

Anisotropic Huber-L¹ Optical Flow

Manuel Werlberger¹
werlberger@icg.tugraz.at

Werner Trobin¹
trobin@icg.tugraz.at

Thomas Pock¹
pock@icg.tugraz.at

Andreas Wedel²
wedel@cs.uni-bonn.de

Daniel Cremers²
dcremers@cs.uni-bonn.de

Horst Bischof¹
bischof@icg.tugraz.at

¹ Institute for Computer Graphics and
Vision
Graz University of Technology

² Department of Computer Science
University of Bonn

Abstract

The presented work is motivated by the problem of restoring severely degraded historic video material via an optical flow-based interpolation. In order to increase the robustness as well as the accuracy of discontinuity preserving variational optical flow models, we propose two extensions. First, we will discuss the deficiencies of an isotropic Total Variation regularization and introduce an anisotropic (*i.e.* image-driven) regularization based on the robust Huber norm. Second, to cope with the gross outliers contained in historic video material, we propose a novel spatio-temporal regularization approach. Instead of assuming gradual flow changes over time, we impose a symmetry constraint with respect to a central frame. The benefit of the suggested enhancements is illustrated qualitatively on historic video material and quantitatively on the Middlebury optical flow benchmark.

1 Introduction

The estimation of the optical flow between two images, usually taken in close temporal succession, is one of the key problems in low-level vision. Ignited by the two seminal works of Lucas & Kanade [1] and Horn & Schunck [2], a diverse range of optical flow estimation techniques have been developed and we refer to the surveys [3, 4, 5] for a detailed review of the vast amount of related literature.

In order to compare the accuracy of these flow estimation techniques and, even more importantly, to illustrate where they still fail, challenging benchmark sequences with a reliable ground truth are required. The so-called Middlebury dataset [6] is the most recent effort in this direction and we will use these sequences throughout this work. On the corresponding evaluation site at <http://vision.middlebury.edu/flow/>, discontinuity preserving variational models based on Total Variation (TV) regularization and L¹ data terms are

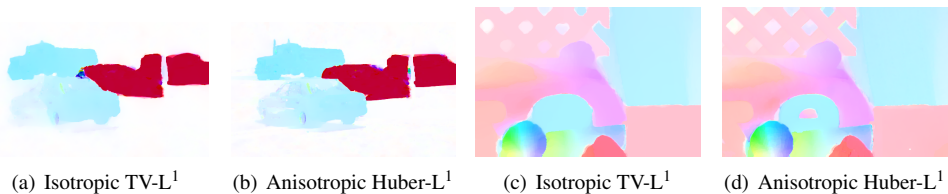


Figure 1: Comparison of isotropic and anisotropic Huber regularization on (a), (b) the Dumptruck and (c), (d) the RubberWhale sequence (AAE = $4.06^\circ \rightarrow 2.93^\circ$, AEPE = $0.13 \rightarrow 0.09$). Please note the hole in the semicircle-shaped element and the more distinct lattice-edges in the anisotropic case

among the most accurate flow estimation techniques, but there is still room for improvement. While Papenberg *et al.* [10] mainly focused on improving the data conservation constraints, we will take a closer look at the deficiencies of isotropic TV regularization and propose an improved regularizer in Section 2.

Total Variation regularization is an L^1 penalization of the flow gradient magnitudes, and due to the tendency of the L^1 norm to favor sparse solutions (*i.e.* lots of ‘zeros’), the fill-in effect caused by the regularizer leads to piecewise constant solutions in weakly textured areas. This effect, known as ‘staircasing’ in a 1D setting, can be reduced significantly by using a quadratic penalization for small gradient magnitudes while sticking to linear penalization for larger magnitudes to maintain the discontinuity preserving properties known from TV. Huber [8] proposed such a function as an alternative to quadratic penalization in the field of robust statistics, and Shulman and Hervé were the first to apply it in the context of flow estimation [14].

Based on the two observations that motion discontinuities often occur along object boundaries and that in turn object boundaries often coincide with large image gradients, Nagel and Enkelmann [10] proposed to adapt the regularization to the local image structure. Even for the quadratic regularizers used at that time, their anisotropic (image-driven) regularization decreased the well-known oversmoothing effects of the Horn & Schunck model [4] by impeding smoothing across image edges. Therefore, we propose to replace the isotropic TV regularization, *e.g.* used in [10, 14], with an anisotropic (image-driven) Huber regularization term.

A second contribution of this paper is motivated by an application: the restoration of historic video material via flow-based video interpolation. To cope with the gross outliers contained in the historic video material (*cf.* Figs. 4 and 5), we propose a novel spatio-temporal regularization approach in Section 3 and compare it against the well-known spatio-temporal regularization proposed in [17]. After a short discussion of implementation details in Section 4 and an evaluation of the presented models on the Middlebury dataset in Section 5, we conclude with a few final remarks in Section 6.

To facilitate further development and to simplify comparisons to alternative flow estimation methods, the Matlab source of the presented flow estimation algorithm can be downloaded from <http://www.gpu4vision.org>. In addition, we offer a GPU-based high-performance implementation in binary form to encourage the use of state-of-the-art flow estimation techniques in higher-level vision applications.

2 Anisotropic Huber-L¹ Optical Flow

To make this paper self-contained, we briefly discuss the starting point for our extensions – a disparity preserving, spatially continuous formulation of the optical flow problem based on an L¹ data term and isotropic TV regularization [□, □]. For two input images I_0 and I_1 defined on a rectangular domain $\Omega \in \mathbb{R}^2$, the model can be stated as

$$\min_{\vec{u}} \left\{ \int_{\Omega} \sum_{d=1}^2 |\nabla u_d| + \lambda |I_1(\vec{x} + \vec{u}(x)) - I_0(\vec{x})| d\vec{x} \right\}, \quad (1)$$

where $\vec{u}(\vec{x}) = (u_1(\vec{x}), u_2(\vec{x}))^T$, $u_d : \Omega \rightarrow \mathbb{R}$ and the free parameter λ is used to balance the relative weight of data and regularization term. In contrast to [□], we linearize the data term, which yields a convex optimization problem

$$\min_{\vec{u}} \left\{ \int_{\Omega} \sum_{d=1}^2 |\nabla u_d| + \lambda |\rho(\vec{u}(\vec{x}))| d\vec{x} \right\}, \quad (2)$$

where $\rho(\vec{u}(\vec{x})) = \vec{u}(\vec{x})^T \nabla I_1(\vec{x}) + I_1(\vec{x}) - I_0(\vec{x})$ is the well-known optical flow constraint equation. In order to accommodate large displacements, this linearized model has to be embedded in a coarse-to-fine/warping scheme [□]. Nevertheless, to streamline the presentation we will only consider the linearized model from now on.

The functional (2) is hard to minimize in this setting, so to simplify the optimization procedure we introduce an auxiliary variable \vec{v} and a coupling term to ensure that \vec{v} is a close approximation of \vec{u}

$$\min_{\vec{u}, \vec{v}} \left\{ \int_{\Omega} \sum_{d=1}^2 \left[|\nabla u_d| + \frac{1}{2\theta} (u_d - v_d)^2 \right] + \lambda |\rho(\vec{v}(\vec{x}))| d\vec{x} \right\}, \quad (3)$$

where θ is a small positive constant [□].

To introduce the anisotropic Huber regularization, we start with a simple manipulation of a discontinuity preserving, image-driven regularization term

$$\int_{\Omega} \sum_{d=1}^2 \Psi_{\varepsilon} \left\{ (\nabla u_d)^T D \nabla u_d \right\} d\vec{x} = \int_{\Omega} \sum_{d=1}^2 \Psi_{\varepsilon} \left\{ \left(D^{1/2} \nabla u_d \right)^T \left(D^{1/2} \nabla u_d \right) \right\} d\vec{x}, \quad (4)$$

where D is a symmetric, positive definite diffusion tensor and $\Psi_{\varepsilon}(s^2)$ a (robust) penalty function. Replacing the isotropic TV regularization in (3) with such an image-driven regularizer leads to

$$\min_{\vec{u}, \vec{v}} \left\{ \int_{\Omega} \sum_{d=1}^2 \left[|\vec{q}_d|_{\varepsilon} + \frac{1}{2\theta} (u_d - v_d)^2 \right] + \lambda |\rho(\vec{v}(\vec{x}))| d\vec{x} \right\}, \quad \text{with} \quad (5)$$

$$|\vec{q}_d|_{\varepsilon} = \begin{cases} \frac{|\vec{q}_d|^2}{2\varepsilon} & |\vec{q}_d| \leq \varepsilon \\ |\vec{q}_d| - \frac{\varepsilon}{2} & \text{else} \end{cases}, \quad (6)$$

where $\vec{q}_d = D^{1/2} \nabla u_d$ for brevity. A reasonable choice for the diffusion tensor is $D^{1/2} = \exp(-\alpha |\nabla I|^{\beta}) \vec{n} \vec{n}^T + \vec{n}^{\perp} \vec{n}^{\perp T}$, where $\vec{n} = \frac{\nabla I}{|\nabla I|}$, and \vec{n}^{\perp} denotes a vector normal to \vec{n} .

Chambolle proposed a projected gradient descent algorithm to minimize the ROF model, which is based on a dual formulation of the TV [5]. In order to obtain a similar algorithm to minimize (5), we need the Legendre-Fenchel (LF) dual [13] of the anisotropic Huber norm $f_H(\vec{q}_d) = |\vec{q}_d|_\varepsilon$, which is given as

$$f_H^*(\vec{p}_d) = \sup_{\vec{q}_d} \left\{ \vec{q}_d \cdot \vec{p}_d - f_H(\vec{q}_d) \right\} = I_{\{|\vec{p}_d| \leq 1\}} + \varepsilon \frac{|\vec{p}_d|^2}{2}, \quad (7)$$

where $*$ denotes the LF transform, \vec{p} is the dual variable, and the so-called indicator function $I_\Sigma(\vec{x})$ is 0 if $\vec{x} \in \Sigma$ and ∞ otherwise. The biconjugate of f_H can be obtained by applying the LF transform to the conjugate $f_H^*(\vec{p}_d)$:

$$f_H^{**}(\vec{q}_d) = \sup_{\vec{p}_d} \left\{ \vec{p}_d \cdot \vec{q}_d - f_H^*(\vec{p}_d) \right\} = \sup_{|\vec{p}_d| \leq 1} \left\{ \vec{p}_d \cdot \vec{q}_d - \varepsilon \frac{|\vec{p}_d|^2}{2} \right\}. \quad (8)$$

The LF transform is self-inversive due to $f_H(\vec{q}_d)$ being convex and lower semi-continuous in \vec{q}_d . Therefore, the biconjugate is equivalent to the initial anisotropic Huber norm $f_H^{**}(\vec{q}_d) = f_H(\vec{q}_d)$. Incorporating this equality in (5) yields

$$\min_{\vec{u}, \vec{v}} \sup_{|\vec{p}_d| \leq 1} \left\{ \int_{\Omega} \sum_{d=1}^2 \left[\left(D^{1/2} \nabla u_d \right) \cdot \vec{p}_d - \varepsilon \frac{|\vec{p}_d|^2}{2} + \frac{1}{2\theta} (u_d - v_d)^2 \right] + \lambda |\rho(\vec{v}(\vec{x}))| d\vec{x} \right\}. \quad (9)$$

This convex minimization problem in \vec{u} and \vec{v} can be solved iteratively by an alternating minimization procedure:

1. For fixed \vec{v} , solve:

$$\min_{\vec{u}} \sup_{|\vec{p}_d| \leq 1} \left\{ \int_{\Omega} \sum_{d=1}^2 \left[\left(D^{1/2} \nabla u_d \right) \cdot \vec{p}_d - \varepsilon \frac{|\vec{p}_d|^2}{2} + \frac{1}{2\theta} (u_d - v_d)^2 \right] d\vec{x} \right\} \quad (10)$$

This optimization problem is similar to the well-known ROF model [14] and can be solved using a variant of the projected gradient descent algorithm proposed in [5]. Applying the divergence theorem on (10) and using $D^{1/2} = [D^{1/2}]^\top$ we get

$$\min_{\vec{u}} \sup_{|\vec{p}_d| \leq 1} \left\{ \int_{\Omega} \sum_{d=1}^2 \left[-u_d \operatorname{div} \left(D^{1/2} \vec{p}_d \right) - \varepsilon \frac{|\vec{p}_d|^2}{2} + \frac{1}{2\theta} (u_d - v_d)^2 \right] d\vec{x} \right\}, \quad (11)$$

which is now a pointwise problem in \vec{u} , and hence

$$\frac{\partial}{\partial u_d} : -\operatorname{div} \left(D^{1/2} \vec{p}_d \right) + \frac{1}{\theta} (u_d - v_d) = 0 \quad \rightarrow \quad u_d = v_d + \theta \operatorname{div} \left(D^{1/2} \vec{p}_d \right) \quad (12)$$

The functional derivative of (10) w.r.t. \vec{p}_d is given by

$$\frac{\partial}{\partial \vec{p}_d} : D^{1/2} \nabla u_d - \varepsilon \vec{p}_d = D^{1/2} \nabla \left(v_d + \theta \operatorname{div} \left(D^{1/2} \vec{p}_d \right) \right) - \varepsilon \vec{p}_d \quad (13)$$

Embedding this into a projected gradient descent update scheme, leads to the iterative scheme

$$u_d^{n+1} = v_d^n + \theta \operatorname{div} \left(D^{1/2} \bar{p}_d^{n+1} \right), \quad (14)$$

$$\bar{p}_d^{n+1} = \frac{\bar{p}_d^n + \tau \left(D^{1/2} \nabla u_d^{n+1} - \varepsilon \bar{p}_d^n \right)}{\max \left\{ 1, \left| \bar{p}_d^n + \tau \left(D^{1/2} \nabla u_d^{n+1} - \varepsilon \bar{p}_d^n \right) \right| \right\}}, \quad (15)$$

where the step-width is given by $\tau = 1/(4 + \varepsilon)$.

2. For fixed \vec{u} , solve for the auxiliary variable \vec{v} :

$$\min_{\vec{v}} \left\{ \int_{\Omega} \frac{1}{2\theta} \sum_{d=1}^2 (u_d - v_d)^2 + \lambda \int_{\Omega} |\rho(\vec{v}(\vec{x}))| d\vec{x} \right\} \quad (16)$$

Here a simple thresholding step with three different cases yields a direct solution [9]:

$$\vec{v}^{n+1} = \vec{u}^{n+1} + \begin{cases} \lambda \theta \nabla I_1 & \text{if } \rho(\vec{u}^{n+1}) < -\lambda \theta |\nabla I_1|^2 \\ -\lambda \theta \nabla I_1 & \text{if } \rho(\vec{u}^{n+1}) > \lambda \theta |\nabla I_1|^2 \\ -\rho(\vec{u}^{n+1}) \frac{\nabla I_1}{|\nabla I_1|^2} & \text{else} \end{cases} \quad (17)$$

The difference between an isotropic TV and an isotropic Huber regularization is illustrated in Fig. 2, showing a rendering of u_1 from the `Dimetron` sequence (cf. Fig. 7(a)). The color coded flow is superimposed as texture. While the TV regularization causes the characteristic piecewise constant levels (cf. Fig. 2(a), AAE = 3.03°, AEPE = 0.16), the Huber regularization (cf. 2(b), $\varepsilon = 0.01$, AAE = 2.77°, AEPE = 0.14) yields a significantly smoother result. In Fig. 2(c) the influence of the threshold ε , which defines the location of the transition from a quadratic to a linear penalty, on the average angular error (AAE, cf. [9]) and the average end-point error (AEPE, cf. [9]) is shown on the `Dimetron` dataset. Please note that the Huber norm approaches the TV as $\varepsilon \rightarrow 0$. The benefits of integrating directional information are depicted in Fig. 1.

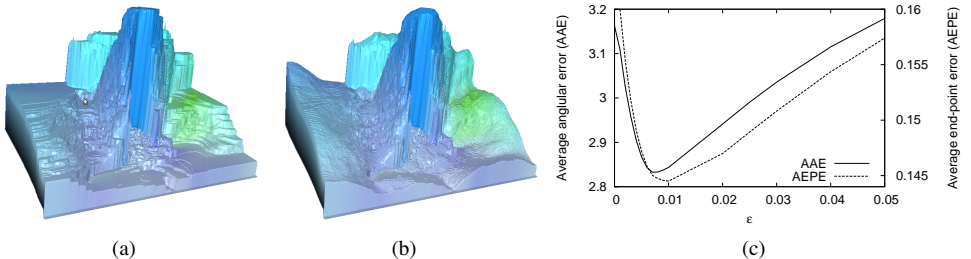


Figure 2: Comparing (a) the staircasing afflicted TV regularization and (b) the Huber regularization on the `Dimetron` dataset; (c) average angular and average end-point error as a function of the Huber norm parameter ε

3 A Temporal Symmetry Constraint

A majority of optical flow estimation methods impose only spatial continuity constraints to solve the aperture problem. Widely cited extensions to a spatio-temporal continuity include [10] for a causal method, [11] for a method that is symmetric *w.r.t.* ‘past’ and ‘future’ in the image sequence and therefore only suitable for offline processing and [12] for a patch-based variant. Here we will only consider the spatio-temporal regularization presented in [13], which is based on the assumption that the flow field changes gradually over time. This assumption has been used successfully on the well-known Yosemite benchmark sequence (with clouds), where the average angular error could be reduced from 2.44° to 1.78° by including a temporal continuity constraint [13]. Considering the quoted improvements, it is surprising that not a single one out of 27 flow estimation methods currently listed on the Middlebury benchmark site employs such a spatio-temporal continuity constraint. Upon closer inspection, however, it turns out that the properties of the Yosemite sequence (steady camera motion through a distant, static scene combined with relatively small displacements) are perfectly suitable for temporal smoothing, whereas the Middlebury benchmark sequences contain more complex motions as well as significantly larger displacements, so using the spatio-temporal regularization proposed in [13] has adverse effects on the estimated flows.

Instead of assuming gradual flow changes through time, we propose a 3-frame method that ‘mirrors’ the flow symmetric *w.r.t.* the central frame by extending the anisotropic flow (5) with one additional data fidelity term

$$\min_{\vec{u}, \vec{v}} \left\{ \int_{\Omega} \sum_{d=1}^2 \left[|\vec{q}_d|_{\varepsilon} + \frac{1}{2\theta} (u_d - v_d)^2 \right] + \lambda |\rho_{1c}(\vec{v}(\vec{x}))| + \lambda |\rho_{3c}(\vec{v}(\vec{x}))| d\vec{x} \right\}, \quad (18)$$

where ρ_{1c} and ρ_{3c} denote the linearized brightness constancy constraints between the first and the central frame and between the third and the central frame, respectively:

$$\rho_{1c}(\vec{v}(\vec{x})) = -\vec{v}(\vec{x})^T \nabla I_1(\vec{x}) + I_1(\vec{x}) - I_c(\vec{x}) \quad (19)$$

$$\rho_{3c}(\vec{v}(\vec{x})) = \vec{v}(\vec{x})^T \nabla I_3(\vec{x}) + I_3(\vec{x}) - I_c(\vec{x}) . \quad (20)$$

[13] shows how to extend the thresholding scheme (17) using a second data fidelity term.

Conceptually, this model implies – independently for every pixel – a linear motion from the first to the third frame, which is obviously constrained to be symmetric to the central



Figure 3: Comparison of: (a) ground truth; (b) spatio-temporal TV regularization (AAE=11.70°, AEPE=1.75); (c) anisotropic Huber-L¹ flow with symmetry constraint (AAE=6.32°, AEPE=0.70) on the Urban3 sequence. For the result without symmetry constraint we refer to Fig. 7(h)

frame. The key difference to [17] is that the frames one and three are both warped towards the central frame, so this model is applicable to a wider class of motions. We illustrate this difference using the `Urban3` sequence of the Middlebury database. The color-coded ground truth flow is depicted in Fig. 3(a), the result of a spatio-temporal TV regularization (AAE=11.70°, AEPE=1.75) can be observed in Fig. 3(b), and the result of the proposed model (AAE=6.32°, AEPE=0.70) is depicted in Fig. 3(c). The simultaneous presence of outliers (red and green blobs) and heavy oversmoothing in Fig. 3(b) indicates that the as-

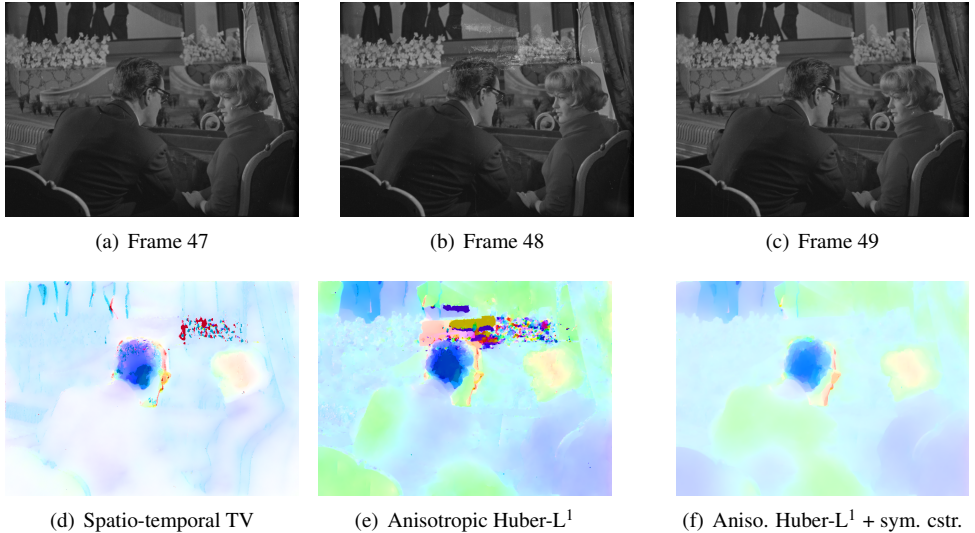


Figure 4: ‘Krems’ sequence

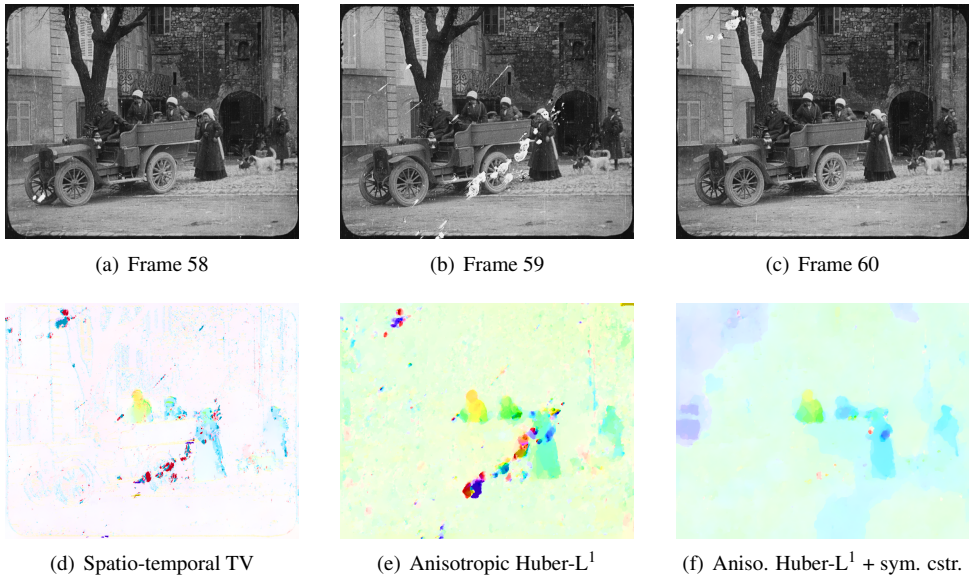


Figure 5: Sequence ‘Car Ride in the Pyrenees’, 1910

assumption of a gradual flow field change over time is invalid for this sequence.

The proposed symmetry constraint is specifically targeted at video restoration applications, where single frames are degraded, *e.g.* with blobs and scratches, and trades some flexibility for robustness. Hence, compared to a purely spatial regularization, the accuracy is slightly inferior on the clean Middlebury data (*cf.* Fig. 7(h) and Table 1). However, the comparison on the sequences shown in Figs. 4 and 5, where the central frames are severely degraded, illustrates that the imposed symmetry of the flow increases the robustness of the estimate in the presence of gross outliers, compared to both spatio-temporal TV regularization [10] as well as the anisotropic (two-frame) flow proposed in Section 2.

4 Implementation

As stated in Section 2, the proposed algorithm is based on a linearized data term and therefore has to be embedded in a coarse-to-fine/warping scheme. For all experiments in this paper, the image pyramid was constructed using a scale factor of 0.8. In a slight departure from [10], we additionally decomposed every image in the pyramid into a ‘structure’ and a ‘texture’ part via ROF denoising [14] ($\lambda = 10$). The actual input to the flow algorithm was a linear combination of these two parts with the relative weights structure:texture = 1:4 to reduce the impact of illumination changes.

The derivative operators were discretized as forward differences with reflecting (Neumann) boundary conditions. Hence, the divergence operator is based on backward differences with Dirichlet boundary conditions. The iterative solution procedure starts with $\vec{u} = 0$ on the coarsest scale of the pyramid and since we aim for high-accuracy, we repeatedly warp the ‘moving’ image [6] with a median-filtered version of the current solution. This 3x3 median filter is also applied when prolonging the flow to the next finer level in the image pyramid, because it helps to get rid of outliers.

We implemented our proposed optical flow approach using Matlab and in addition we have a GPU-accelerated C++ implementation, using the Nvidia CUDA framework. The parallelization capabilities of modern GPUs are well suited to optimize such variational problems.

Average angle error	avg. rank	Amy (Hidden texture)			Mequon (Hidden texture)			Schefflera (Hidden texture)			Wooden (Hidden texture)			Grove (Synthetic)			Urban (Synthetic)			Yosemite (Synthetic)			Teddy (Steebo)		
		GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1	GT	im0	im1
		all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
CompOF [27]	4.8	3.44 ₉	11.2 ₉	4.04 ₉	2.51 ₂	9.77 ₈	1.74 ₁	3.93 ₄	10.6 ₄	2.04 ₁	3.87 ₈	18.8 ₈	2.19 ₇	3.17 ₁	4.00 ₁	2.92 ₁	3.64 ₈	13.8 ₉	3.64 ₅	2.17 ₂	3.35 ₁	2.51 ₁₂	3.08 ₂	7.04 ₂	3.65 ₇
Adaptive [26]	4.9	3.29 ₁	9.43 ₁	2.28 ₁	3.10 ₆	11.47	2.46 ₇	6.58 ₉	15.7 ₉	2.52 ₅	3.14 ₁	15.6 ₁	1.56 ₁	3.67 ₇	4.46 ₆	3.48 ₈	3.32 ₁	13.0 ₂	2.38 ₁	2.76 ₁₂	4.39 ₁₂	1.93 ₇	3.58 ₅	8.18 ₅	2.88 ₂
Anso. Huber-L1 [28]	6.2	3.71 ₃	10.1 ₃	3.08 ₃	4.36 ₁₃	13.0 ₁₈	3.77 ₁₃	6.92 ₁₂	15.3 ₇	3.60 ₁₂	3.54 ₂	15.9 ₂	2.04 ₅	3.38 ₉	4.45 ₄	2.47 ₂	3.88 ₂	12.9 ₁	2.74 ₂	3.37 ₁₈	4.35 ₁₁	2.85 ₁₈	3.16 ₃	7.52 ₃	2.90 ₃
Spatically variant [22]	6.7	3.73 ₄	10.2 ₃	3.33 ₃	3.02 ₅	11.0 ₈	2.67 ₄	5.95 ₅	13.8 ₈	2.35 ₂	3.67 ₇	19.3 ₈	1.84 ₃	3.81 ₉	4.51 ₁₄	3.69 ₁₁	4.48 ₅	16.0 ₈	3.90 ₆	2.11 ₄	3.26 ₂	2.12 ₈	4.66 ₁₀	9.41 ₁₀	4.35 ₁₂
TV-L1-improved [20]	7.6	3.35 ₂	9.63 ₂	2.62 ₂	2.82 ₄	10.7 ₂	2.23 ₃	6.50 ₆	15.8 ₁₀	2.73 ₆	3.80 ₆	21.3 ₁₁	1.75 ₂	3.34 ₂	4.38 ₄	2.39 ₁	5.97 ₁₈	18.1 ₁₄	5.67 ₁₂	3.57 ₁₇	4.92 ₁₈	3.43 ₁₈	4.01 ₈	9.84 ₁₁	3.44 ₄
Average end-point error	avg. rank	Amy (Hidden texture)			Mequon (Hidden texture)			Schefflera (Hidden texture)			Wooden (Hidden texture)			Grove (Synthetic)			Urban (Synthetic)			Yosemite (Synthetic)			Teddy (Steebo)		
Adaptive [26]	4.8	0.09 ₁	0.26 ₁	0.06 ₁	0.23 ₂	0.78 ₈	0.18 ₁	0.54 ₁₀	1.19 ₁₂	0.21 ₅	0.18 ₁	0.91 ₂	0.10 ₁	0.88 ₄	1.25 ₄	0.73 ₅	0.50 ₂	1.28 ₂	0.31 ₂	0.14 ₁₀	0.16 ₁₂	0.22 ₈	0.55 ₂	1.37 ₄	0.79 ₃
CompOF [27]	6.0	0.11 ₇	0.28 ₅	0.10 ₈	0.18 ₆	0.63 ₃	0.12 ₁	0.31 ₄	0.75 ₄	0.18 ₁	0.19 ₂	0.97 ₅	0.12 ₃	0.87 ₇	1.31 ₁	1.00 ₁₁	1.73 ₁₉	1.73 ₉	0.87 ₁₃	0.11 ₄	0.12 ₃	0.22 ₈	0.68 ₅	1.48 ₅	0.95 ₇
Anso. Huber-L1 [28]	6.2	0.10 ₂	0.28 ₅	0.08 ₃	0.31 ₁₁	0.88 ₈	0.28 ₁₂	0.55 ₁₂	1.13 ₉	0.29 ₁₂	0.20 ₄	0.92 ₃	0.13 ₆	0.84 ₇	1.20 ₂	0.70 ₂	0.39 ₁	1.23 ₃	0.28 ₁	0.17 ₁₈	0.15 ₁₀	0.27 ₁₆	0.64 ₁	1.36 ₂	0.79 ₃
Spatically variant [22]	6.8	0.10 ₃	0.27 ₄	0.08 ₃	0.22 ₂	0.75 ₅	0.19 ₁	0.43 ₃	1.00 ₈	0.18 ₁	0.19 ₂	1.05 ₇	0.10 ₁	1.05 ₁₀	1.41 ₁₁	1.16 ₁₅	0.59 ₅	1.61 ₈	0.43 ₄	0.13 ₈	0.11 ₁	0.28 ₁₇	0.95 ₁₂	1.72 ₁₁	1.28 ₁₅
TV-L1-improved [20]	7.8	0.09 ₁	0.26 ₁	0.07 ₂	0.20 ₄	0.71 ₄	0.16 ₁	0.53 ₉	1.18 ₁₁	0.22 ₆	0.21 ₇	1.24 ₁₄	0.11 ₃	0.90 ₆	1.31 ₁	0.72 ₃	1.51 ₁₉	1.93 ₁₁	0.84 ₉	0.18 ₁₈	0.17 ₁₄	0.31 ₁₉	0.73 ₉	1.62 ₉	0.87 ₆

Figure 6: Screenshots of the Middlebury benchmark results; More details and the given references are available at <http://vision.middlebury.edu/flow/>.

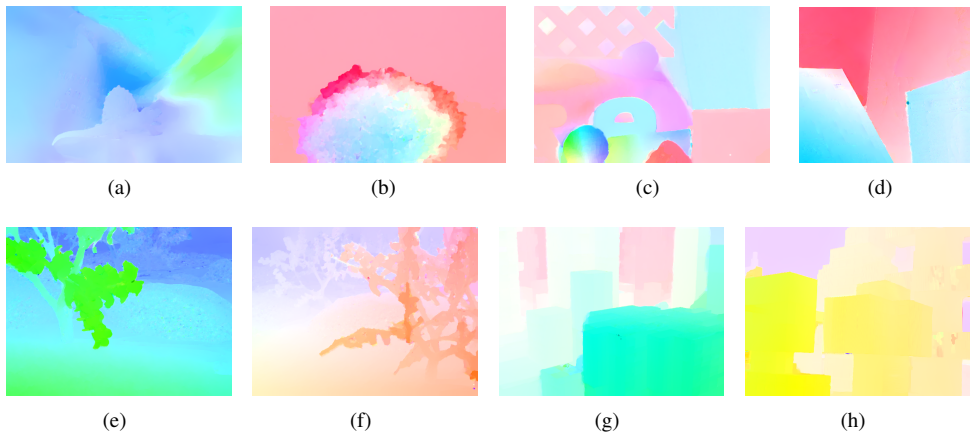


Figure 7: Anisotropic Huber- L^1 flow results on the training sequences of the Middlebury benchmark dataset. The second row in Table 1 lists the respective average end-point errors

5 Evaluation

To quantify the improvements of the proposed anisotropic Huber regularization over TV regularization and to compare it to other state-of-the-art algorithms, we evaluated our algorithm on the Middlebury benchmark sequences [10]. As required, the parameters were held constant for all sequences and, in fact, for all the other experiments presented in this paper as well. We used a scale factor of 0.8 for the image pyramid and set $\lambda = 40$, $\theta = 0.1$, $\varepsilon = 0.01$, $\alpha = 5.0$ and $\beta = 0.5$. At the time of submission, the proposed method was ranked within the top four results for almost all error measures listed on <http://vision.middlebury.edu/flow/>, cf. Fig. 6 for the average end-point error and the angular error (both ranked third). While the anisotropic Huber- L^1 model yields accurate results on most sequences, there are some notable outliers: *Mequon*, *Schefflera*, and *Yosemite*. On the *Mequon* sequence, the university logo on the T-shirt causes strong edges in the image, thereby inducing a large anisotropy in the regularization. At the same time, the T-shirt is only weakly shaded, hence the regions close to the motion discontinuity around the two figures are handled incorrectly. The fragmented occlusion caused by the *Schefflera* leaf in the right/top corner of the aptly named sequence is hard to get right for local optimization approaches, let alone the repetitive texture and the illumination changes. On the noisy *Yosemite* sequence, the accuracy could easily be improved by pre-smoothing or a significantly lower λ . For completeness, we also give an overview of the results on the Middlebury training dataset in Fig. 7. The corresponding average end-point errors are shown in the second row of Table 1. In the first row of Table 1, the results of an isotropic TV regularization can be seen and the last row lists the results obtained when imposing the symmetry constraint we proposed in Section 3. As discussed previously, this modeling constraint is too restrictive for the complex motions contained in the Middlebury sequences. However, for the degraded historic sequences in Figs. 4 and 5 the qualitative improvement is evident, although we do not have ground truth data to back this up quantitatively.

Our C++/CUDA implementation of the proposed algorithm is highly configurable and the user can either emphasize performance or accuracy. Using the ‘high-accuracy’ settings mentioned above with 10 warps and 50 iterations per warp, the flow estimation takes 1.13

Algorithm	Dimetrodon	Hydrangea	RubberWhale	Venus	Grove2	Grove3	Urban2	Urban3
I-TV- L^1	0.16	0.16	0.12	0.37	0.14	0.64	0.41	0.91
A-Huber- L^1	0.14	0.16	0.09	0.34	0.14	0.55	0.40	0.48
A-Hub.- L^1 SY	n.a.	0.33	0.13	n.a.	0.14	0.61	0.77	0.70

Table 1: The average end-point error (AEPE) on the Middlebury training sequences for isotropic TV- L^1 (I-TV- L^1), anisotropic Huber- L^1 (A-Huber- L^1), and anisotropic Huber- L^1 with symmetry constraint (A-Hub.- L^1 SY)

seconds on the Urban sequence (640x480) on a recent 64bit Linux system, equipped with an Intel Core 2 Quad at 2.66 GHz and an Nvidia GTX 280 GPU. Enabling the symmetry condition increases the amount of data to be processed, hence the calculation time increases to 1.23 seconds on Urban.

6 Conclusion and Future Work

In this paper we illustrated the benefits of an image-driven, discontinuity preserving regularization term over the widely used isotropic TV regularization. Moreover, we demonstrated that the staircasing effect in weakly textured areas can be reduced substantially by using the Huber norm instead of the TV. We also demonstrated in which situations the spatio-temporal regularization approach introduced in [14] fails and proposed a more robust alternative. Due to the promising results so far, we plan to extend this spatio-temporal regularization from the current ‘linear parametric’ model to higher-order polynomials.

7 Acknowledgement

This work was partly supported by the Austrian Research Promotion Agency (FFG) within the vdQA Project (816003).

References

- [1] J. F. Aujol, G. Gilboa, T. Chan, and S. J. Osher. Structure-texture image decomposition: Modeling, algorithms, and parameter selection. *International Journal of Computer Vision*, 67(1):111–136, 2006.
- [2] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *Proceedings 13th International Conference on Computer Vision*, 2007.
- [3] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27(3):433–466, September 1995.
- [4] M. J. Black and P. Anandan. Robust dynamic motion estimation over time. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pages 296–302, June 1991.
- [5] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1-2):89–97, 2004.

- [6] D. J. Fleet and Y. Weiss. Optical flow estimation. In N. Paragios, Y. Chen, and O. Faugeras, editors, *Handbook of Mathematical Models in Computer Vision*, chapter 15, pages 239–258. Springer, 2006.
- [7] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17: 185–203, 1981.
- [8] P. J. Huber. Robust regression: Asymptotics, conjectures and Monte Carlo. *Annals of Statistics*, 1(5):799–821, 1973.
- [9] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 674–679, April 1981.
- [10] H.-H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(5):565–593, 1986.
- [11] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, 67(2):141–158, April 2006.
- [12] I. Patras, M. Worring, and R. van den Boomgaard. Dense motion estimation using regularization constraints on local parametric models. *IEEE Transactions on Image Processing*, 13(11):1432–1443, November 2004.
- [13] R. T. Rockafellar. *Convex analysis*. Princeton Landmarks in Mathematics. Princeton University Press, Princeton, NJ, 1997. ISBN 0-691-01586-4. Reprint of the 1970 original, Princeton Paperbacks.
- [14] L. Rudin, S. J. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
- [15] D. Shulman and J.-Y. Hervé. Regularization of discontinuous flow fields. In *Proceedings Workshop on Visual Motion*, pages 81–86, 1989.
- [16] A. Wedel, T. Pock, J. Braun, U. Franke, and D. Cremers. Duality TV-L1 flow with fundamental matrix prior. In *Proceedings Image and Vision Computing New Zealand*, November 2008.
- [17] J. Weickert and C. Schnörr. Variational optic flow computation with a spatio-temporal smoothness constraint. *Journal of Mathematical Imaging and Vision*, 14(3):245–255, 2001.
- [18] J. Weickert, A. Bruhn, T. Brox, and N. Papenberg. A survey on variational optic flow methods for small displacements. In O. Scherzer, editor, *Mathematical Models for Registration and Applications to Medical Imaging*, pages 103–136. Springer, Berlin, 2006.
- [19] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV-L1 optical flow. In *Pattern Recognition*, volume 4713 of *LNCS*, pages 214–223, 2007.