Diffeomorphic Image Registration
-
An Overview

Steffen Czolbe

December 10, 2019

Contents

1 Diffeomorphisms

A diffeomorphism is a smooth, bijective and thus invertible function, whose inverse is also smooth. Due to the strong guarantees on the smoothness of diffeomorphic transformations, it is a natural choice for the modeling of shape variations of biological anatomy. When soft tissue deforms, connected sets remain connected, disjoint sets remain disjoint and smoothness of anatomical features such as curves and surfaces is preserved.

2 Diffeomorphic Deformation Models

Most image registration techniques used in medical settings are based on diffeomorphic deformation the image domain (Schmidt-Richberg 2014), and diffeomorphic registration has evolved to a de-facto standard in many clinical studies (CitationRequired 9999). While diffeomorphisms can model most deformations of medical anatomies well, their use is debatable in some settings, such as sliding motions along organs (CitationRequired 9999), pre- to post-surgery of cancer, or registration to a template (Sabuncu et al. 2009).

Three popular choices of diffeomorphic deformation models are:

1. Large deformation diffeomorphic metric mapping (LDDMM) (Faisal Beg et al. 2005), which model the diffeomorphic transformation as a time-dependent geodesic flow inspired by fluid dynamics, and proposes a method of optimization

2. Freeform deformations, which inherently are not limited to diffeomorphisms, but require explicit regularization terms to ensure diffeomorphic deformations.

3. Stationary velocity fields (SVFs) (Arsigny et al. 2006), which model the diffeomorphic transformation as a flow over a static (non time-dependent) velocity field. SVFs are less computationally expensive than LDDMM, but lack a metric on the space of diffeomorphisms, which is important for performing statistics such as PCA or regression (Pai et al. 2016).

We will now further look at LDDMM.

3 Large deformation diffeomorphic metric mapping (LDDMM)

The LDDMM algorithm, introduced in Faisal Beg et al. (2005), has been a centerpiece of establishing diffeomorphic registration. It expands the large deformation model based on the model of viscous fluids (Christensen, Rabbitt, and Miller 1996) by proposing a gradient-descent based optimization algorithm based on the derivation of the Euler-Lagrange equation for the variational minimization of vector fields.

3.1 Model of Flows

In this setting, the diffeomorphic transformation φ of the domain is generated as the endpoint $\varphi = \phi_1$ ot the flow starting at time 0 and ending at time 1. For the viscous fluid analogy, the position of a particle dropped into the flow at position $\mathbf{x}$ and time $t = 0$ is $\phi_t(\mathbf{x})$ at time t.

The flow is modeled by a series of time-dependent velocity vector field $v_t : \Omega \rightarrow \mathbb{R}^n, t \in [0, 1]$, with each vector field encoding the direction of the flow at time point t. The velocity vector field thus encodes the gradient of the flow with regards to time:

$$\dot{\phi}_t = v_t(\phi_t) \ . \tag{1}$$

This gives a path $\phi_t : \Omega \rightarrow \Omega, t \in [0, 1]$, starting with the identity mapping $\phi_0 = id$ at $t = 0$ and terminating at $t = 1$. The diffeomorphic transformation is given by

$$\varphi = \phi_1 = \phi_0 + \int_0^1 v_t(\phi_t) \, \mathrm{d}t \ . \tag{2}$$

This formulation of the flow ensures that the transformation is invertable, and both the flow and its inverse are continuously differentiable.

3.2 Regularization

The velocity vector field is regularized by norm $\| \cdot \|_V$, and define $\|f\|_V = \|Lf\|_2$, with differential operator L of the Cauchy-Navier type, $L = -\alpha \nabla^2 + \gamma I$. Identity I, $\nabla^2 = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \frac{\partial^2}{\partial x_3^2}$. The coefficient α enforces smoothness, with higher values enforcing a higher regularity. Coefficient γ is chosen to be positive so that the operator is non-singular (= has an inverse).

3.3 Energy Function

Under the model of the flow, the regularization by a norm on the velocity field, and a square error intensity matching loss, the energy of the system is given by

$$E(v) = \int_0^1 \|v_t\|_V \, \mathrm{d}t + \frac{1}{\sigma^2} \|I_0 \circ \phi_1^{-1} - I_1\|^2 \ . \tag{3}$$

The optimal transformation is then defined by the sequence of velocity fields minimizing the energy

$$\hat{v} = \underset{v : \dot{\phi}_t = v_t(\phi_t)}{\operatorname{argmin}} \ E(v) \ . \tag{4}$$

3.4 LDDMM Algorithm

We now present the discretized implementation of LDDMM, following (Faisal Beg et al. 2005). We simulate the flow at discrete time steps $t_j \in [0, T], j \in [0, N]$ with time-interval $\delta t = \frac{N}{T}$. An overview is given in Algorithm 1. Individual points (1) - (14) are numbered according to the paper, and further explained in this section.

Algorithm 1: LDDMM

Input: Images I_0, I_1, N discrete steps

Initialize $k = 0, v^k = id, \nabla_{v^k} E = \mathbf{0}$

repeat

(1) Calculate new estimate of velocity $v^{k+1} = v^k - \epsilon \nabla_{v^k} E$

(2) Every 10 iterations: Reparametrize the velocity field to be constant speed.

(3) For $j = N - 1$ to 0: calculate backward flow $\phi^{k+1}_{t_j,T}$.

(4) For $j = 0$ to $N - 1$: calculate forward flow $\phi^{k+1}_{t_j,0}$.

(5) For $j = N - 1$ to 0: push forward source image $J^0_{t_j} = I_0 \circ \phi^{k+1}_{t_j,0}$

(6) For $j = 0$ to $N - 1$: pull back target image $J^1_{t_j} = I_1 \circ \phi^{k+1}_{t_j,T}$

(7) For $j = 0$ to $N - 1$: calculate image gradient $\nabla J^0_{t_j}$.

(8) For $j = 0$ to $N - 1$: calculate the jacobian determinant $|\nabla \phi_{t_j}|$

(9) For $j = 0$ to $N - 1$: calculate the gradient $\nabla_{v^{k+1}} E$

(10) Calculate the norm of the gradient $\|\nabla_{v^{k+1}} E\|$. Stop if small.

(11) Calculate new Energy $E(v)$.

(12) $k = k + 1$. Stop if max iterations reched.

until *stopping condition (10) or (12) triggered*

(13) Denote final velocity field as $\hat{v}$.

(14) Calculate the length of the path on the manifold.

Result: Diffeomorphic transformation $\hat{v}$

(1) New estimate of velocity

Optimization of the diffeomorphic transformation via gradient descent. ϵ is the learning rate, $\nabla_{v^k} E$ the gradient of the energy function, as calculated in step (9).

(2) Reparameterization

Since the underlying flow is geodesic, the flow should have constant speed ($\|v_t\|_V = \text{const}$). This property can be numerically achieved by time reparameterization, and doing so speeds up convergence. We re-parameterize the velocity vector fields as

$$\tilde{v}_{t_j} = \frac{\text{length}}{T} \frac{v_{t_j}}{\|v_{t_j}\|_V} \tag{5}$$

with length $= \sum_{j=0}^{N-1} \|v_{t_j}\|_V$. To calculate the norm $\|f\|_V = \|Lf\|_2$, the Cauchy-Navier operator $L = -\alpha\nabla^2 + \gamma I$ is discretized via finite differences:

$$Lf(\mathbf{x}) = (-\alpha\nabla^2 + \gamma I) f(\mathbf{x}) \tag{6}$$

$$= -\alpha\left(\sum_{d=0}^{D} \frac{f(\mathbf{x} - \Delta_d) - 2f(\mathbf{x}) + f(\mathbf{x} + \Delta_d)}{\Delta_d^2}\right) + \gamma f(\mathbf{x}) \ , \tag{7}$$

where d iterates of the dimensions of the domain and Δ_d is a small step along the basis vector of dimension d.

Note: This implementation of the reparameterization is a slight simplification. The original paper additionally reparamterizes time-steps t_j.

(3) Calculation of backward flows

Next, we calculate the backward flows $\phi_{t_j,T}$. The flow $\phi_{t_j,T}(\mathbf{x})$ describes the position at time T of a particle that is at position $\mathbf{x}$ at time t_j. Thus, the forward flows can be used to obtain the position of any particle at any time slot t_j at the end of the time interval $[0, T]$, which will later be used to *backwards* morph the target image I_1 to an earlier point in time t_j.

The flow sequence is calculated iteratively for $j = N - 1$ to 0, starting with $\phi_{t_N,T} = \phi_{T,T} = id$ and then following the sequence with

$$\phi_{t_j,T}(\mathbf{x}) = \phi_{t_{j+1},T}(\mathbf{x} + \alpha) \ , \tag{8}$$

with α initialized as 0 and then iteratively estimated as

$$\alpha = \delta t \ v_{t_j}\left(\mathbf{x} + \frac{\alpha}{2}\right) \tag{9}$$

for a suggested 5 iterations. Both the flow and the velocity vector field are defined at Cartesian points within the domain (the pixel's coordinates). For non-Cartesian grid points, interpolation (bi- / tri-linear, or gaussian kernels) is used.

(4) Calculation of forward flows

Similar to the previous step, we now calculate the forward flows $\phi_{t_j,0}$. While the flow goes backwards in time, it will be used in the next step to push *forward* the image I_0.

The flow sequence is calculated iteratively for $j = 0$ to $N-1$, starting with $\phi_{t_0,0} = \phi_{0,0} = id$ and then following the sequence with

$$\phi_{t_{j+1},0}(\mathbf{x}) = \phi_{t_j,0}(\mathbf{x} - \alpha) \ , \tag{10}$$

with α calculated iteratively as

$$\alpha = \delta t \ v_{t_j}\left(\mathbf{x} - \frac{\alpha}{2}\right) \ . \tag{11}$$

Similar to the previous step, interpolation is necessary.

(5) Morph Image I_0

We forward-push image I_0 along the flow to time points t_j and obtain a series of images $J^0_{t_j}$. Images $J^0_{t_j}$ are formed by sampling I_0 at coordinates given by the forward flow $\phi^{k+1}_{t_j,T}(\mathbf{x})$. To sample the image at non-Cartesian coordinates, interpolation is necessary.

We create a series of images for $j = 0$ to $N-1$:

$$J^0_{t_j} = I_0 \circ \phi_{t_j,0} \tag{12}$$

(6) Morph Image I_1

Analog to the previous step, we pull-back image I_1 along the flow to time points t_j and obtain a series of images $J^1_{t_j}$. For $j = N-1$ to 0:

$$J^1_{t_j} = I_1 \circ \phi_{t_j,T} \tag{13}$$

(7) Calculate image gradient $\nabla J^0_{t_j}$

The gradient for all the push-forward images $J^0_{t_j}$ is calculated. Gradient computation of images can be performed via finite differences, Soebel/Perwitt operators or Gaussian derivatives.

(8) Calculate Jacobian determinant of the transformation $|\nabla\phi_{t_j,T}|$

The Jacobian determinant of all backward-flows is calculated. This is done by first calculating the gradient $\nabla\phi_{t_j,T}$, which is a 2×2 matrix in 2d, or a 3×3 matrix in 3d, and then calculating the determinant of it. The determinant of the Jacobian is used in the construction of the energy function derivative.

The determinant can further be used to verify the bijectivity of the transformation. Only functions with a positive Jacobian determinant in the local neighborhood of $\mathbf{x}$ posses an inverse function that is continuously differentiable. Since this property is a prerequisite for a diffeomorphism, we can use it to check that the numerical approximation is stable.

(9) Calculate the gradient $\nabla_{v^{k+1}}E$

The intermediate results of the previous steps are combined to calculate the gradient of the Energy function. For $j = 0$ to $N-1$:

$$(\nabla_{v^{k+1}}E_{t_j})_V = 2v_{t_j} - K\left(\frac{2}{\sigma^2}|\nabla\phi_{t_j,T}|\nabla J^0_{t_j}(J^0_{t_j} - J^1_{t_j})\right) \tag{14}$$

The gradient is calculated in the local space V of the manifold. Operator $K = (LL)^{-1}$ is defined to lift the second term out of this space. K can be efficiently implemented in the Fourier domain. Let $\Delta x_i = \frac{1}{N_i}$ be the normalized spacing of the Cartesian points (pixels) of domain Ω. We can then implement $f(\mathbf{x}) = K(g(\mathbf{x}))$ for the Fourier transforms F, G of f, g as

$$F(\mathbf{k}) = \frac{G(\mathbf{k})}{A^2(\mathbf{k})} \ , \tag{15}$$

where frequency $\mathbf{k} = (k_1, k_2, ..., k_d)$ for $\Omega \subset \mathbb{R}^d$ and

$$A(\mathbf{k}) = \gamma + 2\alpha\sum_{i=1}^{d}\frac{1-\cos(2\pi\Delta x_i k_i)}{\Delta x_i^2} \ . \tag{16}$$

(10) Calculate the norm of the gradient $\|\nabla_{v^{k+1}}E\|$

If the norm of the gradient is small, we are close to an extrema and can stop the gradient descent.

(11) Calculate new Energy $E(v)$

Use Equation 3 to calculate the remaining energy in the system. As this is the quantity we are minimizing, observing it's change lets us monitor the optimization process.

(12) $k = k + 1$. Stop if too large

Increment the loop counter. Abort if a predefined maximum iteration count is reached.

(13) Denote final velocity field as $\hat{v}$

The optimization result as an approximation of the true velocity field. Thus we denote it as $\hat{v}$.

(14) Calculate the length of the path on the manifold

The length of the path on the manifold between the identity transformation and the obtained flow is given by

$$\text{Length}(id, \phi_{0,T}) = \sum_{j=0}^{N-1} \|\hat{v}_{t_j}\|_V \delta t \ . \tag{17}$$

This is the length of the geodesic and hence the estimated metric between the given images.

References

Arsigny, Vincent, Olivier Commowick, Xavier Pennec, and Nicholas Ayache (2006). “A log-euclidean framework for statistics on diffeomorphisms”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer Verlag, pp. 924–931.

Christensen, Gary E., Richard D. Rabbitt, and Michael I. Miller (1996). “Deformable templates using large deformation kinematics”. In: *IEEE Transactions on Image Processing* 5.10, pp. 1435–1447.

CitationRequired (9999). “TODO”. In:

Faisal Beg, Mirza, Michael I Miller, Alain Trouvétrouv, and Laurent Younes (2005). *Computing Large Deformation Metric Mappings via Geodesic Flows of Diffeomorphisms*. Tech. rep. 2, pp. 139–157.

Pai, Akshay, Stefan Sommer, Lauge Sorensen, Sune Darkner, Jon Sporring, and Mads Nielsen (June 2016). “Kernel Bundle Diffeomorphic Image Registration Using Stationary Velocity Fields and Wendland Basis Functions”. In: *IEEE Transactions on Medical Imaging* 35.6, pp. 1369–1380.

Sabuncu, Mert R, B T Thomas Yeo, Koen Van Leemput, Tom Vercauteren, and Polina Golland (2009). “Asymmetric image-template registration.” In: *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention* 12.Pt 1, pp. 565–73.

Schmidt-Richberg, Alexander (2014). *Registration methods for pulmonary image analysis: integration of morphological and physiological knowledge.* Springer Science & Business Media.