Flexible Alignment of Images Using B-Spline Reparameterization and $L^2$-Gradient Flows

Guoliang Xu
State Key Laboratory of Scientific and Engineering Computing Institute of Computational Mathematics, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing 100190, China
Email:xuguo@lsec.cc.ac.cn

Yanmei Zheng
State Key Laboratory of Scientific and Engineering Computing Institute of Computational Mathematics, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing 100190, China
Email:zhengym@lsec.cc.ac.cn

Abstract—In this paper, we present a new flexible alignment method to align two or more similar images. By minimizing an energy functional measuring the difference of the initial image and target image, a $L^2$-gradient flow is derived. The flow is integrated by a finite element method in the spatial direction and a semi-implicit Euler scheme in the temporal direction. The experimental results show that the proposed method is efficient, effective and capable of capturing large variation of the initial and the target images.

I. INTRODUCTION

Image alignment (or registration) is a fundamental task in image processing to overlay two or more images used. It refers to the geometric alignment of a set of images. The set may consist of two or more digital images taken from a single scene at different times, from different sensors, or from different viewpoints, or cross sections of neuron. The goal of registrations is to establish geometric correspondence between the images so that they can be compared, interpolated for further study. Image registration has been used in many fields such as medical images, aerial, satellite, weather, and computer vision [4], [19]. Basically, registration methods can be loosely classified into the following categories:

Alignment techniques in the first category are based on the operations on image intensities, such as correlations [14], and Fourier methods [11], [12]. These two methods are often used together. Fourier based schemes, which are able to estimate large rotations, scalings and translations are often used because of its accurate estimation. Most of the DFT based approaches utilize the shift properly [10] of the Fourier transform, which enables a robust estimation of translations using normalized phase-correlation [9], [18]. To handle rotations and scaling, resampling the Fourier magnitudes on the log-polar grid reduces the problem of estimating the rotation and scaling to one of estimating a 2D translation. Thus, the method relies on dual correlations: once in the log-polar Fourier domain to estimate the rotation and scaling and once more in the spatial domain to recover the residual translation.

The methods in the second category are feature-based (see [3], [16]). In these methods, one first extract distinctive features from each image, to match these features to establish a global correspondence, and to then estimate the geometric transformation between the images. This kind of approaches has been used since the early days of stereo matching, and has more recently gained popularity for image stitching applications [6], [8].

In the third category, elastic model [2] is used which is implement after finding the global difference such as rotation angle, scale parameter, and translate component, to find the correspondence between two images with tiny difference. The author of [5] use B-splines to model the images. And given an elastic method to solving the biological image registration.

A general review of alignment was given in [4], [15]. In the recent applications of medical diagnosis and therapy treatment planning, many combined methods appeared (see [1], [7]).

In this paper, we assume that the global alignment has been conducted using the Fourier based methods. What we need to find is the correspondence $x(u, v)$ between the two similar images. Given an error metric to measure the similarity of the two images, we deduce the Euler-Lagrange operator and the geometric flow first, and then express the correspondence $x$ using B-spline base functions, after solving the derived systems consisting of nonlinear equations, we get the control points. Through an iterative process we can continuously change the initial image to the target image and can find their correspondence.

The rest of this paper is organized as follows. In Section II, some basic setting is given. We deduce the Euler-Lagrange function and $L^2$-gradient flows in Section III, and then give the numerical solving method in Section IV. After solving the systems of equations for obtaining the new control points, a resampling step is taken in Section V. Some experiments result are given in Section VI. We conclude this paper in Section VII.

II. PRELIMINARY

This section introduce necessary material used in this paper.

A. The B-spline

Set $k = 3$, $u_0 = u_1 = u_2 = 0, u_{i+3} = \frac{1}{m}, i = 0, 1, \cdots, m$, $u_{m+4} = u_{m+5} = u_{m+6} = 1$. Then the cubic spline basis functions defined on those nodes are [17]:

$$N_{0,3}(u) = (1 - mu)^3, \quad mu \in [0, 1),$$
\[ N_{1,3}(u) = \begin{cases} 3mu(1 - \frac{3}{2}mu + \frac{7}{12}(mu)^2), & mu \in [0,1), \\
\quad -\frac{1}{4}(mu - 2)^3, & mu \in [1, 2), \\
\quad \frac{3}{8}(mu)^2 - \frac{11}{12}(mu)^3, & mu \in [0, 1), \\
\quad \frac{1}{8}(mu - 2)^2 + \frac{1}{12}(mu - 2)^3, & mu \in [1, 2), \\
\quad \frac{1}{12}(mu - 3)^3, & mu \in [2, 3), \end{cases} \]

\[ N_{2,3}(u) = \begin{cases} \frac{3}{8}(mu)^2 - \frac{11}{12}(mu)^3, & mu \in [0, 1), \\
\quad \frac{1}{8}(mu - 2)^2 + \frac{1}{12}(mu - 2)^3, & mu \in [1, 2), \\
\quad \frac{1}{12}(mu - 3)^3, & mu \in [2, 3), \end{cases} \]

\[ N_{i,3}(u) = \beta^3(mu - i + 1), mu \in [i - 3, i + 1), \quad i = 3, \ldots, m - 1, \]

\[ N_{m,3}(u) = N_{2,3}(1 - u), mu \in [m - 3, m), \]

\[ N_{m+1,3}(u) = N_{1,3}(1 - u), mu \in [m - 2, m), \]

\[ N_{m+2,3}(u) = N_{0,3}(1 - u), mu \in [m - 1, m), \]

and

\[ \beta^3(u) = \begin{cases} \frac{2}{9} - \frac{1}{3}|x|^3, & 0 \leq |x| < 1, \\
\quad \frac{1}{3}(2 - |x|)^3, & 1 \leq |x| < 2. \end{cases} \]

### B. Initial control points

The initial control points we used are (see [17]):

\[ \alpha_i = \begin{cases} \frac{1}{m}, & i = 1, \\
\quad \frac{1}{m}, & i = 2, \ldots, m, \\
\quad 1 - \frac{1}{m}, & i = m + 1. \end{cases} \]

\[ \beta_j = \begin{cases} \frac{1}{m}, & j = 1, \\
\quad \frac{1}{m}, & j = 2, \ldots, n, \\
\quad 1 - \frac{1}{m}, & j = n + 1. \end{cases} \]

These equation can be obtained from:

\[ x(u, v) = \sum_{i=1}^{m+1} \sum_{j=1}^{n+1} (\alpha_i, \beta_j)N_{i,3}(u)N_{j,3}(v) = (u,v), \quad \forall u \in [0, 1], v \in [0, 1]. \]

For easy of description, we reorder the control points \( p_{ij} \) and basic functions of the B-spline map into a 1-dimensional array:

\[ \phi(i-1)(m+1)+j-1(u,v) = N_{i,3}(u)N_{j,3}(v), \quad i = 1, \ldots, m + 1, j = 1, \ldots, n + 1. \]

After we reorder the inner control points, reorder the boundary control points from \((i, j) = (0, 0)\) in a clockwise direction. Then the control points can be represented as:

\[ x_0, \ldots, x_{n_0}, x_{n_0+1}, \ldots, x_{n_1}, \]

where \( x_{i_0}, \ldots, x_{n_0} \) are inner control points, \( x_{n_0+1}, \ldots, x_{n_1} \) are boundary control points, and

\[ n_0 = \prod_{i=1}^{m+1}(m_i + 1) - 1, \quad n_1 = \prod_{i=1}^{m+1}(m_i + 3) - 1. \]

The B-spline basis functions are correspondingly reordered and represented as

\[ \phi_0, \ldots, \phi_{n_0}, \phi_{n_0+1}, \ldots, \phi_{n_1}. \]

### III. ERROR METRIC AND B-SPLINE REPARAMETERIZATION BY L^2-GRADIENT FLOWS

#### A. Problem Description

Given two same size images \( I_0(u, v) \) (reference image) and \( I_1(u, v) \) (test image) in \( \mathbb{R}^2 \) with similarities defined on \([0, 1]^2\).

Find a smooth mapping

\[ x(u, v) : [0, 1]^2 \rightarrow [0, 1]^2, \]

such that

\[ \int_{\Omega} \|I_1(x(u, v)) - I_0(u, v)\|^2 dudv = \min, \]

where \( \Omega = [0, 1]^2 \). In this paper, we choose \( x(u, v) \) as a bivariate bi-cubic B-spline defined on \( \Omega \).

#### B. Assumption

Images are traditionally defined only on the integer grid. Here we assume \( I_0(u, v) \) and \( I_1(u, v) \) are defined as two continuous functions by assuming a bilinear interpolation is enforced in each of the pixels using the density values at the grid points.

#### C. Euler-Lagrange function

Let

\[ E(x) = \int_{\Omega} \| I_1(x(u, v)) - I_0(u, v) \|^2 dudv. \]

Now we construct an \( L^2 \)-gradient flow to minimize the energy functional \( E(x) \). Let

\[ x(u, \epsilon) = x + \epsilon \Phi(u) : u \in [0, 1]^2, \quad \Phi \in C_0^3([0, 1]^2). \]

Then we have

\[ \delta(E(x), \Phi) = \frac{d}{d\epsilon}E(x(\cdot, \epsilon))\bigg|_{\epsilon=0}, \]

where

\[ \delta(E(x), \Phi) = 2\int_{\Omega} \left[ \| I_1(x(u, v)) - I_0(u, v) \| (\nabla_x I_1)^T \delta(x) \right] dudv. \]

It follows from

\[ x = x + \epsilon \Phi, \quad \delta(x) = \Phi. \]

Hence

\[ \delta(E(x), \Phi) = 2\int_{\Omega} \left[ \| I_1(x(u, v)) - I_0(u, v) \| (\nabla_x I_1)^T \Phi \right] dudv. \]

To construct \( L^2 \)-gradient flows moving the \( x \) in the tangent \( D_t x \) directions, \( l = 1, 2 \), we take

\[ \Phi = (D_t x)(D_t x)^T \phi, \quad \phi \in C_0^3([0, 1]^2). \]

Therefore, we construct the following weak form \( L^2 \)-gradient flows moving \( x \) in the \( D_t x \) direction

\[ \int_{\Omega} \frac{\partial x}{\partial t} \phi \ dV = -2 \int_{\Omega} \left[ \| I_1(x(u, v)) - I_0(u, v) \| (D_t x)^T (\nabla_x I_1) (D_t x) \phi \right] dudv, \]

\[ l = 1, 2. \]
D. Remark

Let us explain the reason we take $\Phi$ as (5). Taking $\Phi = D_t x \phi$, we obtain the Euler-Lagrange operator for (4) as

$$2\|I_1(x(u, v)) - I_0(u, v)\|[(\nabla x I_1)^T D_t x].$$

Hence the $L^2$-gradient flow, for (3), moving $x$ in the direction $D_t x$ is

$$\frac{\partial x}{\partial t} = -2\|I_1(x(u, v)) - I_0(u, v)\|[(\nabla x I_1)^T D_t x].$$

The weak-form of this equation is (6). This is the same as taking $\Phi = (D_t x) / (D_t x)^T \varphi$ in (4).

IV. NUMERICAL SOLUTIONS AND ALGORITHM

A. Numerical Solutions

To minimize $E(x)$, we solve (6) interchangeably using finite element method in the spacial discretization and semi-implicit Euler scheme in temporal discretization. More specifically, the term

$$\|I_1(x(u, v)) - I_0(u, v)\|[(\nabla x I_1)^T (\nabla x I_1)]$$

in (6) is treated as a known quantity which is computed from the previous step of the approximation of $x(u, v)$. The term $D_t x$ is treated as an unknown.

Using the ordering of the basis functions and control points in (1), (2), $x$ can be represented as

$$x(u, v) = \sum_{j=0}^{n_0} x_{ij} \phi_j(u, v) + \sum_{j=n_0+1}^{n_1} x_{ij} \phi_j(u, v).$$

(8)

Substituting $x(u, v)$ into (6), and taking the test function $\phi$ as $\phi_i$, for $i = 0, \cdots, n_0$, we can discretize (6) as a set of linear systems of ordinary differential equations (ODE) with the inner control points $x_i$, $i = 0, \cdots, n_0$, as unknowns.

$$\sum_{j=0}^{n_0} m_{ij} \frac{dx_{ij}}{dt} + \sum_{j=0}^{n_0} q_{ij} x_j = \sum_{j=n_0+1}^{n_1} q_{ij} x_j, i = 0, \cdots, n_0, l = 1, 2,$$

(9)

where

$$m_{ij} = \int \phi_i \phi_j \ dV,$$

$$q_{ij} = 2\int [(I_1(x(u, v)) - I_0(u, v))[(D_t x)(\nabla x I_1)](D_t \phi_j)\phi_i] \ dV.$$

For the temporal direction discretization of these ODE systems, we use a forward Euler scheme which finally leads to two linear systems of algebraic equations.

$$\sum_{j=0}^{n_0} \left(m_{ij} + \tau q_{ij}^{(k,1)}\right) x_j^{(k,1)} = \sum_{j=0}^{n_0} m_{ij} x_j^{(k-1,2)} + \sum_{j=n_0+1}^{n_1} q_{ij}^{(k,1)} x_j^{(k,1)},$$

$$\sum_{j=0}^{n_0} \left(m_{ij}^{(1)} + \tau q_{ij}^{(k,2)}\right) x_j^{(k,2)} = \sum_{j=0}^{n_0} m_{ij}^{(1)} x_j^{(k,1)} + \sum_{j=n_0+1}^{n_1} q_{ij}^{(k,2)} x_j^{(k,1)},$$

(10)

where $\tau$ is a temporal step-size, $K$ is the number of total steps.

For $l = 1, 2$, $m_{ij}^{(l)}$ means $m_{ij}$ in $D_t x$ direction, $q_{ij}^{(l)}$ means $q_{ij}$ of step $k$ in $D_t x$ direction, $x_{ij}^{(k,1)}$ means $x_{ij}$ of step $k$ in $D_t x$ direction.

Solving these linear systems via GMRES (see [13]) for $k = 1, \cdots, K$, we obtained the new inner control points of $x$.

B. Algorithm Details

In the computation, we need to decide $\tau$ and the number $N$ of cubic B-spline basic functions we used, just by do experiments to find the best $\tau$ and $N$, while different images need different $N$, just according to the complexity of the images. $\tau = 10^{-4}, N = 24$ is used in this paper.

Three conditions are used in the program: first, about the control points:

$$p_{ij}^{(k,l)} \geq p_{ij}^{(k,l)} - \delta, \quad p_{ij}^{(k,l)} \geq p_{ij}^{(k,l)} - \delta, k = 0, \cdots, K, l = 1, 2.$$ (12)

second, about the vertex $x$: 

$$x_{ij}^{(k,l)} \geq x_{ij}^{(k,l)}, \quad x_{ij}^{(k,l)} \geq x_{ij}^{(k,l)}, k = 0, \cdots, K, l = 1, 2.$$ (13)

third, we define the error between the two images as:

$$\text{Error} = \sqrt{\sum_{i=0}^{w} \sum_{j=0}^{h} [(I_1^{(0,0)}(u_i, v_j) - I_0(u_i, v_j))^2]},$$

$$\text{error}^{(k,l)} = \sqrt{\sum_{i=0}^{w} \sum_{j=0}^{h} (I_1^{(k,l)}(u_i, v_j) - I_0(u_i, v_j))^2},$$

where $w, h$ is the width and height of the two images, if

$$\text{error}^{(k,l)} < \delta,$$

(14)

We consider that the two images have been matched, we should choose different $\delta$ for different images. For simple images here we choose $\delta = 10^{-2}$. We use the following iterative algorithm:

1) Read two images $I_1, I_0$, set $k = 0$, initial B-spline representation of $x^{(0,1)}(u, v)$ such that $x^{(0,1)}(u, v) = (u, v)$.

2) Solve (10), (11) to obtain the new $p_{ij}^{(k,l)}$, and then get the new $x_{ij}^{(k,l)}$.

3) Check the conditions (12), (13), if it is not satisfied, go to step 5. Otherwise, go to step 2.

4) Check whether step $k$ is greater than 5000, if satisfied, give $N$ an increment, then go to step 5. Otherwise go to step 2.

5) Sampling the image $I_1{(x^{(k,l)}(u, v))}$ at the grid points $[u_i, v_j]$, and initial $p_{ij}^{(k,l)} = (\alpha_i, \beta_j)$, then check the conditions (14), if satisfied stop. Otherwise go to step 2.

6) After iterations of step 2, 3, 4, 5, we can find the result $p_{ij}^{(k,l)}$ and $x_{ij}^{(k,l)}$.

V. REGISTRATION

After the deformation function $x(u, v)$ is finally determined, $I_1(x(u, v))$ is the aligned image of $I_0(u, v)$. To obtain a discrete version of $I_1(x(u, v))$ at the grid points $(u_i, v_j)$ with

$$u_i = \frac{i}{M}, v_j = \frac{j}{M}, \quad i = 0, \cdots, M, \quad j = 0, \cdots, N.$$
we first compute $x_{ij} = x(u_i, v_j) \in [0, 1]^2$, then find $k$ and $l$, such that

$$x_{ij} \in [k, k+1] \times [l, l+1].$$

Then the discrete version of $I_1(x(u, v))$ at the grid points $(u_i, v_j)$ is computed as the bilinear interpolation of $I_1(x(u, v))$ over the rectangle $[k, k+1] \times [l, l+1]$ at the point $x_{ij}$.

VI. Examples and Applications

We give several experiments here to illustrate the property of our method. We find our method can align so accurate, which can make the error in (14) be zero in these experiments.

For the seven experiments we want to test the ability of our method, so different images are given here, they are different on size or topology or shape. We use two $128 \times 128$ images as the initial image and target image. For every experiment, just select sixteen or twenty images to demonstrate the evolution, and give the step number below the image. Here for easy of description, we use Rec represents rectangle, and Cir represents circle, for example $12$Rec means the image consists of twelve rectangles, $8$Cir means the image consists of eight circle.

We do double direction tests, and succeed except that the two images are so different, one direction will failed. For experiments of figure 1-4, they were aligned perfectly with zero error. For the three experiments in figure 5-7, we can succeed in changing four circles to twelve rectangles, but failed in the opposite direction. The left three experiments are similar.

From the experiments, we can find that:

1) Experiments in both directions are sometimes not reversible.
2) Different images take different time.
3) Only little time need in transforming the shape from the initial image to the target image, most of the time was taken on the boundary.

VII. Conclusions and Future Works

A new algorithm for elastic alignment has been presented. It combines the ideas of flexible alignment based on B-spline
reparametrization and $L^2$-gradient flow. The method can align simple images perfectly.

And our aim is handling with large images which are the cross sections of the neuron, what we need to find is the correspondence between the two adjacent images, so that we can construct more images in-between, further more, we can reconstruct the 3D structure more accurately, more efficiently. So what we have done just was the preliminary work.

ACKNOWLEDGMENT

The authors would like to thank Project support in part by NSFC grant 60773165, NSFC key project under the grant 10990013.

REFERENCES


