USC-SIPI REPORT #110

Direct Analytical Methods for Solving Poisson Equations in Computer Vision Problems

by

Tal Simchony and Rama Chellappa

July 1987

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Viterbi School of Engineering
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Suite 400
Los Angeles, CA 90089-2564 U.S.A.

Direct Analytical Methods for Solving Poisson Equations in Computer Vision Problems

T Simchony and R Chellappa[1]

Signal and Image Processing Institute
Dept of Electrical Engineering Systems
University of Southern California
Los Angeles , CA , 90089-0272 .

Abstract

The need to solve one or more Poisson equation of the general form:

$$\Delta u = f$$

arises in several computer vision problems such as enforcing integrability in shape from shading , the lightness and optical flow problems . The currently used methods for solving these Poisson equations are iterative . In this paper we first discuss direct analytical methods for solving these equations on a rectangular domain . We then suggest some embedding techniques that may be useful when boundary conditions (obtained from stereo , self shadowing and occluding boundary) are defined on arbitrary contours . The suggested algorithms are computationally efficient due to the use of fast orthogonal transforms. Application to lightness problems and optical flow are also discussed . The algorithm resulting from the direct analytical methods for the computation of optical flow is new . A proof for the existence and convergence of the flow estimates is also given .

1 Introduction

The need to solve one or more Poisson equation of the general form :

$$\Delta u = f$$

arises in several Computer Vision problems such as Shape from Shading [1] , the Lightness problem [2] and the Computation of Optical flow . The currently used methods for solving these equations are iterative . In this paper we discuss direct analytical methods [3] for solving the Poisson equations arising in the problems mentioned above . Specifically , we show how direct methods can be used to enforce integrability in Shape from Shading (SFS) problems when appropriate Dirichlet or Neuman boundary Conditions are given . Occluding boundary [4] and self shadowing information give us the orientation of the surface on the boundary , corresponding to Dirichlet boundary conditions for the equations solving for surface orientation , and Neuman boundary conditions for the equations solving for surface height . As stereo information usually gives us the surface height along a closed contour [4] , one can use this information and the intensity to obtain the orientation along this contour , constituting the Dirichlet boundary condition for the equations solving for both surface orientation and height . The method using Sine transform and periodic boundary conditions reduces to an algorithm similar to the recently developed method of Frankot and Chellappa [5] for enforcing integrability in the SFS problem .

Very often the information provided from stereo and occluding boundary are defined on arbitrary contours. To handle these cases we discuss some embedding techniques [6] . We further illustrate the applications of the method suggested here to the lightness problem [2] , to a direct method for estimating depth from shading [7] . Application of the direct analytical method to the optical flow problem , gives a new algorithm for computing the flow estimates. Using arguments similar to [7] , we prove the convergence of the iterative computations. As mentioned before , these algorithms are implemented using Sine or Cosine transforms and hence are computationally efficient.

The organization of this paper is as follows:
Section 2 discusses the importance of enforcing integrability in SFS in the context of ambiguity introduced by the existing algorithms . We suggest direct methods for enforcing integrability in SFS problem for the case of rectangular Dirichlet or Neuman boundary conditions . The case of non rectangular boundary contour is handled through embedding techniques [6] in Section 3 . Application to lightness problem , a direct solution to SFS problem and the new Optical flow algorithm with a proof for existence and convergence are in Sections 4 , 5 , 6 ,respectively.

2 Enforcing Integrability in SFS for Rectangular Boundary Conditions

2.1 Need for the Integrability Condition

2.1.1 Ambiguities in SFS Algorithms and the need for Integrability

An integrable surface is characterized by the property that the value of the integral between two points on the surface is independent of the contour of integration . Let $Z(x,y)$ be the unknown surface height , $Z_x = \frac{\partial Z(x,y)}{\partial x}$ $Z_y = \frac{\partial Z(x,y)}{\partial y}$, f,g be the surface orientation in stereoscopic coordinates and $n = (n_1, n_2, n_3)$, the surface normal. In deriving iterative solutions to the SFS problem by calculus of variations , it appears to be much more efficient to solve for the surface orientation (in one of the coordinate systems) and then to solve for the surface height . Then the question of integrability of Z and consistency between Z_x and Z_y arises . A reasonable consistency constraint to place on the surface slopes is that they satisfy

$$[Z_x]_y = [Z_y]_x \quad \forall x,y \in \Omega$$

The above consistency constraint corresponds to $Z(x,y)$ being

[1] Partially supported by the NSF Grant No. DCI-84-51010 and matching funds from IBM and AT&T .

TH0210-5/87/0000/0044$01.00 © 1987 IEEE

a C^2 function of x and y . Integrability is an inherent property of any C^2 surface . In general any solution obtained by calculus of variation is C^2 [8] but the problem is that the C^2 property will correspond to Z_x and Z_y f and g or n_1 n_2 but not to the surface Z. The importance of integrability can be illustrated using simple arguments . We first show that well posed problems may become ambiguous if integrability is not enforced . Let $E(x, y)$ be the observed image intensity related to a C^2 surface $Z(x, y)$ as

$$E(x, y) = R(Z_x, Z_y, \beta, l, \rho) \tag{1}$$

where β is the illumination direction vector , l is the vector from the surface to the camera , ρ is the albedo term , and Z_x and Z_y are the surface slopes defined above . In the case of a Lambertian surface , one can further write (1) as ,

$$E(x, y) = \frac{\rho\beta \cdot (-Z_x, -Z_y, 1)}{(1 + Z_x^2 + Z_y^2)^{\frac{1}{2}}} \tag{2}$$

It has been shown that the solution for the Shape from $E(x, y)$ may be found under different restrictions on E , β , l (eye at camera) . Using the characteristic strips approach , sufficient conditions for uniqueness of the SFS problem have been established for the case of an extremal contour and source at the eye [9] and for general illumination direction with given boundary conditions (Dirichlet) and at most a unique maximal point of the intensity array [4] .

Even if sufficient conditions are satisfied , due to the specific algorithms used one may get ambiguous solutions if integrability is not enforced . For instance ,consider the solutions based on finding the surface orientation in stereoscopic coordinates (f,g) and then constructing the surface . Poggio [10] suggested minimizing

$$\int\int (E - R(f, g))^2 + \lambda(f_x^2 + f_y^2 + g_x^2 + g_y^2) dxdy \tag{3}$$

Given homogeneous f on the boundary , and g arbitrary ,for the case when the light source is at the eye (orthogonal to the x,y plane) and the surface is Lambertian , the image irradiance equation is

$$R(f, g) = \frac{4 - f^2 - g^2}{4 + f^2 + g^2}$$

Without enforcing integrability the cost function is symmetric in f and therefore possible solutions are $(\pm f; g)$. If we now solve for Z using $-f$ and g and the given Dirichlet boundary conditions we end up with a wrong solution . The above boundary condition for f and g can occur when we rotate the x,y plane around the x axis . Thus the original well posed problem becomes ambiguous without forcing Z to be C^2 (integrability) . Similar arguments can be made for the minimization problem suggested in [11] as

$$\int\int [(E - n \cdot \beta)^2 + \lambda(n_x^2 + n_y^2) + \mu(n^2 - 1)] dxdy \tag{4}$$

where n is the surface normal and $\|n\| = 1 \; \forall (x, y) \in \Omega$.

When the surface normal components (n_1, n_2) are varied independently , given a source at eye , and zero boundary conditions on n_1 and n_2 the solutions (n_1, n_2, n_3) , $(-n_1, n_2, n_3)$, $(n_1, -n_2, n_3)$, $(-n_1, -n_2, n_3)$ will minimize (4) . Notice that $\pm Z(x, y)$ is an inherent ambiguity of the SFS problem with homogeneous boundary conditions .

All the above methods lack the important property of enforcing integrability . We can easily see that by enforcing integrability the ambiguity is resolved in the first example (3) . In the last example (4) two of the four ambiguous solutions drop and the remaining two correspond to the ambiguity $\pm Z$. This ambiguity can only be resolved by using some additional information such as the low frequency information [5] or $Z(x, y)$ at one point (where Z is not zero) .

One wonders if ambiguities in SFS can be resolved using smoothness criterion . To answer this , consider the 2-D example of

$$Z(x, y) = \begin{cases} 0 & \text{if } x^2 + y^2 > 1 \\ e^{-\frac{1}{1 - x^2 - y^2}} & \text{if } x^2 + y^2 \leq 1 \end{cases}$$

With the trivial boundary conditions for Z and its derivatives given on the contour $(-2, -2 \; ; \; -2, 2 \; ; \; 2, 2 \; ; \; 2, -2)$ and Lambertian reflectance map assuming source at eye , both $Z(x, y)$ and $-Z(x, y)$ have the same observed image intensity since :

$$R(Z_x, Z_y) = \frac{1}{(1 + Z_x^2 + Z_y^2)^{\frac{1}{2}}}$$

The ambiguity can not be resolved through smoothing constraints because both $Z(x, y)$ and $-Z(x, y)$ have infinitely many continuous derivatives , and will respond in the same way to smoothing constraints . The above ambiguity is related to the given SFS problem and not to the solving method . Therefore it can only be eliminated by additional information for example : sparse height measurements .

The examples discussed so far and the discussions in [1] , [5] should convince the reader that integrability is an important issue in SFS algorithms . Integrability was enforced in [12] by minimizing the following cost function:

$$\epsilon^2 \sum_{i=1}^{n} \sum_{j=1}^{m} (E_{ij} - R(p_{ij}, q_{ij}))^2 + \frac{\lambda}{\epsilon^2} \sum_{i=1}^{n} \sum_{j=1}^{m} e_{ij}^2$$

Where e_{ij} is the itegrability penalty term which corresponds to an estimate for the integral counter-clockwise around an elementary square path , with the picture cell (i,j) in the lower left corner , i.e.

$$e_{ij} = \frac{\epsilon}{2}[p_{i,j} + p_{i+1,j} + q_{i+1,j} + q_{i+1,j+1} - p_{i+1,j+1} - p_{i,j+1} - q_{i,j+1} - q_{i,j}]$$

The gradient of the penalty function was set to zero to get a set of nonlinear equations . The Jaccobi Picard relaxation algorithm was used . The algorithm converges relatively slowly . In this work integrability was not enforced strictly, but a penalty term was used . Horn and Brooks [1] were unable to derive via calculus of variation a converging scheme enforcing integrability strictly. Instead they imposed integrability using a penalty term with the following cost function :

$$\int\int_{\Omega} \underbrace{(E(x, y) - R(p, q))^2}_{\textit{irradiance equation}} + \underbrace{\lambda(p_y - q_x)^2}_{\textit{integrability term}} \; dxdy$$

to obtain a slowly converging scheme . They derived an iterative method with appropriate integrability penalty term for the surface represented as a function of the normal n . The method enabled the use of information from occluding boundary . Recently , [5] a method was suggested for enforcing integrability using projection onto convex sets . The approach is to project the possibly nonintegrable surface slopes estimates onto the nearest integrable surface slopes in the least square sense . They used an orthogonal projection for the case that the surface slopes are represented by finite sets of orthogonal

, integrable basis functions . The suggested method can only handle rectangular domain with periodic boundary conditions . The method can incorporate additional information such as low resolution height-data , and has faster rates of convergence . Horn and Brooks [1] looked into the problem of finding Z from p and q . They suggested minimizing

$$\int\int [(Z_x - p)^2 + (Z_y - q)^2] dx dy$$

with the corresponding Euler equation is

$$\Delta Z = p_x + q_y \qquad (5)$$

The solution for this problem in [1] is through iterative Jacobbi Picard Method . The rate of convergence for this method is in O (n^4) operations for an $n \times n$ 2-D problem . We suggest the use of direct Poisson methods. [3] The direct method solves the problem using transform techniques (usually Sine transform) in O (n^2 log n) operations . We use (5) to enforce integrability . In each iteration we solve for Z and then estimate new p and q which are integrable . Thus we are able to deal with general boundary conditions given on a general contour , and do it in the same order of time needed in [5] .

2.2 A New Algorithm for Enforcing Integrability.

2.2.1 Dirichlet Boundary Conditions

In this section we assume that we work on a rectangular domain.
Let us look at the Poisson equation for the Dirichlet boundary condition

$$\Delta Z = f$$

With $Z = g$ on the boundary . Suppose we discretize the Laplacian to get

$$\alpha^2(1 \ -2 \ 1)\ Z_{ij} + \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix} Z_{ij} = (\Delta y)^2 \bar{f}_{ij} \qquad (6)$$

where $\bar{f}_{ij}$ is the function f modified for boundary conditions and $\alpha^2 = (\Delta y / \Delta x)^2$. Let

$$Z_{ij} = \sum_{n=1}^{N_x - 1} a_{nj} \sin(\frac{ni\pi}{N_x}) \qquad (7)$$

By substituting (7) into (6) we obtain

$$\sum_{n=1}^{N_x-1} [(\alpha^2(2\cos\frac{n\pi}{N_x} - 2) - 2)a_{nj} + a_{nj+1} + a_{nj-1}] \cdot \sin\frac{in\pi}{N_x} = (\Delta y)^2 \bar{f}_{ij} \qquad (8)$$

Expending $\bar{f}_{ij}$ in terms of Sinusoidal basis functions

$$\bar{f}_{ij} = \sum_{n=1}^{N_x-1} F_{nj} \sin\frac{n\pi i}{N_x} \qquad (9)$$

and substituting (9) into(8) and equating coefficients of the sine terms

$$[\ 1\ [\alpha^2(2\cos(\frac{n\pi}{N_x}) - 2) - 2]\ 1\ ]\ a_{nj} = (\Delta y)^2 F_{nj} \qquad (10)$$

We are then left with a problem of solving for $j = 1, \cdots, N_y - 1$ a tri- diagonal matrix. It can be done efficiently using the LU decomposition technique [13] , in $O(N_y N_x)$ operations. For more details on the use of discrete Sine transform to diagonalize a symmetric tri-diagonal Toeplitz matrix the reader is referred to [14] .
In simple words , the most efficient algorithm takes a Sine transform of the input data along each column , and then solves the resulting tri-diagonal matrix problems corresponding to each row efficiently using the LU decomposition. Although it is less efficient we may use Sine transform in the other direction as well. The reason lies in the fact that the Sine transform can be implemented using FFT [14].
The DFT of p_x *and* q_x can be related to the DFT of p *and* q, if central differencing method is assumed to approximate the derivatives. The difference operator in the x direction appears as $j sin(\omega_x)$ [5]. We can use this fact when we have periodic boundary conditions to take only Fourier transform of p,q in order to calculate the Sine transform coefficient of the right hand side of (5) . We will reconsider this formulation when we discuss the relation between the current algorithm and the algorithm presented in [5] .
The algorithm can now be stated as:

1. get F_{nj} from $\bar{f}_{ij}$ using discrete Sine transformation
2. solve the system of tri-diagonal equations for the $a_{nj}s$
3. back transform $a \rightarrow Z$

As all operations can be done in place, storage is minimized. As we suggested before, we can transform the problem in both x and y directions to get the following problem:

$$[\alpha^2(2cos(\omega_x) - 2) + (2cos(\omega_y) - 2)]Z(\underline{\omega}) = \Delta y^2 F(\underline{\omega}) \quad (11)$$

This is a little less efficient but more straight forward.
The resulting algorithm is:

1. get $F(\underline{\omega})$ from $\bar{f}$ using discrete Sine transform in x and y direction
2. solve the system of diagonal equations

$$Z(\underline{\omega}) = \frac{\Delta y^2\ F(\underline{\omega})}{\alpha^2(2cos(\underline{\omega_x}) - 2) + (2cos(\omega_y) - 2)} = \frac{-\Delta y^2 F(\underline{\omega})}{4[\alpha^2 sin^2(\frac{\omega_x}{2}) + sin^2(\frac{\omega_y}{2})]} \quad (12)$$

3. get $Z(x, y)$ by inverse transforming $Z(\underline{\omega})$ (Sine transform)

2.2.2 Relation to Frankot , Chellappa [5] Algorithm

Lee [7] quotes the following results from [15] .
The eigen values of A , the discretized Laplacian operator are

$$\lambda ij = -4[sin^2\frac{\pi i}{2(N_x)} + sin^2\frac{\pi j}{2(N_y)}] \text{ where } N_x = \frac{1}{\Delta x}\ N_y = \frac{1}{\Delta y}$$

assume $N_x = N_y$ then
$A = H \Lambda H$, $H = S \otimes S$ a tensor product
S is a $N_x - 1 \times N_x - 1$ matrix .

We can easily relate our equation to the method in [5] using Fourier series expansion of the function Z . By Fourier transforming (5)

$$\mathcal{F}\{\Delta Z = p_x + q_y)\} \Rightarrow (-u^2 - v^2)Z(u,v) = jup(u,v) + jvq(u,v)$$

We have

$$Z(u,v) = \frac{-jup(u,v) - jvq(u,v)}{u^2 + v^2}$$

similar to (21) in [5] . We have established the correspondence between the two formulations in the continuous case.
We can take a further step and look at the DFT of the difference equation

$$DFT\left\{\left[(1\ -2\ 1) + \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}\right] Z_{(x,y)} = \right.$$

$$\left. \frac{\Delta y}{2}((1\ 0\ -1)p_{(x,y)} + \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} q_{(x,y)})\right\}$$

$$\Rightarrow (2cos(\omega_x) - 2 + 2cos(\omega_y) - 2)\ Z(\underline{\omega}) = jsin(\omega_x)p(\underline{\omega}) + jsin(\omega_y)q(\underline{\omega})$$

$$\Rightarrow [-4sin^2(\frac{\omega_x}{2})\ -\ 4sin^2(\frac{\omega_y}{2})]\ Z(\underline{\omega}) =$$

$$-[[\frac{sin(\omega_x)}{cos(\frac{\omega_x}{2})}]^2 + [\frac{sin(\omega_y)}{cos(\frac{\omega_y}{2})}]]Z(\underline{\omega}) =$$

$$jsin(\omega_x)p(\underline{\omega}) + jsin(\omega_y)q(\underline{\omega})$$

$$\Rightarrow\ Z(\underline{\omega}) = \frac{-jsin(\omega_x)p(\underline{\omega}) - jsin(\omega_y)q(\underline{\omega})}{\frac{sin^2(\omega x)}{cos^2(\frac{\omega_x}{2})} + \frac{sin^2(\omega y)}{cos^2(\frac{\omega_y}{2})}}$$

when $\omega_x \rightarrow o\ cos^2(\frac{\omega_x}{2}) \rightarrow 1$. For low frequencies our result is similar to the result obtained in [5] . For high frequencies we attenuate the corresponding coefficients since our discrete operator has a low-pass filter response. Simulation results using central differencing for the derivatives in the projection algorithm show that the surface Z obtained in [5] may suffer from high frequency oscillations. The high frequency attenuation in the new algorithm given here turn out to be in our favour. Furthermore , the new algorithm enables us to work with a general domain (described later) - and also lets us add information on Z on the boundary .

2.2.3 Neuman Boundary Conditions.

Often the only information available on the surface boundary is the orientation. This is the case with self shadow and occluding boundary [4]

Let us look now at the Neuman boundary condition.

$$\Delta Z = f\ in\ R$$

$$\frac{\partial Z}{\partial \eta} = g\ on\ \partial R$$

Where η is the direction orthogonal to the boundary. After proper discretization and first order approximation on the boundary we get the following matrix equation:

$$A = \begin{bmatrix} B & 2I & & \\ I & B & I & \\ & I & B & I \\ & & 2I & B \end{bmatrix} \quad B = \begin{bmatrix} -4 & 2 & & \\ 1 & -4 & 1 & \\ & 1 & -4 & 1 \\ & & 2 & -4 \end{bmatrix}$$

We first note that the matrix A is singular. We can determine the solution in the Neuman case only up to a constant. In computer vision problems , we will have to choose it or get the constant from physical properties . .
Next we can assume that :

$$Z_{ij} = \sum_{n=0}^{N} a_{nj} cos\frac{in\pi}{N} \quad (13)$$

with $\Delta x = \Delta y$
Note that this time we use a Cosine transform. Also note that the boundary needs to be determined as well.
Substituting (13) into (6) we get

$$\sum_{n=0}^{N}[(2cos\frac{n\pi}{N} - 2)a_{nj} + a_{nj+1} - 2a_{nj} + a_{nj-1}]cos\frac{in\pi}{N} = (\Delta y)^2\tilde{f}_{ij} \quad (14)$$

Expanding $\tilde{f}_{ij}$ in terms of Sinusoidal basis functions

$$\tilde{f}_{ij} = \sum_{n=0}^{N} F_{nj} cos\frac{n\pi i}{N} \quad (15)$$

and substituting (15) into (14) and equating coefficients of the cosine terms

$$(2cos\frac{n\pi}{N} - 2)a_{nj} + a_{nj+1} - 2a_{nj} + a_{nj-1} = (\Delta y)^2 F_{nj} \quad (16)$$

The tri-diagonal systems (one for each row) are non singular except for $n = 0$, which is treated separately.
Other ways to solve the problem include Block Cyclic Reduction [3] .

3 Integrability Using Embedding Techniques for Non-Rectangular Boundary Conditions

The boundary conditions for the SFS problems are usually given on a closed contour. This is the case when the boundary is obtained from self shadows, stereo, occluding boundary [4] , and usually includes orientation and occasionally the depth of the surface. It is rare that the contours form a rectangular domain . The direct approach presented in the previous section applies only to rectangular domains. We have to look for a modification of the problem to deal with irregular domain. The solution for the problem lies in embedding the irregular region in a rectangular domain and solving a modified problem . We will show how to handle irregular Dirichlet boundary conditions problem . Similar techniques are known [6] for the Neuman boundary conditions also .
The initial problem is:

$$\Delta u = f \text{ in } R \quad (17)$$

$$u = g \text{ on } \partial R \quad (18)$$

after discretization we have

$$\Delta_h u = f \text{ in } R_h$$

$$u = g \text{ on } \partial R_h$$

We embed the region R_h in a discrete rectangular region R'_h such that $R_h \subset R'_h$ and $\partial R_h \subset R'_h \bigcup \partial R'_h$ and let $S_h = \partial R_h \bigcap R'_h$. We extend the functions f and g to the regions R'_h and $\partial R_h \bigcup \partial R'_h$ respectively and solve for:

$$\left.\begin{array}{ll}\Delta_h u = f \text{ in } & R'_h - S_h \\ u = g \text{ on } & S_h \bigcup \partial R'_h\end{array}\right\} \quad (19)$$

The corresponding matrix equations are $Ax = f$
The solution to the above problem satisfies the original problem (17) . Let p be the number of grid points in S_h . We modify p rows of A and f corresponding to the equation $u = g$ on S_h and replace them by:

$$\Delta_h u = f \text{ on } S_h$$

We name the modified vector $\bar{f}$. The new matrix B thus defined is block tri-diagonal Toeplitz and the new problem thus defined can be solved using Sine transform . We will have to relate the solution to $By = \bar{f}$ to the problem $Ax = f$ in order to find the solution to the original problem. We follow the work in [6] .
Assume for simplicity that the first p rows of A need to be modified , we can achieve it by multiplying A by a permutation matrix. We should not do it explicitly in the computational procedure. It should be done implicitly by indexing, or else we destroy the special structure of B.
We now partition A in the form:

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$$

where A_1 is a $p \times n$ matrix and A_2 is an $(n-p) \times n$ matrix. We can write

$$B = \begin{pmatrix} B_1 \\ A_2 \end{pmatrix}$$

where B_1 is a $p \times n$ matrix. In the case of the Dirichlet problem B is a block tri-diagonal Toeplitz matrix.
We are given $Ax = f$ and write

$$f = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$$

Partition the vector $\bar{f}$ to

$$\bar{f} = \begin{pmatrix} \bar{f}_1 \\ f_2 \end{pmatrix}$$

for an arbitrary W $p \times p$ matrix we define $\bar{W}(n \times p)$ by

$$\bar{W} = \begin{pmatrix} W \\ O \end{pmatrix} \quad (20)$$

in our problem W can be chosen to be the identity matrix.
Define the $p \times p$ matrix C by:

$$C = A_1 B^{-1} \bar{W}$$

Assume there exists a solution β , $p \times 1$ *vector* for the equation

$$C\beta = f_1 - A_1 B^{-1} \bar{f}$$

Then it is easy to verify that the solution x to the equation $Ax = f$ is given by

$$x = B^{-1}(\bar{f} + \bar{W}\beta)$$

In [6] , the authors show that the method for finding x is valid whenever the original system $Ax = f$ is consistent.
Next, they use the matrix inversion formula to find the inverse of
$[B + FG] = A$ as

$$A^{-1} = (B + FG)^{-1} = B^{-1}(I - F(I + GB^{-1}F)^{-1}GB^{-1}),$$

where $F = \bar{W}$, and G is the $p \times n$ matrix is given by

$$G = W^{-1}(A_1 - B_1)$$

If A is nonsingular we find A^{-1}.
Now we can relate the solution of $By = \bar{f}$ to the solution of $Ax = f$ using the following steps:

1. Compute $C = A_1 B^{-1} \bar{W}$ since
$$A_1 = (I\ 0) \Rightarrow C = (I\ 0)B^{-1}\begin{pmatrix} I \\ 0 \end{pmatrix}$$
2. Compute $y = B^{-1}\bar{f}$
3. Solve the equation $C\beta = f_1 - A_1 y$

Additional computation is required to compute the matrix C and factor it . The applications we suggest in this paper need to solve the Poisson equations repeatedly , thus we perform the above computation once as part of a preprocessing stage. It is important to note that C is positive definite in this case. We can use Cholesky decomposition to compute LL^T decomposition of C [13] .

The solution x can be obtained from:

$$x = B^{-1}(\bar{f} + \bar{W}\beta) = B^{-1}(\bar{f} + \begin{pmatrix} I \\ 0 \end{pmatrix}\beta)$$

If we store y and $\bar{B} = B^{-1}\bar{W}$ then x can be computed from $x = y + \bar{B}\beta$. For our problems , the first method is more efficient.

4 Application to Lightness Problem [2]

The lightness of a surface is the perceptual correlate of its reflectance. The irradiance at a point is proportional to the product of the illuminance and reflectance at the corresponding point at the surface. The lightness problem is to compute lightness from image irradiance without any precise knowledge of either the reflectance or illuminance.
Following [2], if E = irradiance , S = illuminance , R(x,y) = reflectance then

$$E(x, y) = S(x, y) \times R(x, y)$$

Taking natural logarithms

$$e(x, y) = s(x, y) + r(x, y) \quad (21)$$

and applying the Lapalacian operator Δ on (21) gives

$$d(x, y) = \Delta e(x, y) = \Delta s(x, y) + \Delta r(x, y)$$

If a Mondrian illuminance is assumed to vary smoothly then $\Delta s(x, y)$ is finite everywhere.
$\Delta r(x, y)$ exhibits pulse doublets at intensity edges separating neighboring regions. A thresholding operator T can be applied to discard the illuminance component $T[d(x, y)] = \Delta r(x, y) = f(x, y)$. The reflectance map R is given by the inverse logarithm of the solution to Poisson's equation:

$$\Delta r(x, y) = f(x, y)$$

Given the proper boundary condition we can choose the corresponding direct method to solve the problem.

5 Application to a Direct Approach to SFS

Lee [7] suggested the use of a direct method for solving the nonlinear differential equation corresponding to the shape from shading problem :

$$\Delta p = \lambda h^2[R(p,q) - E]\ \frac{\partial R(p,q)}{\partial p}$$

$$\Delta q = \lambda h^2[R(p,q) - E]\ \frac{\partial R(p,q)}{\partial q}$$

with the corresponding matrix equations:

$$Mx = \lambda h^2 b(x)$$

$$M = \begin{pmatrix} A & O \\ O & A \end{pmatrix}$$

$$A = \begin{pmatrix} B & I & & & \\ I & B & I & & \\ & & & & \\ & & I & B & I \\ & & & I & B \end{pmatrix} \tag{22}$$

$$B = \begin{pmatrix} -4 & 1 & & & \\ 1 & -4 & 1 & & \\ & & & & \\ & & 1 & -4 & 1 \\ & & & 1 & -4 \end{pmatrix}$$

$$b = \{\ \ldots\ (R(p_{ij},q_{ij}) - E)\frac{\partial R(p_{ij},q_{ij})}{\partial p_{ij}}$$
$$\ldots\ (R(p_{ij},q_{ij}) - E)\frac{\partial R(p_{ij},q_{ij})}{\partial q_{ij}}\ldots\}^T$$

and

$$x = \{p_{11}\ldots p_{ik}\ldots p_{kk}\ldots q_{11}\ldots q_{ik}\ldots q_{kk}\}^T$$

Since M is nonsingular for the Dirichlet boundary condition , the suggested iterations are :

$$x^{m+1} = \lambda h^2 M^{-1} b(x^m))$$

where x^o is an arbitrary initial condition.

Lee's algorithm does not enforce integrability . Using our approach the above algorithm can be modified in the following manner.

1. calculate $p^{(m+1)}$ *and* $q^{(m+1)}$ using Lee's algorithm
2. project $p^{(m+1)}$ *and* $q^{(m+1)}$ solving the equation

$$\Delta Z^{n+1} = \frac{\partial p^{n+1}}{\partial x} + \frac{\partial q^{n+1}}{\partial y}$$

3. use Z^{n+1} to determine $\bar{p}^{n+1}$ and $\bar{q}^{n+1}$ using central differencing
4. return to stage 1

Furthermore, if the boundary conditions are given on an irregular contour we can use embedding techniques.
If we are given Dirichlet boundary condition for $Z(x,y)$ as in the case of stereo we can use the same matrix C for both steps, because we are embedding the same matrix A.
The efficiency of Lee algorithm is yet to be determined but it is the only method with a proof for convergence amongst all the suggested iterative methods.

6 Application to the Optical Flow Problem

Optical flow is the distribution of apparent velocities of irradiance patterns in the dynamic image. We use a temporal series of images to compute the velocity field [2] .
If we want to minimize

$$\begin{aligned} \varepsilon(u,v) &= \alpha^2 \int\int \underbrace{(u_x^2+u_y^2)+(v_x^2+v_y^2)}_{smoothing\ term}\ dxdy \\ &+ \int\int_\Omega \underbrace{(E_x u + E_y v + E_t)^2}_{the\ flow\ equation}\ dxdy \end{aligned} \tag{23}$$

The Euler Lagrange equations for the cost function ε are given by

$$E_x^2 u + E_x E_y v = \alpha^2 \Delta u - E_x E_t \tag{24}$$

$$E_x E_y u + E_y^2 v = \alpha^2 \Delta v - E_y E_t \tag{25}$$

Since we can estimate E_x , E_y , E_t from the given two images by

$$[E_x]_{ijk} = \frac{1}{2h}(E_{i+1jk} - E_{i-1jk})$$

$$[E_y]_{ijk} = \frac{1}{2h}(E_{ij+1k} - E_{ij-1k})$$

$$[E_t]_{ijk} = \frac{1}{h}(E_{ijk+1} - E_{ijk})$$

Equations (24) and (25) constitute a set of Poisson equations. In this discussion $h = \Delta x = \Delta y = \Delta t$ in Section 2 . One can rewrite (24) as

$$Mx = \frac{h^2}{\alpha^2} b(x)$$

where

$$M = \begin{bmatrix} A & \\ & A \end{bmatrix}$$

and A is as in (22) in the shape from shading problem .
Let

$$x = (u_{11}\ldots u_{ik}\ldots u_{kk} v_{11}\ldots v_{ik}\ldots v_{kk})^T$$

$$b(x) = \{\ldots\frac{\nu_{ijk}}{\mu_{ijk}}[E_x]_{ijk}\ldots\frac{\nu_{ijk}}{\mu_{ijk}}[E_y]_{ijk}\ldots\}^T \cdot \frac{\alpha^2}{h^2}$$

where

$$\mu_{ijk} = ([E_x]_{ijk})^2 + ([E_y]_{ijk})^2 + \frac{4}{h^2}\alpha^2$$

$$\nu_{ijk} = [E_x]_{ijk}\bar{u}_{ijk} + [E_y]_{ijk}\bar{u}_{ijk} + [E_t]_{ijk}$$

$$\bar{u}_{ijk} = \frac{1}{4}(u_{i-1,j,k} + u_{i,j+1,k} + u_{i+1,j,k} + u_{i,j-1,k})$$

and

$$\bar{v}_{ijk} = \frac{1}{4}(v_{i-1,j,k} + v_{i,j+1,k} + v_{i+1,j,k} + v_{i,j-1,k})$$

Again, we suggest the algorithm described in Section 5:

$$x^{n+1} = \frac{h^2 M^{-1}\ b(x^n)}{\alpha^2}$$

obtained by the direct method and appropriate boundary conditions.

We now discuss sufficient conditions for convergence of the above algorithm.

To see convergence it is easier to use the form

$$b_{u_{ij}}(u_{ij}v_{ij}) \triangleq [[E_x^2]_{ijk}\ u_{ij} + [E_x]_{ijk}[E_y]_{ijk}v_{ij} + [E_x]_{ijk}[E_t]_{ijk}]$$

$$b_{v_{ij}}(u_{ij}v_{ij}) \triangleq [[E_x]_{ijk}[E_y]_{ijk}u_{ij} + [E_y^2]_{ijk}\ v_{ijk} + [E_x]_{ijk}[E_t]_{ijk}]$$

It is obvious that $b_{u_{ij}}$ and $b_{v_{ij}}$ are Lipschitz functions [7] :

$$b_{u_{ij}}(u_{ij}, v_{ij}) - b_{u_{ij}}(u^1_{ij}, v^1_{ij}) \leq L^1\{(u-u^1)^2 + (v-v^1)^2\}^{\frac{1}{2}}$$

and

$$b_{v_{ij}}(u_{ij}, v_{ij}) - b_{v_{ij}}(u^1_{ij}\ v^1_{ij}) \leq L^2\{(u-u^1)^2 + (v-v^1)^2\}^{\frac{1}{2}}$$

Once we replace $\{[E_x]_{ijk}\}$ by $max_{ijk}\ |\{[E_x]_{ijk}\}|$ and $[E_x]_{ijk}$ by $max_{ijk}\ |\{[E_y]_{ijk}\}|$ and $\{[E_t]_{ijk}\}$ by $max_{ijk}\ |\{[E_t]_{ijk}\}|$,b becomes a linear function in u and v .

We now define $\nu_0 = max_{ijl}\{L^l_{ij}\}$ and $\lambda = \frac{1}{\alpha^2}$

We follow a theorem of Lee [7] to show that for $\lambda \in [0\ 2\pi^2\nu_0^{-1}[1 - \pi^2h^2/24]^2]$ the algorithm will converge .

We first assume that a solution exists, therefore it satisfies:

$$Mx = \lambda h^2 b(x)$$

Next, we note that

$$x^{m+1} - x = \lambda h^2 M^{-1}[b(x^m) - b(x)]$$

so

$$\|(x^{(m+1)}) - x\|_2 \leq \lambda h^2 \|M^{-1}\|_2 \nu_0 \|x^m - x\|_2$$

$$\|M^{-1}\|_2 = [8sin^2(\pi h/2)]^{-1} < [2\pi^2h^2(1 - \pi^2h^2 24)^2]^{-1}$$

$$\|x^{m+1} - x\|_2 < \lambda h^2[2\pi^2h^2(1 - \pi^2h^2/24)^2]^{-1}\nu_0\|x^m - x\|_2$$

Now, since $\lambda < 2\pi^2\nu_0^{-1}(1 - \pi^2h^2/24)^2$

$\Rightarrow \lambda[2\pi^2(1 - \pi^2h^2/24)^2]^{-1}\nu o < 1 \Rightarrow x^m$ converge to x.

Existence of a solution can be proved easily for the discrete version of the original penalty function. Consider the discrete function

$$\varepsilon_d(u, v) = \sum_i \sum_j (\alpha^2 s_{ij} + r_{ij})$$

$$s_{ij} = (u_{i+1j} - u_{ij})^2 + (u_{ij+1} - u_{ij})^2 + (v_{i+l,j} - v_{ij})^2 + (v_{ij+1} - v_{i,j})^2$$

$$r_{ij} = ([E_x]_{ij}u_{ij} + [E_y]_{ij}v_{ij} + [E_t]_{ij})^2$$

To minimize the above equation we have to solve a large system of sparse equations with $\lambda = \frac{1}{\alpha^2}$:

$$u_{ij} = \bar{u}_{ij} - \lambda h^2[[E_x]_{ij}u_{ij} + [E_y]_{ij}v_{ij} + [E_t]_{ij}[E_x]_{ij}]$$

$$v_{ij} = \bar{v}_{ij} - \lambda h^2[[E_x]_{ij}u_{ij} + [E_y]_{ij}v_{ij} + [E_t]_{ij}[E_y]_{ij}]$$

Note that we get exactly the same equation that we got using a discretized version of the Euler Lagrange equation.

Since we can restrict u and v to a rectangular closed domain from physical bounds on the maximum velocity, then $\varepsilon_d(u,v)$ becomes a continuous function defined on a compact subset S^{N^2} of R^{N^2} and therefore the minimum for the discrete problem exists. Thus for the discrete problem we have established the existence of a minimum, and stated sufficient conditions for the convergence of the suggested algorithm.

Acknowledgment

We wish to thank Professor Daniel Meiron of Caltech for introducing us to numerical methods of solving Elliptic Equations , and Vered Rom for helpful discussions.

References

[1] B. K. P. Horn and M. J. Brooks, "The Variational Approach to Shape from Shading", *Computer Vision , Graphics , and Image Processing*, vol.33, pp. 174–208, 1986.

[2] D. Terzopoulos, "Image Analysis Using Multigrid Relaxation Methods", *IEEE Transactions on Pattern Analysis and Machine Inteligence*, vol.8, pp. 129–139, March 1986.

[3] B. Buzbee, G. Golub, and C. Nielson, "On Direct Methods For Solving Poisson's Equations", *SIAM Journal of Numerical Analysis*, vol.7 no.4, pp. 627–656, 1970.

[4] A. Blake, "On the Geometric Information Obtainable from Simultaneous Observation of Stereo Contour and Shading", In *Report CSR 205-86*, Department of Computer Science , University of Edinbugh, 1986.

[5] R. T. Frankot and R. Chellappa, "A Method for Enforcing Integrability in Shape from Shading Algorithms", *IEEE Transactions on Pattern Analysis and Machine Inteligence*, Accepted for Publication, (Also presented at the First Int. Conf. on Computer Vision , London June 1987).

[6] B. Buzbee et al, "The Direct Solution of the Discrete Poisson Equation on Irregular Regeons", *SIAM Journal of Numerical Analysis*, vol.8 no.4, pp. 722–736, 1971.

[7] D. Lee, A Provably Convergent Algorithm for Shape from Shading, In *Proc. DARPA Image Undestanding Workshop*, pages 489–496, Miami Beach,Florida, December 1985.

[8] R. Courant and D. Hilbert, *"Methods of Mathematical Physics"*, Volume vol. 1, Interscience, New York, 1953.

[9] A.Bruss, "Is What You See What You Get", In *Proc. Int. Joint Conf. on Artificial Intelligence*, Karlsruhe,Germany, August 1983.

[10] T. Poggio and V. Torre, "Ill posed Problems and Regularization Analysis in Early Vision", *Artificial Intelligence Lab , M.I.T*, A.I.M 773, 1984.

[11] M. J. Brooks and B. K. P. Horn, Shape and Source From Shading, In *Proc. Int. Joint Conf. on Artificial Intelligence*, pages 932–936, Los Angeles,California, August 1985.

[12] M. Strat, *"A Numerical Method for Shape From Shading from a Single Image"*, M.S. thesis, M.I.T, Department of Elect. Engrg. and Comp. Sci., 1979.

[13] G. Strang, *"Linear Algebra and its Applications"*, Academic Press, New York, 1976.

[14] A. Rosenfeld and A. C. Kak, *"Digital Picture Processing"*, Volume vol. 1, Academic Press, New York, 1982.

[15] G. Smith, *"Numerical Solution of Parcial Differential Equations: Finite Difference Methods"*, University Press, Oxford, 1978.

Direct Analytical Methods for Solving Poisson Equations in Computer Vision Problems

T Simchony and R Chellappa[1]

Signal and Image Processing Institute
Dept of Electrical Engineering Systems
University of Southern California
Los Angeles , CA , 90089-0272 .

Abstract

The need to solve one or more Poisson equation of the general form:

$$\Delta u = f$$

arises in several computer vision problems such as enforcing integrability in shape from shading , the lightness and optical flow problems . The currently used methods for solving these Poisson equations are iterative . In this paper we first discuss direct analytical methods for solving these equations on a rectangular domain . We then suggest some embedding techniques that may be useful when boundary conditions (obtained from stereo , self shadowing and occluding boundary) are defined on arbitrary contours . The suggested algorithms are computationally efficient due to the use of fast orthogonal transforms. Application to lightness problems and optical flow are also discussed . The algorithm resulting from the direct analytical methods for the computation of optical flow is new . A proof for the existence and convergence of the flow estimates is also given .

1 Introduction

The need to solve one or more Poisson equation of the general form :

$$\Delta u = f$$

arises in several Computer Vision problems such as Shape from Shading [1] , the Lightness problem [2] and the Computation of Optical flow . The currently used methods for solving these equations are iterative . In this paper we discuss direct analytical methods [3] for solving the Poisson equations arising in the problems mentioned above . Specifically , we show how direct methods can be used to enforce integrability in Shape from Shading (SFS) problems when appropriate Dirichlet or Neuman boundary Conditions are given . Occluding boundary [4] and self shadowing information give us the orientation of the surface on the boundary , corresponding to Dirichlet boundary conditions for the equations solving for surface orientation , and Neuman boundary conditions for the equations solving for surface height . As stereo information usually gives us the surface height along a closed contour [4] , one can use this information and the intensity to obtain the orientation along this contour , constituting the Dirichlet boundary condition for the equations solving for both surface orientation and height . The method using Sine transform and periodic boundary conditions reduces to an algorithm similar to the recently developed method of Frankot and Chellappa [5] for enforcing integrability in the SFS problem .

Very often the information provided from stereo and occluding boundary are defined on arbitrary contours. To handle these cases we discuss some embedding techniques [6] . We further illustrate the applications of the method suggested here to the lightness problem [2] , to a direct method for estimating depth from shading [7] . Application of the direct analytical method to the optical flow problem , gives a new algorithm for computing the flow estimates. Using arguments similar to [7] , we prove the convergence of the iterative computations. As mentioned before , these algorithms are implemented using Sine or Cosine transforms and hence are computationally efficient.

The organization of this paper is as follows:
Section 2 discusses the importance of enforcing integrability in SFS in the context of ambiguity introduced by the existing algorithms . We suggest direct methods for enforcing integrability in SFS problem for the case of rectangular Dirichlet or Neuman boundary conditions . The case of non rectangular boundary contour is handled through embedding techniques [6] in Section 3 . Application to lightness problem , a direct solution to SFS problem and the new Optical flow algorithm with a proof for existence and convergence are in Sections 4 , 5 , 6 ,respectively.

2 Enforcing Integrability in SFS for Rectangular Boundary Conditions

2.1 Need for the Integrability Condition

2.1.1 Ambiguities in SFS Algorithms and the need for Integrability

An integrable surface is characterized by the property that the value of the integral between two points on the surface is independent of the contour of integration . Let $Z(x,y)$ be the unknown surface height , $Z_x = \frac{\partial Z(x,y)}{\partial x}$ $Z_y = \frac{\partial Z(x,y)}{\partial y}$, f, g be the surface orientation in stereoscopic coordinates and $n = (n_1, n_2, n_3)$, the surface normal. In deriving iterative solutions to the SFS problem by calculus of variations , it appears to be much more efficient to solve for the surface orientation (in one of the coordinate systems) and then to solve for the surface height . Then the question of integrability of Z and consistency between Z_x and Z_y arises . A reasonable consistency constraint to place on the surface slopes is that they satisfy

$$[Z_x]_y = [Z_y]_x \quad \forall x, y \in \Omega$$

The above consistency constraint corresponds to $Z(x,y)$ being

[1] Partially supported by the NSF Grant No. DCI-84-51010 and matching funds from IBM and AT&T .

a C^2 function of x and y . Integrability is an inherent property of any C^2 surface . In general any solution obtained by calculus of variation is C^2 [8] but the problem is that the C^2 property will correspond to Z_x *and* Z_y f and g or n_1 n_2 but not to the surface Z. The importance of integrability can be illustrated using simple arguments . We first show that well posed problems may become ambiguous if integrability is not enforced . Let $E(x,y)$ be the observed image intensity related to a C^2 surface $Z(x,y)$ as

$$E(x,y) = R(Z_x, Z_y, \beta, l, \rho) \qquad (1)$$

where β is the illumination direction vector , l is the vector from the surface to the camera , ρ is the albedo term , and Z_x and Z_y are the surface slopes defined above . In the case of a Lambertian surface , one can further write (1) as ,

$$E(x,y) = \frac{\rho\beta \cdot (-Z_x, -Z_y, 1)}{(1+Z_x^2+Z_y^2)^{\frac{1}{2}}} \qquad (2)$$

It has been shown that the solution for the Shape from $E(x,y)$ may be found under different restrictions on E , β , l (eye at camera) . Using the characteristic strips approach , sufficient conditions for uniqueness of the SFS problem have been established for the case of an extremal contour and source at the eye [9] and for general illumination direction with given boundary conditions (Dirichlet) and at most a unique maximal point of the intensity array [4] .

Even if sufficient conditions are satisfied , due to the specific algorithms used one may get ambiguous solutions if integrability is not enforced . For instance ,consider the solutions based on finding the surface orientation in stereoscopic coordinates (f,g) and then constructing the surface . Poggio [10] suggested minimizing

$$\int\int (E - R(f,g))^2 + \lambda(f_x^2 + f_y^2 + g_x^2 + g_y^2)dxdy \qquad (3)$$

Given homogeneous f on the boundary , and g arbitrary ,for the case when the light source is at the eye (orthogonal to the x,y plane) and the surface is Lambertian , the image irradiance equation is

$$R(f,g) = \frac{4 - f^2 - g^2}{4 + f^2 + g^2}$$

Without enforcing integrability the cost function is symmetric in f and therefore possible solutions are $(\pm f; g)$. If we now solve for Z using $-f$ and g and the given Dirichlet boundary conditions we end up with a wrong solution . The above boundary condition for f and g can occur when we rotate the x,y plane around the x axis . Thus the original well posed problem becomes ambiguous without forcing Z to be C^2 (integrability) . Similar arguments can be made for the minimization problem suggested in [11] as

$$\int\int [(E - n \cdot \beta)^2 + \lambda(n_x^2 + n_y^2) + \mu(n^2 - 1)]dxdy \qquad (4)$$

where n is the surface normal and $\|n\| = 1 \ \forall (x,y) \in \Omega$.

When the surface normal components (n_1, n_2) are varied independently , given a source at eye , and zero boundary conditions on n_1 and n_2 the solutions (n_1, n_2, n_3) , $(-n_1, n_2, n_3)$, $(n_1, -n_2, n_3)$, $(-n_1, -n_2, n_3)$ will minimize (4) . Notice that $\pm Z(x,y)$ is an inherent ambiguity of the SFS problem with homogeneous boundary conditions .

All the above methods lack the important property of enforcing integrability . We can easily see that by enforcing integrability the ambiguity is resolved in the first example (3) . In the last example (4) two of the four ambiguous solutions drop and the remaining two correspond to the ambiguity $\pm Z$. This ambiguity can only be resolved by using some additional information such as the low frequency information [5] or $Z(x,y)$ at one point (where Z is not zero) .

One wonders if ambiguities in SFS can be resolved using smoothness criterion . To answer this , consider the 2-D example of

$$Z(x,y) = \begin{cases} 0 & \text{if } x^2+y^2 > 1 \\ e^{-\frac{1}{1-x^2-y^2}} & \text{if } x^2+y^2 \le 1 \end{cases}$$

With the trivial boundary conditions for Z and its derivatives given on the contour $(-2,-2\ ;\ -2,2\ ;\ 2,2\ ;\ 2,-2)$ and Lambertian reflectance map assuming source at eye , both $Z(x,y)$ and $-Z(x,y)$ have the same observed image intensity since :

$$R(Z_x, Z_y) = \frac{1}{(1+Z_x^2+Z_y^2)^{\frac{1}{2}}}$$

The ambiguity can not be resolved through smoothing constraints because both $Z(x,y)$ and $-Z(x,y)$ have infinitely many continuous derivatives , and will respond in the same way to smoothing constraints . The above ambiguity is related to the given SFS problem and not to the solving method . Therefore it can only be eliminated by additional information for example : sparse height measurements .

The examples discussed so far and the discussions in [1] , [5] should convince the reader that integrability is an important issue in SFS algorithms . Integrability was enforced in [12] by minimizing the following cost function:

$$\epsilon^2 \sum_{i=1}^{n}\sum_{j=1}^{m}(E_{ij} - R(p_{ij}, q_{ij}))^2 + \frac{\lambda}{\epsilon^2}\sum_{i=1}^{n}\sum_{j=1}^{m} e_{ij}^2$$

Where e_{ij} is the itegrability penalty term which corresponds to an estimate for the integral counter-clockwise around an elementary square path , with the picture cell (i,j) in the lower left corner , i.e.

$$e_{ij} = \frac{\epsilon}{2}[p_{i,j} + p_{i+1,j} + q_{i+1,j} + q_{i+1,j+1} - p_{i+1,j+1} - p_{i,j+1} - q_{i,j+1} - q_{i,j}]$$

The gradient of the penalty function was set to zero to get a set of nonlinear equations . The Jaccobi Picard relaxation algorithm was used . The algorithm converges relatively slowly . In this work integrability was not enforced strictly, but a penalty term was used . Horn and Brooks [1] were unable to derive via calculus of variation a converging scheme enforcing integrability strictly. Instead they imposed integrability using a penalty term with the following cost function :

$$\int\int_{\Omega} \underbrace{(E(x,y) - R(p,q))^2}_{\textit{irradiance equation}} + \underbrace{\lambda(p_y - q_x)^2}_{\textit{integrability term}} \quad dxdy$$

to obtain a slowly converging scheme . They derived an iterative method with appropriate integrability penalty term for the surface represented as a function of the normal n . The method enabled the use of information from occluding boundary . Recently , [5] a method was suggested for enforcing integrability using projection onto convex sets . The approach is to project the possibly nonintegrable surface slopes estimates onto the nearest integrable surface slopes in the least square sense . They used an orthogonal projection for the case that the surface slopes are represented by finite sets of orthogonal

, integrable basis functions . The suggested method can only handle rectangular domain with periodic boundary conditions . The method can incorporate additional information such as low resolution height-data , and has faster rates of convergence . Horn and Brooks [1] looked into the problem of finding Z from p and q . They suggested minimizing

$$\int\int[(Z_x - p)^2 + (Z_y - q)^2]dxdy$$

with the corresponding Euler equation is

$$\Delta Z = p_x + q_y \tag{5}$$

The solution for this problem in [1] is through iterative Jacobbi Picard Method . The rate of convergence for this method is in O (n^4) operations for an $n \times n$ 2-D problem . We suggest the use of direct Poisson methods. [3] The direct method solves the problem using transform techniques (usually Sine transform) in O (n^2 log n) operations . We use (5) to enforce integrability . In each iteration we solve for Z and then estimate new p and q which are integrable . Thus we are able to deal with general boundary conditions given on a general contour , and do it in the same order of time needed in [5] .

2.2 A New Algorithm for Enforcing Integrability.

2.2.1 Dirichlet Boundary Conditions

In this section we assume that we work on a rectangular domain.
Let us look at the Poisson equation for the Dirichlet boundary condition

$$\Delta Z = f$$

With $Z = g$ on the boundary . Suppose we discretize the Laplacian to get

$$\alpha^2(1\ -2\ 1)\ Z_{ij} + \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix} Z_{ij} = (\Delta y)^2 \tilde{f}_{ij} \tag{6}$$

where $\tilde{f}_{ij}$ is the function f modified for boundary conditions and $\alpha^2 = (\Delta y/\Delta x)^2$. Let

$$Z_{ij} = \sum_{n=1}^{N_x-1} a_{nj} \sin(\frac{ni\pi}{N_x}) \tag{7}$$

By substituting (7) into (6) we obtain

$$\sum_{n=1}^{N_x-1} [(\alpha^2(2\cos\frac{n\pi}{N_x} - 2) - 2)a_{nj} + a_{nj+1} + a_{nj-1}] \cdot \sin\frac{in\pi}{N_x} = (\Delta y)^2\ \tilde{f}_{ij} \tag{8}$$

Expending $\tilde{f}_{ij}$ in terms of Sinusoidal basis functions

$$\tilde{f}_{ij} = \sum_{n=1}^{N_x-1} F_{nj} \sin\frac{n\pi i}{N_x} \tag{9}$$

and substituting (9) into(8) and equating coefficients of the sine terms

$$[\ 1\ [\alpha^2(2\cos(\frac{n\pi}{N_x}) - 2) - 2]\ 1\ ]\ a_{nj} = (\Delta y)^2 F_{nj} \tag{10}$$

We are then left with a problem of solving for $j = 1, \cdots, N_y - 1$ a tri- diagonal matrix. It can be done efficiently using the LU decomposition technique [13] , in $O(N_y N_x)$ operations. For more details on the use of discrete Sine transform to diagonalize a symmetric tri-diagonal Toeplitz matrix the reader is referred to [14] .
In simple words , the most efficient algorithm takes a Sine transform of the input data along each column , and then solves the resulting tri-diagonal matrix problems corresponding to each row efficiently using the LU decomposition. Although it is less efficient we may use Sine transform in the other direction as well. The reason lies in the fact that the Sine transform can be implemented using FFT [14].
The DFT of p_x *and* q_x can be related to the DFT of p *and* q, if central differencing method is assumed to approximate the derivatives. The difference operator in the x direction appears as $jsin(\omega_x)$ [5]. We can use this fact when we have periodic boundary conditions to take only Fourier transform of p,q in order to calculate the Sine transform coefficient of the right hand side of (5) . We will reconsider this formulation when we discuss the relation between the current algorithm and the algorithm presented in [5] .
The algorithm can now be stated as:

1. get F_{nj} from $\tilde{f}_{ij}$ using discrete Sine transformation
2. solve the system of tri-diagonal equations for the $a_{nj}s$
3. back transform $a \to Z$

As all operations can be done in place, storage is minimized. As we suggested before, we can transform the problem in both x and y directions to get the following problem:

$$[\alpha^2(2cos(\omega_x) - 2)\ +\ (2cos(\omega_y) - 2)]Z(\underline{\omega}) = \Delta y^2 F(\underline{\omega}) \tag{11}$$

This is a little less efficient but more straight forward.
The resulting algorithm is:

1. get $F(\underline{\omega})$ from $\tilde{f}$ using discrete Sine transform in x and y direction
2. solve the system of diagonal equations

$$\begin{aligned} Z(\underline{\omega}) &= \frac{\Delta y^2\ F(\underline{\omega})}{\alpha^2(2cos(\underline{\omega}_x) - 2) + (2cos(\omega_y) - 2)} \\ &= \frac{-\Delta y^2 F(\underline{\omega})}{4[\alpha^2 sin^2(\frac{\omega_x}{2})\ +\ sin^2(\frac{\omega_y}{2})]} \end{aligned} \tag{12}$$

3. get $Z(x, y)$ by inverse transforming $Z(\underline{\omega})$ (Sine transform)

2.2.2 Relation to Frankot , Chellappa [5] Algorithm

Lee [7] quotes the following results from [15] .
The eigen values of A , the discretized Laplacian operator are

$$\lambda ij = -4[sin^2\frac{\pi i}{2(N_x)} + sin^2\frac{\pi j}{2(N_y)}] \text{ where } N_x = \frac{1}{\Delta x}\ N_y = \frac{1}{\Delta y}$$

assume $N_x = N_y$ then
$A = H\Lambda H$, $H = S \otimes S$ a tensor product
S is a $N_x - 1 \times N_x - 1$ matrix .

We can easily relate our equation to the method in [5] using Fourier series expansion of the function Z . By Fourier transforming (5)

$$\mathcal{F}\{\Delta Z = p_x + q_y)\} \Rightarrow (-u^2 - v^2)Z(u,v) = jup(u,v) + jvq(u,v)$$

We have

$$Z(u,v) = \frac{-jup(u,v) - jvq(u,v)}{u^2+v^2}$$

similar to (21) in [5] . We have established the correspondence between the two formulations in the continuous case.
We can take a further step and look at the DFT of the difference equation

$$DFT\left\{\left[(1\ -2\ 1) + \begin{pmatrix}1\\-2\\1\end{pmatrix}\right] Z_{(x,y)} = \frac{\Delta y}{2}((1\ 0\ -1)p_{(x,y)} + \begin{pmatrix}1\\0\\-1\end{pmatrix} q_{(x,y)})\right\}$$

$$\Rightarrow (2cos(\omega_x)-2+2cos(\omega_y)-2)\ Z(\underline{\omega}) = jsin(\omega_x)p(\underline{\omega})+jsin(\omega_y)q(\underline{\omega})$$

$$\Rightarrow [-4sin^2(\frac{\omega_x}{2})\ -\ 4sin^2(\frac{\omega_y}{2})]\ Z(\underline{\omega}) =$$
$$-[[\frac{sin(\omega_x)}{cos(\frac{\omega_x}{2})}]^2 + [\frac{sin(\omega_y)}{cos(\frac{\omega_y}{2})}]]Z(\underline{\omega}) =$$
$$jsin(\omega_x)p(\underline{\omega}) + jsin(\omega_y)q(\underline{\omega})$$

$$\Rightarrow\ Z(\underline{\omega}) = \frac{-jsin(\omega_x)p(\underline{\omega}) - jsin(\omega_y)q(\underline{\omega})}{\frac{sin^2(\omega x)}{cos^2(\frac{\omega_x}{2})} + \frac{sin^2(\omega y)}{cos^2(\frac{\omega_y}{2})}}$$

when $\omega_x \to o\ cos^2(\frac{\omega_x}{2}) \to 1$. For low frequencies our result is similar to the result obtained in [5] . For high frequencies we attenuate the corresponding coefficients since our discrete operator has a low-pass filter response. Simulation results using central differencing for the derivatives in the projection algorithm show that the surface Z obtained in [5] may suffer from high frequency oscillations. The high frequency attenuation in the new algorithm given here turn out to be in our favour. Furthermore , the new algorithm enables us to work with a general domain (described later) - and also lets us add information on Z on the boundary .

2.2.3 Neuman Boundary Conditions.

Often the only information available on the surface boundary is the orientation. This is the case with self shadow and occluding boundary [4]

Let us look now at the Neuman boundary condition.

$$\Delta Z = f\ in\ R$$

$$\frac{\partial Z}{\partial \eta} = g\ on\ \partial R$$

Where η is the direction orthogonal to the boundary. After proper discretization and first order approximation on the boundary we get the following matrix equation:

$$A = \begin{bmatrix} B & 2I & & \\ I & B & I & \\ & I & B & I \\ & & 2I & B \end{bmatrix} B = \begin{bmatrix} -4 & 2 & & \\ 1 & -4 & 1 & \\ & 1 & -4 & 1 \\ & & 2 & -4 \end{bmatrix}$$

We first note that the matrix A is singular. We can determine the solution in the Neuman case only up to a constant. In computer vision problems , we will have to choose it or get the constant from physical properties . .
Next we can assume that :

$$Z_{ij} = \sum_{n=0}^{N} a_{nj} cos\frac{in\pi}{N} \quad (13)$$

with $\Delta x = \Delta y$
Note that this time we use a Cosine transform. Also note that the boundary needs to be determined as well.
Substituting (13) into (6) we get

$$\sum_{n=0}^{N}[(2cos\frac{n\pi}{N} - 2)a_{nj} + a_{nj+1} - 2a_{nj} + a_{nj-1}]cos\frac{in\pi}{N} = (\Delta y)^2 \bar{f}_{ij} \quad (14)$$

Expanding $\bar{f}_{ij}$ in terms of Sinusoidal basis functions

$$\bar{f}_{ij} = \sum_{n=0}^{N} F_{nj} cos\frac{n\pi i}{N} \quad (15)$$

and substituting (15) into (14) and equating coefficients of the cosine terms

$$(2cos\frac{n\pi}{N} - 2)a_{nj} + a_{nj+1} - 2a_{nj} + a_{nj-1} = (\Delta y)^2 F_{nj} \quad (16)$$

The tri-diagonal systems (one for each row) are non singular except for $n = 0$, which is treated separately.
Other ways to solve the problem include Block Cyclic Reduction [3] .

3 Integrability Using Embedding Techniques for Non-Rectangular Boundary Conditions

The boundary conditions for the SFS problems are usually given on a closed contour. This is the case when the boundary is obtained from self shadows, stereo, occluding boundary [4] , and usually includes orientation and occasionally the depth of the surface. It is rare that the contours form a rectangular domain . The direct approach presented in the previous section applies only to rectangular domains. We have to look for a modification of the problem to deal with irregular domain. The solution for the problem lies in embedding the irregular region in a rectangular domain and solving a modified problem . We will show how to handle irregular Dirichlet boundary conditions problem . Similar techniques are known [6] for the Neuman boundary conditions also .
The initial problem is:

$$\Delta u = f \text{ in } R \quad (17)$$

$$u = g \text{ on } \partial R \quad (18)$$

after discretization we have

$$\Delta_h u = f \text{ in } R_h$$

$$u = g \text{ on } \partial R_h$$

We embed the region R_h in a discrete rectangular region R_h' such that $R_h \subset R_h'$ and $\partial R_h \subset R_h' \bigcup \partial R_h'$ and let $S_h = \partial R_h \bigcap R_h'$. We extend the functions f and g to the regions R_h' and $\partial R_h \bigcup \partial R_h'$ respectively and solve for:

$$\left.\begin{array}{ll}\Delta_h u = f \text{ in } & R_h' - S_h \\ u = g \text{ on } & S_h \bigcup \partial R_h'\end{array}\right\} \quad (19)$$

The corresponding matrix equations are $Ax = f$
The solution to the above problem satisfies the original problem (17) . Let p be the number of grid points in S_h . We modify p rows of A and f corresponding to the equation $u = g$ on S_h and replace them by:

$$\Delta_h u = f \text{ on } S_h$$

We name the modified vector $\bar{f}$. The new matrix B thus defined is block tri-diagonal Toeplitz and the new problem thus defined can be solved using Sine transform . We will have to relate the solution to $By = \bar{f}$ to the problem $Ax = f$ in order to find the solution to the original problem. We follow the work in [6] .
Assume for simplicity that the first p rows of A need to be modified , we can achieve it by multiplying A by a permutation matrix. We should not do it explicitly in the computational procedure. It should be done implicitly by indexing, or else we destroy the special structure of B.
We now partition A in the form:

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$$

where A_1 is a $p \times n$ matrix and A_2 is an $(n-p) \times n$ matrix. We can write

$$B = \begin{pmatrix} B_1 \\ A_2 \end{pmatrix}$$

where B_1 is a $p \times n$ matrix. In the case of the Dirichlet problem B is a block tri-diagonal Toeplitz matrix.
We are given $Ax = f$ and write

$$f = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$$

Partition the vector $\bar{f}$ to

$$\bar{f} = \begin{pmatrix} \bar{f}_1 \\ f_2 \end{pmatrix}$$

for an arbitrary W $p \times p$ matrix we define $\bar{W}(n \times p)$ by

$$\bar{W} = \begin{pmatrix} W \\ O \end{pmatrix} \quad (20)$$

in our problem W can be chosen to be the identity matrix.
Define the $p \times p$ matrix C by:

$$C = A_1 B^{-1} \bar{W}$$

Assume there exists a solution β , $p \times 1$ *vector* for the equation

$$C\beta = f_1 - A_1 B^{-1} \bar{f}$$

Then it is easy to verify that the solution x to the equation $Ax = f$ is given by

$$x = B^{-1}(\bar{f} + \bar{W}\beta)$$

In [6] , the authors show that the method for finding x is valid whenever the original system $Ax = f$ is consistent.
Next, they use the matrix inversion formula to find the inverse of
$[B + FG] = A$ as

$$A^{-1} = (B+FG)^{-1} = B^{-1}(I - F(I + GB^{-1}F)^{-1}GB^{-1}),$$

where $F = \bar{W}$, and G is the $p \times n$ matrix is given by

$$G = W^{-1}(A_1 - B_1)$$

If A is nonsingular we find A^{-1}.
Now we can relate the solution of $By = \bar{f}$ to the solution of $Ax = f$ using the following steps:

1. Compute $C = A_1 B^{-1} \bar{W}$ since
$$A_1 = (I \ 0) \Rightarrow C = (I \ 0) B^{-1} \begin{pmatrix} I \\ 0 \end{pmatrix}$$
2. Compute $y = B^{-1} \bar{f}$
3. Solve the equation $C\beta = f_1 - A_1 y$

Additional computation is required to compute the matrix C and factor it . The applications we suggest in this paper need to solve the Poisson equations repeatedly , thus we perform the above computation once as part of a preprocessing stage. It is important to note that C is positive definite in this case. We can use Cholesky decomposition to compute LL^T decomposition of C [13] .

The solution x can be obtained from:

$$x = B^{-1}(\bar{f} + \bar{W}\beta) = B^{-1}(\bar{f} + \begin{pmatrix} I \\ 0 \end{pmatrix}\beta)$$

If we store y and $\bar{B} = B^{-1}\bar{W}$ then x can be computed from $x = y + \bar{B}\beta$. For our problems , the first method is more efficient.

4 Application to Lightness Problem [2]

The lightness of a surface is the perceptual correlate of its reflectance. The irradiance at a point is proportional to the product of the illuminance and reflectance at the corresponding point at the surface. The lightness problem is to compute lightness from image irradiance without any precise knowledge of either the reflectance or illuminance.
Following [2], if E = irradiance , S = illuminance , R(x,y) = reflectance then

$$E(x,y) = S(x,y) \times R(x,y)$$

Taking natural logarithms

$$e(x,y) = s(x,y) + r(x,y) \quad (21)$$

and applying the Lapalacian operator Δ on (21) gives

$$d(x,y) = \Delta e(x,y) = \Delta s(x,y) + \Delta r(x,y)$$

If a Mondrian illuminance is assumed to vary smoothly then $\Delta s(x,y)$ is finite everywhere.
$\Delta r(x,y)$ exhibits pulse doublets at intensity edges separating neighboring regions. A thresholding operator T can be applied to discard the illuminance component $T[d(x,y)] = \Delta r(x,y) = f(x,y)$. The reflectance map R is given by the inverse logarithm of the solution to Poisson's equation:

$$\Delta r(x,y) = f(x,y)$$

Given the proper boundary condition we can choose the corresponding direct method to solve the problem.

5 Application to a Direct Approach to SFS

Lee [7] suggested the use of a direct method for solving the nonlinear differential equation corresponding to the shape from shading problem :

$$\Delta p = \lambda h^2[R(p,q) - E]\,\frac{\partial R(p,q)}{\partial p}$$

$$\Delta q = \lambda h^2[R(p,q) - E]\,\frac{\partial R(p,q)}{\partial q}$$

with the corresponding matrix equations:

$$Mx = \lambda h^2 b(x)$$

$$M = \begin{pmatrix} A & O \\ O & A \end{pmatrix}$$

$$A = \begin{pmatrix} B & I & & \\ I & B & I & \\ & & & \\ & I & B & I \\ & & I & B \end{pmatrix} \tag{22}$$

$$B = \begin{pmatrix} -4 & 1 & & & \\ 1 & -4 & 1 & & \\ & & & & \\ & & 1 & -4 & 1 \\ & & & 1 & -4 \end{pmatrix}$$

$$b = \{\ \ldots\ (R(p_{ij},q_{ij}) - E)\frac{\partial R(p_{ij},q_{ij})}{\partial p_{ij}}$$
$$\ldots\ (R(p_{ij},q_{ij}) - E)\frac{\partial R(p_{ij},q_{ij})}{\partial q_{ij}}\ldots\}^T$$

and

$$x = \{p_{11}\ldots p_{ik}\ldots p_{kk}\ldots q_{11}\ldots q_{ik}\ldots q_{kk}\}^T$$

Since M is nonsingular for the Dirichlet boundary condition , the suggested iterations are :

$$x^{m+1} = \lambda h^2 M^{-1} b(x^m))$$

where x^o is an arbitrary initial condition.

Lee's algorithm does not enforce integrability . Using our approach the above algorithm can be modified in the following manner.

1. calculate $p^{(m+1)}$ *and* $q^{(m+1)}$ using Lee's algorithm
2. project $p^{(m+1)}$ *and* $q^{(m+1)}$ solving the equation

$$\Delta Z^{n+1} = \frac{\partial p^{n+1}}{\partial x} + \frac{\partial q^{n+1}}{\partial y}$$

3. use Z^{n+1} to determine $\bar{p}^{n+1}$ and $\bar{q}^{n+1}$ using central differencing
4. return to stage 1

Furthermore, if the boundary conditions are given on an irregular contour we can use embedding techniques.
If we are given Dirichlet boundary condition for $Z(x,y)$ as in the case of stereo we can use the same matrix C for both steps, because we are embedding the same matrix A.
The efficiency of Lee algorithm is yet to be determined but it is the only method with a proof for convergence amongst all the suggested iterative methods.

6 Application to the Optical Flow Problem

Optical flow is the distribution of apparent velocities of irradiance patterns in the dynamic image. We use a temporal series of images to compute the velocity field [2] .
If we want to minimize

$$\begin{aligned} \varepsilon(u,v) &= \alpha^2 \int\!\!\int \underbrace{(u_x^2+u_y^2)+(v_x^2+v_y^2)}_{smoothing\ term}\,dxdy \\ &+ \int\!\!\int_\Omega \underbrace{(E_x u + E_y v + E_t)^2}_{the\ flow\ equation}\,dxdy \end{aligned} \tag{23}$$

The Euler Lagrange equations for the cost function ε are given by

$$E_x^2 u + E_x E_y v = \alpha^2\Delta u - E_x E_t \tag{24}$$

$$E_x E_y u + E_y^2 v = \alpha^2 \Delta v - E_y E_t \tag{25}$$

Since we can estimate E_x , E_y , E_t from the given two images by

$$[E_x]_{ijk} = \frac{1}{2h}(E_{i+1jk} - E_{i-1jk})$$

$$[E_y]_{ijk} = \frac{1}{2h}(E_{ij+1k} - E_{ij-1k})$$

$$[E_t]_{ijk} = \frac{1}{h}(E_{ijk+1} - E_{ijk})$$

Equations (24) and (25) constitute a set of Poisson equations. In this discussion $h = \Delta x = \Delta y = \Delta t$ in Section 2 . One can rewrite (24) as

$$Mx = \frac{h^2}{\alpha^2}b(x)$$

where

$$M = \begin{bmatrix} A & \\ & A \end{bmatrix}$$

and A is as in (22) in the shape from shading problem .
Let

$$x = (u_{11}\ldots u_{ik}\ldots u_{kk}v_{11}\ldots v_{ik}\ldots v_{kk})^T$$

$$b(x) = \{\ldots\frac{\nu_{ijk}}{\mu_{ijk}}[E_x]_{ijk}\ldots\frac{\nu_{ijk}}{\mu_{ijk}}[E_y]_{ijk}\ldots\}^T\cdot\frac{\alpha^2}{h^2}$$

where

$$\mu_{ijk} = ([E_x]_{ijk})^2 + ([E_y]_{ijk})^2 + \frac{4}{h^2}\alpha^2$$

$$\nu_{ijk} = [E_x]_{ijk}\bar{u}_{ijk} + [E_y]_{ijk}\bar{u}_{ijk} + [E_t]_{ijk}$$

$$\bar{u}_{ijk} = \frac{1}{4}(u_{i-1,j,k} + u_{i,j+1,k} + u_{i+1,j,k} + u_{i,j-1,k})$$

and

$$\bar{v}_{ijk} = \frac{1}{4}(v_{i-1,j,k} + v_{i,j+1,k} + v_{i+1,j,k} + v_{i,j-1,k})$$

Again, we suggest the algorithm described in Section 5:

$$x^{n+1} = \frac{h^2 M^{-1}\ b(x^n)}{\alpha^2}$$

obtained by the direct method and appropriate boundary conditions.

We now discuss sufficient conditions for convergence of the above algorithm.
To see convergence it is easier to use the form

$$b_{u_{ij}}(u_{ij}v_{ij}) \triangleq [[E_x^2]_{ijk}\ u_{ij} + [E_x]_{ijk}[E_y]_{ijk}v_{ij} + [E_x]_{ijk}[E_t]_{ijk}]$$

$$b_{v_{ij}}(u_{ij}v_{ij}) \triangleq [[E_x]_{ijk}[E_y]_{ijk}u_{ij} + [E_y^2]_{ijk}\ v_{ijk} + [E_x]_{ijk}[E_t]_{ijk}]$$

It is obvious that $b_{u_{ij}}$ and $b_{v_{ij}}$ are Lipschitz functions [7] :

$$b_{u_{ij}}(u_{ij}, v_{ij}) - b_{u_{ij}}(u_{ij}^1, v_{ij}^1) \leq L^1\{(u-u^1)^2 + (v-v^1)^2\}^{\frac{1}{2}}$$

and

$$b_{v_{ij}}(u_{ij}, v_{ij}) - b_{v_{ij}}(u_{ij}^1\ v_{ij}^1) \leq L^2\{(u-u^1)^2 + (v-v^1)^2\}^{\frac{1}{2}}$$

Once we replace $\{[E_x]_{ijk}\}$ by $max_{ijk}\ |\{[E_x]_{ijk}\}|$ and $[E_x]_{ijk}$ by $max_{ijk}\ |\{[E_y]_{ijk}\}|$ and $\{[E_t]_{ijk}\}$ by $max_{ijk}\ |\{[E_t]_{ijk}\}|$,b becomes a linear function in u and v .
We now define $\nu_0 = max_{ijl}\{L_{ij}^l\}$ and $\lambda = \frac{1}{\alpha^2}$

We follow a theorem of Lee [7] to show that for
$\lambda \in [0\ 2\pi^2\nu_0^{-1}[1-\pi^2h^2/24]^2]$ the algorithm will converge .
We first assume that a solution exists, therefore it satisfies:

$$Mx = \lambda h^2 b(x)$$

Next, we note that

$$x^{m+1} - x = \lambda h^2 M^{-1}[b(x^m) - b(x)]$$

so

$$\|(x^{(m+1)}) - x\|_2 \leq \lambda h^2 \|M^{-1}\|_2 \nu_0 \|x^m - x\|_2$$

$$\|M^{-1}\|_2 = [8sin^2(\pi h/2)]^{-1} < [2\pi^2h^2(1-\pi^2h^2 24)^2]^{-1}$$

$$\|x^{m+1} - x\|_2 < \lambda h^2[2\pi^2h^2(1-\pi^2h^2/24)^2]^{-1}\nu_0\|x^m - x\|_2$$

Now, since $\lambda < 2\pi^2\nu_0^{-1}(1-\pi^2h^2/24)^2$
$\Rightarrow \lambda[2\pi^2(1-\pi^2h^2/24)^2]^{-1}\nu o < 1 \Rightarrow x^m$ converge to x.
Existence of a solution can be proved easily for the discrete version of the original penalty function. Consider the discrete function

$$\varepsilon_d(u,v) = \sum_i\sum_j(\alpha^2 s_{ij} + r_{ij})$$

$$s_{ij} = (u_{i+1j} - u_{ij})^2 + (u_{ij+1} - u_{ij})^2 + (v_{i+l,j} - v_{ij})^2 + (v_{ij+1} - v_{i,j})^2$$

$$r_{ij} = ([E_x]_{ij}u_{ij} + [E_y]_{ij}v_{ij} + [E_t]_{ij})^2$$

To minimize the above equation we have to solve a large system of sparse equations with $\lambda = \frac{1}{\alpha^2}$:

$$u_{ij} = \bar{u}_{ij} - \lambda h^2[[E_x]_{ij}u_{ij} + [E_y]_{ij}v_{ij} + [E_t]_{ij}[E_x]_{ij}]$$

$$v_{ij} = \bar{v}_{ij} - \lambda h^2[[E_x]_{ij}u_{ij} + [E_y]_{ij}v_{ij} + [E_t]_{ij}[E_y]_{ij}]$$

Note that we get exactly the same equation that we got using a discretized version of the Euler Lagrange equation.
Since we can restrict u and v to a rectangular closed domain from physical bounds on the maximum velocity, then $\varepsilon_d(u,v)$ becomes a continuous function defined on a compact subset S^{N^2} of R^{N^2} and therefore the minimum for the discrete problem exists. Thus for the discrete problem we have established the existence of a minimum, and stated sufficient conditions for the convergence of the suggested algorithm.

Acknowledgment

We wish to thank Professor Daniel Meiron of Caltech for introducing us to numerical methods of solving Elliptic Equations , and Vered Rom for helpful discussions.

References

[1] B. K. P. Horn and M. J. Brooks, "The Variational Approach to Shape from Shading", *Computer Vision , Graphics , and Image Processing*, vol.33, pp. 174–208, 1986.

[2] D. Terzopoulos, "Image Analysis Using Multigrid Relaxation Methods", *IEEE Transactions on Pattern Analysis and Machine Inteligence*, vol.8, pp. 129–139, March 1986.

[3] B. Buzbee, G. Golub, and C. Nielson, "On Direct Methods For Solving Poisson's Equations", *SIAM Journal of Numerical Analysis*, vol.7 no.4, pp. 627–656, 1970.

[4] A. Blake, "On the Geometric Information Obtainable from Simultaneous Observation of Stereo Contour and Shading", In *Report CSR 205-86*, Department of Computer Science , University of Edinbugh, 1986.

[5] R. T. Frankot and R. Chellappa, "A Method for Enforcing Integrability in Shape from Shading Algorithms", *IEEE Transactions on Pattern Analysis and Machine Inteligence*, Accepted for Publication, (Also presented at the First Int. Conf. on Computer Vision , London June 1987).

[6] B. Buzbee et al, "The Direct Solution of the Discrete Poisson Equation on Irregular Regeons", *SIAM Journal of Numerical Analysis*, vol.8 no.4, pp. 722–736, 1971.

[7] D. Lee, A Provably Convergent Algorithm for Shape from Shading, In *Proc. DARPA Image Undestanding Workshop*, pages 489–496, Miami Beach,Florida, December 1985.

[8] R. Courant and D. Hilbert, *"Methods of Mathematical Physics"*, Volume vol. 1, Interscience, New York, 1953.

[9] A.Bruss, "Is What You See What You Get", In *Proc. Int. Joint Conf. on Artificial Intelligence*, Karlsruhe,Germany, August 1983.

[10] T. Poggio and V. Torre, "Ill posed Problems and Regularization Analysis in Early Vision", *Artificial Intelligence Lab , M.I.T*, A.I.M 773, 1984.

[11] M. J. Brooks and B. K. P. Horn, Shape and Source From Shading, In *Proc. Int. Joint Conf. on Artificial Intelligence*, pages 932–936, Los Angeles,California, August 1985.

[12] M. Strat, *"A Numerical Method for Shape From Shading from a Single Image"*, M.S. thesis, M.I.T, Department of Elect. Engrg. and Comp. Sci., 1979.

[13] G. Strang, *"Linear Algebra and its Applications"*, Academic Press, New York, 1976.

[14] A. Rosenfeld and A. C. Kak, *"Digital Picture Processing"*, Volume vol. 1, Academic Press, New York, 1982.

[15] G. Smith, *"Numerical Solution of Parcial Differential Equations: Finite Difference Methods"*, University Press, Oxford, 1978.