

# Registration and Analysis of Dynamic Magnetic Resonance Image Series

Inauguraldissertation

zur  
Erlangung der Würde eines

*Dr. sc. med.*

vorgelegt der Medizinischen Fakultät der Universität Basel

von  
Robin Sandkühler

aus  
Gelsenkirchen  
Deutschland

Basel, 2020

Originaldokument gespeichert auf dem Dokumentenserver der Universität Basel  
[edoc.unibas.ch](https://edoc.unibas.ch)

Genehmigt von der Medizinischen Fakultät auf Antrag von

Prof. Dr. Philippe C. Cattin, Universität Basel  
*Dissertationsbetreuer*

Prof. Dr. Philipp Latzin, PhD, Inselspital, Universitätsspital Bern  
*Korreferent*

Prof. Dr. Ivana Išgum, Amsterdam University Medical Center  
*Externe Gutachterin*

Dr. Christoph Jud, Universität Basel  
*Zusätzlicher Betreuer*

Basel, den

Prof. Dr. Primo Schär  
*Dekan*

Imagination is more important than knowledge.  
Knowledge is limited. Imagination encircles the world.

Albert Einstein





# Contents

<i>Acknowledgements</i>	<b>vii</b>
<i>Summary</i>	<b>ix</b>
<i>Zusammenfassung</i>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Medical Background</b>	<b>5</b>
2.1 Cystic Fibrosis . . . . .	5
2.2 Medical Imaging . . . . .	5
2.2.1 Magnetic Resonance Imaging of the Lungs . . . . .	6
2.2.2 MR Ventilation and Perfusion of the Lungs . . . . .	6
2.2.3 Medical Image Representation . . . . .	8
<b>3 Medical Image Registration</b>	<b>11</b>
3.1 Non-Parametric Registration . . . . .	11
3.2 Parametric Registration . . . . .	13
3.2.1 Transformation Models . . . . .	14
3.3 Diffeomorphic Registration . . . . .	15
3.4 Demons Registration . . . . .	16
3.4.1 Diffeomorphic Demons . . . . .	17
3.5 Learning-based Image Registration . . . . .	18
3.6 Image Similarity Measures . . . . .	20
3.7 Regularizer . . . . .	23
3.8 Optimizer . . . . .	24
3.8.1 Gradient-Free Optimization . . . . .	25
3.9 Sliding-Organ Motion Registration . . . . .	25
<b>4 Machine Learning</b>	<b>27</b>
4.1 Artificial Neural Networks . . . . .	28
4.1.1 Multilayer Perceptron . . . . .	29
4.2 Convolutional Neural Networks . . . . .	33
4.3 Recurrent Neural Networks . . . . .	36
4.3.1 Long Short-Term Memory Networks . . . . .	37
4.3.2 Gated Recurrent Units . . . . .	38
4.4 Deep Neural Networks . . . . .	39
4.4.1 Deep Neural Networks for Semantic Segmentation . . . . .	41
<b>5 Adaptive Graph Diffusion Regularisation for Discontinuity Preserving Image Registration</b>	<b>43</b>
5.1 Introduction . . . . .	45

## Contents

5.2	Background . . . . .	46
5.3	Method . . . . .	47
5.4	Results . . . . .	51
5.5	Conclusion . . . . .	53
<b>6</b>	<b>Recurrent Registration Neural Networks for Deformable Image Registration</b>	<b>55</b>
6.1	Introduction . . . . .	57
6.2	Background . . . . .	59
6.3	Methods . . . . .	61
6.4	Experiments and Results . . . . .	64
6.5	Conclusion . . . . .	67
<b>7</b>	<b>AIRLab: Autograd Image Registration Laboratory</b>	<b>69</b>
7.1	Introduction . . . . .	71
7.2	Background . . . . .	73
7.3	Autograd Image Registration Laboratory . . . . .	75
7.4	Conclusion . . . . .	86
<b>8</b>	<b>Gated Recurrent Neural Networks for Accelerated Ventilation MRI</b>	<b>89</b>
8.1	Introduction . . . . .	91
8.2	Background . . . . .	92
8.3	Method . . . . .	94
8.4	Experiments and Results . . . . .	96
8.5	Conclusion . . . . .	98
<b>9</b>	<b>Weakly Supervised Learning Strategy for Lung Defect Segmentation</b>	<b>99</b>
9.1	Introduction . . . . .	101
9.2	Method . . . . .	102
9.3	Experiments and Results . . . . .	105
9.4	Conclusion . . . . .	107
<b>10</b>	<b>Discussion and Conclusion</b>	<b>109</b>
	<b>Bibliography</b>	<b>113</b>
	<i>Publications</i>	<b>129</b>

# Acknowledgements

I like to thank my supervisor Prof. Dr. Philippe C. Cattin for the opportunity to work with him on the field of medical image registration. I am very grateful for his inspiring mindset and his guidance during this time of my life. Furthermore, I thank Prof. Dr. Philipp Latzin for his support and his medical feedback during this work. A special thanks go to Christoph Jud for the fruitful and interesting discussion at the blackboard. In addition, I would like to thank Simon Andermatt for sharing his experience and enthusiasm in the field of deep learning. My work would not have been possible without images. Therefore I thank Prof. Dr. Oliver Bieri, Grzegorz Bauman, and Orso Pusterela for their work on the MR sequence development and their support regarding any questions about MR physics. I also would like to thank Sylvia Nyilas and Corin Willers for their engagement in acquiring these images at the hospital. Finally, I would like to thank my family and friends for their support and encouragement during this time.

I want to thank all current and past members of the Center for medical Image Analysis and Navigation (CIAN) (in alphabetical order): Simon Andermatt, Florentin Bieder, Philippe C. Cattin, Natalia Chicherova, Corinne Eymann-Baier, Balazs Faludi, Beat Fasel, Sara Freund, Ketut Fundana, Alina Giger, Antal Horváth, Lorenzo Iafolla, Christoph Jud, Samaneh Manavi, Nadia Möri, Peter von Niederhäusern, Simon Pezold, Tiziano Ronchetti, Adrian Schneider, Sebastian Scherer, Eva Schnider, Carlo Seppi, Bruno Sempéré, Aydin Ürgen, Jörg Wagner, Reinhard Wendler, Lilian Witthauer, Julia Wolleb, Stephan Wyder, Marek Zelechowski.



# Summary

Cystic fibrosis (CF) is an autosomal-recessive inherited metabolic disorder that affects all organs in the human body. Patients affected with CF suffer particularly from chronic inflammation and obstruction of the airways. Through early detection, continuous monitoring methods, and new treatments, the life expectancy of patients with CF has been increased drastically in the last decades. However, continuous monitoring of the disease progression is essential for a successful treatment. The current state-of-the-art method for lung disease detection and monitoring is computed tomography (CT) or X-ray. These techniques are ill suited for the monitoring of disease progressions because of the ionizing radiation the patient is exposed during the examination. Through the development of new magnetic resonance imaging (MRI) sequences and evaluation methods, MRI is able to measure physiological changes in the lungs. The process to create physiological maps, i.e., ventilation and perfusion maps, of the lungs using MRI can be split up into three parts: *MR-acquisition*, *image registration*, and *image analysis*.

In this work, we present different methods for the image registration part and the image analysis part. We developed a graph-based registration method for 2D dynamic MR image series of the lungs in order to overcome the problem of sliding motion at organ boundaries. Furthermore, we developed a human-inspired learning-based registration method. Here, the registration is defined as a sequence of local transformations. The sequence-based approach combines the advantage of dense transformation models, i.e., large space of transformations, and the advantage of interpolating transformation models, i.e., smooth local transformations.

We also developed a general registration framework called Autograd Image Registration Laboratory (AIRLab), which performs automatic calculation of the gradients for the registration process. This allows rapid prototyping and an easy implementation of existing registration algorithms.

For the image analysis part, we developed a deep-learning approach based on gated recurrent units which is able to calculate ventilation maps with less than a third of the number of images of the current method. Automatic defect detection in the estimated MRI ventilation and perfusion maps is essential for the clinical routine to automatically evaluate the treatment progression. We developed a weakly supervised method that is able to infer a pixel-wise defect segmentation by using only continuous global label during training. In this case, we directly use the lung clearance index (LCI) as a global weak label, without any further manual annotations. The LCI is a global measure to describe ventilation inhomogeneities of the lungs and is obtained by a multiple breath washout test.



# Zusammenfassung

Zystische Fibrose (ZF) ist eine autosomal-rezessiv vererbte Stoffwechselerkrankung. Patienten mit dieser Erkrankung leiden besonders unter einer chronischen Entzündung und Blockierung der Atemwege. Durch eine frühzeitige Erkennung und neue Behandlungsmethoden ist die Lebenserwartung von Patienten mit ZF in den letzten Jahrzehnten drastisch gestiegen. Eine kontinuierliche Überwachung des Krankheitsverlaufes und der entsprechenden Medikation ist aber nach wie vor essenziell um einen guten therapeutischen Erfolg zu erzielen. Häufig eingesetzte Methoden für die Überwachung sind bildbasierte Verfahren auf der Basis von Röntgenstrahlen wie die Computertomographie oder das Thoraxröntgen. Für eine regelmässige Anwendung sind diese Methoden aufgrund der hohen Strahlenbelastung nicht geeignet. Durch die Entwicklung neuer Sequenzen für die Magnetresonanztomographie (MRT) und entsprechende Auswertelgorithmen ist es nun möglich physiologische Änderungen des Lungengewebes mittels MR zu messen. Da eine MR Untersuchung zu keiner Strahlenbelastung des Patienten führt, eignet sich MR besonders um den Verlauf der Erkrankung sowie der Behandlung zu überwachen. Der Prozess mit dem physiologische Karten der Lunge mit MR erzeugt werden können setzt sich im Wesentlichen aus drei Teilen zusammen: *MR-Aufnahme*, *Bildregistrierung* und der *Bildanalyse*.

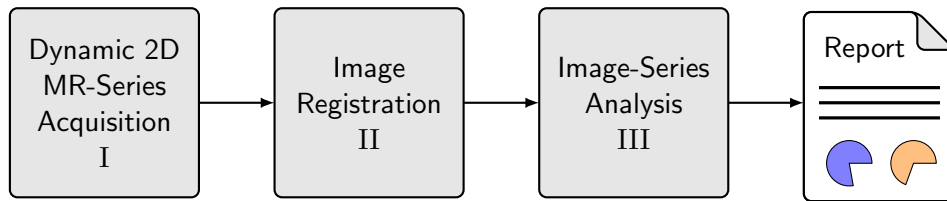
Im Rahmen dieser Arbeit wurden Methoden entwickelt, die in den Bereichen der Bildregistrierung und der finalen Bildanalyse Verwendung finden. Die Bildregistrierung legt die Grundlage für die weitere Verarbeitung der Daten und muss daher besonders genau sein. Wir haben eine Graph-basierte Registrierungsmethode speziell für die Registrierung von dynamischen 2D Bildserien der Lunge entwickelt. Zudem haben wir eine Registrierungsmethode entwickelt die sich daran orientiert wie ein Mensch zwei Bilder registrieren würde. Dabei handelt es sich um eine lernbasierte Methode, bei der die finale Transformation aus einer Sequenz von lokalen Transformation besteht. Basierend auf den Erfahrungen, die wir durch die Entwicklung der Registrierungsmethoden erhalten haben, konnten wir das allgemeine Registrierungsframework AIRLab entwickeln. Der Vorteil dieses Frameworks ist die automatische Bestimmung der Gradienten der Kostenfunktion bezüglich der Transformationsparameter. Es eignet sich besonders für die schnelle Entwicklung neuer Ideen oder das Reproduzieren bestehender Registrierungsmethoden.

Im Bereich der Bildanalyse haben wir eine Methode entwickelt die für die Bestimmung der Ventilationskarten weniger Bilder benötigt als die aktuell verwendete Methode. Dies ermöglicht eine kürzere Untersuchungszeit und damit weniger Stress für den Patienten. Für eine kontinuierliche Überwachung des Krankheitsverlaufes ist eine automatische Auswertung der physiologischen Karten notwendig. Dafür haben wir eine lernbasierte Segmentierungsmethode entwickelt die eine pixelweise Segmentierung der defekten Bereiche in den Ventilationskarten erzeugt. Unsere Methode verwendet zum Lernen dieser Segmentierung nur ein globales schwaches Label und zwar den Lung Clearance Index (LCI).





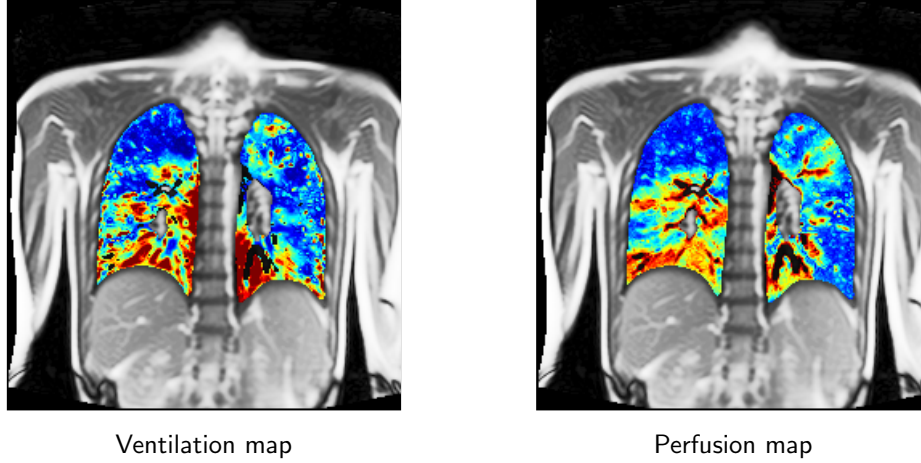
Cystic fibrosis (CF) is an autosomal recessive genetic disorder affecting multiple organs. The most serious impact in people with the severe disease phenotype, is on the respiratory system [Elborn \(2016\)](#); [Elborn et al. \(1991\)](#); [O’Sullivan and Freedman \(2009\)](#). Through early detection mechanisms and new treatments, the life expectancy of patients with CF has been increased by 40 years in the last decades [Elborn \(2016\)](#). The current monitoring system relies on lung function tests, e.g. spirometry, multiple-breath washout, or radiation-based imaging methods like computed tomography (CT) or X-ray. However, lung function tests are not able to show regional changes of the lung function or structure. CT and X-ray are problematic for monitoring the disease progression due to a high cumulative radiation dose associated with frequent follow-up examinations especially in children and young adults. Recently, a new magnetic resonance imag-



**Figure 1.1:** Automatic process pipeline for the generation of MR ventilation maps, MR perfusion maps, and report.

ing (MRI) pulse sequence termed ultra-fast steady-state free precession (ufSSFP) has been proposed for the assessment of pulmonary lung function [Bauman and Bieri \(2017\)](#); [Bauman et al. \(2016\)](#). ufSSFP imaging exploits very short echo and repetition times and thus offers high signal intensity in the lung tissue in comparison to contemporary MR pulse sequences. An overview of the process pipeline for the generation of MRI ventilation and perfusion maps is shown in Figure 1.1. The process to obtain the physiological properties of the lungs using MRI contains three major steps (Figure 1.1). In the first step, several dynamic 2D image series at different positions are acquired using the ufSSFP MR sequence in order to cover the entire lung volume. Each single image series contains 140 images. Image series are acquired in *free-breathing* and without echocardiogram-gating. For further analysis of the image data corresponding anatomical structures need to be spatially aligned. The alignment is achieved by performing image registration for each image in the series. Based on the registration result a dedicated spectral analysis method of the time-resolved image series called matrix pencil decomposition is performed [Bauman and Bieri \(2017\)](#). The results of this analysis are ventilation and perfusion maps of the lungs. Based on the ventilation and perfusion map computation a report is automatically generated to aid the disease follow-up process.

## 1 Introduction



**Figure 1.2:** MR Ventilation and perfusion maps of a patient with CF. Computed with the method presented by [Bauman and Bieri \(2017\)](#) based on the ufSSFP MR image sequence.

## Contribution

In this work, we present different methods for the *image registration* (II) and the *image analysis* (III) part of the process pipeline shown in Figure 1.1. The major part of this work was done for the image registration (II).

Registration of the pulmonary structures in medical images is a very complex task due to sliding-organ motion and intensity changes of corresponding structures due to physiological changes. Since the subsequent image analysis methods rely on the results of the image registration, the registration has a direct influence on the accuracy of the generated ventilation and perfusion maps. In order to improve the registration result, we developed a graph-based registration method to overcome the problem of sliding-organ motion with a high registration accuracy at the sliding-organ boundaries. Deep learning methods have become state-of-the-art methods for the task of image segmentation and image classification. In the past years, learning-based registration methods have shown good performance, especially in terms of computation time. However, current learning-based registration methods are built on feed forward neural networks where the transformation is directly estimated. Our presented approach is designed as a sequence-based model using a recurrent neural network at its core. In each iteration, a local transformation is estimated by the network until both images are properly aligned. The sequence-based approach combines the advantage of dense transformation models, i.e., large space of transformations, and the advantage of interpolating transformation models, i.e., smooth local transformations. Furthermore, this allows an additional dimension for the regularization, i.e., the number of local transformation used to spatially align both images.

Finally, we propose a general registration framework the Autograd Image Registration Laboratory (AIRLab) to speed up the development of new registration methods

and the reproducibility of current methods. In AIRLab, the gradients are calculated automatically which is a key feature to speed up the development process.

For the image analysis part (III), we developed a method able to calculate ventilation maps by using fewer images than the current state-of-the-art method while keeping the same quality of the functional maps. This allows a reduction in the scanning time which decreases overall examination time. Automatic defect detection of functional lung impairment in the generated ventilation and perfusion maps for the generated ventilation and perfusion map is essential for the clinical routine. However, obtaining pixel-wise labeled ground-truth data is difficult as it is a very time-consuming process. We developed a weakly supervised defect segmentation method for ventilation maps without any human annotations. For this, we used the lung clearance index (LCI) as a global weak label. The LCI is derived from clinically well established multi-breath washout examinations which is a common biomarker for the detection of ventilation inhomogeneities of the lungs.

## Outline

The medical background, the technical details of the new MR image sequence, and the method for the calculation of the ventilation and perfusion maps is described in Chapter 2. A description of the mathematical background on medical image registration and the different classes of registration methods is given in Chapter 3. Chapter 4 contains the background of the used machine learning method in this work. The adaptive graph diffusion registration method is described in Chapter 5 and the learning-based registration method using recurrent neural networks is presented in Chapter 6. In Chapter 7, we described the developed registration framework AIRLab. The accelerated method for the generation of ventilation maps is presented in Chapter 8 and the weakly supervised defect segmentation method is described in Chapter 9. Finally, in Chapter 10, we discussed our work and described possible future work for further improvements.



# Medical Background

According to the World Health Organization (WHO) chronic lung disease like chronic obstructive pulmonary disease (COPD), or lower respiratory infections are predicted to be in the top five of leading causes of mortality in the world by 2030 [Mathers and Loncar \(2006\)](#).

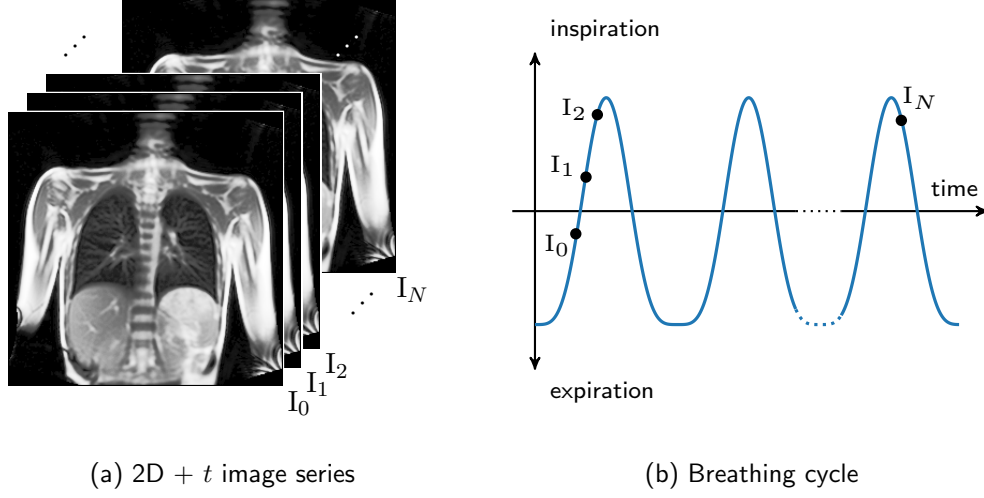
## 2.1 Cystic Fibrosis

Cystic Fibrosis (CF) is an autosomal recessive genetic disorder [Elborn \(2016\)](#) affecting the cystic fibrosis transmembrane conductance regulator (CFTR) gene. It affects multiple organs and was first diagnosed by Dorothy Anderson in 1938 and described in [Andersen \(1938\)](#). The most serious impact in people with the severe disease phenotype, is on the respiratory system and the increased susceptibility of individuals to colonisation and infection of the airways with Gram-negative bacteria, e.g., *Pseudomonas aeruginosa*, [Elborn \(2016\)](#); [Elborn et al. \(1991\)](#); [O’Sullivan and Freedman \(2009\)](#). Through the development of new treatment methods and early detection mechanisms, the life expectancy for patients diagnosed with CF is about 40 years [Elborn \(2016\)](#). Current monitoring systems for the disease progression and the evaluation of the treatment are pulmonary function test, e.g., the multiple breath nitrogen washout test (N<sub>2</sub>-MBW) [Robinson et al. \(2009\)](#) for the estimation of the lung clearance index (LCI), or ionizing radiation-based imaging methods like computed tomography (CT) or X-ray [Ley \(2015\)](#). However, ionizing radiation-based imaging methods like CT or X-ray are ill suited for continues monitoring over a long period of time or for pediatric applications.

## 2.2 Medical Imaging

Medical imaging is an essential part of the modern clinical routine. There are several different imaging modalities used in the clinic today. The development of medical imaging started with the discovery of the X-rays by Wilhelm Röntgen on November 8, 1895 [Berger et al. \(2018\)](#). Followed by the development of computed tomography (CT) [Hounsfield \(1975\)](#). CT uses projections, i.e., X-ray images, from different angles in order to compute a cross-sectional image of the region of interest instead of a single projection obtained by X-ray. With CT it is now possible to obtain a 3D image of different parts of the human body. In the 1970 magnetic resonance imaging (MRI) was presented [Damadian \(1971\)](#); [Lauterbur \(1973\)](#); [Mansfield and Grannell \(1975\)](#). In comparison to CT, MRI does not use ionizing radiation for image generation and is able to generate images with high soft-tissue contrast. MRI exploits the property of spin in different nuclei in order to generate signal, i.e., an image. If the hydrogen nucleus is used as a target, MRI is called proton MRI because the hydrogen nucleus consists of a single proton. Its high abundance in the human body predetermined the usage of the hydrogen nucleus as a target for MRI.

## 2 Medical Background



**Figure 2.1:** (a) Time-resolved 2D image series acquired with the ufSSFP MR sequence, and (b) corresponding position of the image in the breathing cycle.

### 2.2.1 Magnetic Resonance Imaging of the Lungs

MRI has shown very good results for general soft tissue imaging of several anatomical structures, e.g., the brain. However, the application of proton MRI of the lungs is difficult due to the low proton density of the lung parenchyma in comparison to other tissue types, constant breathing, and cardiac motion [Kauczor and Wielpütz \(2018\)](#). In the past, several methods have been presented in order to reduce motion artifacts by using physiological gating mechanism [Biederer et al. \(2002\)](#) or dedicated MR acquisition techniques [Jud et al. \(2018a\)](#).

A method to increase the image contrast inside the lung is the usage of hyperpolarized gases [van Beek et al. \(2004\)](#). Here, the patient inhales the hyperpolarized gas during the image acquisition. However, these methods require specialized equipment for the examination which increases the complexity of the acquisition procedure.

### 2.2.2 MR Ventilation and Perfusion of the Lungs

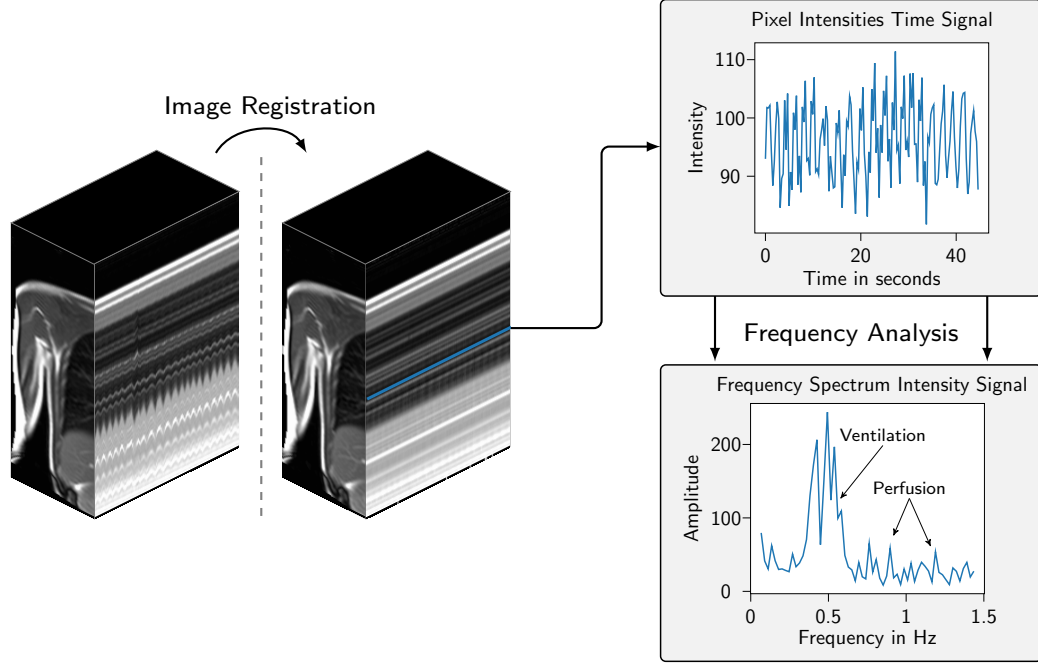
In [Bauman et al. \(2016\)](#) a new MR sequence was presented which is able to measure physiological changes of the lungs without using specialized equipment. The presented sequence is a time-resolved 2D image sequence named ultra-fast steady-state free precession (ufSSFP). In order to measure the physiological properties of the lungs, and in particular the ventilation, the acquisition of the image series is performed in free-breathing as shown in Figure 2.1. The ufSSFP sequence uses Cartesian gradient switching patterns and optimized excitation pulses to accelerated the pulse sequence. Furthermore, partial echo readouts and ramp sampling techniques are used. This allows for remarkably shorter echo time (TE) and repetition time (TR) in comparison to contemporary



**Figure 2.2:** MR images acquired with the ufSSFP sequence at different coronal slice positions from posterior to anterior.

techniques, which improves the signal in the lung parenchyma and reduces imaging artifacts. Here, the artifacts are mostly motion and banding artifacts. In order to cover the complete lungs 8 to 12 coronal slices with 140 2D images per slice position are acquired. An example of MR images from different coronal slice positions is shown in Figure 2.2. The total examination time is about 10 minutes.

Based on the acquired image data it is possible to compute spatial pulmonary ventilation and perfusion maps. Two methods have been developed in the past for the map generation. The first was presented in [Bauman et al. \(2009\)](#) using Fourier decomposition (FD) and the second in [Bauman and Bieri \(2017\)](#) using the matrix pencil (MP) method to calculate the spectral components of the time-resolved signal modulations (intensity signals). Figure 2.3 shows a general overview of the map calculation process. The first step in the processing pipeline is the image registration step. Image registration is needed because the image series is acquired in free-breathing and therefore corresponding structures are not aligned over time. The registration of images of the anterior parts of the thorax is a challenging task due to sliding-organ motion. A detailed description of this registration problem is given in Section 3.9. After the image registration step, corresponding anatomical structures are aligned and the actual map calculation can be performed. Therefore, the time-varying pixel intensity signal at each spatial position is analyzed. For each time signal, a frequency analysis is performed by detecting corresponding frequencies for the ventilation and the perfusion. Figure 2.4 shows an example of the spatial ventilation and perfusion maps for the MP method [Bauman and Bieri \(2017\)](#) for selected coronal slices of the lung of a CF patient.



**Figure 2.3:** Schematic process of MR perfusion and ventilation computation based on a dynamic 2D ufSSFP MR image series.

### 2.2.3 Medical Image Representation

Medical images appear in a wide variety of different formats depending on the imaging method and the application. The major image formats are 2D, 2D+ $t$ , and 3D image data. An example for 2D image data is the image generated by an X-ray system. Medical image time series, i.e., 2D+ $t$ , are obtained for intraoperative imaging with 2D ultrasound or an X-ray C-arc. Furthermore, images series are also used for diagnostics as for example the presented ufSSFP MR sequence. The acquisition of 3D image data is standard in the medical environment and mainly generated by 3D CT or 3D MR systems. Through the development of new methods, 4D imaging methods have become popular in the last decade [Jud et al. \(2018a\)](#); [von Siebenthal et al. \(2007\)](#).

In this work, we will focus on 2D, 2D +  $t$ , and 3D medical image data. Figure 2.5 shows an example of the relevant medical image formats used in this work. For processing and storing of the image, a digital representation is needed.

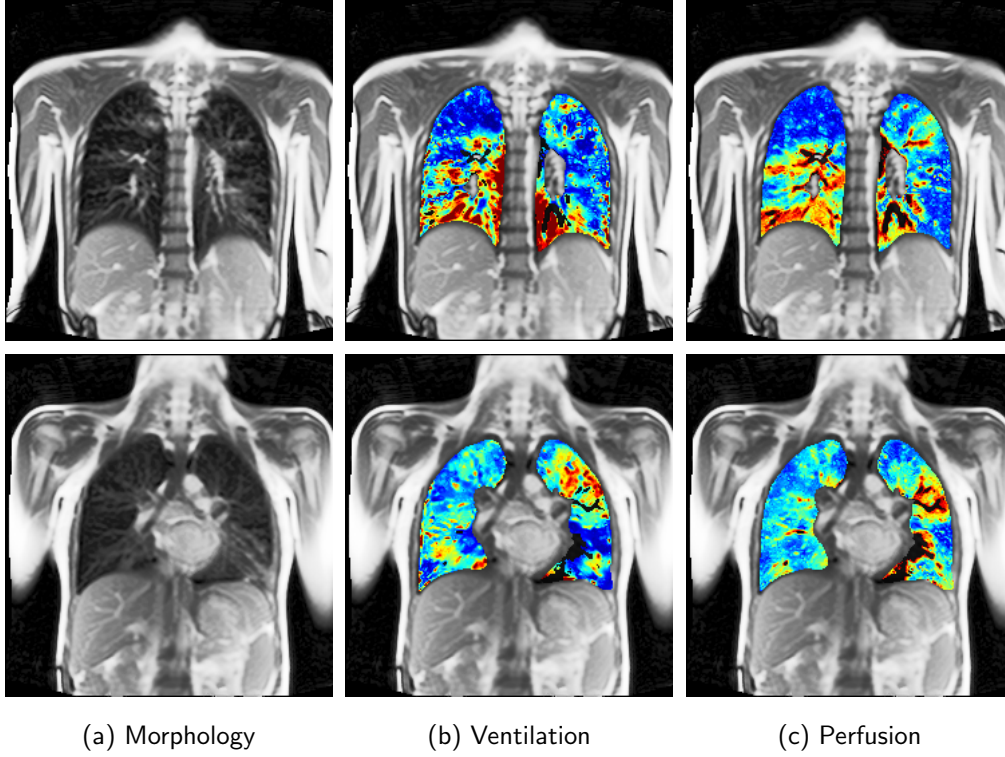
We define a medical imaging system as a function

$$S : \Omega_O \rightarrow \mathcal{X} \quad (2.1)$$

which maps from the object domain  $\Omega_O$  to the continuous image domain  $\mathcal{X} \subset \mathbb{R}^d$ . The output of the imaging system  $S$  is an image  $I$  defined as

$$I : \mathcal{X} \rightarrow \mathbb{R}^c, \quad (2.2)$$





**Figure 2.4:** Example of spatial ventilation and perfusion maps for selected coronal slices of the lung of a patient with CF.

were  $c$  is the number of color channels of the image. In this work, we focus on grayscale images with only one channel ( $c = 1$ ). In order to process the image, a compatible representation is needed. Therefore, the image signal is discretized in space. The size of the discretization is given as a pixel spacing  $\Delta x, \Delta y, \Delta z$  for each spatial dimension  $x, y, z$ . Figure 2.6b shows the effect of the image intensities after a spatial discretization. The discrete image domain  $X$  is then defined as a set of points

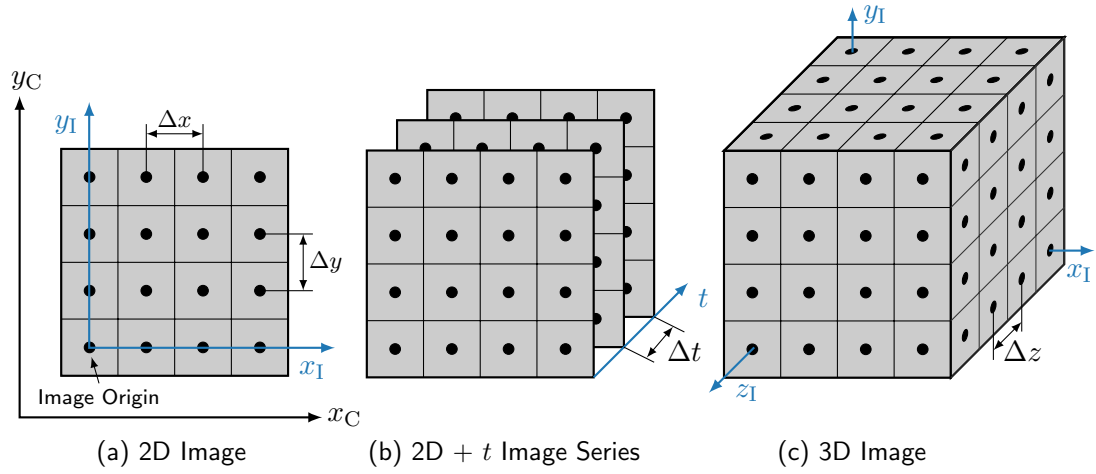
$$X := \{x_i\}_{i=1}^N, x_i \in \mathcal{X}, \quad (2.3)$$

where  $N$  is the number of pixels in 2D or voxels in 3D. In the case of an image time series, a discretization  $\Delta t$  is also performed on the time domain  $t$ .

Besides the discretization in space, a discretization of the image intensity range of the image is performed called quantization. Therefore, the intensity range of the image is discretized in  $B$  bins. The number of bins controls the number of possible different grayscale values in the image. For a given image  $I : X \rightarrow \mathbb{R}$  the intensity quantization is defined as

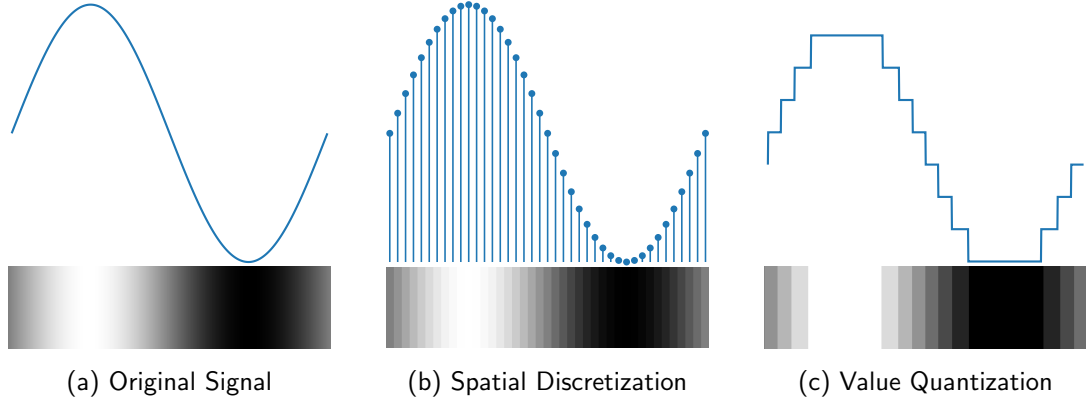
$$\tilde{I}(x_i) = \Delta_B \left\lfloor \frac{I(x_i)}{\Delta_B} + \frac{1}{2} \right\rfloor, \quad (2.4)$$

## 2 Medical Background



**Figure 2.5:** Medical image representation for 2D images, 2D+ $t$  image series, and 3D image volumes.  $(x_C, y_C)$  is the camera coordinate system.

where  $\Delta_B$  is the distance between each bin. Figure 2.6c shows an example of an intensity quantization with  $B = 16$  bins. Most of today's imaging systems in the clinical environment are digital imaging systems. This means no physical image is created as for example in older X-ray systems.



**Figure 2.6:** Effect of the signal discretization (b) in the spatial domain and (c) for the value quantization with 16 bins.

# Medical Image Registration

Medical image registration has been an active field of research over the last decades. Compared to other research areas in the field of medical image analysis, image registration is about the estimation of spatial correspondence between two or more images. Important registration problems are 2D-2D, 3D-3D, 2D-3D, and the registration of images from different imaging modalities, e.g., CT-MR.

The pairwise-registration problem of a fixed image  $F : \mathcal{X} \rightarrow \mathbb{R}$  and a moving image  $M : \mathcal{X} \rightarrow \mathbb{R}$  on the image domain  $\mathcal{X} \subset \mathbb{R}^d, d \in \mathbb{N}_{>0}$  can be described as a regularized minimization problem of the form

$$f^* = \arg \min_{f \in \mathcal{F}} \int_{\mathcal{X}} \mathcal{S}(F(x), M(x) \circ f(x)) + \lambda \mathcal{R}(f(x)) dx. \quad (3.1)$$

Here,  $f^* : \mathcal{X} \rightarrow \mathbb{R}^d$  is the transformation of interest and a minimizer of (3.1),  $\mathcal{F}$  the space of admissible transformations, and  $\mathcal{S} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  is an image similarity measure. Often  $\mathcal{S}$  is called data term. Furthermore,  $M(x) \circ f(x)$  defines the deformation of the moving image  $M$  with the transformation  $f$ , with  $M(x + f(x))$ . The regularizer  $\mathcal{R} : \mathbb{R}^d \rightarrow \mathbb{R}$  restricts the space of admissible transformations by adding prior knowledge to the minimization problem. The trade-off between the data term  $\mathcal{S}$  and the regularizer  $\mathcal{R}$  is achieved by the scalar weight  $\lambda \in \mathbb{R}_{\geq 0}$ . Several different approaches have been presented in the past in order to solve (3.1). A survey of different medical registration method is given by [Maintz and Viergever \(1998\)](#); [Sotiras et al. \(2013\)](#). In this work, we define four different categories of registration methods: *non-parametric*, *parametric*, *gradient-free*, and *learning-based* methods.

## 3.1 Non-Parametric Registration

Non-parametric methods are using the Calculus of Variations to find a minimizer of (3.1). One of the first method which described the image registration or image matching problem in the form of (3.1) was described by [Horn and Schunck \(1981\)](#). Given the 2D image sequence  $I(x, y, t) : \mathbb{R}^3 \rightarrow \mathbb{R}$ , where  $x, y$  represents the spatial coordinates of the image and  $t$  the time. The major assumption to compute the *optical flow* between two images acquired with a time delay  $\delta t$  is the brightness assumption [Horn and Schunck \(1981\)](#)

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t). \quad (3.2)$$

Here,  $\delta x$  and  $\delta y$  are the distances of corresponding brightness patterns of two images. Through a Taylor-Series expansion of the right side of (3.2), we obtain

$$I(x, y, t) = I(x, y, t) + \delta x \frac{\partial I}{\partial x} + \delta y \frac{\partial I}{\partial y} + \delta t \frac{\partial I}{\partial t} + \epsilon \quad (3.3)$$

$$0 = \frac{\partial I}{\partial x} \frac{\delta x}{\delta t} + \frac{\partial I}{\partial y} \frac{\delta y}{\delta t} + \frac{\partial I}{\partial t} + \frac{\epsilon}{\delta t}. \quad (3.4)$$

### 3 Medical Image Registration

Here,  $\epsilon$  contains higher order terms and for  $\delta t \rightarrow 0$ ,  $\delta x \rightarrow 0$ , and  $\delta y \rightarrow 0$ , we obtain the classical optical flow assumption

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \quad (3.5)$$

with the flow

$$u = \frac{dx}{dt}, v = \frac{dy}{dt}. \quad (3.6)$$

We can observe that (3.5) is an ill-posed problem as there is only one equation for the two unknown  $u$  and  $v$ , with  $f = [u, v]$ . This problem is also known as the aperture problem [Horn and Schunck \(1981\)](#). In order to solve this problem, the regularization

$$\mathcal{R}(u, v) = \int_{\mathcal{X}} \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial u}{\partial y} \right)^2 + \left( \frac{\partial v}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 dx \quad (3.7)$$

was presented by [Horn and Schunck \(1981\)](#). The final minimization problem is then given as

$$\mathcal{L} = \int_{\mathcal{X}} \left( \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} \right)^2 + \lambda \left( \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial u}{\partial y} \right)^2 + \left( \frac{\partial v}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 \right) dx. \quad (3.8)$$

As (3.8) is of the form

$$\mathcal{L} = \int_{\mathcal{X}} F(x, u, v, u', v') dx \quad (3.9)$$

the Euler-Lagrange formalisms

$$\frac{\partial F}{\partial u} - \frac{d}{dx} \frac{\partial F}{\partial u_x} - \frac{d}{dy} \frac{\partial F}{\partial u_y} = 0 \quad (3.10)$$

$$\frac{\partial F}{\partial v} - \frac{d}{dx} \frac{\partial F}{\partial v_x} - \frac{d}{dy} \frac{\partial F}{\partial v_y} = 0 \quad (3.11)$$

with

$$u_x = \frac{\partial u}{\partial x} \quad u_y = \frac{\partial u}{\partial y} \quad v_x = \frac{\partial v}{\partial x} \quad v_y = \frac{\partial v}{\partial y} \quad (3.12)$$

can be used to obtain the final differential equations

$$u \left( \frac{\partial I}{\partial x} \right)^2 + v \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} = \lambda (\nabla^2 u) - \frac{\partial I}{\partial x} \frac{\partial I}{\partial t} \quad (3.13)$$

$$v \left( \frac{\partial I}{\partial x} \right)^2 + u \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} = \lambda (\nabla^2 v) - \frac{\partial I}{\partial y} \frac{\partial I}{\partial t}. \quad (3.14)$$

Here,  $\nabla^2 = \sum_{i=1}^n \frac{\partial^2 g}{\partial^2 x_i}$  is the Laplace operator. In order to solve (3.13)-(3.14) a discretization of the transformation  $u$  and  $v$  is performed. [Horn and Schunck \(1981\)](#) used Lagrange-Multiplier and the Gauss-Seidel method to find a solution of (3.13)-(3.14).

In the past, several extensions of the optical flow method of [Horn and Schunck \(1981\)](#) have been presented. Extensions are mostly related to changes of the image similarity term, i.e., the brightness assumption, or the regularization. In (3.2) it is assumed that moving structures do not change their intensity values between two images. However, this holds not true for most natural images, and especially not for medical images. [Hermosillo et al. \(2002\)](#) presented different image similarity measures, e.g., Mutual Information, Cross Correlation, to overcome the problem of brightness changes in images. They integrated these similarity measures into the variational registration framework. A detailed overview of different image similarity measures and their applications is given in Section 3.6.

As mentioned above the regularizer  $\mathcal{R}$  is a necessary part for the variational registration approach and it allows the usage of prior knowledge of the transformation  $f$ . [Nagel and Enkelmann \(1986\)](#) present a regularization method, which was extended by [Alvarez et al. \(2000\)](#) which also use information of the image to control the regularization. Their loss function for the registration of 2D images has the form

$$\mathcal{J}(M, F, f) = \int_{\mathbb{R}^2} (F(x) - M(x + f(x)))^2 + \lambda \operatorname{tr} ((\nabla f(x))^T D(\nabla F)(\nabla f(x))) dx \quad (3.15)$$

with

$$D(\nabla F) = \frac{1}{|\nabla F|^2 + 2\alpha^2} \left[ \begin{pmatrix} \frac{\partial F}{\partial y} \\ -\frac{\partial F}{\partial x} \end{pmatrix} \begin{pmatrix} \frac{\partial F}{\partial y} \\ -\frac{\partial F}{\partial x} \end{pmatrix}^T + \alpha^2 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right]. \quad (3.16)$$

From (3.16), we can observe that the transformation is not smoothed across image boundaries in  $F$  for  $|\nabla F| \gg \alpha$ . The Euler-Lagrange equations for (3.15) are then

$$[F(x) - M(x + f(x))] \frac{\partial}{\partial x} M(x + f(x)) + \lambda \operatorname{div} (D(\nabla F) \nabla u) = 0 \quad (3.17)$$

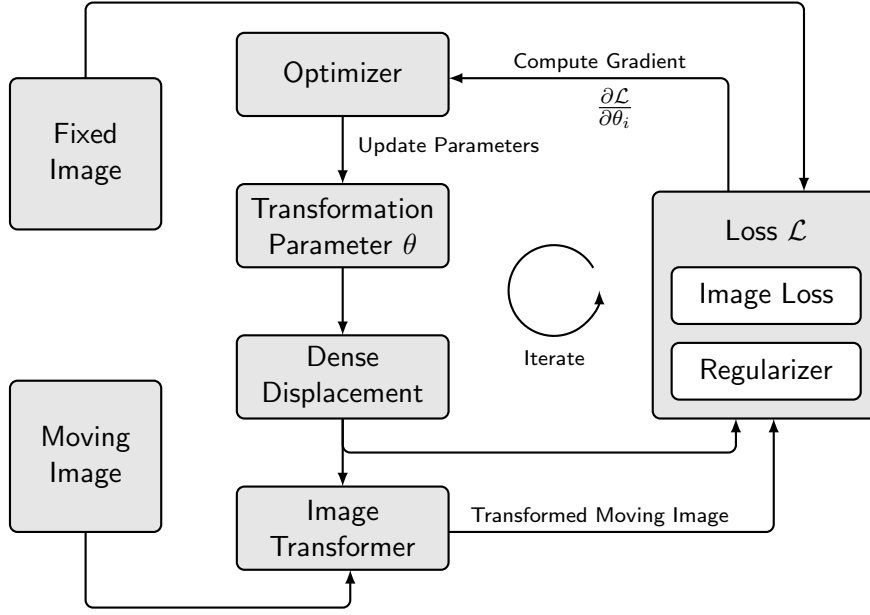
$$[F(x) - M(x + f(x))] \frac{\partial}{\partial y} M(x + f(x)) + \lambda \operatorname{div} (D(\nabla F) \nabla v) = 0. \quad (3.18)$$

## 3.2 Parametric Registration

Compared to the non-parametric registration methods described in Section 3.1 parametric registration methods use a predefined transformation model  $f_\theta$  for the transformation  $f$ . Here,  $\theta$  are the parameters of the transformation model. The registration problem is now defined as

$$f_\theta^* = \arg \min_{f_\theta \in \mathcal{F}_{f_\theta}} \int_{\mathcal{X}} \mathcal{S}(F(x), M(x) \circ f_\theta(x)) + \lambda \mathcal{R}(f_\theta(x)) dx, \quad (3.19)$$

where  $\mathcal{F}_{f_\theta}$  is the space of admissible functions for the transformation model  $f_\theta$ . In contrast to non-parametric registration methods, prior information of the transformation  $f$  can be added by the regularizer  $\mathcal{R}$  and the transformation model  $f_\theta$ . The process of parametric image registration is shown in Figure 3.1.



**Figure 3.1:** Parametric Image Registration Process.

### 3.2.1 Transformation Models

As mentioned above the transformation model is an implicit regularizer of the transformation, as depending on the properties of the transformation model. There are two major types of transformation models used in medical image registration: *dense* and *interpolating*. For the dense transformation a displacement vector

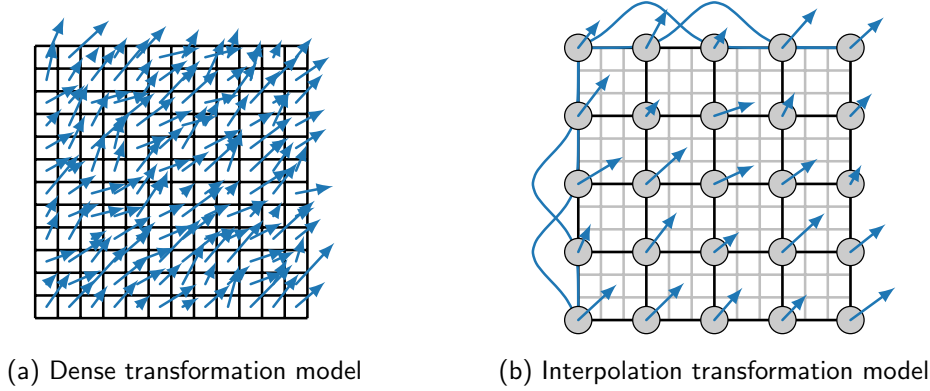
$$f_{\theta}(x) = \theta_x \quad (3.20)$$

is defined at the center position  $x$  of each pixel in the image, with  $\theta_x = (\vartheta_1, \vartheta_2, \dots, \vartheta_d) \in \mathbb{R}^d$ , where  $d$  is the dimension of the image data.

Interpolating transformation models are of the form

$$f_{\theta}(x) = \sum_{i=1}^N \theta_i k(x, c_i), \quad (3.21)$$

where  $\{c_i\}_{i=1}^N, c_i \in \mathcal{X}$  are the positions of the fixed regular grid points in the image domain,  $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  the interpolating function, and  $N$  the number of grid points. The transformation between the grid points is then an interpolation of the grid point values  $\theta_i$  with the kernel function  $k$ . An example of both transformation models is shown in Figure 3.2. Interpolating transformation models normally contain fewer parameters compared to the dense transformation model. In case the chosen interpolating function is smooth, the final transformation is smooth by definition. However, the disadvantage of interpolating transformation models is the smaller function space compared to dense



**Figure 3.2:** Dense and interpolation transformation models.

methods. These limitations are very important if it comes to model complex transformation, e.g., in the presence of sliding-organ motion (Section 3.9). Several different interpolating functions with different properties have been used in the past. Interpolating functions with a compact support like the third order B-spline function

$$k_{B1D}(x, y) := \begin{cases} \frac{2}{3} - |x - y|^2 + \frac{|x - y|^3}{2}, & 0 \leq |x - y| < 1 \\ \frac{(2 - |x - y|)^3}{6}, & 1 \leq |x - y| < 2 \\ 0, & 2 \leq |x - y| \end{cases} \quad (3.22)$$

used in [Rueckert et al. \(1999\)](#) or the  $C^4$  Wendland function

$$k_W(x, y) = \psi_{3,2} \left( \frac{\|x - y\|}{\sigma} \right), \quad (3.23)$$

$$\psi_{3,2}(r) = (1 - r)_+^6 \frac{3 + 18r + 35r^2}{3}, \quad (3.24)$$

used in [Jud et al. \(2016a,b\)](#) are common functions. Here,  $(\cdot)_+$  is the maximum function  $\max(0, x)$  and  $\sigma \in \mathbb{R}$  controls the kernel width. Interpolating functions without a compact support could also be used, e.g., the Gaussian function or thin plate splines.

### 3.3 Diffeomorphic Registration

Diffeomorphic registration methods restrict the space of admissible transformation as solutions to (3.1) to the space of diffeomorphic transformations  $Diff(\mathcal{X})$ . The major advantage of diffeomorphic transformations is that they are invertible and therefore preserve the topology of the image structures. A first approach of diffeomorphic registration was presented in [Christensen et al. \(1996\)](#); [Dupuis and Grenander \(1998\)](#); [Trouné \(1995\)](#). Based on this the large deformation diffeomorphic metric mapping (LDDMM) was presented by [Beg et al. \(2005\)](#). Here, the transformation of interest  $f \in Diff(\mathcal{X})$  is defined

### 3 Medical Image Registration

as the endpoint, i.e.,  $f = \phi_1$ , of the flow  $\phi_t : \mathcal{X} \rightarrow \mathcal{X}$ , on the unit time domain  $t \in [0, 1]$ . The flow is defined by the ordinary differential equation (ODE)

$$\frac{d\phi}{dt} = v_t(\phi_t), \quad (3.25)$$

and parametrized by a time-varying velocity field  $v_t : \mathcal{X} \rightarrow \mathbb{R}^d$ , with  $\phi_0 = id$ . The final transformation is then defined as

$$f = \phi_1 = \phi_0 + \int_0^1 v_t(\phi_t) dt \quad (3.26)$$

According to [Beg et al. \(2005\)](#) the solution of (3.25) is a diffeomorphism if the vector field  $v_t$  is smooth.

However, due to the time dependent velocity field  $v_t$  LDDMM methods are highly computationally complex. In order to reduce the computational complexity but keep the diffeomorphic characteristics of the LDDMM transformation a parametrization of the flow  $\phi_t$  with a stationary velocity field  $v$  was presented by [Arsigny et al. \(2006\)](#); [Ashburner \(2007\)](#); [Hernandez et al. \(2007\)](#). By using a stationary velocity field, the ODE (3.25) for the flow simplifies to

$$\frac{d\phi}{dt} = v(\phi_t). \quad (3.27)$$

As shown in [Hernandez et al. \(2007\)](#) the solution of (3.27) is a one-parameter subgroup with the infinitesimal generator  $v$ . Here,  $v \in \mathfrak{g}$  is the Lie algebra and  $\phi$  is an element of the Lie group  $G$  of diffeomorphic transformations  $Diff(\mathcal{X})$ . A diffeomorphism can be achieved with the exponential map  $\exp(v) : \mathbb{R}^d \rightarrow Diff(\mathcal{X})$  which maps from the Lie algebra to the Lie group. The final transformation is then defined as

$$f = \phi_1 = \exp(v), \quad (3.28)$$

where  $\exp(\cdot)$  defines the matrix exponential. By using the exponential map the inverse transformation is defined as

$$f^{-1} = \exp(v)^{-1} = \exp(-v). \quad (3.29)$$

The computation of the matrix exponential can be computationally very costly depending on the matrix size. There are several methods introduced in the past to compute the matrix exponential [Moler and Loan \(2003\)](#).

### 3.4 Demons Registration

In [Thirion \(1998\)](#) a new method to solve the registration problem (3.1) was presented. The name *Demons-Registration* is related to the idea of demons from the thermodynamic presented by James Clerk Maxwell. Thirion's demons registration methods can



**Algorithm 1** Demons registration framework1: **Inputs:** $F, M, N$  := number of iterations,  $\sigma$  := kernel size,  $\alpha$  := step size2: **Initialise:** $f \leftarrow 0$ 3: **for**  $i = 1$  to  $N$  **do**4:    $v \leftarrow \nabla S(F, M \circ f)$  compute image force (demons)5:    $f \leftarrow f + \alpha v$  update transformation6:    $f \leftarrow K_\sigma * f$  smooth transformation

be classified as a dense parametric registration approach consisting of two major parts. These two parts are the computation of the force of each demon  $v_i \in \mathbb{R}^d$  at each pixel position  $x_i$  in the image, and the regularization of the displacement field. A general overview of the demons registration method is given in Algorithm 1. Thirion presented a force calculation based on the optical flow equation (3.2) with

$$v_i = \frac{(M(x_i) \circ f(x_i) - F(x_i)) \nabla(M(x_i) \circ f(x_i))}{(\nabla(M(x_i) \circ f(x_i)))^2 + (M(x_i) \circ f(x_i) - F(x_i))^2}, \quad (3.30)$$

where  $\nabla$  is the nabla operator. For the regularization Thirion presented a diffusion based approach defined as

$$f_t = K_\sigma * (f_{t-1} + \alpha v) \quad (3.31)$$

where  $K_\sigma$  is a Gaussian kernel function of size  $\sigma$ ,  $*$  the convolution operator, and  $\alpha$  the step size in a gradient descend manner.

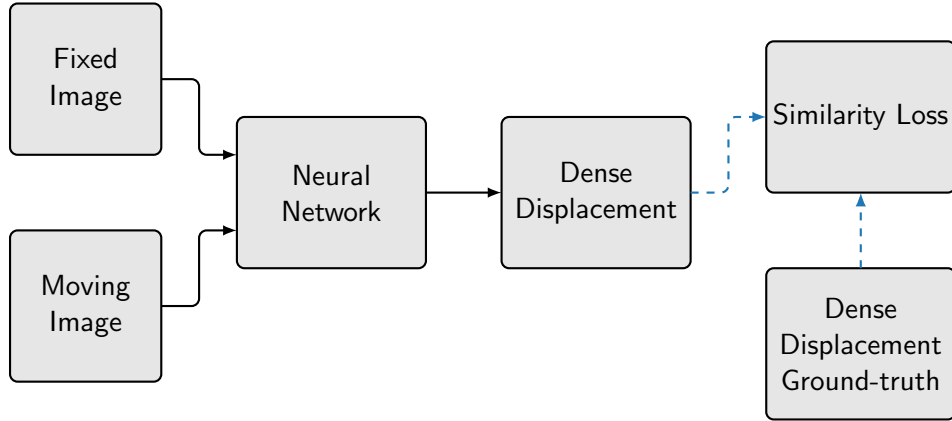
Several works investigated the theoretical background of the demons registration method. In [Pennec et al. \(1999\)](#) it was shown that the force calculation (3.30) presented by Thirion is similar to a second-order update scheme. For the regularization it was shown in [Cahill et al. \(2009\)](#) that the Gaussian kernel is the Green's function of the diffusion equation. This shows that the demons registration method is an efficient implementation of the diffusion-based image registration. In the past, several extensions of the demons registration method have been presented. The main changes are applied to the force computation using different image similarity functions [Lorenzi et al. \(2013\)](#) or the regularization [Cahill et al. \(2009\)](#); [Papież et al. \(2018\)](#); [Papież et al. \(2014\)](#); [Sandkühler et al. \(2018b\)](#). Furthermore, [Santos-Ribeiro et al. \(2016\)](#) presented an extension of the force update weight  $\alpha$  using momentum.

### 3.4.1 Diffeomorphic Demons

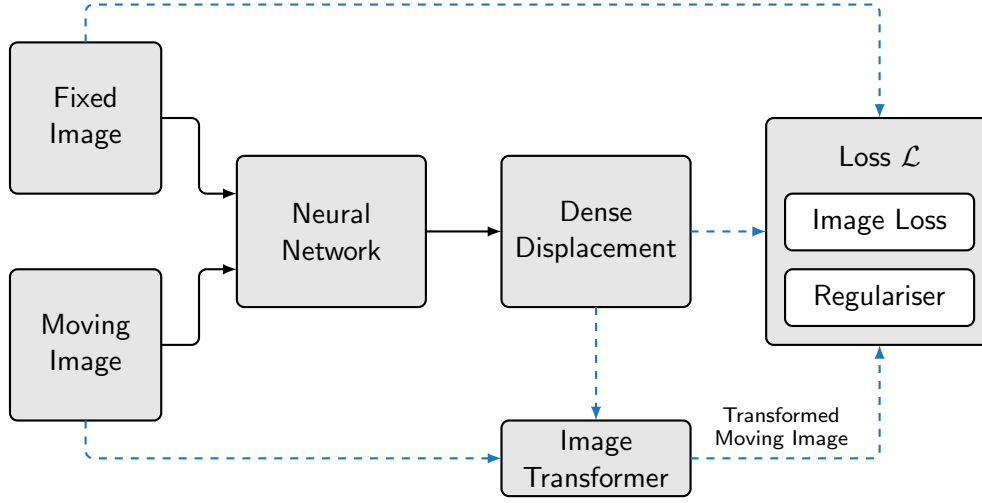
An essential extension was the introduction of diffeomorphic transformation (Section 3.3) to the demons registration [Vercauteren et al. \(2007\)](#). In order to compute the matrix exponential (3.28) the scaling and squaring algorithm can be used [Moler and Loan \(2003\)](#). Several modifications of the diffeomorphic demons have been presented in the past, e.g., [Arsigny et al. \(2006\)](#); [Vercauteren et al. \(2008\)](#).

### 3.5 Learning-based Image Registration

All mentioned registration methods from above, i.e., non-parametric, parametric, and demons registration, share a common issue. Namely, for each pair of images, the minimization problem (3.1) needs to be solved. Neural networks (NNs) especially convolutional neural networks (CNNs) have shown a drastic performance improvement for several tasks, e.g., image segmentation [Ronneberger et al. \(2015\)](#) and image classification [Krizhevsky et al. \(2012\)](#). In general, the usage of neural networks can be split in two parts: the *training* and the *inference* part. During training, a suited configuration of the parameter of the neural network is learned to perform the given task using appropriate training data. In the inference part the network is only evaluated and the parameters of the network stay constant. A major advantage of using NNs to perform the task of image registration is that a lower execution time for the inference part compared to a much longer execution time during training. A more detailed description of neural networks is given in Chapter 4. With this strategy, no optimization is needed for the registration of two images in the inference part which drastically reduces the time for the registration. The idea is that the transformation of interest  $f$  is modeled as a neural network.



**Figure 3.3:** Learning-based image registration with full supervision. Solid paths are used only for the inference and dashed path used during training.



**Figure 3.4:** Unsupervised learning-based image registration. Solid paths are used only for the inference and dashed path used during training.

There are two major directions presented in the literature for training a neural network for the task of image registration: *fully supervised* and *unsupervised*. Fully supervised learning for image registration were presented as the FlowNet [Dosovitskiy et al. \(2015\)](#), the FlowNet 2.0 [Ilg et al. \(2017\)](#) for 2D time series or in [Eppenhof et al. \(2018\)](#) for 3D medical images. An overview of the general fully supervised learning process for image registration is shown in Figure 3.3. For fully supervised learning the ground truth transformation between the fixed and the moving image is needed. Compared to other fully supervised learning tasks, e.g., image segmentation or classification, the generation of ground truth transformation is highly complex and time consuming. For the training of the FlowNet animated image sequences were used and [Eppenhof et al. \(2018\)](#) used artificial generated transformations for lung images to train the network.

For unsupervised learning there is no need to provide ground truth data. In the context of image registration only the fixed and the moving image are needed during the training. Figure 3.4 shows the structure of unsupervised learning for image registration. It shows that the unsupervised case is closely related to the classical image registration problem (3.1). In contrast to supervised methods, the moving image is transformed with the transformation of the network and then compared to the fixed image using a proper image similarity measure as presented in Section 3.6. Furthermore, a regularizer (Section 3.7) for the transformation can be added to the overall loss function. The basis for this kind of registration networks is the Spatial Transformer Network [Jaderberg et al. \(2015\)](#). Based on this idea, several registration networks were developed in the past [Dalca et al. \(2018b\)](#); [de Vos et al. \(2017\)](#); [Hu et al. \(2018\)](#); [Stergios et al. \(2018\)](#). A diffeomorphic extension is presented by [Dalca et al. \(2018a\)](#); [Krebs et al. \(2019\)](#); [Krebs et al. \(2018b\)](#). An overview of learning-based registration methods is given in [Fu et al. \(2019\)](#); [Haskins et al. \(2019\)](#).

### 3.6 Image Similarity Measures

A key element in image registration is the way both images are compared in terms of a pixel-wise similarity. There exists a variety of different image similarity measures for different kind of registration tasks. The selection of a suitable similarity measure is essential for image registration and especially for medical image registration. The brightness assumption (3.2) introduced by [Horn and Schunck \(1981\)](#) is a very basic measure and does not hold for all medical registration problems. A major problem for medical image registration is the intensity difference of similar structures in the fixed image  $F$  and the moving image  $M$ . These intensity changes occur due to physiological properties, e.g., perfusion, or through different imaging modalities, e.g.,  $F$  is an MR image and  $M$  is an CT image. In the following, we assume a discrete image domain  $X := \{x_i\}_{i=1}^N, x_i \in \mathcal{X}$ . Often the image measures are only evaluated on a subset of the image domain  $\tilde{X} \subset X$  in order to speed-up the registration process.

#### Mean Squared Error

The Mean Squared Error (MSE) similarity measure is similar to the brightness assumption of [Horn and Schunck \(1981\)](#). As mentioned above, similar structures are often represented with different image intensities which is not considered by the MSE similarity measure. However, the MSE is computationally efficient and therefore frequently used. The MSE is defined as

$$\mathcal{S}_{\text{MSE}}(F, M, f) = \frac{1}{|X|} \sum_{x \in X} (F(x) - M(x + f(x)))^2, \quad (3.32)$$

where  $|\cdot|$  is the cardinality.

#### Mutual Information

Mutual Information (MI) as image similarity was presented by [Wells et al. \(1996\)](#) for the registration of CT and MR images. The MI is defined as

$$\mathcal{S}_{\text{MI}}(F, M, f) = H(F) + H(M \circ f) - H(F, M \circ f), \quad (3.33)$$

where  $H(\cdot)$  is the marginal entropy and  $H(\cdot, \cdot)$  the joint entropy. In order to compute the entropy, we interpret  $F$  and  $M \circ f$  as two discrete random variables  $A$  and  $B$  with the joint probability  $p_{A,B}(a, b)$  and the marginal probabilities  $p_A(a)$  and  $p_B(b)$ . The joint Shannon entropy of  $A$  and  $B$  is defined as

$$H(A, B) = - \sum_{a \in A} \sum_{b \in B} p_{A,B}(a, b) \log_2(p_{A,B}(a, b)), \quad (3.34)$$

and the marginal Shannon entropy of a discrete random variable  $A$  is defined as

$$H(A) = - \sum_{a \in A} p_A(a) \log_2(p_A(a)). \quad (3.35)$$

### 3.6 Image Similarity Measures

For the computation of the MI similarity a differentiable estimation of joint probability function  $p_{A,B}$  and the marginal probability functions  $p_A$  and  $p_B$  is needed. A possible method is described in the following.

We assume that the images  $F : X \rightarrow \mathcal{I}_F$  and  $M \circ f : X \rightarrow \mathcal{I}_M$  map to different intensity ranges  $\mathcal{I}_F \subset \mathbb{R}$  and  $\mathcal{I}_M \subset \mathbb{R}$ . In the next step, we discretize the intensity ranges into  $B$  bins with  $I_F = \{b_i^F\}_{i=1}^{B_F}, b_i^F \in \mathcal{I}_F$  and  $I_M = \{b_i^M\}_{i=1}^{B_M}, b_i^M \in \mathcal{I}_M$ . Based on this, we calculate the matrices  $P^F \in \mathbb{R}^{|I_F| \times |X|}$  and  $P^M \in \mathbb{R}^{|I_M| \times |X|}$  with

$$P_{ij}^F = \frac{1}{\sigma_{I_F}} \exp\left(-\left(\frac{F(x_j) - b_i^F}{2\sigma_{I_F}}\right)^2\right), \quad (3.36)$$

$$P_{ij}^M = \frac{1}{\sigma_{I_M}} \exp\left(-\left(\frac{M(x_j) - b_i^M}{2\sigma_{I_M}}\right)^2\right). \quad (3.37)$$

The discrete marginal probabilities are then defined as

$$p_i^F = \frac{\frac{1}{|X|} \sum_{j=1}^{|X|} \frac{1}{2\pi} P_{ij}^F}{\sum_{i=1}^{|I_F|} \frac{1}{|X|} \sum_{j=1}^{|X|} \frac{1}{2\pi} P_{ij}^F} \quad \text{and} \quad p_i^M = \frac{\frac{1}{|X|} \sum_{j=1}^{|X|} \frac{1}{2\pi} P_{ij}^M}{\sum_{i=1}^{|I_M|} \frac{1}{|X|} \sum_{j=1}^{|X|} \frac{1}{2\pi} P_{ij}^M}. \quad (3.38)$$

The marginal Shannon entropies are then

$$H(F) = - \sum_{j=1}^{|I_F|} p_j^F \log_2(p_j^F) \quad \text{and} \quad H(M) = - \sum_{j=1}^{|I_M|} p_j^M \log_2(p_j^M). \quad (3.39)$$

An approximation of the the joint probability can be achieved with

$$p_{ij}^{F,M} = \frac{\frac{1}{|X|} \sum_{m=1}^{|X|} \frac{1}{2\pi} P_{im}^F P_{jm}^M}{\sum_{i=1}^{|I_F|} \sum_{j=1}^{|I_M|} \frac{1}{|X|} \sum_{m=1}^{|X|} \frac{1}{2\pi} P_{im}^F P_{jm}^M}. \quad (3.40)$$

With this the joint Shannon entropy is defined as

$$H(F, M \circ f) = - \sum_{i=1}^{|I_F|} \sum_{j=1}^{|I_M|} p_{ij}^{F,M} \log_2(p_{ij}^{F,M}). \quad (3.41)$$

With the formulation of (3.36) and (3.37), we explicitly use a Gaussian kernel density estimator, with the variances  $\sigma_{I_F}$  and  $\sigma_{I_M}$ , for the estimation of the probabilities. Furthermore, we assume that the Gaussian kernel for the density estimation is uncorrelated.

### 3 Medical Image Registration

#### Structural Similarity Index Measure

The structural similarity index measure (SSIM) was presented as an image quality criterion by [Wang et al. \(2004\)](#). For two local image patches  $a \in \mathbb{R}^d$ ,  $b \in \mathbb{R}^d$  the SSIM is defined as

$$\text{SSIM}(a, b) = l(a, b)^\alpha c(a, b)^\beta s(a, b)^\gamma, \quad (3.42)$$

with  $\alpha, \beta, \gamma \in [0, 1]$ . The SSIM combines three different similarity measures: the luminance ( $l$ )

$$l(a, b) = \frac{2\mu_a\mu_b + c_1}{\mu_a^2 + \mu_b^2 + c_1}, \quad (3.43)$$

the contrast ( $c$ )

$$c(a, b) = \frac{2\sigma_a\sigma_b + c_2}{\sigma_a^2 + \sigma_b^2 + c_2}, \quad (3.44)$$

and the structure ( $s$ )

$$s(a, b) = \frac{\sigma_{ab} + c_3}{\sigma_a\sigma_b + c_3}. \quad (3.45)$$

Here,  $\mu$  is the mean and  $\sigma$  the standard deviation of an image patch. Furthermore, the correlation coefficient is defined as  $\sigma_{ab}$ . The constants  $c_1, c_2, c_3 \in \mathbb{R}$  are used to reduce numerical instabilities in case the denominator of (3.43), (3.44), and (3.45) is close to zero. In order to use the SSIM as a similarity measure for image registration, (3.42) is computed it for every patch in the fixed and moving image. The SSIM is then defined as

$$\mathcal{S}_{\text{SSIM}}(F, M, f) = \frac{1}{|X|} \sum_{\substack{x \in F \\ y \in M \circ f}} l[x, y]^\alpha c[x, y]^\beta s[x, y]^\gamma. \quad (3.46)$$

Here,  $x$  and  $y$  are corresponding image patches of the fixed image  $F$  and the warped moving image  $M \circ f$ .

#### Normalized Cross Correlation

The normalized cross correlation (NCC) for the fixed image  $F$  and the warped moving image  $M \circ f$  is defined as

$$\mathcal{S}_{\text{NCC}}(F, M, f) = \frac{\sum_{x \in X} (F(x) - \mu_F)(M(x + f(x)) - \mu_{M \circ f})}{\sum_{x \in X} (F(x) - \mu_F) \sum_{x \in X} (M(x + f(x)) - \mu_{M \circ f})}, \quad (3.47)$$

where  $\mu_F$  and  $\mu_{M \circ f}$  are the mean of the fixed and the warped moving image respectively. A patch-based version of the NCC similarity measure was presented by [Cachier and Pennec \(2000\)](#). They described an efficient computation of the local NCC and its gradient using convolutions was introduced.

### Normalized Gradient Fields

Normalized gradient fields (NGFs) were presented by [Haber and Modersitzki \(2006\)](#) for the registration of multi-modal image data. The NGF is defined as

$$\mathcal{S}_{\text{NGF}}(F, M, f) = \frac{1}{|X|} \sum_{x \in X} \|n(F, x) \times n(M \circ f, x)\|^2, \quad (3.48)$$

where

$$n(I, x) := \begin{cases} \frac{\nabla I(x)}{\|\nabla I(x)\|}, & \nabla I(x) \neq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (3.49)$$

is the normalized gradient of an image  $I$ . According to [Haber and Modersitzki \(2006\)](#) the definition of the normalized gradient (3.49) is very sensitive to noise and not differentiable in homogeneous areas of the image. In order to overcome this issue, a regularized normalized gradient field is defined as

$$n(I, x)_{\mathcal{E}} := \frac{\nabla I(x)}{\|\nabla I(x)\|_{\mathcal{E}}}, \quad \|\nabla I(x)\|_{\mathcal{E}} := \sqrt{\nabla I(x)^T \nabla I(x) + \mathcal{E}^2}. \quad (3.50)$$

For the estimation of  $\mathcal{E}$ , [Haber and Modersitzki \(2006\)](#) propose

$$\mathcal{E} = \frac{\eta}{|X|} \sum_{x \in X} |\nabla I(x)|, \quad (3.51)$$

where  $\eta$  is the estimated noise level.

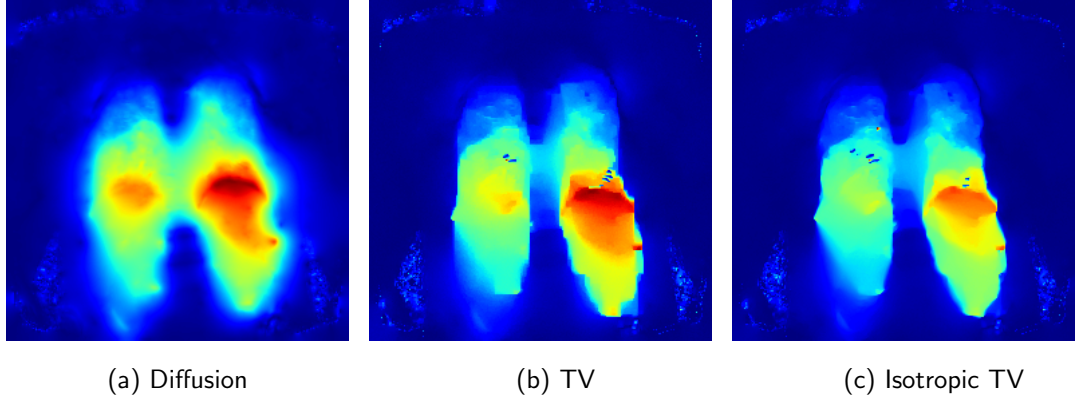
## 3.7 Regularizer

Regularization of the transformation is part of most image registration methods, either non-parametric (Section 3.1) or parametric (Section 3.2). For non-parametric registration methods the regularization is essential to obtain a solution (aperture problem [Horn and Schunck \(1981\)](#)). With a regularizer, we are able to restrict the space of admissible transformations by making use of prior knowledge of the transformation. In the following, general regularizers used in medical image registration are presented. Figure 3.5 shows the result of the presented regularization methods for the registration of MR images of the human thorax.

### Diffusion Regularizer

The diffusion regularizer enforces the transformation  $f$  to be smooth on the complete image domain  $\mathcal{X}$ .

$$\mathcal{R}_{\text{Diff}} = \frac{1}{|X|} \sum_{x \in X} \sum_{i=1}^d \|\nabla f_i(x)\|_2^2 \quad (3.52)$$



**Figure 3.5:** Results of different regularization methods on the final dense transformation model.

#### Total Variation

Total variation (TV) regularization was first used for image denoising [Rudin et al. \(1992\)](#) and later in the field of image registration [Pock et al. \(2007\)](#); [Sun et al. \(2010\)](#). [Vishnevskiy et al. \(2014\)](#) applied TV regularization to parametric transformation models. Similar to the diffusion regularization the TV regularization enforces a smooth displacement, but in addition it allows discontinuities in the displacement. The TV regularization is defined as

$$\mathcal{R}_{\text{TV}} = \frac{1}{|X|} \sum_{x \in X} \sum_{i=1}^d \sqrt{\nabla f_i(x)^2}. \quad (3.53)$$

#### Isotropic Total Variation

In order to be able to describe discontinuities that are not aligned with the coordinate axis the isotropic TV regularization

$$\mathcal{R}_{\text{iTV}} = \frac{1}{|X|} \sum_{x \in X} \sqrt{\sum_{i=1}^d \nabla f_i(x)^2} \quad (3.54)$$

was introduced for image registration by [Vishnevskiy et al. \(2016\)](#).

### 3.8 Optimizer

Optimizers are essential for medical image registration and have a major impact on the registration result. In [Klein et al. \(2007\)](#) the influence of different optimizers for a fixed transformation model and loss function was evaluated. The major class of optimization methods used in image registration are gradient-based optimization methods, e.g., gradient descent, limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS). In



order to use gradient-based optimization methods the gradient of the minimization problem (3.1) is needed.

For some image registration problems, especially 3D image registration the gradient computation is very time-consuming. To overcome this problem, the evaluation and the gradient computation of the equation of the minimization problem (3.1) are only performed on a subset  $\tilde{X} \subset X$  of the image domain. Optimizers that are able to work with the stochastic approximation of the gradient are for example the adaptive stochastic gradient descent (ASGD) [Klein et al. \(2008\)](#), the stochastic LBFGS [Qiao et al. \(2015\)](#), the ADAM optimizer [Kingma and Ba \(2014\)](#), the ADADELTA optimizer [Zeiler \(2012\)](#), or the Adagrad optimizer [Duchi et al. \(2011\)](#).

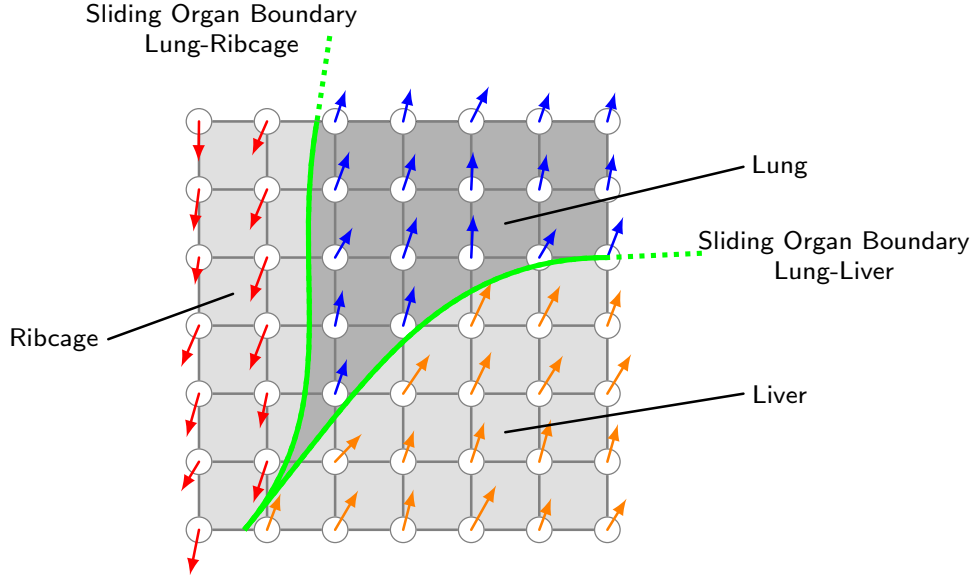
#### 3.8.1 Gradient-Free Optimization

As mentioned before the major class of optimization methods used in image registration require the analytic calculation of the gradient of the minimization problem (3.1). In [Glocker et al. \(2008\)](#); [Heinrich et al. \(2013\)](#) gradient-free registration methods based on discrete Markov random fields were presented. The registration problem is reformulated to a classification problem and a discrete optimization is used. Without the need for the gradients non-differentiable image similarity measures and regularization methods could be used.

## 3.9 Sliding-Organ Motion Registration

Sliding-organ motion is a major challenge in medical image registration. This kind of motion occurs at organ boundaries where each organ has a different motion. A very important example of sliding-organ motion is the registration of images of the human thorax. Figure 3.6 shows a schematic view of the dominant motion of the human thorax during breathing. It shows three different anatomical structures: the liver, the lung, and the ribcage. Each of these structures possesses a different motion during breathing. The major difference in terms of motion is between the ribcage and the liver and lung, where the motion between the liver and the lung only slightly differs. These kind of transformations are difficult to obtain, for a registration method, as they require two properties. The first is the smoothness property of the transformation inside each organ. The second property is the sharp transition, i.e., the discontinuity of the transformation, at corresponding organ boundaries. In order to separate the different transformations from each other a detection of this boundary is necessary. However, a segmentation of the different transformation regions based on image intensities is very difficult, as the motion is not always correlated to the intensities. An example for this is the boundary between the liver and the ribcage shown in Figure 3.6.

In the past, several registration methods using different transformation models (dense, interpolating) have been introduced to improve the registration of sliding organs. To preserve local discontinuities with an interpolating transformation model, a stationary first order B-Spline kernel with a TV regularization was introduced in [Vishnevskiy et al.](#)



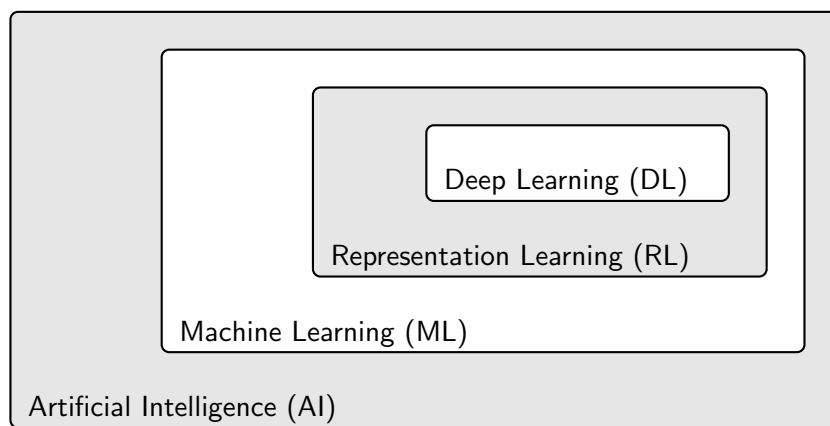
**Figure 3.6:** Sliding-organ boundaries schematic for thorax images between the liver, the ribcage, and the lung. Arrows indicate the corresponding motion of each organ during exhalation. Sliding-organ boundaries are shown in green.

(2016). Creating a non-stationary kernel based on the image intensities was presented in Jud et al. (2016b). A method that uses the local statistics of the motion field to distinguish between different motion and their influence on the parameter update was presented in Jud et al. (2017). Dense transformation models are very well suited for modeling discontinuities of the transformation, but a smooth transformation inside each organ must be enforced. An approach where bilateral filtering is used as regularization to preserve the discontinuities of the transformation was presented in Papież et al. (2014) for a dense transformation model and in Jud et al. (2016a) for interpolating transformation models. Kiriyanthan et al. (2012); Kiriyanthan et al. (2016) presented a method where the registration and the corresponding motion segmentation is performed in an alternating manner. Using supervoxels to model sliding boundaries was shown in Heinrich et al. (2013). Direction-dependent regularization methods based on the image intensities or the transformation are presented in Demirovic et al. (2009); Nagel and Enkelmann (1986); Schmidt-Richberg et al. (2012). Graph-based formulation for TV regularization was presented in Bagnato et al. (2009) and in Papież et al. (2017) a non-local regularization based on the minimum spanning tree was developed.

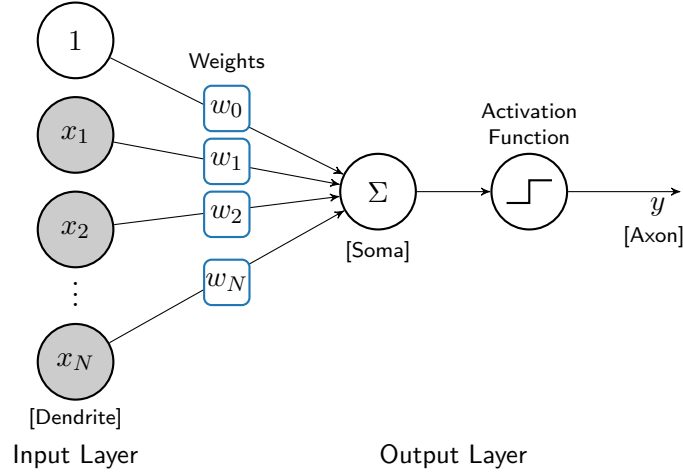
# Machine Learning

*Machine Learning* (ML) gained attention in the public domain in the last years through the presentation of computer programs that are able to master classical Atari games [Mnih et al. \(2013\)](#) or AlphaGo [Silver et al. \(2016\)](#) which outperforms humans in the board game Go. In the scientific community, *Deep Learning* (DL) has lead to a paradigm shift across several domains. As shown in Figure 4.1 deep learning methods are part of *Representation Learning* (RL) methods. Compared to classical machine learning methods, e.g., Support Vector Machine (SVM), where the features for a specific task are handcrafted, representation learning methods learn an internal representation that is needed to perform a specific task from given raw training data [Goodfellow et al. \(2016\)](#); [LeCun et al. \(2015\)](#). One of the first ideas of deep learning was the *Multilayer Perceptron* (MLP). The MLP is an extension of the by [Rosenblatt \(1958\)](#) presented perceptron. Until the development of the backpropagation algorithm by [Parker \(1985\)](#); [Rumelhart et al. \(1985\)](#); [Werbos \(1974\)](#), MLPs were difficult to train. Furthermore, it was shown in [Blum and Rivest \(1989\)](#) that finding the corresponding weights to perform a task correctly, e.g., classification, for a three-layer MLP is NP-complete without using the backpropagation algorithm.

Current deep learning methods applied to images are mostly *Convolutional Neural Networks* (CNNs) [LeCun \(1989\)](#). A milestone of deep learning and CNNs was the presentation of a classification network based on CNNs [Krizhevsky et al. \(2012\)](#). It drastically outperformed other state-of-the-art methods at the ImageNet Large Scale Visual Recognition Challenge. Since then, most of the presented neural networks for image processing are based on CNNs. Famous examples are the generative adversarial neural networks (GANs) for image generation [Goodfellow et al. \(2014\)](#) or the U-Net for semantic medical image segmentation [Ronneberger et al. \(2015\)](#).



**Figure 4.1:** Overview of different learning definitions and their relations according to [Goodfellow et al. \(2016\)](#).



**Figure 4.2:** Schematic model of the perceptron presented by [Rosenblatt \(1958\)](#). Corresponding biological notations are given in square brackets.

In general, machine learning is about the development and application of methods that are capable to perform a specific task without being given a specific description on how to perform it. They are able to specialize themselves to the task by learning from given training data. According to [Deisenroth et al. \(2019\)](#) machine learning is based on three concepts: *model*, *data*, and *learning*. The model can be described as a function

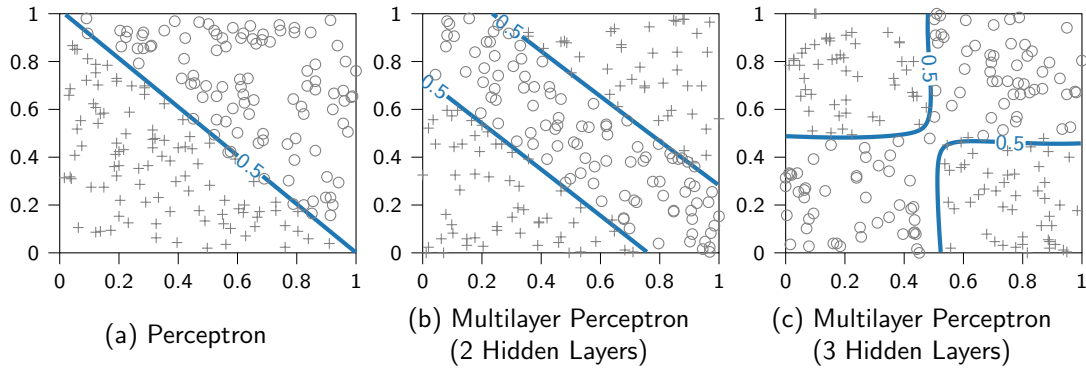
$$f_{\theta} : \mathbb{R}^{d_I} \rightarrow \mathbb{R}^{d_O} \quad (4.1)$$

that is used to approximate an unknown function  $f$ . Here,  $\theta$  are the parameters of the model,  $\mathbb{R}^{d_I}$  the input domain, and  $\mathbb{R}^{d_O}$  the output domain. The aim is to find a configuration of the parameters  $\theta$  in order to obtain an accurate approximation of the function  $f$ . As mentioned before, machine learning methods are capable to specialize themselves to a specific task. The specialization in this case is finding the best parameter configuration  $\theta^*$  for the given training data. This process is called learning.

## 4.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) are models used for machine learning that was inspired by biological neural networks. In [McCulloch and Pitts \(1943\)](#) a first logical neuron is presented to describe the logical functions of the brain. Based on this [Rosenblatt \(1958\)](#) presented the *perceptron*. A graphical representation of the perceptron is shown in Figure 4.2. The perceptron consists of two layers: the input layer and the output layer and is defined as

$$y(x) = \begin{cases} 1, & \text{if } \sum_{i=1}^N x_i w_i + w_0 > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (4.2)$$

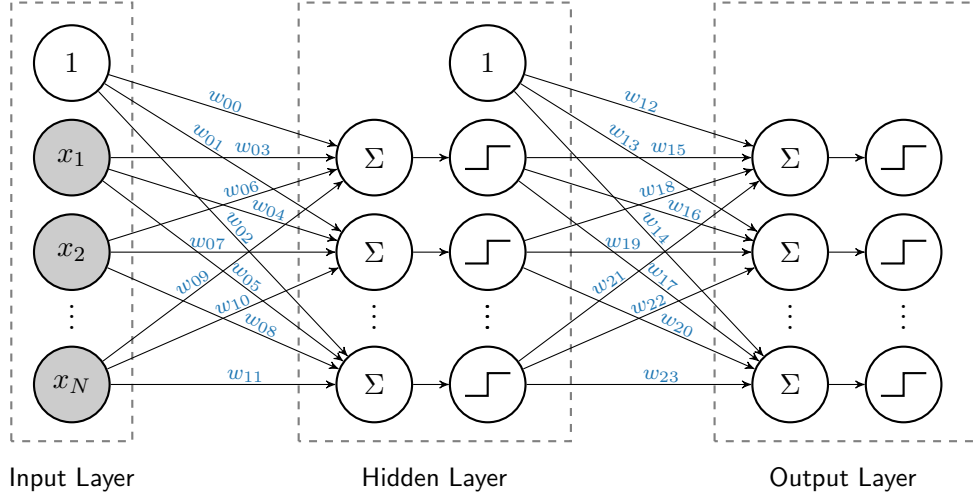


**Figure 4.3:** Classification results for two classes for different ANNs models. (a) shows the learned classification function with the perceptron model, (b) MLP model with two hidden layers with 10 neurons each, and (c) MLP with three hidden layers and 50 neurons each. All models were trained with the backpropagation algorithm and the sigmoid activation function. The blue line shows the learned class boundary.

Here,  $x_1, \dots, x_N$  are the inputs of the perceptron and  $w_0, \dots, w_N$  are the learnable parameters, i.e., the weights, of the network. Since the definition of the perceptron is inspired by biological neural networks, we can also find the corresponding biological elements in the perceptron. The input  $x_1, \dots, x_N$  corresponds to the different inputs of a neuron called dendrites and the output  $y$  corresponds to the axon. In the cell body, the soma, the different input signals are added up and in case a given threshold is reached a signal is emitted. This corresponds to the weighted sum and the threshold function of the perceptron. With this simple neural network, we are able to perform a simple task, e.g., classification. Figure 4.3a shows the result of the perceptron used for a 2D classification problem. The perceptron is able to learn only linear classification functions. This was the major weakness that was claimed by [Minsky and Papert \(1969\)](#). In detail, they prove that the perceptron is not able to learn the logic XOR function. This statement leads to the so-called *AI Winter*. During this time, research in this field was reduced to a minimum.

#### 4.1.1 Multilayer Perceptron

The extension of the perceptron is the *Multilayer Perceptron* (MLP). Compared to the original perceptron, a MLP contains additional hidden layers. Figure 4.4 shows a schematic view of an MLP with one hidden layer. With the development of the backpropagation algorithm [Parker \(1985\)](#); [Rumelhart et al. \(1985\)](#); [Werbos \(1974\)](#) it is possible to efficiently optimize the parameters of an MLP. An MLP can have several hidden layers and multiple neurons per layer. The output of the  $j$ -th neuron in the first



**Figure 4.4:** Schematic model of a multilayer perceptron (MLP) with one hidden layer.

hidden layer of an MLP is defined as

$$h_j = \phi \left( \sum_{i=1}^{N_{\text{Input}}} w_{i,j} x_i + w_{0,j} \right) \quad (4.3)$$

and for the  $j$ -th output neuron as

$$o_j = \phi \left( \sum_{i=1}^{N_{\text{Hidden}}} w_{i,j} h_i + w_{0,j} \right), \quad (4.4)$$

where  $\phi(\cdot)$  is an activation function. With a MLP it is possible to learn non-linear classification functions. The perceptron and the MLP are part of so-called *feed forward* neural networks, because there are no feedback connections of the output back to the model [Goodfellow et al. \(2016\)](#). Neural Networks with feedback connections are called *recurrent* neural networks (RNNs). A detailed description of RNNs is given in Section 4.3. Figure 4.3b shows the result of a learned non-linear classification function with two hidden layers and in Figure 4.3c for three hidden layers. All of them were trained using the backpropagation algorithm and the sigmoid activation function.

### Activation Functions

An essential part of a single neuron (perceptron) is the activation function. With the activation functions, the model is transformed from a linear to a non-linear model. The original perceptron presented by [Rosenblatt \(1958\)](#) used the Heaviside activation function. In the following, we will present a selection of common activation functions used in ANNs. Figure 4.5 shows the corresponding plots of the presented activation functions.

## Heaviside Activation Function

$$\text{Heaviside}(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \quad (4.5)$$

## Hyperbolic Tangent

$$\text{Tanh}(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \quad (4.6)$$

## Sigmoid

$$\text{Sigmoid}(x) = \frac{1}{1 + \exp(-x)} \quad (4.7)$$

## Rectified Linear Unit

$$\text{ReLU}(x) = \max(x, 0) \quad (4.8)$$

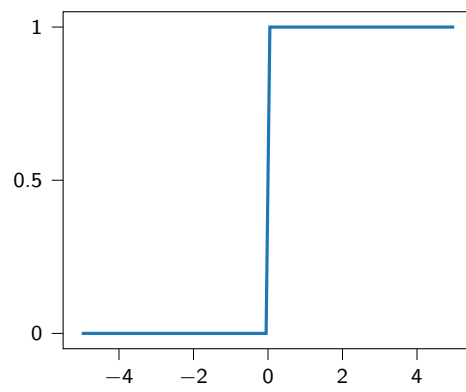
## Leaky Rectified Linear Unit

$$\text{LeakyReLU}(x) = \max(x, 0) + \text{slope} \min(0, x) \quad (4.9)$$

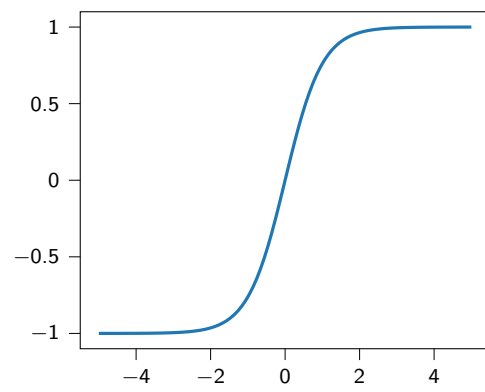
## Exponential Linear Unit

$$\text{ELU}(x) = \max(x, 0) + \min(0, \alpha \exp(x) - 1) \quad (4.10)$$

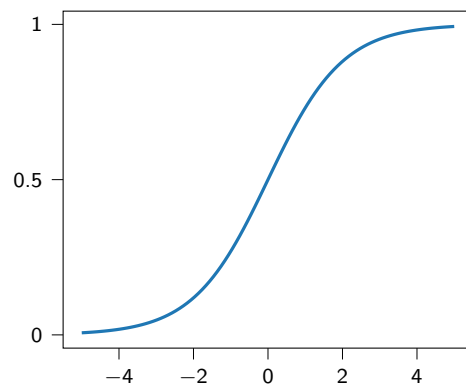
#### 4 Machine Learning



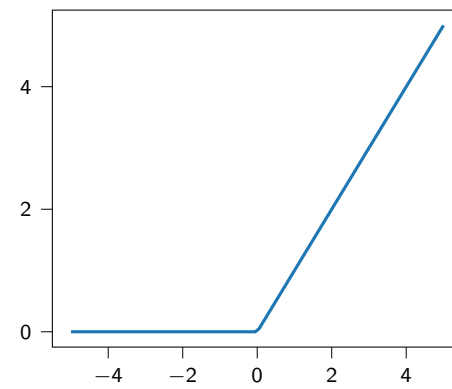
(a) Heaviside



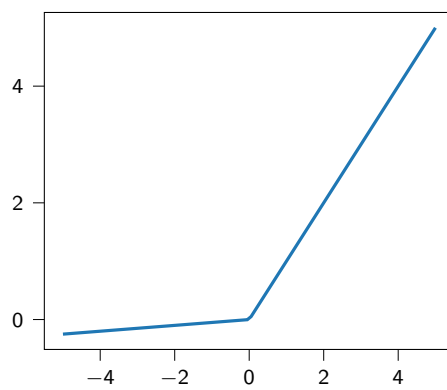
(b) Hyperbolic Tangent



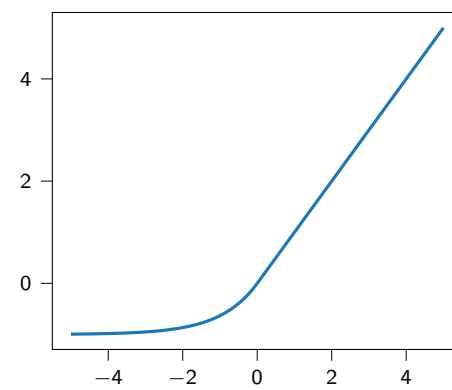
(c) Sigmoid



(d) ReLU



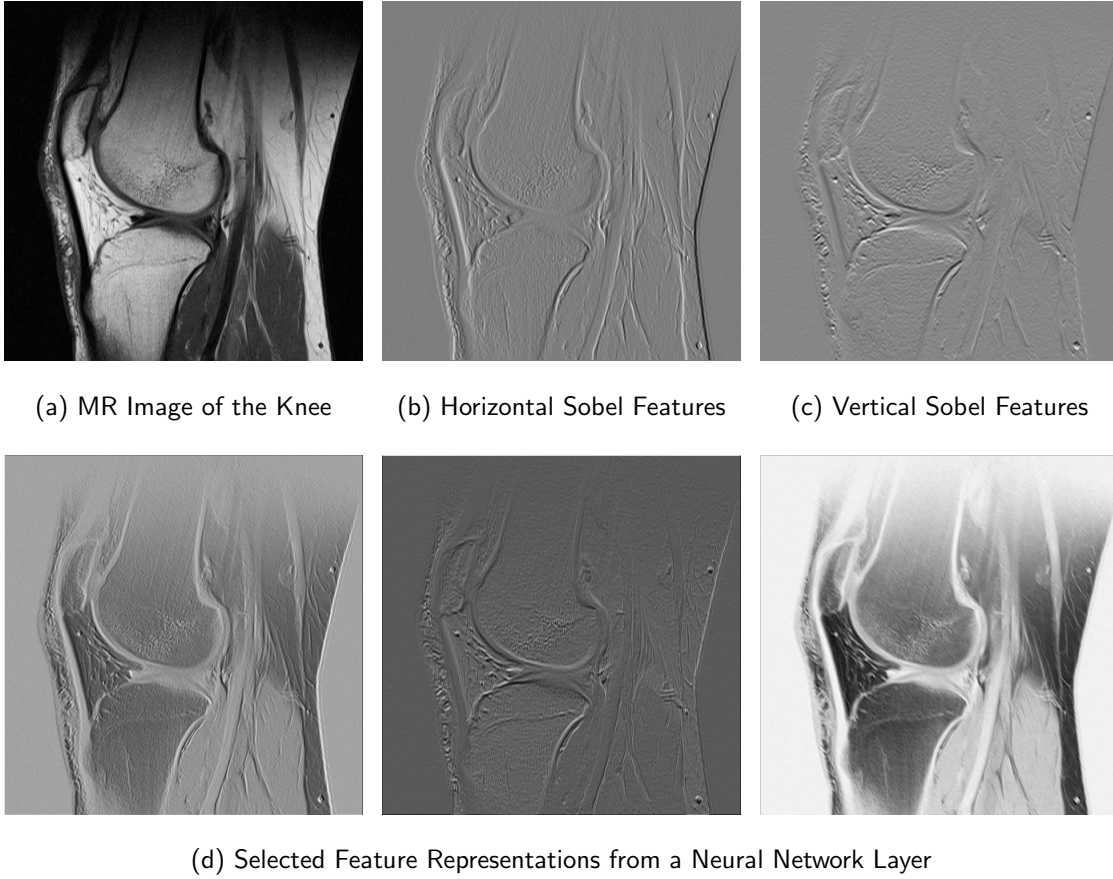
(e) LeakyReLU  
(slope 0.05)



(f) ELU  
( $\alpha = 1$ )

**Figure 4.5:** Different activation functions used for ANNs.

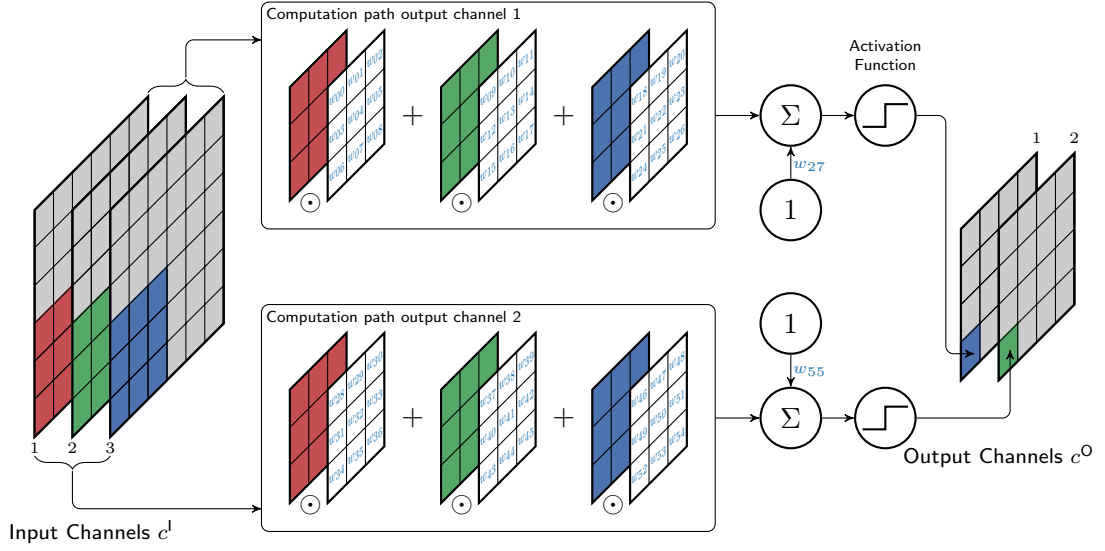




**Figure 4.6:** Examples of handcrafted and generated features extracted for an MR image of the knee. (a) MR image of the knee, (b) Sobel horizontal edge detection features, (c) Sobel vertical edge detection features, and (d) selected features of a CNN layer trained for image reconstruction in an autoencoder network.

## 4.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have shown huge success in the field of ANN research, especially in the field of image processing, e.g., [Krizhevsky et al. \(2012\)](#). They were introduced by [LeCun \(1989\)](#). The main idea behind the CNNs described by [LeCun \(1989\)](#) is the use of local connectivity. Compared to the MLP (Figure 4.4), the weights in each layer are locally shared. According to [LeCun \(1989\)](#) this results in a better generalization of the model because of the reduced number of parameters. The local connectivity can be modeled as a convolution which leads to the name convolutional neural networks. First CNNs for digit classification were presented in [LeCun et al. \(1990\)](#) and further extensions in [Lecun et al. \(1998\)](#). A convolution of two 2D functions



**Figure 4.7:** Example of a convolutional neural network layer with three input channels and two output channels. Here  $\odot$  is the Hadamard product and  $w_{00}$  to  $w_{55}$  are the learnable parameters.

$f : \mathbb{R}^2 \rightarrow \mathbb{R}$  and  $k : \mathbb{R}^2 \rightarrow \mathbb{R}$  is defined as

$$(f * g)(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\tau, \gamma) k(x - \tau, y - \gamma) d\tau d\gamma. \quad (4.11)$$

where  $f$  is the *signal* and  $k$  is the *kernel*.

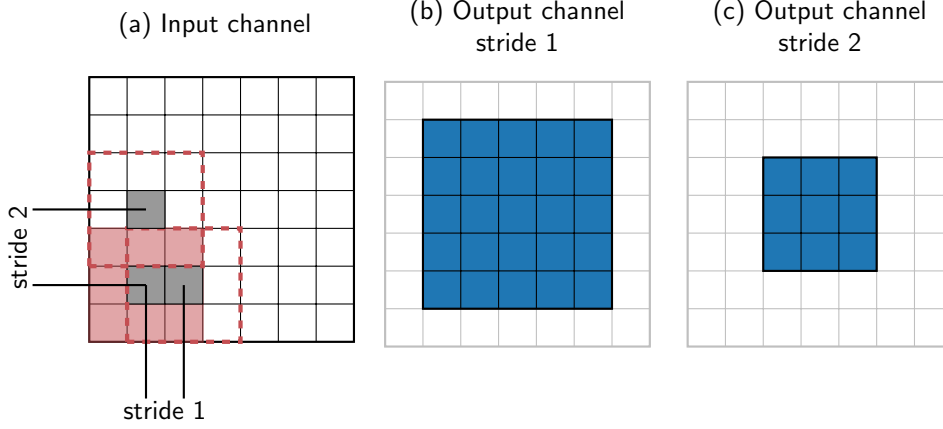
In general, the input data for a convolutional neural network are discrete signals, e.g., images. With the commutative property of the convolution [Goodfellow et al. \(2016\)](#), one can write the discrete convolution as

$$(f * g)(i, j) = \sum_{m=-M}^M \sum_{n=-N}^N f(i - m, j - m) k(m, n), \quad (4.12)$$

where  $M$  and  $N$  are the width and the height of the kernel  $k$ . Furthermore, we assume that the coordinate origin is in the center of the kernel. However, in most machine learning libraries the cross-correlation

$$(f \star g)(i, j) = \sum_{m=-M}^M \sum_{n=-N}^N f(i + m, j + n) k(m, n), \quad (4.13)$$

is used instead of the convolution [Goodfellow et al. \(2016\)](#). The major aim of the convolution layer is to extract local features of the input with a stationary kernel  $k$ . The kernel size is in general much smaller than the image size. Typical kernel sizes for 2D images are  $3 \times 3$ ,  $5 \times 5$ , or  $7 \times 7$ . Before CNNs, convolutions were also used



**Figure 4.8:** Visualization of the output of a strided convolution for (b) a stride of 1 and (c) a stride of 2. The blue colored regions representing the new size of the output for different strides.

to extract features from images, but there the kernel function was handcrafted. Those features are then used for example by SVMs to perform a classification tasks. Known kernel functions are for example the Sobel edge detection filter. Figure 4.6 shows an example of the features generated with the Sobel filter from an MR image of the knee. Furthermore, it shows selected features of a CNN layer of an autoencoder network that was trained for the task of image compression and reconstruction. It can be observed that some features are similar to the Sobel features.

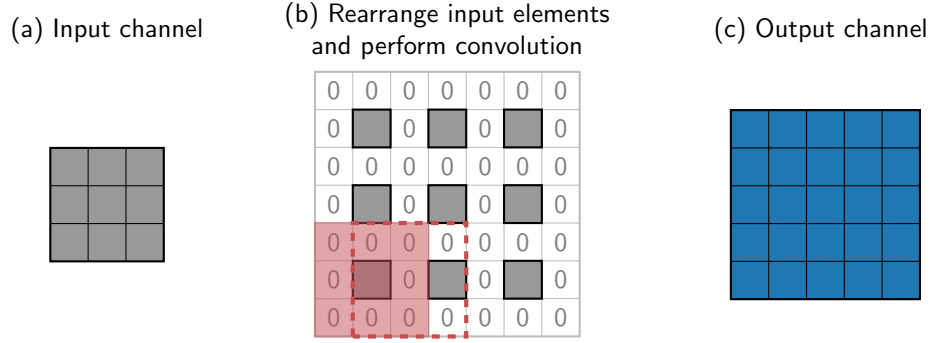
CNN layers are part of representation learning methods (Figure 4.1). As mentioned above, representation learning methods are able to learn an internal representation to fulfill a specific task. In a CNN layer, the internal representation is defined by the weights of the kernel. This means a CNN layer selects the features that are needed to perform the task by learning the corresponding weights of the kernel. Figure 4.7 shows the general structure of a CNN layer with the corresponding learnable weights. In general, a CNN layer uses a multichannel convolution in order to learn more than one feature per layer. The  $j$ -th output channel of a CNN layer is then defined as

$$c_j^O(x, y) = \phi \left( \sum_{i=1}^{N_I} (c_i^I * k_{i,j})(x, y) + b_j \right), \quad (4.14)$$

where  $i$  is the index for the input channels,  $c^I$  the input channels,  $c^O$  are the output channels (feature maps),  $k_{i,j}$  is the kernel applied to the  $i$ -th input channel,  $b_j$  the bias,  $\phi(\cdot)$  the activation function, and  $N_I$  the number of input channels.

### Strided Convolutions

There are several specializations of convolutions used in CNNs. One specialization is the strided convolution. Strided convolutions are used in order to reduce spatial resolution



**Figure 4.9:** Visualization of a transposed convolution. The input channel is first rearranged with a given stride and filled with zeros (b). In addition, in this example zero padding is applied at the image border. After this, a convolution is applied which results in the final output channel (c).

of the input channel (down-sampling). The strided convolution is defined as

$$(f * g)(i, j, s) = \sum_{m=-M}^M \sum_{n=-N}^N f(is + m, js + n)k(m, n), \quad (4.15)$$

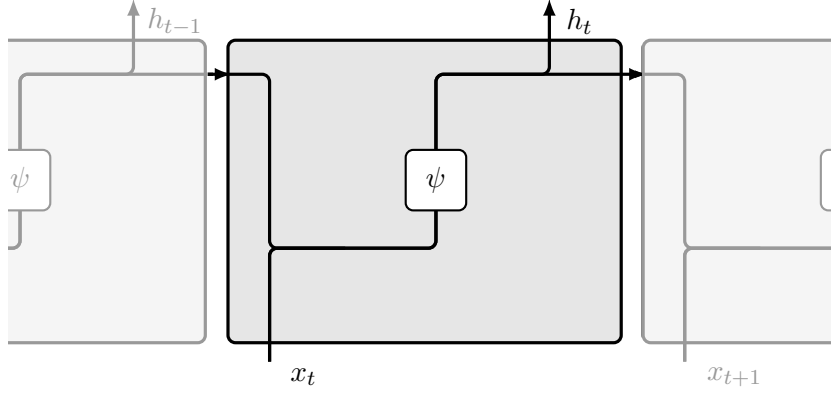
where  $s$  is the stride. It is possible to define different strides for each spatial dimension. Figure 4.8 shows an example of a strided convolution. Besides the strided convolution other down-sampling mechanisms are used. Examples are maximum pooling or average pooling layers where either the maximum or the average of a local neighborhood is selected. Maximum and average pooling are additional layers, which are arranged after the convolutional layer.

### Transposed Convolutions

Transposed convolutions can be used to increase the spatial resolution of the input channel (up-sampling). Figure 4.9 shows an example of the effect of transposed convolutions used for up-sampling. Up-sampling layers are used for example in the generator network of GANs [Goodfellow et al. \(2014\)](#) or in the decoder part of an autoencoder network. Alternatives for up-sampling are classical interpolation methods, e.g., linear interpolation. In [Wojna et al. \(2019\)](#) a comparison of different up-sampling techniques is presented.

## 4.3 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a special class of neural networks for the processing of sequential data. Compared to feedforward neural networks described in Section 4.1 RNN poses feedback connections. In detail, this means that the current output of an RNN is depending on the last output. With this property, RNNs are well suited for the



**Figure 4.10:** Unrolled recurrent neural network.

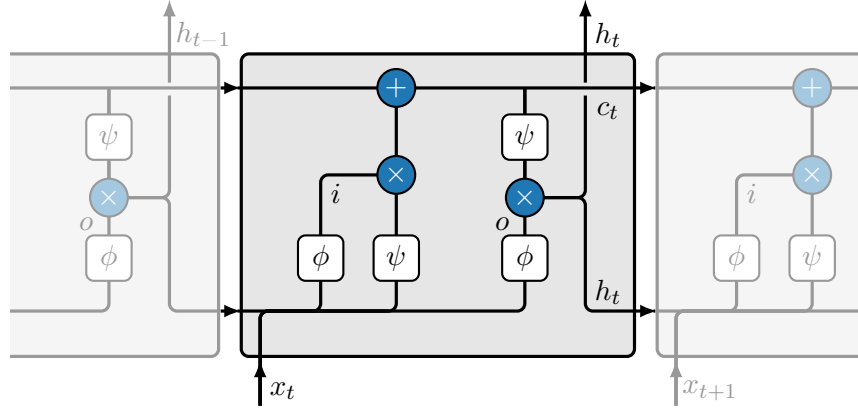
detection of features that are encoded in the sequence itself. A RNN is defined as

$$h_t = \psi(Wx_t + Uh_{t-1}), \quad (4.16)$$

where  $h_t$  is the output of the RNN,  $x_t$  is the input at time  $t$ , and  $\psi$  is the activation function. The learnable weights of the RNN are the weighting matrix  $W$  of the input and  $U$  the weight matrix of the last output at  $t - 1$ . A graphical representation of an unfolded RNN is shown in Figure 4.10. RNNs can also be seen as a special method to share parameters. Compared to CNNs where the parameter sharing is done for the spatial domain RNNs share parameter across the sequence. By using the same parameter configuration for each element of the sequence RNNs are able to handle sequences with an arbitrary sequence length. In this work, RNNs are used to build a sequence-based image registration method (Section 6) and for the generation of ventilation maps (Section 8).

#### 4.3.1 Long Short-Term Memory Networks

A major limitation of RNNs is to learn long-term dependencies. By applying the same function on the complete input sequence, a deep computation graph is built. The computation graph contains all operations performed on the input data. It is needed to update the parameters using the gradients during the backpropagation step. Deep computation graphs can suffer from the so-called vanishing and exploding gradient problem [Goodfellow et al. \(2016\)](#). These effects are not specific to RNNs but for all very deep neural networks. For feed forward neural networks the depth of the computation graph depends on the model design. In comparison, for RNNs it depends on the sequence length. The Long Short-Term Memory neural networks (LSTMs) are an extension of the base RNN (4.16), presented by [Hochreiter and Schmidhuber \(1997\)](#), to improve the learning of long-term dependencies. LSTMs belong to the class of gated RNNs and defined by



**Figure 4.11:** Unrolled Long Short-Term Memory neural network.

$$i_t = \phi(W_i x_t + U_i h_{t-1} + b_i), \quad (4.17)$$

$$o_t = \phi(W_o x_t + U_o h_{t-1} + b_o), \quad (4.18)$$

$$c_t = c_{t-1} + i_t \odot \psi(W_c x_t + U_c h_{t-1} + b_c), \quad (4.19)$$

$$h_t = o_t \odot \psi(c_t). \quad (4.20)$$

Here,  $i_t$  the input gate,  $o_t$  the output gate with the corresponding weights  $W$ ,  $U$  and the bias  $b$ . The cell state is defined as  $c_t$  and the output as  $h_t$ . As activation functions the sigmoid function  $\phi(\cdot)$  is used for the gates and for the remaining the hyperbolic tangent  $\psi(\cdot)$  is used. Figure 4.11 shows a graphical representation of (4.17), (4.18), (4.19), and (4.20).

#### 4.3.2 Gated Recurrent Units

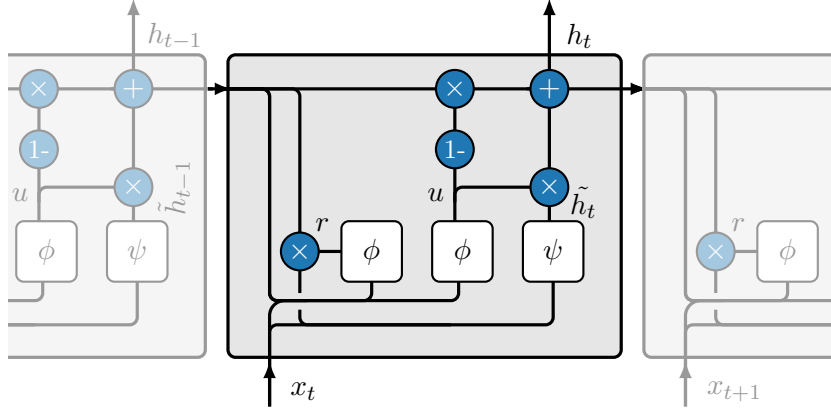
The Gated Recurrent Unit (GRU) was presented by [Cho et al. \(2014\)](#). Similar to LSTMs a GRU contains two gates: the *reset* gate and the *update* gate. The update gate controls the influence of the last state and the current input to the next output state. A GRU cell is defined by

$$r_t = \phi(W_r x_t + U_r h_{t-1}), \quad (4.21)$$

$$u_t = \phi(W_z x_t + U_z h_{t-1}), \quad (4.22)$$

$$\tilde{h}_t = \psi(W x_t + U(r_t \odot h_{t-1})), \quad (4.23)$$

$$h_t = (1 - z) \odot h_{t-1} + u_t \odot \tilde{h}_t. \quad (4.24)$$



**Figure 4.12:** Unrolled gated recurrent unit.

Here,  $r_t$  is the reset gate,  $u_t$  the update gate,  $\tilde{h}_t$  the proposal state, and  $h_t$  the output of the GRU cell at the time  $t$ . Figure 4.12 shows a graphical representation of (4.21), (4.22), (4.23), and (4.24).

## 4.4 Deep Neural Networks

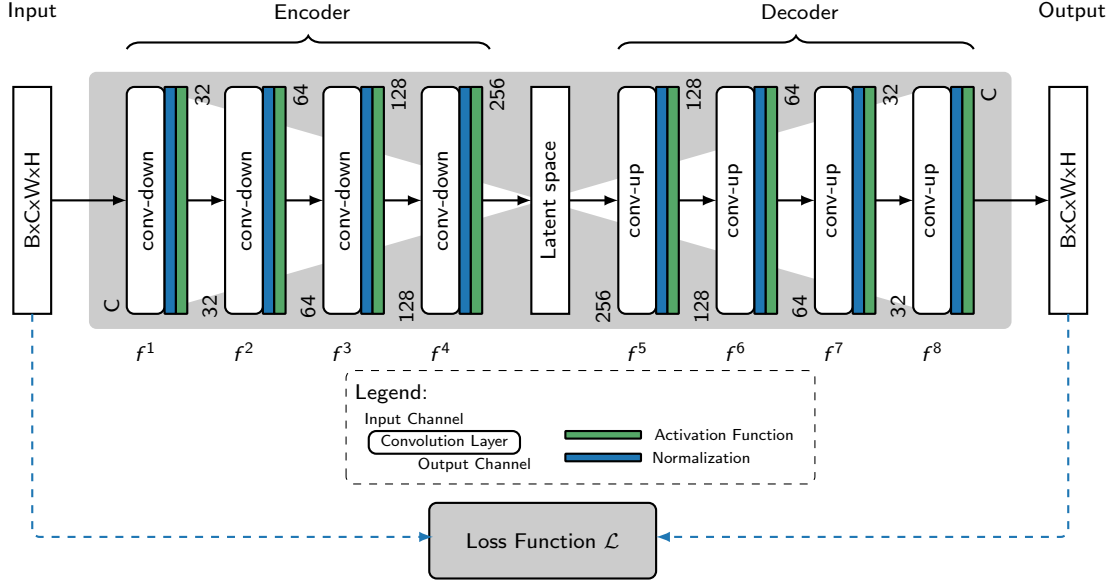
Deep neural network (DNN) models are an essential part of current machine learning methods and consist of one or more hidden layers. An example is the MLP described in Section 4.1.1. In this section, we will describe how to build a DNN for a specific task with the methods described above and further we will introduce additional methods that are used in DNNs in order to improve the performance. Figure 4.13 shows an example of a feed forward deep neural network, i.e., an autoencoder network, with 8 layers  $f^1, \dots, f^8$ . There are two major parts in the presented autoencoder network: the *encoder*  $E$  with the layer  $f_E^1, \dots, f_E^4$  and the *decoder*  $D$  with the layer  $f_D^5, \dots, f_D^8$ . The goal is to train the encoder  $E$  to transform the input image into a low dimensional feature space, i.e., latent space, and then reconstruct the image from this feature space using the decoder  $D$ . An application for such an autoencoder network is image compression. A similar autoencoder structure is used in Section 9 to generate a weakly supervised segmentation method.

For the encoder the output is defined as

$$y_E = E(x, \theta_E) = f_E^4(f_E^3(f_E^2(f_E^1(x, \theta_1), \theta_2), \theta_3), \theta_4) \quad (4.25)$$

and for the decoder as

$$y_D = D(x, \theta_D) = f_D^8(f_D^7(f_D^6(f_D^5(x, \theta_5), \theta_6), \theta_7), \theta_8), \quad (4.26)$$



**Figure 4.13:** Deep neural network architecture example of a so-called autoencoder network for dimension reduction for image compression. Conv-down blocks reducing the spatial resolution in each layer by either a strided convolution or a pooling operation. Conv-up blocks increase the spatial dimension by using bilinear interpolation or transposed convolutions. (B: Batch Size, C: Channels, W: Image width, H: Image height)

where  $\theta_l$  are the learnable weights of the  $l$ -th layer. The final autoencoder is then defined as

$$y = A(x, \theta) = D(E(x, \theta_E), \theta_D). \quad (4.27)$$

Each layer in the encoder  $E$  is defined as

$$f_E^l(x) = \psi(g_{\text{norm}}(g_{\text{conv}}(x, s, \theta_{\text{conv}}^l), \theta_{\text{norm}}^l)), \quad (4.28)$$

where  $g_{\text{conv}}$  is the strided convolution with the kernel parameter  $\theta_{\text{conv}}^l$  and the stride  $s$  in order to reduce the spatial resolution of the input. It is also possible to use pooling operation as maximum pooling or average pooling in order to reduce the spatial resolution. The output of the convolution is then passed to the normalization  $g_{\text{norm}}$  with the learnable parameter  $\theta_{\text{norm}}^l$ . Different normalization methods for neural networks have been presented in the literature. The common normalization methods are batch-normalization Ioffe and Szegedy (2015), instance normalization Ulyanov et al. (2016), or group normalization Wu and He (2019). The intention of the normalization is to keep the distribution of the features close to normal distribution  $\mathcal{N}(0, 1)$ . However, the normalization is not mandatory to build a deep neural network. Finally, the results are passed to the activation function  $\psi(\cdot)$ .

For the decoder  $D$  each layer is defined as

$$f_D^l(x) = \psi(g_{\text{norm}}(g_{\text{conv}}(g_{\text{up}}(x), \theta_{\text{conv}}^l), \theta_{\text{norm}}^l)), \quad (4.29)$$



where  $g_{\text{up}}$  is a bilinear up-sampling method to increase the spacial resolution, but also transposed convolutions could be used.

#### Parameter Gradient Calculation and Update (Training)

In order to find the best parameter setting  $\theta^*$  of the encoder and the decoder to compress and reconstruct the image, the network needs to be trained. For the training a scalar-valued loss function  $\mathcal{L}(x, A(x, \theta))$  is needed. The aim of this loss function is to compare the input of the network  $x$  with the output of the network. For the autoencoder example, the loss function can be seen as an image similarity measure described in Section 3.6. With this it is possible to update the parameter  $\theta$  of the network using the backpropagation algorithm by calculating the gradient of each parameter

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \left( \frac{\partial \mathcal{L}(x, A(x, \theta))}{\partial \theta_1}, \frac{\partial \mathcal{L}(x, A(x, \theta))}{\partial \theta_2}, \dots, \frac{\partial \mathcal{L}(x, A(x, \theta))}{\partial \theta_L} \right). \quad (4.30)$$

As shown in (4.27) deep neural networks are a composition of several functions. The mathematical method to calculate the gradient is the chain rule. The backpropagation algorithm presented by [Parker \(1985\)](#); [Rumelhart et al. \(1985\)](#); [Werbos \(1974\)](#) is a method for the fast computation of the analytic gradient of each parameter in the network by reducing the number of repeated calculations. Finding the best parameter setting  $\theta^*$  for the model that reduces the loss function  $\mathcal{L}$  can be written as a minimization problem of the form

$$\theta^* = \arg \min_{\theta} \sum_{x_i \in \mathcal{D}} \mathcal{L}(x_i, A(x_i, \theta)), \quad (4.31)$$

where  $x_i$  is an element of the training data set  $\mathcal{D}$ . Since the gradient of each parameter can be calculated, gradient based optimization methods like, e.g., the stochastic gradient descent, can be used. Several optimization methods have been presented in the past to improve the optimization process. Examples are the ADAM optimizer [Kingma and Ba \(2014\)](#), the ADADELTA optimizer [Zeiler \(2012\)](#), or the Adagrad optimizer [Duchi et al. \(2011\)](#).

#### 4.4.1 Deep Neural Networks for Semantic Segmentation

Deep neural networks have shown huge success in several tasks of machine learning problems. For example, in image classification with AlexNet [Krizhevsky et al. \(2012\)](#), GoogleNet [Szegedy et al. \(2015\)](#), or ResNet [He et al. \(2016\)](#) and in image generation with the presentation of GANs [Goodfellow et al. \(2014\)](#). The aim of semantic segmentation is to perform a pixel-wise classification instead of a classification of the whole image. [Long et al. \(2015\)](#) presented a method for a pixel-wise segmentation based on CNNs. Combining an encoder and a decoder part to obtain a pixel-wise classification was presented in [Noh et al. \(2015\)](#); [Ronneberger et al. \(2015\)](#). All of these methods have in common that they are trained in a *fully supervised* manner. This means for each training image a corresponding pixel-wise ground-truth label is needed. However, obtaining

this ground-truth segmentation is a time consuming and costly process, especially in the field of medical image processing. In order to reduce the number of labeled data, *semi supervised* methods were presented. Semi supervised methods are able to learn to generate pixel-wise segmentation with only a limited number of pixel-wise annotated data in combination with unlabeled data. Examples of such methods were presented in [Hong et al. \(2015\)](#); [Papandreou et al. \(2015a\)](#); [Souly et al. \(2017\)](#). However, for some training data, a manual classification on the pixel level is still needed for the training. In contrast to fully or semi supervised methods, *weakly supervised* methods only use a global label per image. [Andermatt et al. \(2019\)](#); [Vorontsov et al. \(2019\)](#) used domain translation to perform the segmentation using only the domain label. Methods that use no supervision for training are called *unsupervised* methods. Here, no labeled data is used during training. Unsupervised methods are used for medical segmentation tasks are also known as *anomaly detection* or out of distribution detection (ODD). For the anomaly detection or ODD a network, for example, an autoencoder is trained to reconstruct images of a specific domain for example brain images of healthy subjects. If an image that is not part of the trained distribution is passed through the network, the network should not be able to reconstruct the image properly. Comparing the input with the out shows the difference, i.e., the anomaly contained in the image [Baur et al. \(2018\)](#); [Schlegl et al. \(2017\)](#); [Zimmerer et al. \(2019, 2018\)](#).

# Adaptive Graph Diffusion Regularisation for Discontinuity Preserving Image Registration

In the following paper, we present a method for the registration of 2D and 3D image data. We focus on the problem of the registration of images containing sliding-organ motion and therefore discontinuities in the displacement field. Our presented method is based on graph diffusion. The results show that our method is able to register images that contain sliding organ boundaries and that it preserves the discontinuities of the transformations.

## Publication

The paper was presented at the 8th International Workshop on Biomedical Image Registration (WBIR) in Leiden, Netherlands, 2018<sup>1</sup>.

---

<sup>1</sup>[https://link.springer.com/chapter/10.1007/978-3-319-92258-4\\_3](https://link.springer.com/chapter/10.1007/978-3-319-92258-4_3)



# Adaptive Graph Diffusion Regularisation for Discontinuity Preserving Image Registration

Robin Sandkühler<sup>1</sup>, Christoph Jud<sup>1</sup>, Simon Pezold<sup>1</sup>, Philippe C. Cattin<sup>1</sup>

## Abstract

Registration of thoracic images is central when studying for example physiological changes of the lung. Due to sliding organ motion and intensity changes based on respiration the registration of thoracic images is challenging. We present a novel regularisation method based on adaptive anisotropic graph diffusion. Without the need of a mask it preserves discontinuities of the transformation at sliding organ boundaries and enforces smoothness in areas with similar motion. The graph diffusion regularisation provides a direct way to achieve anisotropic diffusion at sliding organ boundaries by reducing the weight of corresponding edges in the graph which cross the sliding interfaces. Since the graph diffusion is defined by the edge weights of the graph, we develop an adaptive edge weight function to detect sliding boundaries. We implement the adaptive graph diffusion regularisation method in the Demons registration framework. The presented method is tested on synthetic 2D images and on the public 4D-CT DIR-Lab data set, where we are able to correctly detect the sliding organ boundaries.

## 5.1 Introduction

The registration of images of the human thorax is essential for the analysis or the monitoring of physiological properties of the upper abdomen. However, thoracic images are affected by sliding organ motion at the thoracic cavity, and corresponding mass points undergo cyclic intensity changes over the respiratory cycle. If images are affected by sliding organ motion the global smoothness assumption of the transformation often does not hold, because of local discontinuities in the transformation at these boundaries.

Several registration approaches have been presented to overcome the trade-off between global smoothness and local discontinuity preservation. Based on their definition parametric approaches are more likely to achieve global smoothness, if the chosen basis function is smooth. In order to preserve local discontinuities with a parametric approach, a stationary first order B-spline kernel combined with a Total Variation (TV) regularisation is introduced in [Vishnevskiy et al. \(2016\)](#). A non-stationary kernel is described in [Jud et al. \(2016a\)](#). Here, a smooth kernel is locally adapted in its shape based on the image intensities. Non-parametric approaches are well suited for local discontinuity preservation, as they directly estimate the transformation for each pixel. Different direction-dependent regularisation methods based on the image intensities or the transformation are presented in [Demirovic et al. \(2009\)](#); [Nagel and Enkelmann \(1986\)](#);

---

<sup>1</sup>Department of Biomedical Engineering, University of Basel, Allschwil, Switzerland

Schmidt-Richberg et al. (2012). In Papież et al. (2014) an adaptive Gaussian regularisation based on bilateral filtering is shown. Segmentation of the sliding boundaries can also be used by either building a motion segmentation Kiriyanthan et al. (2016) during the registration, or assuming a prior segmentation to locally adapt the transformation model Hua et al. (2017). Several graph based approaches have been introduced in the past. Using a graph based formulation for TV regularisation Bagnato et al. (2009) or a non-local regularisation based on the minimum spanning tree of a graph Papież et al. (2017). In both methods the image intensities are used to calculate the edge weights of the graph.

As mentioned before parametric approaches are well suited for global smoothness, and non-parametric regularisation, especially anisotropic diffusion Demirovic et al. (2009); Nagel and Enkelmann (1986); Schmidt-Richberg et al. (2012), are effective to preserve local discontinuities. In order to achieve both, we propose the graph diffusion as regularisation method for image registration. Graph diffusion allows global smoothness while at the same time local differences on the pixel level are considered. It was shown to be a valid regularisation operator for kernel-based learning algorithms Smola and Kondor (2003) and a reliable and computationally efficient method in the area of edge-preserving image smoothing Zhang and Hancock (2008). Anisotropic diffusion can be achieved with graph diffusion in a straightforward way by modifying the edge weight between nodes. In order to achieve anisotropic diffusion at local discontinuities, we need to reduce the edge weights of nodes which are located on different sites of a sliding organ boundary.

In this work, we present the adaptive anisotropic graph diffusion regularisation method (A<sup>2</sup>GD) to enforce global smoothness and preserve local discontinuities of the transformation. We achieve this without prior information (e.g. segmentation) of the sliding organ boundaries. A local adaptive edge weight function is developed to create anisotropic diffusion only at sliding organ interfaces and isotropic diffusion in areas with similar motion. The proposed regularisation is implemented in the Demons framework Thirion (1998) and replaces the isotropic diffusion regularisation. To the extent of our knowledge, graph diffusion has not been used as regularisation method for image registration before.

## 5.2 Background

Let  $T, R : \mathcal{X} \rightarrow \mathbb{R}$  be the target and reference image over the image domain  $\mathcal{X}$ . We define the image domain  $\mathcal{X} = \{x_i\}_{i=1}^n$  as a set of regular distributed grid points  $x_i$  in  $d$ -dimensions. The registration problem can be defined as a regularised minimisation problem:

$$f = \arg \min_u \mathcal{S}[T, R_u] + \varphi \mathcal{R}[u], \quad (5.1)$$

where transformation of interest  $f : \mathcal{X} \rightarrow \mathbb{R}^d$  is a minimiser of (5.1). Here,  $\mathcal{S}[\cdot, \cdot]$  is the similarity measure for the target image  $T$  and the transformed reference image  $R_f$  with  $R_f(x_i) = R(x_i + f(x_i))$ . In order to restrict the space of admissible transformations,

prior knowledge of the transformation space (e.g. smoothness) can be applied to the registration problem by the regularisation term  $\mathcal{R}[\cdot]$ . The parameter  $\varphi$  controls the influence of the regularisation e.g. the smoothness of the transformation. A possible method to find a transformation that minimises (5.1) is the Demons method proposed by Thirion [Thirion \(1998\)](#). He proposed an iterative method to determine  $f$  by alternating minimising the similarity  $\mathcal{S}$  and computing the regularisation  $\mathcal{R}$ . A generalisation of this idea is shown in Algorithm 2. Thirion proposed an isotropic diffusion process as

---

**Algorithm 2** Demons registration framework

---

1: **Inputs:**

$T, R, N$  := number of iterations,  $\sigma$  := kernel size,  $\alpha$  := step size

2: **Initialise:**

$f \leftarrow 0$

3: **for**  $i = 1$  to  $N$  **do**

4:    $s \leftarrow \nabla S[T, R_f]$  compute image force (demons)

5:    $f \leftarrow f + \alpha s$  update transformation

6:    $f \leftarrow K_\sigma * f$  smooth transformation

---

regularisation to smooth the transformation. Isotropic diffusion can be achieved by

$$\mathcal{R}_{\text{ID}}[f] = K_\varphi^G * f^l \quad l = 1, \dots, d \quad (5.2)$$

for each spatial dimension  $l$ . Here,  $*$  is the convolution and  $K_\varphi^G$  is a Gaussian kernel (diffusion kernel) with a kernel size of  $\sqrt{2\varphi}$  [Babaud et al. \(1986\)](#).

### 5.3 Method

We summarise in the following the definition of diffusion on graphs, how it can be calculated efficiently, and how we use it as regularisation for image registration. Further, we propose our extension the adaptive anisotropic graph diffusion regulariser ( $A^2\text{GD}$ ), and an edge weight function to detect sliding organ boundaries.

#### Adaptive Graph Diffusion Regularisation

In order to apply graph diffusion as regularisation  $\mathcal{R}_{\text{GD}}[f]$ , the transformation  $f$  is modelled as an undirected weighted grid graph  $\mathcal{G} = (V, E, W)$  with  $n$  nodes. Each node  $v_i \in V$ , with the node position  $x_i \in \mathcal{X}$ , represents  $f(x_i)$ . The set of edges is given as  $E \subseteq V \times V$ . An edge  $e_{ij} \in E$  connects the nodes  $v_i$  and  $v_j$ . The weight matrix  $W \in \mathbb{R}^{n \times n}$  contains the edge weights of the graph with  $W(i, j) = w(e_{ij})$  and  $w : E \rightarrow [0, 1]$ . A central element in spectral graph theory is the graph Laplacian matrix  $L \in \mathbb{R}^{n \times n}$ . The Laplacian matrix is a symmetric matrix and is defined as  $L = D - W$

## 5 Adaptive Graph Diffusion Regularisation

with

$$L(i, j) = \begin{cases} D(i, j) - W(i, j), & \text{if } i = j \\ -W(i, j), & \text{if } e_{ij} \in E \\ 0, & \text{otherwise,} \end{cases} \quad (5.3)$$

where  $D \in \mathbb{R}^{n \times n}$  is the diagonal degree matrix with  $D(i, i) = \sum_{j=1}^n W(i, j)$ .

The diffusion rate or the flow between two nodes is determined by the weight of the edge between both nodes. This allows to define arbitrary anisotropic diffusion by only modifying the edge weights of the graph. In order to define the diffusion process on graphs, Equation (5.2) can be rewritten as

$$\mathcal{R}_{\text{GD}}[f] = K_{\varphi}^{\text{GD}} \bar{f}^l = \exp(-\varphi L) \bar{f}^l \quad l = 1, \dots, d. \quad (5.4)$$

Here,  $\bar{f}^i \in \mathbb{R}^{n \times 1}$  is the column vector representation of the transformation  $f$  and  $K_{\varphi}^{\text{GD}} \in \mathbb{R}^{n \times n}$  is the graph diffusion kernel [Kondor and Lafferty \(2002\)](#); [Zhang and Hancock \(2008\)](#) with the matrix exponential  $\exp(\cdot)$ . If the edge weights are set according to the node position  $x_i$  and  $x_j$  with  $w(e_{ij}) = \exp(-\|x_i - x_j\|^2/4\varphi)$ , then (5.2) and (5.4) will provide the same results. We refer the reader to [Zhang and Hancock \(2008\)](#) for a more detailed description of graph diffusion.

In order to achieve an adaptive graph diffusion regularisation we make the graph diffusion kernel  $K_{\varphi}^{\text{GD}}$  non-static. We do this by updating the edge weights in the weight matrix  $W_k$  in each iteration  $k$  with the edge weight function presented below. The final adaptive graph diffusion regularisation is defined as

$$\mathcal{R}_{A^2\text{GD}}[f_{k+1}, W_k] = K_{\varphi}^{\text{GD}}(W_k) \bar{f}_{k+1}^l = \exp(-\varphi L(W_k)) \bar{f}_{k+1}^l \quad l = 1, \dots, d. \quad (5.5)$$

### Computation of the graph diffusion

Calculating the matrix exponential of the Laplace matrix  $L$  is the major computational challenge for the graph diffusion. For the matrix  $L$ , the matrix exponential can be defined as

$$\exp(L) = \sum_{k=0}^{\infty} \frac{1}{k!} L^k. \quad (5.6)$$

Different approaches exist to compute the matrix exponential [Moler and Loan \(2003\)](#). The graph Laplacian matrix  $L$  can be of high order, therefore the computation becomes costly. However, the explicit calculation of the matrix exponential is not required for the graph diffusion. In order to compute the graph diffusion only the action of the graph diffusion kernel to the transformation vector is needed (5.4). It has been shown in [Saad \(1992\)](#) that those kind of actions can be efficiently approximated with the Krylov subspace projection methods. Therefore,  $\exp(L)f$  is approximated by an element of the Krylov subspace  $K_m(L, f) = \text{span}\{f, Lf, L^2f, \dots, L^{m-1}f\}$ , where  $m$  is the dimension of the Krylov subspace. Normally the Krylov space dimension ( $m < 50$ ) is much smaller compared to  $n$  which can be in the range of a few million in case of 3D registration. Since



$L$  is a hermitian matrix, the Lanczos algorithm [Lanczos \(1950\)](#) offers a computationally efficient way to find the approximating element of  $K_m(L, f)$ . The final approximation of (5.4) is then given as

$$\bar{f}_{\text{smooth}}^l = \exp(-\varphi L) \bar{f}^l \approx \|\bar{f}^l\|_2 P \exp(Q) e_1. \quad (5.7)$$

Here,  $P \in \mathbb{R}^{n \times m}$  is the projection matrix,  $Q \in \mathbb{R}^{m \times m}$  a symmetric tridiagonal matrix, and  $e_1$  is the first unit vector. Both matrices  $P$  and  $Q$  are the results of the Lanczos algorithm. The approximation only requires the matrix exponential of a matrix with the order of  $m$  instead of the order of  $n$ . Since  $Q$  is a symmetric tridiagonal matrix, we compute the matrix exponential of  $Q$  by  $\exp(Q) = \Lambda \exp(\Gamma) \Lambda^{-1}$ . Each column in  $\Lambda$  is an eigenvector of  $Q$  and  $\Gamma$  is a diagonal matrix of the corresponding eigenvalues. The matrix exponential of  $\Gamma$  is the exponential of each diagonal element of  $\Gamma$ .

#### Local edge update

The graph diffusion allows an anisotropic diffusion process by modifying corresponding edge weights. However, detecting the corresponding edges only in one information domain (e.g. image intensities) is challenging, since sliding organ boundaries not always correspond to intensity differences (see Figure 5.1, Case (I)). We define five cases (see Figure 5.1) to adapt the edge weights either for anisotropic or isotropic diffusion. In Case (I) and (II), the edges crosses a sliding organ boundary and therefore their edge weights need to be zero to enforce anisotropic diffusion. The other edges are either inside the same organ (Case (III) and (V)) or crossing an organ boundary where both organs move similarly (Case (IV)). In cases (III) to (V) isotropic diffusion is desired and therefore the edge weights should be one. We propose a locally adaptive edge weight function to cover all five cases based on three different information domains. The first two are the image intensity domain (Case III, V)

$$w_{\text{img}}(e_{ij}) = \exp(-\delta_{\text{img}} \|T(x_i) - T(x_j)\|), \quad (5.8)$$

and the transformation domain (Case I-V)

$$w_{\text{f}}(e_{ij}) = \exp(-\delta_{\text{f}} \|f(x_i) - f(x_j)\|), \quad (5.9)$$

with the scaling parameter  $\delta_{\text{img}}$  and  $\delta_{\text{f}}$ . The third domain is described by the relation between the direction of the image gradient and the direction of the transformation (Case II, IV) based on the Nagel-Enkelmann operator [Nagel and Enkelmann \(1986\)](#). Adapting the pixel-wise Nagel-Enkelmann operator to an edge weight function for the graph diffusion results in

$$w_{\perp}(e_{ij}) = 0.5 \left( \frac{\|\nabla I_{\text{max}}(e_{ij}) f(x_i)^T\|}{\|\nabla I_{\text{max}}(e_{ij})\| \|f(x_i)\|} + \frac{\|\nabla I_{\text{max}}(e_{ij}) f(x_j)^T\|}{\|\nabla I_{\text{max}}(e_{ij})\| \|f(x_j)\|} \right) \quad (5.10)$$

with

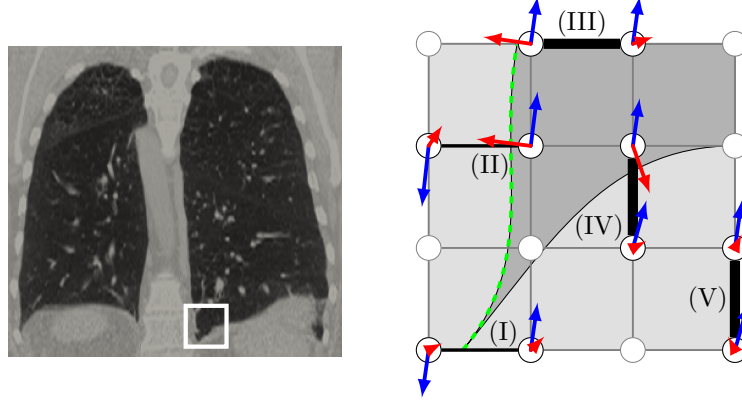
$$\nabla I_{\text{max}}(e_{i,j}) = \begin{cases} \nabla T(x_i), & \|\nabla T(x_i)\| \geq \|\nabla T(x_j)\| \\ \nabla T(x_j), & \text{otherwise.} \end{cases} \quad (5.11)$$

## 5 Adaptive Graph Diffusion Regularisation

The final weight function is then given as

$$w(e_{ij}) = \tau w_f(e_{ij}) + (1 - \tau)[(w_{\text{img}}(e_{ij})w_f(e_{ij}) + (1 - w_{\text{img}})w_{\perp}(e_{ij})]. \quad (5.12)$$

The scale function  $\tau \in [0, 1]$  in (5.12) allows to adapt the scale between the image based domains (5.8) and (5.10) and the transformation domain (5.9).



**Figure 5.1:** Different cases, which are considered for the edge weight adaptation at sliding interfaces (dashed green) based on image gradients (red), transformation (blue) and intensity differences (gray values).

The final edge weights are thresholded with

$$\bar{w}_k(e_{ij}) = \begin{cases} 0, & w_k(e_{ij}) < 0.5 \\ 1, & \text{else} \end{cases}. \quad (5.13)$$

In order to reduce the effect of oscillating edge weights an exponential smoothing

$$W_{k+1}(i, j) = \rho W_k(i, j) + (1 - \rho)\bar{w}_{k+1}(e_{ij}) \quad (5.14)$$

is applied, where  $\rho$  is the exponential decay rate.

### Node Isolation

A node  $v_i$  is isolated from the graph, if all edges connected to this node have a zero weight  $D(i, i) = 0$ . Isolated nodes can cause artefacts in the transformation, because they are excluded from the regularisation. In order to prevent node isolation, all weights of the edges connected to a node  $v_i$  with  $D(i, i) < \beta$  will be set to one. We choose  $\beta \leq 2$  in 2D and  $\beta \leq 3$  in 3D, so that the reset will not affect nodes at sliding boundaries.

### Transformation Update

The update of the transformation in the Demons framework can be written in the form of a gradient descent update

$$f_{k+1} = f_k + \eta_k \nabla \mathcal{S}[T, R_{f_k}]. \quad (5.15)$$

In order to improve the convergence of the Demons framework we replace (5.15) by the well known Momentum optimiser equations

$$b_{k+1} = \alpha_k v_k + \eta_k \nabla \mathcal{S}[T, R_{f_k}] \quad (5.16)$$

$$f_{k+1} = f_k + b_{k+1} \quad (5.17)$$

as introduced in Santos-Ribeiro et al. (2016). We set  $\eta_k = 1/(\|\nabla R_{f_k}\|^2 + \psi_k)$  according to Thirion (1998), where  $\psi_k \in \mathbb{R}_{>0}$  allows a modification of the step size in each iteration.

## 5.4 Results

We evaluated our adaptive graph diffusion regularisation method on synthetic 2D images with given ground truth, and on the public available DIR-Lab data sets.

### Synthetic Experiments

We define a target image  $T$  with a size of  $256 \times 256$  pixels and the intensities  $T(x) = x \times e_1/\pi$ , where  $e_1$  is a unit vector (Figure 5.2a). The inner part of the target image is rotated by 15 degrees and the outer part is rotated in the opposite direction, in order to simulate a sliding boundary (Figure 5.2b).

The  $\mathcal{S}_{\text{MSE}}[T, R_f] = 1/n \sum_{i=1}^n (T(x_i) - R_f(x))^2$  metric is used as similarity measure. We choose  $w(e_{ij}) = 0.5 \cdot (1 + \|f(x_i)^T f(x_j)\|_2 / (\|f(x_i)\|_2 \|f(x_j)\|_2))$  as edge weight function, because the sliding organ boundary in the image can be mainly described by direction differences of the displacement. Further, we used a multi-scale approach with three resolutions  $\{64, 128, 256\}$ . The regularisation parameter are set to  $\varphi_0 = \{4, 8, 13\}$  and  $\varphi_{\min} = \{1, 2, 2\}$ . In each iteration the current regularisation weight is given by  $\varphi_k = \varphi_0 * \exp(-0.05k) + \varphi_{\min}$ . All remaining parameters are set as follows:  $m = 30$ ,  $\psi = 0.01$ ,  $\rho = 0.9$ ,  $\alpha = 0.9$ , and  $N = \{100, 100, 200\}$  iterations. All edge weights  $w(e_{ij})$  are initialised with one.

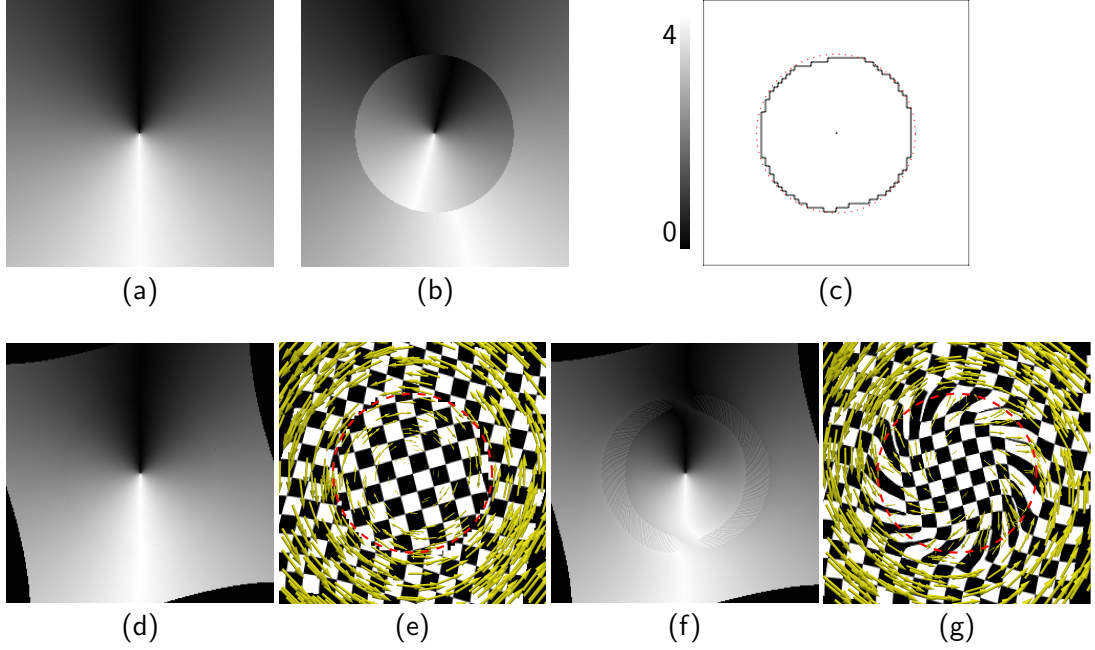
The results of the proposed regularisation method for the synthetic experiments are shown in Figure 5.2d with edge update and in Figure 5.2f without edge update. With edge updates during the registration the ground truth displacement is very well estimated. The sliding boundary is clearly detected as we can see a reduction of the degree matrix values only at the sliding organ boundary (Figure 5.2c). If we apply the transformation result from the method without the edge update to a checker board, we get strong distortion at the sliding boundary (Figure 5.2g). Compared to this, our presented method reduce the distortion at sliding organ boundaries to a minimum (Figure 5.2e).

### DIR-Lab Data Set

The publicly available DIR-Lab<sup>2</sup> data set contains 10 4D-CT image series of different individuals. All images have a size between  $256 \times 256 \times 94$  voxel and  $512 \times 512 \times 136$

---

<sup>2</sup><http://www.dir-lab.com>



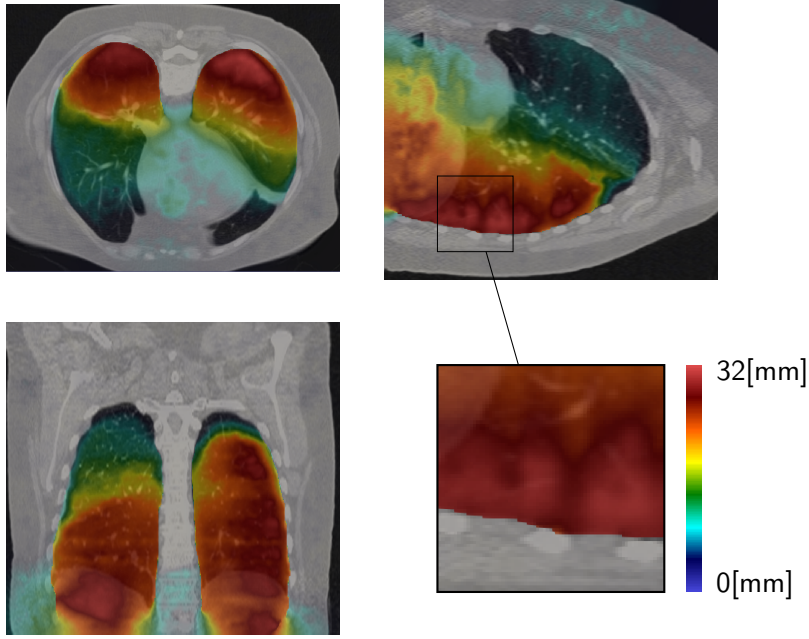
**Figure 5.2:** Target image (a) and reference image (b) for the synthetic experiment. The red circle shows the sliding organ boundary between the inner and the outer region. Registration result with edge update (d) and the corresponding warped checker board (e), and without edge update (f) and the corresponding warped checker board (g). The graph degree matrix at the end of the registration is shown in (c).

voxel, with a voxel size in the range of  $0.97 \times 0.97 \times 2.5 \text{ mm}^3$  and  $1.16 \times 1.16 \times 2.5 \text{ mm}^3$ . For all images we clip the intensities between 50 HU and 1200 HU, scale them in a range of  $[0, 1]$ , and resample all to a voxel size of  $1 \times 1 \times 1 \text{ mm}^3$ . For the evaluation of the registration result, the DIR-Lab data set provides 300 landmarks for the maximal inhalation and maximal exhalation of the breathing cycle. As images similarity measure the normalised local cross correlation  $\mathcal{S}_{\text{lcc}}$  is used. We use the derivative approximation of  $\mathcal{S}_{\text{lcc}}$  introduced in Cachier and Pennec (2000).

**Table 5.1:** Mean snap to voxel TRE in millimetre based on the 300 landmarks of DIR-Lab data set.

Case	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	mean
No Reg.	3.87	4.34	6.96	9.86	7.51	10.93	11.03	15.01	7.94	7.33	8.48
MST Papież et al. (2017)	<b>0.83</b>	<b>0.87</b>	<b>1.10</b>	1.96	<b>1.36</b>	1.77	1.58	2.08	1.50	1.40	1.44
A <sup>2</sup> GD	1.12	1.20	1.27	<b>1.61</b>	1.53	<b>1.32</b>	<b>1.38</b>	<b>1.45</b>	<b>1.36</b>	<b>1.34</b>	<b>1.36</b>

A multi-scale approach with four scales is used for the registration. At the end of each scale level, the transformation result is projected to the next scale level as initial value. All parameters are set as follows:  $N = \{200, 200, 200, 300\}$ ,  $m = 30$ ,  $\rho = \{0.9\}$ ,



**Figure 5.3:** Displacement field of the #8 case of the DIR-Lab data set.

$\sigma_{\text{lcc}} = \{2, 8, 10, 15\}$ ,  $\delta_{\text{img}} = 5$ ,  $\delta_f = \{1, 0.5, 0.25, 0.125\}$ ,  $\psi = \{0.5, 0.3, 0.2, 0.2\}$  and increased with  $10^{-3}$ , The regularisation parameter is set to  $\varphi = \{2, 3, 3, 3\}$ . We set  $\tau_0 = 0$  and increase it linear with a factor of  $10^{-5}$  in each iteration. With this configuration the average run time is 5 minutes for one image pair on a GPU.

In Table 5.1 the mean TRE for all 10 data sets based on the 300 landmarks is shown. We compare our method to the MST based graph regularisation method [Papież et al. \(2017\)](#). As shown in Figure 5.3 our method is able achieve global smoothness of the transformation and preserves discontinuities at sliding organ boundaries.

## 5.5 Conclusion

We presented a novel regularisation method based on adaptive anisotropic graph diffusion. Without the need of a prior segmentation of the sliding organ boundaries our proposed regularisation method enforces global smoothness and preserves local discontinuities. In order to achieve anisotropic diffusion at sliding organ boundaries we developed an adaptive edge weight function based on local image intensities and the transformation. The results show that we are able to well detect the sliding organ boundaries and preserve the discontinuities in the transformation for the synthetic examples and for the DIR-Lab data set. We achieve a sub-millimetre difference, if we compare our TRE results to results of state of the art methods.

## Acknowledgement

The authors would like to thank the Swiss National Science Foundation for funding this project (SNF 320030\_149576).

# Recurrent Registration Neural Networks for Deformable Image Registration

In the following paper, we present a learning-based registration method. The registration method is inspired by how a human would register two images using only local transformations. We reformulate the registration problem as a sequence of local deformations. In order to model this sequence, we based our model on gated recurrent neural networks, called *Recurrent Registration Neural Networks* (R2N2). With a sequence-based model, we are able to apply local transformations until both images are properly aligned. The network should therefore learn, where both images are not correctly spatially aligned and how to align them properly. The results show that our method is able to register two images with the same accuracy as the B-spline method, but 15 times faster.

## Publication

This paper was presented at the Conference for Neural Information Processing Systems (NeurIPS), Vancouver, Canada, 2019<sup>1</sup>.

---

<sup>1</sup><https://papers.nips.cc/paper/9080-recurrent-registration-neural-networks-for-deformable-image-registration>





# Recurrent Registration Neural Networks for Deformable Image Registration

Robin Sandkühler<sup>1</sup>, Simon Andermatt<sup>1</sup>, Grzegorz Bauman<sup>2</sup>, Sylvia Nyilas<sup>3</sup>,  
Christoph Jud<sup>1</sup>, Philippe C. Cattin<sup>1</sup>

## Abstract

Parametric spatial transformation models have been successfully applied to image registration tasks. In such models, the transformation of interest is parameterized by a fixed set of basis functions as for example B-splines. Each basis function is located on a fixed regular grid position among the image domain because the transformation of interest is not known in advance. As a consequence, not all basis functions will necessarily contribute to the final transformation which results in a non-compact representation of the transformation. We reformulate the pairwise registration problem as a recursive sequence of successive alignments. For each element in the sequence, a local deformation defined by its position, shape, and weight is computed by our recurrent registration neural network. The sum of all local deformations yield the final spatial alignment of both images. Formulating the registration problem in this way allows the network to detect non-aligned regions in the images and to learn how to locally refine the registration properly. In contrast to current non-sequence-based registration methods, our approach iteratively applies local spatial deformations to the images until the desired registration accuracy is achieved. We trained our network on 2D magnetic resonance images of the lung and compared our method to a standard parametric B-spline registration. The experiments show, that our method performs on par for the accuracy but yields a more compact representation of the transformation. Furthermore, we achieve a speedup of around 15 compared to the B-spline registration.

## 6.1 Introduction

Image registration is essential for medical image analysis methods, where corresponding anatomical structures in two or more images need to be spatially aligned. The misalignment often occurs in images from the same structure between different imaging modalities (CT, SPECT, MRI) or during the acquisition of dynamic time series (2D+t, 4D). An overview of registration methods and their different categories is given in [Sotiras et al. \(2013\)](#). In this work, we will focus on parametric transformation models in combination with learning-based registration methods. There are mainly two major classes

---

<sup>1</sup>Department of Biomedical Engineering, University of Basel, Allschwil, Switzerland

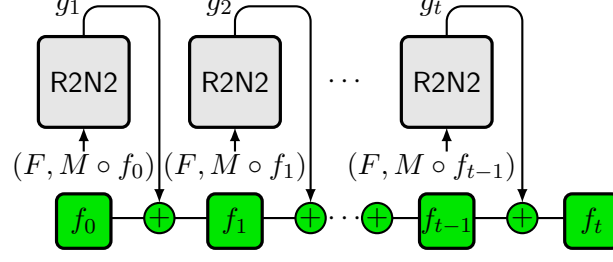
<sup>2</sup>Division of Radiological Physics, Department of Radiology, University of Basel Hospital, Basel, Switzerland

<sup>3</sup>Pediatric Respiratory Medicine, Department of Pediatrics, Inselspital, Bern University Hospital, University of Bern, Bern, Switzerland

of parametric transformation models used in medical image registration. The first class are the dense transformation models or so-called optical-flow [Horn and Schunck \(1981\)](#). Here, the transformation of each pixel in the image is directly estimated (Figure 6.2a). The second class of models are interpolating transformation models (Figure 6.2b). Interpolating transformation models approximate the transformation between both images with a set of fixed basis functions (e.g. Gaussian, B-spline) among a fixed grid of the image domain [Jud et al. \(2016a,b\)](#); [Rueckert et al. \(1999\)](#); [Vishnevskiy et al. \(2016\)](#). These models reduce the number of free parameters for the optimization, but restrict the space of admissible transformations. Both transformation models have advantages and disadvantages. Dense models allow preservation of local discontinuities of the transformation, while the interpolating models achieve a global smoothness if the chosen basis function is smooth.

Although the computation time for the registration has been reduced in the past, image registration is still computationally costly, because a non-linear optimization problem needs to be solved for each pair of images. In order to reduce the computation time and to increase the accuracy of the registration result, learning-based registration methods have been recently introduced. As the registration is now separated in a training and an inference part, a major advantage in computation time for the registration is achieved. A detailed overview of deep learning methods for image registration is given in [Haskins et al. \(2019\)](#). The FlowNet [Dosovitskiy et al. \(2015\)](#) uses a convolutional neural network (CNN) to learn the optical flow between two input images. They trained their network in a supervised fashion using ground-truth transformations from synthetic data sets. Based on the idea of the spatial transformer networks [Jaderberg et al. \(2015\)](#), unsupervised learning-based registration methods were introduced [Dalca et al. \(2018b\)](#); [de Vos et al. \(2017\)](#); [Hu et al. \(2018\)](#); [Stergios et al. \(2018\)](#). All of these methods have in common that the output of the network is directly the final transformation. In contrast, sequence-based methods do not estimate the final transformation in one step but rather in a series of transformations based on observations of the previous transformation result. This process is iteratively continued until the desired accuracy is achieved. Applying a sequence of local or global deformations is inspired by how a human would align two images by applying a sequence of local or global deformations. Sequence-based methods for rigid [Liao et al. \(2017\)](#); [Miao et al. \(2018\)](#) and for deformable [Krebs et al. \(2017\)](#) registration using reinforcement learning methods were introduced in the past. However, the action space for deformable image registration can be very large and the training of deep reinforcement learning methods is still very challenging.

In this work, we present the *Recurrent Registration Neural Network* (R2N2), a novel sequence-based registration method for deformable image registration. Figure 6.1 shows the registration process with the R2N2. Instead of learning the transformation as a whole, we iteratively apply a network to detect local differences between two images and determine how to align them using a parameterized local deformation. Modeling the final transformation of interest as a sequence of local parametric transformations instead of a fixed set of basis functions enables our method to extend the space of admissible transformations, and to achieve a global smoothness. Furthermore, we are



**Figure 6.1:** Sequence-based registration process for pairwise deformable image registration of a fixed image  $F$  and a moving image  $M$ .

able to achieve a compact representation of the final transformation. As we define the resulting transformation as a recursive sequence of local transformations, we base our architecture on recurrent neural networks. To the best of our knowledge, recurrent neural networks are not used before for deformable image registration.

## 6.2 Background

Given two images that need to be aligned, the fixed image  $F : \mathcal{X} \rightarrow \mathbb{R}$  and the moving image  $M : \mathcal{X} \rightarrow \mathbb{R}$  on the image domain  $\mathcal{X} \subset \mathbb{R}^d$ , the pairwise registration problem can be defined as a regularized minimization problem

$$f^* = \arg \min_f \mathcal{S}[F, M \circ f] + \lambda \mathcal{R}[f]. \quad (6.1)$$

Here,  $f^* : \mathcal{X} \rightarrow \mathbb{R}^d$  is the transformation of interest and a minimizer of (6.1). The image loss  $\mathcal{S} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  determines the image similarity of  $F$  and  $M \circ f$ , with  $(M \circ f)(x) = M(x + f(x))$ . In order to restrict the transformation space by using prior knowledge of the transformation, a regularization loss  $\mathcal{R} : \mathbb{R}^d \rightarrow \mathbb{R}$  and the regularization weight  $\lambda$  are added to the optimization problem. The regularizer is chosen depending on the expected transformation characteristics (e.g. global smoothness or piece-wise smoothness).

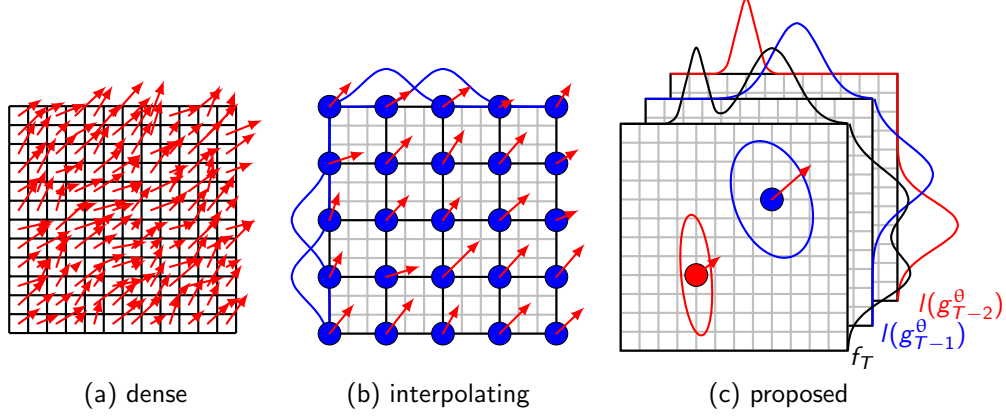
### Transformation

In order to optimize (6.1) a transformation model  $f_\theta$  is needed. The minimization problem then becomes

$$\theta^* = \arg \min_\theta \mathcal{S}[F, M \circ f_\theta] + \lambda \mathcal{R}[f_\theta], \quad (6.2)$$

where  $\theta$  are the parameters of the transformation model. There are two major classes of transformation models used in image registration: dense and interpolating. In the dense case, the transformation at position  $x$  in the image is defined by a displacement vector

$$f_\theta(x) = \theta_x, \quad (6.3)$$



**Figure 6.2:** Dense, interpolating, and proposed transformation models.

with  $\theta_x = (\vartheta_1, \vartheta_2, \dots, \vartheta_d) \in \mathbb{R}^d$ . For the interpolating case the transformation at position  $x$  is normally defined in a smooth basis

$$f_\theta(x) = \sum_i^N \theta_i k(x, c_i). \quad (6.4)$$

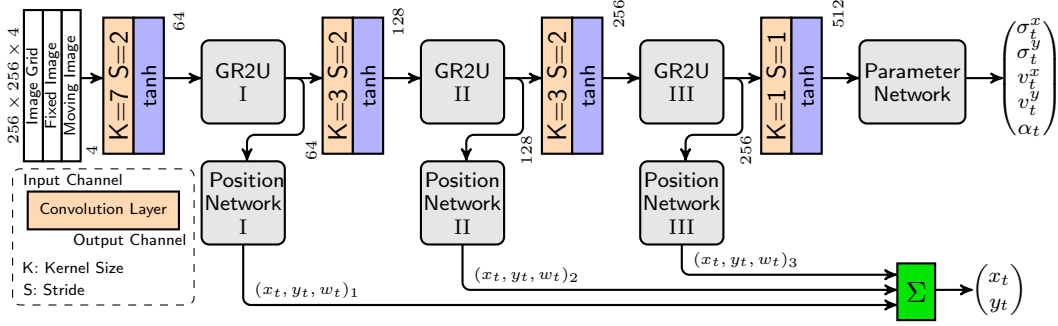
Here,  $\{c_i\}_{i=1}^N, c_i \in \mathcal{X}$  are the positions of the fixed regular grid points in the image domain,  $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  the basis function, and  $N$  the number of grid points. The transformation between the control points  $c_i$  is an interpolation of the control point values  $\theta_i \in \mathbb{R}^d$  with the basis function  $k$ . A visualization of a dense and an interpolating transformation model is shown in Figure 6.2.

### Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a class of neural networks designed for sequential data. A simple RNN has the form

$$h_t = \phi(Wx_t + Uh_{t-1}), \quad (6.5)$$

where  $W$  is a weighting matrix of the input at time  $t$ ,  $U$  is the weight matrix of the last output at time  $t-1$ , and  $\phi$  is an activation function like the hyperbolic tangent or the logistic function. Since the output at time  $t$  directly depends on the weighted previous output  $h_{t-1}$ , RNNs are well suited for the detection of sequential information which is encoded in the sequence itself. RNNs provide an elegant way of incorporating the whole previous sequence without adding a large number of parameters. Besides the advantage of RNNs for sequential data, there are some difficulties to address e.g. the problem to learn long-term dependencies. The long short-term memory (LSTM) architecture was introduced in order to overcome these problems of the basic RNN [Hochreiter and Schmidhuber \(1997\)](#). A variation of the LSTM, the gated recurrent unit (GRU) was presented by [Cho et al. \(2014\)](#).



**Figure 6.3:** Network architecture of the presented Recurrent Registration Neural Network.

## 6.3 Methods

In the following, we will present our *Recurrent Registration Neural Network* (R2N2) for the application of sequence-based pairwise medical image registration of 2D images.

### Sequence-Based Image Registration

Sequence-based registration methods do not estimate the final transformation in one step but rather in a series of local transformations. The minimization problem for the sequence-based registration is given as

$$\theta^* = \arg \min_{\theta} \frac{1}{T} \sum_{t=1}^T \mathcal{S}[F, M \circ f_t^{\theta}] + \lambda \mathcal{R}[f_T]. \quad (6.6)$$

Compared to the registration problem (6.2) the transformation  $f_t^{\theta}$  is now defined as a recursive function of the form

$$f_t^{\theta}(x, F, M) = \begin{cases} 0, & \text{if } t = 0, \\ f_{t-1}^{\theta} + l(x, g_{\theta}(F, M \circ f_{t-1}^{\theta})) & \text{else.} \end{cases} \quad (6.7)$$

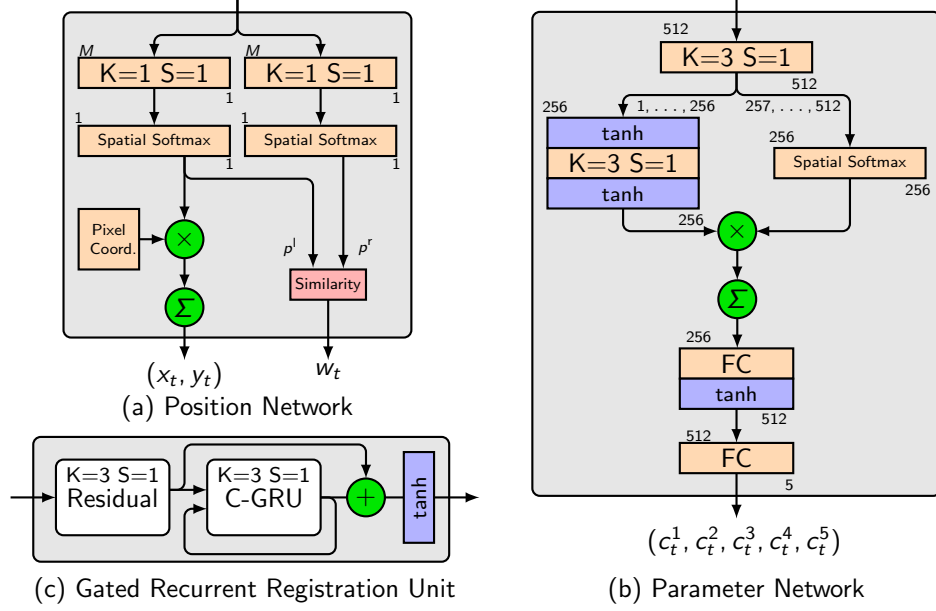
Here,  $g_{\theta}$  is the function that outputs the parameter of the next local transformation given the two images  $F$  and  $M \circ f_t^{\theta}$ . In each time step  $t$ , a local transformation  $l : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^2$  is computed and added to the transformation  $f_t^{\theta}$ . After transforming the moving image  $M$  with  $f_t^{\theta}$ , the result is used as input for the next time step, in order to compute the next local transformation as shown in Figure 6.1. This procedure is repeated until both input images are aligned. We define a local transformation as a Gaussian function

$$l(x, \tilde{x}_t, \Gamma_t, v_t) = v_t \exp \left( -\frac{1}{2} (x - \tilde{x}_t)^T \Sigma(\Gamma_t)^{-1} (x - \tilde{x}_t) \right), \quad (6.8)$$

where  $\tilde{x}_t = (x_t, y_t) \in \mathcal{X}$  is the position,  $v_t = (v_t^x, v_t^y) \in [-1, 1]^2$  the weight, and  $\Gamma_t = \{\sigma_t^x, \sigma_t^y, \alpha_t\}$  the shape parameter with

$$\Sigma(\Gamma_t) = \begin{bmatrix} \cos(\alpha_t) & -\sin(\alpha_t) \\ \sin(\alpha_t) & \cos(\alpha_t) \end{bmatrix} \begin{bmatrix} \sigma_t^x & 0 \\ 0 & \sigma_t^y \end{bmatrix} \begin{bmatrix} \cos(\alpha_t) & -\sin(\alpha_t) \\ \sin(\alpha_t) & \cos(\alpha_t) \end{bmatrix}^T. \quad (6.9)$$

Here,  $\sigma_t^x, \sigma_t^y \in \mathbb{R}_{>0}$  control the width and  $\alpha_t \in [0, \pi]$  the rotation of the Gaussian function. The output of  $g_\theta$  is defined as  $g_\theta = \{\tilde{x}_t, \Gamma_t, v_t\}$ . Compared to the interpolating registration model shown in Figure 6.2b, the position  $\tilde{x}_t$  and shape  $\Gamma_t$  of the basis functions are not fixed during the registration in our method (Figure 6.2c).



**Figure 6.4:** Architectures for the position network, the parameter network, and the gated recurrent registration unit.

## Network Architecture

We developed a network architecture to approximate the unknown function  $g_\theta$ , where  $\theta$  are the parameters of the network. Since the transformation of the registration is defined as a recursive sequence, we base our network up on GRUs due to their efficient gated architecture. An overview of the complete network architecture is shown in Figure 6.3. The input of the network are two images, the fixed image  $F$  and the moving image  $M \circ f_t$ . As suggested in Liu et al. (2018), we attached the position of each pixel as two additional coordinate channels to improve the convolution layers for the handling of spatial representations. Our network contains three major sub-networks to generate the parameters of the local transformation: the gated recurrent registration unit (GR2U), the position network, and the parameter network.

**Gated Recurrent Registration Unit** Our network contains three GR2U for different spatial resolutions ( $128 \times 128$ ,  $64 \times 64$ ,  $32 \times 32$ ). Each GR2U has an internal structure as shown in Figure 6.4c. The input of the GR2U block is passed through a residual network,

with three stacked residual blocks [He et al. \(2016\)](#). If not stated otherwise, we use the hyperbolic tangent as activation function in the network. The core of each GR2U is the C-GRU block. For this, we adopt the original GRU equations shown in [Cho et al. \(2014\)](#) in order to use convolutions instead of a fully connected layer as presented in [Andermatt et al. \(2017\)](#). In contrast to [Andermatt et al. \(2017\)](#), we adapt the proposal gate (6.12) for use with convolutions, but without factoring  $r_j$  out of the convolution. The C-GRU is then defined by:

$$r^j = \psi \left( \sum_i^I (x * w_r^{i,j}) + \sum_k^J (h_{t-1}^k * u_r^{k,j}) + b_r^j \right), \quad (6.10)$$

$$z^j = \psi \left( \sum_i^I (x * w_z^{i,j}) + \sum_k^J (h_{t-1}^k * u_z^{k,j}) + b_z^j \right), \quad (6.11)$$

$$\tilde{h}_t^j = \phi \left( \sum_i^I (x * w^{i,j}) + \sum_k^J ((r_j \odot h_{t-1}^k) * u^{k,j}) + b^j \right), \quad (6.12)$$

$$h_t^j = (1 - z^j) \odot h_{t-1}^j + z^j \odot \tilde{h}_t^j. \quad (6.13)$$

Here,  $r$  represents the reset gate,  $z$  the update gate,  $\tilde{h}_t$  the proposal state, and  $h_t$  the output at time  $t$ . We define  $\phi(\cdot)$  as the hyperbolic tangent,  $\psi(\cdot)$  represents the logistic function, and  $\odot$  is the Hadamard product. The convolution is denoted as  $*$  and  $u, w, b$  are the parameters to be learned. The indices  $i, j, k$  correspond to the input and output/state channel index. We also applied a skip connection from the output of the residual block to the output of the C-GRU.

**Position Network** The architecture of the position network is shown in Figure 6.4a and contains two paths. In the left path, the position of the local transformation  $x_t^n$  is calculated using a convolution layer followed by the *spatial softmax* function [Finn et al. \(2016\)](#). Here,  $n$  is the level of the spatial resolution. The spatial softmax function is defined as

$$p_k(c_{ij}^k) = \frac{\exp(c_{ij}^k)}{\sum_{i'} \sum_{j'} \exp(c_{i'j'}^k)}, \quad (6.14)$$

where  $i$  and  $j$  are the spatial indices of the  $k$ -th feature map  $c$ . The position is then calculated by

$$x_t^n = \left( \sum_i \sum_j p(c_{ij}) X_{ij}^n, \sum_i \sum_j p(c_{ij}) Y_{ij}^n \right), \quad (6.15)$$

where  $(X_{ij}^n, Y_{ij}^n) \in \mathcal{X}$  are the coordinates of the image pixel grid. As shown in Figure 6.3 an estimate of the current transformation position is computed on all three spatial

levels. The final position is calculated as a weighted sum

$$\tilde{x}_t = \frac{\sum_n^3 x_t^n w_t^n}{\sum_n^3 w_t^n}. \quad (6.16)$$

The weights  $w_t^n \in \mathbb{R}$  are calculated on the right side of the position block. For this, a second convolution layer and a second *spatial softmax* layer are applied to the input of the block. We calculated the similarity of the left spatial softmax  $p^l(c_{ij})$  and the right spatial softmax  $p^r(c_{ij})$  as the weight of the position at each spatial location

$$w_t^n = 2 - \sum_i \sum_j \left| p^l(c_{ij}) - p^r(c_{ij}) \right|. \quad (6.17)$$

This weighting factor can be interpreted as certainty measure of the estimation of the current position at each spatial resolution.

**Parameter Network** The parameter network is located at the end of the network. Its detailed structure is shown in Figure 6.4b. The input of the parameter block is first passed through a convolution layer. After the convolution layer, the first half of the output feature maps is passed through a second convolution layer. The second half is applied to a *spatial softmax* layer. For each element in both outputs, a point-wise multiplication is applied, followed by an average pooling layer down to a spatial resolution of  $1 \times 1$ . We use a fully connected layer with one hidden layer in order to reduce the output to the number of needed parameters. The final output parameters are then defined as

$$\sigma_t^x = \psi(c_t^1) \sigma_{\max}, \quad \sigma_t^y = \psi(c_t^2) \sigma_{\max}, \quad v_t^x = \phi(c_t^3), \quad v_t^y = \phi(c_t^4), \quad \alpha_t = \psi(c_t^5) \pi, \quad (6.18)$$

where  $\phi(\cdot)$  is the hyperbolic tangent,  $\psi(\cdot)$  the logistic function, and  $\sigma_{\max}$  the maximum extension of the shape.

## 6.4 Experiments and Results

**Image Data** We trained our network on images of a 2D+t magnetic resonance (MR) image series of the lung. Due to the low proton density of the lung parenchyma in comparison to other body tissues as well as strong magnetic susceptibility effects, it is very challenging to acquire MR images with a sufficient signal-to-noise ratio. Recently, a novel MR pulse sequence called ultra-fast steady-state free precession (ufSSFP) was proposed [Bauman et al. \(2016\)](#). ufSSFP allows detecting physiological signal changes in lung parenchyma caused by respiratory and cardiac cycles, without the need for intravenous contrast agents or hyperpolarized gas tracers. Multi-slice 2D+t ufSSFP acquisitions are performed in free-breathing.

For a complete chest volume coverage, the lung is scanned at different slice positions as shown in Figure 6.5. At each slice position, a dynamic 2D+t image series with 140

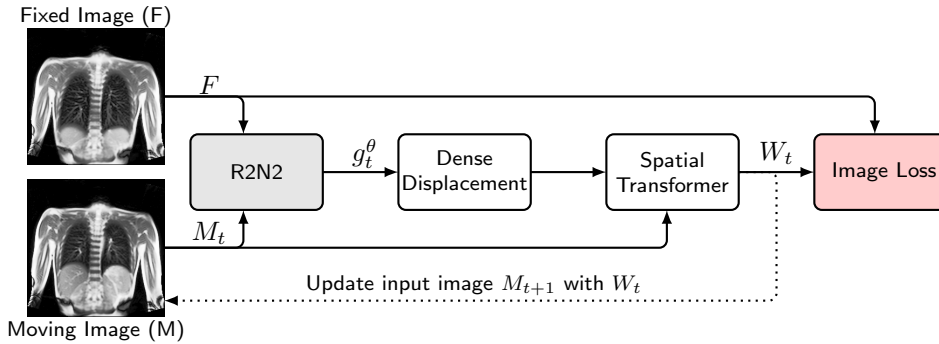




**Figure 6.5:** Maximum inspiration (top row) and maximum expiration (bottom row) for different slice positions of one patient from back to front.

images is acquired. For the further analysis of the image data, all images of one slice position need to be spatially aligned. We choose the image which is closest to the mean respiratory cycle as fixed image of the series. The other images of the series are then registered to this image. Our data set consists of 48 lung acquisitions of 42 different patients. Each lung scan contains between 7 and 14 slices. We used the data of 34 patients for the training set, 4 for the evaluation set, and 4 for the test set.

**Network Training** The network was trained in an unsupervised fashion for  $\sim 180,000$  iterations with a fixed sequence length of  $t = 25$ . Figure 6.6 shows an overview of the training procedure. We used the Adam optimizer [Kingma and Ba \(2014\)](#) with the AMSGrad option [Reddi et al. \(2019\)](#) and a learning rate of 0.0001. The maximum shape size is set to  $\sigma_{\max} = 0.3$  and the regularization weight to  $\lambda_{R2N2} = 0.1$ . For the regularization of the network parameter, we use a combination of [Srivastava et al. \(2014\)](#) particularly the use of Gaussian multiplicative noise and dropconnect [Wan et al. \(2013\)](#). We apply multiplicative Gaussian noise  $\mathcal{N}(1, \sqrt{0.5}/0.5)$  to the parameter of the proposal and the output of the C-GRU. As image loss function  $\mathcal{S}$  the mean squared error (MSE) loss is used and as transformation regularizer  $\mathcal{R}$  the isotropic total variation (TV). The training of the network was performed on an NVIDIA Tesla V100 GPU.



**Figure 6.6:** Unsupervised training setup ( $W_t$  is the transformed moving image).

**Table 6.1:** Mean target registration error (TRE) for the proposed method R2N2 and a standard B-spline registration (BS) for the test data set in millimeter. The small number is the maximum TRE for all images for this slice.

Patient	Slice 1	Slice 2	Slice 3	Slice 4	Slice 5	Slice 6	Slice 7	Slice 8	mean
R2N2	1.26 1.85	1.08 2.14	1.13 1.82	1.23 2.58	1.47 2.74	1.12 1.51	0.92 1.33	1.04 1.87	1.16
BS	1.28 1.81	1.16 2.0	1.40 2.52	1.15 2.67	0.96 1.71	0.99 1.41	0.84 1.14	1.02 1.65	1.10
R2N2	0.84 1.99	0.92 2.49	0.79 1.04	0.81 1.2	0.74 1.43	—	—	—	0.82
BS	1.50 5.07	0.69 1.73	0.73 1.05	0.77 1.13	0.86 1.76	—	—	—	0.91
R2N2	1.65 3.88	1.06 2.55	0.86 2.08	0.83 1.48	0.80 1.39	0.73 1.08	—	—	0.99
BS	1.15 2.73	0.81 1.42	0.75 1.64	0.79 1.14	0.72 0.94	0.83 1.95	—	—	0.84
R2N2	1.30 3.03	0.77 0.98	0.79 2.07	1.09 1.92	0.84 1.12	—	—	—	0.96
BS	1.09 3.15	0.78 1.01	0.73 1.73	1.09 2.5	0.79 1.13	—	—	—	0.90

**Experiments** We compare our method against a standard B-spline registration method (BS) implemented in the AirLab framework [Sandkühler et al. \(2018a\)](#). The B-spline registration use three spatial resolutions (64, 128, 256) with a kernel size of (7, 21, 57) pixels. As image loss the MSE and as regularizer the isotropic TV is used, with the regularization weight  $\lambda_{BS} = 0.01$ . We use the Adam optimizer [Kingma and Ba \(2014\)](#) with the AMSGrad option [Reddi et al. \(2019\)](#), a learning rate of 0.001, and we perform 250 iterations per resolution level.

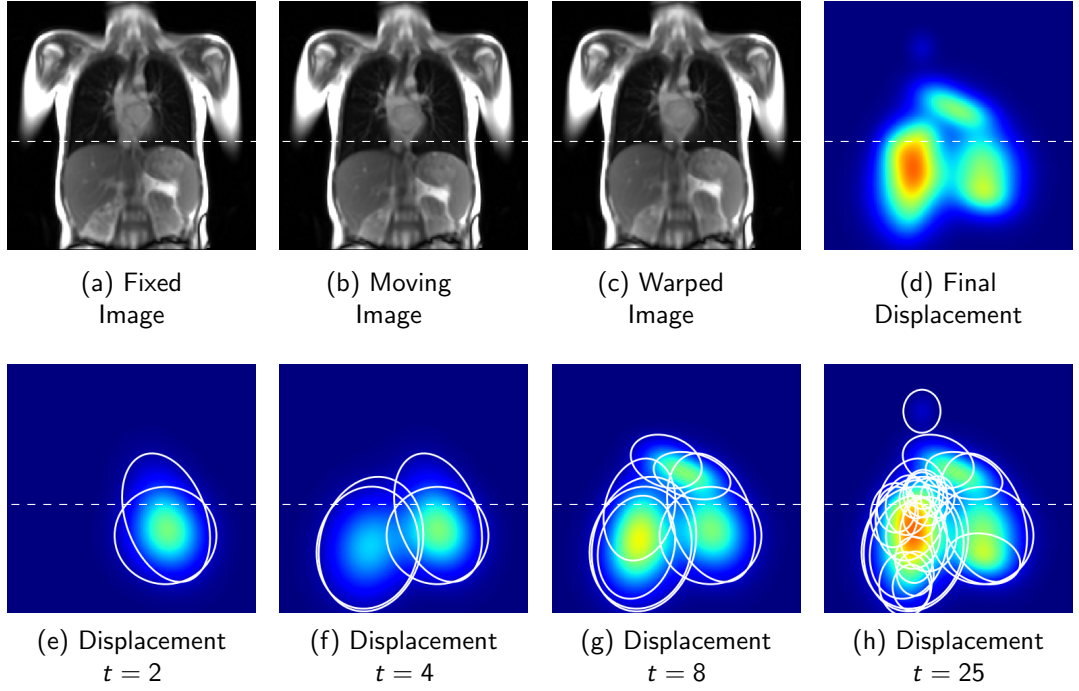
From the test set we select 21 images of each slice position, which corresponds to one breathing cycle. We then select corresponding landmarks in all 21 images in order to compute the registration accuracy. The target registration error (TRE) of the registration is defined as the mean root square error of the landmark distance after the registration. The results in Table 6.1 show that our presented method performed on par with the standard B-spline registration in terms of accuracy. Since the slice positions are manually selected for each patient, we are not able to provide the same amount of slices for each patient. Despite that the image data is different at each slice position, we see a good generalization ability of our network to perform an accurate registration independently of the slice position at which the images are acquired.

Our method achieve a compact representation of the final transformation, by using only  $\sim 7.6\%$  of the amount of parameters than the final B-spline transformation. Here, the number of parameters of the network are not taken into account only the number of parameters needed to describe the final transformation. For the evaluation of the computation time for the registration of one image pair, we run both methods on an NVIDIA GeForce GTX 1080. The computation of the B-spline registration takes  $\sim 4.5s$  compared to  $\sim 0.3s$  for our method.

An example registration result of our presented method is shown in Figure 6.7. It can be seen that the first local transformations the network creates are placed below the diaphragm (white dashed line) (Figure 6.7a), where the magnitude of the motion between the images is maximal. Also the shape and rotation of the local transformations are computed optimally in order to apply a transformation only at the liver and the lung and not on the ribs. During the next time steps, we can observe that the shape of the local transformation is reduced to align finer details of the images (Figure 6.7g-h).

## 6.5 Conclusion

In this paper, we presented the *Recurrent Registration Neural Network* for the task of deformable image registration. We define the registration process of two images as a recursive sequence of local deformations. The sum of all local deformations yields the final spatial alignment of both images. Our designed network can be trained end-to-end in an unsupervised fashion. The results show that our method is able to accurately register two images with a similar accuracy compared to a standard B-spline registration method. We achieve a speedup of  $\sim 15$  for the computation time compared to the B-spline registration. In addition, we need only  $\sim 7.6\%$  of the amount of parameters to describe the final transformation than the final transformation of the standard B-spline



**Figure 6.7:** Top Row: Registration result of the proposed recurrent registration neural network for one image pair. Bottom Row: Sequence of local transformations after different time steps.

registration. In this paper, we have shown that our method is able to register two images in a recursive manner using a fixed number of steps. For future work we will include uncertainty measures for the registration result as a possible stopping criteria. This could then be used to automatically determine the number of steps needed for the registration. Furthermore, we will extend our method for the registration of 3D volumes.

## Acknowledgements

We would like to thank Oliver Bieri, Orso Pusterla (Division of Radiological Physics, Department of Radiology, University Hospital Basel, Switzerland), and Philipp Latzin (Pediatric Respiratory Medicine, Department of Pediatrics, Inselspital, Bern University Hospital, University of Bern, Switzerland) for their support during the development of this work. Furthermore, we thank the Swiss National Science Foundation for funding this project (SNF 320030\_149576).

# AIRLab: Autograd Image Registration Laboratory

In the following paper, we present the Autograd Image Registration Laboratory (AIRLab). AIRLab is a registration framework based on the PyTorch<sup>1</sup> framework. It is designed for the development of new registration methods and for the fast implementation of current methods, by using the automatic gradient calculation provided by the PyTorch framework. In the AIRLab framework a wide variety of different tools for image registration are implemented, e.g., B-spline transformations, Demons registration, diffeomorphic transformations, and several image similarity measures. The work is a cooperation between Christoph Jud and myself Robin Sandkühler. The main idea of the AIRLab framework and the implementation of the basic framework containing, transformation models, image similarity measures, and different registration types were done by myself Robin Sandkühler. I also wrote the part of the diffeomorphic transformation, the similarity measures, and I performed the experiments for the evaluation part in the following paper. The other parts of the paper were mainly written by Christoph Jud. He implemented the functions which allow the registration of images with different domain sizes, and the automatic evaluation part. The code for this framework is open-source<sup>2</sup>

## Technical Report

This technical report has been submitted to arxiv.org<sup>3</sup> Furthermore, the AIRLab framework was presented as a tutorial at the Conference for Medical Image Computing and Computer Assisted Surgery (MICCAI), Shenzhen, China, 2019.

---

<sup>1</sup>PyTorch framework: <https://pytorch.org>

<sup>2</sup>AIRLab framework: <https://github.com/airlab-unibas/airlab>

<sup>3</sup><https://arxiv.org/abs/1806.09907>



# AIRLab: Autograd Image Registration Laboratory

Robin Sandkühler<sup>1</sup>, Christoph Jud<sup>1</sup>, Simon Andermatt<sup>1</sup>, Philippe C. Cattin<sup>1</sup>

## Abstract

Medical image registration is an active research topic and forms a basis for many medical image analysis tasks. Although image registration is a rather general concept specialized methods are usually required to target a specific registration problem. The development and implementation of such methods has been tough so far as the gradient of the objective has to be computed. Also, its evaluation has to be performed preferably on a GPU for larger images and for more complex transformation models and regularization terms. This hinders researchers from rapid prototyping and poses hurdles to reproduce research results. There is a clear need for an environment which hides this complexity to put the modeling and the experimental exploration of registration methods into the foreground. With the “Autograd Image Registration Laboratory” (AIRLab), we introduce an open laboratory for image registration tasks, where the analytic gradients of the objective function are computed automatically and the device where the computations are performed, on a CPU or a GPU, is transparent. It is meant as a laboratory for researchers and developers enabling them to rapidly try out new ideas for registering images and to reproduce registration results which have already been published. AIRLab is implemented in Python using PyTorch as tensor and optimization library and SimpleITK for basic image IO. Therefore, it profits from recent advances made by the machine learning community concerning optimization and deep neural network models.

The presented draft of this paper outlines AIRLab with first code snippets and performance analyses. A more exhaustive introduction will follow as a final version soon.

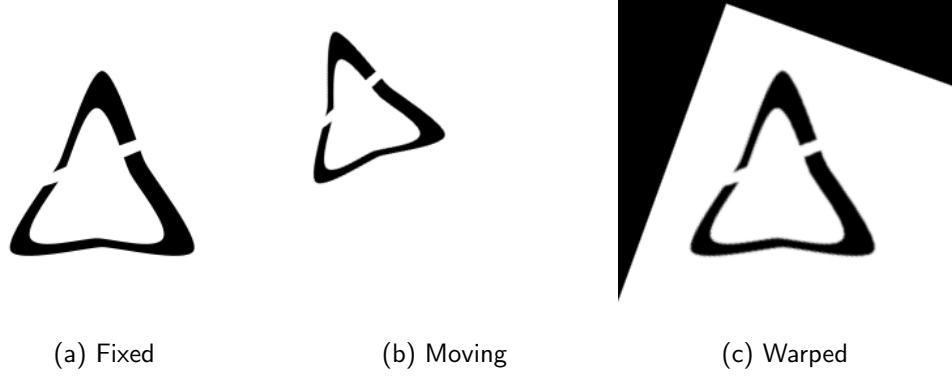
## 7.1 Introduction

The registration of images is a growing research topic and forms an integral part in many medical image analysis tasks [Viergever et al. \(2016\)](#). It is referred to as the process of finding corresponding structures within different images. There is a large number of applications where image registration is inevitable such as e.g. the fusion of different modalities, monitoring anatomical changes, population modelling or motion extraction.

Image registration is a nonlinear, ill-posed problem which is approached by optimizing a regularized objective. What is defined as quite general requires usually specialized objective functions and implementations for applying it to specific registration tasks. The development of such specific registration methods has been tough so far and their implementation tedious. This is because gradients have to be computed within the

---

<sup>1</sup>Department of Biomedical Engineering, University of Basel, Allschwil, Switzerland



**Figure 7.1:** (a) Fixed AIRLab image, (b) moving AIRLab image and (c) warped moving AIRLab image after registration.

optimization whose implementations are error-prone, especially for 3D objectives. Furthermore, for large 3D images, the computational demand is usually high and a parallel execution on a GPU unavoidable. These are problems which hinder researchers from playing around with different combinations of objectives and regularizers and rapidly trying out new ideas. Similarly, the effort to reproduce registration results is often out of proportion. There is a clear need for an environment which hides this complexity, enables rapid prototyping and simplifies reproduction.

In this paper, we introduce “Autograd Image Registration Laboratory” (AIRLab), an image registration environment - or a laboratory - for rapid prototyping and reproduction of image registration methods. Thus, it addresses researchers and developers and simplifies their work in the exploration of different registration methods, in particular also with upcoming complex deep learning approaches. It is written in Python and based on the tensor library PyTorch [Paszke et al. \(2017\)](#). It heavily uses features from PyTorch such as autograd, the rich family of optimizers and the transparent utilization of GPUs. In addition, SimpleITK [Lowekamp et al. \(2013\)](#) is included for data input/output to support all standard image file formats. AIRLab comes along with state-of-the-art registration components including various image similarity measures, regularization terms and optimizers. Experimenting with such building blocks or trying out new ideas, for say a regularizer, becomes almost effortless as gradients are computed automatically. Finally, example implementations of standard image registration methods are provided such as Optical Flow [Horn and Schunck \(1981\)](#), Demons [Thirion \(1998\)](#) and Free Form Deformations [Rueckert et al. \(1999\)](#). Deep learning based models are currently not implemented in the AIRLab framework, but we will integrate them in future releases. AIRLab is licensed under the Apache License 2.0 and available on GitHub<sup>4</sup>.

In the following, we first provide a brief background about medical image registration followed by the description of AIRLab, its building blocks and its features. Finally, we

<sup>4</sup><https://github.com/airlab-unibas/airlab>



provide registration experiments with standard registration methods which are implemented in AIRLab including performance analyses and code snippets.

The present draft of this paper roughly introduces AIRLab and is intended for the presentation at the 8th International Workshop on Biomedical Image Registration in Leiden. A more detailed final version will follow soon.

## 7.2 Background

### Image Registration

Let  $\mathcal{X} := \{x_i\}_{i=1}^N$  be a set of  $N$  points arranged in a regular grid which covers the joint image domain of a moving and fixed image  $I_M, I_F : \mathcal{X} \rightarrow \mathbb{R}$ . The images map the  $d$ -dimensional spatial domain  $\mathcal{X} \subset \mathbb{R}^d$  to intensity values. Furthermore, let  $f : \mathcal{X} \rightarrow \mathbb{R}^d$  spatially transform the coordinate system of the moving image. Image registration can be formulated as a regularized minimization problem

$$f^* = \arg \min_f \mathcal{S}_{\mathcal{X}}(I_M \circ f, I_F) + \lambda \mathcal{R}(f, \mathcal{X}), \quad (7.1)$$

where  $\mathcal{S}_{\mathcal{X}}$  is a similarity measure between the transformed moving image and the fixed image and  $\mathcal{R}$  is a regularization term which operates on  $f$  on the domain  $\mathcal{X}$ . The two terms are balanced by  $\lambda$  and  $\circ$  is the function composition. An example for a similarity measure is the mean squared error measure for monomodal image registration

$$\mathcal{S}_{\text{MSE}} := \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \left( I_M(x + f(x)) - I_F(x) \right)^2, \quad (7.2)$$

where  $|\cdot|$  is the cardinality of a set. An exemplary regularization term is the diffusion regularization which favours smooth transformations

$$\mathcal{R}_{\text{diff}} := \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \sum_{i=1}^d \|\nabla f_i(x)\|^2 \quad (7.3)$$

where  $i$  indexes the space dimension. In the Sections 7.3 and 7.3, the similarity measures and regularizers which are implemented in AIRLab are described in more detail.

**Transformation** Transformation models  $f$  can be divided in basically four types: *linear*, *non-linear/dense*, *non-linear/interpolating* and *hybrid*. Linear transformations, available in AIRLab, transform each point  $x$  with a linear map  $A$

$$f(x) := A\tilde{x}, \quad (7.4)$$

where  $A$  is an rotation/translation matrix up to 3 in 2D and 6 degrees of freedom in 3D and  $x$  stands in homogeneous coordinates  $\tilde{x}$ . The class non-linear/parametric transform

models consists mainly of two types: interpolating models and dense models. Non-linear/interpolating transformations are defined in an interpolating basis on a coarse grid of  $n < N$  control points

$$f(x) := \sum_{i=1}^n c_i k(x_i, x), \quad (7.5)$$

where  $c_i \in \mathbb{R}^d$  and  $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  is the basis function. Common basis functions are the B-spline [Rueckert et al. \(1999\)](#) or Wendland kernel [Jud et al. \(2016b\)](#) which both are implemented as example basis in AIRLab (see Section 7.3). The advantage of non-linear/interpolating transformation models are, that they are computationally efficient. Furthermore, if  $k$  is smooth they inherently yield smooth transformations.

In non-linear dense transformation models, each point in the image can be transformed individually in  $d$  dimensions giving a maximum flexibility. To still be able to reach reasonable registration results the regularization term is inevitable. Hierarchical models can be seen as hybrid interpolating and dense models. Their hierarchical structure enables them to capture large deformations [Jud et al. \(2018b\)](#). Dense transformation models are supported as well by AIRLab while hybrid are planned.

**Optimization** The similarity measure depends nonlinearly on the moving image  $I_M$ , which makes an analytical solution to Equation (7.1) intractable. Because in non-linear registration the number of parameters of  $f$  is in the millions, gradient based optimization is usually the only choice to reach a *locally* optimal transformation  $f^*$ . Having PyTorch at hand, state-of-the-art optimizers are available in AIRLab such as LBFGS [Byrd et al. \(1994\)](#) and Adam [Kingma and Ba \(2014\)](#).

## Image Registration Frameworks

There are already a considerable amount of medical image registration frameworks available which are valuable enrichments to the community. Their focus and intensions are diverse, ranging from rich frameworks to specific implementations. For an exhaustive list and comparison of such image registration software we refer to [Keszei et al. \(2017\)](#). Gradient free approaches as e.g. the MRF-based method of [Glocker et al. \(2008\)](#) are out of scope of the current implementation of AIRLab.

The Insight Segmentation and Registration Toolkit (ITK) [Yoo et al. \(2002\)](#) is a comprehensive framework for diverse image processing and analysis task written in C++. It is mainly intended for the use as a library for developers who want to implement ready-to-use software. The registration tool Elastix [Klein et al. \(2010\)](#) is based on ITK and provides a collection of algorithms commonly used in image registration. It can also be used out-of-the-box with the possibility of a detailed configuration script. Furthermore, its plug-in architecture allows to integrate custom parts into the software. The extension SimpleElastix [Marstal et al. \(2016\)](#) offers bindings to other languages such as Python, Java, Ruby and more. Elastix and SimpleElastix are strong if one needs some

flexibility in choosing and combining different registration components for a specific registration task. SuperElastix is a registration framework, that allows the combination of different existing registration frameworks [Berendsen et al. \(2016\)](#). Scalable Image Analysis and Shape Modelling (Scalismo) [Bouabene et al. \(2015\)](#); [Lüthi et al. \(2017\)](#) is a library mainly for statistical shape modeling written in scala. It provides also image registration functionality and can be interactively executed similar to SimpleElastix. Advanced Normalization Tools (ANTs) [Avants et al. \(2011\)](#) is based on ITK as well. It provides a command line tool including large deformation registration algorithms with standard similarity measures. The Automated Image Registration software AIR [Woods et al. \(1998\)](#) is written in C and provides basic registration functionality for linear and polynomial non-linear image alignment up to the twelfth order. The Medical Image Registration ToolKit (MIRTK) [Rueckert et al. \(1999\)](#); [Schnabel et al. \(2001\)](#) is a collection of libraries and command-line tools for image and point-set registration. Various registration methods based on free form deformations are provided. Flexible Algorithms for Image Registration (FAIR) [Modersitzki \(2009\)](#) is a software package written in MATLAB comprising various similarity measures and regularizers.

None of the mentioned software packages are suited for rapid prototyping in the development of image registration algorithms. This is mainly because: (I) For the optimization, gradients have to be provided explicitly. For complex transformation models, regularization terms and similarity measures, their implementation is highly error-prone. (II) For medical images, the computational demand is usually high and therefore the execution has to be performed on a GPU. The development for GPUs without an appropriate framework is not trivial. (III) The majority of the frameworks are written in C++. Thus, the development within those frameworks needs good expertise in this language. Furthermore, the number of code lines required for C++ implementations in these frameworks do not agree with the concept of rapid prototyping.

## 7.3 Autograd Image Registration Laboratory

AIRLab is a rapid prototyping environment for medical image registration. Its unique characteristics are the automatic differentiation and the transparent usage of GPUs. It is written in the scripting language Python and heavily uses key functionality of PyTorch [Paszke et al. \(2017\)](#).

The main building blocks constitute:

- Automatic differentiation
- Similarity measures
- Transformation models
- Image warping
- Regularization terms
- Optimizers

## Automatic Symbolic Differentiation

A key feature of AIRLab is its automatic *symbolic* differentiation of the objective function. This means, that only the forward function has to be provided by the developer and the gradient which is required for the optimization is derived through automatic differentiation (AD). AIRLab borrows the AD functionality of PyTorch. It is one of the fastest dynamic AD frameworks currently available. Its strong points are:

- *Dynamic*: the function which is *symbolically* differentiated is defined by the computations which are run on the variables. Hence, no static graph structure has to be built which fosters rapid prototyping.
- *Immediate*: only tensor computations which are necessary for differentiation are recorded
- *Core logic*: a low overhead is needed as the AD logic is written in C++ and was carefully tuned

Please cf. [Paszke et al. \(2017\)](#) for more details.

## Similarity Measures

We list here the main building blocks required for medical image registration which are provided by AIRLab.

- **Mean Squared Errors (MSE)**: a simple and fast to compute point-wise measure which is well suited for monomodal image registration

$$\mathcal{S}_{\text{MSE}} := \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \left( I_{\text{M}}(x + f(x)) - I_{\text{F}}(x) \right)^2. \quad (7.6)$$

Class name: MSE

- **Normalized Correlation Coefficient (NCC)**: a point-wise measure as  $\mathcal{S}_{\text{MSE}}$ . It is targeted to image registration tasks, where the intensity relation between the moving and the fixed images is linear

$$\mathcal{S}_{\text{NCC}} := \frac{\sum I_{\text{F}} \cdot (I_{\text{M}} \circ f) - \sum \text{E}(I_{\text{F}}) \text{E}(I_{\text{M}} \circ f)}{|\mathcal{X}| \cdot \sum \text{Var}(I_{\text{F}}) \text{Var}(I_{\text{M}} \circ f)}, \quad (7.7)$$

where the sums go over the image domain  $\mathcal{X}$ , E is the expectation value (or mean) and Var is the variance of the respective image.

Class name: NCC

- **Local Cross Correlation (LCC)**: is the localized version of  $\mathcal{S}_{\text{NCC}}$  where the expectation value and the variance for a given  $x$  are computed in a local neighborhood of  $x$ . In AIRLab  $\mathcal{S}_{\text{LCC}}$  is implemented with efficient convolution operations. Notice that the exact gradient is computed using autograd and no gradient approximation is performed in contrast to [Cachier and Pennec \(2000\)](#).

Class name: LCC

- **Structural Similarity Index Measure (SSIM)**: is a generalization of the LCC similarity measure and was presented by [Wang et al. \(2004\)](#) as an image quality criterion. The SSIM for two local image patches  $a \in \mathbb{R}^d$ ,  $b \in \mathbb{R}^d$  is defined as

$$\text{SSIM}(a, b) = l(a, b)^\alpha c(a, b)^\beta s(a, b)^\gamma, \quad (7.8)$$

with  $\alpha, \beta, \gamma \in [0, 1]$ . The SSIM combines three different similarity measures: the luminance ( $l$ )

$$l(a, b) = \frac{2\mu_a\mu_b + c_1}{\mu_a^2 + \mu_b^2 + c_1}, \quad (7.9)$$

the contrast ( $c$ )

$$c(a, b) = \frac{2\sigma_a\sigma_b + c_2}{\sigma_a^2 + \sigma_b^2 + c_2}, \quad (7.10)$$

and the and structure ( $s$ )

$$s(a, b) = \frac{\sigma_{ab} + c_3}{\sigma_a\sigma_b + c_3}. \quad (7.11)$$

Here,  $\mu$  is the mean,  $\sigma$  the standard deviation of an image patch,  $\sigma_{ab}$  the correlation coefficient, and  $c_1, c_2, c_3 \in \mathbb{R}$  are used to reduce numerical instabilities. For a complete image the SSIM is defined as

$$\mathcal{S}_{\text{SSIM}}(I_F, I_M, f) = \frac{1}{|\mathcal{X}|} \sum_{\substack{x \in I_F \\ y \in I_M \circ f}} l[x, y]^\alpha c[x, y]^\beta s[x, y]^\gamma. \quad (7.12)$$

Class name: **SSIM**

- **Mutual Information (MI)**: was presented as image similarity measure for multimodal image registration by [Viola and Wells \(1997\)](#); [Wells et al. \(1996\)](#). It is defined as

$$\mathcal{S}_{\text{MI}}(I_F, I_M, f) := H(I_F) + H(I_M \circ f) - H(I_F, I_M \circ f), \quad (7.13)$$

where  $H(\cdot)$  is the marginal entropy and  $H(\cdot, \cdot)$  the joint entropy. Class name: **MI**

- **Normalized Gradient Fields (NGF)**: is a image similarity measure defined as

$$\mathcal{S}_{\text{NGF}}(I_F, I_M, f) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \|n(I_F, x) \times n(I_M \circ f, x)\|^2, \quad (7.14)$$

with

$$n(I, x)_\mathcal{E} := \frac{\nabla I(x)}{\|\nabla I(x)\|_\mathcal{E}}, \quad (7.15)$$

$$\|\nabla I(x)\|_\mathcal{E} := \sqrt{\nabla I(x)^\top \nabla I(x) + \mathcal{E}^2}, \quad (7.16)$$

for multimodal image registration developed by [Haber and Modersitzki \(2006\)](#).

For the estimation of  $\mathcal{E}$ , [Haber and Modersitzki \(2006\)](#) propose

$$\mathcal{E} = \frac{\eta}{|X|} \sum_{x \in X} |\nabla I(x)|, \quad (7.17)$$

where  $\eta$  is the estimated noise level. Class name: NGF

## Transformation Models

AIRLab supports three major types of transformation models: linear/dense, non-linear/interpolating and dense models (hybrid models are planned).

### Linear/dense

Currently, AIRLab supports rigid, similarity and affine transformations for 2D and 3D image data.

Class name: `RigidTransformation`,  
`SimilarityTransformation`, `AffineTransformation`

### Non-linear/interpolating

as mentioned with Equation (7.5), non-linear/interpolating models have fewer control points as image points are available. The displacement  $f(x)$  for a given point  $x$  in the image is interpolated from neighboring control points by the respective basis function. In AIRLab, two exemplary basis functions are implemented:

- B-spline: the standard B-spline kernel, which is used in the Free Form Deformation (FFD) algorithm of [Rueckert et al. \(1999\)](#)

$$k_{B_{1D}}(x, y) := \begin{cases} \frac{2}{3} - |r|^2 + \frac{|r|^3}{2}, & 0 \leq |r| < 1 \\ \frac{(2-|r|)^3}{6}, & 1 \leq |r| < 2 \\ 0, & 2 \leq |r|, \end{cases} \quad (7.18)$$

$$r = x - y. \quad (7.19)$$

In addition, AIRLab supports B-spline kernels of arbitrary order (first order are used in [Vishnevskiy et al. \(2016\)](#) and third order in the FFD [Rueckert et al. \(1999\)](#)). An order  $p$  is derived by convolving the zeroth order B-spline  $p + 1$  times with it self:

$$B_0(r) := \begin{cases} 1 & |r| < \frac{\delta}{2} \\ 0 & \text{otherwise} \end{cases} \quad (7.20)$$

$$B_i := B_0 * B_{i-1} \quad (7.21)$$

where  $B_3$  corresponds to  $k_{B_{1D}}$  and  $*$  is the convolution. The control points have a spacing of  $\delta$  which implicitly defines the extent of the kernel. With increasing

order, the control point support of the kernel is increased by one for each additional order.

Class name: `BsplineTransformation`

- Wendland: a family of compact radial basis functions, which is used for image registration in [Jud et al. \(2016a,b\)](#). AIRLab supports a Wendland kernel which is in  $C^4$ :

$$k_W(x, y) = \psi_{3,2} \left( \frac{\|x - y\|}{\sigma} \right), \quad (7.22)$$

$$\psi_{3,2}(r) = (1 - r)_+^6 \frac{3 + 18r + 35r^2}{3}, \quad (7.23)$$

where  $a_+ = \max(0, a)$  and  $\psi_{3,2}$  is the Wendland function of the second kind and positive definite in  $d \leq 3$  dimensions. The scaling  $\sigma$  can also be provided for each space dimension separately to achieve an anisotropic support.

Class name: `WendlandKernelTransformation`

For the non-linear/interpolating transformation models, the transposed convolution is applied (cf. [Dumoulin and Visin \(2016\)](#)) which is available in PyTorch. It is an up-sampling operation where the interpolation kernel can be provided. That means in our case, the control points are “up-sampled” and interpolated using the basis function of choice.

#### Non-linear/dense

the simpler model is the dense transformation model, where each point in the image can be independently transformed. That means, there are  $nd$  parameters (number of image points times number of space dimensions). To achieve a meaningful transformation, strong regularization is required.

#### Image Warping

To compare the transformed moving image with the fixed image within the similarity measures the coordinate system of the moving image has to be warped. As it is mostly done in image registration, AIRLab performs backward warping. That means, the transformation is defined on the fixed image domain where the displacement vectors point to the corresponding points in the moving image. To transform the moving image, it is backward warped into the coordinate system of the fixed image. This prevents holes occurring in the warped image.

The warping is performed in normalized coordinates in the interval  $[-1, 1]^d$ . The points which are transformed out of the fixed image region are identified by checking if  $x + f(x)$  falls outside the normalized interval. For illustration please see following snippet:

## 7 AIRLab: Autograd Image Registration Laboratory

```
(...)
displacement = self._grid + displacement

mask = th.zeros_like(self._fixed_image.image,
                      dtype=th.uint8, device=self._device)
for dim in range(displacement.size()[-1]):
    mask += displacement[... , dim].gt(1) +
            displacement[... , dim].lt(-1)

mask = mask == 0
(...)
```

Because displaced points not necessarily fall onto the pixel-grid, interpolation is required. Currently, AIRLab supports linear interpolation while B-spline interpolation is planned as up-coming feature. The warping is performed by the grid sampler of PyTorch which utilizes the GPU.

### Diffeomorphic Transformation

Diffeomorphic transformations models are often used in medical image registration because of their topology preserving characteristics. These types of transformations defining a bijective transformation between the fixed image domain and the moving image domain. First approaches of diffeomorphic image registration were presented in [Christensen et al. \(1996\)](#); [Dupuis and Grenander \(1998\)](#); [Trouvé \(1995\)](#). With this the large deformation diffeomorphic metric mapping (LDDMM) was presented by [Beg et al. \(2005\)](#). The LDDMM method possesses a high computational complexity, due to the time dependent velocity used for the calculation of the final transformation. In order to reduce the computational complexity the usage of a stationary velocity field was presented by [Arsigny et al. \(2006\)](#); [Ashburner \(2007\)](#); [Hernandez et al. \(2007\)](#). The final transformation is then defined as

$$f = \exp(v), \quad (7.24)$$

where  $\exp(\cdot)$  defines the matrix exponential and  $v : \mathcal{X} \rightarrow \mathbb{R}^d$  the input vector field. In this setting the inverse transformation  $f^{-1}$  can be obtained by

$$f^{-1} = \exp(v)^{-1} = \exp(-v). \quad (7.25)$$

In AIRLab diffeomorphic transformation are supported for *all* interpolating and dense transformation models. We based our implementation in AIRLab on previous implemented diffeomorphic transformations [Abadi et al. \(2016\)](#); [Krebs et al. \(2018a\)](#).

### Different Image Domain Size

The registration of images with different image domains is a common problem in the field of medical image registration. Such problems occur for example if the image modality of both images differ. We consider two image domains as different, if the extent or the spacing of the fixed and the moving image are different. Handling different image domain



is challenging in a pixel-level environment due to the fact that we directly operate on the pixel level. In order to handle images with different image domains, we resample both image to the same pixel spacing and extend the image size if needed. The computational complexity, which is increased by extending the image size is normally negligible due to the highly optimized operations used on the GPU.

### Regularization

There are three different types of regularization terms in AIRLab. (I) Regularizers on the displacement field  $f$  commonly used in FFD registration, (II) regularizers on the parameters of  $f$  elaborated in [Jud et al. \(2016a,b\)](#); [Vishnevskiy et al. \(2016\)](#) and (III) the Demons regularizers which regularize the displacement field  $f$  by filtering it in each iteration. Note that Demons regularizers are not differentiated, because in Demons approaches the optimization is an iteration scheme where the image forces (gradient of similarity measure) are evaluated to update the current displacement field and alternately the displacement field is regularized using filtering. We first list the regularization terms which operate on the displacement field  $f$ .

- Diffusion: a regularizer which penalizes changes in the transformation  $f$

$$\mathcal{R}_{\text{diff}} := \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \sum_{i=1}^d \|\nabla f_i(x)\|_2^2. \quad (7.26)$$

Class name: `DiffusionRegulariser`

- Anisotropic Total Variation: a regularizer which favours piece-wise smooth transformations  $f$

$$\mathcal{R}_{\text{anisoTV}} := \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \sum_{i=1}^d |\nabla f_i(x)|. \quad (7.27)$$

It is anisotropic which means its influence is aligned to the coordinate axes.

Class name: `TVRegulariser`

- Isotropic Total Variation: the isotropic version of the anisotropic regularizer

$$\mathcal{R}_{\text{isoTV}} := \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \|\nabla f(x)\|_2. \quad (7.28)$$

Both TV regularizers are not differentiable, therefore, the subgradient of zero is taken at zero.

Class name: `IsotropicTVRegulariser`

- Sparsity: a regularizer which penalizes non-zero parameters

$$\mathcal{R}_{\text{sparse}} := \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \|f(x)\|_1. \quad (7.29)$$

Class name: `SparsityRegulariser`

### Regularizers on Parameters

The listed regularization terms are also available for regularizing the parameters of  $f$ . The parameters which should be regularized are passed to the regularizer as an array, a name and a weighting. In this way, one can individually weight subsets of parameters, belonging for example to different hierarchical levels, cf. the following example:

```
(...)
reg_param = paramRegulariser.L1Regulariser(
    "trans_parameter",
    weight=weight_parameter[level])

registration.set_regulariser_parameter([reg_param])
(...)
```

### Demons Regularizers

Currently, there are two Demons regularizers available in AIRLab:

- Kernel: an arbitrary convolution kernel for filtering the displacement field. An example is the Gaussian kernel which is used originally in the Demons algorithm [Thirion \(1998\)](#).  
Class name: `GaussianRegulariser`
- Graph Diffusion: the diffusion is performed by spectral graph diffusion. The graph can be utilized in order to handle the sliding organ problem. In this case, the graph is built during the optimization as proposed by [Sandkühler et al. \(2018b\)](#).  
Class name: `GraphDiffusionRegulariser`

### Optimizers

AIRLab includes a rich family of optimizers which are available in PyTorch including LBFGS, ASGD and Adam. They are tailored to optimize functions with a high number of parameters and thus are well suited for non-linear image registration objectives. We refer to [Ruder \(2016\)](#) for a detailed overview of first order gradient based optimizers. As PyTorch also supports *no-grad* computations, iteration schemes as used in the Demons algorithm are also supported. The following snippet is an example usage of *no-grad* taken from the Demons regularizer.

```
(...)
def regularise(self, data):
    for parameter in data:
        # no gradient calculation for the
        # demons regularisation
        with th.no_grad():
            self._regulariser(parameter)
(...)
```

**Table 7.1:** Execution statistics of 2D images with different sizes (pixels) in seconds.

Experiment	Hardware	64	128	256	512	1024	2048	4096
Dense + $\mathcal{R}_{\text{diff}}$ + $\mathcal{S}_{\text{MSE}}$	CPU	2.29	3.29	5.57	14.11	46.44	187.68	832.77
Dense & Diffeomorph + $\mathcal{R}_{\text{diff}}$ + $\mathcal{S}_{\text{MSE}}$	CPU	4.52	9.75	25.81	81.55	309.06	1367.49	5834.58
Dense + $\mathcal{R}_{\text{diff}}$ + $\mathcal{S}_{\text{MSE}}$	GPU GTX 1080	4.41	4.39	4.39	4.29	4.63	9.21	30.91
Dense & Diffeomorph + $\mathcal{R}_{\text{diff}}$ + $\mathcal{S}_{\text{MSE}}$	GPU GTX 1080	7.36	7.32	6.59	6.61	9.14	24.34	89.85

### Registration Evaluation

Performance evaluation of the developed registration method is essential. Due to the fact that ground truth transformations are highly difficult to obtain especially for medical images other performance measures are used. Over the last years the evaluation of the registration is performed on a selected set of corresponding landmarks. Several datasets have been provided to evaluate registration algorithm. Here, the POPI [Vandemeulebroucke et al. \(2007\)](#) or the DirLab [Castillo et al. \(2009\)](#) dataset are mostly used in the past. These datasets contain 3D CT images of the upper thorax and corresponding landmarks. However, normally the developer is responsibly to implement the necessary functions for the pre-processing of the image data or the evaluation of the landmarks. In AIRLab, we provide an evaluation pipeline for download, pre-processing, and evaluation. This means in detail, that the developer can plug in the new registration algorithm and AIRLab take care of the complete evaluation process. For the comparison of corresponding landmarks, we use the mean square distance of the landmark positions. The included automatic evaluation pipeline makes it very simple to compare various versions of the registration algorithm during the development. To the best of our knowledge, AIRLab is the first registration framework that provide such an automatic evaluation pipeline. The automatic evaluation is done for the POPI dataset [Vandemeulebroucke et al. \(2007\)](#).

### Upcoming Features

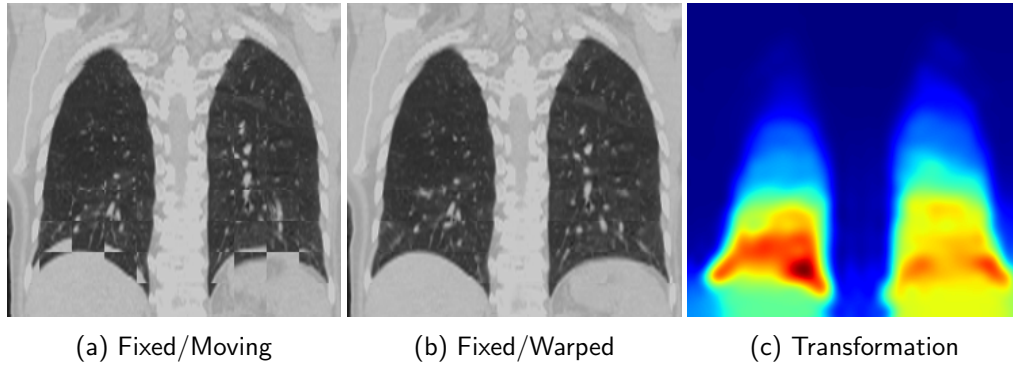
In this section, we list the features which did not make it into the present version, which however are planned for integration into AIRLab soon.

- Interpolation: B-spline interpolation for the image warping.
- Registration: Registration of images with more than one color channel, e.g., RGB images.
- Registration: Extending AIRLab for learning-based registration.

### Experiments

In this section, we provide image registration examples. We have implemented two classic registration algorithms within AIRLab and show their qualitative performance on synthetic examples and on a DirLab dataset [Castillo et al. \(2009\)](#). Quantitative analyses will follow in the final version of this paper.

## Image Registration Algorithms



**Figure 7.2:** FFD registration result. (a) Fixed image and moving image as checkerboard, (b) fixed image and warped moving image as checkerboard and (c) final transformation visualized as the magnitudes of the displacements.

The following algorithms have been implemented:

- *Rigid*: a simple objective with a similarity transformation has been set up, where the  $\mathcal{S}_{\text{MSE}}$  similarity metric has been optimized with Adam. Two AIRLab images have been registered, where the moving image has been rotated, translated, and scaled. In Figure 7.1, the registration result is depicted.
- *FFD*: the Free Form Deformations algorithm [Rueckert et al. \(1999\)](#) was implemented. As in the original paper, a third order B-spline kernel has been used for the parametric transformation model. Furthermore, the  $\mathcal{S}_{\text{NCC}}$  similarity measure with the  $\mathcal{R}_{\text{anisoTV}}$  regularizer on the displacement field have been applied. The overall objective has been optimized with Adam. For the experiment, an image pair of the DirLab [Castillo et al. \(2009\)](#) has been registered. To illustrate the result, in Figure 7.2, a slice through the volume is visualized. A multi-resolution strategy has been implemented performing  $\{300, 200, 50\}$  iterations for the *FFD* algorithm. The detailed parameter configuration can be found in the source-code.
- *Diffeomorphic*: the Demons algorithm [Thirion \(1998\)](#) was implemented using the  $\mathcal{S}_{\text{MSE}}$  similarity measure with the Gaussian Demons regularizer. Furthermore, we used the diffeomorphic option of the transformation. The Diffeomorphic Demons algorithm has been applied to the circle and C example. For better illustration, see Figure 7.3 (d)-(g), a shaded circle has been warped with the final transformation. In addition, we applied a diffeomorphic B-spline registration to the circle and C example. As similarity measure also the  $\mathcal{S}_{\text{MSE}}$  was used. The results are shown in Figure 7.3 (h)-(k), a shaded circle has been warped with the final transformation.

The following snippet illustrates how to setup a registration algorithm in AIRLab with the *Rigid* registration example:

## 7 AIRLab: Autograd Image Registration Laboratory

```
(...)
# all imports
registration = PairwiseRegistration(dtype=dtype,
                                   device=device)

# choose the rigid transformation model
transformation = SimilarityTransformation(moving_image,
                                           opt_cm=False)

# initialize the translation with the center of mass
# of the fixed image
transformation.init_translation(fixed_image)

registration.set_transformation(transformation)

# choose the Mean Squared Error as image loss
image_loss = MSELoss(fixed_image, moving_image)
registration.set_image_loss([image_loss])

# choose the Adam optimizer to minimize the objective
optimizer = th.optim.Adam(
    transformation.parameters(), lr=0.01)
registration.set_optimizer(optimizer)
registration.set_number_of_iterations(1000)

# start the registration
registration.start()

# warp the moving image with the final transformation result
displacement = transformation.get_displacement()
warped_image = warp_image(moving_image, displacement)
(...)
```

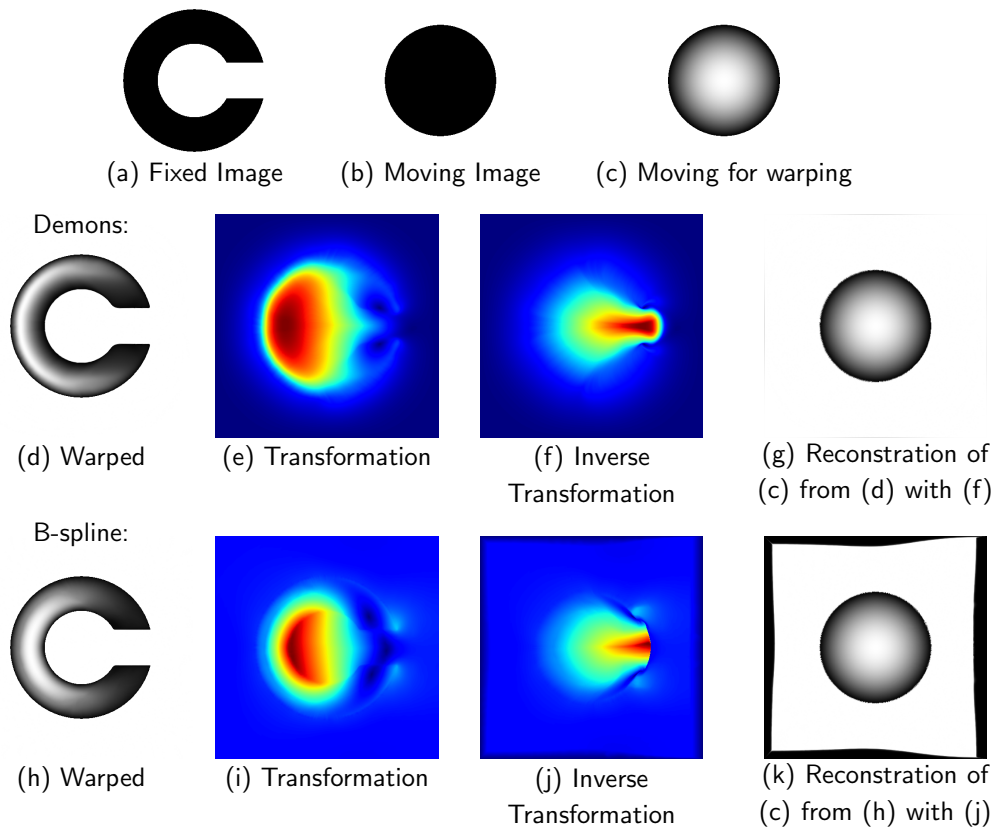
### Performance Analysis

All experiments have been conducted using an NVIDIA GeForce GTX 1080 GPU. We evaluate the performance of AIRLab for different image sizes and different computation hardware. Furthermore, we evaluated the computational effort for the diffeomorphic transformation compared to the non-diffeomorphic transformation models. The performance comparison of CPU and GPU is listed in Table 7.1.

Because the **GaussianRegulariser** is not differentiated, there is less computational time spent by autograd for the *Demons* example.

## 7.4 Conclusion

We have introduced AIRLab, an environment for rapid prototyping and reproduction of medical image registration algorithms. It is written in the scripting language Python and



**Figure 7.3:** (a) Fixed C image, (b) moving circle image and (c) shaded circle image. Results for the diffeomorphic demons((d)-(g)): (d) warped shaded circle, (e) final transformation and (f) inverse transformation visualized as the magnitudes of the displacements and (g) reconstruction of the moving image with the inverse transformation. ((h)-(k)) shows the result for the diffeomorphic B-spline registration method.

## 7 AIRLab: Autograd Image Registration Laboratory

heavily uses functionality of PyTorch. The unique feature compared to existing image registration software is the automatic differentiation which fosters rapid prototyping. AIRLab is freely available under the Apache License 2.0 and accessible on GitHub: <https://github.com/airlab-unibas/airlab>.

With AIRLab, we hope that we can make a valuable contribution to the medical image registration community, and we are looking forward to see researchers and developers which actively use AIRLab in their work. Finally, we encourage them also to contribute to future developments of AIRLab.

## Acknowledgments

The authors thank Alina Giger, Reinhard Wendler, and Simon Pezold for their great support.



# Gated Recurrent Neural Networks for Accelerated Ventilation MRI

MRI ventilation and perfusion maps have shown great success in the past. In the following paper, we present a method that is able to reduce the number of images needed to calculate ventilation maps by maintaining the quality of the maps at the same time. Since the images that are acquired to calculate the ventilation maps are acquired as a continuous sequence, we based our model on a stacked bidirectional gated recurrent neural network. The results show that our method is able to calculate ventilation maps by using only 40 images as compared to the current state-of-the-art method with 140 images.

## Publication

This paper was presented at the 10th International Workshop on Machine Learning in Medical Imaging (MLMI) at the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Shenzhen, China, 2019<sup>1</sup>.

---

<sup>1</sup>[https://link.springer.com/chapter/10.1007/978-3-030-32692-0\\_63](https://link.springer.com/chapter/10.1007/978-3-030-32692-0_63)



# Gated Recurrent Neural Networks for Accelerated Ventilation MRI

Robin Sandkühler<sup>1</sup>, Grzegorz Bauman<sup>2</sup>, Sylvia Nyilas<sup>3</sup>, Orso Pusterla<sup>2</sup>, Corin Willers<sup>3</sup>  
 Oliver Bieri<sup>1,2</sup>, Philipp Latzin<sup>3</sup>, Christoph Jud<sup>1</sup>, Philippe C. Cattin<sup>1</sup>

## Abstract

Thanks to recent advancements of specific acquisition methods and post-processing, proton Magnetic Resonance Imaging became an alternative imaging modality for detecting and monitoring chronic pulmonary disorders. Currently, ventilation maps of the lung are calculated from time-resolved image series which are acquired under free breathing. Each series consists of 140 coronal 2D images containing several breathing cycles. To cover the majority of the lung, such a series is acquired at several coronal slice-positions. A reduction of the number of images per slice enable an increase in the number of slice-positions per patient and therefore a more detailed analysis of the lung function without adding more stress to the patient. In this paper, we present a new method in order to reduce the number of images for one coronal slice while preserving the quality of the ventilation maps. As the input is a time-dependent signal, we designed our model based on Gated Recurrent Units. The results show that our method is able to compute ventilation maps with a high quality using only 40 images. Furthermore, our method shows strong robustness regarding changes in the breathing cycles during the acquisition.

## 8.1 Introduction

Physiological lung imaging is a vital examination technique for the early detection and monitoring of chronic pulmonary disorders like cystic fibrosis. Due to the very low proton density of the lung parenchyma, the application of proton Magnetic Resonance Imaging (MRI) for functional pulmonary assessment is challenging. Different methods based on the inhalation of hyperpolarized gases during the MRI acquisition were presented in the past [van Beek et al. \(2004\)](#). However, broad clinical application of hyperpolarized gas MRI is not feasible as it necessitates specialized equipment and personal. Furthermore, it requires specific breathing manouvers and the cooperation of the patients, which can not always be granted especially in pediatric subjects.

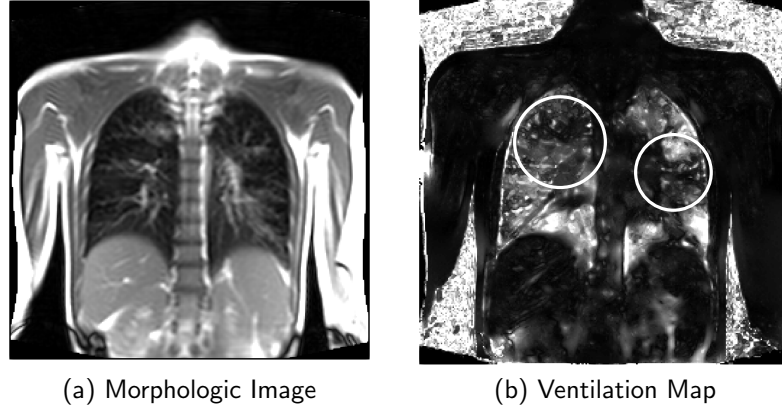
Recently, rapid acquisition pulse sequences such as ultra-fast balanced steady-state free precession (uf-bSSFP) [Bauman et al. \(2016\)](#) have shown promise for proton MRI of pulmonary structures and functions. Standard and widely available proton MRI has

---

<sup>1</sup>Department of Biomedical Engineering, University of Basel, Allschwil, Switzerland

<sup>2</sup>Division of Radiological Physics, Department of Radiology, University of Basel Hospital, Basel, Switzerland

<sup>3</sup>Pediatric Respiratory Medicine, Department of Pediatrics, Inselspital, Bern University Hospital, University of Bern, Bern, Switzerland



**Figure 8.1:** (a) Sample coronal image from an MR acquisition with the uf-bSSFP sequence of a patient with cystic fibrosis, and (b) a corresponding ventilation map . The circles mark areas with potential ventilation defects.

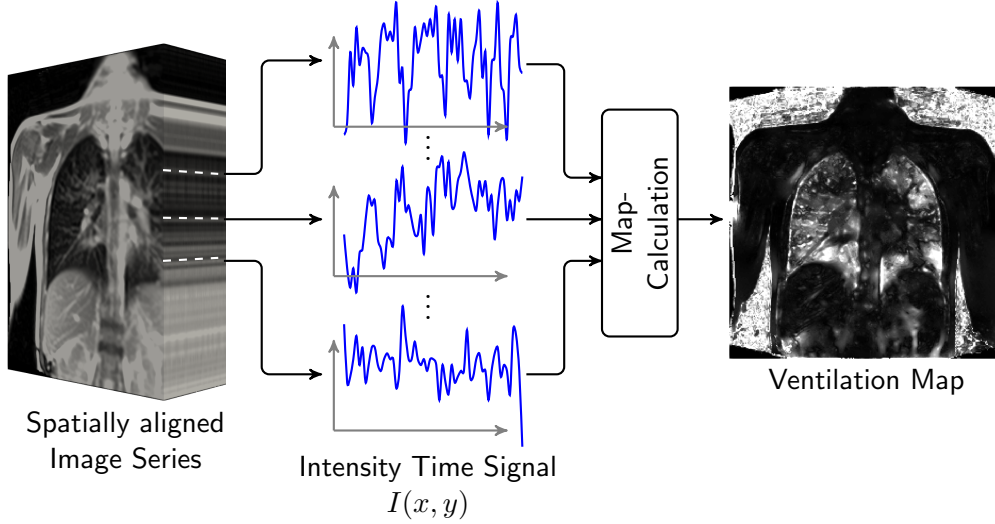
the potential to become a viable radiation-free alternative imaging modality compared to CT or SPECT to visualize physiological properties of the lung. MR ventilation maps of the lung can be calculated from a coronal time-resolved 2D image series acquired during free breathing. At every breath, the lung expands and contracts creating signal modulations associated with pulmonary ventilation that are detectable by MRI. After nonrigid image registration [Sandkühler et al. \(2018b\)](#) to align the pulmonary structures in the image series, the corresponding frequency for the ventilation can be detected. Figure 8.1 shows a representative MR image acquired with the uf-bSSFP sequence and the corresponding ventilation map. The in [Bauman and Bieri \(2017\)](#) presented state-of-the-art method for the calculation of ventilation maps (Figure 1) requires 140 images per coronal slice-position in order to compute the ventilation map. The acquisition time per image series is about one minute. A reduction of the number of images per slice enable an increase in the number of slices per patient within the same examination time. This is important, as an increase of the overall scan time could increase the stress for the patient inside the MR, especially for pediatric patients.

In this paper, we present a novel method for the estimation of ventilation maps using an image series composed of only 40 images. We trained a stacked bidirectional Gated Recurrent Units (SB-GRU) to estimate the ventilation maps based on the time-varying intensity signal of the image series. The results show that the presented method can compute accurate ventilation maps using less than a third of the images.

## 8.2 Background

MRI image acquisition is performed with a time-resolved 2D uf-bSSFP sequence [Bauman et al. \(2016\)](#). This pulse sequence is accelerated thanks to optimized excitation pulses and Cartesian gradient switching patterns, accompanied by partial echo readouts and

ramp sampling techniques. As a result echo time (TE) and repetition time (TR) are shortened which improves signal in the lung parenchyma and reduces motion as well as off-resonance artifacts known as banding artifacts. The chest volume was covered using 8 to 12 coronal slices with 140 images per slice position resulting in a total acquisition time of about 10 minutes per examination.



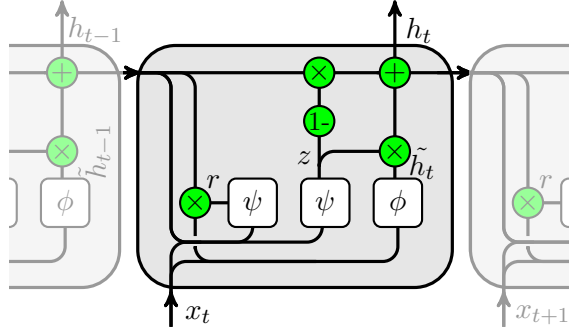
**Figure 8.2:** Computation process of MRI ventilation maps.

### Ventilation MRI

The ventilation maps are computed according to [Bauman and Bieri \(2017\)](#). In the first step, the respiratory motion in the time-resolved uf-bSSFP data was compensated with a two-dimensional nonrigid image registration algorithm [Sandkühler et al. \(2018b\)](#). Subsequently, an automated lung segmentation was performed [Andermatt et al. \(2016\)](#). After image registration and segmentation, the matrix pencil analysis of the time course was performed voxel-wise to estimate the amplitudes  $A_r$  of the respiratory frequencies of the signal modulations in the lung parenchyma. The estimated amplitudes were used to calculate fractional ventilation maps

$$v_m = \frac{A_r}{A_{DC} + 0.5A_r - BG} 100[\%], \quad (8.1)$$

where  $A_{DC}$  is the amplitude of the baseline signal and  $BG$  the background noise. An overview of the calculation process is shown in Figure 8.2. For further details, the reader is referred to [Bauman and Bieri \(2017\)](#).



**Figure 8.3:** Unrolled gated recurrent unit cell.

### 8.3 Method

The image series that is acquired for the ventilation maps contains time related information, i.e., the signal intensity changes over time. In order to take this time information into account, we use a Recurrent Neural Network (RNN) for the computation of ventilation maps with only 40 images in a series. We choose 40 images to ensure that the input signal contains at least 2-3 respiratory cycles. Compared to feedforward networks, RNNs contain an internal state and the output of an RNN depends on the current input and the internal state. A simple RNN has the form

$$h_t = \phi(Wx_t + Uh_{t-1}), \quad (8.2)$$

where  $x_t$  is the input,  $h_t$  the output of the RNN at time  $t$ , and  $\phi$  is an activation function. Here, the internal state is the last output  $h_{t-1}$  of the RNN. The matrices  $W$  and  $U$  are the learnable weights. Based on the basic RNN (8.2), several extensions were developed to overcome characteristic issues, e.g., the vanishing gradient problem. A popular RNN extension is the Long Short-Term Memory (LSTM) network [Hochreiter and Schmidhuber \(1997\)](#).

#### Gated Recurrent Units

A simplified version of the LSTM are GRUs [Cho et al. \(2014\)](#). GRUs show a good performance compared to the LSTM but with fewer parameters. According to [Cho et al. \(2014\)](#), a vanilla GRU cell is described by:

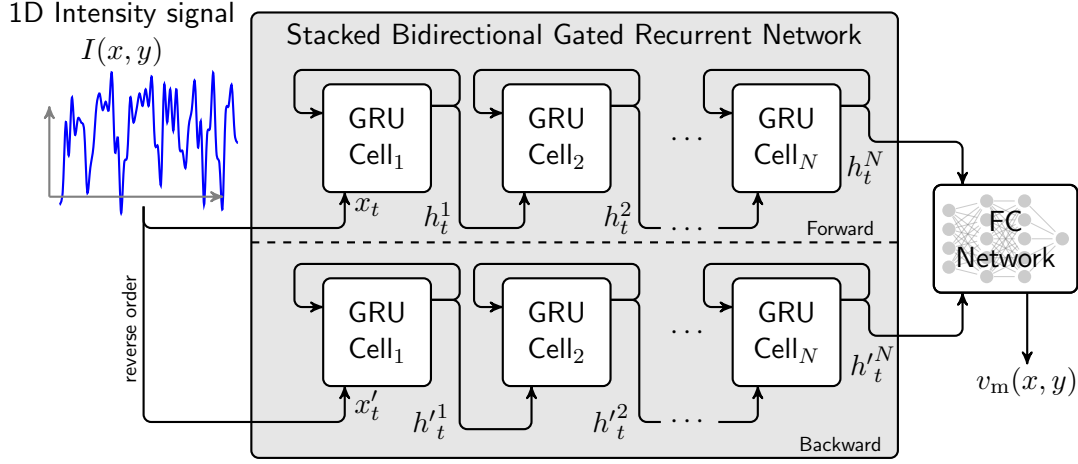
$$r_t = \psi(W_r x_t + U_r h_{t-1}), \quad (8.3)$$

$$z_t = \psi(W_z x_t + U_z h_{t-1}), \quad (8.4)$$

$$\tilde{h}_t = \phi(W x_t + U(r_t \odot h_{t-1})), \quad (8.5)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t. \quad (8.6)$$

Here,  $r$  represents the reset gate,  $z$  the update gate,  $\tilde{h}$  the proposal state,  $h_t$  the output at time  $t$ ,  $\phi(\cdot)$  is the hyperbolic tangent,  $\psi(\cdot)$  represents the logistic function, and  $\odot$  is



**Figure 8.4:** Stacked Bidirectional Recurrent Neural Network architecture for the computation of ventilation maps.

the Hadamard product. Each gate contains its own learnable weights  $W$  for the input  $x_t$  and  $U$  for the hidden state  $h_{t-1}$ . A graphical representation of the Equations (8.3)-(8.6) is shown in Figure 8.3, where the circles describe element wise operations.

### Stacked Bidirectional Gated Recurrent Units

Bidirectional GRUs consists of two separate GRU networks, with a forward and backward direction respectively. In the backward direction, the input sequence is processed in reverse order. This allows an analysis of the time signal independent of the direction of the signal. Furthermore, we use a stacked bidirectional GRU (SB-GRU), in order to model more complex signals. A stacked GRU is defined as a concatenation of a number of single GRU cells, where the input of the current GRU cell is the output of the previous one. In this paper, we used a GRU cell given by

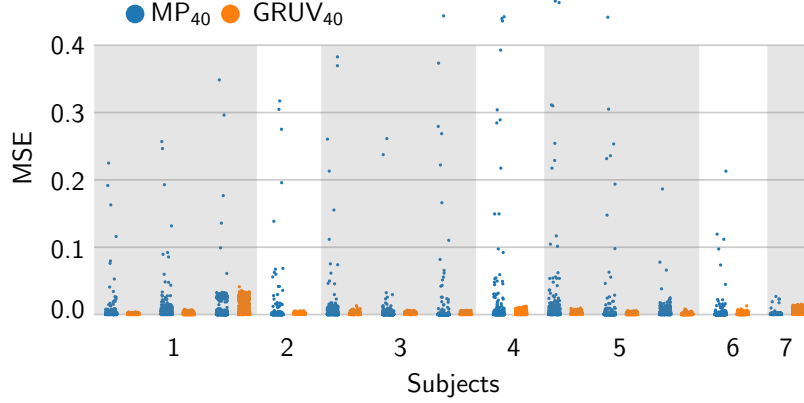
$$r_j = \psi \left( W_r^j x_t + b_{wr}^j + U_r^j h_{t-1}^j + b_{hr}^j \right), \quad (8.7)$$

$$z_j = \psi \left( W_z^j x_t + b_{wz}^j + U_z^j h_{t-1}^j + b_{hz}^j \right), \quad (8.8)$$

$$\tilde{h}_t^j = \phi \left( W^j x_t + b_x^j + r_j \odot \left( U^j h_{t-1}^j + b_h^j \right) \right), \quad (8.9)$$

$$h_t^j = (1 - z_j) \odot h_{t-1}^j + z_j \odot \tilde{h}_t^j, \quad (8.10)$$

where  $j$  is the index of a GRU cell and  $b$  the corresponding bias. The input of our network is a 1D intensity time signal  $I(x, y)$  from the spatially aligned image series as shown in Figure 8.2. After the input is processed by the SB-GRU, the output of the forward and the backward GRU are concatenated and applied to a fully connected network with one hidden layer in order to reduce the output to the scalar ventilation map value  $v_m(x, y)$ .



**Figure 8.5:** MSE values for the ventilation map calculated with the presented method  $\text{GRUV}_{40}$  and the state of the art method  $\text{MP}_{40}$  using a 2D time series with 40 images. Both methods were compared against the ventilation maps of the gold standard matrix pencil method  $\text{MP}_{140}$  using 140 images.

Here  $x$  and  $y$  are the spatial locations of the image series. A complete 2D ventilation map is created by evaluating the network at each spatial location. The final network is shown in Figure 8.4.

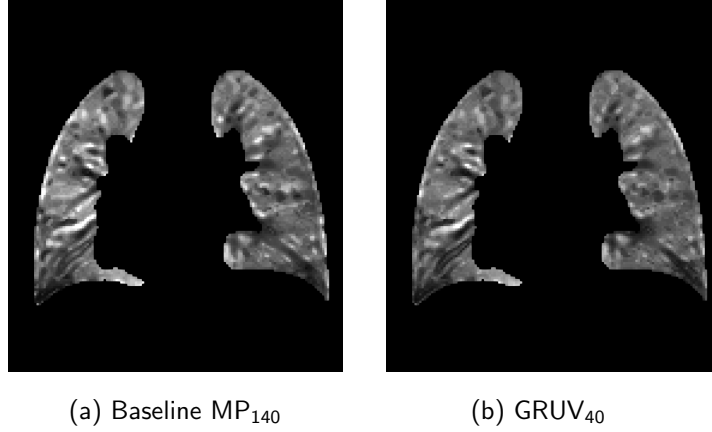
## 8.4 Experiments and Results

For our experiments, we used a data set of 84 subjects with an average of two examinations per subject. In each examination, 8-10 dynamic 2D image series at different coronal positions were acquired. We divided our data set into a training set with 70 subjects, an evaluation set with 7 subjects, and a test set with 7 subjects. All examinations were performed on a 1.5 Tesla whole-body MR-scanner (MAGNETOM Aera, Siemens Healthineers, Erlangen, Germany) using a 12-channel thorax and a 24-channel spine receiver coil array. Each subject was scanned during free breathing with the uf-bSSFP sequence [Bauman et al. \(2016\)](#). The main pulse sequence parameters were as follows: field-of-view  $375 \times 375 - 425 \times 425 \text{ mm}^2$ , slice thickness 12 mm, TE/TR/TA = 0.67 ms/1.46 ms/119 ms, flip angle  $60^\circ$ , bandwidth 2056 Hz/pixel, matrix  $128 \times 128$  (interpolated to  $256 \times 256$ ), 3.33 images/s, parallel imaging GRAPPA factor 2.

We used 10 stacked GRU cells with a hidden state size of 300 for the forward and the backward GRU. The hidden layer of the fully connected network was set to 1024 with the RELU activation function and the logistic function  $\frac{1}{1+\exp(-x)}$  as activation function for the last layer. For the optimization of the network parameters, we used the Adam optimizer [Kingma and Ba \(2014\)](#) with a learning rate of 0.0001.

For the training of the network, we used spatially aligned image series with 140 images. The ventilation maps used as gold standard for the training and the lung segmentation





**Figure 8.6:** (a) Example MRI ventilation maps of a single slice of the gold standard (140 images), and (b) the presented method GRUV (40 images).

were obtained as described in Section 8.2. In each training iteration, a continuous image sequence of 40 images was randomly selected from one slice of the training set. The 1D intensity signals  $I(x, y)$  of this sequence were applied to the network. During training, we restrict  $x$  and  $y$  only to locations inside the lung mask. We use the mean squared error (MSE) loss function to compare the network output with the ground truth during training.

For the evaluation of our method, we used a window of the first 40 images of the complete image series from one coronal slice and computed the ventilation maps using the presented method (GRUV<sub>40</sub>) and the matrix pencil method (MP<sub>40</sub>). Both methods were compared to the gold standard matrix pencil method (MP<sub>140</sub>) [Bauman and Bieri \(2017\)](#) which uses 140 images of the series for the computation of the ventilation maps. For the error calculation of the MP<sub>40</sub> and the GRUV<sub>40</sub> when compared to the MP<sub>140</sub> we used MSE metric. To show that our method is robust against breathing variability within the 40 images, the window was shifted for one image and the ventilation maps were computed. This process was repeated until the end of the image series was reached and was done for all coronal slices of each examination. The results for the 7 test subjects are shown in Figure 8.5. Each subject contains either one or three examinations. The results show that the MP<sub>40</sub> and the GRUV<sub>40</sub> method are able to calculate correct ventilation maps. However, the error variance of our method is much smaller compared to the MP<sub>40</sub> method. This indicates that our method is robust against breathing variability within the 40 image window. Qualitative results GRUV<sub>40</sub> are shown in Figure 8.6. Here, we can observe a lower amplitude for the map intensities for the GRUV<sub>40</sub> when compared to the MP<sub>140</sub> result. The reason for this effect could be the fact that only 1D time signals are used to compute the corresponding ventilation value.

## 8.5 Conclusion

In this paper, we presented a novel method for the calculation of ventilation maps based on a 2D time-resolved dynamic MRI image series acquired with the uf-bSSFP sequence. Our aim was to calculate ventilation maps which are equivalent to the one calculated with the state of the art method, but with fewer images. In order to encounter the time dependencies in the image series, we designed a network based on SB-GRUs. We show that our method is able to estimate correct ventilation maps for a given image series with only 40 when compared to the current state-of-the-art method. Furthermore, we observe that our method is robust against breathing variabilities inside the 40 images. But, we also discover for some ventilation maps generated with our method a drop of the global intensity amplitude compared to the state-of-the-art method. We believe, we could overcome this issue by taking spatial information into account but we leave this for future work.

### Acknowledgement

The authors would like to thank the Swiss National Science Foundation for funding this project (SNF 320030\_149576).

# Weakly Supervised Learning Strategy for Lung Defect Segmentation

In the following paper, we address the problem of weakly supervised semantic segmentation. Monitoring the disease progression and the effect of the medication is essential for the treatment of CF patients. The possibility to observe physiological induced changes of the lung was realized with the new MR sequence called ufSSFP. It is now possible to obtain MR ventilation maps based on a dynamic 2D MR image time series. For the clinical routine, the automatic segmentation of impaired areas within these ventilation maps is essential. We developed a weakly supervised segmentation method to segment those impaired areas using only a weak label during training. As a weak label, we use the lung clearance index (LCI). The LCI is a continuous global measure for ventilation inhomogeneities of the lungs. Our method is able to infer a pixel-wise segmentation from the global LCI value. In detail this means that our method does not need any manual labeled data.

## Publication

This paper was presented at the 10th International Workshop on Machine Learning in Medical Imaging (MLMI) at the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Shenzhen, China, 2019<sup>1</sup>.

---

<sup>1</sup>[https://link.springer.com/chapter/10.1007/978-3-030-32692-0\\_62](https://link.springer.com/chapter/10.1007/978-3-030-32692-0_62)



# Weakly Supervised Learning Strategy for Lung Defect Segmentation

Robin Sandkühler<sup>1</sup>, Christoph Jud<sup>1</sup>, Grzegorz Bauman<sup>2</sup>, Corin Willers<sup>3</sup>, Orso Pusterla<sup>2</sup>, Sylvia Nyilas<sup>3</sup>, Alan Peters<sup>4</sup>, Lukas Ebner<sup>4</sup>, Enno Stranziger<sup>4</sup>, Oliver Bieri<sup>1,2</sup>, Philipp Latzin<sup>3</sup>, Philippe C. Cattin<sup>1</sup>

## Abstract

Through the development of specific magnetic resonance sequences, it is possible to measure the physiological properties of the lung parenchyma, e.g., ventilation. Automatic segmentation of pathologies in such ventilation maps is essential for the clinical application. The generation of labeled ground truth data is costly, time-consuming and requires much experience in the field of lung anatomy and physiology. In this paper, we present a weakly supervised learning strategy for the segmentation of defected lung areas in those ventilation maps. As a weak label, we use the Lung Clearance Index (LCI) which is measured by a Multiple Breath Washout test. The LCI is a single global measure for the ventilation inhomogeneities of the whole lung. We designed a network and a training procedure in order to infer a pixel-wise segmentation from the global LCI value. Our network is composed of two autoencoder sub-networks for the extraction of global and local features respectively. Furthermore, we use self-supervised regularization to prevent the network from learning non-meaningful segmentations. The performance of our method is evaluated by a rating of the created defect segmentations by 5 human experts, where over 60% of the segmentation results are rated with very good or perfect.

## 9.1 Introduction

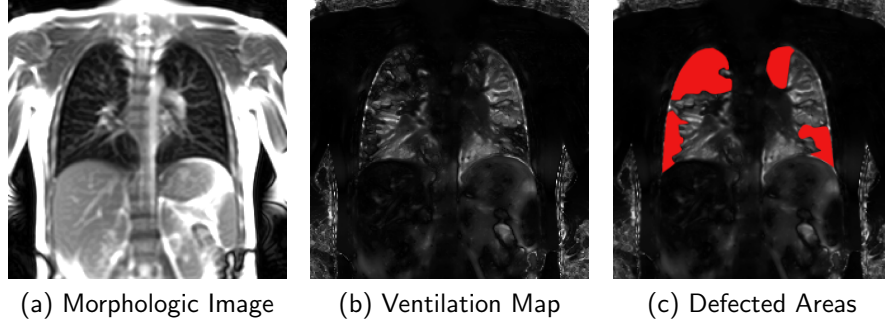
Thanks to the development of dedicated image acquisition methods, MRI has become a new modality allowing for spatial analysis of time-dependent physiological changes of the lung, e.g., ventilation [Bauman et al. \(2016\)](#). The ventilation maps are computed from a time-resolved image series and the detection of the corresponding ventilation frequencies in the image data, after the images are spatially aligned [Bauman and Bieri \(2017\)](#). An example is shown in Figure 9.1. The automatic labeling of defected areas in these maps is a complex task for naïve methods such as thresholding because the value range depends on the relative signal intensity of the MR images. More sophisticated

<sup>1</sup>Department of Biomedical Engineering, University of Basel, Allschwil, Switzerland

<sup>2</sup>Division of Radiological Physics, Department of Radiology, University of Basel Hospital, Basel, Switzerland

<sup>3</sup>Pediatric Respiratory Medicine, Department of Pediatrics, Inselspital, Bern University Hospital, University of Bern, Bern, Switzerland

<sup>4</sup>Department of Diagnostic, Interventional and Pediatric Radiology, Inselspital, Bern University Hospital, University of Bern, Bern, Switzerland



**Figure 9.1:** (a) MRI acquisition of a patient with cystic fibrosis, (b) corresponding ventilation maps, and (c) assumed impaired areas.

methods, e.g., supervised segmentation methods require accurate labeled ground truth data, which is difficult to obtain, even for clinical experts.

Compared to supervised methods, unsupervised methods do not require labeled training data. There are several unsupervised learning methods for medical image segmentation. An unsupervised lesion detection using an adversarial auto-encoder was presented in [Chen and Konukoglu \(2018\)](#). The method does not use any labels during training but requires a training set of image data with only healthy subjects. A weakly supervised method for the segmentation of brain tumors was shown in [Andermatt et al. \(2019\)](#) by using only one *binary label* for each image (healthy, non-healthy). The methods presented in [Papandreou et al. \(2015b\)](#); [Pathak et al. \(2015\)](#) use class level labels of the image content as a weak label in order to create a semantic object segmentation.

In this paper, we present a first *proof-of-concept* for the segmentation of the defected areas in the ventilation maps using a weakly supervised learning strategy based on the Lung Clearance Index (LCI). The LCI is a single value which expresses the healthiness of the whole lung and corresponds to the global ventilation inhomogeneities [Robinson et al. \(2009\)](#). Unlike binary or class labels, the LCI is an *unbounded value* where a value of 6-7 indicates a healthy lung. A higher LCI is correlated to more ventilation inhomogeneities. We designed a network that is able to perform a pixel-wise segmentation using only the LCI value as a weak label during training. Furthermore, we use *self-supervised regularization* to prevent the network from learning a non-meaningful segmentation. The performance of our method is evaluated by a rating of the segmentation results by 5 human experts. Initial results show that over 60% of the segmentation results are scored with very good or perfect.

## 9.2 Method

In this section, we describe our network design and training procedure in order to segment the defected areas in the lung using only the LCI value as a continuous label. For this,

we use the relation

$$\frac{\text{defected lung volume}}{\text{total lung volume}} \propto LCI,$$

which has been proven to be valid [Nyilas et al. \(2017\)](#).

### Lung Clearance Index

The Lung Clearance Index (LCI) is a global measure for the ventilation inhomogeneities of the lung, as a result of respiratory diseases like cystic fibrosis or chronic obstructive pulmonary disease [Robinson et al. \(2009\)](#). It is defined in lung volume turnovers (TO), i.e., the number of breathing cycles that are necessary in order to clear the lung from a previous inhaled tracer gas. A common method for the determination of the LCI is the Nitrogen Multiple Breath Washout (N<sub>2</sub>-MBW). Here the patient starts breathing 100 % oxygen, and the remaining nitrogen concentration of the previously breathed ambient air is measured. If the concentration of the N<sub>2</sub> is below 2.5 % of the starting concentration the test stops. The LCI is then defined by

$$LCI = \frac{V_{CE}}{FRC} \quad FRC = \frac{\text{expired volume N}_2}{c_{\text{start}}^{\text{N}_2} - c_{\text{end}}^{\text{N}_2}}, \quad (9.1)$$

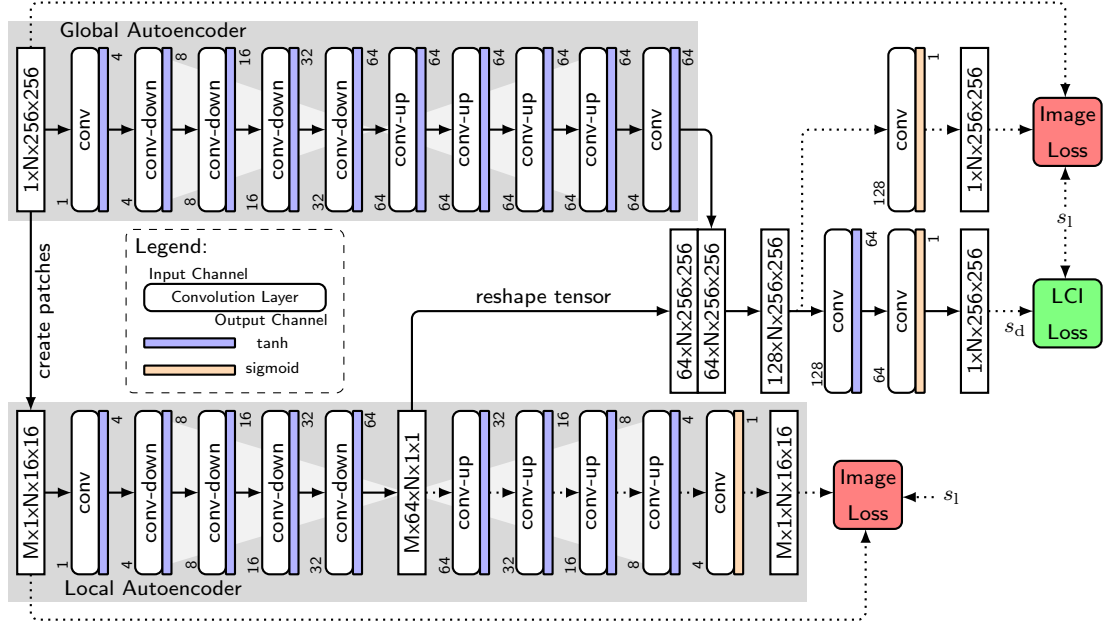
where  $V_{CE}$  is the expired volume during the complete measurement. The LCI for a healthy lung is 6-7 TO. For a more detailed description of the LCI, we refer the reader to [Robinson et al. \(2009\)](#). Here, we normalize the LCI between 0 and 1 by assuming a minimum LCI value of 5 and a maximum LCI value of 20.

### Ventilation Maps

For the calculation of the ventilation maps a 2D dynamic time series with the ultra-fast steady-state free precession pulse sequence (uf-bSSFP) [Bauman et al. \(2016\)](#) is acquired during free breathing. The respiratory motion is compensated with a pairwise non-rigid image registration [Sandkühler et al. \(2018b\)](#) and automated lung segmentation is performed [Andermatt et al. \(2016\)](#). In the final step, the ventilation maps are calculated using a matrix pencil decomposition as described in [Bauman and Bieri \(2017\)](#). To cover the full chest volume, 8-12 ventilation maps with a thickness of 12 mm at different coronal positions are computed. All maps are stacked together to obtain the final image volume with  $N \times 256 \times 256$ , where  $N$  is the number of acquired slices. This image volume is used as input for the network. We need to process all slices at ones, because the LCI value is related to the complete lung and not to a single slice.

### Network Model

An overview of the complete network structure is shown in Figure 9.2. Our network model consists of two major parts. The first part is the global autoencoder (GAE) where the input is the above-described image volume. As the ventilation maps have a slice thickness of 12 mm, neighboring slices can show substantially different content. To



**Figure 9.2:** Model for the weakly supervised segmentation of defected lung areas.

overcome this issue, all operations are performed on a single slice only and not across slices. All convolutional layers use a kernel size of  $1 \times 3 \times 3$ . The **conv-down** block reduces the spatial dimension by using a strided convolution with a stride of  $1 \times 2 \times 2$ . The **conv-up** block increases the spatial resolution by a factor of 2 using bilinear upsampling followed by a convolutional layer. Compared to the classical autoencoder approach the number of feature channels is not reduced in the decoder of the GAE. The input of the second part of the network the local autoencoder (LAE), are overlapping patches of the input volume. For each coronal slice  $M$  2D patches of the input volume with a size of  $16 \times 16$  and a stride of 1 are generated. We consider only patches where the region of the patches overlaps with the lung mask. The size of the latent variable of the LAE is set to 64. In contrast to the GAE, the number of feature channels is reduced in the decoder part of the LAE. For the final pixel-wise defect map, all patches of the input volume are encoded using the encoder of the LAE, reshaped and concatenated with the output of GAE. The embeddings of the patches are placed at the center position of the patch in the image volume. In this way, we get a feature map with the spatial dimensions of the input volume and 128 feature channels. The final network output, the pixel-wise defect segmentation  $s_d$  is obtained after two convolutional layer and has the final size of  $1 \times N \times 256 \times 256$ .



### Network Training

In order to train the parameter of our network with the LCI as weak label, we define the LCI loss as

$$\mathcal{L}_{\text{LCI}}(s_d, s_l, LCI) = \left( \frac{\sum_{x \in \mathcal{X}} s_d(x)}{\sum_{x \in \mathcal{X}} s_l(x)} - LCI \right)^2, \quad (9.2)$$

where  $\mathcal{X} \subset \mathbb{R}^3$  is the domain of the lung mask  $s_l$ , with  $s_l(x) = 1$ . The the output of our network, i.e. the segmentation of the defect areas of the input ventilation map is defined as  $s_d : \mathcal{X} \rightarrow [0, 1]$ . However, the LCI loss is based on the ratio of the defected lung volume to the whole lung volume and thus not a pixel-wise loss. This can lead to a segmentation result that minimize the LCI loss but do not correspond to the defected areas. We use *self-supervised regularization* during training of our network to prevent the network from overfitting on the LCI loss and learning a non-meaningful segmentation. As shown in Figure 9.2, our network contains different paths, some of which are only used for the *self-supervised regularization* during training (dotted) and some are used during training and inference (dense). We use the decoder part of the LAE only during training for the reconstruction of the image patches. With this, we enforce the network to learn an embedding for the classification of the image patch which is strongly related to the image content. The image reconstruction loss for the LAE is defined as

$$\mathcal{L}_{\text{LAE}}(I_{\text{patch}}, \tilde{I}_{\text{patch}}) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \frac{1}{|\mathcal{P}_x|} \sum_{p \in \mathcal{P}_x} (I_{\text{patch}}(x, p) - \tilde{I}_{\text{patch}}(x, p))^2, \quad (9.3)$$

where  $\mathcal{P}_x \subset \mathbb{R}^2$  is the image patch domain at location  $x$  of the input image. For the feature regularization of the GAE the same approach is used, but here the output of the GAE is concatenated with the embedding of the LAE before the image is reconstructed. The global reconstruction loss is defined as

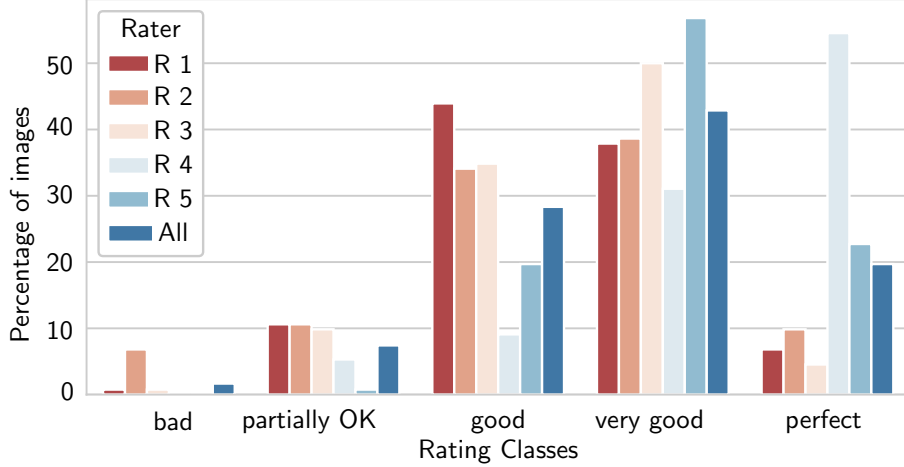
$$\mathcal{L}_{\text{GAE}}(I, \tilde{I}) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} (I(x) - \tilde{I}(x))^2. \quad (9.4)$$

The final loss for the training of the network is then given as

$$\mathcal{L} = \mathcal{L}_{\text{LCI}} + \mathcal{L}_{\text{LAE}} + \mathcal{L}_{\text{GAE}}. \quad (9.5)$$

### 9.3 Experiments and Results

For our experiments, we use a data set of 35 subjects with 2 examinations per subject on average. We divided our data set in a training set with 28 subjects and in a test set with 7 subjects. In each examination 8-12 dynamic 2D image series were acquired. All examinations were performed on a 1.5 Tesla whole-body MR-scanner (MAGNETOM Aera, Siemens Healthineers, Erlangen, Germany) using a 12-channel thorax and a 24-channel spine receiver coil array. Each subject was scanned during free breathing with



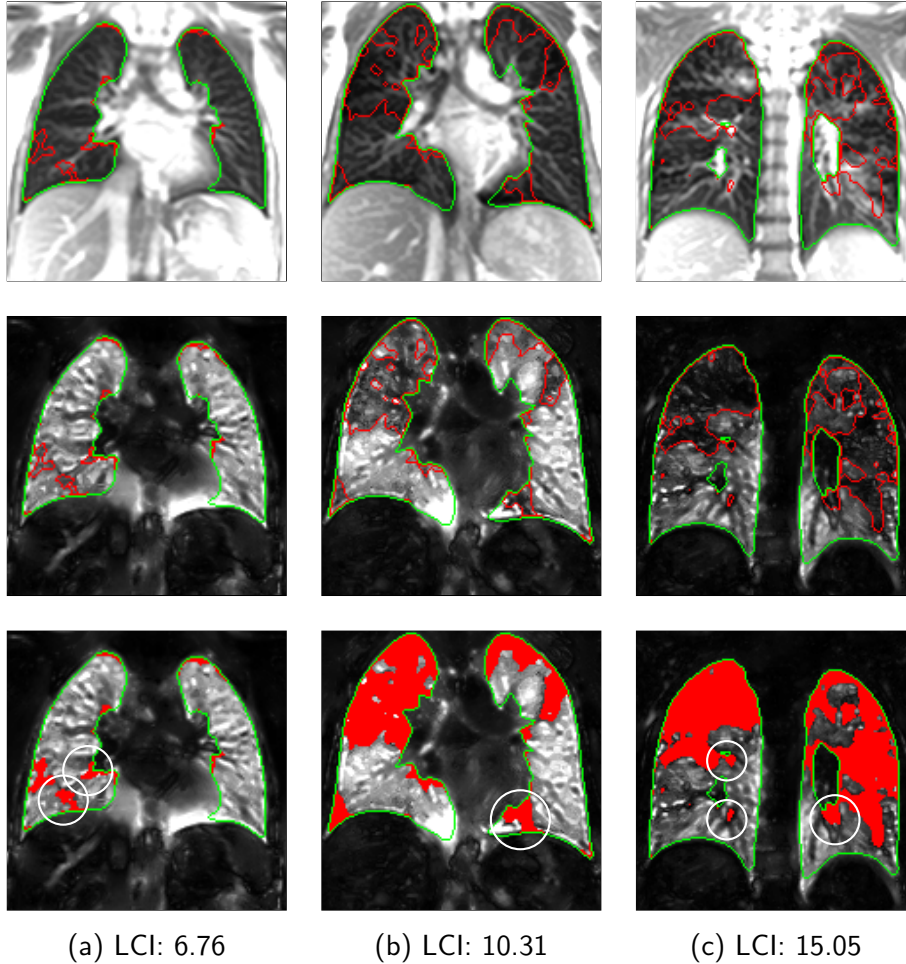
**Figure 9.3:** Result of the human expert rating of the presented defect segmentation.

the uf-bSSFP sequence [Bauman et al. \(2016\)](#). The main pulse sequence parameters were as follows: field-of-view  $375 \times 375 - 425 \times 425 \text{ mm}^2$ , slice thickness 12 mm, TE/TR/TA = 0.67 ms/1.46 ms/119 ms, flip angle  $60^\circ$ , bandwidth 2056 Hz/pixel, matrix  $128 \times 128$  (interpolated to  $256 \times 256$ ), 160 coronal images, 3.33 images/s, parallel imaging GRAPPA factor of 2. Ventilation maps and the lung segmentation were then computed as described in Section 9.2.

Due to the problem of relative values of the ventilation maps caused by the relative values generated by the MR, we normalized each slice independently. We performed a z-normalization followed by clipping of the values at  $[-4, 4]$  and a transformation to  $[0, 1]$ . In the final step, a histogram stretching was performed using the 2 and 98 percentile of the intensities. The normalization was only performed on intensity values inside the lung.

For the evaluation, we applied our method to the 7 subjects of the test set. Because the derivation of pixel-wise manually labeled ground truth remains difficult, we evaluated our method by using a rating of the final defect segmentation by 5 human experts. The rating scheme is defined as the percentage of correct defect segmentation for a given ventilation map: *bad* : 0%-20%, *partially OK* : 21%-40%, *good* : 41%-60%, *very good* : 61%-80%, *perfect* : 81%-100%. For the evaluation the lung ventilation map and the morphological image are available. Each rater scores all 2D segmentation results of the test set, which contains 132 images in total. Rater 1 and 2 are a senior resident radiologist with 20 respectively 10 years of experience. Rater 3 has over 5 years of research experience in the field of MRI lung imaging. Rater 4 is a chest fellow (radiologist) with 4 years of experience in chest imaging. Rater 5 is a physician with 2 years of experience of thorax imaging. The results of the human expert evaluation presented in Figure 9.3, show that over 60% of the segmentation results are rated with very good or perfect. Selected segmentation results of different subjects with different LCIs are shown in Figure 9.4.

The results of our method demonstrate, that we are able to segment most of the impaired areas with ventilation defects by using *only* the LCI for training. However, in some areas an incorrect segmentation is observed in close approximation to blood vessels as shown in Figure 9.4. This might be reasoned by the evidence that the non-ventilated blood vessels are showing the similar characteristics as impaired areas in the ventilation maps.



**Figure 9.4:** Selected defect segmentation results (red) for different subjects with different LCI values. Top row: morphological image with defect segmentation outline. Middle row: ventilation maps with defect segmentation outline. Bottom row: ventilation maps with defect segmentation overlay. Green shows the outline of the given lung segmentation [Andermatt et al. \(2016\)](#). Incorrect defect segmentations are highlighted with white circles.

## 9.4 Conclusion

We presented a first *proof-of-concept* of our *weakly supervised* learning strategy for the segmentation of defected areas in MRI ventilation maps. We designed a network that

## 9 Weakly Supervised Lung Defect Segmentation

is able to generate a pixel-wise segmentation of lung defects in MRI ventilation maps using the LCI value as a weak label. Our network model consists of two major parts for the global and local feature extraction. Both features are then combined to estimate the final segmentation. Furthermore, we use *self-supervised regularization* to prevent the network from learning a non-meaningful segmentation. We evaluated the performance of our method with a rating of the segmentation result by 5 human experts. The results show that over 60% of all segmentations are scored with very good or perfect. However, we observed that our method has a tendency for over-segmentation especially in regions where vessels are located. This over-segmentation could be removed by providing a vessel segmentation as a post-processing step or by adding this to the training procedure, but we will leave this for future work.

### Acknowledgement

The authors would like to thank the Swiss National Science Foundation for funding this project (SNF 320030\_149576).

## Discussion and Conclusion

In this work, we presented different methods to improve the calculation and evaluation of MR ventilation and perfusion maps. With the development of the ufSSFP MR sequence, it is now possible to observe physiological changes based on ventilation and perfusion inside the lungs. Compared to other imaging modalities that are applicable for lung imaging, e.g., X-ray or CT, MRI does not use ionizing radiation. Based on this, MRI is an alternative imaging modality for the detection and long-term monitoring of pulmonary diseases. The process to create ventilation and perfusion maps from a dynamic  $2D + t$  ufSSFP MR sequence consists of three major parts: *MR-acquisition*, *image registration*, and *image analysis*.

This work focuses on the image registration and the image analysis parts. Image registration of the thorax is a challenging task due to sliding-organ motion at the lung cavity and intensity changes. The intensity changes are due to physiological changes inside the tissue caused by ventilation and perfusion. We developed a computational efficient registration method based on graph diffusion (GDR) for the registration of the dynamic 2D images series which is able to handle sliding-organ motion. For the registration of a 2D time series containing 140 images and a resolution of  $256 \times 256$  pixels our method needs 2 min to perform the registration. This registration method is now implemented in the current pipeline for the calculation of the ventilation and perfusion maps.

The second registration method, we developed is inspired by how a human would register an image using only local transformations and thus belongs to the class of learning-based registration methods. Here, the process of registration is separated in a learning phase and a training phase. We developed a method that models the registration as a sequence of local transformations. As we have a sequence of transformations, we based our method on gated recurrent units (GRU) called *Recurrent Registration Neural Networks* (R2N2). With this method, we are able to speed-up the registration process compared to the classical B-spline registration method. Rewriting the registration problem as sequence allows the model to apply local transformations until both images are properly aligned. This is difficult to obtain with current feed forward learning-based registration methods. Furthermore, the sequence-based approach allows an additional dimension for the regularization, i.e., the number of local transformation used to spatially align both images. The code for the R2N2 is publicly available<sup>1</sup>.

With the experience we gained through the development of new registration methods, we developed a registration framework called AIRLab. AIRLab stands for *Autograd Image Registration Laboratory* and is based on the PyTorch<sup>2</sup> framework. With AIRLab, we presented a framework that allows rapid-prototyping of new registration methods and offers an elegant way to implement existing registration methods. The key feature of the AIRLab framework is the automatic gradient calculation offered by the PyTorch framework. Furthermore, AIRLab benefits from current developments from the machine learning community and the efficient GPU support, which allows a fast development of

## 10 Discussion and Conclusion

new image registration methods. We implemented the current state-of-the-art registration methods, e.g., B-spline, Demons, diffeomorphic transformations, in the AIRLab framework. AIRLab is an open-source project<sup>3</sup>.

Besides the registration, we also worked on the image analysis part. Here, we developed a deep neural network model that reduces the number of images needed to calculate ventilation maps. We were able to reduce the number of images from 140 to 40 by keeping the quality of the maps. Since the image series is acquired as a continuous sequence, we use a stacked bidirectional gated recurrent neural network model.

For the clinical routine, the automatic analysis of the ventilation and perfusion map is essential. In detail this means an automatic detection of impaired regions in the ventilation and perfusion maps. Deep neural networks have shown great performance for fully supervised segmentation tasks. However, generating training data with a pixel-wise label (segmentation) is very time-consuming. Furthermore, using a pixel-wise label limits the network performance to human performance. In order to generate a pixel-wise segmentation, we developed a weakly supervised method that is able to infer a pixel-wise segmentation from a continuous global label. As global label, we use the lung clearance index (LCI). The LCI is a global measure for ventilation inhomogeneities of the lungs and is measured by a multiple breath washout test. Compared to other methods, we do not need any manually labeled data. The results show that, we are able to robustly detect defects in the ventilation maps.

### Future work

Our learning-based registration method R2N2 currently uses a fixed sequence length to perform the registration. This is a limitation to the system since the registration is ideally performed with a minimal number of steps. For future work, we will extend the method to automatically determine the minimal number of steps. Examples for this can also be found in the field of reinforcement learning.

In the AIRLab framework, several state-of-the-art tools for image registration are implemented. We will continue with the implementation of new tools as for example for group-wise or multi-channel image registration.

The current ventilation map generation method computes the ventilation map using only the 1D time signal of one pixel without using further spatial information of the neighboring pixels. We will extend this method by using the whole image as input at each time point for the GRU. This can be achieved by replacing the fully connected layers of the GRU with convolutions. Furthermore, we can also think of using new neural network architectures for sequential data, e.g., Transformers.

Our weakly supervised segmentation method showed very good results for the defect detection in MRI ventilation maps. However, we observed some misclassifications in areas close to blood vessels. This is caused by the fact that vessels are not ventilated

---

<sup>1</sup>R2N2 framework: <https://github.com/RobinSandkuehler/r2n2>

<sup>2</sup>PyTorch framework: <https://pytorch.org>

<sup>3</sup>AIRLab framework: <https://github.com/airlab-unibas/airlab>

and therefore they share similar characteristics as lung defects. For future work, we will investigate how these misclassifications can be removed. The first approach may be a preprocessing step using a simple vessel segmentation or adding the vessel segmentation to the learning process of the network. We will extend and evaluate our method for defect detection in MRI perfusion maps. Furthermore, we will investigate if we can include the idea of anomaly detection in our framework.

## Conclusion

In this work, we have presented methods to improve the process of calculation and analysis of MRI ventilation and perfusion maps. Our registration method is currently used in the MRI ventilation and perfusion computation pipeline. We developed a new sequence-based registration method using gated recurrent neural networks. For the ventilation map generation, we presented a method which is able to calculate ventilation maps with only a third of the original method by maintaining the quality of the ventilation maps. Since automatic map analysis is important in the clinical routine, we developed a weakly supervised defect segmentation method where no manual labeled training data is needed. The weak label we use is the lung clearance index obtained from a multiple breath washout test. With this method, we were able to detect most of the impaired regions in the ventilation maps.





# Bibliography

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283. 80
- Alvarez, L., Weickert, J., and Sánchez, J. (2000). Reliable estimation of dense optical flow fields with large displacements. *International Journal of Computer Vision*, 39(1):41–56. 13
- Andermatt, S., Horváth, A., Pezold, S., and Cattin, P. (2019). Pathology segmentation using distributional differences to images of healthy origin. In Crimi, A., Bakas, S., Kuijf, H., Keyvan, F., Reyes, M., and van Walsum, T., editors, *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 228–238, Cham. Springer International Publishing. 42, 102
- Andermatt, S., Pezold, S., and Cattin, P. (2016). Multi-dimensional gated recurrent units for the segmentation of biomedical 3d-data. In *Deep Learning and Data Labeling for Medical Applications*, pages 142–151. Springer. 93, 103, 107
- Andermatt, S., Pezold, S., and Cattin, P. (2017). Automated Segmentation of Multiple Sclerosis Lesions using Multi-Dimensional Gated Recurrent Units. In *International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Springer. 63
- Andersen, D. H. (1938). Cystic Fibrosis of the Pancreas and its Relation to Celiac Disease: A Clinical and Pathologic Study. *JAMA Pediatrics*, 56(2):344–399. 5
- Arsigny, V., Commowick, O., Pennec, X., and Ayache, N. (2006). A log-euclidean framework for statistics on diffeomorphisms. In Larsen, R., Nielsen, M., and Sparring, J., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2006*, pages 924–931, Berlin, Heidelberg. Springer Berlin Heidelberg. 16, 17, 80
- Ashburner, J. (2007). A fast diffeomorphic image registration algorithm. *NeuroImage*, 38(1):95 – 113. 16, 80
- Avants, B. B., Tustison, N. J., Song, G., Cook, P. A., Klein, A., and Gee, J. C. (2011). A reproducible evaluation of ants similarity metric performance in brain image registration. *Neuroimage*, 54(3):2033–2044. 75
- Babaud, J., Witkin, A. P., Baudin, M., and Duda, R. O. (1986). Uniqueness of the Gaussian kernel for scale-space filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(1):26–33. 47
- Bagnato, L., Frossard, P., and Vandergheynst, P. (2009). Optical flow and depth from motion for omnidirectional images using a TV-L1 variational framework on graphs. In *16th IEEE International Conference on Image Processing*, pages 1469–1472. 26, 46

## Bibliography

- Bauman, G. and Bieri, O. (2017). Matrix pencil decomposition of time-resolved proton MRI for robust and improved assessment of pulmonary ventilation and perfusion. *Magnetic Resonance in Medicine*, 77(1):336–342. 1, 2, 7, 92, 93, 97, 101, 103
- Bauman, G., Puderbach, M., Deimling, M., Jellus, V., Chef d’hotel, C., Dinkel, J., Hintze, C., Kauczor, H.-U., and Schad, L. R. (2009). Non-contrast-enhanced perfusion and ventilation assessment of the human lung by means of fourier decomposition in proton MRI. *Magnetic Resonance in Medicine*, 62(3):656–664. 7
- Bauman, G., Pusterla, O., and Bieri, O. (2016). Ultra-fast steady-state free precession pulse sequence for Fourier decomposition pulmonary MRI. *Magnetic Resonance in Medicine*, 75(4):1647–1653. 1, 6, 64, 91, 92, 96, 101, 103, 106
- Baur, C., Wiestler, B., Albarqouni, S., and Navab, N. (2018). Deep autoencoding models for unsupervised anomaly segmentation in brain MR images. In *International MICCAI Brainlesion Workshop*, pages 161–169. Springer. 42
- Beg, M. F., Miller, M. I., Trouvé, A., and Younes, L. (2005). Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International Journal of Computer Vision*, 61(2):139–157. 15, 16, 80
- Berendsen, F. F., Marstal, K., Klein, S., and Staring, M. (2016). The design of superelastix — a unifying framework for a wide range of image registration methodologies. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 498–506. 75
- Berger, M., Yang, Q., and Maier, A. (2018). *X-ray Imaging*, pages 119–145. Springer International Publishing, Cham. 5
- Biederer, J., Reuter, M., Both, M., Muhle, C., Grimm, J., Graessner, J., and Heller, M. (2002). Analysis of artefacts and detail resolution of lung MRI with breath-hold T1-weighted gradient-echo and T2-weighted fast spin-echo sequences with respiratory triggering. *European Radiology*, 12(2):378–384. 6
- Blum, A. and Rivest, R. L. (1989). Training a 3-node neural network is NP-complete. In *Advances in Neural Information Processing Systems*, pages 494–501. 27
- Bouabene, G., Gerig, T., Lüthi, M., Forster, A., Madsen, D., and Rahbani, D. (2015). Scalismo - scalable image analysis and shape modelling. [Online; accessed 25-June-2018]. 75
- Byrd, R. H., Nocedal, J., and Schnabel, R. B. (1994). Representations of quasi-newton matrices and their use in limited memory methods. *Mathematical Programming*, 63(1-3):129–156. 74
- Cachier, P. and Pennec, X. (2000). 3D non-rigid registration by gradient descent on a Gaussian-windowed similarity measure using convolutions. In *Proceedings IEEE*

- Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 182–189. 22, 52, 76
- Cahill, N. D., Noble, J. A., and Hawkes, D. J. (2009). A demons algorithm for image registration with locally adaptive regularization. In Yang, G.-Z., Hawkes, D., Rueckert, D., Noble, A., and Taylor, C., editors, *Medical Image Computing and Computer-Assisted Intervention*, pages 574–581, Berlin, Heidelberg. Springer Berlin Heidelberg. 17
- Castillo, R., Castillo, E., Guerra, R., Johnson, V. E., McPhail, T., Garg, A. K., and Guerrero, T. (2009). A framework for evaluation of deformable image registration spatial accuracy using large landmark point sets. *Physics in Medicine & Biology*, 54(7):1849. 84, 85
- Chen, X. and Konukoglu, E. (2018). Unsupervised detection of lesions in brain MRI using constrained adversarial auto-encoders. *arXiv preprint arXiv:1806.04972*. 102
- Cho, K., van Merriënboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078. 38, 60, 63, 94
- Christensen, G. E., Rabbitt, R. D., and Miller, M. I. (1996). Deformable templates using large deformation kinematics. *IEEE Transactions on Image Processing*, 5(10):1435–1447. 15, 80
- Dalca, A. V., Balakrishnan, G., Guttag, J., and Sabuncu, M. (2018a). Unsupervised learning for fast probabilistic diffeomorphic registration. *Medical Image Computing and Computer Assisted Intervention*, 11070:729–738. 19
- Dalca, A. V., Balakrishnan, G., Guttag, J., and Sabuncu, M. R. (2018b). Unsupervised learning for fast probabilistic diffeomorphic registration. In Frangi, A. F., Schnabel, J. A., Davatzikos, C., Alberola-López, C., and Fichtinger, G., editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, pages 729–738, Cham. Springer International Publishing. 19, 58
- Damadian, R. (1971). Tumor detection by nuclear magnetic resonance. *Science*, 171(3976):1151–1153. 5
- de Vos, B. D., Berendsen, F. F., Viergever, M. A., Staring, M., and Išgum, I. (2017). End-to-end unsupervised deformable image registration with a convolutional neural network. In Cardoso, M. J., Arbel, T., Carneiro, G., Syeda-Mahmood, T., Tavares, J. M. R., Moradi, M., Bradley, A., Greenspan, H., Papa, J. P., Madabhushi, A., Nascimento, J. C., Cardoso, J. S., Belagiannis, V., and Lu, Z., editors, *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 204–212, Cham. Springer International Publishing. 19, 58
- Deisenroth, M. P., Faisal, A. A., and Ong, C. S. (2019). *Mathematics For Machine Learning*. <https://mml-book.github.io/>. 28

## Bibliography

- Demirovic, D., Serifovic, A., and Cattin, P. C. (2009). An anisotropic diffusion regularized demons for improved registration of sliding organs. In *18th International Electrotechnical and Computer Science Conference (ERK)*, page BM.1.4. 26, 45, 46
- Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., and Brox, T. (2015). Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision*, pages 2758–2766. 19, 58
- Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for on-line learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159. 25, 41
- Dumoulin, V. and Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*. 79
- Dupuis, P. and Grenander, U. (1998). Variational problems on flows of diffeomorphisms for image matching. *Quarterly of Applied Mathematics*, LVI(3):587–600. 15, 80
- Elborn, J. S. (2016). Cystic fibrosis. *The Lancet*, 388(10059):2519 – 2531. 1, 5
- Elborn, J. S., Shale, D., and Britton, J. (1991). Cystic fibrosis: current survival and population estimates to the year 2000. *Thorax*, 46(12):881–885. 1, 5
- Eppenhof, K. A., Lafarge, M. W., Moeskops, P., Veta, M., and Pluim, J. P. (2018). Deformable image registration using convolutional neural networks. In *Medical Imaging 2018: Image Processing*, volume 10574, page 105740S. International Society for Optics and Photonics. 19
- Finn, C., Tan, X. Y., Duan, Y., Darrell, T., Levine, S., and Abbeel, P. (2016). Deep spatial autoencoders for visuomotor learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 512–519. 63
- Fu, Y., Lei, Y., Wang, T., Curran, W. J., Liu, T., and Yang, X. (2019). Deep learning in medical image registration: A review. *arXiv preprint arXiv:1912.12318*. 19
- Glocker, B., Komodakis, N., Tziritas, G., Navab, N., and Paragios, N. (2008). Dense image registration through mrfs and efficient linear programming. *Medical Image Analysis*, 12(6):731 – 741. Special issue on information processing in medical imaging 2007. 25, 74
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. 27, 30, 34, 37
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680. 27, 36, 41

- Haber, E. and Modersitzki, J. (2006). Intensity gradient based registration and fusion of multi-modal images. In Larsen, R., Nielsen, M., and Sporring, J., editors, *Medical Image Computing and Computer-Assisted Intervention*, pages 726–733, Berlin, Heidelberg. Springer Berlin Heidelberg. 23, 77, 78
- Haskins, G., Kruger, U., and Yan, P. (2019). Deep learning in medical image registration: A survey. *arXiv preprint arXiv:1903.02026*. 19, 58
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. 41, 63
- Heinrich, M. P., Jenkinson, M., Brady, M., and Schnabel, J. A. (2013). MRF-based deformable registration and ventilation estimation of lung CT. *IEEE Transactions on Medical Imaging*, 32(7):1239–1248. 25
- Heinrich, M. P., Jenkinson, M., Papież, B. W., Glesson, F. V., Brady, S. M., and Schnabel, J. A. (2013). Edge- and detail-preserving sparse image representations for deformable registration of chest MRI and CT volumes. In Gee, J. C., Joshi, S., Pohl, K. M., Wells, W. M., and Zöllei, L., editors, *Information Processing in Medical Imaging*, pages 463–474, Berlin, Heidelberg. Springer Berlin Heidelberg. 26
- Hermosillo, G., Chef d’Hotel, C., and Faugeras, O. (2002). Variational methods for multimodal image matching. *International Journal of Computer Vision*, 50(3):329–343. 13
- Hernandez, M., Bossa, M. N., and Olmos, S. (2007). Registration of anatomical images using geodesic paths of diffeomorphisms parameterized with stationary vector fields. In *IEEE 11th International Conference on Computer Vision*, pages 1–8. 16, 80
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9:1735–80. 37, 60, 94
- Hong, S., Noh, H., and Han, B. (2015). Decoupled deep neural network for semi-supervised semantic segmentation. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 1495–1503. Curran Associates, Inc. 42
- Horn, B. K. and Schunck, B. G. (1981). Determining optical flow. *Artificial intelligence*, 17(1-3):185–203. 11, 12, 13, 20, 23, 58, 72
- Hounsfield, G. (1975). Method of and apparatus for examining a body by radiation such as X or gamma radiation. (US3924131A). 5
- Hu, Y., Modat, M., Gibson, E., Li, W., Ghavami, N., Bonmati, E., Wang, G., Bandula, S., Moore, C. M., Emberton, M., Ourselin, S., Noble, J. A., Barratt, D. C., and Vercauteren, T. (2018). Weakly-supervised convolutional neural networks for multimodal image registration. *Medical Image Analysis*, 49:1 – 13. 19, 58

## Bibliography

- Hua, R., Pozo, J. M., Taylor, Z. A., and Frangi, A. F. (2017). Multiresolution eXtended Free-Form Deformations (XFFD) for non-rigid registration with discontinuous transforms. *Medical Image Analysis*, 36:113–122. 46
- Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T. (2017). Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 19
- Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*. 40
- Jaderberg, M., Simonyan, K., Zisserman, A., et al. (2015). Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025. 19, 58
- Jud, C., Möri, N., Bitterli, B., and Cattin, P. C. (2016a). Bilateral regularization in reproducing kernel Hilbert spaces for discontinuity preserving image registration. In Wang, L., Adeli, E., Wang, Q., Shi, Y., and Suk, H.-I., editors, *Machine Learning in Medical Imaging*, pages 10–17, Cham. Springer International Publishing. 15, 26, 45, 58, 79, 81
- Jud, C., Möri, N., and Cattin, P. C. (2016b). Sparse kernel machines for discontinuous registration and nonstationary regularization. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 449–456. 15, 26, 58, 74, 79, 81
- Jud, C., Nguyen, D., Sandkühler, R., Giger, A., Bieri, O., and Cattin, P. C. (2018a). Motion aware MR imaging via spatial core correspondence. In Frangi, A. F., Schnabel, J. A., Davatzikos, C., Alberola-López, C., and Fichtinger, G., editors, *Medical Image Computing and Computer Assisted Intervention*, pages 198–205, Cham. Springer International Publishing. 6, 8
- Jud, C., Sandkühler, R., and Cattin, P. C. (2018b). An inhomogeneous multi-resolution regularization concept for discontinuity preserving image registration. In *International Workshop on Biomedical Image Registration*, pages 3–12. Springer. 74
- Jud, C., Sandkühler, R., Möri, N., and Cattin, P. C. (2017). Directional averages for motion segmentation in discontinuity preserving image registration. In Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D. L., and Duchesne, S., editors, *Medical Image Computing and Computer Assisted Intervention*, pages 249–256, Cham. Springer International Publishing. 26
- Kauczor, H.-U. and Wielpütz, M. O., editors (2018). *MRI of the Lung*. Springer International Publishing. 6
- Keszei, A. P., Berkels, B., and Deserno, T. M. (2017). Survey of non-rigid registration tools in medicine. *Journal of digital imaging*, 30(1):102–116. 74

- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. 25, 41, 65, 67, 74, 96
- Kiriyanthan, S., Fundana, K., Majeed, T., and Cattin, P. C. (2012). A primal-dual approach for discontinuity preserving registration. In *9th IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 350–353. 26
- Kiriyanthan, S., Fundana, K., Majeed, T., and Cattin, P. C. (2016). A primal-dual approach for discontinuity preserving image registration through motion segmentation. *Int. J. Computational and Mathematical Methods in Medicine*. 26, 46
- Klein, S., Pluim, J. P. W., Staring, M., and Viergever, M. A. (2008). Adaptive stochastic gradient descent optimisation for image registration. *International Journal of Computer Vision*, 81(3):227. 25
- Klein, S., Staring, M., Murphy, K., Viergever, M. A., and Pluim, J. P. (2010). Elastix: a toolbox for intensity-based medical image registration. *IEEE Transactions on Medical Imaging*, 29(1):196–205. 74
- Klein, S., Staring, M., and Pluim, J. P. (2007). Evaluation of optimization methods for nonrigid medical image registration using mutual information and B-splines. *IEEE Transactions on Image Processing*, 16(12):2879–2890. 24
- Kondor, R. I. and Lafferty, J. D. (2002). Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 315–322, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. 48
- Krebs, J., Delingette, H., Mailhé, B., Ayache, N., and Mansi, T. (2019). Learning a probabilistic model for diffeomorphic registration. *IEEE Transactions on Medical Imaging*, 38(9):2165–2176. 19
- Krebs, J., Mansi, T., Delingette, H., Zhang, L., Ghesu, F. C., Miao, S., Maier, A. K., Ayache, N., Liao, R., and Kamen, A. (2017). Robust non-rigid registration through agent-based action learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 344–352. Springer. 58
- Krebs, J., Mansi, T., Mailhé, B., Ayache, N., and Delingette, H. (2018a). Learning structured deformations using diffeomorphic registration. *arXiv preprint arXiv:1804.07172*. 80
- Krebs, J., Mansi, T., Mailhé, B., Ayache, N., and Delingette, H. (2018b). Unsupervised probabilistic deformation modeling for robust diffeomorphic registration. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 101–109. Springer. 19

## Bibliography

- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105. 18, 27, 33, 41
- Lanczos, C. (1950). An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Natl. Bur. Stand. B*, 45:255–282. 49
- Lauterbur, P. (1973). Image formation by induced local interactions: Examples employing nuclear magnetic resonance. *Nature*, 242:190–191. 5
- LeCun, Y. (1989). Generalization and network design strategies. Technical Report CRG-TR-89-4, University of Toronto. 27, 33
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444. 27
- LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., and Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. In Touretzky, D. S., editor, *Advances in Neural Information Processing Systems 2*, pages 396–404. Morgan-Kaufmann. 33
- Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324. 33
- Ley, S. (2015). Lung imaging. *European Respiratory Review*, 24(136):240–245. 5
- Liao, R., Miao, S., de Tournemire, P., Grbic, S., Kamen, A., Mansi, T., and Comaniciu, D. (2017). An artificial agent for robust image registration. In *Thirty-First AAAI Conference on Artificial Intelligence*. 58
- Liu, R., Lehman, J., Molino, P., Such, F. P., Frank, E., Sergeev, A., and Yosinski, J. (2018). An intriguing failing of convolutional neural networks and the coordconv solution. In *Advances in Neural Information Processing Systems*, pages 9605–9616. 62
- Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440. 41
- Lorenzi, M., Ayache, N., Frisoni, G., and Pennec, X. (2013). LCC-demons: A robust and accurate symmetric diffeomorphic registration algorithm. *NeuroImage*, 81:470 – 483. 17
- Lowekamp, B. C., Chen, D. T., Ibáñez, L., and Blezek, D. (2013). The design of SimpleITK. *Frontiers in neuroinformatics*, 7:45. 72
- Lüthi, M., Gerig, T., Jud, C., and Vetter, T. (2017). Gaussian process morphable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 75



- Maintz, J. and Viergever, M. A. (1998). A survey of medical image registration. *Medical Image Analysis*, 2(1):1 – 36. 11
- Mansfield, P. and Grannell, P. (1975). "diffraction" and microscopy in solids and liquids by NMR. *Physical Review B*, 12(9):3618. 5
- Marstal, K., Berendsen, F., Staring, M., and Klein, S. (2016). Simpleelastix: A user-friendly, multi-lingual library for medical image registration. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 574–582. IEEE. 74
- Mathers, C. D. and Loncar, D. (2006). Projections of global mortality and burden of disease from 2002 to 2030. *PLOS Medicine*, 3(11):1–20. 5
- McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133. 28
- Miao, S., Piat, S., Fischer, P., Tuysuzoglu, A., Mewes, P., Mansi, T., and Liao, R. (2018). Dilated fcn for multi-agent 2d/3d medical image registration. In *Thirty-Second AAAI Conference on Artificial Intelligence*. 58
- Minsky, M. and Papert, S. A. (1969). *Perceptrons: An introduction to computational geometry*. MIT press. 29
- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*. 27
- Modersitzki, J. (2009). *FAIR: Flexible Algorithms for Image Registration*, volume 6. SIAM. 75
- Moler, C. and Loan, C. V. (2003). Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1):801–836. 16, 17, 48
- Nagel, H. H. and Enkelmann, W. (1986). An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(5):565–593. 13, 26, 45, 46, 49
- Noh, H., Hong, S., and Han, B. (2015). Learning deconvolution network for semantic segmentation. In *IEEE International Conference on Computer Vision*, pages 1520–1528. 41
- Nyilas, S., Bauman, G., Sommer, G., Stranzinger, E., Pusterla, O., Frey, U., Korten, I., Singer, F., Casaulta, C., Bieri, O., and Latzin, P. (2017). Novel magnetic resonance technique for functional imaging of cystic fibrosis lung disease. *European Respiratory Journal*, 50(6). 103
- O’Sullivan, B. P. and Freedman, S. D. (2009). Cystic fibrosis. *The Lancet*, 373(9678):1891–1904. 1, 5

## Bibliography

- Papandreou, G., Chen, L.-C., Murphy, K. P., and Yuille, A. L. (2015a). Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *The IEEE International Conference on Computer Vision*. 42
- Papandreou, G., Chen, L.-C., Murphy, K. P., and Yuille, A. L. (2015b). Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*. 102
- Papież, B. W., Franklin, J. M., Heinrich, M. P., Gleeson, F. V., Brady, M., and Schnabel, J. A. (2018). Gifted demons: deformable image registration with local structure-preserving regularization using supervoxels for liver applications. *Journal of Medical Imaging*, 5(2):024001. 17
- Papież, B. W., Heinrich, M. P., Fehrenbach, J., Risser, L., and Schnabel, J. A. (2014). An implicit sliding-motion preserving regularisation via bilateral filtering for deformable image registration. *Medical Image Analysis*, 18(8):1299–1311. 46
- Papież, B. W., Szmul, A., Grau, V., Brady, J. M., and Schnabel, J. A. (2017). *Non-local Graph-Based Regularization for Deformable Image Registration*, pages 199–207. Springer International Publishing, Cham. 26, 46, 52, 53
- Papież, B. W., Heinrich, M. P., Fehrenbach, J., Risser, L., and Schnabel, J. A. (2014). An implicit sliding-motion preserving regularisation via bilateral filtering for deformable image registration. *Medical Image Analysis*, 18(8):1299 – 1311. Special Issue on the 2013 Conference on Medical Image Computing and Computer Assisted Intervention. 17, 26
- Parker, D. (1985). Learning-logic, technical report tr-47. Technical report, Center for Computational Research in Economics and Management Science, M.I.T. 27, 29, 41
- Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch. In *NIPS-W*. 72, 75, 76
- Pathak, D., Krahenbuhl, P., and Darrell, T. (2015). Constrained convolutional neural networks for weakly supervised segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*. 102
- Pennec, X., Cachier, P., and Ayache, N. (1999). Understanding the “demon’s algorithm”: 3D non-rigid registration by gradient descent. In Taylor, C. and Colchester, A., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI’99*, pages 597–605, Berlin, Heidelberg. Springer Berlin Heidelberg. 17
- Pock, T., Urschler, M., Zach, C., Beichel, R., and Bischof, H. (2007). A duality based algorithm for TV-L1-optical-flow image registration. In Ayache, N., Ourselin, S., and Maeder, A., editors, *Medical Image Computing and Computer-Assisted Intervention*, pages 511–518, Berlin, Heidelberg. Springer Berlin Heidelberg. 24

- Qiao, Y., Sun, Z., Lelieveldt, B. P. F., and Staring, M. (2015). A stochastic quasi-newton method for non-rigid image registration. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A., editors, *Medical Image Computing and Computer-Assisted Intervention*, pages 297–304, Cham. Springer International Publishing. 25
- Reddi, S. J., Kale, S., and Kumar, S. (2019). On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*. 65, 67
- Robinson, P. D., Goldman, M. D., and Gustafsson, P. M. (2009). Inert gas washout: theoretical background and clinical utility in respiratory disease. *Respiration*, 78(3):339–355. 5, 102, 103
- Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, Cham. Springer International Publishing. 18, 27, 41
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386. 27, 28, 30
- Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*. 82
- Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268. 24
- Rueckert, D., Sonoda, L. I., Hayes, C., Hill, D. L. G., Leach, M. O., and Hawkes, D. J. (1999). Nonrigid registration using free-form deformations: application to breast MR images. *IEEE Transactions on Medical Imaging*, 18(8):712–721. 15, 58, 72, 74, 75, 78, 85
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science. 27, 29, 41
- Saad, Y. (1992). Analysis of some krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis*, 29(1):209–228. 48
- Sandkühler, R., Jud, C., Andermatt, S., and Cattin, P. C. (2018a). Airlab: Autograd image registration laboratory. *arXiv preprint arXiv:1806.09907*. 67
- Sandkühler, R., Jud, C., Pezold, S., and Cattin, P. C. (2018b). Adaptive graph diffusion regularisation for discontinuity preserving image registration. In *International Workshop on Biomedical Image Registration*, pages 24–34. Springer. 17, 82, 92, 93, 103
- Santos-Ribeiro, A., Nutt, D. J., and McGonigle, J. (2016). *Inertial Demons: A Momentum-Based Diffeomorphic Registration Framework*, pages 37–45. Springer International Publishing, Cham. 17, 51

## Bibliography

- Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U., and Langs, G. (2017). Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*, pages 146–157. Springer. 42
- Schmidt-Richberg, A., Werner, R., Handels, H., and Ehrhardt, J. (2012). Estimation of slipping organ motion by registration with direction-dependent regularization. *Medical Image Analysis*, 16(1):150 – 159. 26, 46
- Schnabel, J. A., Rueckert, D., Quist, M., Blackall, J. M., Castellano-Smith, A. D., Hartkens, T., Penney, G. P., Hall, W. A., Liu, H., Truwit, C. L., et al. (2001). A generic framework for non-rigid registration based on non-uniform multi-level free-form deformations. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 573–581. Springer. 75
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503. 27
- Smola, A. J. and Kondor, R. (2003). Kernels and Regularization on Graphs. In Schölkopf, B. and Warmuth, M. K., editors, *Learning Theory and Kernel Machines*, pages 144–158, Berlin, Heidelberg. Springer Berlin Heidelberg. 46
- Sotiras, A., Davatzikos, C., and Paragios, N. (2013). Deformable medical image registration: A survey. *IEEE Transactions on Medical Imaging*, 32(7):1153–1190. 11, 57
- Souly, N., Spampinato, C., and Shah, M. (2017). Semi supervised semantic segmentation using generative adversarial network. In *The IEEE International Conference on Computer Vision (ICCV)*. 42
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958. 65
- Stergios, C., Mihir, S., Maria, V., Guillaume, C., Marie-Pierre, R., Stavroula, M., and Nikos, P. (2018). Linear and deformable image registration with 3d convolutional neural networks. In *Image Analysis for Moving Organ, Breast, and Thoracic Images*, pages 13–22. Springer. 19, 58
- Sun, D., Roth, S., and Black, M. J. (2010). Secrets of optical flow estimation and their principles. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2432–2439. IEEE. 24

- Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9. 41
- Thirion, J.-P. (1998). Image matching as a diffusion process: An analogy with Maxwell’s demons. *Medical Image Analysis*, 2(3):243–260. 16, 46, 47, 51, 72, 82, 85
- Trouvé, A. (1995). An infinite dimensional group approach for physics based models in pattern recognition. *preprint*. 15, 80
- Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2016). Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*. 40
- van Beek, E. J., Wild, J. M., Kauczor, H.-U., Schreiber, W., Mugler III, J. P., and de Lange, E. E. (2004). Functional MRI of the lung using hyperpolarized 3-helium gas. *Journal of Magnetic Resonance Imaging*, 20(4):540–554. 6, 91
- Vandemeulebroucke, J., Sarrut, D., Clarysse, P., et al. (2007). The popi-model, a point-validated pixel-based breathing thorax model. In *XVth international conference on the use of computers in radiation therapy (ICCR)*, volume 2, pages 195–199. 84
- Vercauteren, T., Pennec, X., Perchant, A., and Ayache, N. (2007). Non-parametric diffeomorphic image registration with the demons algorithm. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 319–326. Springer. 17
- Vercauteren, T., Pennec, X., Perchant, A., and Ayache, N. (2008). Symmetric log-domain diffeomorphic registration: A demons-based approach. In *International conference on medical image computing and computer-assisted intervention*, pages 754–761. Springer. 17
- Viergever, M., Maintz, J., Klein, S., Murphy, K., Staring, M., and Pluim, J. (2016). A survey of medical image registration - under review. *Medical Image Analysis*, 33:140–144. 71
- Viola, P. and Wells, W. M. (1997). Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154. 77
- Vishnevskiy, V., Gass, T., Székely, G., and Goksel, O. (2014). Total Variation regularization of displacements in parametric image registration. In *International MICCAI Workshop on Computational and Clinical Challenges in Abdominal Imaging*, pages 211–220. Springer. 24
- Vishnevskiy, V., Gass, T., Székely, G., Tanner, C., and Goksel, O. (2016). Isotropic total variation regularization of displacements in parametric image registration. *IEEE Transactions on Medical Imaging*, 36(2):385–395. 24, 25, 45, 58, 78, 81

## Bibliography

- von Siebenthal, M., Szekely, G., Gamper, U., Boesiger, P., Lomax, A., and Cattin, P. (2007). 4d MR imaging of respiratory organ motion and its variability. *Physics in Medicine & Biology*, 52(6):1547. 8
- Vorontsov, E., Molchanov, P., Byeon, W., De Mello, S., Jampani, V., Liu, M.-Y., Kadoury, S., and Kautz, J. (2019). Boosting segmentation with weak supervision from image-to-image translation. *arXiv preprint arXiv:1904.01636*. 42
- Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., and Fergus, R. (2013). Regularization of neural networks using dropconnect. In *International Conference on Machine Learning*, pages 1058–1066. 65
- Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612. 22, 77
- Wells, W. M., Viola, P., Atsumi, H., Nakajima, S., and Kikinis, R. (1996). Multi-modal volume registration by maximization of mutual information. *Medical Image Analysis*, 1(1):35–51. 20, 77
- Werbos, P. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, Cambridge, MA. 27, 29, 41
- Wojna, Z., Ferrari, V., Guadarrama, S., Silberman, N., Chen, L.-C., Fathi, A., and Uijlings, J. (2019). The devil is in the decoder: Classification, regression and gans. *International Journal of Computer Vision*, 127(11):1694–1706. 36
- Woods, R. P., Grafton, S. T., Holmes, C. J., Cherry, S. R., and Mazziotta, J. C. (1998). Automated image registration: I. general methods and intrasubject, intramodality validation. *Journal of Computer Assisted Tomography*, 22(1):139–152. 75
- Wu, Y. and He, K. (2019). Group normalization. 40
- Yoo, T. S., Ackerman, M. J., Lorensen, W. E., Schroeder, W., Chalana, V., Aylward, S., Metaxas, D., and Whitaker, R. (2002). Engineering and algorithm design for an image processing API: a technical report on ITK-the insight toolkit. *Studies in Health Technology and Informatics*, pages 586–592. 74
- Zeiler, M. D. (2012). ADADELTA: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*. 25, 41
- Zhang, F. and Hancock, E. R. (2008). Graph spectral image smoothing using the heat kernel. *Pattern Recognition*, 41(11):3328–3342. 46, 48
- Zimmerer, D., Isensee, F., Petersen, J., Kohl, S., and Maier-Hein, K. (2019). Unsupervised anomaly localization using variational auto-encoders. In Shen, D., Liu, T., Peters, T. M., Staib, L. H., Essert, C., Zhou, S., Yap, P.-T., and Khan, A., editors, *Medical Image Computing and Computer Assisted Intervention*, pages 289–297, Cham. Springer International Publishing. 42

- Zimmerer, D., Kohl, S. A., Petersen, J., Isensee, F., and Maier-Hein, K. H. (2018). Context-encoding variational autoencoder for unsupervised anomaly detection. *arXiv preprint arXiv:1812.05941*. 42





# Publications

- **Robin Sandkühler**, Simon Andermatt, Grzegorz Bauman, Sylvia Nyilas, Christoph Jud and Philippe C. Cattin: *Recurrent Registration Neural Networks for Deformable Image Registration*, Advances in Neural Information Processing Systems, 2019
- **Robin Sandkühler**, Christoph Jud, Grzegorz Bauman, Corin Willers, Orso Pusterla, Sylvia Nyilas, Alan Peters, Lukas Ebner, Enno Stranziger, Oliver Bieri, Philipp Latzin and Philippe C. Cattin: *Weakly Supervised Learning Strategy for Lung Defect Segmentation*. International Workshop on Machine Learning in Medical Imaging at the International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2019
- **Robin Sandkühler**, Grzegorz Bauman, Sylvia Nyilas, Orso Pusterla, Corin Willers, Oliver Bieri, Philipp Latzin, Christoph Jud, and Philippe C. Cattin: *Gated Recurrent Neural Networks for Accelerated Ventilation MRI*. International Workshop on Machine Learning in Medical Imaging at the International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2019
- Christoph Jud, Damien Nguyen, Alina Giger, **Robin Sandkühler**, Miriam Krieger, Tony Lomax, Rares Salomir, Oliver Bieri, Philippe C. Cattin: *Accelerated Motion-Aware MR Imaging via Motion Prediction from K-Space Center*, arXiv preprint arXiv:1908.09560, 2019
- Christoph Corin Willers, Orso Pusterla, Grzegorz Bauman, Simon Andermatt, Sylvia Nyilas, Francesco Santini, Simon Pezold, **Robin Sandkühler**, Kathryn Ramsey, Philippe C. Cattin, Oliver Bieri, Philipp Latzin: *Inter-reader variation in lung segmentation of functional lung MRI quantification*, European Respiratory Journal, 2019
- **Robin Sandkühler**, Christoph Jud, Simon Andermatt, and Philippe C. Cattin: *AirLab: Autograd Image Registration Laboratory*. arXiv preprint arXiv:1806.09907, 2018
- Christoph Jud, Damien Nguyen, **Robin Sandkühler**, Alina Giger, Oliver Bieri, and Philippe C. Cattin: *Motion Aware MR Imaging via Spatial Core Correspondence*. International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2018
- Alina Giger, **Robin Sandkühler**, Christoph Jud, Grzegorz Bauman, Oliver Bieri, Rares Salomir, and Philippe C. Cattin: *Respiratory Motion Modelling using cGANs*. International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2018

## Publications

- **Robin Sandkühler**, Christoph Jud, Simon Pezold, and Philippe C. Cattin: *Adaptive Graph Diffusion Regularisation for Discontinuity Preserving Image Registration*. International Workshop on Biomedical Image Registration, Springer, 2018
- Christoph Jud, **Robin Sandkühler**, and Philippe C. Cattin: *An Inhomogeneous Multi-resolution Regularization Concept for Discontinuity Preserving Image Registration*. International Workshop on Biomedical Image Registration, Springer 2018
- Christoph Jud, **Robin Sandkühler**, Nadia Möri, and Philippe C Cattin: *Directional Averages for motion segmentation in discontinuity preserving image registration*. International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2017
- Christoph Jud, Alina Giger, **Robin Sandkühler**, and Philippe C Cattin: *A Localized Statistical Motion Model as a Reproducing Kernel for Non-rigid Image Registration*. International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2017
- **Robin Sandkühler**, Christoph Jud, and Philippe C. Cattin: *On a Spectral Image Metric for non-rigid group-wise Registration of Dynamic MR Image Series*. Pulmonary Image Analysis Workshop at the International Conference on Medical Image Computing and Computer-Assisted Intervention, 2016