Motion Compensated Frame Interpolation with a Symmetric Optical Flow Constraint

Lars Lau Rakêt[†], Lars Roholm[†], Andrés Bruhn[‡] and Joachim Weickert[⋆]

[†]Department of Computer Science, University of Copenhagen
Universitetsparken 5, 2100 Copenhagen, Denmark
[‡]Institute for Visualization and Interactive Systems, University of Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany
[⋆]Mathematical Image Analysis Group, Saarland University
Campus E 1.7, 66123 Saarbrücken, Germany

Abstract. We consider the problem of interpolating frames in an image sequence. For this purpose accurate motion estimation can be very helpful. We propose to move the motion estimation from the surrounding frames directly to the unknown frame by parametrizing the optical flow objective function such that the interpolation assumption is directly modeled. This reparametrization is a powerful trick that results in a number of appealing properties, in particular the motion estimation becomes more robust to noise and large displacements, and the computational workload is more than halved compared to usual bidirectional methods. The proposed reparametrization is generic and can be applied to almost every existing algorithm. In this paper we illustrate its advantages by considering the classic TV-L^1 optical flow algorithm as a prototype. We demonstrate that this widely used method can produce results that are competitive with current state-of-the-art methods. Finally we show that the scheme can be implemented on graphics hardware such that it becomes possible to double the frame rate of 640×480 video footage at 30 fps, i.e. to perform frame doubling in realtime.

1 Introduction

Frame interpolation is the process of creating intermediate images in a sequences of known images. The process has many uses, for example video post-processing and restoration, temporal upsampling in HDTVs to enhance viewing experience as well as a number of more technical applications, e.g. in video coding.

In this work we consider optical flow based frame rate upsampling which performs interpolation along the motion trajectories. With this application in mind we propose to reparametrize the optical flow energy such that it fits better to the given problem. The reparametrized energy has a symmetric data fidelity term, that uses both surrounding frames as references. We show that one can improve modern frame interpolation methods substantially by this powerful generic trick, that can be incorporated in existing schemes without requiring major adaptations. We analyze the reparametrization, and show experimentally that it has a great effect on the stability and robustness of the interpolation process.

The idea to symmetrize data matching terms to achieve better results has already established its usefulness in other areas. In image registration Christensen and Johnson [1] explored the benefit of penalizing consistency, by jointly estimating forward and backward transforms, and requiring that they were inverses of one another. A similar idea was applied to the optical flow problem by Alvarez et al. [2], who imposed an additional consistency term. Later that same year Alvarez et al. [3] proposed a reparametrization similar to the one derived in this paper in order to avoid a reference frame, and thereby increase flow consistency. However, they did not use the obtained symmetric flow directly, but interpolated flow values at pixel position of a reference image in order to obtain a flow comparable to the standard asymmetric flow. Recently Chen [4] used a symmetric data term for surface velocity estimation, noting the property that motion vector length is halved, which in turn gives better handling of large displacements.

Apart from being algorithmically different, the difference between the justification given in this paper and the justifications of Alvarez et al. [3] and Chen [4] is that we have chosen the symmetric data fidelity term because it explicitly models the standard interpolation assumption, rather than improves some notion of consistency or better handles large displacements. In turn this also means that we use the estimated flows directly on the unknown frame, and thereby avoid problems of temporal warping. As we will show, the mentioned benefits are clearly reflected in the results. They demonstrate that using a symmetric flow for interpolation is generally better than using either forward or backward flows or both.

The rest of this paper is organized as follows. In the next section we review the estimation process for duality based TV-L^1 optical flow. In Section 3 we discuss a standard method for motion compensated frame interpolation, and in Section 4 we present our reparametrization of the optical flow energy. In Section 5 we consider examples and compare to current state-of-the-art methods, and finally we conclude the paper with discussion and outline future directions in Section 6.

2 Duality Based TV-L^1 Optical Flow

Optical flow estimation concerns the determination of apparent (projected) motion. Given a sequence of temporally indexed images I_t, we want to estimate the optical flow $\boldsymbol{v}$ such that the motion matches the image sequence with respect to some measure. This is often done by computing the flow as the minimizer of an energy of the type

$$E(\boldsymbol{v}) = \lambda F(I, \boldsymbol{v}) + R(\boldsymbol{v}) \tag{1}$$

where F is a positive functional measuring data fidelity and R is a regularization term. Many energies of this type have been suggested throughout the years, and a large variety of resolution strategies exist. Here we will focus on the TV-L^1

energy, where data fidelity between two frames I_0 and I_1 is measured by the L^1-norm of the difference

$$F(I_0, I_1, \boldsymbol{v}) = \int \|I_1(\boldsymbol{x} + \boldsymbol{v}(\boldsymbol{x})) - I_0(\boldsymbol{x})\| \,\mathrm{d}\boldsymbol{x}, \tag{2}$$

and the regularization term R penalize the total variation of the estimated motion

$$R(\boldsymbol{v}) = \int \|\mathscr{D}\boldsymbol{v}(\boldsymbol{x})\| \,\mathrm{d}\boldsymbol{x}, \tag{3}$$

which, depending on the definition of the operator $\mathscr{D}$ can give different forms of the vectorial total variation [5]. Here we will take $\mathscr{D}$ to be the 1-Jacobian of Goldluecke et al. [6], since this choice of regularizer does not suffer from the channel-smearing of usual definition of vectorial total variation [5]. In order to efficiently minimize E we introduce two relaxations. First we linearize the data fidelity term $I_1(\cdot + \boldsymbol{v}) - I_0 \approx \rho(\boldsymbol{v})$, where $\cdot$ is a placeholder for the argument of the function (i.e. $\boldsymbol{x}$),

$$\rho(\boldsymbol{v}) = I_1(\cdot + \boldsymbol{v}_0) - I_0 + \boldsymbol{J}_{I_1}(\cdot + \boldsymbol{v}_0)(\boldsymbol{v} - \boldsymbol{v}_0) \tag{4}$$

where $\boldsymbol{J}_{I_1}$ is the Jacobian of I_1, and $\boldsymbol{v}_0$ is the current estimate of $\boldsymbol{v}$. We further relax E by introducing an auxiliary variable $\boldsymbol{u}$ that splits data fidelity and regularization in two quadratically coupled energies:

$$E_1(\boldsymbol{v}) = \int \lambda\|\rho(\boldsymbol{v})(\boldsymbol{x})\| + \frac{1}{2\theta}\|\boldsymbol{v}(\boldsymbol{x}) - \boldsymbol{u}(\boldsymbol{x})\|^2 \,\mathrm{d}\boldsymbol{x}, \tag{5}$$

$$E_2(\boldsymbol{u}) = \int \frac{1}{2\theta}\|\boldsymbol{v}(\boldsymbol{x}) - \boldsymbol{u}(\boldsymbol{x})\|^2 + \|\mathscr{D}\boldsymbol{u}(\boldsymbol{x})\| \,\mathrm{d}\boldsymbol{x}. \tag{6}$$

This relaxation was first proposed by Zach et al. [7], and has a number of advantages, most notably that the first problem can be solved pointwise which makes the solution very easy to implement on massively parallel processors like graphics processing units (GPUs). For completeness we will give the minimizing pointwise solution to (5) in the general case where $\rho(\boldsymbol{v})(\boldsymbol{x}) = \boldsymbol{a}^\top \boldsymbol{v} + b$, $\boldsymbol{a} \in \mathbb{R}^d$ and $b \in \mathbb{R}$, which is given as

$$\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{u}(\boldsymbol{x}) - \pi_{\lambda\theta[-\boldsymbol{a},\boldsymbol{a}]}\left(\boldsymbol{u} + \frac{b}{\|\boldsymbol{a}\|^2}\boldsymbol{a}\right) \tag{7}$$

where $\pi_{\lambda\theta[-\boldsymbol{a},\boldsymbol{a}]}$ is the projection onto the line segment joining the vectors $-\lambda\theta\boldsymbol{a}$ and $\lambda\theta\boldsymbol{a}$, which is given by

$$\pi_{\lambda\theta[-\boldsymbol{a},\boldsymbol{a}]}\left(\boldsymbol{u} + \frac{b}{\|\boldsymbol{a}\|^2}\boldsymbol{a}\right) = \begin{cases} -\lambda\theta\boldsymbol{a} & \text{if} \quad \boldsymbol{a}^\top\boldsymbol{u} + b < -\lambda\theta\|\boldsymbol{a}\|^2 \\ \lambda\theta\boldsymbol{a} & \text{if} \quad \boldsymbol{a}^\top\boldsymbol{u} + b > \lambda\theta\|\boldsymbol{a}\|^2 \\ \frac{\boldsymbol{a}^\top\boldsymbol{u}+b}{\|\boldsymbol{a}\|^2}\boldsymbol{a} & \text{if} \quad |\boldsymbol{a}^\top\boldsymbol{u} + b| \leq \lambda\|\boldsymbol{a}\|^2 \end{cases} . \tag{8}$$

For $\boldsymbol{a} = \nabla I_1(\boldsymbol{x}+\boldsymbol{v}_0)$ and $b = I_1(\boldsymbol{x}+\boldsymbol{v}_0) - I_0(\boldsymbol{x}) - \nabla I_1(\boldsymbol{x}+\boldsymbol{v}_0)^\top \boldsymbol{v}_0$ the above expression reduces to the result of Zach et al. [7]. In the general case of vector valued images, (5) can be minimized by the method presented in [8]. We will not replicate the minimizer of the regularization energy (6) here, but note that it can be minimized effectively following an iterative pointwise Bermùdez-Moreno type algorithm (see Goldluecke et al. [6]). For further implementation details we refer to Section 5.

3 Motion Compensated Frame Interpolation

Given two images I_0 and I_1 and an estimate of the (forward) optical flow $\boldsymbol{v}_f$ we are interested in estimating the in-between image $I_{1/2}$ (the methods are easily extended to any in-between frame I_t, $t \in (0,1)$). A simple approach is to assume that the motion vectors are linear through $I_{1/2}$ and then fill in $I_{1/2}$ using the computed flow. However, since $\boldsymbol{v}_f$ is of sub-pixel accuarcy, the points $\boldsymbol{x}+1/2\boldsymbol{v}_f(\boldsymbol{x})$ that are hit by the motion vectors are generally not pixel positions. This is often solved by warping the flow to the temporal position of the intermediate frame $I_{1/2}$ (see e.g. [9], [10], [11]), in which one defines a new flow $\boldsymbol{v}_f^{1/2}$ from $I_{1/2}$ to I_1

$$\boldsymbol{v}_f^{1/2}(\mathrm{round}(\boldsymbol{x} + 1/2\boldsymbol{v}_f(\boldsymbol{x}))) = 1/2\boldsymbol{v}_f(\boldsymbol{x}), \tag{9}$$

where the round function rounds the argument to nearest pixel value in the domain. There are some drawbacks to this approach. First, if the area around $\boldsymbol{x}$ in I_0 is occluded in I_1, there are likely multiple flow candidates assigned to the point $\mathrm{round}(\boldsymbol{x} + 1/2\boldsymbol{v}_f(\boldsymbol{x}))$. In the converse situation, i.e. dis-occlusion from I_0 to I_1 there may be pixels that are not hit by a flow vector, thus leaving holes in the flow.

While the first problem can be solved by choosing the candidate vector with the best data fit, the solution for the problem of dis-occlusions in not that simple. Here we will simply fill the holes in the flow field by an outside-in filling strategy. With a dense flow we can then interpolate $I_{1/2}$ using the *forward* scheme

$$I_{1/2}(\boldsymbol{x}) = \frac{1}{2}\left(I_0(\boldsymbol{x} - \boldsymbol{v}_f^{1/2}(\boldsymbol{x})) + I_1(\boldsymbol{x} + \boldsymbol{v}_f^{1/2}(\boldsymbol{x}))\right), \tag{10}$$

or consider the backward flow $\boldsymbol{v}_b$ (i.e. the flow from I_1 to I_0) and use a backward scheme accordingly. We will in addition consider a *bidirectional* interpolation scheme where the frame is interpolated as the average frames obtained by the forward and backward schemes.

One can sophisticate the interpolation methods by estimating occluded regions and selectively interpolating from the correct frame. We will not pursue any occlusion reasoning here, but refer to [10] and [12] for details.

4 Reparametrizing Optical Flow for Interpolation

The approach presented in the previous section is the standard procedure for frame interpolation and serves as backbone in many algorithms ([9], [13], [14],

[11]). In this section we will reparametrize the original energy functional so the recovered flow is better suited for interpolation purposes. The reparametrization turns out to be beneficial on a number of levels: It makes the temporal warping of the flow superfluous, eliminates the need to calculate flows in both directions, improves handling of large motion, and increase overall robustness.

The original optical flow energy functional take as argument an optical flow $\boldsymbol{v}$ that is defined on a continuous domain. In practice, however, we only observe images at discrete pixels, and the optical flow is typically only estimated at the points corresponding to the pixels in I_0. Since we assume that the intermediate frame $I_{1/2}$ can be obtained from linearly following the flow vectors, we propose to reparametrize the data fidelity functional F using this assumption, so that it is given as

$$\frac{1}{2}\int \|I_1(\boldsymbol{x}+\boldsymbol{v}_s(\boldsymbol{x})) - I_0(\boldsymbol{x}-\boldsymbol{v}_s(\boldsymbol{x}))\| \,\mathrm{d}\boldsymbol{x}. \tag{11}$$

We note that in this parametrization, the coordinates of the optical flow matches those of the intermediate frame $I_{1/2}$, and using this data term will thus eliminate the need for warping of the flow, since interpolation can directly be done similarly to (10). Because the motion vectors of the symmetric flow $\boldsymbol{v}_s$ are only half of the ones of e.g. the forward flow $\boldsymbol{v}_f$, we need to halve the corresponding λ to keep comparison fair, which is the reason for the factor $1/2$.

Linearizing the data matching term (11) around $\boldsymbol{v}_0$ gives

$$\rho(\boldsymbol{v}_s) = I_1(\cdot+\boldsymbol{v}_0) - I_0(\cdot-\boldsymbol{v}_0) + (\boldsymbol{J}_{I_1}(\cdot+\boldsymbol{v}_0) + \boldsymbol{J}_{I_0}(\cdot-\boldsymbol{v}_0))(\boldsymbol{v}_s-\boldsymbol{v}_0) \tag{12}$$

which is similar to (4). In 1D the corresponding split energy term (5) is easily minimized using (7), and in general using the L^1-L^2 minimization from [8].

The differences between (12) and (4) are that we now allow sub-pixel matching in both surrounding images, and instead of a single Jacobian we have a sum of two. Thinking of this linearization as a finite difference scheme corresponding to a linearized differential form of the data fidelity term, we see that the temporal derivative is represented by a central finite difference scheme, as opposed to the typical forward differences (4). In addition the sum of the two Jacobians should make the estimation procedure more robust to noise, as the noise amplification caused by derivative estimation is now averaged over two frames. This has previously been used heuristically to improve accuracy in asymmetric flow estimation (see e.g. [15]). Finally we note that the motion vectors will only have half the length of the ones obtained from the regular parametrization. This will make the method better suited to handle large displacements compared to traditional methods that only make use of a one-sided linearization.

5 Results

Motion compensated frame interpolation finds many uses, ranging from the more technical applications such as video coding [13] to disciplines like improving viewing experience [14] or restoration of historic material [11]. For the former type of application the reconstruction quality in terms of quantitative measures is of great importance. For the latter types it is hard to devise specific measures of quality, as the human visual system is very tolerant to some types of errors, while it is unforgiving to other types of errors. In fact, the types of tolerated errors may even depend on the specific sequence.

For the results presented in the following we use a setup where we solve (5) and (6) iteratively in a coarse-to-fine pyramid as illustrated in Algorithm 1. We use $\ell_{\max} = 70$ levels and a scale factor of 0.95. On each level $w_{\max} = 60$ warps are performed, and within each warp we minimize (5) with linearized data fidelity term (12) using (7), followed by minimization of (6) using using $i_{\max} = 5$ inner iterations of a Bermùdez-Moreno type algorithm [6]. We fix the coupling parameter $\theta = 0.2$.

Algorithm 1: Computation of TV-L^1 optical flow.

```
Data: Two images I_0 and I_1
Result: Symmetric optical flow field v_s
for ℓ = ℓ_max to 0 do
    // Pyramid levels
    Downsample the images I_0 and I_1 to current pyramid level
    for w = 0 to w_max do
        // Warping
        Compute v_s pointwise as the minimizer of E_1 (5) with data fidelity (12)
        for i = 0 to i_max do
            // Inner iterations
            Compute u_s as the minimizer of E_2 (using methods presented in [6])
        end
        Upscale v and u to next pyramid level
    end
end
```

As our first experiment we compare the four different types of interpolation suggested in the previous sections, on the four *High-speed camera* training sequences of the Middlebury Optical Flow benchmark. Figure 1 shows the effect of varying the data term weight λ in terms of the mean absolute interpolation error (MAIE). We see that the symmetric flow outperforms the conventional approaches, and that it is typically less sensitive in terms of the choice of λ. In particular we see that the difficult Beanbags sequence which contains large displacements is handled much better by the symmetric scheme. By evaluation on the Middlebury training set it was found that $\lambda = 35$ gave the best overall performance for the symmetric flow, and that $\lambda = 20$ gave the best performance for the other three methods. These λ values will be used in the rest of the experiments presented in this section.

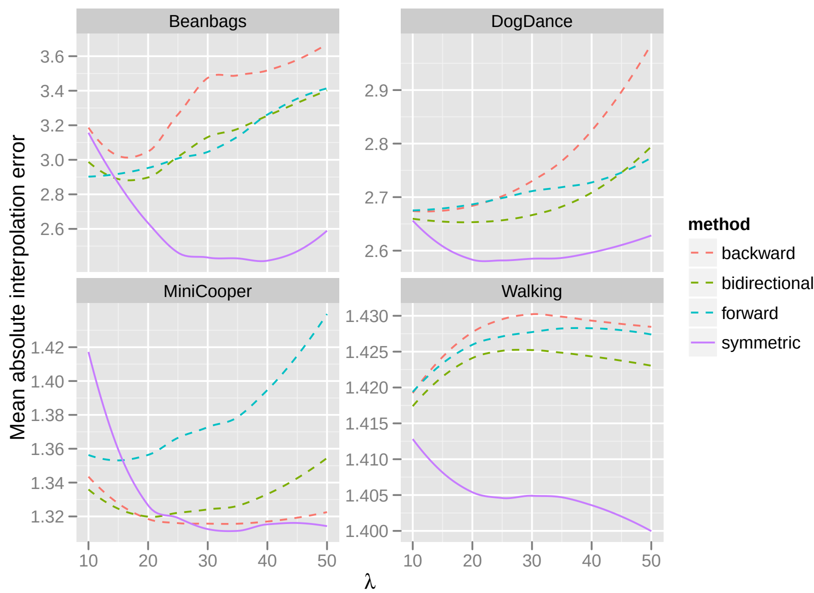


Fig. 1: Performance for varying λ on the four *High-speed camera* training sequences from the Middlebury Optical Flow benchmark [9].

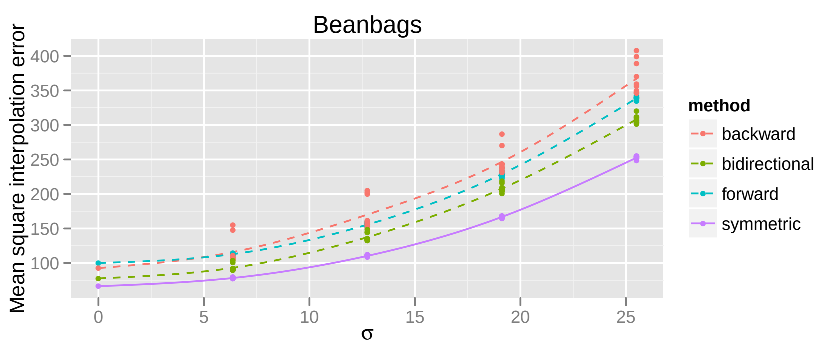


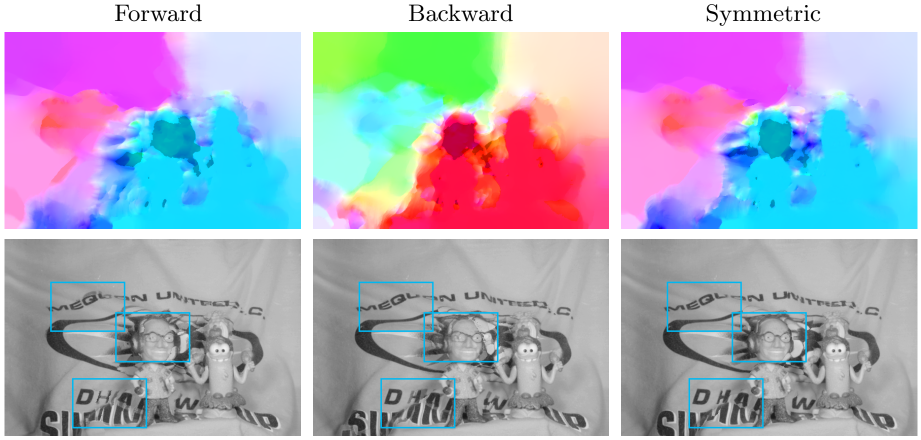
Fig. 2: Mean square interpolation error performance under additive $\mathcal{N}(0, \sigma^2)$ noise for varying σ. Results are for the Beanbags sequence, and are based on 10 independent replications.

For our second example we consider the results of interpolation under noise. Figure 2 shows the mean square interpolation error performance of the four methods on the Beanbags sequence with additive $\mathcal{N}(0, \sigma^2)$ noise. The improved robustness of the symmetric interpolation method is clearly visible from the distances between the MAIEs to the asymmetric methods that increase as the standard deviation of the noise increases. In addition we see that the variance of the MAIEs across the independent replications is significantly lower for the symmetric method compared to the three other methods.

Fig. 3: Frames 7 (I_0), 10 ($I_{1/2}$) and 13 (I_1) of the Mequon sequence.

As our third example, consider the frames given in Figure 3. The sequence has large displacements (> 35 pixels) and severe deformations, which makes the estimation of $I_{1/2}$ very difficult. Figure 4 shows the three different flows $\boldsymbol{v}_f$, $\boldsymbol{v}_b$ and $\boldsymbol{v}_s$ along with the corresponding interpolated frames. Zoom ins of details can be found in Figure 5. We see that the result generated by the symmetric flow is visually more pleasing than the ones produced by the forward and backward flows, a fact that is also clearly reflected in the MAIEs and root mean square interpolation errors (RMSIE).

Forward | Backward | Symmetric



MAIE 8.72, RMSIE 20.30 | MAIE 8.85, RMSIE 20.09 | MAIE 8.23, RMSIE 19.02

Fig. 4: Results for the Mequon sequence. Top row: Color coded optical flows, buttom row: Interpolation results. Zoom ins of details can be found in Figure 5.

Finally let us compare the method to some methods of the current state-of-the-art. Table 1 holds the RMSIEs for six sequences from the Middlebury

Fig. 5: Details of the interpolated Mequon frame from Figure 4.

Table 1: RMSIE for different Middlebury sequences. Bold indicates the best result. † Results are taken from [12]. ‡ Marked algorithms have not been implemented by their respective authors, but by the Middlebury authors [9].

Method	Dimetrodon	Venus	Hydrangea	RubberWhale	MiniCooper	Walking
Symmetric TV-L^1	1.93	3.45	3.36	**1.46**	**3.96**	**2.89**
Chen and Lorenz [16]	1.95	3.63	—	—	—	—
Werlberger et al. [11]	1.93	—	—	—	4.55	3.97
Stich et al. [12]	**1.78**	**2.88**	**2.57**	1.59	—	—
Bruhn et al. [17]†,‡	2.59	3.73	—	—	—	—
Pyramid Lukas-Kanade†,‡	2.49	3.67	—	—	—	—

Optical Flow benchmark and results for a number of methods. While the results cannot fully match the results of Stich et al. [12], which gives significantly better results on 3 of the sequences, our method outperforms all other approaches, including the recent and much more complex methods of Chen and Lorenz [16] and Werlberger et al. [11].

Real-time Performance

In the presented setup we only have to compute a single flow field between two images and fill in the intermediate frame from the trajectories. The runtime of the interpolation is dominated by the time it takes to compute the flow field, and at a slight cost in accuracy (5 pyramid levels with a scale factor of 2, and 30 warps per level, 1 level of median filtering) the flow fields can be computed in real-time (∼35 fps) for 640×480 images using an NVIDIA Tesla C2050 GPU, which in turn means that we can do real-time frame doubling of 30fps video footage at a resolution of 640×480.

6 Conclusion and Outlook

We have proposed a method for motion compensated frame interpolation that is based on interpolation along the motion vectors. The main contribution is to use the assumption that an in-between frame can be reached by linearly following the motion vectors for reparametrizing the optical flow energy, such that the coordinate system of the flow matches that of the in-between frame. We have showed that one can improve frame interpolation methods substantially using this powerful and generic symmetric parametrization, and that the parametrization can be incorporated in existing frame interpolation methods with only little adaptation. Using a simple TV-L^1 optical flow algorithm as prototype we demonstrated results that are competitive with recent methods that are highly sophisticated. The proposed method can be implemented very efficiently, and using graphics hardware we succeeded in doubling the frame rate of 640×480 video in real time.

The presented work can be extended in a number of directions. The most obvious extension would be to use the symmetric data term with a more advanced optical flow method. If the goal is to improve viewing experience, a spatial regularization of the interpolated frames could probably improve the perceived quality. Spatial regularization could be done by means of total variation (see e.g. [14] and [11]) or by edge enhancing diffusion [18]. The latter has been shown to have very good interpolation properties, and has been successfully used in image compression [19] and for motion compensated deinterlacing [20]. To improve reconstruction quality in terms of e.g. RMSIE, one could do occlusion reasoning and selectively interpolate from the non-occluded frame, or compute motion trajectories over several frames [21] and use this information for interpolation.

References

1. Christensen, G.E., Johnson, H.J.: Consistent image registration. IEEE Transactions on Medical Imaging **20** (2001) 568–582
2. Alvarez, L., Deriche, R., Papadopoulo, T., Sánchez, J.: Symmetrical dense optical flow estimation with occlusions detection. International Journal of Computer Vision **75** (2007) 371–385
3. Alvarez, L., Castaño, C., García, M., Krissian, K., Mazorra, L., Salgado, A., Sánchez, J.: Symmetric optical flow. In Díaz, R., Pichler, F., Arencibia, A., eds.: Computer Aided Systems Theory–EUROCAST 2007. Volume 4739 of Lecture Notes in Computer Science. Springer (2007) 676–683
4. Chen, W.: Surface velocity estimation from satellite imagery using displaced frame central difference equation. To appear in IEEE Transactions on Geoscience and Remote Sensing, (2012)
5. Bresson, X., Chan, T.: Fast dual minimization of the vectorial total variation norm and application to color image processing. Inverse Problems and Imaging **2** (2008) 455–484
6. Goldluecke, B., Strekalovskiy, E., Cremers, D.: The natural total variation which arises from geometric measure theory. SIAM Journal on Imaging Sciences **5** (2012) 537–563

7. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime TV-L^1 optical flow. In Hamprecht, F., Schnörr, C., Jähne, B., eds.: Pattern Recognition. Volume 4713 of Lecture Notes in Computer Science. Springer (2007) 214–223
8. Rakêt, L.L., Roholm, L., Nielsen, M., Lauze, F.: TV-L^1 optical flow for vector valued images. In Boykov, Y., Kahl, F., Lempitsky, V., Schmidt, F., eds.: Energy Minimization Methods in Computer Vision and Pattern Recognition. Volume 6819 of Lecture Notes in Computer Science. Springer (2011) 329–343
9. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. International Journal of Computer Vision **31** (2011) 1–31
10. Herbst, E., Seitz, S., Baker, S.: Occlusion reasoning for temporal interpolation using optical flow. Technical Report UW-CSE-09-08-01, Department of Computer Science and Engineering, University of Washington (2009)
11. Werlberger, M., Pock, T., Unger, M., Bischof, H.: Optical flow guided TV-L^1 video interpolation and restoration. In Boykov, Y., Kahl, F., Lempitsky, V., Schmidt, F., eds.: Energy Minimization Methods in Computer Vision and Pattern Recognition. Volume 6819 of Lecture Notes in Computer Science. Springer (2011) 273–286
12. Stich, T., Linz, C., Albuquerque, G., Magnor, M.: View and time interpolation in image space. Computer Graphics Forum **27** (2008) 1781–1787
13. Huang, X., Rakêt, L.L., Luong, H.V., Nielsen, M., Lauze, F., Forchhammer, S.: Multi-hypothesis transform domain Wyner-Ziv video coding including optical flow. In: Multimedia Signal Processing. (2011)
14. Keller, S., Lauze, F., Nielsen, M.: Temporal super resolution using variational methods. In Mrak, M., Grgic, M., Kunt, M., eds.: High-Quality Visual Experience: Creation, Processing and Interactivity of High-Resolution and High-Dimensional Video Signals. Springer (2010)
15. 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., Schmidt, F., eds.: Statistical and Geometrical Approaches to Visual Motion Analysis. Volume 5064 of Lecture Notes in Computer Science. Springer (2009) 23–45
16. Chen, K., Lorenz, D.: Image sequence interpolation using optimal control. Journal of Mathematical Imaging and Vision **41** (2011) 222–238
17. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. International Journal of Computer Vision **61** (2005) 211–231
18. Weickert, J.: Theoretical foundations of anisotropic diffusion in image processing. In Kropatsch, W.G., Klette, R., Solina, F., eds.: Theoretical Foundations of Computer Vision. Volume 11 of Computing Supplement., Springer (1994) 221–236
19. Galić, I., Weickert, J., Welk, M., Bruhn, A., Belyaev, A., Seidel, H.P.: Image compression with anisotropic diffusion. Journal of Mathematical Imaging and Vision **31** (2008) 255–269 10.1007/s10851-008-0087-0.
20. Ghodstinat, M., Bruhn, A., Weickert, J.: Deinterlacing with motion-compensated anisotropic diffusion. In Cremers, D., Rosenhahn, B., Yuille, A.L., Schmidt, F.R., eds.: Statistical and Geometrical Approaches to Visual Motion Analysis. Springer-Verlag, Berlin, Heidelberg (2009) 91–106
21. Volz, S., Bruhn, A., Valgaerts, L., Zimmer, H.: Modeling temporal coherence for optical flow. In Metaxas, D.N., Quan, L., Sanfeliu, A., Gool, L.J.V., eds.: IEEE International Conference on Computer Vision (ICCV). (2011) 1116–1123