Saarland University
Faculty of Natural Sciences and Technology I
Department of Computer Science
Master's Program in Computer Science

Master's Thesis

A Generic Framework For Smoothness Terms Of Arbitrary Order

submitted by

Alexander Hewer

on February 15, 2013

Supervisor

Prof. Dr. Joachim Weickert

Advisor

Prof. Dr. Joachim Weickert

Reviewers

Prof. Dr. Joachim Weickert

Prof. Dr. Andrés Bruhn

Statement

Hereby I confirm that this thesis is my own work and that I have documented all sources used.

Saarbrücken, February 15, 2013

Declaration of Consent

Herewith I agree that my thesis will be made available through the library of the Computer Science Department.

Saarbrücken, February 15, 2013

Acknowledgements

I want to thank the following people who made this work possible:

Prof. Joachim Weickert for accepting my proposal as the topic of my Master's Thesis.

Prof. Andrés Bruhn for again agreeing to become my second reviewer.

Dr. Stephan Didas for allowing me now for the third time to reuse the source code he provided me with during my Bachelor's thesis.

The staff of the LTM (Lehrstuhl für technische Mechanik) at Saarland University for allowing me to use their license of Vic-2D. In particular I want to thank Tobias Scheffer and Henning Seibert for their great support and very helpful feedback.

I also want to thank my friends and my family who supported me during my research.

Contents

So-called Variational Methods play a fundamental role in today's world of Image Processing and Computer Vision. They are used in various areas to solve arising problems. These Methods make use of a smoothness assumption on the solution they are trying to find. Here often first-order assumptions are formulated. However, these assumptions may fail completely in some special cases. Moreover we will realise that fixing the smoothness order to a specific value might not be the best idea as it limits the application area of the used method. Furthermore we will discover that often higher order information of the obtained solution like for example its derivatives may provide additional insight into the structure of the solution. Motivated by these observation we will try to design a new flexible-order smoothness term that can be adapted to the problem at hand by manipulating a single parameter. Additionally it is designed in such a way that derivatives of the solution are also estimated.

We will proceed as follows: In the first chapter we will discuss Variational Methods where we will focus on the modelling process. Here we will learn how to design a so-called Image Restoration approach and an Optic Flow approach. In both cases we will pay much attention on how a-priori knowledge about the expected solution influences the design process. After we completed the modelling we will in both cases try to solve a problem where the solution violates our a-priori knowledge that was true for another solution on purpose. In both cases the designed approach will not provide us with satisfactory results. We then continue in chapter 2 by designing a flexible-order smoothness term where the order of the smoothness assumption can be manipulated by a single parameter. We also try to model it in such a way that also estimates of the derivatives of the solution are computed. In chapter 3 we then deal with the discrete aspects before we perform experiments in chapter 4. In chapter 5 we will give a summary and discuss some additional research topics.

1 Introduction

In today's area of Visual Computing *Variational Methods* play a very important role. These methods are relatively easy to design and have therefore been successfully applied to various problems including

- Image Denoising (cf. [23])
- Motion Estimation (cf. [12], [8], [20])
- Analysis of Meteorological Data (cf. [5])
- Stereo Reconstruction (cf. [9])

Taking into account this widespread nature of such methods it is only natural to extend existing models in order to improve the quality of their solutions or to enlarge their field of application.

This is also the reason why we will aim at designing a new type of *Smoothness Term* in this work. However, before proceeding, it is on the one hand necessary to explain why such a new type is needed and how it may improve existing methods. On the other hand we still do not know what role a *Smoothness Term* plays in the design of Variational Methods.

To address these two questions we will give a short introduction to the design of such methods in this chapter. We will start off by investigating how Variational Methods that we consider are modelled in the general case. Here we will learn that the methods considered by us find the solution to a given problem by minimising an *Energy Functional* and that modelling this Functional in an appropriate way is the main part of the design process. After we treated the general case we will turn to two specific cases:

- A relatively simple case: Image Regularisation
- and a more complex example: Optic Flow Estimation

In both cases we will focus on the modelling process and in particular which role *a-priori* knowledge about the structure of the solution plays in this process. Furthermore we will consider the shortcomings and limitations of standard choices for the Smoothness Term. In addition to that we will also think of possible extensions to both methods.

1.1 General Case

When we encounter a problem that we want to solve we are usually provided with some data f that is related to this problem. Furthermore this data f is only defined on some domain. Let us call this domain Ω. In this work Ω has always the following properties:

- $\Omega \subset \mathbb{R}^2$
- Ω is open and connected
- Ω is bounded
- Ω is rectangular

Now let us turn to the co-domain I of f: In this work $I \subset \mathbb{R}$ is always bounded and depends on the problem at hand. Furthermore the data may also change over time. So we distinguish between the two cases:

- $f : \Omega \to I$
  In this case the data does not depend on the time. $f(x, y)$ refers to the value at the spatial coordinates $(x, y) \in \Omega$.
- $f : \Omega \times \mathbb{R}_0^+ \to I$
  This time the data depends also on a temporal component. Then $f(x, y, t)$ refers to the value at the spatial coordinates $(x, y) \in \Omega$ and time $t \in \mathbb{R}_0^+$.

As we only have access to the data f on our domain Ω we will only compute our solution $\mathbf{u}$ there as well. The co-domain of $\mathbf{u}$ again strongly depends on the problem we want to solve. For example u may be

- scalar-valued (Image Restoration)
- vector-valued (Optic Flow Estimation)
- matrix- or tensor-valued
- a set of tensor-valued functions (This will be the case in chapter 2)

We furthermore impose the following basic restrictions on the data f and the solution $\mathbf{u}$:

- $\mathbf{u}$ should be at least two times continuously differentiable on Ω and continuous on the boundary of Ω
- f should be at least continuous on Ω

For the sake of brevity we will often omit the parameters of functions. Thus we will write in the following f instead of $f(x, y)$ and $\mathbf{u}$ instead of $\mathbf{u}(x, y)$ if the meaning is clear from the context.

Now we can finally turn to question how we can obtain the solution $\mathbf{u}$ of our problem if we are provided with the data f. Variational Methods obtain this solution by minimising a given *Energy Functional*. This implies that the modelling process is now reduced to finding an appropriate Energy Functional

$$E_f(\mathbf{u}) = \int_\Omega \big(F\left(\mathbf{u}, f\right)\big)\ d\mathbf{x},\ \mathbf{x} \in \Omega$$

such that its minimiser represents the solution to the problem.

This modelling process is often split into two parts. In a first step a meaningful assumption is formulated on how the ideal solution should interact with the provided data. This assumption is also called the *Data Term* $D_f(\mathbf{u})$. Next we model an assumption about the structure of the solution itself. This is called the *Smoothness Term* $S(\mathbf{u})$. Here often a so-called first-order smoothness assumption is used, that is, an assumption on the first-order derivatives of $\mathbf{u}$. The Smoothness Term is usually weighted with a positive scalar α representing the importance of the term. A reason for a high weight might be for example that the data is not very reliable. Thus our Energy Functional has now the structure

$$E_f(\mathbf{u}) = \int_\Omega \Big(D_f(\mathbf{u}) + \alpha S(\mathbf{u})\Big)\ d\mathbf{x},\ \mathbf{x} \in \Omega$$

We, however, discover that the Data Term and the Smoothness Term may model contradicting ideas. This means that the solution $\mathbf{u}$ represents the best compromise between these assumptions.

Finally we give some information on the basic properties of the functionals E_f we will model in the following:

- E_f must be bounded from below
- The *Euler-Lagrange Equations* of E_f must exist. These equations are necessary conditions for a minimum. We will use them in the third chapter to find the minima of our functionals.
- E_f must be equipped with the appropriate natural boundary conditions. We remember that we are only working on the domain Ω and are computing derivatives. Roughly speaking these boundary conditions specify what happens to these derivatives on the boundary of Ω. They are called natural because they arise in the derivation of the Euler-Lagrange Equations (cf. [10], [23])
- E_f is continuous.

In the next two sections we will now deal with specific examples of such Energy Functionals. In both cases we will focus mainly on the aspect how the type of the problem and our a-priori knowledge about the solution influence the modelling process of both the Data Term and the Smoothness Term. Moreover we will discover that a-priori knowledge about one solution does not lead to a Smoothness Term that will perform well in all cases. In particular we will see that the usual choice of first-order smoothness assumptions might fail in special situations. We will also discuss possible extensions to both methods that may allow the simultaneous estimation of the solution and its derivatives.

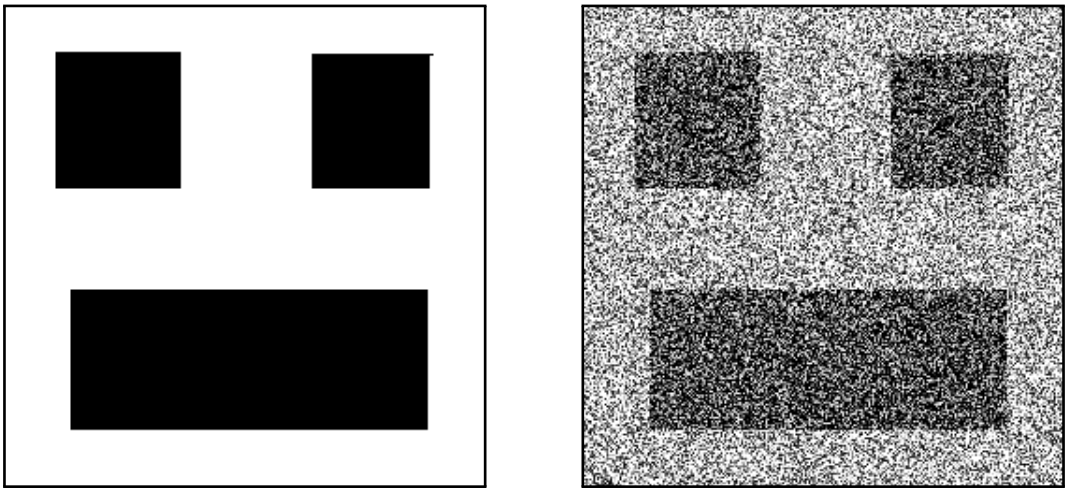


Figure 1.1: Original image on the left and noisy version on the right.

1.2 Specific Case: Image Restoration

In Image Processing we are often facing the problem that we are given an image f that is degraded due to noise like in figure 1.1. Of course we now want to find a way to denoise this image. That is we want to find an image u such that

- u is similar to f in some way
- u contains no noise

Before we are turning to the problem itself we first discuss the co-domains of f and u. In the following we are dealing with gray-value images where the gray values lie in the interval $I = [0, 255]$. Thus we have $f : \Omega \to [0, 255]$. As our solution u is an image itself we have that u has the same co-domain like f and thus $u : \Omega \to [0, 255]$.

Now let us solve this problem by using a Variational Approach. We remember that these methods obtain the solution by minimising a given energy. So we have to model this Energy Functional in such a way that a minimiser fulfills our desired properties we gave above.

1.2.1 Modelling the Data Term

First we want to model the similarity between u and f, that is the desired interaction of the data and the solution. This similarity could be expressed as follows

$$u - f$$

We, however, realise that this expression is not bounded from below. So there is no way of finding a reasonable minimiser. Taking now the absolute value of the expression gives

$$|u - f|$$

This expression is now bounded from below but suffers from the problem that it is now no longer differentiable with respect to u everywhere, which unfortunately leads to a Euler-Lagrange Equation that is numerically hard to handle. Squaring the expression finally leads to our final Data Term

$$D_f^{\mathrm{I}} := (u - f)^2$$

which is now bounded from below and differentiable everywhere. This finishes the modelling process of the Data Term of our functional. In this case the modelling of this term was relatively easy, we will see later on in the second case that the design of the Data Term can also pose quite a challenge.

1.2.2 Modelling the Smoothness Term

Now that we have our Data Term we still have to model our Smoothness Term, that is, our assumption on the structure of our solution u. To this end we will consider the second property we stated earlier, namely that u should contain no noise. If we take a look at the noisy image in figure 1.1 we find that the noise creates first-order derivatives that are non-zero. On the other hand we see that the original picture has large areas where the gray values remain constant in space, so there we have that the first-order derivatives vanish. This observation motivates the idea to assume that our solution has this property as well:

$$\|\nabla u\|^2 = \left\| \begin{pmatrix} u_x \\ u_y \end{pmatrix} \right\|^2 = u_x^2 + u_y^2 \approx 0$$

This represents our Smoothness Term. As it demands the same smoothness in every connection, we will call it a *homogeneous isotropic* smoothness assumption.

In this case the contractive nature of Data Term and Smoothness Term becomes apparent: The Data Term wants to preserve the noise whereas the Smoothness Term tries to remove it.

Weighting now our Smoothness Term with a positive scalar α and combining it with our previously derived Data Term D_f^{I} yields our first Energy Functional:

$$E_f(u) = \int_\Omega \underbrace{\left(\underbrace{(u - f)^2}_{\text{Data Term}} + \alpha \underbrace{\|\nabla u\|^2}_{\text{Smoothness Term}} \right)}_{=F} d\mathbf{x},\ \mathbf{x} \in \Omega$$

We see that the integrand F depends on u and ∇u, that is,

$$F(u, u_x, u_y) = (u - f)^2 + \alpha \|\nabla u\|^2$$

With this knowledge we can derive the Euler-Lagrange Equation of E_f and the associated boundary conditions:[1]

$$0 = F_u - \frac{\partial}{\partial x} F_{u_x} - \frac{\partial}{\partial y} F_{u_y}$$

with the Boundary Condition

$$\mathbf{n}^\top \left(\begin{pmatrix} F_{u_x} \\ F_{u_y} \end{pmatrix} \right) = 0$$

where $\mathbf{n}$ is the outer surface normal on Ω.

In our case we have then the equation

$$0 = u - f - \alpha \text{ div } \Big(D \ \nabla u \Big)$$

where

$$D = I := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

is the so-called *Diffusion Tensor* (cf. [23]). In a general setting it represents a symmetric matrix that is at least positive-semidefinite. The Boundary Condition reads

$$\mathbf{n}^\top \begin{pmatrix} u_x \\ u_y \end{pmatrix} = 0$$

Formally these condition states that the derivatives of u should vanish across the boundary. This makes sense because we can not compute them there because we have no information what happens outside of our domain. Informally we learn that these condition implies that the Smoothness Term is automatically fulfilled across the boundary.

[1] Here we see why it is important that the Data Term should be differentiable everywhere with respect to u.

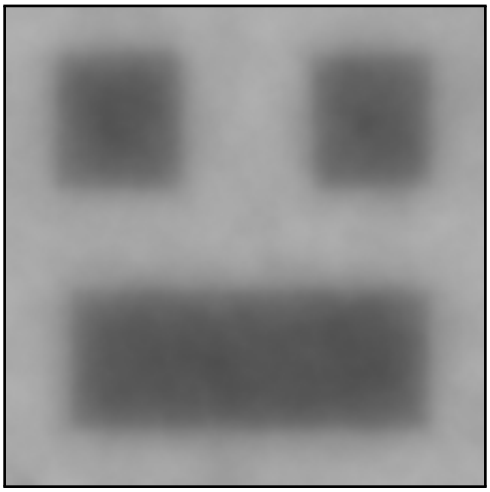


Figure 1.2: Our first model with $\alpha = 100$ removes the noise but also important image features like edge information.

Using this ansatz to denoise our given image leads to a result that is not quite satisfactory: We see in figure 1.2 that on the one hand the noise is removed but on the other hand also important image features like object edges are destroyed. Of course we are now interested in extending our method to preserve these important information. This means we have to alter our ideas about the solution:

- u is similar to f in some way
- u contains no noise but still object edges

Intuitively we realise that this implies that we have to modify the Smoothness Term in such a way that it preserves these image edges, that is, it should assume that the solution is *piecewise* constant.

Until now we used a quadratic penaliser for our smoothness assumption:

$$\Psi_{\mathrm{WT}}(\|\nabla u\|^2) := \|\nabla u\|^2$$

In literature this penaliser is often called the *Whittaker-Tikhonov* penaliser (cf. [23]). It always penalises derivations of our assumption very severely, which leads to a huge energy generation. Thus is becomes very attractive to fulfill the Smoothness Term as then a lot energy can be "saved". There are, however, other choices for the penaliser. Each function $\Psi(s^2)$ with the following properties is a suitable candidate (cf. [23]):

- Ψ_S is differentiable and increasing: $\Psi'_S\left(s^2\right) > 0$,
- $\Psi_S\left(s^2\right)$ is convex in s and
- There exist constants $c_1, c_2 > 0$ such that $c_1 s^2 \leq \Psi_S\left(s^2\right) \leq c_2 s^2$ for all s^2.

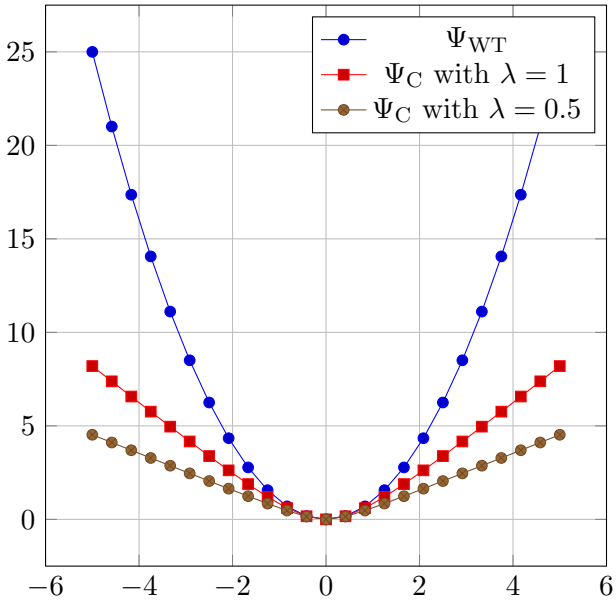


Figure 1.3: Plots of Ψ_{WT} and of Ψ_{C} with different values for λ.

Our task is now to find one that can make it unattractive to fulfill the Smoothness Term in some cases such that the edges are preserved. Let us to this end have a look at the famous *Charbonnier* penaliser (cf. [23]):

$$\Psi_{\mathrm{C}}(s^2) := 2\lambda^2\sqrt{1+\frac{s^2}{\lambda^2}} - 2\lambda^2$$

where $\lambda > 0$ is the so-called *contrast parameter*. Inspecting the energy generation for different choices of λ in figure 1.3 reveals that this penaliser meets our requirements: It produces less energy in the case of a violation of the smoothness constraint than the Whittaker-Tikhonov penaliser. Moreover these observations imply that a fulfillment will lead to a smaller "energy saving", which in turn could make it more attractive to fulfill the Data Term instead. As we know that edge information is stored in the data we conclude that using Ψ_C will make it more attractive to preserve edges.

Using now Ψ_C as the penaliser yields our new Smoothness Term:

$$\Psi_{\mathrm{C}}(\|\nabla u\|^2)$$

As this term demands a piecewise smoothness in each direction, we call this term a piecewise isotropic smoothness assumption. Our new energy is now given by:

$$E_f(u) = \int_\Omega \left((u-f)^2 + \alpha\ \Psi_C(\|\nabla u\|^2) \right) d\mathbf{x},\ \mathbf{x} \in \Omega$$

Employing Ψ_C also leads to a change in our Euler-Lagrange Equation that now becomes nonlinear in u:

$$0 = u - f - \alpha \operatorname{div}\left(D_{\Psi_{\mathrm{C}}}(u) \nabla u \right)$$

where we have the new Diffusion Tensor

$$D_{\Psi_C}(u) := \begin{pmatrix} \Psi'_C(\|\nabla u|^2) & 0 \\ 0 & \Psi'_C(\|\nabla u\|^2) \end{pmatrix}$$

We remark that we can also provide a *generic* version of our two Smoothness Terms and the associated Diffusion Tensors:

$$\begin{aligned} S^{\mathrm{I}}_{\Psi}(u) &:= \Psi(\|\nabla u\|^2) \\ D^{\mathrm{I}}_{\Psi}(u) &:= \Psi'(\|\nabla u\|^2)\ I \end{aligned}$$

that now depend on the choice of Ψ:

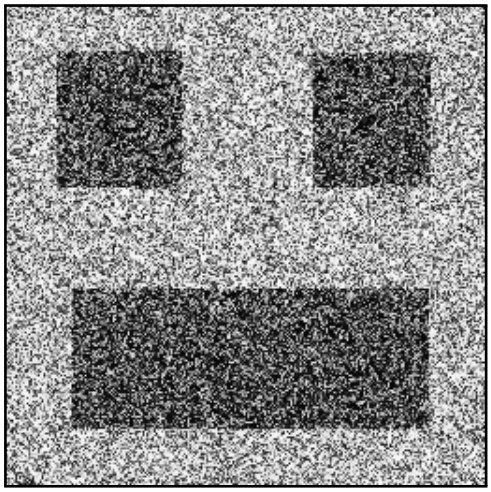
- Ψ_{WT} leads to our homogeneous isotropic Smoothness Term:

$$\begin{aligned} S^{\mathrm{I}}_{\Psi_{\mathrm{WT}}}(u) &= \|\nabla u\|^2 \\ D^{\mathrm{I}}_{\Psi_{\mathrm{WT}}}(u) &= \Psi'_{\mathrm{WT}}(\|\nabla u\|^2)\ I = I \end{aligned}$$

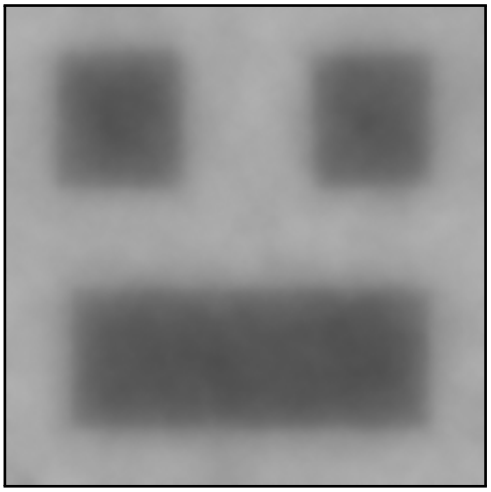
- Ψ_{C} leads to our piecewise isotropic Smoothness Term:

$$\begin{aligned} S^{\mathrm{I}}_{\Psi_{\mathrm{C}}}(u) &= \Psi_{\mathrm{C}}(\|\nabla u\|^2) \\ D^{\mathrm{I}}_{\Psi_{\mathrm{C}}}(u) &= \Psi'_{\mathrm{C}}(\|\nabla u\|^2)\ I \end{aligned}$$

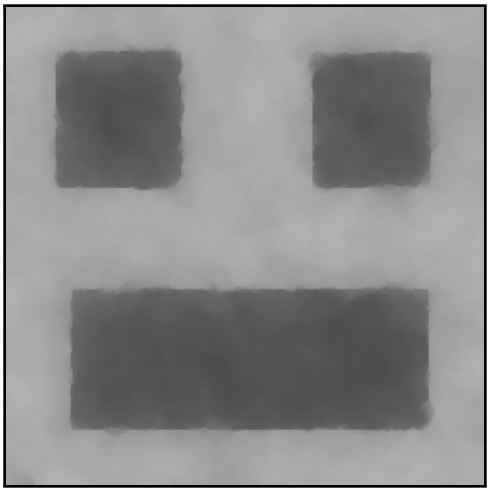
By inspecting figure 1.4 we realise that this new method delivers satisfactory results: The noise is removed while object edges are presered. We discover, however, that the choice of λ is not trivial: On the one hand choosing a too small value will cause our method to preserve noise as well because it misinterpretes it as edge information. On the other hand a too large value will again lead to the destruction of vital edge information. Thus it has to be chosen in such a way that in regions of noise it is attractive to fulfill the Smoothness Term whereas at object edges the Data Term should be fulfilled instead.



(a) $\lambda = 0.1$



(b) $\lambda = 10$



(c) $\lambda = 2$

Figure 1.4: Results of our new model with $\alpha = 100$ and different choices for λ.

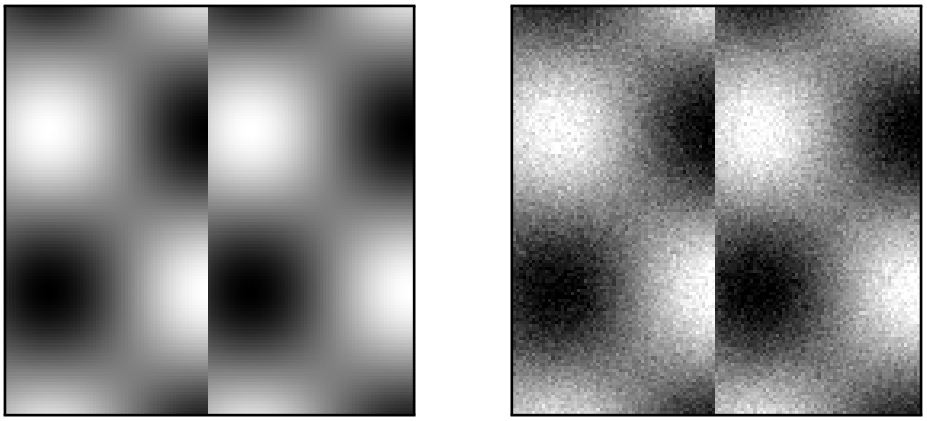


Figure 1.5: Original image on the left and noisy version on the right. (source: [6])

1.2.3 Limitations and possible extensions

In the previous section we modelled our Smoothness Terms by exploiting a-priori knowledge about our original image, namely that it is constant or at least piecewise constant. We now want to find out if our Smoothness Terms still perform well if the considered image does not share this property. This actually means that the original image should violate our smoothness assumption by construction. To this end we use the image depicted in figure 1.5. We see that this time the original image is no longer piecewise constant: We have a large discontnuity in the centre that separates two regions where the data is continuously changing in space.

In a first try we use our method with our homogeneous isotropic Smoothness Term $S^{\mathrm{I}}_{\Psi_{\mathrm{WT}}}$ in order to restore the original image from the noisy data. We see in figure 1.6 that the non-constant behaviour of the data is preserved in the two regions. Unfortunately we also discover that information about the discontinuity is lost in the process.

Next we apply our method that uses the Charbonnier penaliser $S^{\mathrm{I}}_{\Psi_{\mathrm{C}}}$ to denoise the data. We in see in figure 1.6 that this time we succeed in preserving the discontinuity in the centre. But we fail, however, to preserve the non-constant behaviour in the two regions. Instead we enforced the image to be piecewise constant in these regions.

We realise that these bad results can be explained by taking into account that our smoothness assumption was incorrect from the start. Of course we now want to know how we can adjust our model to cope with such images as well. One possibility would be to replace the first-order smoothness constraint by a second-order one, that is, we would assume that the second-order derivatives vanish, which would explicitly allow the non-constant behaviour of the data. But this means we are again exploiting knowledge about the ideal solution to design a new smoothness assumption. We learned, however, that using such a *fixed*-order assumption that worked for one image may fail for another one if they do not share the same properties. As a remedy we might think of designing a *flexible*-order Smoothness Term. In this case it would be possible to adapt the Smoothness Term to the problem at hand instead of relying on a fixed-order one.

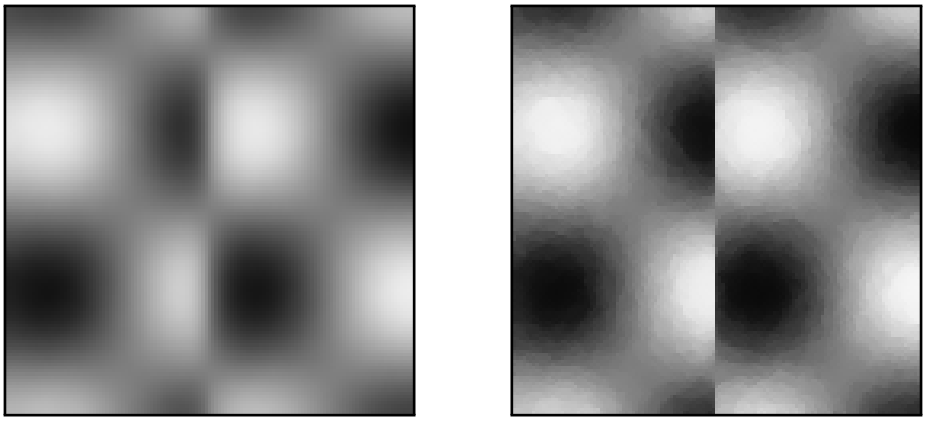


Figure 1.6: Our attempts to denoise the data. **Left:** Result with Ψ_{WT}, **Right:** Result with Ψ_{C}

In addition to this identified shortcoming of a fixed-order smoothness assumption we find that sometimes derivatives of an image can also be of great importance in Image Processing. For example it is possible to use the absolute value of the gradient $\|\nabla f\|$ of the image f to identify object edges (cf. [24]).

Taking into account these two observations motivates the idea to design a method that is capable of simultaneously denoising given data with a flexible-order smoothness assumption and estimating derivatives that are related to the denoised image. However, before we begin working on this idea we will first investigate if such a method might be also required in another field like *Optic Flow Estimation*, which we want to explore next.

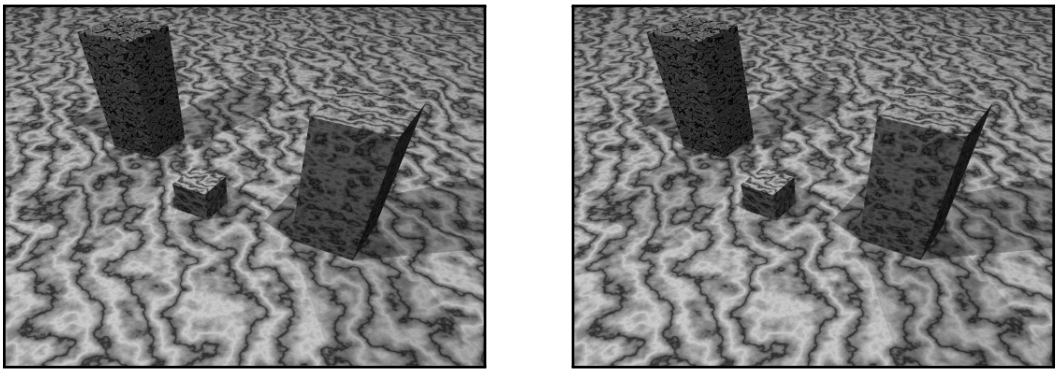


Figure 1.7: Frames 150 and 151 of the *new marble sequence*. (source: [1]).

1.3 Specific Case: Optic Flow Estimation

A different branch of Image Processing is concerned with motion estimation. More precisely, we are given a pair of images of an image sequence f like in figure 1.7. The given sequence describes a movement of some objects that we want to estimate. In the following we are interested in finding the so-called *Displacement Field* $\mathbf{u}$ such that

- $\mathbf{x} + \mathbf{u}$ is the new position of an object in the second image that was previously located at $\mathbf{x}$ in the first image where $\mathbf{x} := \begin{pmatrix} x \\ y \end{pmatrix}$
- we have an estimate of $\mathbf{u}$ everywhere, even in regions where the data is unreliable

Like in the previous case we will first briefly discuss the properties of our data f and the solution $\mathbf{u}$ before turning to the problem itself. Let f be a gray-value image sequence with 256 gray values that starts at time 0. So we have $f : \Omega \times \mathbb{R}_0^+ \to [0, 255]$. We write $f(x, y, t)$ for the gray-value at time $t \in \mathbb{R}_0^+$ and spatial coordinates $(x, y) \in \Omega$. Unlike in the previous case we now have more restrictive requirements on f: f must be two times continuously differentiable in $\Omega \times \mathbb{R}_0^+$.[2] In the following we will refer to the first image as $f(x, y, t)$ and to the second one as $f(x, y, t + 1)$ where t is fixed. Let us now proceed with the properties of $\mathbf{u}$. As we require $\mathbf{u}(x, y)$ to be a two-dimensional displacement vector at the coordinates $(x, y) \in \Omega$ its co-domain should lie in $\mathbb{R}^2$. So we have

$$\mathbf{u} := \begin{pmatrix} u \\ v \end{pmatrix} : \Omega \to I$$

where $I \subset \mathbb{R}^2$ is bounded. Again we require the solution $\mathbf{u}$ to be two times continuously differentiable in Ω and to be continuous on the boundary of Ω.

As our solution is two-dimensional we first have to think of a way to visualise its contained information in a meaningful way. Here we will employ two variants:

[2] As images might violate this constraint it is common to convolve the initial data with a *Gaussian Kernel* to make them differentiable and to work with this smoothed data instead.

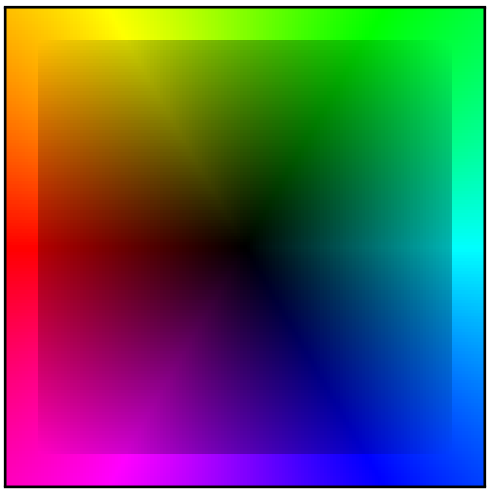


Figure 1.8: Example plot of a two-dimensional vector field.

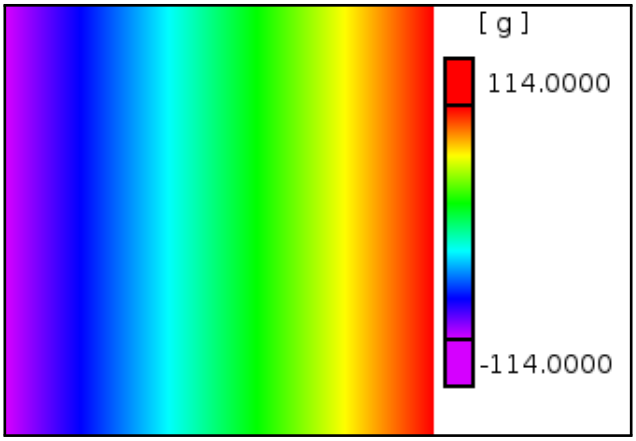


Figure 1.9: Visualisation of some scalar-valued function g.

1. We will visualise the information in a single plot where the direction of the displacement vector is represented by the color and its magnitude is shown by the brightness of this color. Furthermore the plot is surrounded by a frame that roughly shows the directions the colors represent. An example plot is provided in figure 1.8.

2. In the second variant we will visualise the information of u and v in two separate plots where we will visualise the magnitude of their values. In addition to that we will also provide a scale where the minimum and maximum value can be inspected. Again we provide an example plot, this time in figure 1.9.

Again we are now facing the problem how to model the Energy Functional in such a way that its minimiser has the desired properties we stated earlier. Once more we will focus on how a-priori knowledge influences our design process of both the Data Term and the Smoothness Term.

1.3.1 Modelling the Data Term

We start with our first requirement: $\mathbf{x} + \mathbf{u}$ should be the new position of an object at time $t + 1$ that was originally located at $\mathbf{x}$ at time t. Unfortunately we discover that it is now no longer possible to formulate this desired property in an explicit way. By inspecting our image pair in figure 1.7, however, we realise that spatial image features like for example the gray value do not change over time. This means that the following expression holds if $\mathbf{u}$ represents the correct displacement:

$$f(x, y, t) = f(x + u, y + v, t + 1)$$

We can exploit this observation to construct our first Data Term, the so-called *Brightness Constancy Assumption* (cf. [9]):

$$(f(x + u, y + v, t + 1) - f(x, y, t))^2$$

This term is obviously minimised by the correct $\mathbf{u}$ if the gray value of objects remains constant over time. But we identify a problem with this term: Its dependence on the data might result in a non-convex behaviour, which would also lead to a non-convex Energy Functional. As this type of functional has the unpleasant property of having multiple local minima finding a good minimum is a very demanding task (cf. [9]). We will see later on that the so-called *Warping Strategy* (cf. [9], [15]) can be used to obtain at least a good minimiser. For the time being, however, we want to construct a Data Term that is at least convex. To this end we assume that the displacement $\mathbf{u}$ is sufficiently small such that we can approximate our term by using a first order Taylor Expansion:

$$\begin{aligned}
&\Big(f(x + u, y + v, t + 1) - f(x, y, t)\Big)^2 \\
&\quad \approx \Big(f(x, y, t) + u f_x(x, y, t) + v f_y(x, y, t) + f_t - f(x, y, t)\Big)^2 = \Big(u f_x + v f_y + f_t\Big)^2
\end{aligned}$$

Let us now inspect the *Hessian Matrix* of this term:

$$2 \begin{pmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{pmatrix} = 2 \nabla_2 f \nabla_2 f^\top$$

where the *indexed* ∇-Operator signals that only the derivatives with respect to the first two (spatial) components should be contained in the result vector. Thus in our case we have

$$\nabla_2 f = \begin{pmatrix} f_x \\ f_y \end{pmatrix}.$$

We will call this vector the *spatial gradient* of f.

The matrix $\nabla_2 f \nabla_2 f^\top$ is called the *Structure Tensor* [3] of f (cf. [23]). It has the eigenvectors

$$v_1 = \frac{\nabla_2 f}{\|\nabla_2 f\|^2}$$
$$v_2 = \frac{\nabla_2 f^\perp}{\|\nabla_2 f^\perp\|^2}$$

where $\nabla_2 f^\perp$ is the vector orhogonal to $\nabla_2 f$.

The associated eigenvalues are given by

$$\mu_1 = \|\nabla f_2\|^2$$
$$\mu_2 = 0$$

Thus the matrix is positive-semidefinite, which in turn causes the Hessian matrix to become positive-semdidefinite and thus means that our Data Term is now convex.

This kind of linearised data assumptions can also be conveniently rewritten in a quadratic form, the so-called *Motion Tensor Notation* (cf. [9], [8]):

$$w^\top J w$$

where

$$w := \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

and

$$J := \begin{pmatrix} J_{11} & J_{12} & J_{13} \\ J_{12} & J_{22} & J_{23} \\ J_{13} & J_{23} & J_{33} \end{pmatrix}$$

[3] We will later on discuss the meaning of the Structure Tensor in more detail.

is a positive-semidefinite symmetric matrix, the so-called *Motion Tensor* associated to the linearised constancy assumption.

In our case we will call the Motion Tensor associated to the linearised Brightness Constancy Assumption J_{BCA}. It is given by

$$J_{\text{BCA}} := \begin{pmatrix} f_x^2 & f_x f_y & f_x f_t \\ f_x f_y & f_y^2 & f_y f_t \\ f_x f_t & f_y f_t & f_t^2 \end{pmatrix}$$

Unfortunately we identify some problems with our Data Term:
Basically we want to solve

$$u f_x + v f_y + f_t = 0$$

That is, we have one equation for two unknowns, which has infinitely many solutions. Let us now compare this equation to an ordinary line equation

$$\mathbf{n}^\top (\mathbf{x} - \mathbf{y}) = \mathbf{n}^\top \mathbf{x} - \mathbf{n}^\top \mathbf{y} = 0$$

where $\mathbf{n}$ is the line normal and $\mathbf{y}$ is a given point on the line. If we perform the following substitutions

$$\begin{aligned} \mathbf{n} &:= \nabla_2 f \\ \mathbf{x} &:= \mathbf{u} \\ -\mathbf{n}^\top \mathbf{y} &:= f_t \end{aligned}$$

we find that our Data Term is in fact a squared line equation:

$$(\nabla_2 f^\top \mathbf{u} + f_t)^2 = 0$$

Informally this means that our solution consists of a line that is normal to the spatial image gradient $\nabla_2 f$. In addition to that we discover that we can compute nothing at all if $\nabla_2 f$ vanishes. This problem is known in literature as the infamous *Aperture Problem* (cf. [9]).

Moreover we also realise that if the gray-value changes, if the data is corrupted due to noise or if objects move outside our visible domain the Data Term will represent no reliable tool to estimate the motion. We say in these cases that our data is *not reliable*.

Let us now focus on the case of the violation of the Brightness Constancy Assumption. If this violation appeared due to additive illumination changes we can cope with this problem by employing the well-known Gradient Constancy Assumption (cf. [15]). This assumption models the idea that the gradient of the image data does not change over time. It is given by

$$\|\nabla_2 f(x + u, y + v, t + 1) - \nabla_2 f(x, y, t)\|^2$$

Again we perform a first-order Taylor Expansion to obtain the linearised version:

$$\begin{aligned}\|\nabla_2 f(x+u,y+v,t+1) - \nabla_2 f(x,y,t)\|^2 = \\ \Big(f_x(x+u,y+v,t+1) - f_x(x,y,t)\Big)^2 + \Big(f_y(x+u,y+v,t+1) - f_y(x,y,t)\Big)^2 \\ \approx \Big(uf_{xx} + vf_{xy} + f_{xt}\Big)^2 + \Big(uf_{xy} + vf_{yy} + f_{yt}\Big)^2\end{aligned}$$

Here we finally see why we demanded that the data should be two times continuously differentiable. Once again we deal with the question if this term is convex. Computing the Hessian Matrix yields

$$2\begin{pmatrix} f_{xx}^2 & f_{xx}f_{xy} \\ f_{xx}f_{xy} & f_{yy}^2 \end{pmatrix} + 2\begin{pmatrix} f_{xy}^2 & f_{xy}f_{yy} \\ f_{xy}f_{yy} & f_{yy}^2 \end{pmatrix} = 2\Big(\nabla_2 f_x \nabla_2 f_x^\top + \nabla_2 f_y \nabla_2 f_y^\top\Big)$$

This is a sum of two Structure Tensors. As both matrices are positive-semidefinite the sum will also be at least positive-semidefinite.[4] Thus we can conclude that this term is in fact at least convex.

The Motion Tensor associated to the linearised Gradient Constancy Assumption is given by

$$J_{\text{GCA}} := \begin{pmatrix} f_{xx}f_{xx} + f_{xy}f_{xy} & f_{xy}f_{xx} + f_{yy}f_{xy} & f_{xt}f_{xx} + f_{yt}f_{yx} \\ f_{xy}f_{xx} + f_{yy}f_{xy} & f_{xy}f_{xy} + f_{yy}f_{yy} & f_{xt}f_{xy} + f_{yt}f_{yy} \\ f_{xt}f_{xx} + f_{yt}f_{yx} & f_{xt}f_{xy} + f_{yt}f_{yy} & f_{xt}f_{xt} + f_{yt}f_{yt} \end{pmatrix}$$

We are now interested in the question if this term also suffers from the Aperture Problem. By inspecting our new constraint

$$\underbrace{(uf_{xx} + vf_{xy} + f_{xt})^2}_{=0} + \underbrace{(uf_{xy} + vf_{yy} + f_{yt})^2}_{=0} = 0$$

we see that it can be expressed in the form of two line equations:

$$\begin{aligned}\nabla_2 f_x^\top \mathbf{u} + f_{xt} = 0 \\ \nabla_2 f_y^\top \mathbf{u} + f_{yt} = 0\end{aligned}$$

Intuitively we now discover that if those lines are not parallel they will intersect and this intersection point will be our unique solution. However, we also realise that we can not rely on our data to always produce such lines. Furthermore a line equation may also not exist if the derivatives are vanishing.

Finally we can exploit the Motion Tensor Notation to merge both the Brightness Constancy Assumption and the Gradient Constancy Assumption into a single Motion Tensor:

$$J_{BGA} := w^\top \Big(J_{BCA} + \gamma J_{GCA}\Big) w$$

[4] It is also possible for the result to become positive-definite.

This new Data Term models the idea that both the gray-value and the spatial image gradient do not change over time. We furthermore weighted the Gradient Constancy Assumption with a nonnegative scalar γ in order to adjust the importance of the term. Ultimately this new constraint will provide three line equations. We can, however, again not rely on these lines to be not parallel or even to exist.

The Gradient and Brightness Constancy Assumptions are actually only two of many assumptions on spatial image features. The interested reader is encouraged to have a look at [15] for an overview. For now we stop working on the Data Term and proceed with the Smoothness Term in order to deal with the Aperture Problem.

1.3.2 Modelling the Smoothness Term

In the preceding section we discovered that our Data Term can not be used to fulfill our second requirement we have on the solution, namely to estimate the motion everywhere. We learned that our Data Term will provide us with three line equations that only the correct displacement vector $\mathbf{u}$ will fulfill. But only in the case if two of those described lines are not parallel we have access to a unique solution. In the other cases our solution lies either on a line or we have no information at all. So we conclude that our *local* information is not sufficient in these cases to determine the unique solution.

Thus we have to think about a strategy to *transport* information from the neighborhood to our current position and merge it locally to estimate our solution.

Again we take a close look at our data in figure 1.7: We see that our observed motion consists entirely of a *translation*. The key observation is now that a translational motion is a motion that remains constant in space. That is we have

$$\|\nabla u\| \approx 0$$
$$\|\nabla v\| \approx 0$$

We now square these both terms [5] and compute their sum to obtain our first Smoothness Term:

$$\|\nabla u\|^2 + \|\nabla v\|^2$$

Clearly this term is minimised if u and v are constant, that is, describing at most a translational motion. Again we are demanding a solution that is smooth in each direction, thus we have a homogeneous isotropic smoothness assumption.

Let us now combine our Data Term and Smoothness Term to obtain our first Energy Functional for this problem

$$E(\mathbf{u}) = \int_\Omega \underbrace{\left(\mathbf{w}^\top \left(J_{BCA} + \gamma J_{\mathrm{GCA}}\right)\mathbf{w} + \alpha \ \left(\|\nabla u\|^2 + \|\nabla v\|^2\right)\right)}_{=F} \ d\mathbf{x}$$

where we again weighted the Smoothness Term by a positive scalar α.

[5] Again to ensure differentiability everywhere.

We see that the usage of our Smoothness Term has the desired effect: It links the local solution to its neighborhood. This means that if the local data can not provide enough information to determine the unique solution the Smoothness Term will *fill in* this missing information. In literature this is called the *filling-in* effect (cf. [9]). Note that this is only a very rough description of the interaction between the Data Term and Smoothness Term, for a more in-depth treatment see [22] or [9].

We see that this time our integrand F depends on $u, v, \nabla u, \nabla v$, that is, we have

$$F(u, v, u_x, u_y, v_x, v_y) = \mathbf{w}^\top \Big(J_{BCA} + \gamma J_{\text{GCA}} \Big) \mathbf{w} + \alpha \ \Big(\|\nabla u\|^2 + \|\nabla v\|^2 \Big)$$

With this information we are again able to compute the general structure of the Euler-Lagrange Equations:

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

with the associated Boundary Conditions

$$\begin{aligned} \mathbf{n}^\top \begin{pmatrix} F_{u_x} \\ F_{u_y} \end{pmatrix} &= 0 \\ \mathbf{n}^\top \begin{pmatrix} F_{v_x} \\ F_{v_y} \end{pmatrix} &= 0 \end{aligned}$$

where $\mathbf{n}$ is again the outer surface normal on the boundary of Ω.

In our case we have the equations

$$\begin{aligned} 0 &= J_{11}\ u + J_{12}\ v + J_{13} - \alpha\ \text{div}(D\ \nabla u) \\ 0 &= J_{12}\ u + J_{22}\ v + J_{23} - \alpha\ \text{div}(D\ \nabla v) \end{aligned}$$

where

$$D = I$$

is again the Diffusion Tensor originating from the Smoothness Term. The Boundary Conditions are given by

$$\begin{aligned} \mathbf{n}^\top \nabla u &= 0 \\ \mathbf{n}^\top \nabla v &= 0 \end{aligned}$$

Unfortunately we discover in figure 1.10 that our approach is suffering from a similar problem as before: The motion seems to be estimated but motion discontinuities are not preserved. Again we are interested in adjusting our model to allow these discontinuities to be present.

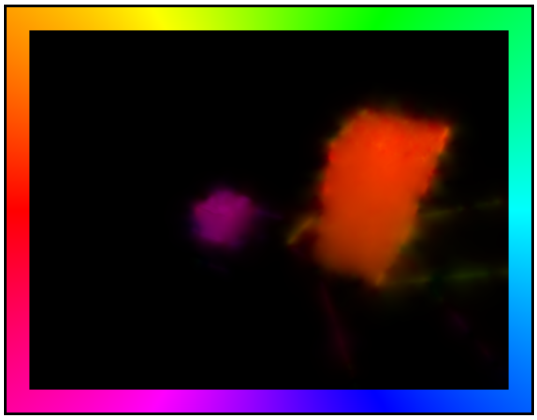


Figure 1.10: Result from our first approach. We see that motion discontinuities are not preserved.

To this end we inspect our image pair: Here we discover that the motion is piecewise translational and in addition to that may exhibit an *anisotropic* behaviour: Along the object edges the motion remains constant whereas across the edges it shows discontinuities. These observations imply that we have to model an anisotropic smoothness assumption.

Let us now investigate how we can formulate these observations in a mathematical way: First we consider the Structure Tensor of the displacement part u:

$$\nabla u \nabla u^\top$$

We remember that this matrix possesses the eigenvalues

$$\begin{aligned} \mu_1 &= \|\nabla u\|^2 \\ \mu_2 &= 0 \end{aligned}$$

with the associated eigenvectors

$$\begin{aligned} v_1 &= \frac{\nabla u}{\|\nabla u\|^2} \\ v_2 &= \frac{\nabla u^\perp}{\|\nabla u^\perp\|^2} \end{aligned}$$

We discover that this tensor describes the behaviour of u locally: The directions of the changes are contained in the eigenvectors of the matrix whereas the magnitude of the changes is located in the associated eigenvalues. Let us now combine the Structure Tensors of u and v:

$$\nabla u \nabla u^\top + \nabla v \nabla v^\top = \begin{pmatrix} u_x^2 + v_x^2 & u_x u_y + v_x v_y \\ u_x u_y + v_x v_y & u_y^2 + v_y^2 \end{pmatrix}$$

The result is the Structure Tensor of $\mathbf{u}$ that describes the behaviour of our Displacement Field $\mathbf{u}$ locally. It is at least positive-semidefinite as it describes the sum of positive-semidefinite matrices. By computing its trace

$$\operatorname{tr}\begin{pmatrix} u_x^2 + v_x^2 & u_x u_y + v_x v_y \\ u_x u_y + v_x v_y & u_y^2 + v_y^2 \end{pmatrix} = \|\nabla u\|^2 + \|\nabla v\|^2$$

we surprisingly obtain our previous Smoothness Term. Thus we modelled previously that the sum of the eigenvalues[6] of the combined Structure Tensors should vanish. As our eigenvalues are nonnegative this means we assumed that both eigenvalues should vanish, which actually would imply an isotropic behaviour of $\mathbf{u}$: namely that it is translational in every direction, which in turn was our modelling idea.

So in order to model an anisotropic behaviour we have to ensure that the sum of the eigenvalues is allowed to be non-zero and that both eigenvalues may be different[7] To this end we use the Smoothness Term presented in [18]:

$$\operatorname{tr}\ \Psi_{\mathrm{C}}\left(\nabla u \nabla u^\top + \nabla v \nabla v^\top\right)$$

Here the Charbonnier Penaliser Ψ_{C} is applied to the eigenvalues of the Structure Tensor before the trace is computed. Like in the previous case we observe that if we choose λ accordingly it may become unattractive to enforce the sum of the eigenvalues to become 0 in some situations. In these situations it may be better to fulfill the Data Term instead as there again more energy can be "saved". As information about motion discontinuities is contained inside the data we conclude that this new term provides the property of allowing these discontinuities in the solution. Furthermore we note that the solution may also behave anisotropically because both eigenvalues are penalised separately: There may be for example cases when it is attractive to enforce one of the eigenvalues to become 0 whereas it is unattractive to do the same for the other one.

As this Smoothness Term demands a smoothness that is piecewise and also may behave differently in each directions we call it a piecewise aninsotropic smoothness assumption.

Using this new term in our Energy Functional yields

$$E(\mathbf{u}) = \int_\Omega \left(\mathbf{w}^\top \left(J_{BCA} + \gamma J_{\mathrm{GCA}}\right)\mathbf{w} + \alpha\ \operatorname{tr}\ \Psi_{\mathrm{C}}\left(\nabla u \nabla u^\top + \nabla v \nabla v^\top\right)\right) d\mathbf{x}$$

This functional leads to the following Euler-Lagrange Equations[8]:

$$0 = J_{11}\ u + J_{12}\ v + J_{13} - \alpha\ \operatorname{div}\left(D_{\Psi_{\mathrm{C}}}(\mathbf{u})\ \nabla u\right)$$
$$0 = J_{12}\ u + J_{22}\ v + J_{23} - \alpha\ \operatorname{div}\left(D_{\Psi_{\mathrm{C}}}(\mathbf{u})\ \nabla v\right)$$

[6] Reminder: The trace of a symmetric matrix is the sum of its eigenvalues.

[7] This also includes the case where both eigenvalues are non-zero. In this case the Structure Tensor of $\mathbf{u}$ is even positive-definite.

[8] Deriving those is actually highly non-trivial. For more information see [18].

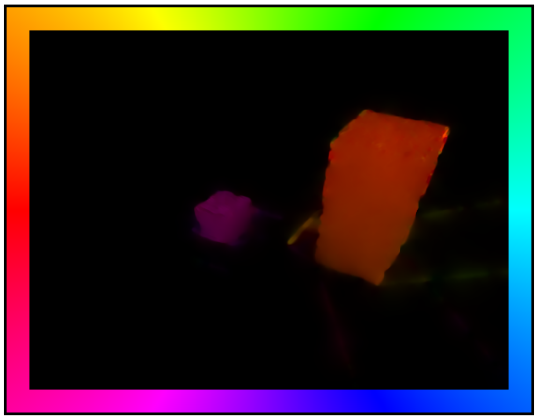


Figure 1.11: Result from our second approach. This time motion discontinuities are not destroyed.

where we have again a new Diffusion Tensor:

$$D_{\Psi_{\mathrm{C}}}(\mathbf{u}) := \Psi'_{\mathrm{C}}\left(\nabla u \nabla u^{\top} + \nabla v \nabla v^{\top}\right)$$

Note that Ψ'_{C} is applied to the eigenvalues of the underlying matrix.

Like in the case of Variational Image Restoration we can provide a generic form of our used Smoothness Terms and the associated Diffusion Tensors:

$$S^{\mathrm{F}}_{\Psi}(\mathbf{u}) := \mathrm{tr}\ \Psi\left(\nabla u \nabla u^{\top} + \nabla v \nabla v^{\top}\right)$$
$$D^{\mathrm{F}}_{\Psi}(\mathbf{u}) := \Psi'\left(\nabla u \nabla u^{\top} + \nabla v \nabla v^{\top}\right)$$

that again depend on the choice of Ψ:

- The Whittaker-Tikhonov penaliser creates our homogeneous isotropic term:[9]

$$S^{\mathrm{F}}_{\Psi_{\mathrm{WT}}}(\mathbf{u}) = \|\nabla u\|^2 + \|\nabla v\|^2$$
$$D^{\mathrm{F}}_{\Psi_{\mathrm{WT}}}(\mathbf{u}) = \Psi'_{\mathrm{WT}}\left(\nabla u \nabla u^{\top} + \nabla v \nabla v^{\top}\right) = I$$

- The Charbonnier penaliser leads to the piecewise anisotropic assumption:

$$S^{\mathrm{F}}_{\Psi_{\mathrm{C}}}(\mathbf{u}) = \mathrm{tr}\ \Psi_{\mathrm{C}}\left(\nabla u \nabla u^{\top} + \nabla v \nabla v^{\top}\right)$$
$$D^{\mathrm{F}}_{\Psi_{\mathrm{C}}}(\mathbf{u}) = \Psi'_{\mathrm{C}}\left(\nabla u \nabla u^{\top} + \nabla v \nabla v^{\top}\right)$$

We see in figure 1.11 that this method using our new term $S^{\mathrm{F}}_{\mathrm{C}}(\mathbf{u})$ meets our requirements: Motion discontinuities are preserved in the solution.

[9] Note that this is a special case of the Smoothness Term. (cf. [18])

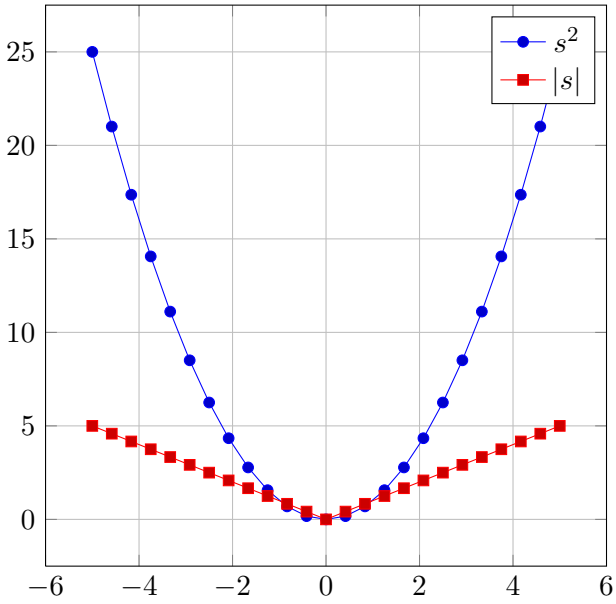


Figure 1.12: Plots of the quadratic function s^2 and the L_1-norm $|s|$.

1.3.3 A Robust Data Term

We now return to the improvement of our Data Term. We remember that we identified the following problems

- No unique estimation of the solution in regions possible where the Aperture Problem is present
- No estimation possible in the case of unreliable data

We dealt with the first issue by introducing the Smoothness Term that complemented the Data Term.

Let us now focus on the second issue: We realise that in the case of

- degraded data due to noise
- vanishing objects
- violation of our data assumptions

it makes no sense to minimise the energy generated by the Data Term as the minimiser will not be related to the correct solution by construction. This means we have to find a way to attenuate the energy generation of the Data Term in such a way that the fulfillment of the Smoothness Term becomes a viable option in these cases. In literature the quadratic penaliser is often replaced by a linear one like for example the *regularised* L_1*-norm* (cf. [9]):

$$\Psi_D(s^2) = \sqrt{s^2 + \epsilon} \approx |s|$$

where $\epsilon > 0$ ensures the differentiability in 0. We see in figure 1.12 that this penalisation generates in fact much less energy than the quadratic one.

Like in [8] we will employ a separate penalisation of our two assumptions:

$$\Psi_D\left(\mathbf{w}^\top J_{\mathrm{BCA}}\mathbf{w}\right) + \gamma\ \Psi_D\left(\mathbf{w}^\top J_{\mathrm{GCA}}\mathbf{w}\right)$$

This is motivated by the fact the if one assumption is violated the energy generated by the remaining one may still be useful for determining the solution.

Our Energy Functional is now given by

$$E_f(\mathbf{u}) = \int_\Omega \Bigg(\underbrace{\Psi_D\left(\mathbf{w}^\top J_{\mathrm{BCA}}\mathbf{w}\right) + \gamma\ \Psi_D\left(\mathbf{w}^\top J_{\mathrm{GCA}}\mathbf{w}\right)}_{=:D_f^{\mathrm{F}}(\mathbf{u})} + \alpha\ S_\Psi^{\mathrm{F}}(\mathbf{u}) \Bigg)\, d\mathbf{x}$$

We observe that this new Data Term leads to an unpleasant change in our Euler-Lagrange Equations:

$$\begin{aligned}
0 = &\left(\Psi'_D\left(\mathbf{w}^\top J_{BCA}\mathbf{w}\right) J_{BCA_{11}} + \gamma\ \Psi'_D\left(\mathbf{w}^\top J_{GCA}\mathbf{w}\right) J_{GCA_{11}}\right)\ u \\
&+ \left(\Psi'_D\left(\mathbf{w}^\top J_{BCA}\mathbf{w}\right) J_{BCA_{12}} + \gamma\ \Psi'_D\left(\mathbf{w}^\top J_{GCA}\mathbf{w}\right) J_{GCA_{12}}\right)\ v \\
&+ \left(\Psi'_D\left(\mathbf{w}^\top J_{BCA}\mathbf{w}\right) J_{BCA_{13}} + \gamma\ \Psi'_D\left(\mathbf{w}^\top J_{GCA}\mathbf{w}\right) J_{GCA_{13}}\right) - \alpha\ \mathrm{div}\left(D_\Psi^{\mathrm{F}}(\mathbf{u})\ \nabla u\right) \\
0 = &\left(\Psi'_D\left(\mathbf{w}^\top J_{BCA}\mathbf{w}\right) J_{BCA_{12}} + \gamma\ \Psi'_D\left(\mathbf{w}^\top J_{GCA}\mathbf{w}\right) J_{GCA_{12}}\right)\ u \\
&+ \left(\Psi'_D\left(\mathbf{w}^\top J_{BCA}\mathbf{w}\right) J_{BCA_{22}} + \gamma\ \Psi'_D\left(\mathbf{w}^\top J_{GCA}\mathbf{w}\right) J_{GCA_{22}}\right)\ v \\
&+ \left(\Psi'_D\left(\mathbf{w}^\top J_{BCA}\mathbf{w}\right) J_{BCA_{23}} + \gamma\ \Psi'_D\left(\mathbf{w}^\top J_{GCA}\mathbf{w}\right) J_{GCA_{23}}\right) - \alpha\ \mathrm{div}\left(D_\Psi^{\mathrm{F}}(\mathbf{u})\ \nabla v\right)
\end{aligned}$$

The Motion Tensor Entries now depend nonlinearly on $\mathbf{u}$. In order to restore readability we now introduce the following abbreviations for the new Motion Tensor entries:

$$\begin{aligned}
&\widehat{J}_{**} := \left(\Psi'_D\left(\mathbf{w}^\top J_{BCA}\mathbf{w}\right) J_{BCA_{**}} + \gamma\ \Psi'_D\left(\mathbf{w}^\top J_{GCA}\mathbf{w}\right) J_{GCA_{**}}\right) \\
&\text{with } (**) \in \{(11), (12), (13), (22), (23)\}
\end{aligned}$$

These abbreviations allow us to rewrite the Euler-Lagrange Equations in our usual way:

$$\begin{aligned}
0 &= \widehat{J}_{11}\ u + \widehat{J}_{12}\ v + \widehat{J}_{13} - \alpha\ \mathrm{div}\left(D_\Psi^{\mathrm{F}}(\mathbf{u})\ \nabla u\right) \\
0 &= \widehat{J}_{12}\ u + \widehat{J}_{22}\ v + \widehat{J}_{23} - \alpha\ \mathrm{div}\left(D_\Psi^{\mathrm{F}}(\mathbf{u})\ \nabla v\right)
\end{aligned}$$

We can see in figure 1.13 that we were successful: The effect of vanishing objects is in fact drastically reduced by replacing our quadratic penaliser in the Data Term by this new one. This type of Data Term is also called a *Robust Data Term* as it provides robustness against unreliable data (cf. [9]). We will refer to this Data Term in the following as $D_f^{\mathrm{F}}(\mathbf{u})$.

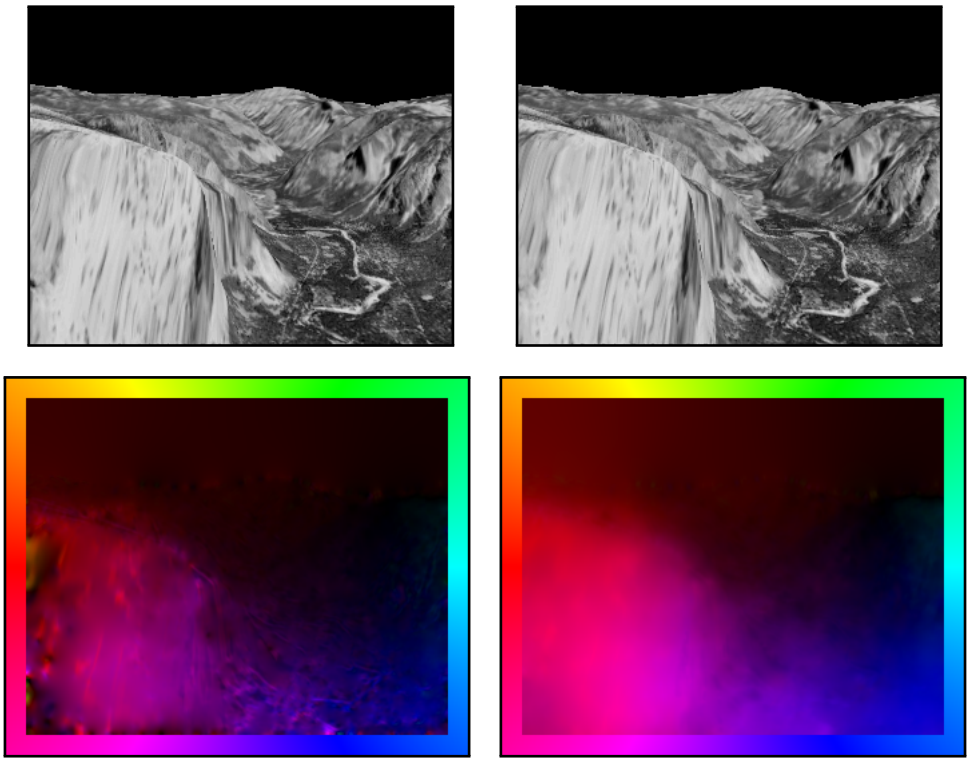


Figure 1.13: Experiment inspired by [8]: Comparison between our old Data Term and our new one. **Top Row:** Frames 8 and 9 of the *Yosemite sequence without clouds* (source: [4], [3]), **Bottom Row:** Left: Old Data Term, Right: New Data Term.

1.3.4 Dealing with large displacements

Until now we only considered the case of small displacements that allowed us to perform the linearisation of our data assumptions. In the case of large displacements this is, however, in general no longer a viable option as a linearisation might not approximate the true function properly.

Let us now have a look at the Energy Functional where our Robust Data Term was not linearised:

$$E_f(\mathbf{u}) = \int_\Omega \Bigg(\underbrace{\Psi_D((f(x+u,y+v,t+1)-f(x,y,t))^2) + \gamma\Psi_D(\|\nabla_2 f(x+u,y+v,t+1)-\nabla_2 f(x,y,t)\|^2)}_{=:\widehat{D}_f^{\mathrm{F}}(\mathbf{u})} + \alpha\ S_\Psi^{\mathrm{F}}(\mathbf{u}) \Bigg)\, d\mathbf{x}$$

In the following we will refer to this Data Term as $\widehat{D}_f^{\mathrm{F}}(\mathbf{u})$.

We realise that this Functional may be non-convex and thus may have multiple minima. In literature here often the so-called *Warping Strategy* (cf. [9], [15]) is used to find at least a good minimiser.

The Warping Strategy considers the data f on different scales such that

- f^1 is the original data
- f^{s+1} is the data on a smaller scale than f^s
- $f^{\max}$ is the data on the smallest scale
- and the difference between two scales is not very big

The main idea of this strategy is motivated by the following observation: Let us assume that we are given a part of the solution $\mathbf{u}^s$ on the scale s and that we only have to compute the missing part $\mathbf{du}^s$. If this missing part is sufficiently small we may for example linearise the Brightness Constancy Assumption with respect to $\mathbf{du}^s$ as follows:

$$\begin{aligned}\Big(f^s(x+u^s+du^s,y+v^s+dv^s,t+1)-f^s(x,y,t)\Big)^2 \approx \\ \Big(\underbrace{f^s(x+u^s,y+v^s,t+1)-f^s(x,y,t)}_{\approx f_t} + du^s f_x^s(x+u^k,y+v^s,t+1) + \\ dv^s f_y^s(x+u^k,y+v^s,t+1)\Big)^2 \approx \\ \Big(du^s f_x^s + dv^s f_y^s + f_t^s\Big)^2\end{aligned}$$

where we only performed the linearisation in x- and y-direction and held the time coordinate fixed at $t+1$.

A similar approach is also possible for the Gradient Constancy Assumption ([9]), which fortunately means that we can compute our Motion Tensor entries like before by using now the images $f^s(x,y,t)$ and $f^s(x+u^s, y+v^s, t+1)$. Hence, we are now trying to estimate the motion between these two images.

In addition to that we realise that motions will become smaller if we use an appropriate downscaling of our image data. So the smallest scale should be chosen in such a way that $u^{\max} = 0$, that is, only sufficiently small displacements remain there.

Then the Warping Strategy starts on the smallest scale $s = \max$ and repeats the following steps until the original scale $s = 1$ of the data is reached:

1. Obtain our solution $\mathbf{du}^s$ on the current scale by minimising the Energy Functional

$$E_f(\mathbf{du}^s) = \int_{\Omega^s} \left(D^{\mathrm{F}}_{f^s}(\mathbf{du}^s) + \alpha \; S^{\mathrm{F}}_{\Psi}(\mathbf{u}^s + \mathbf{du}^s) \right) d\mathbf{x}$$

   where

   - Ω^s represents the domain of the current scale
   - $D^{\mathrm{F}}_{f^s}$ is our Robust Data Term that is this time linearised like above by using the images $f^s(x,y,t)$ and $f^s(x+u^s, y+v^s, t+1)$.
   - S^{F}_{Ψ} now depends on $\mathbf{du}^s$ and $\mathbf{u}^s$.

   Its Euler-Lagrange Equations are given by

$$0 = \widehat{J}^s_{11} du^s + \widehat{J}^s_{12} dv^s + \widehat{J}^s_{13} - \alpha \; \mathrm{div}\left(D^{\mathrm{F}}_{\Psi}(\mathbf{u}^s + \mathbf{du}^s) \; (\nabla u^s + \nabla du^s) \right)$$
$$0 = \widehat{J}^s_{12} du^s + \widehat{J}^s_{22} dv^s + \widehat{J}^s_{23} - \alpha \; \mathrm{div}\left(D^{\mathrm{F}}_{\Psi}(\mathbf{u}^s + \mathbf{du}^s) \; (\nabla v^s + \nabla dv^s) \right)$$

   where the notation $\widehat{J}^s_{**}$ indicates that the Motion Tensor entry is derived by using the images $f^s(x+u^s, y+u^s, t+1)$ and $f^s(x,y,t)$.

   We also have the new Boundary Conditions

$$\mathbf{n}^\top \begin{pmatrix} F_{u_x} \\ F_{u_y} \end{pmatrix} = \mathbf{n}^\top \nabla \; (\mathbf{u}^s + \mathbf{du}^s) = 0$$
$$\mathbf{n}^\top \begin{pmatrix} F_{v_x} \\ F_{v_y} \end{pmatrix} = \mathbf{n}^\top \nabla \; (\mathbf{v}^s + \mathbf{dv}^s) = 0$$

2. Compute $\mathbf{u}^{s-1}$ by taking the sum $\mathbf{u}^s + \mathbf{du}^s$ and by transfering the result to the next finer scale.

3. Proceed with the next scale by setting $s = s - 1$.

When we finally have reached the finest scale $s = 1$ we again perform the first step to obtain $\mathbf{du}^s$. Then we compute our final solution by taking the sum $\mathbf{u} = \mathbf{u}^1 + \mathbf{du}^1$.

Thus we split our original possibly non-convex problem into a series of problems with linearised Data Terms. It can be shown that these problems are in our case even strictly convex and have a unique solution if the data has some properties. Note, however, that the proof of this statement is highly non-trivial and requires advanced knowledge in the field of *Functional Analysis* and therefore is well beyond the scope of this work. The interested reader is encouraged to consult [22] on this matter.

We will see later on in chapter 3 how this strategy is applied in the discrete setting.

This once again finishes our modelling process in the case of the Optic Flow Estimation. We remark that this time it posed quite a challenge.

1.3.5 Limitations and possible extensions

So far we tested our method only with one specific type of motion, namely a translational one. Moreover we even used knowledge about the correct motion in our design process to model the Data Term and in particular the Smoothness Term. Roughly speaking we created the constraint that the first-order derivatives of our solution should vanish. That is, we again modelled a first-order Smoothness Term. In terms of the motion it represents the assumption that it should be translational or piecewise translational. This translational motion belongs to the class of so-called *Rigid Body Motions*. These are motions that do not change the distance between points that belong to the moving object. (cf. [14])

Like in the previous case we now want to test our method with an image pair that violates this assumption on purpose. This means we require a motion that is no longer a Rigid Body Motion. Such motions occur for example in *Uniaxial Tensile Experiments* that are conducted in Mechanical Engineering.

Figure 1.14 shows such an experiment. We see that the material sample is stretched along the y-axis. By inspecting the image pair closer we find that in the centre nearly no motion in y-direction is occuring. In the upper region, however, we see a motion towards the top whereas in the lower region the sample moves downward. If we now assume that the motion is continuously changing we conclude that we have $v_y \neq 0$, that is our displacement in y-direction is no longer constant in space, which clearly represents a violation of our smoothness assumption.

Unfortunately this image pair was taken from an actual real world experiment and thus no ground truth is available. As replacement we will use the result of the software *Vic-2D 2009* (cf. [2]) as a *reference solution*. This software is based on the *Principle of Digital Image Correlation* ([19]).

We will now use our method equipped with the Robust Data Term that was not linearised to compute the motion.

In a first experiment we use the Smoothness Term Ψ^{F}_{WT}. Unfortunately we observe a devastating result in figure 1.15: On the one hand we see that we can not estimate the correct direction in the upper half of the image. On the other hand we also discover that we seem to estimate the wrong magnitude of the displacements.

Next we try out our Smoothness Term $\Psi^{\mathrm{F}}_{\mathrm{C}}$. We observe that this time we can better estimate the magnitude of the displacements. Also the directional information looks a lot better in this experiment. But we observe some strange behaviour in the structure of the solution: Instead of showing a linearly changing motion we have a piecewise constant one. However, in terms of our assumption this behaviour is correct: We expected the solution to be piecewise constant.

Again these observations force us to admit that our assumptions on the correct motion have been incorrect from the start. Once more we are tempted to design a second-order smoothness term that allows the linearly changing motion in the result. But this would also be a fixed-order assumption that is bound to data with this very property. Again we are motivated to design a flexible-order model that can be adapted to the problem at hand.

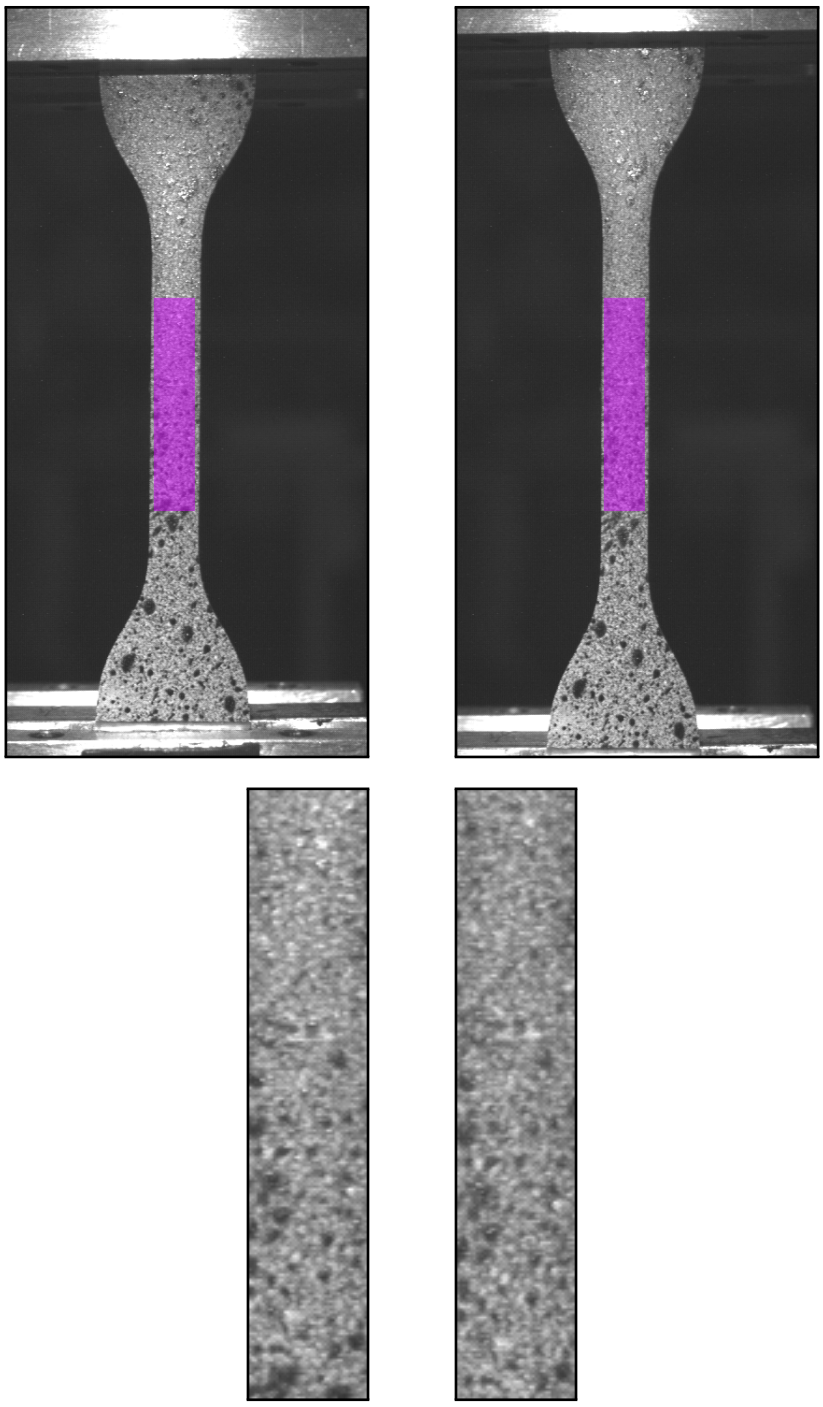


Figure 1.14: Our test images of a non-rigid motion. **Top Row**: Two frames of an Uniaxial Tensile Experiment where the area of interest is colored in purple. **Bottom Row**: The respective enlarged areas of interest.

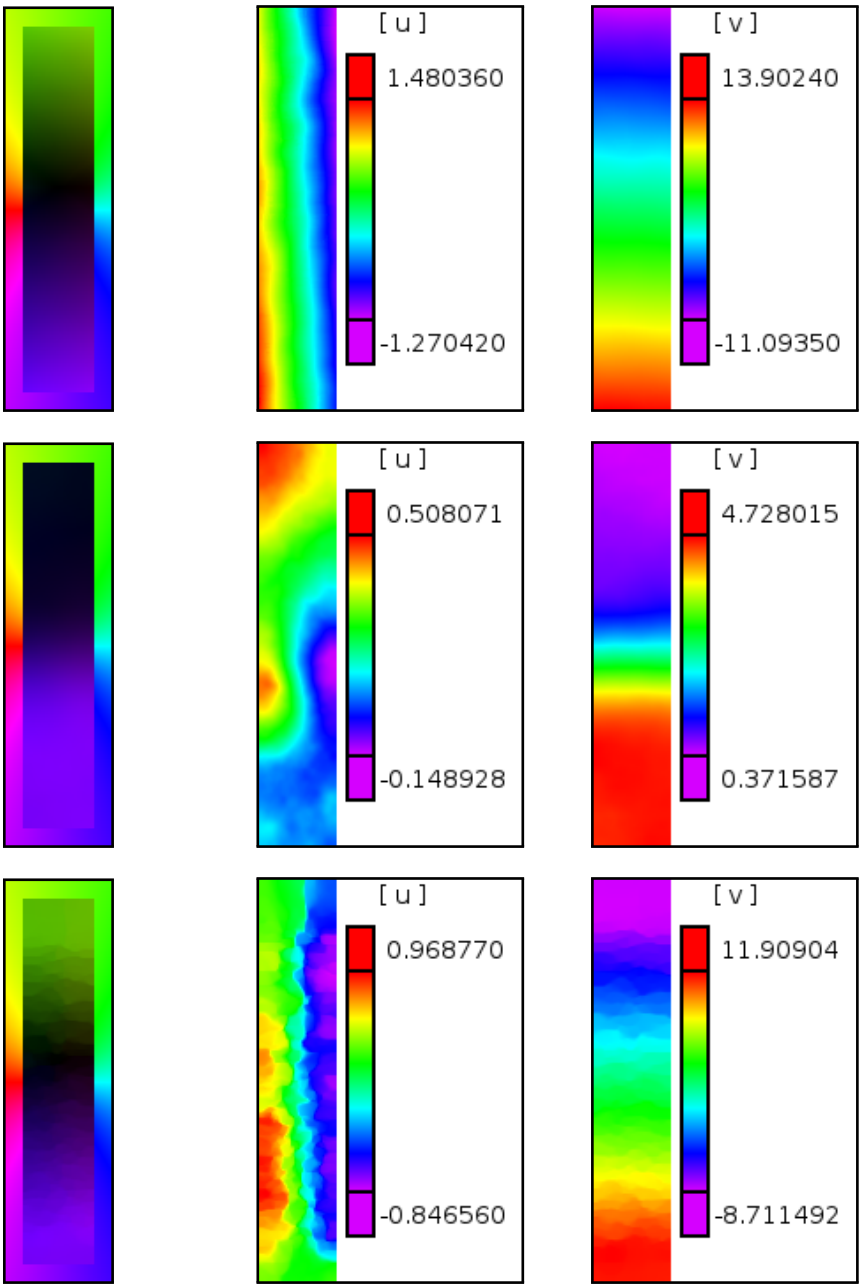


Figure 1.15: Reference solution and attempts of our motion estimation. **Top Row**: Reference solution from Vic2D. **Centre Row**: Our attempt with Ψ_{WT}. **Bottom Row**: Our attempt with Ψ_{C}.

In the case of Variational Image Restoration we briefly discussed the possible benefit of higher order derivatives of the solution. Of course we are now interested if derivatives of a motion also play an important role. Let us to this end have a look at the *Displacement Gradient*:

$$\nabla \mathbf{u} := \begin{pmatrix} u_x & u_y \\ v_x & v_y \end{pmatrix}$$

In Mechanical Engineering this matrix is often used to compute the so-called *Lagrangean Strain Tensor*:

$$E := \begin{pmatrix} e_{11} & e_{12} \\ e_{12} & e_{22} \end{pmatrix} = \frac{1}{2}\Big((\nabla \mathbf{u} + I)^\top (\nabla \mathbf{u} + I) - I \Big)$$

This quantity describes how much a motion differs locally from a Rigid Body Motion (cf. [14]). If E degenerates to the trivial matrix

$$E = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

the motion represents a Rigid Body Motion. Otherwise it is said to be non-rigid.

Let us now shortly investigate its entries in order to understand their meaning if E does not vanish[10]:

- e_{11}: Strain in x-direction

  This component measures how line elements parallel to the x-axis change their length with respect to their initial length. If the value is positive we have a stretching of the material. If it is negative we have a so-called compression.

- e_{22}: Strain in y-direction:

  Like above but for line elements parallel to the y-axis.

- e_{12}: Shear Strain

  Roughly speaking this quantity is related to angle changes that occur during deformations.

Again we are motivated to design a new Smoothness Term type that allows the simultaneous estimation of the motion and its derivatives while using a flexible-order smoothness assumption.

[10] Here we use the terminology presented in presented in [2]

2 A new type of Smoothness Term

In the previous chapter we focused on the modelling process of Variational Methods. We discovered that fixing the smoothness assumption to a specific type of data may lead to dissatisfactory results in the special cases where we considered a different type of data that violated this assumption. Moreover we realised that it may be beneficial to estimate also higher order derivatives of the solution as well as these provide additional insight into the structure of the solution.

These observations motivate the idea to design a new Smoothness Term type that should allow

- to be adapted to the problem at hand with a single parameter
- the estimation of higher order derivatives of the solution
- and to be used in different Variational Methods

In this chapter we will first look briefly at some known methods. Unfortunately we will discover that none of them meets our requirements. Thus we will proceed by designing our own term that finally provides the desired functionality. After that we will combine this term with an Image Restoration Method and an Optic Flow Estimation Method in order to investigate the structure of the Euler-Lagrange Equations.

2.1 Related Work

We remark that higher order smoothness assumptions represent no new idea: [21] for example proposed an Optic Flow method with a second-order smoothness assumption. However, we remember that we wanted to avoid such fixed assumptions in order to avoid the problem we encountered in the previous chapter.

Another approach by [5] formulates in each point of the domain an Energy Functional in order to estimate the Displacement Field and its first-order derivatives locally. Ultimately it uses a second-order smoothness assumption and is moreover bound to an Optic Flow problem.

[6] on the other hand proposed a flexible-order smoothness term where the order can be adjusted by manipulating a single parameter in order to denoise images. Unfortunately it provides no estimates of the derivatives of the solution.

Finally we discuss the method proposed by [13]: Here the Smoothness Term of an Optic Flow approach was manipulated in such a way that the method simultaneously estimates the Displacement Field and the associated Displacement Gradient. Moreover it uses a second-order smoothness assumption.

Thus we conclude that none of these methods provides a suitable candidate for our needs.

2.2 A First Idea

Unfortunately none of the presented methods fully met our requirements. Therefore we will now design our own Smoothness Term type. To this end we first consider the method from [13] that allows a simultaneous estimation of the Optic Flow and the associated Displacement Gradient:

$$E_f(\mathbf{u}, A) = \int_\Omega \left(D_f^{\mathrm{F}}(\mathbf{u}) + \alpha \left(\|A - \nabla \mathbf{u}\|_2^2 + \alpha \text{ tr } \Psi_C \left(\sum_{i,j=1}^{2} \nabla a_{ij} \nabla a_{ij}^\top \right) \right) \right) d\mathbf{x}$$

Let us first investigate the properties of the used data and the solution. Like in the case of the Optic Flow Estimation we considered in the first chapter we are working with a gray-valued image sequence with the same properties like before. That is we have once again $f : \Omega \times \mathbb{R}_0^+ \to [0, 255]$, which is two-times continuously differentiable. By turning now to the solution we discover a novelty: In addition to our Displacement Field $\mathbf{u} = (u, v)^\top$ we have also a 2×2 matrix-valued function or *matrix field* $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$. This means our solution actually consists of a set of a vector-valued function and a matrix-valued one. For their co-domains we have

- $\mathbf{u} : \Omega \to I_1 \subset \mathbb{R}^2$ and
- $A : \Omega \to I_2 \subset \mathbb{R}^{2\times 2}$

where I_1 and I_2 should be bounded. Unlike in the first chapter we are now not going to model the functional but going to understand the model assumptions. We discover that the functional is using our linearised Robust Data Term $D_f^{\mathrm{F}}(\mathbf{u})$ that we derived in the previous chapter. The other terms, however, do not reveal their model idea directly, which forces us to investigate their meaning.

To this end we will first hold A fixed and only minimise with respect to $\mathbf{u}$. This means we are minimising the following functional[1]:

$$E_f^1(\mathbf{u}) = \int_\Omega \left(D_f^{\mathrm{F}}(\mathbf{u}) + \alpha \|A - \nabla \mathbf{u}\|_2^2 \right) d\mathbf{x}$$

where

- $\|\cdot\|_2$ is the *Frobenius Norm*, which can be interpreted as the generalisation of the *Euclidean Norm* to matrices
- α is again a positive scalar

[1] Note that the other part does not depend on $\mathbf{u}$ and thus degenerates to an additive constant that can be omitted.

- and $\nabla \mathbf{u}$ is the Displacement Gradient we encountered in the previous chapter.

Thus we see that a minimiser has the following properties

- $\mathbf{u}$ fulfills the given Data Term
- $\nabla \mathbf{u}$ is similar to A in the squared Frobenius Norm

This functional actually reminds us of the Optic Flow Method we encountered in the first chapter. But this time the functional does not explicitly model the idea that the derivatives of $\mathbf{u}$ should vanish. Moreover we note that we are using a similarity constraint on one part of the solution that interacts with the other. As this means that we want the two parts to "agree" on a solution we may call this new type of term *Agreement Term*. We see that this term is weighted by α expressing its importance.

In the next step we will now hold $\mathbf{u}$ fixed and minimise with respect to A. The resulting Energy Functional is given by

$$E^2(A) = \alpha \int_\Omega \left(\|A - \nabla \mathbf{u}\|_2^2 + \alpha \text{ tr } \Psi_C \left(\sum_{i,j=1}^{2} \nabla a_{ij} \nabla a_{ij}^\top \right) \right) \, d\mathbf{x}$$

where

$$\text{tr } \Psi_C \left(\sum_{i,j}^{2} \nabla a_{ij} \nabla a_{ij}^\top \right)$$

seems to represent a first-order smoothness assumption on the matrix field A. It resembles strongly the Smoothness Term

$$S^{\mathrm{F}}_{\Psi_{\mathrm{C}}}(\mathbf{u}) = \text{tr } \Psi_{\mathrm{C}} \left(\nabla u \nabla u^\top + \nabla v \nabla v^\top \right)$$

we encountered in the Optic Flow Estimation modelling process. This time we are again combining the Structure Tensors of the individual entries of A to compute the Structure Tensor of A that describes the behaviour of A locally. Like in the last chapter applying the Charbonnier penaliser to the eigenvalues of the resulting tensor and taking the trace of the resulting matrix yields a piecewise anisotropic smoothness assumption. By replacing Ψ_C by Ψ_{WT} we obtain a homogeneous isotropic term:

$$\text{tr } \Psi_{\mathrm{WT}} \left(\sum_{i,j=1}^{2} \nabla a_{ij} \nabla a_{ij}^\top \right) = \sum_{i,j=1}^{2} \|\nabla a_{ij}\|^2$$

Like before we can formulate a generic version for these two Smoothness Terms and their associated Diffusion Tensors:

$$S^M_\Psi(A) = \text{tr } \Psi \left(\sum_{i,j=1}^{2} \nabla a_{ij} \nabla a_{ij}^\top \right)$$

$$D^M_\Psi(A) = \Psi' \left(\sum_{i,j=1}^{2} \nabla a_{ij} \nabla a_{ij}^\top \right)$$

that depend on the choice of Ψ.

Let us now return to our energy: We remark that the outer α does only influence the value of the minimum and not the minimiser itself, which enables us to replace it by the value 1:

$$E^2(A) = \int_\Omega \left(\|A - \nabla \mathbf{u}\|_2^2 + \alpha \;\; S_\Psi^M(A) \right) d\mathbf{x}$$

Again we find the properties of the minimiser A:

- A and $\nabla \mathbf{u}$ are similar in the squared Frobenius Norm
- A has a first-order smoothness where the type of the smoothness depends on the choice of Ψ

We see that the Smoothness Term is weighted by α that mirrors the desired importance of the assumption.

Let us now combine these findings to formulate the properties of the minimiser of the whole Energy Functional

$$E_f(\mathbf{u}, A) = \int_\Omega \left(D_f^{\mathrm{F}}(\mathbf{u}) + \alpha \;\; \left(\|A - \nabla \mathbf{u}\|_2^2 + \alpha \;\; S_\Psi^M(A) \right) \right) d\mathbf{x}$$

1. $\mathbf{u}$ fulfills a given Data Term
2. A and $\nabla \mathbf{u}$ are similar
3. A has a first-order smoothness

Let us now interprete the consequences of these properties. If we combine the statements 2 and 3 we realise that A should be an estimate of the Displacement Gradient $\nabla \mathbf{u}$ and should have a first-order smoothness at the same time. By merging now the properties 2 and 3 we find that $\mathbf{u}$ should describe the motion between the two images and that its Displacement Gradient should be similar to A. As A should have a first-order smoothness we conclude that $\nabla \mathbf{u}$ should have this smoothness as well, which causes $\mathbf{u}$ to have a second-order smoothness. This means that the properties 2 and 3 represent a second-order smoothness assumption for $\mathbf{u}$ that may also be piecewise and anisotropic.

Moreover we realise that we also gain access to an estimate of the Displacement Gradient of $\mathbf{u}$ in the form of A.

By inspecting this construction to achieve a second-order smoothness assumption for $\mathbf{u}$ we find that a similar approach could be used to model a second-order smoothness assumption for A, which in turn would create a third-order smoothness assumption for $\mathbf{u}$. To this end let us now replace the third wanted property by two new ones:

1. $\mathbf{u}$ fulfills a given Data Term
2. A and $\nabla \mathbf{u}$ are similar
3. ∇A and B are similar

4. B has a first-order smoothness

By using the the same reasoning as before we conclude that properties 2 to 4 describe a third-order smoothness assumption for $\mathbf{u}$. Moreover we gain access to an estimate of the second-order derivatives of $\mathbf{u}$ in the form of B.

We, however, have a problem that keeps us from modelling the Energy Functional right away: We do not have a reasonable definition of ∇A if A is matrix-valued. Furthermore we have no means of modelling the similarity between two quantities that are neither scalar-valued nor vector-valued nor matrix-valued. So before proceeding we have to formulate some definitions that allow us the modelling.

2.3 Preparations

Definition 1. Let A be a tensor. Then we define the following quantities:

- The set

$$\mathrm{Idx}(A) := \{\ p \mid A_p \text{ is entry of } A\ \}$$

  is called the *Index Set* of A. Thus it represents the set of all indexes of A.

- The number $\mathrm{ord}(A) \in \mathbb{N}_0$ with

$$\mathrm{Idx}(A) \subset \mathbb{N}^{\mathrm{ord}(A)}$$

  is called the *order* of the tensor A.

Definition 2. Let A be a tensor-valued function defined on $\mathbb{R}^2$ that is continuously differentiable. Then we define ∇A as the tensor A^* containing the entries:

$$A^*_{p1} = \frac{\partial}{\partial x} A_p$$
$$A^*_{p2} = \frac{\partial}{\partial y} A_p$$

where

$$p \in \mathrm{Idx}(A)$$
$$pi := (p, i) \in \mathrm{Idx}(A^*) = \mathrm{Idx}(A) \times \{1, 2\}$$

with $i \in \{1, 2\}$.

We, however, have to consider the special case of A being scalar-valued as in this case we have $\mathrm{Idx}(A) = \emptyset$ and $\mathrm{ord}(A) = 0$. Here we define ∇A as the ordinary gradient:

$$A^*_1 = \frac{\partial}{\partial x} A$$
$$A^*_2 = \frac{\partial}{\partial y} A$$

We notice that applying this operator to a tensor-valued function A increases the order by 1 and doubles the entry amount.

Definition 3. Let A and B be tensors with the same Index Set $\mathrm{Idx}(A)$. Then we define the sum of both tensors as the tensor C containing the entries

$$C_p := A_p + B_p$$

with $p \in \mathrm{Idx}(A)$ and $\mathrm{Idx}(C) = \mathrm{Idx}(A)$.

Again we have to consider the special case of A being a scalar. Then we define the above sum to degenerate to the usual sum:

$$C = A + B$$

Definition 4. Let A be a tensor. Then we define the *Generalised Frobenius Norm* as

$$\|A\|_2 := \sqrt{\sum_{p \in \mathrm{Idx}(A)} A_p^2}$$

Again we have a special case if A is a scalar. Here we define the sum to degenerate to a single summand:

$$\sum_{p \in \mathrm{Idx}(A)} A_p^2 = A^2$$

and thus

$$\|A\|_2 = |A|$$

Conventions

As a convention we assume that A_p degenerates always to A if the Index Set of A is empty. In addition to that we will sometimes speak of tensors instead of tensor-valued functions if the meaning is clear from the context.

2.4 Designing a flexible Smoothness Term

Now we are finally prepared to model our first idea. We wanted to modify the following Energy Functional

$$E_f(\mathbf{u}, A) = \int_\Omega \left(D_f^{\mathrm{F}}(\mathbf{u}) + \alpha \ \left(\|A - \nabla \mathbf{u}\|_2^2 + \alpha \ \ S_\Psi^M(A) \right) \right) d\mathbf{x}$$

where the minimiser has the properties

1. $\mathbf{u}$ fulfills a given Data Term
2. A and $\nabla \mathbf{u}$ are similar
3. A has a first-order smoothness

by replacing the third demanded property by two new ones:

1. **u** fulfills a given Data Term
2. A and $\nabla \mathbf{u}$ are similar
3. ∇A and B are similar
4. B has a first-order smoothness

Let us start by modelling the fourth property. In the previous section we defined the generalised ∇-operator for tensors of arbitrary order. Thus we are now able to compute ∇A: The result is a tensor A^* containing all first-order derivatives of A with respect to x and y:

$$\begin{aligned} A^*_{p1} &= \frac{\partial}{\partial x} A_p \\ A^*_{p2} &= \frac{\partial}{\partial y} A_p, \ p \in \mathrm{Idx}(A) \end{aligned}$$

As A is a second-order tensor containing 4 entries this means A^* will be a third-order tensor consisting of 8 entries. Taking into to account this knowledge we conclude that B should have the same structure as A^* and so should have the same Index Set. Thus $B : \Omega \to I_3 \subset \mathbb{R}^{2\times 2\times 2}$ where I_3 should again be bounded. As usual we demand the solution part B to be two-times continuously differentiable.

This enables us finally to model the similarity between ∇A and B by using our Generalised Frobenius Norm:

$$\begin{aligned} \|B - \nabla A\|_2^2 &= \left(\sqrt{\sum_{p \in \mathrm{Idx}(A)} \left(B_{p1} - \frac{\partial}{\partial x} A_p \right)^2 + \left(B_{p2} - \frac{\partial}{\partial y} A_p \right)^2} \right)^2 \\ &= \left(B_{111} - \frac{\partial}{\partial x} A_{11} \right)^2 + \left(B_{112} - \frac{\partial}{\partial y} A_{11} \right)^2 + \left(B_{121} - \frac{\partial}{\partial x} A_{12} \right)^2 + \left(B_{122} - \frac{\partial}{\partial y} A_{12} \right)^2 + \\ &\quad \left(B_{211} - \frac{\partial}{\partial x} A_{21} \right)^2 + \left(B_{212} - \frac{\partial}{\partial y} A_{21} \right)^2 + \left(B_{221} - \frac{\partial}{\partial x} A_{22} \right)^2 + \left(B_{222} - \frac{\partial}{\partial y} A_{22} \right)^2 \end{aligned}$$

Thus we realise that if we are given tensors C and D with the same order $\mathrm{ord}(C)$, entry amount and Index Set $\mathrm{Idx}(C)$ we can always state a similarity constraint using the Generalised Frobenius Norm:

$$\|C - D\|_2^2 = \begin{cases} \sum_{p \in \mathrm{Idx}(C)} (C_p - D_p)^2, & \text{if } \mathrm{ord}(C) > 0 \\ (C - D)^2, & \text{if } \mathrm{ord}(C) = 0 \end{cases}$$

This finishes the modelling process for the fourth property. Let us now continue with the last one: The first-order smoothness of B. To this end we first consider our

Smoothness Term we were using for A:

$$S_\Psi^M(A) = \text{tr } \Psi \left(\sum_{i,j=1}^{2} \nabla a_{ij} \nabla a_{ij}^\top \right)$$

If we rewrite S^M by using the Index-Set notation we arrive at:

$$\text{tr } \Psi \left(\sum_{i,j=1}^{2} \nabla a_{ij} \nabla a_{ij}^\top \right) = \text{tr } \Psi \left(\sum_{p \in \text{Idx}(A)} \nabla A_p \nabla A_p^\top \right)$$

By using this new notation we discover that this Smoothness Term can now be used for B as well:

$$\text{tr } \Psi \left(\sum_{p \in \text{Idx}(B)} \nabla B_p \nabla B_p^\top \right)$$

Here we discover that again the Structure Tensors of all entries of B are combined to compute the Structure Tensor of B. By applying our previously acquired knowledge about this tensor we realise that it describes the behaviour of B locally. And again the chosen function Ψ determines the details of the smoothness assumption. Furthermore this observation implies that it will also work for every tensor-valued function C.

So by using the Index Set notation we created a term that automatically adapts to the tensor the term is applied to.

Thus we arrive at a *Generalised First-Order Smoothness Term* that works with tensors of arbritrary order and entry amount:

$$S_\Psi^{\text{G}}(C) := \text{tr } \Psi \left(\sum_{p \in \text{Idx}(C)} \nabla C_p \nabla C_p^\top \right)$$

with the associated Diffusion Tensor:

$$D_\Psi^{\text{G}}(C) := \Psi' \left(\sum_{p \in \text{Idx}(C)} \nabla C_p \nabla C_p^\top \right)$$

Applying this term to our previous solution types produces our previous Smoothness Terms:

- u is scalar-valued:

$$S_\Psi^{\text{G}}(u) = \Psi \left(|\nabla u|^2 \right) = S_\Psi^{\text{I}}(u)$$

- $\mathbf{u}$ is vector-valued:

$$S_\Psi^{\text{G}}(\mathbf{u}) = \text{tr } \Psi \left(\sum_{p \in \text{Idx}(\mathbf{u})} \nabla \mathbf{u}_p \nabla \mathbf{u}_p^\top \right) = \text{tr } \Psi \left(\nabla u \nabla u^\top + \nabla v \nabla v^\top \right) = S_\Psi^{\text{F}}(\mathbf{u})$$

- A is matrix-valued:

$$S_\Psi^{\mathrm{G}}(A) = \operatorname{tr}\ \Psi\left(\sum_{p\in\mathrm{Idx}(A)} \nabla A_p \nabla A_p^\top\right) = \operatorname{tr}\ \Psi\left(\sum_{i,j}^{2} \nabla a_{ij} \nabla a_{ij}^\top\right) = S_\Psi^{\mathrm{M}}(A)$$

We remark that in the scalar-valued case, that is, $\mathrm{ord}(C) = 0$ we encounter a special case: Choosing $\Psi = \Psi_{\mathrm{C}}$ here creates a piecewise isotropic smoothness assumption instead of an anisotropic one like in the other cases.

This concludes the modelling for the fifth demanded property.

Finally we hold $\mathbf{u}$ and A fixed and model the energy

$$E^3(B) = \int_\Omega \left(\|B - \nabla A\|_2^2 + \alpha\ S_\Psi^G(B)\ \right) d\mathbf{x}$$

We see that a minimiser fulfills our wanted properties:

- ∇A and B are similar
- B has a first-order smoothness

Replacing now our previous second-order smoothness assumption by this new term finally yields our sought Energy Functional

$$E_f(\mathbf{u}, A, B) = \int_\Omega \left(D_f^{\mathrm{F}}(\mathbf{u}) + \alpha\ \left(\|A - \nabla \mathbf{u}\|_2^2 + \alpha\ \left(\|B - \nabla A\|_2^2 + \alpha\ S_\Psi^G(B) \right) \right) \right) d\mathbf{x}$$

Again we can verify that the minimiser $(\mathbf{u}, A, B)$ meets our requirements we stated earlier:

1. $\mathbf{u}$ fulfills a given Data Term
2. A and $\nabla \mathbf{u}$ are similar
3. ∇A and B are similar
4. B has a first-order smoothness

This modelling approach is now very motivating. We used the same approach that was used to create a second-order assumption to model a third-order one for the solution $\mathbf{u}$. Of course now the question occurs if this process can be extended to create Smoothness Terms of arbritrary order.

In order to investigate this question we first introduce a new notation for our displacement field:

- $A^1 := \mathbf{u}$

Then we consider our Energy Functional from the previous chapter that uses a first-order Smoothness Term:

$$E_f(A^1) = \int_\Omega \left(D_f^{\mathrm{F}}(A^1) + \alpha \; S_\Psi^G(A^1) \right) d\mathbf{x}$$

Next we again change the notation:

- $A^2 := \mathbf{u}$
- $A^1 := A$

in order to rewrite the Energy Functional from [13] that introduced a second-order smoothness assumption:

$$E_f(A^1, A^2) = \int_\Omega \left(D_f^{\mathrm{F}}(A^2) + \alpha \left(\|A^1 - \nabla A^2\|_2^2 + \alpha \; S_\Psi^{\mathrm{G}}(A^2) \right) \right) d\mathbf{x}$$

And finally we alter the notation for the last time:

- $A^3 := \mathbf{u}$
- $A^2 := A$
- $A^1 := B$

for rewritting our previous Energy Functional that we modelled to use a third-order Smoothness Term for the Displacement Field:

$$E_f(A^1, A^2, A^3) = \int_\Omega \left(D_f^{\mathrm{F}}(A^3) + \alpha \left(\|A^2 - \nabla A^3\|_2^2 + \alpha \left(\|A^1 - \nabla A^2\|_2^2 + \alpha \; S_\Psi^{\mathrm{G}}(A^1) \right) \right) \right) d\mathbf{x}$$

By taking now a very close look at these functionals we recognise the following pattern:

$$E_f(\Lambda) = \int_\Omega \left(D_f^{\mathrm{F}}(A^n) + \alpha \; M_\Psi^n(\Lambda) \right) d\mathbf{x}$$

with

$$M_\Psi^k := \begin{cases} S_\Psi^G(A^k) & \text{if } k = 1 \\ \left(\|A^{k-1} - \nabla A^k\|_2^2 + \alpha \; M_\Psi^{k-1}(\Lambda) \right) & \text{else} \end{cases}$$

where

- $\Lambda = \bigcup_{k=1}^n A^k$ is a set of tensor-valued functions
- $\alpha > 0$ as always
- and n represents the order of the smoothness assumption.

Furthermore we observed that the structure of A^k for $k < n$ was entirely determined by A^{k+1} where A^k had to share the following properties with ∇A^{k+1}:

- The Index Set $\mathrm{Idx}(\nabla A^{k+1})$,
- the order $\mathrm{ord}(\nabla A^{k+1})$
- and the entry amount $|\mathrm{Idx}(\nabla A^{k+1})|$

For the Displacement Field A^n, however, the structure was explicitly given:

- $\mathrm{ord}(A^n) = 1$
- $\mathrm{Idx}(A^n) = \{1, 2\}$
- $|\mathrm{Idx}(A^n)| = 2$ entries

Thus we see that the structure of all other tensors is implicitly determined by the Displacement Field A^n. As applying the ∇-operator increases the order by 1 and doubles the entries we have for $k < n$:

- $\mathrm{ord}(A^k) = \mathrm{ord}(A^{k+1}) + 1 = \mathrm{ord}(A^n) + n - k = n - k + 1$
- $\mathrm{Idx}(A^k) = \mathrm{Idx}(A^{k+1}) \times \{1, 2\} = \mathrm{Idx}(A^n) \times \overbrace{\{1, 2\} \times \ldots \times \{1, 2\}}^{n-k \text{ times}} = \overbrace{\{1, 2\} \times \ldots \times \{1, 2\}}^{n-k+1 \text{ times}}$
- $|\mathrm{Idx}(A^k)| = 2 \cdot 2^{n-k}$ entries

As usual we demand A^k to be defined on Ω, to have a bounded co-domain and to be twice continuously differentiable.

All in all we see that this is an Energy Functional that uses a *recursively* defined Smoothness Term $M^n_\Psi(\Lambda)$ that allows to adjust the order of the smoothness by changing the single parameter n. This implies that $M^k_\Psi(\Lambda)$ is a k-th order smoothness assumption for A^k.

Let us now verify this observation by using an inductive reasoning.

Base Case: $m = 1$

If we set $k = 1$ we arrive at

$$M^1_\Psi(\Lambda) = S^G_\Psi(A^1)$$

which is by construction a first-order smoothness assumption for A^1. Thus for $k = 1$ the observation holds true.

Inductive Step: $(m - 1) \to m$

Here we assume that $M^{m-1}_\Psi(\Lambda)$ represents a smoothness term with order $m-1$ for A^{m-1} and that $m - 1 < n$. By setting now $k = m - 1$ we get the term

$$M^m_\Psi(\Lambda) = \left(\|A^{m-1} - \nabla A^m\|^2_2 + \alpha \; M^{m-1}_\Psi(\Lambda) \right)$$

Let us now first construct the energy

$$E^{m-1}(A^{m-1}) = \int_\Omega \left(\|A^{m-1} - \nabla A^m\|_2^2 + \alpha \; M_\Psi^{m-1}(\Lambda) \right) d\mathbf{x}$$

that we want to minimise with respect to A^{m-1}.[2] Like before we can identify the following properties of the minimiser:

- A^{m-1} is similar to ∇A^m in the squared Generalised Frobenius Norm
- A^{m-1} has a $m-1$-th-order smoothness

Now we minimise the same energy with respect to A^m and discover the property of the minimiser:

- ∇A^m is similar to A^{m-1} in the squared Generalised Frobenius Norm

We remember that A^{m-1} has a $m-1$-th-order smoothness. This means that ∇A^m should also obtain this property because of the similarity contraint, which in turn will cause A^m to have a m-th order smoothness. This concludes the Inductive Step.

Thus we see that our observation was correct: $M_\Psi^k(\Lambda)$ is a k-th order smoothness assumption for A^k. Moreover this smoothness may be

- homogeneous and isotropic if $\Psi = \Psi_{\mathrm{WT}}$
- or piecewise and anisotropic if $\Psi = \Psi_{\mathrm{C}}$

Furthermore we discovered in the run of the inductive reasoning an interesting feature of this approach: We saw in the Inductive Step that A^{m-1} should be similar to ∇A^m, which means that the tensor A^{m-1} should contain estimates of the first-order derivatives of A^m. Let us now investigate the contents of A^m: Here we distinguish between two cases:

1. $m = n$, then we have that A^m is the Displacement Field, that is, the part of the solution that directly interacts with the given data. This implies that A^{m-1} contains an estimate of the first-order derivatives of the Displacement Field.
2. $m < n$, then we have that A^m is an estimate of the first-order derivatives of A^{m+1}. But this means that A^{m-1} is a coarse estimate of the second-order derivatives of A^{m+1}.

By taking into account this observation we find the following connection between the tensors A^k:

- A^n is the Displacement Field
- A^{n-1} is an estimate of the first-order derivatives of the Displacement Field

[2] Thus we are again holding the other parts of the solution fixed.

- A^{n-2} is an estimate of the second-order derivatives of the Displacement Field
- ...
- $A^1 = A^{n-(n-1)}$ is an estimate of the $n-1$-th-order derivatives of the Displacement Field

Thus we can conclude that we derived a Smoothness Term for an Optic Flow Estimation that

- allows to set the smoothness order by a single parameter $n > 0$
- and estimates besides the Displacement Field also derivatives of up to $n-1$-th order of it if $n > 1$

We realise that we can also use it for Variational Image Restoration:

$$E_f(\Lambda) = \int_\Omega \left(D_f^{\mathrm{I}}(A^n) + \alpha \; M_\Psi^n(\Lambda) \right) d\mathbf{x}$$

with

$$M_\Psi^n := \begin{cases} S_\Psi^G(A^k) & \text{if } k = 1 \\ \left(\|A^{k-1} - \nabla A^k\|_2^2 + \alpha \; M_\Psi^{k-1}(\Lambda) \right) & \text{else} \end{cases}$$

where

- A^n is now scalar-valued as it represents the wanted denoised image
- $D_f^{\mathrm{I}}(A^n)$ is our Data Term for Variational Image Restoration we modelled in the previous chapter:

$$D_f^{\mathrm{I}}(A^n) = (A^n - f)^2$$

- and f is now a gray-value image that needs only to be continuous like in chapter 1

However, we discover that the structure of A^k with $k \in [1, n]$ changes:

- For A^n we have
  - A^n is scalar-valued and thus $\mathrm{ord}(A^n) = 0$
  - A^n does not use any indexes, so we have $\mathrm{Idx}(A^n) = \emptyset$
  - A^n has only one entry
- For A^k with $k \in [1, n-1]$ we get:
  - $\mathrm{ord}(A^k) = \mathrm{ord}(A^{k+1}) + 1 = \mathrm{ord}(A^n) + n - k = n - k$
  - $\mathrm{Idx}(A^k) = \mathrm{Idx}(A^{k+1}) \times \{1,2\} = \mathrm{Idx}(A^n) \times \overbrace{\{1,2\} \times \ldots \times \{1,2\}}^{n-k \text{ times}} = \overbrace{\{1,2\} \times \ldots \times \{1,2\}}^{n-k \text{ times}}$

– $|\mathrm{Idx}(A^k)| = 1 \cdot 2^{n-k}$ entries

Here we saw again how the structure of A^k with $k \neq n$ is implicitly determined by the solution part A^n that interacts directly with the Data Term.

Moreover we find that for $n = 1$ we have the special case of the Smoothness Term: It is

- homogeneous and isotropic if $\Psi = \Psi_{\mathrm{WT}}$
- piecewise and isotropic if $\Psi = \Psi_{\mathrm{C}}$ instead of anisotropic as in the remaining cases.

We conclude that we succeeded in modelling a new type of Smoothness Term that on the one hand allows to set the order of the smoothness assumption in a flexible way by changing a single parameter n. On the other hand it simultaneously estimates the solution A^n and its derivatives of up to order $n-1$ if $n > 1$. Moreover we see that the new approaches degenerate to our basic functionals we derived in the first chapter if we set $n = 1$. In the following we will refer to these new methods as the Generic Optic Flow Estimation and the Generic Image Restoration.

However, we remember that we need to derive the Euler-Lagrange Equations in order to obtain a minimiser. Therefore we will next explore how this equations and the arising Boundary Conditions look like: We will first investigate the case of Generic Image Restoration. Then we proceed with the case of Generic Optic Flow Estimation where we will use like above the linearised version of our Robust Data Term. Finally we will have a look at the Generic Optic Flow Estimation where we are using the Warping Strategy to find a minimiser of the functional where the Data Term was not linearised.

2.5 Euler-Lagrange Equations for Generic Image Restoration

By looking at the functional for our Generic Image Restoration

$$E_f(\Lambda) = \int_\Omega \underbrace{\left(D_f^{\mathrm{I}}(A^n) + \alpha \; M_\Psi^n(\Lambda) \right)}_{=F} \, d\mathbf{x}$$

we see that the integrand F depends this time on each $A^k \in \Lambda$ and the first-order derivatives of its entries in x- and y-direction. This observation enables us to state a general version of the Euler-Lagrange Equations and the associated Boundary Conditions:

For each A_p^k with $p \in \mathrm{Idx}(A^k)$ and $k \in [1, n]$ we have the equation

$$0 = F_{A_p^k} - \frac{\partial}{\partial x} F_{A_{p_x}^k} - \frac{\partial}{\partial y} F_{A_{p_y}^k}$$

and the Boundary Condition

$$\mathbf{n}^\top \begin{pmatrix} F_{A_{p_x}^k} \\ F_{A_{p_y}^k} \end{pmatrix} = 0$$

Now we will compute these equations and Boundary Conditions of the functional for different values of n.

By setting $n = 1$ we receive our functional from the first chapter[3]:

$$E_f(A^1) = \int_\Omega \left(D_f^{\mathrm{I}}(A^1) + \alpha \; S_\Psi^G(A^1) \right) d\mathbf{x}$$

The Euler-Lagrange Equation is given by

- For A^1: [4]

$$0 = A^1 - f + \alpha \; \mathrm{div}\Big(D_\Psi^G(A^1) \; \nabla A^1 \Big)$$

This is our first type of Euler-Lagrange Equation:

1. Equations that are generated by the participation of A^k in the Data Term and the Smoothness Term

The Boundary Condition is given by

- For A_1:

$$\mathbf{n}^\top \nabla A^1 = 0$$

We also found our first type of Boundary Condition:

1. Boundary Conditions originating from the Smoothness Term

We continue by setting $n = 2$ and arrive at the functional:

$$E_f(A^1, A^2) = \int_\Omega \left(D_f^{\mathrm{I}}(A^2) + \alpha \; \left(\|A^1 - \nabla A^2\| + \alpha \; S_\Psi^G(A^1) \right) \right) d\mathbf{x}$$

Again we compute first the Euler-Lagrange Equations:

- For A^1:

$$0 = A_1^1 - A_x^2 - \alpha \; \mathrm{div}\Big(D_\Psi^G(A^1) \nabla A_1^1 \Big)$$
$$0 = A_2^1 - A_y^2 - \alpha \; \mathrm{div}\Big(D_\Psi^G(A^1) \nabla A_2^1 \Big)$$

We discovered a new type of Euler-Lagrange Equation:

2. Equations generated by the participation of A^k in the Agreement Term and the Smoothness Term

Moreover we find that this type seems to replace the first one if $n > 1$ is used, which means that the first type is a special case that only occurs for $n = 1$.

[3] Note that we are using a different notation this time.

[4] Remember that A^n is scalar-valued and has therefore an empty Index Set.

- For A^2 we get:

$$0 = A^2 - f + \alpha \left(A^1_{1_x} - A^2_{xx} + A^1_{2_y} - A^2_{yy} \right)$$

Again we found a new type of Euler-Lagrange Equation:

3. Equations that are created by the participation of A^k in the Data Term and the Agreement Term

Now we have a look at the Boundary Conditions:

- For A^1:

$$\mathbf{n}^\top \nabla A^1_1 = 0$$
$$\mathbf{n}^\top \nabla A^1_2 = 0$$

- For A^2:

$$\mathbf{n}^\top \begin{pmatrix} A^1_1 - A^2_x \\ A^1_2 - A^2_y \end{pmatrix} = 0$$

Also a new type of Boundary Condition is produced by this functional:

2. Boundary Conditions generated by the Agreement Term

Informally this Boundary Condition mirrors the assumption that the respective parts of the Agreement Term are always fulfilled across the boundary. This actually reminds us of the Boundary Conditions that are produced by the Smoothness Term, which express the assumption that the smoothness is automatically fulfilled across the boundary.

Again we increase n to $n = 3$ and obtain the functional:

$$E_f(A^1, A^2, A^3) = \int_\Omega \left(D^{\mathrm{I}}_f(A^3) + \alpha \left(\|A^2 - \nabla A^3\| + \alpha \left(\|A^1 - \nabla A^2\| + \alpha \, S^G_\Psi(A^1) \right) \right) \right) d\mathbf{x}$$

Once again we focus first on the Euler-Lagrange Equations:

- For A^1:

$$0 = A^1_{11} - A^2_{1_x} - \alpha \, \mathrm{div} \left(D^G_\Psi(A^1) \nabla A^1_{11} \right)$$
$$0 = A^1_{12} - A^2_{1_y} - \alpha \, \mathrm{div} \left(D^G_\Psi(A^1) \nabla A^1_{12} \right)$$
$$0 = A^1_{21} - A^2_{2_x} - \alpha \, \mathrm{div} \left(D^G_\Psi(A^1) \nabla A^1_{21} \right)$$
$$0 = A^1_{22} - A^2_{2_y} - \alpha \, \mathrm{div} \left(D^G_\Psi(A^1) \nabla A^1_{22} \right)$$

- For A^2:

$$0 = A_1^2 - A_x^3 + \alpha \left(A_{11_x}^1 - A_{1_{xx}}^2 + A_{12_y}^1 - A_{1_{yy}}^2 \right)$$
$$0 = A_2^2 - A_y^3 + \alpha \left(A_{21_x}^1 - A_{2_{xx}}^2 + A_{22_y}^1 - A_{2_{yy}}^2 \right)$$

So we discovered a fourth type of Euler-Lagrange Equation:

4. Equations arising from the participation of A^k in two different Agreement Terms

We realise that this type of equation might occur very often if the amount of Agreement Terms increases, which makes this equation type the most common one for high values of n.

- For A^3:

$$0 = A^3 - f + \alpha \left(A_{1_x}^2 - A_{xx}^3 + A_{2_y}^2 - A_{yy}^3 \right)$$

For the Boundary Conditions we obtain:

- For A^1:

$$\mathbf{n}^\top \nabla A_{11}^1 = 0$$
$$\mathbf{n}^\top \nabla A_{12}^1 = 0$$
$$\mathbf{n}^\top \nabla A_{21}^1 = 0$$
$$\mathbf{n}^\top \nabla A_{22}^1 = 0$$

- For A^2:

$$\mathbf{n}^\top \begin{pmatrix} A_{11}^1 - A_{1_x}^2 \\ A_{12}^1 - A_{1_y}^2 \end{pmatrix} = 0$$
$$\mathbf{n}^\top \begin{pmatrix} A_{21}^1 - A_{2_x}^2 \\ A_{22}^1 - A_{2_y}^2 \end{pmatrix} = 0$$

- For A^3:

$$\mathbf{n}^\top \begin{pmatrix} A_1^2 - A_x^3 \\ A_2^2 - A_y^3 \end{pmatrix} = 0$$

Thus we see that this time no new Boundary Conditions emerge.

Let us now consider the functional for $n = 4$:

$$E_f(A^1, A^2, A^3, A^4) =$$
$$\int_\Omega \left(D_f^{\mathrm{I}}(A^4) + \alpha \left(\|A^3 - \nabla A^4\| + \alpha \left(\|A^2 - \nabla A^3\| + \alpha \left(\|A^1 - \nabla A^2\| + \alpha \, S_\Psi^G(A^1) \right) \right) \right) \right) d\mathbf{x}$$

Like usual we start by investigating the Euler-Lagrange Equations:

- For A^1:

$$
\begin{aligned}
0 &= A^1_{111} - A^2_{11_x} - \alpha \ \mathrm{div}\Big(D^G_\Psi(A^1) \nabla A^1_{111} \Big) \\
0 &= A^1_{112} - A^2_{11_y} - \alpha \ \mathrm{div}\Big(D^G_\Psi(A^1) \nabla A^1_{112} \Big) \\
0 &= A^1_{121} - A^2_{12_x} - \alpha \ \mathrm{div}\Big(D^G_\Psi(A^1) \nabla A^1_{121} \Big) \\
0 &= A^1_{122} - A^2_{12_y} - \alpha \ \mathrm{div}\Big(D^G_\Psi(A^1) \nabla A^1_{122} \Big) \\
0 &= A^1_{211} - A^2_{21_x} - \alpha \ \mathrm{div}\Big(D^G_\Psi(A^1) \nabla A^1_{211} \Big) \\
0 &= A^1_{212} - A^2_{21_y} - \alpha \ \mathrm{div}\Big(D^G_\Psi(A^1) \nabla A^1_{212} \Big) \\
0 &= A^1_{221} - A^2_{22_x} - \alpha \ \mathrm{div}\Big(D^G_\Psi(A^1) \nabla A^1_{221} \Big) \\
0 &= A^1_{222} - A^2_{22_y} - \alpha \ \mathrm{div}\Big(D^G_\Psi(A^1) \nabla A^1_{222} \Big)
\end{aligned}
$$

- For A^2:

$$
\begin{aligned}
0 &= A^2_{11} - A^3_{1_x} + \alpha \ \Big(A^1_{111_x} - A^2_{11_{xx}} + A^1_{112_y} - A^2_{11_{yy}} \Big) \\
0 &= A^2_{12} - A^3_{1_y} + \alpha \ \Big(A^1_{121_x} - A^2_{12_{xx}} + A^1_{122_y} - A^2_{12_{yy}} \Big) \\
0 &= A^2_{21} - A^3_{2_x} + \alpha \ \Big(A^1_{211_x} - A^2_{21_{xx}} + A^1_{212_y} - A^2_{21_{yy}} \Big) \\
0 &= A^2_{22} - A^3_{2_y} + \alpha \ \Big(A^1_{221_x} - A^2_{22_{xx}} + A^1_{222_y} - A^2_{22_{yy}} \Big)
\end{aligned}
$$

- For A^3:

$$
\begin{aligned}
0 &= A^3_1 - A^4_x + \alpha \ \Big(A^2_{11_x} - A^3_{1_{xx}} + A^2_{12_y} - A^3_{1_{yy}} \Big) \\
0 &= A^3_2 - A^4_y + \alpha \ \Big(A^2_{21_x} - A^3_{2_{xx}} + A^2_{22_y} - A^3_{2_{yy}} \Big)
\end{aligned}
$$

- For A^4:

$$
0 = A^4 - f + \alpha \ \Big(A^3_{1_x} - A^4_{xx} + A^3_{2_y} - A^4_{yy} \Big)
$$

So this time we observed also no new type of Euler-Lagrange Equation. As we are again only using Agreement Terms and the Smoothness Term we will not encounter new Boundary Conditions.

We observe that increasing n even further will not produce new types of Euler-Lagrange Equations because no new term interactions are possible. Hence, we will now briefly summerise our findings:

First we focus on the Euler-Lagrange Equation types where we identified four versions:

1. If A^k participates in both the Data Term and the Smoothness Term we obtain a Euler-Lagrange Equation with the structure

$$0 = A^k - f + \alpha \operatorname{div}\left(D_\Psi^G(A^k)\nabla A^k\right)$$

   We have determined that this equation only occurs in the special case of $n = 1$ for A^1.

2. Equations created by the participation of A^k in the Data Term and in the Agreement Term:

$$0 = A^k - f + \alpha \left(A^{k-1}_{1_x} - A^k_{xx} + A^{k-1}_{2_y} - A^k_{yy}\right)$$

   This type of equation only occurs in the case of A^n for $n > 1$.

3. Two equations that are created by the participation of A^k in the Agreement Term and in the Smoothness Term:

$$0 = A^k_{p1} - A^{k+1}_{p_x} - \alpha \operatorname{div}\left(D_\Psi^G(A^k)\nabla A^k_{p1}\right)$$
$$0 = A^k_{p2} - A^{k+1}_{p_y} - \alpha \operatorname{div}\left(D_\Psi^G(A^k)\nabla A^k_{p2}\right)$$

   where $p \in \mathrm{Idx}(A^{k+1})$. We always encounter this type in the case of A^1 for $n > 1$.

4. Two equations that are created by the participation of A^k in two different Agreement Terms

$$0 = A^k_{p1} - A^{k+1}_{p_x} + \alpha \left(A^{k-1}_{p11_x} - A^k_{p1_{xx}} + A^{k-1}_{p12_y} - A^k_{p1_{yy}}\right)$$
$$0 = A^k_{p2} - A^{k+1}_{p_y} + \alpha \left(A^{k-1}_{p21_x} - A^k_{p2_{xx}} + A^{k-1}_{p22_y} - A^k_{p2_{yy}}\right)$$

   where $p \in \mathrm{Idx}(A^{k+1})$.

   This type of equation occurs for A^k with $1 < k < n$ and $n > 2$, thus making it the most common one for high values of n.

In the case of Boundary Conditions we encountered two types:

1. Boundary Conditions produced by the Smoothness Term:

$$\mathbf{n}^\top \nabla A^k_p = 0$$

   where $p \in \mathrm{Idx}(A^k)$. We have seen that these only occur for A^1.

2. Boundary Conditions originating from the Agreement Term:

$$\mathbf{n}^\top \begin{pmatrix} A^{k-1}_{p1} - A^k_{p_x} \\ A^{k-1}_{p2} - A^k_{p_y} \end{pmatrix} = 0$$

where $p \in \mathrm{Idx}(A^k)$. These occur for A^k for $1 < k \leq n$ if $n > 1$.

Thus we are finished with classifying the Euler-Lagrange Equations and Boundary Conditions belonging to our Generic Image Restoration. These will play a fundamental role in the next chapter where we will discretise these equations in order to find a minimiser in a discrete setting.

2.6 Euler-Lagrange Equations for Generic Optic Flow Estimation without Warping

Now we turn to the Optic Flow Estimation that uses our new term. The Energy Functional is given by:

$$E_f(\Lambda) = \int_\Omega \underbrace{\left(D^{\mathrm{F}}_f(A^n) + \alpha \; M^n_\Psi(\Lambda) \right)}_{=F} d\mathbf{x}$$

We notice that once again the integrand F depends on $A^k \in \Lambda$ and the first-order derivatives of its entries in x- and y-direction. This means we can use the same general version of the Euler-Lagrange Equations and the associated Boundary Conditions we encountered in the previous section:

For each A^k_p with $p \in \mathrm{Idx}(A^k)$ and $k \in [1, n]$ we have the equation

$$0 = F_{A^k_p} - \frac{\partial}{\partial x} F_{A^k_{p_x}} - \frac{\partial}{\partial y} F_{A^k_{p_y}}$$

and the Boundary Condition

$$\mathbf{n}^\top \begin{pmatrix} F_{A^k_{p_x}} \\ F_{A^k_{p_y}} \end{pmatrix} = 0$$

Luckily this finding implies that we will encounter the same types of Euler-Lagrange Equations and Boundary Conditions we already saw in the previous section.

Thus we have as Boundary Conditions again the two types:

1. Boundary Conditions produced by the Smoothness Term:

$$\mathbf{n}^\top \nabla A^k_p = 0$$

where $p \in \mathrm{Idx}(A^k)$. We have seen that these only occur for A^1.

2. Boundary Conditions originating from the Agreement Term:

$$\mathbf{n}^\top \begin{pmatrix} A^{k-1}_{p1} - A^k_{p_x} \\ A^{k-1}_{p2} - A^k_{p_y} \end{pmatrix} = 0$$

where $p \in \mathrm{Idx}(A^k)$. These occur for A^k for $1 < k \leq n$ if $n > 1$.

By inspecting the Euler-Lagrange Equation types

1. Equations created by the participation of A^n in both the Data Term and the Smoothness Term
2. Equations created by the participation of A^n in the Data Term and in the Agreement Term
3. Equations that are created by the participation of A^k in the Agreement Term and in the Smoothness Term
4. Equations that are created by the participation of A^k in two different Agreement Terms

we realise that the types 1 and 2 will have a different structure in this case because on the one hand A^n is vector-valued and on the other hand we have a different Data Term this time.

Thus in order to obtain the structure of these types we will inspect the functionals for $n = 1$ and $n = 2$.

We start with $n = 1$ and obtain the energy that corresponds to our usual Optic Flow Estimation approach we modelled in the first chapter:

$$E_f(A^1) = \int_\Omega \left(D^{\mathrm{F}}_f(A^1) + \alpha \; S^G_\Psi(A^1) \right) d\mathbf{x}$$

For the Euler-Lagrange Equations we get:

- For A^1: [5]

$$0 = \widehat{J}_{11} \; A^1_1 + \widehat{J}_{12} \; A^1_2 + \widehat{J}_{13} - \alpha \; \mathrm{div}\Big(D^G_\Psi(A^1) \nabla A^1_1 \Big)$$
$$0 = \widehat{J}_{12} \; A^1_1 + \widehat{J}_{22} \; A^1_2 + \widehat{J}_{23} - \alpha \; \mathrm{div}\Big(D^G_\Psi(A^1) \nabla A^1_2 \Big)$$

  We see that this time we obtain two equations due to A^1 being vector-valued. Furthermore we see that the contribution from the Data Term is different from the one in the Image Restoration case.

[5] Remember that unlike in the Image Restoration Case A^n is this time vector-valued.

Let us now turn to the functional where $n = 2$, which is actually the approach by [13]:

$$E_f(A^1, A^2) = \int_\Omega \left(D_f^{\mathrm{F}}(A^2) + \alpha \ \left(\|A^1 - \nabla A^2\|_2^2 + \alpha \ S_\Psi^G(A^1) \right) \right) d\mathbf{x}$$

Again we derive the Euler-Lagrange Equations by using our general formula:

- For A^2:

$$0 = \widehat{J}_{11} \ A_1^2 + \widehat{J}_{12} \ A_2^2 + \widehat{J}_{13} + \alpha \ \left(A_{11_x}^1 - A_{1_{xx}}^2 + A_{12_y}^1 - A_{1_{yy}}^2 \right)$$
$$0 = \widehat{J}_{12} \ A_1^2 + \widehat{J}_{22} \ A_2^2 + \widehat{J}_{23} + \alpha \ \left(A_{21_x}^1 - A_{2_{xx}}^2 + A_{22_y}^1 - A_{2_{yy}}^2 \right)$$

  This is the new variant of the second equation type. Again we see that we obtain two equations this time.

- For A^1:

$$0 = A_{11}^1 - A_{1_x}^2 - \alpha \ \mathrm{div} \left(D_\Psi^G(A^1) \ \nabla A_{11}^1 \right)$$
$$0 = A_{12}^1 - A_{1_y}^2 - \alpha \ \mathrm{div} \left(D_\Psi^G(A^1) \ \nabla A_{12}^1 \right)$$
$$0 = A_{21}^1 - A_{2_x}^2 - \alpha \ \mathrm{div} \left(D_\Psi^G(A^1) \ \nabla A_{21}^1 \right)$$
$$0 = A_{22}^1 - A_{2_x}^2 - \alpha \ \mathrm{div} \left(D_\Psi^G(A^1) \ \nabla A_{22}^1 \right)$$

  These are the equations that are created by the participation of A^k in the Agreement Term and the Smoothness Term. As suspected in the beginning we see that this type of equation does not change.

Thus we are finally ready to summarise our findings about the Euler-Lagrange Equations and the Boundary Conditions that arise in the Generic Optic Flow approach:

For the Euler-Lagrange Equations we have again 4 types where only the first two differ from the ones we have in the Image Restoration case:

1. If A^k participates in both the Data Term and the Smoothness Term we obtain this time two Euler-Lagrange Equations with the structure

$$0 = \widehat{J}_{11} \ A_1^k + \widehat{J}_{12} \ A_2^k + \widehat{J}_{13} - \alpha \ \mathrm{div} \left(D_\Psi^G(A^k) \ \nabla A_1^k \right)$$
$$0 = \widehat{J}_{12} \ A_1^k + \widehat{J}_{22} \ A_2^k + \widehat{J}_{23} - \alpha \ \mathrm{div} \left(D_\Psi^G(A^k) \ \nabla A_2^k \right)$$

   Once again these equations only occur in the special case of $n = 1$ for A^1.

2. The two equations that are created by the participation of A^k in the Data Term and in the Agreement Term:

$$0 = \widehat{J}_{11}\ A_1^k + \widehat{J}_{12}\ A_2^k + \widehat{J}_{13} + \alpha\ \left(A_{11_x}^{k-1} - A_{1_{xx}}^k + A_{12_y}^{k-1} - A_{1_{yy}}^k\right)$$
$$0 = \widehat{J}_{12}\ A_1^k + \widehat{J}_{22}\ A_2^k + \widehat{J}_{23} + \alpha\ \left(A_{21_x}^{k-1} - A_{2_{xx}}^k + A_{22_y}^{k-1} - A_{2_{yy}}^k\right)$$

These equations only occur in the case of A^n for $n > 1$.

3. Two equations that are created by the participation of A^k in the Agreement Term and in the Smoothness Term:

$$0 = A_{p1}^k - A_{p_x}^{k+1} - \alpha\ \mathrm{div}\left(D_\Psi^G(A^k)\ \nabla A_{p1}^k\right)$$
$$0 = A_{p2}^k - A_{p_y}^{k+1} - \alpha\ \mathrm{div}\left(D_\Psi^G(A^k)\ \nabla A_{p2}^k\right)$$

where $p \in \mathrm{Idx}(A^{k+1})$. We always encounter this type in the case of A^1 for $n > 1$.

4. Two equations that are created by the participation of A^k in two different Agreement Terms

$$0 = A_{p1}^k - A_{p_x}^{k+1} + \alpha\ \left(A_{p11_x}^{k-1} - A_{p1_{xx}}^k + A_{p12_y}^{k-1} - A_{p1_{yy}}^k\right)$$
$$0 = A_{p2}^k - A_{p_y}^{k+1} + \alpha\ \left(A_{p21_x}^{k-1} - A_{p2_{xx}}^k + A_{p22_y}^{k-1} - A_{p2_{yy}}^k\right)$$

where $p \in \mathrm{Idx}(A^{k+1})$.

This type of equation occurs for A^k with $1 < k < n$ and $n > 2$, thus making it also in the Optic Flow case the most common one for high values of n.

In the case of Boundary Conditions we have the same two types as in the Image Restoration case:

1. Boundary Conditions produced by the Smoothness Term:

$$\mathbf{n}^\top \nabla A_p^k = 0$$

where $p \in \mathrm{Idx}(A^k)$. We have seen that these only occur for A^1.

2. Boundary Conditions originating from the Agreement Term:

$$\mathbf{n}^\top \begin{pmatrix} A_{p1}^{k-1} - A_{p_x}^k \\ A_{p2}^{k-1} - A_{p_y}^k \end{pmatrix} = 0$$

where $p \in \mathrm{Idx}(A^k)$. These occur for A^k for $1 < k \leq n$ if $n > 1$.

Hence we now have also access to the Euler-Lagrange Equations and Boundary Conditions that belong to the Generic Optic Flow approach. However, we remember that we used the linearised version of our Robust Data Term for modelling the Energy Functional. In the next chapter we will use the Robust Data Term that was not linearised and explore how the Euler-Lagrange Equations and Boundary Conditions look like if we are applying the Warping Strategy. These are actually the last ingredients we require before we can get started with the discretisaion in chapter 3.

2.7 Euler-Lagrange Equations for Optic Flow Estimation with Warping

We now combine our new Optic Flow approach with our Robust Data Term that was not linearised and obtain the functional:

$$E_f(\Lambda) = \int_\Omega \left(\widehat{D}_f^{\mathrm{F}}(A^n) + M_\Psi^n(\Lambda) \right) d\mathbf{x}$$

Like in chapter 1 we now use the Warping Strategy and consider our image sequence f on different scales:

- f^1 is the original data
- f^{s+1} is the data on a smaller scale than f^s
- $f^{\max}$ is the data on the smallest scale

In order to allow the linearisation of the Data Term we again split the Displacement Field on the current scale s into a known part $A^{n,s}$ and an unknown part $dA^{n,s}$.[6] For Completeness we note that $dA^{n,s}$ and $A^{n,s}$ use the same Index Set.

This implies that our solution on the current scale s will now have the structure $\Lambda^s := \bigcup_{i=1}^{n-1} A^{i,s} \cup dA^{n,s}$ where $A^{k,s}$ represents the tensor A^k on the current scale s and $dA^{n,s}$ is the unknown part of the Displacement Field.

Similar to our method in chapter 1 we will use the following approach to obtain a minimiser Λ of our functional: We start at the scale $s = \max$ and repeat the following steps until we reach the scale $s = 1$:

1. Obtain our solution Λ^s on the current scale by minimising the Energy Functional

$$E_f(\Lambda^s) = \int_\Omega \underbrace{\left(D_{f^s}^{\mathrm{F}}(dA^{n,s}) + \alpha\ M_\Psi^{n,s}(\Lambda^s) \right)}_{=F} d\mathbf{x}$$

with

$$M_\Psi^{k,s}(\Lambda^s) := \begin{cases} S_\Psi^G(A^{k,s} + dA^{k,s}) & , n = 1,\ k = 1 \\ S_\Psi^G(A^{k,s}) & , n \neq 1,\ k = 1 \\ \left(\|A^{k-1,s} - \nabla(A^{k,s} + dA^{k,s})\|_2^2 + \alpha\ M_\Psi^{k-1,s}(\Lambda^s) \right) & , n \neq 1,\ k = n \\ \left(\|A^{k-1,s} - \nabla(A^{k,s})\|_2^2 + \alpha\ M_\Psi^{k-1,s}(\Lambda^s) \right) & , 1 < k < n \end{cases}$$

where

- Ω^k again represents the domain of the current scale

[6] Note that we do not have to split the remaining parts of our solution as these do not directly interact with the Data Term.

- $D^{\mathrm{F}}_{f^s}$ is our Robust Data Term that is this time linearised like in chapter 1 by using the images $f^s(x,y,t)$ and $f^s(x+A_1^{n,s}, y+A_2^{n,s}, t+1)$.
- and $A^{n,s}$ is like stated above not part of the solution Λ^s as it is already known

Similar to our previous cases the integrand F depends on $A^{k,s}$ for $k \in [1, n-1]$, on $dA^{n,s}$ and on the derivatives of their entries. Thus we can use our previous general formulas for the Euler-Lagrange Equations and Boundary Conditions:

For each C_p with $p \in \mathrm{Idx}(C)$ and $C \in \Lambda^s$ we have

$$0 = F_{C_p} - \frac{\partial}{\partial x} F_{C_{p_x}} - \frac{\partial}{\partial y} F_{C_{p_y}}$$

and the Boundary Condition

$$\mathbf{n}^\top \begin{pmatrix} F_{C_{p_x}} \\ F_{C_{p_y}} \end{pmatrix} = 0$$

2. Compute $A^{n,s-1}$ by taking the sum $A^{n,s} + dA^{n,s}$ and by transferring the result to the next finer scale. [7]

3. Proceed with the next scale by setting $s = s - 1$

When we finally have reached the finest scale $s = 1$ we again perform the first step to obtain Λ^1. Then we compute our final solution as follows:

- For the Displacement Field we compute like in the first chapter the sum of $A^{n,1}$ and $dA^{n,1}$:

$$A^1 = A^{n,1} + dA^{n,1}$$

- For the remaining parts A^k with $k \neq n$ we have

$$A^k = A^{k,1}$$

Let us now investigate the Euler-Lagrange Equations and Boundary Conditions generated by this functional. Like in the previous sections we use the general formulas to obtain the types that occur.[8]

For the Euler-Lagrange Equations we have:

1. If $dA^{k,s}$ participates in both the Data Term and the Smoothness Term we obtain again two Euler-Lagrange Equations with a different structure than before

$$0 = \widehat{J}^s_{11}\, dA_1^{k,s} + \widehat{J}^s_{12}\, dA_2^{k,s} + \widehat{J}^s_{13} - \alpha\, \mathrm{div}\Big(D^G_\Psi(A^{k,s} + dA^{k,s})\, \nabla(A_1^{k,s} + dA_1^{k,s}) \Big)$$

$$0 = \widehat{J}^s_{12}\, dA_1^{k,s} + \widehat{J}^s_{22}\, dA_2^{k,s} + \widehat{J}^s_{23} - \alpha\, \mathrm{div}\Big(D^G_\Psi(A^{k,s} + dA^{k,s})\, \nabla(A_2^{k,s} + dA_2^{k,s}) \Big)$$

where

[7] In the discrete case we will also transfer the remaining solution parts to the next finer scale.

[8] These derivations work analogously to the previous cases and are omitted this time for the sake of brevity.

- the notation $\widehat{J}_p^s$ with $p \in \mathrm{Idx}(\widehat{J}^s)$ again indicates that the Motion Tensor entry is derived by using the images $f^s(x+A_1^{k,s}, y+A_2^{k,s}, t+1)$ and $f^s(x,y,t)$ and performing only a linearisation in the $x-$ and y-direction.
- and the Diffusion Tensor depends on both the known part $A^{k,s}$ and the unknown part $dA^{k,s}$ of the Displacement Field.

Once again these equations only occur in the special case of $n = 1$ for $dA^{1,s}$.

2. The two equations that are created by the participation of $dA^{k,s}$ in the Data Term and in the Agreement Term that also show a different structure than in the usual Generic Optic Flow case:

$$0 = \widehat{J}_{11}^s \; dA_1^{k,s} + \widehat{J}_{12}^s \; dA_2^{k,s} + \widehat{J}_{13}^s + \alpha \; \left(A_{11_x}^{k-1,s} - A_{1_{xx}}^{k,s} - dA_{1_{xx}}^{k,s} + A_{12_y}^{k-1,s} - A_{1_{yy}}^{k,s} - dA_{1_{yy}}^{k,s} \right)$$
$$0 = \widehat{J}_{12}^s \; dA_1^{k,s} + \widehat{J}_{22}^s \; dA_2^{k,s} + \widehat{J}_{23}^s + \alpha \; \left(A_{21_x}^{k-1,s} - A_{2_{xx}}^{k,s} - dA_{2_{xx}}^{k,s} + A_{22_y}^{k-1,s} - A_{2_{yy}}^{k,s} - dA_{2_{yy}}^{k,s} \right)$$

These equations only occur in the case of $dA^{n,s}$ for $n > 1$.

3. Two equations that are created by the participation of $A^{k,s}$ in the Agreement Term and in the Smoothness Term. Here we have to distinguish between two cases:
   - For $n = 2$ we have:
$$0 = A_{p1}^{k,s} - A_{p_x}^{k+1,s} - dA_{p_x}^{k+1,s} - \alpha \; \mathrm{div} \left(D_\Psi^G(A^{k,s}) \; \nabla A_{p1}^{k,s} \right)$$
$$0 = A_{p2}^{k,s} - A_{p_y}^{k+1,s} - dA_{p_y}^{k+1,s} - \alpha \; \mathrm{div} \left(D_\Psi^G(A^{k,s}) \; \nabla A_{p2}^{k,s} \right)$$
   - Otherwise the equations have the usual structure:
$$0 = A_{p1}^{k,s} - A_{p_x}^{k+1,s} - \alpha \; \mathrm{div} \left(D_\Psi^G(A^{k,s}) \; \nabla A_{p1}^{k,s} \right)$$
$$0 = A_{p2}^{k,s} - A_{p_y}^{k+1,s} - \alpha \; \mathrm{div} \left(D_\Psi^G(A^{k,s}) \; \nabla A_{p2}^{k,s} \right)$$

   where $p \in \mathrm{Idx}(A^{k+1,s})$. We always encounter this type in the case of $A^{1,s}$ for $n > 1$.

4. Two equations that are created by the participation of $A^{k,s}$ in two different Agreement Terms. Again we have to pay attention to two different cases:
   - For $n > 2$ we get in the case of $k = n - 1$:
$$0 = A_{p1}^{k,s} - A_{p_x}^{k+1,s} - dA_{p_x}^{k+1,s} + \alpha \; \left(A_{p11_x}^{k-1,s} - A_{p1_{xx}}^{k,s} + A_{p12_y}^{k-1,s} - A_{p1_{yy}}^{k,s} \right)$$
$$0 = A_{p2}^{k,s} - A_{p_y}^{k+1,s} - dA_{p_y}^{k+1,s} + \alpha \; \left(A_{p21_x}^{k-1,s} - A_{p2_{xx}}^{k,s} + A_{p22_y}^{k-1,s} - A_{p2_{yy}}^{k,s} \right)$$
   - Otherwise we obtain our familiar equations:
$$0 = A_{p1}^{k,s} - A_{p_x}^{k+1,s} + \alpha \; \left(A_{p11_x}^{k-1,s} - A_{p1_{xx}}^{k,s} + A_{p12_y}^{k-1,s} - A_{p1_{yy}}^{k,s} \right)$$
$$0 = A_{p2}^{k,s} - A_{p_y}^{k+1,s} + \alpha \; \left(A_{p21_x}^{k-1,s} - A_{p2_{xx}}^{k,s} + A_{p22_y}^{k-1,s} - A_{p2_{yy}}^{k,s} \right)$$

where $p \in \mathrm{Idx}(A^{k+1,s})$.

This type of equation again occurs for $A^{k,s}$ with $1 < k < n$ and $n > 2$.

And finally we have a look at the Boundary Conditions:

1. Boundary Conditions produced by the Smoothness Term: Once again we have to consider two cases:
   - For $k = n = 1$ we have:
   $$\mathbf{n}^\top \left(\nabla A_p^{k,s} + \nabla dA_p^{k,s} \right) = 0$$
   - Otherwise:
   $$\mathbf{n}^\top \nabla A_p^{k,s} = 0$$

   where $p \in \mathrm{Idx}(A^{k,s})$. These only occur for $A^{1,s}$ or if $n = 1$ for $dA^{1,s}$.

2. Boundary Conditions originating from the Agreement Term:
   - For $k = n$ where $n > 1$:
   $$\mathbf{n}^\top \begin{pmatrix} A_{p1}^{k-1,s} - A_{p_x}^{k,s} - dA_{p_x}^{k,s} \\ A_{p2}^{k-1,s} - A_{p_y}^{k,s} - dA_{p_y}^{k,s} \end{pmatrix} = 0$$
   - Otherwise:
   $$\mathbf{n}^\top \begin{pmatrix} A_{p1}^{k-1,s} - A_{p_x}^{k,s} \\ A_{p2}^{k-1,s} - A_{p_y}^{k,s} \end{pmatrix} = 0$$

   where $p \in \mathrm{Idx}(A^{k,s})$[9]. These occur for $A^{k,s}$ for $1 < k \leq n$ and $n > 1$. They emerge for $dA^{n,s}$ if $n > 1$.

This concludes our classification of the Euler-Lagrange Equations and Boundary Conditions in this case as well.

[9]Note that $dA^{n,s}$ and $A^{n,s}$ have the same Index Set.

3 Discrete Aspects

Until now we only worked in a continuous setting, that is, we used functions that were defined on a continuous domain Ω. Roughly speaking that means we were able to evaluate these functions *everywhere* inside this domain. *Real-World Data* on the other hand is unfortunately of discrete nature. This implies for example that we can only access the gray values of a given image at prescribed locations, the so-called sample locations. As we are planning to perform experiments with some discrete data in the next chapter we have to derive discrete versions of our models we introduced in the previous chapter.

We will start by discretising our domain itself and by introducing some basic notation. After that we will aim at finding discrete versions of our Euler-Lagrange Equations. Here will first deal with the ones in the Generic Image Restoration case. Then we proceed with equations of the Generic Optic Flow approach without Warping before we finally investigate the case of Optic Flow Estimation where we are using the Warping Strategy. During this discretisaion process we will observe that the discretisation of the Euler-Lagrange Equations will lead to large possibly nonlinear systems of equations that we have to solve in order to obtain a minimiser. Here we will employ a *fixed-point iteration*. Finally we will discuss some details of the Warping Strategy in the discrete setting.

3.1 Basic Discretisations and Notation

First we will consider our domains we used in the run of this work. In the Image Restoration case we worked with a spatial domain $\Omega \subset \mathbb{R}^2$ whereas in the Optic Flow case we extended it with a temporal component: $\Omega \times \mathbb{R}_0^+$. We first deal with the spatial one.

As our domain Ω is rectangular and bounded we can state explicitly its boundary coordinates:

- (a, c) is the coordinate of the lower left boundary point
- (b, d) is the coordinate of the upper right boundary point

Thus we have $\Omega =]a, b[\times]c, d[$ [1]. We discretise now Ω on a regular grid, where we take N samples in x-direction and M samples in y-direction. Thus we can compute the distance between two sample locations:

- $h_x := \frac{b-a}{N}$ is the distance between two samples along the x-axis

[1] Note that we have to use open intervals as our domain is open.

- $h_y := \frac{d-c}{M}$ is the distance between two samples along the y-axis

For simplicity we now assume that $h = h_x = h_y$.

Hence we can now define the coordinates of our sample locations:

$$x_{i,j} := (a + (i - \frac{1}{2})h, c + (j - \frac{1}{2})h)$$

with $i \in [1, N]$ and $j \in [1, M]$. We see that we are using a displacement of $\frac{1}{2}h$ in order to avoid having sample locations on the boundaries[2]. In order to simplify things even more we will additionally assume that $a = c = 0$ and we arrive at

$$x_{i,j} = ((i - \frac{1}{2})h, (j - \frac{1}{2})h)$$

Now that we know the locations of our sample points we can also evaluate expressions g at these positions:

- $[g]_{i,j} \approx g((i - \frac{1}{2})h, (j - \frac{1}{2})h)$ is the approximation of the expression g at the sample location $x_{i,j}$.

If g is for example our gray-value image f then $[f]_{i,j}$ will be an approximation of the gray-value at the sample location $x_{i,j}$.

Let us now focus on the spatio-temporal domain $\Omega \times \mathbb{R}_0^+$. Here we will start by replacing the unbounded temporal part of the domain by a bounded one: $\Omega \times]0, e[$. By performing now the discretisation like before we arrive at the spatio-temporal sample locations

$$x_{i,j,l} := ((i - \frac{1}{2})h, (j - \frac{1}{2})h, (l - \frac{1}{2})h_t)$$

with $i \in [1, N]$, $j \in [1, M]$ and $l \in [1, L]$ where L denotes our number of samples in the t-direction. It is common to set $h_t = 1$ (cf. [9]) in this case and thus:

$$x_{i,j,l} := ((i - \frac{1}{2})h, (j - \frac{1}{2})h, l - \frac{1}{2})$$

Again we introduce a notation for the evaluation of expressions g at these locations:

$$[g]_{i,j,l} \approx g((i - \frac{1}{2})h, (j - \frac{1}{2})h, l - \frac{1}{2})$$

So if g is our gray-value image sequence f then $[f]_{i,j,l}$ will be an approximation of the gray-value at the sample location $x_{i,j,l}$. We observe that if we hold l fixed we will gain access to the individual frames of our image sequence.

In the following we will assume that each frame of our sequence was presmoothed by applying a Gaussian Convolution with standard deviation σ to the original frame.[3]

In the next section we will now discuss more advanced expressions that are evaluated at the sample locations.

[2]Remember that Ω is open, thus the boundary does not belong to Ω.

[3]This improves the differentiability of our image frames in the discrete case.

3.2 Advanced Discretisations

In the previous section we evaluated so far only relatively easy expressions on our discretised domains. Now we turn to more sophisticated expressions that on the one hand involve derivatives of functions and on the other hand are subject to Boundary Conditions.

Data Term Contribution

We will first focus on the contributions that are generated in the Euler-Lagrange Equations by the Data Term.

Let us to this end have a look at the first type of Euler-Lagrange Equations that arises in the Generic Image Restoration setting:

$$0 = \underbrace{A^1 - f}_{\text{Data Term contribution}} + \alpha \operatorname{div}\left(D_\Psi^G(A^1) \nabla A^1 \right)$$

Here we can easily find a discrete version of the part originating from the Data Term by applying our previous knowledge:

$$\left[A^1 - f\right]_{i,j} = \left[A^1\right]_{i,j} - [f]_{i,j}$$

Now we turn to the first type that emerges in the Generic Optic Flow case:

$$0 = \overbrace{\widehat{J}_{11}\ A_1^1 + \widehat{J}_{12}\ A_2^1 + \widehat{J}_{13}}^{\text{Data Term contribution}} - \alpha \operatorname{div}\left(D_\Psi^G(A^1)\ \nabla A_1^1 \right)$$

$$0 = \underbrace{\widehat{J}_{12}\ A_1^1 + \widehat{J}_{22}\ A_2^1 + \widehat{J}_{23}}_{\text{Data Term contribution}} - \alpha \operatorname{div}\left(D_\Psi^G(A^1)\ \nabla A_2^1 \right)$$

Thus we discover that we have to find discrete versions of the Motion Tensor entries in this case. We remember that we are using a combination of the Brightness Constancy Assumption and the Gradient Constancy Assumption to construct our Robust Data Term. Their Motion Tensors are given by

$$J_{\text{GCA}} = \begin{pmatrix} f_{xx}f_{xx} + f_{xy}f_{xy} & f_{xy}f_{xx} + f_{yy}f_{xy} & f_{xt}f_{xx} + f_{yt}f_{yx} \\ f_{xy}f_{xx} + f_{yy}f_{xy} & f_{xy}f_{xy} + f_{yy}f_{yy} & f_{xt}f_{xy} + f_{yt}f_{yy} \\ f_{xt}f_{xx} + f_{yt}f_{yx} & f_{xt}f_{xy} + f_{yt}f_{yy} & f_{xt}f_{xt} + f_{yt}f_{yt} \end{pmatrix}$$

$$J_{\text{BCA}} = \begin{pmatrix} f_x^2 & f_x f_y & f_x f_t \\ f_x f_y & f_y^2 & f_y f_t \\ f_x f_t & f_y f_t & f_t^2 \end{pmatrix}$$

Hence, this time we are facing a more demanding task: We have to approximate derivatives at given sample locations $x_{i,j}$ in our domain.

In the following we summarise the finite differences that we use to approximate the arising derivatives.

For derivatives not including the temporal component we will use averaged central differences from the two frames l and $l+1$ that participate in the current motion estimation:

$$
\begin{aligned}
[f_x]_{i,j} &\approx \frac{1}{2}\frac{1}{2h}\sum_{i=0}^{1}\Big([f]_{i+1,j,l+i} - [f]_{i-1,j,l+i}\Big)\\
[f_y]_{i,j} &\approx \frac{1}{2}\frac{1}{2h}\sum_{i=0}^{1}\Big([f]_{i,j+1,l+i} - [f]_{i,j-1,l+i}\Big)\\
[f_{xy}]_{i,j} &\approx \frac{1}{2}\frac{1}{4h^2}\sum_{i=0}^{1}\Big(\Big([f]_{i+1,j+1,l+i} - [f]_{i-1,j+1,l+i}\Big) - \Big([f]_{i+1,j-1,l+i} - [f]_{i-1,j-1,l+i}\Big)\Big)\\
[f_{xx}]_{i,j} &\approx \frac{1}{2}\frac{1}{h^2}\sum_{i=0}^{1}\Big([f]_{i+1,j,l+i} - 2\,[f]_{i,j,l+i} + [f]_{i-1,j,l+i}\Big)\\
[f_{yy}]_{i,j} &\approx \frac{1}{2}\frac{1}{h^2}\sum_{i=0}^{1}\Big([f]_{i,j+1,l+i} - 2f_{i,j,l+i} + f_{i,j-1,l+i}\Big)
\end{aligned}
$$

In order to approximate the derivatives including the temporal component we will use central differences for the spatial components and simple forward differences for the temporal component:

$$
\begin{aligned}
[f_t]_{i,j} &\approx [f]_{i,j,l+1} - [f]_{i,j,l}\\
[f_{xt}]_{i,j} &\approx \frac{1}{2h}\Big(\Big([f]_{i+1,j,l+1} - [f]_{i-1,j,l+1}\Big) - \Big([f]_{i+1,j,l} - [f]_{i-1,j,l}\Big)\Big)\\
[f_{yt}]_{i,j} &\approx \frac{1}{2h}\Big(\Big([f]_{i,j+1,l+1} - [f]_{i,j-1,l+1}\Big) - \Big([f]_{i,j+1,l} - [f]_{i,j-1,l}\Big)\Big)
\end{aligned}
$$

Thus we can now state our discrete versions of the Motion Tensors:

$$
\begin{aligned}
[J_{\mathrm{GCA}}]_{i,j} = &\begin{pmatrix}
[f_{xx}]_{i,j}[f_{xx}]_{i,j} & [f_{xy}]_{i,j}[f_{xx}]_{i,j} & [f_{xt}]_{i,j}[f_{xx}]_{i,j}\\
[f_{xy}]_{i,j}[f_{xx}]_{i,j} & [f_{xy}]_{i,j}[f_{xy}]_{i,j} & [f_{xt}]_{i,j}[f_{xy}]_{i,j}\\
[f_{xt}]_{i,j}[f_{xx}]_{i,j} & [f_{xt}]_{i,j}[f_{xy}]_{i,j} & [f_{xt}]_{i,j}[f_{xt}]_{i,j}
\end{pmatrix}\\
&+\begin{pmatrix}
[f_{xy}]_{i,j}[f_{xy}]_{i,j} & [f_{yy}]_{i,j}[f_{xy}]_{i,j} & [f_{yt}]_{i,j}[f_{yx}]_{i,j}\\
[f_{yy}]_{i,j}[f_{xy}]_{i,j} & [f_{yy}]_{i,j}[f_{yy}]_{i,j} & [f_{yt}]_{i,j}[f_{yy}]_{i,j}\\
[f_{yt}]_{i,j}[f_{yx}]_{i,j} & [f_{yt}]_{i,j}[f_{yy}]_{i,j} & [f_{yt}]_{i,j}[f_{yt}]_{i,j}
\end{pmatrix}\\
[J_{\mathrm{BCA}}]_{i,j} = &\begin{pmatrix}
\left[f_x^2\right]_{i,j} & [f_x]_{i,j}[f_y]_{i,j} & [f_x]_{i,j}[f_t]_{i,j}\\
[f_x]_{i,j}[f_y]_{i,j} & \left[f_y^2\right]_{i,j} & [f_y]_{i,j}[f_t]_{i,j}\\
[f_x]_{i,j}[f_t]_{i,j} & [f_y]_{i,j}[f_t]_{i,j} & \left[f_t^2\right]_{i,j}
\end{pmatrix}
\end{aligned}
$$

Finally we obtain the discrete version of the Motion Tensor belonging to our Robust Data Term by

$$
\left[\widehat{J}_p\right]_{i,j} \approx \Psi'_D\left(\mathbf{w}_{i,j}^{\top}\,[J_{BCA}]_{i,j}\,\mathbf{w}_{i,j}\right)\left[J_{BCA_p}\right]_{i,j} + \gamma\ \Psi'_D\left(\mathbf{w}_{i,j}^{\top}\,[J_{GCA}]_{i,j}\,\mathbf{w}_{i,j}\right)\left[J_{GCA_p}\right]_{i,j}
$$

with $p \in \mathrm{Idx}(\widehat{J})$.

We remember that in the Warping Strategy, however, we are using the frames $f^s(x, y, t)$ and $f^s(x + A_1^{n,s}, y + A_2^{n,s}, t + 1$ on the scale s where $A^{n,s}$ is the known part of the Displacement Field to compute the Motion Tensors. Here we will use the notation

$$\left[\widehat{J}_p^s\right]_{i,j}$$

for the approximated Motion Tensor entries on the current scale.

Smoothness Term Contribution

By taking into account again the first type produced in the Optic Flow case

$$\begin{aligned} 0 &= \widehat{J}_{11}\ A_1^1 + \widehat{J}_{12}\ A_2^1 + \widehat{J}_{13} - \alpha\ \mathrm{div}\left(D_\Psi^G(A^1)\ \nabla A_1^1\right) \\ 0 &= \widehat{J}_{12}\ A_1^1 + \widehat{J}_{22}\ A_2^1 + \widehat{J}_{23} - \alpha\ \mathrm{div}\left(D_\Psi^G(A^1)\ \nabla A_2^1\right) \end{aligned}$$

we see that the contribution of the Smoothness Term, however, poses again a problem: We have to approximate derivatives on our discretised domain. Luckily expressions of the form

$$\mathrm{div}\left(D_\Psi^G(A)\nabla A_p\right)$$

where

- A is some tensor
- $p \in \mathrm{Idx}(A)$
- $D_\Psi^G(A)$ is our well-known diffusion tensor with

$$D_\Psi^G(A) = \Psi'\left(\sum_{p\in\mathrm{Idx}(A)} \nabla A_p \nabla A_p^\top\right)$$

arise very often in the field of Image Processing (cf. [9], [23], [18]) and thus discretisations of such expressions are well researched.

A common strategy works as follows:

1. Find a suitable approximation of the Diffusion Tensor in the point $x_{i,j}$:

$$\left[D_\Psi^G A\right]_{i,j} := \begin{pmatrix} a_{i,j} & b_{i,j} \\ b_{i,j} & c_{i,j} \end{pmatrix}$$

2. Finally find an approximation of

$$\left[\mathrm{div}\left(\left[D_\Psi^G(A)\right]_{i,j} \nabla A_p\right)\right]_{i,j}$$

We will first focus on the second case where we will restrict ourselves us to the usage of so-called 9-point schemes: Here the above expression is approximated by using the centre point $x_{i,j}$ and its surrounding 8 neighbors:

$$\Big[\operatorname{div}\Big(\big[D_\Psi^G(A)\big]_{i,j}\,\nabla A_p\Big)\Big]_{i,j} \approx \sum_{(\hat{x},\hat{y})\in\mathcal{N}} [\mathcal{D}]_{i,j}(\hat{x},\hat{y})\;\; [A_p]_{i+\hat{x},j+\hat{y}}$$

where $\mathcal{N} := [-1,1]\times[-1,1]$ denotes the set of the *relative coordinates* around the point $x_{i,j}$ and $[\mathcal{D}]_{i,j} : \mathcal{N} \to \mathbb{R}$ is a so-called *stencil* containing the weights for $[A_p]_{i,j}$ and its 8 neighbors. It has the following structure:

$[\mathcal{D}]_{i,j}(-1,1)$	$[\mathcal{D}]_{i,j}(0,1)$	$[\mathcal{D}]_{i,j}(1,1)$
$[\mathcal{D}]_{i,j}(-1,0)$	$[\mathcal{D}]_{i,j}(0,0)$	$[\mathcal{D}]_{i,j}(1,0)$
$[\mathcal{D}]_{i,j}(-1,-1)$	$[\mathcal{D}]_{i,j}(0,-1)$	$[\mathcal{D}]_{i,j}(1,-1)$

But we discover one issue here: We remember that we have to respect the given Boundary Conditions:

$$\mathbf{n}^\top \nabla A_p = 0$$

Before we proceed we introduce the notation for the *Characteristic Function* χ in order to integrate Boundary Conditions into our presented discretisations:

$$\chi_{[X]\times[Y]} := \chi_{[X]\times[Y]}(i,j) = \begin{cases} 1 & \text{if } i \in X \wedge j \in Y \\ 0 & \text{else} \end{cases}$$

Let us now focus on two possible methods that we will use for performing the aforementioned discretisation:

- The very common Standard Discretisation (cf. [23], [9])

  Like mentioned above we consider the approximation of the Diffusion Tensor $D_\Psi^G(A)$ at the sample location $x_{i,j}$. Then we can apply the the Standard Discretisation stencil depicted in 3.1. We see that it has pleasant and simple structure. But unfortunately it has some unpleasant drawbacks: It suffers from so-called dissipative artifacts and has a very poor rotational invariance(cf. [26]).

- The Family of Non-Standard Discretisations as presented in [26].

  In order to apply this novel approach we have to change our strategy a little bit: Instead of using an approximation in the point $x_{i,j}$ of our Diffusion Tensor we have

to perform this approximation in the location $x_{i+\frac{1}{2},j+\frac{1}{2}}$:

$$\left[D_\Psi^G A\right]_{i+\frac{1}{2},j+\frac{1}{2}} = \begin{pmatrix} [a]_{i+\frac{1}{2},j+\frac{1}{2}} & [b]_{i+\frac{1}{2},j+\frac{1}{2}} \\ [b]_{i+\frac{1}{2},j+\frac{1}{2}} & [c]_{i+\frac{1}{2},j+\frac{1}{2}} \end{pmatrix}$$

Then we can use the discretisation stencil that is shown in 3.2. We realise that this stencil provides a whole family of possible stencils that depend on the two parameters α_D and β_D. [26], however, pointed out that the following constraint has to be fulfilled to ensure the stability of numerical methods:

$$|\beta_D| \leq 1 - 2\alpha_D$$

Let us now focus on the role of these two parameters:

- If α_D is chosen very large the effect of dissipative artifacts of the disretisation is reduced and thus increases the "sharpness" of the stencil. It is, however, not avisable to choose the largest possible value $\frac{1}{2}$ as this choice could lead to systems of equations that are no longer irreducible, that is, the set of unknowns is divided into subsets that are not interacting with each other. This in turn leads to very undesirable checkerboard patterns in the results. (cf. [26])
- A large $|\beta_D|$ can help to improve the nonnegativity of the stencil, that is, it may reduce the number of negative weights outside the centre.

If we are using this type of discretisation we will always use $\alpha_D = 0.45$ and $\beta_D = 0$.[4]

Fortunately we see in both stencil variants that the Boundary Conditions are automatically fulfilled.

In the following we will use the Standard Discretisation only in cases where the Diffusion Tensor degenerates to the identity matrix I, that is, in cases where the Whittaker-Tikhonov penaliser Ψ_{WT} is used as Ψ. This implies that we do not need to find a suitable disretisation of the $D_\Psi^G(A)$ in this case as it is already given by I.

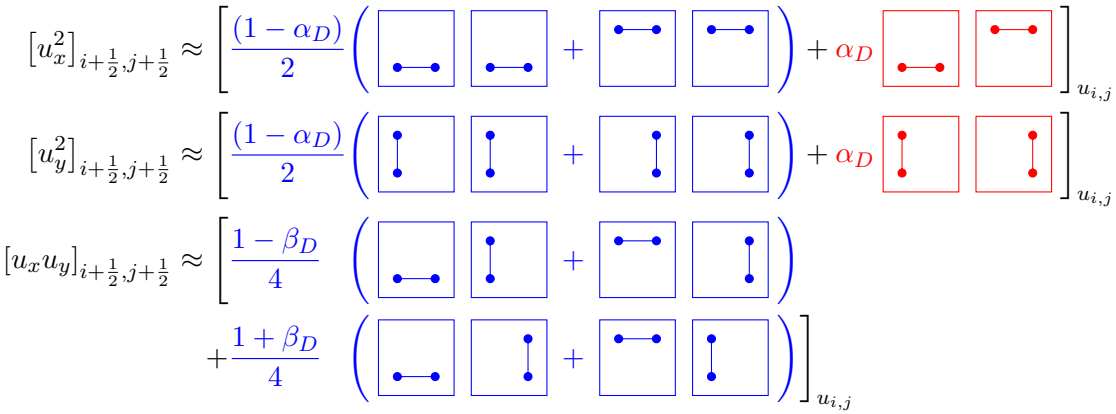
If we are using the Charbonnier penaliser Ψ_{C} in the Smoothness Term, however, we will make use of the Non-Standard Discretisation. But this time we then have to find a suitable discretisation of the associated Diffusion Tensor $D_\Psi^G(A)$ at the sample location $x_{i+\frac{1}{2},j+\frac{1}{2}}$. Here we will make use of the following strategy:

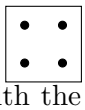
1. Apply the method presented in [26] to gain access to the Structure Tensors $\nabla A_p \nabla A_p^\top$ with $p \in \mathrm{Idx}(A)$ at the position $x_{i+\frac{1}{2},j+\frac{1}{2}}$. Let now for simplicity u be an entry of A. Then we can compute

$$\left[\nabla u \nabla u^\top\right]_{i+\frac{1}{2},j+\frac{1}{2}} = \begin{pmatrix} \left[u_x^2\right]_{i+\frac{1}{2},j+\frac{1}{2}} & [u_x u_y]_{i+\frac{1}{2},j+\frac{1}{2}} \\ [u_x u_y]_{i+\frac{1}{2},j+\frac{1}{2}} & \left[u_y^2\right]_{i+\frac{1}{2},j+\frac{1}{2}} \end{pmatrix}$$

[4] Thus we want some "sharpness" and pay no attention to the nonnegativity.

as follows



The points  inside $[\ldots]_{u_{i,j}}$ represent the four points $x_{i,j}$, $x_{i+1,j}$, $x_{i+1,j+1}$ and $x_{i,j+1}$ with the point $x_{i,j}$ located in the lower left. The lines connecting two points p and q indicate forward differences using the values u_p and u_q.[5]

We actually discover that the parameter α_D describes the ratio between the arithmetic and the geometric mean.[6]

2. Now we can evaluate $D^G_\Psi(A)$ at the sample location $x_{i+\frac{1}{2},j+\frac{1}{2}}$ by using the obtained approximations of the Structure Tensors:

$$\left[D^G_\Psi(A)\right]_{i+\frac{1}{2},j+\frac{1}{2}} \approx \Psi'\left(\sum_{p\in \mathrm{Idx}(A)}\left(\left[\nabla A_p \nabla A_p^\top\right]_{i+\frac{1}{2},j+\frac{1}{2}}\right)\right)$$

[5] Here we used the notation presented in [26].
[6] This is also the case in the presented stencil.

$$\frac{1}{2h^2}$$

$\chi_{[2,N]\times[1,M-1]}\left(-\frac{b_{i-1,j}+b_{i,j+1}}{2}\right)$	$\chi_{[1,N]\times[1,M-1]}\left(c_{i,j+1}+c_{i,j}\right)$	$\chi_{[1,N-1]\times[1,M-1]}\left(\frac{b_{i+1,j}+b_{i,j+1}}{2}\right)$
$\chi_{[2,N]\times[1,M]}\left(a_{i-1,j}+a_{i,j}\right)$	$-\Bigg(\chi_{[2,N]\times[1,M-1]}\left(-\frac{b_{i-1,j}+b_{i,j+1}}{2}\right)+\chi_{[1,N]\times[1,M-1]}\left(c_{i,j+1}+c_{i,j}\right)$ $+\chi_{[1,N-1]\times[1,M-1]}\left(\frac{b_{i+1,j}+b_{i,j+1}}{2}\right)+\chi_{[2,N]\times[1,M]}\left(a_{i-1,j}+a_{i,j}\right)$ $+\chi_{[1,N-1]\times[1,M]}\left(a_{i+1,j}+a_{i,j}\right)+\chi_{[2,N]\times[2,M]}\left(\frac{b_{i-1,j}+b_{i,j-1}}{2}\right)$ $+\chi_{[1,N]\times[2,M]}\left(c_{i,j-1}+c_{i,j}\right)+\chi_{[1,N-1]\times[2,M]}\left(-\frac{b_{i+1,j}+b_{i,j-1}}{2}\right)\Bigg)$	$\chi_{[1,N-1]\times[1,M]}\left(a_{i+1,j}+a_{i,j}\right)$
$\chi_{[2,N]\times[2,M]}\left(\frac{b_{i-1,j}+b_{i,j-1}}{2}\right)$	$\chi_{[1,N]\times[2,M]}\left(c_{i,j-1}+c_{i,j}\right)$	$\chi_{[1,N-1]\times[2,M]}\left(-\frac{b_{i+1,j}+b_{i,j-1}}{2}\right)$

Figure 3.1: The Standard Discretisation stencil.

$$\frac{1}{2h^2}$$

$\chi_{[2,N]\times[1,M-1]}\big((\beta_D-1)b_{i-\frac{1}{2},j+\frac{1}{2}} + \alpha_D\big(a_{i-\frac{1}{2},j+\frac{1}{2}} + c_{i-\frac{1}{2},j+\frac{1}{2}}\big)\big)$	$\chi_{[1,N-1]\times[1,M-1]}\big((1-\alpha_D)c_{i+\frac{1}{2},j+\frac{1}{2}} - \alpha_D a_{i+\frac{1}{2},j+\frac{1}{2}} - \beta_D b_{i+\frac{1}{2},j+\frac{1}{2}}\big)+$ $\chi_{[2,N]\times[1,M-1]}\big((1-\alpha_D)c_{i-\frac{1}{2},j+\frac{1}{2}} - \alpha_D a_{i-\frac{1}{2},j+\frac{1}{2}} - \beta_D b_{i-\frac{1}{2},j+\frac{1}{2}}\big)$	$\chi_{[1,N-1]\times[1,M-1]}\big((\beta_D+1)b_{i+\frac{1}{2},j+\frac{1}{2}} + \alpha_D\big(a_{i+\frac{1}{2},j+\frac{1}{2}} + c_{i+\frac{1}{2},j+\frac{1}{2}}\big)\big)$
$\chi_{[2,N]\times[1,M-1]}\big((1-\alpha_D)a_{i-\frac{1}{2},j+\frac{1}{2}} - \alpha_D c_{i-\frac{1}{2},j+\frac{1}{2}} - \beta_D b_{i-\frac{1}{2},j+\frac{1}{2}}\big)+$ $\chi_{[2,N]\times[2,M]}\big((1-\alpha_D)a_{i-\frac{1}{2},j-\frac{1}{2}} - \alpha_D c_{i-\frac{1}{2},j-\frac{1}{2}} - \beta_D b_{i-\frac{1}{2},j-\frac{1}{2}}\big)$	$-\chi_{[1,N-1]\times[1,M-1]}\big((1-\beta_D)b_{i+\frac{1}{2},j+\frac{1}{2}} + (1-\alpha_D)\big(a_{i+\frac{1}{2},j+\frac{1}{2}} + c_{i+\frac{1}{2},j+\frac{1}{2}}\big)\big)$ $-\chi_{[2,N]\times[1,M-1]}\big((-1-\beta_D)b_{i-\frac{1}{2},j+\frac{1}{2}} + (1-\alpha_D)\big(a_{i-\frac{1}{2},j+\frac{1}{2}} + c_{i-\frac{1}{2},j+\frac{1}{2}}\big)\big)$ $-\chi_{[1,N-1]\times[2,M]}\big((-1-\beta_D)b_{i+\frac{1}{2},j-\frac{1}{2}} + (1-\alpha_D)\big(a_{i+\frac{1}{2},j-\frac{1}{2}} + c_{i+\frac{1}{2},j-\frac{1}{2}}\big)\big)$ $-\chi_{[2,N]\times[2,M]}\big((1-\beta_D)b_{i-\frac{1}{2},j-\frac{1}{2}} + (1-\alpha_D)\big(a_{i-\frac{1}{2},j-\frac{1}{2}}c_{i-\frac{1}{2},j-\frac{1}{2}}\big)\big)$	$\chi_{[1,N-1]\times[1,M-1]}\big((1-\alpha_D)a_{i+\frac{1}{2},j+\frac{1}{2}} - \alpha_D c_{i+\frac{1}{2},j+\frac{1}{2}} - \beta_D b_{i+\frac{1}{2},j+\frac{1}{2}}\big)+$ $\chi_{[1,N-1]\times[2,M]}\big((1-\alpha_D)a_{i+\frac{1}{2},j-\frac{1}{2}} - \alpha_D c_{i+\frac{1}{2},j-\frac{1}{2}} - \beta_D b_{i+\frac{1}{2},j-\frac{1}{2}}\big)$
$\chi_{[2,N]\times[2,M]}\big((\beta_D+1)b_{i-\frac{1}{2},j-\frac{1}{2}} + \alpha_D\big(a_{i-\frac{1}{2},j-\frac{1}{2}} + c_{i-\frac{1}{2},j-\frac{1}{2}}\big)\big)$	$\chi_{[1,N-1]\times[2,M]}\big((1-\alpha_D)c_{i+\frac{1}{2},j-\frac{1}{2}} - \alpha_D a_{i+\frac{1}{2},j-\frac{1}{2}} - \beta b_{i+\frac{1}{2},j-\frac{1}{2}}\big)+$ $\chi_{[2,N]\times[2,M]}\big((1-\alpha_D)c_{i-\frac{1}{2},j-\frac{1}{2}} - \alpha_D a_{i-\frac{1}{2},j-\frac{1}{2}} - \beta_D b_{i-\frac{1}{2},j-\frac{1}{2}}\big)$	$\chi_{[1,N-1]\times[2,M]}\big((\beta_D-1)b_{i+\frac{1}{2},j-\frac{1}{2}} + \alpha_D\big(a_{i+\frac{1}{2},j-\frac{1}{2}} + c_{i+\frac{1}{2},j-\frac{1}{2}}\big)\big)$

Figure 3.2: The Family of Non-Standard Discretisations.

Again we consider briefly the Euler-Lagrange Equations that arised for $k = n = 1$ in the Warping Strategy:

$$0 = \widehat{J}^s_{11}\ dA^{1,s}_1 + \widehat{J}^s_{12}\ dA^{1,s}_2 + \widehat{J}^s_{13} - \alpha\ \text{div}\ \left(D^G_\Psi(A^{1,s} + dA^{1,s})\ \nabla(A^{1,s}_1 + dA^{1,s}_1) \right)$$
$$0 = \widehat{J}^s_{12}\ dA^{1,s}_1 + \widehat{J}^s_{22}\ dA^{1,s}_2 + \widehat{J}^s_{23} - \alpha\ \text{div}\ \left(D^G_\Psi(A^{1,s} + dA^{1,s})\ \nabla(A^{1,s}_2 + dA^{1,s}_2) \right)$$

We see that in this case our expression depends on the known part of the Displacement Field $A^{1,s}$ and also on the unknown part $dA^{1,s}$. Therefore we will use the following discretisation here instead:

$$\left[\text{div}\left(\left[D^G_\Psi(A^{1,s} + dA^{1,s})\right]_{i,j}\ \nabla(A^{1,s}_p + dA^{1,s}_p)\right)\right]_{i,j} \approx$$
$$\sum_{(\hat{x},\hat{y})\in\mathcal{N}} \left(\left[\mathcal{D}^s\right]_{i,j}(\hat{x},\hat{y})\ \left[A^{1,s}_p\right]_{i+\hat{x},j+\hat{y}} + \left[\mathcal{D}^s\right]_{i,j}(\hat{x},\hat{y})\ \left[dA^{1,s}_p\right]_{i+\hat{x},j+\hat{y}} \right)$$

with

- $p \in \text{Idx}(A^{1,s})$
- $(i,j) \in [1, N^s] \times [1, M^s\}$ where N^s and M^s are the dimensions of the current scale s
- $\mathcal{D}^s$ indicates that the stencil is computed by using the dimensions N^s and M^s.[7]

Furthermore this time the Diffusion Tensor D^G_Ψ depends on $A^{1,s}$ and $dA^{1,s}$, which means that we have to apply the following discretisation if we are using the Charbonnier penaliser:

$$\left[D^G_\Psi(A^{1,s} + dA^{1,s})\right]_{i+\frac{1}{2},j+\frac{1}{2}} \approx \Psi'\left(\sum_{p\in\text{Idx}(A^{1,s})} \left(\left[\nabla(A^{1,s}_p + dA^{1,s}_p)\ \nabla(A^{1,s}_p + dA^{1,s}_p)^\top\right]_{i+\frac{1}{2},j+\frac{1}{2}} \right) \right)$$

Agreement Term Contribution

Finally we will consider the contribution of the Agreement Term to the Euler-Lagrange Equations. To this end we have a look at the fourth type of equation arising:

$$0 = \overbrace{A^k_{p1} - A^{k+1}_{p_x}}^{\text{Contribution of Term 1}} + \overbrace{\alpha\ \left(A^{k-1}_{p11_x} - A^k_{p1_{xx}} + A^{k-1}_{p12_y} - A^k_{p1_{yy}} \right)}^{\text{Contribution of Term 2}}$$
$$0 = \underbrace{A^k_{p2} - A^{k+1}_{p_y}}_{\text{Contribution of Term 1}} + \underbrace{\alpha\ \left(A^{k-1}_{p21_x} - A^k_{p2_{xx}} + A^{k-1}_{p22_y} - A^k_{p2_{yy}} \right)}_{\text{Contribution of Term 2}}$$

where

[7] Remember that these dimensions are used in the characteristic function.

- $p \in \mathrm{Idx}(A^{k+1})$ as usual

This contribution of two Agreement Terms to two equations poses the most brutal challenge we have faced so far: Like before we have to approximate derivatives and at the same time we have to obey the following Boundary Conditions:

- Condition created by the first term:

$$\mathbf{n}^\top \begin{pmatrix} A^k_{p1} - A^{k+1}_{p_x} \\ A^k_{p2} - A^{k+1}_{p_y} \end{pmatrix} = 0$$

- Conditions created by the second term:

$$\mathbf{n}^\top \begin{pmatrix} A^{k-1}_{p11} - A^k_{p1_x} \\ A^{k-1}_{p12} - A^k_{p1_y} \end{pmatrix} = 0$$
$$\mathbf{n}^\top \begin{pmatrix} A^{k-1}_{p21} - A^k_{p2_x} \\ A^{k-1}_{p22} - A^k_{p2_y} \end{pmatrix} = 0$$

  which we can rewrite as

$$\mathbf{n}^\top \begin{pmatrix} A^{k-1}_{q1} - A^k_{q_x} \\ A^{k-1}_{q2} - A^k_{q_y} \end{pmatrix} = 0$$

  where $q \in \mathrm{Idx}(A^k)$.

In order to find suitable discretisations for

- the contribution of the first term:

$$A^k_{p1} - A^{k+1}_{p_x}$$
$$A^k_{p2} - A^{k+1}_{p_y}$$

- and the ones of the second term:

$$A^{k-1}_{p11_x} - A^k_{p1_{xx}} + A^{k-1}_{p12_y} - A^k_{p1_{yy}}$$
$$A^{k-1}_{p21_x} - A^k_{p2_{xx}} + A^{k-1}_{p22_y} - A^k_{p2_{yy}}$$

  which we may again rewrite as

$$A^{k-1}_{q1_x} - A^k_{q_{xx}} + A^{k-1}_{q2_y} - A^k_{q_{yy}}$$

  where $q \in \mathrm{Idx}(A^k)$.

we will use the method presented in [13].

First we introduce the energy

$$E(A^{c-1}, A^c) := \frac{1}{2} \int_\Omega \underbrace{\left(\left\| A^{c-1} - \nabla A^c \right\|_2^2 \right)}_{F} d\mathbf{x}$$

By investigating this functional we discover that its integrand F depends on the entries of A^{c-1} and the derivatives of the entries of A^c.

Thus we have the Euler-Lagrange Equations:

- For A^{c-1} we have

  $$0 = F_{A^{c-1}_{p1}}$$
  $$0 = F_{A^{c-1}_{p2}}$$

  where $p \in \mathrm{Idx}(A^c)$ and thus

  $$0 = A^{c-1}_{p1} - A^c_{p_x}$$
  $$0 = A^{c-1}_{p2} - A^c_{p_y}$$

  Here the right handside corresponds exactly to our contribution from the first term if we set $c = k + 1$.

- For each A^c_p with $p \in \mathrm{Idx}(A^c)$ we have

  $$0 = -\frac{\partial}{\partial x} F_{A^c_{p_x}} - \frac{\partial}{\partial y} F_{A^c_{p_y}}$$

  and thus

  $$0 = A^{c-1}_{p1_x} - A^c_{p_{xx}} + A^{c-1}_{p2_y} - A^c_{p_{yy}}$$

  This time the right handside corresponds to the contribution from the second term if we set $c = k$.

As Boundary Conditions we have for each A^c_p:

$$\mathbf{n}^\top \begin{pmatrix} A^{c-1}_{p1} - A^c_{p_x} \\ A^{c-1}_{p2} - A^c_{p_y} \end{pmatrix} = 0$$

We notice that we get the Boundary Conditions from the first term by setting $c = k$ and the ones from the second term by setting $c = k + 1$.

In the second step a discrete version of this functional is introduced that obeys these Boundary Conditions:

$$E^D(\cdot) := \frac{1}{2} \sum_{i=1}^{N-1} \sum_{j=1}^{M} \sum_{p \in \mathrm{Idx}(A^c)} \left(\left[A^{c-1}_{p1} \right]_{i,j} - \frac{\left[A^c_p \right]_{i+1,j} - \left[A^c_p \right]_{i,j}}{h} \right)^2$$
$$+ \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{M-1} \sum_{p \in \mathrm{Idx}(A^c)} \left(\left[A^{c-1}_{p2} \right]_{i,j} - \frac{\left[A^c_p \right]_{i,j+1} - \left[A^c_p \right]_{i,j}}{h} \right)^2$$

where E^D now depends on the vector consisting of the following entries:

- $\left[A_p^c\right]_{i,j}$ with $p \in \mathrm{Idx}(A^c)$, $(i,j) \in [1,N] \times [1,M]$ and
- $\left[A_q^{c-1}\right]_{i,j}$ with $q \in \mathrm{Idx}(A^{c-1}) = \mathrm{Idx}(A^c) \times \{1,2\}$, $(i,j) \in [1,N] \times [1,M]$

By taking now the derivative with respect to $\left[A_p^c\right]_{i,j}$ we receive our desired approximation of the contribution of the second term:

$$\frac{\partial E^D}{\partial \left[A_p^c\right]_{i,j}} =$$

$$\chi_{[1,N-1]\times[1,M]}\left(\frac{\left[A_{p1}^{c-1}\right]_{i,j}}{h} - \frac{\left[A_p^c\right]_{i+1,j} - \left[A_p^c\right]_{i,j}}{h^2}\right) -$$

$$\chi_{[2,N]\times[1,M]}\left(\frac{\left[A_{p1}^{c-1}\right]_{i-1,j}}{h} - \frac{\left[A_p^c\right]_{i,j} - \left[A_p^c\right]_{i-1,j}}{h^2}\right) +$$

$$\chi_{[1,N]\times[1,M-1]}\left(\frac{\left[A_{p2}^{c-1}\right]_{i,j}}{h} - \frac{\left[A_p^c\right]_{i,j+1} - \left[A_p^c\right]_{i,j}}{h^2}\right) -$$

$$\chi_{[1,N]\times[2,M]}\left(\frac{\left[A_{p2}^{c-1}\right]_{i,j-1}}{h} - \frac{\left[A_p^c\right]_{i,j} - \left[A_p^c\right]_{i,j-1}}{h^2}\right)$$

$$\approx \left[A_{p1_x}^{c-1} - A_{p_{xx}}^c + A_{p2_y}^{c-1} - A_{p_{yy}}^c\right]_{i,j}$$

Taking now the derivatives with respect to $\left[A_{p1}^{c-1}\right]_{i,j}$ and $\left[A_{p2}^{c-1}\right]_{i,j}$ leads to the approximations of the contribution of the second term:

$$\frac{\partial E^D}{\partial \left[A_{p1}^{c-1}\right]_{i,j}} =$$

$$\chi_{[1,N-1]\times[1,M]}\left(\left[A_{p1}^{c-1}\right]_{i,j} - \frac{\left[A_p^c\right]_{i+1,j} - \left[A_p^c\right]_{i,j}}{h}\right) \approx \left[A_{p1}^{c-1} - A_{p_x}^c\right]_{i,j}$$

$$\frac{\partial E^D}{\partial \left[A_{p2}^{c-1}\right]_{i,j}} =$$

$$\chi_{[1,N]\times[1,M-1]}\left(\left[A_{p2}^{c-1}\right]_{i,j} - \frac{\left[A_p^c\right]_{i,j+1} - \left[A_p^c\right]_{i,j}}{h}\right) \approx \left[A_{p2}^{c-1} - A_{p_y}^c\right]_{i,j}$$

Thus we now have access to our wanted discretisations that also obey the given Boundary Conditions.

Again we visit briefly the Warping case. Here we can use a similar approach to obtain discretisations of equations where the unknown part of the Displacement Field $dA^{n,s}$ is involved:

$$\begin{aligned}
&\left[A^{n-1,s}_{p1_x} - A^{n,s}_{p_{xx}} - dA^{n,s}_{p_{xx}} + A^{n-1,s}_{p2_y} - A^{n,s}_{p_{yy}} - dA^{n,s}_{p_{yy}}\right]_{i,j} \approx \\
&\quad \chi_{[1,N^s-1]\times[1,M^s]}\left(\frac{\left[A^{n-1,s}_{p1}\right]_{i,j}}{h} - \frac{[A^{n,s}_p]_{i+1,j} - [A^{n,s}_p]_{i,j}}{h^2} - \frac{[dA^{n,s}_p]_{i+1,j} - [dA^{n,s}_p]_{i,j}}{h^2}\right) \\
&\quad - \chi_{[2,N^s]\times[1,M^s]}\left(\frac{\left[A^{n-1,s}_{p1}\right]_{i-1,j}}{h} - \frac{[A^{n,s}_p]_{i,j} - [A^{n,s}_p]_{i-1,j}}{h^2} - \frac{[dA^{n,s}_p]_{i,j} - [dA^{n,s}_p]_{i-1,j}}{h^2}\right) \\
&\quad + \chi_{[1,N^s]\times[1,M^s-1]}\left(\frac{\left[A^{n-1,s}_{p2}\right]_{i,j}}{h} - \frac{[A^{n,s}_p]_{i,j+1} - [A^{n,s}_p]_{i,j}}{h^2} - \frac{[dA^{n,s}_p]_{i,j+1} - [dA^{n,s}_p]_{i,j}}{h^2}\right) \\
&\quad - \chi_{[1,N^s]\times[2,M^s]}\left(\frac{\left[A^{n-1,s}_{p2}\right]_{i,j-1}}{h} - \frac{[A^{n,s}_p]_{i,j} - [A^{n,s}_p]_{i,j-1}}{h^2} - \frac{[dA^{n,s}_p]_{i,j} - [dA^{n,s}_p]_{i,j-1}}{h^2}\right)
\end{aligned}$$

$$\begin{aligned}
&\left[A^{n-1,s}_{p1} - A^{n,s}_{p_x} - dA^{n,s}_{p_x}\right]_{i,j} \approx \\
&\quad \chi_{[1,N^s-1]\times[1,M^s]}\left(\left[A^{n-1,s}_{p1}\right]_{i,j} - \frac{[A^{n,s}_p]_{i+1,j} - [A^{n,s}_p]_{i,j}}{h} - \frac{[dA^{n,s}_p]_{i+1,j} - [dA^{n,s}_p]_{i,j}}{h}\right)
\end{aligned}$$

$$\begin{aligned}
&\left[A^{n-1,s}_{p2} - A^{n,s}_{p_y} - dA^{n,s}_{p_y}\right]_{i,j} \approx \\
&\quad \chi_{[1,N^s]\times[1,M^s-1]}\left(\left[A^{n-1,s}_{p2}\right]_{i,j} - \frac{[A^{n,s}_p]_{i,j+1} - [A^{n,s}_p]_{i,j}}{h} - \frac{[dA^{n,s}_p]_{i,j+1} - [dA^{n,s}_p]_{i,j}}{h}\right)
\end{aligned}$$

where $p \in \mathrm{Idx}(A^{n,s})$.

Thus we have now finally collected all ingredients needed to discretise our Euler-Lagrange Equations.

3.3 Preparations for the Fixed-Point Schemes

Before we poceed with the discretisation of the equations and the setup of our fixed-point step schemes we have to make some preparations.

First of all we make the following obervations:

- By discretising our Euler-Lagrange Equations in every point of our discrete domain we will gain access to a very large system of equations. Furthermore we discover that our unknowns can be arranged into sets where each set belongs to a specific part of our solution A^k with $k \in [1, n]$.

- The arising system of equations is possibly nonlinear if a Robust Data Term is used or if the Charbonnier penaliser is involved

Our first observation implies that the structure of our system of equations depends on the choice of n, the desired smoothness order. Moreover we realise that these sets of unknowns will cause our system to consist of individual blocks. The structure of such a block is then determined by the Euler-Lagrange Equation type that is generated by the solution part A^k the corresponding set belongs to. These facts actually imply that we have to setup a fixed-point iteration scheme for all arising block structures.

By considering the second observation we learn that we have to choose a appropriate solving technique for nonlinear systems.

A common strategy for solving such nonlinear systems of equations is for example *Kačanov's method* (cf. [23]).

Roughly speaking it repeats the following steps until a solution is found:

1. Evaluate the nonlinear expressions for our current approximation of the solution and hold these expressions fixed. This procedure turns the nonlinear system into a linear system of equations.

2. Apply a standard solver like for example *Gauss-Seidel* to solve the arising linear system, which provides us with a new estimate of the solution.

3. Return to step 1 if we have not found the solution yet.

This means that we have actually two two-fixed point iterations: An outer one for evaluating the nonlinearities and an inner one for solving the linear systems.

We will use a more modern technique that has been successfully used for such problems, the so-called method of *Frozen Coefficients* (cf. [8], [7], [16]). It can be interpreted as a modification of Kačanov's method where the second step is replaced by

2. Perform only one iteration of the chosen solver on the linear system

So we are only using a very coarse estimate of the solution of the current linear system to update the nonlinearities. We notice that we have this time only one fixed-point iteration that always evaluates the nonlinearities at the previous iteration step. Moreover we discover that if we have no nonlinearities this method will boil down to an usual fixed-point iteration for linear systems, which means that we may use it for both types of systems.

As base solver for those linear systems we will use extrapolating variants of different Gauss-Seidel solvers. The extrapolation parameter ω should be chosen like in the case of *Successive Over-relaxation* (cf. [17]):

$$\omega \in [1, 2[$$

Finally we give some conventions and notations we use for the fixed-point steps:

- At the beginning of our current iteration step we set $k = n$ and repeat the following two steps until $k = 0$:
  1. Perform the fixed-point iteration step on the unknowns set belonging to A^k
  2. Set $k = k - 1$

  Hence we are always beginning with the unknowns set interacting directly with the data.
- In an unknowns set belonging to A^k we process the entries A_p^k with $p \in \mathrm{Idx}(A_p^k)$ in a lexicographical order. An exception is the Displacement Field: Here we will process both entries simultaneously.
- The individual unknowns $\left[A_p^k\right]_{i,j}$ are then also visited in a lexicographical order
- Thus we can split our set of relative neighborhood coordinates $\mathcal{N}^0 := \mathcal{N} \backslash \{(0,0)\}$ into
  - the set of already visited neighbors $\mathcal{N}^-$ and
  - the set $\mathcal{N}^+$ containing the neighbors that have not been processed yet
- We use the notation

  $$\left[A_p^{k,m}\right]_{i,j}$$

  to denote the value of the unknown $\left[A_p^k\right]_{i,j}$ at the iteration step m. This notation is extended in the Warping case to

  $$\left[A_p^{k,m,s}\right]_{i,j}$$

  where s represents the current scale we are working on.
- For the first iteration step $m = 1$ we are using for each unknown an arbitrary initialisation. In the case of Warping, however, we will alter this convention a little bit as we will see later on.
- The same notation as above is also employed to signal that a quantity $[G]_{i,j}$ should be evaluated with values from the specific iteration step m:

  $$[G^{m,s}]_{i,j}$$

  In the following G is either our stencil $\mathcal{D}$ or a Motion Tensor Entry $\widehat{J}_p$ with $p \in \mathrm{Idx}(\widehat{J})$.

  In the Warping case the extended notation is used:

  $$[G^{m,s}]_{i,j}$$

  to signal that we are on the scale s.

3.4 Fixed-Point Schemes for Image Restoration

Let us now state the fixed-point schemes for the different block types arising in the Generic Image Restoration case.

We begin with the first type where $k = n = 1$. Here the Euler-Lagrange Equation is given by:

$$0 = A^1 - f + \alpha \operatorname{div}\Big(D_\Psi^G(A^1) \nabla A^1 \Big)$$

Applying our discretisation yields:

$$0 = \left[A^1\right]_{i,j} - [f]_{i,j} - \alpha \sum_{(\hat{x},\hat{y})\in\mathcal{N}} [\mathcal{D}]_{i,j}(\hat{x},\hat{y}) \ \left[A^1\right]_{i+\hat{x},j+\hat{y}}$$

And thus we are able to state the fixed-point step:

$$\begin{aligned}\left[A^{1,m+1}\right]_{i,j} = (1-\omega)A^{1,m} + \omega \left(1 - \alpha \ [\mathcal{D}^m]_{i,j}(0,0)\right)^{-1} \cdot \\ \Bigg([f]_{i,j} + \alpha \Bigg(\sum_{(\hat{x},\hat{y})\in\mathcal{N}^-} [\mathcal{D}^m]_{i,j}(\hat{x},\hat{y}) \ \left[A^{1,m+1}\right]_{i+\hat{x},j+\hat{y}} + \\ \sum_{(\hat{x},\hat{y})\in\mathcal{N}^+} [\mathcal{D}^m]_{i,j}(\hat{x},\hat{y}) \ \left[A^{1,m}\right]_{i+\hat{x},j+\hat{y}} \Bigg) \Bigg)\end{aligned}$$

Now we continue with the second type that arises for $k = n$ if $n > 1$:

$$0 = A^n - f + \alpha \left(A^{n-1}_{1_x} - A^n_{xx} + A^{n-1}_{2_y} - A^n_{yy} \right)$$

Performing the discretisation yields:

$$\begin{aligned}0 = [A^n]_{i,j} - [f]_{i,j} + \alpha \Bigg(\\ \chi_{[1,N-1]\times[1,M]} \Bigg(\frac{\left[A_1^{n-1}\right]_{i,j}}{h} - \frac{[A^n]_{i+1,j} - [A^n]_{i,j}}{h^2} \Bigg) - \\ \chi_{[2,N]\times[1,M]} \Bigg(\frac{\left[A_1^{n-1}\right]_{i-1,j}}{h} - \frac{[A^n]_{i,j} - [A^n]_{i-1,j}}{h^2} \Bigg) + \\ \chi_{[1,N]\times[1,M-1]} \Bigg(\frac{\left[A_2^{n-1}\right]_{i,j}}{h} - \frac{[A^n]_{i,j+1} - [A^n]_{i,j}}{h^2} \Bigg) - \\ \chi_{[1,N]\times[2,M]} \Bigg(\frac{\left[A_2^{n-1}\right]_{i,j-1}}{h} - \frac{[A^n]_{i,j} - [A^n]_{i,j-1}}{h^2} \Bigg) \Bigg)\end{aligned}$$

and thus we have as fixed-point step

$$\begin{aligned}
\left[A^{n,m+1}\right]_{i,j} &= (1-\omega)\left[A^{n,m}\right]_{i,j} + \\
&\omega\left(1+\frac{\alpha}{h^2}\left(\chi_{[1,N-1]\times[1,M]}+\chi_{[2,N]\times[1,M]}+\chi_{[1,N]\times[1,M-1]}+\chi_{[1,N]\times[2,M]}\right)\right)^{-1}\cdot \\
&\Bigg([f]_{i,j} - \alpha\Bigg(\chi_{[1,N-1]\times[1,M]}\Bigg(\frac{\left[A_1^{n-1,m}\right]_{i,j}}{h}-\frac{\left[A^{n,m}\right]_{i+1,j}}{h^2}\Bigg) \\
&\quad -\chi_{[2,N]\times[1,M]}\Bigg(\frac{\left[A_1^{n-1,m}\right]_{i-1,j}}{h}+\frac{\left[A^{n,m+1}\right]_{i-1,j}}{h^2}\Bigg) \\
&\quad +\chi_{[1,N]\times[1,M-1]}\Bigg(\frac{\left[A_2^{n-1,m}\right]_{i,j}}{h}-\frac{\left[A^{n,m}\right]_{i,j+1}}{h^2}\Bigg) \\
&\quad -\chi_{[1,N]\times[2,M]}\Bigg(\frac{\left[A_2^{n-1,m}\right]_{i,j-1}}{h}+\frac{\left[A^{n,m+1}\right]_{i,j-1}}{h^2}\Bigg)\Bigg)\Bigg)
\end{aligned}$$

Now we consider the third type that emerges for A^1 in the case of $n > 1$:

$$\begin{aligned}
0 &= A^1_{p1} - A^2_{p_x} - \alpha\ \mathrm{div}\Big(D^G_\Psi(A^1)\nabla A^1_{p1}\Big) \\
0 &= A^1_{p2} - A^2_{p_y} - \alpha\ \mathrm{div}\Big(D^G_\Psi(A^1)\nabla A^1_{p2}\Big)
\end{aligned}$$

where $p \in \mathrm{Idx}(A^2)$.

Performing the discretisation gives:

$$\begin{aligned}
0 &= \chi_{[1,N-1]\times[1,M]}\Bigg(\left[A^1_{p1}\right]_{i,j}-\frac{\left[A^2_p\right]_{i+1,j}-\left[A^2_p\right]_{i,j}}{h}\Bigg)-\alpha\sum_{(\hat{x},\hat{y})\in\mathcal{N}}[\mathcal{D}]_{i,j}(\hat{x},\hat{y})\ \left[A^1_{p1}\right]_{i+\hat{x},j+\hat{y}} \\
0 &= \chi_{[1,N]\times[1,M-1]}\Bigg(\left[A^1_{p2}\right]_{i,j}-\frac{\left[A^2_p\right]_{i,j+1}-\left[A^2_p\right]_{i,j}}{h}\Bigg)-\alpha\sum_{(\hat{x},\hat{y})\in\mathcal{N}}[\mathcal{D}]_{i,j}(\hat{x},\hat{y})\ \left[A^1_{p2}\right]_{i+\hat{x},j+\hat{y}}
\end{aligned}$$

As we have two equations we also get two fixed-point steps:

$$\left[A_{p1}^{1,m+1}\right]_{i,j} = (1-\omega)\left[A_{p1}^{1,m}\right]_{i,j} + \omega\left(\chi_{[1,N-1]\times[1,M]} - \alpha\ \left[\mathcal{D}^m\right]_{i,j}(0,0)\right)^{-1} \cdot \left(\chi_{[1,N-1]\times[1,M]}\left(\frac{\left[A_p^{2,m+1}\right]_{i+1,j} - \left[A_p^{2,m+1}\right]_{i,j}}{h}\right) + \alpha\left(\sum_{(\hat{x},\hat{y})\in\mathcal{N}^-}\left[\mathcal{D}^m\right]_{i,j}(\hat{x},\hat{y})\ \left[A_{p1}^{1,m+1}\right]_{i+\hat{x},j+\hat{y}} + \sum_{(\hat{x},\hat{y})\in\mathcal{N}^+}\left[\mathcal{D}^m\right]_{i,j}(\hat{x},\hat{y})\ \left[A_{p1}^{1,m}\right]_{i+\hat{x},j+\hat{y}}\right)\right)$$

$$\left[A_{p2}^{1,m+1}\right]_{i,j} = (1-\omega)\left[A_{p2}^{1,m}\right]_{i,j} + \omega\left(\chi_{[1,N]\times[1,M-1]} - \alpha\ \left[\mathcal{D}^m\right]_{i,j}(0,0)\right)^{-1} \cdot \left(\chi_{[1,N]\times[1,M-1]}\left(\frac{\left[A_p^{2,m+1}\right]_{i,j+1} - \left[A_p^{2,m+1}\right]_{i,j}}{h}\right) + \alpha\left(\sum_{(\hat{x},\hat{y})\in\mathcal{N}^-}\left[\mathcal{D}^m\right]_{i,j}(\hat{x},\hat{y})\ \left[A_{p2}^{1,m+1}\right]_{i+\hat{x},j+\hat{y}} + \sum_{(\hat{x},\hat{y})\in\mathcal{N}^+}\left[\mathcal{D}^m\right]_{i,j}(\hat{x},\hat{y})\ \left[A_{p2}^{1,m}\right]_{i+\hat{x},j+\hat{y}}\right)\right)$$

Finally we turn to the fourth type that is present for $1 < k < n$ for $n > 2$:

$$0 = A_{p1}^k - A_{p_x}^{k+1} + \alpha\ \left(A_{p11_x}^{k-1} - A_{p1_{xx}}^k + A_{p12_y}^{k-1} - A_{p1_{yy}}^k\right)$$
$$0 = A_{p2}^k - A_{p_y}^{k+1} + \alpha\ \left(A_{p21_x}^{k-1} - A_{p2_{xx}}^k + A_{p22_y}^{k-1} - A_{p2_{yy}}^k\right)$$

where $p \in \mathrm{Idx}(A^{k+1})$.

The discretisation is given by:

$$
\begin{aligned}
0 = \chi_{[1,N-1]\times[1,M]}\Bigg(\quad \left[A^k_{p1}\right]_{i,j} - \frac{\left[A^{k+1}_{p}\right]_{i+1,j} - \left[A^{k+1}_{p}\right]_{i,j}}{h}\Bigg) + \\
\alpha\ \Bigg(\chi_{[1,N-1]\times[1,M]}\Bigg(\ \frac{\left[A^{k-1}_{p11}\right]_{i,j}}{h} - \frac{\left[A^{k}_{p1}\right]_{i+1,j} - \left[A^{k}_{p1}\right]_{i,j}}{h^2}\Bigg) - \\
\chi_{[2,N]\times[1,M]}\Bigg(\frac{\left[A^{k-1}_{p11}\right]_{i-1,j}}{h} - \frac{\left[A^{k}_{p1}\right]_{i,j} - \left[A^{k}_{p1}\right]_{i-1,j}}{h^2}\Bigg) + \\
\chi_{[1,N]\times[1,M-1]}\Bigg(\ \frac{\left[A^{k-1}_{p12}\right]_{i,j}}{h} - \frac{\left[A^{k}_{p1}\right]_{i,j+1} - \left[A^{k}_{p1}\right]_{i,j}}{h^2}\Bigg) - \\
\chi_{[1,N]\times[2,M]}\Bigg(\frac{\left[A^{k-1}_{p12}\right]_{i,j-1}}{h} - \frac{\left[A^{k}_{p1}\right]_{i,j} - \left[A^{k}_{p1}\right]_{i,j-1}}{h^2}\Bigg)\Bigg)
\end{aligned}
$$

$$
\begin{aligned}
0 = \chi_{[1,N]\times[1,M-1]}\Bigg(\quad \left[A^k_{p2}\right]_{i,j} - \frac{\left[A^{k+1}_{p}\right]_{i,j+1} - \left[A^{k+1}_{p}\right]_{i,j}}{h}\Bigg) + \\
\alpha\ \Bigg(\chi_{[1,N-1]\times[1,M]}\Bigg(\ \frac{\left[A^{k-1}_{p21}\right]_{i,j}}{h} - \frac{\left[A^{k}_{p2}\right]_{i+1,j} - \left[A^{k}_{p2}\right]_{i,j}}{h^2}\Bigg) - \\
\chi_{[2,N]\times[1,M]}\Bigg(\frac{\left[A^{k-1}_{p21}\right]_{i-1,j}}{h} - \frac{\left[A^{k}_{p2}\right]_{i,j} - \left[A^{k}_{p2}\right]_{i-1,j}}{h^2}\Bigg) + \\
\chi_{[1,N]\times[1,M-1]}\Bigg(\ \frac{\left[A^{k-1}_{p22}\right]_{i,j}}{h} - \frac{\left[A^{k}_{p2}\right]_{i,j+1} - \left[A^{k}_{p2}\right]_{i,j}}{h^2}\Bigg) - \\
\chi_{[1,N]\times[2,M]}\Bigg(\frac{\left[A^{k-1}_{p22}\right]_{i,j-1}}{h} - \frac{\left[A^{k}_{p2}\right]_{i,j} - \left[A^{k}_{p2}\right]_{i,j-1}}{h^2}\Bigg)\Bigg)
\end{aligned}
$$

These equations lead to the fixed-point steps:

$$\left[A_{p1}^{k,m+1}\right]_{i,j} = (1-\omega)\left[A_{p1}^{k,m}\right]_{i,j} +$$
$$\omega\left(\chi_{[1,N-1]\times[1,M]} + \frac{\alpha}{h^2}\left(\chi_{[1,N-1]\times[1,M]} + \chi_{[2,N]\times[1,M]} + \chi_{[1,N]\times[1,M-1]} + \chi_{[1,N]\times[2,M]}\right)\right)^{-1}\cdot$$
$$\left(\chi_{[1,N-1]\times[1,M]}\left(\frac{\left[A_p^{k+1,m+1}\right]_{i+1,j} - \left[A_p^{k+1,m+1}\right]_{i,j}}{h}\right) -\right.$$
$$\alpha\left(\chi_{[1,N-1]\times[1,M]}\left(\frac{\left[A_{p11}^{k-1,m}\right]_{i,j}}{h} - \frac{\left[A_{p1}^{k,m}\right]_{i+1,j}}{h^2}\right) -\right.$$
$$\chi_{[2,N]\times[1,M]}\left(\frac{\left[A_{p11}^{k-1,m}\right]_{i-1,j}}{h} + \frac{\left[A_{p1}^{k,m+1}\right]_{i-1,j}}{h^2}\right) +$$
$$\chi_{[1,N]\times[1,M-1]}\left(\frac{\left[A_{p12}^{k-1,m}\right]_{i,j}}{h} - \frac{\left[A_{p1}^{k,m}\right]_{i,j+1}}{h^2}\right) -$$
$$\left.\left.\chi_{[1,N]\times[2,M]}\left(\frac{\left[A_{p12}^{k-1,m}\right]_{i,j-1}}{h} + \frac{\left[A_{p1}^{k,m+1}\right]_{i,j-1}}{h^2}\right)\right)\right)$$

$$\left[A_{p2}^{k,m+1}\right]_{i,j} = (1-\omega)\left[A_{p2}^{k,m}\right]_{i,j} +$$
$$\omega\left(\chi_{[1,N]\times[1,M-1]} + \frac{\alpha}{h^2}\left(\chi_{[1,N-1]\times[1,M]} + \chi_{[2,N]\times[1,M]} + \chi_{[1,N]\times[1,M-1]} + \chi_{[1,N]\times[2,M]}\right)\right)^{-1}\cdot$$
$$\left(\chi_{[1,N]\times[1,M-1]}\left(\frac{\left[A_p^{k+1,m+1}\right]_{i,j+1} - \left[A_p^{k+1,m+1}\right]_{i,j}}{h}\right) -\right.$$
$$\alpha\left(\chi_{[1,N-1]\times[1,M]}\left(\frac{\left[A_{p21}^{k-1,m}\right]_{i,j}}{h} - \frac{\left[A_{p2}^{k,m}\right]_{i+1,j}}{h^2}\right) -\right.$$
$$\chi_{[2,N]\times[1,M]}\left(\frac{\left[A_{p21}^{k-1,m}\right]_{i-1,j}}{h} + \frac{\left[A_{p2}^{k,m+1}\right]_{i-1,j}}{h^2}\right) +$$
$$\chi_{[1,N]\times[1,M-1]}\left(\frac{\left[A_{p22}^{k-1,m}\right]_{i,j}}{h} - \frac{\left[A_{p2}^{k,m}\right]_{i,j+1}}{h^2}\right) -$$
$$\left.\left.\chi_{[1,N]\times[2,M]}\left(\frac{\left[A_{p22}^{k-1,m}\right]_{i,j-1}}{h} + \frac{\left[A_{p2}^{k,m+1}\right]_{i,j-1}}{h^2}\right)\right)\right)$$

Thus we are finished with the fixed-point schemes that are needed for the Generic Image Restoration.

3.5 Fixed-Point Schemes for Optic Flow without Warping

Now we address the fixed-point schemes for the Generic Optic Flow setting where we are using a linearised Data Term. We remember from the previous chapter that the third and fourth equation types have the same structure as the ones in the Image Restoration case. Thus we may reuse the fixed-point schemes we already derived for those types. The first and second equation types, however, have a different structure this time.

The first type arises for $k = n = 1$ and is given by:

$$0 = \widehat{J}_{11}\ A_1^1 + \widehat{J}_{12}\ A_2^1 + \widehat{J}_{13} - \alpha\ \mathrm{div}\left(D_\Psi^G(A^1)\nabla A_1^1\right)$$
$$0 = \widehat{J}_{12}\ A_1^1 + \widehat{J}_{22}\ A_2^1 + \widehat{J}_{23} - \alpha\ \mathrm{div}\left(D_\Psi^G(A^1)\nabla A_2^1\right)$$

The discrete versions are given by:

$$0 = \left[\widehat{J}_{11}\right]_{i,j}\left[A_1^1\right]_{i,j} + \left[\widehat{J}_{12}\right]_{i,j}\left[A_2^1\right]_{i,j} + \left[\widehat{J}_{13}\right]_{i,j} - \alpha \sum_{(\hat{x},\hat{y})\in\mathcal{N}} [\mathcal{D}]_{i,j}(\hat{x},\hat{y})\ \left[A_1^1\right]_{i+\hat{x},j+\hat{y}}$$
$$0 = \left[\widehat{J}_{12}\right]_{i,j}\left[A_1^1\right]_{i,j} + \left[\widehat{J}_{22}\right]_{i,j}\left[A_2^1\right]_{i,j} + \left[\widehat{J}_{23}\right]_{i,j} - \alpha \sum_{(\hat{x},\hat{y})\in\mathcal{N}} [\mathcal{D}]_{i,j}(\hat{x},\hat{y})\ \left[A_2^1\right]_{i+\hat{x},j+\hat{y}}$$

We may use the following fixed-point iteration step to obtain $A_1^{1,m+1}$ and $A_2^{1,m+1}$ simultaneously:

$$\begin{pmatrix}\left[A_1^{1,m+1}\right]_{i,j}\\ \left[A_2^{1,m+1}\right]_{i,j}\end{pmatrix} = (1-\omega)\begin{pmatrix}\left[A_1^{1,m}\right]_{i,j}\\ \left[A_2^{1,m}\right]_{i,j}\end{pmatrix} + \omega\ \begin{pmatrix}m_{11} & m_{12}\\ m_{12} & m_{22}\end{pmatrix}^{-1}\begin{pmatrix}b_1\\ b_2\end{pmatrix}$$

where

$$m_{11} = \left[\widehat{J}_{11}^m\right]_{i,j} - \alpha\,[\mathcal{D}^m]_{i,j}(0,0)$$
$$m_{12} = \left[\widehat{J}_{12}^m\right]_{i,j}$$
$$m_{22} = \left[\widehat{J}_{22}^m\right]_{i,j} - \alpha\,[\mathcal{D}^m]_{i,j}(0,0)$$

and

$$b_1 = -\left[\widehat{J}_{13}^m\right]_{i,j} +$$
$$\alpha\Bigg(\sum_{(\hat{x},\hat{y})\in\mathcal{N}^-} [\mathcal{D}^m]_{i,j}(\hat{x},\hat{y})\ \left[A_1^{1,m+1}\right]_{i+\hat{x},j+\hat{y}} + \sum_{(\hat{x},\hat{y})\in\mathcal{N}^+} [\mathcal{D}^m]_{i,j}(\hat{x},\hat{y})\ \left[A_1^{1,m}\right]_{i+\hat{x},j+\hat{y}}\Bigg)$$

$$b_2 = -\left[\widehat{J}_{23}^{m}\right]_{i,j} +$$
$$\alpha\Bigg(\sum_{(\hat{x},\hat{y})\in\mathcal{N}^-} [\mathcal{D}^m]_{i,j}(\hat{x},\hat{y}) \ \left[A_2^{1,m+1}\right]_{i+\hat{x},j+\hat{y}} + \sum_{(\hat{x},\hat{y})\in\mathcal{N}^+} [\mathcal{D}^m]_{i,j}(\hat{x},\hat{y}) \ \left[A_2^{1,m}\right]_{i+\hat{x},j+\hat{y}} \Bigg)$$

This is the extrapolating variant of the so-called *Point-Coupled Gauss-Seidel Relaxation Step* with *Frozen Coefficients* ([8]). We see that the possibly nonlinear expressions like the stencil $\mathcal{D}$ and the Motion Tensor are evaluated at the old iteration step m.

For the second type that arises for $k = n$ if $n > 1$ we have:

$$0 = \widehat{J}_{11}\ A_1^n + \widehat{J}_{12}\ A_2^n + \widehat{J}_{13} + \alpha\ \left(A_{11_x}^{n-1} - A_{1_{xx}}^n + A_{12_y}^{n-1} - A_{1_{yy}}^n\right)$$
$$0 = \widehat{J}_{12}\ A_1^n + \widehat{J}_{22}\ A_2^n + \widehat{J}_{23} + \alpha\ \left(A_{21_x}^{n-1} - A_{2_{xx}}^n + A_{22_y}^{n-1} - A_{2_{yy}}^n\right)$$

Their discretisation is given by:

$$\begin{aligned}
0 = \left[\widehat{J}_{11}\right]_{i,j}[A_1^n]_{i,j} + \left[\widehat{J}_{12}\right]_{i,j}[A_2^n]_{i,j} + \left[\widehat{J}_{13}\right]_{i,j} + \\
\alpha\ \Bigg(\chi_{[1,N-1]\times[1,M]}\left(\frac{\left[A_{11}^{n-1}\right]_{i,j}}{h} - \frac{[A_1^n]_{i+1,j} - [A_1^n]_{i,j}}{h^2}\right) - \\
\chi_{[2,N]\times[1,M]}\left(\frac{\left[A_{11}^{n-1}\right]_{i-1,j}}{h} - \frac{[A_1^n]_{i,j} - [A_1^n]_{i-1,j}}{h^2}\right) + \\
\chi_{[1,N]\times[1,M-1]}\left(\frac{\left[A_{12}^{n-1}\right]_{i,j}}{h} - \frac{[A_1^n]_{i,j+1} - [A_1^n]_{i,j}}{h^2}\right) - \\
\chi_{[1,N]\times[2,M]}\left(\frac{\left[A_{12}^{n-1}\right]_{i,j-1}}{h} - \frac{[A_1^n]_{i,j} - [A_1^n]_{i,j-1}}{h^2}\right)\Bigg)
\end{aligned}$$

$$\begin{aligned}
0 = \left[\widehat{J}_{12}\right]_{i,j}[A_1^n]_{i,j} + \left[\widehat{J}_{22}\right]_{i,j}[A_2^n]_{i,j} + \left[\widehat{J}_{23}\right]_{i,j} + \\
\alpha\ \Bigg(\chi_{[1,N-1]\times[1,M]}\left(\frac{\left[A_{21}^{n-1}\right]_{i,j}}{h} - \frac{[A_2^n]_{i+1,j} - [A_2^n]_{i,j}}{h^2}\right) - \\
\chi_{[2,N]\times[1,M]}\left(\frac{\left[A_{21}^{n-1}\right]_{i-1,j}}{h} - \frac{[A_2^n]_{i,j} - [A_2^n]_{i-1,j}}{h^2}\right) + \\
\chi_{[1,N]\times[1,M-1]}\left(\frac{\left[A_{22}^{n-1}\right]_{i,j}}{h} - \frac{[A_2^n]_{i,j+1} - [A_2^n]_{i,j}}{h^2}\right) - \\
\chi_{[1,N]\times[2,M]}\left(\frac{\left[A_{22}^{n-1}\right]_{i,j-1}}{h} - \frac{[A_2^n]_{i,j} - [A_2^n]_{i,j-1}}{h^2}\right)\Bigg)
\end{aligned}$$

Again we use a point-coupled iteration step:

$$\begin{pmatrix} \left[A_1^{1,m+1}\right]_{i,j} \\ \left[A_2^{1,m+1}\right]_{i,j} \end{pmatrix} = (1-\omega) \begin{pmatrix} \left[A_1^{1,m}\right]_{i,j} \\ \left[A_2^{1,m}\right]_{i,j} \end{pmatrix} + \omega \begin{pmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{pmatrix}^{-1} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

where this time

$$\begin{aligned} m_{11} &= \left[\widehat{J}_{11}^m\right]_{i,j} + \frac{\alpha}{h^2}\left(\chi_{[1,N-1]\times[1,M]} + \chi_{[2,N]\times[1,M]} + \chi_{[1,N]\times[1,M-1]} + \chi_{[1,N]\times[2,M]}\right) \\ m_{12} &= \left[\widehat{J}_{12}^m\right]_{i,j} \\ m_{22} &= \left[\widehat{J}_{22}^m\right]_{i,j} + \frac{\alpha}{h^2}\left(\chi_{[1,N-1]\times[1,M]} + \chi_{[2,N]\times[1,M]} + \chi_{[1,N]\times[1,M-1]} + \chi_{[1,N]\times[2,M]}\right) \end{aligned}$$

and

$$\begin{aligned} b_1 = -\left[\widehat{J}_{13}^m\right]_{i,j} - \\ \alpha\Bigg(\chi_{[1,N-1]\times[1,M]}\left(\frac{\left[A_{11}^{n-1,m}\right]_{i,j}}{h} - \frac{[A_1^{n,m}]_{i+1,j}}{h^2}\right) - \\ \chi_{[2,N]\times[1,M]}\left(\frac{\left[A_{11}^{n-1,m}\right]_{i-1,j}}{h} + \frac{\left[A_1^{n,m+1}\right]_{i-1,j}}{h^2}\right) + \\ \chi_{[1,N]\times[1,M-1]}\left(\frac{\left[A_{12}^{n-1,m}\right]_{i,j}}{h} - \frac{[A_1^{n,m}]_{i,j+1}}{h^2}\right) - \\ \chi_{[1,N]\times[2,M]}\left(\frac{\left[A_{12}^{n-1,m}\right]_{i,j-1}}{h} + \frac{\left[A_1^{n,m+1}\right]_{i,j-1}}{h^2}\right)\Bigg) \end{aligned}$$

$$b_2 = -\left[\widehat{J}_{23}^{m}\right]_{i,j} - \alpha\Bigg(\chi_{[1,N-1]\times[1,M]}\left(\frac{\left[A_{21}^{n-1,m}\right]_{i,j}}{h} - \frac{[A_2^{n,m}]_{i+1,j}}{h^2}\right) - \chi_{[2,N]\times[1,M]}\left(\frac{\left[A_{21}^{n-1,m}\right]_{i-1,j}}{h} + \frac{\left[A_2^{n,m+1}\right]_{i-1,j}}{h^2}\right) + \chi_{[1,N]\times[1,M-1]}\left(\frac{\left[A_{22}^{n-1,m}\right]_{i,j}}{h} - \frac{[A_2^{n,m}]_{i,j+1}}{h^2}\right) - \chi_{[1,N]\times[2,M]}\left(\frac{\left[A_{22}^{n-1,m}\right]_{i,j-1}}{h} + \frac{\left[A_2^{n,m+1}\right]_{i,j-1}}{h^2}\right)\Bigg)$$

This concludes the fixed-point schemes for the Optic Flow case with a linearised Data Term.

3.6 Fixed-Point Schemes for Optic Flow with Warping

Finally we consider the case of our true Generic Optic Flow approach, that is, the Data Term has not been linearised and we are using a Warping Strategy to find a minimiser.

Hence we are working on different discrete domains whose dimensions N^s and M^s depend on the current scale s. Furthermore we are using a splitting of our Displacement Field in an unknown part $dA^{n,s}$ and a known part $A^{n,s}$. We learnt that this splitting leads to a change in the Euler-Lagrange Equations, which requires us to update our fixed-point schemes. For the sake of brevity, however, we will now only state the final fixed-point steps.

For the first type we have:

$$\begin{pmatrix}\left[dA_1^{1,m+1,s}\right]_{i,j}\\ \left[dA_2^{1,m+1,s}\right]_{i,j}\end{pmatrix} = (1-\omega)\begin{pmatrix}\left[dA_1^{1,m,s}\right]_{i,j}\\ \left[dA_2^{1,m,s}\right]_{i,j}\end{pmatrix} + \omega \begin{pmatrix} m_{11} & m_{12}\\ m_{12} & m_{22}\end{pmatrix}^{-1}\begin{pmatrix} b_1\\ b_2\end{pmatrix}$$

where

$$\begin{aligned} m_{11} &= \left[\widehat{J}_{11}^{m,s}\right]_{i,j} - \alpha\,[\mathcal{D}^{m,s}]_{i,j}\,(0,0)\\ m_{12} &= \left[\widehat{J}_{12}^{m,s}\right]_{i,j}\\ m_{22} &= \left[\widehat{J}_{22}^{m,s}\right]_{i,j} - \alpha\,[\mathcal{D}^{m,s}]_{i,j}\,(0,0)\end{aligned}$$

and

$$b_1 = -\left[\widehat{J}_{13}^{m,s}\right]_{i,j} + \alpha\Bigg(\sum_{(\hat{x},\hat{y})\in\mathcal{N}^-}\left[\mathcal{D}^{m,s}\right]_{i,j}(\hat{x},\hat{y})\ \left[dA_1^{1,m+1,s}\right]_{i+\hat{x},j+\hat{y}} + \sum_{(\hat{x},\hat{y})\in\mathcal{N}^+}\left[\mathcal{D}^{m,s}\right]_{i,j}(\hat{x},\hat{y})\ \left[dA_1^{1,m,s}\right]_{i+\hat{x},j+\hat{y}} + \sum_{(\hat{x},\hat{y})\in\mathcal{N}}\left[\mathcal{D}^{m,s}\right]_{i,j}(\hat{x},\hat{y})\ \left[A_1^{1,s}\right]_{i+\hat{x},j+\hat{y}}\Bigg)$$

$$b_2 = -\left[\widehat{J}_{23}^{m,s}\right]_{i,j} + \alpha\Bigg(\sum_{(\hat{x},\hat{y})\in\mathcal{N}^-}\left[\mathcal{D}^{m,s}\right]_{i,j}(\hat{x},\hat{y})\ \left[dA_2^{1,m+1,s}\right]_{i+\hat{x},j+\hat{y}} + \sum_{(\hat{x},\hat{y})\in\mathcal{N}^+}\left[\mathcal{D}^{m,s}\right]_{i,j}(\hat{x},\hat{y})\ \left[dA_2^{1,m,s}\right]_{i+\hat{x},j+\hat{y}} + \sum_{(\hat{x},\hat{y})\in\mathcal{N}}\left[\mathcal{D}^{m,s}\right]_{i,j}(\hat{x},\hat{y})\ \left[A_2^{1,s}\right]_{i+\hat{x},j+\hat{y}}\Bigg)$$

As the fixed-point step for the second type we use:

$$\begin{pmatrix}\left[dA_1^{1,m+1,s}\right]_{i,j}\\ \left[dA_2^{1,m+1,s}\right]_{i,j}\end{pmatrix} = (1-\omega)\begin{pmatrix}\left[dA_1^{1,m,s}\right]_{i,j}\\ \left[dA_2^{1,m,s}\right]_{i,j}\end{pmatrix} + \omega\ \begin{pmatrix}m_{11} & m_{12}\\ m_{12} & m_{22}\end{pmatrix}^{-1}\begin{pmatrix}b_1\\ b_2\end{pmatrix}$$

where

$$m_{11} = \left[\widehat{J}_{11}^{m,s}\right]_{i,j} + \frac{\alpha}{h^2}\left(\chi_{[1,N^s-1]\times[1,M^s]} + \chi_{[2,N^s]\times[1,M^s]} + \chi_{[1,N^s]\times[1,M^s-1]} + \chi_{[1,N^s]\times[2,M^s]}\right)$$

$$m_{12} = \left[\widehat{J}_{12}^{m,s}\right]_{i,j}$$

$$m_{22} = \left[\widehat{J}_{22}^{m,s}\right]_{i,j} + \frac{\alpha}{h^2}\left(\chi_{[1,N^s-1]\times[1,M^s]} + \chi_{[2,N^s]\times[1,M^s]} + \chi_{[1,N^s]\times[1,M^s-1]} + \chi_{[1,N^s]\times[2,M^s]}\right)$$

and

$$b_1 = -\left[\widehat{J}_{13}^{m,s}\right]_{i,j} -$$
$$\alpha\Bigg(\chi_{[1,N^s-1]\times[1,M^s]}\Bigg(\frac{\left[A_{11}^{n-1,m,s}\right]_{i,j}}{h} - \frac{[A_1^{n,s}]_{i+1,j} + [dA_1^{n,m,s}]_{i+1,j}}{h^2}\Bigg) -$$
$$\chi_{[2,N^s]\times[1,M^s]}\Bigg(\frac{\left[A_{11}^{n-1,m,s}\right]_{i-1,j}}{h} + \frac{[A_1^{n,s}]_{i-1,j} + \left[dA_1^{n,m+1,s}\right]_{i-1,j}}{h^2}\Bigg) +$$
$$\chi_{[1,N^s]\times[1,M^s-1]}\Bigg(\frac{\left[A_{12}^{n-1,m,s}\right]_{i,j}}{h} - \frac{[A_1^{n,s}]_{i,j+1} + [dA_1^{n,m,s}]_{i,j+1}}{h^2}\Bigg) -$$
$$\chi_{[1,N^s]\times[2,M^s]}\Bigg(\frac{\left[A_{12}^{n-1,m,s}\right]_{i,j-1}}{h} + \frac{[A_1^{n,s}]_{i,j-1} + \left[dA_1^{n,m+1,s}\right]_{i,j-1}}{h^2}\Bigg)\Bigg)$$

$$b_2 = -\left[\widehat{J}_{23}^{m,s}\right]_{i,j} -$$
$$\alpha\Bigg(\chi_{[1,N^s-1]\times[1,M^s]}\Bigg(\frac{\left[A_{21}^{n-1,m,s}\right]_{i,j}}{h} - \frac{[A_2^{n,s}]_{i+1,j} + [dA_2^{n,m,s}]_{i+1,j}}{h^2}\Bigg) -$$
$$\chi_{[2,N^s]\times[1,M^s]}\Bigg(\frac{\left[A_{21}^{n-1,m,s}\right]_{i-1,j}}{h} + \frac{[A_2^{n,s}]_{i-1,j} + \left[dA_2^{n,m+1,s}\right]_{i-1,j}}{h^2}\Bigg) +$$
$$\chi_{[1,N^s]\times[1,M^s-1]}\Bigg(\frac{\left[A_{22}^{n-1,m,s}\right]_{i,j}}{h} - \frac{[A_2^{n,s}]_{i,j+1} + [dA_2^{n,m,s}]_{i,j+1}}{h^2}\Bigg) -$$
$$\chi_{[1,N^s]\times[2,M^s]}\Bigg(\frac{\left[A_{22}^{n-1,m,s}\right]_{i,j-1}}{h} + \frac{[A_2^{n,s}]_{i,j-1} + \left[dA_2^{n,m+1,s}\right]_{i,j-1}}{h^2}\Bigg)\Bigg)$$

We focus now on the third and fourth equation type. We recall from the previous chapter that they have the same structure like in the ususal Optic Flow case but only if the Displacement Field is not participating. Thus in these cases we can reuse our known fixed-point schemes. In the other cases these schemes change a little bit.

For the third type where $n = 2$ we have then:

$$\left[A_{p1}^{1,m+1,s}\right]_{i,j} = (1-\omega)\left[A_{p1}^{1,m,s}\right]_{i,j} + \omega\left(\chi_{[1,N^s-1]\times[1,M^s]} - \alpha\ \left[\mathcal{D}^{m,s}\right]_{i,j}(0,0)\right)^{-1}\cdot$$
$$\left(\chi_{[1,N^s-1]\times[1,M^s]}\left(\frac{\left[A_p^{2,s}\right]_{i+1,j} - \left[A_p^{2,s}\right]_{i,j}}{h} + \frac{\left[dA_p^{2,m+1,s}\right]_{i+1,j} - \left[dA_p^{2,m+1,s}\right]_{i,j}}{h}\right) + \right.$$
$$\alpha\left(\sum_{(\hat{x},\hat{y})\in\mathcal{N}^-}\left[\mathcal{D}^{m,s}\right]_{i,j}(\hat{x},\hat{y})\ \left[A_{p1}^{1,m+1,s}\right]_{i+\hat{x},j+\hat{y}} + \right.$$
$$\left.\left.\sum_{(\hat{x},\hat{y})\in\mathcal{N}^+}\left[\mathcal{D}^{m,s}\right]_{i,j}(\hat{x},\hat{y})\ \left[A_{p1}^{1,m,s}\right]_{i+\hat{x},j+\hat{y}}\right)\right)$$

$$\left[A_{p2}^{1,m+1,s}\right]_{i,j} = (1-\omega)\left[A_{p2}^{1,m,s}\right]_{i,j} + \omega\left(\chi_{[1,N^s]\times[1,M^s-1]} - \alpha\ \left[\mathcal{D}^{m,s}\right]_{i,j}(0,0)\right)^{-1}\cdot$$
$$\left(\chi_{[1,N^s]\times[1,M^s-1]}\left(\frac{\left[A_p^{2,s}\right]_{i,j+1} - \left[A_p^{2,s}\right]_{i,j}}{h} + \frac{\left[dA_p^{2,m+1,s}\right]_{i,j+1} - \left[dA_p^{2,m+1,s}\right]_{i,j}}{h}\right) + \right.$$
$$\alpha\left(\sum_{(\hat{x},\hat{y})\in\mathcal{N}^-}\left[\mathcal{D}^{m,s}\right]_{i,j}(\hat{x},\hat{y})\ \left[A_{p2}^{1,m+1,s}\right]_{i+\hat{x},j+\hat{y}} + \right.$$
$$\left.\left.\sum_{(\hat{x},\hat{y})\in\mathcal{N}^+}\left[\mathcal{D}^{m,s}\right]_{i,j}(\hat{x},\hat{y})\ \left[A_{p2}^{1,m,s}\right]_{i+\hat{x},j+\hat{y}}\right)\right)$$

And finally for the fourth type with $k = n - 1$ we have the fixed-point step:

$$\left[A_{p1}^{n-1,m+1,s}\right]_{i,j} = (1-\omega)\left[A_{p1}^{k,m,s}\right]_{i,j} +$$
$$\omega\left(\chi_{[1,N^s-1]\times[1,M^s]}+\frac{\alpha}{h^2}\left(\chi_{[1,N^s-1]\times[1,M^s]}+\chi_{[2,N^s]\times[1,M^s]}+\chi_{[1,N^s]\times[1,M^s-1]}+\chi_{[1,N^s]\times[2,M^s]}\right)\right)^{-1}\cdot$$
$$\left(\chi_{[1,N^s-1]\times[1,M^s]}\left(\frac{[A_p^{n,s}]_{i+1,j}-[A_p^{n,s}]_{i,j}}{h}+\frac{\left[dA_p^{n,m+1,s}\right]_{i+1,j}-\left[dA_p^{n,m+1,s}\right]_{i,j}}{h}\right)-\right.$$
$$\alpha\left(\chi_{[1,N^s-1]\times[1,M^s]}\left(\frac{\left[A_{p11}^{n-2,m,s}\right]_{i,j}}{h}-\frac{\left[A_{p1}^{n-1,m,s}\right]_{i+1,j}}{h^2}\right)-\right.$$
$$\chi_{[2,N^s]\times[1,M^s]}\left(\frac{\left[A_{p11}^{n-2,m,s}\right]_{i-1,j}}{h}+\frac{\left[A_{p1}^{n-1,m+1,s}\right]_{i-1,j}}{h^2}\right)+$$
$$\chi_{[1,N^s]\times[1,M^s-1]}\left(\frac{\left[A_{p12}^{n-2,m,s}\right]_{i,j}}{h}-\frac{\left[A_{p1}^{n-1,m,s}\right]_{i,j+1}}{h^2}\right)-$$
$$\left.\left.\chi_{[1,N^s]\times[2,M^s]}\left(\frac{\left[A_{p12}^{n-2,m,s}\right]_{i,j-1}}{h}+\frac{\left[A_{p1}^{n-1,m+1,s}\right]_{i,j-1}}{h^2}\right)\right)\right)$$

$$\left[A_{p2}^{n-1,m+1,s}\right]_{i,j} = (1-\omega)\left[A_{p2}^{k,m,s}\right]_{i,j} +$$
$$\omega\left(\chi_{[1,N^s]\times[1,M^s-1]}+\frac{\alpha}{h^2}\left(\chi_{[1,N^s-1]\times[1,M^s]}+\chi_{[2,N^s]\times[1,M^s]}+\chi_{[1,N^s]\times[1,M^s-1]}+\chi_{[1,N^s]\times[2,M^s]}\right)\right)^{-1}\cdot$$
$$\left(\chi_{[1,N^s]\times[1,M^s-1]}\left(\frac{[A_p^{n,s}]_{i,j+1}-[A_p^{n,s}]_{i,j}}{h}+\frac{\left[dA_p^{n,m+1,s}\right]_{i,j+1}-\left[dA_p^{n,m+1,s}\right]_{i,j}}{h}\right)-\right.$$
$$\alpha\left(\chi_{[1,N^s-1]\times[1,M^s]}\left(\frac{\left[A_{p21}^{n-2,m,s}\right]_{i,j}}{h}-\frac{\left[A_{p2}^{n-1,m,s}\right]_{i+1,j}}{h^2}\right)-\right.$$
$$\chi_{[2,N^s]\times[1,M^s]}\left(\frac{\left[A_{p21}^{n-2,m,s}\right]_{i-1,j}}{h}+\frac{\left[A_{p2}^{n-1,m+1,s}\right]_{i-1,j}}{h^2}\right)+$$
$$\chi_{[1,N^s]\times[1,M^s-1]}\left(\frac{\left[A_{p22}^{n-2,m,s}\right]_{i,j}}{h}-\frac{\left[A_{p2}^{n-1,m,s}\right]_{i,j+1}}{h^2}\right)-$$
$$\left.\left.\chi_{[1,N^s]\times[2,M^s]}\left(\frac{\left[A_{p22}^{n-2,m,s}\right]_{i,j-1}}{h}+\frac{\left[A_{p2}^{n-1,m+1,s}\right]_{i,j-1}}{h^2}\right)\right)\right)$$

After a lot of work we are now almost ready to proceed with the experiments.

3.7 Details on the Warping Strategy

Finally we discuss the details of the Warping Strategy in the discrete case. We recollect that we want to consider our data f on different scales s such that:

- f^1 is the original data
- f^{s+1} is the data on a smaller scale than f^s
- $f^{\max}$ is the data on the smallest scale
- and the difference between two scales is not very big

Our original data $f^1 := f$ is already given. It has the dimensions $N^1 \times M^1 := N \times M$. In order to obtain f^s we are reducing the sample amount taken in x-direction and y-direction by multiplying the original sample amounts with a power of a so-called *reduction factor* $\eta \in [0.5, 1[$:

$$N^s = \lfloor \eta^{s-1} N \rfloor$$
$$M^s = \lfloor \eta^{s-1} M \rfloor$$

where $\lfloor x \rfloor$ indicates that we are rounding x down to the first integer y with $y \leq x$.

We realise that if η is chosen very large the differences between two different scales might become very small.

Let us now turn to the data on the smallest scale $s = \max$. Here we will prescribe a smallest allowed sample amount μ such that the following hold

- Either we have
$$N^{\max} = \mu \text{ and } M^{\max} \geq \mu$$
- or
$$N^{\max} \geq \mu \text{ and } M^{\max} = \mu$$

Thus our number of scales is implicitly determined by η and μ.

As we are reducing our samples we have to resample our data on the current scale. For this *downsampling process* we will use a simple *linear interpolation* after we prepared the data with a suitable presmoothing[8]. Actually we will use the same procedure to transfer quantities from the lower scale s to the next higher one. As we are this time performing an *upsampling* we do not need to presmooth the respective quantity before the process.

[8]This presmoothing is this time necessary to avoid aliasing effects.

Furthermore we need the frame $f^s(x + A_1^{n,s}, y + A_2^{n,s}, t + 1)$ on the current scale s to compute our Motion Tensor. This implies that we have to evaluate our image data at sample locations that are not given, which means that we are again facing a resampling process. Here also the linear interpolation is used.

Finally we discuss the intialisations we will use on each scale s:

- The unknown part of the Displacement Field $dA^{n,s}$ is always initialised to an arbitrary guess.
- For the remaining parts $A^{k,s}$ with $k \in [1, n]$ we distinguish between two cases:
  - If we are on the smallest scale $s = \max$ we will set the known part of the Displacement Field $A^{n,\max}$ to 0. The remaining parts of the solution $A^{k,\max}$ are initialised to arbitrary values.
  - Otherwise we will transfer the sum $A^{n,s+1} + dA^{n,s+1}$ from the lower scale $s+1$ to the current one s in order to use its value as the current known part of the Displacement Field $A^{n,s}$. For all other solution parts $A^{k,s}$ we will use the interpolated values from their counterparts on the smaller scale $s + 1$, which serves the purpose of providing the fixed-point iteration with a suitable first guess.

Fortunately this concludes the discrete aspects of our generic methods.

4 Experiments

Until now we worked really hard to design new methods and to perform the discretisation of them. But we still need to verify how these methods perform in practice. To this end we will conduct some experiments in this chapter.

First we will consider some synthetically generated data we will test our Generic Image Restoration with. These experiments can be seen as a proof of concept as we will try to adapt our generic method to the current structure of the data.

Next we will proceed by analysing how our Image Restoration approach performs if we consider the image from chapter 1 where an ordinary approach did not produce satisfactory results.

After that we will have a look at our Generic Optic Flow method. Here we will first investigate if it performs better than a ususal Optic Flow method by considering the Uniaxial Tensile Experiment from chapter 1.

Finally we will push our Optic Flow method to the limit by considering a so-called *Biaxial Tensile Experiment*, which motivates the usage of a smoothness assumption of order 4.

4.1 Overview on the Parameters

Before we start with the experiments we give a short summary of the parameters that are used for the different methods.

For the Generic Image Restoration approach we have the parameters

Parameter	Description
n	The desired smoothness order the generic approach should use, also estimates of all derivatives of the solution up to order $n-1$ are computed
α	Weight of the Smoothness Term / Agreement Terms
Ψ	The Penaliser to use in the Smoothness Term
λ	The contrast parameter for the Charbonnier penaliser, only used if $\Psi = \Psi_C$
α_D	First parameter for Non-Standard Discretisation, always set to 0.45 and only used if $\Psi = \Psi_C$
β_D	Second parameter for Non-Standard Discretisation, always set to 0 and only used if $\Psi = \Psi_C$
$m_{\max}$	The iteration amount to use for solving the systems of equations
ω	The extrapolation parameter, always set to 1.97

In the case of Optic Flow we have to consider the following *additional* parameters:

Parameter	Description
σ	Standard deviation of the Gaussian Kernel that is used to presmooth the data, always set to 1
η	Reduction factor of the Warping Strategy, always set to 0.9
μ	Smallest allowed sample amount in the Warping Strategy, always set to 16
γ	Weight of the Gradient Constancy Assumption in the Data Term, always set to 1
ϵ	Parameter of the regularised L_1-norm that is used as penaliser in the Robust Data Term, always set to 0.1

4.2 Synthetic Experiments

In this series of experiments we evaluate some functions on a given discrete grid and examine if our Image Restoration approach can reconstruct this data without destroying too much information. Thus we are trying to find out if our approach can really be adapted in a flexible way to the structure of the current data.

Experiment 1

In the first experiment we evaluate the function $g_1(x, y) = x$ on a 229×229 grid where the origin is placed at the centre $(115, 115)$ of the grid. A plot of the function is given in figure 4.1 . We see that it produces data that shows a constant behavior along the y-axis and a linear behavior in x-direction. Therefore we can conclude that is has a first-order smoothness in y-direction and a second-order one in x-direction.

In the first part of the experiment we use the following set of parameters:

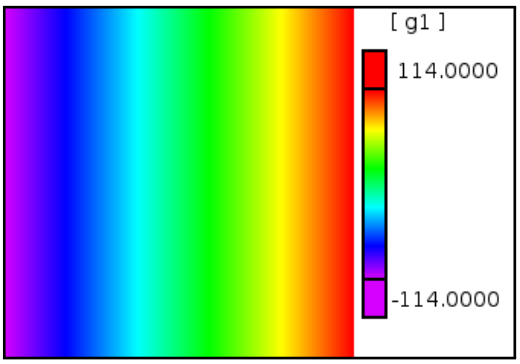
n	α	Ψ	$m_{\max}$
1	200	Ψ_{WT}	2000

Thus we are modelling a first-order homogeneous isotropic smoothness assumption. We see in figure 4.1 that information of the original structure is lost in the progress: We observe for example that the minimum and maximum are not preserved.

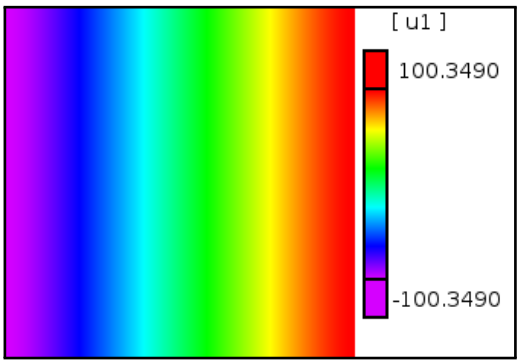
In the second part we utilise now the parameters:

n	α	Ψ	$m_{\max}$
2	200	Ψ_{WT}	2000

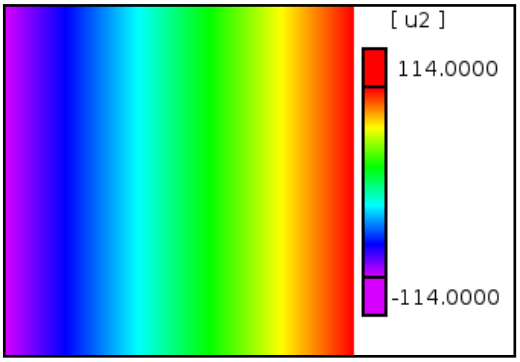
Hence, we are using now a homogeneous isotropic second-order assumption. Thus we are now adapting our smoothness assumption to the data by manipulating the single parameter n. The result in figure 4.1 is very motivating: The structure of our original function g_1 is well preserved. We observe also that our estimated derivatives of the solution actually correspond to the ones of the original function.



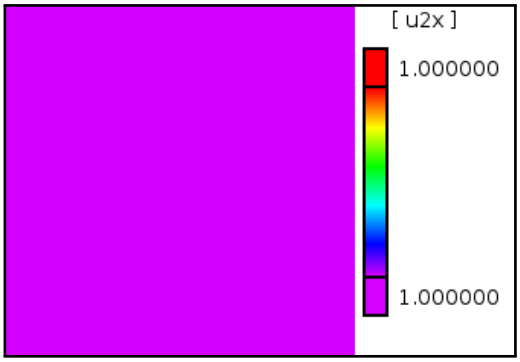
(a) Our function g_1.



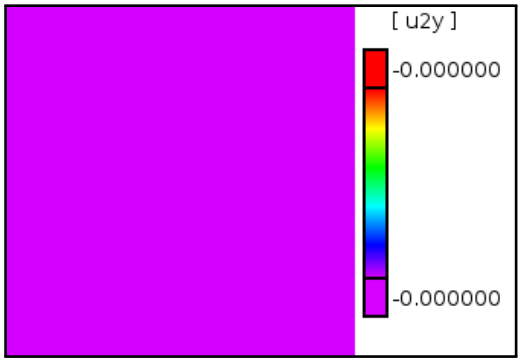
(b) Solution u for $n = 1$.



(c) Solution u for $n = 2$.



(d) Estimate of u_x for $n = 2$.



(e) Estimate of u_y for $n = 2$.

Figure 4.1: **Synthetic Experiment 1:** The function g_1 and results for different values of n.

Experiment 2

In the next experiment we will now consider a more complicated function: the absolute value of the x-value: $g_2(x, y) = |x|$. Again we evaluate it on a 229×229 grid where the origin is placed at the centre $(115, 115)$. The plot can be inspected in figure 4.2: We learn that the function has a piecewise second-order smoothness where the discontinuity is located in the centre.

For the first part we now use the parameters

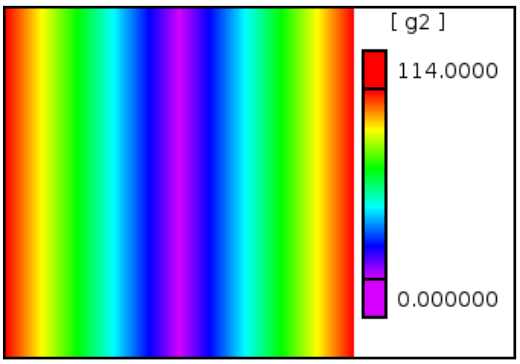
n	α	Ψ	$m_{\max}$
2	200	Ψ_{WT}	2000

That is, we are again using the second-order homogeneous isotropic Smoothness Term that performed very well in the previous experiment. This time, however, the result is less satisfying. We see in figure 4.2 that our method has some problems in the centre: It tries to remove the discontinuity. We can also see this effect in the estimated derivative in x-direction: The discontinuity is smoothed out.

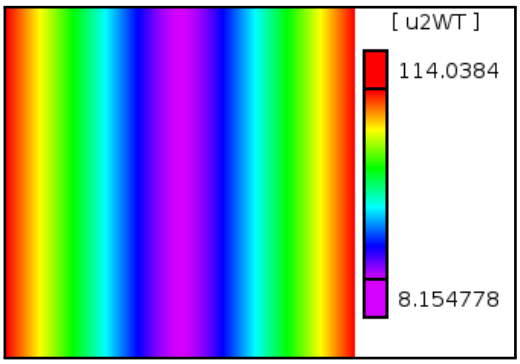
In the second part we change the parameters a little bit:

n	α	Ψ	$m_{\max}$	λ
2	200	Ψ_{C}	2000	0.01

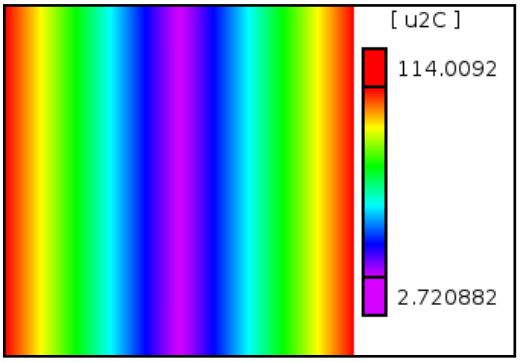
So we changed our assumption from a homogeneous isotropic one to a piecewise anisotropic one. We realise that this makes sense as our data g_2 has a piecewise second-order smoothness as stated above. By inspecting the results in figure 4.2 we notice that now the method tries to preserve the discontinuity in the centre. The result is, however, not perfect but still better than in the homogeneous case.



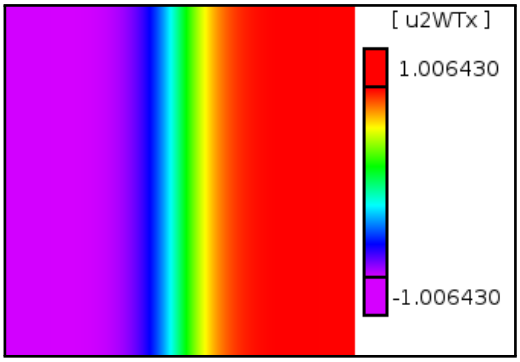
(a) Plot of our function g_2.



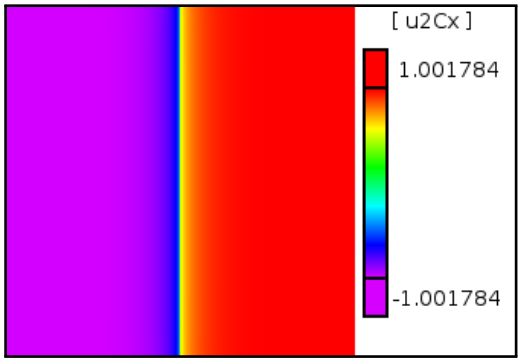
(b) Solution u for $\Psi = \Psi_{\mathrm{WT}}$.



(c) Solution u for $\Psi = \Psi_{\mathrm{C}}$.



(d) Estimate of u_x for $\Psi = \Psi_{\mathrm{WT}}$.



(e) Estimate of u_x for $\Psi = \Psi_{\mathrm{C}}$.

Figure 4.2: **Synthetic Experiment 2:** Original function g_2 and results for $n = 2$ and different choices for Ψ.

Experiment 3

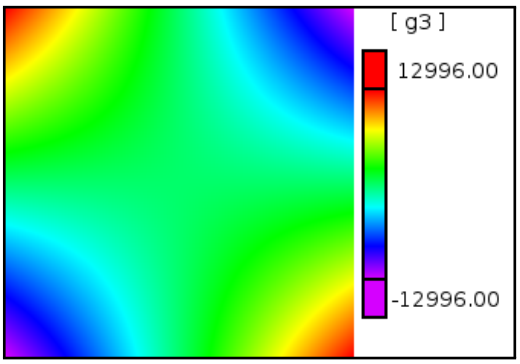
In the previous experiments we considered functions that showed only a non-constant behaviour in the x-direction. Now we finally turn to a function that shows such a behaviour in both directions: $g_3(x, y) = x\ y$. It is evaluated on the same grid like the other functions. The plot can be inspected in figure 4.3.

By studying this function a little bit we realise that it has a third-order smoothness as its third-order derivatives are the first ones that vanish all.

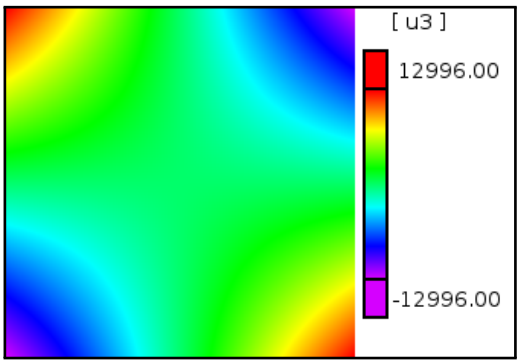
This observation motivates the usage of a homogeneous third-order smoothness assumption:

n	α	Ψ	$m_{\max}$
3	200	Ψ_{WT}	2000

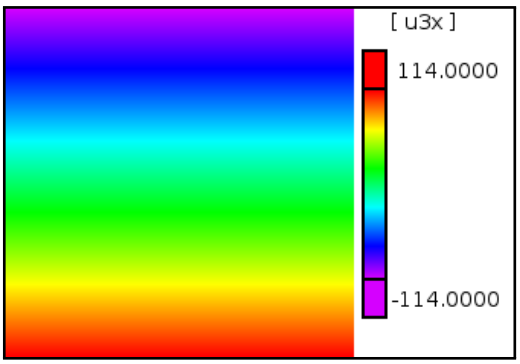
By inspecting the results in figure 4.3 we discover that this was the right choice: We see in our solution that no information of the structure of our original data was lost. Moreover we realise that we have this time also access to estimates of the second-order derivatives of our solution because we used $n = 3$.



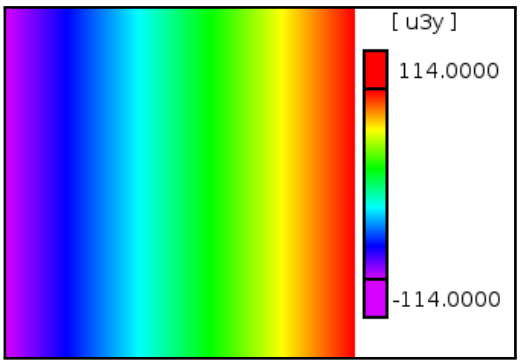
(a) Plot of our function g_3.



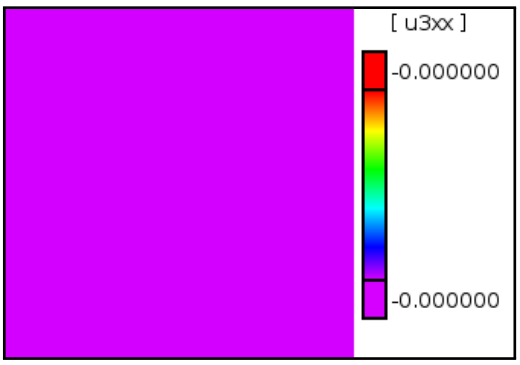
(b) Our solution u for $n = 3$.



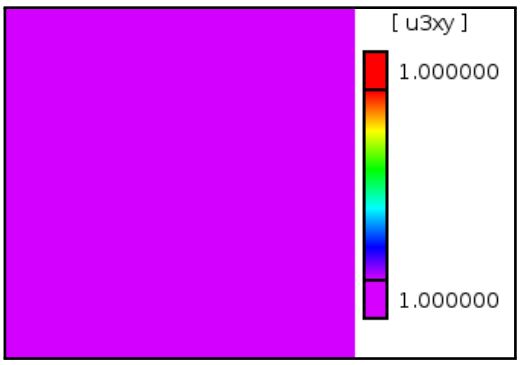
(c) Estimate of u_x.



(d) Estimate of u_y.



(e) Estimate of u_{xx}.



(f) Estimate of u_{xy}.

Figure 4.3: **Synthetic Experiment 3:** Original function g_3 and our solution u and estimates of its derivatives for $n = 3$.

4.3 Image Restoration Experiments

Finally we return to our noisy image we encoutered in chapter 1. We remember that we used in our first attempt a homogeneous first-order smoothness assumption. The used parameters are:

n	α	Ψ	$m_{\max}$
1	20	Ψ_{WT}	2000

We see in figure 4.4 again the dissatisfactory result: On the one hand the non-constant behaviour is preserved but on the other hand the discontinuity in the centre is lost.

We recollect that we also used a isotropic piecewise first-order assumption to denoise the image:

n	α	Ψ	$m_{\max}$	λ
1	20	Ψ_{C}	2000	1

Again the result did not satisfy us: This time the discontinuity was preserved but the non-constant behaviour was turned into a piecewise constant one.

Now we use the parameters

n	α	Ψ	$m_{\max}$
2	20	Ψ_{WT}	2500

Hence we are using a second-order homogeneous smoothness assumption. Unfortunately we see in figure 4.4 that we again are losing information about the discontinuity.

By replacing now our homogeneous assumption by a piecewise anisotropic one we can solve the problem:

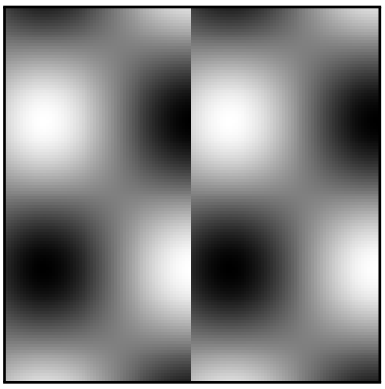
n	α	Ψ	$m_{\max}$	λ
2	20	Ψ_{C}	2500	0.02

This time we see in figure 4.4 that both the discontinuity in the centre and the non-constant behaviour in the two regions is preserved, which represents our best result.

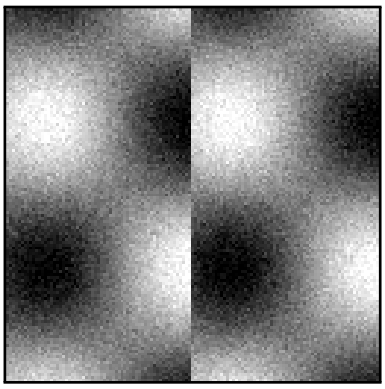
We remember that we discussed in chapter 1 the usefulness of first-order derivatives of a given image. In particular we learned that the magnitude of the image gradient $\|\nabla f\|$ can be used to detect edges in the image. For visualising the detected edges we might create an image g such that

$$g(x,y) = \begin{cases} 255 & \text{if } \|\nabla f(x,y)\| > t \\ 0 & \text{else} \end{cases}$$

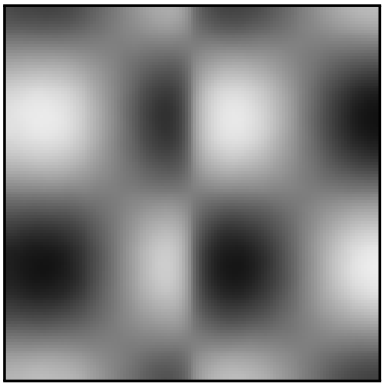
where t is a given *threshold*. Thus we are coloring each candidate for an edge white and everything else black. We will now use this ansatz to detect the edges in our best solution. Here we are using $t = 45$ and the estimated gradient of our solution. We see in figure 4.5 a very satisfying result: The preserved discontinuity in the centre is detected as an edge. Thus we conclude that our estimated derivatives of the solution are in fact quite useful.



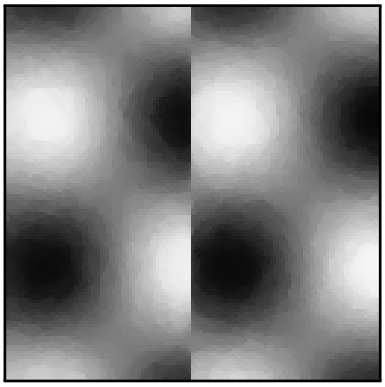
(a) Original image we already encountered in chapter 1.



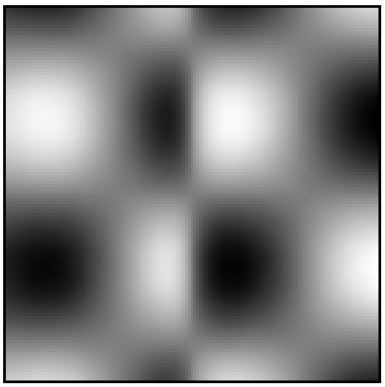
(b) Noisy version of the image.



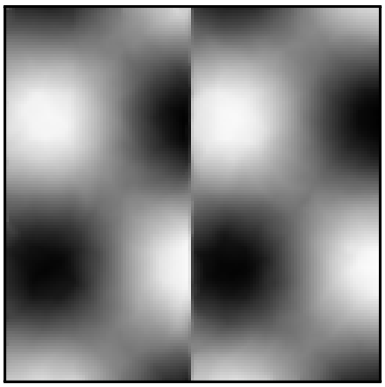
(c) The result of our denoising attempt from chapter 1, where we used $n = 1$ and $\Psi = \Psi_{\mathrm{WT}}$.



(d) Our second attempt from chapter 1, here the Charbonnier penaliser was used instead.

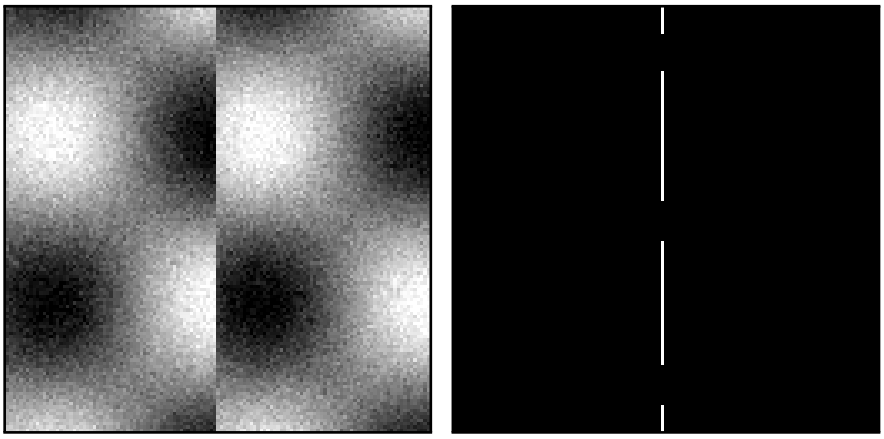


(e) Result from our generic approach with $n = 2$ and $\Psi = \Psi_{\mathrm{WT}}$.



(f) Our best result that we obtained for $n = 2$ and $\Psi = \Psi_{\mathrm{C}}$.

Figure 4.4: **Denoising Experiment:** Results of our generic approach for different choices for n and Ψ.



(a) Our noisy image.

(b) Detected edges by using $t = 45$ and the magnitude of the estimated gradient of our best solution.

Figure 4.5: **Edge Detection:** We see that we can use the estimates of the derivatives of our solution to perform an edge detection.

4.4 Optic Flow Experiments

Experiment 1

Now we return to our Uniaxial Tensile Experiment we considered in chapter 1 that is shown in figure 4.6. There we tried to estimate the motion by using a first-order piecewise anisotropic smoothness assumption:

n	α	Ψ	$m_{\max}$	λ
1	2000	Ψ_{C}	10000	0.01

We observe in figure 4.7 that this choice leads to a piecewise constant motion.

Now we use the parameters

n	α	Ψ	$m_{\max}$
2	2000	Ψ_{WT}	40000

So we are using now a homogeneous second-order assumption. By inspecting the result in figure 4.7 we learn that this time the estimated motion is linearly changing in space and also that its magnitude is more similar to the reference solution. Thus we conclude that our Generic Optic Flow approach seems to work, too.

Again we think about the potential usage of the derivatives of the motion we discussed in chapter 1. We remember that we can compute the Strain Tensor E by means of the Displacement Gradient $\nabla \mathbf{u}$. We will now compute this tensor by means of our estimated derivatives of our solution. In figure 4.8 we discover that the obtained strain is very homogeneous and looks similar to the one provided by Vic-2D. Moreover we see that the strain in y-direction is positive indicating that a stretching is occurring, which we can

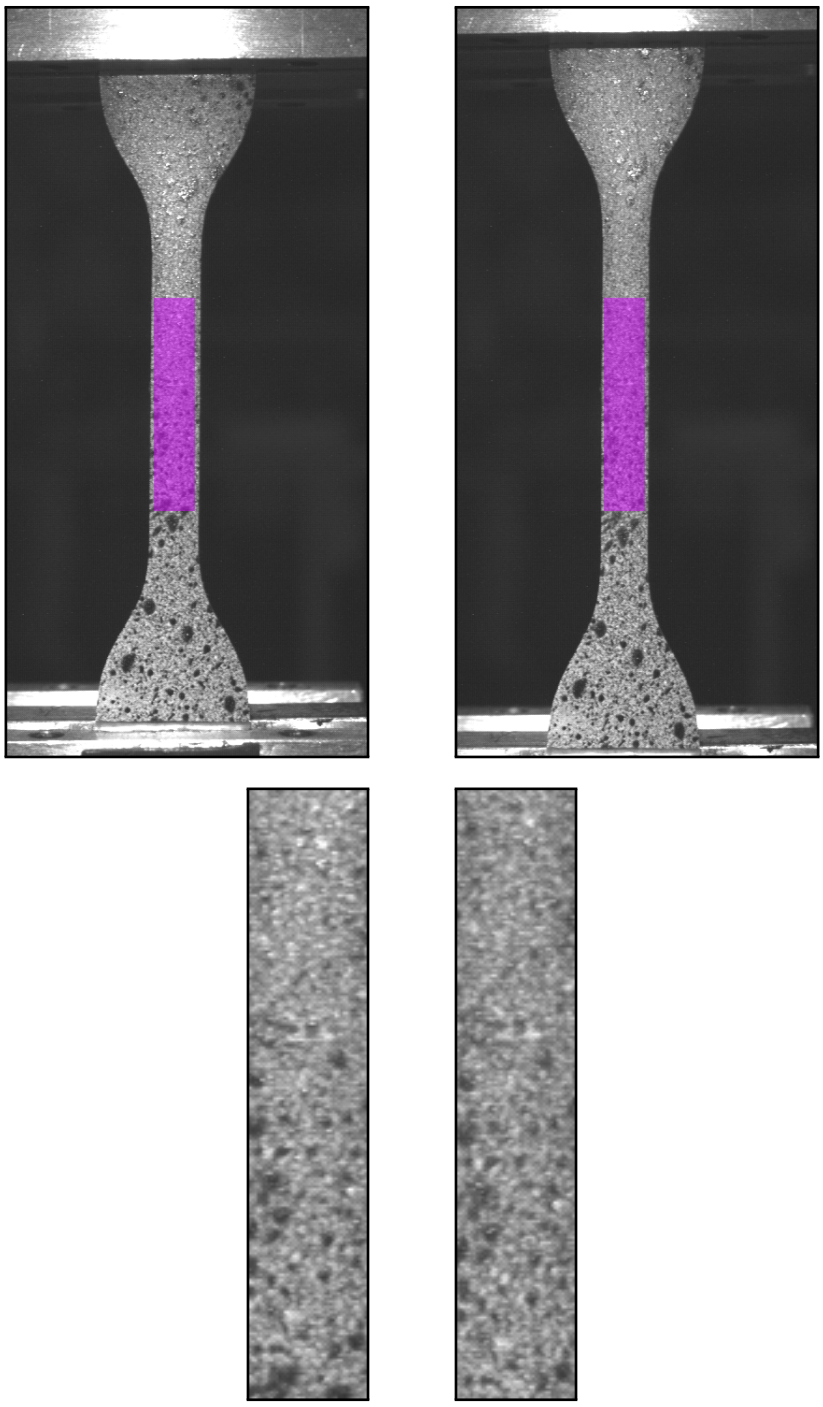


Figure 4.6: **Optic Flow Experiment 1**: **Top Row**: Two frames of an Uniaxial Tensile Experiment where the area of interest is colored in purple. **Bottom Row**: The respective enlarged areas of interest.

actually observe in the original images. However, we note that in the case of the shear component we see a large difference between our obtained minimum and maximum and the ones provided by Vic-2D.

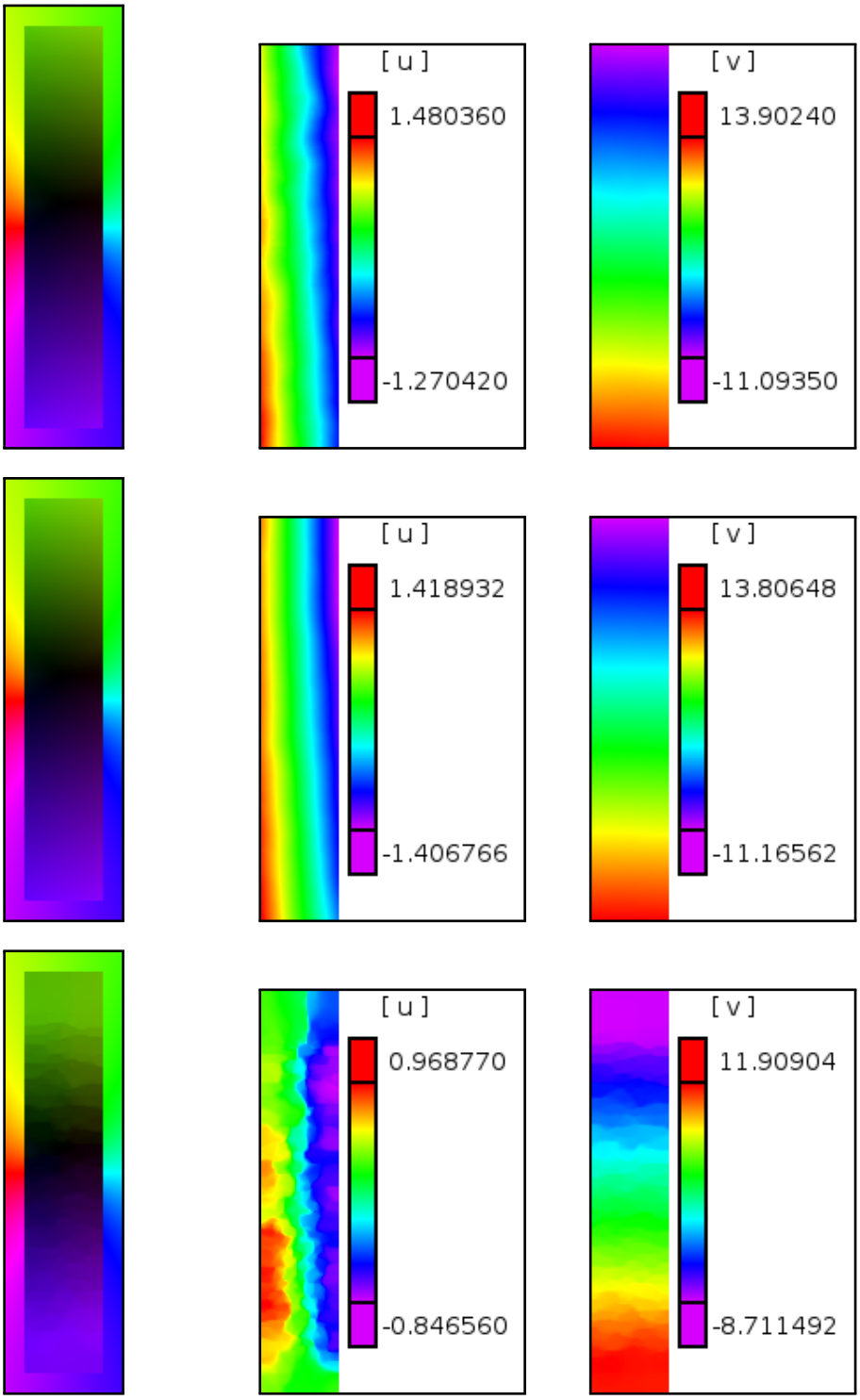


Figure 4.7: **Optic Flow Experiment 1**: **Top Row**: Reference Motion, **Centre Row**: Motion for $n = 2$ and $\Psi = \Psi_{\mathrm{WT}}$, **Bottom Row**: Motion for $n = 1$ and Ψ_{C}.

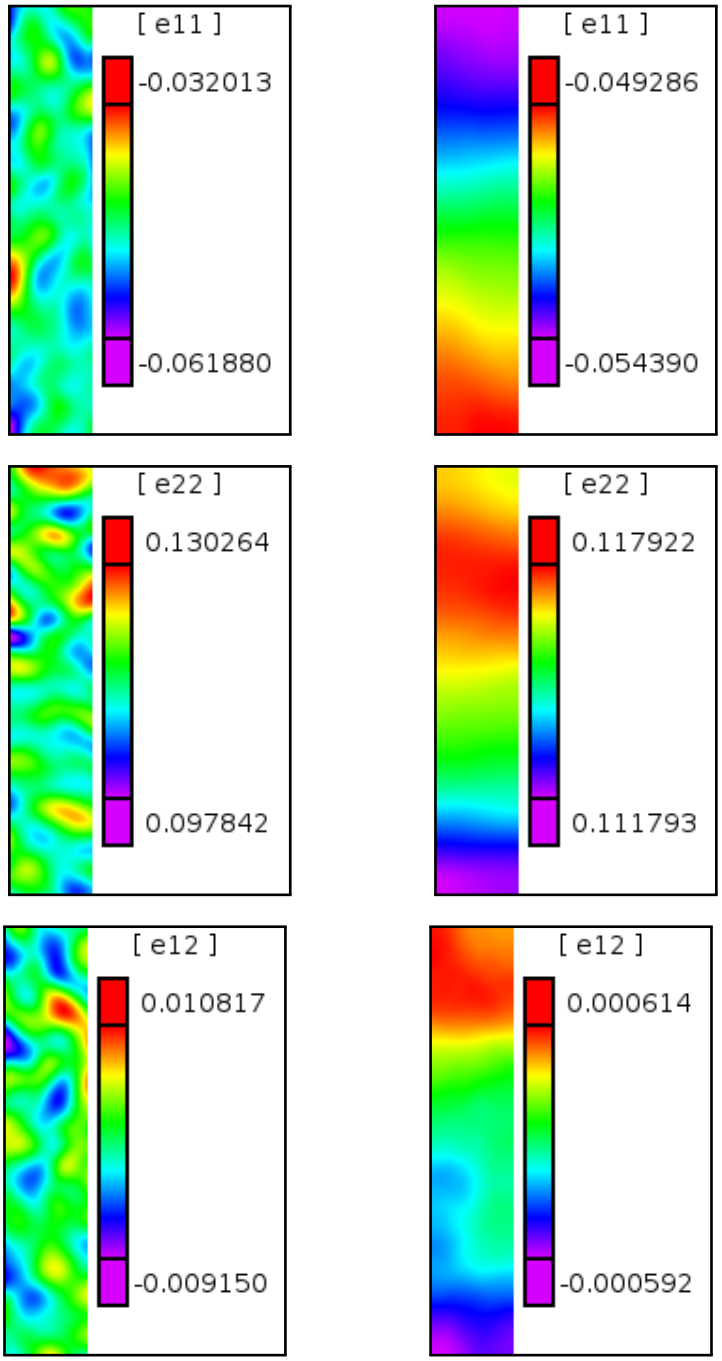


Figure 4.8: **Optic Flow Experiment 1**: **Left Column**: Reference strain, **Right Column**: Strain obtained for $n = 2$.

Experiment 2

In our final Optic Flow experiment we consider another experiment that is depicted in 4.9: A so-called Biaxial Tensile Experiment where we are using data with the new sample shape proposed by [11]. We see that in such an experiment the sample is stretched in two directions. Here we will again use the solution provided by Vic-2D as our reference solution.

In a first step we will apply our homoegeneous second-order assumption to estimate the motion and its first-order derivatives:

n	α	Ψ	$m_{\max}$
2	2000	Ψ_{WT}	40000

The result in figure 4.10 looks very promising: Both the directional information and the magnitude of the motion are very similar to our reference solution.

Let us now have a look at the estimate of v_y: In figure 4.12 we discover that it is not constant as it should be[1]. It actually resembles a quadratic function. But this implies that v itself should be a cubic function.

This observation motivates the usage of a fourth-order smoothness assumption:

n	α	Ψ	$m_{\max}$
4	2000	Ψ_{WT}	70000

We see now in figure 4.11 that the estimated motion is still similar to our reference solution. But in figure 4.12 we observe that v_y now shows a real quadratic structure, which actually indicates that our previous experiment with $n = 2$ had destroyed this information.

Finally we compute the Strain Tensor by using the estimate of the Displacement Gradient we obtained for $n = 4$ and compare the result to Vic-2D[2]. We see in figure 4.13 that the results are again very similar with one exception: the shear strain component.

In addition to that we discover that this time the strain is no longer homogeneous, that is, the strain is no longer constant in space.

4.5 Summary

We have seen in this chapter that our new method delivers acceptable results if we adapted the smoothness assumption via the parameter n. Furthermore we discovered that the estimates of the derivatives of the solution that we obtained can be used to gain access to additional information about the solution.

[1] Remember that we used a second-order assumption.

[2] Note that the white dots in the plot signal that Vic-2D possibly had problems to estimate the strain there.

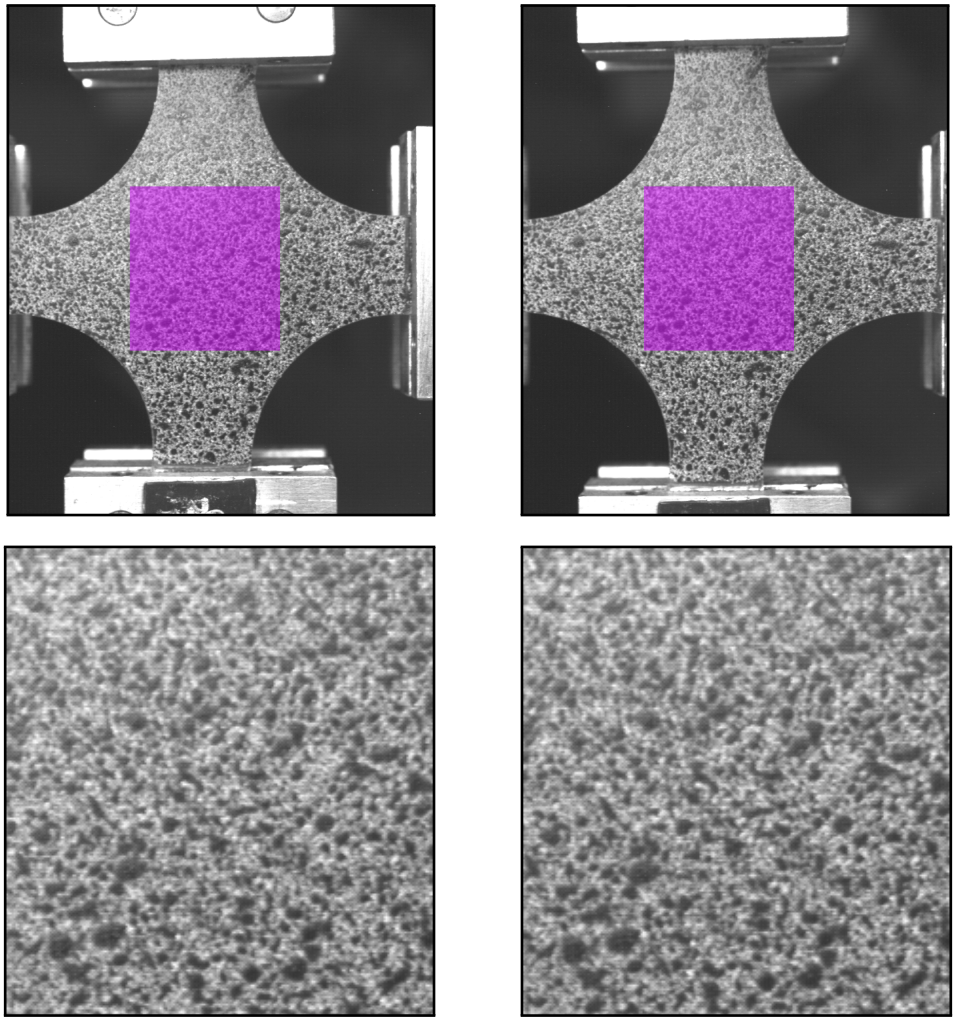


Figure 4.9: **Optic Flow Experiment 2**: **Top Row**: Two frames of an Biaxial Tensile Experiment where the area of interest is colored in purple. **Bottom Row**: The respective enlarged areas of interest.

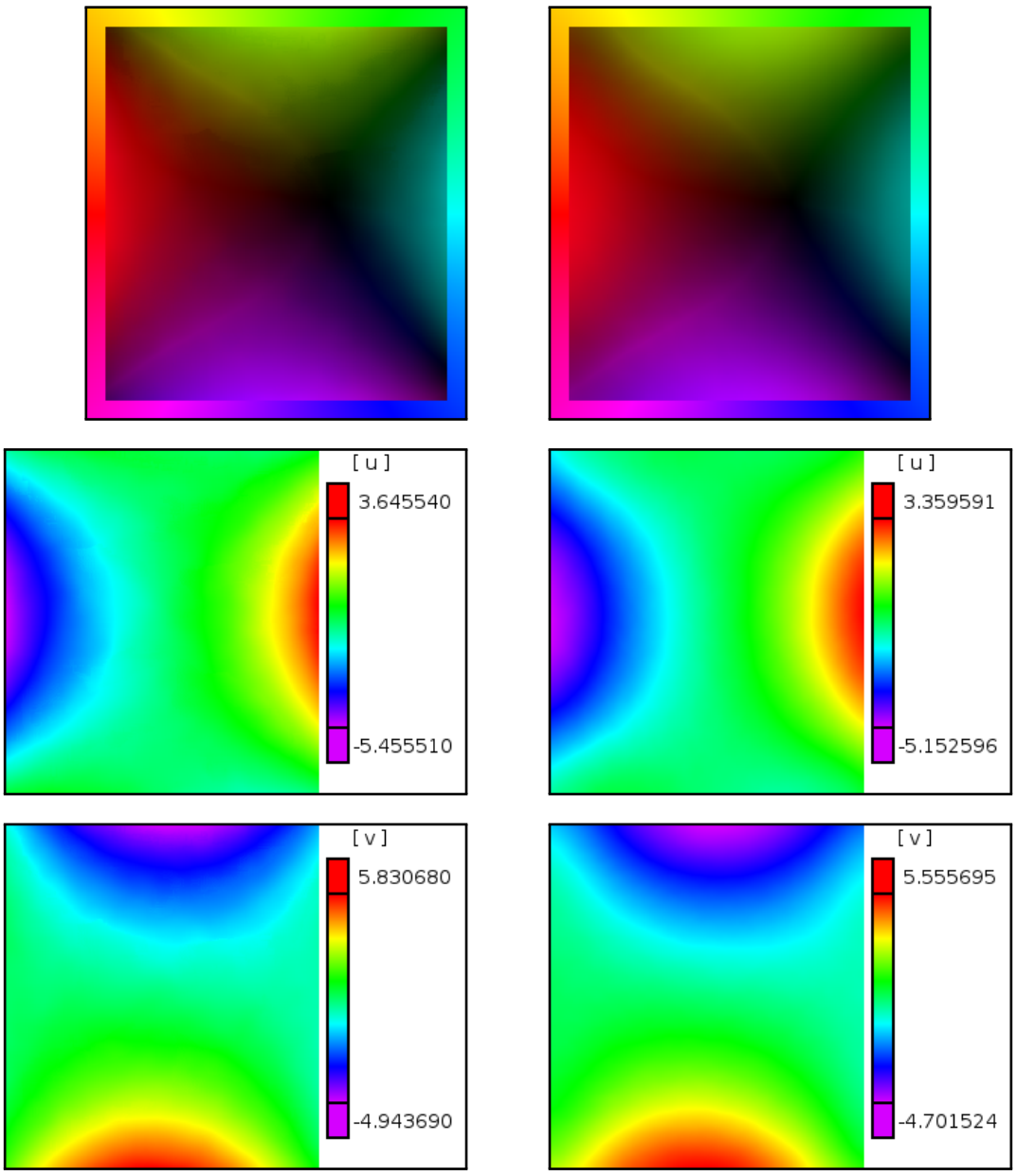


Figure 4.10: **Optic Flow Experiment 2**: **Left Column**: Reference motion, **Right Column**: Motion of our result for $n = 2$.

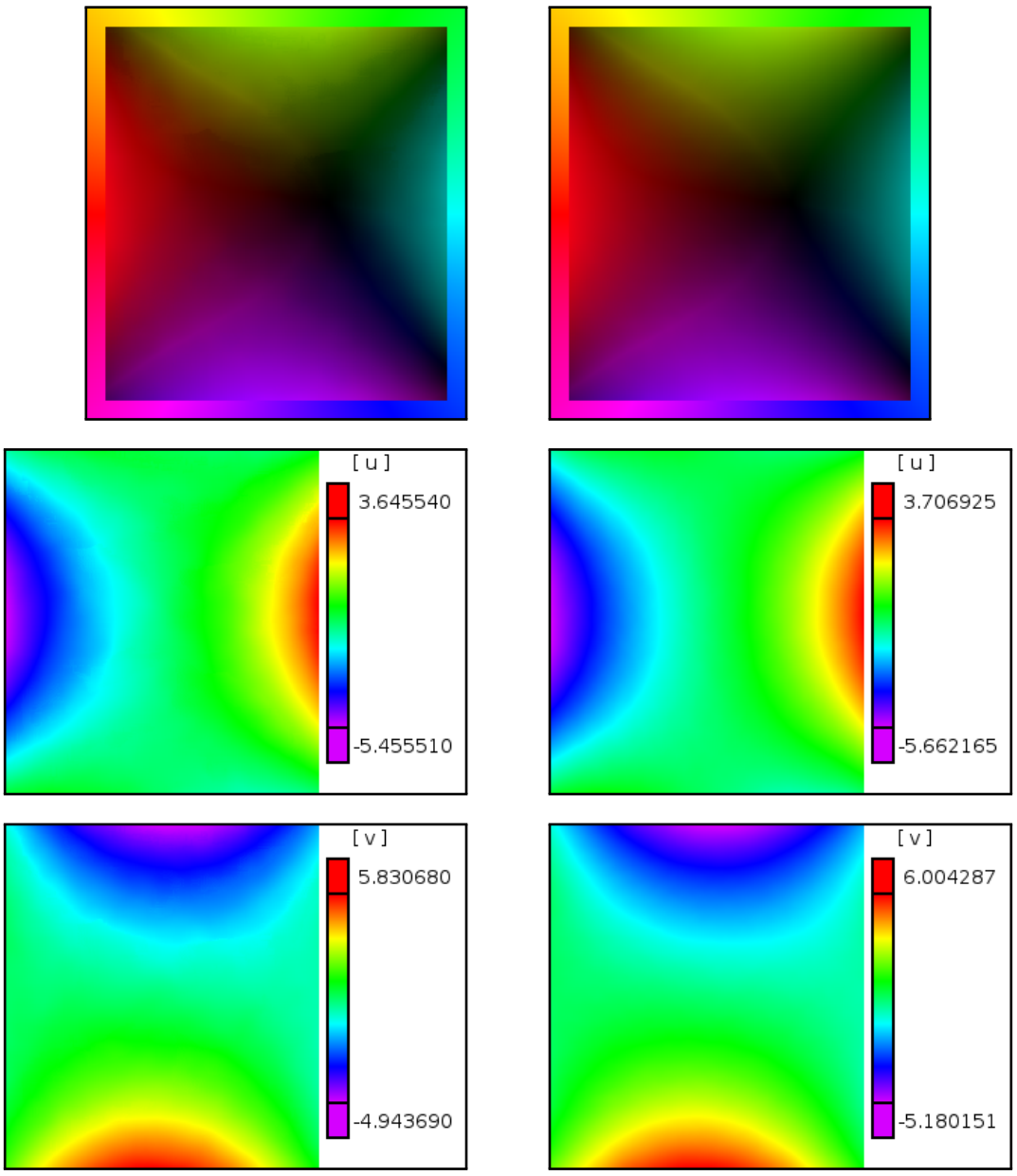


Figure 4.11: **Optic Flow Experiment 2**: **Left Column**: Reference Motion, **Right Column**: Motion of our result for $n = 4$.

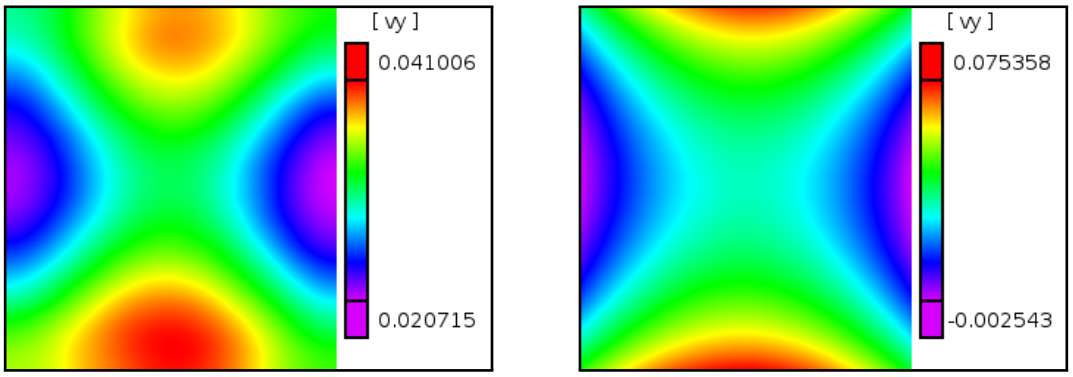


Figure 4.12: **Optic Flow Experiment 2**: **Left**: Estimate of v_y for $n = 2$, **Right**: Estimate of v_y for $n = 4$.

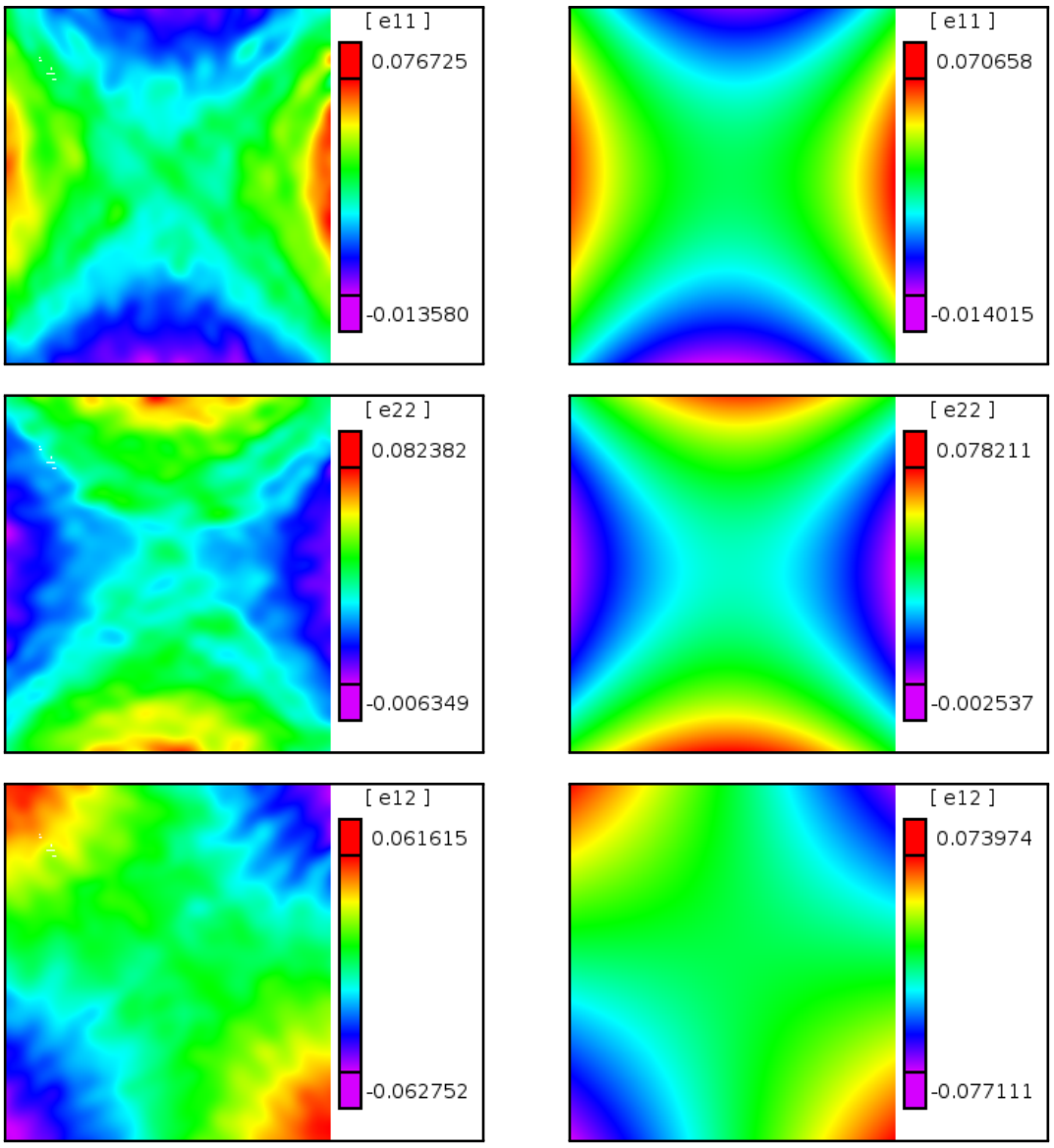


Figure 4.13: **Optic Flow Experiment 2**: **Left Column**: Reference strain, **Right Column**: Strain obtained for $n = 4$.

5 Conclusion

5.1 Summary

In this work we were concerned with Variational Methods where we focused in particular on the modelling process.

In chapter 1 we learned that a Smoothness Term that worked for one set of data might fail in cases where the data did have some special properties. We then argued that fixing the smoothness assumption to a specific idea might be not an optimal choice. Furthermore we discovered that we could extract higher order information form a solution by inspecting its derivatives.

These two findings motivated us in chapter 2 to design a Generic Smoothness Term that can be adapted to different data by manipulating a single parameter. In addition to that we designed it in such a way that higher order derivatives of the solution are also obtained in the process. After that we researched the so-called Euler-Lagrange Equations associated to our new approaches, that is, the necessary conditions our solutions must fulfill.

As we wanted to perform experiments on discrete data we were forced to think about the discretisation process in chapter 3. Here we discussed how the different types of Euler-Lagrange Equations could be discretised and which iterative solver we could use for each type.

In chapter 4 we were then finally able to investigate how our new approaches are performing. We saw in the Image Restoration case that we now could denoise an image where we failed in chapter 1. Furthermore we were also able to extract edge information from this noisy picture by applying our new method. In the Optic Flow experiments we also observed that our new approach seems to work as expected: We could adjust the order of the smoothness assumption by manipulating a single parameter in order to adapt our approach to the current structure of the data. We also succeeded in computing the Strain Tensor by using our estimated derivatives of the solution.

5.2 Outlook

However, we observe that there are some topics left that might be worth investigating.

In this whole work we focused almost entirely on the modelling process and the evaluation of the designed methods. Thus we did not investigate any theoretical properties of our new approaches. For example we do not know if our new Energy Functionals have a unique minimiser[1].

[1]Here we are excluding the Optic Flow case where the Data Term was not linearised

Moreover we only combined our new framework with two Variational Methods: Image Restoration and Optic Flow Estimation. It would be very interesting to know if other methods could also benefit from this new framework.

In terms of numerics we only used a relatively simple method. Again we find it worth investigating if the computing time can be improved by employing another solving technique.

Bibliography

[1] New marble sequence download. `http://i21www.ira.uka.de/image_sequences/`.

[2] *Vic-2D Reference Manual.*

[3] Yosemite sequence without clouds author. `http://cs.brown.edu/~black/`.

[4] Yosemite sequence without clouds download. `http://www.informatik.uni-ulm.de/ni/staff/PBayerl/homepage/animations/index.html`.

[5] Luis Álvarez, Carlos A. Castaño-Moraga, Miguel García, Karl Krissian, Luis Mazorra, Agustín Salgado, and Javier Sánchez. Second order variational optic flow estimation. In *EUROCAST*, pages 646–653, 2007.

[6] Kristian Bredies, Karl Kunisch, and Thomas Pock. Total generalized variation. *SIAM J. Imaging Sciences*, 3(3):492–526, 2010.

[7] A. Bruhn and J.Weickert. Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In *ICCV*, volume 1, pages 749–755. IEEE Computer Society Press, 2005.

[8] Andrés Bruhn. *Variational Optic Flow Computation: Accurate Modelling and Efficient Numerics.* PhD thesis, Saarland University, Saarbrücken, 2006.

[9] Andrés Bruhn. Correspondence problems in computer vision. *Lecture Notes*, 2008.

[10] I.M. Gelfand and S.V. Fomin. *Calculus of Variations [by] I.M. Gelfand [and] S.V. Fomin.* Selected Russian publications in the mathematical sciences. Prentice-Hall, 1964.

[11] S. Diebels H. Seibert, T. Scheffer. Biaxial testing of elastomers - experimental setup, measurement and experimental optimisation of specimen's shape. *Experimental Mechanics*, 2013.

[12] J. Weickert, L. Valgaerts, A. Salgado, B. Rosenhahn, H.-P. Seidel H. Zimmer, A. Bruhn. Complementary optic flow. In *EMMCVPR*, volume 5681, pages 207–220. Springer, 2009.

[13] Alexander Hewer. Joint second order optic flow regularisation and estimation of the associated displacement gradient. Saarland University, Saarbrücken, 2012.

[14] R.W. Ogden. *Non-Linear Elastic Deformations.* Dover, 1997.

[15] Nils Papenberg, Andrés Bruhn, Thomas Brox, Stephan Didas, and Joachim Weickert. Highly accurate optic flow computation with theoretically justified warping. *Int. J. Comput. Vision*, 67(2):141–158, April 2006.

[16] T . Corpetti, E. Memin, P. Perez. Dense estimation of fluid flows. 2002.

[17] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003.

[18] J. Weickert, C. Schnörr. A theoretical framework for convex regularizers in pde-based computation of image motion. *International Journal of Computer Vision*, 45(3):245–264, December 2001.

[19] Michael A. Sutton, Jean-José Orteu, Hubert Schreier. *Image Correlation for Shape, Motion and Deformation Measurements: Basic Concepts,Theory and Applications*. Springer, 2009.

[20] B. Horn, B. Schunck. Determing optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[21] W. Trobin, T. Pock, D. Cremers, and H. Bischof. An unbiased second-order prior for high-accuracy motion estimation. In *dagm*, lncs. Springer, 2008.

[22] Joachim Weickert. Partial differential equations in image processing and computer vision. University of Mannheim, 2001.

[23] Joachim Weickert. Differential equations in image processing and computer vision. *Lecture Notes*, 2008.

[24] Joachim Weickert. Image processing and computer vision. *Lecture Notes*, 2008.

[25] M. Welk, G. Steidl, J. Weickert. Locally analytic schemes: a link between diffusion filtering and wavelet shrinkage. 2008.

[26] Joachim Weickert, Martin Welk, Marco Wickert. Anisotropic diffusion schemes based on discrete energies with nonstandard finite differences. 2011.