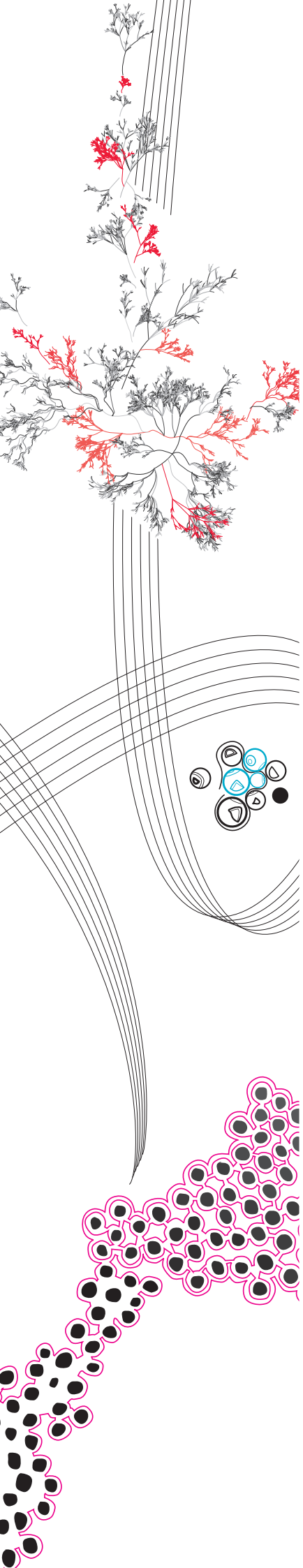


Joint reconstruction and motion estimation via nonconvex optimization for dynamic MRI

MASTER THESIS W.A.P. BASTIAANSEN

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)
Master program Applied Mathematics
Specialization: Systems Theory, Applied Analysis and Computational Science (SACS)
Chair: Applied Analysis

Graduation committee:
Prof. dr. S.A. van Gils (UT)
Dr. C. Brune (UT)
Dr. ir. C.A.T. van den Berg (UMC Utrecht)
Prof dr. ir. B.J. Geurts (UT)

Daily supervisor
Dr. C. Brune
Date of presentation
16-11-2018

UNIVERSITY OF TWENTE.

Abstract

Dynamic magnetic resonance imaging (MRI) is a medical imaging technique. MRI reconstruction methods build upon inverse problems, variational methods and optimization in applied mathematcis. To reduce scanning time in dynamic MRI, subsampling of the measurements is needed in practice. This typically leads to artefacts due to missing information. To tackle those artefacts, time-dependent reconstruction methods, which employ not only the spatial properties of the image sequence, but also the dynamical information, are very promising.

In this thesis a joint reconstruction and motion estimation framework is developed and applied to dynamic MRI reconstruction. The optimization of this joint variational model is challenging since it is nonconvex. Current approaches alternate between the convex subproblems for reconstruction and motion estimation. This approach seems to be working but there is no knowledge about the convergence. To address this problem the full nonconvex model is optimized via an alternating forward-backward splitting algorithm which is related to the PALM algorithm for nonconvex optimization.

The performance of joint reconstruction and motion estimation on dynamic MRI is studied. In real medical datasets no ground truth of the flow fields is available. To address this, artificial data from computer vision, with a ground truth for the flow fields at hand, is used to construct a numerical phantom. In combination with a 4D XCAT phantom, this offers more insight into the influence of incorporating the motion on the reconstruction quality and choice of optimization method. Finally, the model is applied to experimental medical data from the Radiotherapy group of the UMC Utrecht to show its potential for real world scenarios.

Acknowledgments

With this master thesis I conclude my time as a student applied mathematics. Six and a half years ago I exchanged the sunny south of the Netherlands (Brabant!) for the far east. I look back at my time as a student with a big smile on my face. I gained a lot of experiences I never thought I would. Running the final stage of the Batavierenrace, being a board member, being part of a 'dispuut' and so on. For every EC I had to work hard, sometimes because the course was difficult and sometimes because I was giving myself a hard time. I am proud of what I achieved over the last years and this thesis feels like the cherry on the pie. I would not have achieved this without the help of many people.

First of all I would like to thank my supervisor, Christoph Brune, who introduced me to the field of imaging and encouraged me to work on this project. While I was working on my thesis, we had a lot of very good, sometimes really long, discussions. Afterwards I always was full of new ideas and inspiration (and to be honest a bit exhausted as well). Without Christophs enormous knowledge and enthusiasm this thesis would not have been what it is. I asked for a challenge, and that is what I got, but it is save to say he helped me to get the most out of it.

I also want to thank Stephan van Gils for welcoming me into the Applied Analysis group and always supporting my changing plans regarding my master program. Furthermore, I want to thank Yoeri for proofreading my thesis and giving very valuable feedback. I also want to thank Sophie, for being my study-buddy during our master courses, where we became, while discussing mathematics, really good friends.

In this acknowledgements I also want to mention my study advisor Lilian Spijker and (former) bachelor coordinator Brigit Geveling, I struggled a lot in the beginning of my study and without their guidance and support I probably would not have made it past my first year.

Finally I want to thank my 'dispuut' Euphemia for always listening to my stories about my thesis at our drinks, but also for distracting me from it. My fellow board members, Daan and the 'meidenavond' girls for all the support and fun over the years. In particular I want to thank my rocks, Cordelia, Evelien, Hiska, Kim and Rianne for believing in me, when I forgot to do that. Last but not least, I want to thank my parents and sister for supporting me endlessly, even though they have no clue about what I do all day.

Contents

1 Introduction **6**

2 Model for joint reconstruction and motion estimation **9**
2.1 Data fidelity for MRI Reconstruction . . . 9
2.2 Regularizer for MRI reconstruction . . . 11
2.3 Model for motion estimation . . . 13
2.4 Regularizer for motion estimation . . . 14

3 Analysis of joint MRI reconstruction and motion estimation model **16**
3.1 Variational model . . . 16
3.2 Existence of minimizer . . . 17
3.3 Convexity . . . 19
3.4 The Kurdyka-Lojasiewicz property . . . 21

4 Optimization **26**
4.1 Discretization . . . 26
4.2 Splitting methods for convex optimization . . . 27
4.3 Optimization method 1: Alternating PDHG . . . 30
4.4 Nonconvex optimization methods . . . 34
4.5 Optimization method 2: PALM . . . 36
4.6 Comparison of optimization methods . . . 41

5 Results and discussion **44**
5.1 Datasets & performance measures . . . 44
5.2 Subsampling pattern and motion model . . . 48
5.3 Comparison of optimization method on synthetic datasets . . . 50
5.4 Application on experimental medical data . . . 57

6 Conclusion and outlook **59**

A Full results **64**

B Additional definitions and derivations **72**
B.1 Forward difference, central difference and backward difference . . . 72
B.2 Convex conjugates . . . 73
B.3 Adjoint operators . . . 73
B.4 Proximal maps . . . 74

Chapter 1

Introduction

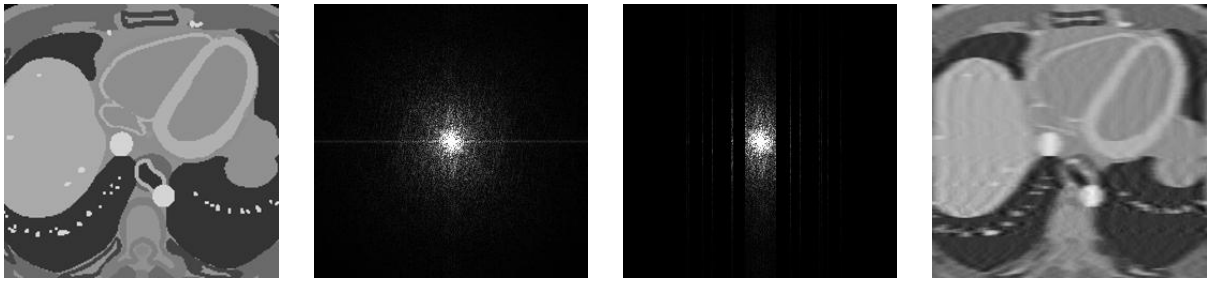
Since the discovery of Magnetic Resonance Imaging (MRI) in 1971 it is a well known form of medical imaging. MRI is based on the phenomenon called Nuclear Magnetic Resonance (NMR). In the presence of a strong magnetic field protons will align with this field. When this magnetic field is turned off, the protons start spinning back to their original position and while spinning they send out a signal. An MR image reflects the intensity of this signal. As the proton density varies for different substances, we can see the structure of the body and abnormalities in an MR image. However this spinning will only happen at a specific strength of the magnetic field, so by creating gradients in the magnetic field we can measure specific positions. The advantage of MR imaging is that no radiation is involved so it will not harm the human body.

The signal received during data acquisition is the Fourier transform of the image, thus the signal has the following form:

$$F(\gamma G_x t, \gamma G_y t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) e^{-i2\pi[x(\gamma G_x t)+y(\gamma G_y T)]} \mathrm{d}x\mathrm{d}y,$$

were $F(\gamma G_x t, \gamma G_y t)$ is the electromagnetic signal, $f(x,y)$ the desired image and G_x and G_y are the gradients of the magnetic field. A clear derivation and more details can be found at [55]. Call $k_x = \gamma G_x t$ and $k_y = \gamma G_y$, data acquisition in MRI is often called sampling in k-space. If the number of measurements in k-space equals the resolution of the desired image, the inverse Fourier transform of the measurement will give the image.

The limitation of MRI is that the measurement time is directly proportional to the number of measurements in k-space. Thus, to shorten the measurement time, subsampling is needed. However, subsampling leads to artefacts due to missing information, see Figure 1.1. As the measurement time for each point in k-space is limited by physical properties of the MRI scanner, research is mainly focused on reconstruction methods to overcome artefacts due to subsampling. Famous are parallel imaging and compressed sensing for static MRI reconstruction. Reconstruction techniques based on parallel imaging are SENSE [45] and GRAPPA [27], which make use of the fact that the spatial sensitivity for receiver coils varies for different positions of the coils in the MRI scanner. Compressed sensing [36] is based on the compressibility of images, by enforcing sparseness in an appropriate transform of the image.



(a) Ground truth image (b) Fully sampled k-space (c) Subsampled k-space (d) Direct inversion of subsampled image

Figure 1.1

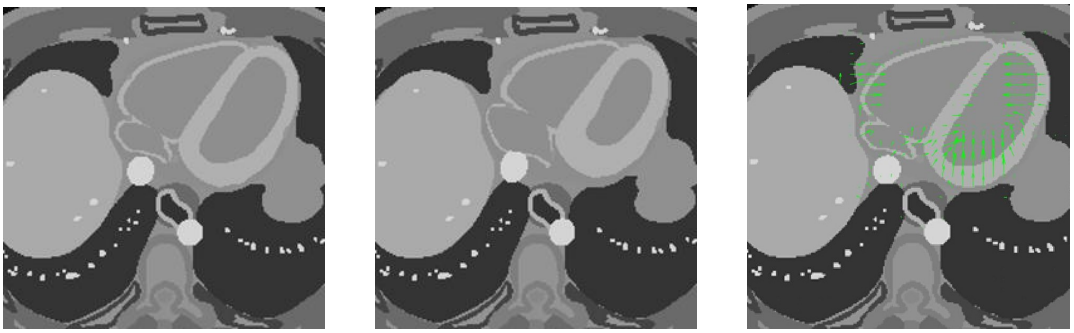
Dynamic MRI is used to capture objects in motion and offers a lot of opportunities. If we would be able to reconstruct fast enough, dynamic MRI could become real time. However dynamic MRI is even more time consuming than static MRI. A limitation in dynamic frame-by-frame reconstruction is the missing temporal information in the modeling. Examples where temporal information is modeled are the low-Rank plus sparse decomposition of [40], were the low-rank matrix captures the static background and the sparse matrix captures the dynamic foreground. Another example is the ICTGV method of [48], which uses stronger temporal regularization for the background and stronger spatial regularization for the dynamic foreground.

Motion estimation is one of the most studied tasks in both imaging and computer vision. In this thesis the motion is modeled via optical flow. Optical flow is defined as apparent motion between two consecutive frames, see Figure 1.2. Note that the optical flow, although closely connected, does not in general coincide with the actual physical motion in a sequence. Since it is a two-dimensional respresentation of the actual three-dimensional movement. Widely used and classical are the works of Lucas and Kanade [34] and Horn and Schunck [28]. Both employ the brightness constancy assumption which assumes that over time the brightness in an image remains constant when following the motion. Their work was fundamental for many other models and is also the basis for the motion estimation model in this thesis.

Prior to motion estimation, the image sequence is often preprocessed: naturally, the higher the quality of the image sequence, the higher the quality of the motion estimation. However, the reconstruction can also benefit from the inclusion of motion information. This is the motivation for developing joint reconstruction and motion estimation models. Dirks [18] developed a general joint reconstruction and motion estimation model, were one of the main contributions is the proof of existence of a minimizer. Frerking [25] extended the model of [18] with a non-linear term for motion estimation. Closely related is the work of Brune [11] where the motion estimation is based on optimal transport. Optical flow estimates the motion between two consequtive timesteps, optimal transport explains how the objects were transported in that timestep. The work of Lucka *et al.* [35] also employs the work of [18] and is applied to photoacoustic tomography.

Unfortunately the joint model by [18] is not convex and therefore the optimization is challenging. In the works of [18], [25] and [35] an alternating approach is proposed, which alternates between the two convex subproblems for reconstruction and motion estimation. However there is no guarantee that this alternating approach will converge: as shown by [44] there is a risk of circling infinitely without converging. To overcome this, various algorithms are developed for the minimization of nonconvex models. Their convergence result relies on the Kurdyka-Lojasowicz property and this result was for instance introduced by by Attouch *et al.* [2].

Another challenge is the evaluation of joint reconstruction and motion estimation models, since in general there is no ground truth available for the flow fields. This makes it hard to judge the quality of the outcome. In computer vision, various artificial datasets are developed with a ground truth for the flow fields. Well known datasets are: Middlebury [4], KITTI [38] and MPI Sintel [37].



(a) Image at t (b) Image at $t+\delta_t$ (c) Flow field

Figure 1.2

Contributions

In this thesis a joint reconstruction and motion estimation framework is developed for dynamic MRI to explicitly model the temporal coherence in a sequence. To address the challenging optimization of such a model, we will develop an algorithm related to recently developed nonconvex optimization algorithms. To evaluate the performance of the joint model for different optimization methods we develop synthetic datasets. These datasets are based on the MPI dataset [37] from computer vision, which has a ground truth for the flow field and the XCAT phantom [49], which is used for medical image reconstruction evaluation and made available by the radiotherapy group of the UMC Utrecht. Finally the model is applied to experimental medical data, also made available by the radiotherapy group of the UMC Utrecht, to show its potential for real word scenarios.

Structure

In chapter 2 we will define the joint reconstruction and motion estimation model. All components are explained in detail and also an overview of other possible choices is provided. In chapter 3 we will analyze the model and prove the existence of a minimizer, nonconvexity of the joint model and that the objective is a KL-function. In chapter 4 the optimization is treated: we will review important literature on both convex and nonconvex optimization and two algorithms are derived. In chapter 5 we will present the developed synthetic dataset and discuss the results of the joint reconstruction and motion estimation for both algorithms. We will compare the two algorithms in different scenarios to get insight about their differences. In chapter 6 the conclusion is given along with suggestions for further research.

Chapter 2

Model for joint reconstruction and motion estimation

In this thesis we consider the following general variational model for joint reconstruction and motion estimation:

$$\min_{u,\mathbf{v}} \quad \int_0^T G(u) + J(Au) + F(B\mathbf{v}) + M(u,\mathbf{v})\mathrm{d}t, \tag{2.1}$$

were $G(u)$ represents the data-fidelity for the image sequence u, which penalizes differences between the image sequence and the observed data. Since the problem of MRI reconstruction is ill-posed due to noise and subsampling we need to use regularization. $J(Au)$ represents the regularization term for the u and is used to incorporate a priori knowledge we have about the image. The problem of motion estimation is also ill-posed and hence we incorporate also a regularizer for the flow field $\mathbf{v}$, namely $F(Bv)$. The motion model is here represented by $M(u,\mathbf{v})$ which couples the image sequence and the flow field.

In this chapter we will define specific choices for $G(u)$, $J(Au)$, $F(B\mathbf{v})$ and $M(u,\mathbf{v})$ and discuss related previous work.

2.1 Data fidelity for MRI Reconstruction

Define the measurements from the MRI scanner as: $f : \Sigma \to \mathbb{C}$, with Σ the k-space. As explained in the introduction, the signal f received during data acquistion is the Fourier transform of the image. Following [23] we assume that the phase of the received signal by the MRI scanner is negligibly small and hence $u \in \mathbb{R}^2$. Define: $u : \Omega \to \mathbb{R}$, where $\Omega \subset \mathbb{R}^2$.

Since the Fourier transform acts on complex valued images, $u : \Omega \to \mathbb{C}$, we define the operator $\mathrm{Re}(\cdot)$ as follows:

$$\begin{aligned} \mathrm{Re} &: \mathbb{C} \to \mathbb{R}, \quad \mathrm{Re}(x + iy) = x, \\ \mathrm{Re}^* &: \mathbb{R} \to \mathbb{C}, \quad \mathrm{Re}^*(x) = x + i0. \end{aligned}$$

Since we will use subsampling to reduce measurement time, we define $P(\cdot)$ as the subsampling operator:

$$P(x) := \begin{cases} 1 & \text{if } x \in \pi(\Sigma) \\ 0 & \text{else} \end{cases},$$

with $\pi(\Sigma)$ the set of spatial coordinates in k-space that are sampled. So for static MRI reconstruction we can define the following forward operator:

Definition 1: Forward operator static MRI reconstruction
Define the operator $F : X(\Omega) \rightarrow Y(\Sigma)$*, with* Σ *the k-space as*

$$F(u) = P(\mathscr{F}(Re^*(u))),$$

with Re^* *the adjoint of the operator Re,* $\mathscr{F}(\cdot)$ *the Fourier Transform,* $P(\cdot)$ *the subsampling operator and X and Y the appropriate function spaces. Then the adjoint operator is:*

$$F^* = Re(\mathscr{F}^{-1}(P(f))),$$

with $f \in \Sigma$.

Since we want to reconstruct dynamic sequences we have to define the forward operator for dynamic MRI. The image sequence $u(\cdot, t)$ is a function of the space-time domain $\Omega \times [0, T]$ to $\mathbb{R} \times [0, T]$. Now we can define:

Definition 2: Forward operator dynamic MRI reconstruction
Define the operator $F : X(\Omega \times [0, T]) \rightarrow Y(\Sigma \times [0, T])$*, with* $\Sigma \times [0, T]$ *the k-t-space as*

$$F(u(\cdot, t)) = P(\mathscr{F}(Re^*(u(\cdot, t)))),$$

with $P(\cdot, t)$ *subsampling operator extended to:*

$$P(x, t) := \begin{cases} 1 & \text{if } x \in \pi(\Sigma, t) \\ 0 & \text{else} \end{cases},$$

with $\pi(\Sigma, t)$ *the set of coordinates in k-t space that are sampled.* $\mathscr{F}(\cdot, t)$ *the Fourier Transform and* $Re^*(\cdot, t)$ *as defined before, for each* t*. X and Y are the appropriate function spaces.*

So for each $t \in [0, T]$, the forward operator for dynamic MRI reconstruction is the forward operator for static MRI reconstruction. Hence frame-by-frame reconstruction is possible. Note that time is defined equally in both the image and measurement domain, so at each point in time we have a set of measurements.

In practice, $\pi(\Sigma, t)$ at time t will follow a pattern. Widely used are Cartesian, radial and spiral patterns: see Figure 2.1 for an example. Most simple is Cartesian sampling, where there is only subsampling in the phase-encoding direction. The classical Fourier transform can be used to model Cartesian subsampling. For radial and spiral subsampling the non-uniform Fourier transform must be applied or first a regridding procedure must be applied. As the optimal subsampling pattern is out of the scope of this thesis, only Cartesian subsampling is considered for simplicity.

As mentioned by [48], the noise present in MRI measurements is Gaussian, so we use the $L^2(\Omega)$ norm for the data-fidelity. The data-fidelity for MRI reconstruction becomes:

$$G(u(\cdot, t)) = \|F(u(\cdot, t) - f(\cdot, t)\|_2^2.$$

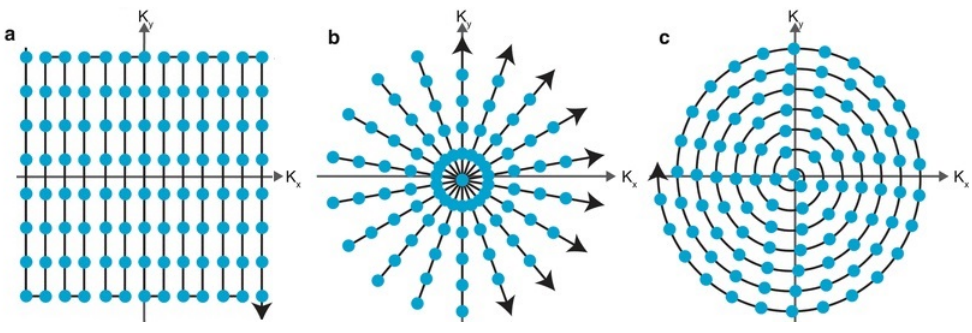


Figure 2.1: Examples of subsampling patterns, a. Cartesian, b. Radial, c. Spiral.

2.2 Regularizer for MRI reconstruction

Due to the presence of noise and subsampling, reconstructing an MR image is ill-posed and hence we have to add regularization for the image sequence $u(\cdot, t)$. In this section different regularizers will be discussed and we will define the regularizer $J(Au)$ in (2.1).

2.2.1 Compressed sensing

In image processing it is well known that images are sparse in some transforms. So if we transform an image with an appropriate transform we only need a small part of that data to reconstruct the full image. This is the idea behind compressed sensing, which was applied to MRI by Lustig *et al.* in [36]. To apply compressed sensing there are three requirements:

1. *The image is sparse in some transformation.*

   In [36] the finite difference, wavelet en discrete cosine transform are used.

2. *The artefacts due to subsampling are incoherent (noise-like) in this transform.*

   To fulfill this requirement random subsampling in the phase encoding direction (y) is used in [36].

3. *We can reconstruct by enforcing both the sparsity and data consistency with the acquired data.*

So the problem that is solved in compressed sensing for MRI can be written as:

$$\begin{aligned} \min_u \quad & \|\Phi(u)\|_1, \\ \text{s.t.} \quad & \|F(u) - f\|_2^2 < \varepsilon, \end{aligned} \tag{2.2}$$

were Φ is the sparsifying transform.Here we will use the Daubechies wavelet as the sparsifying transform.

We can rewrite problem 2.2 to an unconstrained optimization problem by introducing the constant α and moving the constraint to the objective:

$$\min_u \quad \frac{\lambda}{2}\|F(u(\cdot,t)) - f(\cdot,t)\|_2^2 + \alpha\|\Phi(u(\cdot,t))\|_1,$$

where enforcing sparsity in the $\Phi(u)$ transform is used as a regularizer.

2.2.2 TV and TGV regularization for MRI

A famous variational model is the ROF-model [47], were an L^2 data-fidelity is combined with total variation (TV) regularization. Total variation of a function is defined as:

$$TV(u) := \sup_{g \in C_0^\infty(\Omega;\mathbb{R}^d), \|g\|_\infty \leq 1} \int_\Omega u \nabla \cdot g \mathrm{d}x.$$

Depending on the inner norm chosen for $\|g\|_\infty$ we get either isotropic total variation, which favours rounded corners, or anisotropic total variations, which favours square corners. In this thesis isotropic TV-regularization is used, as shapes in biomedical MRI images often do not have sharp corners. For more details on total variations see [13]. TV-regularization has the property that it is edge preserving and results in piece-wise constant images. As MR images are in general not piece-wise constant, TV reconstructed images are often perceived as too cartoon-like or patchy. Another artefact from TV-regularization is the stair-casing effect.

To overcome this, Knoll *et al.* [29] propose total generalized variation (TGV) for MRI reconstruction. In their work they show that using the second order total generalized variation as regularizer gives better reconstructions for static MRI, compared to using TV-regularization. Second order TGV prefers constant and piecewise linear images. As the TGV is also a semi-norm of a Banach space, the analysis and optimization is comparable to the using total variation. The second order total generalized variation is defined as:

$$\mathrm{TGV}^2_\alpha(u) := \min_{u_1} \quad \alpha_1 \int_\Omega |\nabla u - u_1| \mathrm{d}x + \alpha_0 \int_\Omega |\mathscr{E}(u_1)| \mathrm{d}x,$$

with $\mathscr{E}(u_1) = \frac{1}{2}(\nabla u_1 + \nabla u_1^T)$.

According to [29], using $\alpha_0 = 2\alpha_1$ offers in practice good results, so using TGV-regularization instead of TV-regularization does for this parameter choice not result in more parameters to be tuned.

2.2.3 Low-rank plus sparse decomposition

In [40], Otazo *et al.* propose a low-rank plus sparse matrix decomposition as regularization for dynamic MRI reconstruction. The decomposition is performed on the space-time matrix U whose columns contain the image at each time step.

If we can decompose this space-time matrix in a part that represents the static background and a part that represents the dynamic foreground, then the static background will be a matrix of low column rank as each column is nearly the same. The matrix containing the dynamic information will now be defined as the sparse representation of this matrix using a temporal transform. The minimization problem for low-rank plus sparse decomposition is:

$$\min_{L,S} \quad \lambda_L \|L\|_* + \lambda_S \|TS\|_1 + \frac{1}{2}\|F(L+S) - f\|_2^2.$$

with F the forward operator for MRI reconstruction, f the measurements and L the low-rank space-time matrix modeling the static background. $\|\cdot\|_*$ is the nuclear norm, defined as:

$$\|A\|_* = \mathrm{trace}\left(\sqrt{A^* A}\right).$$

S is the space-time matrix modeling the dynamic foreground which is sparse in the transform $T(\cdot)$.

2.2.4 Infimal convolution of total generalized variation functionals for dynamic MRI (ICTGV)

Another possible regularizer is given by Schloegl *et al.* [48]. They argue that in practise it is not always feasible to decompose the image sequence in a static background and dynamic foreground as proposed by [40]. To generalize this idea they propose a model that optimally balances between strong spatial or strong temporal regularization of the sequence. This is done via infimal convolution of total generalized variation functionals (ICTGV).

The ICTGV is defined as:

$$\mathrm{ICTGV}^2_{\beta,\gamma} = \min_{v} \quad \beta \mathrm{TGV}^2_{\beta_1}(u-v) + \gamma \mathrm{TGV}^2_{\beta_2}(v),$$

were β_1 enforces a stronger temporal regularization. This model is implemented in the toolbox AVIONIC [48].

2.2.5 Overview of regularizers for MRI reconstruction

The options treated in this section are summarized in table 2.1. Since we want to incorporate the temporal coherence via joint reconstruction and motion estimation we will not use the low-rank plus sparse and ICTGV regularization since they also model temporal coherence.

In [18] TV-regularization was used as regularizer for the image sequence. We extend the regularization by adding the compressed sensing regularization term to obtain a suitable model for MRI reconstruction. So we define $J(Au)$ as:

$$J(Au) := \alpha_1 \|\nabla u(\cdot, t)\|_1 + \alpha_2 \|\Phi(u(\cdot, t)\|_1.$$

Name	Term	Section
Compressed sensing	$\|\Phi(u)\|_1$	2.2.1
Total variation	$\|\nabla u\|_1$	2.2.2
Total generalized variation	$\alpha_{11}\|\nabla u - u_2\|_1 + \alpha_{10}\|\mathcal{E} u_2\|_1$	2.2.2
Low rank + sparse	$\lambda_L \|L\|_* + \lambda_S \|TS\|_1$	2.2.3
ICTGV	$\min_v \beta \mathrm{TGV}^2_{\beta_1}(u-v) + \gamma \mathrm{TGV}^2_{\beta_2}(v)$	2.2.4

Table 2.1: Different choices for regularization $J(Au)$ of MRI reconstruction

2.3 Model for motion estimation

In this section the goal is to define the motion model $M(u, v)$. As mentioned in the introduction we will use optical flow to model the motion. Optical flow is defined as the apparant motion between two consecutive frames, see Figure 2.2.

A common assumption is that the intensity of the image stays constant over time, which means that after δ_t the following equation must be satisfied:

$$u(\mathbf{x}, t) = u(\mathbf{x} + \mathbf{v}(\mathbf{x}, t), t + \delta_t), \tag{2.3}$$

with $\mathbf{v}(\mathbf{x}, t) : \Omega \times [0, T] \to \mathbb{R}^2 \times [0, T]$ the flow field which represents the displacement. If the movement is small, the linearization of 2.3 will lead to the well-known optical flow constraint (OFC):

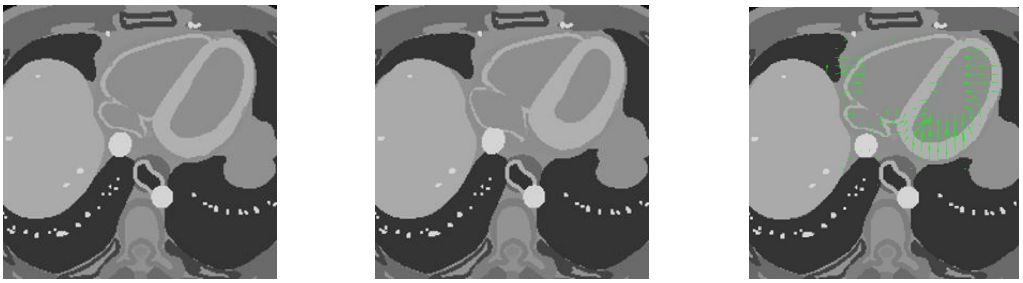
$$\partial_t u(\mathbf{x}, t) + \nabla u(\mathbf{x}, t) \cdot \mathbf{v}(\mathbf{x}, t) = 0,$$

which holds pointwise for each $(\mathbf{x}, t)$.

Now we define:

$$M(u, \mathbf{v}) := \frac{\gamma}{p} \|\partial_t u(\cdot, t) + \nabla u(\cdot, t) \cdot \mathbf{v}(\cdot, t)\|_p^p, \tag{2.4}$$

with $p \in \{1, 2\}$. Choosing $p = 1$ will result in a motion model that is more robust to noise as presented by [3]. Choosing $p = 2$ result in a smooth functional $M(u, \mathbf{v})$ which is favorable from an optimization point of view. Therefore both choices are considered.



(a) Image at t (b) Image at $t + \delta_t$ (c) Flow field

Figure 2.2

The optical flow constraint assumes that over time the intensity in an image remains constant when following the motion, however in changing illumination conditions this condition is violated. Another assumption is that the movement between two frames is small compared to the pixel size, this assumption is in practice also easily violated. In this thesis we will use the model for motion estimation as presented above. In the remainder of this section we will give a summary of related motion estimation models.

Mass preservation
A more general approach is to assume that the total brightness in the image is constant over time. This will lead to the mass preservation constraint.

$$\partial_t u + \nabla(u\mathbf{v}) = 0.$$

This constraint is known in literature as the continuity equation. A derivation can be found in [18]. Note that these two constraints intersect when $\nabla \cdot \mathbf{v} = 0$. So for divergence-free flow fields both the optical flow and the mass preservation constraints are satisfied at the same time. A divergence-free flow fields corresponds to an incompressible flow, so there are no sources or sinks.

Preservation of intensity derivatives
Even more general is the assumption of preservation of the derivatives of intensity, as presented in [11]. Due to illumination changes the optical flow constraint will not hold, however the gradient of the image will remain constant in the case of constant illumination changes.

Non-linear motion model
Recall that the linearization of (2.3) holds if the movement is relatively small. However in many real life applications this is not the case and hence the obtained model for the motion is not valid. To deal with this in [25], [42] and [50] variational models with non-linear models for motion estimation via optical flow are presented. However these models are more difficult from a computational point of view.

In [42] the motion is estimated with a course-to-fine scheme. The main idea behind such a scheme is to create a series of down-sampled images. We start by estimating the motion on the most coarse scale, where the motion is now relatively small. Then we go step-by-step to a more fine scale, using the flow field estimated in the previous scale to get the flow field on the current scale.

In [50] an alternative to a coarse-to-fine scheme is given. They introduce an auxiliary flow field $\mathbf{w}$ to decouple the data-fidelity and regularization, which simplifies the optimization.

Image registration
A different way of modelling motion estimation is image registration. Here the goal is to find a deformation $y : \Omega \to \mathbb{R}^d$ such that for two given images I_1 and I_2, $I_2(y)$ approximates I_1 as good as possible. So instead of a flow field we are looking for a deformation to explain the movement between two images.

Time-dependent motion model
A possible extension to the motion model presented here is to make the motion estimation time dependent. Optical flow describes how to go from one to the next consecutive frame in time. However over time the flow field should also satisfy some regularity constraints. For example one could assume that the flow field should change smoothly over time. Incorporating temporal coherence of the flow fields can be modelled via optimal transport as presented in [11] or via optimal control optical flow [9], [16].

2.4 Regularizer for motion estimation

Classical and widely used is the work of Horn and Schunck [28] who use as motion model the optical flow constraint (2.4) with $p = 2$. The model is equipped with a regularizer since the problem of determining optical flow is ill-posed. To illustrate this, we refer to the so-called aperture problem of [5]. Take a straight moving edge that is observed through a narrow aperture, then only the perpendicular component of motion can be captured. Hence it is impossible to distinguish between different kind of motions, see figure 2.3.

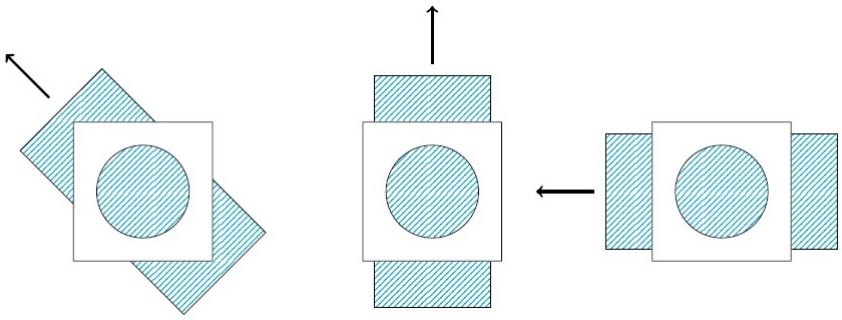


Figure 2.3: Illustration of the aperture problem from [25]. The different type of motions in the aperture are not distinguishable.

The variational model of Horn and Schunck for determining optical flow has the following form:

$$\min_{\mathbf{v}} \quad \int_0^T \frac{1}{2}\|\partial_t u(\cdot,t) + \nabla u(\cdot,t)\cdot\mathbf{v}(\cdot,t)\|_2^2 + \frac{\beta}{2}\|\nabla\mathbf{v}(\cdot,t)\|_2^2 \mathrm{d}t.$$

Using $\|\nabla\mathbf{v}(\cdot,t)\|_2^2$ as a regularizer will result in smooth flow fields. This is a reasonable assumption since an object in a sequence stays connected over time and therefore will have a smooth flow field.

However in the case of multiple moving objects the assumption of a smooth velocity field is not realistic. It would be desirable to allow discontinuites in the flow field on the edges of the objects. Papenberg, Weickert *et al.* [42] proposed TV-regularization to overcome this problem. TV-regularization results in constant areas and discontinuities along edges of objects in the flow field.

In the data considered here there is often a static background and a moving foreground, therefore smooth flow fields are not realistic and we define the following regularization for the flow field $\mathbf{v}$:

$$F(B\mathbf{v}) := \|\nabla\mathbf{v}(\cdot,t)\|_1. \tag{2.5}$$

There are many more possible regularizers, e.g. in [52] PDE-based regularizers are discussed. There regularizers fall into two categories, namely image- or flow-driven.

Image-driven regularizers use the idea that motion boundaries are a subset of the image boundaries. So one way to combine smooth flow fields within an object and sharp edges on the motion boundaries, is to add a function to the regularizer that is small at the image boundaries. A risk of this approach is to get an oversegmented flow field for textured objects.

Another type of regularizers are flow-driven. They also make us of a function that is small at the edges, but now at the edges of the flow field instead of the edges of the image. This will prevent oversegmentation but will have less sharp edges. So the choice depends strongly on the type of problem at hand.

Finally we want to point out the recent development of deep learning methods for motion estimation. A common challenge for these methods is that they require a large set of training data with a ground truth for the flow field, which is hard to obtain. In [20] the authors created an artificial dataset to train a CNN for motion estimation. In [37] the authors trained a CNN unsupervised to address the lack of a ground truth flow field and they learn both the forward and backward flow and make sure this matches. Here we only consider the simple regularizer defined in (2.5) to keep the model simple and general.

Chapter 3

Analysis of joint MRI reconstruction and motion estimation model

In the previous chapter we defined all the main ingredients of our joint model. In the chapter this model will be analyzed. First the model will be transferred to a more general function spaces setting, in which existence of a global minimizer is proved. We will also show that the joint model is not convex in both variables u and $\mathbf{v}$.. Finally the Kurdyka-Lojasiewicz property is treated and we prove that our model satisfies this property, which is a key element for convergence analysis in nonconvex optimization.

3.1 Variational model

Our aim is to minimize the following variational model for joint MRI reconstruction and motion estimation.

Joint reconstruction and motion estimation model

$$\min_{u,\mathbf{v}} \quad \int_0^T G(u) + J(Au) + F(B\mathbf{v}) + M(u,\mathbf{v})\mathrm{d}t. \tag{3.1}$$

with:

$$\begin{aligned} G(u) &= \frac{\lambda}{2}\|F(u(\cdot,t)) - f(\cdot,t)\|_2^2, \\ J(Au) &= \alpha_1\|\nabla u(\cdot,t)\|_1 + \alpha_2\|\Phi(u(\cdot,t))\|_1, \\ F(B\mathbf{v}) &= \beta\|\nabla\mathbf{v}(\cdot,t)\|_1, \\ M(u,\mathbf{v}) &= \frac{\gamma}{p}\|\partial_t u(\cdot,t) + \nabla u(\cdot,t)\cdot\mathbf{v}(\cdot,t)\|_p^p \qquad p\in\{1,2\}. \end{aligned}$$

with image sequence $u(x,t):\Omega\times[0,T]\to\mathbb{R}$ and velocity field $\mathbf{v}(x,t)=(v_1,v_2):\Omega\times[0,T]\to\mathbb{R}^2$ for $\Omega\in\mathbb{R}^d$.

Recall that $F(\cdot,t)$ is the forward operator for dynamic MRI, $f(\cdot,t)\in\Sigma\times[0,T]$ are the (subsampled) measurements in k-t space and $\Phi(\cdot,t)$ is the wavelet transform.

Remark: $F(\cdot,t)=P(\mathscr{F}(\mathrm{Re}^*(u(\cdot,t))))$ is time-dependent, as $P(\cdot)$ is dependent on time, but the operator works on single time frames. Due to $\partial_t u(\cdot,t)$ all frames in the sequence are related and hence we have to minimize for all $t\in[0,T]$ at once.

3.2 Existence of minimizer

In this section we will prove the existence of a global minimizer. This proof is an extension of the proof given in [12]. First we will state the important definitions and theorems needed for the proof. Then we will give an outline of the proof given in [12] and state what is left to prove for the model presented here. We conclude this section by proving this.

3.2.1 Definitions and theorems

To prove the existence of a minimizer we will use the fundamental theorem of optimization:

Theorem 1: Fundamental theorem of optimization
Let (U,τ) be a metric space and $J : U \to \mathbb{R} \cup \infty$ a functional on U. Moreover let J be lower semi-continuous and coercive. Then there exists a minimizer $\bar{u} \in U$ such that:

$$J(\bar{u}) = \inf_{u \in U} J(u).$$

To use this theorem we need lower semicontinuity and coercivity, which is defined as follows.

Definition 4: lower semicontinuity
In U a Banach space with topology τ, functional $J(U,\tau) \to \bar{\mathbb{R}}$ is called lower semicontinuous at u if:

$$J(u) \leq lim\ inf_k J(u_k) \quad as\ k \to \infty, \forall\, u_k \to u \quad in\ topology\ \tau.$$

Definition 5: coercivity
Let (U,τ) we a topological space and $J : U \to \mathbb{R} \cup \infty$ a functional on U. We call J coercive if it has compact sub-level sets. This means there exists an $\alpha \in \mathbb{R}$ such that the set:

$$S(\alpha) := \{u \in U | J(u) \leq \alpha\}$$

is not empty and compact in τ.

To prove coercivity often the theorem of Banach-Alaoglu is employed.

Theorem 2: Banach-Alaoglu
Let U be the dual of a Banach space and $C > 0$. Then the set:

$$\{u \in U : \|u\|_U \leq C\}$$

is compact in the weak-* topology.

3.2.2 Outline of the proof

In [12] the existence of a minimizer is proved for the following joint model:

$$\begin{aligned} \int_0^T \frac{1}{2}\|Ku(\cdot,t) - f(\cdot,t)\|_2^2 + \alpha\|\nabla u(\cdot,t)\|_1 + \beta\|\mathbf{v}(\cdot,t)\|_1 \mathrm{d}t \\ \text{s.t. } \partial_t u(\cdot,t) + \nabla u(\cdot,t)\cdot\mathbf{v} = 0 \quad \text{in } D'(\Omega \times [0,T]), \end{aligned} \tag{3.2}$$

There are two differences between this joint model and our joint model (3.1):

- In (3.2) the motion model appears as a constraint which must be satisfied in the distributional sense. This is equivalent to making (3.2) an unconstraint minimization problem using a L^p penalty term, with $p \in \{1,2\}$, so this is equivalent to our motion model in (3.1).
- In (3.1) we have an additional regularization term for u. We will call this term:

$$J_{CS} := \int_0^T \|\Phi(u)\|_1 \mathrm{d}t.$$

To prove existence of a minimizer for (3.2) the following steps must be followed:

1. Transferring (3.2) to the right function space setting,
2. Checking all assumptions made by [12] for (3.2),
3. Proving lower semicontinuitiy for (3.2),
4. Proving the coercivity of (3.2) using the theorem of Banach-Alaoglu,
5. Proving convergence of the constraint in (3.2),

then from the fundamental theorem of optimization the existence of a minimizer will follow. In the following section we will take these steps to prove existence of a minimizer.

3.2.3 Proof of existence of a minimizer

1. Transferring (3.1) **to the right function space setting**
Following [12] we can write:

$$\begin{aligned} J(u,\mathbf{v}) = \min_{u,\mathbf{v}} & \int_0^T \frac{\lambda}{2}\|F(\cdot,t)u(\cdot,t) - f(\cdot,t)\|_2^2 \mathrm{d}t + \alpha_1 \int_0^T |\nabla u(\cdot,t)|_{BV}^p \mathrm{d}t + \alpha_2 \int_0^T \|\Phi(u(\cdot,t))\|_1 \mathrm{d}t \\ & + \beta \int_0^T |\nabla_x \mathbf{v}(\cdot,t)|_{BV}^q \mathrm{d}t, \\ & \text{s.t. } \partial_t u(\cdot,t) + \nabla u(\cdot,t)\cdot \mathbf{v}(\cdot,t) = 0 \qquad \text{in } D'(\Omega\times[0,T]). \end{aligned}$$

with $(u,\mathbf{v})$ in the set:

$$\left\{(u,\mathbf{v}) : u \in L^{\hat{p}}(0,T:BV(\Omega)), \mathbf{v}\in L^q(0,T;BV(\Omega)), \nabla\cdot\mathbf{v}\in L^{p^*s}(0,T:L^{2k}(\Omega)), \partial_t u + \nabla u\cdot\mathbf{v} = 0\right\},$$

for $s>1$, $k>2$ and p^* such that $\frac{1}{p}+\frac{1}{p^*}=1$ and $\hat{p}=\min\{p,2\}$.

2. Checking assumptions
In the proof of [12] four assumptions are made:

1. The data f is affected by additive Gaussian noise.
2. Finite speed of the velocity field, i.e.:

$$\|\mathbf{v}\|_\infty \le c_v < \infty \quad \text{a.e. } \Omega\times[0,T].$$

3. Bound on the compressibility of $\mathbf{v}$, which means bounding $\nabla\cdot\mathbf{v}$.
4. For $F(\mathbf{1}_t)\neq 0$, $\forall t\in[0,T]$, with $\mathbf{1}_t$ the indicator function.

All four assumptions follow naturally for the model presented here. As mentioned in chapter 2, noise that affect MRI data is indeed Gaussian noise, hence the choice for an L^2 data-fidelity term. We assume that the object that is scanned will remain visible during data acquisition, thus the velocity field is bounded by the resolution of the resulting image. The image is defined on a bounded subset $\Omega\in\mathbb{R}^d$. Hence we have indeed finite speed.

Bounding compressiblity relates to bounding the sources and sinks in the flow field, which is reasonable. Finally for the forward operator it must hold that $F(\mathbf{1}_t)\neq 0$, $\forall t\in[0,T]$. So for $\mathbf{1}_t$, with arbitrary $t\in[0,T]$ we get:

$$Re^*(\mathbf{1}_t) = \mathbf{1}_t + 0i,$$

since the indicator function is a real function,

$$\mathscr{F}(\mathbf{1}_t) = \mathbf{1}_t \cdot \delta(\Sigma),$$

with $\delta(\Sigma)$ the Dirac delta function on the k-space Σ. The Dirac delta function is defined as:

$$\delta(x) = \begin{cases} \infty & \text{if } x = 0 \\ 0 & \text{else} \end{cases}$$

Now for $P(\delta(\Omega))$ at $t \in [0, T]$ to be nonzero, $0 \in \pi(\Sigma)$ at time t. We will see later this will always be the case for the subsampling pattern considered in this thesis, as most of the energy of the image in the Fourier domain is in the center.

3. Lower semicontinuity
In [12] the lower semicontinuity of (3.2) is proved. The term J_{CS} is an affine norm of the form $\|\Phi(u)\|_1$ and hence lower semicontinuous. Since the sum of lower semicontinuous functionals is also lower semicontinuous, we proved lower semicontinuity for (3.1).

4. Coercivity
In [12] coercivity is proved for (3.2) with the additional remark that the functional (3.2) is coercive for any regularizer $J(u(\cdot, t))$ for which holds:

$$R(u(\cdot, t)) \geq |u(\cdot, t)|_{BV}^p,$$

so if we can prove this inequality, coercivity will follow.

We have that:

$$R(u\cdot, t) := |u(\cdot, t)|_{BV}^p + \|\Phi(u(\cdot, t))\|_1,$$

so we must prove

$$\|\Phi(u(\cdot, t))\|_1 = \int_0^T |\Phi(u(\cdot, t))| \mathrm{d}t \geq 0,$$

which holds by the definition of a norm, which concludes the prove for coercivity.

5. Convergence of the constraint
This is proved in [12] and hold also for our model (3.1).

Now the existence of a minimizer follows from the fundamental theorem of optimization. □

As we will see in the next section, the functional (3.1) is nonconvex for u and $\mathbf{v}$, so we are not able to prove uniqueness of the minimizer.

3.3 Convexity

For $J(u, \mathbf{v})$ (3.1) to be a convex functional the following definition from [46] must hold:

Definition 6: Convex functional
Let U be a Banach space, $\Omega \subset U$ a convex subset and $J : \Omega \to \mathbb{R} \cup \infty$ a functional. J is convex if the inequality:

$$J(\alpha u_1 + (1-\alpha) u_2) \leq \alpha J(u_1) + (1-\alpha) J(u_2) \tag{3.3}$$

holds for all $u, v \in \Omega$ and $\alpha \in [0, 1]$. J is strictly convex if (3.3) holds with strict inequality and $\alpha \in (0, 1)$.

Following [46], we know that:

Lemma 1: Convex functionals
The following holds:

1. *For any norm $J(u) = \|u\|$ is convex. Moreover for $p \geq 1$, $J(u) = \|u\|^P$ is convex.*
2. *Any affine norm $J(u) = \|Ku - f\|$ is convex for arbitrary operator K.*

3. *For two convex functions u_1, u_2 it holds: $u_1 + u_2$ is convex.*
4. *The integral of the sum of two functions u_1, u_2 is convex.*

Proof: The proof of 1.1,1.2 and 1.3 can be found in [46]

For 1.4:

$$\begin{aligned} J(\alpha u_1 + (1-\alpha)u_2) &= \int_0^T \alpha u_1 + (1-\alpha)u_2 \mathrm{d}t \\ &= \int_0^T \alpha u_1 \mathrm{d}t + \int_0^T (1-\alpha)u_2 \mathrm{d}t \\ &= \alpha \int_0^T u_1 \mathrm{d}t + (1-\alpha)\int_0^T u_2 \mathrm{d}t \\ &= \alpha J(u_1) + (1-\alpha)J(u_2). \end{aligned}$$

□

Using lemma 1.3 and 1.4 we can analyze the convexity of (3.1) term by term. The data-fidelity term for u, the regularization term for u and the regularization term for $\mathbf{v}$ are all norms or affine norms. However:

$$J_{OFC} = \partial_t u(\cdot, t) + \nabla u(\cdot, t) \cdot \mathbf{v}(x, t) \tag{3.4}$$

this is the only term in (3.1) were u and $\mathbf{v}$ are coupled. As it turns out this term is not jointly convex for u and $\mathbf{v}$.

Theorem 4: Nonconvexity joint model
The term (3.4) *is not jointly convex for u and $\mathbf{v}$. As a consequence the model* (3.1) *is not jointly convex.*

Proof:
For J_{OFC} to be jointly convex for u and $\mathbf{v}$ the following must hold:

$$J_{OFC}(\alpha u + (1-\alpha)u, \alpha \mathbf{v} + (1-\alpha)\mathbf{v}) \leq \alpha J_{OFC}(u, \mathbf{v}) + (1-\alpha) J_{OFC}(u, \mathbf{v}),$$

for all $u \in L^p(0, T; BV(\Omega))$, $\mathbf{v} \in L^q(0, T; BV(\Omega))$ and $\alpha \in [0, 1]$.

Now:

$$\begin{aligned} J_{OFC}(\alpha u_1 + (1-\alpha)u_2, \alpha v_1 + (1-\alpha v_2)) =& \alpha \partial_t u_1 + (1-\alpha)\partial_t u_2 + \alpha^2 \nabla_x u_1 \cdot \mathbf{v}_1 + (1-\alpha)^2 \mathbf{v}_2 \\ &+ \alpha(1-\alpha)(\nabla_x u_1 \cdot \mathbf{v}_2 + \nabla_x u_2 \cdot \mathbf{v}_1), \end{aligned} \tag{3.5}$$

$$\alpha J_{OFC}(u_1, \mathbf{v}_1) + (1-\alpha) J_{OFC}(u_2, \mathbf{v}_2) = \alpha \partial_t u_1 + (1-\alpha)\partial_t u_2 + \alpha \nabla_x u_1 \cdot \mathbf{v}_1 + (1-\alpha)\mathbf{v}_2. \tag{3.6}$$

For J_{OFC} to be convex, (3.5)≤(3.6) must hold. For $\alpha \in [0, 1]$ the following holds:

$$\begin{aligned} \alpha^2 &\leq \alpha, \\ (1-\alpha)^2 &\leq (1-\alpha). \end{aligned}$$

Hence we must prove:

$$\alpha(1-\alpha)(\nabla_x u_1 \cdot \mathbf{v}_2 + \nabla_x u_2 \cdot \mathbf{v}_1) \leq 0,$$

for all $\alpha \in [0, 1]$ and $u \in L^p(0, T; BV(\Omega))$ and $\mathbf{v} \in L^q(0, T; BV(\Omega))$. For $\alpha = 0$ or $\alpha = 1$ this inequality is satisfied, however for $\alpha \in (0, 1)$, this is not the case. Take for example $\mathbf{v}_1 = [1, 0]$ and $\mathbf{v}_2 = [0, 1]$ for all $\mathbf{x} \in \Omega$ and all $t \in [0, 1]$ and $u_1 := x$ and $u_2 := y$, were $\Omega \in \mathbb{R}^{2+}$. Then we get for every $x, y, t \in Q$:

$$\alpha(1-\alpha)\left(\begin{bmatrix}1 & 0\end{bmatrix} \cdot \begin{bmatrix}1 \\ 0\end{bmatrix} + \begin{bmatrix}0 & 1\end{bmatrix} \cdot \begin{bmatrix}0 \\ 1\end{bmatrix}\right) = 2\alpha(1-\alpha),$$

which is greater than zero for every $\alpha \in (0, 1)$. Hence J_{OFC} is not jointly convex and therefore (3.1) is not. □

3.3.1 Biconvexity

Although (3.1) is not jointly convex, the energy is biconvex. Biconvexity is defined as follows.

Definition 7: Biconvex sets and functionals
Let U, V be a Banach space, $\Omega \subseteq U \times V$ a biconvex subset and $J : \Omega \to \mathbb{R} \cup \infty$ a functional.

A set $B \subset U \times V$ is a biconvex subset on $U \times V$ if:

1. *for every $u \in U$ and fixed $\hat{v} \in V$, $B_{\hat{v}}$ is a convex set.*
2. *for every $v \in V$ and fixed $\hat{u} \in U$, $B_{\hat{u}}$ is a convex set.*

A functional J is biconvex if:

1. *$\forall u \in U$ and fixed $\hat{v} \in V$, $J(u, \hat{v})$ is a convex functional.*
2. *$\forall v \in V$ and fixed $\hat{u} \in U$, $J(\hat{u}, v)$ is a convex functional.*

Now we can prove the biconvexity of the joint model.

Theorem 5: Biconvexity of joint model
The model (3.1) is biconvex.

Proof: For

$$J_{OFC} = \partial_t u(\cdot, t) + \nabla u(\cdot, t) \cdot \mathbf{v}(x, t)$$

with fixed $\hat{u}$ and respectively $\hat{\mathbf{v}}$, we can write:

$$J_{OFC}(\hat{u}, \mathbf{v}) = A_u \mathbf{v} - f_u$$
$$J_{OFC}(u, \hat{\mathbf{v}}) = A_{\mathbf{v}} u$$

with $A_u = \nabla u$, $f_u = \partial_t u$ and $A_v = [\partial_t + v_x \partial_x + v_y \partial_y]$. Using Lemma 1.1 and 1.2, J_{OFC} is biconvex. □

3.4 The Kurdyka-Lojasiewicz property

Since our model for joint reconstruction and motion estimation is not jointly convex for both variables u and $\mathbf{v}$ the optimization is challenging. The key property to prove convergence in nonconvex optimization is the Kurdyka-Lojasiewicz property. In this section this property is stated and proved for the discretization of the functional (3.1).

The Kurdyka-Lojasiewicz (KL) property was first introduced by Lojasiewicz [32] for real analytic functions. Kurdyka [30] extended the property to functions definable on the minimal o-structure. Finally Bolte *et al.* [7] extended the KL property for nonsmooth subanalytic functions.

For real analytic functions we can define the KL-property as follows.

Definition 8: KL-property for real anaytic functions [32]
A function $\psi(x)$ satisfies the KL property at point $\bar{x} \in Dom(\partial\psi)$ if in a neighborhood U of $\bar{x}$, there exists a function $\phi(s) = cs^{1-\theta}$ for some $c > 0$ and $\theta \in [0, 1)$ such that the KL inequality holds:

$$\phi'(|\psi(x) - \psi(\bar{x})|)\text{dist}(0, \partial\psi(x)) \geq 1 \quad \text{for any } x \in U \cap \text{dom}(\partial\psi) \text{ and } \psi(x) \neq \psi(\bar{x}). \tag{3.7}$$

where $\text{dist}(x, S) := \inf_y\{\|y - x\| \,|\, y \in S\}$.

This definition can be extended for general proper lower semicontinuous functions. Take $\eta \in (0, +\infty]$. Take $\phi : [0, \eta) \to \mathbb{R}^+$ a concave continuous function such that:

1. $\phi(0) = 0$,
2. ϕ is C^1 on $(0, \eta)$ and continuous at 0,
3. for all $s \in (0, \eta) : \phi'(s) > 0$.

Now we can state:

Definition 9: KL-property for proper lower semicontinuous functions [7]
Let $\psi(x) : \mathbb{R}^d \to (-\infty, +\infty]$ be proper and lower semicontinuous. The function ψ has the KL property at $\bar{x} \in dom(\partial\psi)$ if there exist a η as defined before and a neighborhood U of $\bar{x}$ and a function ϕ as defined before, such that for all:

$$x \in U \cap [\psi(\bar{x}) < \psi(x) < \psi(x) + \eta]$$

the following inequality holds:

$$\phi'(\psi(x) - \psi(\bar{x}))\, dist(0, \partial\psi(x)) \geq 1 \tag{3.8}$$

We say that a function ψ has the KL-property or is a KL-function if it satisfies definition 6 or 7 at each point of $\text{dom}(\partial\psi)$.

3.4.1 KL-property for a two-dimensional problem

To get a better intuition for the KL-property we will consider the following two-dimensional problem:

$$\min_{(x,y)\in\mathbb{R}^2} \psi(x, y) = (xy - 1)^2. \tag{3.9}$$

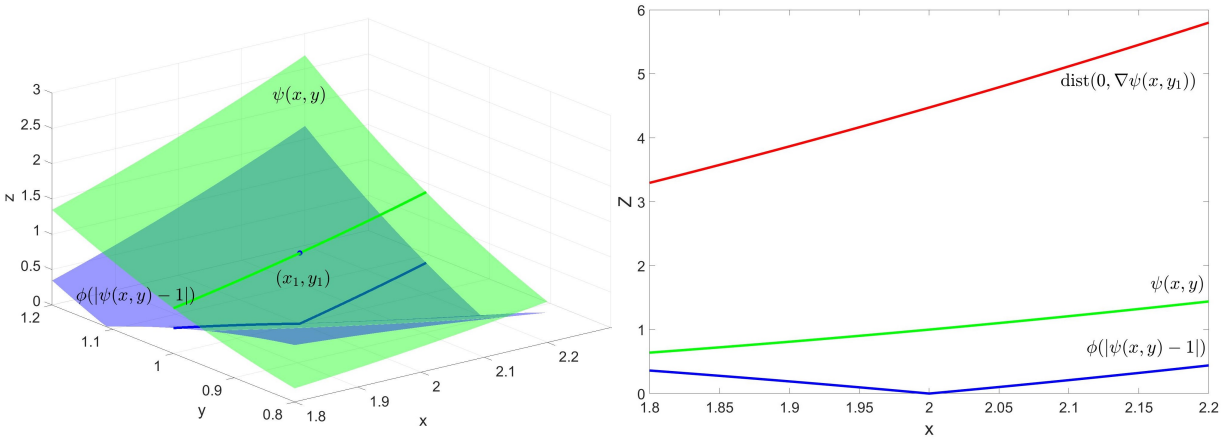
This function is nonconvex in both (x, y), but convex for fixed x or y, hence it is biconvex. We will analyze the KL-property for this function. Note that this is a polynomial function and hence real analytic, so we can use definition 8. The critical points of (3.9) are given by:

$$\nabla\psi = \begin{bmatrix} 2y(xy-1) & 2x(xy-1) \end{bmatrix},$$

so $\psi(x, y)$ is minimal for $y = \frac{1}{x}$.

First consider (x_1, y_1) for which holds $y_1 \neq \frac{1}{x_1}$. This is a noncritical point, which makes it easy to find a neighborhood U_1 were $\text{dist}(0, \nabla\psi(x)$ is not close to zero. Take for example $(x_1, y_1) = (2, 1)$, with $\psi(2, 1) = 1$. Then we can take $c = 1$ and $\theta = 0$, which gives $\phi = s$, so $\phi' = 1$. In figure 3.1a we see in green $\psi(x, y)$ in the neighborhood U_1 and in blue $\phi(|\psi(x, y) - 1|)$. In figure 3.1b we see in red $\text{dist}(0, \nabla\psi(x, y_1)$, for the cross-section (x, y_1) which is clearly greater than one and hence we fulfill the KL-inequality (3.7).

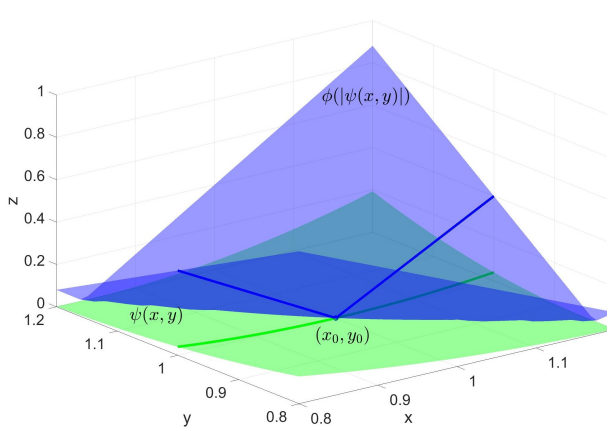
Near a critical point (x_0, y_0) for which holds $y_0 = \frac{1}{x_0}$. $\text{dist}(0, \nabla\psi(x, y)$ will become close to zero in a neighborhood U_0. To fulfill the KL-property we must choose ϕ such that the reparametrization $\phi(|\psi(x)|)$ is *sharp* enough. Sharpness is important since the derivative of ϕ times the small distance between zero and the gradients of ψ must be greater or equal to one. Take for example $(1, 1)$, were $\psi(1, 1) = 0$ and choose $\phi = 2s^{0.5}$. In figure 3.2a we see in green $\psi(x, y)$ near $(1, 1)$ and in blue $\phi(|\psi(x, y)|)$, which is indeed sharper. In figure 3.2b, we see again in red $\text{dist}(0, \nabla\psi(x, y)$ at $y = 1$ near $x = 1$ and we see that this distance is smaller than one. In yellow we see the KL-inequality, which is for $y = 1$ near $x = 1$, greater than one.



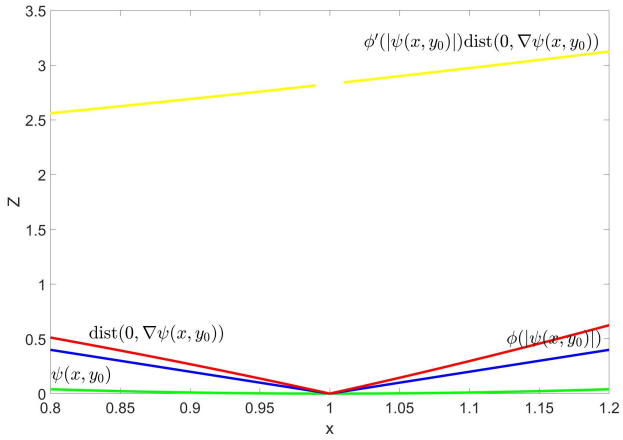
(a) Plot of $\phi(|\psi(x, y)|)$ near $(2, 1)$, with $\phi = s$

(b) cross-section at $y = 1$ around $x = 2$

Figure 3.1



(a) Plot of $\phi(|\psi(x,y)|)$ and $\psi(x,y)$ near $(1,1)$



(b) cross-section at $y = 1$ around $x = 1$

Figure 3.2

From this example we see that it is easy to establish the KL-property near non-critical points. Near critical points we must choose ϕ such that the reparametrization is sharp enough to compensate for the small gradients such that the KL-property is satisfied. In figure 3.3 a plot of $\psi(x,y)$ is shown with both neighborhoods U_1 and U_0. Here we clearly see that near the non-critical point, the function is quite steep and near the critical point it is not. To prove convergence to a minimum for a nonconvex problem, the KL-inequality ((3.7),(3.8)) will be used to obtain a bound for the sequence generated by the algorithm.

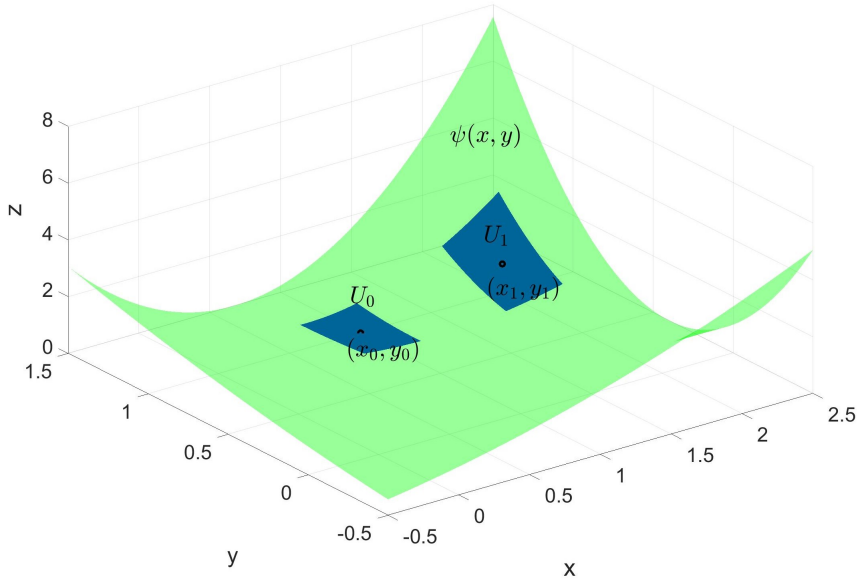


Figure 3.3: Plot of $\psi(x,y)$ with points (x_1,y_1) and (x_0,y_0) and their neighborhoods U_1 and U_0.

3.4.2 KL-property for joint model

To make use of algorithms for optimization of nonconvex functions, we need to prove that the discretization of the model for joint reconstruction and motion estimation is a KL-function.

We will prove that the KL-property holds for $p = 2$. The discretization will be defined in the next chapter,

so here we only state the resulting function:

$$\min_{u\in\mathbb{R}^{N\times t},\mathbf{v}\in\mathbb{R}^{M\times t}} J(u,\mathbf{v}) = \sum_{t=0}^{T}\frac{\lambda}{2}\|Fu_t - f_t\|_2^2 + \alpha_1\|\nabla u_t\| + \alpha_2\|\Phi u_t\|_1 + \sum_{t=0}^{T-1}\beta\|\nabla\mathbf{v}_t\|_1 + \frac{\gamma}{2}\|\partial_t u_t + \nabla u_t\cdot\mathbf{v}_t\|_2^2. \quad (3.10)$$

To prove that this is a KL-function we will use the following lemma.

Lemma 2: KL functions
The following functions satisfy the KL-property:

1. *Real analytic functions*
2. *Semialgebraic functions*
3. *Sum of real analytic and semialgebraic functions*

Proof: For the proof of 1 see the work of Lojasiewicz [32], the proof of 2 follows from the work of Bolte *et al.* [7] since semialgebraic functions are subanalytic. For 3 we refer to [6] were they prove that the sum of two subanalytic functions is subanalytic. Note that both real analytic and semialgebraic functions are subanalytic [6]. □

First we will define what real analytic and semialgebraic functions are. The defintion can be found in [6].

Definition 10: real analytic and semialgebraic functions
A smooth function $\phi(t)$ on $\mathbb{R}$ is analytic if:

$$\left(\frac{\phi^k(t)}{k!}\right)^{\frac{1}{k}}$$

is bounded for all k on any compact set $D\subset\mathbb{R}$. For $\phi(x)$ on $\mathbb{R}^n$ check the analyticity of $\phi(x+ty)$ for any $x,y\in\mathbb{R}^n$.

A function ϕ is called semialgebraic if its graph:

$$\{(x,\phi(x)) : x\in dom(\phi)\}$$

is a semialgebraic set. A set $D\in\mathbb{R}^n$ is called semialgebraic if it can be represented as:

$$D = \bigcup_{i=1}^{s}\bigcap_{j=1}^{t}\{x\in\mathbb{R}^n : p_{ij}(x) = 0, q_{ij} > 0\},$$

with p_{ij}, q_{ij} real polynomial functions for $i\le i\le s$, $1\le j\le t$.

Now we are ready to prove that (3.10) is a KL-function.

Theorem 6: Joint model is a KL-function
The function (3.10) *is a KL-function.*

Proof: The proof consist of two parts:

1. Prove that $\frac{\lambda}{2}\|Fu_t - f_t\|_2^2$ and $\frac{\gamma}{2}\|\partial_t u_t + \nabla u_t\cdot\mathbf{v}_t\|_2^2$ are real analytic functions for each $t\in[0,T]$.
2. Prove that $\alpha_1\|\nabla u_t\|_1$, $\alpha_2\|\Phi u_t\|_1$ and $\beta\|\nabla\mathbf{v}\|_1$ are semialgebraic functions for each $t\in[0,T]$.

If 1 and 2 hold then lemma 2.3 gives that (3.10) is a KL-function.

1. Note that both:

$$\frac{\lambda}{2}\|Fu_t - f_t\|_2^2$$
$$\frac{\gamma}{2}\|\partial_t u_t + \nabla u_t\cdot\mathbf{v}_t\|_2^2$$

are polynomial functions and hence by definition real analytic.

2. In [53] the authors show that the l_1-norm is semialgebraic and conclude that any function of the form $\|Ax\|_1$ is semialgebraic. So we find that:

$$\|\nabla u_t\|_1$$
$$\|\Phi u_t\|_1$$
$$\|\nabla \mathbf{v}_t\|_1$$

are semialgebraic.

So (3.10) is a sum of real analytic and semialgebraic functions and is therefore a KL-function. □

Chapter 4

Optimization

In this chapter the optimization of the joint model is explained. This is challenging since we established in the previous chapter that the functional is nonconvex. However the functional (3.1) is biconvex, so the first approach to optimize the functional is to alternate between the convex subproblems. This approach was used before in this context by [18]. In this chapter we will first treat splitting methods for convex optimization to solve the two convex subproblems. Subsequently we treat optimization methods for nonconvex functionals. Here the KL-property discussed in the previous chapter becomes really important. In the last section of the chapter we will compare both methods for a two-dimensional example. We start this chapter with by stating the discretization of the model.

4.1 Discretization

For the implementation of both algorithms the same discretization will be used. In this section the discretization is treated. We start by the discretization of the image sequence $u(\mathbf{x}, t)$, the measurements $f(\mathbf{x}, t)$ and the flow field $\mathbf{v}(\mathbf{x}, t)$.

Discretization variables
The discrete version of $u(\mathbf{x}, t) \in \Omega \times [0, T]$ is defined as:

$$\begin{aligned} u_{ijt} \in \mathbb{R}^{n_x \times n_y \times n_t}, \text{ with } & i \in \{0 \dots, n_x\}, \\ & j \in \{0, ..., n_y\}, \\ & t \in \{0, \dots, n_t\}. \end{aligned}$$

If we refer to u at time t for all i, j, we will write u_t. For $f \in \Sigma \times [0, T]$ the discretization is:

$$\begin{aligned} f_{ijt} \in \mathbb{C}^{n_x \times n_y \times n_t}, \text{ with } & i \in \left\{-\frac{1}{2}n_x, \dots, \frac{1}{2}n_x\right\}, \\ & j \in \left\{-\frac{1}{2}n_y, \dots, \frac{1}{2}n_y\right\}, \\ & t \in \{0, \dots, n_t\}. \end{aligned}$$

For $\mathbf{v}(\mathbf{x}, t) \in \Omega \times [0, T]$ the discretization is:

$$\begin{aligned} \mathbf{v}_{ijt} \in \mathbb{R}^{n_x \times n_y \times n_t - 1 \times 2}, \text{ with } & i \in \{0, \dots, n_x\}, \\ & j \in \{0, \dots, n_y\}, \\ & t \in \{0, \dots, n_t - 1\}. \end{aligned}$$

additionally we will use the notation $\mathbf{v}_{i,j,t} = \begin{bmatrix} v_{x_{i,j,t}} & v_{y_{i,j,t}} \end{bmatrix}$ to indicate the direction of the flow.

Remark: We used the same discretization of time for all three variables. This means that $\mathbf{v}_t$ at time t describes the deformation from u_t to u_{t+1}. This means also that f_t contains all the measurements corresponding to the image u_t. So we assumed that the time needed for each measurement is negligible small compared to the time between frames in the sequence.

Remark: We can reformulate $u \in \mathbb{R}^{n_x \times n_y \times n_t}$ to $u \in \mathbb{R}^N$, with $N = n_x n_y n_t$ by vectorization, both notations will be used. The same holds for $\mathbf{v} \in \mathbb{R}^M$ with $M = n_x n_y (n_t - 1) \times 2$.

Discretization of operators
Next we have to discretize the operators F, ∇, Φ and ∂_t and their adjoints. For the partial derivates, gradient and divergence we follow [18], and use forward differences for the spatial and temporal derivatives. However, for $\partial_t u + \nabla u \cdot \mathbf{v}$, for the discretization of ∇ central differences are used. This will result in a stable discretization in both variables u and $\mathbf{v}$. The definition of the forward, central and backward differences used can be found in appendix B.

For the operator $F = P(\mathcal{F}(\mathrm{Re}^*(\cdot)))$, we discretize each component as follows. For the operator Re we can write now:

$$\begin{aligned} \mathrm{Re} &: \mathbb{C}^N \to \mathbb{R}^N, \quad \mathrm{Re}(x + iy) = x, \\ \mathrm{Re}^* &: \mathbb{R}^N \to \mathbb{C}^N, \quad \mathrm{Re}^*(x) = x + i0. \end{aligned}$$

For the Fourier transform we use its discrete counterpart applied for each $t \in [0, T]$. We use the implementation of the discrete Fourier transform in the AVIONIC toolbox [48]. Finally the subsampling operator can be discretized as follows:

$$P(x) := \begin{cases} 1 & \text{if } x \in \pi(\Sigma) \\ 0 & \text{else} \end{cases},$$

with $\pi(\Sigma)$ the discretized set of spatial coordinates in k-space that are sampled.

For the wavelet transform Φ we use the discrete Daubechies wavelet with a filter of order four and a scalingsfactor of four. For the implementation we used Wavelab [19].

Now we can define the discretized version of (3.1):

$$\min_{u \in \mathbb{R}^N, \mathbf{v} \in \mathbb{R}^M} J(u, \mathbf{v}) = \sum_{t=0}^{T} \frac{\lambda}{2} \|Fu_t - f_t\|_2^2 + \alpha_1 \|\nabla u_t\| + \alpha_2 \|\Phi u_t\|_1 + \sum_{t=0}^{T-1} \beta \|\nabla \mathbf{v}_t\|_1 + \frac{\gamma}{2} \|\partial_t u_t + \nabla u_t \cdot \mathbf{v}_t\|_2^2 + \chi_+(u_t). \quad (4.1)$$

were $\chi_+(u_t)$ is defined as:

$$\chi_+(u_{i,j,t}) := \begin{cases} 0 & \text{if } u_{i,j,t} \geq 0 \quad \forall i, j, t \\ \infty & \text{else,} \end{cases}$$

to ensure that u maps to positive values.

4.2 Splitting methods for convex optimization

Consider the following general convex minimization problem:

$$\min_{x \in X} T(x),$$

where $T(x)$ is a proper convex functional. The classical way to solve this problem is via the proximal point algorithm:

$$x^{k+1} = (I + \tau \partial T)^{-1}(x^k). \quad (4.2)$$

However, one step of the proximal point algorithm can be just as difficult as the original problem, since the proximal map $(I + \tau \partial T)^{-1}$ is not easy to determine for a general functional or map T. The proximal map is

defined as follows:

$$\begin{aligned} x &= (I + \tau \partial T)^{-1}(y), \\ &= \text{prox}_{\tau}^{T}(y), \\ &= \underset{x}{\text{argmin}} \left\{ \frac{\|x - y\|^2}{2} + \tau T(x) \right\}. \end{aligned} \tag{4.3}$$

Which could be interpreted as an approximation of a gradient descent step, where we seek a point x in the domain of the functional $T(x)$ which is close to the previous point y with step-size τ.

For some functionals the proximal map is easy and quick to determine. The key idea is to split the problem into functionals with easy to determine proximal maps and then apply those proximal maps subsequently to perform a full proximal step as defined in (4.2). One of the most well-known splitting methods is the Douglas-Rachford splitting [21], defined as:

$$x^{k+1} = (I + \tau B)^{-1}((I + \tau A)^{-1}(I - \tau B) + \tau B)(x^k),$$

for $T = A + B$. Originally this method was defined for linear operators. However [31] extended the splitting method for general operators and as it turns out Douglas-Rachford splitting is a special case of the proximal point algorithm [22]. So iterating via Douglas-Rachford splitting will give the same solution as iterating 4.2.

Now consider the following general convex minimization problem which can be written as:

$$\min_{x \in X} \quad F(Kx) + G(x) \tag{4.4}$$

where K is a general bounded linear operator. This problem is called the primal problem. Using the definition of the convex conjugate of proper functional $J : X \to \mathbb{R}$:

$$J^*(p) := \sup_{x \in X} \left\{ \langle p, x \rangle_X - J(x) \right\} \quad \text{for } p \in X^*$$

and $J^{**} = J$, we can write:

$$\begin{aligned} \min_{x \in X} F(Kx) + G(x) &= \min_{x \in X} \sup_{p \in X^*} \langle p, Kx \rangle_{X^*} - F^*(p) + G(x) \\ &= \min_{x \in X} \sup_{p \in X^*} \langle p, Kx \rangle_{X^*} - F^*(x) + G(x) \end{aligned} \tag{4.5}$$

which is the saddle-point formulation for primal problem (4.4), with X^* the dual space of the Banach space X. Using the definition of the Gâteaux derivative, we can write down the optimality conditions:

$$\begin{aligned} -K^T p &\in \partial G(x) \\ Kx &\in \partial J^*(p). \end{aligned} \tag{4.6}$$

Another way of splitting the problem (4.4) is to decouple the problem using the substitution $Kx = w$:

$$\begin{aligned} \min_{x,w} \quad & F(w) + G(x) \\ \text{s.t.} \quad & Kx = w, \end{aligned} \tag{4.7}$$

then we have the following Lagrangian:

$$L(x, w, p) = F(w) + G(x) + \left\langle p, Kx - w \right\rangle$$

with corresponding optimality conditions:

$$\begin{aligned} -K^* p &\in \partial G(x) \\ p &\in \partial J(w) \\ Kx &= w, \end{aligned}$$

which is clearly equivalent to (4.6). So the optimal solution to (4.4) is the same as for the decoupled problem (4.7). Solving (4.7) can be done by the Alternating Direction Method of Multipliers (ADMM) [10] , which turns out to be equivalent to applying the Douglas-Rachford splitting on the dual problem.

If we now apply an appropriate preconditioner using ADMM we arrive at the Primal-Dual Hybrid Gradient method of [15]. This algorithm is widely used and easy to implement. We will use this algorithm to solve the convex subproblems for the minimization the biconvex functional (3.1).

So if we can write the general convex minimization problem (4.4) as a saddle-point problem (4.5), then we can solve this via the following algorithm developed by Chambolle and Pock [15].

Algorithm 1: Primal-Dual Hybrid Gradient (PDHG)

- Initialize: Choose $\tau, \sigma > 0, \theta \in [0,1]$. Set $(x^0, p^0) \in X \times X^*$ and $\bar{x} = x^0$.
- iterate for $k \geq 0$ x^k, p^k and $\bar{x}^k$ as follows:

$$\begin{cases} p^{k+1} = (I + \sigma \partial F^*)^{-1}\left(p^k + \sigma K \bar{x}^k\right) \\ u^{k+1} = (I + \tau \partial G)^{-1}\left(x^k - \tau K^* p^{k+1}\right) \\ \bar{x}^{k+1} = x^{k+1} + \theta\left(x^{k+1} - x^k\right). \end{cases} \tag{4.8}$$

Following [15], we will use $\theta = 1$ in the remainder.

To conclude this section on splitting methods for convex optimization we want to point out another line of splitting methods. Namely the Proximal Forward-Backward(PFB) splitting, also known as Proximal Gradient Descent. This method can be applied to problems consisting of a smooth and non-smooth functional. We take then a forward or explicit step over the smooth part and a proximal or backward step on the non-smooth part. Hence we can write the update as follows:

$$x^{k+} = (I + \tau \partial g)^{-1}(x^k - \tau \nabla f x^k) \tag{4.9}$$

were $T = g + f$, were g is nonsmooth and f smooth. To accelerate this method it is possible to add a inertial term as presented in [33]. We will see later on that this type of splitting method generalizes well in the case of a problem that can be split into a nonsmooth convex functional and a smooth nonconvex functional. For a broader overview on splitting methods for convex optimization we refer to [24].

4.3 Optimization method 1: Alternating PDHG

As established before the functional (3.1) is biconvex. So the first optimization approach as given in [18], is to alternate between the two convex subproblems. So the algorithm will consist of one outer loop, where we alternate between two inner loops. In the inner loops the subproblems are solved.

This gives optimization method 1 below. In the following sections we will derive the algorithms for the two inner loops based on PDHG, see algorithm 1.

Optimization method 1: Alternating PDHG

While error$(u^n, \mathbf{v}^n) >$ tol, iterate for $n \geq 0$:

- Find u^{n+1} by solving the **subproblem for reconstruction**:

$$\begin{aligned} u^{n+1} =& \underset{u}{\operatorname{argmin}} \quad J(u, \mathbf{v}^n), \\ =& \underset{u}{\operatorname{argmin}} \quad \int_0^T \frac{\lambda}{2} \|F(\cdot,t)u(\cdot,t) - f(\cdot,t)\|_2^2 + \alpha_1 \|\nabla_x u(\cdot,t)\|_1 + \alpha_2 \|\Phi(u(\cdot,t))\|_1 \\ &+ \frac{\gamma}{p} \|\partial_t u(\cdot,t) + \nabla u(\cdot,t) \cdot \mathbf{v}^n(\cdot,t)\|_p^p + \chi_+(u(\cdot,t)) \mathrm{d}t. \end{aligned} \tag{4.10}$$

- Find $\mathbf{v}^{n+1}$ by solving the **subproblem for motion estimation**:

$$\begin{aligned} \mathbf{v}^{n+1} =& \underset{\mathbf{v}}{\operatorname{argmin}} \quad J(u^{n+1}, \mathbf{v}), \\ =& \underset{\mathbf{v}}{\operatorname{argmin}} \quad \int_0^T \beta \|\nabla \mathbf{v}(\cdot,t)\|_1 + \frac{\gamma}{p} \|\partial_t u^{n+1}(\cdot,t) + \nabla u^{n+1}(\cdot,t) \cdot \mathbf{v}(\cdot,t)\|_p^p \mathrm{d}t. \end{aligned} \tag{4.11}$$

- error$(u, \mathbf{v}) = \frac{|u^{n+1} - u^n| + |\mathbf{v}^{n+1} - \mathbf{v}^n|}{2|\Omega \times T|}$.

with $p \in \{1, 2\}$. Next we will describe how to solve each subproblem.

Remark: Here we first optimize and then discretize.

4.3.1 Optimization of subproblem for reconstruction

We can rewrite the problem in u (4.10) as a saddle-point problem:

$$\min_u \max_{\mathbf{p}=[p_1,p_2,p_3,p_4]} \int_0^T \chi_+(u) + \langle Ku, \mathbf{p} \rangle - \frac{1}{2\lambda} \|p_1\|_2^2 - \langle p_1, f \rangle - \alpha_1 \delta_{B(L^\infty)}\left(\frac{p_2}{\alpha_1}\right) - \alpha_2 \delta_{B(L^\infty)}\left(\frac{p_3}{\alpha_2}\right) - F^*(p_4) \mathrm{d}t,$$

were we define $F^*(p_4)$ as follows:

$$F^*(p_4) = \begin{cases} \gamma \delta_{B(L^\infty)}\left(\frac{p_4}{\gamma}\right) & \text{if } p = 1, \\ \frac{1}{2\gamma} \|p_4\|_2^2 & \text{if } p = 2. \end{cases}$$

Recall that we can write:

$$A_{\mathbf{v}} := [\partial_t + v_x(\cdot,t)\partial_x + v_y(\cdot,t)\partial_y]\,u(\cdot,t), \tag{4.12}$$

with given $\mathbf{v}(\cdot,t) = \begin{bmatrix} v_x(\cdot,t) & v_y(\cdot,t) \end{bmatrix}$. Now the operator K and its adjoint K^* for the saddle-point problem can be defined as:

$$Ku := \begin{pmatrix} F \\ \nabla \\ \Phi \\ A_{\mathbf{v}} \end{pmatrix} u,$$

$$K^*(p_1,p_2,p_3,p_4) := \begin{pmatrix} F^* & -\mathrm{div} & \Phi^* & A_{\mathbf{v}}^* \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix}.$$

Note that we cannot reconstruct frame-by-frame. For the reconstruction of u^k we need frame u^{k+1}, since we need to determine $\partial_t u^k$ in operator $A_{\mathbf{v}}$. So the problem in u must be solved for all frames simultaneously.

For a derivation of the convex conjugates, adjoint operators and proximal maps we refer to appendix B. Using (4.8) we can write:

Algorithm 2: Subproblem for reconstruction of u

- Initialize: Choose $\tau, \sigma > 0$ such that $\tau\sigma\|K\| \leq 1$. Set $(u^0, \mathbf{p}^0) \in X \times X^*$ and $\bar{u} = u^0$.
- iterate for $k \geq 0$ u^k, $\mathbf{p}^k$ and $\bar{u}^k$ as follows:

$$\begin{cases} \hat{p}_1^{k+1} = p_1^k + \sigma F\overline{u}^k \\ p_1^{k+1} = \dfrac{\hat{p}_1^{k+1} - \sigma f}{\lambda + \sigma} \\ \hat{p}_2^{k+1} = p_2^k + \sigma\nabla\overline{u}^k \\ p_2^{k+1} = \min\left(\alpha_1, \max\left(-\alpha_1, \hat{p}_2^{k+1}\right)\right) \\ \hat{p}_3^{k+1} = p_3^k + \sigma\Phi\overline{u}^k \\ p_3^{k+1} = \min\left(\alpha_2, \max\left(-\alpha_2, \hat{q}_2^{k+1}\right)\right) \\ \hat{p}_4^{k+1} = p_4^k + \sigma A_{\mathbf{v}}\overline{u}^k \\ p_4^{k+1} = \begin{cases} \min\left(\gamma, \max\left(-\gamma, \hat{p}_4^{k+1}\right)\right) & \text{if } p = 1 \\ \frac{\gamma\hat{p}_4}{\sigma+\gamma} & \text{if } p = 2 \end{cases} \\ \hat{u}^{k+1} = u^k - \tau\left(F^* p_1^{k+1} - \mathrm{div}^- \cdot p_2^{k+1} + \phi^* p_3^{k+1} + A_{\mathbf{v}}^* p_4^{k+1}\right) \\ u^{k+1} = \max\left(0, \hat{u}^{k+1}\right) \\ \overline{u}^{k+1} = 2u^{k+1} - u^k \end{cases} \tag{4.13}$$

Remark: If we set $\gamma = 0$ we find an algorithm for frame-by-frame MRI reconstruction, this algorithm will be used for comparison between frame-by-frame reconstruction and joint reconstruction.

Stopping-criterion and parameter choice

To complete the algorithm we have to define a stopping-criterion and choose the constants σ and τ. As the stopping criterion for the Primal-Dual Hybrid Gradient Algorithm we use the primal-dual gap:

$$\varepsilon^k := \frac{p^k + d^k}{n_x \cdot n_y \cdot n_t} < \mathrm{tol},$$

normalized with respect to the size of the problem.

The algorithm converges if $p^k \to 0$ and $d^k \to 0$, if $k \to \infty$, for more details see [26]. Define:

$$p^k = \left| \frac{x^k - x^{k+1}}{\tau^k} - K^* \left(y^k - y^{k+1} \right) \right|,$$
$$d^k = \left| \frac{y^k - y^{k+1}}{\sigma^k} - K \left(x^k - x^{k+1} \right) \right|,$$

with x^k, x^{k+1}, y^k and y^{k+1} the primal and respectively dual variables in iterations k and $k+1$.

Next we must choose σ, τ such that:

$$\sigma \tau \|K\|^2 \leq 1. \tag{4.14}$$

Hence we must determine $\|K\|$, note:

$$\begin{pmatrix} \|F(\cdot)\| \\ \|\nabla(\cdot)\| \\ \|\Phi(\cdot)\| \\ \|A_{\mathbf{v}}\| \end{pmatrix} \leq \begin{pmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \end{pmatrix} \leq \sup_i M_i$$

Hence we must find the operator norms or at least a bound on $\|F\|$, $\|\nabla\|$, $\|\Phi\|$ and $\|A_{\mathbf{v}}\|$. For $\|\nabla\|$ we refer to [15], were the bound for $\|\nabla\|$ is given by $\sqrt{8}$.

For $\|F\|$, recall $\|Fu\| = \|P \circ \mathscr{F} \circ \mathrm{Re}(u)\|$. Hence:

$$\|Fu\| \leq \|P\| \|\mathscr{F}\| \|\mathrm{Re}^*\| \|u\|$$

the Fourier transform is a unitary operator, hence $\|\mathscr{F}\| = 1$. For $\|P\|$:

$$\|Pu\| \leq \|u\|$$

since $P(\cdot)$ subsamples u. So we find the bound $\|P\| \leq 1$. Also for Re we find $\|\mathrm{Re}\| = 1$, since for $u \in \mathbb{R}^N$ the operator is equivalent to the identity operator. So we find the bound $\|F\| \leq 1$.

The wavelet transform based on the Daubechies wavelet is an orthogonal operator and therefore we find $\|\Phi\| = 1$, see [17].

In [18], the following bound for $\|A_{\mathbf{v}}\|$ was found:

$$\|A_{\mathbf{v}}\| \leq \max|\mathbf{v}| \cdot \sqrt{12} \tag{4.15}$$

So we choose:

$$\tau = \sigma = \sup_i \frac{1}{M_i} = \frac{1}{\max(|\mathrm{v}|, 1) \cdot \sqrt{12}}$$

Note that $\max|\mathbf{v}|$ is bounded since we assumed $\|\mathbf{v}\|_{L^\infty} \leq c < \infty$.

4.3.2 Optimization of subproblem for motion estimation

We can rewrite the problem in $\mathbf{v}$ (4.11) to the following saddle point problem:

$$\min_{\mathbf{v}} \max_{q_1, \mathbf{q}_2} \int_0^T -F^*(q_1) + \langle K\mathbf{v}, \mathbf{q} \rangle - \beta \delta_{B(L^\infty)} \mathrm{d}t$$

Define:

$$Ku := \begin{pmatrix} \partial_x u & \partial_y u \\ \nabla_x & 0 \\ 0 & \nabla_x \end{pmatrix} \begin{pmatrix} v_x \\ v_y \end{pmatrix}$$

$$K^*(q_1, q_2) := \begin{pmatrix} (\partial_x u)^* & -\mathrm{div} & 0 \\ (\partial_y u)^* & 0 & -\mathrm{div} \end{pmatrix} \begin{pmatrix} q_1 \\ q_{2x} \\ q_{2y} \end{pmatrix}$$

Recall that we can write for given u:

$$A_u := \begin{bmatrix} \partial_x u & \partial_y u \end{bmatrix} \tag{4.16}$$

which is a linear operator. Next define:

$$F^*(q_1) := \begin{cases} \delta_{B(L^\infty)}\left(\frac{q_1}{\gamma}\right) & \text{if } p = 1 \\ \frac{1}{2\gamma}\|q_1\|_2^2 - \langle q_1, \partial_t u \rangle & \text{if } p = 2 \end{cases}.$$

Now using (4.8) we can write:

Algorithm 3: Subproblem for motion estimation v

- Initialize: Choose $\tau, \sigma > 0$ such that $\tau\sigma\|K\| \leq 1$. Set $(\mathbf{v}^0, \mathbf{q}^0) \in X \times X^*$ and $\bar{\mathbf{v}} = \mathbf{v}^0$.
- iterate for $k \geq 0$ $\mathbf{v}^k$, $\mathbf{q}^k$ and $\bar{\mathbf{v}}^k$ as follows:

$$\begin{cases} \hat{q_1}^{k+1} = q_1^k + \sigma\begin{pmatrix} \partial_x u^{k+1} & \partial_y u^{k+1} \end{pmatrix}\bar{\mathbf{v}} \\ q_1^{k+1} = \begin{cases} \min\left(\gamma, \max\left(-\gamma, \hat{(}q_1) + \sigma\partial_t\right)\right) & \text{if } p = 1 \\ \frac{\gamma(\hat{p_1}^{k+1} + \sigma\partial_t u^{k+1})}{\gamma+\sigma} & \text{if } p = 2 \end{cases} \\ \hat{\mathbf{q}}_2^{k+1} = \mathbf{q}_2^k + \sigma\begin{pmatrix} \nabla & 0 \\ 0 & \nabla \end{pmatrix}\bar{\mathbf{v}}^k \\ \mathbf{q}_2^{k+1} = \min\left(\beta, \max\left(-\beta, \hat{\mathbf{q}}_2\right)\right) \\ \mathbf{v} = \mathbf{v}^k - \tau\left(\left[\left(\partial_x u^{k+1}\right)^* + \left(\partial_y u^{k+1}\right)^*\right] q_1^{k+1} - \begin{pmatrix} \text{div} & 0 \\ 0 & \text{div} \end{pmatrix}\mathbf{q}_2\right) \\ \bar{\mathbf{v}}^{k+1} = 2\mathbf{v}^{k+1} - \mathbf{v}^k \end{cases} \tag{4.17}$$

Stopping criterion and parameter choice
Again we use the primal-dual gap as stopping criterion and we pick for the parameters:

$$\tau = \sigma = \frac{1}{\sqrt{8}}.$$

This bound can be derived as follows. We can write:

$$A_u = \nabla \circ u,$$

so we know:

$$\|A_u\| = \|\nabla\| \cdot \|u\|,$$

we have the following bounds:

$$\|\nabla\| \leq \sqrt{8},$$
$$\|u\| \leq \max|u|$$

Note that $\max|u| \leq 1$ since we scaled all images. So we obtain:

$$\|A_u^k\| \leq \max|u| \cdot \sqrt{8}. \tag{4.18}$$

So for the bound on τ and σ we pick:

$$\tau = \sigma = \sup_i \frac{1}{M_i} = \frac{1}{\sup(\max|u|\sqrt{8}, \sqrt{8})} = \frac{1}{\sqrt{8}}$$

4.4 Nonconvex optimization methods

As shown in [18] and [25] the presented Alternating PDHG algorithm works well in practice. However there is no knowledge about the convergence of this algorithm. This outer, alternating scheme is very similar to the classical Gauss-Seidel method. A key assumption to prove convergence for the Gauss-Seidel method in the case of a convex function, is that the minimum in each step is uniquely attained, otherwise as [44] shows, the method can circle indefinitely without converging. Proving convergence becomes harder in the case of nonconvex functionals.

Besides the fact that it is challenging to state any convergence result for this scheme, this scheme also suffers from practical drawbacks. Namely:

1. How many inner iterations for each subproblem should we take, before going to the next subproblem?
2. How to deal with computational errors that are propagated in each step?

To address these questions and to gain more knowledge about the convergence of the method used, we will explore the field of nonconvex optimization a bit more.

In this section we will first give an overview of well-known nonconvex optimization methods and choose one to obtain a second optimization method, which does not have the drawbacks mentioned above.

Proving convergence for nonconvex optimization methods
As pointed out in chapter 3 the main ingredient to obtain a convergence result is the Kurdyka-Lojasiewicz property, which was defined for nonsmooth functions by [7]. In chapter 3 we also established that our functional $J(u,\mathbf{v})$ satisfies this property, hence we can use algorithms which base their convergence result on this property.

The convergence result was first given in the pioneering work of Attouch *et al.* [2]. The key role of the KL-property is that it allows to prove that the sequence generated by the algorithm is a Cauchy sequence. In [8] an informal recipe for such a prove is given. Here we will only shortly state the key elements.

Take the following minimization problem:

$$\min_z \psi(z)$$

and assume that $\psi(z)$ is a KL-function. Take an algorithm $\mathscr{A}$ which generates a sequence $\{z^k\}_{k\in\mathbb{N}}$. Assume that we can deduce the following bounds:

1. *Sufficient decrease property*: find a positive constant C_1 suc that:

$$C_1\|z^{k+1}-z^k\|^2 \leq \psi(z^k)-\psi(z^{k+1}), \quad \forall k=0,1,\cdots$$

2. *Subgradient lower bound*: find a positive constant C_2 such that:

$$\|w^{k+1}\| \leq C_2\|z^{k+1}-z^k\|, \quad w^k \in \partial\psi(z^k), \quad \forall k=0,1,\ldots$$

Now we call the solution to the minimization problem $\bar{z}$. Then the KL-condition tells us:

$$\phi'(|\psi(z^k)-\psi(\bar{z})|)\text{dist}(0,\partial\psi(z^k)) \leq 1.$$

Then this inequality connects the obtained bounds. We can bound $\psi(z^k)-\psi(\bar{z})$ using the sufficient decrease property and bound $\text{dist}(0,\partial\psi(z^k)$ using the subgradient lower bound. Together this allows for proving that the generated sequence is a Cauchy sequence and convergences, as presented in [8].

Proximal Alternating Lineared Minimization (PALM)
In [8] the convergence result for functions of the form

$$\min_{x\in\mathbb{R}^n, y\in\mathbb{R}^m} F(x,y) = f(x)+g(x)+H(x,y) \tag{4.19}$$

is given. Here f and g are proper lower semicontinuous functions and H satisfies certain smoothness criteria. Note that no assumptions on the convexity are made. The presented algorithm, Proximal Alternating Linearized Minimization (PALM), can be viewed as an alternating proximal forward-backward splitting algorithm.

Since the function $H(x,y)$ is smooth, the forward (or explicit) step will be taken with respect to this function. So we can view this scheme as the proximal regularization of $H(x,y)$, linearized at the given point. So we can write:

$$\begin{aligned} x^{k+1} &\in \arg\min_{x\in\mathbb{R}^n} \left\{ \left\langle x - x^k, \nabla H_x(x^k, y^k) \right\rangle + \|x - x^k\|^2 + \tau f(x) \right\}, \\ y^{k+1} &\in \arg\min_{y\in\mathbb{R}^m} \left\{ \left\langle y - y^k, \nabla_y H(x^{k+1}, y^k) \right\rangle + \|y - y^k\|^2 + \sigma g(x) \right\}, \end{aligned}$$

using the definition of the proximal map (4.3) we can write:

$$\begin{aligned} x^{k+1} &\in \operatorname{prox}^f_{\tau_1}\left(x^k - \tau \nabla_x H(x^k, y^k) \right), \\ y^{k+1} &\in \operatorname{prox}^g_{\tau_2}\left(y^k - \sigma \nabla_y H(x^{k+1}, y^k) \right). \end{aligned} \tag{4.20}$$

Hence, by employing the smoothness in the coupling term and using the proximal forward-backward splitting scheme, explicit updates for x and y are found. This scheme does not have the practical drawbacks mentioned in the beginning of this section. For a more in-depth analysis of the differences between more implicit algorithms such as Alternating PDHG and more explicit algorithm, such as PALM, we refer to [53]

In the next section we will prove that our functional (3.1) is of the form (4.19) and that we can apply the PALM algorithm. In [54] a more general algorithm with block-wise coordinate updates is discussed, as it turns out PALM is a special case of the algorithm presented there.

Algorithms related to PALM
So far only algorithms which update the coordinates in a alternating/block-wise fashion are considered. Another possibility is to update the whole functional in one step. In [39] the algorithm iPiano is discussed. This algorithm is also based on forward-backward splitting and here an inertial term is added. The update for iPiano is defined as:

$$\mathbf{z}^{k+1} = \operatorname{Prox}^{f+g}_{\alpha}\left(\mathbf{z}^k - \alpha \nabla H(\mathbf{z}^k) + \beta(\mathbf{z}^k - \mathbf{z}^{k-1}) \right),$$

where $z = [x \quad y]$. We call $\beta(z^k - z^{k-1})$ the inertial term.

Note that for $f + g = 0$, $\mathbf{z}^k - \alpha\nabla H(\mathbf{z}^k) + \beta(\mathbf{z}^k - \mathbf{z}^{k-1})$ can be seen as an explicit finite difference discretization of the 'Heavy-ball with friction' dynamical system:

$$\ddot{\mathbf{z}}(t) + \gamma \dot{\mathbf{z}}(t) + \nabla H(\mathbf{z}(t)) = 0,$$

where $\ddot{\mathbf{z}}(t)$ gives an acceleration to the system, so adding an inertial term gives the possibility of leaving a local minimum. However there is no guarantee that this always will happen. The iPiano scheme has stricter assumptions for (4.19), namely that f and g must be convex and $\nabla H(x,y)$ must have globally Lipschitz continuous gradients. PALM does not require convexity assumptions and only ask for globally Lipschitz continuous gradients for fixed values of y and x respectively.

Inspired by the iPiano and PALM, iPALM [43] was developed. iPALM can be viewed as a block version of iPiano or an inertial based version of PALM. Then the updates in (4.20) become:

Algorithm 4: iPALM

$$
\begin{aligned}
u_1^k &= x^k + \alpha_1^k(x^k - x^{k-1}) \\
v_1^k &= x^k + \beta_1^k(x^k - x^{k-1}) \\
x^{k+1} &\in \operatorname{prox}_{\tau_1}^f\left(u_1^k - \tau_1 \nabla H_x(v_1^k, y^k)\right)
\end{aligned}
$$

$$
\begin{aligned}
u_2^k &= y^k + \alpha_2^k(y^k - y^{k-1}) \\
v_2^k &= y^k + \beta_2^k(y^k - y^{k-1}) \\
y^{k+1} &\in \operatorname{prox}_{\tau_2}^g\left(u_2^k - \tau_2 \nabla H_y(x^{k+1}, v_2^k)\right).
\end{aligned}
$$

So for $\alpha_i^k = \beta_i^k = 0$ for $i = 1,2$ and for $k \geq 0$ the algorithm is equivalent to PALM and for $\beta_1 = 0$ and $y^k = 0$ for all $k \geq 0$ to iPiano.

4.5 Optimization method 2: PALM

In the previous section we disccused the algorithm PALM, which minimizes problems of the following form:

$$F(x,y) := f(x) + g(y) + H(x,y), \tag{4.21}$$

with $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$. Now we define for our model with $u \in \mathbb{R}^N$ and $\mathbf{v} \in \mathbb{R}^M$:

$$
\begin{aligned}
f(u) &= \sum_{t=0}^{T} \frac{\lambda}{2} \|Fu_t - \tilde{f}_t\|_2^2 + \alpha_1 \|\nabla u_t\|_1 + \alpha_2 \|\Phi u_t\|_1 + \chi_+(u_t) \\
g(\mathbf{v}) &= \sum_{t=0}^{T-1} \beta \|\nabla \mathbf{v}_t\|_1 \\
H(u,\mathbf{v}) &= \sum_{t=0}^{T-1} \frac{\gamma}{p} \|\partial_t u_t + \nabla u_t \cdot \mathbf{v}_t\|_p^p \quad p \in \{1,2\}
\end{aligned} \tag{4.22}
$$

so we can use PALM to optimize our joint model. First we will discuss which properties (4.1) must fulfill for convergence of PALM and then we will formulate an algorithm based on PALM.

Remark: Here we first discretize and then optimize.

Remark: To distinguish the defined function $f(u)$ from the measurements, we will use $\tilde{f}$ for the measurements in this section.

4.5.1 Convergence of PALM

To prove convergence of the PALM algorithm the following properties must be satisfied for (4.21):

1. $f(x) : \mathbb{R}^n \to (-\infty, \infty]$ and $g(y) : \mathbb{R}^m \to (\infty, \infty]$ are proper and lower semicontinuous functions such that $\inf_{\mathbb{R}^n} f(x) > -\infty$ and $\inf_{\mathbb{R}^m} g(y) > -\infty$.
2. $F(x,y)$ is a KL-function.
3. $H(x,y) : \mathbb{R}^n \times \mathbb{R}^m$ is differentiable and $\inf_{\mathbb{R}^n \times \mathbb{R}^m} F(x,y) > -\infty$.
4. For any fixed y the function $x \to H(x,y)$ is $C_{L_1(y)}^{1,1}$, namely the partial gradient $\nabla_x H(x,y)$ is globally Lipschitz with modulus $L_1(y)$. The same must hold for $\nabla_y H(x,y)$.
5. For $i = 1,2$ there exists λ_i^-, $\lambda_i^+ > 0$ such that:

$$
\begin{aligned}
\inf\{L_1(y^k) : k \geq 0\} \geq \lambda_1^- \quad \inf\{L_2(x^k) : k \geq 0\} \geq \lambda_2^- \\
\sup\{L_1(y^k) : k \geq 0\} \leq \lambda_1^+ \quad \sup\{L_2(x^k) : k \geq 0\} \leq \lambda_2^+
\end{aligned}
$$

6. ∇H is Lipschitz continuous on bounded subsets $B_1 \times B_2$ of $\mathbb{R}^n \times \mathbb{R}^m$. So there exists an $M > 0$ such that for any $(x_i, y_i) \in B_1 \times B_2$ for $i = 1, 2$:

$$\left\| \left(\nabla_x H(x_1, y_1) - \nabla_x H(x_2, y_2), \nabla_y H(x_1, y_1) - \nabla_y H(x_2, y_2) \right) \right\| \leq M \left\| (x_1 - x_2, y_1 - y_2) \right\|$$

Then the sequence $(x^k, y^k) \in \mathbb{R}^n \times \mathbb{R}^m$ is generated as follows:

Algorithm 5: PALM

- Initialize: start with x^0 and y^0 in $\mathbb{R}^n \times \mathbb{R}^m$.
- For each $k = 1, 2, \ldots$ generate sequence $\{(x^k, y^k)\}_{k \in \mathbb{N}}$ as follows:
  1. Take $\gamma_1 > 1$ and $\tau_1^k = \frac{\gamma_1}{L_1(y^k)}$. Compute:

$$\begin{aligned} \overline{x}^{k+1} &= y^k - \tau_1^k \nabla_x H(x^k, y^k) \\ x^{k+1} &\in (I + \tau_1^k \partial f)^{-1}(\overline{x}^{k+1}) \end{aligned}$$

  2. Take $\gamma_2 > 1$ and $\tau_2^k = \frac{\gamma_2}{L_2(x^{k+1})}$. Compute:

$$\begin{aligned} \overline{y}^{k+1} &= y^k - \tau_2^k \nabla_y H(x^{k+1}, y^k) \\ y^{k+1} &\in (I + \tau_2^k \partial g)^{-1}(\overline{y}^{k+1}) \end{aligned}$$

Now we can state the following convergence result from [8].

Theorem 7: Convergence of PALM
The sequence (x^k, y^k) for $k \in \mathbb{N}$ generated by PALM which is assumed to be bounded will convergence to a critical point (x^*, y^*) of $F(x, y)$ if assumptions 1 till 6 are satisfied.

So if we can prove that for (4.1) assumption 1 till 6 hold, then we know that the algorithm will converge to a critical point.

Proof of assumptions

Define $f(u)$, $g(\mathbf{v})$ and $H(u, \mathbf{v})$ as in (4.22) and choose $p = 2$ since we need that $H(u, \mathbf{v})$ is a C^1 function.

Assumption 1 and 2
By the definition of a norm we have that:

- $\inf_{\mathbb{R}^N} f(u) = 0$,
- $\inf_{\mathbb{R}^M} g(\mathbf{v}) = 0$,
- $\inf_{\mathbb{R}^N \times \mathbb{R}^M} H(u, \mathbf{v}) = 0$,

so we also have:

$$\inf_{\mathbb{R}^N \times \mathbb{R}^M} F(u, \mathbf{v}) = 0.$$

The properness and lower semi-continuity of $f(u)$ and $g(\mathbf{v})$ is established in chapter 3 so we have proved assumption 1. In chapter 3 we also proved that $F(u, \mathbf{v})$ is a KL function, so we also proved assumption 2.

Assumption 3 and 4
In (4.12) and (4.16) we saw that we can rewrite $H(u, \mathbf{v})$ for fixed values of $\mathbf{v}$ and u respectively. Next calculate the partial derivatives of $H(u, \mathbf{v})$ using the definition of the Gâteaux derivative:

$$\begin{aligned} dH(u, \mathbf{v}^k; \bar{u}) &= \left\langle \gamma A_{\mathbf{v}^k}^* A_{\mathbf{v}^k} u, \bar{u} \right\rangle, \\ dH(u^k, \mathbf{v}; \bar{\mathbf{v}}) &= \left\langle \gamma A_{u^k}^* A_{u^k} \mathbf{v}, \bar{\mathbf{v}} \right\rangle. \end{aligned}$$

As established in (4.12), (4.15) and (4.16), (4.18), $A_{\mathbf{v}}$ and A_u are bounded linear operators, hence their adjoints are as well. Now using the boundedness of products of bounded linear maps we can deduce that the given Gatêaux derivatives can be written as bounded and linear operators. This gives us the Fréchet differentiablity of $H(u,\mathbf{v})$, hence we have proved assumption 3.

Now we can define the gradient of $H(u,v)$:

$$\nabla_u H(u,\mathbf{v}^k) = \gamma A^*_{\mathbf{v}^k} A_{\mathbf{v}^k} u,$$
$$\nabla_{\mathbf{v}} H(u^k,\mathbf{v}) = \gamma A^*_{u^k} A_{u^k} \mathbf{v}.$$

For fixed and arbitrary $\mathbf{v}^k$ calculate:

$$d(\nabla_u H(u,\mathbf{v}^k;\bar{u})) = \left\langle \gamma A^*_{\mathbf{v}^k} A_{\mathbf{v}^k}, \bar{u} \right\rangle$$

which is constant and bounded for arbitrary fixed v^k, hence the Fréchet derivative of $\nabla_u H(u,\mathbf{v}^k)$ exists for fixed v^k and is continuous. So $\nabla_u H(u,\mathbf{v}^k)$ is C^1 and hence L-Lipschitz continuous with constant L_1:

$$L_1(\mathbf{v}^k) = \|\gamma A^*_{\mathbf{v}^k} A_{\mathbf{v}^k}\|_2. \tag{4.23}$$

For fixed and arbitrary u^{k+1} we get:

$$d(\nabla_v H(u^{k+1},\mathbf{v};\bar{\mathbf{v}})) = \left\langle \gamma A^*_{u^k} A_{u^k}, \bar{\mathbf{v}} \right\rangle,$$

so $\frac{d}{d\mathbf{v}} \nabla_v H$ is also a bounded and constant and hence the derivative exists and is continuous. So $\nabla_v H(u^{k+1},\mathbf{v})$ is L-Lipschitz continuous with constant L_2:

$$L_2(u^{k+1}) = \|\gamma A^*_{u^k} A_{u^k}\|_2. \tag{4.24}$$

So we have proved assumption 4.

Assumption 5
Assumption 5 is important, since it guarantees that the step sizes τ_i for $i = 1,2$ are well-defined. As mentioned by [8] we can always write:

$$L_1'(\mathbf{v}^k) := \max\{L_1(\mathbf{v}^k), \mu_1^-\}$$
$$L_2'(u^k) := \max\{L_2(u^k), \mu_2^-\}$$

Note that L_1' and L_2' are still Lipschitz moduli of $\nabla_u H(\cdot,\mathbf{v}^k)$ and $\nabla_{\mathbf{v}} H(u^k,\cdot)$. Hence $\lambda_i^- = \mu_i^-$, with $\mu_i^- > 0$ for $i = 1,2$. Using (4.15) and (4.18), we can write:

$$\begin{aligned} \lambda_1^+ &\le \gamma \max|\mathbf{v}|^2 \cdot 12 \\ \lambda_2^+ &\le \gamma \max|u|^2 \cdot 8 \end{aligned} \tag{4.25}$$

with $\gamma > 0$, which concludes the proof of assumption 5.

Assumption 6
Take any $(u_1,\mathbf{v}_1)$ and $(u_2,\mathbf{v}_2)$ in any bounded subset $B_1 \times B_2 \subset \mathbb{R}^N \times \mathbb{R}^M$. Then we can derive:

$$\begin{aligned} &\left\| \left(\gamma A^*_{\mathbf{v}_1} A_{\mathbf{v}_1} u_1 - \gamma A^*_{\mathbf{v}_2} A_{\mathbf{v}_2} u_2, \gamma A^*_{u_1} A_{u_1} \mathbf{v}_1 - \gamma A^*_{u_2} A_{u_2} \mathbf{v}_2\right)\right\| = \\ &\left\| \left(\gamma \left(A^*_{\mathbf{v}_1} A_{\mathbf{v}_1} u_1 - A^*_{\mathbf{v}_2} A_{\mathbf{v}_2} u_1 + A^*_{\mathbf{v}_2} A_{\mathbf{v}_2} u_1 - A^*_{\mathbf{v}_2} A_{\mathbf{v}_2} u_2\right), \gamma \left(A^*_{u_1} A_{u_1} \mathbf{v}_1 - A^*_{u_2} A_{u_2} \mathbf{v}_1 + A^*_{u_2} A_{u_2} \mathbf{v}_1 - A^*_{u_2} A_{u_2} \mathbf{v}_2\right)\right)\right\| = \\ &\gamma^2 \left\| \left((A^*_{\mathbf{v}_1} A_{\mathbf{v}_1} - A^*_{\mathbf{v}_2} A_{\mathbf{v}_2}) u_1 + A^*_{\mathbf{v}_2} A_{\mathbf{v}_2} (u_1 - u_2), (A^*_{u_1} A_{u_1} - A^*_{u_2} A_{u_2}) \mathbf{v}_1 + A^*_{u_2} A_{u_2} (\mathbf{v}_1 - \mathbf{v}_2)\right)\right\| = \end{aligned}$$

use that for fixed u, $A_{\mathbf{v}}$ is linear in $\mathbf{v}$ and for fixed $\mathbf{v}$, A_u is linear in u,

$$\gamma^2 \left\| \left(A^*_{\mathbf{v}_1 - \mathbf{v}_2} A_{\mathbf{v}_1 - \mathbf{v}_2} u_1 + A^*_{\mathbf{v}_2} A_{\mathbf{v}_2} (u_1 - u_2), A^*_{u_1 - u_2} A_{u_1 - u_2} \mathbf{v}_1 + A^*_{u_2} A_{u_2} (\mathbf{v}_1 - \mathbf{v}_2)\right)\right\| \le$$

use assumption 4,

$$\gamma^2 \max\{L_1(\mathbf{v}_1 - \mathbf{v}_2), L_2(u_1 - u_2)\} \|(u_1,\mathbf{v}_1)\| + \max\{L_1(\mathbf{v}_2), L_2(u_2)\} \|(u_1 - u_2, \mathbf{v}_1 - \mathbf{v}_2)\| \le$$

use $\|(u_1,\mathbf{v}_1)\| \le C$,

$$\gamma^2 \left(\max\{L_1(\mathbf{v}_1 - \mathbf{v}_2), L_2(u_1 - u_2)\} C + \max\left(L_1(\mathbf{v}_2), L_2(u_2)\right)\right) \|(u_1 - u_2, \mathbf{v}_1 - \mathbf{v}_2)\|$$

so $M = \gamma^2 \max\{L_1(\mathbf{v}_1 - \mathbf{v}_2), L_2(u_1 - u_2)\} C + \max\{L_1(\mathbf{v}_2), L_2(u_2)\} > 0$, hence we proved assumption 6.

4.5.2 PALM for joint reconstruction and motion estimation

We established that (4.1) satisfies all assumptions necessary for the appliction of the PALM algorithm, so the next step is to formulate the algorithm for joint reconstruction and motion estimation.

Algorithm 6: PALM for joint reconstruction and motion estimation

- Initialize: start with u^0 and $\mathbf{v}^0$ in $\mathbb{R}^N \times \mathbb{R}^M$.
- For each $k = 1, 2, \ldots$ generate sequence $(u^k, \mathbf{v}^k)$ as follows:
  1. Take $\gamma_1 > 1$ and $\tau_1^k = \frac{\gamma_1}{L_1(\mathbf{v}^k)}$. Compute:
  $$\overline{u}^k = u^k - \tau_1^k \nabla_u H(u^k, \mathbf{v}^k)$$
  $$u^{k+1} = (I + \tau_1^k \partial f)^{-1}(\overline{u}^k)$$
  2. Take $\gamma_2 > 1$ and $\tau_2^k = \frac{\gamma_2}{L_2(u^{k+1})}$. Compute:
  $$\overline{\mathbf{v}}^k = \mathbf{v}^k - \tau_2^k \nabla_{\mathbf{v}} H(u^{k+1}, \mathbf{v}^k)$$
  $$v^{k+1} = (I + \tau_2^k \partial g)^{-1}(\overline{\mathbf{v}}^k)$$

Remark: Since f and g are convex, the proximal map will be single-valued and therefore we can write an equality.

In order to implement the algorithm, we have to make the updates:

$$\begin{aligned} u^{k+1} &= (I + \tau_1^k \partial f)^{-1}(\overline{u}^k), \\ \mathbf{v}^{k+1} &= (I + \tau_2^k \partial g)^{-1}(\overline{\mathbf{v}}^k), \end{aligned} \tag{4.26}$$

explicit. Where (4.26) corresponds to solving the following minimization problms:

$$\begin{aligned} \min_u \quad & f(u) \\ \min_{\mathbf{v}} \quad & g(\mathbf{v}). \end{aligned} \tag{4.27}$$

Calculating the proximal map of the whole function f and g is difficult, but since these functions are convex we can apply a splitting method to simplify this step. Recall from section 4.1 that applying Douglas-Rachford splitting is equivalent to iteration using the full proximal step. Employing the equivalence between Douglas-Rachford splitting and ADMM and the equivalence between preconditioned ADMM and PDHG, we use the steps of PDHG to rewrite (4.26).

As already mentioned in section 4.3, we could rewrite the minimization problems 4.27 to the saddle-point problems:

$$\begin{aligned} &\min_u \quad \max_{\mathbf{p}} \quad \chi_+(u) + \langle Ku, \mathbf{p} \rangle - \frac{1}{2\lambda} \|p_1\|_2^2 - \langle p_1, \tilde{f} \rangle - \alpha_1 \delta_{B(L^\infty)}\left(\frac{p_2}{\alpha_1}\right) - \alpha_2 \delta_{B(L^\infty)}\left(\frac{p_3}{\alpha_2}\right), \\ &\min_{\mathbf{v}} \quad \max_{\mathbf{q}} \quad \langle \nabla \mathbf{v}, \mathbf{q} \rangle - \beta \delta_{B(L^\infty)}\left(\frac{\mathbf{q}}{\beta}\right) \end{aligned}$$

with:

$$K = \begin{pmatrix} F \\ \nabla \\ \Phi \end{pmatrix}.$$

Then we obtain algorithm 7 on the next page.

Algorithm 7: PALM for joint reconstruction and motion estimation

- Initialize: start with u^0 and $\mathbf{v}^0$ in $\mathbb{R}^N \times \mathbb{R}^M$.
- For each $k = 1, 2, \ldots$ generate sequence $\left(u^k, \mathbf{v}^k\right)$ as follows:

  1. Take $\gamma_1 > 1$, $\tau_1^k = \frac{\gamma_1}{L_1(\mathbf{v}^k)}$ and $\tau_1^k \sigma_1^k \|K\|^2 \leq 1$. Compute:

$$\overline{u}^{k+1} = u^k - \tau_1^k \left(\gamma A_{\mathbf{v}^k}^* A_{\mathbf{v}^k} u^k\right)$$

$$u^{k+1} = \begin{cases} \hat{p_1}^{k+1} &= p_1^k + \sigma_1^k F\left(\overline{u}^{k+1}\right) \\ p_1^{k+1} &= \frac{\hat{p_1}^{k+1} - \sigma_1^k \tilde{f}}{\lambda + \sigma_1^k} \\ \hat{p_2}^{k+1} &= p_2^k + \sigma_1^k \nabla \overline{u}^{k+1} \\ p_2^{k+1} &= \min\left(\alpha_1, \max\left(-\alpha_1, \hat{p_2}^{k+1}\right)\right) \\ \hat{p_3}^{k+1} &= p_3^k + \sigma_1^k \Phi\left(\overline{u}^{k+1}\right) \\ p_3^{k+1} &= \min\left(\alpha_2, \max\left(-\alpha_2, \hat{q}_2^{k+1}\right)\right) \\ \overline{\mathbf{p}}^{k+1} &= 2\mathbf{p}^{k+1} - \mathbf{p}^k \\ \hat{u}^{k+1} &= \overline{u}^{k+1} - \tau_1^k \left(F^*\left(\overline{p}_1^{k+1}\right) - \mathrm{div} \cdot \overline{p}_2^{k+1} + \Phi^*\left(\overline{p}_3^{k+1}\right)\right) \\ u^{k+1} &= \max\left(0, \hat{u}^{k+1}\right) \end{cases}$$

  2. Take $\gamma_2 > 1$, $\tau_2^k = \frac{\gamma_2}{L_2(u^{k+1})}$ and $\tau_2^k \sigma_2^k \|\nabla\|^2 \leq 1$. Compute:

$$\overline{\mathbf{v}}^{k+1} = \mathbf{v}^k - \tau_2^k \left(\gamma A_{u^{k+1}}^* A_{u^{k+1}} \mathbf{v}^k\right)$$

$$\mathbf{v}^{k+1} = \begin{cases} \hat{\mathbf{q}}^{k+1} &= \mathbf{q}^k + \sigma_2^k \begin{pmatrix} \nabla & 0 \\ 0 & \nabla \end{pmatrix} \overline{\mathbf{v}}^k \\ \mathbf{q}^{k+1} &= \min\left(\beta, \max\left(-\beta, \hat{\mathbf{q}}\right)\right) \\ \mathbf{v}^{k+1} &= \mathbf{v}^k - \tau_2^k \left(\begin{pmatrix} -\mathrm{div} & 0 \\ 0 & -\mathrm{div} \end{pmatrix} \overline{\mathbf{q}}\right) \end{cases}$$

Parameter choice and stopping criterion

To conclude this section on optimization via PALM we have to set the parameters τ_i^k, σ_i^k and γ_i for $i = 1, 2$ and we have to define a stopping criterion.

The stopping criterion follows naturally from the proof of theorem 7, given in [8]. To prove convergence to a critical point they prove that the sequence $(u^k, \mathbf{v}^k)$ for $k \geq 0$ generated by algorithm 5 is a Cauchy sequence. So the stopping criterion is:

$$\frac{|u^{k+1} - u^k| + |\mathbf{v}^{k+1} - \mathbf{v}^k|}{2|\Omega \times T|} < \mathrm{tol}.$$

Next we will set the parameters. Note that for γ_i we can pick an arbitrary number greater than one, so this means:

$$\begin{aligned} \tau_i^k &= \gamma_i \frac{1}{L_j} \quad i \neq j \\ &\geq \frac{1}{L_j} \quad i \neq j \end{aligned}$$

for $i = 1, 2$.

We also found in (4.25) the following bounds:

$$L_1(\mathbf{v}^k) \leq \gamma \max|\mathbf{v}^k|^2 \cdot 12,$$
$$L_2(u^k) \leq \gamma \max|u^k|^2 \cdot 8,$$

now we can write:

$$\tau_1^k \geq \frac{1}{L_1(\mathbf{v}^k)} \geq \frac{1}{\gamma \max|\mathbf{v}^k|^2 \cdot 12},$$
$$\tau_2^k \geq \frac{1}{L_2(u^k)} \geq \frac{1}{\gamma \max|u^k|^2 \cdot 8},$$

Hence we choose:

$$\tau_1^k = \frac{1}{\gamma \max|\mathbf{v}^k|^2 \cdot 12},$$
$$\tau_2^k = \frac{1}{\gamma \max|u^k|^2 \cdot 8}.$$

And for σ_i^k we choose:

$$\sigma_1^k = \frac{1}{\tau_1^k \|K\|^2} \qquad \sigma_2^k = \frac{1}{\tau_2^k \|\nabla\|^2},$$
$$= \frac{1}{\tau_1^k \cdot 8} \qquad = \frac{1}{\tau_2^k \cdot 8},$$

since $\|\nabla\|^2 \leq 8$, see [15] and for $\|K\|^2$ we found:

$$\|K\|^2 \leq \sup \begin{pmatrix} \|F\|^2 \\ \|\nabla\|^2 \\ \|\Phi\|^2 \end{pmatrix} \leq \begin{pmatrix} 1 \\ 8 \\ 1 \end{pmatrix},$$

see section 4.3.1.

4.6 Comparison of optimization methods

To get insight about the difference between the two presented optimization methods, we will compare them for a two dimensional nonconvex optimization problem. Recall from section 3.4.1 the following minimization problem:

$$\min_{(x,y)\in\mathbb{R}^2} \psi(x,y) = (xy-1)^2, \tag{4.28}$$

note that this function is nonconvex, but convex for fixed x respectively y, this function has the KL-property and is C^2. So we can solve this problem by either alternating PDHG or by PALM. This leads to the two algorithms on the next page.

For Alternating PDHG to converge, we must have:

$$\tau_x p^{k+1} \to 0 \quad \text{for } k \to \infty \tag{4.29}$$
$$\tau_y q^{k+1} \to 0, \quad \text{for } k \to \infty$$

and for PALM we must have:

$$2\tau_1^n (x^n y^n - 1) \to 0 \quad \text{for } n \to \infty \tag{4.30}$$
$$2\tau_2^n (x^{n+1} y^n - 1) \to 0 \quad \text{for } n \to \infty.$$

Alternating PDHG

- Initialize: start with x^1, y^1.
- For $n = 1,2,\ldots$ generate (x^n, y^n) as follows:
  - for $k = 1,2,\ldots$ generate x^{n+1} as follows:
  - Take τ_x, σ_x such that $\tau_x\sigma_x|y_n|^2 \leq 1$

$$\begin{cases} \hat{p}^{k+1} &= p^k + \sigma_x y^n \bar{x}^k \\ p^{k+1} &= \frac{\hat{p}^{k+1}-\sigma_x}{1+0.5\sigma_x} \\ x^{k+1} &= x^k - \tau_x p^{k+1} y^n \\ \bar{x}^{k+1} &= 2x^{k+1} - x^k \end{cases}$$

  - while primal-dual gap > tol.

  - for $k = 1,2,\ldots$ generate (y^{n+1}) as follows:
  - Take τ_y, σ_y such that $\tau_y\sigma_y|x_{n+1}|^2 \leq 1$

$$\begin{cases} \hat{q}^{k+1} &= q^k + \sigma_y x^{n+1} \bar{y}^k \\ q^{k+1} &= \frac{\hat{q}^{k+1}-\sigma_y}{1+0.5\sigma_y} \\ y^{k+1} &= y^k - \tau_y q^{k+1} x^{n+1} \\ \bar{y}^{k+1} &= 2y^{k+1} - y^k \end{cases}$$

  - while primal-dual gap > tol.
- while

$$|x^{n+1} - x^n| + |y^{n+1} - y^n| > \text{tol}$$

PALM

- Initialize: start with x^1, y^1, take $\gamma_1 > 0$, $\gamma_2 > 0$.
- for $n = 1,2,\ldots$ generate (x^n, y^n) as follows:

$$\begin{cases} \tau_1^n &= \frac{\gamma_1}{2(y^n)^2} \\ x^{n+1} &= x^n - 2\tau_1^n \cdot (x^n y^n - 1) y^n \\ \tau_2^n &= \frac{\gamma_2}{2(x^{n+1})^2} \\ y^{n+1} &= y^n - 2\tau_2^n \cdot (x^{n+1} y^n - 1) x^{n+1} \end{cases}$$

- while

$$|x^{n+1} - x^n| + |y^{n+1} - y^n| > \text{tol}$$

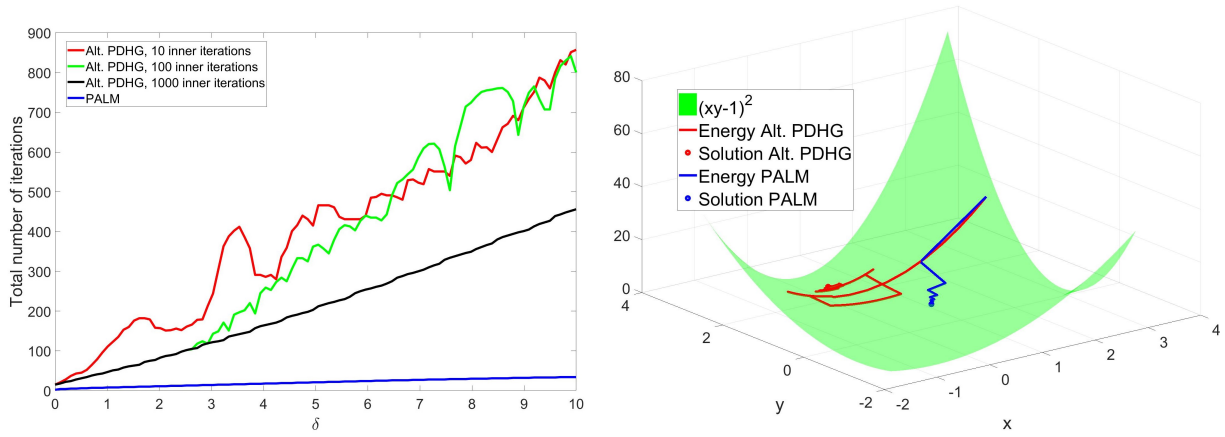
To compare both algorithms we will initialize both algorithms with $(x^1, y^1) = (3,2)$. We will set the tolerance to 10^{-6}. For Alternating PDHG we allow 1000 outer iterations and we will vary the number of inner iterations. For PALM we will allow also 1000 outer iterations.

It is important to realize that for this simple example we are able to explicitly calculate the values for τ_x, τ_y, σ_x and σ_y for Alternating PDHG and τ_1^k and τ_2^k for PALM. However for the more complicated joint reconstruction and motion estimation model we had to estimate upper bounds for these values. We will here investigate how the quality of this estimation influences the performance of the algorithm.

We will perturb the values of the parameters as follows:

$$\tau_x = \sigma_x = \frac{1}{|y^n| + \delta} \qquad \tau_1^n = \frac{1}{2(y^n)^2 + \delta}$$
$$\tau_y = \sigma_y = \frac{1}{|x^{n+1}| + \delta} \qquad \tau_2^n = \frac{1}{2(x^{n+1})^2 + \delta}.$$

In figure 4.1a one can see how the perturbation δ of the parameters influence the number of iterations needed to converge. We see that for Alternating PDHG the number of iterations grows much fast than for PALM. We also see that Alternating PDHG is sensitive for the number of inner iterations. If we choose this number too low, the subproblems might not converge. This result in a suboptimal approximation for x^{n+1}, which will give a suboptimal solution to the subproblem for y^{n+1}. The result of this can be seen in figure 4.1b, here we only allowed 10 inner iterations, which cause Alternating PDHG to 'wander around' the energy landscape before converging.

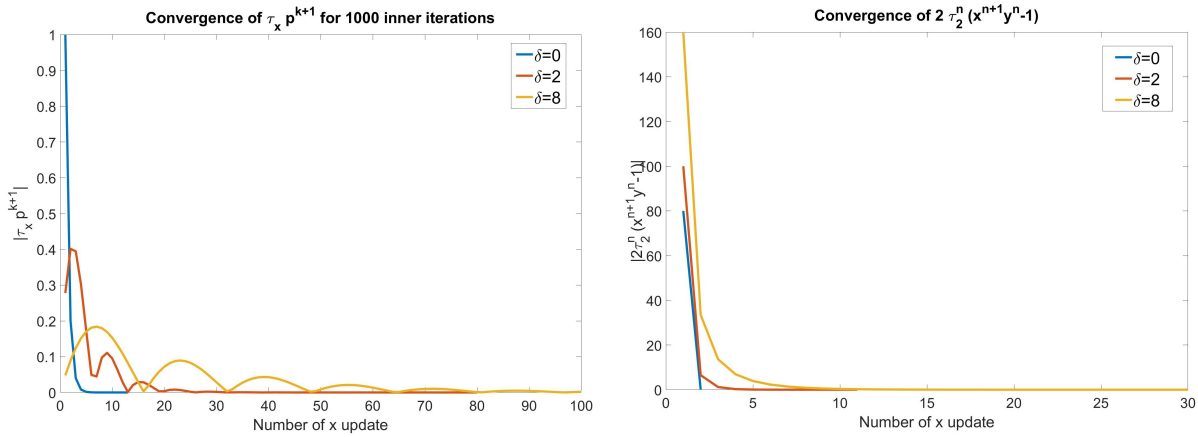


(a) Influence of pertubation δ of parameter values on number of iterations.

(b) Plot of energy for Alternating PDHG (red) with 10 inner iterations and PALM (blue). $\delta = 8$.

Figure 4.1

Finally we plotted the value of (4.29) and (4.30) for each iteration, in figure 4.2a and 4.2b. We see that for PALM, 4.30 converges faster and always decreases, which will result in a decrease in energy. For Alternating PDHG, 4.29 oscillates more as δ increases. This oscillating will also result in an oscillating energy. This indicates why the number of iterations grows more when using Alternating PDHG for increasing values of δ.



(a) Convergence plot for $\tau_x p^{k+1}$ for different values of δ.

(b) Convergence plot for $2\tau_1^n(x^n y^n - 1)$ for different values of δ.

Figure 4.2

Chapter 5

Results and discussion

In this chapter we will present and discuss the results of the joint reconstruction and motion estimation model for dynamic MRI. First we will discuss the developed synthetic datasets for dynamic MRI reconstruction and the used performance measures. Before we can compare both optimization methods we have to define the used subsampling pattern and the motion model for Alternating PDHG. After we defined this, we will use the synthetic datasets to compare the two optimization methods. After comparison of the two methods we will explore how the PALM algorithm performs on an experimental medical dataset.

5.1 Datasets & performance measures

A common challenge is that for real experimental medical data no ground truth for the flow field is avaliable. Having a ground truth at hand gives insight in the quality of the motion estimation. To address this we used the MPI Sintel dataset [14] used in computer vision to compare motion estimation methods. This dataset has a ground truth available for the flow field. We will use this dataset to develop a synthetic image sequence for dynamic MRI reconstruction.

We also make use of the 4D XCAT phantom [49] which was made available by the Radiotherapy group of the UMC Utrecht. This 4D phantom is developed for medical imaging research and we will also use this phantom to compare the two presented algorithms. Finally the Radiotherapy group of the UMC Utrecht shared experimental medical MRI data, which we will use to show the potential of the presented model for real world scenarios. In the following we will discuss each dataset and its purpose in detail.

5.1.1 Synthetic dataset Hand

The first synthetic dataset is developed using the MPI Sintel dataset. The clean version of the sequence 'Ambush_7' was used. We used part of the sequence, were one can see a hand picking up a stick. To create a first easy scenario we took the first frame from the sequence and scaled[1] the magnitude of the corresponding flow field. Then we created the second frame by using the discretization of the optical flow constraint:

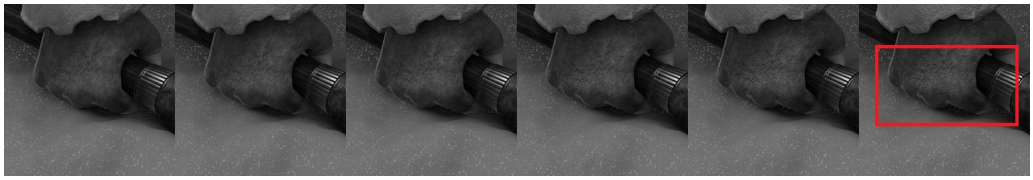
$$\partial_t u_t + \nabla u_t \cdot \mathbf{v}_t = \frac{u_{t+1} - u_t}{\delta_t} + \nabla u_t \cdot \mathbf{v}_t, \qquad \delta_t = 1$$
$$u_{t+1} = u_t - \nabla u_t \cdot \mathbf{v}_t.$$

Then we used the generated frame to generate the next frame, using the scaled ground truth flow field of the second frame. This sequence will satisfy the optical flow constraint, which makes it a good first evaluation. Next the sequence and the ground truth for the flow field are scaled to a size of 256×256 pixels and transformed from RGB to gray values between 0 and 1. In figure 5.1a one can see the whole sequence. The flow field can be seen in figure 5.1c. We will refer to this sequences as the *Hand - flow based* sequence.

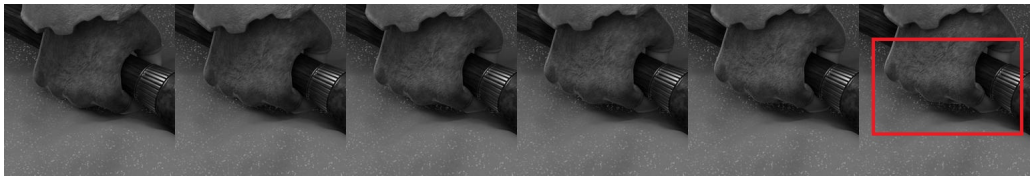
We will also use the original frames of the sequence to create a dataset. Now the image sequence and the flow

[1] The scaling was chosen such that $\max|\mathbf{v}| < 1$, since we have $\delta_x = \delta_y = 1$ (see appendix B).

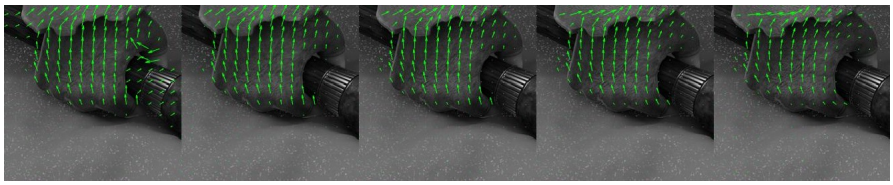
field will not satisfy the optical flow constraint perfectly, which makes that dataset more challenging. We will refer to this sequence as the *Hand - original* sequence, this sequence can be seen in figure 5.1b. The difference in magnitude of the flow field can be seen in figure 5.5b.



(a) Ground truth for the *Hand - flow based* sequence



(b) Ground truth for the *Hand - original* sequence

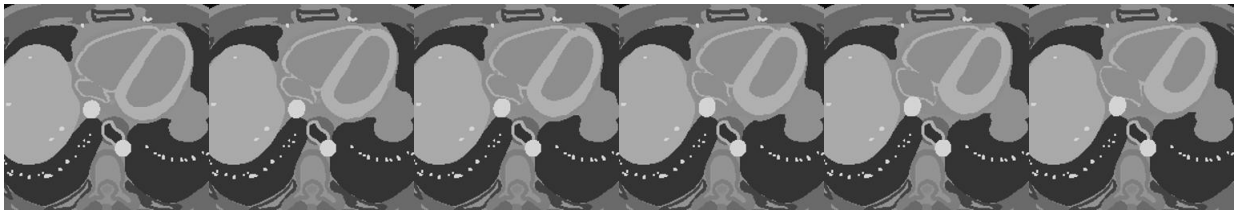


(c) Visualization of the flow field for the *Hand-original sequence*

Figure 5.1: Ground truth Hand sequences, in the red squares we clearly see the difference between the two datasets

5.1.2 XCAT 1 & 2

To investigate the influence of both optimization methods on the quality of the reconstruction the XCAT phantom [49] is used. This phantom is specially developed for the evaluation of medical reconstruction methods. We extracted two sequences from the 4D phantom. The first sequence shows the heart beating. In the second sequence a structure will appear over time due to the heartbeat. This results in a source in the flow field, which is challenging. Both sequences can be seen in figure 5.2.



(a) XCAT sequence 1

(b) XCAT sequence 2

Figure 5.2: Ground truth for the XCAT sequences

5.1.3 Experimental medical data

Finally we will use experimental medical data to show the potential of the model for real world scenarios. This data was made available by the radiotherapy group of the UMC Utrecht. The extracted sequence can be seen in figure 5.3.

In this sequence we see a transversal slice of the abdomen. So the back of the person is in the top of the frame. The movement we observe in the sequence is breathing, which is the upward and downward movement. We also see the aorta and the inferior vena cava. The aorta transports oxygen ridge blood from the heart to the rest of the body, the inferior vena cava transports oxygenated blood back to the heart. In figure 5.4 we see a schematic overview of the liver, both veins and the cross-section.

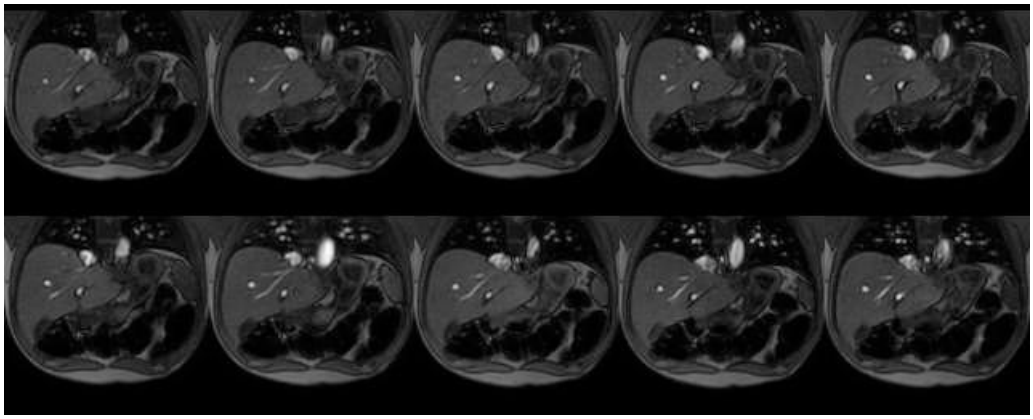


Figure 5.3: Experimental medical data sequence

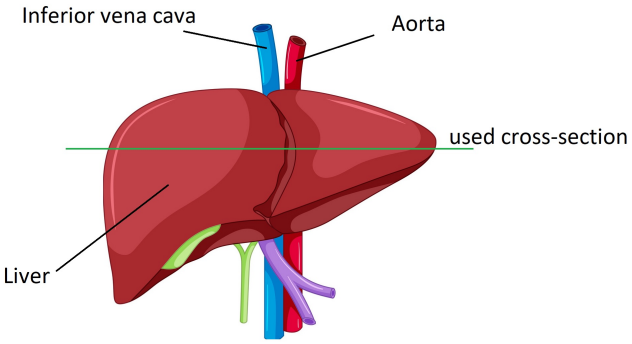


Figure 5.4: schematic overview of organs visuble in the experimental medical dataset.

5.1.4 Performance measures

To measure the quality of the reconstructed image sequence u and the estimated flow field $\mathbf{v}$ we will use the following performance measures.

Peak Signal to Noise Ratio - PSNR
The PSNR is defined as:

$$\text{PSNR}(u, u^0) = -\log\left(\frac{\|u - u^0\|_2}{\|u^0\|_\infty}\right),$$

were u^0 is the ground truth image and u the reconstruction. So the PSNR tells what ratio of the peaks of the signal is noise.

Structural Similarity Index Measure (SSIM)
The PSNR is easy to calculate and has a clear physical meaning but it will not reflect visual quality well, therefore we will also use the SSIM. In [18], [40] and [48] the SSIM is also used to measure the quality of reconstruction so we can compare our results easily.

In [51] the SSIM is defined. The SSIM compares the luminance, contrast and structure of the reconstructed image u and its ground truth u^0. The SSIM is defined as follows:

$$\text{SSIM}(u, u^0) = \frac{(2\mu_u\mu_{u^0} + c_1)(2\sigma_{u,u^0} + c_2)}{(\mu_u^2 + \mu_{u^0}^2 + c_1)(\sigma_u^2 + \sigma_{u^0}^2 + c_2)}$$

with μ the mean of u and u^0 and σ the standard deviation of u and u^0. Were c_1 and c_2 are defined by:

$$\begin{bmatrix} c_1 & c_2 \end{bmatrix} = \begin{bmatrix} (K_1 L)^2 & (K_2 L)^2 \end{bmatrix}$$

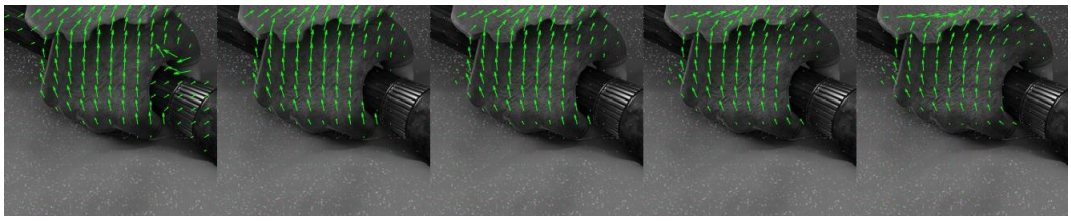
were K_1 and K_2 are small constants and L the range of the image. We use the implementation of [51] with $K = \begin{bmatrix} 0.01 & 0.03 \end{bmatrix}$ and $L = 1$ since all our images are scaled between $[0, 1]$.

Average Endpoint Error - AEE
Finally we use the AEE to measure the quality of the estimated flow field v. The AEE is defined in [41] as:

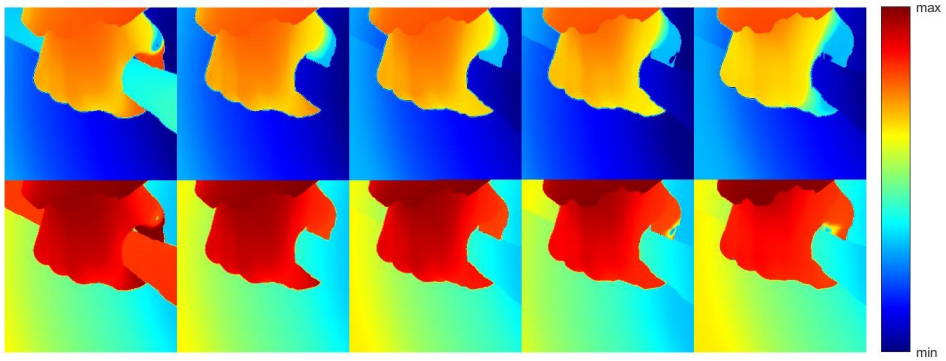
$$\text{AEE}(\mathbf{v}, \mathbf{v}^0) = \frac{1}{N} \sum_{n=1}^{N} \sqrt{(v_{x_n} - v_{x_n}^0)^2 + (v_{y_n} - v_{y_n}^0)^2}$$

with $N = n_x n_y (n_t - 1)$.

To also visually inspect the flow fields we will make use of two plots. The first one can be seen in figure 5.5a, were the green arrows indicate the direction of the flow, but are scaled in such a way that the arrows are visible. In 5.5b we can see the magnitude of the flow field, scaled between the maximum and minimum magnitude that appears in the figure. So magnitude plots with the same figure number, but a different letter have the same scaling and figures with a different figure numbers have a different scaling.



(a) Plot of the direction of the flow field for the *Hand - flow based* sequence, note that the flow field of the *Hand - original* sequence will have a flow field with a similiar direction.



(b) Plot of the magnitude of the flow field, in the upper row we see the magnitude of the *Hand - flow based* sequence, in the lower row the magnitude of the *Hand - original* sequence.

Figure 5.5

5.2 Subsampling pattern and motion model

Before we can compare the two optimization methods, we have to define the subsampling pattern and for Alternating PDHG we have to define the motion model.

5.2.1 Subsampling pattern

Now we will define the subsampling pattern P (2.1). As pointed out by [36] random subsampling will result in incoherent artefacts. However this randomness is limited by the fact that in practise the subsampling pattern must be feasable. In section 2.1.2 we briefly discussed subsampling patterns and assumed that we will use Cartesian subsampling only. Following [48], Variable density Incoherent Spatio-Temporal Acquistion (VISTA) of [1] is used.

This method is based on minimizing the Riesz energy, which in its simplest form comes down to determining the position of N samples such that their mutual distance (in a predefined distance measure) is maximized. However it is desirable to subsample the center of the k-space more, since most of the energy of the image is concentrated there. This is incorporated in the distance measure, such that sampling the center results in lower costs. An example of a resulting subsampling pattern can be found in figure 5.6.

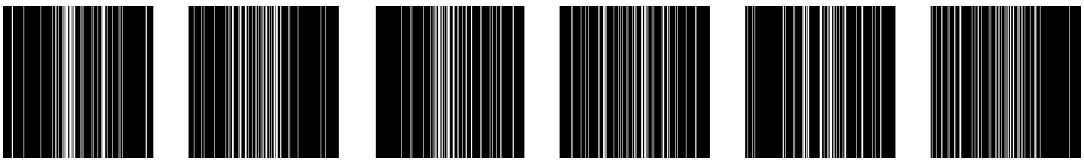


Figure 5.6: Example of a sequence of six frames with a 6 × subsampled VISTA pattern. So for each frame only $\frac{1}{6}$th of the data remains, which are the white lines. We see that for each frame the subsampling pattern is different.

5.2.2 Choice of motion model

Recall that in the definition of our model we could choose $p \in \{1,2\}$ for the motion model:

$$M(u,\mathbf{v}) = \frac{\gamma}{2}\|\partial_t u(\cdot,t) + \nabla u(\cdot,t)\cdot\mathbf{v}(\cdot,t)\|_p^p.$$

When using PALM, $M(u,\mathbf{v})$ had to be smooth and hence we choose $p = 2$. For Alternating PDHG we can use both $p \in \{1,2\}$. In this section we will compare the result of the motion estimation for both values of p when using Alternating PDHG. This comes down to only solving the subproblem for motion estimation in algorithm 3 (4.17). We will compare the results for the *Hand - flow based* sequence, where we take u as the ground truth. β was chosen such that the AEE is minimal.

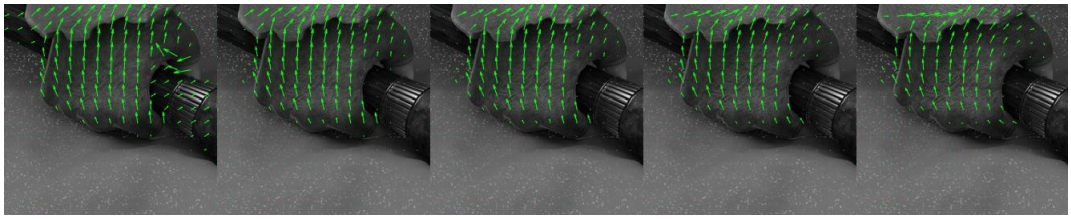
The AEE of the estimated flow field for the *Hand - flow based* sequence can be found in table 5.1. The flow fields can be seen in figure 5.7. From both the figure and the table it becomes clear that the model with $p = 1$ performs better. Due to the movement of the hand the shadow also has a changing intensity. It is known that using an L^1 data-fidelity is more robust to this, as it allows for violations of the optical flow constraint on a sparse set, as explained by [3].

So we will use $p = 1$ in case of optimization via Alternating PDHG. We see that for $p = 2$ the magnitude of the flow field is underestimated, this we will keep in mind when using PALM.

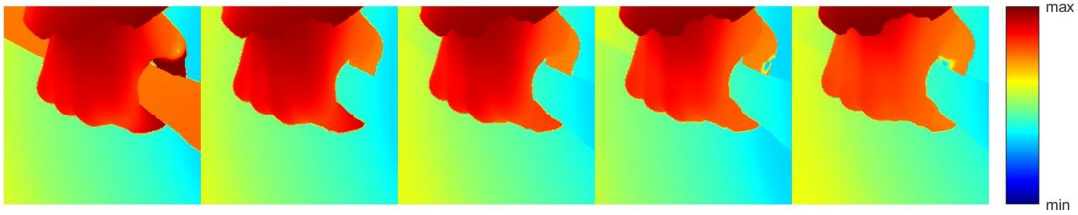
p	**AEE frame 1**	**AEE frame 2**	**AEE frame 3**	**AEE frame 4**	**AEE frame 5**
1	0.0190	0.0120	0.0115	0.0109	0.0099
2	0.2223	0.1881	0.1819	0.1685	0.1547

Table 5.1: AEE of estimated flow field for $p = \{1,2\}$

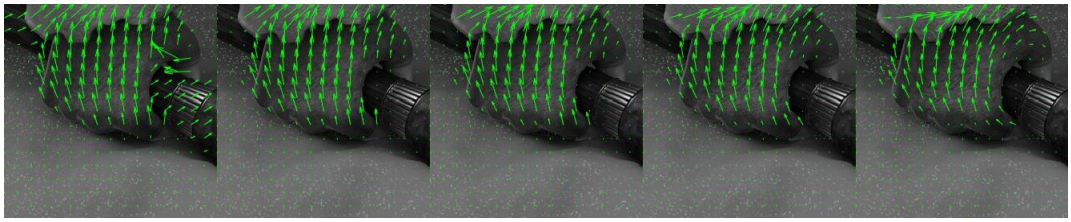
Result for choice of motion for Alternating PDHG



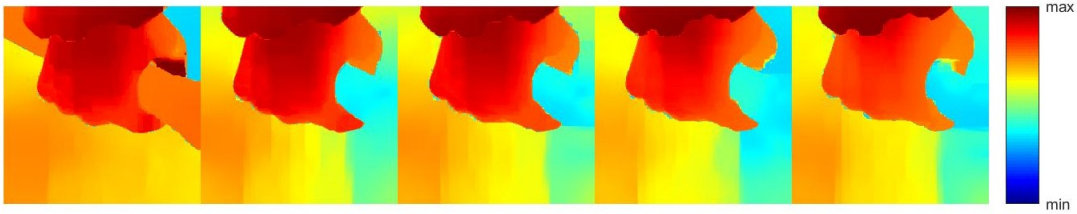
(a) Ground truth flow field for direction



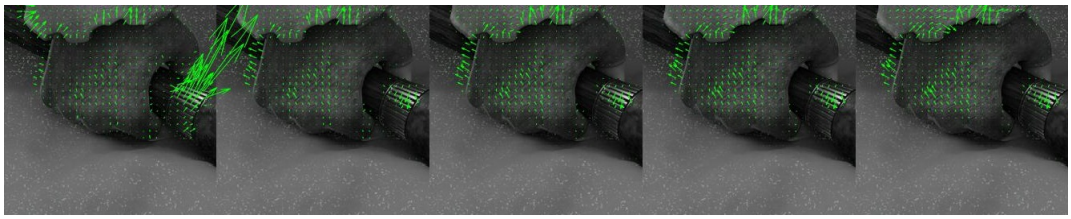
(b) Ground truth flow field for magnitude



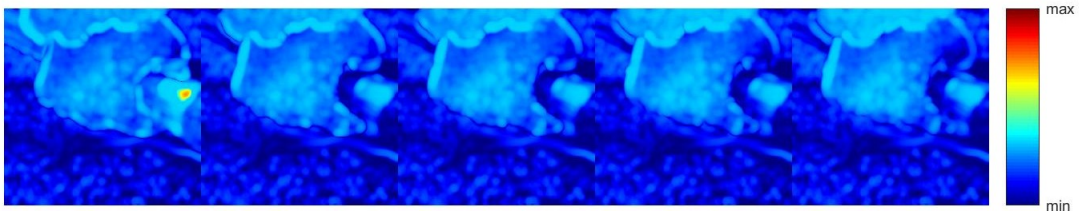
(c) Estimation flow field with $p = 1$ for direction



(d) Estimation flow field with $p = 1$ for magnitude



(e) Estimation flow field with $p = 2$ for direction



(f) Estimation flow field with $p = 2$ for magnitude

Figure 5.7: Result of motion estimation for $p = \{1, 2\}$

5.3 Comparison of optimization method on synthetic datasets

To compare both algorithms the following experiments are performed:

1. ***Hand- flow based*** **sequence**
   Since this sequence is generated using the ground truth flow field, we will use this sequence to verify the joint reconstruction and motion estimation model for both algorithms. All results will be compared to the case when first the image sequence is reconstructed frame-by-frame (so no motion involved) and then subsequently the motion is estimated on the obtained reconstruction.

2. ***Hand - original*** **sequence**
   This sequence will not satisfy the optical flow constraint and therefore the motion estimation will be more challenging. We will investigate how this influences both the reconstruction quality and the quality of the motion estimation.

3. **XCAT 1**
   The first XCAT sequence is used to investigate the influence of the joint model on the reconstruction quality since this sequence is simple in the sense that all objects are piece-wise constant, but has realistic motion. We will investigate the influence of the strength of the coupling between the reconstruction and motion estimation and the influence of higher subsampling factors on the reconstruction quality.

4. **XCAT 2**
   In the second XCAT sequence a structure will appear over time, which is realistic in practice, but very challenging for the motion model. We will use this sequence to compare how both algorithms will respond to this more challenging motion estimation.

For comparison we will always use a subsampling factor of 6, so we will use only $\frac{1}{6}$ of the pixels in the ground truth dataset for reconstruction and motion estimation. The values used as stopping criterion of Alternating PDHG and PALM for these experiments can be found in table 5.2.

All variables are initially zero-valued. The values of λ, α_1 and α_2 are chosen such that they result in the highest reconstruction quality in case of frame-by-frame reconstruction. β is chosen such that we have the best estimation of the flow field in terms of the AEE for motion estimation on the ground truth sequence. We will set γ initially equal, unless stated otherwise.

Alternating PDHG	
Tolerance alternations	10^{-4}
Maximal number of alternations	100
Tolerance inner iterations	10^{-4}
Maximal number of inner iterations	5000
PALM	
Tolerance iterations	10^{-4}
Maximal number of iterations	200

Table 5.2: Values used as stopping criterion of Alternating PDHG and PALM. We choose for 10^{-4} as tolerance because in all cases there was almost no improvement in SSIM or PSNR for the reconstruction if we set the tolerance lower.

5.3.1 *Hand - flow based* sequence

The results for the *Hand - flow based* sequence in terms of the performance measures can be found in table 5.3. We see that for both algorithms, on average and for each frame separately, using a joint model improves both the reconstruction quality and quality of motion estimation. So using a joint model is as expected beneficial. Using PALM gives even better results, so it is also beneficial to use a nonconvex optimization method.

In table 5.3 we see that the AEE for the flow field of frame 2 estimated via alternating PDHG is much higher than for the other frames. If we look at this flow field in figure 5.10 (c), we see that the flow fields 'bump into each other'. This creates folding, which we can see in figure 5.8. Note that these artefacts do not appear when using PALM.

In figures 5.9 and 5.10 the full results are given. From these figures we see that the reconstruction is slightly better when using PALM and for the motion estimation we see that PALM underestimates the magnitude of the flow. We already observed this underestimation when using $p = 2$ in section 5.2. This underestimation however, still gives lower AEE than the AEE for Alternating PDHG. Which performs better in terms of magnitude. However the *direction* of the flow field is often completely wrong (hence the folding), this results in a higher AEE, despite a better estimation of the magnitude. Here we see that the AEE punishes errors in terms of direction more than errors in terms of magnitude.

Measure	**Algorithm**	**Average**	**Frame 1**	**Frame 2**	**Frame 3**	**Frame 4**	**Frame 5**	**Frame 6**
SSIM	only recon.	0.6608	0.2451	0.4886	0.8385	0.8154	0.8033	0.7739
	alt. PDHG	0.8409	0.8263	0.8270	0.8571	0.8525	0.8453	0.8371
	PALM	*0.8672*	*0.8808*	*0.8780*	*0.8710*	*0.8657*	*0.8586*	*0.8493*
PSNR	only recon.	23.75	11.13	13.04	32.64	30.76	29.34	25.59
	alt. PDHG	33.00	31.55	31.61	34.38	34.10	33.65	32.69
	PALM	*35.02*	*35.14*	*35.39*	*35.41*	*35.15*	*34.75*	*34.28*
AEE	after recon.	1.1865	1.0651	2.8459	0.4865	0.3644	1.1705	
	alt. PDHG	0.6594	0.3902	1.5007	0.3144	0.2303	0.8614	
	PALM	*0.1834*	*0.2227*	*0.1887*	*0.1822*	*0.1686*	*0.1549*	

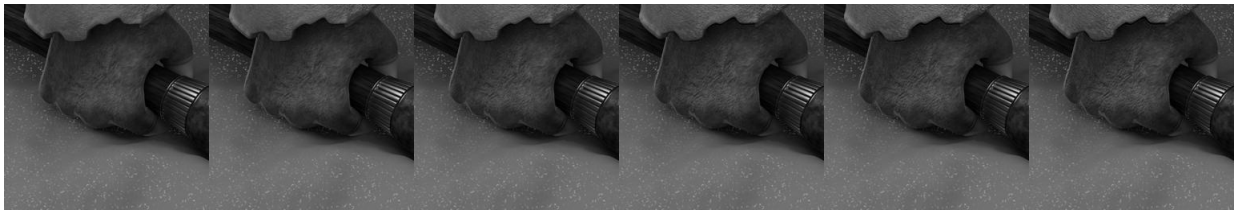
Table 5.3: Results of joint reconstruction and motion estimation for the 'Hand - flow based' sequence

(a) Boundary hand and shadow

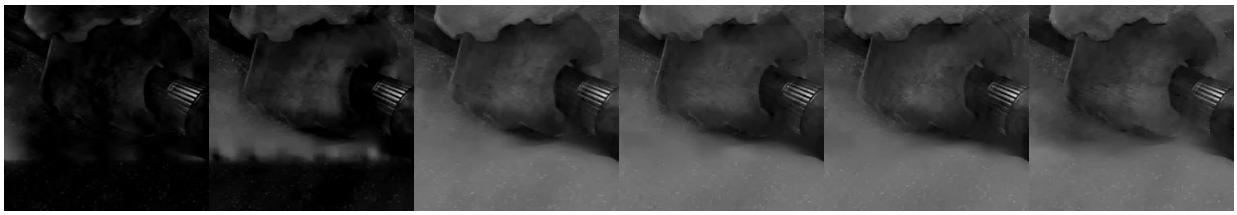
(b) Boundary hand and sleeve

Figure 5.8: Close up of frame 2 were we can see folding artefacts. Left: ground truth, middle: final reconstruction using alternating PDHG, right: final reconstruction using PALM. We see that this folding does not happen in the reconstruction obtained via PALM.

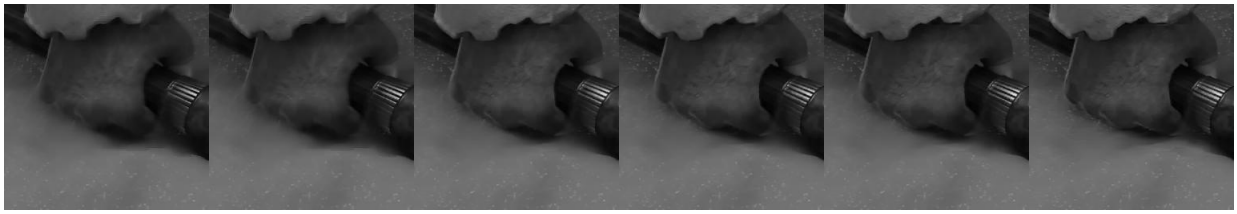
Result of reconstruction for *Hand - flow based* sequence



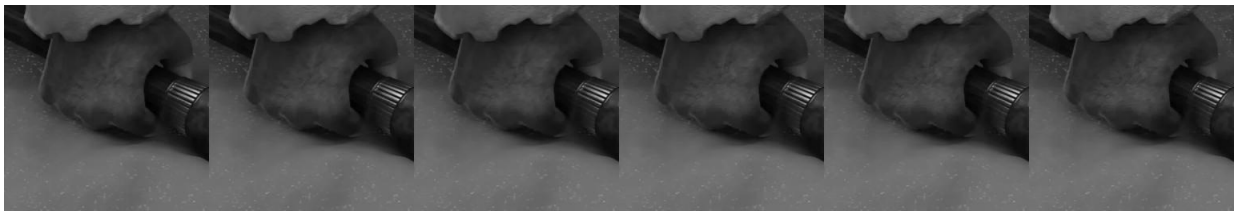
(a) Ground truth image sequence



(b) Result for only reconstruction



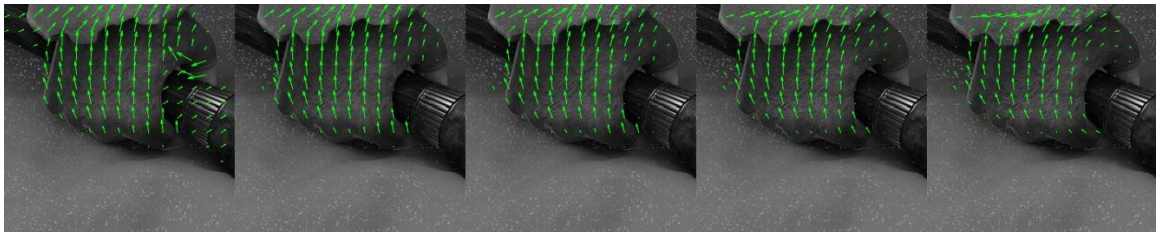
(c) Result for joint reconstruction and motion estimation via Alternating PDHG



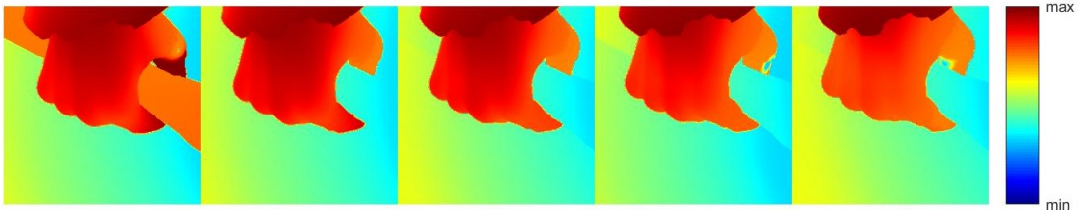
(d) Result reconstruction for joint reconstruction and motion estimation via PALM

Figure 5.9

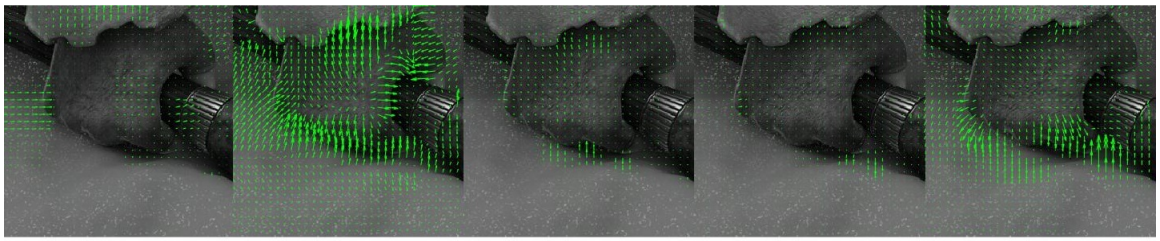
Result of motion estimation for *Hand - flow based* sequence



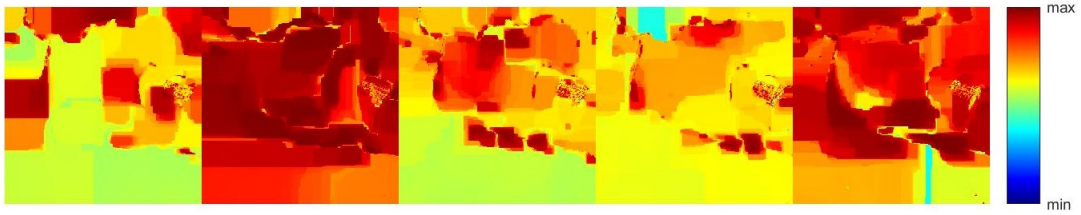
(a) Ground truth flow field for direction



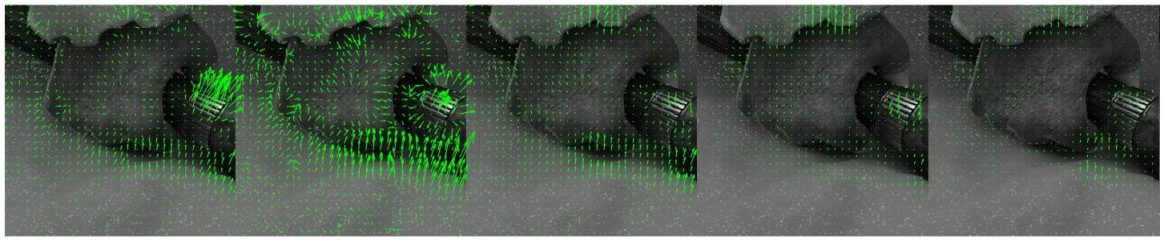
(b) Ground truth flow field for magnitude



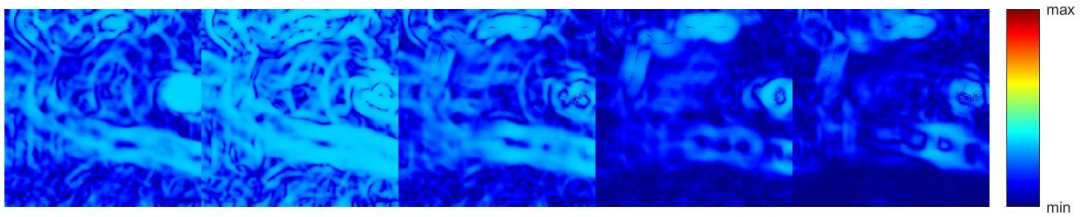
(c) Result direction motion estimation for joint reconstruction and motion estimation via Alternating PDHG



(d) Result magnitude motion estimation for joint reconstruction and motion estimation via Alternating PDHG



(e) Result direction motion estimation for joint reconstruction and motion estimation via PALM



(f) Result magnitude motion estimation for joint reconstruction and motion estimation via PALM

Figure 5.10

5.3.2 *Hand - original* sequence

In table 5.4 one can see the results for the *Hand- original* sequence. Here the motion estimation is more difficult since the motion does not satisfy the optical flow constraint. We see that on average joint reconstruction and motion estimation gives better results. The quality of the motion estimation dropped, which we expected since it cannot be explained perfectly by the optical flow constraint. This results also in a lower reconstruction quality. So the model for motion estimation is a limiting factor here. If we now compare both algorithms we see that for alternating PDHG the reconstruction quality of frames 3,4 and 5 drops in terms of SSIM compared to frame-by-frame reconstruction. So when the motion estimation becomes more challenging, errors, for example the folding mentioned earlier, are propagated in the sequence. When using PALM the quality improves for each frame compared to frame-by-frame reconstruction. The full results can be found in the appendix (A.1,A.2).

Measure	**Algorithm**	**Average**	**Frame 1**	**Frame 2**	**Frame 3**	**Frame 4**	**Frame 5**	**Frame 6**
SSIM	only recon.	0.6635	0.2450	0.4897	0.8416	0.8228	0.8109	0.7713
	alt. PDHG	0.7971	0.7694	0.7866	0.8161	0.8163	0.8077	0.7866
	PALM	*0.8276*	*0.8246*	*0.8369*	*0.8356*	*0.8297*	*0.8257*	*0.8132*
PSNR	only recon.	23.80	11.13	13.05	32.85	30.95	29.77	25.03
	alt. PDHG	30.98	28.80	30.16	32.49	32.63	31.94	29.87
	PALM	*31.94*	*30.02*	*32.22*	*33.06*	*32.73*	*32.35*	*31.22*
AEE	after recon.	2.27	2.4688	3.5035	1.7320	1.4432	2.1865	
	alt. PDHG	1.97	2.1205	2.5730	1.6888	1.4340	2.0121	
	PALM	*1.83*	*2.2264*	*1.8840*	*1.8208*	*1.6868*	*1.5495*	

Table 5.4: Results of joint reconstruction and motion estimation for the 'Hand - original' sequence

5.3.3 XCAT 1 sequence

In table 5.5 we see the results for the first XCAT sequence. Unfortunately we have no ground truth for the motion so we can inspect the flow fields only visually. The flow fields can be seen in figure A.4. We see again a lower magnitude for the flow in the case of using PALM. In the direction plots (a,c) we see that overall the hearts moves in the right direction, so the flow fields do make sense. When using Alternating PDHG we again see the folding artefacts due to mistakes in the motion estimation. The full result for the reconstruction can be found in figure A.3.

Looking more carefully in table 5.5,we see that the joint reconstruction models outperforms the frame-by-frame reconstruction on average. However for frame 3, we see that frame-by-frame reconstruction gives a better results than both joint algorithms. This can be explained by the fact that the XCAT sequence consist of piece-wise constant objects, which makes reconstruction easier. The used subsampling pattern for frame 3 can now still capture enough information for a good reconstruction. Since there are errors in the motion estimation included and this decreases the quality in this case.

Since the optimization method PALM gives better results we will now investigate for this method the influence of the coupling parameter γ and the influence of the subsampling factor on the reconstruction quality of the first XCAT sequence.

Measure	**Algorithm**	**Average**	**Frame 1**	**Frame 2**	**Frame 3**	**Frame 4**	**Frame 5**	**Frame 6**
SSIM	only recon.	0.7547	0.2843	0.4832	0.9751	0.9361	0.9283	0.9211
	alt. PDHG	0.8993	0.8392	0.8633	0.9245	0.9289	0.9265	0.9134
	PALM	*0.9539*	*0.9523*	*0.9569*	*0.9568*	*0.9554*	*0.9546*	*0.9473*
PSNR	only recon.	22.79	7.87	10.86	35.49	29.18	28.16	25.18
	alt. PDHG	28.84	25.24	25.99	31.03	30.89	30.51	29.41
	PALM	*33.44*	*32.41*	*33.93*	*34.35*	*33.93*	*33.50*	*32.53*

Table 5.5: Results of joint reconstruction and motion estimation for the 'XCAT 1' sequence

Influence of coupling γ between reconstruction and motion estimation
For joint reconstruction and motion estimation via PALM we will now investigate the influence of the parameter γ, which influences the impact of the motion estimation on the reconstruction. The results can be found in table 5.6.

We see that on average the SSIM is the highest for $\gamma = 0.25$, but the PSNR is the highest for $\gamma = 0.5$. For $\gamma = 0.1$ we see the PSNR of the first two frames drop in quality, this are the two frames with the lowest reconstruction quality in the case of frame-by-frame reconstruction and hence we see that a stronger coupling between the motion estimation and reconstruction is needed, since this models the temporal information. For a too strong coupling of $\gamma = 1.5$ however, the reconstruction quality drops as well, since then the flow fields dominate the reconstruction.

γ	Measure	Average	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6
0.1	SSIM	0.9608	0.9541	0.9653	0.9712	0.9655	0.9595	0.9492
	PSNR	30.84	23.03	26.63	35.79	34.47	33.09	32.00
0.25	SSIM	0.9626	0.9643	0.9666	0.9663	0.9633	0.9615	0.9538
	PSNR	33.59	31.20	33.38	35.37	34.62	33.94	33.04
0.5	SSIM	0.9584	0.9587	0.9620	0.9615	0.9593	0.9583	0.9508
	PSNR	33.81	32.81	34.31	34.81	34.25	33.79	32.90
0.75	SSIM	0.9558	0.9549	0.9591	0.9588	0.9571	0.9563	0.9487
	PSNR	33.62	32.68	34.15	34.52	34.05	33.63	32.71
1	SSIM	0.9539	0.9523	0.9569	0.9568	0.9554	0.9546	0.9473
	PSNR	33.44	32.41	33.93	34.35	33.93	33.50	32.53
1.5	SSIM	0.9494	0.9474	0.9522	0.9526	0.9514	0.9500	0.9429
	PSNR	33.03	31.90	33.46	34.01	33.61	33.10	32.08

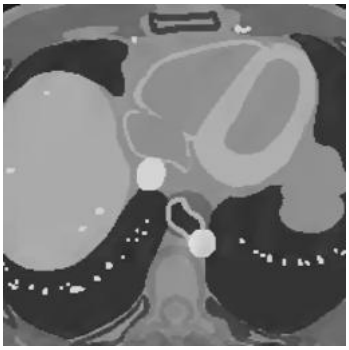
Table 5.6: Reconstruction quality for different values of γ for the XCAT 1 sequence

Influence of subsampling factor
Finally we will investigate the influence of higher subsampling factors on the reconstruction quality. The result can be found in table 5.7. First observation is that the joint approach outperforms frame-by-frame reconstruction for all subsampling factors on average. For the joint approach we see that the quality of the reconstruction up to a subsampling factor of 10 is quite good, however from a subsampling of 12 and higher, the quality decreases.

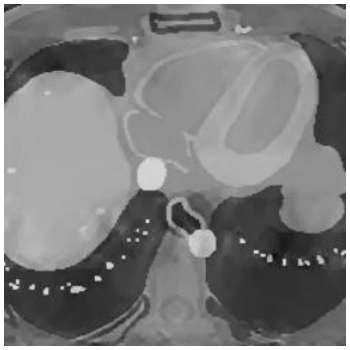
The reconstruction of frame 5 for each subsampling factor can be seen in figure 5.11 in the case of frame-by-frame reconstruction and in figure 5.12 in the case of joint reconstruction. We see that for $R = 12$ and $R = 14$ the joint reconstruction is blurry, but the shapes in the image are better visible than for frame-by-frame reconstruction.

	4	6	8	10	12	14
Frame-by-frame						
Average SSIM	0.8675	0.7547	0.6663	0.6161	0.5575	0.5178
Average PSNR	29.83	22.79	18.31	16.08	14.99	14.45
Joint						
Average SSIM	0.9752	0.9584	0.9446	0.9250	0.8348	0.8187
Average PSNR	36.59	33.81	32.06	30.49	23.53	23.03

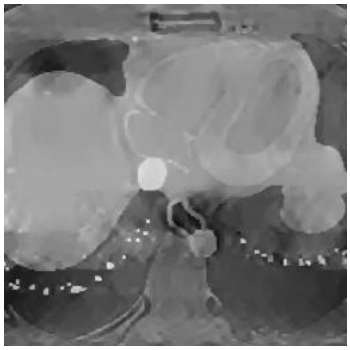
Table 5.7: Average reconstruction quality for different subsampling factors for the XCAT 1 sequence with $\gamma = 0.5$, compared to frame-by-frame reconstruction.



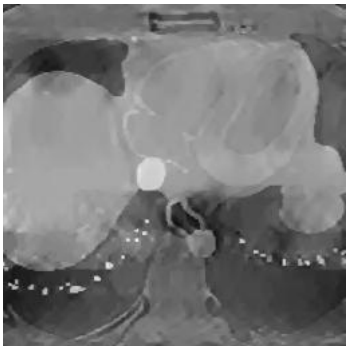
(a) $R = 4$



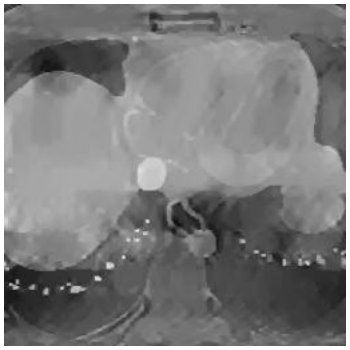
(b) $R = 6$



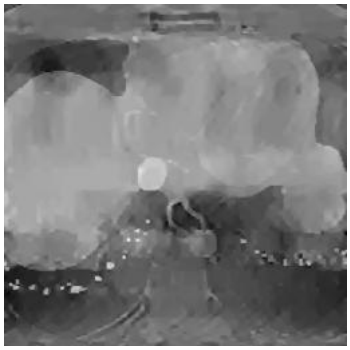
(c) $R = 8$



(d) $R = 10$

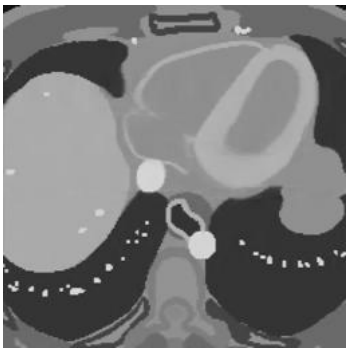


(e) $R = 12$

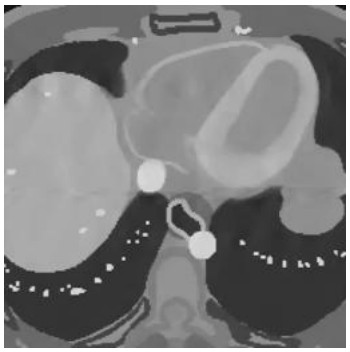


(f) $R = 14$

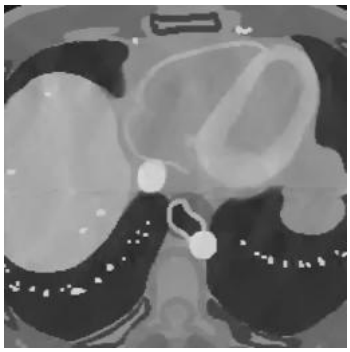
Figure 5.11: Result of frame-by-frame reconstruction of frame 5 for different subsampling factors R.



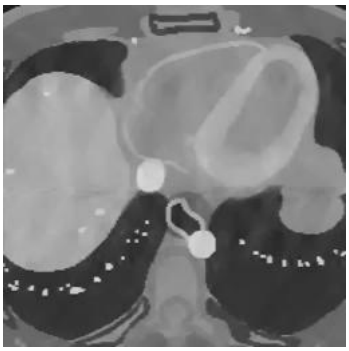
(a) $R = 4$



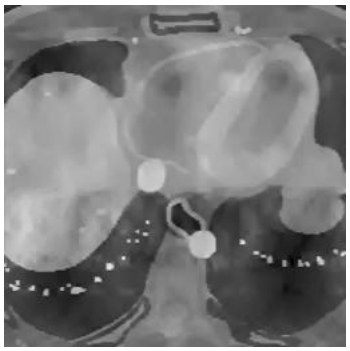
(b) $R = 6$



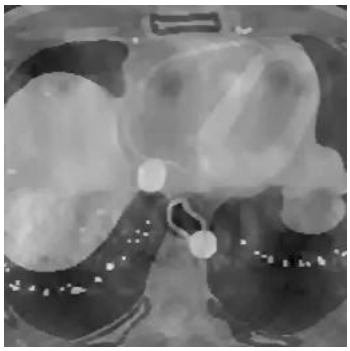
(c) $R = 8$



(d) $R = 10$



(e) $R = 12$



(f) $R = 14$

Figure 5.12: Result of joint reconstruction via PALM of frame 5 for different subsampling factors R

5.3.4 XCAT 2 sequence

For this sequence we find results comparable to the results of the XCAT 1 sequence, see table 5.8. In this sequence a structure appears over time. We see in figure 5.13 that for reconstruction using alternating PDHG this structure is not present but for PALM it is. The full results can be found in the appendix (A.5,A.6).

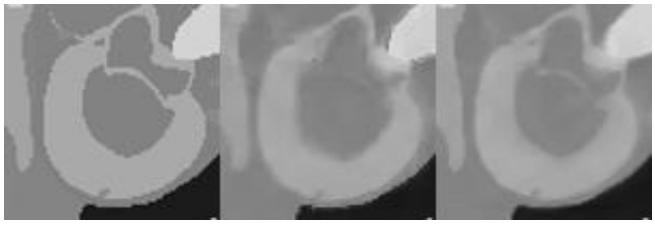


Figure 5.13: Close up of frame 5 were the structure appeared. Left: ground truth, middle: reconstruction via alt. PDHG, right: reconstruction via PALM

Measure	Algorithm	Average	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6
SSIM	only recon.	0.7542	0.3376	0.4972	0.9797	0.9615	0.9456	0.8038
	alt. PDHG	0.9064	0.8554	0.8894	0.9297	0.9317	0.9255	0.9064
	PALM	*0.9639*	*0.9581*	*0.9642*	*0.9643*	*0.9665*	*0.9660*	*0.9643*
PSNR	only recon.	22.29	7.87	11.61	36.74	31.54	29.33	16.68
	alt. PDHG	29.23	26.05	27.72	31.27	31.21	30.54	28.57
	PALM	*34.55*	*32.98*	*34.64*	*35.15*	*35.16*	*34.79*	*34.58*

Table 5.8: Results of joint reconstruction and motion estimation for the XCAT 2 sequence

5.4 Application on experimental medical data

To show the potential of the work presented here we will test the PALM algorithm for joint reconstruction and motion estimation on the experimental medical data. We will compare the joint reconstruction and motion estimation model with frame-by-frame reconstruction in terms of the SSIM and PSNR for different subsampling factors. In table 5.9 we see that the joint reconstruction performance is better than frame-by-frame reconstruction and the improvement grows as the subsampling factor gets higher.

In figure 5.14b we can see the reconstruction for $R = 4$. In 5.14c we see the direction of the flow fields. We see a clear breathing pattern, first the flow is mainly upwards, during exhalation, then the flow is downward in the opposite direction during inhalation. Besides the breathing pattern we see also see blood flowing through both veins, which creates sources and sink in the flow fields. This is also reflected in figure 5.14d. So our joint model also results in a meaningful motion estimation on experimental medical data.

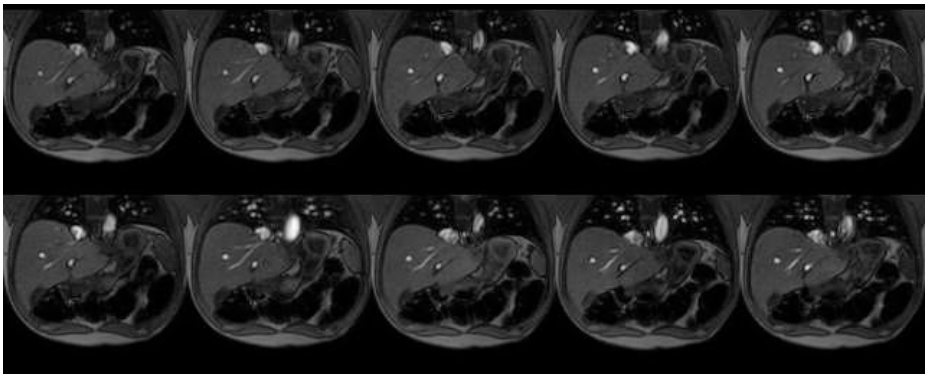
The reconstruction for the other subsampling factors can be found in the appendix, figure A.7 till figure A.10. For $R = 8$ the motion estimation is given in figure A.11 and figure A.12. Here we still observe the same motion pattern as for $R = 4$, so for a higher subsampling factor the motion estimation remains meaningful.

R	Frame-by-frame	Joint
4	0.8532	0.9006
6	0.7142	0.8580
8	0.6066	0.7904
10	0.5107	0.7417
12	0.4982	0.7330

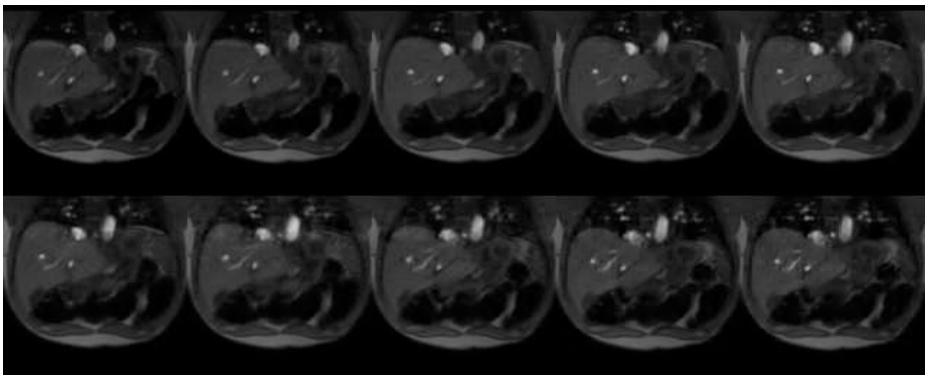
R	Frame-by-frame	Joint
4	31.02	33.58
6	26.53	31.91
8	24.42	29.69
10	23.08	28.09
12	22.92	27.95

Table 5.9: Average SSIM and PSNR for frame-by-frame reconstruction and joint reconstruction for experimental medical data

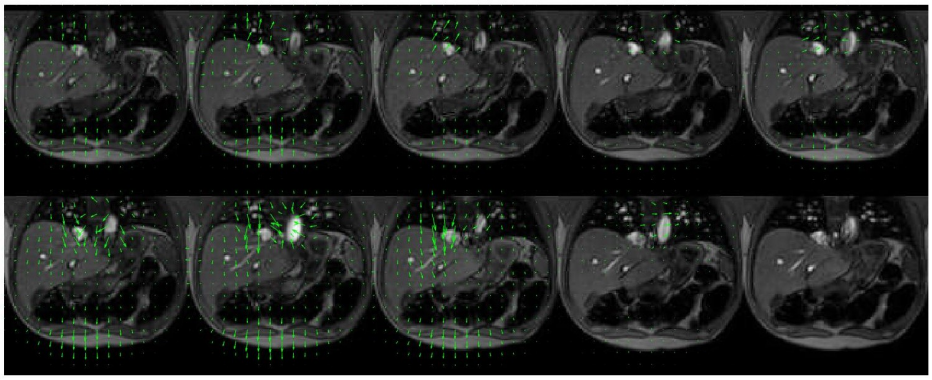
Results for experimental medical sequence for $R = 4$



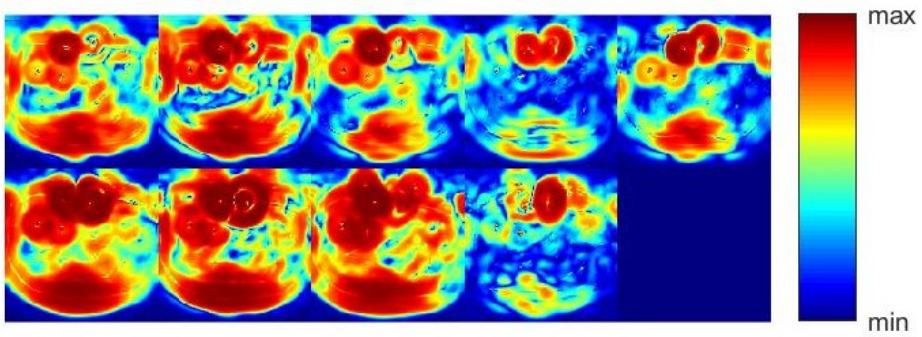
(a) Ground truth



(b) Reconstruction for $R = 4$



(c) Direction of estimated flow field



(d) Magnitude of estimated flow field

Figure 5.14: Results of joint reconstruction and motion estimation for experimental medical data

Chapter 6

Conclusion and outlook

Conclusion

To reduce scanning times in dynamic MRI, subsampling of the measurements is needed in practise. With increasing subsampling factors, frame-by-frame reconstruction does in general lead to artefacts and insufficient reconstruction quality. Employing information from neighboring frames in time can offer a way to increase the quality.

To achieve this, we developed a joint reconstruction and motion estimation model for dynamic MRI. Jointly estimating the motion will result in flow fields which explicitly explain the connection between two frames. The motion estimation is based on the optical flow constraint and the reconstruction uses Compressed Sensing as regularization.

We proved the existence of a minimizer, but could not prove uniqueness since the joint model is nonconvex. Due to this nonconvexity the optimization is challenging. In [18], [25] and [35] this is addressed by solving the convex subproblems for reconstruction and motion estimation in an alternating fashion. However no knowledge about the convergence of this approach is known. To address this we developed an algorithm based on the Proximal Alternating Linearized Minimization (PALM) algorithm [8]. The convergence to a minimum of the PALM algorithm relies on the Kurdyka-Lojasiewicz property. We proved that the KL-property holds for the presented joint model. All other properties that must be satisfied to guarantee the convergence of the PALM algorithm to a minimum are proved as well.

To compare both optimization methods we also need a ground truth for the flow field. To address the lack of ground truth for medical datasets, we developed a synthetic dataset from datasets used in computer vision for motion estimation evaluation. These datasets have a ground truth for the flow field available. Combined with the XCAT phantom [49], which is specially developed for the evaluation of medical reconstruction methods, we compared both optimization approaches.

In all our experiments we saw that using a joint reconstruction and motion estimation model instead of a frame-by-frame reconstruction enhances the reconstruction quality. We also observed that using PALM as optimization method gives a higher reconstruction quality than using Alternating PDHG as optimization method in our cases.

It became clear that the quality of the motion estimation is a limiting factor for the overall quality of the reconstruction. We saw that the estimated flow fields reflect the motion in the sequence, but there were some limitations. In section 5.2.2 we saw that using an L^1 data-fidelity resulted in better motion estimation. So we used $p = 1$ for the Alternating PDHG algorithm. However in the case of MRI reconstruction, the motion estimation introduced folding artefacts. When using PALM as optimization method these artefacts were not present. For the PALM algorithm we needed $p = 2$ since we needed to satisfy a smoothness requirement. We saw in section 5.2.2 already that using $p = 2$ resulted in a flow field which underestimated the magnitude of the flow, we saw this also in all later experiments.

For PALM as optimization method we also explored the influence of the strength of the coupling between the reconstruction and motion estimation on the reconstruction quality. We saw that in case of a very weak coupling the reconstruction did not benefit from the temporal information the motion offers, in case of very strong coupling new artefacts are introduced due to imperfect motion estimation. We also looked at the robustness for higher subsampling factors, from which we saw that joint reconstruction is beneficial for all subsampling factors. Finally we observed in our cases that PALM is more robust for an appearing structure in the sequence over time than Alternating PDHG.

To show the potential for real world scenarios, the presented joint model optimized via PALM, was applied on an experimental medical dataset of a transversal cross-section of the abdomen. The use of the joint model resulted in higher reconstruction qualities than for a frame-by-frame approach. We also saw that the obtained flow fields reflected the expected motion in the sequence.

Outlook

To conclude we will evaluate on choices made in this thesis and give suggestions for further research.

Forward operator MRI reconstruction
When we defined the forward operator for MRI reconstruction we made two assumptions. Firstly, we assumed that the phase of the received signal by the MRI scanner is negligible small, and hence the resulting image is real-valued. This assumption is chosen mainly because the motion estimation is defined on real-valued images and since in the joint model the motion is directly incorporated we needed the resulting reconstructed image to be real-valued as well. However, neglectingnthe complex part of the reconstruction can result in the loss of information which might influence the reconstruction quality. Secondly, we only considered Cartesian subsampling patterns. However the chosen subsampling pattern has a lot of influence on the artefacts. A good suggestion for further research would be to investigate the influence of different types of subsampling pattern (Cartesian, radial, spiral) and what the optimal subsampling patterns would be.

Motion model
As mentioned before, the used motion model has some limitations. Bu using the PALM algorithm no convexity restrictions are imposed on the model. So one possible extension would be to use a non-linear model for the motion estimation, which could better deal with large displacements. Another suggestion is to make the motion estimation also dependent on all available frames. A possible model extension is given by [9] and [16], were the optical flow is modeled in the field of optimal control. In this model we seek for a flow field, which explains the movement in the image sequence over the whole sequence, instead of frame-by-frame. Additionally, in this model we could reconstruct more frames that the number of frames for which we have given measurements. So we could also interpolate.

Regularization
For both the image sequence u and the flow field $\mathbf{v}$, we made use of simple regularization terms. This choice allowed us to prove the KL-property and to study the optimization in more detail. Using more advanced regularizers will make the presented optimization more involved, but might improve the results. An interesting extension would be to use the Low-rank plus Sparse decomposition of [40] as regularization for the reconstruction. This decomposition is based on separation of the static background and dynamic foreground. Using this regularization in a joint reconstruction and motion estimation model might work quite well as the motion could be estimated only on the sparse dynamic foreground. Then the separation in the low-rank and sparse parts is also motivated by the explicitly modeled motion.

Adding inertia
A straightforward extension for the PALM algorithm is to add an inertia term as mentioned in section 4.4. This will lead to the iPalm algorithm [43]. The inertia term can be seen as an explicit finite difference discretization of the Heavy-ball with friction dynamical system. Intuitively, adding momentum to the system makes it possible to 'jump' out of a local minimum and seek for a better minimum in a nonconvex landscape. In [43] they show that choosing the right parameters for the initial term gives better results than using no inertial term.

Bibliography

[1] R. Ahmad, H. Xue, S. Giri, Y. Ding, J Craft, and O. P. Simonetti. Variable density incoherent spatiotemporal acquisition (VISTA) for highly accelerated cardiac MRI. *Magnetic Resonance in Medicine*, 74(5):1266–1278, 2015.

[2] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal Alternating Minimization and Projection Methods for Nonconvex Problems: An Approach Based on the Kurdyka-Łojasiewicz Inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.

[3] G. Aubert, R. Deriche, and P. Kornprobst. Computing Optical Flow via Variational Techniques. *SIAM Journal on Applied Mathematics*, 60(1):156–182, 1999.

[4] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R Szeliski. A Database and Evaluation Methodology for Optical Flow. *International Journal of Computer Vision*, 92:1–31, 2011.

[5] M. Bertero, T.A. Poggio, and V. Torre. Ill-posed problems in early vision. *Proceedings of the IEEE*, 76(8):869–889, 1988.

[6] J. Bochnak, M. Coste, and M. Roy. *Real Algebraic Geometry*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[7] J. Bolte, A. Daniilidis, and A. Lewis. The Łojasiewicz Inequality for Nonsmooth Subanalytic Functions with Applications to Subgradient Dynamical Systems. *SIAM Journal on Optimization*, 17(4):1205–1223, 2007.

[8] J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014.

[9] A. Borzì, K. Ito, and K. Kunisch. Optimal control formulation for determining optical flow. *SIAM Journal on Scientific Computing*, 24(3):818–847, 2003.

[10] S. Boyd, N. Parikh, E. Chu, and J. Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends R in Machine Learning*, 3(1):1–122, 2010.

[11] C. Brune. *4D imaging in tomography and optical nanoscopy*. Phd thesis, University of Münster, Germany, 2010.

[12] M. Burger, H. Dirks, and C. Schönlieb. A Variational Model for Joint Motion Estimation and Image Reconstruction. *SIAM Journal on Imaging Sciences*, 11(1):94–128, 2016.

[13] M. Burger and S. Osher. A Guide to the TV Zoo. pages 1–70. 2013.

[14] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conf. on Computer Vision (ECCV)*, pages 611–625, 2012.

[15] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[16] K. Chen and D. A. Lorenz. Image sequence interpolation Using Optimal Control. *Journal of Mathematical Imaging and Vision*, 41(1):222–238, 2011.

[17] I. Daubechies. Orthonormal Bases of Compactly Supported Wavelets II. Variations on a Theme. *SIAM Journal on Mathematical Analysis*, 24(2):499–519, 1993.

[18] H. Dirks. *Variational Methods for Joint Motion Estimation and Image Reconstruction*. PhD thesis, University of Münster, Germany, 2015.

[19] D. Donoho, A. Maleki, and M. Sharam. WaveLab. `http://statweb.stanford.edu/~wavelab/Wavelab_850/index_wavelab850.html`.

[20] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. FlowNet: Learning Optical Flow With Convolutional Networks. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015.

[21] J. Douglas and H. H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society*, 82(2):421–421, 1956.

[22] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1-3):293–318, 1992.

[23] M. J. Ehrhardt and M. M. Betcke. Multi-Contrast MRI Reconstruction with Structure-Guided Total Variation. *SIAM Journal on Imaging Sciences*, 9(3):1084–1106, 2015.

[24] E. Esser, X. Zhang, and T. F. Chan. A General Framework for a Class of First Order Primal-Dual Algorithms for Convex Optimization in Imaging Science. *SIAM Journal on Imaging Sciences*, 3(4):1015–1046, 2010.

[25] L. Frerking. *Variational Methods for Direct and Indirect Tracking in Dynamic Imaging*. PhD thesis, University of Münster, Germany, 2016.

[26] T. Goldstein, M Li, X. Yuan, E. Esser, and R. Baraniuk. Adaptive Primal-Dual Hybrid Gradient Methods for Saddle-Point Problems. 2013.

[27] Mark A. Griswold, Peter M. Jakob, Robin M. Heidemann, Mathias Nittka, Vladimir Jellus, Jianmin Wang, Berthold Kiefer, and Axel Haase. Generalized Autocalibrating Partially Parallel Acquisitions (GRAPPA). *Magnetic Resonance in Medicine*, 47(6):1202–1210, 2002.

[28] B. K.P. Horn and B. G. Schunck. Determining Optical Flow. *Artificial Inteligence*, 17:185–203, 1980.

[29] F. Knoll, K. Bredies, T. Pock, and R. Stollberger. Europe PMC Funders Group Second Order Total Generalized Variation (TGV) for MRI. *Magnetic Resonance in Medicine*, 65(2):480–491, 2014.

[30] Krzysztof Kurdyka. On gradients of functions definable in 0-minimal structures. *Annales de l'institut Fourier*, 1(2):131–295, 1998.

[31] P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal of Numerical Analysis*, 16(6):964–979, 1979.

[32] S. Lojasiewicz. Sur la géométrie semi-et sous-analytique. *Annales de l'institut Fourier*, 43(5):1575–1595, 1993.

[33] D. A. Lorenz and T. Pock. An Inertial Forward-Backward Algorithm for Monotone Inclusions. *Journal of Mathematical Imaging and Vision*, 51(2):311–325, 2015.

[34] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI)*, 2:674–679, 1981.

[35] F. Lucka, N. Huynh, M. Betcke, E. Zhang, P. Beard, B. Cox, and S. Arridge. Enhancing Compressed Sensing 4D Photoacoustic Tomography by Simultaneous Motion Estimation. *eprint arXiv:1802.05184*, pages 1–29, 2018.

[36] M. Lustig, D. Donoho, and J. M. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.

[37] D. Maurer and A. Bruhn. ProFlow: Learning to Predict Optical Flow. *Preprint: arXiv:1806.00800v1*, pages 1–16, 2018.

[38] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[39] P. Ochs, Y. Chen, T. Brox, and T. Pock. iPiano: Inertial Proximal Algorithm for Non-Convex Optimization. *SIAM Journal on Imaging Sciences*, 7(2):1388–1419, 2014.

[40] R. Otazo, E. Candès, and D. K. Sodickson. Low-rank plus sparse matrix decomposition for accelerated dynamic MRI with separation of background and dynamic components. *Magnetic Resonance in Medicine*, 73(3):1125–1136, 2015.

[41] M. Otte and H. Nagel. Optical flow estimation: Advances and comparisons. In *Computer Vision-ECCV'94*, pages 49–60. Springer, Berlin, Heidelberg, 1994.

[42] N. Papenberg, A Bruhn, T. Brox, S. Didas, and J. Weickert. Highly Accurate Optic Flow Computation with Theoretically Justified Warping. *International Journal of Computer Vision*, 67(2):141–158, 2006.

[43] T. Pock and S. Sabach. Inertial Proximal Alternating Linearized Minimization (iPALM) for Nonconvex. *SIAM Journal on Imaging Sciences*, 9(4):1756–1787, 2016.

[44] M. J.D. Powell. On search directions for minimization algorithms. *Mathematical Programming*, 4(1):193–201, 1973.

[45] K. P. Pruessmann, M. Weiger, M. B. Scheidegger, and P. Boesiger. SENSE: Sensitivity encoding for fast MRI. *Magnetic Resonance in Medicine*, 42(5):952–962, 1999.

[46] R.T. Rockafellar. *Convex Analysis*, pages 32,33,46. 1970.

[47] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.

[48] M. Schloegl, M. Holler, A. Schwarzl, K. Bredies, and R. Stollberger. Infimal convolution of total generalized variation functionals for dynamic MRI. *Magnetic Resonance in Medicine*, 78(1):142–155, 2017.

[49] W. P. Segars, G. Sturgeon, S. Mendonca, Jason Grimes, and B. M.W. Tsui. 4D XCAT phantom for multimodality imaging research. *Medical Physics*, 37(9):4902–4915, 2010.

[50] F. Steinbruecker, T. Pock, and D. Cremers. Large displacement optical flow computation without warping. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1609–1614. IEEE, 2009.

[51] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[52] J Weickert and C Schnörr. A theoretical framework for convex regularization in PDE-based computation of image motion. *Intl. J. Computer Vision*, 45(3):245–264, 2001.

[53] Y. Xu and W. Yin. A Block Coordinate Descent Method for Regularized Multiconvex Optimization with Applications to Nonnegative Tensor Factorization and Completion. *SIAM Journal on Imaging Sciences*, 6(3):1758–1789, 2013.

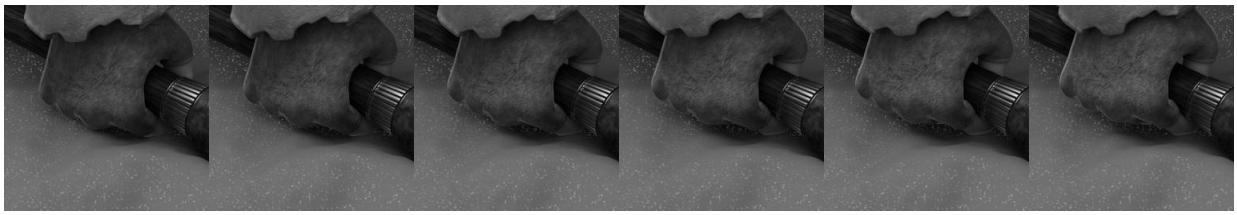
[54] Y. Xu and W. Yin. A Globally Convergent Algorithm for Nonconvex Optimization Based on Block Coordinate Update. *Journal of Scientific Computing*, 72(2):700–734, 2017.

[55] G. L. Zeng. *Medical Image Reconstruction*, chapter 10, pages 175–192. 2010.

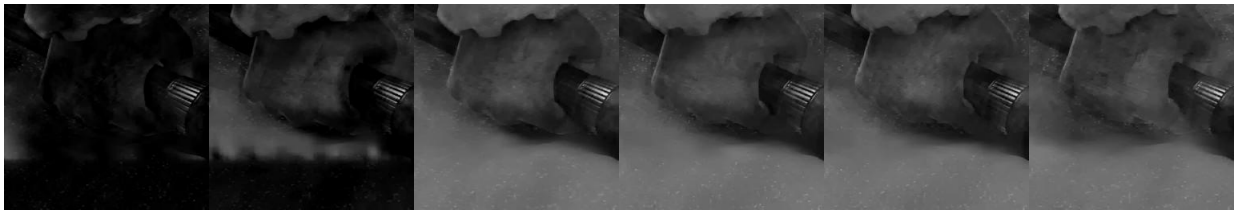
Appendix A

Full results

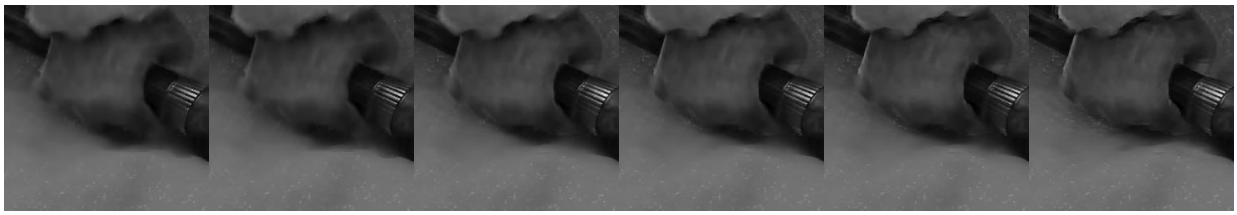
Result of reconstruction for *Hand - original* sequence



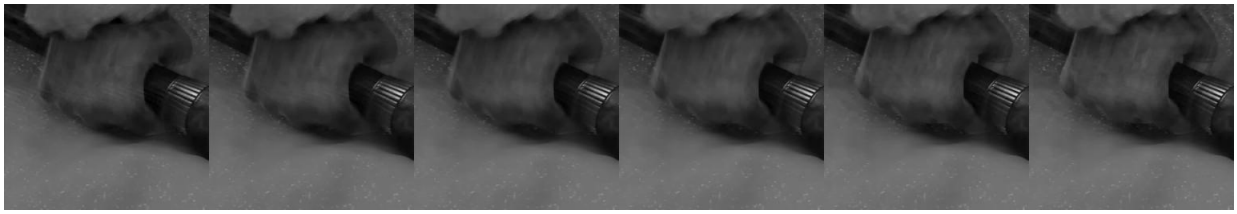
(a) Ground truth image sequence



(b) Result for frame-by-frame reconstruction



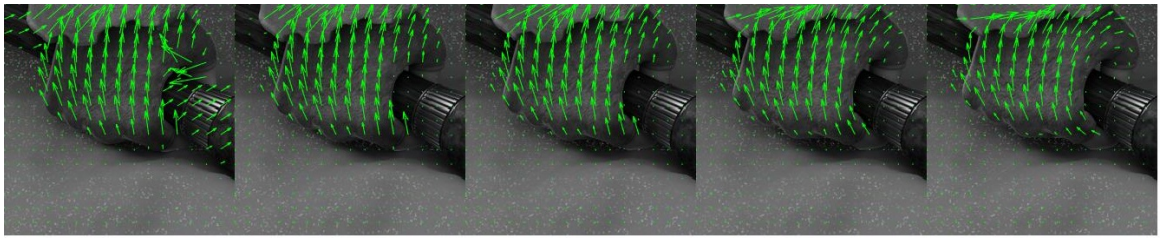
(c) Result for joint reconstruction and motion estimation via Alternating PDHG



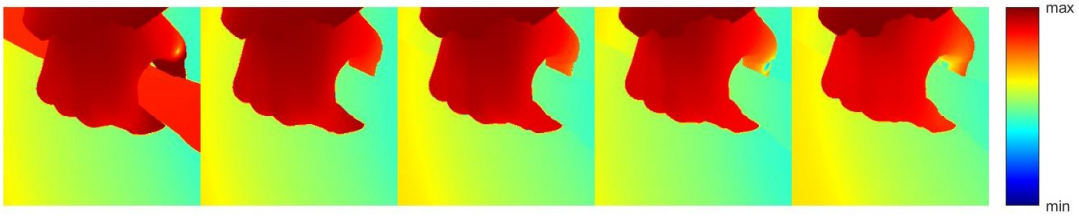
(d) Result reconstruction for joint reconstruction and motion estimation via PALM

Figure A.1

Result of motion estimation for *Hand - orginal* sequence

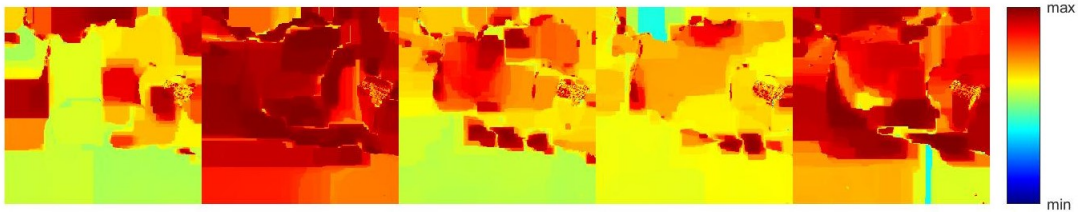


(a) Ground truth for flow field direction

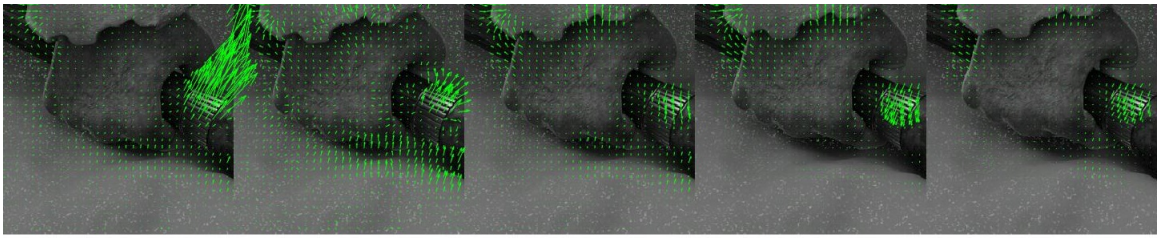


(b) Ground truth flow field for magnitude

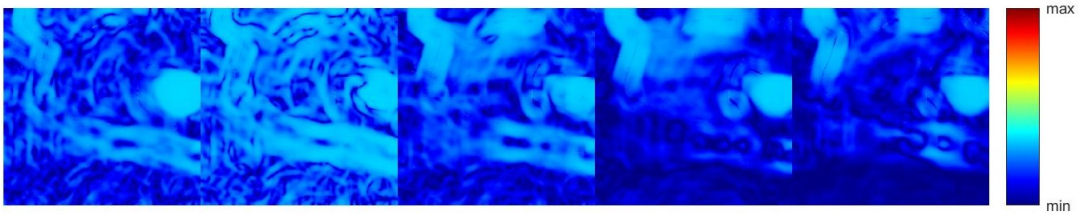
(c) Result direction motion estimation for joint reconstruction and motion estimation via Alternating PDHG



(d) Result magnitude motion estimation for joint reconstruction and motion estimation via Alternating PDHG



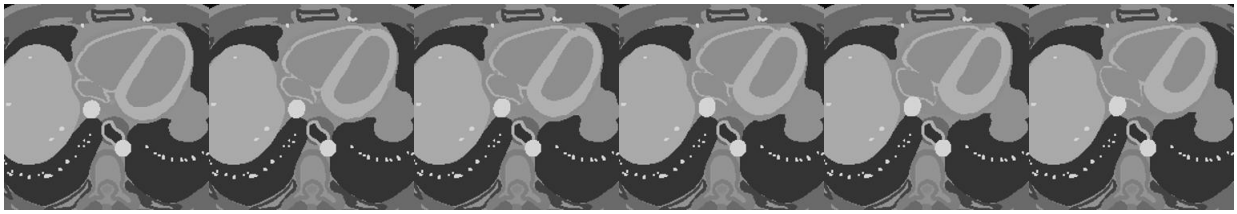
(e) Result direction motion estimation for joint reconstruction and motion estimation via PALM



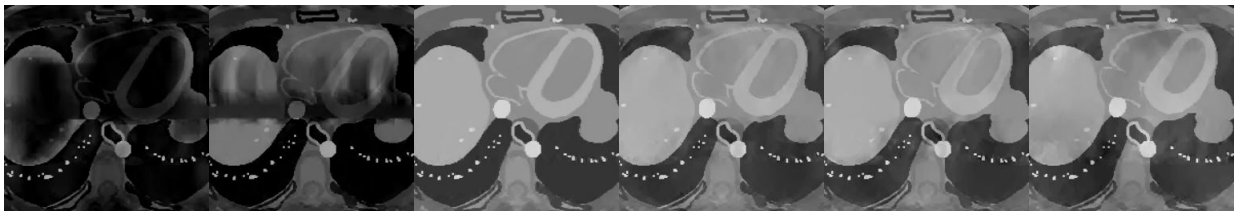
(f) Result magnitude motion estimation for joint reconstruction and motion estimation via PALM

Figure A.2

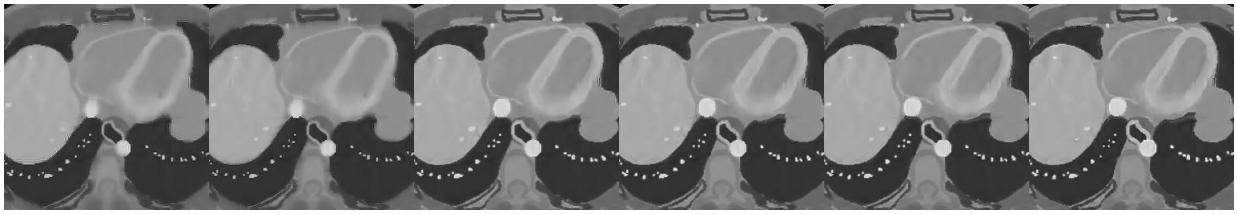
Result of reconstruction for XCAT 1 sequence



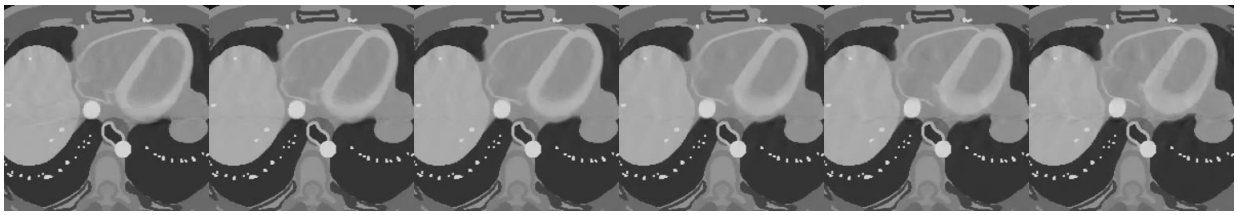
(a) Ground truth image sequence



(b) Result for frame-by-frame reconstruction



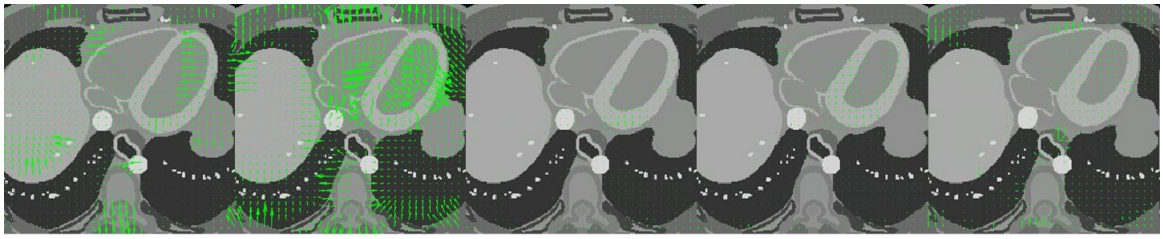
(c) Result for joint reconstruction and motion estimation via Alternating PDHG



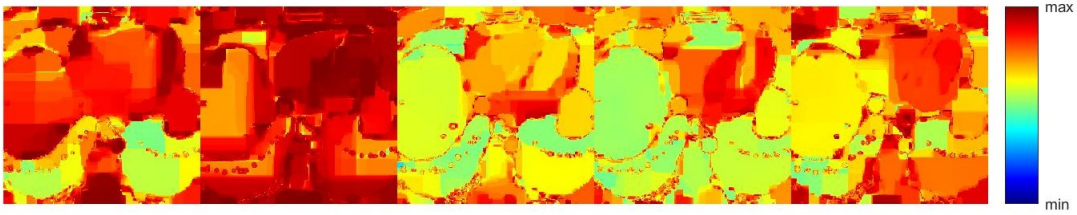
(d) Result reconstruction for joint reconstruction and motion estimation via PALM

Figure A.3

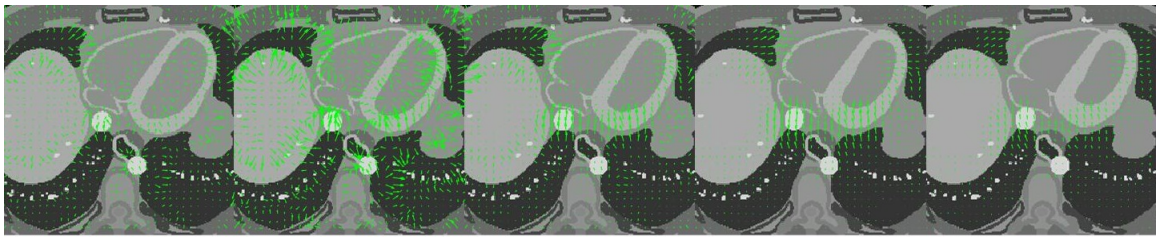
Result of motion estimation for XCAT 1 sequence



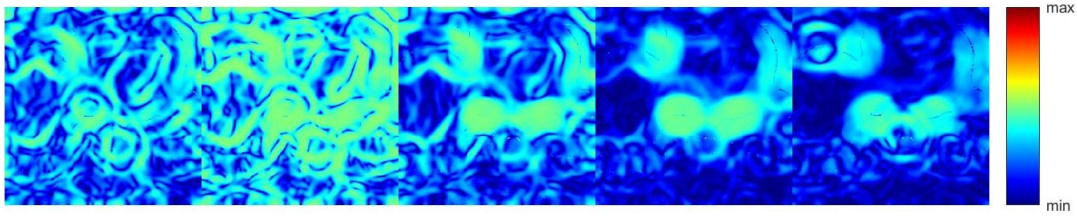
(a) Result direction motion estimation for joint reconstruction and motion estimation via Alternating PDHG



(b) Result magnitude motion estimation for joint reconstruction and motion estimation via Alternating PDHG



(c) Result direction motion estimation for joint reconstruction and motion estimation via PALM



(d) Result magnitude motion estimation for joint reconstruction and motion estimation via PALM

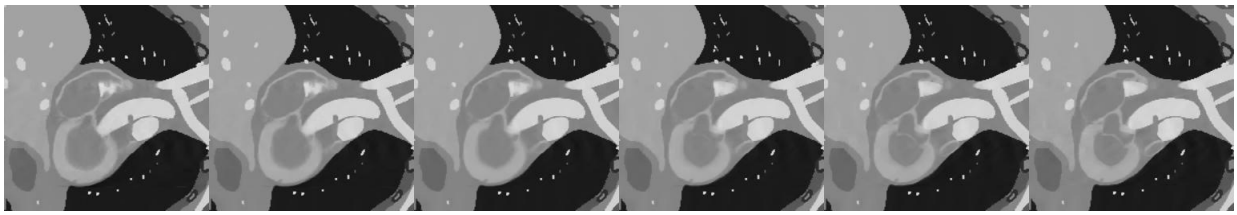
Figure A.4

Results for reconstruction of the XCAT 2 sequence

(a) Ground truth image sequence

(b) Result for only reconstruction

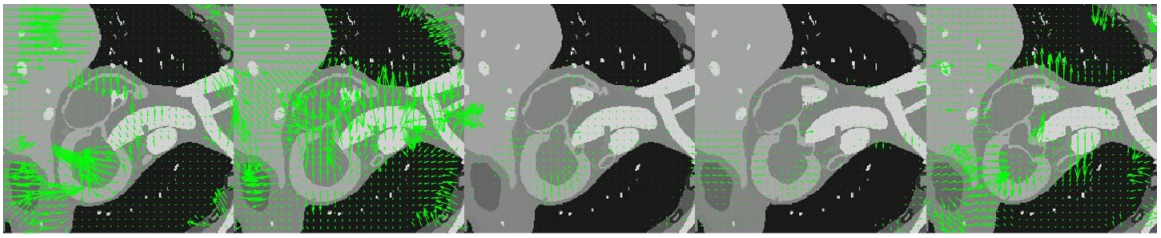
(c) Result for joint reconstruction and motion estimation via Alternating PDHG



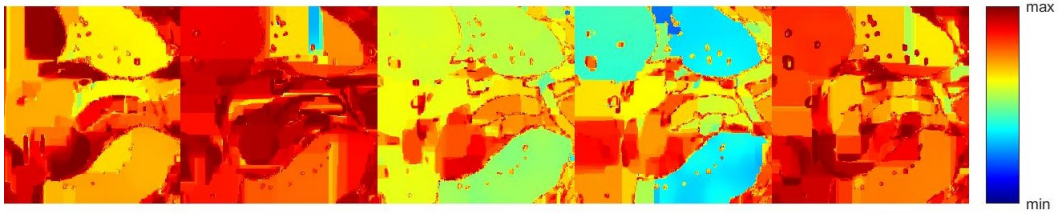
(d) Result reconstruction for joint reconstruction and motion estimation via PALM

Figure A.5: Result for 'XCAT 2' sequence

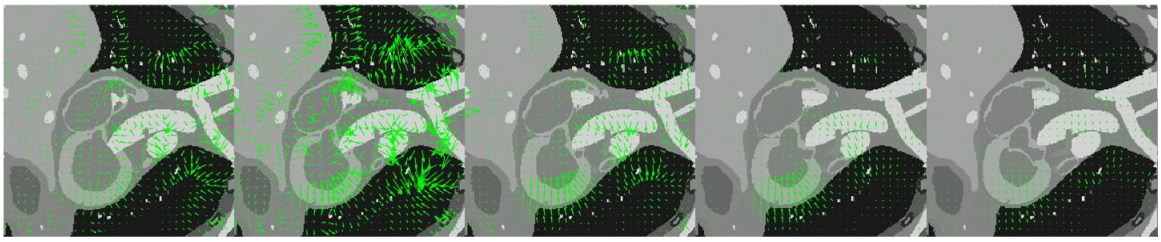
Result of motion estimation for XCAT 2 sequence



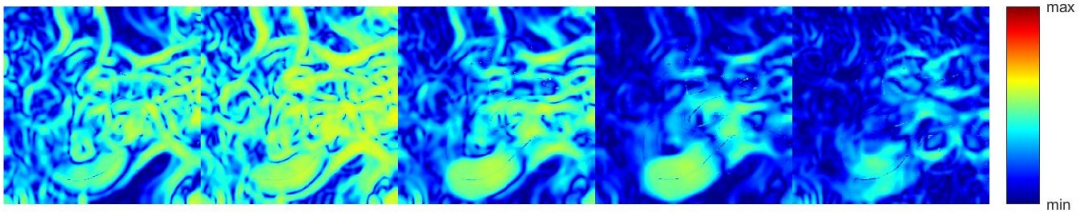
(a) Result direction motion estimation for joint reconstruction and motion estimation via Alternating PDHG



(b) Result magnitude motion estimation for joint reconstruction and motion estimation via Alternating PDHG



(c) Result direction motion estimation for joint reconstruction and motion estimation via PALM



(d) Result magnitude motion estimation for joint reconstruction and motion estimation via PALM

Figure A.6

Reconstruction of experimental data for $R = 6$

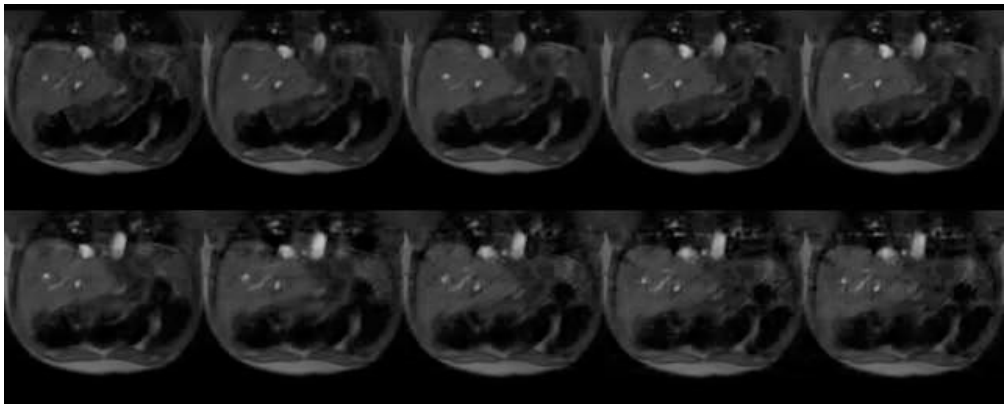


Figure A.7

Reconstruction of experimental data for $R = 8$

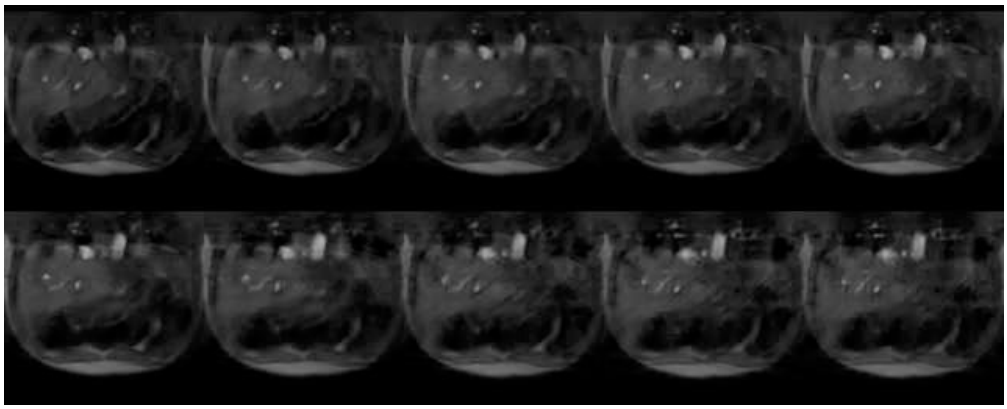


Figure A.8

Reconstruction of experimental data for $R = 10$

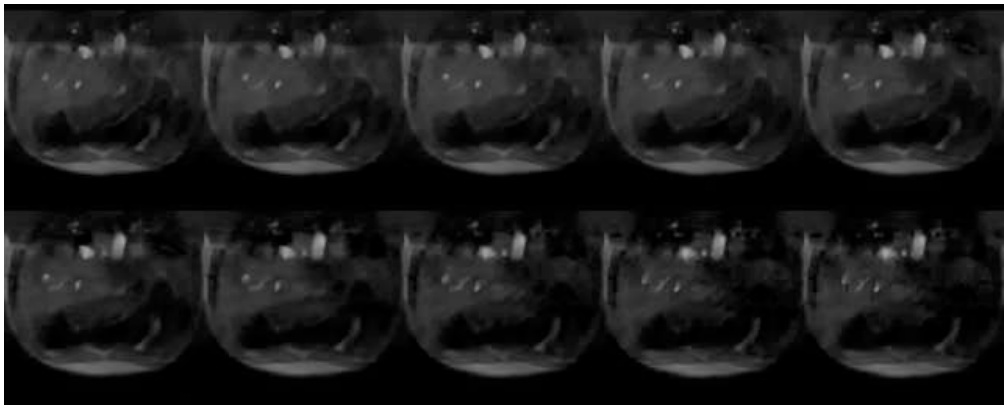


Figure A.9

Reconstruction of experimental data for $R = 12$

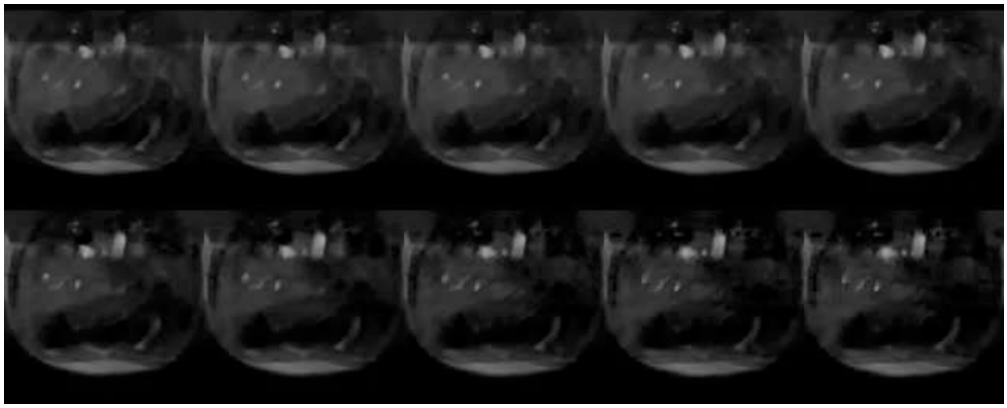


Figure A.10

Result direction motion estimation of experimental data for $R = 8$

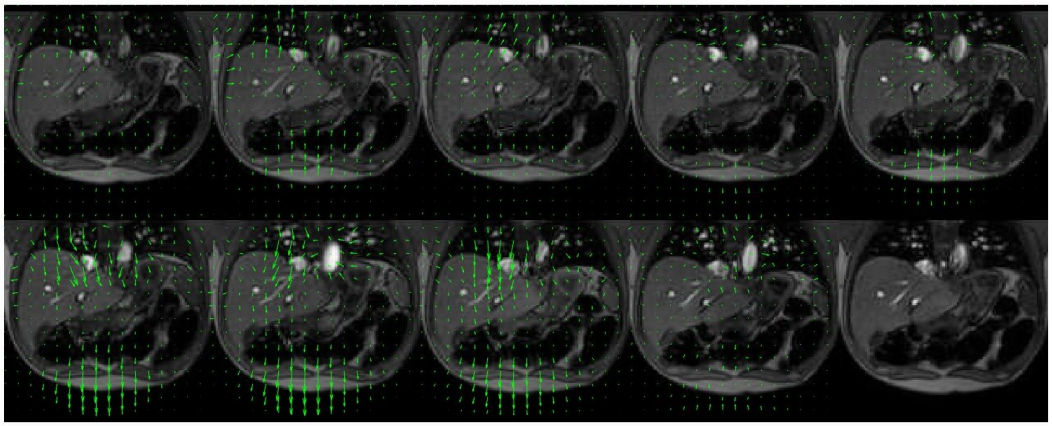


Figure A.11

Result magnitude motion estimation of experimental data for $R = 8$

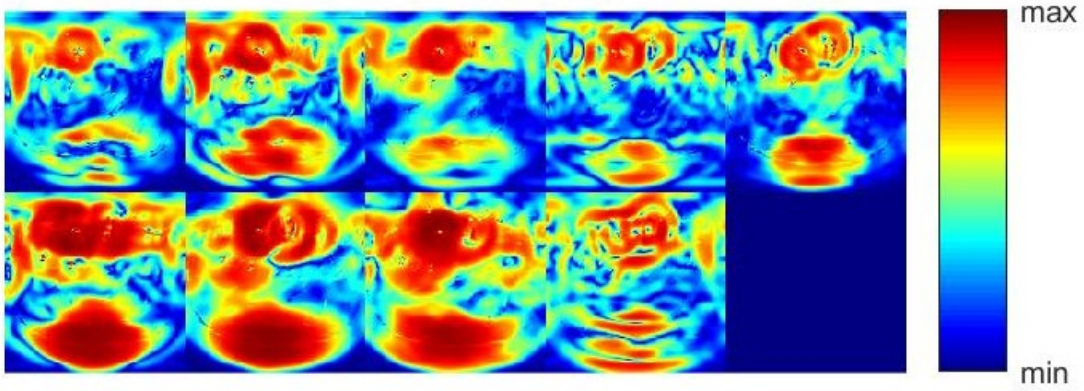


Figure A.12

Appendix B

Additional definitions and derivations

B.1 Forward difference, central difference and backward difference

Forward differences

$$\partial_t^+ u_t = \frac{u_{i,j,t+1} - u_{i,j,t}}{\delta_t},$$

$$\nabla_{x_1}^+ u_{i,j,t} = \begin{cases} \frac{u_{i+1,j,t} - u_{i,j,t}}{\delta_x} & \text{if } i > 0 \text{ and } i < n_x \\ 0 & \text{if } i = 0 \text{ or } i = n_x \end{cases},$$

$$\nabla_{x_2}^+ u_{i,j,t} = \begin{cases} \frac{u_{i,j+1,t} - u_{i,j,t}}{\delta_y} & \text{if } i > 0 \text{ and } i < n_y \\ 0 & \text{if } i = 0 \text{ or } i = n_y \end{cases},$$

Central differences

$$\nabla_{x_1}^{\pm} u_{i,j,t} = \begin{cases} \frac{u_{i+1,jt,} - u_{i-1,j,t}}{2\delta_x} & \text{if } i > 0 \text{ and } i < n_x \\ 0 & \text{if } i = 0 \text{ or } i = n_x \end{cases},$$

$$\nabla_{x_2}^+ u_{i,j,t} = \begin{cases} \frac{u_{i,j+1,t} - u_{i,j-1,t}}{2\delta_y} & \text{if } i > 0 \text{ and } i < n_y \\ 0 & \text{if } i = 0 \text{ or } i = n_y \end{cases}.$$

Note that central differences is self-adjoint. The adjoint of the forward differences are the backward differences. Hence for the divergence of $\mathbf{p}_{i,j,t} = \begin{bmatrix} p_{x_{i,j,t}} & p_{y_{i,j,t}} \end{bmatrix}$ we have the following discretization:

Backward differences

$$\begin{aligned} \operatorname{div}^- \left(\mathbf{p}_{i,j,t} \right) = & \begin{cases} \frac{p_{x_{i,j,t}} - p_{x_{i-1,j,t}}}{\delta_{x_1}} & \text{if } i > 0 \text{ and } i < n_x \\ \frac{p_{x_{i,j,t}}}{\delta_x} & \text{if } i = 0 \\ -\frac{p_{x_{i-1,j,t}}}{\delta_x} & \text{if } i = n_x \end{cases} \\ & + \begin{cases} \frac{p_{y_{i,j,t}} - p_{y_{i,j-1,t}}}{\delta_y} & \text{if } j > 0 \text{ and } j < n_y \\ \frac{p_{y_{i,j,t}}}{\delta_y} & \text{if } j = 0 \\ -\frac{p_{y_{i,j-1,t}}}{\delta_y} & \text{if } j = n_y \end{cases}. \end{aligned}$$

Remark: We will set $\delta_t = \delta_x = \delta_y = 1$.

B.2 Convex conjugates

B.2.1 Useful identities

From the definition of the Convex Conjugate of proper functional $J : U \to \mathbb{R}$:

$$J^*(p) := \sup_{u \in U} \{\langle p, u \rangle_U - J(u)\} \quad \text{for } p \in U^*$$

we can derive:

$$(J(\cdot + a))^* = J^*(\cdot) - \langle \cdot, a \rangle, \tag{B.1}$$

$$(\lambda J(\cdot))^* = \lambda J^*\left(\frac{\cdot}{\lambda}\right), \tag{B.2}$$

for $\lambda > 0$.

B.2.2 Affine Linear L^2 norm

The Convex conjugate of $\frac{\lambda}{2}\|Ku - f\|_2^2$ is $\frac{1}{2\lambda}\|p\|_2^2 + \langle p, f \rangle$.

Proof Using Example 2.2.23 of [18] gives that for $J(u) = \frac{\lambda}{2}\|Ku\|_2^2$ the convex conjugate is $J^*(p) = \frac{1}{2\lambda}\|p\|_2^2$. Combining this result with (B.1) we get:

$$\begin{aligned}\left(\frac{\lambda}{2}\|Ku - f\|_2^2\right)^* &= \frac{1}{2\lambda}\|p\|_2^2 - \langle p, -f \rangle \\ &= \frac{1}{2\lambda}\|p\|_2^2 + \langle p, f \rangle.\end{aligned}$$

B.2.3 L^1 norm

For $J(x) = \alpha\|x\|_1$ the convex conjugate is $J^*(p) = \alpha\delta_{B(L^\infty)}(p/\alpha)$.

Proof Using Example 2.2.4 of [18] we find that for general norm $J(x) = \|x\|_\chi$ defined on measurable space χ the convex conjugate $J^*(x^*) = \delta_{B(\chi^*)}$. Were $\delta_{B(\chi^*)}$ denotes the indicator function of the unit ball in χ^* defined as:

$$\delta_{B(\chi^*)}(x^*) = \begin{cases} 0 & \text{if } x^* \in B(\chi^*) \\ \infty & \text{else.} \end{cases}$$

Hence we need the dual space of L^1 which is L^∞. Now employ (B.2), and we obtain:

$$J^*(p) = \alpha\delta_{B(L^\infty)}(p/\alpha).$$

B.3 Adjoint operators

B.3.1 Adjoint of Fourier transform

Recall that $F(\cdot) := P(\mathscr{F}(\mathrm{Re}(\cdot)))$. The adjoint operator of the Fourier transform is:

$$\begin{aligned}\langle \mathscr{F}(f), g \rangle &= \int_{-\infty}^{\infty} \mathscr{F}(f)\overline{g(x)}\mathrm{d}x \\ &= \int_{-\infty}^{\infty} \frac{1}{2\sqrt{\pi}} \int_{-\infty}^{\infty} f(t)e^{-ixt}\mathrm{d}t\overline{g(x)}\mathrm{d}x \\ &= \int_{-\infty}^{\infty} f(t)\frac{1}{2\sqrt{\pi}} \int_{-\infty}^{\infty} e^{i\bar{x}t}\overline{g(x)}\mathrm{d}x\mathrm{d}t \\ &= \int_{-\infty}^{\infty} f(t)\frac{1}{2\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-ixt}\overline{g(x)}\mathrm{d}x\mathrm{d}t \\ &= \int_{-\infty}^{\infty} f(t)\overline{\mathscr{F}^{-1}(g(t))}\mathrm{d}t \\ &= \langle f, \mathscr{F}^{-1}(g) \rangle.\end{aligned}$$

hence we have proved that for $f, g : L^2(\mathbb{R}^2) \to \mathbb{R}$ the Fourier transform is a unitary operator, as $\mathscr{F}^* = \mathscr{F}^{-1}$.

The implies:

$$\begin{aligned}\|\mathscr{F}\|^2 &= \langle \mathscr{F}u, \mathscr{F}u \rangle \\ &= \langle u, \mathscr{F}^* \mathscr{F}u \rangle \\ &= \langle u, \mathscr{F}^{-1} \mathscr{F}u \rangle \\ &= \langle u, u \rangle \\ &= \|u\|^2.\end{aligned}$$

hence $\|\mathscr{F}\| = 1$.

B.3.2 Adjoint of gradient operator

For $u, p : L^2(\Omega) \to \mathbb{R}$, with u of compact support:

$$\begin{aligned}\int_\Omega \nabla u \cdot p \mathrm{d}x &= \int_\Omega \nabla \cdot (uv) \mathrm{d}x - \int_\Omega u \nabla \cdot p \mathrm{d}x \\ &= \int_{\partial\Omega} u \cdot p \cdot n \mathrm{d}\Gamma - \int_\Omega u \nabla \cdot p \mathrm{d}x \\ &= -\int_\Omega u \nabla \cdot p \mathrm{d}x,\end{aligned}$$

using partial integration and the Divergence theorem, the adjoint operator of $\nabla(\cdot) = -\mathrm{div}(\cdot)$.

B.3.3 Adjoint of wavelet transform

Finally we must determine the adjoint operator for the Wavelet transform $\Phi(\cdot)$.

We used the Daubechies wavelet, which is consist of an orthonormal basis and hence is also a unitary operator. So the adjoint operator is the inverse of the Daubechies wavelet transform. See [17] for more details.

B.4 Proximal maps

Recall the definition of the proximal map:

$$\begin{aligned}x &= (I + \tau \partial T)^{-1}(y), \\ &= \mathrm{prox}_\tau^T(y), \\ &= \arg\min_x \left\{ \frac{\|x - y\|^2}{2} + \tau T(x) \right\}.\end{aligned}$$

B.4.1 Convex conjugate of affine L^2 norm

We have to determine the proximal map of the following functional:

$$F = \frac{1}{2\lambda} \|p\|_2^2 + \langle p, f \rangle,$$

then we must solve:

$$\mathrm{prox}_\sigma^F = \arg\min_p \left\{ \frac{\|p - \hat{p}\|_2^2}{2} + \sigma \left(\frac{\|p\|_2^2}{2\lambda} + \langle p, f \rangle \right) \right\},$$

simplify the norm and inner product,

$$= \arg\min_p \left\{ \frac{\|p - (\hat{p} - \sigma f)\|_2^2}{2} + \frac{\sigma \|p\|_2^2}{2\lambda} \right\}.$$

Using the definition of the Gâteaux derivative the mimimum must sastify:

$$p - (\hat{p} - \sigma f) + \frac{\sigma}{\lambda} p = 0$$
$$p = \frac{\lambda(\hat{p} - \sigma f)}{\lambda + \sigma}.$$

B.4.2 Convex conjugate of L^1 norm

We have to determine the proximal map of:

$$G = \alpha \delta_{B(L^\infty)}\left(\frac{p}{\alpha}\right) = \begin{cases} 0 & \text{if } p \in B(L^\infty) \\ \infty & \text{else.} \end{cases}$$

then we must solve:

$$\text{prox}_\sigma^G = \arg\min_p \left\{ \frac{\|p - \hat{p}\|_2^2}{2} + \sigma\alpha\delta_{B(\chi^*)}\left(\frac{p}{\alpha}\right) \right\}$$

If $\left|\frac{p}{\alpha}\right|$ is greater than α, then $\delta_{B(L^\infty)}\left(\frac{p}{\alpha}\right) = \infty$, else $\delta_{B(L^\infty)}\left(\frac{p}{\alpha}\right) = 0$. So we find that:

$$p = \frac{\hat{p}}{\max(\alpha, |\hat{p}|)}$$

is the minimum. In case we have to determine the proximal map of:

$$G = \alpha \delta_{B(L^\infty)}\left(\frac{p}{\alpha}\right) - \langle p, f \rangle,$$

we find:

$$p = \frac{\hat{p} + \sigma f}{\max(\alpha, |\hat{p} + \sigma f)|}$$

since we can write:

$$\begin{aligned} \text{prox}_\sigma^G &= \arg\min_p \left\{ \frac{\|p - \hat{p}\|_2^2}{2} + \sigma\delta_{B(L^\infty)}\left(\frac{p}{\alpha}\right) - \sigma\langle p, f\rangle \right\} \\ &= \arg\min_p \left\{ \frac{\|p - (\hat{p} + \sigma f)\|_2^2}{2} + \sigma\delta_{B(L^\infty)}\left(\frac{p}{\alpha}\right) \right\}. \end{aligned}$$

B.4.3 χ_+

Finally we need to determine the proximal map of:

$$H = \chi_+(u_{i,j,t}) := \begin{cases} 0 & \text{if } u_{i,j,t} \geq 0 \quad \forall i,j,t \\ \infty & \text{else,} \end{cases}$$

so we must minimize:

$$\text{prox}_\sigma^H = \arg\min_u \left\{ \frac{\|u - \hat{u}\|_2^2}{2} + \sigma\chi_+(u) \right\}.$$

If $\hat{u} \geq 0$, then $u = \hat{u}$ will give: $\text{prox}_\sigma^H = 0$. If $\hat{u} < 0$ then $\chi_+(u) = \infty$, so then we choose $u = 0$ and accept the difference $\|u - \hat{u}\|_2^2$. Hence we find:

$$u = \max(0, \hat{u}).$$