Computer Science Department of The University of Auckland
CITR at Tamaki Campus (http://www.tcs.auckland.ac.nz)

CITR-TR-46 June 1999

The Lawn-Mowing Algorithm for Noisy Gradient Vector Fields

Lyle Noakes[1], Ryszard Kozera [2], and Reinhard Klette [3]

Abstract

In this paper we analyze a specific problem within the context of recovering the geometric shape of an unknown surface from multiple noisy shading patterns generated by consecutive parallel illuminations by different light-sources.

Shading-based single-view shape recovery in computer vision often leads to vector fields (i.e. estimated surface normals) which have to be integrated for calculations of height or depth maps. We present an algorithm for enforcing the integrability condition of a given non-integrable vector field which ensures a global suboptimal solution by local optimizations. The scheme in question relies neither on a priori knowledge of boundary conditions nor on other global constraints imposed on the so-far derived noise contaminated gradient integration techniques. The discussion is supplemented by examples illustrating algorithm performance.

[1] Department of Mathematics

[2]Department of Computer Science
The University of Western Australia, Nedlands, WA 6907, Australia

[3] The University of Auckland, Computer Science Department, CITR,
Tamaki Campus (Building 731), Glen Innes, Auckland, New Zealand

The Lawn-Mowing Algorithm for Noisy Gradient Vector Fields

Lyle Noakes*[a], Ryszard Kozera*[b], and Reinhard Klette*[c]

[a]Department of Mathematics
[b]Department of Computer Science
The University of Western Australia, Nedlands, WA 6907, Australia

[c]CITR Tamaki, University of Auckland
Tamaki Campus, Building 731, Auckland, New Zealand

ABSTRACT

In this paper we analyze a specific problem within the context of recovering the geometric shape of an unknown surface from multiple noisy shading patterns generated by consecutive parallel illuminations by different light-sources. Shading-based single-view shape recovery in computer vision often leads to vector fields (*i.e.* estimated surface normals) which have to be integrated for calculations of height or depth maps. We present an algorithm for enforcing the integrability condition of a given non-integrable vector field which ensures a global suboptimal solution by local optimizations. The scheme in question relies neither on *a priori* knowledge of boundary conditions nor on other global constraints imposed on the so-far derived noise contaminated gradient integration techniques. The discussion is supplemented by examples illustrating algorithm performance.

Keywords: Integrability, Photometric Stereo Method, vector fields, surface normals, surface integration, height map.

1. INTRODUCTION

A monochrome picture of a smooth object surface typically exhibits brightness variation or *shading*. Of interest to researchers in computer vision has been the problem of how object shape may be extracted from image shading. It has been shown in Subsection 10.10 in Horn[10] that this *shape-from-shading problem* corresponds to solving a first-order partial differential equation. Specifically, one seeks a *surface function* u, representing the depth map (*i.e.* the distances between the focal point of the camera and surface points) in the direction of the $z-$axis, satisfying the so-called *image irradiance equation*. In the case of a *Lambertian surface* (a perfect light diffuser), illuminated from direction (p_1, p_2, p_3), the image irradiance equation takes the special form

$$\frac{p_1 u_x(x,y) + p_2 u_y(x,y) - p_3}{\sqrt{p_1^2+p_2^2+p_3^2}\,\sqrt{u_x^2(x,y)+u_y^2(x,y)+1}} \cdot \rho(x,y) \;=\; E(x,y) \qquad (1)$$

over a compact image domain Ω. Here E is an image intensity formed by orthographic projection of light onto a plane parallel to the xy-plane defined on an image domain Ω, and ρ denotes the surface albedo. In practice the irradiance values E are degraded by measurement errors, see Schlüns.[25] The unknown surface S is the graph of function u. This corresponds to an assumption that the surface S is *visible* with respect to the given camera position. In

*Correpondence:
[a]Email: lyle@maths.uwa.edu.au; WWW: http://maths.uwa.edu.au/~lyle; Telephone: +61 89 380 3358; Fax: +61 89 380 1028
[b]Email: ryszard@cs.uwa.edu.au; WWW: http://www.uwa.edu.au/~ryszard; Telephone: +61 89 380 2708; Fax: +61 89 380 1089
[c]Email: r.klette@auckland.ac.nz; WWW: http://citr.auckland.ac.nz/~rklette; Telephone: +64 9 373 7599; Fax: +64 9 373 7001

shading-based shape recovery it is only reasonable to try to determine the surface function u up to shape preserving translations $u + c$ (relative depth values), in contrast to multiple-view situations which may allow to reconstruct absolute depth values. A single image shape-from-shading problem (with no additional constraints) constitutes, in general, an *ill-posed problem.* For this we refer to Brooks and Chojnacki,[4] Brooks, Chojnacki, and Kozera,[2,3,1] Deift and Sylvester,[5] Dupuis and Oliensis,[6,7] Kimmel and Bruckstein,[12] Chapter 7 in Klette, Schlüns, and Koschan,[14] Kozera,[16–18] Oliensis,[21] Onn and Bruckstein[22] and Rouy and Tourin.[23]

In contrast to single image shape-from-shading, the shape of a Lambertian surface is uniquely/generically uniquely determined by a triplet/pair of images. Related shape recovery techniques in computer vision are called *three/two source Photometric Stereo Methods* (2S PSM or 3S PSM). Different shading patterns are obtained by consecutive illuminations of a given surface from three/two different light-sources, see Subsection 10.16 in Horn,[10] Chapter 8 in Klette, Schlüns, and Koschan,[14] Kozera,[16,17] Onn and Bruckstein,[22] and Woodham.[27] The three/two source Photometric Stereo Method, for a Lambertian surface, is modelled by the following system of first-order non-linear PDE's:

$$\frac{p_{i1}u_x(x,y) + p_{i2}u_y(x,y) - p_{i3}}{\sqrt{p_{i1}^2 + p_{i2}^2 + p_{i3}^2}\,\sqrt{u_x^2(x,y) + u_y^2(x,y) + 1}} \cdot \rho(x,y) \;=\; E_i(x,y), \quad (\text{i} = 1,2,3 \text{ or i} = 1,2) \tag{2}$$

defined over an intersection of three/two image domains. Here, for each i as specified above, the corresponding vector (p_{i1}, p_{i2}, p_{i3}) represents a direction vector to a light source. The albedo varies in general for different (x, y) positions, but remains constant for the different images $i = 1, 2, 3$ or $i = 1, 2$.

As it turns out the entire shape reconstruction process can often be decomposed into two independent steps: *gradient computation* (an algebraic step) and *gradient integration* (an analytic step). It can be shown that Photometric Stereo Methods allow that the gradient $\vec{v}_{grad} = (u_x, u_y)$ can be uniquely/generically uniquely expressed in terms of three/two images and light-source directions, see Subsection 10.16 in Horn,[10] Chapter 8 in Klette, Schlüns, and Koschan,[14] Kozera,[16,17] Onn and Bruckstein,[22] and Woodham.[27] Practically, non-uniform albedo, non-Lambertian reflectance (existence of highlights or of a mirroring component), and the presence of different types of shadows are the most critical problems. Several sections in Klette, Schlüns, and Koschan[14] discuss ways of albedo-independent shape recovery, for the problem of eliminating highlights see *e.g.* Schlüns and Teschner,[24] and the presence of shadows is discussed in Schlüns.[26]

Subject to *integrability conditions* a given vector field can be subsequently integrated. In the case of a surface function $u \in C^2(\Omega)$ and Ω being simply connected, the integrability condition reads as

$$u_{xy}(x,y) = u_{yx}(x,y). \tag{3}$$

In the case of a surface function $u \in C^1(\Omega)$, Ω connected and $\gamma_l \subset \Omega$ being a loop, the corresponding integrability condition is

$$\int_{\gamma_l} u_x dx + u_y dy = 0. \tag{4}$$

The reconstructed surface S coincides then with the graph of the surface function u which is defined, up to a constant, by

$$u(x,y) = u(x_0,y_0) + \int_{\gamma} u_x dx + u_y dy, \tag{5}$$

with $\gamma \subset \Omega$ being an arbitrary path connecting a starting point $(x_0, y_0) \in \Omega$ with an end point $(x, y) \in \Omega$.

In this paper, we present a simple geometric global algorithm (we call it the *Lawn-Mowing Algorithm* for obvious reasons) which globally corrects a given, integrable or non-integrable vector field

$$\vec{v}(x,y) = (v_1(x,y), v_2(x,y)).$$

For testing this algorithm we study a non-integrable vector field derived from a genuine gradient vector field subsequently contaminated by noise. This corresponds to situations in image acquisition, where each captured image $E_i(x, y)$ is influenced by the presence of camera noise and further measurement errors.[25]

The techniques developed so-far for integrating a vector field $\vec{v}(x, y)$ are based on finding a surface function that best fits a given imperfect needle diagram, *i.e.* a vector field $\vec{v}(x, y)$ subject to some additional constraints. Klette and Schlüns[13] provide a review of integration techniques as suggested in computer vision, and propose experimental evaluation approaches for testing such techniques. Basically, there are two different categories of approaches, *local path integration techniques* following (5) such as Healey and Jain,[9] Luo and Schlüns,[19] and Wu and Li,[28] or *global optimisation techniques* incorporating the integrability condition (3) as well as further global constraints. Horn proposes a global method, see [10, Subsections 11.7-11.8] or [11], that relies on minimizing different best fit functionals. For example, one of Horn's variants minimizes the following functional

$$\int_{\Omega}((u_x(x,y) - v_1(x,y))^2 + (u_y(x,y) - v_2(x,y))^2)d\Omega \tag{6}$$

whose Euler-Lagrange equation, is the specific Poisson equation $\triangle u = f$. The problem with this is the necessity to provide additional boundary conditions. Such a constraint contravenes somehow the main premise favoring the Photometric Stereo Method as an alternative to single image shape from shading. The latter with no boundary conditions, as opposed to the Photometric Stereo Method, is generically *ill-posed.* In addition, resorting to calculus of variations increases the order of the PDE in question. This in turn, seems to be not natural and not advantageous from the point of view of numerical discretization of differential operators. Another global minimization variant was introduced by Frankot and Chellappa,[8] where the projection to the closest function $\tilde{u}$ expanded by Fourier series is used. A related integration algorithm is detailed in Klette, Schlüns and Koschan.[14] Both the local as well as the global techniques produce incorrect results for surfaces with a large variance in depth, and both are very sensitive to noise.[13] The Frankot-Chellappa method may be the best choice so far. However, a deficiency of this method comes from the implicit requirement that $\tilde{u}$ be periodic.

This paper presents an algorithm which combines, local adaptation with global optimisation. The algorithm is oriented towards *correcting a non-integrable vector field $\vec{v}$ prior to gradient integration.* For the new algorithm neither *a priori* boundary nor periodicity constraints are assumed. The proposed scheme is based on least-square optimization directly applied to the contaminated vector field $\vec{v}(x,y)$, with no recourse whatsoever to variational calculus. A discrete image domain Ω (*i.e.* a homogeneously two-dimensional cellular complex as discussed by Klette and Kovalevsky[15]) is first decomposed into a family of translated subsquares $\{SQ_t\}_{t\in T}$ intersecting along common boundaries. The suboptimal solution is then determined by finding an optimal solution over each $\Omega-$partitioning subsquare. We start with the bottom-left subsquare and scan the whole domain Ω along the horizontal direction. Consequently, the proposed Lawn-Mowing Algorithm, is sequential and obeys a straightforward principle. Namely, upon correcting the values of $\vec{v}$ to $\hat{\vec{v}}$ over a particular SQ_i, the SQ_i gradient boundary values are inherited by adjacent subsquares for which the Lawn-Mowing Algorithm has not been applied so-far. The following discussion is supplemented by a number of (initial) illustrative examples, pictures and pertinent experimental results. In the future the new algorthm should be studied in combining and comparing it with existing integration techniques, especially for real images. In this short note we stay with synthetic examples.

2. PRELIMINARIES

We first recall some classical results and notations used in linear algebra which will be invoked later in this paper. Let $L : \mathbb{R}^n \to \mathbb{R}^m$ be a linear operator (represented by a matrix) with $n > m$. Given $L(\vec{v}) \neq \vec{u}$, for some $\vec{v} \in \mathbb{R}^n$ and $\vec{u} \in \mathbb{R}^m$, the task is to find the closest $\hat{\vec{v}} \in \mathbb{R}^n$ such that $L(\hat{\vec{v}}) = \vec{u}$. It is well-known in linear algebra that $\|\vec{v} - \hat{\vec{v}}\|$ is minimized by the orthogonal projection of $\vec{v}$ on the affine subspace (see Fig. 1)

$$A_L = \{\vec{v}_a \in \mathbb{R}^n : \vec{v}_a = \vec{v}_s + \sum_{i=1}^{k} \alpha_i \vec{f_i}\} ,$$

where vectors $\{\vec{f_1}, \vec{f_2}, \ldots, \vec{f_k}\}$ span $Ker(L) \subset \mathbb{R}^n$, the kernel of L, and $\vec{v}_s$ forms a particular solution to the problem $L(\vec{x}) = \vec{u}$, where $\vec{u} \in \mathbb{R}^m$.

We assume here that $Ker(L) \neq \{\vec{0}\}$. The explicit formula for $\hat{\vec{v}} = \vec{v}_s + \sum_{i=1}^{k} \hat{\alpha}_i \vec{f_i}$ can be found by noting that $\vec{v} - \hat{\vec{v}} \perp \vec{f_j}$, for each $j \in [1, 2, \ldots, k]$, which leads ultimately to the following $k \times k$-linear system:

$$\sum_{i=1}^{k} \hat{\alpha}_i < \vec{f_i}|\vec{f_j} > = < \vec{v} - \vec{v}_s|\vec{f_j} > , \tag{7}$$

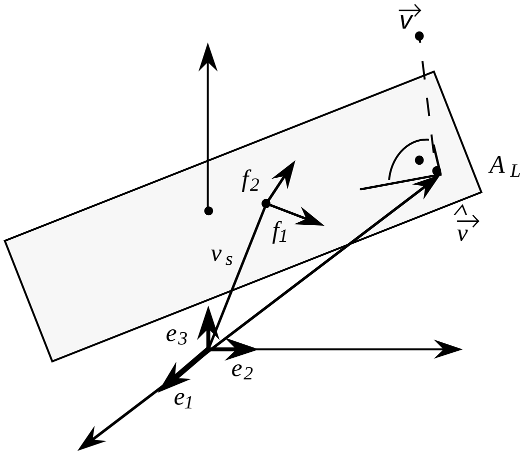


Figure 1. The best integrable fit $\hat{\vec{v}}$ to the inconsistent vector field $\vec{v}$ forms an orthogonal projection of $\vec{v}$ onto an affine subspace A_L.

which in principle be solved by Gaussian elimination. An alternative would be to apply the *pseudoinverse matrix technique*. In doing so the following lemma can be proved:

Lemma 2.1. *The vector $\vec{v} - \hat{\vec{v}}$ is orthogonal to $Ker(L)$ if and only if there exists $\vec{w} \in \mathbb{R}^m$ such that $\vec{v} - \hat{\vec{v}} = L^T(\vec{w})$.*

The proof is here omitted. If we additionally assume that L has maximal $Rank(L) = m$ it is clear that $LL^T : \mathbb{R}^m \to \mathbb{R}^m$ is non-singular. Otherwise, there would exist a vector $\vec{x} \neq \vec{0}$ in $\mathbb{R}^m$ such that $LL^T(\vec{x}) = 0$, and therefore $< LL^T(\vec{x}), \vec{x} > = 0$. Furthermore, $0 = < LL^T(\vec{x}), \vec{x} > = < L^T(\vec{x}), L^T(\vec{x}) >$ implies that $L^T(\vec{x}) = 0$ and as $m = Rank(L) = Rank(L^T)$ we arrive at $\vec{x} = \vec{0}$, a contradiction.

Consequently, since $(LL^T)^{-1}$ is well-defined and as $L(\vec{v}) = \vec{u}$, there exists a *unique* vector $\vec{w}$ such that $\vec{v} - \hat{\vec{v}} = L^T(\vec{w})$. In order to find $\vec{w}$ note that

$$\begin{aligned} \vec{v} - \hat{\vec{v}} &= L^T(\vec{w}) \ , \\ L(\vec{v}) - L(\hat{\vec{v}}) &= LL^T(\vec{w}) \ , \\ L(\vec{v}) - \vec{u} &= LL^T(\vec{w}) \ , \\ \vec{w} &= (LL^T)^{-1}(L(\vec{v}) - \vec{u}) \ . \end{aligned} \tag{8}$$

With this in mind the corrected vector $\hat{\vec{v}}$ stands as

$$\hat{\vec{v}} = \vec{v} - L^T(LL^T)^{-1}(L(\vec{v}) - \vec{u}) \ . \tag{9}$$

The transformation $L_{ps} : \mathbb{R}^m \to \mathbb{R}^n$, where $L_{ps} = L^T(LL^T)^{-1}$, is usually called *pseudoinverse of* L. It should be emphasized, however, that in specific applications, taking into account the computational cost, it might be more efficient to solve directly $L(\vec{v}) - \vec{u} = LL^T(\vec{w})$ and then substitute for $\hat{\vec{v}} = \vec{v} - L^T(\vec{w})$ instead of finding the pseudoinverse matrix L_{ps}. Indeed, theoretically Photometric Stereo Methods may also lead to situations where the direct application of pseudoinverses is practically impossible due to the size of matrices involved - about 70 Gbyte RAM (for more details see Noakes and Kozera[20]).

The main premise favoring the use of our Lawn-Mowing Algorithm refers to the computational difficulty which arises in a corresponding global optimization problem. We show that the integrability condition (4) transformed into its discrete analogue (11) yields, over a discrete image domain Ω, a large system of linear equations in many unknowns. Solving such a system in the least-square sense (see above) constitutes a formidable computational task.

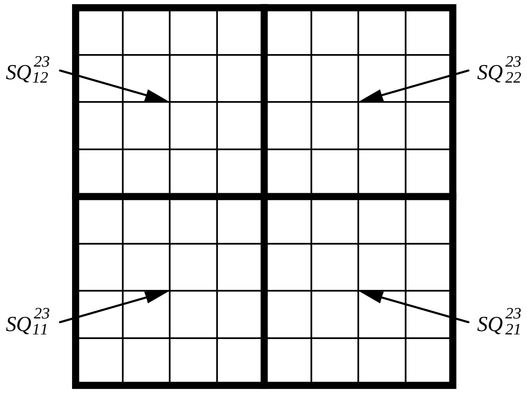


Figure 2. Covering a square image domain Ω by a family $\{SQ^{23}_{ij}\}_{1\leq i,j\leq 2}$ of subsquares. Each subsquare SQ^{23}_{ij} consists of 2^4 atomic squares.

Dividing such a task into local and thus computationally tractable subproblems constitutes a major motivation justifying the derivation of the Lawn-Moving Algorithm. The annotated experimental results which are presented in Section 4, fully support the above expectations and conjectures.

3. THE LAWN-MOWING ALGORITHM

This section introduces the Lawn-Mowing Algorithm designed to correct a given non-integrable vector field $\vec{v}$ into an integrable vector field $\hat{\vec{v}}_{LM} \in A_L$ "close" to $\hat{\vec{v}}$. It only determines a global suboptimal solution. As already mentioned in Section 1, we partition an image domain Ω into translated subdomains and then compute the best local fit to the inconsistent vector field separately over each image subdomain.

Assume that $(k, l) \in \mathbb{N} \times \mathbb{N}$ is a fixed tuple with $k < l$. We divide $\Omega = [0, 1] \times [0, 1]$ into 2^{2l} *atomic subsquares* of the form

$$S^l_{i_q j_g} = [(i_g - 1)/2^l, i_g/2^l] \times [(j_g - 1)/2^l, j_g/2^l]$$

for $1 \leq i_g, j_g \leq 2^l$. In order to decrease the computational cost of a global minimization procedure outlined in the previous section (with $k = l$), we decompose Ω into a family of smaller subsquares each comprising of 2^{2k} atomic subsquares $S^l_{i_q j_g}$. Each subsquare has the form

$$SQ^{kl}_{ij} = [(i - 1)2^{k-l}, i2^{k-l}] \times [(j - 1)2^{k-l}, j2^{k-l}],$$

where $1 \leq i, j \leq 2^{l-k}$. The case when $l = 3$ and $k = 2$ is shown in Fig. 2.

¿From now on we confine our analysis to each SQ^{kl}_{ij} having 2^{2k} atomic squares. Abbreviated we call it S^k in the sequel. We also consider a local Cartesian coordinate system embedded in each subsquare S^k and thus a grid of indexed points (vertices of atomic squares) (i, j), where $0 \leq i, j \leq 2^k$.

The unknown surface function u is considered to be indexed accordingly, *i.e.* u^j_i denotes its value at point (i, j). For $\Delta x = \Delta y = 1/2^{k+1}$, *central-difference approximations* of derivatives are defined as

$$v_x[i, j] = \frac{u^j_{i+1} - u^j_i}{2^k} \quad \text{and} \quad v_y[i, j] = \frac{u^{j+1}_i - u^j_i}{2^k} \tag{10}$$

for each side of a subsquare in x and y directions, respectively whether the side is horizontal or vertical. The case for $k = 1$ is illustrated in Fig. 3.

Note that the y-component of the path integration formula (4), *i.e.* $\int_\gamma u_y dy$ vanishes along each horizontal line (as then $\dot{\gamma}_1(t) = 0$). Similarly the x-component of the path integration formula (4), *i.e.* $\int_\gamma u_x dx$ vanishes along each

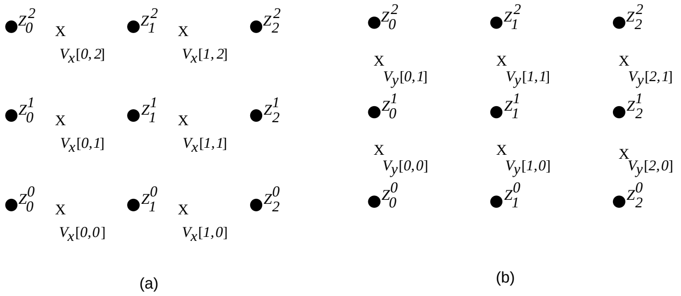


Figure 3. Central difference derivative approximations (when $k = 1$): (a) for u_x and (b) for u_y.

vertical line (as then $\dot{\gamma}_2(t) = 0$). Thus along horizontal lines we consider only u_x derivatives and along vertical lines only u_y derivatives, accordingly. Consequently, along each atomic subsquare, the integrability condition (4) (modulo truncation error assumed to be dominated by noise errors) translates into a simple geometrical form

$$v_x[i,j] + v_y[i+1,j] - v_x[i,j+1] - v_y[i,j] = 0. \tag{11}$$

The latter holds (modulo truncation error) for any integrable vector field.

Reconstructed vector fields using shading-based shape recovery techniques are noisy. For modelling noisy vector fields on synthetic surface functions, we add Gaussian noise, with zero mean and a selected standard deviation, to each $v_x[i,j]$ and to each $v_y[i,j]$, independently. In general, for real as well as for synthetic surfaces, the problem is to reconstruct a surface function u from $2^{k+1}(2^k+1)$ noise contaminated differences. In order to introduce our global Lawn-Mowing Algorithm we consider first four separate cases with different *a priori* constraints imposed on S^k. The next four subsections discuss each case in question.

3.1. Subsquare Grid with no Constraints

In this case we assume that no S^k−boundary constraints have been given. We assign to the non-integrable vector field $\vec{v}_h = (v_x[i,j], v_y[i,j])$ in total $2^{k+1}(2^k+1)$ free variables $\vec{v}_z$ corresponding to the unknown corrected values of the integrable vector field $\hat{\vec{v}}_h$. More specifically, for $u_x[i,j]$ we have $2^k(2^k+1)$ free variables

$$x_1 = v_x[0,0], \ldots, x_{2^k} = v_x[2^k-1,0], x_{2^k+1} = v_x[0,1], \ldots, x_{22^k} = v_x[2^k-1,1],$$
$$x_{2^{2k}} = v_x[0,2^k], \ldots, x_{2^k(2^k+1)} = v_x[2^k-1,2^k].$$

Analogously, for $u_y[i,j]$ we have $2^k(2^k+1)$ variables

$$y_1 = v_y[0,0], \ldots, y_{2^k} = v_y[0,2^k-1], y_{2^k+1} = v_y[1,0], \ldots, y_{2^k} = v_y[1,2^k-1],$$
$$y_{2^{2k}} = v_y[2^k,0], \ldots, y_{2^k(2^k+1)} = v_y[2^k,2^k-1].$$

Applying 2^{2k} integrability constraints (11) (along each atomic square) we arrive at a homogeneous system of 2^{2k} linear equations in $2^{k+1}(2^k+1)$ unknowns. Using notation from Section 2 this system can be treated as

$$L_k^h(\vec{v}_z) = \vec{u}_h \quad \text{for} \quad L_k^h(\vec{v}_h) \neq \vec{u}_h, \text{where } \vec{v}_z, \vec{v}_h \in \mathbb{R}^{2^{k+1}(2^k+1)}, \vec{0} = \vec{u}_h \in \mathbb{R}^{2^{2k}}$$

and

$$L_k^h : \mathbb{R}^{2^{k+1}(2^k+1)} \to \mathbb{R}^{2^{2k}}$$

is a linear operator. Note that for $n = 2^{k+1}(2^k+1)$ and $m = 2^{2k}$ condition $n > m$ holds. It holds $Rank(L_k^h) = m$. Consequently, $\hat{\vec{v}}_h$ can be found as shown in Section 2. Note that as k increases the dimensions of both linear spaces $\mathbb{R}^m$ and $\mathbb{R}^n$ grow exponentially making the computational task not feasible. For that reason the global optimization algorithm has been abandoned and a suboptimal approach is alternatively proposed. More precisely, the original

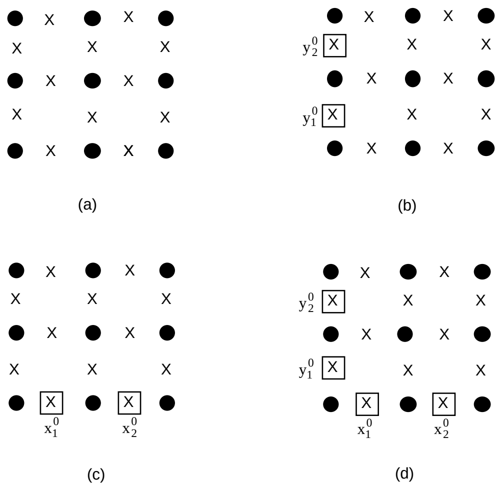


Figure 4. Subsquare grids with (a) no constraints, (b) left constraints, (c) bottom constraints, and (d) left and bottom constraints, for $k = 1$.

$l-$resolution problem is reduced to a collection of computationally tractable $k-$resolution problems, for $k < l$. – In the case of $k = 1$ (see also Fig. 4a) we have $L_1^h(\vec{v}_z) = \vec{u}_h$, where $L_1^h : \mathbb{R}^{12} \to \mathbb{R}^4$ is defined as

$$L_1^h = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix},$$

$\vec{v}_z = (x_1, x_2, x_3, x_4, x_5, x_6, y_1, y_2, y_3, y_4, y_5, y_6)$ and $\vec{u}_h = (0, 0, 0, 0) \in \mathbb{R}^4$. Analogously, for any $1 < k < l$ one can find an explicit formula for L_k^h. For example, for $k = 2$ an explicit formula determining $L_2^h : \mathbb{R}^{40} \to \mathbb{R}^{16}$ can still be quickly established by hand. It is evident, however, that finding the matrix L_k^h (for $k > 2$) consitutes a more laborious task and soon requires the use of a computer.

3.2. Subsquare Grid with Left Constraints

In this case we assume that left S^k-boundary constraints are given. Following the notation from the previous subsection the variables $y_1 = v_y[0,0], \ldots, y_{2^k} = v_y[0, 2^k - 1]$ are assumed to be already assigned to the specific corrected values, namely, $y_1 = y_1^0, y_2 = y_2^0, \ldots, y_{2^k} = y_{2^k}^0$. Note that the noisy vector field $\vec{v}_l$ has now a lower dimension. Indeed, a simple inspection shows that upon using 2^{2k} integrability constraints (11) (along each atomic square) we arrive at an inhomogeneous system of 2^{2k} linear equations in $2^{k+1}(2^k+1) - 2^k = 2^k(2^{k+1}+1)$ unknowns. Applying the notation from Section 2 this system can be treated as $L_k^l(\vec{v}_z) = \vec{u}_l$, for $L_k^l(\vec{v}_l) \neq \vec{u}_l$, where

$$\vec{v}_z, \vec{v}_l \in \mathbb{R}^{2^k(2^{k+1}+1)},\ \vec{0} \neq \vec{u}_l \in \mathbb{R}^{2^{2k}} \text{ and } L_k^l : \mathbb{R}^{2^k(2^{k+1}+1)} \to \mathbb{R}^{2^{2k}}$$

is a linear operator. Note that for $n = 2^k(2^{k+1} + 1)$ and $m = 2^{2k}$ the condition $n > m$ holds. As in the previous subsection it is $Rank(L_k^l) = m$. Consequently, $\hat{\vec{v}}_l$ can be found as shown in Section 2. – In the case of $k = 1$ (see also Fig. 4b) we have $L_1^l(\vec{v}_z) = \vec{u}_l$, where $L_1^l : \mathbb{R}^{10} \to \mathbb{R}^4$ is defined as

$$L_1^l = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & -1 & 0 & 1 \end{pmatrix},$$

$\vec{v}_z = (x_1, x_2, x_3, x_4, x_5, x_6, y_3, y_4, y_5, y_6)$, and $\vec{u}_l = (y_1^0, 0, y_2^0, 0)$. Similarly to the last subsection the other matrices L_k^l, for $k \geq 2$, can be determined with the aid of a computer.

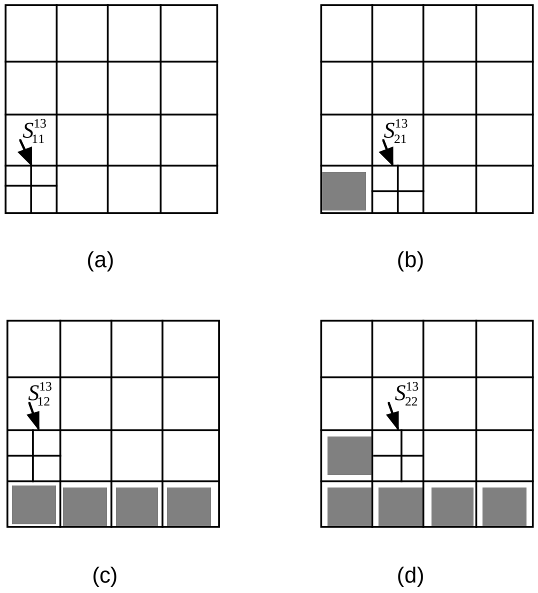


Figure 5. A global suboptimal Lawn-Mowing Algorithm applied for $k = 1$ and $l = 3$.

3.3. Subsquare Grid with Bottom Constraints

In this case we assume that bottom S^k-boundary constraints are given. Similarly to the convention introduced in Subsection 3.2 the variables $x_1 = v_x[0,0], \ldots, x_{2^k} = v_x[2^k-1, 0]$ are assumed now to be already assigned to the "corrected values", namely, $x_1 = x_1^0, x_2 = x_2^0, \ldots, x_{2^k} = x_{2^k}^0$. Similarly to the previous subsection the noisy vector field $\vec{v}_b$ has now a lower dimension. It is evident that 2^{2k} integrability constraints (11) lead to an inhomogeneous system of 2^{2k} linear equations in $2^{k+1}(2^k+1) - 2^k = 2^k(2^{k+1}+1)$ unknowns. Using the notation from Section 2 this system can be treated as $L_k^b(\vec{v}_z) = \vec{u}_b$, for $L_k^b(\vec{v}_b) \neq \vec{u}_b$, where

$$\vec{v}_z, \vec{v}_b \in \mathbb{R}^{2^k(2^{k+1}+1)},\ \vec{0} \neq \vec{u}_b \in \mathbb{R}^{2^{2k}} \text{ and } L_k^b : \mathbb{R}^{2^k(2^{k+1}+1)} \to \mathbb{R}^{2^{2k}}$$

is a linear operator. It it easy to verify that $Rank(L_k^b) = m$. Consequently, $\hat{\vec{v}}_b$ can be found as shown in Section 2. – In the case of $k = 1$ (see also Fig. 4c) we have $L_1^b(\vec{v}_z) = \vec{u}_b$, where $L_1^b : \mathbb{R}^{10} \to \mathbb{R}^4$ is defined as

$$L_1^b = \begin{pmatrix} -1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix},$$

$\vec{v}_z = (x_3, x_4, x_5, x_6, y_1, y_2, y_3, y_4, y_5, y_6)$ and $\vec{u}_b = (-x_1^0, -x_2^0, 0, 0)$. Matrices L_k^b, for $k \geq 2$, as in the previous sections, can be determined.

3.4. Subsquare Grid with Left and Bottom Constraints

Finally, we assume that left and bottom S^k-boundary constraints are provided. Following the notation from Subsection 3.1 the variables $y_1 = v_y[0,0], \ldots, y_{2^k} = v_y[0, 2^k-1]$ are assigned to $y_1 = y_1^0, y_2 = y_2^0, \ldots, y_{2^k} = y_{2^k}^0$ and variables $x_1 = v_x[0,0], \ldots, x_{2^k} = v_x[2^k-1, 0]$ are assigned to $x_1 = x_1^0, x_2 = x_2^0, \ldots, x_{2^k} = x_{2^k}^0$. Clearly, by using 2^{2k} integrability constraints (11) we arrive at an inhomogeneous system of 2^{2k} linear equations in $2^{k+1}(2^k+1) - 2^{k+1} = 2^{2k+1}$ unknowns. Using notation from Section 2 this system can be treated as $L_k^{lb}(\vec{v}_z) = \vec{u}_{lb}$, for $L_k^{lb}(\vec{v}_{lb}) \neq \vec{u}_{lb}$, where

$$\vec{v}_z, \vec{v}_{lb} \in \mathbb{R}^{2^{2k+1}},\ \vec{0} \neq \vec{u}_{lb} \in \mathbb{R}^{2^{2k}} \text{ and the mapping } L_k^{lb} : \mathbb{R}^{2^{2k+1}} \to \mathbb{R}^{2^{2k}}$$

is a linear operator. One can easily show that $Rank(L_k^{lb}) = m$. Consequently, $\hat{\vec{v}}_{lb}$ can be found as shown in Section 2. – In the case of $k = 1$ (see also Fig. 4d) we have $L_1^{lb}(\vec{v}_z) = \vec{u}_{lb}$, where $L_1^{lb} : \mathbb{R}^8 \to \mathbb{R}^4$ is defined as

$$L_1^{lb} = \begin{pmatrix} -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & -1 & 0 & 1 \end{pmatrix},$$

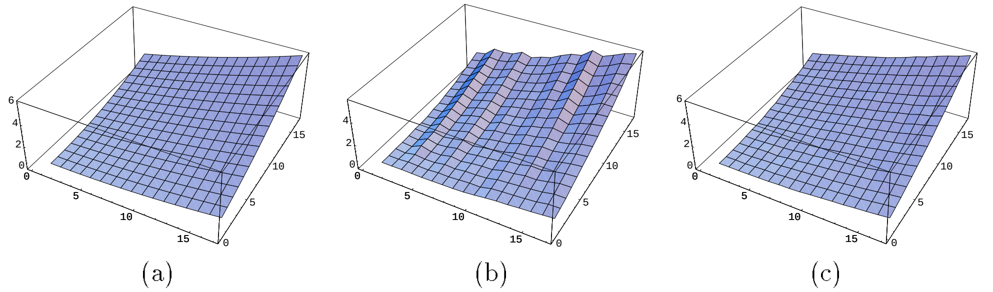
$\vec{v}_z = (x_3, x_4, x_5, x_6, y_3, y_4, y_5, y_6)$, and $\vec{u}_{lb} = (-x_1^0 + y_1^0, -x_2^0, y_2^0, 0)$. Matrices L_k^{lb}, for $k \geq 2$, as in the previous cases can be found.

3.5. Global Suboptimal Lawn-Mowing Algorithm

In this subsection we present a global suboptimal Lawn-Mowing Algorithm by gathering together the so-far discussed cases. We assume that $(k, l) \in \mathbb{N} \times \mathbb{N}$ satisfy $k < l$. The *Lawn-Mowing Algorithm* can be summarised as follows:

- We start with the left bottom subsquare and apply the least-square optimization algorithm with no constraints over SQ_{11}^{kl} (see Subsection 3.1). The computed best fit values of $\hat{\vec{v}}$ over SQ_{11}^{kl} provide the left boundary conditions to the next subsquare SQ_{21}^{kl} (see Fig. 5a, where $l = 3$ and $k = 1$).
- We apply now the least-square optimization algorithm with left constraints (see Subsection 3.2) over SQ_{21}^{kl} (see Fig. 5b). This step is repeated until the last subsquare in the first row, *i.e.* the subsquare $SQ_{2^{l-k}1}^{kl}$, is reached and the optimization algorithm with left constraints is performed over $SQ_{2^{l-k}1}^{kl}$.
- In the sequel, we pass to the second row and apply the least-square optimization algorithm with the bottom constraints inherited from SQ_{11}^{kl} (see Subsection 3.3) over the most left subsquare SQ_{12}^{kl} (see Fig. 5c). Computed values of $\hat{\vec{v}}$ provide the left boundary conditions to the next subsquare SQ_{22}^{kl}. The latter also inherits bottom boundary conditions from the subsquare SQ_{21}^{kl}. Therefore the least-square optimization algorithm with left and bottom constraints (see Subsection 3.4) over SQ_{22}^{kl} (see Fig. 5d) is now applied. The latter step is repeated up until the last subsquare in the second row, *i.e.* the subsquare $SQ_{2^{l-k}2}^{kl}$, is reached and the optimization algorithm with left and bottom constraints is applied.
- The entire procedure is continued over each row until the right top subsqaure $SQ_{2^{l-k}2^{l-k}}^{kl}$ is optimized. The compositions of all local optimal solutions constitute a global suboptimal corrected vector field $\hat{\vec{v}}_{LM}$.

The integrable vector field $\hat{\vec{v}}_{LM}$ rendered by the Lawn-Mowing Algorithm usually does not coincide with the closest integrable vector field $\hat{\vec{v}}$ to $\vec{v}$. In Section 4 it is shown that this algorithm still produces a very good initial estimate of $\hat{\vec{v}}$, at least in the examples considered.



(a) (b) (c)

Figure 6. (a) A graph of a surface function $u(x, y) = x^2 + 3xy + 2y^2$ forming a test surface for shape reconstruction. The reconstructed surface (for $k = l = 4$) computed from (b) a contaminated gradient vector field $\vec{v}$, and (c) a rectified vector field $\hat{\vec{v}}$ via a global least-square optimization algorithm.

4. EXPERIMENTATION AND CONCLUSIONS

We briefly outline the performance of the introduced algorithm. The test programs were run in Mathematica on a PC 233MHz Pentium 2 with 64Mb RAM. In evaluating the Lawn-Mowing Algorithm one global least-square optimization algorithm is performed (when $k = l$) as well as essential cases with $k < l$. Of course, the case $k = l$ is only possible for small test grids. In practice the global optimum is found by some more costly method, such as Leap-Frog Algorithm (see Noakes and Kozera[20]). In doing so the Gaussian noise was added (with a mean set to zero and a standard deviation set to 0.04) to the consistent gradient vector field (u_x, u_y) obtained from the surface function $u : [0,1] \times [0,1] \rightarrow [0,6]$ defined as $u(x,y) = x^2 + 3xy + 2y^2$. The graph of u is shown in Fig. 6a.

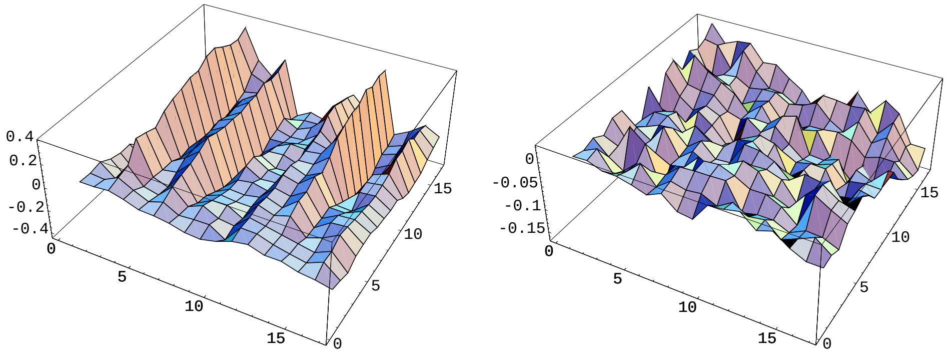
Upon using a standard trapezoidal integration scheme applied to the contaminated vector field $\vec{v}(x,y)$ the surface $\vec{v}_{grad} = (x, y, u_c(x,y))$ is first reconstructed (see Fig. 6b). This result is obviously improved by using a global least-square optimization algorithm correcting $\vec{v}(x,y)$ and then by integrating analogously $\hat{\vec{v}}$. The reconstructed surface $(x, y, u_r(x,y))$ is plotted in Fig. 6c. The absolute errors $u - u_c$ and $u - u_r$ are plotted in Fig. 7a and Fig. 7b, accordingly.

The Lawn-Mowing Algorithm was tested for different parameters k and l. The following run-time results have been obtained:

SQ_t size	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
run-time for $l = 4$	1.07sec.	1.02sec.	4.96sec.	15.74sec.	not applicable
run-time for $l = 5$	3.90sec.	2.38sec.	6.16sec.	65.67sec.	∞ out of memory

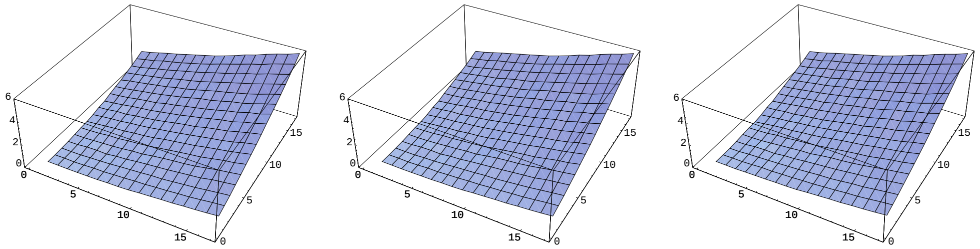
We close this section by emphasizing the following conclusions and observations:

- Note that for $l = 5$ the global optimization fails to find the solution. With a real image resolution of *e.g.* $l = 7 > 5$ a global optimization will clearly fail accordingly. The Lawn-Mowing Algorithm, applied with $k = 3$, alleviates the above "out of memory problem" in run-time performance equal to $31.16\ sec\ << \infty$.
- The algorithm implemented in a more efficient language provides much shorter execution times. The tendency of an improvement, with $k < l$ as opposed to $k = l$, remains however intact. It should also be noticed that for $k_1 < k_2$ the relationship $t^l_{k_1} < t^l_{k_2}$ may not always hold. This was experimentally confirmed for both k_1 and k_2 being close to one. Clearly, the latter reflects the computational trade-off between the number of local optimizations to be performed and the dimension of the S^k-space over which the least-square algorithm is consecutively applied.



(a) (b)

Figure 7. The absolute errors between the ideal surface and the reconstructed one (for $k = l = 4$) computed from (a) the contaminated gradient vector field, and (b) the rectified vector field via a global least-square optimization algorithm.



(a) (b) (c)

Figure 8. The reconstructed surface by using the Lawn-Mowing Algorithm for $l = 4$ and (a) $k = 3$, (b) $k = 2$, and (c) $k = 1$.

- Note that the Lawn-Mowing Algorithm (based on least-square optimization) cannot remove the entire noise from $\hat{\vec{v}}$ and thus is unable to retrieve the genuine $\vec{v}_{grad}$. First of all it does not produce the global optimum $\hat{\vec{v}}_{LM} \neq \hat{\vec{v}}$. Secondly, recall that the omission of the truncation error (treated as negligible) is only a minor contributor to the above mentioned property. Indeed, as Gaussian noise is added in all directions, any perturbation within the affine subspace A_L (comprising integrable vector fields, modulo truncation error) remains undetectable since then $\vec{v} = \hat{\vec{v}}$. The only detectable and removable noise component belongs to the orthogonal complement $A_L^{\perp}$. In the case of a homogeneous system, assuming that $rank(L_k^h) = m$, the ratio of $(dim(ker(L_k^h)))/(2^{k+1}(2^k+1))) = (2^k(2^k+2))/(2^{k+1}(2^k+1)) \approx 0.5$ indicates the proportion of the unremovable noise from the total noise incorporated in a perturbed vector field $\vec{v}$. A similar ratio can be calculated for subsquares SQ_t with various boundary constraints.
- the Lawn-Mowing Algorithm finds a suboptimal solution (but not a local minimum as the least-square has only one critical point being a global minimum) which still renders a very good surface estimate (see Fig. 8). A more sophisticated Algorithm which provably computes a global optimum (called 2-D Leap-Frog Algorithm) is discussed in a paper by Noakes and Kozera.[20]

ACKNOWLEDGEMENTS

The vector field integration project of the authors has been supported by a Gledden Fellowship reward at UWA Nedlands, Perth. The authors thank Mrs. Claire Fowles for helping with drawing figures.

References

[1] M. J. Brooks, W. Chojnacki, and R. Kozera, "Shape without shading," *Quarterly of Applied Mathematics* **50**(1), pp. 27–38, 1992.

[2] M. J. Brooks, W. Chojnacki, and R. Kozera, "Circularly symmetric eikonal equations and non-uniqueness in computer vision," *Journal of Mathematical Analysis and Applications* **165**(1), pp. 192–215, 1992.

[3] M. J. Brooks, W. Chojnacki, and R. Kozera, "Impossible and ambiguous shading patterns," *International Journal of Computer Vision* **7**(2), pp. 119–126, 1992.

[4] M. J. Brooks and W. Chojnacki, "Direct computation of shape from shading," in *Proceedings of the Twelfth International Conference on Pattern Recognition*, pp. 114–119, IEEE Computer Society Press, Los Alamitos, CA, (Jerusalem, Israel, October 9–13), 1994.

[5] P. Deift and J. Sylvester, "Some remarks on the shape-from-shading problem in computer vision," *Journal of Mathematical Analysis and Applications* **84**(1), pp. 235–248, 1981.

[6] P. Dupuis and J. Oliensis, "Direct method for reconstructing shape from shading," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 453–458, IEEE Computer Society Press, Los Alamitos, CA, (Champaign, IL, June 15–18), 1992.

[7] P. Dupuis and J. Oliensis, "An optimal control formulation and related numerical methods for a problem in shape reconstruction," *Annals of Applied Probability* **4**(2), pp. 278–346, 1994.

[8] R. T. Frankot and R. Chellappa, "A method of enforcing integrability in shape from shading algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **10**(4), pp. 439–451, 1988.

[9] G. Healey and R. Jain, "Depth recovery from surface normals," in: Proc. *Int. Conf. Pattern Recognition 1984*, Montreal, July 30-August 2, 1984, pp. 894–896.

[10] B. K. P. Horn, *Robot Vision*, McGraw-Hill, New York, Cambridge, MA, 1986.

[11] B. K. P. Horn, "Height and gradient from shading," *International Journal of Computer Vision* **5**(1), pp. 37–75, 1990.

[12] R. Kimmel and A. Bruckstein, "Global shape from shading," *Computer Vision and Image Understanding* **62**(3), pp. 360–369, 1995.

[13] R. Klette and K. Schlüns, "Height data from gradient fields," in: Proc. *Machine Vision Applications, Architectures, and System Integration*, SPIE 2908, Boston, Massachusetts, 18-19 November 1996, pp. 204–215.

[14] R. Klette, K. R. Schlüns, and A. Koschan, *Computer Vision - Three Dimensional Data from Images*, Springer, Singapore, 1998.

[15] R. Klette, V. Kovalevsky, and B. Yip, "On the length estimation of digital curves," in: SPIE proceedings 3811 (1999).

[16] R. Kozera, "Existence and uniqueness in photometric stereo," *Applied Mathematics and Computation* **44**(1), pp. 1–104, 1991.

[17] R. Kozera, "On shape recovery from two shading patterns," *International Journal of Pattern Recognition and Artificial Intelligence* **6**(4), pp. 673–698, 1992.

[18] R. Kozera, "A note on complete integrals and uniqueness in shape from shading," *Applied Mathematics and Computation* **73**(1), pp. 1–37, 1995.

[19] J. Luo and K. Schlüns, "Height from gradients in polar coordinates", in: Proc. *Image and Vision Computing New Zealand*, CITR Tamaki, Auckland, 16-18 November 1998, pp. 130–135.

[20] L. Noakes and R. Kozera, "A 2-D Leap-Frog Algorithm for optimal surface reconstruction," in: SPIE proceedings 3811 (1999).

[21] J. Oliensis, "Uniqueness in shape from shading," *International Journal of Computer Vision* **6**(2), pp. 75–104, 1991.

[22] R. Onn and A. Bruckstein, "Integrability disambiguates surface recovery in two-image photometric stereo," *International Journal of Computer Vision* **5**(1), pp. 105–113, 1990.

[23] E. Rouy and A. Tourin, "A viscosity solutions approach to shape-from-shading," *SIAM Journal of Mathematical Analysis* **29**, pp. 867–884, 1992.

[24] K. Schlüns and M. Teschner, "Fast separation of reflection components and its application in 3D shape recovery," in: Proc. *3rd Color Imaging Conference*, Scottsdale, Arizona, 7-10 November 1995, pp. 48–51.

[25] K. Schlüns, "The irradiance error and its effect in photometric stereo," in: Proc. *Digital Image & Vision Computing - DICTA'97*, Massey University, Albany/Auckland, 10-12 December 1997, pp. 539 – 544.

[26] K. Schlüns, "Shading based 3D shape recovery in the presence of shadows," in: Proc. *Digital Image & Vision Computing - DICTA'97*, Massey University, Albany/Auckland, 10-12 December 1997, pp. 539 – 544.

[27] R. J. Woodham, "Photometric method for determining surface orientations from multiple images," *Optical Engineering* **19**, pp. 139–144, 1980.

[28] Z. Wu and L. Li, "A line-integration based method for depth recovery from surface normals," *Computer Vision, Graphics and Image Processing* **43**, pp. 53–66, 1988.