Shape-from-Shading by Iterative Fast Marching for Vertical and Oblique Light Sources *

Ariel Tankus (arielt@post.tau.ac.il)
School of Computer Science
Tel-Aviv University
Tel-Aviv, 69978

Nir Sochen (sochen@post.tau.ac.il)
School of Mathematics
Tel-Aviv University
Tel-Aviv, 69978

Yehezkel Yeshurun (hezy@post.tau.ac.il)
School of Computer Science
Tel-Aviv University
Tel-Aviv, 69978

Abstract. *Shape-from-Shading (SfS) is a fundamental problem in Computer Vision. Its goal is to solve the image irradiance equation. One prominent solution is the Fast Marching Method of Kimmel & Sethian. When the light source is oblique, Kimmel & Sethian proposed to rotate the image to the light source coordinate system and then solve an 'almost' Eikonal equation. This paper presents a new iterative variant of the Fast Marching Method which copes better with images taken under oblique light sources. Robustness is achieved by avoiding the change of coordinate system. The advantages of the proposed method are demonstrated on synthetic and real images.*

1. Introduction

Shape-from-Shading (SfS) is one of the fundamental problems in Computer Vision. First introduced by Horn in the 1970s [9], its goal is to solve the image irradiance equation, which relates the reflectance map to image intensity. An efficient way to solve this equation numerically is the celebrated Fast Marching Method of Sethian [15], [32].

Various methodologies have been proposed since the introduction of the field of Shape-from-Shading by Horn [8], [9], [10] in the 1970s. Horn's book [11] reviews the early approaches which include characteristic strips and Calculus of Variations. Zhang et al. [38] categorizes Shape-from-Shading techniques by their modus operandi. Namely, minimization approaches: [39], [21]; propagation approach: [1]; local approach: [20]; linear approaches: [26], [36]. A newer minimization approach is that of Robles-Kelly & Hancock [27],

* This research has been supported in part by Tel-Aviv University fund, the Adams Super-Center for Brain Studies, the Israeli Ministry of Science, the ISF Center for Excellence in Applied Geometry, the Minerva Center for geometry, and the A.M.N. fund.

which uses the Mumford-Shah functional to derive diffusion kernels. Other researchers put topological properties of the surface to use (e.g., Kimmel & Bruckstein [14]) or employ deformable models (e.g., Samaras & Metexas [29]). These are only examples, as the amount of work in the field of Shape-from-Shading is too large to describe herein.

Of particular relevance to this paper are works which utilize Level-Set and Fast Marching methodologies (see [32] for a deep insight). These approaches refer to the image irradiance equation as describing the motion of a front (e.g., [25], [16]). The Fast Marching Method re-orders the computation, to make it a one-pass solution of the Eikonal equation, based on the observation that the upwind difference structure of the numerical approximation allows us to propagate information "one way", that is from smaller values to larger values ([30], [31]). Sethian [30] proves the Fast Marching Method converges to the viscosity solution (see: [5], [22] for the definition and properties of viscosity solutions).

Kimmel & Sethian [15] implemented the Fast Marching Method as an optimal algorithm for surface reconstruction. They referred to the image irradiance equation as an Eikonal equation for vertical light sources. Solution of the equation for oblique light sources is obtained by rotation of the image coordinate system to that of the light source (as inspired by [20]).

While the Fast Marching Method is a highly efficient numerical solution to the image irradiance equation for vertical light sources, it is suboptimal for oblique light sources. For non-vertical light sources, the rotation of coordinate system requires an a-priori knowledge of the depth of the surface. As this knowledge is exactly the goal of the algorithm, one must employ an approximation, which reduces the robustness of the algorithm. This paper presents two new ways to employ the Fast Marching Method for oblique light sources as well. The first algorithm iteratively repeats the rotation with improved depth maps, while the second algorithm iteratively applies the complete Fast Marching Method for the Eikonal equation in the case of an oblique light source and avoids any rotation. Comparison with the original algorithm [15] would demonstrate that the second algorithm overcomes the limitations of the original.

The paper is organized as follows. First, we present the notation and basic assumptions (Sect. 2), and review the Fast Marching Method (Sect. 3). We then propose the two iterative methods for improved accuracy in cases where the light source is oblique (Sects. 4, 5). Section 6 compares the original method with the two new ones on both synthetic and real-life images. Finally, Sect. 7 draws the conclusions.

2. Notation and Assumptions

Let us first describe the notation and assumptions that hold throughout this paper. Photographed surfaces are assumed representable by functions of real-world coordinates. $z(x, y)$ denotes the depth function in a real-world Cartesian coordinate system whose origin is at camera plane. A real-world coordinate $(x, y, z(x, y))$ is projected orthographically onto image point (x, y). The intensity and surface normal at this image point are denoted: $I(x, y)$ and $\vec{N}(x, y)$, respectively. The intensity function $I(x, y)$ is assumed to be a positive, Lipschitz continuous function and lower than 1 (in order to ensure the existence of the strict viscosity subsolution) (see [28] for details). The scene object is Lambertian, and is illuminated by a point light source at infinity whose direction is: $\vec{L} = (p_s, q_s, -1)$.

3. The Fast Marching Method

This section reviews the Fast Marching method of Kimmel and Sethian [15] for vertical and oblique light sources.

3.1. MOTIVATION

The Shape-from-Shading problem for a Lambertian surface under directional light (assuming orthographic projection) is not well posed and may have infinitely many solutions (see, for example: [2], [3], [6], [4], [23], [7], [19], [18]). Various methodologies were suggested in the literature to deal with the ill-posedness (one such example is Photometric Stereo: [12], [37], [24], [17]). However, one may enforce uniqueness by adding the Dirichlet boundary conditions. Indeed, in many applications it is unrealistic to assume that one has these data, which are the goal of the algorithm. But as we would see, in the case of Fast Marching, Dirichlet boundary conditions are required merely at critical image points, not on image boundaries. At critical points it is possible to obtain the true depth by global topology solvers (e.g., [14]; see [15] for more details). We are therefore interested in Fast Marching techniques.

The next subsection would present a consistent and monotone ("upwind") numerical scheme which lies at the heart of the Fast Marching Method, and is the key to obtaining a unique viscosity solution. For the Eikonal equation, Rouy & Tourin [28] showed that an iterative algorithm based on this scheme with Dirichlet boundary conditions on image boundaries and at all critical points converges towards the viscosity solution with the same boundary conditions. While the algorithm of Rouy & Tourin [28] requires the Dirichlet boundary conditions on image boundaries and at all critical points, the Fast Marching Method needs the Dirichlet conditions only at critical points. Existence of the viscosity solution was proven in [22] and uniqueness, in [28]

and [13]. Sethian [30] proved that the Fast Marching Method produces a solution that everywhere satisfies the discrete version of the Eikonal equation. We next describe the algorithm.

3.2. FAST MARCHING FOR VERTICAL LIGHT SOURCES

The algorithm of Kimmel and Sethian [15] stems from the orthographic image irradiance equation:

$$I(x,y) = \vec{L} \cdot \vec{N}(x,y) = \frac{p_s z_x + q_s z_y + 1}{\|\vec{L}\| \sqrt{z_x^2 + z_y^2 + 1}} \tag{1}$$

For a vertical light source, that is $\vec{L} = (0,0,-1)$, the equation becomes an Eikonal equation which can be written as:

$$p^2 + q^2 = \tilde{F}^2 \tag{2}$$

where $p \stackrel{\text{def}}{=} z_x$, $q \stackrel{\text{def}}{=} z_y$ and $\tilde{F} = \sqrt{(I(x,y))^{-2} - 1}$.

Following [15], we use the numerical approximation (originally introduced in [28] as a modification of the scheme of [25]):

$$\begin{aligned} p_{ij} &\approx \max\{D_{ij}^{-x} z, -D_{ij}^{+x} z, 0\} \\ q_{ij} &\approx \max\{D_{ij}^{-y} z, -D_{ij}^{+y} z, 0\} \end{aligned}$$

where $D_{ij}^{-x} z \stackrel{\text{def}}{=} \frac{z_{ij} - z_{i-1,j}}{\Delta x}$ is the standard backward derivative and $D_{ij}^{+x} z \stackrel{\text{def}}{=} \frac{z_{i+1,j} - z_{ij}}{\Delta x}$, the standard forward derivative in the x-direction ($z_{ij} \stackrel{\text{def}}{=} z(i\Delta x, j\Delta y)$). $D_{ij}^{-y} z$ and $D_{ij}^{+y} z$ are defined in a similar manner for the y-direction.

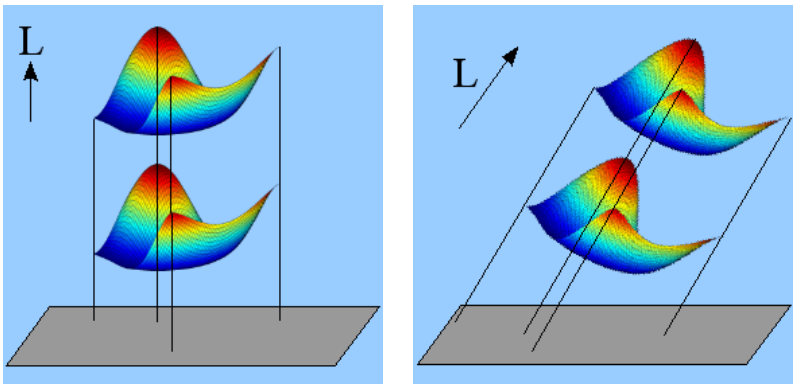
Substituting the numerical approximation into Eq. 2, we get the discrete equation:

$$\left(\max\{D_{ij}^{-x} z, -D_{ij}^{+x} z, 0\}\right)^2 + \left(\max\{D_{ij}^{-y} z, -D_{ij}^{+y} z, 0\}\right)^2 = \tilde{F}_{ij}^2 \tag{3}$$

where $\tilde{F}_{ij} \stackrel{\text{def}}{=} \tilde{F}(i\Delta x, j\Delta y)$. The solution of this equation at point (i,j), assuming depth is known at neighboring pixels, is:

$$z_{ij} = \begin{cases} \min\{z_1, z_2\} + \tilde{F}_{ij}, & \text{if} \quad \mid z_2 - z_1 \mid \geq \tilde{F}_{ij} \\ \frac{1}{2}\left(z_1 + z_2 \pm \sqrt{2\tilde{F}_{ij}^2 - (z_1 - z_2)^2}\right), & \text{if} \quad \mid z_2 - z_1 \mid < \tilde{F}_{ij} \end{cases} \tag{4}$$

where $z_1 \stackrel{\text{def}}{=} \min\{z_{i-1,j}, z_{i+1,j}\}$ and $z_2 \stackrel{\text{def}}{=} \min\{z_{i,j-1}, z_{i,j+1}\}$.



a. b.

Figure 1. Demonstration of the invariance properties of the orthographic image irradiance equation (a) and the equation used after rotation of the image to the light source coordinates (b).

3.3. Fast Marching in Light Source Coordinates

For oblique light sources (i.e., $\vec{L} \neq (0,0,-1)$), Kimmel & Sethian [15] adopted the idea of Lee & Rosenfeld [20] to rotate the brightness image to light source coordinates. This yields an 'almost' Eikonal equation (as [15] called it), which is solved in a manner similar to the vertical case, but in the new coordinate system.

Rotation to the light source coordinate system is, however, nontrivial. The image irradiance equation (Eq. 1) is invariant to depth translation. That is, $z(x,y)$ and $z(x,y)+c$ (for a constant c) generate identical irradiance. This occurs at coordinates (x,y) which are the camera coordinates (See Fig. 3.3a). Following the rotation, one solves the vertical light source case of the image irradiance equation (Eq. 2), which is also invariant to depth translation. However, it is now solved in a different coordinate system, so the direction of invariance is the direction of the new z-axis (i.e., the light source direction). Figure 3.3b demonstrates the invariance following the rotation.

Because of its dependence on surface depth, the rotation of the two surfaces: $z(x,y)$ and $z(x,y)+c$ to light source coordinates differ in the general case. In particular, the new (x,y) coordinates, which we denote: (x',y'), may be different. But these coordinates are also the new image coordinates. Thus, the image pixels used for the computation are different: $I(x',y')$ vs. $I(x'+cl_1, y'+cl_2)$, respectively, where $l_1 = p_s/\|\vec{L}\|$ and $l_2 = q_s/\|\vec{L}\|$ (the proof is omitted for brevity). Thus, a depth translation, which should preserve the irradiance under the orthographic model, requires a translation of the (x,y) coordinates as well, due to the rotation to light source coordinates. For

works on the perspective model, see: [33], [34], [35]. This would be further demonstrated by experimental results (Sect. 6.2).

As a result of the dependence of image coordinates on surface depth, the new image coordinates may lie outside image boundaries. No doubt, this results in loss of information. In our implementation, for pixels outside image boundaries, we duplicated the intensity of the nearest pixel on the boundary.

Another source of error in the calculation of the rotation is the use of an approximation for the depth of the surface. An approximation is necessary because the true depth is yet unknown when the rotation to light source coordinates takes place. Kimmel & Sethian [15] suggested to approximate depth as the minimal depth of neighboring pixels.

The use of the approximated depth for the rotation to light source coordinates results in an inaccurate rotation. Following that, the algorithm solves the vertical light source problem in light source coordinates, and rotates the resultant surface back to the original coordinates. The "inverse" rotation, however, is not exactly inverse to the first rotation, as it uses a more accurate depth map. An inaccurate rotation affects the shape of the $[xy]$ domain of the reconstructed surface. Consequently, a rectangular image is not necessarily reconstructed in a rectangular $[xy]$ domain, even though the projection model is orthographic.

4. A Rotation Iterative Solution

One way to improve the results of the Fast Marching Method for oblique light sources is to try and reduce the approximation error of the rotation to the light source coordinates.

The suggested method is iterative. It uses the depth recovered by the Fast Marching Method to recalculate the rotation. With this new rotation, rotate the image once again to the light source coordinates, and solve the vertical problem in the new coordinate system. From the depth so obtained, recalculate the rotation, solve the vertical problem and so forth. We would call this method: *Iterative Rotation*.

5. The Equation Iterative Solution

A different modus operandi to overcome the aforementioned flaws of the Fast Marching Method is to avoid any rotation to light source coordinates at all. Instead, we solve a series of Eikonal equations which are approximations to the image irradiance equation. Each equation should refine the approximation of its predecessor.

To formulate the approximate equations, we transform the image irradiance equation for an oblique light source (Eq. 1) into the form:

$$p^2 + q^2 = F^2(p, q) \tag{5}$$

where:

$$F(p,q) \stackrel{\text{def}}{=} \sqrt{1 - \left(\frac{p_s p + q_s q + 1}{\| \vec{L} \| \, I(x,y)} \right)^2}$$

A significant difference between the vertical and oblique cases is the dependence of F on p and q.

An important observation described in [15] is that when updating the depth values according to the discrete equation (Eq. 3), information always flows from small to large values. Based on this, the Fast Marching Method reconstructs depth in an "upwind" fashion. It first sets all z values to the correct height values at local minima and to infinity elsewhere. Then, every step extends the reconstruction to higher depths. Reconstruction is thus achieved in one pass.

Nevertheless, a single pass may not be enough to solve the aforementioned formulation of the oblique problem (Eq. 5), because the approximate solution (the right-hand side of Eq. 4) depends on F, which depends on both p and q. Hence, we suggest another iterative method. At each iteration, F is calculated using the depth recovered at the preceding iteration:

$$p_{n+1}^2 + q_{n+1}^2 = F^2(p_n, q_n) \tag{6}$$

where p_n and q_n are the values of p and q at the $n^{\underline{\text{th}}}$ iteration. The algorithm is thus:

1. Step 0: Initialize (p_0, q_0) by the Fast Marching Method of Kimmel & Sethian [15].

2. Step n:

   a) Based on the approximation (p_n, q_n), calculate the right-hand side of Eq. 6, namely, evaluate $F^2(p_n, q_n)$.

   b) As we computed $F^2(p_n, q_n)$, Eq. 6 is now Eikonal. Use Eq. 4 to obtain a solution: (p_{n+1}, q_{n+1}).

   c) Following each iteration we normalize the depth function $z(x, y)$ (divide by the mean z value) to compensate for the lack of knowledge of grid size $(\Delta x, \Delta y)$.

3. Let $n := n + 1$, and repeat Step n.

We call this method: *Iterative Equation*.

This iterative process results in a series of Eikonal equations, each solved by the Fast Marching Method. Sethian [30] showed that the Fast Marching Method produces a solution that everywhere satisfies the discrete version of the Eikonal equation. Therefore, the Fast Marching solution of each of the equations in the series satisfies the discrete version of that equation. As a result, when the series of solutions to the Eikonal equations converges, the limit is the correct solution of the discrete version of the original equation (i.e., the solution of the image irradiance equation with an oblique light source).

Empirically, in all experiments the series of solutions converged. In fact, very few iterations were necessary to obtain this convergence (i.e., to get close enough to the limit).

6. Experimental Results

6.1. The Experiments

To evaluate the contribution of the proposed algorithms, we compared them with the original formulation of the Fast Marching Method [15]. The evaluation involved both synthetic images and real-life images. The synthetic images were produced from a given depth map using the image irradiance equation (Eq. 1). The derivatives in the equation were calculated numerically.

The initialization of the algorithms is based on points of local minima. For synthetic images, these were extracted automatically from the true depth map. For real images, they were located visually in each photograph by a human viewer, and their depths were arbitrarily set to the same constant. To demonstrate the aforementioned undesired features of the Fast Marching Method [15] (Sect. 3.3), we ran the algorithms twice for each surface. In the second run, the depth of the original initialization (described above) was translated by a constant. Theoretically, this should merely translate the whole reconstruction along the z-axis by the same constant.

In our comparison we checked five iterations of the iterative Fast Marching Methods for each example. For the Iterative Equation method, we found out that all iterations (maybe except for the first one) yielded visually-identical images, which implies the suggested algorithm converges very fast.

To quantitatively evaluate the performance of the algorithms on synthetic data, we adopted three criteria from Zhang et al. [38]. These are: mean depth error, standard deviation of depth error, and mean gradient error. For completeness, we also supply the standard deviation of the gradient error, even though it is considered nonphysical.

Table I. Error rates for the algorithms on $z(x,y) = 100 + \cos\left(\sqrt{x^2 + (y-2)^2}\right)$.

Algorithm:	No. of Iters.:	Mean Depth Error:	Std. Dev. of Depth Error:	Mean Grad. Error:	Std. Dev. of Grad. Error:
Fast Marching:	1	0.51687	0.29194	2.20442	1.01374
Iterative Rotation:	1	0.51636	0.29127	2.16523	1.01180
Iterative Rotation:	2	0.51692	0.29179	2.20046	1.01357
Iterative Rotation:	3	0.51697	0.29185	2.20573	1.01359
Iterative Rotation:	4	0.51699	0.29189	2.20869	1.01362
Iterative Rotation:	5	0.51697	0.29186	2.20626	1.01355
Fast Marching:	1	0.51687	0.29194	2.20442	1.01374
Iterative Equation:	1	0.37269	0.28148	1.05731	0.88570
Iterative Equation:	2	0.37213	0.28217	1.05174	0.88535
Iterative Equation:	3	0.37189	0.28203	1.05107	0.88494
Iterative Equation:	4	0.37188	0.28202	1.05104	0.88493
Iterative Equation:	5	0.37188	0.28202	1.05104	0.88493

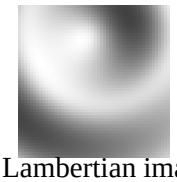
6.2. COMPARATIVE EVALUATION

Figure 2 compares the original Fast Marching Method with the two iterative methods (Iterative Rotation and Iterative Equation) on the following depth map:

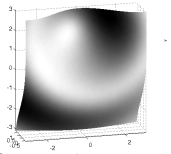
$$z(x,y) \stackrel{\text{def}}{=} 100 + \cos\left(\sqrt{x^2 + (y-2)^2}\right)$$

where: $x, y \in [-3.0788, 3.0788]$ (image size: 50×50 pixels). The Iterative Rotation does not improve upon the original Fast Marching; their reconstructions are very similar. The original Fast Marching Method reconstructed only a part of the cosine function (the upper right part of the surface in Fig. 2C, iteration: 0), due to the translated $[xy]$ coordinates in the calculation. This part corresponds to the lower-left part of the original surface (Fig. 2B). The rest of the reconstruction is a result of a calculation using pixels outside image boundaries (as described in Sect. 3.3). The Iterative Equation, on the other hand, reconstructed the right-hand side of the cosine correctly, with more noise on the left hand side (part of the elevated domain of the cosine appears almost flat there). Table I presents the error rates according to the aforementioned criteria. As expected, all error rates of the Iterative Rotation and Fast Marching methods are very close to one another. The Iterative Equation algorithm obtained lowest error rates according to all criteria.

Figure 3 shows the famous example of the Vase ($x, y \in [-63.5, 63.5]$;



A. Lambertian image.



B. Original surface.

iter#	Iterative Rotation:	Iterative Equation:
0.		
1.		
2.		
3.		
4.		
5.		

C. Reconstruction comparison.

Figure 2. Three variants of the Fast Marching Method for $z(x,y) = 100 + \cos\left(\sqrt{x^2 + (y-2)^2}\right)$. Each row corresponds to a different iteration (Row 0 is the original Fast Marching). Lighting is identical for all reconstructions, and is equal to that of (A).

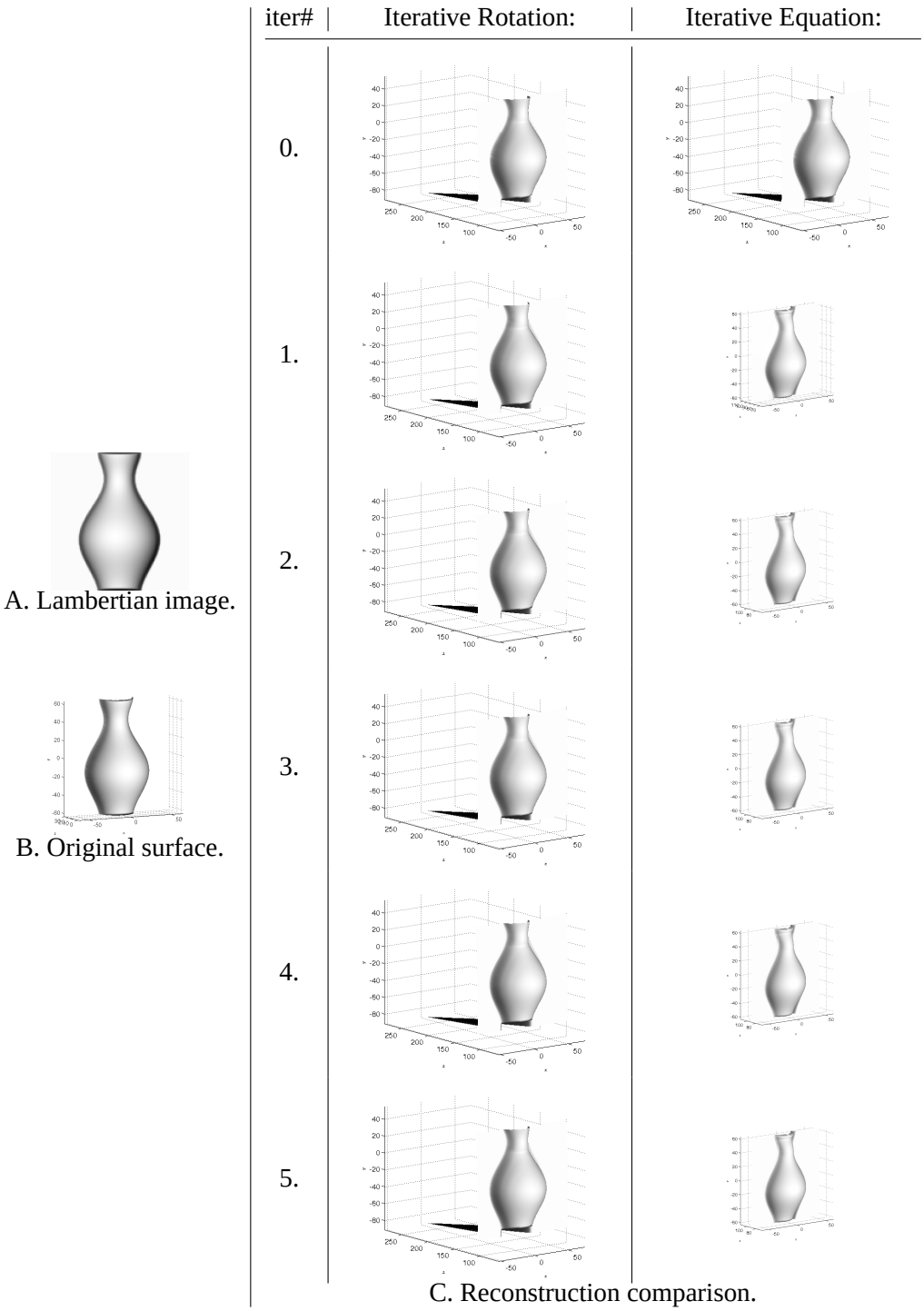


Figure 3. Three variants of the Fast Marching Method for the Vase example. Each row corresponds to a different iteration (Row 0 is the original Fast Marching). Lighting is identical for all reconstructions, and is equal to that of (A).

Table II. Error rates for the algorithms on the Vase example.

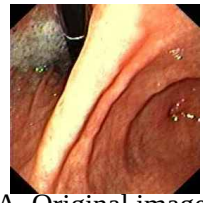
Algorithm:	No. of Iters.:	Mean Depth Error:	Std. Dev. of Depth Error:	Mean Grad. Error:	Std. Dev. of Grad. Error:
Fast Marching:	1	7.95881	5.87365	13.99250	26.45452
Iterative Rotation:	1	8.04165	5.84769	13.99933	26.28860
Iterative Rotation:	2	8.07961	5.85130	13.99862	26.29129
Iterative Rotation:	3	8.04709	5.85250	13.99551	26.36240
Iterative Rotation:	4	8.07280	5.85283	13.99254	26.34710
Iterative Rotation:	5	8.05496	5.85172	13.98958	26.34410
Fast Marching:	1	7.95881	5.87365	13.99250	26.45452
Iterative Equation:	1	4.12683	3.36455	5.70288	14.27210
Iterative Equation:	2	4.47005	3.49335	5.48740	14.09266
Iterative Equation:	3	4.52750	3.51408	5.45683	14.09894
Iterative Equation:	4	4.54517	3.51939	5.44802	14.09603
Iterative Equation:	5	4.55495	3.51974	5.44278	14.08841

image size: 128×128; background depth: 100). The original Fast Marching yielded a sharp bulge at the foot of the vase. The bulge appears in a domain whose reconstruction was computed from pixels outside image boundaries. Iterative Rotation recovers similar surfaces. The Iterative Equation method, on the other hand, reconstructed a much smoother vase. Its stronger resemblance to the original is not only visible but can also be quantified by all error measures (Table II). The Iterative Rotation and Fast Marching methods equate; the mean depth error of Iterative Rotation is even a little higher than that of Fast Marching.

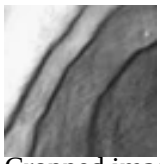
Figure 4 introduces a real-world example taken by endoscopy from the gastric angulus[1] (cropped image size: 64×64). The algorithm of Kimmel and Sethian reconstructs two of the gastric folds. However, the reconstructed wall of the gastric angulus seems to consist of perpendicular planes (instead of small folds on a main low-convexity surface). On the right-hand side of the reconstruction, the surface appears "higher" (i.e., larger y-rates[2]) than on the left-hand side. Thus, the $[xy]$ domain of the reconstructed surface is not rectangular. As explained in Sect. 3.3, this is due to inaccurate rotation to the light source coordinate system. Iterative Rotation seems to have improved

[1] Original is from www.gastrolab.net, courtesy of The Wasa Workgroup on Intestinal Disorders, GASTROLAB, Vasa, Finland.

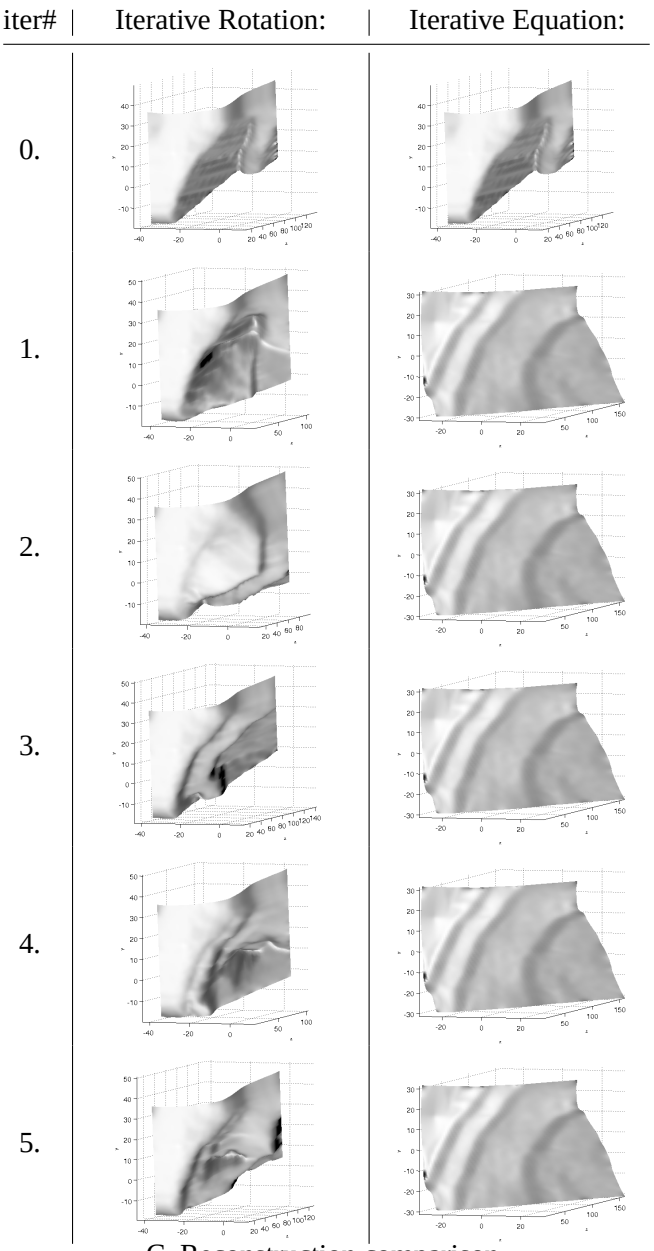
[2] Recall, that the $[xy]$ domain is perpendicular to the optical axis, so the coordinate system of the reconstructed surfaces is: x – to the right; y – up; z – away from the viewer.



A. Original image.

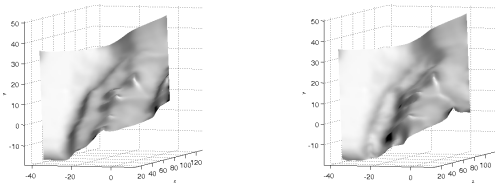


B. Cropped image.



C. Reconstruction comparison.

Figure 4. Three variants of the Fast Marching Method for an endoscopic image of the Gastric Angulus. Each row corresponds to a different iteration (Row 0 is the original Fast Marching). Lighting is identical for all reconstructions, and is approximately that of (B). Only (B) was used for the reconstruction.



Iteration #19. Iteration #20.

Figure 5. Reconstruction of the Gastric Angulus example by the Iterative Rotation method. The iterations following the $19^{\underline{\text{th}}}$ and $20^{\underline{\text{th}}}$ are repetitions of these two (there is no visual difference). The running conditions described in Fig. 4 are valid here as well.

the reconstructed shape. After 19 iterations, it begins to flip between two reconstructed surfaces (Fig. 5). The three gastric folds are reconstructed in both states, with better results in the odd states. In the Iterative Rotation reconstruction, some cavities are present near the central fold. The upper-right part of the surface is also not faithfully recovered. The contours of the recovered folds are not as smooth as in the original image. The Iterative Equation method seems to have reconstructed the three gastric folds in quite an accurate manner. Its accuracy appears to be higher than that of the Iterative Rotation method and in less iterations: only 1 iteration was necessary for the Iterative Equation to converge.

6.3. Comparison of Robustness in Depth Translation

In this subsection, we would like to evaluate the robustness of the algorithms in depth translations. We therefore juxtapose the reconstructions of surfaces $z(x, y)$ and $z(x, y) + c$ (c is constant) by the three methods. To obtain reconstructions of $z(x, y) + c$, we increased the initial depth values (at minima points) by a constant with respect to the initial values employed to reconstruct $z(x, y)$. Theoretically, the reconstructions should be identical up to depth translation, due to the invariance of the orthographic image irradiance equation.

In the following examples, only one iteration of the iterative methods is displayed, for the sake of brevity.

Figure 6 shows the reconstructions of the Cosine example of Fig. 2 by the three methods. The reconstructions in the middle column are identical to those of Fig. 2. The initializations used for creating them were taken from the original depth map. The rightmost column was created with the translated initializations, so reconstructions should be translated. Nevertheless, only the Iterative Equation method reconstructed a surface of the same shape for the two initializations. The Fast Marching and Iterative Rotation methods were highly affected by the depth translation. As Fig. 7 demonstrates, the difference between the surfaces reconstructed by Fast Marching with different

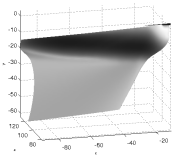
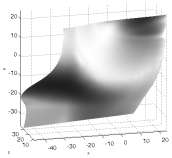
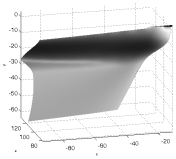
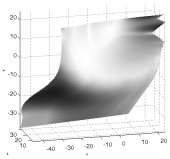
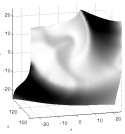
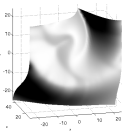
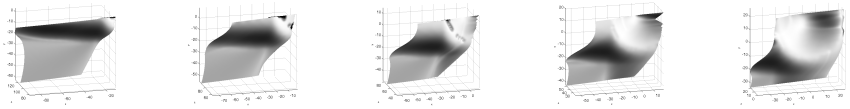
Algorithm:	Reconstruction from Original Initialization:	Reconstruction from Translated Initialization:
Fast Marching:		
Iterative Rotation: (Iteration #5)		
Iterative Equation: (Iteration #5)		

Figure 6. Comparison of the three algorithms on the Cosine example. Each algorithm was run with two initializations which were identical up to a constant translation along the z-direction ($c = -90$). Only the Iterative Equation algorithm remained invariant to the translation, while the two others showed a significant change between initializations. Thus, the Iterative Equation algorithm better maintains the invariance to depth translations.



$c = 0$ $c = 24.75$ $c = 49.5$ $c = 74.25$ $c = 99$

Figure 7. Reconstruction of the Cosine example by the Fast Marching method with different translations of the depth function. Depth translation results in $[xy]$ translation in the calculation of rotation to light source coordinates. Below each reconstruction is its translation (c) with respect to the original depth function.

depth translations is due to translation in the x and y coordinates in the calculation of the rotation to light source coordinates (as explained in Sect. 3.3). This translation requires some of the pixels to be taken from outside image boundaries, so in practice their values are duplicated from boundary pixels. Other pixels are simply shifted in place when the depth map is translated. The specific translation used (namely, $c = -90$) seems to have improved the reconstruction drastically. Indeed, it reduced the amount of pixels outside image boundaries prominently.

Figure 8 displays the reconstructed Vase (see Fig. 3) for the original and translated initializations. For the Fast Marching method the translation was so large that the vast majority of pixels were taken from outside image bound-

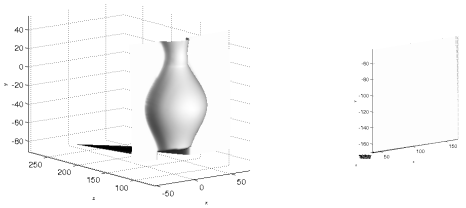
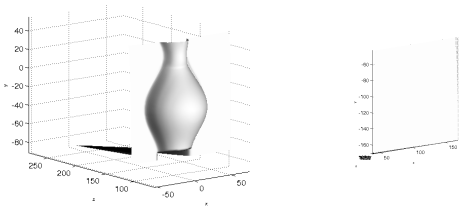
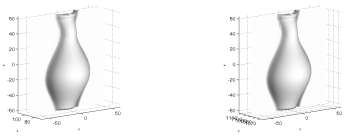
Algorithm:	Reconstruction from Original Initialization:	Reconstruction from Translated Initialization:
Fast Marching:		
Iterative Rotation: (Iteration #5)		
Iterative Equation: (Iteration #5)		

Figure 8. Comparison of the three algorithms on the Vase example. The initialization was translated by $c = +1000$ with respect to the original one. The Fast Marching Method and the Iterative Rotation methods yielded similar results. Both changed substantially with the change of initialization. In contrast, the Iterative Equation method maintained the invariance to depth translation.

aries. Thus, the reconstructed surface is almost planar, showing no sign of the original structure of the vase. Again, the change in reconstruction is prominent for the Fast Marching and Iterative Rotation methods, but not for Iterative Equation.

Figure 9 presents the original and translated reconstructions for the Gastric Angulus example of Fig. 4. The depth-translated initialization shifts the reconstructions of the Fast Marching and Iterative Rotation methods in the $[xy]$ plane. Thus, some of the pixels are evaluated outside image boundaries. Pay attention that in this real-life example the true depth at minima points was a-priori unknown, and the algorithm was initialized based on a human guess. A different guess could result in a significant change to the reconstruction. The Iterative Equation maintained its response in spite of the depth translation.

From the figures, one can see that the Fast Marching and Iterative Rotation methods were highly affected by the translation in contrast with the theoretic invariance of the underlying equation. This demonstrates the drawbacks of rotation to light source coordinates discussed in detail in Sect. 3.3. As opposed to these two algorithms, the variation in reconstruction by the Iterative Equa-

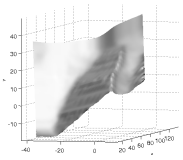
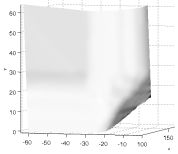
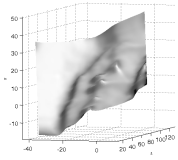
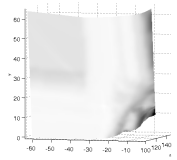
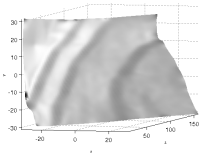
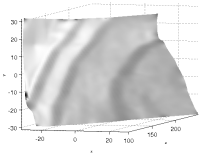
Algorithm:	Reconstruction from Original Initialization:	Reconstruction from Translated Initialization:
Fast Marching:		
Iterative Rotation: (Iteration #19)		
Iterative Equation: (Iteration #5)		

Figure 9. Comparison of the three algorithms on the Gastric Angulus example with two initializations. The difference between initializations was $c = +90$. Only the Iterative Equation method exhibited invariance to depth translation, while the two others showed a pronounced change between initializations.

tion method was very small. Quantification of the results in the form of depth and gradient errors appears in Tables III and IV (for the synthetic examples only). Table III (the Cosine example) confirms the visual impression that reconstructions by the Fast Marching and Iterative Rotation methods were improved by the specific translation selected for this example. Indeed, due to the translation the error rates dropped significantly with respect to those of Table I. However, they are still slightly higher than those of the Iterative Equation method (except for the standard deviation of the gradient error, which is considered nonphysical). In Table IV (the Vase example), all error rates of the Fast Marching and Iterative Rotation methods altered with respect to the corresponding values in Table II. Their change is not as strong as for the Cosine example, but is still higher than that of the Iterative Equation method. In both Tables III and IV variations in the error rates of Iterative Equation are only minor. Pay attention, that identical error rates do not imply identical reconstructions. Nevertheless, a significant change to these measures certainly indicates a notable change in surface shape.

We see, that in all examples, the Iterative Equation method appears to outrank the methods which rotate the image to the light source coordinate system: Fast Marching and Iterative Rotation.

Table III. Comparison of algorithms on $z(x,y) = 100 + \cos\left(\sqrt{x^2 + (y-2)^2}\right)$, with initialization translated by -90. Pay attention to the sharp change in all measures of the Fast Marching and Iterative Rotation methods with respect to Table I.

Algorithm:	No. of Iters.:	Mean Depth Error:	Std. Dev. of Depth Error:	Mean Grad. Error:	Std. Dev. of Grad. Error:
Fast Marching:	1	0.38683	0.28843	1.43113	0.80503
Iterative Rotation:	1	0.38089	0.27686	1.40019	0.75335
Iterative Rotation:	2	0.37521	0.28065	1.37679	0.77419
Iterative Rotation:	3	0.35533	0.26359	1.28275	0.72974
Iterative Rotation:	4	0.35734	0.26579	1.29597	0.74782
Iterative Rotation:	5	0.35535	0.26404	1.28020	0.73735
Fast Marching:	1	0.38683	0.28843	1.43113	0.80503
Iterative Equation:	1	0.36179	0.27772	1.02285	0.86817
Iterative Equation:	2	0.35912	0.27893	1.00283	0.85665
Iterative Equation:	3	0.35695	0.27596	1.00111	0.85411
Iterative Equation:	4	0.35850	0.27733	1.00394	0.85671
Iterative Equation:	5	0.35869	0.27748	1.00448	0.85694

When comparing the complexity of the three algorithms, no doubt the original one is the fastest, by containment. However, as the examples show, the speed in this case is at the expense of accuracy. The Iterative Equation method converges very fast and no more than 2 iterations were ever required to obtain it, so the speed difference turns out to be of secondary importance.

7. Conclusions

This research proposes an efficient and robust solution to the problem of Shape-from-Shading which handles both vertical and oblique light sources under the orthographic projection model. The suggested solution is a variant of the Fast Marching Method of Kimmel and Sethian [15]. It employs the Fast Marching Method iteratively for oblique light sources. Each iteration solves an approximation to the image irradiance equation. The resultant solution serves for successive refinement of the approximating equation. We called this algorithm: the Iterative Equation method. When this refinement process converges, convergence is to the correct solution of the original equation.

We compared reconstruction by the original Fast Marching Method, the Iterative Rotation method (which successively refines the rotation to light

Table IV. Comparison of algorithms on the Vase example with translated initialization (+1000). Note the significant change in mean gradient error of the original Fast Marching Method with respect to Table II.

Algorithm:	No. of Iters.:	Mean Depth Error:	Std. Dev. of Depth Error:	Mean Grad. Error:	Std. Dev. of Grad. Error:
Fast Marching:	1	9.36239	5.74319	14.35475	18.95690
Iterative Rotation:	1	9.36239	5.74319	14.35475	18.95690
Iterative Rotation:	2	9.36239	5.74319	14.35475	18.95690
Iterative Rotation:	3	9.36239	5.74319	14.35475	18.95690
Iterative Rotation:	4	9.36239	5.74319	14.35475	18.95690
Iterative Rotation:	5	9.36239	5.74319	14.35475	18.95690
Fast Marching:	1	9.36239	5.74319	14.35475	18.95690
Iterative Equation:	1	4.60417	3.56963	5.45874	14.25520
Iterative Equation:	2	4.60904	3.57800	5.44469	14.26232
Iterative Equation:	3	4.60800	3.57882	5.44469	14.26547
Iterative Equation:	4	4.60788	3.57916	5.44462	14.26633
Iterative Equation:	5	4.60779	3.57945	5.44472	14.26703

source coordinates) and the Iterative Equation method on both synthetic and real-life examples (from endoscopy). We also demonstrated why rotation of the image to light source coordinates, as required by the Fast Marching and Iterative Rotation methods, is unstable. The Iterative Equation method outperformed the two other methods, and remained invariant to depth translations (due to its convergence to the correct solution).

In terms of runtime, indeed the original Fast Marching Method is faster than the suggested ones. However, convergence of the Iterative Equation method is very fast; in all examples no more than 2 iterations were ever necessary.

References

1. Bichsel, M. and A. P. Pentland: 1992, 'A Simple Algorithm for Shape from Shading'. In: *Computer Vision and Pattern Recognition*. pp. 459–465.
2. Brooks, M. J. and W. Chojnacki: 1994, 'Direct Computation of Shape from Shading'. In: *Proceedings of the International Conference on Pattern Recognition*. Israel, pp. 114–119.
3. Brooks, M. J., W. Chojnacki, and R. Kozera: 1992a, 'Circularly-Symmetric Eikonal Equations and Non-uniqueness in Computer Vision'. *Journal of Mathematical Analysis and Applications* **165**(1), 192–215.

4. Brooks, M. J., W. Chojnacki, and R. Kozera: 1992b, 'Impossible and Ambiguous Shading Patterns'. *International Journal of Computer Vision* **7**(2), 119–126.
5. Crandall, M. G. and P.-L. Lions: 1983, 'Viscosity Solutions of Hamilton-Jacobi Equations'. *Transactions of the American Mathematical Society* **277**(1), 1–42.
6. Deift, P. and J. Sylvester: 1981, 'Some Remarks on the Shape-from Shading Problem in Computer Vision'. *Journal of Mathematical Analysis and Applications* **84**, 235–248.
7. Dupuis, P. and J. Oliensis: 1992, 'Direct Method for Reconstructing Shape from Shading,'. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Champaign, Illinois, pp. 453–458.
8. Horn, B.: 1975, 'Obtaining Shape from Shading Information'. In: P. H. Winston (ed.): *The Psychology of Computer Vision*, Computer Science Series. McGraw-Hill Book Company, Chapt. 4, pp. 115–155.
9. Horn, B. K. P.: 1977, 'Image Intensity Understanding'. *Artificial Intelligence* **8**(2), 201–231.
10. Horn, B. K. P.: 1986, *Robot Vision*. The MIT Press/McGraw-Hill Book Company.
11. Horn, B. K. P. and M. J. Brooks (eds.): 1989, *Shape from Shading*. The MIT Press.
12. Ikeuchi, K., H. K. Nishihara, B. K. Horn, P. Sobalvarro, and S. Nagata: 1986, 'Determining Grasp Configurations using Photometric Stereo and the PRISM Binocular Stereo System'. *The International Journal of Robotics Research* **5**(1), 46–65.
13. Ishii, H.: 1987, 'A Simple, Direct Proof of Uniqueness for Solutions of the Hamilton-Jacobi Equations of Eikonal Type'. *Proceedings of the American Mathematical Society* **100**(5), 247–251.
14. Kimmel, R. and A. M. Bruckstein: 1995, 'Global Shape from Shading'. *Computer Vision and Image Understanding* **62**(3), 360–369.
15. Kimmel, R. and J. A. Sethian: 2001, 'Optimal Algorithm for Shape from Shading and Path Planning'. *Journal of Mathematical Imaging and Vision* **14**(3), 237–244.
16. Kimmel, R., K. Siddiqi, B. B. Kimia, and A. M. Bruckstein: 1995, 'Shape from Shading: Level Set Propagation and Viscosity Solutions'. *International Journal of Computer Vision* **16**(2), 107–133.
17. Klette, R., R. Kozera, and K. Schlüns: 1999, 'Shape from Shading and Photometric Stereo Methods'. In: B. Jaehne, H. Haussecker, and P. Geissler (eds.): *Handbook of Computer Vision and Applications*, Vol. 2 of *Signal Processing and Pattern Recognition*. San Diego California: Academic Press Inc., Chapt. 19, pp. 532–590.
18. Klette, R., K. Schlüns, and A. Koschan: 1998, *Computer Vision: Three Dimensional Data from Images*. Springer.
19. Kozera, R.: 1997, 'Uniqueness in Shape from Shading Revisited'. *Journal of Mathematical Imaging and Vision* **7**(2), 123–138.
20. Lee, C.-H. and A. Rosenfeld: 1985, 'Improved Methods of Estimating Shape from Shading using the Light Source Coordinate System'. *Artificial Intelligence* **26**, 125–143.
21. Lee, K. M. and C.-C. J. Kuo: 1993, 'Shape from Shading with a Linear Triangular Element Surface Model'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**(8), 815–822.
22. Lions, P.-L.: 1982, *Generalized Solutions of Hamilton-Jacobi Equations*. London: Pitman.
23. Oliensis, J.: 1991, 'Uniqueness in Shape from Shading'. *International Journal of Computer Vision* **6**(2), 75–104.
24. Onn, R. and A. Bruckstein: 1990, 'Integrability Disambiguates Surface Recovery in Two-Image Photometric Stereo'. *International Journal of Computer Vision*.
25. Osher, S. and J. A. Sethian: 1988, 'Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulation'. *Journal of Computational Physics* **79**, 12–49.

26. Pentland, A. P.: 1984, 'Local Shading Analysis'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**(2), 170–187.
27. Robles-Kelly, A. and E. R. Hancock: 2002, 'Model Acquisition using Shape-from-Shading'. In: F. J. Perales and E. R. Hancock (eds.): *The* $2^{\underline{nd}}$ *International Workshop on Articulated Motion and Deformable Objects*. Palma de Mallorca, pp. 43–55.
28. Rouy, E. and A. Tourin: 1992, 'A Viscosity Solutions Approach to Shape-from-Shading'. *SIAM Journal of Numerical Analysis* **29**(3), 867–884.
29. Samaras, D. and D. Metaxas: 1999, 'Coupled Lighting Direction and Shape Estimation from Single Images'. *Proceedings of the Seventh IEEE International Conference on Computer Vision* **2**, 868–874.
30. Sethian, J. A.: 1996a, 'A Fast Marching Level Set Method for Monotonically Advancing Fronts'. *Proceedings of the National Academy of Science of the USA* **93**, 1591–1595.
31. Sethian, J. A.: 1996b, 'A Review of the Theory, Algorithms, and Applications of Level Set Methods for Propagating Interfaces'. In: *Acta Numerica*. Cambridge University Press.
32. Sethian, J. A.: 1999, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge Monograph on Applied and Computational Mathematics. Cambridge University Press, 2 edition.
33. Tankus, A., N. Sochen, and Y. Yeshurun: 2003, 'A New Perspective [on] Shape-from-Shading'. In: *Proceedings of the* $9^{\underline{th}}$ *IEEE International Conference on Computer Vision*, Vol. II. Nice, France, pp. 862–869.
34. Tankus, A., N. Sochen, and Y. Yeshurun: 2004a, 'Perspective Shape-from-Shading by Fast Marching'. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. I. Washington, DC, pp. 43–49.
35. Tankus, A., N. Sochen, and Y. Yeshurun: 2004b, 'Reconstruction of Medical Images by Perspective Shape-from-Shading'. In: *Proceedings of the International Conference on Pattern Recognition*, Vol. 3. Cambridge, UK, pp. 778–781.
36. Tsai, P.-S. and M. Shah: 1994, 'Shape from Shading using Linear Approximation'. *Image and Vision Computing* **12**(8), 487–498.
37. Woodham, R. J.: 1989, 'Photometric method for determining surface orientation from multiple images'. in [11], pp. 513–531.
38. Zhang, R., P.-S. Tsai, J. E. Cryer, and M. Shah: 1999, 'Shape from Shading: A Survey'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(8), 690–705.
39. Zheng, Q. and R. Chellappa: 1991, 'Estimation of Illuminant Direction, Albedo, and Shape from Shading'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(7), 680–702.