

A Duality Based Approach for Realtime TV- L^1 Optical Flow

C. Zach¹, T. Pock², and H. Bischof²

¹ VRVis Research Center

² Institute for Computer Graphics and Vision, TU Graz

Abstract. Variational methods are among the most successful approaches to calculate the optical flow between two image frames. A particularly appealing formulation is based on total variation (TV) regularization and the robust L^1 norm in the data fidelity term. This formulation can preserve discontinuities in the flow field and offers an increased robustness against illumination changes, occlusions and noise. In this work we present a novel approach to solve the TV- L^1 formulation. Our method results in a very efficient numerical scheme, which is based on a dual formulation of the TV energy and employs an efficient point-wise thresholding step. Additionally, our approach can be accelerated by modern graphics processing units. We demonstrate the real-time performance (30 fps) of our approach for video inputs at a resolution of 320×240 pixels.

1 Introduction

The recovery of motion from images is a major task of biological and artificial vision systems. The main objective of optical flow methods is to compute a flow field estimating the motion of pixels in two consecutive image frames. Since optical flow is an highly ill-posed inverse problem, using pure intensity-based constraints generally results in an under-determined system of equations, which is generally known as the *aperture problem*. In order to solve this problem some kind of regularization is needed to obtain physically meaningful displacement fields.

In their seminal work [13], Horn and Schunck studied a variational formulation of the optical flow problem.

$$\min_{\mathbf{u}} \left\{ \int_{\Omega} |\nabla u_1|^2 + |\nabla u_2|^2 d\Omega + \lambda \int_{\Omega} (I_1(\mathbf{x} + \mathbf{u}(\mathbf{x})) - I_0(\mathbf{x}))^2 d\Omega \right\}. \quad (1)$$

I_0 and I_1 is the image pair, $\mathbf{u} = (u_1(\mathbf{x}), u_2(\mathbf{x}))^T$ is the two-dimensional displacement field and λ is a free parameter. The first term (regularization term) penalizes high variations in \mathbf{u} to obtain smooth displacement fields. The second term (data term) is also known as the optical flow constraint. It assumes, that the intensity values of $I_0(\mathbf{x})$ do not change during its motion to $I_1(\mathbf{x} + \mathbf{u}(\mathbf{x}))$.

Since the Horn-Schunck model penalizes deviations in a quadratic way, it has two major limitations. It does not allow for discontinuities in the displacement field, and it does not handle outliers in the data term robustly. To overcome these limitations, several models including robust error norms and higher order data terms have been proposed. Since discontinuities in the optical flow appear often in conjunction with high image gradients, several authors replace the homogeneous regularization in the Horn-Schunck model with an anisotropic diffusion approach [15, 19]. Other proposed modifications substitute the squared penalty functions in the Horn-Schunck model with more robust variants. Blake and Anandan [5] apply estimators from robust statistics and obtain a robust and discontinuity preserving formulation for the optical flow energy. Aubert et al. [3] analyze energy functionals for optical flow incorporating an L^1 data fidelity term and a general class of discontinuity preserving regularization forces. Papenberg et al. [16] employ a differentiable approximation of the TV (resp. L^1) norm and formulate a nested iteration scheme to compute the displacement field.

Most approaches for optical flow computation replace the nonlinear intensity profile $I_1(\mathbf{x} + \mathbf{u}(\mathbf{x}))$ by a first order Taylor approximation to linearize the problem locally. Since such approximation is only valid for small displacements, additional techniques are required to determine the optical flow correctly for large displacements. Scale-space approaches [1] and coarse-to-fine warping (e.g. [2, 14, 7]) provide solutions to optical flow estimation with large displacements.

In several applications, such as autonomous robot navigation, it is necessary to calculate displacement fields in real-time. Real-time optical flow techniques typically consider only the data fidelity term to generate displacement fields [10, 18]. One of the first variational approaches to compute the optical flow in real-time was presented by Bruhn et al. [8, 9]. In their work a highly efficient multi-grid approach is employed to obtain real-time or near real-time performance. The aim of their approach is very similar to our objective: obtaining robust and discontinuity preserving solutions for optical flow with highly efficient implementations. Nevertheless, we utilize a completely different solution strategy as described in the next sections.

2 TV- L_1 Optical Flow

In the basic setting two image frames I_0 and $I_1 : (\Omega \subseteq \mathbb{R}^2) \rightarrow \mathbb{R}$ are given. The objective is to find the disparity map $\mathbf{u} : \Omega \rightarrow \mathbb{R}^2$, which minimizes an image-based error criterion together with a regularization force. In this work we focus on the plain intensity difference between pixels as the image similarity score. Hence, the target disparity map \mathbf{u} is the minimizer of

$$\int_{\Omega} \left\{ \lambda \phi(I_0(\mathbf{x}) - I_1(\mathbf{x} + \mathbf{u}(\mathbf{x}))) + \psi(\mathbf{u}, \nabla \mathbf{u}, \dots) \right\} d\mathbf{x}, \quad (2)$$

where $\phi(I_0(\mathbf{x}) - I_1(\mathbf{x} + \mathbf{u}(\mathbf{x})))$ is the image data fidelity, and $\psi(\mathbf{u}, \nabla \mathbf{u}, \dots)$ depicts the regularization term inducing the shape prior. λ weights between the

data fidelity and the regularization force. Selecting $\phi(x) = x^2$ and $\psi(\nabla \mathbf{u}) = |\nabla \mathbf{u}|^2$ results in the Horn-Schunck model [13].

The choice of $\phi(x) = |x|$ and $\psi(\nabla \mathbf{u}) = |\nabla \mathbf{u}|$ yields to the following functional consisting of an L_1 data penalty term and total variation regularization:

$$E = \int_{\Omega} \left\{ \lambda |I_0(\mathbf{x}) - I_1(\mathbf{x} + \mathbf{u}(\mathbf{x}))| + |\nabla \mathbf{u}| \right\} d\mathbf{x}. \quad (3)$$

Although Eq. 3 seems to be simple, it offers some computational difficulties. The main reason is that both the regularization term and the data term are not continuously differentiable. One approach is to replace $\phi(x) = |x|$ and $\psi(\nabla \mathbf{u})$ with differentiable approximations $\phi_{\varepsilon}(x^2) = \sqrt{x^2 + \varepsilon^2}$ and $\psi_{\varepsilon}(\nabla \mathbf{u}) = \sqrt{|\nabla \mathbf{u}|^2 + \varepsilon^2}$, and to apply a numerical optimization technique on this slightly modified functional (e.g. [12, 7]).

In this paper we employ a rather different approach. In [11] Chambolle proposed an efficient and exact numerical scheme to solve the Rudin-Osher-Fatemi energy [17] for total variation based image denoising. In the following, we will describe how to adopt this approach for the optical flow case.

2.1 The 1D Stereo Case

In this section we restrict the disparities to be non-zero only in the horizontal direction, e.g. a normalized stereo image pair is provided. Hence, $\mathbf{u}(\mathbf{x})$ reduces to a scalar $u(\mathbf{x})$, and we use the (sloppy) notation $\mathbf{x} + u(\mathbf{x})$ for $\mathbf{x} + (u(\mathbf{x}), 0)^T$. The following derivation is based on [4], but adapted to the stereo/optical flow setting. At first, we linearize image I_1 near $\mathbf{x} + u_0$, i.e.

$$I_1(\mathbf{x} + u) = I_1(\mathbf{x} + u_0) + (u - u_0) I_1^x(\mathbf{x} + u_0),$$

where u_0 is a given disparity map and I_1^x is the derivative of the image intensity I_1 wrt. the x -direction. Using the first order Taylor approximation for I_1 means, that the following procedure needs to be embedded into an iterative warping approach to compensate for image nonlinearities. Additionally, a multi-level approach is employed to allow large disparities between the images.

For fixed u_0 and using the linear approximation for I_1 , the TV- L_1 functional (Eq. 3) now reads as:

$$E = \int_{\Omega} \left\{ \lambda |u I_1^x + I_1(\mathbf{x} + u_0) - u_0 I_1^x - I_0| + |\nabla u| \right\} d\mathbf{x}. \quad (4)$$

In the following, we denote the current residual $I_1(\mathbf{x} + u_0) + (u - u_0) I_1^x - I_0$ by $\rho(u, u_0, \mathbf{x})$ (or just $\rho(u)$ by omitting the explicit dependency on u_0 and \mathbf{x}). Moreover, we introduce an auxiliary variable v and propose to minimize the following convex approximation of Eq. 4:

$$E_{\theta} = \int_{\Omega} \left\{ |\nabla u| + \frac{1}{2\theta} (u - v)^2 + \lambda |\rho(v)| \right\} d\mathbf{x}, \quad (5)$$

where θ is a small constant, such that v is a close approximation of u . This convex minimization problem can be optimized by alternating steps updating either u or v in every iteration:

1. For v being fixed, solve

$$\min_u \int_{\Omega} \left\{ |\nabla u| + \frac{1}{2\theta}(u-v)^2 \right\} d\mathbf{x}. \quad (6)$$

This is the total variation based image denoising model of Rudin, Osher and Fatemi [17].

2. For u being fixed, solve

$$\min_v \int_{\Omega} \left\{ \frac{1}{2\theta}(u-v)^2 + \lambda |\rho(v)| \right\} d\mathbf{x}. \quad (7)$$

This minimization problem can be solved point-wise, since it does not depend on spatial derivatives of v .

An efficient solution for the first step (Eq. 6) was proposed in [11], which uses a dual formulation of Eq. 6 to derive an efficient and globally convergent scheme. Since this algorithm is an essential part of our method, we reproduce the relevant results from [11]:

Proposition 1 *The solution of Eq. (6) is given by*

$$u = v - \theta \operatorname{div} \mathbf{p}, \quad (8)$$

where $\mathbf{p} = (p^1, p^2)$ fulfills

$$\nabla(\theta \operatorname{div} \mathbf{p} - v) = |\nabla(\theta \operatorname{div} \mathbf{p} - v)| \mathbf{p}, \quad (9)$$

which can be solved by the following iterative fixed-point scheme:

$$\mathbf{p}^{k+1} = \frac{\mathbf{p}^k + \tau \nabla(\operatorname{div} \mathbf{p}^k - v/\theta)}{1 + \tau |\nabla(\operatorname{div} \mathbf{p}^k - v/\theta)|}, \quad (10)$$

where $\mathbf{p}^0 = \mathbf{0}$ and the time step $\tau \leq 1/8$.

The next proposition characterizes the minimizer of the second part (Eq. 7):

Proposition 2 *The solution of the minimization task in Eq. 7 is given by the following thresholding step:*

$$v = u + \begin{cases} \lambda \theta I_1^x & \text{if } \rho(u) < -\lambda \theta (I_1^x)^2 \\ -\lambda \theta I_1^x & \text{if } \rho(u) > \lambda \theta (I_1^x)^2 \\ -\rho(u)/I_1^x & \text{if } |\rho(u)| \leq \lambda \theta (I_1^x)^2. \end{cases} \quad (11)$$

This means, that the image residual $\rho(v)$ is allowed to vanish, if the required step from u to v is sufficiently small. Otherwise, v makes a bounded step from u , such that the magnitude of the residual decreases.

The proposition above can be shown directly by analyzing the three possible cases, $\rho(v) > 0$ (inducing $v = u - \lambda \theta I_1^x$), $\rho(v) < 0$ ($v = u + \lambda \theta I_1^x$) and $\rho(v) = 0$ ($v = u - \rho(u)/I_1^x$).

2.2 Generalization to Higher Dimensions

In this section we extend the method introduced in the previous section to optical flow estimation, i.e. an N -dimensional displacement map \mathbf{u} is determined from two given N -D images I_0 and I_1 . The first order image residual $\rho(\mathbf{u}, \mathbf{u}_0, \mathbf{x})$ wrt. a given disparity map \mathbf{u}_0 is now $I_1(\mathbf{x} + \mathbf{u}_0) + \langle \nabla I_1, \mathbf{u} - \mathbf{u}_0 \rangle - I_0(\mathbf{x})$. Additionally, we write u_d for the d -th component of \mathbf{u} ($d \in \{1, \dots, N\}$).

The generalization of Eq. 5 to more dimensions is the following energy:

$$E_\theta = \int_{\Omega} \left\{ \sum_d |\nabla u_d| + \sum_d \frac{1}{2\theta} (u_d - v_d)^2 + \lambda |\rho(\mathbf{v})| \right\} d\mathbf{x}. \quad (12)$$

Similar to the stereo setting, minimizing this energy can be performed by alternating optimization steps:

1. For every d and fixed v_d , solve

$$\min_{u_d} \int_{\Omega} \left\{ |\nabla u_d| + \frac{1}{2\theta} (u_d - v_d)^2 \right\} d\mathbf{x}. \quad (13)$$

This minimization problem is identical to Eq. 6 and can be solved by the same procedure. Note, that the dual variables are introduced for every dimension, e.g. Eq. 8 now reads as

$$u_d = v_d - \theta \operatorname{div} \mathbf{p}_d. \quad (14)$$

2. For \mathbf{u} being fixed, solve

$$\min_{\mathbf{v}} \sum_d \frac{1}{2\theta} (u_d - v_d)^2 + \lambda |\rho(\mathbf{v})|. \quad (15)$$

The following proposition generalizes the thresholding step from Proposition 2 to higher dimensions:

Proposition 3 *The solution of the minimization task in Eq. 15 is given by the following thresholding step:*

$$\mathbf{v} = \mathbf{u} + \begin{cases} \lambda \theta \nabla I_1 & \text{if } \rho(\mathbf{u}) < -\lambda \theta |\nabla I_1|^2 \\ -\lambda \theta \nabla I_1 & \text{if } \rho(\mathbf{u}) > \lambda \theta |\nabla I_1|^2 \\ -\rho(\mathbf{u}) \nabla I_1 / |\nabla I_1|^2 & \text{if } |\rho(\mathbf{u})| \leq \lambda \theta |\nabla I_1|^2. \end{cases} \quad (16)$$

This proposition essentially states, that the N -dimensional optimization problem can be reduced to a one-dimensional thresholding step, since \mathbf{v} always lies on the line l^\perp going through \mathbf{u} with direction ∇I_1 (for every \mathbf{x}). This can be seen as follows: The first part in Eq. 15, $\sum_d (u_d - v_d)^2 / 2\theta$, is basically the squared distance of \mathbf{v} to \mathbf{u} , and the second part, $\lambda |\rho(\mathbf{v})|$, is the unsigned distance to the line $l : \rho(\mathbf{w}) = 0$, i.e. $I_1(\mathbf{x} + \mathbf{u}_0) + \langle \nabla I_1, \mathbf{w} - \mathbf{u}_0 \rangle - I_0(\mathbf{x}) = 0$. If we consider all \mathbf{v}_μ with a fixed distance μ to \mathbf{u} , then the functional in Eq. 15 is minimized for the \mathbf{v}_μ closest to the line l (with minimal normal distance). This is also valid for the true minimizer, hence the optimum for Eq. 15 is on l^\perp . In addition, the one-dimensional thresholding step in gradient direction can be applied (Proposition 2), resulting in the presented scheme.

3 Implementation

This section gives details on the employed numerical procedure and on the GPU-accelerated implementation for the proposed TV- L^1 optical flow approach. Although the discussion in Section 2.2 is valid for any image dimension $N \geq 2$, our GPU-based implementation is specifically tailored for the case $N = 2$.

3.1 Numerical Scheme

The generally non-convex energy functional for optical flow (Eq. 3) becomes a convex minimization problem after linearization of the image intensities (Eq. 4), but this linearization is only valid for small displacements. Hence, the energy minimization procedure is embedded into a coarse-to-fine approach to avoid convergence to unfavorable local minima. We employ image pyramids with a downsampling factor of 2 for this purpose. Beginning with the coarsest level, we solve Eq. 3 at each level of the pyramid and propagate the solution to the next finer level. This solution is further used to compute the coefficients of the linear residual function ρ by sampling I_0 and I_1 using the corresponding pyramid levels. Hence, the warping step for I_1 takes place only once per level. ∇I_1 is approximated by central differences. At the beginning of a new level, \mathbf{v} is initialized with \mathbf{u} , and all \mathbf{p}_d are set to $\mathbf{0}$. At the coarsest level, the displacement field \mathbf{u} starts with $\mathbf{0}$.

Avoiding poor local minima is not the only advantage of the coarse-to-fine approach. It turns out, that the filling-in process induced by the regularization occurring in textureless region is substantially accelerated by a hierarchical scheme as well.

The minimization procedure alternates one step of the fixed-point scheme to update all \mathbf{p}_d (and therefore \mathbf{u} , Eq. 10) with the thresholding step from Proposition 3 to improve \mathbf{v} . The implementation of the fixed-point update (Eq. 10) uses backward differences to approximate $\mathbf{div} \mathbf{p}$ and forward differences for the numerical gradient computation in order to have mutually adjoint operators [11].

3.2 Acceleration by Graphics Processing Units

Numerical methods working on regular grids, e.g. rectangular image domains, can be effectively accelerated by modern graphics processing units (GPUs). We employ the huge computational power and the parallel processing capabilities of GPUs to obtain a fully accelerated implementation of our optical flow approach. The GPU-based procedure is essentially a straightforward Cg implementation of the numerical schemes (Eqs. 10 and 16) with few modifications described as follows.

If we write down the alternating minimization steps explicitly, iteration k performs the following updates on \mathbf{u} , \mathbf{v} and \mathbf{p}_d :

$$\begin{aligned}
1a. \quad & \mathbf{v}^{k+1} \leftarrow TH(\mathbf{u}^k) \\
1b. \quad & u_d^{k+1} \leftarrow v_d^{k+1} - \theta \mathbf{div} \mathbf{p}_d^k \quad \text{for } d \in \{1, 2\} \\
2. \quad & \mathbf{p}_d^{k+1} \leftarrow \frac{\mathbf{p}_d^k + \tau/\theta \nabla u_d^{k+1}}{1 + \tau/\theta |\nabla u_d^{k+1}|} \quad \text{for } d \in \{1, 2\},
\end{aligned} \tag{17}$$

where $TH(\cdot)$ denotes the thresholding step from Eq. 16. These steps can be immediately implemented on the GPU by appropriate fragment programs using two rendering passes.³ The first pass implements steps 1a and 1b from Eq. 17. The values \mathbf{v}^{k+1} are used only temporarily within the shader program and need not to be saved explicitly. \mathbf{u}^{k+1} is written to the target texture. The second shader program corresponds with step 2 from Eq. 17. It turns out, that the utilization of the fragment processors can be improved by updating \mathbf{u} and \mathbf{p}_d for two pixels simultaneously. The shader programs work on the left and on the right half of the images in parallel, with appropriate handling of border pixels. Our implementation encodes the two components of \mathbf{u} using full 32-bit precision, and the overall four components of \mathbf{p}_1 and \mathbf{p}_2 are compressed to 16-bit half precision floating point numbers.

We currently use a fixed but tunable number of iterations on each level in our implementation, since determining the maximum update $|\mathbf{u}^{k+1} - \mathbf{u}^k|$ still requires an expensive reduction operation even on modern GPUs.

4 Results

At first, we provide timing results for our optical flow approach depicted in Table 1. Two hardware setups were used to obtain the timing results: a desktop PC equipped with a NVidia GeForce 7800 GS card, and a high-end laptop supplied with a NVidia GeForce Go 7900 GTX graphics board. The timing results were obtained under the Linux operating system with recent OpenGL graphics drivers and the Cg 1.5 toolkit. The timings in Table 1 are given in frames per second for the depicted fixed number of iterations on each level of the image pyramid. The measured times include the texture uploads to video memory and the final visualization of the obtained displacement field. The timing results indicate, that real-time performance with more than 30 frames per second can be achieved at 256×256 pixels resolution with our approach. Frames from a live video demo application are shown in Figure 1, which continuously reads images from a firewire camera and visualizes the optical flow for adjacent frames. Real-time performance can be achieved with the mobile hardware setup.

Figure 2 shows common test sequences for optical flow, in particular the *Ettlinger Tor*, the *Rheinhafen* and the *Yosemite* sequences, and their respective flow

³ A single pass variant using only one shader program is possible as well, but the observed performance is inferior in almost all cases.

Image resolution	GeForce 7800 GS			GeForce Go 7900 GTX		
	50 It.	100	200	50 It.	100	200
128×128	56	32.1	17.5	95	57.6	30.9
256×256	18	9.6	5	34.1	17.5	8.9
512×512	5	2.6	1.3	9.3	4.7	2.3

Table 1. Observed frame rates at different image resolutions and with varying number of iterations on our tested hardware.

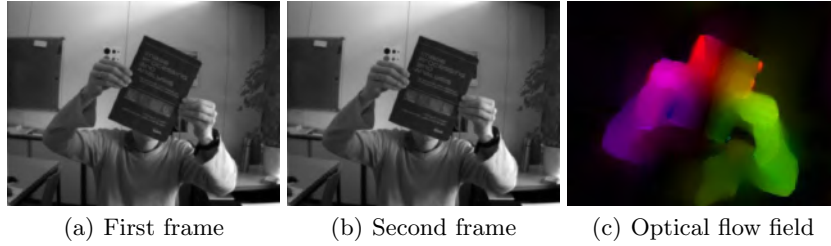


Fig. 1. Captured frames and generated optical flow field using our live video application. The image resolution is 320×240 , and 50 iterations are performed on each level of the image pyramid. The framerate is close to 30 frames per second in this setting. The flow field is visualized using hue to indicate the direction and intensity for the magnitude.

fields. The results for these datasets indicate, that the reduced 16-bit resolution for the dual variables \mathbf{p}_d does not severely affect the quality of the obtained flow fields. Table 2 specifies the obtained average angular error (AAE) of the flow field for the Yosemite dataset wrt. the provided ground truth. If the completely homogeneous sky region is excluded from the AAE calculation, the flow field is essentially converged after 50 iterations. Enabling more iterations yields to slightly inferior results, since the $TV-L^1$ energy favors piecewise constant flow fields in the limit. If the sky region is included in the evaluation, the AAE error decreases by increasing the number of iterations. In this case the flow field in the sky region converges relatively slowly to the zero displacement.

	50 It.	150	250
Without sky	2.85°	2.88°	2.89°
With sky	5.06°	3.7°	3.27°

Table 2. Average angular error for frame 8 and 9 of the Yosemite sequence at different number of iterations.

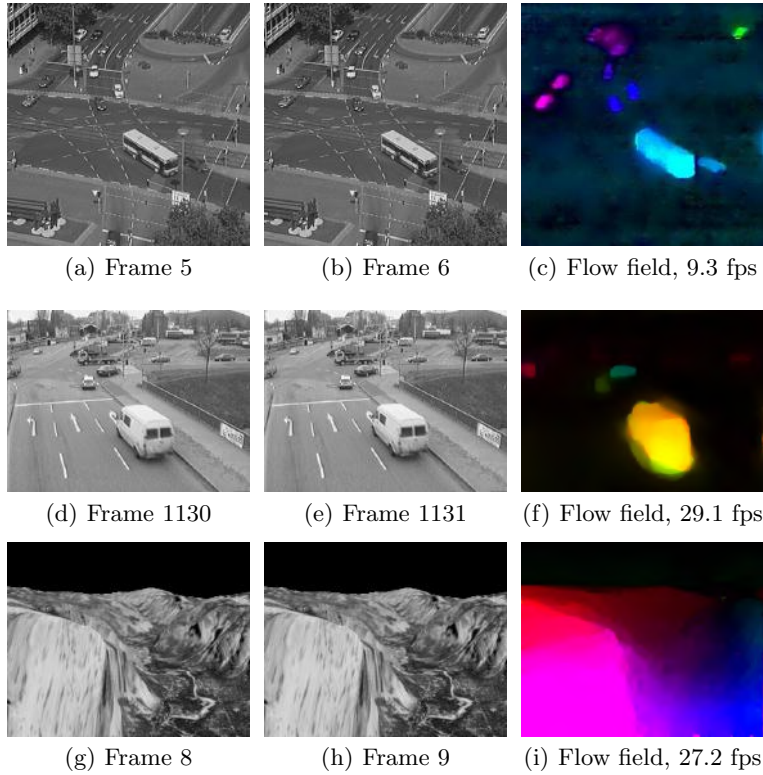


Fig. 2. Sample images and obtained flow fields for the *Ettlinger Tor* (512×512 pixels), the *Rheinhafen* (320×240 pixels) and the *Yosemite* (320×256 pixels) sequences.

5 Conclusion

We presented a novel approach for efficient optical flow estimation using a TV- L_1 energy functional. We developed a novel fast numerical scheme which can be efficiently implemented on modern graphics processing units. With this we can show real-time performance using online video streams. The correctness and quality of our implementation is demonstrated on several datasets.

Future work includes the extension of our approach to handle color images as well. Additionally, other image similarity measures, e.g. based on intensity gradients, need to be further explored. The edge preserving nature of total variation can be enhanced, if a suitable weighted TV-norm/active contour model is applied [6]. Future work will address the incorporation of these feature for stereo and optical flow estimation.

Finally, switching from an OpenGL-based implementation to the newer CUDA GPU programming framework is expected to increase the observed performance substantially.

References

1. L. Alvarez, J. Weickert, and J. Sánchez. A scale-space approach to nonlocal optical flow calculations. In *Proceedings of the Second International Conference on Scale-Space Theories in Computer Vision*, pages 235–246, 1999.
2. P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *Int. J. Comput. Vision*, 2:283–310, 1989.
3. G. Aubert, R. Deriche, and P. Kornprobst. Computing optical flow via variational techniques. *SIAM J. Appl. Math.*, 60(1):156–182, 1999.
4. J.-F. Aujol, G. Gilboa, T. Chan, and S. Osher. Structure-texture image decomposition—modeling, algorithms, and parameter selection. *Int. J. Comput. Vision*, 67(1):111–136, 2006.
5. M.J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *ICCV93*, pages 231–236, 1993.
6. X. Bresson, S. Esedoglu, P. Vanderghelynst, J. Thiran, and S. Osher. Fast Global Minimization of the Active Contour/Snake Model. *Journal of Mathematical Imaging and Vision*, 2007.
7. T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision (ECCV)*, pages 25–36, 2004.
8. A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnörr. Variational optical flow computation in real time. *IEEE Transactions on Image Processing*, 14(5):608–615, 2005.
9. A. Bruhn, J. Weickert, T. Kohlberger, and C. Schnörr. A multigrid platform for real-time motion computation with discontinuity-preserving variational methods. *Int. J. Comput. Vision*, 70(3):257–277, 2006.
10. T. A. Camus. Real-time quantized optical flow. *Journal of Real-Time Imaging*, 3:71–86, 1997. Special Issue on Real-Time Motion Analysis.
11. A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1–2):89–97, 2004.
12. T. F. Chan, G. H. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. In *ICAOS '96 (Paris, 1996)*, volume 219, pages 241–252, 1996.
13. B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
14. E. Mémin and P. Pérez. Hierarchical estimation and segmentation of dense motion fields. *Int. J. Comput. Vision*, 46(2):129–155, 2002.
15. H.-H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8:565–593, 1986.
16. N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. *Int'l J. Computer Vision*, pages 141–158, 2006.
17. L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
18. R. Strzodka and C. Garbe. Real-time motion estimation and visualization on graphics cards. In *IEEE Visualization 2004*, pages 545–552, 2004.
19. J. Weickert and T. Brox. Diffusion and regularization of vector- and matrix-valued images. *Inverse Problems, Image Analysis and Medical Imaging. Contemporary Mathematics*, 313:251–268, 2002.