A Multiresolution Approach for Shape from Shading Coupling Deterministic and Stochastic Optimization

Alain Crouzil, Xavier Descombes, and Jean-Denis Durou

Abstract—Shape from shading is an ill-posed inverse problem for which there is no completely satisfactory solution in the existing literature. In this paper, we address shape from shading as an energy minimization problem. We first show that the deterministic approach provides efficient algorithms in terms of CPU time, but reaches its limits since the energy associated with shape from shading can contain multiple deep local minima. We derive an alternative stochastic approach using simulated annealing. The obtained results strongly outperform the results of the deterministic approach. The shortcoming is an extreme slowness of the optimization. Therefore, we propose a hybrid approach which combines the deterministic and stochastic approaches in a multiresolution framework.

Index Terms—Shape from shading, optimization, simulated annealing, multiresolution.

1 Introduction

SHAPE from shading (SFS) is a well-known problem in computer vision [1] that consists of recovering the 3D-shape of a scene through the analysis of the gray levels in a single image of this scene. This problem is ill-posed except if some additional assumptions, which are recalled in [2], are made.

If the visible part of a scene can be described by the equation $z = h(x, y)$, where Oz coincides with the optical axis, then, at each point (x, y) in the image where h is derivable, the basic equation of SFS is the "image irradiance equation" [3]:

$$R(\overrightarrow{S}, p(x,y), q(x,y)) = E(x,y), \tag{1}$$

where E designates the gray level, R the reflectance map, $\overrightarrow{S}$ the lighting beam, and p and q are the usual notations for $\partial h/\partial x$ and $\partial h/\partial y$. In (1), the unknowns are $\overrightarrow{S}$, p, and q.

The methods of resolution of SFS can be classified into three groups: the classical methods of resolution of PDEs (characteristic strips expansion [3], level sets methods [4], power series expansion [5], and viscosity solutions [6], [7], [8]); the methods based on optimization, on which we will focus further on; all other methods, among which the method by Tsai and Shah [9] gives rather good results [10] and is surprisingly simple to implement. In their survey, Zhang et al. [11] concluded that the best results, in terms of accuracy, were given by the optimization methods, although they are also the slowest. In this paper, we address optimization also because it offers a general framework that is very adaptable to various situations. Indeed, the problem is modeled by a functional which has to be minimized [12], defined by the sum of several terms. The first term reflects the data and is minimal when (1) is satisfied everywhere on the reconstruction domain Ω. For the problem to be well-posed, some extra terms, reflecting some a priori knowledge of the solution, are added. A certain number of local deterministic algorithms have been proposed to minimize various functionals [13], [14], [15], [16], [17], [18]. In this paper, we obtain more costly but global solutions by a simulated annealing (SA) scheme.

If, as most authors, we suppose that $\overrightarrow{S}$ is known, thanks to preprocessing by a specific method, then we can write $R(p(x,y), q(x,y))$. Using the excellent discussion on the variational formulation of SFS made in [12], we use the following functional:

$$\begin{aligned}\mathcal{F}_1(p,q) = &\iint_{(x,y)\in\Omega} \left[R(p(x,y),q(x,y)) - E(x,y)\right]^2 dx\,dy \\ &+ \lambda_{\text{int}} \iint_{(x,y)\in\Omega} \left[\partial p/\partial y(x,y) - \partial q/\partial x(x,y)\right]^2 dx\,dy \\ &+ \lambda_{\text{smo}} \iint_{(x,y)\in\Omega} \left[|\nabla p(x,y)|^2 + |\nabla q(x,y)|^2\right] dx\,dy,\end{aligned} \tag{2}$$

where the unknowns are the functions p and q and where λ_{int} and λ_{smo} are two positive constants named "integrability factor" and "smoothing factor." Besides the problem of selecting adequate values for the constants λ_{int} and λ_{smo}, the choice of (p, q) as unknowns poses another important problem: that of the computation of h from (p, q). The equations linking these functions being $\partial h/\partial x = p$ and $\partial h/\partial y = q$, we will then use, in a second stage, the following functional proposed by Horn and Brooks in [12]:

$$\begin{aligned}\mathcal{F}_2(h) = \iint_{(x,y)\in\Omega} &\Big[\left(\partial h/\partial x(x,y) - p(x,y)\right)^2 \\ &+ \left(\partial h/\partial y(x,y) - q(x,y)\right)^2\Big]\, dx\,dy.\end{aligned} \tag{3}$$

The two main strategies allowing us to numerically find the minimum of a functional are [16]: on the one hand, to directly minimize the discrete formulation of the functional, named an "energy" and denoted by ϵ; on the other hand, to produce one Euler equation for each unknown function (a discrete version of the Euler equations can be obtained either by discretizing them or by differentiating the energy, as shown in [12]). It is surprising to find out that most of the algorithms of SFS use the second choice. As quoted by Szeliski [16], solving the discrete Euler equations is not strictly equivalent to minimizing the energy, since all local minima and maxima (and inflexion points) of the energy are solutions of these equations. Moreover, when the Euler equations are non linear (as in SFS), they have to be solved iteratively and there exists no general proof of convergence for the produced schemes (proofs of divergence have even been given for two such methods in [19]). Finally, a third important advantage in directly minimizing the energy is that no knowledge of h, p, or q on the boundary of Ω is necessary (a discussion on this point can be found in [2], where it is shown why boundary conditions are not necessary to make the problem well-posed, apart from the classical concave/convex ambiguity). Two kinds of methods of minimization can be distinguished:

- The first family of optimization methods are the deterministic methods. At each step k of the iteration, the computation of the new configuration ω^{k+1} uses the gradient of ϵ at the step k (and, sometimes, at the steps $0, \cdots, k-1$) and needs, most of the time, the search for a minimizer d^k of the function $\phi_k(d) = \epsilon(\omega^k - d\,v^k)$, where v^k is a "descent direction" (this task is named "line search"). The conjugate gradient descent, which is used by Szeliski in [16], provides good results. The first method that we propose in this paper, named **M1**, uses, on the one hand, the very classical optimal gradient descent and, on the other hand, a method of line search based on parabolic interpolation (see [2]).

- *A. Crouzil and J.-D. Durou are with the Institut de Recherche en Informatique de Toulouse, Université Paul Sabatier, 118 route de Narbonne, 31062 Toulouse Cedex, France. E-mail: {crouzil, durou}@irit.fr.*
- *X. Descombes is with Ariana, joint research group CNRS/INRIA/UNSA, INRIA, 2004 route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex, France. E-mail: Xavier.Descombes@sophia.inria.fr.*

Manuscript received 2 May 2002; revised 1 Feb. 2003; accepted 1 Apr. 2003. Recommended for acceptance by M.A.T. Figueiredo, E.R. Hancock, M. Pelillo, and J. Zerubia.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 118623.

- The second family of optimization methods are the stochastic methods, which allow us to find a global minimum of an energy, contrary to the other methods. In Section 3, we will use the SA algorithm (method **M2**).

In Section 2, we present a first new method, named **M1**, for which some results are shown. In Section 3, we embed the SFS problem in a Bayesian framework and present an SA scheme, named **M2**, which is shown to work better than **M1** on a complex image. In Section 4, we show that the main problem with **M2**, i.e., slowness, can be diminished by using a multiresolution version of **M1**, after having applied **M2** to the image of smallest resolution. This leads to a third new method, named **M3**. Conclusion and future work are covered in Section 5.

2 Deterministic Minimization for Shape from Shading

From now on, we suppose that $\overrightarrow{S} = (0\,0\,1)^T$ and that the surface is Lambertian, so that the image irradiance equation (1) takes the particular form called the "eikonal equation" [20], whose discrete form is:

$$\frac{E_{\max}}{\sqrt{1 + {p_{i,j}}^2 + {q_{i,j}}^2}} = E_{i,j}, \tag{4}$$

where $E_{\max}$ is the maximal value of E (this value is reached at points where $\nabla h = 0$). For the synthetic images that will be used in the tests, the gray levels will be computed from (1), with $E_{\max} = 255$. Consider the distance $\delta = 0.1$ between two neighboring pixels and denote as D the discrete domain corresponding to Ω. Using forward finite differences, the energy ϵ_1 corresponding to the functional $\mathcal{F}_1$ is:

$$\begin{aligned}\epsilon_1(\omega_1) = \delta^2 &\sum_{(i,j)\in D}\left(\frac{E_{\max}}{\sqrt{1 + {p_{i,j}}^2 + {q_{i,j}}^2}} - E_{i,j}\right)^2 \\ &+ \lambda_{\text{int}} \sum_{(i,j)\in \widetilde{D}} \left[(p_{i,j+1} - p_{i,j}) - (q_{i+1,j} - q_{i,j})\right]^2 \\ &+ \lambda_{\text{smo}} \sum_{(i,j)\in \widetilde{D}} \Big[(p_{i+1,j} - p_{i,j})^2 + (p_{i,j+1} - p_{i,j})^2 \\ &\qquad + (q_{i+1,j} - q_{i,j})^2 + (q_{i,j+1} - q_{i,j})^2\Big],\end{aligned} \tag{5}$$

where $\omega_1 = (p_{i,j}, q_{i,j})_{(i,j)\in D}$ and where $\widetilde{D}$ is the subset of D containing the pixels (i,j) such that $(i+1,j)$ and $(i,j+1)$ are in D. It has been proven in [12] that this energy is essentially independent of the image resolution, which will be of importance when we will deal with multiresolution. In the same way, the energy ϵ_2 associated with $\mathcal{F}_2$ is:

$$\epsilon_2(\omega_2) = \sum_{(i,j)\in \widetilde{D}} \Big[(h_{i+1,j} - h_{i,j} - \delta\, p_{i,j})^2 + (h_{i,j+1} - h_{i,j} - \delta\, q_{i,j})^2\Big], \tag{6}$$

where $\omega_2 = (h_{i,j})_{(i,j)\in D}$.

Given an initial configuration $\omega_1^0 = (p_{i,j}^0, q_{i,j}^0)_{(i,j)\in D}$, the first stage of our first method, named **M1**, is an iterative process defined by the following steps:

1. Compute $\nabla\epsilon_1$ at the current configuration ω_1^k.
2. Line search for a local minimizer d^k of the function $\phi_k(d) = \epsilon_1(\omega_1^k - d\,\nabla\epsilon_1(\omega_1^k))$.
3. Compute the new configuration $\omega_1^{k+1} = \omega_1^k - d^k\,\nabla\epsilon_1(\omega_1^k)$.

The iteration is stopped at the first configuration $\omega_1^\star$ for which $|\nabla\epsilon_1(\omega_1^\star)| < \beta\sqrt{2N}$ (in our tests, β is equal to 1.0). The factor $\sqrt{2N}$, with $N = \text{card}(D)$, means that the norm of $\nabla\epsilon_1$ is roughly proportional to the square root of its number of coordinates. Given $\omega_1^\star$ and an initial configuration $\omega_2^0 = (h_{i,j}^0)_{(i,j)\in D}$, the second stage of **M1** is an iterative process which is, in all points, similar to the previous one. The first configuration $\omega_2^\star$ verifying $|\nabla\epsilon_2(\omega_2^\star)| < \beta\sqrt{N}$ is the final result of the method. Now, let us comment on the tests.

First of all, it must be said that all our tests are performed without any boundary condition (see [2] for further details). We tested **M1** on three synthetic images of size 64×64, represented in Figs. 1b, 2b, and 3b. For each image, the domain D contains all the pixels. However, for Fig. 2b, we will see that **M1** is improved if D is shared into two parts: one part containing the pixels of the vase and the other the pixels of the background.

Fig. 1b corresponds to the simple shape represented in Fig. 1a. The values used for the factors λ_{int} and λ_{smo} are, respectively, equal to 10 and to 50. Starting from the initial configurations ω_1^0 and ω_2^0 corresponding to the shape in Fig. 1c, the method **M1** converges toward the shape represented in Fig. 1d in about 3 sec. of CPU time on a 733 MHz PC. The L_2-distance between Fig. 1d and the real shape Fig. 1a, using the same scale as on the z axis, is equal to 0.015.

Consider now the more complex image shown in Fig. 2b, which corresponds to a vase on a background (Fig. 2a). The values used for the factors λ_{int} and λ_{smo} are the same as previously. Starting from the shape in Fig. 1c, the solution is represented in Fig. 2c. It is not as satisfactory as the previous result. Indeed, compared to Fig. 1a, the shape in Fig. 2a is much more complex because there are sharp edges at the top and at the bottom of the vase, as well as silhouettes on both sides. When a pixel (i,j) is situated on the background and $(i+1,j)$ is situated on the vase, it makes no sense to keep terms like $(p_{i+1,j} - p_{i,j})^2$ or $(h_{i+1,j} - h_{i,j} - \delta\, p_{i,j})^2$ in the energies ϵ_1 or ϵ_2 because h, p, and q are not differentiable there. To overcome this difficulty, we remove, in ϵ_1 and in ϵ_2, the terms which simultaneously imply pixels lying on the background and pixels lying on the vase. Doing so, each new energy can be separated into two parts: $\epsilon(\omega) = \epsilon^{\text{back}}(\omega^{\text{back}}) + \epsilon^{\text{vase}}(\omega^{\text{vase}})$, where ω^{back} contains the unknowns for the background pixels and ω^{vase} the unknowns for the vase pixels so that $\omega = (\omega^{\text{back}}, \omega^{\text{vase}})$. Therefore, the minimization of ϵ can be separated into two independent problems: the minimization of ϵ^{back}, and that of ϵ^{vase}. The only additional problem with this way of processing is that ω_2^{back} and ω_2^{vase} will be known only up to two independent constants, so we will have to join the two reconstructed shapes (see [2]). The result obtained with this new version of **M1** is shown in Fig. 2d, a result which is more satisfactory. The CPU time is about 20 sec. and the L_2-distance between Fig. 2d and the real shape Fig. 2a is equal to 0.761.

Now, let us study a much more complex shape, named DEM and represented in Fig. 3a, associated to the image shown in Fig. 3b. The values of the factors λ_{int} and λ_{smo} are now equal, respectively, to 500 and to 20 (the problem consisting of choosing "good values" for these factors is left out for future work). Starting from a shape "similar" to the real shape, represented in Fig. 3c, **M1** yields Fig. 3d as a solution, which is a good result. But, starting from Fig. 3e, the result obtained (see Fig. 3f) is not satisfactory at all, even from a qualitative point of view. This last result shows that **M1** has a crucial weakness: The choice of the starting shape is of great importance (compare Fig. 3d and Fig. 3f), as ever with deterministic minimization.

Now, we address SA to overcome this major problem.

3 A Bayesian Approach for Shape from Shading

Hereafter, we reformulate the SFS problem into a Bayesian framework. Let E denote the data, $S \subset \mathbb{Z}^2$ the lattice (set of sites), and $\Lambda \subset \mathbb{R}^2$ the set of states. A configuration in $\Lambda^{\text{card(S)}}$ is denoted by ω and the state of the site $s = (i,j)$ by $\omega_s = (p_s, q_s)$. We want to find the configuration which maximizes the a posteriori probability $P(\omega|E)$. Using the Bayes rule, we have:

$$P(\omega|E) = \frac{P(E|\omega)P(\omega)}{P(E)} \propto P(E|\omega)P(\omega). \tag{7}$$

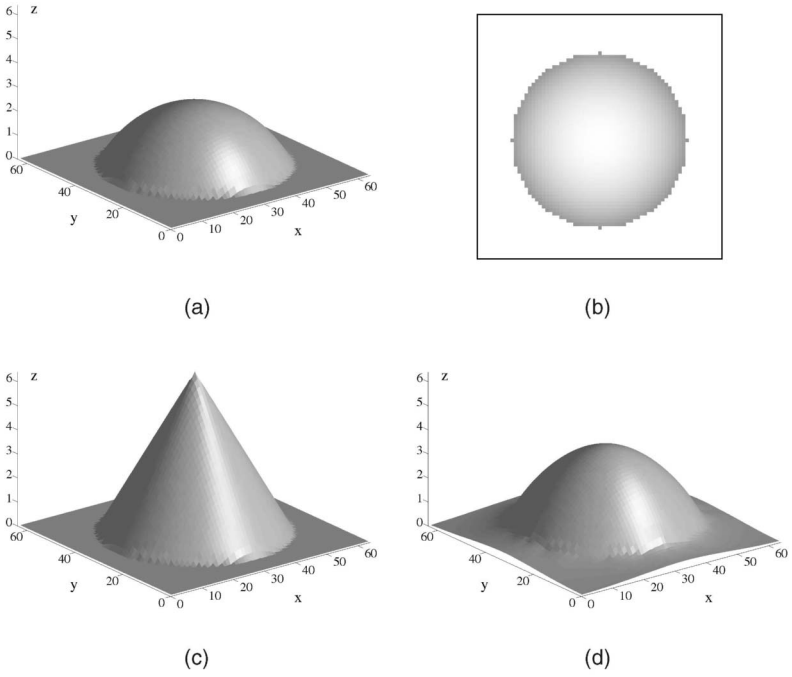


Fig. 1. (a) Spherical cap on a background and (b) associated image. (c) Starting shape and (d) result by $\mathbf{M1}$ ($\lambda_{\text{int}} = 10$; $\lambda_{\text{smo}} = 50$).

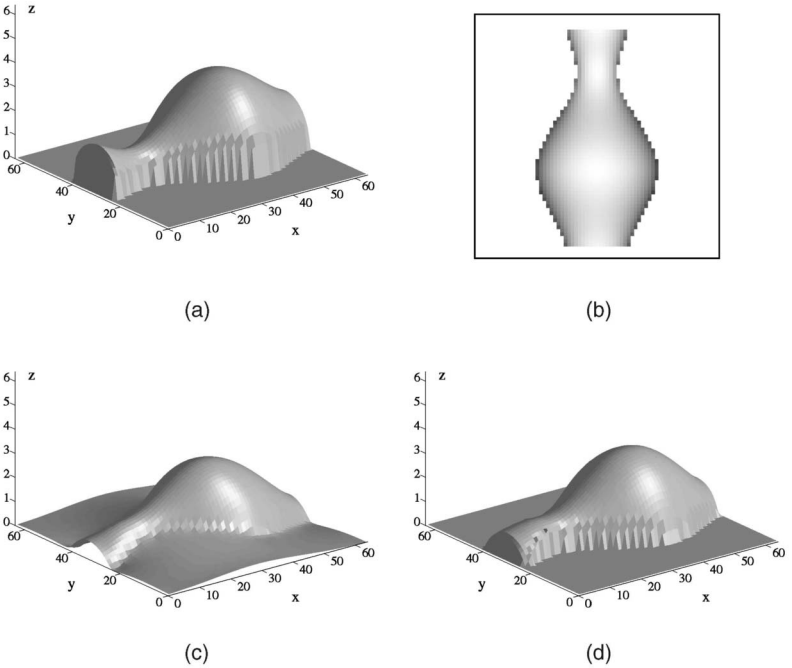


Fig. 2. (a) Vase on a background and (b) associated image. $\mathbf{M1}$ applied (c) to the whole image and (d) to two subparts ($\lambda_{\text{int}} = 10$; $\lambda_{\text{smo}} = 50$).

The first factor of the right-hand side is the likelihood and represents the information provided by the data. The second factor, $P(\omega)$, is referred to as the prior. This probability distribution embeds some a priori properties, such as smoothness, which constrain the solution. The Markov Random Fields (MRF) are common and efficient models in this framework [21]. MRF are discrete stochastic processes whose global properties are controlled by means of local properties [22]. The Hammersley-Clifford theorem [22] allows us to write an MRF as a Gibbs field:

$$P(\omega) = \frac{1}{Z}\exp[-U(\omega)] = \frac{1}{Z}\exp\left[-\sum_{c\in\mathcal{C}} V_c(\omega_s, s \in c)\right], \qquad (8)$$

where U is the energy, V_c is a function from $\Lambda^{\text{card}(c)}$ onto $\mathbb{R}^+$, and Z is the partition function. Here, V_c refers to the potential associated with the clique c, which is a finite subset of S, and C is the set of cliques.

We derive a Markov Random Field, adapted to the SFS problem, which contains four factors, $P(E|\omega)P_1(\omega)P_2(\omega)P_3(\omega)$. The three last factors define the prior and, respectively, define a smoothing constraint, an integrability constraint, and a prior on the (p, q) distribution.

In the previous section, we have derived an energy ϵ_1 (see (5)) modeling the SFS problem (in fact, the first stage of the problem). This energy embeds the eikonal equation, the integrability constraint, and a smoothing constraint. Besides, this energy can be written as a sum of local functions which depend only on

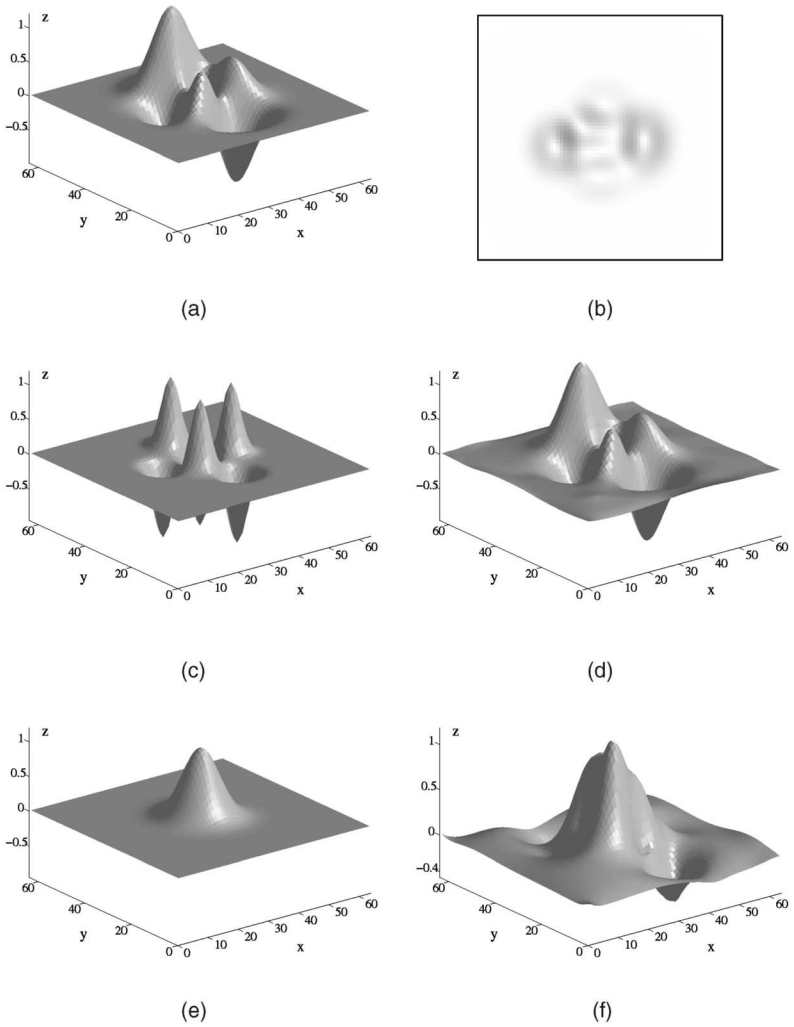


Fig. 3. (a) DEM and (b) associated image. (c) First starting shape and (d) result by $\mathbf{M1}$ ($\lambda_{\text{int}} = 500$; $\lambda_{\text{smo}} = 20$). (e) Second starting shape and (f) result by $\mathbf{M1}$ ($\lambda_{\text{int}} = 500$; $\lambda_{\text{smo}} = 20$).

neighboring pixels. Therefore, ϵ_1 can be considered as the energy associated with a Gibbs field. We thus propose to write the three first factors as follows:

$$P(E|\omega)P_1(\omega)P_2(\omega) = \frac{1}{Z}\exp[-\epsilon_1(\omega)]. \tag{9}$$

To define the prior on the (p, q) distribution, we consider the sites to be independent with respect to the factor $P_3(\omega)$ and we assume that the normal to the surface is uniformly distributed on the Northern hemisphere of the Gaussian sphere. This induces the following (see [2]):

$$P_3(\omega) = \prod_{s\in S} \pi_3(\omega_s) \propto \prod_{s\in S} \frac{\rho_s}{(1+\rho_s{}^2)^{3/2}}, \tag{10}$$

where $\pi_3(\omega_s)$ is the prior probability density of ω_s and $\rho_s = \sqrt{p_s{}^2 + q_s{}^2}$. We obtain the Maximum A Posteriori (MAP) by minimizing the so-called energy which, if we assume a conditional independency of the likelihood, is given by the following functional:

$$U(\omega) = \sum_{s\in S} -\log(\pi(E_s|\omega_s)) + \sum_{c\in\mathcal{C}} V_c(\omega_s, s \in c), \tag{11}$$

where π denotes the local likelihood. This energy is usually not convex. To obtain the MAP criterion, we run an SA algorithm which escapes from the local minima [23]. It is expressed as follows:

1. Initialize a random configuration $\omega^0 = (\omega_s^0,\ s \in S)$, set $T = T_0$ and $k = 1$.
2. For each site s (the sites being empirically numbered from 1 to $\text{card}(S)$):
   a. Choose a random value *new* in Λ following a proposal distribution $Q(\omega_s^k = cur \to \omega_s^{k+1} = new)$, where *cur* is the current state of s and *new* is the proposed state.
   b. Compute the acceptation ratio:

$$R = \left(\frac{P(\omega^{new})}{P(\omega^{cur})}\right)^{1/T_k} \frac{Q(\omega_s^k = new \to \omega_s^{k+1} = cur)}{Q(\omega_s^k = cur \to \omega_s^{k+1} = new)}, \tag{12}$$

   where

$$\omega^{\text{value}} = \left(\omega_1^{k+1}, \cdots, \omega_{s-1}^{k+1}, \omega_s = \text{value}, \omega_{s+1}^k, \cdots, \omega_{\text{card}(S)}^k\right),$$

   for $\text{value} = cur, new$.
   c. Accept the proposition with probability $\min(1, R)$. If the proposition is accepted, set ω_s^{k+1} to *new*, else set ω_s^{k+1} to *cur*.

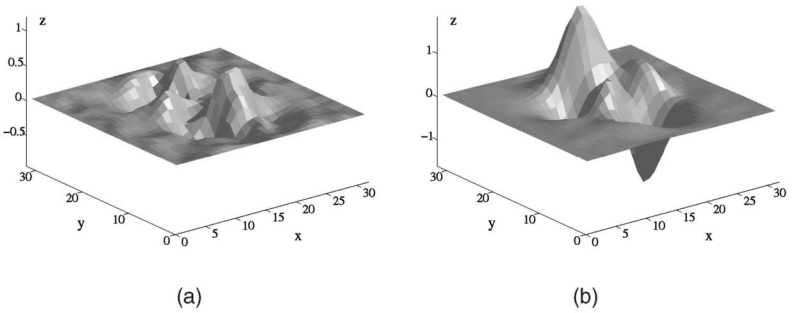


Fig. 4. M2 applied to image Fig. 3b: (a) $\alpha = 0.99$ and (b) $\alpha = 0.999998$ ($\lambda_{\text{int}} = 500$; $\lambda_{\text{smo}} = 20$).

3. If the stopping criterion is not reached, decrease the temperature $T_{k+1} = f_{\text{dec}}(T_0, k)$, increment k, and go to 2.

To reduce the CPU time of the optimization, we have to reduce the time necessary to simulate the proposal and the number of steps necessary to reach the convergence. To reduce the number of steps, we have to consider a proposal similar to the considered model. In the case of a model with interactions, this leads to a proposal that takes a long time to simulate. If we consider the data values in the proposal, we have to compute a different proposal at each site. Therefore, we propose considering only the noninteracting factor of the prior model in the proposal distribution.

The proposal defined in the SA algorithm is as follows:

$$Q(\omega_s^k = cur \rightarrow \omega_s^{k+1} = new) = Q(\omega_s^{k+1} = new) = \pi_3(new). \quad (13)$$

Denote

$$\begin{aligned}\omega\backslash\{s\} &= \left(\omega_1^{new}, \cdots, \omega_{s-1}^{new}, \omega_{s+1}^{new}, \cdots, \omega_{\text{card}(S)}^{new}\right)\\ &= \left(\omega_1^{cur}, \cdots, \omega_{s-1}^{cur}, \omega_{s+1}^{cur}, \cdots, \omega_{\text{card}(S)}^{cur}\right).\end{aligned}$$

The acceptation ratio is then given by:

$$R = \left(\frac{\pi(E_s|\omega_s^{new})P_1(\omega_s^{new}|\omega\backslash\{s\})P_2(\omega_s^{new}|\omega\backslash\{s\})}{\pi(E_s|\omega_s^{cur})P_1(\omega_s^{cur}|\omega\backslash\{s\})P_2(\omega_s^{cur}|\omega\backslash\{s\})}\right)^{1/T}\left(\frac{\pi_3(new)}{\pi_3(cur)}\right)^{1/T-1}, \quad (14)$$

where $\pi(E_s|\omega_s) \propto \exp\left[-\delta^2\left(E_{\max}/\sqrt{1+{p_s}^2+{q_s}^2} - E_s\right)^2\right]$. So, using the Markov property, we have to compute:

$$R = \exp\left[-\frac{F(p_s^{new}, q_s^{new}) - F(p_s^{cur}, q_s^{cur})}{T}\right]\left(\frac{\rho_s^{new}(1+(\rho_s^{cur})^2)^{3/2}}{\rho_s^{cur}(1+(\rho_s^{new})^2)^{3/2}}\right)^{1/T-1} \quad (15)$$

with:

$$\begin{aligned}F(p_s, q_s) &= \delta^2\left(\frac{E_{\max}}{\sqrt{1+{p_s}^2+{q_s}^2}} - E_s\right)^2 + \lambda_{\text{smo}}\\ &\sum_{s'\in\{s_1,s_2,s_3,s_4\}}\left[(p_{s'}-p_s)^2 + (q_{s'}-q_s)^2\right] + \lambda_{\text{int}}\left\{\left[(p_{s_1}-p_s)-(q_{s_2}-q_s)\right]^2\right.\\ &\left. + \left[(p_s-p_{s_3})-(q_{s_6}-q_{s_3})\right]^2 + \left[(p_{s_5}-p_{s_4})-(q_s-q_{s_4})\right]^2\right\},\end{aligned}$$

where $s_1 = s+(0,1)$, $s_2 = s+(1,0)$, $s_3 = s+(0,-1)$, $s_4 = s+(-1,0)$, $s_5 = s+(-1,1)$, $s_6 = s+(1,-1)$.

We show results on the DEM for which the method **M1** fails. The cooling scheme $T_k = f_{\text{dec}}(T_0, k)$, which should be of the form $T_k = \frac{T_0}{\log(k+1)}$ (see [23]), is crucial in an SA scheme. In practice, a geometrical scheme $T_k = \alpha^k T_0$ is often used to speed up the convergence. In image segmentation or image restoration, the values of α reported in the literature lie between 0.95 and 0.99. The tests that we performed have revealed that this cooling scheme is too fast for the SFS problem. Using this, the SA algorithm converges to a local minimum that is far from the actual surface (see the result obtained with $\alpha = 0.99$ in Fig. 4a). To actually converge to the global minimum, we have to use $\alpha = 0.999998$, which leads to 6.10^6 iterations. The result is shown in Fig. 4b, using a plane as initial configuration. We have obtained similar results using a random configuration as initialization. Because of the stochastic perturbations during the iterative scheme, the configuration has escaped from the local minima. The result is very close to the true surface.[1] It has been obtained on an image of the DEM of size 32×32 and has required about one hour of CPU time. The L_2-distance between Fig. 4b and the real shape Fig. 3a is equal to 0.353.

4 A Multiresolution Hybrid Method

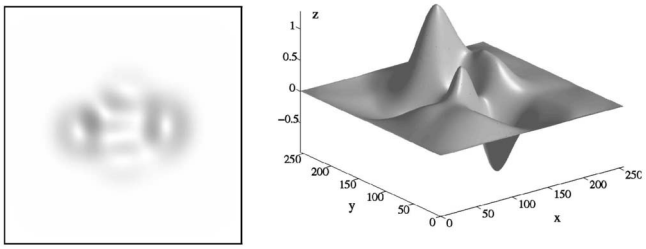
To handle images having a more realistic size than 32×32, it is necessary to speed up the convergence. The SA is helpful to escape from the local minima, but, if the configuration is in the region of attraction of the global minimum, method **M1** is much more efficient. This motivates a multiresolution approach, **M3**. The idea is to run an SA at a low resolution to obtain the global minimum at this resolution. We then compute **M1** at higher resolutions, taking the previous result as initial configuration. Therefore, we assume that the global minimum at low resolution belongs to the region of attraction of the global minimum when projected at higher resolutions.

Multiresolution has already been used in the context of shape from shading [24], [16], [25] in order to accelerate the convergence of classical algorithms. Multiresolution techniques in image analysis are generally based on the construction of an image pyramid. For each level, the image is obtained by blurring, then subsampling the image of the previous level. If the low pass filter used is the Gaussian filter, the pyramid is called a Gaussian pyramid.

In the case of SFS, the relation between the data (the image) and the unknown (the shape) is not linear. Consequently, reductions of the shape resolution and of the image resolution do not product the same effects. For this reason, we use the nonlinear multiresolution algorithm proposed in [25], which gives a better approximation of the shape resolution reduction. More precisely, the ideal situation would be to blur the shape in order to reduce its resolution and then to calculate the corresponding images. Unfortunately, the shape is the unknown of the problem, but the shape slopes $\rho_{i,j} = \sqrt{{p_{i,j}}^2 + {q_{i,j}}^2}$ can be computed from the gray levels using (4). Peleg and Ron [25] have shown that blurring the slopes is an approximation giving a better estimate than directly blurring the image.

If E^0 denotes the initial image (finest resolution) and E^l the image of size $2n \times 2n$ at level l, the coarser image E^{l+1} is computed using the following steps:

1. A movie corresponding to this processing can be found at the following address: http://www.irit.fr/~Jean-Denis.Durou/RECHERCHE/RECUIT/IMAGES/recuit.mpg.



(a) (b)

Fig. 5. (a) Image 256×256 of the DEM and (b) result by **M3** using four levels ($\lambda_{\text{int}} = 500$; $\lambda_{\text{smo}} = 20$).

1. Calculate slopes $\rho^l_{i,j}$ from gray levels using eikonal equation (4) at level l.
2. Blur the slope array using a Gaussian-like convolution mask.
3. Subsample the slope array by discarding every other row and column to form ρ^{l+1} of size $n \times n$.
4. Calculate the corresponding image using eikonal equation (4) at level $l+1$.

After the pyramid construction, a first result is obtained on the coarser level and this result is then oversampled by interpolation and used as the starting configuration for the next level. This process is repeated until the result with the original resolution is obtained.

Method **M1** can fail to lead to local minima and method **M2** is time demanding. Therefore, we chose to use method **M2** at the coarser level of the pyramid in order to avoid local minima and method **M1** at the other levels. The slowest method is then only used on a small image and, for the other levels, the number of iterations is reduced because the starting configuration is close to the final one.

In our experiments, we used linear interpolation for the result propagation of (p, q) between levels and the convolution mask recommended in [25] to blur the slope array during pyramid construction. We tested **M3** on the image 256×256 of the DEM represented in Fig. 5a. The pyramid was thus created with four different resolutions (256×256, 128×128, 64×64, 32×32). The result is represented in Fig. 5b. It is a little less satisfactory than Fig. 4b, but the L_2-distance between Fig. 5b and the real shape of Fig. 3a is equal to only 0.471. Regarding the CPU time, it is almost totally due to the SA (about one hour, as already stated). If we had applied **M2** to Fig. 5a, the CPU time would have been equal to something like 4^3 hours!

5 Conclusion

In this paper, we have dealt with SFS expressed as a minimization problem. The nonlinearity of the SFS equation leads to the minimization of a highly nonconvex functional. In such a case, the deterministic approaches converge to a local minimum of the functional. They provide accurate results in simple cases where the surface itself is convex or concave or when we can provide a good initial configuration, which means a configuration lying in the region of attraction of the global minimum. An alternative is to consider a stochastic algorithm that allows escape from local minima. We have shown on a shape, referred to as DEM, which contains several convex and concave areas, that the SA algorithm provides accurate results, even in this difficult case. The local minima of the functional fall very deeply and we had to consider an unusually slow cooling scheme for the temperature. This fact may be one reason explaining that, to our knowledge, the literature does not mention the association of SFS and SA. The counterpart of the success of the SA in minimizing such complex functionals is the slowness of this algorithm. In order to combine the accuracy of the SA and the efficiency of the deterministic algorithm, we have proposed a multiresolution approach in which the SA is used at the lowest resolution and provides a good initial configuration for higher resolutions that are solved deterministically.

The functional we defined is written as the sum of local constraints and can be interpreted as the energy of a Markov Random Field. On each pixel, the acceptation ratio depends only on six neighbors. Therefore, we are currently studying a parallel version of the SA algorithm. In future work, we will also try to determine how factors λ_{int} and λ_{smo} could be automatically estimated.

Acknowledgments

This work has been partially supported by the GdR ISIS/CNRS.

References

[1] *Shape from Shading,* B.K.P. Horn and M.J. Brooks, eds. MIT Press, 1989.
[2] A. Crouzil, X. Descombes, and J.-D. Durou, "A Multiresolution Approach for Shape from Shading Coupling Deterministic and Stochastic Optimization," technical report, IRIT, Univ. Paul Sabatier, Toulouse, France, 2003.
[3] B.K.P. Horn, "Obtaining Shape from Shading Information," *The Psychology of Computer Vision,* P.H. Winston, ed., chapter 4, pp. 115-155, McGraw-Hill, 1975.
[4] R. Kimmel and A.M. Bruckstein, "Tracking Level Sets by Level Sets: A Method for Solving the Shape from Shading Problem," *Computer Vision and Image Understanding,* vol. 62, no. 2, pp. 47-58, July 1995.
[5] J.-D. Durou and D. Piau, "Ambiguous Shape from Shading with Critical Points," *J. Math. Imaging and Vision,* vol. 12, no. 2, pp. 99-108, Apr. 2000.
[6] P.-L. Lions, E. Rouy, and A. Tourin, "Shape-from-Shading, Viscosity Solutions and Edges," *Numerische Mathematik,* vol. 64, pp. 323-353, 1993.
[7] P. Dupuis and J. Oliensis, "Direct Method for Reconstructing Shape from Shading," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 453-458, June 1992.
[8] M. Falcone and M. Sagona, "An Algorithm for the Global Solution of the Shape-from-Shading Model," *Proc. Ninth IEEE Int'l Conf. Image Analysis and Processing,* vol. I, pp. 596-603, Sept. 1997.
[9] P.-S. Tsai and M. Shah, "Shape from Shading Using Linear Approximation,}" *Image and Vision Computing,* vol. 12, no. 8, pp. 487-498, Oct. 1994.
[10] P. Daniel and J.-D. Durou, "From Deterministic to Stochastic Methods for Shape From Shading," *Proc. Fourth Asian Conf. Computer Vision,* pp. 187-192, Jan. 2000.
[11] R. Zhang, P.-S. Tsai, J.E. Cryer, and M. Shah, "Shape from Shading: A Survey," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 8, pp. 690-706, Aug. 1999.
[12] B.K.P. Horn and M.J. Brooks, "The Variational Approach to Shape From Shading," *Computer Vision, Graphics, and Image Processing,* vol. 33, no. 2, pp. 174-208, Feb. 1986.
[13] K. Ikeuchi and B.K.P. Horn, "Numerical Shape from Shading and Occluding Boundaries," *Artificial Intelligence,* vol. 17, no. 1-3, pp. 141-194, Aug. 1981.
[14] R.T. Frankot and R. Chellappa, "A Method for Enforcing Integrability in Shape from Shading Algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 10, no. 4, pp. 439-451, July 1988.
[15] B.K.P. Horn, "Height and Gradient from Shading," *Int'l J. Computer Vision,* vol. 5, no. 1, pp. 37-75, Aug. 1990.
[16] R. Szeliski, "Fast Shape from Shading," *Computer Vision, Graphics, and Image Processing: Image Understanding,* vol. 53, no. 2, pp. 129-153, Mar. 1991.
[17] K.M. Lee and C.-C.J. Kuo, "Shape from Shading with a Linear Triangular Element Surface Model," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 8, pp. 815-822, Aug. 1993.
[18] P.L. Worthington and E.R. Hancock, "New Constraints on Data-Closeness and Needle Map Consistency for Shape-from-Shading," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 12, pp. 1250-1267, Dec. 1999.
[19] J.-D. Durou and H. Maître, "On Convergence in the Methods of Strat and of Smith for Shape from Shading," *Int'l J. Computer Vision,* vol. 17, no. 3, pp. 273-289, Mar. 1996.
[20] A. Blake, A. Zisserman, and G. Knowles, "Surface Descriptions from Stereo and Shading," *Image and Vision Computing,* vol. 3, no. 4, pp. 183-191, 1985.
[21] G. Winkler, *Image Analysis, Random Fields and Dynamic Monte Carlo Methods: A Mathematical Introduction.* Springer-Verlag, 1995.
[22] J. Besag, "Spatial Interaction and the Statistical Analysis of Lattice Systems," *J. Royal Statistical Soc. B,* vol. 36, no. 2, pp. 192-326, July 1974.
[23] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 6, no. 6, pp. 721-741, Nov. 1984.
[24] D. Terzopoulos, "Efficient Multiresolution Algorithms for Computing Lightness, Shape from Shading, and Optical Flow," *Proc. Fourth Nat'l Conf. Artificial Intelligence,* pp. 314-317, Aug. 1984.
[25] S. Peleg and G. Ron, "Nonlinear Multiresolution: A Shape-from-Shading Example," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no. 12, pp. 1206-1210, Dec. 1990.