Intermediate Flow Field Filtering in Energy Based Optic Flow Computations

Laurent Hoeltgen, Simon Setzer, and Michael Breuß

Mathematical Image Analysis Group,
Faculty of Mathematics and Computer Science
Saarland University, Campus E1.1, 66041 Saarbrücken, Germany
{hoeltgen,setzer,breuss}@mia.uni-saarland.de

Abstract. The Euler-Lagrange framework and splitting based methods are among the most popular approaches to solve variational optic flow problems. These methods are commonly embedded in a coarse-to-fine strategy to be able to handle large displacements. While the use of a denoising filter inbetween the warping is an important tool for splitting based approaches, such a practice is rather uncommon for the Euler-Lagrange method. The question arises, why there is this surprising difference in optic flow methods. In previous works it has also been stated that the use of such a filtering leads to a modification of the underlying energy functional, thus, there seems to be a difference in the energies that are actually minimised depending on the chosen algorithmic approach.

The goal of this paper is to address these fundamental issues. By a detailed numerical study we show in which way a filtering affects the evolution of the energy for the above mentioned frameworks. Doing so, we not only give many new insights on the use of filtering steps, we also bridge an important methodical gap between the two commonly used implementation approaches.

Keywords: Optic flow, Warping, Median filter, Euler-Lagrange equation, Splitting based methods.

1 Introduction

The most successful class of methods for optic flow estimation is based on minimising an energy formulation[1]. Such an energy combines a data term expressing the constancy of some property of the input images and a smoothness term penalising fluctuations in the flow field. Based on the seminal work of Horn and Schunck [1], sophisticated models have been developed and a high degree of accuracy and robustness has been achieved, see e.g. [2,3,4,5,6,7,8,9,10,11] for some influential publications.

One may distinguish energy based optic flow methods not only by the underlying model, but also by the algorithmic realisation. There are two main approaches for implementations. The first one, we call it DEO (discrete energy optimisation), relies on the direct discretisation of the energy functional, followed by the application of an optimisation scheme for this discrete energy. The second one is to discretise the Euler-Lagrange

[1] http://vision.middlebury.edu/flow/

Y. Boykov et al. (Eds.): EMMCVPR 2011, LNCS 6819, pp. 315–328, 2011.

(EL) equations that describe a necessary condition for a minimiser of the continuous energy functional. In recent years, both approaches have achieved a high technical level, employing latest optimisation techniques [4,12], multigrid methods [13] and parallel implementations on graphics hardware [14,15].

While the individual paths in modeling and implementation for the optic flow problem have received much attention, there are not many papers that deal with fundamental principles such as the use of the warping strategy, see for example [3]. A quite influential recent work in this respect is the paper of Sun, Roth and Black [16], where important building blocks of modern algorithms are examined. One issue of specific importance discussed there is the application of a denoising step inbetween the warping levels as performed in [17]. It is stressed that the use of the median filter to denoise the computed flow fields leads to better results while it also modifies the energy functional that is minimised, as additional terms representing the filter need to be added.

For motivating our present work, let us point out explicitly that the investigations in [16] lead to some important questions. On the one hand, a filtering of flow fields during warping is a very important building block for use within the DEO approach, cf. [4,16,17]. On the other hand, it seems to be completely unusual to apply such a filtering in the context of the EL method; see for instance the works [3,11,18] that are supposed to employ sophisticated techniques in that direction and where a denoising step is not reported. Given that both approaches are supposed to finally serve the same purpose, it seems to be interesting to explore why there is this important (and somewhat surprising) methodical gap. Let us point out clearly, that this issue is not only an algorithmic one, but it leads to important questions concerning the modeling of optic flow. Note that Sun et al. [16] have shown that the median filtering of flow fields is equivalent to the modification of the original energy functional. Therefore, the optic flow models actually addressed by the two implementation approaches seem to differ on a fundamental level, as the DEO method solves a model which includes terms corresponding to the median filter. The question arises, if the EL approach should be modified accordingly from the beginning on.

In our present paper, we clarify these issues. Thereby, we complement and extend the discussion of Sun et al. [16] in several ways, and we also bridge the gap between the DEO and EL implementation approaches.

Let us note that the results in [16] are supposed to hold for general energy-based optic flow models. In order to put our investigation in concrete terms, we make use of an L^1-penaliser in both data and smoothness term, and we employ, for simplicity, just the grey value constancy assumption. This can be considered as a basic modern optic flow model, cf. [3,15,17]. Concerning the numerical realisation, we employ a version of the recent primal-dual hybrid gradient (PDHG) algorithm for optimising the discrete energy in the DEO method, see e.g. [4]. For the EL approach we employ the same strategy as in [3] where we make use of the classic successive over-relaxation (SOR) method [19] for solving the arising systems of linear equations. These are embedded in a loop of fixed-point iterations for updating the nonlinearities. The discretisation of the derivatives is in both cases fourth-order accurate, as often employed in schemes today.

Thus, using one and the same optic flow model, we give a detailed numerical study of the approaches, to optimise the discrete energy and the Euler-Lagrange formalism, with respect to median filtering of flow fields. The results are:

1. Concerning the DEO approach, we show that the filtering effect is beneficial when employing a *low number of optimisation steps*. While the result with median filtering is better than without, the energy of that solution is in comparison substantially higher, confirming results in [16] that were also obtained by a discrete optimisation strategy.
2. We obtain a corresponding result for the EL approach when we employ a *low number of fixed-point iterations* to update nonlinearities.
3. When increasing the number of optimisation or fixed-point iterations respectively, the filtering becomes less effective. By employing a *large number of iterations*, one obtains equivalent results for the algorithms with and without filter steps.
4. The number of iterations needed to reach approximately convergence is in our tests higher for the PDHG algorithm than for the fixed-point iterations in the EL method, while these iterations are numerically roughly of the same cost. Thus, one may infer that flow field filtering is in general more useful for the DEO approach. For numerical setups as often employed in the EL method, a filtering is not necessary and a modification of the energy is not a suitable model.

As another contribution, we compare the classic median filtering on fixed masks to its data-adaptive application [20]. The latter serves not only as a reference filter, it also evaluates the idea of structure adaptivity while being as close as possible to the standard median filtering. We obtain the same behaviour as above for this reference method. This shows that data-adaptivity alone is not enough in order to construct a better filter, one needs a combination of more sophisticated modeling components such as proposed in [16].

Our paper is organised as follows. In Section 2 we briefly review the optic flow model used here. This is followed by a description of the algorithms in Section 3. In Section 4, we perform extensive numerical experiments illustrating the results mentioned above.

2 The Optic Flow Model

Variational formulations for the computation of optic flow have proven to be highly successful since they offer transparent modeling while yielding, at the same time, excellent results. They have been studied for three decades, beginning with the work of Horn and Schunck [1], and, as a consequence, there exist a multitude of formulations that take into account various model assumptions. In the following, we will consider the TV-L^1 energy functional [3,15]. It is given by

$$E[u,v] = \int_\Omega |f(x+u, y+v, t+1) - f(x,y,t)| + \lambda \left\| \begin{pmatrix} \nabla u \\ \nabla v \end{pmatrix} \right\| \mathrm{d}x\mathrm{d}y \qquad (1)$$

where f is our image sequence, u and v the components of the sought flow field in x- resp. y-direction, Ω is the considered spatial image domain, and λ is a regularisation

weight specifying the relative importance between the two penalty terms. Furthermore, we have employed in eq. (1) the definition

$$\left\| \begin{pmatrix} \nabla u \\ \nabla v \end{pmatrix} \right\| := \sqrt{(\partial_x u)^2 + (\partial_y u)^2 + (\partial_x v)^2 + (\partial_y v)^2} \tag{2}$$

This energy exhibits a data term that models brightness constancy as well as a regulariser that enforces piecewise smoothness of u and v while still respecting discontinuities in the flow field. In comparison with quadratic penalty terms (as used in [1]), this energy formulation has the advantage that it is more robust towards strong outliers. However, due to the non-differentiability of the L^1-norm, its minimisation is considerably more difficult than for quadratic penalisers. Nevertheless, the good results that can be achieved with non-quadratic terms justify the frequent use of this formulation in various forms, see e.g. [11,15,17].

3 Numerical Approaches

3.1 The Euler-Lagrange Framework

The EL framework, as employed in e.g. [3,11], considers the Euler-Lagrange equations of the corresponding variational formulation and tries to solve the resulting system of partial differential equations (PDEs). The EL setting requires the occurring terms to be differentiable. Therefore, we approximate eq. (1) by considering the function

$$\Psi\left(s^2\right) := \sqrt{s^2 + \varepsilon^2} \tag{3}$$

with a small parameter ε which leads us to the following energy formulation

$$\begin{aligned} E\left[u, v\right] = \int_\Omega \Psi\big(\left(f\left(x+u, y+v, t+1\right) - f\left(x, y, t\right)\right)^2\big) \\ + \lambda\, \Psi\left(|\nabla u|^2 + |\nabla v|^2\right) \,\mathrm{d}x\mathrm{d}y \end{aligned} \tag{4}$$

According to the calculus of variations, the minimiser of eq. (4) must necessarily fulfil the following system of nonlinear PDEs

$$\begin{aligned} \Psi'\left(I_z^2\right) \cdot I_x I_z - \lambda \operatorname{div}\left(\Psi'\left(|\nabla u|^2 + |\nabla v|^2\right) \nabla u\right) = 0 \\ \Psi'\left(I_z^2\right) \cdot I_y I_z - \lambda \operatorname{div}\left(\Psi'\left(|\nabla u|^2 + |\nabla v|^2\right) \nabla v\right) = 0 \end{aligned} \tag{5}$$

with reflecting boundary conditions and where we used the abbreviations

$$\begin{aligned} I_x &:= \partial_x f\left(x+u, y+v, t+1\right) \\ I_y &:= \partial_y f\left(x+u, y+v, t+1\right) \\ I_z &:= f\left(x+u, y+v, t+1\right) - f\left(x, y, t\right) \end{aligned} \tag{6}$$

Our means to deal with these equations will essentially be identical to the approach presented in [3]. Let us nevertheless go into some detail as this is important for the understanding of our numerical study. We solve the system in (5) through a fixed-point

iteration which we embed further into a multiscale coarse-to-fine warping strategy. Starting with the zero flow field on the coarsest level, we consider iteratively

$$\begin{aligned}\Psi'\left(\left(I_z^{k+1}\right)^2\right)\cdot I_x^k I_z^{k+1} - \lambda \operatorname{div}\left(\Psi'\left(|\nabla u^{k+1}|^2 + |\nabla v^{k+1}|^2\right)\nabla u^{k+1}\right) = 0\\ \Psi'\left(\left(I_z^{k+1}\right)^2\right)\cdot I_y^k I_z^{k+1} - \lambda \operatorname{div}\left(\Psi'\left(|\nabla u^{k+1}|^2 + |\nabla v^{k+1}|^2\right)\nabla v^{k+1}\right) = 0\end{aligned} \tag{7}$$

where I_x^k, I_y^k, I_z^k are defined according to eq. (6) by using the flow field components u and v from iteration k. Each time we reach a fixed-point, we advance to the next finer level and use the solution from the previous level as an initialisation. Since the above approach still yields a nonlinear system, we apply a first order Taylor expansion and approximate

$$I_z^{k+1} \approx I_z^k + I_x^k du^k + I_y^k dv^k \tag{8}$$

where $u^{k+1} = u^k + du^k$ with a known part u^k from coarse levels and an unknown increment du^k. The same holds, of course for v^{k+1}. For better readability let

$$(\Psi')_D^k = \Psi'\left(\left(I_z^k + I_x^k du^k + I_y^k dv^k\right)^2\right) \tag{9}$$

$$(\Psi')_S^k = \Psi'\left(|\nabla\left(u^k + du^k\right)|^2 + |\nabla\left(v^k + dv^k\right)|^2\right) \tag{10}$$

Then the first equation in (7) can be written as

$$(\Psi')_D^k \cdot \left(I_x^k\left(I_z^k + I_x^k du^k + I_y^k dv^k\right)\right) - \lambda \operatorname{div}\left((\Psi')_S^k \nabla\left(u^k + du^k\right)\right) = 0 \tag{11}$$

and the second equation can be reformulated in a similar way. Finally, by introducing a second fixed point strategy on du^k and dv^k we obtain a linear problem of the form

$$(\Psi')_D^{k,l} \cdot \left(I_x^k\left(I_z^k + I_x^k du^{k,l+1} + I_y^k dv^{k,l+1}\right)\right) - \lambda \operatorname{div}\left((\Psi')_S^{k,l} \nabla\left(u^k + du^{k,l+1}\right)\right) = 0 \tag{12}$$

for the first equation. A similar expression can again be derived for the second one. As initialisation, we will always use $du^{k,0} = dv^{k,0} = 0$. In the forthcoming experiments, this linear system will be solved using the SOR scheme. Thus, we basically have to perform two nested iterations. The *inner iteration* being the iterations of the SOR solver and the *outer iteration* refers to the fixed point approach on the index l.

3.2 Discrete Energy Optimisation

As described in the introduction, the strategy of the DEO approach is to first define a discrete version of our energy functional and then compute a minimiser via an optimisation method. More specifically, we discretise all the variables as well as the gradient operator and linearise the data-term in order to obtain a convex approximation. Analogously to the Euler-Lagrange approach of Section 3.1, we embed this in a coarse-to-fine warping framework to be able to deal with large displacements.

Let us assume that we have already computed the flow field (u^k, v^k) on the warping level k. We denote the grid on which u^k and v^k are defined by Ω_k. Then, (u^{k+1}, v^{k+1}) are computed on the next level as a minimiser of the functional

$$\underbrace{\sum_{(i,j)\in\Omega_k} |I_{z_{i,j}} + I_{x_{i,j}}(u_{i,j} - u^k_{i,j}) + I_{y_{i,j}}(v_{i,j} - v^k_{i,j})|}_{=:F(w)} + \lambda \underbrace{\sum_{(i,j)\in\Omega_k} \left\| \begin{pmatrix} \nabla u_{i,j} \\ \nabla v_{i,j} \end{pmatrix} \right\|}_{=:G(Dw)} \tag{13}$$

with $I_{x_{i,j}}$ (resp. $I_{y_{i,j}}$) evaluated at warped positions. As indicated in eq. (13), we use the abbreviated notation $F(w) + \lambda G(Dw)$ with $w := (u, v)$ and where the linear mapping D implements the gradient operators. Note that both F and G are convex functions. To minimise discrete energy functions which consist of the sum of convex terms, so-called *splitting based methods* have become popular in recent years in image processing and computer vision, cf. [15,21,22]. The main idea behind these methods is to treat the different terms of the energy function separately in each iteration and thus to decompose the problem into subproblems which are easy to solve. We follow [4] and use this algorithm to compute a minimiser of (13):

Algorithm (PDHG)
Initialisation: $w^{k+1,-1} := w^{k+1,0} := 0$, $p^0 := 0$
For $l = 0, \ldots, N-1$ *repeat*

$$\begin{aligned} &\text{Step 1:} \quad p^{l+1} = \operatorname*{argmin}_{p\in C} \left\{ \tfrac{1}{2} \left\| p - \left(p^l + \sigma D \left(2w^{k+1,l} - w^{k+1,l-1} \right) \right) \right\|^2 \right\} \\ &\text{Step 2:} \quad w^{k+1,l+1} = \operatorname*{argmin}_{w} \left\{ \tfrac{1}{2} \left\| w - \left(w^{k+1,l} - \tau D^{\mathrm{T}} p^{l+1} \right) \right\|^2 + \tau F(w) \right\} \end{aligned} \tag{14}$$

Output: $w^{k+1} := w^{k+1,N}$

The set C is defined as $C := \{p : \|p_{i,j}\| \leq \lambda,\ \forall (i,j) \in \Omega_k\}$. In [23], a similar algorithm which uses $w^{k+1,l}$ instead of the extrapolation $2w^{k+1,l} - w^{k+1,l-1}$ in the first minimisation problem of the above algorithm was proposed for image processing applications. We also refer to [4] for a detailed analysis. The authors of [23] call their method a primal-dual hybrid gradient (PDHG) algorithm and we also choose this terminology for the slightly different version used here. Observe that the PDHG algorithm was characterised as an inexact Uzawa method in [23,24], see also [25]. Furthermore, it corresponds to Algorithm 1 in [4] (with $\theta = 1$).

For the step length parameters σ and τ satisfying $\sigma\tau < 1/\|D\|^2$ and any initial values, the sequence $(w^{k+1,l})_l$ generated by the PDHG algorithm is guaranteed to converge to a minimiser of the energy functional in (13), c.f. [4,23].

Solving the two minimisation problems in each iteration of the PDHG algorithm can be done explicitly. Clearly, we can compute the orthogonal projection in the first step independently for each pixel via

$$\operatorname*{argmin}_{p\in C} \left\{ \tfrac{1}{2} \|p - \tilde{p}\|^2 \right\} = \left(\frac{\tilde{p}_{i,j}}{\max(1, \|\tilde{p}_{i,j}\|/\lambda)} \right)_{i,j} \tag{15}$$

Interestingly, the minimisation problem in the second step of the PDHG algorithm also decouples, and for any $\widetilde{w}$ the components of the minimiser

$$\hat{w} = \operatorname*{argmin}_{w} \left\{ \tfrac{1}{2}\|w - \widetilde{w}\|^2 + \tau F(w) \right\} \tag{16}$$

are given for each $(i,j) \in \Omega_k$ as follows. Let us define $a := \begin{pmatrix} I_{x_{i,j}} \\ I_{y_{i,j}} \end{pmatrix}$ and $\xi := I_{z_{i,j}} + (I_{x_{i,j}}\, I_{y_{i,j}})^{\mathrm{T}} \widetilde{w}_{i,j}$, then we have

$$\hat{w}_{i,j} = \widetilde{w}_{i,j} + \begin{cases} \tau a, & \text{if } \xi < -\tau\|a\|^2 \\ -\tau a, & \text{if } \xi > \tau\|a\|^2 \\ -\xi a/\|a\|^2, & \text{if } |\xi| \leqslant \tau\|a\|^2 \end{cases} \tag{17}$$

4 Experiments

We present a detailed numerical study making use of the *Rubberwhale* sequence from the popular Middlebury Computer Vision page[2]. The exact ground truth of this sequence is known and publicly available, thus allowing us to analyse the behaviour of the average endpoint error (AEE) as well as the discrete version of the energy of the flow field, c.f. (1). In accordance to the results of this detailed exposition, we summarise corresponding experiments for other sequences later on.

As for the *parameters of our algorithms*, we paid special interest on the influence of varying numbers of iterations. The regularisation parameter λ in the energy functional (1) was fixed throughout all the *Rubberwhale* tests at 5. For the *Yosemite* experiments it was set to 2 and during the *Marble* evaluation we used a value of 20. The EL framework used $\varepsilon = 10^{-3}$ in (3). We further chose the step length parameters $\sigma = 7.8$ and $\tau = 0.02$ in the PDHG algorithm.

The *warping pyramid* is always computed to the maximal possible extend. The scaling parameter is fixed at 0.95 in each image direction although a value of 0.5 also seems to be a common choice, see e.g. [15,16]. Let us emphasize that all the tests that we perform here can also be done for such a smaller scaling parameter. We observed the same behaviour for such setups as for the experiments reported below and include one example for the scaling parameter 0.5 at the end of Section 4.1. The main reason for us to use the scaling parameter 0.95 is that it seems to be frequently used for the EL approach, see [3,11]. Since we will perform comparative tests with this framework as well, it appeared more appropriate to us to choose 0.95.

In addition to the experiments performed with standard median filtering we also consider a reference method in order to evaluate the influence of the filter choice. We employ a structure adaptive filter as proposed in [16]. The authors of [16] suggested a filter that relies on the computation of weights over a given mask, whereas our reference filter adapts the mask itself. The idea of such a construction is that the point masks adapt locally to the variation of flow field values while taking into account the Euclidean

[2] `http://vision.middlebury.edu/flow/`

distance to the origin pixel for which it is set up. Large deviations in the flow field values are penalised, however, it may grow around corners or along strong structures in the flow. Finally, a median filter is applied on this resulting mask; see [20] for a detailed description and theoretical investigations.

The classic median filter we employ has a standard 5×5 square shaped mask and is applied twice at each flow field component between the warping steps. The adaptive median filter used a threshold of 0.65 for the tonal difference and a maximal length of 3. This yields a maximal mask size similar to the one for the classic median filter and allows us to give a fair comparison. The adaptive filter is also applied twice inbetween the different levels. Let us emphasize at this point, that none of the filters is applied anymore once we reach the finest resolution level.

With the above described setting we will analyse *(i)* the evolution of the AEE as well as the energy with and without additional filtering and *(ii)* the convergence behaviour of both algorithms for low and high numbers of iterations. In this context we will also briefly comment on the influence of varying numbers of inner and outer iterations in the EL approach.

4.1 The Impact of Intermediate Flow Field Filtering

In a first series of tests, we analyse the influence of an additional median or adaptive median filtering.

PDHG Algorithm. Our findings for the PDHG framework are presented in Table 1. For low numbers of iterations, e.g. 10-15, they correspond to the results described in [16]: applying a filter improves the endpoint error, however it also increases the energy. Interestingly, this phenomenon vanishes when we increase the number of iterations. For large numbers of optimisation steps, we observe that the effect of filtering on the error becomes negligible and that there is practically no difference in the energy between filtered and non filtered solutions. Also note that during the energy evolution, there are in practice always energy fluctuations that decrease in size with the number of optimisation steps. These cannot be captured by our tables.

Table 1. Rubberwhale experiment: Behaviour of the PDHG algorithm for different filters and numbers of iterations. For low numbers of iterations, additional filtering yields a decrease in the error and an increase in the energy. This effect decreases for higher numbers of iterations. Both the error as well as the energy value are not influenced by the filtering when the algorithm is close to convergence.

Iterations	None		Median		Struct. adapt.	
	Energy	Error	Energy	Error	Energy	Error
10	230705.0	0.2321	231755.0	0.2300	231664.5	0.2296
15	220609.0	0.1952	221089.5	0.1933	221056.5	0.1925
75	206292.5	0.1401	205922.0	0.1397	206109.0	0.1394
750	204376.0	0.1347	204316.5	0.1341	204572.0	0.1341

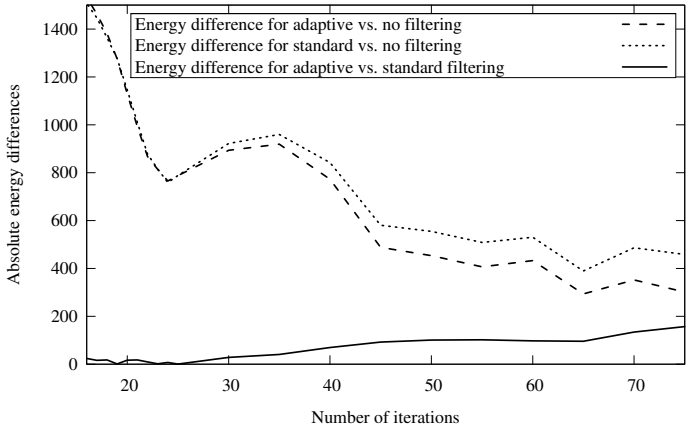

Fig. 1. Rubberwhale experiment: absolute energy differences for different filtering methods and the PDHG algorithm. The longer one iterates, the influence of filtering decreases.

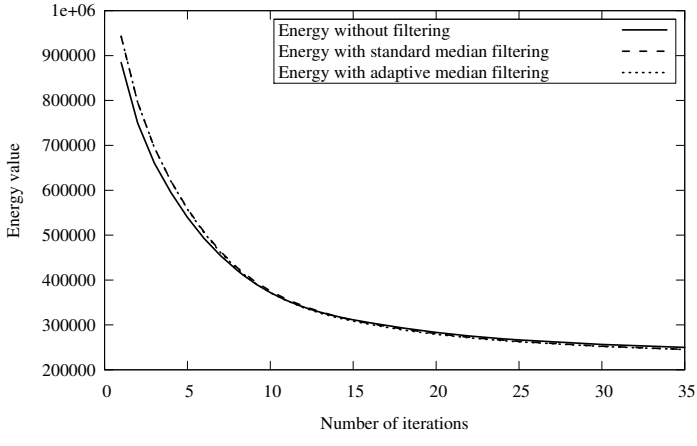

Fig. 2. Rubberwhale experiment: energy evolution of the PDHG algorithm with different filters and a scaling parameter of 0.5 for the coarse-to-fine strategy. Although a low number of iterations yields a higher energy when applying additional filtering, this effect vanishes after approximately 15 optimisation steps.

Figure 1 depicts the absolute value of the difference in the energy with and without additional filtering, giving us further detailed information of the numerical energy evolution. For low number of iterations the impact of filtering is strong in the beginning but slowly wears off as the number of iterations increases. Furthermore, the figure also depicts the difference of the energy between the standard and adaptive median filtering. Interestingly, this difference is rather small, suggesting that it might not be that important to consider adaptivity alone for improving filters. We supplement this study with

Table 2. Rubberwhale experiment: behaviour of the EL approach with a fixed number of 40 inner iterations with SOR for different filters and numbers of outer iterations. For low numbers of outer iterations, additional filtering yields a decrease in the error and an increase in the energy. This effect is no longer visible for high numbers of outer iterations (close to convergence). Both the error and the energy value are not influenced by the filtering if more than 200 outer iterations are applied.

Outer Iterations	None		Median		Struct. adapt.	
	Energy	Error	Energy	Error	Energy	Error
5	202828.5	0.1338	203216.0	0.1310	203218.5	0.1307
10	203328.0	0.1341	202768.5	0.1321	202822.5	0.1319
75	204245.5	0.1348	204187.0	0.1342	204427.0	0.1341
200	204342.0	0.1350	204285.5	0.1343	204529.5	0.1342

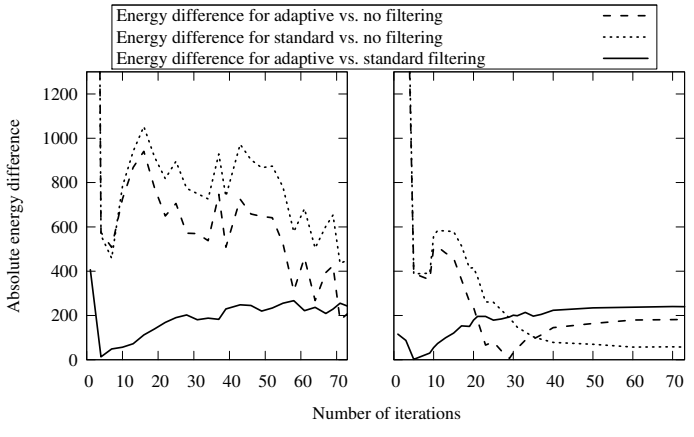

Fig. 3. Rubberwhale experiment: absolute energy difference with different filtering methods for 4 (left) and 40 (right) inner iterations with SOR. For small numbers of outer iterations the impact of filtering is again significantly larger than for high numbers of outer iterations. Note that the decrease is significantly faster with a high number of inner iterations. The y-axis uses the same scaling in both figures.

an experiment evaluating a warping scaling factor of 0.5. See Fig. 2. We observe the same qualitative behaviour as detailed via Table 1 and Figure 1 for the case 0.95. As the choice of small warping scale factors do not give more insight in the context of our study, we conclude the investigation of different scaling factors here with this example.

Euler Lagrange Approach. We conducted two test series. One where we applied 40 *inner iterations* with the SOR solver and another where we only applied 4. The former guarantees that the linear system inside the fixed-point iteration is always solved very accurately whereas the latter corresponds to more commonly used number of iterations.

Table 3. Rubberwhale experiment: comparison of the energy values for different filters and very high numbers of iterations. The number of iterations was chosen such that all methods have practically converged. The first number for the EL framework indicates the outer iterations, the second one corresponds to the iterations for the SOR solver. In the converged state there is no substantial difference between filtered and unfiltered solutions.

Algorithm	None		Median		Struct. adapt.	
	Energy	Error	Energy	Error	Energy	Error
PDHG (800 it.)	204958.0	0.1357	204302.0	0.1340	204566.0	0.1341
EL (75/04 it.)	204856.3	0.1355	204202.3	0.1341	204432.5	0.1340
EL (75/07 it.)	204916.5	0.1362	204197.0	0.1340	204436.5	0.1339
EL (75/40 it.)	204245.5	0.1348	204187.0	0.1342	204427.0	0.1341

Table 4. The Yosemite test: the comparison of DEO and EL frameworks confirming the results of our detailed investigation with the Rubberwhale sequence. We observe a decreasing influence of the filter in the energy, see especially the EL results. While the filter results show some benefit here, note that the reported iterates have not yet fully converged.

PDHG					
Iterations	Endpoint Error			Absolute energy difference	
	None	Median	Str. adapt.	None/Median	None/Str. adapt.
10	0.2577	0.2309	0.1958	1250.4	1014.4
15	0.2194	0.1997	0.1729	1035.0	1022.0
75	0.1985	0.1899	0.1735	200.6	205.4
200	0.2022	0.1916	0.1760	278.4	270.8
EL with 4 inner SOR steps					
Iterations	Endpoint Error			Absolute energy difference	
	None	Median	Str. adapt.	None/Median	None/Str. adapt.
5	0.1738	0.1601	0.1597	938.0	905.4
15	0.1772	0.1734	0.1727	149.4	144.6
75	0.1887	0.1764	0.1775	154.2	152.2
125	0.1842	0.1786	0.1781	79.0	73.4

Fig. 3 depicts the influence of the filtering for the just described setting. Using 40 inner iterations with SOR, we obtain the same behaviour as for the PDHG algorithm concerning the fixed-point iterations. If we use only 4 inner iterations, a decrease of the filtering influence is still there for higher numbers of outer iterations. However, it is much less pronounced. This may be due to the fact that the linear system inside the EL equations is not solved accurately enough, possibly involving a numerical blurring effect. All in all, we observe a similar behaviour in the EL setting as for the PDHG formulation.

Table 5. The Marble test: We observe no effect on the AEE when using a filter. While we observe some energy fluctuations in this example, comparing the energy differences after 10 and 200 steps for the PDHG approach clearly shows the expected decrease.

PDHG					
Iterations	Endpoint Error			Absolute energy difference	
	None	Median	Str. adapt.	None/Median	None/Str. adapt.
10	0.1787	0.1778	0.1775	4710	4764
15	0.1713	0.1709	0.1709	2372	2432
75	0.1575	0.1574	0.1576	1232	1232
200	0.1582	0.1579	0.1581	1478	1458
EL with 4 inner SOR steps					
Iterations	Endpoint Error			Absolute energy difference	
	None	Median	Str. adapt.	None/Median	None/Str. adapt.
5	0.1565	0.1565	0.1565	2362	2362
15	0.1575	0.1574	0.1574	818	818
75	0.1589	0.1589	0.1589	1372	1376
125	0.1590	0.1590	0.1590	1466	1466

4.2 Euler-Lagrange and Splitting Methods Close to Convergence

Table 3 depicts the energy value of the considered algorithms for very high numbers of iterations. We applied 800 steps with the PDHG algorithm and 75 outer iterations as well as a varying number of inner SOR steps within the EL approach. At this point all considered methods have practically reached convergence.

Two important things become immediately apparent. The differences in the energy do not vary significantly, whether we apply a filter or not. The smoothing effect of the filtering even results in this example in a slightly lower energy than without filtering, with negligible consequences on the AEE. Finally, both frameworks yield very similar energy values.

4.3 Results for Further Test Sequences

Because of space restrictions, we have selected just two more image sequences, namely the *Marble*[3] and the *Yosemite*[4] sequences. For other sequences the results are similar.

5 Conclusions

In our paper, we have clarified the mechanism behind the filtering of flow fields during warping. We think that by our investigations the effect of this technique is now well-understood. In this, we have complemented and extended the previous work [16].

[3] Available from `http://i21www.ira.uka.de/image_sequences/`
[4] Available from `http://vision.middlebury.edu/flow/data/`

Furthermore, we have closed an important methodical gap between the two main algorithmic approaches in modern optic flow computation. Since most papers in the field of optic flow employ just one of these techniques, we also hope to improve by the current work the mutual understanding of researchers following mainly one of the paths. In our future work, we strive for other deeper insights into numerical schemes in computer vision.

References

1. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artificial Intelligence 17, 185–203 (1981)
2. Black, M.J., Anandan, P.: The robust estimation of multiple motions: parametric and piecewise smooth flow fields. Computer Vision and Image Understanding 63, 75–104 (1996)
3. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
4. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. Journal of Mathematical Imaging and Vision 40, 120–145 (2011)
5. Mémin, E., Pérez, P.: Hierarchical estimation and segementation of dense motion fields. International Journal of Computer Vision 46, 129–155 (2002)
6. Nir, T., Bruckstein, A.M., Kimmel, R.: Over-parametrized variational optical flow. International Journal of Computer Vision 76, 205–216 (2008)
7. Wedel, A., Cremers, D., Pock, T., Bischof, H.: Structure- and motion-adaptive regularization for high accuracy optic flow. In: Proc. 2009 IEEE International Conference on Computer Vision. IEEE Computer Society Press, Kyoto (2009)
8. Werlberger, M., Pock, T., Bischof, H.: Motion estimation with non-local total variation regularization. In: Proc. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Press, San Francisco (2010)
9. Weickert, J., Schnörr, C.: A theoretical framework for convex regularizers in PDE-based computation of image motion. International Journal of Computer Vision 45, 245–264 (2001)
10. Xu, L., Jia, J., Matsushita, Y.: Motion detail preserving optical flow estimation. In: Proc. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Press, San Francisco (2010)
11. Zimmer, H., Bruhn, A., Weickert, J., Valgaerts, L., Salgado, A., Rosenhahn, B., Seidel, H.-P.: Complementary optic flow. In: Cremers, D., Boykov, Y., Blake, A., Schmidt, F.R. (eds.) EMMCVPR 2009. LNCS, vol. 5681, pp. 207–220. Springer, Heidelberg (2009)
12. Goldluecke, B., Cremers, D.: Convex relaxation for multilabel problems with product label spaces. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6315, pp. 225–238. Springer, Heidelberg (2010)
13. Bruhn, A., Weickert, J., Kohlberger, T., Schnörr, C.: A multigrid platform for real-time motion computation with discontinuity-preserving variational methods. International Journal of Computer Vision 70, 257–277 (2006)
14. Gwosdek, P., Zimmer, H., Grewenig, S., Bruhn, A., Weickert, J.: A highly efficient GPU implementation for variational optic flow based on the Euler-Lagrange framework. In: Proc. 2010 ECCV Workshop on Computer Vision with GPUs, Heraklion, Greece (September 2010) accepted
15. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime TV- L1 optical flow. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) DAGM 2007. LNCS, vol. 4713, pp. 214–223. Springer, Heidelberg (2007)

16. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2432–2439. IEEE, Los Alamitos (2010)
17. Wedel, A., Pock, T., Zach, C., Bischof, H., Cremers, D.: An improved algorithm for TV-L^1 optical flow. In: Cremers, D., Rosenhahn, B., Yuille, A.L., Schmidt, F.R. (eds.) Statistical and Geometrical Approaches to Visual Motion Analysis. LNCS, vol. 5604, pp. 23–45. Springer, Heidelberg (2009)
18. Zimmer, H., Bruhn, A., Weickert, J.: Optic flow in harmony (2011) to appear in International Journal of Computer Vision
19. Saad, Y.: Iterative methods for sparse linear systems, 2nd edn. SIAM, Philadelphia (2000)
20. Welk, M., Breuß, M., Vogel, O.: Morphological amoebas are self-snakes. Journal of Mathematical Imaging and Vision 39, 87–99 (2011)
21. Wang, Y., Yang, J., Yin, W., Zhang, Y.: A new alternating minimization algorithm for total variation image reconstruction. SIAM Journal on Imaging Sciences 1, 248–272 (2008)
22. Goldstein, T., Osher, S.: The split bregman method for L^1-regularized problems. SIAM Journal on Imaging Sciences 2, 323–343 (2009)
23. Zhang, X., Burger, M., Osher, S.: A unified primal-dual algorithm framework based on Bregman iteration. Journal of Scientific Computing 46, 20–46 (2011)
24. Esser, E., Zhang, X., Chan, T.F.: A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. SIAM Journal on Imaging Sciences 3, 1015–1046 (2010)
25. Arrow, K.J., Hurwicz, L., Uzawa, H.: Studies in linear and non-linear programming. Stanford University Press, Stanford (1958)