

# Integrated Parallel 2D-Leap-Frog Algorithm for Noisy Three Image Photometric Stereo

Ryszard Kożera<sup>1</sup>✉, Felicja Okulicka-Dłużewska<sup>2</sup>, and Lyle Noakes<sup>3</sup>

<sup>1</sup> Faculty of Applied Mathematics and Informatics,  
Warsaw University of Life Sciences-SGGW, Nowoursynowska Street 159,  
02-776 Warsaw, Poland

ryszaard.kożera@gmail.com

<sup>2</sup> Faculty of Mathematics and Information Science,  
Warsaw University of Technology, Koszykowa Street 75, 00-662 Warsaw, Poland  
f.okulicka@mini.pw.edu.pl

<sup>3</sup> School of Mathematics and Statistics, The University of Western Australia,  
35 Stirling Highway, Crawley, Perth, WA 6009, Australia  
lyle.noakes@uwa.edu.au

**Abstract.** In this paper a feasible computational scheme for reconstructing a smooth Lambertian surface  $S_L$  from noisy images is discussed. The noiseless case of Photometric Stereo relies on solving image irradiance equations. In fact, the entire shape recovery consists of gradient computation and gradient integration. The presence of added noise re-transforms the latter (depending on the adopted model) into a high-dimensional linear or non-linear optimization, solvable e.g. by a 2D-Leap-Frog. This algorithm resorts to the overlapping local image snapshot optimizations to reduce a large dimension of the original optimization task. Several practical steps to improve the feasibility of 2D-Leap-Frog are integrated in this work. Namely, an initial guess is obtained from a linear version of denoising Photometric Stereo. A non-integrable vector field estimating the normals to  $S_L$  is rectified first to yield an initial guess  $S_{L_a} \approx S_L$  for a non-linear 2D-Leap-Frog. Computationally, the integrability of non-integrable normals is enforced here by Conjugate Gradient which avoids numerous inversions of the large size matrices. In sequel,  $S_{L_a}$  is fed through to the adjusted version of non-linear 2D-Leap-Frog. Such setting not only improves the recovery of  $S_L$  (from  $S_{L_a} \approx S_L$  to  $\hat{S}_{L_a} \approx S_L$ ) but also it removes potential outliers (upon enforcing a continuity on  $\hat{S}_{L_a}$ ) occurring in the previous version of 2D-Leap-Frog. In addition, a speed-up of shape reconstruction is achieved with parallelization of non-linear 2D-Leap-Frog applied to the modified cost function. The experiments are performed on images with different resolutions and varying number of kernels. Finally, the comparison tests between standard 2D-Leap-Frog (either linear or non-linear) and its improved outlier-free version are presented illustrating differences in the quality of the reconstructed surface.

**Keywords:** Noisy Photometric Stereo · 2D-Leap-Frog · Parallelization

## 1 Introduction

In a single image *shape-from-shading problem* one searches for a smooth ( $C^1$  or  $C^2$  class) surface  $S$  obtained from its image  $\Omega$  and illuminated by a “distant” light-source direction  $p = (p_1, p_2, p_3)$  - here the light-source is positioned either at infinity or is located sufficiently far away. As shown in [1], the shape-from-shading problem is modeled by the corresponding *image irradiance equation*, which over  $\Omega$  reads as:

$$R(n_1(x, y, z), n_2(x, y, z), n_3(x, y, z)) = E(x, y). \quad (1)$$

Here, the function  $E : \Omega \rightarrow [0,1]$  measures the intensity of the light reflected from the surface  $S$  while being illuminated from the direction  $p$ . left-hand side of (1) i.e. the mapping  $R$  is called a *reflectance map*. The latter encapsulates the surface’s light reflectance properties. It is controlled by the shape of  $S$  and by the physical structure of the material covering  $S$ . Finally, the vector  $n(x, y, z) = (n_1(x, y, z), n_2(x, y, z), n_3(x, y, z))$  represents the normal to  $S$  at a given point  $(x, y, z) \in S$ . If additionally  $S = \text{graph}(u)$  (here  $u : \Omega \rightarrow \mathbb{R}$  is a smooth function) then  $z = u(x, y)$  and  $n(x, y, z)$  modulo sign reads as  $n(x, y) = (u_x(x, y), u_y(x, y), -1)$ . Consequently, the original image irradiance equation (1) reduces into  $R(x, y) = E(x, y)$ . Finding the exact formula for the reflectance map  $R$  related to the specific materials coating  $S$  constitutes a non-trivial task. A possible approach is to use a look-up table found with the aid of the so-called *calibration hemi-sphere* (see [1]). On the other hand, for certain materials the mapping  $R$  can be determined (or closely approximated) by exploiting the respective laws of optics [1, 2]. Indeed, if one deals with a particular class of *Lambertian surfaces* then  $R$  is proportional to the  $\cos(\alpha)$ , where  $\alpha$  is the angle between the normal  $n \perp S$  and light-source illumination direction  $p$ . Thus for the Lambertian surface  $S_L$  the image irradiance equation (1) is reformulated into a well-known form [1]:

$$\frac{p_1 u_x(x, y) + p_2 u_y(x, y) - p_3}{\sqrt{p_1^2 + p_2^2 + p_3^2} \sqrt{u_x^2(x, y) + u_y^2(x, y) + 1}} = E(x, y). \quad (2)$$

Clearly one admits in (2) an arbitrary constant shift of  $u$  and hence any vertical translation of  $S_L$ . Naturally this preserves the shape of  $S_L$ . However, as it turns out, (2) is still generically *ill-posed* [1, 3–9]. There are different techniques relying on availability of extra information or based on thinning the class of admissible solutions to enforce either a partial or a full *well-posedness* of (2) - see e.g. [1, 7, 9, 10]. One of the adopted feasible stances to disambiguate (2) is to illuminate  $S_L$  consecutively from multiple linearly independent directions. Such approach imposes extra constraints on the reconstruction problem (2) and is called *Photometric Stereo* - see [1, 7, 9, 10]. As well-known, three light-source Photometric Stereo is sufficient for a unique surface recovery (modulo its vertical shift  $C$ ). The case of two light-source Photometric Stereo is more complicated as merely a generic uniqueness, modulo  $u + C$ , prevails (see [10–12]).

Consequently, its analogue of noisy model is more intricate and as such is omitted here. Though four (or more) light-sources yield also uniqueness, its noisy Photometric Stereo model (a natural extension of three images) becomes more robust as it tightens stronger a negative influence of noise on reconstructed  $S_L$  (given more available data). In this paper we focus on three image Photometric Stereo. The reconstruction process is split into two explicit independent phases: *an algebraic* and *an analytical step*. First, the gradient  $\nabla u$  is uniquely computed from the following system (here (2) is used with each light-source direction, respectively):

$$\frac{\langle n|p \rangle}{\|n\| \cdot \|p\|} = E_p(x, y), \quad \frac{\langle n|q \rangle}{\|n\| \cdot \|q\|} = E_q(x, y), \quad \frac{\langle n|r \rangle}{\|n\| \cdot \|r\|} = E_r(x, y), \quad (3)$$

over  $\Omega = \Omega_1 \cap \Omega_2 \cap \Omega_3$  (see e.g. [1, 10, 13]) which yields a unique  $\nabla u$  expressed explicitly in terms of  $E_p, E_q, E_r, p, q$  and  $r$ . Both symbols  $\langle \cdot | \cdot \rangle$  and  $\| \cdot \|$  from above represent an Euclidean dot product and the corresponding norm in  $\mathbb{R}^3$ , respectively. The next step involves the verification of *the continuous integrability condition* (applied to computed  $\nabla u$ )  $\int_{\gamma_c} u_x dx + u_y dy = 0$ , which is to hold along each closed curve  $\gamma_c \in C^1(\Omega)$  over simply-connected  $\Omega$ . The *integration* formula  $u(x, y) = u(x_0, y_0) + \int_{\gamma} u_x dx + u_y dy$  gives  $u \in C^1(\Omega)$  (modulo constant  $C = u(x_0, y_0)$  taken as arbitrary) - see [14]. Here  $\gamma \in C^1(\Omega)$  is an arbitrary curve joining any  $(x, y) \in \Omega$  with a fixed  $(x_0, y_0) \in \Omega$ . Note that the above integrability condition is often replaced by  $u_{xy} = u_{yx}$ , if admissible class of  $u \in C^1(\Omega)$  is trimmed to  $C^2(\Omega)$ . Real images  $\hat{E}_p, \hat{E}_q$  and  $\hat{E}_r$  are digitized forms of  $E_p, E_q$  and  $E_r$  as they are represented over pixels instead of being measured over each point  $(x, y) \in \Omega$ . In addition, an extra camera noise infiltrates such digitization process. Commonly in computer vision one admits a Gaussian noise added to all images with mean  $\mu = 0$  and varying standard deviation  $\sigma \in [0.01, 0.10]$ . The linear approach to handle noisy Photometric Stereo relies on rectifying computed digitized *non-integrable vector field*  $v = (v_1, v_2) \in \mathbb{R}^2$  (rendered from (3) with  $\hat{E}_p, \hat{E}_q$  and  $\hat{E}_r$  entered) to the unique closest integrable  $\hat{v}$ , which satisfies a digitized analogue of continuous integrability condition - see e.g. [15–17]. The majority of such schemes rely on solving the corresponding *linear discrete optimization task* which often depends on a large number of parameters representing image resolution. Such emerging challenging computational burden can be alleviated e.g. by applying a *linear 2D-Leap-Frog* or Conjugate Gradient (see [18–22]). Both schemes manage to deal with multiple inversions of large size matrices (see also [22, 24, 25]). However a linear rectification is based on statistical principle of maximum likelihood, which assumes a Gaussian noise to be generated at the level of computed  $v$ . This implicit assumption is clearly violated here. Indeed, in reality the Gaussian noise contaminating three input images loses its character once filtered through the solutions  $(u_x, u_y)$  of (3) to yield  $v$ . As shown in [26] this fact is manifested by poorer shape reconstruction if either  $\sigma \geq 0.05$  or normals  $n \perp S_L$  vary too abruptly. In order to remove such discrepancy a *non-linear* digitized version of (2) is introduced (see e.g. [26]). More specifically, for a noisy image  $\hat{E}_p$  (also for  $\hat{E}_q$  and  $\hat{E}_r$ ) with  $N \times N$  pixel resolution,

a central-difference derivative approximation reformulates (2) into a *non-linear discrete minimization problem* in  $\hat{u} \in \mathbb{R}^{N^2-4}$ :

$$\mathcal{E}_p(\hat{u}) = \sum_{i,j=2}^{i,j=N-1} \left( \frac{p_1 \frac{\hat{u}_{i+1,j} - \hat{u}_{i-1,j}}{2\Delta x} + p_2 \frac{\hat{u}_{i,j+1} - \hat{u}_{i,j-1}}{2\Delta y} - p_3}{\|p\| \sqrt{\left(\frac{\hat{u}_{i+1,j} - \hat{u}_{i-1,j}}{2\Delta x}\right)^2 + \left(\frac{\hat{u}_{i,j+1} - \hat{u}_{i,j-1}}{2\Delta y}\right)^2 + 1}} - \hat{E}_p(i, j) \right)^2, \quad (4)$$

where  $\hat{E}_p(i, j)$  (and  $\hat{u}_{i,j}$ ) stands for the value of  $\hat{E}$  (or of  $\hat{u}$ ) at  $(i, j)$ -pixel. To simplify further computation one also assumes  $\Delta x = \Delta y = \Delta$  and  $\Omega = [0, 1] \times [0, 1]$ . We also use notation  $\mathcal{E}_p^{ij}$  for each component of (4). Note that all boundary pixels  $\in \partial\Omega$  are omitted in (4) since central-difference derivative approximation does not permit numerical estimates of  $\nabla u$  over  $\partial\Omega$ . In addition, four  $\Omega$  corner pixels are also excluded from  $\hat{u}_{i,j}$  due to the horizontal (vertical) nature of central-difference derivative approximation. Upon adding all three performance indices (see (4)) one defines a *total performance index* (or a cost or energy function):

$$\mathcal{E}(\hat{u}) = \mathcal{E}_p(\hat{u}) + \mathcal{E}_q(\hat{u}) + \mathcal{E}_r(\hat{u}), \quad (5)$$

to be minimized with  $\hat{u}_{opt} \in \mathbb{R}^{N^2-4}$ . Each component of (5) over  $(i, j)$ -pixel is denoted here by  $\mathcal{E}^{ij}$ . The non-linearity of (5) impacts now even heavier on some optimization schemes like Newton's Method which (once applied to (5)) faces multiple inversions of large size matrices  $D^2\mathcal{E} \in M_{(N^2-4) \times (N^2-4)}(\mathbb{R})$  (with  $N$  big). One of feasible computational schemes to handle (5) is a *non-linear 2D-Leap-Frog* (see [26, 27]) which likewise its linear counterpart decomposes (5) into multiple low dimension overlapping optimization tasks. To test the performance of non-linear 2D-Leap-Frog, the initial guess  $\hat{u}_0$  in [26] is generated synthetically by perturbing ideal solution  $u$  of (3) at centers of each image pixels with the respective Gaussian noise (this time  $\sigma$  is large to substantially distort  $u$ ). In case of real Photometric Stereo, the function  $u$  is unknown and therefore the scheme for finding a good initial guess  $\hat{u}_0$  requires some adjustment. In addition, though a non-linear 2D-Leap-Frog performs better than a linear model of denoising Photometric Stereo, the reconstructed surface  $\hat{S}_{L_a}$  is contaminated by the outliers mainly occurring along  $\partial\Omega$  (see [26]). A median filter in a post-reconstruction phase is applied to remove such outliers in [28, 29]. However, the latter does not yield satisfactory results if either outliers are too big or are too close to each others. This is also addressed in this paper by incorporating a new outlier removal scheme directly into the reconstruction step. It is achieved by supplementing local costs functions (5) with extra measurement to enforce the continuity of  $\hat{u}$  (which is to obeying both (5) and  $C^0$  constraint). Finally, to speed up the computational process we implement modified 2D-Leap-Frog in parallel setting (unmodified parallel version was already tested in [23]). This paper integrates all important components of noisy Photometric Stereo summarized as:

1. Finding an initial guess  $\hat{u}_0$  for non-linear optimization scheme. Conjugate Gradient is applied to the linear model of denoising Photometric Stereo [21].

2. Modification of (4) (into (6)) incorporating the continuity constraint to filter out the outliers from the reconstructed surface - see Subsect. 2.1. A non-linear 2D-Leap-Frog performing this task is applied in this paper - see Subsect. 2.2.
3. Parallelization of modified 2D-Leap-Frog used in step 2 (see Sects. 3 and 4).

## 2 Continuity Adjustment and Non-linear 2D-Leap-Frog

A detailed determination of a decent initial guess  $\hat{u}_0$  with the aid of Conjugate Gradient is given in [21, 22] and as such is here omitted.

### 2.1 Continuity Enforcement

We enforce now the continuity constraint in (5) only along the boundary pixels of  $\Omega$  (though it can be equally done over entire image). Focusing on such pixels accounts to the outliers' localization which predominantly occur along the boundary pixels (see [26]). The latter has a three-fold justification. *Firstly*, the image irradiance equation (3) are accounted in (5) only over internal image pixels due to the central-difference derivative approximations applied to  $\nabla u$ . *Secondly*,  $C^1$  implies  $C^0$  and thus internal pixels implicitly assume continuity (also in its discrete form). *Lastly*, 2D-Leap-Frog (linear or non-linear one - see [18, 26]) relies on freezing majority of variables and minimizing the corresponding performance index (e.g. (5)) over small number of free variables (which this time can be computationally handled e.g. by Newton's Method). The free variables represent  $\hat{u}_{i,j}$  over  $(i, j)$ -pixels which all belong to the selected  $\Omega$ -snapshot (either a square or a rectangle). The optimization process relies on varying unlocked variables (and thus snapshots) covering  $\Omega$  in overlapping mode. Once a total snapshot coverage of  $\Omega$  with no self repetitions is accomplished (consecutive or distributed mode is usually admitted) one iteration of 2D-Leap-Frog is completed. Generically, there are 9 types of local snapshot optimizations (see [18]) i.e.: *bottom-left, top-left, top-right, bottom-right corner snapshots and bottom-, left-, top-, right- boundary snapshots and finally the internal snapshot* (applied the most frequently). 2D-Leap-Frog adaptively and iteratively changes the unlocked value of  $\hat{u}_{i,j}$  by including the  $(i, j)$ -pixel in neighbouring local snapshot optimizations during a single iteration. Noticeably, pixels in  $\partial\Omega$  have in their proximity a scarcer number of adjacent overlapping snapshots (which size is also permitted to vary) as opposed to the internal ones. This impacts on the adaptivity quality of 2D-Leap-Frog over all boundary pixels. For  $\delta_{i,j} = (\hat{u}_{i+1,j} - \hat{u}_{i,j})^2$  and  $\rho_{i,j} = (\hat{u}_{i,j+1} - \hat{u}_{i,j})^2$  the forward-difference continuity correction over the left-boundary pixels is:  $\mathcal{E}_L(\hat{u}) = \sum_{j=2}^{N-1} \delta_{1,j} + \sum_{i=2}^{N-2} \delta_{i,1}$ . Similarly one defines  $\mathcal{E}_T$ ,  $\mathcal{E}_R$  and  $\mathcal{E}_B$  over top-, right- and bottom- boundaries, respectively. Coupling the latter with (5) yields a *modified performance index function*:

$$\mathcal{E}_{C^0}(\hat{u}) = \mathcal{E}(\hat{u}) + \mathcal{E}_L(\hat{u}) + \mathcal{E}_T(\hat{u}) + \mathcal{E}_R(\hat{u}) + \mathcal{E}_B(\hat{u}). \quad (6)$$

The task is now to find  $\hat{u}_{opt}$  which globally minimizes (6). The initial condition  $\hat{u}_0$  is computed as indicated in Step 1 (see the closing of Sect. 1). A non-linear

2D-Leap-Frog to find a local minimum of (6) is used in this paper. As shown in [26] 2D-Leap-Frog converges to a critical point (usually a local minimum) for any smooth performance index function. In addition, an extra analysis in [26] formulates sufficient conditions under which a global minimum for  $\mathcal{E}$  is reached. A similar analysis for  $\mathcal{E}_{C^0}$  exceeds the scope of this paper and is omitted. Newton’s Method is applied to all local optimizations with the first initial guess as the respective values from  $\hat{u}_0$  over the first snapshot. During the  $s$ -iteration of 2D-Leap-Frog (and selection of  $t$ -snapshot) an initial guess is formed by these values of  $u^{(s,t-1)} \in \mathbb{R}^{N^2-4}$  which correspond to the unlocked variables of  $t$ -snapshot optimization. Upon completing  $t$ -snapshot optimization we update  $u^{(s,t-1)}$  to  $u^{(s,t)} \in \mathbb{R}^{N^2-4}$  by feeding in  $u^{(s,t-1)}$  all optimal values of unlocked pixels standing for  $t$ -snapshot minimization. The common overlap (e.g. for small size snapshots) involves either horizontal or vertical half-snapshot translations. The stopping condition for 2D-Leap-Frog amounts here to a cap on a prescribed number of iterations. Experiments show that the corresponding energy is substantially decreased only within the first 5–12 initial iterations. The subsequent steps diminish the energy marginally while still consuming considerable amount of time and memory access.

## 2.2 Local Snapshot Optimizations

We introduce now local performance indices over *internal*, *left-boundary* and *bottom-left corner snapshots*, respectively. The remaining 6 types of local optimizations are analogous. Note that over each snapshot the respective local performance index should be calculated over all  $(k, l)$ -pixels which energy  $\mathcal{E}^{kl}$  is well-defined and depends on any unlocked  $(i, j)$ -pixel (here  $\hat{u}_{ij}$  is relaxed to a free variable). Otherwise the total performance index  $\mathcal{E}_{C^0}$  may increase despite decreasing inappropriately chosen local energy function. Indeed, for the wrongly omitted pixels the increase of energy may well exceed its trimming gained over all pixels contributing to the local performance index. The size of each snapshot can vary but it cannot cross over certain values (limited by selected local optimization scheme), as otherwise computational burden is too excessive. In this paper  $6 \times 6$ ,  $5 \times 6$  and  $5 \times 5$  pixel-sizes for three types of snapshots are applied. The snapshot’s size cannot be too small either to avoid too dense coverage of  $\Omega$  by local optimizations (each depending on iterative numerical computations).

Over each snapshot’s type *Mathematica FindMinimum* function is invoked with the default Newton’s Method and default number of iterations.

(a) *Internal snapshot optimization*: is generically dealt by 2D-Leap-Frog. Recall that continuity constraint is left-out as generic snapshot  $\Omega^g$  (here  $6 \times 6$  mask shifted over  $\Omega$ ) is isolated from  $\partial\Omega$ . The essential local variables in  $\Omega^g$  (see Fig. 1a) are represented by either unlocked variables  $\{v_1, v_2, v_3, v_4\}$  or locked variables  $\{w_1, w_2, \dots, w_{20}\}$  over the corresponding pixels. The rest of the blank pixels in Fig. 1a refer to locked pixels which do not contribute to the local

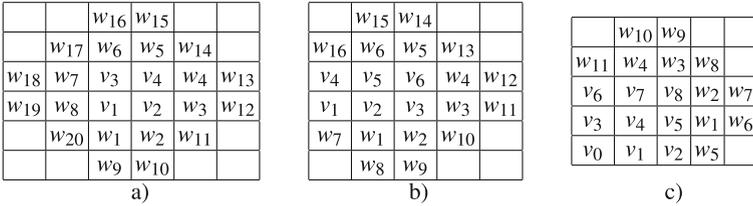
energy  $\mathcal{E}^g$  depending here on free variables  $v = (v_1, v_2, v_3, v_4)$  and defined over  $\{v_1, v_2, v_3, v_4, w_1, w_2, \dots, w_8\}$ -pixels as:

$$\mathcal{E}^g(v) = \sum_{i=1}^4 \mathcal{E}_{v_i}^{gg} + \sum_{j=1}^8 \mathcal{E}_{w_j}^{gg}, \quad (7)$$

where (see (4) and Fig. 1a) e.g.  $\mathcal{E}_{v_2}^g = \mathcal{E}_{v_2}^{gp} + \mathcal{E}_{v_2}^{gq} + \mathcal{E}_{v_2}^{gr}$  (at  $v_2$ -pixel) reads as:

$$\mathcal{E}_{v_2}^{gp} = \left( \frac{p_1 \frac{w_3 - v_1}{2\Delta} + p_2 \frac{v_4 - w_2}{2\Delta} - p_3}{\|p\| \sqrt{1 + \left(\frac{w_3 - v_1}{2\Delta}\right)^2 + \left(\frac{v_4 - w_2}{2\Delta}\right)^2}} - \hat{E}_p(i(v_2), j(v_2)) \right)^2, \quad (8)$$

with  $\mathcal{E}_{v_2}^{gq}$  and  $\mathcal{E}_{v_2}^{gr}$  computed similarly to (8) upon replacing  $p$  with  $q$  or  $r$ , accordingly. Here  $(i(v_2), j(v_2))$  represents a center of a pixel in  $\Omega$  which coincides with the  $v_2$ -pixel of  $\Omega^g$  masking the respective image sub-domain during Leap-Frog  $(s, t)$ -iteration. The remaining 11 energies in (7) are analogously treated upon using (4), (8) and Fig. 1a. The current values of contributing locked variables in  $\Omega^g$  are assigned to the corresponding updates of  $\Omega$ -pixels computed during  $(s, t - 1)$ -iteration of 2D-Leap-Frog. The same procedure applies to the initial guess  $(v_1^0, v_2^0, v_3^0, v_4^0)$  needed to minimize (7).



**Fig. 1.** The main three types of snapshots for modified 2D-Leap-Frog.

(b) *Left-boundary snapshot optimization*: incorporates also the continuity constraint over  $\Omega^{lb}$  (here  $5 \times 6$  mask translated over  $\Omega$ ). This time the essential local variables in  $\Omega^{lb}$  (see Fig. 1b) are represented by either unlocked variables  $\{v_1, v_2, \dots, v_6\}$  or locked variables  $\{w_1, w_2, \dots, w_{16}\}$  over the corresponding pixels. Again the remaining blank pixels in Fig. 1b refer to locked pixels not participating in local energy  $\mathcal{E}^{lb}$  depending here on free variables  $v = (v_1, v_2, \dots, v_6)$  and defined over  $\{v_2, v_3, v_5, v_6, w_1, w_2, \dots, w_6\}$ -pixels (as far as discrete image irradiance equation (4) is concerned) and also incorporating forward-difference continuity over  $\{v_1, v_4, w_7\}$ -pixels as follows:

$$\mathcal{E}^{lb}(v) = \sum_{i=2, i \neq 4}^6 \mathcal{E}_{v_i}^{lb} + \sum_{j=1}^6 \mathcal{E}_{w_j}^{lb} + \mathcal{E}_c^{lb}, \quad (9)$$

where  $\mathcal{E}_{v_i}^{lb}$  and  $\mathcal{E}_{w_j}^{lb}$  are calculable similarly to (7) (see Fig. 1b) and  $\mathcal{E}_c^{lb} = (v_2 - v_1)^2 + (v_5 - v_4)^2 + (v_4 - v_1)^2 + (w_{16} - v_4)^2 + (v_1 - w_7)^2$ . Similarly, the current values of contributing locked variables in  $\Omega^{lb}$  are the updates of  $\Omega$ -pixel values computed during  $(s, t - 1)$ -iteration of 2D-Leap-Frog. A similar approach is used to determine the initial condition  $(v_1^0, v_2^0, \dots, v_6^0)$  needed to minimize (9). The remaining three cases of this type are treated analogously.

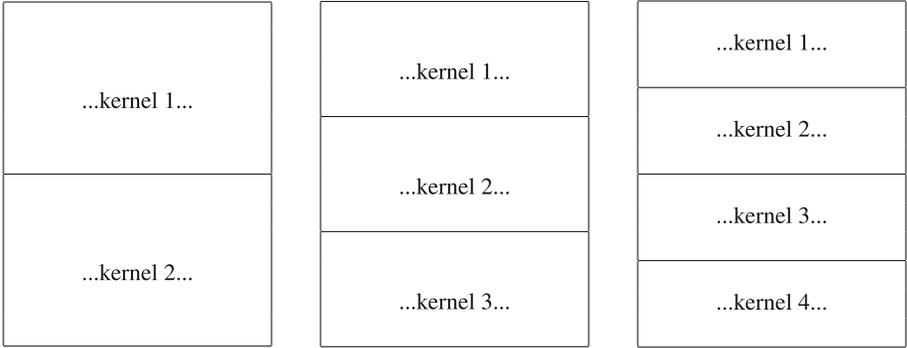
(c) *Bottom-left corner optimization*: over  $\Omega^{blc}$  (a  $5 \times 5$  mask moved over  $\Omega$ ) and its three analogues cover the neighbourhoods of four corners of  $\Omega$ . Recall that corner pixel values  $u(1, 1)$ ,  $u(1, N)$ ,  $u(N, 1)$  and  $u(N, N)$  are originally excluded thus admitting a digitized solution  $\hat{u} \in \mathbb{R}^{N^2-4}$ . These four missing values can be filled after termination of 2D-Leap-Frog by a straightforward 4-parameter linear optimization in fact enforcing the continuity of extended  $\hat{u} \in \mathbb{R}^{N^2}$  at four corner pixels (see [26]). The alternative is to integrate the latter into each corner snapshot optimization which is adopted in this paper for modified 2D-Leap-Frog (thus from now on it is assumed that  $\hat{u} \in \mathbb{R}^{N^2}$ ). Here the essential local variables in  $\Omega^{blc}$  (see Fig. 1c) are represented by either unlocked variables  $\{v_0, v_1, \dots, v_8\}$  or locked variables  $\{w_1, w_2, \dots, w_{11}\}$  over the corresponding pixels. The omitted blank pixels in Fig. 1c refer to locked variables which do not contribute to local energy  $\mathcal{E}^{blc}$  depending now on free variables  $v = (v_0, v_1, \dots, v_8)$  and defined over  $\{v_4, v_5, v_7, v_8, w_1, w_2, w_3, w_4\}$ -pixels (as far as discrete image irradiance equation (4) is concerned) and also including forward-difference continuity over  $\{v_0, v_1, v_2, v_3, v_4, v_6\}$ -pixels as follows:

$$\mathcal{E}^{blc}(v) = \sum_{i=4, i \neq 6}^8 \mathcal{E}_{v_i}^{blc} + \sum_{j=1}^4 \mathcal{E}_{w_j}^{blc} + \mathcal{E}_c^{blc}, \quad (10)$$

where again  $\mathcal{E}_{v_i}^{blc}$  and  $\mathcal{E}_{w_j}^{blc}$  are defined similarly to (7) (see also Fig. 1c) and  $\mathcal{E}_c^{blc} = (v_4 - v_1)^2 + (v_5 - v_2)^2 + (v_2 - v_1)^2 + (w_5 - v_2)^2 + (v_4 - v_3)^2 + (v_7 - v_6)^2 + (v_6 - v_3)^2 + (w_{11} - v_6)^2 + (v_1 - v_0)^2 + (v_3 - v_0)^2$ . The last two components in  $\mathcal{E}_c^{blc}$  enforce the continuity of  $\hat{u} \in \mathbb{R}^{N^2}$  at added  $(1, 1)$ -pixel. Again the current values of contributing locked variables in  $\Omega^{blc}$  are the updated values of  $\Omega$ -pixel computed during  $(s, t - 1)$ -iteration of 2D-Leap-Frog. A similar approach applies here to determine an initial guess  $(v_0^0, v_2^0, \dots, v_8^0)$  needed to minimize (10). Again the remaining three cases of this type are treated analogously.

### 3 Parallelization of Modified 2D-Leap-Frog

The parallel version is implemented on the PC four core shared memory machine with the processor Intel Core, CPU 3.00 GHz, 4.00 GB RAM (by using *Mathematica 9.0* package). The parallel (2, 3 or 4)-kernels are launched by *Mathematica LaunchKernel[]* and *DistributeDefinitions[]* functions. For  $k$  kernels used, the entire image  $\Omega$  is divided into disjoint  $k$ -horizontal parts, i.e.  $\Omega = \cup_{i=1}^k \Omega_i$  and  $\Omega_i \cap \Omega_j = \emptyset$ , for  $i \neq j$  - see Fig. 2. Each  $\Omega_i$  is swept out by one iteration of  $i$ -th kernel with (interchangeably first horizontally and then vertically) overlapping



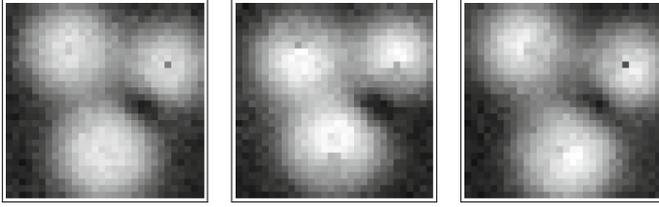
**Fig. 2.** Divisions of  $\Omega$ -grid in the parallel version of modified 2D-LF.

half-snapshots local optimizations. Next, border pixels along  $\partial\Omega_i$  (or  $\partial\Omega_{i+1}$ ) between two adjacent sub-regions  $\Omega_i$  and  $\Omega_{i+1}$  are updated by single horizontal run of 2D-LF procedure. Naturally, different “lines” of border pixels are covered in parallel. This terminates one iteration of parallel modified 2D-Leap-Frog (over  $\Omega$ ). Note that the sequential 2D-Leap-Frog (which sweeps  $\Omega$  with consecutive horizontal overlaps) varies from parallel 2D-Leap-Frog (border pixels between two adjacent sub-regions are performed at the end of each iteration). The latter may cause minor dissimilarities in reconstructed surfaces for sequential versus parallel ones or in parallel solutions computed with different number of kernels.

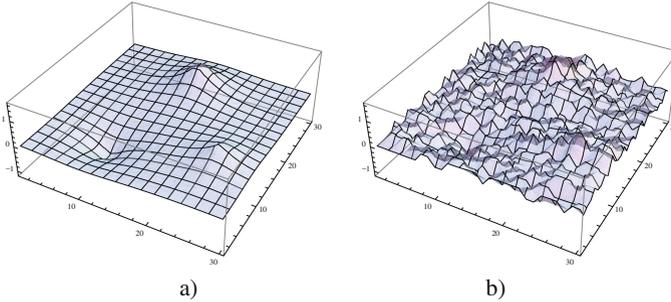
## 4 Numerical Experiments

Input images examined in this paper have  $16 \times 16$ ,  $32 \times 32$  or  $64 \times 64$  pixel resolutions. To test the performance of 2D-Leap-Frog, we eliminate a potential non-conformity of the adopted exact Lambertian model assumed to be ingrained in real images, by considering exclusively synthetic Lambertian images  $E_p$ ,  $E_q$  and  $E_r$ . To secure the latter, all images are generated by substituting ideal  $\nabla u$  (for which  $S_L = \text{graph}(u)$ ) into (3). In sequel, the Gaussian noise is supplemented to  $E_p$ ,  $E_q$  and  $E_r$  yielding the respective noisy images  $\hat{E}_p$ ,  $\hat{E}_q$  and  $\hat{E}_r$  which in turn serve as input data. It is also implicitly assumed that the entire  $\Omega = [0, 1] \times [0, 1]$ . The case of  $(x, y) \in \Omega$  for which  $\cos(\alpha) < 0$  (representing invisible part of  $S_L$  - see (2)) is here artificially admitted in order to simplify the implementation of modified non-linear 2D-Leap-Frog. Still image irradiance equation (3) over invisible part of  $S_L$  can be set-up and analyzed. Newton’s Method is used for local snapshot optimizations (a default option in *Mathematica FindMinimum* function).

*Example 1.* Consider the Lambertian surface  $S_{L_1} = \text{graph}(u_1)$ , where function  $u_1(x, y) = \frac{1}{16}(20f((x, y), w_1) - 15f((x, y), w_2) + 12f((x, y), w_3))$ , with  $w_1 = (\frac{3}{4}, \frac{1}{2})$ ,  $w_2 = (\frac{1}{4}, \frac{1}{3})$ ,  $w_3 = (\frac{1}{3}, \frac{4}{5})$  and  $f(\tilde{v}_1, \tilde{v}_2) = e^{-100\|\tilde{v}_1 - \tilde{v}_2\|^2}$  for  $\tilde{v}_1, \tilde{v}_2 \in \mathbb{R}^2$ ,

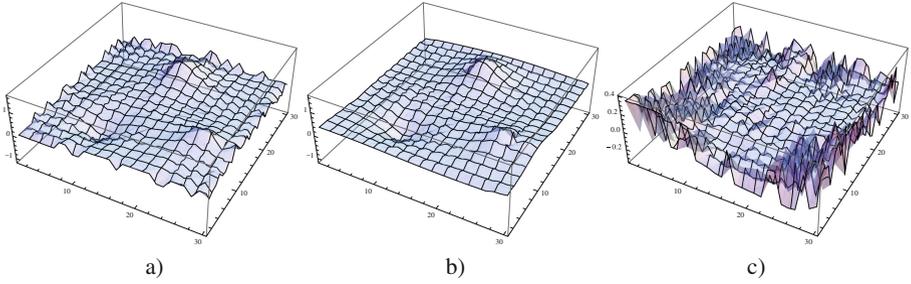


**Fig. 3.** Noisy digitized images  $\hat{E}_p$ ,  $\hat{E}_q$  and  $\hat{E}_r$  for surface  $S_{L_1}$  with Gaussian noise  $\mathcal{N}(0.0, 0.05)$ .



**Fig. 4.** (a) Ideal  $S_{L_1}$ , (b) reconstructed  $S_{L_{1a}}$  by using CG (with Gaussian noise  $\mathcal{N}(0.0, 0.05)$ ).

is defined over  $\Omega = [0, 1] \times [0, 1]$ . The light-source directions are:  $p = (0, 0, -1)$ ,  $q = (0, \frac{1}{3}, -\frac{1}{\sqrt{2}})$  and the third one is  $r = (\frac{1}{\sqrt{7}}, 0, -\frac{1}{\sqrt{2}})$ . The noiseless digitized images  $E_p$ ,  $E_q$  and  $E_r$  of  $S_{L_1}$  are contaminated with the Gaussian  $\mathcal{N}(0.0, 0.05)$  noise to yield  $\hat{E}_p$ ,  $\hat{E}_q$  and  $\hat{E}_r$  shown in Fig. 3. The ideal surface  $S_{L_1}$  and the reconstructed surface  $S_{L_{1a}}$  by Conjugate Gradient (abbreviated with CG - see [21]) are shown in Fig. 4 with  $32 \times 32$  pixel image resolution. The latter forms an initial guess to the non-linear 2D-Leap-Frog (with 2D-LF shorthand notation). Figure 5 demonstrates two surfaces  $\hat{S}_{L_{1a}}^e$  and  $\hat{S}_{L_{1a}}^{not(o)}$  (and the difference between them) obtained without or with 2D-LF outlier removal modification (with 13 iterations). Next the parallel version of modified 2D-LF is tested (with 6 iterations). Figure 6 shows the reconstructed surfaces  $\hat{S}_{L_{1a}}^{not(o)}$  using either 2, 3 or 4 kernels (for  $32 \times 32$  image resolution). The respective *speed-ups* of the parallel modified 2D-LF for different image resolutions and for varying number of kernels are presented in Table 1a. Note that the speed-up can be greater than the number of created kernels. It is the result of the speed of the memory access. Each kernel operating in a parallel mode deals with updating smaller tables as compared with a single kernel performing a similar task on its own - thus a memory access is quicker.  $\square$

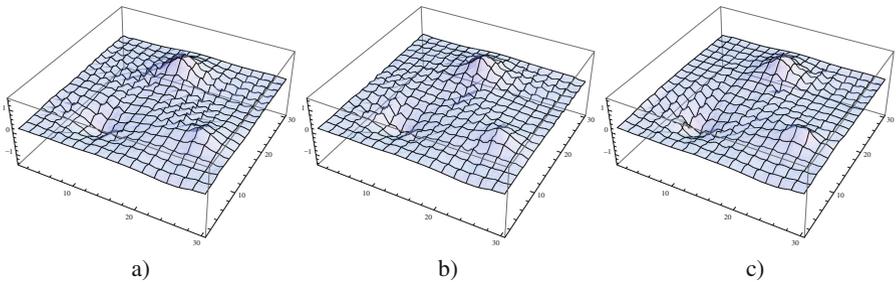


**Fig. 5.**  $S_{L_1}$  reconstructed by 2D-LF: (a)  $\hat{S}_{L_{1a}}^o$  without, (b)  $\hat{S}_{L_{1a}}^{not(o)}$  with outlier removal (with Gaussian noise  $\mathcal{N}(0.0, 0.05)$ ), (c) the difference between  $\hat{S}_{L_{1a}}^{not(o)}$  and  $\hat{S}_{L_{1a}}^o$ .

**Table 1.** Speed-ups in parallel modified 2D-LF Algorithm for  $S_{L_1}$  and  $S_{L_2}$ .

Resolution	a) Example 1				b) Example 2			
	1 kernel	2 kernels	3 kernels	4 kernels	1 kernel	2 kernels	3 kernels	4 kernels
16 × 16	1	1.5	1.5	1.6	1	1.2	1.3	1.2
32 × 32	1	2.0	2.3	2.8	1	2.2	2.7	2.9
64 × 64	1	2.4	3.0	3.6	1	2.0	2.6	3.4

*Example 2.* Let  $S_{L_2} = graph(u_2)$ , where  $u_2(x, y) = (0.75 + \frac{1}{3} * (1 - tanh((x + y - 2)^2 + (x - y)^2 - \frac{25}{3})))$  is defined over  $\Omega = [0, 1] \times [0, 1]$ . The light-source directions are:  $p = (0, 0, -1)$ ,  $q = (1 - \sqrt{3}, 0, -1 - \sqrt{3}) / (2\sqrt{2}) = (-0.258819, 0.0, -0.965926)$  and the third one  $r = (\frac{1}{2} \sin \frac{\pi}{24}, \frac{\sqrt{3}}{2} \sin \frac{\pi}{24}, -\cos \frac{\pi}{24}) = (0.0652631, 0.113039, -0.991445)$ . The noiseless digitized images  $E_p, E_q$  and  $E_r$  of  $S_2$  upon contaminated with Gaussian noise  $\mathcal{N}(0.0, 0.05)$  yield the respective noisy images  $\hat{E}_p, \hat{E}_q$  and  $\hat{E}_r$  (see Fig. 7). Figure 8 illustrates the ideal surface  $S_{L_2}$  and the reconstructed one  $S_{L_{2a}}$  by CG Algorithm (with  $32 \times 32$  image resolution). As previously  $S_{L_{2a}}$  forms an



**Fig. 6.**  $S_{L_1}$  reconstructed by parallel modified 2D-LF using (a) 2, (b) 3 and (c) 4 kernels.

initial guess to the non-linear 2D-LF. Figure 9 illustrates the surfaces  $\hat{S}_{L_{2a}}^o$  and  $\hat{S}_{L_{2a}}^{not(a)}$  (and the difference between them) generated without or with 2D-LF outlier removal (with 13 iterations). Finally, the parallel version of modified 2D-LF (with 6 iterations) is exercised. Figure 10 illustrates the reconstructed surfaces  $\hat{S}_{L_{2a}}^{not(0)}$  upon applying either 2, 3 or 4 kernels (for  $32 \times 32$  image resolution). The resulting accelerations reached by parallel modified 2F-LF for different image resolutions and varying number of kernels are shown in Table 1b.  $\square$

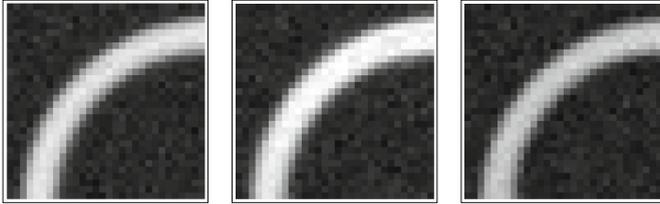


Fig. 7. Noisy digitized images  $\hat{E}_p$ ,  $\hat{E}_q$  and  $\hat{E}_r$  for surface  $S_{L_2}$  with Gaussian noise  $\mathcal{N}(0.0, 0.05)$ .

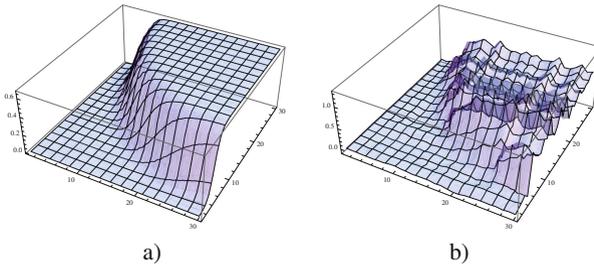


Fig. 8. (a) Ideal  $S_{L_2}$ , (b) reconstructed  $S_{L_{2a}}$  by CG (with Gaussian noise  $\mathcal{N}(0.0, 0.05)$ ).

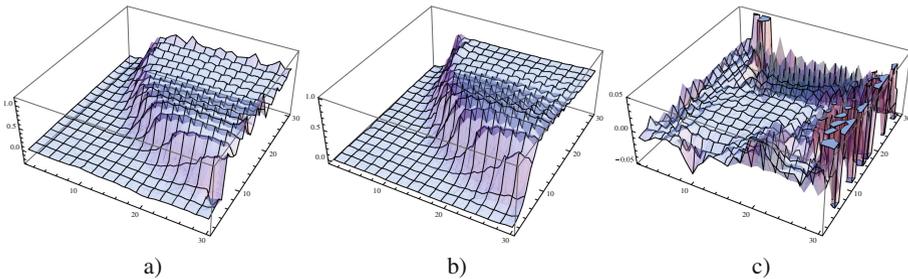
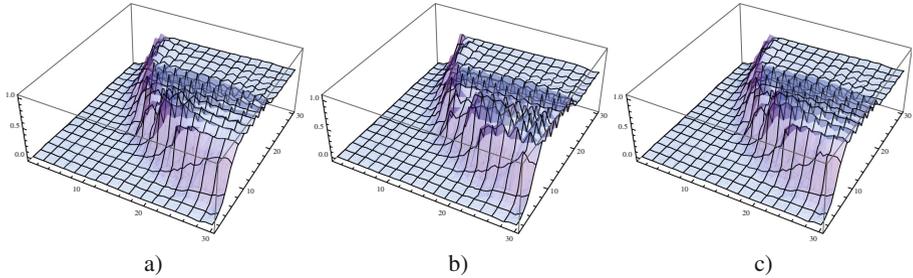


Fig. 9.  $S_{L_2}$  reconstructed by 2D-LF: (a)  $\hat{S}_{L_{2a}}^o$  without, (b)  $\hat{S}_{L_{2a}}^{not(o)}$  with outlier removal (with Gaussian noise  $\mathcal{N}(0.0, 0.05)$ ), (c) the difference between  $\hat{S}_{L_{2a}}^{not(o)}$  and  $\hat{S}_{L_{2a}}^o$ .



**Fig. 10.**  $S_{L_2}$  reconstructed by parallel modified 2D-LF using a) 2, b) 3 and c) 4 kernels.

## 5 Conclusions

In this paper we propose an integrated feasible computational scheme i.e. *a modified non-linear 2D-Leap-Frog* (including its parallelization) to reconstruct the Lambertian surface from three light-source noisy Photometric Stereo images. The initial guess obtained from Conjugate Gradient (see [21]) and the continuity enforcement (obliterating the boundary outliers) combined with parallelization of 2D-Leap-Frog confirm to perform very satisfactory in all conducted experiments. Note that 2D-Leap-Frog is versatile as adaptable to *any optimization task depending on large number of variables* (see e.g. [30]). Future work includes genuine camera images with large image resolution, for which the size of each snapshot may vary and unshaded pixels are excluded (to yield  $\Omega \subset [0, 1] \times [0, 1]$ ). The latter most likely requires a more powerful hardware amenable to parallel computation. Another potential investigation refers to two-source noisy Photometric Stereo (see [10–12]), where only generic uniqueness prevails and as such should be accounted in 2D-Leap-Frog. Finally, a similar analysis to [26] establishing sufficient conditions for *modified non-linear 2D-Leap-Frog to render a global minimum* of (6) would undeniably complement this work. More work on related topic can also be found in [31–33].

## References

1. Horn, B.K.P.: Robot Vis. McGraw-Hill, New York (1986)
2. Luneburg, R.K.: Mathematical Theory of Optics. University of California Press, Berkeley and Los Angeles (1964)
3. Kozera, R.: Uniqueness in shape from shading revisited. *J. Math. Imag. Vision* **7**, 123–138 (1997)
4. Horn, B.K.P., Brooks, M.J.: Shape from Shading. MIT Press, Cambridge (1989)
5. Kozera, R.: On complete integrals and uniqueness in shape from shading. *Appl. Math. Comput.* **73**(1), 1–37 (1995)
6. Brooks, M.J., Chojnacki, W., Kozera, R.: Impossible and ambiguous shading patterns. *Int. J. Comput. Vis.* **7**(2), 119–126 (1992)
7. Oliensis, J.: Uniqueness in shape from shading. *Int. J. Comput. Vis.* **6**(2), 75–104 (1991)

8. Brooks, M.J., Chojnacki, W., Kozera, R.: Circularly symmetrical eikonal equations and non-uniqueness in computer vision. *J. Math. Anal. Appl.* **165**(1), 192–215 (1992)
9. Brooks, M.J., Chojnacki, W.: Direct computation of shape-from-shading. In: 12th International Conference on Pattern Recognition, Jerusalem, Israel, pp. 114–119. IEEE Computer Society Press, Los Alamitos, CA (1994)
10. Kozera, R.: Existence and uniqueness in photometric stereo. *Appl. Math. Comput.* **44**(1), 1–104 (1991)
11. Kozera, R.: On shape recovery from two shading patterns. *Int. J. Pattern Recognit. Artif. Intell.* **6**(4), 673–698 (1992)
12. Onn, R., Bruckstein, A.: Uniqueness in shape from shading. *Int. J. Comput. Vis.* **5**(1), 105–113 (1990)
13. Woodham, R.J.: Photometric method for determining surface orientation from multiple images. *Opt. Eng.* **19**(1), 139–144 (1980)
14. Horn, B.K.P.: Height and gradient from shading. *Int. J. Comput. Vis.* **5**(1), 37–75 (1990)
15. Wei, T., Klette, R.: On depth recovery from gradient vector field. In: Bhat-tacharaya, B.B., et al. (eds.) *Algorithms, Architectures and Information Systems Security*, pp. 765–797. World Scientific Publishing Co., Pte. Ltd., Singapore (2009)
16. Simchony, T., Chellappa, R., Shao, M.: Direct analytical methods for solving poisson equations in computer vision problems. *IEEE Trans. Pattern. Recognit. Mach. Intell.* **12**(5), 435–446 (1990)
17. Frankot, R.T., Chellappa, R.: A method of enforcing integrability in shape from shading algorithms. *IEEE Trans. Pattern. Recognit. Mach. Intell.* **10**(4), 439–451 (1988)
18. Noakes, L., Kozera, R.: The 2-D Leap-Frog, noise and digitization. In: Bertrand, G., et al. (eds.) *Digital and Image Geometry. LNCS*, vol. 2243, pp. 352–364. Springer, Heidelberg (2001)
19. Noakes, L., Kozera, R.: The Lawn-Mowing Algorithm for noisy gradient vector fields. In: Latecki, L.J. (ed.) *Vision Geometry VIII. Proceedings SPIE*, vol. 3811, pp. 305–316. Denver, CO, USA (1999)
20. Noakes, L., Kozera, R.: The 2-D Leap-Frog Algorithm for optimal surface reconstruction. In: Latecki, L.J. (ed.) *Vision Geometry VIII. Proceedings SPIE*, vol. 3811, pp. 317–328. Denver, CO, USA (1999)
21. Kozera, R., Okulicka-Dłużewska, F.: Conjugate gradient in noisy photometric stereo. In: Chmielewski, L.J., Kozera, R., Shin, B.-S., Wojciechowski, K. (eds.) *ICCVG 2014. LNCS*, vol. 8671, pp. 338–346. Springer, Heidelberg (2014)
22. Saad, Y.: *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia (2003)
23. Cameron, T., Kozera, R., Datta, A.: A parallel Leap-Frog Algorithm for 3-source photometric stereo. In: Wojciechowski, K., et al. (eds.) *ICCVG 2004*, pp. 95–102. Springer, Dordrecht (2006)
24. Hackbush, W.: *Iterative Solution of Large Sparse Systems of Equations*. Springer, New York (1994)
25. van der Vorst, H.A.: *Iterative Krylov Methods for Large Linear Systems*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge (2009)

26. Noakes, L., Kožera, R.: Nonlinearities and noise reduction in 3-source photometric stereo. *J. Math. Imag. Vis.* **18**(3), 119–127 (2003)
27. Noakes, L., Kožera, R.: Denoising images: non-linear Leap-Frog for shape and light-source recovery. In: Asano, T., Klette, R., Ronse, C. (eds.) *Geometry, Morphology, and Computational Imaging*. LNCS, vol. 2616, pp. 419–436. Springer, Heidelberg (2003)
28. Kožera, R., Tchórzewski, J.: Outlier removal in 2D Leap Frog Algorithm. In: Cortesi, A., Chaki, N., Saeed, K., Wierzchoń, S. (eds.) *CISIM 2012*. LNCS, vol. 7564, pp. 146–157. Springer, Heidelberg (2012)
29. Arce, G.R.: *Non-Linear Signal Processing: A Statistical Approach*. Wiley, New Jersey (2005)
30. Noakes, L.: A global algorithm for geodesics. *J. Math. Austral. Soc. Ser. A* **64**, 37–50 (1999)
31. Castelán, M., Hancock, E.R.: Imposing integrability in geometric shape-from-shading. In: Sanfeliu, A., Ruiz-Shulcloper, J. (eds.) *CIARP 2003*. LNCS, vol. 2905, pp. 196–203. Springer, Heidelberg (2003)
32. Hurt, N.E.: Mathematical methods in shape-from-shading: a review of recent results. *Acta Appl. Math.* **23**, 163–188 (1991)
33. Wöhler, C.: *3D Computer Vision: Efficient Methods and Applications*. Springer-Verlag, Berlin Heidelberg (2009)