Reliable and Efficient Computation of Optical Flow

Article *in* International Journal of Computer Vision · August 1998

DOI: 10.1023/A:1008005509994 · Source: DBLP

Shang-Hong Lai
National Tsing Hua University
291 PUBLICATIONS **4,472** CITATIONS

Baba Vemuri
University of Florida
324 PUBLICATIONS **13,626** CITATIONS

International Journal of Computer Vision 29(2), 87–105 (1998)
© 1998 Kluwer Academic Publishers. Manufactured in The Netherlands.

Reliable and Efficient Computation of Optical Flow*

SHANG-HONG LAI
Siemens Corporate Research, 755 College Road East, Princeton, NJ 08540

BABA C. VEMURI
Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611

Received June 20, 1995; Accepted October 30, 1997

Abstract. In this paper, we present two very efficient and accurate algorithms for computing optical flow. The first is a modified gradient-based regularization method, and the other is an SSD-based regularization method. For the gradient-based method, to amend the errors in the discrete image flow equation caused by numerical differentiation as well as temporal and spatial aliasing in the brightness function, we propose to selectively combine the image flow constraint and a contour-based flow constraint into the data constraint by using a reliability measure. Each data constraint is appropriately normalized to obtain an approximate minimum distance (of the data point to the linear flow equation) constraint instead of the conventional linear flow constraint. These modifications lead to robust and accurate optical flow estimation. We propose an incomplete Cholesky preconditioned conjugate gradient algorithm to solve the resulting large and sparse linear system efficiently. Our SSD-based regularization method uses a normalized SSD measure (based on a similar reasoning as in the gradient-based scheme) as the data constraint in a regularization framework. The nonlinear conjugate gradient algorithm in conjunction with an incomplete Cholesky preconditioning is developed to solve the resulting nonlinear minimization problem. Experimental results on synthetic and real image sequences for these two algorithms are given to demonstrate their performance in comparison with competing methods reported in literature.

Keywords: motion estimation, motion analysis, optical flow computation, regularization

1. Introduction

Optical flow computation is a fundamental problem in the motion analysis of image sequences. Although Verri and Poggio (1989) have proved that the optical flow is, in general, different from the motion field—which is the projections of the motions of scene points relative to the observer—nevertheless, optical flow provides very important information for estimating 3D velocity fields, analyzing rigid and nonrigid motion, segmenting the image into regions based on their motion, or recovering 3D structure of objects in the image. In recent years, many techniques for the computation of optical flow have been proposed in literature. These can be classified into gradient-based, correlation-based, energy-based, and phase-based methods (Barron et al., 1994).

The gradient-based approach is dependent on the image flow constraint equation, which is derived from the brightness constancy assumption as well as the first-order Taylor series approximation (Horn and Schunck, 1981). Using the image flow constraint equation alone is insufficient to compute the optical flow since each equation involves two different variables. Horn and Schunck (1981) introduced a first-order smoothness measure to constraint the flow field and solve for the flow. Since the smoothness constraint is invalid across the motion boundary, Nagel and Enkelmann (1986)

*This research was supported in part by the NSF grant ECS–9210648

proposed the "oriented smoothness" measure to suppress the smoothness constraint in the direction orthogonal to the boundaries. However, the smoothness across motion boundaries problem can be resolved by using the regularization with discontinuities (Terzopoulos, 1986).

The variation of an intensity function with time may be caused by geometric or photometric changes. Although we focus on the constant illumination problem in this paper, our proposed methods can be easily generalized to the gradual time-varying illumination case by using the generalized brightness change model proposed in (Negahdaripour and Yu, 1993).

The image flow constraint equation $I_x u + I_y v + I_t = 0$, where $I(x, y, t)$ is the intensity function, can be derived from the brightness constancy assumption expressed as $I(x + u\delta t, y + v\delta t, t + \delta t) = I(x, y, t)$ for a discrete time system or as $\frac{dI}{dt} = 0$ for a continuous time system. The derivation of the image flow constraint equation involves the first-order Taylor series approximation in the discrete time case. For both cases, numerical differentiation is required to approximate the partial derivatives in the image flow constraint equation. Approximation errors in the partial derivatives are inevitable due to inaccurate numerical approximation as well as the temporal and spatial aliasing in the sampling of the brightness function $I(x, y, t)$. These approximation errors are one of the main source of errors in the image flow constraint equation. Numerical differentiation techniques are usually based on the assumption that good approximation to the function in the neighborhood of the differentiation point can be achieved via a low-order polynomial function, especially a linear function when using forward/backward finite differences and a quadratic function when using a central difference formula. Therefore, large numerical approximation errors are more likely to happen at the locations of the function with large nonlinear components. In addition, locations of the brightness function with rapid changes either temporally or spatially are susceptible to significant temporal or spatial aliasing, respectively. Thus, we conclude that locations of large errors in the image flow constraints usually reside in areas with large nonlinear components or with fast temporal or spatial changes in the brightness function.

As discussed above, one of the major problems with the gradient-based approach is the unreliability of the image flow constraint equation in the areas where the local brightness function is highly nonlinear or fast changing, e.g., in the neighborhood of brightness discontinuities. Therefore, it is reasonable to say that the gradient constraint equation is reliable only when the local brightness function is well approximated by a linear function. Most of the gradient-based methods do not account for the reliability of the gradient constraint equation. However, in (Heitz and Bouthemy, 1993) a hypotheses testing technique was reported to determine whether the image flow constraint equation is valid at each site in the image plane.

Recently, Weber and Malik (1995) proposed a total least squares estimation of multiple optical flow constraints for each pixel obtained by convolving the original image sequence with a set of linear spatiotemporal filter kernels to robustly compute the optical flow. Black and Anandan (1996) presented a robust estimation framework for optical flow computation to compensate for the inaccuracies in the optical flow constraints and the imposed smoothness constraints. Based on this robust estimation framework, to deal with multiple motions, Black and Jepson (1996) modeled the optical flow using low-order polynomial parametric models allowing for local deformations. The images were, however, required to be segmented. The accuracy of this flow estimation relies on the result of image segmentation, which is required in the preprocessing stage to define regions for approximation by the parametric motion models. More recently, Ju et al. (1996) proposed a multilayer locally affine motion model and regularization with transparency based on the above robust estimation framework to better deal with the multiple motion problem. The resulting algorithm employed the Expectation Maximization (EM) method to minimize the energy function. *Note that the traditional image flow constraint, which is susceptible to errors in regions with high nonlinearity in the intensity function, has been employed in all of the above papers.*

In this paper, we propose a more appropriate usage of the data constraints for optical flow computation in the gradient-based and the sum-of-squared-differences (SSD) approaches, respectively. We focus on the single motion problems in this paper, since we use the standard regularization to impose the smoothness constraint all over the flow field. However, our proposed algorithms can be easily modified to deal with multiple motions by changing the L^2 norm in the smoothness constraint to the L^1 norm, which has been shown to allow for discontinuities in the estimated function (Kumar et al., 1996; Rudin et al., 1993).

The above mentioned image flow constraint is unreliable at the locations where the brightness function contains large nonlinear components. These locations usually occur in the neighborhood of discontinuities of the brightness function. However, the contour-based methods can give reliable estimates of the optical flow along the zero-crossing contours, which may in some cases correspond to the discontinuity locations of the image brightness function. The contour-based methods use a different flow constraint along the zero-crossing contours. This flow constraint along contours is obtained by replacing the image brightness function I by the convolution of I with the second derivative of Gaussian smoothing function. In this paper, we propose to selectively combine the image flow constraint and the contour-based flow constraint at reliable locations in a regularization framework. This leads to an accurate estimation of the optical flow.

The correlation-based approach locally finds the displacement vector (u, v) between two images I_0 and I_1 at the location (x, y) by minimizing the SSD function

$$\begin{aligned}\mathrm{SSD}(\mathbf{x}; u, v) \\ &= \sum_{j=-k}^{k}\sum_{i=-k}^{k} w(i, j)[I_0(x+i, y+j) \\ &\quad - I_1(x+i+u, y+j+v)]^2\end{aligned}$$

where $\mathbf{x} = (x, y)$ and $w(i, j)$ is the weighting function. In this SSD function, the summation is performed in a window of size $(2k+1) \times (2k+1)$ centered at (x, y). Most correlation-based methods perform an extensive search for the displacement vector (u, v) in a finite integer-pair set and find the pair with the smallest SSD value as the solution displacement. Since the search region of the displacement vector is discretized for this extensive search, the accuracy of the computed optical flow is limited by this discretization. To obtain more reliable flow estimates over the entire image domain, Anandan (1989) treated the estimates provided by the matching process as the data constraints for optical flow with appropriate confidence measure, and incorporated a smoothness constraint on the optical flow. Recently, Szeliski and Coughlan (1994) used the two-dimensional spline model to represent the flow field and minimized the following SSD function

$$E(\mathbf{u}) = \sum_{j=1}^{n}\sum_{i=1}^{n}[I_1(i+u_{ij}, j+v_{ij}) - I_0(i, j)]^2 \qquad (1)$$

where the vector $\mathbf{u}$ is the concatenation of the flow components u_{ij} and v_{ij}. A modified Levenberg-Marquardt algorithm was then employed to solve this nonconvex optimization problem. They reported very accurate results using this method. The 2D spline models for optical flow field assume the flow field to be well approximated by the 2D spline basis functions in the patches of a preset size. However, it is difficult to incorporate discontinuities into these 2D spline patches. This problem was partly solved in (Szeliski and Shum, 1995) by the quadtree spline method, which is based on a subdivision of the flow field. The result of this subdivision is, however, sensitive to a threshold parameter.

In this paper, we use the standard finite difference discretization on the flow field and take a modified SSD function as the data constraint energy in a regularization framework. We employ an incomplete Cholesky preconditioned nonlinear conjugate gradient algorithm in conjunction with a coarse-to-fine strategy to efficiently minimize the total energy function. Experimental results on the standard synthetic image sequence using our algorithm compare favorably to the best existing results reported in literature. Our algorithm computes dense optical flow estimates from (just) two consecutive frames and allows for the motion discontinuities to be incorporated.

The remainder of this paper is organized as follows. In the next section, we present our modified gradient-based method that selectively combines the region-based gradient constraint and the contour-based gradient constraint in a regularization framework. In Section 3, our SSD-based regularization technique for optical flow computation from two images of a time sequence is proposed. The experimental results for both the algorithms on synthetic and real image sequences are presented in Section 4. Finally, we conclude in Section 5.

2. Modified Gradient-Based Method

At the core of the standard gradient-based approach is the image flow constraint equation,

$$\begin{aligned}I_x(x, y, t)u(x, y, t) + I_y(x, y, t)v(x, y, t) \\ + I_t(x, y, t) = 0 \qquad (2)\end{aligned}$$

where $I(x, y, t)$ is the image brightness function at location (x, y) and time t, I_x, I_y and I_t are the partial derivatives of I with respect to x, y and t, respectively, and (u, v) is the optical flow field. The image flow constraint equation has been used in different ways to obtain the optical flow estimates as reported in the literature on gradient-based methods (Horn and Schunck, 1981; Lucas and Kanade, 1981; Nagel, 1983; Uras et al., 1988). In a regularization framework, the image flow constraint equation is regarded as the data constraint and an additional smoothness constraint is imposed on the optical flow.

The image flow equation can be derived by taking the first-order Taylor series approximation in the equation expressing the image constancy assumption namely, $I(x, y, t) = I(x + u\delta t, y + v\delta t, t + \delta t)$. The higher order terms are neglected under the assumption that the time step δt between consecutive frames is arbitrarily small, which is most often violated in practice. In reality, the time step is usually fixed, which may lead to a temporal aliasing problem. As discussed in Section 1, the image flow constraint is susceptible to errors in areas with large nonlinear components or with fast temporal or spatial changes in the brightness function possibly due to inaccurate approximation in numerical differentiation or temporal or spatial aliasing in the sampling of the brightness function. The reliability of the image flow constraint equation depends on the magnitudes of the higher order derivatives of image brightness function. If the image brightness function in the neighborhood of a point is well approximated by a linear or a quadratic function, i.e., its higher order derivatives are small, then the partial derivatives in the image flow constraint equation at this point can be accurately estimated by standard forward/backward/central difference formulas. In addition, less aliasing usually occurs in the areas where the local brightness function is well approximated linearly or quadratically. Thus, the image flow constraint is reliable in these areas. On the contrary, the image flow constraint is unreliable at the locations with significant higher order derivatives.

Since the use of an unreliable flow constraint will degrade the accuracy of the optical flow estimation, we propose to disable the unreliable image flow constraints by thresholding a reliability measure, which we will define subsequently. We will then selectively combine the image flow constraint and the contour-based flow constraint in a regularization framework. In addition, we employ an appropriate normalization procedure on the flow constraint at each point which amounts to using the shortest distance between the linear flow constraint and the point (u, v) in the space of flow vectors. To efficiently solve the resulting large and sparse linear system, we employ an incomplete Cholesky preconditioned conjugate gradient algorithm, which exploits the special structure of the stiffness matrix for the optical flow problem.

2.1. Incorporating the Contour-Based Flow Constraint

The contour-based methods (Hildreth, 1984; Waxman and Wohn, 1985; Duncan and Chou, 1992) compute the optical flow along the zero-crossing contours by using the following flow constraint equation

$$\begin{aligned} &S_x(x, y, t)u(x, y, t) + S_y(x, y, t)v(x, y, t) \\ &\quad + S_t(x, y, t) = 0 \end{aligned} \tag{3}$$

where $S(x, y, t)$ is the convolution of $I(x, y, t)$ with the second derivative of the Gaussian function, S_x, S_y, and S_t are the partial derivatives of S with respect to x, y and t, respectively. In (Hildreth, 1984), the function S was chosen to be the convolution of the Laplacian of Gaussian (LOG) filter with the image brightness function, i.e., $\nabla^2 G * I$, where $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$, $G(x, y)$ is the 2D spatial Gaussian function and the convolution is performed in 2D.

We now incorporate the above two flow constraint equations into a regularization framework. The contour-based flow constraint is applied only along the zero-crossings of $\nabla^2(G * I)$, however, its (or the image flow constraint's) applicability at any point in the image is governed by a reliability measure (to be defined later) exceeding a threshold. The variational formulation of the optical flow problem using this selective scheme for the data constraint leads to minimizing the following functional

$$\begin{aligned} &\int_\Omega \alpha((\nabla I \bullet \mathbf{u}) + I_t)^2 + \lambda\left(\|\nabla u\|_2^2 + \|\nabla v\|_2^2\right) d\mathbf{x} \\ &\quad + \oint_{\text{zero-crossing}} \beta(\nabla S \bullet \mathbf{u} + S_t)^2 \, d\mathbf{x} \end{aligned} \tag{4}$$

where $\mathbf{u}(\mathbf{x}, t) = (u(\mathbf{x}, t), v(\mathbf{x}, t))$ is the optical flow field to be estimated, $\mathbf{x} = (x, y)$ is a 2D vector, Ω is the 2D image domain, α, λ and β are the weighting functions associated to the image flow constraint, smoothness constraint and contour-based flow constraint, respectively. In this paper, the weighting function λ serves as a regularization parameter and is chosen to be a constant. The weighting functions $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$ have value 0 when the associated function does not pass the reliability test, to be defined subsequently, at the location **x** and takes a value 1 elsewhere. A discrete version of the above energy can be written as

$$\sum_{i\in D\backslash N_I}(I_{x,i}u_i + I_{y,i}v_i + I_{t,i})^2 + \sum_{i\in Z\backslash N_S}(S_{x,i}u_i + S_{y,i}v_i + S_{t,i})^2 + \lambda\sum_{i\in D}\left(u_{x,i}^2 + u_{y,i}^2 + v_{x,i}^2 + v_{y,i}^2\right). \quad (5)$$

The subscript i denotes the ith location, subscripts x, y and t denote the partial derivatives along the corresponding directions, u_i and v_i are the components of the discretized flow vector at the ith location, D contains all the discretized locations in the image domain, the set Z contains the discretized locations along the zero-crossing contours, $N_I \subset D$ is the set of locations where the reliability measures for the function I exceed a threshold, and $N_S \subset Z$ is the set of locations with the reliability measures for the function S exceeds a threshold.

2.2. Rejecting Unreliable Data Constraints

The image flow data constraint and the contour-based flow constraint are obtained by using the first-order Taylor series approximation of the functions I and S, respectively, thus each constraint is unreliable in the regions where the underlying function (I or S) is not well approximated by a linear function locally, which means the function contains significant nonlinear terms. In Fig. 1, the image flow constraint errors for the translating square and translating tree sequences at a particular frame are shown. Note that the errors in the flow constraints are computed using the correct flow vectors supplied for synthesizing this sequence. In this paper, we define a function $\delta(E, x, y, t)$ to be a measure of reliability for the linear flow constraint E (=I or S) at

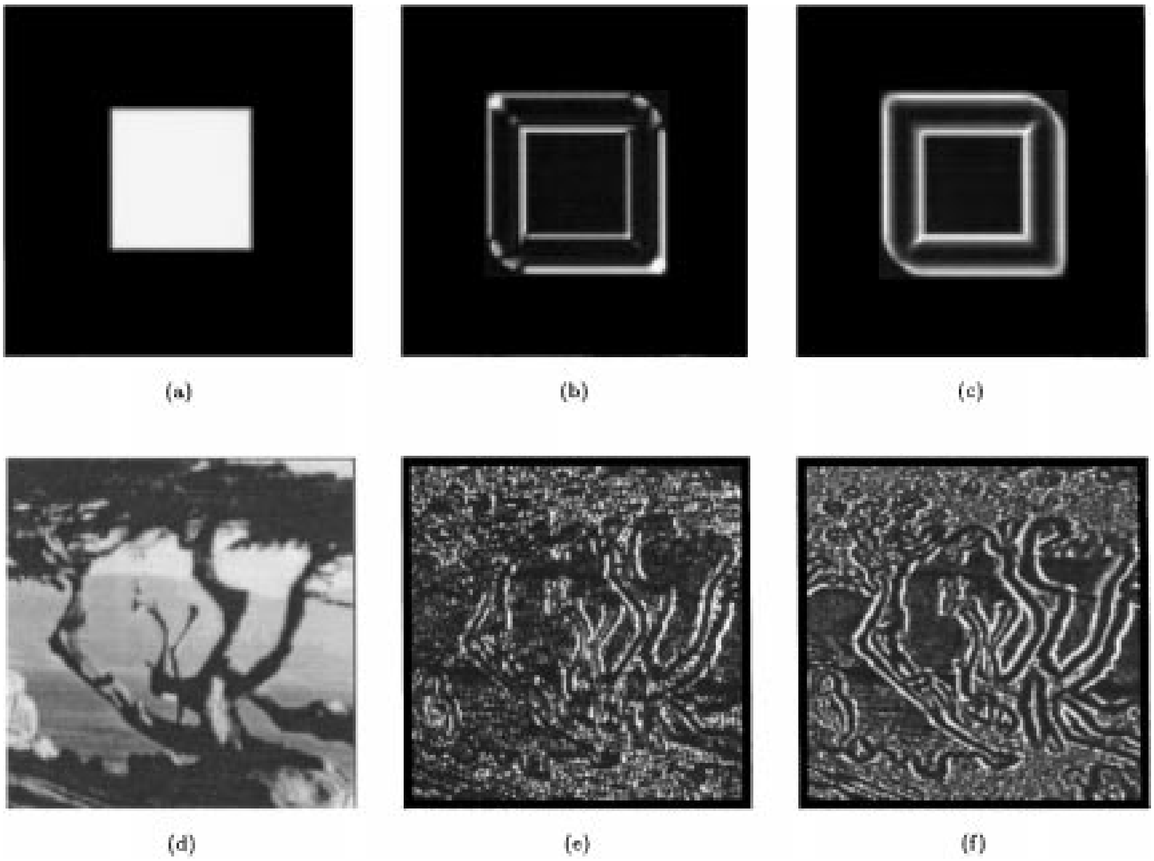


Figure 1. One frame of the (a) *Square* 2 and (d) *Translating Tree* sequence, the computed image flow constraint error maps using the correct flow vectors are shown in (b) and (e), and the computed reliability measure maps are depicted in (c) and (f), respectively.

the location (x, y) and time t as follows

$$\delta(E, x, y, t) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} \sum_{k=-1}^{1} \frac{(E(x+i, y+j, t+k) - E_0 - E_x i - E_y j - E_t k)^2}{E_x^2 + E_y^2 + E_t^2 + \epsilon}, \tag{6}$$

where E_0 is the mean of the function E in the small neighborhood of (x, y, t), E_x, E_y, and E_t are the partial derivatives of E at (x, y, t) along x, y, and t directions, respectively, and ϵ is a small positive constant used to avoid devision by 0. This constant ϵ was set to 1 in our implementation. The partial derivatives of the function E in Eq. (6) can be estimated by standard finite difference methods. The reliability measure $\delta(E, x, y, t)$ is the sum of the squared linear approximation errors of the function E at the point (x, y, t) normalized by the sum of squared partial derivatives, i.e., $E_x^2 + E_y^2 + E_t^2$. This normalization is done to obtain the minimum distance errors instead of the absolute errors. This measure indicates whether or not a first-order approximation suffices. Therefore, the corresponding data constraint is reliable when the measure δ is small and unreliable when δ is large. We can set a threshold to reject the data constraints with a high δ function value. This threshold was set to 0.5 in our implementation. The reliability measure maps for the translating square and the translating tree examples are depicted in Figs. 1(c) and (f), respectively. Note the correspondences between the flow constraint error maps and the corresponding reliability measure maps from this figure.

2.3. Normalizing the Data Constraint

The image flow or the contour flow data constraint in the above energy function implicitly encodes a weighting $\sqrt{I_{x,i}^2 + I_{y,i}^2}$ and $\sqrt{S_{x,i}^2 + S_{y,i}^2}$, respectively, at each site i. This causes high-gradient locations and locations in slowly varying regions to be weighted very differently, leading to the flow estimation mainly dominated by the data constraints at the high-gradient locations. Unfortunately, high-gradient locations can usually be caused by brightness discontinuities, noise, etc., and the corresponding flow constraints are unreliable due to the fact that the image function at these locations usually contains large nonlinear components and can have a large temporal or spatial aliasing problem, as was discussed earlier.

To amend this inappropriate weighting on the data constraints used in the traditional gradient-based approach, we use the minimum distance between the flow vector and the linear flow constraint as the new data constraint. Since the minimum distance between a point $(\hat{u}, \hat{v})$ and the linear constraint $E_x u + E_y v + E_t = 0$ is $\frac{|E_x \hat{u} + E_y \hat{v} + E_t|}{\sqrt{E_x^2 + E_y^2}}$, we can divide each original data constraint by the normalization factor $\sqrt{I_{x,i}^2 + I_{y,i}^2}$ or $\sqrt{S_{x,i}^2 + S_{y,i}^2}$ (depending on whether it is an image flow or an contour flow constraint) to obtain the new data constraint. However, errors in the flow constraints with very small gradients will be greatly amplified after this normalization process. To avoid this over-amplification problem and the problem with division by zero, the normalization factors are modified to $\sqrt{I_{x,i}^2 + I_{y,i}^2 + c}$ and $\sqrt{S_{x,i}^2 + S_{y,i}^2 + c}$ respectively, where c is a positive constant. Note that the optimal choice of the constant c should be related to the variance of the errors in the flow constraints. We empirically set it to a value 10 in all of our experiments. Let's denote the *normalized flow constraints* by $\bar{I}_{x,i} u_i + \bar{I}_{y,i} v_i + \bar{I}_{t,i} = 0$ and $\bar{S}_{x,i} u_i + \bar{S}_{y,i} v_i + \bar{S}_{t,i} = 0$, then the energy function to be minimized becomes

$$\begin{aligned} f(\mathbf{u}) = & \sum_{i \in D \setminus N_I} (\bar{I}_{x,i} u_i + \bar{I}_{y,i} v_i + \bar{I}_{t,i})^2 \\ & + \sum_{i \in Z \setminus N_S} (\bar{S}_{x,i} u_i + \bar{S}_{y,i} v_i + \bar{S}_{t,i})^2 \\ & + \lambda \sum_{i \in D} \left(u_{x,i}^2 + u_{y,i}^2 + v_{x,i}^2 + v_{y,i}^2\right) \end{aligned} \tag{7}$$

where the vector $\mathbf{u}$ is the concatenation of all the components u_i and v_i.

The above weighting on the flow vector to achieve minimum distance error for each flow constraint in the energy function is $\frac{E_x^2 + E_y^2}{E_x^2 + E_y^2 + c}$. The relationship between the original quadratic weighting factor without normalization for each flow constraint in the energy function is shown in Fig. 2(a), while the normalized weighting function is plotted in Fig. 2(b). The normalized energy function is still convex and quadratic, which can be solved very efficiently via the incomplete Cholesky preconditioned conjugate gradient algorithm to be presented subsequently.

2.4. Incomplete Cholesky Preconditioned Conjugate Gradient

The minimization of the above energy leads to solving a linear system of equations $\mathbf{Ku} = \mathbf{b}$ where the stiffness

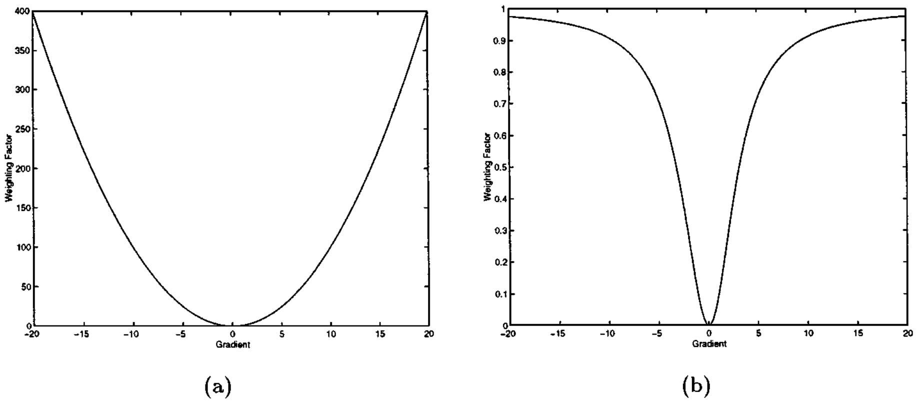


Figure 2. The weighting factor of each flow constraint in the energy function using (a) original quadratic weighting and (b) the normalized weighting.

matrix $\mathbf{K} \in \Re^{2n^2 \times 2n^2}$ is symmetric positive-definite and it has the following 2×2 block structure.

$$\mathbf{K} = \begin{bmatrix} \lambda \mathbf{K}_s + \bar{\mathbf{E}}_{xx} & \bar{\mathbf{E}}_{xy} \\ \bar{\mathbf{E}}_{xy} & \lambda \mathbf{K}_s + \bar{\mathbf{E}}_{yy} \end{bmatrix}, \qquad (8)$$

where $\mathbf{K}_s \in \Re^{n^2 \times n^2}$ is the discrete 2D Laplacian matrix from the membrane smoothness constraint, $\bar{\mathbf{E}}_{xx}$, $\bar{\mathbf{E}}_{xy}$, and $\bar{\mathbf{E}}_{yy}$ are all $n^2 \times n^2$ diagonal matrices with entries $\bar{I}_{x,i}^2 + \bar{S}_{x,i}^2$, $\bar{I}_{x,i}\bar{I}_{y,i} + \bar{S}_{x,i}\bar{S}_{y,i}$ and $\bar{I}_{y,i}^2 + \bar{S}_{y,i}^2$, respectively. Note that the values of $\bar{I}_{x,i}$ and $\bar{I}_{y,i}$ are set to zeros when the image flow constraint is disabled at the ith location due to reliability measure test. Similarly, the values of $\bar{S}_{x,i}$ and $\bar{S}_{y,i}$ are set to zeros when the contour flow constraint is not used at the ith location.

To solve this linear system for optical flow estimation, we use the preconditioned conjugate gradient algorithm (Golub and Van Loan, 1989) with an incomplete Cholesky preconditioner $\mathbf{P}$ (Meijerink and van der Vorst, 1977; Elman, 1986), given in the following.

1. Initialize $\mathbf{u}_0$; compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{K}\mathbf{u}_0$; $k = 0$.
2. Solve $\mathbf{P}\mathbf{z}_k = \mathbf{r}_k$; $k = k + 1$.
3. If $k = 1$, $\mathbf{p}_1 = \mathbf{z}_0$; else compute $\beta_k^D = \mathbf{r}_{k-2}^T \mathbf{z}_{k-2}$, $\beta_k = \frac{\alpha_{k-1}^N}{\beta_k^D}$, and update $\mathbf{p}_k = \mathbf{z}_{k-1} + \beta_k \mathbf{p}_{k-1}$.
4. Compute $\alpha_k^N = \mathbf{r}_{k-1}^T \mathbf{z}_{k-1}$, $\alpha_k^D = \mathbf{p}_k^T \mathbf{K}\mathbf{p}_k$, and $\alpha_k = \alpha_k^N / \alpha_k^D$;
5. Update $\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k \mathbf{K}\mathbf{p}_k$, $\mathbf{u}_k = \mathbf{u}_{k-1} + \alpha_k \mathbf{p}_k$.
6. If $\mathbf{r}_k \simeq \mathbf{0}$, stop; else go to step 2.

The preconditioner $\mathbf{P}$ is chosen as the incomplete Cholesky factorization of the matrix $\mathbf{K}$. A good preconditioner can drastically accelerate the convergence rate of the conjugate gradient algorithm. There are two criteria for designing a good preconditioner $\mathbf{P}$ for the matrix $\mathbf{K}$. First, the preconditioner $\mathbf{P}$ has to be a good approximation to $\mathbf{K}$ so that the condition number of the preconditioned linear system is dramatically reduced. Secondly, there must exist a very fast numerical method to solve the auxiliary linear system $\mathbf{P}\mathbf{z} = \mathbf{r}$ required in the preconditioned CG algorithm. Taking these two criteria into consideration, a good preconditioner for the above stiffness matrix $\mathbf{K}$ of the optical flow problem can be obtained via the incomplete Cholesky factorization of $\mathbf{K}$.

The standard Cholesky factorization of the sparse matrix $\mathbf{K}$ “fills in” entries in the band between nonzero off-diagonals, which means the sparsity structure will be destroyed after the factorization. The idea of incomplete Cholesky factorization is to find an approximate Cholesky factorization of the matrix $\mathbf{K}$, i.e., $\mathbf{K} \approx \mathbf{L}\mathbf{L}^T$, such that the lower triangular matrix $\mathbf{L}$ has the similar sparsity structure. In addition, the product $\mathbf{L}\mathbf{L}^T$ at the locations with nonzero entries in $\mathbf{L}$ or $\mathbf{L}^T$ still has the same values as those in $\mathbf{K}$. Therefore, the preconditioner $\mathbf{P} = \mathbf{L}\mathbf{L}^T$ is a good approximation to $\mathbf{K}$. Since the matrix $\mathbf{K}$ is sparse and well structured, the matrix $\mathbf{L}$ is also sparse and well structured. Thus, the solution to the auxiliary linear system $\mathbf{P}\mathbf{z} = \mathbf{r}$ in the preconditioning step of the preconditioned conjugate gradient algorithm can be obtained via forward and backward substitutions very efficiently. Furthermore, we use the block version (Golub and Van Loan, 1989) of the incomplete Cholesky factorization to take advantage of the 2×2 block structure of the matrix $\mathbf{K}$.

The matrix $\mathbf{K}$ in Eq. (8) has diagonal blocks $\lambda \mathbf{K}_s + \bar{\mathbf{E}}_{xx}$ and $\lambda \mathbf{K}_s + \bar{\mathbf{E}}_{yy}$, each of which is block tridiagonal. The off-diagonal elements of $\mathbf{K}$ are diagonal blocks, each of which is a diagonal matrix. In addition, the off-diagonal elements of the block tridiagonal matrices $\lambda \mathbf{K}_s + \bar{\mathbf{E}}_{xx}$ and $\lambda \mathbf{K}_s + \bar{\mathbf{E}}_{yy}$ are diagonal. The factoring matrix in the incomplete Cholesky factorization has the similar sparse structure as that of $\mathbf{K}$ and takes the following form:

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22}, \end{bmatrix} \tag{9}$$

where the submatrices $\mathbf{L}_{11}$, $\mathbf{L}_{21}$ and $\mathbf{L}_{22}$ are of size $n^2 \times n^2$. Our incomplete Cholesky factorization is based on the construction of an incomplete block preconditioner in (Golub and Van Loan, 1989). The submatrices $\mathbf{L}_{11}$, $\mathbf{L}_{22}$ and $\mathbf{L}_{21}^T$ are chosen to be block lower bidiagonal matrices with the diagonal blocks being lower bidiagonal and the lower diagonal blocks being upper bidiagonal, i.e., the matrices $\mathbf{L}_{11}$, $\mathbf{L}_{22}$ and $\mathbf{L}_{21}$ have the following structures,

$$\mathbf{L}_{ij} = \begin{bmatrix} \mathbf{G}_{ij}^{(1)} & & & \\ \mathbf{H}_{ij}^{(1)} & \mathbf{G}_{ij}^{(2)} & & \\ & \ddots & \ddots & \\ & & \mathbf{H}_{ij}^{(n-1)} & \mathbf{G}_{ij}^{(n)} \end{bmatrix}, \tag{10}$$

for $(i, j) \in \{(1, 1), (2, 2)\}$, and

$$\mathbf{L}_{21} = \begin{bmatrix} \mathbf{G}_{21}^{(1)^T} & \mathbf{H}_{21}^{(1)^T} & & \\ & \mathbf{G}_{21}^{(2)^T} & \ddots & \\ & & \ddots & \mathbf{H}_{21}^{(n-1)^T} \\ & & & \mathbf{G}_{21}^{(n)^T} \end{bmatrix}, \tag{11}$$

where

$$\mathbf{G}_{ij}^{(k)} = \begin{bmatrix} \alpha_{ij,1}^{(k)} & & & \\ \beta_{ij,1}^{(k)} & \alpha_{ij,2}^{(k)} & & \\ & \ddots & \ddots & \\ & & \beta_{ij,n-1}^{(k)} & \alpha_{ij,n}^{(k)} \end{bmatrix} \in \Re^{n \times n},$$

$$\mathbf{H}_{ij}^{(k)} = \begin{bmatrix} \gamma_{ij,1}^{(k)} & \delta_{ij,1}^{(k)} & & \\ & \gamma_{ij,2}^{(k)} & \ddots & \\ & & \ddots & \delta_{ij,n-1}^{(k)} \\ & & & \gamma_{ij,n}^{(k)} \end{bmatrix} \in \Re^{n \times n},$$

for $k = 1, \ldots, n$. The nonzero entries in the sparse matrix $\mathbf{L}$ are computed by equating the entries of the product $\mathbf{L}\mathbf{L}^T$ to those in the matrix $\mathbf{K}$ at the locations with nonzero entries in $\mathbf{L}$. Thus, the nonzero entries in $\mathbf{L}_{11}$ are given by

$$\alpha_{11,l}^{(k)} = \begin{cases} \sqrt{\mathbf{K}_{p,p} - \left(\beta_{11,l-1}^{(k)}\right)^2}, \quad k = 1, \ 1 \le l \le n, \\ \sqrt{\mathbf{K}_{p,p} - \left(\beta_{11,l-1}^{(k)}\right)^2 - \left(\delta_{11,l}^{(k-1)}\right)^2 - \left(\gamma_{11,l}^{(k-1)}\right)^2}, \\ \qquad 2 \le k \le n, \ 1 \le l \le n, \end{cases} \tag{12}$$

$$\beta_{11,l}^{(k)} = \begin{cases} 0, \quad l = 0, n, \ 1 \le k \le n, \\ \mathbf{K}_{p,p-1} / \alpha_{11,l}^{(k)}, \quad k = 1, 1 \le l \le n-1, \\ \left(\mathbf{K}_{p,p-1} - \gamma_{11,l}^{(k-1)} \delta_{11,l-1}^{(k-1)}\right) / \alpha_{11,l}^{(k)}, \\ \quad 2 \le k \le n, \ 1 \le l \le n-1, \end{cases} \tag{13}$$

$$\gamma_{11,l}^{(k)} = \mathbf{K}_{p,p-n} / \alpha_{11,l}^{(k)}, \quad 1 \le k \le n-1, \ 1 \le l \le n, \tag{14}$$

$$\delta_{11,l}^{(k)} = \begin{cases} 0, \quad l = 0, n, 1 \le k \le n-1, \\ -\beta_{11,l}^{(k)} \gamma_{11,l}^{(k)} / \alpha_{11,l+1}^{(k)}, \\ \quad 1 \le l \le n-1, 1 \le k \le n-1, \end{cases} \tag{15}$$

where $p = (k-1)n + l$. Similarly, the nonzero entries in $\mathbf{L}_{21}$ can be written as follows:

$$\alpha_{21,l}^{(k)} = \bar{E}_{xy,p} / \alpha_{11,l}^{(k)}, \quad 1 \le k, l \le n, \tag{16}$$

$$\beta_{21,l}^{(k)} = \begin{cases} -\alpha_{21,l}^{(k)} \beta_{11,l}^{(k)} / \alpha_{11,l+1}^{(k)}, \\ \qquad 1 \le k \le n, \ 1 \le l \le n-1, \\ 0, \quad 1 \le k \le n, \ l = n, \end{cases} \tag{17}$$

$$\delta_{21,l}^{(k)} = -\alpha_{21,l+1}^{(k)} \delta_{11,l}^{(k)} / \alpha_{11,l}^{(k+1)}, \quad 1 \le k, l \le n-1, \tag{18}$$

$$\begin{aligned} \gamma_{21,l}^{(k)} = &- \left(\alpha_{21,l}^{(k)} \gamma_{11,l}^{(k)} + \beta_{21,l}^{(k)} \delta_{11,l}^{(k)} \right. \\ &\left. + \delta_{21,l-1}^{(k)} \beta_{11,l-1}^{(k+1)}\right) / \alpha_{11,l}^{(k+1)} \\ &\qquad 1 \le k \le n-1, \ 1 \le l \le n. \end{aligned} \tag{19}$$

The nonzero entries in the submatrix $\mathbf{L}_{22}$ have the similar forms to those in $\mathbf{L}_{11}$, given by Eqs. (12)–(15), except for the matrix $\mathbf{K}$ which is replaced by $\lambda \mathbf{K}_s +$

$\bar{\mathbf{E}}_{yy} - \mathbf{L}_{21}\mathbf{L}_{21}^T$ in the equations for $\alpha_{22,l}^{(k)}$, $\beta_{22,l}^{(k)}$, $\delta_{22,l}^{(k)}$ and $\gamma_{22,l}^{(k)}$.

The incomplete Cholesky factorization of the matrix $\mathbf{K}$ given above can be computed in $O(N)$ operations, where $N(=n^2)$ is the number of discretization points. After the factorization, the preconditioner $\mathbf{P}$ is chosen as $\mathbf{L}\mathbf{L}^T$, which is close to $\mathbf{K}$ and has a nice structure. Thus, the preconditioned conjugate gradient algorithm requires $O(N)$ operations in each iteration.

3. SSD-Based Regularization Method

The SSD function in Eq. (1) can be used as the data constraint energy in a regularization framework. However, we modify the data constraint by using the normalized difference instead of the absolute difference in the traditional SSD approach. For a data constraint $I_1(x_i + u_i, y_i + v_i) - I_0(x_i, y_i)$ at the ith discretized location (x_i, y_i) with a displacement vector given by (u_i, v_i), we normalize it by a mean slope factor $\sqrt{I_{x,i}^2+I_{y,i}^2+c}$, where $I_{x,i} = \frac{1}{2}(\frac{\partial I_1(x_i+u_i,y_i+v_i)}{\partial x} + \frac{\partial I_0(x_i,y_i)}{\partial x})$, $I_{y,i} = \frac{1}{2}(\frac{\partial I_1(x_i+u_i,y_i+v_i)}{\partial y} + \frac{\partial I_0(x_i,y_i)}{\partial y})$, and c is a constant used to avoid the error amplification at the locations with very small gradients. This normalization is used to approximate the minimum distance between the point (u_i, v_i) and the constraint surface $I_1(x_i + u_i, y_i + v_i) - I_0(x_i, y_i) = 0$ by the normalized distance. The normalization used here is similar to that used in our modified gradient-based algorithm described in Section 2. This leads to a weighted SSD measure with the weight for each data constraint determined by the above normalization factor.

Incorporating the normalized SSD constraints into the regularization formulation, we obtain the total energy function

$$f(\mathbf{u}) = \sum_{i \in D} \frac{(I_1(x_i + u_i, y_i + v_i) - I_0(x_i, y_i))^2}{I_{x,i}^2 + I_{y,i}^2 + c} + \lambda\left(u_{x,i}^2 + u_{y,i}^2 + v_{x,i}^2 + v_{y,i}^2\right), \tag{20}$$

where D is the set of all points in the image domain. Note that the data constraint, $I_1(x_i + u_i, y_i + v_i) = I_0(x_i, y_i)$, is not linear in the flow vector u_i or v_i. Therefore, the energy function given in Eq. (20) is not a quadratic in the flow variable u_i or v_i. If the Taylor expansion of the data constraint is taken to first order and I_t is approximated by the difference between I_1 and I_0 at the same location, then we obtain the image flow constraint in Eq. (2). *Consequently, the gradient-based regularization approach can be perceived as a first-order approximation to the SSD-based regularization formulation.* The energy function to be minimized in the gradient-based regularization is quadratic and convex, while the energy function in Eq. (20) is neither quadratic nor convex. For the quadratic and convex energy function in the gradient-based regularization, the energy minimization can be accomplished by solving a symmetric positive definite (SPD) linear system. The minimization of the nonquadratic and nonconvex energy function is more difficult and requires more computational effort. The SSD-based regularization usually can give a more accurate estimation than the gradient-based regularization, since the data constraint used in the SSD-based regularization is more precise than the one in the gradient-based regularization. In addition, the SSD-based regularization method requires only two frames to compute the optical flow, while the gradient-based methods usually use more than two frames for the optical flow computation. However, the SSD-based regularization method requires more computational effort to minimize the energy function.

The energy function in Eq. (20) is a nonlinear function with a large number of variables. We use the nonlinear conjugate gradient method (Gill et al., 1981) with the incomplete Cholesky preconditioning to efficiently minimize this nonlinear energy function. The nonlinear conjugate gradient algorithm is quite efficient in the unconstrained optimization of large-scale nonlinear functions (Gill et al., 1981). Although this algorithm can only find the local minima, we can obtain very accurate solution when the initial solution is close to the true solution. For example, small displacement between consecutive frames can be easily found when we start the nonlinear CG from zero initial values. In fact, a very good initial solution can be obtained from the optical flow estimated in the previous frames. In addition, a coarse-to-fine strategy can be employed to account for large-displacement problems and to accelerate the convergence rate of the nonlinear CG algorithm.

The incomplete Cholesky preconditioning was employed in our modified gradient-based method described in Section 2 to efficiently solve the large and sparse linear system with the specially structured stiffness matrix for the optical flow problem. This preconditioner is very efficient since the incomplete Cholesky decomposition was performed to approximate the large and sparse stiffness matrix by exploiting its special structure. We can apply the same preconditioning technique in the nonlinear conjugate gradient algorithm to minimize the energy function by appropriately approximating the Hessian matrix. With this preconditioning, the convergence of the nonlinear conjugate gradient algorithm can be speeded up dramatically.

3.1. Nonlinear Conjugate Gradient Algorithm

The nonlinear conjugate gradient method for minimizing general nonlinear functions is outlined as follows (Gill et al., 1981):

```
1. $\mathbf{d}_0 = \mathbf{r}_0 = -f'(\mathbf{u}_0)$.
2. Find $\alpha_i$ that minimizes $f(\mathbf{u}_i + \alpha_i \mathbf{d}_i)$.
3. $\mathbf{u}_{i+1} = \mathbf{u}_i + \alpha_i \mathbf{d}_i$.
4. $\mathbf{r}_{i+1} = -f'(\mathbf{u}_{i+1})$.
5. $\beta_{i+1} = \frac{\mathbf{r}_{i+1}^T \mathbf{r}_{i+1}}{\mathbf{r}_i^T \mathbf{r}_i}$ or $\beta_{i+1} = \max\{\frac{\mathbf{r}_{i+1}^T(\mathbf{r}_{i+1} - \mathbf{r}_i)}{\mathbf{r}_i^T \mathbf{r}_i}, 0\}$.
6. $\mathbf{d}_{i+1} = \mathbf{r}_{i+1} + \beta_{i+1} \mathbf{d}_i$.
7. If $\mathbf{r}_k \simeq \mathbf{0}$, stop; else go to step 2.
```

Step 2 requires a line search method to minimize the function $f(\mathbf{u}_i + \alpha_i \mathbf{d}_i)$ at $\mathbf{u}_i$ along the direction $\mathbf{d}_i$. The Newton-Raphson method and the Secant method are two popular line search schemes to approximate the best α_i. We employ the Newton-Raphson method in the nonlinear CG. The Taylor series approximation of the function f up to the second-order terms is used to derive the approximate α_i, giving

$$\alpha_i = -\frac{f'(\mathbf{u}_i)\mathbf{d}_i}{\mathbf{d}_i^T f''(\mathbf{u}_i)\mathbf{d}_i}. \tag{21}$$

In step 5, there are two ways for choosing β. The first is Fletcher-Reeves formula and the other is the Polak-Ribiere formula (Gill et al., 1981). We adopt the latter since it usually converges much more quickly, although the nonlinear CG needs to restart when $\beta = 0$. Note that $f'(\mathbf{u})$ and $f''(\mathbf{u})$ in this algorithm denote the gradient vector and the Hessian matrix of the function f given in Eq. (20). To simplify the algorithm, we regard the normalization factor $1/(I_{x,i}^2 + I_{y,i}^2 + c)$ as a constant in each iteration and compute it using the newest (u_i, v_i) updated in the previous iteration. Thus, we have

$$f'(\mathbf{u}) = \begin{bmatrix} f'_{d,u} \\ f'_{d,v} \end{bmatrix} + \lambda \begin{bmatrix} \mathbf{K}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_s \end{bmatrix} \mathbf{u} \tag{22}$$

$$f''(\mathbf{u}) = \begin{bmatrix} \mathbf{K}_{d,uu} + \lambda \mathbf{K}_s & \mathbf{K}_{d,uv} \\ \mathbf{K}_{d,uv} & \mathbf{K}_{d,vv} + \lambda \mathbf{K}_s \end{bmatrix} \tag{23}$$

where the matrix $\mathbf{K}_s$ is the discrete Laplacian matrix obtained from the membrane smoothness constraint, the vectors $f'_{d,u}$ and $f'_{d,v}$ are obtained by concatenating the components $f'_{d,u,i}$ and $f'_{d,v,i}$ respectively, $\mathbf{K}_{d,uu}$, $\mathbf{K}_{d,uv}$ and $\mathbf{K}_{d,vv}$ are all diagonal matrices with the ith diagonal entries $K_{d,uu,i}$, $K_{d,uv,i}$ and $K_{d,vv,i}$, respectively. The components $f'_{d,u,i}$, $f'_{d,v,i}$, $K_{d,uu,i}$, $K_{d,uv,i}$ and $K_{d,vv,i}$ corresponding to the ith location are defined as follows:

$$f'_{d,u,i} = \frac{1}{I_{x,i}^2 + I_{y,i}^2 + c} \frac{\partial I_1(x_i + u_i, y_i + v_i)}{\partial x} \times (I_1(x_i + u_i, y_i + v_i) - I_0(x_i, y_i)) \tag{24}$$

$$f'_{d,v,i} = \frac{1}{I_{x,i}^2 + I_{y,i}^2 + c} \frac{\partial I_1(x_i + u_i, y_i + v_i)}{\partial y} \times (I_1(x_i + u_i, y_i + v_i) - I_0(x_i, y_i)) \tag{25}$$

$$K_{d,uu,i} = \frac{1}{I_{x,i}^2 + I_{y,i}^2 + c} \left(\frac{\partial I_1(x_i + u_i, y_i + v_i)}{\partial x} \right)^2 \tag{26}$$

$$K_{d,uv,i} = \frac{1}{I_{x,i}^2 + I_{y,i}^2 + c} \left(\frac{\partial I_1(x_i + u_i, y_i + v_i)}{\partial x} \right) \times \left(\frac{\partial I_1(x_i + u_i, y_i + v_i)}{\partial y} \right) \tag{27}$$

$$K_{d,vv,i} = \frac{1}{I_{x,i}^2 + I_{y,i}^2 + c} \left(\frac{\partial I_1(x_i + u_i, y_i + v_i)}{\partial y} \right)^2 \tag{28}$$

Once again, the normalization factor $1/(I_{x,i}^2 + I_{y,i}^2 + c)$ is computed using the most recently updated flow vector (u_i, v_i). Note that the second-order partial derivatives of the SSD function, $K_{d,uu,i}$, $K_{d,uv,i}$ and $K_{d,vv,i}$, are obtained by ignoring the terms involving second-order partial derivatives, which are more sensitive to the high-frequency noise. We can usually obtain more robust convergence by using the above approximation. In the above equations, the values of the image function I_1 and its partial derivatives at $(x_i + u_i, y_i + v_i)$ need to be computed and may be computed using any interpolation technique. In our implementation, we simply use the finite difference in a 3×3 window to compute the partial derivatives at the nodal points and then use the bilinear interpolation to obtain the values at the sub-pixel location.

3.2. Preconditioning Technique

For the nonlinear conjugate gradient algorithm, an ideal preconditioner is the Hessian matrix of f. However, it is too expensive to solve the auxiliary linear system associated with the Hessian preconditioner. Since the structure of the Hessian matrix, given in Eq. (23), is exactly the same as that of the stiffness matrix in our modified gradient-based method, we can still apply the incomplete Cholesky preconditioner used in our modified gradient-based method as a good approximation to the Hessian matrix here. As discussed in Section 2, the incomplete Cholesky preconditioner is very efficient in preconditioning sparse and specially-structured symmetric positive definite (SPD) matrices and the Hessian matrix meets all of these conditions. Thus, we use the nonlinear conjugate gradient algorithm with the incomplete Cholesky preconditioning to solve the nonlinear optimization problem in our SSD-based regularization method.

The adaptive preconditioned conjugate gradient (APCG) described in (Lai and Vemuri, 1997) may also be used for solving the aforementioned optimization problem efficiently. However, it is not as effective as the incomplete Cholesky preconditioner since the APCG assumes that the data, i.e., the (u, v) field, is uncorrelated in designing the preconditioner. This, however, is not the case for the optical flow computation problem. The incomplete Cholesky preconditioning is better suited as it accounts for the correlation in the (u, v) field. The details of this preconditioner was described in Section 2.

4. Experimental Results

In this section, we present results of testing our modified gradient-based regularization algorithm and SSD-based regularization algorithm on a variety of synthetic and real image sequences. Comparisons of our results with the previously reported results (Barron et al., 1994; Szeliski and Coughlan, 1994; Weber and Malik, 1995; Ju et al., 1996) are made to demonstrate the performance of our algorithms described in earlier sections.

The angular error measure (Barron et al., 1994) between the computed optical flow and the true flow is adopted in the experiments as a basis for comparison with the previously reported results. Unlike some other methods which only produce sparse optical flow, both of our regularization-based methods presented in this paper give dense optical flow estimates with 100% density.

4.1. Experiments with Modified Gradient-Based Method

In the implementation of our modified gradient-based regularization method, we treat the zero-crossing points of $S(x, y, t)$ as the discontinuity locations. The computation of the derivatives I_x, I_y and I_t is very noise sensitive. Therefore, we pre-smooth the image brightness function with the Gaussian filter ($\sigma = 1.5$) prior to using the central difference method to approximate the derivatives. The function $S(x, y, t)$ is generated by convolving the original intensity function $I(x, y, t)$ with a mask corresponding to the Laplacian of Gaussian (LOG) filter (with $\sigma = 1.5$) (Marr, 1982) over the image plane at each time instance. The zero-crossings of the function $S(x, y, t)$ are searched along the x and y directions in the current frame to determine if there is a sign change at each location and the corresponding slope is greater than a threshold. We use the regularization parameter $\lambda = 0.4$ for most of the experiments presented in this section, unless otherwise stated explicitly. We found that the results are insensitive to the regularization parameter λ when it is in the range between 0.1 and 1. The constant c in the normalization factor was set to 10 for all the experiments presented in this paper.

Four standard synthetic image sequences are tested by using our modified gradient-based regularization method to compute the optical flow. These four image sequences are the translating *Square*, the *Diverging Tree*, the *Translating Tree* and the *Yosemite* sequences used in (Barron et al., 1994) and also by other researchers. Figure 3 shows one frame from each sequence. The image sizes for the translating square, diverging/translating tree and Yosemite are 100×100, 150×150 and 316×252, respectively.

Our modified gradient-based method contains several modifications, including the incorporation of contour-based flow constraint, normalization of data constraint, and rejection of unreliable constraints. To demonstrate the usefulness of all the components in our algorithm, we test our algorithm with each modification component successively imposed. The accuracy in the computed optical flow with the addition of each component in the translating tree and Yosemite image sequences are shown in Tables 1 and 2, respectively. For all these experiments, the accuracy is reported for a flow of 100% density computed using our incomplete Cholesky preconditioned conjugate gradient

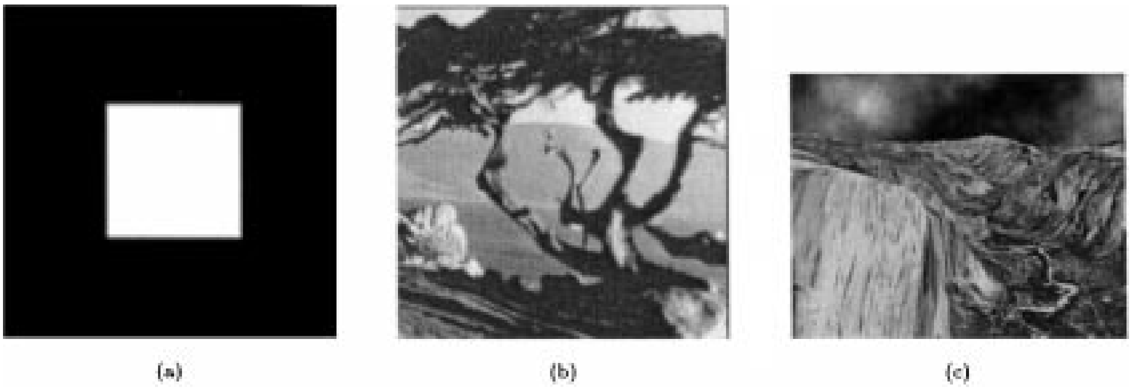


Figure 3. Frames from the (a) *Square* 2, (b) *Diverging/Translating Tree* and (c) *Yosemite* image sequences.

algorithm. Note that the incomplete Cholesky preconditioned conjugate gradient is used to solve the linear systems for all the different settings described in the table. Theoretically speaking, the use of different numerical methods has no influence on the accuracy of the optical flow estimation as long as they are guaranteed to converge to the true solution and the stopping criterion is appropriately set. In (Barron et al., 1994), the computed flow using Horn and Schunck's numerical scheme was reported for an iteration limit of 100 iterations. We found that the computed solution after 100 iterations of Horn and Schunck's numerical scheme is far from the true solution of the linear system, due to its slow convergence rate. Note that our proposed incomplete Cholesky preconditioned CG algorithm is much more efficient than the numerical scheme proposed in (Horn and Schunck, 1981) for the optical flow computation problem. Hence, we obtain much better results—when using only the image flow constraints—than the results of applying Horn and Schunck's method reported in (Barron et al., 1994). We will subsequently address this issue in more detail.

From Table 1, we can see each modification component contributes substantial improvement to the accuracy of the algorithm for the translating tree example. Similarly, Table 2 shows the improvement of using

Table 1. Summary of the results of applying our gradient-based algorithm on the translating tree sequence with each modification component successively imposed. The accuracy is calculated from the computed flow of 100% density.

Modification components used	Avg. error (deg)	St. dev. (deg)
Image constraint	0.48	0.39
Image constraint + contour constraint	0.46	0.35
Image constraint + contour constraint + normalization	0.41	0.30
Image constraint + contour constraint + normalization + rejection	0.40	0.28

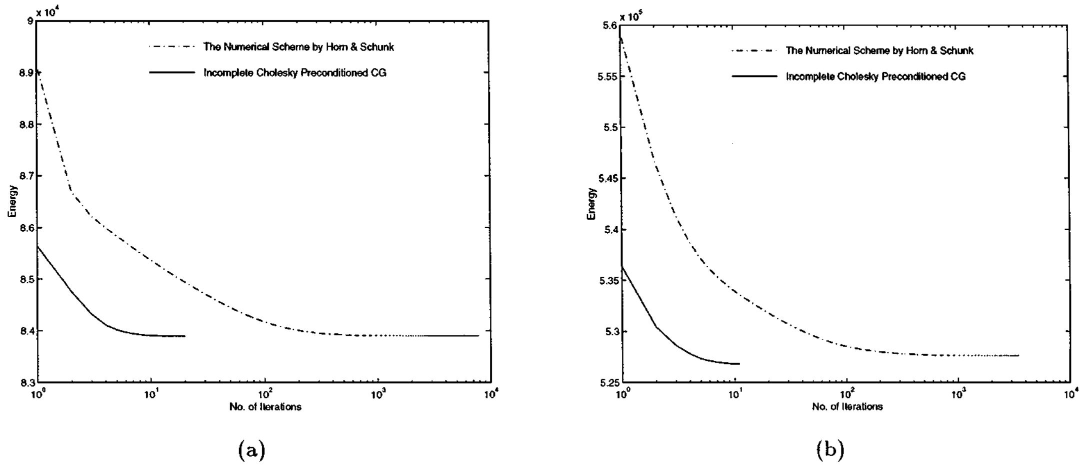
Table 2. Summary of the results of applying our gradient-based algorithm on the Yosemite sequence with each modification component successively imposed. The accuracy is calculated from the computed flow of 100% density with the sky region ignored.

Modification components used	Avg. error (deg)	St. dev. (deg)
Image constraint	2.29	1.79
Image constraint + contour constraint	2.29	1.79
Image constraint + contour constraint + normalization	2.11	1.57
Image constraint + contour constraint + normalization + rejection	1.99	1.41

the modification components in our algorithm on the Yosemite sequence. Note that there is no change in accuracy with the inclusion of contour-based flow constraints. The normalization scheme yields the best improvements for this example. These comparisons provide better insight of performance improvement from each individual modification component in our modified gradient-based algorithm for accurate optical flow computation.

Note that the stiffness matrix **K** in the linear system to be solved is symmetric positive definite (SPD) due to the fact that the energy function to be minimized is quadratic and convex. In (Horn and Schunck, 1981), a variant of Jacobi iteration that is based on splitting the stiffness matrix **K**, a 2×2 block matrix, into a diagonal block matrix and an off-diagonal block matrix was proposed to solve the linear system of the optical flow computation problem. The diagonal block matrix captures the diagonal of each block and off-diagonal block matrix captures the remaining off-diagonal entries. The numerical update scheme is then obtained by taking the term with the diagonal block matrix to the left-hand side of the linear system and the remaining terms are absorbed in the right-hand side which is computed using the solution updated at the previous iteration. However, it is not known if this variant of Jacobi iteration can converge to the unique (true) solution. In addition, Jacobi iteration has been recognized to be less efficient than gradient-type algorithms for solving the linear system with an SPD matrix (Golub and Van Loan, 1989). Our incomplete Cholesky preconditioned CG is guaranteed to converge to the true solution for a linear system with an SPD matrix (Golub and Van Loan, 1989), which is exactly the type of the linear system for the optical flow computation problem. In addition, the proposed incomplete Cholesky preconditioner approximates the stiffness matrix by taking advantage of its special sparse structure as discussed in Section 2.4. This approximation is accomplished by using the incomplete Cholesky decomposition of two similarly structured matrices, leading to a very efficient preconditioning effect.

To demonstrate the efficiency of the proposed numerical algorithm, we depict the convergence curves of both algorithms for the above two examples in Figs. 4(a) and (b). The convergence curve depicts the rate of decrease in the energy function with the increase of number of iterations. Note that the x-axis, i.e., number of iterations, in Fig. 4 is on a logarithmic scale to show two convergence curves of very different convergence rates in the same figure. In Fig. 4(a), the proposed incomplete Cholesky preconditioned CG requires 11 iterations to converge to the solution when employing the image flow constraint alone, whereas the numerical scheme of Horn and Schunck takes 7800 iterations to converge to a solution with average error 0.84° and standard deviation 1.18°. This solution obtained after 7800 iterations of Horn and Schunck's numerical scheme is less accurate than the computed solution using 11 iterations of our incomplete Cholesky preconditioned CG (see Table 1, row 1). For the above computation, the incomplete Cholesky preconditioned CG took 4.88 s of CPU time on a SPARC-20 workstation for the translating



(a)

(b)

Figure 4. Convergence curves of the incomplete Cholesky preconditioned conjugate gradient algorithm and the numerical scheme of Horn and Schunck for the (a) *Translating Tree* and (b) *Yosemite* image sequences.

tree example, while the Horn and Schunck's numerical scheme took 629.04 s of the SPARC-20 CPU time. Similarly, for the example of Yosemite sequence, the convergence curves for both numerical methods are shown in Fig. 4(b). With 10 iterations of our incomplete Cholesky preconditioned CG, the optical flow is converged at the accuracy shown in Table 2, while the numerical scheme of Horn and Schunck requires 3500 iterations to achieve similar accuracy. In terms of computational time for this example, our algorithm and Horn and Schunck's numerical scheme took 11.45 and 1041.94 CPU seconds on a SPARC-20 workstation, respectively.

In the following, we compare the results of our modified gradient-based algorithm to the results of other gradient-based methods reported in literature (Barron et al., 1994; Weber and Malik, 1995; Ju et al., 1996) on several standard image sequences. The motion in the Square 2 sequence is a constant translation. In our experiment, the regularization parameter λ is chosen to be 0.1, and the weighting parameters α and β for the image flow constraint and contour-based flow constraint are set to 1. The optical flow obtained after 50 iterations of the incomplete Cholesky preconditioned CG algorithm is shown in Fig. 5(a). The errors of our modified gradient-based regularization method and the other gradient-based methods reported in (Barron et al., 1994) for this translating square example are given in Table 3. In this example, when comparing the results from those gradient-based methods that yield estimates of 100% flow density, our modified gradient-based method dramatically outperforms them. Our modified gradient-based method also compares favorably with the best existing gradient-based methods that produce sparse flow estimates, even though our method yields estimates of 100% flow density.

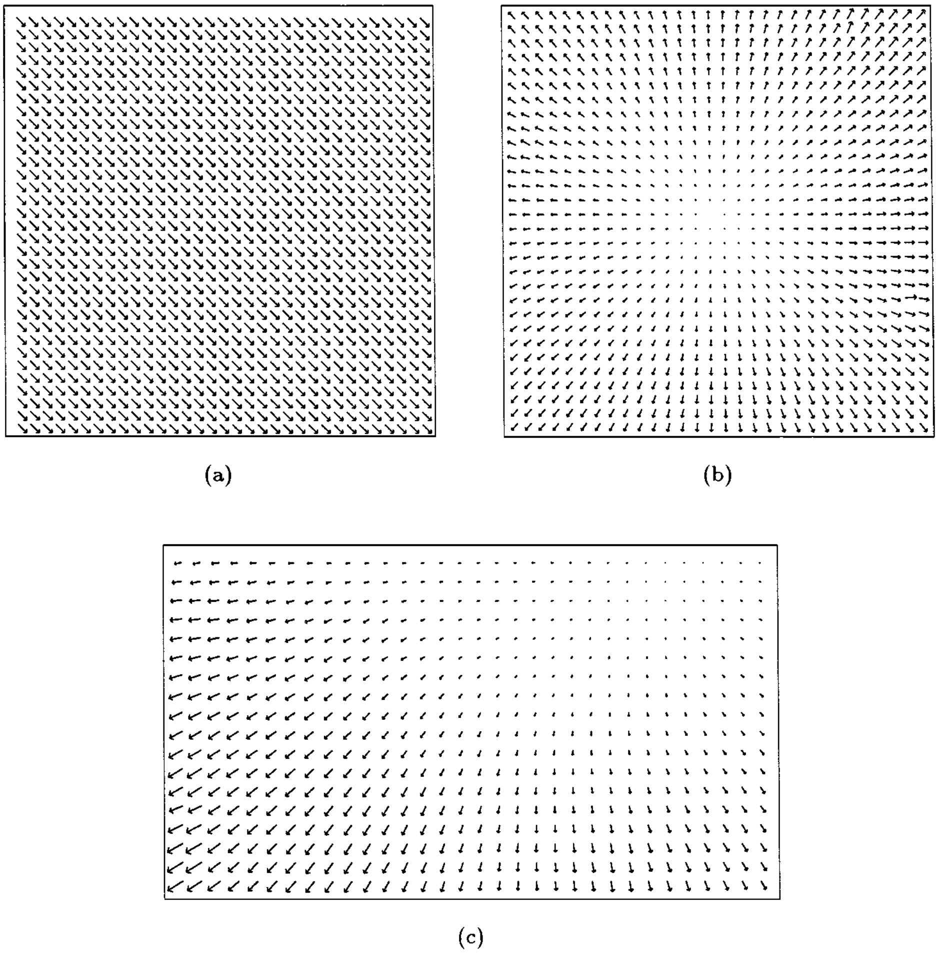


Figure 5. Computed optical flow of the (a) *Square* 2, (b) *Diverging Tree* and (c) *Yosemite* image sequences after 50 iterations of the incomplete Cholesky preconditioned CG algorithm.

Table 3. Summary of *Square* 2 results.

Technique	Avg. error (deg)	St. dev. (deg)	Density (%)
Horn and Schunck (modified)	32.81	13.67	100
Nagel	34.57	14.38	100
Lucas and Kanade ($\lambda_2 \geq 1.0$)	0.21	0.16	7.9
Uras et al.	0.16	0.10	26.1
Lai and Vemuri (gradient-based)	0.15	0.09	100

Table 4. Summary of *Diverging Tree* results.

Technique	Avg. error (deg)	St. dev. (deg)	Density (%)
Horn and Schunck (modified)	2.55	3.67	100
Nagel	2.94	3.23	100
Uras et al.	4.64	3.48	100
Lucas and Kanade ($\lambda_2 \geq 1.0$)	1.94	2.06	48.2
Weber and Malik	3.18	2.50	88.6
Lai and Vemuri (gradient-based)	1.34	1.05	100

The second synthetic data set that we used is the *Diverging Tree* image sequence. The underlying motion in this example is a bilinear expansion from the center of the image. The error in the computed optical flow after 50 iterations of the incomplete Cholesky preconditioned CG algorithm is presented in Table 4, along with results from other gradient-based methods in literature (Barron et al., 1994; Weber and Malik, 1995). Once again, our modified gradient-based regularization method produces 100% flow estimates and the result compares favorably to the best existing methods which produce dense as well as sparse flow estimates. The computed optical flow is shown in Fig. 5(b).

We also tested our modified gradient-based algorithm on the more complex *Yosemite* image sequence (Barron et al., 1994). The result of our modified gradient-based method on this example is compared with other gradient-based methods reported in literature (Barron et al., 1994; Weber and Malik, 1995; Ju et al., 1996) in Table 5. The superscript "*" is used to indicate that the flow error is computed by discarding the sky region from the images. The result of our method in this example compares favorably with other best existing gradient-based methods. The optical flow obtained after 50 iterations of this algorithm is depicted in Fig. 5(c).

Our modified gradient-based algorithm was also tested on two real data sets namely, the *NASA* and *Hamburg Taxi* sequences. Figure 6 shows one frame each of these two sequences and the computed optical flow after 50 iterations of the incomplete Cholesky preconditioned conjugate gradient algorithm.

Table 5. Summary of *Yosemite* results for gradient-based methods.

Technique	Avg. error (deg)	St. dev. (deg)	Density (%)
Horn and Schunck (modified)	9.78	16.19	100
Uras et al.	8.94	15.61	100
Lucas and Kanade ($\lambda_2 \geq 1.0$)	4.28	11.41	35.1
Weber and Malik	4.31	8.66	64.2
Black and Anandan*	4.46	4.21	100
Black and Jepson*	2.29	2.25	100
Ju et al.*	2.16	2.0	100
Lai and Vemuri* (gradient-based)	1.99	1.41	100

4.2. *Experiments with SSD-Based Regularization Method*

The experimental results of using our SSD-based regularization method on two synthetic and two real image sequences are presented here. The two synthetic image sequences are the *Translating Tree* and the *Yosemite* image sequences. Frames from these sequences are shown in Fig. 3. Our results on these two examples are compared to the best optical flow estimates reported in literature.

Only two consecutive frames are used to compute the optical flow in our SSD-based regularization method, although more frames can be incorporated to obtain more consistent and smoothed optical flow estimates over time. In this section, we demonstrate that the results obtained by our two-frame SSD-based method compare favorably to other methods.

In our implementation, a coarse-to-fine strategy is combined with the preconditioned nonlinear conjugate gradient algorithm to find the minimum energy solution. Three resolutions are used in the coarse-to-fine strategy and 50 iterations of the preconditioned nonlinear conjugate gradient algorithm are computed for each resolution. By using the coarse-to-fine strategy, we can deal with large displacement problems with better convergence property during the solution search. The regularization parameter λ is empirically chosen to be 0.4 for the implementation of our SSD-based method. The results are quite stable for the regularization parameter in the range between 0.1 and 1. The constant c

in the normalization factor was set to 100 for all the experiments presented in this paper.

For the translating tree example, each frame contains more complicated highly textured regions, which result in multiple local minima in the associated energy function to be minimized. The computed optical flow is shown in Fig. 7(a). The error in the computed optical flow is depicted in Table 6 along with the errors from the best existing results reported in literature (Barron et al., 1994; Weber and Malik, 1995; Szeliski and Coughlan, 1994). Our SSD-based regularization method compares favorably to the best existing methods in this example. Although the Fleet and Jepson's

Table 6. Summary of *Translating Tree* results.

Technique	Avg. error (deg)	St. dev. (deg)	Density (%)
Horn and Schunck (modified)	2.02	2.27	100
Uras et al.	0.62	0.52	100
Szeliski and Coughlan	0.35	0.34	100
Lucas and Kanade ($\lambda_2 \geq 5.0$)	0.56	0.58	13.1
Weber and Malik	0.49	0.35	96.8
Fleet and Jepson	0.23	0.19	49.7
Lai and Vemuri (gradient-based)	0.40	0.28	100
Lai and Vemuri (SSD-based)	0.30	0.27	100

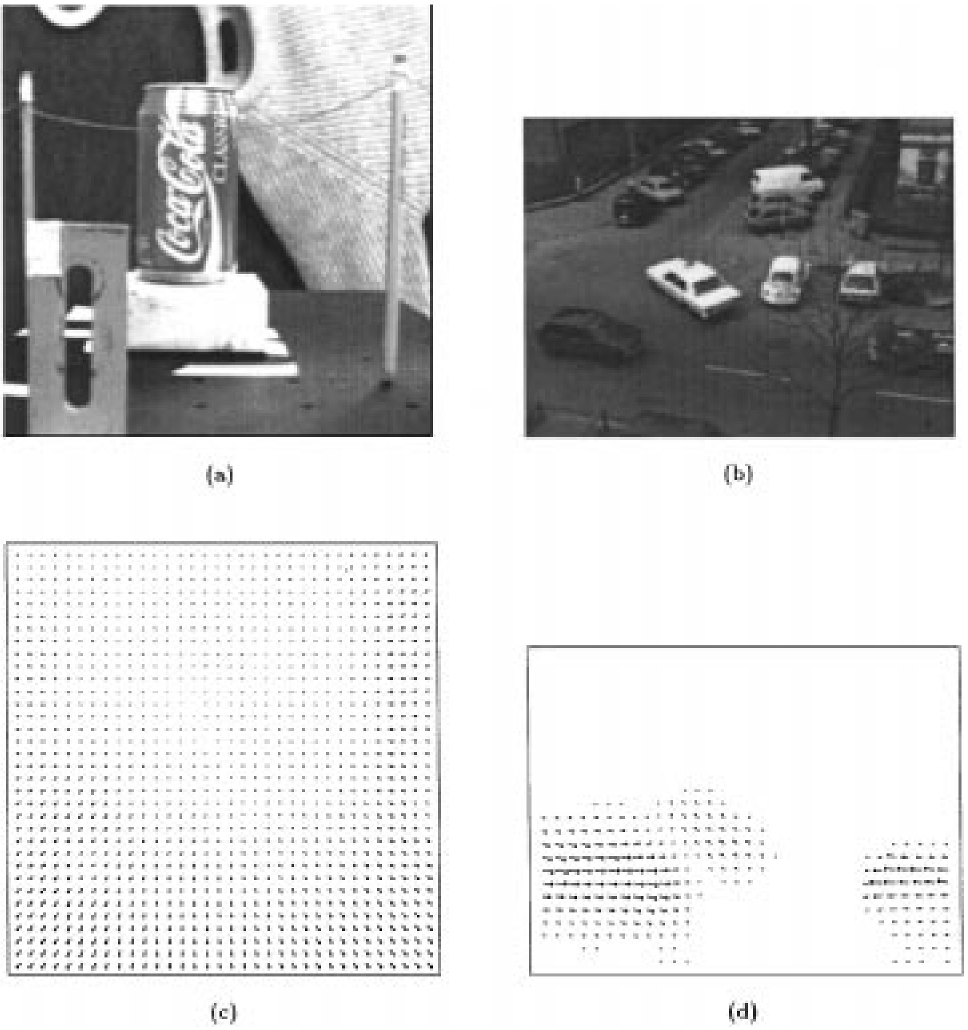


Figure 6. One frame from (a) the *NASA* sequence (b) the *Hamburg Taxi* sequence. Computed optical flow of (c) the *NASA* sequence (b) the *Hamburg Taxi* sequence using our modified gradient-based method after 50 iterations of the incomplete Cholesky preconditioned CG algorithm.

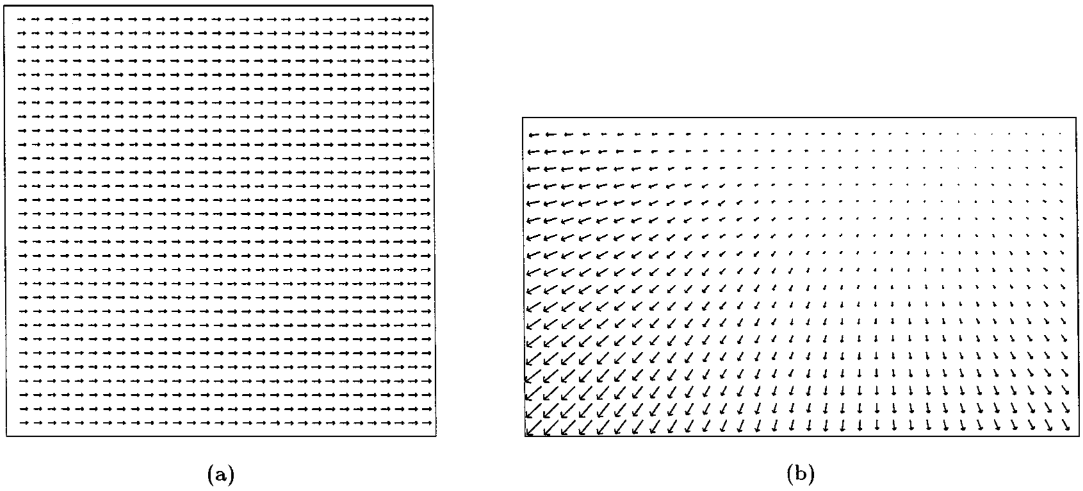


Figure 7. Computed optical flow of the (a) *Translating Tree* and (b) *Yosemite* image sequences, using the preconditioned nonlinear CG algorithm.

phase-based method produces sparse but more accurate result than our SSD-based method, their result was obtained by using 21 frames and we only used two consecutive frames in our SSD-based method. Most of the methods listed in Table 6 require more than two frames to compute the optical flow, except our SSD-based method and the method by Szeliski and Coughlan (1994). Note that the result of Szeliski and Coughlan in this table is adopted for the two-frame case to compare with the result of our two-frame algorithm. It is obvious that a more accurate estimation can be obtained by incorporating more frames into the energy function as shown in (Szeliski and Coughlan, 1994).

We also tested our SSD-based algorithm on the *Yosemite* sequence. We compared our modified gradient-based method with the other gradient-based methods in Table 5 for the *Yosemite* sequence. In Table 7, we present the results for our SSD-based method as well as our modified gradient-based algorithms along with the best results reported in literature (Barron et al., 1994; Szeliski and Coughlan, 1994; Weber and Malik, 1995; Ju et al., 1996) for this sequence. Once again, the sign "*" followed by a method indicates that the error was computed without the sky region. Both our SSD-based method and modified gradient-based method compare favorably to the best existing methods in this example. Note that only two consecutive frames are used in our SSD-based algorithm to generate accurate and dense optical flow estimates with 100% density. The average error in the computed flow using our modified gradient-based method is slightly smaller than that obtained using our SSD-based scheme possibly due to the fact that the former uses more than two frames in the computation of optical flow, thereby smoothing out the errors. In addition, our modified gradient-based method is computationally much faster than our SSD-based method, since the former leads to solving a linear system whereas the latter leads to a nonlinear and nonconvex optimization problem.

Finally, our SSD-based regularization method is applied to two real image sequences namely, the *SRI* sequence and the *Hamburg Taxi* sequence. One frame of each sequence and the computed optical flow are shown in Fig. 8. The computed optical flow between

Table 7. Summary of *Yosemite* results.

Technique	Avg. error (deg)	St. dev. (deg)	Density (%)
Horn and Schunck (modified)	9.78	16.19	100
Uras et al.	8.94	15.61	100
Fleet and Jepson	4.63	13.42	34.1
Lucas and Kanade ($\lambda_2 \geq 1.0$)	4.28	11.41	35.1
Weber and Malik	4.31	8.66	64.2
Black and Anandan*	4.46	4.21	100
Szeliski and Coughlan*	2.45	3.05	100
Black and Jepson*	2.29	2.25	100
Ju et al.*	2.16	2.0	100
Lai and Vemuri* (gradient-based)	1.99	1.41	100
Lai and Vemuri* (SSD-based)	2.04	1.52	100

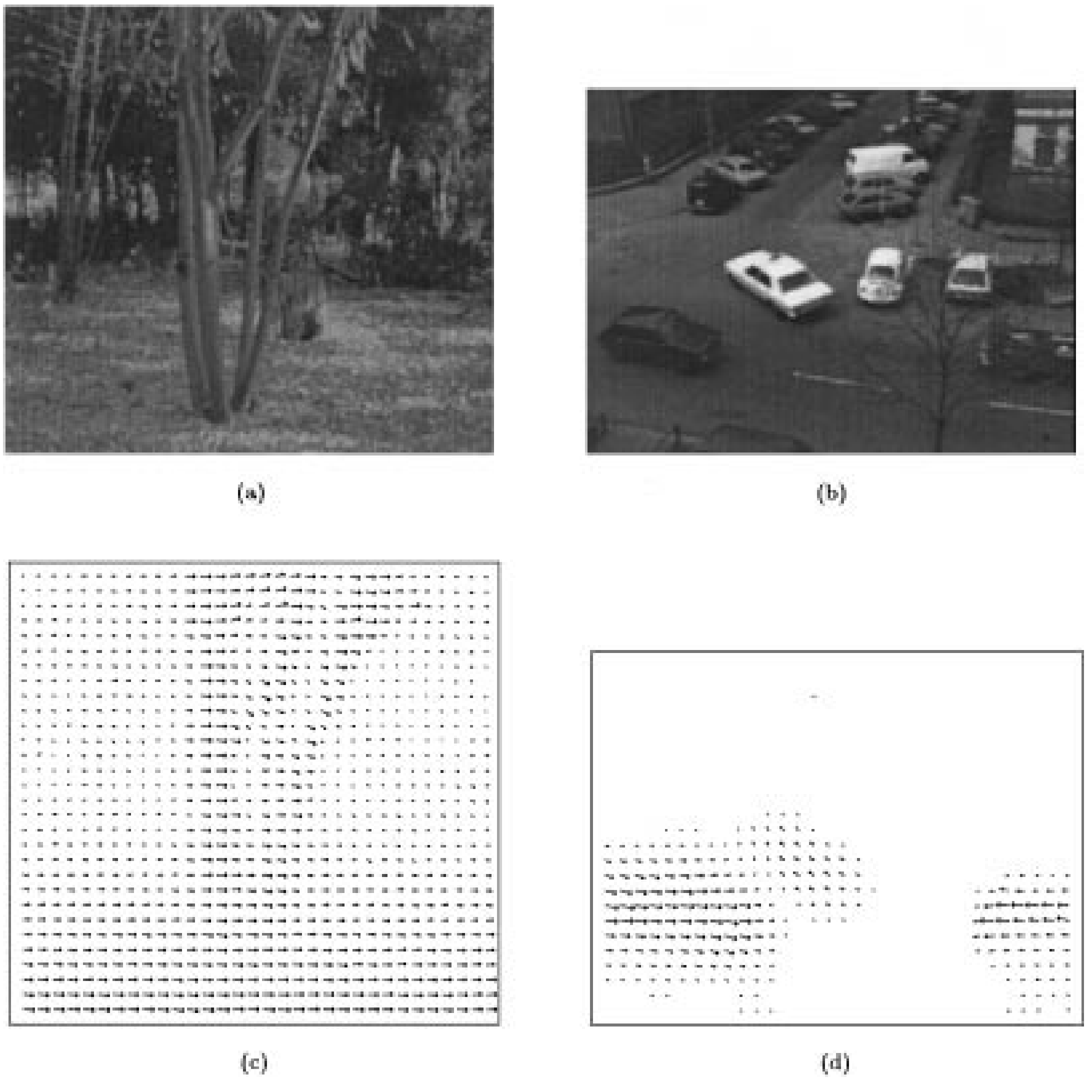


Figure 8. One frame from (a) the *SRI* sequence (b) the *Hamburg Taxi* sequence. Computed optical flow of (c) the *SRI* sequence (b) the *Hamburg Taxi* sequence using our SSD-based regularization method after 150 iterations of the preconditioned nonlinear conjugate gradient algorithms performed in three resolutions.

two consecutive images was obtained after 150 iterations of the preconditioned nonlinear conjugate gradient algorithm performed using a coarse-to-fine three-level multiresolution strategy.

5. Conclusion

In this paper, we presented two new algorithms for reliable and efficient computation of the optical flow from an image sequence. One is the modified gradient-based regularization algorithm, and the other is the SSD-based regularization algorithm. The modified gradient-based method selectively combines the image flow constraint and the contour-based flow constraint into the data constraint. In addition, each data constraint is appropriately normalized to yield a minimum distance constraint instead of the conventional image flow constraint. These modifications combined with a measure which rejects unreliable constraints leads to a reliable and accurate optical flow estimation as is evidenced from our experimental results. We developed a fast numerical algorithm that is based on the incomplete Cholesky preconditioned conjugate gradient for solving the linear system associated with this modified gradient-based method. The flow estimates obtained using our method are more accurate than those obtained from other gradient-based techniques reported in literature.

Our SSD-based regularization method uses a normalized SSD measure as the data constraint in a regularization framework. The resulting energy function is neither quadratic nor convex. The nonlinear conjugate gradient in conjunction with an incomplete Cholesky preconditioning was developed to minimize this energy function. Very accurate results were obtained by applying this algorithm to (only) two consecutive images of a sequence.

Comparing the two algorithms proposed in this paper, we conclude that the modified gradient-based regularization algorithm is computationally more efficient than the SSD-based regularization algorithm, while the latter can provide accurate optical flow estimation from only two consecutive frames in a sequence. Also, we have experimentally demonstrated that our algorithms for optical flow computation compare favorably to the best existing methods reported to date in literature. Our future work will focus on automatically detecting motion discontinuities in the framework presented here.

References

Anandan, P. 1989. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310.

Barron, J.L., Fleet, D.J., and Beauchemin, S.S. 1994. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77.

Black, M. and Anandan, P. 1996. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104.

Black, M. and Jepson, A. 1996. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Trans. Patt. Anal. Mach. Intell.*, 18(10):972–986.

Duncan, J.H. and Chou, T.-C. 1992. On the detection of motion and the computation of optical flow. *IEEE Trans. Patt. Anal. Mach. Intell.*, PAMI-14(3):346–352.

Elman, H.C. 1986. A stability analysis of incomplete LU factorizationss. *Mathematics of Computation*, 47(175):191–217.

Gill, P.E., Murray, W., and Wright, M.H. 1981. *Practical Optimization*. Academic Press: London, New York.

Golub, G.H. and Van Loan, C.F. 1989. *Matrix Computations*. 2nd edition, The Johns Hopkins University Press.

Heitz, F. and Bouthemy, P. 1993. Multimodal estimation of discontinuous optical flow using Markov random fields. *IEEE Trans. Patt. Anal. Mach. Intell.*, PAMI-15:1217–1232.

Hildreth, E.C. 1984. Computations underlying the measurement of visual motion. *Artificial Intelligence*, 23:309–354.

Horn, B.K.P. and Schunck, B.G. 1981. Determining optical flow. *Artificial Intelligence*, 17:185–203.

Ju, S., Black, M., and Jepson, A. 1996. Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, pp. 307–314.

Kumar, A., Tannenbaum, A., and Balas, G. 1996. Optical-flow: A curve evolution approach. *IEEE Trans. Image Processing*, 5(4):598–610.

Lai, S.-H. and Vemuri, B.C. 1997. Physically based adaptive preconditioning for early vision. *IEEE Trans. Patt. Anal. Mach. Intel.*, 19:594–607.

Lucas, B. and Kanade, T. 1981. An iterative image registration technique with an application to stereo vision. In *Proc. DARPA Image Understanding Workshop*, pp. 121–130.

Marr, D. 1982. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Freeman.

Meijerink, J.A. and van der Vorst, H.A. 1977. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Mathematics of Computation*, 31(137):148–162.

Nagel, H.-H. 1983. Constraints for the estimation of displacement vector fields from image sequences. In *Proc. IJCAI83*, Karlsruhe, Germany, pp. 945–951.

Nagel, H. and Enkelmann, W. 1986. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. Patt. Anal. Mach. Intell.*, PAMI-8(5):565–593.

Negahdaripour, S. and Yu, C. 1993. A generalized brightness change model for computing optical flow. In *Proc. International Conference on Computer Vision*, Berlin, Germany, pp. 2–11.

Rudin, L., Osher, S., and Fatemi, E. 1993. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268.

Szeliski, R. and Coughlan, J. 1994. Hierarchical spline-based image registration. In *Proc. IEEE Conference on Computer Vision Pattern Recognition*, Seattle, Washington, pp. 194–201.

Szeliski, R. and Shum, H. 1995. Motion estimation with quadtree splines. In *Proc. International Conference on Computer Vision*, Cambrige, MA, pp. 757–763.

Terzopoulos, D. 1986. Regularization of inverse visual problems involving discontinuities. *IEEE Trans. Patt. Anal. Mach. Intell.*, PAMI-8(4):413–424.

Uras, S., Girosi, F., Verri, A., and Torre, V. 1988. A computational approach to motion perception. *Biol. Cybern.*, 60:79–87.

Verri, A. and Poggio, T. 1989. Motion field and optical flow: Qualitative properties. *IEEE Trans. Patt. Anal. Mach. Intell.*, 11(5):490–498.

Waxman, A. and Wohn, K. 1985. Contour evolution, neighborhood deformation, and global image flow: Planner surfaces in motion. *International Journal of Robotics Research*, 4:95–108.

Weber, J. and Malik, J. 1995. Robust computation of optical flow in a multiscale differential framework. *International Journal of Computer Vision*, 14(1):67–81.