

Fast Quadratic Shape From Shading With Unknown Light Direction

This paper was downloaded from TechRxiv (https://www.techrxiv.org).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

30-07-2021 / 05-08-2021

CITATION

Yalpani, Kuros (2021): Fast Quadratic Shape From Shading With Unknown Light Direction. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.15082116.v2

DOI

10.36227/techrxiv.15082116.v2

Fast Quadratic Shape From Shading With Unknown Light Direction

Kuros Yalpani

August 3, 2021

Abstract

An algorithm is proposed that extracts 3D shape from shading information in a digital image. The algorithm assumes that there is only a single source of light producing the image, that the surface of the shape giving rise to the image is Lambertian (matte) and that its shape can be locally approximated by a quadratic function. Previous work shows that under these assumptions, robust shape from shading is possible, though slow for large images because a non-linear optimization method is applied in order to estimate local quadratic surface patches from image intensities. The work presented here shows that local quadratic surface patch estimates can be computed, without prior knowledge of the light source direction, via a linear least squares optimization, thus greatly improving the algebraic complexity and run-time of the existing algorithms.

1 Introduction

Beginning with the work of B. Horn [1, 2, 3], Shape from Shading (SfS), algorithms enjoy a long history in computing science research [4, 5, 6, 7, 8], continue to attract theoretical investigation in the field of computer vision [9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20] and find many practical applications, e.g. in planetary research (*Photoclinometry*), industrial automation, medical imaging or face recognition [21, 22, 23, 24, 25, 26, 27, 28, 29].

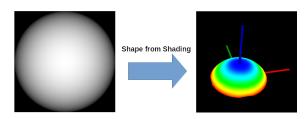


Figure 1: Shape from shading example

The key objective of an SfS algorithm (see figure 1) is to process shading information, appearing as varying gray-scale intensities in a digital image, in order to recover the 3D shape of an object or a set of objects depicted in the image. Often, the resulting shape description serves as further input to some other computer vision related or automated reasoning task. It is well known that humans have the ability to estimate the shape of an object by interpreting shading in their visual scene as a cue, however, exactly how humans accomplish this task is still the subject of active biological research [30]. Like many other computer vision related tasks, SfS methods typically base shape recovery on the physical theory underpinning image

formation [31]. Even so, deducing the shape (often modelled as a height or alternatively a depth field in 3D space) of an object from its corresponding image intensities, e.g. by retracing the path of light from an image location back to the point on the object forming the image, is difficult because in general, the image formation transformation is not, in the mathematical sense, invertible. The fact that image formation reduces the 3D space of a scene to the 2D space of the image and because natural surfaces have varying reflectance properties and a complex geometry, in general implies that image intensity at a candidate point in the image is the aggregate result of the radiance of many other surface points in the scene. Thus, in general the relationship between image intensity and corresponding point on the source object is ambiguous.

1.1 The Lambertian Model

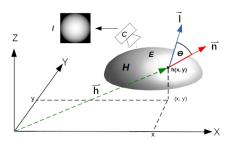


Figure 2: Lambertian model

One way to deal with the ambiguity of image formation is to make assumptions about the problem domain that simplify the image formation model. Consequently, a great deal of theoretical research in SfS has focused on the so called Lambertian model [31] and its related applications. The Lambertian model describes the reflection properties of a matte surface, when it is illuminated by a point source of light. In some natural settings, e.g. marble statues under appropriate illumination, planetary or moon surface features under the illumination of the sun or facial skin under suitable light, to name a few, the Lambertian model can serve as a good approximation. As discovered by J. H. Lambert [32], when a Lambertian surface is illuminated by a point source of light, it appears equally bright in all viewing directions. This type of surface has diffuse radiance, meaning that regardless of its distribution, incident radiation at a point on the surface is scattered in all directions uniformly [33].

As illustrated in figure 2, the Lambertian model has a simple geometric interpretation: Given a Lambertian surface H defined by a so called height function h(x, y), let θ denote the angle between the the normal vector $\vec{\mathbf{n}}$ (at a given point (x, y, h(x, y)) on the surface) and the light vector $\vec{\mathbf{l}}$ (representing a point source of light). Then the

corresponding image intensity I is given by the following equation

$$I = \rho \cos \theta \tag{1}$$

where ρ models surface reflectance (albedo), which often is assumed to be a constant equal to 1. Equation 1 is often represented as the equivalent vector dot product

$$I = \frac{\vec{\mathbf{l}} \cdot \vec{\mathbf{n}}}{\|\vec{\mathbf{l}}\| \|\vec{\mathbf{n}}\|} \tag{2}$$

where $\vec{\mathbf{n}} = (-p, -q, 1)$ (p,q) are the partial derivatives of H w.r.t x,y direction respectively). An implicit assumption in the above model is that the camera C uses orthographic projection (infinite focus) for imaging, which is more suitable for imaging distant objects, instead of the perspective model. The simplicity that results from this is that an image point is in 1-1 correspondence with a scene point. However, there are problems with this model, notably [34] showed that when images are made with orthographic projection, due to an effect called the bas-relief ambiguity, SfS algorithms produce ambiguous results for certain types of image features.

Despite its apparent simplicity, equation 2 is a first order non-linear partial differential equation for H in the differentials p and q. It has no obvious algebraic solution and to date, there is no single general numerical approach for solving it. A multitude of SfS methods have been proposed [7], each with specific (sometimes implicit) constraints on ρ , \vec{l} or the underlying surface geometry in order to achieve certain theoretical results or meet the requirements of their respective applications.

SfS methods also differ in the way they approach the problem numerically. They may apply global energy minimization techniques in order to solve equation 2 simultaneously across all pixels in the image, whereas others may use so called *propagation* methods that progressively expand a region of known height around an initial point until the entire image is covered. Yet other so called local numerical methods compute height at each pixel in the image by combining the image intensities in a small region around a given pixel to a system of constraints that is then solved for each pixel independently. Lastly, while many SfS methods compute height directly at a given pixel, in contrast, some SfS methods apply an intermediate step, so that in a first pass across the image, the surface normal \vec{n} is computed at every image pixel and in a secondary pass, so called normal integration, the resulting vector field is integrated to the surface height (H). Although normal integration is an elementary concept in Calculus, it remains a surprisingly difficult problem to solve numerically and is still the subject of active research [36].

1.2 Locally Quadratic Surfaces

The results presented in this work, are largely based on the image formation model presented in [16], which assumes a single image as input that is produced with orthographic projection and by illuminating a Lambertian surface (H) having constant albedo ($\rho=1$ for all pixels in the image), with a point source of light and significantly H is locally quadratic. According to [16], a quadratic surface is one whose height function (see figure 2) is a bi-variate polynomial of degree two although they do not formally define

what *local* in this case means. However, their underlying intention lies in their approach to SfS, i.e. subdividing the input image into a mosaic of small and overlapping image patches such that each patch is roughly the image of a quadratic surface. The success of the approach is predicated upon a proof they provide, i.e. when H is a quadratic surface that has an invertible Hessian matrix, then equation 2 has a unique solution if the light direction is known, and four distinct solutions if the light direction is unknown. Therefore, given a light source position, a search for a quadratic surface matching a given image patch I_p is guaranteed to succeed. As they show, the search can be limited to points on a conic section that is unique to I_p . Each point on the conic represents a unique quadratic surface proposal for I_p . In a final step, a global numerical optimization method selects all quadratic surface proposals that minimize an overall error relative to the input image.

More recently, [10] also considers the case of SfS for quadratic surfaces. Similar to [16], an approach is presented in order to recover a locally quadratic surface patch, but in contrast to [16], their approach does not require prior knowledge of the light source and is neural network based. They show that the set of quadratic surface proposals that are consistent with the spatial derivatives of intensity at a single image point defines a two-dimensional algebraic variety embedded in the five-dimensional space of quadratic shapes. They devise a neural network, called a point processor, that computes an approximation of the variety from the image intensity and its derivatives at a given image point. Since the point processor, acts only as a constraint on the set of possible quadratic shapes at a point in the image, additional constraints are applied in order to find the unique quadratic shape satisfying the image constraints. They show two alternative ways to do this: In the first case, using two uncalibrated photometric stereo images, i.e. two images of the same scene but varying light vector, they compute for the same pixel in the two different images, the intersection of the respective varieties, using a maximum likelihood measure. They show empirically (but without proof) that the intersection of the varieties approximates a point, hence the unique quadratic surface. In the second approach, a continuity constraint is enforced across a neighborhood of the selected point in the image.

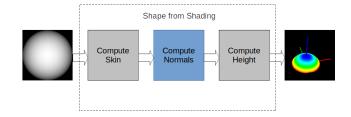


Figure 3: General approach

The work presented in this paper is based on the observation (to be proven in section 3) that in the case of a quadratic surface, image intensities relate linearly to the components of the surface normal. This leads to the following three step approach to SfS (see figure 3):

• Compute skin: or each pixel, a locally quadratic analytic representation of image intensities is computed.

- Compute normal: based on a small region of pixels, and the analytic representations of the previous step, a surface normal is computed via linear least squares.
- Compute height: compute light vector based on local estimates of normal vectors, or alternatively compute surface height globally based on a normal integration algorithm.

The core of the work presented here focuses on step two, namely computing the surface normal, which effectively is the core task in [16] and [10]. Although height recovery is a fundamental post-processing step for the SfS approach presented here, it is considered out of scope. In order to verify the results of this work (computing normals), instead it shown how to recover the light vector. Note that in [16] a non-linear least-squares approximation (Levenberg-Marquardt) is required to do the same, whereas in [10] a neural net approach is required contrasting the purely algebraic approach used here. Refer to the conclusion in section 5 for more on this issue.

2 Definitions

In the following sections, unless specified otherwise, lower case letters generally denote scalar variables and upper case letters denote bi-variate functions over the reals. Furthermore, bold font lower case letters represent column vectors over the real numbers. If $\mathbf{a}_{n\times 1}$ $(n \in \mathbb{N}, n > 0)$ denotes an n dimensional column vector then $\mathbf{a}_{1\times n}^T$ denotes its transpose. For example, an arbitrary coordinate $(x,y) \in \mathbb{R}^2$ is represented by the column vector $\mathbf{x} = [x \ y]^T$. Bold font uppercase letters represent matrices over real numbers. If $\mathbf{H}_{n \times m}$ $(n \in \mathbb{N}, n > 0, m \in \mathbb{N}, m > 0)$ denotes an $n \times m$ dimensional real matrix then $\mathbf{H}_{m \times n}^T$ denotes its transpose. Let $\mathbf{l} = [l_x \ l_y \ l_z]^T$, where $l_x \in \mathbb{R}$, $l_y \in \mathbb{R}$, $l_z \in \mathbb{R}$ and $l_z > 0$, denote a light source. As the algorithm presented here recovers the light direction only, assume further that $\sqrt{l_x^2 + l_y^2 + l_z^2} = 1$. In the remaining sections let $\mathbf{x} = [x \ y]^T (x \in \mathbb{R}, y \in \mathbb{R})$ denote an arbitrary point in \mathbb{R}^2 . Let

$$H(\mathbf{x}) = h_0 + h_1 x + h_2 y + h_3 x^2 + h_4 x y + h_5 y^2,$$

$$\forall i \in \mathbb{N}, 0 < i < 5, h_i \in \mathbb{R} \quad (3)$$

denote the quadratic height function of a surface illuminated by **l**. For convenience, let $\mathbf{h} = [h_0 \cdots h_5]^T$ denote the coefficients of $H(\mathbf{x})$ and be referred to as the *surface coefficients* of H. Furthermore, it is assumed that the Hessian of H is invertible, i.e.

$$4h_3h_5 - h_4^2 \neq 0 \tag{4}$$

By [16] this implies that equation 2 has a unique solution if I is known and four distinct solutions otherwise. Let

$$P(\mathbf{x}) = \frac{\partial H}{\partial x}(\mathbf{x}) = 2h_3x + h_4y + h_1,$$

$$Q(\mathbf{x}) = \frac{\partial H}{\partial y}(\mathbf{x}) = 2h_5y + h_4x + h_2$$
(5)

For convenience and when there is no ambiguity let $p = P(\mathbf{x})$ and $q = Q(\mathbf{x})$. Let $\mathbf{n} = [-p - q \ 1]^T$ denote the vector normal to H at \mathbf{x} . Using the notation defined above,

equation 2 can be expressed as follows:

$$S(\mathbf{x}) = I(\mathbf{x})^2 = \frac{(\mathbf{n} \cdot \mathbf{l})^2}{\|\mathbf{n}\|^2}$$
 (6)

Let $\mathbf{u} = [u_0 \ \cdots \ u_5]^T$, where $\forall i \in \mathbb{N}, 0 \le i \le 5, u_i \in \mathbb{R}$, and let

$$u_0 = h_1^2 + h_2^2 + 1$$
 (7a) $u_3 = 4h_3^2 + h_4^2$ (7d)

$$u_1 = 4h_3h_1 + 2h_4h_2$$
 (7b) $u_4 = 4h_4(h_3 + h_5)$ (7e)

$$u_2 = 2h_4h_1 + 4h_5h_2$$
 (7c) $u_5 = 4h_5^2 + h_4^2$ (7f)

Let

$$U(\mathbf{x}) = u_0 + u_1 x + u_2 y + u_3 x^2 + u_4 x y + u_5 y^2 \quad (8)$$

By substituting the expressions for p and q (equations 5) into the denominator of equation 6 and expansion and simplification of the resulting terms, it is straightforward to show the following identity

$$\|\mathbf{n}\|^2 = p^2 + q^2 + 1 = U(\mathbf{x}) \tag{9}$$

The identity above relates the components of \mathbf{u} , which are solely functions of the surface coefficients to the magnitude squared of the normal vector (\mathbf{n}) at a given image point. Therefore they will be henceforth referred to as the *normal coefficients* of H. Furthermore, since $p^2 + q^2 + 1 > 0$ it also follows that $U(\mathbf{x}) > 0, \forall x \in \mathbb{R}, \forall y \in \mathbb{R}$. Let $d_0 \in \mathbb{R}$, $d_1 \in \mathbb{R}$, $d_2 \in \mathbb{R}$, where

$$d_0 = 2l_x h_3 + l_y h_4 \tag{10}$$

$$d_1 = 2l_u h_5 + l_x h_4 \tag{11}$$

$$d_2 = l_x h_1 + l_y h_2 - l_z (12)$$

By substituting the expressions for p and q (equations 5) into the numerator of equation 6 and expansion and simplification of the resulting terms, it is straightforward to show the following identity

$$(\mathbf{n} \cdot \mathbf{l})^2 = (d_0 x + d_1 y + d_2)^2 \tag{13}$$

Let

$$M(\mathbf{x}) = (d_0 x + d_1 y + d_2)^2 \tag{14}$$

and let $\mathbf{m} = [m_0 \cdots m_5]^T$ where $\forall i \in \mathbb{N}, 0 \le i \le 5, m_i \in \mathbb{R}$, denote the coefficients of the quadratic function by expansion of the terms in $M(\mathbf{x})$, i.e.

$$m_0 = d_2^2$$
 (15a) $m_3 = d_0^2$ (15d)

$$m_1 = 2d_0d_2$$
 (15b) $m_4 = 2d_0d_1$ (15e)

$$m_2 = 2d_1d_2$$
 (15c) $m_5 = d_1^2$ (15f)

Because the components of \mathbf{m} are related to the vector projection $\mathbf{n} \cdot \mathbf{l}$ they are referred to as the *projection coefficients*.

By combining the identities 9 and 14 with equation $\ref{eq:combined}$ it follows that

$$S(\mathbf{x}) = \frac{M(\mathbf{x})}{U(\mathbf{x})} \tag{16}$$

Since, $M(\mathbf{x})$ and $U(\mathbf{x})$ are both quadratic polynomials and furthermore $U(\mathbf{x}) > 0$, $S(\mathbf{x})$ is a rational function

defined everywhere over \mathbb{R}^2 . Furthermore, it is easy to verify, using the quotient rule of differentiation, that all higher order derivatives of $S(\mathbf{x})$ are also rational functions, where the denominator is a power of $U(\mathbf{x})$. Hence, all higher order derivatives of $S(\mathbf{x})$ are defined everywhere in \mathbb{R}^2 . $\forall i \in \mathbb{N}, 0 \le i \le 5$, let $s_i \in \mathbb{R}$ and assume

$$s_0 = S(\mathbf{c})$$
 $s_1 = S_x(\mathbf{c})$ $s_2 = S_y(\mathbf{c})$
 $s_3 = \frac{1}{2}S_{xx}(\mathbf{c})$ $s_4 = S_{xy}(\mathbf{c})$ $s_5 = \frac{1}{2}S_{yy}(\mathbf{c})$ (17)

Furthermore, let $\mathbf{s} = [s_0 \cdots s_5]$. The components of \mathbf{s} are referred to as the *image coefficients* of H at \mathbf{c} .

3 Shape From Shading Using Least Squares

As was noted in the previous section, the values of the components of ${\bf u}$ solely depend on the components of ${\bf h}$. In this section it is shown that that ${\bf u}$ can be computed using a linear least squares approximation that depends only on the known image coefficients and the known pixel position. In section 4 it is then shown how to compute ${\bf h}$ from ${\bf u}$ in a single step.

In the rest of this section, let $\mathbf{c} = [c_x \ c_y], \ c_x \in \mathbb{R}, c_y \in \mathbb{R}$ denote an arbitrary coordinate in \mathbb{R}^2 . Furthermore, $\forall i \in \mathbb{N}, 0 \leq i \leq 5$, let $q_i \in \mathbb{R}$ let

$$Q(\mathbf{x}) = q_0 + q_1 x + q_2 y + q_3 x^2 + q_4 x y + q_5 y^2 \tag{18}$$

denote a general quadratic function of two variables, where $\mathbf{q} = [q_0 \cdots q_5]^T$ denotes the coefficient vector of Q.

Next, several identities with regards to the translation of the coordinate system are proven.

Lemma 3.1 Let $\nabla Q(\mathbf{c})$ and $\nabla^2 Q(\mathbf{c})$ denote the gradient vector and Hessian matrix respectively of Q at \mathbf{c} . Then

$$Q(\mathbf{x}) = Q(\mathbf{c}) + \nabla Q(\mathbf{c})(\mathbf{x} - \mathbf{c})^T + \frac{1}{2} \big[(\mathbf{x} - \mathbf{c}) \nabla^2 Q(\mathbf{c})(\mathbf{x} - \mathbf{c})^T \big]$$

Proof Since $Q(\mathbf{x})$ is a quadratic function of two variables, therefore it is a real analytic function over \mathbb{R}^2 (see proposition 2.2.2 in [37]). Furthermore, because $Q(\mathbf{x})$ is a real analytic function, then it has a Taylor series expansion at \mathbf{c} (see theorem 2.2.5 in [37]). However, since $Q(\mathbf{x})$ is a quadratic function, all terms in the Taylor series expansion of $Q(\mathbf{x})$ at \mathbf{c} of order 3 or greater must be zero, thus the statement of the lemma is proven.

Let

$$\mathbf{D_c} = \begin{bmatrix} 1 & c_x & c_y & c_x^2 & c_x c_y & c_y^2 \\ 0 & 1 & 0 & 2c_x & c_y & 0 \\ 0 & 0 & 1 & 0 & c_x & 2c_y \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Lemma 3.2 Let $\mathbf{q_c}$ denote the coefficient vector of the quadratic function $Q(\mathbf{x} + \mathbf{c})$. Then

$$q_c = D_c \cdot q$$

Proof By lemma 3.1

$$Q(\mathbf{x} + \mathbf{c}) = Q(\mathbf{c}) + \nabla Q(\mathbf{c}) \ \mathbf{x}^{T} + \frac{1}{2} \left[\mathbf{x} \ \nabla^{2} Q(\mathbf{c}) \ \mathbf{x}^{T} \right]$$
(19)

The coefficient vector $\mathbf{q_c}$ of the quadratic function in equation 19 is therefore:

$$\begin{aligned} \mathbf{q_c} &= \begin{bmatrix} Q(\mathbf{c}) \ Q_x(\mathbf{c}) \ Q_y(\mathbf{c}) \ \frac{1}{2} Q_{xx}(\mathbf{c}) \ Q_{xy}(\mathbf{c}) \ \frac{1}{2} Q_{yy}(\mathbf{c}) \end{bmatrix}^T \\ &= \begin{bmatrix} q_0 + c_x q_1 + c_y q_2 + c_x^2 q_3 + c_x c_y q_4 + c_y^2 q_5 \\ q_1 + 2 c_x q_3 + c_y q_4 \\ q_2 + c_x q_4 + 2 c_y q_5 \\ q_3 \\ q_4 \\ q_5 \end{bmatrix} \\ &= \begin{bmatrix} 1 & c_x & c_y & c_x^2 & c_x c_y & c_y^2 \\ 0 & 1 & 0 & 2 c_x & c_y & 0 \\ 0 & 0 & 1 & 0 & c_x & 2 c_y \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{bmatrix} \\ &= \mathbf{D_c} \cdot \mathbf{q} \end{aligned}$$

Note that since $\mathbf{D_c}$ is an upper triangular matrix with diagonal elements consisting of all ones, it follows that $|\mathbf{D_c}|=1>0$. Hence, $\mathbf{D_c}$ has an inverse and is positive definite.

Let

$$\mathbf{P_c} = \begin{bmatrix} s_0 & 0 & 0 & 0 & 0 & 0 \\ s_1 & s_0 & 0 & 0 & 0 & 0 \\ s_2 & 0 & s_0 & 0 & 0 & 0 \\ s_3 & s_1 & 0 & s_0 & 0 & 0 \\ s_4 & s_2 & s_1 & 0 & s_0 & 0 \\ s_5 & 0 & s_2 & 0 & 0 & s_0 \end{bmatrix}$$

Theorem 3.3

$$\mathbf{m} = \mathbf{D}_{\mathbf{c}}^{-1} \cdot \mathbf{P}_{\mathbf{c}} \cdot \mathbf{D}_{\mathbf{c}} \cdot \mathbf{u} \tag{20}$$

Proof From lemma 3.2 it follows that

$$M(\mathbf{x} + \mathbf{c}) = M(\mathbf{c}) + \nabla M(\mathbf{c}) \cdot \mathbf{x} + \frac{1}{2} \left[\mathbf{x}^T \nabla^2 M(\mathbf{c}) \cdot \mathbf{x} \right]$$
(21)

By lemma 3.1, the coefficients of the left hand side of equation 21 are $\mathbf{D_c} \cdot \mathbf{m}$. To prove the theorem, it suffices to show that the coefficients of the right hand side of equation 21 are given by $(\mathbf{P_c} \cdot \mathbf{D_c} \cdot \mathbf{u})$. By the proof of lemma 3.2, the coefficients of the right hand side of equation 21 are:

$$\left[M(\mathbf{c})\ M_x(\mathbf{c})\ M_y(\mathbf{c})\ \frac{1}{2}M_{xx}(\mathbf{c})\ M_{xy}(\mathbf{c})\ \frac{1}{2}M_{yy}(\mathbf{c})\right]^T \qquad (22)$$

From equation 16, it follows that $U(\mathbf{x}) \cdot S(\mathbf{x}) = M(\mathbf{x})$ and the entries in 22 can be rewritten as

$$\begin{bmatrix} U(\mathbf{c})S(\mathbf{c}) \\ U(\mathbf{c})S_x(\mathbf{c}) + U_x(\mathbf{c})S(\mathbf{c}) \\ U(\mathbf{c})S_y(\mathbf{c}) + U_y(\mathbf{c})S(\mathbf{c}) \\ U(\mathbf{c}) \cdot \frac{1}{2}S_{xx}(\mathbf{c}) + U_x(\mathbf{c})S_x(\mathbf{c}) + \frac{1}{2}U_{xx}(\mathbf{c}) \cdot S(\mathbf{c}) \\ U(\mathbf{c})S_{xy}(\mathbf{c}) + U_x(\mathbf{c})S_y(\mathbf{c}) + U_y(\mathbf{c})S_x(\mathbf{c}) + U_{xy}(\mathbf{c})S(\mathbf{c}) \\ U(\mathbf{c}) \cdot \frac{1}{2}S_{yy}(\mathbf{c}) + U_y(\mathbf{c})S_y(\mathbf{c}) + \frac{1}{2}U_{yy}(\mathbf{c}) \cdot S(\mathbf{c}) \end{bmatrix}$$

$$(23)$$

Furthermore, substituting the expressions for the image coefficients (see 17) into the entries of 23 and factoring, results in

$$\begin{bmatrix} s_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_1 & s_0 & 0 & 0 & 0 & 0 & 0 \\ s_2 & 0 & s_0 & 0 & 0 & 0 & 0 \\ s_3 & s_1 & 0 & s_0 & 0 & 0 & 0 \\ s_4 & s_2 & s_1 & 0 & s_0 & 0 & 0 \\ s_5 & 0 & s_2 & 0 & 0 & s_0 \end{bmatrix} \begin{bmatrix} U(\mathbf{c}) \\ U_x(\mathbf{c}) \\ U_y(\mathbf{c}) \\ U_{xx}(\mathbf{c}) \\ U_{xy}(\mathbf{c}) \\ U_{yy}(\mathbf{c}) \end{bmatrix} = \mathbf{P_c} \cdot \mathbf{u_c}$$

By lemma 3.2

$$P_c u_c = P_c \cdot D_c \cdot u$$



Note that $\mathbf{P_c}$ is a lower triangular matrix and is singular iff $s_0 = 0$. Furthermore, if $\mathbf{P_c}$ is non-singular then $|\mathbf{P_c}| = (s_0)^6 > 0$ and therefore is positive definite.

Let $\mathbf{P} = \mathbf{P_0}$. By theorem 3.3, $\mathbf{P_c}$ is similar to \mathbf{P} via the change of basis matrix $\mathbf{D_c}$, i.e.

$$\mathbf{D_c} \cdot \mathbf{P} \cdot \mathbf{D_c}^{-1} = \mathbf{P_c} \tag{24}$$

Furthermore, by theorem 3.3

$$\mathbf{m} = \mathbf{D}_{\mathbf{c}}^{-1} \cdot \mathbf{P}_{\mathbf{c}} \cdot \mathbf{D}_{\mathbf{c}} \cdot \mathbf{u}$$

$$\mathbf{D}_{\mathbf{c}} \cdot \mathbf{m} = \mathbf{P}_{\mathbf{c}} \cdot \mathbf{D}_{\mathbf{c}} \cdot \mathbf{u}$$

$$\mathbf{D}_{\mathbf{c}} \cdot \mathbf{m} = \mathbf{D}_{\mathbf{c}} \cdot \mathbf{P} \cdot \mathbf{u}$$

$$\mathbf{D}_{\mathbf{c}} \cdot \mathbf{P} \cdot \mathbf{u} - \mathbf{D}_{\mathbf{c}} \cdot \mathbf{m} = 0$$
(25)

It is straightforward to transform equation 25 to the following under determined homogeneous linear system in twelve unknowns, i.e. the components of ${\bf u}$ and ${\bf m}$ vectors combined:

$$\begin{bmatrix} \mathbf{D_c} \cdot \mathbf{P} & -\mathbf{D_c} \end{bmatrix}_{6 \times 12} \cdot \begin{bmatrix} \mathbf{u}^T \mathbf{m}^T \end{bmatrix}_{1 \times 12}^T = \mathbf{0}_{1 \times 12}^T$$
 (26)

By selecting enough points in the domain (i.e. in the neighborhood of a candidate point at the center of the coordinate system) it is possible to extend the system represented by equation 26 in order to make it over-determined. For a given $n \in \mathbb{N}, n > 1$, choose a unique set of points $\mathbf{p}_i \in \mathbb{R}^2, i \in \mathbb{N}, 0 \leq i \leq n$ and let $\mathbf{D}_i = \mathbf{D}_{\mathbf{p}_i}$. WLOG assume $\mathbf{p}_0 = \mathbf{0}$ and extend the system in equation 26 to the following:

$$\begin{bmatrix} \mathbf{P} & -\mathbf{I} \\ \mathbf{D}_{1} \cdot \mathbf{P} & -\mathbf{D}_{1} \\ \dots & \dots \\ \mathbf{D}_{n} \cdot \mathbf{P} & -\mathbf{D}_{n} \end{bmatrix}_{(6n) \times 12} \cdot \left[\mathbf{u}^{T} \mathbf{m}^{T} \right]_{1 \times 12}^{T} = \mathbf{0}_{(6n) \times 1}^{T}$$
(27)

Next, it is shown that the over-determined linear system in equation 27 provides sufficient constraints in order to solve for the six (unknown) components of ${\bf u}$.

Theorem 3.4 Assume P is nonsingular and let

$$\mathbf{A} = egin{bmatrix} \mathbf{P} & -\mathbf{I} \ \mathbf{D}_1 \cdot \mathbf{P} & -\mathbf{D}_1 \ \dots & \dots \ \mathbf{D}_n \cdot \mathbf{P} & -\mathbf{D}_n \end{bmatrix}$$

and let $\mathbf{B} = \mathbf{A}^T \cdot \mathbf{A}$. Then \mathbf{B} has a principal minor of rank 6.

Proof By multiplication of the terms of the block matrices \mathbf{A}^T and \mathbf{A} , it is straightforward to show that

$$\mathbf{B} = \mathbf{A}^T \cdot \mathbf{A}$$

$$= \begin{bmatrix} \mathbf{P}^T & \mathbf{P}^T \cdot \mathbf{D}_1^T & \cdots & \mathbf{P}^T \cdot \mathbf{D}_n^T \\ -I^T & -\mathbf{D}_1^T & \cdots & -\mathbf{D}_n^T \end{bmatrix} \cdot \begin{bmatrix} \mathbf{P} & -\mathbf{I} \\ \mathbf{D}_1 \cdot \mathbf{P} & -\mathbf{D}_1 \\ \cdots & \cdots \\ \mathbf{D}_n \cdot \mathbf{P} & -\mathbf{D}_n \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{P}^T \cdot \mathbf{G} \cdot \mathbf{P} & -\mathbf{P}^T \cdot \mathbf{G} \\ -\mathbf{G} \cdot \mathbf{P} & \mathbf{G} \end{bmatrix}$$

where

$$\mathbf{G} = \sum_{i=0}^{n} \mathbf{D}_{i}^{T} \cdot \mathbf{D}_{i} \tag{28}$$

Since $\forall i \in \mathbb{N}, 0 \leq i \leq n, \mathbf{D}_i$ is positive definite, then so is $\mathbf{D}_i \cdot \mathbf{D}_i^T$. Furthermore, since the sum of two positive definite matrices is positive definite, it follows that \mathbf{G} is positive definite. Since \mathbf{P} is nonsingular, it follows that $|\mathbf{P}^T \cdot \mathbf{G} \cdot \mathbf{P}| = |\mathbf{P}^T| \cdot |\mathbf{G}| \cdot |\mathbf{P}| > 0$. Therefore the submatrix $\mathbf{P}^T \cdot \mathbf{G} \cdot \mathbf{P}$, which is a principal minor of \mathbf{B} , is positive definite and has rank 6.

4 Derivation Of Surface Coefficients

In order to compute surface height as well the light direction, the surface normals p and q are required. These are expressed in terms of components of \mathbf{h} (see also the system of equations 5). This section shows how to compute \mathbf{h} from \mathbf{u} , where \mathbf{u} was obtained using the method described in the previous section. Consistent with the results in [16] for the case when the light vector is unknown, upto a set of four distinct solutions for \mathbf{h} are expected.

Since by initial assumption $(4h_3h_4 - h_4^2) \neq 0$ it follows immediately from equations 7b and 7c that there is a unique solution for h_1 and h_2 , i.e.

$$[h_1 \ h_2]^T = \frac{1}{2} \begin{bmatrix} 2h_3 \ h_4 \\ h_4 \ 2h_5 \end{bmatrix}^{-1} [u_1 \ u_2]^T$$
 (29)

Therefore, the main task is to compute h_3 , h_4 and h_5 from equations eqs. (7d) to (7f). First, a method for computing the magnitudes $|h_3|$, $|h_4|$ and $|h_5|$ is presented. Following, it is shown how to resolve their signs. Let $w = h_4^2$. By equations 7d and 7f

$$h_3^2=\frac{u_3-w}{4} \qquad (30) \qquad h_5^2=\frac{u_5-w}{4} \qquad (31)$$
 Therefore, if w (i.e. $|h_4|$) is known then so are $|h_3|$ and

Therefore, if w (i.e. $|h_4|$) is known then so are $|h_3|$ and $|h_5|$ and therefore, the main task reduces to the computation of w. Let $a = u_4^2 + (u_3 - u_5)^2$, $b = -\frac{u_4^2}{2}(u_3 + u_5)$, $c = \frac{u_4^4}{16}$.

Lemma 4.1 w is the solution of the quadratic equation c + b w + a $w^2 = 0$.

Proof The lemma is proven by substituting equations 30

and 31 into equation 7e:

$$\Rightarrow \frac{u_4^2}{16} = h_4^2(h_3^2 + 2h_3h_5 + h_5^2)$$

$$\Rightarrow \left[\frac{u_4^2}{16} - h_4^2(h_3^2 + h_5^2)\right]^2 = 4h_4^4h_3^2h_5^2$$

$$\Rightarrow \left[\frac{u_4^2}{16} - w\left(\frac{u_3 - w + u_5 - w}{4}\right)\right]^2 =$$

$$4w^2\left(\frac{u_3 - w}{4}\right)\left(\frac{u_5 - w}{4}\right)$$

$$\Rightarrow \left[\frac{u_4^2}{4} - w(u_3 + u_5 - 2w)\right]^2 =$$

$$4w^2(u_3 - w)(u_5 - w)$$

$$\Rightarrow \frac{u_4^4}{16} - \frac{u_4^2}{2}w(u_3 + u_5 - 2w) + w^2(u_3 + u_5 - 2w)^2 =$$

$$4w^2u_3u_5 + 4w^2(-(u_3 + u_5)w + w^2)$$

$$\Rightarrow \frac{u_4^4}{16} - \frac{u_4^2}{2}w(u_3 + u_5 - 2w) + w^2(u_3 + u_5)^2 = 4w^2u_3u_5$$

$$\Rightarrow \frac{u_4^4}{16} - \frac{u_4^2}{2}(u_3 + u_5)w + (u_4^2 + (u_3 - u_5)^2)w^2 = 0$$

$$\Rightarrow c + b w + a w^2 = 0$$

By lemma 4.1, $|h_4|$ can be computed by applying the quadratic formula to a, b and c. Subsequently, $|h_3|$, $|h_4|$ can be computed using equations 30, 31. Next, it is shown how to compute the signs of the resulting solution set.

Lemma 4.2
$$u_4 = 0$$
 iff $h_4 = 0$

Proof Note if $h_4 = 0$ then by equation 7e, $u_4 = 0$. Conversely, assume that $u_4 = 0$. By lemma 4.1, two cases need to be considered. Assuming a = 0 then

$$w = -\frac{c}{b} = \frac{u_4^4}{16} \cdot \frac{2}{u_4^2(u_3 + u_5)} = \frac{u_4^2}{4(u_3 + u_5)} = h_4^2$$

$$\iff 4h_4^2(u_3 + u_5) = u_4^2$$
(32)

Hence by equation 32 $u_4 = 0$ iff $h_4 = 0$. Next, assume $a \neq 0$. Then by lemma 4.1 (and after simplification)

$$w = \frac{u_4^2}{4} \left[\frac{(u_3 + u_5) \pm \sqrt{4u_3u_5 - u_4^2}}{a} \right]$$
$$= \frac{u_4^2}{4} \left[\frac{(u_3 + u_5) \pm \sqrt{4u_3u_5 - u_4^2}}{u_4^2 + (u_3 - u_5)^2} \right]$$
(33)

Therefore when $u_4 = 0$

$$h_4^2 = \frac{0}{4} \left[\frac{(u_3 + u_5) \pm \sqrt{4u_3u_5 - 0}}{0 + (u_3 - u_5)^2} \right] = 0$$

By lemma 4.2, u_4 can be used to differentiate between two cases $h_4 = 0$ and $h_4 \neq 0$. When $h_4 = 0$, the following four solutions apply

$$\mathbf{h}_0 = [0 \ h_1 \ h_2 \ + |h_3| \ 0 \ + |h_5|]$$

$$\mathbf{h}_1 = [0 \ h_1 \ h_2 \ - |h_3| \ 0 \ + |h_5|]$$

$$\mathbf{h}_2 = [0 \ h_1 \ h_2 \ + |h_3| \ 0 \ - |h_5|]$$

$$\mathbf{h}_3 = [0 \ h_1 \ h_2 \ - |h_3| \ 0 \ - |h_5|]$$

where the components h_1 and h_2 have been computed using equation 29.

When $h_4 \neq 0$, by lemma 4.1, there are either one $(4u_3u_5-u_4^2=0)$ or two solutions for w. The rest of the section proves that for each possible choice of w, there are only two possible solutions for \mathbf{h} . Hence in total, just like in the case $h_4=0$, there are at most four possible coefficient vectors as a solution.

Lemma 4.3 Let $a \in \mathbb{R}$, $b \in \mathbb{R}$, $c \in \mathbb{R}$, $d \in \mathbb{R}$ where |a| = |c|, |b| = |d|. Assume that $a \cdot b \neq 0$ and $a \cdot b = c \cdot d$. Then either a = c and b = d or a = -c and b = -d.

Proof Let $a \in \mathbb{R}$, $b \in \mathbb{R}$, $c \in \mathbb{R}$, $d \in \mathbb{R}$ where |a| = |c|, |b| = |d|. Assume that $a \cdot b \neq 0$ and $a \cdot b = c \cdot d$. Clearly, when a = b and c = d or a = -b and c = -d, the lemma holds. Assume for the sake of contradiction that a = c and b = -d.

$$\implies a \cdot b = c \cdot d$$
, by initial assumption
 $\implies a \cdot b = a \cdot -b$, by substitution for c and d
 $\implies b = -b$

Since by assumption $a \cdot b \neq 0$ and hence $b \neq 0$, this is a contradiction. Using the same argument it can be shown that a = -c and b = d leads to a contradiction as well. Hence, either a = c and b = d or a = -c and b = -d.

Lemma 4.4 Let $a \in \mathbb{R}$, $b \in \mathbb{R}$, $c \in \mathbb{R}$, $d \in \mathbb{R}$ where |a| = |c|, |b| = |d|. Assume that $a+b \neq 0$ and a+b = c+d. Then a = c and b = d.

Proof Let $a \in \mathbb{R}$, $b \in \mathbb{R}$, $c \in \mathbb{R}$, $d \in \mathbb{R}$ where |a| = |c|, |b| = |d|. Assume that $a + b \neq 0$ and a + b = c + d. Since $a + b \neq 0$, it follows that either $a \neq 0$ or $b \neq 0$. WLOG, assume that $a \neq 0$. If b = 0 then

$$\implies a+b=c+d$$
, by initial assumption
 $\implies a=c$, since $|b|=|d|=0$

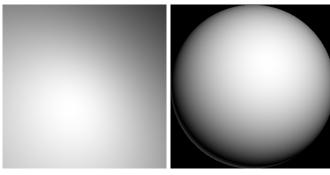
which proves the lemma. Next, assume that $b \neq 0$ and assume that for the sake of contradiction that a=c and b=-d.

$$\implies a+b=c+d$$
, by initial assumption
 $\implies a+b=a+(-b)$, by substitution for c and d
 $\implies b=-b$

which is a contradiction since $b \neq 0$. Using the argument it can be proven that the two other cases a = -c, b = d and a = -c, b = -d lead to contradictions as well. Hence a = c and b = d.

Theorem 4.5 Let $i \in \mathbb{N}$ (i = 3, 4, 5), and let $g_i \in \mathbb{R}$, where $|g_i| = |h_i|$. Assume that $u_4 \neq 0$ and $u_4 = 4g_4(g_3 + g_5)$, then either $g_i = h_i$ (i = 3, 4, 5) or $g_i = -h_i$ (i = 3, 4, 5).

Proof Let $i \in \mathbb{N}$ (i = 3,4,5), and let $g_i \in \mathbb{R}$, where $|g_i| = |h_i|$. Assume that $u_4 \neq 0$ and $u_4 = 4g_4(g_3 + g_5)$. By lemma 4.3 two cases need to be considered. Either $h_4 = g_4$ and $h_3 + h_5 = g_3 + g_5$, in which case it follows from lemma 4.4 that $h_3 = g_3$ and $h_5 = g_5$ or, $h_4 = -g_4$ and $h_3 + h_5 = -(g_3 + g_5) = (-g_3) + (-g_5)$. Again from lemma 4.4 it follows that $h_3 = -g_3$ and $h_5 = -g_5$. Since, either $h_3 = g_3$, $h_4 = g_4$ and $h_5 = g_5$ or $h_3 = -g_3$, $h_4 = -g_4$ and $h_5 = -g_5$, the lemma is proven.



(a) paraboloid $\mathbf{h} = [0\ 0\ 0\ 1\ 0.2\ 1]$

(b) sphere with radius 0.5

Figure 4: images illuminated with $\mathbf{l} = [0.2\ 0.3\ 1]$

5 Results And Discussion

The theory presented in the previous sections is validated by a C++ implementation [38]. It shows that surface normals and light direction can be recovered fast and accurately for the case of a paraboloid and a sphere, but leaves out more complex examples because they require advanced techniques for computing height (see [40]), which is beyond the aim of this paper. Recall from figure 3 that surface height, or alternatively light recovery, is a multi-step process:

- Compute skin: At pixel location, the image coefficients are computed by linear least squares fitting of a second order polynomial to image intensities over a small square window of image.
- Compute normal coefficients: Using the approach outlined in section 3 at each pixel location, normal coefficients are computed from the image coefficients.
- Compute surface coefficients: Using the approach described in section 4 surface coefficients (and by equation 5, surface normals) are computed from normal coefficients.
- Compute light direction (or height): In the final step the light direction is computed using a linear least squares approach by applying equation 6 as a linear constraint at each pixel, in a small square pixel window, on the light vector coefficients l_x , l_y and l_z . The coefficients of the linear constraints, $I(\mathbf{x})$, $p(\mathbf{x})$ and $q(\mathbf{x})$ are known values.

The computed light vector directions for two 400×400 pixel sample images (see figure 4) of a paraboloid and a sphere give the point clouds in figures 5 and 6 respectively.

Table 1 shows the approximate computation times for both images based on a single threaded algorithm (running on an Intel i7-1185G7 processor). These indicate a significant improvement compared to [16], where under the assumption that the light vector is known, they apply a parallelized algorithm (no processor specified) on an 8 core machine, where they claim a one minute processing time for computing local quadratic surface proposals for an 128x128 pixel image.

image	С	u	h	1	total
parab.	0.54s	6.3s	0.13s	0.016s	$7\mathrm{s}$
sphere	0.54s	6.3s	0.16s	0.016s	$7\mathrm{s}$

Table 1: computation times

Table 2 shows the computational accuracy for the computed light vectors.

Recall from section 4 that there are up to 4 different solutions to the light equation. Accordingly (see figures 5 and 6), the point clouds for the light vector computation consist of 4 clusters of points.

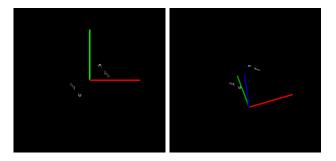


Figure 5: Top and perspective views of point clouds for computed light vector directions for a paraboloid

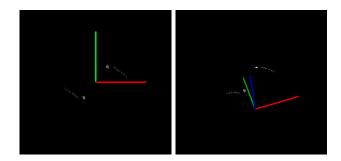


Figure 6: Top and perspective views of point clouds for computed light vector directions for a sphere

When it is known that the image is that of a paraboloid or a sphere, there is no clustering algorithm (e.g. kmeans) required in order to determine the cluster centers (i.e. the desired solutions). Rather it suffices to partition the solution set depending on the sign of the computed second order surface coefficients (i.e. the signs of h_3 , h_4 and h_5 ; see also [38]). In this case, the cluster center is computed as the Euclidean average (center of mass) of the points in a given partition. The accuracy of the result is shown in table 2 as percentage deviation from the ideal vector $\mathbf{l} = [0.2 \ 0.3 \ 1]$ (or normalized [0.19 0.28 0.94]).

image	estimated \mathbf{l}	deviation (θ°)	accuracy
parab.	[-0.27 -0.10 0.96]	35	61%
	$[0.27 \ 0.10 \ 0.96]$	12	87%
	$[-0.17 - 0.24 \ 0.96]$	37	59%
	$[0.17 \ 0.24 \ 0.96]$	3	97%
sphere	[-0.38, -0.19, 0.90]	43	52%
	[0.38, 0.19, 0.90]	12	86%
	[0.19, 0.25, 0.95]	2	98%
	[-0.19, -0.25, 0.95]	38	58%

Table 2: computation accuracy

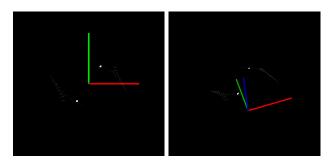


Figure 7: Top and perspective views of point clouds for computed light vector directions for an ideal sphere

Table 3 shows the accuracy of the algorithm in the case of an ideal sphere, i.e. where the image coefficients are generated synthetically as opposed to the standard approach (table 2) where image coefficients are computed in the skinning step with the least squares method. Since in the case of the ideal sphere the accuracy of the computed light vector is nearly 100%, it shows that the loss of accuracy in the standard case is solely due to skinning. Hence, the algorithm outlined in this work requires smooth (locally quadratic) patches and is sensitive to the quality of the analytic description of the source image.

image	estimated l	deviation (θ°)	accuracy
sphere	[-0.46, -0.12, 0.86]	46	49%
	[0.46, 0.12, 0.86]	19	79%
	[0.19, 0.28, 0.94]	0	100%
	[-0.19, -0.28, 0.94]	40	56%

Table 3: Accuracy of light direction estimation in case of an ideal sphere

Finally, when comparing the approach presented here with that of [16], it must be noted that [16] (like many other approaches, see surveys [7] and [8]) combine the step of computing surface normals with that of surface heights into a single numerical step. The approach advocated here clearly separates these two steps into normal finding and so called normal integration (see [36]). That is, the runtime comparison of the algorithm here with that of [16] is not on an equal basis. However, separating the two steps (normal finding and integration) allows the approach presented here to compute the light vector in an efficient manner. This cannot be done with the approach [16]. It remains to be seen if an approach purely based on normal integration combined with the normal finding approach presented here, can outperform the algorithm in [16]. Nonetheless, it seems logical to take these two steps separately, i.e. approaching the shape from shading problem primarily as a normal finding problem and taking normal integration as a separate generalized problem of recovering surface from differentials. Because the latter does not bear any relationship to the non-linear light equation, it may offer a simpler and thus more efficient solution than the combined approach.

References

[1] B. Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. Technical report, MIT, 1970.

- [2] The Psychology of Computer Vision, chapter Obtaining shape from shading information, pages 123–171. McGraw-Hill, 1975.
- [3] Berthold K. P. Horn and Michael J. Brooks The variational approach to shape from shading. In *Computer Vision, Graphics, and Image Processing Vision*, pages 174–208, 1986.
- [4] Ikeuchi K. et al. *Photometric Stereo*, pages 107–123. Springer, Cham, 2020.
- [5] Arcangelo Distante and Cosimo Distante. Shape from Shading, pages 413–478. Springer International Publishing, Cham, 2020.
- [6] Breuß M. and Mansouri Yarahmadi A. *Perspective Shape from Shading*. Springer, Cham., 2020.
- [7] Jean-Denis Durou, Maurizio Falcone, and Manuela Sagona. Numerical methods for shape-from-shading: A new survey with benchmarks. *Computer Vision and Image Understanding*, 109(1):22 43, 2008.
- [8] Ruo Zhang, Ping-Sing Tsai, J. E. Cryer, and M. Shah. Shape-from-shading: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, 1999.
- [9] B. Haefner, S. Peng, A. Verma, Y. Quéau, and D. Cremers. Photometric depth super-resolution. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(10):2453-2464, 2020.
- [10] K. Heal, J. Wang, S. J. Gortler, and T. Zickler. A lighting-invariant point processor for shading. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 91–99, 2020.
- [11] Paul Henderson and Vittorio Ferrari. Learning singleimage 3d reconstruction by generative modelling of shape, pose and shading. *International Journal of Computer Vision*, 128(4):835–854, Apr 2020.
- [12] Xu Cao, Michael Waechter, Boxin Shi, Ye Gao, Bo Zheng, and Yasuyuki Matsushita. Stereoscopic flash and no-flash photography for shape and albedo recovery. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.
- [13] Min Qi, Yingbo Wang, and Yuelei Xu. Research on height restoration in shape-from-shading method. In Proceedings of the 2019 2nd International Conference on Watermarking and Image Processing, ICWIP 2019, page 50-54, New York, NY, USA, 2019. Association for Computing Machinery.
- [14] Dawei Yang and Jia Deng. Shape from shading through shape evolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [15] Daniel Maurer, Yong Chul Ju, Michael Breuß, and Andrés Bruhn. Combining shape from shading and stereo: A joint variational method for estimating depth, illumination and albedo. *International Journal* of Computer Vision, 126(12):1342–1366, Dec 2018.

- [16] Y. Xiong, A. Chakrabarti, R. Basri, S. J. Gortler, D. W. Jacobs, and T. Zickler. From shading to local shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(1):67–79, 2015.
- [17] Hiroaki Santo, Masaki Samejima, and Yasuyuki Matsushita. Numerical shape-from-shading revisited. IPSJ Transactions on Computer Vision and Applications, 10(1):8, Jun 2018.
- [18] Bjoern Haefner, Yvain Quéau, Thomas Möllenhoff, and Daniel Cremers. Fight ill-posedness with ill-posedness: Single-shot variational depth super-resolution from shading. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [19] Daniel N. Holtmann-Rice, Benjamin S. Kunsberg, and Steven W. Zucker. Tensors, differential geometry and statistical shading analysis. *Journal of Mathe*matical Imaging and Vision, 60(6):968–992, Jul 2018.
- [20] Cheffler R., Mansouri Yarahmadi A., Breuß M., and Köhler E. Fa graph theoretic approach for shape from shading. In Energy Minimization Methods in Computer Vision and Pattern Recognition, 2018.
- [21] M Hess, K Wohlfarth, A Grumpe, C Wöhler, O Ruesch, and B Wu. Atmospherically compensated shape from shading on the martian surface: towards the perfect digital terrain model of mars. In *Interna*tional archives of the photogrammetry, remote sensing and spatial information sciences, 2019.
- [22] Elodie Lesage, Frédéric Schmidt, François Andrieu, and Hélène Massol. Constraints on effusive cryovolcanic eruptions on europa using topography from galileo images, 2020.
- [23] Oleg Alexandrov and Ross A. Beyer. Multiview shape-from-shading for planetary images. *Earth and Space Science*, 5(10):652–666, 2018.
- [24] S. Wang, T. Wu, K. Wang, Z. Peng, N. M. Kwok, and T. Sarkodie-Gyan. 3d particle surface reconstruction from multi-view 2d images with structure from motion and shape from shading. *IEEE Transactions on Industrial Electronics*, pages 1–1, 2020.
- [25] Steve Zhu. Shape from shading: A vital shape extraction technology and its applications. *Quality, suppl. VISION & SENSORS*, pages 8–9, 2020.
- [26] A. Pinto, S. Goldenstein, A. Ferreira, T. Carvalho, H. Pedrini, and A. Rocha. Leveraging shape, reflectance and albedo from shading for face presentation attack detection. *IEEE Transactions on Infor*mation Forensics and Security, 15:3347–3358, 2020.
- [27] Information Technology in Biomedicine. Advances in Intelligent Systems and Computing, chapter Evaluation of Shape from Shading Surface Reconstruction Quality for Liver Phantom. Springer, Cham., 2021.
- [28] Y. Zhang, C. Zhang, W. Wang, Y. Chen, Z. Ji, and H. Liu. Portrait relief modeling from a single image. *IEEE Transactions on Visualization and Computer Graphics*, 26(8):2659–2670, 2020.

- [29] L. Jiang, J. Zhang, B. Deng, H. Li, and L. Liu. 3d face reconstruction with geometry details from a single image. *IEEE Transactions on Image Processing*, 27(10):4756-4770, 2018.
- [30] Phillip J. Marlow, Scott W.J. Mooney, and Barton L. Anderson. Photogeometric cues to perceived surface shading. Current Biology, 29(2):306 – 311.e3, 2019.
- [31] Arcangelo Distante and Cosimo Distante. *Radiometric Model*, pages 57–77. Springer International Publishing, Cham, 2020.
- [32] Photometria, Sive De Mensura Et Gradibus Luminis, Colorum Et Umbrae (1760). Nabu Press, 2012.
- [33] Machine Vision, chapter 9. McGraw-Hill, Inc., 1995.
- [34] Peter N. Belhumeur, David J. Kriegman, and Alan L. Yuille. The bas-relief ambiguity. *International Journal of Computer Vision*, 35(1):33–44, Nov 1999.
- [35] E. Prados and O. Faugeras. Shape from shading: a well-posed problem? In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), volume 2, pages 870–877 vol. 2, 2005.
- [36] Yvain Quéau, Jean-Denis Durou, and Jean-François Aujol. Normal integration: A survey. Journal of Mathematical Imaging and Vision, 60(4):576–593, May 2018.
- [37] Steven Krantz and Harold Parks. A Primer of Real Analytic Functions. 01 2002.
- [38] https://gitlab.com/kyalpani/sfs.
- [39] Les Piegl and Wayne Tiller. The NURBS Book. Springer-Verlag, Berlin, Heidelberg, 1995.
- [40] Min Qi, Yingbo Wang, and Yuelei Xu. Research on height restoration in shape-from-shading method. In Proceedings of the 2019 2nd International Conference on Watermarking and Image Processing, ICWIP 2019, page 50-54, New York, NY, USA, 2019. Association for Computing Machinery.
- [41] Ira Kemelmacher-Shlizerman and Ronen Basri. 3d face reconstruction from a single image using a single reference face shape. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 33, 2011.
- [42] R. B. Rusu and S. Cousins. 3d is here: Point cloud library (pcl). In 2011 IEEE International Conference on Robotics and Automation, pages 1–4, 2011.
- [43] Sven Loncaric. A survey of shape analysis techniques. $Pattern\ Recognition,\ 31(8):983-1001,\ 1998.$
- [44] Dizhong Zhu and William A P Smith. Least squares surface reconstruction on arbitrary domains, 2020.
- [45] Ronen Basri, David Jacobs, and Ira Kemelmacher. Photometric stereo with general, unknown lighting. *International Journal of Computer Vision*, 72(3):239–257, May 2007.

- [46] R. Kozera, L. Noakes, and S. Collings. A piecewise quadratic approach to single image shape from shading. In *Pattern Recognition, International Conference on*, volume 4, page 40126, Los Alamitos, CA, USA, aug 2002. IEEE Computer Society.
- [47] A. Pentland. Photometric motion. In [1990] Proceedings Third International Conference on Computer Vision, pages 178–187, 1990.