Contents lists available at ScienceDirect

# Computer Vision and Image Understanding



journal homepage: www.elsevier.com/locate/cviu

## Optical-flow based on an edge-avoidance procedure

## Pierre-Marc Jodoin<sup>a,\*</sup>, Max Mignotte<sup>b</sup>

<sup>a</sup> Département d'Informatique, Université de Sherbrooke, 2500 Boul. de l'Université, Sherbrooke, Que., Canada J1K 2R1 <sup>b</sup> Département d'Informatique et de Recherche Opérationnelle (DIRO), Université de Montréal, P.O. Box 6128, Stn. Centre-Ville, Montréal, Que., Canada H3C 3J7

#### ARTICLE INFO

Article history: Received 31 May 2006 Accepted 8 December 2008 Available online 24 December 2008

Keywords: Optical flow Motion estimation Information fusion Mean-shift

### ABSTRACT

This paper presents a differential optical flow method which accounts for two typical motion-estimation problems: (1) flow regularization within regions of uniform motion while (2) preserving sharp edges near motion discontinuities *i.e.*, where motion is multimodal by nature. The method proposed is a modified version of the well known Lucas–Kanade (LK) algorithm. While many edge-preserving strategies try to minimize the effect of *outliers* by using a line process or a robust function, our method takes a novel approach to solve the problem. Based on documented assumptions, our method computes motion with a classical least-squares fit on a local neighborhood shifted away from where motion is likely to be multimodal. In this way, the inherent bias due to multiple motion around moving edges is avoided instead of being compensated. This edge-avoidance procedure is based on the non-parametric mean-shift algorithm which shifts the LK integration window away from local sharp edges. Our method also locally regularizes motion by performing a fusion of local motion estimates. The regularization is made with a covariance filter which minimizes the effect of uncertainties due in part to noise and/or lack of texture. Our method is compared with other edge-preserving methods on image sequences representing different challenges. Crown Copyright © 2008 Published by Elsevier Inc. All rights reserved.

#### 1. Introduction

In the past 30 years, numerous solutions have been proposed to solve the optical flow problem [1-3]. As observed by Barron et al. [3], optical flow techniques can be divided into families among which are the phase-based methods [4], spectral-based methods [5], energy-based methods [6-8], Markovian methods [9-11], and differential methods [12-14]. Choosing one approach versus another depends very much on the application and the nature of the flow to be estimated. For instance, a densely cluttered scene with different global movements might be amenable to a spectral-based method, whereas a differential method might be better suited for a scene exhibiting objects (such as cars for instance) moving in front of a fixed background. These techniques are various and adapted to all kinds of situations. However, most of them are expressed as global optimization problems involving a data conservation constraint likelihood term and a spatial coherence constraint prior term (also called regularization term) [15].

The data conservation constraint is generally built upon the *brightness constancy assumption* which stipulates that the brightness of a single point remains constant with time. This simple assumption is frequently used to develop simple optical flow

methods that generate fairly good results. However, as one might expect, the brightness constancy assumption is only valid for ideal noise-free scenes and thus almost never holds exactly. Furthermore, due to lack of texture and occlusion, different motions within a local area can be indistinguishable even though only one is valid. In other words, the brightness constancy assumption only partially constrains the data leaving the problem ill-posed since this assumption allows several solutions as being "optimal", although only one is objectively correct. This problem is well documented and referred to as the *aperture problem* [1,15].

To obtain accurate estimates, a *spatial coherence constraint* has to be added to the first constraint. While some authors see this additional constraint as an *integration window* [13,16,17], others see it as a *prior function* [11,12] modeling the way motion vectors should be distributed in the final vector field. The choice of spatial constraint is often the key element differentiating one approach from another. Typically, these spatial constraints assume that motion is locally uniform and changes smoothly across regions of the scene. Unfortunately, this isotropic assumption is violated when a region spans a motion discontinuity [15,18]. As explained by Black and Anandan [15], the spatial neighborhood must be large enough to sufficiently constrain the solution but also small enough to avoid spanning multiple motions.

Many popular motion estimation approaches were built upon these competing constraints [7,12,13], which often ignore the *multimodal* nature of motion near moving edges. As a result, the estimated motion in these areas is often imprecise and blurry. To gain more accuracy and better preserve motion discontinuities,

<sup>\*</sup> Corresponding author. E-mail addresses: Pierre-Marc.Jodoin@usherbrooke.ca (P.-M. Jodoin), mignotte@ iro.umontreal.ca (M. Mignotte).

URLs: http://www.dmi.usherb.ca/~jodoin/ (P.-M. Jodoin), http://www.iro.umontreal. ca/~mignotte/ (M. Mignotte).

several solutions have been proposed. As mentioned by Thompson [18], the optical flow literature proposes two broad families of solutions designed to preserve sharp motion edges: the *flow-based family* and the *image-based family*.

#### 1.1. Flow-based methods

The flow-based methods typically allow for mixed motion distributions near boundaries. In fact, these methods assume that the imprecision around moving edges is due to the presence of "outliers" pooling from a spatial neighborhood and corrupting the final solution. Consequently, this family of solutions aims to minimize the influence of these undesired values.

One popular method for dealing with outliers is the use of a robust error function that gives a relative influence to outliers, preventing them from corrupting the final solution. Black and Anandan [15,19] (soon followed by Odobez and Bouthemy [20]) were the first to explicitly use robust functions to minimize the effect of those measures violating the data conservation or the spatial coherence constraint. They adapted their robust framework to two common motion estimation techniques: the recovery of multiple parametric motion models and the recovery of piecewise-smooth flow fields. Aubert et al. [21] presented a variational technique whose convergence was carefully demonstrated. Their method is presented as a differential method similar to that of Horn and Schunck [12] but with an edge preserving regularization term and a half-quadratic optimizer. Weickert [22] also proposes a variational technique implementing a robust isotropic regularization term. The use of such regularization term prevents the method from smoothing across motion discontinuity. Shortly after, Weickert and Schnörr [23] proposed another variational method that could be seen as an extension of Weickert's technique [22]. In their paper, they propose an anisotropic flow-driven regularization technique which not only prevents from smoothing across flow discontinuity, but also encourages smoothing along flow edges. This method implements a diffusion tensor embedded in a diffusionreaction system minimized with a downhill search technique. More recently, Brox et al. [24] proposed a variational technique minimizing an energy function composed of three terms: a brightness consistency term, a novel gradient constancy term and a robust spatio-temporal smoothness term.

Another class of flow-based optical flow methods is those implementing a *least-median-of-squares* technique. As opposed to the robust methods which minimize the influence of outliers, these methods explicitly detect and reject outliers. The retained measures (called *inliers*) are then used to estimate the flow. Bab-Hadiashar and Suter [25] and Ong and Spann [26] proposed interesting work in that field.

Other flow-based methods consider motion vectors as "estimates" that are to be fused together. During the fusion process, each estimate is given an importance value which makes it more or less influent locally. In this way, a region with a high confidence will propagate more information than a region with a low confidence. An early paper in that field has been published by Singh [6] in 1990. In this work, Singh uses a multiresolution-two-step sum of squared differences (SSD) procedure to help preserve flow discontinuity without any prior knowledge on the location of the boundary. In a first step, Singh's method estimates a variancecovariance matrix based on a SSD measure over a correlation window. This matrix is then used as a confidence measure to propagate the flow using neighborhood information. More recently, Comaniciu [27] presented a non-parametric fusion approach that appears as an improvement of Singh's method [6]. Comaniciu's method pre-estimates the flow with a simple biased-least-squares method before locally fusing the motion vectors with a multiscale mean-shift procedure.

Let us also mention the work by Farneback [28] in which 3D orientation tensors are combined to affine parametric motion models for the estimation of the optical flow. Farneback shows that the fundamental relations between the 3D orientation tensors and optical motion as a good solution for quickly estimating precise motion fields. The author argues that this approach can be combined with a region-growing segmentation algorithm to sharpen the results [28].

#### 1.2. Image-based methods

The second class of edge-preserving optical flow methods attempt to locate motion discontinuities [18]. These methods often assume that a flow boundary always corresponds to an intensity edge. Based on this assumption, most image-based methods make sure that measures gathered from opposite side of an intensity edge never influence each other. In that perspective, several approaches coping with motion discontinuities use the Markov Random Field (MRF) formulation inspired by Geman and Geman's work [29]. Most of these MRF methods rely on a line process to estimate border locations and keep points from opposite sides of the border from influencing each other [10,11]. In particular, this is what Black [30] does when he jointly estimates a motion vector field together with a motion region map. To do so, this method minimizes an energy-based Markovian function made up of an intensity model, a border model and a motion model. Local constraints on motion and intensity allow for the preservation of sharp flow discontinuities.

Other image-based methods estimate motion based on a preestimated segmentation map. These methods make the underlying assumption that a motion discontinuity cannot span a region of uniform intensity. In that perspective, a color region map is preestimated and the optical flow is computed within the color regions. Among the first contributions in this area is the work by Fuh and Maragos [31] who proposed a region-based matching method combined with a post-processing median filtering. A few years later, Meyer and Bouthemy [32], Dang et al. [33], and Black and Jepson [34] proposed methods to fit affine models within pre-estimated regions of uniform intensity. In the latter paper, the method has been made iterative to better refine the flow.

Many image-based variational techniques have also been proposed. These methods can be seen as extensions of Horn and Schnunk's original method [12]. These variational methods use a regularization term to smooth out the flow based on the image content. For instance, in their 1999 paper, Alvarez et al. [35] proposed a variational method that reduces its regularization in the vicinity of all image edges. This is done with a non-linear regularizer inversely proportional to the image gradient. Such an approach is sometimes referred to as an image-based isotropic variational approach [23]. Another class of variational techniques comprises the image-based anisotropic methods [23]. These methods implement a regularization term that both prevents smoothing flow across image edges and encourages smoothing along image edges. Anisotropic image-based methods were proposed by Nagel-Enkelmann [36], Nagel [14], Schnörr [37], and more recently, Alvarez et al. [38].

#### 1.3. Our method

In this paper, a modification to the well known Lucas–Kanade (LK) algorithm [13] is proposed. The objective of our method is twofold: (1) minimize uncertainties (often caused by noise and lack of texture) by strongly constraining the flow within regions of uniform movement while (2) preserving flow discontinuities around moving objects. Since our method is based on a least-squares fit (and thus is sensitive to multimodal motion) the

key idea is to avoid computing flow in areas where motion is likely to be multimodal. Following the same assumption made by most image-based methods, our approach assumes that every motion boundary correspond to a strong intensity edge. More specifically, in areas near a strong intensity gradient, our algorithm computes motion with a neighborhood window shifted away from the nearest intensity edge. In this way, the flow is computed over sections of the scene where motion is likely to be unimodal. Our least-squares fitting algorithm can thus preserve sharp motion boundaries by avoiding having to deal with multiple motions. To our knowledge, such an avoidance procedure has never been investigated before. To better constrain the solution, our method implements a fusion procedure similar to the flow-based method proposed by Singh [6]. This fusion procedure implements a covariance filter that locally averages motion. In the experimental section, our method is compared with other techniques whose purpose is also to preserve motion discontinuities. Results are obtained after processing synthetic, realistic, and real sequences.

The remainder of this paper is organized as follows. Because our method is a modified version of LK, an introduction of the LK method [13] is first presented in Section 2. Section 3 then presents our modifications to LK, which includes a covariance filter and an edgeavoidance procedure based on the mean-shift [39] algorithm. Section 5 then presents the optical flow methods to which we compare our method. This section also includes results obtained on various synthetic, realistic and real image sequences. Section 6 discusses our method and concludes.

#### 2. Lucas-Kanade motion estimation

The Lucas–Kanade (LK) approach was first introduced as a least-squares fitting method applied to stereovision [13]. However, its extension to motion estimation is trivial and goes as follows. Let  $S = \{s = (i, j) | i \in [0, \mathcal{M}], j \in [0, \mathcal{M}]\}$  denotes a 2D lattice of size  $\mathcal{N} \times \mathcal{M}$  and I(s, t) the intensity of the site *s* at time *t*. In our implementation, I(s, t) takes a value between 0 and 255. Considering the brightness constancy assumption and assuming that the velocity is locally linear, LK looks for a vector field  $V = \{\vec{v}_s = (u_s, v_s) | s \in S, \ \vec{v}_s \in \mathbb{R}^2\}$  that minimizes the residual quadratic error

$$E(\vec{\nu}_s) = \sum_{r \in \eta_s} \left[ I(r,t) - I(r + \vec{\nu}_s, t+1) \right]^2, \quad \forall \ s \in S$$

$$\tag{1}$$

where  $\eta_s$  is a neighborhood window of size  $N \times N$  centered on site *s*.  $E(\vec{v}_s)$  can be reformulated based on its Taylor expansion:

$$E(\vec{v}_{s}) \approx \sum_{r \in \eta_{s}} \left[ u_{s} I_{x} + v_{s} I_{y} + I_{T} \right]^{2}$$

$$= \sum_{r \in \eta_{s}} \left[ \nabla I^{T} \vec{v}_{s} + I_{T} \right]^{2}$$
(2)
(3)

where  $I_x$ ,  $I_y$  and  $I_T$  are, respectively, the spatial and temporal derivatives over site *r* at time *t* [16,40]. As mentioned by Lucas and Kanade, this quadratic error function can be minimized by setting its first derivative to zero:  $\frac{\partial E(\bar{v})}{\partial \bar{v}} = 0$ , which formally corresponds to

$$\sum_{r\in\eta_s} \nabla I \left( \nabla I^{\mathrm{T}} \vec{v}_s + I_T \right) = \mathbf{0} \tag{4}$$

or, equivalently,

$$\begin{bmatrix} \begin{pmatrix} \sum_{r} I_x^2 & \sum_{r} I_x I_y \\ \sum_{r} I_x I_y & \sum_{r} I_y^2 \end{pmatrix} \vec{\nu}_s + \begin{pmatrix} \sum_{r} I_T I_x \\ \sum_{r} I_T I_y \end{pmatrix} \end{bmatrix} = 0.$$
(5)

Although the assumption that velocity is locally linear is true in regions of constant flow, it can be a problem when *N* is large and/or when  $\eta_s$  spans motion discontinuities. Thus, to minimize the influence of outliers, many authors add a weighting term  $W_i$  which gives more influence to pixels that are close of the center of  $\eta_s$  than those at the periphery [3]. This added term is mathematically expressed as

$$\left[ \begin{pmatrix} \sum_{r} W_{r} I_{x}^{2} & \sum_{r} W_{r} I_{x} I_{y} \\ \sum_{r} W_{r} I_{x} I_{y} & \sum_{r} W_{r} I_{y}^{2} \end{pmatrix} \vec{v}_{s} + \begin{pmatrix} \sum_{r} W_{r} I_{T} I_{x} \\ \sum_{r} W_{r} I_{T} I_{y} \end{pmatrix} \right]$$
(6)

where *W* typically contains Gaussian isotropic values. However, since the flow in our method is already constrained by a fusion procedure (see Section 3), *W* is assigned spatial gradient data  $(W_r = ||\nabla I_r||^2)$  in such a way that more influence is given to those sites located in textured areas.

To simplify the notation, it is common to rewrite Eq. (6) as

$$M_s \vec{v}_s + \vec{b}_s = 0 \tag{7}$$

where  $M_s$  is the 2 × 2 matrix in Eq. (6) and  $\vec{b}_s$  is the 2D vector in the same equation. Following Eq. (7), the least-squares solution can be obtained after a simple matrix inversion

$$\vec{v}_s = -M_s^{-1} b_s. \tag{8}$$

Of course, LK provides only a solution to those sites  $s \in S$  for which  $M_s$  is not singular. This singularity problem was first addressed by Nagel [41] and then by Barron et al. [3]. In the latter paper, a simple solution is proposed: reject every unreliable estimate  $v_s$  for which the eigenvalues  $\lambda_1$  and  $\lambda_2$  of  $M_s$ , are below a given threshold  $\tau$ . Although this approach is intuitively acceptable, it allows flow fields of density much lower than 100% (35% for the YOSEMITE sequence and 39% for the TRANSLATING TREE sequence [3]). Other authors [27,16] proposed adding a bias to  $M_s$  to ensure its invertibility. Mathematically, this can be formulated as

$$(M_s + \beta I_d)\vec{\nu}_s + \vec{b}_s = 0 \tag{9}$$

where  $I_d$  is the identity matrix and  $\beta$  may be a constant [27] or proportional to the covariance of the noise [16]. In this paper, we use a bias that is different from the one of Eq. (9). As shown in Algorithm 3, the bias we use is a small random white noise added to the input image sequence. This noise adds a bias to the spatial gradient  $(I_x, I_y)$  but does not affect the temporal gradient  $I_T$ . Thus, our bias makes  $M_s$  invertible everywhere without adding any significant error to the results, as can be seen in Figs. 16 and 17.

Optical flow obtained from (Eq. (8) or Eq. (9)) is often considered as being the standard LK solution. In practice, however, to gain more accuracy, it may be good idea to implement a Newton-Raphson-like iterative version of this scheme [17]. Furthermore, to better cope with large displacements, one must to implement LK in a multiresolution framework.

#### 2.1. Iterative LK

In this section, we present an iterative version of the traditional LK method, primarily inspired of the work by Bouguet [17] and Black and Anandan [15]. Let  $\vec{v}_s^k$  be the motion vector on site *s* after k - 1 iterations and  $\Delta \vec{v}_s^k$  the incremental motion vector computed during the *k*th iteration. Here, the goal is to estimate the  $\Delta \vec{v}_s^k$  that will best minimize the residual error

$$E(\vec{v}_s^{k+1}) = \sum_{r \in \eta_s} \left[ I(r + \vec{v}_s^k + \Delta \vec{v}_s^k, t) - I(r, t+1) \right].$$

$$(10)$$

According to Eq. (8), the *k*th motion increment can be computed by

$$\Delta \vec{v}_s^k = -(M_s)^{-1} \vec{b}_s^k \tag{11}$$
 where

$$\vec{b}_{s}^{k} = \begin{pmatrix} \sum_{r} W_{r} I_{T}^{r} I_{x} \\ \sum_{r} W_{r} I_{T}^{k} I_{y} \end{pmatrix}$$
$$I_{T}^{k} = I(r + \vec{v}_{s}^{k}, t) - I(r, t + 1)$$

where value  $I(r + \vec{v}_s^k, t)$  is computed with a bilinear interpolation. After k iterations, the motion vector on site s is given by

$$\vec{v}_s^{k+1} = \vec{v}_s^k + \Delta \vec{v}_s^k \tag{12}$$

or equivalently

$$\vec{v}_s^{k+1} = \vec{v}_s^k - M_s^{-1} \vec{b}_s^k. \tag{13}$$

#### 2.2. Multiresolution LK

As mentioned previously, a multiresolution framework is frequently a good solution to deal with large displacements. Its is also a good solution to help regularize the flow in noisy and/or textureless areas. When implementing a multiresolution framework, two pyramids based on I(t) and I(t + 1) need to be built. The process of building an image pyramid is often referred to as "image decimation" [42,43]. In most pyramid representation, the original image (of size  $\mathcal{N} \times \mathcal{M}$ ) appears at the bottom of the pyramid, *i.e.*, at level 0. The original image is first convoluted by a low-pass filter and then decimated by a factor of two in each dimension. The resulting image (of size  $\mathcal{N}/2 \times \mathcal{M}/2$ ) is then placed at level 1 of the pyramid. The same two operations are then applied to the image at level 1 to produce the next  $\mathcal{N}/4 \times \mathcal{M}/4$  pyramid level. This process is repeated up until when the desired number of pyramid level has been reached. In our application, as shown in Fig. 1(a), a  $3 \times 3$ low-pass Bartlett filter [42] is used. Some authors refer to this filter pyramidal or weighted average by some authors [44]. Since each level  $I^{L}$  has a size of  $\mathcal{N}/2^{L} \times \mathcal{M}/2^{L}$ , the pyramids can have up to  $\min[\log_2(\mathcal{N}), \log_2(\mathcal{M})]$  levels. However, our tests revealed that more than four levels does not provide any major advantage, at least for those sequences we worked with and for which motion never exceeds 5 pixels.

Once the two input pyramids have been computed, the flow is estimated from the highest level of the pyramid down to level 0.



**Fig. 1.** (a) The pyramid level  $I^{l+1}$  is obtained after convoluting  $I^{l}$  with *h* and decimating  $I^{L}$  by a factor 2 in each dimension. (b) When projecting a level L + 1down to level L, a simple bilinear interpolation method is used.

At each level, vector field  $V^L$  is iteratively estimated with Eq. (13) after which it is projected down to level L - 1. The downscaling operation is done with a bilinear interpolation as illustrated in Fig. 1(b). The iterative and multiresolution version of LK is presented in Algorithm 1. Notice that at the highest level, the flow is initialized with zero values.

Algorithm 1: Multiresolution and Iterative LK Algorithm

I(s,t)	Image at time t
V	Vector field
L	Pyramid-level index
$N_i$	Number of iterations

Build a n-level pyramid for 
$$I(t)$$
 and  $I(t + 1)$ .  
 $V \leftarrow 0$   
For  $L = Level_{Max}$  to 0 do  
compute  $I_x^L$  and  $I_y^L$   
For  $k = 0$  to  $N_i$  do  
For each site  $s \in S^L$  do  
compute  $(M_s^L)^{-1}$   
 $I_T^{L^k} \leftarrow I^L(s + \vec{v}_s^k, t) - I^L(s, t + 1)$   
Compute  $\vec{b}_s^{L^k}$   
 $\vec{v}_s^{k+1} \leftarrow \vec{v}_s^k - M_s^{L-1} \vec{b}_s^k$   
If  $L > 0$  then  
 $V^{L-1} \leftarrow$  downscale  $V^L$ 

#### 3. Our method

The LK algorithm is simple and generates fairly good results over a wide range of real image sequences, hence its popularity. However, it is widely accepted that its implementation suffers from two fundamental limitations. First, since (Eq. (13) and (8)) do not model the inherent uncertainties caused by noise and low contrast regions [16], the output vector field V may be locally inaccurate, especially if the neighborhood  $\eta_s$  is small. Second, the LK solution hardly takes account of multiple motions and thus generates blurry edges around moving objects. This is especially true when LK is implemented with a multiresolution framework.

#### 3.1. Dealing with uncertainties

Several solutions are conceivable to alleviate the problem of uncertainties. While some have adapted Eq. (13) to a probabilistic framework [16] to account for noise, others have replaced V by a piecewise-smooth vector field made up of parametric models [45-47]. Also, a variety of filters has been proposed, from simple median filters [46] to more elaborate Kalman-like filters [8,27].

For our method, every vector  $\vec{v}_s$  is considered as an "estimate" that is to be fused locally with its neighbors to yield a better result [27]. Assuming that  $\vec{v}_s$  has a 2 × 2 covariance matrix  $C_s$  proportional to the variance of the noise, the vectors surrounding site s can be fused with a linear combination [8,27] of the form

$$\vec{\nu}_{s} = P_{s} \sum_{i \in \zeta_{s}} C_{i}^{-1} \vec{\nu}_{i}$$

$$P_{s} = \left(\sum_{i \in \zeta_{s}} C_{i}^{-1}\right)^{-1}$$

$$(14)$$

where  $\zeta_s$  is a neighborhood window of size  $\mathscr{X} \times \mathscr{X}$  around site *s*. Here,  $C_i^{[-1]}$  can be seen as a "confidence measure" that give more or less influence to an estimate  $\vec{v}_i$ . In this way, a vector  $\vec{v}_i$  with a large confidence will influence more  $\vec{v}_s$  than another one with a smaller confidence. Here,  $C_i$  is a 2 × 2 covariance matrix computed as follows:

$$C_{s} = \begin{pmatrix} \sum_{t \in \beta_{s}} (u_{t} - u_{s})^{2} & \sum_{t \in \beta_{s}} (u_{t} - u_{s})(v_{t} - v_{s}) \\ \sum_{t \in \beta_{s}} (u_{t} - u_{s})(v_{t} - v_{s}) & \sum_{t \in \beta_{s}} (v_{t} - v_{s})^{2} \end{pmatrix}_{/A*A}$$
(15)

where  $\beta_s$  is a neighborhood window of size  $\Lambda \times \Lambda$  around *s* and  $\vec{v}_t = (u_t, v_t)$ . This matrix is similar to the one proposed by Singh [6]. Notice that such fusion procedure is sometimes referred to as a *Best Linear Unbiased Estimate* (BLUE) by some authors [27,48]. Combining Eqs. (13) and (14), the iterative LK procedure can be rewritten as

$$\vec{\nu}_{s}^{k+1} = P_{s}^{k} \sum_{i \in \zeta_{s}} C_{i}^{k^{-1}} \left( \vec{\nu}_{i}^{k} + M_{i}^{-1} \vec{b}_{i}^{k} \right).$$
(16)

This scheme makes sense intuitively since it encourages flow to propagate from high-confidence regions (regions with low covariance) to regions of low confidence. To make sure  $C_i$  is invertible, its eigenvalues are forced to be larger or equal to 0.001 (here in pixel-distance units). This is done *via* a singular value decomposition.

For applications in which processing time is a determining factor, the fusion procedure of Eq. (16) may be replaced by a simpler and yet faster, Gaussian filter.

#### 3.2. Dealing with multimodal motion

As previously mentioned, estimating motion with an isotropic approach such as LK often results in a flow with blurry edges, which explains why so many robust functions and anisotropic filters have thus far been proposed. In this contribution, the way multiple motion is handled is based on the following four assumptions:

- moving objects are textured enough to have their motion correctly estimated by a differential method;
- (2) motion boundaries are close to sharp intensity edges;
- (3) in regions with no sharp intensity edge, motion is locally invariant;
- (4) motion estimated away from flow discontinuities is reasonably accurate.

From these four assumptions (that are generally accepted in the literature [15,18]) a fundamental observation can be made: two close sites that are not separated by an intensity edge tend to have similar motion vectors. To illustrate this assertion, let us take the three sites s, p and c shown in Fig. 2. Each of these sites has a *true* motion vector  $\vec{r}_s, \vec{r}_p$  and  $\vec{r}_c^{-1}$  that are to be estimated by an optical flow method (here LK). Since Eq. (13) includes no edge information, a vector  $\vec{v}_s^k$  computed with the standard LK approach will be corrupted by the dual nature of the motion around *s*. In other words, because  $\vec{v}_s$  (and  $\vec{v}_c$ ) is estimated with a neighborhood window of size  $N \times N$ , the bimodal nature of motion around the site *s* will locally bias the flow. Since *s* is a close neighbor of *c*,  $\vec{v}_s \approx \vec{v}_c$  whereas  $\vec{v}_s \neq \vec{v}_c$ , hence the blurry contour. As a consequence, flow estimated with a least-squares fit method close to a sharp intensity edge is likely to be corrupted by multiple motions.

Also, with the above four assumptions, it may be assumed (without prior information on the true nature of the flow) that



**Fig. 2.** Synthetic motion returned by Eq. (13) *versus* the real motion boundary. The arrows point at three positions (*s*, *p* and *c*) for which  $\vec{r_s} \approx \vec{r_p}$  and  $\vec{r_s} \neq \vec{r_c}$ .

 $\vec{v}_s \approx \vec{v}_p$  and  $\vec{v}_s \neq \vec{v}_c$ . In fact, because no intensity edge separates s and p, motion has to change smoothly between these neighbors and thus  $\vec{v}_s \approx \vec{v}_p$ . Similarly, since s and c are separated by a sharp intensity edge, it cannot be assumed that  $\vec{v}_s = \vec{v}_c$ , hence why  $\vec{v}_s \neq \vec{v}_c$ . Of course, there are many cases for which two close sites separated by an intensity edge have the same motion (a background made of stationary objects is a good example). Thus, from the previous four assumptions, one can assume that site s and c are *likely* (but not certain) to have different motion.

From these observations, it may be inferred that:

(1) when two neighbors *s* and *p* are not separated by an intensity edge and *s* is closer than *p* to a motion boundary, then  $||\vec{v}_s - \vec{r'}_s|| \ge ||\vec{v}_p - \vec{r'}_s||$ . In other words, a motion vector estimated away from a strong intensity edge is more likely to be accurate;



**Fig. 3.** Zoom on a frame of MOM AND DAUGHTER sequence. Every vector shows the estimated mean-shift displacement between a site s and a site p.

<sup>&</sup>lt;sup>1</sup>  $\mathscr{V}$  is the *true* vector field whereas v is the one estimated by LK.

(2) when the distance between a site *p* and the closest motion boundary is larger than *N* (*N* being the size of  $\eta_s$ ), then  $\vec{v}_p \approx \vec{\psi_p}$  (a similar conclusion was already stated by Thompson in [18]).

Consequently, to fight against the influence of multiple motions and thus keep sharp edges, Eq. (16) will be rewritten as:

$$\vec{\nu}_{s}^{k+1} = P_{\delta}^{k} \sum_{i \in \zeta_{\delta}} C_{i}^{k^{-1}} \left( \vec{\nu}_{i}^{k} + M_{i}^{-1} \vec{b}_{i}^{k} \right)$$
(17)

where

$$\delta = \begin{cases} s & \text{if } \mathcal{D}(s) \ge N \\ p & \text{otherwise} \end{cases}$$
(18)

where *N* denotes the LK window size (see Eq. (1)). Here,  $\mathscr{D}(\cdot)$  denotes the distance to the nearest intensity edge and *p* is a neighbor of site *s* located further away from that same edge (see Fig. 3). The latter equations may be understood as follows: when *s* is close to an intensity edge,  $\vec{v}_s$  estimated with the standard LK approach is likely to be corrupted by multiple motions. It is thus preferable to compute  $\vec{v}_s$  with a neighborhood window  $\eta_p$ , shifted away from the nearest intensity edges. In this way,  $\vec{v}_s$  is computed with a neighborhood that is more likely to contain a unimodal motion.

As previously mentioned, in real-life scenarios, many strong edges may not correspond to a motion discontinuity. In these cases, estimating the flow with Eq. (17) could seem, at first glance, to be a source of error. Fortunately though, when an intensity edge does not correspond to a motion discontinuity, this means that *s* and *p* are located in an area of uniform motion. Thus, since *s* and *p* are neighbors, motion is likely to be nearly constant between them in such a way that  $\vec{\forall}_s \approx \vec{\forall}_p$ . Also, the fusion procedure of Eq. (17) helps significantly constrain the flow and thus reduce, if not prevent, error propagation.

#### 3.3. Mean-shift

As shown in Eq. (18), when  $\mathscr{D}(s) < N$ ,  $\vec{v}_s^k$  is computed over a neighboring site *p*, located further away from the nearest intensity edge. In this way,  $\vec{v}_s$  will be less likely to be corrupted by multiple motion. From Section 3.2's four assumptions, a good site *p* must respect the following two criteria

- (1) p must be a neighbor of site s with  $\mathscr{D}(p) \ge \mathscr{D}(s)$  (assumptions 2 and 4).
- (2)  $\|\nabla I(p,t)\| \approx 0$  (assumptions 3).

From these criteria, we found that the *mean-shift* procedure [39] offers an appropriate strategy to determine p given s and I(t). Mean-shift is a simple iterative non-parametric estimator of density gradient that was first introduced by Fukunaga and Hostetler [49] and adapted to imagery by Comaniciu and Meer [39]. Mean-shift is based on the multivariate kernel density estimate

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{j=1}^n K\left(\frac{x - x_j}{h}\right)$$
(19)

where  $K(\cdot)$  is a kernel of radius h and  $\{x_j\}_{j=1...n}$  is a set of n points of dimension d. With this density estimate, the density gradient can be expressed as

$$\nabla \hat{f}(x) = \frac{1}{nh^d} \sum_{j=1}^n \nabla K\left(\frac{x - x_j}{h}\right)$$
(20)

where, as suggested by Comaniciu and Meer [39],  $K(\cdot)$  can be replaced by the Epanechnikov kernel. However, as mentioned in a

second paper [50], other kernel functions can be used. With the Epanechnikov kernel, the last equation can be redefined as

$$\nabla \hat{f}(x) = \frac{n_x}{n(h^d c_d)} \frac{d+2}{h^2} M_h(x) \tag{21}$$

where  $M_h(x)$  is called the sample mean-shift, i.e.,

$$M_{h}(x) = \frac{1}{n_{x}} \sum_{x_{j} \in S_{h}(x)} (x_{j} - x)$$
(22)

where  $S_h(x)$  is a *d*-dimensional hypersphere of radius *h* and of volume  $c_d$ . This hypersphere is centered on *x* and contains  $n_x$  points. Mean-shift is thus a fairly simple iterative procedure as shown in the following algorithm.

#### Algorithm 2: Mean-Shift Algorithm

Image at time tI(t)PVector field containing the meanshift vector of every pixel. P = 0do  $P_{prev} = P$ for each pixel x in I(t) do Get every pixel located inside the hypersphere  $S_h(x)$ With these pixels, compute the sample mean-shift  $M_h(x)$ Translate  $S_h(x)$  by  $M_h(x)$  $\vec{P_s} = \vec{P_s} + M_h(x)$ while  $||P - P_{prev}|| >$  accuracy threshold

When using mean-shift to filter an image, the iterative procedure is applied on data  $x_i$  located in a so-called *spatial-range* domain. The spatial domain refers to the 2D space of lattice S while the range domain refers to the pixel color/intensity level. In this context, each site  $s \in S$  is associated with a point  $x_s$  in a *d*-dimensional spatial-range domain. Here, the first two dimensions of  $x_s$  correspond to the  $I \times I$ spatial Euclidean coordinates while the other dimensions correspond to the intensity (or color) observed over site s. Thus, d is always set to 5  $(i, j, Red_s, Green_s, Blue_s)$  for color images and to 3 (*i*, *j*, *Intensity*<sub>s</sub>) for grayscale images. After successive mean-shift iterations, the hypervolume  $S_h$  is shifted from its initial location  $x_s$  to a final position  $x_p$  where the local gradient is null. We define the *mean-shift vector* as vector  $\vec{P}_s$  linking site *s* to site *p*:  $p = s + \vec{P}_s$ . This last relation is the one we have chosen to use in Eq. (18). By the very nature of mean-shift, p is always located further away from the nearest intensity edge than *s* and  $||\nabla f(x_p)|| \approx 0$ . Also, in general, the stronger the intensity gradient is around site s, the larger  $\vec{P}_s$  will be. These are the reasons why we consider that mean-shift meets the two criteria presented at the beginning of this section. To make sure p is a neighbor of s, the length of  $\vec{P}_s$  is clamped to a maximum value:  $||\vec{P}_s|| = \min(||P_s||, N)$ . The reason why the mean-shift vector length is limited to N can be understand as follows. As illustrated in Fig. 2 and mentioned in the second observation of Section 3.2, a pixel located at a distance  $\ge N$  of the nearest edge is less likely to be corrupted by multiple motion. On the other hand, a motion vector  $\vec{v}_p$  estimated with a window located too far away from its original pixel *s* will violate the first observation of Section 3.2. Therefore, by making sure the mean-shift displacement is limited to N, both observations are being respected. For more details on mean-shift, please refer to [39].



Fig. 4. Synthetic sequences. (a) YOSEMITE WITH SKY (b) YOSEMITE WITHOUT SKY and (c) the shapes sequence.

Fig. 3 shows a mean-shift vector field *P* with vectors linking site *s* to site *p*. As can be seen, the closer to the edge a site *s* is, the larger the mean-shift displacement is.

#### 4. Other methods implemented

In Section 5, the proposed method is compared with eight other optical flow methods whose objective is to preserve sharp motion discontinuities (except for Horn and Schunck). In this section, we review in detail these methods and underline how they were implemented numerically.

#### 4.1. Horn and Schunck

We have compared our method with differential optical flow methods among which some seek to preserve sharp discontinuities. All differential methods we have implemented are based on the data conservation constraint:

$$I_{x}u + I_{y}v + I_{T} = 0 (23)$$

where  $I_x$  and  $I_y$  are the spatial derivatives,  $I_T$  the temporal derivative and (u, v) the horizontal and vertical image velocity at a site  $s \in S$ . Because this formulation involves one equation and two unknowns, it admits an infinite number of solutions. As previously mentioned, the standard solution for handling this ill-posed problem is to add a spatial coherence term  $E_{sp}$  whose essential role is to constrain the solution. One common formulation of  $E_{sp}$  is the membrane model proposed by Horn and Schunck [12]

$$E_{sp}(\vec{\nu}) = ||\nabla u||^2 + ||\nabla \nu||^2$$
(24)

or, when considering the discrete version of  $\nabla$ ,

$$E_{sp}(\vec{\nu}) = \frac{1}{4} \sum_{r \in \mathscr{G}_s} \left[ (u_r - u_s)^2 + (\nu_r - \nu_s)^2 \right]$$
(25)

where  $\mathscr{G}_s$  contains the four neighbors of size *s*. In this context, the energy function  $E(\vec{v})$ , to be minimized at every site  $s \in S$ , is represented by

$$E(\vec{\nu}) = (I_x u + I_y \nu + I_T)^2 + \frac{\alpha}{4} \sum_{r \in \mathscr{G}_s} \left[ (u_r - u_s)^2 + (\nu_r - \nu_s)^2 \right]$$
(26)

#### Algorithm 3: Our Method Processing one Pyramid Level

I(t)	Image at time t
$P_s$	Mean-shift vector
$V, V^{pre}$	<sup>ev</sup> Vector fields
$C_s$	Variance-covariance matrix of site a
$N_l$	Noise Level

 $P \leftarrow$  Apply mean-shift on I(t)

```
for each site s \in S do

rnd \leftarrow random value between 0 and N_l

I(s,t), I(s,t+1) += rnd

\nabla I \leftarrow Spatial gradient of I(t)

V, V^{prev} \leftarrow 0

do

I_T \leftarrow I(s + V^{prev}, t) - I(s, t+1)

/* Motion Estimation */

for each site s \in S do
```

for each site  $s \in S$  do  $\delta \leftarrow s + P_s$ Compute  $M_{\delta}^{-1}$  and  $\vec{b}_{\delta}^k$   $\vec{v}_s \leftarrow \vec{v}_s^{prev} + M_{\delta}^{-1}\vec{b}_{\delta}^k$   $V^{prev} \leftarrow V$ /\* Covariance filtering \*/ Compute  $C_s, \forall s \in S$ for each site  $s \in S$  do  $\delta \leftarrow s + P_s$  $\vec{v}_s \leftarrow P_{\delta} \sum_{i \in \zeta_{\delta}} C_i^{-1} \vec{v}_i^{prev}$ 

while  $||V - V^{prev}|| >$ accuracy threshold

where  $\alpha$  is a constant giving more or less importance to the spatial constraint. With the Euler–Lagrange equations, the underlying computation in Eq. (26) reduces to a Jacobi optimization, namely

$$u^{[k+1]} = \bar{u}^{[k]} - I_x \frac{I_x u^{[k]} + I_y v^{[k]} + I_T}{\alpha^2 + I_x^2 + I_y^2}$$
(27)

$$v^{[k+1]} = \bar{v}^{[k]} - I_y \frac{I_x u^{[k]} + I_y v^{[k]} + I_T}{\alpha^2 + I_x^2 + I_y^2}$$
(28)

where  $(\bar{u}, \bar{\nu})$  is the local mean of the four nearest neighbors *s*. In our implementation, a total of 1000 iterations is used and  $\alpha$  is set between 5 and 30 depending on the sequence.

#### 4.2. Black and Anandan

It is well known that the quadratic functions of Eq. (26) give an overwhelming importance to outliers which is a major cause of blurry edges. Among the first *robust* formulation of Eq. (26) was Black and Anandan's proposal [15]. The robust gradient-based formulation they proposed has the following form:

$$E(\vec{v}) = \rho(I_x u + I_y v + I_T, \sigma_D) + \alpha \sum_{r \in \mathscr{G}_s} [\rho(u_r - u_s, \sigma_S) + \rho(v_r - v_s, \sigma_S)]$$

where, in our implementation, the importance function  $\rho$  is the Lorentzian function [15]

Table 1

Results: YOSEMITE sequence with sky.

Technique	$\psi_E$	$\sigma_{\psi_E}$	$\psi_E^e$	$\sigma_{\psi^e_E}$
Horn and Schunck	7.35	9.77	_	-
LK (Algorithm 1)	8.54	14.94	-	-
Black	5.94	9.31	-	-
IB-Iso	8.25	9.16	-	-
IB-Aniso	8.28	11.55	-	-
FB-Iso	7.16	12.80	-	-
FB-Aniso	8.65	10.31	-	-
HS–LK	7.9	11.20	-	-
Our method	6.20	13.7	-	-

Table 2

Results: YOSEMITE sequence without sky.

Technique	$\psi_E$	$\sigma_{\psi_E}$	$\psi^e_E$	$\sigma_{\psi^e_E}$
Horn and Schunck	3.50	4.44	-	_
LK (Algorithm 1)	3.42	3.65	-	-
Black	2.30	1.66	-	-
IB-Iso	5.87	6.31	-	-
IB-Aniso	4.35	5.51	-	-
FB-Iso	2.92	3.20	-	-
FB-Aniso	4.96	6.45	-	-
HS–LK	3.66	3.21	-	-
Our method	1.81	1.78	-	-

Table 3

K	lesul	ts:	TRANSLATING	SHAPES	sequence.	

Technique	$\psi_E$	$\sigma_{\psi_E}$	$\psi_E^e$	$\sigma_{\psi^e_E}$
Horn and Schunck	18.7	22.1	31.1	27.6
LK (Algorithm 1)	8.9	19.0	25.8	27.3
Black	10.7	16.8	24.4	22.5
IB-Iso	9.5	15.7	23.7	19.6
IB-Aniso	14.0	19.1	29.2	23.8
FB-Iso	8.0	14.4	19.7	19.8
FB-Aniso	7.2	13.2	18.6	18.2
HS–LK	5.0	12.2	13.0	18.7
Our method	2.3	10.0	6.9	17.3

$$\rho(\mathbf{x},\sigma) = \log\left(1 + \frac{1}{2}\left(\frac{\mathbf{x}}{\sigma}\right)^2\right). \tag{29}$$

As suggested by Black and Anandan,  $E(\vec{v})$  can be minimized with a multiresolution Successive Over-Relaxation method with a fixed number of iterations (20 in our case) on each level of the pyramid. Following Black and Anandan's recommendations, the values of  $\sigma_D$  and  $\sigma_S$  are also lowered according to an annealing schedule [15]. This schedule comprises six stages for which  $\sigma_D$  and  $\sigma_S$  vary, respectively, from  $18/\sqrt{2}$  to  $5/\sqrt{2}$  and  $3/\sqrt{2}$  to  $0.03/\sqrt{2}$ . The variable  $\alpha$  was set to 0.2.



Fig. 5. Realistic sequences. (a) translating tree (b) diverging tree (c) rotating bonsai (d) cars over park and (e) parthenon.



Fig. 6. Real sequences. (a) claire (b) Karlsruhe (c) mom and daughter (d) taxi and (e) flower.



Fig. 7. Results for the TRANSLATING SHAPES sequence.

#### Table 4

Results: TRANSLATING TREE sequence.

Technique	$\psi_E$	$\sigma_{\psi_E}$	$\psi_E^e$	$\sigma_{\psi^e_E}$
Horn and Schunck	1.83	2.15	-	-
LK (Algorithm 1)	1.65	4.20	-	-
Black	1.14	1.03	-	-
IB-Iso	4.21	3.84	-	-
IB-Aniso	1.89	1.41	-	-
FB-Iso	1.30	1.64	-	-
FB-Aniso	2.00	2.10	-	-
HS–LK	3.71	3.80	-	-
Our method	0.80	1.68	-	-

#### Table 5

Results: DIVERGING TREE sequence.

Technique	$\psi_E$	$\sigma_{\psi_E}$	$\psi^e_E$	$\sigma_{\psi^e_E}$
Horn and Schunck	2.30	2.06	-	_
LK (Algorithm 1)	4.60	3.39	-	-
Black	2.25	1.30	-	-
IB-Iso	6.40	5.01	-	-
IB-Aniso	2.50	3.85	-	-
FB-Iso	2.16	1.79	-	-
FB-Aniso	2.69	2.47	-	-
HS–LK	5.00	3.39	-	-
Our method	2.48	1.71	-	-

#### Table 6

Results: realistic CARS OVER PARK sequence.

Technique	$\psi_E$	$\sigma_{\psi_E}$	$\psi^e_E$	$\sigma_{\psi^e_E}$
Horn and Schunck	12.0	18.7	33.0	31.2
LK (Algorithm 1)	9.5	20.6	25.9	38.4
Black	8.0	5.7	25.4	28.4
IB-Iso	13.9	18.4	32.8	29.0
IB-Aniso	10.1	19.2	29.0	35.7
FB-Iso	7.8	15.2	20.0	29.2
FB-Aniso	7.2	16.6	22.1	32.3
HS-LK	9.2	18.3	27.2	34.4
Our method	4.0	16.9	16.2	34.1

## Table 7

Results: realistic PARTHENON sequence.

Technique	$\psi_E$	$\sigma_{\psi_E}$	$\psi^e_E$	$\sigma_{\psi^e_E}$
Horn and Schunck	12.2	18.5	18.2	21.7
LK (Algorithm 1)	14.7	24.1	22.5	28.3
Black	10.4	18.1	15.5	21.4
IB-Iso	16.0	17.6	21.3	19.0
IB-Aniso	14.1	21.2	18.7	23.8
FB-Iso	10.7	18.0	16.1	21.0
FB-Aniso	11.0	16.9	17.0	19.5
HS–LK	13.1	16.7	17.3	19.4
Our method	9.80	20.7	15.2	25.5

Table 8	B
---------	---

Results: realistic ROTATING BONSAI sequence.

Technique	$\psi_E$	$\sigma_{\psi_E}$	$\psi^e_E$	$\sigma_{\psi^e_{\scriptscriptstyle E}}$
Horn and Schunck	22.1	22.3	36.1	25.2
LK (Algorithm 1)	10.4	21.5	25.4	28.1
Black	14.1	19.4	30.1	23.0
IB-Iso	15.6	18.5	28.9	21.1
IB-Aniso	22.8	21.9	37.0	25.1
FB-Iso	13.7	19.2	29.1	22.3
FB-Aniso	11.8	20.9	27.2	25.3
HS–LK	8.8	15.8	20.3	20.9
Our method	5.3	15.0	13.2	20.1

#### 4.3. Image-based isotropic regularization

Although Horn and Schunck's method enjoys a great deal of popularity and is still frequently used, as mentioned previously its "blind" quadratic regularizer  $E_{sp}$  is well known to blur-out motion discontinuities. Consequently, variational methods with a different regularization term have been proposed to better preserve flow in those areas. A survey of some of these methods has been published by Weickert and Schnörr [23].

One such method we have implemented uses an image-based robust isotropic regularization factor similar to the one proposed by Alvarez et al. [35]:

$$E_{sp} = \Psi(||\nabla I||^2)(||\nabla u||^2 + ||\nabla v||^2).$$
(30)

The goal of this regularization function is to strongly regularize flow in textureless areas and reduce smoothing at image boundaries. Since  $\Psi(\cdot)$  is a decreasing and strictly positive function, the Charbonnier function has been implemented:  $\Psi(s^2) = 1/\sqrt{1 + s^2/\lambda^2}$ . In this way, as opposed to Horn and Schunck's method which uniformly regularizes the flow, the  $\Psi$  function prevents Alvarez' method from smoothing across an image edge. In other words, in the vicinity of an image edge (*i.e.*, where  $||\nabla I||$  is large) the influence of the prior function  $E_{sp}$  is significantly reduced. This makes the method less prone to smooth out the flow where there is likely to be a motion discontinuity.

As mentioned by Alvarez et al. [35], following the Euler–Lagrange equations, the to-be-optimized optical flow (u, v) must satisfy the following diffusion–reaction system at its steady state (*i.e.*,  $k \to \infty$ )

$$\frac{\partial u}{\partial k} = \alpha \operatorname{div}(\Psi(||\nabla I||^2) \nabla u) - I_x(I_x u + I_y \nu + I_T)$$
(31)

$$\frac{\partial \nu}{\partial k} = \alpha \operatorname{div}(\Psi(||\nabla I||^2) \nabla \nu) - I_y(I_x u + I_y \nu + I_T)$$
(32)

where k denotes an artificial evolution parameter that should not be confused with time variable t of the image sequence. This system can be solved with a modified-explicit scheme whose numerical structure is given by

$$\frac{u^{[k+1]} - u^{[k]}}{\tau} = I_x (I_x u^{[k+1]} + I_y v^{[k]} + I_T) - \alpha A (u^{[k]}, v^{[k]})$$
(33)

where  $A(u^k)$  is the numerical approximation of  $\operatorname{div}(\alpha \Psi(||\nabla I||^2)\nabla u)$ , k is the iteration index and  $\tau$  is the optimization time step (again, not to confuse with the time variable t or the temporal derivative index T) that we set to 0.25. Note that this optimization scheme comes from Weickert and Schnörr's (2001) paper [23] and will be used



Fig. 8. Results for the CARS OVER PARK sequence. The red channel contains the magnitude of the vectors pointing upward and the blue channel contains the magnitude of the vectors pointing downward. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)



Fig. 9. Results for the ROTATING BONSAI sequence.

for *flow-based isotropic* and the *flow-based anisotropic* methods. Here, a total of 1000 iterations are used together with  $\alpha$  set between 100 and 400 and  $\lambda$  between 0.5 and 10. Variables  $\alpha$  and  $\lambda$  are manually adjusted to each video sequence. Notice that v is optimized in similar way.

#### 4.4. Image-based anisotropic regularization

The anisotropic image-based variational method we have implemented is the one proposed by Nagel and Enkelmann [14,36] which uses second-order spatial derivatives to constrain the flow. The key idea of their method is the use of an "oriented smoothness" constraint that suppresses smoothing across image edges while encouraging smoothing along image edges. The energy function their method minimizes has the following shape [14,3]:

$$E(\vec{\nu}) = (I_x u + I_y \nu + I_T)^2 + \alpha^2 \operatorname{tr}((\nabla \vec{\nu})^{\mathrm{T}} W(\nabla \vec{\nu}))$$
(34)

with

$$W = \frac{1}{I_x^2 + I_y^2 + 2\gamma} \begin{pmatrix} I_y^2 + \gamma & -I_x I_y \\ -I_x I_y & I_x^2 + \gamma \end{pmatrix}$$
(35)

where  $\gamma$  and  $\alpha$  are two constants set between 2 and 5 depending on the nature of the flow to be estimated and "tr" is the trace operator. This functional may be minimized by a Gauss–Seidel optimizer as explained in [3,14]. All details of our implementation have been taken from Appendix A of Barron et al.'s technical report [51]. However, in contrast to what Barron et al. [3] suggested, a fixed number of 500 iterations have been used.

#### 4.5. Flow-based isotropic regularization

Another kind of modification to Horn and Schunck's variational model leads to the so-called "flow-based" methods [21–23,52,53]. These isotropic flow-based methods are implemented on top of a robust regularizer that reduces the influence of outliers. The method we have implemented is similar to the one proposed by Weickert [22] and to the *2D version* of Weickert and Schnörr [52] spatiotemporal method. For this method, the energy functional to be minimized is expressed as:

$$E(u, v) = (I_x u + I_y v + I_T)^2 + \alpha \Psi(||\nabla u||^2 + ||\nabla v||^2)$$
(36)

where  $\Psi(\cdot)$  is a robust function. In this way, this flow-based isotropic method strongly regularizes the flow in areas where  $||\nabla u||^2$  and  $||\nabla v||^2$  are small and reduce regularization where  $||\nabla u||^2$  and/or  $||\nabla v||^2$  is strong, *i.e.*, in the vicinity of a motion discontinuity.

The gradient descent of this functional leads to the reaction–diffusion system:

$$\frac{\partial u}{\partial k} = \nabla \left( \Psi'(||\nabla u||^2 + ||\nabla v||^2) \nabla u \right) - \frac{I_x}{\alpha} (I_x u + I_y v + I_T)$$
(37)

$$\frac{\partial \boldsymbol{\nu}}{\partial \boldsymbol{k}} = \nabla \left( \Psi'(||\nabla \boldsymbol{u}||^2 + ||\nabla \boldsymbol{\nu}||^2) \nabla \boldsymbol{\nu} \right) - \frac{I_y}{\alpha} (I_x \boldsymbol{u} + I_y \boldsymbol{\nu} + I_T)$$
(38)

where  $\Psi'(s^2)$  is the derivative of  $\Psi(s^2)$  (with respect to  $s^2$ ) that we set equal to  $\frac{1}{\sqrt{(1+s^2/\lambda^2)}}$  as suggested by Weickert [22]. As is the case for the image-based isotropic method, we resort to a modified-explicit scheme [52] to perform the gradient descent. A total number of 1000 iterations is used, with  $\lambda$  set between 0.01 and 0.1 and  $\alpha = 300$ .



Fig. 10. Results for the PARTHENON sequence.

#### 4.6. Flow-based anisotropic regularization

Flow-based anisotropic regularization motion estimation methods are similar to the previously introduced anisotropic imagebased methods. The flow-based anisotropic methods encourage regularization *along* flow discontinuities while discouraging regularization *across* flow discontinuities [23,54,55]. This family of methods is thus clearly different from the flow-based isotropic methods which only prevent smoothing across motion discontinuity. The anisotropic method we have implemented was taken from Weickert and Schnörr [23] which minimizes the following functional:

$$E(u, v) = (I_x u + I_v v + I_T)^2 + \alpha \operatorname{tr} \Psi(\nabla u \nabla u^{\mathrm{T}} + \nabla v \nabla v^{\mathrm{T}})$$
(39)

where  $\Psi(\cdot)$  is a robust function. Using the Euler–Lagrange theorem, the gradient descent of this functional leads to the reaction–diffusion system [23]

$$\frac{\partial u}{\partial k} = \frac{I_x}{\alpha} (I_x u + I_y v + I_T) + \operatorname{div}(\Psi'(J)\nabla u)$$
(40)

$$\frac{\partial v}{\partial k} = \frac{I_y}{\alpha} (I_x u + I_y v + I_T) + \operatorname{div}(\Psi'(J)\nabla v)$$
(41)

where

 $J = \nabla u \nabla u^{\mathrm{T}} + \nabla v \nabla v^{\mathrm{T}} \tag{42}$ 

$$\Psi'(J) = \Psi'(\sigma_1)\vec{\mu}_1\vec{\mu}_1^{\rm T} + \Psi'(\sigma_2)\vec{\mu}_2\vec{\mu}_2^{\rm T}$$
(43)

where  $\sigma_1, \sigma_2$  and  $\vec{\mu}_1, \vec{\mu}_2$  are the eigenvalues and the eigenvectors of *J*, and  $\Psi'(s^2)$  is fixed to  $\frac{1}{\sqrt{(1+s^2/z^2)}}$ . The optimization is performed through a modified-explicit scheme [52] together with a total of 1000 iterations. The variable  $\lambda$  is set between 0.01 and 0.1 and  $\alpha$  between to 100 and 500. As is the case for the other methods, these values are adjusted according to the nature of the flow to be estimated.

#### 4.7. Combined local-global method

The traditional Lucas–Kanade method which estimates flow with a simple matrix inversion (see Eq. (8)) is clearly a local approach since it provides no means to propagate flow. However, a basic iterative scheme such as the one of Algorithm 1, or a more elaborate one such as the one we propose in Algorithm 3, can allow a LK-based method to propagate flow. Thus, we believe that our method should be compared with a recently-published method which explicitly combines the Lucas–Kanade and Horn and Schunck method into a synthetic "local–global" method [56]. Using the notations

$$\begin{split} \vec{w}_{s} &= (u_{s}, v_{s}, 1)^{\mathrm{T}} \\ ||\vec{w}_{s}||^{2} &= ||\nabla u_{s}||^{2} + ||\nabla v_{s}||^{2} \\ \nabla_{3} I &= (I_{x}, I_{y}, I_{T})^{\mathrm{T}} \\ J_{\rho}(\nabla_{3} I) &= W_{\rho} * (\nabla_{3} I \nabla_{3} I^{\mathrm{T}}) \end{split}$$
(44)



Fig. 11. Results for the FLOWER sequence.

where  $W_{\rho}$  is a Gaussian kernel with standard deviation  $\rho$  and  $J_{\rho}(\nabla_3 I)$  is a *structure tensor*. The authors argue that the Lucas–Kanade and Horn and Schunck energy functionals can be, respectively, expressed as

$$E_{\rm LK}(\vec{w}) = \vec{w}^{\rm T} J_{\rho}(\nabla_3 I) \vec{w} \tag{45}$$

$$E_{\rm HS}(\vec{w}) = \int \vec{w}^{\rm T} J_0(\nabla_3 I) \vec{w} + \alpha ||\nabla w||^2 \, dx dy \tag{46}$$

where  $J_0(\nabla_3 I)$  is a structure tensor with a zero standard deviation. A local–global energy function can thus be obtained by simply replacing  $J_0(\nabla_3 I)$  by  $J_\rho(\nabla_3 I)$  with  $\rho > 0$ :

$$E_{\rm HS-LK}(\vec{w}) = \int \vec{w}^{\rm T} J_{\rho}(\nabla_3 I) \vec{w} + \alpha ||\nabla w||^2 \, dx dy.$$
(47)

As can be seen,  $E_{\rm HS-LK}$  implements two spatial coherence constraints: an integration window (first term) and a prior function (second term). These two constants implicitly compensate for their mutual limitations and thus generate a better constrained flow. The authors argue that more accurate results can be obtained with a non-quadratic variation of  $E_{\rm LK-HS}$  whose formulation is as follows:

$$E_{\mathrm{HS-LK}}(\vec{w}) = \int \Psi_1(\vec{w}^{\mathrm{T}} J_{\rho}(\nabla_3 I) \vec{w}) + \alpha \Psi_2(||\nabla w||^2) dx dy$$
(48)

where  $\Psi_1(s^2)$  and  $\Psi_2(s^2)$  are non-quadratic robust functions. All results reported in this section with the subscript "HS–LK" have been obtained after minimizing Eq. (48). As mentioned by the authors, the Euler–Lagrange equations of  $E_{\text{HS-LK}}$  are given by

$$\begin{aligned} \mathbf{0} &= \operatorname{div}(\Psi_{2}'(||\nabla_{3}w||^{2})\nabla u) \\ &- \frac{1}{\alpha}\Psi_{1}'\Big(\vec{w}^{\mathsf{T}}J_{\rho}(\nabla_{3}I)\vec{w}\Big)(W_{11}u + W_{12}v + W_{13}) \\ \mathbf{0} &= \operatorname{div}(\Psi_{2}'(||\nabla_{3}w||^{2})\nabla v) \\ &- \frac{1}{\alpha}\Psi_{1}'\Big(\vec{w}^{\mathsf{T}}J_{\rho}(\nabla_{3}I)\vec{w}\Big)(W_{21}u + W_{22}v + W_{23}) \end{aligned}$$

where  $W_{ij}$  is a component of the structure tensor  $J_{\rho}(\nabla_3 I)$ . In our implementation,  $\Psi_i(s^2) = \frac{1}{\sqrt{1+s^2/\lambda_i^2}}$  and the vector field v is computed with a Successive Over-Relaxation optimization scheme [56]. The number of iterations is set to 1000 and variables  $\lambda_1$  and  $\lambda_2$  are assigned values ranging between 0.05 and 0.005, depending on the nature of the scene.

Bruhn et al. [56] stipulate that the method can be made "3D" through the use of spatio-temporal filters. Although this modification may improve motion estimation, to make the comparison style fair for the other methods implemented in a "2D" style, we have implemented what Bruhn et al. call the *non-quadratic 2D* energy function (here Eq. (48)).



Fig. 12. Results for the TAXI sequence.

#### 5. Results

In this section, an extensive set of results obtained with our method and the ones introduced in the previous section are presented. These results have been obtained on synthetic, realistic, and real sequences. For those sequences with a ground truth vector field, two angular error metrics taken from Barron et al. [3] are used to provide a quantitative measure of quality. For the ones with unknown ground truth, vector fields snapshots are provided.

#### 5.1. Setup and implementation details

In order to correctly gauge performance, all optic flow methods were implemented in a similar matter. They share the same spatial and temporal gradient function, the same multiresolution framework and they process video sequences that were pre-filtered by the same low-pass spatio-temporal filter. In fact, all image sequences presented in this paper have been pre-filtered by a spatio-temporal Gaussian filter. For each sequence, the spatial standard deviation of the filter is set to 1.5 whereas, the temporal standard deviation is set to 1.5 for sequences vosemite, translation TREE, DIVERGENCE TREE, CARS OVER PARK, FLOWER, and MOM AND DAUGHTER and to 0.5 for sequences ROTATING BONSAI, PARTHENON, TRANSLATING SHAPES, CLAIRE, TAXI, and KARLSRUHE. The spatial derivatives in *x* and *y* used by every method is approximated with a central difference. As for the temporal derivative, a simple two-frame difference is used through every sequence.

Since our method is built upon a multiresolution framework, each method has also been implemented in a multiresolution fashion in order to make the comparison fair for every approach. Here, the multiresolution framework is the same for every method: the coarse-scale solution obtained at level L serves as initial data for estimating the flow at the level L - 1. The reader should be aware that the number of pyramid levels is a fundamental issue when estimating motion. For instance, we observed that for scenes with *global* motion such as *Yosemite* or the *Translating Tree* sequence, a high pyramid with many levels improves estimation of the flow. On the other hand, scenes with *local* motion, *i.e.*, motion with small-scale details such as the *Taxi* or the *Bonsai* sequences for instance, a pyramid implementation with fewer levels is preferable. For these reasons, the number of pyramid levels used by each method has been manually adjusted to each sequence. Typically, the number of levels is set between 1 and 3 for each method.

Since a uniform implementation framework is used for every method, some results reported in this paper differ slightly from those reported in previous papers. For instance, our implementation of Horn and Schunck's method produces, for the *Yosemite* sequence, an average angular error of 7.35 as opposed to 11.26 in Barron et al. [3]. This is mainly due to the fact that our implementation of HS has been made multiresolution.

Also, when we implemented these methods, we first used the parameter sets provided by the authors. However, we came to realize that these parameters are not well suited for every sequences. For instance, the  $\alpha$  regularization term of the Horn and Schunck method needs to be large when estimating a "global" optical flow (*i.e.*, a vector field with little or no motion discontinuities) such as the TRANSLATING TREE, the DIVERGING TREE and the YOSEMITE SEQUENCES. Indeed, when  $\alpha$  is large, the flow diffusion is maximal and regions with little or no texture can be compensated by a



Fig. 13. Results for the MOM AND DAUGHTER sequence.

strong regularization. On the other hand, using a large  $\alpha$  value on "local" sequences such as KARLSRUHE, TAXI and the MOVING SHAPES, has the effect of over-regularizing the flow and thus generate "superblurry" motion fields. Thus, for those local sequences, a smaller  $\alpha$  value is better suited. In other words, a good set of parameters for a global sequence is not necessarily appropriate for a local sequence. Thus, to fairly compare the methods, we believe that the parameters of every method need to be adjusted to the content of the sequence. Using only one parameter set per method would be harmful for most methods. Unfortunately, since most original papers do not provide a parameter set for local and global sequences, we had to heuristically adjust it. Consequently, the parameters were optimized to obtain the best results based on the content of the scenes.

Let us also mention that since processing speed is not the core of this paper, a large number of iterations (sometimes much greater than necessary) has been given to every method. This is to ensure that every method converges towards a stable solution. In most cases, the specific number of iterations has been taken from the original papers and, in some cases, increased slightly.

For our method,  $\mathscr{X} = \Lambda = N = 9$  or  $11^2$  (for Algorithm 1,  $N \in \{7, 9, 11\}$ ) and the noise level bias (variable  $N_l$  in Algorithm 3) between 2.0 and 5.0 for every sequence.

Notice that the noise level has also been used in the implementation of Algorithm 1 to make sure LK always produces vector fields of 100% density. In fact, the Algorithm 1 is identical to our method except for the edge-avoidance procedure and the fusion procedure (Eq. (16)). In this way, the results clearly illustrate the difference between our method and the traditional iterative LK algorithm. Also, as shown in Algorithm 3, our method requires that the covariance matrix  $C_s$  be computed for each  $s \in S$  at each optimization iteration. However, we observed empirically that computing  $C_s$  only once at the first iteration and reusing it afterward does not significantly reduce the quality of the results. In this way,  $C_s$  is estimated once at every scale of the pyramid. We thus adopted that strategy as a means to save processing time.

The number of iterations for Algorithms 1 and 3 is set to three whereas the number of mean-shift iterations is set to four. For each example, the radius of the mean-shift hypervolume  $S_h(x)$  is set to N in the spatial domain and to 10 in the range domain. Notice that all flow fields presented in this section have a density of 100%.

#### 5.2. Metrics

As mentioned, three kinds of sequences are used to compare the methods: synthetic, realistic and real sequences. While synthetic sequences are composed of pure computer-generated images, realistic sequences are made up of real-world images with simulated motion. Both synthetic and realistic sequences come with a ground truth vector field which makes it possible to quantitatively compare the methods. Following Barron et al. [3], we implemented the average angular error metric to evaluate the distance between the ground truth vector field  $\mathscr{V}$  and the estimated vector field  $\hat{v}$ , namely

$$\bar{\psi}_E = \frac{1}{\mathcal{N} \times \mathcal{M}} \sum_{s \in S} \arccos(\vec{\nu}_s \cdot \vec{\mathcal{V}}_s) \tag{49}$$

<sup>&</sup>lt;sup>2</sup>  $\mathscr{X}$  is the fusion window half-size in Eq. (14) and  $\Lambda$  is the window half-size used to compute the covariance matrix  $C_s$  in Eq. (15).



Fig. 14. Results for the CLAIRE sequence.

where  $\vec{v}_s$  and  $\vec{v'}_s$  are normalized 3D vectors:  $\vec{v}_s = \frac{(u,v,1)}{\sqrt{u^2+v^2+1}}$ . Because our method is meant to preserve sharp discontinuities, the metric was also implemented on vectors located at a distance lower or equal than 10 pixels from a motion edge, *i.e.*,

$$\bar{\psi}_{E}^{e} = \frac{1}{\mathcal{N} \times \mathcal{M}} \sum_{\substack{s \in S \\ \mathscr{D}_{s} \leq 10}} \arccos(\vec{\nu}_{s} \cdot \vec{\mathcal{V}}_{s})$$
(50)

where  $\mathcal{D}_s$  is the distance in pixels between site *s* and the nearest motion edge. This metrics is used to evaluate how accurate the opti-

cal flow algorithms are near flow discontinuities. The results are also presented in terms of the standard deviation

$$\sigma_{\psi_E} = \sqrt{\frac{1}{\mathcal{N} \times \mathcal{M}} \sum_{s \in S} (\arccos(\vec{\nu}_s \cdot \vec{\mathcal{V}}_s) - \bar{\psi}_E)^2}$$
(51)

and

$$\sigma_{\psi_{E}^{e}} = \sqrt{\frac{1}{\mathcal{N} \times \mathcal{M}} \sum_{\substack{s \in S \\ \mathscr{D}_{s} \leqslant 10}} (\arccos(\vec{\nu}_{s} \cdot \vec{\mathcal{V}}_{s}) - \bar{\psi}_{E}^{e})^{2}}.$$
(52)



Fig. 15. Results for the KARLSRUHE sequence.

#### 5.3. Synthetic sequences

We have compared the nine methods over three synthetic sequences, namely YOSEMITE WITH SKY, YOSEMITE WITHOUT SKY, and TRANSLATING SHAPES. Notice that in the case of the YOSEMITE WITH SKY sequence, although the first version of the ground truth file had a sky with a translational velocity of 1.0, we used the more recent one with a velocity of 2.0. The two YOSEMITE sequences were taken from Barron et al. [3] and Michael Black's web site (http://www.cs.brown.edu/people/black/) while the TRANSLATING SHAPES sequence was



Fig. 16. The effect of the noise level variable on the angular error of four sequences.

computer generated. The latter sequence exhibits two arbitrary shapes running horizontally in front of a uniform background. The three sequences are shown in Fig. 4 and the quantitative results are presented in Tables 1–3. The results for the TRANSLATING SHAPES sequence are also plotted in Fig. 7. In this figure, the first row presents the magnitude of the estimated vector field (in red) overlapped with a time frame. These images (as well as the zoom on the vector field) illustrate visually how precise each method is on that sequence.

Notice that for these three synthetic sequences, our method generates results that are at least as precise than the ones produced by the other methods. This is especially true near the motion edges of the TRANSLATING SHAPES sequence.

#### 5.4. Realistic sequences

The methods were also tested over the five realistic sequences presented in Fig. 5. The first two are the famous TRANSLATING TREE and DIVERGING TREE sequences [3] for which motion is global with no discontinuities. The other three sequences were computer generated with real images. For the CARS OVER PARK sequence, four cars (cropped from the KARLSRUHE sequence) are pasted on top of a picture of Central Park. The four cars move upward while the background is diagonally shifted toward the lower left corner. For the ROTATING BONSAI sequence, an image of a bonsai (a small tree in a pot) rotates in front of a flat motionless background. The last realistic sequence is the PARTHENON sequence for which an image of the Parthenon is plotted in front of an image of Florence's Duomo. In this sequence, the Parthenon moves horizontally toward the right whereas the background has a counterclockwise rotation. The reason why we picked these sequences was to illustrate how good our method is with sequences having different amounts of texture.

Quantitative results for the realistic sequences are presented in Tables 4–8. As we did for the synthetic TRANSLATING SHAPES sequence, the magnitude of the estimated vector fields have been overlapped with a time frame to illustrate how precise the methods are near motion boundaries. This is shown in Figs. 8–10. Except for the DIVERGING TREE sequence, our approach appears to be either the best or the second best method according to both quantitative measures. This is true whether the moving objects have a translational (CARS OVER PARK) or a rotational motion (ROTATING BONSAI). Even on highly textured scenes, *i.e.*, scenes for which most strong color/intensity edges do not correspond to a motion discontinuity, the results suggest that our method is as good as, if not better than the others.

#### 5.5. Real sequences

As shown in Fig. 6, five well known image sequences have been used to compare the methods. These sequences are CLAIRE, KARLSRUHE,

MOM AND DAUGHTER, TAXI, and FLOWER. Qualitative results are presented in Figs. 11–15.

#### 6. Discussion

The results presented in the previous section demonstrate that our method is competitive on all types of sequences. This includes sequences with global motion (YOSEMITE with and without sky, TRANSLATING and DIVERGING TREE) and local motion (KARLSRUHE, TAXI, TRANS-LATING SHAPES and BONSAI). Our method also shows good performance in estimating translating motion, rotating motion and diverging motion. Our method works well over sequences exhibiting large textureless backgrounds (BONSAI, TRANSLATING SHAPES, and CLAIRE) and on more highly textured sequences (PARTHENON, YOSEMITE, and FLOWER). These results illustrates the fact that the combination of a fusion procedure (Eq. (14)) and an edge avoidance procedure makes the method efficient on various sequences representing different challenges. While the mean-shift-based avoidance procedure preserves sharp motion boundaries, the covariance filter smoothes out the vector field and thus minimizes errors due to lack of texture, occlusion and noise. More specifically, the fact that our method relies on the assumption that image and motion boundaries coincide might suggest that our method is error-prone in regions where the number of motion boundaries is larger than the number of image edges (such as in yosemite, translating and diverging tree sequences). This would certainly be true if only the edge-avoidance procedure (first part of Algorithm 2) was used to estimate the flow. However, since the covariance filter is used at each iteration, errors that could eventually be induced by the mean-shift-based motion estimation are significantly reduced. Also, since the magnitude of the meanshift vector  $P_s$  (see Section 3.3) is limited to N, the avoidance procedure cannot induce a large error. Our method is thus efficient for estimating motion over highly textured scenes.

While testing our approach, we observed two limitations with our method. The first one concerns the processing time. Even by calculating  $C_s$  only once during the first iteration, our method is approximately 5 times slower than LK. Since this can be a major limitation for some applications, one might consider a technical solution to that problem. In fact, our method can be implemented on a parallel architecture (such as a programmable graphics card for example) and work in interactive time. Such implementation is possible because calculation over each site  $s \in S$  (at each stage of the algorithm) is independent of the processing of its neighbors. The parallel implementation would thus be effective for the meanshift calculation, motion estimation, and covariance filtering. Also, as we previously mentioned, another way to accelerate our method is by replacing the covariance filter with a simple Gaussian lowpass filter. This would greatly improve the processing times, although at the expense of precision. The second limitation of our method comes from the fact that, in some specific cases, despite the covariance filter, some mean-shift vectors  $P_s$  induce errors which could not be compensated. This is typically true when the intensity edges of two objects cross. Fig. 18 shows two examples for which a background edge is taken as a part of the moving object. As mentioned, since the magnitude of the mean-shift vectors is limited to N these errors are very local. One way to minimize this problem would be to include a motion segmentation step to the motion estimation process. This way, only the mean-shift vectors  $P_s$  located in the vicinity of a moving edge would be retained. With such modification, our approach would then become a motion segmentation/estimation method.

Finally, we have tested the influence of the most important variables of our method. These variables are the neighborhood window size N and the noise level  $N_l$ . As mentioned before, for every example presented in this paper, N was set to 9 or 11 and the noise level between 2 and 5. We observed that the resulting vector fields react



Fig. 17. Vector fields obtained with our method with different neighborhood window size *N* and different noise levels.

smoothly to a change of these variables and that other values could have been used. To illustrate this assertion, Fig. 17 presents different vector fields obtained on the CLAIRE and the TAXI sequences

with different values for *N* and for the noise level variable. The resulting vector fields react smoothly to a change of these two variables. In Fig. 17, we plotted the angular error  $(\psi_E)$  of four

P.-M. Jodoin, M. Mignotte/Computer Vision and Image Understanding 113 (2009) 511-531



Fig. 18. Two examples for which the mean-shift-based avoidance procedure has locally induced an error.

sequences processed with different noise levels. For those highly textured sequences (YOSEMITE and TRANSLATION TREE) a larger noise level tends to slightly raise the angular error; for less textured sequences, a higher noise level clearly reduces the angular error ( $\psi_E$ ). In other words, the use of such a bias makes our method estimate vector field with 100% density without inducing any significant error, even on highly textured sequences that might require little or no bias.

#### Acknowledgments

The authors are very grateful to Amar Mitiche for his constructive comments and to John Barron for his precious help.

#### References

- A. Mitiche, P. Bouthemy, Computation and analysis of image motion: a synopsis of current problems and methods, Int. J. Comput. Vis. 19 (1) (1996) 29–55.
- [2] H. Nagel, Image sequence evaluation 30 years and still going strong, in: Proc. Int. Conf. Pattern Recog., 2000, pp. 1149–1158.
- [3] J. Barron, D. Fleet, S. Beauchemin, Performance of optical flow techniques, Int. J. Comput. Vis. 12 (1) (1994) 43-77.
- [4] D. Fleet, A. Jepson, Computation of component image velocity from local phase information, Int. J. Comput. Vis. 5 (1) (1990) 77–104.
- [5] M. Langer, R. Mann, Optical snow, Int. J. Comput. Vis. 55 (1) (2003) 55-71.
- [6] A. Singh, An estimation-theoretic framework for image-flow computation, in: Proc. Int. Conf. Comput. Vis., 1990, pp. 168–177.
- [7] P. Anandan, A computation framework and an algorithm for the measurement of visual motion, Int. J. Comput. Vis. 2 (1989) 219–232.
- [8] A. Singh, Incremental estimation of image flow using a Kalman filter, J. Vis. Commun. Image Rep. 3 (1992) 39–57.
- [9] E. Memin, P. Perez, Hierarchical estimation and segmentation of dense motion fields, Int. J. Comput. Vis. 46 (2) (2002) 129-155.
- [10] 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.
- [11] J. Konrad, E. Dubois, Bayesian estimation of motion vector fields, IEEE Trans. Pattern Anal. Mach. Intell. 14 (9) (1992) 910–927.
- [12] B. Horn, B. Schunck, Determining optical flow, Artif. Intell. 17 (1981) 185-203.
- [13] B. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision (darpa), in: Proc. DARPA Image Understand. Workshop, 1981, pp. 121–130.
- [14] H. Nagel, On the estimation of optical flow relations between different approaches and some new results, Artif. Intell. 33 (3) (1987) 298–324.
- [15] M. Black, P. Anandan, The robust estimation of multiple motions: parametric and piecewise-smooth flow fields, Comput. Vis. Image Understand. 63 (1) (1996) 75–104.
- [16] E. Simoncelli, E. Adelson, D. Heeger, Probability distributions of optical flow, in: Proc. Conf. Vis. Pattern Recog., 1991, pp. 310–315.
- [17] J.-Y. Bouguet, Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the Algorithm, Technical Report, Intel Corporation, 1999.
- [18] B. William Thompson, Exploiting discontinuities in optical flow, Int. J. Comput. Vis. 30 (3) (1998) 163–173.
- [19] M. Black, P. Anandan, A model for the detection of motion over time, in: Proc. Int. Conf. Comput. Vis., 1990, pp. 33–37.

- [20] J.M. Odobez, P. Bouthemy, MRF-based motion segmentation exploiting a 2d motion model robust estimation, in: Proc. Int. Conf. Image Proc., 1995, pp. 628–631.
- [21] G. Aubert, R. Deriche, P. Kornprobst, Computing optical flow via variational techniques, SIAM J. Appl. Math. 60 (1) (1999).
- [22] J. Weickert, On discontinuity-preserving optic flow, in: Proc. Comput. Vis. Mob. Rob. Work., Santorini, 1998, pp. 115–122.
- [23] J. Weickert, C. Schnörr, A theoretical framework for convex regularizers in PDE-based computation of image motion, Int. J. Comput. Vis. 45 (3) (2001) 245–264.
- [24] T. Brox, A. Bruhn, N. Papenberg, J. Weickert, High accuracy optical flow estimation based on a theory for warping, in: Proc. Eur. Conf. Comput. Vis., 2004, pp. 25–36.
- [25] A. Bab-Hadiashar, D. Suter, Robust optic flow computation, Int. J. Comput. Vis. 29 (1) (1998) 59-77.
- [26] E. Ong, M. Spann, Robust optical flow computation based on least-median-ofsquares regression, Int. J. Comput. Vis. 30 (1) (1999) 51-82.
- [27] D. Comaniciu, Nonparametric information fusion for motion estimation, in: Proc. Conf. Vis. Pattern Recog., 2003, pp. 59–68.
- [28] G. Farneback, Very high accuracy velocity estimation using orientation tensors, parametric motion and simultaneous segmentation of motion field, in: Proc. Int. Conf. Comput. Vis., 2001, pp. 171–177.
- [29] S. Geman, D. Geman, Stochastic relaxation, Gibbs distributions, and the bayesian restoration of images, IEEE Trans. Pattern Anal. Mach. Intell. 6 (6) (1984) 721–741.
- [30] M. Black, Combining intensity and motion for incremental segmentation and tracking over long image sequences, in: Proc. Sec. Eur. Conf. Comput. Vis., 1992, pp. 485–493.
- [31] C. Fuh, P. Maragos, Region-based optical flow estimation, in: Proc. Conf. Vis. Pattern Recog., 1989, pp. 130–135.
- [32] F. Meyer, P. Bouthemy, Region-based tracking using affine motion models in long image sequences, CVGIP Image Understand. 60 (2) (1994) 119–140.
- [33] V. Dang, A.R. Mansouri, J. Konrad, Motion estimation for region-based video coding, in: Proc. Int. Conf. Image Proc., 1995, pp. 2189–2192.
- [34] M. Black, A. Jepson, Estimating optical flow in segmented images using variable-order parametric models with local deformations, IEEE Trans. Pattern Anal. Mach. Intell. 18 (10) (1996) 972–986.
- [35] L. Alvarez, J. Esclarin, M. Lefebure, J.A. Sánchez, PDE model for computing the optical flow, in: Proc. XVI Cong. de Ecuac. Difer. y Aplic, 1999, pp. 1349–1356.
- [36] H. Nagel, W. Enkelmann, Investigation of second order gray value variations to estimate corner point displacements, in: Proc. Int. Conf. Pattern Recog., 1982, pp. 768–773.
- [37] C. Schnörr, On functionals with greyvalue-controlled smoothness terms for determining optical flow, IEEE Trans. Pattern Anal. Mach. Intell. 15 (10) (1993) 1074–1079.
- [38] L. Alvarez, J. Weickert, J. Sánchez, Reliable estimation of dense optical flow fields with large displacements, Int. J. Comput. Vis. 39 (1) (2000) 41–56.
- [39] D. Comaniciu, P. Meer, Mean shift analysis and applications, in: Proc. Int. Conf. Comput. Vis., 1999, pp. 1197–1203.
- [40] E. Trucco, A. Verri, Introductory Techniques for 3-D Computer Vision, Prentice Hall, 1998.
- [41] H. Nagel, Recent advances in image sequence analysis, in: Proc. Prem. Coll. Image Trait. Synt. Tech. et App., 1984, pp. 545–558.
- [42] M. Kraus, M. Strengert, Pyramid filters based on bilinear interpolation, in: proc. of Int. Conf. Comput. Graph. Theo. App., Barcelona, Spain, 2007, pp. 21–28.
- [43] A. Bovik, Handbook of Image and Video Processing, second ed., Academic Press, 2005.
- [44] R. Gonzalez, R. Woods, Digital Image Processing, Addison-Wesley Longman Publishing Co. Inc., Boston, MA, USA, 2001.
- [45] C. Stiller, J. Konrad, Estimating motion in image sequences: a tutorial on modeling and computation of 2d motion, IEEE Signal Proc. Mag. 16 (4) (1999) 70–91.

- [46] J. Choi, S. Lee, S. Kim, Segmentation and motion estimation of moving objects for object-oriented analysis-synthesis coding, in: Proc. Int. Conf. Aco. Speech Signal Proc., 1995, pp. 2431–2434.
- [47] J. Wang, E. Adelson, Representing moving images with layers, IEEE Trans. Image Proc. Spec. Issue Image Seq. Comp. 3 (5) (1994) 625–638.
- [48] X.-S. Zhou, D. Comaniciu, A. Gupta, An information fusion framework for robust shape tracking, IEEE Trans. Pattern Anal. Mach. Intell. 27 (1) (2005) 115–129.
- [49] K. Fukunaga, L. Hostetler, The estimation of the gradient of a density function, IEEE Trans. Info. Theory 21 (1975) 32–40.
- [50] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Trans. Pattern Anal. Mach. Intell. 24 (5) (2002) 603–619.
- [51] J. Barron, D. Fleet, S. Beauchemin, Performance of Optical Flow Techniques, Technical Report 299, University of Western Ontario, Department of CS, 1993.
- [52] J. Weickert, C. Schnörr, Variational optic flow computation with a spatiotemporal smoothness constraint, J. Math. Imaging Vis. 14 (3) (2001) 245– 255.
- [53] R. Deriche, P. Kornprobst, G. Aubert, Optical-flow estimation while preserving its discontinuities: a variational approach, in: Proc. Asian Conf. Comput. Vis. Invited Sess. Papers, 1996, pp. 71–80.
- [54] J. Weickert, Coherence-enhancing diffusion of colour images, Image Vis. Comput. 17 (3-4) (1999) 201–212.
- [55] R. Kimmel, R. Malladi, N. Sochen, Images as embedded maps and minimal surfaces: movies, color, texture, and volumetric medical images, Int. J. Comput. Vis. 39 (2) (2000) 111–129.
- [56] A. Bruhn, J. Weickert, Schnörr. Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods, Int. J. Comput. Vis. 61 (3) (2005) 211–231.