Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit

Inpainting Methods for Optical Flow

Ediba Žugor

Course of Study:	Informatik
Examiner:	Prof. Dr. -Ing. Andrés Bruhn
Supervisor:	Daniel Maurer Michael Stoll
Commenced:	2017-07-10
Completed:	2018-01-05

Abstract

Current methods for computing optical flow are based on a four-step pipeline. The goal of the first step is finding point correspondences between two consecutive images. The aim of the second step is filtering problematic or even false correspondences. The purpose of the third step—inpainting, is filling in the missing information from the neighborhood. The final step refines the obtained dense flow field using a variational approach. Up to now, there was little research that deals with the inpainting step and no work if a variational approach could improve the inpainting step. A common technique for the final step of the optical flow pipeline is minimizing an energy functional. In contrast, this thesis uses the minimization of an energy function for the inpainting step, which is also, the focus of this thesis. The inpainting energy functional consists of a similarity term and a smoothness term. For the smoothness term several possible extensions are proposed, that incorporate image information and enable an anisotropic smoothing behavior. Finally, all extensions are compared with each other and with the results from EpicFlow [RWHS15].

Acknowledgments

I would first like to thank Professor Andrés Bruhn. The door to Professor Bruhn's office was always open whenever I had a question about my thesis or writing. I have been extremely lucky to have a supervisor who cared so much about my work and give me so many valuable comments on the research.

I would also like to thank Michael Stoll for giving me brief explanations regarding the code and for his assistance and dedicated involvement in every step throughout the process. I thank Daniel Maurer for his infinite patience and always encouraging way of helping, by correcting my numerous mistakes. His suggestions were invaluable.

You should know that your support and help was worth more than I can express on paper.

I am grateful to my parents and my sister Indira, who have provided me through moral and emotional support in my life. All the support and encouragement they gave me over the years was the greatest gift anyone has ever given me.

Many Thanks!

Contents

1 Introduction **7**
- 1.1 Application . . . 7
- 1.2 Motivation . . . 7
- 1.3 The Goal of the Thesis . . . 9
- 1.4 Outline of the Thesis . . . 9

2 Background **11**
- 2.1 Problem Formulation . . . 11
  - 2.1.1 Gaussian Presmoothing . . . 12
  - 2.1.2 The Grey Value Constancy Assumption . . . 12
- 2.2 Aperture Problem . . . 13
- 2.3 Variational Methods . . . 14
  - 2.3.1 Horn and Schunck . . . 15
  - 2.3.2 Minimisation . . . 16
- 2.4 Challenges . . . 20
- 2.5 Advanced Data Terms . . . 20
  - 2.5.1 Higher Order Constancy Assumptions . . . 21
  - 2.5.2 Color Extension . . . 22
  - 2.5.3 Robust Data Terms . . . 23
  - 2.5.4 The Warping Strategy . . . 24
- 2.6 Optic Flow Pipeline . . . 26
  - 2.6.1 Matching . . . 27
  - 2.6.2 Filtering . . . 27
  - 2.6.3 Inpainting . . . 27
  - 2.6.4 Variational Refinement . . . 28

3 Advanced Smoothness Terms **29**
- 3.1 Non-quadratic Regularisers . . . 30
- 3.2 Anisotropy . . . 30
  - 3.2.1 Isotropic Smoothness Terms . . . 31
  - 3.2.2 Anisotropic Smoothness Term . . . 31
- 3.3 Higher Order Terms . . . 33
  - 3.3.1 Direct Approaches . . . 33
  - 3.3.2 Indirect Approaches . . . 34
  - 3.3.3 Combined Approaches . . . 35

4 Related Work **37**

5 Extensions **41**
5.1 Deep Matching 41
5.2 Inpainting 43
5.2.1 Inpainting Smoothness Term 45
5.3 Variational Refinement 48

6 Evaluation and Experiments **49**
6.1 Comparison of the Smoothness Term 50
6.1.1 Minimisation of the Inpainting Functional 50
6.1.2 Results 54
6.2 Comparison with EpicFlow 54
6.2.1 Minimisation 58
6.2.2 Results 58

7 Conclusion **65**

Bibliography **67**

1 Introduction

1.1 Application

In the field of object tracking, optical flow has a primary role for helping to estimate object velocity and position in the next frame. In the world of visual effects, besides object tracking, it is used in $3D$ reconstruction, motion blur etc.

A good algorithm yields important information about apparent motion of objects, like information about the velocities and the directions of objects. Apart from object tracking, computing optical flow can also be applied for industrial automation purposes, for example driver assistance systems. To increase its autonomy, a vehicle needs a way to perceive its environment and adapt its actions using sensors for that purpose.

Even though computing optical flow was originally developed for computer and robot vision application, it has found its way in physics, especially the physics of fluids. Liu and Shen [LS08] showed the quantitative connection between optical flow and fluid flow for typical flow visualizations. Another usage of optical flow is in military applications, like moving target detection in noisy infrared image sequences [CBS00].

1.2 Motivation

Calculating optical flow is an important problem in computer vision. The goal is estimating the motion field between two consecutive images (see Figure 1.1). The visual similarity between two regions is the most important clue for finding the optical flow, however, due to illumination changes, occlusions, blur etc., it is often unreliable [BTS17]. For correct estimation of dense flow fields, other limiting constraints have to be added, like local smoothness assumptions [HS81]. Incorporating a smoothness assumption with a linearised data term into a variational energy function, usually relies on having to use image pyramids for large displacements. In cases where the determined motion on a coarse scale is not close to the correct motion, this approach will fail.

Other methods include sparse descriptor matching techniques, like SIFT [Low04]. The sparse descriptor matching techniques are fast and reliable approaches, which can find a global match, but they contain only matches that are determined for points with high confidence [BTS17]. Therefore, they leave gaps in the motion field and since a motion for which no match is found cannot be considered, this does not represent an optimal approach

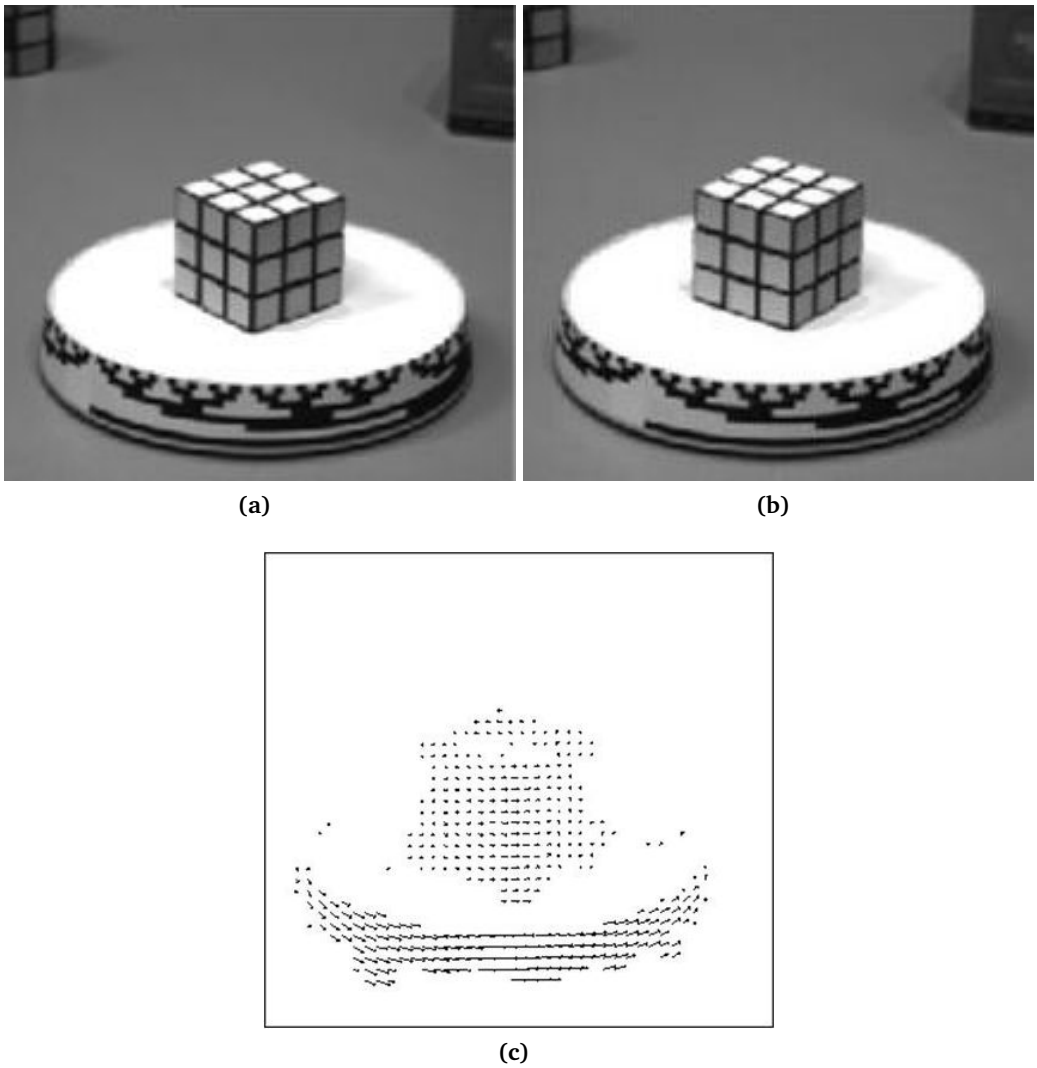


Figure 1.1: Computing the correspondence points. **(a) Top Left:** Image frame at time t. **(b) Top Right:** Image frame at time t+1. **(c) Bottom:** Optical flow: pixel motion field for frames a and b. Figure from [RN95].

by itself. However, using the sparse descriptor matching technique as the initial step of optical flow algorithms mostly leads to improving the final flow computation.

Current methods for estimating the optical flow rely on a four-step pipeline, in further text denoted as the *optic flow pipeline*. The four steps are:

- **matching** - finding point correspondences with descriptors, like SIFT,
- **outlier filtering** - removing problematic or false correspondences,
- **inpainting** - filling in the missing information from the neighbours,

- **variational refinement** - refining the inpainted dense flow field with a variational method.

For the first step of the optic flow pipeline new techniques have been proposed, like in Bailer et al. [BTS17], who obtains good results for the EpicFlow method [RWHS15], by using patch matching techniques. The second step was improved by e.g. the forward-backward check [XCS+06]. For the final step a few improvements have been proposed (see e.g. [MSB17a]).

1.3 The Goal of the Thesis

The main focus of this thesis is the third step, namely the inpainting step. Until now, there was little research in this area. An important question remains unanswered—if improving the inpainting step will improve the final optical flow. The goal of the work presented here is to develop a new inpainting method, based on a variational approach. Further, the developed inpainting method shall be improved by using more advanced smoothness terms and proposing new smoothness term, which rely on the smoothness terms presented by Maurer et al. [MSV+17]. The final results will be compared with the inpainting step of the EpicFlow [RWHS15] algorithm.

1.4 Outline of the Thesis

The application and motivation for the thesis are provided in Sections 1.1 and 1.2. The thesis goal is described in Section 1.3. The final Section 1.4 of this chapter gives a general overview of this thesis.

The remaining chapters are organized as follows:

- **Chapter 2** contains the introduction of the optical flow concept, aperture problem and how to solve a simple, global case computation of the optic flow presented by Horn and Schunck [HS81]. After showing the biggest constraints of the model of Horn and Schunck, some more advanced data terms, based on higher order constancy assumptions, color extension and more robust penalisers are described. The final section in this chapter introduces the backbone for the rest of the thesis, namely the optic flow pipeline with a short overview of every step.
- **Chapter 3** presents more advanced smoothness terms, including concepts like anisotropy and non-quadratic regularisers. Besides first other smoothness terms, also second order smoothness terms are presented, which are later used for a possible improvements.
- **Chapter 4** reviews the related work and background theories in inpainting. A particular focus is placed on the techniques that has been chosen for comparing the final results within this thesis.

- **Chapter 5** contains the main work done in this thesis. The sections are divided similar like the optic flow pipeline. Hence, first the deep matching algorithm is introduced, then new extensions of the smoothness terms from Chapter 3, but with the aim of improving the inpainting step and finally, the energy functional for the variational refinement is presented.
- **Chapter 6** gives an overview of the datasets used, the error calculation techniques, and the parameter settings. Finally, the experimental results are presented with a comparison to the EpicFlow.
- **Chapter 7** summarizes the work done in this thesis and gives some insights into the limitations and advantages of the work.

2 Background

This chapter contains the basics regarding the computations of the optical flow with some of its challenges, like the aperture problem. The most common example of optical flow estimation is presented—the method of Horn and Schunck [HS81] and an adequate minimization. Furthermore, novel advanced data terms are presented with modified constraints for a more accurate estimation of the optical flow. The last section gives a short overview of a commonly used pipeline for optical flow estimation.

Most of the concepts and examples used in the following, can be found in [Bru16] and [Bru15].

2.1 Problem Formulation

Given two consecutive image frames I^1 and I^2, interpreted as a set of pixels, where ${I^1}_{i,j}$ is the grey value of image I^1 at pixel (i, j) and ${I^2}_{i,j}$ is the grey value of image I^2 at pixel (i, j), the goal is to compute the motion field (u, v) between both frames, see Figure 2.1.

Optical flow is presented as a $2D$ velocity field and results from moving objects in the scene or camera motion. It can also be described as tracking pixels from one frame to next, which results in a series of vectors for every pixel. Assuming that certain features of a pixel do not change between frames, the grey value constancy assumption can be considered

$$ {I^1}_{i,j} - {I^2}_{i+u,j+v} = 0. \tag{2.1} $$

In case that multiple pixels have the same grey value, or no pixel with the same grey value, the solution of the equation above may be non-unique or non-existing for some pixels. The defined problem is restricted to integer displacements, making it unsuitable for real world image sequences.

Provided that I^1 and I^2 are sampled instances from a continuous sequence $I_0(x, y, t)$ with $(x, y)^\top \in \Omega$ denoting the location within the image domain $\Omega \subset \mathbb{R}^2$ and $t \in {\mathbb{R}_0}^+$ denoting the time, it is possible to switch from integer to continuous modelling.

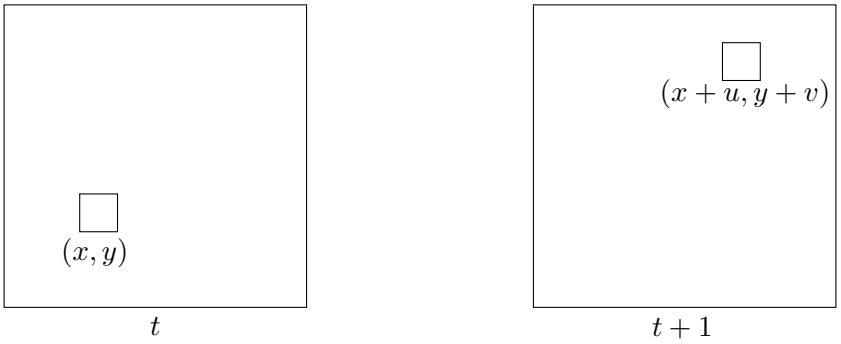


Figure 2.1: Two consecutive image frames at times t and $t + 1$. **(a) Left:** At time t the object is at position (x, y). **(b) Right:** At the next time frame, the object is at position $(x + u, y + v)$. Figure adapted from [BT05].

2.1.1 Gaussian Presmoothing

In order to reduce the influence of noise and outliers, while preserving the mean value, each image of the image sequence I_0 can be convolved with a $2D$ Gaussian K_σ of mean $\mu = 0$ and standard deviation σ:

$$I(x, y, t) = K_\sigma * I_0(x, y, t), \tag{2.2}$$

where $K_\sigma(x, y) := \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$.

After the convolution with a Gaussian, the image sequence becomes infinitely many times differentiable, i.e. $I \in C^\infty$, which is important for computing image derivatives for some differential methods.

2.1.2 The Grey Value Constancy Assumption

In case that $u(x, y, t)$ and $v(x, y, t)$ denote the displacements in $x-$ and $y-$direction, respectively, the continuous grey value constancy assumption reads

$$I(x, y, t) - I(x + u, y + v, t + 1) = 0, \tag{2.3}$$

where $I(x, y, t)$ and $I(x, y, t + 1)$ are two consecutive frames for fixed t.

If the displacement is very small, the left hand side of the equation can be linearised via first order Taylor expansion around the point $(x, y, t)^\top$:

$$\begin{aligned} 0 &= I(x, y, t) - I(x + u, y + v, t + 1) \\ &= I_x(x, y, t)u + I_y(x, y, t)v + I_t(x, y, t), \end{aligned} \tag{2.4}$$

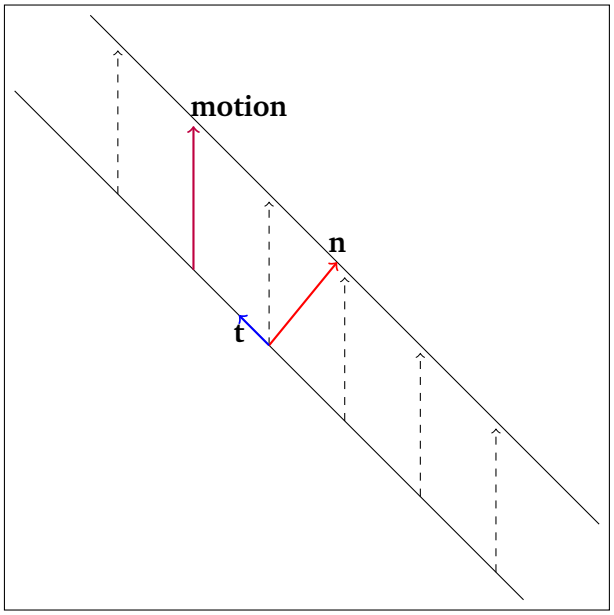


Figure 2.2: Aperture problem. Vector **n** (red) represents the flow normal to the image edge. Vector **t** (blue) represents the flow tangential to the image edge. Only component **n** , which is in the direction of the spatial gradient, can be measured. Figure adapted from [Fis17].

where subscripts denote partial derivatives.

This yields the **brightness constancy constraint equation (BCCE)**

$$I_x u + I_y v + I_t = 0. \tag{2.5}$$

2.2 Aperture Problem

One of the reasons for losing much of the structural information of the original scene during the imaging process is having a $3D$ scene and trying to project it to a $2D$ image, therefore estimating the correct projected motion is ill-posed and requires additional constraints.

Given a single equation with two unknowns u and v, the Equation (2.5) is ill-posed and thus cannot be computed uniquely. Only the flow component parallel to the spatial gradient $\nabla I = (I_x, I_y)^\top$ can be computed via BCCE (as illustrated in Figure 2.2):

$$0 = I_x u + I_y v + I_t = \begin{pmatrix} u \\ v \end{pmatrix}^\top \nabla I + I_t. \tag{2.6}$$

The BCCE will not be violated by adding arbitrary flow components orthogonal to ∇I. This is called aperture problem.

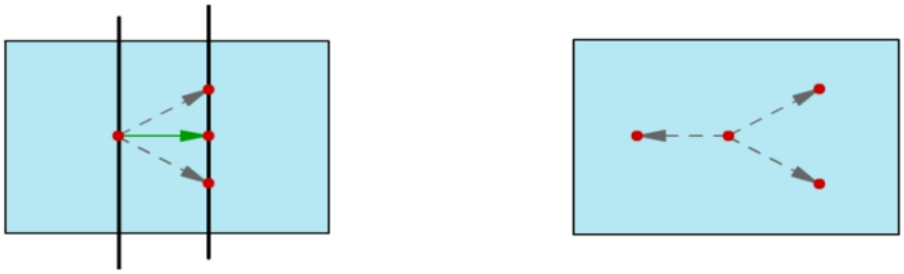


Figure 2.3: Aperture problem. **(a) Left:** Aperture problem when $|\nabla I| \neq 0$. Single image pixel corresponds to a line. **(b) Right:** Nothing can be said about the flow when $|\nabla I| = 0$. Figure from [Bru06].

The flow vector $(u, v)^\top$ can be expressed via the basis vectors $\mathbf{n} = \frac{\nabla I}{|\nabla I|}$ and $\mathbf{t} = \frac{\nabla I}{|\nabla I|}^\perp$, which represent the flow normal and tangential to the image edge respectively:

$$\begin{aligned} \begin{pmatrix} u \\ v \end{pmatrix} &= \left[\begin{pmatrix} u \\ v \end{pmatrix}^\top \frac{\nabla I}{|\nabla I|} \right] \frac{\nabla I}{|\nabla I|} + \left[\begin{pmatrix} u \\ v \end{pmatrix}^\top \frac{\nabla I}{|\nabla I|}^\perp \right] \frac{\nabla I}{|\nabla I|}^\perp \\ &= \begin{pmatrix} u_n \\ v_n \end{pmatrix} + \begin{pmatrix} u_t \\ v_t \end{pmatrix}. \end{aligned} \tag{2.7}$$

From Equation (2.6) follows $-I_t = \begin{pmatrix} u \\ v \end{pmatrix}^\top \nabla I$, thus the normal flow becomes:

$$\begin{pmatrix} u_n \\ v_n \end{pmatrix} = \left[\begin{pmatrix} u \\ v \end{pmatrix}^\top \frac{\nabla I}{|\nabla I|} \right] = -\frac{I_t}{|\nabla I|} \frac{\nabla I}{|\nabla I|} = \frac{-I}{I_x^2 + I_y^2} \begin{pmatrix} I_x I_t \\ I_y I_t \end{pmatrix}. \tag{2.8}$$

Only the normal flow can be computed from the BCCE without additional constraints. In case where $|\nabla I| = 0$ no image information is available, therefore, nothing can be said about the flow and not even the normal flow can be calculated (see Figure 2.3).

Finding an additional constraint that yields a second equation in the same unknowns can help to overcome the aperture problem. It is most commonly assumed that the flow field is smooth. This approach was developed by Horn and Schunck [HS81] and will be discussed in detail in next section.

2.3 Variational Methods

The most widely used techniques for optic flow computation are differential methods. Differential methods employ additional constraints to the estimation of the optical flow.

Depending on whether the constraints are applied to neighboring pixels, or to all pixels, differential methods are divided into local methods and global methods. Local methods are more robust under noise, but do not yield very dense flow fields, whereas global methods produce dense flow fields. Even a combination of local and global methods has been proposed by Bruhn et al. [BWS05].

Two basic approaches for estimating the optical flow are developed by Horn and Schunck [HS81] and Lucas and Kanade [LK+81]. Lucas and Kanade proposed a local method under the assumption that small regions of pixels have the same flow, meaning that the flow is constant in a local neighbourhood. The optical flow is computed based on the least squares criterion for all pixels in the neighbourhood. Compared to point-wise methods it is less sensitive to noise and often resolves the ambiguity problem of the BCCE (Equation (2.5)). The model of Horn and Schunck is based on global optimization and will be briefly discussed in this section. In this thesis the main focus will be on global methods.

2.3.1 Horn and Schunck

To overcome the aperture problem an additional smoothness constraint is needed. The additional constraint and the BCCE, are combined to form an energy functional

$$E(u,v) = \int_{\Omega} E_D(u,v) + \lambda E_S(u,v) dxdy. \tag{2.9}$$

$E_D(u,v)$ represents the data term and it models constancy assumptions on image features and E_S is the smoothness term, which penalises variations in the flow field. The parameter λ, also called the smoothness weight, represents a balance between the two terms. A bigger λ means more influence of the smoothness term and more information will be provided from the neighbouring pixels.

Under the assumption that the flow field is globally smooth, the method of Horn and Schunck overcomes the aperture problem by penalising large spatial flow gradients ∇u and ∇v:

$$E_S(u,v) = |\nabla u|^2 + |\nabla v|^2. \tag{2.10}$$

Combining the BCCE from Equation as the data term E_D and the additional smoothness constraint, the following energy functional is obtained

$$E(u,v) = \int_{\Omega} \underbrace{(I_x u + I_y v + I_t)^2}_{\text{data term}} + \lambda \underbrace{(|\nabla u|^2 + |\nabla v|^2)}_{\text{smoothness term}} dxdy. \tag{2.11}$$

Besides penalising deviations from smoothness of the flow field, the smoothness constraint has an additional advantage, namely the so-called filling-in-effect. If $|\nabla I| \approx 0$, no data information is available and the only contribution for the estimation will be from the smoothness constraint. The solution for u and v will be propagated from the neighbours.

2.3.2 Minimisation

Horn and Schunck propose to compute the optical flow as the minimizer of the energy functional (2.11). Deriving the minimizer for the proposed energy functional means deriving minimising functions u and v, that fits best to all model assumptions, which can be contradictory. The energy functional is convex and thus has a unique solution, which can be found by any globally convergent algorithm [Bru06].

The energy functional (2.11) is of the following form

$$E(u,v) = \int_\Omega F(x,y,u,v,u_x,u_y,v_x,v_y)dxdy. \tag{2.12}$$

Minimising a function, like the one above, can be done in three steps:

1. Euler-Lagrange equations
2. Discretisation
3. Iterative solution

Every step will be presented in detail. The minimisation is an adaptation from [Bru16].

1. **Euler-Lagrange equations**

   The necessary conditions to minimize E are the Euler-Lagrange equations

   $$\begin{aligned} F_u - \frac{\partial}{\partial x}F_{u_x} - \frac{\partial}{\partial y}F_{u_y} = 0, \\ F_v - \frac{\partial}{\partial x}F_{v_x} - \frac{\partial}{\partial y}F_{v_y} = 0, \end{aligned} \tag{2.13}$$

   with Neumann boundary conditions

   $$\mathbf{n}^\top \begin{bmatrix} F_{u_x} \\ F_{u_y} \end{bmatrix} = 0 \quad \text{and} \quad \mathbf{n}^\top \begin{bmatrix} F_{v_x} \\ F_{v_y} \end{bmatrix} = 0, \tag{2.14}$$

   where $\mathbf{n}$ is the unit normal vector.

   The integrand has the following form

   $$F = (I_x u + I_y v + I_t)^2 + \lambda(u_x^2 + u_y^2) + \lambda(v_x^2 + v_y^2), \tag{2.15}$$

   with partial derivatives

$$\begin{aligned} F_u &= 2I_x(I_x u + I_y v + I_t), \\ F_{u_x} &= 2\lambda u_x, \\ F_{u_y} &= 2\lambda u_y, \\ F_v &= 2I_y(I_x u + I_y v + I_t), \\ F_{v_x} &= 2\lambda v_x, \\ F_{v_y} &= 2\lambda v_y. \end{aligned} \tag{2.16}$$

After inserting all the above terms in the Euler-Lagrange equations, the following equations are derived

$$\begin{aligned} 0 &= 2I_x^2 u + 2I_x I_y v + 2I_x I_t - 2\lambda u_{xx} - 2\lambda u_{yy}, \\ 0 &= 2I_y^2 v + 2I_x I_y u + 2I_y I_t - 2\lambda v_{xx} - 2\lambda v_{yy}. \end{aligned} \tag{2.17}$$

If taken into consideration that the Laplacian operator has the following form $\Delta u = u_{xx} + u_{yy}$, the above equations are equal to those below

$$\begin{aligned} \lambda \Delta u &= (I_x u + I_y v + I_t) I_x, \\ \lambda \Delta v &= (I_x u + I_y v + I_t) I_y, \end{aligned} \tag{2.18}$$

with Neumann boundary conditions: $\mathbf{n}^\top \nabla u = 0$ and $\mathbf{n}^\top \nabla v = 0$.

2. **Discretisation**

For solving the Euler-Lagrange equations numerically, discretisation is necessary. The image is sampled on a rectangular grid at regular intervals (h_x is the spacing in x direction, h_y is the spacing in y direction and h_t is the distance between both frames and in general h_t is assumed to be 1) with indices i, j representing the intersection of the i^{th} row and j^{th} column. To fully discretize the Euler-Lagrange equation three steps are required:

a) Discretisation of the flow functions u and v:

$$u_{i,j} = u(i \cdot h_x, j \cdot h_y) \qquad v_{i,j} = v(i \cdot h_x, j \cdot h_y), \tag{2.19}$$

for $i = 1, ..., N$ and $j = 1, ..., M$.

b) The entries I_x^2, I_y^2, I_t^2, I_xI_y, I_xI_t and I_yI_t should also be discretised, which requires the discretisation of I_x, I_y and I_t and also I

$$I_{i,j,t} = I(i \cdot h_x, j \cdot h_y, t), \tag{2.20}$$

for $i = 1, ..., N$, $j = 1, ..., M$ and $t = 1, 2$.

I_x and I_y can be discretised via average central differences, whereas I_t via forward differences

$$\begin{aligned}
[I_x]_{i,j} &= \frac{1}{2}\left(\frac{I_{i+1,j,t+1} - I_{i-1,j,t+1}}{2h_x} + \frac{I_{i+1,j,t} - I_{i-1,j,t}}{2h_x}\right), \\
[I_y]_{i,j} &= \frac{1}{2}\left(\frac{I_{i,j+1,t+1} - I_{i,j-1,t+1}}{2h_y} + \frac{I_{i,j+1,t} - I_{i,j-1,t}}{2h_y}\right), \\
[I_t]_{i,j} &= \frac{I_{i,j,t+1} - I_{i,j,t}}{h_t}.
\end{aligned} \tag{2.21}$$

c) Discretisation of $\Delta u = u_{xx} + u_{yy}$ and $\Delta v = v_{xx} + v_{yy}$ can be based on nested central differences with half the grid sizes $\frac{1}{2}h_x$ and $\frac{1}{2}h_y$, respectively.

$\Delta u = u_{xx} + u_{yy}$ reads

$$\begin{aligned}
\Delta u &= (u_x)_x + (u_y)_y \\
&\approx \frac{(u_x)_{i+\frac{1}{2},j} - (u_x)_{i-\frac{1}{2},j}}{2(\frac{1}{2}h_x)} + \frac{(u_y)_{i,j+\frac{1}{2}} - (u_y)_{i,j-\frac{1}{2}}}{2(\frac{1}{2}h_y)} \\
&\approx \frac{\frac{u_{i+1,j}-u_{i,j}}{2(\frac{1}{2}h_x)} - \frac{u_{i,j}-u_{i-1,j}}{2(\frac{1}{2}h_x)}}{2(\frac{1}{2}h_x)} + \frac{\frac{u_{i,j+1}-u_{i,j}}{2(\frac{1}{2}h_y)} - \frac{u_{i,j}-u_{i,j-1}}{2(\frac{1}{2}h_y)}}{2(\frac{1}{2}h_y)} \\
&= \frac{u_{i+1,j} - u_{i,j}}{h_x^2} - \frac{u_{i,j} - u_{i-1,j}}{h_x^2} + \frac{u_{i,j+1} - u_{i,j}}{h_y^2} - \frac{u_{i,j} - u_{i,j-1}}{h_y^2}.
\end{aligned} \tag{2.22}$$

Following the same procedure for Δv

$$\begin{aligned}
\Delta v &= (v_x)_x + (v_y)_y \\
&\approx \frac{(v_x)_{i+\frac{1}{2},j} - (v_x)_{i-\frac{1}{2},j}}{2(\frac{1}{2}h_x)} + \frac{(v_y)_{i,j+\frac{1}{2}} - (v_y)_{i,j-\frac{1}{2}}}{2(\frac{1}{2}h_y)} \\
&\approx \frac{\frac{v_{i+1,j}-v_{i,j}}{2(\frac{1}{2}h_x)} - \frac{v_{i,j}-v_{i-1,j}}{2(\frac{1}{2}h_x)}}{2(\frac{1}{2}h_x)} + \frac{\frac{v_{i,j+1}-v_{i,j}}{2(\frac{1}{2}h_y)} - \frac{v_{i,j}-v_{i,j-1}}{2(\frac{1}{2}h_y)}}{2(\frac{1}{2}h_y)} \\
&= \frac{v_{i+1,j} - v_{i,j}}{h_x^2} - \frac{v_{i,j} - v_{i-1,j}}{h_x^2} + \frac{v_{i,j+1} - v_{i,j}}{h_y^2} - \frac{v_{i,j} - v_{i,j-1}}{h_y^2}.
\end{aligned} \tag{2.23}$$

Combining all the above derivations yields the following two equations

$$\begin{aligned} 0 &= [I_x]_{i,j}^2 u_{i,j} + [I_x]_{i,j}[I_y]_{i,j} v_{i,j} + [I_x]_{i,j}[I_t]_{i,j} - \lambda \sum_{l \in (x,y)} \sum_{\widetilde{(i,j)} \in N_l(i,j)} \frac{u_{\widetilde{i,j}} - u_{i,j}}{h_l^2}, \\ 0 &= [I_x]_{i,j}[I_y]_{i,j} u_{i,j} + [I_y]_{i,j}^2 v_{i,j} + [I_y]_{i,j}[I_t]_{i,j} - \lambda \sum_{l \in (x,y)} \sum_{\widetilde{(i,j)} \in N_l(i,j)} \frac{v_{\widetilde{i,j}} - v_{i,j}}{h_l^2}, \end{aligned} \tag{2.24}$$

for $i = 1, ..., N$ and $j = 1, ..., M$, where $N_l(i,j)$ denotes the set of neighbours of pixel i, j in direction of axis l.

3. Iterative solution

To compute the optical flow, it is necessary to solve a system of two equations for every point in an image, or $2(N \cdot M)$ unknowns in total. Solving those equations with a standard method, like the Gauß-Jordan elimination would be very costly. Horn and Schunck used iterative solvers, like the Gauß-Seidel method (see Algorithm 2.1).

Algorithm 2.1 Gauß Seidel

procedure GAUSS-SEIDEL ITERATIVE METHOD
  $N_l(i,j)^- \leftarrow$ set of neighbours of pixel (i,j) in direction of axis l
  that have already been computed in the current iteration.
  $N_l(i,j)^+ \leftarrow$ set of neighbours of pixel (i,j) in direction of axis l
  that still have to be computed in the current iteration.
  Initialize starting values for all pixels (i,j) of the optical flow$(u_{i,j}, v_{i,j})$.
  $k \leftarrow$ solver iteration index of the current processed image.
  loop:

$$u_{i,j}^{k+1} = \frac{(-[I_x]_{i,j}[I_t]_{i,j} - ([I_x]_{i,j}[I_y]_{i,j} v_{i,j}^k - \lambda \sum_{l \in x,y} \sum_{\widetilde{(i,j)} \in N_l^-(i,j)} \frac{u_{\widetilde{i,j}}^{k+1}}{h_l^2} - \lambda \sum_{l \in x,y} \sum_{\widetilde{(i,j)} \in N_l^+(i,j)} \frac{u_{\widetilde{i,j}}^{k}}{h_l^2}))}{[I_x]_{i,j}^2 + \lambda \sum_{l \in x,y} \sum_{\widetilde{(i,j)} \in N_l(i,j)} \frac{1}{h_l^2}}$$

$$v_{i,j}^{k+1} = \frac{(-[I_y]_{i,j}[I_t]_{i,j} - ([I_x]_{i,j}[I_y]_{i,j} u_{i,j}^{k+1} - \lambda \sum_{l \in x,y} \sum_{\widetilde{(i,j)} \in N_l^-(i,j)} \frac{v_{\widetilde{i,j}}^{k+1}}{h_l^2} - \lambda \sum_{l \in x,y} \sum_{\widetilde{(i,j)} \in N_l^+(i,j)} \frac{v_{\widetilde{i,j}}^{k}}{h_l^2}))}{[I_y]_{i,j}^2 + \lambda \sum_{l \in x,y} \sum_{\widetilde{(i,j)} \in N_l(i,j)} \frac{1}{h_l^2}}$$

  goto *loop* to process all images in sequence.
end procedure

One shortcoming of this numerical solver is that it depends on the initial guess. If the initial guess is bad, the solver may require thousands of iterations per time-step.

2.4 Challenges

As mentioned in [LZS13] the difficulties in estimating the optic flow, for the method of Horn and Schunck, arise because:

1. Computing dense flow fields is expensive.
2. Occluded regions and motion discontinuities are hard to model—fast methods usually assume that all pixels move in the same direction, this leads to bad estimations at occlusion boundaries.
3. Displacements which are larger than the object structure lead to more difficult matching.
4. Optical flow does not represent the true motion field, because illumination changes are also reflected in calculating the optical flow.

Other challenges related to the application field are accuracy and robustness. For real-time applications, computational speed and real-time capability are also crucial issues [WC11].

2.5 Advanced Data Terms

The data term models constancy assumptions on image features. The most common used data term is the one based on the linearised grey value constancy assumption. Unfortunately, it is constrained only in one direction (see Section 2.2), making the aperture problem always present. This section will mainly contains more advanced data terms.

To make more advanced data terms, four improvements should be considered:

1. **Multiple constraints** - not only one constraint, like the grey value constancy assumption can be considered, but multiple constraints, including higher order constancy assumptions.
2. **Color extension** - extension of all constraints to the RGB model.
3. **Robust data terms** - for a more robust version of the data term, other non-quadratic penalisers should be used.
4. **Refraining from Taylor linearisation** - for big displacements, the Taylor linearisation will not be valid anymore.

2.5.1 Higher Order Constancy Assumptions

Previously, only the grey value constancy assumption was used as a data term

$$E_D(u,v) = (I_x u + I_y v + I_t)^2. \tag{2.25}$$

In comparison to the grey value constancy assumption, other possible constancy assumptions have certain advantages.

The main drawback of the above data term is its susceptibility to slight changes in brightness, which often appear in natural scenes [BBPW04], thus it can not cope with additive illumination changes. A solution to this problem was proposed by Uras et al. [UGVT88] and it is called the gradient constancy assumption

$$\begin{aligned} I_x(x,y,t) - I_x(x+u, y+v, t+1) = 0, \\ I_y(x,y,t) - I_y(x+u, y+v, t+1) = 0. \end{aligned} \tag{2.26}$$

After linearising the above equations, the following two equations will be derived

$$\begin{aligned} I_{xx} u + I_{xy} v + I_{xt} = 0, \\ I_{yx} u + I_{yy} v + I_{yt} = 0. \end{aligned} \tag{2.27}$$

Having these two equations, it is enough to provide a unique solution for u and v, thus the aperture problem is not always present.

The gradient constancy assumption, which is invariant to the changes in brightness, allows some small variations in the grey value and penalises illumination changes less severely.

The constancy assumptions can be of higher order, like the second order spatial image Hessian $\mathcal{H}I = \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix}$, proposed by Papenberg et al. [PBB+06]:

$$\begin{aligned} I_{xx}(x,y,t) - I_{xx}(x+u, y+v, t+1) = 0, \\ I_{xy}(x,y,t) - I_{xy}(x+u, y+v, t+1) = 0, \\ I_{xy}(x,y,t) - I_{xy}(x+u, y+v, t+1) = 0, \\ I_{yy}(x,y,t) - I_{yy}(x+u, y+v, t+1) = 0. \end{aligned} \tag{2.28}$$

The linearisation reads

$$\begin{aligned} I_{xxx}u + I_{xxy}v + I_{xxt} &= 0, \\ I_{xyx}u + I_{xyy}v + I_{xyt} &= 0, \\ I_{xyx}u + I_{xyy}v + I_{xyt} &= 0, \\ I_{yyx}u + I_{yyy}v + I_{yyt} &= 0. \end{aligned} \tag{2.29}$$

For the second order constancy assumption four equations are derived, but the second and third equations are identical. Having three unique equations and only two unknowns, makes the aperture problem not always present, like in the case of gradient constancy assumption.

Gradient and the Hessian are well suited for translational and divergent motions, however they contain directional information, making it not optimal in cases where objects rotate. To discard the directional information, the constancy of the magnitude of the spatial image gradient was proposed [PBB+06]

$$|\nabla_2 I(x,y,t)|_\epsilon - |\nabla_2 I(x+u, y+v, t+1)|_\epsilon = 0, \tag{2.30}$$

where $|x|_\epsilon = \sqrt{x^2 + \epsilon}$ is the regularised magnitude, required to guarantee differentiability. The following linearisation holds

$$[|\nabla_2 I|_\epsilon]_x u + [|\nabla_2 I|_\epsilon]_y v + [|\nabla_2 I|_\epsilon]_t = 0. \tag{2.31}$$

Another example for assuring rotation invariance is the constancy of the trace of the Hessian, which comes down to the Laplacian

$$|\Delta I(x,y,t) - \Delta I(x+u, y+v, t+1)| = 0, \tag{2.32}$$

or the constancy of the determinant of the Hessian

$$det(\boldsymbol{\mathcal{H}} I(x,y,t)) - det(\boldsymbol{\mathcal{H}} I(x+u, y+v, t+1)) = 0. \tag{2.33}$$

2.5.2 Color Extension

Previous assumptions have only been made on grey value images. Additional information can be obtained from color images. Regarding the data term, extensions can be made using the RGB colour sequence $\mathbf{I}^i = (I^1, I^2, I^3)^\top$, where $(I^1, I^2, I^3)^\top$ are all three $(R, G, B)^\top$ color channels

$$\begin{aligned} I^1(x,y,t) - I^1(x+u, y+v, t+1) &= 0, \\ I^2(x,y,t) - I^2(x+u, y+v, t+1) &= 0, \\ I^3(x,y,t) - I^3(x+u, y+v, t+1) &= 0. \end{aligned} \tag{2.34}$$

Coupling the above equations yields

$$E_{D-color}(u,v) = \sum_{c=1}^{3}(\mathbf{I}^c(x,y,t) - \mathbf{I}^c(x+u, y+v, t+1))^2 dxdy. \tag{2.35}$$

After linearising the data term, the above equation reads

$$E_{D-color}(u,v) = \sum_{c=1}^{3}(\mathbf{I}_x^c u + \mathbf{I}_y^c v + \mathbf{I}_t^c)^2 dxdy. \tag{2.36}$$

For a unique solution all three constraints should be fulfilled jointly by one displacement field.

2.5.3 Robust Data Terms

For a good optical flow algorithm, it is necessary to overcome many challenges. One of them is robustness to outliers, which include occlusions. Outliers caused by noise or occlusions had a big influence in the energy function of Horn and Schunck, mostly due to the quadratic penaliser. For reducing influence of the outliers non-quadratic penalisers can be used. Changing the energy function 2.11 to

$$E(u,v) = \int_\Omega \Psi\big((I_x u + I_y v + I_t)^2\big) + \lambda(|\nabla u|^2 + |\nabla v|^2) dxdy, \tag{2.37}$$

where $\Psi(s^2)$ is the sub-quadratic penaliser (described below), can reduce the influence of outliers.

Sub-Quadratic Penaliser

To reduce the influence of outliers caused by noise or occlusions, the robust sub-quadratic penaliser is used

$$\Psi_{SQ}(s^2) = \sqrt{s^2 + \epsilon^2}, \tag{2.38}$$

where ϵ is a small positive constant. The function is illustrated in Figure 2.4.

For the energy functional (2.37) with $\Psi_{SQ}(s^2)$ being the sub-quadratic penaliser, computing the Euler-Lagrange equation will give the following two equations (c.f. [Bru16])

$$\begin{aligned} 0 &= \Psi'_{SQ}\big((I_x u + I_y v + I_t)^2\big)\big({I_x}^2 u + I_x I_y v + I_x I_t\big) - \lambda \Delta u, \\ 0 &= \Psi'_{SQ}\big((I_x u + I_y v + I_t)^2\big)\big(I_x I_y u + {I_y}^2 v + I_y I_t\big) - \lambda \Delta v, \end{aligned} \tag{2.39}$$

with reflecting Neumann boundary conditions $n^\top \nabla u = 0$ and $n^\top \nabla v = 0$.

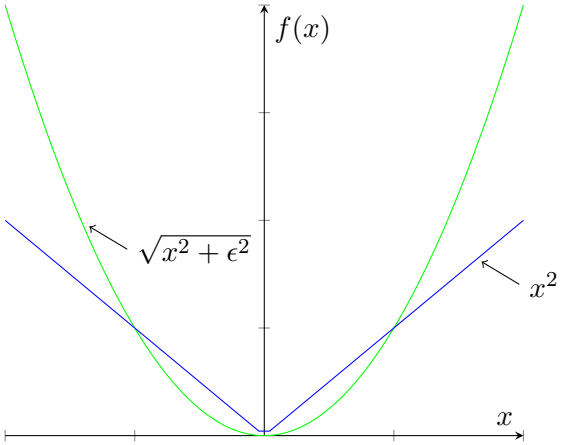


Figure 2.4: In the case of data terms with linearised constancy assumptions, $\Psi_{SQ}(s^2)$ should be positive, increasing, sub-quadratic and strictly convex. Figure adapted from [Bru16].

Due to the factor $\Psi'_{SQ}((I_x u + I_y v + I_t)^2)$, with

$$\Psi'_{SQ}(s^2) = \frac{1}{2\sqrt{s^2+\epsilon^2}}, \tag{2.40}$$

these equations are nonlinear in u and v.

2.5.4 The Warping Strategy

In Section 2.1 the brightness constancy assumption was linearised under the assumption that the displacement is very small. For large displacements, this constraint might be violated.

Without the linearisation, the Horn and Schunck energy functional is given by

$$E(u,v) = \int_\Omega \underbrace{(I(x+u,y+v,t+1) - I(x,y,t))^2}_{\text{data term}} + \lambda \underbrace{(|\nabla u|^2 + |\nabla v|^2)}_{\text{smoothness term}} dxdy. \tag{2.41}$$

The next step for minimizing this energy functional is computing Euler-Lagrange equations, which yields

$$\begin{aligned} I_x(x+u,y+v,t+1)(I(x+u,y+v,t+1) - I(x,y,t)) - \lambda\nabla u = 0, \\ I_y(x+u,y+v,t+1)(I(x+u,y+v,t+1) - I(x,y,t)) - \lambda\nabla v = 0, \end{aligned} \tag{2.42}$$

with $\mathbf{n}^\top \nabla u = 0$ and $\mathbf{n}^\top \nabla v = 0$.

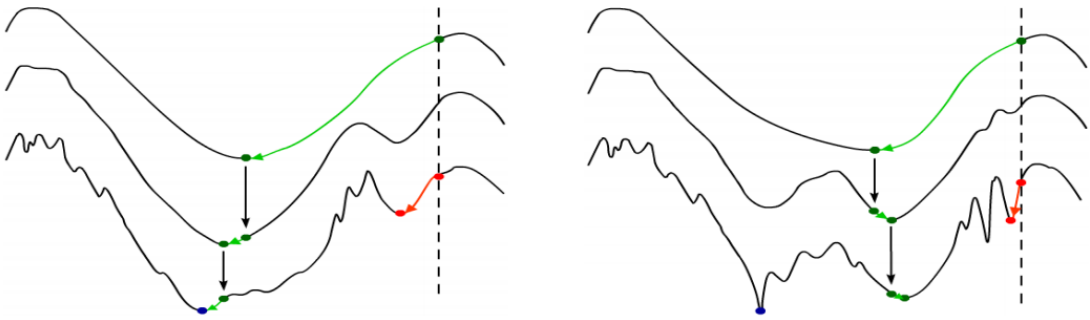


Figure 2.5: Non-convex functions with multiple local minimas. **(a) Left:** Initialisation leads to finding the global minima. **(b) Right:** Initialisation leads to finding a good enough local minima. Figure from [Bru06].

The equations are now implicit in u and v due to the use of the constancy assumption without linearisation. A direct discretisation is thus not possible. Another challenge is the non-convexity. Non-convex functions can have multiple local minimas (see Figure 2.5), which all satisfy the Euler-Lagrange equations. A minimisation concept has to be developed for finding the global minimum, or a good enough local minimum [Bru06].

To deal with both challenges an incremental coarse-to-fine fixed point iteration is introduced with the following steps:

1. **Fixed point iteration**

   For faster convergence and better stability in [Bru06] an approach is used that is semi-implicit in the data term related contributions (expressions from k and $k+1$) and fully implicit in the smoothness term related contributions (expressions only from $k+1$), where k denotes the iteration number.

   The iteration step $k+1$ is given by

$$\begin{aligned} I_x(x+u^k, y+v^k, t+1)(I(x+u^{k+1}, y+v^{k+1}, t+1) - I(x,y,t)) - \lambda \Delta u^{k+1} = 0, \\ I_y(x+u^k, y+v^k, t+1)(I(x+u^{k+1}, y+v^{k+1}, t+1) - I(x,y,t)) - \lambda \Delta v^{k+1} = 0. \end{aligned} \tag{2.43}$$

2. **Incremental computation**

   In this step the unknown flow field u^{k+1} is split into two parts:

   - already known displacement from old time step k: $(u^k, v^k)^\top$,
   - the unknown displacement increment from new time step $k+1$: $(du^k, dv^k)^\top$, hence

$$\begin{aligned} u^{k+1} &= u^k + du^k, \\ v^{k+1} &= v^k + dv^k. \end{aligned} \tag{2.44}$$

Combining the equations above and linearising the data term related contribution via first order Taylor expansion, with respect to du^k and dv^k, the following equation is derived

$$\begin{aligned} I(x + u^{k+1}, y + v^{k+1}, t + 1) = I(x + u^k + du^k, y + v^k + dv^k, t + 1) \approx \\ I(x + u^k, y + v^k, t + 1) + I_x(x + u^k, y + v^k, t + 1)du^k + I_y(x + u^k, y + v^k, t + 1)dv^k. \end{aligned} \tag{2.45}$$

This linearisation has been postponed for correct handling of large displacements [Bru06]. The constancy assumption (2.3) is now approximated as a series of linearised ones.

Inserting the Equation (2.45) in the Equation (2.43) yields

$$\begin{aligned} 0 &= I_x(x + u^k, y + v^k, t + 1)(I_x(x + u^k, y + v^k, t + 1)du^k + I_y(x + u^k, y + v^k, t + 1)dv^k \\ &\quad + \underbrace{I(x + u^k, y + v^k, t + 1) - I(x, y, t))}_{\approx I_t} - \lambda\Delta(u^k + du^k), \\ 0 &= I_y(x + u^k, y + v^k, t + 1)(I_x(x + u^k, y + v^k, t + 1)du^k + I_y(x + u^k, y + v^k, t + 1)dv^k \\ &\quad + \underbrace{I(x + u^k, y + v^k, t + 1) - I(x, y, t))}_{\approx I_t} - \lambda\Delta(v^k + dv^k). \end{aligned} \tag{2.46}$$

3. **Coarse-to-fine strategy**

   The final step is embedding the fixed point iteration into a coarse-to-fine strategy to avoid local minima. Large local minima can be avoided by starting from a coarse level and then successively refining the resolution. The coarse level can be derived by convolving I with a Gaussian with a large standard deviation and then decreasing the standard deviation to derive finer resolutions. At coarse grids, the small local minimas will vanish, improving the initialisation, which is mostly responsible for finding the global minimum or a good enough local minimum, as illustrated in Figure 2.5.

2.6 Optic Flow Pipeline

The previously described warping strategy solved problems on a coarser grid and then used those solutions as an initialisation for finer grids. At coarser scales details would disappear, including small objects [MSB17a]. Consequently large displacements still represent an open problem in the optical flow estimation, especially for small objects that move fast.

Another approach is starting with a good initialisation and using the variational approach to refine this initialisation. For this purpose, current methods use the optic flow pipeline, consisting of four steps:

$$matching \rightarrow outlier filtering \rightarrow inpainting \rightarrow variational\ refinement$$

Figure 2.6: Optic Flow Pipeline

The aim of this pipeline is to produce better results with a better initialisation.

2.6.1 Matching

Given two images I_1 and I_2 of the same scene, the matching problem is defined as finding correspondences in I_2 for N points located in I_1. In order to find correspondences between two images, measurable characteristics are required. These measurable characteristics, including corners, edges and blobs, are called image features [Alh11].

Image matching is made of two steps, extraction of local descriptors and matching them. The local descriptors are usually histograms of image measurements like in [Low04]. After extraction of local descriptors, the matching step is done based on a similarity measure, comparing each pair of features. Provided that a good matching operator is found, it should be robust to large displacements and motion discontinuities. The matches should be rather dense to obtain a good initialisation [MSB17a].

2.6.2 Filtering

Sparse features can be computed robustly and can capture long-range motions, but due to outliers and uneven covering of the images, generic interpolators do not work well [WB15]. After the matching step it is often required to preprocess the matches or filter them. The most significant reason for filtering is the removal of outliers, which can deteriorate the estimation.

2.6.3 Inpainting

After filtering, very few matches will remain. However for the initialization of the final step, a dense flow field is needed. For this goal, the remaining matches should be interpolated. The third step and the main topic of this thesis is inpainting. Inpainting focuses on the sparse-to-dense interpolation, like in EpicFlow [RWHS15], where the fill-in is done based on the nearest neighbours. The main challenge remains finding the optimal sparse-to-dense algorithm, which will provide a good initialisation for the next step.

2.6.4 Variational Refinement

The variational refinement is the final step in the optical flow pipeline. Standard variational energy minimization is computed on the inpainted flow field to obtain the final flow estimation. The inpainted flow is used as an initialisation of an energy functional.

3 Advanced Smoothness Terms

According to Wedel and Cremers [WC11] the three main purposes of the applied smoothness assumptions are:

- The fill-in effect, or propagating information from the neighboring pixels into regions with low texture.
- Respect image discontinuities, i.e. preserve edges.
- Robustness with respect to outliers.

The smoothness term penalises fluctuations in the flow field, but can not remove illumination artifacts. The simplest quadratic smoothness term, used in Chapter 2 reads

$$E_s(u, v) = |\nabla u|^2 + |\nabla v|^2. \tag{3.1}$$

Because the given regulariser enforces the same smoothness in every direction, it is called the homogeneous (isotropic) smoothness term. The regulariser does not respect any flow discontinuities, thus different image objects, which move in different directions, or with different magnitudes, could not be tracked and semantically important edges may get dislocated [Bru06] . To avoid the stated limitations more advanced smoothness terms are presented in this chapter.

Those more advanced smoothness terms consider the following three concepts:

1. **Non-quadratic regularisers** - Instead of a quadratic penaliser in the smoothness term, similar like for the data term, a different penaliser $\Psi(s^2)$ can be used. The smoothness term looks like following

$$E_{s-robust}(u, v) = \Psi(|\nabla u|^2 + |\nabla v|^2). \tag{3.2}$$

2. **Anisotropy** - Integrating directional information in the smoothness term.
3. **Higher order terms** - In contrast to first order smoothness terms, second order smoothness terms can capture the estimation of piecewise affine flows.

3.1 Non-quadratic Regularisers

Apart from the sub-quadratic regulariser from Section 2.7 with the main purpose of permitting piecewise smooth results, other non-quadratic regularisers are also used with other properties:

- The edge-enhancing **Perona-Malik (Lorentzian)** regulariser [PM90]:

$$\Psi_{PM}(s^2) = \epsilon^2 \log\left(1 + \frac{s^2}{\epsilon^2}\right). \tag{3.3}$$

- The edge preserving **Charbonnier** regulariser [CBAB94]:

$$\Psi_{CH}(s^2) = 2\epsilon^2\sqrt{1 + \frac{s^2}{\epsilon^2}}, \tag{3.4}$$

where ϵ is a small positive constant.

3.2 Anisotropy

After computing the Euler-Lagrange equations for the method of Horn and Schunck the following equations are derived, as in Chapter 2:

$$\begin{aligned} \lambda\Delta u &= (I_x u + I_y v + I_t)I_x, \\ \lambda\Delta v &= (I_x u + I_y v + I_t)I_y, \end{aligned} \tag{3.5}$$

where the left hand side can be rewritten as

$$\begin{aligned} \lambda(u_{xx} + u_{yy}) &= \lambda\text{div}\Bigg(\underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}_{\text{D}} \underbrace{\begin{pmatrix} u_x \\ u_y \end{pmatrix}}_{\nabla u}\Bigg), \\ \lambda(v_{xx} + v_{yy}) &= \lambda\text{div}\Bigg(\underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}_{\text{D}} \underbrace{\begin{pmatrix} v_x \\ v_y \end{pmatrix}}_{\nabla v}\Bigg), \end{aligned} \tag{3.6}$$

with $D = I$ being the so-called diffusion tensor.

For a more general case, the diffusion tensor can also be defined as

$$D = \begin{pmatrix} d_1 & d_{12} \\ d_{12} & d_2 \end{pmatrix}. \tag{3.7}$$

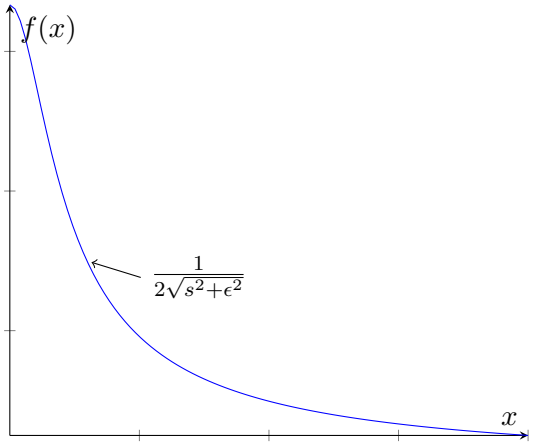


Figure 3.1: The weighting function $g(s^2)$ should be positive and decreasing. Figure adapted from [Bru16].

3.2.1 Isotropic Smoothness Terms

In case, where $d_{12} = 0$ and $d_1 = d_2$, the smoothness term is called isotropic. One example is the *flow-driven isotropic smoothness term*, proposed by Schnörr [Sch94]:

$$\Psi_{SQ}(|\nabla u|^2 + |\nabla v|^2), \tag{3.8}$$

where $D = \Psi'_{SQ}(|\nabla u|^2 + |\nabla v|^2)I$.

Another example for isotropic smoothness terms is the *isotropic image-driven smoothness term* [ÁELS99] :

$$g(|\nabla I|^2)(|\nabla u|^2 + |\nabla v|^2) \tag{3.9}$$

where $D = g(|\nabla I|^2)I$.

In this case the spatial image gradient $|\nabla I|^2$ serves as a fuzzy edge detector. If the gradient is large, the diffusion process should be inhibited [Bru06]. However, this means that at noisy pixels, the diffusion will be very small. Since the use of directional information is not integrated in the diffusion tensor and both eigenvalues are scaled with the same factor, the diffusion will be reduced also orthogonal to the gradient.

This is possible using a positive and decreasing weighting function $g(s^2)$, like $g(s^2) = \frac{1}{2\sqrt{s^2+\epsilon^2}}$ and $\epsilon > 0$, as illustrated in Figure 3.1.

3.2.2 Anisotropic Smoothness Term

Anisotropic image-driven regularisers take into account directional information from image structures. This information can be obtained by considering the structure tensor J_ρ by means of an eigenvalue decomposition, where J_ρ reads

$$J_\rho = K_\rho * (\nabla I \nabla I^\top) = \begin{pmatrix} K_\rho * I_x^2 & K_\rho * I_x I_y \\ K_\rho * I_x I_y & K_\rho * I_y^2 \end{pmatrix}. \tag{3.10}$$

$\nabla I = (I_x, I_y)^\top$ is the spatial gradient and K_ρ is the Gaussian kernel with standard deviation ρ, with $*$ being the convolution operator. The directions $\mathbf{r_1}$ and $\mathbf{r_2}$ are calculated as the normalised eigenvectors of the structure tensor J_ρ

$$J_\rho = (\mathbf{r}_1, \mathbf{r}_2) \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} \mathbf{r}_1^\top \\ \mathbf{r}_2^\top \end{pmatrix}. \tag{3.11}$$

The eigenvectors $\mathbf{r}_1$ and $\mathbf{r}_2$ represent directions with highest and lowest contrast respectively, whereas λ_1 and λ_2 are the corresponding eigenvalues, representing the contrast [Bru06]. $\mathbf{r}_1$ and $\mathbf{r}_2$ form an orthonormal system, where the vector $\mathbf{r}_1$ points across image structures and $\mathbf{r}_2$ along them. This allows to apply different kind of smoothings along and across image structures [HSW15].

An example of the anisotropic first order smoothness term is the *constraint adaptive regulariser* from Zimmer et al. [ZBW11]:

$$E_{S-con-adapt} = \Psi_{S_1}\Big(\left(\mathbf{r}_1^\top \nabla u\right)^2 + \left(\mathbf{r}_1^\top \nabla v\right)^2\Big) + \Psi_{S_2}\Big(\left(\mathbf{r}_2^\top \nabla u\right)^2 + \left(\mathbf{r}_2^\top \nabla v\right)^2\Big), \tag{3.12}$$

where $\Psi_{S_1}(s^2) = \Psi_{PM}(s^2)$ and $\Psi_{S_2}(s^2) = s^2$.

The diffusion tensor reads:

$$D_{con-adapt} = (\mathbf{r}_1, \mathbf{r}_2) \begin{pmatrix} \Psi_{S_1}\text{'} & 0 \\ 0 & \Psi_{S_2}\text{'} \end{pmatrix} \begin{pmatrix} \mathbf{r}_1^\top \\ \mathbf{r}_2^\top \end{pmatrix}, \tag{3.13}$$

where $\Psi_{S_1}\text{'} = \Psi_{S_1}\text{'}\Big(\left(\mathbf{r}_1^\top \nabla u\right)^2 + \left(\mathbf{r}_1^\top \nabla v\right)^2\Big)$ and $\Psi_{S_2}\text{'} = \Psi_{S_2}\text{'}\Big(\left(\mathbf{r}_2^\top \nabla u\right)^2 + \left(\mathbf{r}_2^\top \nabla v\right)^2\Big)$.

In case of the Zimmer et al. regulariser, the function Ψ_{S_1} performs a more robust penalisation in the $\mathbf{r}_1$-direction, meaning that it will be less smoothing in the direction across the image edges for the purpose of preserving image edges. However, in the $\mathbf{r}_2$-direction there is a quadratic penalisation with the function $\Psi_{S_2}(s^2) = s^2$. Consequently, $\Psi_{S_2}\text{'}(s^2) = 1$ and the second eigenvalue of the diffusion tensor is $\Psi_{S_2}\text{'}\Big(\left(\mathbf{r}_2^\top \nabla u\right)^2 + \left(\mathbf{r}_2^\top \nabla v\right)^2\Big) = 1$.

3.3 Higher Order Terms

All previously shown smoothness terms were first order smoothness terms and therefore cannot capture the estimation of piecewise affine flows. Second order smoothness terms penalise non-affine flow fields.

Second order strategies are divided into three classes:

- Direct approaches
- Indirect approaches (coupling models)
- Combined approaches

For every of those three classes of smoothness terms, one isotropic and two anisotropic variants are presented and represent an adaptation from [MSV+17].

3.3.1 Direct Approaches

In the method of Horn and Schunck the first-order smoothness term directly penalises the magnitude of the gradient (first order derivative). In contrast, direct second-order smoothness term directly penalise second order derivatives of the flow.

Isotropic:

One possible way to design the smoothness term is by directly penalising the Hessian $\boldsymbol{\mathcal{H}}$ [DSV+14]

$$E_{s-iso}(\boldsymbol{w}) = \Psi\Big(||\boldsymbol{\mathcal{H}}u||_F^2 + ||\boldsymbol{\mathcal{H}}v||_F^2\Big), \tag{3.14}$$

where the flow field is $\boldsymbol{w} = (u, v)^\top$, $||\boldsymbol{\mathcal{H}} \cdot ||_F$ is the Frobenius norm of the Hessian and Ψ a penaliser function that allows to preserve discontinuities in the flow field.

Using the structure of the image to regularise the flow can be beneficial, therefore using anisotropic variants for different kinds of smoothing along and across image structures will also be presented here. $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ are the eigenvectors of the structure tensor. Both directional components are penalised differently to achieve an anisotropic behaviour. Different penalisation functions Ψ_1 and Ψ_2 can be applied along and across image structures. Depending on if the directions $\boldsymbol{r}_{k,(k=1,2)}$ are penalised jointly or separately, two cases can be distinguished:

Single anisotropic case:

$$E_{s-aniso-single}(\boldsymbol{w}) = \sum_{l=1}^{2} \Psi_l \left(\sum_{k=1}^{2} \Big(\boldsymbol{r}_k^\top \boldsymbol{\mathcal{H}} u \boldsymbol{r}_l\Big)^2 + \Big(\boldsymbol{r}_k^\top \boldsymbol{\mathcal{H}} v \boldsymbol{r}_l\Big)^2 \right). \tag{3.15}$$

Double anisotropic case:

$$E_{s-aniso-double}(\boldsymbol{w}) = \sum_{l=1}^{2}\sum_{k=1}^{2}\Psi_{l,k}\Big(\Big(\boldsymbol{r}_k^\top \boldsymbol{\mathcal{H}} u \boldsymbol{r}_l\Big)^2 + \Big(\boldsymbol{r}_k^\top \boldsymbol{\mathcal{H}} v \boldsymbol{r}_l\Big)^2\Big), \tag{3.16}$$

where $\Psi_{l,k,(l=1,2;k=1,2)}$ are penalisation functions that allow to preserve discontinuities in the flow field.

Apart from the Hessian, the Laplacian can also be used, like the example from Chan et al. [CMM00]. The direct second order approach penalises kinks. However, with the lack of penalising first order derivatives, there is no possibility of modelling jumps.

3.3.2 Indirect Approaches

Another approach, which uses auxiliary functions, is the indirect approach. It approximates first order derivatives and then enforces smoothness assumptions on the auxiliary functions. Consequently, it is possible to model discontinuities in both first and second order derivatives.

Isotropic:

$$E_{s-iso}(\boldsymbol{w}) = \inf_{\boldsymbol{a},\boldsymbol{b}}\Big\{\Psi\Big(|\nabla u - \boldsymbol{a}|^2 + |\nabla v - \boldsymbol{b}|^2\Big) + \beta\Psi\Big(||\boldsymbol{\mathcal{J}a}||_F^2 + ||\boldsymbol{\mathcal{J}b}||_F^2\Big)\Big\}. \tag{3.17}$$

The vector fields $\mathbf{a} = (a_1, a_2)^\top$ and $\mathbf{b} = (b_1, b_2)^\top$ are the auxiliary functions that approximate the gradients ∇u and ∇v, respectively, whereas $\boldsymbol{\mathcal{J}a}$ and $\boldsymbol{\mathcal{J}b}$ are the Jacobians of these vector fields. λ is used as the weighting parameter.

The first term couples ∇u with $\mathbf{a}$ and ∇v with $\mathbf{b}$ and is called the coupling term. However, the aim of the second term is the smoothness of $\mathbf{a}$ and $\mathbf{b}$, hence it is referred to as the smoothness term. This approach does not require the explicit estimation of the second order derivatives [HSW15]. Apart from the isotropic case, the two anisotropic variant are the following:

Single anisotropic case:

$$\begin{aligned} E_{s-aniso-single}(\boldsymbol{w}) = \inf_{\boldsymbol{a},\boldsymbol{b}}\Bigg\{&\sum_{l=1}^{2}\Psi_l\Big(\Big(\boldsymbol{r}_l^\top(\nabla u - \boldsymbol{a})\Big)^2 + \Big(\boldsymbol{r}_l^\top(\nabla v - \boldsymbol{b})\Big)^2\Big) + \\ &\beta\sum_{l=1}^{2}\Psi_l\Big(\sum_{k=1}^{2}\big(\boldsymbol{r}_k^\top\boldsymbol{\mathcal{J}a}\boldsymbol{r}_l\big)^2 + \big(\boldsymbol{r}_k^\top\boldsymbol{\mathcal{J}b}\boldsymbol{r}_l\big)^2\Big)\Bigg\}. \end{aligned} \tag{3.18}$$

Similar to the previous case, the single anisotropic case can be extended:

Double anisotropic case:

$$E_{s-aniso-double}(\boldsymbol{w}) = \inf_{\boldsymbol{a},\boldsymbol{b}} \Bigg\{ \sum_{l=1}^{2} \Psi_l \Big(\Big(\boldsymbol{r}_l^\top (\nabla u - \boldsymbol{a}) \Big)^2 + \Big(\boldsymbol{r}_l^\top (\nabla v - \boldsymbol{b}) \Big)^2 \Big) + \beta \sum_{l=1}^{2} \sum_{k=1}^{2} \Psi_{l,k} \Big(\big(\boldsymbol{r}_k^T \boldsymbol{\mathcal{J}} \boldsymbol{a} \boldsymbol{r}_l \big)^2 + \big(\boldsymbol{r}_k^\top \boldsymbol{\mathcal{J}} \boldsymbol{b} \boldsymbol{r}_l \big)^2 \Big) \Bigg\}, \tag{3.19}$$

where $\boldsymbol{r}_k^\top \boldsymbol{\mathcal{J}} \boldsymbol{a} \boldsymbol{r}_l$ represents an approximation of $\boldsymbol{r}_k^\top \boldsymbol{\mathcal{H}} u \boldsymbol{r}_l$ and $\boldsymbol{r}_k^\top \boldsymbol{\mathcal{J}} \boldsymbol{b} \boldsymbol{r}_l$ represents an approximation of $\boldsymbol{r}_k^\top \boldsymbol{\mathcal{H}} v \boldsymbol{r}_l$.

3.3.3 Combined Approaches

Another way to include both, first and second order derivatives, in the smoothness term is the combined approach. As opposed to the indirect approach, the estimation of the second order derivatives, is done explicitly, by penalising both first and second order derivatives in one function. The trade off between both the first and second order term is done via a weight λ with flow fields $\boldsymbol{w}_1 = (u_1, v_1)^\top$ and $\boldsymbol{w}_2 = (u_2, v_2)^\top$. The isotropic variant reads:

Isotropic:

$$E_{s-iso}(\boldsymbol{w}) = \inf_{\boldsymbol{w_1}+\boldsymbol{w_2}=\boldsymbol{w}} \Bigg\{ \Psi \Big(|\nabla u_1|^2 + |\nabla v_1|^2 \Big) + \lambda \Psi \Big(||\boldsymbol{\mathcal{H}} u_2||_F^2 + ||\boldsymbol{\mathcal{H}} v_2||_F^2 \Big) \Bigg\}. \tag{3.20}$$

As in the previous cases the single anisotropic and double anisotropic cases are:

Single anisotropic case:

$$E_{s-aniso-single}(\boldsymbol{w}) = \inf_{\boldsymbol{w_1}+\boldsymbol{w_2}=\boldsymbol{w}} \Bigg\{ \sum_{l=1}^{2} \Psi_l \Big(\Big(\boldsymbol{r}_l^\top \nabla u_1 \Big)^2 + \Big(\boldsymbol{r}_l^\top \nabla v_1 \Big)^2 \Big) + \lambda \sum_{l=1}^{2} \Psi_l \Bigg(\sum_{k=1}^{2} \Big(\boldsymbol{r}_k^\top \boldsymbol{\mathcal{H}} u_2 \boldsymbol{r}_l \Big)^2 + \Big(\boldsymbol{r}_k^\top \boldsymbol{\mathcal{H}} v_2 \boldsymbol{r}_l \Big)^2 \Bigg) \Bigg\}. \tag{3.21}$$

Double anisotropic case:

$$E_{s-aniso-double}(\boldsymbol{w}) = \inf_{\boldsymbol{w_1}+\boldsymbol{w_2}=\boldsymbol{w}} \Bigg\{ \sum_{l=1}^{2} \Psi_l \Big(\Big(\boldsymbol{r}_l^\top \nabla u_1 \Big)^2 + \Big(\boldsymbol{r}_l^\top \nabla v_1 \Big)^2 \Big) + \lambda \sum_{l=1}^{2} \sum_{k=1}^{2} \Psi_{l,k} \Big(\Big(\boldsymbol{r}_k^\top \boldsymbol{\mathcal{H}} u_2 \boldsymbol{r}_l \Big)^2 + \Big(\boldsymbol{r}_k^\top \boldsymbol{\mathcal{H}} v_2 \boldsymbol{r}_l \Big)^2 \Big) \Bigg\}. \tag{3.22}$$

4 Related Work

Dealing with long-range motion is still an unsolved task in computer vision. During the coarse-to-fine estimation of the optical flow, most approaches get stuck in the local minima due to bad initialization and consequently, may fail to estimate large displacements. Furthermore, coarse-to-fine algorithms tend to remove small objects at coarser scales.

Brox et al. [BBM09] proposed an approach on how to handle large displacements by extending the global energy functional with a similarity term, hereby integrating the information of the region correspondence. The main idea is to create a hierarchy of regions and a boundary map $g(x)$ of the image. The correspondences between the regions are computed by nearest neighbor matching and are integrated in the enegy functional:

$$\begin{aligned} E(\mathbf{w}(\mathbf{x})) = & \int \underbrace{\Psi_{SQ}\Big(|I_2(\mathbf{x}+\mathbf{w}(\mathbf{x})) - I_1(\mathbf{x})|^2\Big) + \gamma\Psi_{SQ}\Big(|\nabla I_2(\mathbf{x}+\mathbf{w}(\mathbf{x})) - \nabla I_1(\mathbf{x})|^2\Big)}_{\text{data term}} d\mathbf{x} \\ & + \beta \int \underbrace{\sum_{j=1}^{5} \rho_j(\mathbf{x})\Psi_{SQ}\Big((u(\mathbf{x}) - u_j(\mathbf{x}))^2 + (v(\mathbf{x}) - v_j(\mathbf{x}))^2\Big)}_{\text{similarity term}} d\mathbf{x} \\ & + \lambda \int \underbrace{\Psi_{SQ}\Big(|\nabla u(\mathbf{x})|^2 + |\nabla v(\mathbf{x})|^2 + g(\mathbf{x})|^2\Big)}_{\text{smoothness term}} d\mathbf{x}. \end{aligned} \tag{4.1}$$

I_1 and I_2 are the two input images, $\mathbf{w} = (u, v)$ is the sought optical flow field, and $\mathbf{x} = (x, y)$ is a point in the image. $(u_j, v_j)(\mathbf{x})$ is the motion vector derived by region matching the j^{th} nearest neighbor (the indexing goes to 5), at position $\mathbf{x}$. ρ_j is an indicator function. If $\rho_j = 0$, then there is no correspondence at this position. Otherwise, ρ_j is equal to a predefined distance based confidence. λ, β and γ are parameters, with predefined values of 100, 25 and 5 respectively.

The boundary map $g(\mathbf{x})$ is integrated in the smoothness term to avoid smoothing across strong region boundaries.

Another similar approach is the one from Weinzaepfel et al. [WRHS13], on how to handle large displacements and avoid error propagation in multi-scale schemes. The matches are again integrated in the similarity term E_M, which is a part of the energy functional

$$E(\mathbf{w}) = \int_{\Omega} E_D + \lambda E_S + \beta E_M dxdy. \tag{4.2}$$

The similarity term relies on the deep matching algorithm, briefly described in the next section. λ and β are parameters, also called smoothness weight and similarity weight, respectively.

Including a matching term in the variational approach allows handling of large displacements with the strengths of a variational approach. The main drawback of this approach is that local descriptors are only reliable at sample points.

Current approaches are based on the four-step optic flow pipeline. Besides the matching, filtering and variational refinement step, this pipeline contains the inpainting step, which uses the matches for filling in the missing information from the neighbours and hence, obtaining a dense flow. This step improves bad initialisation and makes approaches less sensitive to non-reliable or even bad sample points.

In order to improve the bad initialization Revaud et al. [RWHS15] presented a novel approach for estimating optical flow, focusing on large displacements with significant occlusions. The approach is called *Edge-Preserving Interpolation of Correspondences for Optical Flow (EpicFlow)*.

The EpicFlow model uses matches, computed with a matching algorithm, as the input and then filters non-reliable points. The last two steps are:

- **Edge-preserving interpolation from a sparse set of matches.** After removing the outliers, the next step of the optical flow pipeline is the sparse-to-dense interpolation. The interpolation is done by filling in the missing information from the neighbourhood. Hence, the closest neighbours should be found. Motion boundaries often tend to appear at image edges. Therefore, the sparse-to-dense interpolation is based on an edge-aware geodesic distance for the computation of the closest neighbours. Compared to the Euclidean distance, the edge-aware distance offers a natural way to handle motion discontinuities. The cost map of the edge-aware geodesic distance is based on the *Structure Edge Detector (SED)*.
- **Variational refinement, using the output of the interpolation step.** The final step of the optical flow pipeline for EpicFlow is rather simple. The data term E_{D-epic} combines the normalised gradient constancy assumption (2.27) with the BCCE (2.5), briefly described in [MSB17a]. The smoothness term is a first order flow-driven regulariser, with image-based weighing

$$E_{S-epic}(\mathbf{w}) = g(|\nabla I|) \cdot \Psi_{SQ}\Big(|\nabla u|^2 + |\nabla v|^2\Big), \tag{4.3}$$

  where $\mathbf{w} = (u, v)$ is the sought optical flow and the additional weighting function $g(s^2)$ is defined as follows

$$g(s^2) = e^{-\frac{s^2}{\epsilon_g}}. \tag{4.4}$$

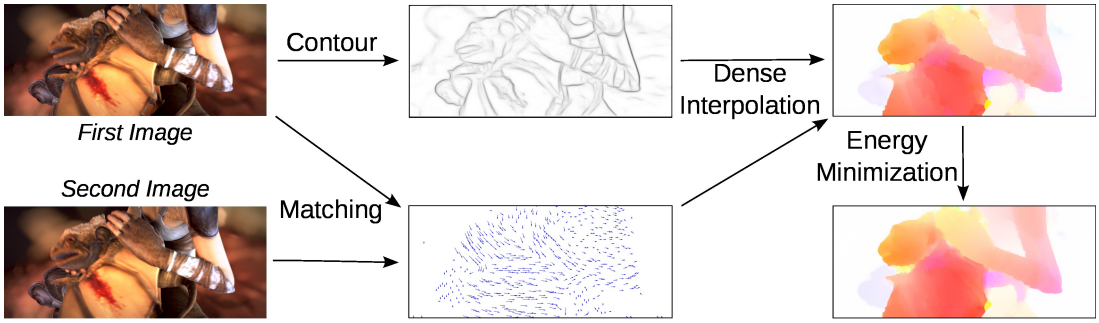


Figure 4.1: Deep matching is used to compute the sparse set of matches, SED [DZ13] is used to compute edges of the first image. The output of SED and deep matching is used as input for computing the dense correspondence field, which is again used as the initialization for computing the variational energy minimization. Figure from [RWHS15].

ϵ_g is a positive fixed constant. In the later text this additional weighting function is referred to as *geodesic* penaliser and is used for the integration of the image gradient information in the proposed smoothness terms.

Figure 4.1 illustrates an overview of EpicFLow.

The main drawback of the EpicFlow algorithm is the lack of robustness with respect to input matching noise. The robustness of EpicFlow relies on an intuitive treshold to remove some noisy matches. Hu et al. [HLS17] proposed a new technique to overcome this weakness, called *Robust Interpolation Method of Correspondences (RicFlow)*.

Under the assumption that most scenes of interest consist of regions with a constant motion pattern, the input image I is segmented in K dis-joint superpixels s_k, where the segmentation S reads

$$S = \left\{ s_k | \bigcup_{k=1}^{K} s_k = I \text{ and } \forall k \neq m, s_k \bigcap s_m = \varnothing \right\}. \tag{4.5}$$

For each superpixel s_k one superpixel flow is generated. For the superpixels without any matches, the superpixel flow is set according to the nearest valid superpixel, like illustrated in Figure 4.2. Each of the superpixels s_k meets a constant affine model A_k, due to the constant motion patter. To escape the vulnerability of the input matching noise, the model for each of the superpixels is estimated via a RANSAC-like method from its support neighbors based on a graph constructed on superpixels. The refinement step is similar like for EpicFlow.

Another way to compute the sparse-to-dense interpolation is the one presented by Wulff and Black [WB15], using a learned set of full-frame basis flow fields. To learn the principal components of natural flow fields, four Hollywood movies are used. Flow fields are then represented as a weighted sum of the basis flow fields $\mathbf{b}_n$, $n = 1...N$, with weights w_n:

Figure 4.2: Two examples of support neighbours (n = 100). The support neighbours are represented by color with blue meaning they are closer and red meaning they are farther away from the white superpixel seed. The distance is calculated via a geodesic approximation, as in EpicFlow. Figure from [HLS17].

$$\mathbf{w} \approx \sum_{n=1}^{N} w_n \mathbf{b}_n, \tag{4.6}$$

where $\mathbf{w} = (u, v)^\top$ is the flow field. For the basis flow fields $\mathbf{b}_n$, separate basis vectors are assumed for the horizontal and vertical flow components. The horizontal motion is spanned by $\{\mathbf{b}_n\}_{n=1,\dots,\frac{N}{2}}$, and the vertical by $\{\mathbf{b}_n\}_{n=\frac{N}{2}+1,\dots,N}$.

An interpolant is learned from training optical flow fields using a robust principal component analysis (PCA) to deal with noise in the training flow. Since PCA-Flow does not contain high-frequency spatial information, flow at motion boundaries will be over-smoothed. To escape over-smoothing at motion boundaries Wulff and Black proposed a layered flow model where each layer is a PCA-Flow field estimated from a subset of the sparse matches. The final dense flow field is computed combining the flow fields from each layer using Markov Random Field (MRF).

5 Extensions

The backbone for building this chapter and presenting the work done in this thesis, was the optic flow pipeline. As previously described, the first step is matching, hence, the first section contains the matching algorithm, termed deep matching, and the description of its main features. In the second section the inpainting energy functional is presented, which will be used for the evaluation part. For the energy functional more advanced smoothness terms are proposed, as an upgrade of some of the previously presented smoothness terms in the Chapter 3. The last section presents some insides in the final refinement of the optical flow.

5.1 Deep Matching

For finding correspondence points, the SIFT [Low04] descriptor matching approach can be used. SIFT can be computed as a set of orientation histograms on $4 \cdot 4$ pixel neighbours. Each descriptor contains an array of 4 histograms around the pixel and each histogram contains 8 bins like shown in Figure 5.1. The SIFT feature vector has $4 \cdot 4 \cdot 8 = 128$ elements. Having used only simple, local, rectangular patches, it is constrained to finding only rigid matches.

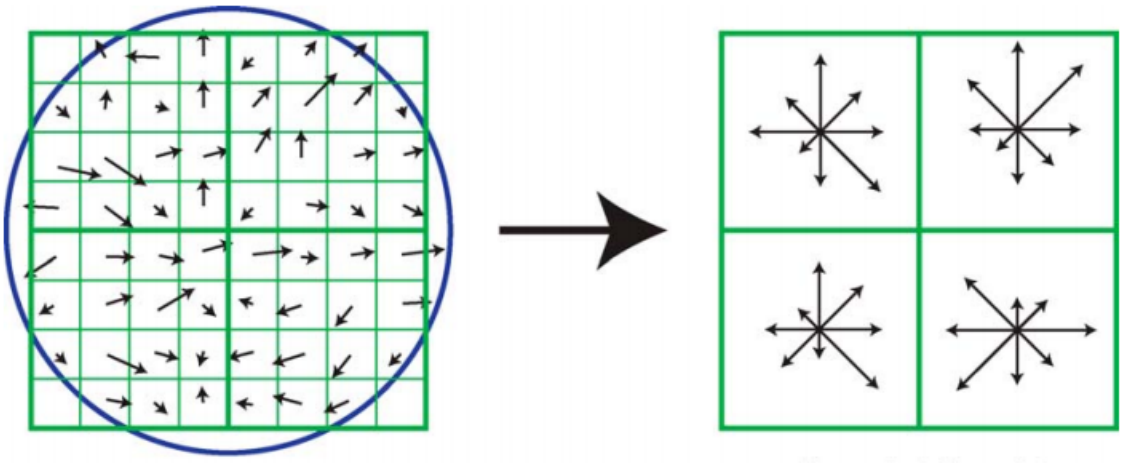


Figure 5.1: (a) Left: Image gradients. **(a) Right:** Keypoint descriptor. Figure from [Low04].

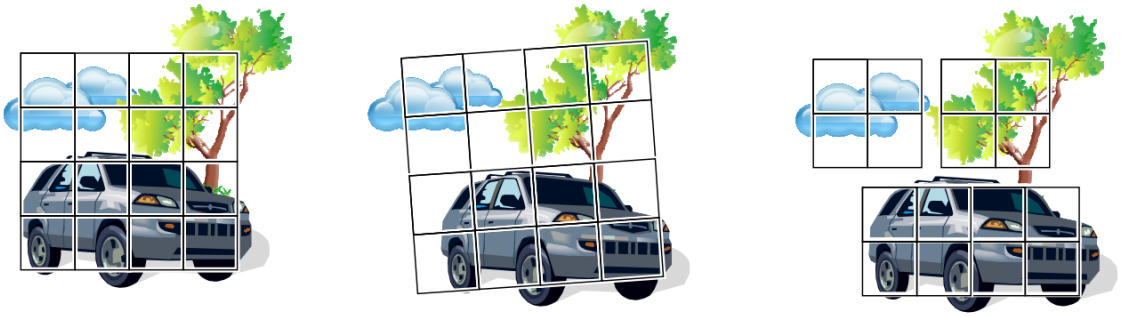


Figure 5.2: Each sub-quadrant has an additional extra degree to internally re-optimize its position. **(a) Left:** Reference image. **(b) Center:** Position of a patch with optimal SIFT matching. **(a) Right:** The SIFT patch is split into 4 x 4 sub-quadrants with re-optimized positions. Figure from [WRHS13].

Descriptor matching approaches, like HOG and SIFT, rely on rigid patches and are successful for rigid motion, but are not so precise for non-rigid motion, nor for weak or repetitive textures. To overcome this drawback Weinzaepfel et al. [WRHS13] introduced a novel approach called deep matching.

Splitting the SIFT patch into 4 sub-quadrants and later optimizing the position of each of those sub-quadrants, yields a coarse non-rigid matching, as shown in Figure 5.2. Applying this algorithm recursively, a non-rigid matching with explicit pixel-wise correspondences can be achieved.

The next problem was of weak or repetitive textures. Weinzapefel et al. [WRHS13] presented the matching algorithm as a multi-layer architecture. Deep matching considers patches at several scales. At each scale, local, rigid matches are computed. Local matches are propagated up the hierarchy, leading to a bottom-up approach. Patches are broken into sub-patches, making a quadtree-like subdivision approach (see Figure 5.3).

To deal with the problem of rigid matches, every sub-patch on each level has an extra degree of freedom to locally re-optimize the positions of each quadrant.

Advantages:

- Having a quadtree-like subdivision with easier matching problems on coarser grids yields to correctly matching repetitive patterns.
- Even in weakly textured areas, the method retrieves dense correspondences from every matched patch.
- Dealing with various sources of image deformations.

All the above named advantages are the reason for using this algorithm as the matching step for the work done in this thesis. After computing the matches with deep matching a sparse set of matches will remain. According to [RWHS15] one match per 90 pixels

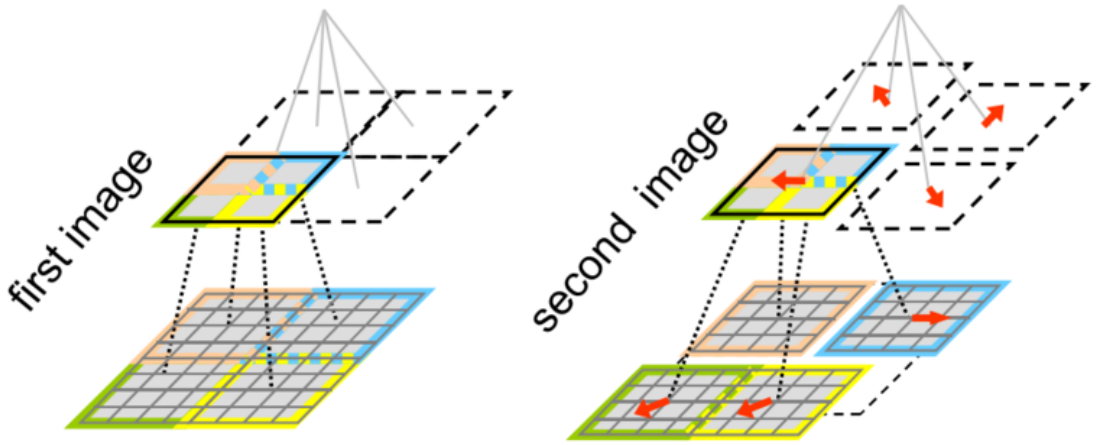


Figure 5.3: Each patch is broken into sub-patches, leading to a quadtree-like subdivision. Every sub-patch can optimize its position. Figure from [WRHS13].

is obtained, however for the final step of the optic flow pipeline every pixel should be initialised.

Apart from the deep matching algorithm, the CPM method [LHS+17] is also going to be used in the evaluation part of this thesis. The CPM algorithm, like the deep matching algorithm, makes use of the SIFT features to measure the similarities of matches. However, compared to deep matching, CPM has higher density of the input matches (see Figure 5.4).

5.2 Inpainting

The output of the previous step is a sparse set of matches $M = \{(x_t, y_t), (x_{t+1}, y_{t+1})\}$, where each match defines correspondence between a pixel (x_t, y_t) in the first frame t, and a pixel (x_{t+1}, y_{t+1}) in the second frame $t + 1$, thus $(u_M, v_M) = (x_{t+1} - x_t, y_{t+1} - y_t)$.

The next step is the sparse-to-dense interpolation from the already computed matches. For the inpainting functional, a functional of a form like following is used:

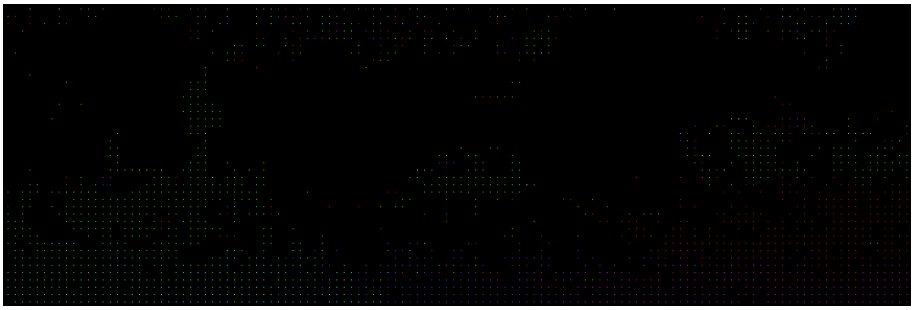
$$E_{\text{inpaint}}(u, v) = \int_{\Omega} c \cdot E_M(u, v) + (1 - c) \cdot \alpha_{inpaint} E_S(u, v) dxdy \tag{5.1}$$

where E_M is the similarity term, $\alpha_{inpaint}$ the smoothness parameter and c is an indicator function, defined as

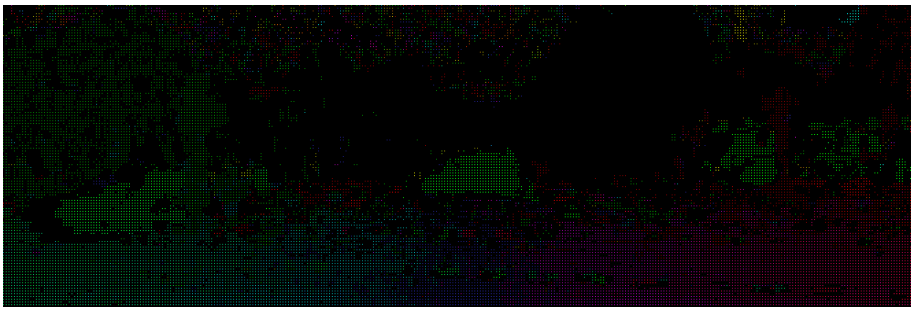
$$c(x, y) = \begin{cases} 1 & \text{if there is a match at pixel } (x, y), \\ 0 & \text{otherwise.} \end{cases} \tag{5.2}$$

(a)

(b)



(c)



(d)

Figure 5.4: Training sequence #15 of the KITTI 2015 benchmark [MG15]: **(a) Top:** Image frame at time t. **(b) Upper Middle:** Image frame at time t+1 **(c) Lower Middle:** Matches computed with the deep matching algorithm between frames t and t+1. For an image of size 1242 x 375 pixels, in total 6802 matches are computed. **(d) Bottom:** Matches computed with the CPM algorithm, in total 44634 matches are computed.

The similarity term is defined as

$$E_M = \Psi\Big((u - u_M)^2 + (v - v_M)^2\Big). \tag{5.3}$$

At positions with a match ($c = 1$), the only contribution comes from the similarity term E_M. However, for positions without matches ($c = 0$), the information regarding the optical flow is filled in from the neighbours.

During the design of the energy functional, the similarity term is fixed and the main challenge remains to design a good smoothness term.

5.2.1 Inpainting Smoothness Term

The smoothness terms described in Chapter 3 had one main drawback, namely, the motion boundaries would be smoothed. Consequently, the smoothness terms should be improved.

A possible extension is to integrate the image gradient information in the smoothness term, by adding an additional weighting function $g(s^2)$. This weighing function depends on the magnitude of the spatial image gradient $|\nabla I|$. Multiplying the smoothness term with $g(|\nabla I|^2)$ scales the eigenvalues of the diffusion tensor, hence smoothing is reduced at image edges. The function g should be chosen as described in Chapter 3. In this case, both eigenvalues will be scaled with the same factor. A further possible extension is scaling the eigenvalues with the gradient projection $g(|\nabla I^\top \mathbf{r}_i|^2)$, $i = \{1, 2\}$, where $\mathbf{r}_i$ are the eigenvectors of either the structure tensor, or the regularisation tensor [ZBW11].

This section proposes image driven first and second order smoothness terms, build on the anisotropic first order regularisation (see Equation (3.12)) and direct single anisotropic second order regularisation (see Equation (3.15)) from Chapter 3.

Extensions of the First Order Smoothness Terms

The previously described constraint adaptive regulariser (see Equation (3.12)) relates to the following diffusion tensor

$$D_{con-adapt} = (\mathbf{r}_1, \mathbf{r}_2) \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix} \begin{pmatrix} \mathbf{r}_1^\top \\ \mathbf{r}_2^\top \end{pmatrix}, \tag{5.4}$$

where $d_1 = \Psi_{S_1}{}'\Big(\Big(\mathbf{r}_1^\top \nabla u\Big)^2 + \Big(\mathbf{r}_1^\top \nabla v\Big)^2\Big)$ and $d_2 = \Psi_{S_2}{}'\Big(\Big(\mathbf{r}_2^\top \nabla u\Big)^2 + \Big(\mathbf{r}_2^\top \nabla v\Big)^2\Big)$.

Based on this smoothness term and the weighing functions, two other smoothness terms are proposed:

- Firstly

$$E_{\text{grad-con}} = \sum_{i=1}^{2} g\Big(|\nabla I|^2\Big) \cdot \Psi_i\Big(|\nabla u^\top \mathbf{r}_i|^2 + |\nabla v^\top \mathbf{r}_i|^2\Big), \tag{5.5}$$

  which will be referred to as the *gradient constraint adaptive* smoothness term. The related diffusion tensor reads

$$D_{\text{grad-con}} = (\mathbf{r}_1, \mathbf{r}_2) \begin{pmatrix} g\Big(|\nabla I|^2\Big) \cdot d_1 & 0 \\ 0 & g\Big(|\nabla I|^2\Big) \cdot d_2 \end{pmatrix} \begin{pmatrix} \mathbf{r}_1^\top \\ \mathbf{r}_2^\top \end{pmatrix}. \tag{5.6}$$

- Secondly

$$E_{\text{proj-con}} = \sum_{i=1}^{2} g\Big(|\nabla I^\top \mathbf{r_i}|^2\Big) \cdot \Psi_i\Big(|\nabla u^\top \mathbf{r}_i|^2 + |\nabla v^\top \mathbf{r}_i|^2\Big), \tag{5.7}$$

  which will be referred to as the *projection constraint adaptive* smoothness term.

  The related diffusion tensor reads

$$D_{\text{proj-con}} = (\mathbf{r}_1, \mathbf{r}_2) \begin{pmatrix} g\Big(|\nabla I^\top \mathbf{r}_1|^2\Big) \cdot d_1 & 0 \\ 0 & g\Big(|\nabla I^\top \mathbf{r}_2|^2\Big) \cdot d_2 \end{pmatrix} \begin{pmatrix} \mathbf{r}_1^\top \\ \mathbf{r}_2^\top \end{pmatrix}, \tag{5.8}$$

  where $d_1 = \Psi_{S_1}{}'\Big(\Big(\mathbf{r}_1^\top \nabla u\Big)^2 + \Big(\mathbf{r}_1^\top \nabla v\Big)^2\Big)$ and $d_2 = \Psi_{S_2}{}'\Big(\Big(\mathbf{r}_2^\top \nabla u\Big)^2 + \Big(\mathbf{r}_2^\top \nabla v\Big)^2\Big)$, as before.

Extensions of the Second Order Smoothness Terms

In the Chapter 3 direct second order regularisation strategies are presented. The focus will be on the single anisotropic. Most of the concepts in this section, regarding the diffusion process and second order regularisation can be found more extensive in [MSV+17].

Compared to the first order regularisers, which lead to a second order diffusion process, the direct second order regularisers leads to a fourth order diffusion process.

For the single anisotropic regularisation (see Equation (3.15)) the related diffusion tensor reads

$$D_{\text{aniso-single}} = \begin{pmatrix} \Psi_1{}' & 0 & 0 & 0 \\ 0 & \Psi_2{}' & 0 & 0 \\ 0 & 0 & \Psi_1{}' & 0 \\ 0 & 0 & 0 & \Psi_2{}' \end{pmatrix}, \tag{5.9}$$

where Ψ_1' $= \Psi_1$'$(|\boldsymbol{\mathcal{H}}u\mathbf{r}_1|^2 + |\boldsymbol{\mathcal{H}}v\mathbf{r}_1|^2)$ and Ψ_2' $= \Psi_2$'$(|\boldsymbol{\mathcal{H}}u\mathbf{r}_2|^2 + |\boldsymbol{\mathcal{H}}v\mathbf{r}_2|^2)$

In addition to the single anisotropic regularisation, two other smoothness terms are proposed:

- Firstly

$$E_{\text{grad-aniso-single}}(\boldsymbol{w}) = \sum_{l=1}^{2} g\Big(|\nabla I|^2\Big) \cdot \Psi_l\Bigg(\sum_{k=1}^{2}\Big(\boldsymbol{r_k}^\top \boldsymbol{\mathcal{H}}u\boldsymbol{r}_l\Big)^2 + \Big(\boldsymbol{r}_k^\top \boldsymbol{\mathcal{H}}v\boldsymbol{r}_l\Big)^2\Bigg), \quad (5.10)$$

  where $\boldsymbol{w} = (u, v)^\top$ is the flow field and the related diffusion tensor reads

$$D_{\text{grad-aniso-single}} = \begin{pmatrix} g \cdot \Psi_1\text{'} & 0 & 0 & 0 \\ 0 & g \cdot \Psi_2\text{'} & 0 & 0 \\ 0 & 0 & g \cdot \Psi_1\text{'} & 0 \\ 0 & 0 & 0 & g \cdot \Psi_2\text{'} \end{pmatrix}, \quad (5.11)$$

  where $g = g\Big(|\nabla I|^2\Big)$. This smoothness term will be referred to as the *gradient single anisotropic* smoothness term.

- Secondly

$$E_{proj-aniso-single} = (\boldsymbol{w}) = \sum_{l=1}^{2} g\Big(|\nabla I^\top \mathbf{r}_l|^2\Big) \cdot \Psi_l\Bigg(\sum_{k=1}^{2}\Big(\boldsymbol{r_k}^\top \boldsymbol{\mathcal{H}}u\boldsymbol{r}_l\Big)^2 + \Big(\boldsymbol{r}_k^\top \boldsymbol{\mathcal{H}}v\boldsymbol{r}_l\Big)^2\Bigg), \quad (5.12)$$

  In the following text this will be denoted as *projection single anisotropic* smoothness term. The related diffusion tensor reads

$$D_{\text{proj-aniso-single}} = \begin{pmatrix} g1 \cdot \Psi_1\text{'} & 0 & 0 & 0 \\ 0 & g2 \cdot \Psi_2\text{'} & 0 & 0 \\ 0 & 0 & g1 \cdot \Psi_1\text{'} & 0 \\ 0 & 0 & 0 & g2 \cdot \Psi_2\text{'} \end{pmatrix}, \quad (5.13)$$

  where $g_1 = g\Big(|\nabla I^\top \mathbf{r}_1|^2\Big)$ and $g_2 = g\Big(|\nabla I^\top \mathbf{r}_2|^2\Big)$.

5.3 Variational Refinement

For the final step of the optic flow pipeline—variational refinement, the energy functional is of the form like (2.9). For the data term both the brightness (2.3) and gradient (2.26) constancy assumption are used and for the smoothness term the single anisotropic term from the indirect approach (3.18). Combined together the following energy functional is derived:

$$\begin{aligned} E_{\text{var}}(\mathbf{w}) = \int_\Omega & \underbrace{\Psi_{CH}\Big(\big(I_1(\mathbf{x}+\mathbf{w}) - I_0(\mathbf{x})\big)^2\Big) + \gamma \cdot \Psi_{CH}\Big(|\nabla I_1(\mathbf{x}+\mathbf{w}) - \nabla I_0(\mathbf{x})|^2\Big)}_{\text{data term}} \, d\mathbf{x} \\ & + \alpha \int_\Omega \inf_{\boldsymbol{a},\boldsymbol{b}} \underbrace{\Bigg\{ \sum_{l=1}^{2} \Psi_l\Big(\big(\boldsymbol{r}_l^\top(\nabla u - \boldsymbol{a})\big)^2 + \big(\boldsymbol{r}_l^\top(\nabla v - \boldsymbol{b})\big)^2\Big)}_{\text{coupling term}} \\ & + \beta \underbrace{\sum_{l=1}^{2} \Psi_l\Bigg(\sum_{k=1}^{2} \big(\boldsymbol{r}_k^\top \boldsymbol{\mathcal{J}} \boldsymbol{a} \boldsymbol{r}_l\big)^2 + \big(\boldsymbol{r}_k^\top \boldsymbol{\mathcal{J}} \boldsymbol{b} \boldsymbol{r}_l\big)^2\Bigg)\Bigg\}}_{\text{smoothness term}} d\mathbf{x}, \end{aligned} \tag{5.14}$$

where $I_1 = I(t+1)$ and $I_0 = I(t)$. Ψ_1 and Ψ_2 are the Perona-Malik and Charbonnier penaliser, respectively, with $\epsilon = 0.01$.

This functional is further used for the evaluation step of the thesis. Since the goal of this thesis is not optimizing this functional, the parameters γ, α and β are fixed at 69.3, 186.2 and 9.83 respectively.

6 Evaluation and Experiments

Using an benchmark with known camera motion and scene structure, it is possible to compute optical flow for a real image sequence, which will represent the ground truth of the optical flow [BE96]. The evaluation of the optical flow can be done either quantitatively and/or qualitatively. Qualitative flow evaluation is useful for a general judgment [BT05] and it is usually done if no ground truth exists.

Given the ground truth flow u^t, it is possible to measure the quality of the computed, estimated flow field u^e. The quality metrics are defined as follows [Fle92]:

- *Average Endpoint Error:*

$$AEE(u^t, u^e) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \sqrt{(u^t_{i,j} - u^e_{i,j})^2 + (v^t_{i,j} - v^e_{i,j})^2}. \tag{6.1}$$

- *Bad Pixel Error:*

$$BP(u^t, u^e) = \frac{100}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} 1_{(\sqrt{(u^t_{i,j} - u^e_{i,j})^2 + (v^t_{i,j} - v^e_{i,j})^2} > T)}. \tag{6.2}$$

The evaluation and experiment error will be displayed using the colour plot, where directions are visualized as colour and magnitude as brightness [Bru16]. An illustration is shown in Figure 6.1. Given the vector field $\mathbf{u} = (\mathbf{u}^\top_{1,1}, ..., \mathbf{u}^\top_{N,M})^\top$ with $\mathbf{u}_{i,j} = (u_{i,j}, v_{i,j})^\top$, polar coordinates are used for $\mathbf{u}_{i,j}$ with radius $r_{i,j}$ and angle $\phi_{i,j}$.

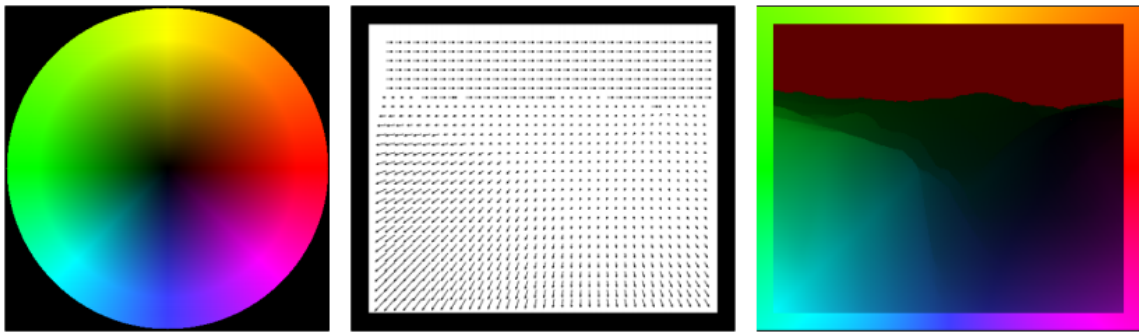


Figure 6.1: (a) Left: Colour code. **(b) Center:** Vector representation of the flow field. **(c) Right:** Corresponding colour representation. Figure from [Bru06].

The last missing thing to quantitatively evaluate the optical flow are the test datasets. Because of the complex test scenes, the following two are used:

- **MPI-Sintel dataset** [BWSB12] is a large and difficult dataset. It includes large displacements and significant occlusion.
- **Kitti 2015 dataset** [MG15] is composed of real-world images captured from a moving vehicle (see Figure 6.2). Main challenges of the Kitti 2015 dataset are specularities, dominant non-fronto-parallel surfaces, variable displacement magnitudes and changes in the illumination [BDB13].

During the evaluation, only the training sequences are used from both datasets, since the ground truth is provided. For the Kitti 2015 dataset, both the occ and noc training sets are used and compared separately.

6.1 Comparison of the Smoothness Term

One of the core contributions of this thesis is to introduce smoothness terms, which respect motion boundaries. In Section 5.2.1 possible transformations are presented. In the following, four of them are evaluated, together with the constraint adaptive regulariser and additionally the single anisotropic regularisation.

For this part of the evaluation only the inpainting step is considered with the energy functional (5.1). For the evaluation both deep matches and CPM are used as the input.

Evaluation Setup. To implement the similarity term (5.3) of the inpainting energy functional the Charbonnier penaliser (3.4) is used with ϵ fixed at 0.01.

For the experiments, the complete KITTI 2015 occ and noc training data sets and MPI Sintel training clean data set are used to evaluate the performance of all regularisers from Section 5.2.1. Therefore, first ϵ_g from the geodesic penaliser (4.4) is optimized w.r.t the percentage of erroneous pixels (BP) using downhill simplex on a small subset of the training data. For the optimization of the $\alpha_{inpaint}$, the complete training data sets are used for the evaluation of common metrics - AEE and BP in case of KITTI 2015 and AEE in case of MPI Sintel.

6.1.1 Minimisation of the Inpainting Functional

The energy functional (5.1) is non-convex and non-linear. To solve it, the framework of Brox et al. [BBPW04] is used with an incremental reduced coarse-to-fine scheme [MSB17a], using 5 resolution levels. The downsampling factor is fixed at $\eta = 0.95$. The remaining equations are still non-linear due to the robust penalisers. Therefore, 20 inner fixed point iterations are applied and 5 outer iterations.

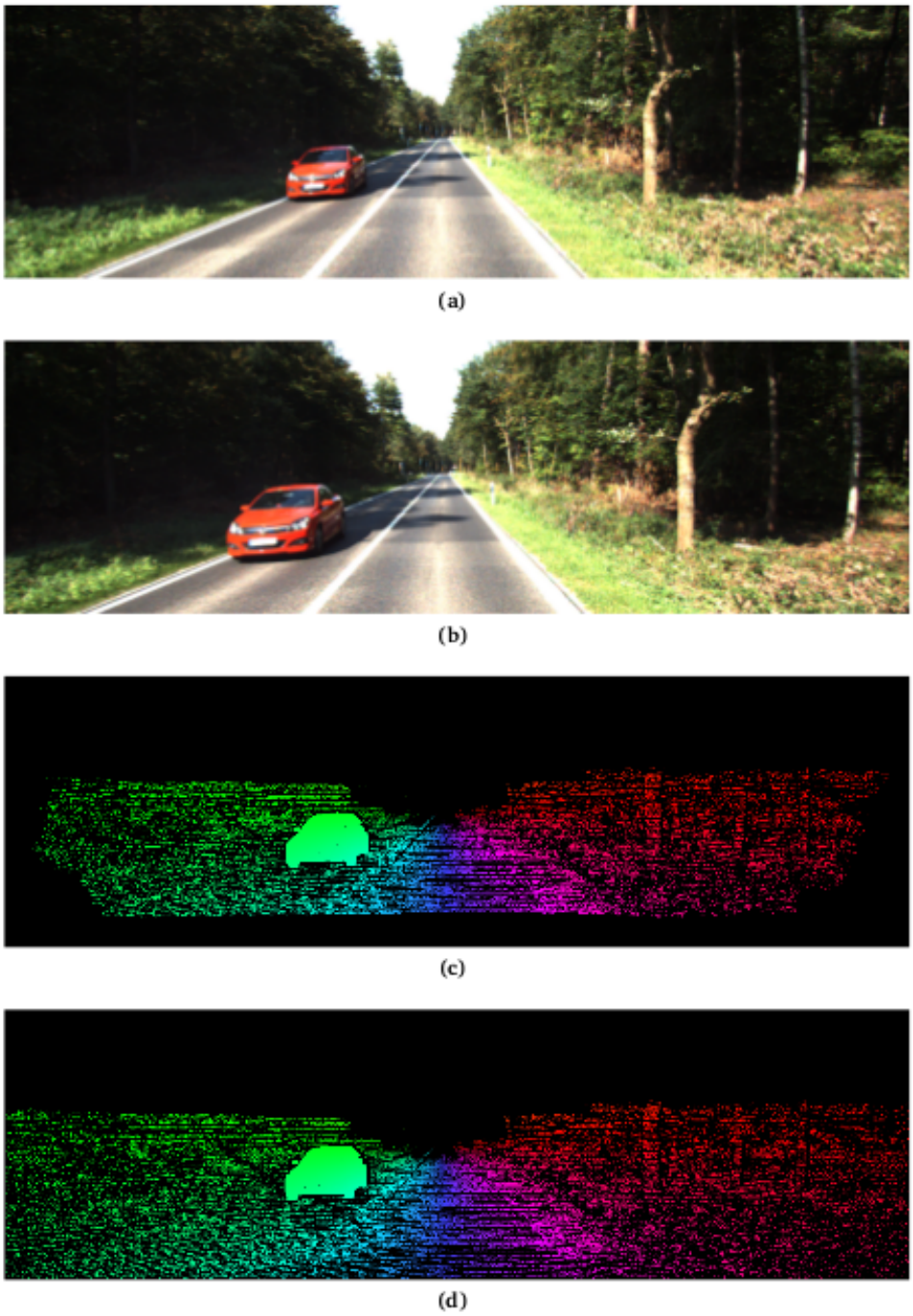


Figure 6.2: Training sequence #63 of the KITTI 2015 benchmark [MG15]. **(a) Top:** Image frame at time t. **(b) Upper Middle:** Image frame at time t+1. **(c) Lower Middle:** Ground truth for noc training sequence. **(d) Bottom:** Ground truth for occ training sequence.

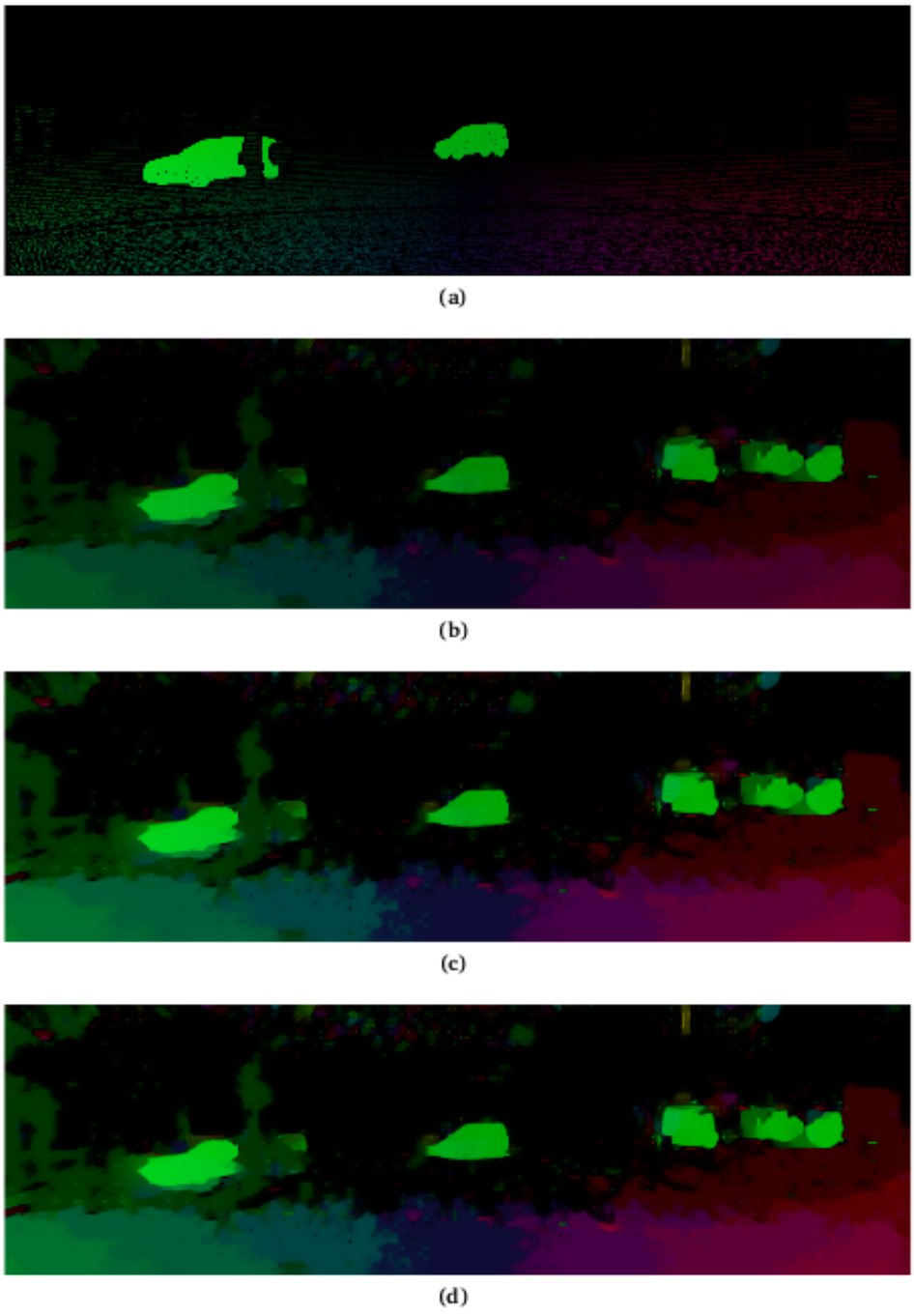
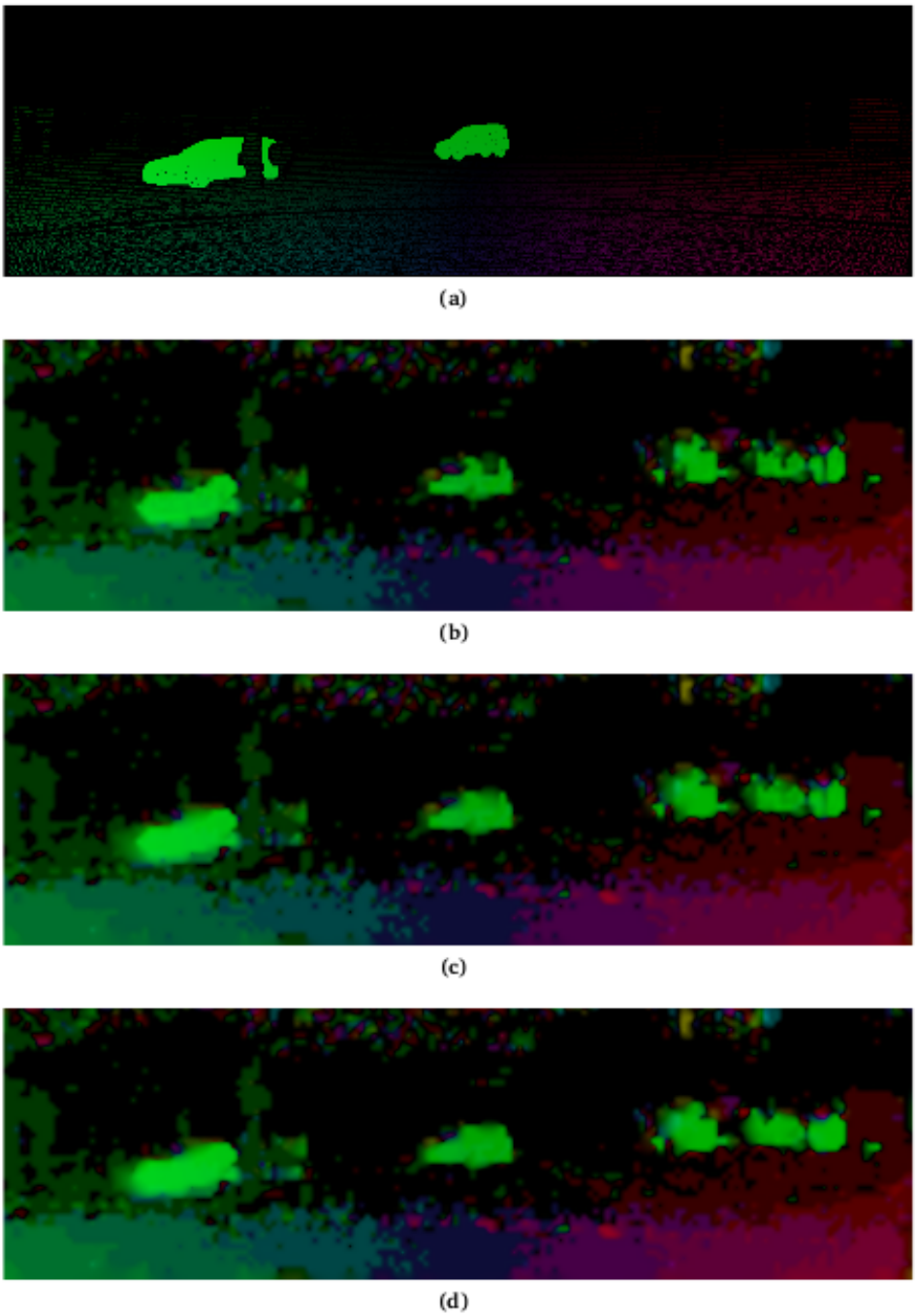


Figure 6.3: Training sequence #15 of the KITTI 2015 benchmark [MG15]. Deep matches are used as the input for the inpainting step with different first order smoothness terms. **(a) Top** Ground truth. **(b) Upper Middle:** Constraint adaptive regulariser. **(c) Lower Middle:** Gradient constraint adaptive regulariser. **(d) Bottom:** Projection constraint adaptive regulariser.



(a)

(b)

(c)

(d)

Figure 6.4: Training sequence #15 of the KITTI 2015 benchmark [MG15]. Deep matches are used as the input for the inpainting step with different second order smoothness terms. **(a) Top:** Ground truth. **(b) Upper Middle:** Single anisotropic regulariser. **(c) Lower Middle:** Gradient single anisotropic regulariser. **(d) Bottom:** Projection single anisotropic regulariser.

6.1.2 Results

After the first optimization $\epsilon_g \approx 155$. For this value, the geodesic penaliser gets close to 1, making the difference between the constraint adaptive regulariser, and the gradient- and projection- constraint adaptive regulariser, almost non-existent. The same goes for the second order terms. In Figures 6.3, 6.4, 6.5 and 6.6 the datasets are used to compare the different types of regularisers.

- *Constraint Adaptive Regulariser*. Using the first order regularisers (Figure 6.3 and Figure 6.5), obvious sharper edges are achieved compared to the second order regularisers (Figure 6.4 and 6.6).
- *Gradient Constraint Adaptive regulariser* and *Projection Constraint Adaptive regulariser*. Using the gradient- and projection- constraint adaptive regulariser does not show a noticeable visual improvement in this sequence, mostly due to the very large ϵ_g, making the whole function $g(s^2)$ close to one.
- *Single Anisotropic regulariser*. One can clearly see that all of the second order regularisers (Figure 6.4 and Figure 6.6) give oversegmented results and they also have a problem with not respecting motion discontinuities.
- *Gradient Single Anisotropic regulariser* and *Projection Single Anisotropic regulariser*. The gradient- and projection- single anisotropic regularisers show improvement w.r.t. motion discontinuities, compared to the single anisotropic case, but still not as good as the first order regularisers.

The Function (5.1) is very sensitive to outliers, which is mostly visible during the comparison between the inpainting, which uses deep matches as input and inpainting, which uses CPM as input. CPM has higher density of the input matches, compared to deep matches, therefore, it is more sensitive to having more outliers. In case of an outlier, the indicator function c has the value 1 and hence, there will be no smoothing. The associated metrics are listed in Tables 6.1 and 6.2.

6.2 Comparison with EpicFlow

In the Chapter 4 the last two steps of EpicFlow are shortly explained. In this Section, the comparison is twofold. Firstly, the results of the EpicFlow inpainting step are compared with the results of the inpainting function (5.1) with the proposed smoothness terms, as explained previously. Secondly, the results of the EpicFlow with the refinement are compared to the results derived after the variational refinement step.

Order	Regularisation Type	Kitti 2015				Sintel
		noc		*occ*		clean
		AEE	BP (%)	AEE	BP (%)	AEE
1^{st}	Con. adaptive	4.899	21.335	10.823	30.167	3.036
	Gradient con. adaptive	4.889	21.249	10.823	30.165	3.053
	Projection con. adaptive	**4.882**	**21.179**	11.234	**29.982**	3.054
2^{nd}	Single aniso.	5.037	21.875	9.819	30.484	3.442
	Gradient single aniso.	5.034	21.797	10.136	30.002	3.439
	Projection single aniso.	5.086	21.766	9.818	30.329	3.442
	EpicFlow[1]	5.12	24.74	**9.37**	31.96	**2.68**

Table 6.1: Inpainting results using deep matches, optimized on average endpoint error (AEE), percentage of erroneous pixels (BP), where endpoint error > 3px, for the different regularisers on the KITTI 2015 [MG15] for both, OCC-training sets and NOC-training sets and on the MPI Sintel clean training set.

[1] Data taken from [MSB17a]

Order	Regularisation Type	Kitti 2015				Sintel
		noc		*occ*		clean
		AEE	BP (%)	AEE	BP (%)	AEE
1^{st}	Con. adaptive	5.008	25.298	11.069	33.114	3.064
	Gradient con. adaptive	5.016	25.014	11.161	32.885	3.063
	Projection con. adaptive	4.987	25.037	11.025	32.906	3.086
2^{nd}	Single aniso.	4.802	24.886	9.407	30.387	3.442
	Gradient single aniso.	4.802	24.887	9.268	30.360	3.442
	Projection single aniso.	4.802	24.885	9.267	30.360	3.438
	EpicFlow[1]	**3.66**	**16.43**	**7.77**	**23.36**	**2.19**

Table 6.2: Inpainting results for CPM, optimized on average endpoint error (AEE), percentage of erroneous pixels (BP), where endpoint error > 3px, for the different regularisers on the KITTI 2015 [MG15] for both, OCC-training sets and NOC-training sets and on the MPI Sintel clean training set.

[1] Data taken from [MSB17a]

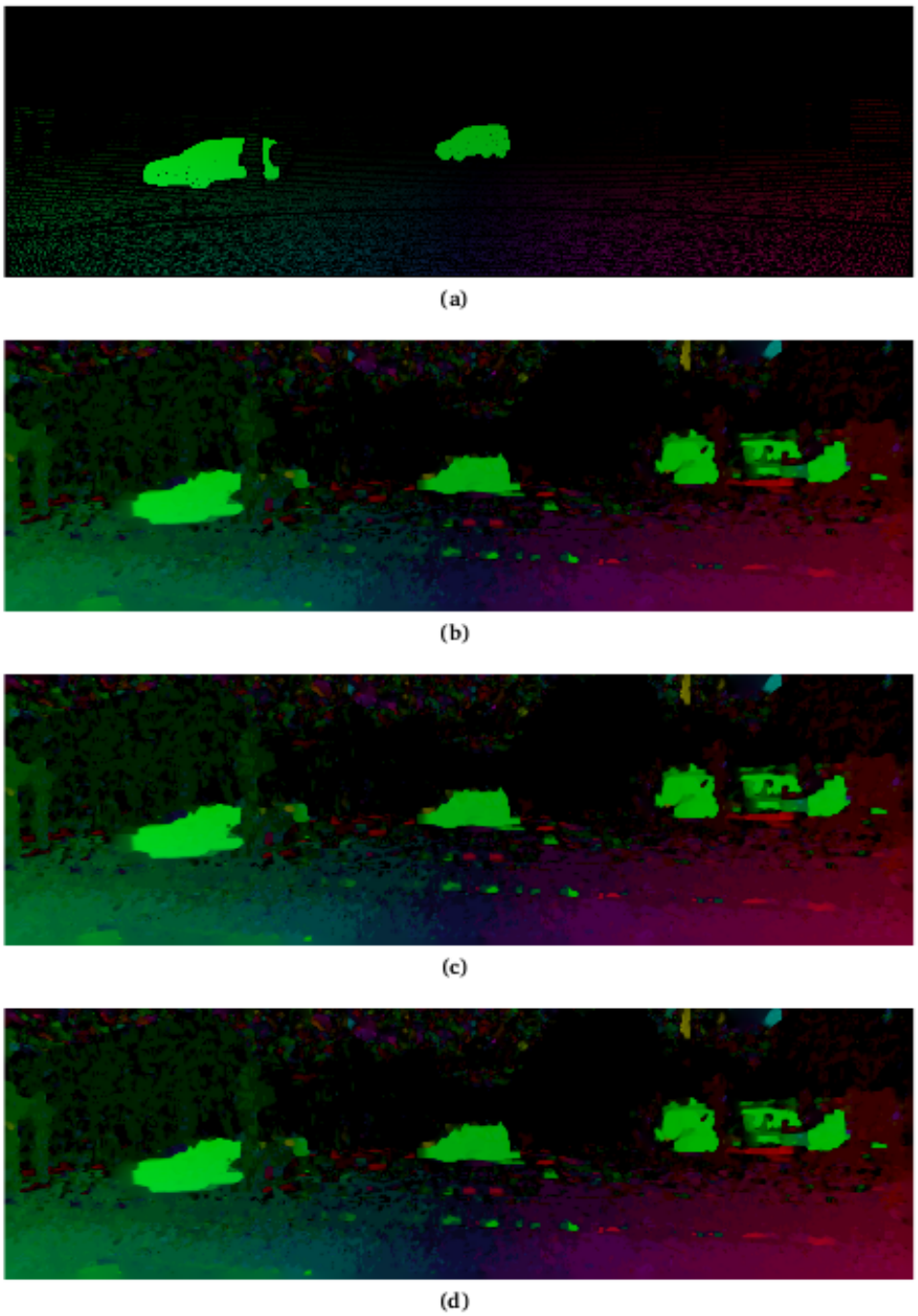


Figure 6.5: Training sequence #15 of the KITTI 2015 benchmark [MG15]. CPM are used as the input for the inpainting step with different first order smoothness terms. **(a) Top** Ground truth. **(b) Upper Middle:** Constraint adaptive regulariser. **(c) Lower Middle:** Gradient constraint adaptive regulariser. **(d) Bottom:** Projection constraint adaptive regulariser.

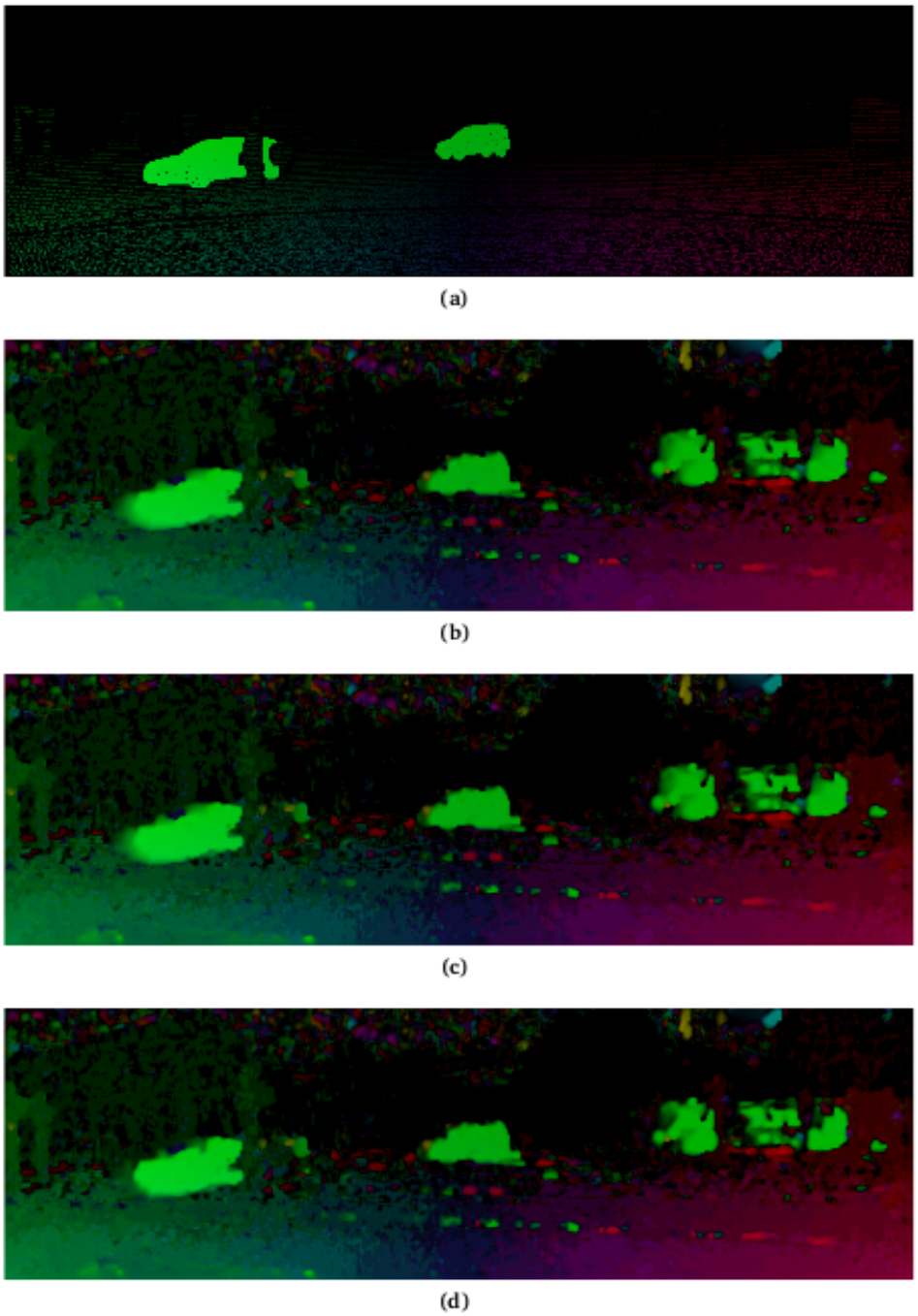


Figure 6.6: Training sequence #15 of the KITTI 2015 benchmark [MG15]. CPM are used as the input for the inpainting step with different second order smoothness terms. **(a) Top:** Ground truth. **(b) Upper Middle:** Single anisotropic regulariser. **(c) Lower Middle:** Gradient single anisotropic regulariser. **(d) Bottom:** Projection single anisotropic regulariser.

6.2.1 Minimisation

To minimise the non-convex energy functional (5.14) the associated Euler Lagrange equations are derived and solved by means of an reduced incremental coarse-to-fine fixed point iteration with a downsampling factor of $\eta = 0.95$, using 5 resolution levels. On each resolution level a flow increment du and dv and the increments da and db of the auxiliary functions are computed [MSB17b]. The resulting non-linear system of equations are solved as a series of linear systems by keeping the non-linear expressions from the data and smoothness term fixed. The optical flow is not estimated from scratch, instead, the flow provided by the preceding inpainting step is refined.

6.2.2 Results

Again, the results are computed for both deep matching and CPM. The results are provided in the Tables 6.1, 6.2, 6.3 and 6.4.

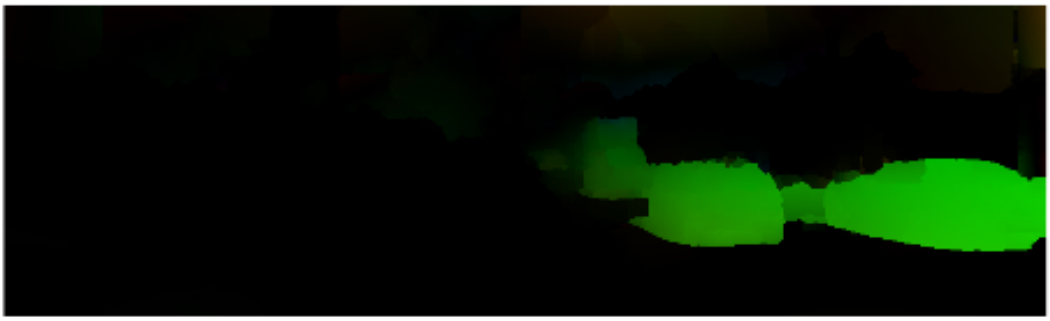
Inpainting step with deep matching. Table 6.1 shows results for the inpainting step of both EpicFlow and inpainting with the energy functional (5.14). Because deep matches do not contain enough matches to extrapolate the needed information of motion boundaries, the proposed inpainting step outperforms EpicFlow inpainting on the Kitti 2015 benchmark for all, but the AEE of the occ data set. EpicFlow gives better results on the Sintel benchmark. The best results for the inpainting functional are derived using the projection constraint adaptive regulariser. Figure 6.7 illustrates the difference between the two EpicFlow method and the proposed method with the projection constraint adaptive smoothness term.

Inpainting step with CPM. The CPM matches are denser than deep matches and thus may contain more information about motion boundaries. EpicFlow outperforms the proposed inpainting method on all benchmarks. The results are visible in Table 6.3. The best results are obtained using the projection single anisotropic smoothness term and therefore, this smoothness term is used for the comparison with the inpainting step of the EpicFlow method in Figure 6.8.

Variational refinement. Since the EpicFlow method relies on simple models for refinement (see Equation (4.3)), the refinement cannot keep up with the preceding pipeline [MSB17a]. Therefore, there is not much room for improvement. Figures 6.9 and 6.10 illustrate the difference between the variational refinement of the EpicFlow and the proposed method, using both CPM and deep matches as the input of the inpainting step. The best results are derived using the gradient constraint adaptive smoothness term and the projection single anisotropic smoothness term for deep matches and CPM respectively.

(a)

(b)



(c)

Figure 6.7: Training sequence #52 of the KITTI 2015 benchmark [MG15]. Results obtained for the inpainting step and using deep matches as the input **(a) Top:** Ground truth. **(b) Middle:** Inpainting with the projection constraint adaptive smoothness term. **(c) Bottom:** EpicFlow inpainting step.

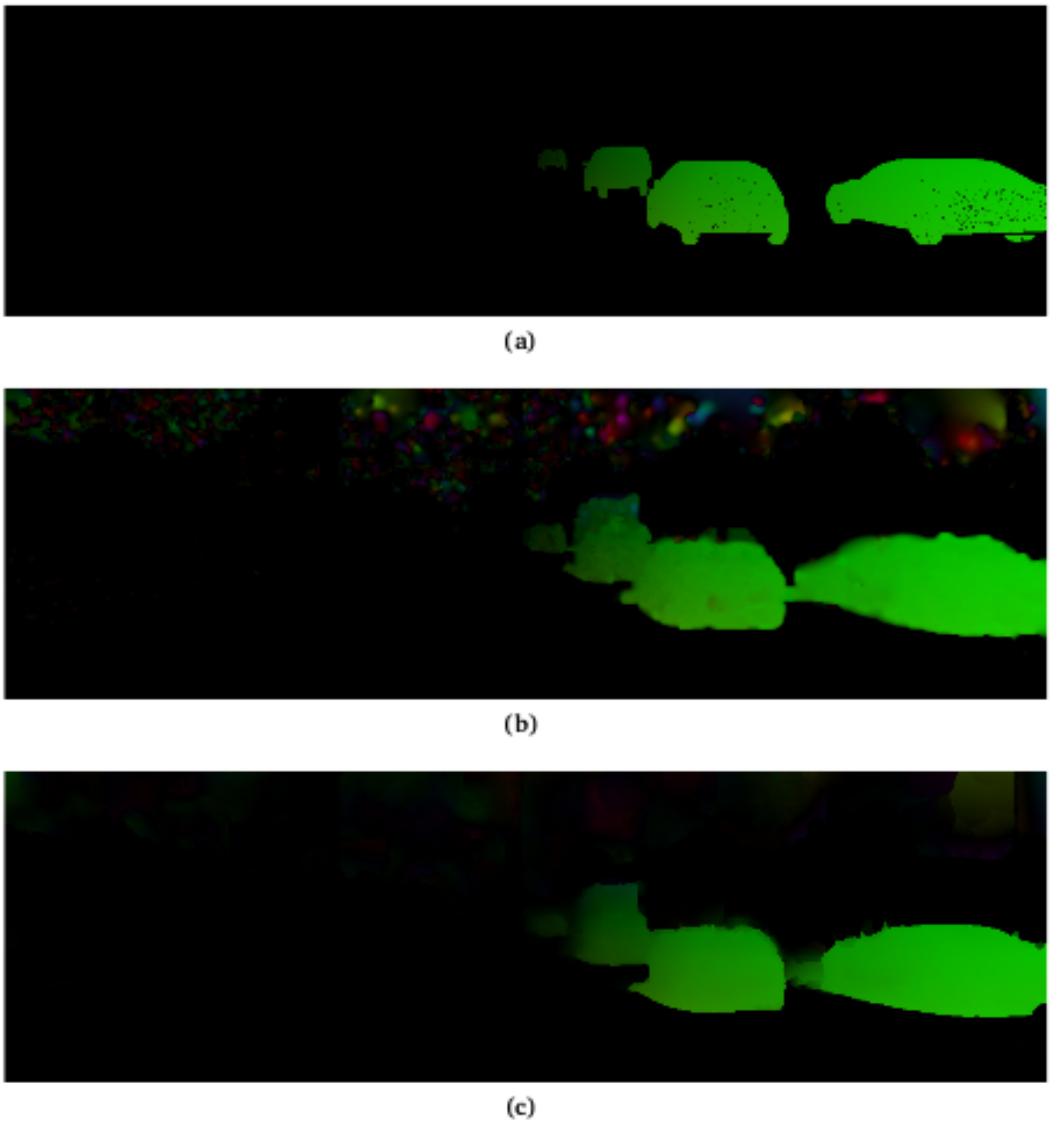


Figure 6.8: Training sequence #52 of the KITTI 2015 benchmark [MG15]. Results obtained for the inpainting step and using CPM as the input. **(a) Top:** Ground truth. **(b) Middle:** Inpainting with the projection single anisotropic smoothness term. **(c) Bottom:** EpicFlow inpainting step.

Order	Regularisation Type	Kitti 2015			
		noc		*occ*	
		AEE	BP (%)	AEE	BP (%)
1^{st}	Con. adaptive	4.176	16.006	9.596	23.458
	Gradient con. adaptive	**4.175**	**16.005**	9.573	**23.440**
	Projection con. adaptive	4.178	16.007	9.552	23.453
2^{nd}	Single aniso.	4.262	16.245	9.106	23.576
	Gradient single aniso.	4.289	16.312	9.061	23.566
	Projection single aniso.	4.226	16.250	**9.016**	23.617
	EpicFlow [1]	4.71	20.06	9.18	28.38

Table 6.3: Results of the reduced variational refinement with deep matches initialization for the different regularisers, optimized on average endpoint error (AEE), percentage of erroneous pixels (BP), where endpoint error > 3px, on the KITTI Flow 2015 [MG15] for both, OCC-training set and NOC-training set.

[1] Data taken from [MSB17a]

Order	Regularisation Type	Kitti 2015			
		noc		*occ*	
		AEE	BP (%)	AEE	BP (%)
1^{st}	Con. adaptive	3.960	15.789	9.741	23.210
	Gradient con. adaptive	3.959	15.791	9.672	23.227
	Projection con. adaptive	3.983	15.806	9.711	23.207
2^{nd}	Single aniso.	3.650	14.798	8.004	**21.512**
	Gradient single aniso.	3.656	14.873	8.251	21.626
	Projection single aniso.	3.642	14.790	7.978	21.594
	EpicFlow [1]	**3.43**	**14.58**	**7.78**	22.86

Table 6.4: Results of the reduced variational refinement with CPM initialization for the different regularisers, optimized on average endpoint error (AEE), percentage of erroneous pixels (BP), where endpoint error > 3px, on the KITTI Flow 2015 [MG15] for both, OCC-training set and NOC-training set.

[1] Data taken from [MSB17a]

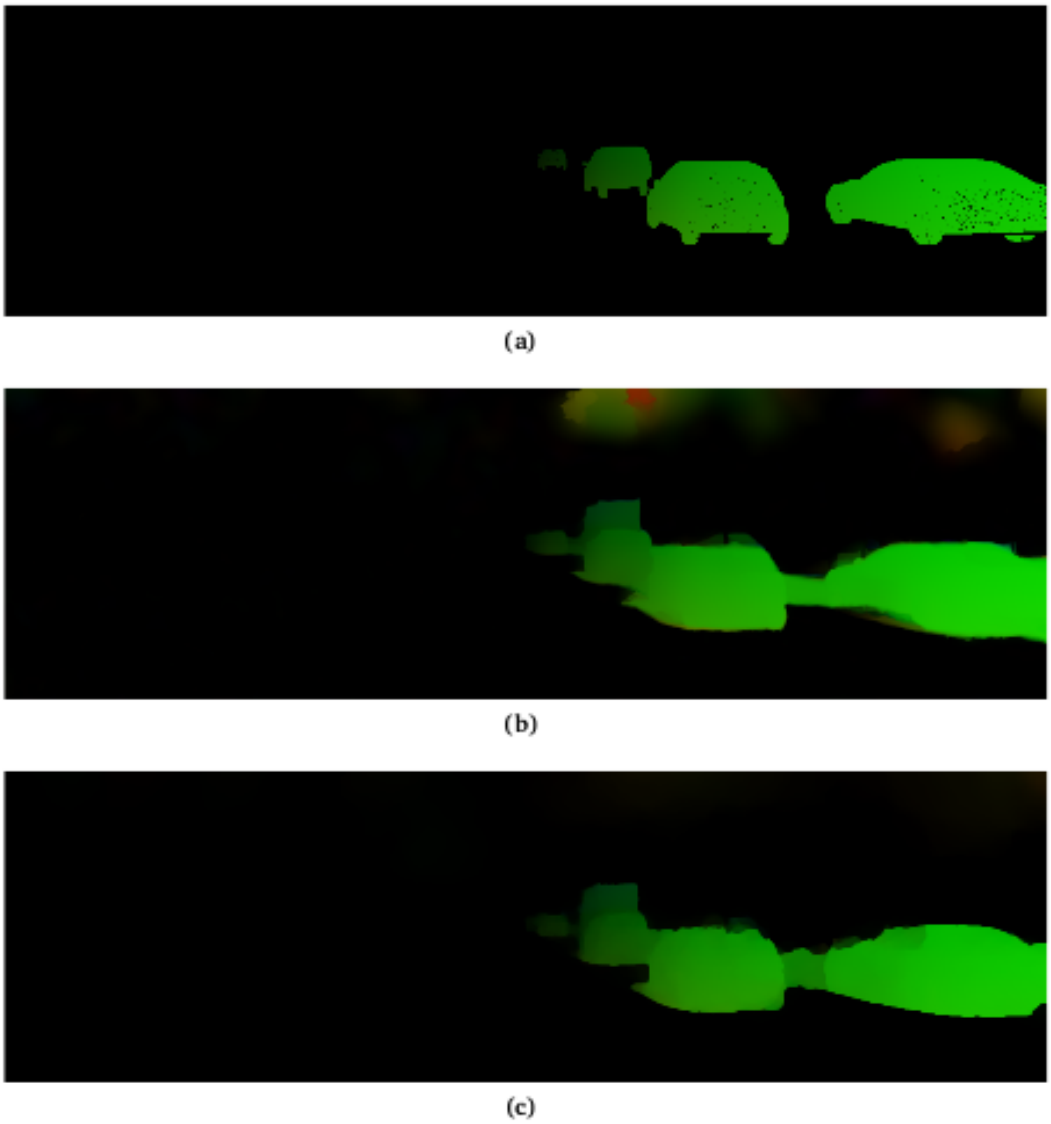
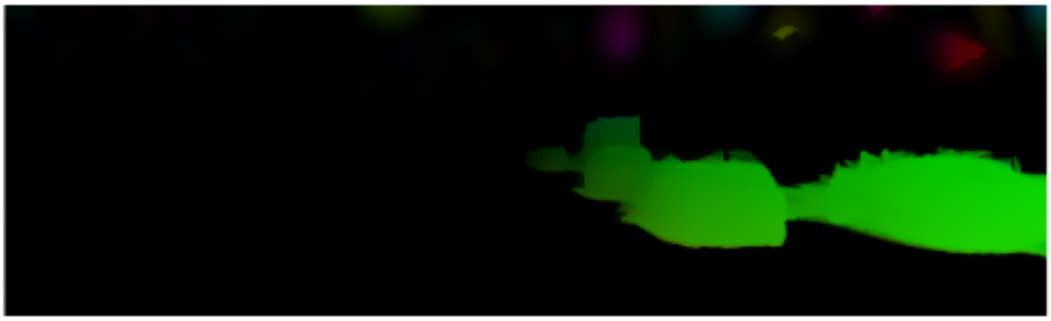


Figure 6.9: Training sequence #52 of the KITTI 2015 benchmark [MG15]. Results obtained for the variational refinement and using deep matches as the input for the inpainting step. **(a) Top:** Ground truth. **(b) Middle:** Inpainting with the gradient constraint adaptive smoothness term. **(c) Bottom:** EpicFlow with deep matches.

(a)



(b)

(c)

Figure 6.10: Training sequence #52 of the KITTI 2015 benchmark [MG15]. Results obtained for the variational refinement and using CPM as the input for the inpainting step. **(a) Top:** Ground truth. **(b) Middle:** Inpainting with the projection single anisotropic smoothness term. **(c) Bottom:** EpicFlow with CPM initialisation.

7 Conclusion

This thesis is build upon the optic flow pipeline. The focus is on improving the inpainting step for a better initialisation of the final step, namely, variational refinement. Similar to the final step, for this approach an energy functional has to be minimized. Inspired by the large displacement optical flow of Brox et al. [BBM09], this approach integrates a similarity term within a variational approach for optical flow. The inpainting energy functional consists of a similarity term and a smoothness term.

In this thesis new smoothness terms are presented for a possible improvement of the inpainting step. The proposed second order smoothness terms are an upgrade of the second order smoothness terms of Maurer et al. [MSV+17]. Furthermore, the proposed first order smoothness terms are build on the constraint adaptive regulariser from Zimmer et al. [ZBW11]. In addition to that, the final results of both, the inpainting step and the variational refinement, are compared with the corresponding steps of the existing EpicFlow method [RWHS15].

In Chapter 2 the basic concepts have been introduced. Advanced smoothness terms are then presented in the Chapter 3. Chapter 4 gives a view into the recent work done regarding this topic, with a particular focus on the EpicFlow method. In Chapter 5 several smoothness term extensions were presented, where the information of the image edges was embedded in the anisotropic smoothing behaviour. Finally, for comparing all extension with each other, in Chapter 6 an evaluation methodology and test datasets were presented. In addition, a comparison with the EpicFlow method was performed.

The evaluation shows that the constraint adaptive and the proposed projection constraint adaptive and gradient constraint adaptive regularisation give very similar results for both benchmarks and using both deep matches and CPM. This is the result of getting a large $\epsilon_g = 155$ after the optimisation process and therefore making the additional weighting function close to 1. However, the proposed second order regularisations qualitatively respect better motion boundaries compared to the direct single anisotropic regularisator, when using deep matches.

The inpainting step of EpicFlow outperforms the proposed model on the Sintel dataset on all tests using both deep matches and CPM. In case of the Kitti 2015 dataset, the proposed method outperforms the inpainting step of EpicFlow using deep matches. However, using CPM EpicFlow gets more information about motion boundaries and therefore gives better results for both the inpainting step and the variational refinement.

The inpainting energy functional contains only the similarity term and the smoothness term. At pixels with a match, the indicator function $c(x, y)$ has the value 1, hence, there

will be no smoothing, making this functional very sensitive to outliers and occlusions. This energy functional could be further improved, e.g. by integrating a matching score ρ for the reliability of the matches, like in [BM11]. The isotropic and single anisotropic variants of the the direct approach were upgraded, however, the indirect approach and the combined approach were not improved, hence, this could be a starting point for new improvements.

In conclusion, the inpainting energy functional with the proposed smoothness terms shows improvements to the EpicFlow model only in cases where not enough information about motion boundaries can be obtained from the matches. However, in case of dense matching input, the EpicFlow model outperforms the proposed model.

Bibliography

[ÁELS99] L. Álvarez, J. Esclarín, M. Lefébure, J. Sánchez. "A PDE model for computing the optical flow." In: *Proceedings of the XVI Congreso de Ecuaciones Diferenciales y Aplicaciones* (Sept. 1999), pp. 1349–1356 (cit. on p. 31).

[Alh11] F. Alhwarin. "Fast and robust image feature matching methods for computer vision applications." PhD thesis. University of Bremen, 2011 (cit. on p. 27).

[BBM09] T. Brox, C. Bregler, J. Malik. "Large displacement optical flow." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 41–48 (cit. on pp. 37, 65).

[BBPW04] T. Brox, A. Bruhn, N. Papenberg, J. Weickert. "High accuracy optical flow estimation based on a theory for warping." In: *European Conference on Computer Vision (ECCV)* (2004), pp. 25–36 (cit. on pp. 21, 50).

[BDB13] J. Braux-Zin, R. Dupont, A. Bartoli. "A general dense image matching framework combining direct and feature-based costs." In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (2013), pp. 185–192 (cit. on p. 50).

[BE96] J. Barron, R. Eagleson. "Recursive estimation of time-varying motion and structure parameters." In: *Pattern Recognition* (1996), pp. 797–818 (cit. on p. 49).

[BM11] T. Brox, J. Malik. "Large displacement optical flow: descriptor matching in variational motion estimation." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.3 (2011), pp. 500–513 (cit. on p. 66).

[Bru06] A. Bruhn. "Variational optic flow computation." PhD thesis. Department of Mathematics and Computer Science, Saarland University, 2006 (cit. on pp. 14, 16, 25, 26, 29, 31, 32, 49).

[Bru15] A. Bruhn. *Lecture notes in Computer Vision*. 2015 (cit. on p. 11).

[Bru16] A. Bruhn. *Lecture notes in Correspondence Problems in Computer Vision*. 2016 (cit. on pp. 11, 16, 23, 24, 31, 49).

[BT05] L. Barron J., A. Thacker N. *Tutorial: Computing 2D and 3D optical flow*. Tutorial. Manchester, 2005. URL: http://www.tina-vision.net/docs/memos/2004-012.pdf (cit. on pp. 12, 49).

[BTS17] C. Bailer, B. Taetz, D. Stricker. "Optical Flow Fields: Dense Correspondence Fields for Highly Accurate Large Displacement Optical Flow Estimation." In: *CoRR* (2017) (cit. on pp. 7, 9).

[BWS05] A. Bruhn, J. Weickert, C. Schnörr. “Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods.” In: *International Journal of Computer Vision* (Feb. 2005), pp. 211–231 (cit. on p. 15).

[BWSB12] D. J. Butler, J. Wulff, G. B. Stanley, M. J. Black. “A naturalistic open source movie for optical flow evaluation.” In: *Proceedings of the European Conference on Computer Vision (ECCV)* (Oct. 2012), pp. 611–625 (cit. on p. 50).

[CBAB94] P. Charbonnier, L. Blanc-Feraud, G. Aubert, M. Barlaud. “Two deterministic half-quadratic regularization algorithms for computed imaging.” In: *Proceedings of the Image Processing, IEEE International Conference*. Vol. 2. IEEE. 1994, pp. 168–172 (cit. on p. 30).

[CBS00] G. Castellano, J. Boyce, M. Sandler. “Regularized CDWT optical flow applied to moving-target detection in IR imagery.” In: *Machine Vision and Applications* (May 2000), pp. 277–288 (cit. on p. 7).

[CMM00] T. Chan, A. Marquina, P. Mulet. “High-order total variation-based image restoration.” In: *SIAM J. Sci. Comput.* (Feb. 2000), pp. 503–516 (cit. on p. 34).

[DSV+14] O. Demetz, M. Stoll, S. Volz, J. Weickert, A. Bruhn. “Learning brightness transfer functions for the joint recovery of illumination changes and optical flow.” In: *Proceedings of the European Conference on Computer Vision (ECCV)* (2014), pp. 455–471 (cit. on p. 33).

[DZ13] P. Dollár, C. L. Zitnick. “Structured forests for fast edge detection.” In: *IEEE International Conference on Computer Vision (ICCV)*. 2013, pp. 1841–1848 (cit. on p. 39).

[Fis17] R. Fisher. *CVonline: The evolving, distributed, non-proprietary, on-line compendium of computer vision*. 2017. URL: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT12/node4.html (cit. on p. 13).

[Fle92] D. J. Fleet. *Measurement of image velocity*. Kluwer Academic Publishers, 1992 (cit. on p. 49).

[HLS17] Y. Hu, Y. Li, R. Song. “Robust interpolation of correspondences for large displacement optical flow.” In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017 (cit. on pp. 39, 40).

[HS81] B. K. Horn, B. G. Schunck. “Determining optical flow.” In: *Artificial Intelligence* 17.1 (1981), pp. 185–203 (cit. on pp. 7, 9, 11, 14, 15).

[HSW15] D. Hafner, C. Schroers, J. Weickert. “Introducing maximal anisotropy into second order coupling models.” In: *German Conference on Pattern Recognition* (Oct. 2015), pp. 79–90 (cit. on pp. 32, 34).

[LHS+17] Y. Li, Y. Hu, R. Song, P. Rao, Y. Wang. “Coarse-to-fine PatchMatch for Dense Correspondence.” In: *IEEE Transactions on Circuits and Systems for Video Technology* (June 2017) (cit. on p. 43).

[LK+81] B. D. Lucas, T. Kanade, et al. “An iterative image registration technique with an application to stereo vision (DARPA).” In: *Proceedings of the 1981 DARPA Image Understanding Workshop* (Apr. 1981), pp. 121–130 (cit. on p. 15).

[Low04] D. G. Lowe. “Distinctive image features from scale-invariant keypoints.” In: *International Journal of Computer Vision* 60.2 (Nov. 2004), pp. 91–110 (cit. on pp. 7, 27, 41).

[LS08] T. Liu, L. Shen. “Fluid flow and optical flow.” In: *Journal of Fluid Mechanics* 614 (2008), pp. 253–291 (cit. on p. 7).

[LZS13] M. Leordeanu, A. Zanfir, C. Sminchisescu. “Locally affine sparse-to-dense matching for motion and occlusion estimation.” In: *IEEE International Conference on Computer Vision (ICCV)*. 2013, pp. 1721–1728 (cit. on p. 20).

[MG15] M. Menze, A. Geiger. “Object scene flow for autonomous vehicles.” In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015 (cit. on pp. 44, 50–53, 55–57, 59–63).

[MSB17a] D. Maurer, M. Stoll, A. Bruhn. “Order-adaptive and illumination-aware variational optical flow refinement.” In: *Proceedings of the British Machine Vision Conference* (2017) (cit. on pp. 9, 26, 27, 38, 50, 55, 58, 61).

[MSB17b] D. Maurer, M. Stoll, A. Bruhn. “Order-Adaptive Regularisation for Variational Optical Flow: Global, Local and in Between.” In: *International Conference on Scale Space and Variational Methods in Computer Vision (SSVM)*. 2017, pp. 550–562 (cit. on p. 58).

[MSV+17] D. Maurer, M. Stoll, S. Volz, P. Gairing, A. Bruhn. “A comparison of isotropic and anisotropic second order regularisers for optical flow.” In: *Proceedings of the International Conference on Scale Space and Variational Methods in Computer Vision (SSVM)* (June 2017), pp. 537–549 (cit. on pp. 9, 33, 46, 65).

[PBB+06] N. Papenberg, A. Bruhn, T. Brox, S. Didas, J. Weickert. “Highly accurate optic flow computation with theoretically justified warping.” In: *International Journal of Computer Vision* 67.2 (Apr. 2006), pp. 141–158 (cit. on pp. 21, 22).

[PM90] P. Perona, J. Malik. “Scale-Space and Edge Detection Using Anisotropic Diffusion.” In: *IEEE Trans. Pattern Anal. Mach. Intell.* 12 (July 1990), pp. 629–639 (cit. on p. 30).

[RN95] S. J. Russell, P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 1995 (cit. on p. 8).

[RWHS15] J. Revaud, P. Weinzaepfel, Z. Harchaoui, C. Schmid. “EpicFlow: Edge-preserving interpolation of correspondences for optical flow.” In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015 (cit. on pp. 3, 9, 27, 38, 39, 42, 65).

[Sch94] C. Schnörr. “Unique reconstruction of piecewise-smooth images by minimizing strictly convex nonquadratic functionals.” In: *Journal of Mathematical Imaging and Vision* 4 (1994), pp. 189–198 (cit. on p. 31).

[UGVT88] S. Uras, F. Girosi, A. Verri, V. Torre. “A computational approach to motion perception.” In: *Biological Cybernetics* 60.2 (Dec. 1988), pp. 79–87 (cit. on p. 21).

[WB15] J. Wulff, M. J. Black. “Efficient sparse-to-dense optical flow estimation using a learned basis and layers.” In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 120–130 (cit. on pp. 27, 39).

[WC11] A. Wedel, D. Cremers. *Stereo scene flow for 3D motion analysis*. 1st. Springer Publishing Company, 2011 (cit. on pp. 20, 29).

[WRHS13] P. Weinzaepfel, J. Revaud, Z. Harchaoui, C. Schmid. “DeepFlow: Large displacement optical flow with deep matching.” In: *IEEE International Conference on Computer Vision (ICCV)*. Dec. 2013 (cit. on pp. 37, 42, 43).

[XCS+06] J. Xiao, H. Cheng, H. Sawhney, C. Rao, M. Isnardi. “Bilateral Filtering-Based Optical Flow Estimation with Occlusion Detection.” In: *Proceedings of the European Conference on Computer Vision (ECCV)* (2006), pp. 211–224 (cit. on p. 9).

[ZBW11] H. Zimmer, A. Bruhn, J. Weickert. “Optic flow in harmony.” In: *International Journal of Computer Vision* 93.3 (July 2011), pp. 368–388 (cit. on pp. 32, 45, 65).

All links were last followed on Januar 02, 2018.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

---

place, date, signature