High Accuracy Optical Flow Method Based on a Theory for Warping: Implementation and Qualitative/Quantitative Evaluation

Mohammad Faisal and John Barron

Dept. of Computer Science
The University of Western Ontario
London, Ontario, Canada, N6A 5B7
{mfaisal,barron}@csd.uwo.ca

Abstract. We describe the implementation of a 2D optical flow algorithm published in the European Conference on Computer Vision (ECCV 2004) by Brox et al. [1] (best paper award) and a qualitative and quantitative evaluation of it for a number of synthetic and real image sequences. Their optical flow method combines three assumptions: a brightness constancy assumption, a gradient constancy assumption and a spatio-temporal smoothness constraint. A numerical scheme based on fixed point iterations is used. Their method uses a coarse-to-fine warping strategy to measure larger optical flow vectors. We have investigated the algorithm in detail and our evaluation of the method demonstrates that it produces very accurate optical flow fields from only 2 input images.

Keywords: optical flow, regularization, warping, multiscale pyramid, brightness/gradient,smoothing constraints.

1 Introduction

Optical flow estimation is still an open research areas in computer vision. While many methods have been proposed, Brox, Bruhn, Papenberg and Weickert [1] presented a variational approach at ECCV 2004 that they claimed gave the best quantitative flow (up to that time). Later, Papenberg, Bruhn, Brox, Didas and Weickert [2] added a few additional constraints to this algorithm and got even better results. We implemented Brox et al.'s algorithm (we started before Papenberg et al.'s algorithm was published) and investigated why it produced such good flow fields using only 2 frames and with such poor temporal intensity differentiation (simple pixel differences). We quantitatively evaluated our implementation for a number of image sequences including the Yosemite Fly-Through sequence made in 1988 by Lynn Quan at SRI International [3]. It consists of 15 textured depth maps of a mountain range and a valley in Yosemite National Park (with the camera moving towards a point in the valley to create a diverging flow field) and fractals generated clouds that moved left to right at 2 pixels per frame. Given a depth map and the instantaneous camera motion, the correct image velocity field can be computed [4]. This sequence is perhaps one of the

M. Kamel and A. Campilho (Eds.): ICIAR 2007, LNCS 4633, pp. 513–525, 2007.

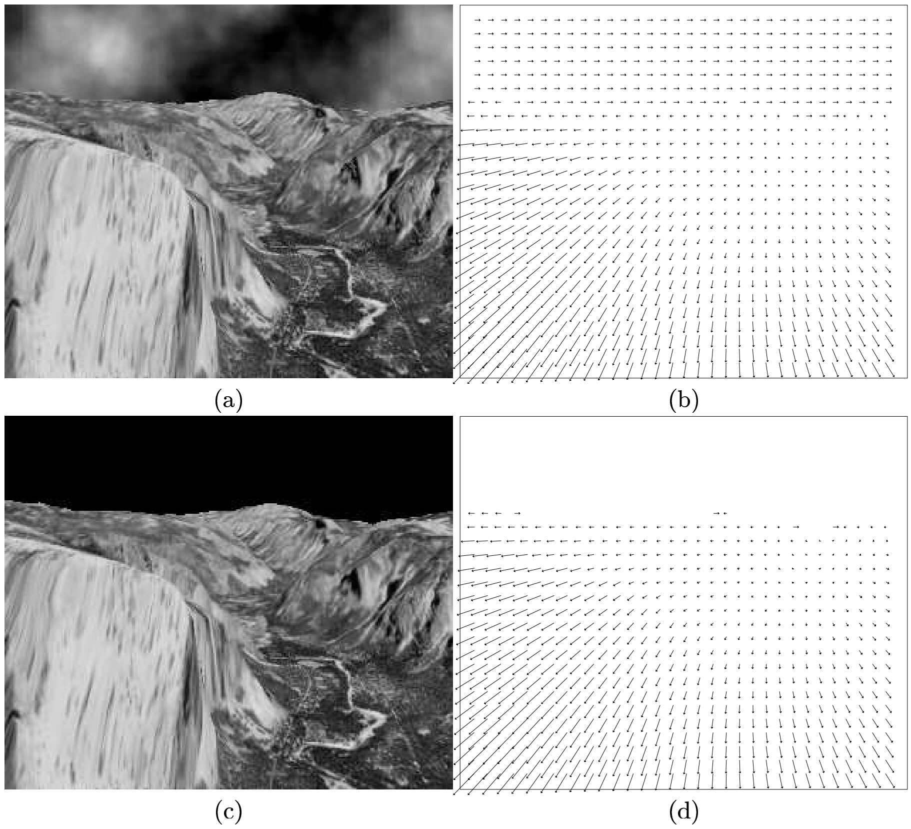


Fig. 1. (a) The middle frame of the cloudy **Yosemite** sequence and (b) the correct flow field for (a). (c) The middle frame for cloudless **Yosemite** sequence and (d) the correct flowfield for (c).

most complex synthetic image sequences freely available as the mountain range has varying depth and the clouds are deformable objects. Michael Black suggests that because there is no real ground truth for the cloud motion (they are modelled as fractals undergoing Brownian motion)[1] reporting errors for the cloud region is a bit meaningless. Thus, the sequence is often evaluated without the clouds; in this paper we evaluate the flow for both sequences. Figure 1 shows the central cloudy and cloudless frames of this sequence plus their correct flows.

2 Relevant Literature

Here, we briefly review the papers, whose cloudy/cloudless Yosemite results were quoted by Brox et al. [1]. These techniques comprise the best contemporary

[1] http://www.cs.brown.edu/ black/

optical flow methods and all give a quantitative analysis of the cloudy and/or cloudless Yosemite sequences.

The classical Horn and Schunck optical flow algorithm [5] is a global regularization of a data term consisting of the motion constraint equation and a smoothness term constraining the velocity to vary smoothly everywhere. Lucas and Kanade [6] assume the velocity is constant in local neighbourhoods and formulate a least squares calculation of the velocity for each neighbourhood. Ju, Black and Jepson [7] propose a "Skin and Bones" model to compute optical flow using a parametrized (affine) flow model with a smoothness constraint on the flow parameters to ensure continuity of motion between patches. Lai and Vemuri [8] propose a gradient-based regularization method that includes a contour based motion constraint equation that is enforced only at zero-crossings. Bab-Hadiashar and Suter [9] formulate the measurement of optical flow as a over-determined set of linear equations. They use 2 robust estimation techniques: Least Median of Squares and Least Median of Squares Orthogonal Distances. Alvarez, Wickert and Sánchez [10] introduce improvements to Nagel and Enkelmann's [11] optical flow work that include hierarchical processing and an energy term that is invariant under linear brightness changes. Farnebäck's [12] algorithm has 3 distinct components: estimation of spatio-temporal tensors, estimation of parametric motion models and simultaneous segmentation of the motion field. Mémin and Pérez [13] propose a robust energy-based model for the incremental estimation of optical flow in a hierarchical piece-wise parametric minimization of an energy functional in regular or adaptive meshes at each hierarchical level from the coarsest to the finest levels. Bruhn, Weickert and Schnörr [14,15] propose a method that combined local and global methods, in particular, those of Horn and Schunck and Lucas and Kanade. The data term in the Horn and Schunck regularization is now replaced by the least squares Lucas and Kanade constraint.

3 3D Angular Error

All authors used the angular error measure of error proposed by David Fleet (see Barron et al. [16]). Velocity may be viewed as a space-time direction vector $(u, v, 1)$ in units of (pixel, pixel, frame). We can measure errors as angular deviations from the correct space-time orientation. We write velocities as 3D normalized vectors, $\hat{v} \equiv \frac{1}{\sqrt{u^2+v^2+1}}(u, v, 1)^T$. The angular error between the correct velocity $\boldsymbol{v}_c$ and an estimate $\boldsymbol{v}_e$ is then: $\psi_E = \arccos(\hat{v}_c \cdot \hat{v}_e)$. This metric takes into account both direction and magnitude error as a single number.

4 Brox et al.'s Variational Model

The Brox et al. [1] method is composed of several constraints. The **Grayvalue Constancy Assumption** requires that the grayvalue of a pixel does not change as it undergoes motion: $I(x, y, t) = I(x + u, y + v, t + 1)$. A 1^{st} order Taylor series expansion leads to the standard motion constraint equation $I_x u + I_y v + I_t = 0$.

The **Gradient Constancy Assumption** requires that the gradient of the image grayvalues not to vary due to displacement: $\nabla I(x,y,t) = \nabla I(x+u, y+v, t+1)$. Brox et al.'s **Smoothness Assumption** is the standard Horn and Schunck smoothness constraint. Lastly, Brox et al. use a **Multiscale** hierarchical (pyramid) approach, which we describe below.

Brox et al's energy function penalizes deviation from their model assumptions. Their data term is:

$$E_{Data}(u,v) = \int_{\Omega} \Psi(|I(\mathbf{x}{+}\mathbf{w}) - I(\mathbf{x})|^2 + \gamma|\nabla I(\mathbf{x}{+}\mathbf{w}) - \nabla I(\mathbf{x})|^2)\mathbf{dx}. \quad (1)$$

where Ω is the region of interest (the image) over which the minimization is done. $\Psi(s^2) = \sqrt{s^2 + \epsilon^2}$ The small positive constant ϵ keeps $\Psi(s^2)$ convex which helps the overall minimization process as there is now a unique global minimum for most images (consider two constant grayvalue images as one possible exception). Brox et al. use $\epsilon = 0.0001$. A Horn and Schunck smoothness term is also used:

$$E_{smooth}(u,v) = \int_{\Omega} \Psi(|\nabla_3 u|^2 + |\nabla_3 v|^2)\mathbf{dx}, \quad (2)$$

with the same function Ψ. The spatio-temporal gradient $\nabla_3 = (\partial_x, \partial_y, \partial_t)^\top$ indicates that a spatio-temporal smoothness assumption is involved. Here, we have only used two images so the spatio-temporal gradient (∇_3) becomes the spatial gradient (∇_2). The total energy is the weighted sum between the data term and the smoothness term

$$E(u,v) = E_{Data} + \alpha E_{smooth}, \quad (3)$$

for some regularization parameter $\alpha > 0$. Note that $E(u,v)$ is highly nonlinear. The goal is to find u and v values that minimizes this energy over the whole image.

We use the same mathematical abbreviations that Brox et al. used in the paper (via their Equation (8) in the algorithm description [1]): $I_x = \partial_x I(\mathbf{x}+\mathbf{w})$, $I_y = \partial_y I(\mathbf{x}+\mathbf{w})$, $I_z = I(\mathbf{x}+\mathbf{w}) - I(\mathbf{x})$, $I_{xx} = \partial_{xx} I(\mathbf{x}+\mathbf{w})$, $I_{xy} = \partial_{xy} I(\mathbf{x}+\mathbf{w})$, $I_{yy} = \partial_{yy} I(\mathbf{x}+\mathbf{w})$, $I_{xz} = \partial_x I(\mathbf{x}+\mathbf{w}) - \partial_x I(\mathbf{x})$ and $I_{yz} = \partial_y I(\mathbf{x}+\mathbf{w}) - \partial_y I(\mathbf{x})$. The functional to be minimized is:

$$\begin{aligned} f &= \Psi\left(|I(\mathbf{x}+\mathbf{w}) - I(\mathbf{x})|^2 + \gamma(|\nabla I(\mathbf{x}+\mathbf{w}) - \nabla I(\mathbf{x})|^2\right) + \alpha\Psi\left(|\nabla_3 u|^2 + |\nabla_3 v|^2\right) \\ &= \Psi\left(I_z^2 + \gamma(I_{xz}^2 + I_{yz}^2)\right) + \alpha\Psi\left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right). \quad (4) \end{aligned}$$

We compute $u^{k+1} = u^k + du^k$ and $v^{k+1} = v^k + dv^k$ from (u^k, v^k) and unknown correction terms (du^k, dv^k). Brox et al. use an arbitrary reduction factor $\eta \in (0,1)$ when down sampling in the pyramid, typically $\eta \in [0.80, 0.95]$, which allows smooth flow projections between adjacent images in the pyramid. An outer fixed iteration is used to solve a linear system of equations in terms of $du^{k,l+1}$ and $dv^{k,l+1}$ at level $k+1$, holding $(\Psi\,')^{k,l}_{Data}$ and $(\Psi\,')^{k}_{Smooth}$ constant. A second fixed inner iteration was used to handle the nonlinearity due to $(\Psi\,')^{k,l}_{Data}$ and

$(\Psi\ ')^k_{Smooth}$ by updating these values each time the outer iteration converged. Once both inner and outer iterations have converged we have the flow at level k, giving us $\mathbf{w^k} = (\mathbf{u^k}, \mathbf{v^k})$. This is then to be used as the initial solution for $\mathbf{w^{k+1}}$ on the next finer level. The iterative equations that minimize the Euler-Lagrange equations are:

$$\begin{aligned}
&(\Psi\ ')^{k,l}_{Data}\left(I^k_x(I^k_z + I^k_x du^{k,l+1} + I^k_y dv^{k,l+1})\right.\\
&\left.+\gamma\left[I^k_{xx}(I^k_{xz} + I^k_{xx} du^{k,l+1} + I^k_{xy} dv^{k,l+1}) + I^k_{xy}(I^k_{yz} + I^k_{xy} du^{k,l+1} + I^k_{yy} dv^{k,l+1})\right]\right)\\
&-\alpha\ \mathbf{Div}\left((\Psi\ ')^k_{Smooth}\nabla_3(u^k + du^{k,l+1})\right) = 0 \qquad (5)
\end{aligned}$$

and

$$\begin{aligned}
&(\Psi\ ')^{k,l}_{Data}\left(I^k_y(I^k_z + I^k_x du^{k,l+1} + I^k_y dv^{k,l+1})\right.\\
&\left.+\gamma\left[I^k_{yy}(I^k_{yz} + I^k_{xy} du^{k,l+1} + I^k_{yy} dv^{k,l+1}) + I^k_{xy}(I^k_{xz} + I^k_{xx} du^{k,l+1} + I^k_{xy} dv^{k,l+1})\right]\right)\\
&-\alpha\ \mathbf{Div}\left((\Psi\ ')^{k,l}_{Smooth}\nabla_3(v^k + dv^{k,l+1})\right) = 0. \qquad (6)
\end{aligned}$$

To compute I^{k+1}_* terms from I^k_* terms we use:

$$I^{k+1}_z \approx I^k_z + I^k_x du^k + I^k_y dv^k, \qquad (7)$$

$$I^{k+1}_{xz} \approx I^k_{xz} + I^k_{xx} du^k + I^k_{xy} dv^k \quad \text{and} \qquad (8)$$

$$I^{k+1}_{yz} \approx I^k_{yz} + I^k_{xy} du^k + I^k_{yy} dv^k. \qquad (9)$$

We have not been able to solve the linear system of equations using common numerical methods such as Gauss-Seidel or Successive Over Relaxation (SOR) as Brox et al. did (a typical experience was the first 20-30 iterations with convergence followed by rapid divergence). Instead we make 2 modifications to the equations to obtain 2 variants on Brox et al.'s algorithm, which we present in the next 2 sections. We adopt a different approach to solve the set of equations using standard averaging as Horn and Schunck did, as that leads to convergence for the equations (7×7 averaging was found to be the best). We adopted Brox's technique from his Ph.D. thesis [17] to solve the smoothing term but then used iterations of Cramer's rule rather than directly using Gauss-Seidel or SOR (this was our $4 - pt$ algorithm) to get convergence.

4.1 7 × 7 Averaging

We can express Equations (5) and (6) as a linear system of equations:

$$A(du^{k,l+1}) + B(dv^{k,l+1}) = E(du^{k,l+1}) \qquad (10)$$

$$C(du^{k,l+1}) + D(dv^{k,l+1}) = F(dv^{k,l+1}), \qquad (11)$$

where:

$$A = (\Psi')^{k,l+1}_{Data}\left(I^k_x I^k_x + \gamma(I^k_{xx} I^k_{xx} + I^k_{xy} I^k_{xy})\right), \qquad (12)$$

$$B = (\Psi')^{k,l+1}_{Data}\left(I^k_x I^k_y + \gamma(I^k_{xx} I^k_{xy} + I^k_{xy} I^k_{yy})\right), \qquad (13)$$

$$\begin{aligned}
C &= (\Psi')^{k,l+1}_{Data}\left(I^k_y I^k_x + \gamma(I^k_{yy} I^k_{xy} + I^k_{xy} I^k_{xx})\right), && (14)\\
D &= (\Psi')^{k,l+1}_{Data}\left(I^k_y I^k_y + \gamma(I^k_{yy} I^k_{yy} + I^k_{xy} I^k_{xy})\right), && (15)\\
E &= \alpha\ (\Psi\ ')^{k,l}_{Smooth}\mathbf{Div}\left(\nabla_3(u^k + du^{k,l+1})\right) \quad \text{and} \\
&\quad - (\Psi\ ')^{k,l}_{Data}\left(I^k_x I^k_z + \gamma(I^k_{xx} I^k_{xz} + I^k_{xy} I^k_{yz})\right) && (16)\\
F &= \alpha\ (\Psi\ ')^{k,l}_{Smooth}\mathbf{Div}\left(\nabla_3(v^k + dv^{k,l+1})\right) \\
&\quad - (\Psi\ ')^{k,l}_{Data}\left(I^k_y I^k_z + \gamma(I^k_{yy} I^k_{yz} + I^k_{xy} I^k_{xz})\right). && (17)
\end{aligned}$$

After some algebra and using the standard approximation $X_{xx} + X_{yy} \approx \overline{X} - X$ as in [5] we can write:

$$E = \alpha\ (\Psi\ ')^{k,l}_{Smooth}\left((\overline{u^k} - u^k) + (\overline{du^{k,l+1}} - du^{k,l+1})\right) - e \qquad (18)$$

where

$$e = (\Psi\ ')^{k,l}_{Data}\left(I^k_x I^k_z + \gamma(I^k_{xx} I^k_{xz} + I^k_{xy} I^k_{yz})\right) \qquad (19)$$

and

$$F = \alpha\ (\Psi\ ')^{k,l}_{Smooth}\left((\overline{v^k} - v^k) + (\overline{dv^{k,l+1}} - dv^{k,l+1})\right) - f \qquad (20)$$

where

$$f = (\Psi\ ')^{k,l}_{Data}\left(I^k_y I^k_z + \gamma(I^k_{yy} I^k_{yz} + I^k_{xy} I^k_{xz})\right). \qquad (21)$$

We can solve for $du^{k,l+1}$ and $dv^{k,l+1}$ using Crammer's rule as:

$$\begin{bmatrix} du^{k,l+1} \\ dv^{k,l+1} \end{bmatrix} = \frac{1}{det}\begin{bmatrix}
D\alpha(\Psi\ ')^{k,l}_{Smooth}(\overline{u^k} + \overline{du^{k,l}} - u^k) - De \\
\alpha^2((\Psi\ ')^{k,l}_{Smooth})^2(\overline{u^k} + \overline{du^{k,l}} - u^k) - \alpha(\Psi\ ')^{k,l}_{Smooth}e \\
-B\alpha(\Psi\ ')^{k,l}_{Smooth}(\overline{v^k} + \overline{dv^{k,l}} + v^k) + Bf \\
\\
-C\alpha(\Psi\ ')^{k,l}_{Smooth}(\overline{u^k} + \overline{du^{k,l}} + u^k) + Ce \\
A\alpha(\Psi\ ')^{k,l}_{Smooth}(\overline{v^k} + \overline{dv^{k,l}} - v^k) - Af \\
\alpha^2((\Psi\ ')^{k,l}_{Smooth})^2(\overline{v^k} + \overline{dv^{k,l}} + v^k) - \alpha(\Psi\ ')^{k,l}_{Smooth}f
\end{bmatrix}. \qquad (22)$$

where $det = AD + A\alpha(\Psi')^{k,l}_{Smooth} + D\alpha(\Psi')^{k,l}_{Smooth} + \alpha^2((\Psi')^{k,l}_{Smooth})^2 - BC$. We initialize $du^{k,1}$ and $dv^{k,1}$ to 0. Then we solve for the next set of $(du^{k,l+1}, dv^{k,l+1})$, etc., by using these two equations and computing $\overline{du^{k,l}}$, $\overline{dv^{k,l}}$, $\overline{u^k}$, $\overline{v^k}$ by taking the average of a 7 × 7 window around each pixel.

4.2 4-Point Differences

Brox [17] suggests using finite differences to calculate ∇u :

$$|\nabla u_{i,j}| = \sqrt{\left(\frac{u_{i+1,j} - u_{i-1,j}}{2}\right)^2 + \left(\frac{u_{i,j+1} - u_{i,j-1}}{2}\right)^2} \qquad (23)$$

Once we have the gradient magnitude ($|\nabla u|$), we can calculate the *div* term for u as:

$$\begin{aligned}
&\partial_x\left(\Psi'(\nabla u^2)u_x\right) + \partial_y\left(\Psi'(\nabla u^2)u_y\right) \\
&= \tfrac{\Psi'_{i+1,j}+\Psi'_{i,j}}{2}(u_{i+1,j} - u_{i,j}) - \tfrac{\Psi'_{i-1,j}+\Psi'_{i,j}}{2}(u_{i,j} - u_{i-1,j}) \qquad (24)\\
&+ \tfrac{\Psi'_{i,j+1}+\Psi'_{i,j}}{2}(u_{i,j+1} - u_{i,j}) - \tfrac{\Psi'_{i,j-1}+\Psi'_{i,j}}{2}(u_{i,j} - u_{i,j-1}),
\end{aligned}$$

where Ψ' refers to $(\Psi')^{k,l}_{Smooth} = (|\nabla(u^k + du^{k,l})|^2 + |\nabla(v^k + dv^{k,l})|^2)$. We replace the Ψ''s by some weight coefficients w_i as follows: $w_1 = \frac{\Psi'_{i+1,j}+\Psi'_{i,j}}{2}$, $w_2 = \frac{\Psi'_{i-1,j}+\Psi'_{i,j}}{2}$, $w_3 = \frac{\Psi'_{i,j+1}+\Psi'_{i,j}}{2}$, $w_4 = \frac{\Psi'_{i,j-1}+\Psi'_{i,j}}{2}$ and $w = w_1 + w_2 + w_3 + w_4$. We can write a similar equation for $\partial_x \left(\Psi'(\nabla v^2) v_x\right) + \partial_y \left(\Psi'(\nabla v^2) v_y\right)$. The *div* term in Equations (5) and (6) now become:

$$\begin{aligned} &w_1(u_{i+1,j}) + w_2(u_{i-1,j}) + w_3(u_{i,j+1}) + w_4(u_{i,j-1}) - w(u_{i,j}) \\ &+w_1(du_{i+1,j}) + w_2(du_{i-1,j}) + w_3(du_{i,j+1}) + w_4(du_{i,j-1}) - w(du_{i,j}) \end{aligned} \tag{25}$$

and

$$\begin{aligned} &w_1(v_{i+1,j}) + w_2(v_{i-1,j}) + w_3(v_{i,j+1}) + w_4(v_{i,j-1}) - w(v_{i,j}) \\ &+w_1(dv_{i+1,j}) + w_2(dv_{i-1,j}) + w_3(dv_{i,j+1}) + w_4(dv_{i,j-1}) - w(dv_{i,j}). \end{aligned} \tag{26}$$

The iterative solution is almost the same as for the averaging method but now we replace the Ψ'_{Smooth} with appropriate weight w_*. Again, using Crammer's rule we obtain:

$$\begin{bmatrix} du^{k,l+1} \\ dv^{k,l+1} \end{bmatrix} = \frac{1}{det} \begin{bmatrix} D\alpha(\overline{u^k} + \overline{du^{k,l}} - wu^k) - De + \alpha^2 w(\overline{u^k} + \overline{du^{k,l}} - wu^k) - \alpha we \\ -B\alpha(\overline{v^k} + \overline{dv^{k,l}} + wv^k) + Bf \\ A\alpha(\overline{v^k} + \overline{dv^{k,l}} - wv^k) - Af + \alpha^2 w(\overline{v^k} + \overline{dv^{k,l}} - wv^k) - \alpha wf \\ -C\alpha(\overline{u^k} + \overline{du^{k,l}} + wu^k) + Ce \end{bmatrix}. \tag{27}$$

5 Experimental Results

Brox et al. used a 5 tap filter to compute the 1^{st} order spatial derivatives. They used the coefficients: {0.8666,-0.6666,0.0,0.6666,-0.8666}. They used a 3-tap filter with coefficients: {-0.5,0.0,0.5} on the 1^{st} order spatial derivatives to compute the 2^{nd} order spatial derivatives. The input images were initially presmoothed with a 2D Gaussian with $\sigma = 1.3$.

Since the pyramid levels have different widths and heights, when the velocities are projected down from a coarse level to a finer level we distributed the velocities in the finer level by multiplying the integer pixel coordinates by $1/\eta$. This leaves some holes (along lines) in the flow field, these were filled by averaging.

We coded our algorithm in Tinatool (version 5.0) [18,19], an open source X windows based software for Computer Vision algorithms.

5.1 Synthetic Results

Figure 2 shows the cloudy and cloudless Yosemite flows computed by our implementation. We used the same parameter setting as Brox et al. [1], namely $\sigma = 1.3$, $\alpha = 80$ and $\gamma = 100$, 10 inner and outer iterations, a reduction factor of $\eta = 0.95$ and 77 levels in the pyramid. We can see from the error intensity images that most of the error occurs at the discontinuity between the mountain and the clouds and in the clouds themselves (for the cloudy data). The later is to be expected as optical flow requires a rigidity assumption and clouds are non-rigid.

Table 1. Comparison of results from the literature for the cloudy Yosemite data

Technique	Angular Error	St. Dev.
Horn-Schunck, modified [16]	9.78°	16.19°
Alvarez et al. [10]	5.53°	7.40°
Weickert et al. [14]	5.18°	8.68°
Brox et al. 4-pt	**5.07°**	**9.90°**
Mémin-Pérez [13]	4.69°	6.89°
Brox et al. 7×7	**3.60°**	**10.2°**
Brox et al. [1]	2.46°	7.31°

Table 2. Comparison of results from the literature for the cloudless Yosemite data

Technique	Angular Error	St. Dev.
Brox et al. 4-pt	**2.63°**	**11.68°**
Ju et al. [7]	2.16°	2.00°
Bab-Hadiashar-Suter [9]	2.05°	2.92°
Lai-Vemuri et al. [8]	1.99°	1.41°
Brox et al. [1]	1.59°	1.39°
Mémin and Pérez [13]	1.58°	1.21°
Weickert et al. [14]	1.46°	1.50°
Farnebäck [12]	1.14°	2.14°
Brox et al. 7×7	**0.54°**	**5.37°**

Table 1 shows that our best results for the cloudy Yosemite data are not quite as good as Brox et al.'s but better than the other results. We tested the variation of the basic parameters for the algorithm (see Table 3 for some results). Table 2 shows that, while out **4-pt** results for the cloudless Yosemite data are the worst, our $\mathbf{7} \times \mathbf{7}$ results are the best, even better than Brox et al.'s result. One of the main strengths of Brox el al.'s method is the use of slowly changing pyramidal images to compute good temporal derivatives. We always obtained better results when higher reduction factors (η). Figure 3 shows the evolution of $\mathbf{I}(\mathbf{x} + \omega)$ into $\mathbf{I}(\mathbf{x})$. By gradually changing the image size between adjacent levels in the pyramid it is possible to maintain good flow until the final image is reached.

We tested Brox et al.'s differentiation against Simoncelli derivatives [20] (which are known to be good) by keeping all other things but the differentiation the same. At the initial level in the pyramid both differentiation methods produced similar results (12.57° for Simoncelli and 11.25° for Brox et al.). Brox et al. derivatives were slightly better here than Simoncell derivatives because it is difficult to warp the 7 images required by Simoncelli filtering; we have to warp by 3 images to the left and to the right of the middle image. For lower level images in the pyramid the error from bilinear interpolation and violation of the implicit constant motion assumption became very significant. For example, by 10 levels down the pyramid, Brox et al.'s accuracy was a bit more than 3 times better than Simoncelli's.

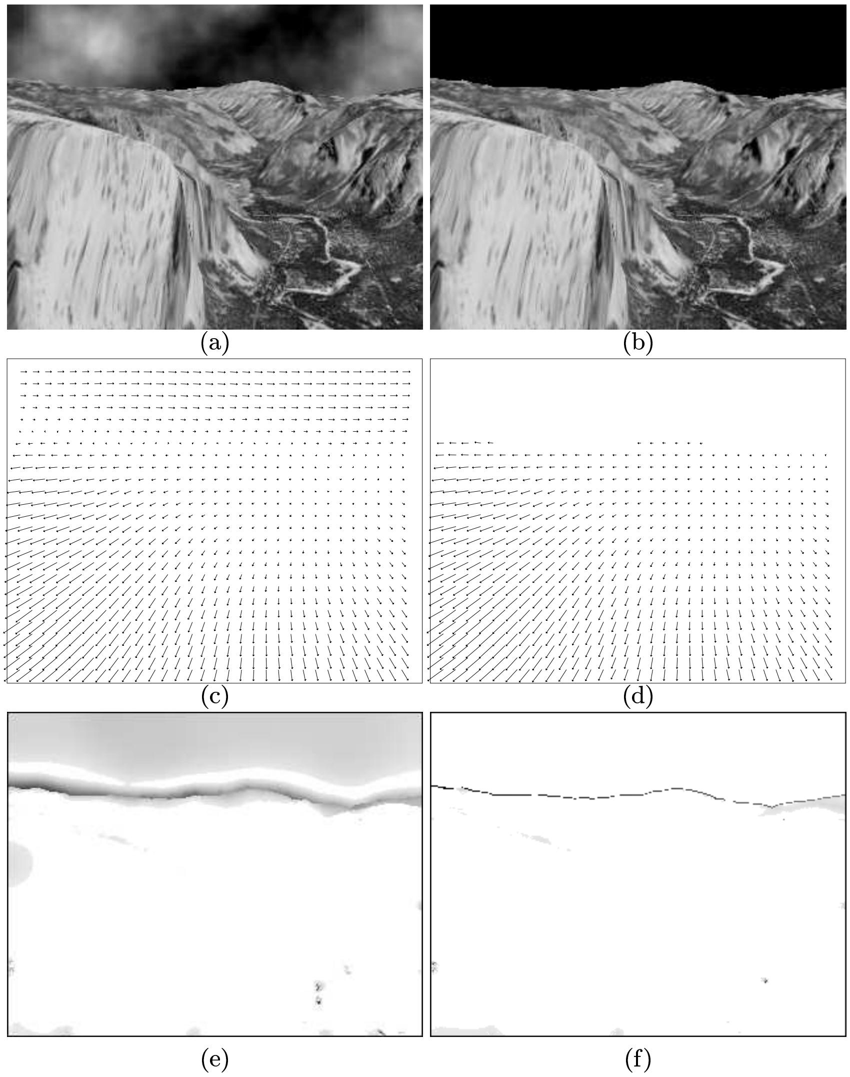


Fig. 2. (a) Frame 8 of the cloudy Yosemite sequence. (b) Frame 8 of the cloudless Yosemite sequence. (c)-(d) Best flow fields for (a) and (b). (e)-(f) Error images for (c) and (d).

As can be seem in Table 3, increased smoothing (α) tends to produce more accurate results for the Yosemite sequence. If α is set to 0 we obtain very poor results; the smoothness term is required as one would expect. Higher values of γ weighted the gradient constraint more importantly and usually also reduced error. Turning off γ resulted in less accurate flow (now effectively becoming hierarchical Horn and Schunck). Similarly, more presmoothing (σ) produced less error at first and then more error as σ increased [16].

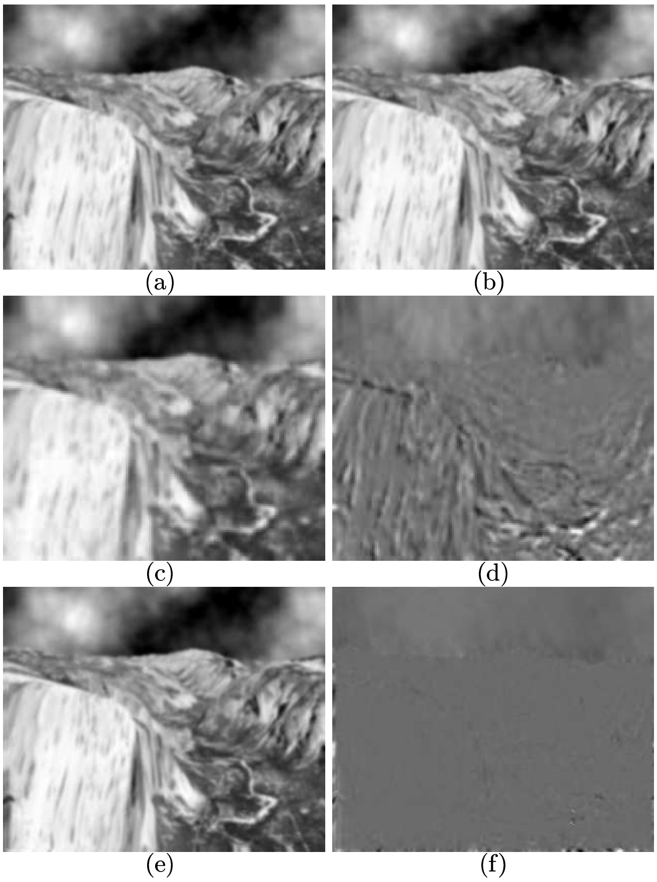


Fig. 3. (a)-(b) Frames 8 and 9 of the cloudy Yosemite sequence at level 0 of the pyramid. (c) $I(x + \omega)$ and (d) I_z after computation of the flow at the top level and projection to level 1. (e)-(f) The transformation of $I(x+w)$ and I_z at level 5. I_z is now approaching 0 while $I(x + w)$ is essentially $I(x)$ and I_z is close to 0.

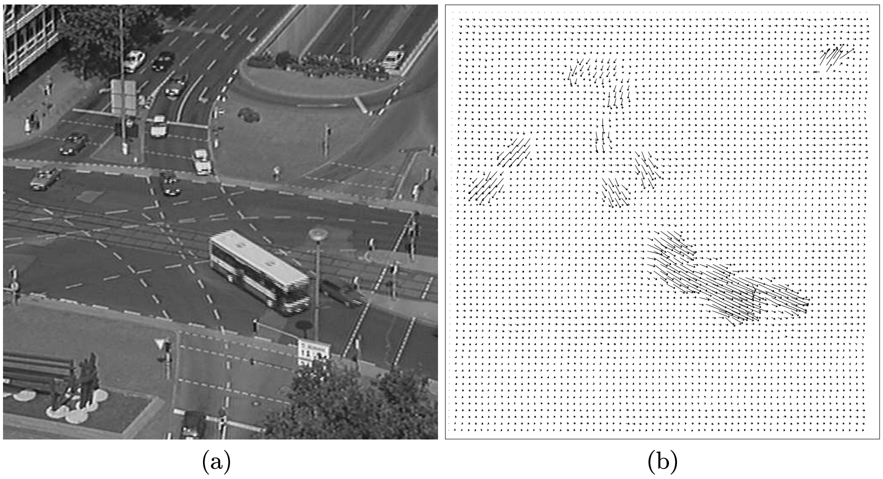
We found that the value of the offset around the flow fields was significant even through we use reflecting boundary conditions as suggested by Brox in our differentiation. By imposing a 2 or 6 pixel wide border around the image and thresholding very small velocities (magnitudes $> 10^{-1}$ or $> 10^{-3}$) we were able to improve accuracy by up to 10% with slightly less than 100% density (about 99%+).

5.2 Real Image Results

We tested our implementation on frames 5 and 6 of the **Ettlinger Tor** traffic sequence, as shown in Figure 4. The flow in Figure 4b qualitatively looks the

Table 3. Error analysis for various parameter settings

Parameter			Angular Error	
σ	α	γ	7×7	4-pt
1.3	80	100	4.69°	5.16°
1.3	60	120	5.45°	5.32°
2.0	90	130	5.26°	5.44°
2.0	25	20	6.16°	6.56°
0.6	70	80	8.44°	6.04°
3.0	60	20	8.92°	12.72°
1.3	80	0	8.54°	14.94°
0.0	80	100	20.8°	10.70°
1.3	0	100	77.02°	77.02°


(a)

(b)

Fig. 4. (a) Ettlinger Tor traffic sequence where the cars and the bus have motion. (d) Computed flow field for Ettlinger Tor traffic sequence.

same as in Brox et al. [1], except here we don't get flow for the part of bus roof top that has uniform white intensity.

6 Discussion

We have verified that Brox et al.'s 2-frame optical flow is better than all other existing methods for both versions of the Yosemite fly-through sequence. The combination of their hierarchical method where adjacent images in the pyramid only change their size slowly and their gradient constraint data term explains their superior performance. We tested Brox et al.'s implementation (executable only provided by Thomas Brox) and found our code often ran significantly slower than theirs. 7×7 averaging may give better results but it requires considerably more computations. We found the implementation of this algorithm to be quite challenging as the original paper did give all the details about some key issues that are essential from a programming point of view. The details were hidden about how to calculate the smoothing term in the linear system of equations. We had to resort to Brox's Ph.D. thesis [17] to find a way to compute $Div(\Psi'(*))$ (at Brox's suggestion) and we still could not get exactly his solution. Our failure to get Brox et al.'s SOR relaxation to work might explain the slightly higher error we sometimes get, compared to their results.

7 Future Work

We would like to improve our implementation of Brox et al.'s method to the point where we get their results. We would also like to implement their 3-frame variant using temporal differentiation. Papenberg et al. [2] proposed the use of two additional constraints into the model and have obtained even better Yosemite error results. It would be interesting to incorporate these constraints into our implementation. At this point, an investigation into the automatic determination of the weighting terms for the various terms in the regularization would be interesting. Lastly, it would be interesting to extend a Papenberg et al. implementation into 3D to allow, for example, the computation of accurate 3D optical flow from two/three volumes of gated MRI datasets.

Acknowledgements. We gratefully acknowledge financial support from NSERC through a Discovery grant. We also thank Thomas Brox for answering our many questions.

References

1. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J. (eds.) ECCV 2004. LNCS, vol. 3021, pp. 25–36. Springer, Heidelberg (2004)
2. Papenberg, N., Bruhn, A., Brox, T., Didas, S., Weickert, J.: Highly accurate optic flow computation with theoretically justified warping. International Journal of Computer Vision 67(2), 141–158 (2006)
3. Black, M., Anandan, A.: The robust estimation of multiple motions: Parametric and picewise smooth flows. Computer Vision and Image Understanding 64(1), 71–104 (1996)
4. Longuet-Higgins, H.C., Prazdny, K.: The interpretation of a moving retinal image. In: Proc. Royal Society London vol. B208, pp. 385–397 (1980)
5. Horn, B., Schunck, B.: Determining optical flow. Artificial Intelligence 17(1-3), 185–203 (1981)
6. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: International Joint Conference on Artificial Intelligence, pp. 674–679 (1981) (DARPA Image Understanding Workshop, pp. 121–130)

7. Ju, S.X., Black, M.J., Jepson, A.D.: Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In: Proceedings of Computer Vision and Pattern Recognition Conference, San Francisco, June 1996, pp. 307–314 (1996)
8. Lai, S.H., Vemuri, B.C.: Reliable and efficient computation of optical flow. International Journal of Computer Vision 29(2), 87–105 (2005)
9. Bab-Hadiashar, A., Suter, D.: Robust optic flow computation. International Journal of Computer Vision 29(1), 59–77 (1998)
10. Alvarez, L., Weickert, J., Sanchez, J.: Reliable estimation of dense optical flow fields with large displacements. International Journal of Computer Vision 39(1), 41–56 (2000)
11. Nagel, H.H., Enkelmann, W.: An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. IEEE Trans. Pattern Analysis and Machine Intelligence 1, 523–530 (1986)
12. Färneback, G.: Very high accuracy velocity estimation using orientation tensors, parametric motion, and simultaneous segmentation of the motion field. In: Proceedings of the Eighth IEEE International Conference on Computer Vision, Vancouver, Canada, July 2001, vol. I, pp. 171–177. IEEE Computer Society Press, Los Alamitos (2001)
13. Mémin, E., Pérez, P.: Hierarchical estimation and segmentation of dense motion fields. International Journal Computer Vision 46(2), 129–155 (2002)
14. Weickert, J., Bruhn, A., Schnörr, C.: Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. Technical report, Dept. of Mathematics, Saarland University, Saarbrücken, Germany (April 2003)
15. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. International Journal of Computer Vision 61(3), 1–21 (2005)
16. Barron, J., Fleet, D., Beauchemin, S.: Performance of optical flow techniques. International Journal of Computer Vision 12(1), 43–77 (1994)
17. Brox, T.: From Pixels to Regions: Partial Differential Equations in Image Analysis. Ph.d thesis, Mathematical Image Analysis Group, Dept. of Mathematics and Computer Science, Sarrland University, Saarbrückem, Germany (April 2005)
18. Pollard, S., Porrill, J., Thacker, N.: Tina programmer's guide. Technical report, Medical Biophysics and Clinical Radiology, University of Manchester (1999), www.niac.man.ac.uk/Tina/docs/programmers/programmers_guide.html
19. Barron, J.: The integration of optical flow into tinatool. Technical report, Dept. of Computer Science, The Univ. of Western Ontario. TR601 (report for OSMIA (Open Source Medical Image Analysis, a EU 5^{th} Framework Programme project)) (2003)
20. Simoncelli, E.: Design of multi-dimensional derivative filters. In: IEEE Int. Conf. Image Processing, pp. 790–793. IEEE Computer Society Press, Los Alamitos (1994)