



ELSEVIER

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Computer Vision
and Image
Understanding

Computer Vision and Image Understanding 97 (2005) 315–346

www.elsevier.com/locate/cviu

Accurate optical flow computation under non-uniform brightness variations

Chin-Hung Teng^a, Shang-Hong Lai^{b,*}, Yung-Sheng Chen^c,
Wen-Hsing Hsu^{a,d}

^a Department of Electrical Engineering, National Tsing Hua University, Hsinchu 300, Taiwan, ROC

^b Department of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan, ROC

^c Department of Electrical Engineering, Yuan Ze University, Chung-Li 320, Taiwan, ROC

^d Institute of Information Science, Academia Sinica, Taipei 115, Taiwan, ROC

Received 10 November 2003; accepted 2 August 2004

Available online 11 September 2004

Abstract

In this paper, we present a very accurate algorithm for computing optical flow with non-uniform brightness variations. The proposed algorithm is based on a generalized dynamic image model (GDIM) in conjunction with a regularization framework to cope with the problem of non-uniform brightness variations. To alleviate flow constraint errors due to image aliasing and noise, we employ a reweighted least-squares method to suppress unreliable flow constraints, thus leading to robust estimation of optical flow. In addition, a dynamic smoothness adjustment scheme is proposed to efficiently suppress the smoothness constraint in the vicinity of the motion and brightness variation discontinuities, thereby preserving motion boundaries. We also employ a constraint refinement scheme, which aims at reducing the approximation errors in the first-order differential flow equation, to refine the optical flow estimation especially for large image motions. To efficiently minimize the resulting energy function for optical flow computation, we utilize an incomplete Cholesky preconditioned conjugate gradient algorithm to solve the large linear system. Experimental results on some synthetic and real image sequences show that the proposed algorithm compares favorably to most existing techniques reported in literature in terms of accuracy in optical flow computation with 100% density.

* Corresponding author. Fax: +886 3 572 3694.

E-mail address: lai@cs.nthu.edu.tw (S.-H. Lai).

© 2004 Elsevier Inc. All rights reserved.

Keywords: Optical flow computation; Motion analysis; Motion estimation; Brightness variation; Robust estimation; Motion discontinuity

1. Introduction

Without doubt, optical flow, which is commonly described as the apparent motion of image brightness patterns in an image sequence, provides very important information for estimating 3D velocity fields, analyzing object motion, segmenting images into regions based on their motions, estimating 3D camera motion parameters or recovering 3D structure of objects in the image. The importance of optical flow (or conventionally image motion) has motivated the development of many methods, which can be roughly classified into four categories: gradient-based, correlation-based, energy-based, and phase-based approaches [1].

The gradient-based method is constructed based on the optical flow constraint, which is derived from the brightness constancy model (BCM) as well as the first-order Taylor series approximation. A notable gradient-based method is Horn and Schunck's regularization method [2], where they estimated the optical flow by utilizing the optical flow constraint in conjunction with a first-order smoothness constraint. Although the traditional gradient-based regularization method does not seem to perform well in benchmarking experimental comparisons [1], Lai and Vemuri [3] demonstrated that, with an efficient algorithm, namely the incomplete Cholesky preconditioned conjugate gradient algorithm, the gradient-based regularization method can produce very accurate optical flow estimation. In fact, the unsatisfactory performance of Horn and Schunck's regularization method [1] is mainly due to the insufficient number of iterations of their numerical method in the experiments. Generally, Horn and Schunck's method can still generate very accurate optical flow as long as sufficient iterations, typically several thousands, are applied. The incomplete Cholesky preconditioning technique can greatly accelerate the convergence speed of the iterative numerical algorithm for the resulting linear system, thus a more stable solution can be obtained within acceptable computing time. This point had been addressed in [3]. However, the gradient-based method still suffers from the following problems. Firstly, due to the brightness constancy assumption implied by the conventional optical flow constraint, the gradient-based approach is inaccurate when the image sequence contains illumination variations in the spatial or temporal domain. Secondly, this approach cannot provide correct image motion in the vicinity of motion discontinuities owing to the smoothness assumption. Thirdly, the first-order Taylor series approximation used in this approach also introduces errors in the optical flow constraint especially for large image motion. Finally, unreliable data constraint introduced by numerical approximation, image aliasing and image noise yields inaccuracy in the optical flow computation. Hence, to obtain a very accurate gradient-based regularization method based on Lai and Vemuri's framework, these issues should be considered comprehensively. In this work, we aim at extending Lai

and Vemuri's framework to take all of the above issues into consideration. We include several methods resolving all the above issues into this framework and modify the numerical algorithm to obtain more accurate optical flow estimation under more realistic situations. In the following, we will roughly describe previous research dealing with the above issues and how we extend Lai and Vemuri's framework to improve the accuracy of optical flow computation.

Brightness constancy model is obtained by assuming that the image brightness values of corresponding points are invariant from one time instant to the next. However, owing to the complex lighting interactions between the imaging system and the scene objects, for instance, moving illumination sources, shadows casting on moving objects, changing surface orientation under directional illumination, automatic gain control (AGC) of camera, etc., the brightness constancy assumption is violated and therefore the BCM-based approaches are inaccurate at these situations. Recently, many methods have been proposed to cope with the problem of brightness variations [4–10]. For example, Hampson and Pesquet [4] introduced an additional coefficient in their pel-recursive motion estimator to estimate both motion and illumination variation fields. Lai [5] utilized a Laplacian-of-Gaussian filter to remove the low-frequency portion of the additive non-uniform illumination factor, thus alleviating the influence of non-uniform illumination variations. Nomura et al. [6] derived two extended optical flow constraints by assuming that the image intensity function can be factorized into a non-uniform illumination function and an image intensity function under uniform illumination. In their derivation, these two constraints are developed under different conditions. For example, one is obtained when the illumination variation is in the spatial domain and the other is derived when the variation is in the temporal domain. Nomura [7] and Zhang et al. [8] had successfully applied these two constraints in detecting image motion fields under non-uniform or non-stationary illumination variations. In some applications such as *Shape-from-X* techniques, the brightness variation provides rather important information for recovering 3D shape of the scene. Therefore, Negahdaripour proposed a revised definition of optical flow to represent the geometric and radiometric variations in dynamic imagery [9]. This revised definition of optical flow is a three-component vector instead of the traditional two-component displacement vector with the third component denoting the brightness variation of this point. Negahdaripour proposed a generalized dynamic image model (GDIM) to account for the radiometric variations in the problem of estimating optical flow with brightness variations. In GDIM, the brightness variation is represented by a multiplier and an offset fields, thus it is more general than Nomura's model [6] that uses only a multiplier field in optical flow computation [7,8]. GDIM was also adopted by Yeasin for estimating optical flow in a log-mapped image plane [11]. In fact, GDIM is not sufficient to obtain complete information about the behavior of brightness variations. Haussecker and Fleet [10] proposed some physical models of brightness variation, including the brightness variations caused by diffusion, moving illumination envelope or changed surface orientation, etc., for optical flow computation. With these physical models, brightness variations can be described more precisely and thus more inside information about the cause of brightness variations can be obtained. However, these physical models

are usually constructed from some prior knowledge about the system, thus their method is only appropriate for the applications where the physical model is known in advance. In general, we cannot assume the physical model underlying the brightness variation is given, therefore we adopt the revised definition of optical flow proposed by Negahdaripour and embed GDIM into a gradient-based regularization method to estimate the revised optical flow in this work. However, this does not imply that the proposed framework is inadequate for the more realistic models proposed by Haussecker and Fleet. Their models can still be used in the proposed framework if the cause of brightness variation is known in advance. Because we embed GDIM into the gradient-based regularization framework, the parameters to be estimated at each pixel are extended from two (image motion) to four (image motion plus brightness variation multiplier and offset fields). Therefore, the incomplete Cholesky preconditioned conjugate gradient algorithm should be modified accordingly. The coefficient matrix grows from a 2×2 block structure to a 4×4 one. In this paper, we derive general formulas for designing the incomplete Cholesky decomposition of the enlarged coefficient matrix such that Lai and Vemuri's framework can be extended to accommodate the situation of brightness variations.

Although the GDIM-based optical flow constraint can correctly model image motion and brightness variation, different types of noise make the constraint unreliable and lead to errors in optical flow estimation. These noises come from several sources. Firstly, numerical approximation of the partial derivatives in the optical flow constraints inevitably introduces errors. Secondly, the derivation of the optical flow constraint involves the first-order Taylor series approximation, thus leading to errors in the constraint. Thirdly, image aliasing as well as image acquisition noises also make the data constraint unreliable. Finally, the optical flow constraints are usually not accurate at locations of motion discontinuities. One way to reduce the influence of unreliable flow constraints on the optical flow estimation is by employing *robust estimation* [12,13]. The square error function, which is often used in least-squares model fitting, is replaced by a robust error function. Robust error function has the characteristics that data constraints with large residues are suppressed in the final estimation, thus getting more robust solution against various kinds of constraint errors and image noises. A notable robust estimation algorithm in optical flow computation is Black and Anandan's method [13], where they utilized the robust error functions in the optical flow and smoothness constraints. However, applying the robust error function leads to minimize a nonlinear function thus requires a more complex numerical minimization algorithm. Therefore, the robust estimation scheme cannot be applied in Lai and Vemuri's framework directly. Fortunately, a reweighted least-squares method can be used to approximate the solution of the robust estimation scheme. A weight, which is derived from the adopted robust error function, is assigned to each flow constraint so that the unreliable flow constraints can be efficiently suppressed. This reweighted method can be easily incorporated into Lai and Vemuri's numerical framework simply by modifying the data constraints at each iteration of the numerical algorithm.

As already mentioned, motion discontinuity or motion boundary is a serious problem in computing optical flow due to the smoothness constraint that is usually imposed in the optical flow estimation algorithm. In fact, motion boundary detection is a difficult problem in computer vision because there is usually no direct information available for motion boundary. To solve this problem, Black and Fleet [14] proposed a motion boundary model that encodes the information of the boundary orientation, velocities of each side, motion of occluding edge over time, and appearance/disappearance of pixels at the boundary to describe the behavior of motion boundary. Although this model can provide rich information about motion boundary, it is complex and difficult to incorporate it into a gradient-based regularization method. Many other techniques were also developed focusing on this topic. For example, robust estimation has been applied to alleviate the errors in the flow constraint near motion boundaries [12,13,15]. Kumar et al. proposed a curve evolution approach that does not smooth the flow-velocity across the edge and hence preserves motion discontinuities [16]. Heitz and Bouthemy described a multimodal estimation algorithm that uses coupled Markov Random Fields to model the motion fields and preserve motion boundaries [17].

For gradient-based regularization methods, the problem of motion boundary can be coped with by adjusting the significance of smoothness constraint according to the locations of motion boundaries. For instance, Nagel introduced the “oriented smoothness” constraint to limit the variation of the motion field only in the directions along which the corresponding motion vector component cannot be inferred from the spatio-temporal gray value variations [18]. More precisely, if the gray value variation contains sufficient information (e.g., corners) to estimate the motion vector, the smoothness contribution to the estimation of both motion vector components should be significantly suppressed. On the other hand, if the gray value variation corresponds locally to a one-dimensional image structure (e.g., edges), the smoothness contribution should be suppressed only in the direction with strong gray value variation. Finally, if the gray values are nearly constant, the smoothness constraint can be freely applied on both motion vector components. Based on these concepts, Nagel introduced a weight matrix to control the smoothness constraint such that the motion field perpendicular to the image gradient and principal curvature directions is varied as gradually as possible. In addition to Nagel’s method, there are other approaches developed to attack this problem. For example, Weickert and Schnörr extended the flow-driven spatial smoothness constraint to spatio-temporal regularizers to improve the accuracy of estimated optical flow fields and hence produce accurate motion boundaries [20]. Their new formulation leads to a rotationally invariant and time symmetric convex optimization problem such that a unique solution can be found by the standard minimization algorithm. In summary, these methods can be classified into two categories: *image-driven* and *flow-driven*, according to the type of information used in determining the strength of smoothness constraint. Examples of image-driven smoothness constraint include Nagel and Enkelmann’s approach [19] and the related subsequent works [21–23] and the works by Schnörr [24] and Snyder [25]. Examples of the flow-driven approach include the works in

[16,20,26,27]. In this work, we develop a *dynamic smoothness adjustment scheme* that is appropriate for Lai and Vemuri's framework to cope with the problem of motion discontinuity. We control the strength of the smoothness constraint by assigning an appropriate weight for each component of the constraint. We found that the angular deviations between neighboring motion vectors¹ can be used to locate the motion discontinuity properly. Therefore, we determine these weights from the current estimated flow fields by considering the statistical characteristics of the angular deviations between adjacent motion vectors. To more precisely characterize the smoothness constraint, these weights are further refined from the edge information in the image domain. From our experimental results, we found that this dynamic weight adjustment scheme can correctly capture motion boundaries and greatly improve the accuracy of optical flow estimation.

Another restriction on the gradient-based method is the error introduced by neglecting the higher-order terms in the Taylor series expansion of the function $I(x + u\delta t, y + v\delta t, t + \delta t)$. This error will be large especially when the sequence contains large image motions. In literature, this problem could be solved by applying the hierarchical motion estimation [28]. The motion vectors estimated from low resolution image sequence could be used to compute the image motion in the high resolution image sequence. In this work, we exploit the estimated optical flow from previous iterations to refine the optical flow constraint, thus reducing the approximation error. We call it a *constraint refinement scheme* for the reason that the optical flow constraints are refined in the same level of images, not from the low resolution images. We explicitly indicate how to modify the numerical algorithm to accommodate the constraint refinement scheme and show how much improvement can be achieved by this constraint refinement scheme through experiments. Although we do not apply the coarse-to-fine scheme, which has been commonly used for motion estimation, in our implementation, it can also be incorporated into the proposed framework easily.

In this paper, we have successfully incorporated the above proposed auxiliary estimation schemes into a gradient-based regularization method. Experiments on real and synthetic image sequences demonstrate that the proposed algorithm can produce excellent performance in estimating optical flow with non-uniform brightness variations. In the following, the details of these auxiliary estimation schemes and the minimization algorithm are described. The remainder of this paper is organized as follows. In the next section, we describe the proposed gradient-based regularization method including the energy-minimization formulation, robust estimation, dynamic smoothness adjustment and constraint refinement scheme. In Section 3, the numerical minimization algorithm, i.e., the incomplete Cholesky preconditioned conjugate gradient algorithm, is presented. Experimental results on synthetic and real image sequences are shown in Section 4. Finally, we conclude in Section 5.

¹ Here, we define the motion vector as $(u, v, 1)$, where (u, v) is the image motion. Thus, the image motions with the same orientation but different magnitude can also be distinguished by the angular deviations between them.

2. Proposed optical flow computation

2.1. Energy function formulation

The original formulation of the gradient-based regularization method proposed by Horn and Schunck [2] involved minimizing an energy functional of the following form:

$$E(u, v) = \int_{\Omega} (I_x u + I_y v + I_t)^2 + \lambda (u_x^2 + u_y^2 + v_x^2 + v_y^2) d\mathbf{x}, \quad (1)$$

where I is the image intensity function, $[u(\mathbf{x}, t), v(\mathbf{x}, t)]^T$ is the motion vector to be estimated, subscripts x , y , and t denote the direction in the partial derivatives, $\mathbf{x} = [x, y]^T$ is a point in the spatial domain, Ω represents the 2D image domain and λ is a parameter controlling the degree of smoothness in the flow field. However, the optical flow constraint embedded in Eq. (1) is derived from the brightness constancy model, which suffers from the problem of brightness variation. To account for brightness variations, Negahdaripour [9] proposed a revised definition of optical flow to represent the geometric and radiometric variations in dynamic imagery. This revised definition of optical flow is represented by the vector field $\mathbf{v} = [\delta x, \delta y, \delta I]^T$, where δx and δy comprise a motion vector that represents the geometric transformation in an image sequence and δI is referred to as the *scene brightness variation field* that describes the radiometric variations in dynamic imagery. Compared to the original definition of optical flow, the revised definition is more general and provides a more complete representation of all the information in an image sequence. With this new definition of optical flow, Negahdaripour proposed a generalized dynamic image model (GDIM) [9,29,30] to represent the scene brightness variation field in dynamic imagery as follows:

$$I(\mathbf{r} + \delta \mathbf{r}) - I(\mathbf{r}) = \delta I(\mathbf{r}) = -m(\mathbf{r})I(\mathbf{r}) - c(\mathbf{r}), \quad (2)$$

where $m(\mathbf{r})$ and $c(\mathbf{r})$ denote the multiplier and offset fields of the scene brightness variation field and $\mathbf{r} = [x, y, t]^T$ is a point in a spatiotemporal domain. With GDIM, an extended optical flow constraint can be developed

$$I_x(\mathbf{r})u(\mathbf{r}) + I_y(\mathbf{r})v(\mathbf{r}) + I_t(\mathbf{r}) + m(\mathbf{r})I(\mathbf{r}) + c(\mathbf{r}) = 0. \quad (3)$$

In fact, using only one of the two fields, say the multiplier field, is sufficient to describe the radiometric variations in an image sequence. However, as indicated by Negahdaripour, in such case the multiplier field may vary arbitrarily, thus making it difficult to estimate via employing the smoothness constraint. By using an additional offset field, we can assume these two fields to be smooth even if the scene brightness variation field changes sharply, which is more appropriate for use in a regularization framework. Consequently, we adopt GDIM-based optical flow constraint to construct the energy function in the gradient-based regularization formulation. However, unlike traditional gradient-based methods which minimize residue of the optical flow constraint, we minimize the minimum distance between

the flow vector and the corresponding linear flow constraint by including a normalization term into the extended optical flow constraint.² This normalization term can amend the over-weighting of the flow constraint at the high-gradient and high-intensity locations, where the optical flow constraints may not be reliable [3]. Thus, the proposed energy function to be minimized in our algorithm can be written in a discrete form as follows:

$$f(\mathbf{u}) = \sum_{i \in D} \left(\frac{I_{x,i}u_i + I_{y,i}v_i + I_{t,i} + m_iI_i + c_i}{\sqrt{I_{x,i}^2 + I_{y,i}^2 + I_i^2 + 1}} \right)^2 + \lambda \sum_{i \in D} (u_{x,i}^2 + u_{y,i}^2 + v_{x,i}^2 + v_{y,i}^2) + \mu \sum_{i \in D} (m_{x,i}^2 + m_{y,i}^2 + c_{x,i}^2 + c_{y,i}^2), \quad (4)$$

where the subscript i denotes the i th location, vector \mathbf{u} is the concatenation of all the flow components u_i and v_i and all the brightness variation multiplier and offset fields m_i and c_i , μ is a parameter controlling the degree of smoothness in the brightness multiplier and offset fields, and D is the set of all the discretized locations in the image domain.

2.2. Robust estimation

Robust estimation has been successfully applied in the computation of optical flow to alleviate large errors in the unreliable flow constraints [12,13]. In robust estimation, the square error function is replaced by a ρ -function to suppress the influence of the flow constraints with large residues on the estimated solution. A typical ρ -function commonly used in computer vision is the Lorentzian function [13,31], given as follows:

$$\rho_{\text{LO}}(x) = \log \left(1 + \frac{x^2}{2} \right),$$

where x is the normalized residue of the data constraint. When the ρ -function is used in model fitting, the influence of each data constraint on the solution is characterized by an influence function ψ , which is the derivative of the ρ -function. For the least-squares estimation, the influence function is linear and therefore the data constraints with large residues will have large influence on the solution. However, most data constraints with large residues in the computation of optical flow are unreliable and should be rejected. For robust estimation, the influence function decreases as the magnitude of the residue increases [13], therefore the estimation is less sensitive to the outliers. By applying the robust error function in the optical flow constraint, we can obtain the following new energy function:

² The minimum distance between the point (u_0, v_0, m_0, c_0) and the hyperplane $I_x u + I_y v + I_t + mI + c = 0$ in the four dimensional space (u, v, m, c) is $|I_x u_0 + I_y v_0 + I_t + m_0 I + c_0| / \sqrt{I_x^2 + I_y^2 + I^2 + 1}$.

$$f(\mathbf{u}) = \sum_{i \in D} \rho_{\text{LO}} \left(\frac{I_{x,i}u_i + I_{y,i}v_i + I_{t,i} + m_i I_i + c_i}{\sigma_r \sqrt{I_{x,i}^2 + I_{y,i}^2 + I_i^2 + 1}} \right) + \lambda \sum_{i \in D} (u_{x,i}^2 + u_{y,i}^2 + v_{x,i}^2 + v_{y,i}^2) + \mu \sum_{i \in D} (m_{x,i}^2 + m_{y,i}^2 + c_{x,i}^2 + c_{y,i}^2), \quad (5)$$

where σ_r is an estimate of scale parameter. In this paper, we consider the problem of motion discontinuity using the *dynamic smoothness adjustment scheme* described in the next section, thus the robust estimation does not apply to the smoothness constraint. Although ρ -function can improve robustness and accuracy in estimating optical flow, this energy function turns to nonconvex, thus complicating the minimization algorithm in the optical flow computation. Fortunately, an iterative reweighted least-squares method [31,32] can be used to approximate the solution by minimizing the following weighted least-squares energy function iteratively,

$$f(\mathbf{u}) = \sum_{i \in D} w_i \left(\frac{I_{x,i}u_i + I_{y,i}v_i + I_{t,i} + m_i I_i + c_i}{\sqrt{I_{x,i}^2 + I_{y,i}^2 + I_i^2 + 1}} \right)^2 + \lambda \sum_{i \in D} (u_{x,i}^2 + u_{y,i}^2 + v_{x,i}^2 + v_{y,i}^2) + \mu \sum_{i \in D} (m_{x,i}^2 + m_{y,i}^2 + c_{x,i}^2 + c_{y,i}^2), \quad (6)$$

where w_i is the weight for the i th data constraint. This energy function is still quadratic and convex for fixed w_i , thus allows for applying the original numerical minimization algorithm. In the reweighted least-squares method, the weight w_i is given by $\psi(r_i/\sigma_r)/\sigma_r$, where r_i is the residue of the i th data constraint and σ_r is the scale parameter. For the Lorentzian function used in Eq. (5), the weight w_i is computed by the formula

$$w_i = \frac{2\sigma_r^2}{2\sigma_r^2 + r_i^2}. \quad (7)$$

In this work, the residue r_i is defined as $(I_{x,i}\hat{u}_i + I_{y,i}\hat{v}_i + I_{t,i} + \hat{m}_i I_i + \hat{c}_i) / \sqrt{I_{x,i}^2 + I_{y,i}^2 + I_i^2 + 1}$, where $[\hat{u}_i, \hat{v}_i, \hat{m}_i, \hat{c}_i]^T$ is the current estimated flow vector. The scale parameter σ_r is selected as the standard deviation of r_i .

2.3. Dynamic smoothness adjustment

The conventional regularization approach cannot give correct image motion in the vicinity of motion discontinuities owing to the imposed smoothness constraint. In this paper, we develop a *dynamic smoothness adjustment scheme* to effectively suppress the smoothness constraint at motion boundaries and brightness variation boundaries in the multiplier and offset fields. We control the strength of smoothness constraint by assigning an appropriate weight for each component of the constraint, thus yielding the following energy function:

$$\begin{aligned}
f(\mathbf{u}) = & \sum_{i \in D} w_i \left(\frac{I_{x,i}u_i + I_{y,i}v_i + I_{t,i} + m_i I_i + c_i}{\sqrt{I_{x,i}^2 + I_{y,i}^2 + I_i^2 + 1}} \right)^2 \\
& + \lambda \sum_{i \in D} \left(\alpha_{x,i} u_{x,i}^2 + \alpha_{y,i} u_{y,i}^2 + \beta_{x,i} v_{x,i}^2 + \beta_{y,i} v_{y,i}^2 \right) \\
& + \mu \sum_{i \in D} \left(\gamma_{x,i} m_{x,i}^2 + \gamma_{y,i} m_{y,i}^2 + \delta_{x,i} c_{x,i}^2 + \delta_{y,i} c_{y,i}^2 \right), \tag{8}
\end{aligned}$$

where $\alpha_{x,i}$, $\alpha_{y,i}$, $\beta_{x,i}$, $\beta_{y,i}$, $\gamma_{x,i}$, $\gamma_{y,i}$, $\delta_{x,i}$, and $\delta_{y,i}$ are the weights for the corresponding components of the i th smoothness constraint along x - and y -directions. In this research, we found that the angular deviation between adjacent motion vectors is a good index for locating the motion discontinuities. Hence, we developed the following formulas to determine these weights.

$$r_i = \theta_i - \bar{\theta}, \tag{9a}$$

$$\theta_i = \cos^{-1} \left(\frac{\mathbf{v}_i \cdot \mathbf{v}_{i-1}}{|\mathbf{v}_i| |\mathbf{v}_{i-1}|} \right), \tag{9b}$$

$$\alpha_{x,i} = \beta_{x,i} = \begin{cases} \frac{2\sigma_d^2}{2\sigma_d^2 + r_i^2} & \text{if } r_i > 0 \text{ and } (x_i, y_i) \in \Omega_E, \\ 1 & \text{otherwise,} \end{cases} \tag{9c}$$

where \mathbf{v}_i is defined as $[\hat{u}_i, \hat{v}_i, 1]^T$. The symbol θ_i is the angle between the vector \mathbf{v}_i and its adjacent vector, \mathbf{v}_{i-1} , along x -direction. That is, we take the vector $\mathbf{v}(x, y) = [\hat{u}(x, y), \hat{v}(x, y), 1]^T$ with the index $i = x + yh$, $1 \leq x \leq h$, $1 \leq y \leq w$, where x and y indicate the vertical and horizontal image coordinates, respectively, and w and h denote image width and height. Thus \mathbf{v}_{i-1} is the adjacent vector of \mathbf{v}_i along x -direction and \mathbf{v}_{i-h} is the adjacent vector of \mathbf{v}_i along y -direction. The symbols $\bar{\theta}$ and σ_d are the mean and standard deviation of θ_i over the entire image. The symbol Ω_E denotes the set of all the edge locations in image I .³ The weights $\alpha_{y,i}$ and $\beta_{y,i}$ are determined similarly to those of $\alpha_{x,i}$ and $\beta_{x,i}$ with \mathbf{v}_{i-1} replaced by \mathbf{v}_{i-h} , the adjacent vector along y -direction. We set the same value for $\alpha_{x,i}$ and $\beta_{x,i}$ because both \hat{u}_i and \hat{v}_i form the motion vector at this location together and discontinuity occurs between two adjacent motion vectors, not for separate components of the vectors. In this equation, we only consider positive residue r_i for the reason that the angle θ_i is always positive and the angle θ_i smaller than $\bar{\theta}$ implies that the image motions at these locations of θ_i are very smooth, thus their smoothness constraints should not be suppressed. This is achieved by directly setting the smoothness weights at these locations to 1. The other weights γ and δ also have the same form as those of α and β but with

³ Generally, our algorithm can still produce good performance even if the Ω_E -restriction is removed. This restriction is just utilized to rectify α at the very non-edge locations, thus, in our algorithm, Ω_E is obtained simply through thresholding on the edge strength image.

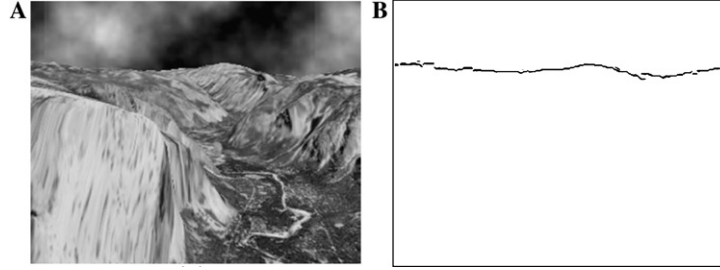


Fig. 1. Computed weights $\alpha_{x,i}$ in our dynamic smoothness adjustment scheme. (A) One frame from *Yosemite* sequence; (B) computed $\alpha_{x,i}$ from the estimated motion field (the value of $\alpha_{x,i}$ is scaled from $[0, 1]$ to $[0, 255]$).

a different definition of θ_i . For determining $\gamma_{x,i}$, $\gamma_{y,i}$, $\delta_{x,i}$ and $\delta_{y,i}$, the values of θ_i are defined as $|\hat{m}_i - \hat{m}_{i-1}|$, $|\hat{m}_i - \hat{m}_{i-h}|$, $|\hat{c}_i - \hat{c}_{i-1}|$, and $|\hat{c}_i - \hat{c}_{i-h}|$, respectively. The formulas in determining these weights imply that the locations with larger residues r_i are more possible to be motion boundaries or brightness variation boundaries in the multiplier and offset fields, thus the weights for the smoothness constraints at these locations should be small. One example of the detected motion boundaries is illustrated in Fig. 1, where the well-known *Yosemite* sequence is shown. For this image sequence, there is a motion boundary between the sky and mountain regions and our dynamic smoothness adjustment scheme can correctly locate this motion boundary as shown in Fig. 1B.

2.4. Constraint refinement

The optical flow constraint is derived under the assumption that the motion vectors are sufficiently small, thus the higher order terms in the Taylor series expansion can be neglected. This approximation degrades the performance of the regularization method for large image motions. In this work, we refine the optical flow estimation by updating the optical flow constraints from the current estimated motion vectors, thus reducing the approximation error. This constraint refinement scheme allows accommodating larger image motions in the optical flow computation.

Suppose $[\hat{u}, \hat{v}]^T$ is the current estimated motion vector, then the true motion vector can be expressed as $[u, v]^T = [\hat{u} + \Delta u, \hat{v} + \Delta v]^T$, where $[\Delta u, \Delta v]^T$ is a residual vector to be estimated. Based on this expression, the refined optical flow constraint equation can be obtained by Taylor approximation at the updated location $(x + \hat{u}, y + \hat{v})$ instead of (x, y) . This new optical flow constraint equation is given by

$$I_x(x + \hat{u}, y + \hat{v}, t + 1)\Delta u + I_y(x + \hat{u}, y + \hat{v}, t + 1)\Delta v + I(x + \hat{u}, y + \hat{v}, t + 1) - I(x, y, t) + mI(x, y, t) + c = 0. \quad (10)$$

In general, we can expect that the magnitudes of Δu and Δv are smaller than those of the true flow vector u and v as long as the estimated \hat{u} and \hat{v} are close to u and v .

Smaller Δu and Δv implies that the errors introduced by the first-order approximation are also smaller than those of the original one. Hence, Eq. (10) is a more accurate optical flow constraint, which leads to the following modified energy function:

$$\begin{aligned} f(\mathbf{u}_c) = & \sum_{i \in D} \frac{w_i}{N_i} [I_x(x_i + \hat{u}_i, y_i + \hat{v}_i, t + 1) \Delta u_i + I_y(x_i + \hat{u}_i, y_i + \hat{v}_i, t + 1) \Delta v_i \\ & + I(x_i + \hat{u}_i, y_i + \hat{v}_i, t + 1) - I(x_i, y_i, t) + m_i I(x_i, y_i, t) + c_i]^2 \\ & + \lambda \sum_{i \in D} [\alpha_{x,i} (\hat{u}_{x,i} + \Delta u_{x,i})^2 + \alpha_{y,i} (\hat{u}_{y,i} + \Delta u_{y,i})^2 + \beta_{x,i} (\hat{v}_{x,i} + \Delta v_{x,i})^2 \\ & + \beta_{y,i} (\hat{v}_{y,i} + \Delta v_{y,i})^2] + \mu \sum_{i \in D} [\gamma_{x,i} m_{x,i}^2 + \gamma_{y,i} m_{y,i}^2 + \delta_{x,i} c_{x,i}^2 + \delta_{y,i} c_{y,i}^2], \quad (11) \end{aligned}$$

where \mathbf{u}_c is the concatenation of all the unknown parameters Δu_i , Δv_i , m_i , and c_i , over all the discretized image locations and N_i is the normalization term defined as $I_x(x_i + \hat{u}_i, y_i + \hat{v}_i, t + 1)^2 + I_y(x_i + \hat{u}_i, y_i + \hat{v}_i, t + 1)^2 + I(x_i, y_i, t)^2 + 1$. This energy function is quite different from Eq. (8) for the optical flow constraint is derived at different locations. However, it can be minimized using the same numerical algorithm with some appropriate modifications as described in the next section.

3. Numerical minimization algorithm

The energy function given by Eq. (8) can be rewritten in a matrix–vector form as $f(\mathbf{u}) = \mathbf{u}^T \mathbf{K} \mathbf{u} - 2\mathbf{u}^T \mathbf{b} + \mathbf{c}$, thus minimizing this quadratic and convex function is equivalent to solving a large linear system $\mathbf{K} \mathbf{u} = \mathbf{b}$. The vector $\mathbf{b} \in \Re^{4wh}$ (w and h denote image width and height), which comes from the coefficients of the first-order term in \mathbf{u} , consists of four subvectors

$$\mathbf{b} = \begin{bmatrix} -\mathbf{e}_{xt} \\ -\mathbf{e}_{yt} \\ -\mathbf{e}_{lt} \\ -\mathbf{e}_t \end{bmatrix}, \quad (12)$$

where \mathbf{e}_{xt} , \mathbf{e}_{yt} , \mathbf{e}_{lt} , and \mathbf{e}_t are all wh -dimensional vectors with entries $w_i I_{x,i} I_{t,i} / N_i$, $w_i I_{y,i} I_{t,i} / N_i$, $w_i I_{t,i} I_{t,i} / N_i$, and $w_i I_{t,i} / N_i$, respectively. $N_i = I_{x,i}^2 + I_{y,i}^2 + I_{t,i}^2 + 1$ is the normalization term and w_i is the weight given by Eq. (7). The matrix $\mathbf{K} \in \Re^{4wh \times 4wh}$, which is induced from the coefficients of the second-order term in \mathbf{u} , is a symmetric positive-definite (SPD) matrix taking the following 4×4 block structure.

$$\mathbf{K} = \begin{bmatrix} \lambda \mathbf{K}_{s,z} + \mathbf{E}_{xx} & \mathbf{E}_{xy} & \mathbf{E}_{xl} & \mathbf{E}_x \\ \mathbf{E}_{xy} & \lambda \mathbf{K}_{s,\beta} + \mathbf{E}_{yy} & \mathbf{E}_{yl} & \mathbf{E}_y \\ \mathbf{E}_{xl} & \mathbf{E}_{yl} & \mu \mathbf{K}_{s,\gamma} + \mathbf{E}_{ll} & \mathbf{E}_l \\ \mathbf{E}_x & \mathbf{E}_y & \mathbf{E}_l & \mu \mathbf{K}_{s,\delta} + \mathbf{E} \end{bmatrix}, \quad (13)$$

where E_{xx} , E_{xy} , E_{xI} , E_x , E_{yy} , E_{yI} , E_y , E_{II} , E_I , and E , which arise from the optical flow constraint, are all $wh \times wh$ diagonal matrices with diagonal entries $w_i I_{x,i}^2 / N_i$, $w_i I_{x,i} I_{y,i} / N_i$, $w_i I_{x,i} I_i / N_i$, $w_i I_{x,i} / N_i$, $w_i I_{y,i}^2 / N_i$, $w_i I_{y,i} I_i / N_i$, $w_i I_{y,i} / N_i$, $w_i I_i^2 / N_i$, $w_i I_i / N_i$, and w_i / N_i , respectively. In this paper, we approximate the partial derivatives of the flow field using forward difference, thus the matrix $K_{s,\alpha}$, which comes from the smoothness constraint, takes the following structure:

$$K_{s,\alpha} = \begin{bmatrix} \mathbf{G}_1 & \mathbf{H}_1 & & & \\ \mathbf{H}_1 & \mathbf{G}_2 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \mathbf{H}_{w-1} \\ & & & & \mathbf{H}_{w-1} & \mathbf{G}_w \end{bmatrix} \in \Re^{wh \times wh}, \quad (14)$$

where the submatrices \mathbf{G}_i and \mathbf{H}_i are all of size $h \times h$. The matrix \mathbf{H}_i is diagonal matrix with diagonal entries $-\alpha_{y,ih+j}$, $1 \leq j \leq h$. Matrix \mathbf{G}_i is symmetric tridiagonal matrix with diagonal entries $g_{i,j}$ and off-diagonal entries $l_{i,j}$, which are given by the following formulas:

$$g_{1,1} = \alpha_{x,2} + \alpha_{y,1+h}, \quad (15a)$$

$$g_{1,j} = \alpha_{x,j} + \alpha_{x,j+1} + \alpha_{y,j+h}, \quad 2 \leq j \leq h-1, \quad (15b)$$

$$g_{1,h} = \alpha_{x,h} + \alpha_{y,h+h}, \quad (15c)$$

$$g_{i,1} = \alpha_{y,(i-1)h+1} + \alpha_{x,(i-1)h+2} + \alpha_{y,(i-1)h+1+h}, \quad 2 \leq i \leq w-1, \quad (15d)$$

$$g_{i,j} = \alpha_{x,(i-1)h+j} + \alpha_{y,(i-1)h+j} + \alpha_{x,(i-1)h+j+1} + \alpha_{y,(i-1)h+j+h}, \\ 2 \leq j \leq h-1, \quad 2 \leq i \leq w-1, \quad (15e)$$

$$g_{i,h} = \alpha_{x,(i-1)h+h} + \alpha_{y,(i-1)h+h} + \alpha_{y,(i-1)h+h+h}, \quad 2 \leq i \leq w-1, \quad (15f)$$

$$g_{w,1} = \alpha_{y,(w-1)h+1} + \alpha_{x,(w-1)h+2}, \quad (15g)$$

$$g_{w,j} = \alpha_{x,(w-1)h+j} + \alpha_{y,(w-1)h+j} + \alpha_{x,(w-1)h+j+1}, \quad 2 \leq j \leq h-1, \quad (15h)$$

$$g_{w,h} = \alpha_{x,(w-1)h+h} + \alpha_{y,(w-1)h+h}, \quad (15i)$$

$$l_{i,j} = -\alpha_{x,(i-1)h+j+1}, \quad 1 \leq i \leq w, \quad 1 \leq j \leq h-1. \quad (15j)$$

$K_{s,\beta}$, $K_{s,\gamma}$, and $K_{s,\delta}$ can be obtained via Eq. (15a)–(15j) with α replaced by β , γ , and δ , respectively.

Solving the linear system $\mathbf{K}\mathbf{u} = \mathbf{b}$ provides an initial estimation of the optical flow. By exploiting this initial estimation, the optical flow constraint can be further refined, thus leading to the energy function given by Eq. (11). Minimizing Eq. (11) is equivalent to solving another linear system $\mathbf{K}_c\mathbf{u}_c = \mathbf{b}_c$. The matrix \mathbf{K}_c has the same structure as \mathbf{K} but with $I_{x,i}$, $I_{y,i}$, and $I_{t,i}$ in the entries of \mathbf{K} replaced by $I_x(x_i + \hat{u}_i, y + \hat{v}_i, t + 1)$, $I_y(x_i + \hat{u}_i, y + \hat{v}_i, t + 1)$, and $I(x_i + \hat{u}_i, y + \hat{v}_i, t + 1) - I(x_i, y_i, t)$, respectively. These values involve the image intensities at non-integer coordinates and are estimated via bilinear interpolation in our implementation. The vector \mathbf{b}_c takes the following block structure:

$$\mathbf{b}_c = \begin{bmatrix} -\tilde{\mathbf{e}}_{xt} + \lambda(\mathbf{s}_{\hat{u}_x} + \mathbf{s}_{\hat{u}_y}) \\ -\tilde{\mathbf{e}}_{yt} + \lambda(\mathbf{s}_{\hat{v}_x} + \mathbf{s}_{\hat{v}_y}) \\ -\tilde{\mathbf{e}}_{It} \\ -\tilde{\mathbf{e}}_t \end{bmatrix} \in \Re^{4wh}, \quad (16)$$

where $\tilde{\mathbf{e}}_{xt}$, $\tilde{\mathbf{e}}_{yt}$, $\tilde{\mathbf{e}}_{It}$, and $\tilde{\mathbf{e}}_t$ are obtained through modifying the entries in \mathbf{e}_{xt} , \mathbf{e}_{yt} , \mathbf{e}_{It} , and \mathbf{e}_t in \mathbf{b} via the same rule as the formation of \mathbf{K}_c , i.e., $I_{x,i}$, $I_{y,i}$, and $I_{t,i}$ are replaced by $I_x(x_i + \hat{u}_i, y + \hat{v}_i, t + 1)$, $I_y(x_i + \hat{u}_i, y + \hat{v}_i, t + 1)$ and $I(x_i + \hat{u}_i, y + \hat{v}_i, t + 1) - I(x_i, y_i, t)$. The vectors $\mathbf{s}_{\hat{u}_x}$, $\mathbf{s}_{\hat{u}_y}$, $\mathbf{s}_{\hat{v}_x}$, and $\mathbf{s}_{\hat{v}_y}$ are induced from the current estimated motion vector $[\hat{u}, \hat{v}]^T$ in the smoothness constraint and are given as follows:

$$\mathbf{s}_{\hat{u}_{x,jh+1}} = \alpha_{x,jh+2}(\hat{u}_{jh+2} - \hat{u}_{jh+1}), \quad 0 \leq j \leq w-1, \quad (17a)$$

$$\begin{aligned} \mathbf{s}_{\hat{u}_{x,jh+k}} &= \alpha_{x,jh+k+1}(\hat{u}_{jh+k+1} - \hat{u}_{jh+k}) - \alpha_{x,jh+k}(\hat{u}_{jh+k} - \hat{u}_{jh+k-1}), \\ 2 \leq k \leq h-1, \quad 0 \leq j \leq w-1, \end{aligned} \quad (17b)$$

$$\mathbf{s}_{\hat{u}_{x,jh+h}} = -\alpha_{x,jh+h}(\hat{u}_{jh+h} - \hat{u}_{jh+h-1}), \quad 0 \leq j \leq w-1, \quad (17c)$$

$$\mathbf{s}_{\hat{u}_{y,k}} = \alpha_{y,k+h}(\hat{u}_{k+h} - \hat{u}_k), \quad 1 \leq k \leq h, \quad (17d)$$

$$\begin{aligned} \mathbf{s}_{\hat{u}_{y,jh+k}} &= \alpha_{y,jh+k+h}(\hat{u}_{jh+k+h} - \hat{u}_{jh+k}) - \alpha_{y,jh+k}(\hat{u}_{jh+k} - \hat{u}_{jh+k-h}), \\ 1 \leq k \leq h, \quad 1 \leq j \leq w-1, \end{aligned} \quad (17e)$$

$$\mathbf{s}_{\hat{u}_{y,(w-1)h+k}} = -\alpha_{y,(w-1)h+k}(\hat{u}_{(w-1)h+k} - \hat{u}_{(w-1)h+k-h}), \quad 1 \leq k \leq h. \quad (17f)$$

The vectors $\mathbf{s}_{\hat{v}_x}$ and $\mathbf{s}_{\hat{v}_y}$ are obtained via Eq. (17a)–(17f) by replacing \hat{u}_i , $\alpha_{x,i}$, and $\alpha_{y,i}$ with \hat{v}_i , $\beta_{x,i}$, and $\beta_{y,i}$, respectively.

In this paper, we apply the preconditioned conjugate gradient algorithm with an incomplete Cholesky preconditioner \mathbf{P} [3,33] to solve this linear system efficiently. The detailed steps of this algorithm (ICPCG) are 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 $\beta_k = \mathbf{r}_{k-1}^T \mathbf{z}_{k-1} / \mathbf{r}_{k-2}^T \mathbf{z}_{k-2}$, and update $\mathbf{p}_k = \mathbf{z}_{k-1} + \beta_k \mathbf{p}_{k-1}$.
- (4) Compute $\alpha_k = \mathbf{r}_{k-1}^T \mathbf{z}_{k-1} / \mathbf{p}_k^T \mathbf{K} \mathbf{p}_k$.
- (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 $\mathbf{r}_k \approx \mathbf{0}$, stop; else go to step 2.

To accelerate the convergence speed of the conjugate gradient algorithm, the preconditioner \mathbf{P} must be carefully designed [3]. It must be a good approximation to \mathbf{K} so that the condition number of the preconditioned linear system is dramatically reduced. In addition, there must exist a numerical method to efficiently solve the linear system $\mathbf{P}\mathbf{z} = \mathbf{r}$ required in the preconditioned conjugate gradient algorithm. Therefore, the incomplete Cholesky factorization is a good choice. That is $\mathbf{P} = \mathbf{L}\mathbf{L}^T \approx \mathbf{K}$. To approximate the matrix \mathbf{K} , \mathbf{L} is designed to possess similar sparse structure as \mathbf{K} , thus leading to the following form:

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} & \mathbf{0} & \mathbf{0} \\ \mathbf{L}_{31} & \mathbf{L}_{32} & \mathbf{L}_{33} & \mathbf{0} \\ \mathbf{L}_{41} & \mathbf{L}_{42} & \mathbf{L}_{43} & \mathbf{L}_{44} \end{bmatrix} \in \Re^{4wh \times 4wh}, \quad (18)$$

where the submatrices \mathbf{L}_{ij} are all of size $wh \times wh$ and take the following structure:

$$\mathbf{L}_{ij} = \begin{bmatrix} \mathbf{G}_{ij}^{(1)} & & & \\ \mathbf{H}_{ij}^{(1)} & \mathbf{G}_{ij}^{(2)} & & \\ & \ddots & \ddots & \\ & & \mathbf{H}_{ij}^{(w-1)} & \mathbf{G}_{ij}^{(w)} \end{bmatrix}, \quad (19)$$

for $i = j$ and

$$\mathbf{L}_{ij} = \begin{bmatrix} \mathbf{G}_{ij}^{(1)T} & \mathbf{H}_{ij}^{(1)T} & & \\ & \mathbf{G}_{ij}^{(2)T} & \ddots & \\ & & \ddots & \mathbf{H}_{ij}^{(w-1)T} \\ & & & \mathbf{G}_{ij}^{(w)T} \end{bmatrix} \quad (20)$$

for $i \neq j$. Matrices $\mathbf{G}_{ij}^{(k)}$ and $\mathbf{H}_{ij}^{(k)}$ are given by

$$\mathbf{G}_{ij}^{(k)} = \begin{bmatrix} a_{ij,1}^{(k)} & & & \\ b_{ij,1}^{(k)} & a_{ij,2}^{(k)} & & \\ & \ddots & \ddots & \\ & & b_{ij,h-1}^{(k)} & a_{ij,h}^{(k)} \end{bmatrix} \in \Re^{h \times h}, \quad (21)$$

$$\mathbf{H}_{ij}^{(k)} = \begin{bmatrix} c_{ij,1}^{(k)} & d_{ij,1}^{(k)} & & \\ & c_{ij,2}^{(k)} & \ddots & \\ & & \ddots & d_{ij,h-1}^{(k)} \\ & & & c_{ij,h}^{(k)} \end{bmatrix} \in \Re^{h \times h}. \quad (22)$$

The nonzero entries in the matrices $\mathbf{G}_{ij}^{(k)}$ and $\mathbf{H}_{ij}^{(k)}$ are obtained 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} . The formulas for determining these entries in \mathbf{L}_{ii} , $i = 1, \dots, 4$ are given by

$$a_{ii,l}^{(k)} = \begin{cases} \sqrt{A_{p,p} - (b_{ii,l-1}^{(k)})^2}, & k = 1, \quad 1 \leq l \leq h, \\ \sqrt{A_{p,p} - (b_{ii,l-1}^{(k)})^2 - (d_{ii,l}^{(k-1)})^2 - (c_{ii,l}^{(k-1)})^2}, & 2 \leq k \leq w, \quad 1 \leq l \leq h, \end{cases} \quad (23)$$

$$b_{ii,l}^{(k)} = \begin{cases} 0, & l = 0, h, \quad 1 \leq k \leq w, \\ A_{p,p-1}/a_{ii,l}^{(k)}, & k = 1, \quad 1 \leq l \leq h-1, \\ (A_{p,p-1} - c_{ii,l}^{(k-1)}d_{ii,l-1}^{(k-1)})/a_{ii,l}^{(k)}, & 2 \leq k \leq w, \quad 1 \leq l \leq h-1, \end{cases} \quad (24)$$

$$c_{ii,l}^{(k)} = A_{p,p-h}/a_{ii,l}^{(k)}, \quad 1 \leq k \leq w-1, \quad 1 \leq l \leq h, \quad (25)$$

$$d_{ii,l}^{(k)} = \begin{cases} 0, & l = 0, h, \quad 1 \leq k \leq w-1, \\ -b_{ii,l}^{(k)}c_{ii,l}^{(k)}/a_{ii,l+1}^{(k)}, & 1 \leq l \leq h-1, \quad 1 \leq k \leq w-1, \end{cases} \quad (26)$$

where $p = (k-1)h + l$, $\mathbf{A} = \mathbf{K}_{ii} - \sum_{j=1}^{i-1} \mathbf{L}_{ij}\mathbf{L}_{ij}^T$, $\mathbf{K}_{ij} \in \Re^{wh \times wh}$ is the submatrix of \mathbf{K} at block row i and block column j . Similarly, the nonzero entries in \mathbf{L}_{ij} , $1 \leq j < i \leq 4$ can be written as follows:

$$a_{ij,l}^{(k)} = \mathbf{B}_p/a_{jj,l}^{(k)}, \quad 1 \leq k \leq w, \quad 1 \leq l \leq h, \quad (27)$$

$$b_{ij,l}^{(k)} = \begin{cases} -a_{ij,l}^{(k)}b_{jj,l}^{(k)}/a_{jj,l+1}^{(k)}, & 1 \leq k \leq w, \quad 1 \leq l \leq h-1, \\ 0, & 1 \leq k \leq w, \quad l = h, \end{cases} \quad (28)$$

$$d_{ij,l}^{(k)} = -a_{ij,l+1}^{(k)}d_{jj,l}^{(k)}/a_{jj,l}^{(k+1)}, \quad 1 \leq k \leq w-1, \quad 1 \leq l \leq h-1, \quad (29)$$

$$c_{ij,l}^{(k)} = -\left(a_{ij,l}^{(k)}c_{jj,l}^{(k)} + b_{ij,l}^{(k)}d_{jj,l}^{(k)} + d_{ij,l-1}^{(k)}b_{jj,l-1}^{(k+1)}\right)/a_{jj,l}^{(k+1)}, \quad 1 \leq k \leq w-1, \quad 1 \leq l \leq h, \quad (30)$$

where $\mathbf{B} = \mathbf{K}_{ij} - \sum_{n=1}^{j-1} \mathbf{L}_{jn}\mathbf{L}_{jn}^T$.

In this numerical minimization algorithm, the parameters $w_{i,b}$, $\alpha_{x,i,b}$, $\alpha_{y,i,b}$, $\beta_{x,i,b}$, $\beta_{y,i,b}$, $\gamma_{x,i,b}$, $\gamma_{y,i,b}$, $\delta_{x,i,b}$ and $\delta_{y,i,b}$ are initially set to 1 and then updated after a fixed number of the incomplete Cholesky preconditioned conjugate gradient (ICPCG) iterations. Combining this weight updating method with our constraint refinement scheme, the final algorithm is listed as follows:

- (1) Set $w_{i,b}$, $\alpha_{x,i,b}$, $\alpha_{y,i,b}$, $\beta_{x,i,b}$, $\beta_{y,i,b}$, $\gamma_{x,i,b}$, $\gamma_{y,i,b}$, $\delta_{x,i,b}$ and $\delta_{y,i,b}$ to 1; $\mathbf{u}_0 = \mathbf{0}$; $k = 0$; $l = 0$.
- (2) if $l = 0$, calculate \mathbf{K} and \mathbf{b} according to Eqs. (12) and (13); else compute \mathbf{K}_c and \mathbf{b}_c .
- (3) Apply ICPCG for n iterations by using \mathbf{u}_0 as the initial vector to obtain the solution $\hat{\mathbf{u}}$.
- (4) Update $w_{i,b}$, $\alpha_{x,i,b}$, $\alpha_{y,i,b}$, $\beta_{x,i,b}$, $\beta_{y,i,b}$, $\gamma_{x,i,b}$, $\gamma_{y,i,b}$, $\delta_{x,i,b}$ and $\delta_{y,i,b}$; $k = k + 1$; $\mathbf{u}_0 = \hat{\mathbf{u}}$.
- (5) If $k < N$, goto step 2.
- (6) If $l = M$, stop; else $k = 0$, $l = l + 1$, set w_i to 1, $\mathbf{u}_0 = \mathbf{0}$, go to step 2.

In this algorithm, \mathbf{u}_0 is the initial solution of the ICPCG algorithm. For fixed $w_{i,b}$, $\alpha_{x,i,b}$, $\alpha_{y,i,b}$, $\beta_{x,i,b}$, $\beta_{y,i,b}$, $\gamma_{x,i,b}$, $\gamma_{y,i,b}$, $\delta_{x,i,b}$ and $\delta_{y,i,b}$, our energy function is quadratic and convex, thus the ICPCG algorithm converges to a unique solution no matter what the initial solution \mathbf{u}_0 is. However, a good initial solution can significantly reduce the total number of iterations required to converge to the final solution, thus we set \mathbf{u}_0 to the previous estimated solution $\hat{\mathbf{u}}$ at step (4) to save computing time. Note that the solution obtained from this algorithm is $[\Delta u_i, \Delta v_i, m_i, c_i]^T$. The final solution should be $[\hat{u}_i + \Delta u_i, \hat{v}_i + \Delta v_i, m_i, c_i]^T$, where $[\hat{u}_i, \hat{v}_i]^T$ is the previous estimated motion vector.

4. Experimental results

In this section, the detailed procedures in our experiments are described and the results of testing our algorithm on a variety of synthetic and real image sequences are presented. In our implementation, the partial derivatives I_x , I_y , and I_t are estimated via averaging four finite differences as used by Horn and Schunck [2]. Other more accurate and sophisticated numerical approximation strategies can also be applied in our implementation and the results can be further improved. However, more accurate approximation strategies often require more image frames and more computing effort (thus, more computing time), especially for our constraint refinement scheme where the partial derivatives are estimated at non-integer coordinates. To eliminate image noise in estimating these partial derivatives, we pre-smooth the image intensity function using the Gaussian filter with standard deviation equal to 1.5. In addition, we implicitly normalize the intensity value I_i that is related to the component m_i in Eq. (4) by a constant 65 to match the scale order of $I_{x,i}$, $I_{y,i}$, and $I_{t,i}$, thus reducing the ill-conditioning of the matrix \mathbf{K} . The regularization parameters λ and μ take the same values in all of our experiments and their values are ranged from 0.5 to 2.5. Although λ and μ are slightly differed for several experiments in this paper, the results are generally insensitive to these parameters in this range.

To evaluate the performance of our algorithm, we adopt the angular error measure [1] between estimated motion field and the true motion field as our performance

index. This error measure can handle large and very small motion vectors without amplification inherent in relative measure of vector difference [1]. Because of its convenience, the angular error measure is widely adopted by many researchers in optical flow computation (e.g. [1,3,8,12,13,16,22,34–37]). Therefore, it provides a good basis for comparison with other techniques reported in literature. Unlike some other methods that produce motion fields only at sparse positions, our algorithm provides optical flow estimation with 100% density. As mentioned in Section 1, the scene brightness variation field is important in some applications, so its accuracy should also be evaluated. However, evaluating the true brightness variation is more difficult because of the non-integer motion fields. Hence, we designed the *Translating Yosemite* sequence with integer motion field to estimate the accuracy of the scene brightness variation field. For other sequences, only the distribution of the scene brightness variation field is displayed via a gray level image.

In our implementation, we update the weights for flow constraint and smoothness constraint every 10 iterations of incomplete Cholesky preconditioned conjugate gradient algorithm and apply the constraint refinement scheme after five updates of these weights ($n = 10, N = 5$ in our final algorithm). We do not set a self-stopping criterion in our algorithm and the results shown in this section are all obtained after 100 preconditioned conjugate gradient algorithm iterations ($M = 2$ in our final algorithm), unless otherwise stated explicitly. Generally, increasing the number of iterations leads to more accurate optical flow but requires more computing time. This is a trade-off between performance and efficiency, which can be determined by the user.

4.1. Results for synthetic image sequences

We test our algorithm by the following synthetic image sequences: *Yosemite*, *Translating/Rotating Yosemite*, and *Translating/Diverging Tree*. Frames from these sequences are displayed in Fig. 2. The image resolution of *Yosemite* sequence is 316×252 pixels. This realistic sequence is very challenging owing to that it contains brightness variations in the sky region, motion discontinuities between sky and mountain regions, large image motions and severe image aliasing at the lower-left corner of the image. In this sequence, the clouds translate to the right with a speed of 2 pixels/frame while the rest of the flow is divergent, with speed of about 5 pixels/frame in the lower left corner.

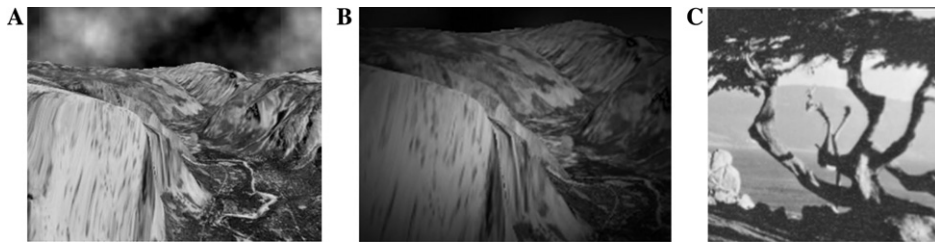


Fig. 2. Frames from: (A) *Yosemite*, (B) *Rotating/Translating Yosemite*, and (C) *Diverging/Translating Tree* image sequences.

*Translating/Rotating Yosemite*⁴ sequences are formed by translating/rotating the center portion of the first frame of *Yosemite* sequence. The *Translating Yosemite* is synthesized by translating the image with a velocity 1 pixel/frame to the right while the *Rotating Yosemite* is obtained by rotating the image clockwise around its center with an angular velocity 0.8 deg/frame. The image sizes of these two sequences are both 200×150 pixels. To simulate brightness variation, a non-uniform illumination should be imposed on these two sequences. Because the reflected diffuse light from object surface is the multiplication of incident light and surface reflectivity, thus we adopt the same equation as in [8] to simulate the spatially and temporally non-uniform illumination. This equation is given in the following:

$$I(x, y, t) = r_1(x, y)r_2(t)f(x, y, t), \quad (31)$$

where $r_1(x, y) = \exp\{-(x - 100)^2/100^2 - (y - 75)^2/75^2\}$ and $r_2(t) = \sin(\pi t/50)$ represent the effect of spatially and temporally non-uniform illumination, respectively. The function $r_1(x, y)$ can be seen as a spatially varying light factor while the function $r_2(t)$ denotes the time-varying global light factor. The function $f(x, y, t)$ is the intensity of the original translating/rotating *Yosemite* sequence without non-uniform illumination, and the function $I(x, y, t)$ is the intensity of the synthesized sequence to be tested in our experiments. Obviously, this non-uniform illumination model is simple for it considers only the diffuse component of the reflected light. Nevertheless, these two sequences are good examples for testing the performance of our algorithm under non-uniform illuminations.

The final two synthetic sequences to be tested are the *Translating/Diverging Tree* sequences. These two sequences were also widely tested by many reported techniques (e.g. [1,3,12,16,34–37]). These two sequences simulate translational camera motion with respect to a textured planar surface. For *Translating Tree* sequence, the camera moves along the horizontal axis producing the motion vectors with speed ranged from 1.73 to 2.26 pixels/frame. For *Diverging Tree* sequence, the camera moves along its line of sight and the image speeds vary from 1.29 pixels/frame on the left side to 1.86 pixels/frame on the right. There are no brightness variations and motion discontinuities in these two sequences, so they can be employed to test the performance of our algorithm under these ideal situations. The image resolutions of these two image sequences are both 150×150 pixels.

The theoretical motion fields of these synthetic image sequences are depicted in Fig. 3. To demonstrate the effectiveness of our algorithm, the angular errors of these synthetic sequences for each auxiliary scheme of our algorithm are listed in Table 1. By considering the average angular error and the standard deviation simultaneously, this table indicates that each auxiliary scheme of our algorithm can effectively improve the accuracy of estimated motion fields excluding some exceptions. These exceptions occur at our dynamic smoothness adjustment scheme where it produces slightly worse performance for *Translating/Rotating Yosemite* and *Translating/Diverging Tree* sequences. This phenomenon originates from the fact that there are no motion discontinuities in these four sequences, thus the original smoothness

⁴ The *Rotating Yosemite* sequence is synthesized in the same manner as that in [8].

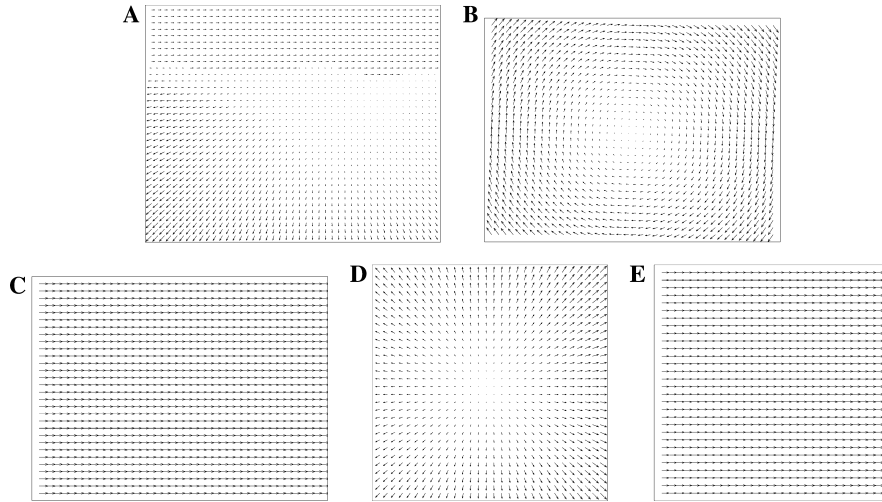


Fig. 3. Theoretical motion fields of: (A) *Yosemite*, (B) *Rotating Yosemite*, (C) *Translating Yosemite*, (D) *Diverging Tree*, and (E) *Translating Tree*.

Table 1

The results for each auxiliary estimation scheme imposed

Image sequence		GR ^a	GRR ^b	GRRD ^c	GRRDC ^d
<i>Yosemite</i>	Average error (deg)	3.96	4.04	2.77	2.70
	SD (deg)	7.63	7.45	5.19	5.20
<i>Translating Yosemite</i> ^e	Average error (deg)	2.25	2.17	2.42	1.26
	SD (deg)	0.93	0.92	1.45	1.07
<i>Rotating Yosemite</i> ^e	Average error (deg)	3.04	2.97	3.04	2.14
	SD (deg)	1.34	1.33	1.26	1.50
<i>Translating Tree</i>	Average error (deg)	0.56	0.51	0.52	0.21
	SD (deg)	0.31	0.25	0.26	0.16
<i>Diverging Tree</i>	Average error (deg)	1.28	1.23	1.52	1.45
	SD (deg)	0.96	0.85	1.16	0.93

^a GR: proposed gradient-based regularization method.

^b GRR: GR + robust estimation scheme.

^c GRRD: GRR + dynamic smoothness adjustment.

^d GRRDC: GRRD + constraint refinement scheme.

^e For these two sequences, we update the weights for flow and smoothness constraints every 20 iterations ($n = 20$) and thus the final results (GRRDC) are obtained after 200 iterations.

assumption (i.e., without the dynamic smoothness adjustment scheme) is perfect for these sequences. The use of our dynamic smoothness adjustment scheme slightly relaxes the smoothness assumption at certain locations, thereby giving less accurate motion fields for these sequences with very smooth optical flow fields. Although it provides slightly degraded accuracy for these sequences, the dynamic smoothness

adjustment scheme is necessary to account for the motion boundaries in general optical flow computation, for example, the *Yosemite* sequence. The estimated motion fields of these sequences by utilizing our algorithm are illustrated in Fig. 4. Obviously, for the *Yosemite* sequence, our algorithm produces accurate image motion in the sky region and provides clear motion boundary between the sky and mountain region. For other sequences, our method gives almost the same motion fields as the true ones shown in Fig. 3.

To demonstrate the excellent performance of our algorithm, our results on the *Yosemite* and *Translating/Diverging Tree* sequences as well as previously reported results in literature (e.g. [1,3,8,12,16,22,34–36]) are listed in Tables 2–4. For *Yosemite* and *Translating Tree* sequences, our algorithm compares favorably with other techniques, even though some of these methods yield only sparse motion fields, where our algorithm produces flows with 100% density. Among the approaches which give dense motion fields, our algorithm obviously outperforms other techniques on these sequences. For the *Diverging Tree* sequence, only Lai and Vemuri's regularization has a better performance than ours. As explained above, the slightly degraded performance of our algorithm is due to the inclusion of our dynamic smoothness adjustment scheme, which slightly relaxes the smoothness constraint. By removing this auxiliary estimation scheme, our algorithm, the GRR shown in Table 4, still has the best performance among the techniques which produce motion fields with 100% density. Theoretically, the sky region of *Yosemite* sequence was synthesized using fractal techniques, thus the usual intensity derivatives method should not work well in this region. This is why many researchers reported their results only on the sky-excluded *Yosemite* sequence (e.g. [3,13,38–42]). However, our method still produces accurate optical flow estimation in this region owing to the generalized dynamic image model and the imposed smoothness constraint. To compare with

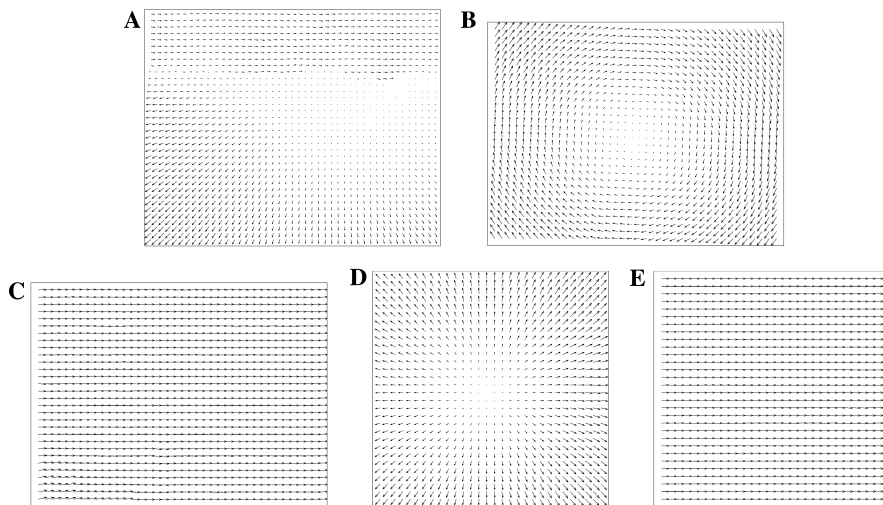


Fig. 4. Estimated motion fields of: (A) *Yosemite*, (B) *Rotating Yosemite*, (C) *Translating Yosemite*, (D) *Diverging Tree*, and (E) *Translating Tree*.

Table 2
Summary of *Yosemite* results (with sky)

Technique	Average error (deg)	SD (deg)	Density (%)
Horn and Schunck (modified)	9.78	16.19	100
Lucas and Kanade ($\lambda_2 \geq 1.0$)	4.28	11.41	35.1
Uras et al. (unthresholded)	8.94	15.61	100
Nagel	10.22	16.51	100
Anandan (unthresholded)	13.36	15.64	100
Singh (step 2)	10.03	13.13	97.7
Heeger (level 1)	9.87	14.74	15.2
Waxman et al. ($\sigma_f = 2.0$)	20.05	23.23	7.4
Fleet and Jepson ($\tau = 2.5$)	4.63	13.42	34.1
Weber and Malik	4.31	8.66	64.2
Zhang et al.	5.59	11.24	100
Alvarez et al.	5.53	7.40	100
Ong and Spann	5.79	12.55	89.9
Odobez and Bouthemy	6.17	12.61	98.1
Kumar et al.	4.04	9.50	34.08
Srinivasan and Chellappa	8.94	10.63	100
Lai and Vemuri (gradient-based)	7.81	14.57	100
Farnebäck (affine) (from [36])	4.84	10.15	100
Farnebäck (constant) (from [36])	6.93	13.51	100
Liu et al. [35]	7.52	13.72	100
Liu et al. [36]	6.84	12.88	100
Gibson and Spann	6.05	13.82	100
Proposed method (GRRDC)	2.70	5.20	100

Table 3
Summary of *Translating Tree* results

Technique	Average error (deg)	SD (deg)	Density (%)
Horn and Schunck (modified)	2.02	2.27	100
Lucas and Kanade ($\lambda_2 \geq 5.0$)	0.56	0.58	13.1
Uras et al. (unthreshold)	0.62	0.52	100
Nagel	2.44	3.06	100
Anandan	4.54	3.10	100
Singh (step 2)	1.11	0.89	99.6
Heeger (level 1)	4.53	2.41	57.8
Waxman et al. ($\sigma_f = 2.0$)	6.66	10.72	1.9
Fleet and Jepson ($\tau = 2.5$)	0.32	0.38	74.5
Weber and Malik	0.49	0.35	96.8
Bober and Kittler	0.54	1.69	100
Ong and Spann	0.39	0.29	95.2
Odobez and Bouthemy	0.48	0.43	98.9
Szeliski and Coughlan	0.35	0.34	100
Srinivasan and Chellappa	0.61	0.26	100
Lai and Vemuri (gradient-based)	0.40	0.28	100
Lai and Vemuri (SSD-based)	0.30	0.27	100
Farnebäck (affine) (from [36])	0.62	1.99	100
Farnebäck (constant) (from [36])	1.44	6.28	100
Liu et al. [35]	0.66	0.83	100
Liu et al. [36]	0.29	0.32	100
Gibson and Spann	0.80	0.67	100
Proposed method (GRRDC)	0.21	0.16	100

Table 4
Summary of *Diverging Tree* results

Technique	Average error (deg)	SD (deg)	Density (%)
Horn and Schunck (modified)	2.55	3.67	100
Lucas and Kanade ($\lambda_2 \geq 1.0$)	1.94	2.06	48.2
Uras et al. (unthreshold)	4.64	3.48	100
Nagel	2.94	3.23	100
Anandan (frames 19 and 21)	7.64	4.96	100
Singh (step 2)	8.40	4.78	99.0
Heeger	4.49	3.10	74.2
Waxman et al. ($\sigma_f = 2.0$)	11.23	8.42	4.9
Fleet and Jepson ($\tau = 2.5$)	0.99	0.78	61.0
Bober and Kittler	3.69	4.39	100
Weber and Malik	3.18	2.50	88.6
Ong and Spann	1.32	1.16	95.1
Odobez and Bouthemy	1.43	1.17	97.4
Kumar et al.	0.76	1.12	32.96
Srinivasan and Chellappa	2.94	1.64	100
Lai and Vemuri (gradient-based)	1.34	1.05	100
Liu et al. [35]	1.86	1.35	100
Gibson and Spann	1.50	2.56	100
Proposed method (GRR)	1.23	0.85	100
Proposed method (GRRDC)	1.45	0.93	100

Table 5
Summary of *Yosemite* results (without sky)

Technique	Average error (deg)	SD (deg)	Density (%)
Black and Anandan	4.46	4.21	100
Szeliski and Coughlan	2.45	3.05	100
Black	3.52	3.25	100
Black and Jepson	2.29	2.25	100
Ju et al.	2.16	2.0	100
Mémin and Pérez [39]	2.34	1.45	100
Mémin and Pérez [42]	1.58	1.21	100
Lai and Vemuri (gradient-based)	1.99	1.41	100
Lai and Vemuri (SSD-based)	2.04	1.52	100
Farnebäck (constant) [40]	1.94	2.31	100
Farnebäck (affine) [40]	1.40	2.57	100
Farnebäck [41]	1.14	2.14	100
Proposed method (GRRDC)	1.52	1.27	100

those techniques only reporting sky-excluded *Yosemite* sequence, we also list our results with these techniques for this sequence in Table 5. To our knowledge, Farnebäck's method has the best performance for the sky-free *Yosemite* sequence reported to date in literature. Our method has a larger average angular error but smaller standard deviation than Farnebäck's method. Considering the average angular error and the standard deviation simultaneously, our algorithm is still comparable to Farnebäck's method for the sky-excluded *Yosemite* sequence.

Scene brightness variation field, which represents the radiometric variations in dynamic imagery, is also quite important in optical flow computation. The estimated scene brightness variation field, which is evaluated by the formula $\delta I_i = -m_i I_i - c_i$, for these synthetic sequences are shown in Fig. 5. This figure reveals that, for the *Yosemite* sequence, it contains obvious brightness variations in the sky region with one side becoming brighter and the other side getting darker as we observe in this sequence. For *Rotating/Translating Yosemite*, the imposed illuminations are multiplicative Gaussian functions (see Eq. (31)), thus the true brightness variation should depend on this Gaussian function and the original image intensity function. The estimated scene brightness variation fields are shown in Figs. 5B and C, where the rough image structure is revealed and there are highlights in the center portion of the image, which comes from the imposed Gaussian function. These results coincide with our expectation. The *Translating/Diverging Tree* are synthesized to simulate translational camera motion with respect to a textured planar surface, thus there are no brightness variations in these two sequences which coincides with Figs. 5D and E. Although Fig. 5 can correctly describe the behavior of brightness variation of these sequences, it can not give a quantitative accuracy measure. The accuracy of the estimated scene brightness variation field can be calculated by the following error measure:

$$\begin{aligned} BVE(x, y) &= |BV_c(x, y) - BV_e(x, y)| \\ &= |I(x + u, y + v, t + 1) - I(x, y, t) + m(x, y)I(x, y, t) + c(x, y)|, \end{aligned} \quad (32)$$

where $BV_c(x, y) = I(x + u, y + v, t + 1) - I(x, y, t)$ is the true brightness variation and $BV_e(x, y) = -m(x, y)I(x, y, t) - c(x, y)$ is the estimated one. Computation of the true

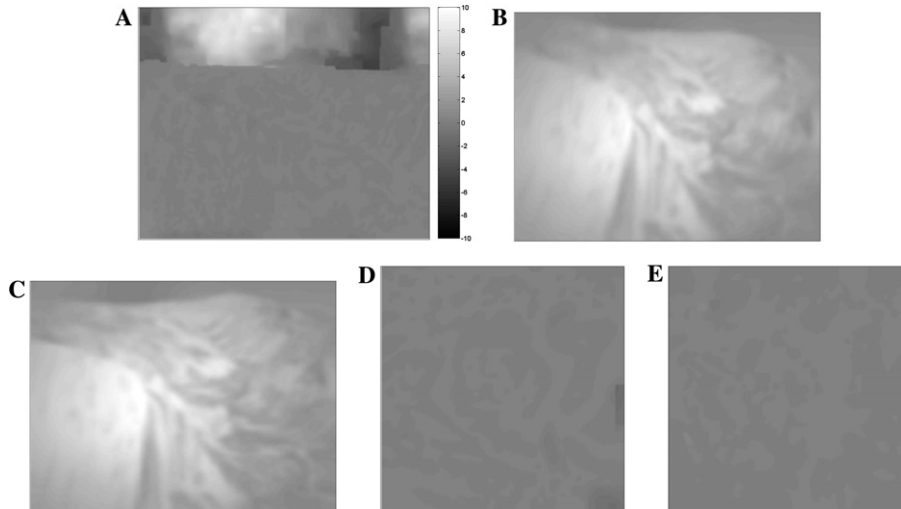


Fig. 5. Estimated scene brightness variation fields of: (A) *Yosemite*, (B) *Rotating Yosemite*, (C) *Translating Yosemite*, (D) *Diverging Tree*, and (E) *Translating Tree* (The unit of the gray bar is gray level.)

scene brightness variation field, which involves the motion vector $[u, v]^T$, implies that the interpolation technique is inevitable when u and v are not integers. This interpolation technique will introduce errors, thus making this error measure inaccurate. Therefore, we only compute the accuracy of the estimated scene brightness variation field of the *Translating Yosemite* sequence, which was synthesized with integer motion field, in this paper. Table 6 lists the average and standard deviation of the brightness variation error for the experiment on the *Translating Yosemite* sequence. To demonstrate how accurately our algorithm can achieve, the error of the BCM-based method is also listed for comparison. For BCM-based method, the brightness variation multiplier and offset fields are assumed to be zero, so the $BVE(x, y)$ is equal to $|I(x + u, y + v, t + 1) - I(x, y, t)|$, the true brightness variations of this testing sequence. Clearly, our algorithm produces very accurate brightness variation field with the average brightness variation error less than 1 gray-level as indicated in Table 6. In fact, because of the pre-smoothing operation in our algorithm, the estimated brightness variation field is computed based on a smoothed version of the input image sequence. If we calculate this error measure using the smoothed images, the average brightness variation error even lower to 0.05 gray-level with a standard deviation equal to 0.06 gray-level.

Theoretical speaking, the constraint refinement scheme can allow our algorithm to accommodate larger image motions. This characteristic is stressed by varying the motion speeds of the *Translating Yosemite* sequences and the results are listed in Table 7. We synthesized the *Translating Yosemite* sequences with uniform flow speeds ranged from 1 pixel/frame to 4 pixels/frame. To focus on the issue of motion

Table 6
Brightness variation error of *Translating Yosemite* sequence

Method	Average error (gray-level)	SD (gray-level)
BCM-based method	4.00	2.44
Proposed method (GRRDC)	0.50	0.42

Table 7
The results of our algorithm with/without the constraint refinement scheme

Motion speed		GR ^a	GRC ^b
1 pixel/frame	Average error (deg)	0.00	0.00
	SD (deg)	0.00	0.00
2 pixels/frame	Average error (deg)	0.65	0.01
	SD (deg)	0.26	0.01
3 pixels/frame	Average error (deg)	0.58	0.02
	SD (deg)	0.26	0.11
4 pixels/frame	Average error (deg)	9.83	0.05
	SD (deg)	13.76	0.30

^a GR: proposed gradient-based regularization method.

^b GRC: GR + constraint refinement scheme.

Table 8
Computing time of proposed algorithm

Method	<i>Translating Tree</i> (150×150)	<i>Yosemite</i> (316×252)
GR (20 iterations)	0.53 s ^a (0.60° ^b)	2.41 s (4.14°)
GR (50 iterations)	1.24 s (0.56°)	5.56 s (3.96°)
GRR (50 iterations)	1.47 s (0.51°)	6.67 s (4.04°)
GRRD (50 iterations)	1.52 s (0.52°)	6.80 s (2.77°)
GRRDC (100 iterations)	3.09 s (0.21°)	13.94 s (2.70°)

^a This program was executed on a PC running Windows 2000 with Pentium IV 2.4 GHz CPU and 768 MB RAM.

^b The number in the parenthesis is the corresponding average angular error.

speeds in these experiments, we do not impose non-uniform illuminations on these sequences. As indicated in Table 7, the results obtained with the constraint refinement scheme are much more accurate than those without the refinement.

In some applications, efficiency is an important issue in optical flow estimation. Because of the iterative procedures, the time complexity of our algorithm depends on the number of iterations required in the numerical minimization process. Table 8 lists the computing time of the proposed algorithm running on a PC with a Pentium IV 2.4 GHz CPU and 768MB RAM for the *Yosemite* and *Translating Tree* sequences. As shown in this table, the computing time is approximately linearly dependent with the total number of iterations. In fact, in our algorithm, efficiency and performance can be carefully managed by controlling the number of iterations. For instance, by setting the total number of iterations of our algorithm (GR) for the *Yosemite* sequence to 20 ($n = 20$, $N = 0$, $M = 0$ in our final algorithm), we can obtain the motion estimation with 4.14° for the average angular error and 7.70° for the standard deviation. The computing time on the same computer is about 2.41 s, which is only about one-sixth of the time required to produce our best result in Table 1. The computing time is greatly reduced; however, the obtained accuracy is just slightly degraded. This slightly degraded accuracy still outperforms all the other results shown in Table 2.

Finally, we test the sensitivity of our algorithm with respect to the regularization parameter λ . We vary λ from 0.5 to 2.5 and calculate the average angular error of *Yosemite* and *Translating/Diverging Tree* sequences. The results are depicted in Fig. 6, which reveals that our algorithm has almost the same performance at this range of λ . This phenomenon demonstrates that our algorithm is insensitive to the regularization parameter λ at this range as claimed in the beginning of this section.

4.2. Results for real image sequences

We also tested our algorithm on three real image sequences, namely, the *Toy Car*, *Moving Camera*, and *Hamburg Taxi* sequences. The image size of *Toy Car* sequence is 360×240 pixels. Fig. 7A shows one frame from the *Toy Car* sequence. In this sequence, there is a toy car moving from top-left to the bottom-right corner of the image. A shadow casting on the right portion of the image introduces brightness

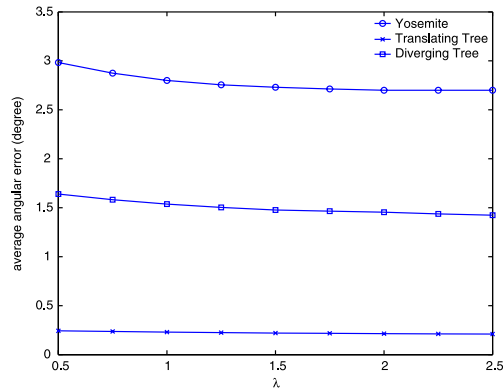


Fig. 6. The average angular errors for different values of the regularization parameter λ .

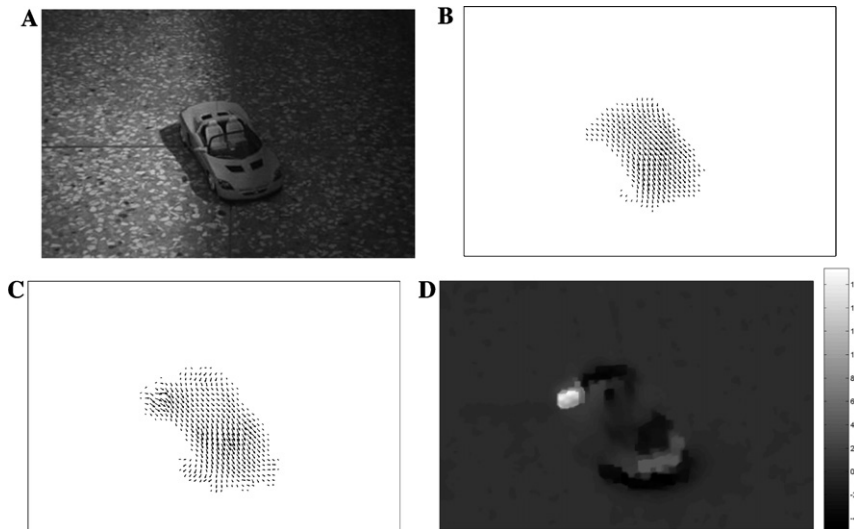


Fig. 7. Experimental results of *Toy Car* sequence. (A) One frame from this sequence; (B) Estimated motion field by using proposed method; (C) Estimated motion field by applying Lai and Vemuri's method [3]; (D) Computed scene brightness variation field by using proposed method.

variations when the car moves into the shadow. The moving speed of the car is about 3–4 pixels/frame. Fig. 7B depicts the motion field estimated by our algorithm. To illustrate the effectiveness of our algorithm, the image motion obtained by employing Lai and Vemuri's gradient-based regularization method (see [3]) is shown in Fig. 7C for comparison. As shown in this figure, our algorithm yields more correct motion field than that produced by Lai and Vemuri's method. The motion boundary can be clearly seen from the motion field estimated by our algorithm. However, our algorithm still can not give correct motion vectors in the region of the car shadow, where

the corresponding image motion should be zero. In fact, this is a generalized aperture problem (see [9]), for the shadow is so clear that our algorithm can not distinguish whether the intensity variation is caused by the motion or the shadow. We also show the brightness variation field estimated by our algorithm in Fig. 7D. The highlight in this figure comes from the shadow of the car, where the image brightness is greatly increased because of the moving shadow.

The *Moving Camera* sequence was obtained by translating a camera with respect to a board. A light source was moving along with the camera during the video acquisition, thus causing the highlight in the image as revealed in Fig. 8A. The camera was moved from the left side to the right side of the board, thus the image motion should move to the left. The motion speed is about 1–3 pixels/frame. The image size of this sequence is 360×240 . Fig. 8B shows the estimated motion field by our algorithm. Again, Fig. 8C shows the results by applying Lai and Vemuri's gradient-based method. Because of the brightness constancy assumption imposed in Lai and Vemuri's method, their result reveals inaccurate motion fields in the highlight region of the image where there exists clear brightness variations. However, our algorithm produces very accurate image motion all over the image. Fig. 8D illustrates our estimated scene brightness variation field. As indicated in this figure, the image brightness becomes brighter in the right portion of the highlight and gets darker in the left portion of the highlight. This phenomenon coincides with the image sequence acquisition for the light source was moved from left to right.

The *Hamburg Taxi* is also a well-known testing sequence for optical flow computation (e.g. [1,3,12,19,20,22,27,34,35,37]). The experimental results for this sequence are displayed in Fig. 9. There are no obvious brightness variations in this sequence,

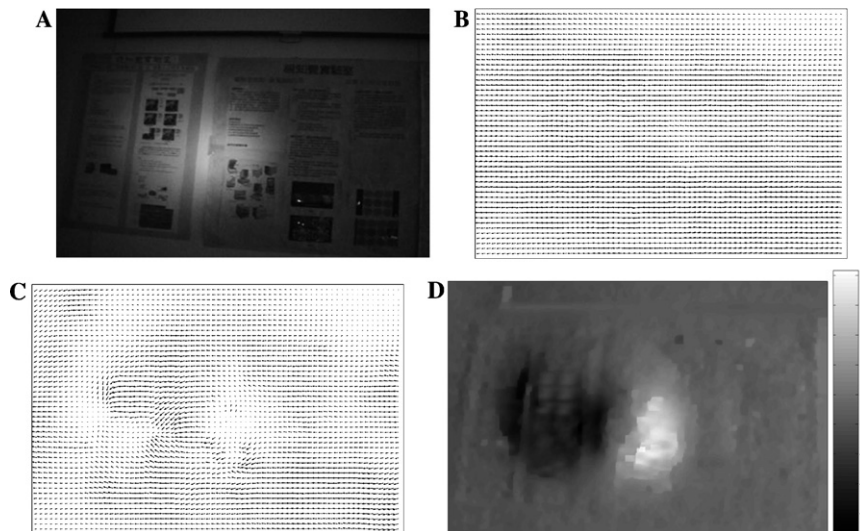


Fig. 8. Experimental results of *Moving Camera* sequence. (A) One frame from this sequence; (B) Estimated motion field by using proposed method; (C) Estimated motion field by applying Lai and Vemuri's method [3]; (D) Computed scene brightness variation field by using proposed method.

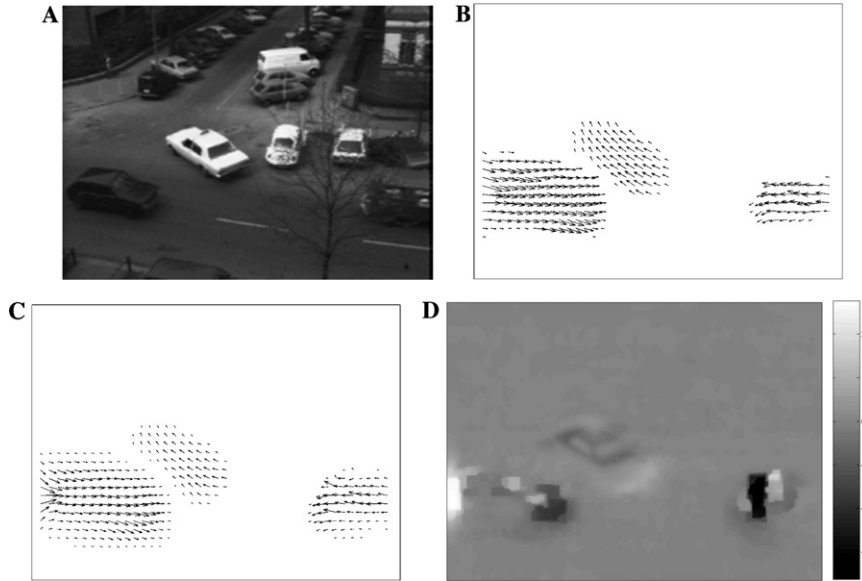


Fig. 9. Experimental results of *Hamburg Taxi* sequence. (A) One frame from this sequence; (B) Estimated motion field by using proposed method; (C) Estimated motion field by applying Lai and Vemuri's method [3]; (D) Computed scene brightness variation field by using proposed method.

thus Lai and Vemuri's result is comparable with ours. However, because of our dynamic smoothness adjustment scheme, our algorithm produces sharper motion discontinuities than those from Lai and Vemuri's method. A notable point from the results can be seen from the van in the lower right portion of the image where there exist unusual brightness variations as shown in Fig. 9D. This phenomenon may be caused by the occlusion of the tree on the van. This occlusion will make the image brightness to be slightly varied.

5. Conclusion

In this paper, we developed a very accurate gradient-based regularization algorithm for optical flow computation under non-uniform brightness variations. Our algorithm was constructed from the GDIM-based optical flow constraint and three auxiliary estimation schemes: the robust estimation, dynamic smoothness adjustment, and constraint refinement scheme. We adopted GDIM to cope with the problem of brightness variations and applied robust estimation to reduce the influence of unreliable image flow constraints. We proposed a dynamic smoothness adjustment scheme to effectively suppress the smoothness constraint in the vicinity of motion discontinuities, thereby yielding more distinct motion boundaries. Finally, a constraint refinement estimation scheme was utilized to reduce the error introduced by the first-order optical flow constraint, thus leading to more accurate optical flow

estimation especially for large image motion. The solution of the resulting linear system for the above optical flow regularization problem was obtained by using an efficient numerical algorithm, i.e., the incomplete Cholesky preconditioned conjugate gradient algorithm. Experimental results on some real and synthetic image sequences showed that our method produces nearly the most accurate optical flow estimation among the techniques reported in literature to date.

Acknowledgments

This work was supported by the Program for Promoting Academic Excellence of University (89-E-FA04-1-4) and National Science Council, Taiwan, Republic of China, under the Grant Nos. NSC 92-2213-E-007-018 and NSC 92-2213-E-007-028.

References

- [1] J.L. Barron, D.J. Fleet, S.S. Beauchemin, System and experiment performance of optical flow techniques, *Int. J. Comput. Vision* 12 (1) (1994) 43–77.
- [2] B.K.P. Horn, B.G. Schunck, Determining optical flow, *Artif. Intell.* 17 (1981) 185–203.
- [3] S.H. Lai, B.C. Vemuri, Reliable and efficient computation of optical flow, *Int. J. Comput. Vision* 29 (2) (1998) 87–105.
- [4] F.J. Hampson, J.C. Pesquet, Motion estimation in the presence of illumination variations, *Signal Process.: Image Commun.* 16 (2000) 373–381.
- [5] S.H. Lai, Adaptive motion estimation for image sequences under non-uniform illumination variations, in: *Proceedings International Computer Symposium: Workshop on Image Processing and Pattern Recognition*, Chiayi, Taiwan, 2000, pp. 223–230.
- [6] A. Nomura, H. Miike, K. Koga, Determining motion fields under non-uniform illumination, *Pattern Recogn. Lett.* 16 (1995) 285–296.
- [7] A. Nomura, Spatio-temporal optimization method for determining motion vectors fields under non-stationary illuminations, *Image Vision Comput.* 18 (2000) 939–950.
- [8] L. Zhang, T. Sakurai, H. Miike, Detection of motion fields under spatio-temporal non-uniform illuminations, *Image Vision Comput.* 17 (1999) 309–320.
- [9] S. Negahdaripour, Revised definition of optical flow: integration of radiometric and geometric cues for dynamic scene analysis, *IEEE Trans. Pattern Anal. Mach. Intell.* 20 (9) (1998) 961–979.
- [10] H.W. Haussecker, D.J. Fleet, Computing optical flow with physical models of brightness variation, *IEEE Trans. Pattern Anal. Mach. Intell.* 23 (6) (2001) 661–673.
- [11] M. Yeasin, Optical flow in log-mapped image plane—a new approach, *IEEE Trans. Pattern Anal. Mach. Intell.* 24 (1) (2002) 125–131.
- [12] E.P. Ong, M. Spann, Robust optical flow computation based on least-median-of-squares regression, *Int. J. Comput. Vision* 31 (1) (1999) 51–82.
- [13] M.J. Black, P. Anandan, The robust estimation of multiple motions: parametric and piecewise-smooth flow fields, *Comput. Vision Image Und.* 63 (1) (1996) 75–104.
- [14] M.J. Black, D.J. Fleet, Probabilistic detection and tracking of motion boundaries, *Int. J. Comput. Vision* 38 (3) (2000) 231–245.
- [15] D. Shulman, J.Y. Hervé, Regularization of discontinuous flow fields, in: *Proceedings Workshop on Visual Motion*, 1989, pp. 81–86.
- [16] A. Kumar, A.R. Tannenbaum, J. Balas, Optical flow: a curve evolution approach, *IEEE Trans. Image Process.* 5 (4) (1996) 598–610.
- [17] F. Heitz, P. Bouthemy, Multimodal estimation of discontinuous optical flow using Markov random fields, *IEEE Trans. Pattern Anal. Mach. Intell.* 15 (12) (1993) 1217–1232.

- [18] H.-H. Nagel, Constraints for the estimation of displacement vector fields from image sequences, in: *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, vol. 2, 1983, pp. 945–951.
- [19] H.-H. Nagel, W. Enkemann, An investigation of smoothness constraints for the estimation of displacement vector fields from image sequence, *IEEE Trans. Pattern Anal. Mach. Intell.* 8 (5) (1986) 565–593.
- [20] J. Weickert, C. Schnörr, Variational optic flow computation with a spatio-temporal smoothness constraint, *J. Math. Imaging Vision* 14 (2001) 245–255.
- [21] H.-H. Nagel, On the estimation of optical flow: relations between different approaches and some new results, *Artif. Intell.* 33 (1987) 299–324.
- [22] L. Alvarez, J. Weickert, J. Sanchez, Reliable estimation of dense optical flow fields with large displacements, *Int. J. Comput. Vision* 39 (1) (2000) 41–56.
- [23] H.-H. Nagel, Extending the ‘oriented smoothness constraint’ into temporal domain and the estimation of derivatives optical flow, in: O. Faugeras (Ed.), *Computer Vision-ECCV’90*, Lecture Notes in Computer Science, vol. 427, Springer, Berlin, 1990, pp. 139–148.
- [24] C. Schnörr, Determining optical flow for irregular domains by minimizing quadratic functionals of a certain class, *Int. J. Comput. Vision* 6 (1991) 25–38.
- [25] M.A. Snyder, On the mathematical foundations of smoothness constraints for the determination of optical flow and for surface reconstruction, *IEEE Trans. Pattern Anal. Mach. Intell.* 13 (1991) 1105–1114.
- [26] R. Deriche, P. Kornprobst, G. Aubert, Optical-flow estimation while preserving its discontinuities: a variational approach, in: *Proceedings Second Asian Conference on Computer Vision, ACCV’95*, Singapore, 1995, pp. 290–295.
- [27] G. Aubert, R. Deriche, P. Kornprobst, Computing optical flow via variational techniques, *SIAM J. Appl. Math.* 60 (1) (1999) 156–182.
- [28] J.R. Bergen, P. Anandan, K.J. Hanna, R. Hingorani, Hierarchical model-based motion estimation, in: *Proceeding of European Conference on Computer Vision*, 1992, pp. 237–252.
- [29] S. Negahdaripour, A. Shokrollahi, M. Gennert, Relaxing the brightness constancy assumption in computing optical flow, in: *Proceeding ICIP-89*, Singapore, 1989.
- [30] S. Negahdaripour, C.H. Yu, A generalized brightness change model for computing optical flow, in: *Proceedings International Conference on Computer Vision*, Germany, 1993.
- [31] S.H. Lai, Robust image matching under partial occlusion and spatially varying illumination change, *Computer Vision Image Und.* 78 (2000) 84–98.
- [32] G. Li, Robust regression, in: D.C. Hoaglin, F. Mosteller, J.W. Tukey (Eds.), *Exploring Data Tables, Trends, and Shapes*, Wiley, New York, 1985, pp. 281–343.
- [33] G.H. Golub, C.F.V. Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1989.
- [34] S. Srinivasan, R. Chellappa, Noise-resilient estimation of optical flow by use of overlapped basis functions, *J. Opt. Soc. Am. A* 16 (3) (1999) 494–507.
- [35] H. Liu, T. Hong, M. Herman, R. Chellappa, A general motion model and spatio-temporal filters for computing optical flow, *Int. J. Comput. Vision* 22 (1997) 141–172.
- [36] H. Liu, R. Chellappa, A. Rosenfeld, Accurate dense optical flow estimation using adaptive structure tensor and a parametric model, in: *Proceedings International Conference on Pattern Recognition*, Québec, Canada, 2002.
- [37] D. Gibson, M. Spann, Robust optical flow estimation based on a sparse motion trajectory set, *IEEE Trans. Image Process.* 12 (4) (2003) 431–445.
- [38] M.J. Black, Recursive nonlinear estimation of discontinuous flow fields, in: *Proceedings of European Conference on Computer Vision*, 1994, pp. 138–145.
- [39] E. Mémin, P. Pérez, Dense estimation and object-based segmentation of the optical flow with robust techniques, *IEEE Trans. Image Process.* 7 (5) (1998) 703–719.
- [40] G. Farneback, Fast and accurate motion estimation using orientation tensors and parametric motion models, in: *Proceedings of International Conference on Pattern Recognition*, 2000, pp. 135–139.

- [41] G. Farnebäck, Very high accuracy velocity estimation using orientation tensors, parametric motion, and simultaneous segmentation of the motion field, in: *Proceedings of International Conference on Computer Vision*, 2001, pp. 171–177.
- [42] E. Mémin, P. Pérez, Hierarchical estimation and segmentation of dense motion fields, *Int. J. Comput. Vision* 46 (2) (2002) 129–155.