Optical Flow Driven Frame Interpolation

Lars Lau Rakêt[†] and Søren Forchhammer[‡]

[†]*Department of Computer Science, University of Copenhagen, Universitetsparken 5, Copenhagen 2100, Denmark*
[‡]*Department of Photonics Engineering, Technical University of Denmark, Building 343, Lyngby 2800, Denmark.*

[†]*larslau@diku.dk,* [‡]*sofo@fotonik.dtu.dk*

Abstract: Variational motion estimation techniques have improved tremendously in recent years—both in terms of speed and accuracy. In this paper a method for optical flow driven frame interpolation is proposed, where the optical flow is recovered from a TV-L^1 energy. We first propose a number of modifications to the optical flow algorithm in order to improve interpolation accuracy. We then move on and propose the inclusion of an additional symmetric optical flow in a standard forward-backward frame interpolation scheme. We demonstrate that the proposed method consistently outperforms interpolation using state of the art overlapped block motion compensation and previous methods using TV-L^1 optical flow, with an average improvement of more than 1 dB.

OCIS codes: 000.0000, 150.4620, 110.4153.

1. Introduction

The problem of frame interpolation find uses in a number of fields, e.g. video post processing, restoration of historic material, and video coding. For some applications the goal is to satisfy a viewer, in which case the main concern often is that the results look good [1], rather than minimizing some error measure. Here we will consider the problem for technical applications where error measures are more important than a crisp result. One of these types of application is distributed video coding [2]. In video coding applications discrete methods like block matching has been used very successfully, and variational motion estimation methods have largely been overlooked. One reason for this is that optical flow fields are dense, and thus problematic to code. In distributed video coding, the source statistics are exploited at decoder side, eliminating the problem of coding the flow field motion vectors. Such a setup makes it possible to exploit the highly accurate motion estimates of modern optical flow methods [3].

2. 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 v such that the motion matches the image sequence while still maintaining sufficient regularity. Here we will consider a TV-L^1 energy for the optical flow estimation, which is given by

$$E(v) = \int \|I_1(x+v(x)) - I_0(x)\| \, \mathrm{d}x + \int \|Dv(x)\| \, \mathrm{d}x, \tag{1}$$

where the first term is a L^1 norm of the difference between I_0 and the motion-compensated version of I_1, and the second term is a total variation (TV) regularization, which is to be understood as the integral over the Frobenius norm of the derivative of v. This type of regularization will smooth the estimated motion while still allowing for sharp motion boundaries. In order to efficiently minimize E we introduce two relaxations. First we linearize the data fidelity term $I_1(x+v) - I_0(x) \approx \rho(v)(x)$, where ρ is the first order Taylor approximation

$$\rho(v)(x) = I_1(x+v_0) - I_0 + (v(x)-v_0)^\top \nabla I_1(x+v_0) \tag{2}$$

with v_0 the current estimate of v around x. We further relax E by introducing an auxiliary variable u that splits data fidelity and regularization in two quadratically coupled energies:

$$E_1(v) = \int \lambda \|\rho(v)(x)\| + \frac{1}{2\theta}\|v(x)-u(x)\|^2 \, \mathrm{d}x, \tag{3}$$

$$E_2(u) = \int \frac{1}{2\theta}\|v(x)-u(x)\|^2 + \|Du(x)\| \, \mathrm{d}x, \tag{4}$$

The above relaxation was first proposed by Zach et al. [4], and has a number of advantages, most notably that the two problems can be solved pointwise which makes the solution very easy to implement on massively parallel processors like graphics processing units (GPUs). We will not replicate the minimizing solutions to (3) and (4) here, but note that (3) can easily be solved by the method presented in [4] in the case of grayscale images and in the general case of vector valued images it can be solved by the method presented in [5]. The regularization energy (4) is minimized by the projection method of Chambolle [6, 7].

In order to improve interpolation quality we use a specialized coarse-to-fine pyramidal implementation of the above algorithm (for more details on standard implementations refer to [5, 8]). We have 70 pyramid levels with a scaling factor of 0.95, where each pyramid level is smoothed with a Gaussian with standard deviation $\frac{\sqrt{2}}{4}$ before downscaling to the coarser level. On each level we do 30 warps of first solving (3) and then solving (4) using 10 iterations of the algorithm of Bresson [7], with $\lambda = 3$ and $\theta = 0.2$, where in order to improve interpolation quality, ρ has been weighted by the gradient magnitude $\|\nabla I_1(x+v_0)+0.01\|$ (slightly shifted to avoid division by 0) in the minimization of (3) [9]. Additional improvement of interpolation quality was found by applying a 3×3 median filter of the flow after upscaling to the next pyramid level [10].

3. Frame Interpolation algorithm and results

We are interested in interpolating an in-between frame $I_{1/2}$ using only the two surrounding frames I_0 and I_1. We first note that the optical flow algorithm presented in the previous section is asymmetric, since the (forward) flow estimated from I_0 to I_1 is not the same as the (backward) flow from I_1 to I_0. In addition the forward flow will have a coordinate system corresponding to the pixels in I_0 and the backward flow follows the coordinate system given by the pixels in I_1, so in order to use these flows to interpolate at pixel positions in $I_{1/2}$ we need to warp the flows [11–13] to match the intermediate frame. This is done by assuming that the intermediate frame follows the estimated motion linearly, and then defining the warped forward flow as the flow from $I_{1/2}$ to I_1, which is approximated by

$$v_f^{1/2}(\mathrm{round}(x + 1/2 v_f(x))) = 1/2 v_f(x), \tag{5}$$

where the round function rounds to nearest pixel. The warped backward flow is estimated similarly. There are problems in the warping procedure, first multiple flow vectors may hit the same pixel $\mathrm{round}(x+1/2v_f(x))$ (typically occlusion), which can be dealt with by choosing the vector with best data fidelity. A more serious problem is the problem of dis-occlusion which causes holes in the warped flow. We will correct this by filling holes using an outside-in strategy, however ideally one would reason about depth and occlusion in the interpolation procedure, which should give better results [12].

With the warped flows, the straightforward approach for interpolation is to interpolate along the flow vectors,

$$I_{1/2}(x) = \frac{1}{2}(I_1(x + v_f^{1/2}(x)) + I_0(x + v_b^{1/2}(x))), \tag{6}$$

however, since we have discarded occlusion information by filling holes and clearing collisions, the warped forward flow should minimize $I_1(x + v_f^{1/2}(x)) + I_1(x - v_f^{1/2}(x))$, and vice versa for the backward flow. Even though the two computed flows are symmetric around $I_{1/2}$, they will be different since they originated from asymmetric flows. We propose to include a symmetric flow estimate which is calculated directly using the pixel positions of the unknown frame $I_{1/2}$, to complement the two asymmetric flows. This flow v_{sym} is calculated using the reparametrization of (3) suggested in [8, 14], which will also help in producing a more robust flow. With these three flows we can do the interpolation as follows

$$\begin{aligned} I_{1/2}(x) = \frac{1}{6}(&I_1(x + v_f^{1/2}(x)) + I_1(x - v_b^{1/2}(x)) + I_1(x + v_{\mathrm{sym}}(x)) \\ +\ &I_0(x - v_f^{1/2}(x)) + I_0(x + v_b^{1/2}(x)) + I_0(x - v_{\mathrm{sym}}(x))), \end{aligned} \tag{7}$$

i.e. the interpolation is the average of the two surrounded frames warped to the center using the three different flows.

We will evaluate (7) which we will denote 3OF on the test sequences (QCIF, 15 fps) *Coastguard* QP=26, *Foreman* QP=25, *Hall* QP=24 and *Soccer* QP=25, where we interpolate every other frame and compare to the overlapped block motion compensation (OBMC) method from [3] and the TV-L^1 optical flow (OF) method from [2]. The results can be found in Table 1 where we see that the proposed method outperforms OBMC and OF on all sequence, with an average increase in PSNR of 1.26 dB over OBMC and 2.25 dB over OF.

Sequence	OBMC	OF	3OF
Coastguard	31.83	30.92	32.71
Foreman	29.26	29.28	30.19
Hall	36.46	32.28	37.25
Soccer	21.30	22.43	23.75

Table 1. Average PSNR across the 74 interpolated frames for the four test sequences.

4. Conclusion

We have proposed an improved algorithmic setup for the TV-L^1 optical flow method, as well as a method of interpolation that uses three complementary motion estimates to build the in-between image frame. We have demonstrated that the methods significantly improve interpolation quality over methods used in current state-of-the-art distributed video coding [2], which in turn means that the inclusion of this frame interpolation method in a the codec should improve the performance of the codec further.

References

1. S. Keller, F. Lauze, and M. Nielsen, "Temporal super resolution using variational methods," in "High-Quality Visual Experience: Creation, Processing and Interactivity of High-Resolution and High-Dimensional Video Signals," , M. Mrak, M. Grgic, and M. Kunt, eds. (Springer, 2010).
2. X. Huang, L. L. Rakêt, H. V. Luong, M. Nielsen, F. Lauze, and S. Forchhammer, "Proc. multi-hypothesis transform domain Wyner-Ziv video coding including optical flow," in "Multimedia Signal Processing," (2011).
3. X. Huang and S. Forchhammer, "Cross-band noise model refinement for transform domain Wyner-Ziv video coding," Signal Processing: Image Communication **27**, 16–30 (2005).
4. C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L^1 optical flow," in "Pattern Recognition," , vol. 4713 of *Lecture Notes in Computer Science*, F. Hamprecht, C. Schnörr, and B. Jähne, eds. (Springer, 2007), pp. 214–223.
5. L. L. Rakêt, L. Roholm, M. Nielsen, and F. Lauze, "TV-L^1 optical flow for vector valued images," in "Energy Minimization Methods in Computer Vision and Pattern Recognition," , vol. 6819 of *Lecture Notes in Computer Science*, Y. Boykov, F. Kahl, V. Lempitsky, and F. Schmidt, eds. (Springer, 2011), pp. 329–343.
6. A. Chambolle, "An algorithm for total variation minimization and applications," Journal of Mathematical Imaging and Vision **20**, 89–97 (2004).
7. X. Bresson and T. Chan, "Fast dual minimization of the vectorial total variation norm and application to color image processing," Inverse Problems and Imaging **2**, 455–484 (2008).
8. L. L. Rakêt, L. Roholm, A. Bruhn, and J. Weickert, "Motion compensated frame interpolation with a symmetric optical flow constraint," in "Advances in Visual Computing," (Springer, 2012), Lecture Notes in Computer Science.
9. H. Zimmer, A. Bruhn, and J. Weickert, "Optic flow in harmony," International Journal of Computer Vision **93**, 368–388 (2011).
10. M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof, "Anisotropic Huber-L1 optical flow," in "BMVC," (2009).
11. S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," International Journal of Computer Vision **31**, 1–31 (2011).
12. E. Herbst, S. Seitz, and S. Baker, "Occlusion reasoning for temporal interpolation using optical flow," Tech. Rep. UW-CSE-09-08-01, Department of Computer Science and Engineering, University of Washington (2009).
13. M. Werlberger, T. Pock, M. Unger, and H. Bischof, "Optical flow guided TV-L^1 video interpolation and restoration," in "Energy Minimization Methods in Computer Vision and Pattern Recognition," , vol. 6819 of *Lecture Notes in Computer Science*, Y. Boykov, F. Kahl, V. Lempitsky, and F. Schmidt, eds. (Springer, 2011), pp. 273–286.
14. L. Alvarez, C. Castaño, M. García, K. Krissian, L. Mazorra, A. Salgado, and J. Sánchez, "Symmetric optical flow," in "Computer Aided Systems Theory–EUROCAST 2007," , vol. 4739 of *Lecture Notes in Computer Science*, R. Díaz, F. Pichler, and A. Arencibia, eds. (Springer, 2007), pp. 676–683.