ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

Optical Flow Estimation via Steered-L^1 Norm

Ammar Zayouna
School of Science and Technology
Middlesex University
London NW4 4BT, UK
a.zayouna@mdx.ac.uk

Richard Comley
School of Science and Technology
Middlesex University
London NW4 4BT, UK
r.comley@mdx.ac.uk

Daming Shi
School of Science and Technology
Middlesex University
London NW4 4BT, UK
d.shi@mdx.ac.uk

ABSTRACT

Global variational methods for estimating optical flow are among the best performing methods due to the sub-pixel accuracy and the 'fill-in' effect they provide. The fill-in effect allows optical flow displacements to be estimated even in low and untextured areas of the image. The estimation of such displacements are induced by the smoothness term. The L^1 norm provides a robust regularisation term for the optical flow energy function with a very good performance for edge-preserving. However this norm suffers from several issues, among these is the isotropic nature of this norm which reduces the fill-in effect and eventually the accuracy of estimation in areas near motion boundaries. In this paper we propose an enhancement to the L^1 norm that improves the fill-in effect for this smoothness term. In order to do this we analyse the structure tensor matrix and use its eigenvectors to steer the smoothness term into components that are 'orthogonal to' and 'aligned with' image structures. This is done in primal-dual formulation. Results show a reduced end-point error and improved accuracy compared to the conventional L^1 norm.

Keywords

Optical flow, Variational methods, $TV-L^1$, Structure tensor.

1 INTRODUCTION

Optical flow is an important cue in image processing applications. It can be defined as the estimation of image point displacements over time [1], [2]. Such images are taken for the same scene at successive moments in time. Methods for finding optical flow can be classified in many ways. An early classification can be found in the work of Barron et al. [3], which classified optical flow algorithms into four main groups. One of these groups relies on the computation of optical flow using the calculus of variations and is thus denoted as 'Variational methods'. Variational methods for estimating optical flow belong to the highest-accuracy methods. These methods find the optical flow displacement field by minimising an energy function mainly comprising data and smoothness terms:

$$E = \alpha E_{data} + E_{smooth} \tag{1}$$

where α controls the weight between the two terms. The data term E_{data} is based on the brightness constancy assumption, where it is assumed that illumination between images does not change over time:

$$I_2(\mathbf{x}+\mathbf{u}, t+1) = I_1(\mathbf{x}, t)$$

where $\mathbf{u} = (u, v)$ is the displacement for each pixel in the x and y directions respectively, $\mathbf{x} = (x, y)$ is the pixel coordinates, and $t \in [0, T]$ is the time reference. This term is linearised using the 'Taylor Expansion' to obtain what is known as the 'optical flow constraint' [4]:

$$I_x u + I_y v + I_t = 0 \tag{2}$$

where subscripts denote partial derivatives. The brightness constancy assumption does not always hold, as it gets violated when illumination changes between the two images, for example due to shadows and shading. The smoothness (or regularity term) on the other hand is based on the spatial constancy assumption, where it is assumed that the neighbouring pixels in the first image are still neighbours in the second image. Hence diverse displacements are penalised. This assumption also does not always hold as it gets violated in some areas such as motion boundaries, where pixels in the first image are no longer neighbours in the second image due to motion or occlusion. One example of an optical flow energy function can be found in the early work of Horn-Schunck [1], where they proposed to minimise the following energy function:

$$E = \int_\Omega \big(\overbrace{\alpha (I_x u + I_y v + I_t)^2}^{E_{data}} + \overbrace{|\nabla u, \nabla v|^2}^{E_{smooth}} \big) dxdy \tag{3}$$

where Ω is a 2D image domain and $\nabla u = (u_x, u_y)$ and $\nabla v = (v_x, v_y)$. The solution is found using the Euler-Lagrange equations, which results in a couple of simultaneous equations. The displacements then can be

easily found by solving the resulting system of equations. In that algorithm a quadratic norm is used in the smoothness term. The quadratic norm penalises the flow field severely in all directions, hence it produces blurry motion edges due to this penalisation.

The displacement field is piecewise in nature, hence a good choice for a smoothness term is a piecewise function that characterises the piecewise nature of such a displacement field. The total variation L^1 is an example of such functions. Indeed this piecewise function can characterise the flow field efficiently. However, this norm suffers from two main issues. First, it is not continuously differentiable. This issue was addressed by Chambolle [5]. Later Zach et al. [6] used this solution to propose an optical flow estimation algorithm in a primal-dual formulation. Primal-dual algorithms, in addition to their accuracy and good edge-preserving qualities (due to the use of the L^1 norm in the smoothness term), can be easily parallelised using modern graphics hardware [7], [8], [9].

The second issue is that the L^1 norm is isotropic, hence the fill-in effect [10] reduces along motion boundaries. In this paper we propose to improve L^1 by the use of eigenvectors of the local structure tensor. We derive the formulation for this in the primal-dual settings. The eigenvectors of the structure tensor were used by Zimmer et al. [11] to improve the fill-in effect using robust functions which approximate the behaviour of the L^1 norm as a smoothness term. In the current work we apply it directly to the total variation L^1 norm, which is non-trivial due to the non-continuous differentiability of the L^1 norm. Additionally we use a data term inspired by the delayed linearisation data term proposed by Brox et al. [2].

This paper is organised as follows, Section-2 examines some related work. Section-3 includes a brief introduction for the notion of 'Structure tensor'. Section-4 introduces the method. Section-5 discusses the implementation and results. Section-6 concludes this paper and proposes several enhancements to be investigated in the future.

2 RELATED WORK

Since the marquee work of Horn-Schunck [1], a lot of research has been dedicated to improve the estimation of global optical flow algorithms. Both the data and the smoothness terms have undergone a lot of improvements. The Horn-Schunck method belongs to what is known as the 'Global methods'. Lucas-Kanade [12] proposed to calculate optical flow by assuming that the displacement field is constant in a small local neighbourhood, hence this type of method was called 'Local methods'. Bruhn et al. [4] proposed to combine the global and local methods by integrating neighbouring pixels in the data term using a Gaussian filter kernel, in what is known as the Combined Local-Global (CLG) method. Brox et al. [2] proposed to delay linearisation of the data term of the optical flow equation. This enabled the computation of high accuracy displacement fields. Additionally to improve robustness several algorithms extended the data term to include image gradients, thus improving the robustness to illumination changes [2], [13], [11]. Wedel et al. [8] proposed to improve the robustness to illumination changes by introducing a structure-texture decomposition step before the minimisation.

In the smoothness term, Horn-Schunck [1] used a quadratic function as a regularisation term. The quadratic function penalises the flow field severely regardless of the flow magnitude, thus introducing blurriness across motion boundaries. To remedy this, several methods used robust functions in the smoothness term such as the Lorentzian function [14], the charbonneir [4] and the robust L^1 approximation function [4]. The total variation L^1 norm was also used as a smoothness term. The problem with the L^1 norm is that it is not continuously differentiable. Chambolle [5] proposed a numerical scheme to solve the $TV - L^1$ minimisation and applied it to image denoising and zooming. Zach el al. [6] used this scheme under primal-dual formulation minimisation to estimate optical flow. Drulea et al. [9] used this scheme to find the optical flow field and used a CLG data term. In addition to that [9] used a diffusion tensor [15] to improve the fill-in effect in low and untextured areas. However a drawback of using a diffusion tensor is that it produces over-segmentation, this is because this diffusion tensor is a function of image gradients. To improve the fill-in effect, Sun et al. [16] and later Zimmer et al. [11] analysed the image structure tensor to obtain eigenvectors, and used these eigenvectors to improve the fill-in effect. Hence the direction of penalisation is adapted to the direction of the local image structure, while the magnitude of this penalisation depends on the flow field magnitude.

Despite the recent advances in estimating optical flow fields using methods that are not variational, the variational methods are still needed. Probabilistic methods for example, despite their popularity, lack the sub-pixel accuracy of the variational methods. Hence variational methods are generally used as a final stage to refine the estimation of the displacement field [17], [18], [19], [20].

3 STRUCTURE TENSOR AND STEERING IMAGE DERIVATIVES

The structure tensor of a 2D image is a 2×2 matrix that contains information about the structure orientation

in a certain neighbourhood in that image. The initial structure tensor J_0 can be expressed as follows [21]:

$$J_0 = \nabla I \nabla I^T = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

where I is a 2D image, ∇I is the image gradients $(I_x, I_y)^\mathsf{T}$ in the x, and y directions. The structure tensor J is obtained by integrating information in a certain neighbourhood. This is done by convolving the initial structure tensor J_0 with a Gaussian filter kernel G_ρ, where ρ is the standard deviation. The structure tensor is expressed as follows:

$$J = G_\rho * J_0 \tag{4}$$

where '$*$' denotes convolution. This structure tensor can be analysed and two orthonormal eigenvectors with corresponding eigenvalues are obtained. The first eigenvector can be formulated as $(\cos\phi, \sin\phi)$ and the second as $(-\sin\phi, \cos\phi)$ [22]. The eigensystem obtained can be used to give information about the local image structures. The eigensystem can be written in the following form:

$$J\mathbf{e} = \lambda\mathbf{e}$$

where $\mathbf{e}$ are the set of eigenvectors ($\mathbf{e}_1$, $\mathbf{e}_2$...$\mathbf{e}_n$), and λ are the corresponding eigenvalues (λ_1, λ_2,... λ_n), and n is the size of the square matrix J.

The first eigenvector corresponds to the largest eigenvalue points across the dominant structure in the neighbourhood, while the second eigenvector corresponds to the smaller eigenvalue points along that structure. Figure-1 depicts two eignevectors obtained by analysing the structure tensor at a certain location. The first eigenvector is depicted here in green, while the second eigenvector is depicted in red. This image is obtained from the Middlebury dataset [23].

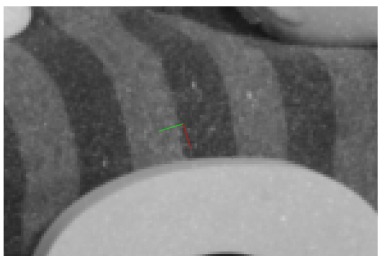

Figure 1: Eigenvectors directions of a structure tensor. The green line corresponds to the first eigenvector pointing across area with high gradient, while the red line corresponds to the second eigenvector pointing along area with high gradient.

The eigenvectors can be used to obtain what is known as the 'Steered image derivatives', where image gradients are steered from the conventional x and y direction to directions 'orthogonal to', and 'aligned with' the local image structures [22], [24]. This can be written in the following form:

$$I_o = \cos\phi . I_x + \sin\phi . I_y \tag{5}$$

$$I_a = -\sin\phi . I_x + \cos\phi . I_y \tag{6}$$

where ϕ is the angle of the first eigenvector and it is obtained via the structure tensor matrix.

4 STEERED–L^1 OPTICAL FLOW

It is desired in this paper to use the robust L^1 norm known for its edge preserving performance in the computation of an optical flow field. To this end it is required to minimise the following equation:

$$E = \int_\Omega \Big(\alpha E_{data}(I_1, I_2, \mathbf{u}) + |\nabla \mathbf{u}| \Big) dxdy \tag{7}$$

where $E_{data}(I_1, I_2, \mathbf{u})$ will be introduced later. The minimisation of this equation is non-trivial since the smoothness term used (the L^1 norm) is not continuously differentiable. Hence, following the primal-dual formulation [6], an auxiliary variable $\bar{\mathbf{u}}$ is introduced and the energy function takes the following form[1]:

$$E = \int_\Omega \Big(\alpha E_{data}(I_1, I_2, \mathbf{u}) + \frac{1}{2\theta}(\mathbf{u} - \bar{\mathbf{u}})^2 + |\nabla \mathbf{u}| \Big) \tag{8}$$

where θ is a small constant. This equation is split into a dual and a primal equation, the dual equation is written as follows:

$$E_{dual} = \int_\Omega \Big(\frac{1}{2\theta}(\mathbf{u} - \bar{\mathbf{u}})^2 + |\nabla \mathbf{u}| \Big) \tag{9}$$

and the primal equation is written as:

$$E_{primal} = \int_\Omega \Big(\alpha E_{data}(I_1, I_2, \mathbf{u}) + \frac{1}{2\theta}(\mathbf{u} - \bar{\mathbf{u}})^2 \Big) \tag{10}$$

The minimisation of this system of equations is performed in primal-dual steps. In the following subsections we discuss the primal and the dual steps in detail.

4.1 The Dual Step

The aim of the dual step is to minimise $\mathbf{u}$ while keeping $\bar{\mathbf{u}}$ fixed. We propose to steer the derivatives of the displacement fields according to the local structure. Hence Equation-9 is written as follows:

$$E_{dual} = \int_\Omega \Big(|\mathbf{e}^\mathsf{T} \nabla u| + \frac{1}{2\theta}(u - \bar{u})^2 + |\mathbf{e}^\mathsf{T} \nabla v| + \frac{1}{2\theta}(v - \bar{v})^2 \Big) \tag{11}$$

[1] Starting from this point, the notation '*dxdy*' is omitted for brevity.

where:

$$\mathbf{e}^{\mathsf{T}} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix}$$

are the eigenvectors of the structure tensor, and $\nabla u, \nabla v$ are expressed here as $(u_x, u_y)^{\mathsf{T}}$ and $(v_x, v_y)^{\mathsf{T}}$. In this way the directions of the smoothness term is adapted to the direction of the local image structure, while the magnitude of the penalisation is adapted to the flow field itself [11].

To solve the minimisation, Euler-Lagrange equations are obtained [9]. The first equation obtained is the following:

$$-div\big(\mathbf{e}^{\mathsf{T}}.\frac{\nabla u}{|\nabla u|}\big) + \frac{1}{\theta}(u-\bar{u}) = 0 \tag{12}$$

which can be re-written as follows:

$$u = \theta.div(\mathbf{e}^{\mathsf{T}}.\mathbf{p}_u) + \bar{u} \tag{13}$$

where $\mathbf{p}_u = \frac{\nabla u}{|\nabla u|}$, it follows that:

$$\mathbf{p}_u.|\nabla u| - \nabla u = 0, \quad |\mathbf{p}_u| \leq 1 \tag{14}$$

Substituting Equation-13 in Equation-14, the following equation is obtained:

$$\mathbf{p}_u.\Big|\nabla\big(div(\mathbf{e}^{\mathsf{T}}.\mathbf{p}_u) + \bar{u}/\theta\big)\Big| - \nabla\big(div(\mathbf{e}^{\mathsf{T}}.\mathbf{p}_u) + \bar{u}/\theta\big) = 0.$$

Adding $\mathbf{p}_u$ to both sides of the above equation yields the following fixed-point iteration to find $\mathbf{p}_u$:

$$\mathbf{p}_u^{k+1} = \frac{\mathbf{p}_u^k + \tau.\nabla\Big(div(\mathbf{e}^{\mathsf{T}}.\mathbf{p}_u^k) + \bar{u}/\theta\Big)}{1 + \tau.\Big|\nabla\big(div(\mathbf{e}^{\mathsf{T}}.\mathbf{p}_u^k) + \bar{u}/\theta\big)\Big|} \tag{15}$$

where k is the iteration count, and τ is the step size. In the same way $\mathbf{p}_v$ can be obtained, and it is calculated using the following fixed-point iteration:

$$\mathbf{p}_v^{k+1} = \frac{\mathbf{p}_v^k + \tau.\nabla\Big(div(\mathbf{e}^{\mathsf{T}}.\mathbf{p}_v^k) + \bar{v}/\theta\Big)}{1 + \tau.\Big|\nabla\big(div(\mathbf{e}^{\mathsf{T}}.\mathbf{p}_v^k) + \bar{v}/\theta\big)\Big|}. \tag{16}$$

The terms $\mathbf{e}^{\mathsf{T}}.\mathbf{p}_u^k$, and $\mathbf{e}^{\mathsf{T}}.\mathbf{p}_v^k$ can be replaced by the alternative notations $\mathbf{p}_{su}$ and $\mathbf{p}_{sv}$, where:

$$\mathbf{p}_{su} = \mathbf{e}^{\mathsf{T}}.[p_{1u},\ p_{2u}]^{\mathsf{T}} \tag{17}$$

$$\mathbf{p}_{sv} = \mathbf{e}^{\mathsf{T}}.[p_{1v},\ p_{2v}]^{\mathsf{T}} \tag{18}$$

Hence, Equation-15 and Equation-16 can be written in the following way:

$$\mathbf{p}_u^{k+1} = \frac{\mathbf{p}_u^k + \tau.\nabla\Big(div\ \mathbf{p}_{su} + \bar{u}/\theta\Big)}{1 + \tau.\Big|\nabla\big(div\ \mathbf{p}_{su} + \bar{u}/\theta\big)\Big|}. \tag{19}$$

$$\mathbf{p}_v^{k+1} = \frac{\mathbf{p}_v^k + \tau.\nabla\Big(div\ \mathbf{p}_{sv} + \bar{v}/\theta\Big)}{1 + \tau.\Big|\nabla\big(div\ \mathbf{p}_{sv} + \bar{v}/\theta\big)\Big|}. \tag{20}$$

4.2 The Primal Step

The aim of the primal step is to minimise $\bar{\mathbf{u}}$ while keeping $\mathbf{u}$ fixed. Starting from a non-linearised data term, we propose to minimise the following data term:

$$E_{Primal} = \int_\Omega \Big(\alpha|I_2(\mathbf{x}+\mathbf{u}) - I_1(\mathbf{x})|^2 + \gamma|\nabla I_2(\mathbf{x}+\mathbf{u}) - \nabla I_1(\mathbf{x})|^2 + \frac{1}{2\theta}(\mathbf{u}-\bar{\mathbf{u}})^2\Big) \tag{21}$$

where α here is the weight of the data term, and γ is the weight of the image gradient term which is added here to improve the robustness to illumination changes. This robustness can be further improved using the robust function $\Psi(s^2) = \sqrt{s^2+\varepsilon^2}$, where ε is a small constant.

$$E_{Primal} = \int_\Omega \Big(\alpha\ \Psi\big(|I_2(\mathbf{x}+\mathbf{u}) - I_1(\mathbf{x})|^2\big) + \gamma\ \Psi\big(|\nabla I_2(\mathbf{x}+\mathbf{u}) - \nabla I_1(\mathbf{x})|^2\big) + \frac{1}{2\theta}(\mathbf{u}-\bar{\mathbf{u}})^2\Big). \tag{22}$$

The second image $I_2(\mathbf{x}+\mathbf{u})$ can be written in the following form using the Taylor Expansion:

$$I_2(\mathbf{x}+\mathbf{u}) = I_2(\mathbf{x}+\mathbf{u_0}) + (\mathbf{u}-\mathbf{u_0})\nabla I_2(\mathbf{x}+\mathbf{u_0}) \tag{23}$$

where $\mathbf{u_0} = (u_0, v_0)$ is the initial displacement of the flow field. Similarly image gradients can be linearised, and the image gradients term can be rewritten as follows:

$$I_{2x}(\mathbf{x}+\mathbf{u}) = I_{2x}(\mathbf{x}+\mathbf{u_0}) + (\mathbf{u}-\mathbf{u_0})\nabla I_{2x}(\mathbf{x}+\mathbf{u_0}). \tag{24}$$

$$I_{2y}(\mathbf{x}+\mathbf{u}) = I_{2y}(\mathbf{x}+\mathbf{u_0}) + (\mathbf{u}-\mathbf{u_0})\nabla I_{2y}(\mathbf{x}+\mathbf{u_0}). \tag{25}$$

Plugging all these terms together yield the following equation:

$$E_{primal} = \int_\Omega \alpha\ \Psi\Big(\big|I_{t0} + (\mathbf{u}-\mathbf{u_0})\nabla I_2\big|^2\Big) + \gamma\ \Psi\Big(\big|\big(I_{tx} + (\mathbf{u}-\mathbf{u_0})\nabla I_{2x}\big), \big(I_{ty} + (\mathbf{u}-\mathbf{u_0})\nabla I_{2y}\big)\big|^2\Big)\Big) + \frac{1}{2\theta}(\mathbf{u}-\bar{\mathbf{u}})^2 \tag{26}$$

where the notation $(\mathbf{x}+\mathbf{u_0})$ was omitted from $\nabla I_2, \nabla I_{2x}, \nabla I_{2y}$ for brevity. The solution requires the minimisation of $E(\bar{\mathbf{u}})$. Hence Equation-26 is written as follows:

$$E_{primal} = \int_\Omega \alpha\ \Psi\Big(\big|I_{t0} + (\bar{\mathbf{u}}-\mathbf{u_0})\nabla I_2\big|^2\Big) + \gamma\ \Psi\Big(\big|\big(I_{tx} + (\bar{\mathbf{u}}-\mathbf{u_0})\nabla I_{2x}\big), \big(I_{ty} + (\bar{\mathbf{u}}-\mathbf{u_0})\nabla I_{2y}\big)\big|^2\Big)\Big) + \frac{1}{2\theta}(\mathbf{u}-\bar{\mathbf{v}})^2 \tag{27}$$

with the following abbreviation used:

$$\begin{aligned} I_{t0} &= I_2(\mathbf{x}+\mathbf{u_0}) - I_1(\mathbf{x}) \\ I_{tx} &= \frac{\partial}{\partial x} I_2(\mathbf{x}+\mathbf{u_0}) - \frac{\partial}{\partial x} I_1(\mathbf{x}) \\ I_{ty} &= \frac{\partial}{\partial y} I_2(\mathbf{x}+\mathbf{u_0}) - \frac{\partial}{\partial y} I_1(\mathbf{x}) \end{aligned} \tag{28}$$

The minimisation of Equation-27 can be easily performed by setting the derivatives of the equation with respect to $\bar{u}, \bar{v}$ equal to zero. This leads to the following set of equations:

$$\begin{aligned}&\big[\alpha\,\Psi_1^{'}.I_{2x}^2+\gamma\,\Psi_2^{'}(I_{2xx}^2+I_{2yx}^2)+\frac{1}{\theta}\big]\bar{u}\\&+\big[\alpha\,\Psi_1^{'}.I_{2x}I_{2y}+\gamma\,\Psi_2^{'}.I_{2xy}(I_{2xx}+I_{2yy})\big]\bar{v}\\&=-\big[\alpha\,\Psi_1^{'}r_{t0}I_{2x}+\gamma\,\Psi_2^{'}r_{tx0}I_{2xx}+\gamma\,\Psi_2^{'}r_{ty0}I_{2xy}\big]+\frac{u}{\theta}.\end{aligned} \tag{29}$$

Similarly derivation with respect to $\bar{v}$ yields the following equation:

$$\begin{aligned}&\big[\alpha\,\Psi_1^{'}.I_{2x}I_{2y}+\gamma\,\Psi_2^{'}I_{2yx}(I_{2xx}+I_{2yx})\big]\bar{u}\\&+\big[\alpha\,\Psi_1^{'}.I_{2y}^2+\gamma\,\Psi_2^{'}.(I_{2xy}^2+I_{2yy}^2)\big]\bar{v}\\&=-\big[\alpha\,\Psi_1^{'}r_{t0}I_{2y}+\gamma\,\Psi_2^{'}r_{tx0}I_{2xy}+\gamma\,\Psi_2^{'}r_{ty0}I_{2yy}\big]+\frac{v}{\theta}\end{aligned} \tag{30}$$

where $\Psi^{'}$ is the derivative of Ψ. Ψ_1 and Ψ_2 are defined as follows:

$$\Psi_1=\Psi\Big(\big|I_{t0}+(\bar{\mathbf{u}}-\mathbf{u_0})\nabla I_2\big|^2\Big)$$

$$\Psi_2=\Psi\big(\big|\big(I_{tx}+(\bar{\mathbf{u}}-\mathbf{u_0})\nabla I_{2x}\big),\big(I_{ty}+(\bar{\mathbf{u}}-\mathbf{u_0})\nabla I_{2y}\big)\big|^2\big)$$

The values of $r_{t0},\ r_{tx0},\ r_{ty0}$ are given as follows:

$$\begin{aligned}r_{t0}&=I_{t0}-u_0I_{2x}-v_0I_{2y}\\r_{tx0}&=I_{tx}-u_0I_{2xx}-v_0I_{2xy}\\r_{ty0}&=I_{ty}-u_0I_{2xy}-v_0I_{2yy}\end{aligned} \tag{31}$$

4.3 Colour Image Realisation

The algorithm discussed so far can work on grey-scale images. Colour images offer richer photometric information compared to gray-scale images [25], [26]. It is possible to extend this algorithm to work with colour images. In RGB images, which is an additive colour model, colour is encoded in three channels (Red, Green and Blue). In order to be able to extend the work in our algorithm to colour images, the primal step is extended to incorporate the three colour channels. Hence Equation-29 and Equation-30 are written in the following form:

$$\begin{aligned}&\big[\alpha\,(\Psi_1^c)^{'}.(I_{2x}^c)^2+\gamma\,(\Psi_2^c)^{'}((I_{2xx}^c)^2+(I_{2yx}^c)^2)+\frac{1}{\theta}\big]\bar{u}^c\\&+\big[\alpha\,(\Psi_1^c)^{'}.I_{2x}^cI_{2y}^c+\gamma\,(\Psi_2^c)^{'}.I_{2xy}^c(I_{2xx}^c+I_{2yy}^c)\big]\bar{v}^c\\&=-\big[\alpha\,(\Psi_1^c)^{'}r_{t0}^cI_{2x}^c+\gamma\,(\Psi_2^c)^{'}r_{tx0}^cI_{2xx}^c+\gamma\,(\Psi_2^c)^{'}r_{ty0}^cI_{2xy}^c\big]+\big(\frac{u}{\theta}\big)^c\end{aligned} \tag{32}$$

$$\begin{aligned}&\big[\alpha\,(\Psi_1^c)^{'}.I_{2x}^cI_{2y}^c+\gamma\,(\Psi_2^c)^{'}I_{2yx}^c(I_{2xx}^c+I_{2yx}^c)\big]\bar{u}^c\\&+\big[\alpha\,(\Psi_1^c)^{'}.(I_{2y}^c)^2+\gamma\,(\Psi_2^c)^{'}.((I_{2xy}^c)^2+(I_{2yy}^c)^2)\big]\bar{v}^c\\&=-\big[\alpha\,(\Psi_1^c)^{'}r_{t0}^cI_{2y}^c+\gamma\,(\Psi_2^c)^{'}r_{tx0}^cI_{2xy}^c+\gamma\,(\Psi_2^c)^{'}r_{ty0}^cI_{2yy}^c\big]+\big(\frac{v}{\theta}\big)^c\end{aligned} \tag{33}$$

where $c\in\{c_1,c_2,c_3\}$ are the three colour channels in the RGB colour model. The values of $\mathbf{u}$ and $\bar{\mathbf{u}}$ are replicated at each iteration to cope with the three colour channels, and thus to obtain $\mathbf{u}^c$ and $\bar{\mathbf{u}}^c$. Additionally the values of $\mathbf{u}^c$ and $\bar{\mathbf{u}}^c$ are averaged before starting the dual step (see Algorithm-1).

4.4 Extended Intermediate filtering

Median filtering is used in optical flow algorithms to improve the computation of the displacement fields. The use of median filtering was found especially to be useful in the algorithms following the primal-dual formulation [8]. A Median filter is applied in each warp to the estimated flow field u, v to remove outliers. Median filters work by replacing the value of a certain pixel with the median pixel value in a certain neighbourhood. One can say that the use of median filters in this case encourages smoother solution (i.e. without outliers) in the estimated flow field. In this context we propose to extend the intermediate filtering by adding another filtering step. In the intermediate filtering stage we opt to use a bi-lateral filter [27] in addition to the median filter. Bi-lateral filters are known to have a good edge preserving performance. Unlike the Gaussian filter, the bi-lateral filter changes weight according to spatial distance and the colour (or intensity) difference.

$$I_f(x_i)=\frac{1}{K}\sum_{x\in\Omega_n}g(x_i-x)s(I(x_i)-I(x)).I(x) \tag{34}$$

where I is the original image, I_f is the filtered version of the image, K is a normalising term, Ω_n is the neighbourhood region, $g(x_i-x)$ is the kernel determining the weight based on spatial distance (which can be Gaussian), and $s(I(x_i)-I(x))$ is the kernel determining the weight based on colour difference.

The intermediate filtering proposed here is a two stage filtering that includes both the median and bi-lateral filters (see Algorithm-1). We call the new intermediate filtering 'Extended Intermediate Filtering' (EIF). Applying this filtering in each warp was found to improve the accuracy of the optical flow computation.

5 EXPERIMENTS

5.1 Implementation

The algorithm is written in MATLAB. Since this algorithm is variational, it can only detect small displacements. Hence the minimisation is performed in a Coarse-to-Fine (C2F) framework. To this end, the image sequence is downscaled several times to obtain a pyramid of images. The optical flow is first found in the coarsest version of the image sequence, the estimated flow is then propagated to the next finer layer and used as an initialisation for the solution in that layer. At

each layer, the second image is warped towards the first image using the flow estimated in the previous pyramid layer. A fine resolution pyramid is chosen for the minimisation [2] with 80 layers and downscaling ratio[2] of 0.95. Image and flow field resize is performed via bicubic interpolation. Image gradients are obtained via the kernel $[-1,\ 9,\ -45,\ 0,\ 45,\ -9,\ 1]/(60)$ [4]. The divergence and derivative for the variable $\mathbf{p}$ are approximated using the three point kernel $[-1,\ 0,\ 1]$. At each layer the second image and its derivatives are warped six times. At each layer the structure tensor is computed, with image derivatives computed using a 5×5 optimised derivative filter D [22]:

$$D = (0.0234, 0.2415, 0.4700, 0.2415, 0.0234)^{\mathsf{T}} * (0.0838, 0.3323, 0, -0.3323, -0.0838) \quad (35)$$

After calculating the structure tensor, eigen-decomposition is performed to find the two eigenvectors. To obtain $(div\ \mathbf{p}_{su})$ and $(div\ \mathbf{p}_{sv})$ the following filter kernels are used [29]:

$$h_x = \frac{1}{32}\begin{bmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{bmatrix} \quad (36)$$

$$h_y = \frac{1}{32}\begin{bmatrix} -3 & 10 & -3 \\ 0 & 0 & 0 \\ 3 & 10 & 3 \end{bmatrix} \quad (37)$$

The minimisation of the aforementioned formulation is done in primal-dual and C2F frameworks [6]. The parameters of the algorithm were set to the following values ($\alpha = 1/4700, \gamma = 1, \tau = 1/10, \theta = 1/10$, $\varepsilon = 0.001$). The minimisation pseudo-code is depicted in Algorithm-1.

Algorithm 1: Implementation Algorithm of steered-L^1 norm.

Input: Images I_1 & I_2,
number of pyramid layers $L = 80$, current layer l
Create pyramid of images with L layer, and a downscale ratio of 0.95;
initialization;
l=1;
initialise (u_l, v_l) to $(\mathbf{0}, \mathbf{0})$;
while $l \leq L$ **do**
  Up-scale size of (u_{l-1}, v_{l-1}) to (u_l, v_l);
  Find eigenvectors of the structure tensor of I_1;
  while *No. of Warps* $\leq$ *6* **do**
    Warp I_{2l}, I_{2x}, I_{2y} towards I_{1l}, I_{1x}, I_{1y};
    while *No. of iterations* $\leq$ *20* **do**
      Replicate the terms and update the auxiliary variable $\bar{u}^c, \bar{v}^c$ by solving the simultaneous equations (Equatin-32, 33) ;
      Average $\bar{u}^c, \bar{v}^c$ to yield $\bar{u}, \bar{v}$;
      Update u_l using Equation-13, and similarly update v_l;
      Calculate p_{su}, p_{sv} via (Equation-17, 18);
      Update p_u, p_v (Equation-19, 20);
    Apply median filter to u_l, v_l;
    Apply bi-later filter to u_l, v_l;

Output: (Displacement field (u, v))

5.2 Results

Several datasets are available to use for assessment of optical flow methods [23], [30]. In this section we use some of these datasets to assess the work of our algorithm. We highlight the improvement that the steered-L^1 norm introduces over the use of the L^1 norm.

5.2.1 Middlebury Dataset

The Middlebury dataset has been used in assessing the performance of optical flow algorithms for many years [23]. It contains synthetic and non-synthetic image sequences, and it includes some image sequences with known ground truth which can be used for training. To examine the performance differences that the steered-L^1 has made to the accuracy of optical flow estimation, optical flow displacement field is computed for the eight sequences that have a known ground truth, and the Average End-Point Error (AEPE) [23] is computed for these sequences. Table-1 depicts the difference in AEPE using the two norms. Table-1 shows clearly improved results obtained via using the steered-L^1 norm as the smoothness term.

Our method currently has an average rank of (57.5) on the AEPE Middlebury benchmark ranking table, and an average rank of (57.8) on the Average Angular Error AAE table. Figure-2 depicts a segment of the AEPE ranking table[3].

To further asses the performance of the proposed algorithm, we compare the results of this method with other methods sharing similar principals in the Middlebury ranking table. The first method to compare with is the improved $TV - L^1$ algorithm [8]. This algorithm follows a similar minimisation framework in a primal-dual formulation. However, it differs in the data term where a structure-texture decomposition is used to improve the robustness to illumination changes. In addition to that, there is a difference in the smoothness term where we steer it in accordance with the local structure. The second algorithm is the Large Displacement Optical Flow LDOF [28]. The third method to compare with is the CLG-TV [9], where the authors use a combined local-global data term (CLG). Additionally, the authors in this paper use an anisotropic diffusion filter to improve the fill-in effect. Table-2 illustrates the

[2] In general the number of layers in the pyramid can be chosen such that the discrete derivative filter kernel can be applied at the coarsest layer [28]. However 80 layers was enough to give good results in our experiments.

[3] `http://vision.middlebury.edu/flow/eval/`

	RubberWhale	Dimetrodon	Urban2	Urban3	Venus	Grove2	Grove3	Hydrangea	Average
L^1	0.09	0.15	0.55	0.49	0.32	0.18	0.57	0.17	0.32
Steered-L^1	0.08	0.14	0.53	0.46	0.31	0.17	0.57	0.16	0.30

Table 1: AEPE for colour images of the Middlebury dataset depicting the difference between the total variation L^1 and the steered-L^1 smoothness terms.

Average endpoint error	avg. rank	Army (Hidden texture) GT im0 im1 all	disc	untext	Mequon (Hidden texture) GT im0 im1 all	disc	untext	Schefflera (Hidden texture) GT im0 im1 all	disc	untext	Wooden (Hidden texture) GT im0 im1 all	disc	untext
EpicFlow [103]	55.0	0.12 74	0.36 88	0.09 60	0.25 62	0.85 70	0.21 72	0.39 54	1.00 63	0.25 66	0.19 55	1.01 64	0.11 51
ComplOF-FED-GPU [35]	56.1	0.11 54	0.29 58	0.10 78	0.21 30	0.78 58	0.14 18	0.32 42	0.79 46	0.17 15	0.19 55	0.99 63	0.11 51
Classic++ [32]	57.4	0.09 30	0.25 36	0.07 15	0.23 51	0.78 58	0.19 53	0.43 60	1.00 63	0.22 57	0.20 61	1.11 72	0.10 46
Steered-L1 [120]	57.5	0.09 30	0.22 16	0.08 42	**0.14** 1	0.49 2	0.12 5	0.28 24	0.69 32	0.16 12	0.18 50	1.06 67	0.09 25
HBM-GC [106]	57.9	0.14 89	0.28 51	0.12 91	0.26 68	0.69 37	0.22 75	0.34 45	0.75 39	0.22 57	0.21 67	0.77 28	0.15 75
Aniso. Huber-L1 [22]	58.5	0.10 44	0.28 51	0.08 42	0.31 81	0.88 75	0.28 86	0.56 79	1.13 72	0.29 78	0.20 61	0.92 58	0.13 65
TF+OM [100]	59.3	0.10 44	0.26 42	0.07 15	0.22 37	0.66 30	0.19 53	0.36 49	0.78 44	0.39 86	0.20 61	0.89 53	0.13 65
Average endpoint error	**avg. rank**	**Grove (Synthetic) GT im0 im1 all**	**disc**	**untext**	**Urban (Synthetic) GT im0 im1 all**	**disc**	**untext**	**Yosemite (Synthetic) GT im0 im1 all**	**disc**	**untext**	**Teddy (Stereo) GT im0 im1 all**	**disc**	**untext**
EpicFlow [103]	55.0	0.89 64	1.31 74	0.69 55	0.53 58	1.31 53	0.34 49	0.10 4	**0.11** 1	0.17 13	0.67 54	1.43 57	0.87 59
ComplOF-FED-GPU [35]	56.1	0.89 64	1.29 66	0.73 63	1.25 90	1.74 87	0.64 85	0.14 47	0.13 32	0.30 80	0.64 51	1.50 61	0.83 53
Classic++ [32]	57.4	0.87 57	1.30 68	0.66 53	0.47 42	1.62 76	0.33 43	0.17 81	0.14 54	0.32 90	0.79 75	1.64 71	0.92 65
Steered-L1 [120]	57.5	0.89 64	1.24 55	0.91 76	1.71 106	1.68 81	0.94 101	0.26 112	0.18 92	0.71 115	1.06 90	1.80 84	1.64 98
HBM-GC [106]	57.9	0.67 29	0.97 26	0.52 35	0.63 70	0.81 7	0.44 65	0.22 105	0.19 103	0.36 96	0.54 38	1.21 46	0.78 48
Aniso. Huber-L1 [22]	58.5	0.84 54	1.20 52	0.70 58	0.39 23	1.23 47	0.28 18	0.17 81	0.15 64	0.27 64	0.64 51	1.36 50	0.79 49
TF+OM [100]	59.3	0.98 82	1.31 74	1.03 85	0.56 62	1.55 73	0.33 43	0.16 72	0.17 82	0.27 64	0.76 67	1.59 69	0.98 72

Figure 2: A segment of the Middleburry ranking table for end-point error. The proposed method is denoted as 'Steered-L^1'.

	Average rank	Army	Mequon	Schefflera	Wooden	Grove	Urban	Yosemite	Teddy
LDOF	80.5	$0.12_{(74)}$	$0.23_{(84)}$	$0.43_{(60)}$	$0.45_{(98)}$	$1.01_{(86)}$	$1.10_{(86)}$	$0.12_{(27)}$	$0.94_{(86)}$
CLG-TV	69.5	$0.11_{(54)}$	$0.32_{(84)}$	$0.55_{(77)}$	$0.25_{(78)}$	$0.92_{(71)}$	$0.47_{(42)}$	$0.17_{(81)}$	$0.74_{(65)}$
Improved $TV-L^1$	63.8	$0.09_{(30)}$	$0.20_{(28)}$	$0.53_{(73)}$	$0.21_{(67)}$	$0.90_{(67)}$	$1.51_{(101)}$	$0.18_{(88)}$	$0.73_{(62)}$
Steered-L^1	57.5	$0.09_{(30)}$	$0.14_{(1)}$	$0.28_{(24)}$	$0.18_{(50)}$	$0.89_{(64)}$	$1.71_{(106)}$	$0.26_{(112)}$	$1.06_{(90)}$

Table 2: AEPE comparison for four algorithms including the algorithm proposed in this paper. Numbers in brackets indicate the ranking of the specific image sequence results, for example the results of the 'Mequon' sequence of our algorithm is ranked first. Numbers in blue indicate the highest rank.

AEPE for those methods at the time of writing this paper, the AEPE values are copied directly from the ranking table. Numbers between brackets indicate the ranking position for the particular image sequence.

It can be noticed that our algorithm performs better on non-synthetic image sequences. This can be seen clearly by comparing the AEPE and rank of individual non-synthetic image sequences like 'Mequon' (ranked 1st.) and 'Army' (ranked 30th.) in comparison with 'Urban'(ranked 106th.) and 'Yosemite'(ranked 112th.).

The estimated optical flow can be qualitatively assessed by visualising the displacement field. The colour code used in this paper is depicted in Figure-3 [23], where the direction of displacements is coded by the hue, and the magnitude of the displacements is coded by the saturation. Figure-4 depicts several examples for colour-coded results of the estimated optical flow field alongside their colour-coded ground truth for comparison.

5.2.2 *MPI-Sintel Dataset*

MPI-Sintel [30] is another dataset used to test optical flow methods. It is a synthetic image sequence taken from an animated 3D short film, it contains complex motion with varied textures. Images used in this

Figure 3: Colour code used to visualise the optical flow displacement fields.

dataset are rendered in three 'passes'. The first pass is 'albedo' which is the simplest rendering and does not contain illumination effect and has a piecewise constant colour. Hence, the data (brightness) constancy assumptions holds across the whole image. The second pass is the 'clean' pass which includes illumination effects (e.g. shading, specular reflections). The final pass is the one that matches the 'final' version of the film. This pass includes more complex effects and adds motion blur, atmospheric effect, colour correction, etc. This dataset is more challenging due to the inclusion of large motion and occlusion.

It was reported in [30] that methods with high-ranking on the Middlebury dataset have more difficulty estimating optical flow on this dataset. For example on the Middlebury ranking table the method 'Improved-TV-L^1' has an average rank of 63.8, which is much higher

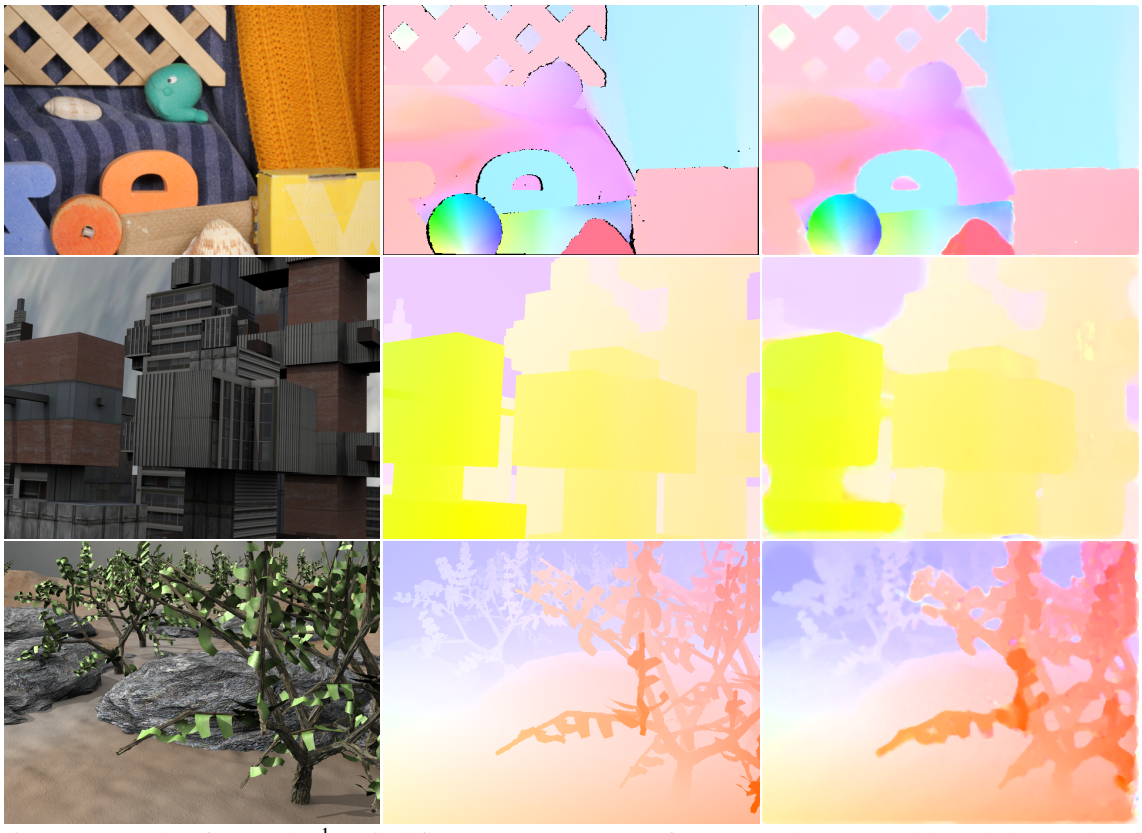

Figure 4: Results of Steered-L^1 optical flow. Image sequences from top to bottom 'RubberWhale', 'Urban3', 'Grove3'. Left column: First frame of the image sequence taken from the Middlebury dataset. Middle Column: Colour-coded ground truth. Right column: Colour-coded results.

than the ranking for LDOF [28] which has an average rank of 80.5. However, the same methods have an opposite order in the MPI-Sintel table, where LDOF is ranked higher than the Improved-TV-L^1. In this subsection we try to examine the effect of different levels of rendering on the estimation of optical flow using the proposed method in this paper. To this end the algorithm is applied to several image sequences at the clean and the final passes. The results of the estimation are compared both qualitatively and quantitatively. Table-3 illustrates the results of applying the algorithm for several examples from the training dataset of the MPI-Sintel benchmark.

Image clip	clean	final
alley_1	0.18	0.19
ambush_5	1.72	2.78
bamboo_1	0.23	0.23
Average	0.71	1.07

Table 3: AEPE for selected frames from MPI-Sintel dataset.

The AEPE results of our method now on the MPI-Sintel is 10.864 for the clean pass and 12.277 for the final pass. The relatively high AEPE in the case of this benchmark rate can be attributed to the complex motion and the effects included in this sequence, such as shading, specular reflection, motion blur, etc. The 'staircasing' effect, which is induced by the use of the L^1 norm, also contributes to the error rate. To deal with these issues and to improve the performance of this algorithm several suggestions are discussed in Section-6.

6 CONCLUSION AND FUTURE WORK

In this paper we have introduced a modified total variation L^1 norm to estimate optical flow denoted as 'Steered-L^1', which can be used to enhance the fill-in effect and hence the estimation accuracy in algorithms following the primal-dual formulation. In the proposed algorithm, the eigenvectors of the structure tensor are used to steer the displacement field derivatives into two components, one orthogonal and the other parallel to the local structure. This improves the fill-in performance of the total variation L^1 optical flow which reduces the error rate and improves the accuracy of estimation. It was shown experimentally that the utilisation of this steered norm improves the performance of the optical flow estimation and decreases the error in computation. Additionally, a high accuracy data term is used in the spirit of the delayed linearisation data constancy term proposed by Brox et al. [2]. This data term is augmented with image gradient to improve

Figure 5: Results on selected frames of MPI-Sintel. [30] Displacement field computed between the first frame (frame_0001) and the second (frame_0002). Top to bottom rows: alley_1, ambush_5, bamboo_1. Left column to right: First frame (frame_0001), ground truth, clean, final.

the robustness to illumination changes. Moreover an 'Extended Intermediate Filtering' EIF is proposed to enhance the displacement field estimation.

Despite the improved performance that this algorithm provides, it still suffers from a high AEPE in some scenarios. Further improvements are being investigated to render a more accurate optical flow estimation. In the next step of this research, other colour spaces are to be investigated such as the HSV (hue, saturation and value) colour space. The different channels of the HSV offer a more robust photometric performance [11]. This is expected to help deal with the illumination effects such as shadows and shading, and improve the performance especially in test benchmarks that include many such effects (e.g. MPI-Sintel).

The total variation L^1 is a piecewise constant function, hence it encourages a piecewise smooth solution for the displacement field. This produces artificial boundaries in the estimated optical flow field, a phenomena denoted as the 'stair-casing effect' [31]. In relatively smooth areas the performance of the quadratic norm is superior to the L^1 norm. To address this issue we are going to investigate the proposed method using the 'Huber-L^1' norm [31]. The Huber-L^1 norm behaves as the L^1 norm in areas with high gradients (i.e. motion boundaries), and behaves as a quadratic L^2 norm in the areas with lower gradients, hence offering enhanced performance [31], [32]. It will be interesting to investigate the performance of steering the Huber-L^1 norm using the structure tensor as was done with the L^1 in this paper.

Algorithms following the primal-dual formulation are generally used if speed of implementation is needed. These algorithms enable easy parallelisation on graphical hardware [9], [6], [7]. The current algorithm was implemented in MATLAB and the speed of implementation was not of concern. For example the time needed to estimate the optical flow displacement field for a colour image sequence of size 380×420 is around 457 seconds. A C/C++ and parallel implementation of this algorithm is going to be investigated in the future.

7 REFERENCES

[1] Horn, B. K. P. and Schunck, B. G. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[2] Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. High accuracy optical flow estimation based on a theory for warping, May 2004.

[3] Barron, J. L., Fleet, D. J., and Beauchemin, S. S. Performance of optical flow techniques. *InternationalL Journal of Computer Vision*, 12:43–77, 1994.

[4] Bruhn, A., Weickert, J., and Schnörr, C. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61:211–231, 2005.

[5] Chambolle, A. An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20(1-2):89–97, January 2004.

[6] Zach, C., Pock, T., and Bischof, H. A duality based approach for realtime tv-l1 optical flow. In *Proceedings of the 29th DAGM conference on Pattern recognition*, pages 214–223, Berlin, Heidelberg, 2007. Springer-Verlag.

[7] Werlberger, M., Pock, T., and Bischof, H. Motion estimation with non-local total variation regularization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, USA, June 2010.

[8] Wedel, A., Pock, T., Zach, C., Bischof, H., and Cremers, D. *Statistical and Geometrical Approaches to Visual Motion Analysis: International Dagstuhl Seminar, Dagstuhl Castle, Germany, July 13-18, 2008. Revised Papers*, chapter An Improved Algorithm for TV-L 1 Optical Flow, pages 23–45. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[9] Drulea, M. and Nedevschi, S. Total variation regularization of local-global optical flow. In *Intelligent Transportation Systems (ITSC), 2011 14th*

International IEEE Conference on, pages 318–323, Oct 2011.

[10] Bruhn, A. and Weickert, J. Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In *10th IEEE International Conference on Computer Vision (ICCV 2005), 17-20 October 2005, Beijing, China*, pages 749–755, 2005.

[11] Zimmer, H., Bruhn, A., and Weickert, J. Optic flow in harmony. *Int. Journal Comput. Vision*, 93(3):368–388, July 2011.

[12] Lucas, B. D. and Kanade, T. An iterative image registration technique with an application to stereo vision (ijcai). In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 674–679, April 1981.

[13] Papenberg, N., Bruhn, A., Brox, T., Didas, S., and Weickert, J. Highly accurate optic flow computation with theoretically justified warping. *Int. J. Comput. Vision*, 67(2):141–158, April 2006.

[14] Black, M. J. and Anandan P. A framework for the robust estimation of optical flow. In *IEEE ICCV*, pages 231–236, 1993.

[15] Perona, P. and Malik, J. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(7):629–639, July 1990.

[16] Sun, D., Roth, S., Lewis, J. P., and Black, M. J. *Computer Vision – ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part III*, chapter Learning Optical Flow, pages 83–97. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[17] Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazirbas, C., Golkov, V., Smagt, P. v.d., Cremers, D., and Brox, T. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, Dec 2015.

[18] Chen, Z., Jin, H., Lin, Z., Cohen, S., and Wu, Y. Large displacement optical flow from nearest neighbor fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[19] Revaud, J., Weinzaepfel, P., Harchaoui, Z., and Schmid, C. EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In *Computer Vision and Pattern Recognition*, 2015.

[20] Weinzaepfel, P., Revaud, J., Harchaoui, Z., and Schmid, C. DeepFlow: Large displacement optical flow with deep matching. In *ICCV 2013 - IEEE Intenational Conference on Computer Vision*, pages 1385–1392, Sydney, Australia, December 2013. IEEE.

[21] Brox, T., Weickert, J., Burgeth, B., and Mrázek, P. Nonlinear structure tensors. *Image Vision Comput.*, 24(1):41–55, January 2006.

[22] Roth, S. and Black, M. J. Steerable random fields. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007.

[23] Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J., and Szeliski, R. A database and evaluation methodology for optical flow. *Int. J. Comput. Vision*, 92(1):1–31, March 2011.

[24] Freeman, W. T. and Adelson, E. H. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:891–906, 1991.

[25] Mileva, Y., Bruhn, A., and Weickert, J. Illumination-robust variational optical flow with photometric invariants. In *In DAGM-Symposium, LNCS 4713*, pages 152–162, 2007.

[26] Andrews, R. J. and Lovell, B. C. Color optical flow. In *Proceedings Workshop on Digital Image Computing*, pages 135–139, 2003.

[27] Durand, F. and Dorsey, J. Fast bilateral filtering for the display of high-dynamic-range images. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '02, pages 257–266, New York, NY, USA, 2002. ACM.

[28] Brox, T. and Malik, J. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513, 2011.

[29] Scharr, H., Black, M.J., and Haussecker, H.W. Image statistics and anisotropic diffusion. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 840–847 vol.2, Oct 2003.

[30] Butler, D.J., Wulff, J., Stanley, G.B., and Black, M.J. A Naturalistic Open Source Movie for Optical Flow Evaluation. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *ECCV*, volume 7577, pages 611–625, 2012.

[31] Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., and Bischof, H. Anisotropic Huber-L1 optical flow. In *Proceedings of the British Machine Vision Conference (BMVC)*, London, UK, September 2009.

[32] Black, M. J., Sapiro, G., Marimont, D. H., and Heeger, D. Robust anisotropic diffusion. *Trans. Img. Proc.*, 7(3):421–432, March 1998.