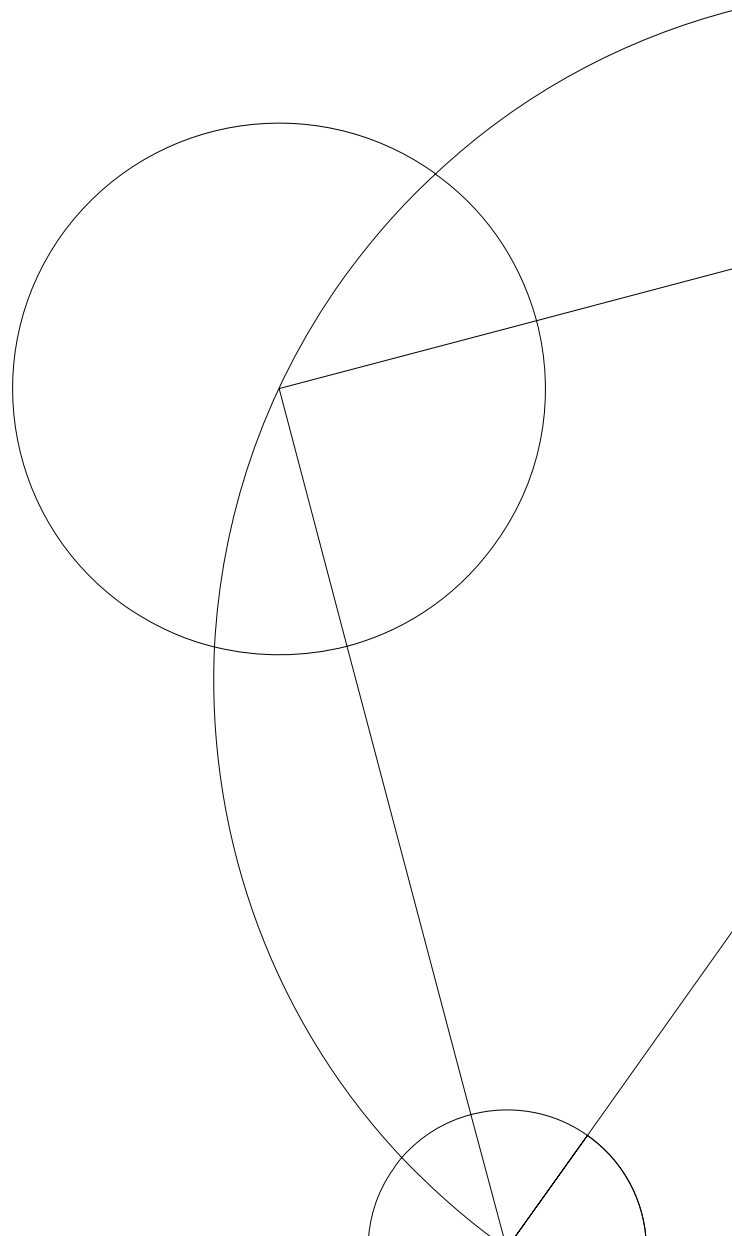# Duality based optical flow algorithms with applications

University of Copenhagen prize thesis

Lars Lau Rakêt

January 14, 2013

Abstract

We consider the popular TV-L^1 optical flow formulation, and the so-called duality based algorithm for minimizing the TV-L^1 energy. The original formulation is extended to allow for vector valued images, and minimization results are given. In addition we consider different definitions of total variation regularization, and related formulations of the optical flow problem that may be used with a duality based algorithm. We present a highly optimized algorithmic setup to estimate optical flows, and give five novel applications. The first application is registration of medical images, where X-ray images of different hands, taken using different imaging devices are registered using a TV-L^1 optical flow algorithm. We propose to regularize the input images, using sparsity enhancing regularization of the image gradient to improve registration results. The second application is registration of 2D chromatograms, where registration only have to be done in one of the two dimensions, resulting in a vector valued registration problem with values having several hundred dimensions. We propose a novel method for solving this problem, where instead of a vector valued data term, the different channels are coupled through the regularization. This results in a simple formulation of the problem, that may be solved much more efficiently than the conventional coupling. In the third application of the TV-L^1 optical flow algorithm we consider the problem of interpolating frames in an image sequence. We propose to move the motion estimation from the surrounding frames directly to the unknown frame by parametrizing the optical flow objective function such that the interpolation assumption is directly modeled. This reparametrization is a powerful trick that results in a number of appealing properties, in particular the motion estimation becomes more robust to noise and large displacements, and the computational workload is more than halved compared to usual bidirectional methods. Finally we consider two applications of frame interpolation for distributed video coding. The first of these considers the use of depth data to improve interpolation, and the second considers using the information from partially decoded video frames to improve interpolation accuracy in high-motion video sequences.

Notes

This thesis was awarded the University of Copenhagen silver medal at the 2013 annual commemoration. A number of typos present in the original thesis has been corrected in the present version, and several references have been updated.

Lars Lau Raket

Contents

Chapter 1

Introduction

This thesis is an answer to the call for prize papers announced at University of Copenhagen's annual commemoration 2011. In particular, it is an answer to the topic "Regularized energy methods in image analysis", proposed by Department of Computer Science.

For the energy method in question, we consider the TV-L^1 optical flow formulation, which has received a lot of attention in recent years. With the introduction of the so-called duality based method for minimizing this energy, Zach et al. (2007) opened the door to an entirely new way of estimating optical flow, that has fundamentally changed the field.

While the method introduced by Zach et al. (2007) is powerful, the original formulation is somewhat limiting. We begin this thesis with a theoretical section, where we first review the original formulation. We then consider extensions to allow for vector valued images, which will make it possible to estimate optical flows using color images. This extension was originally presented in Rakêt et al. (2011). We furthermore consider alternative definitions of the total variation term that is used for regularizing the results. A number of related formulations of the optical flow problem that fit into the duality based algorithm are reviewed, and in relation to this, we propose new data and regularization terms, and give directions on the minimization of the the corresponding energies.

We finally end the theoretical chapter by presenting a highly optimized algorithmic setup to estimate optical flows, and give results for some of the presented algorithms on benchmark data from the Middlebury Optical Flow Database (Baker et al. 2011).

The second part of this thesis consists of five novel applications of optical flow. The first application is registration of medical images, where X-ray images of different hands, taken using different imaging devices are registered using a TV-L^1 optical flow algorithm. In addition we consider the use of sparsity enhancing regularization of the input images, in order to improve registration results.

The second application considers registration of 2D chromatograms. For this particular dataset, registration is only necessary in one of the two dimensions of the data. With a fixed second dimension we may consider this as a vector valued registration problem with values having several hundred dimensions. A novel method for solving this is proposed, where instead of a coupling the different channels through the data term, the coupling is done through regularization.

This results in a very simple formulation of the problem, which may in addition be solved much more efficiently than the conventional coupling. This method, which may be used on many types of data, has originally been developed for the presented example in Rakêt & Markussen (2014), where the registration is used as a preprocessing step, prior to analysis of the dataset.

In the third application of the TV-L^1 optical flow algorithm we consider the problem of interpolating unknown frames in an image sequence. We propose to move the motion estimation from the surrounding frames directly to the unknown frame, by parametrizing the optical flow objective function such that the interpolation assumption is directly modeled. This reparametrization is a powerful trick that results in a number of appealing properties, in particular the motion estimation becomes more robust to noise and large displacements, and the computational workload is more than halved compared to usual bidirectional methods. This method was originally presented in Rakêt et al. (2012a).

Finally we consider two applications of frame interpolation meant to be used in distributed video coding setups. The first of these consider the use of depth data to improve interpolation quality. We show that including a standard asymmetric data term for the depth data with the symmetric data term presented in the previous application gives significantly better interpolation results than using either of the terms on their own. This application has been developed for the distributed video codec by Salmistraro, Rakêt, Zamarin, Ukhanova & Forchhammer (2013).

For the second application of interpolation in a distributed video coding setup, we consider using the information from partially decoded video frames to improve accuracy in high-motion video sequences. We develop a method to generate rough estimates of the frame to be decoded in pixel domain based on the decoded information in transform domain. With these initial estimates we are able to use a TV-L^1 optical flow method to fill in the fine details from the two known surrounding key frames. This method is used in the distributed video codec described in Luong et al. (2013), and have resulted in a significant bitrate saving, compared to the current state-of-the-art codec SING (Luong et al. 2012).

All results of this thesis are, unless otherwise mentioned, original and unpublished, and have been independently developed by the author.

Chapter 2

Optical Flow

The optical flow problem dates back to the works of Lucas & Kanade (1981) and Horn & Schunck (1981), that respectively proposed *local* and *global* resolution strategies. Given two images I_0 and I_1 the main problem in optical flow is defining a map v, such that the difference between I_1 warped according to v and I_0

$$I_1(\boldsymbol{x} + v(\boldsymbol{x})) - I_0(\boldsymbol{x}) \tag{2.1}$$

is close to zero. Solving (2.1) equal to zero is problematic in a number of ways. It is ill-posed; in the standard case we have one-dimensional brightness images, and for each point $\boldsymbol{x}$ we need to estimate the two components of v from a single equation. In addition the problem is highly non-linear. To deal with these problems the term (2.1) is typically linearized by means of its first order Taylor approximation in v. Local methods assume that the displacement $v(\boldsymbol{x})$ is similar in a neighborhood of $\boldsymbol{x}$, which typically gives enough linear independent equations in the channels of v for proper estimation. In contrast global optical flow methods typically use a pointwise data term based on the linearization of (2.1), but adds a regularization term, that penalizes erratic behavior of v, giving an energy that must be minimized in order to estimate v.

In the original formulation by Horn & Schunck (1981), optical flow is defined as "the distribution of apparent velocities of movement of brightness patterns in an image", which is directly compatible with (2.1) with grayscale images. Rather than this original definition, optical flow is today often thought of as the projected scene flow (Barron et al. 1994), that is the true motion of the objects in the scene as seen from the image plane.

Today the variational global approach to optical flow estimation is by far the method of choice for high accuracy optical flow algorithms, and judging from the authoritative Middlebury optical flow benchmark (Baker et al. 2011) the optical flow problem is essentially solved. The accuracy of optical flow algorithms have only increased marginally since 2010, where Xu et al. (2010) presented their estimation framework (Xu et al. 2012), with average endpoint errors of the estimated motion vectors that are typically less than one fifth of a pixel. So why still consider this problem, if one can only hope for pushing accuracy on the second or third decimals of benchmark data? A number of prominent reasons comes to mind. First, many of the top performing methods require large amounts of time (up to 10 hours) to compute a single displacement field

3

for small resolution images. Secondly, it seems that almost all top performing methods are either very complex in their formulation, or relies on solving the optical flow problem using highly sophisticated setups. In addition many methods rely on 'tricks' (Sun et al. 2010), and proper tuning of a large number of parameters. Finally, it seems that most focus has been on a single benchmark dataset, which means that many methods are essentially tailored to the specific evaluation setup. The consequence of this is that only little of the work that has been put in to solving the optical flow problem given by the Middlebury benchmark, has actually been transferred to possibly benefit related problems such as processing of video data or registration of medical images.

In this chapter we will review the so-called duality based optical flow method with a special focus on the TV-L^1 optical flow formulation, which is both fast and has been used in many different applications, demonstrating its robustness.

2.1 Duality based optical flow

Given a domain $\mathscr{T} \subseteq \mathbb{R}^d$ and a sequence of images $I_t : \mathscr{T} \to \mathbb{R}^k$, $I = (I_t)_{t \in T}$ for suitable T, we want to estimate the optical flow $v : \mathscr{T} \to \mathscr{T}$ such that the motion matches the image sequence with respect to some measure. We will consider a variational approach where the flow v is estimated as a minimizer of an energy on the form

$$E(v) = \lambda F(I, v) + G(v) \tag{2.2}$$

with F being a positive functional measuring data fidelity, and where G acts as a regularization term. Many energies of this type have been suggested throughout the years, and a large variety of solution methods exist (Horn & Schunck 1981, Papenberg et al. 2006, Zach et al. 2007, Zimmer et al. 2011). Here we will focus on a specific relaxation of the problem, and consider the minimization methods in this framework. The relaxed energy is obtained by introducing an auxiliary variable, effectively splitting the minimization problem in two quadratically coupled problems

$$E(u, v) = \lambda F(I, v) + \frac{1}{2\theta} \int \|v(\boldsymbol{x}) - u(\boldsymbol{x})\|^2 \, \mathrm{d}\boldsymbol{x} + G(u). \tag{2.3}$$

For $\theta \to 0$ a minimizer of (2.2) and (2.3) will clearly be the same, so the hope is that for θ small, a minimizer of the relaxed energy (2.3) will be close to a minimizer of the original energy (2.2). It may seem troublesome to introduce an auxiliary variable, since one has to iteratively solve the two energies

$$E_1(v) = \lambda F(I, v) + \frac{1}{2\theta} \int \|v(\boldsymbol{x}) - u(\boldsymbol{x})\|^2 \, \mathrm{d}\boldsymbol{x}, \tag{2.4}$$

$$E_2(u) = \frac{1}{2\theta} \int \|v(\boldsymbol{x}) - u(\boldsymbol{x})\|^2 \, \mathrm{d}\boldsymbol{x} + R(u), \tag{2.5}$$

and for a wide variety of choices for F and G, methods exist that directly target (relaxed) variants of the original energy (2.2). The splitting on the other hand also has a number of advantages, typically the two sub-problems (2.4) and (2.5) are much easier to solve, and in a number of important cases the minimization problems may very easily be solved on massively parallel processors

such as GPUs. Another positive feature is that data-matching and regularization are done independently, so one may easily replace one without changing the minimization of the other—a fact that makes comparison of different types of energies uncomplicated and fair, since the minimization is done in a fully comparable framework.

2.1.1 TV-L^1 optical flow

The by now classic duality based TV-L^1 optical flow algorithm of Zach et al. (2007) uses an L^1 norm for the data matching term F, and a vectorial total variation term for the regularization G, giving an energy of the form

$$E(v) = \lambda \int_{\mathscr{T}} \|R(v)(\boldsymbol{x})\| \, \mathrm{d}\boldsymbol{x} + \int_{\mathscr{T}} \|\mathscr{D}v(\boldsymbol{x})\| \, \mathrm{d}\boldsymbol{x}, \qquad (2.6)$$

where R is the given constancy assumption, that is typically defined from some variant of (2.1).

The TV-L^1 formulation was originally proposed by Brox et al. (2004), who also described a modern implementation in details, and gave a theoretical account for the choices. This algorithm marked a turning point with respect to optical flow accuracy, that also helped boosting the performance of later algorithms. The estimation in Brox et al. (2004) is based on the Euler-Lagrange framework, which require smooth functionals, and so the Euclidian norms in (2.6) are replaced with Charbonnier functions

$$\| \cdot \|_\varepsilon = \sqrt{\| \cdot \|^2 + \varepsilon^2} \qquad (2.7)$$

where ε is some small number.

Zach et al. (2007) proposed to recover a minimizer of (2.6) by iteratively minimizing the two convex quadratically coupled problems described in (2.3). In the given formulation, they are

$$E_1(v) = \lambda \int_{\mathscr{T}} \|\rho(v)(\boldsymbol{x})\| \, \mathrm{d}x + \frac{1}{2\theta} \int_{\mathscr{T}} \|v(\boldsymbol{x}) - u(\boldsymbol{x})\|^2 \, \mathrm{d}\boldsymbol{x}, \qquad (2.8)$$

$$E_2(u) = \int_{\mathscr{T}} \|\mathscr{D}_S u(\boldsymbol{x})\| \, \mathrm{d}x + \frac{1}{2\theta} \int_{\mathscr{T}} \|v(\boldsymbol{x}) - u(\boldsymbol{x})\|^2 \, \mathrm{d}\boldsymbol{x}. \qquad (2.9)$$

where ρ is the linearization of a grayscale data fidelity term R, and the chosen total variation term is defined as the sum of the total variation over all channels

$$\int_{\mathscr{T}} \|\mathscr{D}_S u(\boldsymbol{x})\| \, \mathrm{d}\boldsymbol{x} = \sum_{i=1}^{d} \int_{\mathscr{T}} \|\nabla u_i(\boldsymbol{x})\| \, \mathrm{d}\boldsymbol{x}. \qquad (2.10)$$

The flow is recovered by iteratively minimizing these energies in a coarse-to-fine pyramid scheme for some small θ.

For one-dimensional images, using for example only the brightness, we get a linearized data term of the form $\rho(v)(\boldsymbol{x}) = \boldsymbol{a}^\top v(\boldsymbol{x}) + b$, $\boldsymbol{a} \in \mathbb{R}^d$ and $b \in \mathbb{R}$. The minimizer of E_1 can be computed using the results given in Zach et al. (2007). These results are replicated in general form in the following lemma.

Lemma 2.1.1. For $\rho(v)(\boldsymbol{x}) = \boldsymbol{a}^\top v(\boldsymbol{x}) + b$, the minimizer of E_1 is given by

$$v(\boldsymbol{x}) = u(\boldsymbol{x}) - \pi_{\lambda\theta[-\boldsymbol{a},\boldsymbol{a}]}\left(u(\boldsymbol{x}) + \frac{b}{\|\boldsymbol{a}\|^2}\boldsymbol{a}\right) \tag{2.11}$$

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

$$\pi_{\lambda\theta[-\boldsymbol{a},\boldsymbol{a}]}\left(u(\boldsymbol{x}) + \frac{b}{\|\boldsymbol{a}\|^2}\boldsymbol{a}\right) = \begin{cases} -\lambda\theta\boldsymbol{a} & \text{if} \quad \rho(u)(\boldsymbol{x}) < -\lambda\theta\|\boldsymbol{a}\|^2 \\ \lambda\theta\boldsymbol{a} & \text{if} \quad \rho(u)(\boldsymbol{x}) > \lambda\theta\|\boldsymbol{a}\|^2 \\ \frac{\rho(u)(\boldsymbol{x})}{\|\boldsymbol{a}\|^2}\boldsymbol{a} & \text{if} \quad |\rho(u)(\boldsymbol{x})| \leq \lambda\theta\|\boldsymbol{a}\|^2 \end{cases}. \tag{2.12}$$

The regularization energy (2.9) is elegantly minimized by the method of Chambolle (2004). The solution is reproduced in the following lemma.

Lemma 2.1.2 (Chambolle)**.** The minimizer u of E_1 is given coordinatewise by

$$u_i = v_i - \theta\,\nabla \cdot p_i$$

for $i = 1, \ldots, d$, where $p_i : \mathscr{T} \to \mathbb{R}^d$ can computed by the iterative fixed-point scheme

$$p_i^{n+1} = \frac{\theta\,\boldsymbol{p}_i^n + \tau\nabla(\theta\,\nabla \cdot p_i^n - v_i)}{\theta + \tau|\nabla(\theta\,\nabla \cdot p_i^n - v_i)|}.$$

These lemmas provide an elegant, and easily implementable solution to the relaxed optical flow problem given by (2.8) and (2.9). On the other hand, the given formulation is somewhat restrictive, for example it does not allow for the use of vector valued images such as color images. In the next section, the problem of having using an L^1-norm for data term with vector valued images is considered. The problem is analyzed from a convex analysis point of view, and a solution to the resulting energy (2.8) is given. These results were originally proposed by Rakêt et al. (2011).

2.1.2 Minimizing affine L^1-L^2 energies

Consider an L^1-L^2 energy of the following form

$$E_1(v) = \lambda \int_{\mathscr{T}} \|Av(\boldsymbol{x}) + b(\boldsymbol{x})\|\,\mathrm{d}x + \frac{1}{2}\int_{\mathscr{T}} \|v(\boldsymbol{x}) - u(\boldsymbol{x})\|^2\,\mathrm{d}\boldsymbol{x} \tag{2.13}$$

where $A : \mathbb{R}^d \to \mathbb{R}^k$. Because no differential of v is involved, the minimization of (2.13) boils down to a pointwise minimization of a strictly convex cost function of the form

$$f(\boldsymbol{v}) = \lambda\|A\boldsymbol{v} + \boldsymbol{b}\| + \frac{1}{2}\|\boldsymbol{v} - \boldsymbol{u}\|^2. \tag{2.14}$$

In the following we present the tools used for solving the minimization problem (2.14). We recall first a few elements of convex analysis, the reader can refer to Ekeland & Teman (1999) for a complete introduction to convex analysis in both finite and infinite dimension. Here we will restrict ourselves to finite dimensional problems.

A function $f : \mathbb{R}^d \to \mathbb{R}$ is one-homogeneous if $f(\lambda \boldsymbol{x}) = \lambda f(\boldsymbol{x})$, for all $\lambda > 0$. For a one-homogeneous function, it is easily shown that its Legendre-Fenchel transform

$$f^*(\boldsymbol{x}^*) = \sup_{\boldsymbol{x} \in \mathbb{R}^d} \left\{ \langle \boldsymbol{x}, \boldsymbol{x}^* \rangle - f(\boldsymbol{x}) \right\} \tag{2.15}$$

is the characteristic function of a closed convex set $\mathfrak{C}$ of $\mathbb{R}^d$,

$$d_{\mathfrak{C}}(\boldsymbol{x}^*) := f^*(\boldsymbol{x}^*) = \begin{cases} 0 & \text{if } \boldsymbol{x}^* \in \mathfrak{C}, \\ +\infty & \text{otherwise.} \end{cases} \tag{2.16}$$

The one-homogeneous functions that will interest us here are of the form $f(\boldsymbol{x}) = \|A\boldsymbol{x}\|$ where $A : \mathbb{R}^d \to \mathbb{R}^k$ is linear, and $\| \cdot \|$ is the usual Euclidean norm of $\mathbb{R}^k$. The computation of the associated Fenchel transform involves the Moore-Penrose pseudoinverse $A^\dagger$ of A. We recall its construction.

The kernel (or null-space) of A, denoted $\mathrm{Ker}\, A$, is the vector subspace of the $\boldsymbol{v} \in \mathbb{R}^d$ for which $A\boldsymbol{v} = 0$. The image of A, denoted $\mathrm{Im}\, A$, is the subspace of $\mathbb{R}^k$ reached by A. The orthogonal complement of $\mathrm{Ker}\, A$ is denoted $\mathrm{Ker}\, A^\perp$. Denote by ι the inclusion map $\mathrm{Ker}\, A^\perp \to \mathbb{R}^d$ and let π be the orthogonal projection $\mathbb{R}^k \to \mathrm{Im}\, A$. It is well known that the composition map $B = \pi \circ A \circ \iota$

$$\mathrm{Ker}\, A^\perp \overset{\iota}{\longrightarrow} \mathbb{R}^d \overset{A}{\longrightarrow} \mathbb{R}^k \overset{\pi}{\longrightarrow} \mathrm{Im}\, A \tag{2.17}$$

is a linear isomorphism between $\mathrm{Ker}\, A^\perp$ and $\mathrm{Im}\, A$. The Moore-Penrose pseudoinverse $A^\dagger$ of A is defined as

$$A^\dagger = \iota \circ B^{-1} \circ \pi. \tag{2.18}$$

With this, the following lemma provides the Legendre-Fenchel transform of $f(\boldsymbol{x})$:

Lemma 2.1.3. The Legendre-Fenchel tranform of $\boldsymbol{x} \mapsto \|A\boldsymbol{x}\|$ is the characteristic function $d_{\mathfrak{C}}$ of the elliptic ball $\mathfrak{C}$ given by the set of $\boldsymbol{x}$'s in $\mathbb{R}^d$ that satisfy the following conditions

$$A^\dagger A \boldsymbol{x} = \boldsymbol{x} \tag{2.19}$$

$$\boldsymbol{x}^\top A^\dagger A^{\dagger\top} \boldsymbol{x} \leq 1. \tag{2.20}$$

From the properties of pseudoinverses, the equality $\boldsymbol{x} = A^\dagger A \boldsymbol{x}$ means that $\boldsymbol{x}$ belongs to $\mathrm{Ker}\, A^\perp$. In fact, $A^\dagger A$ is the orthogonal projection on $\mathrm{Ker}\, A^\perp$. On this subspace, $A^\dagger A^{\dagger\top}$ is positive definite and the inequality thus defines an elliptic ball.

The lemma will not be proven here, but we indicate how it can be done. In the case where A is the identity $\mathbb{I}_d$ of $\mathbb{R}^d$, it is easily shown that $\mathfrak{C}$ is the unit sphere of $\mathbb{R}^d$. The case where A is invertible follows easily, while the general case follows from the latter using the structure of pseudoinverse (see Golub & van Loan (1989) for instance).

We can now state the main result which allows to generalize the TV-L^1 algorithm from Zach et al. (2007) to calculate the optical flow between two vector valued images.

Proposition 2.1.4. The minimizer of the function

$$f(\boldsymbol{v}) = \lambda \|A\boldsymbol{v} + \boldsymbol{b}\| + \frac{1}{2}\|\boldsymbol{v} - \boldsymbol{u}\|^2$$

is given as follows.

(i) In the case $\boldsymbol{b} \notin \operatorname{Im} A$, $f(\boldsymbol{v})$ is smooth. It can be minimized by usual methods.

(ii) In the case where $\boldsymbol{b} \in \operatorname{Im} A$, $f(\boldsymbol{v})$, which fails to be smooth for $\boldsymbol{v} \in \operatorname{Ker} A + A^\dagger b$, reaches its unique minimum at

$$\boldsymbol{v} = \boldsymbol{u} - \pi_{\lambda\mathfrak{C}}\left(\boldsymbol{u} + A^\dagger \boldsymbol{b}\right) \tag{2.21}$$

where $\pi_{\lambda\mathfrak{C}}$ is the projection onto the convex set $\lambda\mathfrak{C} = \{\lambda\boldsymbol{x}, \ \boldsymbol{x} \in \mathfrak{C}\}$, with $\mathfrak{C}$ as described in Lemma 2.1.3.

Proof. To see (i), write $\boldsymbol{b}$ as $A\boldsymbol{b}_0 + \boldsymbol{b}_1$, with $\boldsymbol{b}_0 = A^\dagger\boldsymbol{b}$, $A\boldsymbol{b}_0$ being then orthogonal projection of $\boldsymbol{b}$ onto $\operatorname{Im} A$, while $\boldsymbol{b}_1$ is the residual of the projection. The assumption of (i) implies that $\boldsymbol{b}_1 \neq \boldsymbol{0}$ is orthogonal to the image of A. One can then write

$$\|A\boldsymbol{v} + \boldsymbol{b}\| = \|A(\boldsymbol{v} + \boldsymbol{b}_0) + \boldsymbol{b}_1\| = \sqrt{\|A(\boldsymbol{v} + \boldsymbol{b}_0)\|^2 + \|\boldsymbol{b}_1\|^2} \tag{2.22}$$

which is always strictly positive as $\|\boldsymbol{b}_1\|^2 > 0$, and smoothness follows.

In the situation of (ii), since $\boldsymbol{b} \in \operatorname{Im} A$, we can do the substitution $\boldsymbol{v} \leftarrow \boldsymbol{v} + A^\dagger\boldsymbol{b}$ in function (2.14) and the resulting function has the same form as a number of functions found in Chambolle (2004) and Chambolle & Pock (2011). We refer the reader to them for the computation of minimizers. $\qquad\square$

Proposition 2.1.4 generalizes Lemma 2.1.1 since, on one-dimensional spaces, elliptic balls are simply line segments. The next examples extends to multi-dimensional values.

Example 2.1.1. Consider the minimization problem

$$\arg\min_{\boldsymbol{v}} \left(\lambda\|A\boldsymbol{v} + \boldsymbol{b}\| + \frac{1}{2}\|\boldsymbol{v} - \boldsymbol{u}\|^2\right), \qquad \lambda > 0. \tag{2.23}$$

where $A \in \mathbb{R}^{k \times 2}$ and $\boldsymbol{b} \in \operatorname{Im} A$. If A has maximal rank (i.e. 2), then is is well known that the 2×2 matrix $C = A^\dagger A^{\dagger\top}$ is symmetric and positive definite (Golub & van Loan 1989). The set $\mathfrak{C}$ is then an elliptic disc determined by the eigenvectors and eigenvalues of C. The projection may be computed by the efficient algorithm described in Example 2.1.3, which has much better properties than the method originally suggested in Rakêt et al. (2011).

In the case that the matrix has two linearly dependent columns $\boldsymbol{a} \neq \boldsymbol{0}$ and $c\boldsymbol{a}$, a series of straightforward calculations give

$$\operatorname{Ker} A = \mathbb{R}\boldsymbol{y}, \quad \operatorname{Ker} A^\perp = \mathbb{R}\boldsymbol{x}, \quad \operatorname{Im} A = \mathbb{R}\boldsymbol{a} \tag{2.24}$$

with $\boldsymbol{x} = \frac{1}{1+c^2}(1, c)^\top$ and $\boldsymbol{y} = \frac{1}{1+c^2}(-c, 1)^\top$ an orthonormal basis of $\mathbb{R}^2$, and

$$A^\dagger A^{\dagger\top} = \frac{1}{(1+c^2)^2\|\boldsymbol{a}\|^2} \begin{pmatrix} 1 & c \\ c & c^2 \end{pmatrix}. \tag{2.25}$$

If $c = 0$, the inequality (2.20) from Lemma 2.1.3, just amounts to

$$\frac{u_1^2}{\|\boldsymbol{a}\|^2} \leq 1 \iff -\|\boldsymbol{a}\| \leq u_1 \leq \|\boldsymbol{a}\|, \qquad \boldsymbol{u} = (u_1, u_2)^\top \qquad (2.26)$$

that is a vertical strip, while equality (2.19) in Lemma 2.1.3 simply says that $u_2 = 0$, thus $\mathfrak{C}$ is the line segment

$$[-\|\boldsymbol{a}\|\boldsymbol{x}, \|\boldsymbol{a}\|\boldsymbol{x}] \subset \mathbb{R}^2. \qquad (2.27)$$

The case where $c \neq 0$ is identical, and obtained for instance by rotating the natural basis of $\mathbb{R}^2$ to the basis $(\boldsymbol{x}, \boldsymbol{y})$. ○

Example 2.1.2. Consider again the minimization problem (2.23), but this time assuming that $\boldsymbol{b} \notin \operatorname{Im} A$. Using (2.22) we can rewrite the minimization problem as

$$\arg\min_{\boldsymbol{v}} \left(\lambda \sqrt{\|A(\boldsymbol{v} + \boldsymbol{b}_0)\|^2 + \|\boldsymbol{b}_1\|^2} + \frac{1}{2}\|\boldsymbol{v} - \boldsymbol{u}\|^2 \right), \qquad \lambda > 0. \qquad (2.28)$$

The minimizing $\boldsymbol{v}$ is found by solving the equation

$$\lambda \frac{A^\top A(\boldsymbol{v} + \boldsymbol{b}_0)}{\|A\boldsymbol{v} + \boldsymbol{b}\|} + \boldsymbol{v} - \boldsymbol{u} = 0$$

which may be done by gradient descent or a (quasi-)Newton method. ○

Example 2.1.3. Consider the problem of projecting a point $\boldsymbol{x}_0$ onto the ellipsoid given by

$$\mathfrak{C} = \{\boldsymbol{x} \in \mathbb{R}^k \mid \boldsymbol{x}^\top C \boldsymbol{x} \leq 1\},$$

where $\boldsymbol{x}_0 \notin \mathfrak{C}$.

We may formulate this as the problem of finding the minimizer $\hat{\boldsymbol{x}}$ given by

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x} \in \mathfrak{C}} \|\boldsymbol{x} - \boldsymbol{x}_0\|^2.$$

This problem can be solved by introducing a Lagrange multiplier ξ, giving the objective function

$$f(\boldsymbol{x}, \lambda) = \|\boldsymbol{x} - \boldsymbol{x}_0\|^2 + \xi(\boldsymbol{x}^\top C\boldsymbol{x} - 1).$$

From the condition that

$$\frac{\partial}{\partial \boldsymbol{x}} f(\boldsymbol{x}, \xi) = 2(\boldsymbol{x} - \boldsymbol{x}_0) + 2\xi C\boldsymbol{x} = \boldsymbol{0},$$

we get that

$$\hat{\boldsymbol{x}} = (\xi C + \mathbb{I})^{-1}\boldsymbol{x}_0.$$

However we need to determine the value of the Lagrange multiplier ξ. Since we assumed that $\boldsymbol{x}_0$ was outside the ellipsoid, we know that the projected point will lie on the boundary of the ellipse $\partial \mathfrak{C}$, that is ξ is a root of

$$G(\xi) = ((\xi C + \mathbb{I})^{-1}\boldsymbol{x}_0)^\top C(\xi C + \mathbb{I})^{-1}\boldsymbol{x}_0 - 1. \qquad (2.29)$$

We can use the following theorem due to Kiseliov (1994) to determine the correct value of ξ.

9

Theorem 2.1.5. The root ξ^* of (2.29) is unique and can be found by the iterative Newton process

$$\xi_0 = 0, \qquad \xi_{n+1} = \xi_n - \frac{G(\xi_n)}{G'(\xi_n)},$$

where $\xi_k \uparrow \xi^*$. The rate of convergence is quadratic.

Proof. Since we are assuming that $\boldsymbol{x}_0 \notin \mathfrak{C}$, it holds that

$$G(0) = \boldsymbol{x}_0^\top C \boldsymbol{x}_0 - 1 > 0, \qquad \lim_{\xi \to \infty} G(\xi) = -1 < 0,$$

which gives that a root exists in $[0, \infty)$.

Since

$$\boldsymbol{0} = \frac{\mathrm{d}}{\mathrm{d}\xi}(\xi C + \mathbb{I})^{-1}(\xi C + \mathbb{I}) = \left(\frac{\mathrm{d}}{\mathrm{d}\xi}(\xi C + \mathbb{I})^{-1}\right)(\xi C + \mathbb{I}) + (\xi C + \mathbb{I})^{-1}\left(\frac{\mathrm{d}}{\mathrm{d}\xi}(\xi C + \mathbb{I})\right),$$

we have that

$$\frac{\mathrm{d}}{\mathrm{d}\xi}(\xi C + \mathbb{I})^{-1} = -(\xi C + \mathbb{I})^{-1}C(\xi C + \mathbb{I})^{-1} = -(\xi C + \mathbb{I})^{-2}C.$$

Using this we can differentiate G

$$G'(\xi) = -2\boldsymbol{x}_0^\top(\xi C + \mathbb{I})^{-3}C^2\boldsymbol{x}_0$$

and in addition

$$G''(\xi) = 6\boldsymbol{x}_0^\top(\xi C + \mathbb{I})^{-4}C^3\boldsymbol{x}_0,$$

where we have used the commutativity of C and $(\xi C + \mathbb{I})^{-1}$. Since $(\xi C + \mathbb{I})^{-1}$ has full rank for all $\xi \geq 0$ we see that

$$G'(\xi) < 0, \qquad G''(\xi) > 0$$

so the solution ξ^* is unique. $\qquad\square$

Kiseliov (1994) in addition gives a non-linear version of the Newton process described in the above theorem, which is even more efficient. Compared to the added complexity of the implementation, the overall gain of using such an algorithm is limited, and we will recommend the process described here. ○

2.1.3 Alternative optical flow formulations

In the wake of the algorithm by Zach et al. (2007), a large number of duality based or primal-dual methods emerged in optical flow estimation. The initial focus has mainly been on improving regularization. Wedel et al. (2009b) considers structure and motion adaptive regularization. Werlberger et al. (2009) has gone further to consider full anisotropic regularization, where regularization directions are weighted differently by means of a diffusion tensor. In addition the L^1 norm used in the regularization of the original TV-L^1 were replaced with a Huber norm that is smooth at the origin, thus eliminating the staircasing effect of the regularization. In Werlberger et al. (2010) non-local total variation

is considered, where a low level image segmentation is integrated in the regularization. This in turn produces very sharp motion boundaries, and preserves small scale structures in the flow very well.

In addition to the refinement of regularization techniques, some work has been done on reformulating data terms. Wedel et al. (2008) shows how to minimize a sum of two L^1 data terms for one-dimensional images. Recognizing the pointwise structure of many data terms, Steinbrücker et al. (2009a) proposed to use brute-force minimization of the data fidelity energy (2.4), without linearizing the optical flow constraint (2.1). A number of more advanced pointwise data terms are considered in Steinbrücker et al. (2009b), but unfortunately the quality of the resulting flows are not as impressive as one could hope for. Werlberger et al. (2010) use truncated normalized cross correlation for their data term. This data term is attractive because of its invariance to multiplicative illumination changes in the scene. It is however not defined pointwise, and thus needs a more complex minimization strategy. This is done by a second order approximation of the data term, in contrast to the usual first order approximation. Building on these ideas, Panin (2012) considers a mutual information data terms. Although the benchmark optical flow results of Panin (2012) cannot compete with a highly optimized TV-L^1 implementation (Wedel et al. 2009a), the algorithm shows impressive results under less optimal conditions such as noise and transformations of the values in one of the images to be registered.

In the following we will consider some examples of alternative data and regularization terms, and consider how they may be minimized. At the end we will consider other extensions.

Example 2.1.4 (L^2 data term). The cost function

$$f(\boldsymbol{v}) = \frac{\lambda}{2}\|A\boldsymbol{v} + \boldsymbol{b}\|^2 + \frac{1}{2}\|\boldsymbol{v} - \boldsymbol{u}\|^2$$

is clearly smooth and convex in $\boldsymbol{v}$, so there is a unique minimizer to the problem that can be found as the solution to

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{v}}f(\boldsymbol{v}) = \boldsymbol{0}.$$

We readily get that

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{v}}f(\boldsymbol{v}) = \boldsymbol{v} - \boldsymbol{u} + \lambda\theta A^\top(A\boldsymbol{v} + \boldsymbol{b}) = (\mathbb{I} + \lambda\theta A^\top A)\boldsymbol{v} - \boldsymbol{u} + \lambda\theta A^\top\boldsymbol{b},$$

and the solution to the problem is

$$\boldsymbol{v} = (\mathbb{I} + \lambda\theta A^\top A)^{-1}(\boldsymbol{u} - \lambda\theta A^\top\boldsymbol{b}).$$

For comparison purposes it may be interesting to rewrite this solution as

$$\boldsymbol{v} = \boldsymbol{u} - \lambda\theta(\mathbb{I} + \lambda\theta A^\top A)^{-1}A^\top(A\boldsymbol{u} + \boldsymbol{b}).$$

Note that for $k = 1$, $A^\top = \boldsymbol{a}$, so the above formula becomes

$$\boldsymbol{v} = \boldsymbol{u} - \frac{\lambda\theta(\boldsymbol{a}^\top\boldsymbol{u} + b)\lambda\theta}{1 + \lambda\theta\|\boldsymbol{a}\|^2}\boldsymbol{a}. \tag{2.30}$$

○

Example 2.1.5 (Charbonnier norm). While the approach for minimizing the vector valued data term in Section 2.1.2 is nice from a theoretical point of view, the actual implementation of the solution given in Proposition 2.1.4 is not very practical. The checks needed in order to determine which category the given point falls into, and the iterative procedures needed to project onto an ellipsoid, or for the solution of (2.28), result in an algorithm that is hard to implement and quite slow. The original algorithm presented in Rakêt et al. (2011) dealt with this somewhat inelegantly, by ensuring full rank of of the matrix A by means of regularization, followed by projection of $\boldsymbol{b}$ onto $\operatorname{Im} A$.

By replacing the Euclidian norm with the Charbonnier norm $\|\cdot\|_\varepsilon$ given in (2.7), we may avoid the checks related to the cases of Proposition 2.1.4, and instead just perform iterative minimization at all points following Example 2.1.2. ○

Example 2.1.6 (Interval data term). Consider the following penalty function

$$\varphi(x) = -\mathbb{1}_{(-\infty,c_1)}(x)(x-c_1) + \mathbb{1}_{(c_2,\infty)}(x)(x-c_2)$$

where $c_1 \leq 0 \leq c_2$ and $\mathbb{1}$ is the indicator function. This type of penalty may be an interesting data term when data is very noisy, or in general when a perfect data fit is not realizable over most of the image.

First consider a function of the form

$$\varphi(x) + \beta(x-y)^2.$$

If $y \in [c_1, c_2]$, it is minimized by $x = y$. For $y \geq c_2$ we want to minimize

$$\mathbb{1}_{(c_2,\infty)}(x)(x-c_2) + \beta(x-y)^2.$$

If $\beta(y-c_2) \leq 1/2$ the minimizer is $x = c_2$, and otherwise it is $x = y - 1/2\beta$. A similar expression is found for $y \leq c_1$, giving the final solution

$$x = \begin{cases} y + 1/2\beta & \text{if } y \in (-\infty, c_1 - 1/2\beta) \\ c_1 & \text{if } y \in [c_1 - 1/2\beta, c_1) \\ y & \text{if } y \in [c_1, c_2] \\ c_2 & \text{if } y \in (c_2, c_2 + 1/2\beta] \\ y - 1/2\beta & \text{if } y \in (c_2 + 1/2\beta, \infty) \end{cases}.$$

Consider now the function

$$f(\boldsymbol{v}) = \varphi(\boldsymbol{a}^\top \boldsymbol{v} + b) + \beta\|\boldsymbol{v} - \boldsymbol{u}\|^2.$$

Denoting $\rho(\boldsymbol{v}) = \boldsymbol{a}^\top \boldsymbol{v} + b$, the solution can be found as

$$\boldsymbol{v} = \begin{cases} \boldsymbol{u} + 1/2\beta\boldsymbol{a} & \text{if } \rho(\boldsymbol{u}) \in (-\infty, c_1 - 1/2\beta\|\boldsymbol{a}\|^2) \\ \boldsymbol{u} - (\rho(\boldsymbol{u}) - c_1)\boldsymbol{a}/\|\boldsymbol{a}\|^2 & \text{if } \rho(\boldsymbol{u}) \in [c_1 - 1/2\beta\|\boldsymbol{a}\|^2, c_1) \\ \boldsymbol{u} & \text{if } \rho(\boldsymbol{u}) \in [c_1, c_2] \\ \boldsymbol{u} - (\rho(\boldsymbol{u}) - c_2)\boldsymbol{a}/\|\boldsymbol{a}\|^2 & \text{if } \rho(\boldsymbol{u}) \in (c_2, c_2 + 1/2\beta\|\boldsymbol{a}\|^2] \\ \boldsymbol{u} - 1/2\beta\boldsymbol{a} & \text{if } \rho(\boldsymbol{u}) \in (c_2 + 1/2\beta\|\boldsymbol{a}\|^2, \infty) \end{cases}$$

we see that for $c_1 = c_2 = 0$, the solution is identical to the one given in Lemma 2.1.1. In addtion it is interesting to note that similar calculations can be used to give an explicit solution to the truncated L^1 data term, which was minimized by brute force by Steinbrücker et al. (2009b). ○

Example 2.1.7 (Vectorial total variation). As already mentioned the total variation of a vector valued function is not uniquely defined, and the different definitions will give results with different properties. In the following it is assumed that $d = 2$ for simplicity.

We have already introduced the channel-by-channel definition of the vectorial total variation (2.10), which is used in the original formulation by Zach et al. (2007).

The canonical definition of vectorial total variation, which is also the definition used by the original TV-L^1 algorithm of Brox et al. (2004), is

$$\int_{\mathscr{T}} \|\mathscr{D}_F u(\boldsymbol{x})\| \, \mathrm{d}\boldsymbol{x} = \sup_{p \in \mathscr{P}} \int_{\mathscr{T}} \langle u(\boldsymbol{x}), \nabla \cdot p(\boldsymbol{x}) \rangle \, \mathrm{d}\boldsymbol{x} \tag{2.31}$$

with $p : \mathbb{R}^2 \to \mathbb{R}^{2 \times 2}$, and where $\mathscr{P} = \{p \in C_c^1(\mathbb{R}^2, R^{2 \times 2}) : \|p\|_2 \le 1\}$. It is worth noting that the definition of Zach et al. (2007) corresponds to the requirement that $\|p\|_\infty \le 1$ in $\mathscr{P}$. If we assume that u is smooth, using integration by parts with proper boundary conditions yields that

$$\sup_{p \in \mathscr{P}} \int_{\mathscr{T}} \langle u(\boldsymbol{x}), \nabla \cdot p(\boldsymbol{x}) \rangle \, \mathrm{d}\boldsymbol{x} = \sup_{p \in \mathscr{P}} \int_{\mathscr{T}} \sum_{i=1}^{2} \langle \nabla u_i(\boldsymbol{x}), p_i(\boldsymbol{x}) \rangle \, \mathrm{d}\boldsymbol{x}. \tag{2.32}$$

For $\nabla u \ne \boldsymbol{0}$ the supremum is found to be

$$p_i = \frac{\nabla u_i}{\|\nabla u\|}, \qquad \|\nabla u\| = \sqrt{\|\nabla u_1\|^2 + \|\nabla u_2\|^2} \tag{2.33}$$

and for $\nabla u = \boldsymbol{0}$, p can be any function in $\mathscr{P}$, which in turn means that for smooth u

$$\int \|\mathscr{D}u(\boldsymbol{x})\| \, \mathrm{d}\boldsymbol{x} = \int \sqrt{\|\nabla u_1(\boldsymbol{x})\|^2 + \|\nabla u_2(\boldsymbol{x})\|^2} \, \mathrm{d}\boldsymbol{x} = \int \|\nabla u(\boldsymbol{x})\| \, \mathrm{d}\boldsymbol{x}. \tag{2.34}$$

This definition has some very nice properties. In particular it is rotationally invariant, and couples the channels by weighting the regularization differently across the different channels.

This definition is not directly compatible with Chambolle's algorithm (Lemma 2.1.2), however one may use the following algorithm proposed by Bresson & Chan (2008), which is a direct extension.

Lemma 2.1.6. The minimizer u of (2.9) is given by

$$u = v - \theta \nabla \cdot p \tag{2.35}$$

which can be solved with the convergent semi-implicit gradient descent scheme

$$p^{n+1} = \frac{p^n + \tau \nabla(\nabla \cdot p^n - v/\theta)}{1 + \tau \|\nabla(\nabla \cdot p^n - v/\theta)\|} \tag{2.36}$$

where $\tau \le 1/8$, $p^0 = \boldsymbol{0}$.

Recently Goldlücke et al. (2012) introduced an alternative definition of vectorial total variation. Assuming sufficient smoothness of u, this definition, which we will denote by

$$\int_{\mathscr{T}} \|\mathscr{D}_J u(\boldsymbol{x})\| \, \mathrm{d}\boldsymbol{x}$$

corresponds to the integral over the largest singular value of the derivative matrix of u. This definition smooths in a single direction across channels, and thus does not suffer from the channel smearing effects of the two previously defined methods. The large number of examples given to color imaging by Goldlücke et al. (2012) are very convincing, showing consistently better results of this method in different applications. For the minimization of the energy (2.9) with this choice of total variation, we will use the Bermùdez-Moreno (BM) type algorithm given in Goldlücke et al. (2012), which is directly comarable to the solutions given in lemmas 2.1.2 and 2.1.6.

In the following these three definitions will be denoted by TV_S, TV_F, and TV_J respectively. ○

Example 2.1.8 (1-harmonic regularization)**.** As we have seen so far, it is common practice in optical flow estimation to formulate the regularization of the optical flow by means of an L^p norm of the flow gradient. However, when considering the nature of optical flow fields, one realizes that this is perhaps not the natural type of regularization. An optical flow field describes the motion of a projected scene. Considering a scene where all motion is parallel to the camera plane, and objects are rigid and move in a single spatial direction. In this setup the displacement vectors of an object in the projected image will point in the same direction, but the magnitude of the flow vectors will vary depending on the distance of the particular part of the object to the camera. This suggests that one should regularize direction to a higher extend than magnitude.

Additional directional regularization has been proposed by Gai & Stevenson (2010) in the form of an additional 1-harmonic regularization term (Vese & Osher 2002) to the original TV-L^1 method Wedel et al. (2009a). A fully polar representation of optical flow was considered by Adato et al. (2011), who demonstrate good results on the Middlebury training data by completely decoupling the angular and magnitude component of the flow. The chosen representation does however increase the complexity of the formulation of the problem considerably. In addition both methods are quite slow, and they owe much of their precision to being built upon existing well-performing methods (Wedel et al. (2009a) and Sun et al. (2010) respectively). It turns out that one may combine the elements of the two mentioned methods, in an elegant manner, which in addition has very attractive computational properties. The regularization term we propose to use is the following

$$G(v) = \alpha \int_{\mathscr{T}} \big|\nabla\|v(\boldsymbol{x})\|\big|\,\mathrm{d}\boldsymbol{x} + (1-\alpha) \int_{\mathscr{T}} \left\|\nabla \tfrac{v(\boldsymbol{x})}{\|v(\boldsymbol{x})\|}\right\|\,\mathrm{d}\boldsymbol{x}, \qquad 0 \le \alpha \le 1.$$

The above regularization term is similar to the one of Adato et al. (2011), since it completely decouples magnitude and direction, however, instead of having to solve a constrained problem with an angular component, we use a 1-harmonic term as in Gai & Stevenson (2010).

We are now interested interested in minimizing an energy of the form

$$E(v) = \lambda F(v) + G(v).$$

Using the standard quadratic decoupling (2.3) will introduce a coupling of magnitude and direction in the regularization, and in order to avoid that, we propose

the following decoupling

$$E_1(v) = \lambda F(v) + \frac{1}{2\theta} \int_{\mathscr{T}} \|v(\boldsymbol{x}) - u(\boldsymbol{x})\|^2 \, \mathrm{d}\boldsymbol{x},$$

$$E_2(u) = \frac{1}{2\theta_1} \int_{\mathscr{T}} \left| \|v(\boldsymbol{x})\| - \|u(\boldsymbol{x})\| \right|^2 \, \mathrm{d}\boldsymbol{x} + \frac{1}{2\theta_2} \int_{\mathscr{T}} \left| \frac{v(\boldsymbol{x})}{\|v(\boldsymbol{x})\|} - \frac{u(\boldsymbol{x})}{\|u(\boldsymbol{x})\|} \right|^2 \, \mathrm{d}\boldsymbol{x} + G(v).$$

In this formulation magnitude and directions are independent, and representing flows this way will even allow for simultaneous minimization of the magnitude and regularization parts of the flow. The magnitude regularization just corresponds to one-dimensional total variation regularization, and may be minimized following Chambolle (2004). The directional regularization may be solved efficiently following Rakêt & Nielsen (2012).

When considering the splitting scheme, it does not seem as elegant as the typical quadratic splitting, as it cannot directly be formulated as a single energy in two variables. For θ, θ_1, and θ_2 sufficiently small however, the two solutions should converge to each other. In addition, if one considers the splitting as an iterative estimation process, it does makes sense that minimization of the data term gives a solution with fundamentally different properties than the estimate produced by minimizing the regularization term. In this light it makes sense that one should treat the previous estimate differently in the two minimization problems. This also gives a good explanation for the widespread use and success of intermediate median filtering in duality based optical flow estimation (Wedel et al. 2009a, Sun et al. 2010). ○

Example 2.1.9 (Illumination and occlusion modeling). Illumination changes in image sequences present a major problem for conventional optical flow estimation. Changing light conditions from one frame to the next may render the data term completely unable to match objects in the scene. Another problem is the issue of occlusions, when an object (partly) disappears from one frame to the next. This will naturally lead to violations of the optical flow constraint.

In the TV-L^1 optical flow setting, Chambolle & Pock (2011) proposed to model violations of the data term by adding a compensating term $\beta c(\boldsymbol{x})$ to the linearized optical flow constraint $\rho(v)(\boldsymbol{x})$. Illumination changes are expected to affect the residual $\rho(v)(\boldsymbol{x})$ similarly in connected regions, so Chambolle & Pock (2011) proposed to regularize c using total variation. This new illumination field may seamlessly be integrated to the algorithm in a similar fashion to was has been done so far: We split data and regularization of c using a quadratic term and minimize iteratively.

A similar method has been used for occlusion detection by Ayvaci et al. (2012). A compensating term c is added to the data fidelity functional, however since only occlusions are modeled, a sparsity enhancing L^0 regularization is proposed. In the presented setup this gives the following data energy

$$E_1(v, c) = \lambda \int_{\mathscr{T}} \|\rho(v)(\boldsymbol{x}) - c(\boldsymbol{x})\| \, \mathrm{d}\boldsymbol{x} + \frac{1}{2\theta} \int_{\mathscr{T}} \|v(\boldsymbol{x}) - u(\boldsymbol{x})\|^2 \, \mathrm{d}\boldsymbol{x} + \beta \|c\|_{L^0},$$

where the L^0 norm is defined as

$$\|c\|_{L^0} = \int_{\mathscr{T}} \|c(\boldsymbol{x})\|_{\ell_0} \, \mathrm{d}\mu(\boldsymbol{x}), \qquad \|y\|_{\ell_0} = \begin{cases} 0 & \text{if } y = 0 \\ 1 & \text{otherwise} \end{cases},$$

15

with μ denoting the Hausdorff measure. In order to minimize the full energy using specific solvers Ayvaci et al. (2012) iteratively approximate the L^0 term by weighted L^1 terms. This does however seem somewhat unnecessary, as a closed-form pointwise solution for c is easily found. In the duality based setting v may be calculated using Lemma 2.1.1, and the minimizer of c is given pointwise as

$$
c(\boldsymbol{x}) = \begin{cases} \rho(v)(\boldsymbol{x}) & \text{if } \|\rho(v)(\boldsymbol{x})\| > {}^{\beta}\!/\!_{\lambda} \\ 0 & \text{otherwise} \end{cases}.
$$

This means that we simply have a thresholding step where if the data residual $\rho(v)(\boldsymbol{x})$ is too big, it is considered an occlusion, and motion in the area (which in principle is not defined) is fully determined by the regularization term. ∘

2.2 Algorithm

This section will describe a general algorithmic framework for estimating different types of duality based optical flows from energies on the form (2.3). As already mentioned the duality based approach has good computational properties, because the solutions to the two sub-energies may be done in parallel. This makes the algorithm perfectly suited for massively parallel processors.

The structure of the algorithm is depicted in Algorithm 2.1.

Data: Two images I_0 and I_1
Result: The optical flow field v
for $\ell = \ell_{\max}$ *to* 0 **do**
    // Pyramid levels
    Downsample the images I_0 and I_1 to current pyramid level
    for $w = 0$ *to* $w_{\max}$ **do**
        // Warping
        Compute v as the minimizer of E_1 (2.4)
        for $i = 0$ *to* $i_{\max}$ **do**
            // Inner iterations
            Compute u as the minimizer of E_2 (2.5)
        end
        Upscale v and u to next pyramid level
    end
end

Algorithm 2.1: Computation of duality based optical flow.

The standard settings of the algorithm is described in the following. Unless specifically mentioned, these are the settings used in the calculations described in the rest of this thesis.

Pyramid An image pyramid is built, where on each level, prior to down-sampling to the next pyramid level, the images are smoothed with a Gaussian function of standard deviation σ. The downsampling is done by means of linear interpolation. Evaluation at non-pixel positions in images is done by bicubic interpolation.

Warping   At the beginning of each warp, the image I_1 is warped according to the current estimate v. If median filtering is used to remove outliers (it is not in the standard setting), it is performed on v prior to warping of I_1.

Upscaling   Flows are upscaled using linear interpolation, and their values are divided by the downscale factor of the pyramid, in order for vector lengths to match the current image size. This is followed by an application of a 3×3 median filter.

The standard parameters of the flow algorithm are given in Table 2.1.

Table 2.1: Standard parameters of the optical flow algorithm depicted in Algorithm 2.1

Parameter	value
$\ell_{\max}$	70
downscale factor	0.95
σ	$\frac{\sqrt{2}}{4}$
$w_{\max}$	90
$i_{\max}$	20
λ	50
θ	0.2

The algorithm has been implemented in CUDA C in order to take advantage of the thousands of cores on modern GPUs.

2.3   Results

This section presents optical flow results for the TV-L^1 algorithm using the implementation proposed in the previous section. In addition some of the extensions from Section 2.1.3 are considered, and analyzed.

Choosing TV definition   Example 2.1.7 introduced three different definitions of vectorial total variation. Using the algorithm described in the previous section, we can consider the difference in terms of accuracy of the resulting TV-L^1 optical flow algorithm on the Middlebury Optical Flow Database training data (Baker et al. 2011). Figure 2.1 shows the average endpoint errors (AEEs) for the different definitions on the training sequences, as a function of λ. We see that generally the definitions that couple channels, TV_F and TV_J, perform better than the uncoupled regularization TV_S. Figure 2.2 shows the average performance over all test sequence, and we see an average difference in AEE of approximately 0.01 between using coupled or uncoupled regularization, for the respective optimal choices of λs. This may not seem like much, but since optical flow algorithms today have so high accuracies, an average improvement of this order will give significantly different rankings of the algorithms on for example the Middlebury ranking (as of January 14, 2013 the average difference between

the two top ranking methods in terms of AEE, *MDP-Flow2* and *NN-Field* is
0.0025). Considering the coupled methods there seem to be little difference in
the optimal area, and both choices seem reasonable. Based on this analysis we
choose the definition TV_J of Goldlücke et al. (2012) as our standard method,
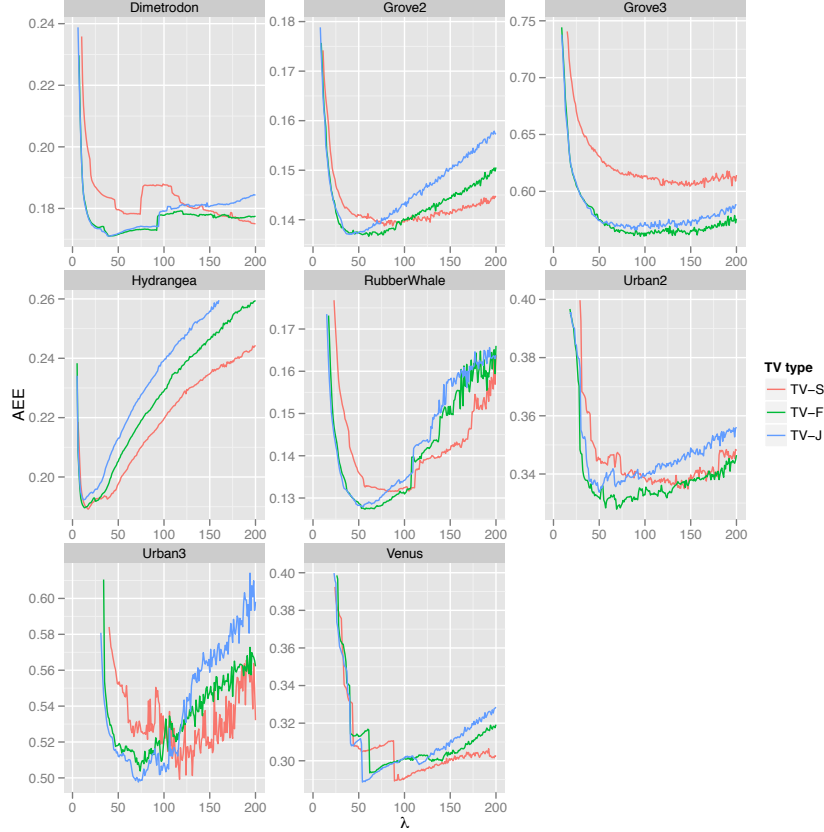because of its nice theoretical properties.

Figure 2.1: Average endpoint errors for the three different definitions of total
variation TV_S, TV_F, and TV_J on training data plotted as a function of λ.

Comparisson of methods   With the algorithmic setup in place, we can eval-
uate the accuracy of the method. For a baseline method we consider the pre-
sented algorithm with standard parameters, using a one-dimensional data term,
and TV_J for regularization. Table 2.2 shows results for this method, compared
to results of the original algorithm by Zach et al. (2007), an improved version of
this algorithm presented by Wedel et al. (2009a), which among other things uses
structure-texture decomposition of the input images to remove lighting artifacts,
and uses intermediate median filtering. Finally we compare to the results pre-
sented in Rakêt et al. (2011), that are based on a color image data term solved
by Proposition 2.1.4.

We see that despite of its simplicity, the baseline method presented here
gives slightly better average results than the ones of Wedel et al. (2009a) and

Table 2.2: Average endpoint error results for the Middlebury optical flow database training sequences for different variants of the TV-L^1 optical flow algorithm. Baseline is the method following Algorithm 2.1 with standard parameters, using a 1D data term with grayscale images, and TV$_J$ for regularization. The second method is the RGB algorithm presented in Rakêt et al. (2011). Third method is the original method of Zach et al. (2007), with results are taken from Wedel et al. (2009a). The last column holds results of the so-called TV-L^1-improved algorithm (Wedel et al. 2009a). Bold indicates the best result among the four

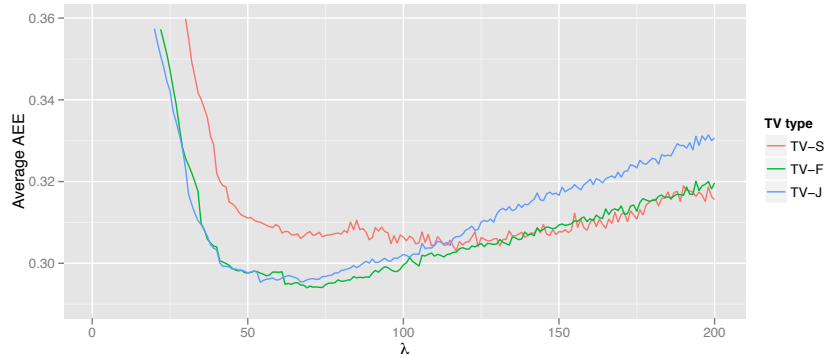|	baseline	Rakêt et al.	Zach et al.	Wedel et al.
Dimetrodon	0.17	**0.16**	0.26	0.19
Grove2	**0.14**	0.15	0.19	0.15
Grove3	**0.57**	**0.57**	0.76	0.67
Hydrangea	0.21	0.25	0.26	**0.15**
RubberWhale	0.13	0.17	0.22	**0.09**
Urban2	0.33	0.36	0.65	**0.32**
Urban3	0.51	**0.50**	1.07	0.63
Venus	0.31	0.49	0.48	**0.26**
average	**0.296**	0.331	0.486	0.308

Figure 2.2: Average AEE computed across all sequences in Figure 2.1.

Rakêt et al. (2011). This evaluation is of course not optimal, since it is based on training data, however, it seems to be a general fact that carefully optimized implementations are of great importance when computing optical flow.

Figure 2.3: Frames 10 and 11 of the *Army* sequence of the Middlebury Optical Flow Database test set.

Smoothness and visualization of optical flows   Consider the two frames from the *Army* sequence in Figure 2.3. The ground truth motion represented in color coding can be found in Figure 2.4. In this representation hue codes the direction of the flow vectors, and saturation indicates the length of the vectors. Black corresponds to occluded areas.

Figure 2.5 shows the estimates produced with the baseline method for three different values of λ. We see that with the standard choice of $\lambda = 50$ we are able to properly estimate most details. Increasing λ produces a less regular flow with more noise artifacts, and decreasing it produces flows where small structures are blurred out. In contrast, substituting the data term with the interval data term presented in Example 2.1.6, and varying the size of the interval where no penalty is given, produces another type of smooth flows. With this data term small variations are ignored in the minimization, and the estimation is driven by big differences that may be found at for example edges. This is evident from the results, where edges are preserved quite well, while the interior of objects are very smooth.
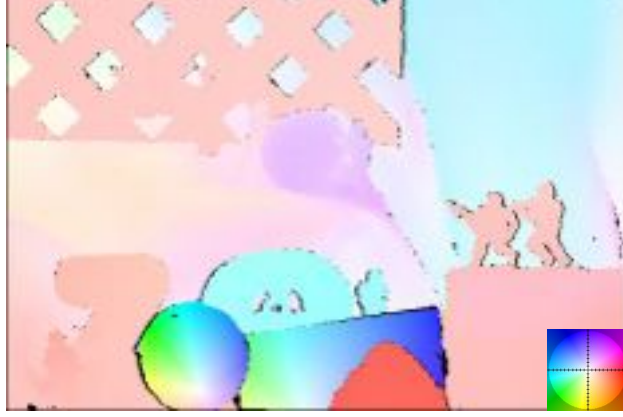
Figure 2.4: Ground truth optical flow between the two frames from Figure 2.3. Flow vectors are coded according to the color legend in the lower right corner.

| $\lambda = 10$ | $\lambda = 50$ | $\lambda = 100$ |

Figure 2.5: Optical flow between the two frames from Figure 2.3 with baseline method.

| $\lambda = 100,$ | $\lambda = 100,$ | $\lambda = 100,$ |
| $-c_1 = c_2 = 0.02$ | $-c_1 = c_2 = 0.01$ | $-c_1 = c_2 = 0.005$ |
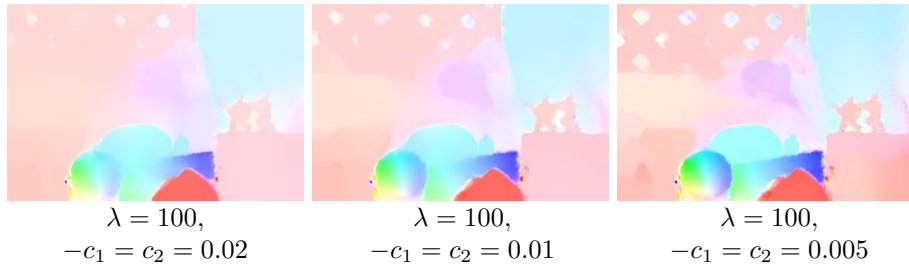
Figure 2.6: Optical flow between the two frames from Figure 2.3 with interval data term (Example 2.1.6), and everything else as in the baseline method.

Chapter 3

Applications

This chapter presents five novel applications of the optical flow methods reviewed so far. The first application considers registration of X-ray images of hands. The second application presents a specialized registration problem for 2D chromatograms, and introduces a novel solution strategy. The last three applications look into interpolation in image sequences. The first of these presents a general method where optical flow data terms are reparametrized to fit the interpolation assumptions, which turns out to produce superior results compared to conventional methods. The last two applications are related to distributed video coding. We consider interpolation when depth information of the scene is available, and finally we consider how to give new estimates of in-between frames using partially decoded information about the frame in question.

3.1 Registration of X-ray images

One of the major application areas of registration algorithms is medical images, where proper analysis of the data is often impossible without first registering the images. In this setting TV-L^1 registration offers a number of advantages over typical methods. In particular the parallel nature of the algorithms, and the associated speed when implemented on massively parallel processors, but also the robustness of the L^1 norms used in both data and regularization terms. This makes TV-L^1 a good choice for out-of-the-box registration to a large body of problems. On the downside, TV-L^1 does not necessarily produce diffeomorphic registrations, and in applications where this is of importance, one may either try to manually enforce this behavior, or consider methods for diffeomorphic registration (see for example Sommer et al. (2012) and references therein). Furthermore, if images are taken on completely different imaging devices, and cannot easily be brought on similar scales, one will prefer a data term suitable for these types of problems. One type of data term that can handle such problems is mutual information, which has recently been considered by Panin (2012). The TV-L^1 optical flow method has previously been used for registration in Pock et al. (2007), where CT scans of lungs as well as brain MRI images were registered.

Consider the X-ray images of two different hands in Figure 3.1. Registering I_1 to I_0 is by no means a simple task: The two hands have significantly different

bone structure; the contrast of the images and the amount of noise is different; and the two lead tags that are placed in the image represent information that should not be registered. This last issue may very well ruin the registration process for many types of methods. Undeterred by these facts, a registration using the standard settings with $\lambda = 10$ and one level of 3×3 median filtering has been performed. The registration results can be found in Figure 3.2. It is seen that we get a quite decent registration, with the only big artifact being caused by the lead tag in I_1. However, this artifact only causes local changes, which do not propagate down to the hand, as can be seen in the deformation visualization. Note that the lead tag in I_0 does not cause any artifacts. This is due to the fact that the corresponding area in I_1 is homogeneous, which means that the updates caused caused by the data term in this area are essentially zero (Lemma 2.1.1).

3.1.1 Registration of structural images

The problem of registering the images in Figure 3.1 is in many ways different from the optical flow problem applied to video sequences. In video data, the same objects are occurring in consecutive pairs of images, and we want to use the fine details in the images to get correct correspondences. In the case of images of two different objects, a true one-to-one correspondence does not exist, and trying to match fine details may just amount to noise matching—both in terms of actual noise, or person specific structure that may from a larger perspective be considered as inter-personal serially correlated noise structures. This means that one may benefit from filtering or regularization of the images thereby enhancing dominant structures and removing fine details. Such an approach contrasts the successful structure-texture decomposition used in several optical flow algorithms (Wedel et al. 2009a, Sun et al. 2010), where the structural part of the images obtained from regularization are subtracted from the original image. This produces an image that mainly contains texture details. The structures that are removed often include shadows and general illumination changes, which will in turn give better estimates.

Here we will consider two types of regularization. ROF regularization (Rudin et al. 1992) where we regularize the observed image I' using total variation regularization

$$E_{\mathrm{ROF}}(I) = \beta \int_{\mathscr{T}} \|I(\boldsymbol{x}) - I'(\boldsymbol{x})\|^2 \, \mathrm{d}\boldsymbol{x} + \int_{\mathscr{T}} \|\nabla I(\boldsymbol{x})\| \, \mathrm{d}\boldsymbol{x}. \qquad (3.1)$$

Furthermore, we consider a type of regularization that enhances sparsity even further, where the L^1 norm of the gradient is replaced by an L^0 norm that penalize all deviations from $\mathbf{0}$ equally

$$E_{L^0}(I) = \beta \int_{\mathscr{T}} \|I(\boldsymbol{x}) - I'(\boldsymbol{x})\|^2 \, \mathrm{d}\boldsymbol{x} + \int_{\mathscr{T}} \|\nabla I(\boldsymbol{x})\|_{\ell_0} \, \mathrm{d}\mu(\boldsymbol{x}), \qquad (3.2)$$

where μ denotes the Hausdorff measure.

The ROF model may be minimized effectively using Lemma 2.1.2, and the minimizer of (3.2) may be computed using the method described in Xu et al. (2011). As opposed to many other types of regularization, these methods are edge preserving, and the L^0 regularization of the gradient produces cartoon-like
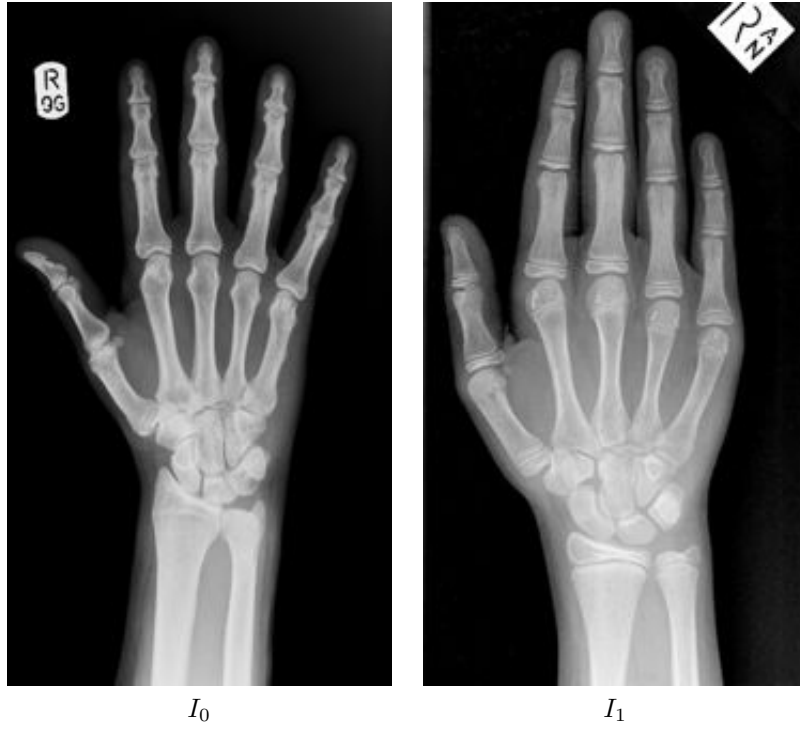
I_0 I_1

Figure 3.1: X-ray images of two different hands. Sources are http://mrmackenzie.co.uk/2011/11/01/x-rays-in-medicine/ and http://images.suite101.com/460740_com_28_hand.jpg.

I_1 registered | Deformation of coordinate system | Difference between I_0 and registered I_1
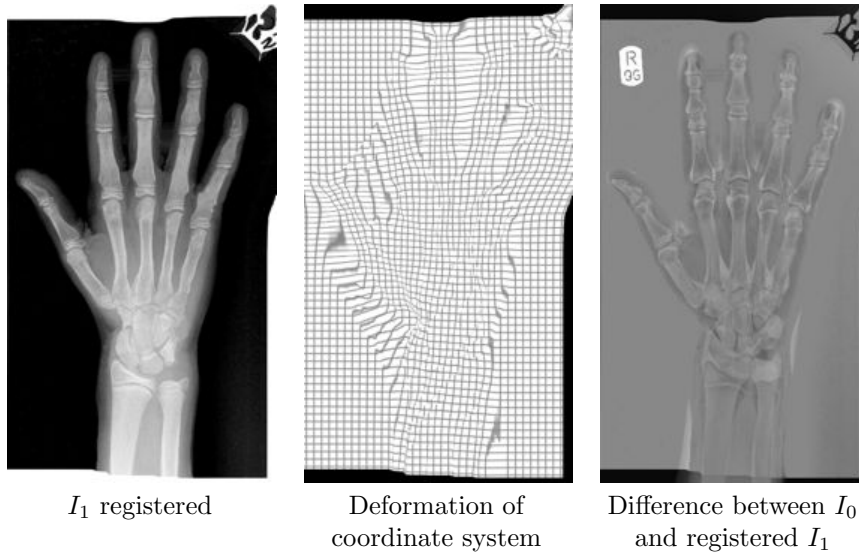
Figure 3.2: Registration of I_1 to I_0 using the standard TV-L^1 algorithm with $\lambda = 1$ and one level of 3×3 median filtering.

results with strong edges and large flat regions of zero gradient. The regularization results of these methods with respectively $\beta = 10$ and $\beta = 200$ can be found in Figure 3.3. The corresponding registrations, computed using the same parameters as before, are found in figures 3.4 and 3.5. While the results look very similar to the ones found in Figure 3.2, we see that the results for the regularized images tend to deform the coordinate system outside the hand to a greater extend than the non-regularized case. The reason for this is that the gradients driving the registration are small or zero inside the bones after regularization. This is desirable both from a practical point of view, since the purpose of these types of registrations are typically to align bones in a similar coordinate system, as opposed to bending and deforming the interior of bones to produce a good match. A somewhat similar effect can also be achieved by using the interval data term from Example 2.1.6 instead of the L^1 norm in the optical flow algorithm.

ROF regularized I_0 ROF regularized I_1

L^0 regularized I_0 L^0 regularized I_1

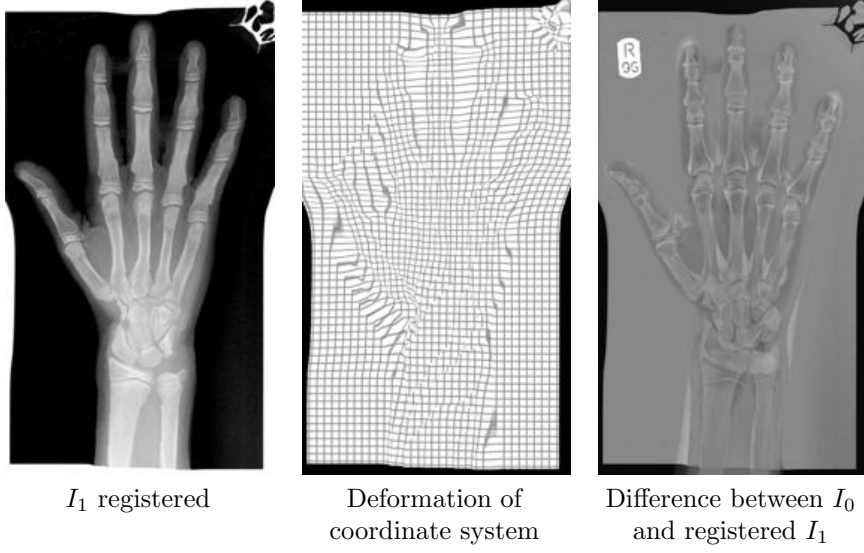Figure 3.3: Results of ROF and L^0 regularization on the images from Figure 3.1.

I_1 registered	Deformation of coordinate system	Difference between I_0 and registered I_1

Figure 3.4: Registration of I_1 to I_0 after ROF regularization using the standard TV-L^1 algorithm with $\lambda = 1$ and one level of 3×3 median filtering.
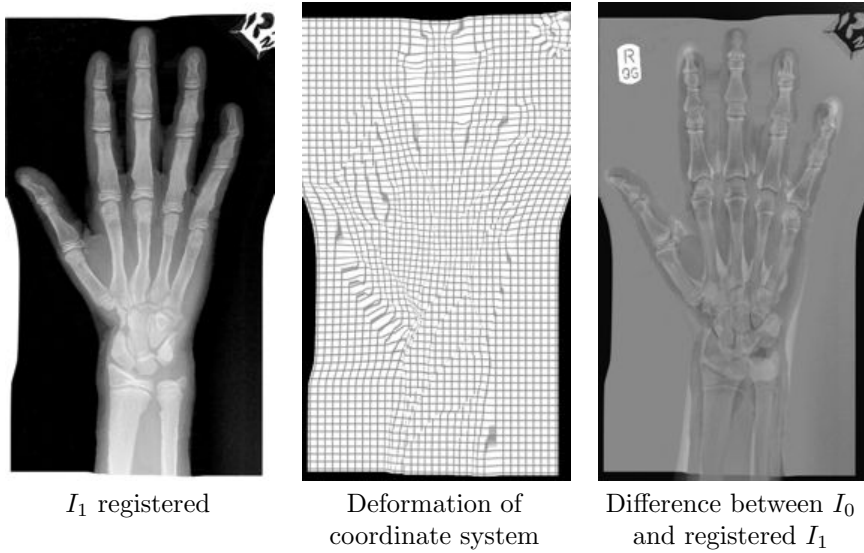

I_1 registered	Deformation of coordinate system	Difference between I_0 and registered I_1

Figure 3.5: Registration of I_1 to I_0 after L^0 regularization using the standard TV-L^1 algorithm with $\lambda = 1$ and one level of 3×3 median filtering.
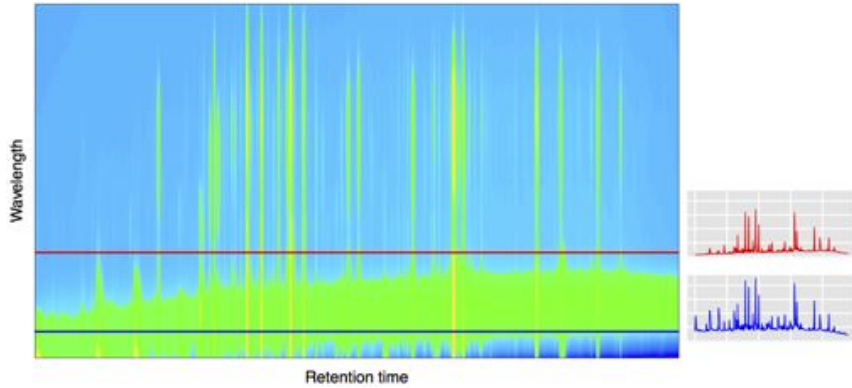
Figure 3.6: Example of a chromatogram along with the absorbance (A.U.) curves corresponding to two fixed wavelengths.

3.2 Registration of 2D chromatograms

Chromatography is a process for separating mixtures. One use of chromatography is measuring relative proportions of analytes in a number of mixtures, to determine differences. An example of a 2D chromatogram is shown in Figure 3.6. The chromatograms we are considering have been generated using ultra-high-performing liquid chromatography with diode-array-detection (Petersen et al. 2011). The chromatograms consists of 209 wavelengths each measured at 24,000 retention times. The subject of the analysis is rapeseed seedlings having been exposed to different levels glyphosate (commonly known as Roundup®).

From the images arising from this procedure will have shifts in retention time, but because of the experimental setup, no such shifts occur in the wavelength dimension. This means that we have a one-dimensional registration problem for a two-dimensional image.

Consider the single wavelength of four chromatograms shown in Figure 3.7. The retention time shifts are clearly visible. Furthermore there seem to be a varying detector sensitivity, resulting in some of the curves consistently having higher peaks that others. Finally there are small variations that cannot be explained by the mentioned issues, and which can be ascribed to serially correlated effects and noise.

3.2.1 Registration algorithm

Given two chromatograms $I_0, I_1 : \mathscr{T} \to \mathbb{R}$ of size $k \times n$, where $\mathscr{T} = \mathscr{T}_w \times \mathscr{T}_t$, consider the problem of estimating the disparity $v : \mathscr{T}_t \to \mathscr{T}_t$ such that $I_1(w, t + v(t))$ is properly registered to $I_0(w, t)$. Because of the varying detector sensitivity, a robust data term such as an L^1 norm is preferable.

From the point of view of Section 2.1.2 the natural formulation of the data term is as a vector valued problem. Let

$$\boldsymbol{I}_i(t) = \begin{pmatrix} I_i(w_1, t) \\ \vdots \\ I_i(w_k, t) \end{pmatrix}.$$
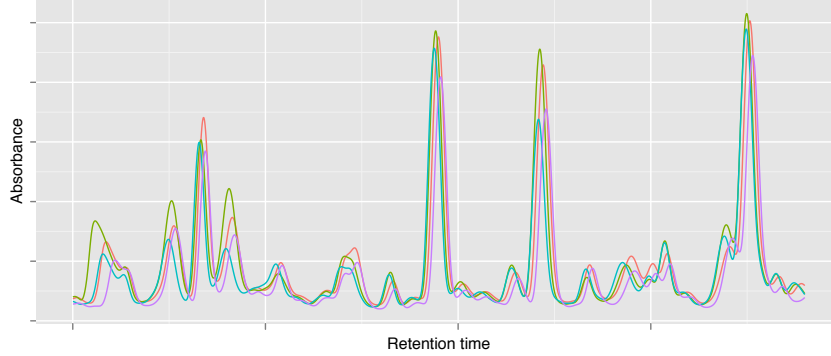
Figure 3.7: A range of retention times for a single wavelength for four chromatograms.

The optical flow constraint may then be written as

$$\boldsymbol{I}_1(t + v(t)) - \boldsymbol{I}_0(t) = \boldsymbol{0}.$$

Linearizing this around a given estimate v_0, we get the following system of equations

$$\underbrace{\partial_t \boldsymbol{I}_1(t + v_0)}_{\boldsymbol{a}} v(t) \underbrace{-\partial_t \boldsymbol{I}_1(t + v_0)v_0 + \boldsymbol{I}_1(t + v_0) - \boldsymbol{I}_0(t)}_{\boldsymbol{b}} = \boldsymbol{0}.$$

Considering an L^1 norm of this linearization of this data term, we see that the case (ii) of Proposition 2.1.4 is very easily calculated, however, it seems unlikely that it will ever be the case that $\boldsymbol{b} \in \operatorname{Im} \boldsymbol{a}$ for just a moderate number of wavelengths. This means that we will almost surely be in the less attractive case (i) where we have to minimize by some iterative procedure.

An novel alternative for registering this dataset has been described in Rakêt & Markussen (2014). The idea is to treat the one-dimensional vector valued registration problem as a two-dimensional problem, and couple the different vector channels through the regularization rather than through the data term. The method is generally applicable, and works by posing an d dimensional registration problem with data taking values in a k dimensional space, as a one-dimensional registration problem on a $d + 1$ dimensional domain. This is done by treating the vector channels as an added dimension to the domain. This way the regularization will be $d + 1$ dimensional, and by enforcing strong (or increasing) weight on the regularity across this new dimension, information is propagated between the different channels of the image to produce a registration that is homogeneous along the new dimension.

As described above, we start out by estimating disparities for each wavelength. In the given example we are interested in a robust L^1 norm for the data term. The robustness is important because of the varying detector sensitivity and serially correlated effects, where for example an L^2 norm may cause problems in relation to outliers. For regularization, we are interested in a term, that in addition to imposing regularity on the estimated disparities, regularize across wavelengths. Since one must expect drifts in retention time to be continuous,

29

the registration should be smooth, and therefore we will regularize using squared gradient magnitude instead of total variation. The energy to be minimized looks as follows

$$E(v) = \lambda \int_{\mathscr{T}} \|I_1(w, t + v_w(t)) - I_0(w, t)\| \, \mathrm{d}w \, \mathrm{d}t + \int_{\mathscr{T}} \|\nabla_{w,t} v_w(t)\|^2 \, \mathrm{d}t.$$

This functional is minimized following the methods described in Chapter 2 where the data term is iteratively approximated by its first-order Taylor approximation around the given estimate v_0^w

$$\rho(v)(w, t) = \partial_t I_1(w, t + v_0^w(t))(v(t) - v_0^w(t)) + I_1(w, t + v_0^w(t)) - I_0(w, t).$$

Furthermore data-fidelity and regularization are decoupled by means of a quadratic proximity term

$$E(\boldsymbol{v}, \boldsymbol{v}') = \lambda \int_{\mathscr{T}} \|\rho(v_w)(w, t)\| \, \mathrm{d}w \, \mathrm{d}t$$
$$+ \frac{1}{2\theta} \int_{\mathscr{T}} \|v_w(t) - v_w'(t)\|^2 \, \mathrm{d}t + \int_{\mathscr{T}} \|\nabla_{w,t} v_w'(t)\|^2 \, \mathrm{d}t$$

where θ is sufficiently small. Using Proposition 2.1.4, the pointwise solution in v_w is found to be

$$v_w(t) = v_w'(t) - \lambda\theta \begin{cases} -\partial_t I_1(w, t + v_0^w(t)) & \text{if } \frac{\rho(v)(w,t)}{\lambda\theta} < -|\partial_t I_1(w, t + v_0^w(t))|^2 \\ \partial_t I_1(w, t + v_0^w(t)) & \text{if } \frac{\rho(v)(w,t)}{\lambda\theta} > |\partial_t I_1(w, t + v_0^w(t))|^2 \\ \frac{\rho(v)(w,t)}{\partial_t I_1(w, t + v_0^w(t))} & \text{if } \frac{|\rho(v)(w,t)|}{\lambda\theta} \leq |\partial_t I_1(w, t + v_0^w(t))|^2 \end{cases}.$$

The problem in v_w' is just a standard Tikhonov regularization problem, and can easily be solved using standard methods. E is minimized iteratively in a coarse-to-fine manner, where the input images and the corresponding disparities are gradually upsampled in the retention time dimension, but the wavelength dimension is kept at its original size. Following Algorithm 2.1 we use $\ell_{\max} = 160$ and a scaling factor between levels of 0.97, yielding a downsampling factor at the coarsest level of approximately 130. $w_{\max} = 100$ warps are performed at each level, and λ was set to 60, while θ was fixed at 0.1.

Figure 3.8 shows the individual wavelength registration curves (gray) of the described method, as well as the average (red) for two 2D chromatograms. The weighting of the wavelength dimension of the gradient in the Tikhonov regularization are respectively a factor 0 (i.e. registering each wavelength independently) and 10. As we can see, with the higher weight we are able to propagate information between wavelengths very well, and end up with a uniform result across wavelengths. Note in addition that the average curves are quite different in the two cases.

The average registration is used as the final single disparity $v : \mathscr{T}_t \to \mathscr{T}_t$. The registration was then done by warping the chromatograms according to v for each wavelength. The result of the registration procedure on the data in Figure 3.7 can be found in Figure 3.9. We see that the data is very well aligned after registration.
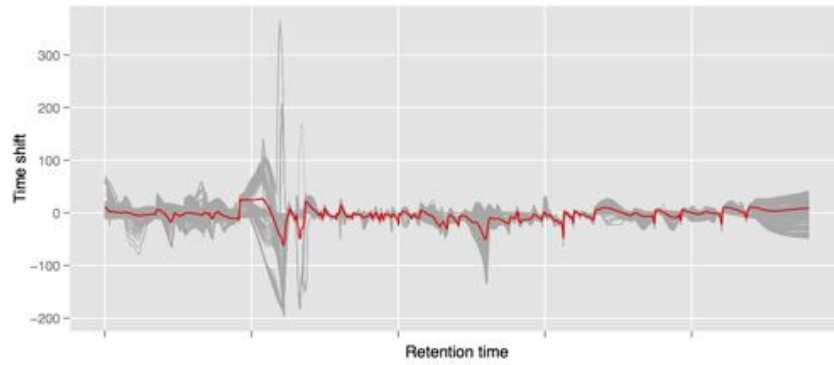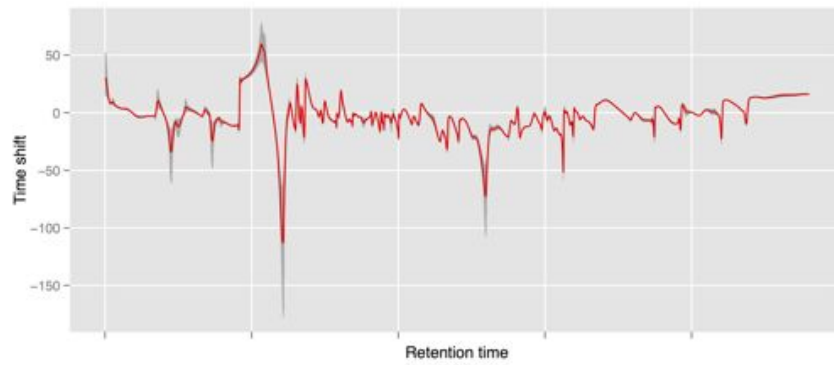
Figure 3.8: Registration curves of individual wave lengths (gray) with the average registration plotted on top (red).
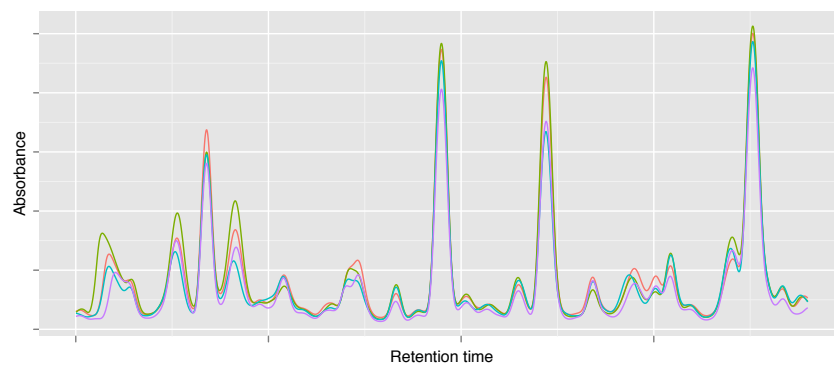

Figure 3.9: The chromatograms from Figure 3.7 registrated along retention time.

3.3 Image interpolation with a symmetric optical flow constraint

Frame interpolation is the process of creating intermediate images in a sequences of known images. The process has many uses, for example video post-processing and restoration, temporal upsampling in HDTVs to enhance viewing experience, as well as a number of more technical applications, in for example in video coding (Girod et al. 2005, Huang et al. 2011).

In this section we will review optical flow based frame rate upsampling which performs interpolation along the motion trajectories. In particular, we will review the method presented in Rakêt et al. (2012a), where the optical flow energy is reparametrized such that it fits better to the given problem. The reparametrized energy has a symmetric data fidelity term, that uses both surrounding frames as references. We show that one can improve modern frame interpolation methods substantially by this powerful generic trick, that can be incorporated in existing schemes without requiring major adaptations. We analyze the reparametrization, and show experimentally that it has a substantial effect on the stability and robustness of the interpolation process.

The idea to symmetrize data matching terms to achieve better results has already established its usefulness in other areas. In image registration Christensen & Johnson (2001) explored the benefit of penalizing consistency, by jointly estimating forward and backward transforms, and requiring that they were inverses of one another. A similar idea was applied to the optical flow problem by Alvarez et al. (2007a), who imposed an additional consistency term. Later that same year Alvarez et al. (2007b) proposed a reparametrization similar to the one derived here, in order to avoid a reference frame, and thereby increase flow consistency. However, they did not use the obtained symmetric flow directly, but interpolated flow values at pixel position of a reference image in order to obtain a flow comparable to the standard asymmetric flow. Recently Chen (2012) used a symmetric data term for surface velocity estimation, noting the property that motion vector length is halved, which in turn gives better handling of large displacements.

Apart from being algorithmically different, the difference between the justification for the reparametrization given here and the justifications of Alvarez et al. (2007b) and Chen (2012) is that we have chosen the symmetric data fidelity term because it explicitly models the standard interpolation assumption, rather than improves some notion of consistency, or better handles large displacements. In turn this also means that we may use the estimated flows directly on the unknown frame, and thereby avoid the problems related to temporal warping. As we will show, the mentioned benefits are clearly reflected in the results. It is demonstrated that using a symmetric flow for interpolation is generally better than using either forward or backward flows or both.

3.3.1 Motion Compensated Frame Interpolation

Given two images I_0 and I_1 and an estimate of the (forward) optical flow v_f we are interested in estimating the in-between image $I_{1/2}$ (the methods are easily extended to any in-between frame I_t, $t \in (0,1)$). A simple approach is to assume that the motion vectors are linear through $I_{1/2}$ and then fill in $I_{1/2}$ using the computed flow. However, since v_f is of sub-pixel accuarcy, the points $\boldsymbol{x} + {}^{1}/{}_{2}v_f(\boldsymbol{x})$ that are hit by the motion vectors are generally not pixel positions. This is often solved by warping the flow to the temporal position of the intermediate frame $I_{1/2}$ (Baker et al. 2011, Herbst et al. 2009, Werlberger et al. 2011), in which one defines a new flow $v_f^{1/2}$ from $I_{1/2}$ to I_1

$$v_f^{1/2}(\text{round}(\boldsymbol{x} + {}^{1}/{}_{2}v_f(\boldsymbol{x}))) = {}^{1}/{}_{2}v_f(\boldsymbol{x}), \qquad (3.3)$$

where the round function rounds the argument to nearest pixel value in the domain. There are some drawbacks to this approach. First, if the area around $\boldsymbol{x}$ in I_0 is occluded in I_1, there are likely multiple flow candidates assigned to the point $\text{round}(\boldsymbol{x} + {}^{1}/{}_{2}v_f(\boldsymbol{x}))$. In the converse situation, i.e. dis-occlusion from I_0 to I_1 there may be pixels that are not hit by a flow vector, thus leaving holes in the flow. While the first problem can be solved by choosing the candidate vector with the best data fit, that is the candidate v_f for which $\|I_1(\boldsymbol{x}+v_f(\boldsymbol{x}))-I(\boldsymbol{x})\|$ is smallest, the solution for the problem of dis-occlusions in not that simple. Here we will simply fill the holes in the flow field by an outside-in filling strategy. With a dense flow we can then interpolate $I_{1/2}$ using the forward scheme

$$I_{1/2}(\boldsymbol{x}) = \frac{1}{2}\left(I_0(\boldsymbol{x} - v_f^{1/2}(\boldsymbol{x})) + I_1(\boldsymbol{x} + v_f^{1/2}(\boldsymbol{x}))\right), \qquad (3.4)$$

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

One can sophisticate the interpolation methods by estimating occluded regions and selectively interpolating from the correct frame. We will not pursue any occlusion reasoning here, but refer to Herbst et al. (2009) and Stich et al. (2008) for details.

3.3.2 Reparametrizing Optical Flow for Interpolation

The approach presented in the previous section is the standard procedure for frame interpolation and serves as backbone in many algorithms (Baker et al. 2011, Huang et al. 2011, Keller et al. 2010, Werlberger et al. 2011). In this section we will reparametrize the original energy functional so the recovered flow is better suited for interpolation purposes. The reparametrization turns out to be beneficial on a number of levels: It makes the temporal warping of the flow superfluous, eliminates the need to calculate flows in both directions, improves handling of large motion, and increase overall robustness.

The original optical flow energy functional take as argument an optical flow v that is defined on a continuous domain. In practice, however, we only observe images at discrete pixels, and the optical flow is typically only estimated at the points corresponding to the pixels in I_0. Since we assume that the intermediate

frame $I_{1/2}$ can be obtained from linearly following the flow vectors, we propose to reparametrize the data fidelity functional of the TV-L^1 optical flow energy using this assumption, so that it is given as

$$\frac{1}{2} \int_{\mathscr{T}} \|I_1(\boldsymbol{x} + v_s(\boldsymbol{x})) - I_0(\boldsymbol{x} - v_s(\boldsymbol{x}))\| \, \mathrm{d}\boldsymbol{x}. \tag{3.5}$$

We note that in this parametrization, the coordinates of the optical flow matches those of the intermediate frame $I_{1/2}$, and using this data term will thus eliminate the need for warping of the flow, since interpolation can directly be done similarly to (3.4). Because the motion vectors of the symmetric flow v_s are only half of the ones of the forward or backward flows, we need to halve the corresponding λ to keep comparison fair, which is the reason for the factor $1/2$.

Linearizing the data matching term (3.5) around v_0 gives

$$\rho(v_s) = I_1(\cdot + v_0) - I_0(\cdot - v_0) + (J_{I_1}(\cdot + v_0) + J_{I_0}(\cdot - v_0))(v_s - v_0) \tag{3.6}$$

which is of the form (2.14). For grayscale images the corresponding split energy term (2.8) is easily minimized using Lemma 2.1.1, and in general using the L^1-L^2 minimization described in Proposition 2.1.4.

The differences between (3.6) and the conventional linearization are that we now allow sub-pixel matching in both surrounding images, and instead of a single Jacobian we have a sum of two. Thinking of this linearization as a finite difference scheme corresponding to a linearized differential form of the data fidelity term (Horn & Schunck 1981), we see that the temporal derivative is represented by a central finite difference scheme, as opposed to the typical forward differences. In addition the sum of the two Jacobians should make the estimation procedure more robust to noise, as the noise amplification caused by derivative estimation is now averaged over two frames—a fact that has previously been used heuristically to improve accuracy in asymmetric flow estimation (Wedel et al. 2009a). Finally we note that the motion vectors will only have half the length of the ones obtained from the regular parametrization. This will make the method better suited to handle large displacements compared to traditional methods that only make use of a one-sided linearization.

3.3.3 Results

Motion compensated frame interpolation finds many uses, ranging from the more technical applications such as video coding (Girod et al. 2005, Huang et al. 2011, Luong et al. 2012) to disciplines like improving viewing experience (Keller et al. 2010) or restoration of historic material (Werlberger et al. 2011). For the former type of application the reconstruction quality in terms of quantitative measures is of great importance. For the latter types it is hard to devise specific measures of quality, as the human visual system is very tolerant to some types of errors, while it instantly notices other types.

For the results presented in the following, the optical flows have been computed using the algorithm illustrated in Algorithm 2.1, with standard parameters, except: On each level $w_{\max} = 60$ warps are performed with $i_{\max} = 5$ inner iterations of a Bermùdez-Moreno type algorithm for minimizing the vectorial total variation TV_J.
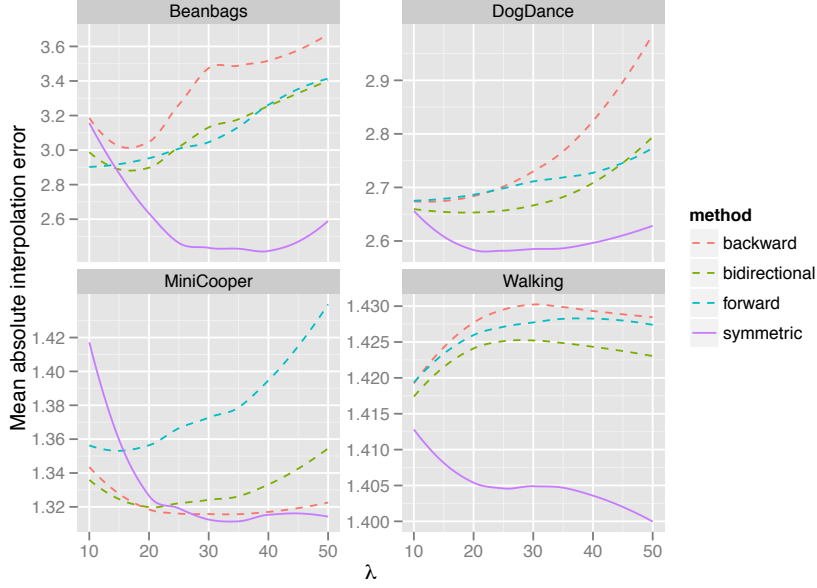
Figure 3.10: Performance for varying λ on the four *High-speed camera* training sequences from the Middlebury Optical Flow benchmark.

As a first experiment we compare the four different types of interpolation suggested in the previous sections, on the four *High-speed camera* training sequences of the Middlebury Optical Flow benchmark (Baker et al. 2011). Figure 3.10 shows the effect of varying the data term weight λ in terms of the mean absolute interpolation error (MAIE). We see that the symmetric flow outperforms the conventional approaches, and that it is typically less sensitive in terms of the choice of λ. In particular we see that the difficult Beanbags sequence which contains large displacements is handled much better by the symmetric scheme. By evaluation on the Middlebury training set it was found that $\lambda = 35$ gave the best overall performance for the symmetric flow, and that $\lambda = 20$ gave the best performance for the other three methods. These λ values will be used in the rest of the experiments presented in this section.

Consider as a second example the results of interpolation under noise presented in Figure 3.11. This figure shows the mean square interpolation error performance of the four methods on the Beanbags sequence with additive $\mathcal{N}(0, \sigma^2)$ noise. The improved robustness of the symmetric interpolation method is clearly visible from the distances between the MAIEs to the asymmetric methods that increase as the standard deviation of the noise increases. In addition we see that the variance of the MAIEs across the independent replications is significantly lower for the symmetric method compared to the three other methods.

Now consider, as a third example, the frames given in Figure 3.12. The sequence has large displacements (> 35 pixels) and severe deformations, which makes the estimation of $I_{1/2}$ very difficult. Figure 3.13 shows the three different flows v_f, v_b and v_s along with the corresponding interpolated frames. Zoom ins of details can be found in Figure 3.14. We see that the result generated by the
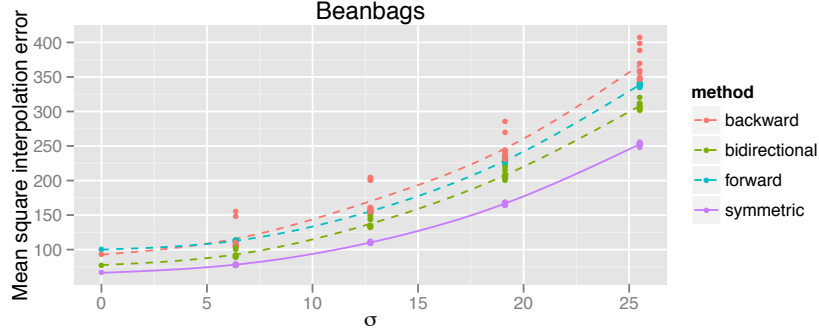
Figure 3.11: Mean square interpolation error performance under additive $\mathcal{N}(0, \sigma^2)$ noise for varying σ. Results are for the Beanbags sequence, and are based on 10 independent replications.

symmetric flow is visually more pleasing than the ones produced by the forward and backward flows, a fact that is also clearly reflected in the MAIEs and root mean square interpolation errors (RMSIE).

Finally let us compare the method to some methods of the current state-of-the-art. Table 3.1 holds the RMSIEs for six sequences from the Middlebury Optical Flow benchmark and results for a number of methods. While the results cannot fully match those of Stich et al. (2008), which gives significantly better results on 3 of the sequences, our method outperforms all other approaches, including the recent and much more complex methods of Chen & Lorenz (2011) and Werlberger et al. (2011).

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

Figure 3.12: Frames 7 (I_0), 10 ($I_{1/2}$) and 13 (I_1) of the Mequon sequence.

Forward	Backward	Symmetric

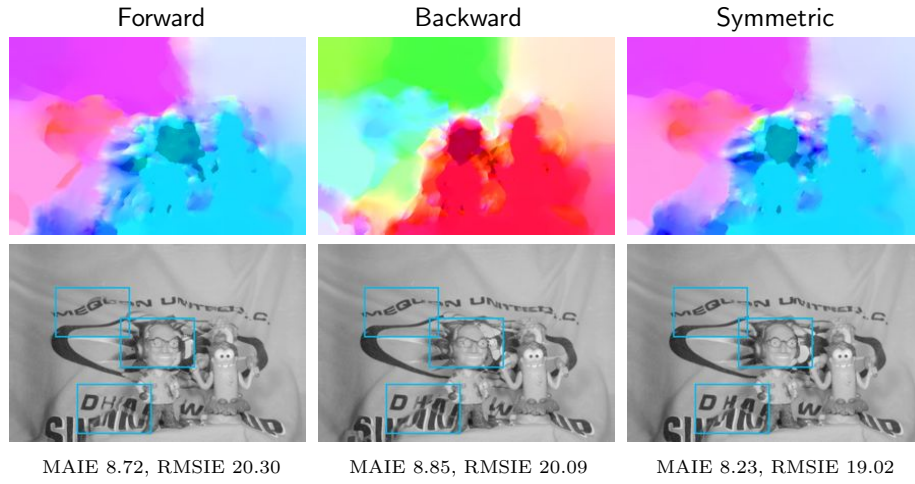| MAIE 8.72, RMSIE 20.30 | MAIE 8.85, RMSIE 20.09 | MAIE 8.23, RMSIE 19.02 |

Figure 3.13: Results for the Mequon sequence. Top row: Color coded optical flows, buttom row: Interpolation results. Zoom ins of details can be found in Figure 3.14.

Ground truth	Forward	Backward	Symmetric

Figure 3.14: Details of the interpolated Mequon frame from Figure 3.13.

Table 3.1: RMSIE for different Middlebury sequences. Bold indicates the best result. [†]Results are taken from Stich et al. (2008). [‡]Marked algorithms have not been implemented by their respective authors, but are based on alternative implementation on the Middlebury Optical Flow database.

Method	Dimetrodon	Venus	Hydrangea	RubberWhale
Symmetric TV-L^1	1.93	3.45	3.36	**1.46**
Chen & Lorenz (2011)	1.95	3.63	—	—
Werlberger et al. (2011)	1.93	—	—	—
Stich et al. (2008)	**1.78**	**2.88**	**2.57**	1.59
Bruhn et al. (2005)[†,‡]	2.59	3.73	—	—
Zitnick et al. (2005)[†]	3.06	5.33	—	—
Black & Anandan (1996)[†]	2.56	3.93	—	—
Pyramid Lukas-Kanade[†,‡]	2.49	3.67	—	—

Method	MiniCooper	Walking
Symmetric TV-L^1	**3.96**	**2.89**
Werlberger et al. (2011)	4.55	3.97

Figure 3.15: Frame 90 of the *Breakdancers* sequence from Microsoft Research, with its corresponding depth map.

3.4 Image interpolation using depth data

In recent years, the 3DTV and 2D-plus-depth formats have seen an increasing popularity. While conventional methods for frame interpolation usually work on a pair of outer images, similar to the one to be interpolated, these hybrid formats offer some new challenges and opportunities.

Distributed video coding (Girod et al. 2005) provides an interesting application of frame interpolation, and variations of both standard and symmetric optical flow based methods have been used in this area (Huang et al. 2011, Rakêt et al. 2012b). In distributed video coding, a set of key frames is coded using conventional coding, and intermediate frames are coded using Wyner-Ziv coding (Aaron et al. 2002). From the point of view of this application, the key frames can be considered given, and in order to decode the intermediate Wyner-Ziv frames, estimates to be used as side information in the decoding process must be generated. For the 2D-plus-depth format, one may code the depth map very efficiently (Zamarin & Forchhammer 2012), and use the depth frames in addition to the given key frames. This is beneficial, since depth frames usually contain most of the information needed for proper motion estimation, as can be seen in Figure 3.15. The following sections describe a scheme that was devised for Salmistraro, Rakêt, Zamarin, Ukhanova & Forchhammer (2013). The converse scheme, where texture frames are used to model depth movement has been investigated in Salmistraro, Zamarin, Rakêt & Forchhammer (2013).

3.4.1 Optical flow computation using brightness and depth

Denote by I_{t-1}, I_{t+1} the two (brightness) key frames, which we want to use for interpolating the intermediate frame I_t. Furthermore, let two depth frames D_t, and D_{t+1} be given. We can generalize the approach presented in Section 3.3, so in addition to the symmetric data term (3.5) we also include an asymmetric

term for the depth frames to the energy

$$
\begin{aligned}
E(v) = {} & \lambda_1 \int_{\mathscr{T}} \|I_{t+1}(\boldsymbol{x} + v(\boldsymbol{x})) - I_{t-1}(\boldsymbol{x} - v(\boldsymbol{x}))\| \, \mathrm{d}\boldsymbol{x} \\
& + \lambda_2 \int_{\mathscr{T}} \|D_{t+1}(\boldsymbol{x} + v(\boldsymbol{x})) - D_t(\boldsymbol{x})\| \, \mathrm{d}\boldsymbol{x} \\
& + \int_{\mathscr{T}} \|\mathscr{D}_J v(\boldsymbol{x})\| \, \mathrm{d}\boldsymbol{x}.
\end{aligned}
\tag{3.7}
$$

With two data terms, this energy does not fit in to any of the methods described so far unless $\lambda_1 = 0$ or $\lambda_2 = 0$. For a sum of two L^1 norms, the solution may however be found explicitly, by means of the results in Wedel et al. (2008).

The advantage of the full formulation (3.7), as opposed to a purely symmetric data term ($\lambda_2 = 0$), is that we have a smaller temporal gap on the depth data, which means that we may recover some of the non-linear motions between the two key frames. In addition, this smaller gap produces a better estimate, as the displacements are smaller. The outer key frames may then help getting the apparent motion right where the depth frames does not supply enough information. This may for example be texture, and shadows that does not show up in the depth maps.

3.4.2 Interpolation

When interpolating a frame in an image sequence, we are interested in using information from both surrounding frames. Thus we compute the forward flow v_f as described in the previous section, and a backward flow v_b, where we interchange I_{t+1} and I_{t-1} and replace D_{t+1} with D_{t-1} in (3.7).

The results are asymmetric because of the depth information, and thus a symmetric interpolation such as (3.4) should not be used. Instead we simply interpolate by

$$
I_t(\boldsymbol{x}) = \frac{1}{2}\big(I_{t-1}(\boldsymbol{x} + v_b(\boldsymbol{x})) + I_{t+1}(\boldsymbol{x} + v_f(\boldsymbol{x}))\big)
$$

where the sub-pixel locations are evaluated using bicubic interpolation.

3.4.3 Results

With the energy (3.7) it is natural to consider three distinct cases. These will be denoted by T2T ($\lambda_2 = 0$), D2T ($\lambda_1 = 0$), and DT2T ($\lambda_1 \neq 0$, $\lambda_2 \neq 0$).

The motion estimates are recovered following Algorithm 2.1 with standard settings, except that $\ell_{\max} = 65$, $i_{\max} = 10$, and $\sigma = 0.5$ for T2T and DT2T, while $\sigma = 0.35$ for D2T. For the T2T method λ_1 was set to 40, for D2T $\lambda_2 = 30$, and $\lambda_1 = 5$, $\lambda_2 = 40$ for DT2T.

We evaluate the method on the sequences *Breakdancers* and *Ballet* from Microsoft Research (Zitnick et al. 2004), and *Dancer* from Nokia Research. We use the central view of the three sequences, at 15 fps downsampled to CIF resolution.

In Table 3.2 we see that the symmetric interpolation T2T gives the worst results, and that D2T that only uses depth improves the average interpolation

quality by 1.4 dB. Combining both the symmetric and depth information in DT2T, we gain an additional 0.8 dB.

Table 3.2: Average peak-signal-to-noise-ratio of interpolated frames for the three different methods for the first 100 frames of the sequences.

Method	Ballet	Breakdancers	Dancer
T2T	34.7	27.5	30.5
D2T	37.0	28.5	31.4
DT2T	38.0	29.0	32.3

Figure 3.16 shows an interesting example from the *Dancer* sequence. In this example there are large movements of shadows, that are not visible in the depth images. The optical flows for the three methods, along with the interpolated frames can be found in Figure 3.17. We see that the shadow on the wall is interpolated quite well for the T2T method, while the movement of the dancer is less well modeled due to the large temporal distance between the key frames. For the D2T method, the movements of the dancer are well captured, but no movement is identified in the wall area, and the interpolation is just the average of the outer frames. The DT2T method identifies both the movement of the dancer as well as the shadows, and gives a better overall result than the two other methods.

Figure 3.16: Frames 94, 95, and 96 and corresponding depth maps of the *Dancer* sequence from Nokia Research.
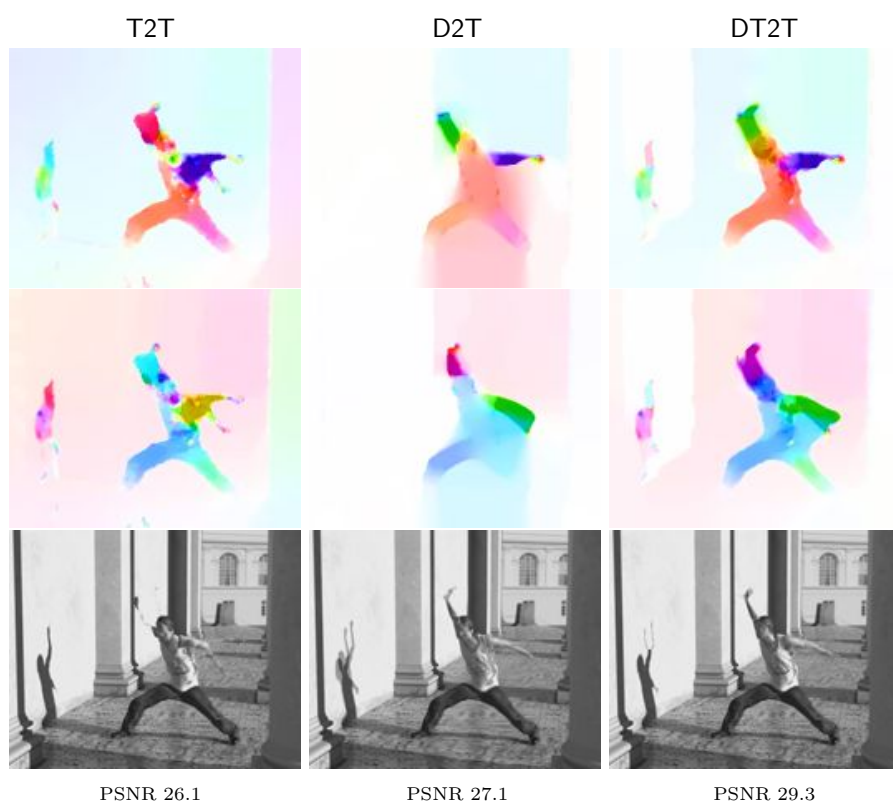
T2T	D2T	DT2T

PSNR 26.1   PSNR 27.1   PSNR 29.3

Figure 3.17: Estimated forward (top) and backward (middle) flow fields, along with the interpolation result (bottom).

3.5 Image interpolation using partially decoded frames

In this section we will consider the refinement of motion estimation and interpolation in a distributed video coding setup. We use transform domain coding with a discrete cosine transform (DCT) like transform. In this setup every decoded bit plane will produce affine constraints on the frame to be decoded, that can be used to refine the estimate of the frame to be decoded. This chapter is based on methods developed for the codec described in Luong et al. (2013).

3.5.1 Initial frame interpolation

For the initial interpolation we are in the setting of Section 3.3. We have two images I_{t-1} and I_{t+1}. The interpolation is done following Rakêt et al. (2012b), where the forward, backward and symmetric interpolations presented in Section 3.3 are computed, and the final result is taken to be the average of the three.

Figure 3.18: Frames 84, 85, and 86 of the *Soccer* sequence

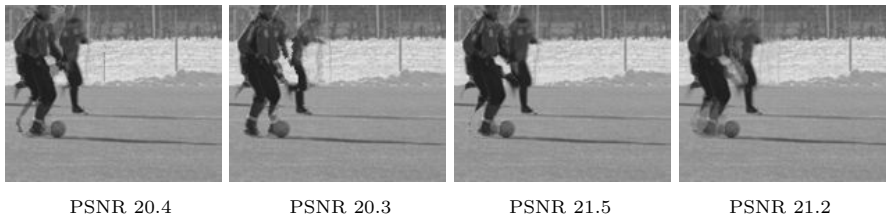| PSNR 20.4 | PSNR 20.3 | PSNR 21.5 | PSNR 21.2 |

Figure 3.19: Forward, backward, and symmetric interpolation results and average, for frame 84 of the *Soccer* sequence.

A specialized coarse-to-fine pyramidal implementation of the above algorithm is used. Following Algorithm 2.1, we have to following modifications to the standard settings: On each level we perform $w_{\max} = 30$ warps using $i_{\max} = 10$ iterations of the BM algorithm of Goldlücke et al. (2012). In order to improve interpolation quality, ρ has been weighted by the gradient magnitude $\|\nabla I_1(\boldsymbol{x} + \boldsymbol{v}_0) + 0.01\|$ (slightly shifted to avoid division by 0) in the minimization of (2.4) (Zimmer et al. 2011), which will allow for more even step sizes in the estimation. With this modification, λ was set to 3.

All in all this produces a more robust flow for interpolation, and combining the symmetric flow with the warped forward and backward flows, we propose

to do the interpolation as follows

$$
\begin{aligned}
I_{1/2}(\boldsymbol{x}) = \frac{1}{6}(&I_1(\boldsymbol{x} + \boldsymbol{v}_f^{1/2}(\boldsymbol{x})) + I_1(\boldsymbol{x} - \boldsymbol{v}_b^{1/2}(\boldsymbol{x})) + I_1(\boldsymbol{x} + \boldsymbol{v}_s(\boldsymbol{x})) \\
+ \ &I_0(\boldsymbol{x} - \boldsymbol{v}_f^{1/2}(\boldsymbol{x})) + I_0(\boldsymbol{x} + \boldsymbol{v}_b^{1/2}(\boldsymbol{x})) + I_0(\boldsymbol{x} - \boldsymbol{v}_s(\boldsymbol{x}))),
\end{aligned}
\tag{3.8}
$$

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

We will evaluate (3.8) which we will denote 3OF on the test sequences (QCIF, 15 fps) *Coastguard* QP=26, *Foreman* QP=25, *Hall* QP=24 and *Soccer* QP=25, where we interpolate every other frame and compare to the overlapped block motion compensation (OBMC) method from Huang & Forchhammer (2005) and the TV-L^1 optical flow (OF) method from Huang et al. (2011). The results are found in Table 3.3, and it can be seen that the proposed method outperforms OBMC and OF on all sequence, with an average increase in PSNR of 1.26 dB over OBMC and 2.25 dB over OF.

Table 3.3: Average PSNR across the 74 interpolated frames for the four test sequences.

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

With initial estimates produced by 3OF we may decode the so-called Wyner-Ziv frames. Decoding is done one bit plane at the time, going from most significant to least significant bit in transform domain. In the next section it is considered how one may link this information to the frame to be decoded in pixel domain.

3.5.2 Upsampling images from DCT coefficients

The frames to be decoded have been transformed using the following DCT-like transform on every 4×4 image patch $\boldsymbol{I}$

$$
\mathrm{DCT}_0(\boldsymbol{I}) = (\boldsymbol{C}\boldsymbol{I}\boldsymbol{C}^\top) \odot \boldsymbol{E},
$$

where $\odot$ denote pointwise multiplication, and

$$
\boldsymbol{C} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix}, \qquad \boldsymbol{E} = \begin{pmatrix} 1/4 & 1/\sqrt{10} & 1/4 & 1/\sqrt{10} \\ 1/\sqrt{10} & 2/5 & 1/\sqrt{10} & 2/5 \\ 1/4 & 1/\sqrt{10} & 1/4 & 1/\sqrt{10} \\ 1/\sqrt{10} & 2/5 & 1/\sqrt{10} & 2/5 \end{pmatrix}.
$$

We see that the DCT at a single point (i, j) can be written as

$$
\underbrace{\boldsymbol{E}_{ij}\boldsymbol{C}_i \otimes \boldsymbol{C}_j}_{\boldsymbol{a}} \boldsymbol{I}^v
$$

where $\boldsymbol{I}^v$ is the vectorization

$$\boldsymbol{I}^v = \begin{pmatrix} I_{00} \\ I_{10} \\ \vdots \\ I_{33} \end{pmatrix}.$$

Decoding is done bit plane wise for the image in question, and gives a set of intervals $d_1 \pm c_1, \ldots, d_k \pm c_k$ (with corresponding DCT vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k$) wherein the DCT coefficients lie. To reconstruct a 4×4 image patch $\boldsymbol{I}$ we can use the fact that the vectorized patch should fulfill

$$A\boldsymbol{I}^v \in \underbrace{[d_1 - c_1, d_1 + c_1] \times \cdots \times [d_k - c_k, d_k + c_k]}_{D} \qquad A = \begin{pmatrix} \boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{a}_k \end{pmatrix}.$$

A simple solution for reconstructing a 4×4 patch consists in ignoring the intervals, and just focusing on the midpoints, then choosing the vectorized patch $\widetilde{\boldsymbol{I}}^v$ as the solution to

$$A\widetilde{\boldsymbol{I}}^v = \begin{pmatrix} d_1 \\ \vdots \\ d_k \end{pmatrix}$$

with the least Euclidian norm, which is simply given as the Moore-Penrose pseudoinverse of the midpoint vector

$$\widetilde{\boldsymbol{I}}^v = A^\dagger \begin{pmatrix} d_1 \\ \vdots \\ d_k \end{pmatrix}.$$

Example 3.5.1. Given $c_1 = \mathrm{DCT}_0(\boldsymbol{I})(0,0)$, $A = \frac{1}{4}\begin{pmatrix} 1 & \cdots & 1 \end{pmatrix}$, and

$$A^\dagger = \tfrac{1}{4}\begin{pmatrix} 1 & \cdots & 1 \end{pmatrix}^\top,$$

which means that $\widetilde{\boldsymbol{I}}$ is a constant block with values $\frac{d_1}{4}$, in other words we fill in the patch in pixel domain by its average. ○

Example 3.5.2. Given an estimate $\boldsymbol{I}^v$ subject to the constraint specified by a vector $\boldsymbol{a}$, we can first check if the estimate is admissible by checking if

$$|\boldsymbol{a}^\top \boldsymbol{I}^v - d| \le c.$$

If so, we will not do anything. When this is not the case we can project the solution onto the planar strip given by the constraint.

Assuming that

$$\boldsymbol{a}^\top \boldsymbol{I}^v - d > c,$$

we want to compute the orthogonal projection of $\boldsymbol{I}^v$ onto

$$\{\boldsymbol{x} \ : \ \boldsymbol{a}^\top \boldsymbol{x} - d = c\},$$

which is equivalent to computing the projection of $\boldsymbol{I}^v - (d + c)$ onto the line given by $\boldsymbol{a}$ plus $d + c$, which is

$$\frac{\boldsymbol{a}^\top (\boldsymbol{I}^v - (d + c))}{\boldsymbol{a}^\top \boldsymbol{a}} \boldsymbol{a} + d + c.$$

○

Upsampling using global regularization Consider now the case of an entire image I, with the 4×4 patches $\boldsymbol{I}_1, \ldots, \boldsymbol{I}_n$, and corresponding DCT intervals $D_1, \ldots, D_n$. We can consider the problem of reconstructing all patches subject to a global roughness penalty. This means that the reconstructed patches interact to give a result that is regular across the entire image. In addition the regularization will make the problem well-posed. A choice of regularization that is both simple and powerful is total variation. The problem may then be formulated as wanting to recover I as the minimizer of

$$\int_{\mathscr{T}} \|\nabla I(\boldsymbol{x})| \, \mathrm{d}\boldsymbol{x} \qquad \text{such that } A\boldsymbol{I}_i \in D_i \text{ for all } i. \tag{3.9}$$

Direct minimization of this problem is not feasible, and instead we propose the following procedure. Starting out from an initial solution, we find a nearby solution with better regularity properties, using the algorithm of Chambolle (2004). All 4×4 blocks of the resulting regularized solution are projected onto the set of admissible solutions, and using this new initial solution we repeat the algorithm. In short, the algorithm is simply an iterative minimization of the energy

$$E(I) = \beta \int \|I(\boldsymbol{x}) - I'(\boldsymbol{x})\|^2 \, \mathrm{d}\boldsymbol{x} + \int \|\nabla I(\boldsymbol{x})| \, \mathrm{d}\boldsymbol{x} \tag{3.10}$$

where I' is the orthogonal projection of a current solution onto the set of admissible solutions given by the constraints $A\boldsymbol{I}_i \in D_i$. The orthogonal projection is computed using an alternating projections method with the result from Example 3.5.2

As previously mentioned, the decoding of a frame is done from most significant bit plane to least significant bit plane. In the following we will denote the corresponding constraints on the DCT coefficients of the decoded bands by DC, AC1, AC2, etc., and use data from decoding of the codec presented in Luong et al. (2013).

Figure 3.20 shows an example of how to convert the DC to an estimate of the unknown image in pixel domain. The particular structure of the DC allows for easy manipulation to create an initial estimate of the frame in question. Using only the midpoint of the given intervals, the solution with least Euclidian norm (Example 3.5.1) is a blocky nearest-neighbor-like interpolation of the image. A smoother estimate may be found by using bicubic interpolation on the same mid-points, which will add some global regularity to the estimate, and finally the quality of this solution may in general be improved by projecting the bicubic solution onto to the space of admissible solutions given by the known coefficients. This final estimate will be used as the initial guess when minimizing (3.10).

Figure 3.21 shows the results of the presented algorithm with an increasing number of decoded bands. For the algorithm we have iterated (3.10) 5000 times with $\beta = 50$, and the projection onto the set of admissible solutions is done by 20 iterations of alternating projection of all individual constraints. While the characteristics of total total variation smoothing are clearly visible, we see that already at four decoded bands (out of 16 in total), the estimates look quite decent.
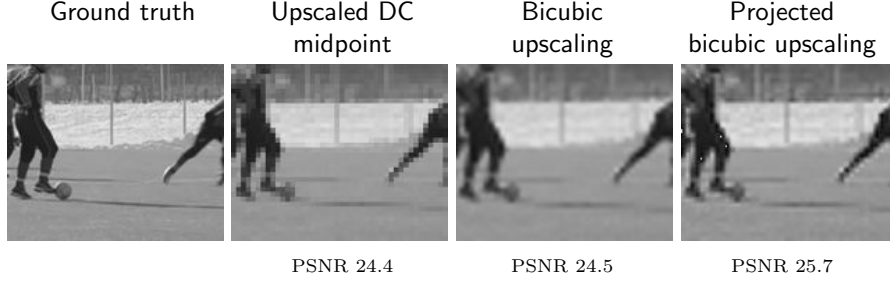
46

Ground truth	Upscaled DC midpoint	Bicubic upscaling	Projected bicubic upscaling
	PSNR 24.4	PSNR 24.5	PSNR 25.7

Figure 3.20: Frame 81 of the *Soccer* sequence and reconstruction of the frame based on the DC.

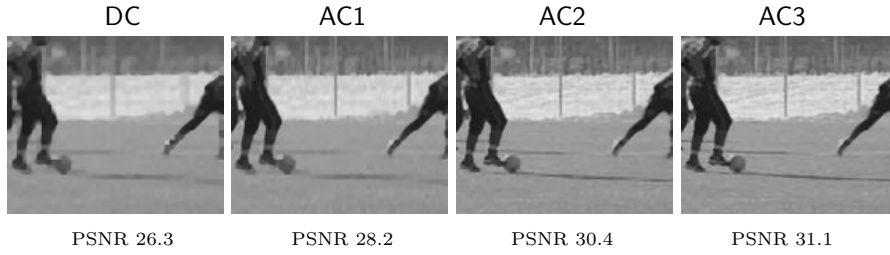| DC	AC1	AC2	AC3
PSNR 26.3	PSNR 28.2	PSNR 30.4	PSNR 31.1

Figure 3.21: Total variation upscaling of frame 81 of the *Soccer* sequence using an increasing number of decoded bands.

3.5.3 Motion reestimation and interpolation

The upsampling method described above provides a link, that allows one to use the information contained in decoded bands in transform domain, to full resolution images in pixel domain. This new information is different from the depth maps used in the interpolation process in Section 3.4, in the sense that it is just an overly smooth estimate of the frame to be interpolated. The goal is then to fill in the fine details from the outer frame onto this smooth estimate.

It turns out that this detail mapping may be done remarkably well by using a slightly modified TV-L^1 optical flow algorithm. With an estimate of the frame in question $\hat{I}_t$ produced by the total variation minimization proposed in the previous section, motion can be estimated directly between the estimate and the corresponding key frames I_{t-1}, I_{t+1}. This reduction of the temporal gap may increase the accuracy in the motion estimation, but more importantly, it eliminates the need for the assumption that motion is linear in-between key frames. The main difficulty is that the estimated frame may be a very coarse approximation of the real solution, in particular when only the DC coefficients have been decoded. To address this, we will use a specialized smoothing strategy prior to downsampling images in the image pyramid. At pyramid level ℓ with downsampling factor γ the Gaussian smoothing compared to the full resolution images has standard deviation $0.5 \cdot \gamma^{-\max(\ell, \ell')}$ where ℓ' is some given level. This means that from level ℓ' and down we will have a fixed total standard deviation compared to the full resolution images, and thus smooth out the finer details at these levels. This smoothing makes it possible to properly estimate motion to the generated estimates. Apart from this specialized downsampling

and smoothing strategy, we use Algorithm 2.1 with $i_{\max} = 15$ and otherwise standard parameters. Table 3.4 gives the levels ℓ' and λ values used for the different number of decoded bands.

Figure 3.22 shows the results of this procedure, where details from the key frames are mapped onto the estimates from Figure 3.21.

Table 3.4: Paramaters used for reestimation with varying number of decoded bands.

Highest decoded	ℓ'	λ
DC	25	15
AC1	15	15
AC2	15	40
AC3	15	60

DC	AC1	AC2	AC3

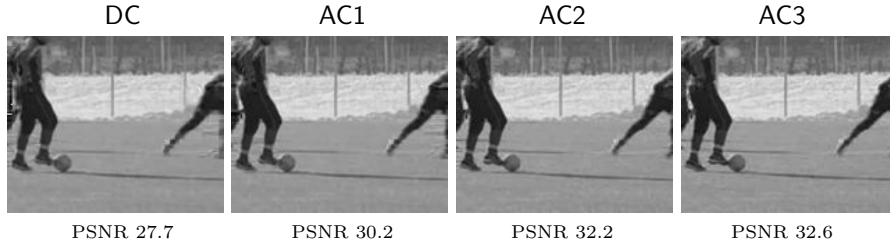| PSNR 27.7	PSNR 30.2	PSNR 32.2	PSNR 32.6

Figure 3.22: Details mapped to the estimates from Figure 3.21 using optical flow from surrounding frames.

The average PSNR values for the four test sequences can be found in Table 3.5. We see that results after reestimation are significantly better for the dynamic sequences *Foreman* and *Soccer*, while the initial estimates provide better results for the almost static *Coastguard* and *Hall* sequences. The parameters in Table 3.4 have been chosen to achieve this, since a proper multi-hypothesis decoding scheme (Huang et al. 2011) may be used to fuse the best parts of different estimates to a single superior estimate. Dynamic sequences generally present a problem in distributed video coding with respect to side information generation (Huang et al. 2011, Rakêt et al. 2012b), which means that improvements as those seen in Table 3.5 will in the end be the ones that deliver the main bitrate saving in the coding process, compared to conventional methods.

Table 3.5: Average PSNR across the 74 interpolated frames for the four test sequences.

Sequence	Initial	DC	AC1	AC2	AC3
Coastguard	32.71	27.74	27.78	33.01	34.02
Foreman	30.19	33.74	34.69	36.04	36.61
Hall	37.25	31.43	33.19	36.22	36.67
Soccer	23.75	29.96	31.49	34.23	34.93

Chapter 4

Conclusions and future directions

This thesis has described the duality based TV-L^1 optical flow method, and variations hereof. Theoretical work on generalizing the original formulation of Zach et al. (2007) has been presented, and a highly optimized algorithmic setup was described. In addition five novel applications were given. We considered registration of medical images and 2D chromatograms, as well as three examples of frame interpolation in video sequences.

It has been demonstrated that the TV-L^1 optical flow algorithm is able to produce good results on benchmark data, and that the robustness of the formulation allows it to be used successfully in a wide range of applications. This robustness is a very important feature of the algorithm, as small benchmark datasets has a tendency to pull the development of algorithms toward algorithms that work mainly on the specific examples available (Austvoll 2005). While the Middlebury benchmark has been built as a response to optical flow algorithms typically only being evaluated on (and overfitted to) the *Yosemite* and *Tree* sequences, the examples available are hardly realistic situations, and good performance on this benchmark does not guarantee good performance in other applications. Recently new benchmark data for optical flow evaluation has been presented (Geiger et al. 2012, Butler et al. 2012), and hopefully these data will help giving rise to new robust optical flow methods.

The most obvious direction of future research is to properly investigate the different extensions presented in Chapter 2, and consider how they interact. Then use this knowledge to build an algorithm with even better performance than what has been presented here.

Another question that has not been considered here is estimation of parameters. Very few successful methods for doing this exist. Zimmer et al. (2011) proposed the so-called optimal prediction principle, where multiple flows with different λ parameters were estimated, and the one with the best predictive qualities (with respect to data fit in a subsequent frame) was chosen. A similar idea was considered in Rakêt (2012) to give a locally varying field of λ values. While these methods have been demonstrated to work well, they are somewhat heuristic. A general and efficient parameter estimation framework would greatly benefit optical flow estimation, as this would remove the need to

tune parameters for specific applications.

An additional point of future investigations is the symmetric interpolation results presented in Section 3.3. The interpolation quality may of course be improved by using a symmetric data term with a more advanced optical flow method, but this is perhaps not the most exciting research direction. If the goal is to improve viewing experience, a spatial regularization of the interpolated frames could probably improve the perceived quality. Spatial regularization may be done by means of total variation (Keller et al. 2010, Werlberger et al. 2011) or by edge enhancing diffusion (Weickert 1994). The latter has been shown to have very good interpolation properties in other areas, and has been successfully used in image compression (Galić et al. 2008) as well as motion compensated deinterlacing (Ghodstinat et al. 2009). A proper generalization to frame interpolation would likely produce very good and robust results. To improve reconstruction quality one could in addition do occlusion reasoning and selectively interpolate from the non-occluded frame, or compute motion trajectories over several frames (Volz et al. 2011) and use this information for interpolation.

Finally, an important point of future research is the approach to the vector valued problem described in Section 3.2, where the coupling of channels is moved to the regularization term, instead of the data term. These uncoupled data terms are somewhat similar to the robustification of Bruhn et al. (2005), who decomposed the original coupling of brightness and gradient suggested in Brox et al. (2004), but kept as a strict requirement that the flows stayed the same (as was the case with brightness and depth in (3.7)). This decoupling was taken further to decouple HSV color channels in Zimmer et al. (2011). It will be interesting to consider how the proposed method compares to usual data term coupling, in particular if it will add any robustness to the estimation, since it allows for a much simpler and more efficient solution than the one described in Proposition 2.1.4.

Bibliography

Aaron, A., Zhang, R. & Girod, B. (2002), Wyner-Ziv coding of motion video, *in* 'Signals, Systems and Computers, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on', Vol. 1, pp. 240 –244 vol.1.

Adato, Y., Zickler, T. & Ben-Shahar, O. (2011), 'A polar representation of motion and implications for optical flow', *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on* pp. 1145–1152.

Alvarez, L., Castaño, C., García, M., Krissian, K., Mazorra, L., Salgado, A. & Sánchez, J. (2007b), Symmetric optical flow, *in* R. Díaz, F. Pichler & A. Arencibia, eds, 'Computer Aided Systems Theory–EUROCAST 2007', Vol. 4739 of *Lecture Notes in Computer Science*, Springer, pp. 676–683.

Alvarez, L., Deriche, R., Papadopoulo, T. & Sánchez, J. (2007a), 'Symmetrical dense optical flow estimation with occlusions detection', *International Journal of Computer Vision* **75**, 371–385.

Austvoll, I. (2005), A study of the Yosemite sequence used as a test sequence for estimation of optical flow, *in* H. Kalviainen, J. Parkkinen & A. Kaarna, eds, 'Image Analysis', Vol. 3540 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 659–668.

Ayvaci, A., Raptis, M. & Soatto, S. (2012), 'Sparse occlusion detection with optical flow', *International Journal of Computer Vision* **97**, 322–338.

Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J. & Szeliski, R. (2011), 'A database and evaluation methodology for optical flow', *International Journal of Computer Vision* **31**(1), 1–31.

Barron, J., Fleet, D. & Beauchemin, S. (1994), 'Performance of optical flow techniques', *International Journal of Computer Vision* **12**, 43–77.

Black, M. J. & Anandan, P. (1996), 'The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields', *Computer Vision and Image Understanding* **63**(1), 75–104.

Bresson, X. & Chan, T. (2008), 'Fast dual minimization of the vectorial total variation norm and application to color image processing', *Inverse Problems and Imaging* **2**(4), 455–484.

Brox, T., Bruhn, A., Papenberg, N. & Weickert, J. (2004), High accuracy optical flow estimation based on a theory for warping, *in* T. Pajdla & J. Matas, eds,

'Computer Vision - ECCV 2004', Vol. 3024 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 25–36.

Bruhn, A., Weickert, J. & Schnörr, C. (2005), 'Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods', *International Journal of Computer Vision* **61**, 211–231.

Butler, D., Wulff, J., Stanley, G. & Black, M. (2012), A naturalistic open source movie for optical flow evaluation, *in* A. W. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato & C. Schmid, eds, 'Computer Vision – ECCV 2012', Vol. 7577 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 611–625.

Chambolle, A. (2004), 'An algorithm for total variation minimization and applications', *Journal of Mathematical Imaging and Vision* **20**, 89–97.

Chambolle, A. & Pock, T. (2011), 'A first-order primal-dual algorithm for convex problems with applications to imaging', *Journal of Mathematical Imaging and Vision* **40**, 120–145.

Chen, K. & Lorenz, D. (2011), 'Image sequence interpolation using optimal control', *Journal of Mathematical Imaging and Vision* **41**, 222–238.

Chen, W. (2012), 'Surface velocity estimation from satellite imagery using displaced frame central difference equation', *IEEE Transactions on Geoscience and Remote Sensing,* **50**(7), 2791 –2801.

Christensen, G. E. & Johnson, H. J. (2001), 'Consistent image registration', *IEEE Transactions on Medical Imaging* **20**(7), 568–582.

Ekeland, I. & Teman, R. (1999), *Convex Analysis and Variational Problems*, SIAM.

Gai, J. & Stevenson, R. (2010), Optical flow estimation with p-harmonic regularization, *in* 'Image Processing (ICIP), 2010 17th IEEE International Conference on', pp. 1969 –1972.

Galić, I., Weickert, J., Welk, M., Bruhn, A., Belyaev, A. & Seidel, H.-P. (2008), 'Image compression with anisotropic diffusion', *Journal of Mathematical Imaging and Vision* **31**, 255–269.

Geiger, A., Lenz, P. & Urtasun, R. (2012), Are we ready for autonomous driving? The KITTI vision benchmark suite, *in* 'Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on', pp. 3354 –3361.

Ghodstinat, M., Bruhn, A. & Weickert, J. (2009), Deinterlacing with motion-compensated anisotropic diffusion, *in* D. Cremers, B. Rosenhahn, A. L. Yuille & F. R. Schmidt, eds, 'Statistical and Geometrical Approaches to Visual Motion Analysis', Springer-Verlag, Berlin, Heidelberg, pp. 91–106.

Girod, B., Aaron, A., Rane, S. & Rebollo-Monedero, D. (2005), 'Distributed video coding', *Proceedings of the IEEE* **93**(1), 71 –83.

Goldlücke, B., Strekalovskiy, E. & Cremers, D. (2012), 'The natural vectorial total variation which arises from geometric measure theory', *SIAM Journal on Imaging Sciences* **5**(2), 537–563.

Golub, G. & van Loan, C. (1989), *Matrix Computations*, The John Hopkins University Press, Baltimore, Maryland.

Herbst, E., Seitz, S. & Baker, S. (2009), Occlusion reasoning for temporal interpolation using optical flow, Technical Report UW-CSE-09-08-01, Department of Computer Science and Engineering, University of Washington.

Horn, B. K. P. & Schunck, B. G. (1981), 'Determining optical flow', *Artificial Intelligence* **17**, 185–203.

Huang, X. & Forchhammer, S. (2005), 'Cross-band noise model refinement for transform domain Wyner-Ziv video coding', *Signal Processing: Image Communication* **27**, 16–30.

Huang, X., Rakêt, L. L., Luong, H. V., Nielsen, M., Lauze, F. & Forchhammer, S. (2011), Multi-hypothesis transform domain Wyner-Ziv video coding including optical flow, *in* 'Multimedia Signal Processing (MMSP), 2011 IEEE 13th International Workshop on', pp. 1–6.

Keller, S., Lauze, F. & Nielsen, M. (2010), Temporal super resolution using variational methods, *in* M. Mrak, M. Grgic & M. Kunt, eds, 'High-Quality Visual Experience: Creation, Processing and Interactivity of High-Resolution and High-Dimensional Video Signals', Springer Berlin Heidelberg, pp. 275–296.

Kiseliov, Y. (1994), 'Algorithms of projection of a point onto an ellipsoid', *Lithuanian Mathematical Journal* **34**, 141–159.

Lucas, B. D. & Kanade, T. (1981), An iterative image registration technique with an application to stereo vision, *in* 'Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)', pp. 674–679.

Luong, H. V., Rakêt, L. L., Huang, X. & Forchhammer, S. (2012), 'Side information and noise learning for distributed video coding using optical flow and clustering', *Image Processing, IEEE Transactions on* **21**(12), 4782–4796.

Luong, H. V., Rakêt, L. L., Salmistraro, M. & Forchhammer, S. (2013), Motion and reconstruction reestimation for distributed video coding. (in preparation).

Panin, G. (2012), Mutual information for multi-modal, discontinuity-preserving image registration, *in* G. Bebis, R. Boyle, B. Parvin, D. Koracin, C. Fowlkes, S. Wang, M.-H. Choi, S. Mantler, J. Schulze, D. Acevedo, K. Mueller & M. Papka, eds, 'Advances in Visual Computing', Vol. 7432 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 70–81.

Papenberg, N., Bruhn, A., Brox, T., Didas, S. & Weickert, J. (2006), 'Highly accurate optical flow computations with theoretically justified warping', *International Journal of Computer Vision* **67**(2), 141–158.

Petersen, I. L., Tomasi, G., Sørensen, H., Boll, E. S., Hansen, H. C. B. & Christensen, J. H. (2011), 'The use of environmental metabolomics to determine glyphosate level of exposure in rapeseed (Brassica napus L.) seedlings', *Environmental Pollution* **159**(10), 3071 – 3077.

Pock, T., Urschler, M., Zach, C., Beichel, R. & Bischof, H. (2007), A duality based algorithm for TV-L^1-optical-flow image registration, *in* N. Ayache, S. Ourselin & A. Maeder, eds, 'Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007', Vol. 4792 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 511–518.

Rakêt, L. L. (2012), Local smoothness for global optical flow, *in* 'Image Processing (ICIP), 2012 19th IEEE International Conference on', pp. 1–4.

Rakêt, L. L. & Markussen, B. (2014), 'Approximate inference for spatial functional data on massively parallel processors', *Computational Statistics & Data Analysis* **72**, 227 – 240.

Rakêt, L. L. & Nielsen, M. (2012), A splitting algorithm for directional regularization and sparsification, *in* 'Pattern Recognition (ICPR), 2012 21st International Conference on', pp. 3094–3098.

Rakêt, L. L., Roholm, L., Bruhn, A. & Weickert, J. (2012a), Motion compensated frame interpolation with a symmetric optical flow constraint, *in* G. Bebis, R. Boyle, B. Parvin, D. Koracin, C. Fowlkes, S. Wang, M.-H. Choi, S. Mantler, J. Schulze, D. Acevedo, K. Mueller & M. Papka, eds, 'Advances in Visual Computing', Vol. 7431 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 447–457.

Rakêt, L. L., Roholm, L., Nielsen, M. & Lauze, F. (2011), TV-L^1 optical flow for vector valued images, *in* Y. Boykov, F. Kahl, V. Lempitsky & F. Schmidt, eds, 'Energy Minimization Methods in Computer Vision and Pattern Recognition', Vol. 6819 of *Lecture Notes in Computer Science*, Springer, pp. 329–343.

Rakêt, L. L., Søgaard, J., Salmistraro, M., Luong, H. V. & Forchhammer, S. (2012b), Exploiting the error-correcting capabilities of low density parity check codes in distributed video coding using optical flow, *in* 'Proceedings of SPIE, the International Society for Optical Engineering', Vol. 8499, SPIE – International Society for Optical Engineering.

Rudin, L. I., Osher, S. & Fatemi, E. (1992), 'Nonlinear total variation based noise removal algorithms', *Phys. D* **60**, 259–268.

Salmistraro, M., Rakêt, L. L., Zamarin, M., Ukhanova, A. & Forchhammer, S. (2013), Texture side information generation for distributed coding of video-plus-depth, *in* 'Image Processing, 2013. ICIP '13. 2013 International Conference on'.

Salmistraro, M., Zamarin, M., Rakêt, L. L. & Forchhammer, S. (2013), Distributed multi-hypothesis coding of depth maps using texture motion information and optical flow, *in* 'Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on', pp. 1685–1689.

Sommer, S., Lauze, F., Nielsen, M. & Pennec, X. (2012), 'Sparse multi-scale diffeomorphic registration: The kernel bundle framework', *Journal of Mathematical Imaging and Vision* pp. 1–17.

Steinbrücker, F., Pock, T. & Cremers, D. (2009a), Large displacement optical flow computation without warping, *in* 'Computer Vision, 2009 IEEE 12th International Conference on', pp. 1609 –1614.

Steinbrücker, F., Pock, T. & Cremers, D. (2009b), Advanced data terms for variational optic flow estimation, *in* M. A. Magnor, B. Rosenhahn & H. Theisel, eds, 'Proceedings of the Vision, Modeling, and Visualization Workshop 2009, November 16-18, 2009, Braunschweig, Germany', DNB, pp. 155–164.

Stich, T., Linz, C., Albuquerque, G. & Magnor, M. (2008), 'View and time interpolation in image space', *Computer Graphics Forum* **27**(7), 1781–1787.

Sun, D., Roth, S. & Black, M. J. (2010), Secrets of optical flow estimation and their principles, *in* 'Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on', pp. 2432 –2439.

Vese, L. A. & Osher, S. J. (2002), 'Numerical methods for p-harmonic flows and applications to image processing', *SIAM Journal on Numerical Analysis* **40**(6), 2085–2104.

Volz, S., Bruhn, A., Valgaerts, L. & Zimmer, H. (2011), Modeling temporal coherence for optical flow, *in* 'Computer Vision (ICCV), 2011 IEEE International Conference on', pp. 1116 –1123.

Wedel, A., Cremers, D., Pock, T. & Bischof, H. (2009b), Structure- and motion-adaptive regularization for high accuracy optic flow, *in* 'Computer Vision, 2009 IEEE 12th International Conference on', pp. 1663 –1668.

Wedel, A., Pock, T., Braun, J., Franke, U. & Cremers, D. (2008), Duality TV-L^1 flow with fundamental matrix prior, *in* 'Image Vision and Computing', Auckland, New Zealand.

Wedel, A., Pock, T., Zach, C., Bischof, H. & Cremers, D. (2009a), An improved algorithm for TV-L^1 optical flow, *in* D. Cremers, B. Rosenhahn, A. Yuille & F. Schmidt, eds, 'Statistical and Geometrical Approaches to Visual Motion Analysis', Vol. 5064 of *Lecture Notes in Computer Science*, Springer, pp. 23–45.

Weickert, J. (1994), Theoretical foundations of anisotropic diffusion in image processing, *in* W. G. Kropatsch, R. Klette & F. Solina, eds, 'Theoretical Foundations of Computer Vision', Vol. 11 of *Computing Supplement*, Springer, pp. 221–236.

Werlberger, M., Pock, T. & Bischof, H. (2010), Motion estimation with non-local total variation regularization, *in* 'Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on', pp. 2464 –2471.

Werlberger, M., Pock, T., Unger, M. & Bischof, H. (2011), Optical flow guided TV-L^1 video interpolation and restoration, *in* Y. Boykov, F. Kahl, V. Lempitsky & F. Schmidt, eds, 'Energy Minimization Methods in Computer Vision and Pattern Recognition', Vol. 6819 of *Lecture Notes in Computer Science*, Springer, pp. 273–286.

Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D. & Bischof, H. (2009), Anisotropic Huber-L^1 optical flow, *in* 'Proceedings of the British Machine Vision Conference (BMVC)', London, UK.

Xu, L., Jia, J. & Matsushita, Y. (2010), Motion detail preserving optical flow estimation, *in* 'Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on', pp. 1293 –1300.

Xu, L., Jia, J. & Matsushita, Y. (2012), 'Motion detail preserving optical flow estimation', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **34**(9), 1744 –1757.

Xu, L., Lu, C., Xu, Y. & Jia, J. (2011), 'Image smoothing via L_0 gradient minimization', *ACM Transactions on Graphics (SIGGRAPH Asia)* **30**(4).

Zach, C., Pock, T. & Bischof, H. (2007), A duality based approach for realtime TV-L^1 optical flow, *in* F. Hamprecht, C. Schnörr & B. Jähne, eds, 'Pattern Recognition', Vol. 4713 of *Lecture Notes in Computer Science*, Springer, pp. 214–223.

Zamarin, M. & Forchhammer, S. (2012), Lossless compression of stereo disparity maps for 3D, *in* 'Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on', pp. 617 –622.

Zimmer, H., Bruhn, A. & Weickert, J. (2011), 'Optic flow in harmony', *International Journal of Computer Vision* **93**, 368–388.

Zitnick, C., Jojic, N. & Kang, S. B. (2005), Consistent segmentation for optical flow estimation, *in* 'Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on', Vol. 2, pp. 1308 –1315 Vol. 2.

Zitnick, L., Kang, S., Uyttendaele, M., Winder, S. & Szeliski, R. (2004), 'High-quality video view interpolation using a layered representation', *ACM Transactions on Graphics* **23**(3), 600–608.