Learning to PostProcess: Diffeomorphic Image Registration with Matrix Exponential

by

Soumyadeep Pal

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science
University of Alberta

Abstract

Diffeomorphic image registration is important for medical imaging studies because of the properties like invertibility, smoothness of the transformation, and topology preservation/non-folding of the grid. Violation of these properties can lead to destruction of the neighbourhood and the connectivity of anatomical structures during image registration. Most of the recent deep learning methods do not explicitly address this folding problem and try to solve it with a smoothness regularization on the registration field.

We present a postprocessing layer for deformable image registration to make a registration field more diffeomorphic by encouraging Jacobians of the transformation to be positive. It is a differentiable layer which takes any registration field as its input, computes exponential of the Jacobian matrices of the input and reconstructs a new registration field from the exponentiated Jacobian matrices using Poisson reconstruction. Our proposed Poisson reconstruction loss enforces positive Jacobians for the final registration field. Thus, our method acts as a post-processing layer without any learnable parameters of its own and can be placed at the end of any deep learning pipeline to form an end-to-end learnable framework. We show the effectiveness of our proposed method for a popular deep learning registration method Voxelmorph and evaluate it with a dataset containing 3D brain MRI scans. Our results show that our post-processing can effectively decrease the number of non-positive Jacobians by a significant amount without any noticeable deterioration of the registration accuracy, thus making the registration field more diffeomorphic. There are also some practical limitations of our method which we demonstrate through experiments on 2D chest

X-ray registration - pointing towards more future work.

Our code is available online at https://github.com/Soumyadeep-Pal/Diffeomorphic-Image-Registration-Postprocess

Preface

Parts of thesis have been published as Soumyadeep Pal, Matthew Tennant, Nilanjan Ray "Towards Positive Jacobian: Learn to Postprocess for Diffeomorphic Image Registration with Matrix Exponential" at 2022 26th International Conference on Pattern Recognition (ICPR).

Acknowledgements

I would like to express my sincere gratitude to my advisor Dr. Nilanjan Ray for his support and encouragement throughout my M.Sc. study. I am extremely thankful for the several insightful and inspiring discussions I have had with him and I really appreciate his openness and support to explore different directions of research aligned with my interests. I have learnt a lot from him about research, scientific temper and life, in general. I would also like to express my sincere gratitude and appreciation for my advisor Dr. Matthew Tennant, who has graciously supported me over the years.

I am extremely grateful and thankful for my parents and brother. To my parents Mr.Basudeb Pal and Mrs.Sarmistha Pal who have raised me with love and have constantly supported and encouraged me in all my pursuits - Thank you and love you.

I would like to thank Dr. Lili Mou for his words of encouragement and his support during my M.Sc. to explore my different research interests. I have gained many research insights and lessons from our numerous discussions.

I am very thankful and grateful for my dear friend Ananya for her support and I am indebted to my friends for countless scientific, non-scientific discussions which has made life fun in general: Archit, Ananya, Shivam, Parash, Taivanbat.

Table of Contents

List of Tables

List of Figures

List of Symbols

Registration

$\mathcal{L}_{poisson}$ Poisson Reconstruction Loss

$\mathcal{L}_{reg}$ Regularization Loss

$\mathcal{L}_{sim}$ Similarity Loss (eg. SSD, NCC, NMI etc.)

Ω Image Domain

ϕ Displacement Field

ψ Deformation Field

F Fixed Image

M Moving Image

Math

$[.,.]$ Lie Bracket

$\mathcal{C}^0(X;Y)$ Set of all continuous mappings from X to Y

$\mathcal{C}^1(X;Y)$ Set of all once differentiable mappings from X to Y

$\overline{A}$ Closure of set A

∂A Boundary of set A

Chapter 1

Introduction

1.1 Deformable Image Registration

Image registration is a challenging task in computer vision which comprises of finding correspondence between two images and spatially aligning them such that those images match [1]. Typically one image is kept fixed and another image (moving image) is transformed for this matching process. Difference in such images may arise due to images being obtained at different times, from different acquisition devices, different perspectives or different acquisition modalities [1]. Based on the type of transformation associated with this registration problem, it can be classified into global transformations and local or non-rigid or deformable transformations. We delve into this division briefly in Section 2.1.

In this thesis, we focus only on deformable image registration. Deformable image registration is one of the fundamental tasks of medical image analysis that has been an active research topic for decades. It is extremely important for radiation therapy [2], morphometry analysis of highly anatomically variable brain MRI images [3, 4], diagnosis through multimodal medical image fusion [5] etc.

The transformation to align a pair of images is a dense, non-linear transformation. We illustrate deformable image registration through an example in Figure 1.1. As shown in Figure 1.1, our goal for deformable registration is to estimate a non-linear deformation field which is used to warp the moving image to match with the fixed

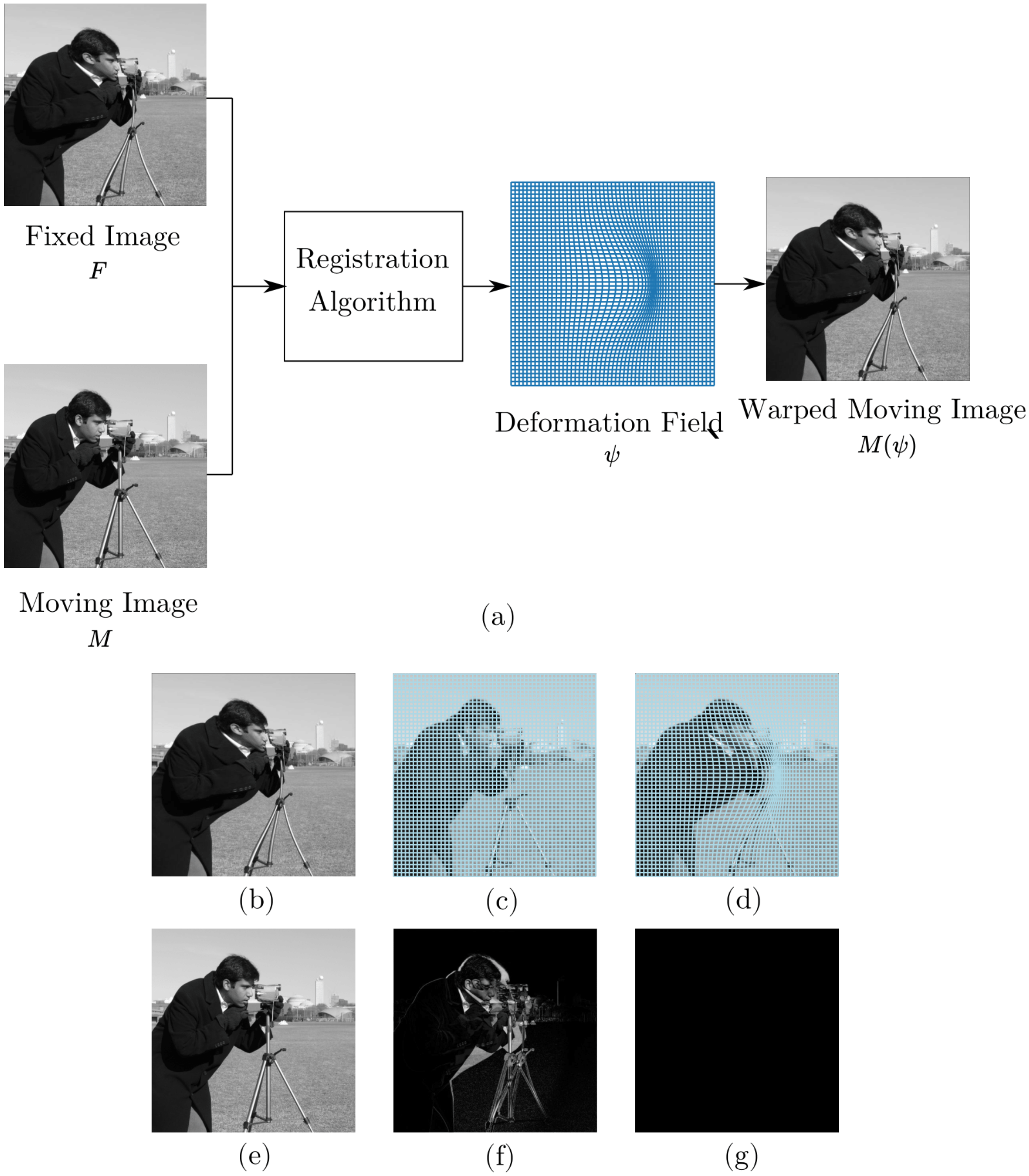


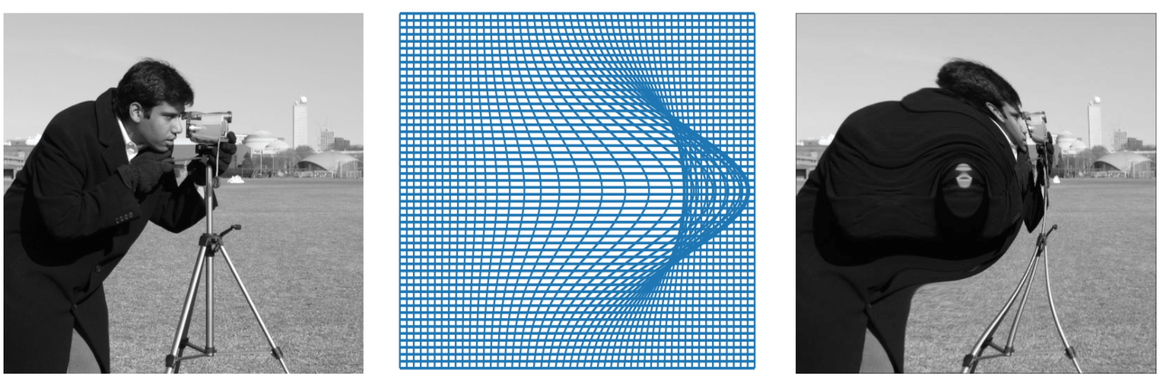
Figure 1.1: Deformable Image Registration: (a) Overview (b) Fixed Image F (c) Moving Image with Grid (d) Warped Moved Image with Grid (e) Moving Image M (f) $|F - M|$ (g) $|F - M(\psi)|$ *Note*: 1. This Cameraman Image is for ease of understanding, we will apply deformable image registration to brain MRI and chest X-ray images. 2. The 'Registration Algorithm' may be a classical algorithm or a deep learning pipeline (Section 2.3)

image. In Figure 1.1, we use a simple image for illustrating the concept. The lower rows contain the deformed images along with their grids providing insights into how a deformation grid can warp the moving image. As we observe in the figure, the difference between the fixed image and the warped image is zero denoting a perfect match. We formalise these notions and pose this as an optimisation problem in Section 2.2.

1.2 Diffeomorphic Image Registration

One of the desirable properties of the transformations for registration in medical imaging is their one-to-one nature or invertibility, which ensures that there is no folding in the grid. Folding of the registration grid over itself can lead to connected sets becoming disconnected and disconnected sets becoming connected thereby destroying the neighborhood structure that is detrimental for anatomical studies in medical imaging [6].

We give an example of such undesirable effects in Figure 1.2. The way to prevent this phenomenon is to ensure diffeomorphic transformations. We describe this in details in Section 2.2.2 and Section 2.2.3.



(a) (b) (c)

Figure 1.2: Undesirable effects of folding: (a) Moving Image (b) Deformation Grid with folding (c) Warped Moving Image

1.3 Challenges and Research Gap

Classical diffeomorphic registration algorithms often use strategies, which ensure smooth, invertible transformations. These traditional registration algorithms are often formulated as an optimization problem where a moving image is warped using a displacement field and the goal is to maximize the similarity between a fixed image and the warped moving image. This is usually solved using an iterative process, which is fairly computationally intensive and time consuming. Moreover, to ensure diffeomorphism, constraints were often applied on the deformation field which may make it more time and computation intensive. We give a brief overview of such algorithms in Section 2.3.1.

Recently, deep learning approaches have been used in solving the deformable registration problem. The advent of deep learning has led to the development of faster and more complex registration algorithms. These approaches maintain similar performance in terms of registration accuracy and are much faster. However, deep learning-based registration methods usually do not explicitly ensure invertibility and non-folding of the transformations. Such foldings in the deformations are usually constrained by enforcing spatial smoothness, which is controlled by a regularization hyperparameter. This is the regularization term $\mathcal{L}_{reg}$ in Equation 2.1. However, a large value of this hyperparameter can lead to inaccurate registration, while a small value can lead to folding and local errors, which makes it challenging to tune it.

Few efforts have been made to explicitly ensure that the transformations estimated using deep learning algorithms are diffeomorphic - this creates the premise for more work.

1.4 Thesis Contributions

In this thesis, we explicitly address the issue of folding with a postprocessing layer, which can be potentially inserted at the end of any registration pipeline giving a deformation field as its output. Our postprocessing step takes a deformation field as its input and provides another deformation with reduced foldings as its output with the help of matrix exponential and Poisson reconstruction. Moreover, this postprocessing layer is completely differentiable, hence it can fit in any deep learning pipeline for registration with end-to-end learning. We describe our proposed method in Chapter 3.

A significant application of deformable registration is the alignment of 3D brain magnetic resonance (MR) images for their analysis. Brain MR images can be acquired from different sensors, different subjects or at different times and thus are misaligned. Moreover, there is often a significant variability [7] between these scans due to different anatomical variations and health states. Deformable image registration is useful in this case for the purpose of comparing different anatomical structures in the brain scans obtained from different sources. A similar application of deformable registration is X-ray chest image analysis - applications being multi-atlas segmentation and pathology classification [8]. In all of these applications, addressing the folding problem is important to maintain anatomical plausibility.

We demonstrate the effectiveness of our method in Chapter 4 by the registration of 3D brain MR scans, which are obtained from [9]. We use our postprocessing layer with a widely used registration method named Voxelmorph [10] and show that it significantly reduces the amount of folding when compared to Voxelmorph.

We further test our method in Chapter 4 using 2D chest X-ray images, obtained from [11, 12]. In these experiments, we discover the limitation of our method - our method, though mathematically sound, can lead to instabilities and may be impractical for training in different settings.

Chapter 2

Image Registration

In this chapter, we discuss about image registration - specifically about deformable and diffeomorphic image registration. We provide a general overview on registration, diffeomorphisms and then discuss some related works. We introduce the classic image registration techniques and then move on to discuss about modern deep learning techniques in deformable image registration.

2.1 General Overview

Image registration entails estimating a mapping between a pair of images - a stationary or fixed image and the image which is spatially transformed called the moving image. The estimated mapping is essentially a mapping from the pixel / voxel locations of the moving image to corresponding locations in the fixed image. The moving image is warped according to this mapping (through interpolation or resampling) such that it matches with the fixed image.

According to the type of transformation estimated, image registration methods can be very broadly categorised into registration with global transformations and non-rigid or deformable registration. Global transformations for registration involves rotations, translations, shearing, scaling and projective transformations - the transformations for this category are applicable globally to the whole image. We demonstrate such global transformations in Figure 2.1 with increasing degrees of freedom along

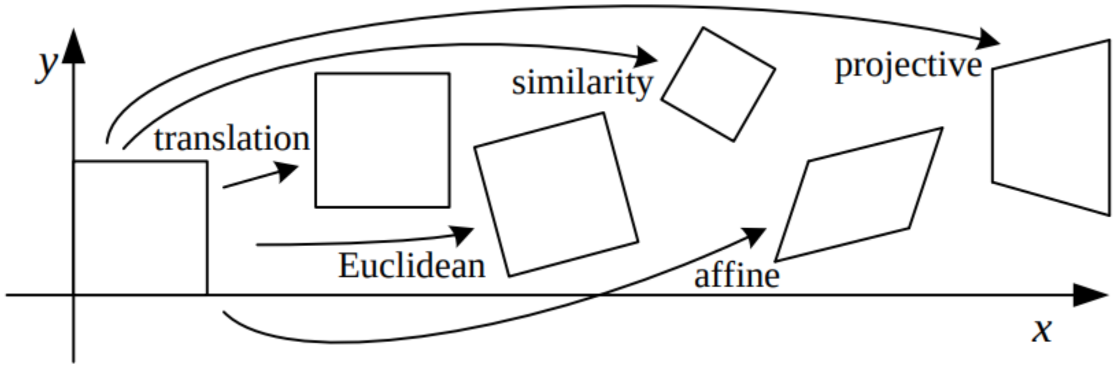


Figure 2.1: Global transformations in registration. Taken from [13]

the right of the x-axis.

In contrast to this, deformable image registration entails warping the moving image with a dense voxel-wise non-linear spatial transformation so that it matches with the fixed image. Thus it is a local operation.

Image registration is particularly useful in the context of medical imaging. Rigid registration is useful in aligning solid structures like bones. However, this does not apply for most of the human body. Non-rigid registration is of particular importance for deformation of soft tissues, neuroanatomy etc. A typical registration process involves a global alignment step between the fixed and moving image and then a deformable registration step. In this thesis, we mainly focus on deformable image registration.

2.2 Deformable Image Registration

We consider $\Omega \subseteq \mathbb{R}^n$ to be the image domain, such that $n = 2$ for 2D images and $n = 3$ for 3D images. Let F be the fixed image and M be the moving image, where $M, F : \Omega \to \mathbb{R}^d$. In this thesis, we mainly perform experiments on structural images for MRI, where $d = 1$. Formally, the goal of deformable registration is to estimate a transformation or deformation field $\psi : \Omega \to \Omega$ which is used to warp M to match with F.

This can be posed as an optimisation problem - the objective of deformable registration is to find an optimal deformation field:

$$\psi^* = \arg\min_{\psi} \mathcal{L}_{sim}(F, M(\psi)) + \lambda \mathcal{L}_{reg}(\psi) \tag{2.1}$$

where ψ^* is the optimal deformation field, $\mathcal{L}_{sim}$ is a dissimilarity loss function and $\mathcal{L}_{reg}$ is the function that enforces a smoothness regularization.

As seen in Equation 2.1, through the term $\mathcal{L}_{sim}$, we aim to decrease the dissimilarity or increase the similarity between the warped and the fixed image. Different such similarity measures have been used in the literature like sum of squared differences (SSD), sum of absolute differences (SAD), normalized cross-correlation (NCC) [14], mutual information (MI) [15] , normalized mutual information (NMI) [16], Normalized Gradient Fields (NGF) [17] etc. In our work, we perform our experiments with NCC as a dissimilarity measure - we describe it in details in Section 3.5.1.

2.2.1 Regularization

Regularization is of particular importance for the problem of registration. This is because image registration is an ill-posed problem, a concept introduced in mathematical physics by J.Hadamard [18]. A mathematical problem is considered Hadamard well-posed if it satisfies the following conditions [18]:

- a solution exists
- the solution is unique
- the solution's behaviour changes continuously with the initial conditions

Now, it is straightforward to realise that the solution of image registration, like most inverse problems is not unique. We give an example of such non-uniqueness phenomenon in Figure 2.2. We consider global transformations in this example for simplicity - this concept will still be valid for deformable registration.

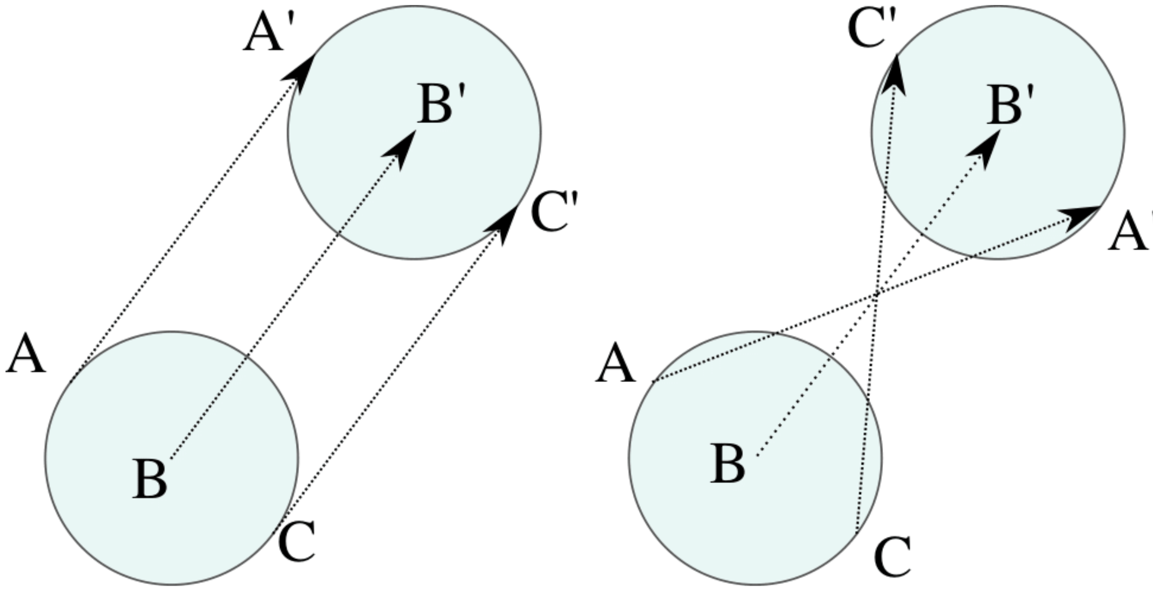


Figure 2.2: Importance of Regularization: We observe that different transformations lead to the same result. A', B', C' are the points in the transformed object (circle) corresponding to the points A, B, C in the moving object. Transformations - Left: Translation Right: Reflection about the line BB' and translation.

A solution for ill-posed solution is using regularization. Hence, different kinds of regularization methods are used in image registration. The regularization methods can be implicit or explicit. Explicit regularization comprises of including a form of penalty in the loss function as in Equation 2.1 or having the penalty condition as a hard constraint [19]. Implicit regularization may constrain the transformations for registration by parameterizing the deformations (eg. radial basis functions [20]) or by constraining the space of deformations (eg. in LDDMM [6]) . We describe such methods in Section 2.3.1.

2.2.2 Diffeomorphisms

In this thesis, we focus on developing diffeomorphic registration fields. Diffeormorphisms are important for image registration because of its different convenient properties. We motivate the importance of diffeomorphic registration in this section after defining diffeomorphisms.

Let Ω be an open set of $\mathbb{R}^d$. An open set U is essentially a set of elements such that for every element x in the set, all elements in a small neighbourhood of x also lies in U.

Definition 2.1 ([21] p.183) *A homeomorphism of* Ω *is a continuous bijection* $\Phi : \Omega \to \Omega$ *such that its inverse,* Φ^{-1}*, is continuous. A diffeomorphism of* Φ *is a continuously differentiable homeomorphism* $\Phi : \Omega \to \Omega$ *such that* Φ^{-1} *is continuously differentiable.*

Now, let us consider a diffeomorphic deformation $f : \Omega \to \Omega$ such that for each point $x \in \Omega$, $y = f(x)$. In such a setting, because of the bijective nature of diffeomorphism, two keys points are enforced that make it an appealing choice for image registration:

- Because the transformation is onto, for every $y \in \Omega$, there exists a corresponding $x \in \Omega$. Thus this prevents any *holes* in the output.
- Since f is one-to-one, for any $x_1, x_2 \in \Omega$, they do not have the same y, i.e. $f(x_1) \neq f(x_2)$. Thus this prevents *folding*. More precisely, if folding occurs, due to crossing of deformation fields, two different points x_1 and x_2 will be mapped to the same y.

2.2.3 Diffeomorphisms and Orientability

The term orientation-preserving is used to denote mappings whose Jacobian determinants are positive. Precisely, a mapping $\Phi \in \mathcal{C}^1(\overline{\Omega}; \mathbb{R}^3)$ is an orientation preserving map if it satisfies $\det \nabla\Phi(x) > 0$ for all $x \in \overline{\Omega}$ [22]. Under cetain boundary conditions, these orientation preserving mappings are shown to be diffeomorphic.

Theorem 2.1 (adapted from Ciarlet. p.225 [22]) *Let* Ω *be a bounded open connected subset of* $\mathbb{R}^n$*. Let* $\Phi_0 \in \mathcal{C}^0(\overline{\Omega}; \mathbb{R}^n)$ *be an injective mapping and let* $\Phi \in \mathcal{C}^0(\overline{\Omega}; \mathbb{R}^n) \cap \mathcal{C}^1(\Omega; \mathbb{R}^n)$ *be a mapping that satisfies:*

$$\det \nabla\Phi(x) > 0 \text{ for all } x \in \Omega$$

$$\Phi(x) = \Phi_0(x) \text{ for all } x \in \partial\Omega$$

then the mapping $\Phi : \overline{\Omega} \to \Phi(\overline{\Omega})$ *is a homeomorphism and* $\Phi : \Omega \to \Phi(\Omega)$ *is a* $\mathcal{C}^1$ *diffeomorphism.*

In this theorem, $\mathcal{C}^0(\overline{\Omega}; \mathbb{R}^n)$ denotes the set of all continuous mappings from $\overline{\Omega}$ to $\mathbb{R}^n$, while $\mathcal{C}^1(\Omega; \mathbb{R}^n)$ denotes the set of all once diffentiable continuous mappings from Ω to $\mathbb{R}^n$.

Simply put, for a continuous mapping to be diffeomorphism, the theorem poses a condition on the determinant of the jacobian of the mapping being positive and on the mapping to be injective at the boundaries of the input open set.

2.3 Related Works

2.3.1 Classical Registration

Classical registration algorithms often model transformations as a physical model. We describe some of the main classical approaches to deformable registration.

- In this type of registration, the image to be deformed is often considered to be an elastic body [23]. The elastic body is deformed by the application of two forces: an external force that ensures that the similarity measure between the fixed and moving image is achieved and an internal force that counteracts any movement due to the elasticity properties of the body. These forces compete against each other until an equilibrium is reached. The image is thus considered to be an elastic membrane whose movements are governed by the Navier-Cauchy PDEs. We describe a similar framework based on fluid flow models in more details in the next group of registration algorithms. The basic formulation of registration considering elastic bodies has been extended in various studies [24, 25].

  One of the main drawbacks of this model is that the elastic constraint is suitable only for small deformations and also that it resulted in inverse-inconsistent transformations. Inverse consistency implies that when F and M in Equation 2.1 are interchanged, the resultant deformation field is the inverse of the former field ϕ. The inverse consistency problem was tackled with the elastic body framework in [26, 27]. Large deformations were modeled by considering a sequence of small deformations using linear elastic models in [28].

- The second catergory of physical models used in registration is the viscous fluid flow model. Here the image is considered to be a viscous fluid - thus the displacements for image registration would follow laws governing the flow of viscous fluids. This was explored in [29], where the displacement flow was modeled using the Navier-Stokes Equation. Unlike the previously described elastic models, this allowed for large deformations while maintaining a homeomorphic mapping. The fluid flow is governed by the following Navier-Stokes equation:

$$\mu\nabla^2\mathbf{v} + (\mu + \lambda)\nabla(\nabla \cdot \mathbf{v}) + b(u) = 0 \tag{2.2}$$

In this PDE, λ and μ denote the fluid viscosity terms. The first term is associated with the constant volume viscous flow of the moving image. The second term in these equations denote the accelaration of the flow in space and thus denotes the local growth or shrinkage of the moving image. The third term denotes the body force on the fluid which is determined by the mismatch between the fixed image and the warped moving image. The PDE is solved using iterative successive overrelaxation - this is stopped when the body force reaches a certain threshold i.e. there is a good match in the fixed and warped moving image. This fluid flow framework was explored more in future work in terms of multiresolution [30] and increasing efficiency in computation [31, 32].

- One of the seminal methods of classical deformable registration comprises of the Demons approaches to registration. This is inspired by Maxwell's demons which is a paradox in the second law of thermodynamics that is ensued by the presence of demons or effectors that allow the travel of certain kind of gaseous molecules from one side of a membrane to other. Hence, this essentially is a process of diffusion. Similar to this phenomenon, this registration approach places demons along objects in the moving image and the moving image is deformed according to the forces associated with the demons.

Concisely, two iterative steps are followed [33]: (a) Estimating the forces associated with the demons (b) Computing the transformation of the deformable field based on such forces. Different choices associated with this process can give rise to different methods - choices being available in selection of demon positions, space of deformations, formula estimating the force of a demon.

This pioneering work was studied and extended based on different optimization [34, 35] and regularization [36, 37] schemes and different similarity measures [38, 39].

- A large body of registration methods use parameterization of displacement fields where a linear parameterization using certain basis functions are used. This presents a form of implicit regularization in the arrangement itself. Considering the displacement for registration to be ϕ, these methods take the general form:

$$\phi(x) = \sum_k p_k B_k(x) \tag{2.3}$$

where $p_k \in \mathbb{R}^d$ are the optimizable parameters and $B_k : \Omega \to \mathbb{R}$ are the basis functions.

Some of the most important families of basis functions include radial basis functions [20], thin plate splines [40, 41], elastic body splines [42], Fourier bases [26], cosine basis [43] etc.

While the above mentioned methods have been successful in image registration, they do not guarantee diffeomorphic transformations. Hence, we consider a body of literature which focuses on diffeomorphic image registration and briefly describe such works as follows:

- One of the main approaches to forming diffeomorphic registration is to consider flows of velocities: a seminal work along this line being the LDDMM framework [6]. The diffeomorphic transformations $\psi : \Omega \to \Omega$ are obtained by integrating velocity fields over time. Considering the velocity fields to be $v : [0,1] \times \Omega \to \mathbb{R}^d$, a transformation ψ is formulated as follows:

$$\begin{aligned} \frac{\partial \varphi(t,x)}{\partial t} &= v(t, \varphi(t,x)) \\ \varphi(0,x) &= x \ \forall \ x \in \Omega \end{aligned} \tag{2.4}$$

  Equation 2.4 has a unique solution for ψ under necessary smoothness conditions for the velocity field [44] - velocity fields are regularized based on Sobolev embedding theory [45]. Thus the ODE is solved till time $t = 1$ to obtain the final deformation field $\psi(x) = \varphi(1, x)$. The solution $\varphi(t, x)$ obtained on solving the above ODE is called the associated flow [46] of the velocity field v, which is a diffeomorphism at all time points t [46](Theorem C.10). Using the constraints of Equation 2.4, the diffeomorphic field is obtained by an optimization similar to Equation 2.1. This is finally solved through two classes of methods namely - relaxation [6] and shooting [47].

- One disadvantage of the LDDMM framework is the high computation cost and the memory complexity associated with the integration of velocity fields. This is combated by the framework DARTEL [48] which considered a stationary i.e. a constant velocity field $v : \Omega \to \mathbb{R}^d$. The final flow associated with the velocity field was estimated using the scaling and squaring approach [49] - this results in an exponentiated flow field thus making the mapping diffeomorphic. The optimization was solved using the Levenberg-Marquardt method [50]. Another advantage of this method is that the inverse transformation can be easily computed simply by backward integration of the negative velocity field -v using scaling and squaring.

- Other diffeomorphic registration approaches are based on physical models and constrained optimization approaches. In [51], the deformable template were modeled as hyperelastic or non-linear elastic membranes and the registration was performed using non-linear finite element approach [51]. Further development involving hyperelastic regularization for diffeomorphic registration were done in [52–55]. Other works apply a volume incompressibility constraint i.e. penalizing expansion and contraction. Rohlfing *et al.* [56] introduce a incompressibility based penalty term via the log of the determinant of the Jacobian while Haber and Modersitzki [57] solve the equivalent KKT conditions with a hard constraint of Jacobian determinant being 1.

2.3.2 Deep Learning Based Registration

Due to the advent of deep learning, recent works have used convolutional neural networks (CNN) to perform registration. These methods can be broadly divided into supervised and unsupervised ones.

Supervised deep learning methods for deformable registration usually learn a deep learning model estimating displacement fields using ground truth images ([58–62]). However, this may result in transformations biased by ground truths and in practical problems, it is very difficult to obtain a large amount of ground-truth information. Thus, unsupervised deep learning approaches have also been developed, which is the focus of this thesis.

In this section, we describe unsupervised approaches using deep learning for image registration. The advent of Spatial Transformer Networks [63] allowed the end-to-end training of registration by optimizing a similarity metric between a fixed image and a warped image.

Vos *et al.* [64] presented a deformable image registration network (DIRnet) - one of the first end-to-end unsupervised deformable registration framework using deep learning. In this work, the deformation field was parameterized using cubic B-splines [65]. Patches extracted from the moving and fixed image were passed into a convolutional neural network regressor which gave the control points of the cubic spline as its output. These control points were used by the spatial transformer [63] for interpolation and warping of the moving image.

Li and Fan [66] developed on this basic idea by performing unsupervised registration on the full spatial dimension of the image. This was possible by the use of fully convolutional networks (FCN)[67] which were used to learn the full deformation field with the fixed and moving image as its input. This was trained using NCC as the similarity metric and a total variation based regularizer [68].

However, these pioneering methods had certain limitations which were overcome by [10]. Vos *et al.* [64] had a implicit regularization scheme, Li and Fan [66] demonstrated their effectiveness on 3D subregions. Balakrishnan *et al.* [10] developed the Voxelmorph (VM) framework where they proposed a UNet [69] based CNN which predicts the deformation field given an input pair of fixed and moving image. The application of spatial transformation in this context is described in Section 4.3.3. A more extensive analysis of VoxelMorph on different registration settings and a vast collection of datasets is performed in [70]. In this thesis, we consider VoxelMorph as our primary DL pipeline for unsupervised image registration.

In [71], registration is performed with a multiscale and multiresolution approach. Here CNNs are trained to obtain a coarse to fine registration field using a B-Spline deformation model. Other related works for unsupervised image registration include [72–74].

These deep learning methods do not typically consider the diffeomorphic properties of the registration field. There have been few works that look into learning diffeomorphic registration fields.

- The first group of works primarily use the scaling and squaring algorithm for producing diffeomorphisms. Dalca *et al.* [75] use a probabilistic generative CNN model to estimate the mean and covariance of a stationary velocity field. A diffeomorphic deformation field is obtained from the velocity field by scaling and squaring layers. Their variational inference formulation also assist in giving an uncertainty estimate of the deformation field. This work was extended in [76] with more analysis, experiments and results along with an extension of their method to surface registration. Diffeomorphic deformation fields are also obtained by similar scaling and squaring of SVFs (stationary vector field) found via conditional variational autoencoders [77, 78]. Mok and Chung [79] use a similar scaling and squaring theme, but also develop a symmetric similarity framework considering both the forward and backward transformation between a pair of images.

- The second group of works employ different kinds of regularization and losses to maintain diffeomorphism. Zhang [80] introduced an inverse consistency loss that penalizes the transformations from the respective inverse mappings along with an anti-folding constraint that penalizes gradient of the flow along a point of folding. Kuang and Schmah [81] use a penalty loss to constrain negative Jacobians and train their CNN with a use cross correlation similarity loss. Kim *et al.* [82] ensure cycle or inverse consistency using two neural networks - one network approximating the transformation from moving to fixed and the other from the fixed to moving image.

- The moving mesh grid parameterization is a technique that leads to diffeomorphic image registration [83, 84]. This method does not rely on any regularization - rather it constrains the Jacobian determinant using a monitor function and solves a divergence curl system for obtaining a diffeomorphic mapping [85]. Sheikhjafari *et al.* [86] trained a UNet which outputs the monitor function for constraining the Jacobian. Then the div-curl system is solved and this is trained in an unsupervised manner to obtain a deep learning approach to the moving mesh method.

In this thesis, we focus along this line of work which develops diffeomorphic registration fields using unsupervised end-to-end neural network training. We develop a postprocessing layer that can reduce the number of non-positive Jacobians and can potentially fit in any deep learning registration pipeline.

Chapter 3

Postprocess Diffeomorphic Image Registration with Matrix Exponential

In this chapter, we discuss the proposed method for obtaining diffeomorphic image registration using matrix exponential. We describe our method in the case of 3D image volumes. This can also be applied to 2D.

Let F, M be two 3D image volumes. As mentioned before, the two image volumes are initially aligned with a global deformation as a preprocessing step such that the remaining misalignment between F and M is non-linear.

A function $g_\theta(F, M)$ is modelled by a convolutional neural network (CNN), such that the neural network outputs a displacement field ϕ [10]. The displacement field is a four dimensional vector that determines the displacement between F and M. Considering Id as an identity transform, the transformaton $\psi = Id + \phi$ is used to warp the moving image, such that for each voxel location q, $F(q)$ and $M(\phi(q))$ are identical.

For our experiments, we use the VoxelMorph-2 architecture [10] as our neural network. We insert a postprocessing layer at the end of the neural network, which takes in the displacement field ϕ and gives a new displacement field ϕ_p as an output, which potentially contains much less folding when compared to ϕ.

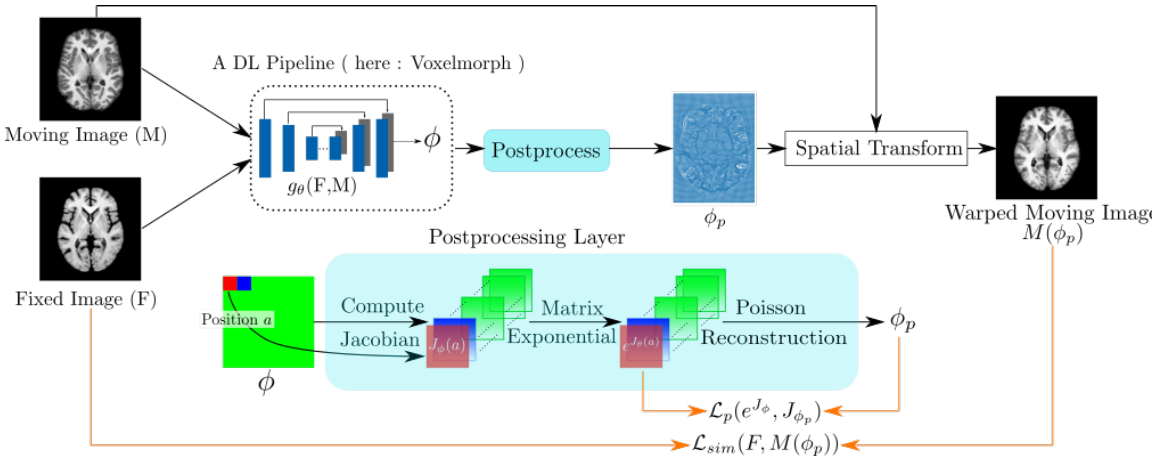


Figure 3.1: Overview of the proposed postprocessing layer for Diffeomorphic Image Registration. The output from the registration pipeline ϕ is the input to out proposed layer which gives ϕ_p as its output. J_ϕ indicate the Jacobian matrices of a displacement field ϕ. We omit the regularization loss for simplicity

3.1 Postprocessing Formulation

The postprocessing layer comprises of 3 steps as shown in Figure 3.1:

- Compute the Jacobian matrices at each voxel for displacement field ϕ
- Compute the Matrix Exponential of each Jacobian matrix
- Reconstruct the postprocessed displacement field ϕ_p by solving Poisson equations

Concretely, let $\phi, \phi_p \in \mathbb{R}^{H\times D\times W\times 3}$ be displacement fields such that $\phi = [\phi_x, \phi_y, \phi_z]$ and $\phi_p = [\phi_{p_x}, \phi_{p_y}, \phi_{p_z}]$. $\phi_i, \phi_{p_i} \in \mathbb{R}^{H\times D\times W}$ denotes the component of the displacement fields along the i-th axis. Let $q = (x, y, z)$ be a voxel location. H, D, W refer to height, width and depth of the image volumes respectively.

The Jacobian matrix of the displacement field $\phi = (\phi_x, \phi_y, \phi_z)$ at q is defined as:

$$Jac(\phi(q)) = \begin{bmatrix} \nabla\phi_x(q) \\ \nabla\phi_y(q) \\ \nabla\phi_z(q) \end{bmatrix} = \begin{bmatrix} \frac{\partial\phi_x(q)}{\partial x} & \frac{\partial\phi_x(q)}{\partial y} & \frac{\partial\phi_x(q)}{\partial z} \\ \frac{\partial\phi_y(q)}{\partial x} & \frac{\partial\phi_y(q)}{\partial y} & \frac{\partial\phi_y(q)}{\partial z} \\ \frac{\partial\phi_z(q)}{\partial x} & \frac{\partial\phi_z(q)}{\partial y} & \frac{\partial\phi_z(q)}{\partial z} \end{bmatrix} \tag{3.1}$$

Thus we have one 3×3 Jacobian matrix for each voxel location in the image volume, resulting in $H \times D \times W$ matrices. We compute the matrix exponential of each of these matrices. Thus for voxel location q, we get the matrix $J'(q)$:

$$J'(q) = \begin{bmatrix} J'_x(q) \\ J'_y(q) \\ J'_z(q) \end{bmatrix} = e^{Jac(\phi(q))} \tag{3.2}$$

We compute the matrix exponential of the 3×3 matrix $Jac(\phi(q))$ by a series summation scheme as follows:

$$e^{Jac(\phi(q))} = \sum_{n=0}^{\infty} \frac{(Jac(\phi(q)))^n}{n!} \tag{3.3}$$

This scheme is in line with [87]. We truncate the series after 20 terms and observe that is good enough to give accurate results.

We observe that at a particular voxel location q, the Laplacian of the postprocessed displacement field is the same as the divergence of the Jacobian at that point. Considering $J'_x(q) = [J'_{x_1}(q), J'_{x_2}(q), J'_{x_3}(q)]$, we have:

$$\Delta\phi_{p_x}(q) = \frac{\partial J'_{x_1}(q)}{\partial x} + \frac{\partial J'_{x_2}(q)}{\partial y} + \frac{\partial J'_{x_3}(q)}{\partial z} = \nabla \cdot J'_x(q) \tag{3.4}$$

Thus, the final postprocessed displacement field is reconstructed from J' by solving the following Poisson equations:

$$\begin{aligned} \Delta\phi_{p_x} &= \nabla \cdot J'_x \\ \Delta\phi_{p_y} &= \nabla \cdot J'_y \\ \Delta\phi_{p_z} &= \nabla \cdot J'_z \end{aligned} \tag{3.5}$$

3.2 Solving the Poisson Equations

We solve the Poisson equations using a fast FFT solver adapting from [88]. In this subsection, we provide a comprehensive overview of the relevant numerical steps and their justification.

3.2.1 Linear System from discretizing Poisson Equations

Let us consider the first equation in Equation 3.5. The other two equations pertaining to the y-axis and z-axis can be solved similarly. The first equation can be written as follows:

$$
\begin{aligned}
&\Delta\phi_{p_x} = \nabla \cdot J'_x \\
\Rightarrow &\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right)\phi_{p_x} = \nabla \cdot J'_x \\
\Rightarrow &\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right)\phi_{p_x} = \frac{\partial J'_{x_1}}{\partial x} + \frac{\partial J'_{x_2}}{\partial y} + \frac{\partial J'_{x_3}}{\partial z}
\end{aligned}
\tag{3.6}
$$

We can compute the right hand side of the above Equation because J'_x is known from Equation 3.2.

We consider the solution of the Poisson Equations on a rectangular grid of dimension H × W × D, corresponding to the image volumes. Because of the rectangular mesh arrangement, we can consider discrete Laplacian using finite differences for the left hand side of Equation 3.6.

For the rectangular mesh, we take second order differences along x, y and z direction for the discrete Laplacian. Specifically, these differences are of the form -1,2,1 along each direction, which combine to form a ”7 point molecule”. Considering the finite differences around a voxel $q = (i, j, k)$, we have the following discrete Poisson equation:

$$
\begin{aligned}
6\phi_{p_x}(i,j,k) - \phi_{p_x}(i-1,j,k) - \phi_{p_x}(i+1,j,k) - \phi_{p_x}(i,j-1,k) \\
-\phi_{p_x}(i,j+1,k) - \phi_{p_x}(i,j,k-1) - \phi_{p_x}(i,j,k+1) = \nabla \cdot J'_x(i,j,k)
\end{aligned}
\tag{3.7}
$$

Now, we recall that ϕ_{p_x} is a grid of size H × W × D. Excluding the boundaries, we convert ϕ_{p_x} into a vector $\mathbf{U}$. Similarly, $\nabla \cdot J'_x$ is vectorized as $\mathbf{F}$. Thus we get the following linear system:

$$
\begin{gathered}
(\mathrm{K3D})\mathbf{U} = \mathbf{F} \\
\mathbf{U}, \mathbf{F} \in \mathbb{R}^M, M = (\mathrm{H}-2)\times(\mathrm{W}-2)\times(\mathrm{D}-2)
\end{gathered}
\tag{3.8}
$$

where K3D is a matrix of size $M \times M$ and Equation 3.7 corresponds to one row of this linear equation.

3.2.2 Eigenvalues and Eigenvectors of K3D

The matrix K3D has a special structure - it is a sparse matrix with 7 non-zero entries which are six -1's and a +6 according to Equation 3.7. This $M \times M$ matrix has M eigenvectors and corresponding M eigenvalues:

$$(\text{K3D})y_{lmn} = \lambda_{lmn}y_{lmn}$$

$$l = 1, ..., H', m = 1, ..., W', n = 1, ..., D'$$

$$H' = H - 2, W' = W - 2, D' = D - 2$$

where y_{lmn} is an eigenvector and λ_{lmn} are the eigenvalues.

The eigenvectors and eigenvalues of K3D are as follows:

$$(i, j, k) \text{ component of } y_{lmn} = y_{lmn}^{ijk} == \sin \frac{il\pi}{H' + 1} \sin \frac{jm\pi}{W' + 1} \sin \frac{kn\pi}{D' + 1}$$
$$\lambda_{lmn} = (2 - 2\cos \frac{l\pi}{H' + 1}) + (2 - 2\cos \frac{m\pi}{D' + 1}) + (2 - 2\cos \frac{n\pi}{W' + 1})$$

We can verify the eigenvectors and eigenvalues of the matrix due to the special sparse structure of the matrix K3D.

3.2.3 Fast Solver using DST

We recall that $\mathbf{F}$ is the vectorized form of $\nabla \cdot J_x'$ which is a 3D array of size $(\text{H} - 2) \times (\text{W} - 2) \times (\text{D} - 2)$. Thus we can index $\mathbf{F}$ as $\mathbf{F_{ijk}}$ which denotes the $(i, j, k)^{th}$ element of $\nabla \cdot J_x'$. With this premise, we perform the 3D Discrete Sine Transform of $\nabla \cdot J_x'$ to get the following:

$$\mathbf{F_{ijk}} = \sum_{l=1}^{H'} \sum_{m=1}^{W'} \sum_{n=1}^{D'} a_{lmn} \sin \frac{il\pi}{H' + 1} \sin \frac{jm\pi}{W' + 1} \sin \frac{kn\pi}{D' + 1} \tag{3.9}$$

We observe that the sin terms of Equation 3.9 together also resemble the (i, j, k) component of the eigenvector of the K3D matrix. Thus from Equation 3.9:

$$\begin{aligned} \mathbf{F_{ijk}} &= \sum_{l=1}^{H'} \sum_{m=1}^{W'} \sum_{n=1}^{D'} a_{lmn} y_{lmn}^{ijk} \\ \Longrightarrow \mathbf{F} &= \sum_{l=1}^{H'} \sum_{m=1}^{W'} \sum_{n=1}^{D'} a_{lmn} y_{lmn} \end{aligned} \tag{3.10}$$

Moreover, by the property of eigenvalues, we have

$$(\text{K3D})^{-1} y_{lmn} = \frac{1}{\lambda_{lmn}} y_{lmn} \tag{3.11}$$

Our goal here is to compute the vector $\mathbf{U}$ which is equivalent to the ϕ_{p_x}. We also note that the eigenvalues of the K3D matrix is always greater than zero - thus the matrix is always invertible. So, from Equation 3.8 we have:

$$\begin{aligned}
\mathbf{U} &= (\text{K3D})^{-1}\mathbf{F} \\
&= (\text{K3D})^{-1} \sum_{l=1}^{H'} \sum_{m=1}^{W'} \sum_{n=1}^{D'} a_{lmn} y_{lmn} \quad \text{[from Equation 3.15]} \\
&= \sum_{l=1}^{H'} \sum_{m=1}^{W'} \sum_{n=1}^{D'} a_{lmn} (\text{K3D})^{-1} y_{lmn} \\
&= \sum_{l=1}^{H'} \sum_{m=1}^{W'} \sum_{n=1}^{D'} \frac{a_{lmn}}{\lambda_{lmn}} y_{lmn} \ \text{[from Equation 3.17]}
\end{aligned} \tag{3.12}$$

Thus we can compute a_{lmn} using a Discrete Sine Transform of $\nabla \cdot J'_x$. y_{lmn} and λ_{lmn} can be obtained from the eigenvector and eigenvalue formulation of the K3D matrix. This is fast because we can go from the frequency space of $\mathbf{F}$ to that of $\mathbf{U}$ by a simple division of scalar values of λ_{lmn}.

3.3 Postprocess Analysis

In our proposed postprocessing formulation, we exploit the properties of matrix exponential and Poisson reconstruct to attain a reduced number of non-positive Jacobians.

For any complex square matrix A, the following identity holds true [89, p.41]:

$$\det(e^A) = e^{tr(A)} \tag{3.13}$$

where $tr(A)$ represents the trace of A. Since we compute J' through matrix exponential of the Jacobian matrices as in Equation 3.2, using the above identity:

$$\det(J'(q)) = \det(e^{Jac(\phi(q))}) = e^{tr(Jac(\phi(q)))} > 0 \tag{3.14}$$

Thus, if the matrices J' are valid Jacobian matrices, then a perfect reconstruction of the field results in a postprocessed field with positive Jacobian matrices in all voxel locations. However, it is not always certain that the exponentiated matrices will be valid Jacobian matrices and thus integrable. Our framework can combat this issue because of the variational properties of the Poisson equation. In the following section, we investigate the conditions required for perfect integrability and also the relevant properties of the Poisson equation that help our framework to reduce non-positive Jacobians.

3.4 Integrability of exponentiated Jacobian matrices J'

A Jacobian matrix for a displacement field (like ϕ or ϕ_p) consists of three rows of vector fields which are the gradients of the x,y and z components of the corresponding displacement field. For example, from Equation 3.1, the Jacobian matrix for the displacement field ϕ is made of three rows which are gradients of ϕ_x, ϕ_y and ϕ_z. Thus, for J' to be integrable, we need J'_x, J'_y and J'_z to be vector fields which are valid gradients of a function.

Theorem 3.1 (Hubbard, 2015. p.571 [90]) *If $U \in \mathbb{R}^3$ is convex, and $\vec{F}$ is a vector field on U, then $\vec{F}$ is the gradient of a function f defined on U if and only if $curl\vec{F} = 0$.*

In our case, the domain of the vector field is the image domain Ω, which essentially lies in a solid parallelopiped. Hence it is a convex set. Thus the vector fields J'_x, J'_y and J'_z need to have the following conditions for them to be valid gradients, resulting in J' to be integrable:

$$\begin{aligned}
\nabla \times J'_x &= 0 \\
\nabla \times J'_y &= 0 \\
\nabla \times J'_z &= 0
\end{aligned} \tag{3.15}$$

Hence we need the above special structure on the exponentiated matrices for perfect integrability, which will result in a reconstructed displacement field with theoretically guaranteed positive Jacobians. However, as we will see in Section 3.4.2, satisfying Equation 3.15, though possible, is non-trivial. However, we can still achieve a reconstructed displacement field which is close to a field with theoretically guaranteed positive Jacobians due to the properties of Poisson equations.

3.4.1 Integrability and Poisson Equations

We get the final postprocessed displacement by solving the Poisson equations given by Equation 3.5 and computing the matrix exponential of the Jacobian matrices of the starting displacement field, ϕ. We consider the variational formulation of the Poisson equation in Equation 3.5 using the Euler-Lagrange equations.

We consider the first equation of Equation 3.5. The solution of the following minimization problem:

$$\min_{\phi_{p_x}} \int_{\Omega} ||\nabla \phi_{p_x} - J'_x||_2^2 \text{ with } \phi_{p_x}|_{\partial\Omega} = \phi^*_{p_x}|_{\partial\Omega} \tag{3.16}$$

is the unique solution of the Poisson equation with the Dirichlet boundary condition [91]:

$$\Delta \phi_{p_x} = \nabla \cdot J'_x \text{ with } \phi_{p_x}|_{\partial\Omega} = \phi^*_{p_x}|_{\partial\Omega}$$

Here $\partial\Omega$ denotes the boundary of the set Ω which is our image domain. Thus, the solution of the Poisson equations in Equation 3.5 signifies minimizing the difference between the exponentiated Jacobian and the Jacobian of the resultant field ϕ_{p_x}.

3.4.2 Condition for perfect Integrability

In this section, we derive the conditions on the initial displacement field ϕ, which results in J' to be an integrable Jacobian i.e. the Jacobian of conservative vector fields.

Lemma 3.2 *If $YX = ZY$, where X, Y, $Z \in \mathbb{R}^{d\times d}$ are square matrices, the following are true:*

- $[X, Y] = (X - Z)Y = A_1Y$
- $[X, [X, Y]] = (X^2 - 2XZ + Z^2)Y = A_2Y$
- $[X, [\underbrace{X, ..., [X}_{\text{(n-2) times}}, [X, Y]]...]] = \left(\sum_{r=0}^{n} \binom{n}{r} X^{n-r}(-Z)^r\right) Y = A_nY$

for $n = 2, 3, ...$ Here, $[.,.]$ denotes the Lie bracket and we consider $A_n = \sum_{r=0}^{n} \binom{n}{r} X^{n-r}(-Z)^r$

Proof.

- $[X, Y] = XY - YX = XY - ZY = (X - Z)Y = A_1Y$
- $[X, [X, Y]] = [X, AY] = XAY - AYX = XAY - AZY = (XA - AZ)Y = (X(X - Z) - (X - Z)Z)Y = (X^2 - 2XZ + Z^2)Y = A_2Y$
- We will prove the general case by induction. We consider the induction hypothesis to be the given statement i.e. the predicate $P(n)$ is the third statement of the lemma.

  Base Case. We consider the base case when $n = 2$. Then

$$
\begin{aligned}
[X, [\underbrace{X, ..., [X}_{\text{(2-2) times}}, [X, Y]]...]] &= [X, [X, Y]] \\
&= (X^2 - 2XZ + Z^2)Y \\
&= \sum_{r=0}^{2} \binom{2}{r} X^{2-r}(-Z)^r
\end{aligned}
$$

Inductive Step. We assume $P(n)$ to be true. Our proof will be complete if

$P(n+1)$ is true.

$$
\begin{aligned}
&[X,[\underbrace{X,...,[X}_{\text{((n+1)-2) times}},[X,Y]]...]] \\
&=[X,[X,[\underbrace{X,...,[X}_{\text{((n-2) times}},[X,Y]]...]]] \\
&=[X,\left(\sum_{r=0}^{n}\binom{n}{r}X^{n-r}(-Z)^{r}\right)Y] \\
&=X\left(\sum_{r=0}^{n}\binom{n}{r}X^{n-r}(-Z)^{r}\right)Y-\left(\sum_{r=0}^{n}\binom{n}{r}X^{n-r}(-Z)^{r}\right)YX \\
&=X\left(\sum_{r=0}^{n}\binom{n}{r}X^{n-r}(-Z)^{r}\right)Y-\left(\sum_{r=0}^{n}\binom{n}{r}X^{n-r}(-Z)^{r}\right)ZY \\
&=\left(\sum_{r=0}^{n}\binom{n}{r}X^{n+1-r}(-Z)^{r}-\sum_{r=0}^{n}\binom{n}{r}X^{n-r}(-Z)^{r}Z\right)Y \\
&=\left(\sum_{r=0}^{n}\binom{n}{r}X^{n+1-r}(-Z)^{r}+\sum_{r=0}^{n}\binom{n}{r}X^{n-r}(-Z)^{r+1}\right)Y \\
&=\left(\sum_{r=0}^{n+1}\binom{n+1}{r}X^{n+1-r}(-Z)^{r}\right)Y
\end{aligned}
$$

which proves $P(n+1)$. So it follows by induction that $P(n)$ is true for all $n \geq 2$. ■

Proposition 3.3 *If J is a Jacobian matrix of a vector field, then e^J will also be a Jacobian matrix of a vector field, when there is a 3×3 matrix A, such that $\frac{\partial J}{\partial x}J = A\frac{\partial J}{\partial x}$, $\frac{\partial J}{\partial y}J = A\frac{\partial J}{\partial y}$, and $\frac{\partial J}{\partial z}J = A\frac{\partial J}{\partial z}$.*

Proof.

Let J be the Jacobian of the vector field $\phi = [\phi_x, \phi_y, \phi_z]$. Rewriting Equation 3.1, Jacobian at voxel location q:

$$
J(q) = \begin{bmatrix} \frac{\partial\phi_x(q)}{\partial x} & \frac{\partial\phi_x(q)}{\partial y} & \frac{\partial\phi_x(q)}{\partial z} \\ \frac{\partial\phi_y(q)}{\partial x} & \frac{\partial\phi_y(q)}{\partial y} & \frac{\partial\phi_y(q)}{\partial z} \\ \frac{\partial\phi_z(q)}{\partial x} & \frac{\partial\phi_z(q)}{\partial y} & \frac{\partial\phi_z(q)}{\partial z} \end{bmatrix} \tag{3.17}
$$

Let us consider:

$$
e^{J(q)} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} F_{x1} & F_{x2} & F_{x3} \\ F_{y1} & F_{y2} & F_{y3} \\ F_{z1} & F_{z2} & F_{z3} \end{bmatrix} \tag{3.18}
$$

For e^J to be a valid Jacobian matrix, from Equation 3.15, we need the following to hold true:

$$\begin{aligned}
\nabla \times F_x &= \begin{vmatrix} i & j & k \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ F_{x1} & F_{x2} & F_{x3} \end{vmatrix} = 0 \\
\nabla \times F_y &= 0 \\
\nabla \times F_z &= 0
\end{aligned} \tag{3.19}$$

The above conditions give rise to a set of 9 equations, which can be equivalently rewritten as:

$$\begin{aligned}
\frac{\partial e^{J(q)}}{\partial x} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{\partial e^{J(q)}}{\partial y} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = 0 \\
\frac{\partial e^{J(q)}}{\partial x} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} - \frac{\partial e^{J(q)}}{\partial z} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = 0 \\
\frac{\partial e^{J(q)}}{\partial y} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} - \frac{\partial e^{J(q)}}{\partial z} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = 0
\end{aligned} \tag{3.20}$$

Thus $e^{J(q)}$ is a valid Jacobian matrix if Equation 3.23 hold true. We also observe that using Lemma 3.2, for $t = x, y, z$:

$$\begin{aligned}
\frac{\partial J}{\partial t} J &= A \frac{\partial J}{\partial t} \\
\implies [J, \frac{\partial J}{\partial t}] &= (J - A) \frac{\partial J}{\partial t} \\
\text{and } [J, [J, \frac{\partial J}{\partial t}]] &= (J^2 + A^2 - 2JA) \frac{\partial J}{\partial t} \\
&\text{... and so on}
\end{aligned} \tag{3.21}$$

We consider $Q_n = \sum_{r=0}^{n} \binom{n}{r} J^{n-r} (-A)^r$

Now, using the differentiation formula [89, p.115], we get:

$$
\begin{aligned}
\frac{\partial}{\partial t}e^{J(q)} &= e^{J(q)}\frac{I-e^{-ad_J}}{ad_J}\frac{\partial J(q)}{\partial t} \\
&= e^{J(q)}\left(\frac{\partial J(q)}{\partial t} - \frac{1}{2!}[J(q), \frac{\partial J(q)}{\partial t}] + \frac{1}{3!}[J(q),[J(q),\frac{\partial J(q)}{\partial t}]] - ...\right) \\
&= e^{J(q)}\left(\frac{\partial J(q)}{\partial t} - \frac{1}{2!}Q_1\frac{\partial J(q)}{\partial t} + \frac{1}{3!}Q_2\frac{\partial J(q)}{\partial t} - ...\right) \\
&= e^{J(q)}\left(I - \frac{1}{2!}Q_1 + \frac{1}{3!}Q_2 - ...\right)\frac{\partial J(q)}{\partial t} \\
&= e^{J(q)}Q\frac{\partial J(q)}{\partial t},\ t = x, y, z,
\end{aligned}
\tag{3.22}
$$

We prove that the series $\left(I - \frac{1}{2!}Q_1 + \frac{1}{3!}Q_2 - ...\right)$ converges to Q in Section A.1.

Using the above result,

$$
\begin{aligned}
&\frac{\partial e^{J(q)}}{\partial x}\begin{bmatrix}0&0&0\\1&0&0\\0&0&0\end{bmatrix} - \frac{\partial e^{J(q)}}{\partial y}\begin{bmatrix}1&0&0\\0&0&0\\0&0&0\end{bmatrix} \\
=&e^{J(q)}Q\frac{\partial J(q)}{\partial x}\begin{bmatrix}0&0&0\\1&0&0\\0&0&0\end{bmatrix} - e^{J(q)}Q\frac{\partial J(q)}{\partial y}\begin{bmatrix}1&0&0\\0&0&0\\0&0&0\end{bmatrix} \\
=&e^{J(q)}Q\left(\frac{\partial J(q)}{\partial x}\begin{bmatrix}0&0&0\\1&0&0\\0&0&0\end{bmatrix} - \frac{\partial J(q)}{\partial y}\begin{bmatrix}1&0&0\\0&0&0\\0&0&0\end{bmatrix}\right) \\
=&e^{J(q)}Q\left(\begin{bmatrix}\frac{\partial^2\phi_x(q)}{\partial y\partial x}&0&0\\\frac{\partial^2\phi_y(q)}{\partial y\partial x}&0&0\\\frac{\partial^2\phi_z(q)}{\partial y\partial x}&0&0\end{bmatrix} - \begin{bmatrix}\frac{\partial^2\phi_x(q)}{\partial x\partial y}&0&0\\\frac{\partial^2\phi_y(q)}{\partial x\partial y}&0&0\\\frac{\partial^2\phi_z(q)}{\partial x\partial y}&0&0\end{bmatrix}\right) \\
=&0 \text{ [Clairaut's theorem]}
\end{aligned}
\tag{3.23}
$$

Now, we can easily verify that the matrix e^J is curl-free. ■

Even though the set of conditions in the lemma is technical, it is quite general and we point out that such conditions hold for a large family of functions. For example, for the 2D cases, we can verify that any harmonic function and its conjugate [92] together obey these conditions and provide us with such conservative vectors fields. Non-trivial families of functions following these conditions also exist in 3D.

3.5 End to End Pipeline

In this thesis, we propose a postprocessing layer, which can be potentially inserted at the end of a general deep learning registration framework and then the system can be trained end-to-end. Along with the postprocessing layer, we also propose an additional Poisson reconstruction loss for training. The full training pipeline is depicted in Figure 3.1.

In our experiments, we formulate the loss used for training according to the form of Equation 2.1. The loss consists of three components: $\mathcal{L}_{sim}$, which discourages dissimilarity between the moving and fixed image, $\mathcal{L}_{reg}$, that penalizes large local variations in the displacement field and $\mathcal{L}_{poisson}$, that helps our postprocessing layer in reducing non-positive Jacobians.

3.5.1 Cross–Correlation Loss

In our experiments, we take $\mathcal{L}_{sim}$ to the cross-correlation loss as in [10]. Essentially, we consider the negative local normalized cross correlation loss as the similarity measure. Considering the fixed image to be F and the moving image to be M, the moving image is spatially transformed using the deformation field ϕ_p. Thus we find the similarity measure between F and $M(\phi_p)$.

The cross-correlation is computed locally i.e. for a voxel location p, we consider a local cubic volume V of size n^3. Considering such a volume V around p, we compute the local mean (we consider the fixed image F here):

$$\hat{F}(p) = \frac{\sum_{i \in V} F(p_i)}{n^3}$$

Similarly, local means around any voxel location is also found for the warped image. Now, the cross-correlation loss is given by [10]:

$$CC(F, M(\phi_p)) = \sum_{p \in \Omega} \frac{\left(\sum_{p_i} (F(p_i) - \hat{F}(p))(M(\phi_p(p_i)) - \hat{M}(\phi_p(p)))\right)^2}{\left(\sum_{p_i} (F(p_i) - \hat{F}(p))^2\right)\left(\sum_{p_i} (M(\phi_p(p_i)) - \hat{M}(\phi_p(p)))^2\right)} \tag{3.24}$$

Here p_i iterates over the cubic volume of size n^3 as mentioned before. For our experiments, we consider $n = 9$. Now, it is to be noted that conventional cross-correlation would be defined as the square root of the terms summed in Equation 3.24. However, the aforementioned form is commonly used in the image registration literature [14].

Efficient Computation. The local cross-correlation loss mentioned in Equation 3.24 is computed efficiently using convolution operation. Specifically, we consider a convolution kernel of size $9\times9\times9$ and convolve it over the image volumes with stride 1 and a padding of 4. This allows us to compute a volume containing $\sum_{p_i}(F(p_i))$ in a single convolution pass. The local means are obtained by simply dividing the above by n^3. Similar computation is applicable for $\sum_{p_i}(M(\phi_p(p_i)))$. Thus this helps in efficient computation of the similarity measure.

3.5.2 Diffusion Regularizer

We use the diffusion regularizer as $\mathcal{L}_{reg}$, which is defined as follows [10]:

$$\mathcal{L}_{reg} = \sum_{p\in\Omega} \|\nabla\phi_p(p)\|^2$$

The diffusion regularizer is a commonly used regularizer in image registration [10, 93, 94] and was inspired from the field of optical flow [95]. Essentially, the registration field in an neighbourhood in an image should vary smoothly. Thus this regularizer adds a smoothness constraint by penalizing the square of the magnitude of the gradient of the deformation field.

3.5.3 Poisson Reconstruction Loss

The Poisson reconstruction step plays a crucial role in reducing the non-positive Jacobians. Note that a lower resultant value in Equation 3.16 implies a better integrability condition for the matrix J'. Hence, adding such three terms for x, y and z components, we introduce a Poisson reconstruction loss $\mathcal{L}_{poisson}$ as follows:

$$\mathcal{L}_{poisson} = \sum_{q} \|e^{Jac(\phi(q))} - Jac(\phi_p(q))\|_2^2 \tag{3.25}$$

3.5.4 Complete Loss

Using the individual components mentioned above, the complete loss function $\mathcal{L}(F, M)$ is as follows:

$$\mathcal{L}(F, M) = -CC(F, M(\phi_p)) + \lambda \sum_{p} \|\nabla \phi_p(p)\|^2 + \lambda_p \mathcal{L}_{poisson}, \tag{3.26}$$

Here λ is the hyperparameter that controls the strength of the diffusion regularizer and λ_p is the hyperparameter determining the strength of the Poisson reconstruction loss.

Thus every component mentioned in our method is differentiable and we train this pipeline end-to-end using the above loss.

Chapter 4

Experiments and Results

4.1 Dataset

In this work, we primarily use the open-access OASIS dataset [9] to evaluate our postprocess step. The dataset contains 414 T1-weighted brain MRI scans from subjects aged 18 to 96. We obtain the preprocessed dataset from [96]. The MRI scans were preprocessed [97] using Freesurfer [98] by standard steps like resampling, bias correction, skull stripping, affine normalization and center cropping into volumes of $160 \times 192 \times 224$. For our experiments we split the dataset into training, validation and test set of sizes 255, 15 and 144 respectively.

We perform atlas-based registration for our experiments i.e. we aim to establish anatomical correspondence between the moving images and the reference image/atlas. An atlas can be a single volume or an average of volumes in the same image space. Atlas-based registration is commonly applied to register inter-subject images. In this paper, we use an atlas constructed from a different dataset [99] and also used in the official implementation of Voxelmorph [10].

4.2 Evaluation Metric

We evaluate the performance of our postprocessing layer with two metrics: Dice Score (DS) and the percentage of non-positive Jacobian determinants ($|J_{\phi_p}| \leq 0$) [79].

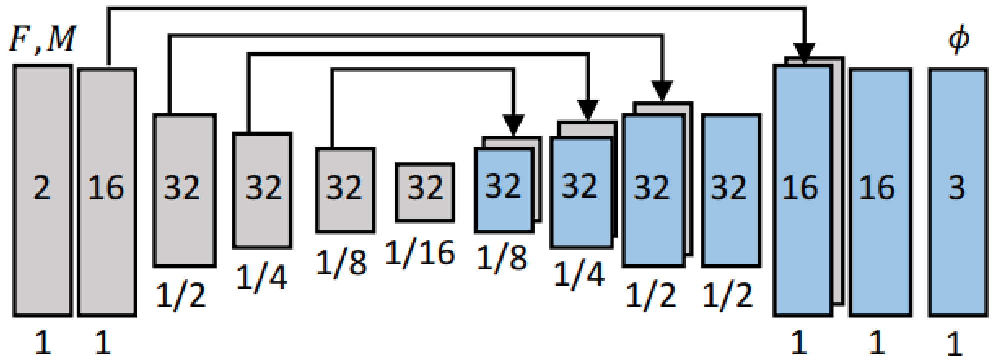


Figure 4.1: Architecture of VoxelMorph-2. Taken from [10]

4.2.1 Dice Score

The Dice Score measures the volume overlap of different segmented anatomical structures. Considering S_F^k and $S_{M(\phi_p)}^k$ to be the sets of voxels for an anatomical structure k for F and $M(\phi_p)$, respectively, the Dice Score [100] is given by :

$$DS(S_F^k, S_{M(\phi_p)}^k) = 2 \times \frac{S_F^k \cap S_{M(\phi_p)}^k}{|S_F^k| + |S_{M(\phi_p)}^k|} \tag{4.1}$$

For our analysis, we consider 30 anatomical structures for the computation of Dice Score [70]. A Dice Score of 1 is highest since it indicates complete overlap and no overlap gives the lowest score of 0.

4.2.2 Non-positive Jacobian determinants

The goal of our postprocessing step is to reduce the number of non-positive Jacobians to ensure a diffeomorphic transformation (Section 2.2.3). Hence we also measure the percentage of voxels which have non-positive Jacobian determinants to evaluate our layer.

4.3 Implementation Details

4.3.1 Architecture

We use the VM2 architecture [10] as the neural network pipeline ($g_\theta(F, M)$) for our experiments. We concatenate the fixed and moving image to obtain an input of di-

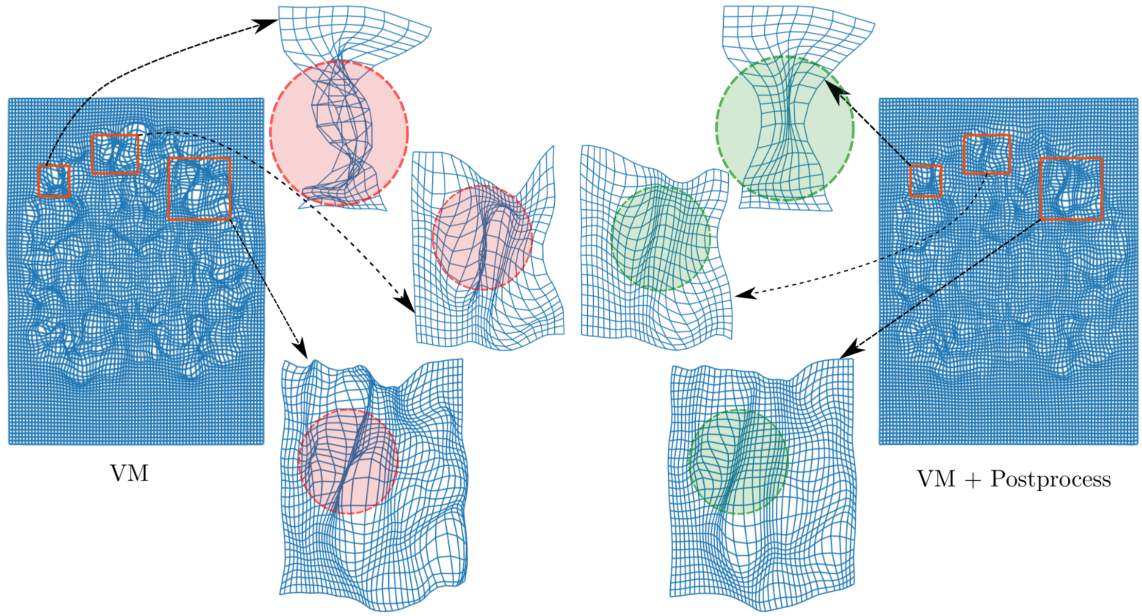


Figure 4.2: Illustration showing the effect of the Postprocessing Layer on foldings in the grid used for registration for one slice. VM = Voxelmorph. Red circles indicate folding of the grid over itself. Green circles indicate no foldings in same region.

mension $160 \times 192 \times 224 \times 2$. The convolutional neural network architecture is based on UNet [69], which contains an encoder stage and a decoder stage which are connected via skip connections. In the VM2 architecture, both the encoder and decoder network consists of convolutional blocks, which typically consist of a convolution with a kernel size of $3 \times 3 \times 3$ followed by a Leaky ReLU activation with a negative slope of 0.2. The encoder networks use the convolutions with a stride of 2 because that reduces the spatial dimensions by half in each layer. As shown in Figure 4.1, the encoder network consists of 5 layers with 16, 32, 32, 32 and 32 channels. Due to the strided convolutions the spatial size of the input volumes along each dimension are reduced till $\frac{1}{16}$ of the original size at the last encoder stage. The decoder network consists of 6 layers of 32, 32, 32, 32, 16 and 16 channels where the spatial dimension is upsampled gradually as shown in Figure 4.1. Like UNet, skip connections are employed to pass features learned in the encoder stage directly to the decoder stage.

Table 4.1: Average Dice Score (higher is better) and Average Percentage of Non-Positive Jacobians (lower is better) with $\lambda = 1, 2$ and increasing λ_p. VM = Voxelmorph. Standard Deviation Given in Parenthesis

Method	λ	λ_p	**Avg. Dice**	**% of** $\lvert J_{\phi_p} \rvert \leq 0$
VM	1.0	-	0.8056 (0.0084)	0.6895 (0.0950)
VM + Postprocess	1.0	0	0.8051 (0.0081)	0.2964 (0.0461)
		0.01	0.8053 (0.0082)	0.2589 (0.0411)
		0.05	0.8045 (0.0087)	0.2255 (0.0378)
		0.1	0.8024 (0.0095)	0.1154 (0.0280)
VM	2.0	-	0.8048 (0.0086)	0.2881 (0.0501)
VM + Postprocess	2.0	0	0.8025 (0.0092)	0.0639 (0.0187)
		0.01	0.8032 (0.0100)	0.0609 (0.0190)
		0.05	0.8043 (0.0096)	0.0672 (0.0201)
		0.1	0.8025 (0.0103)	0.0207 (0.0001)

Table 4.2: Average Run Time in secs for Registration of Pair of Images (lower is better). VM = Voxelmorph. Standard Deviation Given in Parenthesis

Method	**Time in sec**
VM	0.60 (0.10)
VM+Postprocess	1.87 (0.20)

4.3.2 Training details

Since we propose a postprocessing layer in this paper, we compare the results between an existing framework, namely the VoxelMorph framework (VM) [10] and that of our layer used in conjunction with the VoxelMorph framework. We implement our method using Pytorch [101] . We train our models using the Adam optimizer [102] with a learning rate of $1e^{-4}$ for 300 epochs and a batch size of 1. We train both VM and VM in conjunction with our proposed postprocess layer and tune the respective hyperparameters with grid search. Based on the best dice score from the validation set, we get the best result for our setting using $\lambda = 1.0$ and $\lambda_p = 0.01$.

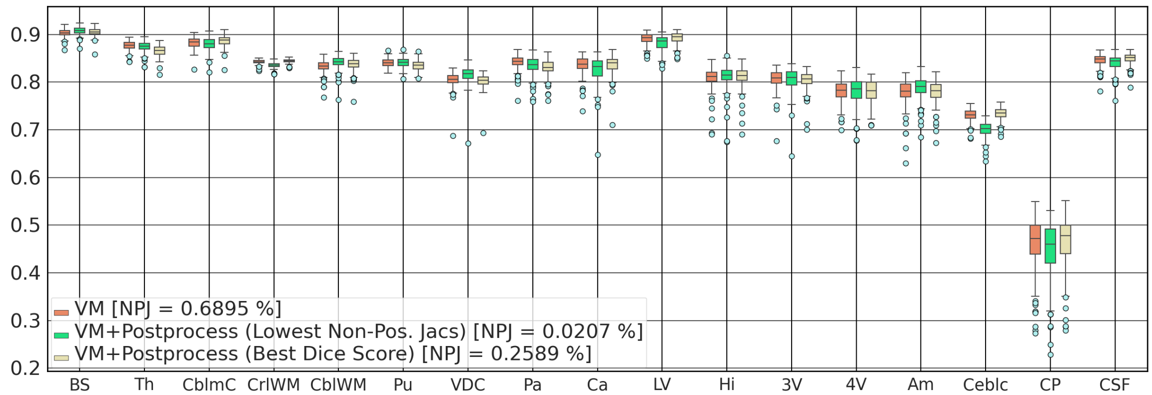


Figure 4.3: Box Plot of Dice Scores of different anatomical structures for VM, model of VM with Postprocessing Layer giving best Dice Score and VM with Postprocessing Layer giving lowest percentage of non-positive Jacobians. NPJ = Percentage of non-positive Jacobians. Structures with left and right hemispheres are combined into one for this illustration. Anatomical structures: Brain Stem (BS), Thalamus (Th), Cerebellum Cortex (CblmC), Cerebral White Matter (CrlWM), Cerebellum WM (CblWM), Putamen (Pu) , Ventral-DC (VDC), Pallidum (Pa), Caudate (Ca), Lateral Ventricle (LV), Hippocampus (Hi), 3rd Ventricle 3V), 4th Ventricle (4V), Amygdala (Am), Cerebral Cortex (Ceblc), Choroid Plexus (CP) and CSF. Table corresponding to this Box Plot is given in Section A.2.

4.3.3 Spatial Transform function

The spatial transformation function, inspired from spatial transformer networks [63] is an integral part of the unsupervised registration DL pipeline. It helps us in computing $M(\phi_p)$ from ϕ_p, which we obtain as an output of the neural network or as an output of our postprocessing layer.

For a voxel location p, we compute $M(\phi_p(p))$ using the eight neighbouring voxels of p (denoted as $\mathcal{Z}(\phi_p(p))$:

$$M(\phi_p(p)) = \sum_{q \in \mathcal{Z}(\phi_p(p))} M(q) \prod_{d \in x,y,z} (1 - |\phi_{p_d}(p) - q_d|)$$

These operations are almost differentiable everywhere - enabling us to train the pipeline end-to-end through backpropagation.

4.4 Results

4.4.1 Registration Performance

Table 4.1 shows the average dice score and the average percentage of voxels with non positive Jacobians for all subjects in the test set for our experiments for different values of λ and λ_p. We observe that adding our postprocessing layer does not noticeably alter the dice score performance, however it reduces the percentage of non-positive Jacobians by a significant amount. Thus our proposed postprocessing layer can reduce folding (see Figure 4.2 for an example) of the registration grid while maintaining a high registration accuracy (in terms of dice score), thus giving more diffeomorphic transformations.

In Figure 4.3, we also show the average dice score for different anatomical structures in the brain as a boxplot. We demonstrate that for VM, VM with postprocessing layer giving the best Dice and VM with postprocessing layer giving the lowest percentage of non-positive jacobians.

4.4.2 Effect of the Poisson Reconstruction Loss

We also demonstrate the effect the proposed reconstruction loss in Table 4.1. We show the average percentage of non-positive jacobians for increasing values of λ_p with the gradient regularization $\lambda = 1, 2$. As the weight of the reconstruction term increases through λ_p, we observe that amount of non-positive Jacobians decreases; however there is not much of a decrease in Dice score. Thus, with increasing λ_p, the reconstruction loss tries to reconstruct a displacement field ϕ_p, whose Jacobian is increasingly closer to e^{J_ϕ} and thus is more diffeomorphic. Hence, our proposed loss is successful in making the deformations more diffeomorphic without sacrificing too much registration accuracy.

4.4.3 Runtime Analysis

Table 4.2 shows the average time required to register a pair of images when we use a VM trained model and when we use a VM model trained along with our layer. We perform the deformable registration of a MRI scan of a test subject to the atlas using a NVIDIA Tesla P100 GPU and an Intel Xeon (E5-2683 v4) CPU. The runtime for registration when we add our layer is greater than that for VM by just about 1.2 seconds. Thus, it maintains the advantage of deep learning methods being faster than traditional registration methods.

4.5 Limitation

In this section we explore a limitation of our work revealed through more experiments. For these experiments, we consider 2D chest X-ray images with high inter-subject anatomical variability for registration using our proposed postprocessing layer and Voxelmorph.

4.5.1 Dataset

For this experiment, we consider the Shenzen Hospital chest X-ray [8, 11, 12] database which was collected in Shenzhen No.3 Peoples Hospital, Guangdong Medical College, Shenzhen, China. It consists of 662 X-ray images of variable sizes with the width and height ranging from 2400 to 3000 pixels. The dataset is essentially a tuberculosis dataset containing 326 normal and 336 pathological images. Manual lung segmentations for each image are present in the dataset, which help us to evaluate our method. We resize each image to 1504×1504 for ease of computation and for uniformity of size among images. For our experiments, we split the dataset into 542, 60 and 60 images as training, validation and test set respectively.

4.5.2 Experimental Setting

As discussed in Section 4.2, we use Dice Score and percentage of non-positive Jacobian as our main evaluation criterion for our method. The dataset cotains masks for lung segmentations for each image. Hence, we calculate Dice Score based on such masks.

For our experiments, we use the same VM2 architecture[10] as mentioned in Section 4.1. We train our models using Adam [102] with a learning rate of 0.001 for 300 epochs and a batch size of 1 and use the mean squared error as the loss function for registration.

4.5.3 Results

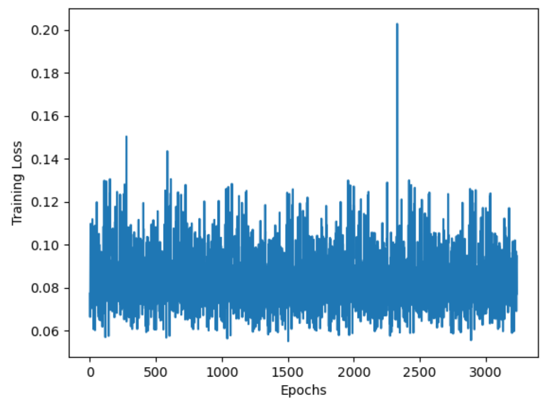
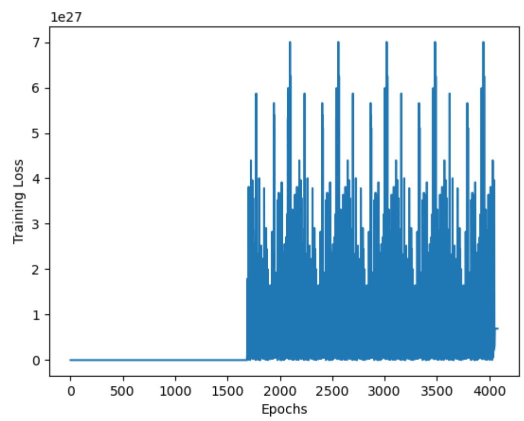
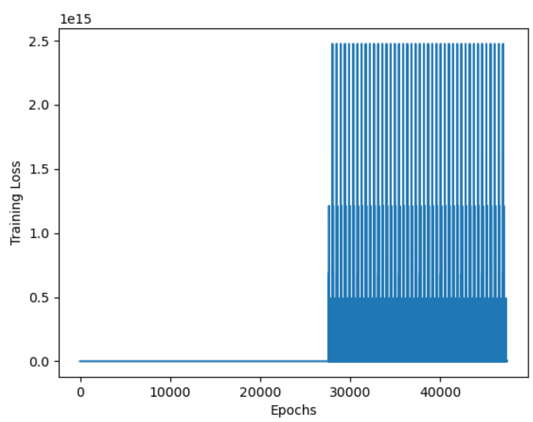
We initially use only Voxelmorph for registration with the Shenzen dataset to understand the viability of Voxelmorph as the underlying deep learning pipeline. In Table 4.3, we present the dice scores for different hyperparameters of λ. We observe that for $\lambda = 0.01$, we attain an acceptable registration performance [103] for the Shenzen dataset.

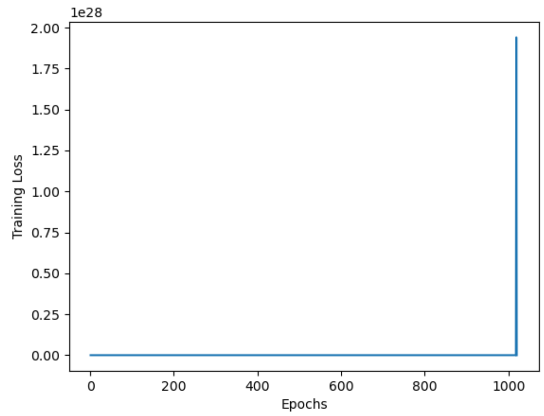
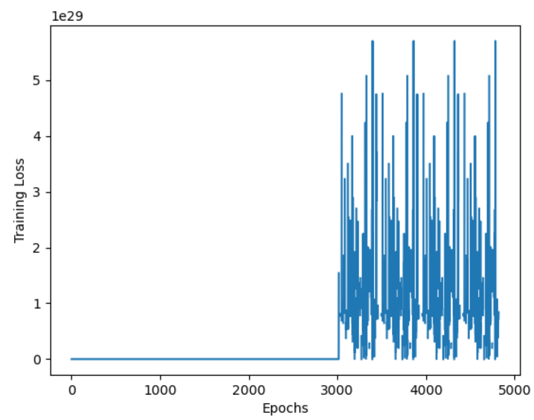
Table 4.3: Dice Score for VM with Shenzen Dataset

λ	**Dice Score**
1.0	0.62
0.5	0.63
0.1	0.80
0.01	0.86

Thus Voxelmorph can serve as an acceptable pipeline for our postprocessing layer. We observe the performance of the postprocessing layer in Table 4.4 - the addition of our layer leads to instability in training. We have demonstrated the training curves for different hyperparameter settings in Table 4.4.

Table 4.4: Training Curves demonstrating instability when training VM2+postprocessing layer with Shenzen Dataset

λ	λ_p	**Training Curves**
1.0	0.0	
1.0	0.001	
0.1	0.001	

<table>
<tr><td>0.01</td><td>0.001</td><td>
</td></tr>
<tr><td>0.01</td><td>1.0</td><td>
</td></tr>
</table>

We observe that either there is no decrease in training loss with sudden spikes in the loss or the training loss completely blows up to an extremely high value. We observe such behaviour across a range of different hyperparameters.

To understand this phenomenon in depth, we choose one such hyperparameter combination i.e. learning rate = 0.001, $\lambda = 0.01$ and $\lambda_p = 0.001$. We track the L2 norm of the gradient with respect to parameters and different stages of our pipeline as shown in Figure 4.4. We observe that the all the tracked norm of gradients except one (gradient of mean of J_ϕ with respect to parameters of the network) become unstable and rise exponentially at around epoch 1000. This is the same time that the loss blows up as seen in Table 4.4 row 4.

Our intuition is that the instability may also arise from unboundedness of the norm of the Jacobian J_ϕ. Hence , we similarly track the L2 norm of the Jacobian in

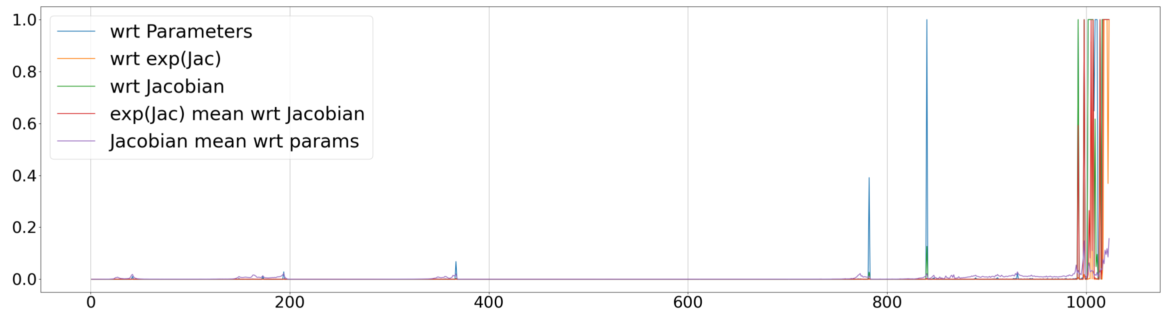


Figure 4.4: Tracking L2 norm of gradients. wrt 'A' denotes L2 norm of gradients of loss with respect to 'A'. 'B' wrt 'A' denotes L2 norm of gradients of 'B' with respect to 'A'. Jacobian = J_ϕ. exp(Jac) = J'.

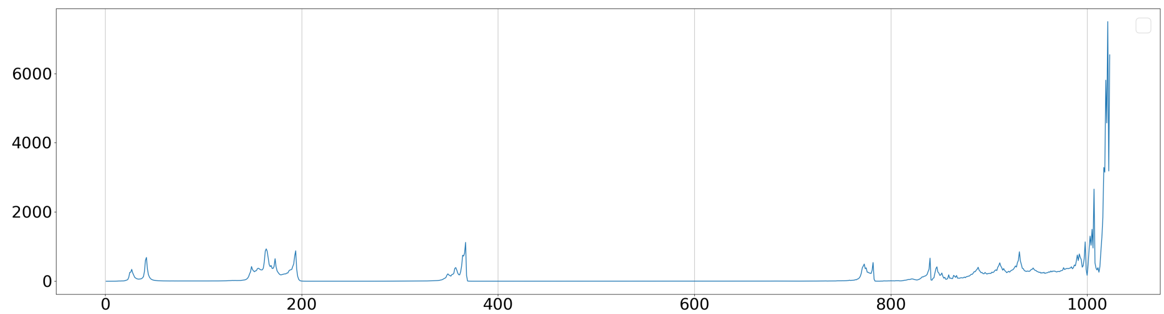


Figure 4.5: Tracking mean L2 norm of Jacobian matrix J_ϕ. Mean is taken over the voxels.

Figure 4.5. We observe that a similar phenomenon occur i.e. the norm increases in an unstable manner around the same epoch.

This presents a practical limitation in our current approach and points towards the need of further investigation.

Chapter 5

Conclusion

Deformable image registration is an important problem in medical image applications. A very important requirement of such deformable registrations is diffeomorphism - diffeomorphic registration preserves topology, prevents singularity. Classical registration literature have solved the diffeomorphic problem by either constraining the space of velocity fields, certain forms of regularization etc. However, classical registration performs slow pairwise registration. The advent of deep learning has enabled fast registration during test time - which is a desirable property. However, due to such advances being recent, there are few efforts to attain diffeomorphic image registration using deep learning.

In this thesis, we have presented a postprocessing layer which can fit in a deep learning registration framework with end-to-end learning. The purpose of our layer is to reduce the number of non-positive jacobians in the displacement field obtained from the DL framework - thus obtaining more diffeomorphic mappings. We evaluate our layer using large scale brain MR dataset with the Voxelmorph framework and show that our layer is successful in reducing folding in the registration grid and maintaining high registration accuracy. Even though we employ a Poisson equation solver, our layer still maintains the advantage of fast registration, a desirable characteristic of deep learning algorithms. We hope that our postprocessing layer can be used in other registration frameworks desiring more diffeomorphic registration fields.

The exponentiated Jacobian is not always integrable, but under certain conditions it can give a valid Jacobian as explored in 3.3 and that will lead to a theoretical guarantee of strictly positive Jacobians. Thus, for future work, we hope to develop constrained registration fields that can lead to a theoretically guaranteed, fully diffeomorphic registration.

Additionally, we demonstrate a limitation of our work based on experiments of 2D chest X-ray data. Addition of our layer makes the training unstable - we find that this occurs due to gradient explosion and is possibly correlated with explosion of norm of Jacobians. Hence this presents a stage for future work which can properly investigate and mitigate this phenomenon, thus helping in development of a general postprocessing layer leading to diffeomorphic registration.

Bibliography

[1] J. Modersitzki, *FAIR: flexible algorithms for image registration*. SIAM, 2009.

[2] S. Oh and S. Kim, "Deformable image registration in radiation therapy," *Radiation oncology journal*, vol. 35, no. 2, p. 101, 2017.

[3] J. Ashburner and K. J. Friston, "Voxel-based morphometry—the methods," *Neuroimage*, vol. 11, no. 6, pp. 805–821, 2000.

[4] C. Gaser, "Structural mri: Morphometry," in *Neuroeconomics*, M. Reuter and C. Montag, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 399–409, ISBN: 978-3-642-35923-1. DOI: 10.1007/978-3-642-35923-1_21. [Online]. Available: https://doi.org/10.1007/978-3-642-35923-1_21.

[5] F. E.-Z. A. El-Gamal, M. Elmogy, and A. Atwan, "Current trends in medical image registration and fusion," *Egyptian Informatics Journal*, vol. 17, no. 1, pp. 99–124, 2016.

[6] 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*, vol. 61, no. 2, pp. 139–157, 2005.

[7] B. Sparks *et al.*, "Brain structural abnormalities in young children with autism spectrum disorder," *Neurology*, vol. 59, no. 2, pp. 184–192, 2002.

[8] L. Mansilla, D. H. Milone, and E. Ferrante, "Learning deformable registration of medical images with anatomical constraints," *Neural Networks*, vol. 124, pp. 269–279, 2020.

[9] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris, and R. L. Buckner, "Open access series of imaging studies (OASIS): Cross-sectional MRI data in young, middle aged, nondemented, and demented older adults," en, *J. Cogn. Neurosci.*, vol. 19, no. 9, pp. 1498–1507, Sep. 2007.

[10] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "An unsupervised learning model for deformable medical image registration," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9252–9260.

[11] S. Jaeger, S. Candemir, S. Antani, Y.-X. J. Wáng, P.-X. Lu, and G. Thoma, "Two public chest x-ray datasets for computer-aided screening of pulmonary diseases," *Quantitative imaging in medicine and surgery*, vol. 4, no. 6, p. 475, 2014.

[12] S. Stirenko *et al.*, "Chest x-ray analysis of tuberculosis by deep learning with segmentation and augmentation," in *2018 IEEE 38th International Conference on Electronics and Nanotechnology (ELNANO)*, IEEE, 2018, pp. 422–428.

[13] R. Szeliski, *Computer vision: algorithms and applications*. Springer Nature, 2022.

[14] B. B. Avants, C. L. Epstein, M. Grossman, and J. C. Gee, "Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain," *Medical image analysis*, vol. 12, no. 1, pp. 26–41, 2008.

[15] P. Viola and W. M. Wells III, "Alignment by maximization of mutual information," *International journal of computer vision*, vol. 24, no. 2, pp. 137–154, 1997.

[16] C. Studholme, D. L. Hill, and D. J. Hawkes, "An overlap invariant entropy measure of 3d medical image alignment," *Pattern recognition*, vol. 32, no. 1, pp. 71–86, 1999.

[17] E. Haber and J. Modersitzki, "Intensity gradient based registration and fusion of multi-modal images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2006, pp. 726–733.

[18] R. A. Willoughby, "Solutions of ill-posed problems (an tikhonov and vy arsenin)," *SIAM Review*, vol. 21, no. 2, p. 266, 1979.

[19] E. Haber and J. Modersitzki, "Volume preserving image registration," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2004, pp. 591–598.

[20] L. Zagorchev and A. Goshtasby, "A comparative study of transformation functions for nonrigid image registration," *IEEE transactions on image processing*, vol. 15, no. 3, pp. 529–538, 2006.

[21] L. Younes, "Ordinary differential equations and groups of diffeomorphisms," in *Shapes and Diffeomorphisms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, pp. 183–203, ISBN: 978-3-662-58496-5. DOI: 10.1007/978-3-662-58496-5_7. [Online]. Available: https://doi.org/10.1007/978-3-662-58496-5_7.

[22] P. G. Ciarlet, *Mathematical elasticity: Three-dimensional elasticity*. SIAM, 2021.

[23] C. Broit, *Optimal registration of deformed images*. University of Pennsylvania, 1981.

[24] R. Bajcsy and S. Kovačič, "Multiresolution elastic matching," *Computer vision, graphics, and image processing*, vol. 46, no. 1, pp. 1–21, 1989.

[25] C. Davatzikos, "Spatial transformation and registration of brain images using elastically deformable models," *Computer Vision and Image Understanding*, vol. 66, no. 2, pp. 207–222, 1997.

[26] G. E. Christensen and H. J. Johnson, "Consistent image registration," *IEEE transactions on medical imaging*, vol. 20, no. 7, pp. 568–582, 2001.

[27] A. Leow *et al.*, "Inverse consistent mapping in 3d deformable image registration: Its construction and statistical properties," in *Biennial International Conference on Information Processing in Medical Imaging*, Springer, 2005, pp. 493–503.

[28] J. He and G. E. Christensen, "Large deformation inverse consistent elastic image registration," in *Biennial International Conference on Information Processing in Medical Imaging*, Springer, 2003, pp. 438–449.

[29] G. E. Christensen, R. D. Rabbitt, and M. I. Miller, "Deformable templates using large deformation kinematics," *IEEE transactions on image processing*, vol. 5, no. 10, pp. 1435–1447, 1996.

[30] W. Crum, C Tanner, and D. Hawkes, "Anisotropic multi-scale fluid registration: Evaluation in magnetic resonance breast imaging," *Physics in Medicine & Biology*, vol. 50, no. 21, p. 5153, 2005.

[31] M. Bro-Nielsen and C. Gramkow, "Fast fluid registration of medical images," in *International conference on visualization in biomedical computing*, Springer, 1996, pp. 265–276.

[32] N. D. Cahill, J. A. Noble, and D. J. Hawkes, "Fourier methods for nonparametric image registration," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2007, pp. 1–8.

[33] J.-P. Thirion, "Image matching as a diffusion process: An analogy with maxwell's demons," *Medical image analysis*, vol. 2, no. 3, pp. 243–260, 1998.

[34] X. Pennec, P. Cachier, and N. Ayache, "Understanding the "demon's algorithm": 3d non-rigid registration by gradient descent," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 1999, pp. 597–605.

[35] T. Vercauteren, X. Pennec, E. Malis, A. Perchant, and N. Ayache, "Insight into efficient image registration techniques and the demons algorithm," in *Biennial International Conference on Information Processing in Medical Imaging*, Springer, 2007, pp. 495–506.

[36] N. D. Cahill, J. A. Noble, and D. J. Hawkes, "A demons algorithm for image registration with locally adaptive regularization," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2009, pp. 574–581.

[37] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache, "Symmetric log-domain diffeomorphic registration: A demons-based approach," in *International conference on medical image computing and computer-assisted intervention*, Springer, 2008, pp. 754–761.

[38] A. Guimond, A. Roche, N. Ayache, and J. Meunier, "Three-dimensional multimodal brain warping using the demons algorithm and adaptive intensity corrections," *IEEE transactions on medical imaging*, vol. 20, no. 1, pp. 58–69, 2001.

[39] M. Modat, T. Vercauteren, G. R. Ridgway, D. J. Hawkes, N. C. Fox, and S. Ourselin, "Diffeomorphic demons using normalized mutual information, evaluation on multimodal brain mr images," in *Medical Imaging 2010: Image Processing*, SPIE, vol. 7623, 2010, pp. 800–807.

[40] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 11, no. 6, pp. 567–585, 1989.

[41] F. L. Bookstein, "Thin-plate splines and the atlas problem for biomedical images," in *Biennial international conference on information processing in medical imaging*, Springer, 1991, pp. 326–342.

[42] M. H. Davis, A. Khotanzad, D. P. Flamig, and S. E. Harms, "A physics-based coordinate transformation for 3-d image matching," *IEEE transactions on medical imaging*, vol. 16, no. 3, pp. 317–328, 1997.

[43] J. Ashburner and K. J. Friston, "Nonlinear spatial normalization using basis functions," *Human brain mapping*, vol. 7, no. 4, pp. 254–266, 1999.

[44] A. Trouvé, "Diffeomorphisms groups and pattern matching in image analysis," *International journal of computer vision*, vol. 28, no. 3, pp. 213–221, 1998.

[45] P. Dupuis, U. Grenander, and M. I. Miller, "Variational problems on flows of diffeomorphisms for image matching," *Quarterly of applied mathematics*, pp. 587–600, 1998.

[46] L. Younes, *Shapes and diffeomorphisms*. Springer, 2010, vol. 171.

[47] F.-X. Vialard, L. Risser, D. Rueckert, and C. J. Cotter, "Diffeomorphic 3d image registration via geodesic shooting using an efficient adjoint calculation," *International Journal of Computer Vision*, vol. 97, no. 2, pp. 229–241, 2012.

[48] J. Ashburner, "A fast diffeomorphic image registration algorithm," *Neuroimage*, vol. 38, no. 1, pp. 95–113, 2007.

[49] C. Moler and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix," *SIAM review*, vol. 20, no. 4, pp. 801–836, 1978.

[50] S. Wright, J. Nocedal, *et al.*, "Numerical optimization," *Springer Science*, vol. 35, no. 67-68, p. 7, 1999.

[51] R. D. Rabbitt, J. A. Weiss, G. E. Christensen, and M. I. Miller, "Mapping of hyperelastic deformable templates using the finite element method," in *Vision Geometry IV*, SPIE, vol. 2573, 1995, pp. 252–265.

[52] M. Droske and M. Rumpf, "A variational approach to nonrigid morphological image registration," *SIAM Journal on Applied Mathematics*, vol. 64, no. 2, pp. 668–687, 2004.

[53] M. Droske and M. Rumpf, “Multiscale joint segmentation and registration of image morphology,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 12, pp. 2181–2194, 2007.

[54] C. Le Guyader and L. A. Vese, “A combined segmentation and registration framework with a nonlinear elasticity smoother,” *Computer Vision and Image Understanding*, vol. 115, no. 12, pp. 1689–1709, 2011.

[55] M. Burger, J. Modersitzki, and L. Ruthotto, “A hyperelastic regularization energy for image registration,” *SIAM Journal on Scientific Computing*, vol. 35, no. 1, B132–B148, 2013.

[56] T. Rohlfing, C. R. Maurer, D. A. Bluemke, and M. A. Jacobs, “Volume-preserving nonrigid registration of mr breast images using free-form deformation with an incompressibility constraint,” *IEEE transactions on medical imaging*, vol. 22, no. 6, pp. 730–741, 2003.

[57] E. Haber and J. Modersitzki, “Numerical methods for volume preserving image registration,” *Inverse problems*, vol. 20, no. 5, p. 1621, 2004.

[58] J. Krebs *et al.*, “Robust non-rigid registration through agent-based action learning,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2017, pp. 344–352.

[59] X. Yang, R. Kwitt, M. Styner, and M. Niethammer, “Quicksilver: Fast predictive image registration–a deep learning approach,” *NeuroImage*, vol. 158, pp. 378–396, 2017.

[60] H. Sokooti, B. De Vos, F. Berendsen, B. P. 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*, Springer, 2017, pp. 232–239.

[61] M.-M. Rohé, M. Datar, T. Heimann, M. Sermesant, and X. Pennec, “Svf-net: Learning deformable image registration using shape matching,” in *International conference on medical image computing and computer-assisted intervention*, Springer, 2017, pp. 266–274.

[62] X. Cao *et al.*, “Deformable image registration based on similarity-steered cnn regression,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2017, pp. 300–308.

[63] M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, “Spatial transformer networks,” *Advances in neural information processing systems*, vol. 28, pp. 2017–2025, 2015.

[64] B. D. de Vos, F. F. Berendsen, M. A. Viergever, M. Staring, and I. Išgum, “End-to-end unsupervised deformable image registration with a convolutional neural network,” in *Deep learning in medical image analysis and multimodal learning for clinical decision support*, Springer, 2017, pp. 204–212.

[65] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. Hill, M. O. Leach, and D. J. Hawkes, "Nonrigid registration using free-form deformations: Application to breast mr images," *IEEE transactions on medical imaging*, vol. 18, no. 8, pp. 712–721, 1999.

[66] H. Li and Y. Fan, "Non-rigid image registration using fully convolutional networks with deep self-supervision," *arXiv preprint arXiv:1709.00799*, 2017.

[67] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[68] V. Vishnevskiy, T. Gass, G. Szekely, C. Tanner, and O. Goksel, "Isotropic total variation regularization of displacements in parametric image registration," *IEEE transactions on medical imaging*, vol. 36, no. 2, pp. 385–395, 2016.

[69] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[70] 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*, vol. 38, no. 8, pp. 1788–1800, 2019.

[71] B. D. De Vos, F. F. Berendsen, M. A. Viergever, H. Sokooti, M. Staring, and I. Išgum, "A deep learning framework for unsupervised affine and deformable image registration," *Medical image analysis*, vol. 52, pp. 128–143, 2019.

[72] J. Neylon, Y. Min, D. A. Low, and A. Santhanam, "A neural network approach for fast, automated quantification of dir performance," *Medical physics*, vol. 44, no. 8, pp. 4126–4138, 2017.

[73] A. Sheikhjafari, M. Noga, K. Punithakumar, and N. Ray, "Unsupervised deformable image registration with fully connected generative neural network," 2018.

[74] C. Stergios *et al.*, "Linear and deformable image registration with 3d convolutional neural networks," in *Image analysis for moving organ, breast, and thoracic images*, Springer, 2018, pp. 13–22.

[75] A. V. Dalca, G. Balakrishnan, J. Guttag, and M. R. Sabuncu, "Unsupervised learning for fast probabilistic diffeomorphic registration," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2018, pp. 729–738.

[76] A. V. Dalca, G. Balakrishnan, J. Guttag, and M. R. Sabuncu, "Unsupervised learning of probabilistic diffeomorphic registration for images and surfaces," *Medical image analysis*, vol. 57, pp. 226–236, 2019.

[77] J. Krebs, H. Delingette, B. Mailhé, N. Ayache, and T. Mansi, "Learning a probabilistic model for diffeomorphic registration," *IEEE transactions on medical imaging*, vol. 38, no. 9, pp. 2165–2176, 2019.

[78] 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*, Springer, 2018, pp. 101–109.

[79] T. C. Mok and A. Chung, "Fast symmetric diffeomorphic image registration with convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4644–4653.

[80] J. Zhang, "Inverse-consistent deep networks for unsupervised deformable image registration," *arXiv preprint arXiv:1809.03443*, 2018.

[81] D. Kuang and T. Schmah, "Faim–a convnet method for unsupervised 3d medical image registration," in *International Workshop on Machine Learning in Medical Imaging*, Springer, 2019, pp. 646–654.

[82] B. Kim, D. H. Kim, S. H. Park, J. Kim, J.-G. Lee, and J. C. Ye, "Cyclemorph: Cycle consistent unsupervised deformable image registration," *Medical Image Analysis*, vol. 71, p. 102 036, 2021.

[83] K. Punithakumar, M. Noga, I. B. Ayed, and P. Boulanger, "Right ventricular segmentation in cardiac mri with moving mesh correspondences," *Computerized Medical Imaging and Graphics*, vol. 43, pp. 15–25, 2015.

[84] K. Punithakumar, P. Boulanger, and M. Noga, "A gpu-accelerated deformable image registration algorithm with applications to right ventricular segmentation," *IEEE Access*, vol. 5, pp. 20 374–20 382, 2017.

[85] J. Liu, *New development of the deformation method*. The University of Texas at Arlington, 2006.

[86] A. Sheikhjafari, D. Krishnaswamy, M. Noga, N. Ray, and K. Punithakumar, "Unsupervised diffeomorphic cardiac image registration using parameterization of the deformation field," *arXiv preprint arXiv:2208.13275*, 2022.

[87] A. Nan, M. Tennant, U. Rubin, and N. Ray, "Drmime: Differentiable mutual information and matrix exponential for multi-resolution image registration," in *Medical Imaging with Deep Learning*, PMLR, 2020, pp. 527–543.

[88] G. Strang, *Computational Science and Engineering*. Wellesley-Cambridge Press, 2007.

[89] B. C. Hall, *Lie Groups, Lie Algebras, and Representations An Elementary Introduction*, 2nd ed. London: Springer, 2015.

[90] J. H. Hubbard and B. B. Hubbard, *Vector calculus, linear algebra, and differential forms: a unified approach*. Matrix Editions, 2015.

[91] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," in *ACM SIGGRAPH 2003 Papers*, 2003, pp. 313–318.

[92] T. W. Gamelin, *Complex Analysis*. New York: Springer, 2001.

[93] B. Fischer and J. Modersitzki, “A unified approach to fast image registration and a new curvature based registration technique,” *Linear Algebra and its applications*, vol. 380, pp. 107–124, 2004.

[94] B. Fischer and J. Modersitzki, “Fast diffusion registration,” *Contemporary Mathematics*, vol. 313, pp. 117–128, 2002.

[95] B. K. Horn and B. G. Schunck, “Determining optical flow,” *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[96] [Online]. Available: https://github.com/adalca/medical-datasets/blob/master/neurite-oasis.md.

[97] A. Hoopes, M. Hoffmann, B. Fischl, J. Guttag, and A. V. Dalca, “Hypermorph: Amortized hyperparameter learning for image registration,” in *International Conference on Information Processing in Medical Imaging*, Springer, 2021, pp. 3–17.

[98] B. Fischl, “Freesurfer,” *Neuroimage*, vol. 62, no. 2, pp. 774–781, 2012.

[99] R. Sridharan *et al.*, “Quantification and analysis of large multimodal clinical image studies: Application to stroke,” in *International Workshop on Multimodal Brain Image Analysis*, Springer, 2013, pp. 18–30.

[100] L. R. Dice, “Measures of the amount of ecologic association between species,” *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[101] A. Paszke *et al.*, “Automatic differentiation in pytorch,” 2017.

[102] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[103] E. Al Safadi and X. Song, “Learning-based image registration with meta-regularization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 928–10 937.

Appendix A:

A.1 Convergence of series in 3.3

Here we prove the convergence of the series $\left(I - \frac{1}{2!}Q_1 + \frac{1}{3!}Q_2 - ...\right)$ where $Q_n = \sum\limits_{r=0}^{n} \binom{n}{r} J^{n-r}(-A)^r$

Proof. We know that if a series converges absolutely, the series converges. So we look at the convergence of the following:

$$\|I\| + \| - \frac{1}{2!}Q_1\| + \|\frac{1}{3!}Q_2\| + \| - \frac{1}{4!}Q_3\| + ...$$
$$= \|I\| + \|\frac{1}{2!}Q_1\| + \|\frac{1}{3!}Q_2\| + \|\frac{1}{4!}Q_3\| + ...$$
$$= \|I\| + \frac{1}{2!}\|(J - A)\| + \frac{1}{3!}\|(J^2 - 2JA + A^2)\| + \frac{1}{4!}\|(J^3 - 3J^2A + 3JA^2 - A^3)\| + ...$$
$$\leq \|I\| + \frac{1}{2!}(\|J\| + \| - A\|) + \frac{1}{3!}(\|(J^2\| + \| - 2JA\| + \|A^2\|)$$
$$+ \frac{1}{4!}(\|J^3\| + \| - 3J^2A\| + \|3JA^2\| + \| - A^3\|) + ... \text{ [Triangle Inequality]}$$
$$\leq \|I\| + \frac{1}{2!}(\|J\| + \|A\|) + \frac{1}{3!}(\|(J\|^2 + 2\|J\|\|A\| + \|A\|^2)$$
$$+ \frac{1}{4!}(\|J\|^3 + 3\|J\|^2\|A\| + 3\|J\|\|A\|^2 + \|A\|^3) + ... \text{ [Submultiplicative Property]}$$
$$= \|I\| + \frac{1}{2!}(\|J\| + \|A\|) + \frac{1}{3!}(\|J\| + \|A\|)^2 + \frac{1}{4!}(\|J\| + \|A\|)^3 + ...$$
$$= \|I\| + \frac{1}{(\|J\| + \|A\|)}(\frac{1}{2!}(\|J\| + \|A\|)^2 + \frac{1}{3!}(\|J\| + \|A\|)^3 + \frac{1}{4!}(\|J\| + \|A\|)^4 + ...)$$
$$= \|I\| + \frac{1}{(\|J\| + \|A\|)}(\frac{1}{2!}(\|J\| + \|A\|)^2 + \frac{1}{3!}(\|J\| + \|A\|)^3 + \frac{1}{4!}(\|J\| + \|A\|)^4 + ...)$$
$$= \|I\| + \frac{1}{(\|J\| + \|A\|)}\left(e^{(\|J\|+\|A\|)} - (\|J\| + \|A\|) - 1\right)$$

Thus the series is absolutely convergent, hence convergent. ∎

A.2 Table corresponding to Figure 4.3

Table A.1: Table of Average Dice Scores (higher is better) of different anatomical structures for Voxelmorph, model of Voxelmorph with Postprocessing Layer giving best Dice Score and Voxelmorph with Postprocessing Layer giving lowest percentage of non-positive Jacobians. NPJ = Percentage of non-positive Jacobians. Structures with left and right hemispheres are combined into one. Anatomical structures: Brain Stem (BS), Thalamus (Th), Cerebellum Cortex (CblmC), Cerebral White Matter (CrlWM), Cerebellum WM (CblWM), Putamen (Pu) , Ventral-DC (VDC), Pallidum (Pa), Caudate (Ca), Lateral Ventricle (LV), Hippocampus (Hi), 3rd Ventricle 3V), 4th Ventricle (4V), Amygdala (Am), Cerebral Cortex (Ceblc), Choroid Plexus (CP) and CSF. VM = Voxelmorph. Standard Deviation Given in Parenthesis.

Anatomical Structures	VM	**VM + Postprocess (Lowest NPJ)**	**VM + Postprocess (Best Dice)**
BS	0.9026 (0.0074)	0.9074 (0.0076)	0.9045 (0.0079)
Th	0.8761 (0.0092)	0.8740 (0.0109)	0.8645 (0.0113)
CblmC	0.8818 (0.0107)	0.8798 (0.0128)	0.8857 (0.0104)
CrlWM	0.8420 (0.0044)	0.8352 (0.0053)	0.8441 (0.0039)
CblWM	0.8320 (0.0127)	0.8405 (0.0133)	0.8365 (0.0130)
Pu	0.8400 (0.0088)	0.8409 (0.0097)	0.8340 (0.0096)
VDC	0.8043 (0.0149)	0.8152 (0.0168)	0.8020 (0.0135)
Pa	0.8405 (0.0151)	0.8336 (0.0195)	0.8298 (0.0166)
Ca	0.8353 (0.0182)	0.8244 (0.0283)	0.8357 (0.0185)
LV	0.8903 (0.0117)	0.8814 (0.0155)	0.8922 (0.0119)
Hi	0.8083 (0.0222)	0.8109 (0.0242)	0.8108 (0.0215)
3V	0.8056 (0.0192)	0.8050 (0.0245)	0.8036 (0.0192)
4V	0.7789 (0.0230)	0.7794 (0.0298)	0.7775 (0.0250)
Am	0.7778 (0.0263)	0.7873 (0.0237)	0.7788 (0.0237)
Ceblc	0.7304 (0.0131)	0.7002 (0.0159)	0.7341 (0.0127)
CP	0.4605 (0.0543)	0.4486 (0.0569)	0.4644 (0.0546)
CSF	0.8447 (0.0130)	0.8393 (0.0165)	0.8482 (0.0122)