15) in the special case of $k_a = k_s = 0$ directly, which is then based on the same reflectance model as the two optimal control methods. The method based on said PDE will be called the direct method since it solves a Hamilton-Jacobi equation directly instead of transforming the PDE to a Hamilton-Jacobi-Bellman equation first using the Laplace transform to obtain an optimal control method.

Intuition says that large focal lengths might constitute a problem for SfS, since the influence of the perspective distortion and the light attenuation term become smaller, i.e., the model approaches an orthographic model, which is known to be ill-posed. On the other hand, small focal lengths may cause large amounts of occlusions, which might lead to heavy discontinuities and also increase the difficulty of the reconstruction.

The Vase Experiment

As the first experiment, we will use the vase surface. Figure 2.7 shows input images and the corresponding ground truth surfaces for different focal lengths. Of course, with increasing focal length, the distance between object and camera has to be enlarged as well. With increasing distance, the light source intensity has to be increased quadratically as well. Table 2.1 shows the rendering parameters for the different images. The two key observations in Figure 2.7 are:

- With increasing focal length, the effect of the light attenuation term becomes smaller, contrast decreases. The image approximates an orthographic image.

- At small focal lengths, the ground truth surface becomes more discontinous. This can be observed at the transition between the vase and the background. For small focal lengths, this is significantly steeper compared to larger focal lengths.

Table 2.1: Rendering parameters for the vase experiment with varying focal lengths.

f	Size	k_a	I_a	k_d	I_d	k_s	I_s	α	h_1	h_2
125	128×128	0	0	1	4000	0	0	1	1	1
250	128×128	0	0	1	25000	0	0	1	1	1
500	128×128	0	0	1	100000	0	0	1	1	1
1000	128×128	0	0	1	400000	0	0	1	1	1
2000	128×128	0	0	1	1600000	0	0	1	1	1

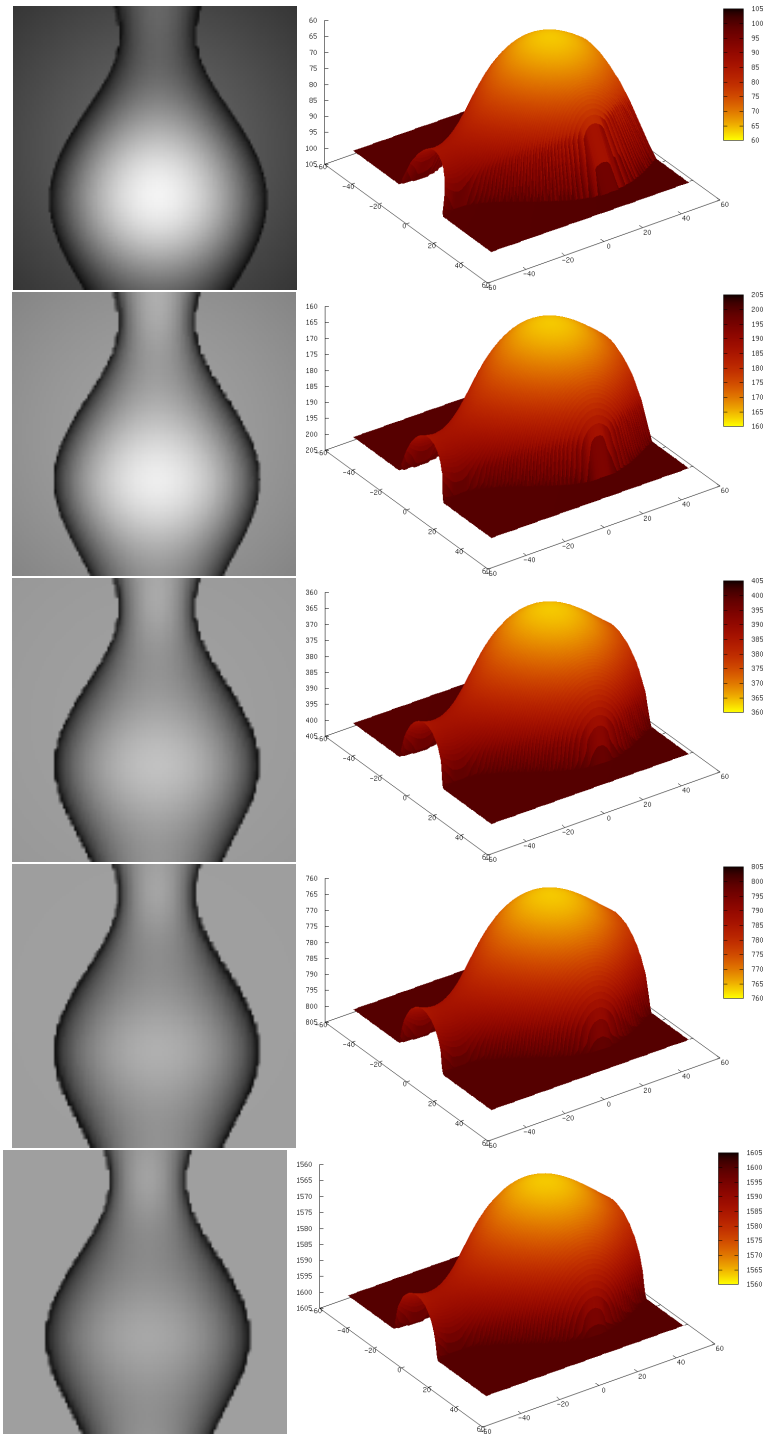

Figure 2.7: Input images and corresponding ground truth surfaces for the vase surface for different focal lengths. From top to bottom: $f = 125$, $f = 250$, $f = 500$, $f = 1000$, and $f = 2000$.

Figures 2.8 and 2.9 show the reconstructions of the images with $f = 125$ and $f = 250$ using the model introduced in this chapter. Note that using the methods of Prados et al. [74] or Cristiani et al. [22] would give visually quite identical results, hence only one result each is shown. Visually these reconstructions look good. The scale is correct, only at the transition between foreground and background the reconstruction is smoothed out a bit. Note that for the higher focal length, the smoothing artefacts at the transition become smaller. This makes sense, since the discontinuity is much less severe in this case.

Figure 2.10 shows the reconstruction for $f = 500$. Here, the reconstruction is even better, but we obtain some oscillations at the transition. This is an effect of the quadratic light attenuation term becoming less important. In this case, however, the effect is not big enough yet to destroy the result. If we look at the reconstructions for $f = 1000$ and $f = 2000$ in Figures 2.11 and 2.12, which are additionally supplied on a larger scale, we observe that the oscillations become dominant. For $f = 1000$, the depth scale of the vase and the background is still correct, however, for $f = 2000$ the vase is estimated on a wrong scale, i.e., behind the background, which is also at a wrong scale. It is possible to avoid this, which will be tackled in a later chapter.

Table 2.2 shows the relative depth errors. The errors reflect the impression of the visual inspection of the results. The smallest errors are obtained for $f = 250$ and $f = 500$, while for smaller focal lengths, the error increases, and for higher focal lengths, the error increases dramatically. This is caused by problems arising at the boundary between vase and background. For high focal lengths, the attenuation term plays only a very small role, and hence grey values in the image have more meaning as angles, while for small focal lengths, the attenuation term plays a significant role as well. The massive errors for the high focal lengths also reflect the visual impression.

Table 2.3 shows the maximal depth errors, i.e. the error of the worst outlier. For the small focal lengths, we see no really bad outliers. However, starting with $f = 500$, which is around four times the image size, the first massive outliers appear. As we can see in Figure 2.10, these outliers are only in very few spots, which results in the overall error and visual quality being rather good. At the high focal lengths, however, these outliers become so frequent that they heavily influence the error and the visual quality of the reconstruction.

However, the relative depth error is not a good measure for this experiment. Since the range of the z-values of the reconstruction remains the same for all experiments, but the distance increases with different focal

Table 2.2: Average depth errors of the vase experiment for different focal lengths.

Focal length f	Direct Method	Prados et al.	Cristiani et al.
125	3.86%	3.81%	3.44%
250	1.40%	1.42%	1.13%
500	1.80%	1.91%	1.72%
1000	9.67%	9.68%	9.68%
2000	12.98%	12.98%	12.98%

Table 2.3: Maximal depth errors of the vase experiment for different focal lengths.

Focal length f	Direct Method	Prados et al.	Cristiani et al.
125	18.23%	18.15%	16.05%
250	5.89%	5.88%	5.12%
500	240.5%	240.5%	240.5%
1000	251.5%	251.5%	251.5%
2000	216.7%	216.7%	216.7%

lengths, a 1% depth error is more meaningful for large focal lengths than for small ones. Therefore, a better measure for this kind of comparison is to multiply the relative depth error with the distance, or, since the distance is proportional to the focal length, the focal length. This error measure is shown in Table 2.4. For this error measure, the failure of the method at high focal lengths becomes even more obvious, while the negative effect of the discontinuity at small focal lengths is reduced. Obviously, at $f = 250$, the error is by far the best. In conclusion, for these standard methods, a small focal length is generally beneficial for the reconstruction quality, unless discontinuities become too prominent. However, note that in Chapter 3, a numerical method will be introduced that can handle these difficulties. It will still fail at the transition between vase and background, but be able to estimate the background correctly. Since the transition boundary only represents a small fraction of the image, this method will produce very small errors, at least compared to the ones presented here.

Table 2.4: Scale-corrected depth errors of the vase experiment for different focal lengths.

Focal length f	Direct Method	Prados et al.	Cristiani et al.
125	4.825	4.78	4.3
250	3.50	3.55	2.83
500	9.00	9.55	8.60
1000	96.7	96.8	96.8
2000	260	260	260

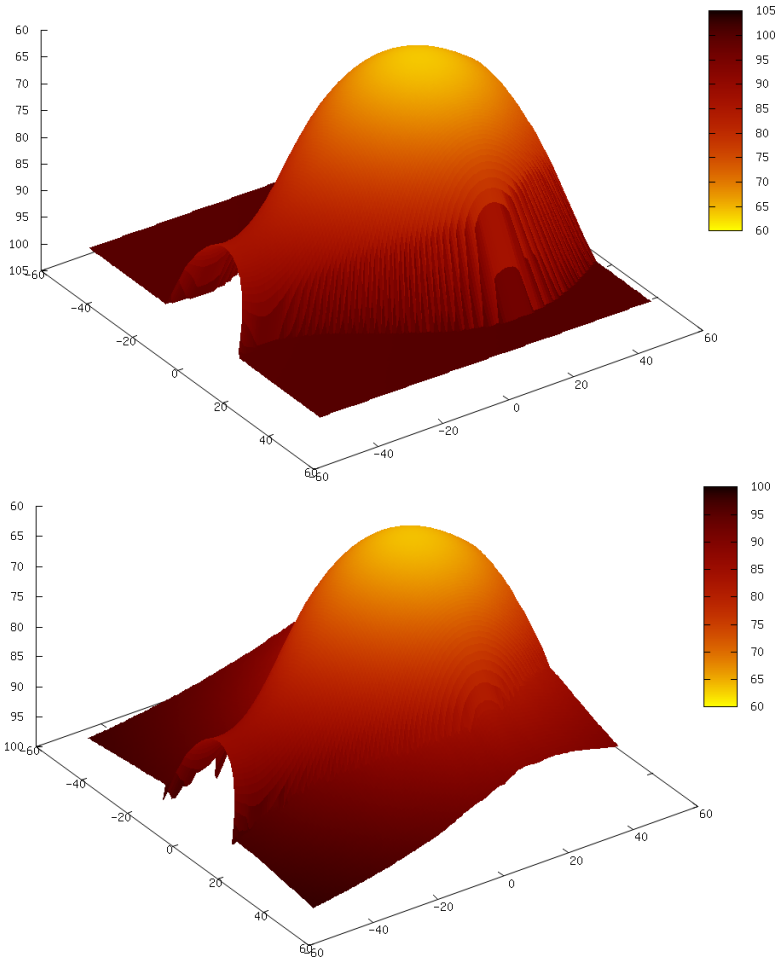

Figure 2.8: Reconstruction of the Lambertian vase using $f = 125$. Top: Ground Truth. Bottom: Reconstruction.

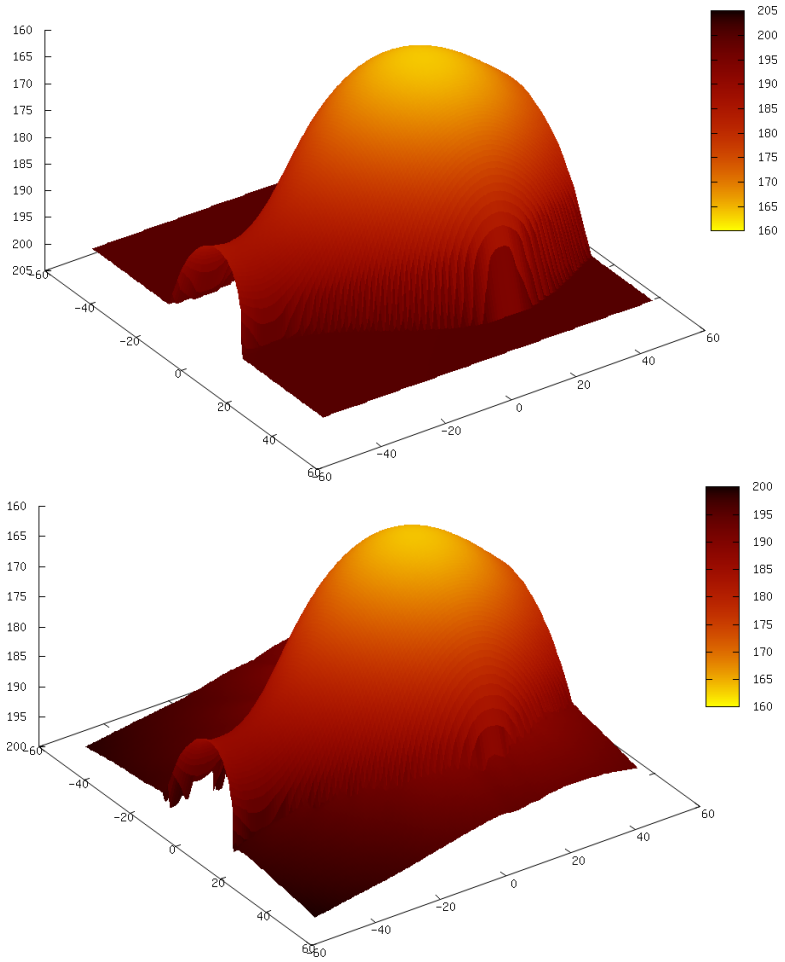

Figure 2.9: Reconstruction of the Lambertian vase using $f = 250$. Top: Ground Truth. Bottom: Reconstruction.

Table 2.5: Maximal scale-corrected depth errors of the vase experiment for different focal lengths.

Focal length f	Direct Method	Prados et al.	Cristiani et al.
125	22.5	22.68	20.06
250	14.70	14.70	12.80
500	1202	1202	1202
1000	2515	2515	2515
2000	4334	4334	4334

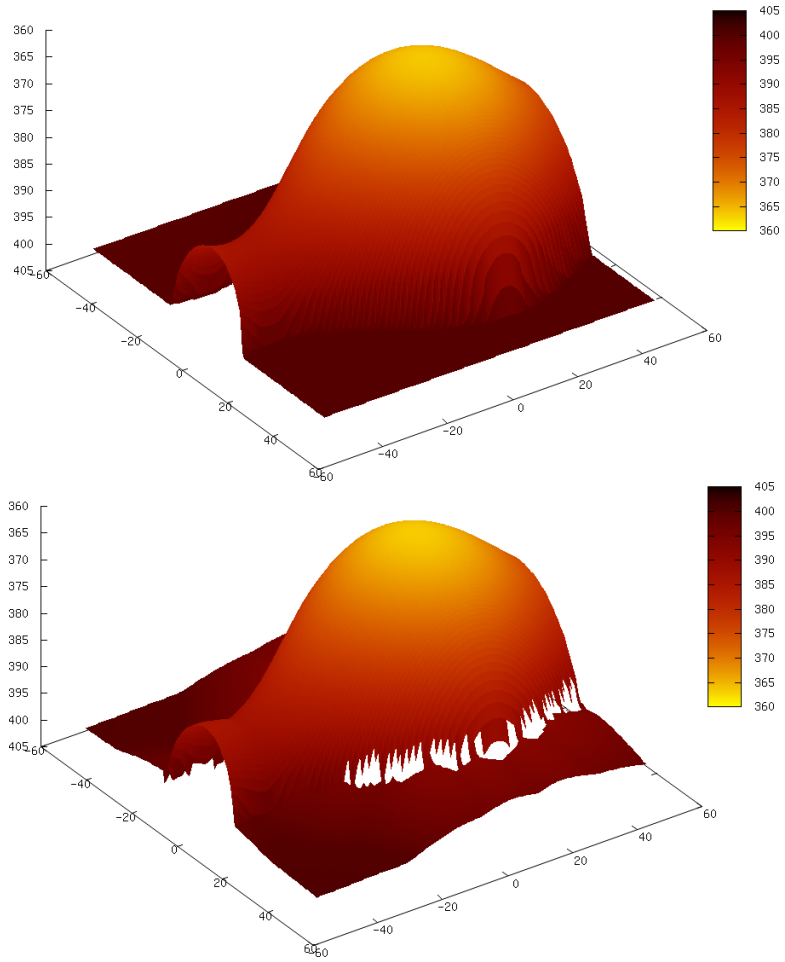

Figure 2.10: Reconstruction of the Lambertian vase using $f = 500$. Top: Ground Truth. Bottom: Reconstruction.

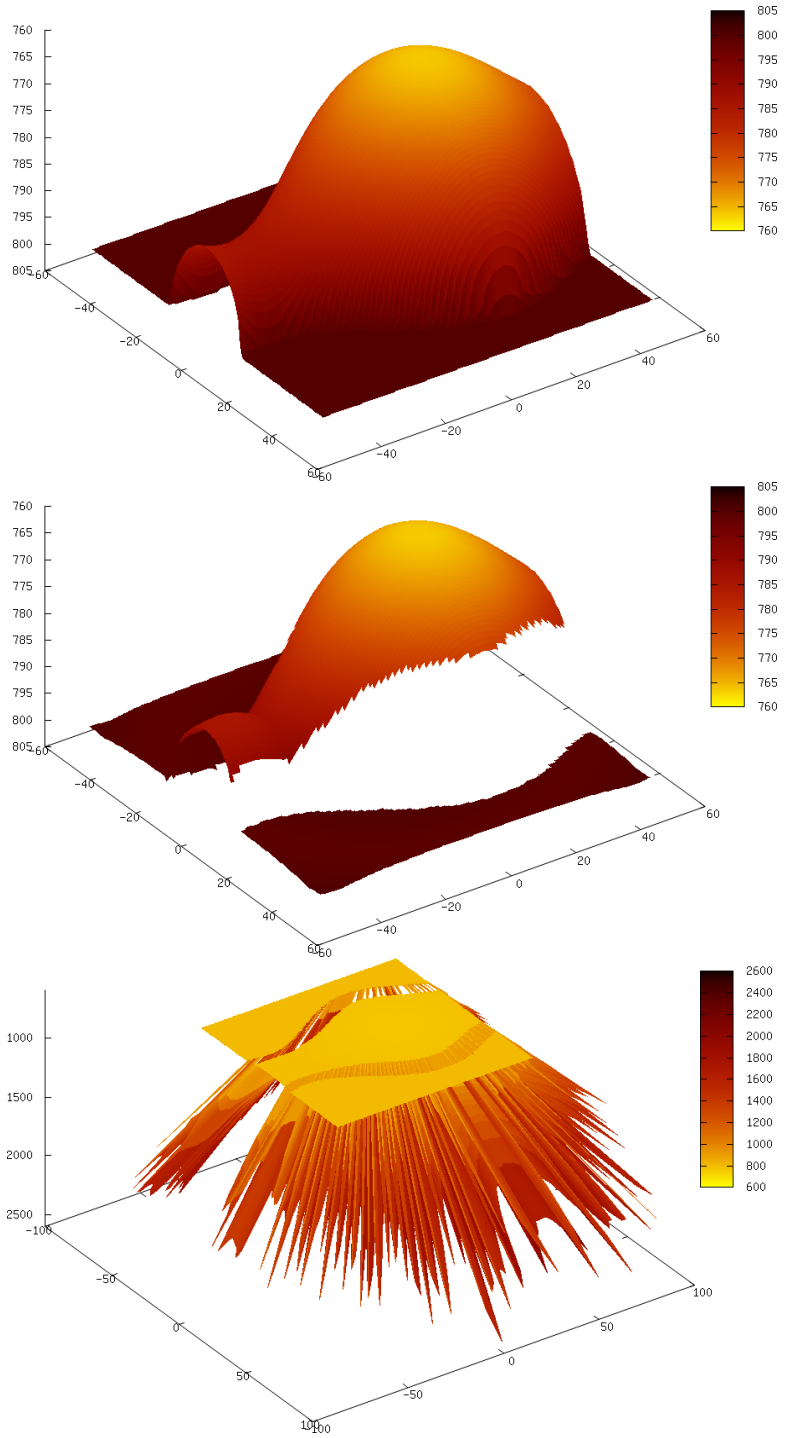

Figure 2.11: Reconstruction of the Lambertian vase using $f = 1000$. Top: Ground Truth. Middle: Reconstruction. Bottom: Reconstruction on a larger scale.

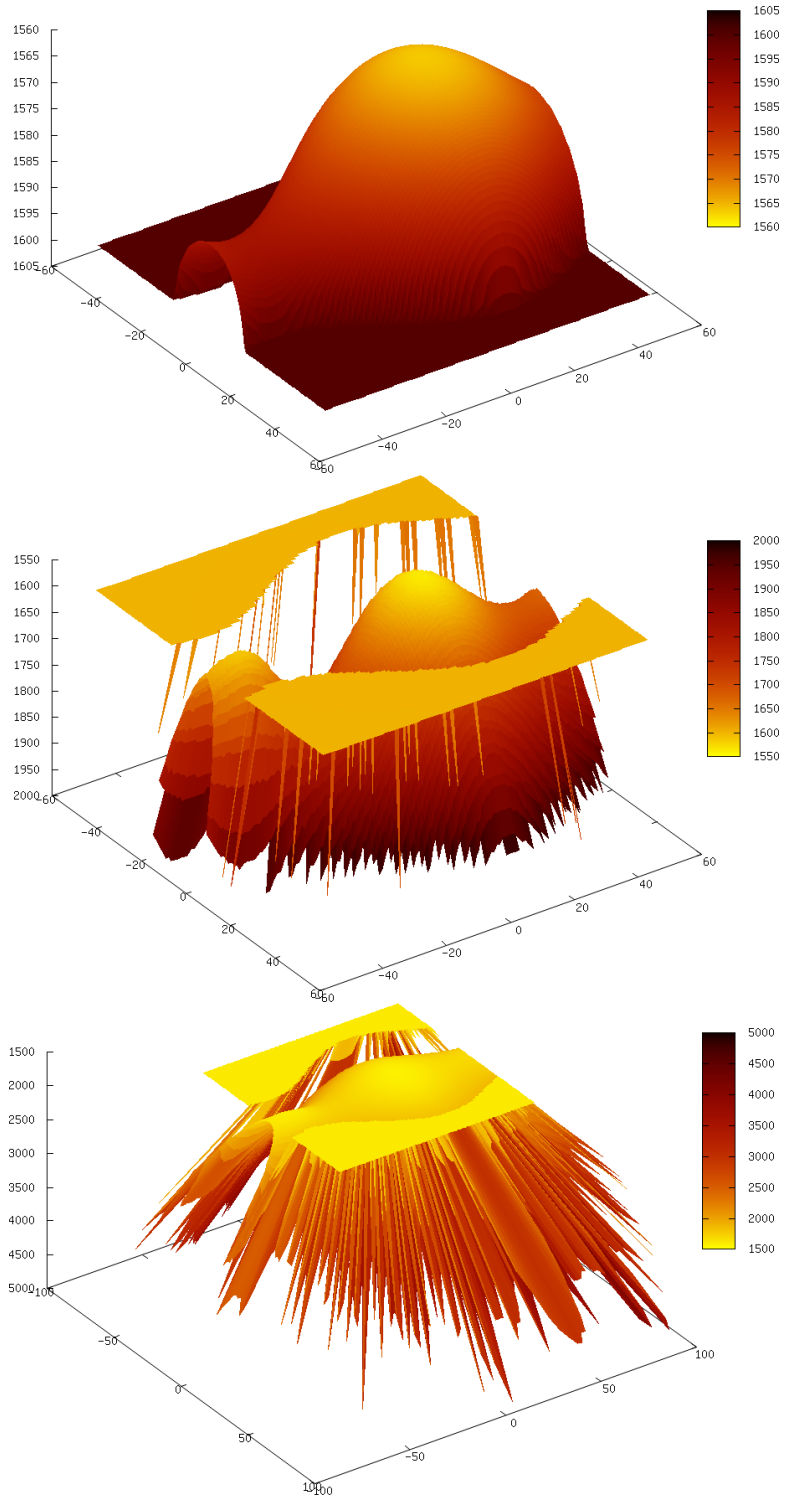

Figure 2.12: Reconstruction of the Lambertian vase using $f = 2000$. Top: Ground Truth. Middle: Reconstruction. Bottom: Reconstruction on a larger scale.

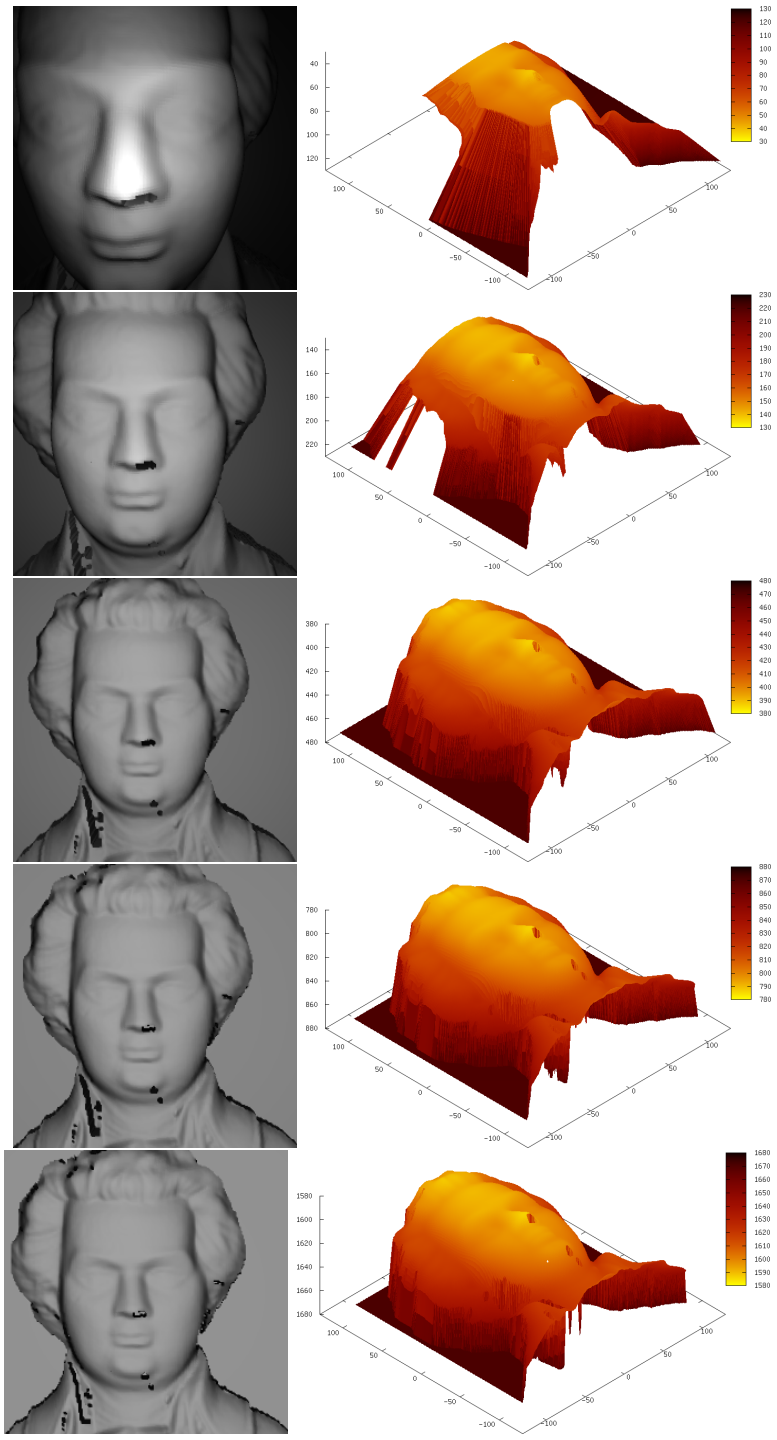

Figure 2.13: Input images and corresponding ground truth surfaces for the Mozart face surface for different focal lengths. From top to bottom: $f = 125$, $f = 250$, $f = 500$, $f = 1000$, and $f = 2000$.

The Mozart Experiment

Up to this point, the experimental evaluation shows a clear tendency that images with small focal length are easier to reconstruct than images with larger focal lengths. To support this, the same analysis is done on a more complex image. Figure 2.13 shows the input images for the Mozart face with different focal lengths. Table 2.6 shows the corresponding rendering parameters. Since this image is with 256×256 pixels twice the size of the vase image in every direction, $f = 125$ is actually very very close to the surface, which results in most of the surface being occluded. This becomes obvious in the ground truths in Figure 2.13.

Figure 2.14 shows the reconstruction of the surface for $f = 125$. In fact, even more of the surface is included in the reconstruction than it is in the ground truth, but the overall shape looks nice. The shape for $f = 250$ in Figure 2.15, however, is mostly visible and the shape is very close to the original. Unfortunately, despite the image is significantly larger than the vase, the first outliers occur again at $f = 500$, cf. Figure 2.16. The shape of the remainder of the reconstruction, however, is still flawless. The amount of outliers increases at $f = 1000$, cf. Figure 2.17, where the shape of the remainder is still very well reconstructed. However, at $f = 2000$ the amount of outliers is so massive that the appearance of Mozart's face is now far from what a face is expected to look like.

Table 2.6: Rendering parameters for the Mozart face experiment with varying focal lengths.

Focal length	Size	k_a	I_a	k_d	I_d	k_s	I_s	α	h_1	h_2
125	256×256	0	0	1	1800	0	0	1	1	1
250	256×256	0	0	1	15000	0	0	1	1	1
500	256×256	0	0	1	100000	0	0	1	1	1
1000	256×256	0	0	1	400000	0	0	1	1	1
2000	256×256	0	0	1	1600000	0	0	1	1	1

This is supported by the scale-corrected average and maximal depth errors shown in Tables 2.7 and. Again, the scale-corrected error becomes the lowest at $f = 250$. At $f = 500$, the first outliers occur, and the errors above $f = 1000$ outliers become so dominant that the reconstruction fails.

Table 2.7: Scale-corrected depth errors of the Mozart face experiment for different focal lengths.

Focal length f	Direct Method	Prados et al.	Cristiani et al.
125	9.737	9.876	10.08
250	10.30	10.33	9.54
500	24.14	24.29	23.83
1000	102.25	103.15	102.88
2000	264.88	264.88	264.88

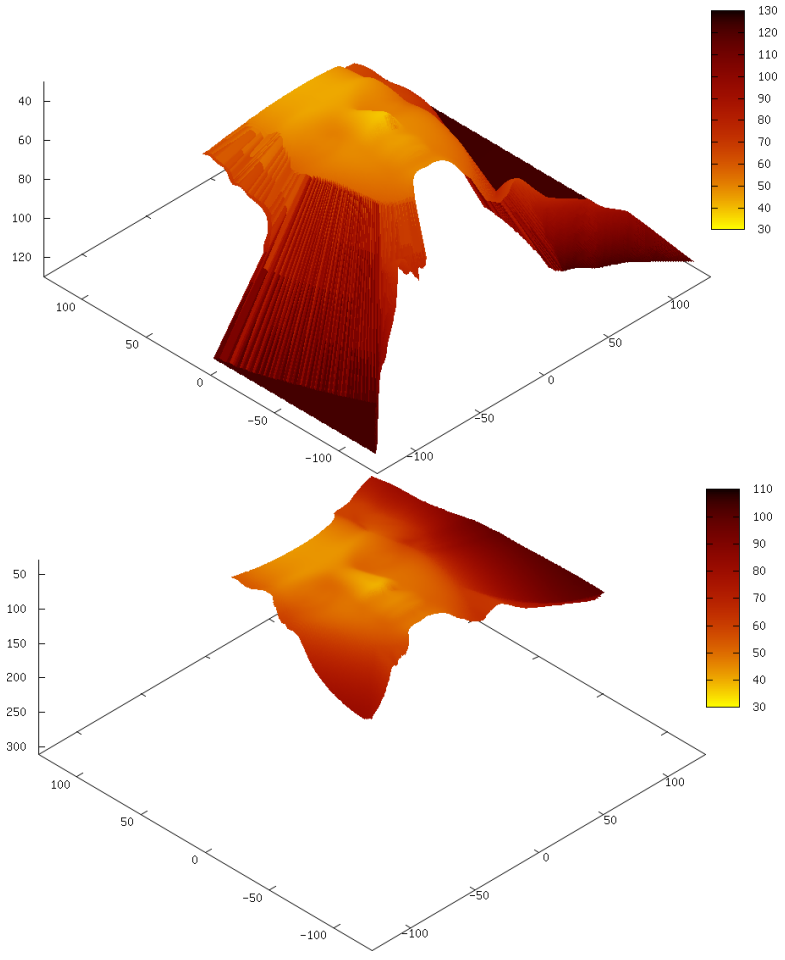

Figure 2.14: Reconstruction of the Lambertian Mozart face using $f = 125$. Top: Ground Truth. Bottom: Reconstruction.

Table 2.8: Maximal scale-corrected depth errors of the Mozart face experiment for different focal lengths.

Focal length f	Direct Method	Prados et al.	Cristiani et al.
125	66.28	66.31	64.74
250	83.91	83.89	86.01
500	1513	1513	1513
1000	2966	2961	2961
2000	5942	5942	5942

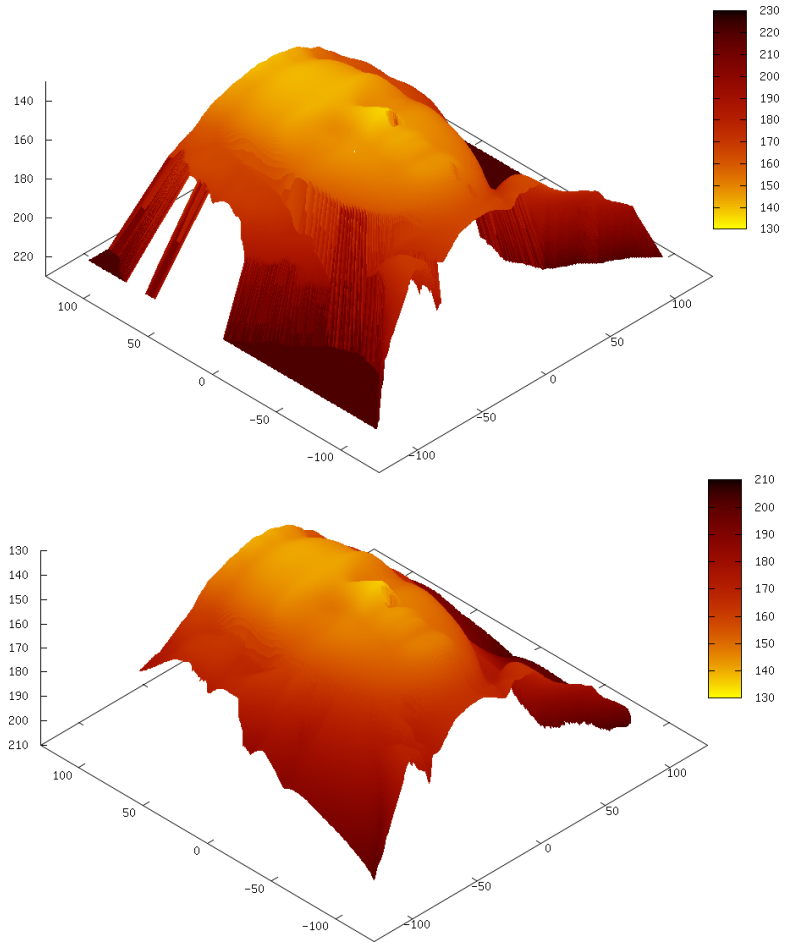

Figure 2.15: Reconstruction of the Lambertian Mozart face using $f = 250$. Top: Ground Truth. Bottom: Reconstruction.

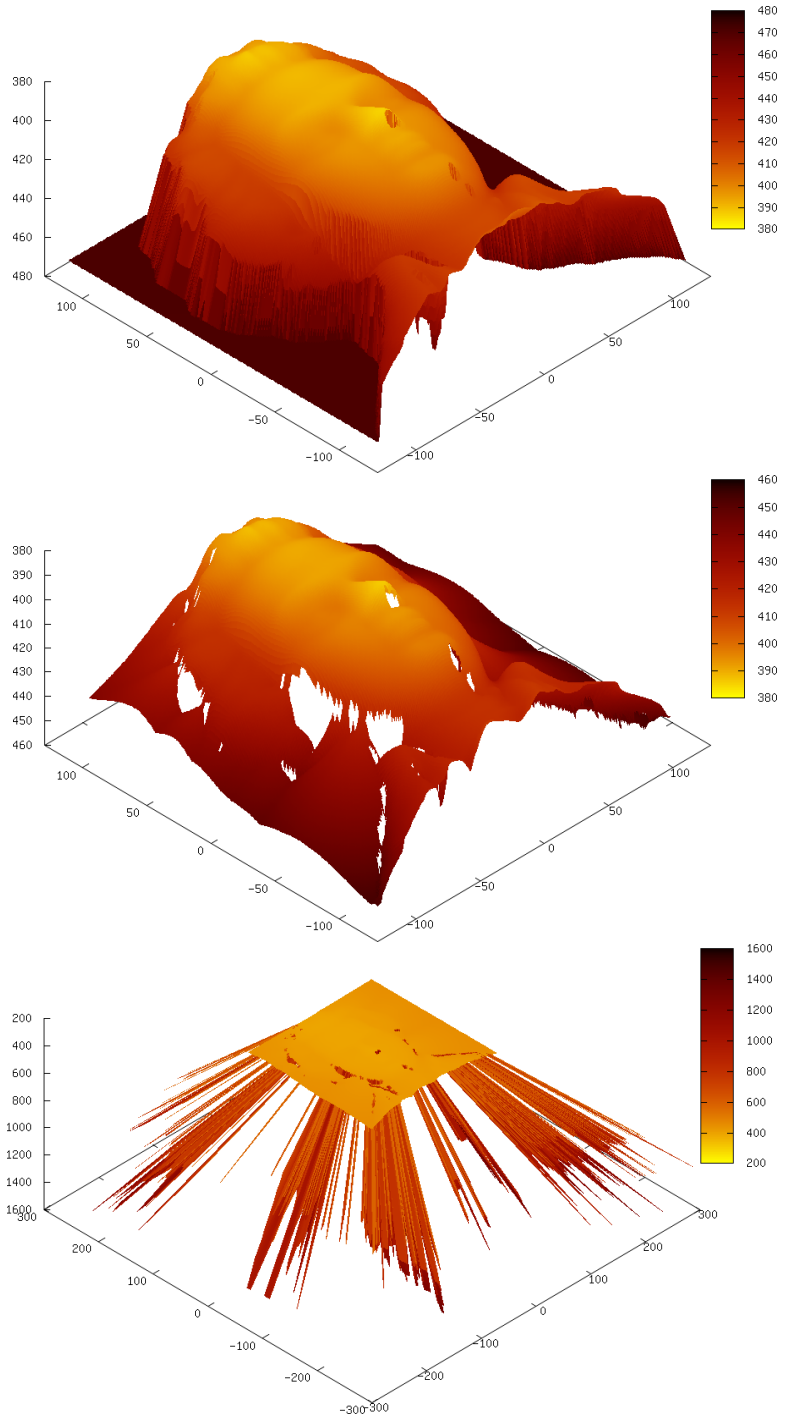

Figure 2.16: Reconstruction of the Lambertian Mozart face using $f = 500$. Top: Ground Truth. Middle: Reconstruction. Bottom: Reconstruction on a larger scale.

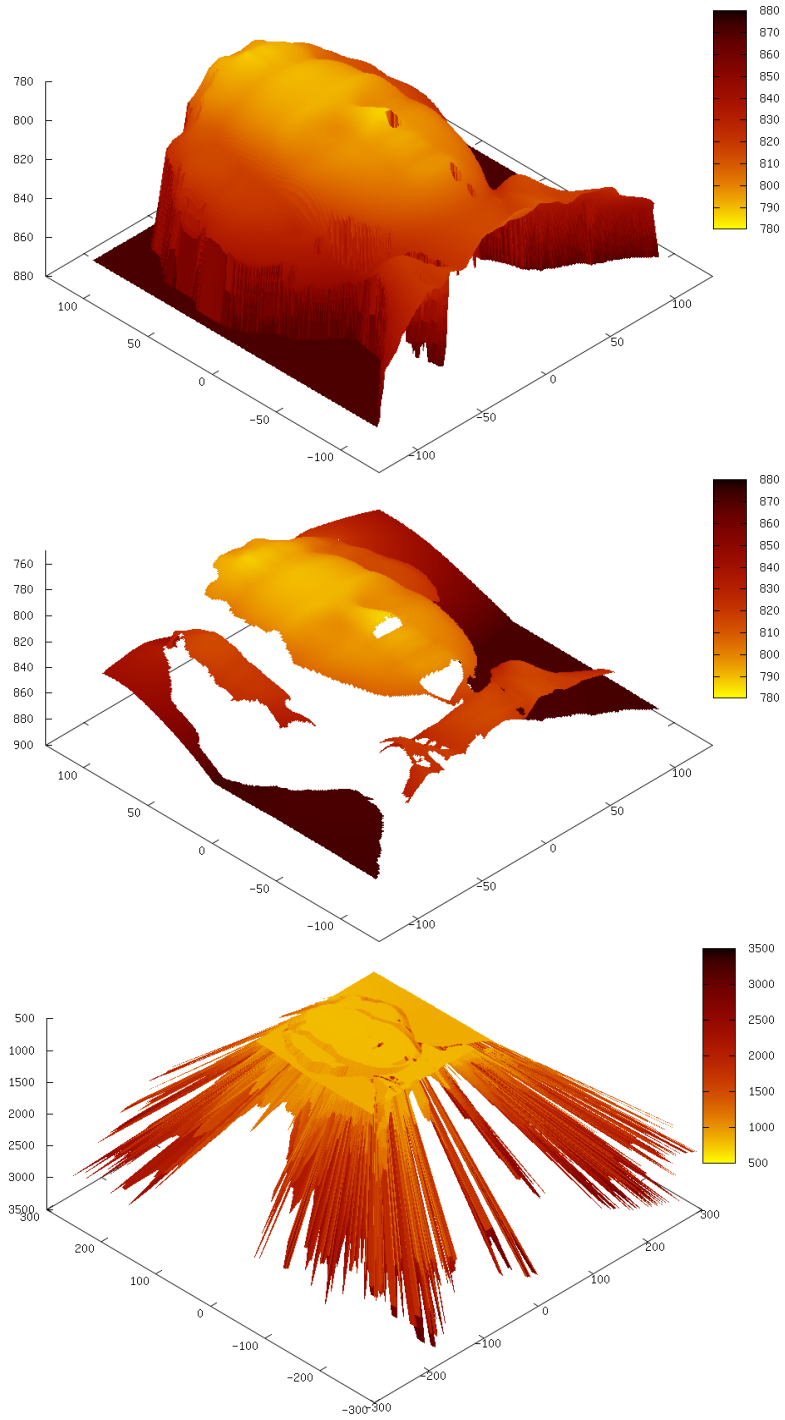

Figure 2.17: Reconstruction of the Lambertian Mozart face using $f = 1000$. Top: Ground Truth. Middle: Reconstruction. Bottom: Reconstruction on a larger scale.

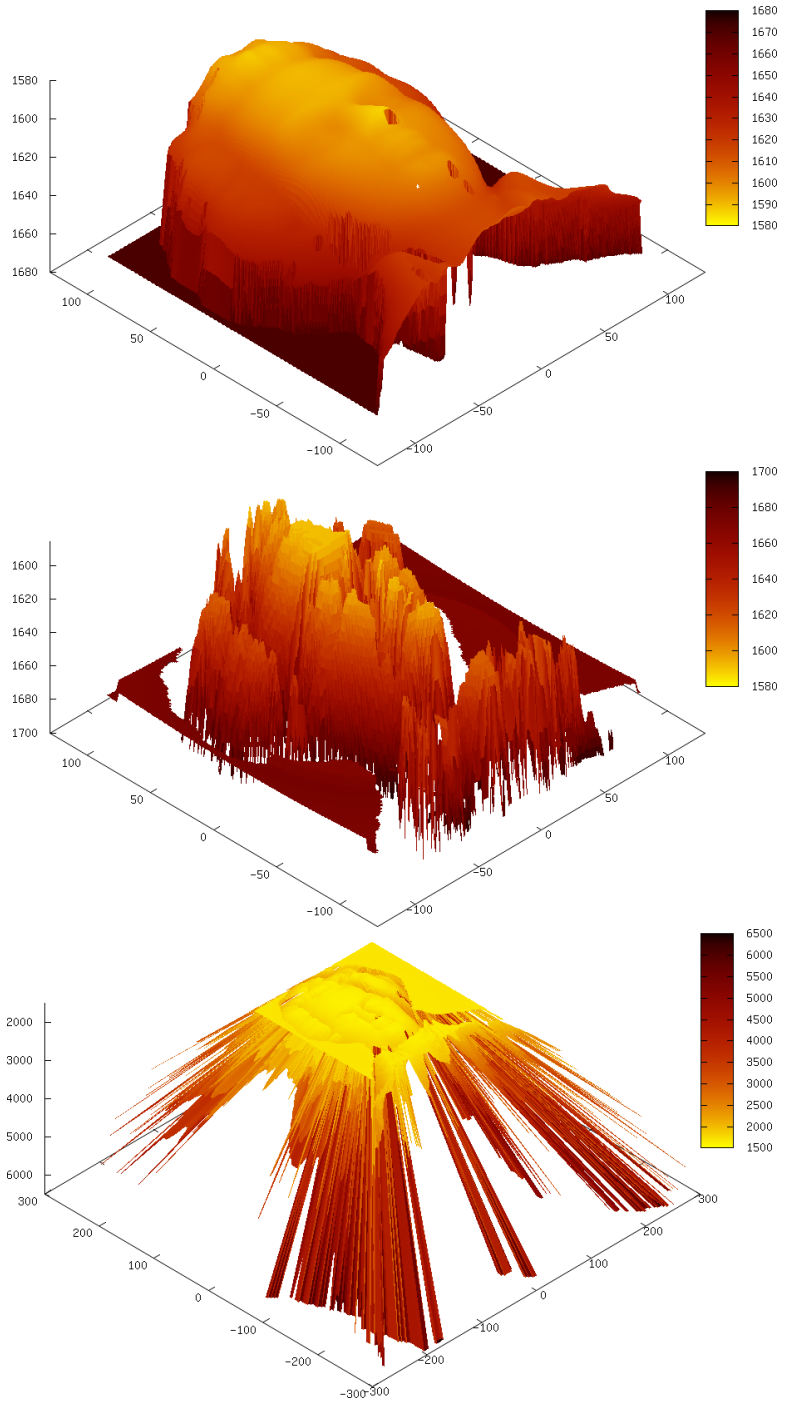

Figure 2.18: Reconstruction of the Lambertian Mozart face using $f = 2000$. Top: Ground Truth. Middle: Reconstruction. Bottom: Reconstruction on a larger scale.

2.8 The Influence of the Specular Term

In this paragraph, the influence of the specular term on the reconstruction quality will be assessed. In contrast to the previous section, comparison with the models of Prados et al. [74] and Cristiani et al. [22] does not make sense, since these models are purely Lambertian and there exists no extension to their PDEs which would include specular highlights. However, it does make sense to compare the reconstruction result using the Phong model to results using no specularity at all, since this will show the importance of modelling specular highlights. From a different point of view, this approach shows why Lambertian models fail for images that contain specular highlights. This will also be addressed for real-world images, but in a later chapter. In this chapter, the focus lies on synthetic images to demonstrate the technical soundness of the model and the effect of different types of specular highlights.

The Vase Experiment

As before, the first experiment will be the vase image. This time, a fixed focal length is used, but different shading parameters. There are two relevant parameters:

- The ratio between the specular and the diffuse term. In the experiment, the starting point will be the purely diffuse case shown in the previous paragraph. Then, the importance of the specular part will be subsequently increased.

- The second parameter is the exponent α in the specular term, which describes the tightness of the specular highlight. All experiments will be done for two different choices of this parameter to investigate its effect.

Since the ground truth solely depends on the focal length and the camera position, there is no need to render new ground truths, since this has already been done in the previous section, cf. Figure 2.7. For this experiment the $f = 500$ case is chosen, since at this focal length the first artifacts at the transition between foreground and background occured. This way, it is possible to investigate whether specular highlights have any positive or negative effect on this.

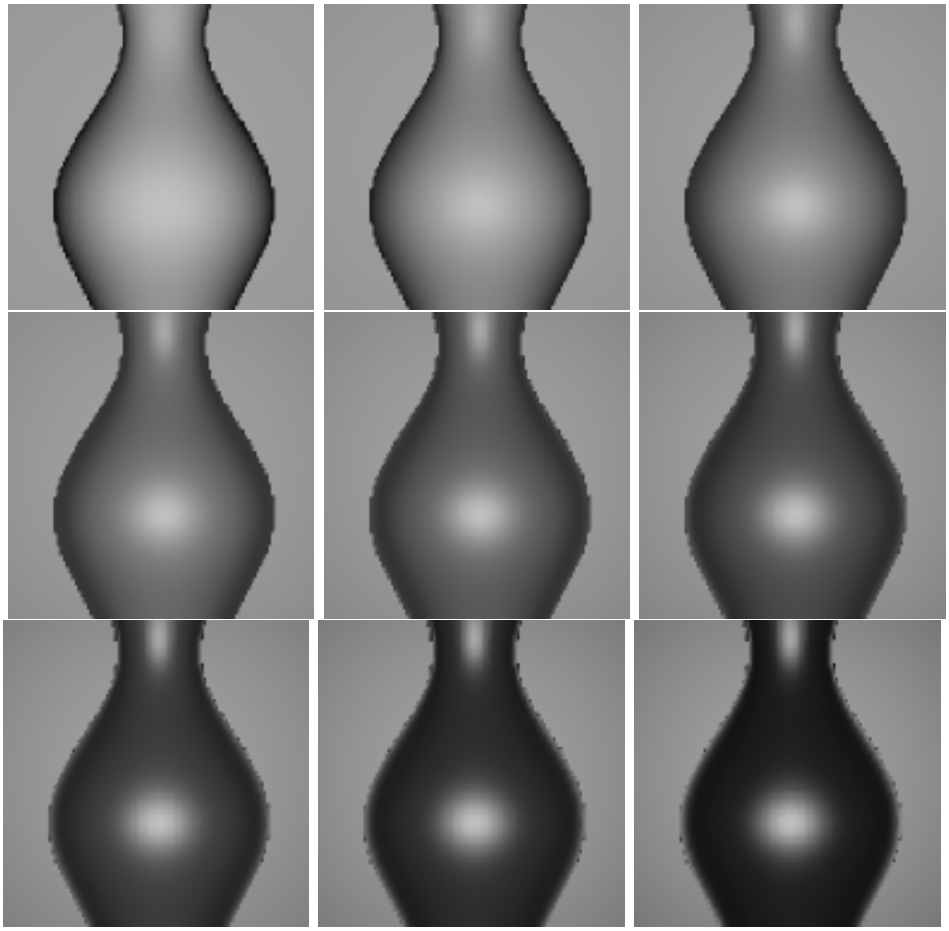

Figure 2.19: Input images of the vase surface with different specular terms. For all images holds $\alpha = 5$ and $k_d = 1 - k_s$. Top left to bottom right: $k_s = 0$, $k_s = 0.1$, $k_s = 0.2$, $k_s = 0.3$, $k_s = 0.4$, $k_s = 0.3$, $k_s = 0.6$, $k_s = 0.7$, $k_s = 0.8$

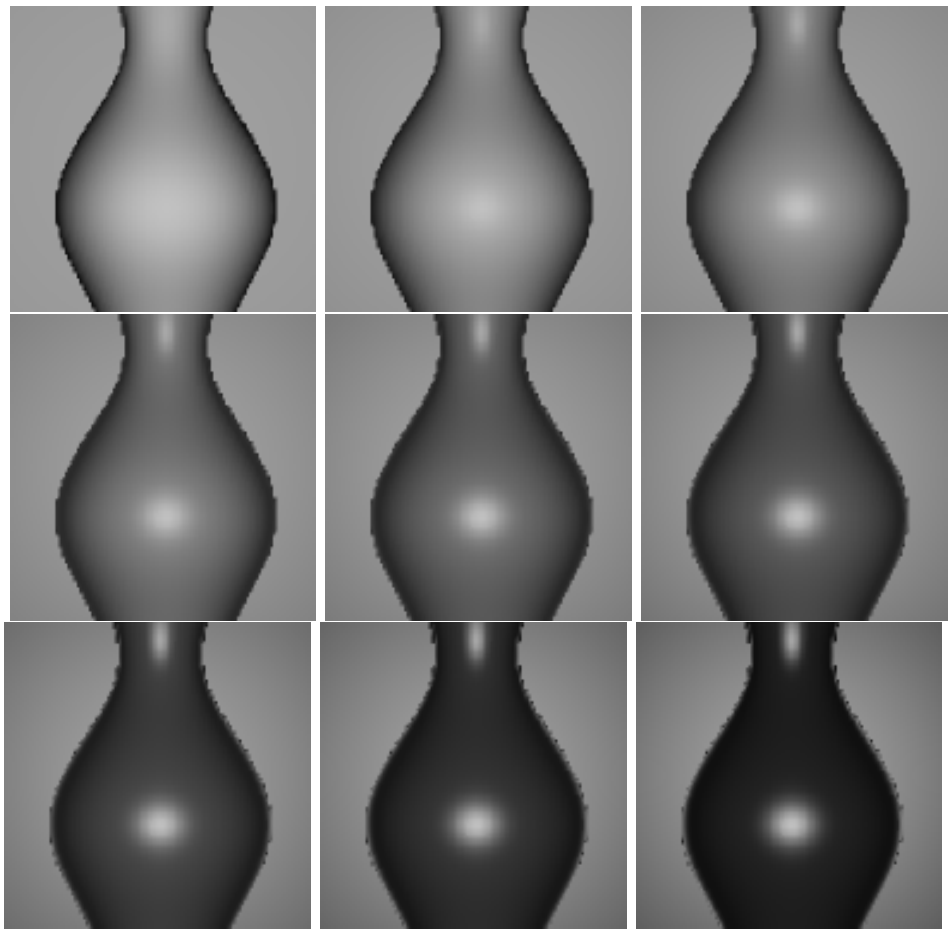

Figure 2.20: Input images vase surface with different specular terms. For all images holds $\alpha = 10$ and $k_d = 1 - k_s$. Top left to bottom right: $k_s = 0$, $k_s = 0.1$, $k_s = 0.2$, $k_s = 0.3$, $k_s = 0.4$, $k_s = 0.3$, $k_s = 0.6$, $k_s = 0.7$, $k_s = 0.8$

Table 2.9: Rendering parameters for the Vase experiment with varying specularities.

f	Size	k_a	I_a	k_d	I_d	k_s	I_s	α	h_1	h_2
500	128×128	0	0	1.0	100000	0.0	100000	5	1	1
500	128×128	0	0	0.9	100000	0.1	100000	5	1	1
500	128×128	0	0	0.8	100000	0.2	100000	5	1	1
500	128×128	0	0	0.7	100000	0.3	100000	5	1	1
500	128×128	0	0	0.6	100000	0.4	100000	5	1	1
500	128×128	0	0	0.5	100000	0.5	100000	5	1	1
500	128×128	0	0	0.4	100000	0.6	100000	5	1	1
500	128×128	0	0	0.3	100000	0.7	100000	5	1	1
500	128×128	0	0	0.2	100000	0.8	100000	5	1	1
500	128×128	0	0	1.0	100000	0.0	100000	10	1	1
500	128×128	0	0	0.9	100000	0.1	100000	10	1	1
500	128×128	0	0	0.8	100000	0.2	100000	10	1	1
500	128×128	0	0	0.7	100000	0.3	100000	10	1	1
500	128×128	0	0	0.6	100000	0.4	100000	10	1	1
500	128×128	0	0	0.5	100000	0.5	100000	10	1	1
500	128×128	0	0	0.4	100000	0.6	100000	10	1	1
500	128×128	0	0	0.3	100000	0.7	100000	10	1	1
500	128×128	0	0	0.2	100000	0.8	100000	10	1	1

To see the importance of including specular highlights in the model, all experiments will be compared to the reconstruction obtained by using a purely Lambertian model.

Figure 2.19 depicts the input images of the vase for this experiment for $\alpha = 5$, 2.20 for $\alpha = 10$. Table 2.9 illustrates the detailed rendering parameters. Note that for all experiments, $k_a = 0$ is chosen. Since this is a synthetic experiment, the ambient light would be just an additive constant to the brightness of every pixel, which would be subtracted from every pixel by the algorithm. Using ambient light only makes sense for real-world images, where some ambient light is present in the scene. Since the focus of this section lies on synthetic images, k_a is always set to 0 here.

Note that, while the vase looks like a simple experiment, this is actually a quite difficult surface to reconstruct. The focus lies on the surface for $f = 500$, since this is an order of magnitude for the focal length that is similar to real-world applications, compared to the small image size. In the previous paragraph this image turned out to be the first image where massive outliers near the boundary occur. One of the issues in this chapter

is the effect of specular highlights on such outliers.

Visual Impression

Before turning to numerical results, the visual impression of the results is certainly a key thing for SfS applications. While it is possible to argue which error measure is the right one for the reconstruction of a surface, it is rather simple to judge visually whether a reconstruction is good or bad, in particular if it is possible to compare it to the ground truth. The result of the base case, $k_s = 0$, has been already presented in the previous paragraph, cf. Figure 2.10. Since this is a purely Lambertian experiment, the effect of the specular term obviously cannot be observed here.

Figures 2.21-2.28 visualise the reconstructions using both the Lambertian and the Phong model for the input images with different specularities and $\alpha = 5$. Figures 2.29-2.36 depict the respective reconstructions for $\alpha = 10$.

The first couple of images for $\alpha = 5$ are very similar to the original shape. In contrast to the reconstruction of the Lambertian image, which is illustrated in in Figure 2.16, the outliers seem to have disappeared. At least visually, this continues for increasing amounts of specular highlights up to $k_s = 0.5$, where the first outliers at the boundary appear again. Then, with further increasing k_s, this effect is getting worse. At $k_s = 0.8$, the effect is very large and large parts of the surface are missing in the Phong reconstruction.

The Lambertian reconstruction generally seem to be a bit less sensitive to outliers at the boundary than the Phong reconstruction. However, for these images, it is possible to observe another very drastic effect: All of the reconstructions are much too close to the camera and much too steep. The Lambertian reconstructions quickly lose their shape, introducing very high gradients and being much too close to the camera. This is caused by the specular highlights, which are misinterpreted by the algorithm as close points, since a key model ingredient is the light attenuation term, which causes brighter points to be interpreted as close points. Without the specular correction term in the Phong model, this causes much too close and too steep reconstructions.

For $\alpha = 10$, the observed effects become even stronger. The Lambertian reconstructions are all too close, closer for increasing k_s and even steeper in their shape than for $\alpha = 5$. In addition to that effect, also outliers at the transition between foreground and background appear more frequently for the higher exponent. For the Phong reconstruction, similar effects appear. Again, the shape is preserved much more accurately, and in the begin-

ning outliers at the transition between foreground and background do not appear, in contrast to the Lambertian image. However, for images where specular highlights dominate, also the outliers at the boundary become very dominant. For the most extreme image with $k_s = 0.8$ and $\alpha = 10$, shown in Figure 2.36, only very few parts of the vase remain. Without knowing the input image, it would be hard to figure out what the surface is supposed to be. This effect is similar for the Lambertian reconstruction, but not quite as bad.

In general, the Phong reconstructions seem a lot more convincing than the ones using the Lambertian model, which indicates that the modelling of specular highlights makes sense for SfS. However, the impression of the reconstruction is disturbed by the presence of many outliers for images with dominant specular highlights. In a later chapter, though, a numerical method will be presented which enables to reduce these outliers drastically and give convincing results even for these extreme images.

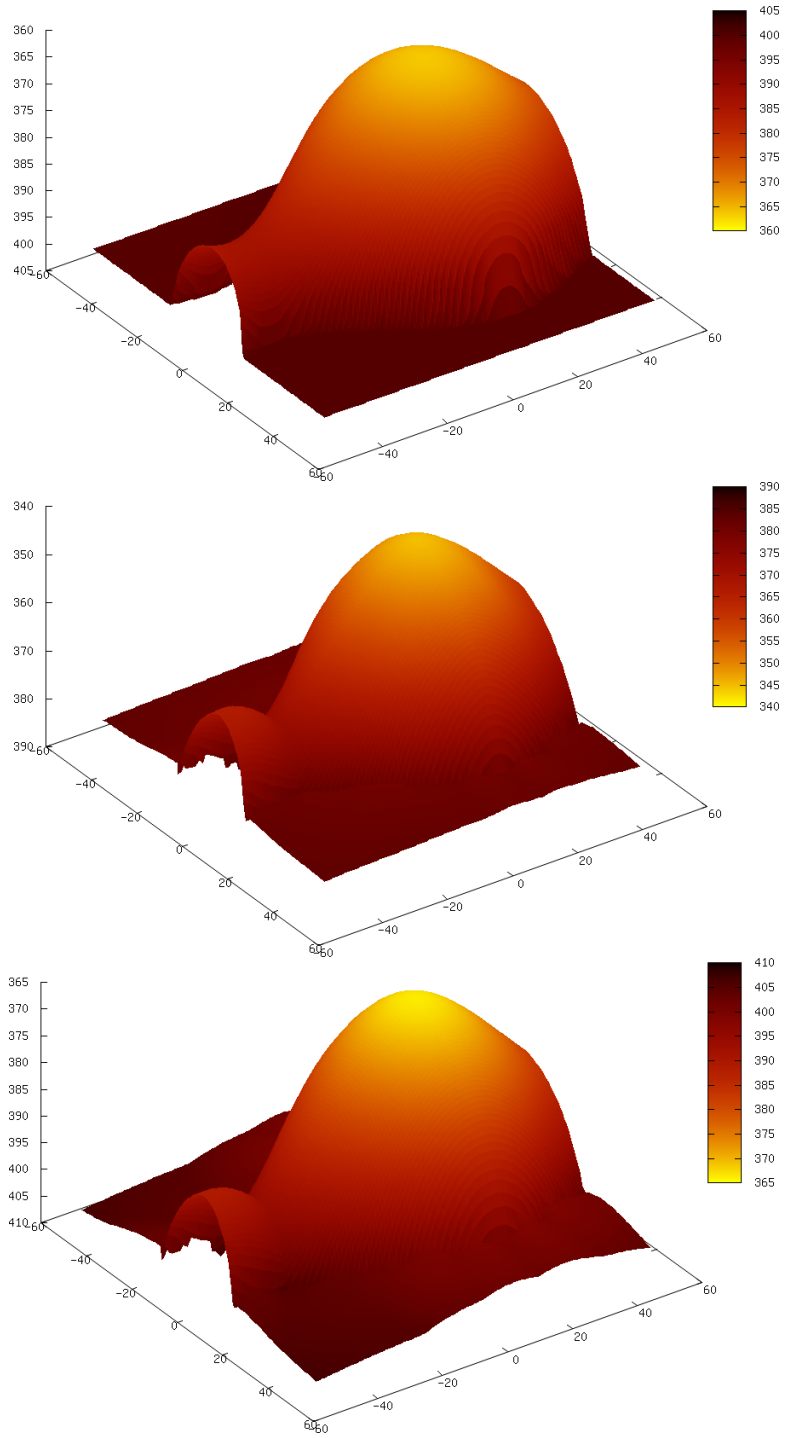

Figure 2.21: Reconstruction of the vase for $k_d = 0.9, k_s = 0.1, \alpha = 5$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

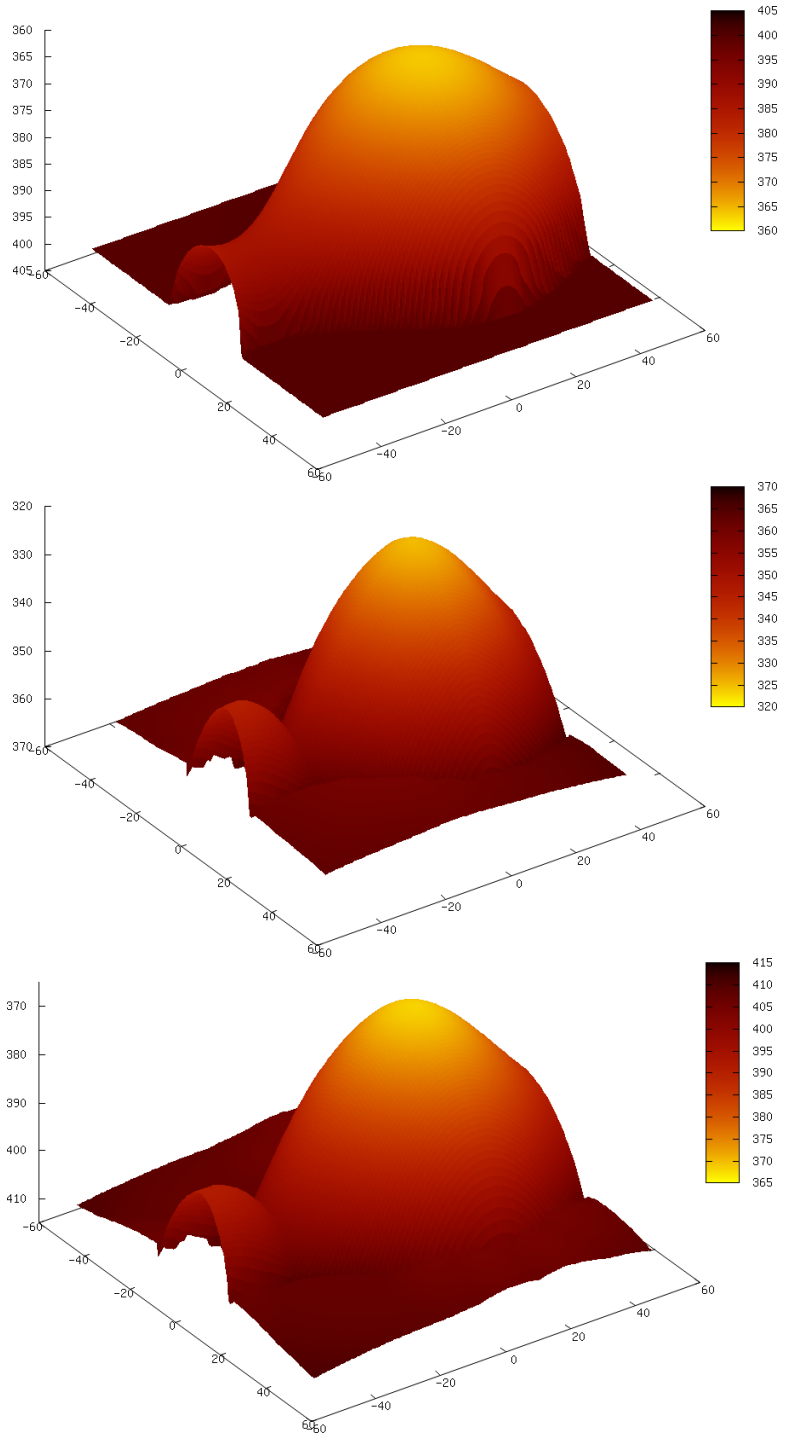

Figure 2.22: Reconstruction of the vase for $k_d = 0.8, k_s = 0.2, \alpha = 5$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

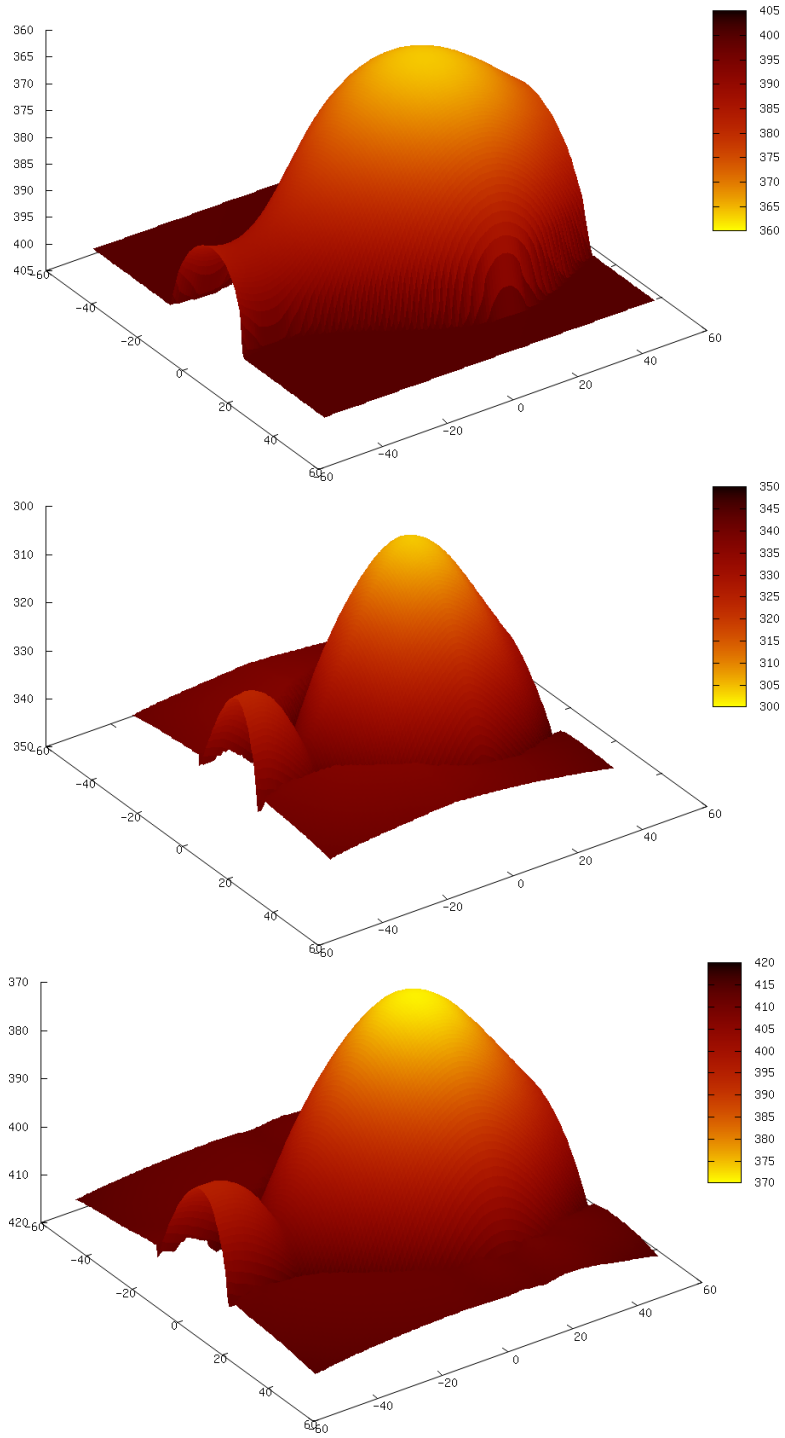

Figure 2.23: Reconstruction of the vase for $k_d = 0.7, k_s = 0.3, \alpha = 5$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

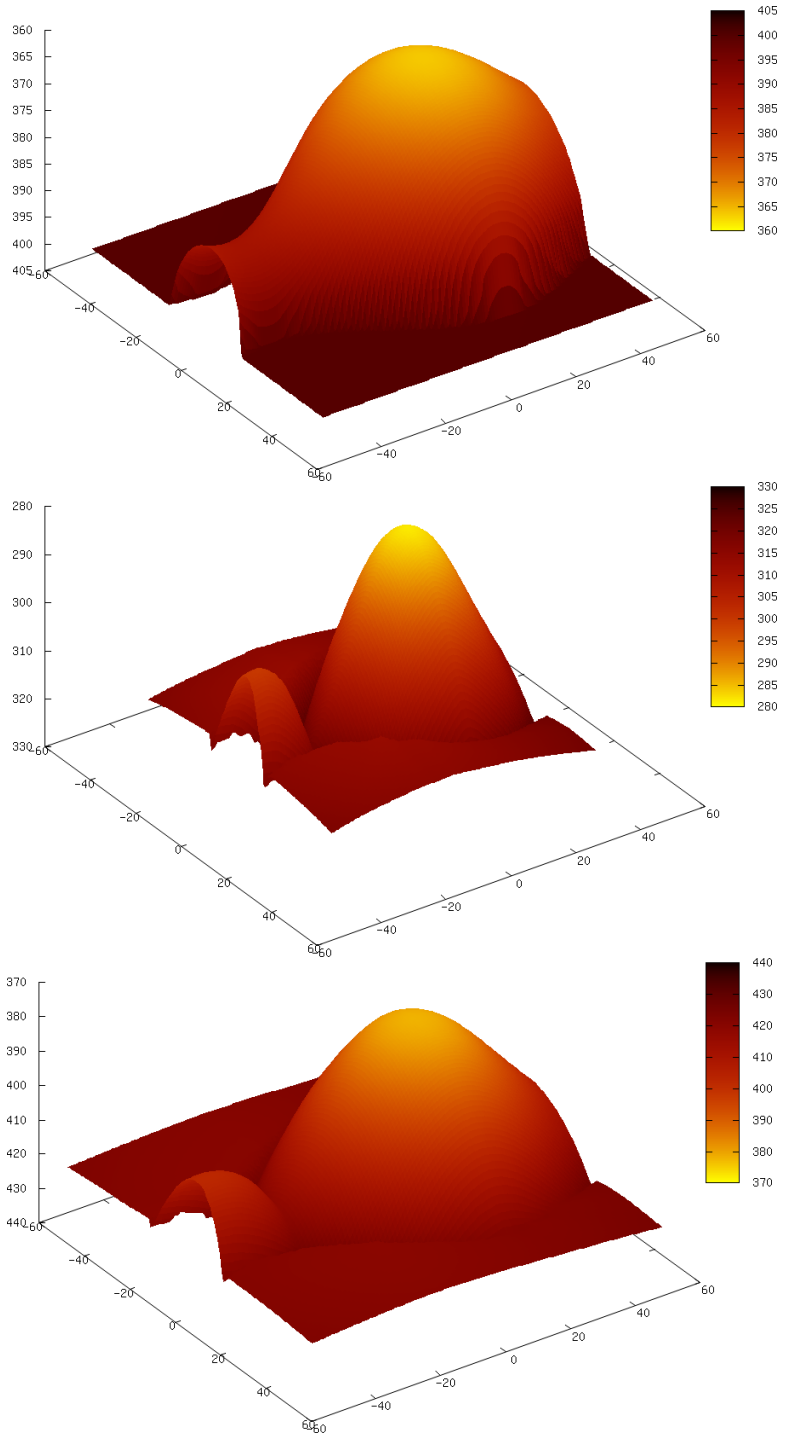

Figure 2.24: Reconstruction of the vase for $k_d = 0.6, k_s = 0.4, \alpha = 5$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

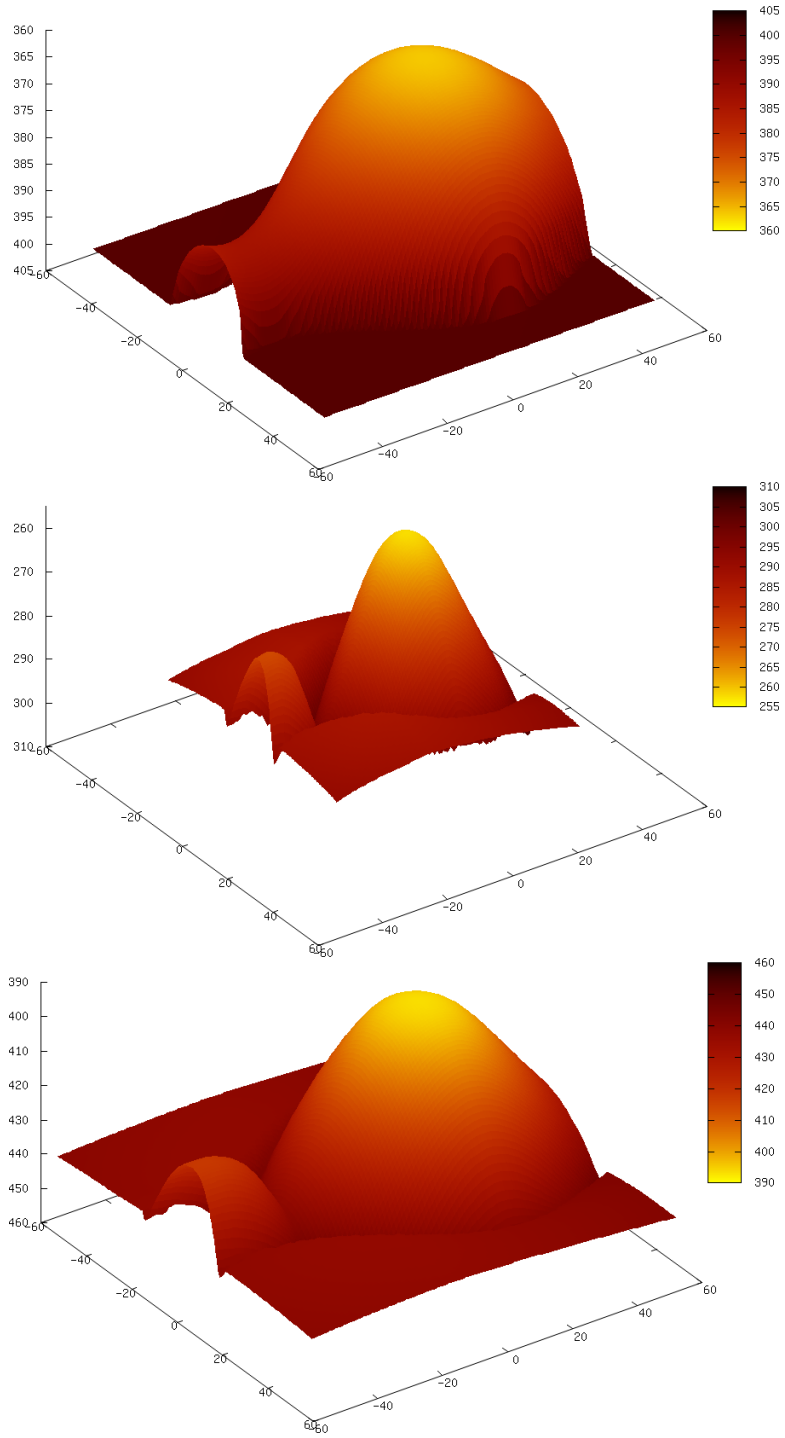

Figure 2.25: Reconstruction of the vase for $k_d = 0.5, k_s = 0.5, \alpha = 5$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

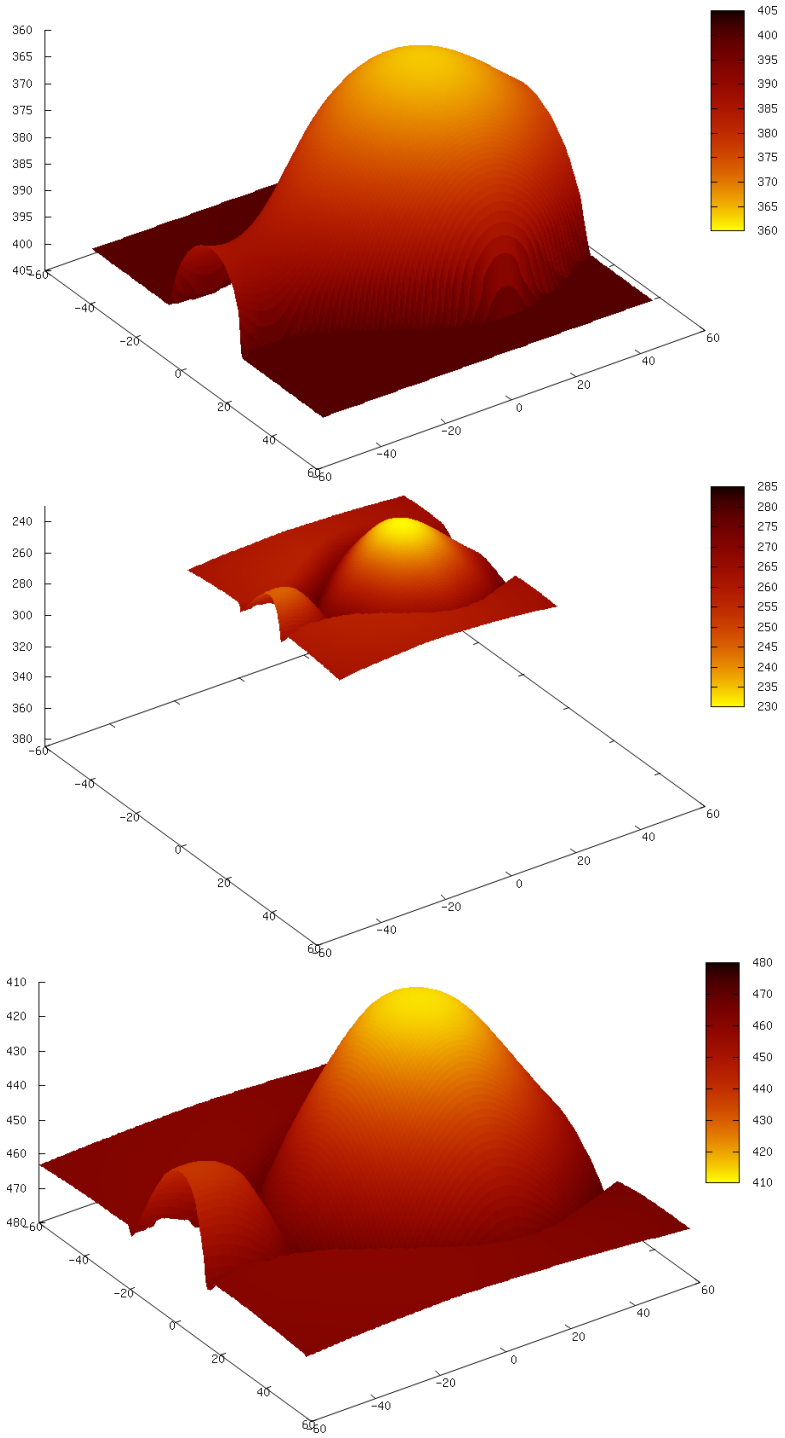

Figure 2.26: Reconstruction of the vase for $k_d = 0.4, k_s = 0.6, \alpha = 5$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

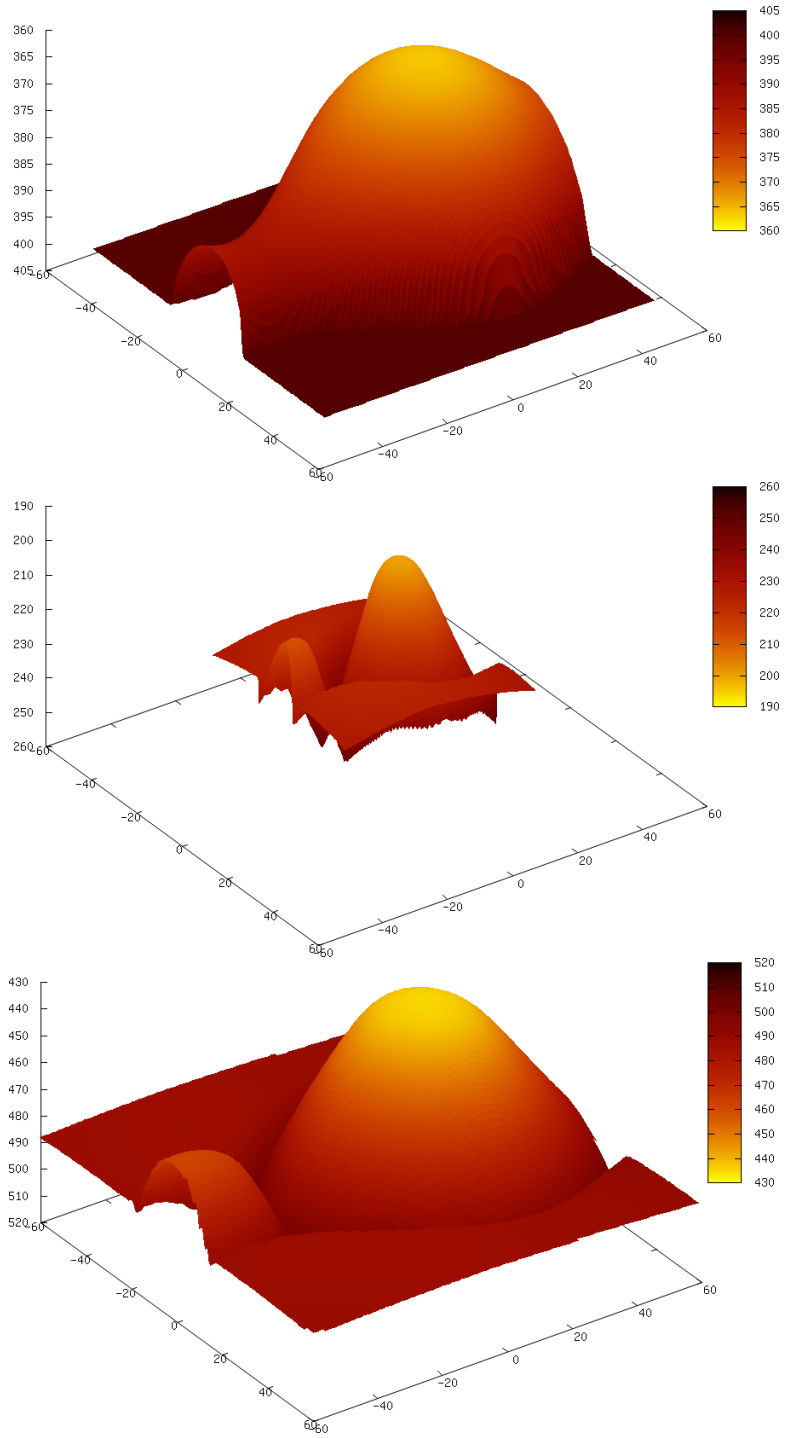

Figure 2.27: Reconstruction of the vase for $k_d = 0.3, k_s = 0.7, \alpha = 5$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

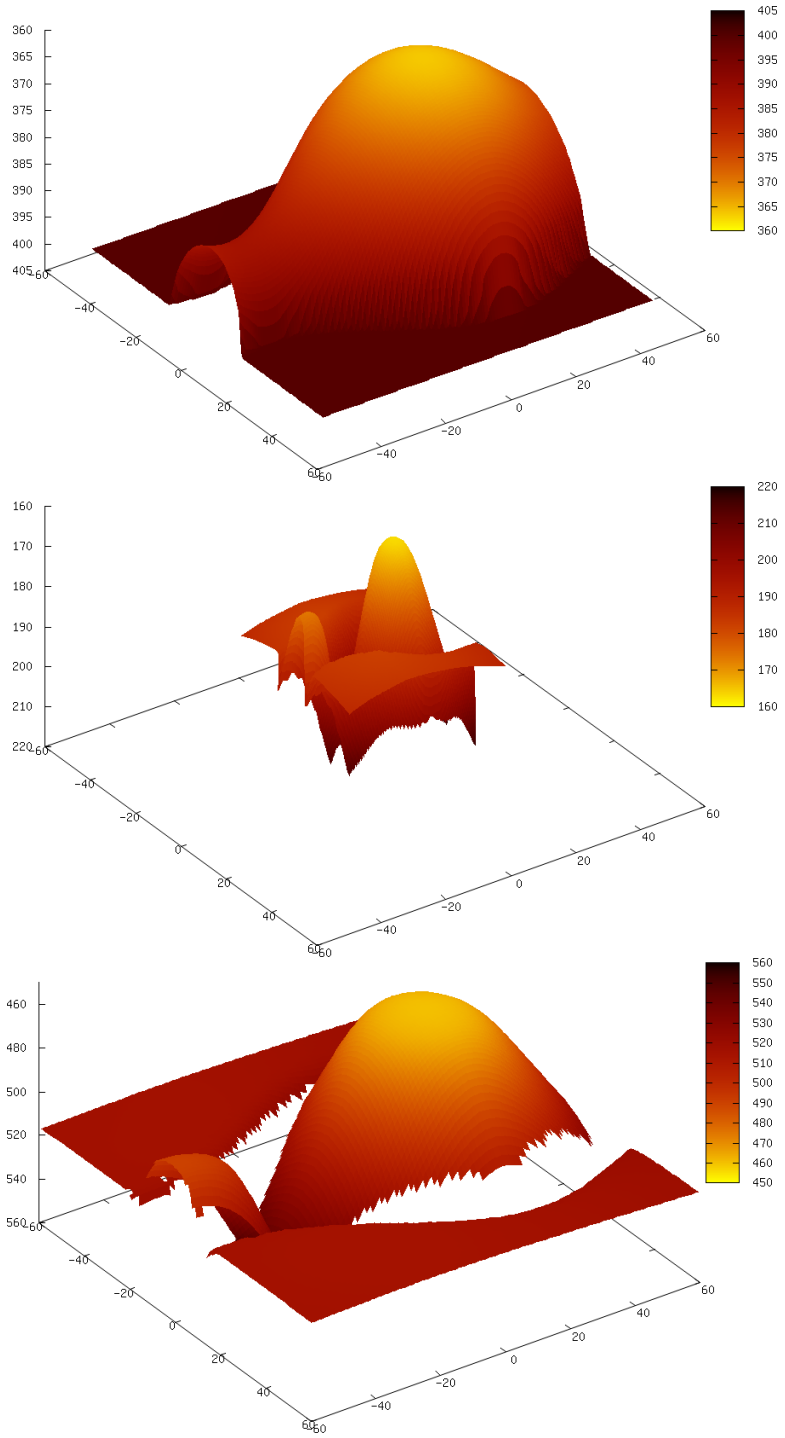

Figure 2.28: Reconstruction of the vase for $k_d = 0.2, k_s = 0.8, \alpha = 5$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

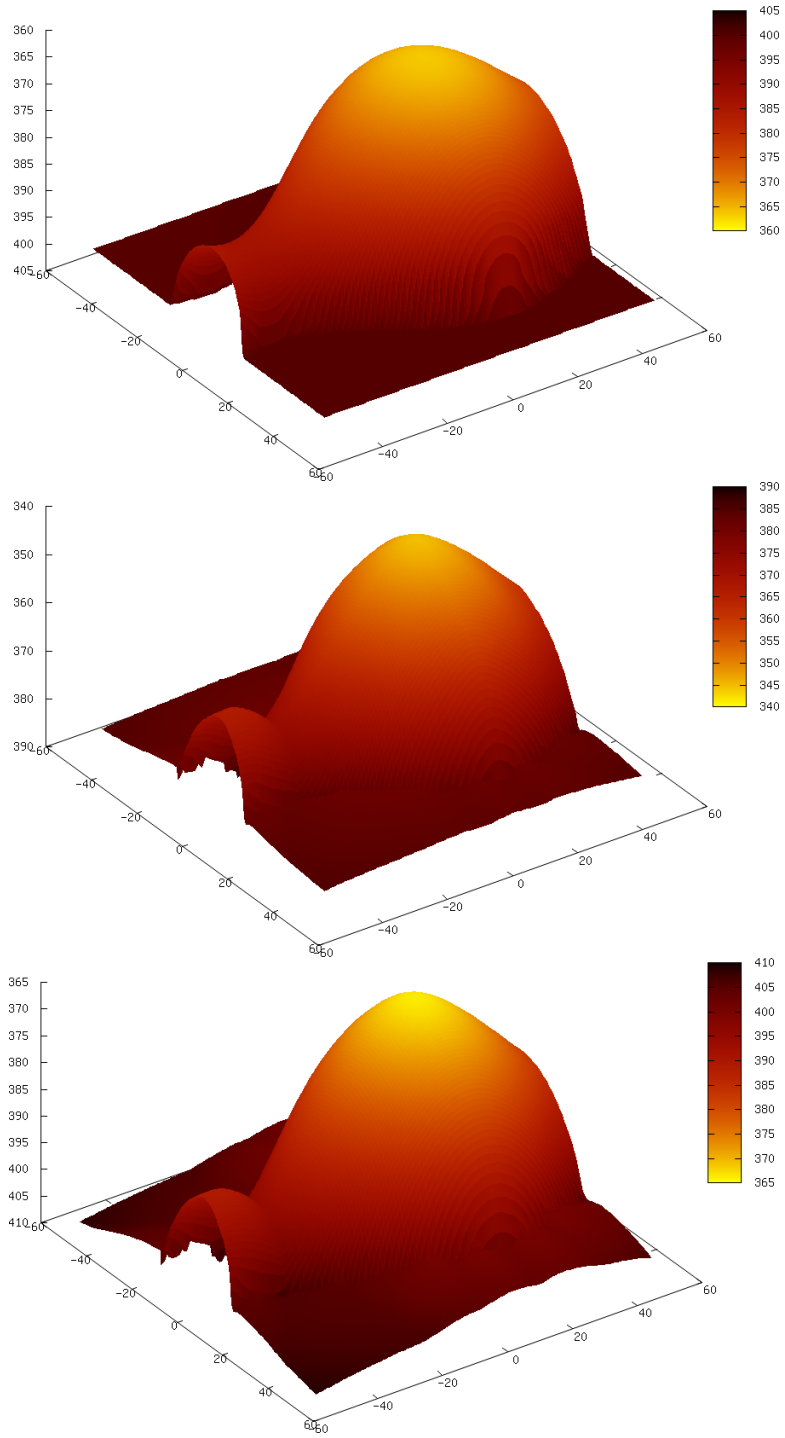

Figure 2.29: Reconstruction of the vase for $k_d = 0.9, k_s = 0.1, \alpha = 10$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

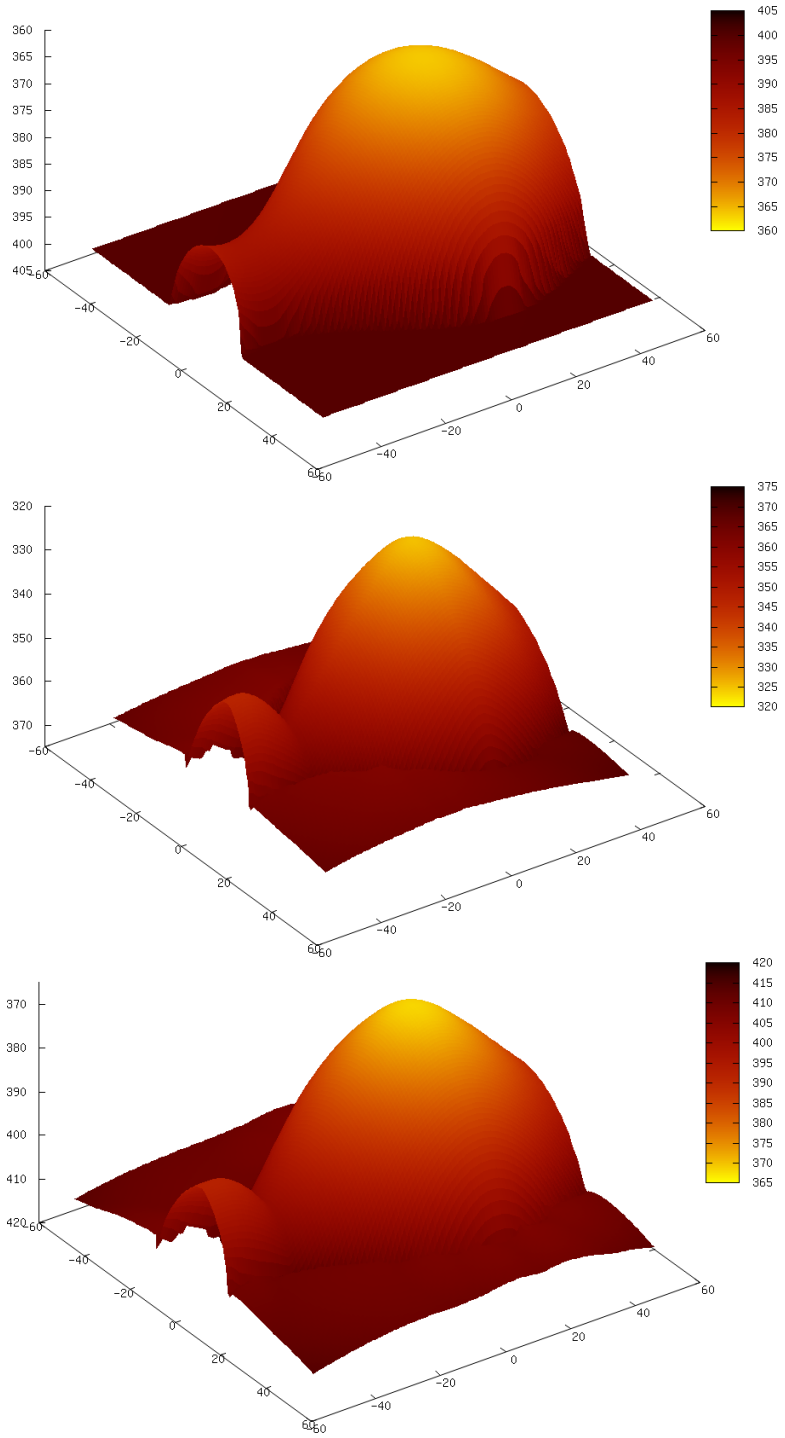

Figure 2.30: Reconstruction of the vase for $k_d = 0.8, k_s = 0.2, \alpha = 10$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

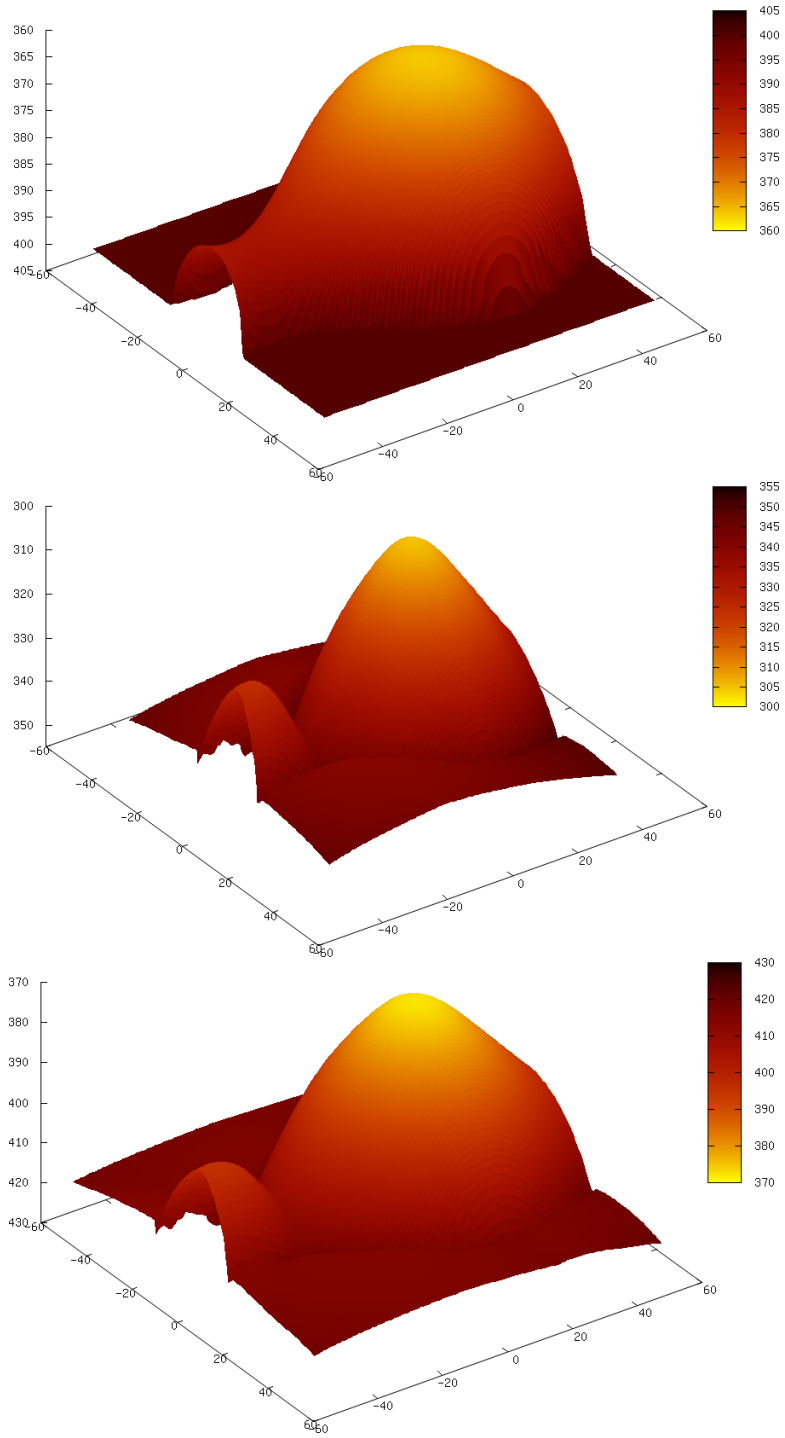

Figure 2.31: Reconstruction of the vase for $k_d = 0.7, k_s = 0.3, \alpha = 10$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

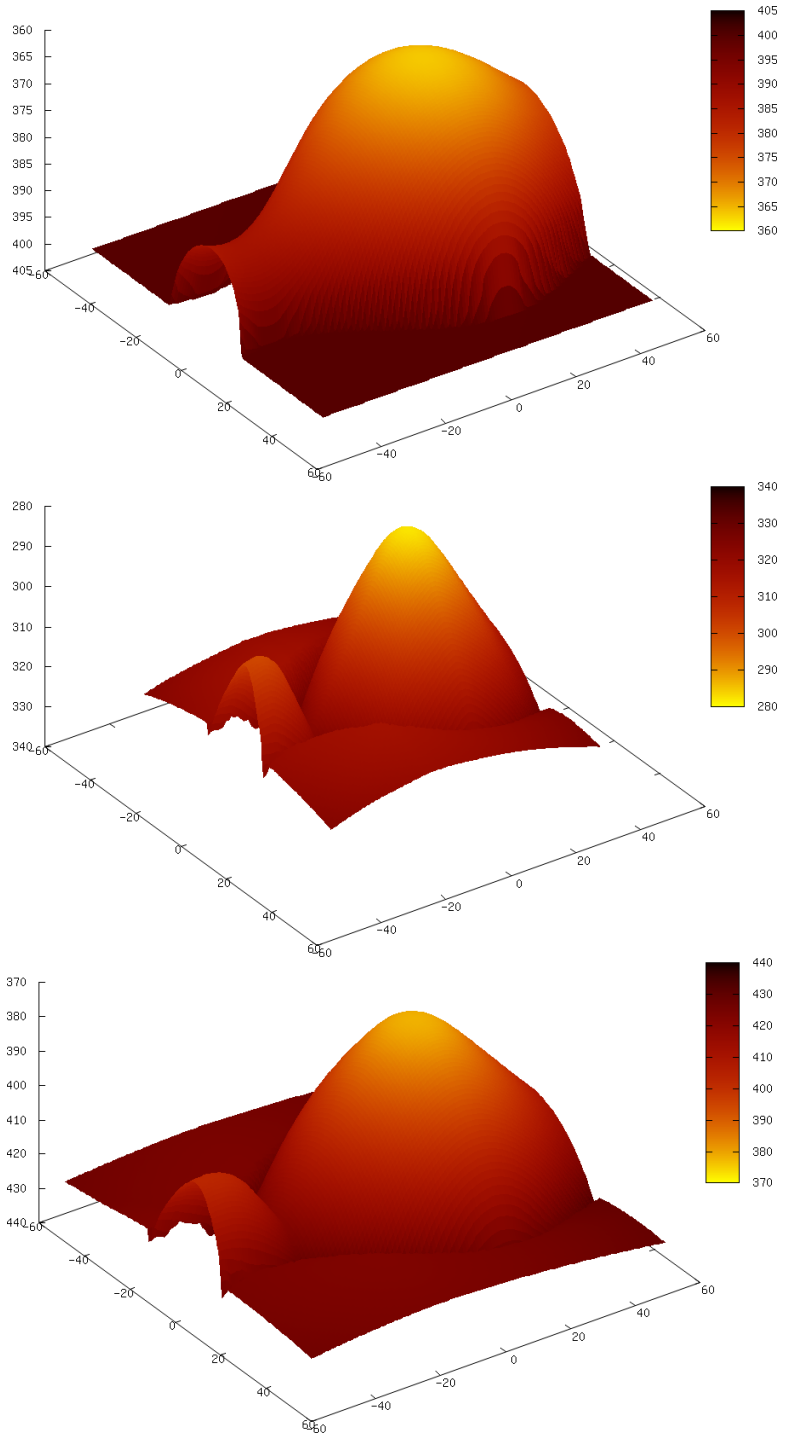

Figure 2.32: Reconstruction of the vase for $k_d = 0.6, k_s = 0.4, \alpha = 10$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

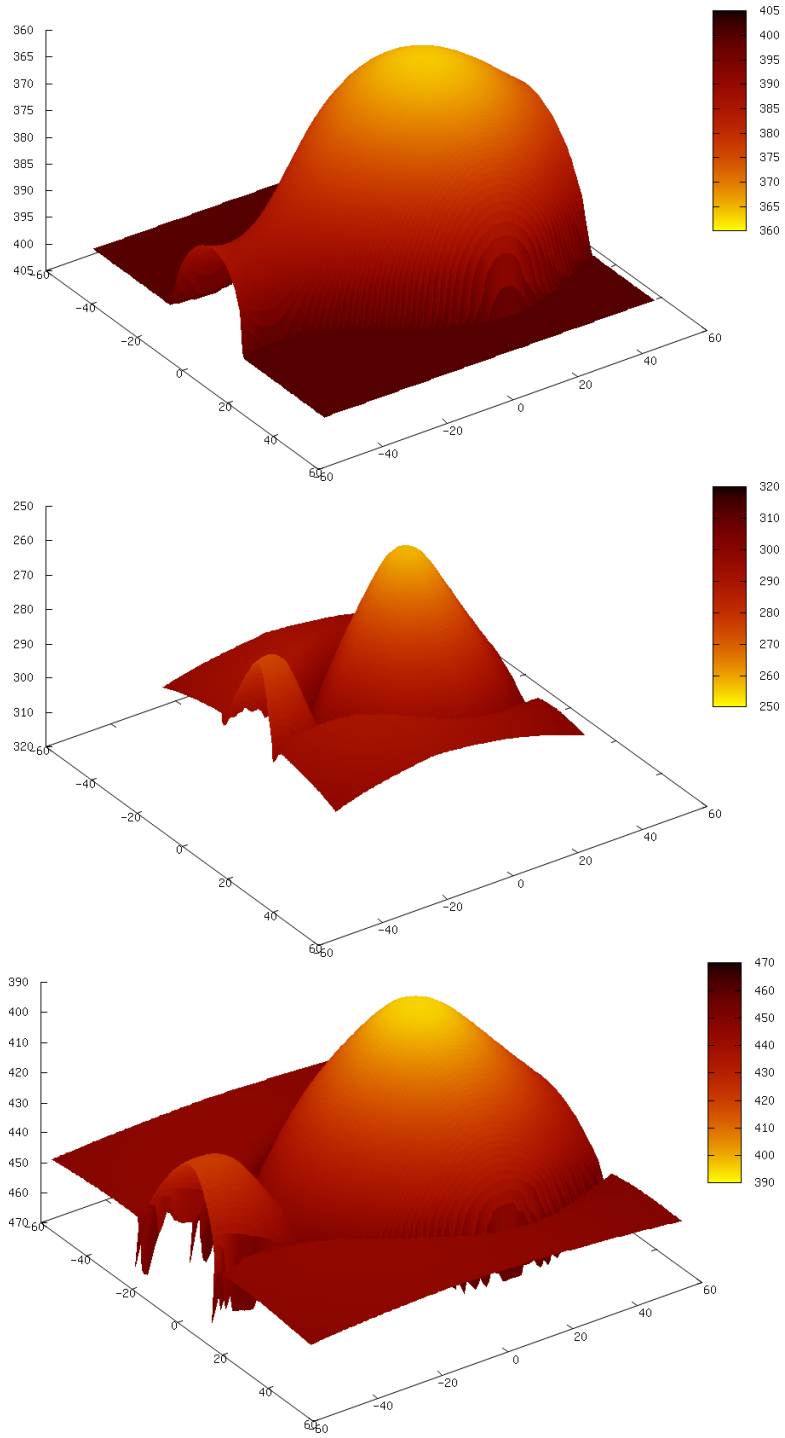

Figure 2.33: Reconstruction of the vase for $k_d = 0.5, k_s = 0.5, \alpha = 10$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

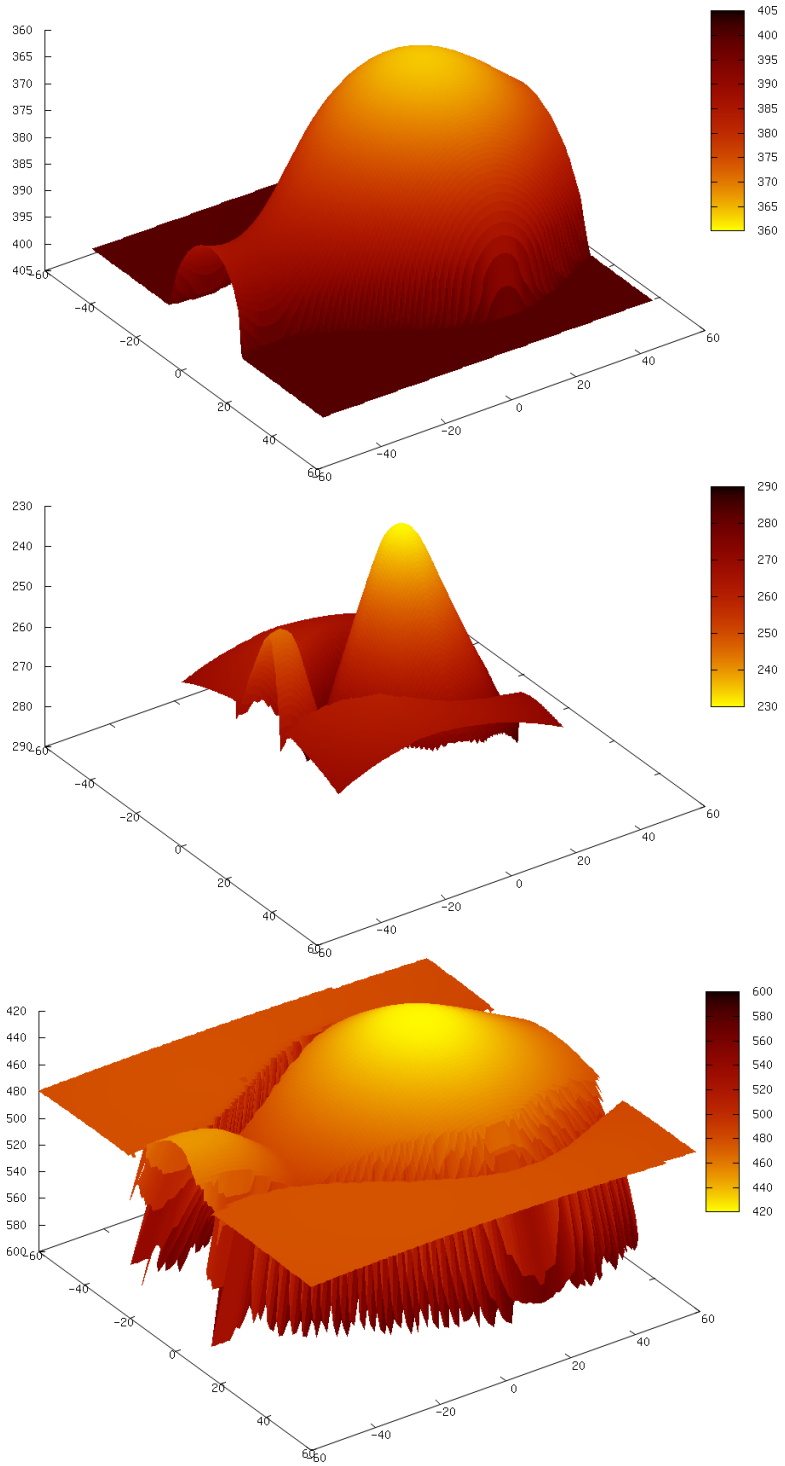

Figure 2.34: Reconstruction of the vase for $k_d = 0.4, k_s = 0.6, \alpha = 10$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

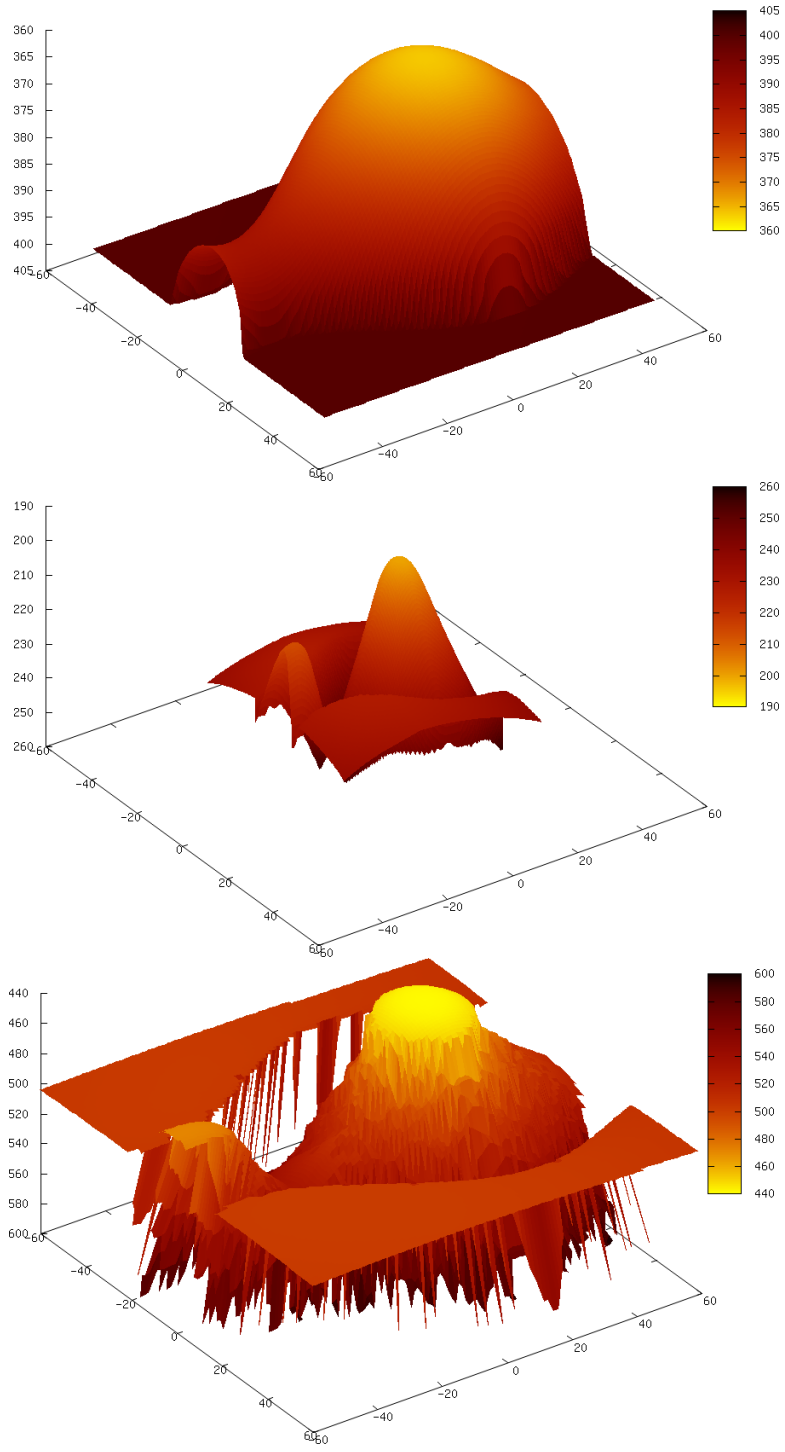

Figure 2.35: Reconstruction of the vase for $k_d = 0.3, k_s = 0.7, \alpha = 10$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

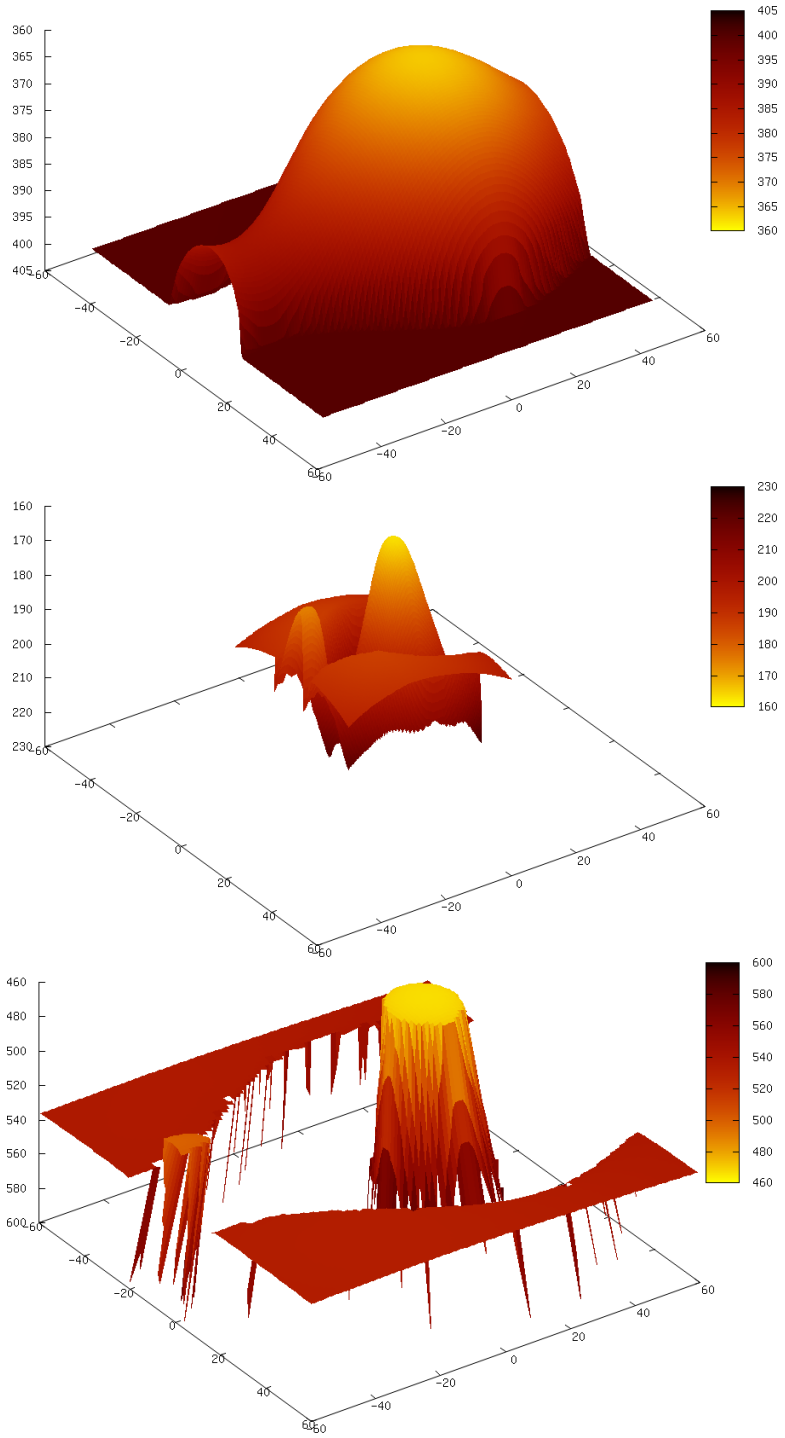

Figure 2.36: Reconstruction of the vase for $k_d = 0.2, k_s = 0.8, \alpha = 10$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

Table 2.10: Depth errors for the vase experiment with varying specularities.

Input Image			Lambertian Model		Phong Model	
k_d	k_s	α	Avg. Error	Max. Error	Avg. Error	Max. Error
0.9	0.1	5	4.18%	5.21%	1.22%	2.78%
0.8	0.2	5	8.91%	10.63%	2.35%	4.88%
0.7	0.3	5	14.08%	16.40%	3.75%	7.24%
0.6	0.4	5	19.80%	22.60%	6.23%	10.72%
0.5	0.5	5	26.16%	29.16%	10.86%	16.46%
0.4	0.6	5	33.38%	36.64%	16.89%	24.00%
0.3	0.7	5	41.72%	45.13%	23.87%	33.56%
0.2	0.8	5	51.72%	55.20%	52.12%	210.43%
0.9	0.1	10	3.94%	5.21%	1.45%	3.00%
0.8	0.2	10	8.41%	10.63%	2.83%	5.32%
0.7	0.3	10	13.31%	16.19%	4.61%	8.00%
0.6	0.4	10	18.82%	22.40%	7.02%	11.41%
0.5	0.5	10	25.06%	29.16%	12.62%	21.47%
0.4	0.6	10	32.14%	36.64%	22.36%	52.89%
0.3	0.7	10	40.31%	45.13%	41.30%	108.68%
0.2	0.8	10	50.18%	55.19%	77.92%	239.11%
0.9	0.1	20	3.64%	5.21%	1.94%	162.62%
0.8	0.2	20	7.75%	10.40%	3.46%	5.78%
0.7	0.3	20	12.37%	16.18%	5.32%	8.35%
0.6	0.4	20	17.47%	22.20%	8.58%	12.54%
0.5	0.5	20	23.45%	28.97%	16.48%	159.41%
0.4	0.6	20	30.28%	36.47%	28.62%	183.04%
0.3	0.7	20	38.22%	44.84%	46.07%	221.88%
0.2	0.8	20	47.96%	54.96%	83.44%	292.25%

Quantitative Evaluation

The second part of the evaluation of this experiment is the discussion of the depth errors. In this analysis, numerical issues will be neglected, since they are the subject of the next chapter. However, this also means some side effects may remain unexplained here. Table 2.10 shows the average and maximal depth errors for the reconstructions of the vase with both models. In addition to the images shown, another set of images with $\alpha = 20$, i.e. a very high exponent, is included in this table.

Looking at the maximal errors, the impression on outliers from the visual investigation is supported. Except for $\alpha = 20$, the outliers remain below

20% of the depth until $k_s = 0.5$, only for even more extreme images, extreme outliers occur again. For very extreme images, in particular for high values of α, these outliers become so dominant that the average error becomes very large as well and for the most extreme images at $\alpha \geq 20$ they even exceed the error of the Lambertian reconstruction.

In contrast to that, the Lambertian reconstructions do not exhibit many outliers, the maximal errors get larger, but never very far above the average error. The Lambertian model can cope much better with outliers than the Phong model. However, the average errors increase with increasing importance of specular highlights. For $k_s = 0.4$, the average error of this method reaches around 20% already, while the reconstruction using the Phong model remains at reasonable 7% in this range.

In terms of the average depth error, it is also very clear that for images containing specular highlights, the Phong model beats the Lambertian model at the reconstruction. However, the presence of a large amount of outliers for extreme images makes the overall performance of the Phong model a bit disappointing so far. This, however, is partially caused by the numerics, which will be introduced only in the next chapter. Later in this thesis, another numerical method will be presented that reduces the outliers drastically and makes it possible to obtain significantly better depth errors as well.

The Mozart Experiment

To investigate the effect of the specular term on a more complex image, the Mozart surface is again the second test case. Since the Mozart surface has a higher resolution, the images are rendered in four times the size, i.e. 256×256. In the vase test case, only images and reconstructions for $\alpha = 5$ and $\alpha = 10$ have been shown. Since this image is larger and the effect is actually visible, results for $\alpha = 20$ will be presented in this chapter as well. Aside from that, exactly the same test parameters will be used for this surface as well. Figures 2.37, 2.38, and 2.39 show the input images for $\alpha = 5$, $\alpha = 10$, and $\alpha = 20$. Table 2.11 shows the respective input parameters.

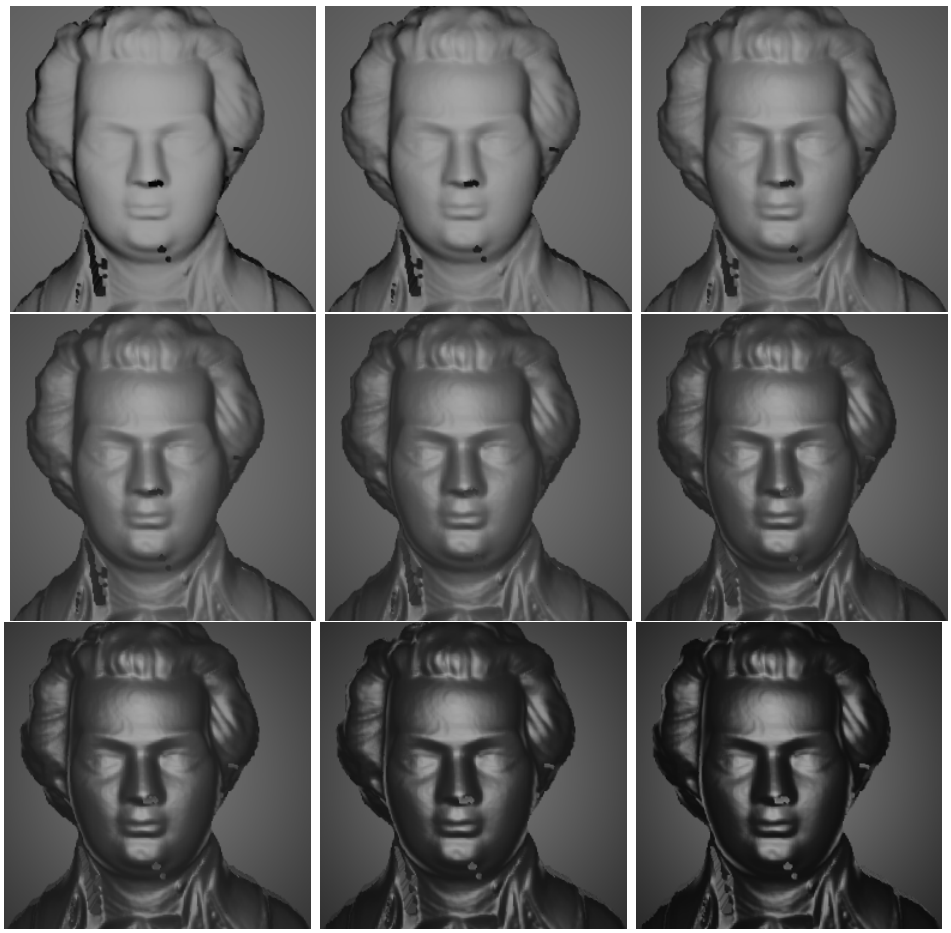

Figure 2.37: Input images for the Mozart face surface with different specular terms. For all images holds $\alpha = 5$ and $k_d = 1 - k_s$. Top left to bottom right: $k_s = 0$, $k_s = 0.1$, $k_s = 0.2$, $k_s = 0.3$, $k_s = 0.4$, $k_s = 0.3$, $k_s = 0.6$, $k_s = 0.7$, $k_s = 0.8$.

Table 2.11: Rendering parameters for the Mozart face experiment with varying specularities.

f	Size	k_a	I_a	k_d	I_d	k_s	I_s	α	h_1	h_2
500	256×256	0	0	1.0	100000	0.0	100000	5	1	1
500	256×256	0	0	0.9	100000	0.1	100000	5	1	1
500	256×256	0	0	0.8	100000	0.2	100000	5	1	1
500	256×256	0	0	0.7	100000	0.3	100000	5	1	1
500	256×256	0	0	0.6	100000	0.4	100000	5	1	1
500	256×256	0	0	0.5	100000	0.5	100000	5	1	1
500	256×256	0	0	0.4	100000	0.6	100000	5	1	1
500	256×256	0	0	0.3	100000	0.7	100000	5	1	1
500	256×256	0	0	0.2	100000	0.8	100000	5	1	1
500	256×256	0	0	1.0	100000	0.0	100000	10	1	1
500	256×256	0	0	0.9	100000	0.1	100000	10	1	1
500	256×256	0	0	0.8	100000	0.2	100000	10	1	1
500	256×256	0	0	0.7	100000	0.3	100000	10	1	1
500	256×256	0	0	0.6	100000	0.4	100000	10	1	1
500	256×256	0	0	0.5	100000	0.5	100000	10	1	1
500	256×256	0	0	0.4	100000	0.6	100000	10	1	1
500	256×256	0	0	0.3	100000	0.7	100000	10	1	1
500	256×256	0	0	0.2	100000	0.8	100000	10	1	1
500	256×256	0	0	1.0	100000	0.0	100000	20	1	1
500	256×256	0	0	0.9	100000	0.1	100000	20	1	1
500	256×256	0	0	0.8	100000	0.2	100000	20	1	1
500	256×256	0	0	0.7	100000	0.3	100000	20	1	1
500	256×256	0	0	0.6	100000	0.4	100000	20	1	1
500	256×256	0	0	0.5	100000	0.5	100000	20	1	1
500	256×256	0	0	0.4	100000	0.6	100000	20	1	1
500	256×256	0	0	0.3	100000	0.7	100000	20	1	1
500	256×256	0	0	0.2	100000	0.8	100000	20	1	1

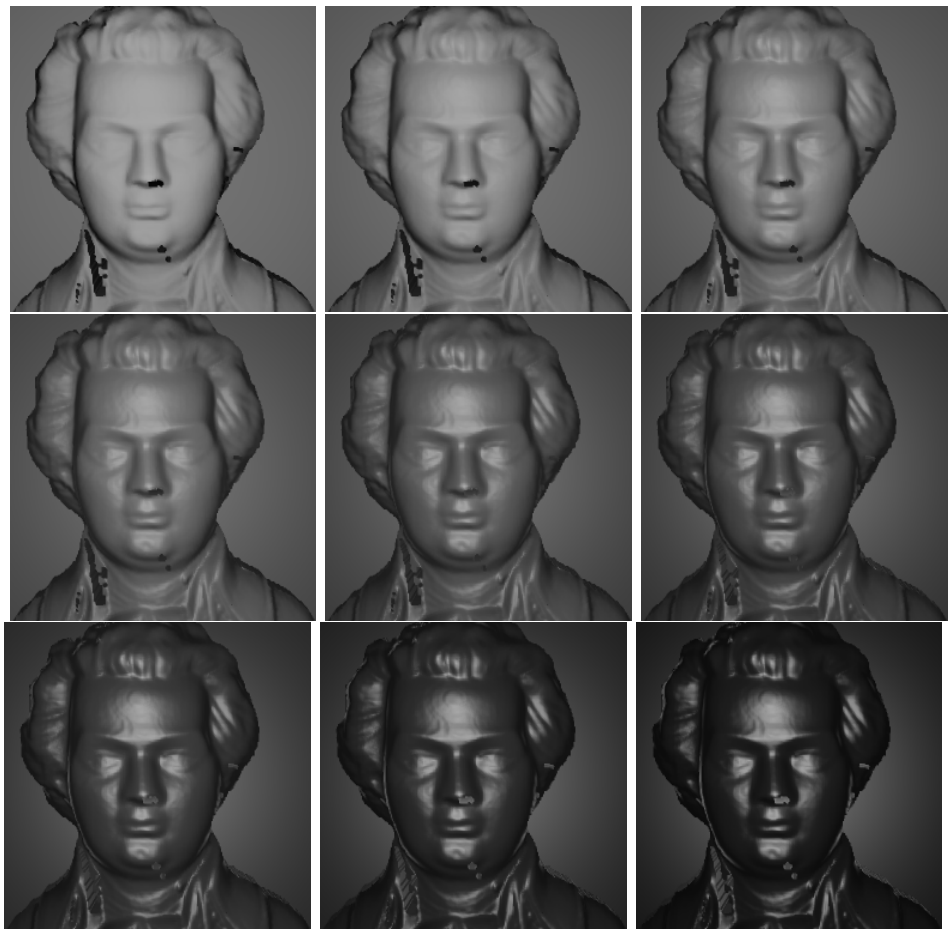

Figure 2.38: Input images for the Mozart face surface with different specular terms. For all images holds $\alpha = 10$ and $k_d = 1 - k_s$. Top left to bottom right: $k_s = 0$, $k_s = 0.1$, $k_s = 0.2$, $k_s = 0.3$, $k_s = 0.4$, $k_s = 0.3$, $k_s = 0.6$, $k_s = 0.7$, $k_s = 0.8$.

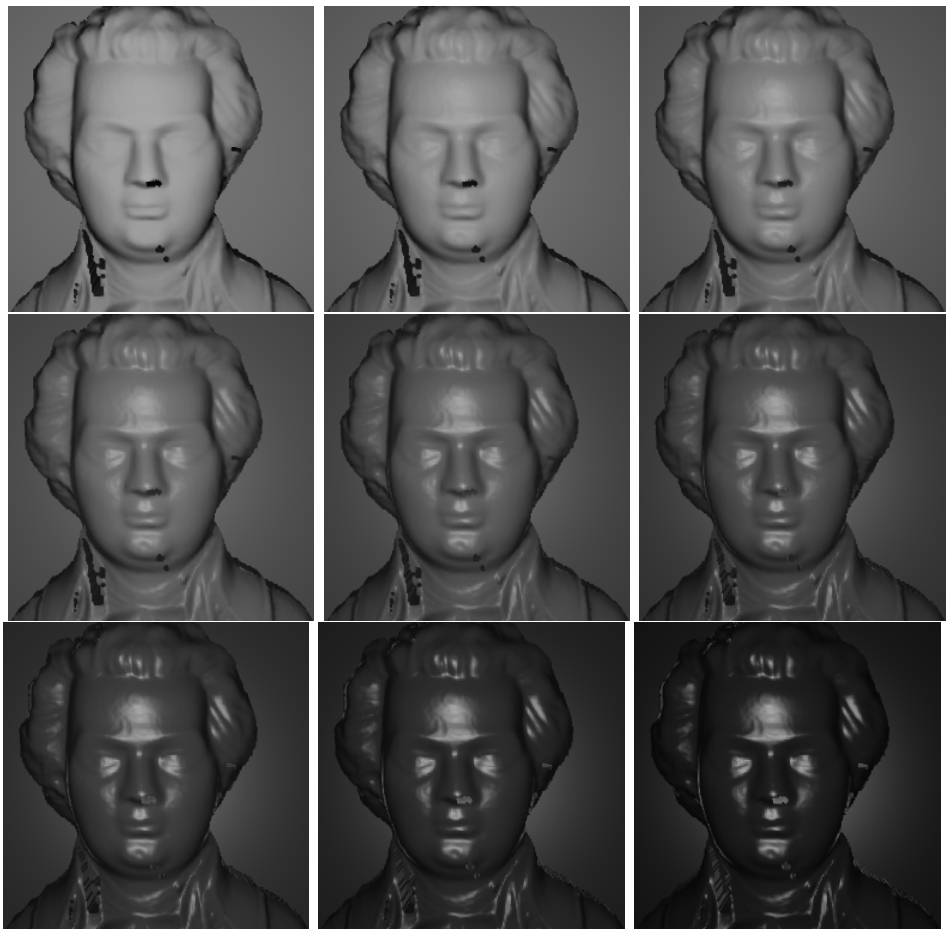

Figure 2.39: Input images for the Mozart face surface with different specular terms. For all images holds $\alpha = 20$ and $k_d = 1 - k_s$. Top left to bottom right: $k_s = 0$, $k_s = 0.1$, $k_s = 0.2$, $k_s = 0.3$, $k_s = 0.4$, $k_s = 0.3$, $k_s = 0.6$, $k_s = 0.7$, $k_s = 0.8$.

Visual Evaluation

Again, the first and possibly most important evaluation of this experiment is the visual assessment of the reconstruction quality. The purely Lambertian experiment on the Mozart face for $f = 500$ contained a significant amount of outliers, cf. Figure 2.16. The depth error, however, was very acceptable, cf. Tables 2.7 and 2.8, as was the visual quality of the result when omitting the outliers.

For this experiment, in addition to the results for $\alpha = 5$ in Figures 2.40-2.47 and the results for $\alpha = 10$ in Figures 2.48-2.55, also the results for $\alpha = 20$ are shown in Figures 2.56-2.63.

In principle, the effects observed are very similar to the vase experiment. For large amounts of specular highlights, the Lambertian reconstructions are a lot too close, i.e., the depth values are much too small. In addition, the face gets distorted for these reconstructions very early. For high amounts of specular highlights, it is impossible to see a face in the reconstructions.

For the Phong reconstructions, the outliers starting at around $k_s = 0.5$ become much more prominent than for the vase experiment. In the plots of the surfaces, this becomes clear when holes in the reconstruction arise. These holes are strong outliers which are omitted from the plot. Starting at $k_s = 0.6$, large parts of the reconstruction are already dominated by outliers. For the most extreme images, the holes get larger and larger, and less of a face remains. However, except for the reconstruction with $k_s = 0.2$ at large values of α, it is still possible to recognise a face. Also, the reconstructions remain at an approximately correct depth scale, at least unless the outliers become too dominant.

Again, the Phong reconstructions are much better than the Lambertian ones. However, outliers are again a problem, even more for this larger image with more edges. The more robust numerical method in Chapter 4 will also lead to much more convincing results for this experiment.

Quantitative Evaluation

The depth errors for all the experiments can be found in Table 2.12. The results are in accordance with the vase experiment and the visual evaluation. While the average depth errors are constantly increasing for the Lambertian reconstruction with increasing k_s, they remain much more stable for the Phong reconstruction, resulting in reasonable errors up to $k_s = 0.4$. However, for larger k_s, outliers become dominant, resulting in very large maximal depth errors. For the most extreme images, outliers become so dominant that the average depth error becomes as bad as for the Lamber-

tian reconstruction.

Essentially, the conclusion of this experiment is the same as the conclusion of the vase experiment: The Phong method is better for the reconstruction of images with specular highlights, however, for very extreme images, outliers become an issue, at least for the numerical method used here.

Table 2.12: Depth errors for the Mozart face experiment with varying specularities.

Input Image			Lambertian Model		Phong Model	
k_d	k_s	α	Avg. Error	Max. Error	Avg. Error	Max. Error
0.9	0.1	5	6.24%	179.03%	3.43%	197.07%
0.8	0.2	5	10.42%	25.37%	3.14%	19.70%
0.7	0.3	5	15.23%	29.88%	3.65%	40.17%
0.6	0.4	5	20.54%	34.99%	4.82%	43.49%
0.5	0.5	5	26.54%	40.56%	8.53%	49.35%
0.4	0.6	5	33.38%	46.75%	15.01%	118.82%
0.3	0.7	5	41.32%	53.80%	27.80%	150.06%
0.2	0.8	5	51.07%	62.23%	64.76%	222.05%
0.9	0.1	10	6.01%	179.03%	3.84%	197.07%
0.8	0.2	10	9.75%	25.09%	3.14%	114.02%
0.7	0.3	10	14.23%	29.78%	4.00%	41.27%
0.6	0.4	10	19.16%	34.85%	6.01%	44.65%
0.5	0.5	10	24.81%	40.23%	11.64%	126.99%
0.4	0.6	10	31.30%	46.41%	25.04%	149.22%
0.3	0.7	10	38.88%	53.47%	50.45%	182.65%
0.2	0.8	10	48.44%	61.94%	88.73%	271.87%
0.9	0.1	20	6.24%	179.03%	4.71%	197.07%
0.8	0.2	20	9.12%	25.03%	4.19%	132.07%
0.7	0.3	20	13.23%	29.50%	5.39%	134.98%
0.6	0.4	20	17.79%	34.55%	8.61%	145.20%
0.5	0.5	20	23.02%	39.93%	18.74%	161.68%
0.4	0.6	20	29.12%	46.11%	31.22%	186.09%
0.3	0.7	20	36.24%	53.01%	55.24%	226.60%
0.2	0.8	20	45.34%	61.38%	108.23%	292.57%

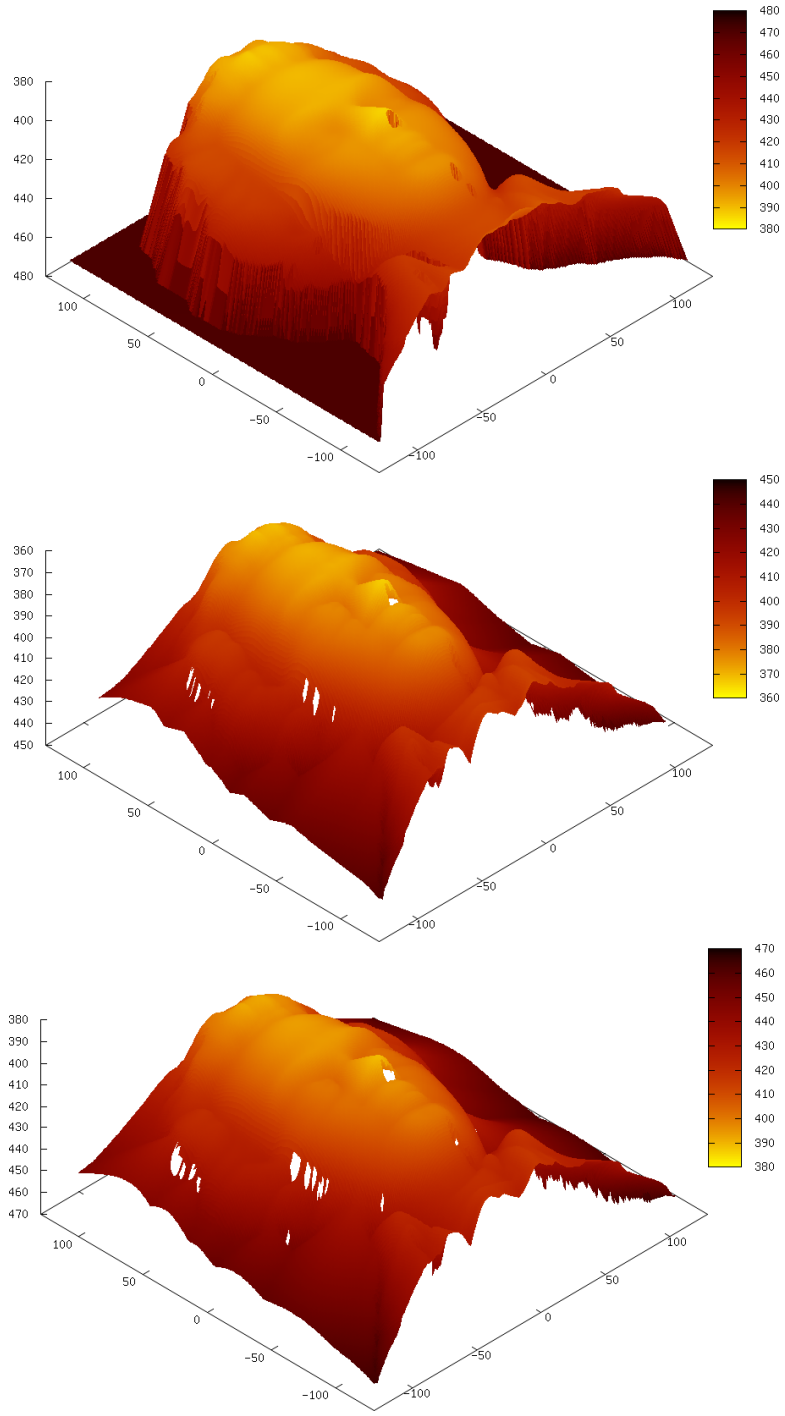

Figure 2.40: Reconstruction of the Mozart face for $k_d = 0.9, k_s = 0.1, \alpha = 5$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

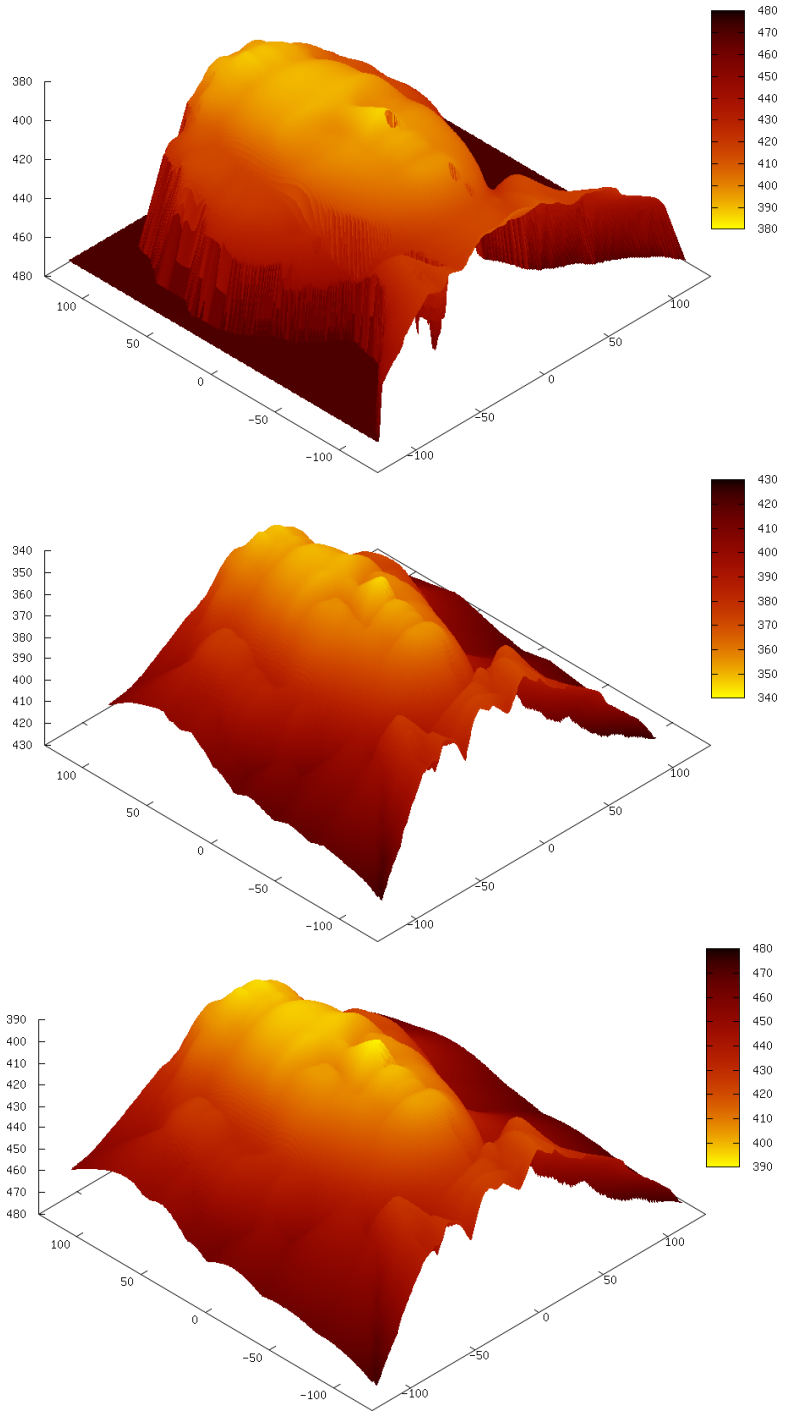

Figure 2.41: Reconstruction of the Mozart face for $k_d = 0.8, k_s = 0.2, \alpha = 5$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

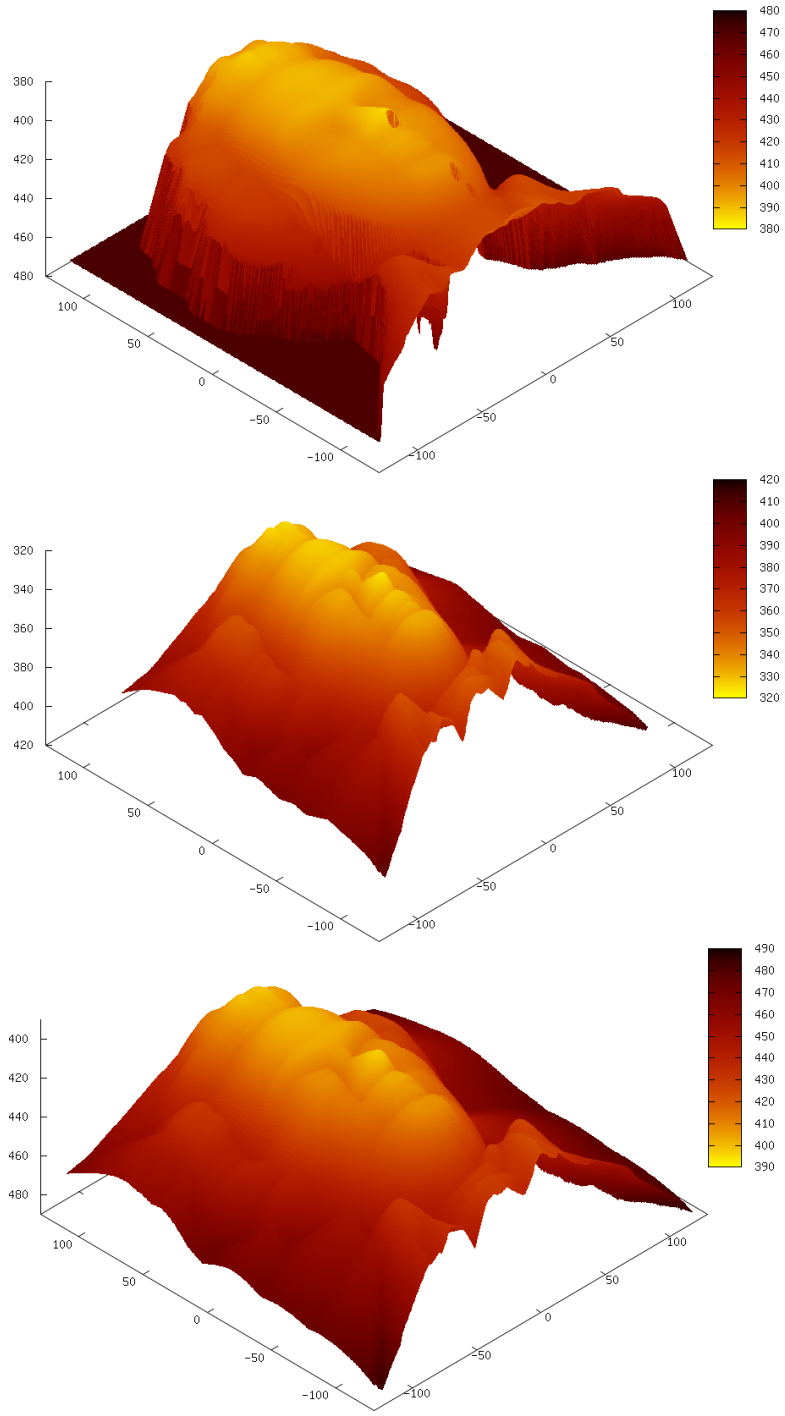

Figure 2.42: Reconstruction of the Mozart face for $k_d = 0.7, k_s = 0.3, \alpha = 5$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

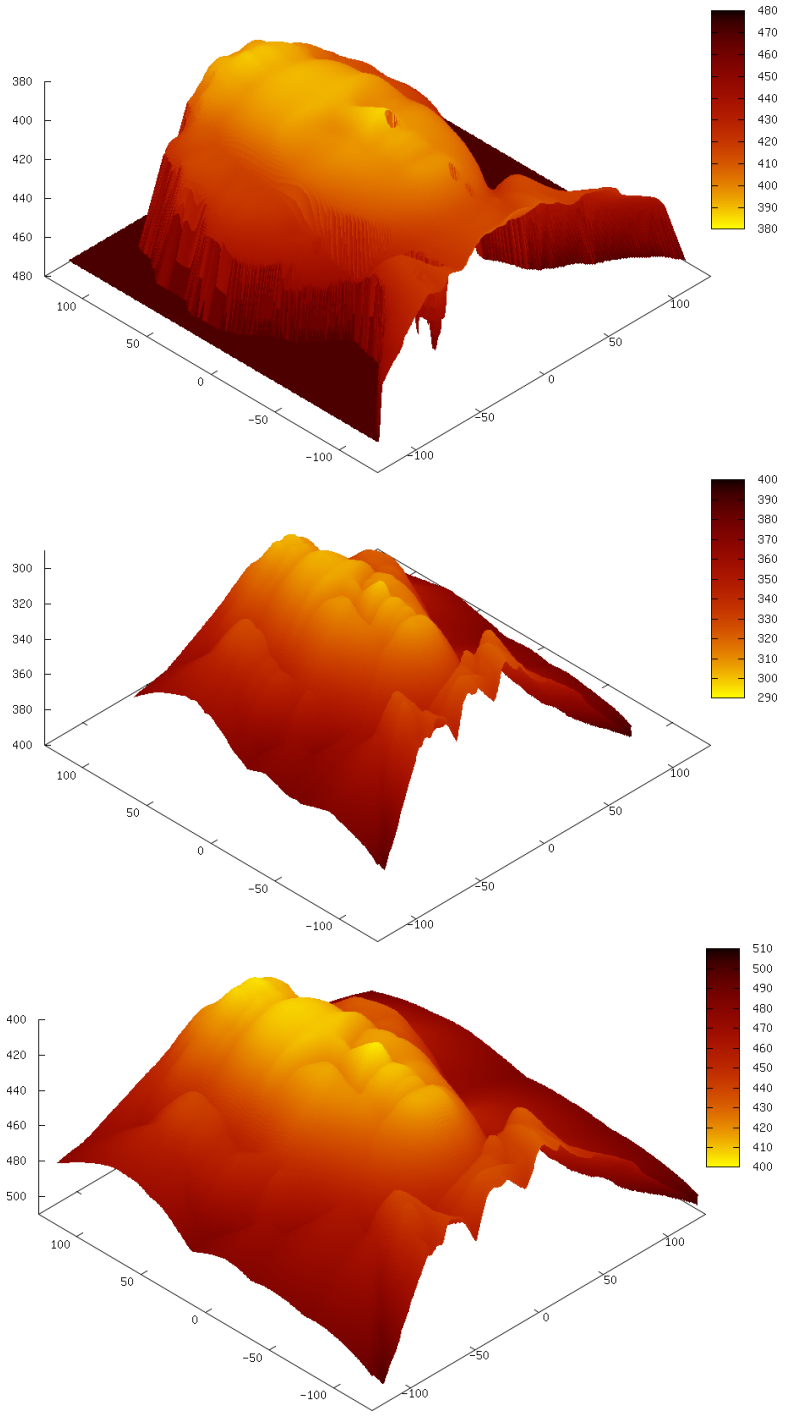

Figure 2.43: Reconstruction of the Mozart face for $k_d = 0.6, k_s = 0.4, \alpha = 5$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

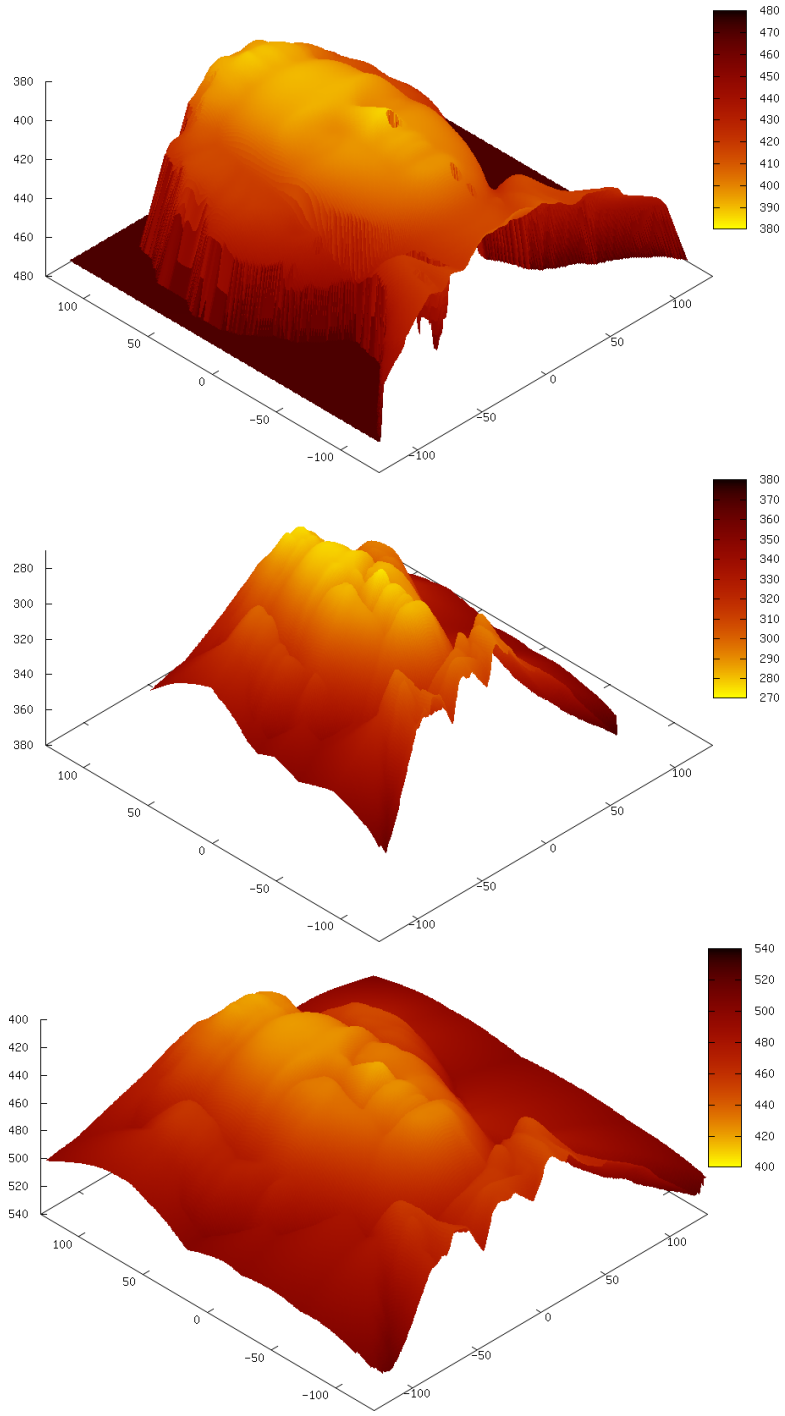

Figure 2.44: Reconstruction of the Mozart face for $k_d = 0.5, k_s = 0.5, \alpha = 5$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

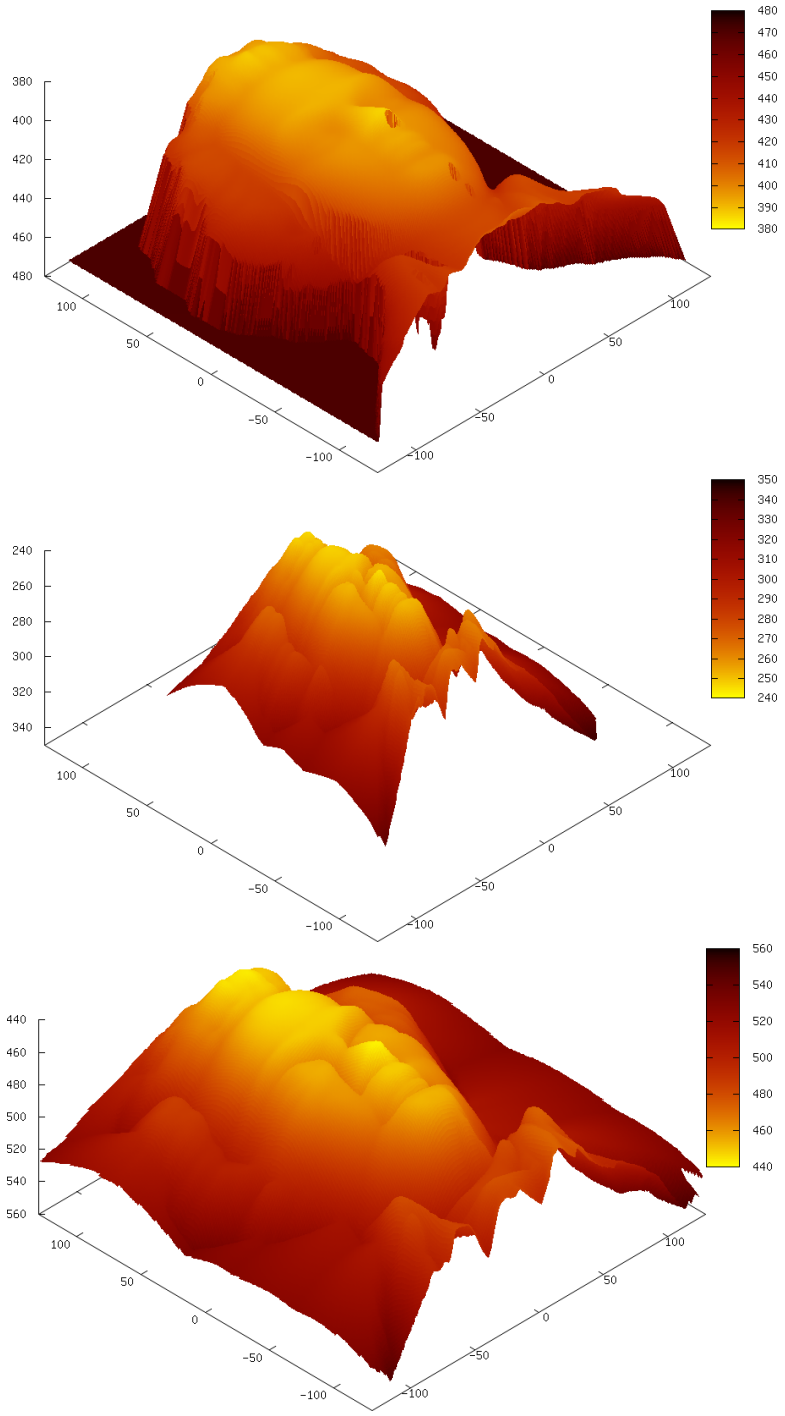

Figure 2.45: Reconstruction of the Mozart face for $k_d = 0.4, k_s = 0.6, \alpha = 5$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

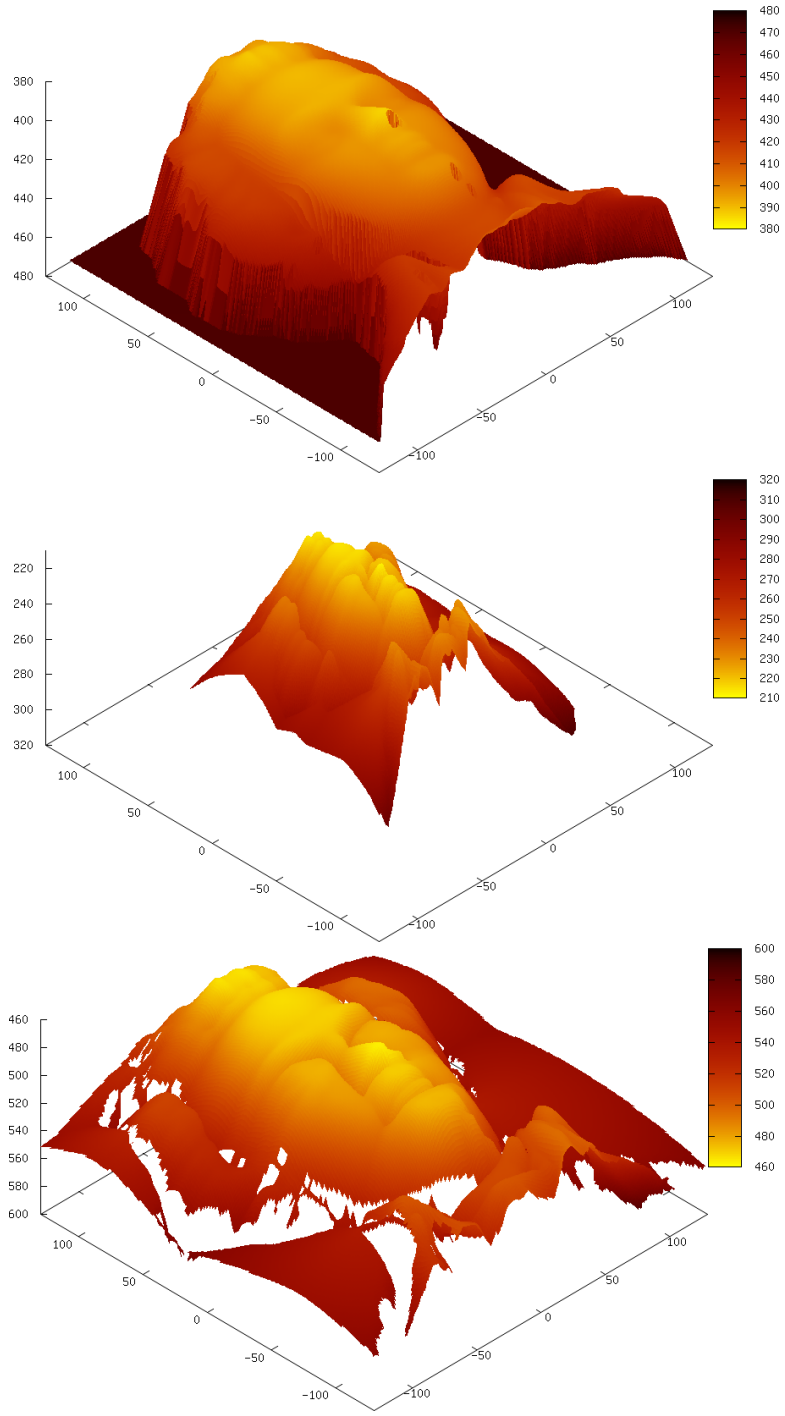

Figure 2.46: Reconstruction of the Mozart face for $k_d = 0.3, k_s = 0.7, \alpha = 5$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

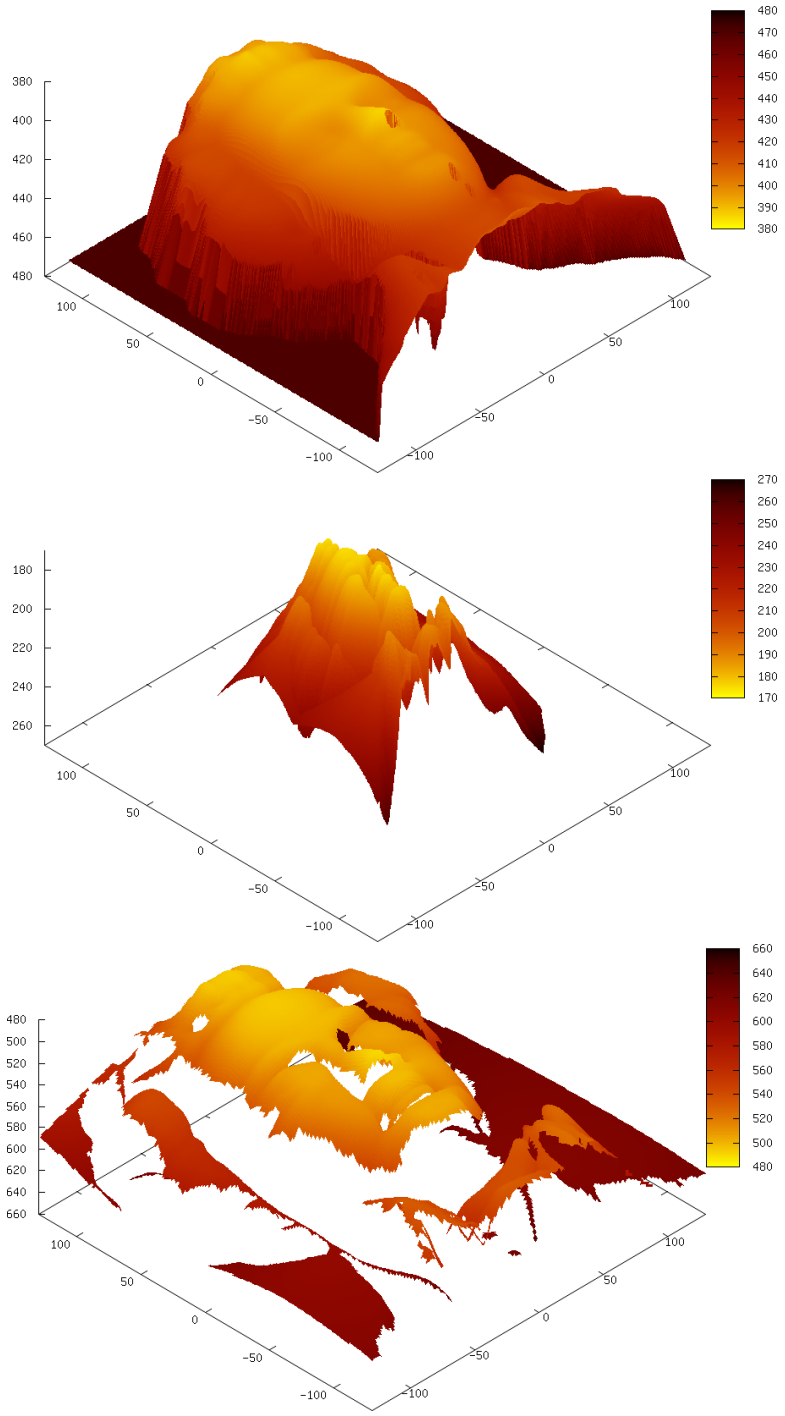

Figure 2.47: Reconstruction of the Mozart face for $k_d = 0.2, k_s = 0.8, \alpha = 5$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

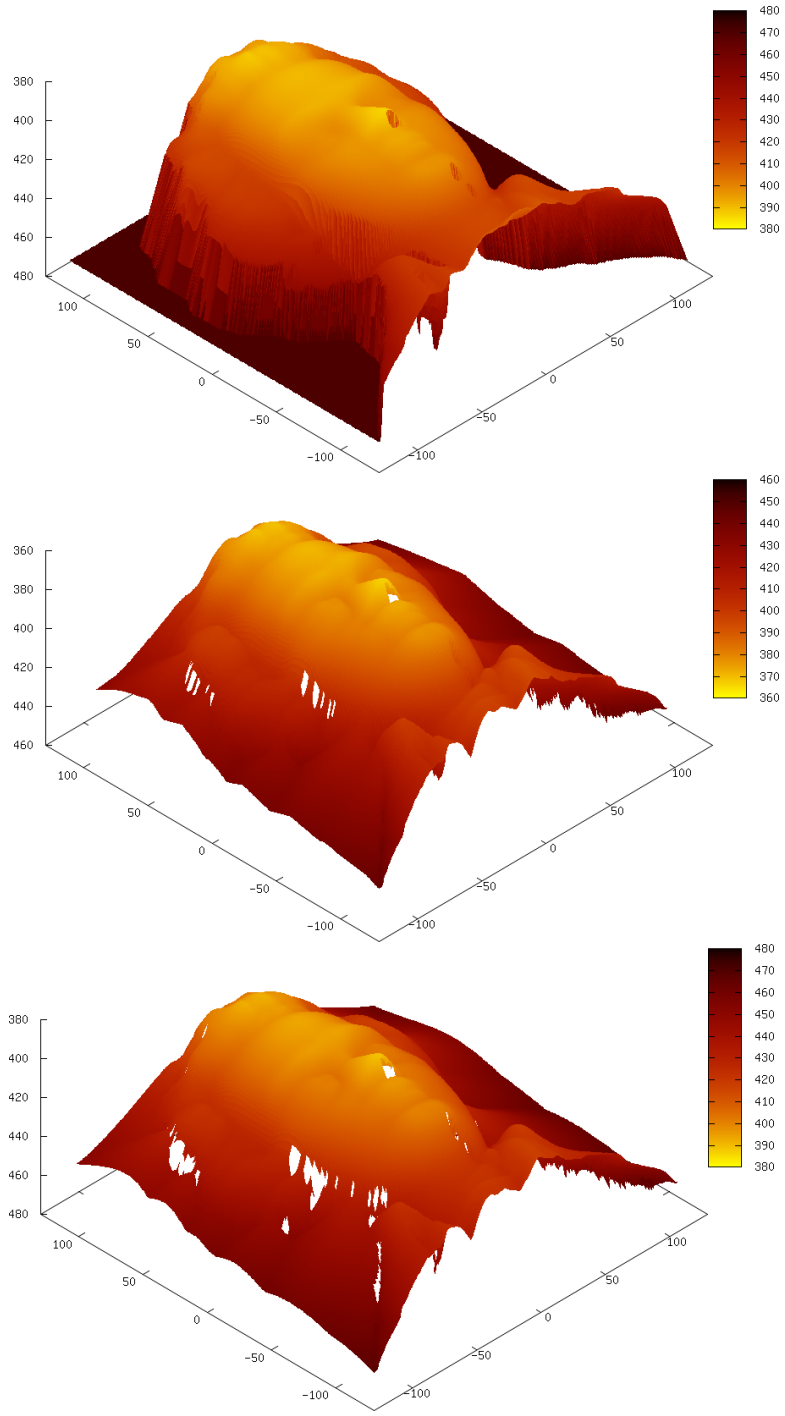

Figure 2.48: Reconstruction of the Mozart face for $k_d = 0.9, k_s = 0.1, \alpha = 10$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

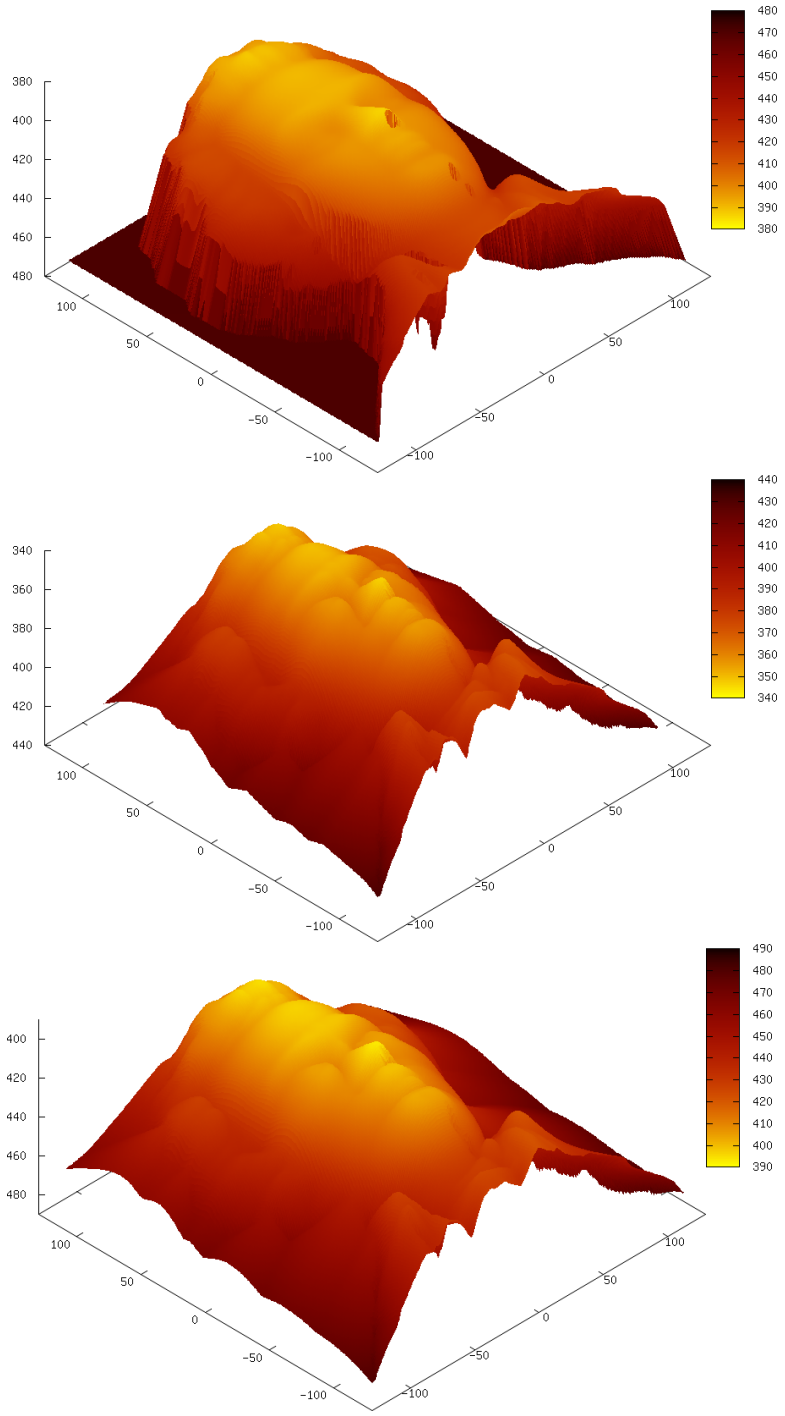

Figure 2.49: Reconstruction of the Mozart face for $k_d = 0.8, k_s = 0.2, \alpha = 10$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

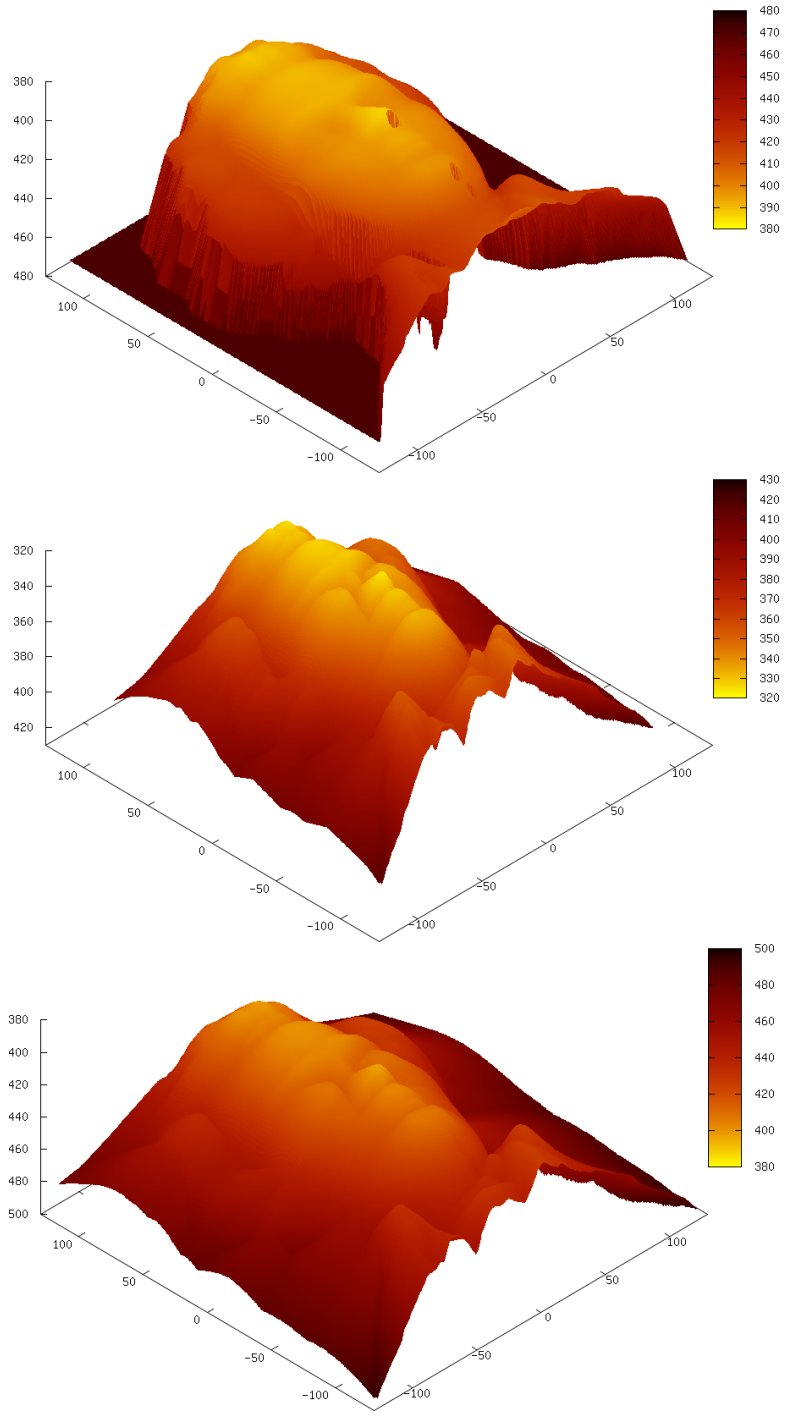

Figure 2.50: Reconstruction of the Mozart face for $k_d = 0.7, k_s = 0.3, \alpha = 10$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

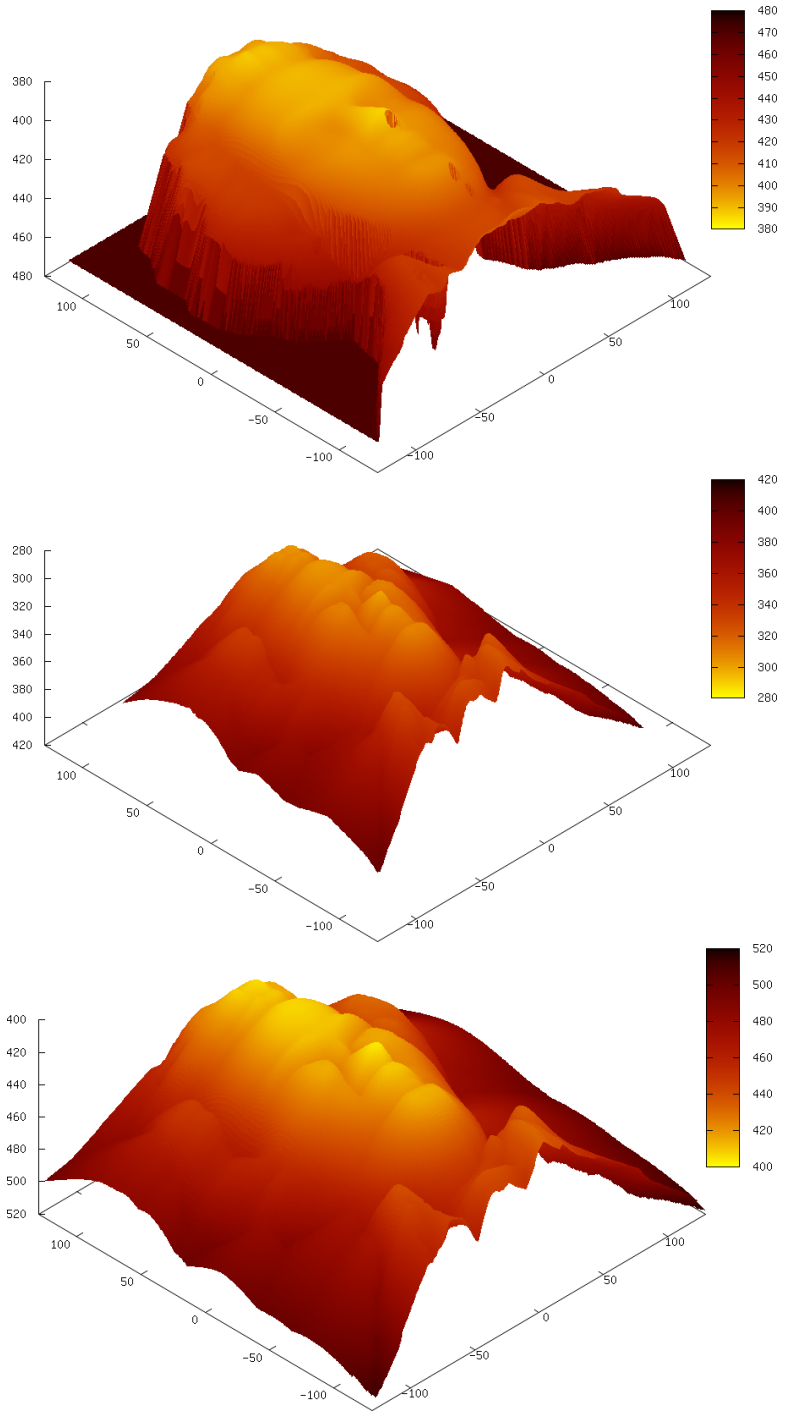

Figure 2.51: Reconstruction of the Mozart face for $k_d = 0.6, k_s = 0.4, \alpha = 10$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

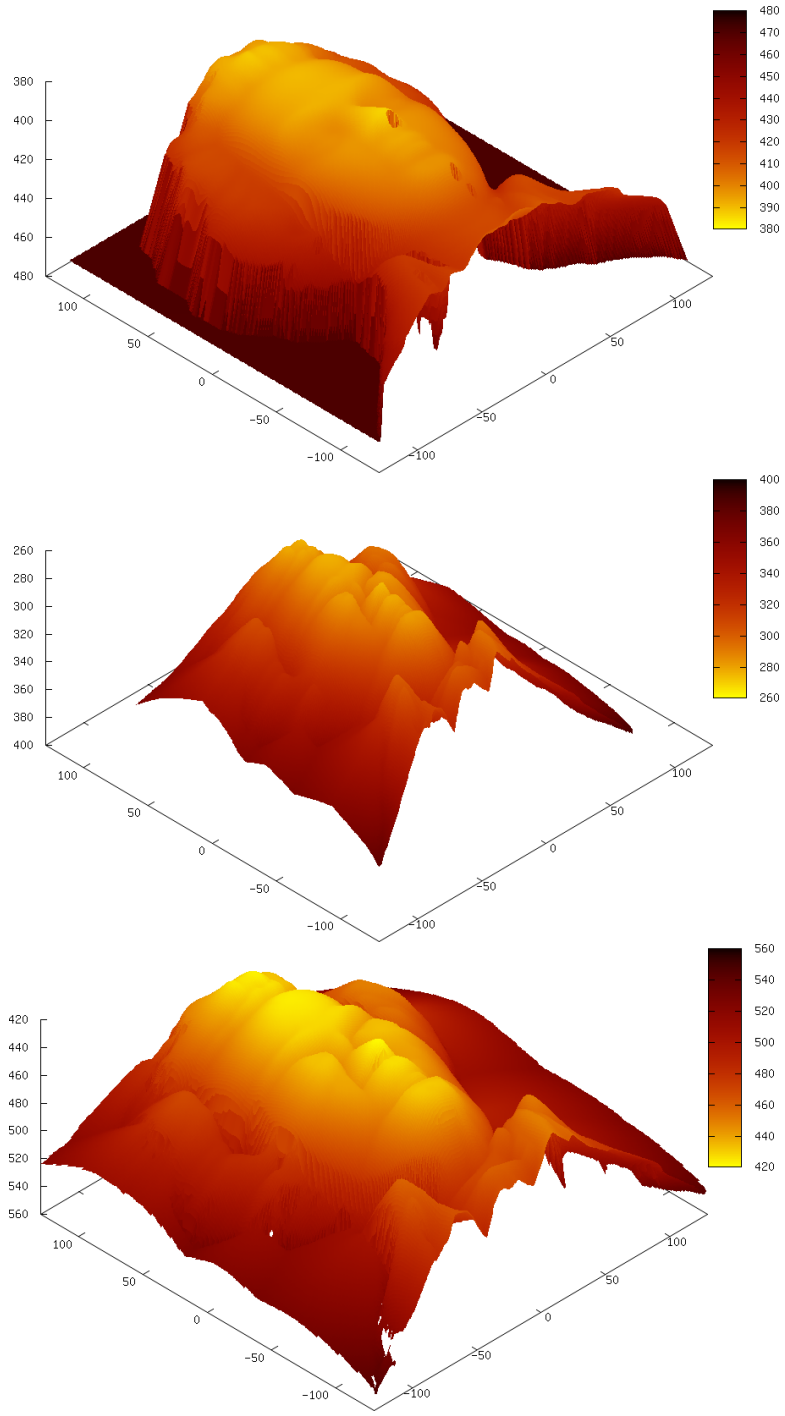

Figure 2.52: Reconstruction of the Mozart face for $k_d = 0.5, k_s = 0.5, \alpha = 10$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

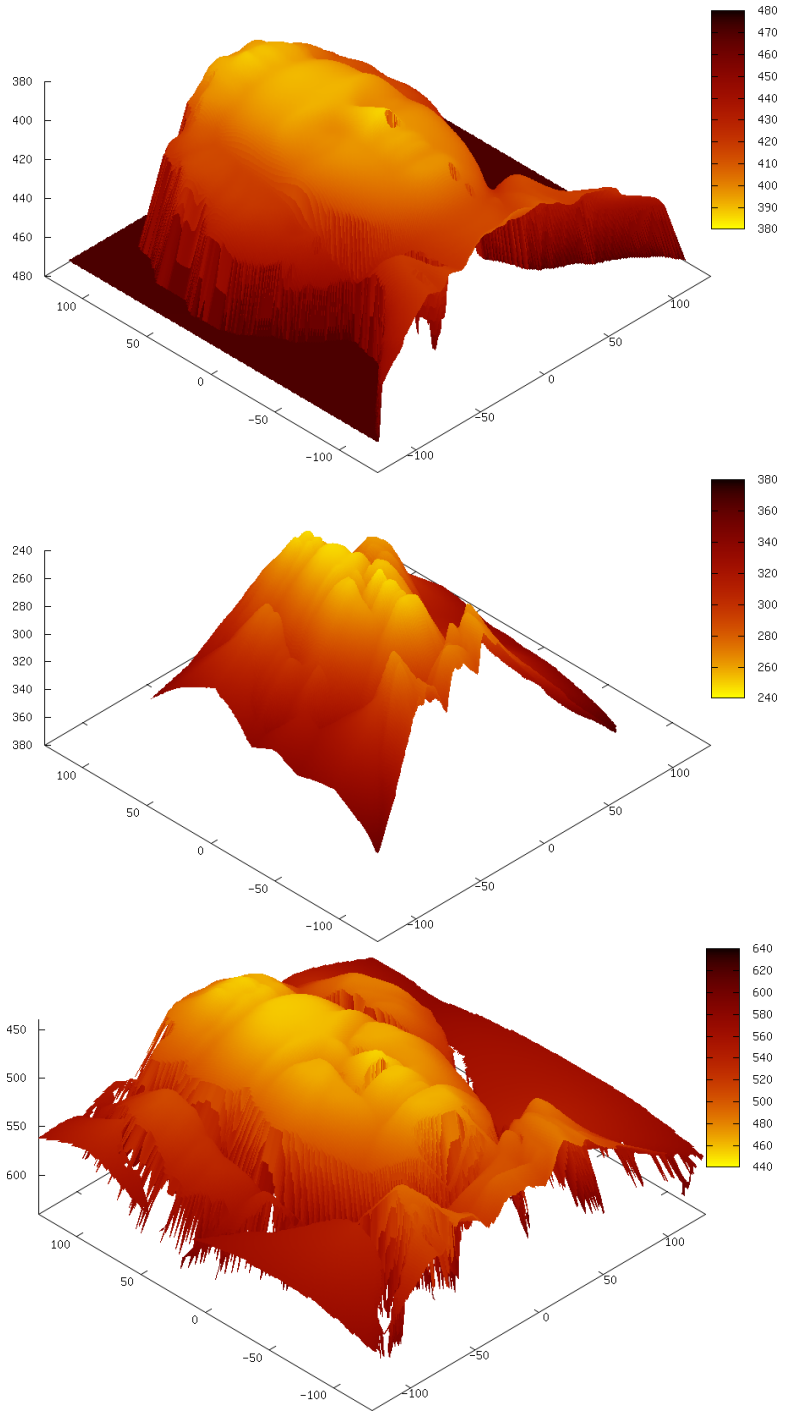

Figure 2.53: Reconstruction of the Mozart face for $k_d = 0.4, k_s = 0.6, \alpha = 10$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

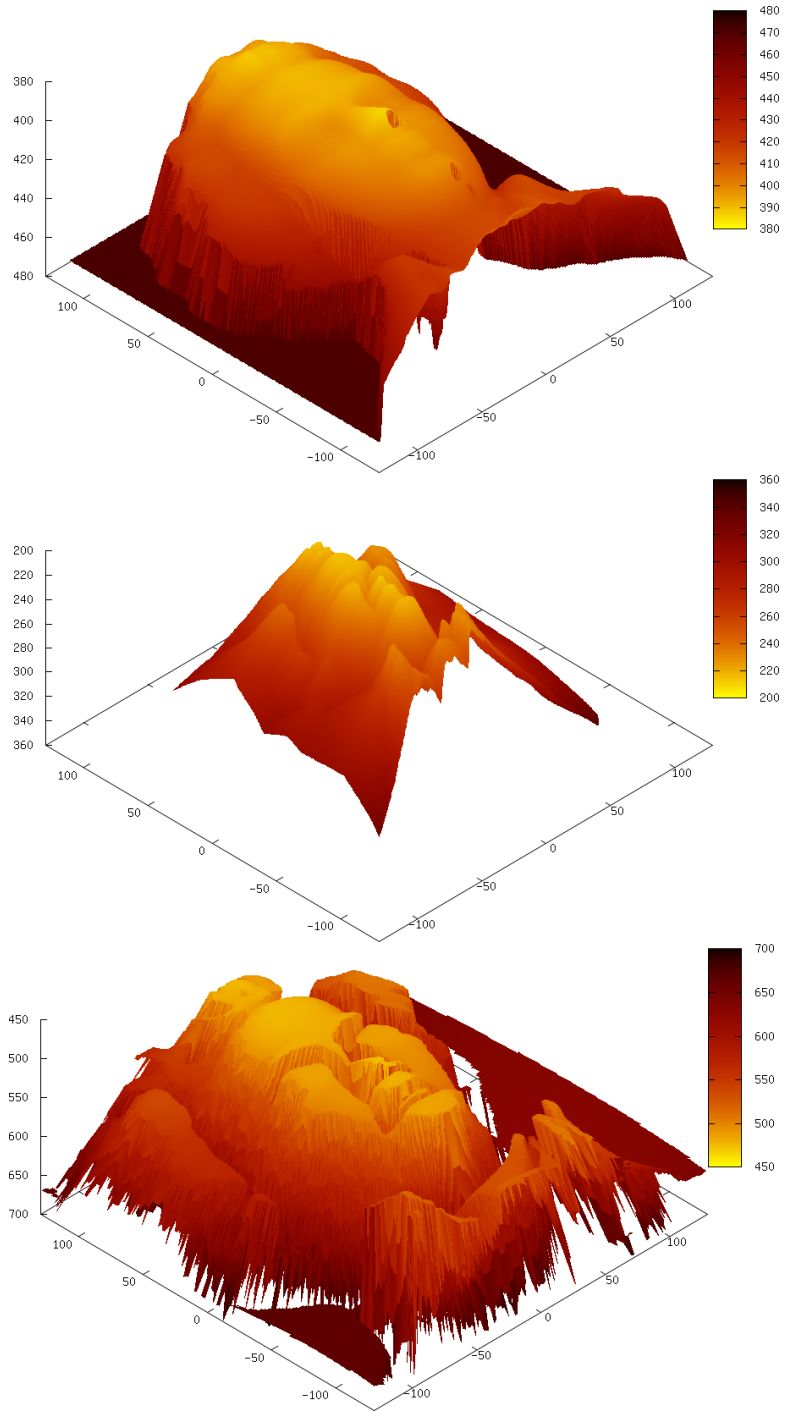

Figure 2.54: Reconstruction of the Mozart face for $k_d = 0.3, k_s = 0.7, \alpha = 10$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

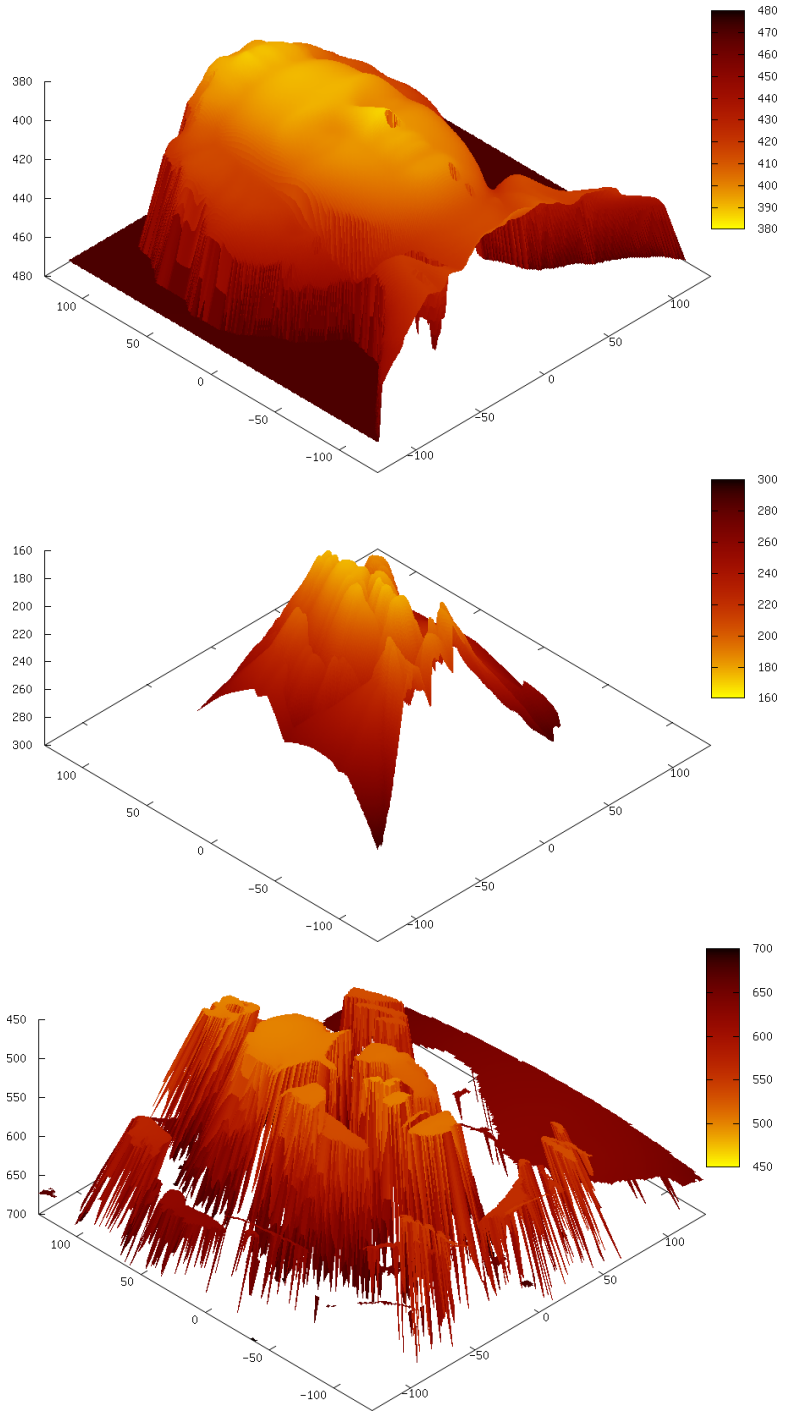

Figure 2.55: Reconstruction of the Mozart face for $k_d = 0.2, k_s = 0.8, \alpha = 10$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

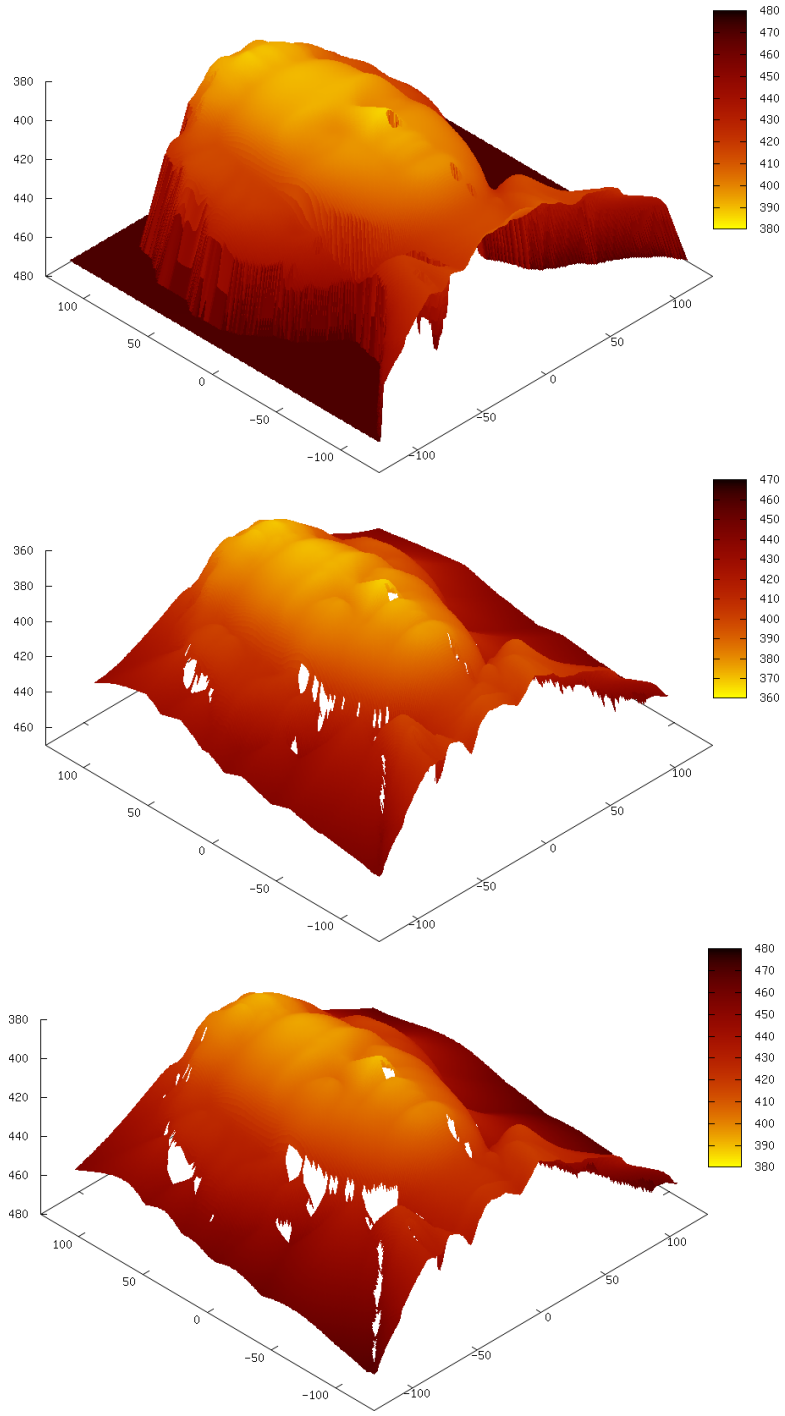

Figure 2.56: Reconstruction of the Mozart face for $k_d = 0.9, k_s = 0.1, \alpha = 20$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

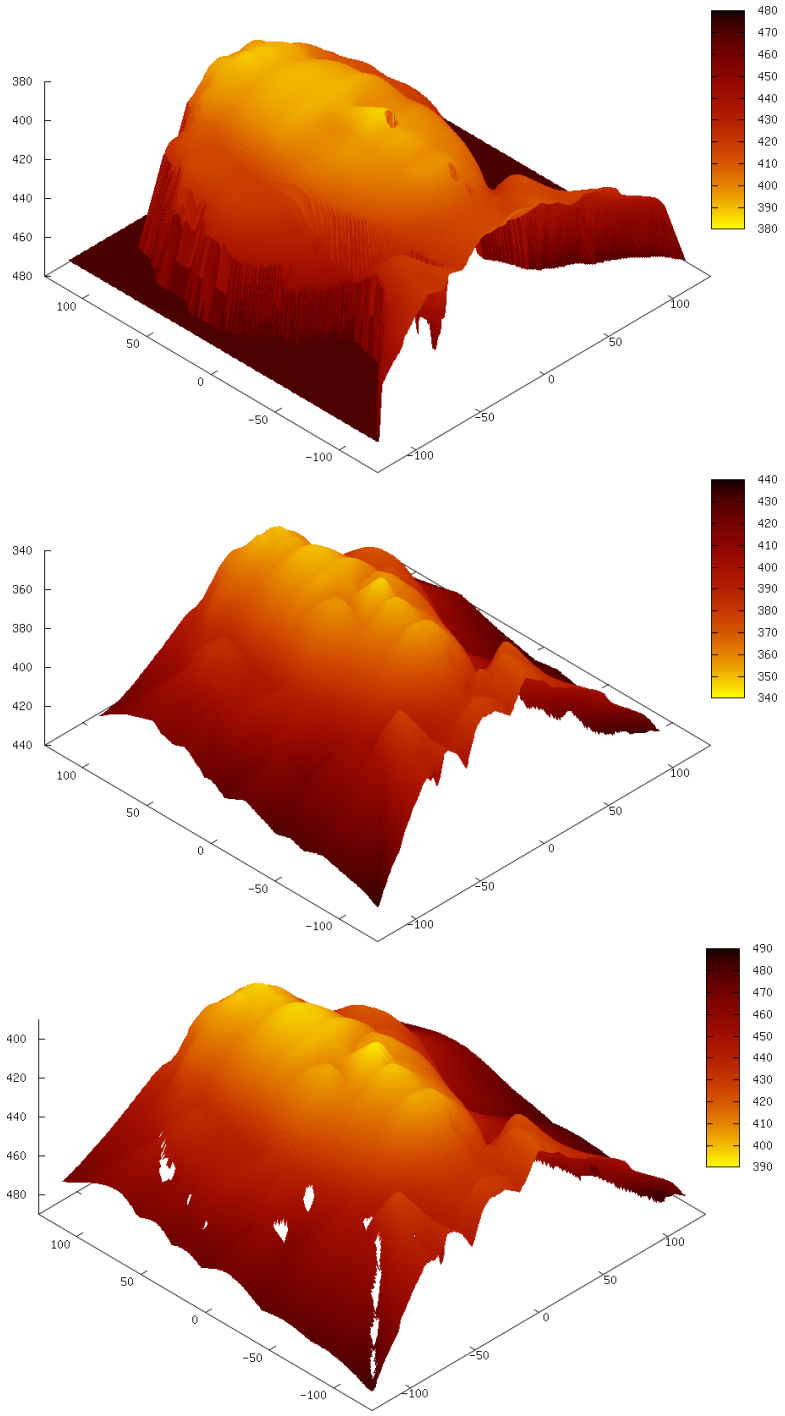

Figure 2.57: Reconstruction of the Mozart face for $k_d = 0.8, k_s = 0.2, \alpha = 20$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

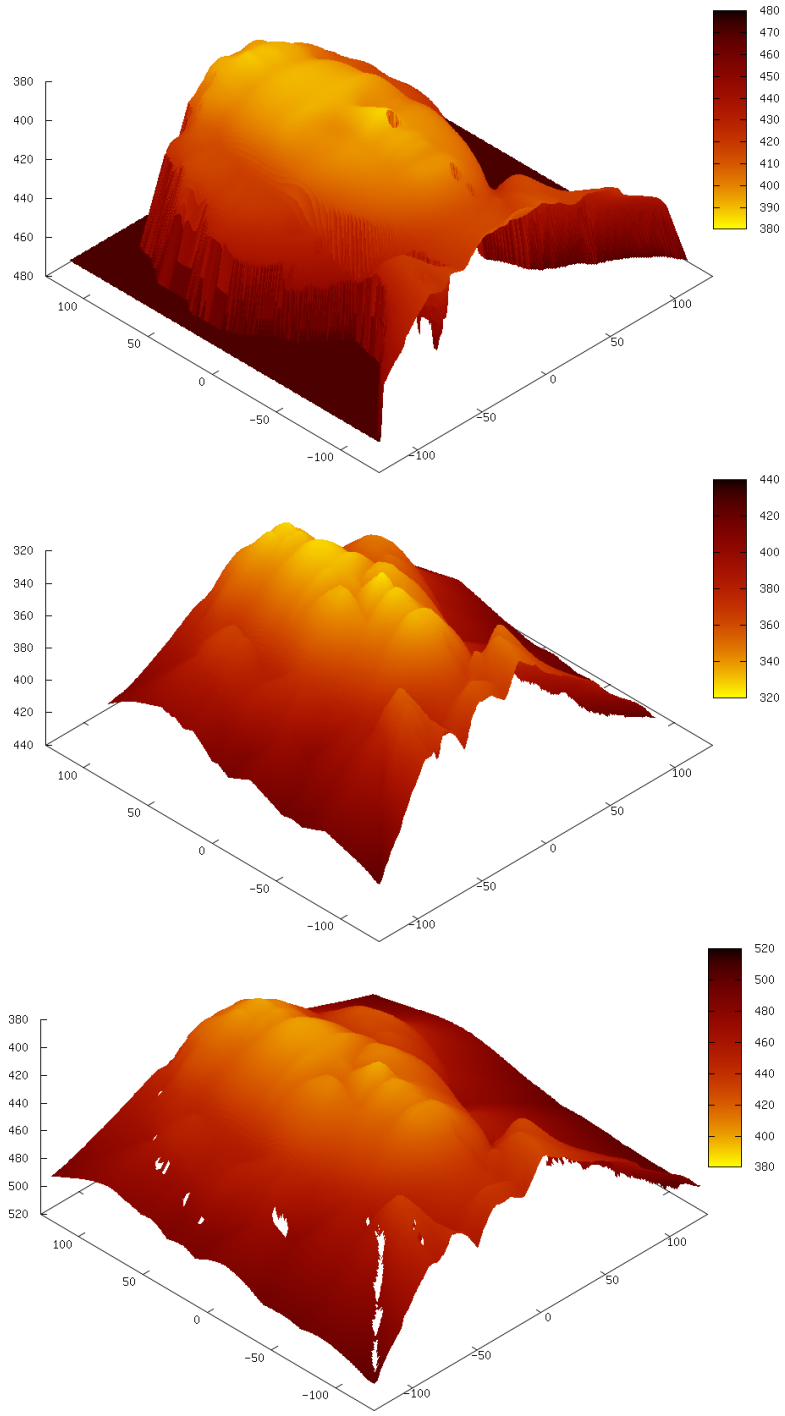

Figure 2.58: Reconstruction of the Mozart face for $k_d = 0.7, k_s = 0.3, \alpha = 20$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

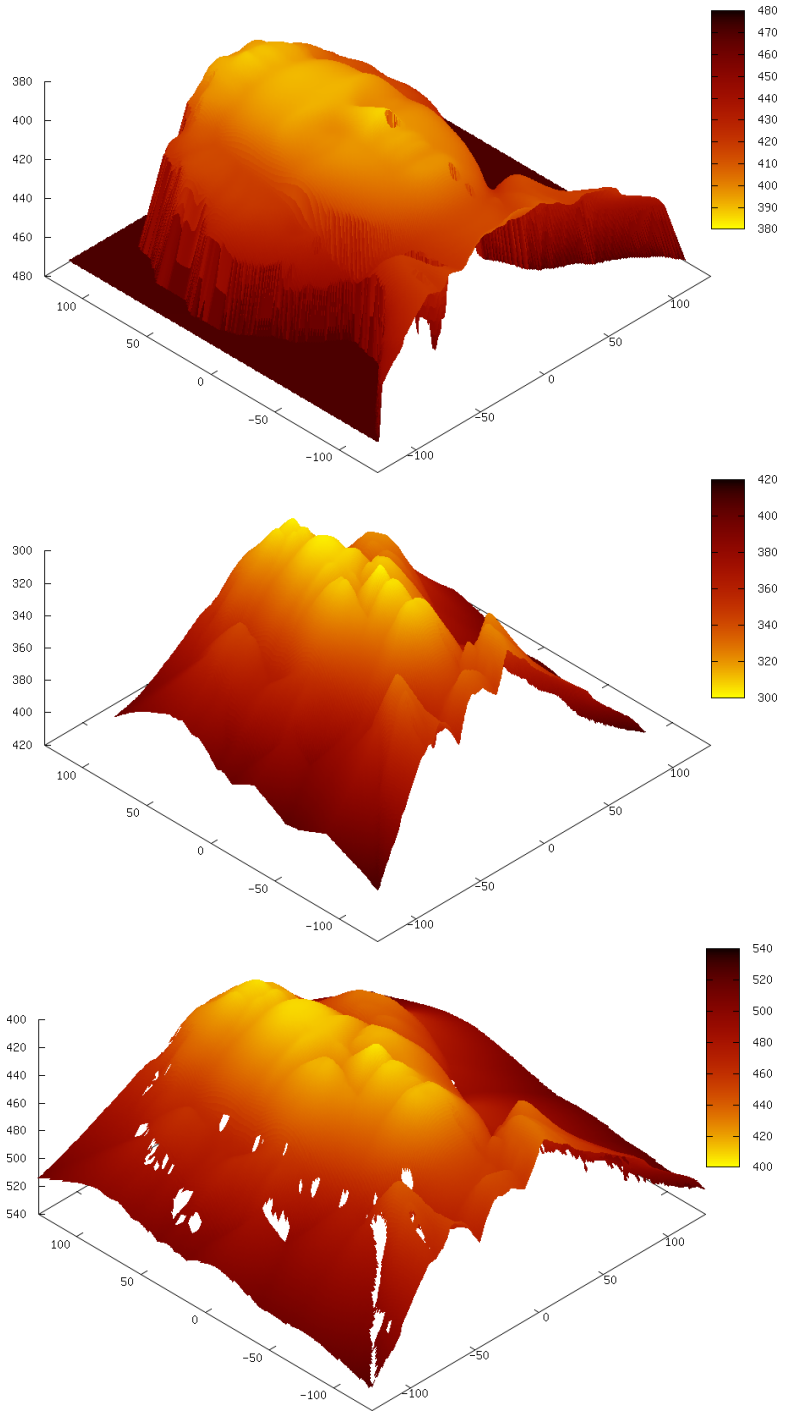

Figure 2.59: Reconstruction of the Mozart face for $k_d = 0.6, k_s = 0.4, \alpha = 20$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

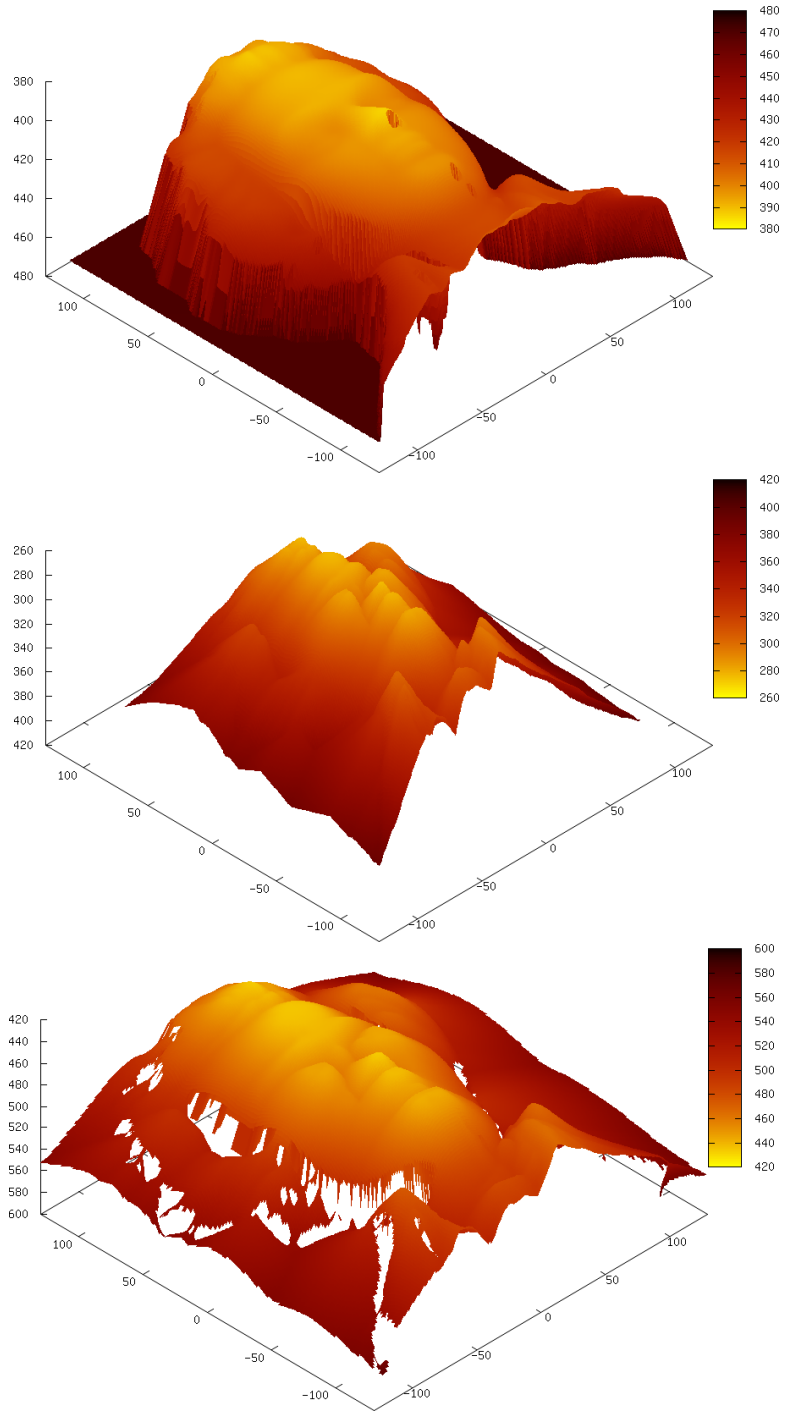

Figure 2.60: Reconstruction of the Mozart face for $k_d = 0.5, k_s = 0.5, \alpha = 20$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

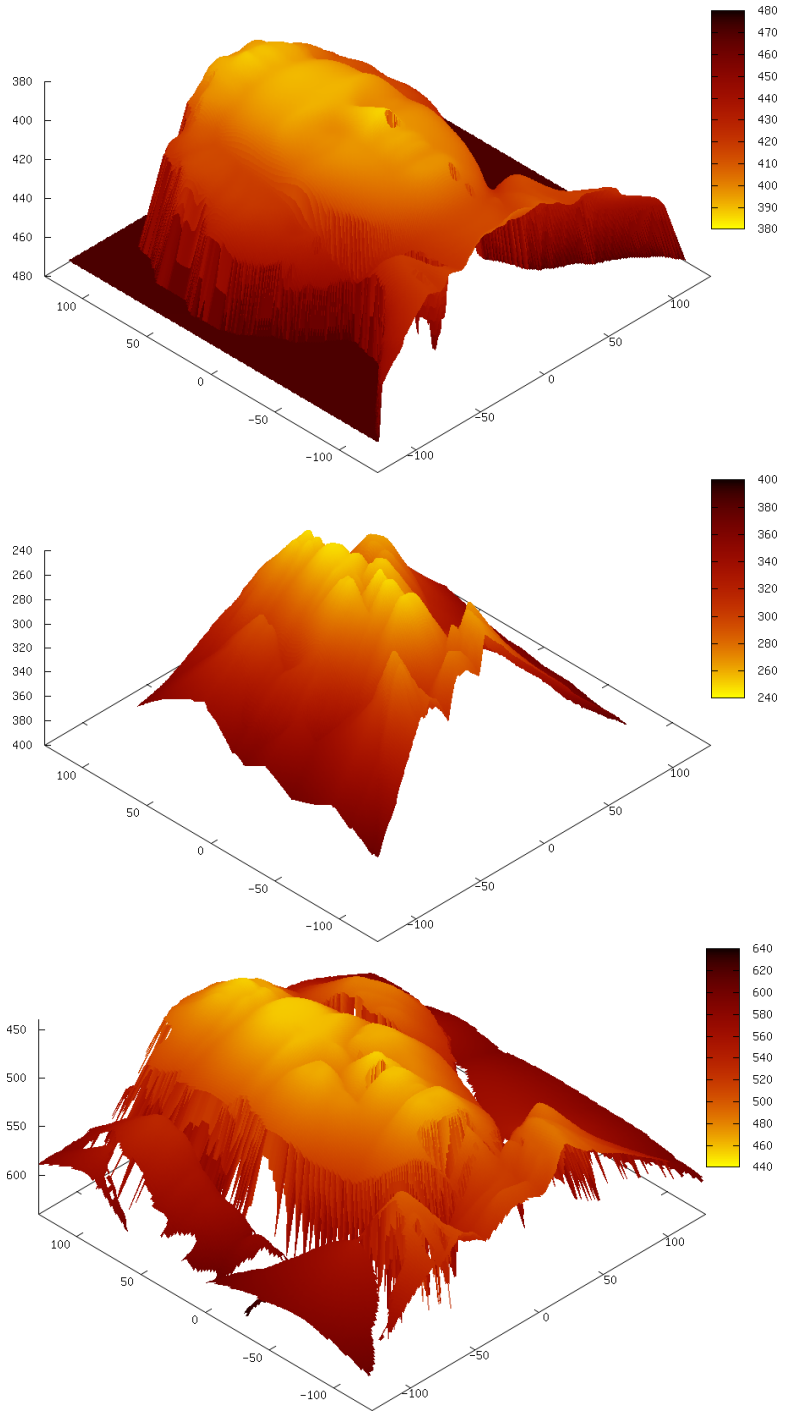

Figure 2.61: Reconstruction of the Mozart face for $k_d = 0.4, k_s = 0.6, \alpha = 20$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

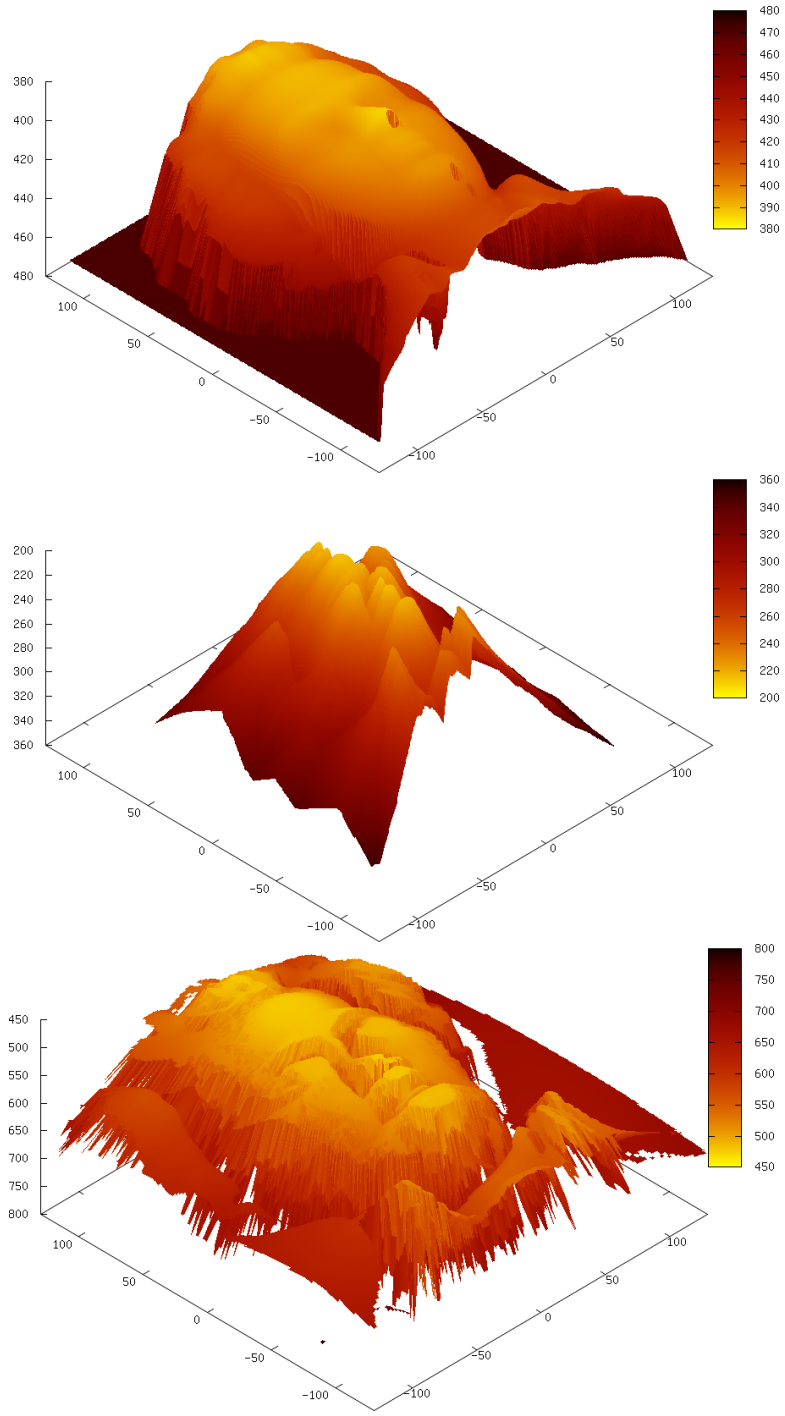

Figure 2.62: Reconstruction of the Mozart face for $k_d = 0.3, k_s = 0.7, \alpha = 20$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

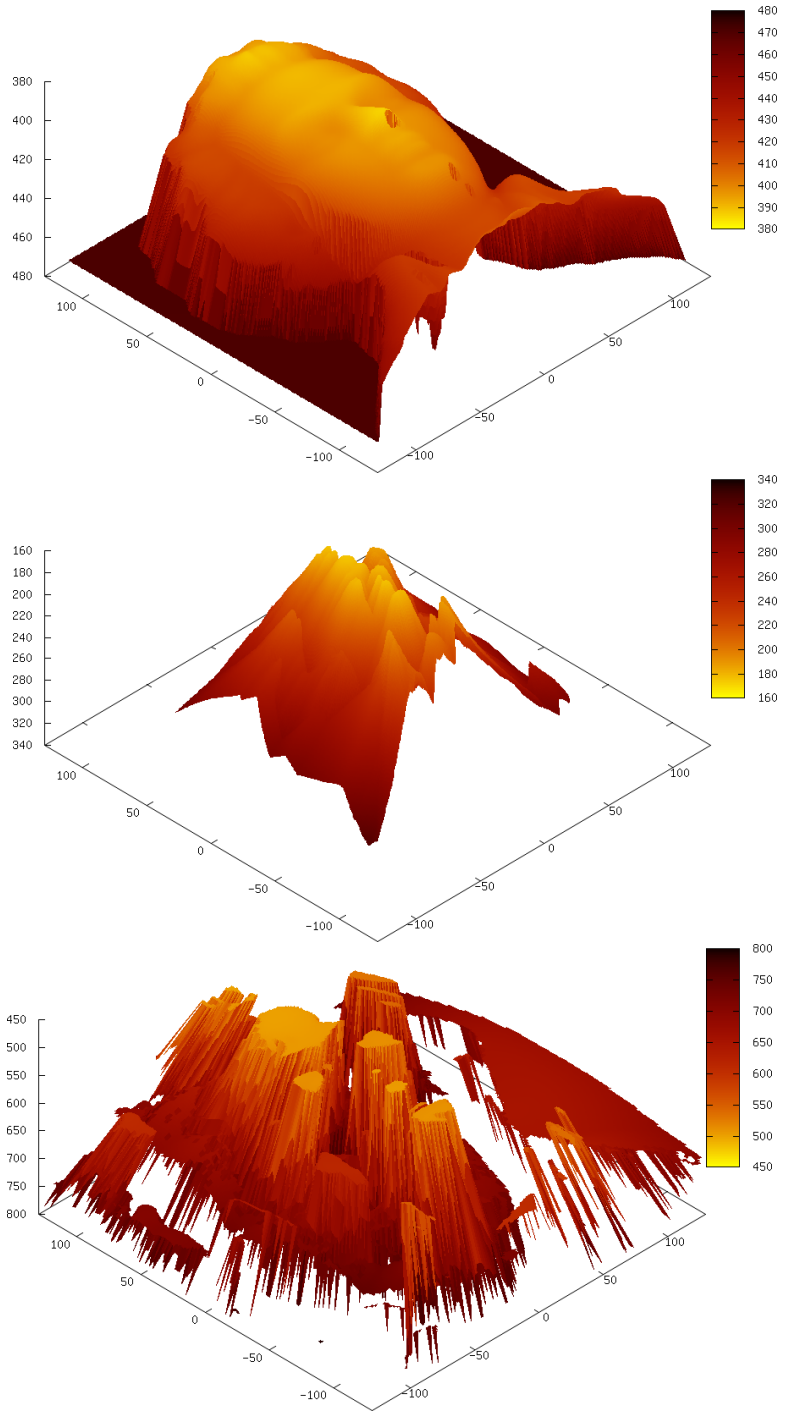

Figure 2.63: Reconstruction of the Mozart face for $k_d = 0.2, k_s = 0.8, \alpha = 20$. Top: Ground truth. Middle: Reconstruction using the Lambertian model. Bottom: Reconstruction using the Phong model.

2.9 Final Conclusions

In this chapter, a new SfS model with its corresponding PDE has been introduced, which allowed to model specular highlights. In the experimental evaluation, the effect of different parameters in the SfS models like focal length or specular highlights have been discussed. This leads to interesting conclusions on the difficulty of SfS methods for different types of images and the suitability of different models for these images.

However, the conclusions from this chapter cannot be final. Several effects, especially outliers at discontinuities, occurred, which might be caused by numerical method. Therefore, it is crucial to discuss numerical issues.

This will be done in the next chapter. Starting from the model PDE presented in this chapter, an efficient numerical method will be introduced. The chapter will also deal with performance issues, comparing the numerical method in this chapter to the numerical methods of Prados et al. and Cristiani et al. In addition to that, also other numerical methods based on other SfS methods will be briefly addressed.

Chapter 3

Numerical Methods for Shape from Shading

In this chapter, numerical methods for SfS are investigated. First, a review of the numerical paths explored for SfS methods since the 1970s is given. Afterwards, the focus lies on state-of-the-art techniques based on the model by Prados et al. [74]. A general numerical framework for the simplified model and the extension presented in the previous chapter is introduced. We will discuss the underlying PDE and do a numerical scale analysis to motivate the exact numerical implementation is given. Advanced techniques to accelerate the convergence of the method are presented, as well as a detailed discussion of the efficiency properties of the different numerical approaches and their benefits and drawbacks. Based on that, a novel, optimised method that combines the benefits of all these methods is proposed.

3.1 Contributions

There are several novelties in this chapter. The main contribution is a direct method for SfS, which solves directly the Hamilton-Jacobi equation of the Phong model introduced in the previous chapter. Combined with introducing several advanced acceleration techniques, an easy-to-code and fast method is obtained. The construction of the method is accompanied by a numerical scale analysis, which motivates several discretisation choices. Such an analysis is a novelty in image processing applications. Furthermore, the efficiency and convergence properties of different numerical schemes are investigated, which leads to interesting conclusions on the differences

between the methods and allows for another novelty, a hybrid method that combines different methods to gain an advantage in terms of performance and accuracy. Finally, the effect of discontinuities on the reconstruction quality is discussed and the relations of these effects to the discretisation are pointed out.

3.2 Hamilton-Jacobi equations and Hamilton-Jacobi-Bellman equations

Most modern SfS methods are based on solving *Hamilton-Jacobi* and *Hamilton-Jacobi-Bellman* equations.

A *Hamilton-Jacobi equation* (HJ equation) is a first-order partial differential equation for a function $u : \Omega \to \mathrm{R}$, $\Omega \subset \mathrm{R}^n$ open, of the form

$$H(x, u, Du)\,, \tag{3.1}$$

where $x \in \Omega$ and Du being the Jacobian of u. The function H is called a *Hamiltonian*. Generally, HJ equations are hyperbolic PDEs, i.e., they describe waves or transportation processes.

In addition to the PDE, suitable boundary conditions have to be imposed. Common boundary conditions for HJ equations are e.g. state constraint boundary conditions or von Neumann boundary conditions. These will be addressed later in this chapter.

Viscosity Solutions

Let

$$F(x, u(x), Du(x)) = 0 \tag{3.2}$$

with $x \in \Omega$ be a Hamilton-Jacobi equation.

The solutions of Hamilton-Jacobi equations are only defined in terms of *viscosity solutions*. First of all, one needs to define what a viscosity solution is.

Definition 3.1. A function $v \in C(\Omega)$ is called a *viscosity subsolution* of (3.2), if for any $\varphi \in C^1(\Omega)$, the inequality

$$F(x_0, u(x_0), D\varphi(x_0)) \leq 0 \tag{3.3}$$

holds at any local maximum point $x_0 \in \Omega$ of $u - \varphi$. u is called a *viscosity supersolution* of (3.2), if the inequality

$$F(x_0, u(x_0), D\varphi(x_0)) \geq 0 \tag{3.4}$$

holds at any local maximum point $x_0 \in \Omega$ of $u - \varphi$.

Finally, u is called a *viscosity solution* of (3.2), if is both a viscosity subsolution and a viscosity supersolution.

Whenever dealing with HJ equations, solving the PDE means looking for viscosity solutions of the PDE. The following proposition answers the question whether a viscosity solution actually solves the PDE.

Proposition 3.2. *1. If v is a viscosity solution of (3.2), then (3.2) is satisfied for every x where u is differentiable.*

2. If u is locally Lipschitz continuous and is a viscosity solution of (3.2), then (3.2) is satisfied almost everywhere in Ω.

The proof for this proposition can be found on page 31 of [4].

3.3 Discretisation

In order to present the basic components of the numerical method, we employ the following notation:

- $v_{i,j}^n$ denotes the approximation of $v\left(ih_1, jh_2, n\tau\right)$.
- i and j are the coordinates of the pixel (i, j) in x_1- and x_2-direction relative to the projection centre, respectively.

A key ingredient of any numerical method for a PDE is the discretisation of derivatives. In this section, we discuss the most important concepts for the discretisation of both derivatives in time and space.

Time Derivatives

Many SfS methods use iterative methods to solve the PDE. The methods discussed and introduced in this chapter follow the same approach. In most cases, this is done by introducing an artificial time into the PDE, e.g. the first-order HJ equation

$$0 = H(x, u, \nabla u) \tag{3.5}$$

which only depends on u and its spatial derivatives. Now, one introduced an additional dependency for u on a time t and constructs the time-dependent PDE

$$u_t = H(x, u, \nabla u)\,. \tag{3.6}$$

The latter PDE now describes an evolving process in time, with the limit of its solutions for $t \to \infty$ being the solution of the first PDE. By this, it is possible to construct an easy-to-code iterative scheme for the numerical solution for this PDE. For this, however, the temporal derivative u_t has to be discretised. For the methods discussed in this chapter, a standard Euler forward discretisation is sufficient. It reads as

$$u_t^n \approx \frac{u^{n+1} - u^n}{\tau} \tag{3.7}$$

at the n-th iteration step, i.e. at time τn for a suitable time step size $\tau > 0$. u^n denotes the unknown function u at iteration n or time τn. The choice of the time step size τ is not arbitrary. There usually is an upper bound for the time step size which is necessary to ensure stability of the numerical process. Such considerations will play a role later in this chapter.

Spatial Derivatives

Obviously, the PDE (3.6) not only depends on derivatives in time, but also on derivatives in space. Since the image data is only given on a discrete grid and so is the resulting depth field for SfS, it is necessary to find suitable approximations for spatial derivatives as well.

For now, all considerations will be done in $1-D$ only, a $1-D$ stencil is used for any partial derivative. For some methods, more complex derivative approximations that compute directional derivatives are used, but they shall not be discussed at this point.

The most simple approximation for a first derivative is an approximation using central differences:

$$u_x \approx \frac{u_{i+1} - u_{i-1}}{2h} \tag{3.8}$$

In this notation, h denotes the grid size. However, such a central approximation is usually not stable for hyperbolic PDEs. Another approach would be to use forward differences

$$u_x \approx \frac{u_{i+1} - u_i}{h} \tag{3.9}$$

or backward differences

$$u_x \approx \frac{u_{i+1} - u_i}{h}\,. \tag{3.10}$$

Using a one-sided derivative approximation in the right direction is a good and stable approximation for the purpose of solving hyperbolic PDEs.

However, the wrong one-sided differences may produce unstable results as well. It might vary over the image which one-sided differences are correct, i.e., just using forward or backward differences is not a good choice as well.

A good solution to this problem, an upwind discretisation, has been proposed for SfS by Rouy and Tourin in 1992 [77], which is based on [61]. They propose a technique to choose the right direction for one-sided differences. They propose to use the following derivative discretisation:

$$u_x = \frac{1}{h}\min\left\{0, u_{i+1} - u_i, u_{i-1} - u_i\right\} . \tag{3.11}$$

This derivative approximation takes the one-sided derivative in the direction from where the information is coming from. This is exactly the right approach for SfS applications. However, this is not a consistent derivative approximation. In the case of Eikonal equations in which Rouy and Tourin used this approximation, the approximation was actually consistent, since in that case, partial derivatives only appear squared. However, for SfS it is necessary to use partial derivatives that are not squared. In that case, the last expression in the minimum in (3.11) has the wrong sign. Therefore, if the minimum turns out to take the value of $u_{i-1} - u_i$, the sign needs to be reversed to $u_i - u_{i-1}$ in order to actually obtain a consistent derivative approximation. This upwind discretisation will be the basis of the numerical method introduced in this chapter.

3.4 Previous Numerical Methods

There exists a vast amount of literature on numerical methods for SfS in general. It would be far beyond the scope of this thesis to attempt a full review of the existing literature. Instead, we will concentrate on the most important papers here. This review of previous work should at least give a good impression on previous research in this field. Many of the numerical methods used for SfS come from the time when orthographic models have been used for SfS. Therefore the first part of this review will consist of a survey of the most important numerical concepts for classic SfS. Also, we try to give an overview on the most important numerical methods for perspective SfS. A special focus lies here on the most modern methods that are based on Prados' perspective model with light attenuation.

Readers interested in more references related to SfS may take a look at the website of Durou [25], who made a publication list with SfS-related papers including BibTeX entries for all of them.

3.5 Numerical Methods for Orthographic Shape from Shading

For orthographic SfS, a lot of numerical approaches have been employed. The first SfS method by Horn in 1970 [32] used characteristic strips to solve the problem. Essentially, the method consisted of solving a system of five ordinary differential equations. Details on this method can be found in [32, 33, 35, 38]. The characteristic strips method was one of very few non-iterative methods for orthographic SfS. In 1988, Bruckstein [15] proposed a much simpler contour-based non-iterative method which gave qualitatively similar results.

Classic SfS methods have been dominated by variational methods, usually with the following concept: The squared difference between brightness function of the surface and the grey values of the image is integrated over the whole image domain. This functional is minimised by solving its Euler-Lagrange equations. In general, the squared difference between brightness function and image grey value, which is often also called the data term, is not enough for obtaining a working method. For this reason, most authors use a so-called smoothness term, which employs additional constraints such as smoothness of the reconstructed surface. The best-known method of this type is the one by Horn and Brooks [36]. Ikeuchi and Horn [39] suggested a numerical strategy to avoid unbound gradients by applying stereographic projection to the gradient. A similar strategy has been done by Smith in

1973 [83], who used orthographic projection of the Gaussian sphere onto a plane tangent instead of stereographic projection. The last method has not been explicitly formulated as a variational approach, however, it can be rationalised as a variational method, as done by Horn and Brooks in [37]. In 1993, Vega and Yang published a method for improving the stability of the algorithm by Horn and Brooks by using a geometrical interpretation of this algorithm [93].

In essence, all these variational methods are based on the same numerical approach. Derivatives are mostly approximated using central differences. The only major difference lies in smoothness terms. For instance, Zheng and Chellappa introduced a method that employed an intensity gradient constraint instead of an assumption on the regularity of the surface gradient [108]. Malik and Maydan introduced a method that required not smoothness of the image, but only piecewise smoothness [52].

One issue that is often neglected in the discussion of classic SfS methods is the fact that the variational methods that formed the basis of classic SfS research did not actually reconstruct the depth of the surface. They only aimed at reconstructing the gradient of the depth, or the direction of the surface normal, which is equivalent. This solution is obtained by solving the Euler-Lagrange equations in the unknown functions p and q, which correspond to the x- and y-derivatives of the depth, respectively. To obtain the actual depth and thereby the shape of the surface, it was necessary to recover the depth from the gradient information. Methods that do so are called depth-from-gradient methods. The best-known such method is the one Frankot and Chellappa published in 1988 [28], which integrated the gradient field in the frequency domain. There is also more recent work on depth-from-gradient methods by Agrawal et al. [1], for more methods in this field see the references within that paper. Note these depth-from-gradient methods did not only serve the purpose to recover the final depth, it was also necessary to apply them after each or at least every couple of iterations, since the numerical methods yield inconsistent gradient fields which do not correspond to any surface, and the depth-from-gradient methods project the iterated solution to a legal gradient field.

One exception to this is a work by Horn and Brooks [37], who proposed a variational method for direct depth recovery. However, they obtained a divergent scheme. In [94,98], a variational direct method has been developed, which yields to a convergent scheme.

There are plenty other methods for classic SfS which are based on very different numerical ideas, like geometric aspects [51,62], conjugate gradient methods [48], only locally recovering the shape [49,63,103], linear approximation of the reflectance function [6,91], only to name a couple of examples.

The most comprehensive survey work on numerical methods for classic SfS up to the mid-1990s can be found in the paper of Zhang et al. [105], which also introduces the synthetic test surfaces that have been most frequently used in most SfS publications. However, the conclusion of said paper was that classic SfS would not work at all. This is not entirely true. In [94, 98], it is shown that using direct depth recovery, i.e. a numerical method directly aimed at recovering the depth without the need for a depth-from-gradient method and adaptive smoothness terms, it is possible to obtain visually convincing results even for orthographic SfS. The numerical method was, as traditional numerical approaches, based on minimising an energy functional using the Euler-Lagrange formalism. There are other methods like the one by Ahmed and Farag [2], which use an optimal control formalism based on the numerical framework Prados used for his modern perspective method [67] for classic SfS. They also obtain reasonable results.

3.6 Perspective Shape from Shading and Optimal Control Approaches

Since the 1990s, several groups came up with optimal control approaches for SfS. Based on the dynamic programming principle [5], an optimal control method introduces a cost functional, which is a functional of the state and certain control variables. The method then consists of finding a path that minimises this cost functional. For SfS applications, this can be interpreted as paths towards the solution. In the relevant case for SfS, the optimal control in each point always corresponds with the direction of the characteristic line, i.e. the direction into which information is being propagated. Generally, most SfS methods using optimal control approaches are based on a perspective model.

The first to come up with an optimal control approach have been Dupuis and Oliensis between 1991 and 1994 [23,24,56–60]. However, about the same time, Rouy and Tourin proposed their method for perspective SfS solving a Hamilton-Jacobi-Bellman equation [77].

However, not all SfS method use optimal control approaches. In particular, there are some publications dealing with discontinuous surfaces. For example, in 1992 Kimmel and Bruckstein [42] used techniques from differential geometry, fluid dynamics, and numerical analysis an orthographic method that can deal with discontinuous surfaces. In 2000, Soravia [84] discussed SfS with discontinuous intensity functions by solving HJ-equations with discontinuous coefficients. Another perspective SfS method which finds the

surface by synthesis of the image has been proposed by Courteille et al. in 2004 [19].

A very advanced numerical scheme for perspective SfS without light attenuation has been presented by Tankus et al. in 2003 [86–90]. They use a fast marching (FM) method for SfS, which is a very efficient solver. FM methods will be discussed in further detail in the next chapter.

There exist survey papers on perspective SfS, naturally mainly dealing with optimal control approaches, such as the work by Falcone et al. from 2003 [27] and the more general survey of Durou et al. [26].

The most successful optimal control approaches, however, are those using the perspective SfS model with light attenuation by Prados. A review of the most important numerical schemes based on this model can be found in [8, 12]. These methods are also the most influential previous numerical approaches for perspective SfS in general.

Prados' Approach

The most influential optimal control method goes back to Prados et al. [67, 74]. Based on work by Rouy and Tourin [77], they proposed a framework for SfS methods. While the main advance for SfS lies in the improved model, the numerical method to solve it is also interesting. By straightforward application of the brightness equation in the perspective case, one obtains a Hamilton-Jacobi equation. By means of a Legendre transform, this can be transformed into a so-called Hamilton-Jacobi-Bellman (HJB) equation, which is also a hyperbolic PDE. However, the HJB equation for Lambertian perspective SfS with light attenuation derived by Prados contains a supremum over the unit circle in the Hamiltonian. The point in the unit circle at which this supremum, which is actually a maximum since the unit circle is compact, is attained is called the optimal control. In essence, they use an upwind discretisation in space and introduce an artificial time and solve the HJB equation iteratively. However, a crucial part in their numerical part is the determination of the optimal control in each pixel in each iteration step. This is done by analytically finding extremal points in the unit circle by standard calculus. This is quite an expensive step, though. As done in the direct iterative method presented in this chapter, they use a Gauß-Seidel-like solver to accelerate convergence. For more details on optimal control methods, cf. [4, 47].

Cristiani's Approach

As Prados' approach, the approach by Cristiani et al. [22] is based on the perspective SfS model with light attenuation. As Prados' method, it is an optimal control approach. However, while Prados computes the optimal control within the unit circle analytically, Cristiani samples the unit circle and determines the optimal control by comparing these sampled points. In addition to that, he uses a different derivative approximation, a semi-Lagrangian method [21], which takes upwind derivatives in the direction of the optimal control instead of the principal axes of the coordinate system. This increases the rotational invariance of the method and is generally a little more stable. Their results are competitive with the ones of Prados' method, although on close inspection, small artefacts caused by the sampling can occur. A finer sampling can reduce this effect, but at the expense of a higher computational effort. In a standard configuration, their method is a bit faster than Prados' method.

Farag's Approach

The work by Farag et al. [2] is numerically very similar to Prados' work. The main difference for this method lies on the model side, which has already been discussed in the previous chapter. However, with their different model, they use a different derivative approximation as well. Instead of an upwind discretisation, they use a Lax-Friedrichs scheme. This scheme has, compared to an upwind scheme, the major drawback that it is much more diffusive, making the reconstruction of hard edges more difficult. Its advantage lies in the applicability to a larger range of Hamiltonians, in particular many non-convex Hamiltonians, which is of advantage for their different model. Except this detail, their method is practically identical to Prados' framework. A major drawback of their numerical method, though, is that it is not suited for employing in a fast marching method, since such methods rely on upwind-type derivative discretisation. Fast marching methods will be the focus of the next chapter.

3.7 Iteratively solving the PDE

Regardless of the Hamiltonian being used, by introducing an artificial time as in (3.6) the iterative scheme

$$u^{n+1} = u^n - \tau H(x, u, \nabla u) \tag{3.12}$$

is obtained.

This equation is not fully discretised yet. Besides a suitable choice for the spatial derivative discretisation, which has been discussed in Section 3.3, it must be specified from which iteration step the data of u in the Hamiltonian is taken. The simplest choice for this would be an explicit time discretisation. In an explicit scheme, all u data in the Hamiltonian is taken from the old time slice, which results in the scheme

$$u^{n+1} = u^n - \tau H(x, u^n, \nabla u^n)\,. \tag{3.13}$$

An alternative scheme would be a fully implicit scheme, reading as

$$u^{n+1} = u^n - \tau H(x, u^{n+1}, \nabla u^{n+1})\,. \tag{3.14}$$

Generally, implicit schemes are more stable than explicit schemes. However, they are computationally much more involved, since depending on the complexity of the Hamiltonian, highly nonlinear systems of equations have to be solved. Fortunately, there is a third way: Take some parts from the new time slice $n+1$ and others from the old one. This is called a semi-implicit scheme. Obviously, depending on the structure of the Hamiltonian, there might be different ways of doing that.

Before actually proposing a discretisation for the SfS PDE and reviewing the discretisations that were chosen for the optimal control methods for the Lambertian model, a numerical scale analysis of the Phong SfS PDE (2.15) will be done in the next section. This analysis will give insight into numerical properties of the PDE and will give indication for good choices of the discretisation.

Table 3.1: Units

Term	Unit
$[\mathsf{f}], [x_1], [x_2]$	m
$[v], [k_a], [k_d], [k_s], [I_d], [I_s]$	1
$[v_x], [v_y]$	$1/m$
$[I], [I_a]$	$1/m^2$

3.8 Numerical Scale Analysis of the Shape from Shading Equation

In this section, a numerical scale analysis of the SfS PDE for the Phong model (2.15) is performed. While this is a known technique e.g. in the context of computational fluid dynamics [53], applying it in a computer vision context is a novelty. Certainly, such an analysis is not restricted to the problem of SfS and will have merit in other image processing and computer vision contexts as well to motivate, analyse, and improve numerical methods.

In a numerical scale analysis, the variables of interest are expressed via a reference value times a normalised dimensionless new variable. This may help in assessing the relationship between occurring terms and in understanding of the behaviour of numerical solutions. It also helps to find a suitable constraint on the size of the time step.

Dimension of the Model

As a side effect of such a proceeding, it is possible to check if the dimensions in a new model are put together correctly. Table 3.1 shows the units of the occurring terms in the PDE (2.15). It follows easily, that $[Q] = 1$ and $[J] = \frac{1}{m^2} m^2 = 1$. For $[W]$ one obtains

$$[W] = \sqrt{\frac{m^2}{m^2} + \left(\frac{m}{m} + \frac{m}{m}\right)^2 + 1^2} = \sqrt{1^2} = 1 \tag{3.15}$$

so that the composition of terms in the PDE (2.15) is dimensionless, which is perfectly correct.

Notations

In the following investigations scaling in the point location $x = (x_1, x_2)^T$ and in the focal length f is employed. More specifically, the notation

$$x := x_{\text{ref}} \cdot \hat{x} \qquad \text{and} \qquad \mathsf{f} := f_{\text{ref}} \cdot \hat{f} \tag{3.16}$$

with the obvious reference values x_{ref} and f_{ref} is used, and where $\hat{x} = (\hat{x}_1, \hat{x}_2)^T$, $\hat{f}$ are the scaled variables.

Note that the reference values $\hat{x}_1$, $\hat{x}_2$ and $\hat{f}$ should be understood as multiples of a characteristic pixel size h, compare the dimensions given in Table 3.1.

Also, define the notion of a squared ratio via

$$\gamma := \frac{x_{\text{ref}}^2}{f_{\text{ref}}^2}\,. \tag{3.17}$$

The Diffuse Term

In the first part of the analysis, the main part of the diffuse part of the PDE is discussed. In this analysis, the diffuse source term is not considered and will be the subject of a later paragraph.

Lemma 3.3. *For the term $J(x)W(x)$ of the model (2.15) holds*

$$J(x)W(x) \;=\; f_{\text{ref}}^3 \cdot \frac{(I(x) - k_a I_a)\,\hat{f}^2}{\hat{Q}_\gamma} \cdot W_\gamma \tag{3.18}$$

where

$$\hat{Q}_\gamma \;:=\; \frac{\hat{f}}{\sqrt{|\hat{x}|^2 \cdot \gamma + \hat{f}^2}}\,, \quad W_\gamma \;:=\; \sqrt{\hat{f}^2 |\nabla v|^2 + \gamma (\nabla v \cdot \hat{x})^2 + \frac{1}{f_{\text{ref}}^2} \cdot \hat{Q}_\gamma^2}\,. \tag{3.19}$$

Proof. The proof consists of two parts, one for J and one for W.

For $J(x)$ one obtains

$$\begin{aligned}
&\frac{(I(x) - k_a I_a)\,\mathsf{f}^2}{Q(x)} \\
&= (I(x) - k_a I_a) \left(f_{\text{ref}} \cdot \hat{f}\right)^2 \frac{\sqrt{|x_{\text{ref}} \cdot \hat{x}|^2 + \left(f_{\text{ref}} \cdot \hat{f}\right)^2}}{f_{\text{ref}} \cdot \hat{f}} && (3.20)\\
&= f_{\text{ref}}\,(I(x) - k_a I_a)\,\hat{f}^2 \cdot \frac{\sqrt{|\hat{x}|^2 \cdot x_{\text{ref}}^2 + \hat{f}^2 \cdot f_{\text{ref}}^2}}{\hat{f}} && (3.21)\\
&= f_{\text{ref}}^2 \cdot \frac{(I(x) - k_a I_a)\,\hat{f}^2}{\hat{Q}_\gamma}\,. && (3.22)
\end{aligned}$$

We now turn to the term $W(x)$. Plugging in the terms from (3.16), a simple computation gives

$$\begin{aligned}
&\sqrt{\mathsf{f}^2 |\nabla v|^2 + (\nabla v \cdot x)^2 + Q(x)^2} \\
&= \sqrt{\left(f_{\text{ref}} \cdot \hat{f}\right)^2 |\nabla v|^2 + (x_{\text{ref}} \nabla v \cdot \hat{x})^2 + \frac{\left(f_{\text{ref}} \cdot \hat{f}\right)^2}{|x_{\text{ref}} \cdot \hat{x}|^2 + \left(f_{\text{ref}} \cdot \hat{f}\right)^2}} && (3.23)\\
&= f_{\text{ref}} \cdot W_\gamma\,. && (3.24)
\end{aligned}$$

The multiplication of the results from (3.22) and (3.24) concludes the proof. □

Consider now the *range of variables* in (3.18) and (3.19):

- $(I(x) - k_a I_a)$ is in $[0, 1]$.
- $\hat{f}$ is in $(0, 1]$; one may set $\hat{f} = 1$.
- $\hat{x}_1$, $\hat{x}_2$ are in $[-1, 1]$.
- γ is typically in the range of $\left[\frac{1}{2}, 1\right]$ in the experiments. This implies $\hat{Q}_\gamma \in (0, 1]$.
- f_{ref} is typically of the order 10^2 to 10^6 (times h) in the experiments.

These aspects lead to some *conclusions*:

(a) By the factor f_{ref}^3 it becomes clear that the PDE is quite stiff. Moreover, this problem becomes even more significant for large focal lengths.

(b) The term $1/f_{\text{ref}}^2 \cdot \hat{Q}_\gamma^2$ is virtually unimportant. Its strongest contribution is close to the projection centre, roughly decreasing quadratically. In the experiments, this means it decreases to approximately $10^{-1} \cdot 1/f_{\text{ref}}^2$ with the distance from the centre.

(c) The factor $1/\hat{Q}_\gamma$ grows from 1 at the image centre to $(2\gamma + \hat{f}^2)/\hat{f}$ at the image boundary. In the experiments, this is roughly a factor of $2-3$ for $\hat{x}$ near the image boundary.

(d) If $\hat{x}$ is close to the image corners, i.e., for $\hat{x} \approx (\pm 1, \pm 1)^T$, one obtains a notable contribution from the term $\gamma(\nabla v \cdot \hat{x})^2$.

These conclusions show that the spatially dependent contributions involved in the part $J(x)W(x)$ of the PDE are important close to the boundary, see (c)-(d). Moreover, the factor f_{ref}^3 appears in (3.18) which indicates a strong contribution of the Lambertian part within the Phong SfS model (2.15). The term $1/f_{\text{ref}}^2 \cdot \hat{Q}_\gamma^2$ is of negligible size compared to all other contributions.

The Specular Term

Now the contribution by the specular term is considered.

Lemma 3.4. *For*

$$A := \left(\frac{2Q(x)^2}{W(x)^2} - 1 \right)^\alpha \tag{3.25}$$

it holds $A \in [0, 1]$.

Proof. Rewrite the term below the bracket yields

$$\frac{2Q^2}{W^2} - 1 = \frac{2Q^2 - \mathsf{f}^2|\nabla v|^2 + (\nabla v \cdot x)^2 + Q^2}{W^2} = \frac{Q^2 - \mathsf{f}^2|\nabla v|^2 + (\nabla v \cdot x)^2}{W^2}. \tag{3.26}$$

Since it is possible to draw a factor f_{ref}^2 out of the nominator and the denominator in (3.26), in total no contribution in terms of a factor f_{ref} arises. By (2.17) holds $A \in [-1, 1]$. Then $A \in [0, 1]$ follows since a negative number will be mapped to zero by the Phong reflectance rule. □

Slightly rearranging the specular term from (2.15) gives

$$k_s I_s \exp(-2v(x)) A \cdot \frac{W(x)}{Q(x)}. \tag{3.27}$$

Thus, the specular term is similar to $J(x)W(x)$, see (2.15)-(2.17), which can be written as

$$J(x)W(x) \;=\; (I(x) - k_a I_a)\mathsf{f}^2 \cdot \frac{W(x)}{Q(x)}\,. \tag{3.28}$$

The result of a comparison of terms in (3.18) and (3.28) is then summarised via

Proposition 3.5. *For the specular term of the Phong SfS model (2.15)-(2.17) holds*

$$k_s I_s \exp\left(-2v(x)\right) A \cdot \frac{W(x)}{Q(x)} \;=\; f_{\text{ref}} \exp\left(-2v(x)\right) \cdot \frac{k_s I_s}{\hat{Q}_\gamma} A \cdot W_\gamma\,. \tag{3.29}$$

A comparison of results in Lemma 3.3 (setting $\hat{f} := 1$) and Proposition 3.5 reveals an important mechanism of the Phong-based SfS model. The factor f_{ref}^3 in (3.18) is considerably larger than the factor f_{ref} in (3.29). This implies that only at highlights an important contribution due to the specular term is obtained, since only there the factor $\exp(-2v)$ in (3.29) is of the same order of magnitude as f_{ref}^2. In the other parts of the image, the terms also occuring in the Lambertian SfS model dominate the setting.

Note also that this qualitative assessment relies on the magnitude of k_d and k_s.

The Diffuse Source Term

Now turn to the *diffuse source term*, $k_d I_d \exp(-2v(x))$ from (2.15). Also in this case, an important contribution may only arise at highlights. Moreover, no factor f_{ref} arises as in (3.29). Thus, the diffuse source term is rather small compared to the other parts of (2.15).

3.9 The Numerical Method

For actually solving the PDE, the method of artificial time as described in Section 3.3 is used.

Dropping for abbreviation the dependence on spatial and temporal variables, the augmented, time-dependent PDE reading as

$$v_t = \underbrace{f_{\text{ref}}^{-3} JW - f_{\text{ref}}^{-3} \frac{W k_s I_s}{Q} \exp(-2v) \left(\frac{2Q^2}{W^2} - 1 \right)^{\alpha}}_{=: B} - f_{\text{ref}}^{-3} k_d I_d \exp(-2v) \tag{3.30}$$

is to be solved.

In defining τ, the knowledge from the numerical scale analysis and computation in a *pointwise* fashion is used, i.e., $\tau = \tau_{i,j}$. The reason for this is the spatially dependent factor analysed in the previous section, see the conclusions (c) and (d) there. If we would define τ as a constant number globally for the whole image, it would need to satisfy the stability condition near the image boundary, which is the critical region in this context. This would come at the price of converging very slowly in the centre region of the image. Since the aim is to compute the steady state defined by $v_t = 0$, a pointwise choice of τ is feasible.

The discretisation of $I(x)$ and $Q(x)$ is simple as these terms can be evaluated pointwise at all pixels (i, j). This naturally results in a consistent approximation of these terms.

The Source Terms

Being a factor within the part of the scheme where upwinding is employed, $k_s I_s e^{-2v}$ is discretised at pixel (i, j) using the known data at point (i, j), i.e., setting $k_s I_s e^{-2v_{i,j}^n}$. This is also adequate as the corresponding specular term in practice only yields notable contributions at highlights; compare the discussion after Proposition 3.5.

Finally, consider the diffuse source term $f_{\text{ref}}^{-3} k_d I_d e^{-2v}$. Source terms like this typically enforce the use of very small time step sizes when evaluated explicitly, leading to very long computational times. However, in the numerical scale analysis it became clear that this source term generally yields a very small contribution, and even at highlights its contribution is of one order in f_{ref} lower than the specular contribution. Thus, it can be discretised it in an explicit way:

$$f_{\text{ref}}^{-3} k_d I_d e^{-2v(x,t)}|_{(x,t)=(ih_1, jh_2, n\tau)} \approx f_{\text{ref}}^{-3} k_d I_d e^{-2v_{i,j}^n} . \tag{3.31}$$

This is in contrast to previous works [67, 96] where the intuition to discretise source terms implicitly was employed, and it gives a computational advantage w.r.t. efficiency.

Scheme summary. The scheme and its obvious properties by construction can be summarised via

Proposition 3.6. *Let $\hat{B}$ denote the discretised version of term B from (3.30). Then one obtains pixelwise the consistent discretisation*

$$v_{i,j}^{n+1} = v_{i,j}^n - \tau\hat{B} - \tau f_{\text{ref}}^{-3} k_d I_d e^{-2v_{i,j}^n} . \tag{3.32}$$

As the method is explicit, there is a stability condition on the time step size [18].

3.10 Gauß-Seidel-type Iteration

At pixel (i, j) the data

$$\begin{array}{ccc} & v_{i,j+1} & \\ v_{i-1,j} & v_{i,j} & v_{i+1,j} \\ & v_{i,j-1} & \end{array} \tag{3.33}$$

are used within the upwind differences. Consider for the moment that the iteration is done pixel-wise from left to right and, beginning with the top line, from top to bottom over the computational grid. Thus, ascending in i and descending in j, incorporate already computed values into the scheme. In the considered case, this gives

$$v_{x_1}(x,t)|_{(x,t)=(ih_1,jh_2,t)} \approx \min\left(0,\ \frac{v_{i+1,j}^n - v_{i,j}^n}{h_1},\ (-)\frac{v_{i-1,j}^{n+1} - v_{i,j}^n}{h_1}\right) \tag{3.34}$$

$$v_{x_2}(x,t)|_{(x,t)=(ih_1,jh_2,t)} \approx \min\left(0,\ \frac{v_{i,j+1}^{n+1} - v_{i,j}^n}{h_2},\ (-)\frac{v_{i,j-1}^n - v_{i,j}^n}{h_2}\right) \tag{3.35}$$

where $(-)$ means that this term needs to be multiplied by a factor -1 if it is the smallest term in the minimum expression.

3.11 Fast Sweeping

While a Gauß-Seidel-like technique generally speeds up convergence of a method compared to a fully explicit scheme, a Gauß-Seidel-like scheme

usually passes over an image always in the same direction. This can give a directional bias to the solution and have bad convergence properties based on the actual problem. If information needs to propagate against the direction in which the scheme passes over the image, there will be no advantage in the Gauß-Seidel-like approach any more compared to a fully explicit scheme. However, by changing the direction in which the image is processed, it is possible to overcome this problem. Such a technique is called *fast sweeping*.

Now, a sweeping technique adopted from the works [106, 107] is used. The reason behind the use of this technique is as follows. Staying within the Gauß-Seidel-type framework, in a pixel (i, j) updated values are always taken into account just from the left, see (3.34), and from above, see (3.35). Therefore, information is numerically propagated faster in the directions "from left to right" and "from top to bottom" as in all other possible directions. As a remedy, one may switch the iteration directions in the following cyclic fashion:

1. Left $\rightarrow$ Right, and Top $\rightarrow$ Bottom
2. Top $\rightarrow$ Bottom, and Right $\rightarrow$ Left
3. Right $\rightarrow$ Left, and Bottom $\rightarrow$ Top
4. Bottom $\rightarrow$ Top, and Left $\rightarrow$ Right

Exactly this procedure is called *sweeping*. Defining the sweepings as above different values $v_{i\pm1,j\pm1}^{n+1}$ are needed to be taken into account in (3.34)-(3.35) depending on the sweeping direction.

3.12 A Direct Method

As mentioned before, optimal control approaches for Prados' model aim at solving a Hamilton-Jacobi-Bellman equation which is derived from a Hamilton-Jacobi equation using the Legendre transform. In this section, a direct method will be presented that aims at solving the Hamilton-Jacobi equation directly. A natural step would be to do this first for the Lambertian model and then extend the numerics to the extended model presented in Section 2.5. This has been done in [96]. However, since the Lambertian model is only a special case of the Phong model, this section introduces the direct numerical method for the Phong model as well.

Cascading Multigrid

In addition to the sweeping and Gauß-Seidel-type technique, another way to accelerate the convergence speed of the method is a multigrid technique [7]. The principle can be explained quite easily. Since the computational complexity of the method is worse than linear in the number of pixels, it can be beneficial to sample the input image down to a coarser grid, solve the PDE on this coarser grid and use the result as an initialisation for the method on the actual grid. Certainly, this can be iterated to even coarser grids which are subsequently refined. For simplicity, only downsampling by a factor 2 in every dimension will be considered. Consequently, this method will only be evaluated on images with sizes of powers of two in every dimension.

Downsampling of the Image

A natural way of downsampling the image would be to use some standard interpolation technique like linear or cubic interpolation. However, for the application of SfS, this is clearly the wrong approach. Since greater depths generally correspond to darker pixels and the visual impression of the downsampled image is not important, the correct way to sample the image down is to take the minimum of the four pixels that are merged to one pixel. Therefore, just passing over the image and merging blocks of 2×2 pixels to one image in this way is the downsampling method used for this application.

In addition to the downsampling of the input image itself, the reconstruction parameters have to be adjusted. The best way to do that is to adjust the grid widths according to the downsampling. In fact, the only adjustment that is necessary when sampling an image down one level is to double the mesh widths h_1 and h_2. All other parameters remain the same.

Prolongating the Solution to a Finer Grid

A natural way of prolongating the solution on a coarse grid to the next finer grid would be to interpolate the solution. However, for SfS, this might not be correct. The worst thing that could happen is that the interpolated values are too low. For that reason, the most simple and generally correct way is just to repeat the value of a pixel to its neighbours. With that proceeding, no new minima in depth are created and the depth field remains consistent.

As it was necessary to double the mesh widths when switching to a coarser grid, it is equally important to halve the mesh widths when changing to a finer grid.

Iterating on Coarser Levels

The idea is clear: Iterate on a coarse level, then refine, iterate, refine, etc., unless the finest level is reached. One question, however, still remains: When to stop iterating on a coarse level? Basically, there are two different approaches to this.

- Iterate on the coarse levels until convergence, i.e. until a certain threshold for the maximum change in depth is reached.
- Iterate a fixed number of just a few iterations on coarse levels, and only iterate until convergence on finer levels.

Both approaches have merit. Therefore, we implemented both of them. In the experimental evaluation of this chapter, a discussion on the difference between these two approaches will take place and their effect on the computation time will be investigated.

Boundary Conditions

The upwind discretisation employed in (3.34)-(3.35) has the benefit to give automatically reasonable results for the state constraints boundary conditions that should be used [67]. The state constraint boundary conditions can be realised via Dirichlet boundary conditions with a function $\varphi \equiv +\infty$ (in practice, just a very large number is used) at the image boundary.

Note that for this specific upwind discretisation, von Neumann boundary conditions are the same as state constraints boundary conditions, since for state constraint boundary conditions the derivative outside of the image domain the expression in the minimum always evaluates to a positive

value, and is therefore neglected. Likewise, for Neumann boundary pixels, the boundary pixel value is just repeated outside of the image domain, which results in the respective difference in the minimum term being automatically equal to zero. Therefore, these two types of boundary conditions are identical within this discretisation scheme.

3.13 Stability

A meaningful stability notion is a *discrete maximum-principle*: monotony means that given an initial bound on the data, this bound is conveyed to all later times by a monotone method. This is a stability notion since viscosity solutions also obey this kind of comparison principle.

An explicit scheme like (3.32) is typically *conditionally stable*: The use of too large time step sizes leads to instability of numerical solutions. In order to ensure the validity of the discrete maximum-principle, a local version of it has to be used, namely that the update is not allowed to become smaller or larger than the neighbouring data within the stencil. If this condition is met pixel-wise, the validity of the global discrete maximum-principle follows.

In order to realise the described concept, it turns out that it is useful to consider in a first step the update formula (3.32) without the diffuse source term. Since the term $f_{\mathrm{ref}}^{-3} JW$ and the specular term arise additively in

$$\hat{B} \;=\; f_{\mathrm{ref}}^{-3} JW - f_{\mathrm{ref}}^{-3} \frac{W k_s I_s}{Q} \exp\left(-2v\right) \left(\frac{2Q^2}{W^2} - 1 \right)^{\alpha} , \tag{3.36}$$

one may add up the resulting constraints. The diffuse part is discussed separately at the end of this paragraph.

It is useful for the analysis to introduce a simplified notion for the spatial mesh width: let

$$h \;:=\; \min\left(h_1, h_2\right) . \tag{3.37}$$

Now, different contributions to the stability condition are assembled.

Lemma 3.7. *The term $f_{\mathrm{ref}}^{-3} JW$ in $\hat{B}$ implies a restriction*

$$\tau_{i,j} \;\leq\; \frac{\hat{Q}_\gamma h}{\left(I_{i,j} - k_a I_a\right) \hat{f}^2 \sqrt{2\hat{f}^2 + \gamma \left(\hat{x}_1 + \hat{x}_2\right)^2}} \tag{3.38}$$

up to first order in f_{ref}^{-1} on the time step size, where $x = (x_1, x_2)$ are the pixel coordinates.

Remark 3.8. *At the end of this paragraph, it is shown that terms of first order in f_{ref}^{-1} do not give a meaningful contribution to the stability condition.*

Proof of Lemma 3.7. A local discrete maximum principle is valid provided τ satisfies the condition

$$\left|\tau f_{\mathrm{ref}}^{-3} JW\right| \leq \delta v\,, \tag{3.39}$$

where

$$\delta v = \max\left(\left|v_{i+1,j}-v_{i,j}\right|,\left|v_{i-1,j}-v_{i,j}\right|,\left|v_{i,j+1}-v_{i,j}\right|,\left|v_{i,j-1}-v_{i,j}\right|\right). \tag{3.40}$$

By (3.39)–(3.40) it is ensured that the update introduced via $\tau_{i,j} f_{\mathrm{ref}}^{-3} JW$ stays within the convex hull of the data $\{v_{i+1,j}, v_{i,j}, v_{i-1,j}, v_{i,j+1}, v_{i,j-1}\}$ that contribute in the computational stencil. The values in δv are taken from time level n or $n+1$, depending on the sweeping direction, respectively.

Employing the abbreviation $\nabla\hat{v}$ for the discretisation of ∇v within W, one computes

$$|\nabla\hat{v}|^2 \;\leq\; \left(\sqrt{2\delta v^2/h^2}\right)^2 \;=\; 2\delta v^2/h^2\,, \tag{3.41}$$

$$(\nabla\hat{v}\cdot x)^2 \;\leq\; \left((x_1+x_2)\,\delta v/h\right)^2 \;=\; (x_1+x_2)^2\,\delta v^2/h^2\,, \tag{3.42}$$

where $x=(x_1,x_2)$ denotes the pixel coordinates. At the pixel with index (i,j) this is equal to (ih_1, jh_2), i.e. a scaling by use of x_{ref} is not taken into account in the above formulation.

Denoting by $\hat{W}$ the discretised version of W and using $\hat{Q}_{i,j}^2 \leq 1$, one obtains

$$\hat{W} \;\leq\; f_{\mathrm{ref}}\cdot\sqrt{\frac{2\hat{f}^2+\gamma\left(\hat{x}_1+\hat{x}_2\right)^2}{h^2}\delta v^2+\frac{1}{f_{\mathrm{ref}}^2}}\,. \tag{3.43}$$

This infers

$$f_{\mathrm{ref}}^{-3} JW\big|_{(i,j)} \;\leq\; \frac{\left(I_{i,j}-k_a I_a\right)\hat{f}^2}{\hat{Q}_\gamma}\cdot\sqrt{\frac{2\hat{f}^2+\gamma\left(\hat{x}_1+\hat{x}_2\right)^2}{h^2}\delta v^2+\frac{1}{f_{\mathrm{ref}}^2}} \tag{3.44}$$

where $\hat{Q}_\gamma$ depends on (i,j). By identifying the root term in (3.44) with the Euclidean norm of an expression of the format $\sqrt{a^2+b^2}$, employ the

triangle inequality to obtain:

$$\begin{aligned}
&\sqrt{\frac{2\hat{f}^2+\gamma\left(\hat{x}_1+\hat{x}_2\right)^2}{h^2}\delta v^2+\frac{1}{f_{\mathrm{ref}}^2}} \\
&\leq \quad \frac{1}{f_{\mathrm{ref}}}+\delta v\cdot\sqrt{\frac{2\hat{f}^2+\gamma\left(\hat{x}_1+\hat{x}_2\right)^2}{h^2}} \\
&= \quad \frac{\delta v}{h}\cdot\left(\frac{h}{\delta v\cdot f_{\mathrm{ref}}}+\sqrt{2\hat{f}^2+\gamma\left(\hat{x}_1+\hat{x}_2\right)^2}\right). \qquad (3.45)
\end{aligned}$$

This implies

$$\begin{aligned}
&\left|\tau f_{\mathrm{ref}}^{-3}JW\right|\leq\delta v \qquad (3.46)\\
&\Leftrightarrow \quad \tau\cdot\frac{\delta v}{h}\frac{\left(I_{i,j}-k_aI_a\right)\hat{f}^2}{\hat{Q}_\gamma}\left(\frac{h}{\delta v\cdot f_{\mathrm{ref}}}+\sqrt{2\hat{f}^2+\gamma\left(\hat{x}_1+\hat{x}_2\right)^2}\right)\leq\delta v\,,
\end{aligned}$$

or, equivalently,

$$\tau\leq\frac{\hat{Q}_\gamma h}{\left(I_{i,j}-k_aI_a\right)\hat{f}^2\sqrt{2\hat{f}^2+\gamma\left(\hat{x}_1+\hat{x}_2\right)^2}}-\underbrace{f_{\mathrm{ref}}^{-1}\tau h\cdot\frac{\left(I_{i,j}-k_aI_a\right)\hat{f}^2}{\delta v\cdot\hat{Q}_\gamma}}_{\in\mathcal{O}(f_{\mathrm{ref}}^{-1})}. \qquad (3.47)$$

Neglecting as announced the last term since it is in $\mathcal{O}(f_{\mathrm{ref}}^{-1})$ gives the desired result. □

Now approach the factor $\exp\left(-2v_{i,j}^n\right)$ occurring in the specular term and the diffuse source term.

Lemma 3.9. *Let u_{min} denote the minimal distance an object is away from the camera. Assume the validity of the following model assumption:*

$$u\geq u_{min}\ >\ 0\,. \qquad (3.48)$$

Then one finds

$$\exp\left(-2v_{i,j}^n\right)\leq\exp\left(-2\ln\left(u_{min}\right)\right)=\frac{1}{u_{min}^2}\,. \qquad (3.49)$$

Furthermore, since the images are preprocessed to eliminate pixels with grey value 0, there exists an upper bound u_{max} on u.

Remark 3.10. *The assumption (3.48) is not a restriction. This avoids division by zero. Moreover, the situation that an object is directly before the camera is not of practical interest.*

Remark 3.11. *The assumption of having no 'black pixels' is usually made for SfS algorithms without a detailed explanation as given here, see e.g. [67, 74].*

Proof of Lemma 3.9. As I is normalised, by substituting f^2u^2 for r^2 in (2.2) one obtains

$$I_{i,j} - k_a I_a \leq \frac{1}{\mathsf{f}^2 u^2}\,. \tag{3.50}$$

After some trivial manipulations, the latter inequality leads to

$$u \leq \frac{1}{\mathsf{f}\sqrt{\beta}}\,, \quad \text{where} \quad \beta := \min_{i,j}\left(I_{i,j} - k_a I_a\right)\,, \tag{3.51}$$

and where $\beta > 0$ since 'black pixels' are eliminated in a preprocessing step. Thus, by (3.51) one can impose an upper bound $u_{\max}$ on the reconstruction depth.

Furthermore, by (3.50) fix $u_{\min}$ via

$$\mathsf{f}^2 \max_{i,j}\left(I_{i,j} - k_a I_a\right) = \frac{1}{u_{\min}^2}\,. \tag{3.52}$$

To summarise, (3.49) follows by

$$u \in [u_{\min}, u_{\max}] \;\Rightarrow\; \ln(u) \in \left[\ln\left(u_{\min}\right), \ln\left(u_{\max}\right)\right]\,. \tag{3.53}$$

□

Having the result of Lemma 3.9, it is now possible to approach the specular term.

Lemma 3.12. *The specular term implies the stability restriction*

$$\tau_{i,j} \;\leq\; \frac{\hat{Q}_\gamma h}{k_s I_s \sqrt{2\hat{f}^2 + \gamma\left(\hat{x}_1 + \hat{x}_2\right)^2}}\,. \tag{3.54}$$

Proof. By Lemma 3.4, the factor

$$\left(\frac{2Q_{i,j}^2}{\hat{W}^2} - 1\right)^{\alpha} \tag{3.55}$$

within the discretised specular term

$$f_{\mathrm{ref}}^{-2}\frac{\hat{W}_\gamma k_s I_s}{\hat{Q}_\gamma}\exp\left(-2v_{i,j}^n\right)\left(\frac{2Q_{i,j}^2}{\hat{W}^2}-1\right)^\alpha \tag{3.56}$$

can be estimated by 1. Thus, by Lemma 3.4 and Lemma 3.12, find analogously as in Lemma 3.7 the estimate

$$\tau_{i,j} \;\leq\; \frac{f_{\mathrm{ref}}^2 u_{\min}^2 \hat{Q}_\gamma h}{k_s I_s\sqrt{2\hat{f}^2+\gamma\left(\hat{x}_1+\hat{x}_2\right)^2}}\,. \tag{3.57}$$

By (3.52), since $\max_{i,j}\left(I_{i,j}-k_aI_a\right)\in(0,1]$, the number $f_{\mathrm{ref}}^2u_{\min}^2$ is in general larger than 1. This implies the estimate (3.54). □

Now turn to the diffuse source term

$$f_{\mathrm{ref}}^{-3}k_dI_de^{-2v_{i,j}^n}\,. \tag{3.58}$$

As by Lemma 3.9, one could estimate $e^{-2v_{i,j}^n}$ by $u_{\min}^{-2}$. Then (3.52) shows that $f_{\mathrm{ref}}^{-2}u_{\min}^{-2}$ is identical to $\max_{i,j}\left(I_{i,j}-k_aI_a\right)$. A factor f_{ref}^{-1} remains. This factor leaves the time step size restriction due to the diffuse source term of the same order $\mathcal{O}(f_{\mathrm{ref}}^{-1})$ as the term

$$f_{\mathrm{ref}}^{-1}\tau h\cdot\frac{\left(I_{i,j}-k_aI_a\right)\hat{f}^2}{\delta v\cdot\hat{Q}_\gamma} \tag{3.59}$$

neglected after (3.47). Up to the correct signs before the terms, this can be summarised via

Proposition 3.13. *The stability investigation incorporates the following contributions to the time step restriction:*

$$\tau f_{\mathrm{ref}}^{-1}\cdot\left(\frac{\left(I_{i,j}-k_aI_a\right)\hat{f}^2}{\delta v\cdot\hat{Q}_\gamma}h+k_dI_d\right)\leq\delta v\,. \tag{3.60}$$

The restrictions from (3.60) are in practice not of importance, since the remaining terms impose a much harder constraint. Moreover, heuristically the terms multiplied with τ may only become large enough to be of some significance if δv is close to zero, in which case any minor oscillation violating stability is not visible in any of the tests.

This can be summarised, incorporating the results of Lemma 3.7 and Lemma 3.12, by

Proposition 3.14. *For the SfS application, the time step size $\tau_{i,j}$ can be estimated as*

$$CFL \cdot \frac{\hat{Q}_\gamma h}{\sqrt{2\hat{f}^2 + \gamma\left(\hat{x}_1 + \hat{x}_2\right)^2}} \cdot \left(\frac{1}{k_s I_s} + \frac{1}{\left(I_{i,j} - k_a I_a\right)\hat{f}^2}\right) \tag{3.61}$$

where one usually sets $CFL := 0.9 < 1$.

3.14 Effectiveness of the Gauß-Seidel-like Method and Fast Sweeping

In this first evaluation of the numerical method, the goal is to test two of the ingredients of the numerical solver for their influence on the convergence speed. To give one result directly in advance, they do not have an actual effect on the numerical result. However, both the Gauß-Seidel-like scheme and the fast sweeping certainly will have an influence on the number of iterations needed and the time in which the algorithm converges.

This will be the last section in this chapter where results without the Gauß-Seidel-like scheme or fast sweeping will be considered. Note that for results with fast sweeping, one iteration will be a full sweeping cycle, i.e. one iteration with fast sweeping is comparable to four iterations without fast sweeping.

3.15 Comparison of the Different Schemes

Another interesting point of the evaluation of the numerical scheme is comparison against other numerical schemes. Existing methods that are based on similar models are the optimal control schemes of Prados et al. and Cristiani et al. However, since these schemes are only for Lambertian surfaces, the method presented in this chapter is restricted to the Lambertian case as well. For the optimal control approaches, a similar Gauß-Seidel-like scheme has been proposed by the authors [22, 74]. However, they did not use fast sweeping. Since fast sweeping improves the performance of these methods as well, a comparison of all methods using fast sweeping will give a better comparison of the underlying numerical scheme.

Table 3.2 shows the run times and number of iteration for the different schemes on different images. To give a general impression, the table contains all the run times for the focal length experiment in Chapter 2. The tests have been performed without any parallelisation on an Intel Core2 Duo

Table 3.2: Comparison of the run times for the different methods for Lambertian methods. All times are given in seconds.

		Number of Iterations			Run Time		
Surface	f	Direct	Prados	Cristiani	Direct	Prados	Cristiani
Vase	125	61	21	19	2.04	9.83	6.68
Vase	250	83	26	26	2.75	12.12	8.88
Vase	500	64	31	24	2.12	13.58	7.96
Vase	1000	44	19	19	1.26	4.97	5.20
Mozart	125	140	26	29	19.27	49.30	42.57
Mozart	250	78	30	27	10.45	56.79	37.30
Mozart	500	89	33	29	11.89	60.40	39.21
Mozart	1000	102	79	55	11.59	115.92	61.87

E4600 at 2.4 GHz with 2 MB Cache running Linux. The visual results can be found in Section 2.7. As stopping criterion for all experiments a change of less than 10^{-4} in the logarithmic depth v in one iteration for all pixels has been employed. For Cristiani's method, the unit circle has been sampled in eight directions and three rings, which improves the performance compared to more directions. Frequently, even more samples are used for this method, eight directions is the minimum that actually gives convincing results.

This experiment demonstrates that the direct method is significantly faster than the optimal control approaches. Despite taking a lot more iterations, iterations are so much faster that in the end, this method still has a large advantage in terms of computation time. Generally, the direct approach benefits a little from larger focal lengths, while optimal control approaches perform similarly for different focal lengths.

In a second experiment, the run times for different specularities shall be discussed. Again, the images at $f = 500$ are used, the visual results and errors can be found in Section 2.8. Table 3.3 shows iteration numbers and run times for different specular highlights for the vase experiment, Table 3.4 the results for the Mozart face experiment.

This gives interesting results. For very small amounts of specular highlights, the direct method outperforms the optimal control approaches. However, for images where specular highlights dominate, the direct method needs significantly more iterations to converge and becomes slower. This is not so surprising, because after all, the optimal control approaches still solve the easier problem of a Lambertian surface, despite the image contains specular highlights. It is not surprising that the run times remain about constant for these Lambertian models. On the other hand, the Phong PDE

Table 3.3: Comparison of the run times for the different methods on specular images of the vase. All times are given in seconds.

		Number of Iterations			Run Time		
k_s	α	Direct	Prados	Cristiani	Direct	Prados	Cristiani
0.1	5	62	27	22	2.44	12.19	7.58
0.2	5	59	22	18	2.35	10.17	6.29
0.3	5	54	20	18	2.10	9.18	6.21
0.4	5	81	19	15	3.17	8.74	5.22
0.5	5	140	20	15	5.54	9.26	5.36
0.6	5	209	21	15	8.30	9.71	5.20
0.7	5	279	22	17	10.95	10.26	6.19
0.8	5	366	22	18	12.46	10.34	6.31
0.1	10	62	26	22	2.48	11.84	7.62
0.2	10	60	21	21	2.36	9.60	7.33
0.3	10	59	20	17	2.34	9.30	5.88
0.4	10	73	21	18	2.90	9.73	6.23
0.5	10	157	22	19	6.27	10.26	6.58
0.6	10	393	23	20	15.56	10.60	6.95
0.7	10	414	23	20	15.93	10.84	7.02
0.8	10	1290	23	19	42.57	10.77	6.60
0.1	20	66	27	22	2.59	12.32	7.64
0.2	20	57	20	21	2.27	9.12	7.28
0.3	20	55	20	19	2.20	9.29	6.59
0.4	20	86	19	19	3.46	8.88	6.59
0.5	20	308	22	19	12.34	10.19	6.56
0.6	20	524	22	21	20.44	10.20	7.30
0.7	20	1791	23	23	67.87	10.82	8.01
0.8	20	3823	22	24	124.97	10.27	8.38

Table 3.4: Comparison of the run times for the different methods on specular images of the Mozart face. All times are given in seconds.

		Number of Iterations			Run Time		
k_s	α	Direct	Prados	Cristiani	Direct	Prados	Cristiani
0.1	5	88	32	30	13.77	60.03	41.25
0.2	5	85	31	30	13.50	57.97	41.35
0.3	5	83	30	33	13.08	56.43	45.55
0.4	5	88	29	33	13.79	54.64	45.51
0.5	5	141	28	33	22.10	52.68	45.55
0.6	5	250	27	34	38.91	50.71	46.96
0.7	5	286	27	34	42.88	51.32	47.05
0.8	5	385	26	32	46.53	49.23	44.18
0.1	10	87	34	30	13.52	63.22	40.86
0.2	10	91	31	30	14.34	57.13	40.71
0.3	10	89	29	30	14.02	53.25	40.89
0.4	10	89	28	31	14.02	51.95	42.22
0.5	10	159	27	32	25.04	50.32	42.97
0.6	10	725	26	33	112.72	47.95	44.58
0.7	10	769	25	30	110.06	46.19	41.64
0.8	10	1372	24	31	152.77	44.85	42.02
0.1	20	97	33	30	15.26	60.11	40.05
0.2	20	91	31	31	14.40	57.69	41.86
0.3	20	85	29	31	13.55	53.46	42.29
0.4	20	91	27	32	14.42	49.88	43.52
0.5	20	426	26	31	66.38	47.85	41.91
0.6	20	617	25	32	93.82	46.65	43.32
0.7	20	1970	24	29	279.20	44.60	39.25
0.8	20	3689	21	24	326.5	39.19	32.47

is significantly more complex, so the higher run time for the Phong reconstruction is not surprising. For increasing α, the reconstruction becomes even more difficult.

3.16 Effectiveness of the Cascading Multigrid Method

In this paragraph, the effect of the cascading multigrid method shall be evaluated. As mentioned before, there is a key decision to be made when

actually applying the cascading multigrid method: how far to iterate the solution on coarse grids. In this section both variants will be addressed. Since the results are qualitatively the same, the focus will lie on run times. As usual, the evaluation will be done on the vase and the Mozart face, whose reconstructions and depth errors have been discussed in Chapter 2.

Table 3.5 shows the run times and number of iterations on the finest level for the Lambertian vase experiment for all the three numerical methods for both variants of the cascading multigrid. For the results of all methods, both the Gauß-Seidel-like scheme and fast sweeping have been employed.

In all the experiments except one, the direct method achieves a decent speed-up when iterating until convergence on coarse levels. For only a few iterations on coarse levels, the direct method even becomes slower in many cases. The main reason for this can be seen in the convergence analysis from Section 3.17: While iterations are generally much faster than for the optimal control approaches, more iterations are needed to get to a good estimate. Five iterations on the coarse levels are simply not enough for the method to obtain a decent estimate. On the finest level, one ends up with an estimate that is worse than the initial estimate in the method on a single grid. For the optimal control approaches, this is different. On the coarse grids, these methods are only a few iterations away from convergence, which results in no big difference between iterating until convergence and the fixed number of iterations. In some cases, the fixed number of iterations even performs slightly better. The direct method gains the best improvement in terms of final run time on large and on images with a small focal length, while the optimal control approaches achieve similar speed-ups through all experiments, with Cristiani's method being a bit better on the small image.

3.17 Convergence Analysis

In the previous section, it became clear that the direct method takes significantly more iterations to converge than optimal control approaches. Nevertheless its run times were significantly smaller than for the optimal control approaches. This raises an interesting question: How exactly do the two methods reach that result? Do they converge in the same way? One way to evaluate this is to observe one type of error during iteration of the method: The maximum change in logarithmic depth. For the Lambertian experiment with $f = 500$, graphs of this error for both experiments can be found in Figure 3.1. The errors are plotted on a logarithmic scale.

Table 3.5: Comparison of the cascading multigrid performance for all methods (Direct, Prados, and Cristiani). If the iteration number on coarser levels is marked as ∞, then until convergence is meant, "-" means no cascading multigrid has been used. The convergence threshold is always 10^{-4}. All times are given in seconds.

	Iters		Iterations (finest)			Run Time		
Surface	(coarse)	f	D	P	C	D	P	C
Vase	-	125	61	21	19	2.04	9.83	6.68
Vase	∞	125	28	7	8	1.22	3.18	3.17
Vase	5	125	113	7	8	3.82	3.05	3.16
Vase	-	250	83	26	26	2.75	12.12	8.88
Vase	∞	250	32	13	8	1.29	6.88	3.34
Vase	5	250	185	13	8	6.22	6.53	3.24
Vase	-	500	64	31	24	2.12	13.58	7.96
Vase	∞	500	28	14	7	1.19	7.19	3.04
Vase	5	500	176	14	7	5.91	6.71	2.80
Vase	-	1000	44	19	19	1.26	4.97	5.20
Vase	∞	1000	57	4	5	1.74	1.86	1.82
Vase	5	1000	68	4	5	2.03	1.76	1.74
Mozart	-	125	140	26	29	19.27	49.30	42.57
Mozart	∞	125	81	14	17	12.60	19.97	25.65
Mozart	5	125	189	14	17	25.42	19.98	25.07
Mozart	-	250	78	30	27	10.45	56.79	37.30
Mozart	∞	250	35	14	12	6.54	23.68	19.47
Mozart	5	250	285	14	12	38.83	22.62	18.73
Mozart	-	500	89	33	29	11.89	60.40	39.21
Mozart	∞	500	51	18	15	9.14	38.54	25.26
Mozart	5	500	246	19	15	33.13	37.49	22.95
Mozart	-	1000	102	79	55	11.59	115.92	61.87
Mozart	∞	1000	74	25	23	9.50	40.15	29.23
Mozart	5	1000	122	25	23	14.42	37.77	27.83

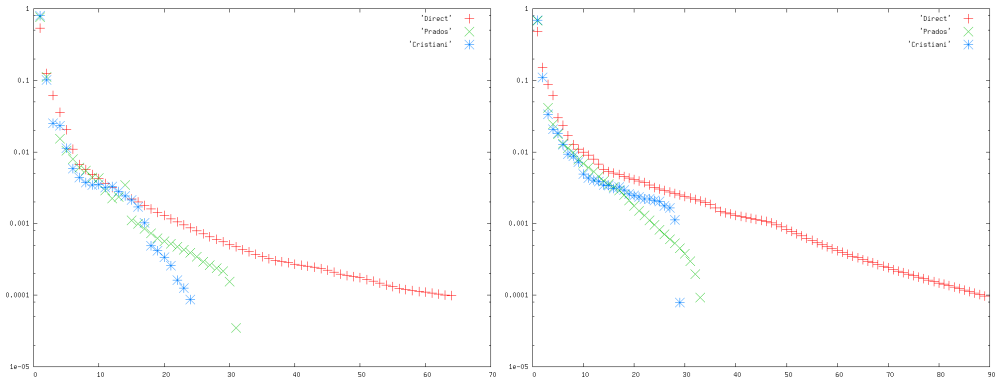

Figure 3.1: Development of the iteration error on vase and Mozart face (both $f = 500$) for the different schemes using a threshold of 10^{-4}.

3.18 A Hybrid Method

The key observation in the previous section is that the optimal control approaches take only very few iterations to get from about 10^{-3} to 10^{-4}, while the direct method takes significantly less time to converge to a maximal change per pixel of 10^{-3}. The natural way to proceed now is to combine both numerical techniques. For that, a very simple approach is followed: up to a certain maximal change per pixel the direct method is used, and then the iteration switches to one of the optimal control approaches.

Since Cristiani's method is generally faster than Prados' method, it should be the better choice for this experiment. For both the vase and the Mozart face, three experiments will be done:

1. Both algorithms are ran separately on the images to a stopping condition of 10^{-6}, which is significantly larger than the usually employed stopping condition. However, this will emphasize the effect of the experiment.

2. The direct method will iterate to a stopping condition of 10^{-3}, afterwards the method is switched to Cristiani's optimal control scheme until the stopping condition of 10^{-6} is reached.

3. In principle the same as in the second experiment, just with 10^{-4} as the intermediate stopping condition.

Figure 3.2 shows the plots of the convergence behaviour of both methods running separately. Figure 3.3 shows the convergence behaviour of the

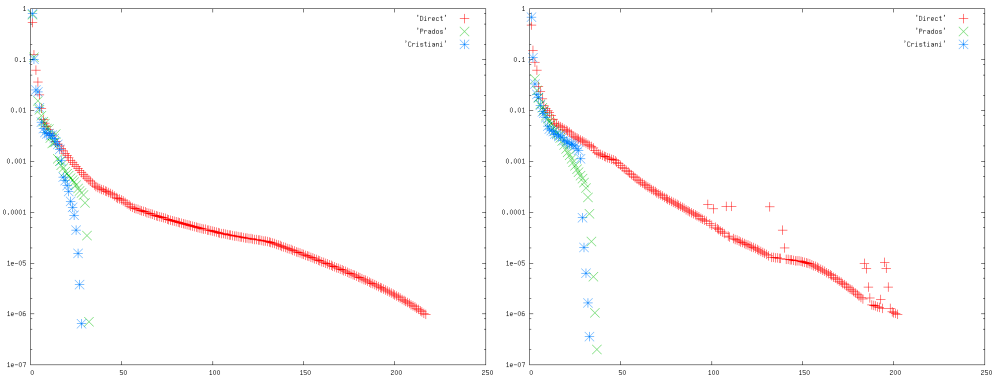

Figure 3.2: Development of the iteration error on vase and Mozart face (both $f = 500$) for the different schemes using a threshold of 10^{-6}.

Table 3.6: Iteration Numbers and run times of the standard (S) methods vs. the hybrid (H) method.

		Direct Method			Cristiani's method			Total
Exp	N/H	Th.	Iters	Time	Th.	Iters	Time	Time
M	S	10^{-6}	202	26.49	-	-	-	26.49
M	S	-	-	-	10^{-6}	33	44.82	44.82
M	H	10^{-3}	48	6.45	10^{-6}	20	26.82	33.27
M	H	10^{-4}	89	11.82	10^{-6}	20	27.40	39.22
V	S	10^{-6}	217	7.10	-	-	-	7.10
V	S	-	-	-	10^{-6}	28	9.48	9.48
V	H	10^{-3}	23	0.77	10^{-6}	19	6.44	7.21
V	H	10^{-4}	64	2.16	10^{-6}	14	4.83	6.99

hybrid method using the 10^{-3} threshold, Figure 3.4 the one of the hybrid method using the 10^{-4} threshold. Table 3.6 shows the run times of the experiments including the run times of each part of the hybrid method.

The non-hybrid results confirm what has also come clear in the previous paragraph. Up to an error of around 10^{-3}, the schemes need about the same amount of iterations. Since the iterations of the direct method are computationally less expensive, it should make sense to do the first iterations using this method. Also, Cristiani's method seems to converge faster in the end than Prados' method. In the actual hybrid method another effect becomes clear. If one iterates the method up to a maximal change of 10^{-3}, then the next maximal change for the subsequent optimal control method

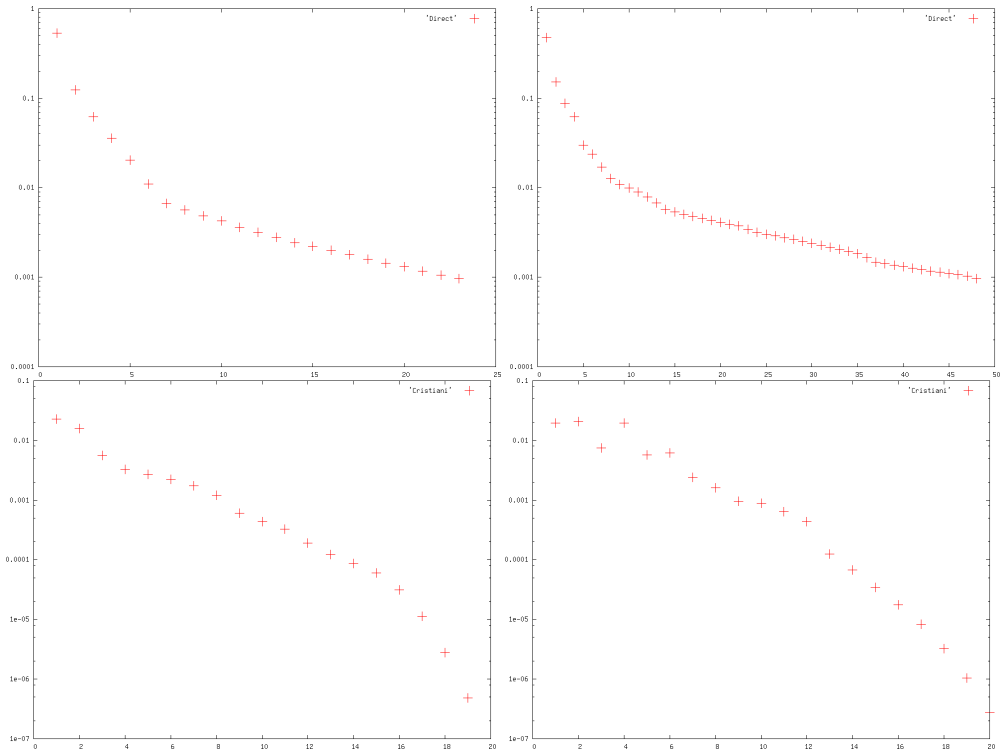

Figure 3.3: Development of the iteration error on vase and Mozart face (both $f = 500$) for the hybrid scheme using an intermediate threshold of 10^{-3}. Top row: Initial iteration using the direct method. Bottom row: Iterating until convergence using Cristiani's method.

will not be 10^{-3}. In fact, it is much closer to 0.05. However, this is an immense improvement compared to the original error at the first iteration for this method. For the Mozart experiment, the result is very clear. It helps to improve the convergence of Cristiani's method to do the first couple of iterations, but it is not possible to outperform the direct method. For the vase, the results are similar. However, for a suitably chosen threshold it is actually possible to even slightly beat the run time of the direct method.

This leads to the conclusion that a combined method cannot improve the performance of the direct method, since the part that is iterated with an optimal control method will still take at least about as long as using the direct method for the whole iteration. However, it is an interesting way to speed up optimal control methods. By doing the first iterations with the direct method, it is possible to obtain a significant improvement in the performance on this method. Taking into consideration that, although

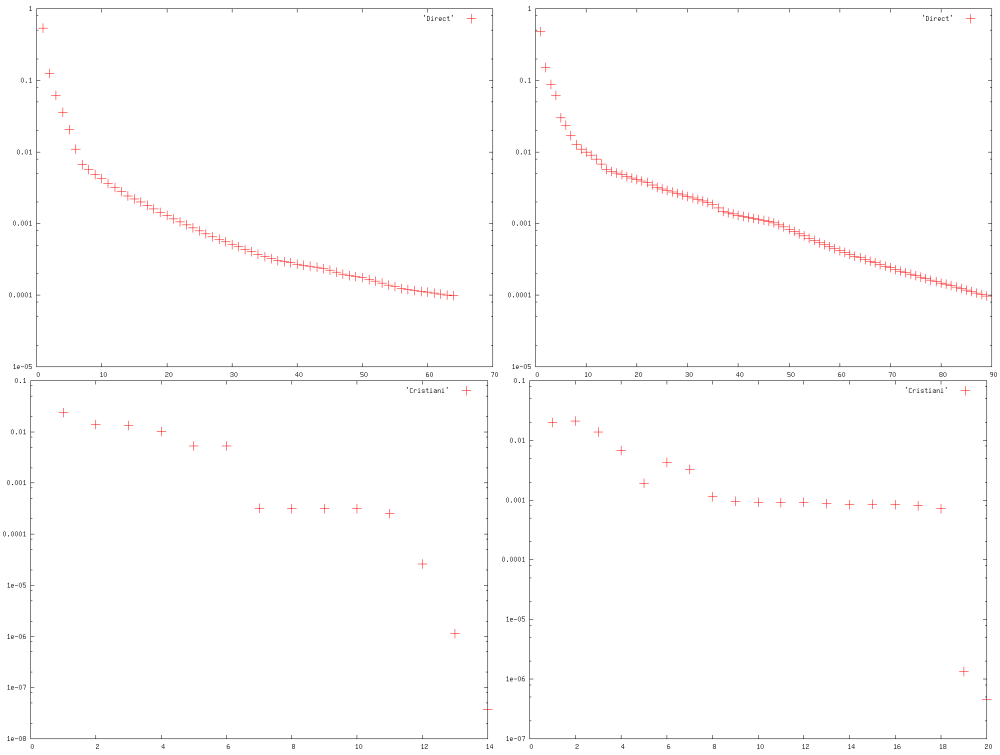

Figure 3.4: Development of the iteration error on vase (V) and Mozart (M) face (both $f = 500$) for the hybrid scheme using an intermediate threshold of 10^{-4}. Top row: Initial iteration using the direct method. Bottom row: Iterating until convergence using Cristiani's method.

qualitatively the methods are equally good, the optimal control approaches exhibit a slightly smaller depth error on some experiments, this might be a good way to improve these methods even a bit further. For practical applications, though, this is hardly relevant, since the direct method will generally be superior.

3.19 Discontinuities

A very interesting point of the method is the handling of discontinuities. Since in a Lambertian setting, the results of the different numerical schemes are basically equivalent, only the direct numerical method presented in Section 3.12 will be used for this evaluation.

Figure 3.5 shows a very instructive test surface and the respective input

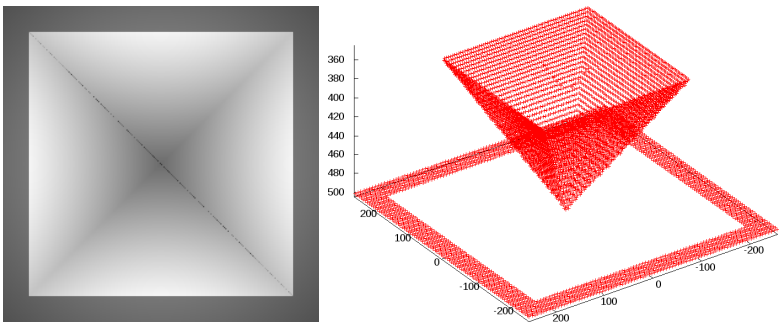

Figure 3.5: Discontinuity test. Pyramid input image and ground truth surface.

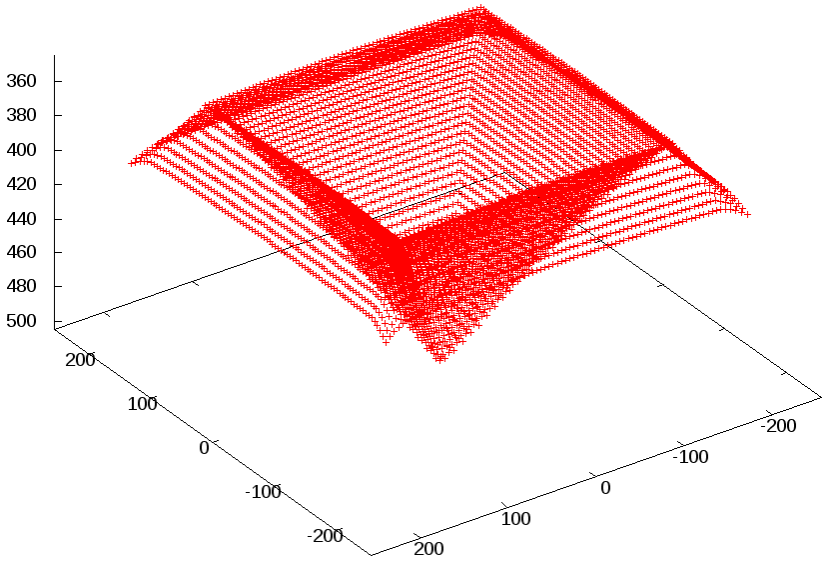

Figure 3.6: Reconstruction of the pyramid with no additional constraints.

image: an upside-down pyramid. The complexity of the pyramid is very limited. Except the fact that it contains several edges, the shape is a very simple one. However, it features a very large distance from the edge of the pyramid to the background plane. What happens at this discontinuity? How does it affect the reconstruction quality. Why does it happen?

Figure 3.6 shows the result of the reconstruction. The discontinuity is not preserved at all. However, the shape of the pyramid appears correct, only the background seems to be affected. Figure 3.7 shows the same reconstruction. However, this time, an additional Dirichlet constraint has been

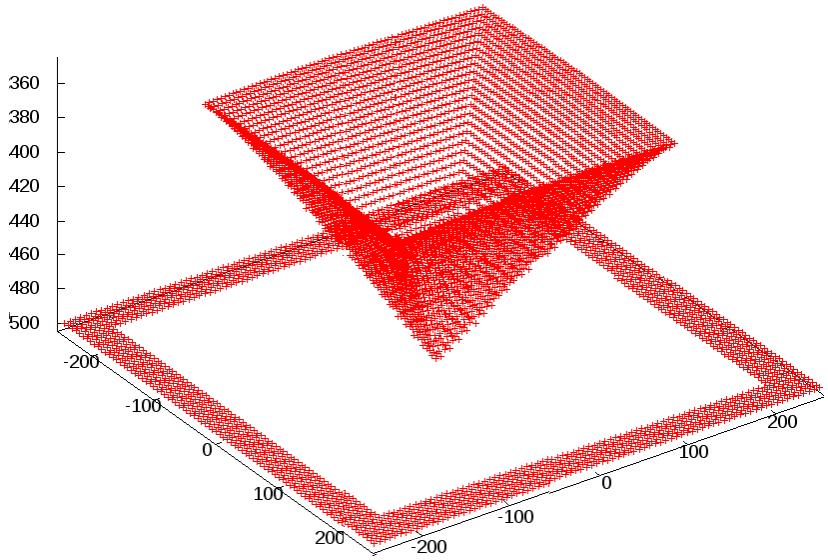

Figure 3.7: Reconstruction of the pyramid with Dirichlet condition on the innermost pixels of the background.

applied within the image: The innermost pixels of the background, i.e., the pixels directly adjacent to the pyramid, are set to their true depth values and are not changed while iterating. The result is impressive, a practically perfect reconstruction is obtained. In a third step, the opposite is done. Instead of setting the innermost pixels of the background to their true depth, the outermost pixels of the pyramid are set to their true value. Figure 3.8 shows the corresponding result. It is identical to the result of the first reconstruction.

Why is this the case? In fact, it is caused by the upwind discretisation, which takes depth derivatives in a pixel only if the neighbouring pixel is smaller than the central pixel of the stencil, cf. equation (3.11). In fact, for the pyramid surface, the closest pixels are in fact the outermost pixels of the pyramid. Although the iterative method comes from large depth values and approaches monotonically decreasing, the actual shape information is propagating from small to large depth values. Therefore, when supplying the true, large values at the innermost pixels of the boundary, none of this information propagates to the pyramid, since the depth values are smaller in this direction. Even more importantly, none of the depth information of the edge of the pyramid can propagate to the background, since the values there are fixed. Without that constraint, the algorithm would try to construct the background in a way that it fits continuously to the edge of the pyramid.

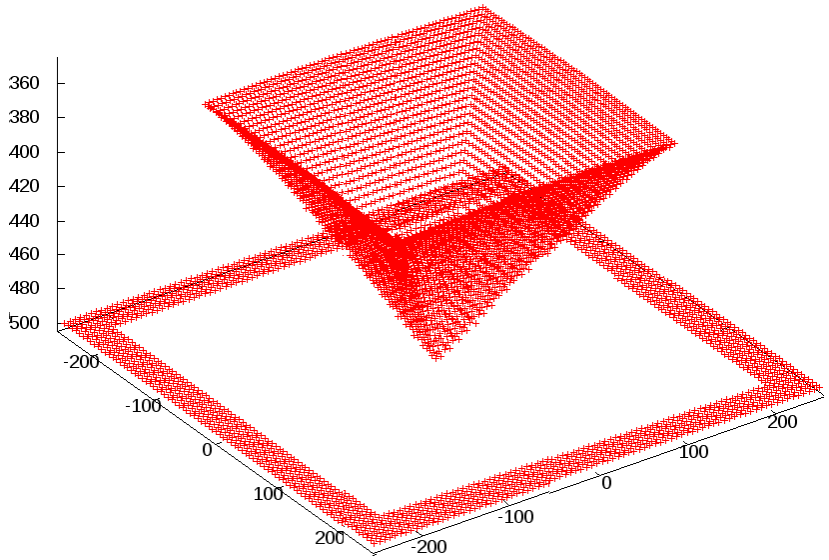

Figure 3.8: Reconstruction of the pyramid with Dirichlet condition on the outermost pixels of the pyramid.

Since information only propagates from near depth values to far depth values, the reconstruction at near depth values is never distorted. This also explains why the opposite experiment did not make any difference compared to the first one. Information does not propagate from the background to the pyramid anyway, so enforcing the grey value at the edge of the pyramid to block faulty information from the background does not change anything.

In fact, information for hyperbolic PDEs like the one considered here always propagates along characteristic lines. In this application with the upwind discretisation, characteristic lines always point against gradient direction. For the upwind scheme, the depth gradient is the direction where the surface depth gets smaller in the fastest way. Therefore, information propagates from small pixels to large pixels. This will play a crucial role in the next chapter, where a different numerical method for this problem is introduced that exploits this property of the PDE.

At discontinuities, the method is certainly not well-posed since solutions may differ depending on the boundary conditions employed at the boundaries of the continuity segments, as has been demonstrated in this section.

Figure 3.9: Average depth and brightness errors of the pyramid experiment. Brightness error given in grey values $(0 - 255)$.

Condition	Depth Error	Brightness Error
None	8.69%	2.48
Background	1.79%	2.74
Pyramid	8.69%	2.48

3.20 Depth Error vs. Brightness Error

In the previous paragraph, the SfS method reconstructed a continuous surface from an image of a discontinuous surface. Obviously, this results in rather large depth errors. However, the method does not optimise a depth error. In fact, the PDE has been derived from the brightness equation. Now, the question arises, in terms of brightness, is the solution obtained in the previous paragraph correct? Within Chapter 2, it would not have been possible to answer that question. However, with the discretisation of spatial derivatives, it is possible to numerically compute the brightness of the surface from the depth and the gradient of the surface. The gradient is obviously computed with the standard upwind scheme which has also been used in the method itself. This approach corresponds to computing the brightness error introduced in Section 2.6.

Table 3.20 shows the depth and brightness errors of this experiment. While the depth errors differ drastically, the brightness errors are always low. Note that the brightness errors here are computed from the image brightness computed in double precision, while the input image is quantised into 256 grey values. Obviously, the ground truth and the continuous solution are both correct surfaces corresponding to the grey value image. In this sense, the numerical SfS method is again ambiguous. However, remember that the method is only claimed to be well-posed for continuous surfaces. This example clearly is not continuous.

Now, the question remains: Is the depth error a valid error measure, or should one rather consider a different error measure? Or is even a completely other error measure a valid one? Horn [35] discussed that question in an orthographic framework. However, his conclusion that the best error would be a depth error is questionable, in particular in the setting where he discussed it. I have been using the brightness error as an error measure for an orthographic SfS before [98]. In a perspective setting, though, saying that the brightness error would be the right error measure is a bit too much. After all, the goal is to recover a shape, not an image. Most authors

of recent SfS seem to share that opinion, since everybody is using the depth error as error measure of choice. However, when looking at depth errors, one should always be aware of discontinuities in the original surface and that they can create ambiguities.

3.21 Final Remarks

In this chapter, a direct method for the Phong SfS model has been introduced. A detailed discussion of discretisation and other numerical issues has been done. In addition to the discretisation and the iterative method, advanced acceleration techniques like fast sweeping and a cascading multigrid method have been proposed. The choice of the discretisation has been motivated by a numerical scale analysis, an approach that is new to image processing applications.

The direct method turned out to have a superior performance to optimal control approaches known from the literature. In the next chapter, another numerical method for the Phong SfS model will be introduced. In addition to an even better performance, this method has additional merit in the quality of the reconstructions.

Chapter 4

Fast Marching Methods for Shape from Shading

4.1 Contributions

In this chapter, a fast marching (FM) method for the novel non-Lambertian SfS model presented in Chapter 2 is introduced. For previous FM methods for SfS, one drawback has always been the necessity of correct depth information being provided at singular points. In contrast to these methods, a very effective technique of getting very precise estimates at singular points is introduced in this chapter. This leads to convincing and stable results. Finally, the applicability of the method to large real-world images is demonstrated, which shows the computational efficiency of this method. The FM method discussed here has been published in [95].

4.2 Fast Marching Methods and Shape from Shading

While SfS methods have been developed since the late 1960s, the fast marching (FM) concept is a much newer one. The concept of FM methods has been brought up by Tsitsiklis in 1995 [92] as a fast solver for Hamilton-Jacobi equations. The same concept has also been developed by Helmsen et al. [30] and Sethian [79–81] in 1996. In contrast to Tsitsiklis' method, the latter two methods are targeted on algorithms using an upwind discretisation of Eikonal equations. While the three methods have been developed

independently of each other under different names, it was Sethian who gave this algorithm the name "Fast Marching method".

Although orthographic SfS can be easily formulated as an Eikonal equation, and while an upwind discretisation has already been applied to SfS in 1992 by Rouy and Tourin [77], it took quite a long time until FM methods reached the SfS world. In the 1990s, Kimmel and Bruckstein [42–44, 46, 82] were the first that proposed algorithms for orthographic SfS based on level sets. Since FM is a technique to solve level-set problems in a fast way, these methods form the basis for the first SfS method using a FM. This was introduced in 2001 by Kimmel and Sethian [45], who applied Sethian's FM solver to the previously developed level-set approach for orthographic SfS.

After three more years, in 2004, Tankus et al. [87] were the first to apply the FM concept to perspective SfS. Their method is already based on a rather advanced SfS model, which does not include light attenuation, but performed well enough to give quite satisfying results on a limited set of real-world images. Tankus et al. applied their method to small patches of endoscopic images. To this end, they linearised the PDE on which their SfS is based on. They also proposed an iterated version of their FM method, which allowed for slightly improved results [89]. In their comparison [90], they also compared to Prados' method. Qualitatively, their results were inferior to those of Prados' iterative method, which might have been caused by their linearisation of the PDE.

Finally, Prados et al. suggested a FM method for his Hamilton-Jacobi framework [76], on which also his perspective Lambertian SfS method including light attenuation. He claims his method would work for all Hamiltonians suited to his framework. However, their publication on this lacks an extensive discussion on reconstruction quality and performance, only one synthetic experiment is shown to prove the concept [76].

All the FM methods for SfS so far have one drawback: They depend on certain depth information to be given. For the SfS method to work, the correct depth must be given at all singular points, which coincide with the local minima in depth. For Tankus' method, which is based on a SfS model without light attenuation, it is not possible to avoid this, since such models are ambiguous with respect to multiplicative factors on the depth.

In this chapter, we will introduce a FM method based on the SfS model introduced in Chapter 2. In addition to a working FM method for this model, also the issue of estimating the correct depth at singular points is discussed. In the Lambertian SfS case, this estimate can also be used as initial data for the FM method Prados suggested.

4.3 Convexity of the Hamiltonian

Although Rouy and Tourin [77] only discuss the applicability of an upwind discretisation, it is intuitively clear that in this case, an upwind scheme is also applicable in the non-convex regions, since information propagates from local depth minima, and specular highlights are concentrated to local depth minima. Loosely spoken, as long as the solution propagates from small to large depth values, the upwind discretisation looks in the right direction. The validity of this argument will be verified experimentally at a later point in this chapter.

4.4 A FM Method for Perspective Phong Shape from Shading

There are many extensive descriptions of the FM method available in the literature, see especially [81].

The basic principle behind the FM scheme applied in the SfS setting is to advance monotonically a front from the foreground of the depicted object to the background. Thereby, the pixels are distinguished by the labels 'known', 'trial' and 'far', respectively, referring thereby via 'known' and 'trial' to the corresponding 3-D depth.

In the beginning, all pixels are labelled as 'far' with their depth values set to infinity. However, since the FM method propagates information from the foreground to the background, it relies on correct depth values being supplied in the pixel which is most in the foreground, i.e. the pixel with minimum depth. In the case of complex images which consist of multiple segments, for each of these segments the correct depth in the point with minimum depth must be supplied. These points are called *singular points*. These singular points are then marked as 'trial', which concludes the initialisation of the method.

For FM methods on SfS it is common to just require this data to be provided. Other methods like [74] and the iterative scheme proposed in Chapter 3, however, do not require the knowledge of given initial depth data. Therefore the goal is to estimate very precisely the locations of singular points and obtain a SfS method using the FM scheme that does not rely on any depth information to be provided. The task of estimating this data will be the subject of the next section.

The 'trial' candidate with the smallest computed depth is then marked as 'known', taking the computed 3-D depth in this point as the estimate. The pixels adjacent in terms of the stencil to the new set of known points are

updated with respect to their label, marking them as 'trial'. The described process is then repeated until all image pixels are marked 'known'.

Fixed-Point Iteration

Updating the depth at 'trial' points consists of solving the discrete form of the model PDE for v in this point. In contrast to other SfS techniques using FM, one needs to solve a nonlinear equation. This is not trivial in this case, since near the solution, the derivative of the model PDE is very low, making standard solvers like the Newton method diverge in most cases. To avoid this, employ the *Regula Falsi*: Starting with two values v_1 and v_2 such that $v_1 < v_2$ and the left-hand side L of the model PDE is negative in v_1 and positive in v_2, one chooses

$$v_3 := \frac{L(v_2)v_2 - L(v_1)v_1}{L(v_2) - L(v_1)}\,, \tag{4.1}$$

which is between v_1 and v_2. If L at v_3 is negative, set $v_1 := v_3$, otherwise set $v_2 := v_3$. Repeating this until v_1 and v_2 are very close together yields an estimate for the solution in this pixel.

Note that computing the derivatives involves computing a minimum. Depending on v_1, v_2 and v_3, these minima might change within the estimation process. Thus, it is necessary to update the values of v_1, v_2, v_3 during the process.

4.5 Initialisation at Singular Points

The FM methods for SFS rely on the knowledge of ground truth data at singular points, i.e. at points with locally minimal depth. However, in general this kind of data is not given. Thus, these depth values need to be estimated. In the experimental section, it will be shown that a good estimate is crucial for the reconstruction quality.

In most other works, this issue is neglected. In [87], the problem is solved by obtaining an initial estimate for the depth using an orthographic SfS method. Their perspective method, however, is not comparable with the one used in this paper, since they neglect the light attenuation term. By doing this, their solution is invariant to multiplicative scalings of the depth. To obtain a working method for the model discussed here, one either needs to know the correct depth at singular points or to estimate both the singular points and their depth.

In this section, ways to estimate the locations of singular points and estimate their depths as correctly as possible will be introduced.

Lambertian Case

For simplicity, we first focus on the Lambertian case, i.e. $k_a = k_s = 0, k_d = 1$. In this model, the brightness of a pixel is determined by two main factors: (i) The angle between surface normal and light source direction ϕ and (ii) the light attenuation because of the distance of the surface point to the light source. Directly from the model one obtains the simple equation

$$I = I_d \frac{\cos\phi}{u^2 f^2} \ . \tag{4.2}$$

Assuming the surface to be continuously differentiable, the points of minimal depth are the points where the derivatives of the depth vanish, which means the surface normal points directly to the viewer. This results in $\phi = 0$, which leads by use of $\cos 0 = 1$ and re-arranging (4.2) to

$$u = \sqrt{I_d \frac{1}{I f^2}} \ . \tag{4.3}$$

Knowing the coordinates of singular points, one can compute the depth. It remains to determine the coordinates of singular points. Singular points are local minima in depth. Since minima in depth mean both less attenuation and a maximum Lambertian reflectance, this suggests that local maxima in image brightness are the singular points. At the image boundary, it might happen that the brightness maxima do not satisfy $\phi = 0$. In this case, there can be errors. However, in most cases, this does not affect the reconstruction quality significantly.

Due to sampling and quantisation artefacts, it is possible that this estimate might be slightly off, both in the location of singular points and in the estimated depth. This effect is usually rather small.

In conclusion, the proposal is to search local maxima in the image and estimate their depth according to equation (4.3). Boundary pixels should not be considered, since the estimate might be incorrect due to ϕ not being zero. The points obtained in this way should be marked as 'trial' points for the subsequent FM method. In the Phong case which follows, the same approach is used.

Phong Case

To obtain a good estimate for singular points in the general case, review the model equation again. Essentially, one has

$$I = k_a I_a + \frac{k_d I_d \cos\phi + k_s I_s \left(\cos\theta\right)^\alpha}{u^2 f^2} \ . \tag{4.4}$$

At singular points, one obtains $\phi = \theta = 0$, which simplifies equation (4.4) to

$$I = k_a I_a + \frac{k_d I_d + k_s I_s}{u^2 f^2} \ . \tag{4.5}$$

Now, after shifting the grey values down by the ambient brightness to $I - k_a I_a$, it is possible to separate diffuse and specular light and compute the diffuse brightness I' by

$$I' = \frac{k_d I_d}{k_d I_d + k_s I_s}(I - k_a I_a) \ . \tag{4.6}$$

Now, one can make use of the equation (4.3) using I' instead of I. An alternative, equivalent approach to this is to solve the Phong brightness equation directly for $\phi = \theta = 0$, which gives the same result.

4.6 Method Error

The proposed procedure may lead to even more accurate results than with the iterative case. As shown before, the solution in the iterative case is somewhat viscous. It was possible to observe that looking at the depth estimate at local minima of the depth. Here, the estimate of the depth at singular points, which coincide with local minima in depth up to discretisation artefacts, is exact. Since singular points are never updated – and even if they would be, it would not change anything since both v_x and v_y would be 0 within the discretisation – this methodical error is avoided here. This can in many cases produce much better results. Later in this chapter, plenty of examples will be given in which the results obtained by the FM method are actually superior to the results obtained by the direct, iterative method from the previous chapter.

4.7 Real-Time Shape from Shading

Unlike the iterative method presented in Chapter 3, which essentially has quadratic complexity, a naïve implementation of a FM method has complexity $O(n \log n)$. Note that there are even variants of FM methods that have only linear complexity [104], however, these methods use quantisation of the output domain to achieve this. For SfS, such a quantisation is far from trivial, since the computation is done in a logarithmic scale, and it is not obvious how to quantise depth. Certainly, it would be possible to use

a rather fine quantisation to technically obtain linear complexity, however, this would lead to a linear complexity with a high constant, which may be inferior to $O(n \log n)$ for practical applications. In this section, it will be demonstrated that even with a straightforward implementation, impressive computation times can be achieved.

4.8 Performance and Accuracy

In this section the goal is to evaluate performance and accuracy of reconstructions using the FM method. The tests are run on the same synthetic images as before.

Lambertian Case

As test images for the Lambertian case, the vase and the Mozart face at different focal lengths are used. Table 4.1 shows the error rates of the reconstructions using the FM method presented in this chapter in comparison to the iterative, direct method used before. Here, something very interesting happens. While the errors at low focal lengths are virtually equivalent, at higher focal lengths the errors of the reconstruction using the FM scheme even go down instead of up. A look at the maximal error column reveals the reason for this: practically no outliers occur. For the vase, even the maximal depth errors are negligible., For the Mozart face smaller outliers occur, but the overall error is still very low, which suggests these outliers are infrequent. At the transition between vase and background, the FM method seems to be a lot more stable.

The main reason for this lives in the way information is propagated. The local maxima in both segments, vase and background, are determined individually at the initialisation. By this, both vase and background are in the right distance to the camera. Now the information propagates and meets at the transition. When the fronts meet, only very few pixels are left that could exhibit outliers. This causes outliers to be very infrequent.

The improved reconstruction quality can also be inspected visually in Figures 4.1 and 4.2. On the first glance, all the reconstructions seem perfect, under consideration of the fact that for $f = 125$, hardly anything of the surface is visible at all. Only at very close inspection it is possible to make out small differences between the reconstructions.

Finally, an important point for the effectiveness of the methods are the run times. Table 4.2 shows the run times of this experiment for both the FM method and the direct iterative method. It is very obvious that the

FM method beats the direct method in terms of performance very easily. For the small vase image, the method is around a factor 10 faster. For the larger Mozart face images, the advantage even increases to a factor of about 15.

Specular Highlights

As before, the next experiment is the investigation of the effect of specular highlights on the method. For this experiment, it is possible to do more different single experiments. Since these experiments exhibit specular highlights, the correct choice for the reconstruction model is obviously the Phong model. However, in the previous chapters, interesting effects occurred when using the Lambertian model on these images. Therefore, four test scenarios are considered: FM method using the Phong model, Direct iterative method using the Phong model, FM method using the Lambertian model, and the Direct iterative method using the Lambertian model.

The error rates for all these experiments can be found in Tables 4.3 and 4.4 for the vase surface and in Tables 4.6 and 4.7 for the Mozart face surface. The first observation is similar to the previous paragraph: reconstructions using the FM method hardly contain any outliers. Even for the most extreme images, the method is stable enough to not contain any outliers above 5% for the vase. For the Mozart face, the outliers are constantly at around 30%, which is not surprising since the original surface contains outliers that are not reconstructed. Overall, the errors are very small when compared to the iterative method. This, however, does not hold in the same way for the reconstructions using the Lambertian model. The error rates go up for the FM method there as well. This suggests that the error made in these reconstructions is actually the model error, while for the reconstructions with the right model, a large portion of the error is caused by frequent outliers.

Figures 4.3-4.6 show some selected reconstructions of vase and Mozart face using the Phong FM method. Again, unlike the iterative reconstructions, the reconstructions are almost perfect with hardly any outliers. All the reconstructions are on the right scale.

In contrast to that, Figures 4.7-4.10 show the reconstructions using the Lambertian FM method. The results are much worse. Still, no significant outliers occur, but the shapes are distorted like for the iterative method and they are much too close to the camera, in particular those for the extreme images. This confirms that the errors in this case are actual model errors.

Finally, Tables 4.5 and 4.8 show the run times for the experiments using the Phong model. While for the iterative method larger run times are obtained for images with strong specular highlights, the run times for the

FM methods remain constant with increasing k_s, resulting in speedups of up to 750, even for the small vase image. This is a massive speedup, which clearly shows that the FM method is by far superior to the iterative method when it comes to computational efficiency.

Table 4.1: Comparison of errors of the FM method to the iterative approach.

			FM Method		Iterative Method	
Image	Size	f	Avg. Err.	Max. Err.	Avg. Err.	Max. Err.
Vase	128^2	125	3.87%	18.26%	3.86%	18.23%
Vase	128^2	250	1.42%	5.91%	1.40%	5.89%
Vase	128^2	500	0.56%	2.20%	1.80%	240.45%
Vase	128^2	1000	0.21%	0.58%	9.67%	251.50%
Vase	128^2	2000	0.24%	0.47%	12.98%	216.71%
Mozart	256^2	125	7.78%	53.02%	7.79%	53.00%
Mozart	256^2	250	4.13%	33.56%	4.12%	33.56%
Mozart	256^2	500	2.63%	33.66%	4.82%	302.63%
Mozart	256^2	1000	1.43%	9.62%	10.22%	296.13%
Mozart	256^2	2000	0.90%	33.29%	13.24%	297.09%

Table 4.2: Comparison of run times of the FM method to the iterative approach. All times given in seconds.

Image	Size	f	Run Time FM	Run Time Direct
Vase	128×128	125	0.20	2.04
Vase	128×128	250	0.18	2.75
Vase	128×128	500	0.18	2.12
Vase	128×128	1000	0.18	1.26
Mozart	256×256	125	0.92	19.27
Mozart	256×256	250	0.79	10.45
Mozart	256×256	500	0.74	11.89
Mozart	256×256	1000	0.79	11.59

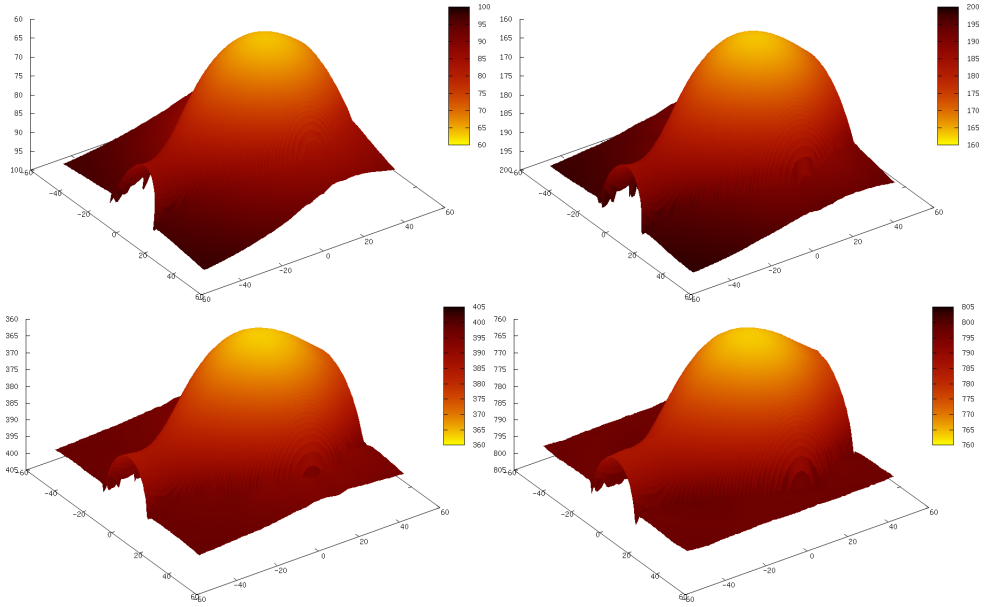

Figure 4.1: Reconstruction results of the Lambertian vase for different focal lengths using the FM method.

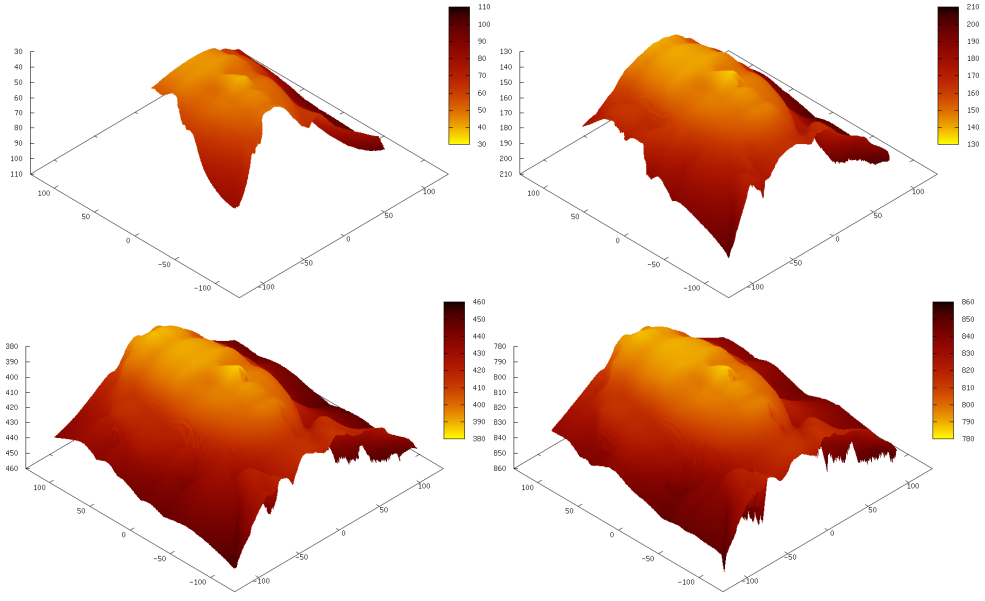

Figure 4.2: Reconstruction results of the Lambertian Mozart face for different focal lengths using the FM method.

Table 4.3: Comparison of errors of the FM method to the iterative approach on specular images of the vase.

		Phong Model			
		FM		Direct	
k_s	α	Avg. Err.	Max. Err.	Avg. Err.	Max. Err.
0.1	5	0.83%	2.57%	1.22%	2.78%
0.2	5	1.06%	2.96%	2.35%	4.88%
0.3	5	1.27%	3.30%	3.75%	7.24%
0.4	5	1.45%	3.64%	6.23%	10.72%
0.5	5	1.63%	3.81%	10.86%	16.46%
0.6	5	1.79%	4.11%	16.89%	24.00%
0.7	5	1.94%	4.37%	23.87%	33.56%
0.8	5	2.07%	4.59%	52.12%	201.43%
0.1	10	0.81%	2.41%	1.45%	3.00%
0.2	10	0.96%	2.62%	2.83%	5.32%
0.3	10	1.10%	2.69%	4.61%	8.00%
0.4	10	1.24%	2.93%	7.02%	11.41%
0.5	10	1.37%	3.14%	12.62%	21.47%
0.6	10	1.52%	3.41%	22.36%	52.89%
0.7	10	1.64%	3.60%	41.30%	180.68%
0.8	10	1.75%	3.80%	77.92%	239.11%
0.1	20	0.77%	2.32%	1.94%	162.62%
0.2	20	0.89%	2.25%	3.46%	5.78%
0.3	20	0.99%	2.35%	5.32%	8.35%
0.4	20	1.09%	2.32%	8.58%	12.54%
0.5	20	1.18%	2.47%	16.48%	159.41%
0.6	20	1.28%	2.63%	28.62%	183.04%
0.7	20	1.38%	2.61%	46.07%	221.88%
0.8	20	1.47%	2.74%	83.44%	292.25%

Table 4.4: Comparison of errors of the FM method to the iterative approach on specular images of the vase.

		Lambertian Model			
		FM		Direct	
k_s	α	Avg. Err.	Max. Err.	Avg. Err.	Max. Err.
0.1	5	5.91%	7.57%	4.18%	5.21%
0.2	5	11.50%	13.19%	8.91%	10.63%
0.3	5	17.36%	19.10%	14.08%	16.40%
0.4	5	23.64%	25.35%	19.80%	22.60%
0.5	5	30.31%	31.96%	26.16%	29.16%
0.6	5	37.77%	39.35%	33.38%	36.64%
0.7	5	46.19%	47.62%	41.72%	45.13%
0.8	5	56.12%	57.35%	51.72%	55.20%
0.1	10	5.87%	7.42%	3.94%	5.21%
0.2	10	11.39%	12.90%	8.41%	10.63%
0.3	10	17.09%	18.56%	13.31%	16.19%
0.4	10	23.33%	24.79%	18.82%	22.40%
0.5	10	30.10%	31.50%	25.06%	29.16%
0.6	10	37.56%	38.90%	32.14%	36.64%
0.7	10	45.98%	47.20%	40.31%	45.13%
0.8	10	55.94%	56.99%	50.18%	55.19%
0.1	20	5.83%	7.33%	3.64%	5.21%
0.2	20	11.17%	12.58%	7.75%	10.40%
0.3	20	16.96%	18.29%	12.37%	16.18%
0.4	20	23.05%	24.32%	17.47%	22.20%
0.5	20	29.81%	31.02%	23.45%	28.97%
0.6	20	37.27%	38.41%	30.28%	36.47%
0.7	20	45.61%	46.68%	38.22%	44.84%
0.8	20	55.61%	56.53%	47.96%	54.96%

Table 4.5: Comparison of run times of the FM method to the iterative approach on specular images of the vase. All times given in seconds.

k_s	α	Run Time FM	Run Time Direct
0.1	5	0.228014	2.44415
0.2	5	0.208013	2.35215
0.3	5	0.208013	2.10413
0.4	5	0.204013	3.1722
0.5	5	0.204013	5.54835
0.6	5	0.204013	8.30852
0.7	5	0.208013	10.9567
0.8	5	0.204013	12.4648
0.1	10	0.212014	2.48015
0.2	10	0.212014	2.36415
0.3	10	0.212014	2.34415
0.4	10	0.212014	2.90418
0.5	10	0.208013	6.27639
0.6	10	0.208013	15.569
0.7	10	0.212014	15.933
0.8	10	0.204013	42.5787
0.1	20	0.216014	2.59616
0.2	20	0.212013	2.27214
0.3	20	0.212013	2.20814
0.4	20	0.208013	3.46422
0.5	20	0.216014	12.3408
0.6	20	0.276018	20.4413
0.7	20	0.208013	67.8722
0.8	20	0.212013	124.976

Table 4.6: Comparison of errors of the FM method to the iterative approach on specular images of the Mozart face.

		Phong Model			
		FM		Direct	
k_s	α	Avg. Err.	Max. Err.	Avg. Err.	Max. Err.
0.1	5	2.98%	33.66%	3.43%	197.07%
0.2	5	3.37%	34.00%	3.14%	109.70%
0.3	5	3.67%	34.00%	3.65%	40.17%
0.4	5	3.94%	34.10%	4.82%	43.49%
0.5	5	4.33%	34.07%	8.53%	49.35%
0.6	5	4.70%	33.14%	15.01%	118.82%
0.7	5	5.00%	32.69%	27.80%	150.06%
0.8	5	5.22%	32.32%	64.76%	222.05%
0.1	10	2.89%	33.98%	3.84%	197.07%
0.2	10	3.22%	34.07%	3.14%	114.02%
0.3	10	3.52%	34.10%	4.00%	41.27%
0.4	10	3.79%	34.40%	6.01%	44.65%
0.5	10	4.07%	34.41%	11.64%	126.99%
0.6	10	4.39%	33.60%	25.04%	149.22%
0.7	10	4.67%	32.91%	50.45%	182.65%
0.8	10	4.89%	32.69%	88.73%	271.87%
0.1	20	2.80%	34.03%	4.71%	197.07%
0.2	20	2.98%	34.34%	4.19%	132.07%
0.3	20	3.23%	34.73%	5.39%	134.98%
0.4	20	3.45%	34.78%	8.61%	145.20%
0.5	20	3.70%	35.15%	18.74%	161.68%
0.6	20	3.97%	34.84%	31.22%	186.09%
0.7	20	4.13%	34.08%	55.24%	226.60%
0.8	20	4.31%	33.53%	108.23%	292.57%

Table 4.7: Comparison of errors of the FM method to the iterative approach on specular images of the Mozart face.

		Lambertian Model			
		FM		Direct	
k_s	α	Avg. Err.	Max. Err.	Avg. Err.	Max. Err.
0.1	5	7.85%	26.82%	6.24%	179.03%
0.2	5	13.47%	25.86%	10.42%	25.37%
0.3	5	19.24%	30.51%	15.23%	29.88%
0.4	5	25.41%	35.76%	20.54%	34.99%
0.5	5	32.26%	41.42%	26.54%	40.56%
0.6	5	39.66%	47.64%	33.38%	46.75%
0.7	5	47.91%	54.66%	41.32%	53.80%
0.8	5	57.56%	62.99%	51.07%	62.23%
0.1	10	7.71%	27.11%	6.01%	179.03%
0.2	10	13.21%	25.65%	9.75%	25.09%
0.3	10	19.08%	30.54%	14.23%	29.78%
0.4	10	25.24%	35.77%	19.16%	34.85%
0.5	10	31.96%	41.27%	24.81%	40.23%
0.6	10	39.41%	47.51%	31.30%	46.41%
0.7	10	47.68%	54.57%	38.88%	53.47%
0.8	10	57.40%	62.91%	48.44%	61.94%
0.1	20	7.53%	27.17%	6.24%	179.03%
0.2	20	12.90%	25.65%	9.12%	25.03%
0.3	20	18.68%	30.34%	13.23%	29.50%
0.4	20	24.87%	35.58%	17.79%	34.55%
0.5	20	31.54%	41.12%	23.02%	39.93%
0.6	20	39.00%	47.38%	29.12%	46.11%
0.7	20	47.31%	54.31%	36.24%	53.01%
0.8	20	57.06%	62.60%	45.34%	61.38%

Table 4.8: Comparison of run times of the FM method to the iterative approach on specular images of the Mozart face. All times given in seconds.

k_s	α	Run Time FM	Run Time Direct
0.1	5	0.88	13.77
0.2	5	0.87	13.50
0.3	5	0.87	13.08
0.4	5	0.85	13.79
0.5	5	0.84	22.10
0.6	5	0.86	38.91
0.7	5	0.85	42.88
0.8	5	0.85	46.53
0.1	10	0.88	13.52
0.2	10	0.87	14.34
0.3	10	0.87	14.02
0.4	10	0.88	14.02
0.5	10	0.87	25.04
0.6	10	0.86	112.7
0.7	10	0.86	110.0
0.8	10	0.86	152.7
0.1	20	0.89	15.26
0.2	20	0.89	14.40
0.3	20	0.88	13.55
0.4	20	0.87	14.42
0.5	20	0.88	66.38
0.6	20	0.89	93.82
0.7	20	0.89	279.2
0.8	20	0.88	326.5

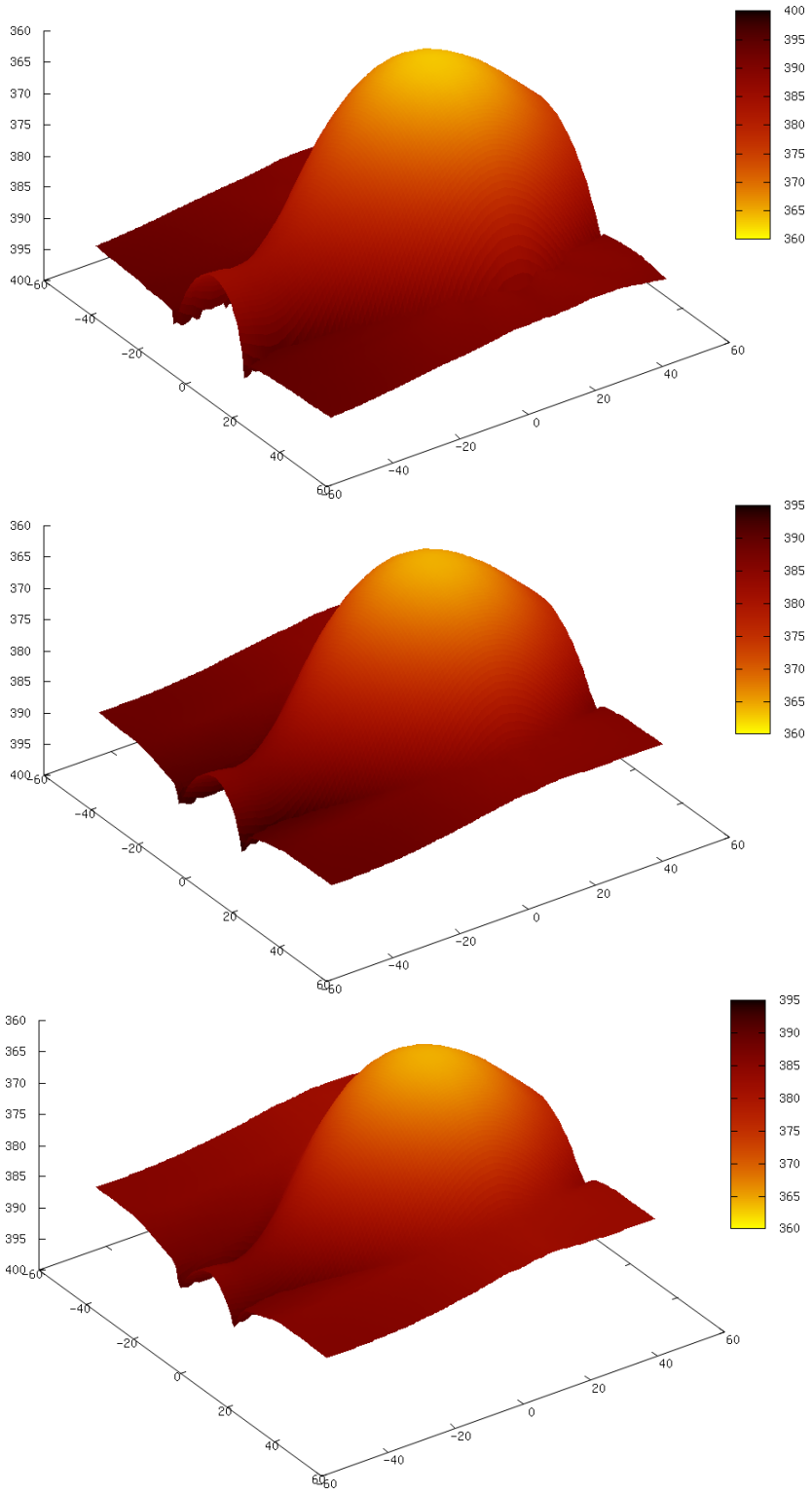

Figure 4.3: Reconstruction results of the vase for $k_s = 0.2, 0.5, 0.8$, $\alpha = 5$ using the Phong FM method.

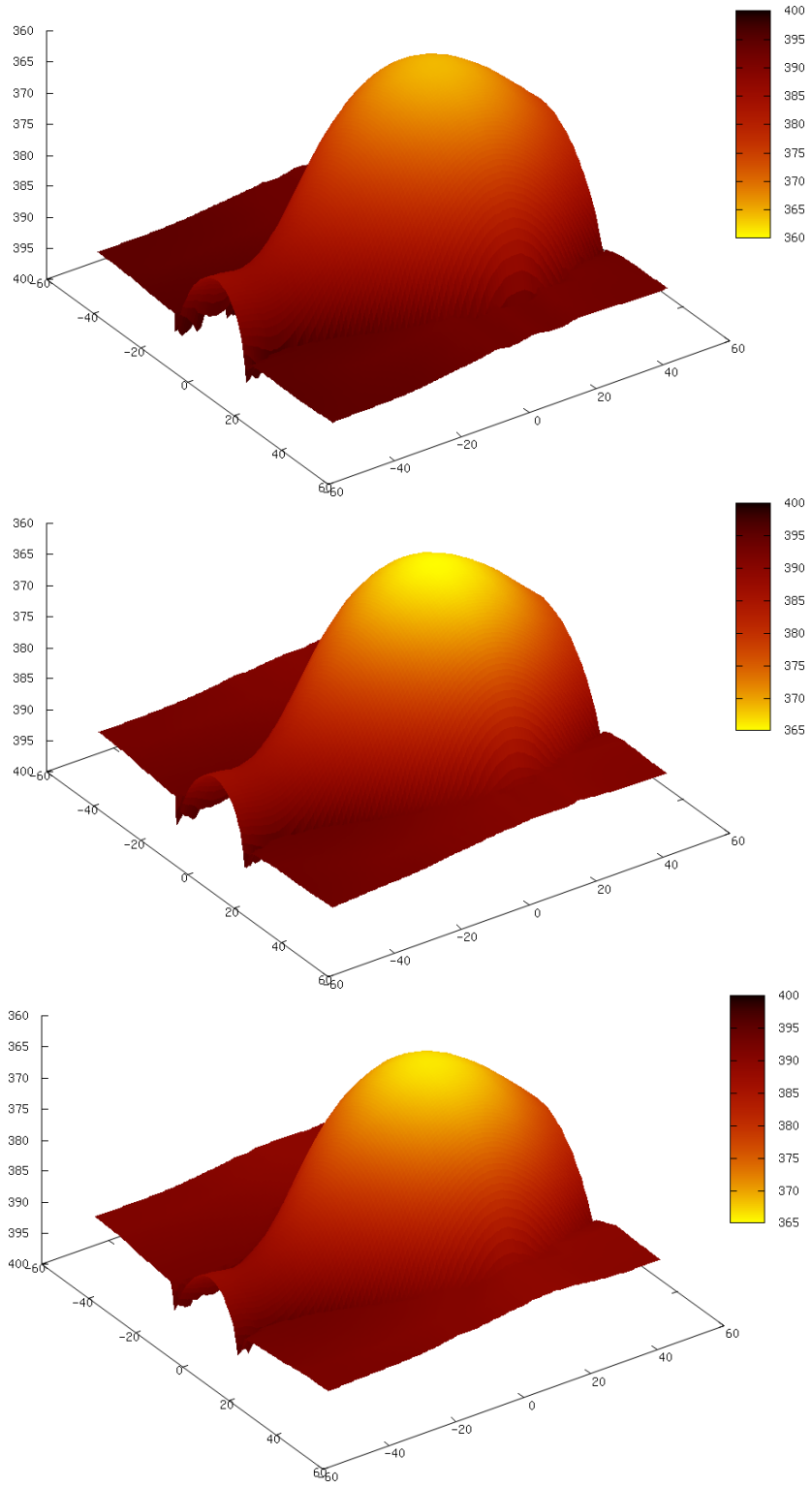

Figure 4.4: Reconstruction results of the vase for $k_s = 0.2, 0.5, 0.8$, $\alpha = 20$ using the Phong FM method.

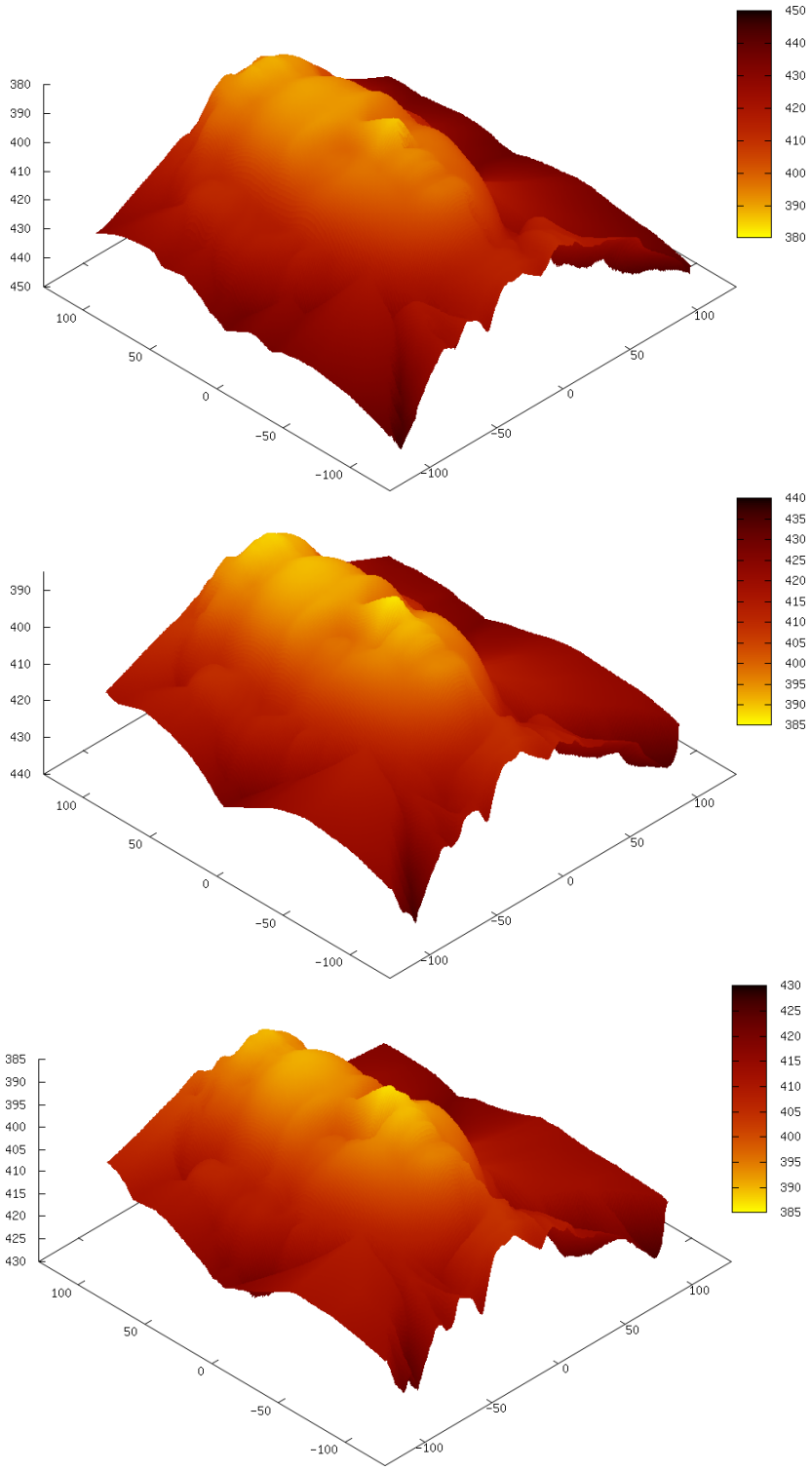

Figure 4.5: Reconstruction results of the Mozart face for $k_s = 0.2, 0.5, 0.8$, $\alpha = 5$ using the Phong FM method.

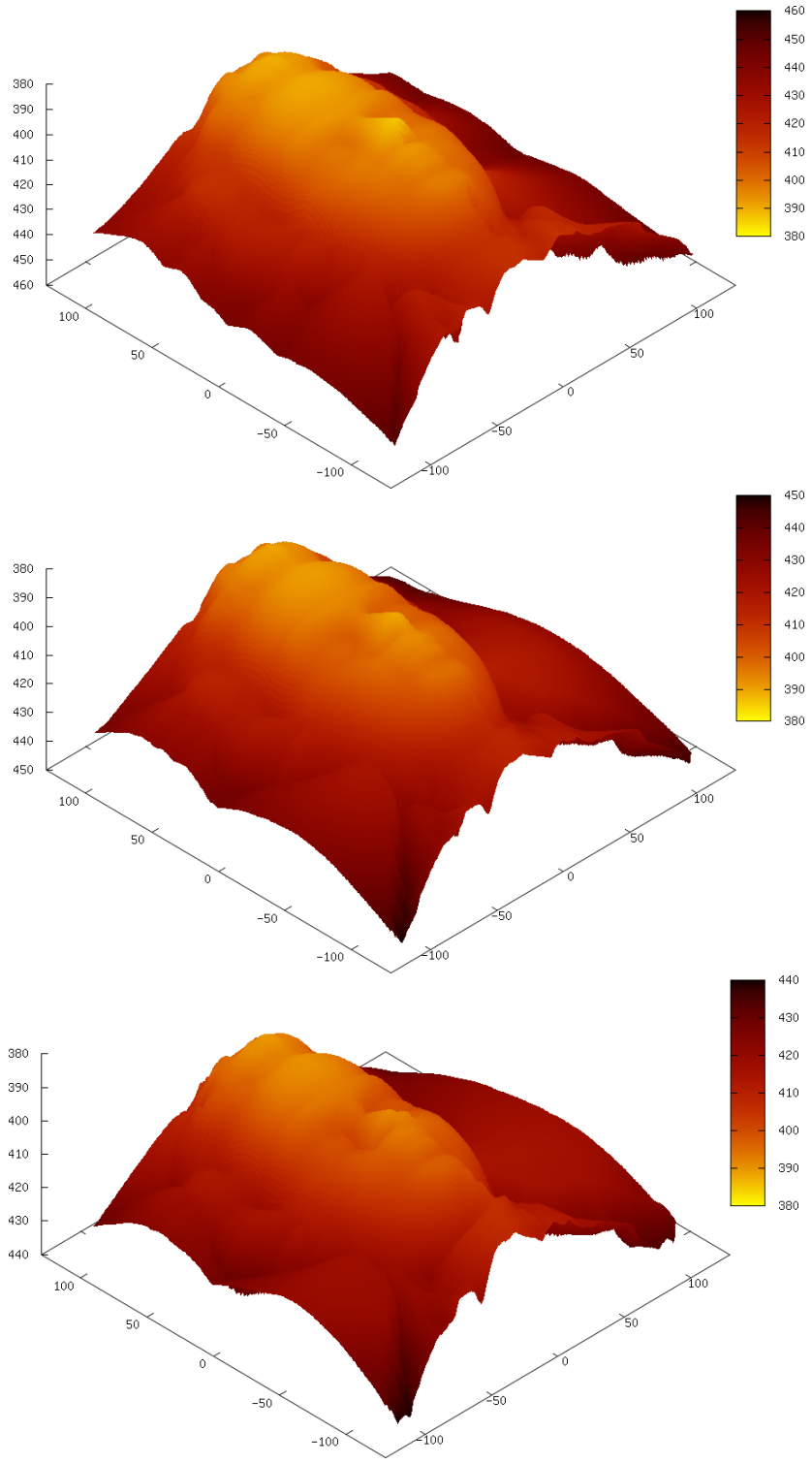

Figure 4.6: Reconstruction results of the Mozart face for $k_s = 0.2, 0.5, 0.8$, $\alpha = 20$ using the Phong FM method.

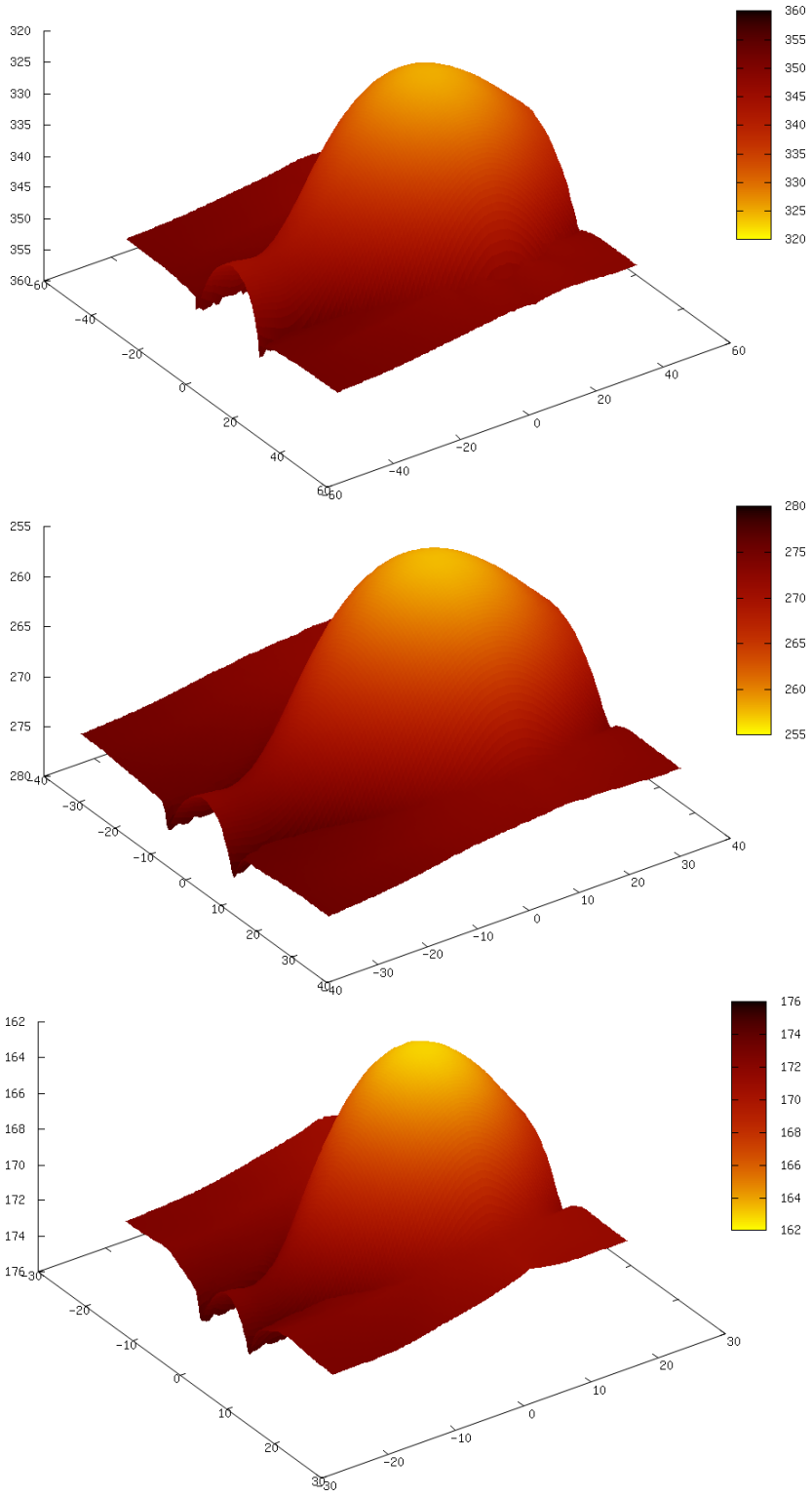

Figure 4.7: Reconstruction results of the vase for $k_s = 0.2, 0.5, 0.8$, $\alpha = 5$ using the Lambertian FM method.

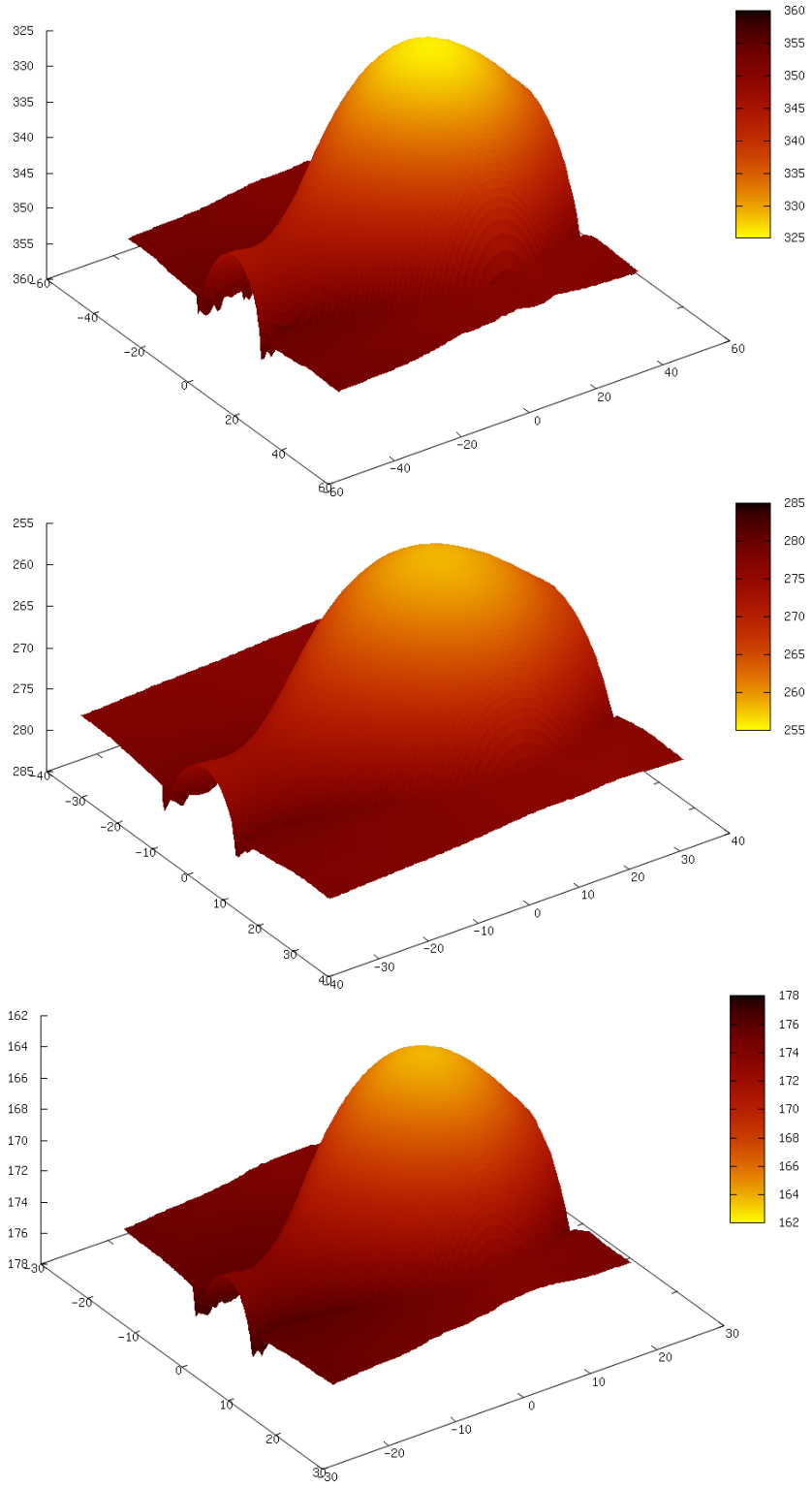

Figure 4.8: Reconstruction results of the vase for $k_s = 0.2, 0.5, 0.8$, $\alpha = 20$ using the Lambertian FM method.

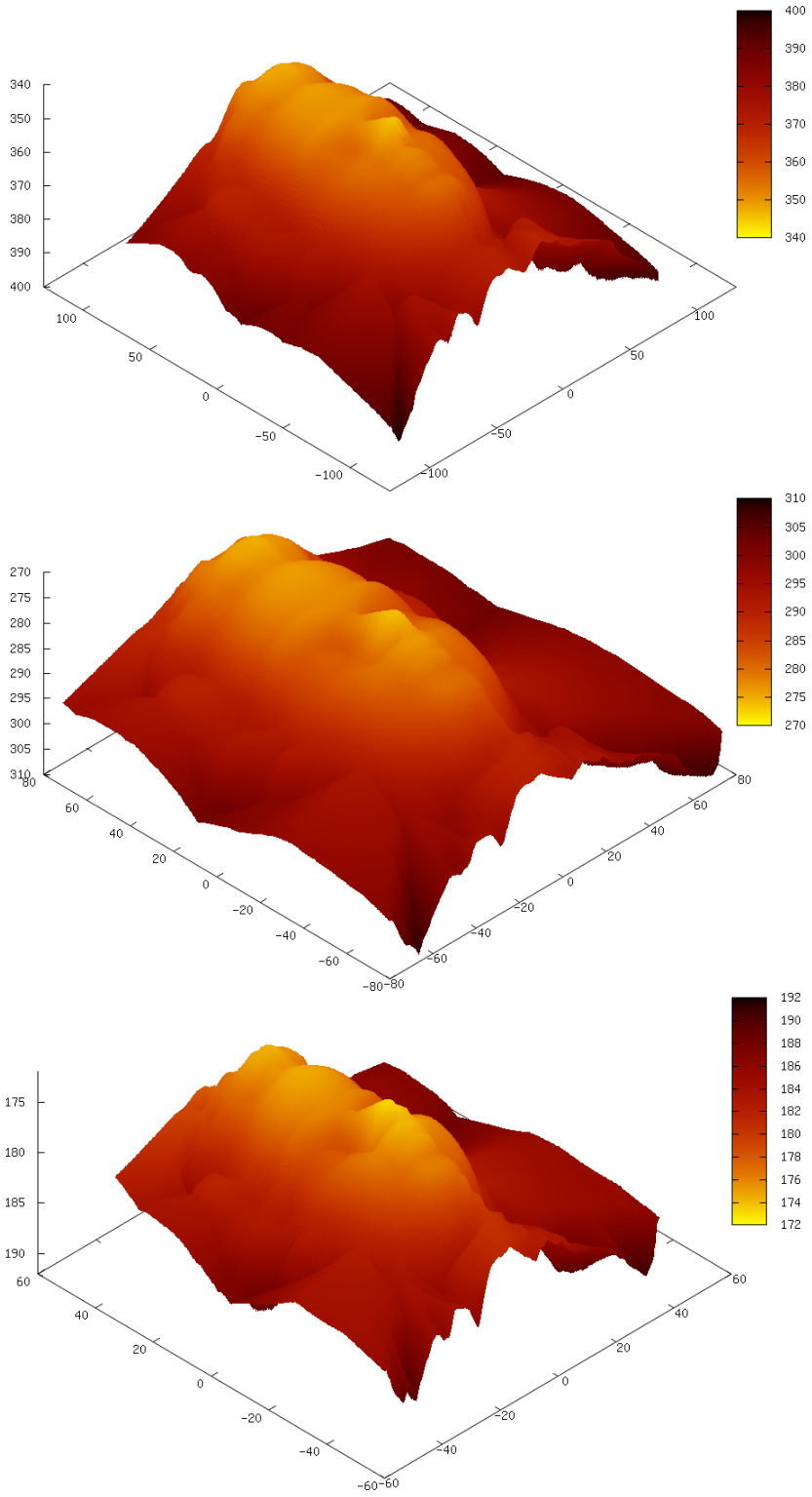

Figure 4.9: Reconstruction results of the Mozart face for $k_s = 0.2, 0.5, 0.8$, $\alpha = 5$ using the Lambertian FM method.

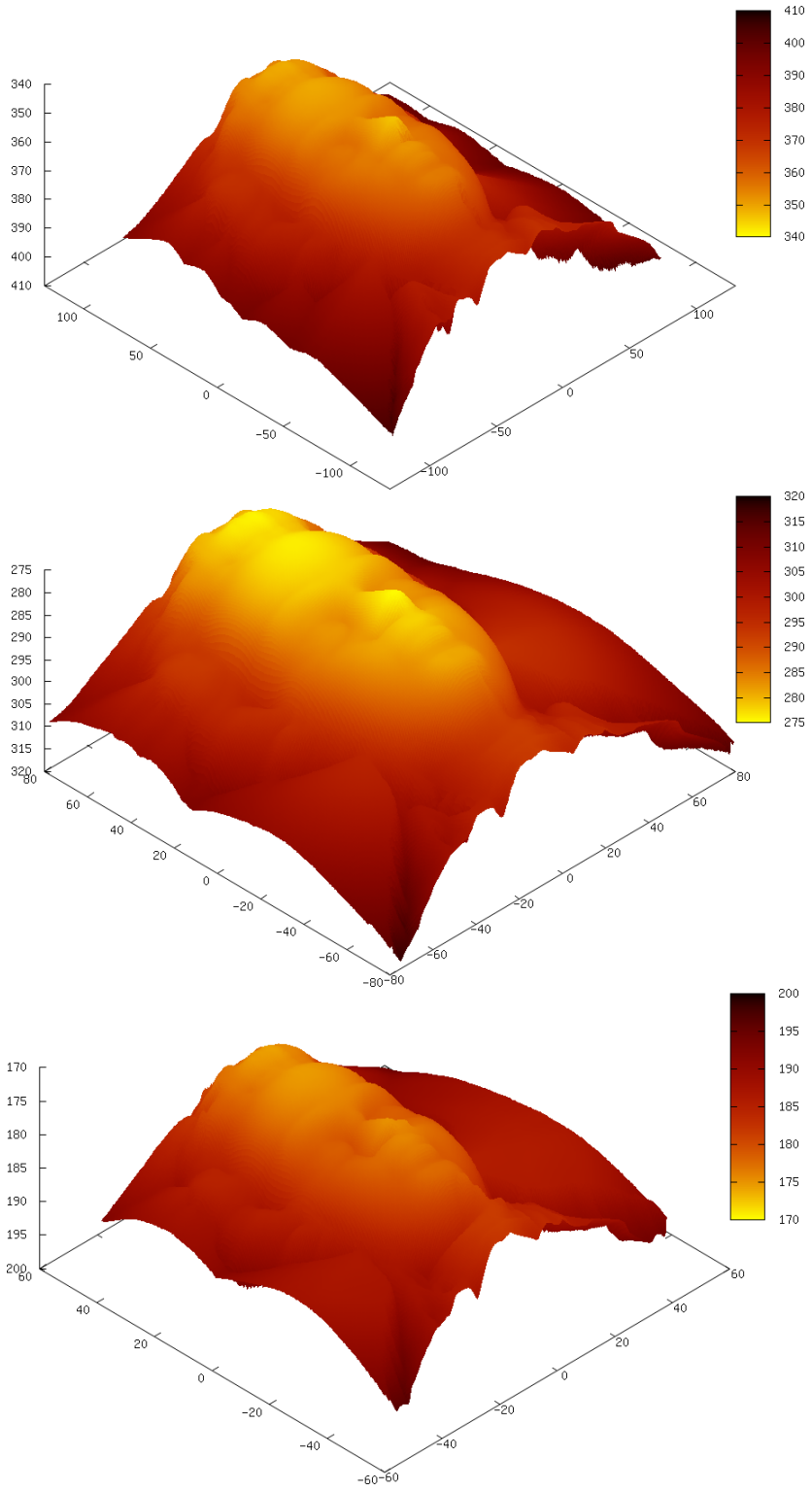

Figure 4.10: Reconstruction results of the Mozart face for $k_s = 0.2, 0.5, 0.8$, $\alpha = 20$ using the Lambertian FM method.

4.9 Real-World Images

So far, no experiments on real images have been shown. SfS on real-world images is a difficult task. However, in this section results will be shown on some real images. These images are chosen carefully, i.e., images are chosen that are possible to reconstruct. It is possible to tackle more difficult images than that, but this shall be discussed in an extra chapter.

There are a couple of differences between synthetic and non-synthetic images. To list only the most important ones,

1. Real surfaces are not Lambertian. They do not even fit to the Phong model. In reality, surfaces are much more complex than the simple models used for it.

2. Reality has no point light surfaces. Every light source has an extension.

3. Light sources are not in the optical centre of the camera. We can put them close to it, but not exactly in that location. This might cause occlusions and shadows in the image.

4. Light can be reflected on a surface and reflected to other surfaces. This is usually not modelled, in particular if there are surrounding objects not present in the scene to be reconstructed.

5. Real cameras are no ideal pinhole cameras. While we can build a pinhole camera from cardboard, we will not actually use that to take pictures. We buy a commercial camera with a lens. These are no actual pinhole cameras.

6. Scenes might not have uniform reflectance properties.

7. While the synthetic images discussed so far have been of size 256×256 maximum, photos taken by modern cameras can have sizes of several megapixels.

With synthetic images, it is possible to control all these things, and create a world that fits perfectly to all model assumptions. However, reality differs from that. The question is only whether a model is close enough to reality to produce convincing results. A standard photo camera is somewhat close to a pinhole camera. A camera flash is not exactly in the optical centre, but close to it. Surfaces exhibit specular highlights, which do not follow the Phong model, but the model might be close enough anyway to produce satisfactory results.

Another important issue is the size of the images. In fact, this is the main reason this experiment has not been included in previous chapters. Real images can be large. With the conventional iterative algorithms, the reconstruction can take a considerable amount of time. While this is not as drastic with the method presented in Chapter 3 as with the optimal control approaches, computation times can still be quite large. In this section, not only the visual reconstruction quality, but also the reconstruction time on large real-world images will be evaluated. This demonstrates quite impressively the advantage of the fast marching method in terms of computational complexity.

Figure 4.11 shows a real input image of three chess figures. The image has been taking with a normal digital camera with built-in flash. The original size of the image is 3264×2448, the focal length has been provided by the camera, and the standard parameters $k_s = 0.7$, $\alpha = 10$ have been used. Figure 4.12 shows the reconstruction results using both the iterative method and the FM method. Both reconstructions are visually quite good, however, as experienced before, the direct method suffers from outliers at discontinuities. This is not the case for the FM method. Overall, the FM results appears a bit better.

Figure 4.11: Photograph of three chess figures.

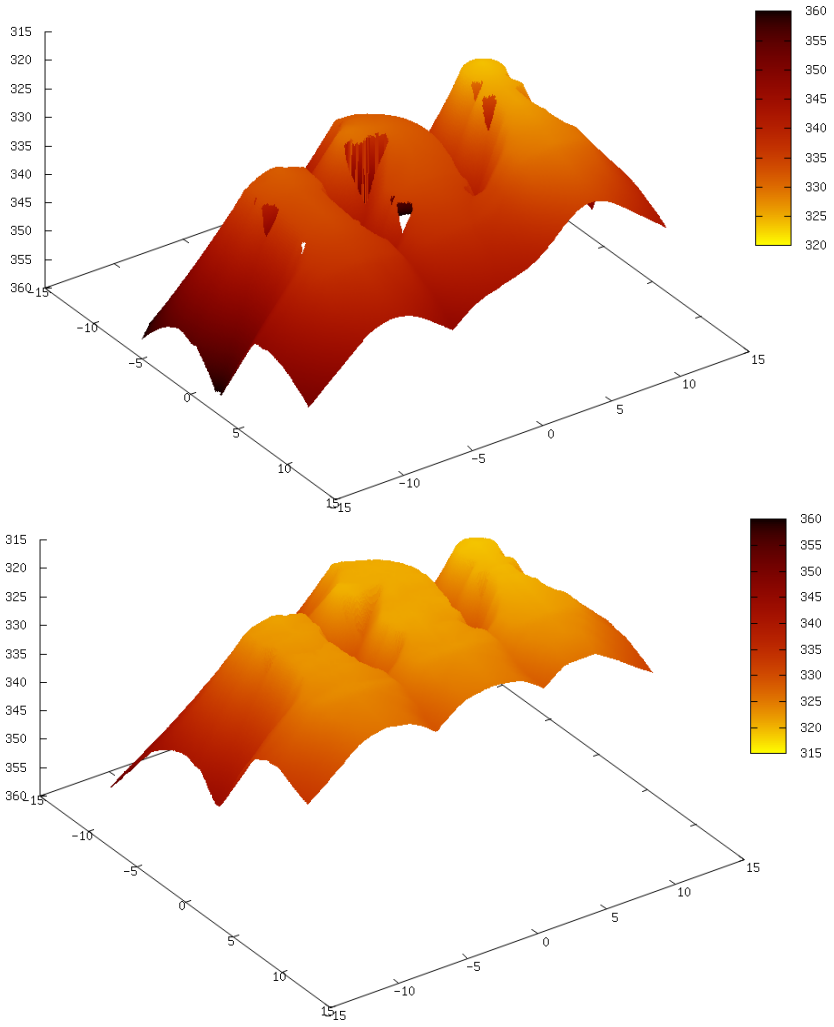

Figure 4.12: Reconstruction of the chess figures using the direct method and the FM method.

Finally, Table 4.9 shows the run times of the FM method on the full image and a downsampled one to 408×306 pixels. Here, the performance advantage of the FM method becomes very clear. Even though cascading multigrid has been used to accelerate the direct method, the reconstruction of the full image still takes almost three hours, while the FM method converges in around two minutes. Also, the comparison to the smaller image shows that the advantage in computation time grows with an increasing size of the problem. For practical applications on large data sets, the FM method is clearly superior.

Table 4.9: Comparison of run times of the FM method to the iterative approach for the chess image. All times given in seconds.

Size	Run Time FM	Run Time Direct Method
408×306	1.52	32.91
3264×2448	133.34	11423

Chapter 5

Parallelisation of Shape from Shading Methods

In the previous chapter, a method was presented that massively improved the performance of the SfS method. The quality of the reconstruction has not been compromised, it has even been improved in some cases. On normal test images for SfS, which usually are in the range of up to 256×256 pixels, computation times close to real-time have been achieved. For very large images of eight Megapixels, the computation times were still staying in a modest range of around two minutes, which is the computation time of a small 256×256 image for many SfS methods.

Nevertheless, one question always remains: Can it be even faster? Certainly, it might be possible to implement small computational tricks like lookup tables for expensive functions like the exponential function or square roots. Another possibility would be to use an $O(n)$ variant of the FM algorithm [104], which involves quantisation of the depth, which could compromise quality if not done very carefully. However, the potential of these acceleration technique is somewhat limited.

In the recent years, there has been another trend towards improvements in computational efficiency: parallelisation. Parallelisation bears a much greater potential for speed-ups than small numerical improvements. This trend has mainly been driven by affordable multi-core CPUs and the increasing popularity of graphics cards suitable for highly parallel computations. In this chapter, the potential of possible parallelisations of the FM SfS method will be discussed. The evaluation of this will be more a proof of concept, an investigation whether further performance improvements are

still possible. In particular on the GPU side, there might be other options to improve the performance, only one possible path is explored here.

5.1 Contributions

In this chapter, two ways of parallelising the FM method for perspective SfS using the Phong model will be presented. One of these methods runs on machines with more than one CPU core, which allows for non-massive parallelisation. The related publication to this is [10], where this method is discussed in a more general context. However, here we concentrate on the application to the SfS method. In a second part, a way to parallelise SfS massively on a GPU is introduced. This has not been published so far.

5.2 Parallelisation of the FM Approach

Parallelising the FM method is a difficult task. By itself, the FM method is very badly suited for parallelisation. As information propagates over the image, only very few pixels are to be computed simultaneously, usually only between two and four pixels at once. After updating these pixels, a global operation on all the pixels marked as trial has to be performed: The pixel with the smallest value has to be determined, and afterwards its neighbours have to be updated. Updating four pixels is certainly not enough to make parallelisation efficient, but in a naïve approach the FM method does not contain more potential of parallelisation.

A more efficient way to make FM methods suited for parallel computing is to decompose the domain in different subdomains. One such technique has been proposed in [31]. In this method, the image is evenly split in domains of equal size, and within each domain, the propagating wave fronts are computed on separate threads. When wave fronts collide, i.e. a wave front hits the boundary of its domain, rather complex correction computations have to be performed.

A much simpler and easier to implement way to parallelise a FM method is described in [10]. Here, different seeds, i.e. different local minima in case of the SfS problem are assigned to different threads, and only at collisions of two wave fronts, which may occur anywhere in the image, it has to be checked which wave front has the better, i.e. lower, result. This involves only very few checks. In the mentioned paper, the method is described in detail and evaluated in the more abstract context of a general Eikonal equation.

For the purpose of SfS, it is sufficient to describe the general idea and a rough description of the implementation. It is the same implementation that has been used in [10], just for the perspective SfS with Phong PDE, which is not exactly an Eikonal equation, but quite close to it. The algorithm can be summarised to the following steps:

1. Local minima in depth are determined and initial values computed, as described in the chapter on the single-threaded FM method.

2. These local minima are assigned to the different threads in a balanced way. In practice, the set of seeds is split in two halves in a way that each of the two threads contains the same number of seeds. When dealing with more than two threads, these two halves are again split in half, usually with alternating splitting direction. In computer graphics, such a splitting concept is related to a kd-tree.

3. Each of the threads starts computing its solution using the normal FM method. However, when updating a pixel, each thread checks whether another thread has already updated this pixel. If that is the case, it compares its value to the previously computed value. If the previously computed value is larger, it is overwritten and the algorithm proceeds as usual. If it is lower, the other wave front has a better solution. In this case, the propagation stops at this point, removing this pixel from the list of trial points and continuing as usual.

4. The algorithm terminates when all threads converge.

This is a very simple, but effective procedure to parallelise a FM on a multi-core architecture. In the following, this method will be applied to SfS on a dual-core CPU, comparing the run times to the sequential algorithm described in the previous chapter.

The Hardware

The experiments are done on the same hardware as the experiments in the chapters before, an Intel Core2 Duo E4600 at 2.4 GHz with 2 MB cache and 2 GB of RAM, running Linux.

Results

To evaluate the performance of this approach, it is evaluated on two test images from the previous chapters: the standard vase test image (Lambertian,

Table 5.1: Run times of parallelised FM on a dual-core CPU.

Image	Size	One Thread	Two Threads	Speed-up factor
Vase	128×128	176ms	91ms	1.93
Chess	3264×2448	132.2s	78.98s	1.67

$f = 500$) and the chess set test image used to evaluate the single-threaded FM method. Since the results are qualitatively the same compared to the results using the single-threaded FM method and only the run times differ, there is no need to show the reconstructions again, they can be found in the respective chapters.

Table 5.1 shows the run times using one and two threads on the dual-core CPU. For the very easy experiment of the vase, an almost perfect speed-up of 1.93 is obtained using two threads. The best possible speed-up that could have been achieved here would be a factor 2. However, for the more complex experiment of the chess figures, also the achieved speed-up becomes smaller, since the domain decomposition is more complex as well and hence not as optimal. For this experiment, a speed-up of 1.67 is achieved, which is still reasonable.

5.3 A Parallel Block FM method

In the previous section, a non-massive parallelisation of the FM method on multi-core CPUs has been introduced. Now, is there also a chance to parallelise FM SfS massively on a GPU? Usually, for the unmodified FM method, this is hardly possible. When a pixel is accepted, four pixels need to be updated independently of each other, then the method must be synchronised again. While this might still be somehow suitable for non-massive parallelisation, this would perform very badly on a GPU, since copying these four pixels into the shared memory of the graphics card alone and back after updating would be slower than updating these four pixels on the CPU.

However, there is a way in-between FM and an iterative method. Splitting up the image into blocks of suitable size for parallelisation, it is possible to use a FM-like idea on these blocks. Initialising all pixels as in the iterative method and iterate them all, independently of each other, parallely on the GPU, until convergence. Since these blocks are relatively small and the computation can be done parallely on the GPU, this can be done rather fast. Then, the block containing the closest point is marked as accepted

and the neighbouring blocks are updated, in parallel and iteratively on the CPU. This is repeated until all blocks are marked accepted. The idea on which this method is based is similar to the one in [40]. In the following, a description of the actually implemented algorithm is given. The algorithm has been implemented using the CUDA framework by NVIDIA [54].

Details on the Implementation

The basic setup for this method is quite simple. First, the image domain is split into square blocks with side lengths that are powers of 2. In practice, this means side lengths of up to 16 pixels, since the shared memory of each processor on the GPU is very limited, and 32×32 depth values are already too much for this very restricted memory, which is necessary to achieve good performance. The actual FM method is running on these blocks instead of pixels. The FM part of the algorithm is still done on the CPU, using a single thread.

First, as for the normal FM method, local brightness maxima are determined and the blocks marked as trial and put into a heap. All pixels in the resulting depth field are initialised to a very large value. The normalised brightness of the input image is transferred into the texture memory of the graphics card. The depth field to store the result is transferred into the global memory of the graphics card

Then, the blocks that contain local brightness maxima are updated. An update of a block is done in the following way: First, the block is transferred into shared memory. In addition to the mere block, also one pixel around the boundary of the block is transferred to shared memory. Then, one iteration of the iterative method is performed, parallely, using a Jacobi solver. The use of a Jacobi solver actually has a very big advantage in this case. For a Jacobi solver, computations on each pixel are independent of the results of the computation of the other pixels. The Gauß-Seidel-like scheme and the fast sweeping method used in the sequential method are abandoned here.

After one iteration has been performed, all threads are synchronised and the next iteration begins. This is repeated until convergence. At blocks of 16×16 pixels, this usually happens in significantly less than 100 iterations, which is far less than the several thousand iterations needed for convergence on a large image. After the block converged, the result is written back to the global memory of the graphics card. In the case that a block has never been updated before, the first iteration is actually an initialisation. Since all values are set to very large, or infinity, the pixels are initialised in the same way as described in the previous chapter, depending on the shading parameters, which have also been transferred to shared memory

for performance reasons.

This update is done for all the initial trial blocks. Since the graphics card can only update a few blocks at once, a limited set of blocks is given to the graphics card at once. Usually, this lies in the range of 16 to 32 blocks at once, which are then scheduled to the actual processors by the graphics card.

Before writing back a block to global memory after convergence, another thing is done. The minimum value of the block is determined by a minimum reduction [54]. Essentially, the block is split in two halves in x-direction, and each pixel in the left part compares itself to its corresponding pixel in the right part. If it is smaller, it overwrites its value with the value of the other pixel. This is done recursively until the leftmost column in the block contains the minimum pixels of each row. Afterwards, an analogous procedure is done in y-direction, but only in the first column. At the end of this procedure, the top left pixel contains the minimum value of the block, this value is returned by the update and used as the value of the block within the heap for the FM method. A similar technique is used for the absolute difference between to iterations, just that here the maximum of the differences is determined. Note that for performing a Jacobi-iteration, it is necessary to reserve twice the space anyway, so these minimum or maximum computations can be done without losing the actual depth data that is iterated.

Once all initial trial blocks have been updated, the normal FM routine commences. First, the block containing the smallest pixel is determined. This block is marked accepted, and its neighbours are set to trial and updated. Note that now, only up to four blocks are to be updated, which is not ideal in terms of performance. We perform this update in the same way as described above. After these blocks are updated, and also their values are updated in the trial heap, the next smallest block is accepted and its non-accepted neighbours updated. This is repeated until all blocks are accepted.

At the end, the resulting depth data is written back from the global memory of the graphics card to the RAM of the host and saved to the hard disk. In the performance evaluation, the initialisation phase, e.g. loading the image to the host memory, and saving the file to the hard disk are excluded from the run times, since these steps are not part of the actual method and depend severely on the hardware used.

The Hardware

We do the experiments on the same hardware as the experiments in the chapters before, an Intel Core2 Duo E4600 at 2.4 GHz with 2 MB cache and 2 GB of RAM, running Linux. For this experiment, though, the graphics card is also relevant. It is a NVIDIA GeForce GTX 260+, 896 MB GDDR3 RAM, 216 streaming multiprocessors.

Experimental Results

The algorithm will be tested on three scales:

- A very small image. Here, one cannot expect a significant speed-up. The results may even become worse.
- A medium-sized image. For that, another image of a chess figure is used, shown in Figure 5.1. This image has size 2112×2112, a size of 4.46 Megapixels.
- A large image. Again, the chess image shown in Figure 4.11 is used.

First of all, the results of all methods are visually and qualitatively identical to the ones obtained using the plain FM method. Figure 5.2 shows the reconstruction of the single chess figure. The reconstruction quality is quite good, comparable to the results obtained for the three figures.

The most important issue, however, are the run times in this case. For all experiments, blocks of size 16×16 pixels have been used. For the initial iteration of all blocks, up to 16 blocks have been processed at once. In the actual FM evolution, however, only up to four blocks have to be updated simultaneously, on average only about three blocks, therefore in this case only about three blocks are updated simultaneously. This is a major issue in the parallelisation of the FM method, since only one block is given to a processor on the graphics card at once, computing only three blocks is certainly not optimal for this architecture.

Nevertheless, as Table 5.2 shows, the results are still quite convincing. At the smallest problem size of only 128×128 pixels (or 8×8 blocks), the result already improves significantly to just 32 milliseconds computation time. While even at this small problem, there is already a considerable advantage to the single-threaded result, at large problem sizes speed-up factors of almost 10 can be achieved, compared to the single core FM implementation, compared to the iterative method, the speed-up even lies in the range of almost a factor 900.

Figure 5.1: Input image: A chess queen.

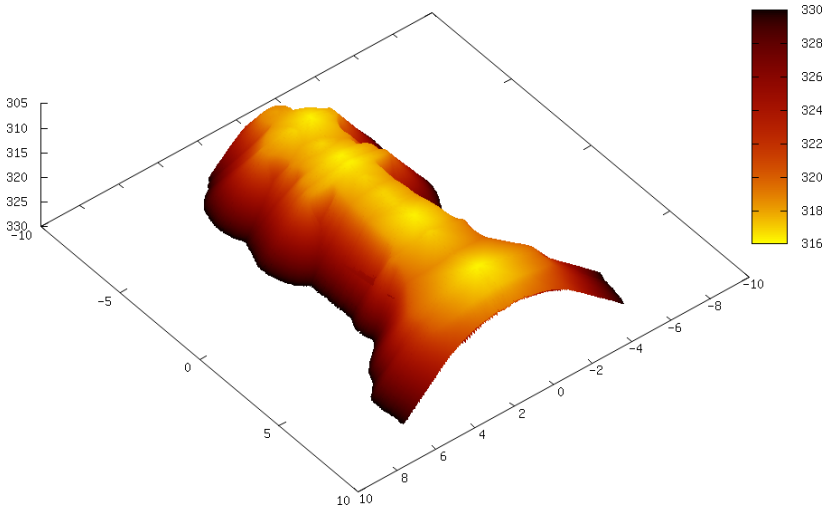

Figure 5.2: Reconstruction result of the queen image.

Table 5.2: Run times of the FM method vs. the parallel block FM method on the GPU.

Image	Size	FM	Block FM	Speed-up
Vase	128×128	176ms	32ms	5.5
Queen	2112×2112	45.76s	7.78	5.9
Chess figures	3264×2448	132.2s	12.60s	10.5

Usually, when bringing a method to the GPU, one hopes to achieve a speed-up in the range of a factor 30 to 40. However, since the FM method is extremely difficult to parallelise, a factor 10 is a good result. After all, it is possible to perform perspective SfS including specular highlights on an eight Megapixel image in 13 seconds, which is good considering the fact that even the fastest iterative method available takes more than three hours and even a FM method takes more than two minutes.

5.4 Conclusions

With the results presented in this chapter, SfS can be considered to run in real-time. For a problem size of 128×128 pixels, which is very common in SfS applications, a reconstruction is possible in just 32 ms, or 31.25 frames per second. And even for very large image sizes of eight Megapixels, it is possible to achieve convincing results in just a couple of seconds.

An important point one should note is that all this can be done on standard PC hardware. At the present date (beginning of 2010), the CPU used for all the computations is worse than the average CPU included in PCs sold to end-users. Also, in particular for gaming PCs, there are many customers who buy more expensive and stronger graphics cards than the GLX 260 that has been used here. Naturally, the computation times will even reduce further with faster hardware becoming available. However, I think the results obtained here give a good impression on what performance is possible for SfS using efficient algorithms, from single-core systems up to parallel computing on a GPU.

Chapter 6

Shape from Shading on Real-World Images

So far, almost exclusively synthetic experiments have been discussed. The only exception to this were the chess images in the previous chapters, where reasonable results have been obtained. In said chapters, however, the focus has been the computation time. In this chapter, the quality of SfS methods on real-world images will be discussed. Also, it will be addressed in which cases SfS methods fail and what can be done about it.

6.1 Contributions

In this chapter, the applicability of SfS on real-world images is assessed. Measures to make SfS workable on a large class of images are discussed. To this end, a sophisticated pipeline using advanced segmentation techniques, local adaptive thresholding, morphological operations, image inpainting using anisotropic diffusion, and advanced SfS methods are employed. A related publication is [100]. In the end of the chapter, ideas on making SfS working for other classes of objects are discussed.

6.2 Shape from Shading is not Enough

Remembering the chess experiment, one might think "SfS seems to work on real-world images". This, however, is not true in general. As an example, consider the input image shown in Figure 6.1. It shows an espresso cup,

Figure 6.1: Photograph of a cup.

acquired by a digital camera with flash. Is it possible to obtain a reasonable reconstruction of this cup using SfS? Applying the Lambertian model by Prados, which is the basis for all recent SfS methods, one obtains the reconstruction shown in Figure 6.2. Certainly, this is not convincing.

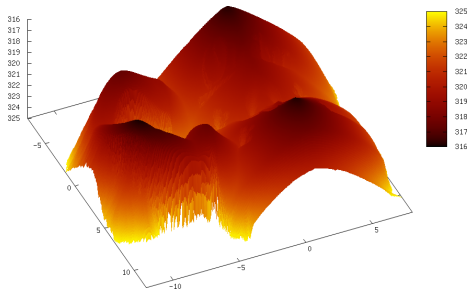

Figure 6.2: Lambertian reconstruction of the unmodified cup image.

Now, one might argue: But there are specular highlights in the scene, why not using a more advanced model? Therefore, we apply the SfS model using the Phong surface model. Unlike the purely Lambertian one, this is not parameter-free (cf. Section 2.5). So one needs to estimate the relation between specular and diffuse light contributions and the tightness of the highlight. Manually optimised, the result shown in Figure 6.3 is obtained. It shows more or less the same systematic errors as the purely Lambertian one, only the outliers get more significant. However, it is clear that at the texture on the cup and the background bad things happen. Dealing with specular highlights certainly improves the reconstruction, but there is much more work to do.

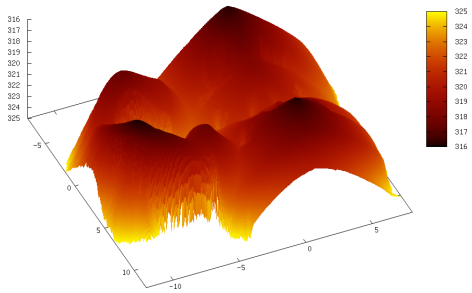

Figure 6.3: Lambertian reconstruction of the unmodified cup image.

In the following, this problem will be tackled for this class of real-world images. In general, these are images of an object of interest in front of some background. The object of interest, however, does not have uniform reflectance properties. Here, the object is textured. This holds for a fairly large class of objects, but certainly not all. For the rest of this chapter, the focus will mainly concentrate on this class of images. At the end of the chapter, some hints on how to deal with other objects will be given as well.

6.3 Problem 1: The Background

The first apparent violation of the model assumptions in the espresso cup image are the different surfaces present in the image. The background clearly has different reflectance properties than the cup itself. In the model, it is assumed that this not to the case. The goal is now to concentrate solely on the foreground object, the cup.

Finding the Region of Interest – Segmentation

In a first step one separates the object of interest from the background. This is necessary since both have incompatible reflectance properties. For this task one uses the active contour model of Chan and Vese [17]. This is a classic level-set-based method that exploits the grey-value difference between object and background.

The Chan-Vese model segments the image domain $\Omega \subset \mathrm{R}^2$ into two regions by minimising the difference between the image intensity $f(\mathbf{x})$: $\Omega \to \mathrm{R}$ and its average value in each region. Additional constraints are imposed on the length of the region boundary C and on the area inside C.

This comes down to minimising the energy

$$\begin{aligned}\mathcal{E}(C, c_1, c_2) = \mu \operatorname{length}(C) + \nu \operatorname{area}(\operatorname{inside}(C)) \\ + \int\limits_{inside(C)} (f - c_1)^2 \mathrm{d}\mathbf{x} + \int\limits_{outside(C)} (f - c_2)^2 \mathrm{d}\mathbf{x},\end{aligned} \tag{6.1}$$

where c_1 and c_2 are the average values of f inside and outside C, and $\mu \geq 0$ and $\nu \geq 0$ are weighting parameters. These weights are important to tune the object detection: A large μ will give a coarse segmentation, while a small μ will detect fine details. As a region-based segmentation model, the Chan-Vese method is fast and robust with respect to initialisation and noise.

In order to further improve the localisation of the object contour, one uses the Chan-Vese result as initialisation for the edge-based geodesic active contour model [16, 41]. The governing evolution equation is given by

$$\begin{aligned}&\partial_t \phi = |\nabla \phi| \operatorname{div}\left(g\left(|\nabla f_\sigma|\right) \tfrac{\nabla \phi}{|\nabla \phi|}\right) && \text{on } \Omega \times [0, \infty), \\ &\phi(\mathbf{x}, 0) = \phi_0(\mathbf{x}) && \text{on } \Omega,\end{aligned} \tag{6.2}$$

where $\phi(\mathbf{x}, t)$ is a level-set function, ϕ_0 a suitable initialisation and $\nabla = (\partial_x, \partial_y)^\top$ is the gradient operator. The edge stopping function g draws the contour towards nearby edges in the presmoothed image f_σ, which is obtained by convolving f with a Gaussian with standard deviation σ. The function $g(s^2)$ is decreasing in s. In this application one chooses the Perona-Malik diffusivity $g_{PM}(s^2) = (1 + s^2/\lambda^2)^{-1}$, where $\lambda > 0$ is some contrast parameter [64]. If the object is bounded by a pronounced edge, the edge-based active contours will generally result in a sharper segmentation than the sole use of the Chan-Vese model.

The Result

It is possible to see the systematic errors both in the Lambertian and Phong case, but the purely Lambertian case is essentially parameter-free, cf. Section 2.5, only Lambertian reflectance is used for the evaluation of this step.

Figure 6.4 shows the obtained segmentation for the cup image. For the segmentation, the intensity of the image was used. Figure 6.5 shows the reconstruction obtained by applying perspective Lambertian SfS only within this mask. Outside the mask, state constraints boundary conditions have been imposed, which are equivalent to Neumann boundary conditions when using upwinding, cf. Section 3.12. Clearly, this improves the reconstruction of the cup. It is has the wrong shape, but on its boundaries, the reconstruction is substantially better.

Figure 6.4: Segmented version of the cup image.

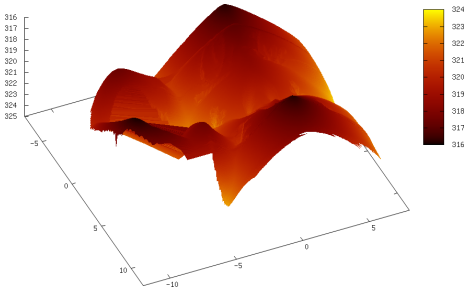

Figure 6.5: Lambertian reconstruction of the segmented image.

6.4 Problem 2: The Texture

Generally, real-world objects do not have a constant albedo. To apply SfS one needs to ensure that the albedo does not vary within the segmented contour. In this approach, the goal is to detect regions of differing albedo and fill in neighbourhood information to obtain homogeneous reflectance properties. In the case of the cup image, the texture on the surface has a different albedo than the rest of the cup. The goal is now to detect this region and deal with the violation of the model assumption of a constant albedo.

In order to identify regions with fluctuating albedo one uses an adaptive thresholding algorithm that works on local windows [78]. Adaptive thresholding is robust with respect to varying illumination conditions within the scene and is widely used, for instance in document analysis. Note that by slightly enlarging the identified regions by morphological erosion it is possible to improve the subsequent interpolation result, preventing artefacts at

the boundaries.

The next step is to interpolate the image in these regions. For this task edge-enhancing anisotropic diffusion (EED) is used [101]. It was shown to perform better for image inpainting and scattered data interpolation than other PDE-based methods [102]. The main idea behind EED is to allow smoothing within homogeneous regions and along image edges, but to reduce smoothing across them. To this end it makes use of a diffusion tensor. In the region that that is to be inpainted steady-state diffusion equation

$$0 = \operatorname{div}\left(g(\nabla u_\sigma \nabla u_\sigma^\top)\nabla u\right), \tag{6.3}$$

with the boundary conditions specified by the surrounding data is solved. Here u_σ is a smoothed version of the evolving image u, obtained by convolving it with a Gaussian of standard deviation σ. The scalar-valued diffusivity g is applied to the eigenvalues of the structure tensor $\nabla u_\sigma \nabla u_\sigma^\top$, while leaving its eigenvectors unchanged. This way, the first eigenvector of the diffusion tensor is parallel to the edge detector ∇u_σ. The desired filter effect comes from the fact that the corresponding eigenvalue is given by $g(|\nabla u_\sigma|^2)$, such that smoothing is reduced at edges, where $|\nabla u_\sigma|$ is large. The second eigenvector is orthogonal to ∇u_σ with corresponding eigenvalue 1. For the diffusivity g one typically chooses the Charbonnier diffusivity $g_C(s^2) = (1 + s^2/\lambda^2)^{-1/2}$, with contrast parameter $\lambda > 0$.

The interpolated image can be seen as an albedo-corrected version of the original image, which now satisfies the assumption of a surface with homogeneous reflectance properties.

The Result

Again, first an investigation of the effect only using a Lambertian SfS model is done, since the systematic errors can be seen there as well, but the method is parameter-free.

First, perform an adaptive thresholding on the image within the cup area, taking a 100×100 window. This gives the inpainting region, which is the black template in Figure 6.6.

After a morphological erosion of this inpainting region in order to enlarge its size, we apply EED with the parameters $\lambda = 2$ and $\sigma = 0.3$ to inpaint the image there. The inpainted image is shown in Figure 6.7. This image can be regarded as a constant albedo version of the original image, within the segmented area. Note that this image still contains specular highlights.

Figure 6.6: Inpainting region obtained by adaptive thresholding.

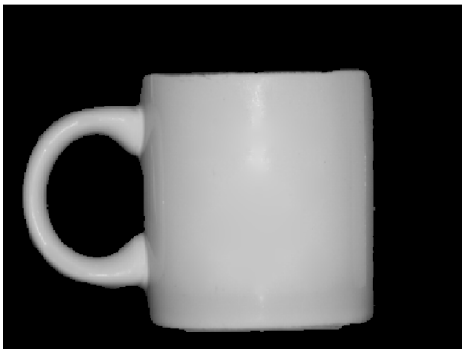

Figure 6.7: Inpainted image of the espresso cup.

Now we reconstruct the surface from the segmented and inpainted data. Figure 6.8 shows the corresponding reconstruction. The shape of the cup obtained by this Lambertian reconstruction looks quite reasonable. However, the cup is estimated much too close to the camera, in particular at specular highlights. Note that the handle, which is pointing slightly towards the background in the original image, is still pulled to the front.

6.5 Dealing with Specular Highlights

Now the only thing left to do is what has been discussed in Chapter 2: dealing with specular highlights by applying the more advanced Phong model. As discussed, we essentially have two parameters now, which has to be fine-tuned manually to obtain the result shown in Figure 6.9. Now the edgy artefacts on the surface are gone, the shape of the cup is recovered quite

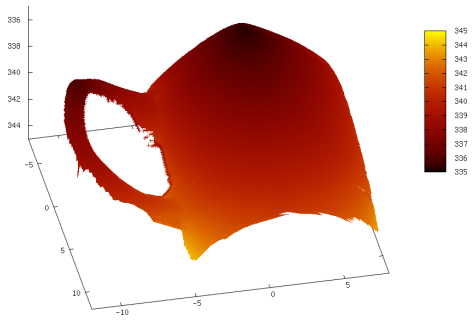

Figure 6.8: Reconstruction of the inpainted cup image using a Lambertian model

well. The most significant improvement can be observed at the handle, which is reconstructed rather good now, and points to the right direction.

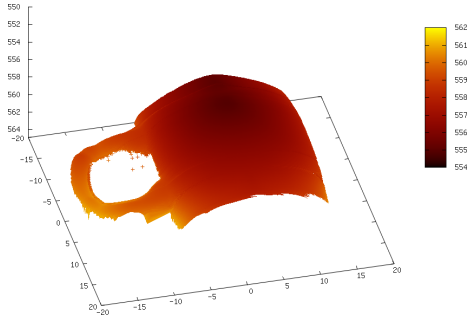

Figure 6.9: Reconstruction of the cup using the Phong model.

Finally Figure 6.10 shows a rendered version of the final reconstruction, with the input image used as texture. The result looks convincing.

6.6 Additional Experiments

In this section, the effectiveness of the method is demonstrated at two additional experiments. In the first experiment, the image of a computer mouse has been taken. Figure 6.11 shows the corresponding input image. As before, the first step consists of segmenting the mouse from the background. To that, the segmentation method introduced in Section 6.3 is used. Since this image has a substantially higher contrast in the hue channel of the HSV

Figure 6.10: Rendered version of the final reconstruction of the espresso cup.

Figure 6.11: Photograph of a computer mouse on a table.

colour space [1] [29] than in the brightness of the image, it is of benefit to use this channel for the segmentation. One obtains the mask shown in Figure 6.12.

The next step is again straightforward. First apply adaptive thresholding to the image to detect the texture. Note that the texture in this case

[1] Of course, this only works since the input image is a colour image from a digital camera.

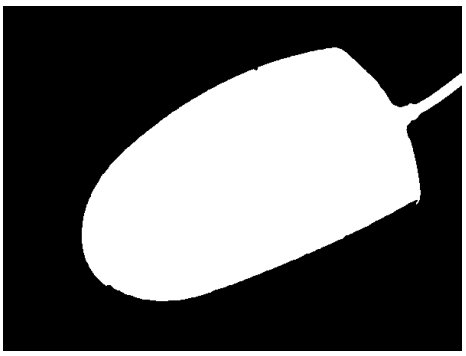

Figure 6.12: Segmentation result for the mouse image.

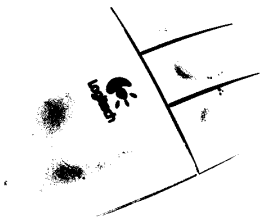

Figure 6.13: Inpainting mask for the mouse image.

not only consists of the logo of the company of the mouse manufacturer, but also the ridges between the buttons are considered as undesired "texture". This is particularly important since in these ridges, shadows occur, which are a violation to the model assumptions. The result of the adaptive thresholding can be inspected in Figure 6.13. After morphologically enlarging the textured region, again, the missing information is interpolated using EED. Figure 6.14 shows the resulting input image.

In the final step, reconstruct the surface. The final result can be found in Figure 6.15.

The second experiment aims at reconstructing the shape of a popular German mathematics handbook [13]. The input image can be found in Figure 6.16. Despite the simplicity of the shape of a book, this is a particularly challenging experiment, since the book contains a lot of texture.

Although reconstruction of the shape in this experiment is quite diffi-

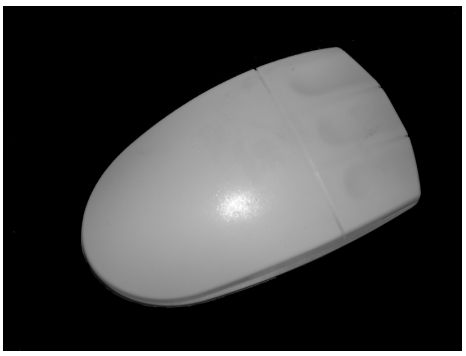

Figure 6.14: Final preprocessed input image for the mouse image.

Figure 6.15: Reconstruction of the computer mouse.

cult, the segmentation is rather easy, in particular if using the hue channel for the segmentation again, one obtains a practically perfect segmentation, shown in Figure 6.17. As usual, the next step consists of finding the texture by adaptive thresholding. The result of this is shown in Figure 6.18. Interpolation with EED yields the result shown in Figure 6.19.

Finally, it is possible to reconstruct the surface of the book as shown in Figure 6.20. Clearly, this reconstruction is significantly better than the unprocessed one. It is not completely perfect, however, the shape of the book can clearly be seen. This reconstruction is a decent estimate of the surface of the book.

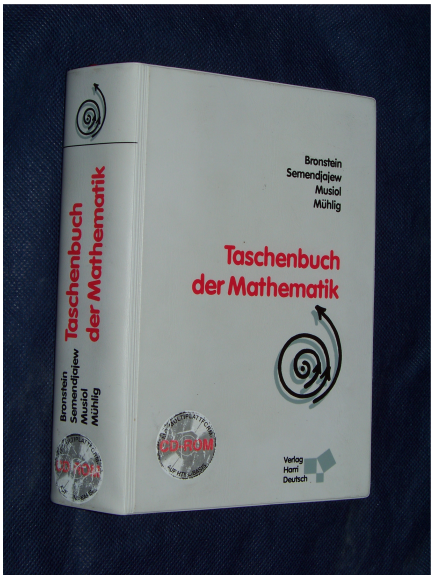

Figure 6.16: Photograph of a book.

6.7 Summary of the Method

In this section, the steps taken to deal with real-world images are summarised. In essence, the method consists of the following steps:

1. *Background segmentation:* The goal of this step is to remove the background of the image from the reconstruction to avoid issued caused by different surfaces in the background. First, use the Chan-Vese segmentation method to obtain a foreground-background segmentation. The advantage of this method is its robustness. However, the resulting segmentation does not precisely fit to the object boundaries. If the input image is a colour image, it might be possible to use the hue channel instead of the image brightness for the segmentation to obtain even better results. Using the segmentation result from the previous step as initialisation, one obtains a more accurate reconstruction of the boundary. Using this segmentation, one obtains a reasonable reconstruction without any background artefacts.

2. *Texture Removal:* In the next step, first determine the texture region by making use of local adaptive thresholding. Using this method, one obtains good estimates for the textured part of the surface. Then, morphologically enlarge this region, remove the grey value information

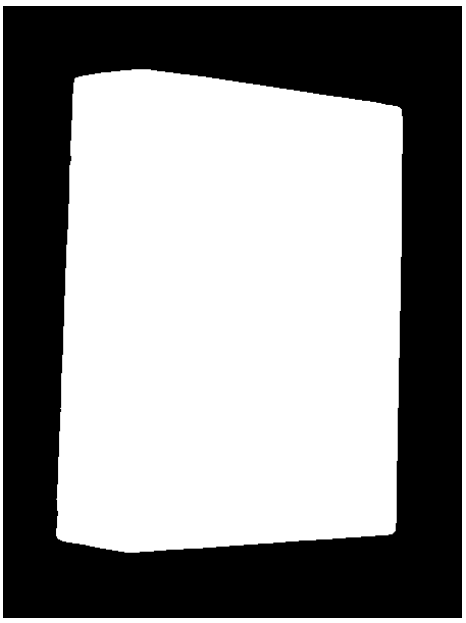

Figure 6.17: Segmentation result for the book image.

within this region from the image and fill in the missing information by inpainting using anisotropic edge-enhancing diffusion.

3. *Surface Reconstruction:* Finally, reconstruct the surface using the model presented in Chapter 2. On the presented test images, this gives very convincing results.

The overall performance of the method is good. The whole preprocessing pipeline only takes a couple of seconds. Even using the iterative SfS method, the overall performance on the espresso cup image sized 306×204 pixels is less than 30 seconds.

6.8 Other Classes of Real-World Images

To conclude this chapter, an interesting point to address is what to do on other real-world images. Certainly the most simple case of a real-world image are the chess input images discussed earlier. These images are not textured, so it is sufficient to just segment the objects of interest from the background. Even textured images like the ones presented in this chapter are possible to reconstruct.

Figure 6.18: Inpainting mask for the book image.

What will happen with surfaces that are globally textured? Or what if there are different surfaces in one scene, which are all to be reconstructed? There exists no SfS framework that can deal with such images, however, it is possible to give ideas how to deal with such scenarios.

Different Objects

Dealing with different object is actually very simple: if one manage to segment these objects from each other, the only thing to do is to supply different parameters to each of the segments and do the reconstruction in each of the segments separately. This sounds trivial, but is essentially the only possible way to deal with such a case. In a purely Lambertian setting, this comes down to estimating the relation between the albedo factors of the different segments.

To automatise this, it might be possible to invent ways to estimate the albedo in different regions. Besides actual measurement of the reflectance properties of objects, which is possible to do if the objects are physically accessible, other information might be used to estimate the albedo. Such information might include, but is not limited to: defocus information, stereo information (if images from other viewpoints are available), photometric

Figure 6.19: Final preprocessed input image for the book image.

Figure 6.20: Reconstruction of the book.

stereo information (if images under other illumination conditions are available). In these cases, it might be possible to roughly estimate the depth somewhere in the segments, and by that estimate the albedo. The simplest way, however, would be the user providing this information. In the case of different specularities, that information has to be determined or provided as well.

Fully Textured Images

Another thing that might happen is that a surface is completely textured. In this case, the remaining information might not be sufficient to inpaint. If the texture can be sharply segmented, the situation is practically the same as in the previous case: In each segment, the reflectance parameters of the respective colour needs to be determined. The easiest way to do this is again by user interaction.

Colour Gradients

Another issue that can arise is that objects do not have a texture that can be sharply segmented. One such example are colour gradients, where an object or a segment of an object has one colour on one side and another colour on the other side, with a smooth transition in between. In this case, further assumptions are necessary, like assuming a linear transition. With such an assumption and the reflectance parameters at several points of the object, it may be possible to interpolate good reflectance parameters over the entire object. For the reconstruction, it is then sufficient to use the right parameters adaptively in every pixel. Estimating such information, however, might be difficult without additional user input.

Chapter 7

Summary and Outlook

The goal of this chapter is to summarise the contributions and findings from this thesis. Furthermore, we will look at the state of SfS up to this point and discuss possible ways to explore in this field.

7.1 Summary

One part of this thesis was to give a survey of the current state-of-the-art of SfS method, both on the model and on the numerical side. In addition to that, we introduced a novel, more sophisticated SfS model which allowed for more accurate reconstructions. We also derived efficient numerical schemes for this model and explored parallelisation techniques for these algorithms. Finally, we looked at real-world images and discussed ways to cope with the challenges presented by them. In this section, we summarise the scientific contributions made in this thesis on models, numerics, parallelisation, and application of SfS.

Model Improvements

On the model side, we

- derived a novel model for SfS. This model was based on the state-of-the-art model by Prados et al. We extended this standard model by introducing specular highlights into the model.
- derived a Hamilton-Jacobi PDE for this model, which presents the basis for all numerical methods introduced in this thesis.

- did an extensive experimental evaluation of the effect of including specular highlights into the model. Even for surfaces with extreme reflectance properties, we managed to obtain decent reconstructions. We verified that using the right reconstruction parameters we can obtain good reconstructions, while purely Lambertian reconstructions were significantly worse even on surfaces with moderate specular highlights.

Numerical Methods

On the numerical side,

- we proposed an iterative method that directly solves the Hamilton-Jacobi equation of the model, in contrast to previous numerical techniques which solved an equivalent Hamilton-Jacobi-Bellman equation using techniques from dynamic programming or semi-Lagrangian methods. The proposed method turned out to be much more efficient than the existing techniques in the literature.
- by doing a numerical scale analysis, we motivated the numerical scheme. By identifying the most dominant terms in the PDE, we could choose a suitable discretisation tailored to our model.
- by using an advanced Gauß-Seidel-like solver along with a fast sweeping technique we managed to further improve the performance of the method.
- we derived a stability criterion on the time step size for this numerical method.
- we analysed the convergence properties of our numerical scheme and one competing numerical scheme to identify their strengths and weaknesses. We proposed a hybrid scheme that combines advantages of different strategies to further improve the performance.
- we introduced a non-iterative fast marching method for the same model. While fast marching methods need Dirichlet data at certain points in the image, we showed that for the presented SfS model, it is possible to obtain this data in a different way, and by that obtain a very fast numerical method which does not rely on any Dirichlet data to be provided by the user.
- we experimentally investigated the differences in the performance between fast marching and the iterative methods on actual test images.

- we experimentally investigated the numerical differences in the final reconstruction between the fast marching method and the iterative method. It turned out that in some cases, fast marching is not only a faster solver, but it can also be more accurate.

Parallelisation

We explored ways to parallelise the very efficient, but difficult to parallelise, fast marching method. We did this in two ways:

- By distributing the singular seed points obtained at initialisation of the FM method to different threads, we were able to distribute the work load to different processors on a multi-core CPU architecture. The main issue in this approach was to handle collisions between the wave fronts of the evolutions of the fast marching method for each thread. With this approach, we managed to outperform other parallelisation approaches for fast marching methods which decomposed the image domain in different parts and handle collisions only at the boundary between these sections. An important feature of SfS for the success of the presented approach was that in most scenes, there are several singular points, and these singular points are not necessarily distributed over the whole image.

- By combining the iterative numerical method we derived with the fast marching method, we succeeded in bringing a fast-marching-like method to a massively parallel architecture. While fast marching methods are not suited for massive parallelisation, we split the image in small blocks, on which we solved the SfS problem iteratively on the GPU. The order in which the blocks are computed and accepted is motivated by the fast marching idea. By this method, we achieve real-time performance on standard test images and in particular very fast reconstruction of very large images compared to purely sequential implementations.

Application on Real-World images

We identified the difficulties for SfS on real-world images, which mainly lie in different reflectance properties for different objects in the scene and different reflectance properties in textured parts of objects. We introduced a framework to preprocess the image in order to achieve a visually convincing results. The framework consisted of

- Segmentation of the object of interest from the background of the image. This segmentation was done using the Chan-Vese method for obtaining a rough segmentation which was then refined by geodesic active contours.

- Identifying textured regions by using local adaptive thresholding.

- Interpolating these regions inpainting by using edge-enhancing anisotropic diffusion. This step can also be understood as an estimation of the correct albedo in these textured region in relation to the general albedo of the object to be reconstructed. The result of this interpolation was a realistic object with a uniform albedo.

- Reconstruction of the interpolated, uniform albedo object using the SfS model introduced in this thesis. Comparisons to a purely Lambertian setting showed that incorporating a specular term clearly improves the accuracy of the reconstruction.

7.2 Outlook

There is still room for some improvement on SfS models. Early SfS techniques contained models of more complex lighting conditions, while modern SfS techniques usually use very simplified lighting models with usually just one single light source located in the optical centre or very close to it. While this assumption clearly simplifies the modelling process, it might be worth to include more than just one light source into the model, and allow to place these light sources at other positions than the projection centre. This might be quite challenging, since as soon as viewing direction and light direction do not coincide any more, shadows may arise. These shadows would need to be incorporated in the model and be estimated to obtain a decent result.

Another aspect on the model side would be the use of other reflectance models. In this thesis, we have seen that using a still rather simple reflectance model, it is possible to significantly improve the reconstruction quality compared to simple Lambertian reflectance. Depending on the surface which should be reconstructed, even better results may be achieved by incorporating other, more complex and realistic surface models. At this point, there is a trade-off between realism and simplicity of the model. The effect on the reconstruction of using a very complex and realistic reflectance model might be rather small to the required effort. Nevertheless, it might be worth to explore this issue in more detail.

A very important point for future SfS research is to explore new fields of application for SfS. So far, the range of actual applications for SfS is somewhat limited. There were approaches to use orthographic SfS on document images acquired by flatbed scanners to obtain an estimate for the distortion of the scanned page, which can in turn be used for correcting this distortion to improve recognition rates of optical character recognition. There was another attempt to use SfS for the reconstruction of small surface patches in endoscopic images. With the efficient numerics we found in this thesis, there might be room for improvement in this field.

We have shown that using efficient numerical methods, it is possible to achieve real-time performance for SfS. This leads to the question whether it is possible to include SfS in an interactive framework. One possible application for such a framework would lie in medical imaging. As one example, endoscopic images fit quite well to existing models, as the light source is attached to the camera and there are no significant other light sources. With some additional effort, mainly in choosing the right reflectance models and parameter estimation, a real-time, interactive framework might be a good and helpful application for SfS.

However, as soon as we embed SfS in such an interactive framework, we have more than one input image. To improve the reconstruction quality, correspondences between subsequent images could be taken into account. In fact, there are several techniques that use correspondences between images to obtain structure information, such as shape from stereo or structure from motion. These methods usually require images to have a lot of texture in order to determine correspondences between the images. On images with sparse texture information, these methods can get into trouble. For SfS, the situation is different, since it works quite well on untextured images, while texture is usually an undesirable feature. An interesting direction for further research would be to find ways for other shape reconstruction techniques to support SfS or vice versa.

Optic flow methods are a class of methods which aim at finding correspondences between subsequent images in an image sequence. While optic flow methods can be used for determining correspondences between subsequent images in a scene, similar correspondences should also arise between the reconstructed surfaces. This information could be used to improve the reconstruction quality. In the other direction, when we are interested in a correct flow field between subsequent images, SfS might be helpful for assisting optic flow methods in untextured areas. In such areas, structure information might be obtained by using SfS. This structure information might be used as an optic flow constraint and thus improve the flow field. In this case, parameter estimation for SfS might not be much of an is-

sue, since the reconstructed surface would not necessarily need to be very accurate to extract the information content of the shading.

Another interesting question is how to determine the shading parameters in a scene. Most SfS methods simply assume that there are uniform reflectance properties in the whole scene and that the camera and lighting conditions are known. Since in a real scene, there frequently is more than one object, this assumption is not applicable in reality. We have seen some approaches to deal with this issue, such as segmentation of objects or interpolation of textured regions. There might be other ways to estimate reflectance properties of objects in a scene. For instance, such techniques might make use of defocus information to get a rough depth estimate which could in turn be used to get an estimate on the surface albedo.

Bibliography

[1] A. Agrawal, R. Raskar, and R. Chellappa. What is the range of surface reconstructions from a gradient field? In A. Leonardis, H. Bischof, and A. Pinz, editors, *Computer Vision – ECCV 2006, Part I*, volume 3951 of *Lecture Notes in Computer Science*, pages 578–591, Berlin, May 2006. Springer.

[2] A. Ahmed and A. Farag. A new formulation for shape from shading for non-Lambertian surfaces. In *Proc. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 17–22. IEEE Computer Society Press, New York, NY, June 2006.

[3] S. Bakshi and Y.-H. Yang. Shape from shading for non-Lambertian surfaces. In *Proc. IEEE International Conference on Image Processing*, volume 2, pages 130–134, Austin, TX, November 1994. IEEE Computer Society Press.

[4] M. Bardi and I. Capuzzo-Dolcetta. *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*. Birkhäuser, Boston, Basel, Berlin, 1997.

[5] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[6] M. Bichsel and A. P. Pentland. A simple algorithm for shape from shading. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 459–465. IEEE Computer Society Press, 1992.

[7] F. Bornemann and P. Deuflhard. Cascadic multigrid methods. In R. Glowinski, J. Periaux, Z. Shi, and O. Widlund, editors, *Domain*

Decomposition Methods in Sciences and Engineering, pages 205–212, Chichester, New York, 1997. John Wiley & Sons.

[8] M. Breuß, E. Cristiani, J.-D. Durou, M. Falcone, and O. Vogel. Numerical algorithms for perspective shape from shading. *Kybernetika*, 46(2):207–225, 2010.

[9] 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*, accepted for publication, 2011.

[10] M. Breuß, E. Cristiani, P. Gwosdek, and O. Vogel. A domain-decomposition-free parallelisation of the fast marching method. Technical Report 250, Department of Mathematics, Saarland University, Saarbrücken, Germany, October 2009.

[11] M. Breuß, O. Vogel, and J. Weickert. Perspective shape from shading for Phong-type non-Lambertian surfaces. Technical Report 216, Department of Mathematics, Saarland University, Saarbrücken, Germany, August 2008.

[12] M. Breuß, O. Vogel, and J. Weickert. Efficient numerical techniques for perspective shape from shading. In *Proc. of Algoritmy*, pages 11–20, Podbanske, Slovakia, March 2009.

[13] I. Bronstein, K. Semendjajew, G. Musiol, and H. Mühlig. *Taschenbuch der Mathematik*. Verlag Harry Deutsch, Thun and Frankfurt am Main, 5th edition edition, 2001.

[14] M. J. Brooks and B. K. P. Horn. Shape and source from shading. In *Proceedings of the International Joint Conference in Artificial Intelligence*, pages 932–936, Los Angeles, CA, August 1985. MIT Press.

[15] A. M. Bruckstein. On shape from shading. *Computer Vision, Graphics, and Image Processing*, 44:139–154, 1988.

[16] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22:61–79, 1997.

[17] T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, February 2001.

[18] R. Courant, K. Friedrichs, and H. Lewy. über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, 100:32–74, 1928.

[19] F. Courteille, A. Crouzil, J.-D. Durou, and P. Gurdjos. Towards shape from shading under realistic photographic conditions. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 2, pages 277–280, Cambridge, UK, August 2004.

[20] 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 Mathematical Society*, 27(1):1–67, 1992.

[21] E. Cristiani. *Fast marching and semi-Lagrangian methods for Hamilton-Jacobi equations with applications*. PhD thesis, University of Rome, Rome, Italy, 2007.

[22] E. Cristiani, M. Falcone, and A. Seghini. Some remarks on perspective shape-from-shading models. In F. Sgallari, F. Murli, and N. Paragios, editors, *Scale Space and Variational Methods in Computer Vision*, volume 4485 of *Lecture Notes in Computer Science*, pages 276–287. Springer, Berlin, May-June 2007.

[23] P. Dupuis and J. Oliensis. An optimal control formulation and related numerical methods for a problem in shape reconstruction. *Annals of Applied Probability*, 4(2):287–346, 1994.

[24] P. Dupuis and J. Oliensis. Shape from shading: Provably convergent algorithms and uniqueness results. In J.-O. Eklundh, editor, *Computer Vision – ECCV '94, Part II*, volume 801 of *Lecture Notes in Computer Science*, pages 259–268, Berlin, 1994. Springer.

[25] J.-D. Durou. Shape from shading web site. http://www.irit.fr/sfs/sfs.html.

[26] J.-D. Durou, M. Falcone, and M. Sagona. A survey of numerical methods for shape from shading. Technical Report 2004-2-R, Institut de Recherche en Informatique de Toulouse, Toulouse, France, January 2004.

[27] M. Falcone, M. Sagona, and A. Seghini. A global algorithm for the shape-from-shading model with "black shadows". In F. Brezzi, A. Buffa, S. Corsaro, and A. Murli, editors, *Numerical Mathematics and Advanced Applications - ENUMATH 2001*, pages 503–512. Springer-Verlag, 2003.

[28] 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, July 1988.

[29] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Addison–Wesley, Reading, second edition, 2002.

[30] J. J. Helmsen, E. G. Puckett, P. Colella, and M. Dorr. Two new methods for simulating photolithography development in 3d. In *Optical Microlithography IX*, pages 253–261, Santa Clara, CA, USA, March 1996. SPIE.

[31] M. Herrmann. A domain decomposition parallelization of the Fast Marching Method. Annual research briefs, Center for Turbulence Research, Stanford, CA, 2003.

[32] B. K. P. Horn. *Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View.* PhD thesis, Department of Electrical Engineering, MIT, Cambridge, MA, 1970.

[33] B. K. P. Horn. Obtaining shape from shading information. In P. H. Winston, editor, *The Psychology of Computer Vision*, chapter 4, pages 115–155. McGraw-Hill, New York, NY, 1975.

[34] B. K. P. Horn. Understanding image intensities. *Artificial Intelligence*, 8:201–231, 1977.

[35] B. K. P. Horn. Height and gradient from shading. *International Journal of Computer Vision*, 5:37–75, 1990.

[36] B. K. P. Horn and M. J. Brooks. Shape and source from shading. In *International Joint Conference on Artificial Intelligence*, pages 932–936, 1985.

[37] B. K. P. Horn and M. J. Brooks. The variational approach to shape from shading. *Computer Vision Graphics and Image Processing*, 33:174–208, 1986.

[38] B. K. P. Horn and M. J. Brooks. *Shape from Shading.* Artificial Intelligence Series. MIT Press, 1989.

[39] K. Ikeuchi and B. K. P. Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17:141–185, 1981.

[40] W. Jeong and R. Whitaker. A fast iterative method for eikonal equations. *SIAM Journal on Scientific Computing*, 30(5):2512–2534, 2008.

[41] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Conformal curvature flows: from phase transitions to active vision. *Archive for Rational Mechanics and Analysis*, 134:275–301, 1996.

[42] R. Kimmel and A. M. Bruckstein. Shape from shading via level sets. CIS Report 9209, Technion, Israel, June 1992.

[43] R. Kimmel and A. M. Bruckstein. Global shape from shading. *Computer Vision and Image Understanding*, 62(3):360–369, 1995.

[44] 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(2):47–58, 1995.

[45] R. Kimmel and J. A. Sethian. Optimal algorithm for shape from shading and path planning. *Journal of Mathematical Imaging and Vision*, 14:237–244, 2001.

[46] 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.

[47] D. E. Kirk. *Optimal Control Theory: An Introduction*. Prentice Hall, Englewood Cliffs, NJ, 1970.

[48] Y. G. Leclerc and A. F. Bobick. The direct computation of height from shading. In *Proc. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 552–558, Lahaina, HI, June 1991. IEEE Computer Society Press.

[49] C.-H. Lee and A. Rosenfeld. Improved methods of estimating shape from shading using the light source coordinate system. *Artificial Intelligence*, 26(2):125–143, 1985.

[50] K. Lee and C.-C. J. Kuo. Shape from shading with a generalized reflectance map model. *Computer Vision and Image Understanding*, 67(2):143–160, August 1997.

[51] K. M. Lee and C.-C. J. Kuo. Shape from shading with a linear triangular element surface model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(8):815–822, 1993.

[52] J. Malik and D. E. Maydan. Recovering three-dimensional shape from a single image of curved objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):555–566, 1989.

[53] A. Meister. *Asymptotic Expansions and Numerical Methods in Computational Fluid Dynamics*, pages 63–137. Narosa Publishing House, New Delhi, 2003.

[54] NVIDIA Corporation. *NVIDIA CUDA Programming Guide*, 3rd edition, February 2010. `http://developer.download.nvidia.com/compute/cuda/3_0/toolkit/docs/NVIDIA_CUDA_ProgrammingGuide.pdf`, Retrieved 10-06-09.

[55] T. Okatani and K. Deguchi. Reconstructing shape from shading with a point light source at the projection center: Shape reconstruction from an endoscope image. In *Proc. 1996 International Conference on Pattern Recognition*, pages 830–834, Vienna, Austria, August 1996.

[56] J. Oliensis. Shape from shading as a partially well-constrained problem. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 54(2):163–183, 1991.

[57] J. Oliensis. Uniqueness in shape from shading. *International Journal of Computer Vision*, 6(2):75–104, 1991.

[58] J. Oliensis and P. Dupuis. Direct method for reconstructing shape from shading. In *Proc. 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 453–458, Champaign, IL, June 1992. IEEE Computer Society Press.

[59] J. Oliensis and P. Dupuis. Direct method for reconstructing shape from shading. In L. B. Wolff, S. A. Shafer, and G. Healey, editors, *Physics-based Vision: Principles and Practice*, pages 17–28. Jones and Bartlett Publishers, 1992.

[60] J. Oliensis and P. Dupuis. A global algorithm for shape from shading. In *Proceedings of the Fourth International Conference on Computer Vision*, pages 692–701, Berlin, Germany, 1993.

[61] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(2):12–49, 1988.

[62] M. A. Penna. A shape from shading analysis for a single perspective image of a polyhedron. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):545–554, 1989.

[63] A. P. Pentland. Shape information from shading: A theory about human perception. In *Second IEEE International Conference on Computer Vision (ICCV)*, pages 404–413, Tarpon Springs, FL, 1988. IEEE Computer Society Press.

[64] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, 1990.

[65] B. T. Phong. Illumination for computer-generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.

[66] B. T. Phong. *Illumination of Computer-Generated Images*. PhD thesis, Department of Computer Science, University of Utah, July 1975.

[67] E. Prados. *Application of the theory of the viscosity solutions to the Shape from Shading problem*. PhD thesis, University of Nice Sophia-Antipolis, October 2004.

[68] E. Prados, F. Camilli, and O. Faugeras. A unifying and rigorous shape from shading method adapted to realistic data and applications. *Journal of Mathematical Imaging and Vision*, 25(3):307–328, 2006.

[69] E. Prados, F. Camilli, and O. Faugeras. A viscosity solution method for shape-from-shading without image boundary data. *Mathematical Modelling and Numerical Analysis (M2AN)*, 40(2):393–412, 2006.

[70] E. Prados and O. Faugeras. A mathematical and algorithmic study of the Lambertian sfs problem for orthographic and pinhole cameras. Technical Report RR-5005, INRIA, november 2003.

[71] E. Prados and O. Faugeras. Perspective shape from shading and viscosity solutions. In *Proc Ninth International Conference on Computer Vision*, volume 2, pages 826–831, Nice, France, October 2003. IEEE Computer Society Press.

[72] E. Prados and O. Faugeras. Unifying approaches and removing unrealistic assumptions in shape from shading: Mathematics can help. In T. Pajdla and J. Matas, editors, *Computer Vision – ECCV 2004,*

Part IV, volume 3024 of *Lecture Notes in Computer Science*, pages 141–154, Berlin, 2004. Springer.

[73] E. Prados and O. Faugeras. A generic and provably convergent shape-from-shading method for orthographic and pinhole cameras. *International Journal of Computer Vision*, 65(1-2):97–125, 2005.

[74] E. Prados and O. Faugeras. Shape from shading: A well-posed problem? In *Proc. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 870–877, San Diego, CA, June 2005. IEEE Computer Society Press.

[75] E. Prados, O. Faugeras, and E. Rouy. Shape from shading and viscosity solutions. In *European Conference on Computer Vision (ECCV)*, volume 2351 of *Lecture Notes in Computer Science*, pages 790–804. Springer, Berlin, 2002.

[76] E. Prados and S. Soatto. Fast marching method for generic shape from shading. In N. Paragios, O. Faugeras, T. Chan, and C. Schnörr, editors, *Variational, Geometric, and Level Set Methods in Computer Vision*, volume 3752 of *Lecture Notes in Computer Science*, pages 320–331. Springer, Berlin, 2005.

[77] E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM Journal of Numerical Analysis*, 29(3):867–884, 1992.

[78] J. Sauvola and M. Pietikainen. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, 2000.

[79] 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.

[80] J. A. Sethian. Fast marching level set methods for three-dimensional photolithography development. In *Optical Microlithography IX*, pages 262–272, Santa Clara, CA, USA, March 1996. SPIE.

[81] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, Cambridge, UK, second edition, 1999. Paperback edition.

[82] I. Shimshoni, R. Kimmel, and A. M. Bruckstein. Global shape from shading. *Computer Vision and Image Understanding*, 64(1):188–189, 1995.

[83] G. Smith. The recovery of surface orientation from image irradiance. In *DARPA Image Understanding Workshop*, Palo Alto, CA, September 1982.

[84] P. Soravia. Optimal control with discontinuous running cost: eikonal equation and shape-from-shading. In *Proceedings of the 39th IEEE Conference on Decision and Control (volume I)*, pages 79–84, Sydney, Australia, December 2000.

[85] K. G. Suffern. *Ray tracing from the ground up*. A. K. Peters, Wellesley, MA, USA, 2007.

[86] A. Tankus, N. Sochen, and Y. Yeshurun. A new perspective [on] shape-from-shading. In *Proc. Ninth International Conference on Computer Vision*, volume 2, pages 862–869, Nice, France, October 2003. IEEE Computer Society Press.

[87] A. Tankus, N. Sochen, and Y. Yeshurun. Perspective shape-from-shading by fast marching. In *Proc. 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 43–49, Washington, DC, June-July 2004. IEEE Computer Society Press.

[88] A. Tankus, N. Sochen, and Y. Yeshurun. Reconstruction of medical images by perspective shape-from-shading. In *Proc. 2004 International Conference on Pattern Recognition*, volume 3, pages 778–781, Cambridge, UK, August 2004.

[89] A. Tankus, N. Sochen, and Y. Yeshurun. Shape-from-shading by iterative fast marching for vertical and oblique light sources. In R. R. Klette, Kozera, J. Weickert, and L. Noakes, editors, *Geometric Properties from Incomplete Data*, volume 1, pages 237–285. Springer, 2005.

[90] A. Tankus, N. Sochen, and Y. Yeshurun. Shape-from-shading under perspective projection. *International Journal of Computer Vision*, 63(1):21–43, June 2005.

[91] P.-S. Tsai and M. Shah. Shape from shading using linear approximation. *Image and Vision Computing Journal*, 12(8):487–498, 1994.

[92] J. N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538, September 1995.

[93] O. Vega and Y. Yang. Shading logic: A heuristic approach to recover shape from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):592–597, 1993.

[94] O. Vogel. Variational shape from shading: Direct depth recovery and inhomogeneous regularisers. Diploma Thesis, 2006.

[95] O. Vogel, M. Breuß, T. Leichtweis, and J. Weickert. Fast shape from shading for Phong-type surfaces. In X.-C. Tai, K. Mørken, M. Lysaker, and K.-A. Lie, editors, *Scale Space and Variational Methods in Computer Vision*, volume 5567 of *Lecture Notes in Computer Science*, pages 733–744. Springer, Berlin, 2009.

[96] O. Vogel, M. Breuß, and J. Weickert. A direct numerical approach to perspective shape-from-shading. In H. Lensch, B. Rosenhahn, H.-P. Seidel, P. Slusallek, and J. Weickert, editors, *Vision, Modeling, and Visualization*, pages 91–100, Saarbrücken, Germany, November 2007.

[97] O. Vogel, M. Breuß, and J. Weickert. Perspective shape from shading with non-Lambertian reflectance. In G. Rigoll, editor, *Pattern Recognition*, volume 5096 of *Lecture Notes in Computer Science*, pages 517–526. Springer, Berlin, June 2008.

[98] O. Vogel, A. Bruhn, J. Weickert, and S. Didas. Direct shape-from-shading with adaptive higher order regularisation. In F. Sgallari, F. Murli, and N. Paragios, editors, *Scale Space and Variational Methods in Computer Vision*, volume 4485 of *Lecture Notes in Computer Science*, pages 871–882. Springer, Berlin, May-June 2007.

[99] O. Vogel and E. Cristiani. Numerical schemes for advanced reflectance models for shape from shading. In *Proc. IEEE International Conference on Image Processing*, Brussels, Belgium, 2011, accepted for publication. IEEE Computer Society Press.

[100] O. Vogel, L. Valgaerts, M. Breuß, and J. Weickert. Making shape from shading work for real-world images. In J. Denzler, G. Notni, and H. Süße, editors, *Pattern Recognition*, volume 5748 of *Lecture Notes in Computer Science*, pages 191–200. Springer, Berlin, 2009.

[101] J. Weickert. Theoretical foundations of anisotropic diffusion in image processing. *Computing Supplement*, 11:221–236, 1996.

[102] J. Weickert and M. Welk. Tensor field interpolation with PDEs. In J. Weickert and H. Hagen, editors, *Visualization and Processing of Tensor Fields*, pages 315–325. Springer, Berlin, 2006.

[103] R. L. Wildey. Radarclinometry for the venus radar mapper. *Photogrammetric Engineering and Remote Sensing*, 52(41-50), January 1986.

[104] L. Yatziv, A. Bartesaghi, and G. Sapiro. O(n) implementation of the fast marching algorithm. *Journal of Computational Physics*, 212(2):393–399, 2006.

[105] 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.

[106] Y.-T. Zhang, H. Zhao, and J. Qian. High order fast sweeping methods for static Hamilton-Jacobi equations. *Journal of Scientific Computing*, 29(1):25–56, 2006.

[107] H. Zhao. A fast sweeping method for eikonal equations. *Mathematics of Computation*, 74(250):603–627, 2004.

[108] Q. Zheng and R. Chellappa. Estimation of illuminant direction, albedo, and shape from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):680–702, 1991.

Own Publications

Journal Papers

1. M. Breuß, E. Cristiani, J.-D. Durou, M. Falcone, and O. Vogel. Numerical algorithms for perspective shape from shading. *Kybernetika*, 46(2):207–225, 2010

2. M. Welk, M. Breuß, and O. Vogel. Morphological amoebas are self-snakes. *Journal of Mathematical Imaging and Vision*, to appear, 2010.

3. 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*, accepted for publication, 2011

Technical Reports

4. M. Breuß, O. Vogel, and J. Weickert. Perspective shape from shading for Phong-type non-Lambertian surfaces. Technical Report 216, Department of Mathematics, Saarland University, Saarbrücken, Germany, August 2008

5. M. Breuß, E. Cristiani, P. Gwosdek, and O. Vogel. A domain-decomposition-free parallelisation of the fast marching method. Technical Report 250, Department of Mathematics, Saarland University, Saarbrücken, Germany, October 2009

6. M. Breuß, O. Vogel, and K. Hagenburg. Newton interpolation with extremely high degrees by Leja ordering and fast Leja points. Tech-

nical Report 273, Department of Mathematics, Saarland University, Saarbrücken, Germany, September 2010.

Conference Papers

7. O. Vogel, A. Bruhn, J. Weickert, and S. Didas. Direct shape-from-shading with adaptive higher order regularisation. In F. Sgallari, F. Murli, and N. Paragios, editors, *Scale Space and Variational Methods in Computer Vision*, volume 4485 of *Lecture Notes in Computer Science*, pages 871–882. Springer, Berlin, May-June 2007

8. O. Vogel, M. Breuß, and J. Weickert. A direct numerical approach to perspective shape-from-shading. In H. Lensch, B. Rosenhahn, H.-P. Seidel, P. Slusallek, and J. Weickert, editors, *Vision, Modeling, and Visualization*, pages 91–100, Saarbrücken, Germany, November 2007

9. O. Vogel, M. Breuß, and J. Weickert. Perspective shape from shading with non-Lambertian reflectance. In G. Rigoll, editor, *Pattern Recognition*, volume 5096 of *Lecture Notes in Computer Science*, pages 517–526. Springer, Berlin, June 2008

10. M. Breuß, O. Vogel, and J. Weickert. Efficient numerical techniques for perspective shape from shading. In *Proc. of Algoritmy*, pages 11–20, Podbanske, Slovakia, March 2009

11. O. Vogel, M. Breuß, T. Leichtweis, and J. Weickert. Fast shape from shading for Phong-type surfaces. In X.-C. Tai, K. Mørken, M. Lysaker, and K.-A. Lie, editors, *Scale Space and Variational Methods in Computer Vision*, volume 5567 of *Lecture Notes in Computer Science*, pages 733–744. Springer, Berlin, 2009

12. O. Vogel, L. Valgaerts, M. Breuß, and J. Weickert. Making shape from shading work for real-world images. In J. Denzler, G. Notni, and H. Süße, editors, *Pattern Recognition*, volume 5748 of *Lecture Notes in Computer Science*, pages 191–200. Springer, Berlin, 2009

13. M. Welk, M. Breuß, and O. Vogel. Differential equations for morphological amoebas. In M. Wilkinson and J. Roerdink, editors, *Mathematical Morphology and Its Application to Signal and Image Processing*, volume 5720 of *Lecture Notes in Computer Science*, pages 104–114. Springer, Berlin, 2009.

14. K. Hagenburg, M. Breuß, O. Vogel, J. Weickert, and M. Welk. A lattice Boltzmann model for rotationally invariant dithering. In G. Bebis, R. Boyle, B. Parvin, D. Koracin, Y. Kuno, J. Wang, R. Pajarola, P. Lindstrom, A. Hinkenjann, M. L. Encarnaco, C. T. Silva, and D. Coming, editors, *Advances in Visual Computing*, volume 5876 of *Lecture Notes in Computer Science*, pages 949–959. Springer, Berlin, 2009.

- 15. O. Vogel and E. Cristiani. Numerical schemes for advanced reflectance models for shape from shading. In *Proc. IEEE International Conference on Image Processing*, Brussels, Belgium, 2011, accepted for publication. IEEE Computer Society Press

Theses

16. O. Vogel. Variational shape from shading: Direct depth recovery and inhomogeneous regularisers. Diploma Thesis, 2006.

17. O. Vogel. Osmosis in image processing. Fortgeschrittenenpraktikum, Saarland University, December 2006.