USC–SIPI REPORT #223

Computation of Dense Optical Flow with A Parametric Smoothness Model

by

Navid Haddadi and C.-C. Jay Kuo

October 1992

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

Computation of Dense Optical Flow with A Parametric Smoothness Model *

Navid Haddadi† and C.-C. Jay Kuo †

October 22, 1992

Abstract

A new algorithm is presented for the computation of dense optical flow and motion boundaries from an image sequence using two or more frames. The algorithm is based on a novel parametric smoothness model by decomposing optical flow into irrotational and solenoidal fields, and imposing the smoothness constraint on each field separately. This model implies smooth translation and rotation of the underlying motion process. In contrast, the smoothness constraints used in all previous work do not distinguish the translational and rotational components but simply combine them as a whole. The derivation of the parametric smoothness model sheds new light on the interpretation of the conventional membrane model. The problem of over-smoothing across motion boundaries can be resolved to a high degree by successively improving the estimate of the parameters of the smoothness model. Significant improvements by the proposed new algorithm over classical gradient based methods have been obtained for a class of test problems.

1 Introduction

Optical flow is a 2-D vector field that measures the disparity between adjacent frames in a sequence of images. The information contained in the computed flow field can be used for image sequence compression [19, 20] or the determination of the relative depth map and/or three-dimensional motion and structure [1]. Since the flow field is usually not unique over extensive regions of its support, there have been many attempts to limit the scope of estimation to regions for which only a good estimate can be obtained. These techniques range from computation of the flow only for highly conspicuous points [3] to flow along some boundaries [5, 11, 26] or some regions obtained by thresholding an appropriate function [7, 15]. Although we do not explicitly address any application issues in this paper,

*This work was supported by a National Science Foundation Young Investigator Award (ASC-9258396).

†The authors are with the Signal and Image Processing Institute and the Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, California 90089-2564. E-mail: haddadi@sipi.usc.edu and cckuo@sipi.usc.edu.

our intended target is motion compensated coding. For such an application, we need an estimate of the flow over the entire region of support, which is often known as dense optical flow. In this paper, we are mainly concerned with the computation of dense optic flow from at least two frames of an image sequence.

The desired algorithm should have the following properties.

- An estimate of the flow should be available for all of the pixels in the region of support with the exception of occluded regions which must be identified.
- The algorithm is able to provide a robust estimate of the flow from two frames and improved estimation quality with more frames. This requirement is needed due to practical limitation on the size of the frame storage buffer in actual implementation.
- The complexity of the algorithm should be low and, hopefully, in the order of $N^2 \log(N)$ per image frame of size $N \times N$.
- The algorithm is capable of estimating subpixel flow fields. This is useful for applications where only a low resolution estimate of the flow is needed so that computational time can be significantly reduced in processing the subsampled image sequence.

The algorithm presented in this paper meets all above requirements. It is based on a regularization approach similar to previous work in [13, 17] but with a novel parametric smoothness model consisting of two parameter functions $\rho(x,y)$ and $\omega(x,y)$. Borrowing techniques from fluid mechanics, we decompose optical flow into two flow fields, i.e. the irrotational and solenoidal fields, and impose the smoothness constraint on each field separately. This implies smooth translation and rotation of the underlying motion process. In contrast, the smoothness constraints used in all previous work do not distinguish the translational and rotational components but simply combine them as a whole. The conventional membrane model is shown to be equivalent to the choice of $\rho(x,y) = 0$ and $\omega(x,y) = 0$ in our model. An accurate estimate of the parameters ρ and ω can greatly enhance the computed flow field. The problem of over-smoothing across motion boundaries can be resolved to a high degree, and an estimate of occluding boundaries is obtained as a by-product. Since the nature of motion is characterized by ρ and ω, we may incorporate a prior knowledge of motion obtained from earlier frames in an image sequence into ρ and ω by assuming that they are smooth time-varying functions. In this work, we focus on the two frame case where an iterative algorithm is proposed for estimating ρ and ω. The general multiple frame case will be reported later.

This paper is organized as follows. Gradient based methods for optical flow computation are reviewed in Section 2. To put various problems arising in optical flow computation into perspective, some fundamental characteristics of optical flow are discussed in Section 3. Section 4 forms the crux of our method, where a parametric smoothness model is derived. In Section 5, we apply our algorithm to a set of four test problems ranging from a simple translation and rotation to combined motion along all axes. We end our presentation in Section 6 with a few concluding remarks. Numerical implementation of the algorithm is given in the Appendix. In particular, we derive equations for both constrained gradient projection method and the Gauss-Seidel relaxation which is appropriate for multigrid implementation.

2 Review of Gradient Based Methods for Computing Optical Flow

There have been many efforts reported in the literature for computing optical flow. Most of these methods can be classified into one of four major categories: gradient based [13], contour based [5, 11, 26], correlation based [2, 23], and spatiotemporal-frequency based [10]. Since our method falls into the gradient based category, its basic idea and some related results are reviewed in this section.

Given an image sequence $E(x, y, t)$, optical flow (u, v) is defined as velocities in a small time duration such that

$$E(x + u\delta t, y + v\delta t, t + \delta t) = E(x, y, t). \tag{2.1}$$

By taking the first term in the Taylor series expansion of (2.1), optical flow can be modeled to the first degree of approximation by the optical flow constraint equation:

$$E_x u + E_y v + E_t = 0, \tag{2.2}$$

where the subscript denotes partial differentiation with respect to the given variable. It is evident that the optical flow constraint imposes only one constraint for two unknown variables per image pixel so that we need at least one more constraint to uniquely determine the flow field. All gradient based methods use the optical flow constraint, but consider different ways to impose additional constraints.

One way to avoid the under-determined problem is to impose the constant velocity constraint for neighboring flow fields. Several methods were proposed under this assumption.

The local optimization method assumes two or more neighboring pixels to have the same displacement value so that a set of linear equations based on the optical flow constraint is obtained [14, 15, 19]. With the clustering method, one attempts to detect clusters of the intersection of optical flow constraint equations from different pixels in the velocity space [21]. With the differential method, one can obtain three more equations for each pixel by differentiating the optical flow constraint equation with respect to x, y and t under the assumption that u and v are constant [9, 24].

Another approach to solve the under-determined problem is via regularization. The method of Horn and Schunck [13] is perhaps the most popular technique for computing optical flow due to the simplicity of its implementation. Here, one assumes that the overall flow satisfies a certain global smoothness constraint. For example, by using the membrane model, one defines an energy function

$$\mathcal{E} = \iint (E_x u + E_y v + E_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2)\, dx\, dy, \tag{2.3}$$

which is minimized for the desired estimate of (u, v). The λ is known as the Lagrange multiplier which determines the degree of smoothness in the computed flow. There are two major difficulties with this method. First, this method ignores the presence of occluded regions and very inaccurate flow may be concluded in these regions. Second, the global smoothness constraint tends to smooth the solution across discontinuities so that there is a leakage of the flow into the stationary background.

An attempt was made in [6] to remedy these problems by setting the smoothness term to zero in regions near contours of zero-crossings under the assumption that motion discontinuity is a subset of intensity zero-crossings. Another solution based on an oriented smoothness constraint was proposed by Nagel in [17]. A weighting matrix depending on gray value variations was introduced so that the smoothness requirement would be retained essentially only for the normal component of the flow. By enforcing the flow at the gray value corners, the initially proposed weighting matrix contained both first- and second-order derivatives of the intensity process. However, Nagel recommended to use only first-order derivatives in his later work [18] to make the computational algorithm simple and robust.

3 Discussion on Optical Flow Constraint Equation and Flow Field Characterization

The validity of the optical flow constraint equation was studied in [22]. In this section, we will provide more insights into this equation and discuss its limitation, which provides us a better understanding of the nature of optical flow.

Given a optical flow field (u, v), we can decompose it into the normal and tangential components denoted by α and β, respectively, along the iso-brightness contours as shown in Figure 1, where the vector (p, q) with $p = E_x$ and $q = E_y$ is the image gradient direction. We can rewrite the GCE, i.e. (2.2), in the product form as

$$(E_x, E_y)(u, v)^T = |(E_x, E_y)|\ |(u, v)| \cos\theta = -E_t, \tag{3.1}$$

where θ is the angle between (E_x, E_y) and (u, v). Thus, based on (3.1), the α component of the flow can be expressed as

$$\alpha = \frac{-E_t}{\sqrt{E_x^2 + E_y^2}}. \tag{3.2}$$

By using the geometry shown in Figure 1, we can also express (u, v) in terms of (α, β)

$$\begin{aligned} u &= \tfrac{E_x}{\sqrt{E_x^2+E_y^2}}\alpha - \tfrac{E_y}{\sqrt{E_x^2+E_y^2}}\beta, \\ v &= \tfrac{E_y}{\sqrt{E_x^2+E_y^2}}\alpha + \tfrac{E_x}{\sqrt{E_x^2+E_y^2}}\beta. \end{aligned} \tag{3.3}$$

It is clear that the optical flow constraint or, equivalently, (3.2) provides information for the α component, but no information for the β component. Thus, the β flow can only be determined by imposing additional constraint such as the smoothness constraint discussed in Section 4.

Even with (3.2), there exist difficulties with the computation of α. First, the computation may be not robust due to noise. Second, it is possible that $E_x^2 + E_y^2 \approx 0$ so that the computed α from (3.2) may be quite erroneous. This will be detailed in Section 3.1, Third, occluded regions exist due to moving objects. While spatio-temporal filters can be used to reduce the effect of noise and occluded regions, the resolution in the computed flow field decreases. Ideas for detecting occluded regions and the spatio-temporal filtering technique will be given in Sections 3.2 and 3.3.

3.1 α-Indeterminate Regions

For regions with a small value of $E_x^2 + E_y^2$, we call them the α-indeterminate regions. We can either interpolate the α flow from the boundaries of such regions or simply report a zero flow

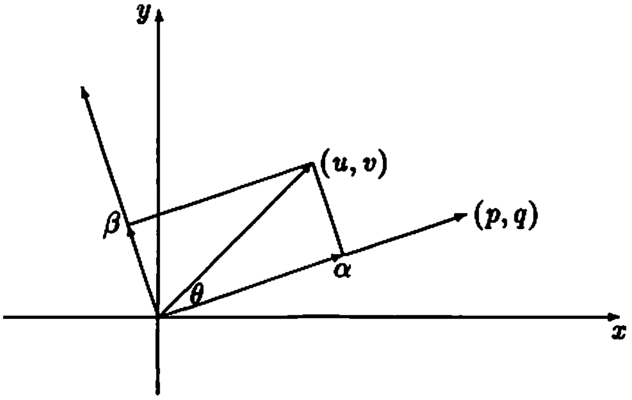

Figure 1: Normal and tangential components of the flow.

in these regions as explained below. Consider a simple example consisting of the translation of a rigid body object formed by regions (b,c,d) over a background (a) as shown in Figure 2. Let us assume that regions (a), (c), and (d) have constant gray values and region (b) has sufficient gray value variation such that the translation of this region can be determined from (3.2). Thus, regions (a), (c) and (d) are α-indeterminate regions with $E_x^2 + E_y^2 = 0$ Obviously, we should report zero flow over the background (a). If region (d) has the same gray value as the background (a), no flow should be reported in that region either. However, if (d) has a different gray value, we may or may not report a flow in that region depending on our belief of whether (d) is part of the object or part of the background. The same argument applies to region (c) except that we may still want to report a flow in that region even if (c) and (a) have the same gray values. Based on this simple example, it becomes obvious that the decision whether to interpolate from the boundaries or to set the flow equal to zero over α-indeterminate regions cannot be entirely based on local information and is not an issue that can be resolved at low-level vision.

We suggest to interpolate the flow from the boundaries of these regions, and also mark them as α-indeterminate. The decision to choose between zero and interpolated values is left to higher level of vision and is not considered in this paper. We denote the α-determinate regions as

$$\Omega_{\text{det}} = \{(x, y) \mid E_x^2 + E_y^2 > \eta\}, \tag{3.4}$$

where, η is a small positive threshold value. The α-indeterminate regions are the compliment

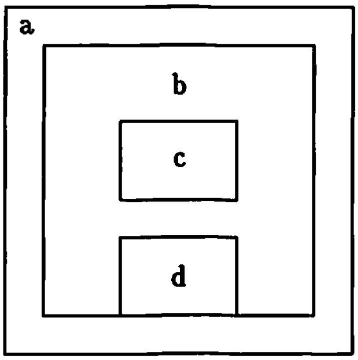

Figure 2: Flow in α-indeterminate regions.

of Ω_{det} denoted by $\overline{\Omega}_{\text{det}}$. To summarize, we have

$$\alpha(x,y) = \begin{cases} \dfrac{-E_t}{\sqrt{E_x^2 + E_y^2}} & (x,y) \in \Omega_{\text{det}}, \\ \text{Indeterminate} & (x,y) \in \overline{\Omega}_{\text{det}}. \end{cases} \tag{3.5}$$

3.2 Occluded Regions

To give an example of occluded regions, consider a simple translation of an object in the scene. There are regions in the first frame that are to be covered in the second frame, and new regions in the second frame that were previously covered in the first frame. We denote the union of *occluded regions* in both frames as Ω_{or}. Even if we knew the location of Ω_{or}, we still could not estimate α from local information since there is no corresponding region in the other frame. Thus, the value of α computed from (3.2) is invalid in Ω_{or}. Note that the value of α is not necessarily zero in occluded regions because the occluded region may not be part of the stationary background. One such example is self-occlusion where parts of the moving object occludes other moving parts of the same object occurring in, say, rotation.

It is possible to determine the occluded regions with the knowledge of exact optical flow (u,v). However, it is usually not easy to determine the exact flow (u,v) if the occluded regions are not known ahead of computation. This dilemma can be resolved by estimating the *non-occluded regions* (NOR) with the definition of optic flow (2.1) and an estimate of (u,v) as described below. Let us denote $E(x_i, y_j, t_k)$ and $E(x_i, y_j, t_k + \delta t)$ as E_1 and E_2, respectively. Without loss of generality, we choose $\delta t = 1$. Given E_2 and an estimate $(\hat{u}, \hat{v})$

of the flow, we can obtain an estimate of E_1 with (2.1), i.e.

$$\hat{E}_1(x_i, y_j) = E_2(x_i + \hat{u}, y_j + \hat{v}). \tag{3.6}$$

By comparing $\hat{E}_1(x_i, y_j)$ and $E_1(x_i, y_j)$, we have an estimate of the non-occluded regions as

$$\hat{\Omega}_{\text{nor}} = \{(x_i, y_j) \,:\, |\hat{E}_1(x_i, y_j) - E_1(x_i, y_j)| < \tau\}, \tag{3.7}$$

where τ is a small number. The estimate $\hat{\Omega}_{\text{nor}}$ can be used to improve the estimated optical flow $(\hat{u}, \hat{v})$ as detailed in Section 4.3.

It is worthwhile to mention a fine point with the computation of (3.6). Note that the range of (u, v) is the 2-D vector field of real numbers while the domain of E is a 2-D integer lattice

$$\Omega = \{(x_i, y_j) \mid i, j = 0, 1, \cdots, N-1\}.$$

To obtain an estimate of E_1 over Ω, we should look for the corresponding subpixel location in E_2 for each pixel in E_1. This means that the computations in (3.6) requires subpixel interpolation. One commonly used 2-D interpolation scheme is the bilinear interpolation.

3.3 Spatio-Temporal Filtering

In the above discussion, we considered the use of image sequence $E(x, y, t)$ to estimate first-order partial derivatives. However, it is sometimes more advantageous to use a smoothed version of the image sequence for two major reasons. First, noises and quantization errors present in an image sequence are magnified by differentiation so that numerical computation of partial derivatives obtained directly from $E(x, y, t)$ often gives a poor approximation of true partial derivatives. The quality of these approximations can be improved by smoothing the intensity $E(x, y, t)$ in both spatial and temporal domains by an isotropic Gaussian filter, i.e.

$$A(x, y, t) = G(x, y, t) * E(x, y, t), \tag{3.8}$$

where

$$G(x, y, t) = c \exp\left(-\frac{x^2 + y^2}{2\sigma_1^2} - \frac{t^2}{2\sigma_2^2}\right)$$

and where σ_1 and σ_2 are the parameters that determine the amount of smoothing in spatial and temporal directions, respectively, and c is the normalization constant. Numerical approximation of partial derivatives of $E(x, y, t)$ can be computed by, say, central differences applied to $A(x, y, t)$.

The second reason to require spatio-temporal filtering has to do with the validity of the optical flow constraint (2.2). It was shown in [22] that the optical flow constraint is valid even across spatial discontinuities for a piecewise continuos intensity process. Temporal discontinuities are more difficult to handle. For example, consider the image of a Lambertian sphere with a time-varying radius $r(t)$ over a stationary background $b(x,y)$. The image of the sphere can be written as

$$E(x,y,t) = \begin{cases} \sqrt{1 - \dfrac{x^2+y^2}{r^2(t)}} & x^2+y^2 < r^2(t), \\ b(x,y) & \text{otherwise.} \end{cases}$$

The exact value of the flow can be computed from (3.3) as

$$(u,v) = \begin{cases} \dfrac{\partial r}{\partial t}\left(\dfrac{x}{r},\dfrac{y}{r}\right) & x^2+y^2 < r^2, \\ (0,0) & \text{otherwise.} \end{cases}$$

It is obvious that if $r(t)$ has a discontinuity at t_0, the flow (u,v) at t_0 cannot be determined. For this reason, we have to avoid significant changes in the temporal domain around the same coordinate (x,y).

In case of an image sequence $E(x_i, y_j, t_k)$ for which the domain of E is only defined on a 3-D lattice, the optical flow constraint is valid if the discretized values of (E_x, E_y, E_t) approximate the differentiation of the continuos model well. Assuming that $E(x,y,t)$ has been sampled at a sufficiently high rate, we can use the Gaussian function in (3.8) for smoothing. By using the properties of convolution, we can differentiate the Gaussian function before applying the convolution to compute various partial derivatives, i.e.

$$\begin{array}{rclcl} A_x & \equiv & (G*E)_x & = & G_x * E, \\ A_y & \equiv & (G*E)_y & = & G_y * E, \\ A_t & \equiv & (G*E)_t & = & G_t * E. \end{array} \tag{3.9}$$

3.4 Flow Field Characterization

To summarize, we can classify an image into four characteristic regions:

1. regions with high gray scale variation such as intensity corners: flow is obtained from the membrane model;

2. α-determinate region: the α component of flow can be determined from (3.2);

3. α-indeterminate region: the α component of flow cannot be determined from (3.2);

4. occluded regions which have no corresponding regions in the other time frame and thus no flow information is available.

For the complete determination of flow, we need both α and β components of the flow. We have seen that the first order approximation of (2.1) by (2.2) contains no information about the β component. However, the β component may be recovered by adding appropriate smoothness constraint in some special cases. For example, in regions where the local gray value structure is sufficiently characteristic of the underlying motion, such as gray value corners and extremum, it was shown in [18] that the membrane model of Horn and Schunck [13] imposes sufficient constraint to fully determine both components of the flow. Another example was considered by Waxman and Whon [25]. They showed that it is possible to recover the β component of the flow from at least 12 local measurements of the α components, if the optical flow is caused by the motion of a planar patch. Their result can also be used to approximate the motion of a quadratic patch.

From the above discussion, we see the limitation of the optical flow constraint equation in determining the optical flow. The importance of the smoothness constraint cannot be overlooked. Most smoothness constraints derived so far do not take the underlying motion into consideration. In the following section, we will derive a smoothness model by using concepts borrowed from fluid dynamics.

4 Derivation of a Parametric Smoothness Model

4.1 Flow Model Based on Fluid Dynamics

Given a 2-D vector field $\boldsymbol{\mu}(x, y)$, we can determine its divergence and curl, respectively, by

$$\nabla\cdot\boldsymbol{\mu} = \rho, \quad \text{and} \quad \nabla\times\boldsymbol{\mu} = \omega\mathbf{i}_z,$$

where $\mathbf{i}_z = \mathbf{i}_x \times \mathbf{i}_y$ is the unit vector perpendicular to the plane. By using the terminology from fluid dynamics, the field $\boldsymbol{\mu}$ is called *solenoidal* if $\nabla\cdot\boldsymbol{\mu} = 0$ and *irrotational* if $\nabla\times\boldsymbol{\mu} = 0$. We can decompose a general velocity field $\boldsymbol{\mu}$ into the solenoidal and irrotational components, i.e.

$$\boldsymbol{\mu} = \boldsymbol{\mu}_s + \boldsymbol{\mu}_i, \tag{4.1}$$

where $\boldsymbol{\mu}_s$ and $\boldsymbol{\mu}_i$ satisfy the following constraints:

$$\nabla\cdot\boldsymbol{\mu}_s = 0, \qquad \nabla\times\boldsymbol{\mu}_s = \omega\mathbf{i}_z, \tag{4.2}$$

and

$$\nabla \cdot \boldsymbol{\mu}_i = \rho, \qquad \nabla \times \boldsymbol{\mu}_i = 0. \tag{4.3}$$

Let us assume that $\boldsymbol{\mu}_s = 0$ in (4.1) so that the flow $\boldsymbol{\mu} = (u, v)^T$ contains only the irrotational component $\boldsymbol{\mu}_i = (u_i, v_i)^T$, i.e. $u = u_i$ and $v = v_i$. Clearly, u and v are not independent variable due to the irrotational constraint. We know from (4.3) that they are related via

$$\frac{\partial v}{\partial x} = \frac{\partial u}{\partial y}.$$

We can derive an equivalent but more useful expression by introducing a scalar potential function ϕ such that

$$\frac{\partial \phi}{\partial x} = u_i, \quad \text{and} \quad \frac{\partial \phi}{\partial y} = v_i. \tag{4.4}$$

Then, the first equation in (4.3) is reduced to

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = \rho. \tag{4.5}$$

If ρ was known exactly, then we could solve (4.5) to obtain ϕ and consequently (u, v). In general, ρ is not known and therefore we must use an estimate of ρ, denoted as $\hat{\rho}$. In this case, we can solve for ϕ by minimizing the cost function

$$\mathcal{E}_i = \iint (\phi_{xx} + \phi_{yy} - \hat{\rho})^2 \, dx \, dy.$$

Now we assume that $\boldsymbol{\mu}_i = 0$ in (4.1) and the flow $\boldsymbol{\mu} = (u, v)^T$ contains only the solenoidal component $\boldsymbol{\mu}_s = (u_s, v_s)^T$, i.e. $u = u_s$ and $v = v_s$. We see from (4.2) that u and v are related via

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0,$$

known as the divergence free constraint. Due to this constraint, we can introduce a stream function ψ such that

$$-\frac{\partial \psi}{\partial y} = u_s, \quad \text{and} \quad \frac{\partial \psi}{\partial x} = v_s. \tag{4.6}$$

The second equation in (4.2) can be rewritten as

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = \omega.$$

Similar to the above discussion, we require minimization of circulation of the flow and obtain the following cost function

$$\mathcal{E}_s = \iint (\psi_{xx} + \psi_{yy} - \hat{\omega})^2 \, dx \, dy.$$

The general optical flow consists of both the irrotational and solenoidal components. Let us use the same framework and introduce the potential function ϕ and the stream function ψ. We know from (4.4) and (4.6) that the velocity field can be expressed as

$$u = u_i + u_s = \phi_x - \psi_y,$$
$$v = v_i + v_s = \phi_y + \psi_x.$$

Combing the two constraints, we get the cost functional for the general flow case, i.e.

$$\mathcal{E}_g = \iint (\phi_{xx} + \phi_{yy} - \hat{\rho})^2 \, dx \, dy + \iint (\psi_{xx} + \psi_{yy} - \hat{\omega})^2 \, dx \, dy. \tag{4.7}$$

Finally, by recognizing that

$$\phi_{xx} + \phi_{yy} = u_x + v_y, \qquad \text{and} \qquad \psi_{xx} + \psi_{yy} = v_x - u_y,$$

we can rewrite the cost function (4.7) as

$$\mathcal{E}_g = \iint (u_x + v_y - \hat{\rho})^2 \, dx \, dy + \iint (v_x - u_y - \hat{\omega})^2 \, dx \, dy. \tag{4.8}$$

Evidently, the flow computed from (4.8) depends only on the estimates of ρ and ω. It is instructive to look at the Euler equations of this energy function. The overall integral is of the form

$$\iint F(u, v, u_x, v_x, u_y, v_y) \, dx \, dy,$$

with the corresponding Euler equations

$$\begin{aligned} F_u - \frac{\partial}{\partial x} F_{u_x} - \frac{\partial}{\partial y} F_{u_y} &= 0, \\ F_v - \frac{\partial}{\partial x} F_{v_x} - \frac{\partial}{\partial y} F_{v_y} &= 0. \end{aligned}$$

We assume that $(\hat{\rho}, \hat{\omega})$ is not a function of (u, v) or its derivatives. Performing the required differentiations and simplifying, we obtain the following set of Euler equations:

$$\begin{aligned} \nabla^2 u - (\hat{\rho}_x - \hat{\omega}_y) &= 0, \\ \nabla^2 v - (\hat{\rho}_y + \hat{\omega}_x) &= 0. \end{aligned} \tag{4.9}$$

Given a pair of $(\hat{\rho}, \hat{\omega})$ and a set of boundary conditions, we can solve (4.9) for its solution denoted by $(\tilde{u}, \tilde{v})$ that minimizes the energy function (4.8). Therefore, under fixed boundary conditions, there is a one-to-one relationship between $(\hat{\rho}, \hat{\omega})$ and $(\tilde{u}, \tilde{v})$.

4.2 Equivalence to Membrane Model

Now, we are ready to incorporate the smoothness model into the optical flow constraint equation with regularization. Consider the energy function

$$\mathcal{E} = \iint (A_x u + A_y v + A_t)^2 \, dx \, dy + \lambda \iint (u_x + v_y - \hat{\rho})^2 + (v_x - u_y - \hat{\omega})^2 \, dx \, dy. \quad (4.10)$$

where the first integrand is the smoothed version of the optical flow constraint equation. The corresponding Euler equations are of the form

$$\begin{array}{rcl} \nabla^2 u - (\hat{\rho}_x - \hat{\omega}_y) & = & \frac{1}{\lambda}(A_x u + A_y v + A_t)A_x, \\ \nabla^2 v - (\hat{\rho}_y + \hat{\omega}_x) & = & \frac{1}{\lambda}(A_x u + A_y v + A_t)A_y. \end{array} \quad (4.11)$$

Comparing (4.11) to the Euler equations of the membrane model of Horn and Schunck [13], we obtain the following equivalence theorem.

Theorem 1 *If $G(x,y) = \delta(x,y)$, $\hat{\rho} = 0$ and $\hat{\omega} = 0$ then the system (4.11) of Euler equations is the same as that derived from the membrane model given in (2.3).*

This result sheds new light on the interpretation of the membrane model in terms of divergence and curl of the flow field, i.e. the membrane model is equivalent to assuming a curl-free and divergence-free flow field. With our generalized model (4.10), any prior knowledge of either curl or divergence of the flow can be incorporated through the Euler equations in (4.11).

Furthermore, by comparing (4.9) and (4.11), we see that the optical flow constraint basically provides the right-hand-side of (4.11). Even if the right-hand-side of (4.11) becomes singular, i.e. $A_x u + A_y v + A_t \approx 0$, $A_x \approx 0$, or $A_y \approx 0$, the solution to (4.11) will not be oversmoothed due to the existence of ρ and ω. Thus, the nature of motion characterized by the functions ρ and ω can play an important role in optical flow computation. Given a sequence of images, it is possible to estimate the ρ and ω by assuming that they are smooth time-varying functions. We will focus on the case consisting of only two image frames in this work, and describe an iterative procedure to estimate ρ and ω in the next section.

4.3 Estimation of ρ and ω Using Two Image Frames

Let us start with a membrane model by assuming $(\hat{\rho}, \hat{\omega}) = 0$ everywhere and obtain an initial estimate (u^0, v^0) by minimizing (4.10). From (u^0, v^0) we obtain an estimate of occluded regions as described in section 3.2. Given an estimate of the location of occluded regions with their boundaries, $\hat{\Omega}^0_{\rm or}$, and an estimate of the flow in those regions (u^0, v^0), we require

$$(\hat{\rho}^0, \hat{\omega}^0) = \begin{cases} (\mathrm{div}(u^0, v^0), \mathrm{curl}(u^0, v^0)) & (x, y) \in \hat{\Omega}^0_{\rm or}, \\ 0 & \text{otherwise.} \end{cases}$$

Then, we formulate an optimization problem for minimizing $\mathcal{E}$ in (4.10) subject to the two constraints:

$$\begin{array}{rcl} (u^1, v^1) & = & (u^0, v^0), \quad (x, y) \in \hat{\Omega}^0_{\rm or}, \\ (\hat{\rho}, \hat{\omega}) & = & (\hat{\rho}^0, \hat{\omega}^0). \end{array} \tag{4.12}$$

Although, these parameters ρ and ω are chosen to be nonzero only over $\hat{\Omega}^0_{\rm or}$, they cannot affect the solution, denoted by (u^1, v^1), over the occluded regions since the solution is fixed over these regions by the first constraint. The parameters in fact force a membrane model over non-occluded regions with the exception on their boundaries. The boundary conditions of each non-occluded region is no longer zero since $(\hat{\rho}, \hat{\omega})$ is non-zero on the boundaries of these regions. To conclude, at the first iteration, we force a membrane model subject to certain boundary conditions derived from (u^0, v^0) and, therefore, obtain a better solution than the one provided by the membrane model if a good approximation of $\Omega_{\rm or}$ and the value of flow in that region is available.

We can generalize the above procedure to obtain an iterative algorithm. In each iteration of the algorithm we use the previous estimates of the flow to determine a new estimate of $\hat{\Omega}^\ell_{\rm or}$. In practice, we have found that better results can be obtained in the subsequent iteration of the above algorithm, if we replace the constraints in (4.12) with the following set of constraints:

$$\begin{array}{rcl} (u^{\ell+1}, v^{\ell+1}) & = & (u^0, v^0) \quad (x, y) \in \hat{\Omega}^\ell_{\rm or}, \\ (\hat{\rho}, \hat{\omega}) & = & \begin{cases} (\mathrm{div}(u^0, v^0), \mathrm{curl}(u^0, v^0)) & (x, y) \in \hat{\Omega}^\ell_{\rm or}, \\ (\mathrm{div}(\tilde{u}^\ell, \tilde{v}^\ell), \mathrm{curl}(\tilde{u}^\ell, \tilde{v}^\ell)) & \text{otherwise,} \end{cases} \end{array} \tag{4.13}$$

where the superscript ℓ denotes the ℓth iteration of the algorithm and

$$\begin{array}{rcl} \tilde{u}^\ell & = & \text{local average } (u^\ell), \\ \tilde{v}^\ell & = & \text{local average } (v^\ell). \end{array}$$

The above set of constraints are the same as the constrains in (4.12) with two notable exceptions. First, the estimate of $\Omega_{\rm or}$ is obtained from the estimates of (u, v) computed in the previous cycle of the algorithm. Second, instead of requiring that the ρ and ω of the flow to be zero over non-occluded regions, we estimate them from a smoothed version of (u, v) computed in the previous iteration.

We summarize the overall algorithm in Table 1.

Table 1: Optical flow computation algorithm with two frames

```
Initialization
    • Set $(\hat{\rho},\hat{\omega}) = 0$ and obtain $(u^0, v^0)$ as the minimum point of the energy function in (4.10).
    • Compute an estimate of occluded regions $\hat{\Omega}^0_{\mathrm{or}}$ from $(u^0, v^0)$.
    • Compute $(u^1, v^1)$ as the minimum point of the energy function in (4.10) subject to the constraints in (4.12).
For $\ell = 1, 2, \cdots$ do
    • Smooth $(u^\ell, v^\ell)$ by convolving it with a Gaussian window.
    • Compute an estimate of occluded region $\Omega^\ell_{\mathrm{or}}$ from $(u^\ell, v^\ell)$.
    • Compute $(u^{\ell+1}, v^{\ell+1})$ as the minimum point of the energy function in (4.10). subject to the constraints in (4.13).
```

5 Experimental Results

We have chosen to test our algorithm on complex synthetic images (in the sense that the motion boundaries are a small subset of the intensity zero-crossings) so that the result can be compared with the ground truth. Four sets of experiments representative of various problems encountered in computation of optical flow have been performed. In all cases, the input images are 64 x 64 pixels.

- *Experiment 1*: Irrotational flow.

  A sphere (Figure 3a) of radius 20 pixels is translated perpendicular to the plane of view so that in the second frame the sphere has a radius of 21 pixels. The true optic flow (Figure 3b) has sharp discontinuities on the motion boundaries.

- *Experiment 2*: Solenoidal flow.

  A sphere (Figure 4a) is rotated 5 degrees along the axis passing through the center of the sphere and perpendicular to the plane of view. The true flow (Figure 4b) achieves its maximum on the boundaries of the sphere.

- *Experiment 3*: Pure translation.

  A sphere (Figure 5a) is translated 0.5 pixels to east and 0.5 pixels to the south. The underlying motion field (Figure 5b) is piecewise constant.

- *Experiment 4*: General motion.

  The motion of a sphere (Figure 6a) is the combination of the above tree types of motion so that the flow field has complex characteristics (Figure 6b).

In each experiment, we first pre-smooth the input images by a spatio-temporal Gaussian filter, where 3 × 3 spatial windows are used to maintain a good resolution of discontinuities in the flow field. Based on the pre-processed images, partial derivatives A_x and A_y are computed using 2-point central differences and A_t is computed by first order forward differences since we assume only two frames are available. The bilinear interpolation is used for getting the gray values at subpixel locations. In all experiments, we use regularization constant $\lambda = 1000$ and threshold value $\tau = 10$. The relaxation equations are iteratively updated until the residual drops below 10^{-5}. To evaluate the quality of the flow estimates, we use three different measures of error:

$$\begin{aligned}
\text{Mean squared error} &= \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} |u_{ij} - \hat{u}_{ij}|^2 + |v_{ij} - \hat{v}_{ij}|^2, \\
\text{Average absolute magnitude error} &= \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} \left|\sqrt{u_{ij}^2 + v_{ij}^2} - \sqrt{\hat{u}_{ij}^2 + \hat{v}_{ij}^2}\right|, \\
\text{Average absolute phase error} &= \frac{1}{n^2}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} |\cos^{-1}(\nu_{ij}^T\, \hat{\nu}_{ij})|.
\end{aligned}$$

where (u_{ij}, v_{ij}) are the true values of the flow, $(\hat{u}_{ij}, \hat{v}_{ij})$ are the estimated values of the flow, and $\nu_{ij} = (u_{ij}^2 + v_{ij}^2 + 1)^{-1/2}(u, v, 1)^T$. The phase error defined above is similar to the one used in [4] and has the desirable property of not magnifying the vector differences of large and small flow values.

Table 2 summarizes the results of the experiments. In this table, the first row of each experiment corresponds to the optical flow obtained by assuming $\hat{\rho} = 0$ and $\hat{\omega} = 0$ which we know to be equivalent to the Horn and Schunck solution. The second row in this table corresponds to the enhanced solution obtained by our algorithm after 5 iterations of updating ρ and ω (further iterations lead into only marginal improvements). The enhanced flow exhibits consistent improvement over the Horn and Schunck solution. We see about 40% improvement in all three measures of error. The third row of each experiment in the table lists the measures of error of the enhanced estimate considered only over non-occluded regions Ω_{nor} as defined in Section 3.2. The density values in the last column, represent the percentage of the pixels for which the error is computed. The estimate of non-occluded regions obtained after the 5th iteration of our algorithm is shown in Figure 7.

While the three types of error measure defined above are useful for global evaluation of the quality of the optical flow, a more detailed analysis of the flow field requires local

Table 2: Comparison of results for Experiments 1–4.

	Mean Squared Error	Average Absolute Phase Error	Average Absolute Magnitude Error	Density
		Experiment 1		
Horn & Schunck	0.0826	8.06°	0.1567	100%
Enhanced Estimate	0.0510	4.48°	0.0885	100%
Estimate over Ω_{nor}	0.0234	3.33°	0.0585	95%
		Experiment 2		
Horn & Schunck	0.0852	7.72°	0.1813	100%
Enhanced Estimate	0.0506	3.53°	0.0984	100%
Estimate over Ω_{nor}	0.0308	2.85°	0.0719	96%
		Experiment 3		
Horn & Schunck	0.0244	4.74°	.0870	100%
Enhanced Estimate	0.0139	2.63°	.0465	100%
Estimate over Ω_{nor}	0.0091	2.19°	.0366	97%
		Experiment 4		
Horn & Schunck	0.2314	11.35°	0.2624	100%
Enhanced Estimate	0.1567	6.41°	0.1599	100%
Estimate over Ω_{nor}	0.0680	4.73°	0.1048	95%

estimates of the error. Let us define the local squared error as

$$\epsilon_{ij} = \sqrt{|u_{ij} - \hat{u}_{ij}|^2 + |v_{ij} - \hat{v}_{ij}|^2}. \tag{5.1}$$

The 3-D plots in Figures 3–6 show the value of ϵ for the Horn & Schunck and enhanced estimates of the flow field in the four experiments, respectively. In all cases, we observe that maximum error occurs on the boundary of motion. The main reason for this is that the estimate of A_t is unreliable in these regions. However, we note that ϵ drops off rapidly from its maximum for the enhanced estimate of the flow. Cross-section plots of ϵ in Figures 3–6 verify this observation, where results obtained from the Horn and Schunck method and our method are denoted by dotted and dashed lines, respectively. Indeed, the main disadvantage of the membrane model is the fact that the solution is smoothly connected across the motion boundaries. The cross-section plots clearly show that our algorithm provides an effective method to overcome this problem.

6 Conclusions and Extensions

In this paper, we discussed the limitation of the optical flow constraint equation by decomposing the flow into α and β components. Motivated by concepts from fluid dynamics, we derived a parametric smoothness model by decomposing the flow into irrotational and solenoidal fields, and imposing the smoothness constraint on each field separately. This implies smooth translation and rotation of the underlying motion process. We focused on the two frame case in this research and proposed an iterative procedure to improve the estimate of parameters ρ and ω and obtain a more accurate result of the computed optical flow. An estimate of occluded regions, where the motion compensation error is likely above a given threshold, can also be obtained as a by-product of the algorithm. We showed that significant improvements can be achieved with the proposed algorithm over the classical regularization approach for a set of test problems.

The effective application of our algorithm to a sequence of images consisting of multiple frames is under our current investigation. The knowledge of $\hat{\rho}$ and $\hat{\omega}$ obtained from earlier frames should help their estimate in later frames. We are also interested in the hierarchical representation of images, where our optical flow algorithm can be applied to the low resolution image, so that multiple pixel motion can also be characterized by using the same algorithm.

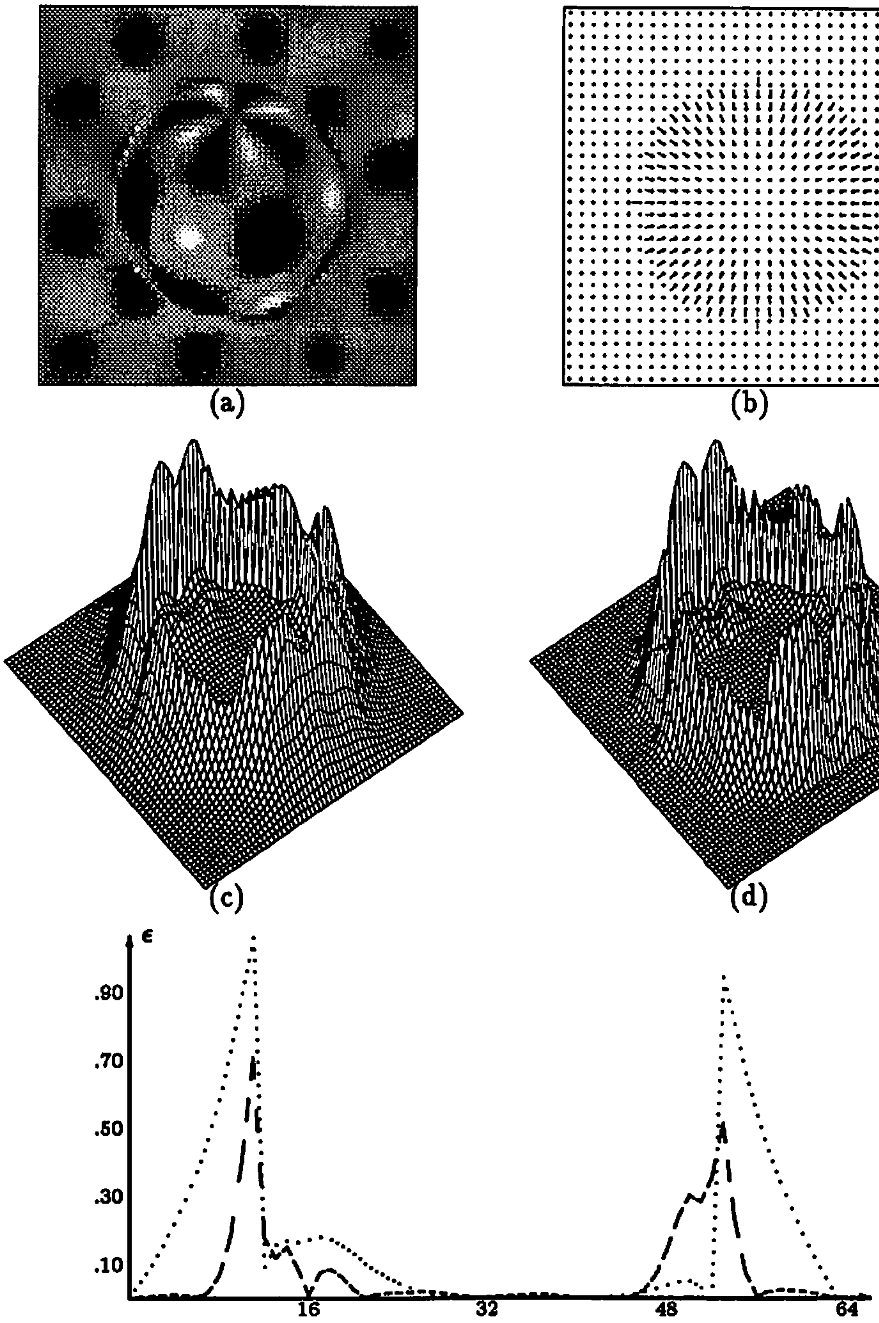

Figure 3: Experiment 1. (a) First frame of input image of a painted sphere with radius 20. (b) True motion of the sphere as it approaches the observer. (c) 3-D plot of the root squared error of the smooth solution. (d) 3-D plot of the root squared error of the enhanced solution. (e) Vertical cross-section of plots in (c) and (d)

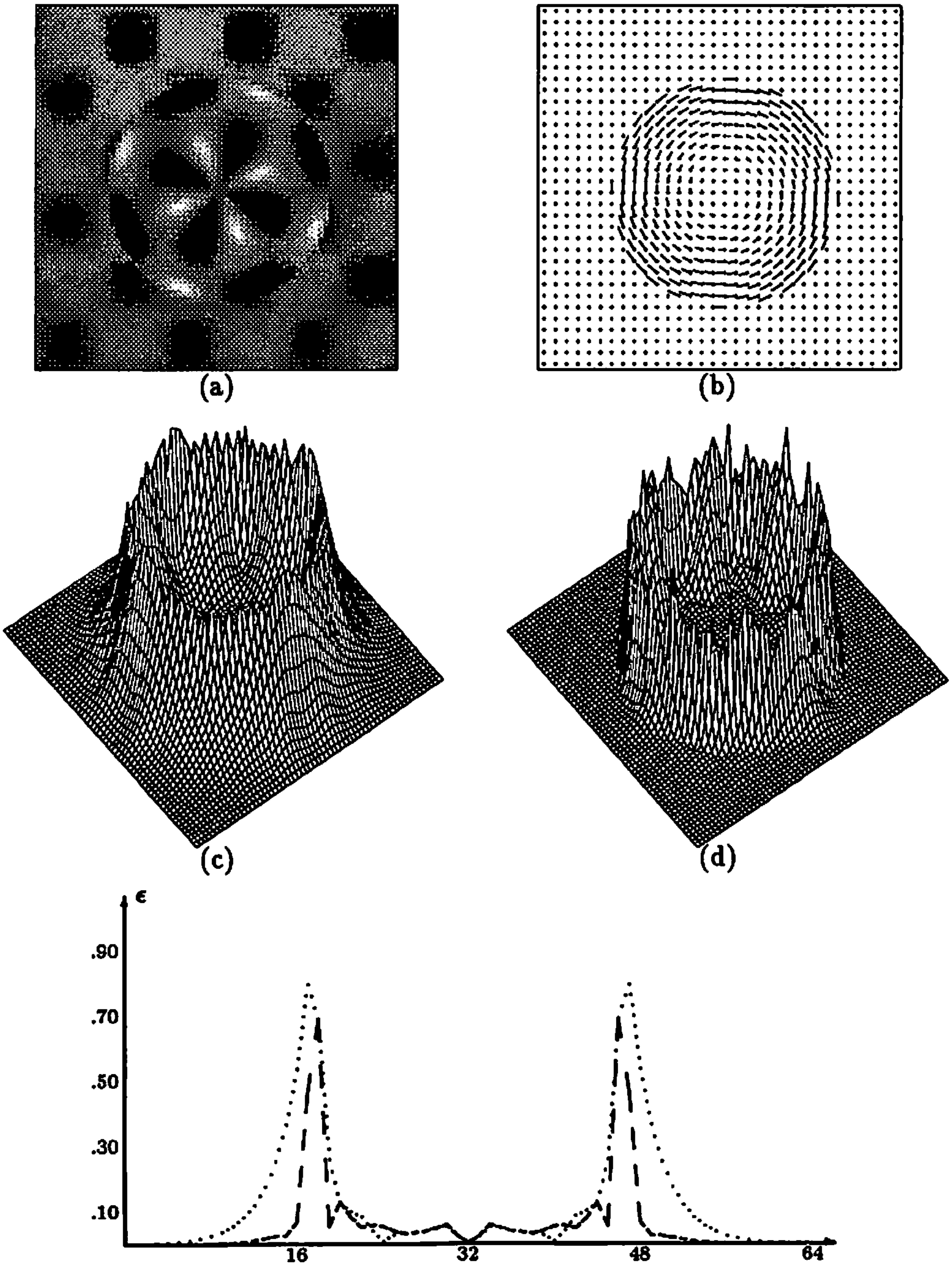

Figure 4: Experiment 2. (a) First frame of input image of a painted sphere with radius 20. (b) True motion of the sphere as it rotates 5 degrees. (c) 3-D plot of the root squared error of the smooth solution. (d) 3-D plot of the root squared error of the enhanced solution. (e) Diagonal cross-section of plots in (c) and (d).

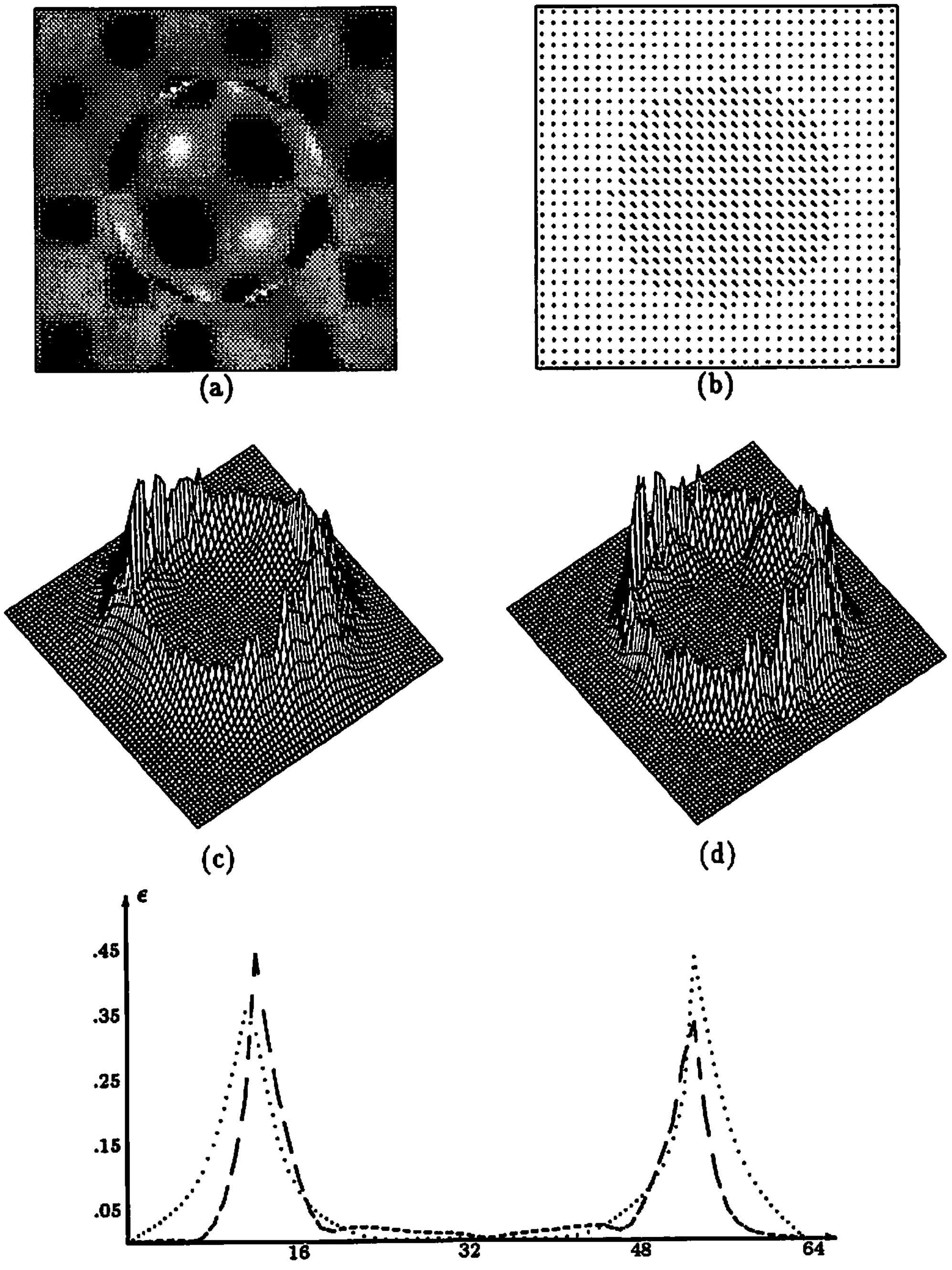

Figure 5: Experiment 3. (a) First frame of input image of a painted sphere with radius 20. (b) True motion of the sphere as it translates toward south-east direction. (c) 3-D plot of the root squared error of the smooth solution. (d) 3-D plot of the root squared error of the enhanced solution. (e) Horizontal cross-section of plots in (c) and (d).

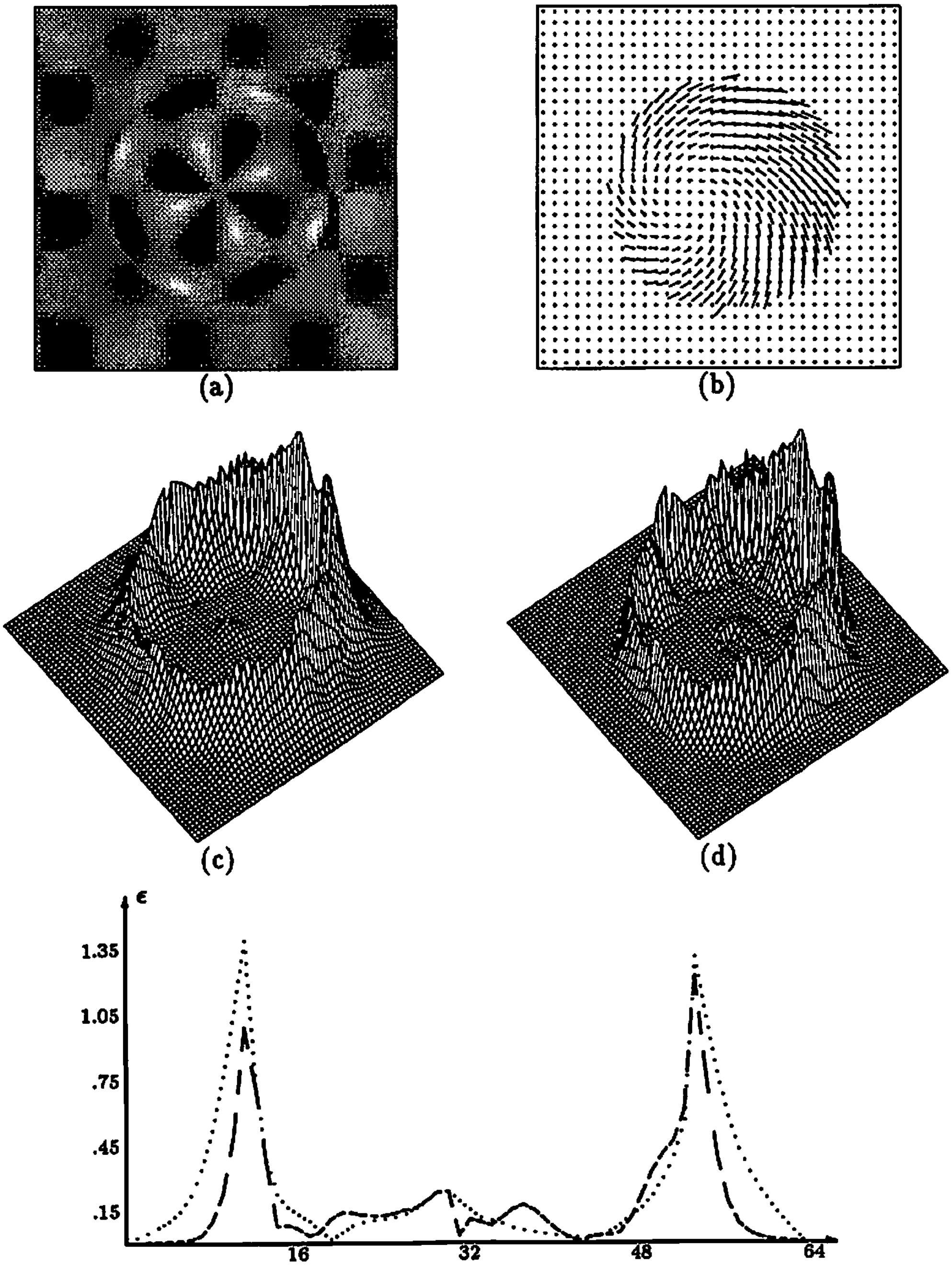

Figure 6: Experiment 4. (a) First frame of input image of a painted sphere with radius 20. (b) True motion of the sphere as it undergoes a complex motion. (c) 3-D plot of the squared error of the smooth solution. (d) 3-D plot of the squared error of the enhanced solution. (e) Vertical cross-section of plots in (c) and (d).

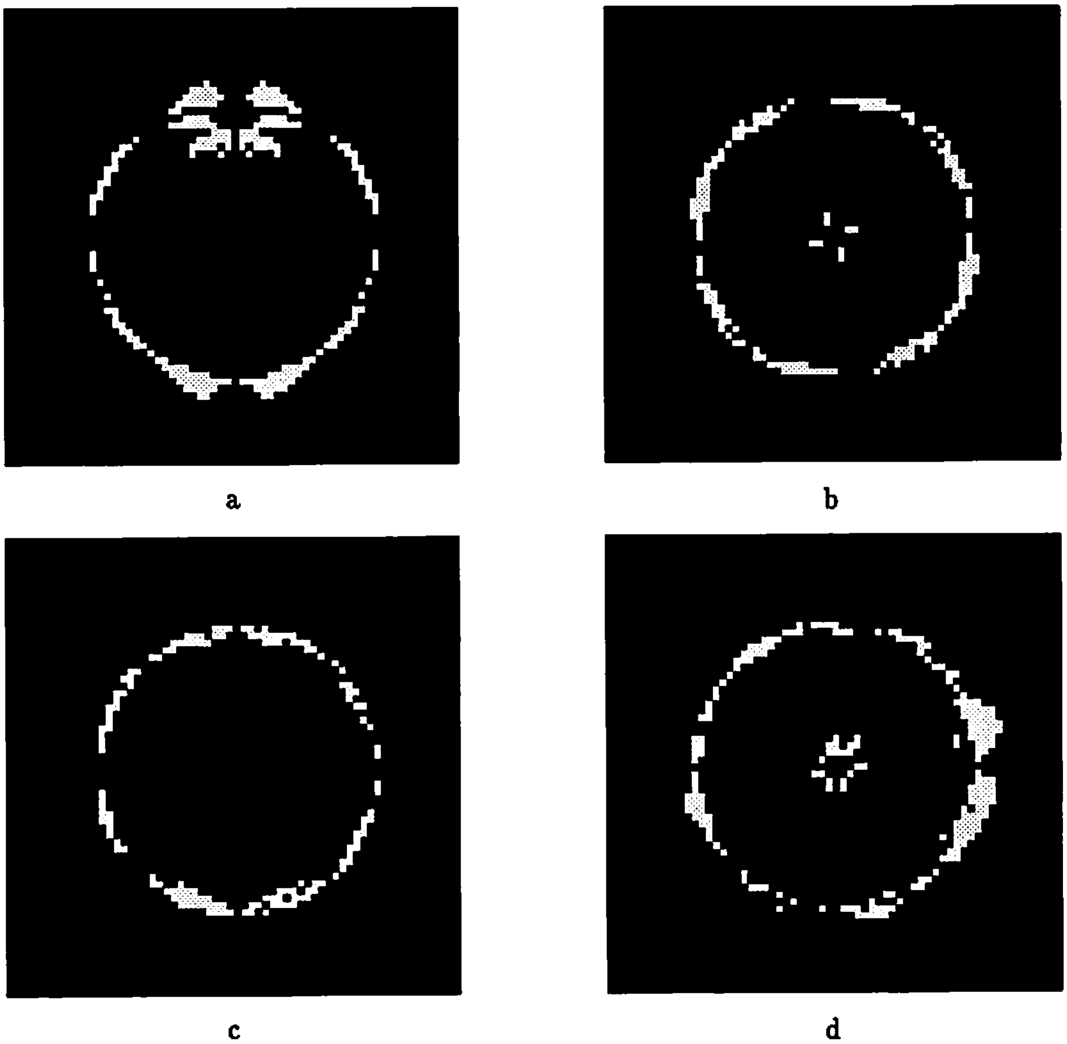

Figure 7: Estimate of occluded regions obtained after five cycles of incremental correction algorithm (a)–(d) Estimate of occluded regions of experiments 1–4, respectively.

Appendix: Numerical Implementation

In this appendix, we derive both Gauss-Seidel relaxation and the gradient projection equations required for minimizing the energy function in (4.10) subject to the constraints in (4.12) or (4.13).

A.1 Discretization

Let the region of support, Ω, be a $N \times N$ rectangular grid

$$\Omega = \{(i,j) \mid 1 \leq i \leq N, \quad 1 \leq j \leq N\}.$$

Any function, $u(x,y)$, defined over Ω is represented as a N^2 element lexicographically ordered vector of the samples of $u(x,y)$

$$\mathbf{u} = \{u(i,j) \mid (i,j) \in \Omega\}.$$

Partial derivatives can be approximated by forward differences, backward differences, or central differences:

$$\left(\frac{\partial \mathbf{u}}{\partial x}, \frac{\partial \mathbf{u}}{\partial y}\right)^T_{ij} = \underbrace{\begin{pmatrix} \mathbf{u}_{i+1j} - \mathbf{u}_{ij} \\ \mathbf{u}_{ij+1} - \mathbf{u}_{ij} \end{pmatrix}}_{\text{forward}} = \underbrace{\begin{pmatrix} \mathbf{u}_{ij} - \mathbf{u}_{i-1j} \\ \mathbf{u}_{ij} - \mathbf{u}_{ij-1} \end{pmatrix}}_{\text{backward}} = \underbrace{\frac{1}{2}\begin{pmatrix} \mathbf{u}_{i+1j} - \mathbf{u}_{i-1j} \\ \mathbf{u}_{ij+1} - \mathbf{u}_{ij-1} \end{pmatrix}}_{\text{central}}.$$

Assuming zero boundary conditions, partial derivatives can be written as matrix-vector products. For forward differences, for example, we have

$$\frac{\partial}{\partial x}\mathbf{u} = \mathbf{F}_x\,\mathbf{u}, \qquad \frac{\partial}{\partial y}\mathbf{u} = \mathbf{F}_y\,\mathbf{u};$$

where

$$\mathbf{F}_x = \begin{pmatrix} I & & & \\ -I & \ddots & & \\ & \ddots & \ddots & \\ & & -I & I \end{pmatrix} \quad \mathbf{F}_y = \begin{pmatrix} \mathbf{T}_f & & \\ & \ddots & \\ & & \mathbf{T}_f \end{pmatrix} \quad \mathbf{T}_f = \begin{pmatrix} 1 & & & \\ -1 & \ddots & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix}.$$

Similarly, matrix operators can be defined for backward and central differences, we denote these by $\mathbf{B}_x$, $\mathbf{B}_y$, $\mathbf{C}_x$, and $\mathbf{C}_y$, respectively. Using forward differences, (4.10) can be written as

$$\mathcal{E}_f = \|\mathbf{P}\mathbf{u} + \mathbf{Q}\mathbf{v} + \mathbf{t}\|^2 + \lambda\|\mathbf{F}_x\mathbf{u} + \mathbf{F}_y\mathbf{v} - \rho\|^2 + \lambda\|\mathbf{F}_x\mathbf{v} - \mathbf{F}_y\mathbf{u} - \omega\|^2,$$

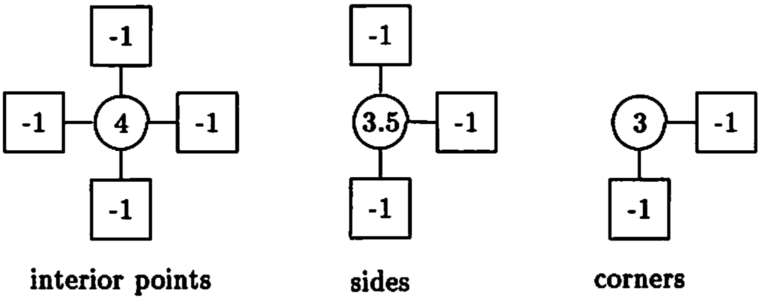

Figure 8: Stencil operators of matrix $\frac{1}{\lambda}\mathbf{H}$

where $\mathbf{P}$ and $\mathbf{Q}$ are $N^2 \times N^2$ diagonal matrices with their diagonal elements equal to samples of A_x and A_y, respectively, and $\mathbf{t}$ is a vector formed by lexicographical ordering of samples of A_t. A similar expression is also obtained for backward differences, $\mathcal{E}_b$. It turns out that by estimating $\mathcal{E}$ with the average of the forward and the backward differences, $\mathcal{E} = \frac{1}{2}(\mathcal{E}_f + \mathcal{E}_b)$, some cross terms cancel which otherwise would have made the subsequent computations unnecessarily more complex. After some algebraic manipulations, $\mathcal{E}$ can be expressed as

$$\mathcal{E} = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}^T \begin{pmatrix} \mathbf{H}+\mathbf{P}^2 & \mathbf{PQ} \\ \mathbf{PQ} & \mathbf{H}+\mathbf{Q}^2 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} + 2 \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}^T \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix} + \mathbf{t}^T\mathbf{t} + (\rho^T\rho + \omega^T\omega), \quad \text{(A.1)}$$

where

$$\begin{aligned} \mathbf{H} &= \frac{\lambda}{2}(\mathbf{F}_x^T\mathbf{F}_x + \mathbf{F}_y^T\mathbf{F}_y + \mathbf{B}_x^T\mathbf{B}_x + \mathbf{B}_y^T\mathbf{B}_y), \\ \mathbf{f} &= \mathbf{P}\mathbf{t} + \lambda(\mathbf{C}_x\rho + \mathbf{C}_y\omega), \\ \mathbf{g} &= \mathbf{Q}\mathbf{t} + \lambda(\mathbf{C}_y\rho + \mathbf{C}_x\omega). \end{aligned}$$

The iterative solution of this optimization problem requires evaluation of matrix-vector products of the form $\mathbf{Hu}$. It is convenient to express these products as a stencil of $\mathbf{H}$ operating on a 2-D array of $u(x_i, y_j)$. The stencil operators of $\mathbf{H}$ for various positions of the 2-D array are shown in figure (8).

A.2 Gauss-Seidel Relaxation Method

We divide (u, v) into two groups, (u^+, v^+) and (u^-, v^-), where the variables with a plus sign superscript are constrained. Also, we define $\mathbf{A}^+$ and $\mathbf{A}^-$ as subsets of identity matrix such that

$$\mathbf{u} = \mathbf{A}^+\mathbf{u}^+ + \mathbf{A}^-\mathbf{u}^- \qquad \text{and} \qquad \mathbf{v} = \mathbf{A}^+\mathbf{v}^+ + \mathbf{A}^-\mathbf{v}^-.$$

Substituting the above expressions for (u, v) in (A.1) and equating the partial derivatives with respect to (u^-, v^-) to zero, the following normal equations are obtained

$$\begin{pmatrix} \mathbf{A}^- & 0 \\ 0 & \mathbf{A}^- \end{pmatrix} \begin{pmatrix} \mathbf{H}+\mathbf{P}^2 & \mathbf{PQ} \\ \mathbf{PQ} & \mathbf{H}+\mathbf{Q}^2 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} = - \begin{pmatrix} \mathbf{A}^- & 0 \\ 0 & \mathbf{A}^- \end{pmatrix} \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}.$$

Using the stencil operator of $\mathbf{H}$, this can be written as

$$\begin{pmatrix} 4\lambda + p_{ij}^2 & p_{ij}q_{ij} \\ p_{ij}q_{ij} & 4\lambda + q_{ij}^2 \end{pmatrix} \begin{pmatrix} u_{ij} \\ v_{ij} \end{pmatrix} = - \begin{pmatrix} 4\lambda\bar{u} \\ 4\lambda\bar{v} \end{pmatrix}, \qquad (x_i, y_j) \notin \Omega_{\text{or}},$$

where

$$\begin{aligned} \bar{u} &= \frac{1}{4}(u_{ij+1} + u_{ij-1} + u_{i+1j} + u_{i-1j} + \frac{f_{ij}}{\lambda}), \\ \bar{v} &= \frac{1}{4}(v_{ij+1} + v_{ij-1} + v_{i+1j} + v_{i-1j} + \frac{g_{ij}}{\lambda}). \end{aligned}$$

Solving for (u_{ij}, v_{ij}), we obtain the following update equations

$$\begin{aligned} u_{ij} &\leftarrow \bar{u} - \frac{p_{ij}\bar{u} + q_{ij}\bar{v}}{4\lambda + p_{ij}^2 + q_{ij}^2} p_{ij}, \\ v_{ij} &\leftarrow \bar{v} - \frac{p_{ij}\bar{u} + q_{ij}\bar{v}}{4\lambda + p_{ij}^2 + q_{ij}^2} q_{ij}, \end{aligned} \qquad (x_i, y_j) \notin \Omega_{\text{or}}. \tag{A.2}$$

A.3 Constrained Projection Method

The update equations of the projection algorithm for problems of the form

$$\begin{array}{rl} \text{minimize} & \mathcal{E}(\mu) \\ \text{subject to} & \mathbf{D}\mu = \mu_0 \end{array}$$

is given by [16, 8]

$$\mu^{\ell+1} = \mu^\ell + \alpha\, \mathbf{c}^\ell,$$

where $\mathbf{c}$ is the projected negative gradient, $\mathbf{c}^\ell = -[\mathbf{I} - \mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{D}]\nabla\mathcal{E}(\mu^\ell)$, and α is chosen to minimize $\mathcal{E}(\mu^{\ell+1})$.

Either of the constraints in (4.12) or (4.13) can be expressed as

$$\mathbf{A}\mathbf{u} = \mathbf{u}_0, \qquad \text{or} \qquad \mathbf{A}\mathbf{v} = \mathbf{v}_0,$$

where $\mathbf{A}$ is a matrix with N^2 columns and as many rows as the number of constraint points (i.e. number of elements in Ω_{or}). Each row of $\mathbf{A}$ consists of 1's in positions corresponding to $(x_i, y_j) \in \Omega_{\mathrm{or}}$ and 0's elsewhere. In our case, $\boldsymbol{\mu} = (\mathbf{u}, \mathbf{v})^T$ and $\mathbf{D} = \mathrm{diag}\{\mathbf{A}, \mathbf{A}\}$. We note that

$$\begin{aligned}
[\mathbf{I} - \mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{D}] &= \begin{pmatrix} \mathbf{I} - \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A} & 0 \\ 0 & \mathbf{I} - \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A} \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{I} - \mathbf{A}^T\mathbf{A} & 0 \\ 0 & \mathbf{I} - \mathbf{A}^T\mathbf{A} \end{pmatrix},
\end{aligned}$$

where we have used the fact that $\mathbf{A}\mathbf{A}^T = \mathbf{I}$. Matrix $\mathbf{A}^T\mathbf{A}$ is a $N^2 \times N^2$ diagonal matrix with 0's on the diagonal except for the rows that correspond to point $(x_i, y_j) \in \Omega_{\mathrm{or}}$ which are 1. $\mathbf{I} - \mathbf{A}^T\mathbf{A}$ is simply the logical compliment of $\mathbf{A}^T\mathbf{A}$. We denote $I - \mathbf{A}^T\mathbf{A}$ by

$$\Delta = \mathrm{diag}\ \{0 \quad \text{if} \quad (x_i, y_j) \in \Omega_{\mathrm{or}};\ 1 \quad \text{if} \quad (x_i, y_j) \in \Omega_{\mathrm{nor}}\}.$$

The value of α can be found by a straightforward application of calculus and is given by

$$\alpha = \frac{\begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}^T \begin{pmatrix} \mathbf{r} \\ \mathbf{s} \end{pmatrix} + \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}^T \begin{pmatrix} \mathbf{H} + \mathbf{P}^2 & \mathbf{P}\mathbf{Q} \\ \mathbf{P}\mathbf{Q} & \mathbf{H} + \mathbf{Q} \end{pmatrix} \begin{pmatrix} \mathbf{r} \\ \mathbf{s} \end{pmatrix}}{\begin{pmatrix} \mathbf{r} \\ \mathbf{s} \end{pmatrix}^T \begin{pmatrix} \mathbf{H} + \mathbf{P}^2 & \mathbf{P}\mathbf{Q} \\ \mathbf{P}\mathbf{Q} & \mathbf{H} + \mathbf{Q} \end{pmatrix} \begin{pmatrix} \mathbf{r} \\ \mathbf{s} \end{pmatrix}},$$

where

$$\begin{aligned}
\mathbf{r} &= \Delta[(\mathbf{H} + \mathbf{P}^2)\,\mathbf{u} + \mathbf{P}\mathbf{Q}\,\mathbf{v} + \mathbf{f}], \\
\mathbf{s} &= \Delta[(\mathbf{H} + \mathbf{Q}^2)\,\mathbf{v} + \mathbf{P}\mathbf{Q}\,\mathbf{u} + \mathbf{g}].
\end{aligned}$$

Computing the gradient of $\mathcal{E}$ and simplifying the result, the update equations can be written as

$$\begin{aligned}
\mathbf{u}^{\ell+1} &= \mathbf{u}^{\ell} - \alpha\,\mathbf{r}, \\
\mathbf{v}^{\ell+1} &= \mathbf{v}^{\ell} - \alpha\,\mathbf{s}.
\end{aligned}$$

Each iteration of this algorithm requires 4 operations by stencil of $\mathbf{H}$ and about 10 evaluations of inner products.

References

[1] G. Adive, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 7, pp. 384–401, 1985.

[2] P. Anandan, *Measuring Visual Motion from Image Sequences*, Univ. of Massachusetts, Amherst: COINS Dept., 1987.

[3] S. T. Barnard and W. B. Thombson, "Disparity analysis of images," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 2, pp. 333–340, 1980.

[4] J. L. Barron, D. J. Fleet, S. S. Beauchemin, and T. A. Burkitt, "Performance of Optical Flow Techniques," in *IEEE Conf. on Computer Vision & Pattern Recognition*, pp. 236–242, 1992.

[5] B. F. Buxton and H. Buxton, "Computation of Optical Flow from Motion of Edge Features in Image Sequences," *Image and Vision Computing*, Vol. 2, pp. 59–75, 1984.

[6] N. Cornelius and T. Kanade, "Adaptive optical flow to measure object motion in reflectance and X-ray image sequence," in *ACM Siggraph/Sigart Interdisciplinary Workshop on Motion*, (Toronto, Ont.), pp. 50–58, 1983.

[7] D. Fleet and A. Jepson, "Computation of Component Image Velocity from Local Phase Information," *International Journal of Computer Vision*, Vol. 5, pp. 77–104, 1990.

[8] W. E. L. Grimson, *From Images to Surfaces*, Cambridge, Massachusetts: MIT Press, 1981.

[9] R. Haralick and J. Lee, "The facet approach to optical flow," in *Image Understanding Workshop* (L. Baumann, ed.), (Arlington, VA), pp. 84–93, Science Applications, 1983.

[10] D. J. Heeger, "Optical Flow Using Spatiotemporal Filters," *International Journal of Computer Vision*, pp. 279–302, 1988.

[11] E. Hildreth, "Computing the velocity field along the contours," in *ACM Siggrapg/Sigart Interdisciplinary Workshop on Motion: Representation and Perception*, (Tronto, Ont.), pp. 26–32, 1983.

[12] B. Horn, "Obtaining shape from shading information," in *Shape From Shading* (Horn and Brooks, eds.), Cambridge, Massachusetts: MIT Press, 1989.

[13] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, Vol. 17, pp. 185–203, 1981.

[14] J. K. Kearney, W. B. Thompson, and D. L. Boley, "Optical Flow Estimation: An Error Analysis of Gradient-Based Methods with Local Optimization," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 9, pp. 229–244, 1987.

[15] B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," in *Proc. IJCAI*, pp. 674–679, 1981.

[16] D. G. Luenberger, *Optimization by Vector Space Methods*, New York: John Wiley & Sons, Inc., 1969.

[17] H.-H. Nagel, "Displacement vectors derived from second-order intensity variations in image sequences," *Computer, Vision, Graphics, and Image Processing*, Vol. 21, pp. 85–117, 1983.

[18] H. H. Nagel, "On the Estimation of Optical Flow: Relations between Different Approaches and Some New Results," *Artificial Intelligence*, Vol. 33, pp. 299–325, 1987.

[19] A. N. Netravali and J. D. Robbins, "Motion Compensated Television Coding—Part I," *The Bell System Technical Journal*, pp. 631–670, March 1979.

[20] A. N. Netravali and J. D. Robbins, "Motion Compensated Coding: Some New Results," *The Bell System Technical Journal*, pp. 1735–1745, Nov. 1980.

[21] B. Schunck, "Image flow segmentation and estimation by constant line clustering," *IEEE trans. on Pattern Recognition and Machine Intelligence*, Vol. 11, pp. 1010–1027, 1989.

[22] B. G. Schunck, "The Image flow Constraint Equation," *Computer, Vision, Graphics, and Image Processing*, Vol. 35, pp. 20–46, 1986.

[23] A. Singh, *Optic Flow Computation: A Unified Perspective*, Los Alamitos, CA: IEEE Computer Society Press, 1991.

[24] O. Tretiak and L. Pastor, "Velocity estimation from image sequences with second order differential operators," in *Int. Conf. on Pattern Recognition*, (Montreal, Que.), pp. 16–19, 1984.

[25] A. M. Waxman and K. Wohn, "Contour evolution, neighborhood deformation and global image flow: Planar surface in motion," *Int'l J. Robotics*, Vol. 4, pp. 95–108, 1985.

[26] A. M. Waxman, J. Wu, and F. Bergholm, "Convected Activation Profiles and Measurement of Visual Motion," in *IEEE Conf. on Computer Vision & Pattern Recognition*, (Ann Arbor, Mich.), pp. 717–722, 1988.