

Numerical Solution of the Perspective Shape-from-Shading Problem

Emiliano Cristiani^{*†}

Università di Roma "La Sapienza"

E-mail: cristiani@dmmm.uniroma1.it

Maurizio Falcone

Università di Roma "La Sapienza"

E-mail: falcone@mat.uniroma1.it

Alessandra Seghini

Università di Roma "La Sapienza"

E-mail: seghini@mat.uniroma1.it

We study the Perspective Shape from Shading problem from the numerical point of view presenting a simple algorithm to compute its solution. The scheme is based on a semi-Lagrangian approximation of the first order Hamilton-Jacobi equation related to the problem. The scheme is converging to the weak solution (in the viscosity sense) of the equation and allows to compute accurately regular as well as non regular solutions. Since the partial differential equation must be complemented with boundary conditions on the silhouette, we analyse the effect of various types of boundary conditions on the numerical solution. Some tests on synthetic and real images are presented.

Control Systems: Theory, Numerics and Applications

30 March - 1 April 2005

Rome

^{*}Speaker.

[†]This work has been partially supported by the Galileo 2003 Project PLATONOV and by the INDAM Project 2004 "Controllo e Numerica".

1. Introduction

The Perspective Shape from Shading (PSFS in the sequel) problem has attracted several authors in the last few years. This attention is certainly due to the fact that this problem is more realistic with respect to the classical Shape from Shading (SFS in the sequel) problem, mainly because we assume that the objective of the camera is close to the object instead of being far away (cfr. [13]). This assumption gives rise to a perspective deformation of the object in the image which must be included in the model.

Two models have been recently proposed by Courteille et alia [6] and Prados-Faugeras [16] (see also [15], [18] and [17]). In the first model the objective is close to the object and the light source is supposed to be far away so that the light rays are parallel, in the second model the objective and the light source are located at the same point, close to the image. Both models can be described by a first order partial differential equation of Hamilton-Jacobi type although the equations do not coincide. In this paper we present an approximation scheme for the maximal solution of the first model equation, show that the corresponding fixed point problem has a unique solution and discuss the effect of different types of boundary conditions on the numerical solution. The general framework of this work is the theory of viscosity solutions which can be found in [1]. In the last section we present several tests on real and synthetic images.

2. The model problem and its discretization

Let us define our model problem adopting the same notation used in [6]. The point (X_0, Y_0) is the principal point of the image, d and d' are respectively the distance of the objective from the perspective plane (the film) and the distance of the objective from the (flat) background, l and $l' = \frac{d'}{d}l$ are respectively the length of a segment in the perspective plane (*i.e.* in the image) and the length of the real segment corresponding to it (see Figure 1 and [6] for more details). The

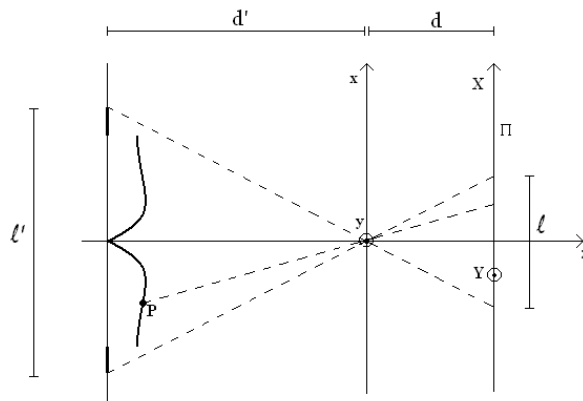


Figure 1: Diagram of the optical lens and of the perspective transformation

representation of the surface in terms of the (X, Y) coordinates of the points in the perspective plane Π is given by three parametric equations

$$x = r(X, Y), y = s(X, Y), z = t(X, Y) \tag{2.1}$$

where (see [6])

$$\begin{cases} r(X, Y) = \frac{X-X_0}{d}t(X, Y) \\ s(X, Y) = \frac{Y-Y_0}{d}t(X, Y) \end{cases} \quad (2.2)$$

Then the problem amounts to compute the third component t . This is the most difficult task since t is the solution of the following eikonal type equation

$$\left[\frac{d}{\bar{t}(X, Y)} \right]^2 |\nabla t(X, Y)|^2 = \frac{I_{max}^2}{I'(X, Y)^2} - 1 \quad \text{in } \Omega \quad (2.3)$$

where Ω is the internal region bounded by the silhouette of the object ($\partial\Omega$ will denote its boundary) which is embedded in a rectangular domain Q ,

$$\bar{t}(X, Y) = t(X, Y) + (X - X_0, Y - Y_0) \cdot \nabla t(X, Y), \quad (2.4)$$

$$I'(X, Y) = \frac{I(X, Y)}{\cos^4 \alpha(X, Y)}, \quad (2.5)$$

$$\cos^4(\alpha(X, Y)) = \frac{d^4}{((X - X_0)^2 + (Y - Y_0)^2 + d^2)^2}, \quad (2.6)$$

and I_{max} is a constant depending on parameters of the problem. The set $Q \setminus \Omega$ is the background.

Defining

$$f(X, Y) \equiv \frac{1}{d^2} \left(\frac{I_{max}^2}{I'(X, Y)^2} - 1 \right) \quad (2.7)$$

we can write (2.3) as

$$|\nabla t(X, Y)| = \sqrt{f(X, Y)} \left| \bar{t}(X, Y) \right|. \quad (2.8)$$

We want to write (2.8) in a fixed point form and construct an approximation scheme for this equation. To this end it is important to note that \bar{t} has a sign. In fact, the exterior normal to the original surface in the point P is given by

$$\hat{n}(P) = N(P)/|N(P)| \quad (2.9)$$

where

$$N(P) \equiv (dt_X(X, Y), dt_Y(X, Y), -\bar{t}(X, Y)) \quad (2.10)$$

and since $-\bar{t}$ must be positive (according to the orientation of the z axis in Figure 1, \bar{t} must be negative. This implies that (2.8) is in fact

$$|\nabla t(X, Y)| + \sqrt{f(X, Y)}(t(X, Y) + (X - X_0, Y - Y_0) \cdot \nabla t(X, Y)) = 0 \quad (2.11)$$

which can be written in short as

$$H((X, Y), t, \nabla t) = 0, \quad \text{in } \Omega \quad (2.12)$$

where the Hamiltonian H represents the left-hand side of (2.11).

Let us consider equation (2.11) complemented with the Dirichlet boundary condition

$$t = g(X, Y) \quad \text{on } \partial\Omega, \quad \text{where } -d' \leq g \leq 0 \quad (2.13)$$

The usual semi-Lagrangian scheme for (2.11)-(2.13) is

$$t(X, Y) = F[t](X, Y) \quad \text{in } \Omega \quad (2.14)$$

where

$$t(X, Y) = F[t](X, Y) \equiv \frac{1}{1+h} \inf_{a \in B(0,1)} \{t(b_h(X, Y, a))\} \quad \text{on } \Omega, \quad (2.15)$$

$$b_h(X, Y, a) = (X, Y) + h \left(\frac{-a}{\sqrt{f}} - (X, Y) \right) \quad (X, Y) \in \Omega, \quad a \in B(0,1) \quad (2.16)$$

and $B(0, 1)$ is the unit ball in \mathbb{R}^2 .

Let us examine the properties of the F operator in order to guarantee convergence for the fixed point iteration. First, let us introduce the following space:

$$W = \{w : \Omega \rightarrow \mathbb{R}, \text{ such that } w|_{\partial\Omega} = g\} \quad (2.17)$$

Note that W is a space of functions satisfying the Dirichlet boundary condition $w = g$ on $\partial\Omega$.

Lemma 2.1. *Under the above assumptions, the following properties hold true:*

- a) F is a contraction mapping in $L^\infty(\Omega)$;
- b) F is monotone, i.e. $s \leq t$ implies $F[s] \leq F[t]$;
- c) Let $V = \{w \in W : -d' \leq w(X, Y) \leq 0\}$, then $F : V \rightarrow V$;

Proof

a) Let us take two functions, t and s . For every $\xi = (X, Y)$, we have

$$\begin{aligned} F[t](\xi) - F[s](\xi) &\leq \frac{1}{1+h} \left[\inf_{a \in B(0,1)} \left\{ t \left(\xi + h \left(\frac{-a}{\sqrt{f}} - \xi \right) \right) \right\} - \inf_{a \in B(0,1)} \left\{ s \left(\xi + h \left(\frac{-a}{\sqrt{f}} - \xi \right) \right) \right\} \right] \leq \\ &\leq \frac{1}{1+h} \left[t \left(\xi + h \left(\frac{-a^*}{\sqrt{f}} - \xi \right) \right) - s \left(\xi + h \left(\frac{-a^*}{\sqrt{f}} - \xi \right) \right) \right] \leq \\ &\leq \frac{1}{1+h} \|t - s\|_\infty \end{aligned} \quad (2.18)$$

where a^* is the direction where the infimum for s is attained. Replacing the role of t and s one obtains the reverse inequality and proves

$$\|F[t] - F[s]\|_\infty \leq \frac{1}{1+h} \|t - s\|_\infty. \quad (2.19)$$

b) The monotonicity with respect to t is a direct consequence of the definition of F since the coefficient in front of the *inf* is strictly positive.

c) It is a direct consequence of b) and of

$$F[0] = 0, \quad F[-d'] = \frac{-d'}{1+h} > -d'. \quad (2.20)$$

■

Now let us examine the algorithm. Lemma 2.1 assures that, starting from any initial guess t^0 which satisfies the boundary conditions, the fixed point iteration

$$t^{n+1} = F[t^n] \quad (2.21)$$

converges to the unique solution t^* (fixed point).

We note that a direct consequence of the above Lemma is that one can obtain a monotone increasing convergence just starting from a sub-solution, *e.g.* choosing $t^0 = -d'$ in the internal nodes and imposing the Dirichlet boundary condition $t^0 = g(X, Y)$ on $\partial\Omega$. Moreover, the property b) guarantees that $\bar{t} < 0$ for all $(X, Y) \in \Omega$ at every iteration, so the equation associated to the problem is always (2.11).

It is well known that the classical SFS problem (corresponding to the orthogonal projection) exhibits a convex/concave ambiguity which corresponds to the fact that the associated eikonal equation has more than one solution (even if we assume regularity). This has motivated a number of contributions (cfr. [19], [14], see also the survey paper [9]) which propose different methods to select a single solution of the equation. Typical recipes to select a solution are, *f.e.*, to fix the heights at points where the light intensity is maximal or to compute the so-called maximal solution (*i.e.* the viscosity solution which is bigger than all the others). It has been shown in [20] that the analogous scheme for the SFS problem converges to the maximal solution of the equation as far as the space discretization step Δx and h satisfy the compatibility condition

$$\frac{Ch}{\Delta x} \leq 1 \quad (2.22)$$

where C is the maximum norm of the vectorfield b_h (see also [11] and [4] for a different scheme). It is worth to note that a similar ambiguity also appears in the PSFS problem, but the counter example for uniqueness is more complicated to describe.

3. Creation of a virtual perspective image and computation

In this section we will describe how we construct a virtual image to test the PSFS algorithm described in the previous section. We developed this procedure to produce some synthetic images that will be used as benchmarks for our algorithm in the last section.

The starting point is the choice of a surface $u = u(x, y)$. Note that all figures refer to the example of a tent ($u(x, y) = 1 - |x|$, $(x, y) \in [-1, 1]^2$) but the procedure is valid for every (graph) surface.

Pre-processing

Given $u = u(x, y)$, we compute (analytically or numerically) the unit normal vector $\hat{n}(P)$ at every point and we compute the light function

$$I(x, y) = \vec{\omega} \cdot \hat{n}(P)$$

where $\vec{\omega}$ is the direction of the source light. In our tests we fixed $\vec{\omega} \equiv (0, 0, 1)$ so we have

$$I(x, y) = \frac{1}{\sqrt{1 + u_x^2 + u_y^2}}$$

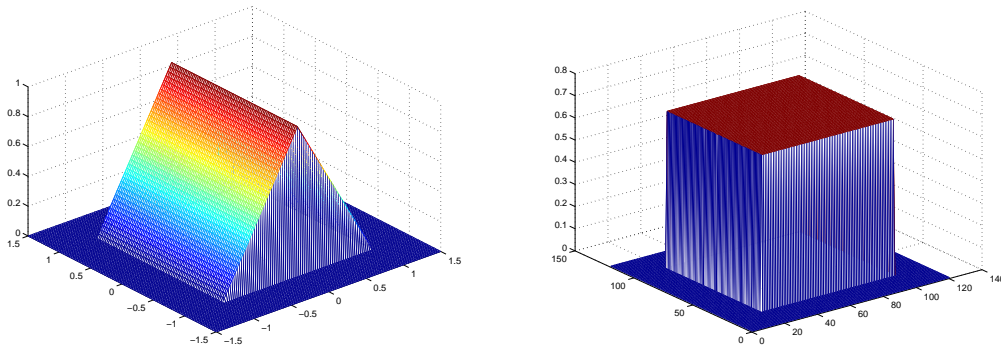


Figure 2: Initial surface $u(x,y)$ (left) and computed light function $I(i', j')$ (right)

We consider a rectangular $n \times n$ grid $G' \equiv \{(x_{i'}, y_{j'})\}$ where $i' = 1, \dots, n$ and $j' = 1, \dots, n$ on which we compute two matrices $I(i', j')$ and $u(i', j')$, $i', j' = 1, \dots, n$ (see figure 2).

Let us define parameters of the experiment $X_0, Y_0, d, d', l, l' = \frac{d'}{d}l$, (see figure 1 and [6]) and then we proceed to "take a photograph" of the surface. Every discrete coordinates (i', j') correspond to a point $(x, y, u(x, y))$ belonging to the surface and every point $(x, y, u(x, y))$ is associated to a pair (X, Y) on the perspective plane by the transformation

$$\begin{cases} \frac{X-X_0}{d} = \frac{x}{u} \\ \frac{Y-Y_0}{d} = \frac{y}{u} \end{cases} \tag{3.1}$$

Varying i', j' in $\{1, \dots, n\}$, we obtain the set

$$\{(X_{i'j'}, Y_{i'j'})\}_{i',j'=1,\dots,n} \tag{3.2}$$

which is the (discrete) domain of the perspective image (see Figure 3).



Figure 3: In white: domain of the perspective image on XY plane. In black: background

Remark 3.1. *The transformation (3.1) is not injective. In particular, a point (X, Y) associated to a point $(x, y, u(x, y))$ belonging to the background can coincide with a point (\tilde{X}, \tilde{Y}) associated to a point $(\tilde{x}, \tilde{y}, \tilde{u}(\tilde{x}, \tilde{y}))$ belonging to the surface. This is the case of the tent as shown in Figure 4.*

This is due to the fact that distance camera-object is finite which implies the existence of some areas in full shade (see Figure 5).

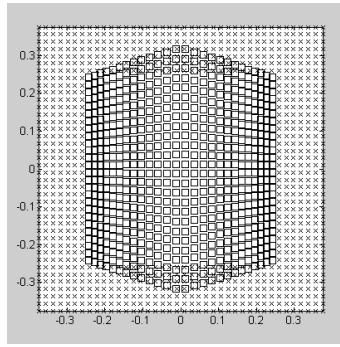


Figure 4: Discrete domain of the perspective image. CROSS: point coming from the background, SQUARE: point coming from the object (there are two overlapping regions)

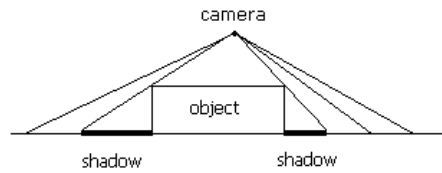


Figure 5: areas in full shade

We want to stress that - as a result of transformation (3.1) - the set (3.2) loses the initial ordering in the sense that two adjacent points $(X_1, Y_1), (X_2, Y_2)$ in the perspective XY -plane are not necessarily coming from two adjacent points $(x_1, y_1, u(x_1, y_1)), (x_2, y_2, u(x_2, y_2))$. Then, the resulting ordering does not coincide with that of a structured grid.

In order to overcome this difficulty and to have an easy implementation of the algorithm, we discretized again the perspective image using a *new* rectangular $n \times n$ grid G indexed by $i = 1 \dots, n$ and $j = 1 \dots, n$. This provides an easy correspondence between (i', j') and (x, y) and between (i, j) and (X, Y) .

By means of a *not* straightforward process we can find the relationship between (i, j) and (i', j') . Roughly speaking, we associate a pair (X, Y) to every node (i, j) and then we find the four closest points to (X, Y) in the set (3.2). These four points are associated to four nodes in the grid G' on which the original light function I and the original surface u are defined. Given this correlation, we easily obtain the matrix representation $I(i, j)$ of the function $I(X, Y)$, that is the perspective image in the XY -plane simply applying an interpolation rule (typically linear interpolation). Moreover, we can compute, in the same way, the function $t(X, Y)$, that is the solution to our problem.

Note that in the case of the tent the computation of $I(X, Y)$ is trivial because $I(x, y)$ is constant. See Figure 6.

Boundary conditions and Computation

The assignment of boundary conditions is very easy and consists in computation of $t(X, Y)$ (as showed above) only in the required nodes. As mentioned before, we choose $t^0 \equiv -d'$ in internal nodes (that is inside Ω). Moreover, it is strongly recommended initializing external nodes (that is

POS (CSTNA2005) 008

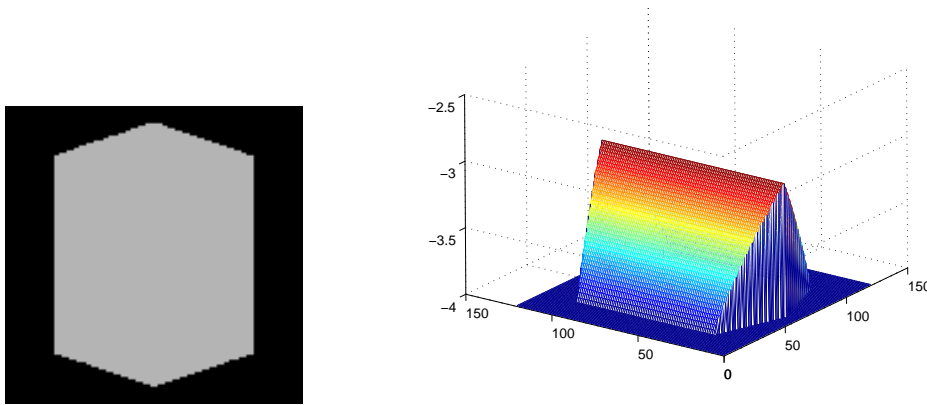


Figure 6: gray level of the photograph (left) and the solution $t(X,Y)$ of equation (right)

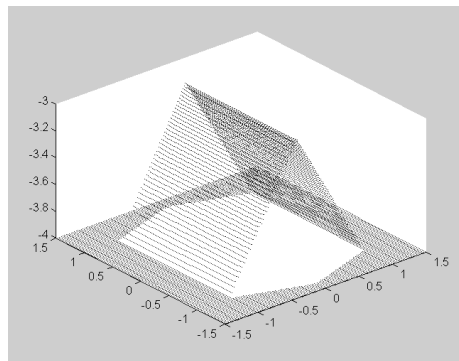


Figure 7: approximate solution (r,s,t) of the problem given in parametric form

outside Ω , where no computation is needed) with the value $t^0 = 0$ which is the greatest value t can attain. This choice avoid the risk that the scheme uses some values coming from the background, because the evaluation of the infimum will automatically reject them.

Now we are ready to compute the solution using the scheme (2.14) and the fixed point technique.

Post-processing

Once we computed the approximate solution $t(X,Y)$, we can easily compute $r(X,Y)$ and $s(X,Y)$ as in (2.2) and then we can draw the surface

$$\{(r(X,Y),s(X,Y),t(X,Y)), \quad (X,Y) \in \Omega\}$$

given in parametric form to be compared with the exact solution in terms of u and I (see Figure 7).

Finally we note that, due to Remark 3.1, the computed surface with its background can have some "holes" in its domain in correspondence with areas in full shade. In other words, in the process we may lose some informations about the initial surface (see Figure 7 and 8). In Figure 8-right black areas correspond to points not visible by the objective.

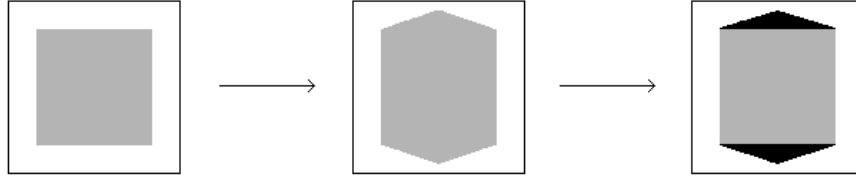


Figure 8: Initial domain of the object (left), perspective domain of the photograph (center) and reconstructed domain of the object (right)

4. Other boundary conditions and their effects

Beside the difficulty related to the concave/convex ambiguity which is behind the non uniqueness of viscosity solutions there is another difficulty which arises in the PDE approach. It is well known that in this approach one has to complement the equation with some boundary conditions to select a unique solution and to run the algorithm. This is a limitation with respect to minimization algorithms where such boundary conditions are not needed and the search for the solution is done via a gradient method or a line search algorithm. Naturally the solution computed by those algorithms will, in general, be different from the exact solution. However, in practical applications boundary conditions on the surface are seldom known, so it useful to analyse in more detail the effect of different types of boundary conditions on the solution in order to define a minimal set of conditions which will allow to compute the exact solution.

In this section, we will briefly analyse the effect of Dirichlet, Neumann and state constraints boundary conditions on subsets of the boundary. Let us note first that boundary conditions should be imposed in a weak sense. The typical condition which defines a viscosity subsolution u for (2.12) requires that for any test function $\varphi \in C^1(\bar{\Omega})$ and $x \in \partial\Omega$ local maximum point for $u - \varphi$

$$\min\{H(x, u(x), D\varphi(x)), B(x, u, D\varphi(x))\} \leq 0 \quad (4.1)$$

where the function B is the operator describing the boundary conditions, *f.e.* $B(x, u, Du) = u - g$ for the Dirichlet condition. Similarly, the boundary condition for supersolutions requires that for any test function $\varphi \in C^1(\bar{\Omega})$ and $x \in \partial\Omega$ local minimum point for $u - \varphi$

$$\max\{H(x, u(x), D\varphi(x)), B(x, u, D\varphi(x))\} \geq 0. \quad (4.2)$$

The effect of the Dirichlet condition is to impose a value on u according to the above conditions, in particular the value $u(x) = g(x)$ is set at every point where $H(x, u(x), D\varphi(x)) \geq 0$ (for subsolutions) and $H(x, u(x), D\varphi(x)) \leq 0$ (for supersolutions).

Neumann boundary conditions correspond to the operator $B(x, u, Du) = \partial u / \partial n(x) - m(x)$ where $n(\cdot)$ represents the outward normal to the domain Ω . A typical use of it is when we know (or we presume) that the level curves of the surface are orthogonal to the boundary $\partial\Omega$ or to a subset of it where we simply choose $m(x) = 0$.

The state constraints boundary condition is different from the above conditions since we do not impose neither a value for u nor a value for its normal derivative $\partial u / \partial n(x)$ (cfr. [3]). In this

respect it has been interpreted as a "no boundary condition" choice although this interpretation is rather sloppy. In fact, a real function u bounded and uniformly continuous is said to be a *state constraints viscosity solution* if and only if it is a subsolution (in the viscosity sense) in Ω and a supersolution in $\bar{\Omega}$ (i.e. up to the boundary). It can be also stated as a Dirichlet boundary condition simply setting

$$g = C_g = \text{constant} \quad \text{provided} \quad C_g > \max_{x \in \Omega} u(x)$$

(note that in our problem, by Lemma 2.1, an easy choice satisfying the above condition is $C_g = 0$). By this choice (4.1) is trivially satisfied, whereas (4.2) requires (strictly)

$$H(x, u(x), D\phi(x)) \geq 0. \quad (4.3)$$

In our algorithm, the fixed point operator F looks for a minimum on neighbouring points $b_h(X, Y, a)$ so we can obtain the same result if we define the solution outside Ω to be $t = C \geq 0$ so that searching for a minimum all directions $a \in B(0, 1)$ will be excluded. The effect of state constraints boundary condition is to look for the minimum inside Ω .

It is interesting to note that if we adopt state constraints boundary conditions and we start from $t^0 \equiv C = \text{constant}$, the scheme will produce the sequence $t^n = (1 + h)^{-n}C$ which converges to 0 everywhere in Ω . Clearly, $t = 0$ is not a meaningful solution. However, if we fix the value at even a single point $x^* \in \Omega$ the solution will have a minimum at x^* and we will have $u(x) > u(x^*)$ for every $x \in \Omega$. So the effect of the state constraints boundary conditions is to produce solutions which increase rapidly when x gets close to the boundary $\partial\Omega$ (see e.g. the tests in [17] where these conditions have been extensively applied).

5. Numerical tests

In this section we present some numerical tests on synthetic images, on a real non perspective image with an artificial perspective deformation and finally on a real photograph with visible perspective deformation.

5.1 Synthetic images

For tests on synthetic images we have chosen in all cases the following parameters:

$$X_0 = 0, \quad Y_0 = 0, \quad d = 1, \quad d' = 4, \quad l = 0.75, \quad l' = 3.$$

The computational procedure follows the steps described in the previous sections. Both G and G' are 121×121 grids and the number of controls for the discretization of the unit ball $B(0, 1)$ is 16 (all placed on the boundary $\partial B(0, 1)$). The iterative algorithm stops when $\|t^{n+1} - t^n\|_\infty \leq \varepsilon$.

In each subsection we present a) the original surface, b) its light function (in the xy -plane), c) the light function in the perspective XY -plane (that is the photograph) and d) the reconstructed surface. Finally, we compute the error estimate in L^∞ -norm comparing the solution t (computed during the preprocessing step) with the approximate solution of the equation (2.11). Note that what we name "solution t " was actually computed by an interpolation so this is not a comparison with the real exact solution t . Nevertheless, this is a very reasonable way to calculate the accuracy of the algorithm

because computation starts from the function $I(i, j)$ which was computed by the same interpolation too.

Remark: we choose a variable step discretization h in (2.15) depending on X, Y and a in such a way that

$$h(X, Y, a) \left(\frac{-a}{\sqrt{f}} - (X, Y) \right) = \Delta x \quad \text{for all } X, Y, a$$

where Δx is the space step discretization. This trick reduces the number of iterations needed to reach convergence. Finally, we define $h_{min} := \min_{X, Y, a} h(X, Y, a)$.

Tent (I discontinuous)

This test is a slight modification of the tent used in Section 3. The main difference here is that I is discontinuous (whereas in the previous example I was constant). The solution t is non regular but the boundary conditions are very simple, 0 on every side of the square. One can see that the algorithm is accurate around the kinks and that the error in the max norm is about $10\Delta x$ (see Table 1).

$$u(x, y) = \begin{cases} 2(1 - |y|) & x \in [-1, 1], \quad y \in [-1, -\frac{1}{2}|x| - \frac{1}{2}] \\ 2(1 - |y|) & x \in [-1, 1], \quad y \in [\frac{1}{2}|x| + \frac{1}{2}, 1] \\ 1 - |x| & \text{otherwise} \end{cases}$$

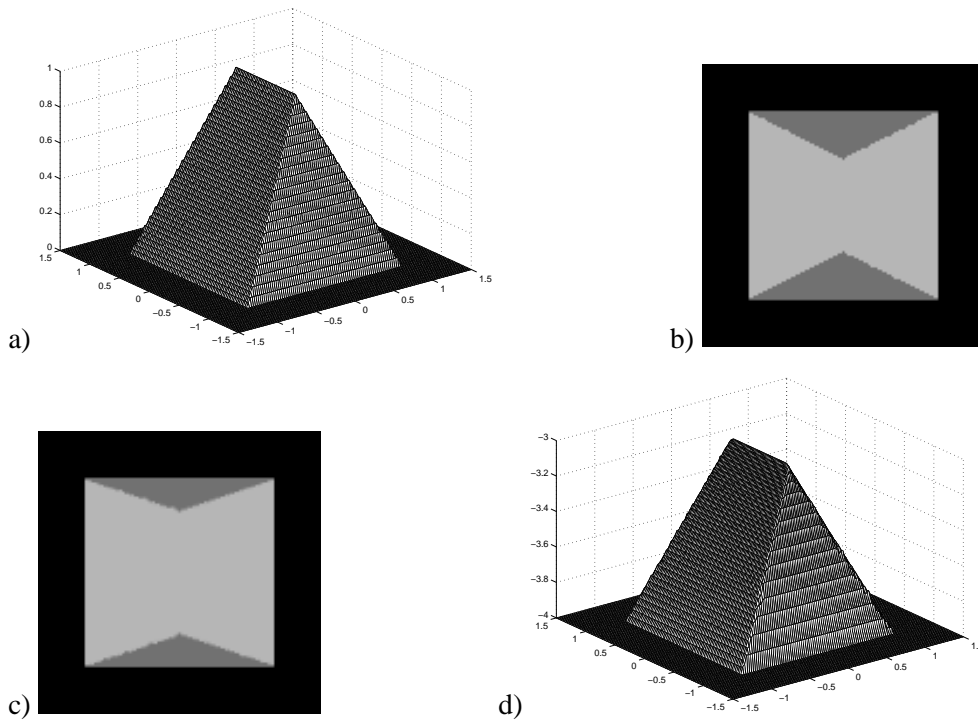


Figure 5.1: a) the original surface, b) its light function in the xy -plane, c) light function in the perspective XY -plane, d) the approximate surface.

POS (CSTNA2005) 008

number of iterations	h_{min}	Δx	ϵ	L^∞ error
52	0.0046	0.00625	10^{-7}	0.064

Table 1: L^∞ error.

Very regular surface (I continuous)

This test has been created to check the accuracy on a regular surface which has a line of singular points (where $I(X,Y) = 1$). This line is truncated in the real computation and substituted by the value 0.9999. The boundary conditions are not homogeneous: they are 0 on the left and right hand sides of the square, and $b(x,y) = 1 - x^2$ on the top and bottom sides of the square. The algorithm stops after 1613 iteration with a $h_{min} = 8.8 \cdot 10^{-5}$. It is interesting to note that also in this case the error is $10\Delta x$.

$$u(x,y) = 1 - x^2, \quad (x,y) \in [-1, 1]^2$$

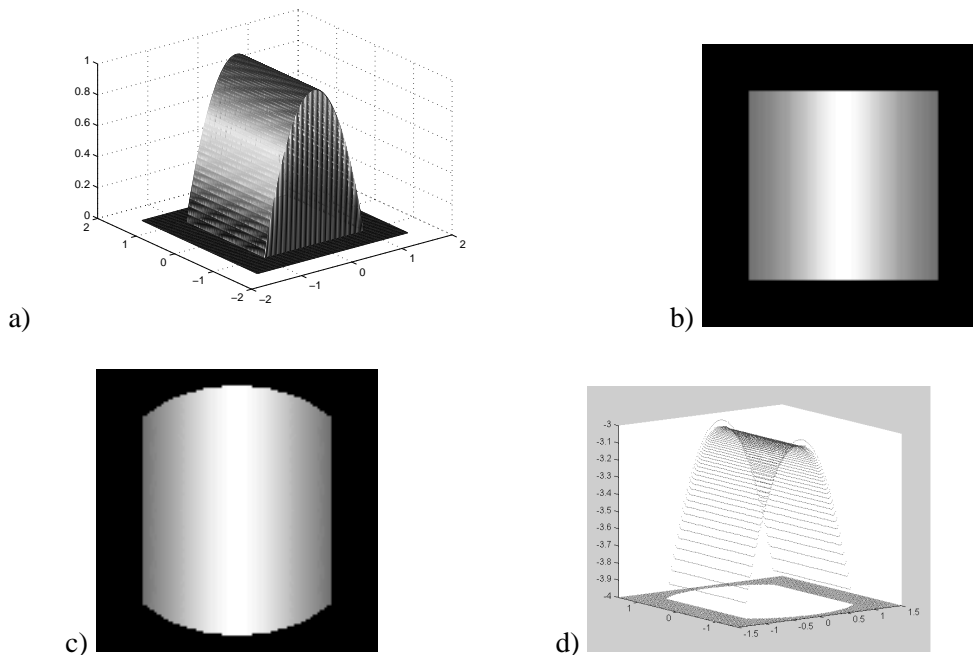


Figure 5.2: a) the original surface, b) its light function in the xy -plane, c) light function in the perspective XY -plane, d) the approximate surface.

number of iterations	h_{min}	Δx	ϵ	L^∞ error
1613	$8.8 \cdot 10^{-5}$	0.00625	10^{-7}	0.0418

Table 2: L^∞ error.

Synthetic book image (I continuous)

The last test is a tentative to reconstruct a synthetic surface as close as possible to the shape of a single page of a book. Again I and u are regular. The boundary conditions are not homogeneous: 0 on the left hand side of the square, a positive constant on the right-hand side and a polynomial function $g(X, Y)$ on the top and bottom sides of the square. Note that $h_{min} = 1.428 \cdot 10^{-4}$ and that the approximate solution after 260 iterations has an error of the order $10\Delta x$.

$$u(x, y) = \frac{1}{120} \left(b(225|x|)^3 + c(225|x|)^2 + d \cdot 225|x| \right)$$

where

$$b = 1.09 \cdot 10^{-5}, \quad c = -6.21 \cdot 10^{-3}, \quad d = 0.883$$

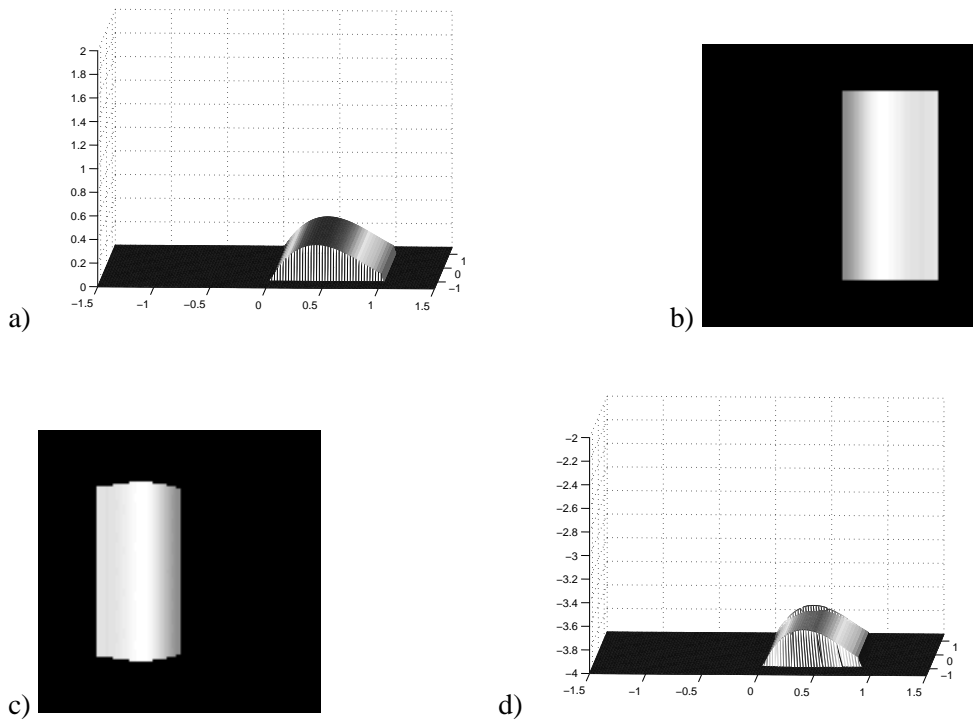


Figure 5.3: a) the original surface, b) its light function in the xy -plane, c) light function in the perspective XY -plane, d) the approximate surface.

number of iterations	h_{min}	Δx	ϵ	L^∞ error
260	$1.428 \cdot 10^{-4}$	0.00625	10^{-7}	0.0536

Table 3: L^∞ error.

POS (CSTNA2005) 008

5.2 Real image with synthetic perspective

In this test we used a *real* photograph of a vase with a negligible perspective deformation, so we modified it by an artificial perspective deformation as in the previous tests.

For this image the parameters values are $d = 1$, $l' = 2$, $d' = 4$, $l = \frac{d}{d'}l' = 0.5$, $\Delta x = 0.0042$ and $I_{max} = 1$.

Figure 9 shows the photograph and the reconstructed surface computed using Dirichlet boundary

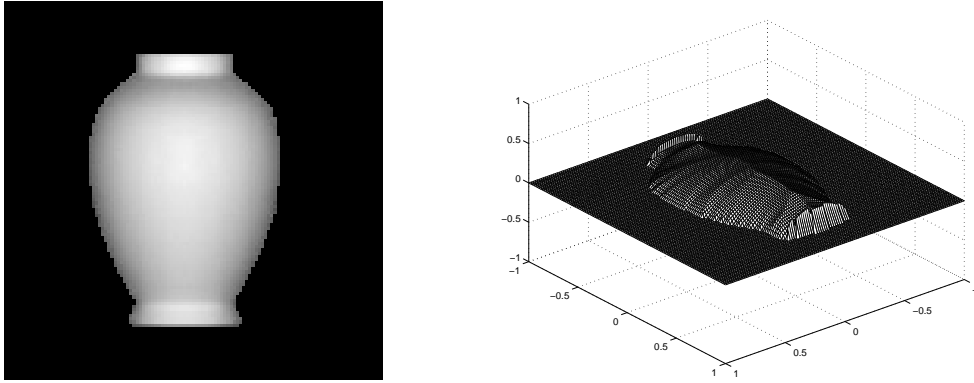


Figure 9: photograph, 121×121 pixels (left) and reconstructed surface with Dirichlet boundary condition (right)

conditions $u = g$, where g is the real height of the vase on $\partial\Omega$.

It is easy to verify that the numerical solution does not match the boundary conditions on the top and on the bottom of the vase (the error is high particularly on the top).

Numerical tests show that in this simple case the values of $t(X, Y)$ inside the domain depend only on its values on the left and right boundaries of the image. Therefore we can substitute Dirichlet boundary condition by state constraints on the top and bottom part of the boundary without any change in the solution. Figure 10-left shows the approximate solution in this case. As expected, the approximate solution is very good even if we do not impose Dirichlet boundary condition on the whole boundary. We want to emphasize that the knowledge of the exact solution t on $\partial\Omega$ can be considered in general a completely *non-realistic* assumption (because the height of the surface is exactly what we want to know) but in this simple case we are able to compute the exact solution also under realistic assumptions.

Finally, we computed the solution with Dirichlet boundary condition on the right and left side and Neumann boundary condition elsewhere. This is "realistic" boundary condition because we can assume that vase is flat on the top and on the bottom. Figure 10-right shows the result.

The average error (with respect to the exact solution) of the three tests is 0.043. In all cases, the iterative procedure converges in 65 iterations, with $\varepsilon = 10^{-7}$.

5.3 Real image

In this test we used a real photograph where the effect of perspective is visible.

The surface is a sheet of paper with the shape of a roof tile. For this image the parameter values are: $l = 6.91mm$, $d = 5.8mm$, $l' = 200mm$, $d' = \frac{l'}{l}d = 167.87mm$, $\Delta x = 0.05mm$. We note that we

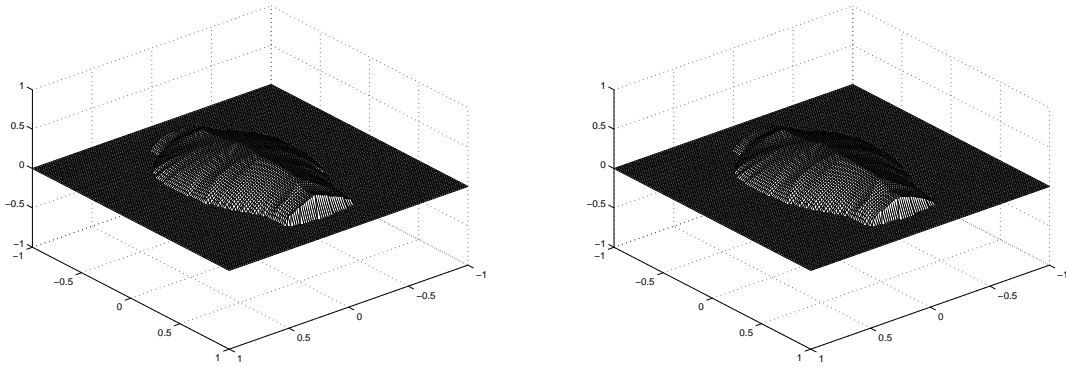


Figure 10: reconstructed surface with Dirichlet and state constraints boundary condition (left) and reconstructed surface with Dirichlet and Neumann boundary condition (right)

performed the light correction (2.5) in the preprocessing step, so we can assume $I_{max} = 1$ during computation. Figure 11 shows the photograph (128×128 pixels) and the surface reconstructed

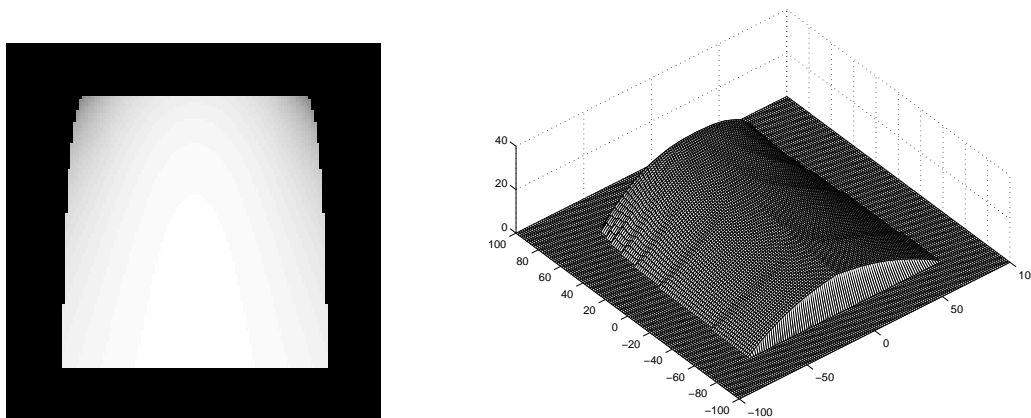


Figure 11: photographs, 128×128 pixels (left) and reconstructed surface with Dirichlet and state constraints boundary condition (right)

using Dirichlet boundary condition only on the left and right sides of the boundary and state constraints elsewhere (top and bottom sides). We can see that the solution is quite good considering the fact that light source (flash camera) is not far from the object and that direction of light source is not perfectly vertical as the mathematical model would have required.

We also tried to reconstruct the surface with two more practical boundary conditions. In the first case, we fixed a Dirichlet condition t^0 only on a vertical line in the center of the image (column 64) and then we turned over the computed surface with respect to the value t^0 (see Figure 12-right). Note that the solution is not very sensitive with respect to value t^0 , so a rough knowledge of the behaviour of the surface can be sufficient. We can see that the solution is quite good. We have a large maximum norm error on the boundary (17.7mm, 41% of the maximum height of the tile), but not inside. In fact, assuming that the reconstructed surface in Figure 11-right is the exact solution,

the average error on all nodes for Figure 12-left is about 1.2mm.

In the second case, we fixed a Dirichlet condition t^0 only on the point (64, 64) (at the center of the image) and then we turned over the computed surface as before. Note that in this case the solution has a shape very different from the expected solution since it has a global maximum at the central point (64, 64).

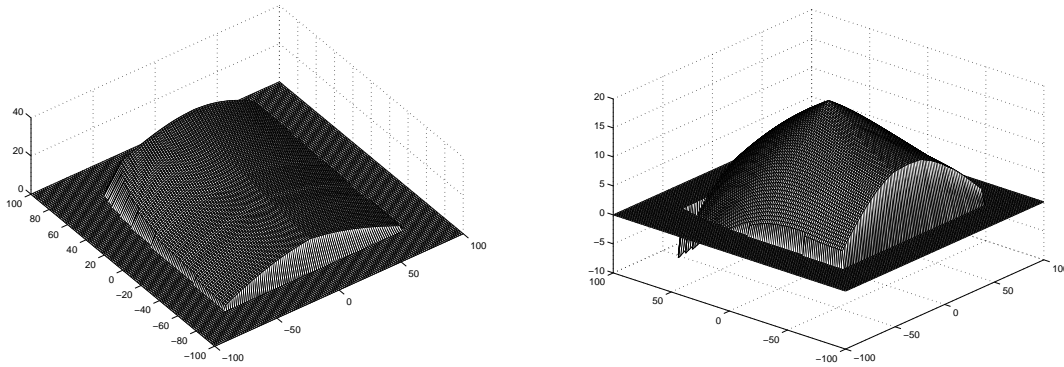


Figure 12: reconstructed surface with Dirichlet boundary condition on the center line (left) and reconstructed surface with Dirichlet boundary condition on one point (right, different scale)

In these three tests the iterative procedure converges respectively in 167, 185 and 190 iterations, with $\varepsilon = 10^{-6}$.

6. Summary and Conclusions

The above experiments show that the scheme proposed in this paper always converges to an approximate solution. The scheme is accurate also for non regular surfaces provided a correct boundary condition is imposed, this means that either we know the value of the height or we know its behaviour at some part of the boundaries (where we impose Neumann or state constraints boundary conditions). The use of state constraints on the whole boundary is risky in this model since it produces nice results for particular surfaces where the optimal vectorfield is pointing strictly inward the domain of computation, but in general this boundary condition will not give a correct solution.

References

- [1] G. Barles, *Viscosity solutions of Hamilton-Jacobi equations*, Springer Verlag, 1994.
- [2] F. Camilli and M. Falcone, *An approximation scheme for the maximal solution of the shape-from-shading model*, Proceedings ICIP 96 *International Conference on Image Processing* vol. I, IEEE Inc., 1996, 49-52.
- [3] F. Camilli and M. Falcone, *Approximation of optimal control problems with state constraints: estimates and applications*, B.S. Mordukhovic, H.J. Sussman eds., “Nonsmooth analysis and geometric methods in deterministic optimal control”, IMA Volumes in Applied Mathematics 78, Springer Verlag, 1996, 23-57.

- [4] F. Camilli and L. Grune, *Numerical approximation of the maximal solution for a class of degenerate Hamilton-Jacobi equations*, SIAM J Num. Anal. **38** (2000), 1540-1560.
- [5] F. Camilli and A. Siconolfi, *Maximal subsolutions for a class of degenerate Hamilton-Jacobi problems*, Indiana Univ. Math. J. **48** (1999), 1111-1131.
- [6] F. Courteille, A. Crouzil, J.D. Durou, P. Gurdjos, *Towards shape from shading under realistic photographic conditions*, *Proceedings of the 17th International Conference on Pattern Recognition - ICPR 2004*, vol.2, Cambridge, 277-280.
- [7] M. G. Crandall, H. Ishii and P. L. Lions, *User's guide to viscosity solutions of second order partial differential equations*, Bull. Amer. Math. Soc., **27** (1992), 1-67.
- [8] J.D. Durou, M. Falcone, M. Sagona, *A survey of numerical methods for Shape-from-Shading*, Rapport de Recherche IRIT n. 2004-2-R, 2004.
- [9] J.D. Durou, D. Piau, *Ambiguous Shape from Shading with critical points*, Journal of Mathematical Imaging and Vision, **12** (2000), 99-108.
- [10] M. Falcone, Ch. Makridakis, *Numerical methods for viscosity solutions and applications*, World Scientific, Singapore, 2001.
- [11] M. Falcone, M. Sagona. *An algorithm for the global solution of the shape-from-shading model*, LNCS 1310 International Conference on Image Analysis and Processing, 1997, 1596-1603.
- [12] M. Falcone, M. Sagona e A. Seghini, *A global algorithm for the Shape-from-Shading problem with black shadows*, in F. Brezzi, A. Buffa, S. Corsaro, A. Murli (eds) "Numerical Mathematics and Advanced Applications" - ENUMATH 2001, Springer Verlag, 503-512.
- [13] B.K.P. Horn and M.J. Brooks, *Shape from Shading*, The MIT Press, 1989.
- [14] P.L. Lions, E. Rouy and A. Tourin, *Shape from shading, viscosity solution and edges*, Numerische Mathematik, **64** (1993), 323-353.
- [15] E. Prados, *Application of the theory of viscosity solutions to the Shape-from-Shading problem*, PhD Thesis, University of Nice - Sophia Antipolis, 2004.
- [16] E. Prados, O. Faugeras, *Perspective Shape-from-Shading and viscosity solutions*, IEEE, Proceedings of ICCV'03, 2003, 826-831.
- [17] E. Prados, O. Faugeras, F. Camilli *Shape-from-Shading: a well posed-problem ?*, Rapport de Recherche n. 5297. INRIA, 2004.
- [18] E. Prados, O. Faugeras, et E. Rouy, *Shape from shading and viscosity solutions*, Rapport de Recherche RR-4638, INRIA, November 2002.
- [19] E. Rouy and A. Tourin, *A viscosity solutions approach to Shape from Shading*, SIAM J. Numer. Anal., **29** (1992), 867-884.
- [20] M. Sagona, *Numerical methods for degenerate eikonal type equations and applications*, PhD Thesis, Università di Napoli "Federico II", 2001.
- [21] M. Sagona e A. Seghini, *An adaptive scheme on unstructured grids for the shape-from shading problem*, in [10], 197-219.