Optical Flow Guided TV-L^1 Video Interpolation and Restoration

Manuel Werlberger, Thomas Pock, Markus Unger, and Horst Bischof*

Institute for Computer Graphics and Vision, Graz University of Technology, Austria
{werlberger,pock,unger,bischof}@icg.tugraz.at

Abstract. The ability to generate intermediate frames between two given images in a video sequence is an essential task for video restoration and video post-processing. In addition, restoration requires robust denoising algorithms, must handle corrupted frames and recover from impaired frames accordingly. In this paper we present a unified framework for all these tasks. In our approach we use a variant of the TV-L^1 denoising algorithm that operates on image sequences in a space-time volume. The temporal derivative is modified to take the pixels' movement into account. In order to steer the temporal gradient in the desired direction we utilize optical flow to estimate the velocity vectors between consecutive frames. We demonstrate our approach on impaired movie sequences as well as on benchmark datasets where the ground-truth is known.

1 Introduction

With the rise of digitalization in the film industry, the restoration of historic videos gained in importance. The cause of degradation of the original material is to chemical decomposition or the abrasion of repeated playback. Another typical artifact with aged films is the occurrence of noise, be it because of dirt, dust, scratches, or just because of long-term storage. For an overview of potential artifacts and their occurrence we want to refer to the survey of Kokaram [1]. In areas where film reels get bonded together when played back, several sequent frames often are entirely destructed. The recreation of frames and parts of sequences can be used to restore corrupted film segments on the one hand, but on the other the generation of time interpolated viewpoints is essential for the post production industry. A prevalent application is the generation of slow-motion or the simulation of tracking shots (a film sequence where the camera is mounted on a wheeled platform). A lot of methods deal with the problem of frame interpolation. Let us describe some approaches in this field:

Image-based rendering often relies on additional geometric constraints of the scenes like *e.g.* the 3D scene geometry, scene depth or epipolar constraints [2,3,4,5,6]. With the known set of 2D images and the additional geometric constraints a 3D model is built and used to render novel views. To achieve a higher

* This work was supported by the BRIDGE project HD-VIP (no. 827544).

Y. Boykov et al. (Eds.): EMMCVPR 2011, LNCS 6819, pp. 273–286, 2011.

Fig. 1. Image sequence interpolation. The first and the last frames are given and the frames in between are generated by our method. In the cutout in the second row it can be seen that motion is interpolated naturally in the generated frames (the heads movement, mouth closing, *etc.*). Note that the occluded region are also handled correctly despite of the noisy input material. This is nicely illustrated by continuing the white pole in the background when the head disoccludes this region.

accuracy when relying on the reconstructed 3D geometry it is common practice to use calibrated and synchronized camera arrays. Besides that, the movie industry often uses more simple approaches to create visual effects that need interpolated frames. **View morphing** [7,8,9,10,11] uses some correspondences between images to transform both into an intermediate position and afterwards superimpose and blend both warped images to generate a single interpolated result. Correspondences can be introduced by a user that defines corresponding points manually, or by some feature matching algorithm. The general problem of image morphing techniques is that the warping and blending might introduce errors when dealing with complex motions between the known images. Especially in presence of (dis-)occlusions (caused by moving objects) the approach might exhibit artifacts in these regions. In [12], a path-based approach is introduced that estimates a path for every pixel. To estimate intermediate pixels, the paths are linearly blended between the known positions. Due to the definition of a path in every pixel, no holes are produced when generating interpolated frames. Although, close relations to optical flow [13] are given, the authors emphasize that there is no need for calculating optical flow to create pixel correspondences. They call their approach an inverse optical flow algorithm. Recently a method of computing optical flow in an optimal control framework [14] is used for image sequence interpolation [15]. Optical flow in the control framework searches for the flow field such that the interpolated image matches the given input image and is therefore very suitable for the sequence interpolation task.

In this work we propose a method that is capable of image sequence denoising, restorating impaired frames, reconstructing completely lost frames and interpolating images between frames. Note that all this is handled within a rather simple model. Though, the results can compete with current state-of-the art and even outperform some existing methods. As a motivation we show two frames of a historical video sequence and two in-between images generated by our method in Fig. 1. The cutout region shows that existing motion (head movement) and deformation (closing of the mouth) are handled correctly and produce reasonable intermediate images. As our model is closely related to variational image denoising we first give some insights into this topic.

In the field of image restoration, total variation based methods were studied intensly in the last decade starting with the introduction of the so-called ROF denoising model in the seminal work of Rudin, Osher and Fatemi [16]. Minimizing the energy functional

$$\min_u \left\{ \int_\Omega |\nabla u| + \lambda \left(u - f\right)^2 \, dx \right\} \tag{1}$$

computes the desired solution u that is most likely fitting the observed input image f both defined in the image domain Ω. $|\nabla u|$ denotes the well-known total variation used as smoothness prior and preserving the edges of the reconstruction u. This model is appropriate to remove additive Gaussian noise. Similar to the ROF model, the TV-L^1 model [17,18]

$$\min_u \left\{ \int_\Omega |\nabla u| + \lambda \left|u - f\right| \, dx \right\} \tag{2}$$

is a popular approach when it comes to removing impulse noise. Besides the ability to remove strong outliers, the substitution of the ℓ_2 norm in the ROF model with the ℓ_1 norm makes the TV-L^1 model contrast invariant. In addition to image denoising applications, (2) can be used for shape denoising and feature selection tasks [19].

In this paper we extend the idea of the TV-L^1 denoising approach for the task of film restoration and frame interpolation. We introduce the proposed model in Section 2. Then, in Section 3 we discuss the optimization procedure. Next, the applications and results are presented in Section 4 and finally, Section 5 concludes the work and gives an outlook.

2 The Optical Flow Guided TV-L^1 Model

In this section we define a variant of the TV-L^1 model that operates on a spatio-temporal volume. For tracking and segmentation it has been shown in [20,21] that the representation of image sequences in a space-time volume is beneficial. In the following we will show that this is also true if a spatio-temporal TV-L^1 model is utilized for restoration and video processing tasks. The drawback of the methods proposed up to now is the definition of the temporal gradient. To handle

moving objects and especially the associated (dis-)occlusions, we incorporate optical flow to guide the temporal derivative. Hence, the time derivatives are computed along objects movement trajectories. In order to be robust against occlusions and disocclusions we compute the optical flow in both direction which we will denote as forward flow $v^+ = \left(v_1^+, v_2^+\right)^T : \Omega \times T \to \mathbb{R}^2$ and backward flow $v^- = \left(v_1^-, v_2^-\right)^T : \Omega \times T \to \mathbb{R}^2$. To estimate the optical flow for the existing frames, we use the implementation of [22] as it is publicly available, is reasonable fast to compute and achieves good results especially for the frame interpolation Middlebury benchmark [23]. The approach uses a combination of classical optical flow constraint [13] and a weighted regularization that considers the strength and the direction of underlying image edges. For an overview on optical flow methods we refer to [23,24,25,26,27] and the references therein.

We propose to minimize the following optical flow driven TV-L^1 energy functional:

$$\min_u \int_{\Omega \times T} |\nabla_v u| + \lambda(x,t)\,|u - f|\; dx\, dt \tag{3}$$

The input sequence $f : \Omega \times T \to \mathbb{R}$ and the sought solution $u : \Omega \times T \to \mathbb{R}$ are defined in a space-time volume and $\lambda(x,t)$ defines the trade-off between regularization and data term. The spatio-temporal gradient operator is defined as

$$\nabla_v = (\partial_{x_1}, \partial_{x_2}, \partial_{t^+}, \partial_{t^-})^T \quad , \tag{4}$$

where the components ∂_{x_1} and ∂_{x_2} denote the spatial derivatives. The temporal derivatives are directed by the corresponding optical flow vectors (v^+, v^-) and are denoted as ∂_{t^+} and ∂_{t^-}. We may assume that u is sufficiently smooth such that ∇_v exists:

$$\begin{cases} \partial_{x_1} u = \lim\limits_{h \to 0^+} \dfrac{u(x_1 + h, x_2, t) - u(x,t)}{h} \\ \partial_{x_1} u = \lim\limits_{h \to 0^+} \dfrac{u(x_1, x_2 + h, t) - u(x,t)}{h} \\ \partial_{t^+} u = \lim\limits_{h_t \to 0^+} \dfrac{u(x + h_t v^+(x,t), t + h_t) - u(x,t)}{h_t \sqrt{1 + ||v^+(x,t)||_2^2}} \\ \partial_{t^-} u = \lim\limits_{h_t \to 0^+} \dfrac{u(x + h_t v^-(x,t), t + h_t) - u(x,t)}{h_t \sqrt{1 + ||v^-(x,t)||_2^2}} \end{cases} . \tag{5}$$

3 Minimizing the Optical Flow Guided TV-L^1 Model

3.1 Discretization

First, we define the image sequence within a three dimensional, regular Cartesian grid of the size $M \times N \times K : \{(m,n,k) : 1 \le m \le M, 1 \le n \le N, 1 \le k \le K\}$. The grid size is 1 and the discrete pixel positions within the defined volume are given by (m,n,k). We use finite dimensional vector spaces $X = \mathbb{R}^{MNK}$

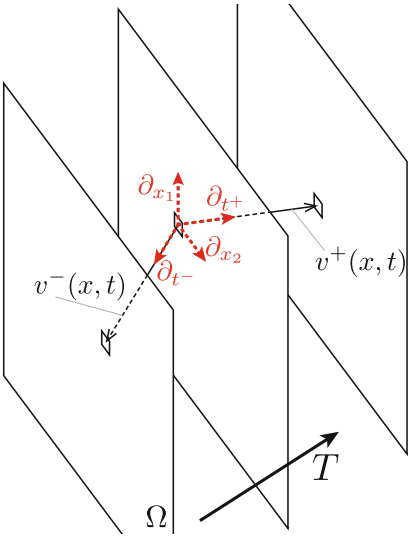


Fig. 2. Layout of the spatio-temporal volume and the defined gradient operator (4), which directions are marked as red vectors. The forward v^+ and backward optical flow v^- guides the temporal components of this gradient operator.

and $Y = \mathbb{R}^{4MNK}$ equipped with standard scalar products denoted by $\langle \cdot, \cdot \rangle_X$ and $\langle \cdot, \cdot \rangle_Y$. Next, we define the discretized versions of the anisotropic spatio-temporal gradient operator ∇_v (see (4) and (5)) with h and h_t controlling the spatial and temporal influence of the regularization

$$(D_v u)_{m,n,k} = \big(\underbrace{(D_{x_1} u)_{m,n,k}, (D_{x_2} u)_{m,n,k}}_{spatial}, \underbrace{(D_{t^+} u)_{m,n,k}, (D_{t^-} u)_{m,n,k}}_{temporal} \big)^T \quad . \tag{6}$$

The spatial gradients are discretized using simple finite differences yielding

$$(D_{x_1} u)_{m,n,k} = \begin{cases} \frac{u_{m+1,n,k} - u_{m,n,k}}{h} & \text{if } m < M \\ 0 & \text{if } m = M \end{cases} , \tag{7}$$

$$(D_{x_2} u)_{m,n,k} = \begin{cases} \frac{u_{m,n+1,k} - u_{m,n,k}}{h} & \text{if } n < N \\ 0 & \text{if } n = N \end{cases} . \tag{8}$$

For the flow-guided temporal gradients we use a linear interpolation within the plane, if the flow vector points in-between discrete locations on the pixel grid. In the following we describe the interpolation for the spatial component guided with the forward flow. The same is of course valid for the component where the gradient is steered with the backward optical flow. The coordinates used for linearly interpolating the position yield

$$(\overline{m}^+, \overline{n}^+, k) = \left(m + h_t v^{1+}_{m,n,k}, n + h_t v^{2+}_{m,n,k}, k \right) \quad , \tag{9}$$

with the neighboring coordinates used for the linear interpolation

$$\begin{aligned} \overline{m}_1^+ &= \lfloor \overline{m}^+ \rfloor \ , \quad \overline{m}_2^+ = \overline{m}_1^+ + 1 \ , \\ \overline{n}_1^+ &= \lfloor \overline{n}^+ \rfloor \ , \quad \overline{n}_2^+ = \overline{n}_1^+ + 1 \ . \end{aligned} \tag{10}$$

and the corresponding weighting factors are computed as the distances

$$d_m^+ = \overline{m}_2^+ - \overline{m}^+ \quad \text{and} \qquad d_n^+ = \overline{n}_2^+ - \overline{n}^+ \ . \tag{11}$$

The temporal gradient operators then yield

$$(D_{t^+}u)_{m,n,k} = \begin{cases} \frac{a^+ + b^+ + c^+ + d^+ - e^+}{h_t \sqrt{1 + ||v^+_{m,n,k}||_2^2}} & \text{if} \quad a^+, b^+, c^+, d^+, e^+ \in \Omega \times T \\ 0 & \text{else} \ , \end{cases} \tag{12}$$

$$(D_{t^-}u)_{m,n,k} = \begin{cases} \frac{a^- + b^- + c^- + d^- - e^-}{h_t \sqrt{1 + ||v^-_{m,n,k}||_2^2}} & \text{if} \quad a^-, b^-, c^-, d^-, e^- \in \Omega \times T \\ 0 & \text{else} \ , \end{cases} \tag{13}$$

with

$$\begin{aligned} a^+ &= d_m^+ d_n^+ u_{\overline{m}_1^+, \overline{n}_1^+, k+1} \\ b^+ &= (1 - d_m^+) d_n^+ u_{\overline{m}_2^+, \overline{n}_1^+, k+1} \\ c^+ &= d_m^+ (1 - d_n^+) u_{\overline{m}_1^+, \overline{n}_2^+, k+1} \\ d^+ &= (1 - d_m^+)(1 - d_n^+) u_{\overline{m}_2^+, \overline{n}_2^+, k+1} \\ e^+ &= u_{m,n,k} \end{aligned} \qquad \left| \qquad \begin{aligned} a^- &= d_m^- d_n^- u_{\overline{m}_1^-, \overline{n}_1^-, k} \\ b^- &= (1 - d_m^-) d_n^- u_{\overline{m}_2^-, \overline{n}_1^-, k} \\ c^- &= d_m^- (1 - d_n^-) u_{\overline{m}_1^-, \overline{n}_2^-, k} \\ d^- &= (1 - d_m^-)(1 - d_n^-) u_{\overline{m}_2^-, \overline{n}_2^-, k} \\ e^- &= u_{m,n,k+1} \end{aligned} \right. \tag{14}$$

To implement the operator D_v we use a sparse matrix representation. This is of particular interest because we will also need to implement the adjoint operator D_v^* which is defined through the identity $\langle D_v u, p \rangle_Y = \langle u, D_v^* p \rangle_X$. Note that here, the adjoint operator is simply the matrix transpose.

Now, (3) can be rewritten in the discrete setting as the following minimization problem

$$\min_u \ ||D_v u||_{\ell_1} + ||\Lambda(u - f)||_{\ell_1} \quad , \quad \text{with } \Lambda = \operatorname{diag}(\lambda) \ . \tag{15}$$

Dualizing both ℓ_1 norms in (15) yields the saddle-point problem

$$\begin{aligned} &\min_u \max_{p,q} \langle D_v u, p \rangle_Y + \Lambda \langle u - f, q \rangle_X \quad , \\ &\qquad \text{s.t.} \quad 0 \le u \le 1 \ , \quad ||p||_\infty \le 1 \quad \text{and} \quad -1 \le q \le 1 \end{aligned} \tag{16}$$

where $p = \left(p^1_{m,n,k}, p^2_{m,n,k}, p^3_{m,n,k}, p^4_{m,n,k} \right) \in Y$ and $q = (q_{m,n,k}) \in X$ denote the dual variables, and the discrete maximum norm $||p||_\infty$ is defined as

$$\begin{aligned} ||p||_\infty &= \max_{m,n,k} |p_{m,n,k}| \quad , \\ |p_{m,n,k}| &= \sqrt{(p^1_{m,n,k})^2 + (p^2_{m,n,k})^2 + (p^3_{m,n,k})^2 + (p^4_{m,n,k})^2} \end{aligned} \tag{17}$$

3.2 Primal-Dual Algorithm

In order to optimize (16) we use the first order primal-dual algorithm proposed by Chambolle and Pock [28] yielding the following algorithm for updating primal and dual variables:

$$\begin{cases} p^{n+1} = \Pi_{\mathcal{B}_1}\left[p^n + \sigma\left(D_v\left(2u^n - u^{n-1}\right)\right)\right] \\ q^{n+1} = \Pi_{[-1,1]}\left[q^n + \sigma\Lambda\left((2u^n - u^{n-1}) - f\right)\right] \\ u^{n+1} = u^n - \tau\left(D_v^* p + \Lambda q\right) \end{cases} \tag{18}$$

The projection $\Pi_{\mathcal{B}_\zeta}$ is a simple point-wise projections onto the ball with the radius ζ and $\Pi_{[a,b]}$ denotes the point-wise truncation to the interval $[a, b]$. τ and σ are the primal and dual update step sizes satisfying $\tau\sigma L^2 \leq 1$, where $L^2 = \|(D_v^*, \Lambda^*)\|^2$.

3.3 The Optical Flow Guided TV-L^1 Model for Frame Interpolation

The task of frame interpolation needs a more detailed setting yielding the desired result. The main demand on intermediate frames between two images is the interpolation of occurring movements. The aim is to generate frames interpolating the moving objects along the movement trajectories in a natural and appropriate way. Hence, the presented approach is suitable due to the incorporation of optical flow information for computing the temporal derivatives. To accomplish the task of frame interpolation the spatio-temporal volume gets scaled to the desired size. The new and unknown frames are currently left blank for the input volume f. In these areas no data term can be computed and therefore the values for λ are set to zero for all unknown pixels. In terms of optical flow, the forward and backward flow are first computed between all the known neighboring frames. To get estimates for the optical flow at currently unknown positions, and hence describe the motion trajectories of the pixels for the intermediate positions, we assume a linear movement between the available frames. To fill-in optical flow vectors at unknown positions the known flow fields are rescaled with a 'stretch factor' so that the resultant vectors describe the optical flow between the known frames and the subsequent (unknown) frame. The positions for the initial points of optical flow vectors within unknown frames are located by propagating the known vectors through the unknown parts of the volume. Therefore, the vector coordinates at available frames are taken and propagated to the point where the vector aims at. This is most likely an unknown position due to the previous rescaling of the vector field. As the vectors generally not end at a discrete pixel location, the vector field is then filled with linearly interpolated values calculated with the factors from (11). A visual description of this process is given in Figure 3. When solving (3) with this setting the regularization propagates the desired pixel values along the motion trajectories and results in adequate intermediate frames. Due to the robustness of the L^1 norm against outlier this approach is also suitable for restorating or denoising corrupted frames.

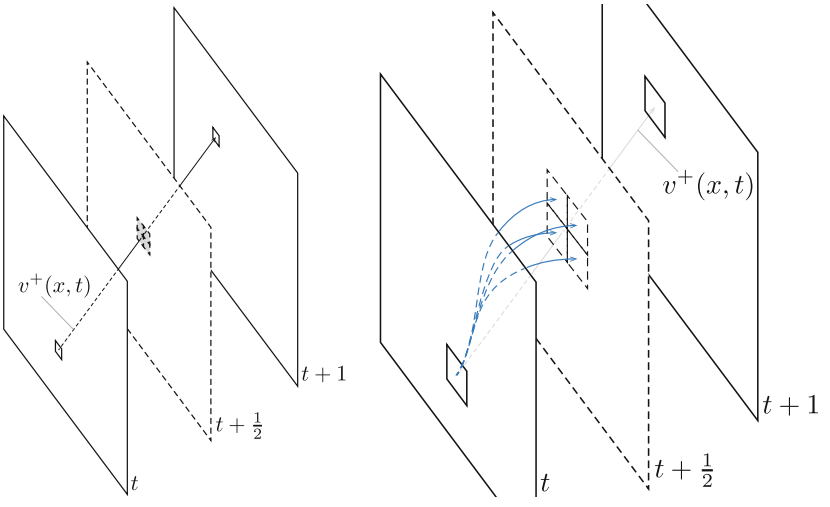


Fig. 3. Flow propagation: The left visualization shows basic setting of an unknown frame ($t, t+1$: known; $t+\frac{1}{2}$: unknown) and an exemplar forward flow vector. The right image shows the relevant pixels that are used when the optical flow is propagated using the linear interpolation factors as in (11).

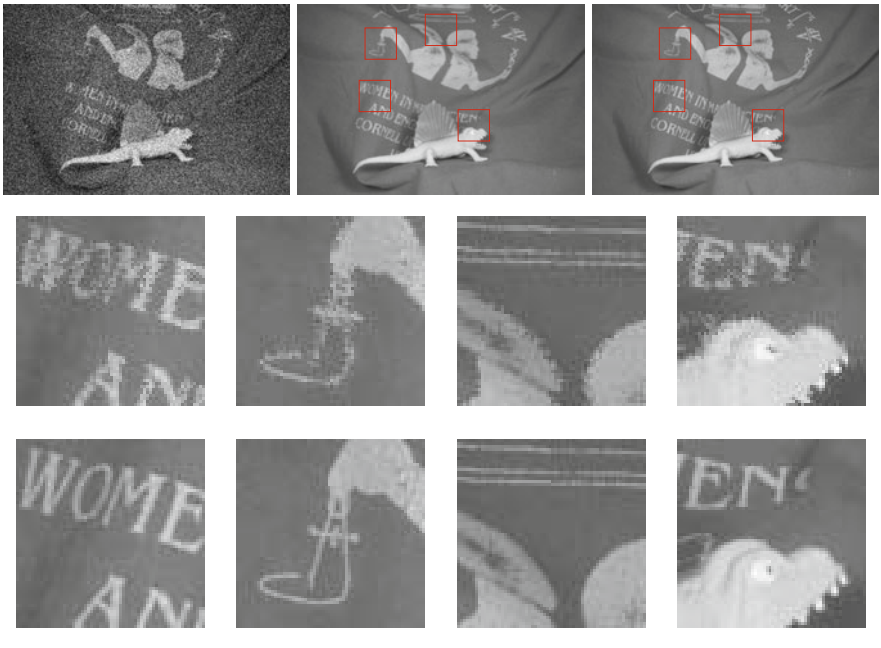


Fig. 4. Single frame denoising of `Dimetrodon` dataset. The first row shows the noisy input image, and the denoised images with the TV-L^1 model and the TV-L^1 model (left to right). Zoomed regions show results of the TV-L^1 model in first row and TV-L^1 model in the last row.

4 Application and Numerical Evaluations

4.1 Denoising

In this section, we want to highlight the benefit of the directed temporal gradients in the proposed method compared to the classical TV-L^1 denoising extended to a spatio-temporal volume but with standard temporal gradients. In Fig. 4 a sequence of three frames is taken and random Gaussian noise is added to the middle frame. To obtain reasonable optical flow we compute the flow vectors between the intact frames and use the setting as described in Section 3.3 to obtain flow vectors for the noisy image. The weighting parameter is set to $\lambda(x,t) = 2$ for all the pixels within the space-time volume. To increase the temporal influence of the regularization we choose $h = 10$ and $h_t = 1$.

4.2 Inpainting

Historic video material is often impaired and for the next application we will deal with partly damaged areas within frames. Small outliers and artifacts might be handled by denoising methods as shown in Section 4.1. Nevertheless, the impairments are often bigger than such algorithms can handle. To use our method for restoration, we compute the optical flow between the neighboring image pairs. As the impairments will cause outliers in the optical flow field, we cannot use

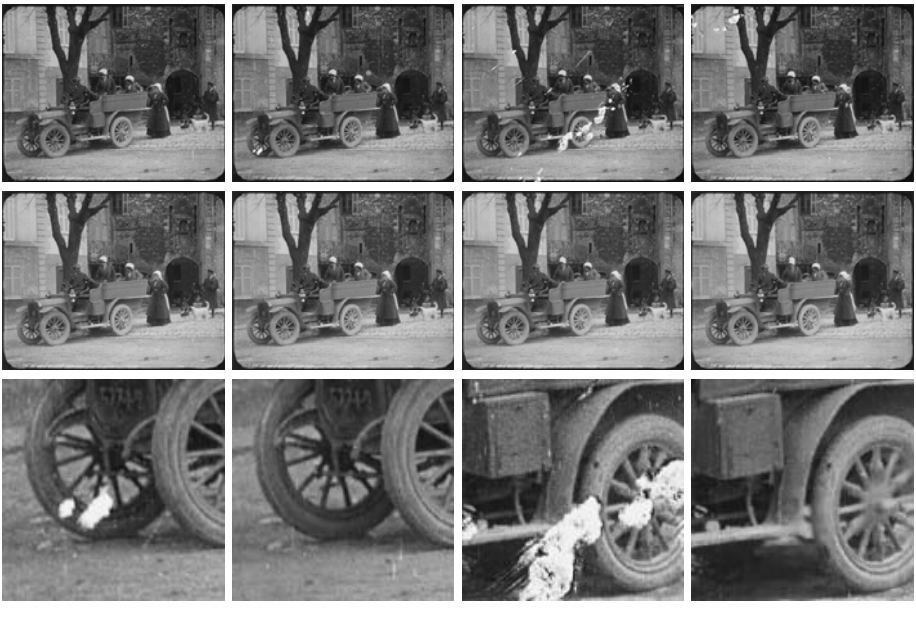
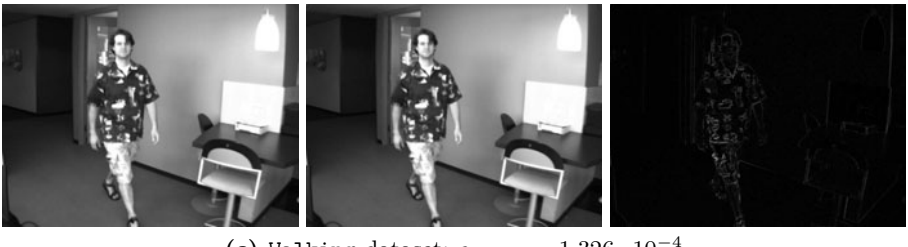


Fig. 5. Inpainting corrupted regions. The first row shows the input images, the second row the restorated frames and the third shows 2 pairs of images where the left is the impaired region and the right shows the result of our approach.

(a) `Dimetrodon`-dataset: $e_{\mathrm{RMSE}} = 7.478 \cdot 10^{-3}$

(b) `MiniCooper`-dataset: $e_{\mathrm{RMSE}} = 3.167 \cdot 10^{-4}$



(c) `Walking` dataset: $e_{\mathrm{RMSE}} = 1.326 \cdot 10^{-4}$

Fig. 6. Frame interpolation on different dataset. left column: ground-truth u^*; middle column: our interpolation result u; right column: difference image u_{diff}.

Fig. 7. Comparison to [12] (first row). Our result (second row) produces almost identical results compared to current state-of-the art.

Fig. 8. Interpolating the `Basketball`-sequence

those vectors as guidance when solving (3). There are of course several methods to obtain reasonable flow fields. In the following example we simply inpaint the flow fields with the help of our model but neglect the direction for the temporal gradient in this case. These interpolated flow fields are then used to solve (3). Of course more sophisticated methods like vector inpainting methods [29] can be used here. Still, our approach is robust enough to handle gross outliers like in Figure 5. In this experiment the values for λ are set to $\lambda(x,t) = 2$ if the pixels are considered good and to $\lambda(x,t) = 0$ otherwise. The weightings of the regularity is chosen as $h = 2.5$ and $h_t = 1$.

4.3 Image Sequence Interpolation

For image interpolation we use the ability of regularizing along the temporal trajectory in the space-time volume as described in Section 3.3. When solving (3) in such a setting the intermediate frames are generated and the pixels' movement are reasonably interpolated with the propagated optical flow vectors. For a numerical evaluation of interpolating intermediate frames within an image sequence we use sequences of the Middlebury benchmark [23], where interpo-

Fig. 9. Interpolating the `Backyard`-sequence

lated frames are available as ground truth. The ground-truth datasets, as well as the input data, are available at the Middlebury website[1]. For comparison we show the interpolated image u, the ground-truth u^* and the difference image $u_{\text{diff}} = |u^* - u|$ in Fig. 6. As an quantitative measurement we give the root mean squared error (RMSE) $e_{\text{RMSE}} = \sqrt{\frac{1}{MN}\sum_{n=1}^{N}\sum_{m=1}^{M}\left(u^*_{m,n} - u_{m,n,k}\right)^2}$. Next, we compare our approach to an interpolation result of [12] in Fig. 7 where our method generates almost identical intermediate frames. In Fig. 8 and Fig. 9 two examples of the Middlebury database are shown. Here larger displacements, faster movements, small scaled structures and complex occlusions are the major difficulties within the two presented datasets. The zoomed regions show that again the interpolated movement is reasonable and the (dis-)occluded regions are handled robustly.

5 Conclusion

In this paper we presented a method that can handle the problem of sequence denoising, restorating partly corrupted frames within image sequences, recover completely lost frames and interpolate intermediate frames between two given images. We proposed to use an optical flow directed temporal gradient to preserve and, in case of frame interpolation, generate natural object movements.

Since the method is very dependent on the quality of the used optical flow, the robustness on generating the unknown flow vectors could be investigated in the future. Especially for the restoration task in Section 4.2 the optical flow cannot be computed in impaired regions and must be interpolated somehow in these areas. Although we have shown that our method is robust enough to use very simple completion techniques, a further investigation in such interpolation techniques would even further increase the robustness and applicability of our approach.

References

1. Kokaram, A.C.: On Missing Data Treatment for Degraded Video and Film Archives: A Survey and a New Bayesian Approach. IEEE Trans. on IP (2004)
2. Chen, S.E., Williams, L.: View Interpolation for Image Synthesis. In: SIGGRAPH (1993)
3. McMillan, L., Bishop, G.: Plenoptic Modeling: An Image-Based Rendering System. In: ACM SIGGRAPH (1995)
4. Faugeras, O., Robert, L.: What Can Two Images Tell Us About a Third One? International Journal of Computer Vision (1996)
5. Wexler, Y., Sashua, A.: On the Synthesis of Dynamic Scenes from Reference Views. In: IEEE Conference on Computer Vision and Pattern Recognition (2000)
6. Vedula, S., Baker, S., Kanade, T.: Image-based spatio-temporal modeling and view interpolation of dynamic events. ACM Transactions on Graphics, TOG (2005)

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

7. Beier, T., Neely, S.: Feature-Based Image Metamorphosis. In: ACM SIGGRAPH (1992)
8. Seitz, S.M., Dyer, C.R.: View Morphing. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (1996)
9. Wolberg, G.: Image morphing: a survey. The Visual Computer (1998)
10. Schaefer, S., McPhail, T., Warren, J.: Image deformation using moving least squares. ACM Transactions on Graphics, TOG (2006)
11. Stich, T., Linz, C., Albuquerque, G., Magnor, M.: View and Time Interpolation in Image Space. Computer Graphics Forum (2008)
12. Mahajan, D., Huang, F.C., Matusik, W., Ramamoorthi, R., Belhumeur, P.: Moving Gradients: A Path-Based Method for Plausible Image Interpolation. In: ACM SIGGRAPH (2009)
13. Horn, B.K., Schunck, B.G.: Determining Optical Flow. Artificial Intelligence (1981)
14. Borzì, A., Ito, K., Kunisch, K.: Optimal Control Formulation for Determining Optical Flow. SIAM Journal on Scientific Computing (2006)
15. Chen, K., Lorenz, D.A.: Image Sequence Interpolation Using Optimal Control. Journal of Mathematical Imaging and Vision (2011)
16. Rudin, L.I., Osher, S.J., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena (1992)
17. Nikolova, M.: A Variational Approach to Remove Outliers and Impulse Noise. Journal of Mathematical Imaging and Vision (2004)
18. Chan, T.F., Esedoglu, S.: Aspects of Total Variation Regularized L[sup 1] Function Approximation. SIAM Journal of Applied Mathematics (2005)
19. Pock, T.: Fast Total Variation for Computer Vision. PhD thesis, Graz University of Technology (2008)
20. Mansouri, A.R., Mitiche, A., Aron, M.: PDE-based region tracking without motion computation by joint space-time segmentation. In: ICIP (2003)
21. Unger, M., Mauthner, T., Pock, T., Bischof, H.: Tracking as Segmentation of Spatial-Temporal Volumes by Anisotropic Weighted TV. In: Cremers, D., Boykov, Y., Blake, A., Schmidt, F.R. (eds.) EMMCVPR 2009. LNCS, vol. 5681, pp. 193–206. Springer, Heidelberg (2009)
22. Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., Bischof, H.: Anisotropic Huber-L1 Optical Flow. In: BMVC (2009)
23. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., Szeliski, R.: A Database and Evaluation Methodology for Optical Flow. IJCV (2011)
24. Barron, J.L., Fleet, D., Beaucheming, S.S.: Performance of optical flow techniques. International Journal of Computer Vision (1994)
25. Beauchemin, S.S., Barron, J.L.: The Computation of Optical Flow. ACM Computing Surveys (1995)
26. Aubert, G., Deriche, R., Kornprobst, P.: Computing Optical Flow via Variational Techniques. SIAM J. Appl. Math. (2000)
27. Fleet, D.J., Weiss, Y.: Optical Flow Estimation (2006)
28. Chambolle, A., Pock, T.: A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. J. of Math. Imaging and Vision (2010)
29. Berkels, B., Kondermann, C., Garbe, C., Rumpf, M.: Reconstructing Optical Flow Fields by Motion Inpainting. In: Cremers, D., Boykov, Y., Blake, A., Schmidt, F.R. (eds.) EMMCVPR 2009. LNCS, vol. 5681, pp. 388–400. Springer, Heidelberg (2009)