**Monograph:**
Junxiao, Gu. and Kadirkamanathan, V. (1994) Shape From Shading: A B-Splines Based Algorithm. Research Report. ACSE Research Report 545 . Department of Automatic Control and Systems Engineering

# SHAPE FROM SHADING: A B-SPLINES BASED ALGORITHM

**Gu Junxiao**
Jinhang Research Institute of Computing Technology
Ministry of Astronautics Industry of China

**Visakan Kadirkamanathan**
Department of Automatic Control & Systems Engineering
University of Sheffield
PO Box 600, Mappin Street, Sheffield S1 4DU, UK
Email: visakan@acse.sheffield.ac.uk

Research Report No. **545**

Department of Automatic Control & Systems Engineering
The University of Sheffield

October 1994

# SHAPE FROM SHADING: A B-SPLINES BASED ALGORITHM

**Gu Junxiao**
Jinhang Research Institute of Computing Technology
Ministry of Astronautics Industry of China

**Visakan Kadirkamanathan**
Automatic Control & Systems Engineering Department
University of Sheffield, UK

**Abstract**

In this paper, we propose a B-spline based approach to the *shape from shading* problem. Its basis lies in approximating a smooth surface by uniform bicubic B-spline basis functions. Since the surface normal in a patch is uniquely determined by heights of its sixteen neighbour vertices, the image brightness is directly related to the control vertices. The control vertices are then determined by minimising a cost functional corresponding to the squared brightness error, optimised by conjugate gradient method. The proposed algorithm does not require any integrability constraint or knowledge of boundary conditions. Using several experiments with synthetic images, we demonstrate the performance of the algorithm.

# 1 Introduction

The problem of shape from shading is that of recovering the 3-D surface shape from a single 2-D shaded image data based on the *reflectance map* and in some cases, the boundary orientation. Though shape from shading appears a simple task for the brain, it is quite a difficult task for machines. Much of the early research into shape from shading, formulated and pioneered by Horn [1], [2], [3], [4], [5], [6], was aimed at obtaining the surface orientation. The variational approach [3], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], in which a surface orientation field is characterised by its slopes $p(x,y) = \partial z/\partial x$ and $q(x,y) = \partial z/\partial y$ has been used to determine the orientation.

The variational approach results in a set of first-order partial differential equations. The difficulty in solving these equations are two-fold. Firstly, appropriate boundary conditions are required and secondly, the nonlinear equations are solved by iterative algorithms whose convergence properties are not well understood [5], [12], [17]. The algorithm suffers due to the non-integrability of computed $p(x,y)$ and $q(x,y)$ and the ill-posedness of the problem [11]. A different approach in which the height of the surface was extracted from the image was proposed by Pentland [9], [10]. He related the height to image brightness in closed form, with a linearised reflectance map in the Fourier Transform domain.

We propose a new B-spline based approach to shape from shading in this paper. It stems from the idea of approximating a order-2 continuous surface by a linear combination of a set of uniform bicubic B-spline basis functions. Since the surface normal is determined by its sixteen neighbour control vertices, we can relate the image brightness directly to the height of the control vertices via the reflectance map. A cost function is defined in terms of the squared brightness error and a regularisation term that minimise the *bending energy* [18] to make the problem well-posed.

Our approach is similar to that of Lee and Kuo [19], but major differences exist. Firstly, they use triangular surface patches as basis functions which are not $C^2$ continuous and hence computing the derivatives are difficult. This forces them to use linearised reflectance map in relating image brightness to nodal heights. In our case, since bicubic B-splines are $C^2$ continuous, the exact relationship between the height, brightness and the reflectance map can be established, which is the second difference. Finally, we use conjugate gradient optimisation which is computationally demanding than the multigrid technique used by Lee and Kuo [19]. The computational complexity is justified on the grounds of seeking the true optimum set of parameters for shape reconstruction. Note that Bolle and Cooper [20] use global quadratic polynomials as basis functions to represent, not the shape, but the image itself and then transform the estimated coefficients or parameters for comparison with those of known shapes.

## 2 Image irradiance equation and image formation

We assume that the shaded image is formed by an orthographic projection of a Lambertian surface with a distant single point light source, as in [21]. The surface shape is described in the cartesian coordinates $(x, y, z)$ and expressed by the equation $z = z(x, y)$. With the viewing direction aligned with the negative $z$-axis, under the orthographic projection of the surface, the shaded image coordinates are also $(x, y)$. The surface orientation is specified in the gradient space $(p, q)$, where the surface normal is $[p, q, -1]$, given by,

$$p(x,y) = \frac{\partial z(x,y)}{\partial x} \qquad q(x,y) = \frac{\partial z(x,y)}{\partial y} \tag{1}$$

Image formation can be described by a single equation called the *image irradiance equation* [6], [21], where image brightness is related to surface orientation through the reflectance map. The reflectance map provides information on the reflection property of the surface and of the light sources. With gradient space used to represent surface orientation, the image irradiance equation is given by,

$$I(x,y) = R(p,q) \tag{2}$$

where $I(x, y)$ is the brightness at a given point $(x, y)$ and $R(p, q)$ is the reflectance map. Under the assumption of orthographic projection (Lambertian surface and distant single point light source), the form of the reflectance map is usually given as,

$$R(p,q) = \eta \frac{1 + p_s p + q_s q}{\sqrt{1+p^2+q^2}\sqrt{1+p_s^2+q_s^2}} = \eta \frac{k_\tau + k_p p + k_q q}{\sqrt{1+p^2+q^2}} \tag{3}$$

where $\eta$ is the *albedo* of the surface, $(p, q, -1)$ is the gradient of the surface at $(x, y)$, $(p_s, q_s, -1)$ is the illumination direction and,

$$\begin{aligned} k_p &= \frac{p_s}{\sqrt{1+p_s^2+q_s^2}} = \sin\tau \ \sin\sigma \\ k_q &= \frac{q_s}{\sqrt{1+p_s^2+q_s^2}} = \cos\tau \ \sin\sigma \\ k_\tau &= \frac{1}{\sqrt{1+p_s^2+q_s^2}} = \cos\sigma \end{aligned} \tag{4}$$

with $\tau$, $\sigma$ being the tilt and slant angles made by illumination direction with the $x$–axis and $z$–axis respectively.

Horn *et al.*, [3], [4], [5] estimated the surface orientation $\mathbf{n} = (p, q)$ at each image point directly, the underlying theme being the satisfaction of the image irradiance equation $I(x, y) - R(p, q) = 0$ (2) or at least the minimisation of the difference over the image domain $\Omega$.

$$\mathbf{n}_* = \arg\min_{\mathbf{n}} \int_\Omega \{I(x,y) - R(\mathbf{n}(x,y))\}^2 \, dx \, dy \tag{5}$$

This is an ill-posed problem which is made well-posed by including regularisation terms [22].

We adopt an alternative, but similar, approach where the shape of the surface is described in terms of a set of parameters, say $\mathbf{a}$ (*ie.*, $z = z(x, y; \mathbf{a})$), then the optimal set of parameters $\mathbf{a}_*$ are obtained by minimising a cost function as in (5) including a regularisation term in which the parameter $\mathbf{a}$ appears through the $p$, $q$ terms in the reflectance map $R(p, q)$ given by (3). In the next section, we look at a particular surface shape description and the set of parameters that need to be optimised.

## 3 Surface description with B-splines

A general surface can be described as a linear combination of a set of basis functions $B_{i,j}$ as,

$$z(x, y) = \sum_{i=0}^{N} \sum_{j=0}^{N} v_{i,j} B_{i,j}(x, y) \tag{6}$$

where $v_{i,j}$ are the coefficients known as control vertices.

To describe a 2-D surface, the control vertices are placed on a topologically rectangular array $(i, j)$, called the *control mesh* or *control graph* over the image plan, shown in Figure 1. For a given image of $2(N-3) \times 2(N-3)$, the number of control vertices are $N \times N$.

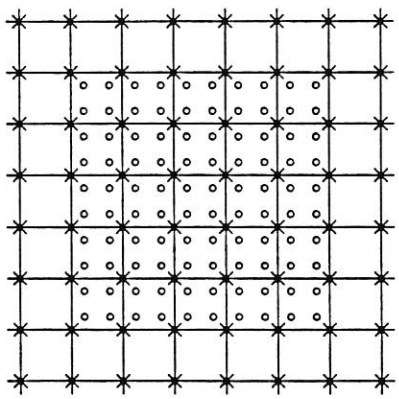

Figure 1: Control vertices and Image pixels: × denotes the position of control vertices and ○ indicates image pixels.

The surface being recovered from the shading image is formed by scaling the sum of basis functions over the image domain $\Omega$. The scale factors are control vertices.

The two dimensional basis functions are formed from the one dimensional uniform bicubic B-splines as *tensor-product B-spline*, given by [23],

$$B_{i,j}(x, y) = B_i(x) B_j(y) \tag{7}$$

The basis $B(x)$ is non-zero over only four successive intervals as shown in Figure 2. This

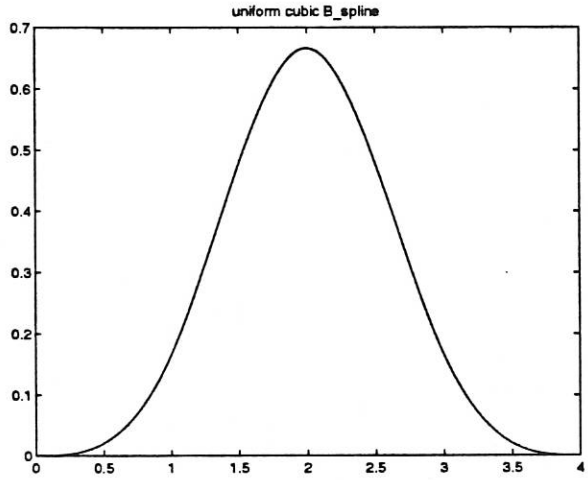


Figure 2: The uniform cubic B-spline. The non-zero portion is composed of four polynomial segments $b_3(x), b_2(x), b_1(x), b_0(x)$.

non-zero portion of the cubic B-spline consists, from right to left, four basis segments $b_3(x)$, $b_2(x)$ ,$b_1(x)$, $b_0(x)$, given by, [23],

$$\begin{aligned} b_0(x) &= \tfrac{1}{6}x^3 \\ b_1(x) &= \tfrac{1}{6}\left(1 + 3x + 3x^2 - 3x^3\right) \\ b_2(x) &= \tfrac{1}{6}\left(4 - 6x^2 + 3x^3\right) \\ b_3(x) &= \tfrac{1}{6}\left(1 - 3x + 3x^2 - x^3\right) \end{aligned} \tag{8}$$

They combine to form a piecewise cubic polynomial curve which has positional, first derivative and second derivative continuity ($C^2$ continuity) at the joints between successive segments. The 2-D basis function $B_{i,j}(x, y)$ is shown in Figure 3 and is non-zero over a $4 \times 4$ region in the image plane.

Using the bicubic B-spline basis functions, any $C^2$ surface can be described by,

$$\begin{aligned} z(x,y) &= \sum_{i=0}^{N}\sum_{j=0}^{N} v_{i,j} B_i(x) B_j(y) \\ &= \sum_{i=0}^{N}\sum_{j=0}^{N}\left[\sum_{r=0}^{3}\sum_{s=0}^{3} v_{i+r,j+s} b_r(x) b_s(y)\right] \end{aligned} \tag{9}$$

where $x, y \in [0, 1]$. The height at position $(x, y)$ inside the square patch is uniquely determined by its sixteen neighbour control vertices $V_{ij} = [v_{i,j}, v_{i,j+1}, \ldots, v_{i+3,j+3}]^T$, shown in Figure 4.

It is straight forward to see that the partial derivatives of the surface with respect to (w.r.t) $x$ and $y$ at any point inside this square patch is also affected by its sixteen control

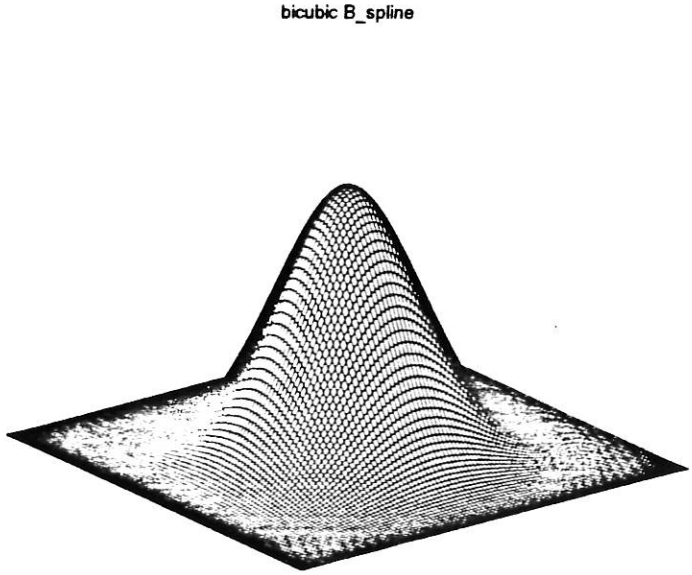


Figure 3: The non-zero portion of a single uniform bicubic B-spline $B_{0,0}(x,y)$, formed by, uniform cubic B-splines $B_0(x), B_0(y)$.

vertices as follows:

$$p_{ij}(x,y) = \sum_{r=0}^{3}\sum_{s=0}^{3} v_{i+r,j+s}\frac{\partial b_r(x)}{\partial x}b_s(y) = \mathbf{c}_x(x,y)V_{ij} \tag{10}$$

$$q_{ij}(x,y) = \sum_{r=0}^{3}\sum_{s=0}^{3} v_{i+r,j+s}b_r(x)\frac{\partial b_s(y)}{\partial y} = \mathbf{c}_y(x,y)V_{ij} \tag{11}$$

where $\mathbf{c}_x(x,y)$, $\mathbf{c}_y(x,y)$ are $(16\times 1)$vectors which depend only on the image point $(x,y)$. Note that $\mathbf{c}_x(x,y)$ is given by,

$$\mathbf{c}_x(x,y) = [C_x^{xy}(0,0),\ldots,C_x^{xy}(r,s),\ldots C_x^{xy}(3,3)]^T \tag{12}$$

and

$$C_x^{xy}(r,s) = \frac{\partial b_r(x)}{\partial x}b_s(y) \tag{13}$$

For notational convenience, let us define the matrix $\mathbf{C}_x^{xy}$ as a $4\times 4$ matrix with components from $\mathbf{c}_x(x,y)$ using $(r,s)$ as the matrix elements, *ie.*,

$$\mathbf{C}_x^{xy} = \begin{bmatrix} C_x^{xy}(0,0) & C_x^{xy}(0,1) & C_x^{xy}(0,2) & C_x^{xy}(0,3) \\ C_x^{xy}(1,0) & C_x^{xy}(1,1) & C_x^{xy}(1,2) & C_x^{xy}(1,3) \\ C_x^{xy}(2,0) & C_x^{xy}(2,1) & C_x^{xy}(2,2) & C_x^{xy}(2,3) \\ C_x^{xy}(3,0) & C_x^{xy}(3,1) & C_x^{xy}(3,2) & C_x^{xy}(3,3) \end{bmatrix} \tag{14}$$

The terms $\mathbf{c}_y(x,y)$ and $\mathbf{C}_y^{xy}$ can be similarly expressed. For a given image point $(x,y)$ inside the $(i,j)$ square patch, the surface orientation $p_{ij}(x,y)$ and $q_{ij}(x,y)$ are scaled combination of sixteen neighbour vertices.

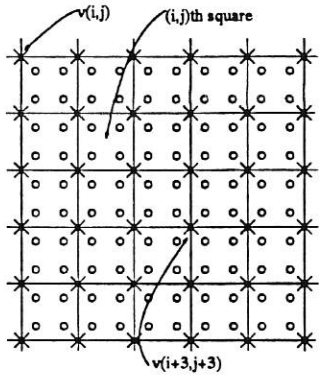


Figure 4: Control vertices, Image pixels for the $(i,j)$ Square Patch.

The second order derivatives of the surface are given by,

$$\begin{aligned}
\frac{\partial^2 z_{ij}(x,y)}{\partial x^2} &= \sum_{r=0}^{3}\sum_{s=0}^{3} v_{i+r,j+s}\frac{\partial^2 b_r(x)}{\partial x^2}b_s(y) \\
\frac{\partial^2 z_{ij}(x,y)}{\partial y^2} &= \sum_{r=0}^{3}\sum_{s=0}^{3} v_{i+r,j+s}b_r(x)\frac{\partial^2 b_s(y)}{\partial y^2} \\
\frac{\partial^2 z_{ij}(x,y)}{\partial x \partial y} &= \sum_{r=0}^{3}\sum_{s=0}^{3} v_{i+r,j+s}\frac{\partial b_r(x)}{\partial x}\frac{\partial b_s(y)}{\partial y}
\end{aligned} \tag{15}$$

These second order derivatives are used in obtaining the regularisation term.

The B-spline representation of the surface can be viewed as a parametrised representation with the control vertices representing the set of unknown parameters for the given shape. In the next section, we will formulate the B-spline based algorithm as an optimisation task in which the optimal set of control vertices that satisfy the irradiance equation is sought.

# 4 The B-spline based algorithm

The given image of $2(N-3)\times 2(N-3)$ has pixel brightness values $I_{mn}$ and the corresponding reflectance map $R(p_{mn}, q_{mn})$ given by (3) where,

$$p_{mn} = p_{ij}(x,y) \qquad q_{mn} = q_{ij}(x,y) \tag{16}$$

and $(m,n)$ is a point inside the $(i,j)$ square patch as shown in Figure 4. The indices $(m,n)$ are given by,

$$\begin{array}{llll} m = 2i & (x = 0.25) & m = 2i+1 & (x = 0.75) \\ n = 2j & (y = 0.25) & n = 2j+1 & (y = 0.75) \end{array} \tag{17}$$

Inside the square patches formed by four control vertices, there are four image pixels. Their positions inside the square are given by $(x,y) = (0.25, 0.25)$, $(x,y) = (0.25, 0.75)$, $(x,y) =$

$(0.75, 0.25), (x, y) = (0.75, 0.75)$. Hence, the surface orientation of these four positions are,

$$\begin{aligned} p_{ij}(0.25, 0.25) &= V_{ij}^T \mathbf{c}_x^{00} & p_{ij}(0.25, 0.75) &= V_{ij}^T \mathbf{c}_x^{01} \\ p_{ij}(0.75, 0.25) &= V_{ij}^T \mathbf{c}_x^{10} & p_{ij}(0.75, 0.75) &= V_{ij}^T \mathbf{c}_x^{11} \end{aligned} \tag{18}$$

where $\mathbf{c}_x^{00} = \mathbf{c}_x(0.25, 0.25)$, $\mathbf{c}_x^{01} = \mathbf{c}_x(0.25, 0.75)$, $\mathbf{c}_x^{10} = \mathbf{c}_x(0.75, 0.25)$, $\mathbf{c}_x^{11} = \mathbf{c}_x(0.75, 0.75)$, as given in (12). Similar expressions for $\mathbf{c}_y$ can be given. See Appendix for the numerical values of these vectors.

Using the image irradiance equation (2), we can now relate the brightness values at each pixel $I_{mn}$ with the height of the control vertices $v_{ij}$ appearing through substitution for $R(p, q)$ using (3), (10) and (11) with $p_{mn}, q_{mn}$ for $m, n = 0, \ldots, 2(N-3)$.

The optimal set of control vertices are those that minimise the brightness error and the problem is posed as an optimisation task where, the following cost function is minimised:

$$E_0 = \sum_{m=0}^{2(N-3)} \sum_{n=0}^{2(N-3)} \left[ k_p p_{mn} + k_q q_{mn} + k_r - I_{mn} \sqrt{1 + p_{mn}^2 + q_{mn}^2} \right]^2 \tag{19}$$

However, this is an ill-posed problem and we need to add a regularisation term in order to make the problem well-posed [22].

The regularisation term added to the brightness error cost term is the bending energy evaluated over the entire image plane $\Omega$, given by [18],

$$P_0 = \int_\Omega \left\{ \left( \frac{\partial^2 z}{\partial x^2} \right)^2 + 2 \left( \frac{\partial^2 z}{\partial x \partial y} \right)^2 + \left( \frac{\partial^2 z}{\partial y^2} \right)^2 \right\} dx\, dy \tag{20}$$

Since the surface is formed by patches of uniform bicubic splines with $C^2$ continuity, (20) can be easily integrated over the image plane. For example, consider the first term in (20):

$$\int_\Omega \left( \frac{\partial^2 z}{\partial x^2} \right)^2 dx\, dy = \sum_{i=0}^{N} \sum_{j=0}^{N} \int_{square_{(ij)}} \left( \frac{\partial^2 z_{ij}}{\partial x^2} \right)^2 dx\, dy \tag{21}$$

where the integration is carried out within each square patch. Substituting (15) above, we get,

$$\begin{aligned} \int_\Omega \left( \frac{\partial^2 z}{\partial x^2} \right)^2 dx\, dy &= \sum_{i=0}^{N} \sum_{j=0}^{N} \int_{square_{(ij)}} \left\{ \sum_{r=0}^{3} \sum_{s=0}^{3} v_{i+r,j+s} \frac{\partial^2 b_r(x)}{\partial x^2} b_s(y) \right\}^2 dx\, dy \\ &= \sum_{i=0}^{N} \sum_{j=0}^{N} V_{ij}^T \mathbf{H}_{xx} V_{ij} \end{aligned} \tag{22}$$

where $\mathbf{H}_{xx}$ is a $16 \times 16$ matrix. Similarly,

$$\int_\Omega \left( \frac{\partial^2 z}{\partial y^2} \right)^2 dx\, dy = \sum_{i=0}^{N} \sum_{j=0}^{N} V_{ij}^T \mathbf{H}_{yy} V_{ij} \tag{23}$$

$$\int_{\Omega} 2\left(\frac{\partial^2 z}{\partial x \partial y}\right)^2 dx\, dy \;=\; \sum_{i=0}^{N}\sum_{j=0}^{N} V_{ij}^T \mathbf{H}_{xy} V_{ij} \tag{24}$$

giving the regularisation term as,

$$P_0 = \sum_{i=0}^{N}\sum_{j=0}^{N} V_{ij}^T \mathbf{H} V_{ij} \tag{25}$$

where $\mathbf{H} = \mathbf{H}_{xx} + \mathbf{H}_{xy} + \mathbf{H}_{yy}$, whose values are given in the Appendix.

The optimal set of control vertices $v_{ij}$ for $i, j = 1, \ldots, N$ are estimated by minimising the total cost function $E$ given by,

$$\begin{aligned} E &= E_0 + \lambda P_0 \qquad (26)\\ &= \sum_{m=0}^{2(N-3)}\sum_{n=0}^{2(N-3)} \left\{ k_p p_{mn} + k_q q_{mn} + k_r - I_{mn}\sqrt{1 + p_{mn}^2 + q_{mn}^2} \right\}^2 + \lambda \sum_{i=0}^{N}\sum_{j=0}^{N} V_{ij}^T \mathbf{H} V_{ij} \end{aligned}$$

where $\lambda > 0$ is the regularisation parameter which provides a trade-off between smoothing and satisfying irradiance equation.

The regulariser $\lambda$ is chosen to have some initial value which is reduced gradually to near zero values, as in [12], [13], [19], [24]. The minimisation of (27) is carried out using the *conjugate gradient descent* method. Since changing $\lambda$ will change the conjugate direction, $\lambda$ is changed after every ten conjugate iterations and the negative gradient is then used as the optimum search direction following the change.

Applying conjugate gradient method to the cost function $E$, the control vertices can be iterated using,

$$V_{t+1} = V_t + \gamma \Delta V_t \tag{27}$$

where $\gamma$ is the optimum step length and $\Delta V$ is the conjugate gradient search direction. The search direction at the $(t+1)$ iteration is a linear combination of the negative gradient and the last step search direction, given by,

$$\Delta V_{t+1} = \Delta V_t - \nabla E_V \tag{28}$$

where $\nabla E_V$ is the gradient of the cost function with respect to the control vertices,

$$\nabla E_V = \left[\frac{\partial E}{\partial v_{11}}, \ldots, \frac{\partial E}{\partial v_{NN}}\right]^T \tag{29}$$

A control vertex affects only its sixteen neighbour square patches. With four pixels in each square patch, there are 64 nonzero elements in the summation required in the term $E_0$ for

each control vertex. If we rewrite equation (19) as $E_0 = \sum\sum e_{mn}^2$, then, $\partial(e_{mn}^2)/\partial v_{ij}$ is nonzero only when,

$$\begin{array}{llll} i-3 \le m/2 \le i & (m \text{ odd}) & i-3 \le (m-1)/2 \le i & (m \text{ even}) \\ j-3 \le n/2 \le j & (n \text{ odd}) & j-3 \le (n-1)/2 \le j & (n \text{ even}) \end{array} \tag{30}$$

On the boundaries and corners, there are less nonzero terms in the summation for $\nabla E_V$.

The optimum step length $\gamma$ is found by solving a nonlinear equation. Since the surface orientation $p_{mn}, q_{mn}$ are linear functions of the control vertices $V$,

$$\begin{aligned} p_{mn}(V+\gamma\Delta V) &= p_m n(V) + \gamma p_{mn}(\Delta V) = p_{mn} + \gamma \\ \Delta p_{mn} q_{mn}(V+\gamma\Delta V) &= q_m n(V) + \gamma q_{mn}(\Delta V) = q_{mn} + \gamma \Delta q_{mn} \end{aligned} \tag{31}$$

Substituting these in equation (19) gives a function dependent on the single variable $\gamma$,

$$E_0(\gamma) = k_2\gamma^2 + k_1\gamma + k_0 - 2\sum_{m=0}^{2(N-3)}\sum_{n=0}^{2(N-3)}\left[\sqrt{c_0 + c_1\gamma + c_2\gamma^2}\,(b_0 I_{mn} + b_1 I_{mn}\gamma)\right] \tag{32}$$

where,

$$\begin{aligned} k_0 &= \sum_{m=0}^{2(N-3)}\sum_{n=0}^{2(N-3)}[b_0^2 + c_0 I_{mn}^2], \qquad c_0 = 1 + p_{mn}^2 + q_{mn}^2, \qquad b_0 = k_p p_{mn} + k_q q_{mn} + k_r \\ k_1 &= \sum_{m=0}^{2(N-3)}\sum_{n=0}^{2(N-3)}[2b_0 b_1 + c_1 I_{mn}^2], \quad c_1 = 2p_{mn}\Delta p_{mn}\; q_{mn}\Delta q_{mn}, \quad b_1 = k_p \Delta p_{mn} + k_q \Delta q_{mn} \\ k_2 &= \sum_{m=0}^{2(N-3)}\sum_{n=0}^{2(N-3)}[b_1^2 + c_2 I_{mn}^2], \qquad c_2 = (\Delta p_{mn})^2 (\Delta q_{mn})^2 \end{aligned} \tag{33}$$

The optimum $\gamma$ is found using Newton-Raphson method to solve the equation

$$\frac{dE(\gamma)}{d\gamma} = 0 \tag{34}$$

resulting in

$$\gamma^{k+1} = \gamma^k - E'/E'' \tag{35}$$

where $E'$ and $E''$ are first and second derivatives of $E(\gamma)$ with respect to $\gamma$ evaluated at $\gamma^k$.

The B-spline based algorithm is as follows:

1. **Initialisation**: Set all control vertices to zero (*ie.*, set all $p$, $q$ to 0). Then calculate gradient of the cost function w.r.t control vertices. Let the negative gradient be the conjugate direction.
2. **Iterations**:
   (a) Calculate the optimum step length in the conjugate direction.
   (b) Update control vertices using the step length and direction.
   (c) Calculate new conjugate direction.
   (d) If cost function is below a threshold then stop, else go to (a).

# 5 Experimental Results

In the first experiment, an example is chosen where the shape of the surface changes periodically in both $x$ and $y$ directions. The underlying shape is generated from the equation $z(x, y) = sin(0.2x) + cos(0.4y)$. The shaded image for this surface is synthesised with the light source at a tilt angle of $\sigma = 15°$ and a slant angle of $\tau = 10°$. Figure 5 shows the shaded image, original shape of the surface, the recovered shape and the cost function decrease against number of observations. The reconstructed and the original shapes are shown in the same scale.

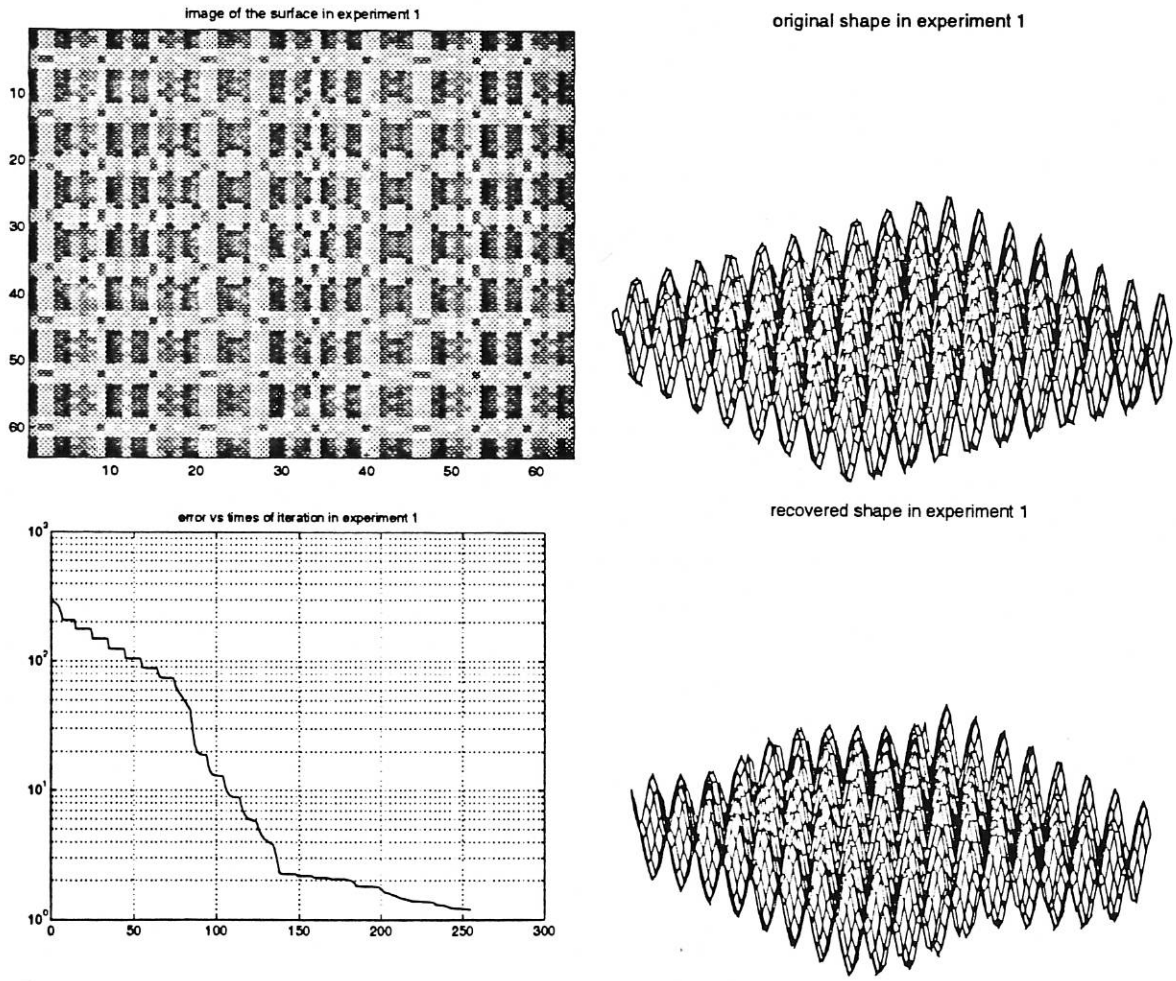


Figure 5: Experiment 1: (a) Shaded image (b) Original shape of image (c) Cost function Vs iteration (d) Recovered shape.

The reconstructed shape captures the essential variation, represented by the sinusoids. Comparing the shape near the edges indicate the presence of errors in the height of the shape, more than in the shape itself. The accuracy in the centre of the surface is sufficiently

good.

The second experiment involves a shape in which the surface changes only in one direction, the underlying shape being described by $z(x, y) = sin(0.4x)$. The shaded image is obtained, as before, with a tilt angle of $\sigma = 15°$ and a slant angle of $\tau = 10°$. Figure 6 shows the results for this experiment.

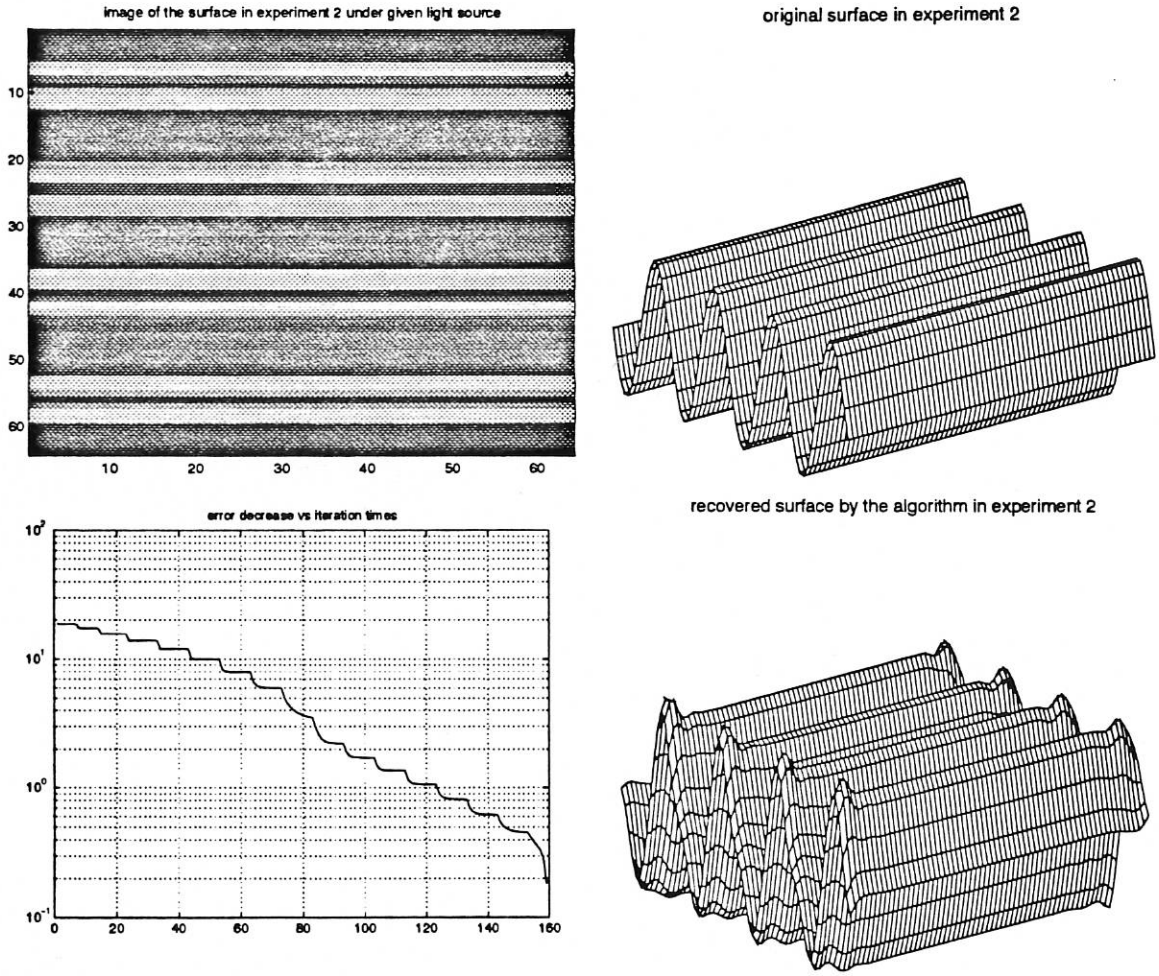


Figure 6: Experiment 2: (a) Shaded image (b) Original shape of image (c) Cost function Vs iteration (d) Recovered shape.

Once again, the results demonstrate that the essential features in the shape of the surface have been recovered. The shape near the edges exhibit aberrations and error in the height of the surface is also present. Much of the shape near the centre show good accuracy.

Finally, in the third experiment, a more complex shape give by $z(x, y) = sin(0.01x^2 + 0.01y^2)$, is used. Once again, the shaded image is generated with $\sigma = 15°$ and $\tau = 10°$. The results are shown in Figure 7.

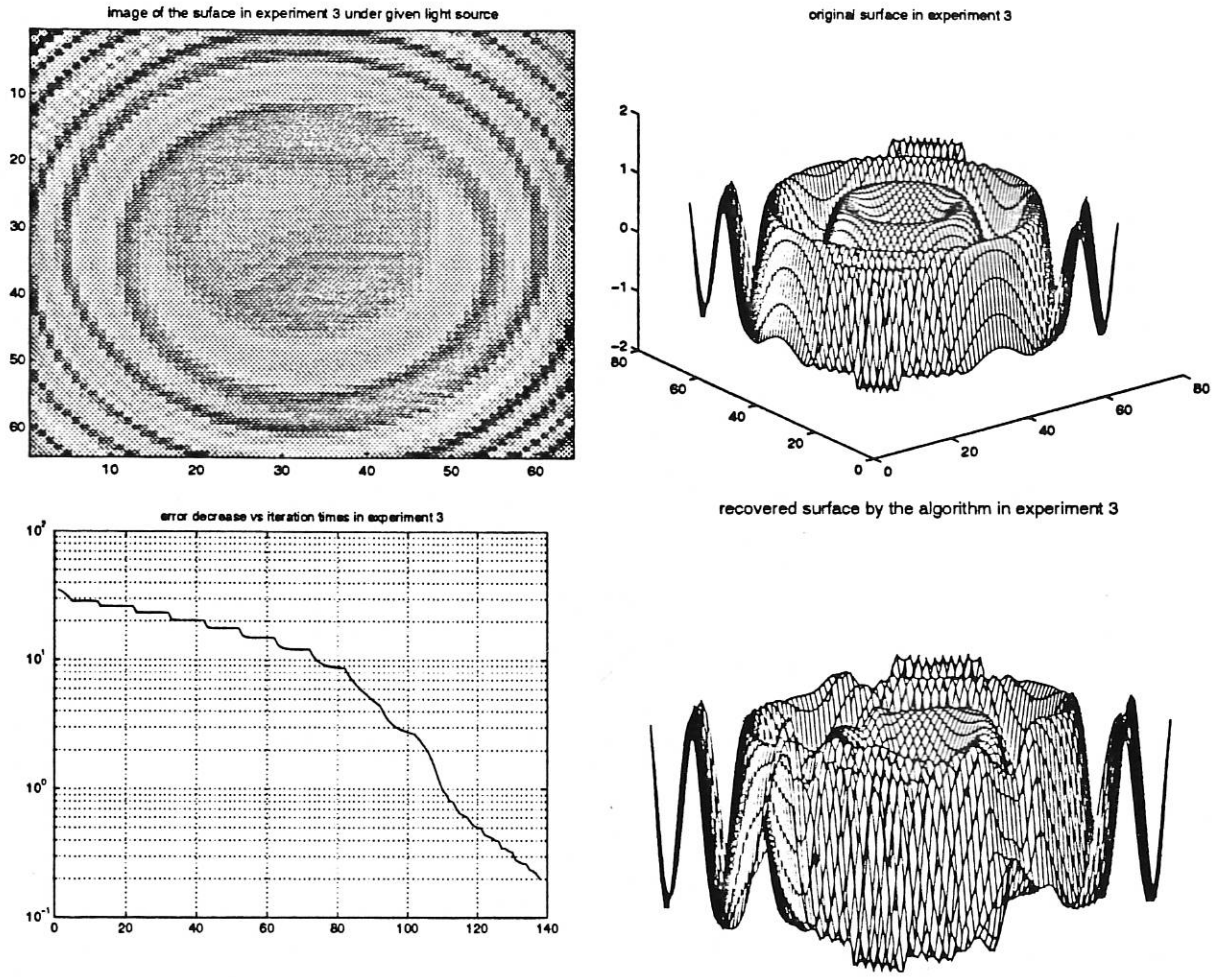


Figure 7: Experiment 3: (a) Shaded image (b) Original shape of image (c) Cost function Vs iteration (d) Recovered shape.

The results for the third experments, where a much wider bandwidth of the surface shape is used, show the errors appearing near the boundaries and in regions where the information in the original image is compressed. The essential features of the shape however, were recovered. Some of the results have been presented in [25].

The results in the three difficult problems demonstrate that the B-spline based algorithm achieved a good degree of approximation, given that no additional information other than the shaded image and the direction of light source was used. It should be borne in mind that the problem of recovering the shape is an ill-posed problem, more so without using additional constraints. The degradation of the shape near the boundaries is due to the small amount of information available to reliably estimate the control vertices. This accuracy can be significantly increased if additional constraints or information is used, such as surface orientation near the boundaries.

The application of the B-spline based algorithm is not without its limitations. The use of piece-wise continuous surface patches to represent the shape assumes that the object being recovered is smooth and devoid of sharp edges. Furthermore, the control vertices are placed on a regular grid and would therefore require the region of the image in which the object lies to be rectangular. Hence, the algorithm is not suitable for the application of object identification in a real scene. It is more suited to recovering the shape of terrains as required in tasks involving vehicle navigation and aerial photograph reconstruction.

## 6 Conclusions

We have a presented a new and efficient algorithm for the shape from shading problem. This algorithm differs from the traditional variational approach and adopts the surface interpolation idea by using B-splines to represent the shape of a surface. It is similar in spirit to the approach of Lee and Kuo [19] but does not require the linearisation approximation used in their work.

In our method, the surface is approximated by a linear combination of 2-D uniform bicubic spline basis functions with the control vertices placed on a regular grid in the image plane. The control vertices are estimated by minimising the total squared brightness error subject to the irradiance equation. Regularisation is used to make the problem well-posed in which the bending energy is used to constrain the shape to be smooth. The experimental results on three difficult shapes demonstrate the effectiveness of the algorithm to recover shape from a single shaded image.

## Appendix

The values for the coefficients in equations (12), (14) and (18) are:

$$\mathbf{C}_x^{00} = \begin{bmatrix} -1.98 \times 10^{-2} & -1.72 \times 10^{-1} & -8.86 \times 10^{-2} & -7.32 \times 10^{-4} \\ -2.86 \times 10^{-2} & -2.49 \times 10^{-1} & -1.28 \times 10^{-1} & -1.06 \times 10^{-3} \\ 4.61 \times 10^{-2} & 4.02 \times 10^{-1} & 2.07 \times 10^{-1} & 1.71 \times 10^{-3} \\ 2.19 \times 10^{-3} & 1.91 \times 10^{-2} & 9.85 \times 10^{-3} & 8.14 \times 10^{-5} \end{bmatrix} \tag{36}$$

$$\mathbf{C}_y^{00} = \mathbf{C}_x^{00^T} \tag{37}$$

$$\mathbf{C}_x^{01} = \begin{bmatrix} -7.32 \times 10^{-4} & -8.86 \times 10^{-2} & -1.72 \times 10^{-1} & -1.98 \times 10^{-2} \\ -1.06 \times 10^{-3} & -1.28 \times 10^{-1} & -2.49 \times 10^{-1} & -2.86 \times 10^{-2} \\ 1.71 \times 10^{-3} & 2.07 \times 10^{-1} & 4.02 \times 10^{-1} & 4.61 \times 10^{-2} \\ 8.14 \times 10^{-5} & 9.85 \times 10^{-3} & 1.91 \times 10^{-2} & 2.19 \times 10^{-3} \end{bmatrix} \tag{38}$$

$$\mathbf{C}_x^{10} = -\mathbf{C}_x^{01} \qquad \mathbf{C}_y^{10} = -\mathbf{C}_y^{01} \tag{39}$$

$$\mathbf{C}_x^{11} = -\mathbf{C}_x^{00} \qquad \mathbf{C}_y^{11} = -\mathbf{C}_y^{00} \tag{40}$$

The value of the symmetric $\mathbf{H}$ matrix in equation (25):

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} \tag{41}$$

where $\mathbf{H}_{11}$, $\mathbf{H}_{22}$ are symmetric and

$$\mathbf{H}_{21} = \mathbf{H}_{12}^T \qquad \mathbf{H}_{22} = \mathbf{H}_{11}^\circ \tag{42}$$

with $[.]^\circ$ representing a 180° rotation of an $n \times n$ matrix such that

$$\mathbf{H}_{22}(i,j) = \mathbf{H}_{11}(n+1-i, n+1-j) \tag{43}$$

The matrix $\mathbf{H}_{11}$ is as follows:

$$\begin{bmatrix} 3.82 \times 10^{-3} & 6.19 \times 10^{-3} & -3.02 \times 10^{-3} & -5.29 \times 10^{-5} & 6.19 \times 10^{-3} & -9.39 \times 10^{-3} & -8.81 \times 10^{-3} & 1.60 \times 10^{-3} \\ 6.19 \times 10^{-3} & 5.54 \times 10^{-2} & 1.78 \times 10^{-2} & -3.02 \times 10^{-3} & -9.39 \times 10^{-3} & -2.96 \times 10^{-2} & -6.68 \times 10^{-2} & -8.81 \times 10^{-3} \\ -3.02 \times 10^{-3} & 1.78 \times 10^{-2} & 5.54 \times 10^{-2} & 6.19 \times 10^{-3} & -8.81 \times 10^{-3} & -6.68 \times 10^{-2} & -2.96 \times 10^{-2} & -9.39 \times 10^{-3} \\ -5.29 \times 10^{-5} & -3.02 \times 10^{-3} & 6.19 \times 10^{-3} & 3.82 \times 10^{-3} & 1.60 \times 10^{-3} & -8.81 \times 10^{-3} & -9.39 \times 10^{-3} & 6.19 \times 10^{-3} \\ 6.19 \times 10^{-3} & -9.39 \times 10^{-3} & -8.81 \times 10^{-3} & 1.60 \times 10^{-3} & 5.54 \times 10^{-2} & -2.96 \times 10^{-2} & -2.24 \times 10^{-2} & 1.74 \times 10^{-2} \\ -9.39 \times 10^{-3} & -2.96 \times 10^{-2} & -6.68 \times 10^{-2} & -8.81 \times 10^{-3} & -2.96 \times 10^{-2} & 3.16 \times 10^{-1} & -3.48 \times 10^{-2} & -2.24 \times 10^{-2} \\ -8.81 \times 10^{-3} & -6.68 \times 10^{-2} & -2.96 \times 10^{-2} & -9.39 \times 10^{-3} & -2.24 \times 10^{-2} & -3.48 \times 10^{-2} & 3.16 \times 10^{-1} & -2.96 \times 10^{-2} \\ 1.60 \times 10^{-3} & -8.81 \times 10^{-3} & -9.39 \times 10^{-3} & 6.19 \times 10^{-3} & 1.74 \times 10^{-2} & -2.24 \times 10^{-2} & -2.96 \times 10^{-2} & 5.54 \times 10^{-2} \end{bmatrix} \tag{44}$$

and matrix $\mathbf{H}_{12}$ is as follows:

$$\begin{bmatrix} -3.02 \times 10^{-3} & -8.81 \times 10^{-3} & 1.00 \times 10^{-2} & 1.83 \times 10^{-3} & -5.29 \times 10^{-5} & 1.60 \times 10^{-3} & 1.87 \times 10^{-3} & 1.03 \times 10^{-4} \\ -8.81 \times 10^{-3} & -2.24 \times 10^{-2} & -2.12 \times 10^{-2} & 1.00 \times 10^{-2} & 1.60 \times 10^{-3} & 1.74 \times 10^{-2} & 1.74 \times 10^{-2} & 1.83 \times 10^{-3} \\ 1.00 \times 10^{-2} & -2.12 \times 10^{-2} & -2.24 \times 10^{-2} & -8.81 \times 10^{-3} & 1.83 \times 10^{-3} & 1.74 \times 10^{-2} & 1.74 \times 10^{-2} & 1.60 \times 10^{-3} \\ 1.83 \times 10^{-3} & 1.00 \times 10^{-2} & -8.81 \times 10^{-3} & -3.02 \times 10^{-3} & 1.03 \times 10^{-4} & 1.87 \times 10^{-3} & 1.60 \times 10^{-3} & -5.29 \times 10^{-5} \\ 1.78 \times 10^{-2} & -6.68 \times 10^{-2} & 2.12 \times 10^{-2} & 1.74 \times 10^{-2} & -3.02 \times 10^{-3} & -8.81 \times 10^{-3} & 1.00 \times 10^{-2} & 1.83 \times 10^{-3} \\ -6.68 \times 10^{-2} & -3.48 \times 10^{-2} & -3.42 \times 10^{-2} & 2.12 \times 10^{-2} & -8.81 \times 10^{-3} & -2.24 \times 10^{-2} & 2.12 \times 10^{-3} & 1.00 \times 10^{-2} \\ 2.12 \times 10^{-2} & -3.42 \times 10^{-2} & -3.48 \times 10^{-2} & -6.68 \times 10^{-2} & 1.00 \times 10^{-2} & 2.12 \times 10^{-3} & -2.24 \times 10^{-2} & -8.81 \times 10^{-3} \\ 1.74 \times 10^{-2} & 2.12 \times 10^{-2} & -6.68 \times 10^{-2} & 1.78 \times 10^{-2} & 1.83 \times 10^{-3} & 1.00 \times 10^{-2} & -8.81 \times 10^{-3} & -3.02 \times 10^{-3} \end{bmatrix} \tag{45}$$

## Acknowledgements

The first author acknowledges the support of the British Council for funding the research study programme at the Department of Automatic Control & Systems Engineering, University of Sheffield, UK, under the Sino – British Friendship Scholarship Scheme.

## References

[1] B. K. P. Horn. *Shape from shading: A method for obtaining the shape of a smooth opaque object from one view.* PhD Thesis, Department of Electrical Engineering, Massachusetts Institute of Technology, USA, 1970.

[2] B. K. P. Horn. Obtaining shape from shading information. In P. H. Winston (ed.), *Psychology of computer vision.* McGraw–Hill, New York, 1976.

[3] K. Ikeuchi & B. K. P. Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, Vol. 17, pp. 141-184, 1981.

[4] M. J. Brooks & B. K. P. Horn. Shape and source from shading. In *Proc. Int. Joint Conf. on Artificial Intelligence*, pp. 932-936, 1985.

[5] B. K. P. Horn & M. J. Brooks. The variational approach to shape from shading. *Computer Vision, Graphics and Image Processing*, Vol. 33, No. 2, pp. 174-208, 1986.

[6] B. K. P. Horn. *Robot vision.* MIT Press, MA: Cambridge, 1986.

[7] R. J. Woodham. A cooperative algorithm for determining surface orientation from single view. In *Proc. Int. Joint Conf. on Artificial Intelligence*, 1977.

[8] G. B. Smith. The relationship between image irradiance and surface orientation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1983.

[9] A. P. Pentland. Local shading analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No. 2, pp. 170-187, 1984.

[10] A. P. Pentland. Shape information from shading: a theory about human perception. In *Proc. Int. Conf. on Computer Vision*, pp. 404-413, 1988.

[11] R. T. Frankot & R. Chellappa. A method for forcing integrability in shape from shading algorithm. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-10, No. 4, pp. 439-451, 1988.

[12] B. K. P. Horn. Height and gradient from shading. *International Journal of Vision*, Vol. 5, No. 1, pp. 584-595, 1990.

[13] R. Szeliski. Fast shape from shading. *Computer Vision, Graphics and Image Processing*, Vol. 53, No. 2, 1991.

[14] Q. Zheng & R. Chellappa. Estimation of illumination direction, albedo and shape from shading. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-13, No. 7, 1991.

[15] B. K. P. Horn, R. Szeliski & A. L. Yuille. Impossible shading image. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-15, No. 2, 1993.

[16] O. E. Vega & Y. H. Yang. Shading logic: A heuristic approach to recover shape from shading. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-15, No. 6, 1993.

[17] D. Lee. A provably convergent algorithm for shape from shading. In B. Horn & M. Brooks (eds.). *Shape from shading*, MIT Press, MA: Cambridge, 1989.

[18] F. L. Bookstein. Principal warp: thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-11, No. 6, pp. 567-585, 1989.

[19] K. M. Lee & C. C. J. Kuo. Shape from shading with a linear triangular element surface model. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-15, No. 8, pp. 815-822, 1993.

[20] R. M. Bolle & D. B. Cooper. Bayesian recognition of local 3-D shape by approximating image intensity functions with quadratic polynomials. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-16, No. 4, 1984.

[21] R. J. Woodham. Photometric method for determining shape from shading. In S. Ullman and W. Richards (eds.), *Image understanding*, Ablex Publishing Corp., 1984.

[22] T. Poggio, V. Torre & C. Koch. Computational vision and regularization. *Nature*, Vol.317 26, pp. 314-319, 1985.

[23] R. H. Bartels, J. C. Beatty & B. A. Barsky. *An introduction to splines for use in computer graphics and geometric modelling.* Morgan Kaufmann Publishers Inc., California, 1987.

[24] Y. G. Leclerc & A. F. Bobick. The direct computation of height from shading. In *Proc. IEEE Conf. in Computer Vision and Pattern Recognition*, pp. 552-558, 1991.

[25] J. Gu & V. Kadirkamanathan. A B-Spline based algorithm for shape from shading. In *Proc. IAPR Workshop on Machine Vision Applications*, 1994.