

TESIS DOCTORAL

PARALLELIZATION AND DEEP LEARNING
TECHNIQUES FOR THE REGISTRATION
AND RECONSTRUCTION OF DYNAMIC
CARDIAC MAGNETIC RESONANCE IMAGING

ELENA MARTÍN GONZÁLEZ

Directores de Tesis

Dr. Carlos Alberola López

Dr. Pablo Casaseca de la Higuera

Doctorado en Tecnologías de la Información y las Telecomunicaciones
Escuela Técnica Superior de Ingenieros de Telecomunicación
Universidad de Valladolid



Universidad de Valladolid

UNIVERSIDAD DE VALLADOLID
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN
DEPARTAMENTO DE TEORÍA DE LA SEÑAL Y COMUNICACIÓN
E INGENIERÍA TELEMÁTICA
PROGRAMA DE DOCTORADO EN TECNOLOGÍAS DE LA INFORMACIÓN
Y LAS TELECOMUNICACIONES

TESIS DOCTORAL

PARALLELIZATION AND DEEP
LEARNING TECHNIQUES FOR THE
REGISTRATION AND
RECONSTRUCTION OF DYNAMIC
CARDIAC MAGNETIC RESONANCE
IMAGING

Presentada por
Elena Martín González
para optar al grado de Doctora por la Universidad de Valladolid

Dirigida por
Carlos Alberola López
Pablo Casaseca de la Higuera

MAYO 2023

Título: Parallelization and Deep Learning Techniques
Title for the Registration and Reconstruction of
Dynamic Cardiac Magnetic Resonance Imaging

Autor: Elena Martín González
Author

Directores: Carlos Alberola López y
Advisors Pablo Casaseca de la Higuera

Departamento: Teoría de la Señal y Comunicaciones e Ingeniería
Department Telemática

TRIBUNAL / *Committee*

PRESIDENTE:
President

VOCAL:
Vocal

SECRETARIO:
Secretary

acuerda otorgarle la calificación de

En Valladolid, a de de 2023.

INFORMACIÓN DE CONTACTO:

Contact information:

📍 **Elena Martín González**

Laboratorio de Procesado de Imagen (LPI)

Dept. Teoría de la Señal y Comunicaciones e Ingeniería Telemática

ETS. Ingenieros de Telecomunicación

Universidad de Valladolid

Campus Miguel Delibes sn. 47011

Valladolid, Spain

@ emargon@lpi.tel.uva.es

🌐 <https://www.lpi.tel.uva.es/emargon>

🆔 <https://orcid.org/0000-0002-5922-4960>

A mi padre y a mi madre.

*“Confía en el tiempo, que suele dar dulces salidas a
muchas amargas dificultades.”*

— Miguel de Cervantes

Acknowledgements

First and foremost, I would like to express my heartfelt gratitude to my supervisors, Carlos and Pablo, for their support, guidance and encouragement throughout my doctoral journey. Their invaluable insights and constructive feedback have been instrumental in shaping my research and helping me to stay focused and motivated. I am deeply grateful for their mentorship and their unconditional dedication.

I am also sincerely grateful to the team of researchers at the LPI for their assistance and support throughout my time here. I want to thank Santiago, Daniel, and Óscar for supporting me at the beginning of my research, and to Elisa and Rosa for helping me see it through to the end. Thank you all for your support and advice.

I am also thankful to my family and loved ones for their unwavering support and understanding during the long and often challenging journey towards completing my Doctoral Thesis. My parents, Juan and Marisa, have been my biggest supporters and I am deeply grateful for their love and encouragement. I wish I could share this success with both of them.

Overall, I am extremely grateful to everyone who has contributed to my success. Thank you all for your support and guidance, and for helping me to reach this important milestone.

Resumen

La imagen por resonancia magnética (MRI) es una técnica de imagen médica que produce imágenes detalladas de los órganos y tejidos corporales sin utilizar radiaciones ionizantes. En el campo de la cardiología, la resonancia magnética cardíaca (CMR), también conocida como MRI cardíaca, es una valiosa herramienta para la evaluación de pacientes con distintas patologías. Las modalidades de CMR como cine cardíaco y perfusión de primer paso permiten visualizar el comportamiento dinámico del miocardio y evaluar la perfusión sanguínea en el tejido cardíaco respectivamente, pero requieren contención de la respiración durante la adquisición y presentan dificultades técnicas debido a la dinámica del corazón. Además, la CMR presenta otros inconvenientes, como la duración del protocolo, los elevados costes y la probabilidad de degradación de la imagen debido a movimientos cardíaco y respiratorio irregulares. Existe una gran demanda de mejora de la eficiencia de la adquisición de datos para reducir los tiempos de las pruebas o mejorar las resoluciones espacio-temporales.

Para aumentar la comodidad del paciente, es necesario reducir la cantidad de información necesaria para la reconstrucción de las imágenes, es decir, recoger sólo una parte del espacio k (submuestreo), lo que se traduce en una mayor complejidad en el procesamiento de las adquisiciones. El problema surge porque la información del problema de reconstrucción es insuficiente para determinar una solución, es decir, el problema está mal condicionado, por lo que hay que añadir información y restricciones adicionales. Esto da lugar a una función de coste que se optimiza para proporcionar una solución regularizada.

Esta Tesis se centra en el reto de reconstruir eficientemente imágenes de resonancia magnética dinámicas a partir de datos altamente submuestrea-

dos, aprovechando el movimiento presente en las secuencias dinámicas para utilizar la información redundante tanto en tiempo como en espacio. Para ello, partimos de la solución *groupwise compressed sensing* (GWCS), previamente desarrollada en el grupo de investigación.

Por un lado, se explora un marco paralelo para GWCS. GWCS es un problema altamente paralelizable, por lo que haremos uso del framework OpenCLIPER, también desarrollado previamente en el grupo, para su implementación. Esta solución está compensada en movimiento para aumentar el carácter *sparse* de la solución, por lo que tenemos que paralelizar tanto el algoritmo de registro *groupwise* para estimar el movimiento, que se hará usando FFDs con B-splines cúbicos, como el algoritmo de optimización de la propia reconstrucción, para lo que usaremos NESTA. Se analizan los resultados obtenidos con y sin estimación y compensación de movimiento, lo que nos permite afirmar que la solución es clínicamente viable en términos de tiempos de ejecución, y apta para cualquier dispositivo informático que disponga de una implementación OpenCL.

Por otro lado, se propone una solución tipo GWCS basada en deep learning. Proponemos sustituir los pasos de optimización por aprendizaje profundo para que las reconstrucciones sean más rápidas y se reduzca la complejidad computacional. Primero creamos una solución rápida para el registro con DL no supervisado, llamada dGW, y luego una solución DL autosupervisada para la reconstrucción compensada en movimiento que hace uso del registro previamente entrenado, llamada SSMoComp. En cuanto a dGW, los resultados fueron comparables a los de una solución basada en la optimización, mientras que los tiempos de ejecución del registro se redujeron claramente. En cuanto a SSMoComp, se comparó con una solución del estado del arte y los resultados favorecieron nuestro diseño. Se adaptó además una versión modificada de la solución cine DL para la perfusión de primer paso, denominada SECRET. En comparación con los enfoques más avanzados, el método SECRET mantiene reconstrucciones de buena calidad para tasas de aceleración más altas, con tiempos de

entrenamiento bajos y tiempos de reconstrucción reducidos.

Abstract

Magnetic Resonance Imaging (MRI) is a medical imaging technique that produces detailed images of the body organs and tissues without the use of ionizing radiation. In the field of cardiology, cardiac magnetic resonance (CMR), also known as cardiac MRI, is a valuable tool for the evaluation of patients with different pathologies. CMR modalities such as cardiac cine and first-pass perfusion allow practitioners to visualize the myocardium dynamic behavior and to evaluate blood perfusion in the heart tissue respectively, but they require breath-holds during acquisition and have technical difficulties due to the dynamics of the heart. In addition, there are other drawbacks to CMR, including the protocol duration, high costs, and the likelihood of image degradation due to irregular cardiac and respiratory motion. There is a high demand to improve the efficiency of data collection to reduce scan times or to improve spatio-temporal resolutions.

To increase patient comfort, the amount of information required for image reconstruction must be reduced, i.e. collecting only a portion of k-space (undersampling), which results in increased complexity in processing the acquisitions. The issue arises because the information in the reconstruction problem is insufficient to determine a solution, i.e. the problem is ill-conditioned, so additional information and constraints must be added. This results in a cost function that is optimized to provide a regularized solution.

The focus of this Thesis is on the challenge of efficiently reconstructing dynamic MRI images from highly undersampled data, taking advantage of the motion present in the dynamic sequences to utilize the redundant information in both time and space. For this purpose, we start from the groupwise compressed sensing (GWCS) solution, previously reported by

our research group.

On the one hand, a parallel framework for GWCS is explored. GWCS is a highly parallelizable problem, so we will make use of the OpenCLIPER framework, another previous work of ours, for its implementation. This solution is motion compensated to increase the sparse character of the solution, so we have to parallelize: 1) the groupwise registration algorithm to estimate the motion, which will be done using FFDs with cubic B-splines, and 2) the optimization algorithm of the reconstruction itself, for which we will use NESTA. Results obtained with and without motion estimation and compensation are analyzed to conclude that the solution is clinically viable in terms of execution times, and suitable for any computing device which has an OpenCL implementation.

On the other hand, we propose a GWCS-like approach that leverages deep learning to enhance the reconstruction process. Our approach eliminates the need of optimization steps and utilizes deep learning techniques instead to speed up reconstructions and reduce computational complexity. We first create a fast solution for registration with unsupervised DL, called dGW, and then a self-supervised DL solution for motion-compensated reconstruction (SSMoComp) that relies on the previously trained registration. Regarding dGW, we found that it achieved comparable accuracy to traditional optimization-based approaches, but with significantly reduced registration runtimes. As for SSMoComp, we conducted a comparative analysis with a state-of-the-art solution and observed that our design outperformed it, yielding superior results. A modified version of the cine DL solution was additionally adapted for first-pass perfusion, called SECRET. Compared with state-of-the-art approaches, the SECRET method maintains good quality reconstructions for higher acceleration rates, with low training and very fast reconstruction times.

Contents

Acknowledgements	xiii
Resumen	xv
Abstract	xix
Contents	xxi
List of Acronyms	xxv
List of Figures	xxxvi
List of Tables	xxxvii
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	6
1.3 Methodology	7
1.4 Materials	9
1.5 Publications	10
1.6 Document Overview	12
2 Background	15
2.1 Magnetic Resonance Principles	15
2.1.1 The Physics of MRI	15
2.1.2 Image Formation	21
2.1.2.1 K-space	23
2.1.2.2 Multislice vs. 3D Imaging	23

2.1.3	Image Artifacts	24
2.2	Cardiac Magnetic Resonance Imaging: CMR	25
2.2.1	Cine Imaging	26
2.2.2	Perfusion Imaging	27
2.3	Image Reconstruction	28
2.3.1	Parallel Imaging	30
2.3.1.1	SENSE	31
2.3.2	Compressed Sensing	32
2.3.3	Groupwise Compressed Sensing	35
2.4	Image Registration	37
2.4.1	Deformation	39
2.5	Registration and Reconstruction Acceleration: Paralleliza- tion and DL Techniques	44
2.5.1	Parallelization using GPU Programming	44
2.5.1.1	Related Work	45
2.5.2	Deep Learning	47
2.5.2.1	Related Work	50
2.6	Conclusion	54
3	DL Groupwise Registration: dGW	55
3.1	Introduction	55
3.2	Architecture	59
3.3	Template Update	60
3.4	Loss Function	61
3.5	Training	62
3.6	Implementation	64
3.7	Performance Analysis and Hyperparameter Selection	64
3.8	Experiments	67
3.8.1	Experiment 1: Iterations, Time Sequence Ordering and an Alternative Similarity Metric	69

3.8.2	Experiment 2: Performance Comparison with Another DL Architecture	71
3.8.3	Experiment 3: dGW vs. an Optimization-Based Registration Approach	76
3.8.4	Experiment 4: Dynamic Image Reconstruction	77
3.9	Discussion	78
4	Cine MR reconstruction with ME/MC using OpenCLIPER	83
4.1	Introduction	83
4.2	Parallel Implementation of GW	86
4.3	Parallel Implementation of NESTA	87
4.4	Algorithm Implementation on a Generic Parallel Device	88
4.4.1	Generalities	90
4.4.2	Data Structures	91
4.5	Evaluation	92
4.6	Discussion	101
5	DL Cine Reconstruction with MC: SSMoComp	103
5.1	Introduction	103
5.2	Evaluation	105
5.3	Comparison between SSMoComp and OpenCLIPER	107
5.3.1	Prior to Comparison	107
5.3.2	Experiment 1	110
5.3.3	Experiment 2	110
5.4	Discussion	113
6	DL Perfusion Reconstruction: SECRET	117
6.1	Introduction	117
6.2	Dataset	119
6.3	Implementation details	120
6.4	Evaluation	121
6.5	Results and Discussion	122

7 Conclusions and future work	127
7.1 Conclusion	127
7.2 Contributions	129
7.3 Future work	131
Appendices	135
Appendix A Forward and Backward Operators	135
A.1 Control Point Mesh	135
A.2 GW Forward Transformation	137
A.3 GW Backward Transformation	138
A.4 Image Formation: Intensity Interpolation	139
Appendix B Graphical user interfaces	141
B.1 Training GUI	141
B.2 Registration GUI	144
Bibliography	144

List of Acronyms

This section lists (in alphabetic order) all the acronyms that have been used in this Thesis dissertation.

1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional
ADMM	Alternating Direction Method of Multipliers
AF	Acceleration Factor
AI	Artificial Intelligence
B₀	Main (static) magnetic field
B₁	Radio-frequency magnetic field
BART	Berkeley Advanced Reconstruction Toolbox
BLAST	Broad-use Linear Acquisition Speed-up Technique
BN	Batch Normalization
bSSFP	Balanced Steady-State Free Precession
CAD	Coronary Artery Disease
CC	Cross-Correlation
CMR	Cardiovascular Magnetic Resonance
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CS	Compressed Sensing
CT	Computed Tomography

CUDA	Compute Unified Device Architecture
dGW	Deep Groupwise
DICOM	Digital Imaging and Communication On Medicine
DL	Deep Learning
DMA	Direct Memory Access
DSP	Digital Signal Processor
ECG	Electrocardiogram
FFD	Free Form Deformation
FFT	Fast Fourier Transform
FID	Free Induction Decay
FOCUSS	FOCal Underdetermined System Solver
FOV	Field of View
FPGA	Field-Programmable Gate Array
FPP	First Pass Perfusion
FPP-CMR	First Pass Perfusion Cardiac Magnetic Resonance
GPGPU	General-Purpose Computing on Graphics Processing Units
GPU	Graphics Processing Unit
GRAPPA	GeneRalized Autocalibrating Partially Parallel Acquisition
GW	Groupwise
GWCS	Groupwise Motion Compensated Compressed Sensing
HCM	Hypertrophic Cardiomyopathy
iFFT	Inverse Fast Fourier Transform

kNN	K-Nearest Neighbor
LDDMM	Large Deformation Diffeomorphic Metric Mapping
LV	Left Ventricle
MC	Motion Compensation
ME	Motion Estimation
MI	Mutual Information
ML	Machine Learning
MoDL	Model-Based Deep Learning
MR	Magnetic Resonance
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
MSI	Multi-Spectral Imaging
NCC	Normalized Cross-Correlation
NegCC	Negative of Cross-Correlation
NESTA	Nesterov's Algorithm
NMR	Nuclear Magnetic Resonance
NRMSE	Normalized Root Mean Squared Error
OpenCLIPER	OpenCL-based framework for Image ProcEssing and Reconstruction
PCA	Principal Component Analysis
PET	Positron Emission Tomography
PI	Parallel Imaging
PPG	Photoplethysmography
PSNR	Peak Signal-to-Noise Ratio
PW	Pairwise
QI	Quality Index

QILV	Quality Index based on Local Variance
ReLU	Rectified Linear Unit
RF	Radio-Frequency
ROI	Region Of Interest
RV	Right Ventricle
SECRET	Self-Supervised aCcelerated REconsTruction
SENSE	Sensitivity Encoding
SER	Signal to Error Ratio
SNR	Signal to Noise Ratio
SPECT	Single Photon Emission Computerized Tomography
SSD	Sum of Square Differences
SSIM	Structural Similarity Index Metric
SSMoComp	Self-Supervised Motion Compensated
TE	Echo Time
TPS	Thin-Plate Spline
TR	Repetition Time
tTV	Temporal Total Variation
VM	VoxelMorph

List of Figures

1.1	Relation between each of the contributions related to this Thesis grouped by the major objectives stated in Section 1.2.	13
2.1	Precession. a) Magnetic moments of nuclear spins pointing in random directions due to thermal motion. b) Once B_0 is applied, magnetic moments tends to align with it. c) Displacement of M out of its equilibrium alignment by the application of B_1 magnetic field. d) Precession of M in the transverse plane at the Larmor frequency after the application of B_1 .	16
2.2	Magnetization patterns. Evolution of components. a) RF pulse application for arbitrary alpha, b) relaxation for RF pulse of arbitrary alpha, c) evolution of the magnetisation components in relaxation after a 90° RF pulse and after an RF pulse of less than 90° .	18
2.3	Relaxation of both (a) longitudinal (M_z recovery) and (b) transverse (M_{xy} decay) components for different constant decays, together with FID.	19
2.4	Simplified pulse sequence representing slice selection, frequency encoding and phase encoding stages. An example of k-space can also be observed.	22
2.5	Frames extracted from cine series. Four frames of a short axis acquisition selected to show the contractile function of the heart, which is evident in the third one.	27

- 2.6 Frames extracted from FPP-CMR series. Four frames of a short axis acquisition selected to show the passage of the contrast agent through the heart. It can be seen how in the first image the contrast has not yet arrived and in the following images how it passes from the RV to the LV. . . . 28
- 3.1 Outline of the proposed dGW registration network. A 2D cardiac cine sequence consisting of N frames enters the network, one at a time together with the template, which is calculated as described in Section 3.3. The output of the N executions of the CNN is a sequence of N 2D deformation fields. The original frames, together with their two-channel deformation fields, enter the Spatial transformation block, giving rise to the registered images. This process is applied $L + 1$ times (see Section 3.5), both for training and for prediction. 58
- 3.2 Proposed convolutional architecture of the dGW network. The number of channels is shown above each layer. Skip connections involve concatenation of feature maps extracted in the encoding stage with new feature maps from the decoding stage. 59
- 3.3 dGW registration pipeline. The original 2D cardiac cine MRI sequence \mathbf{m} enters both the network and the template selection block (see Section 3.3, $l = 0$). The output is the registered sequence $\mathbf{m}(\mathcal{T})_0$, which is used to calculate the template for $l = 1$. This is cycled until $l \leq L = 5$ 63
- 3.4 Loss evolution during training. 65

- 3.5 Differences between the SSIM samples for the validation sets in Database I (left) and Database II (right). (Upper row) Pairwise differences in Stage 1 with each parameter taking a non-null value and the SSIM sample obtained from the network in Stage 0. (Lower row) Pairwise differences in Stage 2 for the best selection of each parameter within the pair indicated above. 68
- 3.6 SSD obtained in dGW registration according to the number of iterations. 71
- 3.7 Differences between $l = 0$ and $l = L = 5$ in dGW registration quality. We show systole and a diastole superimposed frames for two slices in Database II. Leftmost column: dGW registration with $l = 0$. Rightmost column: dGW registration with $l = L = 5$. Pink: registered systole with higher intensity. Green: registered diastole with higher intensity. . 72
- 3.8 (Left) Distribution of SSIM samples in dGW registration, using NegCC as similarity metric, SSD with null hyperparameters and SSD with $\lambda_3 = 10^{-7}$. (Right) Differences between the SSIM samples on the left. 73
- 3.9 Differences between the implementation with NegCC as a similarity metric and two models with SSD in dGW registration quality. We show systole and a diastole superimposed frames for a slice of a patient in Database II. From left to right: dGW registration with SSD and $\lambda_3 = 10^{-7}$, dGW registration with SSD and all hyperparameters null, dGW registration with NegCC and $\lambda_3 = 10^{-7}$. Pink: registered systole with higher intensity. Green: registered diastole with higher intensity. 73

- 3.10 Performance comparison between dGW and other alternatives. Four different similarity metrics are used. Upper line: SSIM (left) and MI (right); Lower line: SER (left) and NCC (right). 74
- 3.11 Differences between dGW and the eight VM implementation in registration quality. We show systole and a diastole superimposed frames for a slice of a patient in Database II. Leftmost column: unregistered cine cardiac MRI (up), dGW registration (down). Middle column: the four VM-2 implementations. Rightmost column: the four VMdiff implementations. Pink: registered systole with higher intensity. Green: registered diastole with higher intensity. 75
- 3.12 Differences between GW and dGW registration. From left to right: unregistered cine cardiac MRI, GW registration, dGW registration. Pink: registered systole with higher intensity. Green: registered diastole with higher intensity. . . 77
- 3.13 Average SSIM obtained, for different AF values, between the ME/MC reconstructions carried out with dGW and GW, with respect to the fully sampled reconstruction. 78
- 3.14 Systole and diastole reconstructions with AF=8. Upper row: systole; lower row, diastole. Columns: from left to right, fully sampled reconstruction, GW and dGW. 79
- 4.1 Functional units of the proposed reconstruction system. Gray boxes represent necessary albeit unproductive work, which is taken care of by OpenCLIPER. White boxes represent actual, productive work. Each white box is abstracted as a *process*, which may be composed of several other processes. U , U^H , E , E^H are the internal NESTA operators. tTV: temporal Total Variation; GW: Groupwise Registration. 89

-
- 4.2 Boxplots of SSIM (a), NCC (b), and SER (c) for experiments AF4, AF4CC, and AF8 on both OpenCLIPER, which makes use of NESTA, and BART (which uses ADMM). No significant differences have been found. 94
- 4.3 Example of reconstructions using BART and OpenCLIPER for the experiments AF4, AF8, and AF8 with ME/MC; the latest only for OpenCLIPER. Fully sampled images are shown as reference. End-systole and end-diastole frames are shown in each case, as well as the intensities along time of the vertical profile marked with red line in reference images. Last row shows the error images (mean squared error, MSE, between reference and reconstructed image). 96
- 4.4 Example of reconstructions using BART and OpenCLIPER for the experiments AF4, AF8, and AF8 with ME/MC; the latest only for OpenCLIPER. Fully sampled images are shown as reference. End-systole and end-diastole frames are shown in each case, as well as the intensities along time of the vertical profile marked with red line in reference images. Last row shows the error images (mean squared error, MSE, between reference and reconstructed image). 97
- 4.5 Boxplots for experiments AF4, AF4CC, and AF8 on (a) GeForce 2080Ti and (b) RTX6000. Times reported are per whole slice stack. 98
- 4.6 Boxplots of computing time for AMD Radeon RX 5700XT (GFX1010) and NVIDIA Quadro RTX 6000 (RTX6000) for the three experiments. Notice that red-shaded boxes coincide with the red-shaded boxes in Figure 4.5. Times reported are per whole slice stack. 99
- 4.7 Boxplots of SSIM for AF8 on OpenCLIPER with (right, green) and without (left, red) ME/MC. 99

4.8	Boxplots of NCC (a) and SER (b) for AF8 on OpenCLIPER with (right, green) and without (left, red) ME/MC.	100
5.1	Flow chart illustrating the proposed SSMoComp method. Blue lines represent steps that only take place during training. The inputs of the framework are the undersampled k-space data \mathbf{y} and the sampling masks A , resulting in the reconstructed cine images $\hat{\mathbf{m}}$ as output, and $\hat{\mathbf{y}}$ if required.	104
5.2	Example of reconstructions using OpenCLIPER without ME/MC and with two iterations of ME/MC for the experiments carried out to define the tolerance of the stopping criterion. Reconstructions for two subjects in Database I and Database II from undersampled k-space data with AF=10, with tolerance of the stopping criterion equals to 0.01, 0.005 and 0.001, are displayed along with ground truth images and zero-filled reconstructions that serve as a reference.	109
5.3	Boxplots of PSNR (a), SSIM (b), NRMSE (c), QILV (d) and execution time (e) for AF=10 in Database I on OpenCLIPER with (green) and without (red) ME/MC and SSMoComp (yellow). Results for zero-filled reconstructions are also shown for comparison purposes.	111
5.4	Boxplots of PSNR (a), SSIM (b), NRMSE (c), QILV (d) and execution time (e) for AF=10 in Database II on OpenCLIPER with (green) and without (red) ME/MC and SSMoComp (yellow). Results for zero-filled reconstructions are also shown for comparison purposes.	112
5.5	Boxplots of PSNR (a), SSIM (b), NRMSE (c) and QILV (d) for AF=10 in Database I random folds on OpenCLIPER with (green) and without (red) ME/MC and SSMoComp before (yellow) and after (maroon). Results for zero-filled (blue) reconstructions are also shown for comparison purposes.	114

- 5.6 Boxplots of PSNR (a), SSIM (b), NRMSE (c) and QILV (d) for AF=10 in Database II random folds on OpenCLIPER with (green) and without (red) ME/MC and SSMoComp before (yellow) and after (maroon). Results for zero-filled (blue) reconstructions are also shown for comparison purposes. 115
- 5.7 Boxplots of PSNR (a), SSIM (b), NRMSE (c) and QILV (d) for AF=10 in Database I and Database II random folds on OpenCLIPER with (green) and without (red) ME/MC and SSMoComp before (yellow) and after (maroon). Results for zero-filled (blue) reconstructions are also shown for comparison purposes. 116
- 6.1 Flow chart illustrating the proposed SECRET method for FPP-CMR. Blue lines represent steps that only take place during training. The inputs of the framework are the under-sampled (k,t)-space data \mathbf{y} and the (k,t)-sampling masks A , resulting in the reconstructed contrast-enhanced dynamic images $\hat{\mathbf{m}}$ as output, and $\hat{\mathbf{y}}$ if required. 119
- 6.2 SECRET and MoDL (K=1) reconstructions obtained from 6 \times and 10 \times undersampled FPP-CMR data for two representative subjects. The reference images are displayed for comparison, in addition to CS reconstruction for 10 \times . The RV, LV and myocardial enhancement time frames are shown for one short axis slice. 123
- 6.3 PSNR, SSIM and NRMSE between the reference images and the reconstructions obtained with SECRET and MoDL methods, for 3 \times , 6 \times and 10 \times acceleration factors, for all patients in the test dataset. 124
- 6.4 Representative image profile across the heart demonstrating that the SECRET framework improves consistency across time frames. 125

6.5	Quantitative maps (K^{Trans}) obtained from $6\times$ and $10\times$ undersampled data using MoDL and the SECRET methods. The reference image is displayed for comparison.	126
A.1	Control point mesh design for non-rigid transformation of an object.	137
A.2	Interpolation after (a) forward and (b) backward deformation. (c) Detail to better understand the interpolation after backward deformation. The contribution of the green point I to the adjacent points is shown. It should be noted that these points will also be affected by contributions from other points, e.g. point that has been numbered “4” is also affected by the orange and yellow points.. . . .	140
B.1	Training GUI - dGW tab.	142
B.2	Training GUI - VM tab.	143
B.3	Training GUI - VMdiff tab.	143
B.4	Registration GUI. (a) Main window. (b) Display window: it shows two videos, the dynamic series to be registered (left) and the series once registered (right).	145

List of Tables

4.1	Mean SSIM values evaluated on undersampled BART reconstructions for a single slice of all patients and the three experiments (AF4, AF4CC, and AF8). Maximal values (bold highlighted) indicate the optimal regularization weight (λ in Equation (2.21)) for the experiments.	93
4.2	Performance comparison between cuFFT and clFFT libraries on various devices. A single experiment consists in transforming a k-space dataset (160×160 images, 19 coils, 20 frames, complex floats) 1000 times. Each experiment is run 100 times on each library and device. For each combination we show mean execution times (mean) and their standard deviation (std), both in seconds.	101
5.1	Sweep λ and number of iterations of dGW.	106
5.2	Image quality metrics for reconstructions using OpenCLIPER without ME/MC and with two iterations of ME/MC for the experiments carried out to define the tolerance of the stopping criterion. PSNR, SSIM, NRMSE and the quality index based on local variance (QILV) values obtained in reconstructions for test subset in Database I and entire Database II from undersampled k-space data with AF=10, with tolerance of the stopping criterion equals to 0.01, 0.005 and 0.001. QIs from zero-filled reconstructions are exposed as reference.	108

Chapter 1

INTRODUCTION

This Chapter introduces and motivates this Thesis dissertation. The objectives pursued, the methodology used to achieve these objectives, as well as the materials and resources available during the development of this Thesis, are described. The publications and communications that have resulted from this work are also enumerated.

1.1 Motivation

MRI is a medical imaging method that produces detailed images of the body organs and tissues. It works by using energy released from hydrogen atoms in the body, which are present in water and fat. When protons are subjected to a strong and uniform magnetic field and then energized with radio-frequency (RF) waves, they give off echoes that are detected by antennas –also known as coils– located near the part of the body being examined. These echoes, which are received as RF waves, are then used to create images of the tissues.

The basis of this phenomenon was firstly described by Isidor Rabi in 1938. A few years later, in 1946, Felix Bloch and Edward Purcell, from two independent research groups, demonstrated the nuclear magnetic resonance (NMR) phenomenon, and developed equations explaining the origin and

properties of the NMR signal [1, 2]. In 1973, Paul Lauterbur realized that the information from NMR signals could be recovered in the form of images [3]. Later, Peter Mansfield improved the technique to reduce the time taken to perform a scan, which led to the use of this non-invasive technique for practical applications and resulted in the use of NMR, also called MRI, as a diagnostic tool [4, 5].

MRI is a safe imaging modality that can be performed repeatedly or in dynamic studies without concerns about radiation exposure because there is no ionizing radiation involved [6]. This is a substantial advantage for MRI compared to other imaging modalities, such as computed tomography (CT), single photon emission computed tomography (SPECT) and positron emission tomography (PET). CT uses multiple X-ray beams from various angles and positions to create a detailed image, which implies a high exposure to ionizing radiation. For SPECT or PET scans, on the other hand, the patient must be injected with a liquid containing small radioactive particles, i.e., the patient is exposed to ionizing radiation.

There are some other advantages of MRI, as well as disadvantages or limitations. On the one hand, MRI is a very flexible and powerful technique that can provide not only anatomical, but also physical, chemical, metabolic, and functional information about the imaged tissue or organ. The main imaging advantage of MRI is its soft tissue contrast, making it ideal for imaging many organs and tissues. Its application for hard tissues (i.e., bone and teeth) poses some issues due to the very low level of water contained in these tissues [7]. This technique can image slices in any direction, as well as imaging complete three-dimensional (3D) volumes. The arbitrary slice orientation capability in MRI allows for easy scan planning and image plane alignment. Furthermore, MRI has numerous contrast mechanisms that can be used to differentiate between different tissues without the administration of contrast agents, although they may be necessary for certain studies.

On the other hand, the MRI system is very expensive (hundreds of

thousands of euros depending on the strength of the magnetic field), making it suitable only for hospitals and radiology centers. In addition, MRI is a relatively slow imaging technique, which may be problematic if the patient moves during the scan. Further, the tight space inside the MRI scanner and loud noise are problems for claustrophobic patients. Finally, the strong magnetic field involved poses a contraindication for imaging patients with certain metal implants or implanted devices [6].

In the world of cardiology, around which this Thesis revolves, CMR, also known as cardiac MRI, plays an important role, being the non-invasive gold standard method to assess cardiac function and anatomy, so it is a valuable tool for the evaluation of patients with different pathologies. Its basic principles are the same as MRI but with certain characteristics of its own, such as the use of electrocardiogram (ECG) or photoplethysmography (PPG) gating, as well as the need of rapid imaging techniques or sequences. By combining a variety of such techniques into protocols, key functional and morphological features of the cardiovascular system can be assessed [8].

Some criticisms of CMR include the lengthy test duration and high costs, as well as the possibility that some patients with claustrophobia may not be able to complete the study. Despite improvements in pacemaker compatibility with CMR, there are still some devices that are contraindicated for use with this type of imaging. Additionally, in routine clinical practice, CMR images are often taken during breath-holds to avoid respiratory artefacts that can affect image quality. This is especially important in cine images, which show the movement of the heart muscle during the cardiac cycle and require each slice to be taken during a breath-hold, which can significantly extend the time needed for the scan [9].

In general, the relative discomfort due to lengthy breath-holds or, in the case of free-breathing acquisitions, the overall scan duration, are considered drawbacks of CMR over other imaging modalities. In addition, the likelihood of image degradation due to irregular cardiac and respiratory motion tends to amplify with longer measurement times. Therefore, there

has been high demand to improve the efficiency of data collection, thereby permitting improved spatio-temporal resolutions or reduced scan times or combinations thereof.

As outlined above, cardiac cine MRI allows practitioners to visualize the myocardium dynamic behaviour throughout the cardiac cycle, which also makes it possible to calculate descriptive parameters of its function and anatomy, as well as to detect and assess abnormalities in cardiac muscle contractility. In a conventional cine examination, a set of two-dimensional (2D) slices covering the entire cardiac volume—or, at least, the left ventricle (LV)—is acquired. To mitigate the motion effect in acquisition, current techniques either require the cooperation of the patient to maintain a state of apnea during acquisition or make use of navigators, which tell the machine at what times to perform the partial acquisition phases of the image. They also require synchronisation with cardiac activity signals, such as ECG/PPG. Thus, the ultimate goal to be achieved is the improvement of patient comfort during the test.

First-pass perfusion (FPP) is another CMR modality that will be analysed in this Thesis. This modality provides means for visualizing the passage of a contrast agent through the heart during the cardiac cycle for a given slice, which allows the physician to evaluate the blood perfusion in the heart tissue for abnormality inspection. Quantitative maps can be associated to the perfusion mechanism that provide point-wise information for tissue analysis. In terms of acquisition, cine and FPP have differences, but they both share technical difficulties due to the dynamics of the heart and the need of apnea for conventional sequences.

For either modality, patient comfort would increase if the amount of information required for image reconstruction could be diminished. This translates into increased complexity in processing the acquisitions to reconstruct the images, but this is where technology plays a role. Specifically, the image information in MRI is not acquired directly in the image space but rather in k-space, which contains information about spatial frequencies

within the image. The image space and k-space are related by a Fourier transform. Once the field-of-view (FOV) and spatial resolution of the image that we wish to obtain have been prescribed, the k-space information that must be acquired is determined by the Nyquist criterion [10, 11] if conventional processing is carried out for reconstruction. One way of reducing image acquisition time is to reduce the number of profiles that populate the k-space. However, adopting this measure without further provision leads to negative effects on image quality [12].

A well-known approach to avoid this effect consists in reconstructing the dynamic series by optimizing an objective function, which typically consists of two terms, namely, a data fidelity term and a regularization term. The first term uses the same information as the conventional processing we have referred to in the previous paragraph. Regarding the second term, current solutions (other than ours) propose a spatio-temporal smoothness function, typically defined on the basis of the total variation, calculated directly on the image data. In our case we propose to apply this function to cardio-respiratory motion-compensated data. This can be done by groupwise (GW) registration, and the overall procedure is referred to as groupwise compressed sensing (GWCS) [13]. The inclusion of this term provides an improvement in the reconstruction by increasing the sparse nature of the solution. In particular, in the case of perfect motion compensation, each material point of the myocardium would remain static along the respiratory and cardiac dimensions; for this reason, the signal would be highly redundant and therefore expressible in a very small number of coefficients in a transformed domain (which is the meaning of the term *sparse* in this context).

However, the problem stems from the fact that in order to estimate the movement it is necessary to have the images, when our objective is precisely to carry out these reconstructions. Therefore, the problem is posed as successive approximations, i.e., motion can be estimated from a first reconstruction without motion compensation, motion is then compensated

for, a new refined solution can be provided, as the process starts over. The issue is that this procedure is computationally expensive.

Overall, GWCS produces good results, but it is slow and computationally demanding as it involves iterative processes to solve both the registration problem and the reconstruction itself. We believe that replacing the optimization steps with deep learning (DL) will make the reconstructions faster, and the computational complexity will be handled during the training phase. Additionally, the problem is parallelizable, and since all current solutions are limited to single-vendor devices, we suggest a device-agnostic solution.

In this Thesis we analyse and explore fast computational techniques for accelerated reconstruction of CMR dynamic images. As for cine images, two types of solutions will be sought. On one hand, we will depart from the GWCS solution, which has a bottleneck in the motion estimation (ME) and motion compensation (MC) step, and we will develop a parallel framework to give rise to an open-source device-agnostic solution. On the other, we will propose a GWCS-like solution fully based on DL. As for FPP-CMR, a modified version of the cine DL solution will be adapted to the specificities of this modality.

1.2 Objectives

The overall objective of this Thesis is to accelerate the reconstruction of 2D CMR dynamic images, for both cine and perfusion, from undersampled acquisitions. The ultimate practical goal is to increase patient comfort by shortening acquisition time, on the one hand, and to contribute to a better efficiency by achieving clinically-viable reconstruction times that would be compatible with re-acquisitions in case they were needed. We will depart from GWCS [13], a ME/MC reconstruction solution previously proposed by the research group in which this Thesis has been carried

out. Although GWCS yields competitive results, it is time-consuming and requires significant computational resources as it goes through iterative processes to solve the registration problem and the reconstruction itself. Both parallelization and DL techniques will be used to substantially speed up computations. Then, the DL solution will be adapted to solve the problem of FPP-CMR reconstruction.

In order to achieve this objective, the following specific objectives must be met:

1. To give rise to a parallel device-agnostic version of GWCS, with special emphasis on the ME/MC procedure. Results should have high quality images as well as speed in the reconstruction procedure so as to achieve clinically-viable delays.
2. To design a DL architecture that achieves GW monomodal registration of cardiac cine MRI in affordable devices.
3. To design a self-supervised DL-based solution that replicates GWCS performance at much lower computing times.
4. To adapt the self-supervised kernel of the previous solution to account for FPP-CMR reconstruction.

Figure 1.1, which will be also referred to in Section 1.5, provides a graphical overview of the specific objectives just enumerated.

1.3 Methodology

The methodology of this Thesis is based on the general research phases defined by Glass [14] and extended by Kontio [15]. The engineering method consists in observing existing solutions, proposing a better solution and developing it to subsequently measure, analyse and repeat until no further

improvements are possible. The particular steps followed throughout this Thesis can be described as follows: [16]:

- **Informational phase** consists in observing the current state-of-art and practice to identify problems and potential solutions. We found that the registration performed in the ME/MC step is a bottleneck of the GWCS reconstruction pipeline, due to its computational cost and time consumption. To solve this complex computation we can use parallelization techniques, as well as DL techniques. We identified a Graphics Processing Unit (GPU) programming solution, using OpenCL language, and also, we identified valid DL architectures for estimating motion in the context of the efficient dynamic MRI reconstruction problem.
- **Propositional phase** consists in proposing a hypothesis or a method. We aim to reduce the execution time of the groupwise registration algorithm. For this purpose, we propose, on the one hand, a GPU implementation based on free form deformations (FFDs) that allows the parallel execution of those operations that are executed pixel by pixel or point by point, and on the other hand, a DL implementation that delegates the complexity of the processing to the training stage.
- **Analytical phase** consists in analysing the proposition previously stated. Regarding the parallelization of the groupwise registration algorithm by GPU programming, we analyse which of the operations carried out by FFDs are parallelizable. Regarding the implementation of groupwise registration by means of DL, we look at which neural network architectures and configurations are appropriate for this type of image problems, as well as the complexity of such configurations, in order to avoid the need for expensive equipment for their development and use.
- **Evaluative phase** consists in testing and evaluating the propo-

sition empirically. The proposed and implemented methods were validated through several performance experiments, including measurements of both registration and reconstruction time, as well as image quality metrics. Different databases and GPUs were used in those experiments where possible.

Furthermore, as part of this Thesis, we conducted a **dissemination phase** in which we shared our motivation, final conclusions, methods, and results with the research community through international journals and conferences. This included a detailed comparison with current research in the field.

1.4 Materials

To prove the validity of the proposed rapid approaches throughout this Thesis, experiments were carried out with acquisitions performed in different MR scanners from two different vendors (Philips and Siemens) and with different pathologies (hypertrophic cardiomyopathy, coronary artery disease, patients with no cardiac pathologies). Throughout the development of the thesis, different GPUs have been used. Depending on the experiment, both the database used and the hardware resources vary. All these resources are presented below:

- Main workstation characteristics:
 - CPU Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz
 - 16 GB RAM
- GPUs used:
 - AMD Radeon RX 480
 - AMD Radeon RX 5700 XT

- NVIDIA GeForce RTX 2080 Ti
- NVIDIA Quadro RTX 6000
- Clinical tests used:
 - ♥ **Cine MRI:** clinical magnetic resonance imaging test performed to capture motion. Cine studies are typically obtained by repeatedly imaging the heart at a single slice location throughout the cardiac cycle.
 - ♥ **Perfusion MRI:** clinical magnetic resonance imaging test performed on patients with known or suspected coronary artery disease to examine the blood flow to the heart both at rest and under stress.

Details of the specific experiments are provided in each chapter.

1.5 Publications

Below, a list of publications related to this Thesis is included, as well as a diagram in which they are interrelated with the objectives previously stated (see Figure 1.1). The list is divided into three parts: indexed international journals, conference presentations and electronic posters.

- Indexed international journals:
 - **Martín-González, E.**, Sevilla, T., Revilla-Orodea, A., Casaseca-de-la-Higuera, P., Alberola-López, C. (2020). Groupwise Non-Rigid Registration with Deep Learning: An Affordable Solution Applied to 2D Cardiac Cine MRI Reconstruction. *Entropy*. 22(6):687. The organising committee of the Annual Congress of the Spanish Society of Biomedical Engineering invited us to submit the work to this journal. Journal Citation Reports: SCI JCR 2020 IF=2.587, Q2, 38/86. Physics, Multidisciplinary.

- **Martín-González, E.**, Moya-Sáez, E., Menchón-Lara, R.M., Royuela-del-Val, J., Palencia-de-Lara, C., Rodríguez-Cayetano, M., Simmross-Wattenberg, F., Alberola-López, C. (2021). A clinically viable vendor-independent and device-agnostic solution for accelerated cardiac MRI reconstruction. *Computer Methods and Programs in Biomedicine*. 207:106143. Journal Citation Reports: SCI JCR 2021 IF=7.027, Q1, 20/98. Engineering, Biomedical.
- Conference presentations:
 - **Martín-González, E.**, Casaseca-de-la-Higuera, P., San-José-Revuelta, L.M., Alberola-López, C. (2019). Groupwise Deep Learning-based Approach for Motion Compensation. Application to Compressed Sensing 2D Cardiac Cine MRI Reconstruction. In *Proceedings of the 37th Annual Congress of the Spanish Society of Biomedical Engineering (CASEIB 2019)*, pp:299-302, Santander, Spain. This work won second place for the José María Ferrero Corral prize in the communications competition held by the organising committee of the Annual Congress of the Spanish Society of Biomedical Engineering.
 - **Martín-González, E.**, Alskaf, E., Chiribiri, A., Casaseca-de-la-Higuera, P., Alberola-López, C., Nunes, R.G., Correia, T. (2021). Physics-Informed Self-supervised Deep Learning Reconstruction for Accelerated First-Pass Perfusion Cardiac MRI. In *Proceedings of the 4th International Workshop of Machine Learning for Medical Image Reconstruction (MLMIR 2021)*, held in conjunction with the 24th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2021).
 - **Martín-González, E.**, Alskaf, E., Chiribiri, A., Casaseca-

- de-la-Higuera, P., Alberola-López, C., Nunes, R.G., Correia, T. (2022). The deep SECRET to accelerated first-pass perfusion cardiac MRI. In *Proceedings of the Joint Annual Meeting ISMRM-ESMRMB & ISMRT 31st Annual Meeting (ISMRM 2022)*.
- **Martín-González, E.**, Alskaf, E., Chiribiri, A., Casaseca-de-la-Higuera, P., Alberola-López, C., Nunes, R.G., Correia, T. (2022). Deep SECRET: self-supervised accelerated myocardial perfusion MRI. In *Proceedings of the 2nd ISMRM Iberian Chapter Annual Meeting (ISMRM Iberian Chapter 2022)*.
- Electronic posters:
 - Rodríguez-Galván, J.R., **Martín-González, E.**, Alberola-López, C. (2021). A Network Solution for Self-supervised 2D Cardiac Cine Reconstruction that includes Motion Compensation. *1st ISMRM Iberian Chapter Annual Meeting (ISMRM Iberian Chapter 2021)*. Online.
 - **Martín-González, E.**, Rodríguez-Galván, J.R., Alberola-López, C. (2021). Efficient 2D Cardiac MR Cine Reconstruction through a Self-Supervised Motion Compensated (SSMoComp) Architecture. *38th Annual Scientific Meeting of the European Society for Magnetic Resonance in Medicine and Biology (ESMRMB 2021)*. Online.

1.6 Document Overview

In order to best reflect the research work that has resulted in this Thesis dissertation, this document is divided into seven chapters. The current chapter is complemented with Chapter 2, which deals with the background

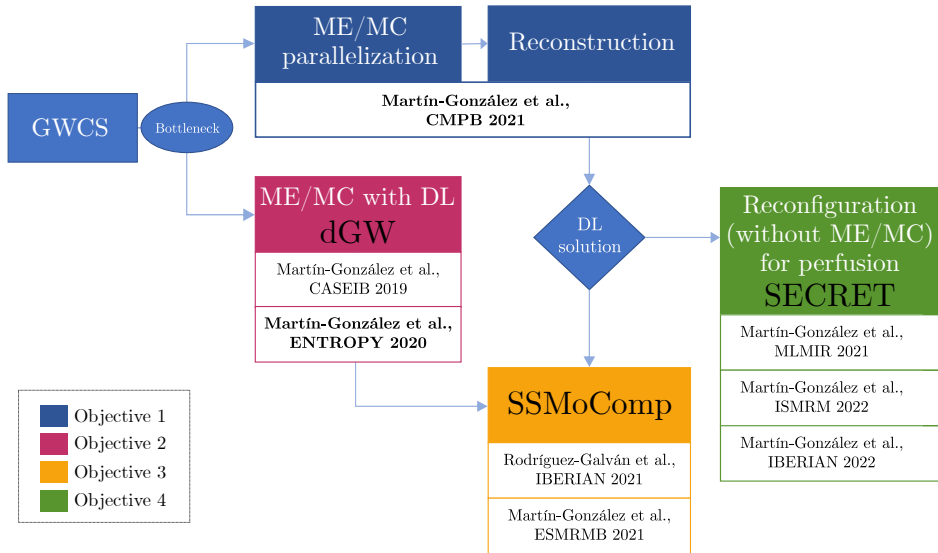


Figure 1.1: Relation between each of the contributions related to this Thesis grouped by the major objectives stated in Section 1.2.

knowledge necessary to understand the methods described in Chapters 3, 4, 5 and 6. These four chapters contain the methods developed in this Thesis, together with the description of the evaluation results and a discussion about each of them. Chapter 7 introduces some final remarks, summarizes the contributions of the thesis and proposes some ideas for future work. A more detailed description is portrayed next:

- **Chapter 1:** *Introduction*, which introduces and motivates this Thesis dissertation. The objectives pursued, the methodology used to achieve these objectives, as well as the materials and resources available during the development of this Thesis, are described. The publications and communications that have resulted from this work are also enumerated.
- **Chapter 2:** *Background*, which contains the background knowledge needed to fully understand the different issues addressed throughout this dissertation. It starts by explaining the principles of MRI and

the fundamentals of CMR together with a brief list of its applications and limitations, and finishes with the elemental knowledge needed to understand the registration and reconstruction algorithms, as well as the basics on GPU programming and DL techniques. The state-of-the-art is also reviewed.

- **Chapter 3:** *DL Groupwise Registration: dGW*, presents the work in which DL techniques are used to perform cine CMR registration using a GW paradigm.
- **Chapter 4:** *Cine MR Reconstruction with ME/MC using OpenCLIPER*, focuses on the GPU parallelization of our reference ME/MC reconstruction algorithm, GWCS. For this purpose, we use the OpenCLIPER framework, developed within this research group.
- **Chapter 5:** *DL Cine Reconstruction with MC: SSMoComp*, presents the SSMoComp framework, a self-supervised scheme for motion-compensated cine CMR reconstruction. This motion compensation is carried out with dGW, another contribution of this Thesis. Furthermore, SSMoComp is compared with the parallel implementation of GWCS in OpenCLIPER.
- **Chapter 6:** *DL Perfusion Reconstruction: SECRET*, presents SECRET, an adaptation of work carried out with DL techniques to FPP-CMR.
- **Chapter 7:** *Conclusions and future work*, which concludes this Thesis dissertation. It includes our contributions but also pinpoints our limitations as well as hypothesizes some future lines of research.

Chapter 2

BACKGROUND

This Chapter contains the background knowledge needed to fully understand the different issues addressed throughout this dissertation. It starts by explaining the principles of MRI and the fundamentals of CMR together with a brief list of its applications and limitations, and finishes with the elemental knowledge needed to understand the registration and reconstruction algorithms, as well as GPU programming and DL techniques.

2.1 Magnetic Resonance Principles

2.1.1 The Physics of MRI

Atoms, which make up objects, contain a nucleus with protons and neutrons, and have electrons orbiting around it. When an external magnetic field has the same frequency as the nuclei, it will interact with the atoms, causing NMR to occur. This is referred to as resonance.

Subatomic particles (protons, neutrons and electrons being the three main ones) have a property called *spin*, to which a small *magnetic moment* is associated. Nuclei with an odd atomic number have a net nuclear magnetic moment and are said to be MR active. There are many MR active elements such as carbon (^{13}C), fluorine (^{19}F), phosphorus (^{31}P),

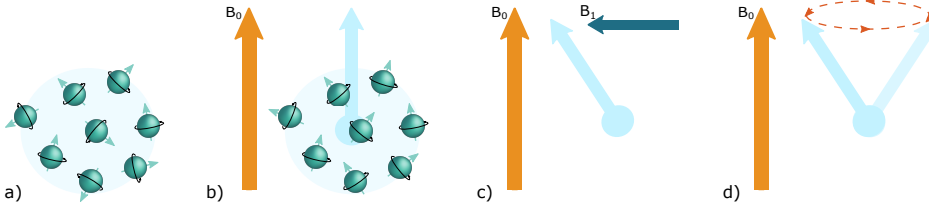


Figure 2.1: Precession. a) Magnetic moments of nuclear spins pointing in random directions due to thermal motion. b) Once B_0 is applied, magnetic moments tends to align with it. c) Displacement of M out of its equilibrium alignment by the application of B_1 magnetic field. d) Precession of M in the transverse plane at the Larmor frequency after the application of B_1 .

sodium (^{23}Na), and hydrogen (^1H); the latter being the most widely used nucleus in MRI because of its abundance in the body [6].

In the absence of an external magnetic field, the magnetic moments of all nuclear spins point in random directions (Figure 2.1a). Despite this, when an external magnetic field B_0 is applied, spins tend to align with such a field, and start to precess around the axis of B_0 tilted at a constant angle, which depends on the initial position of the spin when B_0 is turned on. The frequency of spin precession (ω_L) is proportional to the strength of the B_0 field (in units of Tesla, T) and is known as the Larmor frequency:

$$\omega_L = \gamma B_0 \text{ [rad/s]}, \quad (2.1)$$

where γ , the gyromagnetic ratio, is a nucleus-specific constant (42.58 MHz/T for hydrogen) [6]. When B_0 is applied, the distribution of the angles is random, and thus, no net magnetization is expected. However, the thermal motion of the molecules favours the direction parallel to B_0 (Figure 2.1b). The magnetic moments of the precessing spins sum up to a net non-zero magnetization, M_0 , also precessing around the axis of B_0 [6, 17].

The precessing magnetization M_0 has a non-zero component along the

axis of B_0 (called the *longitudinal direction*) and a zero component in the perpendicular plane (called the *transverse plane*). M_0 is too small relative to B_0 to be directly measured while in the longitudinal direction; therefore, it is better detected if tipped away from the longitudinal direction. This is achieved by the application of a RF magnetic field, B_1 , in a direction perpendicular to B_0 (Figure 2.1c). The frequency of this RF field is tuned to the Larmor frequency to achieve the resonance condition, enabling efficient energy transfer to the spins. The pulse is modeled as [6]:

$$\vec{B}_1(t) = B_1(t) \cdot [\cos(\omega_L t)\vec{x} - \sin(\omega_L t)\vec{y}], \quad (2.2)$$

where \vec{x} and \vec{y} are two orthogonal unit vectors in the transverse plane. As a result of the combined effect of B_0 and B_1 fields, the spins precess around an axis that is the vector sum of them, eventually spiraling down from the longitudinal direction to the transverse plane (Figure 2.1d).

The RF magnetic field, B_1 , is used to rotate the magnetization into a plane that is perpendicular to B_0 . This field is typically applied for a brief duration, τ , causing the magnetization to rotate by a specific angle, known as the *flip angle*, around the axis of B_1 . This flip angle, α , is calculated using the equation provided:

$$\alpha = \gamma B_1 \tau \quad (2.3)$$

where τ is the duration of the RF pulse, which is assumed to be constant with magnitude equal to B_1 . A flip angle of 90° applied to a magnetization initially aligned along the longitudinal direction (usually taken as the z -axis) tips the magnetization completely into the transverse plane, producing maximum signal. This RF pulse is called an *excitation pulse*. Excitation pulses of smaller flip angles ($<90^\circ$) are also commonly used to partially rotate the longitudinal magnetization. In this case, the excitation RF pulse, α , creates a transverse magnetization component $M_{xy} = M_0 \sin(\alpha)$,

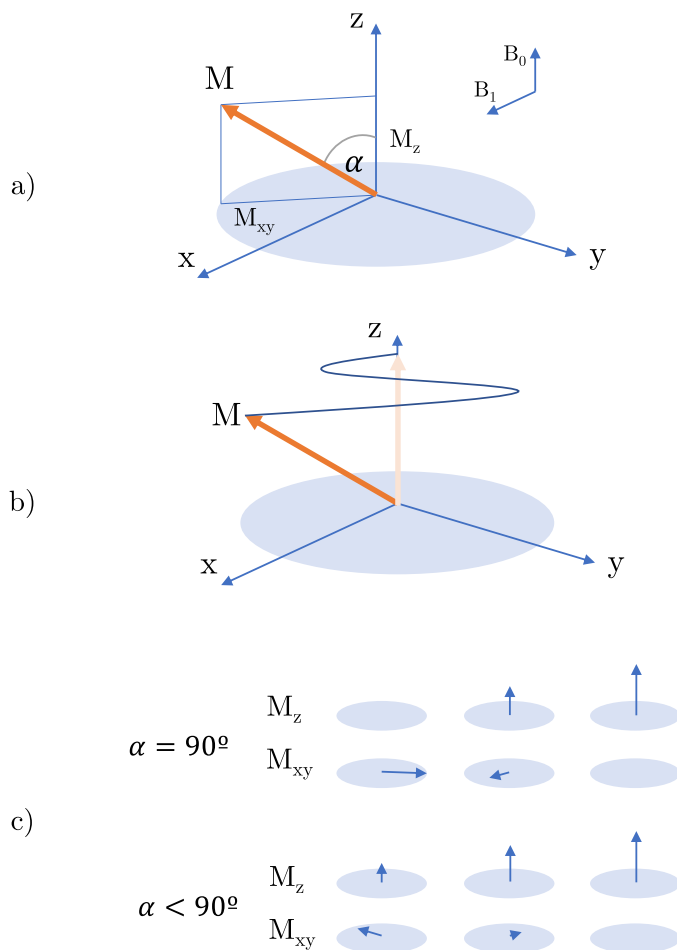


Figure 2.2: Magnetization patterns. Evolution of components. a) RF pulse application for arbitrary alpha, b) relaxation for RF pulse of arbitrary alpha, c) evolution of the magnetisation components in relaxation after a 90° RF pulse and after an RF pulse of less than 90° .

leaving the longitudinal magnetization at a value of $M_z = M_0 \cos(\alpha)$ [6].

If a loop of wire (RF coil) oriented orthogonal to B_0 is brought close enough, the precession of the magnetization M_0 induces an electric current in the coil according to the Faraday's law of induction. This is the MR signal used to construct the MRI image, which will be described further

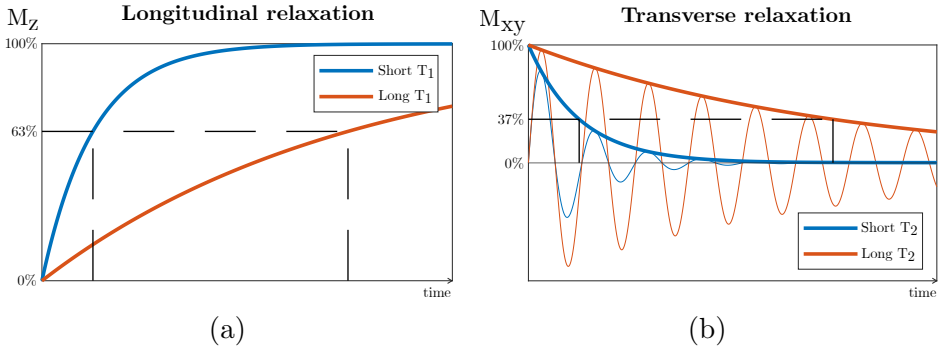


Figure 2.3: Relaxation of both (a) longitudinal (M_z recovery) and (b) transverse (M_{xy} decay) components for different constant decays, together with FID.

below.

When the RF pulse used to excite the magnetization vector is turned off, the vector begins to return to its equilibrium position (aligned with the longitudinal direction and at a magnitude of M_0). This process results in both the transverse M_{xy} and longitudinal M_z magnetization components decaying and recovering at different rates, with time constants T_2 and T_1 , respectively (as shown in Figure 2.3). The signal received from the transverse magnetization's decay is referred to as the free induction decay (FID) and has the shape of an exponentially decreasing sine wave (see Figure 2.3b).

The process of getting back to normal is described by an exponential growth and is defined by the constant T_1 . It takes T_1 for M_z to reach 63% of its final value (M_z is close to M_0 after $5 \times T_1$). T_1 is known as the *spin-lattice* or longitudinal relaxation time constant and is unique to each type of tissue. The way M_z recovers over time, t , can be represented mathematically as

$$M_z = M_0 \left(1 - e^{-\frac{t}{T_1}} \right), \quad (2.4)$$

where $t = 0$ is the time point immediately after the end of the excitation pulse.

In contrast to the growing behavior of M_z during relaxation, M_{xy} experiences an exponential decay during relaxation. The time constant that characterizes the decay of M_{xy} is called *spin-spin*, or transverse, relaxation time, which is a tissue-specific characteristic and is denoted by T_2 . It should be noted that T_2 is always less than or equal to the longitudinal relaxation time constant T_1 . The decay of the transverse magnetization (in the rotating frame of reference) is mathematically expressed as

$$M_{xy} = M_0 e^{-\frac{t}{T_2}} \quad (2.5)$$

In practice, two factors contribute to the actual decay of M_{xy} . The first is the natural spin-spin interaction, characterized by the T_2 time constant. The second factor is denoted by T_2' time constant, which is a result of the inhomogeneity in the B_0 field ($1/T_2' = \gamma \Delta B_0$), which causes differences between the resonance frequencies of the neighboring precessing nuclei. The effective transverse decaying time constant is denoted T_2^* , given by

$$\frac{1}{T_2^*} = \frac{1}{T_2} + \frac{1}{T_2'}. \quad (2.6)$$

T_2^* is always shorter than T_2 . It should be noted that while the T_2 decaying effect is an intrinsic property of the tissue, the T_2' effect can be reversed. Both T_1 and T_2 are field dependent. At higher magnetic (B_0) fields, T_1 becomes longer, whereas T_2 stays approximately the same.

The effect of the net magnetization under the application of B_1 can be described by a phenomenological differential equation known as the *Bloch equation*, which relates the rate of change of (\mathbf{M}) in the presence of a magnetic field (\mathbf{B}):

$$\frac{d}{dt} \mathbf{M}(t) = \gamma \mathbf{M}(t) \times \mathbf{B}(t) - \frac{M_x(t) \vec{x} + M_y(t) \vec{y}}{T_2} - \frac{(M_z(t) - M_0) \vec{z}}{T_1} \quad (2.7)$$

where \mathbf{M} and \mathbf{B} are the vector notations for the magnetization vector and

the applied magnetic field, respectively, and \times denotes the cross (vector) product [6].

The vector \mathbf{B} encompasses both the constant magnetic field, B_0 , and any additional RF field, B_1 . The first component of the Bloch equation depicts the rotation of the magnetization in response to the magnetic field, while the second and third components depict the transverse and longitudinal decay of the magnetization. When relaxation effects are disregarded, the Bloch equation results in the fundamental Larmor rotation of the magnetization vector \mathbf{M} around the direction of \mathbf{B} .

To obtain sufficient data for reconstructing the MRI image, RF excitation and signal recording are usually repeated multiple times. The series of RF pulses and accompanying gradients (described in Section 2.1.2) form the MRI pulse sequence. Broadly speaking, the time duration between two successive excitation RF pulses is called the repetition time (TR), and the time duration between the application of an excitation RF pulse and signal acquisition is called the echo time (TE). While increasing TR allows for more longitudinal magnetization recovery and stronger signals, increasing TE causes more transverse magnetization decay and signal loss [6].

2.1.2 Image Formation

Now that we have discussed the fundamental concepts of NMR, we can delve into how MRI generates images of the spatial distribution of spins through the process of relaxation in an external magnetic field. Specifically, we will explain how this principle is used to create 2D images.

By altering the strength of the external magnetic field in specific locations, it is possible to distinguish the positions of different spins within the body. This is because the resonance frequency of each spin is proportional to the strength of the magnetic field at its location. This technique provides the means to limit spin excitation to a single slice within the body. The orientation of the selected slice in MRI can be selected arbitrarily by simple

application of a field B_G , whose amplitude varies linearly with the position and the strengths of three gradients:

$$B_G = G_x x + G_y y + G_z z \quad (2.8)$$

The presence of such gradient modifies the precession frequency of the spins of the object along its direction

$$\omega(x, y, z) = \omega_0 + \gamma B_G. \quad (2.9)$$

In addition to selecting slices, the gradient fields can also be used to identify specific positions within a slice through a process called *frequency encoding*. This technique creates a one-dimensional projection of the image along the direction of the gradient field. Similar to frequency encoding, *phase encoding* uses gradient fields. The net effect of applying both phase encoding and frequency encoding is modulating the signal in each pixel with a position-dependent phase and frequency. All signals are then recorded into a 2D data matrix, called k-space.

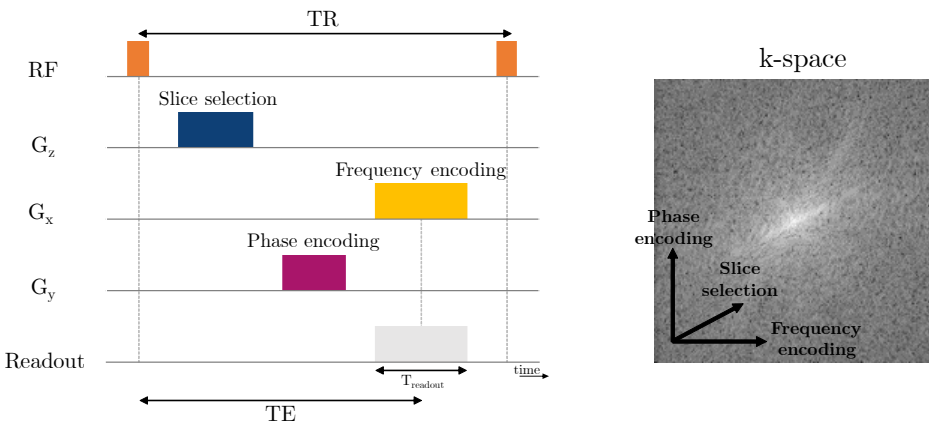


Figure 2.4: Simplified pulse sequence representing slice selection, frequency encoding and phase encoding stages. An example of k-space can also be observed.

2.1.2.1 K-space

The method for obtaining an MR image using frequency encoding results in $N_x \times N_y$ pieces of data, where each signal is generated from all the spins in the specific slice being examined. This collection of signals forms the spatial frequency content of the image, also known as the k-space. The relationship between the image (I) and the k-space (or signal, S) is determined through the use of the Fourier transform

$$S(t) = S(k_x(t), k_y(t)) = \iint_{-\infty}^{\infty} I(x, y) e^{-j2\pi(k_x(t)x + k_y(t)y)} \quad (2.10)$$

where k_x and k_y are related to the applied gradients by

$$k_x(t) = \frac{\gamma}{2\pi} \int_0^t G_x(\tau) d\tau, \quad k_y(t) = \frac{\gamma}{2\pi} \int_0^t G_y(\tau) d\tau. \quad (2.11)$$

The criterion of filling k-space is called the k-space trajectory. If the data in k-space lie on a uniform grid, i.e., if k-space is filled line by line in a cartesian fashion, the Fast Fourier Transform (FFT) algorithm is implemented to efficiently reconstruct the MRI image.

The gradient waveforms can be altered to create a variety of k-space trajectories, including spiral and radial. These trajectories are characterized by their ability to oversample the k-space center, which makes them less prone to artifacts. However, using non-linear trajectories requires additional steps, such as regridding, in order to properly reconstruct images using FFT techniques.

2.1.2.2 Multislice vs. 3D Imaging

There are two ways to study a whole volume with an MRI exam: multislice and volumetric 3D imaging. Multislice involves taking multiple 2D slices to cover the volume. This method has low resolution in the slice selection

direction since it requires a large slice thickness to achieve an acceptable signal-to-noise ratio. Volumetric 3D imaging, on the other hand, acquires the signal by exciting the whole volume, providing high resolution in all directions with improved signal-to-noise ratio. However, this method requires a longer scan time, which makes it more susceptible to problems caused by patient movement.

2.1.3 Image Artifacts

Artifacts are unwanted elements that appear in an MRI image. These can be caused by issues with the MRI scanner, such as poor performance or malfunction, as well as factors like patient movement, incorrect scan parameter selection, or inherent physical limitations. Examples of common MRI artifacts include:

- **Motion artifacts** One of the main reasons for inconsistencies or errors in MRI images is the movement of the patient during the scan. This movement can be intentional or unintentional, and include breathing, heartbeats, eye movements, swallowing, blood flow and many others. This motion can cause issues with data acquisition and result in blurred or distorted areas in the image, particularly along the phase-encoding direction in Cartesian acquisitions.
- **Ringing artifact** The ringing artifact, Gibbs artifact or truncation artifact is a visual phenomenon characterized by a series of bright and dark stripes that appear along the edges of an image. This artifact is caused by the sharp truncation of k-space. This results in a lack of high-frequency information, which becomes visible as the Gibbs artifact along sharp edges in the image.
- **Partial volume artifact** The partial volume artifact occurs when there is a mix of different tissues within a single imaging voxel,

leading to a blurry appearance in the image. This can also happen when the slice thickness used for imaging is too large.

- **Aliasing artifact** Aliasing, fold-over, or wrap-around artifact occurs when the field of view of the imaging device is smaller than the size of the organ being imaged. As a result, tissues outside of the field of view will appear to fold back into the field of view, potentially obscuring other details in the image.
- **Off-resonance artifact** Spins that rotate at different frequencies are referred to as off-resonance. This can be a natural characteristic of the tissue, such as the frequency difference between fat and water, or it can be caused by variations in the magnetic field or the tissue's susceptibility to the field.

2.2 Cardiac Magnetic Resonance Imaging: CMR

Cardiovascular imaging, which involves the use of various techniques to examine the heart and blood vessels, has long been a key part of patient diagnosis and treatment. One popular method is echocardiography, which is relatively inexpensive and portable. X-ray-based techniques, such as fluoroscopy and CT scans, are also commonly used. CMR imaging has become the gold standard for evaluating heart function, due to its high tissue contrast and spatial resolution, 3D imaging capabilities, lack of ionizing radiation, and ability to adjust various parameters to assess different aspects of cardiac function. As CMR technology continues to improve, its applications are expanding. In addition to providing anatomical information, CMR imaging can also provide functional, perfusion, viability, and metabolic information about the heart muscle, as well as information about blood vessels and flow dynamics [6].

This section briefly presents the CMR techniques used in this Thesis, namely cardiac cine MR and FPP-CMR.

2.2.1 Cine Imaging

Cine CMR is a highly effective method for imaging the heart's contractile function due to its high spatial and temporal resolution. It is commonly used in CMR examinations and is considered the best non-invasive option for assessing contractile function. Cine CMR can provide numerous details about the heart's function, including overall function, the movement of the LV and right ventricle (RV) walls, ventricular volume, ejection fraction, and mass measurements. It is often used as a benchmark for evaluating LV remodeling and as reference method for other imaging techniques.

The CMR sequence is a technique used to create a movie-like representation of the heart that shows its contractile function, as can be seen in Figure 2.5. The standard protocol involves obtaining images through a process called *segmented acquisition*, which involves dividing the acquisition of data over multiple heartbeats to improve resolution. Each segment represents a different phase of the cardiac cycle and is acquired at a different time within the RR interval. The raw data are then sorted by cardiac phase and the resulting images can be displayed as a movie of the beating heart. For patients with breathing problems or irregular heart rhythms, real-time cine techniques are available that allow for free-breathing and do not require an ECG signal.

Typically, a stack of multiple closely spaced (contiguous or with 1-2 mm gaps) short-axis slices of 6-8 mm thickness is acquired to provide full coverage of the LV and RV. Short-axis views, perpendicular to the long-axis views, can be planned on long-axis scout images. In addition, cine images can be obtained in multiple long-axis orientation, such as the two-chamber, three-chamber, or four-chamber views.

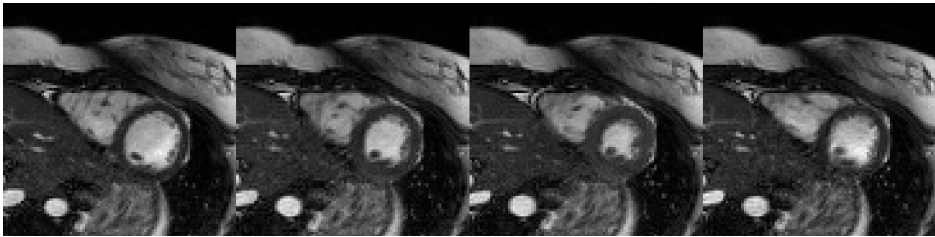


Figure 2.5: Frames extracted from cine series. Four frames of a short axis acquisition selected to show the contractile function of the heart, which is evident in the third one.

2.2.2 Perfusion Imaging

Over the last decade, a series of MRI technical and clinical advancements have led to the growing use of myocardial perfusion CMR imaging for the detection of coronary artery disease (CAD). CAD is the occlusion of the coronary arteries usually caused by atherosclerosis, which causes abnormalities in blood flow to the heart. Non-invasive imaging techniques that are widely used clinically for the evaluation of CAD are SPECT and PET, but the reference for non-invasive myocardial perfusion quantification is PET [18]. However, the clinical value of FPP-CMR has been shown in comparison to these techniques [18–21], having emerged as an alternative way of detecting blood flow anomalies without the use of potentially harmful ionising radiation. In addition, FPP-CMR has other advantages, such as higher spatial resolution, wider availability and lower scan cost compared to PET.

Perfusion CMR acquires a temporal series of single-shot images that are later played as a movie showing the transit of a gadolinium-based contrast media during its first pass through the myocardium, as can be seen in Figure 2.6. Stress perfusion is performed during pharmacological vasodilation, usually with adenosine. A perfusion defect is typically represented by a “hypo-enhanced,” darker area. As a last step of a CMR examination, the same perfusion scan is repeated at rest, meaning in absence of a stress

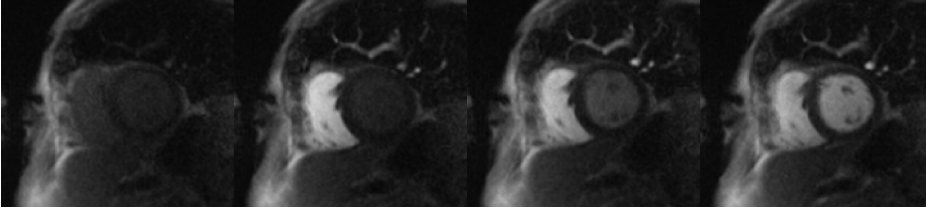


Figure 2.6: Frames extracted from FPP-CMR series. Four frames of a short axis acquisition selected to show the passage of the contrast agent through the heart. It can be seen how in the first image the contrast has not yet arrived and in the following images how it passes from the RV to the LV.

agent. Comparison of stress and rest scans and of delayed-enhancement images at the same slice locations allows for identifying reversibly and irreversibly injured myocardium [6]. This test allows to differentiate abnormal myocardial tissue in which there is an excess accumulation of the contrast agent relative to normal tissue.

2.3 Image Reconstruction

As explained previously, MRI data are not acquired directly in image space. A set of data processing steps are required to turn the acquired raw data or k-space data into actual images that can be interpreted by a clinician. The process of transforming the acquired data to images is called *image reconstruction* and it is carried out by dedicated reconstruction software. In the simple case, where the entire k-space is fully sampled, the inverse Fast Fourier transform (iFFT) can then be used to reconstruct the k-space data into clinically interpretable images, as in Equation (2.12),

$$\mathbf{y} = \mathcal{F}\mathbf{m}, \quad \mathbf{m} = \mathcal{F}^{-1}\mathbf{y}, \quad (2.12)$$

where \mathbf{y} is the acquired k-space data, \mathbf{m} is the image, and \mathcal{F} and \mathcal{F}^{-1} are the FFT and iFFT operators.

Due to the sequential nature of MRI scanning, acquisition time is roughly proportional to the number of k-space samples collected. Therefore, it is desirable to collect as few samples as possible. However, if the sampling rate is reduced below that required by the Nyquist criterion, aliasing artefacts will appear in the image [22].

In general terms, the image reconstruction can be formulated as the following inverse problem:

$$\mathbf{y} = \mathbf{E}\mathbf{m} + \boldsymbol{\epsilon} \quad (2.13)$$

where \mathbf{y} is the measured k-space data, \mathbf{E} is the system matrix or encoding operator, \mathbf{m} is the image and $\boldsymbol{\epsilon}$ is a random noise term. When k-space data are undersampled and noise corrupted, the inverse problem in Equation (2.13) is ill-posed: a solution might not exist, infinite solutions might exist, and it may be unstable with respect to measurement errors. As a result, direct inversion is generally not possible. Instead, an optimal solution in the least-squares sense may be obtained by recasting the problem as the following minimization:

$$\hat{\mathbf{m}} = \arg \min_{\mathbf{m}} \frac{1}{2} \|\mathbf{E}\mathbf{m} - \mathbf{y}\|_{\ell_2}^2 \quad (2.14)$$

Much research effort has been devoted to image reconstruction from an undersampled k-space over the last few decades. Two broad technologies stand out for their importance, namely parallel imaging (PI) and compressed sensing (CS). They will be briefly described below, as well as groupwise compressed sensing, on which we rely for the development of this Thesis. These enable substantial reductions in acquisition time while preserving image quality [22]. In addition, DL techniques have made a breakthrough in the field, sometimes achieving better results than traditional techniques.

In this section, the fundamentals of image reconstruction are described

as well as different ways to accelerate acquisitions. We will focus on techniques that use prior knowledge and regularization, paying more attention to those that have served as the basis for the development of this Thesis.

2.3.1 Parallel Imaging

As mentioned above, it is possible to accelerate the acquisitions by under-sampling the k-space, i.e. collecting less lines in k-space. The resulting spatial aliasing must be removed before the images can be used for clinical purposes. PI has been developed to address this issue [23]. There are some common characteristics shared by all PI methods, which are listed below:

1. Instead of using a single large coil to acquire data, N_c smaller coils are used simultaneously to collect data. These coils are more sensitive to the specific tissue areas closest to them, which provides additional spatial information for image reconstruction. The signal received by each coil s_n can be written as

$$s_n = C_n \cdot s, \quad n = 1, \dots, N_c \quad (2.15)$$

where C_n is the coil sensitivity map and s is the original image.

2. Some k-space lines can be skipped to reduce the scan time. If the Nyquist criterion is not met, the result is an aliased image. The acceleration factor, AF, is defined as the ratio of the amount of k-space data required for a fully sampled image to the amount collected in an accelerated acquisition.
3. An algorithm that takes into account the specific sensitivities of each coil is used to synthesize the undersampled data from each receiver coil into a single, unaliased image, trying to solve the problem in Equation (2.13), where the system matrix E is composed by k-space sampling, FFT and coil sensitivities.

PI can be used with any pulse sequence to reduce scan time. Nevertheless the speed advantage of PI is penalized by a reduction in signal-to-noise ratio (SNR) by a factor of \sqrt{AF} [6].

There are two widely used PI techniques. The SENSitivity Encoding (SENSE) method works by unfolding the aliasing in the image domain using the known coil sensitivity maps, and will be briefly described below [24]. As an alternative, k-space methods try to estimate the missing k-space lines from the acquired data before implementing the Fourier transform as in the GeneRALized Autocalibrating Partially Parallel Acquisition (GRAPPA) method [25]. Both methods involve using additional k-space lines to measure coil sensitivities and using this information to weight image reconstruction. Additional temporal undersampling techniques like TSENSE [26] and TGRAPPA [27] have been developed, but a combination of spatial and temporal undersampling, such as k-t SENSE, k-t GRAPPA and k-t BLAST (Broad-use Linear Acquisition Speed-up Technique), is becoming more popular as it takes advantage of correlations in both k-space and time [28–30].

2.3.1.1 SENSE

This technique is performed in the image domain after reconstruction of data from the individual coils, using the known coil sensitivity maps [24]. The first and most important step in this method is the generation or calculation of the sensitivity maps of the coils. These sensitivity maps quantify the relative weighting of signals coming from different locations within the coverage area of each coil. After the maps are determined, the data from each coil's partial k-space view are obtained and the partial FOV images for each coil are reconstructed. The partial FOV images are then combined using matrix inversion.

The key idea of SENSE is that the superposition of signals occurs with different weights according to the local coil sensitivities. Let us take as an

example an acquisition with two coils and $AF=2$. Due to undersampling, the images obtained from each coil suffer aliasing and therefore each pixel p obtained from each coil has contributions from two pixels (A and B , in this example) of the image we want to reconstruct; which can be written as follows:

$$\begin{aligned} p_1 &= s_{1A} \cdot A + s_{1B} \cdot B \\ p_2 &= s_{2A} \cdot A + s_{2B} \cdot B \end{aligned} \tag{2.16}$$

where s denotes the coils sensitivities. The problem can then be formulated as follows:

$$\mathbf{y} = \mathbf{E}\mathbf{m} = \mathbf{A}\mathcal{F}\mathbf{S}\mathbf{m} \tag{2.17}$$

where \mathbf{A} represents the undersampling mask, and \mathbf{S} denotes the coil sensitivity maps¹.

The matrix inversions necessary for the unfolding process in Equation (2.17) can only be performed if the AF is less than or equal to the number of coils. However, in most cases this is not possible and must be addressed through iterative optimization.

2.3.2 Compressed Sensing

As well as PI, CS is an image reconstruction method for undersampled acquisitions. CS is based on the fact that if the underlying image exhibits transform sparsity, where sparse is understood to mean that the majority of the coefficients of a vector are equal to zero and very few coefficients will contain all the information, and if k-space undersampling results in incoherent artifacts in that transform domain, then the image can be recovered from randomly undersampled frequency domain data, provided an appropriate nonlinear recovery scheme is used [31,32].

¹It should be noted that in the case of single-coil acquisitions \mathbf{S} is a matrix of ones.

The reconstruction is performed by minimizing the ℓ_1 norm of a transformed image, i. e. the sum of the absolute values of the pixels in the transform domain, is minimized. The smaller the ℓ_1 norm, the higher the sparsity. This minimization is subject to data fidelity constraints [31]:

$$\begin{aligned} & \text{minimize } \|\Psi \mathbf{m}\|_{\ell_1} \\ & \text{s.t. } \|\mathbf{E} \mathbf{m} - \mathbf{y}\|_{\ell_2} < \epsilon, \end{aligned} \quad (2.18)$$

ϵ controls the fidelity of the reconstruction to the measured data and Ψ is the linear operator that transforms from pixel representation into a sparse or compressed representation the image \mathbf{m} to be reconstructed. The threshold parameter ϵ is usually set below the expected noise level.

The optimization process of CS is then described by the following equation:

$$\min \|\mathbf{E} \mathbf{m} - \mathbf{y}\|_{\ell_2}^2 + \lambda \|\Psi \mathbf{m}\|_{\ell_1} \quad (2.19)$$

CS involves multiple steps and data transformations, guided by an iterative optimization algorithm, following a scheme similar to this [33]:

1. The first step is to measure the k-space \mathbf{y} . Two characteristics of the acquisition must be taken into account: strong sub-sampling and incoherent sampling. The undersampled k-space is transformed into an image \mathbf{m} using iFFT. This image serves as a starting point for the iterative optimization.
2. The image \mathbf{m} is then transformed into a sparse representation $\Psi \mathbf{m}$. There are various different transformations Ψ that can be beneficial for this purpose.
3. After the Ψ transformation, the noise can be removed by a thresholding procedure. Then, the Ψ -space representation is transformed back into image space with the inverse Ψ transformation (Ψ^{-1}).
4. An FFT is applied to transform the image back to k-space, with all

spatial frequencies so it is necessary to mask this k-space by only depicting the points present in the sampling trajectory. The FFT and masking together are the encoding operator \mathbf{E} in Equation (2.19). The resulting k-space $\mathbf{E}\mathbf{m}$ can then be directly compared to the measured k-space \mathbf{y} .

5. We now subtract the k-space $\mathbf{E}\mathbf{m}$ from step 4 from the measured k-space \mathbf{y} from step 1. The difference ($\mathbf{E}\mathbf{m} - \mathbf{y}$) corresponds to the error (non-consistency) that the thresholding from step 3 has created compared to the measured k-space \mathbf{y} . The image \mathbf{m} is now updated by adding this difference image previous Fourier transformation.
6. Steps 2 to 5 are now repeated. Each iteration will increase the sparsity (in Ψ -space). This is an alternating optimization of data consistency and transform sparsity. The factor λ in the formula is a weighting that defines the trade-off between data consistency and sparsity.

This is repeated until:

- either the least-squares difference of the data consistency term (step 5), is smaller than a predefined threshold ϵ ,

$$\|\mathbf{E}\mathbf{m} - \mathbf{y}\|_{\ell_2} < \epsilon \quad (2.20)$$

- or a predefined number of iterations is reached.

The final image will (very closely) look as if we had measured k-space completely, but at a much shorter scan time.

In the case of dynamic cardiac MRI, along the time dimension, the sparsity is quite high. In CS cardiac cine, only little changes are expected between subsequent time frames due to the high temporal resolution and the static anatomy surrounding the heart. This increases the transform

sparsity along the time dimension, and therefore high accelerations factors can be achieved in CS cardiac cine.

In the last decade, research has proposed combining CS reconstruction with motion correction techniques. For example, the k-t FOCal Underdetermined System Solver (FOCUSS) method incorporates motion estimation to improve sparsity in the sparse representation for better CS reconstruction [34–36]. However, this method only works in situations where all data are acquired in one motion state. Other methods have been proposed to handle data acquired in different motion states, such as combining CS with PI and one-dimensional (1D) translational respiratory motion correction [37], or performing CS reconstruction from data acquired in each motion state and averaging the results after image-based affine registration [38]. However, these methods are limited in their ability to correct for arbitrary non-rigid motion. A CS-based motion correction framework is needed to correct for arbitrary non-rigid motion in the CS reconstruction using data acquired at multiple motion states. A generalized motion correction framework, introduced by Batchelor et al. [39], can correct for general (affine or non-rigid) motion in the image reconstruction by modeling the transformation from the motion free image to the acquired motion corrupted k-space samples at different motion states via a matrix-vector equation. Using information from multiple coils in parallel MRI, the application of this framework has been demonstrated in brain imaging [39], coronary MRI [40], cardiac cine [41, 41–43], and liver MRI [44] to correct for non-rigid motion.

2.3.3 Groupwise Compressed Sensing

GWCS is a reconstruction algorithm that was published in [13] and later improved in [45], and exploits the redundancy of information along the time dimension. It departs from the multi-coil k-space undersampled information and solves a CS reconstruction problem. This is done following

a classical optimization procedure based on a data fidelity term and a regularization term. In this second term, sparsity is fostered by means of penalizing the ℓ_1 norm of the temporal total variation (tTV) of the registered image set; this is where the GW-MC operator \mathcal{T}_{Θ} is applied [13]. Briefly described, the algorithm departs from the k-space subsampled information \mathbf{y} and solves a CS reconstruction problem which gives the resulting image sequence \mathbf{m}_i , at iteration step i , as

$$\mathbf{m}_i = \arg \min_{\mathbf{m}} \frac{1}{2} \|\mathbf{E}\mathbf{m} - \mathbf{y}\|_{\ell_2}^2 + \lambda \|\Phi \mathcal{T}_{\Theta} \mathbf{m}\|_{\ell_1} \quad (2.21)$$

where \mathbf{m}_i is a vectorized stack of image frames, one frame per cardiac phase. \mathbf{E} is the encoding operator that includes the multiplication by the coil sensitivities \mathbf{S} , the intra-frame spatial Fourier transform \mathcal{F} and the application of the undersampling mask \mathbf{A} , as in Equation (2.17). Φ is the tTV operator, \mathcal{T}_{Θ} is the GW transformation for MC that registers all the frames in the sequence to a common reference and λ is a regularization parameter, so the regularization term is given by:

$$\|\Phi \mathcal{T}_{\Theta} \mathbf{m}\|_{\ell_1} = \frac{1}{|\mathcal{X}|N} \int_{\mathbf{x} \in \mathcal{X}} \sum_{n=1}^N |\mathbf{m}_n(\mathcal{T}_n(\mathbf{x})) - \mathbf{m}_{n-1}(\mathcal{T}_{n-1}(\mathbf{x}))| \quad (2.22)$$

where cyclical motion has been considered by setting $\mathbf{m}_N(\mathbf{x}) = \mathbf{m}_0(\mathbf{x})$. As for \mathbf{m}_0 , the transformation \mathcal{T}_{Θ} is the identity, since there are no data from which motion information can be estimated. Then, the iterative refinement process finds \mathbf{m}_i with \mathcal{T}_{Θ} obtained by the heart ME on \mathbf{m}_{i-1} .

The reconstruction process is iterative, so that with each iteration the provided solution is refined. This means that registration has to be performed several times. ME and MC techniques lack the true dynamic image to estimate the motion information from. A common approach is to perform the ME step from an initial reconstruction of the images themselves that, however, will be affected by the artifacts introduced by the undersampling pattern that the initial reconstruction could not

correct, what hinders the estimation of the true motion information. This effect becomes more relevant as the AF increases [13]. The registration itself is also iterative, since transformations and gradients therein must be repeatedly calculated. This is highly computationally expensive, so registration constitutes the bottleneck of the reconstruction algorithm. The following section explains the registration method used.

Equation (2.21) is minimized with the well-known NESTA algorithm based on Nesterov’s method [46]. Specifically, Nesterov’s method iteratively minimizes a function f by estimating three sequences \mathbf{x}_k , \mathbf{y}_k and \mathbf{z}_k . The \mathbf{x}_k sequence corresponds to the sequence we want to estimate (\mathbf{m}_i in the Equation (2.21)), and it is obtained from a weighted sum of the other two sequences. Nesterov’s method can be used for the minimization of both smooth and non-smooth convex functions if using the appropriate smoothing techniques. In [46], the ℓ_1 -norm in Equation (2.21) is component-wise approximated by the well-known Huber function $f_\mu(x)$, which depends on a smoothing parameter μ ; this parameter is iteratively decreased during the optimization process. The starting guess \mathbf{x}_0 is the result of applying the adjoint encoding operator (\mathbf{E}^H) to the subsampled k-space ($\mathbf{x}_0 = \mathbf{E}^H \mathbf{y}$).

2.4 Image Registration

As stated above, image registration is necessary in order to carry out the ME/MC steps that take place in the GWCS algorithm. ME involves the steps to estimate the motion (the deformation created) from a reference and MC refers to the steps in which the information obtained with ME is used to deform each cardiac phase to that reference to achieve a pseudo-static sequence. i.e. the application of \mathcal{T}_Θ . This section focuses on its description.

Image registration is the process of aligning two or more images for subsequent joint processing; it is especially widespread in the medical imaging field either for processing images acquired in different time instants–

or with different imaging devices—for the same patient or to carry out comparative analyses with multiple patients [47, 48]. Registration is a three-tier process, in which the following elements must be defined:

- (a) a deformation model,
- (b) a measure of similarity between the entities to be aligned and
- (c) an optimization procedure.

According to the deformation model, registration may be labelled as rigid or non-rigid; the latter is still subdivided as affine and elastic. Rigid models are often used for brain longitudinal studies on the same patient, while non-rigid or elastic models are used for moving deformable organs, such as the heart. On the one hand, rigid registration uses a simple uniform mapping between images related by rotation and translation, examples of linear transformations. On the other hand, elastic registration allows a non-uniform mapping between images, to correct small, varying discrepancies by deforming one image to match the other. That is, it uses non-linear transformations, providing higher accuracy as well as increased computational complexity. Optimization procedures must be chosen according to the deformation model employed. As for the measure of similarity, both monomodal and multimodal metrics can be found; monomodal registration is based on the assumption that corresponding pixels have similar intensity values; otherwise, multimodal registration tries to find correspondence between images acquired using different imaging modalities. This is because different modalities may assign different intensities to the same structure [49]. Registration of monomodal dynamic medical images is frequently encountered because of patient motion, either involuntary or spontaneous (heart pumping or breathing) as well as small voluntary motion due to discomfort during the scanning or to non-cooperative patients (children, elderly, ...). This is the kind of registration we will focus on. Follow up of patients is also an application domain for this type of registration [50].

Registration of multiple images can be tackled as a pairwise (PW) or as a GW problem. The former may be carried out sequentially, i.e., the next image in the sequence is registered against the previous image or, alternatively, one of the images in the sequence may be selected as the reference image while the others will be registered (in pairs) against that reference. On the other hand, the GW approach consists of a joint optimization problem in which the reference is created with information of the whole temporal sequence. The choice of a reference space can have a large impact on the quality of the result. GW methods typically outperform PW since the problem is solved as a whole. However, this comes at the expense of a higher computational cost, due to their iterative nature [51].

2.4.1 Deformation

As mentioned above, there are two types of image registration depending on the deformation model: rigid and non-rigid, but also there are two types of image registration depending on the deformation parameterization, i.e. how the deformation is represented in a computer: parametric and non-parametric. Parametric image registration involves a model with fixed parameters that transforms the input image. Non-parametric image registration, conversely, does not involve a fixed set of parameters. [52].

In parametric registration, there is a finite set of unknown values known as Θ that are used to create a spatial transformation called \mathcal{T}_Θ using a system of basis functions. For instance, the values of Θ may include displacement vectors located at control points on a grid, and the continuous transformation \mathcal{T}_Θ is created through interpolation using these displacement vectors. Examples of parametric registration are Thin-plate spline (TPS) and FFD with B-splines. The use of TPS interpolation as point-based registration method of medical images was first proposed by Bookstein [53] and it is based on the use of landmarks. This technique involves mapping source and target landmarks onto each other exactly, but in real situations

the positions of the landmarks can only be roughly determined. Meanwhile, in FFD based registration, a B-spline basis function is commonly utilized to build the transformation model. The deformation of an image is determined by a small number of control points that are evenly spaced on the fixed image's grid. This creates two grids that are aligned with each other: a densely populated pixel grid and a more spread out control point grid. Cubic splines, which are bi-cubic polynomials that are parameterized using a grid of control points, are commonly used for parametric registration [54]. These splines have limited influence, only affecting the grid cells around each control point and not affecting the parametric function \mathcal{T}_{Θ} outside of this region.

Non-parametric registration involves treating the transformation parameters, represented by Θ , as an unknown function, which is solved using variational calculus. Examples of non-parametric registration include Demons and Large Deformation Diffeomorphic Metric Mapping (LDDMM). The Demons algorithm is a fast method that utilizes basic image processing techniques such as combining transformation fields, calculating image gradients, and using a Gaussian convolution kernel [55]. LDDMM, on the other hand, defines non-parametric registration as a problem of minimizing energy while ensuring the solution fits the characteristics of diffeomorphism. This approach allows for precise measurements and statistical analysis in the infinite-dimensional space of images, making it widely used in research on anatomical variability [56, 57].

We now describe what the deformation we are focusing on consists in. A basic assumption in the development of mechanical models of the heart is that the heart is a single, spatially continuous entity [58]. A deforming tissue has its particles moved from their reference position, \mathbf{x} , at a time t_0 to a new position, \mathbf{x}' , at time t . The displacement of a particle which moved from position \mathbf{x} to position \mathbf{x}' is a vector, \mathbf{u} , given by

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{x}' - \mathbf{x} \tag{2.23}$$

Equation (2.23) can be rewritten as

$$\mathbf{x}' = \mathbf{x} + \mathbf{u}(\mathbf{x}, t) \quad (2.24)$$

Whether the deformation is rigid or non-rigid is determined by the functional form of $\mathbf{u}(\mathbf{x}, t)$. In the present case, cardiac motion in the images for the ME stage is estimated by using a GW registration method based on the FFD model with cubic B-splines [54]. These B-splines are made up of control points and are defined using polynomials that have a minimal amount of support in relation to a specific degree, smoothness, and domain partition [59]. B-splines pass through a series of control points and create smooth shapes and surfaces. FFD models that use third-order basis functions are frequently used because they offer a good balance between smoothness and support region [60]. Therefore, they are effective for modeling the elastic deformations of the heart and surrounding tissues [61].

The FFD method is effective in creating models for any type of deformation by embedding an object in a space that can be altered [62]. In 2D, an object can be deformed by adjusting the grid of control points with a fixed resolution. The method uses a parametric model that deforms an object by manipulating a mesh of K control points $\{\mathbf{u}_k \leq k \leq K\}$ governed by a set of transformation parameters $\Theta = \{\Theta_n | 1 \leq n \leq N\}$ with $\Theta_n = \{\theta_{n, \mathbf{u}_k}\}$ [61]. FFDs are used to create N non-rigid deformations, one for each time frame in a dynamic series. These deformations, known as \mathcal{T}_{Θ_n} , map the coordinates of each point in the original template to the corresponding coordinates in a specific frame.

Specifically, we apply a GW paradigm in which no particular frame is selected as a reference, but the reference is built along the optimization process as the average of the transformed images. We need to solve the optimization problem in order to determine the set of transformation parameters Θ that describe the spatial transformations \mathcal{T}_{Θ} . As for the metric to be optimized, we have used the sum of the intensity squared

differences (SSD) with respect to the reference image (see Equation (2.25)); this metric is enlarged with smoothness terms so as to force a realistic motion field solution (see Equation (2.26)). For SSD, the optimum template is known to be the average of the registered images [47]. The SSD is defined as

$$\text{SSD}(\mathcal{T}_{\Theta}(\mathbf{x})) = V(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N \left[\mathbf{m}_n(\mathcal{T}_{\Theta_n}(\mathbf{x})) - \frac{1}{N} \sum_{k=1}^N \mathbf{m}_k(\mathcal{T}_{\Theta_k}(\mathbf{x})) \right]^2 \quad (2.25)$$

where $\mathcal{T}_{\Theta} = \{\mathcal{T}_{\Theta_1}, \dots, \mathcal{T}_{\Theta_N}\}$ is the set of transformations, N is the number of frames in the sequence, \mathbf{m}_n the n -th frame and \mathcal{T}_{Θ_n} is the transformation that maps each material point x in the template onto its corresponding position in the n -th frame. Then, the optimization problem to be solved can be expressed as follows:

$$\hat{\Theta} = \arg \min_{\Theta} \left\{ \sum_{n=1}^N \sum_{x \in \mathcal{X}_{cr}} \left[\mathbf{m}_n(\mathcal{T}_{\Theta_n}(\mathbf{x})) - \frac{1}{N} \sum_{k=1}^N \mathbf{m}_k(\mathcal{T}_{\Theta_k}(\mathbf{x})) \right]^2 + \mathcal{R}(\Theta) \right\} \quad (2.26)$$

with \mathcal{X}_{cr} the cardiac region of interest (ROI) to be registered and $\mathcal{R}(\Theta)$ represents an additional regularization term, which will be defined later, in Section 3.4.

The remainder of this section discusses the concept of 2D control points, for which Appendix A provides support. The control points, represented by the set \mathbf{u} , can be placed anywhere within a region of interest in a pixel grid, represented before by \mathcal{X}_{cr} , or across the entire image. The distance between control points is represented by Δ , which is assumed to be constant for simplicity. The position of each control point is constrained such that $-\left\lfloor \frac{K_l}{2} \right\rfloor \leq u_l \leq \left\lfloor \frac{K_l-1}{2} \right\rfloor$, where K_l is the number of control points and N_l is the number of pixels in the region of interest along the l direction, and must also satisfy the condition $\Delta_l K_l \leq N_l$. The center of the control point grid in the image coordinate space is represented by \mathbf{c} , and the position of a control point \mathbf{u} can be calculated as $\mathbf{p}(\mathbf{u}) = \mathbf{p}_{\mathbf{u}} = \mathbf{c} + \Delta \odot \mathbf{u}$,

using the Hadamard product.

The control point grid's resolution affects the elasticity of the transformation. If Δ is lower, the transformation will be more specific to a small area around the control points. On the other hand, if Δ is higher, the control points will have a larger influence on the overall transformation. The radius of influence of control points is given by $r_l = \frac{(E+1)\Delta_l}{2}$, where l denotes the dimension and E , the order of the B-spline functions, which has been set to 3 in this Thesis. Thus, the local neighbourhood affected by a control point is determined by the interval $[C_l^{inf}, C_l^{sup}]$ with $C_l^{inf} = -\left\lfloor \frac{c_l - x_l + r_l}{\Delta_l} \right\rfloor$ and $C_l^{sup} = \left\lfloor \frac{x_l - c_l + r_l}{\Delta_l} \right\rfloor$. The previous expressions can be rewritten as $C_l^{inf} = \max\left(-\left\lfloor \frac{c_l - x_l + r_l}{\Delta_l} \right\rfloor, -\left\lfloor \frac{K_l}{2} \right\rfloor\right)$ and $C_l^{sup} = \min\left(\left\lfloor \frac{x_l - c_l + r_l}{\Delta_l} \right\rfloor, \left\lfloor \frac{K_l - 1}{2} \right\rfloor\right)$ because points outside the image will not have any impact.

Using the preceding definitions, it is possible to obtain the B-spline dependent transformation as:

$$\mathbf{x}' = \mathbf{x} + \sum_{u_1=C_1^{inf}}^{C_1^{sup}} \sum_{u_2=C_2^{inf}}^{C_2^{sup}} \left[\prod_{l=1}^2 B_E \left(\frac{x_l - p_{u_l}}{\Delta_l} \right) \right] \cdot \boldsymbol{\theta}_u \quad (2.27)$$

where the displacements of the control points are denoted by $\boldsymbol{\theta}_u = (\theta_1, \theta_2)$ and the third order B-spline function with Cox-De Boor recursion formula, defined in [63], is represented by $B_{E=3}$. The deformations at each point are obtained through a 2D tensor product of 1D, as described in [54].

2.5 Registration and Reconstruction Acceleration: Parallelization and DL Techniques

In this section we will focus on the techniques used in this Thesis for the acceleration of the registration and reconstruction algorithms. In particular, we will present the parallelization using GPU programming with the OpenCL language through the OpenCLIPER framework as well as the use of DL through convolutional neural networks (CNN) for the reduction of cardiac cine reconstruction times.

2.5.1 Parallelization using GPU Programming

The use of specialized devices for medical image processing, particularly with the advancement of GPUs towards general-purpose computing devices (GPGPU), has gained significant attention in recent years. However, it is important to note that developing software for GPU devices is not straightforward due to the challenges of parallelization and communication and synchronization issues with the Central Processing Unit (CPU). Additionally, developers may focus on only one device type, leading to increased effort if multiple device types need to be supported later on. There are alternative processing devices available, such as digital signal processor (DSPs), field-programmable gate arrays (FPGAs), and many-core systems like Intel Phi, and OpenCL is a solution for supporting multiple device types, but it may not be as simple or mature as its competitor, NVIDIA Compute Unified Device Architecture (CUDA).

2.5.1.1 Related Work

There has been high activity in the field of MR reconstruction, particularly in regards to implementing algorithms on GPUs. Two survey papers, one focusing on GPU-based medical image reconstruction [64] and the other specifically on MR reconstruction [65], provide a comprehensive overview of the field.

The two most important works in MR reconstruction of which we are aware are the Berkeley Advanced Reconstruction Toolbox (BART) [66] and the framework “Gadgetron” [67]. BART is a programming library and a set of command line tools used for magnetic resonance imaging reconstruction [68]. It offers support for parallel computation using multiple CPUs or, for certain algorithms, GPUs with NVIDIA’s CUDA as the only supported API. Gadgetron, on the other hand, is a framework that enables users to create pipelines for streaming data processing through a series of modules called gadgets. The pipeline is set up using an XML description file and offers parallel computing support for both CPUs and GPUs [69] through CUDA.

As for the FFDs, some GPU-based implementations were described in Ansorge et. al. [70], Modat et al. [71], Ruijters et al. [72], Du et al. [73], Ellingwood et al. [74], Punithaku-mar et al. [75]; however, these proposals are pairwise and device-specific.

Developing algorithms with CUDA technology differs significantly based on whether the final computing device is a CPU or NVIDIA GPU. As a result, both BART and Gadgetron have separate versions of the same algorithm specifically written for either the CPU or GPU. This limitation is inherent to the foundations of CUDA and cannot be easily bypassed. It typically manifests as the requirement for two different data structures to hold the same information, namely, one for host data used in CPU computing and another for device data used in GPU computing.

As mentioned above, programming with a GPU is specific to a particular

device. OpenCL [76] is a standard for parallel computing that can be used with devices from various manufacturers, but the programmer must handle device selection, memory management, and other tasks. An OpenCL program consists of three key components: compute devices, kernels, and data buffers. Compute devices can be any processor like GPU or CPU. Kernels are OpenCL programs that work on blocks of data stored in data buffers, which are usually in the form of N-dimensional arrays. These kernels and data buffers are then sent to the compute devices, where the kernels are replicated several times, corresponding to the data dimensions. Each copy of the kernel, known as a “work item”, is assigned to a processing element, and a group of these work items are executed simultaneously on the compute device’s memory, with each work item handling a specific section of the memory. The host program manages the execution of kernels on compute devices. The host program is typically written in C, but there are also bindings available for other languages such as C++ and Python. The main steps of a host program include (a) obtaining information about the platform and available devices on the computer, (b) selecting the devices to be used in execution, (c) creating an OpenCL context, (d) creating a command queue, (e) creating memory buffer objects, (f) transferring data to memory buffers on the device, (g) creating a program object, (h) loading and compiling the kernel source code or loading a precompiled binary OpenCL program, and (i) creating a kernel object, among others.

The OpenCLIPER framework [77] was created to make it easier to work with a range of devices, including CPU, GPU, DSP, and FPGA, and to simplify the process compared to using pure OpenCL. That is why we propose the use of the OpenCLIPER framework for the acceleration of the registration and reconstruction of cine images.

2.5.2 Deep Learning

DL is a subcategory of machine learning (ML), which belongs to a broad set of techniques encompassed in what is known as artificial intelligence (AI). In simple terms, AI refers to systems or machines that mimic human intelligence to perform tasks and can iteratively improve on the information they gather. While AI is about creating human-like machines, ML teaches machines how to learn from data without explicit help from humans, and DL is about computers learning how to think using structures that are, apparently, very close to those modeled on the human brain. The concept of acquiring more significant representations by learning successive layers is a crucial aspect of DL. The depth of a DL model refers to the number of layers used to build it. As data pass through each layer, the model learns increasingly complex representations. Each layer in the network is responsible for extracting meaningful representations from the input data, which are then used to solve the specific problem at hand. These simple layers are combined to form the model. The model comprises a sequence of layers that refine the input data, with the network learning trainable parameters or weights during training. These weights capture the information that the network has acquired from the training data [78]. Two types of learning techniques will be of interest in this theses, namely, supervised, and self-supervised. Supervised learning involves a set of input-output pairs that allow the learning of a system that maps inputs to the correct outputs. This is in contrast with unsupervised learning only, which has input data with no corresponding labels or classifications, and the goal is to discover patterns within the dataset. Self-supervised learning is a variation or enhancement of unsupervised learning in which the system learns to predict part of its input based on optimizing a cost function that does not require labelled data.

In this Thesis we focus on CNNs. CNNs were developed to incorporate two key concepts: preserving spatial locality in images and learning through

successive levels of abstraction. When a network has only one layer, it can only learn simple patterns. However, with multiple layers, the network can learn multiple patterns. The primary building block of a CNN is the convolutional layer, which includes a set of filters that have parameters that are learned during training. Typically, the filter size is smaller than the size of the input image. Each filter convolves with the image, producing an activation map. The output volume of the convolutional layer is created by stacking the activation maps of each filter along the depth dimension. The convolutional layer's local connectivity requires the network to learn filters that provide the maximum response to a specific region of the input. The initial convolutional layers capture low-level features such as lines, while later layers extract high-level features such as shapes and specific objects [78]. Normally, following a convolutional layer, we will see Batch normalization (BN). BN is a technique used for normalizing the activations between layers. This normalization is performed on mini-batches of the data, as opposed to the entire dataset. The purpose of BN is to accelerate the training process and to enable the use of higher learning rates, making learning easier and more efficient [79]. After BN we will find the rectified linear activation or ReLU activation function, defined by $h = \max(0, a)$. The reason why the ReLU is typically used as the activation function in a CNN is to increase the non-linearity of the dataset. ReLUs have become increasingly popular due to two significant advantages they offer. The first is sparsity, which occurs when the output of the ReLU function is less than or equal to zero. The second advantage is a reduced probability of the vanishing gradient problem, as the gradient has a constant value when the output is greater than zero. This results in faster learning, making ReLUs a preferred choice over other activation functions. Other layers we will use are average pooling and deconvolutions. Average Pooling is a pooling operation that calculates the average value for patches of a feature map, and uses it to create a downsampled (pooled) feature map. It is usually used after a convolutional layer. Deconvolutions, fractionally

strided convolutions or transposed convolutional layers carry out a regular convolution but revert its spatial transformation.

Before training a neural network, three additional steps are required during the compilation stage. Firstly, a loss function must be selected to enable the network to evaluate its performance on the training data, allowing it to adjust its parameters in the correct direction. Secondly, an optimizer must be chosen, which provides the mechanism for the network to update its weights based on the data it receives and the loss function. Finally, metrics are defined to monitor during both training and testing phases to assess the network's performance [78].

Backpropagation is a crucial technique in deep learning that enables a network to learn from its mistakes. It works by identifying and correcting errors as soon as they are detected. Each layer of a neural network has its own set of weights that determine the output values for a given input. These weights are initially assigned randomly. The network is then activated for each input in the training set, and the values are propagated from the input stage to the output stage, where a prediction is made. The primary idea behind backpropagation is to propagate the error back through the network and utilize an appropriate optimization algorithm to adjust the neural network weights with the goal of minimizing the error. This process of forwarding values from input to output and backward propagation of errors is repeated several times until the error is below a predefined threshold or a maximum number of iterations is reached. The model is updated in such a way that the loss function is progressively minimized, enabling the network to gradually adjust its internal weights. When dealing with the depth of a network and the backpropagation method, one problem that needs to be considered is the issue of vanishing gradients. Backpropagation involves propagating a feedback signal from the output loss down to earlier layers. However, if this signal has to travel through multiple layers, it may become weak or even disappear completely, making it impossible to train the network. This is referred to as the vanishing gradient problem.

To address this problem, skip and residual connections introduce a linear information carry track that runs parallel to the main layer stack, aiding in the propagation of gradients through arbitrarily deep stacks of layers.

The introduction of AI into the world of cardiovascular imaging greatly broadens the capabilities of the field [80]. In recent times we have seen AI gaining ground and expanding into numerous fields, improving various technologies already in practice. DL became popular in 2012 when a DL-based technique had an overwhelming victory in the computer vision competition [81]. More so, DL techniques improved their performances and by 2015 they had exceeded human accuracy during large-scale visual recognition challenges [82]. DL learns image data directly while traditional techniques require human intervention for feature extraction [81, 83]. A more general overview of DL can be found in [81–87]. DL techniques have the ability to accurately perform image registration and reconstruction on their own. The advantage of this is that the connection between images based on spatial statistics can be learned during a training phase, and during the testing phase, only one pass through the network is needed. This makes the typically slow processes of registration and reconstruction extremely fast and allows for new real-time applications since it can complete image registration or image reconstruction in one step, without the need of multiple iterations.

2.5.2.1 Related Work

DL techniques such as CNNs have been used for image reconstruction in MRI. Many DL frameworks have been proposed and experimental results have demonstrated high-quality image reconstructions over traditional and compressed sensing-based reconstruction methods. Work has been reported on knee [88–90], brain [88–91] and cardiac [92–96] MRI, using both supervised [88, 89, 91] and self-supervised learning [90, 96, 97]. Occasionally, the network is unrolled to mimic a CS iterative reconstruction problem,

giving rise to a cascade of CNNs [88, 89, 93]. The issue of large amounts of training data required has been solved by reducing the number of trainable parameters [88, 98]. However, several limitations have also been identified such as the existing techniques being computationally expensive [99], some frameworks do not apply to parallel imaging [100], and theoretical analysis is still needed to explain why the algorithms work [98]. An interesting survey can be found in [87].

The field of registration has a long record [101]; the DL paradigm has also entered the field with solutions that provide fast registrations once networks have been trained [102]. On the one hand, supervised solutions such as [103] and [104] rely on segmentations or landmarks to estimate the displacements. On the other hand, there are unsupervised solutions such as [105] and [106] that do not require any ground truth. The VoxelMorph learning framework [107, 108] belongs to the second group and parameterizes deformations using a CNN to implement a PW solution. In the GW arena, we are aware of few attempts that make use of DL technology. In [109], the authors describe a framework specifically designed to address two main challenges in multi-spectral imaging (MSI) registration, namely the multi-modal nature of images obtained from different spectra, and the need of joint registration of sequential images. A large number of images was employed, but it was divided into subsets so that not many were left for validation and testing. The resources with which the system was trained are not modest. In addition, prediction times were somewhat high. In [110, 111], the registration is applied to brain MR scans. While in the first case supervised learning is used, in the second case deforming autoencoders are employed. In [112] the authors propose a single-shot learning GW registration network to register respiratory motion-resolved 3D CT images. None of these alternatives apply GW registration to dynamic cardiac sequences.

We should also mention the work of [113], where the authors propose a novel joint learning approach that combines non-rigid GW registration

and reconstruction of highly undersampled dynamic MRI. The proposed approach utilizes an unrolled deep learning architecture that incorporates a GW registration network and a reconstruction network in each iteration. To train the proposed approach, registration loss and reconstruction loss are formulated for each unrolled iteration. The input to the registration network is the undersampled or intermediate reconstructed images, whereas the registration loss is computed based on the fully sampled ground truth dynamic images. On the other hand, the reconstruction loss is evaluated by comparing the reconstruction of each unrolled iteration with the fully sampled ground truth.

Two methods based on DL are briefly described below. The first one is used for reconstruction, while the second is used for registration. The importance of these methods lies in the fact that they will be the subject of comparison in the development of this Thesis.

MoDL MoDL, which stands for Model Based Deep Learning Architecture for Inverse Problems, combines the power of DL with model-based approaches [88, 114]. It uses a CNN as a denoiser and applies it as a regularizer to solve the optimization problem given by:

$$\mathbf{m}_{k+1} = \arg \min_{\mathbf{m}} \|\mathbf{E}\mathbf{m} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{m} - \mathbf{z}_k\|_2^2 \quad (2.28)$$

$$\mathbf{m}_{k+1} = (\mathbf{E}^H \mathbf{E} + \lambda I)^{-1} (\mathbf{E}^H \mathbf{y} + \lambda \mathbf{z}_k) \quad (2.29)$$

where k denotes the k -th iteration and \mathbf{z}_k is the denoised version of \mathbf{m}_k , obtained through a CNN network. MoDL requires supervised learning to optimize the denoiser network. The data consistency layer is immediate by conjugate gradient blocks, but as the input is \mathbf{z}_k and the output is \mathbf{m}_{k+1} , which, in turn, generates a \mathbf{z}_{k+1} , this requires iterating until convergence. The iterative algorithm is unrolled for a fixed number of iterations, K , in which the weights or parameters to be optimized are shared.

The MoDL method has the zero-filled reconstruction, the coil sensitivities and the subsampling mask as inputs, but it also needs the fully sampled images for training. The loss is defined as the mean square error between \mathbf{m}_K and the desired image \mathbf{t} : $C = \sum_{i=1}^{N_{samples}} \|\mathbf{m}_K(i) - \mathbf{t}(i)\|^2$, where $\mathbf{t}(i)$ is the i -th target image.

VoxelMorph VoxelMorph (VM) is a framework that utilizes a fast learning approach to register medical images in a pairwise manner, allowing for deformable image registration [107]. A CNN is used to set the parameters for the function, and these parameters are fine-tuned using a collection of images. When presented with new scans, VM quickly calculates the deformation field by evaluating the function directly.

There are two different alternatives of VM, namely one of the initial unsupervised learning variants (VM-2, as referred to in [107]) and the probabilistic and diffeomorphic formulation (VMdiff) [108]. The two VM implementations use two well-known metrics as their loss function, namely mean square error (MSE) and negative cross-correlation. The former is defined as:

$$\text{MSE}(\mathcal{J}(\mathbf{x})) = \frac{1}{|\mathcal{X}_{cr}|} \sum_{x \in \mathcal{X}_{cr}} (\mathbf{f}(\mathbf{x}) - \mathbf{m}(\mathcal{J}(\mathbf{x})))^2 \quad (2.30)$$

with $\mathbf{f}(\mathbf{x})$ and $\mathbf{m}(\mathcal{J}(\mathbf{x}))$ being the fixed image or template and the deformed images, respectively. While normalized cross-correlation (NCC) is defined as follows:

$$\text{NCC}(\mathcal{J}(\mathbf{x})) = \sum_{x \in \mathcal{X}_{cr}} \frac{(\sum_{x_i} (\mathbf{f}(x_i) - \bar{\mathbf{f}}(\mathbf{x})))(\mathbf{m}(\mathcal{J}(x_i)) - \bar{\mathbf{m}}(\mathcal{J}(\mathbf{x})))^2}{(\sum_{x_i} (\mathbf{f}(x_i) - \bar{\mathbf{f}}(\mathbf{x}))^2) (\sum_{x_i} (\mathbf{m}(\mathcal{J}(x_i)) - \bar{\mathbf{m}}(\mathcal{J}(\mathbf{x})))^2)} \quad (2.31)$$

$\bar{\mathbf{f}}(\mathbf{x})$ and $\bar{\mathbf{m}}(\mathcal{J}(\mathbf{x}))$ denote local mean intensity images. x_i iterates over a square region around x with a 9-pixel size in each dimension. Notice that the better the alignment the higher NCC; additionally, this function

always provides values between zero and one; consequently, the loss function uses the negative of NCC, hereinafter, NegCC.

It should be noted that, VMdiff is a method for aligning images using probabilistic models. It uses a combination of a neural network and unsupervised learning to speed up the process. It also utilizes a unique approach to diffeomorphic transforms, which involves integrating stationary velocity fields through specific network operations that can be differentiated.

2.6 Conclusion

The preceding sections have provided an overview of recent research in the field of MRI reconstruction and registration, a field that is continuously advancing.

In this Thesis, the focus is on improving the process of reconstructing dynamic images in CMR, with the goal of making the test more comfortable for patients, which can be achieved by reducing the amount of information required for image reconstruction. One way to reduce image acquisition time is to reduce the k-space data, but this can negatively impact image quality. To avoid this, the GWCS method makes use of an optimization approach to reconstruct the dynamic series, where the target functions consist of a data fidelity term and a regularization term. The first term uses the same information as conventional processing, while the second term, a spatio-temporal smoothness function, is calculated on motion-compensated data using GW registration. This increases the sparse nature of the solution, making the signal highly redundant and expressible in a small number of coefficients. However, this process is computationally intensive as it involves iterative steps to estimate motion, which is necessary for the reconstruction.

Chapter 3

DL GROUPWISE REGISTRATION: DGW

This Chapter presents the process of designing a DL architecture for registering cardiac cine following a GW paradigm using affordable devices.

3.1 Introduction

As mentioned in Section 2.3 registration constitutes the bottleneck of the reconstruction algorithm. So, in order to reduce that load, we replace the original registration algorithm used to implement operator \mathcal{T}_{Θ} in Equation (2.21) with a DL-based solution, say \mathcal{T}_{DL} , keeping the rest of the reconstruction algorithm unaltered [48].

In this chapter we focus on providing a DL-based elastic GW registration solution with respect to an optimization-based traditional solution. We propose a much simpler network than that described in [109] (see Section 2.5.2.1 for more insight on references in this paragraph); our network fits a single affordable GPU within an average workstation to solve a GW elastic and dynamic registration problem; in addition, results will show that our solution outperforms the increasingly popular VM [107, 108].

We will use this registration DL-based solution to replace the ME/MC

stage described for GWCS in [13]. In this work we include, for completeness, a reconstruction comparative with both results; we show that computational costs are greatly diminished while keeping an acceptable performance.

We have used two databases of 2D cardiac cine MRI with short axis orientation. The first database consists of multi-slice acquisitions from 89 subjects, both healthy and affected by hypertrophic cardiomyopathy (HCM) (in the proportion 1/3 and 2/3, respectively). This database has been previously used by our team elsewhere [115, 116]. This database will be hereinafter referred to as Database I. The different slices have been extracted from each subject as independent 2D dynamic sequences. The dataset has been separated into two subsets, namely training and validation, containing 609 and 151 two-dimensional sequences, respectively (not all data from the database have been used in this study). Each of the dynamic sequences contains $N = 30$ time frames¹. The images were pre-processed so that they had a 1 mm^2 resolution, and then cropped to 320×320 pixels. Finally, each of the dynamic sequences was intensity-normalized between 0 and 1. The second database was used for validation and testing. It consists of 41 subjects with at least an ischemia episode. In this case, acquisitions were single-slice, in which the slice was relatively centered in the base-apex direction; the number of frames was also equal to 30. Images had different in plane resolutions but, as before, they all were pre-processed so that they had a 1 mm^2 resolution, and then cropped to 320×320 pixels. Images in the sequences were also intensity-normalized between 0 and 1. This database will be hereinafter referred to as Database II.

As stated in Section 2.4, the purpose of image registration is to transform the images so that corresponding structures coincide. In the cardiac reconstruction problem, elastic deformations are applied to the original

¹Notice that the cardiac cycle is divided into N equally spaced cardiac phases so separation between frames is patient-dependent. In addition, arrhythmia episodes, provided they occur during the acquisition, are automatically discarded by the MR equipment.

sequence so that the registered sequence looks static; ideally, the registered frames in the time sequence should all be pixelwise equal, so the total variation term would be null. Following [13], an elastic deformation is achieved by means of FFDs [54], using as a metric the SSD (see Equation (2.25)) between the images in the sequence and the reference (the reference would be hereinafter referred to as template); the GW approach consists in solving a joint problem to find out both the optimum transformations of the images in the time sequence to the template, as well as the calculation of the template itself. This decision is made to avoid the bias that would stem from choosing beforehand one of the images in the sequence as the template [117].

Overall, the registration problem is posed as

$$\min_{\mathcal{T}} \sum_{x \in \mathcal{X}_{cr}} \text{SSD}(\mathcal{T}(\mathbf{x})) + F(\mathcal{T}(\mathbf{x})) \quad (3.1)$$

where $F(\mathcal{T}(\mathbf{x}))$ is a function that imposes some regularity conditions or constraints on the transformations and \mathcal{X}_{cr} is some region in the template domain; this region could include the whole template or could be limited to some parts of the template (for instance, the cardiac area). Equations (2.25) and (3.1) show that registration needs the reconstructed images, which are obtained by minimizing Equation (2.21) which, in turn, requires the registration solution. Therefore, as previously stated, the problem is iterative in nature.

In this work, we solve the problem posed in Equations (2.25) and (3.1) by means of a DL-based approach. The pipeline, hereinafter referred to as **deep groupwise** (dGW), is depicted in Figure 3.1; the solution consists of a CNN that is sequentially fed with each of the images in the time sequence to be registered together with the available template. The network provides a sequence of the deformation fields (i.e., transformations $\mathcal{T}_{DL_n}(\mathbf{x}), \forall x \in \mathcal{X}_{cr}, n = \{1, \dots, N\}$) and it is followed by a spatial transformation module that provides the registered images. The solution proposed is iterative;

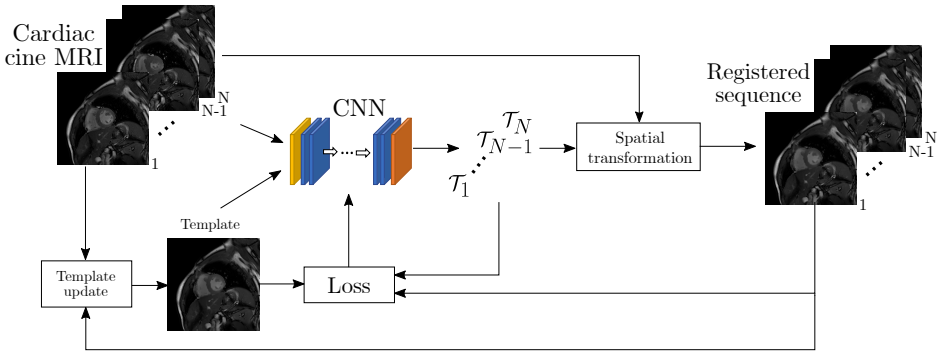


Figure 3.1: Outline of the proposed dGW registration network. A 2D cardiac cine sequence consisting of N frames enters the network, one at a time together with the template, which is calculated as described in Section 3.3. The output of the N executions of the CNN is a sequence of N 2D deformation fields. The original frames, together with their two-channel deformation fields, enter the Spatial transformation block, giving rise to the registered images. This process is applied $L + 1$ times (see Section 3.5), both for training and for prediction.

specifically, we refer to an iteration as the set of actions that begins by feeding the network with the first frame in the data sequence and finishes with the output of the last frame in the sequence. The final solution is obtained after several iterations; this is the first hyperparameter that has been set in the design by means of training and validation. In the first iteration, i.e., when no output is available yet, the template is one of the input frames, which is selected to minimize a geodesic distance criterion; details are provided in Section 3.3. In subsequent iterations, the template is updated by averaging the transformed images from the previous iteration. Let L be the number of template averages performed; the number of iterations will be denoted by $L + 1$. The main ingredients in Figure 3.1 are now described in detail.

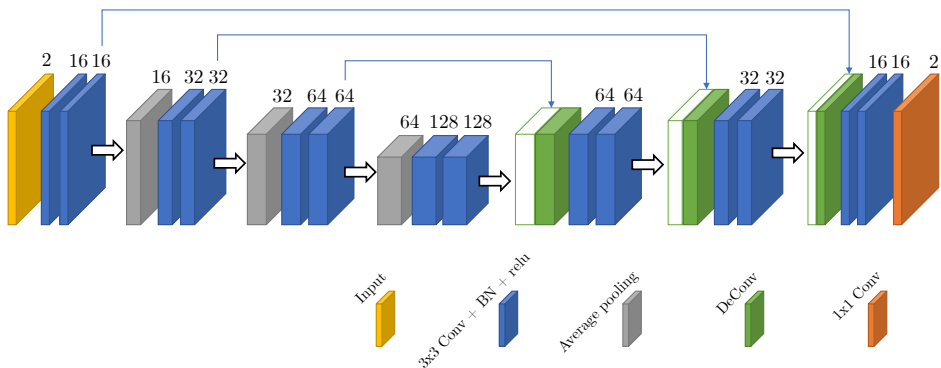


Figure 3.2: Proposed convolutional architecture of the dGW network. The number of channels is shown above each layer. Skip connections involve concatenation of feature maps extracted in the encoding stage with new feature maps from the decoding stage.

3.2 Architecture

We propose a CNN based on the well-known U-net [118], widely used in medical imaging. Figure 3.2 shows the network architecture; the network has some resemblance with the network proposed in [109] although we have reduced substantially the number of filters so that it fits an average GPU. As can be inferred from the figure, both input and output have two channels. With respect to the former, each channel corresponds to a particular image frame and the available template. As for the output, each channel represents the displacement of each material point in one coordinate direction. The figure also shows each constituent block, the details of which are coded both with a color and a legend. Numbers on top of each block in the network indicate the number of channels. Skip connections are also included both to maintain information from previous layers, as well as to avoid the problem of vanishing gradients during backpropagation.

3.3 Template Update

As previously stated, the *Template update* block carries out two different operations, depending on the iteration number. For iterations $l \geq 1$ this module performs a pointwise average of the registered images due to its optimality for SSD [47]. However, any other template construction policy could be accomplished for other types of similarity measures [101]. As for the initial iteration ($l = 0$), we resort to the selection of the template by minimizing the geodesic distance of images to the template on an empirical manifold constructed by means of a k-nearest neighbor (kNN) graph [119]. Specifically, the steps to find the reference using this method are the following [115]:

- (a) Define the similarity between two frames as the residual complexity [119].
- (b) Construct the connected kNN graph of frames based on that similarity, where k is the number of neighbors of each frame. A frame is considered connected to its k-closest frames, the higher k, the more connected the graph.
- (c) Compute the geodesic distance of every node in the graph.
- (d) Select the node, i.e., the frame, with a minimum sum of distances to the rest of the frames. This frame will be taken as the template.

Hereinafter we will refer to this procedure as *automatic template selection*. The template enters the network together with the dynamic sequence, obtaining at its output the deformation fields used by the *Spatial Transformation* block to give rise to the registered sequence. This registered sequence will be used for the calculation of the template in the next iteration.

3.4 Loss Function

The proposed network may be trained by minimizing any derivable loss function; hence, despite in this work we focus on a monomodal application domain, the network lends itself to both monomodal and multimodal registrations by appropriately choosing the similarity function in Equation (3.2). Specifically, in our case the loss function is:

$$\begin{aligned} \mathcal{L}(\mathcal{J}_{DL}(\mathbf{x})) &= \text{Similarity}(\mathcal{J}_{DL}(\mathbf{x})) \\ &+ \text{Regularization}(\mathcal{J}_{DL}(\mathbf{x})) \\ &+ \text{Constraint}(\mathcal{J}_{DL}(\mathbf{x})) \end{aligned} \quad (3.2)$$

As for the similarity metric, we focus on SSD (see Equation (2.25)) for the monomodal case [13]. The network has also been tested with NegCC (see Equation (2.31)).

In addition, since smooth deformation fields are expected, function $F(\cdot)$ in Equation (3.1) comprises some regularization terms in the first and second-order spatial and temporal gradients; specifically:

$$\text{Regularization}(\mathcal{J}_{DL}(\mathbf{x})) = \sum_{x \in \mathcal{X}_{cr}} \sum_{n=1}^N \sum_{p=1}^4 \lambda_p \mathcal{R}_p(\mathcal{J}_{DL_n}(\mathbf{x})) \quad (3.3)$$

with \mathcal{X}_{cr} as defined in Equation (3.1). λ_p are the weights of the regularization terms \mathcal{R}_p , which are defined as:

$$\mathcal{R}_1(\mathcal{J}_{DL}(x)) = \left| \frac{\partial \mathcal{J}_n(x)}{\partial x_1} \right|^2 + \left| \frac{\partial \mathcal{J}_n(x)}{\partial x_2} \right|^2 \quad (3.4)$$

$$\mathcal{R}_2(\mathcal{J}_n(x)) = \left| \frac{\partial^2 \mathcal{J}_n(x)}{\partial x_1^2} \right|^2 + \left| \frac{\partial^2 \mathcal{J}_n(x)}{\partial x_2^2} \right|^2 + 2 \left| \frac{\partial^2 \mathcal{J}_n(x)}{\partial x_1 \partial x_2} \right|^2 \quad (3.5)$$

$$\mathcal{R}_p(\mathcal{J}_{DL_n}(x)) = \left| \frac{\partial^q \mathcal{J}_n(x)}{\partial n^q} \right|^2, \quad q = p - 2, \quad p = \{3, 4\}, \quad (3.6)$$

with $x = (x_1, x_2)$ the two coordinates of material point x . With respect to

the temporal derivatives in Equation (3.6) we have used, in an abuse of notation but for simplicity, the time index n as a continuous variable. All the derivatives are approximated by finite differences.

Function $F(\cdot)$ in Equation (3.1) also comprises the penalty described in [120], intended to constrain the space of solutions. Since the time sequences under study comprise a cardiac cycle due to a synchronized acquisition with the ECG signal, periodicity is assumed, which is enforced by the constraint that the sum of the displacements from a given material point is zero. This can be written as:

$$\text{Constraint}(\mathbf{x}) = \lambda_c \sum_{x \in \mathcal{X}_{cr}} \left(\frac{1}{N} \sum_n (\mathcal{J}_n(\mathbf{x}) - \mathbf{x}) \right)^2 \quad (3.7)$$

3.5 Training

The proposed training scheme is now explained (see Figure 3.3 for reference). Given an epoch, i.e., one passage of the whole training set through the network for parameter update, the order in which dynamic series enter the network is randomly chosen, so that sequences belonging to the same subject do not enter consecutively; then, for each sequence, the steps taken are:

1. Set $l = 0$. The template is calculated as described in Section 3.3.
2. The batch size is taken as the number of frames per slice; therefore, following the definition of iteration previously given, network parameters are updated at the end of each iteration; at this moment, the registered sequence $m(\mathcal{T})_0$ is obtained.
3. Set $l = l + 1$. Update the template as the average of the registered sequence at the previous iteration.

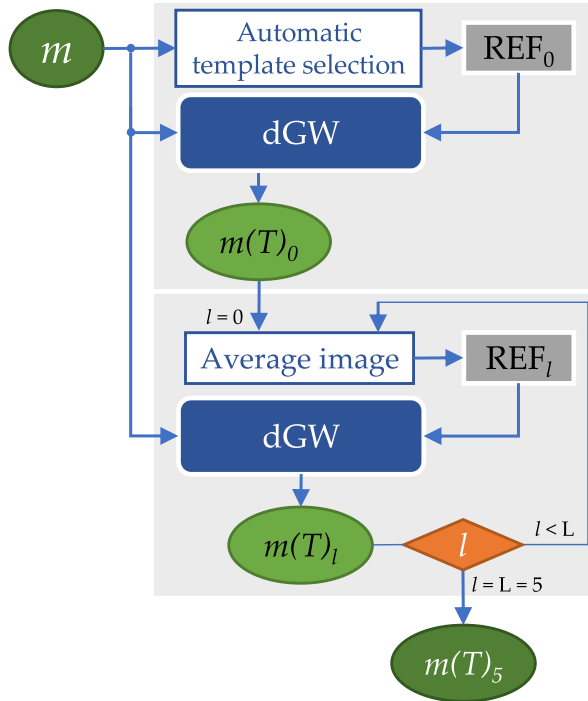


Figure 3.3: dGW registration pipeline. The original 2D cardiac cine MRI sequence m enters both the network and the template selection block (see Section 3.3, $l = 0$). The output is the registered sequence $m(\mathcal{T})_0$, which is used to calculate the template for $l = 1$. This is cycled until $l \leq L = 5$.

4. Steps 2 and 3 are executed while $l \leq L = 5$; consequently, 6 templates are calculated for a given slice. The output at this stage is considered to be the final registration output. The number $L = 5$ is a parameter design that has been set beforehand.

The overall training consists of ten epochs; images employed for testing were those from Database II.

When dGW is used in prediction mode, the steps taken are the same as for training. As is customary, the selection of hyperparameters of the trained networks is carried out using the results on the validation set. Final network performance is calculated on the test set.

3.6 Implementation

This proposed network is implemented in Tensorflow and Keras [121, 122]. The Adam optimizer [123] was used to train the network, with a learning rate of 10^{-4} . We also use `neuron` [124] to carry out spatial transformations. Tensorflow function `add_loss` has been used to accommodate the three terms defined in Equation (3.2), which have been written as different functions.

Each network training for 10 epochs and using a batch size of 30—which corresponds, as previously stated, with the number of frames ($N = 30$)—took 8 hours on one Intel[®] Core[™] i7-4790 CPU @ 3.60GHz with 16GB RAM and one NVIDIA GeForce RTX 2080 Ti GPU. This equipment is deemed as *affordable* since it is quite common in an average research lab. Figure 3.4 shows a graphical representation of the loss function on both the training and the validation datasets from Database I as the number of epochs. Choosing a number of epochs equal to 10 seems an adequate trade-off between performance and runtime.

3.7 Performance Analysis and Hyperparameter Selection

We evaluated registration performance using the structural similarity index measure (SSIM) [125]; this measure incorporates three sources of information to measure similarity between two images, namely luminance, contrast and structure, where the latter is quantified as a measure of cross-correlation. In our problem, because the ideal registration should give rise to a registered dynamic sequence perfectly indistinguishable from the template, SSIM is a suitable measure to evaluate the registration quality. For each dynamic sequence in the validation/test dataset two distant frames are selected to build a sample of SSIM values for a particular algorithm; using

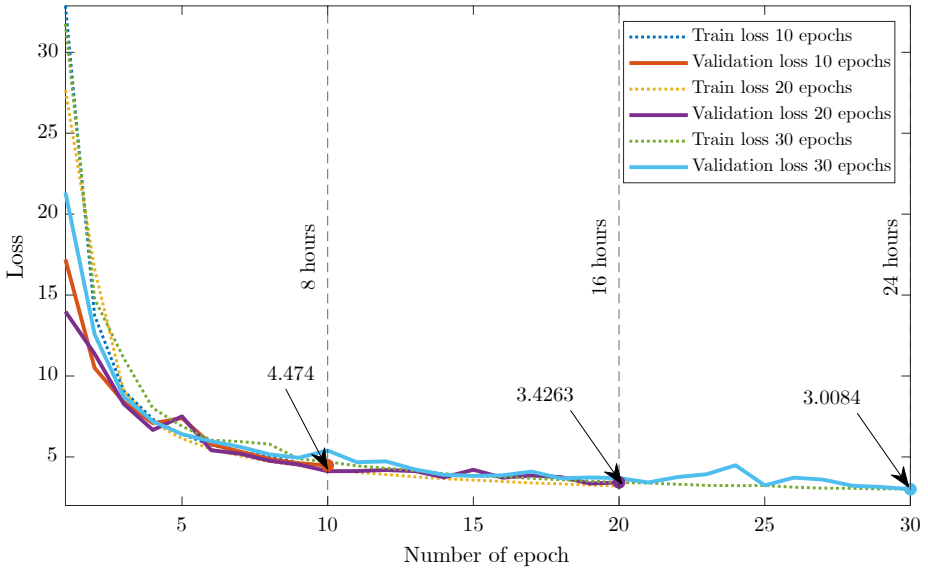


Figure 3.4: Loss evolution during training.

all the frames in the sequence would cause correlation within the sample to grow. These frames correspond to systole and diastole. Comparative performance analysis is carried out on the basis of the sample of SSIM values from each algorithm that enters the comparison; boxplots will be shown, and paired unilateral t-tests p-values [126] will be reported when necessary. Sign-tests have also been calculated to avoid non-normality issues. Unless otherwise stated p-values from t-tests will those reported; sign-test values will only be reported when p-values are in the vicinity of 0.05.

The five hyperparameters to select are those in Equations (3.3) and (3.7). To sample the parameter space, we resorted to a *forward selection* procedure typically used in pattern recognition [127] for feature selection; being exhaustive in our problem would lead to large training and validation times, which we meant to avoid. Specifically, the process consists of several stages:

- **Stage zero:** all the parameters in Equations (3.3) and (3.7) are set

to zero; the network is trained and the SSIM sample is obtained on the validation dataset.

- **Stage one:** we pursue to find the optimum value of each hyperparameter while the others are null. Then, the best among them is selected. For the four parameters in Equation (3.3), we test the values $\{10^{-7}, 5 \times 10^{-7}\}$; for λ_c in Equation (3.7) we tried $\{10^{-7}, \times 10^{-7}\}$; these values were pre-selected on the basis of the validation set in Database I. Then, for each hyperparameter and each value within the pairs just mentioned, we train a network and then a SSIM sample is obtained with each validation dataset. Notice that $2 \times 5 = 10$ networks are trained. SSIM values turned out to be quite close to the all-null solution indicated in Stage 0. Thus, we have obtained the boxplots of the pairwise differences between the SSIM sample with each parameter taking a non-null value and the SSIM sample obtained from the network in Stage 0. Figure 3.5, upper row, shows these boxplots for the validation sets in Database I (left) and Database II (right) for the best selection of each parameter within the pair indicated above (i.e., for the value of this parameter that provides the most favorable result to this parameter); clearly, λ_1 turns out not to be relevant, while the others provide positive-shifted distributions; in the four rightmost boxplots differences turned out to be significant with p-values $< 10^{-4}$. As can be appraised, the highest median corresponds to the activation of $\lambda_3 = 10^{-7}$; this is observed with the two databases. Hence, the result of this stage is the setting of the first order ($p = 3$) temporal derivative defined in Equation (3.6) to the value 10^{-7} .
- **Stage two:** the purpose is to find whether the combination of two non-null hyperparameters (being one of them $\lambda_3 = 10^{-7}$) provides significantly different results than those provided by the network with $\lambda_3 = 10^{-7}$ and the rest of them null. To this end, new networks

are trained with λ_3 fixed while the others can take values within the pairs indicated above. Then, $2 \times 4 = 8$ networks are trained and, for each one, a SSIM sample is obtained with each validation dataset. Should significant differences be found, the second hyperparameter would be that value associated with the minimum p-value. Figure 3.5, lower row, shows the corresponding pairwise differences in this stage for the best selection of each parameter within the pair indicated above for the validation sets in Database I (left) and Database II (right); the figure clearly shows that adding a second parameter does not provide better results than keeping λ_3 activated on its own since differences are negative. Therefore, the forward selection procedure ends at this stage, with the selection of a network with only one hyperparameter activated.

- **Further stages:** in the case that two parameters had been selected, this process would continue by setting a third parameter, with the selected two parameters from stage two remaining fixed, and would continue until all the parameters were set or no significant differences were obtained. Since such differences were not found on stage two, no further stages were tested.

3.8 Experiments

In this section, we provide an overview of the experiments we have conducted; when appropriate, we describe methodological details not directly related to our proposal to make the section self-contained. Unless otherwise stated, all the comparisons will be based on the SSIM distributions of the solutions tested on the test set from Database II.

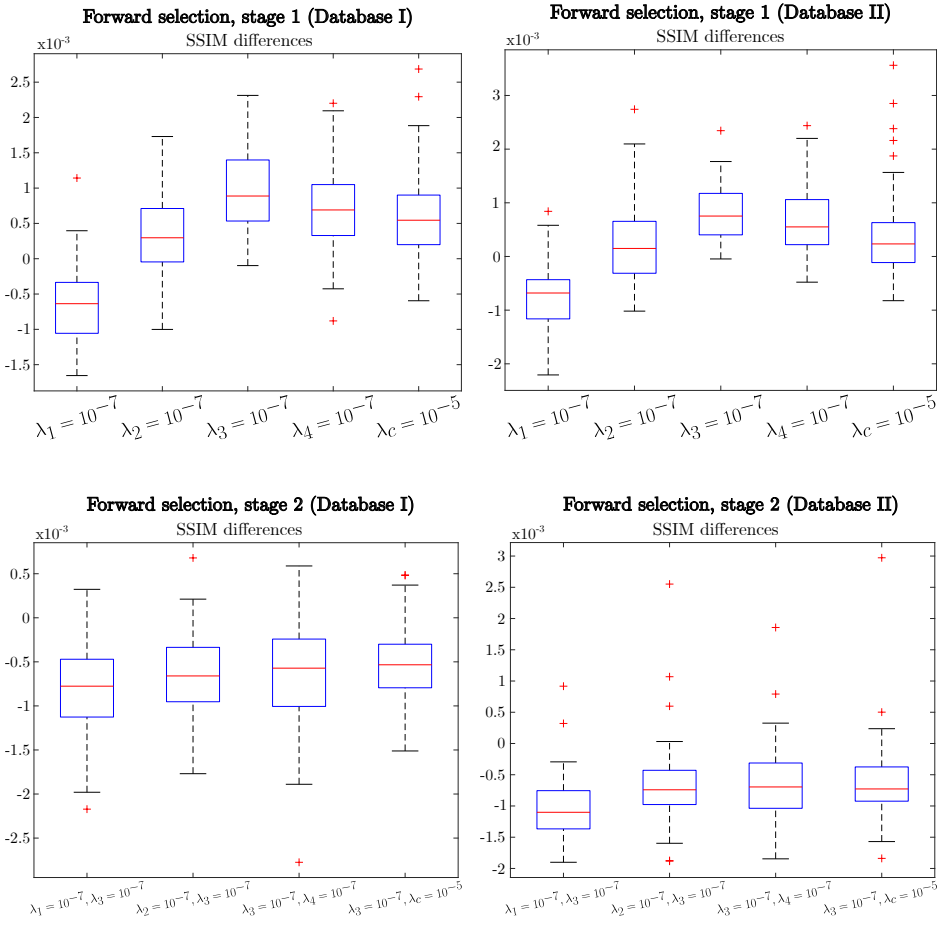


Figure 3.5: Differences between the SSIM samples for the validation sets in Database I (left) and Database II (right). (Upper row) Pairwise differences in Stage 1 with each parameter taking a non-null value and the SSIM sample obtained from the network in Stage 0. (Lower row) Pairwise differences in Stage 2 for the best selection of each parameter within the pair indicated above.

3.8.1 Experiment 1: Iterations, Time Sequence Ordering and an Alternative Similarity Metric

Once the network model has been selected and hyperparameters are set, we meant to test whether our iterative solution i.e., $L = 5$, (recall Section 3.5 and Figure 3.3) was worth taking as opposed to providing a feedforward solution ($L = 0$). To this end, the SSIM distribution of the outputs at $l = 0$ and at $l = L = 5$ have been compared.

For visualization purposes, we also show some images that result from the registration. In this case, we employ the Matlab function `imshowpair` where differences between the images are highlighted by means of pseudocolor; the presence of color means differences between the images with higher intensity for higher differences; green reddish tones can be appraised, depending on which of the two images shows a higher value at a particular pixel. Gray scale is maintained when pixel values are alike. Therefore, the more colored the pixels and higher the intensity in the representation, the higher dissimilarity between the images.

In addition, we also tested whether the order in which the frames enter the network make a difference. To this end, two experiments were conducted. In one of them, the first frame in the sequence was randomly chosen and then the time ordering was followed (when the end of the sequence was reached, we followed its periodic extension, i.e., the N -th frame was followed by frame number 1). The second experiments consisted in random permutation of the frames within the dynamic sequence.

Finally, although we have focused on SSD, we have conducted an experiment to verify that our architecture can adapt to other types of similarity metrics in the loss function. To this end, we have used the NegCC, defined in Equation (2.31); this function is calculated for each image in the sequence, i.e., $1 \leq n \leq N$, and its values are accumulated.

Template selection and batch size remain as described in Sections 3.5 and 3.6 and hyperparameters have all been set to zero. As for the SSD, we have tested both the network with all hyperparameters null as well as the solution we propose as optimal, i.e., that with $\lambda_3 = 10^{-7}$.

As for the importance of having $L = 5$ as opposed to $L = 0$, i.e., an iterative vs. a feedforward solution, a paired t-test was conducted on the outputs obtained from the test data from Database II. The p-value was $< 10^{-7}$, i.e., significant differences existed using $L = 5$ as opposed to $L = 0$. Figure 3.6 shows the evolution of SSD as a function of the number of iterations; we have drawn a vertical line of the number of iterations we set as a design parameter. As can be observed, the SSD value diminishes with the number of iterations but it stabilizes around six. Figure 3.7 shows an example of two registrations with no iterations (left) and with six iterations ($L = 5$, right); the SSD for the case $L = 0$ are **5.65** (upper figure) and **8.67** (lower figure) while for the $L = 5$ case the values are **2.49** (upper figure) and **4.52** (lower figure). Although differences are subtle, one can observe a general higher intensity in the color on the leftmost images; differences have been highlighted by means of arrows.

As for the importance of the ordering of the data frames within the dynamic sequence, the random selection of the first frame in the sequence gave rise to non-significant differences ($p = 0.05$, sign-test $p = 0.0442$). However, the random permutation did give rise to significant differences ($p = 2.05 \cdot 10^{-4}$). Consequently, no differences exist with a random selection of the cardiac cycle start but differences do exist when the time frames are randomly shuffled.

As for using NegCC as opposed to SSD, Figure 3.8 shows boxplots of the solutions; specifically the leftmost figure shows SSIM, from left to right, values of the SSIM distribution for NegCC, SSD with all parameters null and SSD with only λ_3 non-null. Distributions show some degree of overlapping. Pairwise differences are shown in the rightmost figure, where the first boxplot shows the case NegCC - $\lambda_3 = 10^{-7}$ case and the second

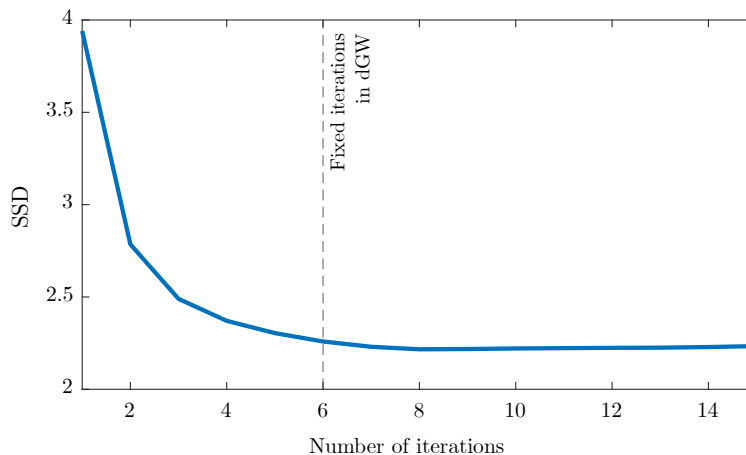


Figure 3.6: SSD obtained in dGW registration according to the number of iterations.

one shows case of NegCC and all-null parameters for SSD. Differences in paired comparisons do exist ($p < 10^{-7}$) for both. Some example images are shown in Figure 3.9.

3.8.2 Experiment 2: Performance Comparison with Another DL Architecture

The performance of our model was compared with two different alternatives of VM, which have been briefly described in Section 2.5.2.1.

VM networks were trained in two different ways as follows: (a) For each slice in the training set, the first image in the sequence is chosen as the template. The network parameters are updated once the remaining frames in the sequence are registered, i.e., batch size equals $N - 1$. Then, the next frame is chosen as the template and the same process is repeated until all of the frames have acted as template; then, we move on to the next slice and continue until the epoch finishes; (b) For each slice in the training set the template is automatically chosen (following the procedure described in Section 3.3 for $l = 0$), so batch size equals N as in dGW

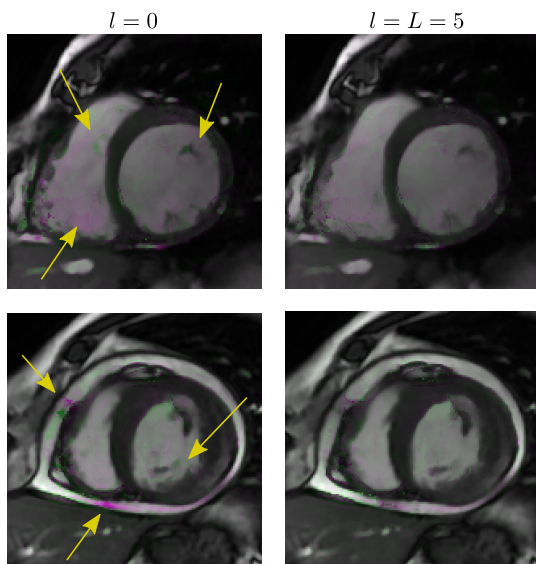


Figure 3.7: Differences between $l = 0$ and $l = L = 5$ in dGW registration quality. We show systole and a diastole superimposed frames for two slices in Database II. Leftmost column: dGW registration with $l = 0$. Rightmost column: dGW registration with $l = L = 5$. Pink: registered systole with higher intensity. Green: registered diastole with higher intensity.

training; then, we move on to the next slice and continue until the epoch finishes. In prediction mode, the network operates as indicated in (b), i.e., the template is automatically chosen and the $N - 1$ remaining frames are registered by means of the network. Comparisons, in this case, are made on the basis of both SSIM as well as on other popular metrics, such as signal to error ratio (SER), Mutual Information (MI) [128] and CC [129]. We also show exemplar images in a pseudocolor composition.

Figure 3.10 shows boxplots that represent the distribution of different values used to measure similarity in the registration. As for this section, the nine rightmost boxplots are of interest; the boxplot labelled as dGW corresponds to our solution. The following four boxplots correspond to the original VM-2; the subindex indicated the type of training applied (see

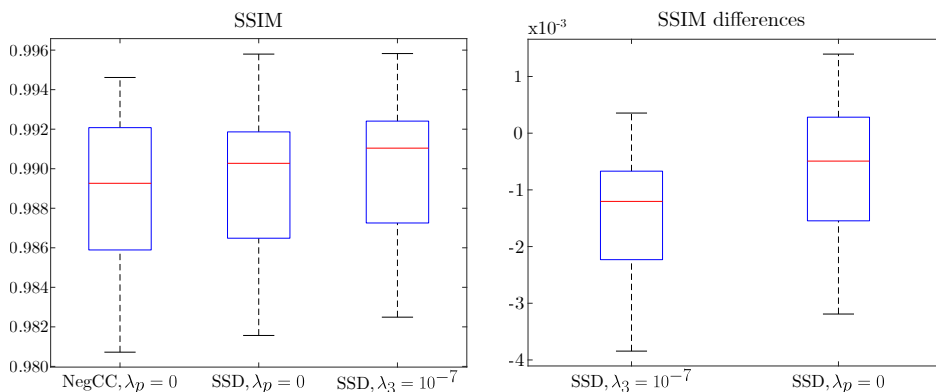


Figure 3.8: (Left) Distribution of SSIM samples in dGW registration, using NegCC as similarity metric, SSD with null hyperparameters and SSD with $\lambda_3 = 10^{-7}$. (Right) Differences between the SSIM samples on the left.

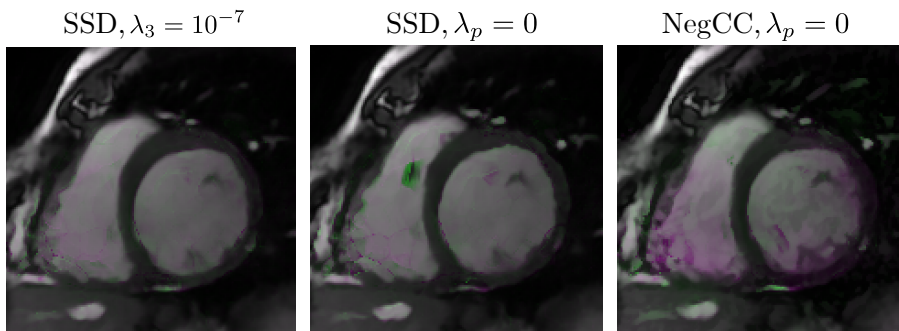


Figure 3.9: Differences between the implementation with NegCC as a similarity metric and two models with SSD in dGW registration quality. We show systole and a diastole superimposed frames for a slice of a patient in Database II. From left to right: dGW registration with SSD and $\lambda_3 = 10^{-7}$, dGW registration with SSD and all hyperparameters null, dGW registration with NegCC and $\lambda_3 = 10^{-7}$. Pink: registered systole with higher intensity. Green: registered diastole with higher intensity.

Section 3.8.2), while the term after the comma indicates the similarity metric (either Equation (2.30) or (2.31)). The last four boxplots show similar results for VMdiff [108]). In this case we show one of the hyper-

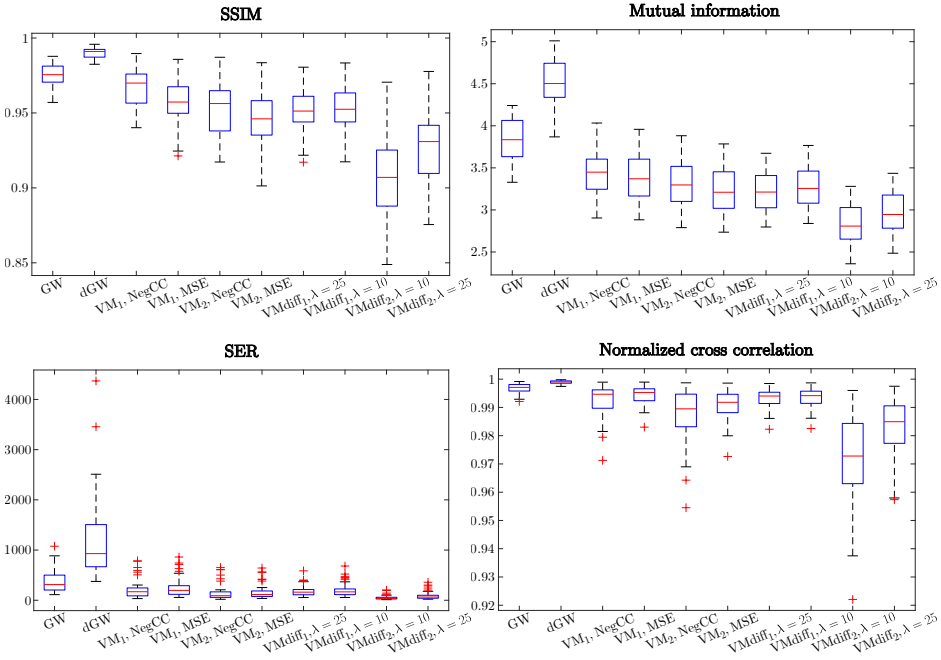


Figure 3.10: Performance comparison between dGW and other alternatives. Four different similarity metrics are used. Upper line: SSIM (left) and MI (right); Lower line: SER (left) and NCC (right).

parameters (λ); since this algorithm requires two, $\lambda = 10$ is accompanied by the parameter $\sigma = 0.02$ while $\lambda = 25$ is accompanied by the parameter $\sigma = 0.01$. As can be inferred from the figure, our algorithm shows a better performance than the VM and VMdiff alternatives. As for SSIM, significant differences exist for all the VM implementations ($p < 10^{-7}$).

As an illustration, Figure 3.11 shows an example of registrations with dGW and the eight VM implementations. As indicated in Section 3.8.2, the presence of color indicates the presence of differences in the images. This example seems to corroborate the general trends shown in Figure 3.10.

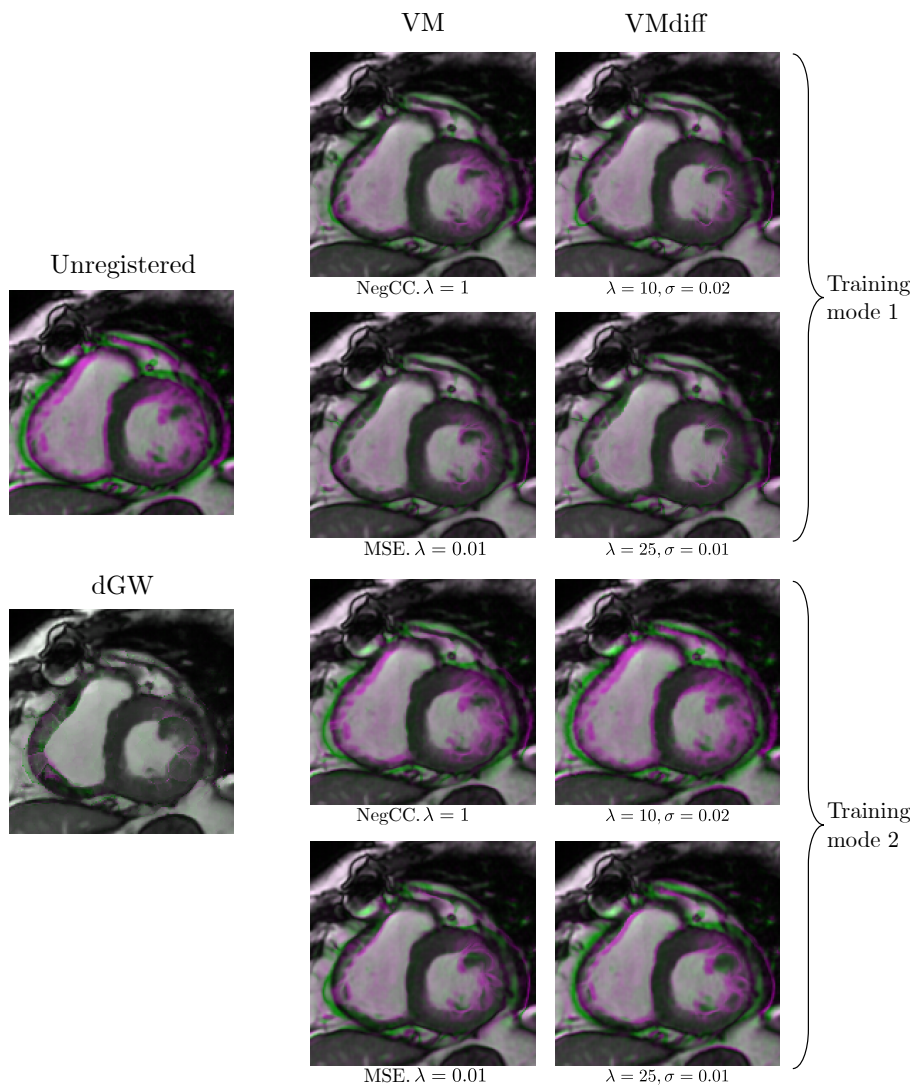


Figure 3.11: Differences between dGW and the eight VM implementation in registration quality. We show systole and a diastole superimposed frames for a slice of a patient in Database II. Leftmost column: unregistered cine cardiac MRI (up), dGW registration (down). Middle column: the four VM-2 implementations. Rightmost column: the four VMdiff implementations. Pink: registered systole with higher intensity. Green: registered diastole with higher intensity.

3.8.3 Experiment 3: dGW vs. an Optimization-Based Registration Approach

Our method dWG has been compared with the optimization-based groupwise registration method described in [13] for ME/MC within an image reconstruction procedure. In this experiment the registration procedure is compared to our DL solution with fully sampled data. In this case, we use the same parameters as those mentioned in Section 3.8.2 as well as the same similar pseudocolor representation of the registered images.

The two rightmost boxplots in Figure 3.10 represent the distribution of the metrics used to measure registration quality for GW and dGW. The times taken in the registration process were 47.13 ± 14.23 for the GW procedure while ours took 5.29 ± 0.06 (both expressed in seconds and with the format average \pm estimated standard deviation).

Figure 3.6 shows the evolution of average SSD with the number of iterations in the algorithm for both GW and dGW and the test set in Database II. We have drawn a vertical line in $L = 5$ to indicate the ordinary number of iterations that we use in dGW. For this experiment, however, we executed a higher number of iterations to check whether this higher number provides additional value; this does not seem to be the case.

As an illustration, Figure 3.12 shows the same example as that in Figure 3.11; the left and right most figures (labelled as “Unregistered” and “dGW”) coincide with the images on the leftmost column in the latter figure. The image in the center is the result for GW registration. Differences in this case are more subtle than for the VM case, accordingly with the boxplots differences in Figure 3.10.

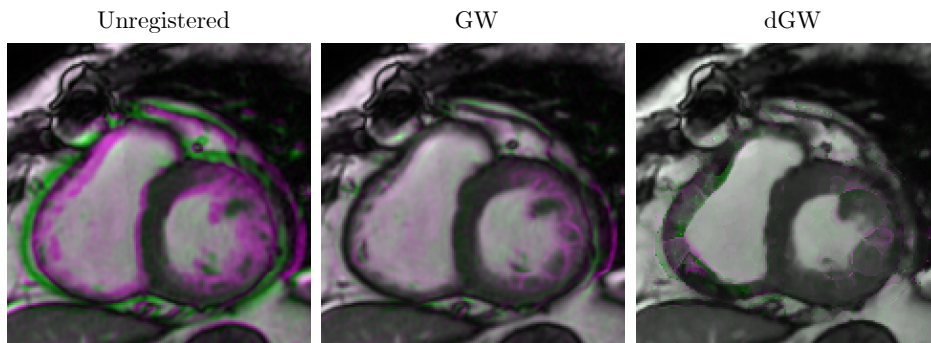


Figure 3.12: Differences between GW and dGW registration. From left to right: unregistered cine cardiac MRI, GW registration, dGW registration. Pink: registered systole with higher intensity. Green: registered diastole with higher intensity.

3.8.4 Experiment 4: Dynamic Image Reconstruction

As a proof of concept, several reconstructions were performed using the GW method published in [13]; this method makes use of an ME/MC procedure using the GW optimization-based registration method mentioned in Section 3.8.3. dGW enters this field as a DL registration alternative to the original optimization-based procedure. The method in [13] is doubly iterative, i.e., it iterates in Equation (2.21) as well as to solve the registration problem, i.e., it also iterates in Equation (3.3). For simplicity, only one iteration in Equation (2.21) has been carried out, both for GW and dGW. The experiments were conducted as follows: for a time sequence, each frame underwent a Fourier transform. Then, a fraction of the horizontal lines in the transform was retained while the rest were set to zero. The ratio between the lines retained and the overall number of lines is known as AF. This operation is modelled by operator E in Equation (2.21). The experiments have been conducted with different AFs.

As a proof of concept, several reconstructions were performed for

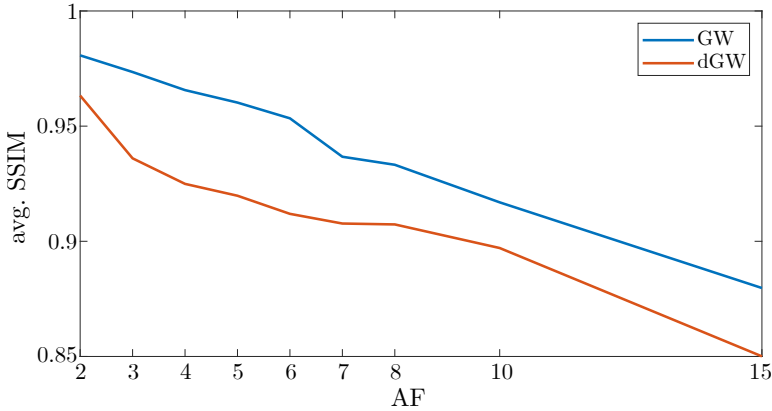


Figure 3.13: Average SSIM obtained, for different AF values, between the ME/MC reconstructions carried out with dGW and GW, with respect to the fully sampled reconstruction.

different AFs, using the dGW method and the GW method published in [13], for which a single iteration of ME/MC is carried out. Figure 3.13 shows the different average SSIM values for the sequence under study, when comparing a fully sampled reconstruction with the ME/MC reconstructions performed with dGW or GW.

Figure 3.14 shows the reconstructed images for $AF=8$. To facilitate visualization, a frame corresponding to systole and another one corresponding to diastole are presented. As stated in 3.1, our reconstruction results seem comparable to those obtained with the original method, while registration times are noticeably reduced. No clear differences between the two types of reconstruction are appraised.

3.9 Discussion

In this chapter we present dGW, a GW registration methodology based on DL techniques that can accommodate both monomodal and multimodal metrics. The registration consists of a network composed of a CNN and a

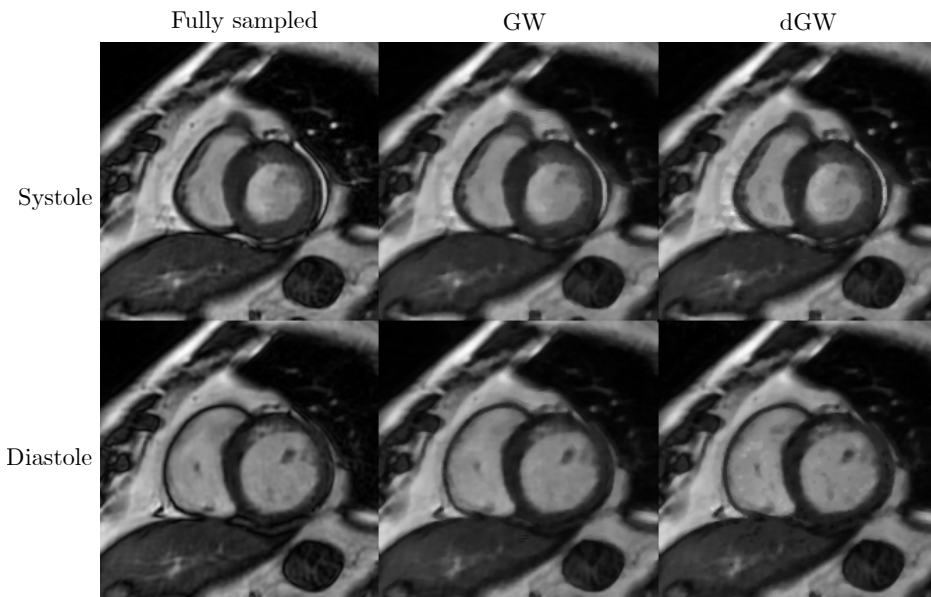


Figure 3.14: Systole and diastole reconstructions with $AF=8$. Upper row: systole; lower row, diastole. Columns: from left to right, fully sampled reconstruction, GW and dGW.

spatial transformation; the network undergoes an unsupervised training for which there is no need of landmarks or any sort of labelling; self similarity within the sequence with respect to the template has been pursued as a design criterion. This self similarity can be defined for either monomodal or multimodal images. Results in Section 3.8.1 reveals that our method can deal with both types of metrics with satisfactory results; SSD turned out to work better than NegCC but the latter was did not go through an hyperparameter optimization procedure, so room for improvement exists.

Our solution is image-based as opposed to patch-based [130, 131]. The latter may need to include an adaptive sampling strategy to assure that a sufficient number of patches is used as well as to select the most representative patches for the entire training set [132]. This overhead is not necessary with our dGW.

We have adopted a methodology based on the forward selection method

to avoid an exhaustive number of hyperparameter combinations. Despite this solution is known to be suboptimal, it is our understanding that this approach is worth taking as a balance between quality and manageability of the required training. We have observed that although most of the hyperparameters (recall Equations (3.1) and (3.7) as well as Figure 3.5) are significant in isolation, when tested in pairs no additional benefits are obtained with respect to using λ_3 as the unique non-null hyperparameter. This may be explained by the fact that the *Spatial transformation* block (recall Figure 3.1) carries out an implicit spatial regularization with the interpolation but no explicit provision is made in Equation (2.25) to foster time continuity in the deformation fields, i.e., to enforce time-smooth deformation fields. In addition, the constraint parameter λ_c (recall Equation (3.7)) does not seem so relevant; the network may be learning some sort of periodicity through the time continuity and the periodicity of the time derivatives through λ_3 (specifically, notice that time derivatives in Equations (3.6) are calculated as “mod N” finite differences, so frame N makes use of the deformation fields at frame 1). Interestingly, as observed in Section 3.8.1, the network is robust to a random selection of the starting point in the cardiac cycle, but performance degrades with a shuffle in cardiac phases within the cardiac cycle. This seems coherent with the fact that shuffled cardiac phases would give rise to discontinuities in the motion fields.

Registration performance in terms of the four metrics shown in Figure 3.10 draws higher figures for our dGW (second leftmost boxplot) with respect to the four implementations of VM (next four boxplots) and the four implementations of VMdiff (four rightmost boxplots). p-values for the eight t-tests provided significant differences as well. We must say, however, that these two approaches are PW and not originally intended for dynamic sequence registration but for an atlas registration, for which the template role would be played by the atlas. This being the case, we have made an effort to adapt the training of these architectures to our

case, as detailed in Section 3.8.2; this adaptation makes the network, in fact, behave as a GW solution since we have used a batch size that comprises all (or most) of the frames in the sequence. No iterations, however, are applied, nor the template is updated. These two shortcomings may explain the lower performance of these networks reflected in Figure 3.10. Unfortunately, no comparisons could be made with [109] because of three reasons; first, the paper, in our opinion, does not include sufficient detail so as to replicate the method. Second, the code is not available. Third, we meant to replicate their network architecture but, when it was launched, our GPU could not give room to the network, so training was prohibitively long. We stress that our approach, on the other side, runs smoothly on an average GPU in the market (recall the specific hardware used from Section 3.6).

dGW was also compared with GW, an optimization-based groupwise registration method used by our team for an MR reconstruction procedure. Figure 3.10 shows better performance from dGW, although differences in an actual image (as shown in Figure 3.12) turned out to not to be so remarkable.

Finally, reconstruction performance has provided comparable results in terms of visual inspection for GW and dGW; it should be noticed, however, that GW should be currently taken as an upper boundary for our performance since the network has not been so far trained with undersampled data. This has been set aside in this paper to focus on the canonical design of the network as a GW registration tool; this is, consequently, a current limitation of our solution. Approaches in which our solution is jointly trained within and end-to-end reconstruction system [88] seem worth exploring. Nevertheless, and as for registration runtimes, results in Section 3.8.3 reveals that dGW has a 9-fold average gain with respect to a traditional optimization-based solution; this is the objective we pursued.

Chapter 4

CINE MR RECONSTRUCTION WITH ME/MC USING OPENCLIPER

In this Chapter, we present our work on developing a device-agnostic and parallel version of GWCS, with a particular focus on the ME/MC procedure. Our objective is to produce high-quality images while minimizing the time required for the reconstruction process, in order to achieve clinically-viable delays.

4.1 Introduction

The dynamic image reconstruction problem is highly parallelizable. This is why we propose its implementation in OpenCLIPER [77], an OpenCL-based framework for medical Image ProcEssing and Reconstruction developed in a previous work, which facilitates the programming of such problems on OpenCL, allowing its execution on any device (CPU, GPU, FPGA...) regardless of the vendor.

Our parallel implementation is coded in OpenCL [76], and all parallel operations are programmed as OpenCL kernels. In practice, high data dimensionality implies that storage of matrices associated with operators is not feasible due to memory requirements. Consequently, operators are

implemented by means of functions and the latter have been the focus of our parallelization effort. In the next two subsections we provide some details on the ME/MC stage and the NESTA algorithm parallelization. Pseudocode is included below, in Algorithm 1.

Algorithm 1 MRI reconstruction with ME/MC

- 1: **Step 0: NESTA multicoil reconstruction**
- 2: **Inputs:** multi-coil k-space subsampled data b , encoding operator $E = \mathcal{A}FS$, sparsifying operator $U = \Phi$, their adjoints and parameters $\lambda, \gamma, L_a \in \mathbb{R}^+$
- 3: **Initialization:** $\mu^{(0)}$, $x_0 = E^H b$, the number of steps $maxIter$ and parameter L_μ
- 4: **for** $t = 0$ to $maxIter$ **do**
- 5: **Step 0.1:** Apply Nesterov's algorithm with $\mu = \mu^{(t)}$
- 6: **for** $k \geq 0$ **do**
- 7: a. Compute the gradient of the ℓ_1 -norm:

$$\nabla f_\mu(Ux_k) = \begin{cases} \frac{1}{\mu} U^H U x_k, & \text{if } |Ux_k| \leq \mu, \\ U^H \text{sgn}(Ux_k), & \text{otherwise,} \end{cases}$$

where function $\text{sgn}()$ applied to vector v means the stack of $\text{sign}(v_i/|v_i|)$, with v_i the i -th vector component.

- 8: b. Compute the gradient of the cost function:

$$\nabla f(x_k) = E^H (Ex_k - b) + \lambda \nabla f_\mu(Ux_k)$$

- 9: c. Compute y_k :

$$y_k = x_k - \frac{1}{\lambda L_\mu + L_a} \nabla f(x_k)$$

- 10: d. compute z_k :

$$z_k = x_0 - \frac{1}{\lambda L_\mu + L_a} \sum_{j \leq k} \alpha_j \nabla f(x_k)$$

where $\alpha_k = 1/2(k+1)$.

11: e. Compute x_{k+1} :

$$x_{k+1} = \tau_k z_k + (1 - \tau_k) y_k.$$

where $\tau_k = 2/(k + 3)$.

12: **Stop** when a given criterion is met

13: **end for**

14: Step 0.2: Decrease the value of $\mu : \mu^{(t+1)} = \gamma \mu^{(t)}$

15: **end for**

16: **Outputs**: reconstructed image $m_0 = x_{k+1}$

17: **for** $i = 1$ to *MotionIters* **do**

18: Step 1: **ME-GW**

19: **Inputs**: reconstructed image m_{i-1}

20: **Initialization**: fix the region of interest, create the control points mesh, compute B-splines products and coefficients, as in [13]

21: **for** $j \geq 0$ **do**

22: Step 1.1: Calculate the pixel-wise displacement fields

23: Step 1.2: Transform images using linear interpolation

24: Step 1.3: Calculate the metric and smoothing terms

25: Step 1.4: Calculate gradients

26: Step 1.5: Update the movements of the control points

27: **Stop** $\frac{1}{KN} \|\theta_{n-1} - \theta_n\| < \epsilon_{\mathcal{J}}$ and $\frac{1}{|\mathcal{X}|} (H_{n-1} - H_n) < \epsilon_H$

28: **end for**

29: **Outputs**: \mathcal{F}_{Θ}

30: Step 2: **NESTA reconstruction with MC**

31: **Inputs**: multi-coil k-space subsampled data b , encoding operator $E = K\mathcal{F}S$, sparsifying operator $U = \Phi\mathcal{F}_{\Theta}$, their adjoints, and parameters $\lambda, \gamma L_a \in \mathbb{R}^+$

32: **Initialization**: $\mu^{(0)}$, $x_0 = m_{i-1}$, the number of steps *maxIter* and parameter L_{μ}

33: *See above (step 0) for further details*

34: **Outputs**: reconstructed image $m_i = x_{k+1}$

35: **end for**

4.2 Parallel Implementation of GW

The registration procedure is determined by the values of the parameters θ_u in Equation (2.27). These parameters are obtained by minimizing the problem defined in Equation (2.26). In this section we will concentrate on the implementation of the most resource-demanding operation, which turns out to be finding the gradient of Equation (2.25). The gradient is, in essence, calculated as follows:

$$\frac{\partial V(\mathbf{x})}{\partial \theta} = \frac{\partial V(\mathbf{x})}{\partial \mathbf{m}} \cdot \frac{\partial \mathbf{m}}{\partial \mathcal{T}_{\Theta}(\mathbf{x})} \cdot \frac{\partial \mathcal{T}_{\Theta}(\mathbf{x})}{\partial \theta} \quad (4.1)$$

where $V(\mathbf{x})$ is the metric, \mathbf{m} represents the image sequence and \mathcal{T}_{Θ} the transformation. Notice that \mathbf{x} will be a grid point (with coordinates given as row and column numbers r , $1 \leq r \leq N_1$, and c , $1 \leq c \leq N_2$, respectively). In addition, each parameter θ is associated to each of the frames (with index n , $1 \leq n \leq N$) as well as to each of the control points (with coordinates, say (r_u, c_u)) and each of the directions of variation, namely, horizontal (index $l = 1$) and vertical ($l = 2$); these indices will be borne in mind for the derivatives.

The first factor in equation Equation (4.1) can be written:

$$\frac{\partial V(x)}{\partial m_{r,c,n}} = \frac{2}{N} \left(m_n(\mathcal{T}_{\Theta_n}(x))_{r,c} - \frac{1}{N_t} \sum_{k=1}^N m_k(\mathcal{T}_{\Theta_k}(x))_{r,c} \right) \quad (4.2)$$

with $m_n(x)$ the n -th frame in sequence m , evaluated at pixel x , and $\mathcal{T}_{\Theta}(\mathbf{x})$ it the transformed position of pixel x in that phase. This equation is computed in parallel for the $N_1 \times N_2 \times N$ pixels.

The second factor in Equation (4.1) is the image gradient with respect to the transformation. The actual operations we perform are the interpolation of the gradient of each frame m_n at position $\mathcal{T}_{\Theta_n}(x)$. The interpolation is carried out within a region of interest, say \mathcal{X}_m , so the number of operations

launched in parallel is $N_{1x_m} \times N_{2x_m} \times N$, where the two first factors are the dimensions of region \mathcal{X}_m .

The last factor in Equation (4.1) is the gradient of the transformation with respect to the variables $\boldsymbol{\theta}_u$ to be optimized; since these variables enter Equation (2.27) as factors, their derivative is straightforward and can be precomputed. Hence, the calculation of the gradient of Equation (4.1) will be performed parallelizing along the rows and columns affected by the B-splines (r_s and c_s), the frame dimension (n), the rows and columns of the control points (r_u and c_u) and the spatial dimension (l), as can be seen below (Equation (4.3)):

$$\frac{\partial V(\mathbf{x})}{\partial \boldsymbol{\theta}_{r_s, c_s, n, r_u, c_u, l}} = \frac{\partial V(\mathbf{x})}{\partial \mathbf{m}_{r_s, c_s, n, l}} \cdot \frac{\partial \mathbf{m}}{\partial \mathcal{J}(\mathbf{x})_{r_s, c_s, n, l}} \cdot \frac{\partial \mathcal{J}(\mathbf{x})}{\partial \boldsymbol{\theta}_{r_s, c_s, r_u, c_u}} \quad (4.3)$$

4.3 Parallel Implementation of NESTA

For the parallelization of NESTA we must pay attention to the operators. Specifically, for the encoding operator E the following parallelizations were made: 1) Multiplication by the coil sensitivities S is performed parallelizing along the spatial dimensions (i.e., $N_1 \times N_2$ operations launched in parallel). 2) The by-frame spatial Fourier transform \mathcal{F} is performed parallelizing along the coils and frames dimensions (i.e., $C \times N_t$ parallel operations) using the cIFFT [133] library. 3) Application of the undersampling mask A is performed parallelizing along the spatial dimensions (i.e., $N_1 \times N_2$ parallel operations). Note that in S , the coils and frames dimensions could have also been parallelized; nevertheless, we empirically verified that this did not constitute a performance gain, presumably caused by the limited number of cores in the hardware. As for the adjoint encoding operator E^H , the following parallelizations were made: 1) Multiplication by coil conjugate sensitivities S^H and, 2) by-frame spatial inverse Fourier transform \mathcal{F}^H . Both of them were parallelized analogously to their counterparts S and \mathcal{F} ,

respectively. 3) Summation of the resulting image sequence in each coil is parallelized along the spatial and temporal dimensions (i.e., $N_1 \times N_2 \times N$ operations launched in parallel).

Regarding the sparsifying operator $U = \Psi = \Phi \mathcal{T}_\Theta$, the temporal total cyclic variation Φ is performed parallelizing along the spatial dimensions (i.e., $N_1 \times N_2$ operations launched in parallel) and the \mathcal{T}_Θ groupwise transformation for MC is performed parallelizing along both the spatial and the frame dimensions (i.e. $N_{1x_m} \times N_{2x_m} \times N_t$ parallel operations). Similarly, the adjoint sparsifying operator U^H is parallelized along the same dimensions as U .

Finally, matrix operations needed in NESTA, such as scaling a vector by a constant or addition of two vectors, are performed efficiently by the cBLAST library [134]. Thus, all the algorithm steps (see Algorithm 1 for details) can be straightforwardly computed by combining operators E and U , their adjoints E^H and U^H , and a few matrix operations.

4.4 Algorithm Implementation on a Generic Parallel Device

We have translated the original method into an actual device-agnostic software solution which (a) finishes in clinically viable times, comparable to other popular reconstruction frameworks, (b) is suitable for execution in parallel devices, and (c) complies with the WORA paradigm (*Write once, run anywhere*) so neither code nor data need be duplicated for CPU/GPU or any other device. As stated in Sections 4.2 and 4.3, we make extensive use of our framework OpenCLIPER to provide this solution.

In this section we provide details on this implementation; Figure 4.1 shows a diagram of our system functional units that will guide us through the description that follows.

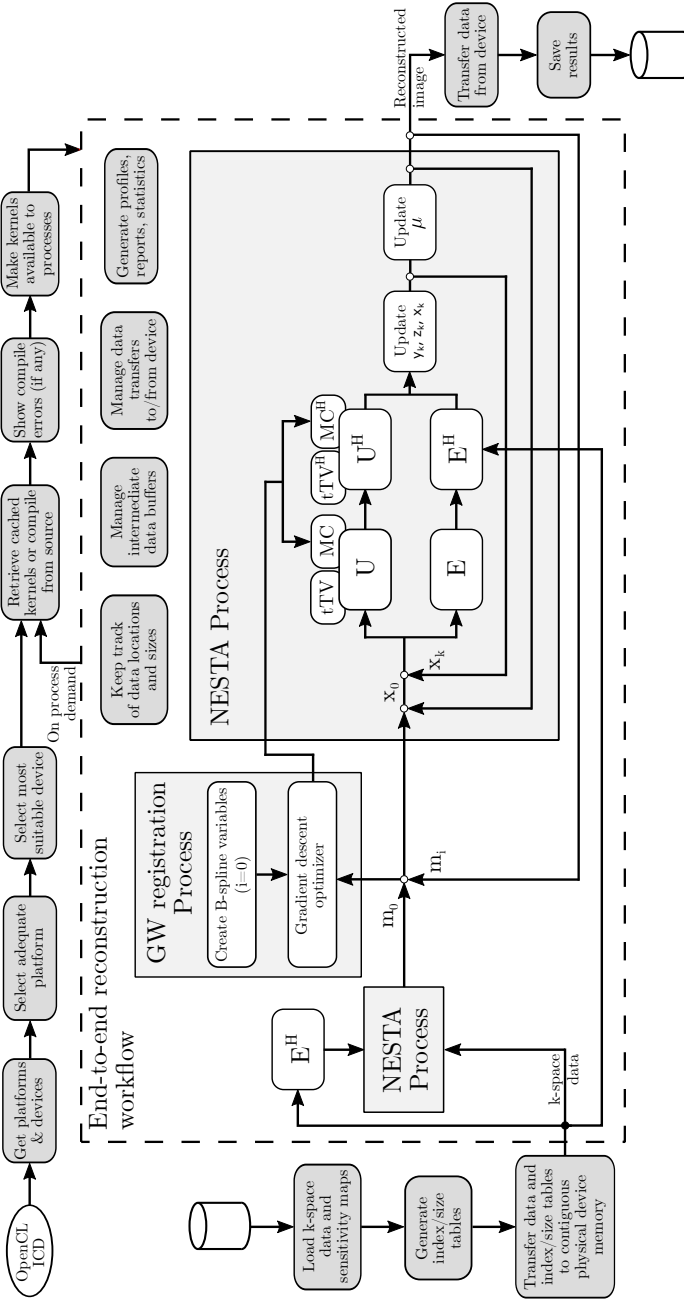


Figure 4.1: Functional units of the proposed reconstruction system. Gray boxes represent necessary albeit unproductive work, which is taken care of by OpenCLIPER. White boxes represent actual, productive work. Each white box is abstracted as a *process*, which may be composed of several other processes. U , U^H , E , E^H are the internal NESTA operators. tTV : temporal Total Variation; GW : Groupwise Registration.

4.4.1 Generalities

Together with the core algorithm —enclosed within the dashed rectangle in Figure 4.1 and its operations represented by rectangles with white background—, several other tasks must be carried out for the system to be functional by itself. These tasks are not only part of the initialization process but they are also active while the actual processing is taking place. All these administrative tasks (shaded boxes in Figure 4.1) are dealt with by OpenCLIPER with minimal programmer intervention. While the system is at work, data pass through several transformations (called *processes* in OpenCLIPER) that need to be connected appropriately.

An appropriate computing device must be selected (top row in Figure 4.1). Since OpenCL supports a wide range of them, an additional workload is the need to detect the available platforms and devices (which may be from different vendors), and to choose the most suitable among them. Moreover, once a device has been selected, kernels must be loaded and compiled for the chosen device. OpenCLIPER simplifies these tasks significantly: kernel loading and compiling is done without user intervention when a process demands it, caching them as necessary so time is dedicated to compilation only once per device (and driver version). Platform/device detection and selection may be either specified by the programmer (via hints such as device type, model, supported CL version, etc.) or, alternatively, left to the framework by just a single line of code. In the latter case—or, in the former, whenever the specified hints match several candidate devices—the *a priori* fastest device is automatically chosen.

All compute-intensive transformations are performed in the computing device, so the system takes full advantage of its parallel capabilities and hence clinically viable execution times may be achieved (see quantification in Section 4.5). The algorithms implemented in OpenCLIPER, which are themselves processes, are made up from several other processes which may be reused at programmer discretion. In this sense, OpenCLIPER provides

a pool of frequently used processes as a ready-to-use toolbox. These tools make the implementation of more involved processes —such as the two described in Section 4.1— easy.

With respect to the four blocks on top of the NESTA Process in Figure 4.1, after the selection of the most appropriate device (either manually or automatically), kernel loading is delayed until a process requires it. At that time, a previously compiled (i.e. cached) version of the kernel is sought. If it does not exist yet, kernel compilation is automatically triggered and the kernel cache is updated (loading of cached kernels is typically two orders of magnitude faster than on-the-fly compilation). Compilation logs (if any) are shown to the user for debugging purposes.

To maximize utilization of the computing device, data objects may be loaded and saved concurrently while the device is busy processing other objects. Since MRI data files are often large, this can save a noticeable amount of processing time per patient.

4.4.2 Data Structures

OpenCLIPER is a software tool that simplifies the burden of keeping track of pointers to data objects and their properties in GPU computing. It does this by encapsulating all data properties within the buffer in the computing device, which makes it transparent to the kernel programmer and eliminates the need for additional kernel arguments. It supports data with an arbitrary number of frames, coils, sensitivity maps, sampling masks and data dimensionality for the special case of MRI data, and arbitrarily complex data for the general case. The framework also provides methods to traverse data buffers along any given dimension. Additionally, OpenCLIPER maps all sub-objects in a data object contiguously in device memory and properly adjusted to hardware alignment, allowing kernel programmers to assume a linear layout when processing compound data objects. Data transfers between host and device are also driven by the

Direct Access Media (DMA) controller to maximize speed. OpenCLIPER has been designed as a development tool for researchers.

4.5 Evaluation

Several reconstructions to test the behaviour of our platform were executed. Reconstructions have been carried out on 7 healthy volunteers courtesy of King’s College London. These data are 2D Cartesian, fully sampled dynamic short axis cine breath-hold ECG-triggered acquisitions in a 1.5 T Philips scanner with a Balanced Steady State Free Precession (bSSFP) sequence. Some relevant parameters of the acquisitions include flip-angle 60° , $TR/TE = 3/1.5$ ms, spatial resolution 2×2 mm², slice thickness 8 mm, 20 cardiac phases, FOV 320×320 mm², 12-14 slices and the number of channels is between 19 and 32, depending on the subject. Both sensitivity maps and k-space data from all coils were provided to us. These datasets were retrospectively subsampled with a Gaussian variable-density random undersampling pattern along the phase encoding direction described in [135] for different values of AF. Hence, for the experiments carried out we have fully sampled images which have been used as a reference for measuring some quality indices (QIs).

First, we have compared our platform with BART [66] to solve Equation (2.21). As for OpenCLIPER we have used NESTA. As for BART¹, since NESTA is not available in that platform, we have used the Alternating Direction Method of Multipliers (ADMM) [136]. As for these comparisons, \mathcal{T}_{Θ} in Equation (2.21) is the identity due to the fact that BART does not incorporate —to the best of our knowledge— a GW ME/MC implementation.

Three experiments have been conducted, namely (a) AF=4 (AF4), (b)

¹Release downloaded on February 25, 2020, from <https://github.com/mrirecon/bart/releases/tag/v0.5.00>

Table 4.1: Mean SSIM values evaluated on undersampled BART reconstructions for a single slice of all patients and the three experiments (AF4, AF4CC, and AF8). Maximal values (bold highlighted) indicate the optimal regularization weight (λ in Equation (2.21)) for the experiments.

Reg. Term	AF4	AF4CC	AF8
$\lambda = 10^{-5}$	0.8776	0.8923	0.8200
$\lambda = 10^{-4}$	0.8832	0.8979	0.8293
$\lambda = 10^{-3}$	0.9037	0.9201	0.8642
$\lambda = 0.01$	0.9056	0.9229	0.8719
$\lambda = 0.1$	0.8955	0.9133	0.8592
$\lambda = 1$	0.8951	0.9128	0.8589

AF=4 plus coil compression (AF4CC) from 19-32 to half of the channels (9-16), and (c) AF=8 (AF8). The parameter λ in Equation (2.21) has been chosen to maximize the SSIM [125] in the BART reconstruction. Table 4.1 shows SSIM values for the three experiments. Thus, $\lambda = 0.01$ has been used for BART and OpenCLIPER reconstructions. For the sake of fairness, since the two optimization algorithms are different, we have chosen the internal NESTA parameters (namely, both μ and the stopping criterion, see lines 5 and 12 in Algorithm 1) to guarantee that SSIM values are comparable for ADMM and NESTA reconstructions; this is graphically shown in Figure 4.2a, with boxplots of SSIM along all the slices of all the patients. Boxplots have been grouped in pairs, i.e., AF4 for OpenCLIPER and BART, and the same structure goes for AF4CC as well as for AF8, as indicated in the horizontal labelling and in the legends on the figures. For completeness, different QIs have been analyzed. Figure 4.2 also includes the corresponding boxplots for the NCC (Figure 4.2b) and SER (Figure 4.2c). Mann Whitney tests [126], show no significant differences in any of these parameters between the compared frameworks.

Figs. 4.3 and 4.4 show two reconstruction examples for two different patients. In both cases, the fully sampled reconstructed images are shown as reference with the images reconstructed using BART and OpenCLIPER

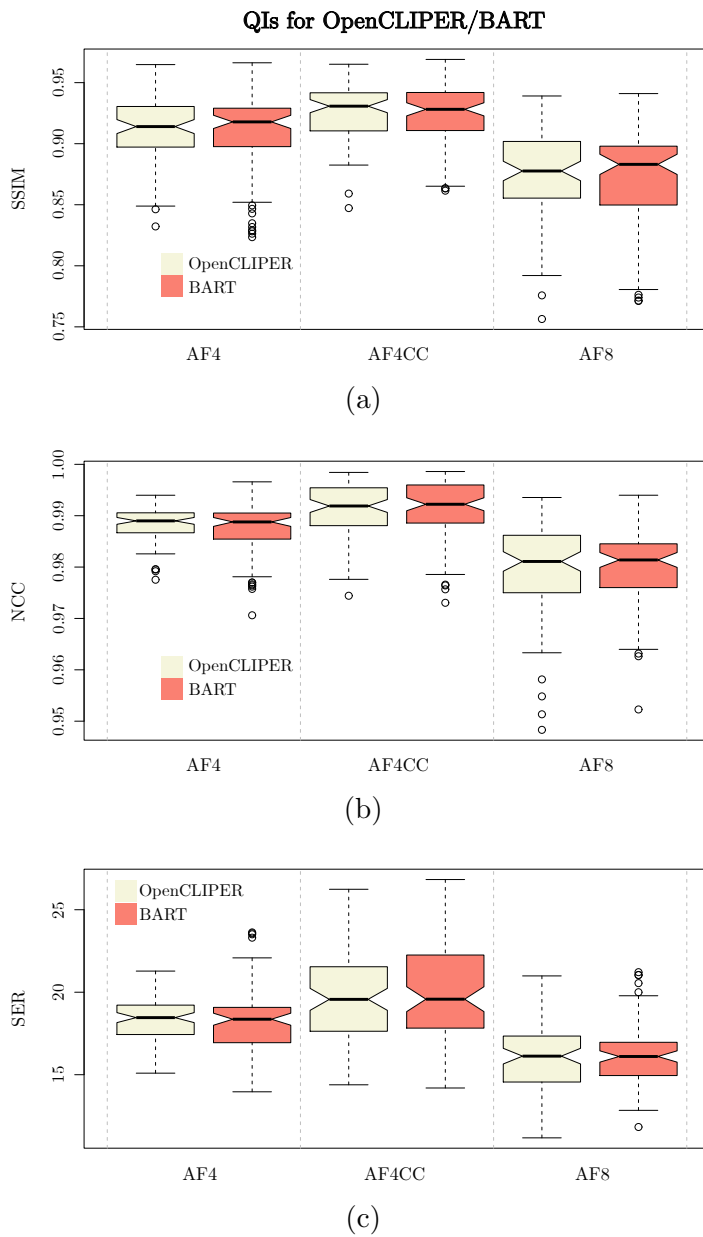


Figure 4.2: Boxplots of SSIM (a), NCC (b), and SER (c) for experiments AF4, AF4CC, and AF8 on both OpenCLIPER, which makes use of NESTA, and BART (which uses ADMM). No significant differences have been found.

for the experiments AF4 and AF8; for the latter, ME/MC reconstruction is also shown for OpenCLIPER. With these parameter setting, execution times in identical computer load situations have been compared. To deal with variability, each experiment has been run one hundred times for each patient.

Four platforms with four different GPUs have been employed. All of them are standard PC-class workstations based on Intel Core or AMD Ryzen processors. The exact GPU models are GeForce 2080Ti and Quadro RTX 6000 from NVIDIA, and Radeon RX 480 and Radeon RX 5700XT from AMD. For reference, some tests have also been carried out using a CPU as the computing device in a 70-thread Intel Xeon server. Notice that BART can only be run on CPU or NVIDIA GPUs, so no further comparison could be done between AMD and NVIDIA as for BART performance.

Figure 4.5 shows boxplots of execution times on the GeForce 2080Ti (a) and RTX6000 (b) for the three experiments, with the same ordering as in Figure 4.2. As for a comparison between NVIDIA and Radeon, Figure 4.6 shows boxplots of the AMD Radeon RX 5700XT (GFX1010) and the NVIDIA Quadro RTX 6000 in the same conditions as in the previous experiment. In terms of performance, the FFT implementation used by the reconstruction algorithm plays a prominent role. BART uses the NVIDIA’s well-known proprietary implementation cuFFT, whereas OpenCLIPER uses the AMD’s open-source implementation clFFT. While the former is highly optimized for NVIDIA GPUs, the latter is conceived to be run on every possible computing device and hence lacks of any specific optimization. This results in clFFT being slower than cuFFT, as shown in table 4.2.

OpenCLIPER compensates this with performance enhancements in other areas, such as parallel loading/saving of data and kernel caching (see section 4.4)².

²In our previous work [77], FFT times reported were overestimated since they incorporated an additional synchronization of the device command queue.

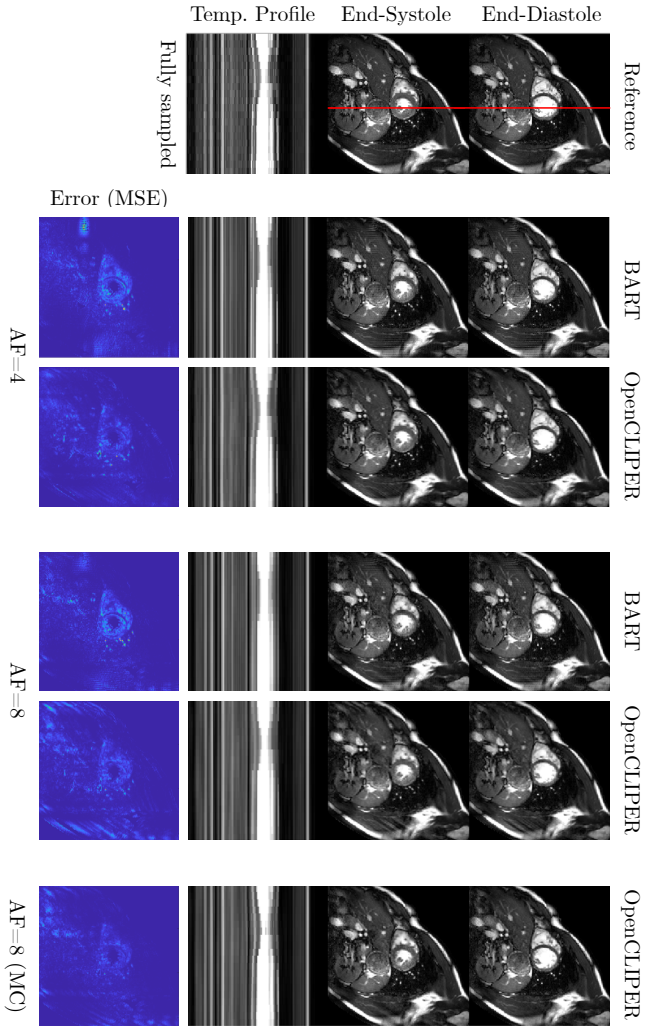


Figure 4.3: Example of reconstructions using BART and OpenCLIPER for the experiments AF4, AF8, and AF8 with ME/MC; the latest only for OpenCLIPER. Fully sampled images are shown as reference. End-systole and end-diastole frames are shown in each case, as well as the intensities along time of the vertical profile marked with red line in reference images. Last row shows the error images (mean squared error, MSE, between reference and reconstructed image).

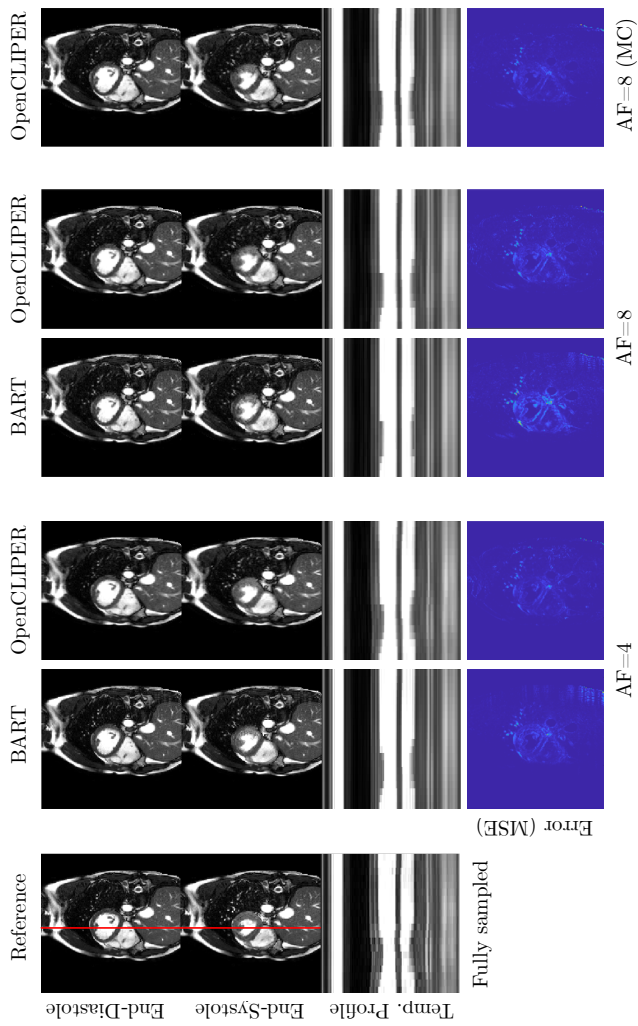


Figure 4.4: Example of reconstructions using BART and OpenCLIPER for the experiments AF4, AF8, and AF8 with ME/MC; the latest only for OpenCLIPER. Fully sampled images are shown as reference. End-systole and end-diastole frames are shown in each case, as well as the intensities along time of the vertical profile marked with red line in reference images. Last row shows the error images (mean squared error, MSE, between reference and reconstructed image).

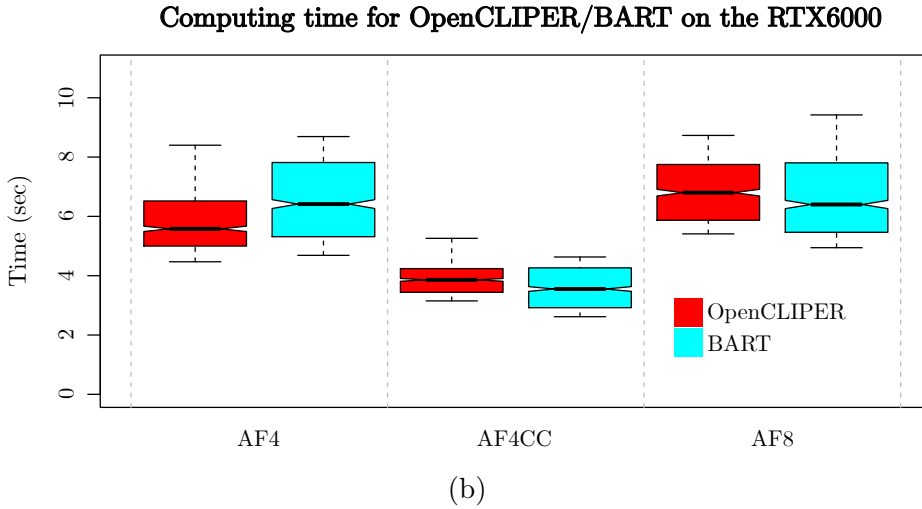
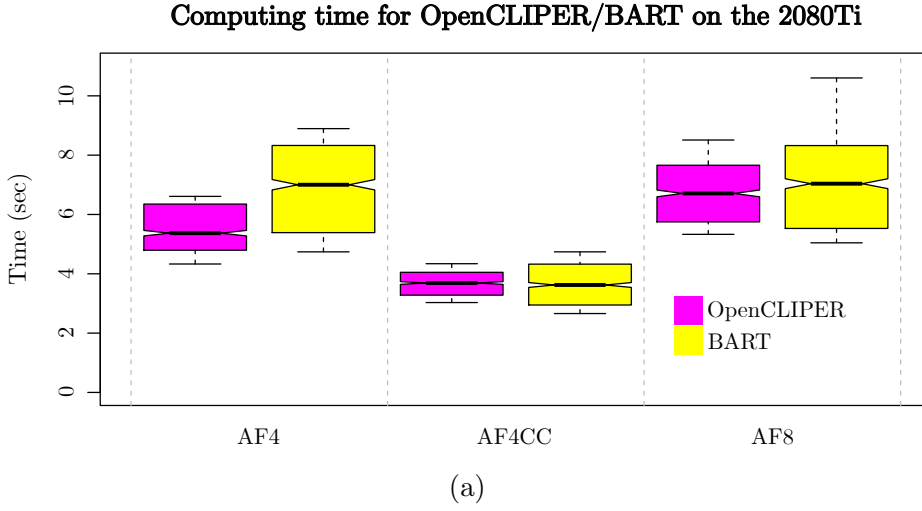


Figure 4.5: Boxplots for experiments AF4, AF4CC, and AF8 on (a) GeForce 2080Ti and (b) RTX6000. Times reported are per whole slice stack.

Figure 4.7 shows boxplots of SSIM for AF8 on OpenCLIPER with and without ME/MC; whereas boxplots for NCC and SER are shown in Figure 4.8. Results of the unilateral Mann-Whitney test for SSIM are significant ($p=0.008$) for ME/MC. If a unilateral signed rank test is run per patient, differences favor ME/MC in six out of the seven patients tested.

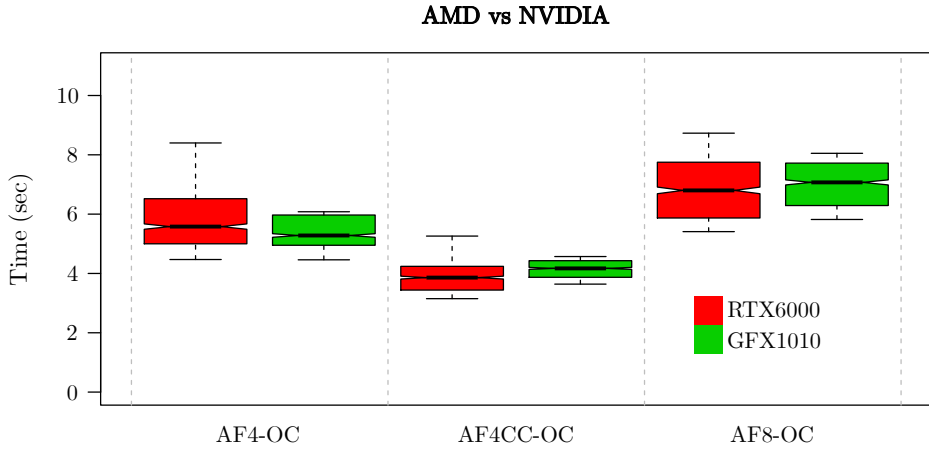


Figure 4.6: Boxplots of computing time for AMD Radeon RX 5700XT (GFX1010) and NVIDIA Quadro RTX 6000 (RTX6000) for the three experiments. Notice that red-shaded boxes coincide with the red-shaded boxes in Figure 4.5. Times reported are per whole slice stack.

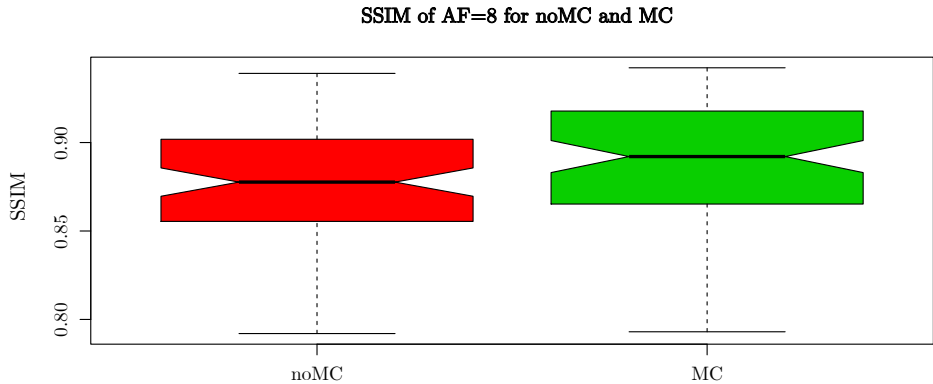


Figure 4.7: Boxplots of SSIM for AF8 on OpenCLIPER with (right, green) and without (left, red) ME/MC.

Mann-Whitney tests are also significant for NCC and SER ($p=0.006$ and $p=0.007$, respectively). The price to pay is the increase in computation time, which reaches a median equal to 62.12 seconds and $(q_1, q_3)=(53, 72.13)$ for 2080Ti, with q_i the i -th quartile. For RTX6000 the median is 60.16

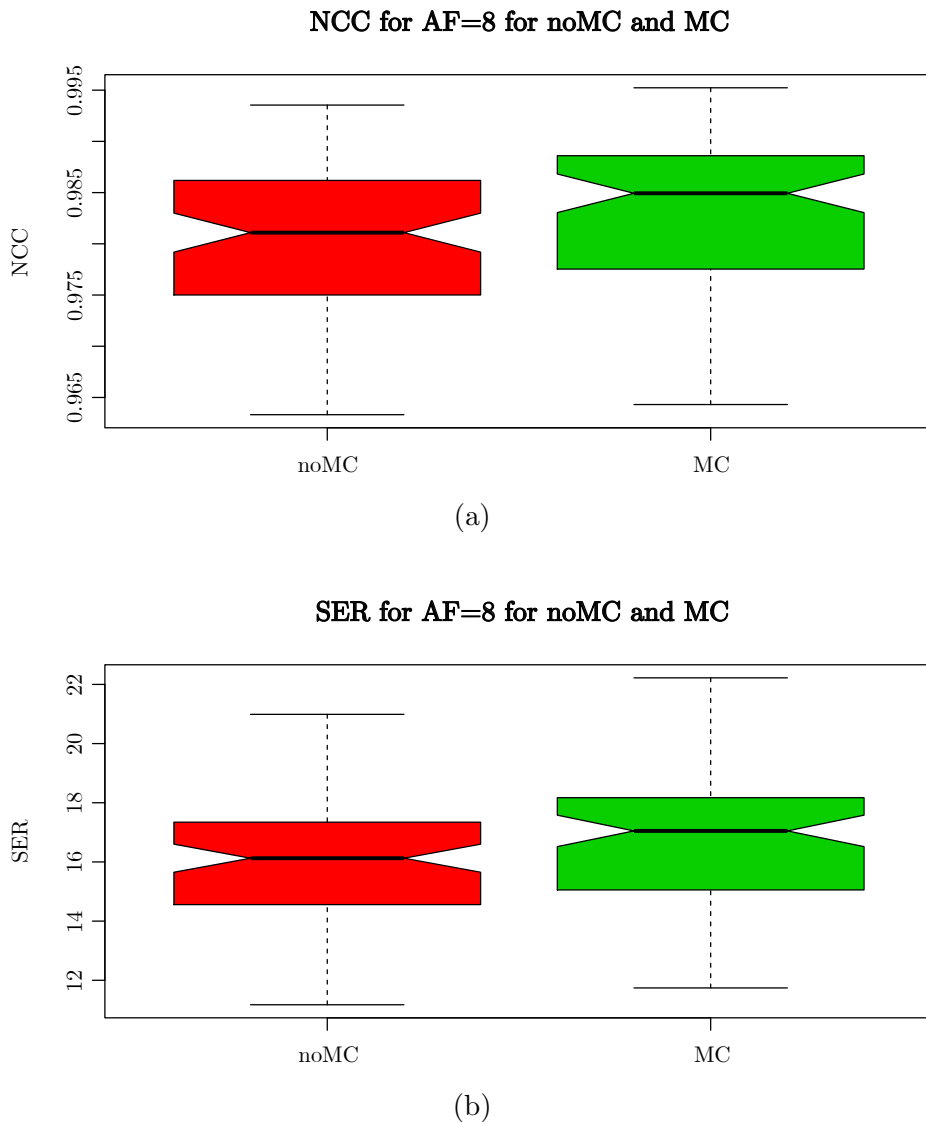


Figure 4.8: Boxplots of NCC (a) and SER (b) for AF8 on OpenCLIPER with (right, green) and without (left, red) ME/MC.

and $(q_1, q_3)=(50.74, 69.24)$ (times reported are per whole slice stack).

Finally, we have also run an experiment on CPU with both BART and OpenCLIPER; the experiment is AF4CC with a randomly selected

Table 4.2: Performance comparison between cuFFT and clFFT libraries on various devices. A single experiment consists in transforming a k-space dataset (160×160 images, 19 coils, 20 frames, complex floats) 1000 times. Each experiment is run 100 times on each library and device. For each combination we show mean execution times (mean) and their standard deviation (std), both in seconds.

Device	cuFFT		clFFT	
	mean	std	mean	std
GeForce 2080Ti	0.6299	0.001	0.6518	0.0227
Quadro RTX 6000	0.5880	0.0016	0.6413	0.0020
Radeon RX 5700XT	N/A	N/A	0.8276	0.0015

patient, ten repetitions. As for the former, the median execution time is 24.58 seconds ($(q_1, q_3) = (24.84, 24, 45)$) while for OpenCLIPER median is 16.33 and $(q_1, q_3) = (16.09, 16.59)$.

4.6 Discussion

OpenCLIPER reveals itself as a device agnostic platform for reconstruction of MR dynamic images. We have shown results for 2D although the code is prepared for higher dimensionality provided that sufficient memory is available. Figure 4.5 shows that our computing times are comparable to those needed by BART; this has been tested in two NVIDIA devices with different memory capacities. Results are not point-by-point comparable since optimization methods used by both approaches are different. However, we meant to be fair by assuring that both platforms gave rise to images with similar qualities; optimization parameters were selected to this end, a goal that seems accomplished according to the evidence shown in Figure 4.2. As for computing times themselves, we observe that our times are fairly similar to those from BART; hence, no obvious losses are appraised by using OpenCLIPER despite some administrative tasks need to be handled, as pointed out in Section 4.4.1, due to its device agnostic character; this overload is represented in Figure 4.1 by the shaded blocks located outside

the dashed box that contains the core of the reconstruction algorithm.

BART cannot be tested in AMD devices; hence only OpenCLIPER enters the comparisons in Figure 4.6. The figure shows that a device one order of magnitude more economical can do a remarkable job. Therefore, OpenCLIPER makes it possible that an affordable device is used for image reconstruction in viable clinical times. When ME/MC enters the algorithm the computing time needed reaches a quantity of about one minute for a multislice reconstruction, a delay that seems also realistic in a clinical setting. Whether ME/MC is worth taking depends on the acceleration factor; for the $AF=8$ case we have employed in our experiments, statistical differences were found.

The execution on CPU revealed that BART needed extra time for image reconstruction with respect to OpenCLIPER. Since no obvious differences were found on the NVIDIA devices, chances are that this is due to the fact that BART, when executed on the CPU, parallelizes only the FFT part of the reconstruction (making use of the threading support in the FFTW library), whereas in OpenCLIPER, every process' kernels are executed in parallel. Additionally, `clFFT` (the OpenCL version of FFT used by OpenCLIPER), has been used in both CPU and GPU experiments since kernel code is unique.

The FFT algorithm is exhaustively used in each reconstruction step. Hence, optimized implementations of this algorithm should have an appreciable impact in the overall reconstruction computing time. Our HIP interface could be an alternative to use other FFT libraries written in CUDA (`cuFFT`, which is known to be highly optimized for NVIDIA devices, might be one of such alternatives if its source code were available). Nevertheless, interfacing overload is non-negligible and this requires further investigation.

Chapter 5

DL CINE RECONSTRUCTION WITH MC: SSMoCOMP

This Chapter focus on designing a self-supervised DL-based solution that replicates GWCS performance at much lower computing times. We also compare these results with those obtained using the device-agnostic implementation described in Chapter 4.

5.1 Introduction

Reduction of MRI acquisition time is a common goal in the MR scientific community. Many reconstruction solutions from undersampled k-space data have been reported and more recently those based on CNN have gained popularity (see 2.5.2.1). In this chapter we present an efficient 2D cardiac MR cine Reconstruction through a **Self-Supervised Motion Compensated** (SSMoComp) architecture. The function we optimize includes MC via elastic GW registration, which is implemented itself by means of the pretrained network dGW described in Chapter 3.

Our proposal uses a DL architecture designed to minimize the objective

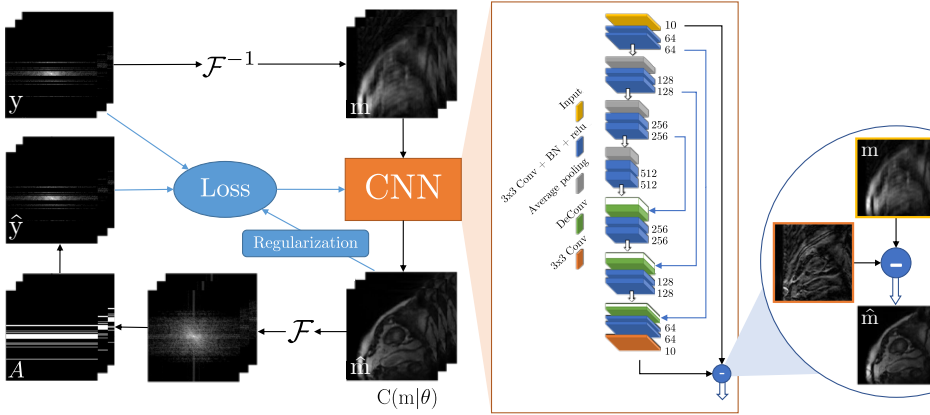


Figure 5.1: Flow chart illustrating the proposed SSMoComp method. Blue lines represent steps that only take place during training. The inputs of the framework are the undersampled k-space data \mathbf{y} and the sampling masks A , resulting in the reconstructed cine images $\hat{\mathbf{m}}$ as output, and $\hat{\mathbf{y}}$ if required.

function in Equation (5.1) below:

$$\hat{\Theta} = \arg \min_{\Theta} \|\mathbf{A}\mathcal{F}\mathbf{C}(\mathbf{m}|\Theta) - \mathbf{y}\|_{\ell_2}^2 + \lambda \|\Phi\mathcal{J}_{DL}(\mathbf{C}(\mathbf{m}|\Theta))\|_{\ell_1} \quad (5.1)$$

where $\mathcal{J}_{DL}(\cdot)$ represents the registration and Φ the tTV operator, $\mathcal{F}(\cdot)$ the Fourier transform, \mathbf{A} the undersampling mask, \mathbf{y} the measurements and $\mathbf{C}(\mathbf{m}|\Theta)$ is the output of the CNN that we train and that is conditional on the parameters Θ that are learned. As for $\mathcal{J}_{DL}(\cdot)$ we have used the result in Chapter 3.

The CNN is based on the well-known U-Net [118], widely used in medical imaging and which has been used during the development of this Thesis. Skip connections are included to maintain information from previous layers, as well as to avoid the problem of vanishing gradients during backpropagation. At the end of the CNN, residual learning has been appended as in [97], subtracting the input of the CNN \mathbf{m} .

The regularization term is calculated by applying \mathcal{J}_{DL} to the estimated reconstructed images, calculating the temporal total variation thereon and

computing the ℓ_1 norm of the result. \mathcal{J}_{DL} can consist of one, two or three iterations of dGW. The load introduced by the regularization term only takes place during the training phase of the network. During training, the CNN weights are updated taking into account this information, relieving the test or prediction phase of this load.

5.2 Evaluation

Training was carried out with Database I. In this case, 10 central frames out of the 30 available were used, and images were resized to 160×160 .

Setting AF=10, experiments were carried out in order to find the best solution. To do so, we sought the λ value giving the best results in terms of peak signal-to-noise ratio (PSNR), normalized root mean square error (NRMSE) and SSIM. A sweep was performed for three different ME/MC stage alternatives using one, two or three iterations of dGW respectively. Training and validation subsets were used to train with Early Stopping. QIs were also obtained with the validation subset in order not to bias the experiments in following section. The training, as well as the experiments, were carried out on our main workstation Intel[®] Core[™] i7-4790 CPU @ 3.60GHz with 16 GB RAM and one NVIDIA GeForce RTX 2080 Ti GPU.

Table 5.1 shows the QIs in terms of PSNR, SSIM and NRMSE, were 0.01 steps have been used for $\lambda \in [0, 0.1]$, and 0.1 steps for $\lambda \in (0.1, 1]$, being λ the weight given to the regularization term, where the registration with dGW takes place. The QI values for the three iteration alternatives of dGW are presented. PSNR, SSIM and NRMSE values obtained from the zero-filled reconstructions and from the SSMoComp reconstructions without any regularization are given for reference.

Results in Table 5.1 show that the best SSMoComp model is the one with $\lambda = 0.2$ and two iterations of dGW. This is the model that will be used in the following experiments.

Table 5.1: Sweep λ and number of iterations of dGW.

	PSNR			SSIM			NRMSE		
Zero-filled	19.72			0.58			0.56		

Number of registrations	1	2	3	1	2	3	1	2	3
λ	23.3			0.64			0,35		
0	24,05	25,05	24,36	0,75	0,81	0,81	0,33	0,29	0,32
0.01	24,45	25,15	14,63	0,81	0,83	0,32	0,31	0,29	0,96
0.02	25,31	25,27	13,97	0,83	0,82	0,23	0,29	0,29	1,05
0.03	25,32	25,19	25,31	0,83	0,83	0,84	0,28	0,29	0,29
0.04	24,98	25,05	24,78	0,8	0,82	0,82	0,3	0,29	0,3
0.05	25,81	25,78	14,99	0,83	0,83	0,36	0,27	0,27	0,93
0.06	25,61	25,87	25,74	0,82	0,84	0,81	0,27	0,27	0,27
0.07	25,71	25,56	25,58	0,83	0,83	0,8	0,27	0,28	0,28
0.08	26,22	26,0	14,21	0,86	0,84	0,23	0,26	0,26	1,01
0.09	25,57	25,8	13,5	0,8	0,82	0,32	0,27	0,27	1,13
0.2	25,83	26,54	12,67	0,83	0,85	0,29	0,27	0,25	1,33
0.3	26,33	25,87	16,18	0,84	0,84	0,39	0,25	0,27	0,8
0.4	26,16	26,1	15,22	0,83	0,82	0,36	0,26	0,26	0,92
0.5	25,03	25,91	26,19	0,77	0,81	0,85	0,3	0,27	0,25
0.6	26,24	26,12	13,6	0,79	0,82	0,3	0,26	0,26	1,16
0.7	25,64	26,24	14,85	0,78	0,82	0,37	0,28	0,26	0,94
0.8	25,38	25,53	15,21	0,76	0,79	0,39	0,28	0,28	0,9
0.9	25,89	24,39	12,63	0,81	0,76	0,28	0,27	0,31	1,31
1.0	24,86	25,43	14,61	0,73	0,8	0,34	0,3	0,28	0,98

5.3 Comparison between SSMoComp and OpenCLIPER

Two proposals have been presented that speed up cine MR reconstructions: the implementation of the reconstruction algorithm with ME/MC in OpenCL with the OpenCLIPER framework and the solution obtained with DL SSMoComp. In this section we compare both proposals. In order to compare the reconstruction carried out with OpenCLIPER with the reconstruction obtained with SSMoComp, a series of experiments were carried out, from which measures of QI and execution times were extracted. For the experiments, we used the previously used databases, namely Database I and Database II (see description in Section 3.1). It should be noted that in order to compare the results obtained with both databases, and given that SSMoComp was trained for $160 \times 160 \times 10$, a spatial resizing is needed in Database II to go from 320×320 to 160×160 . In addition, the 10 central frames of each dynamic cine series are selected.

5.3.1 Prior to Comparison

To ensure fairness in the comparison between OpenCLIPER and SSMoComp, a prior analysis of OpenCLIPER parameters needs to be carried out so that the parameter values enabling reconstruction in both databases are established. For this purpose, three values of the NESTA's stopping criterion tolerance used for problem solving in OpenCLIPER were assessed. Two implementations of the reconstruction algorithm in OpenCLIPER were tested: one without ME/MC and a version with two iterations of ME/MC.

Table 5.2 shows the expected difference between the implementation of the reconstruction algorithm without ME/MC and that with two iterations of ME/MC, which reveals a better QI for the latter. The table shows values

Table 5.2: Image quality metrics for reconstructions using OpenCLIPER without ME/MC and with two iterations of ME/MC for the experiments carried out to define the tolerance of the stopping criterion. PSNR, SSIM, NRMSE and the quality index based on local variance (QILV) values obtained in reconstructions for test subset in Database I and entire Database II from undersampled k-space data with AF=10, with tolerance of the stopping criterion equals to 0.01, 0.005 and 0.001. QIs from zero-filled reconstructions are exposed as reference.

	Database	Zero-filled	OpenCLIPER			OpenCLIPER ME/MC		
			0.01	0.005	0.001	0.01	0.005	0.001
PSNR	I	19.64	24.58	25.37	25.89	24.98	25.52	25.94
	II	15.78	23.33	24.04	24.45	23.70	24.31	24.58
SSIM	I	0.61	0.78	0.80	0.81	0.80	0.81	0.81
	II	0.47	0.74	0.76	0.77	0.76	0.77	0.78
NRMSE	I	0.50	0.29	0.26	0.25	0.28	0.26	0.25
	II	0.76	0.32	0.28	0.27	0.30	0.27	0.27
QILV	I	0.72	0.86	0.88	0.90	0.87	0.88	0.90
	II	0.50	0.86	0.89	0.91	0.88	0.90	0.91

for the QI metrics suggesting that 0.01 and 0.005 tolerances are sufficient and good reconstructions are achieved for both databases. However, Figure 5.2 shows dynamic cine series in Database II that are not well reconstructed. This means that, by setting the stop criterion tolerance to 0.01 or 0.005, the reconstructions work for Database I but not for Database II. Figure 5.2 also shows that with 0.001 tolerance we were able to reconstruct some cine sequences in Database II that were not reconstructed with the other two values. As a conclusion, setting the stopping criterion tolerance to 0.01 works for Database I but does not work for Database II, 0.005 works for Database I but does not work for non-ME/MC reconstructions for Database II and 0.001 works for both databases. Therefore, for the rest of the OpenCLIPER experiments this value was set to 0.001.

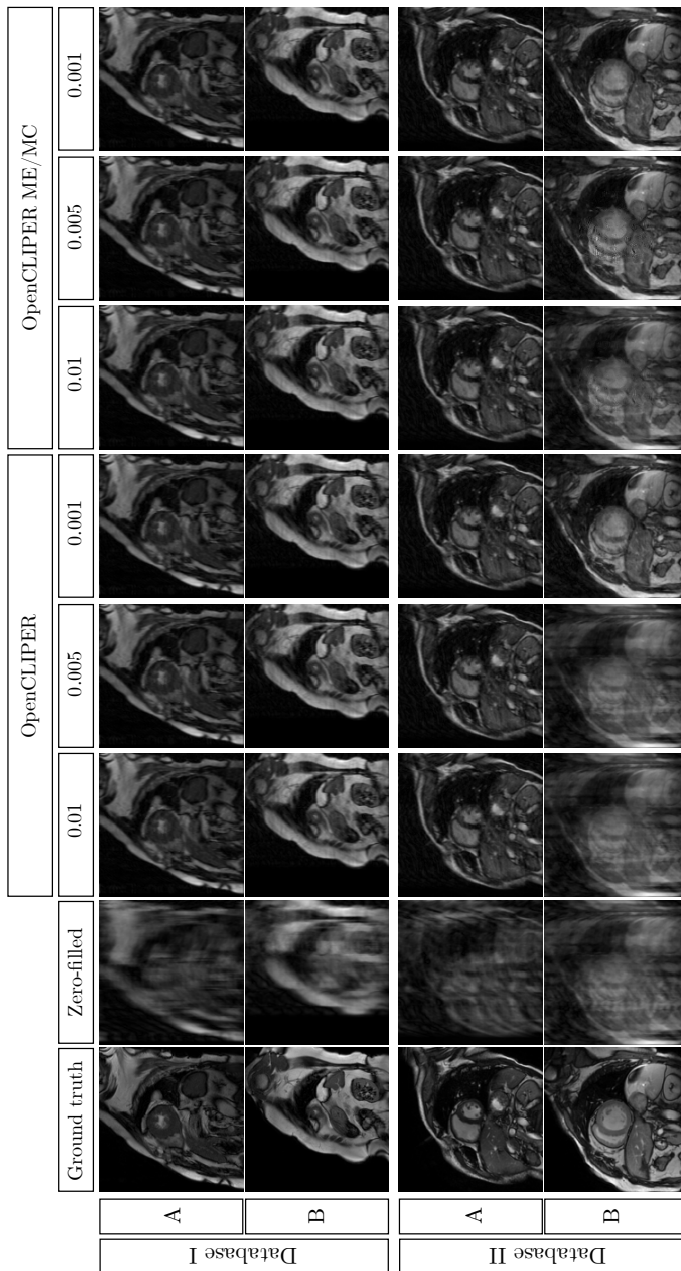


Figure 5.2: Example of reconstructions using OpenCLIPER without ME/MC and with two iterations of ME/MC for the experiments carried out to define the tolerance of the stopping criterion. Reconstructions for two subjects in Database I and Database II from undersampled k-space data with $AF=10$, with tolerance of the stopping criterion equals to 0.01, 0.005 and 0.001, are displayed along with ground truth images and zero-filled reconstructions that serve as a reference.

5.3.2 Experiment 1

Since SSMoComp was trained with Database I, we aim to test its generalization performance with Database II. In addition to SSMoComp, we show in boxplots the results for the test subsets of both databases in runs of the reconstruction algorithm without ME/MC and with two iterations of ME/MC implemented in OpenCLIPER.

Figure 5.3 and Figure 5.4 show boxplots with QIs in terms of PSNR, SSIM, NRMSE obtained in reconstructions carried out with the implementation of the reconstruction algorithm without ME/MC, with two iterations of ME/MC and with SSMoComp, for the retrospectively subsampled data with AF=10 from Database I and Database II, respectively. The results of the zero-filled reconstructions are shown for reference.

Figures show that SSMoComp achieves good reconstruction quality, similar or even better than that from OpenCLIPER, and in fast times when tested on Database I, which was the one it was trained with. However, results are much worse for Database II. The next experiment aims to assess whether generalization can be improved.

5.3.3 Experiment 2

Experiment 1 showed that the network trained with Database I, yielded low performance when tested on Database II, thus showing poor generalization capability. To ensure the network is trained with representative data, we created three randomised folds from Database I and three randomised folds from Database II to fine-tune the previously initialized network. Each of the folds contains three subsets: training, validation and test, with 20, 10 and 11 cine series, respectively. We retrained the network from the best performing combination of parameters assessed with Database I (see Section 5.3.1): $\lambda = 0.2$ and two iterations. As training inputs we combined the created folds from Database I and Database II. So, the final

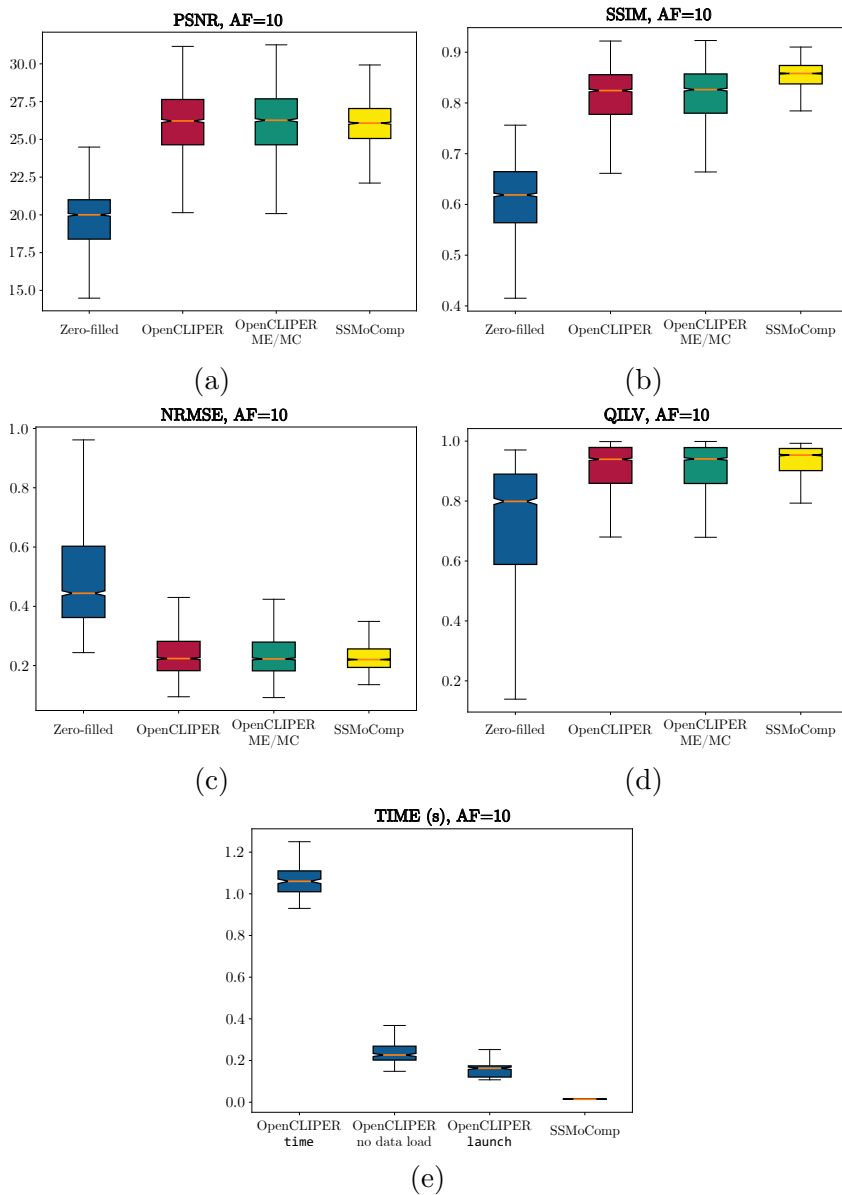


Figure 5.3: Boxplots of PSNR (a), SSIM (b), NRMSE (c), QILV (d) and execution time (e) for AF=10 in Database I on OpenCLIPER with (green) and without (red) ME/MC and SSMoComp (yellow). Results for zero-filled reconstructions are also shown for comparison purposes.

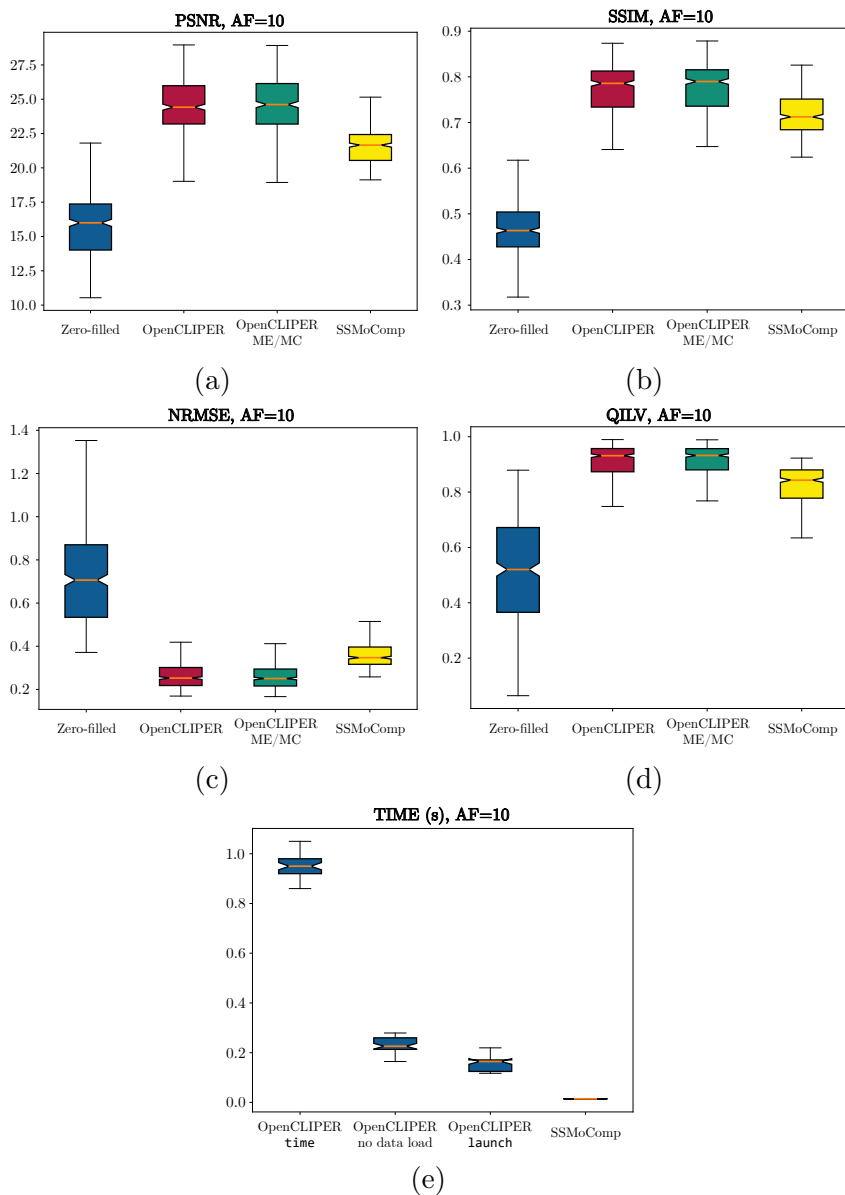


Figure 5.4: Boxplots of PSNR (a), SSIM (b), NRMSE (c), QILV (d) and execution time (e) for AF=10 in Database II on OpenCLIPER with (green) and without (red) ME/MC and SSMoComp (yellow). Results for zero-filled reconstructions are also shown for comparison purposes.

folds included 40, 20 and 22 cine sets for training, validation and test, respectively.

Results are presented in terms of QI with PSNR, SSIM, NRMSE and QILV [137] metrics obtained from retrospectively k-space sub-sampled data with AF=10 with the zero filled reconstructions, OpenCLIPER without ME/MC, OpenCLIPER with two iterations of ME/MC, SSMoComp trained with Database I and SSMoComp tuned with the combination of Database I and II.

Figure 5.5 shows the results obtained on the test fold from Database I. Figure 5.6 shows the equivalent on Database II test fold. Finally, Figure 5.7 shows the aggregated results for both databases.

5.4 Discussion

The optimum SSMoComp configuration, as determined by the highest PSNR, SSIM, and NRMSE scores, was achieved with a weight of $\lambda = 0.2$ and two iterations of dGW.

This solution is similar to the approach proposed by the authors of [113], however, SSMoComp does not require fully sampled data to act as ground truth, since we use self-supervised learning for reconstruction and unsupervised for registration, although it is done offline. This is an important point to bear in mind, as fully sampled images are not always available.

The QI metrics in Database I, (the one used to initially train the network) show good performance. The boxplots reveal that the median values for SSMoComp are generally favorable, with a smaller interquartile range. In contrast, the median values for Database II, which was not part of the initial training, show a different behaviour.

As for execution times, SSMoComp clearly overcomes OpenCLIPER. The time difference between the different SSMoComp experiments may

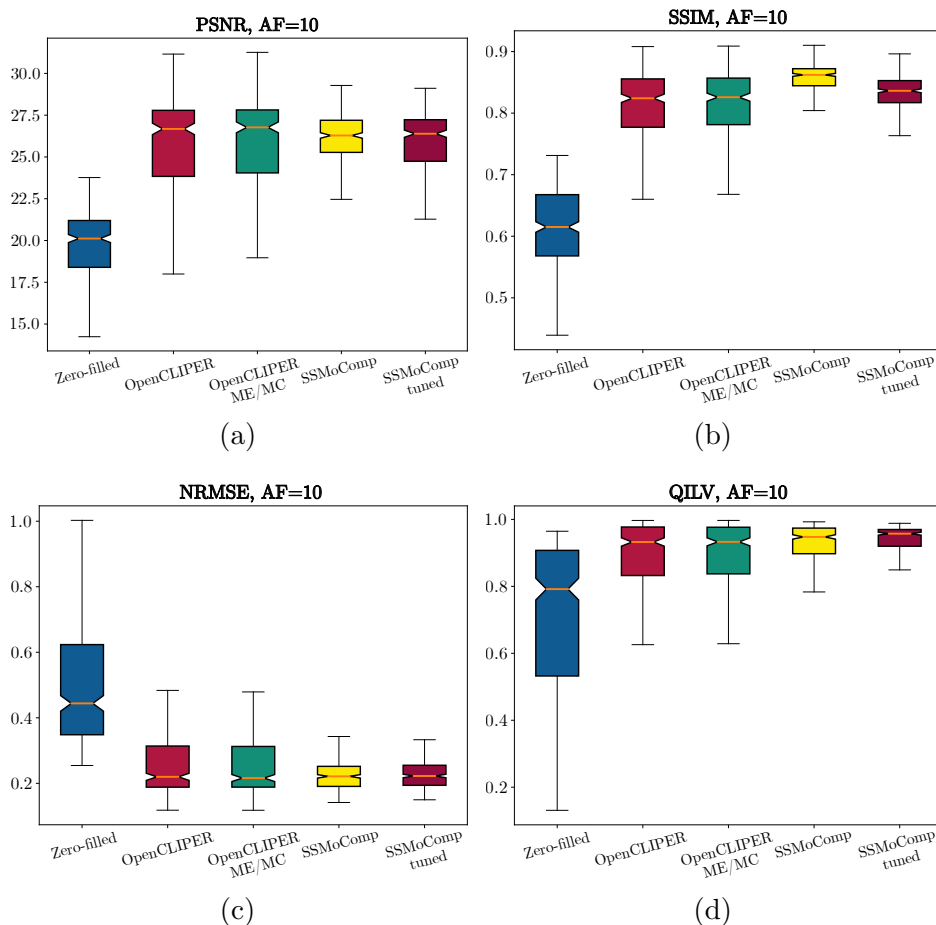


Figure 5.5: Boxplots of PSNR (a), SSIM (b), NRMSE (c) and QILV (d) for AF=10 in Database I random folds on OpenCLIPER with (green) and without (red) ME/MC and SSMoComp before (yellow) and after (maroon). Results for zero-filled (blue) reconstructions are also shown for comparison purposes.

be due to the first run being much slower. However, it only takes 0.015 seconds to get a prediction for a $160 \times 160 \times 10$ sequence.

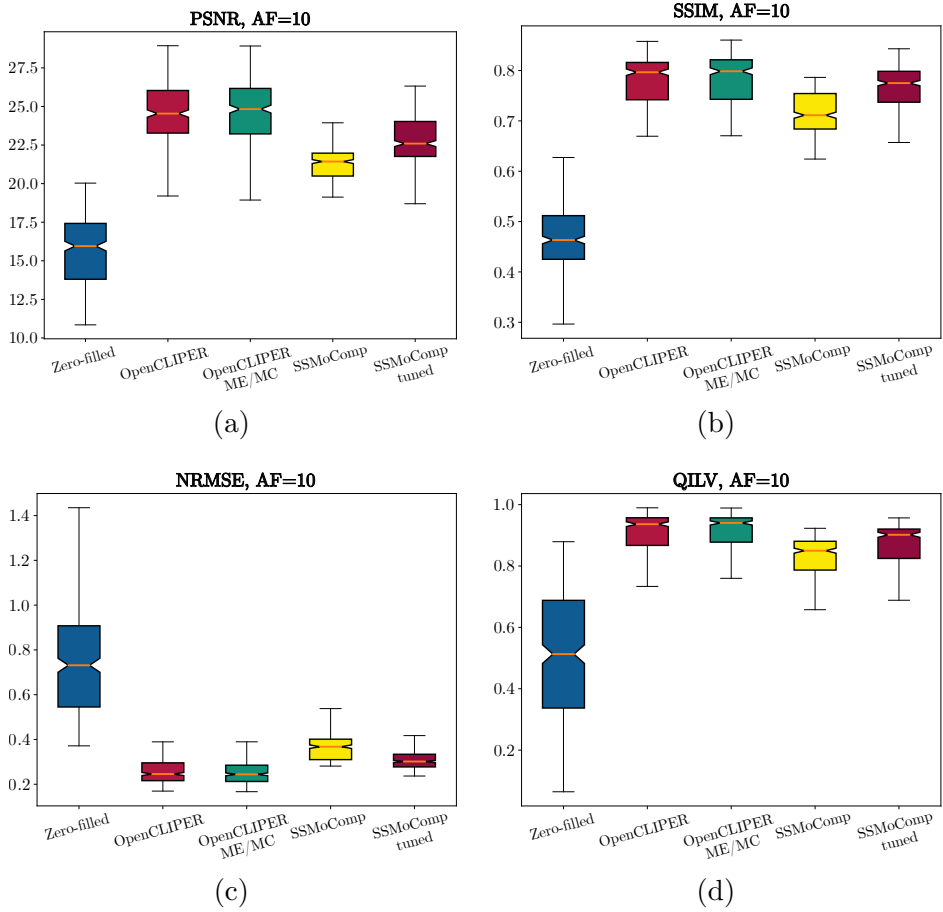


Figure 5.6: Boxplots of PSNR (a), SSIM (b), NRMSE (c) and QILV (d) for AF=10 in Database II random folds on OpenCLIPER with (green) and without (red) ME/MC and SSMoComp before (yellow) and after (maroon). Results for zero-filled (blue) reconstructions are also shown for comparison purposes.

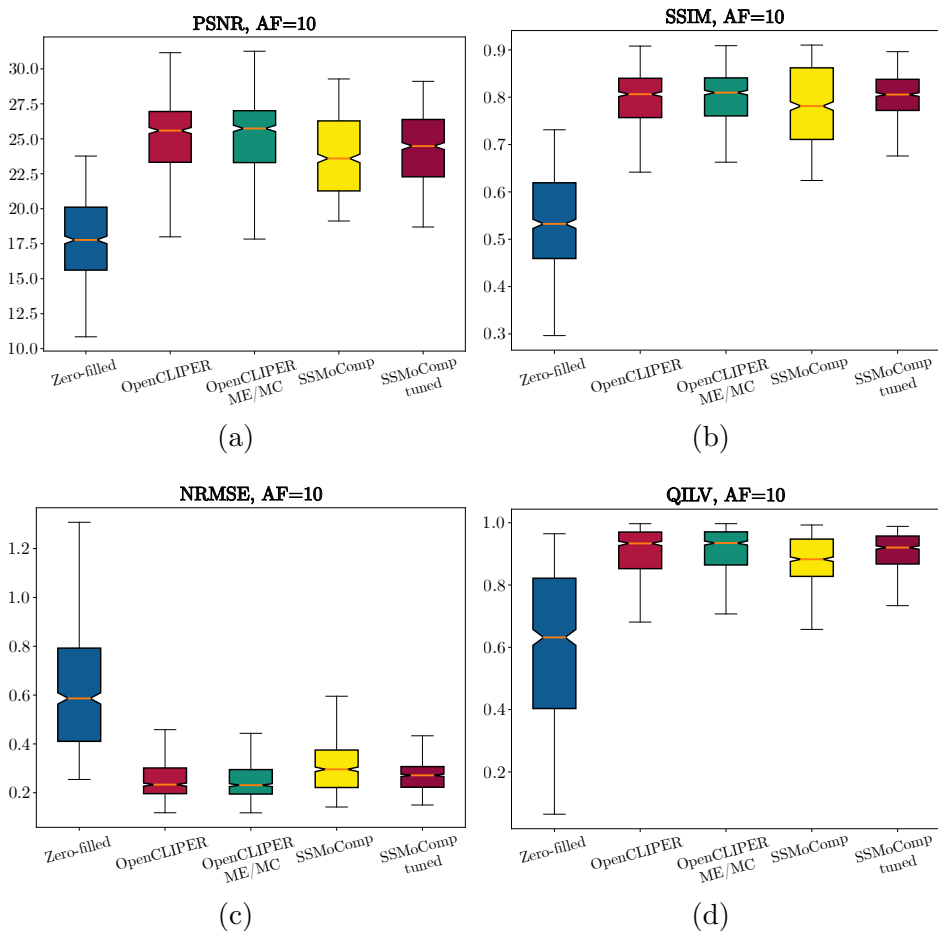


Figure 5.7: Boxplots of PSNR (a), SSIM (b), NRMSE (c) and QILV (d) for AF=10 in Database I and Database II random folds on OpenCLIPER with (green) and without (red) ME/MC and SSMoComp before (yellow) and after (maroon). Results for zero-filled (blue) reconstructions are also shown for comparison purposes.

Chapter 6

DL PERFUSION RECONSTRUCTION: SECRET

This Chapter discusses the adaptation of the self-supervised kernel presented in Chapter 5 to cater to the requirements of FPP-CMR reconstruction.

6.1 Introduction

FPP-CMR time frames must be acquired in real-time to capture the rapid passage of compromised tissues. Thus, undersampled reconstruction methods have been proposed to accelerate FPP-CMR acquisitions as a means to improve spatial resolution and heart coverage [138–140]. However, these methods can lead to long reconstruction times. This chapter presents the work aimed at speeding up reconstructions and obtain the contrast-enhanced dynamic image series from undersampled FPP-CMR using DL. Then, these images will be used to generate quantitative perfusion maps using a tracer kinetic model [141–143].

The problem with supervised learning techniques is the need to have fully sampled reference images to train the network, which are not available in FPP-CMR, particularly at high spatial resolutions. Even though the

field of MRI reconstruction with DL is currently an active area, to our knowledge and during the development of this work, self-supervised DL techniques have not been applied to FPP-CMR reconstruction. In this work, a **SElf-Supervised aCcelerated REconsTruction** (SECRET) DL framework for FPP-CMR is proposed to directly reconstruct contrast-enhanced dynamic image series from undersampled (k,t)-space data.

The proposed SECRET method directly reconstructs contrast-enhanced dynamic images from the undersampled (k,t)-space data. The method we follow here is similar to the one used in cine, except that we have not included explicit regularization and we have changed the residual learning strategy, averaging the zero-filled reconstructions. Considering only the undersampled (k,t)-space data when enforcing data consistency, we can train networks without the need for fully sampled images, simply by making use of the physical models in the reconstruction [97]. This framework can be formulated as follows:

$$\hat{\Theta} = \arg \min_{\Theta} \|\mathbf{A}\mathcal{F}C(\mathbf{m}|\Theta) - \mathbf{y}\|_2 \quad (6.1)$$

where $C(\mathbf{m}|\Theta)$ is the output of a CNN, with Θ the parameter vector to be optimized. Figure 6.1 shows the steps necessary for training our proposed SECRET method for FPP-CMR. First, undersampled (k,t)-space data are transformed to the image domain, obtaining \mathbf{m} . Then, \mathbf{m} enters the CNN to provide the reconstructed contrast-enhanced dynamic images $\hat{\mathbf{m}}$. These images are then transformed back to (k,t)-space and subsampling masks are applied, thus obtaining the undersampled version $\hat{\mathbf{y}}$. Finally, the loss is computed with $\hat{\mathbf{y}}$ and the input \mathbf{y} , to guide the training phase.

The CNN is based on the well-known U-Net [118], widely used in medical imaging, and which has been used in previous chapters. Skip connections are included to maintain information from previous layers, as well as to avoid the problem of vanishing gradients during backpropagation. At the end of the CNN, residual learning has been appended as in [97],

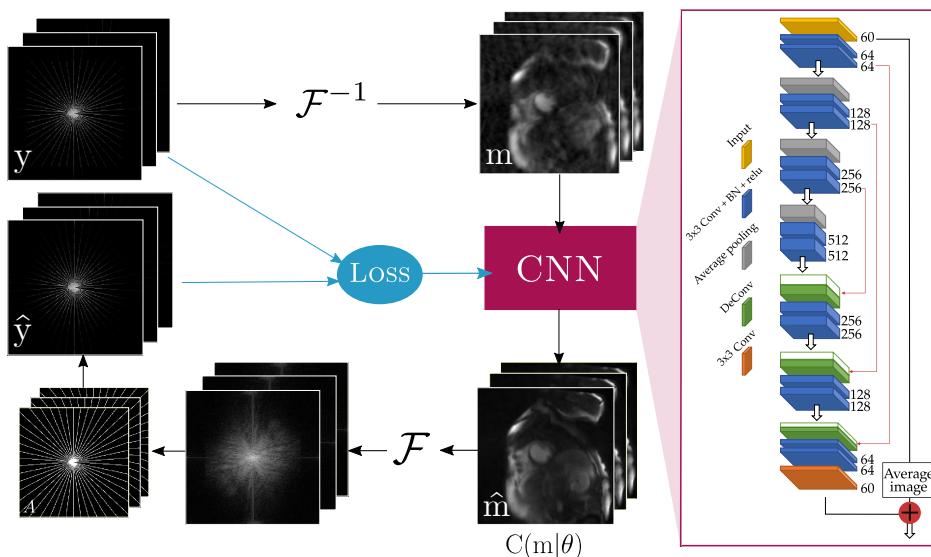


Figure 6.1: Flow chart illustrating the proposed SECRET method for FPP-CMR. Blue lines represent steps that only take place during training. The inputs of the framework are the undersampled (k,t) -space data y and the (k,t) -sampling masks A , resulting in the reconstructed contrast-enhanced dynamic images \hat{m} as output, and \hat{y} if required.

adding the average image of the input m .

6.2 Dataset

Rest and stress FPP-CMR acquisitions were performed in 21 patients using a single-bolus injection of 0.05 mmol/kg Gadobutrol (Gadovist; Bayer, Germany) and a 1.5T CMR scanner (MAGNETOM Aera, Siemens Healthineers, Erlangen, Germany) with an 18-channel chest-coil and a 32-channel spine coil. A free-breathing FLASH perfusion dual-sequence [143] was used to acquire a low-resolution image with low T1-sensitivity for estimating the arterial input function and three short-axis slices (basal, mid and apical) for high resolution myocardial perfusion imaging using the following

parameters: FOV = $340 \times 308 \text{ mm}^2$, in-plane resolution = $2.2 \times 2.2 \text{ mm}^2$, slice thickness = 10mm, TR/TE = 2.1/1ms, flip angle = 8° , parallel imaging acceleration factor 3, saturation recovery time = 100 ms, total scan duration = 60s, contrast agent relaxivity = 5.0L/mmol s. Undersampled datasets were generated for $3\times$, $6\times$ and $10\times$ acceleration factors, following a radial (k,t)-sampling trajectory.

Preprocessing A first step to ensure that all data had the same size, both spatially and temporally, prior to being fed to the CNN, consisted of resizing the DICOM images to obtain a spatial resolution of $2 \times 2 \text{ mm}^2$, padding the k-space to obtain an image size of 256×256 pixels, and interpolating each slice to a fixed number of frames (60 frames). A final step included intensity normalisation so that all contrast-enhanced dynamic image series present intensities between 0 and 1, without losing the contrast variation between frames. In addition, image pre-registration was also carried out to correct for respiratory motion.

Image quality metrics Image quality was assessed in terms of PSNR, SSIM and NRMSE between the reference images and reconstructions obtained with the SECRET, MoDL and CS ($10\times$ only) methods.

6.3 Implementation details

Patients were randomly split into training, validation and test subsets (60%, 16% and 24%, respectively). Each slice is fed into the SECRET framework so that the time frames are stacked in depth, creating a multi-channel image. The proposed method is implemented in Python with Tensorflow [121] and Keras [122], and it took about half an hour of training using the Adam optimizer [123] with a learning rate of 10^{-4} consuming about 3 GB of GPU memory for 100 epochs on one Intel[®] Core[™] i7-4790

CPU @ 3.60GHz with 16 GB RAM and one NVIDIA GeForce RTX 2080 Ti GPU. We use MoDL as a method of comparison. The MoDL training for $K=1$ and 100 epochs took one hour and a half and the MoDL training for $K=10$ and 200 epochs took forty-five hours using the same hardware. Note that after training the SECRET method, it provides a reconstruction of a complete contrast-enhanced dynamic image series in less than a second.

6.4 Evaluation

To perform the evaluation of our SECRET method, MoDL ($K=1$), MoDL ($K=10$), SECRET and CS reconstructions were performed on retrospectively subsampled k-space data with $AF=6$ and $AF=10$. Examples of that reconstructions can be found in Section 6.5. Reconstruction times are also exposed.

Although the images have been pre-registered, some residual movement is still visible. However, the reconstructions obtained with SECRET appear static. We see this fact by 1D projection of the dynamic images through time, for a given slice.

In order to verify that SECRET reconstructions yield valid quantitative maps, similar to those obtained from reference acquisitions, and to show the potential of SECRET for objective, operator-independent myocardial perfusion analysis, contrast transfer coefficient maps (K^{Trans}) are calculated from Patlak's model [144]. This model utilizes linear regression to study the pharmacokinetics of tracers that involve irreversible uptake. It does not rely on any particular compartmental model setup for the tracer and only assumes that the tracer's behavior can be approximated using two compartments.

6.5 Results and Discussion

Figure 6.2 shows the SECRET reconstructions obtained for two representative patients from $6\times$ and $10\times$ undersampled (k,t)-space data together with the reference and MoDL ($K=1$) reconstructions. CS reconstruction is also shown for $10\times$. Three different time frames are shown, corresponding to RV, LV and myocardial enhancement. Although the SECRET reconstructions are slightly blurred, due to residual learning from the average image of the CNN input (which is blurred due to residual motion), it can be seen that they have better quality than the images obtained with MoDL trained in the same amount of time. Moreover, SECRET images maintain the variability of contrast that exists between frames in addition to not losing the structure of the heart.

Figure 6.3 shows results of the FPP-CMR reconstructions in terms of PSNR, SSIM and NRMSE. While the performance of MoDL becomes noticeably worse as the acceleration rate increases, SECRET maintains good image quality even at high acceleration rates. For the $10\times$ accelerated reconstructions, the **median** (interquartile range): PSNR was **34.66** (3.47), **31.46** (3.81), **34.52** (5.43), **30.67** (5.52); SSIM was **0.94** (0.04), **0.92** (0.07), **0.96** (0.06), **0.92** (0.06); NRMSE was **0.12** (0.06), **0.16** (0.10), **0.11** (0.09), **0.17** (0.11) for CS, MoDL ($K=1$), MoDL ($K=10$) and SECRET methods, respectively. The image quality metrics indicate that SECRET images maintain a more stable agreement with the reference as the acceleration factor is increased than MoDL images, which deteriorate with higher acceleration. CS and MoDL ($K=10$) show the best agreement with the reference, but reconstructions take ~ 87.08 s and ~ 1.99 s, respectively, whereas MoDL ($K=1$) takes ~ 0.21 s and SECRET only 0.15s.

Figure 6.4 shows a 1D projection of the dynamic images through time, for a given slice. Note that although the images have been pre-registered, there is still some residual motion. SECRET does not include any explicit

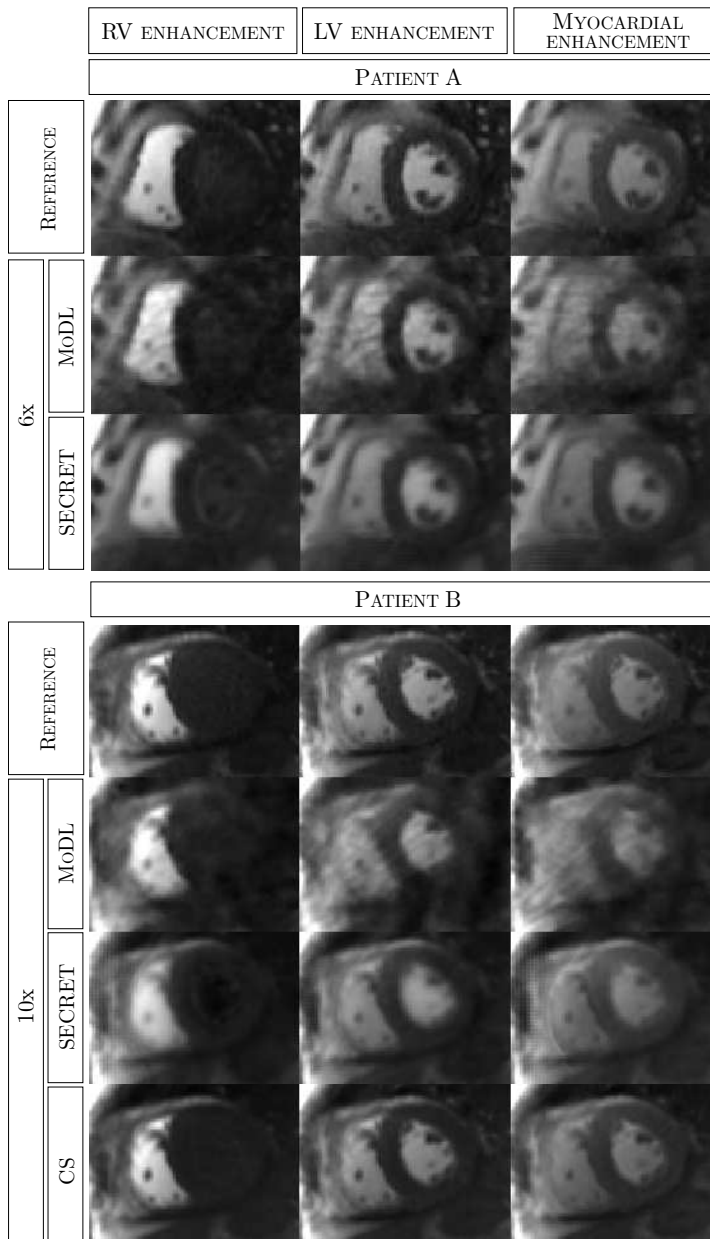


Figure 6.2: SECRET and MoDL ($K=1$) reconstructions obtained from $6\times$ and $10\times$ undersampled FPP-CMR data for two representative subjects. The reference images are displayed for comparison, in addition to CS reconstruction for $10\times$. The RV, LV and myocardial enhancement time frames are shown for one short axis slice.

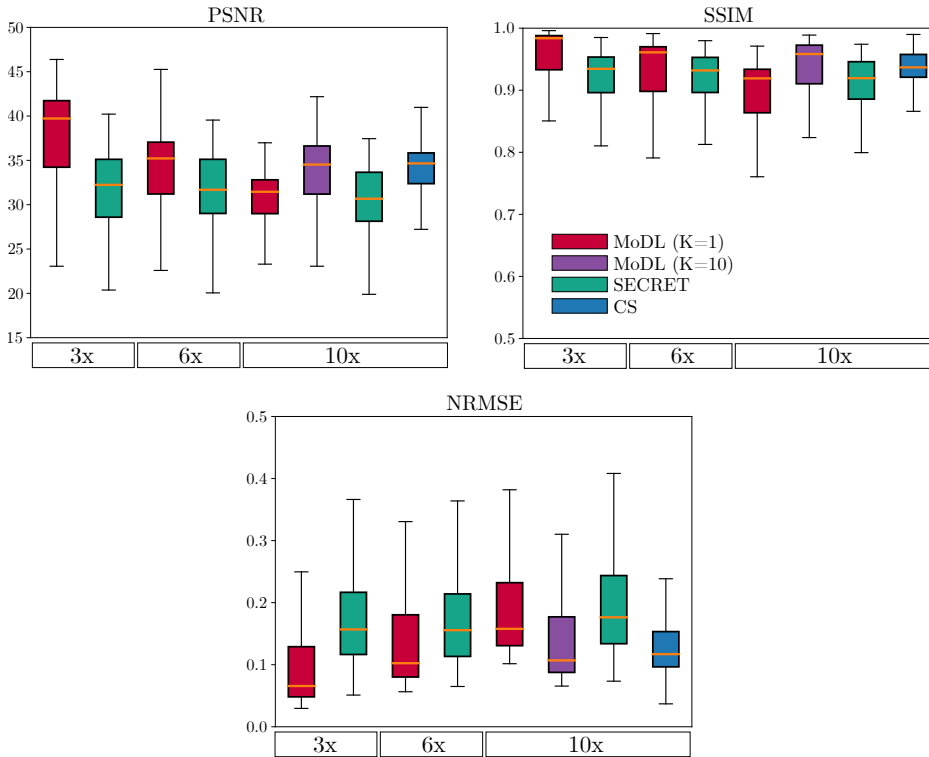


Figure 6.3: PSNR, SSIM and NRMSE between the reference images and the reconstructions obtained with SECRET and MoDL methods, for $3\times$, $6\times$ and $10\times$ acceleration factors, for all patients in the test dataset.

regularization term, however, due to the residual learning performed by the network all reconstructions provided by the framework are inherently corrected. Such good PSNR, SSIM and NRMSE values obtained when the reference images are affected by little respiratory motion, would certainly improve if some regularization were added. This would enable even higher acceleration rates.

Quantitative parameter maps were estimated from the FPP-CMR reconstructions, showing the potential of the technique for an objective and operator-independent analysis of myocardial perfusion. Figure 6.5 displays the contrast transfer coefficient (K^{Trans}) map estimated from fully

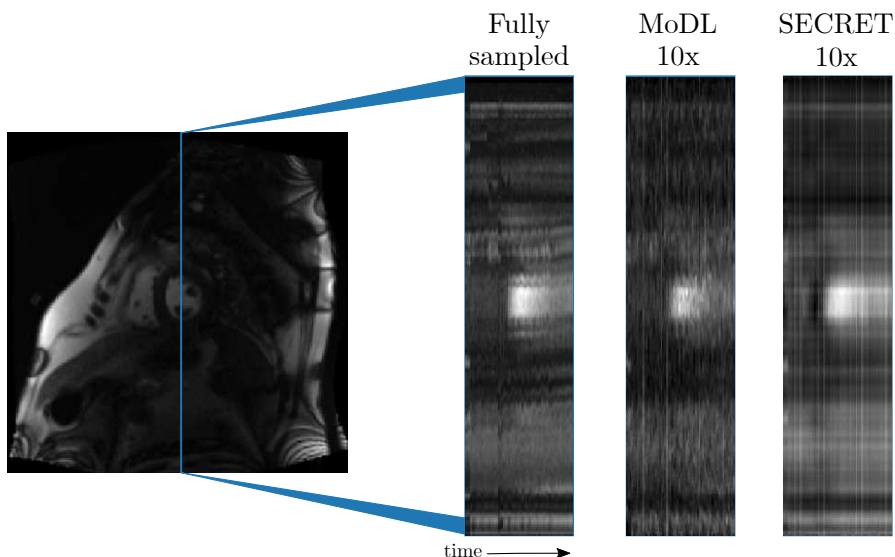


Figure 6.4: Representative image profile across the heart demonstrating that the SECRET framework improves consistency across time frames.

sampled, $6\times$ and $10\times$ undersampled patient data using the MoDL and SECRET methods, through the Patlak model [144]. The image quality of the quantitative maps obtained from the SECRET reconstruction at accelerations $6\times$ and $10\times$ is comparable to the reference images, showing less blurring than MoDL maps.

It is important to note that the challenge with supervised learning techniques, as MoDL, is the requirement for fully sampled reference images to train the network. However, these images are not readily accessible in the case of FPP-CMR, particularly when working with high spatial resolutions. We address this problem by utilizing a self-supervised learning approach that focuses solely on working with undersampled data.

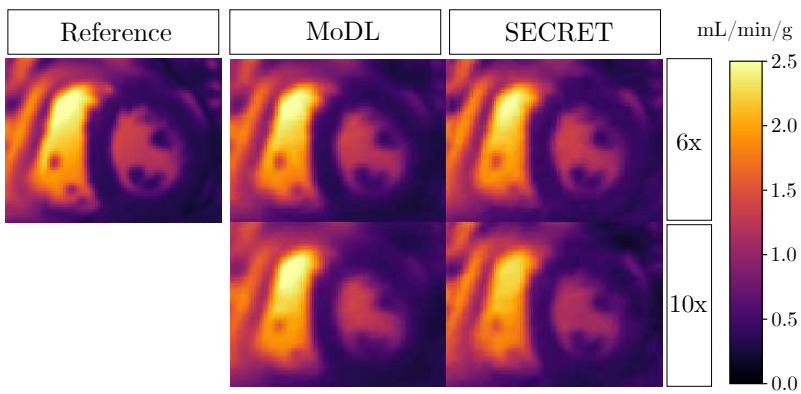


Figure 6.5: Quantitative maps (K^{Trans}) obtained from $6\times$ and $10\times$ undersampled data using MoDL and the SECRET methods. The reference image is displayed for comparison.

Chapter 7

CONCLUSIONS AND FUTURE WORK

This Chapter concludes this Thesis dissertation. It includes our contributions but also pinpoints our limitations as well as hypothesizes some future lines of research.

7.1 Conclusion

An MRI scan is a highly accurate and versatile medical imaging method that gives rise to in-depth images of soft tissue. However, it takes longer to complete than other procedures and requires skilled staff and specialized equipment. Subsampling can reduce scan time but requires additional computational effort to reconstruct the images, with added assumptions and constraints during the optimization process.

The main focus of this Thesis has been to accelerate the reconstruction of 2D CMR dynamic images, for both cine and perfusion, from highly undersampled acquisitions. The research in this Thesis has attempted to use motion as a source of information to reduce computational needs and improve the speed of optimization, while also maintaining the quality of the reconstructions. This approach takes advantage of the redundancy in the images in both spatial and temporal dimensions.

This statement was our main objective, from which four sub-objectives

were identified, namely (1) to give rise to a parallel device-agnostic version of GWCS, with special emphasis on the ME/MC procedure, (2) to design a DL architecture that achieves GW monomodal registration of cardiac cine in affordable devices, (3) to design a self-supervised DL-based solution that replicates GWCS performance at much lower computing times, and (4) to adapt the self-supervised kernel of the previous solution to account for FPP-CMR reconstruction.

The utilization of parallelization and DL techniques to accelerate the registration and reconstruction of dynamic cardiac MRI holds significant clinical implications. These advanced techniques empower faster image processing, resulting in enhanced clinical workflows and improved patient care. First of all, by reducing the processing time required for image registration and reconstruction, these techniques can significantly expedite the analysis and interpretation of cardiac MRI scans. This can improve the efficiency of clinical workflows, allowing healthcare professionals to promptly diagnose and treat cardiac conditions. To continue, this can facilitate the implementation of personalized treatment approaches. By obtaining real-time information on cardiac function and dynamics, clinicians can make more informed decisions regarding the selection and adjustment of treatment strategies. This personalized approach can contribute to better patient outcomes and improved management of cardiovascular conditions.

Finally, we can say that the advancements achieved through this work can serve as a foundation for further research and development. This can lead to the discovery of novel imaging biomarkers, improved diagnostic algorithms, and the development of innovative clinical tools for assessing cardiac health and identifying early signs of cardiovascular diseases.

The contributions derived from this Thesis (described in Section 7.2) demonstrate that the stated objectives have been achieved. This Thesis has certain limitations that suggest further lines of research, which are discussed in Section 7.3.

7.2 Contributions

The main contributions and results of this Thesis are:

- **dGW** DL-based approach for GW elastic registration of a dynamic cine MRI sequence. An unsupervised approach has been used for training and our solution is image-based as opposed to patch-based. Although we have focused on a monomodal application domain, the solution can handle multimodal metrics easily. We have provided an iterative solution and have observed that the iterations play a role in the registration quality. We have identified the network hyperparameters by a forward selection procedure and we have observed the network robustness with the image ordering within the sequence. In addition, we have compared our architecture with a state-of-the-art solution and results favored our design.

Although our registration method is intended for image reconstruction (as pointed out in Chapter 5), any application that requires material point tracking in a 2D dynamic sequence may benefit from it.

- **OpenCLIPER** Parallel groupwise registration system for motion estimation/compensation and a parallel multicoil NESTA subsystem for $\ell_1 - \ell_2$ problem solving. This system is a) clinically viable in terms of execution times, and b) suitable for any computing device which has an OpenCL implementation, including CPUs, GPUs, FPGAs and DSPs from main vendors. The use of our framework OpenCLIPER allowed us to partition the problem in independent black boxes (processes) which are then connected as needed and executed in parallel on any capable device, while the source code remains unique for all prospective computing devices. Device initialization and maintenance is reduced to a minimum as well, while

providing relevant development and debugging aids. Administrative but time-consuming tasks such as data loading/saving and kernel compilation are parallelized or cached so as to minimally affect overall performance.

- **SSMoComp** Efficient 2D cardiac MR cine Reconstruction through a Self-Supervised Motion Compensated architecture. A self-supervised approach has been used for training so there is no need for fully sampled images. The pretrained dGW network is used to carry out the ME/MC phase. It should be recalled that dGW was trained with fully sampled images, so the performance of the overall system can be improved by training dGW with sub-sampled images. We have identified the network hyperparameters, such as the weight of the regularization term λ and the number of iterations of dGW, that provide the best solution, among the values proposed by design. In addition, we have compared our architecture with a state-of-the-art solution and results favored our design.
- **SECRET** Physics-informed self-supervised deep learning reconstruction framework for accelerating FPP-CMR scans. The proposed SECRET method provides FPP-CMR reconstructions directly from the undersampled (k,t)-space data and does not require fully sampled reference data. Compared with state-of-the-art approaches, the SECRET method maintains good quality reconstructions for higher acceleration rates, with low training and very fast reconstruction times. The proposed SECRET method shows promising results, with the potential for improvement coupled with explicit regularization, which will be explored in future work.

7.3 Future work

During the development of this Thesis, we have encountered several limitations that open up new lines of research and improvement of the proposed solutions.

The first limitation is due to the 2D nature of the acquisitions, which means that, in the case of motion-aware solutions, the movement does not faithfully represent the real deformations suffered by the heart. Therefore, the extension to 3D, i.e. dynamic 3D (3D+t), emerges as a natural line of research. Furthermore, 3D reconstruction offers an advantage over 2D reconstruction because the latter requires tedious spatial planning due to the position of the heart and individual differences. The 3D method with isotropic resolution eliminates this issue by allowing for retrospective orientation from standard planes, but requires external contrast. Further work includes extending the 2D reconstruction to the free-breathing 3D problem, which poses additional issues due to typical data volumes that extend far beyond the capacity of a single computing device memory, along with much higher processing times.

Another limitation encountered comes from the fact that all the DL solutions proposed in this Thesis have been based on preprocessed data with an in-plane resolution of 1mm^2 , which forces to preprocess the data at that resolution. This leads to the idea of exploring a single DL solution that allows registrations and reconstructions at different resolutions.

Although the problem is formulated as multi-coil, given that the data we have are single-coil, the proposed solutions have been made ad-hoc for this type of data. The extension to multi-coil would imply a higher consumption of computational resources, but its implementation would be carried out in a straightforward manner.

In the particular case of SECRET, one possibility to consider is the inclusion of a regularization term, in the same way that is added in

SSMoComp.

It is worth mentioning that the retrospective undersampling approach was employed due to the nature of the available datasets, which were in DICOM format. However, we are aware of the importance of evaluating developments in prospectively undersampled data to create a more realistic scenarios.

Additionally, it is important to clarify the rationale behind the cropping of images at a resolution of 1mm^2 , as this decision was driven by the requirements of the CNNs employed in our methodology. To ensure consistency and compatibility within the network architecture, all images were resized to a uniform size. This approach allowed us to effectively process the images within the proposed DL frameworks, facilitating the training and evaluation stages.

While cropping the images may introduce a potential limitation in terms of preserving the full extent of the original data, it was necessary to achieve uniformity and facilitate the application of DL techniques. By resizing the images to a standardized size, we could ensure that each image had the same dimensions, enabling seamless integration into the neural network models. This uniformity was critical for maintaining consistency and facilitating accurate comparisons and evaluations across different datasets.

Future research efforts should explore alternative approaches that preserve the entirety of the acquired data while still maintaining the compatibility required for DL frameworks. Investigating techniques such as dynamic resizing may provide opportunities to leverage the full resolution of the images without compromising the network's compatibility and performance.

In terms of DL architectures or techniques, there are several options in addition to the CNNs used in this work. Some of them include: recurrent neural networks, transformer neural networks, siamese networks, or generative modeling techniques such as generative adversarial networks or diffusion models, the latter being the most popular in a variety of fields

at the moment. Diffusion models hold promise in various image analysis areas, but their direct application to dynamic cardiac MRI registration and reconstruction is relatively unexplored. While diffusion models have potential benefits, their suitability for these specific tasks requires further investigation and validation. Integrating diffusion models with existing techniques or developing specialized adaptations could be a direction for future research. Further exploration and validation are needed to assess the direct application of diffusion models in dynamic cardiac MRI.

In any case, more data are needed, from different resolutions, acquired with scanners from different vendors, to allow for more and better training of the DL solutions and to allow for generalization.

As for parallelization through GPU programming, it would be beneficial to parallelize the implementation of the B-splines through convolutions, as in [145], where the deformation is posed as a cascade of 1D convolutions along different dimensions, achieving great reduction in execution time for evaluation of transformations and gradients.

Appendix A

FORWARD AND BACKWARD OPERATORS

This appendix has been created for the purpose of clarification and completion of Section 2.4.1, where an attempt has been made to describe non-rigid complex displacements as linear combinations of B-splines using a group of control points. These control points will be defined over the original pixel mesh \mathbf{x} and the control point mesh will enclose the ROI inside the image.

The parameters Θ represent the displacements of the control points in the transformation, and are obtained by solving the problem stated in Equation 2.26 using any gradient-based optimization method. These Θ values are then used to transform the original \mathbf{x} values into the transformed \mathbf{x}' values using B-splines. The transformed \mathbf{x}' points can then be used for interpolation.

A.1 Control Point Mesh

The control points will determine the mesh, and their coordinates in the image coordinate space will be denoted by $\mathbf{P} = \{\mathbf{p}_{\mathbf{u}}\} = \{\mathbf{p}_{u_1, \dots, u_L}\}$, where $L = 2$ represents the dimensionality of the mesh. The resolution of the mesh, indicated by $\Delta^{\mathbf{P}}$, is a crucial factor in controlling the transformation

elasticity level. Although the resolution is set to be constant along one direction for simplicity, it may vary in other directions. With finer resolutions, we can achieve highly localized transformations, as each control point will only have an impact within a small vicinity. The size of this vicinity will be determined by a distance or radius of influence

$$r_l^{\mathbf{P}} = \frac{(E + 1)\Delta_l^{\mathbf{P}}}{2} \quad (\text{A.1})$$

in the different directions l . The value E represents the order of the B-spline curve, which is determined by the degree of the constituent polynomials. The compact support of these polynomials is $E + 1$. As a result, the range of pixels in the direction l that are impacted by a particular control point can be determined as

$$\left[\max(p_{u_l} - r_l^{\mathbf{P}}, x_{1,l}), \min(p_{u_l} + r_l^{\mathbf{P}}, x_{2,l}) \right] \quad (\text{A.2})$$

where x_l^{inf} and x_l^{sup} represent the inferior and superior pixel coordinate.

Let's consider a ROI that can be described by its bounding box limits, X_l^{inf} and X_l^{sup} , in the image coordinate space. Our objective is to create a mesh with as few control points as possible to cover the entire bounding box of the ROI. To avoid any boundary effects, we have added a certain margin in each direction, extending these limits. If a control point is positioned beyond these limits, it will not have any effect on the ROI. As a result, we can create a matrix called \mathbf{C} that indicates the lower and upper bounds of the control point mesh in each direction l , $\left[C_l^{\text{inf}}, C_l^{\text{sup}} \right]$. This indexes are defined as:

$$\begin{aligned} C_l^{\text{inf}} &= - \left\lfloor \frac{c_l - X_l^{\text{inf}} + r_l^{\mathbf{P}}}{\Delta_l^{\mathbf{P}}} \right\rfloor \\ C_l^{\text{sup}} &= \left\lfloor \frac{X_l^{\text{sup}} - c_l + r_l^{\mathbf{P}}}{\Delta_l^{\mathbf{P}}} \right\rfloor \end{aligned} \quad (\text{A.3})$$

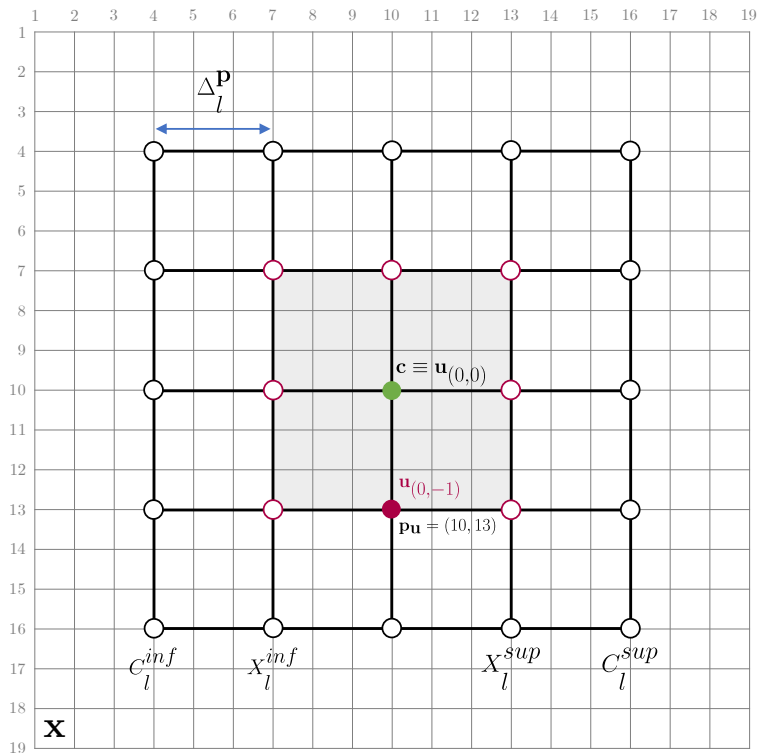


Figure A.1: Control point mesh design for non-rigid transformation of an object.

where \mathbf{c} represents the center of the ROI.

As a result, the mesh will contain $\prod_{l=1}^L (C_l^{sup} - C_l^{inf} + 1)$ control points. The number of control points and the radius of influence $r_l^{\mathbf{p}}$ are important factors that determine the complexity of the compensating transformation, as well as the computational costs associated with the procedure. Figure A.1 illustrates the elements involved in creating the control point mesh, which should aid in understanding the process.

A.2 GW Forward Transformation

The forward transformation is defined as the mapping of the template coordinates to the space of the images to be aligned. By means of this

mapping, what is achieved is to deform the images to be aligned so that they fit the template.

After defining the control point mesh, we can use Equation (2.27) to create a dense transformation \mathcal{T}_{Θ} that maps the original grid of pixels \mathbf{x} to non-grid locations \mathbf{x}' based on Θ . The displacements of the control points, Θ , can be obtained using an optimization method based on a gradient-based algorithm, such as Gradient Descent or Conjugate Gradient, in the ME step (see Equation (2.26)).

After registering, images are aligned to a reference frame using the forward transformation \mathcal{T}_{Θ} . Depending on the application, knowing the transformation that maps coordinates from one image to another can be useful. To define this transformation, the inverse mapping or backward transformation that maps coordinates from the reference frame to the input image coordinate frame is required.

A.3 GW Backward Transformation

Sometimes we need to use the inverse transformations, $\mathcal{T}_{\Theta}^{-1}$. However, we cannot analytically find the inverse of a B-spline. Also, in general, the inverse transformation will not be described by another B-spline transformation. Therefore, we use a numerical approach instead. We approximate $\mathcal{T}_{\Theta}^{-1}$ by finding the set of points $\tilde{\mathbf{x}}$ that are mapped to the regular Cartesian grid of spatial locations, \mathbf{x}' , in the original images when \mathcal{T}_{Θ} is applied. We solve this by

$$\tilde{\mathbf{x}} = \arg \min_{\mathbf{x}} \left\{ \sum_{\mathbf{x}' \in \mathcal{X}} \|\mathcal{T}_{\Theta}(\mathbf{x}) - \mathbf{x}'\|^2 \right\} \quad (\text{A.4})$$

Solving the optimization problem defined in Equation (A.4) is like creating a table for finding $\mathcal{T}_{\Theta}^{-1}$ for each point \mathbf{u} . This method does not require any assumptions about the inverse transformation's deformation model, but it is only applicable to the set of points in \mathcal{X} where Equa-

tion (A.4) is solved, which is consistent with the reconstruction techniques that have been utilized.

A.4 Image Formation: Intensity Interpolation

After calculating the forward and backward transformations, we can generate elastic transformations between images in our dataset by combining them. However, defining the elastic transformation is not sufficient. We must also perform an interpolation over the source images to obtain their intensities associated with the updated mesh resulting from either the forward or backward transformation.

The interpolation methods commonly used are nearest neighbor, linear, and cubic. Nearest neighbor involves associating the intensity of the pixel in the source image nearest to the target point x' . In linear interpolation, the intensity of the target point x' is obtained by weighting the intensity values of adjacent points around x' in the source image. Cubic interpolation is similar to linear interpolation but uses a cubic polynomial kernel instead of linear interpolations along each axis. In this Thesis, bilinear interpolation was utilized to obtain the intensity of the deformed images. Figure A.2 represents the interpolation that takes place after the spatial transformation, i.e. after finding the point mapping between the original image grid and the non-gridded locations, in both the forward and the backward case.

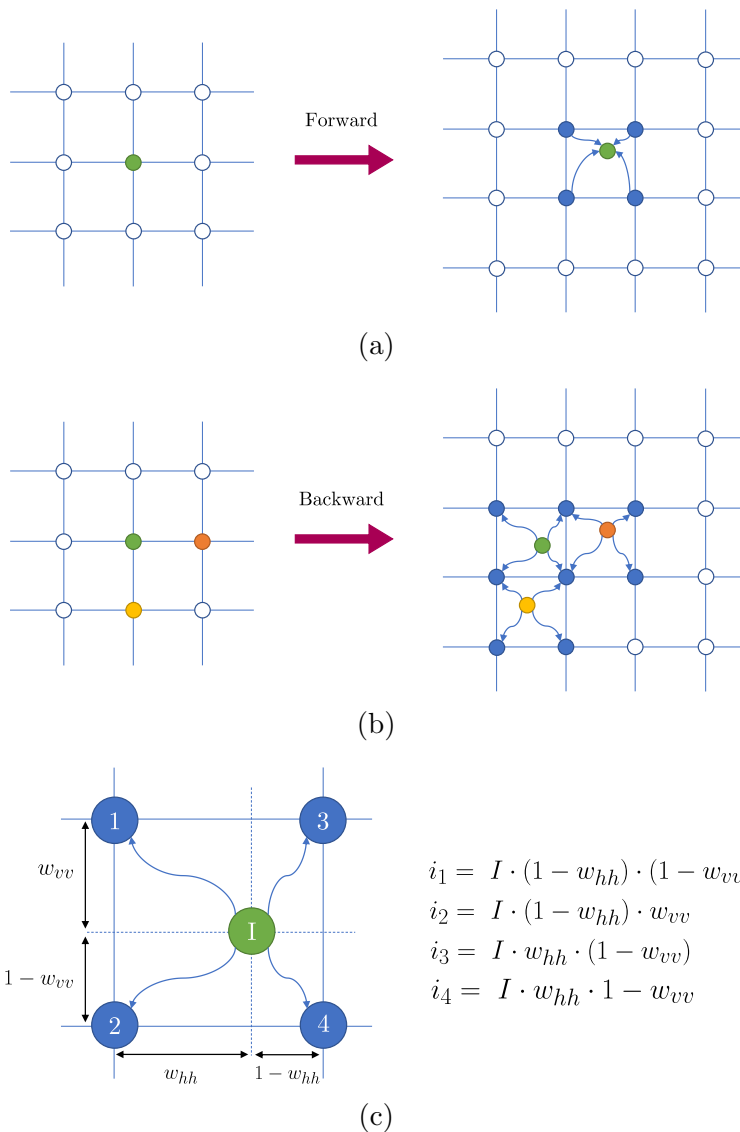


Figure A.2: Interpolation after (a) forward and (b) backward deformation. (c) Detail to better understand the interpolation after backward deformation. The contribution of the green point I to the adjacent points is shown. It should be noted that these points will also be affected by contributions from other points, e.g. point that has been numbered “4” is also affected by the orange and yellow points..

Appendix *B*

GRAPHICAL USER INTERFACES

Using code that has not been personally programmed increases the complexity of successfully executing it. During the development of the work presented in Chapter 3, we decided to create a couple of graphical user interfaces (GUIs) that would facilitate both the execution of training sessions and the use of previously trained networks to make predictions, i.e. register dynamic cardiac cine series.

B.1 Training GUI

In order to simplify the training of our own solution dGW, as well as VM alternatives, we developed a GUI that would allow not only to select which network to train, but also other parameters specific to each of the possible methods. Each of the tabs that make up this GUI correspond to a different network or method, from left to right: dGW, VM-2 and VMdiff. These tabs have a brief description of the corresponding method, as well as drop-down menus and text insertion boxes that allow the user to easily configure the training (see Figures B.1, B.2 and B.3).

Figure B.1 shows the training tab of our proposal, dGW. In the “Training specifications” section there is a drop-down menu that allows the user to choose the images to be used for training, which can be the complete

TRAIN OWN ARCHITECTURE **TRAIN VOXELMORPH** **TRAIN DIFFEOMORPHIC VM**

This program train a deep neural network based on an encoder-decoder architecture with skip connections in an unsupervised way. The goal is to perform the groupwise registration of any cardiac cine MRI sequence, in a short axis orientation. The loss that is minimized, and represents the registration problem is explained below.

The metric is calculated within region X , which may be the whole image or only an estimated area where most of the motion takes place.

$$\int_X \frac{1}{N} \sum_{n=1}^N \left(m_n(T_n(x)) - \sum_{k=1}^N m_k(T_k(x)) \right)^2 dx$$

In this equations, T_c is the cardiac cycle and λ_{p, R_p} and $1-p_i-4$ are the weights of the regularization terms R_p . These terms account for derivatives of the motion vector field; specifically $R1$ and $R2$ are respectively first and second order spatial derivatives while $R3$ and $R4$ are first and second temporal derivatives.

$$\int_X \int_0^{T_c} \sum_{p=1}^4 \lambda_p R_p dt dx$$

The solution is constrained to those sets of transformations with the average deformation of each position equal to the identity transformation:

$$\lambda_c \int_X \frac{1}{N} \sum_{i=1}^N (T_i(x) - x)^2 dx$$

TRAINING:
Being I_a a cardiac cine MRI sequence, we first calculate the reference REF_0 from an empiric manifold created with every frame in the temporal sequence and extracting the frame in the sequence that has the minimum geodesic distance with the others. With that reference a registration is performed obtaining a registered sequence I_0 . Then, the mean or average frame of the registered sequence is calculated, being this the new reference for another groupwise registration. This step is repeated i times or number of updates.

Training scheme for each sequence in the training dataset

TRAINING SPECIFICATIONS

Complete images [320 x 320] Number of epochs:

UPDATES

Initial update: Rest of updates:

REGULARIZATION WEIGHTS

Spatial regularization 1st: Spatial regularization 2nd: Temporal regularization 1st: Temporal regularization 2nd:

CONSTRAINT - Transformation degeneracy

Weight:

TRAIN

Elena Martin González

Figure B.1: Training GUI - dGW tab.

images or ROIs, as well as setting the number of epochs the training will consist of. In the “Updates” section, the initial template is configured, which can be either *Manifold*, in which case the automatic template selection described in section 3.3 will be used, or *Mean*, in which case the average image will be calculated for each dynamic series. The number of updates refers to L . Finally, the hyperparameters belonging to the weights of the loss function can be fixed, i.e. the four weights of the regularization terms (two spatial and two temporal) and the weight corresponding to the transformation degeneracy constraint.

Then, Figure B.2 shows the training tab of VM. As for dGW, the selection of the images to be used for training and the number of epochs is allowed. As for the loss function, it is possible to choose the similarity metric between MSE and NegCC, as well as to set the weight of the regularization term. Finally, Figure B.3 shows the training tab VMdiff. The “Training Specifications” are the same as in the other two tabs. The “Loss Parameters” section allows the user to set the λ and σ values

(see [108]).

TRAIN OWN ARCHITECTURE
TRAIN VOXELMORPH
TRAIN DIFFEOMORPHIC VM

Based on:
An Unsupervised Learning Model for Deformable Medical Image Registration
Guha Balakrishnan, Anyi Zhao, Mert R. Sabuncu, John Guttag, Adrian V. Dalca
CVPR 2018. eprint arXiv:1802.02604
<https://github.com/voxelmorph/voxelmorph>

VoxelMorph is focused on atlas-based registration, but we use its VM-2 variation to train the network to perform a registration of any cardiac cine MRI. The loss function consists on two components: the similarity metric that penalizes differences in appearance, and the regularization term that penalizes local spatial variations. They proposed the negative local cross-correlation as the similarity metric, but also the mean squared error. The local cross-correlation is written as:

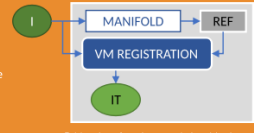
$$CC(F, M(\phi)) = \frac{\sum_{p \in \Omega} (\sum_{p_i} (F(p_i) - \hat{F}(p)))(M(\phi(p_i)) - \hat{M}(\phi(p)))^2}{\left(\sum_{p_i} (F(p_i) - \hat{F}(p))\right) \left(\sum_{p_i} (M(\phi(p_i)) - \hat{M}(\phi(p)))\right)}$$

The complete loss, for both similarity metrics, is therefore:

$$\mathcal{L}(F, M(\phi)) = -CC(F, M(\phi)) + \lambda \sum_{p \in \Omega} \|\nabla \phi(p)\|^2$$

$$\mathcal{L}(F, M(\phi)) = MSE(F, M(\phi)) + \lambda \sum_{p \in \Omega} \|\nabla \phi(p)\|^2$$

For more information see the paper whose reference is at the beginning.



Training scheme for each sequence in the training dataset.

TRAINING SPECIFICATIONS

Complete images [320 x 320] ▾

Number of iterations/epochs:

Steps per epoch: Nsubjects · (Nframes per subject)

SIMILARITY

Negative local cross-correlation ▾

REGULARIZATION

Weight:

Recommendation:
1.0 for ncc
0.01 for mse

TRAIN

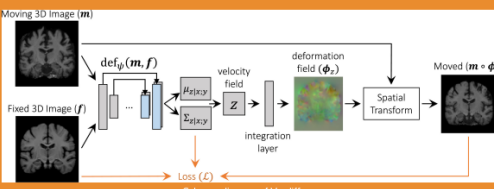
Elena Martin González

Figure B.2: Training GUI - VM tab.

TRAIN OWN ARCHITECTURE
TRAIN VOXELMORPH
TRAIN DIFFEOMORPHIC VM

Based on:
Unsupervised Learning of Probabilistic Diffeomorphic Registration for Images and Surfaces
Adrian V. Dalca, Guha Balakrishnan, John Guttag, Mert R. Sabuncu
Medical Image Analysis, 57:226-236, 10 2019.

VoxelMorph is focused on atlas-based registration, but we use its VMdiff variation to train the network to perform a registration of any cardiac cine MRI. For more information about the network, see reference. The following is a schematic diagram of operation taken from the reference.



Scheme diagram of VMdiff

The code, by default, has values of lambda=10 and sigma=0.02. However, the authors found that for their data, the optimal values were lambda=25 and sigma=0.01. With this application we allow the choice of that loss parameters.

The way the system trains is the same as described in the TRAIN VOXELMORPH tab. Go there for more information.

TRAINING SPECIFICATIONS

Complete images [320 x 320] ▾

Number of iterations/epochs:

Steps per epoch: Nsubjects · (Nframes per subject)

LOSS PARAMETERS

Recommendations for For VM data:

	Code defaults	README
lambda: <input style="width: 50px;" type="text"/>	10	25
sigma: <input style="width: 50px;" type="text"/>	0.02	0.01

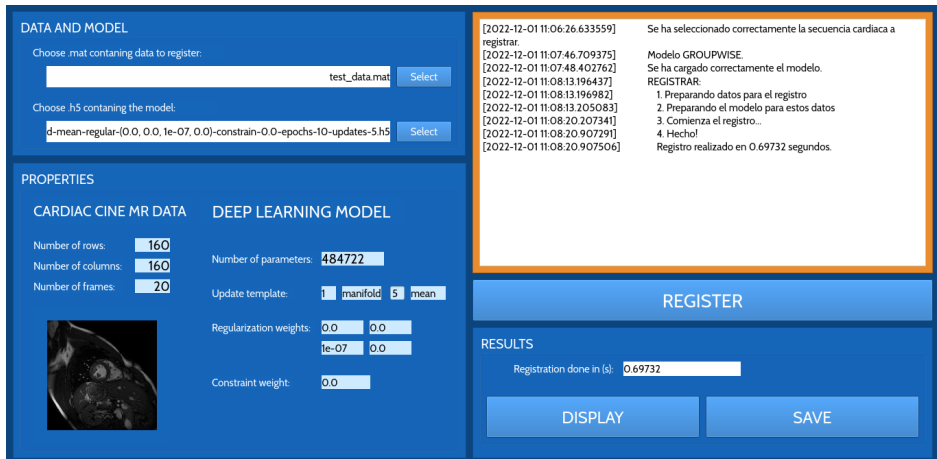
TRAIN

Elena Martin González

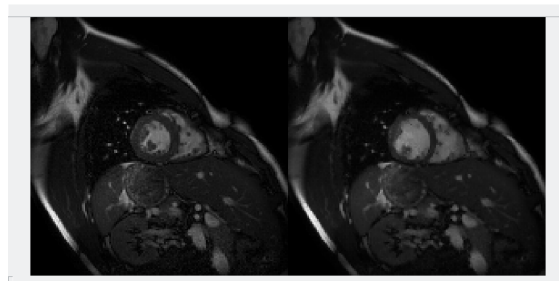
Figure B.3: Training GUI - VMdiff tab.

B.2 Registration GUI

The registration GUI in Figure B.4 allows for one prediction at a time. It can be used to quickly check whether a trained model succeeds in registering a selected dynamic cardiac cine series. With this GUI, when we load the data to be registered, we can see a sample frame, as well as the number of rows, columns and frames the series consists of. We can also see the configuration of the network model we want to test: weights of the regularization and constraint terms, how many and how are the template updates, and the number of parameters the model consists of. All this information is displayed on the left side of the main window (Figure B.4a). On the right, we can see a log of the data and model we have loaded. Once we have the data and the model loaded, we can proceed to “Register” and in the log we will see how long it takes to register. Once the sequence has been registered it can be displayed and/or saved. If the “Display” button is pressed, a new window (Figure B.4b) will appear showing a video of the original series (unregistered) and the registered series (if the selected model works as expected).



(a)



(b)

Figure B.4: Registration GUI. (a) Main window. (b) Display window: it shows two videos, the dynamic series to be registered (left) and the series once registered (right).

Bibliography

- [1] F. Bloch. Nuclear induction. *Physical Review*, 70:460, 10 1946.
- [2] E. M. Purcell, H. C. Torrey, and R. V. Pound. Resonance absorption by nuclear magnetic moments in a solid. *Physical review*, 69(1-2):37, 1946.
- [3] P. C. Lauterbur. Image formation by induced local interactions: Examples employing nuclear magnetic resonance. *Nature 1973 242:5394*, 242:190–191, 1973.
- [4] A. N. Garroway, P. K. Grannell, and P. Mansfield. Image formation in nmr by a selective irradiative process. *Journal of Physics C: Solid State Physics*, 7:L457, 12 1974.
- [5] P. Mansfield and A. A. Maudsley. Medical imaging by nmr. *The British journal of radiology*, 50(591):188–194, 1977.
- [6] E. S. H. Ibrahim. *Heart Mechanics: Magnetic Resonance Imaging—Advanced Techniques, Clinical Applications, and Future Trends*. CRC Press, 2017.
- [7] S. Mastrogiacomo, W. Dou, J. A. Jansen, and X. F. Walboomers. Magnetic resonance imaging of hard tissues and hard tissue engineered bio-substitutes. *Molecular Imaging and Biology*, 21(6):1003–1019, 2019.
- [8] D. Kumar and P. Elliott. *Principles and practice of clinical cardiovascular genetics*. Oxford University Press, USA, 2010.

- [9] E. S. H. Ibrahim, L. Frank, D. Baruah, V. Emre Arpinar, A. S. Nencka, K. M. Koch, L. Tugan Muftuler, O. Unal, J. Stojanovska, J. C. Rubenstein, S. A. Brown, J. Charlson, E. M. Gore, and C. Bergom. Value CMR: Towards a Comprehensive, Rapid, Cost-Effective Cardiovascular Magnetic Resonance Imaging. *International Journal of Biomedical Imaging*, 2021.
- [10] H. Nyquist. Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 47(2):617–644, 1928.
- [11] K. G. Hollingsworth. Reducing acquisition time in clinical mri by data undersampling and compressed sensing reconstruction. *Physics in Medicine & Biology*, 60(21):R297, 2015.
- [12] T. S. Jornada, C. H. Murata, and R. B. Medeiros. Influence of partial k-space filling on the quality of magnetic resonance images. *Radiologia Brasileira*, 49:158–164, 2016.
- [13] J. Royuela-Del-Val, L. Cordero-Grande, F. Simmross-Wattenberg, M. Martín-Fernández, and C. Alberola-López. Nonrigid groupwise registration for motion estimation and compensation in compressed sensing reconstruction of breath-hold cardiac cine MRI. *Magnetic Resonance in Medicine*, 75:1525–1536, 4 2016.
- [14] R. L. Glass. A structure-based critique of contemporary computing research. *Journal of Systems and Software*, 28(1):3–7, 1995.
- [15] J. Kontio et al. *Software engineering risk management: a method, improvement framework, and empirical evaluation*. Helsinki University of Technology, 2001.
- [16] F. Shull, J. Singer, and D. I. Sjøberg. *Guide to advanced empirical software engineering*. Springer, 2007.

- [17] L. G. Hanson. Is quantum mechanics necessary for understanding magnetic resonance? *Concepts in Magnetic Resonance Part A*, 32A:329–340, 9 2008.
- [18] R. Heo, R. Nakazato, D. Kalra, and J. K. Min. Noninvasive Imaging in Coronary Artery Disease. *Seminars in Nuclear Medicine*, 44:398–409, 9 2014.
- [19] J. R. J. Foley. Cardiovascular Magnetic Resonance Imaging for the Investigation of Ischaemic Heart Disease. M.D. thesis, 2018. University of Leeds.
- [20] R. C. Hendel, M. G. Friedrich, J. Schulz-Menger, C. Zemmrich, F. Bengel, D. S. Berman, P. G. Camici, S. D. Flamm, D. Le Guludec, R. Kim, M. Lombardi, J. Mahmarian, U. Sechtem, and E. Nagel. CMR First-Pass Perfusion for Suspected Inducible Myocardial Ischemia. *JACC: Cardiovascular Imaging*, 9:1338–1348, 11 2016.
- [21] J. Schwitter, C. M. Wacker, N. Wilke, N. Al-Saadi, E. Sauer, K. Huettler, S. O. Schönberg, A. Luchner, O. Strohm, H. Ahlstrom, T. Dill, N. Hoebel, and T. Simor. MR-IMPACT II: Magnetic Resonance Imaging for Myocardial Perfusion Assessment in Coronary artery disease Trial: perfusion-cardiac magnetic resonance vs. single-photon emission computed tomography for the detection of coronary artery disease: a comparative multicentre, multivendor trial. *European Heart Journal*, 34:775–781, 3 2013.
- [22] J. Montalt-Tordera, V. Muthurangu, A. Hauptmann, and J. A. Steeden. Machine learning in magnetic resonance imaging: image reconstruction. *Physica Medica*, 83:79–87, 2021.
- [23] A. Deshmane, V. Gulani, M. A. Griswold, and N. Seiberlich. Parallel mr imaging. *Journal of Magnetic Resonance Imaging*, 36(1):55–72, 2012.

- [24] K. P. Pruessmann, M. Weiger, M. B. Scheidegger, and P. Boesiger. Sense: sensitivity encoding for fast mri. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 42(5):952–962, 1999.
- [25] M. A. Griswold, P. M. Jakob, R. M. Heidemann, M. Nittka, V. Jellus, J. Wang, B. Kiefer, and A. Haase. Generalized autocalibrating partially parallel acquisitions (grappa). *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 47(6):1202–1210, 2002.
- [26] P. Kellman, F. H. Epstein, and E. R. McVeigh. Adaptive sensitivity encoding incorporating temporal filtering (tsense). *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 45(5):846–852, 2001.
- [27] F. A. Breuer, P. Kellman, M. A. Griswold, and P. M. Jakob. Dynamic autocalibrated parallel imaging using temporal grappa (tgrappa). *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 53(4):981–985, 2005.
- [28] J. Tsao and S. Kozerke. Mri temporal acceleration techniques. *Journal of Magnetic Resonance Imaging*, 36(3):543–560, 2012.
- [29] J. Tsao, P. Boesiger, and K. P. Pruessmann. k-t BLAST and k-t SENSE: dynamic MRI with high frame rate exploiting spatiotemporal correlations. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 50(5):1031–1042, 2003.
- [30] B. D. Allen, M. Carr, M. P. F. Botelho, A. A. Rahsepar, M. Markl, M. O. Zenge, M. Schmidt, M. S. Nadar, B. Spottiswoode, J. D. Collins, et al. Highly accelerated cardiac MRI using iterative SENSE

- reconstruction: initial clinical experience. *The International Journal of Cardiovascular Imaging*, 32(6):955–963, 2016.
- [31] M. Lustig, D. Donoho, and J. M. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.
- [32] S. Geethanath, R. Reddy, A. S. Konar, S. Imam, R. Sundaresan, R. B. DR, and R. Venkatesan. Compressed sensing MRI: a review. *Critical ReviewsTM in Biomedical Engineering*, 41(3), 2013.
- [33] M. Blasche, C. Forman, and Siemens Healthineers. Compressed sensing—the flowchart. *MAGNETOM Flash*, 14:1–4, 2016.
- [34] H. Jung, K. Sung, K. S. Nayak, E. Y. Kim, and J. C. Ye. k-t FOCUSS: a general compressed sensing framework for high resolution dynamic MRI. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 61(1):103–116, 2009.
- [35] H. Jung, J. Park, J. Yoo, and J. C. Ye. Radial k-t FOCUSS for high-resolution cardiac cine MRI. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 63(1):68–78, 2010.
- [36] H. Jung and J. C. Ye. Motion estimated and compensated compressed sensing dynamic magnetic resonance imaging: What we can learn from video compression techniques. *International Journal of Imaging Systems and Technology*, 20(2):81–98, 2010.
- [37] R. Otazo, D. Kim, L. Axel, and D. K Sodickson. Combination of compressed sensing and parallel imaging with respiratory motion correction for highly-accelerated cardiac perfusion MRI. *Journal of Cardiovascular Magnetic Resonance*, 13(1):1–2, 2011.

- [38] M. Doneva, C. Stehning, K. Nehrke, and P. Börnert. Improving scan efficiency of respiratory gated imaging using compressed sensing with 3d cartesian golden angle sampling. In *Proc Intl Soc Mag Reson Med*, volume 19, page 641, 2011.
- [39] P. G. Batchelor, D. Atkinson, P. Irarrázaval, D. L. G. Hill, J. Hajnal, and D. Larkman. Matrix description of general motion correction applied to multishot images. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 54(5):1273–1280, 2005.
- [40] J. F. M. Schmidt, M. Buehrer, P. Boesiger, and S. Kozerke. Nonrigid retrospective respiratory motion correction in whole-heart coronary mra. *Magnetic Resonance in Medicine*, 66(6):1541–1549, 2011.
- [41] F. Odille, P. A. Vuissoz, P. Y. Marie, and J. Felblinger. Generalized reconstruction by inversion of coupled systems (grics) applied to free-breathing mri. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 60(1):146–157, 2008.
- [42] M. S. Hansen, T. S. Sørensen, A. E. Arai, and P. Kellman. Retrospective reconstruction of high temporal resolution cine images from real-time mri using iterative motion correction. *Magnetic Resonance in Medicine*, 68(3):741–750, 2012.
- [43] M. Usman, D. Atkinson, F. Odille, C. Kolbitsch, G. Vaillant, T. Schaeffter, P. G. Batchelor, and C. Prieto. Motion corrected compressed sensing for free-breathing dynamic cardiac MRI. *Magnetic Resonance in Medicine*, 70(2):504–516, 2013.
- [44] MJ White, DJ Hawkes, A Melbourne, DJ Collins, C Coolens, M Hawkins, MO Leach, and D Atkinson. Motion artifact correction in free-breathing abdominal mri using overlapping partial samples to

- recover image deformations. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 62(2):440–449, 2009.
- [45] J. Royuela-del Val, L. Cordero-Grande, F. Simmross-Wattenberg, M. Martín-Fernández, and C. Alberola-López. Jacobian weighted temporal total variation for motion compensated compressed sensing reconstruction of dynamic mri. *Magnetic Resonance in Medicine*, 77(3):1208–1215, 2017.
- [46] S. Becker, J. Bobin, and E. J. Candès. Nesta: A fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39, 2011.
- [47] M. Polfliet, S. Klein, W. Huizinga, M. M. Paulides, W. J. Niessen, and J. Vandemeulebroucke. Intrasubject multimodal groupwise registration with the conditional template entropy. *Medical Image Analysis*, 46:15–25, 5 2018.
- [48] E. Martín-González, T. Sevilla, A. Revilla-Orodea, P. Casaseca-de-la Higuera, and C. Alberola-López. Groupwise non-rigid registration with deep learning: an affordable solution applied to 2D cardiac CINE MRI reconstruction. *Entropy*, 22(6), 2020.
- [49] J. Woo, M. Stone, and J. L. Prince. Multimodal registration via mutual information incorporating geometric and spatial context. *IEEE Transactions on Image Processing*, 24(2):757–769, 2014.
- [50] F. Alam, S. U. Rahman, A. U. Din, and F. Qayum. Medical image registration: Classification, applications and issues. *Journal of Postgraduate Medical Institute*, 32, 11 2018.
- [51] C. T. Metz, S. Klein, M. Schaap, T. van Walsum, and W. J. Niessen. Nonrigid registration of dynamic medical imaging data using nd+t

- b-splines and a groupwise optimization approach. *Medical Image Analysis*, 15:238–249, 4 2011.
- [52] L. Romero and F. Calderón. A tutorial on parametric image registration. *Scene Reconstruction, Pose Estimation and Tracking*, pages 167–184, 2007.
- [53] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence*, 11(6):567–585, 1989.
- [54] D. Rueckert. Nonrigid registration using free-form deformations: Application to breast MR images. *IEEE Transactions on Medical Imaging*, 18:712–721, 1999.
- [55] J. P. Thirion. Non-rigid matching using demons. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 245–251. IEEE, 1996.
- [56] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International Journal of Computer Vision*, 61(2):139–157, 2005.
- [57] M. I. Miller, A. Trouvé, and L. Younes. Geodesic shooting for computational anatomy. *Journal of Mathematical Imaging and Vision*, 24(2):209–228, 2006.
- [58] R. S. Chadwick. Mechanics of the left ventricle. *Biophysical Journal*, 39:279–288, 9 1982.
- [59] A. F. Frangi, W. J. Niessen, and M. A. Viergever. Three-dimensional modeling for functional analysis of cardiac images: A review. *IEEE Transactions on Medical Imaging*, 20:2–25, 1 2001.
- [60] W. Sun, W. J. Niessen, and S. Klein. Free-form deformation using lower-order B-spline for nonrigid image registration. In *International*

- Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 194–201. Springer, 2014.
- [61] A. Godino-Moya. *Contributions on groupwise registration for cardiac CINE magnetic resonance reconstruction*. PhD thesis, Universidad de Valladolid, 2021.
- [62] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 151–160, 1986.
- [63] C. De Boor. *A practical guide to splines*, volume 27. springer-Verlag New York, 1978.
- [64] P. Després and X. Jia. A review of GPU-based medical image reconstruction. *Physica Medica*, 42:76–92, 2017.
- [65] H. Wang, H. Peng, Y. Chang, and D. Liang. A survey of gpu-based acceleration techniques in mri reconstructions. *Quantitative imaging in medicine and surgery*, 8(2):196, 2018.
- [66] M. Uecker, F. Ong, J. I. Tamir, D. Bahri, P. Virtue, J. Y. Cheng, T. Zhang, and M. Lustig. Berkeley advanced reconstruction toolbox. In *Proc. Intl. Soc. Mag. Reson. Med.*, volume 23, 2015.
- [67] M. Schacht Hansen and T. S. Sørensen. Gadgetron: an open source framework for medical image reconstruction. *Magnetic Resonance in Medicine*, 69(6):1768–1776, 2013.
- [68] J. I. Tamir, F. Ong, J. Y. Cheng, M. Uecker, and M. Lustig. Generalized magnetic resonance image reconstruction using the berkeley advanced reconstruction toolbox. In *ISMRM Workshop on Data Sampling & Image Reconstruction, Sedona, AZ*, 2016.
- [69] T. S. Sorensen, D. Atkinson, T. Schaeffter, and M. S. Hansen. Real-time reconstruction of sensitivity encoded radial magnetic resonance

- imaging using a graphics processing unit. *IEEE Transactions on Medical Imaging*, 28(12):1974–1985, 2009.
- [70] R. E. Ansorge, S. J. Sawiak, and G. B. Williams. Exceptionally fast non-linear 3D image registration using GPUs. In *2009 IEEE Nuclear Science Symposium Conference Record (NSS/MIC)*, pages 4088–4094. IEEE, 2009.
- [71] M. Modat, Z. A. Taylor, J. Barnes, D. J. Hawkes, N. C. Fox, and S. Ourselin. Fast free-form deformation using the normalised mutual information gradient and graphics processing units. *Med Phys. v98 i3*, pages 278–284.
- [72] D. Ruijters, B. M. ter Haar Romeny, and P. Suetens. GPU-accelerated elastic 3D image registration for intra-surgical applications. *Computer methods and programs in biomedicine*, 103(2):104–112, 2011.
- [73] X. Du, J. Dang, Y. Wang, S. Wang, and T. Lei. A parallel non-rigid registration algorithm based on b-spline for medical images. *Computational and Mathematical Methods in Medicine*, 2016, 2016.
- [74] N. D. Ellingwood, Y. Yin, M. Smith, and C. L. Lin. Efficient methods for implementation of multi-level nonrigid mass-preserving image registration on gpus and multi-threaded cpus. *Computer Methods and Programs in Biomedicine*, 127:290–300, 2016.
- [75] K. Punithakumar, P. Boulanger, and M. Noga. A GPU-accelerated deformable image registration algorithm with applications to right ventricular segmentation. *IEEE Access*, 5:20374–20382, 2017.
- [76] The Khronos Group, Inc. [Online] Available: The Khronos Group. <https://www.khronos.org/>.
- [77] F. Simmross-Wattenberg, M. Rodríguez-Cayetano, J. Royuela-del Val, E. Martín-Gonzalez, E. Moya-Sáez, M. Martín-Fernández, and

- C. Alberola-López. Opencliper: an opencl-based c++ framework for overhead-reduced medical image processing and reconstruction on heterogeneous devices. *IEEE journal of biomedical and health informatics*, 23(4):1702–1709, 2018.
- [78] Chollet F. *Deep learning with Python*. Manning Publications, 2018.
- [79] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.
- [80] B. Patel and A. N. Makaryus. Artificial intelligence advances in the world of cardiovascular imaging. *Healthcare 2022, Vol. 10, Page 154*, 10:154, 1 2022.
- [81] K. Suzuki. Overview of deep learning in medical imaging. *Radiological physics and technology*, 10(3):257–273, 2017.
- [82] J. Kim, J. Hong, and H. Park. Prospects of deep learning for medical imaging. *Precision and Future Medicine*, 2(2):37–52, 2018.
- [83] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M. L. Shyu, S. C. Chen, and S. S Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5):1–36, 2018.
- [84] M. Bakator and D. Radosav. Deep learning and medical diagnosis: A review of literature. *Multimodal Technologies and Interaction*, 2(3):47, 2018.
- [85] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

- [86] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [87] E. Ahishakiye, M. Bastiaan Van Gijzen, J. Tumwiine, R. Wario, and J. Obungoloch. A survey on deep learning in medical image reconstruction. *Intelligent Medicine*, 1(03):118–127, 2021.
- [88] H. K. Aggarwal, M. P. Mani, and M. Jacob. Modl: Model-based deep learning architecture for inverse problems. *IEEE Transactions on Medical Imaging*, 38:394–405, 2 2019.
- [89] D. Kocanaogullari and E. M. Eksioglu. Deep learning for mri reconstruction using a novel projection based cascaded network. *IEEE International Workshop on Machine Learning for Signal Processing, MLSP*, 2019-October, 10 2019.
- [90] B. Yaman, S. Amir Hossein Hosseini, S. Moeller, J. Ellermann, K. Uğurbil, and M. Akçakaya. Self-supervised learning of physics-guided reconstruction neural networks without fully sampled reference data. *Magnetic Resonance in Medicine*, 84:3172–3191, 12 2020.
- [91] W. Joon Do, S. Seo, Y. Han, J. Chul Ye, S. Hong Choi, and S. Hong Park. Reconstruction of multicontrast mr images through deep learning. *Medical Physics*, 47:983–997, 3 2020.
- [92] L. Y. Hsu, M. Jacobs, M. Benovoy, A. D. Ta, H. M. Conn, S. Winkler, A. M. Greve, M. Y. Chen, S. M. Shanbhag, W. P. Bandettini, et al. Diagnostic performance of fully automated pixel-wise quantitative myocardial perfusion imaging by cardiovascular magnetic resonance. *JACC: Cardiovascular Imaging*, 11(5):697–707, 2018.
- [93] J. Schlemper, J. Caballero, J. V. Hajnal, A. N. Price, and D. Rueckert. A deep cascade of convolutional neural networks for dynamic mr

- image reconstruction. *IEEE Transactions on Medical Imaging*, 37:491–503, 2 2018.
- [94] T. Küstner, N. Fuin, K. Hammernik, A. Bustin, H. Qi, R. Hajhosseiny, P. G. Masci, R. Neji, D. Rueckert, R. M. Botnar, et al. Cinenet: deep learning-based 3d cardiac cine mri reconstruction with multi-coil complex-valued 4d spatio-temporal convolutions. *Scientific reports*, 10(1):1–13, 2020.
- [95] J. Pan, D. Rueckert, T. Küstner, and K. Hammernik. Learning-based and unrolled motion-compensated reconstruction for cardiac mr cine imaging. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VI*, pages 686–696. Springer, 2022.
- [96] J. Zou, C. Li, S. Jia, R. Wu, T. Pei, H. Zheng, and S. Wang. Selfcolearn: Self-supervised collaborative learning for accelerating dynamic mr imaging. *Bioengineering*, 9(11):650, 2022.
- [97] F. Liu, R. Kijowski, G. El Fakhri, and L. Feng. Magnetic resonance parameter mapping using model-guided self-supervised deep learning. *Magnetic Resonance in Medicine*, 85:3211–3226, 6 2021.
- [98] Chang Min Hyun, Hwa Pyung Kim, Sung Min Lee, Sungchul Lee, and Jin Keun Seo. Deep learning for undersampled mri reconstruction. *Physics in Medicine & Biology*, 63(13):135007, 2018.
- [99] Bo Zhu, Jeremiah Z Liu, Stephen F Cauley, Bruce R Rosen, and Matthew S Rosen. Image reconstruction by domain-transform manifold learning. *Nature*, 555(7697):487–492, 2018.
- [100] Yağmur Güçlütürk, Umut Güçlü, Katja Seeliger, Sander Bosch, Rob van Lier, and Marcel A van Gerven. Reconstructing perceived

- faces from brain activations with deep adversarial neural decoding. *Advances in neural information processing systems*, 30, 2017.
- [101] J. M. Guyader, W. Huizinga, D. HJ. Poot, M. van Kranenburg, A. Uitterdijk, W. J. Niessen, and S. Klein. Groupwise image registration based on a total correlation dissimilarity measure for quantitative mri and dynamic imaging data. *Scientific reports*, 8(1):1–14, 2018.
- [102] G. Haskins, U. Kruger, and P. Yan. Deep learning in medical image registration: a survey. *Machine Vision and Applications*, 31(1):1–18, 2020.
- [103] B. Pontré, B. R. Cowan, E. DiBella, S. Kulaseharan, D. Likhite, N. Noorman, L. Tautz, N. Tustison, G. Wollny, A. A. Young, et al. An open benchmark challenge for motion correction of myocardial perfusion mri. *IEEE Journal of Biomedical and Health Informatics*, 21(5):1315–1326, 2016.
- [104] Y. Hu, M. Modat, E. Gibson, W. Li, N. Ghavami, E. Bonmati, G. Wang, S. Bandula, C. M. Moore, M. Emberton, et al. Weakly-supervised convolutional neural networks for multimodal image registration. *Medical Image Analysis*, 49:1–13, 2018.
- [105] S. Zhao, T. Lau, J. Luo, I. Eric, C. Chang, and Y. Xu. Unsupervised 3d end-to-end medical image registration with volume tweening network. *IEEE Journal of Biomedical and Health Informatics*, 24(5):1394–1404, 2019.
- [106] J. Krebs, T. Mansi, B. Mailhé, N. Ayache, and H. Delingette. 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, 2018.

- [107] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca. Voxelmorph: A learning framework for deformable medical image registration. *IEEE Transactions on Medical Imaging*, 38:1788–1800, 8 2019.
- [108] A. V. Dalca, G. Balakrishnan, J. Guttag, and M. R. Sabuncu. Un-supervised learning of probabilistic diffeomorphic registration for images and surfaces. *Medical Image Analysis*, 57:226–236, 10 2019.
- [109] T. Che, Y. Zheng, J. Cong, Y. Jiang, Y. Niu, W. Jiao, B. Zhao, and Y. Ding. Deep group-wise registration for multi-spectral images from fundus images. *IEEE Access*, 7:27650–27661, 2019.
- [110] S. Ahmad, J. Fan, P. Dong, X. Cao, P. T. Yap, and D. Shen. Deep learning deformation initialization for rapid groupwise registration of inhomogeneous image populations. *Frontiers in Neuroinformatics*, 13:34, 2019.
- [111] H. Siebert and M. P. Heinrich. Deep groupwise registration of MRI using deforming autoencoders. In *Bildverarbeitung für die Medizin 2020*, pages 236–241. Springer, 2020.
- [112] Y. Zhang, X. Wu, H. M. Gach, H. Li, and D. Yang. GroupRegNet: a groupwise one-shot deep learning-based 4D image registration method. *Physics in Medicine & Biology*, 66(4):045030, 2021.
- [113] J. Yang, T. Küstner, P. Hu, P. Liò, and H. Qi. End-to-end deep learning of non-rigid groupwise registration and reconstruction of dynamic mri. *Frontiers in Cardiovascular Medicine*, 9, 2022.
- [114] S. Biswas, H. K. Aggarwal, and M. Jacob. Dynamic mri using model-based deep learning and storm priors: Modl-storm. *Magnetic resonance in medicine*, 82(1):485–494, 2019.

- [115] E. Martín-González, P. Casaseca-de-la Higuera, L. M. San-José-Revuelta, and C. Alberola-López. Groupwise Deep Learning-based Approach for Motion Compensation. Application to Compressed Sensing 2D Cardiac Cine MRI Reconstruction. In *XXXVII Congreso Anual de la Sociedad Española de Ingeniería Biomédica*, pages 299–302, 2019.
- [116] S. Sanz-Estébanez, L. Cordero-Grande, T. Sevilla, A. Revilla-Orodea, R. de Luis-García, M. Martín-Fernández, and C. Alberola-López. Vortical features for myocardial rotation assessment in hypertrophic cardiomyopathy using cardiac tagged magnetic resonance. *Medical Image Analysis*, 47:191–202, 7 2018.
- [117] L. Cordero-Grande, S. Merino-Caviedes, S. Aja-Fernandez, and C. Alberola-Lopez. Groupwise elastic registration by a new sparsity-promoting metric: Application to the alignment of cardiac magnetic resonance perfusion images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:2638–2650, 2013.
- [118] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [119] Q. Feng, Y. Zhou, X. Li, Y. Mei, Z. Lu, Y. Zhang, Y. Feng, Y. Liu, W. Yang, and W. Chen. Liver dce-mri registration in manifold space based on robust principal component analysis open. *Nature Publishing Group*, 2016.
- [120] K. K. Bhatia, J. V. Hajnal, B. K. Puri, A. D. Edwards, and D. Rueckert. Consistent groupwise non-rigid registration for atlas construction. *2004 2nd IEEE International Symposium on Biomedical Imaging: Macro to Nano*, 1:908–911, 2004.

- [121] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [122] F. Chollet et al. Keras, 2015.
- [123] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [124] A. V. Dalca, J. Guttag, and M. R. Sabuncu. Anatomical priors in convolutional networks for unsupervised biomedical segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [125] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 4 2004.
- [126] B. Rosner. *Fundamentals of biostatistics*. Cengage learning, 2015.
- [127] S. Theodoridis and K. Koutroumbas. *Pattern Recognition, Fourth Edition*. Academic Press, Inc., USA, 4th edition, 2008.
- [128] T. M. Cover, J. A Thomas, et al. Entropy, relative entropy and mutual information. *Elements of information theory*, 2(1):12–13, 1991.
- [129] J. P. Lewis. Fast normalized cross-correlation. *Vision Interface*, pages 120–123, 1995.

- [130] Xiao Yang, Roland Kwitt, Martin Styner, and Marc Niethammer. Quicksilver: Fast predictive image registration—a deep learning approach. *NeuroImage*, 158:378–396, 2017.
- [131] H. Sokooti, B. De Vos, F. Berendsen, B. P. F. Lelieveldt, I. Išgum, and M. Staring. Nonrigid image registration using multi-scale 3d convolutional neural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 232–239. Springer, 2017.
- [132] Guorong Wu, Minjeong Kim, Qian Wang, Brent C Munsell, and Dinggang Shen. Scalable high-performance image registration framework by unsupervised deep feature representations learning. *IEEE Transactions on Biomedical Engineering*, 63(7):1505–1516, 2015.
- [133] Advanced Micro Devices, Inc. clFFT. <https://github.com/clMathLibraries/clFFT>.
- [134] C. Nugteren. Clblast: A tuned opencl blas library. In *Proceedings of the International Workshop on OpenCL*, pages 1–10, 2018.
- [135] M. Salman Asif, L. Hamilton, M. Brummer, and J. Romberg. Motion-adaptive spatio-temporal regularization for accelerated dynamic mri. *Magnetic resonance in medicine*, 70:800–812, 2013.
- [136] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.
- [137] S. Aja-Fernández, R. S. J. Estepar, C. Alberola-López, and C. F. Westin. Image quality assessment based on local variance. In *2006 international conference of the ieee engineering in medicine and biology society*, pages 4815–4818. IEEE, 2006.

- [138] S. G. Lingala, Y. Hu, E. Dibella, and M. Jacob. Accelerated dynamic MRI exploiting sparsity and low-rank structure: k-t SLR. *IEEE transactions on medical imaging*, 30:1042, 5 2011.
- [139] R. Otazo, D. Kim, L. Axel, and D. K. Sodickson. Combination of compressed sensing and parallel imaging for highly accelerated first-pass cardiac perfusion mri. *Magnetic resonance in medicine*, 64:767–776, 2010.
- [140] V. Vitanis, R. Manka, D. Giese, H. Pedersen, S. Plein, P. Boesiger, and S. Kozerke. High resolution three-dimensional cardiac perfusion imaging using compartment-based k-t principal component analysis. *Magnetic Resonance in Medicine*, 65:575–587, 2 2011.
- [141] T. Correia, T. Schneider, and A. Chiribiri. Model-based reconstruction for highly accelerated first-pass perfusion cardiac MRI. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 514–522. Springer, 2019.
- [142] L. Y. Hsu, M. Jacobs, M. Benovoy, A. D. Ta, H. M. Conn, S. Winkler, A. M. Greve, M. Y. Chen, S. M. Shanbhag, W. P. Bandettini, and A. E. Arai. Diagnostic performance of fully automated pixel-wise quantitative myocardial perfusion imaging by cardiovascular magnetic resonance. *JACC. Cardiovascular imaging*, 11:697–707, 5 2018.
- [143] P. Kellman, M. S. Hansen, S. Nielles-Vallespin, J. Nickander, R. Themudo, M. Ugander, and H. Xue. Myocardial perfusion cardiovascular magnetic resonance: optimized dual sequence and reconstruction for quantification. *Journal of Cardiovascular Magnetic Resonance*, 19:1–14, 4 2017.
- [144] C. S. Patlak, R. G. Blasberg, and J. D. Fenstermacher. Graphical evaluation of blood-to-brain transfer constants from multiple-time

- uptake data. *Journal of Cerebral Blood Flow & Metabolism*, 3(1):1–7, 1983.
- [145] R. M. Menchón-Lara, J. Royuela-del Val, F. Simmross-Wattenberg, P. Casaseca-de-la Higuera, M. Martín-Fernández, and C. Alberola-López. Fast 4d elastic group-wise image registration. convolutional interpolation revisited. *Computer Methods and Programs in Biomedicine*, 200:105812, 2021.