Proceedings of ALGORITMY 2009
pp. 11–20

# EFFICIENT NUMERICAL TECHNIQUES FOR PERSPECTIVE SHAPE FROM SHADING

MICHAEL BREUSS, OLIVER VOGEL AND JOACHIM WEICKERT *

**Abstract.** The shape-from-shading (SfS) problem is a classic problem in computer vision. The task in SfS is to compute on the basis of the shading variation in a given 2-D image the 3-D depth of the depicted scene. The corresponding mathematical model eventually leads to a boundary value problem for a Hamilton-Jacobi equation. In this paper we evaluate and compare suitable numerical methods. We begin with a brief discussion of four state-of-the-art-approaches in this field. Then we give an extensive numerical comparison, thus evaluating recent improvements in this area. In the course of doing this, we introduce efficient variations of existing schemes. By this systematic investigation, we complement and extend previous works on the numerical side. The paper is finished by a conclusion.

**Key words.** hyperbolic partial differential equation, Hamilton-Jacobi-Bellman equation, Hamilton-Jacobi equation, finite difference method, shape from shading

**AMS subject classifications.** 35L60, 65N06, 65N12, 68U10

## 1. Introduction.

By taking a photograph, for instance with a digital camera, the 3-D world is mapped to a 2-D image. The shape-from-shading (SfS) problem aims to answer the question if it is possible to recover from exactly one given grey value image the 3-D depth of depicted objects by making use of the shading variation. It is a classic problem in computer vision with many potential applications; see e.g. [6, 7, 18] and the references therein for an overview.

**Basic modelling issues.** A key ingredient of mathematical models for such computer vision problems is the *camera model*, i.e. the mathematical representation of the projection performed when mapping the 3-D world to 2-D images by the camera; see for instance [6] for a discussion of this topic. In early SfS models, the camera is assumed to perform an *orthographic projection* of the scene of interest. Concerning this type of models, let us especially mention the pioneering work of Horn [5] who was also the first to model the SfS task using a *partial differential equation* (PDE). However, orthographic models are notorious for their ill-posedness [18].

Only a few years ago, this model assumption has been substituted by employing a *perspective projection* [11, 13]. Especially along with modelling the light originating from a point light source and the consideration of a so-called light attenuation term, the perspective approach eventually yields a well-posed problem [3, 9, 10]. Beside these important developments, also the works [13, 14, 15] consider perspective SfS but do not employ the light attenuation term, so that we do not consider the corresponding methods in this paper.

**Modelling the surface.** A further important modelling issue is concerned with the *reflectance properties* of objects depicted in the given image. In the works discussed up to now, so-called *Lambertian surfaces* are considered. Such surfaces describe diffuse light reflection: The light intensity of some point on an object surface perceived by the observer depends on the angle between light source direction and surface normal

* Mathematical Image Analysis Group, Faculty of Mathematics and Computer Science, Saarland University, Campus E11, 66041 Saarbrücken, Germany
(email: {breuss,vogel,weickert}@mia.uni-saarland.de).

in this point. The advantage of employing Lambertian surfaces is that the model is theoretically relatively easy to access, simplifying especially the modelling process. However, optical real-world phenomena like e.g. specular light reflections are not taken into account by it. Recently, the perspective Lambertian SfS model has been extended [17] to include surfaces which can be described by the Phong reflectance model well-known in computer graphics [4, 8].

**Algorithms.** On the algorithmic side, the first important developments for perspective Lambertian models with light attenuation term are based on a control-theoretic formulation and employ the dynamic programming principle [9]. Then, in [3], a semi-Lagrangian method was developped based on the same model discretising the domain of the arising optimal control problem. The third numerical approach of importance was introduced in [16], where it was suggested to use the Hamilton-Jacobi equation corresponding to the optimal control problem. The question arises which of the resulting numerical approaches is the best one, as this is an important criterion for applications and the discretisation of model extensions.

**Our contribution.** Making use of numerical experiments, we give an in-depth comparison of the approaches for perspective SfS incorporating light attenuation detailed in [3, 9, 16]. Thereby, we introduce improved variations of the two algorithms based on the optimal control approach [3, 9]. Moreover, we investigate the influence of numerical parameter choices on the efficiency of the method based directly on the Hamilton-Jacobi PDE. We complement this comparative study by considering the Phong surface model in one experimental setting in order to gain an insight on the numerical performance trade-off required for incorporating model improvements.

**Paper organisation.** In Section 2, we review the considered approaches to the SfS model in use, and we briefly discuss the arising numerical issues. In Section 3, we are concerned with the comparative study of numerical methods. The paper is finished by a conclusion in Section 4.

**2. Perspective SfS with light attenuation.** Let $x \in \mathbb{R}^2$ be in the image domain $\Omega$, where $\Omega$ is an open set. Furthermore:

- $u := u(x)$ denotes the sought depth map,
- $I := I(x) = \frac{E(x)}{\sigma}$ is the normalised brightness $E(x)$ of the given grey-value image, where $\sigma$ depends on the albedo of the surface, i.e., it depends on the extent to which it diffusely reflects light as well as on the brightness of the light source,
- $\mathsf{f}$ is the focal length denoting the distance between the optical center of the camera and the 2-D plane to which the scene of interest is mapped.

Note, that $u > 0$ holds as the depicted scene is in front of the camera, and that the distance is measured in terms of multiples of $\mathsf{f}$.

We describe due to space limitations only the basics of the considered approaches. For details, we refer the interested reader to the cited works.

**2.1. The optimal control approach.** In [9, 10] it was used that $v = \ln(u)$ is the solution of the Hamilton-Jacobi-Bellman equation

$$-e^{-2v(x,y)} + \sup_{a \in B[0,1]} \{-b(x,y,a) \cdot \nabla v(x,y) - l(x,y,a)\} = 0 \,, \tag{2.1}$$

where $(x,y) \in \Omega$ and where $B[0,1]$ denotes the closed unit ball in $\mathbb{R}^2$. The other expressions are defined as follows. Let $|.|$ denote the Euclidean vector norm, then

$$b(x,y,a) = -JR^T DRa \,, \quad l(x,y,a) = -I(x,y)\mathsf{f}^2\sqrt{1-|a|^2} \,, \tag{2.2}$$

with the matrices

$$R := \frac{1}{\sqrt{x^2+y^2}} \begin{pmatrix} y & -x \\ x & y \end{pmatrix}, \quad D := \begin{pmatrix} \mathsf{f} & 0 \\ 0 & \sqrt{\mathsf{f}^2+x^2+y^2} \end{pmatrix} \tag{2.3}$$

and

$$J := J(x,y) = I(x,y)\mathsf{f}\sqrt{\mathsf{f}^2+x^2+y^2}\,. \tag{2.4}$$

For the numerical solution of the optimal control problem given by (2.1)-(2.4), one has *(i)* to discretise the occuring partial derivatives of $v$ in (2.1), and *(ii)* to find an optimal control $a \in R^2$.

In the already cited works the first-order derivatives part of $\nabla v$ are discretised by using appropriate upwind discretisations as described in [12], leading to a case distinction w.r.t. the upwind directions. It turns out that in order to solve the optimal control problem given by (2.1)-(2.4), we have to search for an optimal $a$ in the entire unit ball. This is done in [9, 10] by computing the analytical solution for $a$ over $B[0,1]$, which is a somewhat standard yet quite complicated procedure.

Having established the discrete formulation, the equation is solved pointwise in an iterative way using an artificial time stepping technique, which is practically a semi-implicit scheme where the implicitness stems from the treatment of the source term. For the arising fixed point iteration, Newton's method is used.

**2.2. The semi-Lagrangian approach.** Developped in [3], this is also an optimal control technique, where equation (2.1) is solved by means of a iterative fixed point procedure. Introducing an artificial time-dependency into the process, indicated by a lower index $h$, one may obtain the equation

$$-v_h(x,y) - \inf_{a\in B[0,1]} \left\{ v_h\left((x,y)+\tau b(x,y,a)\right) + hl(x,y,a)\right\} + \tau e^{-2v_\tau(x,y)} = 0\,. \tag{2.5}$$

This equation can be solved iteratively by employing a sequence $v_\tau^{(k)}$, $k = 0,1,\ldots$, using Newton's method as in the previously described setting until a steady state of $v_h$ is reached. Note that the upwinding strategy previously used is also encoded within the formula (2.5), so that essentially the same upwind method is used here as in Sec. 2.1.

An important technical aspect in optimal control approaches such as in (2.5) is realised within the considered semi-Lagrangian technique. Again the optimal control $a$ has to be sought within the entire unit ball in $\mathbb{R}^2$. This, however, can be done via use of a sampling procedure. In [3], this is done by making use of 8 directions with 3 points in each direction additionally to the origin.

**2.3. The direct Hamilton-Jacobi approach.** Instead of computing the Hamilton-Jacobi-Bellman equation of the optimal control problem, one may stick to solving the corresponding Hamilton-Jacobi PDE

$$\frac{1}{Q} I(x,y)\mathsf{f}^2 W - \exp\left(-2v(x,y)\right) = 0\,, \qquad \text{where} \tag{2.6}$$

$$W := \sqrt{\mathsf{f}^2|\nabla v|^2 + (\nabla v\cdot x)^2 + Q^2}\,, \quad Q := \frac{\mathsf{f}}{\sqrt{x^2+y^2+\mathsf{f}^2}}\,. \tag{2.7}$$

In [16], the first-order derivatives are approximated by upwind differences as in [12]. Also, an artificial time technique is proposed in the latter work yielding an iterative method whose steady-state solution is identical to the numerical solution of (2.7), in a very similar fashion as with the previous two algorithms.

Let us note that here, due to the source term $e^{-2v}$, fixed point iterations, which make use of Newton's method are applied pointwise, too. However, due to the direct Hamilton-Jacobi approach no optimal control needs to be determined.

**2.4. Extension to Phong-type surfaces.** The latter model was extended in [17] to surfaces described by the Phong-model [8], which is used frequently in computer graphics. This model extension results in the Hamilton-Jacobi equation

$$\frac{1}{Q}\left(I(x,y)-k_aI_a\right)\mathrm{f}^2W-k_dI_de^{-2v}-\frac{Wk_sI_s}{Q}e^{-2v}\left(\frac{2Q^2}{W^2}-1\right)^{\alpha}=0 \tag{2.8}$$

with $W$, $Q$ as in (2.7). In this case, the underlying brightness relation reads as

$$I(x,y)=k_aI_a+\sum_{\text{light sources}}\frac{1}{r^2}\left(k_dI_d(x,y)\cos\phi+k_sI_s(x,y)(\cos\theta)^{\alpha}\right), \tag{2.9}$$

where $I_a$, $I_d$, and $I_s$ are the intensities of the ambient, diffuse, and specular components of 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.

Let us note that the ambient light term represents light present everywhere in a given scene. The intensity of diffusely reflected light in each direction is proportional to the cosine of the angle $\phi$ between surface normal and light source direction. The amount of specular light reflected towards the viewer is proportional to $(\cos\theta)^{\alpha}$, where $\theta$ is the angle between the ideal (mirror) reflection direction of the incoming light and the viewer direction, $\alpha$ being a constant modelling the roughness of the material. Employing no ambient and no specular component, we retrieve the Lambertian reflection as a special case in this model.

Concerning the numerical side, let us note that the new third term contributing within (2.8) in comparison to (2.6) involves again sources and first-order derivatives of $v$. In [17], the same upwind discretisation of $\nabla v$ is used in these new terms as well as in the remaining terms already discussed below (2.6), while the new source term is evaluated explicitly.

**2.5. Advanced discretisation issues.** As a building block for the discretisation of spatial derivatives, the stable upwind-type discretisation of Rouy and Tourin [12] is used in the described works. Let $h_x$ and $h_y$ be the spatial mesh widths in $x$- and $y$-direction, respectively. Denoting then by $v_{i,j}$ the value of $v$ at the mesh point $(ih_x,jh_y)^T$, the upwind differencing formulae referred to read as

$$v_x(ih_x,jh_y)\approx\min\left(0,\frac{v_{i+1,j}-v_{i,j}}{h_x},\frac{v_{i-1,j}-v_{i,j}}{h_x}\right), \tag{2.10}$$

$$v_y(ih_x,jh_y)\approx\min\left(0,\frac{v_{i,j+1}-v_{i,j}}{h_y},\frac{v_{i,j-1}-v_{i,j}}{h_y}\right). \tag{2.11}$$

Note that the lower indices in $v_x$ and $v_y$ denote partial derivatives. Also, in (2.10)-(2.11) neither iteration nor time levels of the values of $v$ are specified yet, which in the end will be needed in the methods described before.

The reason for this lack of specification is that we use a combination of a sweeping technique and a Gauß-Seidel-type iteration in order to accelerate convergence, leading especially to a different choice of labels for each sweeping direction.

Let us stress that the use of the Gauß-Seidel-type approach was already implemented in [9], however the sweeping technique can also be combined with *all* introduced algorithmic approaches. We realise this improvement, as documented in the experimental section in this work, leading to more efficient implementations of the two algorithms based on the control theoretic formulation. Also, we use the Gauß-Seidel-type iteration within the semi-Lagrangian approach.

In addition to these relatively simple refinements, we employ a cascading multigrid method, cf. [1], for dealing with the Hamilton-Jacobi PDE. Also in the latter case, we consider two different discretisations of the source term commented on below. A basic point we briefly address are the numerical boundary conditions; however, this does not imply new algorithmic developments.

We proceed by briefly elaborating on all these discretisation issues.

**The Gauß-Seidel strategy.** Notice that at pixel $(i, j)$ the data from the pixels $(i, j)$, $(i \pm 1, j)$ and $(i, j \pm 1)$ contribute in the upwind formulae. For instance, let us assume that we iterate from left to right and, beginning with the top line, from top to bottom over the mesh points. Thus, ascending in $i$ and descending in $j$, we incorporate the available computed values into the scheme, accelerating convergence.

Concentrating for the presentation on the time-level formulation, i.e., incorporating artificial time level $n$ and $n+1$ representing the iteration number together with a time step size $\delta t$ into the presentation, the described procedure yields the formulae

$$v_x(ih_x, jh_y, n\delta t) \approx \min\left(0, \frac{v^n_{i+1,j} - v^n_{i,j}}{h_x}, \frac{v^{n+1}_{i-1,j} - v^n_{i,j}}{h_x}\right), \tag{2.12}$$

$$v_y(ih_x, jh_y, n\delta t) \approx \min\left(0, \frac{v^{n+1}_{i,j+1} - v^n_{i,j}}{h_y}, \frac{v^n_{i,j-1} - v^n_{i,j}}{h_y}\right). \tag{2.13}$$

Let us emphasize that the data $v^{n+1}_{i,j+1}$ and $v^{n+1}_{i-1,j}$ in (2.12)-(2.13) were already computed via the described method, so that they can be used for the computation of $v^{n+1}_{i,j}$.

**Sweeping.** We now turn to the sweeping technique adopted from [19]. The underlying practical problem addressed by this technique is that in the true solution of a hyperbolic problem information is transported along characteristics. Thus, iterating on the discrete level in only one manner – e.g., always ascending in $i$ and descending in $j$ as described above – the true information flow is not realised efficiently. As a remedy, it is obvious to proceed iterating in the following fashion, realising a cyclic definition of discrete propagation directions:

1. Left → Right, and Top → Bottom
2. Top → Bottom, and Right → Left
3. Right → Left, and Bottom → Top
4. Bottom → Top, and Left → Right

Exactly this procedure is called *sweeping.* As is easily seen, depending on the actual sweeping direction within the above cycle, different values $v^{n+1}_{i\pm1,j\pm1}$ are to be taken into account in (2.12)-(2.13).

**Cascading multigrid.** The cascading multigrid routine is a relatively easy-to-use algorithm. Practically, it is a coarse-to-fine strategy, where we start from a coarse level and iterate up to the finest level identical with the original image domain. Thereby, the refinement is always implemented by doubling the number of grid points in each direction, involving linear interpolation from known values to the newly inserted nodes. Of course, this implies that the original image must be given in a size identical to a power of two.

**Source term treatment.** The method used in [16] as well as the algorithms based on the Hamilton-Jacobi-Bellman equation employ an implicit treatment of the source term, leading to the use of Newton's method in order to solve the arising nonlinear equation. However, recently it was shown by a multi-scale analysis that in the direct Hamilton-Jacobi approach it is feasible to discretise the sources in a purely explicit way [2].

**Boundary treatment.** An important issue always is the numerical realisation of correct boundary conditions. In the context of the upwind differencing employed within the considered schemes, the correct, so-called state constraints boundary conditions – practically Dirichlet boundary conditions with value infinity – are satisfied automatically because of the effect of the minimisation procedure within the upwinding formulae. In practice, when making use of upwinding the state constraints are identical to homogeneous Neumann boundary conditions, realising at the discrete level $\nabla u \cdot \vec{n} = 0$ where $\vec{n}$ is the outer unit normal vector at the corresponding image boundary point.

**3. Experimental study of numerical schemes.** In this section, we perform the announced comparative study. For this, we fix the two optimal-control-based algorithms to the quite optimal set-up described in [3, 9]. Concerning the algorithm based on the Hamilton-Jacobi PDE, we study the effects of various choices which can be made, and evaluate the performance.

For the quantitative evaluation, we use synthetic images, as here all parameters of camera and illumination as well as the solution are known. The synthetic images are newly rendered versions of a 'classic' test image in SfS, namely the *vase image*, cf. [18].

The Lambertian vase test image shown in Fig. 3.1 is of size 128 × 128 pixels. It was rendered using the parameters $\mathsf{f} = 492$, $\sigma = 100000$. Also in Fig. 3.1, we depict the vase rendered with a Phong-type surface, where we used $\mathsf{f} = 492$, $I_s = I_d = I_s = 100000$, $k_a = 0$, $k_d = 0.7$, $k_s = 0.3$, $\alpha = 5$. As observable, the difference is largely due to the highlights which can safely be incorporated via the Phong model.

For initialising the iterative process for all algorithms, we solve the optimal control problem analytically for the specific control $a = (0, 0)^T$ in the Lambertian case.

Concerning the algorithms, we use the following notations:

- CFS denotes the semi-Lagrangian algorithm,
- PF denotes the optimal control approach,
- VBW/L denotes the Hamilton-Jacobi-based algorithm for Lambertian surfaces,
- VBW/P denotes the Hamilton-Jacobi-based algorithm for Phong-type surfaces.

At first, we evaluate the use of the sweeping strategy for the CFS and the PF algorithm, as the sweeping scheme was not previously applied in that context. In doing this,

– The stopping criterion (in the following indicated via 'Stop') is realised if the difference of two successive iterates is less than $10^{-5}$ in the maximum-norm.

– The source term is discretised implicitly.

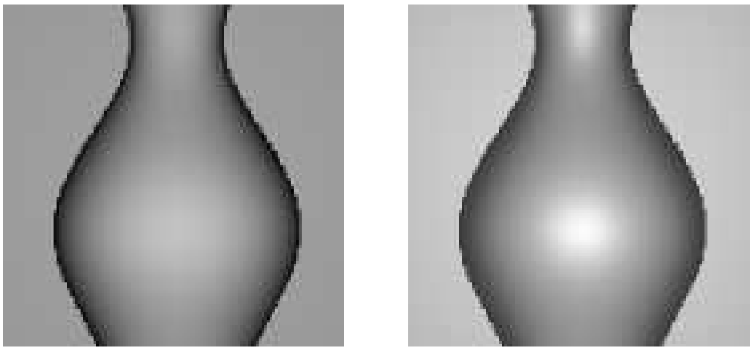

Fig. 3.1. *Synthetic vase input images. Left: Lambertian surface. Right: Phong-type surface.*

– Up to 100 Newton iterations are performed until convergence; in the following this number is indicated via 'Newton'. Note that this number has the role of an upper bound, in practice it is not reached.

We summarise the results by help of Table 3.1. As in all later experiments, the computational times ('Time') were obtained using an implementation in C on a standard PC (Linux, Pentium IV, 3.2 GHz, 2 GB RAM). We measure the error of the methods in terms of the relative depth error, i.e., in percentages of the true depth, as this makes sense in the context of the SfS task. A further interesting number is the number of iterations needed ('Iterations'). Note that one iteration consists practically of four sweeps, as indicated in the table. Also note that we always employ the Gauß-Seidel-type iteration in the algorithms without pointing this out explicitly.

Of course, the input image is the vase image with Lambertian surface. Initialising making use of the described procedure, the initial state features a relative $L_1$ depth error of 16.0792 %.

Table 3.1
*The Sweeping method: Relative $L_1$ depth errors, computational times and number of iterations.*

| Algorithm | Sweeping (y/n) | Error [%] | Time [$s$] | Iterations |
|---|---|---|---|---|
| PF | n | 0.398724 | 56.5 | 228 |
| CFS | n | 0.396839 | 47.0 | 144 |
| PF | y | 0.398724 | 36.6 | $4 \cdot 38$ |
| CFS | y | 0.396839 | 29.2 | $4 \cdot 23$ |

Evidently, making use of the sweeping strategy also renders the methods based on the Hamilton-Jacobi-Bellman equation more efficient. We take note of the reduced number of iterations in order to achieve convergence. In the following, we will always employ sweeping, and note only the number of iterations in terms of four sweeps.

Now we compare the optimal-control-based algorithms PF and CFS with variations of the VBW/L algorithm. The results are summarised in Table 3.2. In addition to the numbers we already introduced above, we also consider:

– Explicit or implicit ('e/i') discretisations of the source term.

– The use of the coarse-to-fine proceeding ('CTF'). Here, the number of iterations only represents the iterations on the finest grid.

As observable, the results are almost identical w.r.t. the measured error, given that the stopping criterion is selected properly. This is no true surprise, as we are essentially dealing with the same model, put into different formulations. Also, the results are very accurate, which is nothing but a quality statement, indicating that the model has reached maturity. Employing the coarse-to-fine procedure gives only slightly worse results. Note also that, at the same prescribed stopping criterion, the explicit formulation of the VBW/L algorithm is advantageous w.r.t. computational times compared to its semi-implicit variant. By the experiments employing an 'early stopping criterion' ($10^{-3}$) we observe that the direct approach needs more iterations than the other approaches until it reaches a high accuracy, while it is very efficient during the first few iterations. Overall, the PDE-based direct approach is more efficient.

Table 3.2
*Comparison of schemes for the Lambertian vase experiment.*

| Algorithm | Stop | CTF | e/i | Newton | Error [%] | Time [$s$] | Iterations |
|---|---|---|---|---|---|---|---|
| PF | $10^{-}5$ | $-$ | i | 100 | 0.398724 | 36.6 | 38 |
| PF | $10^{-}3$ | $-$ | i | 100 | 0.320825 | 18.3 | 18 |
| CFS | $10^{-}5$ | $-$ | i | 100 | 0.396839 | 29.2 | 23 |
| CFS | $10^{-}3$ | $-$ | i | 100 | 0.307 | 22.3 | 17 |
| VBW/L | $10^{-}5$ | $+$ | e | 0 | 0.411916 | 6.7 | 107 |
| VBW/L | $10^{-}3$ | $+$ | e | 0 | 0.394577 | 0.274 | 2 |
| VBW/L | $10^{-}5$ | $-$ | e | 0 | 0.395914 | 8.0 | 153 |
| VBW/L | $10^{-}5$ | $+$ | i | 100 | 0.412128 | 13.5 | 107 |
| VBW/L | $10^{-}5$ | $-$ | i | 100 | 0.395997 | 14.3 | 154 |
| VBW/L | $10^{-}5$ | $+$ | i | 1 | 0.412128 | 8.7 | 107 |
| VBW/L | $10^{-}8$ | $-$ | i | 100 | 0.398601 | 23.6 | 289 |
| VBW/L | $10^{-}7$ | $-$ | i | 100 | 0.3986 | 21.5 | 262 |
| VBW/L | $10^{-}6$ | $-$ | i | 100 | 0.398567 | 18.7 | 222 |
| VBW/L | $10^{-}4$ | $-$ | i | 100 | 0.393843 | 5.4 | 54 |
| VBW/L | $10^{-}3$ | $-$ | i | 100 | 0.396341 | 2.3 | 20 |
| VBW/L | $10^{-}2$ | $-$ | i | 100 | 1.98555 | 0.8 | 5 |
| VBW/L | $10^{-}1$ | $-$ | i | 100 | 3.27122 | 0.5 | 2 |

We now consider Phong-type surfaces. To evaluate the increase in computational time and the difference gained by the model improvement, we use the explicit versions of the VBW/L (L for Lambertian) and VBW/P (P for Phong) algorithms. The stopping criterion is applied if the difference between two successive iterations is less than $10^{-5}$ in the maximum-norm. Table 3.3 summarises the result. We observe only a moderate increase in computational effort, while the quality of the Phong-based reconstruction is clearly better compared to the result of the Lambertian model. This is due to the fact that the Lambertian model estimates the surface too close to the light source, misinterpreting the highlights. We also observe that relaxing the stopping criterion too much may lead to large errors; by this experiment one may conjecture that it is more demanding to reach the steady-state solution of the Phong model in comparison to the Lambertian case. See Fig. 3.2 for a visible account of the experiments.

Table 3.3
*Comparison of schemes for the vase experiment with Phong-surface.*

| Algorithm | input | Stop | Error [%] | Time [$s$] | Iterations |
|---|---|---|---|---|---|
| VBW/L | L | $10^-5$ | 0.411916 | 6.7 | 107 |
| VBW/L | Ph | $10^-5$ | 10.702 | 13.6 | 200 |
| VBW/P | Ph | $10^-5$ | 1.05791 | 19.9 | 281 |
| VBW/P | Ph | $10^-3$ | 5.39 | 1.4$s$ | 21 |

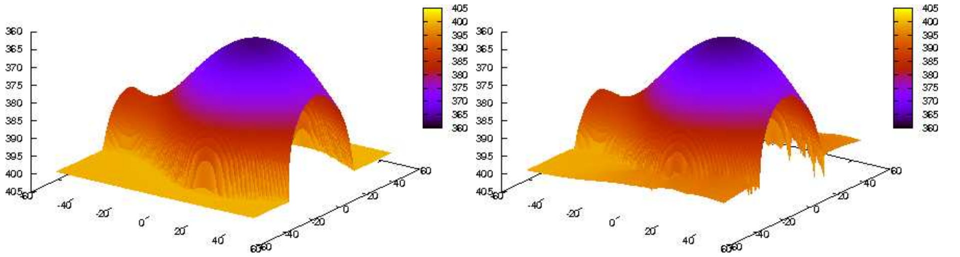

Fig. 3.2. *The Lambertian vase experiment, colour-coded depth maps. Left: Exact solution. Right: Lambertian reconstruction (looks the same for any of the algorithms in use).*

**4. Conclusion.** Our investigations show that the considered algorithms deliver equivalent results of convincing quality. Having investigated some possible algorithmic improvements, we have found that the direct approach based on the Hamilton-Jacobi PDE is most efficient compared to the schemes relying on the Hamilton-Jacobi-Bellman equation. The detailed discussion of the influence of numerical parameters particularly reveals the influence of the stopping criterion on the numerical realisation of the Lambertian and the Phong surface model. There is also no accuracy trade-off when discretising sources here explicitly. Also, improving the SfS model to more general surface models does not result in very high additional computational effort.

## REFERENCES

[1] 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.

[2] M. Breuß, O. Vogel and J. Weickert. Perspective shape from shading for Phong-type non-Lambertian surfaces. Technical Report No. 216, Faculty of Mathematics and Computer Science, Saarland University, August 2008.

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

[4] J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes. *Computer Graphics: Principles and Practice.* Addison–Wesley, 1996.

[5] 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.

[6] B. K. P. Horn. *Robot Vision.* MIT Press, Cambridge, MA, 1986.

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

[8] B. T. Phong. Illumination for computer-generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.

[9] E. Prados, F. Camilli, and O. Faugeras. A unifying and rigorous shape from shading method

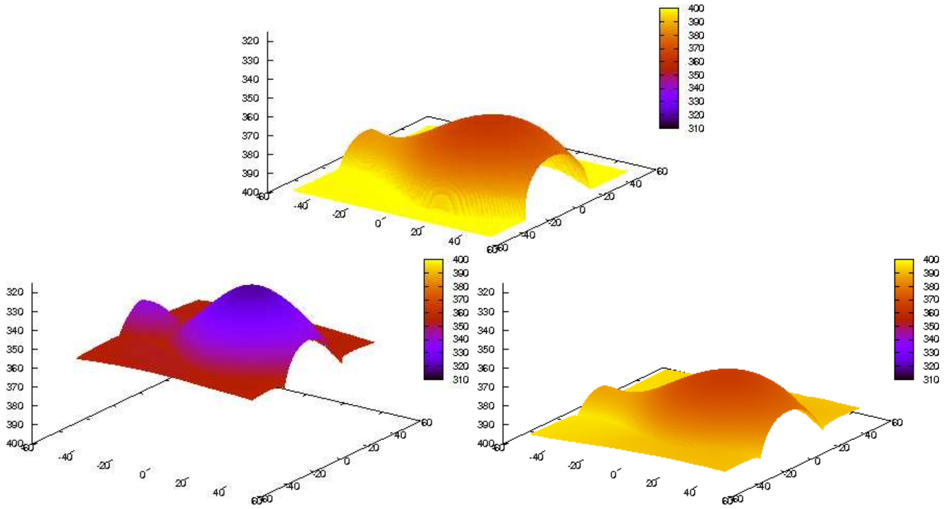

FIG. 3.3. *The Phong-surface vase experiment, colour-coded depth maps. Left: Exact solution. Middle: Lambertian reconstruction. Right: VBW/P reconstruction.*

adapted to realistic data and applications. *Journal of Mathematical Imaging and Vision*, 25(3):307–328, 2006.

[10] 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.

[11] 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.

[12] E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM Journal of Numerical Analysis*, 29(3):867–884, 1992.

[13] 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, Oct. 2003. IEEE Computer Society Press.

[14] 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.

[15] A. Tankus, N. Sochen, and Y. Yeshurun. Shape-from-shading under perspective projection. *International Journal of Computer Vision*, 63(1):21–43, June 2005.

[16] 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.

[17] 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, Berlin, June 2008. Springer.

[18] 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.

[19] H. Zhao. A fast sweeping method for eikonal equations. *Mathematics of Computation*, 74(250):603–627, 2004.