Part I
Deformable Image Registration

1 Introduction

Deformable image registration (DIR) is one of the most studied problems in the medical image analysis field. It has led to elegant mathematical solutions that tie together elements of group theory, differential geometry, and probability theory. This chapter describes the basic building blocks of DIR, and provides the theoretical outlines of some widely used algorithms.

1.1 What is DIR and what are its applications?

In this chapter, we will use the following definition of DIR:

> *DIR is the process of non-linearly transforming the space in which images are defined in order to match corresponding anatomical locations between images.*

There are several key concepts in this definition:

- DIR is about transforming the space in which images "live", rather than images themselves. This is a critical aspect of DIR that can be confusing to newcomers. Mathematically, DIR will be described in terms of functions $\phi : \Omega \to \Omega$ that map points in the image domain $\Omega \subset \mathbb{R}^n$ to other points in Ω. These functions are called *spatial transformations, spatial maps,* or simply *warps.*

- Unlike in rigid or affine registration, in DIR we are interested in transformations that are *non-linear.* This offers many more (in fact, infinitely many) degrees of freedom for transforming images. This in turn allows DIR to match images more precisely. The disadvantage is added computational complexity, and the need to ensure that transformations ϕ are well-behaved. For example, a lot of research has focused on constraining transformations ϕ to be *invertible*, which prevents them from tearing or folding the image domain Ω.

- The goal of DIR is to match up *corresponding anatomical locations* between images. For example, when performing DIR on images of faces, corresponding anatomical locations may be the corners of the eyes and lips, tip of the nose, etc. Clearly, the concept of anatomical correspondence is rather tenuous. For example when it comes to matching hair between different people, or between different images of the same person, correspondence is ill defined. This is also true in medical images. For example, in the brain, we might be able to match up certain places in the lateral ventricles and subcortical gray matter structures such as the thalamus between individuals, but not so for cortical folds, as the folding pattern of every brain is unique like a fingerprint. The presence of tumors and lesions in some individuals and not in others makes it even more difficult to find anatomical correspondences. However, these challenges have not prevented DIR from being one of the most useful techniques in medical imaging research.

An example of DIR between brain MRI scans of two individuals is shown in Figure XXX. Certain corresponding anatomical locations are highlighted with white circles in the first image and yellow circles in the second image. The transformation ϕ maps the white landmarks into new locations, indicated as green circles. The green circles are closer to the corresponding yellow circles.

DIR has many important applications in medical imaging research:

- DIR is used for anatomical normalization in population studies that involve imaging. Consider, for example, functional MRI research, which has transformed the fields of psychology, psychiatry, sociology, linguistics, etc. In a typical study, researchers recruit a cohort of individuals (e.g., 20 volunteers) and scan their brains using fMRI under various experimental conditions (e.g., reading long sentences vs. reading short sentences). Analyzing the MRI data from each participant results in a "heat map" that describes experiment-related brain activity in each location of the brain. In order to describe the effects of the experiment at the *population level*, rather than individual level, the brains of these different individuals are transformed to a "template" brain using DIR, so that the corresponding anatomical regions are matched up between different individuals. **[Reference Atlas Chapter]**

- DIR is used to quantify changes in longitudinal imaging. Patients undergoing monitoring or treatment for chronic diseases are often scanned at different times and changes between these longitudinal scans are used to assess the progression of disease or efficacy of treatment. For example, in multiple sclerosis, the changes in size and number of white matter lesions is used clinically to determine whether a treatment regimen is effective. DIR between images obtained at different time points produces a transformation that matches anatomical locations across time. Examining the geometric properties of this transformation (i.e., whether it shrinks space or expands space at different locations in the image domain) can provide accurate measurements of change in the size of white matter lesions. Such measurements are much more effective than visual examination of longitudinal images, particularly when changes over time are subtle.

- DIR is used to account for motion of organs and tissues, for example during surgery. In image-guided surgery, it is common to acquire pre-operative high-resolution images of the patient's anatomy (e.g., a high-resolution CT) for planning purposes; and to collect lower-resolution images intra-operatively for surgical guidance (e.g., using 3D ultrasound or 2D fluoroscopy). In many parts of the body, such as the abdomen, organs shift, and performing only rigid or affine registration between the preoperative and intraoperative images would result in misalignment and lead to surgical errors. DIR can help account for organ motion.

- DIR is a building block of many image analysis algorithms, among them, atlas-based and multi-atlas segmentation. When you are interested in segmenting a certain anatomical structure and you have an example or set of example images in which this structure has already been segmented, performing DIR between these example images and new (not yet segmented) images provides a set of transformations from the example segmentations to the new image. Transforming the example segmentations (an when there are multiple segmentations, combining them in some smart way) yields a segmentation of the new image. **[POINT TO MALF chapter]**.

1.2 DIR Problem Setup

In most algorithms, DIR is formulated as an energy minimization problem in the following form:

$$\theta^* = \arg\min_{\theta\in\Theta} E[\phi_\theta; I, J] \quad \text{where} \quad E[\phi_\theta; I, J] = \mu[I, J \circ \phi_\theta] + \lambda\rho[\theta]\,. \tag{1}$$

Let us carefully consider each term in the above expression:

- $\phi_\theta : \Omega \to \Omega$ is the *spatial transformation* that maps the image domain $\Omega \subset \mathbb{R}^n$ onto itself. It is a continuous function of n variables that has n components. For example, in two dimensions, we may write ϕ_θ as a vector

  $$\phi_\theta(\mathbf{x}) = \left[\begin{array}{c} \phi_\theta^1(x_1, x_2) \\ \phi_\theta^2(x_1, x_2) \end{array} \right],$$

  where ϕ_θ^1 and ϕ_θ^2 are scalar functions, called the components of ϕ_θ.

- I and J are images. In the theoretical portions of this chapter, we will treat them as continuous functions defined on the image domain, i.e., $I, J : \Omega \to \mathbb{R}$. We shall use $\mathcal{I}_\Omega$ to denote the space of all images on Ω. We typically distinguish between the *reference image* (alternatively called *fixed image*), which remains unchanged over the course of the energy minimization; and the *moving image*, which is affected by the transformation ϕ_θ. In expression (1), I is the reference image and J is the moving image.

- $J \circ \phi_\theta$ is the *transformed moving image.* It is the result of applying the transformation ϕ_θ to the moving image J. The mathematical symbol $\circ$ denotes composition, and will be used throughout this chapter. The composition notation is convenient for avoiding having to write the arguments of images and transforms. $J \circ \phi_\theta$ is a continuous image on the domain Ω, given by

  $$(J \circ \phi_\theta)(\mathbf{x}) = J(\phi_\theta(\mathbf{x}))\,.$$

- θ represents the set of *transformation parameters*. These parameters determine the function ϕ_θ, i.e., different values of θ correspond to different spatial transformations ϕ_θ. The objective of DIR is to determine the transformation parameters that yield the optimal spatial transformation matching the input images I and J.

  – In the first class of DIR methods that we will study, called *parametric DIR*, θ will be a finite vector of unknown values. These coefficients are used to define a spatial transformation ϕ_θ using a system of basis functions. For example, θ may consist of displacement vectors placed on a grid of control points, from which the continuous transformation ϕ_θ is derived using interpolation. When the set of transformation parameters is finite, the registration problem (1) can be solved using multivariate optimization methods, e.g., using gradient descent or Newton's method.

  – *Non-parametric* DIR methods, which we will study later, have infinitely many transformation parameters, that is to say, θ is modeled as an unknown function, and solving the registration problem (1) requires tools of variational calculus. Sometimes in non-parametric methods we do not even distinguish between θ and ϕ_θ, i.e., the unknown parameters and the spatial transformation are the same thing. In this case we can rewrite (1) omitting θ altogether:

  $$\phi^* = \arg \min_{\phi:\Omega\to\Omega} E[\phi; I, J]$$

- Θ represents the set of all possible values that the transformation parameters θ may assume.

- $\mu[\bullet, \bullet]$ is the *image dissimilarity metric*. It takes as input two images and returns a value that measures how anatomically dissimilar the two images are. The greater the dissimilarity, the larger the value of the metric. Image dissimilarity metrics are discussed in **Chapter XXX**. In this Chapter, we will primarily use the *mean squared intensity difference* metric, which has the simple form

  $$\mu_{\text{MSID}}[I, J] = \int_\Omega \left[I(\mathbf{x}) - J(\mathbf{x})\right]^2 \, \mathrm{d}\mathbf{x}\,.$$

  However, other metrics defined in in **Chapter XXX** are also commonly used in DIR, often with better results than the MSID metric.

- $\rho[\theta]$ is the regularization term. It can be used to impose soft constraints on the transformation ϕ_θ. Since DIR problems have a very large number of degrees of freedom in specifying ϕ_θ, some form of regularization is necessary to make the optimization problem (1) well-posed and to prevent unrealistic transformations, e.g. ones that fold the space Ω onto itself. The term $\rho[\theta]$ is designed by the algorithm developer in such a way that when the transformation ϕ_θ becomes unrealistic, the value of $\rho[\theta]$ becomes large, thus penalizing this particular choice of parameters θ. By minimizing the weighted sum of the metric and regularization terms (with weighting provided by the constant λ), registration algorithms seek a tradeoff between good matching of anatomy that is achieved when the metric is minimized, and avoiding unrealistic transformations. Regularization terms are different in different algorithms, but typically involve differential terms that encourage solutions ϕ_θ to be smooth and bounded. In some applications, regularization is used to imbue ϕ_θ with specific physical or geometric properties, such as local preservation of area or volume.

1.3 Outline of the Chapter

In the rest of the chapter, we will explore several representative DIR techniques.

- **Key Mathematical Concepts** will be reviewed in Section 2. Specifically, we will focus on the concept of diffeomorphisms, a subset of spatial transformations that are smooth and invertible.

- **Parametric DIR Methods** will be discussed in Section 3. These methods use interpolation techniques such as *radial basis functions* or *b-splines* to describe continuous transformations ϕ_θ using a finite set of parameters θ.

- **Optical Flow and Related Non-Parametric DIR Methods** will be discussed in Section 4. Optical flow is not strictly a registration technique, but a way of computing velocities of moving objects in pairs of images taken a short time apart. However, the concept of optical flow is very useful to understanding popular non-parametric registration techniques like the Demons algorithm [25] and its more recent variants [27, 5].

- **Large Deformation Diffeomorphic Metric Mapping (LDDMM)**, which will be discussed in Section 5, is a category of non-parametric methods that allow DIR to be formulated as an energy minimization problem while constraining the solutions ϕ to be smooth and invertible (diffeomorphic). LDDMM also provides a mathematically sound way to measure distances and perform statistics in the infinite-dimensional space of images, and for this reason it has found extensive applications in studies of anatomical variability.

- **Emerging DIR Methods** will be briefly discussed in Section 6. These include recent applications of deep learning to solve DIR problems.

2 Mathematical Concepts

This section introduces several mathematical concepts that will be used throughout the rest of the chapter. These concepts primarily involve spatial transformations.

2.1 General Concepts for Transformations

2.1.1 Transformation vs. Displacement.

As noted above, a spatial transformation $\phi : \Omega \to \Omega$ is a mapping of the image domain Ω onto itself. A special case is the *identity transformation* Id that maps every point to itself:

$$\mathrm{Id}(\mathbf{x}) = \mathbf{x} \,. \tag{2}$$

It is sometimes convenient to express spatial transformations as a sum of the identity transformation and a vector-valued function called a *displacement:*

$$\phi(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x}) \,.$$

The distinction between transformations and displacements is simple, but important. The function $\phi(\mathbf{x})$ describes the *new position* of point $\mathbf{x}$ after the transformation. The displacement $\mathbf{u}(\mathbf{x})$ describes the *change* in the position of a point $\mathbf{x}$ under the transformation ϕ. For example, in two dimensions, ϕ might map the point $(2.0, 2.1)$ to a new coordinate $(2.4, 1.9)$. The corresponding displacement is $(0.4, -0.2)$. We will use Greek letters ϕ, ψ to denote transformations, and boldface Roman letters $\mathbf{u}, \mathbf{v}$ to denote displacements.

The reason why it is convenient to represent transformations as displacements is that a given displacement value such as $(0.4, -0.2)$ can be interpreted the same way everywhere in the domain Ω. By contrast, a transformation value (2.4,1.9) has a different interpretation depending on $\mathbf{x}$. In DIR algorithms, we often want to restrict the transformation ϕ to be close to identity over some regions, particularly near the boundary of Ω. Enforcing this condition as $\mathbf{u}(\mathbf{x}){\simeq}\mathbf{0}$ during operations such as smoothing or interpolation is simpler algorithmically than enforcing $\phi(\mathbf{x}) \simeq \mathbf{x}$. Most computer implementations of DIR algorithms represent transformations in memory as displacements.

2.1.2 Composition.

Given a pair of transformations $\phi, \psi : \Omega \to \Omega$, the composition of ϕ and ψ is a new transformation denoted $\phi \circ \psi$, defined as:

$$(\phi \circ \psi)(\mathbf{x}) = \phi(\psi(\mathbf{x})) \,.$$

Intuitively, $\phi{\circ}\psi$ is the transformation we get by first applying the transformation ψ and then applying the transformation ϕ. It is easy (and a good exercise) to show that the displacement $\mathbf{w}$ corresponding to the transformation $\phi \circ \psi$ has the form

$$\mathbf{w}(\mathbf{x}) = \mathbf{v}(\mathbf{x}) + \mathbf{u}(\mathbf{x} + \mathbf{v}(\mathbf{x})) \,,$$

where $\mathbf{u}$ and $\mathbf{v}$ are the displacements corresponding to the transformations ϕ and ψ, respectively.

2.1.3 Computer Implementation.

When representing transformations in the computer we will almost always represent them as displacements. The n components of the displacement $\mathbf{u}$ can be stored as n-dimensional images. The composition of two transformations can be computed using built-in functions for interpolation available in many software libraries. The following MATLAB listings shows functions for applying a spatial transformation to an image and for composing two spatial transformations.

```
function J_warped = warp_image_2D(I, J, ux, uy)
%           I: reference/fixed image
%           J: moving image
%           ux, uy: x and y components of transformation phi
%     J_warped: the transformed (warped) moving image

% Create a coordinate grid over the reference image domain
[x,y] = ndgrid(1:size(I,1), 1:size(I,2));

% Interpolate the moving image
J_warped = interpn(J, px + ux, py + uy, '*linear', 0)

function [wx,wy] = compose_transformations_2D(I, ux, uy, vx, vy)
%           I: reference/fixed image
%           ux, uy: x and y components of transformation phi
%           vx, vy: x and y components of transformation psi
%           wx, wy: x and y components of phi composed with psi

% Create a coordinate grid over the reference image domain
[x,y] = ndgrid(1:size(I,1), 1:size(I,2));

% Perform the composition
wx = vx + interpn(ux, px + vx, py + vy, '*linear', 0)
wy = vy + interpn(uy, px + vx, py + vy, '*linear', 0)
```

2.1.4 Jacobian Determinant

An important property of a transformation around a point is whether it expands or compresses space. For example, with aging, gray matter structures in the brain compress (due to loss of neurites) while the cerebrospinal fluid in the brain expands to fill in the space. The local expansion or compression caused by a spatial transformation ϕ is measured by the determinant of the *Jacobian* matrix:

$$|D\phi| = \det \begin{bmatrix} \frac{\partial \phi^1}{\partial x_1} & \cdots & \frac{\partial \phi^1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \phi^n}{\partial x_1} & \cdots & \frac{\partial \phi^n}{\partial x_n} \end{bmatrix} . \tag{3}$$

To interpret $|D\phi|$ geometrically, consider a small equilateral triangle (or in 3D, a tetrahedron) $T_{\mathbf{x},\epsilon}$ with sides of length ϵ that is centered around a point $\mathbf{x} \in \Omega$. The transformation ϕ will map the corners of $T_{\mathbf{x},\epsilon}$ to new locations, forming a new triangle/tetrahedron $\phi T_{\mathbf{x},\epsilon}$. The ratio of the areas/volumes of $\phi T_{\mathbf{x},\epsilon}$ and $T_{\mathbf{x},\epsilon}$ describes the local expansion/contraction of space around $\mathbf{x}$. The Jacobian determinant $|D\phi|$ is simply the limit of this area/volume ratio as $\epsilon \to 0$.[1] When $|D\phi| = 1$, there is no change in volume, when $0 < |D\phi| < 1$, the transformation is associated with shrinking around $\mathbf{x}$, and when $|D\phi| > 1$, it is associated with expansion. Negative values of $|D\phi|$ occur when the transformation ϕ causes the space to fold.

2.2 Diffeomorphisms[2]

It is impossible to discuss modern approaches to image registration without mentioning diffeomorphisms. A spatial transformation $\phi : \Omega \to \Omega$ is *a diffeomorphism* if ϕ is differentiable and has a differentiable inverse $\phi^{-1} : \Omega \to \Omega$, i.e.,

$$\phi^{-1}(\phi(\mathbf{x})) = \mathbf{x} \quad \forall \mathbf{x} \in \Omega\,.$$

In many applications of DIR, we require the solutions to be diffeomorphisms. There are many reasons for this, some of which include:

- DIR is often used to map per-pixel measurements from the moving image to the fixed image (this data may be in the form of segmentations, statistical maps such as those derived from functional MRI experiments, etc.). In the course of analysis it is often necessary to also be able to map the information in the opposite direction, from fixed to moving. For example, in functional MRI analysis, we may map information from individual brains to a template brain in order to perform group analysis and identify brain regions associated with a particular cognitive function, such as finger tapping. We then want to be able to map these regions back into individual subjects' brains for secondary analyses (e.g., brain network analyses). When DIR produces diffeomorphic transformations, we are guaranteed that informaiton can be mapped in both directions.

- In many research studies, the transformations derived by DIR are used as the source of features for statistical analysis. In *deformation-based morphometry (DBM)* [12], DIR is performed between a pair of scans of the same individual taken some time apart. The Jacobian determinant of the transformation ϕ is then analyzed to identify anatomical regions where tissue has shrunk or expanded over time. Such measures of tissue

[1] Technically, this statement is correct only when using a signed definition of the area/volume. If a transformation changes the order of the vertices of the triangle (e.g., from clockwise to counterclockwise) or the order of the faces of the tetrahedron, then the area/volume is taken with a negative sign.

[2] This section is important for understanding the diffeomorphic registration techniques discussed in XXX. Readers interested in a quick overview of deformable registration can skip the rest of the section.

loss and expansion can be sensitive to subtle atrophy that occurs in the brain in some neurodegenerative diseases, such as Alzheimer's. However, when a transformation ϕ is not invertible, there exist points in Ω where $|D\phi| \leq 0$, which corresponds to the folding or self-intersection of the space, and cannot be interpreted in an anatomically meaningful way. Therefore, in DBM studies, we aim to restrict solutions of DIR to diffeomorphisms.

- When modeling the motion of organs such as the beating heart, the underlying anatomy is known a priori to deform in a way that is diffeomorphic, so it is only natural to require the solutions of DIR to be diffeomorphisms.

Examples of diffeomorphic and non-diffeomorphic transformations in two dimensions are shown in Fig. XXX. An important observation is that *smoothness is not a sufficient condition for a diffeomorphism*: it is possible to have a very smooth transformation that does not have an inverse, just like the function $f(x) = (2x-1)^2$ does not have an inverse on the interval $x \in [0,1]$.

2.2.1 Group of Diffeomorphisms

An important property of diffeomorphisms is that they *form a group under composition*, called Diff. Specifically, it is easy to verify that the four conditions of a set being a group are met:

- **Closure**: $\phi \circ \psi \in \text{Diff}$ for all $\phi, \psi \in \text{Diff}$.

- **Identity**: There exists an element $\text{Id} \in \text{Diff}$ that satisfies $\phi \circ \text{Id} = \text{Id} \circ \phi = \phi$ for all $\phi \in \text{Diff}$.

- **Associativity**: For all $\phi, \psi, \chi \in \text{Diff}$, $(\phi \circ \psi) \circ \chi = \phi \circ (\psi \circ \chi)$.

- **Inverse**: For all $\phi \in \text{Diff}$, there exists $\phi^{-1} \in \text{Diff}$ such that $\phi \circ \phi^{-1} = \phi^{-1} \circ \phi = \text{Id}$.

One important property of diffeomorphisms is that the Jacobian determinant of $\phi \in \text{Diff}$ is non-negative for all $\mathbf{x} \in \Omega$. Conversely, if ϕ is a differentiable vector-valued function that satisfies $|D\phi| > 0$ for all $\mathbf{x} \in \Omega$, then ϕ is a diffeomorphism.

2.2.2 Small Transformations

While it is easy to define what a diffeomorphism is, it is less obvious how one would go about generating diffeomorphisms. A useful building block are the so-called *small transformations*. Let $\mathbf{u} : \Omega \to \mathbb{R}^n$ be a vector-valued function that is differentiable, bounded and has a bounded derivative, i.e., $|\mathbf{u}| + |D\mathbf{u}| \leq M$ for all $\mathbf{x} \in \Omega$ and some $M > 0$. Let us denote the space of all such functions $\mathcal{U}$.

Then for a sufficiently small $\epsilon > 0$, the transformation

$$\psi(\mathbf{x}) = \mathbf{x} + \epsilon \mathbf{u}(\mathbf{x})$$

has a positive Jacobian determinant for every $\mathbf{x} \in \Omega$, and therefore is a diffeomorphism. We call ψ a small transformation because it only slightly deviates from Id.

Now, since a composition of two diffeomorphisms is a diffeomorphism, we can use small transformations to build up larger transformations. Given a set of functions $\mathbf{u}_1, \ldots, \mathbf{u}_n$ in $\mathcal{U}$, we can compose the corresponding small transformations $\psi_i(\mathbf{x}) = \mathbf{x} + \epsilon \mathbf{u}_i(\mathbf{x})$, yielding a "large" transformation ϕ that is diffeomorphic:

$$\phi = \psi_n \circ \psi_{n-1} \circ \ldots \circ \psi_1 \,. \tag{4}$$

The key observation here is that it is relatively easy to generate vector-valued functions that are differentiable and bounded (e.g., this can be done by interpolating a discrete set of vectors placed in Ω). The recipe above provides a way to use such functions to generate diffeomorphic transformations, which form a highly non-linear subspace of the space of all transformations. The general idea of generating diffeomorphic transformations by composing small transformations is used throughout the later portion of this chapter.

2.2.3 Flow Ordinary Differential Equation

Another closely related way to generate diffeomorphisms in $\mathbb{R}^n$ is by means of the flow ordinary differential equation (ODE) [1]. Instead of defining a discrete set of functions $\mathbf{u}_1, \ldots, \mathbf{u}_n$ in $\mathcal{U}$ as we did above, let us consider a function $\mathbf{v}(\mathbf{x}, t) : \Omega \times [0, 1] \rightarrow \mathbb{R}^n$, where t is a time dimension. Let $\mathbf{v}$ be differentiable in t and let $\mathbf{v}(\bullet, t) \in \mathcal{U}$ for all t. We will call this function a *time-varying velocity field*. Think of it as a function that describes the wind velocity in a region of space over time. Now consider a large collection of particles (like dust or seed pods) that are being carried around by the wind. Let the trajectory of the particle located at coordinate $\mathbf{x}$ at time 0 particle be denoted by the function $\phi(\mathbf{x}, t)$. The relationship between the particle trajectories and the wind field is expressed by the following *flow ODE*:

$$\frac{\mathrm{d}\phi}{\mathrm{d}t}(\mathbf{x}, t) = \mathbf{v}(\phi(\mathbf{x}, t), t) \,, \quad \text{subj. to } \phi(\mathbf{x}, 0) = \mathbf{x} \,. \tag{5}$$

The equation states that the trajectory of every particle is tangent to the wind direction at its current location. The existence and uniqueness theorem for first-oder ODEs implies that the solutions of this equation are diffeomorphisms for all $t \in [0, 1]$. [2]. Hence, the flow ODE provides us with another recipe to generate diffeomorphisms given only moderately constrained vector-valued functions. We can think of the flow ODE as a "black box" that takes as its input a smooth and bounded time-varying velocity field and outputs a diffeomorphic transformation $\phi(\mathbf{x}, 1)$.

There is a close relationship between the flow ODE and our previous recipe of generating diffeomorphisms by composing a sequence of small deformations. Let us approximate the solution of the flow ODE using Euler's method. Euler's method is based on the approximation $\frac{\mathrm{d}\phi}{\mathrm{d}t}(\mathbf{x}, t) \simeq \frac{\phi(x, t+\delta t) - \phi(x, t)}{\delta t}$, where $\delta t > 0$

is a small step size. Plugging this approximation into the ODE leads to the following recurrence:

$$\phi(\mathbf{x}, t+\delta t) = \phi(\mathbf{x}, t) + \delta t \cdot \mathbf{v}(\phi(\mathbf{x}, t), t)\,,$$

which can be rewritten as follows:

$$\phi(\mathbf{x}, t+\delta t) = \psi(\phi(\mathbf{x}, t)) \qquad \text{where } \psi(\mathbf{x}) = \mathbf{x} + \delta t \cdot \mathbf{v}(\mathbf{x}, t)\,.$$

The function ψ is a small transformation (since $\mathbf{v}$ is smooth and bounded, and δt is a small value), and thus for every t, the function $\phi(\mathbf{x}, t)$ is simply a sequence of compositions of small transformations, just as in (4) above.

2.2.4 Exponential Map

Consider the natural exponential function e^x from calculus. It has a property that it maps the entire real line to elements of the group of positive real numbers under multiplication. One way to define the natural exponential function is as follows:

$$e^x = \lim_{m\to\infty} \left(1 + \frac{x}{m}\right)^m\,.$$

This definition of exponentiation (i.e., taking an element around the identity element of the group, and repeatedly applying the group operation) can be generalized to other groups. For example, the group of rotation matrices in $\mathbb{R}^3$ under the operation of matrix multiplication has an equivalent exponential map that maps vectors in $\mathbb{R}^3$ to rotation matrices:

$$\exp(\mathbf{w}) = \lim_{m\to\infty} \left\{ I_{3\times 3} + \frac{1}{m} \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} \right\}^m\,.$$

The beauty of this construct is that it allows us to parameterize a non-linear manifold of rotation matrices, which are orthonormal matrices and have only three degrees of freedom despite having nine elements, using elements of the three-dimensional Euclidean space.

The equivalent notion of an exponential map can also be defined for the group of diffeomorphisms:

$$\exp(\mathbf{u}) = \lim_{m\to\infty} \underbrace{(\mathrm{Id} + \frac{1}{m}\mathbf{u}) \circ \ldots \circ (\mathrm{Id} + \frac{1}{m}\mathbf{u})}_{m\,\text{times}}\,. \tag{6}$$

As before, $\exp(\mathbf{u})$ involves the composition of (many) small transformations, and therefore is a diffeomorphism. One of the attactive properties of the exponential map is that if $\phi = \exp(\mathbf{u})$ then the inverse transformation is given by $\phi^{-1} = \exp(-\mathbf{u})$; as is also the case for the natural exponential function and for exponentiation in the group of rotation matrices.

In practice, computing the exponential map as in (6) appears prohibitive. However, an excellent approximation of the exponential map is provided by the recursive *scaling and squaring algorithm* [3]. Observe than when m in (6) is even, due to the associative property of composition, $\exp(\mathbf{u})$ can be computed as the composition of two identical terms, each involving $m/2$ compositions. If we let m be a power of 2, i.e., $m = 2^K$ for some integer K, then the term involving $m/2$ compositions can be computed as the composition of two identical terms each involving $m/4$ compositions, and so on. This leads to the following recursive algorithm:

$$\begin{aligned} \psi_1 &= \mathrm{Id} + \frac{1}{2^K}\mathbf{u} \\ \psi_k &= \psi_{k-1} \circ \psi_{k-1} \qquad \text{for } k = 2, \ldots, K\,. \end{aligned}$$

With this simple algorithm, an excellent approximation of $\exp(\mathbf{u})$ can be obtained in very few iterations (practical values of K are between 5 and 8).

3 Parametric DIR Methods

3.1 Theory.

Parametric techniques were among the earliest solutions to the problem of DIR [7, 21], but they remain widely used today. Several advanced software packages implementing parametric DIR exist, such as `elastix [?]` and `IRTK [21]`. Parametric DIR methods are among the most accessible to newcomers in the field because they do not require advanced mathematics, such as calculus of variations, to describe.

The central idea in parametric DIR methods is to model spatial transformations ϕ_θ using families of continuous functions weighted by scalar coefficients. In most general form, let $f_1 : \Omega \to \mathbb{R}, \ldots, f_K : \Omega \to \mathbb{R}$ be some “family” of smooth functions. Then in parametric DIR, we model ϕ_θ as a weighted sum of these functions with vector-valued coefficients $\theta_1 \in \mathbb{R}^n, \ldots, \theta_K \in \mathbb{R}^n$:

$$\phi_\theta(\mathbf{x}) = \mathbf{x} + \sum_{k=1}^{K} \theta_k f_k(\mathbf{x})\,. \tag{7}$$

Let us note some properties of this transformation:

- When all the coefficients θ_k are zeros, ϕ_θ is the identity transform $\phi_\theta(\mathbf{x}) = \mathbf{x}$.
- The transformation ϕ_θ is a smooth (infinitely differentiable) function as long as each f_k is infinitely differentiable.
- The transformation ϕ_θ is completely determined by the set of coefficients θ_k.
- With the appropriate choice of functions f_k and appropriately large number of functions/coefficients K, any smooth and bounded spatial transformation on domain Ω can be approximated by the weighted sum (7) with arbitrary accuracy.

Which family of functions should we choose? Perhaps the simplest of all functions used in DIR is the family of *radial basis functions,* defined as follows. Let Ω be a rectangular domain in $\mathbb{R}^n$ and let $\mathbf{c}_1 \in \Omega, \ldots, \mathbf{c}_K \in \Omega$ be a set of points in this domain, called *control points.* For example, control points may lie on a uniform grid, as in Figure XXX. Associate each control point $\mathbf{c}_k$ with a continuous function $f_k(\mathbf{x})$:

$$f_k(\mathbf{x}) = \psi(|\mathbf{x} - \mathbf{c}_k|)\,,$$

where $\psi : \mathbb{R} \to \mathbb{R}$ is some continuous function called the *kernel function.* A common choice is the Gaussian $\psi(r) = e^{-0.5r^2}$. Notice that f_k above only depends on the distance from $\mathbf{x}$ to the control point $\mathbf{c}_k$, i.e., it is radially symmetric around $\mathbf{c}_k$. Such functions are called *radial basis functions.* Examples of radial basis functions around two different control points are plotted in Figure XXX. Radial basis functions were used in one of the earliest DIR papers [7], and are just one example of a suitable family of functions for parametric DIR. Other choices are discussed later in this section.

So far, what we have done was to model spatial transformations on the domain Ω using a finite number of coefficients $\theta = \{\theta_1 \ldots, \theta_K\}$. For every choice of the nK real values in θ, we obtain a different spatial transformation. We can now formulate the registration problem as the problem of finding the set of values θ that minimizes the dissimilarity between image I and the transformed image $J \circ \phi_\theta$:

$$\theta^* = \operatorname*{arg\,min}_{\theta \in \mathbb{R}^{nK}} E[\phi_\theta; I, J] = \operatorname*{arg\,min}_{\theta \in \mathbb{R}^{nK}} \mu[I, J \circ \phi_\theta] + \lambda \rho[\theta]\,.$$

For simplicity, consider the mean squared intensity difference (MSID) metric and no regularization. Then the energy can be written down in complete form as:

$$\theta^* = \operatorname*{arg\,min}_{\theta \in \mathbb{R}^{nK}} \int_\Omega [I(\mathbf{x}) - J(\phi_\theta(\mathbf{x}))]^2 \;\mathrm{d}\mathbf{x} \;= \operatorname*{arg\,min}_{\theta \in \mathbb{R}^{nK}} \int_\Omega \left[I(\mathbf{x}) - J(\mathbf{x} + \sum_{k=1}^{K} \theta_k f_k(\mathbf{x})) \right]^2 \;\mathrm{d}\mathbf{x}\,. \tag{8}$$

This is an unconstrained energy minimization problem involving nK unknowns, and can be solved using gradient-based numerical optimization methods. For completeness, let us derive the partial derivative of the energy function E with respect to the m-th coefficient vector θ_m :

$$\frac{\partial E}{\partial \theta_m} = 2 \int_\Omega \left[I(\mathbf{x}) - J(\mathbf{x} + \sum_{k=1}^{K} \theta_k f_k(\mathbf{x})) \right] \left[-\nabla J(\mathbf{x} + \sum_{k=1}^{K} \theta_k f_k(\mathbf{x})) \cdot f_m(\mathbf{x}) \right] \;\mathrm{d}\mathbf{x}\,,$$

or, in more compact form:

$$\frac{\partial E}{\partial \theta_m} = -2 \int_\Omega [I - J \circ \phi_\theta]\, [(\nabla J \circ \phi_\theta) \cdot f_m] \;\mathrm{d}\mathbf{x}\,. \tag{9}$$

3.2 Basic Implementation.

Implementing the above DIR strategy on real discrete images is not difficult. First, let us consider how to discretize the problem. Images I and J will be discrete images (n-dimensional arrays of intensity values). The spatial transformation ϕ_θ may be represented as a set of $n$separate n-dimensional images, one for each of the components of ϕ_θ (i.e., $\phi_\theta^x, \phi_\theta^y, \ldots$). The main operations in (8) and (9) above are the computation of $\phi_\theta(\mathbf{x})$ from the coefficients θ, the computation of $J \circ \phi_\theta$, the computation of $\nabla J \circ \phi_\theta$, and integration over the image domain. Computing $J \circ \phi_\theta$ in the discrete setting involves sampling the values of J at positions ϕ_θ. In MATLAB, this is accomplished using the command `interpn`, which interpolates the values of an array at arbitrary sample points. Program listing XXX gives complete code for parametric DIR using Gaussian radial basis functions.

3.3 Real-World Approaches

While the approach sketched out above is relatively simple, the same basic principle is followed by the state-of-the-art parametric DIR techniques. Modern techniques employ more efficient ways to parameterize spatial transformations, more advanced image dissimilarity metrics, incorporate multi-resolution schemes, and provide options for regularizing the transformation ϕ_θ.

Choice of basis functions. A significant limitation of the simple approach based on radial basis functions is that ϕ_θ is expensive to compute, requiring $O(KN)$ operations for an image with N pixels or voxels. This is because each of the K basis functions f_k is evaluated at each voxel in (7). In fact, after discretization, the computation of ϕ_θ can be expressed in matrix form:

$$\Phi_\theta = \mathrm{Id} + F \cdot \theta$$

where F is a $N \times nK$ matrix whose columns are the basis functions f_k evaluated at all voxels, Φ_θ is an $N \times n$ matrix describing the transformation ϕ_θ at every voxel, and Id is an $N \times n$ matrix representing the identity transformation, i.e., each row in Id contains the coordinates of the corresponding voxel. To make this computation more efficient, real-world algorithms use basis functions $\{f_k\}$ that have *finite support,* meaning that each f_k is non-zero on only a small subset Ω_k of the image domain Ω. In this case, the computational cost of computing ϕ_θ is $O(Km)$, where $m \ll N$ is the average size of the domains $\{\Omega_k\}$. This is equivalent to having F be a sparse matrix.

Rueckert et al. [21] popularized the use of cubic *b-splines* as the basis for parametric DIR. These bi-cubic (or in 3D, tri-cubic) polynomials are parameterized in terms of a uniform grid of control points, and have finite support, such that each control point influences a 4×4 grid cell region around it, and has no effect on ϕ_θ outside of this region. Techniques like `IRTK [21], NiftyReg [18]` and `elastix [?]` use *b-splines* to represent spatial transformations.

Support for advanced metrics. Important practical considerations in the development of parametric DIR tools is support for multiple image dissimilarity metrics, such as the mutual information metric [28, 14, 24] (for across-modality registration), patch-wise cross-correlation metric [9, 4], and other advanced metrics. When using more advanced metrics, the computation of the partial derivatives $\frac{\partial E}{\partial \theta_m}$ becomes more complex. When DIR is implemented using arbitrary metric $\mu[I, J]$, i.e.,

$$E[\phi_\theta; I, J] = \mu[I, J \circ \phi_\theta]\,,$$

the partial derivatives have the form

$$\frac{\partial E}{\partial \theta_m} = \int_\Omega (\delta_J \mu[I, J \circ \phi_\theta]) \cdot (\nabla J \circ \phi_\theta) f_m \, \mathrm{d}\mathbf{x}$$

where $\delta_J \mu : \Omega \to \mathbb{R}$ denotes the partial Gateaux derivative of μ with respect to its second input. Whereas in rigid and affine registration derivatives of the registration objective function can be approximated numerically, in parametric DIR there are many more parameters, and numeric computation of derivatives becomes prohibitively expensive. For mutual information and cross-correlation metrics, efficient schemes for computing gradients exist, but are outside of the scope of this book.

Multi-resolution schemes. Multi-resolution is one of the most effective tools in real-world DIR. A multi-resolution pyramid of images is generated from $I$and J, typically by subsampling by the factor 2 in each dimension several times. For example, if I has dimensions 256×256, the pyramid may contain images of sizes 32×32, 64×64, 128×128, and 256×256. Parametric DIR is performed at the lowest resolution first, and the resulting low-resolution transformation ϕ_θ is interpolated by the factor of 2 to initialize parametric DIR at the next resolution level, and so on. The number of control points K typically also scales with the resolution. Multi-resolution not only helps DIR converge faster but helps avoid local minima at early stages of registration by matching more prominent features in the images that are visible at low resolution. When subsampling images for multi-resolution schemes, it is important to low-pass filter them appropriately to avoid excessive aliasing due to sampling under the Nyquist frequency (**Chapter XXX**). An advantage of low-pass filtering is that numerical approximation of ∇J is more accurate, and hence ∇E is more accurate as well, leading to better optimization convergence.

Regularization. Additional terms can be added to the parametric DIR objective function to impose desired properties on ϕ_θ. For instance, it may be desirable to obtain transformations that approximately preserve local area or volume of structures. Mathematically the amount by which the transformation ϕ_θ expands or compresses the area/volume of an infinitecimal region around a point $\mathbf{x} \in \Omega$ is given by the *determinant of the Jacobian matrix* of ϕ_θ (denoted $|D\phi_\theta|$). When $|D\phi_\theta| < 1$, the transformation compresses space around a point, and when $|D\phi_\theta| > 1$, it expands space. A regularization term may then be formulated to penalize the deviation of $|D\phi_\theta|$ from 1, thus favoring transformations that do not compress or expand space too much.

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^{n \cdot K}} \mu[I, J \circ \phi_\theta] + \lambda \int_\Omega \left(|D\phi_\theta| - 1\right)^2 \, \mathrm{d}\mathbf{x}.$$

An interesting point is that in the parametric DIR framework, it is difficult to enforce strong constraints on the transformation ϕ_θ, i.e., requiring ϕ_θ to be strictly incompressible ($|D\phi_\theta| = 1$ at every point). Since ϕ_θ is formed as a weighted sum of basis functions f_k,

4 Optical Flow and Related Non-Parametric Methods

Optical flow is a class of techniques used to measure the velocity of moving objects in video sequences. For example, in a self-driving car application, the continuous video feed from the car's camera could be used to determine the velocity of surrounding vehicles and pedestrians. In medical imaging, optical flow can be used to characterize the dynamics of moving organs, such as the beating heart in cine MRI. This application of optical flow is explored in Chapter XXX. But optical flow also serves as a great way to introduce non-parametric DIR algorithms.

4.1 Classical Optical Flow

Let $\mathcal{I}(\mathbf{x}, t) : \mathbb{R}^n \times [0, t_{\max}] \to \mathbb{R}$ denote an image changing over time, i.e., a video sequence, where time t spans the interval $[0, t_{\max}]$. The objective of optical flow is to determine at each point in space and time a vector $\vec{v}(\mathbf{x}, t)$ that describes the instantaneous velocity of the object located at point $\mathbf{x}$ at time t.

The classical optical flow method was proposed by Horn and Schunck in 1981 [11] and is summarized below.

4.1.1 Preservation of Intensity Assumption

The optical flow approach is based on the following assumption: *as objects move in the video sequence, their intensity remains nearly constant* (Fig. 1). In general computer vision problems, this is a strong assumption because the shading of objects changes as they move through the scene and because objects may become occluded by other objects. In 3D medical image sequences this assumption is more reasonable, although noise and partial volume artifacts introduce small changes in the intensity of objects as they move.

To express the assumption of intensity preservation mathematically, let us consider two images taken a short time apart, $I(\mathbf{x}, t_0)$ and $I(\mathbf{x}, t_0 + \delta t)$. Suppose that a point at location $\mathbf{x}_0$ at time t_0 has moved to a new location $\mathbf{x}_0 + \delta\mathbf{x}$ at time

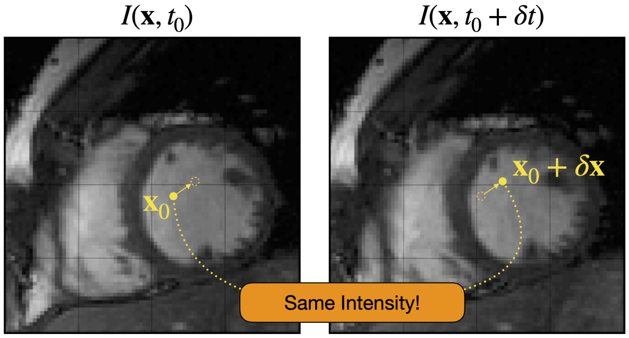
$I(\mathbf{x}, t_0)$
$I(\mathbf{x}, t_0 + \delta t)$
$\mathbf{x}_0$
$\mathbf{x}_0 + \delta\mathbf{x}$
Same Intensity!

Figure 1: Intensity preservation in optical flow. The anatomical location at location $\mathbf{x}_0$ at time t_0 has moved to position $\mathbf{x}_0 + \delta\mathbf{x}$ at time $t_0 + \delta t$. We assume that the image intensity associated with this anatomical location is the same at time t_0 and at time $t_0 + \delta t$.

$t_0 + \delta t$. Since, under our assumption, the intensity of the point $\mathbf{x}_0$ is preserved at its new location $\mathbf{x}_0 + \delta\mathbf{x}$, we have

$$I(\mathbf{x}_0 + \delta\mathbf{x}, t_0 + \delta t) = I(\mathbf{x}_0, t_0)\,.$$

The expression on the left of the equality can be expanded using Taylor's series as follows:

$$\begin{aligned}
&I(\mathbf{x}_0 + \delta\mathbf{x}, t_0 + \delta t) = \\
&\qquad = I(\mathbf{x}_0, t_0 + \delta t) + (\delta\mathbf{x})^T I_{\mathbf{x}}(\mathbf{x}_0, t_0 + \delta t) + \mathcal{O}(\delta^2) = \\
&\qquad\qquad = I(\mathbf{x}_0, t_0) + (\delta\mathbf{x})^T I_{\mathbf{x}}(\mathbf{x}_0, t_0) + (\delta t) I_t(\mathbf{x}_0, t_0) + \mathcal{O}(\delta^2)\,,
\end{aligned}$$

where $I_{\mathbf{x}}$ and I_t denote the partial derivatives of I with respect to $\mathbf{x}$ and t, and $\mathcal{O}(\delta^2)$ represents second and higher-order terms involving $\delta\mathbf{x}$ and δt. Subtracting the first of the two equations above from the second, we obtain

$$0 = (\delta\mathbf{x})^T I_{\mathbf{x}}(\mathbf{x}_0, t_0) + (\delta t) I_t(\mathbf{x}_0, t_0) + \mathcal{O}(\delta^2)\,,$$

and dividing through by Δt, we have

$$\left(\frac{\delta\mathbf{x}}{\delta t}\right)^T I_{\mathbf{x}}(\mathbf{x}_0, t_0) = -I_t(\mathbf{x}_0, t_0) + \mathcal{O}(\delta)\,.$$

Now let us consider the limit of this expression as $\Delta t \to 0$. The expression $\frac{\Delta\mathbf{x}}{\Delta t}$ is the ratio between the distance an object travels over a short time interval Δt and the said time interval, and as $\Delta t \to 0$, it converges to the instantaneous velocity of the object, $\vec{v}(\mathbf{x}_0, t_0)$, which is exactly the quantity that we seek to

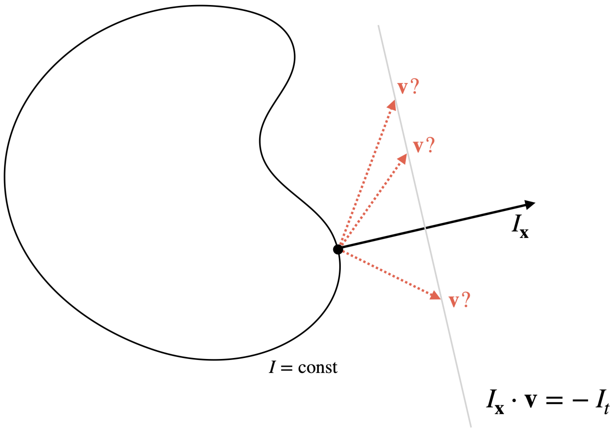

Figure 2: The intensity preservation constraint implies that the velocity vector v is constrained to lie on a line (plane in 3D), but does not uniquely determine it.

estimate in optical flow. Also, the limit of $\mathcal{O}(\delta)$ as $\delta t \to 0$ is zero. Hence, in the limit, the expression above simplifies to

$$\vec{v}(\mathbf{x}_0, t_0)^T I_\mathbf{x}(\mathbf{x}_0, t_0) = -I_t(\mathbf{x}_0, t_0)\,,$$

or just

$$\vec{v}^T I_\mathbf{x} = -I_t\,. \tag{10}$$

This linear relationship between velocity $\vec{v}$, image gradient $I_\mathbf{x}$ and image time derivative I_t is called the *optical flow constraint.* Geometrically, this constraint specifies that at every point $\mathbf{x}_0$, the tip of the velocity vector $\vec{v}$ must lie on a certain line (in 2D), plane (in 3D) or hyper-plane (in ND) that is orthogonal to the image gradient at $\mathbf{x}_0$, as illustrated in Fig. 2. Importantly, the restriction of $\vec{v}$ onto a certain line/plane is not enough to determine $\vec{v}$ uniquely (except if $n = 1$). At each point $(\mathbf{x}_0, t_0)$ the velocity vector $\vec{v}(\mathbf{x}_0, t_0)$ may take as many different values as there are different points in $\mathbb{R}^{n-1}$. Additional assumptions are required in order to uniquely determine $\vec{v}(\mathbf{x}, t)$.

4.1.2 Smoothness Assumption and Formulation As Energy Minimization

Given that the intensity preservation assumption is not enough to determine $\vec{v}(\mathbf{x}, t)$, Horn and Schunck [11] proposed an additional assumption: that the velocity field $\vec{v}(\mathbf{x}, t)$ is smooth across the image domain. Rather than formulating this assumption as a hard constraint, they reformulate the optical flow problem as an energy minimization problem in which violations of the optical flow assumption and smoothness assumption are both penalized.

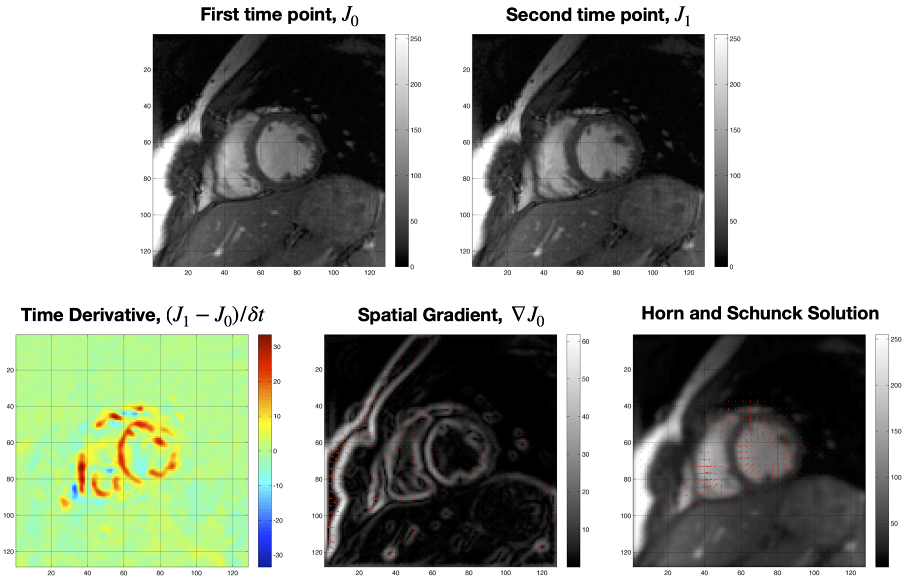

Figure 3: Horn and Schunck optical flow method applied to a pair of images. The red arrows in the last panel show the estimated velocity field.

Going forward, let us focus on a particular moment in time, t_0 and let $\vec{v}(\mathbf{x}) = \vec{v}(\mathbf{x}, t_0)$. Horn and Schunck's [11] optical flow minimization is formulated as

$$\vec{v}^* = \underset{\vec{v}\in\mathcal{V}^n}{\arg\min}\, E[\vec{v}]\,, \quad \text{where} \quad E[\vec{v}] = \int_\Omega \left(\vec{v}^T I_\mathbf{x} + I_t\right)^2 + \lambda^2 \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{\partial v^i}{\partial x_j}\right)^2 \mathrm{d}\mathbf{x}. \tag{11}$$

The left part of the integrand drives $\vec{v}$ to satisfy the intensity preservation assumption (i.e., the more the assumption is violated, the greater the value of the energy E becomes). The right hand of the integrand drives $\vec{v}$ to be smooth, since discontinuities in $\vec{v}$ are associated with large values of the partial derivatives $\frac{\partial v_i}{\partial x_j}$. The positive scalar λ is a coefficient of the method, and is used to balance between the two terms; larger values of λ result in smoother solutions.

The energy functional (11) is minimized using variational techniques (**Section XXX**). The Euler-Lagrange equation for E (see Problem XXX) is

$$\mathcal{L}[\vec{v}] = \left(\vec{v}^T I_\mathbf{x} + I_t\right) I_\mathbf{x} - \lambda^2 \triangle \vec{v} = \mathbf{0}\,,$$

where $\triangle$ is the Laplacian operator. Like in most variational problems involving images, we solve the problem (1) iteratively by taking small steps in the direction of $-\mathcal{L}(\vec{v})$.

4.1.3 Basic Implementation

The discretization and implementation of the optical flow algorithm is relatively easy. First, we must discretize over time. Instead of a continuous image $I(\mathbf{x}, t)$, we consider a pair of n-dimensional images $J_0 = I(\mathbf{x}, t_0)$ and $J_1 = I(\mathbf{x}, t_0 + \delta t)$. The first-order approximation of the time derivative of $I(\mathbf{x}, t)$ is

$$I_t \simeq \frac{J_1 - J_0}{\delta t},$$

and the spatial gradient of $I(\mathbf{x}, t)$ at time t_0 is simply given by $I_{\mathbf{x}} = \nabla J_0$. Hence,

$$\mathcal{L}[\vec{v}] \simeq \left(\vec{v}^T \nabla J_0 + \frac{J_1 - J_0}{\delta t} \right) \nabla J_0 - \lambda^2 \triangle \vec{v}\,.$$

Calculating this quantity involves approximation of the second derivatives of $\vec{v}$ using finite differences (**Section XXX**). Code listing XXX gives an example implementation of optical flow for two-dimensional images, and Fig. 3 gives examples of velocity estimation from a pair of cardiac cine MRI frames. Note how the velocity field seems to capture the compression of the left ventricle and even some of the heart's twisting motion.

4.1.4 Limitations of Optical Flow

A major limitation of optical flow is that the optical flow constraint loses its influence over homogeneous regions in the image. When a large region $R \subset \Omega$ of the image has nearly constant intensity, both the image gradient $I_{\mathbf{x}}$ and the image time derivative I_t vanish on the interior of R. As the result, for any choice of $\vec{v}$ over the interior of R, the first term in (11), $\left(\vec{v}^T I_{\mathbf{x}} + I_t\right)^2$ will be nearly zero, and hence, the minimization of E is almost exclusively driven by the regularization term over the interior of R. This results in surprising, and almost certainly wrong, estimates of velocity over homogeneous regions, with greater velocity at the boundary of the constant intensity regions than on the interior. This is demonstrated in Fig. XXX, where several homogeneous objects are displaced rigidly in the image plane. The recovered velocity field is not at all consistent with rigid motion.

4.2 Non-Parametric DIR using Iterative Optical Flow

4.2.1 Conceptual DIR Algorithm based on Optical Flow

While optical flow can be useful in medical imaging applications for describing the dynamics of moving organs, it is not considered a DIR technique per se. Optical flow only computes an infinitesimal velocity between two timeframes in a video sequence, whereas DIR seeks a deformation of the domain Ω that would match up corresponding anatomical locations between two images. These corresponding locations may be quite far apart from each other, requiring a potentially “large” deformation to match them.

However, it turns out that optical flow can be used as a building block for an effective non-parametric DIR algorithm. Consider the following iterative algorithm:

Algorithm 1 A skeleton of a DIR algorithm based on Optical Flow.

```
1. Let $I$ be the fixed image and $J$ the moving image. Let $J^0 = J$.
2. Iterate for $i = 0, 1, \dots$ :
    (a) Compute optical flow $\vec{v}^i$ between images $I$ and $J^i$
    (b) Let $J^{i+1}(\mathbf{x}) = J^i(\mathbf{x} + \epsilon \vec{v}^i)$ where $\epsilon$ is a (small) constant
    (c) Terminate if the difference between $J^{i+1}$ and $J^i$ is below threshold
```

The basic idea of this algorithm is as follows. At every iteration, the velocity field $\vec{v}^i$ obtained by performing optical flow between images I and J^i can be interpreted as the direction in which objects in I would move if I and J^i were frames from a video. Even if I and J^i are not frames from a video but have similar content (e.g. both are images of brains), the vectors $\vec{v}^i$ may be expected to point from locations in I toward similarly-appearing locations in J^i. This is, of course, a strong assumption. If this assumption is true, then moving the locations in I a small amount along the vectors $\vec{v}^i$ would place them closer to their corresponding anatomical counterparts in J^i. Conversely, deforming the image J^i by the "small" transformation $J^{i+1}(\mathbf{x}) = J^i(\mathbf{x} + \epsilon\vec{v}^i)$ would result in the new image J^{i+1} being better anatomically matched up with I than J^i. Repeating this process iteratively until convergence would result in an image J^M that is most similar anatomically to I.

In practice it turns out that the strong assumption above is actually not unreasonable, and the general algorithm outlined above works quite well for DIR. However, the algorithm above is just a conceptual sketch, not a practical DIR algorithm. One crucial limitation is that at each iteration the image J^i is interpolated, which over just a few iterations would completely degrade the image due to aliasing (Chapter XXX). Also, the algorithm does not actually yield any spatial transformation, just a deformed image J^M. These two limitations can be easily addressed by a small modification to the algorithm:

Algorithm 2 Conceptual DIR algorithm based on Optical Flow.

```
1. Let $I$ be the fixed image and $J$ the moving image. Let $J^0 = J$ and $\phi^0 = \mathrm{Id}$.
2. Iterate for $i = 0, \dots, M$:
    (a) Compute optical flow $\vec{v}^i$ between images $I$ and $J^i$
    (b) Let $\psi^i = \mathrm{Id} + \epsilon\vec{v}^i$, i.e., $\psi^i(\mathbf{x}) = \mathbf{x} + \epsilon\vec{v}^i(\mathbf{x})$
    (c) Let $\phi^{i+1} = \phi^i \circ \psi^i$, i.e., $\phi^{i+1}(\mathbf{x}) = \phi^i(\mathbf{x} + \epsilon\vec{v}^i(\mathbf{x}))$
    (d) Let $J^{i+1} = J^0 \circ \phi^{i+1}$
    (e) Terminate if the difference between $J^{i+1}$ and $J^i$ is below threshold
```

Note that the two algorithms are identical in terms of the outputs J^i when the input images are continuous functions. However, when it comes to practical implementation, the difference is significant! The first algorithm interpolates the moving image repeatedly, whereas in the second algorithm, the moving image is only interpolated once, and the spatial transformations ϕ^i are composed repeatedly, which involves interpolation. It turns out that the latter is less problematic, particularly because in practice additional smoothing will be applied to these spatial transformations to limit aliasing.

4.2.2 The Demons Algorithm

The idea of using optical flow as a step in a DIR algorithm was pioneered by J.P. Thirion [25] in an approach he called "Demons", in reference to a famous thought experiment in Maxwell's theory of thermodynamics. One obvious limitation that Thirion's approach overcame is the computational inefficiency of the algorithm above: *at each iteration it requires us to solve the optical flow problem, which itself requires iterative optimization!* This double iteration would be impractical for real-world registration problems.

Thirion proposed to estimate optical flow in a simpler and more efficient way than in the Horn an Schunck method [11]. The optical flow intensity preservation condition (10) can be interpreted as saying that points on any iso-intensity contour of the image at time t (i.e., points that satisfy $I(\mathbf{x}, t) = \alpha$ for some scalar α) should map to points somewhere on the corresponding contour in the image at time $t + \delta t$ (Fig. 4a). Thirion proposed to strengthen this constraint, by requiring these points on the contour at time t to map to the *closest point* on the contour at time $t + \delta t$ (Fig. 4b). This is equivalent to requiring $\vec{v}$ to be perpendicular to the contour, or parallel to $I_\mathbf{x}$, since the gradient of a function is orthogonal to its iso-intensity contours. Introducing the additional constraint $\vec{v} \parallel I_\mathbf{x}$ makes it possible to solve directly for $\vec{v}$:

$$y \begin{cases} \vec{v}^T I_\mathbf{x} = -I_t \\ \vec{v} \parallel I_\mathbf{x} \end{cases} \qquad \Longrightarrow \qquad \vec{v} = \frac{-I_t}{|I_\mathbf{x}|^2} I_\mathbf{x} \,.$$

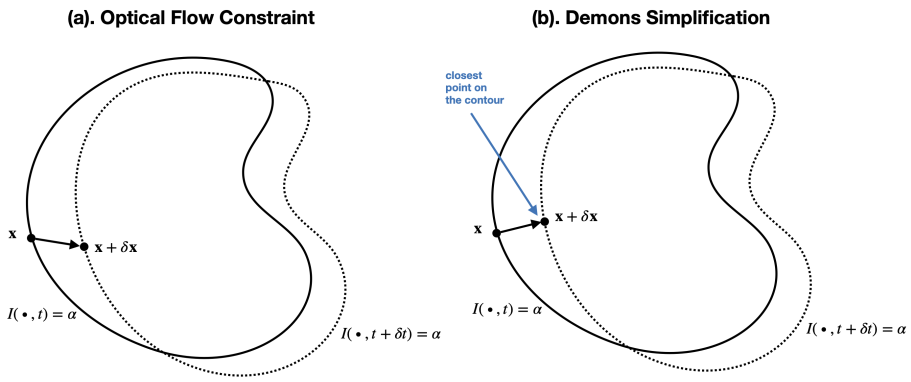
(a). Optical Flow Constraint
(b). Demons Simplification
x
$\mathbf{x} + \delta\mathbf{x}$
$I(\bullet, t) = \alpha$
$I(\bullet, t + \delta t) = \alpha$
closest point on the contour
x
$\mathbf{x} + \delta\mathbf{x}$
$I(\bullet, t) = \alpha$
$I(\bullet, t + \delta t) = \alpha$

Figure 4: Additional optical constraint proposed in the Demons method [25]. The optical flow condition dictates that points on a given iso-intensity contour in I at time t (bold curve, i.e., a set of all points that have intensity α at time t) has to map to some point on the corresponding contour in the image at time $t + \delta t$. The Demons method introduces an additional constraint, that points at time t should map onto the *closest point* on the corresponding contour at time $t + \delta t$.

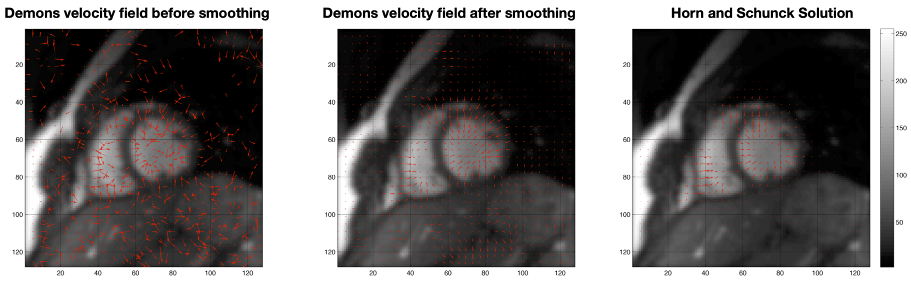
Demons velocity field before smoothing
Demons velocity field after smoothing
Horn and Schunck Solution
20
40
60
80
100
120
20
40
60
80
100
120
250
200
150
100
50

Figure 5: Illustration of Demons velocity computation [25], contrasted with Horn an Schunck [11] optical flow. The velocity field is computed between cine MRI frames from Fig. 3. The Demons velocity field is shown before and after smoothing in (12).

However, the expression on the right is asymptotic at locations with near constant intensity, where $|I_\mathbf{x}| \simeq 0$. It also results in very noisy velocity fields. To overcome this, Thirion proposed modifying the expression for $\vec{v}$ as follows:

$$\vec{v} = G_\sigma * \begin{cases} \frac{-I_t}{|I_\mathbf{x}|^2+|I_t|^2} I_\mathbf{x} & \text{if } |I_\mathbf{x}|^2+|I_t|^2 > \epsilon \\ \mathbf{0} & \text{otherwise} \end{cases} . \tag{12}$$

This formulation reduces the number of points where $\vec{v}$ is unstable, and results in zero velocity over regions of constant intensity. The convolution with the Gaussian kernel G_σ regularizes the velocity field, somewhat similar to the Horn and Schunk approach, but without optimization (Fig. 5). This results in a very fast computation of optical flow. The following algorithm is approximately equivalent to the original Demons formulation:

Algorithm 3 The Compositional Demons Algorithm [25].

```
1. Let $I$ be the fixed image and $J$ the moving image. Let $\phi^0 = \text{Id}$.
2. Iterate for $i = 0, \dots, M$:
    (a) Let $J^i = J \circ \phi^i$
    (b) Compute optical flow $\vec{v}^i$ between images $I$ and $J^i$ as
        $\vec{v}^i = G_\sigma * \begin{cases} -\frac{J^i - I}{|\nabla J^i|^2+|J^i-I|^2} \nabla J^i & \text{if } |\nabla J^i|^2 + |J^i - I|^2 > \epsilon \\ \mathbf{0} & \text{otherwise} \end{cases} .$
    (c) Let $\psi^i = \text{Id} + \epsilon \vec{v}^i$, i.e., $\psi^i(\mathbf{x}) = \mathbf{x} + \epsilon \vec{v}^i(\mathbf{x})$
    (d) Let $\phi^{i+1} = G_\tau * (\phi^i \circ \psi^i)$
    (e) Terminate if $|J \circ \phi^{i+1} - J^i|$ is below threshold
```

There are a several observations to be made about the algorithm:

- The expression in 2(b) is derived from (12) by approximating I_t with $J^i - I$ and approximating $I_\mathbf{x}$ with ∇J^i. It is also possible to approximate $I_\mathbf{x}$ with ∇I or with $\frac{1}{2}(\nabla J^i + \nabla I)$. How these different approximations affect the behavior of Demons is explored in [27].

- In practice, the scaling factor ϵ in step 2(c) can be chosen in different ways. For example, it can be adapted at each iteration so that $\sup_{\mathbf{x}\in\Omega} |\vec{v}^i| \le \epsilon'$, where ϵ' is fixed across iterations (typically on the order of 1/4 to 1/2 of the voxel dimension).

- Observe that convolution with a Gaussian filter is performed in two places: in step 2(b), when regularizing the optical flow, and in step 2(d), when combining the accumulated transformation ϕ^i with the small transformation ψ^i. Vercauteren et al., [27] describe these as "fluid-like" and "diffusion-like" regularization. In practice, both kinds of regularization are needed to obtain a good quality registration.

- Unlike many algorithms in computer vision and medical image analysis, the Demons algorithm is not formulated in a variational fashion, i.e., in terms of some energy functional $E[\phi]$ that is minimized. However, Vercauteren et al., [27] show that it is possible to interpret Demons as energy minimization by introducing an additional concept of a correspondence map.[3]

The Demons algorithm is a very fast non-parametric registration algorithm. It only involves very basic image processing operations such as composition of transformation fields (interpolation), computation of image gradients, and convolution with a Gaussian kernel. It is also highly parallelizable. A MATLAB implementation of the Demons algorithm is provided in Listing XXX.

4.2.3 Log-Domain Diffeomorphic Demons Algorithm

The transformations produced by the compositional Demons algorithm are not guaranteed to be diffeomorphic. In practice, they often are because at each iteration the current tranformation ϕ^i is composed with a small transformation ψ^i. However, the convolution with the Gaussian kernel G_τ in step 2(d) may break the diffeomorphic property, and also the numerical implementation may eventually lead to solutions that are in fact not diffeomorphic. Another limitation of compositional Demons is that it only provides the forward transformation ϕ^i that maps points in the fixed image to corresponding points in the moving image, but the inverse transformation is not generated. Although it is possible to estimate the inverse of a diffeomorphic transformation *post hoc* doing so without significant numerical error is nontrivial and computationally expensive [XXX].

Vercauteren et al. (2008) [26] proposed a modification to the compositional Demons algorithm that guarantees diffeomorphic solutions and outputs both forward and inverse transformations. This approach is called *log-domain Demons.*

[3] The correspondence map $c : \Omega \to \Omega$ can be interpreted as a more "raw", unregularized mapping of pixels in the fixed image I to the pixels in the moving image J. Vercauteren et al. [27] define an energy functional that involves both the unknown transformation and the unknown correspondence map:

$$E[c, \phi] = \lambda_1 \|I - J \circ c\|^2 + \lambda_2 \|\phi - c\| + \lambda_3 \|\nabla \phi\|^2 ,$$

where $\lambda_1, \lambda_2, \lambda_3$ are coefficients corresponding to the three terms in the energy: the intensity similarity (how well the correspondence map matches pixels in the two images); the fidelity of the transformation (how well the transformation ϕ agrees with the correspondence map c); and the regularization term (how noisy is the transformation ϕ). The authors then show that the Demons iterative algorithm correponds to alternatively minimizing $E[c, \phi]$ with respect to c and ϕ.

It takes advantage of the fact that diffeomorphic transformations can be generated from smooth and bounded vector-valued functions using the using the exponential map (see Section 2.2). The main modification from compositional Demons is that instead of directly updating the spatial transformation ϕ^i at each iteration of the algorithm, Vercauteren's method performs updates in the "logarithmic domain", i.e., it updates the vector-valued function $\mathbf{u}^i$, such that the desired spatial transformation is given by

$$\phi^i = \exp(\mathbf{u}^i)\,.$$

The crux of the algorithm is to figure out how to perform the update $\mathbf{u}^i \to \mathbf{u}^{i+1}$ in the logarithmic domain given the optical flow velocity field $\epsilon\vec{v}^i$, such that $\exp(\mathbf{u}^{i+1}) = \exp(\mathbf{u}^i) \circ (\mathrm{Id} + \epsilon\vec{v})$. In principle, doing so requires the inverse of the exponential map (i.e., the logarithm map), which is not well-defined for the infinite-dimensional group of diffeomorphisms, and also very expensive to compute. However, Vercauteren et al. show that the following approximation, based on the Baker-Campbell-Hausdorff formula from the theory of Lie groups [20], is sufficiently accurate in practice:

$$\mathbf{u}^{i+1} \simeq \mathbf{u}^i + \epsilon\,\vec{v}^i + \frac{\epsilon}{2}[\mathbf{u}^i, \vec{v}^i]\,,$$

where the expression $[\mathbf{u}, \mathbf{v}]$ is called the *Lie bracket*, and has the form:

$$[\mathbf{u}, \mathbf{v}] = (D\mathbf{u})\mathbf{v} - (D\mathbf{v})\mathbf{u}\,,$$

and $D\mathbf{u}$ denotes the Jacobian matrix of $\mathbf{u}$. This update takes place directly in the logarithmic domain, and is relatively inexpensive computationally, compared to the update in steps 2(c,d) in the compositive Demons algorithm. Another change in the Vercauteren's algorithm is that regularization with the diffusion kernel G_τ is performed directly in the logarithmic domain. The final algorithm is as follows:

Algorithm 4 The Log-domain Demons Algorithm [26, 27].

```
1. Let $I$ be the fixed image and $J$ the moving image. Let $\mathbf{u}^0(\mathbf{x}) = \mathbf{0}$.
2. Iterate for $i = 0, \ldots, M$:
    (a) Let $J^i = J \circ \phi^i$
    (b) Compute optical flow $\vec{v}^i$ between images $I$ and $J^i$ as in Alg. 3, Step 2(b).
    (c) Let $\mathbf{u}^{i+1} = G_\tau * \left(\mathbf{u}^i + \epsilon\,\vec{v}^i + \frac{\epsilon}{2}[\mathbf{u}^i, \vec{v}^i]\right)$
    (d) Let $\phi^{i+1} = \exp(\mathbf{u}^{i+1})$ and $(\phi^{i+1})^{-1} = \exp(-\mathbf{u}^{i+1})$.
    (e) Terminate if $|J \circ \phi^{i+1} - J^i|$ is below threshold.
```

The log-domain Demons algorithm is computationally efficient. The added computational cost compared to compositional Demons is the computation of the Lie bracket (requires computing first derivatives of $\mathbf{u}^i$ and $\vec{v}^i$) and the handful of additional compositions involved in the scaling and squaring algorithm. Because it is fast, easy to implement, guarantees diffeomorphic transformations, and is very effective in practice, the log-domain Demons algorithm has become one of the most widely used non-parametric registration approaches, and variations of this method are implemented in various registration software packages [xxx]. A number of extensions and improvements to the baseline log-Demons algorithm have been made, including:

- The algorithm can be extended to work with other intensity similarity metrics, such as patch cross-correlation (Sec. XXX) and mutual information (Sec. XXX). For an arbitraty intensity similarity metric $\mu[I, J]$, the optical flow computation in Step 2(b) is replaced with the vector field

  $$\vec{v}^i = -G_\sigma * \nabla_J \mu(I, J_i)\,,$$

  which like the optical flow field corresponds to a small transformation that would pull J_i in the direction that lowers the metric value. In practice, Demons-like registration using the cross-correlation metric is particularly effective for many types of medical images.

- The algorithm can be made symmetric, such that the result of the registration with I as the fixed image and J is the moving image is the inverse of the result of the registration with J as the fixed image and I as the moving image. A symmetric extension of log Demons is described in the original paper [26] and requires only a minor modification to the algorithm, based on computing the optical flow in both directions (fixed to moving and moving to fixed), and a minor increase in computational cost.

- Mansi et al. (2011) [15] proposed a modification of log-domain Demons that constrains transformations to be incompressible ($|D\phi^i| = 1$) over regions of the fixed image. This constraint is useful when performing registration in anatomical structures with known biomechanical properties, such as deforming cardiac tissue.

5 Large Deformation Diffeomorphic Metric Mapping (LDDMM)

5.1 LDDMM Formulation

We now turn our attention to another highly influential approach to non-parametric registration with diffeomorphic constraints, called *Large Deformation Diffeomorphic Metric Mapping (LDDMM)* [6, 17]. Unlike Demons and related registration techniques, LDDMM is formulated explicitly as a problem of minimizing an energy functional over the space of diffeomorphic transformations, Diff. The energy functional is a weighted sum of the similarity metric between the fixed and moving images (with weight $\frac{1}{\sigma^2}$, following the notation in [6]) and a regularization term on the spatial transformation:

$$\phi^* = \arg\min_{\phi \in \mathrm{Diff}} E[\phi]\,, \quad \text{where} \quad \mathrm{E}[\phi] = \frac{1}{\sigma^2}\,\mu(\phi; I, J) + \mathrm{Reg}(\phi)\,. \tag{13}$$

This kind of *variational* formulation of the registration problem has advantages over the *"greedy"* formulation in Demons, which does not directly associate solutions with an energy value. For example, it provides a principled way to compare the fidelity of different solutions $\phi_1, \phi_2, \ldots$, and as we will see below, it actually provides a sensible way to extend the familiar concepts of "distance" and "average" to the highly non-linear space of medical images.

The problem formulated in (13) requires us to search for solutions in the space Diff. Because of the non-linear nature of Diff, doing so is very difficult when representing the transformations ϕ directly, i.e., as vector fields. For example, if ϕ_1 and ϕ_2 are in Diff, there is no guarantee that transformations in the form $w_1\phi_1 + w_2\phi_2$ (where w_1 and w_2 are scalars) will be in Diff. Constraining solutions ϕ to stay in Diff during the minimization of $E[\phi]$ in (13) would be difficult and computationally expensive.

However, as we have seen in Section 2.2.3, diffeomorphic transformations can be generated from less constrained representations (i.e., differentiable and bounded vector fields) in various ways. LDDMM leverages the fact that a family of diffeomorphic transformations $\phi(\mathbf{x}, t)$ parameterized by time $t \in [0, 1]$ can be generated from a time-varying differentiable and bounded *velocity field* $\mathbf{v}(\mathbf{x}, t) : \Omega \times [0, 1] \to \mathbb{R}^n$ using the flow ODE (5). For a given point $\mathbf{x}$ in the space of the fixed image I, the function $\phi(\mathbf{x}, t)$ describes a non-linear trajectory of that point as it moves tangentially to $\mathbf{v}(\mathbf{x}, t)$, like a dust particle in the wind. Let the diffeomorphic transformation $\phi_1(\mathbf{x}) = \phi(\mathbf{x}, 1)$ denote the final position of $\mathbf{x}$. LDDMM reformulates the registration problem (13) as the problem of finding a velocity field $\mathbf{v}$ for which the final transformation $\phi_1(\mathbf{x})$ best matches corresponding points between fixed and moving images, while also subject to a regularization term:

$$\begin{aligned} \mathbf{v}^* = \arg\min_{\mathbf{v} \in \mathcal{V}} E[\mathbf{v}]\,, \quad \text{where} \quad \mathrm{E}[\mathbf{v}] &= \frac{1}{\sigma^2}\,\mu(\phi_1; I, J) + \mathrm{Reg}(\mathbf{v})\,, \\ \text{and} \quad \frac{\mathrm{d}\phi}{\mathrm{d}t}(\mathbf{x},\mathrm{t}) &= \mathbf{v}(\phi(\mathbf{x},\mathrm{t}),\mathrm{t})\,, \quad \text{subj. to } \phi(\mathbf{x},0) = \mathbf{x} \end{aligned} \tag{14}$$

Unlike (13), the search space $\mathcal{V}$ for this energy minimization problem consists of smooth and bounded vector-valued velocity fields, and is much simpler in structure than Diff. For example, given two candidate solutions $\mathbf{v}_1, \mathbf{v}_2 \in \mathcal{V}$, their weighted sums $w_1\mathbf{v}_1 + w_2\mathbf{v}_2$ are also members of the search space $\mathcal{V}$.

Regularization Term Let us now consider the regularization term in (14). We can learn a lot about the properties of the final transformation $\phi_1(\mathbf{x})$ by examining the properties of $\mathbf{v}(\mathbf{x}, t)$. Following our wind/dust analogy, the faster the wind is, the farther the particles are likely to travel; and the smoother the field wind is, the more simple the trajectories of the particles are likely to be. This suggests that it should be possible to measure the regularity of the trajectories $\phi(\mathbf{x}, t)$ and of the final transformation $\phi_1(\mathbf{x})$ using quantities derived directly from the velocity field $\mathbf{v}$.

Let us consider for a moment the squared Euclidean or L^2 norm of $\mathbf{v}$ as a potential regularization term. It is defined as follows:

$$\|\mathbf{v}\|^2_{L^2} = \int\limits_0^1 \int\limits_\Omega |\mathbf{v}(\mathbf{x}, t)|^2 \, \mathrm{d}\mathbf{x} \, \mathrm{d}t \; = \int\limits_0^1 \int\limits_\Omega < \mathbf{v}(\mathbf{x}, t), \mathbf{v}(\mathbf{x}, t) > \, \mathrm{d}\mathbf{x} \, \mathrm{d}t \, .$$

This norm integrates the magnitude of the velocity across space and time dimensions. The faster the "wind" in our analogy, the greater $\|\mathbf{v}\|^2_{L^2}$ will be. Consequently, the particles described by $\phi(\mathbf{x}, t)$ will travel farther. If we use $\mathrm{Reg}(\mathbf{v}) = \|\mathbf{v}\|^2_{L^2}$ as the regularization term in (14), then we would favor solutions that have shorter trajectories, which is reasonable. However, $\|\mathbf{v}\|^2_{L^2}$ does nothing to penalize the noisiness of the velocity field, so we are likely to get noisy solutions.

In LDDMM [6], the authors associate the space of time-varying velocity fields $\mathcal{V}$ with a different non-Euclidean norm called the *Sobolev norm.* The Sobolev norm of a vector field measures both its magnitude and its regularity, and is defined as

$$\|\mathbf{v}\|^2_{\mathcal{V}} = \|\mathcal{L}\mathbf{v}\|^2_{L^2} = \int\limits_0^1 \int\limits_\Omega |\mathcal{L}\mathbf{v}(\mathbf{x}, t)|^2 \, \mathrm{d}\mathbf{x} \, \mathrm{d}t \, ,$$

where $\mathcal{L}$ is a differential operator. In LDDMM, this operator is given the following form:

$$\mathcal{L}\mathbf{v} = \gamma \mathbf{v} - \alpha \Delta \mathbf{v} \, ,$$

where Δ is the Laplacian operator, applied separately to each component of $\mathbf{v}$.[4]; $\gamma > 0$ and $\alpha > 0$ are scalar weights that control the extent to which the magnitude and regularity components contribute to $\|\mathbf{v}\|^2_{\mathcal{V}}$. As is common in the LDDMM literature, we will use the term *kinetic energy* to describe the combined magnitude and regularity of the time-varying velocity field $\mathbf{v}$ captured by the Sobolev norm $\|\mathbf{v}\|^2_{\mathcal{V}}$. In particular, the term "kinetic energy of a diffeomorphic flow ϕ_t" refers to the Sobolev norm of the velocity field $\mathbf{v}$ from which ϕ_t is generated by the flow ODE.

Figure XXX shows examples of diffeomorphic transformations obtained by solving the flow ODE for different synthetically generated velocity fields $\mathbf{v}$, with corresponding values of the kinetic energy. The figure illustrates that the regularization $\mathrm{Reg}(\mathbf{v}) = \|\mathbf{v}\|^2_{\mathcal{V}}$ favors transformations that are not too large and not too irregular.

[4] Specifically, $\Delta \mathbf{v} = \begin{bmatrix} \frac{\partial^2 v^1}{\partial x_1^2} + \ldots + \frac{\partial^2 v^1}{\partial x_n^2} \\ \vdots \\ \frac{\partial^2 v^n}{\partial x_1^2} + \ldots + \frac{\partial^2 v^n}{\partial x_n^2} \end{bmatrix}$.

Similarity Metric. In the original formulation of LDDMM [6], the authors used the sum of squared differences (SSD) similarity metric, due to its mathematical simplicity. Other metrics can also be used in conjunction with LDDMM, but SSD is best suited for understanding the approach. Elsewhere in this chapter, we computed the similarity metric in the space of the fixed image I, with the moving image J warped into the space of I:

$$\mu(\phi; I, J) = \int\limits_{\Omega} |I - J \circ \phi|^2 \, \mathrm{d}\mathbf{x} \, .$$

In the original LDDMM paper [6], the authors chose to compute the similarity metric in the space of the moving image instead, with the fixed image warped back into the space of the moving image using the inverse of the final transformation ϕ_1:

$$\mu(\phi_1; I, J) = \int\limits_{\Omega} \left| I \circ \phi_1^{-1} - J \right|^2 \, \mathrm{d}\mathbf{x} \, .$$

Since it does not really matter whether the similarity is computed in the space of the fixed or moving image, we will follow the convention of the original LDDMM paper here. The complete formulation of the LDDMM registration algorithm is the following energy minimization problem:

$$\begin{aligned} \mathbf{v}^* = \arg\min_{\mathbf{v} \in \mathcal{V}} E[\mathbf{v}; I, J] \, , \quad \text{where} & \\ E[\mathbf{v}; I, J] = \|\mathbf{v}\|_{\mathcal{V}}^2 + \frac{1}{\sigma^2} \int\limits_{\Omega} \left| I \circ \phi_1^{-1} - J \right|^2 \, \mathrm{d}\mathbf{x} \, , & \\ \text{and } \frac{\mathrm{d}\phi}{\mathrm{d}t}(\mathbf{x},\mathrm{t}) = \mathbf{v}(\phi(\mathbf{x},\mathrm{t}),\mathrm{t}) \, , \quad \text{subj. to } \phi(\mathbf{x},0) = \mathbf{x} \, . & \end{aligned} \tag{15}$$

5.2 Approaches to Solving the LDDMM Problem

5.2.1 Variational Approach

The minimization of the energy functional $E[\mathbf{v}; I, J]$ in (15) can be solved using standard variational techniques, i.e., by writing down the the Euler-Lagrange equation for E. The derivation is quite involved (largely due to the presence of the ODE in the formulation) and we direct interested readers to the original paper [6]. For a given value of $t \in [0, 1]$, the Euler-Lagrange equation is:

$$2\mathbf{v}_t - \mathcal{K}\left(\frac{2}{\sigma^2} \left| D\phi_{t,1} \right| \, \nabla I_t (I_t - J_t) \right) = 0 \, , \tag{16}$$

where

$\mathbf{v}_t(\mathbf{x}) = \mathbf{v}(\mathbf{x}, t)$	is the velocity field at a fixed time t
$\phi_t(\mathbf{x}) = \phi(\mathbf{x}, t)$	is the spatial transformation at a fixed time t
$\phi_{t,1} = \phi_1 \circ \phi_t^{-1}$	is the transformation between points at timepoint t and the corresponding points at timepoint 1
$\phi_{t,0} = \phi_0 \circ \phi_t^{-1} = \phi_t^{-1}$	is the transformation between points at timepoint t and the corresponding points at timepoint 0
$I_t = I \circ \phi_{t,0}$	is the fixed image warped into the timepoint t
$J_t = J \circ \phi_{t,1}$	is the moving image warped into the timepoint t
$\|D\phi_{t,1}\|$	is the determinant of the Jacobian of $\phi_{t,1}$
$\mathcal{K}$	is the operator satisfying the relation $\mathcal{K}(\mathcal{L}^\dagger\mathcal{L})\mathbf{u} = \mathbf{u}$ for all vector fields $\mathbf{u} \in \mathcal{U}$, where $\dagger$ denotes the adjoint. $\mathcal{K}$ is the inverse of the operator $\mathcal{L}^\dagger\mathcal{L}$, and does not have a closed form. In practice it is computed by solving the partial differential equation $\mathcal{L}^\dagger\mathcal{L}\mathbf{w} = \mathbf{u}$ for $\mathbf{w} \in \mathcal{U}$.

Given the Euler-Lagrange equation, the optimization problem is discretized and solved sequentially as most other variational optimization problems in this book, i.e., by making small updates in the opposite direction of the left hand side of (16). The following additional considerations are involved in implementing the LDDMM approach computationally:

- The time dimension is discretized into a fixed number of steps (e.g., 40 or 100).

- During optimization, it is necessary to compute and maintain in memory the transformations $\phi_t(\mathbf{x})$ and $\phi_t(\mathbf{x})^{-1}$ for each discrete time step t. These transformations are computed using numerical schemes for approximating the flow ODE (5). While Euler's method can be used for this, Beg et al. [6] use a more accurate two-step semi-Lagrangian scheme [23].

- The partial differential equation corresponding to the application of the operator $\mathcal{K}$ in (16) is solved numerically using Fourier transform. It is also possible to change the definition of $\mathcal{L}$ so that $\mathcal{K}$ corresponds to smoothing with a Gaussian kernel. This does not appear to substantially change the solutions in practice.

- Beg et al. [6] show that, theoretically, the minimizer of (15) has "constant speed" (i.e., $\int_\Omega |\mathcal{L}\mathbf{v}_t|^2 \, \mathrm{d}\mathbf{x} = \text{const}$ as a function of time). In practice, the sequential scheme for minimizing (15) does not yield solutions with constant speed, and Beg et al. [6] recommend reparameterizing the solution $\mathbf{v}(\mathbf{x}, t)$ with respect to t to maintain constant speed after every few update iterations.

Overall, the LDDMM approach yields itself well to computational implementation and is highly parallelizable. Several efficient GPU-based implementations have been developed [XXX]. However, it is quite memory intensive because of the need to maintain several data arrays of the size $\mathcal{O}(NT)$ where N is the size of I in voxels and T is the number of time steps.

5.2.2 Geodesic Shooting

It turns out that we can minimize the LDDMM energy $E[\mathbf{v}; I, J]$ without having to optimize in the high-dimensional (space+time) space $\mathcal{V}$. It turns out that a velocity field $\mathbf{v} \in \mathcal{V}$ that minimizes $E[\mathbf{v}; I, J]$ is entirely determined by the initial velocity $\mathbf{v}_0$. To understand this consider the following analogy:

- Suppose you want to walk between two villages A and B located in a mountainous terrain. A path that you would follow can be described by its coordinates $\mathbf{x}(t)$, with $\mathbf{x}(0) = A$ and $\mathbf{x}(1) = B$. It can also be described by the your velocity vector at every time, $\mathbf{v}(t)$, with the path itself given by the ODE
$$\mathbf{x}(0) = A \quad \text{and} \quad \dot{\mathbf{x}}(t) = \mathbf{v}(t)\,.$$
The length of this path is given simply by
$$L[\mathbf{v}] = \int_0^1 |\mathbf{v}(t)|\,\mathrm{d}t\,.$$

- Now suppose you wanted to walk from A to B covering the shortest possible distance. This shortest path is called the *minimizing geodesic.* On a plane it is a straight line, but on a curved manifold, it is a curve that locally "looks like" a straight line. We can write the problem of finding the minimizing geodesic in terms of $\mathbf{v}$ as follows:
$$\mathbf{v}^* = \arg\min_{\mathbf{v}} L[v] \quad \text{subject to} \;\; \mathbf{x}(1) = B\,,$$
or equivalently,
$$\mathbf{v}^* = \arg\min_{\mathbf{v}} L[v] + \frac{1}{\sigma^2}|\mathbf{x}(1) - B|^2\,. \tag{17}$$
This is analogous to the LDDMM energy minimization described above, and similarly requires us to search for a time-parameterized velocity function $\mathbf{v}(t)$.

- Let us now elaborate on the notion that a geodesic "looks like" a straight line locally. Let X be some point that the curve $\mathbf{x}(t)$ passes through, and let T_X be the tangent plane to the Earth's surface at X. The curvature of the projection of curve $\mathbf{x}(t)$ onto T_X at X is called *geodesic curvature.* A geodesic is a curve whose geodesic curvature is zero at every point. Mathematically, this property is expressed by the second-order ODE:
$$\frac{\mathrm{d}^2 x_k}{\mathrm{d}t^2} + \sum_{i,j} \Gamma^k_{ij} \frac{\mathrm{d}x_i}{\mathrm{d}t}\frac{\mathrm{d}x_j}{\mathrm{d}t} = 0\,, \quad \text{for} \;\; i,j,k \in \{1,2,3\}, \tag{18}$$

or equivalently

$$\frac{\mathrm{d}v_k}{\mathrm{d}t} + \sum_{i,j} \Gamma_{ij}^k v_i v_j = 0\,, \quad \text{for } \; i,j,k \in \{1,2,3\}, \tag{19}$$

where x_k and v_k denote the components of $\mathbf{x}$ and $\mathbf{v}$; and Γ_{ij}^k are the *Christoffel symbols*; second-order properties of the Earth's surface at each point that you might study in a differential geometry course. It is not necessary to know what they are for the purposes of our discussion. What is important is to know that given the initial conditions $\mathbf{x}(0) = A$ and $\dot{\mathbf{x}}(0) = \mathbf{v}(0) = \mathbf{v}_0$ this second-order differential equation *determines the rest of the geodesic*. Following a geodesic means setting out from village A with some initial velocity vector $\mathbf{v}_0$, and walking in such a way that our path is always locally straight, having zero geodesic curvature. For some choice of initial velocity vector $\mathbf{v}_0$, we will end up reaching village B at $t = 1$.

- This general approach of starting out from a given point along a given velocity and following a geodesic is called *geodesic shooting*. It allows us to reformulate the shortest path problem in terms of a much simpler unknown, $\mathbf{v}_0$:

$$\mathbf{v}_0^* = \arg\min_{\mathbf{v}_0 \in \mathbb{R}^3} |\mathbf{x}(1) - B|^2 \quad \text{subject to} \quad \mathbf{x}(0) = A,\; \dot{\mathbf{x}}(0) = \mathbf{v}_0 \text{ and (19).} \tag{20}$$

This problem is similar to rotating a tank to shoot a target, but with the difference that the projectiles move along geodesics. The advantage of this geodesic shooting formulation compared to the the previous formulation (17) is that it reduces the search space by a whole dimension. Another advantage is that even if we do not find the minimum of (20), our solutions $\mathbf{x}(t)$ obtained by solving the ODE (18) are always going to be geodesics (just ones that don't quite get us to village B). This is not the case for the previous formulation (17), where solutions short of the minimum are not guaranteed to be geodesics.

This idea of geodesic shooting extends beautifully to diffeomorphisms. Diffeomorphic flows ϕ_t that correspond to velocity fields $\mathbf{v}_t$ that minimize the LDDMM energy $E[\mathbf{v}; I, J]$ are indeed geodesics on the space Diff, with their length given by

$$L[\phi_1]^2 = \|\mathbf{v}\|_{\mathcal{V}}^2\,.$$

Consequently, these geodesics satisfy the "locally straight" differential equation equivalent to (19). This equation is called the Euler-Poincaré Differential Equation (EPDiff) and has the following form:

$$\frac{\mathrm{d}\mathbf{m}}{\mathrm{d}t} + (D\mathbf{m})\mathbf{v} + (D\mathbf{v})^T\mathbf{m} + \mathbf{m}\,(\nabla \cdot \mathbf{v}) = \mathbf{0}\,, \tag{21}$$

where D denotes the Jacobian matrix, $\nabla \cdot \mathbf{v}$ denotes the divergence operator, and $\mathbf{m}$ is called the *momentum*, given by

$$\mathbf{m} = \mathcal{L}^{\dagger}\mathcal{L}\mathbf{v} \quad \text{(equivalently, } \mathbf{v} = \mathcal{K}\mathbf{m}\text{)}$$

The momentum is a time-varying vector field just like $\mathbf{v}$, with $\mathbf{v}$ essentially being a smoothed version of $\mathbf{m}$ due to the form of the operator $\mathcal{L}$. This momentum has some analogy with the concept of momentum in particle physics.

Like the ODE (19), the EPDiff equation is a differential equation involving $\mathbf{v}$. Consequently, the solutions of (19) are determined by the initial conditions, i.e., the initial velocity field $\mathbf{v}(0)$, or equivalently, the initial momentum field $\mathbf{m}(0)$ (since velocity and momentum are related to each other by the operator $\mathcal{L}$). Therefore, we can reformulate the problem of diffeomorphic image registration as the problem of geodesic shooting: finding the initial velocity $\mathbf{v}_0$ (or initial momentum $\mathbf{m}_0$) for which the geodesic ϕ_t optimally matches images I and J at time 1. In complete form, this problem is formulated as follows [17, 29]:

$$\mathbf{m}_0^* = \arg\min_{\mathbf{m}_0} \int\limits_{\Omega} \left| I \circ \phi_1^{-1} - J \right|^2 , \quad \text{where}$$

$$\begin{cases} \mathbf{v}_t = \mathcal{K}\mathbf{m}_t \\ \frac{\mathrm{d}\mathbf{m}_t}{\mathrm{d}t} + (D\mathbf{m}_t)\mathbf{v}_t + (D\mathbf{v}_t)^T\mathbf{m}_t + \mathbf{m}_t\,(\nabla \cdot \mathbf{v}_t) = \mathbf{0} \\ \frac{\mathrm{d}\phi_t}{\mathrm{d}t} = \mathbf{v}_t \circ \phi_t \end{cases} \tag{22}$$

Solving this problem via gradient descent is computationally feasible. It involves pulling back the gradient of the image similarity metric with respect to ϕ_1 back through the ODEs for ϕ_t and $\mathbf{m}_t$. While the math for this is rather involved, the approach can be efficiently implemented computationally. As in the simple example introduced above, geodesic shooting offers the following advantages over the variational formulation of LDDMM:

- The optimization is over a single vector field $\mathbf{m}_0$, as opposed to the time-varying velocity field $\mathbf{v}$. This reduces the memory requirements of the algorithm and can lead to faster convergence. **Add a sentence on Miaomiao's Fourier domain paper.**

- Even if the minimum of the objective is not reached, solutions that are geodesics in Diff are obtained.

5.3 LDDMM and Computational Anatomy.

5.3.1 Defining Distance on the Space of Diffeomorphic Transformations

In addition to being a formidable image registration method LDDMM gives rise to a family of medical image statistical analysis approaches known as *computational anatomy* [16, 17, 29]. The Sobolev norm $\|\mathbf{v}\|_{\mathcal{V}}^2$ can be used to extend the concept of "distance" to the space of diffeomorphic transformations Diff, and even to the space of medical images. Let $\psi \in \mathrm{Diff}$ be a diffeomorphic transformation on Ω. Then let us define the distance between the identity transformation (i.e., the origin of the space Diff) and ψ as follows:

$$d(\mathrm{Id}, \psi)^2 = \min_{\mathbf{v} \in \mathcal{V} : \phi_1[\mathbf{v}] = \psi} \|\mathbf{v}\|^2_{\mathcal{V}} \,,$$

where $\phi_1[\mathbf{v}]$ denotes the solution of the flow ODE (5) with input $\mathbf{v}$ at time 1. In other words, $d(\mathrm{Id}, \psi)$ is equal to the kinetic energy of the least energetic diffeomorphic flow that has as its end-point the diffeomorphism ψ. For a pair of diffeomorphic transformations ψ, ξ, the distance is defined as

$$d(\xi, \psi) = d(\mathrm{Id}, \psi \circ \xi^{-1}) \,.$$

It quantifies the kinetic energy of the least energetic diffeomorphic flow that maps the end-point of ξ to the end-point of ψ. Although this distance function is not symmetric, it satisfies the properties of a proper distance function on the space Diff: $d(\xi, \psi) \geq 0$ for all $\psi, \xi \in \mathrm{Diff}$; $d(\xi, \psi) = 0$ if and only if $\psi = \xi$; and the triangle inequality $d(\xi, \psi) \leq d(\xi, \eta) + d(\eta, \psi)$ is satisfied for all $\psi, \xi, \eta \in \mathrm{Diff}$ [16].

5.3.2 Defining Distance on the Space of Images

While being able to measure distance between elements of Diff is useful, it would be even more useful to measure distance between images. At first, let us make a rather strong assumption that in our collection of images, any two images can be matched perfectly using some diffeomorphic transformation. This assumption makes sense if all images in the collection are deformations of a template image. It also makes sense for simple images, for example, images of fried eggs. In this setting, we can define the distance between images I and J as follows:

$$d(I, J)^2 = \min_{\mathbf{v} \in \mathcal{V} : I \circ \phi_1^{-1}[\mathbf{v}] = J} \|\mathbf{v}\|^2_{\mathcal{V}} \,,$$

in other words, the kinetic energy of the least energetic diffeomorphic flow that warps I to look like J at time 1. This distance function on images also behaves like a proper distance function (non-negative, zero only if $I = J$, satisfies triangle inequality).

In real-world applications, the assumption that J can be generated from I by a diffeomorphic transformation is too strong, as images have different intensity characteristics, noise, and often the underlying anatomical content is not one-to-one. The concept of distance between images must be relaxed to allow for inexact matching of I and J. A practical approach is to define the inexact distance between I and J as

$$d_{\mathrm{inexact}}(I, J)^2 = \min_{\mathbf{v} \in \mathcal{V}} E[\mathbf{v}; I, J] \,,$$

where $E[\mathbf{v}; I, J]$ is the LDDMM energy from (15), i.e., the sum of the kinetic energy of the diffeomorphic flow that best matches I and J and a residual intensity difference term. While this definition of distance no longer has the rigorous mathematical properties of the exact distance $d(I, J)$, it allows for many practical algorithms where distance between real-world images needs to be computed.[5]

5.3.3 Applications to Statistical Analysis of Images

Medical images are data, and we often want to draw inferences from medical images using statistical analysis. Often we derive some summary features from images (e.g., volume of the hippocampus) and perform statistical analysis on these derived measures (What is the average hippocampus volume in a given group? Does the mean hippocampal volume differ between patients and healthy individuals? Does the hippocampal volume change with age?). But summary features do not capture the entirety of information in the image. It would be nice to be able to perform the same kinds of statistical analyses on the images themselves, treating each image not as a source of one or more summary measures, but as a highly-dimensional multi-variate observation.

Of course, images are collections of voxels, and we can treat the vector of all intensities in an image as a multi-variate observation. But because voxels with the same coordinates across different images come from different anatomical locations, any such analysis will be almost meaningless. For example, averaging a collection of images, voxel-by-voxel, results in a blurry image that looks almost nothing like the source images and tells us nothing about the average anatomy in the collection.

By defining a distance function on the space of images that captures anatomical differences, rather than intensity differences, the LDDMM approach makes it possible to extend various concepts from statistics to the image domain. Many methods in statistics (finding the mean, linear and non-linear regression, dimensionality reduction) are formulated as least square fitting problems. For example, the problem of finding the mean of a sample of multi-variate observations $\mathbf{x}_1, \ldots, \mathbf{x}_n$ can be formulated in terms of minimizing the sum of squared distances:

$$\overline{\mathbf{x}}_{\mathrm{Fr}} = \operatorname*{arg\,min}_{\mathbf{x} \in \mathcal{X}} \sum_{i=1}^{n} d(\mathbf{x}, \mathbf{x}_i)^2 \,. \tag{23}$$

This definition of the mean is called the *Fréchet mean,* and unlike the arithmetic mean given by $\frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i$, it extends to non-Euclidean spaces. As long as the space $\mathcal{X}$ in which the observations $\mathbf{x}_1, \ldots, \mathbf{x}_n$ lie has a distance function, the Fréchet mean is well-defined. It is easy to verify that when $\mathcal{X}$ is a Euclidean space, the Fréchet mean is equal to the arithmetic mean. The Fréchet mean of

[5] The theory of *image metamorphosis* [10, 19] models transformations between images as pairs consisting of a diffeomorphic geometric component and an intensity variation component; and allows for a mathematically robust definition of distance between images that cannot be matched exactly.

a set of images $I_1 \ldots, I_n$ defined on the domain Ω can thus be constructed by solving the following minimization problem:

$$\overline{I}_{\mathrm{Fr}} = \underset{I \in \mathcal{I}_\Omega}{\arg\min} \sum_{i=1}^{n} d_{\mathrm{inexact}}(I, I_i)^2 = \underset{I \in \mathcal{I}_\Omega, \mathbf{v}_1 \in \mathcal{V}, \ldots, \mathbf{v}_n \in \mathcal{V}}{\arg\min} \sum_{i=1}^{n} E[\mathbf{v}_i; I, I_i].$$

This involves simultaneously searching for a "synthetic" mean image I and a set of velocity fields $\mathbf{v}_1, \ldots, \mathbf{v}_n$ that optimally warp I to match the input images. An efficient way to solve this minimization, proposed by Joshi et al., [13], is to alternate minimizing over I and $\mathbf{v}_1, \ldots, \mathbf{v}_n$. The resulting algorithm alternates between n independent LDDMM registrations and arithmetic averaging of image intensities, and yields excellent mean images (Figure XXX). These images are commonly used to create *population-specific templates* for groupwise analysis of medical imaging data [XXX].

Many other problems in statistics have been extended to the space of images. For example, the equivalent of linear regression (i.e., fitting a slope and an intercept of a line that best describes the relationship between a dependent variable, like hippocampal volume, and an independent variable, like age) is *geodesic regression* [8]. In geodesic regression with a scalar independent variable, we have a set of observations $(t_i \in [0,1], I_i \in \mathcal{I}_\Omega)$ and we one seeks to find a time-parameterized sequence of images $I_t = I_0 \circ \phi_t$ that forms a geodesic in the space of images, such that the sum of squared distances from the observed images to I_t is minimized. Specifically, one solves a minimization problem

$$(I_0^*, \mathbf{v}^*) = \underset{I_0 \in \mathcal{I}_\Omega, \mathbf{v} \in \mathcal{V}}{\arg\min} \sum_{i=1}^{n} d(I_0 \circ \phi_{t_i}^{\mathbf{v}}, I_i)^2$$

that is conceptually similar to the Fréchet mean problem in that it requires nested minimization. The concept of non-linear regression (i.e., fitting a quadratic or cubic curve to data in a least squares sense) has also been extended to the manifold of images [22]. So have been the concepts of mixed modeling for longitudinal image analysis [XXX], principal components analysis for data dimensionality reduction [XXX] and numerous other data analysis techniques that can be framed in terms of distance minimization.

6 Emerging Methods in Image Registration

References

[1] Vladimir I Arnold. Sur la géométrie différentielle des groupes de lie de dimension infinie et ses applicationsa l'hydrodynamique des fluides parfaits. *Ann. Inst. Fourier*, 16(1):319–361, 1966.

[2] Vladimir Igorevich Arnold. Ordinary differential equations, translated and edited by richard a. silverman, 1998.

[3] Vincent Arsigny, Olivier Commowick, Xavier Pennec, and Nicholas Ayache. A log-euclidean framework for statistics on diffeomorphisms. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 924–931. Springer, 2006.

[4] 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. *Med Image Anal*, Jun 2007. In press.

[5] 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. *Med Image Anal*, 12(1):26–41, Feb 2008.

[6] M. Faisal Beg, Michael I. Miller, Alain Trouvé, and Laurent Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *Int. J. Comput. Vision*, 61(2):139–157, 2005.

[7] F. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:567–585, 1989.

[8] P Thomas Fletcher. Geodesic regression and the theory of least squares on riemannian manifolds. *International journal of computer vision*, 105(2):171–185, 2013.

[9] Derek LG Hill, Colin Studholme, and David John Hawkes. Voxel similarity measures for automated image registration. In *Visualization in Biomedical Computing 1994*, volume 2359, pages 205–216. International Society for Optics and Photonics, 1994.

[10] Darryl Holm, Alain Trouvé, and Laurent Younes. The euler-poincaré theory of metamorphosis. *Quarterly of Applied Mathematics*, 67(4):661–685, 2009.

[11] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[12] Xue Hua, Suh Lee, Igor Yanovsky, Alex D Leow, Yi-Yu Chou, April J Ho, Boris Gutman, Arthur W Toga, Clifford R Jack, Matt A Bernstein, Eric M Reiman, Danielle J Harvey, John Kornak, Norbert Schuff, Gene E Alexander, Michael W Weiner, Paul M Thompson, and Alzheimer's Disease Neuroimaging Initiative. Optimizing power to track brain degeneration in Alzheimer's disease and mild cognitive impairment with tensor-based morphometry: an ADNI study of 515 subjects. *Neuroimage*, 48(4):668–681, Dec 2009.

[13] S. Joshi, Brad Davis, Matthieu Jomier, and Guido Gerig. Unbiased diffeomorphic atlas construction for computational anatomy. *Neuroimage*, 23 Suppl 1:S151–S160, 2004.

[14] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multi-modality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, 16(2):187–198, Apr 1997.

[15] Tommaso Mansi, Xavier Pennec, Maxime Sermesant, Hervé Delingette, and Nicholas Ayache. ilogdemons: A demons-based registration algorithm for tracking incompressible elastic biological tissues. *International journal of computer vision*, 92(1):92–111, 2011.

[16] Michael I Miller, Alain Trouve, and Laurent Younes. On the metrics and euler-lagrange equations of computational anatomy. *Annu Rev Biomed Eng*, 4:375–405, 2002.

[17] Michael I Miller, Alain Trouvé, and Laurent Younes. Geodesic shooting for computational anatomy. *J Math Imaging Vis*, 24(2):209–228, Jan 2006.

[18] Marc Modat, Gerard R Ridgway, Zeike A Taylor, Manja Lehmann, Josephine Barnes, David J Hawkes, Nick C Fox, and Sébastien Ourselin. Fast free-form deformation using graphics processing units. *Comput Methods Programs Biomed*, 98(3):278–84, Jun 2010.

[19] Casey L Richardson and Laurent Younes. Metamorphosis of images in reproducing kernel hilbert spaces. *Advances in Computational Mathematics*, 42(3):573–603, 2016.

[20] Wulf Rossmann. *Lie groups: an introduction through linear groups*, volume 5. Oxford University Press on Demand, 2006.

[21] 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 Trans Med Imaging*, 18(8):712–721, Aug 1999.

[22] Nikhil Singh and Marc Niethammer. Splines for diffeomorphic image regression. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 121–129. Springer, 2014.

[23] Andrew Staniforth and Jean Côté. Semi-lagrangian integration schemes for atmospheric models—a review. *Monthly weather review*, 119(9):2206–2223, 1991.

[24] Colin Studholme, Derek LG Hill, and David J Hawkes. An overlap invariant entropy measure of 3d medical image alignment. *Pattern recognition*, 32(1):71–86, 1999.

[25] Jean-Philippe Thirion. Non-rigid matching using demons. In *CVPR '96: Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, page 245. IEEE Computer Society, 1996.

[26] Tom Vercauteren, Xavier Pennec, Aymeric Perchant, and Nicholas Ayache. Symmetric log-domain diffeomorphic registration: a demons-based approach. *Med Image Comput Comput Assist Interv*, 11(Pt 1):754–61, 2008.

[27] Tom Vercauteren, Xavier Pennec, Aymeric Perchant, and Nicholas Ayache. Diffeomorphic demons: efficient non-parametric image registration. *Neuroimage*, 45(1 Suppl):S61–72, Mar 2009.

[28] W. M. Wells, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis. Multimodal volume registration by maximization of mutual information. *Med Image Anal*, 1(1):35–51, Mar 1996.

[29] Laurent Younes, Felipe Arrate, and Michael I. Miller. Evolutions equations in computational anatomy. *NeuroImage*, 45(1, Supplement 1):S40 – S50, 2009. Mathematics in Brain Imaging.