Robust Shape from Shading

A. G. Jones and C. J. Taylor

Dept. of Medical Biophysics, Stopford Building, University of Manchester, Manchester, M13 9PT. Phone: 061 275–5130. Fax 061 275–5145

Abstract

Existing Shape from Shading algorithms are not robust enough to work reliably with real images. We present a new paradigm for Shape from Shading based upon a *scale space* surface representation. Using this representation, the Shape from Shading problem becomes one of forming a surface from gaussian basis functions in such a way that the shading in the original image is explained. The algorithm is both fast and robust, producing convincing results. In its current form, the algorithm requires the reflectance function to be approximately circularly symmetric. We show results for synthetic images, both with and without added noise, and for real Scanning Electron Microscope images.

1 Introduction

Images of three-dimensional objects often show variations in brightness, or shading, across object surfaces, and this information provides an important visual cue for recovering the shape of an object. Exploiting this information is one of the classic problems of computer vision, known as the *shape from shading* (SFS) problem [1].

To formulate the problem, we define a three-dimensional Euclidean co-ordinate system, where the z axis projects directly towards the viewpoint and the x and y axes coincide with the axes in the image plane. We assume orthographic projection with the light source at infinity. The visible surface is the function z(x,y), and the *surface gradient* is the pair (p,q) where $p = \partial z/\partial x$ and $q = \partial z/\partial y$, ie p and q are the rates of change of surface height in the x and y directions respectively. The observed brightness of a surface point is directly related to the surface gradient, the position of the light source and the reflectance properties of the surface. The relationship between the image and the surface gradient is expressed by the *image irradiance equation* E(x,y) = R(p,q), where E is the image, and R is the *reflectance function* which embodies both the effects of the imaging geometry and the reflectance properties of the surface. We assume that the reflectance function is of a known analytic form.

The SFS problem is to recover the surface function z(x,y) from the brightness image E(x,y) by "inverting" the image irradiance equation. Since the problem is underconstrained, the classical approach has been to introduce a *smoothness constraint* to *regularize* [1] the problem and thus allow a single solution to be found. The problem is then cast as a large non-linear optimisation task, and iterative techniques are used to progressively refine estimates of surface height or gradient. Such algorithms typically suffer a number of drawbacks:

- Convergence is slow and not robust, especially with noisy data.
- The recovered surface gradients may not be *integrable* (ie they cannot be integrated to form a physically realisable surface).
- Algorithm parameters may need careful tuning.

- · Boundary conditions are required.
- The smoothness constraint can make results overly smooth [1].

In this paper we present a novel *scale space* [3] SFS algorithm. The algorithm finds partial solutions at a range of resolutions, or *scales*, and combines these to form a complete solution. The approach has a number of advantages:

- Convergence is robust.
- No explicit smoothness constraint is required.
- The algorithm operates in the space of integrable surfaces, so the solution surface is always integrable.
- No boundary conditions are required.

We show results for synthetic images, both with and without additive noise, and for real Scanning Electron Microscope (SEM) images. The SEM imaging process is particularly appropriate for SFS when the usual assumptions are made, since the projection is orthographic and the "light source" is effectively at infinity.

2 Previous Work

Most Shape from Shading algorithms are based on Horn's formulation, using the *Calculus of Variations* [6], which involves minimizing an integral equation by solving the associated Euler equations using an iterative method. The integral equation is typically of the form:

$$e = \iint_{\Omega} \left(E(x,y) - R(p,q) \right)^2 + \lambda (p_x^2 + p_y^2 + q_x^2 + q_y^2) + \mu ((z_x - p)^2 + (z_y - q)^2) dx dy$$

where λ and μ are scalar multipliers and the subscripts represent partial differentiation. The first term is the brightness error between the image of the solution surface and the observed image; the second term is the smoothness (regularization) constraint; and the final term penalises departure from integrability. Choosing λ and μ is something of a black art, with poor choices preventing convergence or creating overly smooth results. Many algorithms first recover surface gradients which are then cast onto the nearest integrable surface [6][11]. Recently, Horn devised an algorithm that recovers surface height directly, avoiding the problem of ensuring integrability in the gradient field [1]. Szeliski [9] improved upon the slow convergence rates of Horn's relaxation schemes by minimising the integral equation directly using *conjugate gradient descent* [10].

Peleg and Ron [8] obtained faster convergence by using a multiresolution approach. Their approach requires the construction of a hierarchy of images of the blurred surface, but this cannot be constructed by simply blurring the image itself. They describe how an approximation to the image of the blurred surface can be obtained, and we use this technique in our algorithm.

Progress has also been made with local algorithms for SFS, most notably the recent work of Oliensis and Dupuis [2] which casts SFS as an optimal control problem. Their algorithm is fast, robust and gives accurate reconstructions, but is unable to cope with inflections and plateaux in the surface and requires prior knowledge about singular points (points of zero gradient).

3 Scale Space Shape from Shading

In this section, we first describe the scale space surface representation, and then describe our SFS algorithm.

3.1 The scale space surface representation

We define a scale space representation of the surface function z(x,y) as the three-dimensional scalar field $\Phi(x,y;\sigma)$, where σ is the scale parameter [3]. The function $\Phi(x,y;\sigma_k)$ ($\sigma_k \ge 0$) represents the surface at the scale σ_k , and we define a function $\psi(x,y;\sigma_k)$ to be the difference between two successive discrete samples from scale space, ie $\psi(x,y;\sigma_k) = \Phi(x,y;\sigma_k) - \Phi(x,y;\sigma_{k+1})$. We can now write the surface function at the scale σ_k as the sum:

$$\Phi(x,y;\sigma_k) = \sum_{i=k}^{N} \psi(x,y;\sigma_i) + DC$$

where σ_N is the largest scale in the scale space and *DC* is a constant which we assume to be zero, since it represents a constant height offset which cannot be recovered using SFS. In our algorithm, we generate the functions $\psi(x,y;\sigma_N)$, $\psi(x,y;\sigma_{N-1})$, ..., $\psi(x,y;\sigma_0)$ in turn, adding them into the sum until the entire scale space representation has been constructed. At each stage, $\Phi(x,y;\sigma_k)$ represents the current best estimate for the surface function.

3.2 Gaussian basis functions

In the continuous two-dimensional case, it has been shown by Koenderinck [3] that the most suitable basis function for forming a scale space is the gaussian. For a discrete signal, we write each ψ function as the sum of *M* gaussian basis functions, positioned on a regular grid:

$$\psi(x,y;\sigma_k) = \sum_{i=1}^{M} a_i e^{-((x-x_i)^2 + (y-y_i)^2)/2\sigma_k^2}$$

where (x_i, y_i) are the grid co-ordinates of the centre of the gaussian, and a_i is a scalar coefficient.

The SFS problem now becomes one of finding the coefficients a_i for each ψ function. As each ψ function is found it is added to the partial solution obtained so far, building up the description of the surface. Before showing how these coefficients are found, we first discuss how the scale space representation allows us to obtain smooth solutions.

3.3 Surface smoothness

There may be many different surfaces that can explain a given image, so to find a single solution we apply the heuristic that the smoothest explanation is the most plausible. We characterise the "lack of smoothness" of a surface by the integral:

$$S = \iint_{\Omega} p_x^2 + p_y^2 + q_x^2 + q_y^2 \, dxdy$$
$$= \frac{1}{2\pi} \iint_{-\infty}^{\infty} (u^2 + v^2)^2 \, |F[z(x,y)]|^2 \, dudv$$

where u and v are the frequency components in the x and y directions respectively and F is the Fourier transform. The term $(u^2 + v^2)^2$ increases rapidly with frequency, so smooth surfaces concentrate their energy towards the lower frequencies.

Each surface estimate $\Phi(x,y;\sigma_{k+1})$ may be thought of as a low-pass filtered version of $\Phi(x,y;\sigma_k)$, so we should achieve a smoother solution if we explain the image as well as

possible in $\Phi(x,y;\sigma_{k+1})$ before moving to the next scale and solving for $\Phi(x,y;\sigma_{k})$. We therefore ensure the solution is smooth by finding the ψ functions in the order $\psi(x,y;\sigma_{N}), \psi(x,y;\sigma_{N-1}), ..., \psi(x,y;\sigma_{0})$, such that the corresponding surface function always gives a smooth explanation of the image. How we ensure that a smooth surface estimate is found at each scale is described in the next section.

3.4 Finding surface approximations

In this section we develop the algorithm for finding an approximation for the solution surface at each scale σ_k . We first define two related vector spaces: \mathbb{Z} , based upon surface functions z(x,y); and \mathbb{PQ} , based upon surface gradients (p,q). We show how the solution to the approximation problem can be obtained in in \mathbb{PQ} space and converted back into a real surface in the space \mathbb{Z} .

Let $\mathbf{e}_{\mathbf{k}}$ be the set of M gaussian basis functions at the scale σ_k , given by $\mathbf{e}_{\mathbf{k}} = \{G_{ki}: 1 \le i \le M; i \in \mathbb{N}\}$ where $G_{ki} = e^{-((x-x_i)^2 + (y-y_i)^2)/2\sigma_k^2}$. These span a real vector space $\mathcal{Z}_{\mathbf{k}}$ in the nested vector spaces $\mathcal{Z}_{\mathbf{N}} \subseteq \mathcal{Z}_{\mathbf{N}-1} \subseteq ... \subseteq \mathcal{Z}_{\mathbf{k}} \subseteq ... \subseteq \mathcal{Z}_{0}$ (read \subseteq as "is a subspace of"). For each vector space $\mathcal{Z}_{\mathbf{k}}$ we define an isomorphic vector space $\mathcal{PQ}_{\mathbf{k}}$, where a point $\mathbf{g} = (p,q)$ in this space is the pair of gradient functions of a corresponding surface \mathbf{z} in $\mathcal{Z}_{\mathbf{k}}$. We define a linear mapping D from $\mathcal{Z}_{\mathbf{k}}$ to $\mathcal{PQ}_{\mathbf{k}}$, where $\mathbf{g} = (p,q) = D(\mathbf{z}) = (\partial z/\partial x, \partial z/\partial y)$. We obtain a set of basis functions $\mathbf{f}_{\mathbf{k}}$ for $\mathcal{PQ}_{\mathbf{k}}$ by applying the mapping D to the basis functions for $\mathcal{Z}_{\mathbf{k}}$, so:

$$\begin{aligned} \mathbf{f_k} &= D(\mathbf{e_k}) \\ &= \{(-(x-x_i)e^{-((x-x_i)^2 + (y-y_i)^2)/2\sigma_k^2}/\sigma_k^2 , -(y-y_i)e^{-((x-x_i)^2 + (y-y_i)^2)/2\sigma_k^2}/\sigma_k^2): 1 \le i \le M; i \in \mathbb{N} \} \\ &= \{(-(x-x_i)G_{ki}/\sigma_k^2 , -(y-y_i)G_{ki}/\sigma_k^2): 1 \le i \le M; i \in \mathbb{N} \} \end{aligned}$$

We find that there is a simple way of mapping points in \mathbb{Z}_k to their corresponding points in \mathbb{PQ}_k , and vice-versa. We write a typical surface z in \mathbb{Z}_k as the sum:

$$\mathbf{z} = a_1 \cdot G_{k1} + a_2 \cdot G_{k2} + \dots + a_M \cdot G_{kM}$$

The point **g** in \mathbb{PQ}_k corresponding to **z** is given by:

$$\mathbf{g} = (-b_1(x-x_1)G_{k1}/\sigma_k^2 + -b_2(x-x_2)G_{k2}/\sigma_k^2 + \dots + -b_M(x-x_M)G_{kM}/\sigma_k^2, -b_1(y-y_1)G_{k1}/\sigma_k^2 + -b_2(y-y_2)G_{k2}/\sigma_k^2 + \dots + -b_M(y-y_M)G_{kM}/\sigma_k^2)$$

Since *D* is linear (so $D(\lambda G_{ki}) = \lambda D(G_{ki})$), we observe that the coefficients for **g** are the same as those for **z**, ie $b_1 = a_1, b_2 = a_2, ..., b_M = a_M$. We can therefore map a point from one space to the other by a simple change of basis functions, keeping the coefficients unchanged.

We now show how an approximation for the surface function is found at the current scale σ_k . Let the unknown solution to the entire SFS problem be denoted by the point **u**, which lies somewhere in the space \mathbb{Z}_0 . At the scale σ_k we want to find the point in the subspace \mathbb{Z}_k closest to **u**, since this will be the best approximation for **u** (in a least squares sense) at that scale. We cannot find this point by working in \mathbb{Z}_k , since we have no information about surface heights, but if we work in \mathbb{PQ}_k we can use gradient information from the original image to find a suitable approximation.

We therefore reformulate the problem for solution in \mathfrak{PQ}_k . Let v be the point in \mathfrak{PQ}_0 corresponding to the required solution u in \mathfrak{Z}_0 , and let g be a typical point in the space \mathfrak{PQ}_k . Finding g such that ||g-v|| is minimised gives the best approximation for v in \mathfrak{PQ}_k ,

and hence the best approximation for the solution function **u** in \mathbb{Z}_k . Writing **g** as $(p_g,q_g)^{\dagger}$, and **v** as (p_v,q_v) , we define the distance between them as the Euclidean norm:

$$\| \mathbf{g} - \mathbf{v} \| = \sqrt{\sum_{x,y} ((p_g - p_v)^2 + (q_g - q_v)^2)}$$

The distance $\|\mathbf{g}-\mathbf{v}\|$ cannot be measured directly, since \mathbf{v} is unknown. However, we can measure the *brightness error*, Q, between the rendered image of \mathbf{g} and E_b , the estimated image of a surface formed by blurring \mathbf{v} . We find E_b by blurring the observed image using the algorithm of Ron and Peleg, described in Section 3.6. It is shown in Appendix B that by minimising Q we can find the surface approximation of *maximum likelihood*, given the estimated image of the blurred surface, E_b . We define the brightness error as:

$$Q(\mathbf{g}) = \sum_{x,y} (R(p_g, q_g) - E_b(x, y))^2$$

Since we know R and E_b , we can use an appropriate optimisation technique to find the set of coefficients for (p_g, q_g) that minimises Q. Our strategy for finding the next surface approximation $\Phi(x,y;\sigma_k)$, given that we have already found $\Phi(x,y;\sigma_{k+1})$, is therefore as follows:

- 1. Apply the mapping D to $\Phi(x,y;\sigma_{k+1})$, expressing the result in terms of the basis functions f_k . This is our initial estimate, s, for the best approximation to the solution surface in the space \mathfrak{PQ}_k .
- 2. From the initial estimate, s, find a function Δs in \mathfrak{PQ}_k such that $Q(s + \Delta s)$ is a minimum.
- 3. $s + \Delta s$ is expressed in terms of the coefficients $a_1, a_2, ..., a_M$ using the basis functions f_k . The corresponding surface in \mathbb{Z}_k is:

 $\Phi(x,y;\sigma_k) = a_1 \cdot G_{k1} + a_2 \cdot G_{k2} + a_3 \cdot G_{k3} + \dots + a_M \cdot G_{kM}$

The entire surface description is built up by finding each surface approximation $\Phi(x,y;\sigma_k)$ in turn (for k = N, N-1, ..., 1, 0), starting from the initial condition $\Phi(x,y;\sigma_N) = 0$ (ie the first estimate for the surface is a plane). The final solution surface is the function $\Phi(x,y;0)$.

In Step 2 of the algorithm there may be more than one Δs that minimises Q, so we choose the Δs that gives the smoothest surface. It is shown in Appendix B that the *lack-of-smoothness* measure S_k for the surface $\Phi(x,y;\sigma_k)$ is bounded above by $S_k \leq S_{k+1} + C \parallel \Delta s \parallel^2$, where S_{k+1} is the *lack-of-smoothness* for the previous surface $\Phi(x,y;\sigma_{k+1})$ and C is a positive constant. Thus the Δs of *smallest length* certainly gives a smooth surface approximation. Since we are sampling scale space densely, the initial estimate s should be close to a minimum in Q, so little more than gradient descent optimisation should be required to find the minimum in Q of smallest Δs . To be certain of finding the nearest minimum, the optimisation strategy could be restricted to explore the region of \mathcal{PQ}_k near the initial estimate, but in practice we have found such a restriction unnecessary.

3.5 Minimising the brightness error

Our algorithm minimises the brightness error function Q once at each discrete scale, so an efficient optimisation scheme is required. Unfortunately, the brightness error func-

[†] Here subscripts are for notational purposes and do not denote partial differentiation.

tion is difficult to minimise, since it forms a very large non-linear system. We have adopted a two stage approach, with a *Genetic Algorithm* (GA) [4] finding a rough estimate for the solution, followed by refinement by *conjugate gradient descent* [10]. The GA converges rapidly and robustly and is able to cope with plateaux and other anomalies in the objective function, but does not give an accurate solution. Conjugate gradient descent is less robust, but can provide the accuracy required in the final solution, given a reasonable starting approximation.

The Genetic Algorithm

The Genetic Algorithm (GA) is a search algorithm based upon natural genetics and the "survival of the fittest" principle of Darwinian evolution. The GA manipulates a *pool* of candidate solutions, each solution being coded as a bit string, or *chromosome*. Every iteration, the GA finds the fitter (ie better) chromosomes in the pool and "breeds" them together using *genetic operators* to form new chromosomes. These new chromosomes form a new pool, replacing the old one. Some of the new chromosomes are likely to be fitter than the ones from which they were derived, so there will be net movement towards a solution. See [4] for a more detailed description.

The GA owes its efficiency and robustness to a careful balance between *exploitation* (concentrating effort in promising areas of the search space) and *exploration* (making random trial moves in the search space). The GA is most suitable for finding good, but not optimal, solutions to large scale problems. We use the GA to find an initial estimate for the solution, and it performs well in this role. This estimate is then refined using *conjugate gradient descent*.

Conjugate gradient descent

Conjugate gradient descent is an optimisation technique suitable for solving large, nonlinear systems of equations. It uses derivatives to calculate search directions, with each new search direction being *conjugate* to previous search directions – see [10] for a detailed discussion of the algorithm.

In our application, derivatives are easily computed since the brightness error function can be differentiated analytically with respect to the scalar coefficients. The method performs well, converging rapidly to a good solution.

3.6 Finding the image of the blurred surface

We use the technique of Peleg and Ron [8] to obtain the approximate image of a blurred surface. This involves blurring the field $\sqrt{p^2 + q^2}$ derived from the image using the relationship $\sqrt{p^2 + q^2} = R^{-1}(E(x, y))$. The technique requires a circularly symmetric reflectance function, but we have achieved good results for other reflectance functions, provided that they do not deviate too greatly from the circularly symmetric model.

4 Experimental results

We show results for four images: two real SEM images, one of a cylindrical fibre and one of an electron microscope (EM) grid; and two synthetic images of a hemisphere, one noiseless and one with 10% additive gaussian noise. Each image is 64 by 64 pixels with 256 grey levels. We sampled scale space 32 times. Errors in the recovered height are shown for the synthetic image results, but not for the SEM images since no ground truth was available for these.

The SEM images were taken so that the reflectance function was as symmetrical as possible. The reflectance function was modelled using the relationship $R(p,q) = m/(\hat{\mathbf{n}} \cdot \hat{\mathbf{l}})$, where $\hat{\mathbf{n}}$ is the unit surface normal vector, $\hat{\mathbf{l}}$ is the unit vector in the illuminant direction, and the albedo *m* was found by analysing points in the image where $\hat{\mathbf{n}} \cdot \hat{\mathbf{l}} = 1$, since at these points m = R(p,q) = E(x,y). The results appear qualitatively correct for the particular samples used (Figure 1).



SEM image of cylindrical fibre



SEM image of part of an EM grid Recovered surface of EM grid Figure 1: Recovered surfaces for real SEM images

The synthetic images were shaded using the Lambertian model, $R(p,q) = \hat{\mathbf{n}} \cdot \hat{\mathbf{l}}$; the results are shown in Figure 2. Errors are shown in Table 1. The brightness error given is the mean of the absolute error between the original image and the calculated appearance of the solution surface, in grey level units. The height error is the mean of the absolute error between the synthetic data and the solution surface, using a measuring system where the radius of the hemisphere is 24 units.



Recovered surface of cylindrical fibre







Image of hemisphere (10% noise)

Recovered surface (10% noise)

Figure 2: Recovered surfaces for synthetic images of a hemisphere

Surface	Brightness error		Height error	
	Mean	sd	Mean	sd
Hemisphere (radius 24 units)	3.54	12.42	0.13	0.23
Hemisphere, 10% added noise	12.48	20.86	0.93	0.60
SEM image of fibre	0.38	0.94	NA	NA
SEM image of EM grid	0.32	0.98	NA	NA

nuole 1. Litois in ine result	Table	1:	Errors	in	the	results
-------------------------------	-------	----	--------	----	-----	---------

5 Conclusions

We have demonstrated a novel, robust technique for computing Shape from Shading that overcomes many of the drawbacks of existing methods. Our experimental results show that the algorithm can produce convincing results rapidly for both synthetic and real images.

The algorithm constructs a scale space [3] representation of the surface using gaussian basis functions. This choice of representation incorporates the smoothness assumption in a natural way, and no explicit smoothness constraint is required. Using this representation, the Shape from Shading problem becomes one of forming a surface from the basis functions in such a way that the original image is explained. This is done using two optimisation methods: the Genetic Algorithm [4] and conjugate gradient descent [10].

In future work, we intend to reformulate the problem using a dynamic system model [12]. The surface reconstruction will then be created by solving a coupled pair of differential equations using standard numerical techniques. This approach will mean that we no longer need to solve a new initial value problem at each scale, so it should be much more efficient than our current algorithm. We also intend to incorporate stereo information into the algorithm in an attempt to improve accuracy and remove some of the ambiguity inherent in monocular images.

Acknowledgements

The author holds a Total Technology studentship funded by SERC and Leica Cambridge Ltd.

Appendix A

The "lack of smoothness" integral for the surface estimate $\Phi(x,y;s_k)$ ($\Phi(x,y;s_k)$ is represented in \mathcal{PQ}_k by $s + \Delta s$) is given by:

$$S_{k} = \sum_{x,y} \left(\frac{\partial p}{\partial x}\right)^{2} + \left(\frac{\partial p}{\partial y}\right)^{2} + \left(\frac{\partial q}{\partial x}\right)^{2} + \left(\frac{\partial q}{\partial y}\right)^{2}$$

$$= \sum_{x,y} \left\|\frac{\partial \mathbf{s}}{\partial x} + \frac{\partial \Delta \mathbf{s}}{\partial x}\right\|^{2} + \left\|\frac{\partial \mathbf{s}}{\partial y} + \frac{\partial \Delta \mathbf{s}}{\partial y}\right\|^{2}$$

$$\leq \sum_{x,y} \left\|\frac{\partial \mathbf{s}}{\partial x}\right\|^{2} + \left\|\frac{\partial \mathbf{s}}{\partial y}\right\|^{2} + \left\|\frac{\partial \Delta \mathbf{s}}{\partial x}\right\|^{2} + \left\|\frac{\partial \Delta \mathbf{s}}{\partial y}\right\|^{2}$$
(triangle inequality)
$$= S_{k+1} + \sum_{x,y} \left\|\frac{\partial \Delta \mathbf{s}}{\partial x}\right\|^{2} + \left\|\frac{\partial \Delta \mathbf{s}}{\partial y}\right\|^{2}$$

$$= S_{k+1} + \sum_{x,y} \sum_{i=1}^{N} a_{i}^{2} \left(\frac{\partial G_{ki}}{\partial x}\right)^{2} + a_{i}^{2} \left(\frac{\partial G_{ki}}{\partial y}\right)^{2}$$

$$= S_{k+1} + \sum_{i=1}^{N} a_{i}^{2} \sum_{x,y} \left(\frac{\partial G_{ki}}{\partial x}\right)^{2} + \left(\frac{\partial G_{ki}}{\partial y}\right)^{2}$$

where $G_{ki} = e^{-((x-x_i)^2 + (y-y_i)^2)/2\sigma_k^2}$ and S_{k+1} is the lack of smoothness of the solution at the previous scale σ_{k+1} . The sum $\sum_{x,y} (\partial G_{ki}/\partial x)^2 + (\partial G_{ki}/\partial y)^2$ is a positive constant for all *i* so we denote this by *C* and hence write:

$$S_k \leq S_{k+1} + C \sum_{i=1}^{N} a_i^2 = S_{k+1} + C || \Delta s ||^2$$

This upper bound on the lack of smoothness means that the *nearest* local minimum in Q (ie the local minimum of smallest $\|\Delta s\|$) is likely to be the smoothest solution at the current scale.

Appendix B

Let E_b be the approximate image of the blurred solution surface, created by blurring the observed image using the algorithm of Ron and Peleg (see Section 3.6). For a typical point **g** in \mathcal{PQ}_k , we use Bayes's theorem to write the probability of **g** given E_b as:

$$P(\mathbf{g}|E_b) = \frac{P(E_b|\mathbf{g})P(\mathbf{g})}{P(E_b)}$$

If we assume gaussian noise in the measurement E_b , we can obtain:

$$P(\mathbf{g}|E_b) = \frac{e^{(-(E_b - R(p_g, q_g))^2/2s^2)}P(\mathbf{g})}{P(E_b)}$$

where R is the reflectance function, $(p_g, q_g) = \mathbf{g}$, and s is the standard deviation of the measurement noise. We make no prior assumptions so $P(\mathbf{g})$ and $P(E_b)$ are uniform, hence:

$$P(\mathbf{g}|E_b) = k e^{(-(E_b - R(p_g, q_g))^2/2s^2)}$$
 Eqn. 1

The maximum likelihood (ML) estimate for $P(\mathbf{g}|E_b)$ is given by:

$$\mathbf{g}_{ML} = argmax_{\mathbf{g}} P(\mathbf{g}|E_b)$$

Taking -loge of Equation 1, we can write:

$$\mathbf{g}_{ML} = argmin_{\mathbf{g}} \ (E_b - R(p_g, q_g))^2$$

$$= argmin_{\mathbf{g}} Q(\mathbf{g})$$

where $Q(\mathbf{g})$ is the brightness error of the surface \mathbf{g} defined in Section 3.4. Our results suggest that the quality of the approximation E_b is not critical for the success of the algorithm. The error in the approximation E_b decreases with scale, and our algorithm seems able to recover from any surface distortion this error causes at larger scales. In fact, reasonable results can even be achieved if no blurring is done and the original image is used unchanged.

References

- [1] Horn, BKP. "Height and gradient from shading". AI memo No 1105 May 1989 MIT AI lab.
- [2] Dupuis, P and Oliensis, J. "Direct method for reconstructing shape from shading". IEEE Computer Vision and Pattern Recognition Conf proc, pp453–458, 1992.
- [3] Koenderinck J.J. Solid Shape. The MIT Press, Cambridge, MA. 1990.
- [4] Goldberg, David E. Genetic Algorithms in Search, Optimisation and Machine Learning. Reading, MA. Addison–Wesley, 1988.
- [5] Forsyth D, Zisserman A. "Shape from Shading in the light of mutual illumination". Image and Vision Computing. Vol 8 num 1, Feb 1990.
- [6] Horn, B. K. P. "The Variational Approach to Shape from Shading". CVGIP, Vol. 33, No 2, pp174–208, February 1986.
- [7] Burt P.J. "The pyramid as a structure for efficient computation". from *Multiresolution Image Processing and Analysis* ed A. Rosenfeld, Springer-Verlag, 1984.
- [8] Peleg S, Ron G. "Nonlinear multiresolution: A Shape-from-Shading Example". IEEE PAMI vol 12 num 12, Dec 1990.
- [9] Szeliski, R. "Fast Shape from Shading". First European conference on Computer Vision proceedings, Springer-Verlag, 1990.
- [10] Gill PE, Murray W. Practical optimization. Academic Press, New York, 1981.
- [11] Chellapa R, Frankot RT. "A method for enforcing integrability in Shape from Shading algorithms". First Int. Conf. on Computer Vision proceedings, pp118–127, 1987.
- [12] Witkin, A, Terzopoulos, D and Kass, M. "Signal matching through scale space". Fifth Conf. AI proceedings pp714–719, 1986.