Direct Analytical Methods for Solving Poisson Equations in Computer Vision Problems

TAL SIMCHONY, MEMBER, IEEE, RAMA CHELLAPPA, SENIOR MEMBER, IEEE, AND M. SHAO

Abstract—**The need to solve one or more Poisson equations of the general form:**

$$\Delta u = f$$

arises in several computer vision problems such as shape from shading, 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 describe some embedding techniques that may be useful when boundary conditions (obtained from stereo and occluding boundary) are defined on arbitrary contours. The suggested algorithms are computationally efficient owing to the use of fast orthogonal transforms. Applications to shape from shading, lightness, and optical flow problems 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. Experiments using synthetic images indicate that results comparable to multigrid can be obtained in a very small number of iterations.

Index Terms—**Direct methods, lightness problem, multigrid methods, optical flow, shape from shading, transform methods.**

I. Introduction

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

$$\Delta u = f$$

arises in several computer vision problems such as shape from shading (SFS) [1], the lightness problem [2], [3], and the computation of optical flow [4]. The currently used methods for solving these equations are iterative. In this paper we discuss direct analytical methods [5] for solving the Poisson equations arising in the problems mentioned above. Specifically, we show how direct methods can be used to enforce integrability in SFS problems when appropriate Dirichlet or Neumann boundary conditions are given. Occluding boundaries [6] give us the orientation of the surface on the boundary, corresponding to Dirichlet boundary conditions for the equations solving for surface orientation, and Neumann boundary conditions for the equations solving for surface height. As stereo information usually gives us the surface height along a closed contour [6], 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 Fourier transform and periodic boundary conditions reduces to an algorithm similar to the recently developed method of Frankot and Chellappa [7] for enforcing integrability in the SFS problem.

Very often the information provided from stereo and occluding boundary is defined on arbitrary contours. To handle these cases, we discuss some embedding techniques [8]. We further illustrate the applications of the method suggested here to the lightness problem [2], and to a direct method for estimating depth from shading [9]. Application of the direct analytical method to the optical flow problem, gives a new algorithm for computing the flow estimates. Using arguments similar to [9], we prove convergence of the iterative computations. As mentioned before, these algorithms are implemented using sine or cosine transforms and hence are computationally efficient. The direct method is much more efficient than the multigrid techniques for solving Poisson equation on a rectangular domain. This is the case for the lightness problem and the optical flow problem when velocity is known on the boundary. For the SFS problem, since the boundary is irregular it may appear as if the direct method is not computationally attractive compared to the multigrid relaxation method. However, for the specific SFS algorithm considered in this paper which enforces integrability, it turns out that the direct method is preferable. It is also important to note that the direct method can replace the relaxation step in the multigrid algorithm.

To illustrate the usefulness of the direct methods, we performed experiments with synthetic images used in [3]. Results comparable to multigrid were obtained in just a few iterations for the SFS, lightness, and optical flow problems.

The organization of this paper is as follows. Section II 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 the SFS problem for the case of rectangular Dirichlet or Neumann boundary conditions. The case of

Manuscript received February 29, 1988; revised December 29, 1989. Recommended for acceptance by O. D. Faugeras. This work was supported in part by the National Science Foundation under Grant MIP-84-51010 and matching funds from IBM, AT&T, and Hughes Aircraft Company. T. Simchony was also supported by ECI Telecom.

T. Simchony was with the Signal and Image Processing Institute, Department of Electrical Engineering—Systems, University of Southern California, Los Angeles, CA 90089. He is now with ECI Telecom, Petah Tikva, Israel.

R. Chellappa is with the Signal and Image Processing Institute, Department of Electrical Engineering—Systems, University of Southern California, Los Angeles, CA 90089.

M. Shao is with the Department of Mathematics, University of Southern California, Los Angeles, CA 90089.

IEEE Log Number 9034380.

a nonrectangular boundary contour is handled through embedding techniques [8] in Section III. An application to the lightness problem, a direct solution to the SFS problem, and the new optical flow algorithm with a proof for existence and convergence are in Sections IV, V, and VI, respectively. Experimental results are included in Section VII.

II. Enforcing Integrability in SFS for Rectangular Boundary Conditions

A. The SFS Problem and Its Ambiguities

In the SFS problem one tries to reconstruct a surface from its observed image intensity and surface orientation along occluding boundaries. Let $Z(x, y)$ be the unknown surface height, $Z_x = \partial Z(x, y)/\partial x$, $Z_y = \partial Z(x, y)/\partial y$, f, g be the surface orientation in stereoscopic coordinates, and $n = (n_1, n_2, n_3)$, the surface normal. The observed image intensity $E(x, y)$ is related to a C^2 surface $Z(x, y)$ by the irradiance equation:

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

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 order to obtain the surface estimate, one has to solve the irradiance equation with the corresponding boundary conditions. In the case of a Lambertian surface, one can further write (1) as

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

To see that the SFS problem is ill-posed, let us look at the following example. Consider the 2-D example of

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

With the trivial boundary conditions on 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

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

One wonders if ambiguities in SFS can be resolved by using the regularization theory proposed by Tikhonov [10] for solving ill-posed problems, and adapted to many vision problems by Poggio and Torre [11]. The regularization method suggests to convert the ill-posed problem into an optimization problem including the original equation and a regularization term which usually enforces smoothness constraints on the solution. In the example given above, the ambiguity cannot 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 nonlinear structure of the irradiance equation and not to the solving method. Therefore, it can only be eliminated by additional information such as sparse height measurements. Nevertheless, it has been shown [12] 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 strip approach, sufficient conditions for uniqueness of the SFS problem have been established for the case of an extremal contour and source at the eye [13], as well as for a general illumination direction with given boundary conditions (Dirichlet), and at most a unique maximal point of the intensity array [6].

Even if sufficient conditions are satisfied, due to the specific algorithms used, one may get ambiguous solutions if integrability is not enforced.

1) Ambiguities in SFS Algorithms and the Need for Integrability: 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. If we treat the orientation as two independent variables $(p(x, y) = Z_x(x, y), q(x, y) = Z_y(x, y))$ then the problem has infinitely many solutions, as it becomes underdetermined (we have one equation for two independent variables). The ambiguity is reduced significantly if regularization terms are used to enforce smoothness on the solution. The first to adapt the method were Ikeuchi and Horn [14]. They obtained the solutions by first finding the surface orientation in stereoscopic coordinates (f, g) and then constructing the surface. They minimize

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

assuming that the orientation is known on the boundaries from occluding boundaries. By using a different boundary condition that may be obtained from stereo techniques, we show that a well-posed problem may become ambiguous if integrability constraint is not enforced on the orientation. The integrability constraint requires that it is possible to "integrate" $p(x, y)$ & $q(x, y)$ such that the height Z obtained satisfies $p = Z_x$ & $q = Z_y$. Assume we are given a plane rotated around its x axis; then, f is homogeneous ($f = 0$) on the boundary, and g is arbitrary (g depends on the angle of rotation). 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 (3) is symmetric in f and therefore possible solutions are $(\pm f; g)$. If we now solve for Z using $-f$ and g and a given Dirichlet boundary conditions for Z we end up with a wrong solution. Thus the original well-posed problem becomes ambiguous without enforcing integrability. Similar arguments can be made for the minimization problem suggested in [15] as

$$\iint \left[(E - n \cdot \beta)^2 + \lambda(n_x^2 + n_y^2) + \mu(n^2 - 1) \right] dx\, dy \tag{4}$$

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, 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 [7] or $Z(x, y)$ at one point (where Z is not zero).

The examples discussed so far and the discussions in [1], [7] should convince the reader that integrability is an important issue in SFS algorithms. Integrability was enforced in [16] by minimizing the following cost function:

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

where $\epsilon = \Delta x = \Delta y$ and e_{ij} is the integrability penalty term which corresponds to an estimate for the integral in the counterclockwise direction 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 Jacobi–Picard relaxation algorithm was used. The algorithm converges slowly. In this work integrability was not enforced strictly, but a penalty term was used.

Horn and Brooks [1] imposed integrability using a penalty term with the following cost function:

$$\iint_{\Omega} \Big\{ \underbrace{\left(E(x, y) - R(p, q)\right)^2}_{\text{irradiance equation}} + \underbrace{\lambda(p_y - q_x)^2}_{\text{integrability term}} \Big\} \, dx\, dy$$

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, [7] a method was suggested for enforcing integrability using the theory of 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. An orthogonal projection was used for the case when 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. The problem of finding Z from p and q was discussed in [1] and [17]. It was suggested that

$$\iint \left[(Z_x - p)^2 + (Z_y - q)^2 \right] dx\, dy \tag{5}$$

should be minimized leading to the Euler equation

$$\Delta Z = p_x + q_y. \tag{6}$$

The solution for this problem in [1] is through Gauss–Seidel relaxation. 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 [5]. The direct method solves the problem using transform techniques (usually sine transform) in $O(n^2 \log n)$ operations. We use (6) 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 realistic boundary conditions given on a general contour, and do it in the same order of time needed in [7].

B. A New Algorithm for Enforcing Integrability

1) Dirichlet Boundary Conditions: In this section we assume that we are given a rectangular domain.

Let us look at the Poisson equation for the Dirichlet boundary condition

$$\Delta Z = f$$

with $Z = g$ (known) on the boundary. Suppose we discretize the Laplacian on the domain $((0, 0), (N_x, 0), (N_x, N_y), (0, N_y))$ to get

$$\alpha^2(Z_{i-1,j} - 2Z_{i,j} + Z_{i+1,j}) + (Z_{i,j-1} - 2Z_{i,j} + Z_{i,j+1}) = (\Delta y)^2 \tilde{f}_{ij} \tag{7}$$

where $\tilde{f}_{ij}$ is the function f modified for boundary conditions[1] and $\alpha^2 = (\Delta y / \Delta x)^2$. If we transform the 2-D vectors Z and $\tilde{f}$ to 1-D vector using lexicographic order,

[1] Since the value of the function on the boundary is known, we only solve for the interior points. For grid points next to a boundary, the value of the function on the boundary is subtracted from f_{ij} to obtain the correct equation for the discretized Laplacian.

we obtain the corresponding matrix equations:

$$AZ = \tilde{f} \tag{8}$$

$$A = \begin{pmatrix} B & \alpha^2 I & & \\ \alpha^2 I & B & \alpha^2 I & \\ & \alpha^2 I & B & \alpha^2 I \\ & & & B \end{pmatrix}$$

$$B = \begin{pmatrix} -2\ -2\alpha^2 & 1 & & \\ 1 & -2-2\alpha^2 & 1 & \\ & 1 & -2-2\alpha^2 & 1 \\ & & 1 & -2-2\alpha^2 \end{pmatrix}. \tag{9}$$

Lee [9] quotes the following results from [18]. The eigenvalues of A, the discretized Laplacian operator, are

$$\lambda ij = -4\left[\alpha^2 \sin^2\left(\frac{\pi i}{2N_x}\right) + \sin^2\left(\frac{\pi j}{2N_y}\right)\right]. \tag{10}$$

Assume for simplicity that $N_x = N_y = N$; then $A = H\Lambda H$, $H = S \otimes S$ a tensor product, where S is an $N - 1 \times N - 1$ matrix (the sine transform matrix) and Λ is a diagonal matrix consisting of the eigenvalues of A. In the general rectangular case, we transform (8) in both x and y directions to get the following equation:

$$\begin{aligned}&\left[\alpha^2(2\cos(\omega_x) - 2) + (2\cos(\omega_y) - 2)\right]Z(\underline{\omega})\\ &= -4\left[\alpha^2 \sin^2\left(\frac{\omega_x}{2}\right) + \sin^2\left(\frac{\omega_y}{2}\right)\right]Z(\underline{\omega})\\ &= \Delta y^2 F(\underline{\omega})\end{aligned} \tag{11}$$

where $\omega_x = \pi i/N_x$, $i = 1, \cdots, N_x - 1$, $\omega_y = \pi j/N_y$, $j = 1, \cdots, N_y - 1$, and $Z(\underline{\omega})$ and $F(\underline{\omega})$ are the corresponding variables in the transformed domain. For the case of periodic boundary conditions, one can easily get (11) from (7) using the discrete Fourier transform.

The resulting algorithm in the general rectangular case is:

1) Get $F(\underline{\omega})$ from $\tilde{f}$ using the discrete sine transform in both the x and y directions.

2) Solve the set of $N_x N_y$ independent equations

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

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

The efficiency of the method lies in the fact that the discrete sine transform (DST) can be implemented using FFT [19]. For more details on the use of DST to diagonalize a symmetric tridiagonal Toeplitz matrix the reader is referred to [19].

To obtain a more efficient algorithm we use the following formulation, let

$$Z_{ij} = \sum_{n=1}^{N_x - 1} a_{nj} \sin\left(\frac{n\pi i}{N_x}\right). \tag{13}$$

The problem with the sine transformation (13) taking values 0 at $i = 0$ and $i = N_x$ does not matter, as the values corresponding to $i = 0$ and $i = N_x$ are on the boundary (known). We are only interested in values for $i = 1, N - 1$.

By substituting (13) into (7) we obtain

$$\sum_{n=1}^{N_x - 1}\left[a_{n,j-1} + \left(\alpha^2\left(2\cos\left(\frac{n\pi}{N_x}\right) - 2\right) - 2\right)a_{n,j} + a_{n,j+1}\right] \cdot \sin\left(\frac{n\pi i}{N_x}\right) = (\Delta y)^2 \tilde{f}_{ij}. \tag{14}$$

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

$$\tilde{f}_{ij} = \sum_{n=1}^{N_x - 1} F_{nj} \sin\frac{n\pi i}{N_x} \tag{15}$$

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

$$\begin{aligned}&a_{n,j-1} + \left[\alpha^2\left(2\cos\left(\frac{n\pi}{N_x}\right) - 2\right) - 2\right]a_{n,j} + a_{n,j+1}\\ &= (\Delta y)^2 F_{nj}.\end{aligned} \tag{16}$$

We are then left with a problem of solving for $j = 1, \cdots, N_y - 1$ a tridiagonal matrix. It can be efficiently done using the LU decomposition technique [20], in $O(N_y N_x)$ operations. In simple words, the most efficient algorithm takes a sine transform of the input data along each column, and then efficiently solves the resulting tridiagonal matrix problems corresponding to each row using LU decomposition, which is performed faster than a transform in the row direction. This mixed method is more efficient than the full transform method outlined in (12) because it takes only $O(N^2)$ operations to solve a tridiagonal matrix problem using Gauss elimination, as opposed to $O(N^2 \log N)$ operations using the transform method.

The algorithm can now be stated as:

1) Get F_{nj} from $\tilde{f}_{ij}$ using discrete sine transformation.

2) Solve the system of tridiagonal equations for the a_{nj}'s.

3) Back transform $a \rightarrow Z$.

As all operations can be done in place, storage is minimized.

2) Relation to the Frankot–Chellappa Algorithm: We can easily relate our equation to the method in [7] using the Fourier series expansion of the function Z. By Fourier transforming (6)

$$\begin{aligned}&\mathfrak{F}\{\Delta Z = p_x + q_y)\}\\ &\quad\Rightarrow (-u^2 - v^2)Z(u, v)\\ &\quad= (\sqrt{-1}up(u, v) + (\sqrt{-1}vq(u, v).\end{aligned}$$

We have

$$Z(u, v) = \frac{-(\sqrt{-1}up(u, v) - (\sqrt{-1}vq(u, v)}{u^2 + v^2}$$

similar to (21) in [7]. We have thus established the correspondence between the two formulations in the continuous case.

In the case of periodic boundary conditions, the DFT of p_x and q_y 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 takes the form $(\sqrt{-1} \sin(\omega_x)$ [7] in the Fourier domain. We can use this fact to take only Fourier transform of p, q in order to calculate the Fourier transform coefficients of the right-hand side of (7).

We obtain the DFT of the difference equation

$$\begin{aligned}&\text{DFT}\Big\{(Z_{i-1,j} - 2Z_{i,j} + Z_{i+1,j})\\ &\qquad + (Z_{i,j-1} - 2Z_{i,j} + Z_{i,j+1})\\ &\quad= \frac{\Delta y}{2}[(p_{i,j+1} - p_{i,j-1}) + (q_{i+1,j} - q_{i-1,j})]\Big\}\\ &\quad\Rightarrow (2\cos(\omega_x) - 2 + 2\cos(\omega_y) - 2)Z(\underline{\omega})\\ &\quad= (\sqrt{-1}\sin(\omega_x)p(\underline{\omega}) + (\sqrt{-1}\sin(\omega_y)q(\underline{\omega})\\ &\quad\Rightarrow \left[-4\sin^2\left(\frac{\omega_x}{2}\right) - 4\sin^2\left(\frac{\omega_y}{2}\right)\right]Z(\underline{\omega})\\ &\quad= -\left[\left[\frac{\sin(\omega_x)}{\cos\left(\frac{\omega_x}{2}\right)}\right]^2 + \left[\frac{\sin(\omega_y)}{\cos\left(\frac{\omega_y}{2}\right)}\right]\right]Z(\underline{\omega})\\ &\quad= (\sqrt{-1}\sin(\omega_x)p(\underline{\omega}) + (\sqrt{-1}\sin(\omega_y)q(\underline{\omega})\\ &\quad\Rightarrow Z(\underline{\omega})\\ &\quad= \frac{-(\sqrt{-1}\sin(\omega_x)p(\underline{\omega}) - (\sqrt{-1}\sin(\omega_y)q(\underline{\omega})}{\sin^2(\omega_x)/\cos^2\left(\frac{\omega_x}{2}\right) + \sin^2(\omega_y)/\cos^2\left(\frac{\omega_y}{2}\right)}.\end{aligned}$$

Note that when $\omega_x \rightarrow 0 \cos^2(\omega_x/2) \rightarrow 1$. At low frequencies our result is similar to the result obtained in [7]. At 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 [7] may suffer from high frequency oscillations. The high frequency attenuation in the new algorithm given here turns out to be in our favor. Furthermore, the new algorithm enables us to work with a general domain (described later), and allows us to use the explicit value of Z on the boundary as in the case of stereo.

3) Neumann Boundary Conditions: Often the only information available on the surface boundary is the orientation. This is the case with occluding boundaries [6]. The orientation can be used to derive Neumann boundary conditions. The Neumann boundary conditions are also the natural boundary conditions corresponding to the variational problem (5). We obtain the following Euler equation:

$$\begin{aligned}&\Delta Z = f \text{ in } R\\ &\frac{\partial Z}{\partial \eta} = g \text{ on } \partial R(\text{known})\end{aligned}$$

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:

$$AZ = \tilde{f}$$

where

$$A = \begin{bmatrix} B & 2\alpha^2 I & & \\ \alpha^2 I & B & \alpha^2 I & \\ & \alpha^2 I & B & \alpha^2 I \\ & & 2\alpha^2 I & B \end{bmatrix}$$

$$B = \begin{bmatrix} -2 - 2\alpha^2 & 2 & & \\ 1 & -2 - 2\alpha^2 & 1 & \\ & 1 & -2 - 2\alpha^2 & 1 \\ & & 2 & -2 - 2\alpha^2 \end{bmatrix}$$

We first note that the matrix A is singular. (It can be checked by multiplying the matrix A by a vector with all entries equal to 1.) We can determine the solution in the

Neumann case only up to a constant. In computer vision problems, we will have to choose it or get the constant from physical properties.

In the rest of this section we assume for simplicity that $\alpha = 1$. In order to decompose the equations we assume that

$$Z_{ij} = \sum_{n=0}^{N} a_{nj} \cos\left(\frac{in\pi}{N}\right) \tag{17}$$

with

$$\Delta x = \Delta y.$$

In this case we use a cosine transform which diagonalizes the matrix A. Note that the solution on boundary needs to be determined as well.

Substituting (17) into (7) we get

$$\sum_{n=0}^{N} \left[\left(2 \cos\left(\frac{n\pi}{N}\right) - 2 \right) a_{nj} + a_{nj+1} - 2a_{nj} + a_{nj-1} \right] \cdot \cos\left(\frac{in\pi}{N}\right) = (\Delta y)^2 \tilde{f}_{ij}. \tag{18}$$

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

$$\tilde{f}_{ij} = \sum_{n=0}^{N} F_{nj} \cos\left(\frac{n\pi i}{N}\right) \tag{19}$$

and substituting (19) into (18) and equating coefficients of the cosine terms,

$$\left(2 \cos\left(\frac{n\pi}{N}\right) - 2 \right) a_{nj} + a_{nj+1} - 2a_{nj} + a_{nj-1} = (\Delta y)^2 F_{nj}. \tag{20}$$

The tridiagonal systems (one for each row) are nonsingular except for $n = 0$, which is treated separately.

In order to implement the direct method, we use the appropriate 1-D cosine transform [21].

$$F_{ij} = \frac{1}{N}\tilde{f}_{0j} + \frac{2}{N}\sum_{n=1}^{N-1} \tilde{f}_{nj} \cos\left(\frac{n\pi i}{N}\right) + \frac{1}{N}(-1)^i \tilde{f}_{Nj}$$

Note that this transform can be implemented using FFT. The first and the last elements are weighted differently, in order to construct an orthogonal transform. We now either transform F in the other direction to obtain a diagonal set of equations, or solve directly for the a_{ij}'s using LU decomposition.

The inverse transform is of the form

$$x_{ij} = \frac{1}{2} a_{0j} + \sum_{n=1}^{N-1} a_{nj} \cos\left(\frac{n\pi i}{N}\right) + \frac{1}{2}(-1)^i a_{Nj}.$$

Neumann boundary conditions appear as natural boundary conditions for variational problems, involving minimization of integrals of quadratics of the directional derivatives. This is the case with optical flow, where the normal derivatives of u, v (the velocities in the x, y directions respectively) at the boundary are set to zero.

Other ways to solve the problem include block cyclic reduction [5].

III. Integrability Using Embedding Techniques for Nonrectangular 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 stereo, occluding boundary [6]. It 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 an 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 the irregular Dirichlet boundary conditions problem. Similar techniques are also known [8] for the Neumann boundary conditions.

The initial problem is

$$\begin{aligned} \Delta u &= f \text{ in } R \\ u &= g \text{ on } \partial R. \end{aligned} \tag{21}$$

After discretization we have

$$\begin{aligned} \Delta_h u &= f \text{ in } R_h \\ u &= g \text{ on } \partial R_h. \end{aligned} \tag{22}$$

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

$$\left.\begin{aligned} \Delta_h u &= f \text{ in } R_h' - S_h \\ u &= g \text{ on } S_h \cup \partial R_h'. \end{aligned}\right\} \tag{23}$$

The corresponding matrix equations are $Ax = f$.

The solution to the above problem satisfies the discretized problem (22). 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 tridiagonal Toeplitz and the new problem thus defined can be solved using sine transform. We will have to relate the solution of $By = \bar{f}$ to the problem $Ax = f$ in order to find the solution to the original problem. We follow the work in [8].

Assume for simplicity that the first p rows of A need to be modified. This can be achieved by multiplying A by a permutation matrix. We should not do it explicitly in the computational procedure. It should be implicitly done by indexing, or else we destroy the special structure of B.

We now partition A as

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

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 tridiagonal Toeplitz matrix.

We are given $Ax = f$ and write

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

Partition the vector $\bar{f}$ as

$$\bar{f} = \begin{pmatrix} \bar{f}_1 \\ \bar{f}_2 \end{pmatrix}.$$

For an arbitrary $p \times p$ matrix W, we then define a $n \times p$ matrix $\overline{W}$ as

$$\overline{W} = \begin{pmatrix} W \\ O \end{pmatrix}. \qquad (24)$$

In our problem W can be chosen to be the identity matrix.

Define the $p \times p$ matrix C by:

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

Assume that 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

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

In [8], it is shown 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

$$\begin{aligned} A^{-1} &= (B + FG)^{-1} \\ &= B^{-1}\left(I - F(I + GB^{-1}F)^{-1} GB^{-1}\right) \end{aligned}$$

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

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

If A is nonsingular one can 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 $C = A_1 B^{-1} \overline{W}$ 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 C and factorize it. As the applications we suggest in this paper need to solve the Poisson equations repeatedly, we perform the above computation only 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 [20].

The solution x can then be obtained from:

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

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

IV. Application to Lightness Problem

The lightness of a surface is the perceptual correlate of its reflectance [2]. The irradiance at a point is proportional to the product of the illuminance and reflectance at the corresponding point on the surface. The goal is to compute lightness from image irradiance without any precise knowledge of either the reflectance or illuminance.

Following [3], if E = irradiance, S = illuminance, R = reflectance, then

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

Taking natural logarithms,

$$e(x, y) = s(x, y) + r(x, y) \qquad (25)$$

and applying the Laplacian operator Δ on (4) gives

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

If the 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 conditions we can choose the corresponding direct method to solve the problem.

V. Application to a Direct Approach to SFS

Lee [9] suggested the use of a direct method for solving the nonlinear differential equation corresponding to the shape from shading problem (using stereoscopic coordinates):

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

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

with the corresponding matrix equations:

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

where

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

$$A = \begin{pmatrix} B & I & & \\ I & B & I & \\ & I & B & I \\ & & I & B \end{pmatrix} \tag{26}$$

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

$$b = \left\{ \cdots, \left(R(p_{ij}, q_{ij}) - E_{ij}\right) \frac{\partial R(p_{ij}, q_{ij})}{\partial p_{ij}}, \cdots, \left(R(p_{ij}, q_{ij}) - E_{ij}\right) \frac{\partial R(p_{ij}, q_{ij})}{\partial q_{ij}}, \cdots \right\}^T$$

and

$$x = \{p_{11}, \cdots, p_{ik}, \cdots, p_{kk}, \cdots, q_{11}, \cdots, q_{ik}, \cdots, q_{kk}\}^T. \tag{27}$$

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

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

where x^0 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

$$\Delta Z^{m+1} = \frac{\partial p^{m+1}}{\partial x} + \frac{\partial q^{m+1}}{\partial y}. \tag{28}$$

3) Use Z^{m-1} to determine $\tilde{p}^{m+1}$ and $\tilde{q}^{m+1}$ using central differencing.

4) Return to step 1.

A. Comparison with Multigrid Relaxation Methods

In the SFS problem, the domain is usually irregular. It may appear as if the direct method is not computationally attractive compared to the multigrid relaxation method. However, for the specific SFS algorithm considered above which enforces integrability, we argue that the direct method is preferable. We directly solve (28) in each iteration of the algorithm. A multigrid relaxation method requires a lot of iterations to accomplish this step of the algorithm. The equations for the orientation variables (p and q) are nonlinear. The precomputation of the matrix C required by the embedding technique is done once and used throughout the Jacobi–Picard iterations. Furthermore, if we are given Dirichlet boundary conditions for $Z(x, y)$ as in the case of stereo, we can use the same matrix C for both the orientation and depth computation steps of the algorithm, because we embed the same matrix A. In the experiments we conducted, our algorithm required a smaller number of iterations than the multigrid relaxation algorithm used in [3] for the same image. In addition our algorithm enforces integrability while the multigrid relaxation algorithm does not. A recent formulation for the SFS problem in [22] suggests an alternative way for enforcing integrability which may be more suitable for the multigrid relaxation algorithm. The direct method can also replace the relaxation step in the multigrid algorithm, giving an algorithm that combines the benefits of the two.

VI. 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 [3].

We want to minimize

$$\epsilon(u, v) = \alpha^2 \iint \underbrace{(u_x^2 + u_y^2) + (v_x^2 + v_y^2)}_{\text{smoothing term}} dx\, dy + \iint_\Omega \underbrace{(E_x u + E_y v + E_t)^2}_{\text{the flow equation}} dx\, dy. \tag{29}$$

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

$$E_x^2 u + E_x E_y v = \alpha^2 \Delta u - E_x E_t \tag{30}$$

$$E_x E_y u + E_y^2 v = \alpha^2 \Delta v - E_y E_t. \tag{31}$$

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 (30) and (31) constitute a set of Poisson equations. In this discussion $h = \Delta x = \Delta y = \Delta t$ defined in Section II. Although the problem is linear, it is solved using Jacobi–Picard iterations to maintain the special structure of the matrix M defined below. One can solve (30) and (31) for u and v and obtain the following matrix equations:

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

where

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

and A is as in (26).

Let

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

$$b(x) = \left\{ \cdots \frac{\nu_{ijk}}{\mu_{ijk}} [E_x]_{ijk} \cdots \frac{\nu_{ijk}}{\mu_{ijk}} [E_y]_{ijk} \cdots \right\}^T \cdot \frac{\alpha^2}{h_2}$$

where

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

$$\nu_{ijk} = 4 * \left([E_x]_{ijk} \bar{u}_{ijk} + k[E_y]_{ijk} \bar{u}_{ijk} + [E_t]_{ijk}\right)$$

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

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 V:

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

obtained by the direct method and appropriate boundary conditions. We now discuss sufficient conditions for convergence of the above algorithm. Since b is a linear function of u and v it is obvious that $b_{u_{ij}}$ and $b_{v_{ij}}$ are Lipschitz functions [9]:

$$b_{u_{ij}}(u, v) - b_{u_{ij}}(u^1, v^1) \le L^1_{ij} \left\{ (u - u^1)^2 + (v - v^1)^2 \right\}^{1/2}$$

and

$$b_{v_{ij}}(u, v) - b_{v_{ij}}(u^1 \;\; v^1) \le L^2_{ij} \left\{ (u - u^1)^2 + (v - v^1)^2 \right\}^{1/2}.$$

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

We follow a theorem of Lee [9] to show that for $\lambda \in [0,\ 2\pi^2 \nu_0^{-1} [1 - \pi^2 h^2/24]^2]$, the algorithm will converge.

We first assume that a solution satisfying

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

exists. Next, we note that

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

so

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

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

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

Now, since $\lambda < 2\pi^2 \nu_0^{-1} (1 - \pi^2 h^2/24)^2 \Rightarrow \lambda[2\pi^2 (1 - \pi^2 h^2/24)^2]^{-1} \nu_0 < 1 \Rightarrow x^m$ converge to x.

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

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

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

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

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

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

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

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 $\epsilon_d(u, v)$ 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.

VII. Experimental Results

A. *The Lightness Problem*

The lightness algorithm described in Section IV was tested on two synthesized Mondrian scenes [3] consisting of patches of uniform reflectance, subjected to an illuminance which increased exponentially (in the first scene) and quadratically (in the second) from left to right. The original image with exponential illuminance, which is 129 × 129 pixels in size, is shown in Fig. 1(a). The grid function shown in Fig. 1(b) was computed by maintaining only the peaks in the Laplacian of r^h_{ij}. Zero Neumann boundary conditions were provided around the edges of the image. Fig. 1(c) shows the reconstructed Mondrian which now lacks all of the illumination gradient. The experiment was repeated on the Mondrian scene with quadratic illumi-

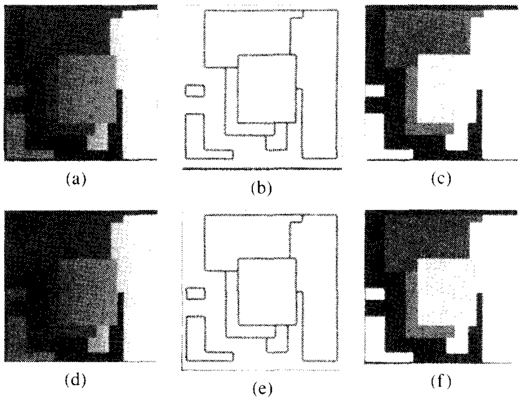


Fig. 1. The lightness problem experiments. (a) The original Mondrian image with exponential illumination. (b) Output of the Laplacian operator and thresholding. (c) Reconstructed Mondrian image, with suppressed exponential illumination. (d) The original Mondrian image with quadratic illumination. (e) Output of the Laplacian operator and thresholding. (f) Reconstructed Mondrian image, with suppressed quadratic illumination. The reconstructed images were obtained in one iteration.

nance in Fig. 1(d). The grid function and the reconstructed image are in Figs. 1(e) and (f), respectively.

The reconstruction of the images from the functions in Figs. 1(b) and (e) needs only one iteration. In comparison, a multigrid lightness algorithm required numerical operations equivalent to 10 iterations on the finest grid to compute each of the solutions, and a single-grid relaxation algorithm needed more than 1000 iterations.

B. The Shape-from-Shading Problem

The SFS algorithm was tested on a synthetically generated image [3] of a Lambertian sphere distantly illuminated from the viewing direction by a point source. The original image, which is 64 × 64 pixels in size, is shown in Fig. 2(a). The image is quantized to 256 irradiance levels. The sphere was assumed to be Lambertian with constant albedo, so we could use the reflectance map:

$$R(p, q) = \max\left[0, \cos(\theta)\right]$$

where

$$\cos(\theta) = \frac{1.0 + p_s p + q_s q}{\sqrt{1.0 + p_s^2 + q_s^2}\sqrt{1.0 + p^2 + q^2}}$$

and where p_s and q_s are the light source direction components. Depth and orientation of the surface were specified on a general closed contour presented in Fig. 2(b). The computation started from the zero initial conditions $p = q = z = 0$. Only two iterations were needed to reconstruct both the orientation and depth maps. The orientation and depth maps are presented in Figs. 2(c) and (d), respectively.

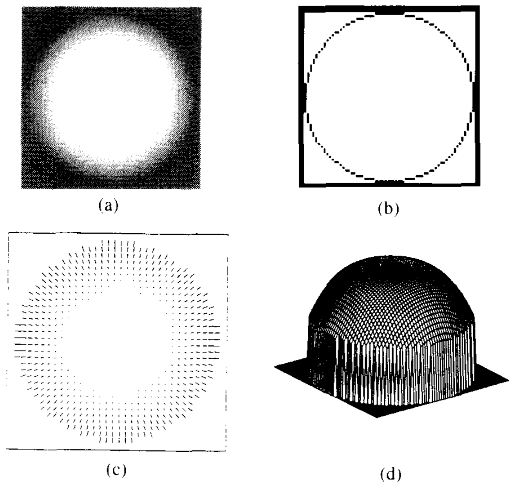


Fig. 2. The shape from shading experiment. (a) The sphere image with light source at the viewing direction. (b) The boundary of the irregular domain. (c) Reconstructed needle map for the surface orientation. (d) Reconstructed depth map of the surface. The reconstructed results were obtained in two iterations.

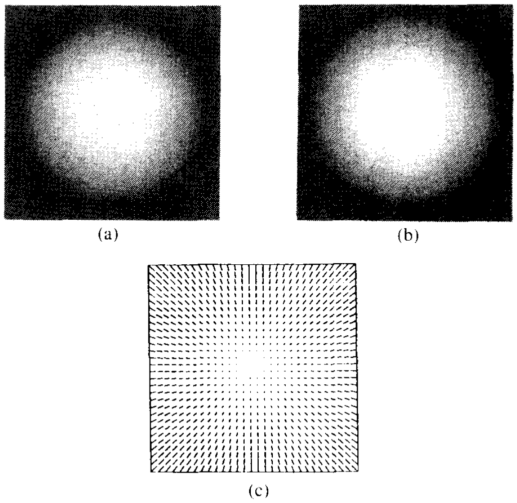


Fig. 3. Optical flow experiment 1. (a) The sphere image at time T_k. (b) The expanding sphere image at time T_{k+1}. (c) Reconstructed needle map for the velocity field. The reconstructed results were obtained in three iterations.

C. The Optical Flow Problem

The optical flow algorithm was first tested on a synthetically generated image [3] of a Lambertian sphere distantly illuminated from the viewing direction by a point source, which is right in front of the sphere. The sphere expanded uniformly over two frames. The first frame, which is 256 × 256, is shown in Fig. 3(a). The next frame, in which the sphere has expanded, is shown in Fig. 3(b). All the images are quantized to 256 levels. The actual velocity field was compared to the simulated result.

The velocity field was specified around a rectangular domain of size 129 × 129. The computation was started from the zero initial approximation $u = v = 0$. The solution was obtained in three iterations, and the accuracy of the result was approximately 10^{-5}. The velocity field is presented in Fig. 3(c). In this experiment we used the exact Dirichlet boundary conditions.

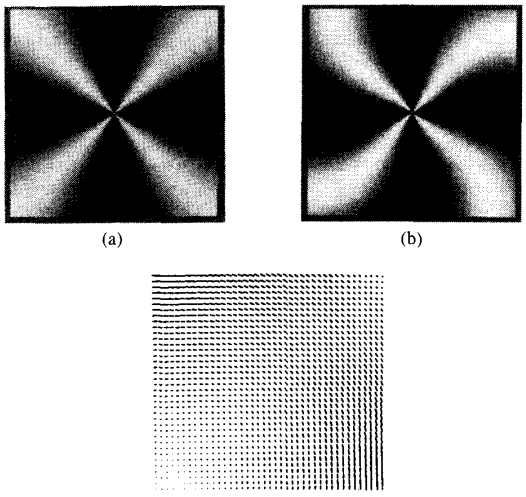


Fig. 4. Optical flow experiment 2. (a) The disk image at time T_k. (b) The rotated disk image at time T_{k+1}. (c) Reconstructed needle map of the top right quadrant of the velocity field. The reconstructed results were obtained in three iterations.

In a second experiment we considered a more involved type of motion. A disk of radius one, with the intensity shown in Fig. 4(a) was rotated about its center with a varying angular velocity ω which depends on the distance from the center as

$$\omega = \omega_0 r(1 - r)$$

where ω_0 is a constant, r is the distance from the center of the sphere. The rotated disk is in Fig. 4(b). We estimated the velocity field, using Dirichlet boundary conditions, and zero initial conditions $u = v = 0$. The solution was obtained in three iterations, but the accuracy was only of the order of 10^{-2}. The velocity field is presented in Fig. 4(c). We also tried to run the algorithm with natural boundary conditions, zero normal derivatives for u, v on the boundary (corresponding to Neumann boundary conditions, discussed in Section II-C). The results were not good owing to the sensitivity of the algorithm to boundary conditions. We conclude that the suggested algorithm is very sensitive to boundary conditions, and should not be used with the natural boundary conditions for optical flow.

Acknowledgment

We are thankful to the anonymous reviewers for several critical comments that led to the improvement of the paper.

References

[1] B. K. P. Horn and M. J. Brooks, "The variational approach to shape from shading," *Comput. Vision, Graphics, Image Processing*, vol. 33, pp. 174-208, Feb. 1986.

[2] B. K. P. Horn, "Determining lightness from an image," *Comput. Graphics, Image Processing*, vol. 3, pp. 277-299, Dec. 1974.

[3] D. Terzopoulos, "Image analysis using multigrid relaxation methods," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 129-139, Mar. 1986.

[4] B. K. P. Horn and B. Schunck, "Determining optical flow," *Artificial Intell.*, vol. 17, pp. 185-203, Aug. 1981.

[5] B. Buzbee, G. Golub, and C. Nielson, "On direct methods for solving Poisson's equations," *SIAM J. Numer. Anal.*, vol. 7, pp. 627-656, Dec. 1970.

[6] A. Blake, "On the geometric information obtainable from simultaneous observation of stereo contour and shading," Dep. Comput. Sci., Univ. Edinburgh, Rep. CSR 205-86, 1986.

[7] R. T. Frankot and R. Chellappa, "A method for enforcing integrability in shape from shading algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 439-451, July 1988.

[8] B. Buzbee *et al.*, "The direct solution of the discrete Poisson equation on irregular regions," *SIAM J. Numer. Anal.*, vol. 8, pp. 722-736, Dec. 1971.

[9] D. Lee, "A provably convergent algorithm for shape from shading," in *Proc. DARPA Image Understanding Workshop*, Miami Beach, FL, Dec. 1985, pp. 489-496.

[10] A. N. Tikhonov and V. A. Arsenin, *Solutions of Ill-Posed Problems*. Washington, DC: Winston, 1977.

[11] T. Poggio and V. Torre, "Ill posed problems and regularization analysis in early vision," Artificial Intell. Lab., Massachusetts Inst. Technol., Cambridge, Rep. A.I.M. 773, Apr. 1984.

[12] A. Blake and A. Zisserman, *Visual Reconstruction*. Cambridge, MA: MIT Press, 1987.

[13] A. Bruss, "Is what you see what you get," in *Proc. Int. Joint Conf. Artificial Intelligence*, Karlsruhe, Germany, Aug. 1983.

[14] K. Ikeuchi and B. K. P. Horn, "Numerical shape from shading and occluding boundaries," *Artificial Intell.*, vol. 17, pp. 141-185, Aug. 1981.

[15] M. J. Brooks and B. K. P. Horn, "Shape and source from shading," in *Proc. Int. Joint Conf. Artificial Intelligence*, Los Angeles, CA, Aug. 1985, pp. 932-936.

[16] M. Strat, "A numerical method for shape from shading from a single image," M.S. thesis, Dep. Elec. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, 1979.

[17] K. Ikeuchi, "Constructing a depth map from images," Artificial Intell. Lab., Massachusetts Inst. Technol., Cambridge, Rep. A.I.M. 744, Aug. 1983.

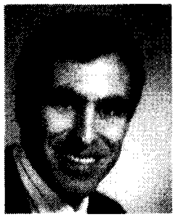
[18] G. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*. London: Oxford, University Press, 1978.

[19] A. Rosenfeld and A. C. Kak, *Digital picture processing*, vol. 1. New York: Academic, 1982.

[20] G. Strang, *Linear Algebra and Its Applications*. New York: Academic, 1976.

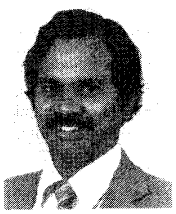
[21] P. N. Swarztrauber, "The methods of cyclic reduction, Fourier analysis and the FACR algorithm for the discrete solution of Poisson's equation on a rectangle," *SIAM Rev.*, vol. 19, pp. 490-501, July 1977.

[22] M. Shao, T. Simchony, and R. Chellappa, "New algorithm for reconstruction of a 3-D depth map from one or more images," in *Proc. Computer Vision and Pattern Recognition Conf.*, Ann Arbor, MI, June 1988.



Tal Simchony (S'86-M'89) was born in Tel Aviv, Israel, on January 18, 1956. He received the B.S. degree in mathematics and computer science and the M.S. degree in applied mathematics from the Tel Aviv University in 1982 and 1985, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1988.

During 1982-1985, he was with ECI Telecom as a software and system engineer. During 1985-1988 he was a Research Assistant at the Signal and Image Processing Institute, USC. He is currently at ECI Telecom as Deputy Chief Engineer working on speech compression algorithms on digital networks. His research interests include optimization, learning, and computer vision.



Rama Chellappa (S'79-M'87-SM'83) was born in Madras, India. He received the B.S. degree (honors) in electronics and communications engineering from the University of Madras in 1975, the M.S. degree (with distinction) in electrical communication engineering from the Indian Institute of Science in 1977, and the M.S. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, in 1978 and 1981, respectively.

During 1979-1981, he was a Faculty Research Assistant at the Computer Vision Laboratory, University of Maryland, College Park. Since 1986, he has been an Associate Professor in the Electrical Engineering—Systems, and as of September 1988, he became the Director of the Signal and Image Institute at the University of Southern California, Los Angeles. His current research interests are in signal and image processing, computer vision, and pattern recognition.

Dr. Chellappa is a member of Tau Beta Pi and Eta Kappa Nu. He coedited two volumes of selected papers on image analysis and processing, published in Fall 1985. He served as Associate Editor for IEEE Transactions on Acoustics, Speech, and Signal Processing and currently is a Co-Editor of *Computer Vision, Graphics and Image Processing: Graphic Models and Image Processing*. He was a recipient of a National Scholarship from the Government of India during 1969-1975. He received the 1975 Jawaharlal Nehru Memorial Award from the Department of Education, Government of India, the 1985 Presidential Young Investigator Award, and the 1985 IBM Faculty Development Award. He was the General Chairman of the 1989 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society Workshop on Artificial Intelligence for Computer Vision, and also Program Co-Chairman of the NSF sponsored Workshop on Markov Random Fields for Image Processing, Analysis, and Computer Vision.

M. Shao, photograph and biography not available at the time of publication.