ESAIM: M2AN 57 (2023) 299–328
https://doi.org/10.1051/m2an/2022080

ESAIM: Mathematical Modelling and Numerical Analysis
www.esaim-m2an.org

3D DIFFEOMORPHIC IMAGE REGISTRATION WITH CAUCHY–RIEMANN CONSTRAINT AND LOWER BOUNDED DEFORMATION DIVERGENCE

Huan Han[1] and Zhengping Wang[2,*]

Abstract. In order to eliminate mesh folding in 3D image registration problem, we propose a 3D diffeomorphic image registration model with Cauchy–Riemann constraint and lower bounded deformation divergence. This model preserves the local shape and ensures no mesh folding. The existence of solution for the proposed model is proved. Furthermore, an alternating directional projection 3D image registration algorithm is presented to solve the proposed model. Moreover, numerical tests show that the proposed algorithm is competitive compared with the other four algorithms.

Mathematics Subject Classification. 65F10, 65M55, 68U10, 65M12, 94A08.

Received June 17, 2021. Accepted September 26, 2022.

1. Introduction

Image registration is spatial position calibration of two given data sets. It is an important tool for medical image analysis, for example, atlas-based image segmentation and shape analysis. Moreover, image registration is also widely used in remote sensing, astronomy and some other fields in industry. An overview of image registration problem can be found in [1–3]. Here we focus on mono-modality 3D nonrigid deformable image registration problem in variational framework. Mathematically, 3D image registration problem can be stated in the following way. Let $\Omega \subset \mathbb{R}^3$ be an open bounded domain on $\mathbb{R}^3$, and let $T, D : \Omega \to \mathbb{R}$ be two real functions. T and D are viewed as two 3D images, for example, T and D are tomography data. In image registration, T and D are named the floating image and the target image, respectively. The registration of T and D is to find a 3D spatial transformation $\mathbf{h} : \Omega \to \Omega$ such that the intensity of $T \circ \mathbf{h}(\cdot)$ looks like $D(\cdot)$ as much as possible. For example, finding an $\mathbf{h}$ such that the sum of squared difference (SSD) $\int_\Omega |T \circ \mathbf{h}(\mathbf{x}) - D(\mathbf{x})|^2 \, \mathrm{d}\mathbf{x}$ reaches its minimum. Addressing this issue, many variational models [4–7,58–60] are proposed. However, mesh folding is not taken into consideration in most of these models. In physical view, mesh folding implies the volumetric vanishing which contradicts physical principle. As a consequence, eliminating mesh folding is an essential technique for 3D image registration. To eliminate mesh folding for 3D image registration, deformation $\mathbf{h} : \Omega \to \Omega$ should be a local inverse function. In diffeomorphic image registration framework, one should ensure that the Jacobian

Keywords and phrases. 3D diffeomorphic image registration, mesh folding, Cauchy–Riemann constraint, lower bounded divergence, alternating direction, projection algorithm.

[1] Department of Mathematics, Wuhan University of Technology, Wuhan 430070, P.R. China.

[2] Center for Mathematical Sciences and Department of Mathematics, Wuhan University of Technology, Wuhan 430070, P.R. China.

*Corresponding author: zpwang@whut.edu.cn

determinant of $\mathbf{h}$ is not equal to 0 (see Appendix C in [8] for details). Moreover, Jacobian determinant of $\mathbf{h}$ denotes the volume stretching rate for the transformation. To make deformation physically meaningful and preserve the orientation, one should ensure that Jacobian determinant of $\mathbf{h}$ is greater than 0 for any $\mathbf{x} \in \Omega$. For the purpose of eliminating mesh folding, Beg *et al.* [9–17] developed a large deformation diffeomorphic metric mapping (LDDMM) framework. In the LDDMM framework, image registration is formulated as a variational problem by viewing the solution as a geodesic flow in space of diffeomorphism. In the LDDMM framework, Dupuis [12] proved that no mesh folding occurred by restricting velocity field to the Sobolev space $W_0^{3,2}(\Omega)$. Moreover, several numerical tests also showed the efficiency of LDDMM. One can refer to [14–16] for details. In addition, by giving the hypothesis that $\mathbf{h}(\mathbf{x})$ is approximately divided into an identity part $\mathbf{x}$ and a displacement part $\mathbf{u}(\mathbf{x})$ (*i.e.*, $\mathbf{h}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$), the 3D image registration problem can be also formulated by the following framework:

$$\bar{\mathbf{u}} = \arg\min_{\mathbf{u}\in\mathcal{K}} S(\mathbf{u}) + \overline{R}(\mathbf{u}), \tag{1}$$

where $\overline{R}(\mathbf{u})$ is regularization (see [4–7]) and $\mathcal{K}$ is an admissible set of solutions. Here and in what follows,

$$S(\mathbf{u}) = \int_\Omega [T(\mathbf{x} + \mathbf{u}(\mathbf{x})) - D(\mathbf{x})]^2 \,\mathrm{d}\mathbf{x}. \tag{2}$$

Many canonical registration models arise from framework (1), for example, Demons model [18], Diffeo Demons model [19], Vectorical total variation (VTV) model [20,21] and Bounded deformation (BD) model [6]. However, mesh folding is not taken into consideration in framework (1). Therefore, searching for a framework (1) based image registration model without mesh folding is a challenging task. To solve this problem, a 3D orientation-preserving variational model is proposed by Zhang and Chen [22] based on framework (1). There are also some other relevant works on 2D quasi-conformal or conformal mapping (see [23–32] for details). Particularly, in [31, 33], authors of this article proposed a 2D diffeomorphic image registration model by constraining the solution into a 2D conformal set. As an extension, to eliminate mesh folding in 3D image registration, we propose a 3D diffeomorphic image registration model with Cauchy–Riemann constraint and lower bounded deformation divergence (see Sect. 2 for details). This model preserves the local shape and ensures no mesh folding. Compared with 2D diffeomorphic image registration model in [31, 33] which is only constrained by two equalities, three inequalities are additionally added in the proposed model to ensure that Jacobian determinant of $\mathbf{h}$ is greater than 0 for any $\mathbf{x} \in \Omega$. In this view, the proposed model is much more technically challenging compared with the model in [31,33]. Around the proposed model, a relaxed model is proposed for the convenience of numerical implementation. The existence of solution for the relaxed model is proved. Moreover, an alternating directional projection 3D image registration algorithm is proposed. Numerical tests show that the proposed algorithm is competitive compared with the other four algorithms.

The rest of this paper is organized as follows: In Section 2, a 3D diffeomorphic image registration model without mesh folding is proposed. Based on Section 2, a relaxed model is also proposed and the existence of its solution is proved in Section 3. In addition, an alternating directional projection algorithm is proposed to solve the proposed model in Section 3. In Section 4, several numerical tests are performed to show the efficiency of the proposed projection algorithm. In Section 5, we summarize our work and list some problems for future research.

2. Proposed 3D diffeomorphic image registration model

At the beginning of this section, let's recall the following canonical theorem in Appendix C of [8].

Theorem 2.1 (Inverse function theorem)**.** *Suppose Ω is a simply connected domain and $\mathbf{h} \in C^1(\Omega)$, then $\mathbf{h} : \Omega \to \Omega$ is a local bijection if and only if $\det(\nabla \mathbf{h}(\mathbf{x})) \neq 0$ for any $\mathbf{x} \in \Omega$.*

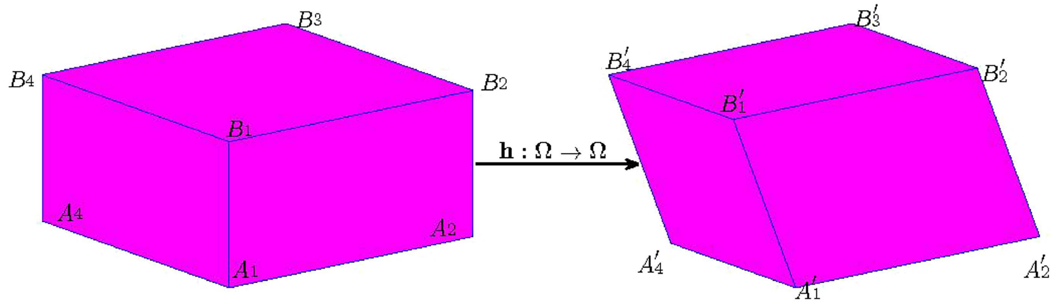

FIGURE 1. Role of $\mathcal{A}$ for producing the diffeomorphic deformation.

In physical view, $\det(\nabla\mathbf{h}(\mathbf{x}))$ denotes the volume stretching rate for the transformation $\mathbf{h}:\Omega\to\Omega$. Stretching rate is positive. Hence, the condition $\det(\nabla\mathbf{h}(\mathbf{x}))\neq 0$ in Theorem 2.1 is replaced by $\det(\nabla\mathbf{h}(\mathbf{x}))>0$ in this article. In addition, by additionally giving the boundary condition $\mathbf{h}(\mathbf{x})=\mathbf{x}$ on $\partial\Omega$ and defining $\mathbf{h}(\mathbf{x})=\mathbf{x}$ if $\mathbf{x}\in\mathbb{R}^3\setminus\Omega$, then $\mathbf{h}\in C^1(\mathbb{R}^3)$ with $\lim_{\|\mathbf{x}\|\to+\infty}\|\mathbf{h}(\mathbf{x})\|=+\infty$. It follows from the corollary on page 200 in [34] that $\mathbf{h}:\mathbb{R}^3\to\mathbb{R}^3$ is a C^1 global diffeomorphism. That is, the local bijection in Theorem 2.1 can be replaced by global bijection if the condition $\mathbf{h}(\mathbf{x})=\mathbf{x}$ on $\partial\Omega$ is added.

By Theorem 2.1, we know that constraint $\det(\nabla\mathbf{h}(\mathbf{x}))>0$ is necessary to eliminate mesh folding in 3D image registration models. To achieve $\det(\nabla\mathbf{h}(\mathbf{x}))>0$ for any $\mathbf{x}\in\Omega$, one can find several recent works which impose this constraint in direct way [35, 36]. However, in 3D space,

$$\begin{aligned}\det(\nabla\mathbf{h}(\mathbf{x}))&=\begin{vmatrix}1+\frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} & \frac{\partial u_1}{\partial x_3}\\ \frac{\partial u_2}{\partial x_1} & 1+\frac{\partial u_2}{\partial x_2} & \frac{\partial u_2}{\partial x_3}\\ \frac{\partial u_3}{\partial x_1} & \frac{\partial u_3}{\partial x_2} & 1+\frac{\partial u_3}{\partial x_3}\end{vmatrix}\\
&=\frac{\partial u_1}{\partial x_1}+\frac{\partial u_2}{\partial x_2}+\frac{\partial u_3}{\partial x_3}+\frac{\partial u_1}{\partial x_1}\frac{\partial u_2}{\partial x_2}-\frac{\partial u_1}{\partial x_2}\frac{\partial u_2}{\partial x_1}+\frac{\partial u_1}{\partial x_1}\frac{\partial u_3}{\partial x_3}\\
&\quad-\frac{\partial u_1}{\partial x_3}\frac{\partial u_3}{\partial x_1}+\frac{\partial u_2}{\partial x_2}\frac{\partial u_3}{\partial x_3}-\frac{\partial u_2}{\partial x_3}\frac{\partial u_3}{\partial x_2}+\frac{\partial u_1}{\partial x_1}\frac{\partial u_2}{\partial x_2}\frac{\partial u_3}{\partial x_3}-\frac{\partial u_1}{\partial x_1}\frac{\partial u_2}{\partial x_3}\frac{\partial u_3}{\partial x_2}\\
&\quad-\frac{\partial u_1}{\partial x_2}\frac{\partial u_2}{\partial x_1}\frac{\partial u_3}{\partial x_3}+\frac{\partial u_1}{\partial x_2}\frac{\partial u_2}{\partial x_3}\frac{\partial u_3}{\partial x_1}+\frac{\partial u_1}{\partial x_3}\frac{\partial u_2}{\partial x_1}\frac{\partial u_3}{\partial x_2}-\frac{\partial u_1}{\partial x_3}\frac{\partial u_2}{\partial x_2}\frac{\partial u_3}{\partial x_1}+1.\end{aligned}$$

Directly controlling $\det(\nabla\mathbf{h}(\mathbf{x}))>0$ seems to be a too complex inequality constraint for 3D image registration problem. It is also not convenient for numerical implementation. To simplify 3D image registration with direct inequality constraint $\det(\nabla\mathbf{h}(\mathbf{x}))>0$, we present a novel 3D diffeomorphic image registration model in this section. Before giving the proposed model, we define

$$\begin{aligned}\mathcal{A}\triangleq\Big\{\mathbf{u}=(u_1,u_2,u_3)^T\in[H_0^\alpha(\Omega)]^3:\frac{\partial u_1(\mathbf{x})}{\partial x_1}=\frac{\partial u_2(\mathbf{x})}{\partial x_2}=\frac{\partial u_3(\mathbf{x})}{\partial x_3}>-1,\\ \frac{\partial u_i(\mathbf{x})}{\partial x_j}=-\frac{\partial u_j(\mathbf{x})}{\partial x_i},1\leq i<j\leq 3\quad\text{for any}\quad\mathbf{x}\in\Omega\Big\}.\end{aligned}\tag{3}$$

Remark 2.2. There are three remarks on the set $\mathcal{A}$:

(i) As it is shown on Figure 1, cuboid $A_1A_2A_3A_4-B_1B_2B_3B_4$ is transformed into a parallelepiped $A_1'A_2'A_3'A_4'-B_1'B_2'B_3'B_4'$ by $\mathbf{h}(\mathbf{x})=\mathbf{x}+\mathbf{u}(\mathbf{x})$. We define $\gamma_{xy}=\angle A_4'A_1'A_2'-\angle A_4A_1A_2$, $\gamma_{xz}=\angle A_4'A_1'B_1'-\angle A_4A_1B_1$, $\gamma_{yz}=\angle A_2'A_1'B_1'-\angle A_2A_1B_1$. Note that here and in what follows, $\angle ABC$ denotes the included angle of edge BA and BC. In elastic mechanics, γ_{xy}, γ_{xz}, γ_{yz} are named shearing strain which represents the angle changes caused by $\mathbf{h}$. In addition, there holds

$$\gamma_{xy} = \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1},\ \gamma_{xz} = \frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1},\ \gamma_{yz} = \frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}. \tag{4}$$

By restricting $\mathbf{u}$ into the set $\mathcal{A}$, $\gamma_{xy} = \gamma_{xz} = \gamma_{yz} = 0$. That is, $\mathcal{A}$ produces diffeomorphic mappings $\mathbf{h}$ which locally preserve the angle. Compared with the conformal set which preserves the angle and stretching rate at the same time, $\mathcal{A}$ is a much more relaxed set.

(ii) Define $\varepsilon_i = \frac{\partial u_i}{\partial x_i}(i = 1, 2, 3)$, then ε_i are named strain on x_i direction. In physical view, $1 + \varepsilon_i$ denotes the stretching rate of $A_1A_2A_3A_4 - B_1B_2B_3B_4$ on x_i direction. This is the main reason why the inequality constraint $\frac{\partial u_i(\mathbf{x})}{\partial x_i} > -1$ is given. In addition, by giving the equality constraint $\frac{\partial u_1(\mathbf{x})}{\partial x_1} = \frac{\partial u_2(\mathbf{x})}{\partial x_2} = \frac{\partial u_3(\mathbf{x})}{\partial x_3}$, one can ensure that the local stretching rate is the same on three different directions, which preserves the local shape. Moreover, by restricting $\mathbf{u}$ into the set $\mathcal{A}$, one can notice that $\operatorname{div}(\mathbf{u}) = \sum_{i=1}^{3} \frac{\partial u_i}{\partial x_i} = 3\frac{\partial u_1}{\partial x_1} > -3$. That is, the divergence of $\mathbf{u}$ is lower bounded.

(iii) $\mathbf{u} \in \mathcal{A}$ implies, $\frac{\partial u_i}{\partial x_i} = \frac{\partial u_j}{\partial x_j}$ and $\frac{\partial u_i}{\partial x_j} = -\frac{\partial u_j}{\partial x_i}(1 \le i < j \le 3)$. Given two fixed i, j, $\frac{\partial u_i}{\partial x_i} = \frac{\partial u_j}{\partial x_j}$ and $\frac{\partial u_i}{\partial x_j} = -\frac{\partial u_j}{\partial x_i}$ is a Cauchy–Riemann equation in 2D space. Therefore, $\frac{\partial u_1}{\partial x_1} = \frac{\partial u_2}{\partial x_2} = \frac{\partial u_3}{\partial x_3}, \frac{\partial u_i}{\partial x_j} = -\frac{\partial u_j}{\partial x_i}, 1 \le i < j \le 3$ can be viewed as an extension of Cauchy–Riemann equations in 3D space [37]. Throughout this paper, we say that $\mathbf{u}$ satisfies Cauchy–Riemann constraint if $\frac{\partial u_i}{\partial x_i} = \frac{\partial u_j}{\partial x_j}$ and $\frac{\partial u_i}{\partial x_j} = -\frac{\partial u_j}{\partial x_i}(1 \le i < j \le 3)$ holds.

Based on these definitions, the proposed 3D diffeomorphic image registration model is formulated by:

$$\mathbf{u} = \arg\min_{\mathbf{u}\in\mathcal{A}} I(\mathbf{u}), \tag{5}$$

where $\alpha > 2.5$, $\mu > 0$,

$$I(\mathbf{u}) \triangleq S(\mathbf{u}) + \mu R_1(\mathbf{u}), \tag{6}$$

and $R_1(\mathbf{u}) = \int_\Omega |\nabla^\alpha \mathbf{u}|^2 \,\mathrm{d}\mathbf{x}$.

Note that throughout this paper, $\Omega = (0, a_1) \times (0, a_2) \times (0, a_3)$, for $\mathbf{x} = (x_1, x_2, x_3) \in \Omega$, $i = 1, 2, 3$ and function $f : \Omega \to \mathbb{R}$, define

$$\frac{\partial^\alpha f(\mathbf{x})}{\partial x_i^\alpha} \triangleq \frac{1}{\Gamma([\alpha]+1-\alpha)}\left(\frac{\partial}{\partial x_i}\right)^{[\alpha]+1} \int_0^{x_i} \frac{f^{(i)}(\mathbf{x}, t)}{(x_i - t)^{\alpha-[\alpha]}}\,\mathrm{d}t, \tag{7}$$

$$\frac{\partial^{\alpha*} f(\mathbf{x})}{\partial x_i^{\alpha*}} \triangleq \frac{1}{\Gamma([\alpha]+1-\alpha)}\left(-\frac{\partial}{\partial x_i}\right)^{[\alpha]+1} \int_{x_i}^{a_i} \frac{f^{(i)}(\mathbf{x}, t)}{(t - x_i)^{\alpha-[\alpha]}}\,\mathrm{d}t. \tag{8}$$

$\nabla^\alpha \mathbf{u} = \left(\frac{\partial^\alpha u_i}{\partial x_j^\alpha}\right)_{3\times 3}$, $\Gamma(s) = \int_0^{+\infty} t^{s-1} e^{-t}\,\mathrm{d}t$, $[\cdot]$ is round down function, $f^{(1)}(\mathbf{x}, t) = f(t, x_2, x_3)$, $f^{(2)}(\mathbf{x}, t) = f(x_1, t, x_3)$, $f^{(3)}(\mathbf{x}, t) = f(x_1, x_2, t)$, $|\nabla^\alpha \mathbf{u}| = \sqrt{\sum_{i,j=1}^{3}\left(\frac{\partial^\alpha u_i}{\partial x_j^\alpha}\right)^2}$ and the details of these two fractional-order derivatives can be founded in [9, 38].

Remark 2.3. There are four remarks on model (5):

(i) Here we use the fractional-order regularization $\int_\Omega |\nabla^\alpha \mathbf{u}|^2 \,\mathrm{d}\mathbf{x}$ because the differential order of the fractional-order operator can change continuously from zero to some positive number, which makes it possible to adjust the differential order to give a more accurate and more general image registration model.

(ii) $\mathbf{u} \in [H_0^\alpha(\Omega)]^3$ implies $\mathbf{u} \in [C^1(\Omega)]^3$ because of the fact that $H_0^\alpha(\Omega) \hookrightarrow C^1(\Omega)(\alpha > 2.5)$ (see [39], Thm. 4.58). That is, deformation $\mathbf{h} : \Omega \to \Omega$ is diffeomorphic. This ensures that the first-order derivatives in $\mathcal{A}$ are well defined. Therefore, model (5) can be extended into diffusion tensor imaging (DTI) $(T, D : \Omega \to \mathrm{SPD}(3))$

registration by replacing $S(\mathbf{u})$ with $\int_\Omega \|T \diamond \mathbf{h}(\mathbf{x}) - D(\mathbf{x})\|^2 \,\mathrm{d}\mathbf{x}$. Note that SPD(3) is a set which contains all 3×3 Symmetric Positive Definite real matrixes, and $T \diamond \mathbf{h}(\mathbf{x})$ is defined by

$$T \diamond \mathbf{h}(\mathbf{x}) = \mathbf{r}_\mathbf{x}[T \circ \mathbf{h}(\mathbf{x})]\mathbf{r}_\mathbf{x}^T \quad \text{with } \mathbf{r}_\mathbf{x} = \mathbf{J}_\mathbf{x}^T\big(\mathbf{J}_\mathbf{x}\mathbf{J}_\mathbf{x}^T\big)^{-\frac{1}{2}} \text{and } \mathbf{J}_\mathbf{x} = \nabla_\mathbf{x}\mathbf{h}^{-1}(\mathbf{x}). \tag{9}$$

One can refer to [11, 16] for details.

(iii) If $\mathbf{u} \in \mathcal{A}$, then for any $\mathbf{x} \in \Omega$,

$$\det(\nabla \mathbf{h}(\mathbf{x})) = \left(1 + \frac{\partial u_1}{\partial x_1}\right)\left[\left(1 + \frac{\partial u_1}{\partial x_1}\right)^2 + \left(\frac{\partial u_1}{\partial x_2}\right)^2 + \left(\frac{\partial u_1}{\partial x_3}\right)^2 + \left(\frac{\partial u_2}{\partial x_3}\right)^2\right] > 0. \tag{10}$$

That is, no mesh folding occurs in proposed model (5). This is the main motivation for us to add constraint $\mathbf{u} \in \mathcal{A}$ in the proposed model (5).

(iv) By letting $\frac{\partial u_1}{\partial x_3} = \frac{\partial u_2}{\partial x_3} = 0$ in $\mathcal{A}$, then $\mathcal{A}$ becomes a 3D conformal set $\mathcal{A}_c$. In this view, equation (5) is a much more relaxed model compared with 3D conformal image registration model.

3. Numerical solution of proposed model

At the beginning of this section, we introduce some definitions.

Here and in what follows, we define

$$R(\mathbf{u}) \triangleq \sum_{1\le i<j\le 3}\int_\Omega \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)^2 + \left(\frac{\partial u_i}{\partial x_i} - \frac{\partial u_j}{\partial x_j}\right)^2 \mathrm{d}\mathbf{x}, \tag{11}$$

and

$$J(\mathbf{u}) \triangleq I(\mathbf{u}) + \Theta R(\mathbf{u}), \tag{12}$$

where $\Theta > 0$ is a large number.

To transform proposed model (5) into a variational problem with less constraints, we make use of the following approximation procedure:

$$\mathbf{u} = \arg\min_{\mathbf{u}\in\mathcal{B}_\varepsilon} J(\mathbf{u}), \tag{13}$$

where $\mathcal{B}_\varepsilon = \{\mathbf{u} = (u_1, u_2, u_3)^T \in [H_0^\alpha(\Omega)]^3 : \frac{\partial u_1(\mathbf{x})}{\partial x_1} \ge -1 + \varepsilon \;\text{ for }\; \text{any } \mathbf{x} \in \Omega\}$ and $\varepsilon > 0$ is small enough.

Remark 3.1. In fact, the constraint set in (13) should be

$$\mathcal{B} = \left\{\mathbf{u} = (u_1, u_2, u_3)^T \in [H_0^\alpha(\Omega)]^3 : \frac{\partial u_1(\mathbf{x})}{\partial x_1} > -1 \;\text{ for }\; \text{any } \mathbf{x} \in \Omega\right\}. \tag{14}$$

Here we introduce an infinite small variable $\varepsilon > 0$ to construct a closed set $\mathcal{B}_\varepsilon$ which is necessary in the proof of Theorem 3.5. Moreover, the solution of (13) approximately satisfies $\frac{\partial u_i(\mathbf{x})}{\partial x_j} + \frac{\partial u_j(\mathbf{x})}{\partial x_i} = 0 (1 \le i < j \le 3)$, $\frac{\partial u_1(\mathbf{x})}{\partial x_1} = \frac{\partial u_2(\mathbf{x})}{\partial x_2} = \frac{\partial u_3(\mathbf{x})}{\partial x_3} \ge -1 + \varepsilon$ for any $\mathbf{x} \in \Omega$ when Θ is large enough (see (29) and (30) for details).

Concerning the existence of solution for the relaxed model (13), we have the following results:

Theorem 3.2. *Assume that* $\max_{\mathbf{x}\in\Omega}|T(\mathbf{x})| < M < +\infty$, $\max_{\mathbf{x}\in\Omega}|D(\mathbf{x})| < M < +\infty$ *and* $\triangle_T \triangleq \{\mathbf{x} : T(\mathbf{x}) \text{ is discontinuous at } \mathbf{x}\}$ *is a zero measure set; then* (13) *admits a solution* $\mathbf{u} \in \mathcal{B}_\varepsilon$.

Proof. Choose a minimizing sequence $\{\mathbf{u}^n\} \subseteq \mathcal{B}_\varepsilon$ of $J(\mathbf{u})$ and let

$$\lim_{n\to\infty} J(\mathbf{u}^n) = m = \inf_{\mathbf{w}\in\mathcal{B}_\varepsilon} J(\mathbf{w}). \tag{15}$$

By the fact $J(\mathbf{u}^n) \leq J(\mathbf{0}) = S(\mathbf{0}) < +\infty$; there exists a constant M_1 such that

$$\|\nabla^\alpha \mathbf{u}^n\|^2_{[L^2(\Omega)]^3} = \int_\Omega |\nabla^\alpha \mathbf{u}^n|^2 \,\mathrm{d}\mathbf{x} \leq M_1. \tag{16}$$

On the other hand; by the Lemma 3.3 in [9], we know that for any $\mathbf{u}^n \in [H_0^\alpha(\Omega)]^3$, there holds

$$\|\mathbf{u}^n\|^2_{[L^2(\Omega)]^3} \leq C\|\nabla^\alpha \mathbf{u}^n\|^2_{[L^2(\Omega)]^3}, \tag{17}$$

for some constant $C = C(\alpha, \Omega) > 0$.

Therefore,

$$\|\mathbf{u}^n\|^2_{\left[H_0^\alpha(\Omega)\right]^3} = \|\mathbf{u}^n\|^2_{[L^2(\Omega)]^3} + \|\nabla^\alpha \mathbf{u}^n\|^2_{[L^2(\Omega)]^3} \leq \tilde{M} < +\infty, \tag{18}$$

for some constant $\tilde{M}$.

By (17) and (18), we know the semi-norm defined by $|\mathbf{u}|_{[H_0^\alpha(\Omega)]^3} = \left(\int_\Omega |\nabla^\alpha \mathbf{u}|^2 \,\mathrm{d}\mathbf{x}\right)^{\frac{1}{2}}$ is equivalent to norm $\|\mathbf{u}\|_{[H_0^\alpha(\Omega)]^3}$ on space $[H_0^\alpha(\Omega)]^3$. That is, $|\mathbf{u}|^2_{[H_0^\alpha(\Omega)]^3} \leq \|\mathbf{u}\|^2_{[H_0^\alpha(\Omega)]^3} \leq C_0|\mathbf{u}|^2_{[H_0^\alpha(\Omega)]^3}$ (see [40], Prop. 1.5).

It follows from the fact $H_0^\alpha(\Omega)$ is compactly embedded into $C^1(\Omega)$ ([39], Thm. 4.58) that there exists a subsequence of $\{\mathbf{u}^n\}$ which is still labeled by n and $\mathbf{u} \in [C^1(\Omega)]^3$ such that

$$\|\mathbf{u}^n - \mathbf{u}\|_{[C^1(\Omega)]^3} \xrightarrow{n} 0. \tag{19}$$

Based on these conclusions, now we claim that

$$J(\mathbf{u}) = \min_{\mathbf{w}\in\mathcal{B}_\varepsilon} J(\mathbf{w}). \tag{20}$$

First, by the lower semi continuity of norm (see D4 on page 639 of [8]) and (18), we obtain that

$$\int_\Omega |\nabla^\alpha \mathbf{u}(\mathbf{x})|^2 \,\mathrm{d}\mathbf{x} \leq \liminf_{n\to+\infty} \int_\Omega |\nabla^\alpha \mathbf{u}^n(\mathbf{x})|^2 \,\mathrm{d}\mathbf{x}, \tag{21}$$

because $|\cdot|_{[H_0^\alpha(\Omega)]^3}$ is a norm in space $[H_0^\alpha(\Omega)]^3$.

Therefore,

$$\mu \int_\Omega |\nabla^\alpha \mathbf{u}(\mathbf{x})|^2 \,\mathrm{d}\mathbf{x} \leq \liminf_{n\to+\infty} \mu \int_\Omega |\nabla^\alpha \mathbf{u}^n(\mathbf{x})|^2 \,\mathrm{d}\mathbf{x}. \tag{22}$$

Furthermore, by (19), we obtain that,

$$\begin{aligned} |S(\mathbf{u}^n) - S(\mathbf{u})| &\leq K \int_\Omega |T(\mathbf{x} + \mathbf{u}^n(\mathbf{x})) - T(\mathbf{x} + \mathbf{u}(\mathbf{x}))| \,\mathrm{d}\mathbf{x} \\ &= K \int_{\Omega\setminus\Delta_T} |T(\mathbf{x} + \mathbf{u}^n(\mathbf{x})) - T(\mathbf{x} + \mathbf{u}(\mathbf{x}))| \,\mathrm{d}\mathbf{x} \xrightarrow{n} 0, \end{aligned} \tag{23}$$

because of the fact that $T(\mathbf{x})$ is continuous on $\Omega \setminus \Delta_T$ and $T(\mathbf{x})$ is bounded on Ω. Note that here $\mathbf{h}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$ maps $\Omega \setminus \Delta_T$ into some set $\Omega \setminus \Delta_{\bar{T}}$ (also for $\mathbf{h}^n(\mathbf{x}) = \mathbf{x} + \mathbf{u}^n(\mathbf{x})$), where $\Delta_{\bar{T}}$ is also a zero measure set. For simplicity, here we still write $\Delta_{\bar{T}}$ as Δ_T.

That is,

$$\lim_{n\to+\infty} S(\mathbf{u}^n) = S(\mathbf{u}). \tag{24}$$

Moreover, it follows from (19) that $\frac{\partial u_i^n(\mathbf{x})}{\partial x_j} \xrightarrow{n} \frac{\partial u_i(\mathbf{x})}{\partial x_j}$ for $i,j=1,2,3$ and any $\mathbf{x}\in\Omega$. This implies $\mathbf{u}\in\mathcal{B}_\varepsilon$ and

$$\lim_{n\to+\infty} R(\mathbf{u}^n) = R(\mathbf{u}). \tag{25}$$

By (22), (24) and (25), we know that

$$m = \liminf_{n\to+\infty} J(\mathbf{u}^n) \geq J(\mathbf{u}). \tag{26}$$

On the other hand, it follows from (15) that

$$J(\mathbf{u}) \geq m, \tag{27}$$

because of the fact that $\mathbf{u}\in\mathcal{B}_\varepsilon$.

By (26) and (27), we conclude the claim (20). Moreover, by using the Sobolev embedding theorem $H_0^\alpha(\Omega) \hookrightarrow C^1(\Omega)$ ([39], Thm. 4.58), $\mathbf{u}\in\mathcal{B}_\varepsilon$ implies $\mathbf{u}\in C^1(\Omega)$. This concludes Theorem 3.2. □

Now, we give the reason to let Θ be a large number in the relaxed model (13). Let $\mathbf{u}^\Theta = (u_1^\Theta, u_2^\Theta, u_3^\Theta)^T$ be the solution of (13); then there holds,

$$0 \leq \Theta R\big(\mathbf{u}^\Theta\big) \leq J\big(\mathbf{u}^\Theta\big) \leq J(\mathbf{0}) = I(\mathbf{0}) \triangleq M \qquad \forall\ \ \Theta > 0. \tag{28}$$

By (28), we obtain that

$$\lim_{\Theta\to+\infty} R\big(\mathbf{u}^\Theta\big) = 0. \tag{29}$$

Concerning the asymptotic analysis of $R(\mathbf{u}^\Theta)$, one can also refer to Appendix C for details.

By (29), we know that three components of $\mathbf{u}^\Theta$ approximately satisfy

$$\frac{\partial u_1^\Theta(\mathbf{x})}{\partial x_1} = \frac{\partial u_2^\Theta(\mathbf{x})}{\partial x_2} = \frac{\partial u_3^\Theta(\mathbf{x})}{\partial x_3} \geq -1+\varepsilon, \frac{\partial u_i^\Theta(\mathbf{x})}{\partial x_j} = -\frac{\partial u_j^\Theta(\mathbf{x})}{\partial x_i}, \tag{30}$$

for $1\leq i<j\leq 3$ when $\Theta>0$ is large enough.

Therefore,

$$\det(\nabla \mathbf{h}^\Theta(\mathbf{x})) = \left(1+\frac{\partial u_1^\Theta}{\partial x_1}\right)\left[\left(1+\frac{\partial u_1^\Theta}{\partial x_1}\right)^2 + \left(\frac{\partial u_1^\Theta}{\partial x_2}\right)^2 + \left(\frac{\partial u_1^\Theta}{\partial x_3}\right)^2 + \left(\frac{\partial u_2^\Theta}{\partial x_3}\right)^2\right] > 0 \tag{31}$$

for any $\mathbf{x}\in\Omega$.

This ensures no mesh folding. It is also the main reason to let Θ be large enough in model (13).

Next, we give some results on numerical implementation of problem (13).

Theorem 3.3. *Let $\mathbf{u}\in\mathcal{B}_\varepsilon$ be the solution of (13), then $\mathbf{u}$ satisfies the following variational inequality*

$$\int_\Omega \mathcal{F}(\mathbf{u})\cdot(\mathbf{w}-\mathbf{u})\,\mathrm{d}\mathbf{x} \geq 0 \quad \text{for all } \mathbf{w}\in\mathcal{B}_\varepsilon, \tag{32}$$

where $\mathcal{F}(\mathbf{u}) = [T(\mathbf{x}+\mathbf{u}(\mathbf{x})) - D(\mathbf{x})]\nabla T(\mathbf{x}+\mathbf{u}(\mathbf{x})) + \mu\,\mathrm{div}^{\alpha *}(\nabla^\alpha \mathbf{u}) + \Theta\mathbf{W}$, $\mathbf{W} = (\epsilon_1(\mathbf{u})+\eta_1(\mathbf{u}), \epsilon_2(\mathbf{u})+\eta_2(\mathbf{u}), \epsilon_3(\mathbf{u})+\eta_3(\mathbf{u}))^T$, $\epsilon_1(\mathbf{u}) = -\frac{\partial}{\partial x_1}\left(\frac{\partial u_1}{\partial x_1}-\frac{\partial u_2}{\partial x_2}\right) - \frac{\partial}{\partial x_1}\left(\frac{\partial u_1}{\partial x_1}-\frac{\partial u_3}{\partial x_3}\right)$, $\epsilon_2(\mathbf{u}) = \frac{\partial}{\partial x_2}\left(\frac{\partial u_1}{\partial x_1}-\frac{\partial u_2}{\partial x_2}\right) - \frac{\partial}{\partial x_2}\left(\frac{\partial u_2}{\partial x_2}-\frac{\partial u_3}{\partial x_3}\right)$, $\epsilon_3(\mathbf{u}) = \frac{\partial}{\partial x_3}\left(\frac{\partial u_1}{\partial x_1}-\frac{\partial u_3}{\partial x_3}\right) + \frac{\partial}{\partial x_3}\left(\frac{\partial u_2}{\partial x_2}-\frac{\partial u_3}{\partial x_3}\right)$, $\eta_1(\mathbf{u}) = -\frac{\partial}{\partial x_2}\left(\frac{\partial u_1}{\partial x_2}+\frac{\partial u_2}{\partial x_1}\right) - \frac{\partial}{\partial x_3}\left(\frac{\partial u_1}{\partial x_3}+\frac{\partial u_3}{\partial x_1}\right)$, $\eta_2(\mathbf{u}) = -\frac{\partial}{\partial x_1}\left(\frac{\partial u_1}{\partial x_2}+\frac{\partial u_2}{\partial x_1}\right) - \frac{\partial}{\partial x_3}\left(\frac{\partial u_2}{\partial x_3}+\frac{\partial u_3}{\partial x_2}\right)$, $\eta_3(\mathbf{u}) = -\frac{\partial}{\partial x_1}\left(\frac{\partial u_1}{\partial x_3}+\frac{\partial u_3}{\partial x_1}\right) - \frac{\partial}{\partial x_2}\left(\frac{\partial u_2}{\partial x_3}+\frac{\partial u_3}{\partial x_2}\right)$.

Proof. Note that $\mathcal{B}_\varepsilon$ is a convex set. Fix $\mathbf{w} \in \mathcal{B}_\varepsilon$, then

$$\mathbf{u} + \tau(\mathbf{w} - \mathbf{u}) = (1-\tau)\mathbf{u} + \tau\mathbf{w} \in \mathcal{B}_\varepsilon \qquad 0 \le \tau \le 1. \tag{33}$$

Set $j(\tau) \triangleq J(\mathbf{u} + \tau(\mathbf{w} - \mathbf{u}))$, by the fact $j(0) \le j(\tau)(0 \le \tau \le 1)$, we know that

$$j'(0) \ge 0. \tag{34}$$

Therefore,

$$j'(0) = \lim_{\tau \to 0^+} \frac{j(\tau) - j(0)}{\tau} = \int_\Omega \mathcal{F}(\mathbf{u}) \cdot (\mathbf{w} - \mathbf{u})\,\mathrm{d}\mathbf{x} \ge 0 \quad \text{for all } \mathbf{w} \in \mathcal{B}_\varepsilon. \tag{35}$$

Note that for any $g \in H_0^\alpha(\Omega)$ implies, $\frac{\partial^k g(\mathbf{x})}{\partial x_i^k}|_{\mathbf{x} \in \partial\Omega} = 0$, $(k = 0, 1, 2, \cdots, [\alpha]; i = 1, 2, 3)$ (*cf.* Sobolev space in [8, 9, 39]). That is, any function on $H_0^\alpha(\Omega)$ satisfies the homogeneous boundary conditions. By giving homogeneous boundary conditions, the following two important properties of fractional-order derivatives are used when deriving (35):

(i) Riemann–Liouville derivatives, Grunwald–Letnikov derivatives and Caputo derivatives are equivalent [38];
(ii) The following integration by parts formula holds

$$\int_{a_i}^{b_i} \xi(\mathbf{x}) \cdot \frac{\partial^\alpha f(\mathbf{x})}{\partial x_i^\alpha}\,\mathrm{d}x_i = \int_{a_i}^{b_i} \frac{\partial^{\alpha*} \xi(\mathbf{x})}{\partial x_i^{\alpha*}} \cdot f(\mathbf{x})\,\mathrm{d}x_i \quad i = 1, 2, 3. \tag{36}$$

$\square$

Now, we discuss the problem of how to find numerical solution of variational inequality (32). Before we give our results, let's recall an important lemma [41] which is necessary in our proof.

Lemma 3.4. *Suppose C is a closed, convex set, $A : C \to \mathbb{R}^3$ is monotone, then $\mathbf{u}$ is the solution of variational inequality*

$$(A\mathbf{u}, \mathbf{w} - \mathbf{u}) \ge 0 \qquad \forall\ \mathbf{w} \in C, \tag{37}$$

if and only if $\mathbf{u}$ solves the fixed point equation

$$\mathbf{u} = P_C[\mathbf{u} - \rho A\mathbf{u}], \tag{38}$$

for an arbitrary positive constant ρ.

For technical demand in later proof, we define $\mathcal{B}_\varepsilon^M = \{\mathbf{u} = (u_1, u_2, u_3)^T \in [H_0^\alpha(\Omega)]^3 : \|\mathbf{u}\|_{[H_0^\alpha(\Omega)]^3} \le M, \frac{\partial u_1(\mathbf{x})}{\partial x_1} \ge -1 + \varepsilon$ for any $\mathbf{x} \in \Omega\}$, where M is a very large number. Note that the solution of (13) is bounded on $[H_0^\alpha(\Omega)]^3$ (*i.e.*, $J(\mathbf{u}) \le J(\mathbf{0}) = \|T(\cdot) - D(\cdot)\|^2_{L^2(\Omega)}$), $\mathcal{B}_\varepsilon^M$ contains all the solutions of (13) when M is large enough. Hence, throughout this paper, we do not distinguish $\mathcal{B}_\varepsilon^M$ and $\mathcal{B}_\varepsilon$ except for the following theorem. Moreover, we assume that $T(\cdot)$ and $\nabla T(\cdot)$ are Lipschitz continuous on Ω with Lipschitz constant L.

Theorem 3.5. *Let* $\sup_{\mathbf{x} \in \Omega} |T(\mathbf{x})| < M < +\infty$, $\sup_{\mathbf{x} \in \Omega} |D(\mathbf{x})| < M < +\infty$, $\sup_{\mathbf{x} \in \Omega} |\nabla T(\mathbf{x})| < M < +\infty$, *and* $\mu \ge 3\text{CML}$ *for some $C = C(\Omega, \alpha) > 0$, then the function $\mathbf{u} \in \mathcal{B}_\varepsilon^M$ is a solution of* (32) *if and only if $\mathbf{u} \in \mathcal{B}_\varepsilon^M$ satisfies*

$$\mathbf{u} = P_{\mathcal{B}_\varepsilon^M}[\mathbf{u} - \rho\mathcal{F}(\mathbf{u})], \tag{39}$$

where $\rho > 0$ is a small number and $P_{\mathcal{B}_\varepsilon^M}(\mathbf{v})$ is projection of $\mathbf{v}$ on convex set $\mathcal{B}_\varepsilon^M$.

Proof. Firstly, we claim that

$$\mathcal{F} \text{ is a monotone operator.} \tag{40}$$

Define $\mathcal{F}(\mathbf{u}) = F_1(\mathbf{u}) + F_2(\mathbf{u}) + F_3(\mathbf{u})$, where $F_1(\mathbf{u}) = [T(\mathbf{x}+\mathbf{u}(\mathbf{x})) - D(\mathbf{x})]\nabla T(\mathbf{x}+\mathbf{u}(\mathbf{x}))$, $F_2(\mathbf{u}) = \mu \mathrm{div}^{\alpha *}(\nabla^\alpha \mathbf{u})$ and $F_3(\mathbf{u}) = \Theta \mathbf{W}$, then for $\mathbf{u}, \mathbf{w} \in \mathcal{B}_\varepsilon^M$,

$$\begin{aligned}
&\int_\Omega [\mathcal{F}(\mathbf{u}) - \mathcal{F}(\mathbf{w})] \cdot (\mathbf{u}-\mathbf{w})\,\mathrm{d}\mathbf{x} \\
&\quad = \int_\Omega \{F_1(\mathbf{u}) + F_2(\mathbf{u}) + F_3(\mathbf{u}) - [F_1(\mathbf{w}) + F_2(\mathbf{w}) + F_3(\mathbf{w})]\} \cdot (\mathbf{u}-\mathbf{w})\,\mathrm{d}\mathbf{x} \\
&\quad = \int_\Omega \{F_1(\mathbf{u}) - F_1(\mathbf{w})\} \cdot (\mathbf{u}-\mathbf{w})\,\mathrm{d}\mathbf{x} + \int_\Omega \{F_2(\mathbf{u}) + F_3(\mathbf{u}) - [F_2(\mathbf{w}) + F_3(\mathbf{w})]\} \cdot (\mathbf{u}-\mathbf{w})\,\mathrm{d}\mathbf{x}.
\end{aligned} \tag{41}$$

Since $F_2(\mathbf{u})$ and $F_3(\mathbf{u})$ are linear operators on $\mathbf{u}$, by using the integration by parts formula, one can easily check that

$$\int_\Omega \{F_2(\mathbf{u}) + F_3(\mathbf{u}) - [F_2(\mathbf{w}) + F_3(\mathbf{w})]\} \cdot (\mathbf{u}-\mathbf{w})\,\mathrm{d}\mathbf{x} = \mu \|\nabla^\alpha(\mathbf{u}-\mathbf{w})\|^2_{L^2(\Omega)} + \Theta R(\mathbf{u}-\mathbf{w}). \tag{42}$$

On the other hand,

$$\begin{aligned}
&\int_\Omega \{F_1(\mathbf{u}) - F_1(\mathbf{w})\} \cdot (\mathbf{u}-\mathbf{w})\,\mathrm{d}\mathbf{x} = \int_\Omega \{F_1(\mathbf{u}) - F_1(\mathbf{w})\} \cdot (\mathbf{u}-\mathbf{w})\,\mathrm{d}\mathbf{x} \\
&\quad = \int_\Omega [T(\mathbf{x}+\mathbf{u}) - D(\mathbf{x})]\nabla T(\mathbf{x}+\mathbf{u}) - [T(\mathbf{x}+\mathbf{w}) - D(\mathbf{x})]\nabla T(\mathbf{x}+\mathbf{w})\} \cdot (\mathbf{u}-\mathbf{w})\,\mathrm{d}\mathbf{x} \\
&\quad = \int_\Omega [T(\mathbf{x}+\mathbf{u}) - D(\mathbf{x})][\nabla T(\mathbf{x}+\mathbf{u}) - \nabla T(\mathbf{x}+\mathbf{w})] \cdot (\mathbf{u}-\mathbf{w})\,\mathrm{d}\mathbf{x} \\
&\qquad + \int_\Omega [T(\mathbf{x}+\mathbf{u}) - T(\mathbf{x}+\mathbf{w})]\nabla T(\mathbf{x}+\mathbf{w}) \cdot (\mathbf{u}-\mathbf{w})\,\mathrm{d}\mathbf{x} \\
&\quad \geq -3ML\|\mathbf{u}-\mathbf{w}\|^2_{L^2(\Omega)}.
\end{aligned} \tag{43}$$

Note that here we use the assumption that $T(\cdot)$ and $\nabla T(\cdot)$ are Lipschitz continuous on Ω with Lipschitz constant L.

It follows from (42) and (43) that

$$\int_\Omega [\mathcal{F}(\mathbf{u}) - \mathcal{F}(\mathbf{w})] \cdot (\mathbf{u}-\mathbf{w})\,\mathrm{d}\mathbf{x} \geq \mu\|\nabla^\alpha(\mathbf{u}-\mathbf{w})\|^2_{L^2(\Omega)} + \Theta R(\mathbf{u}-\mathbf{w}) - 3ML\|\mathbf{u}-\mathbf{w}\|^2_{L^2(\Omega)}. \tag{44}$$

By the Lemma 3.3 in [9], there exists a $C = C(\Omega, \alpha) > 0$ such that $\|\mathbf{u}-\mathbf{w}\|^2_{L^2(\Omega)} \leq C\|\nabla^\alpha(\mathbf{u}-\mathbf{w})\|^2_{L^2(\Omega)}$. Therefore,

$$\int_\Omega [\mathcal{F}(\mathbf{u}) - \mathcal{F}(\mathbf{w})] \cdot (\mathbf{u}-\mathbf{w})\,\mathrm{d}\mathbf{x} \geq \Big(\frac{\mu}{C} - 3ML\Big)\|\mathbf{u}-\mathbf{w}\|^2_{L^2(\Omega)} + \Theta R(\mathbf{u}-\mathbf{w}) \geq 0. \tag{45}$$

Note that here we use the assumption $\mu \geq$ 3CML to ensure $\frac{\mu}{C} - 3ML \geq 0$.

This concludes the claim (40).

On the other hand, $\mathcal{B}_\varepsilon^M$ is a convex and closed set. By Lemma 3.4, we conclude that $\mathbf{u} \in \mathcal{B}_\varepsilon^M$ is a solution of (32) if and only if $\mathbf{u} \in \mathcal{B}_\varepsilon^M$ satisfies

$$\mathbf{u} = P_{\mathcal{B}_\varepsilon^M}[\mathbf{u} - \rho\mathcal{F}(\mathbf{u})]. \tag{46}$$

□

Algorithm 1. Projection algorithm for 3D Diffeomorphic image registration.

Initialization: Given $\mathbf{u}^0 \in \mathcal{B}_\varepsilon$ (*e.g.*, $\mathbf{u}^0 = \mathbf{0}$), $\rho > 0, \mu, \Theta, \alpha$, compute $\mathbf{u}^{n+1}$ by the following procedure.
Set $n = 0$, norm=1, accuracy=0.001 and $K = 50$.
While norm> accuracy and $n < K$
do: Compute $\mathbf{u}^{n+1} = P_{\mathcal{B}_\varepsilon}[\mathbf{u}^n - \rho\mathcal{F}(\mathbf{u}^n)]$ (see Algorithm 2 for details);
Set norm=$\max_{\mathbf{x}\in\Omega} \|\mathbf{u}^n - \mathbf{u}^{n+1}\|$ and $n = n + 1$.
EndWhile
Return: image $T(\mathbf{x} + \mathbf{u}^{K_0}(\mathbf{x}))$ for some $K_0 \leq K$.

Motivated by Theorem 3.5 and Chapter 5 in [42], variational inequality (32) can be solved by iteration formula $\mathbf{u}^{n+1} = P_{\mathcal{B}_\varepsilon}[\mathbf{u}^n - \rho\mathcal{F}(\mathbf{u}^n)](n = 0, 1, 2, \cdots)$. Based on this formula, we give the following 3D image registration algorithm (Algorithm 1).

Remark 3.6. The discretization of $\mathcal{F}(\mathbf{u}^n)$ can be found in Appendix A. The convergence of Algorithm 1 is out of the scope of this paper. Perhaps, we will give some results on this topic in forthcoming work.

This arises a problem of how to compute the projection $P_{\mathcal{B}_\varepsilon}(\mathbf{v})$ for any given 3D vector field $\mathbf{v}$. In fact,

$$P_{\mathcal{B}_\varepsilon}(\mathbf{v}) = \arg\min_{\mathbf{u}\in\mathcal{B}_\varepsilon} \|\mathbf{u} - \mathbf{v}\|_2^2. \tag{47}$$

Equation (47) is essentially a functional minimization problem. It's a nonlocal problem because of the inequality $\frac{\partial u_1(\mathbf{x})}{\partial x_1} \geq -1 + \varepsilon$ in $\mathcal{B}_\varepsilon$.

Now, we give a method to compute $P_{\mathcal{B}_\varepsilon}(\mathbf{v})$.

Ω is discretized in following way: For $N_1, N_2, N_3 \in \mathbb{N}^+$, define $h_1 = \frac{a_1}{N_1+1}$, $h_2 = \frac{a_2}{N_2+1}$, $h_3 = \frac{a_3}{N_3+1}$, $x_p \triangleq x_{1,p} = ph_1$, $y_q \triangleq x_{2,q} = qh_2$, $z_r \triangleq x_{3,r} = rh_3$ and $\mathbf{X}_{p,q,r} = (x_{1,p}, x_{2,q}, x_{3,r})$ for $p = 0, 1, \cdots, N_1 + 1$; $q = 0, 1, \cdots, N_2 + 1$; $r = 0, 1, \cdots, N_3 + 1$. Define $B_{N_\beta}(\beta = 1, 2, 3)$ to be the standard two-point finite difference approximation to $\frac{\partial}{\partial x_\beta}$, for example, the central difference operator

$$B_{N_\beta} = \frac{1}{h_\beta}\begin{pmatrix} 0 & 0.5 & 0 & \cdots & 0 & 0 \\ -0.5 & 0 & 0.5 & \cdots & 0 & 0 \\ 0 & -0.5 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0.5 \\ 0 & 0 & 0 & \cdots & -0.5 & 0 \end{pmatrix}_{N_\beta \times N_\beta}. \tag{48}$$

Let $u_\beta^{p,q,r}(\beta = 1, 2, 3)$ denote the value $u_\beta(\mathbf{X}_{p,q,r})$ and $\mathbf{u}_1 = (u_1^{1,1,1}, u_1^{2,1,1}, \cdots, u_1^{N_1,N_2,N_3})^T_{N\times 1}$, $\mathbf{u}_2 = (u_2^{1,1,1}, u_2^{1,2,1}, \cdots, u_2^{N_1,N_2,N_3})^T_{N\times 1}$, $\mathbf{u}_3 = (u_3^{1,1,1}, u_3^{1,1,2}, \cdots, u_3^{N_1,N_2,N_3})^T_{N\times 1}$, where $N = N_1N_2N_3$. Based on these notations, the constraint $\frac{\partial u_1(\mathbf{x})}{\partial x_1} = \frac{\partial u_2(\mathbf{x})}{\partial x_2} = \frac{\partial u_3(\mathbf{x})}{\partial x_3} \geq -1 + \varepsilon$ in $\mathcal{B}_\varepsilon$ can be rewritten as:

$$B_\beta \mathbf{u}_\beta \geq -C_\varepsilon \qquad \beta = 1, 2, 3, \tag{49}$$

where B_β is an $N \times N$ block diagonal matrix with $B_\beta = \begin{pmatrix} B_{N_\beta} & 0 & \cdots & 0 \\ 0 & B_{N_\beta} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_{N_\beta} \end{pmatrix}$, $C_\varepsilon = (1-\varepsilon, 1-\varepsilon, \cdots, 1-\varepsilon)^T_{N\times 1}$.

Remark 3.7. The constraint in $\mathcal{B}_\varepsilon$ is $\frac{\partial u_1}{\partial x_1} \geq -1+\varepsilon$. To divide the calculation of $P_{\mathcal{B}_\varepsilon}(\mathbf{v})$ into three independent subproblems, we add constraints $\frac{\partial u_\beta}{\partial x_\beta} \geq -1+\varepsilon (\beta = 2,3)$ because of the constraints $\frac{\partial u_i}{\partial x_i} = \frac{\partial u_j}{\partial x_j} (1 \leq i < j \leq 3)$ in the proposed model (5).

Based on (49), the projection $P_{\mathcal{B}_\varepsilon}(\mathbf{v})$ for any given $\mathbf{v} \in \mathbb{R}^{3N}$ can be formulated as

$$P_{\mathcal{B}_\varepsilon}(\mathbf{v}) = \arg\min_{\mathbf{u}\in\mathbb{R}^{3N}} \sum_{\beta=1}^{3}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\sum_{k=1}^{N_3}\left(u_\beta^{i,j,k} - v_\beta^{i,j,k}\right)^2, \qquad \text{s.t. } B_\beta \mathbf{u}_\beta + C_\varepsilon \geq \mathbf{0} \qquad \beta = 1,2,3. \tag{50}$$

Because there is no coupled term of $u_\beta^{i,j,k} (\beta = 1,2,3)$ in (50), equation (50) is equivalent to the following three independent subproblems:

$$\mathbf{u}_\beta = \arg\min_{\mathbf{u}_\beta\in\mathbb{R}^{N}} \sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\sum_{k=1}^{N_3}\left(u_\beta^{i,j,k} - v_\beta^{i,j,k}\right)^2, \qquad \text{s.t. } B_\beta \mathbf{u}_\beta + C_\varepsilon \geq \mathbf{0} \qquad \beta = 1,2,3. \tag{51}$$

These are three convex optimization problems with inequality constraints. Note that here B_β is a block diagonal matrix. Motivated by Han [9,43], we hope to divide 3D problem (50) into several parallel 1D problems. For this purpose, equation (51) is divided into the following 3 parallel 1D subproblems:

(i) For fixed j,k, solving $(u_1^{\cdot,j,k})^{n+1}$ by

$$\left(u_1^{\cdot,j,k}\right)^{n+1} = \arg\min_{X_{j,k}\in\mathbb{R}^{N_1}} \frac{1}{2} X_{j,k}^T H_x X_{j,k} + f_{j,k}^x X_{j,k}, \tag{52}$$

with constraints $B_{N_1} X_{j,k} + C_{N_1,\varepsilon} \geq \mathbf{0}$, where here and in what follows, $X_{j,k}$ is a $N_1 \times 1$ vector, I_K is identity matrix of order K, $f_{j,k}^x = -2(v_1^{1,j,k}, v_1^{2,j,k}, \cdots, v_1^{N_1,j,k})$, $H_x = 2I_{N_1}$, $C_{N_1,\varepsilon} = (1-\varepsilon, 1-\varepsilon, \cdots, 1-\varepsilon)_{1\times N_1}^T$.

(ii) For fixed i,k, solving $(u_2^{i,\cdot,k})^{n+1}$ by

$$\left(u_2^{i,\cdot,k}\right)^{n+1} = \arg\min_{Y_{i,k}\in\mathbb{R}^{N_2}} \frac{1}{2} Y_{i,k}^T H_y Y_{i,k} + f_{i,k}^y Y_{i,k}, \tag{53}$$

with constraints $B_{N_2} Y_{i,k} + C_{N_2,\varepsilon} \geq \mathbf{0}$, where $Y_{i,k}$ is a $N_2 \times 1$ vector, $f_{i,k}^y = -2(v_2^{i,1,k}, v_2^{i,2,k}, \cdots, v_2^{i,N_2,k})$, $H_y = 2I_{N_2}$, $C_{N_2,\varepsilon} = (1-\varepsilon, 1-\varepsilon, \cdots, 1-\varepsilon)_{1\times N_2}^T$.

(iii) For fixed i,j, solving $(u_3^{i,j,\cdot})^{n+1}$ by

$$\left(u_3^{i,j,\cdot}\right)^{n+1} = \arg\min_{Z_{i,j}\in\mathbb{R}^{N_3}} \frac{1}{2} Z_{i,j}^T H_z Z_{i,j} + f_{i,j}^z Z_{i,j}, \tag{54}$$

with constraints $B_{N_3} Z_{i,j} + C_{N_3,\varepsilon} \geq \mathbf{0}$, where $Z_{i,j}$ is a $N_3 \times 1$ vector, $f_{i,j}^z = -2(v_3^{i,j,1}, v_3^{i,j,2}, \cdots, v_3^{i,j,N_3})$, $H_z = 2I_{N_3}$, $C_{N_3,\varepsilon} = (1-\varepsilon, 1-\varepsilon, \cdots, 1-\varepsilon)_{1\times N_3}^T$.

Remark 3.8. By ignoring the constant term in objective function, equations (52)–(54) are equivalent to

$$\left(u_1^{\cdot,j,k}\right)^{n+1} = \arg\min_{u_1^{\cdot,j,k}\in\mathbb{R}^{N_1}} \sum_{i=1}^{N_1}\left(u_1^{i,j,k} - v_1^{i,j,k}\right)^2, \quad \text{s.t.} \quad B_{N_1} u_1^{\cdot,j,k} + C_{N_1,\varepsilon} \geq 0, \tag{55}$$

$$\left(u_2^{i,\cdot,k}\right)^{n+1} = \arg\min_{u_2^{i,\cdot,k}\in\mathbb{R}^{N_2}} \sum_{j=1}^{N_2}\left(u_2^{i,j,k} - v_2^{i,j,k}\right)^2, \quad \text{s.t.} \quad B_{N_2} u_2^{i,\cdot,k} + C_{N_2,\varepsilon} \geq 0, \tag{56}$$

and

$$\left(u_3^{i,j,\cdot}\right)^{n+1} = \arg\min_{u_3^{i,j,\cdot}\in\mathbb{R}^{N_3}} \sum_{k=1}^{N_3}\left(u_3^{i,j,k} - v_3^{i,j,k}\right)^2, \quad \text{s.t.} \quad B_{N_3}u_3^{i,j,\cdot} + C_{N_3,\varepsilon} \geq 0, \tag{57}$$

respectively.

Based on previous discussion, the algorithm to compute $P_{\mathcal{B}_\varepsilon}(\mathbf{v})$ can be summarized as the following Algorithm 2.

Algorithm 2. Alternating directional algorithm for Projection $P_{\mathcal{B}_\varepsilon}(\mathbf{v})$.

Input: 3D vector field $\mathbf{v}$ and $\varepsilon > 0$.
(i). For fixed j, k, solving $(u_1^{\cdot,j,k})^{n+1}$ by (52) (see Algorithm 3 for details);
(ii). For fixed i, k, solving $(u_2^{i,\cdot,k})^{n+1}$ by (53) (see Algorithm 3 for details);
(iii). For fixed i, j, solving $(u_3^{i,j,\cdot})^{n+1}$ by (54) (see Algorithm 3 for details).
Return: $P_{\mathcal{B}_\varepsilon}(\mathbf{v}) = \mathbf{u} \triangleq (u_1, u_2, u_3)^T$.

Subproblems (i)–(iii) are quadratic programming problems of the following form:

$$\bar{X} = \arg\min_{X\in\mathbb{R}^N} \frac{1}{2}X^T H X + f^T X, \quad \text{s.t. } AX \geq b, \tag{58}$$

where $A = \mathbb{M}_{M\times N}(\mathbb{R})$, $b = \mathbb{M}_{M\times 1}(\mathbb{R})$, $f = \mathbb{M}_{N\times 1}(\mathbb{R})$, $H = 2I_N$.

Concerning the numerical implementation of (58), we apply customized Proximal Point Algorithm (PPA) (see [44–47] for details) to general convex optimization problem:

$$X = \arg\min\{\theta(X) : AX \geq b\}, \tag{59}$$

where $\theta(X)$ is a convex function.

The Lagrange function of associated (59) is

$$L(X,\lambda) = \theta(X) - \lambda^T(AX - b) \qquad (X,\lambda) \in \mathbb{R}^N \times \mathbb{R}^M. \tag{60}$$

For given (X^k, λ^k), PPA produces a pair of (X^{k+1}, λ^{k+1}) by

$$\lambda^{k+1} = \left[\arg\max_{\lambda\in\mathbb{R}^M}\left\{L\big(X^k,\lambda\big) - \frac{s}{2}\big\|\lambda - \lambda^k\big\|^2\right\}\right]_+, \tag{61}$$

and

$$X^{k+1} = \arg\min_{X\in\mathbb{R}^N}\left\{L\big(X, 2\lambda^{k+1} - \lambda^k\big) + \frac{r}{2}\big\|X - X^k\big\|^2\right\}. \tag{62}$$

Note that for any $M\times 1$ vector $K = (K_1, K_2, \cdots, K_M)^T$, $[K]_+ = ([K_1]_+, [K_2]_+, \cdots, [K_M]_+)^T$ is defined by

$$[K_i]_+ = \begin{cases} K_i & \text{if} \quad K_i > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{63}$$

If $r, s > 0$ and they satisfy $rs > \|A^T A\|$, the results of [44–46] show that the sequence $W^k = (X^k, \lambda^k)$ generated by PPA satisfies

$$\big\|W^{k+1} - W^*\big\|^2 \leq \big\|W^k - W^*\big\|^2 - \big\|W^{k+1} - W^k\big\|^2, \tag{64}$$

where $W^* = (X^*, \lambda^*)$ and X^* is the solution of (59).

Equation (64) ensures the convergence of PPA.

Apply PPA to the problem (58) by letting

$$\theta(X) = \frac{1}{2}X^T H X + f^T X. \tag{65}$$

Both (61) and (62) are unconstrained optimization problems, which implies the gradient of objective functions equal to 0 at optimal points. Based on this conclusion, the following solution can be easily obtained:

$$\begin{cases} \lambda^{k+1} = \left[\lambda^k - \dfrac{1}{s}(AX^k - b)\right]_+, \\ X_i^{k+1} = \dfrac{1}{2+r}\big(P_i^{k+1} - f_i + rX_i^k\big), \end{cases} \tag{66}$$

where $P^{k+1} = (P_1^{k+1}, P_2^{k+1}, \cdots, P_N^{k+1}) = (2\lambda^{k+1} - \lambda^k)^T A$ and $i = 1, 2, \cdots, N$.

Based on previous discussion, PPA for the quadratic programming problem (58) can be summarized as the following Algorithm 3.

Algorithm 3. PPA for quadratic programming problem (58).

Input: $r, s, A, b, f, \lambda^0, X^0$
Set $k = 0$, norm=1, accuracy=0.001 and $K = 50$.
While norm> accuracy and $k \le K$
do: Compute $\lambda^{k+1} = \left[\lambda^k - \frac{1}{s}(AX^k - b)\right]_+$;
Compute $P^{k+1} = (2\lambda^{k+1} - \lambda^k)^T A$;
Compute $X_i^{k+1} = \frac{1}{2+r}\big(P_i^{k+1} - f_i + rX_i^k\big)$ for $i = 1, 2, \cdots, N$;
Set norm=$\|X^k - X^{k+1}\|$ and $k = k + 1$.
EndWhile
Return: $\bar{X} = X^{K_0}$ for some $K_0 \le K$.

Remark 3.9. One can also use MATLAB code "x=quadrog (H,f,-A,-b)" to solve the quadratic programming problem (58). The numerical techniques for quadrog function in MATLAB can be found in Section 3 of [48]. However, quadrog function converges much slower than Algorithm 3. For example, considering the following quadratic programming problem:

$$\bar{X} = \arg\min_{X\in\mathbb{R}^3}\Big\{(x_1 - 1)^2 + (x_2 + 3)^2 + (x_3 + 4)^2 :\ A_3 X \ge b_3\Big\}, \tag{67}$$

where $A_3 = \begin{pmatrix} 0 & 0.5 & 0 \\ -0.5 & 0 & 0.5 \\ 0 & -0.5 & 0 \end{pmatrix}$, $b_3 = (-0.99, -0.99, -0.99)^T$.

Both MATLAB quadrog function and Algorithm 3 achieve the optimal solution $\bar{X} = (-0.51, -1.98, -2.49)^T$. However, it takes 0.321356 s for MATLAB quadrog function to converge to the optimal solution while Algorithm 3 costs only 0.018727 s. In fact, in numerical tests which will be introduced in Section 4, we have noticed that Algorithm 1 converges much slower when Algorithm 3 is replaced by quadrog function in MATLAB.

4. Numerical test

In this section, two numerical tests are performed to show the efficiency of the proposed Algorithm 1. The aim of Test 1 is to show the sensitivity of Algorithm 1 with respect to the parameters μ, Θ, ε and α. In addition, for the purpose of showing the efficiency of Algorithm 1, in Test 2, four numerical tests are performed using a 3D synthetic image pair (see Test 2 for details) and three 3D medical image pairs, where the 3D medical images in Test 2 can be downloaded from [49–51]. All the numerical tests are performed under Windows 7 and MATLAB R2012b with Intel core i7-6700 CPU @3.40 GHz and 8 GB memory. In these four tests, the proposed algorithm is compared with Diffeomorphic Log Demons Image Registration (DLDIR for short) algorithm [19, 52], MIRT3D algorithm [53], 3D conformal model (One can refer to Appendix B or [54] for details) and MATLAB image registration function "imregister", where the open MATLAB code of DLDIR algorithm and MIRT3D algorithm can be downloaded from [52] and [53], respectively. For quantitative comparison, we compare the following four indexes:

– Relative sum of squared difference (Re-SSD for short) which is defined by

$$\text{Re-SSD}(T, D, \mathbf{u}) = \frac{\text{SSD}(T(\mathbf{x}+\mathbf{u}), D)}{\text{SSD}(T, D)}, \tag{68}$$

where $\text{SSD}(T, D) = \frac{1}{2}\sum_{i,j,k}(T_{i,j,k} - D_{i,j,k})^2$.

– Mesh folding number (MFN for short) which is defined by

$$\text{MFN}(\mathbf{u}) = \sharp(\det J(\mathbf{u}) \leq 0) \qquad \forall \mathbf{x} \in \Omega, \tag{69}$$

where $\det J(\mathbf{u}) = \begin{vmatrix} 1+\frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} & \frac{\partial u_1}{\partial x_3} \\ \frac{\partial u_2}{\partial x_1} & 1+\frac{\partial u_2}{\partial x_2} & \frac{\partial u_2}{\partial x_3} \\ \frac{\partial u_3}{\partial x_1} & \frac{\partial u_3}{\partial x_2} & 1+\frac{\partial u_3}{\partial x_3} \end{vmatrix}$ and for any set A, $\sharp(A)$ denotes the number of elements in A.

– Structural similarity (SSIM for short) [55] which is defined by

$$\text{SSIM}(T, D) = \frac{(2\mu_T\mu_D + C_1)(2\sigma_T\sigma_D + C_2)(\sigma_{TD} + C_3)}{(\mu_T^2 + \mu_D^2 + C_1)(\sigma_T^2 + \sigma_D^2 + C_2)(\sigma_T\sigma_D + C_3)}, \tag{70}$$

where $\bar{N} = N_1N_2N_3$, $\mu_T = \frac{1}{\bar{N}}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\sum_{k=1}^{N_3} T_{i,j,k}$, $\mu_D = \frac{1}{\bar{N}}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\sum_{k=1}^{N_3} D_{i,j,k}$, $\sigma_T = \left(\frac{1}{\bar{N}-1}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\sum_{k=1}^{N_3}(T_{i,j,k} - \mu_T)^2\right)^{\frac{1}{2}}$, $\sigma_D = \left(\frac{1}{\bar{N}-1}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\sum_{k=1}^{N_3}(D_{i,j,k} - \mu_D)^2\right)^{\frac{1}{2}}$, $\sigma_{TD} = \frac{1}{\bar{N}-1}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\sum_{k=1}^{N_3}(T_{i,j,k} - \mu_T)(D_{i,j,k} - \mu_D)$. Note that $C_1 = (K_1L)^2$, $C_2 = (K_2L)^2$ and in this paper, we set $L = 255$, $K_1 = 0.06$, $K_2 = 0.08$ and $C_3 = 0.5C_2$.

– Mutual information (MI for short) [56] which is defined by

$$\text{MI}(T, D) = \sum_{i,j=1}^{255} p^{T,D}(i,j)\log\frac{p^{T,D}(i,j)}{p^T(i)p^D(j)}, \tag{71}$$

where $p^{T,D}(i,j)$ is the joint intensity distribution of image $T(\cdot)$ and $D(\cdot)$; $p^T(i)$ and $p^D(j)$ are the marginal intensity distribution of image $T(\cdot)$ and $D(\cdot)$, respectively.

Remark 4.1. Theoretically, $\text{MFN}(\mathbf{u}) = 0$ implies no mesh folding. In addition, the smaller Re-SSD and the larger SSIM and MI, the better registration result.

Test 1. The aim of this test is to show the sensitivity of Algorithm 1 with respect to the parameters μ, Θ, ε and α. For this purpose, numerical tests are performed on a 3D synthetic image pair (data I) with size $80 \times 80 \times 80$. In data I, the floating image $T(\cdot)$ and target image $D(\cdot)$ are defined by

$$T(\mathbf{x}) = 255\chi_{\Omega_1}(\mathbf{x}), D(\mathbf{x}) = 255\chi_{\Omega_2}(\mathbf{x}),$$

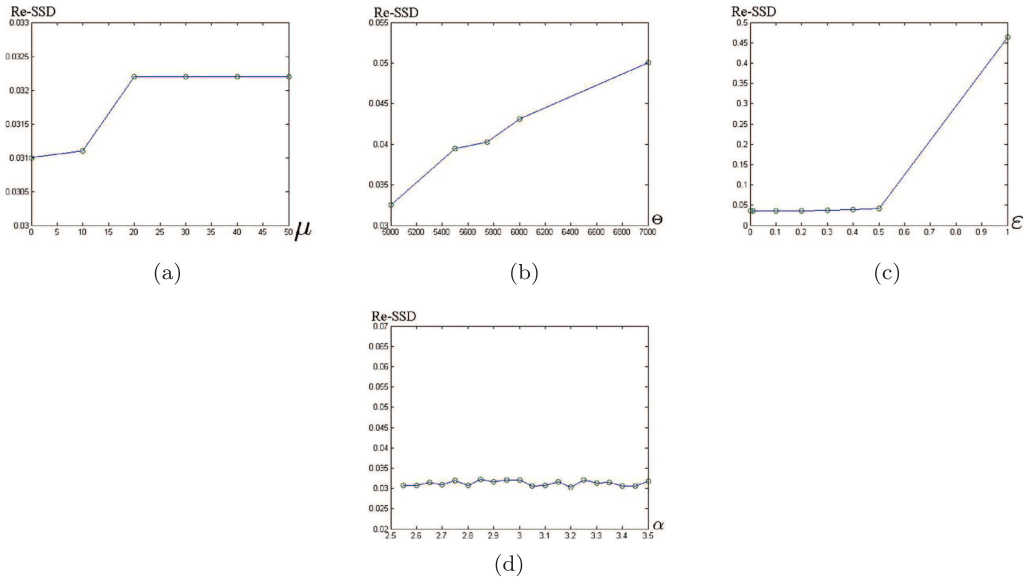

Figure 2. Sensitivity of $\mu, \Theta, \varepsilon, \alpha$. (a) μ/Re-SSD. (b) Θ/Re-SSD. (c) ε/Re-SSD. (d) α/Re-SSD.

Table 1. Quantitative comparison between registration results of five different 3D registration algorithms.

Data	Algorithm	Re-SSD (%)	MFN	SSIM	MI	CPU
	MIRT3D	75.5	2756	0.2243	0.1008	135.6 s
	Log-Demons	91.26	1821	0.0606	0.0483	83.4 s
Data II	MATLAB	99.16	0	0.0159	0.16	3.6 s
	3D conformal model	9.18	0	0.9381	0.4481	412.1 s
	Algorithm 1	3.36	0	0.9916	0.6689	651.3 s
	MIRT3D	18.54	0	0.9796	1.8372	509.3 s
	Log-Demons	5.5	111	0.9937	1.6774	3.57 h
Data III	MATLAB	83.85	0	0.885	1.0723	84.9 s
	3D conformal model	16.32	0	0.978	1.8377	2631.7 s
	Algorithm 1	2.43	0	0.9928	1.8885	1494.3 s
	MIRT3D	25.23	234	0.9695	1.3405	326.2 s
Data IV	Log-Demons	5.67	215	0.9912	1.62	2.65 h
	MATLAB	90.7	0	0.9196	1.0681	33.1 s
	3D conformal model	12.26	0	0.9895	1.5164	3516.6 s
	Algorithm 1	5.17	0	0.9933	1.67	1552.8 s
	MIRT3D	15.86	1023	0.9623	1.3592	68.8 s
Data V	Log-Demons	18.95	0	0.9613	1.3509	1983.2 s
	MATLAB	28.79	0	0.9339	1.1678	14.7 s
	3D conformal model	19.05	0	0.9653	1.3583	1613.6 s
	Algorithm 1	9.68	0	0.9701	1.4271	738.1 s

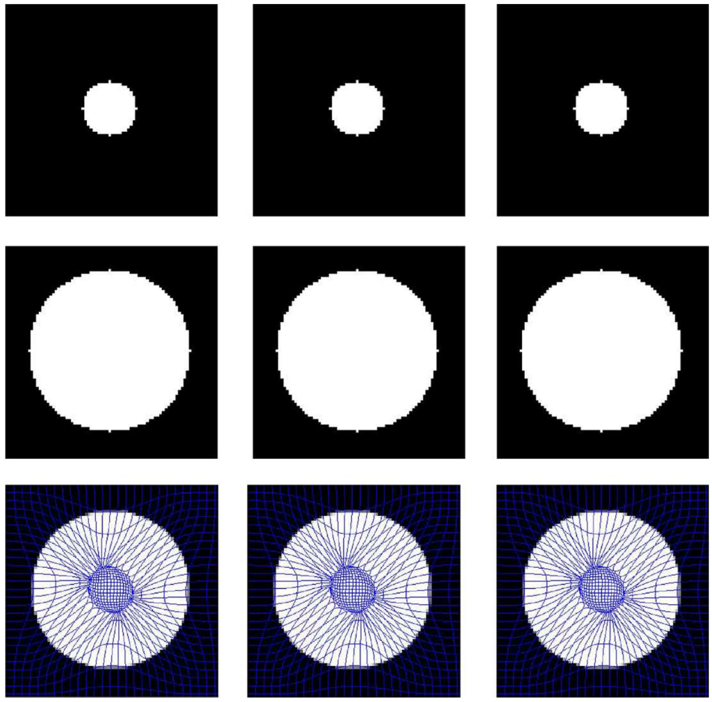

FIGURE 3. Registered images for data II; Row I is the middlemost slice of $T(\cdot)$ on x_1, x_2, x_3 direction, respectively; Row II is the middlemost slice of $D(\cdot)$ on x_1, x_2, x_3 direction, respectively; Row III is the middlemost slice of $T\circ\mathbf{h}(\cdot)$ produced by Algorithm 1 on x_1, x_2, x_3 direction, respectively.

where $\Omega_1 = \left\{\mathbf{x} = (x_1, x_2, x_3)^T : \left(\frac{x_1-40}{20}\right)^2 + \left(\frac{x_2-40}{20}\right)^2 + \left(\frac{x_3-40}{20}\right)^2 \le 1\right\}$, $\Omega_2 = \left\{\mathbf{x} = (x_1, x_2, x_3)^T : \left(\frac{x_1-40}{30}\right)^2 + \left(\frac{x_2-40}{20}\right)^2 + \left(\frac{x_3-40}{20}\right)^2 \le 1\right\}$. Here and in what follows, $\chi_\Omega(\mathbf{x}) = 1$ if $\mathbf{x} \in \Omega$ and $\chi_\Omega(\mathbf{x}) = 0$ if $\mathbf{x}$ is not in Ω.

By giving different $\mu, \Theta, \varepsilon, \alpha$, we use Algorithm 1 to perform the registration on data I. Final registration results are recorded for each given μ, Θ, ε and α. One can see Figure 2 for details.

By Figure 2a, the change of μ nearly has no obvious effect on final Re-SSD. This is mainly caused by the fact that $R(\mathbf{u})$ plays a dominant role in cost functional $J(\mathbf{u})$, because Θ is a very large number. For this reason, the role of $\int_\Omega |\nabla^\alpha \mathbf{u}|^2\,\mathrm{d}\mathbf{x}$ is very limited compared with $R(\mathbf{u})$. By Figure 2b, we know that Re-SSD is also decreased when Θ becomes smaller. However, though we can get a much smaller Re-SSD when Θ is smaller, mesh folding occurs. Hence, Θ is greater than 5000 in our numerical tests. By Figure 2c, we know that smaller ε leads to smaller Re-SSD. In addition, Re-SSD changes only a little when $\varepsilon \in (0, 0.5)$. Especially, there is no obvious change on Re-SSD as $\varepsilon \in (0, 0.01)$. In fact, $B_{\varepsilon_1} \supset B_{\varepsilon_2} (\varepsilon_1 < \varepsilon_2)$. This implies that $J(\mathbf{u})$ achieves a much smaller minimum on B_{ε_1}. Therefore, our numerical results coincide with qualitative analysis. At last, we emphasize that ε should not be too large (In our test, $\varepsilon = 0.01$). As it is shown on Figure 2c, final Re-SSD $= 46.2\%$ as $\varepsilon = 1$. This is a very bad registration result. Concerning the sensitivity of α, one can notice from Figure 2d that the proposed algorithm is not sensitive to the fractional order α. This is mainly caused by the fact that Θ is a large number which weakens the role of fractional regularization $\int_\Omega |\nabla^\alpha \mathbf{u}|^2\,\mathrm{d}\mathbf{x}$.

Remark 4.2. In Theorem 3.5, a sufficient technical condition $\mu \ge 3CML$ (a not small number) is given to ensure that $\mathcal{F}$ is a monotone operator. By Figure 2a, one can notice that the proposed Algorithm 1 is not

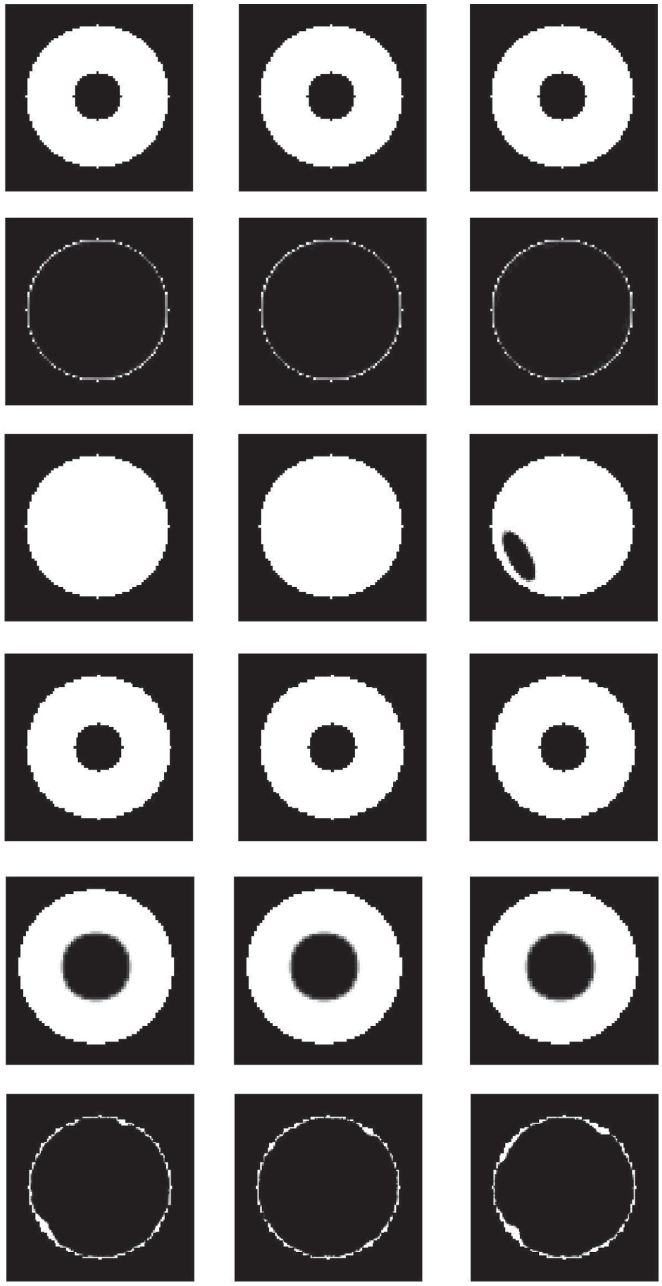

FIGURE 4. Difference for data II; Row I is difference $|T(\cdot) - D(\cdot)|$ for the middlemost slice on x_1, x_2, x_3 direction, respectively; Row II is $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ of Algorithm 1 for the middlemost slice on x_1, x_2, x_3 direction, respectively; Row III is $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ of MATLAB for the middlemost slice on x_1, x_2, x_3 direction, respectively; Row IV is $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ of MIRT3D for the middlemost slice on x_1, x_2, x_3 direction, respectively; Row V is $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ of Log-Demons for the middlemost slice on x_1, x_2, x_3 direction, respectively; Row VI is $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ of 3D conformal model for the middlemost slice on x_1, x_2, x_3 direction, respectively.

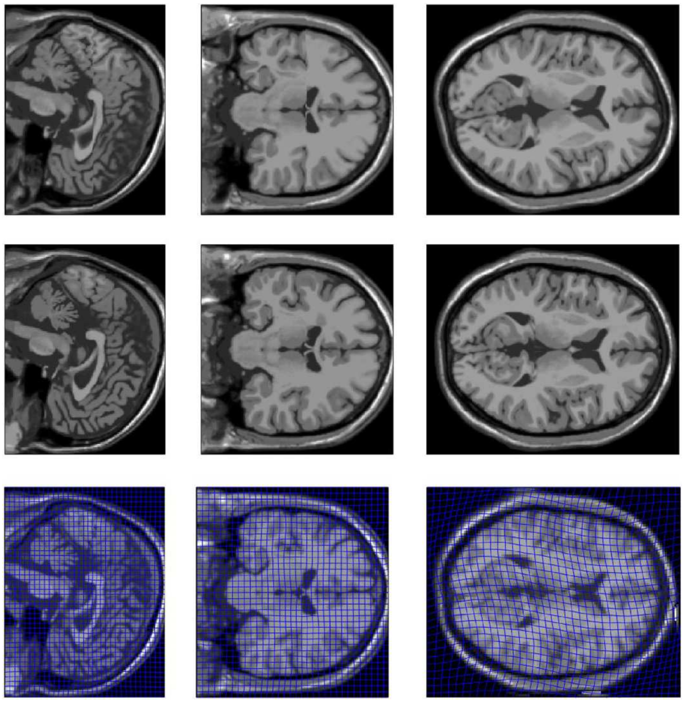

FIGURE 5. Registered images for data III; Row I is the middlemost slice of $T(\cdot)$ on x_1, x_2, x_3 direction, respectively; Row II is the middlemost slice of $D(\cdot)$ on x_1, x_2, x_3 direction, respectively; Row III is the middlemost slice of $T \circ \mathbf{h}(\cdot)$ produced by Algorithm 1 on x_1, x_2, x_3 direction, respectively.

sensitive to the parameter μ. Therefore, in practice, one can also set a small μ which may not satisfy the technical condition $\mu \geq 3CML$ but may help to get a more accurate result.

Test 2. In this test, we perform registration on a synthetic 3D image pair (data II) and three 3D medical pairs: brain image (data III), liver (data IV) image, lung image (data V). Data II is of the size $80 \times 80 \times 80$. The floating image $T(\cdot)$ and target image $D(\cdot)$ of data II are defined by

$$T(\mathbf{x}) = 255\chi_{\bar{\Omega}_1}(\mathbf{x}), D(\mathbf{x}) = 255\chi_{\bar{\Omega}_2}(\mathbf{x}),$$

where $\bar{\Omega}_1 = \left\{\mathbf{x} = (x_1, x_2, x_3)^T : \left(\frac{x_1-40}{10}\right)^2 + \left(\frac{x_2-40}{10}\right)^2 + \left(\frac{x_3-40}{10}\right)^2 \leq 1\right\}$, $\bar{\Omega}_2 = \left\{\mathbf{x} = (x_1, x_2, x_3)^T : \left(\frac{x_1-40}{30}\right)^2 + \left(\frac{x_2-40}{30}\right)^2 + \left(\frac{x_3-40}{30}\right)^2 \leq 1\right\}$.

Data III is a 3D brain image pair of size $181 \times 217 \times 165$. The original data can be downloaded from [49]. Data IV is a 3D liver image pairs of size $512 \times 512 \times 122$. The original data can be downloaded from [50]. Data V is a 3D lung image pairs of size $128 \times 128 \times 64$. The original data can be downloaded from [51]. To verify the efficiency of Algorithm 1, we make a quantitative comparison among five different image registration algorithms: DLDIR algorithm, MIRT3D algorithm, MATLAB, 3D conformal model [54] and proposed Algorithm 1. For this purpose, these five algorithms are used to perform the registration on data II–V, respectively. The registration results are listed on Table 1. Note that here the key parameters in the Algorithm 1 are set as $\alpha = 2.85$, $\Theta = 7500$, $\mu = 5 \times 10^{-6}$ and the parameters used in DLDIR, MIRT3D and MATLAB are default values of the code. In addition, to show the 3D data clearly, for each 3D data pair and its registration result, the middlemost

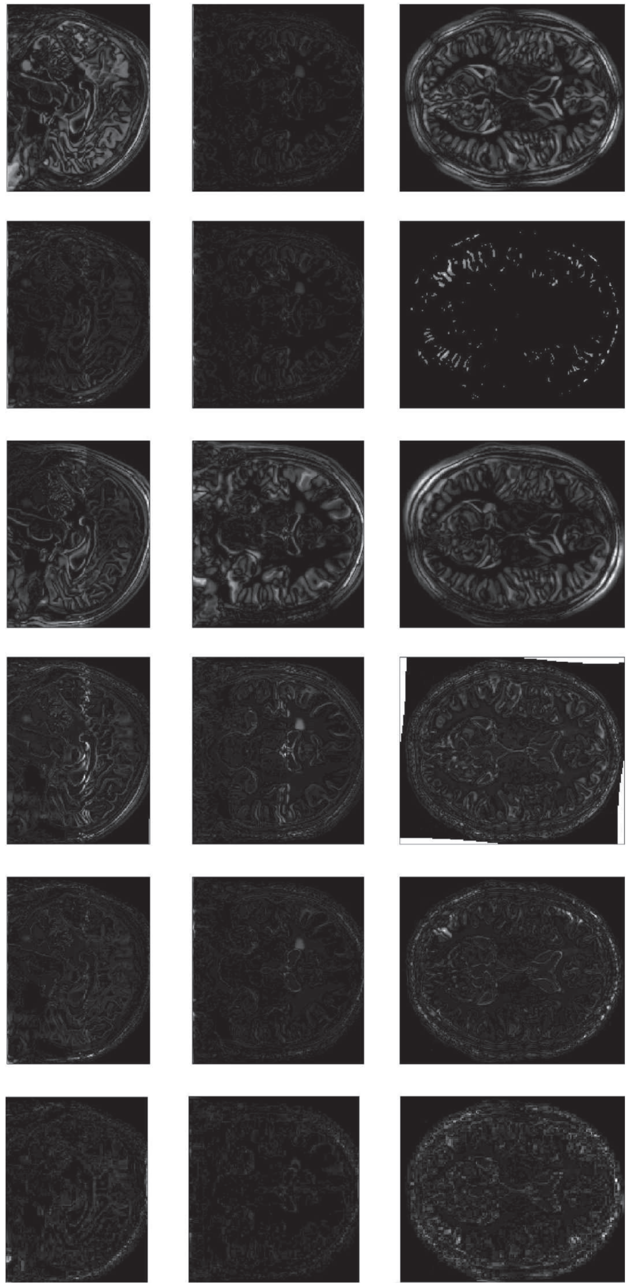

FIGURE 6. Difference for data III; Row I is difference $|T(\cdot) - D(\cdot)|$ for the middlemost slice on x_1, x_2, x_3 direction, respectively; Row II is $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ of Algorithm 1 for the middlemost slice on x_1, x_2, x_3 direction, respectively; Row III is $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ of MATLAB for the middlemost slice on x_1, x_2, x_3 direction, respectively; Row IV is $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ of MIRT3D for the middlemost slice on x_1, x_2, x_3 direction, respectively; Row V is $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ of Log-Demons for the middlemost slice on x_1, x_2, x_3 direction, respectively. Row VI is $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ of 3D conformal model for the middlemost slice on x_1, x_2, x_3 direction, respectively.

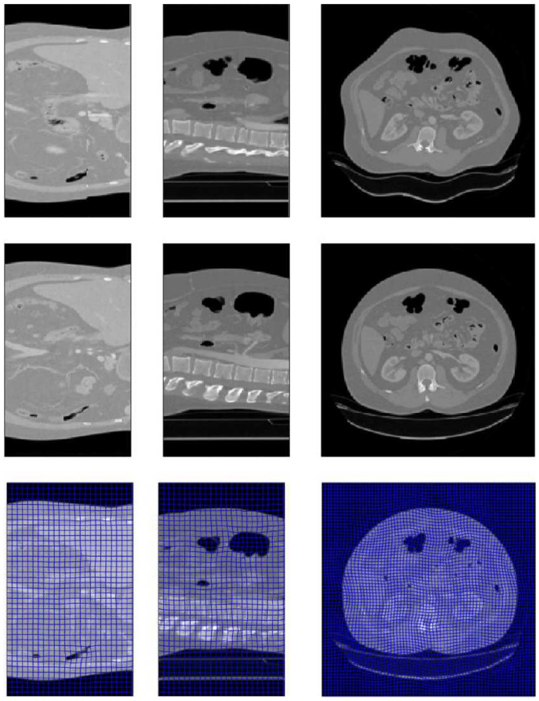

FIGURE 7. Registered images for data IV; Row I is the middlemost slice of $T(\cdot)$ on x_1, x_2, x_3 direction, respectively; Row II is the middlemost slice of $D(\cdot)$ on x_1, x_2, x_3 direction, respectively; Row III is the middlemost slice of $T \circ \mathbf{h}(\cdot)$ produced by Algorithm 1 on x_1, x_2, x_3 direction, respectively.

slices on three directions are displayed. One can see Figures 3–10 for details. Moreover, in order to evaluate the complexity of five algorithms, the CPU consumption of each different resolution for data II–V is also recorded, one can see Table 2 for details.

Firstly, by Figures 3 and 4, we can notice that Algorithm 1 and 3D conformal model perform much better than the other three algorithms. Besides, by Table 1, one can notice that Algorithm 1 achieves the smallest Re-SSD among the five compared algorithms. This shows that Algorithm 1 is competitive to deal with this kind of synthetic 3D images. Moreover, there is large deformation in data II, this implies that the proposed algorithm is efficient for 3D larger deformation image registration.

Secondly, by Figure 5, we can notice that $T \circ \mathbf{h}(\cdot)$ produced by Algorithm 1 looks nearly the same to the target image $D(\cdot)$. Furthermore, the final difference $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ shown on Figure 6 indicates that MIRT3D and MATLAB perform worse than Algorithm 1, 3D conformal model and Log-Demons. Moreover, by Table 1, one can notice that though Log-Demons achieves a satisfactory Re-SSD, mesh folding occurs. In fact, Algorithm 1 can achieve a much smaller Re-SSD than 2.43% listed on Table 1, if Θ is a very small number. In this case, mesh folding also occurs. In this paper, we care more about mesh folding elimination, rather than smaller Re-SSD. Therefore, we don't list these registration results with mesh folding for Algorithm 1.

Thirdly, by Figures 7 and 8, one can see that Algorithm 1 gives a satisfactory registration result for data IV. Furthermore, by Table 1, one can also notice that mesh folding occurs for MIRT3D and Log-Demons. Though no mesh folding for MATLAB, Re-SSD is 90.7%, which is a very bad registration result. Therefore, Re-SSD and its associated MFN for data IV confirm that Algorithm 1 is the best as it has non-physical mesh folding.

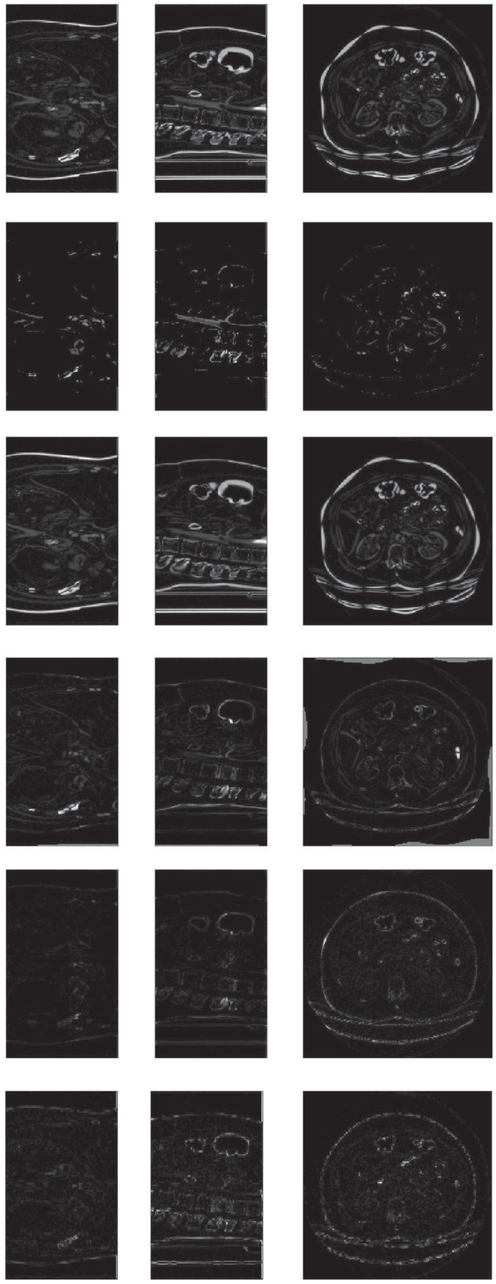

FIGURE 8. Difference for data IV; Row I is difference $|T(\cdot) - D(\cdot)|$ for the middlemost slice on x_1, x_2, x_3 direction, respectively; Row II is $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ of Algorithm 1 for the middlemost slice on x_1, x_2, x_3 direction, respectively; Row III is $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ of MATLAB for the middlemost slice on x_1, x_2, x_3 direction, respectively; Row IV is $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ of MIRT3D for the middlemost slice on x_1, x_2, x_3 direction, respectively; Row V is $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ of Log-Demons for the middlemost slice on x_1, x_2, x_3 direction, respectively. Row VI is $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ of 3D conformal model for the middlemost slice on x_1, x_2, x_3 direction, respectively.

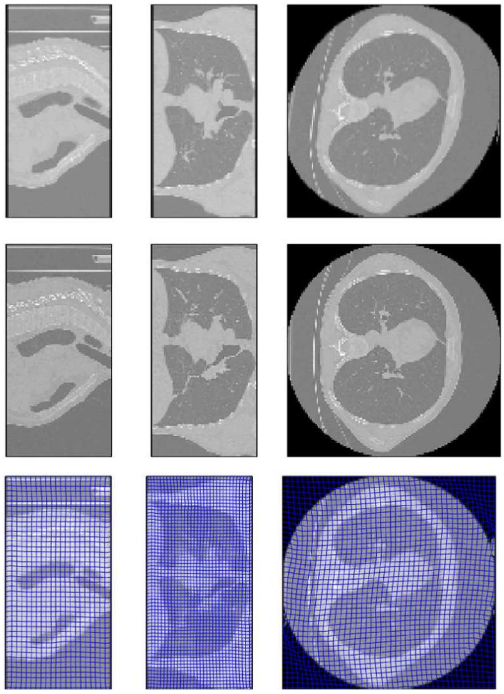

FIGURE 9. Registered images for data V; Row I is the middlemost slice of $T(\cdot)$ on x_1, x_2, x_3 direction, respectively; Row II is the middlemost slice of $D(\cdot)$ on x_1, x_2, x_3 direction, respectively; Row III is the middlemost slice of $T \circ \mathbf{h}(\cdot)$ produced by Algorithm 1 on x_1, x_2, x_3 direction, respectively.

TABLE 2. CPU consumption for different resolutions.

Data	Resolutions	MIRT3D	Log-Demons	MATLAB	3D conformal model	Algorithm 1
Data II	$80 \times 80 \times 80$	135.6 s	83.4 s	3.6 s	412.1 s	651.3 s
	$40 \times 40 \times 40$	13.5 s	35.3 s	1.3 s	75.1 s	83.1 s
	$20 \times 20 \times 20$	1.9 s	32.1 s	0.81 s	8.3 s	10.5 s
Data III	$181 \times 217 \times 165$	509.3 s	3.57 h	84.9 s	2631.7 s	1494.3 s
	$90 \times 108 \times 82$	44.8 s	1328.7 s	36.7 s	472.9 s	712.1 s
	$45 \times 54 \times 41$	16.1 s	24.3 s	6.2 s	85.6 s	82.3 s
Data IV	$512 \times 512 \times 122$	326.2 s	2.65 h	33.1 s	3516.6 s	1552.8 s
	$256 \times 256 \times 61$	48.1 s	1.23 h	8.1 s	416.2 s	316.9 s
	$128 \times 128 \times 30$	11.7 s	619.4 s	2.3 s	129.5 s	71.3 s
Data V	$128 \times 128 \times 64$	68.8 s	1983.2 s	14.7 s	1613.6 s	738.1 s
	$64 \times 64 \times 32$	8.0 s	125.6 s	4.1 s	231.5	131.7 s
	$32 \times 32 \times 16$	0.58 s	12.8 s	2.6 s	32.1 s	61.9 s

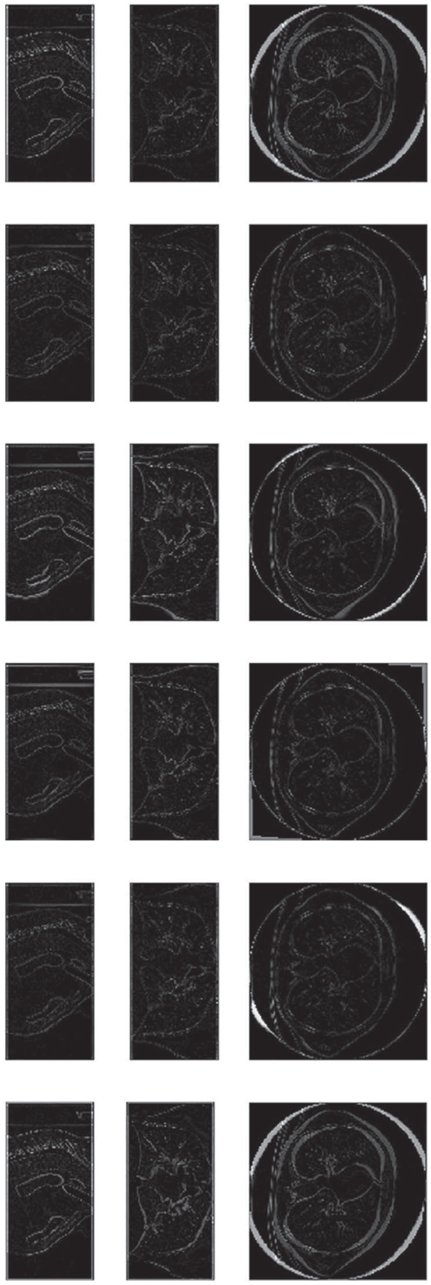

FIGURE 10. Difference for data V; Row I is difference $|T(\cdot) - D(\cdot)|$ for the middlemost slice on x_1, x_2, x_3 direction, respectively; Row II is $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ of Algorithm 1 for the middlemost slice on x_1, x_2, x_3 direction, respectively; Row III is $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ of MATLAB for the middlemost slice on x_1, x_2, x_3 direction, respectively; Row IV is $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ of MIRT3D for the middlemost slice on x_1, x_2, x_3 direction, respectively; Row V is $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ of Log-Demons for the middlemost slice on x_1, x_2, x_3 direction, respectively. Row VI is $|T \circ \mathbf{h}(\cdot) - D(\cdot)|$ of 3D conformal model for the middlemost slice on x_1, x_2, x_3 direction, respectively.

At last, by Figures 9, 10 and Table 1, we see that only MIRT3D produces mesh folding. Algorithm 1 achieves the smallest Re-SSD without mesh folding. This confirms that Algorithm 1 is competitive among the five compared algorithms.

5. Conclusion

In this paper, a 3D diffeomorphic image registration model is presented. In addition, a relaxed model is also proposed and the existence of its solution is proved. Moreover, a 3D projection image registration algorithm is also proposed. Numerical tests show that the proposed algorithm is effective. For further research, we may focus on the acceleration of proposed algorithm (*e.g.*, multi grid method) and landmark based 3D image registration model without mesh folding.

Appendix A. Finite difference discretization of $\mathcal{F}(\mathbf{u})$

For the convenience of description, here and in what follows, we define the gradient of a 3D scalar matrix $\mathbf{F} = (f_{i,j,k})_{(N_1+2)\times(N_2+2)\times(N_3+2)}$ by

$$[\mathbf{f}^x, \mathbf{f}^y, \mathbf{f}^z] \triangleq \nabla \mathbf{F}, \tag{A.1}$$

where $\mathbf{f}^x = (f^x_{i,j,k})_{(N_1+2)\times(N_2+2)\times(N_3+2)}$, $\mathbf{f}^y = (f^y_{i,j,k})_{(N_1+2)\times(N_2+2)\times(N_3+2)}$, $\mathbf{f}^z = (f^z_{i,j,k})_{(N_1+2)\times(N_2+2)\times(N_3+2)}$ are all 3D matrixes with

$$f^x_{i,j,k} = \begin{cases} 0, & i = 0, N_1+1 \\ \frac{f_{i+1,j,k} - f_{i-1,j,k}}{2}, & 1 \le i \le N_1 \end{cases},$$

$$f^y_{i,j,k} = \begin{cases} 0, & j = 0, N_2+1 \\ \frac{f_{i,j+1,k} - f_{i,j-1,k}}{2}, & 1 \le j \le N_2 \end{cases},$$

$$f^z_{i,j,k} = \begin{cases} 0, & k = 0, N_3+1 \\ \frac{f_{i,j,k+1} - f_{i,j,k-1}}{2}, & 1 \le k \le N_3 \end{cases}.$$

Let $\mathbf{h}_{i,j,k} = \mathbf{X}_{i,j,k} + \mathbf{u}(\mathbf{X}_{i,j,k})$, $\mathbf{r} = (T(\mathbf{h}_{i,j,k}))_{(N_1+2)\times(N_2+2)\times(N_3+2)}$, $d_{i,j,k} = D(\mathbf{X}_{i,j,k})$ and

$$[\mathbf{T}_x, \mathbf{T}_y, \mathbf{T}_z] = \nabla \mathbf{r}, \tag{A.2}$$

where $\mathbf{T}_x = (T^x_{i,j,k})_{(N_1+2)\times(N_2+2)\times(N_3+2)}$, $\mathbf{T}_y = (T^y_{i,j,k})_{(N_1+2)\times(N_2+2)\times(N_3+2)}$, $\mathbf{T}_z = (T^z_{i,j,k})_{(N_1+2)\times(N_2+2)\times(N_3+2)}$.

Furthermore, we define

$$\mathbf{\Psi}_x = \left(\psi^x_{i,j,k}\right)_{(N_1+2)\times(N_2+2)\times(N_3+2)}, \tag{A.3}$$

$$\mathbf{\Psi}_y = \left(\psi^y_{i,j,k}\right)_{(N_1+2)\times(N_2+2)\times(N_3+2)}, \tag{A.4}$$

$$\mathbf{\Psi}_z = \left(\psi^z_{i,j,k}\right)_{(N_1+2)\times(N_2+2)\times(N_3+2)}, \tag{A.5}$$

where $\psi^x_{i,j,k} = (T(\mathbf{h}_{i,j,k}) - d_{i,j,k})T^x_{i,j,k}$, $\psi^y_{i,j,k} = (T(\mathbf{h}_{i,j,k}) - d_{i,j,k})T^y_{i,j,k}$, $\psi^z_{i,j,k} = (T(\mathbf{h}_{i,j,k}) - d_{i,j,k})T^z_{i,j,k}$.

Therefore,

$$\{[T(\mathbf{x} + \mathbf{u}(\mathbf{x})) - D(\mathbf{x})]\nabla T(\mathbf{x} + \mathbf{u}(\mathbf{x}))\}_{i,j,k} = \left(\psi^x_{i,j,k}, \psi^y_{i,j,k}, \psi^z_{i,j,k}\right)^T, \tag{A.6}$$

where $i = 1, 2, \cdots, N_1$; $j = 1, 2, \cdots, N_2$; $k = 1, 2, \cdots, N_3$.

In a similar way, let

$$\mathbf{U}_x = \left(u^x_{i,j,k}\right)_{(N_1+2)\times(N_2+2)\times(N_3+2)}, \tag{A.7}$$

$$\mathbf{U}_y = \left(u^y_{i,j,k}\right)_{(N_1+2)\times(N_2+2)\times(N_3+2)}, \tag{A.8}$$

$$\mathbf{U}_z = \left(u^z_{i,j,k}\right)_{(N_1+2)\times(N_2+2)\times(N_3+2)}, \tag{A.9}$$

where $u^x_{i,j,k} = u_1(\mathbf{X}_{i,j,k})$, $u^y_{i,j,k} = u_2(\mathbf{X}_{i,j,k})$, $u^z_{i,j,k} = u_3(\mathbf{X}_{i,j,k})$.

Define

$$[\chi^x, \chi^y, \chi^z] = \nabla\mathbf{U}_x, \quad [\kappa^x, \kappa^y, \kappa^z] = \nabla\mathbf{U}_y, \quad [\omega^x, \omega^y, \omega^z] = \nabla\mathbf{U}_z, \tag{A.10}$$

and

$$\mathbf{E} = \chi^x - \kappa^y = (e_{i,j,k})_{(N_1+2)\times(N_2+2)\times(N_3+2)}, \tag{A.11}$$

$$\mathbf{F} = \chi^x - \omega^z = (f_{i,j,k})_{(N_1+2)\times(N_2+2)\times(N_3+2)}, \tag{A.12}$$

$$\mathbf{G} = \kappa^y - \omega^z = (g_{i,j,k})_{(N_1+2)\times(N_2+2)\times(N_3+2)}, \tag{A.13}$$

$$\mathbf{H} = \chi^y + \kappa^x = (H_{i,j,k})_{(N_1+2)\times(N_2+2)\times(N_3+2)}, \tag{A.14}$$

$$\mathbf{I} = \chi^z + \omega^x = (I_{i,j,k})_{(N_1+2)\times(N_2+2)\times(N_3+2)}, \tag{A.15}$$

$$\mathbf{J} = \kappa^z + \omega^y = (J_{i,j,k})_{(N_1+2)\times(N_2+2)\times(N_3+2)}, \tag{A.16}$$

then $e_{i,j,k} = \left(\frac{\partial u_1}{\partial x_1} - \frac{\partial u_2}{\partial x_2}\right)_{i,j,k}$, $f_{i,j,k} = \left(\frac{\partial u_1}{\partial x_1} - \frac{\partial u_3}{\partial x_3}\right)_{i,j,k}$, $g_{i,j,k} = \left(\frac{\partial u_2}{\partial x_2} - \frac{\partial u_3}{\partial x_3}\right)_{i,j,k}$, $H_{i,j,k} = \left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right)_{i,j,k}$, $I_{i,j,k} = \left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}\right)_{i,j,k}$, $J_{i,j,k} = \left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}\right)_{i,j,k}$.

Furthermore, we define

$$[\mathbf{K}_x, \mathbf{K}_y, \mathbf{K}_z] = \nabla\mathbf{E}, \quad [\mathbf{L}_x, \mathbf{L}_y, \mathbf{L}_z] = \nabla\mathbf{F}, \quad [\mathbf{M}_x, \mathbf{M}_y, \mathbf{M}_z] = \nabla\mathbf{G}, \tag{A.17}$$

$$[\mathbf{N}_x, \mathbf{N}_y, \mathbf{N}_z] = \nabla\mathbf{H}, \quad [\mathbf{O}_x, \mathbf{O}_y, \mathbf{O}_z] = \nabla\mathbf{I}, \quad [\mathbf{P}_x, \mathbf{P}_y, \mathbf{P}_z] = \nabla\mathbf{J}, \tag{A.18}$$

and $\mathbf{Q} = -\mathbf{K}_x - \mathbf{L}_x - \mathbf{N}_y - \mathbf{O}_z = (Q_{i,j,k})_{(N_1+2)\times(N_2+2)\times(N_3+2)}$, $\mathbf{R} = \mathbf{K}_y - \mathbf{M}_y - \mathbf{N}_x - \mathbf{P}_z = (R_{i,j,k})_{(N_1+2)\times(N_2+2)\times(N_3+2)}$, $\mathbf{S} = \mathbf{L}_z - \mathbf{M}_z - \mathbf{O}_x - \mathbf{P}_y = (S_{i,j,k})_{(N_1+2)\times(N_2+2)\times(N_3+2)}$, then

$$Q_{i,j,k} = \{\varepsilon_1(\mathbf{u}) + \eta_1(\mathbf{u})\}_{i,j,k}, R_{i,j,k} = \{\varepsilon_2(\mathbf{u}) + \eta_2(\mathbf{u})\}_{i,j,k}, S_{i,j,k} = \{\varepsilon_3(\mathbf{u}) + \eta_3(\mathbf{u})\}_{i,j,k}, \tag{A.19}$$

where $i = 1, 2, \cdots, N_1$; $j = 1, 2, \cdots, N_2$; $k = 1, 2, \cdots, N_3$.

Now, we focus on the discretization of fractional-order term $\mathrm{div}^{\alpha*}(\nabla^\alpha\mathbf{u})$. Using Grunwald approximation [10, 23, 38, 43], $\frac{\partial^\alpha f(\mathbf{x})}{\partial x_i^\alpha}$, $\frac{\partial^{\alpha*} f(\mathbf{x})}{\partial x_i^{\alpha*}}$ $(i = 1, 2, 3)$ can be approximated by:

$$\frac{\partial^\alpha f(\mathbf{X}_{p,q,r})}{\partial x_i^\alpha} = \delta^\alpha_{i-} f(\mathbf{X}_{p,q,r}) + O(h_i), \quad \frac{\partial^{\alpha*} f(\mathbf{X}_{p,q,r})}{\partial x_i^{\alpha*}} = \delta^\alpha_{i+} f(\mathbf{X}_{p,q,r}) + O(h_i) \tag{A.20}$$

where $\delta^\alpha_{1-} f(\mathbf{X}_{p,q,r}) = \frac{1}{h_1^\alpha}\sum_{l=1}^{p+1}\rho_l^{(\alpha)} f_{p-l+1,q,r}$, $\delta^\alpha_{1+} f(\mathbf{X}_{p,q,r}) = \frac{1}{h_1^\alpha}\sum_{l=1}^{N_1-p+2}\rho_l^{(\alpha)} f_{p+l-1,q,r}$, $\delta^\alpha_{2-} f(\mathbf{X}_{p,q,r}) = \frac{1}{h_2^\alpha}\sum_{l=1}^{q+1}\rho_l^{(\alpha)} f_{p,q-l+1,r}$, $\delta^\alpha_{2+} f(\mathbf{X}_{p,q,r}) = \frac{1}{h_2^\alpha}\sum_{l=1}^{N_2-q+2}\rho_l^{(\alpha)} f_{p,q+l-1,r}$, $\delta^\alpha_{3-} f(\mathbf{X}_{p,q,r}) = \frac{1}{h_3^\alpha}\sum_{l=1}^{r+1}\rho_l^{(\alpha)} f_{p,q,r-l+1}$, $\delta^\alpha_{3+} f(\mathbf{X}_{p,q,r}) = \frac{1}{h_3^\alpha}\sum_{l=1}^{N_3-r+2}\rho_l^{(\alpha)} f_{p,q,r+l-1}$ and $\rho_0^{(\alpha)} = 1$, $\rho_l^{(\alpha)} = (1 - \frac{1+\alpha}{l})\rho_{l-1}^{(\alpha)}$.

Here and in what follows, we use the notations:

$$f_{p,q,r} = f(\mathbf{X}_{p,q,r}). \tag{A.21}$$

Let $U_{q,r} = (f_{1,q,r}, f_{2,q,r}, \cdots, f_{N_1,q,r})^T$, then it follows from (A.20) that

$$\frac{\partial^\alpha U_{q,r}}{\partial x_1^\alpha} \approx B_{N_x} U_{q,r}, \quad \frac{\partial^{\alpha*} U_{q,r}}{\partial x_1^{\alpha*}} \approx B^T_{N_x} U_{q,r}, \tag{A.22}$$

where $\frac{\partial^\alpha U_{q,r}}{\partial x_1^\alpha} = \left(\frac{\partial^\alpha f_{1,q,r}}{\partial x_1^\alpha}, \frac{\partial^\alpha f_{2,q,r}}{\partial x_1^\alpha}, \cdots, \frac{\partial^\alpha f_{N_1,q,r}}{\partial x_1^\alpha}\right)^T$, $\frac{\partial^{\alpha*} U_{q,r}}{\partial x_1^{\alpha*}} = \left(\frac{\partial^{\alpha*} f_{1,q,r}}{\partial x_1^{\alpha*}}, \frac{\partial^{\alpha*} f_{2,q,r}}{\partial x_1^{\alpha*}}, \cdots, \frac{\partial^{\alpha*} f_{N_1,q,r}}{\partial x_1^{\alpha*}}\right)^T$ and $B_{N_x} = \frac{1}{h_1^\alpha}\begin{pmatrix} \rho_1^{(\alpha)} & \rho_0^{(\alpha)} & 0 & \cdots & 0 & 0 \\ \rho_2^{(\alpha)} & \rho_1^{(\alpha)} & \rho_0^{(\alpha)} & \cdots & 0 & 0 \\ \rho_3^{(\alpha)} & \rho_2^{(\alpha)} & \rho_1^{(\alpha)} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho_{N_1-1}^{(\alpha)} & \rho_{N_1-2}^{(\alpha)} & \rho_{N_1-3}^{(\alpha)} & \cdots & \rho_1^{(\alpha)} & \rho_0^{(\alpha)} \\ \rho_{N_1}^{(\alpha)} & \rho_{N_1-1}^{(\alpha)} & \rho_{N_1-2}^{(\alpha)} & \cdots & \rho_2^{(\alpha)} & \rho_1^{(\alpha)} \end{pmatrix}$.

That is,

$$\frac{\partial^{\alpha*}}{\partial x_1^{\alpha*}}\left(\frac{\partial^\alpha U_{q,r}}{\partial x_1^\alpha}\right) = B_{N_x}^T B_{N_x} U_{q,r} \triangleq A_{N_x} U_{q,r}. \tag{A.23}$$

Similarly, there holds

$$\frac{\partial^{\alpha*}}{\partial x_2^{\alpha*}}\left(\frac{\partial^\alpha V_{p,r}}{\partial x_2^\alpha}\right) = B_{N_y}^T B_{N_y} V_{p,r} \triangleq A_{N_y} V_{p,r}, \tag{A.24}$$

and

$$\frac{\partial^{\alpha*}}{\partial x_3^{\alpha*}}\left(\frac{\partial^\alpha W_{p,q}}{\partial x_3^\alpha}\right) = B_{N_z}^T B_{N_z} W_{p,q} \triangleq A_{N_z} W_{p,q}, \tag{A.25}$$

where $V_{p,r} = (f_{p,1,r}, f_{p,2,r}, \cdots, f_{p,N_2,r})^T$, $W_{p,q} = (f_{p,q,1}, f_{p,q,2}, \cdots, f_{p,q,N_3})^T$.

Based on (A.23), (A.24) and (A.25), we define

$$\begin{aligned} &\Pi^{\mathbf{x}} = \left(\Pi_{i,j,k}^x\right)_{(N_1+2)\times(N_2+2)\times(N_3+2)}, \Pi^{\mathbf{y}} = \left(\Pi_{i,j,k}^y\right)_{(N_1+2)\times(N_2+2)\times(N_3+2)}, \\ &\Pi^{\mathbf{z}} = \left(\Pi_{i,j,k}^z\right)_{(N_1+2)\times(N_2+2)\times(N_3+2)}, \end{aligned} \tag{A.26}$$

where

$$\begin{aligned} \Pi_{i,j,k}^x &= \left\{\frac{\partial^{\alpha*}}{\partial x_1^{\alpha*}}\left(\frac{\partial^\alpha u_1}{\partial x_1^\alpha}\right) + \frac{\partial^{\alpha*}}{\partial x_2^{\alpha*}}\left(\frac{\partial^\alpha u_1}{\partial x_2^\alpha}\right) + \frac{\partial^{\alpha*}}{\partial x_3^{\alpha*}}\left(\frac{\partial^\alpha u_1}{\partial x_3^\alpha}\right)\right\}_{i,j,k}, \\ \Pi_{i,j,k}^y &= \left\{\frac{\partial^{\alpha*}}{\partial x_1^{\alpha*}}\left(\frac{\partial^\alpha u_2}{\partial x_1^\alpha}\right) + \frac{\partial^{\alpha*}}{\partial x_2^{\alpha*}}\left(\frac{\partial^\alpha u_2}{\partial x_2^\alpha}\right) + \frac{\partial^{\alpha*}}{\partial x_3^{\alpha*}}\left(\frac{\partial^\alpha u_2}{\partial x_3^\alpha}\right)\right\}_{i,j,k}, \\ \Pi_{i,j,k}^z &= \left\{\frac{\partial^{\alpha*}}{\partial x_1^{\alpha*}}\left(\frac{\partial^\alpha u_3}{\partial x_1^\alpha}\right) + \frac{\partial^{\alpha*}}{\partial x_2^{\alpha*}}\left(\frac{\partial^\alpha u_3}{\partial x_2^\alpha}\right) + \frac{\partial^{\alpha*}}{\partial x_3^{\alpha*}}\left(\frac{\partial^\alpha u_3}{\partial x_3^\alpha}\right)\right\}_{i,j,k}. \end{aligned}$$

Define $\Upsilon^x = (\gamma_{i,j,k}^x)_{(N_1+2)\times(N_2+2)\times(N_3+2)}$, $\Upsilon^y = (\gamma_{i,j,k}^y)_{(N_1+2)\times(N_2+2)\times(N_3+2)}$, $\Upsilon^z = (\gamma_{i,j,k}^z)_{(N_1+2)\times(N_2+2)\times(N_3+2)}$, where $\Upsilon^x = \mathbf{\Psi}_x + \mu\Pi^{\mathbf{x}} + \Theta\mathbf{Q}$, $\Upsilon^y = \mathbf{\Psi}_y + \mu\Pi^{\mathbf{y}} + \Theta\mathbf{R}$, $\Upsilon^z = \mathbf{\Psi}_z + \mu\Pi^{\mathbf{z}} + \Theta\mathbf{S}$.

It follows from (A.6), (A.19) and (A.26) that

$$\left(\gamma_{i,j,k}^x, \gamma_{i,j,k}^y, \gamma_{i,j,k}^z\right)^T = (\mathcal{F}(\mathbf{u}))_{i,j,k}, \tag{A.27}$$

where $i = 1, 2, \cdots, N_1$; $j = 1, 2, \cdots, N_2$; $k = 1, 2, \cdots, N_3$. Note that here the value of elements in matrixes whose formulas are not given in Appendix A is zero, *i.e.*, $(\mathcal{F}(\mathbf{u}))_{0,j,k} = 0$ for $j = 1, 2, \cdots, N_2$; $k = 1, 2, \cdots, N_3$.

Appendix B. 3D conformal image registration model

By [22], we know that for any nD deformation $\mathbf{h}$, a condition for mapping $\mathbf{h}$ to be conformal is

$$\nabla\mathbf{h}\nabla\mathbf{h}^T = \nabla\mathbf{h}^T\nabla\mathbf{h} = \frac{1}{n}\|\nabla\mathbf{h}\|_F^2\mathbf{I} = (\det(\nabla\mathbf{h}))^{\frac{2}{n}}\mathbf{I}, \tag{B.1}$$

where $\nabla\mathbf{h}$ is Jacobian matrix of $\mathbf{h}$, $\|\nabla\mathbf{h}\|_F^2 = \sum_{i,j=1}^{n}\left(\frac{\partial h_i}{\partial x_j}\right)^2$ and $\mathbf{I}$ is an $n \times n$ identity matrix.

Based on (B.1), we define a bounded 3D conformal set

$$\begin{aligned}\mathcal{A} = \Big\{\mathbf{u} = (u_1, u_2, u_3)^T \in [H_0^\alpha(\Omega)]^3 : \det(\nabla\mathbf{h}) > 0, \|\nabla h_1(\mathbf{x})\|_2^2 = \|\nabla h_2(\mathbf{x})\|_2^2 = \|\nabla h_3(\mathbf{x})\|_2^2,\\ \nabla h_i(\mathbf{x})\cdot\nabla h_j(\mathbf{x}) = 0, \qquad \text{for } i \neq j \text{ and } \forall\mathbf{x}\in\Omega\Big\},\end{aligned} \tag{B.2}$$

where $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), h_2(\mathbf{x}), h_3(\mathbf{x}))^T = \mathbf{x} + \mathbf{u}(\mathbf{x})$.

By restricting the solution into a 3D conformal set $\mathcal{A}$, 3D conformal image registration model is formulated by

$$\mathbf{u} = \arg\min_{\mathbf{u}\in\mathcal{A}} S(\mathbf{u}) + \mu R(\mathbf{u}), \tag{B.3}$$

where $\mu > 0$ and $R(\mathbf{u}) = \int_\Omega |\nabla^\alpha\mathbf{u}|^2\,\mathrm{d}\mathbf{x}(\alpha > 2.5)$. Concerning the details of 3D conformal model (B.3) and its numerical implementation, one can see [54] for details.

Remark B.1. Though 3D conformal (B.3) is not competitive to the proposed algorithm, one can notice that (B.3) based 3D multi scale approach achieves a much more accurate result than the proposed algorithm in this article with much more CPU consumption.

Appendix C. Asymptotic analysis of $R(\mathbf{u}^\Theta)$ as $\Theta \to +\infty$

For this purpose, we give a simple asymptotic analysis [57] for the solution of (13) as $\Theta \to +\infty$. Note that $\mathbf{u} \equiv \mathbf{0}$ is the unique minimizer of $J(\mathbf{u})$ if $T(\mathbf{x}) = D(\mathbf{x})$ for any $\mathbf{x}\in\Omega$. There is no need for registration in this case. Therefore, $\mathbf{u} \equiv \mathbf{0}$ is ruled out from the solution set $\mathcal{B}_\varepsilon$. That is, we search for a solution in the function set $\mathcal{D} = \{\mathbf{u}\in\mathcal{B}_\varepsilon : \mathbf{u} \neq \mathbf{0}\}$.

For the convenience of asymptotic analysis, let's introduce some notations.

First, we define $\mathrm{supp}(T) = \{\mathbf{x} : T(\mathbf{x}) \neq 0\}$ and assume $\mathrm{supp}(T) \subset \Omega$. Furthermore, we define $T(\mathbf{x}) = 0$ if $\mathbf{x}\in\mathbb{R}^3\setminus\Omega$, then $T : \mathbb{R}^3 \to \mathbb{R}$ is continuous on $\mathbb{R}^3\setminus\triangle_T$. Similarly, we define $\mathbf{u}(\mathbf{x}) = 0$ if $\mathbf{x}\in\mathbb{R}^3\setminus\Omega$. Then $\mathbf{u}\in\mathcal{B}_\varepsilon$ implies $\mathbf{u}\in[H_0^\alpha(\mathbb{R}^3)]^3$ and

$$J(\mathbf{u}) = \int_{\mathbb{R}^3}(T(\mathbf{x}+\mathbf{u}(\mathbf{x})) - D(\mathbf{x}))^2\,\mathrm{d}\mathbf{x} + \mu R_1(\mathbf{u}) + \Theta R(\mathbf{u}), \tag{C.1}$$

here $R_1(\mathbf{u}) = \int_{\mathbb{R}^3}|\nabla^\alpha\mathbf{u}|^2\,\mathrm{d}\mathbf{x}$ and $R(\mathbf{u}) = \sum_{1\le i<j\le 3}\int_{\mathbb{R}^3}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)^2 + \left(\frac{\partial u_i}{\partial x_i}-\frac{\partial u_j}{\partial x_j}\right)^2\mathrm{d}\mathbf{x}$.

Based on these notations, we define $\mathbf{u}^\eta(\mathbf{x}) = \eta^{-\frac{3}{2}}\mathbf{u}(\frac{\mathbf{x}}{\eta})$ for some $\eta > 0$, then $\mathbf{u}(\mathbf{x}) = \eta^{\frac{3}{2}}\mathbf{u}^\eta(\eta\mathbf{x})$. By simple calculation, one can notice that

$$J(\mathbf{u}) = \frac{1}{\eta^3}\left[\int_{\mathbb{R}^3}\left(T\left(\frac{\mathbf{x}}{\eta}+\eta^{\frac{3}{2}}\mathbf{u}^\eta(\mathbf{x})\right) - D\left(\frac{\mathbf{x}}{\eta}\right)\right)^2\mathrm{d}\mathbf{x} + \mu\eta^5 R_1(\mathbf{u}^\eta) + \Theta\eta^5 R(\mathbf{u}^\eta)\right], \tag{C.2}$$

which indicates that (C.1) is equivalent to minimize the following functional:

$$J^\eta(\mathbf{u}^\eta) = \int_{\mathbb{R}^3}\left(T\left(\frac{\mathbf{x}}{\eta}+\eta^{\frac{3}{2}}\mathbf{u}^\eta(\mathbf{x})\right) - D\left(\frac{\mathbf{x}}{\eta}\right)\right)^2\mathrm{d}\mathbf{x} + \mu\eta^5 R_1(\mathbf{u}^\eta) + \Theta\eta^5 R(\mathbf{u}^\eta), \tag{C.3}$$

under the constraint $\mathbf{u}^\eta \in \mathcal{D}$. Note that $\mathbf{u} \in \mathcal{D}$ implies $\|\mathbf{u}\|_{[C^1(\Omega)]^3} \leq C\|\mathbf{u}\|_{[H_0^\alpha(\Omega)]^3} < +\infty$ because of the fact that $H_0^\alpha(\Omega) \hookrightarrow C^1(\Omega)$ ([39], Thm. 4.58). That is, $\mathbf{u}(\mathbf{x}) = \eta^{\frac{3}{2}}\mathbf{u}^\eta(\eta\mathbf{x})$ is bounded. Therefore, equation (13) is equivalent to minimize the functional (C.3) on the constraint $\mathbf{u}^\eta \in \mathcal{G} = \{\mathbf{u}^\eta \in \mathcal{D} : \eta^{\frac{3}{2}}\mathbf{u}^\eta(\eta\mathbf{x}) < +\infty\}$.

Based on these discussions, one can notice that the first term of (C.3) converges to 0 as $\eta \to 0$. By giving the assumption $\frac{\mu}{\Theta} \ll 1$ and letting $\eta = \Theta^{-\frac{1}{5}}$ and $\eta \to 0(\Theta \to +\infty)$, one can notice that $J^\eta(\mathbf{u}^\eta)/(\Theta\eta^5)$ converges to the functional $R(\mathbf{u})$. That is, $\mathbf{u}^\eta$ is a minimizing sequence of the functional $R(\mathbf{u})$ as $\eta \to 0$.

Let $\mathbf{u}^\eta \in \mathcal{G}$ be a minimizer of $J^\eta(\mathbf{u}^\eta)$, then $J^\eta(\mathbf{u}^\eta)$ is bounded. There exists a subsequence which is still labeled by $\mathbf{u}^\eta$ converges to some $\mathbf{u}^\infty \in [C^1(\Omega)]^3$ (similar to (19)) as $\eta \to 0$. By the similar discussion of (25), one can note that $R(\mathbf{u}^\eta) \to R(\mathbf{u}^\infty)$.

We claim that

$$\mathbf{u}^\infty = \arg\min_{\mathbf{u}\in\mathcal{G}} R(\mathbf{u}). \tag{C.4}$$

Note that here $\min_{\mathbf{u}\in\mathcal{G}} R(\mathbf{u}) = 0$.

Suppose (C.4) does not hold and let $\bar{\mathbf{u}} = \arg\min\limits_{\mathbf{u}\in\mathcal{G}} R(\mathbf{u})$, then there exists a $c > 0$ such that $R(\mathbf{u}^\infty) > c$. Define $\bar{\mathbf{u}}^\eta = \eta^{\frac{3}{2}}\bar{\mathbf{u}}(\frac{\mathbf{x}}{\eta})$, then one can use the similar way of (C.2) and (C.3) to know that

$$J^\eta(\bar{\mathbf{u}}^\eta) = \int_{\mathbb{R}^3}\left(T\left(\frac{\mathbf{x}}{\eta} + \eta^{\frac{3}{2}}\bar{\mathbf{u}}^\eta(\mathbf{x})\right) - D\left(\frac{\mathbf{x}}{\eta}\right)\right)^2 \mathrm{d}\mathbf{x} + \mu\eta^5 R_1(\bar{\mathbf{u}}^\eta) + \Theta\eta^5 R(\bar{\mathbf{u}}^\eta). \tag{C.5}$$

By letting $\eta = \Theta^{-\frac{1}{5}}$ and $\eta \to 0(\Theta \to +\infty)$, one can notice that $J^\eta(\bar{\mathbf{u}}^\eta) \to R(\bar{\mathbf{u}}) = 0$, which contradicts the fact $\mathbf{u}^\eta$ is the minimizer of $J^\eta(\mathbf{u}^\eta)$ as $\eta \to 0$ (*i.e*, $J^\eta(\bar{\mathbf{u}}^\eta) \to R(\bar{\mathbf{u}}) = 0 < c = R(\mathbf{u}^\infty) \leq J^\eta(\mathbf{u}^\eta)$, where $\mathbf{u}^\eta$ is a minimizer of functional $J^\eta(\mathbf{u}^\eta)$).

By claim (C.4), we know that $\lim_{\Theta\to+\infty} R(\mathbf{u}^\Theta) = 0$.

Acknowledgements. This work was supported by National Key Research and Development Program of China (No. 2020YFA0714200), National Nature Science Foundation of China under grant Nos. 11931012, 11871386, 11901443. Thanks also to the two referees for their very helpful and nice comments on the improvement of our paper.

References

[1] M.A. Viergever, J.B.A. Maintz, S. Kleinc, K. Murphy, M. Staring and J.P.W. Pluim, A survey of medical image registration-under review. *Med. Image Anal.* **33** (2016) 140–144.

[2] J. Salvi, C. Matabosch, D. Fofi and J. Forest, A review of recent range image registration methods with accuracy evaluation. *Image Vision Comput.* **25** (2007) 578–596.

[3] T. Makela, P. Clarysse, O. Sipila, N. Pauna, Q. Pham, T. Katila and I. Magnin, A review of cardiac image registration methods. *IEEE Trans. Med. Imag.* **21** (2002) 1011–1021.

[4] N. Chumchob, K. Chen and C. Loeza, A fourth-order variational image registration model and its fast multigrid algorithm. *SIAM J. Multiscale Model. Simul.* **9** (2011) 89–128.

[5] D. Ferreira, E. Ribeiro and C. Barcelos, A variational approach to non-rigid image registration with Bregman divergences and multiple features. Pattern Recognition **77** (2018) 237–247.

[6] Z. Nie and X. Yang, Deformable image registration using functions of bounded deformation. *IEEE Trans. Med. Imag.* **38** (2019) 1488–1500.

[7] G. Hermosillo, C. Hotel and O. Faugeras, Variational methods for multimodal image matching. *Int. J. Comput. Vis.* **50** (2002) 329–343.

[8] L.C. Evans, Partial Differential Equations. American Mathematical Society, Rhode Island (2010).

[9] H. Han, A variational model with fractional-order regularization term arising in registration of diffusion tensor image. *Inverse Probl. Imaging* **12** (2018) 1263–1291.

[10] H. Han and Z. Wang, An alternating direction implicit scheme of a fractional-order diffusion tensor image registration model. *Appl. Math. Comput.* **256** (2019) 105–118.

[11] H. Han and H. Zhou, A variational problem arising in registration of diffusion tensor images. *Acta Math. Sci.* **37** (2017) 539–554.

[12] P. Dupuis, U. Grenander and M.I. Miller, Variational problems on flows of diffeomorphisms for image matching. *Quart. Appl. Math.* **56** (1998) 587–600.

[13] A. Budhiraja, P. Dupuis and V. Maroulas, Large deviations for stochastic flows of diffeomorphisms. *Bernoulli* **16** (2010) 234–257.

[14] M.F. Beg, M.I. Miller, A. Trouv and L. Younes, Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *Int. J. Comput. Vis.* **61** (2005) 139–157.

[15] M. Bruveris, F. Gay-Balmaz, D.D. Holm and T.S. Ratiu, The momentum map representation of images. *J. Nonlinear Sci.* **21** (2011) 115–150.

[16] J. Li, Y. Shi, G. Tran, I. Dinov, D. Wang and A. Toga, Fast local trust region for diffusion tensor registration using exact reorientation and regularization. *IEEE Trans. Med. Imag.* **33** (2014) 1–43.

[17] H. Lombaert, L. Grady, X. Pennec, N. Ayache and F. Cheriet, Spectral log-demons: diffeomorphic image registration with very large deformations. *Int. J. Comput. Vis.* **107** (2014) 254–271.

[18] J.P. Thirion, Image matching as a diffusion process: an analogy with Maxwell's demons. *Med. Image Anal.* **2** (1998) 243–260.

[19] T. Vercauteren, X. Pennecb, A. Perchanta and N. Ayacheb, Diffeomorphic demons: efficient non-parametric image registration. *Neuroimage* **45** (2009) 61–72.

[20] N. Chumchob, Vectorial total variation-based regularization for variational image registration. *IEEE Trans. Image Process.* **22** (2013) 4551–4559.

[21] V. Vishnevskiy, T. Gass, G. Szekely, C. Tanner and O. Goksel, Isotropic total variation regularization of displacements in parametric image registration. *IEEE Trans. Med. Imag.* **36** (2017) 385–395.

[22] D. Zhang and K. Chen, 3D orientation-preserving variational models for accurate image registration. *SIAM J. Imaging Sci.* **13** (2016) 1653–1691.

[23] W. Tian, H. Zhou and W. Deng, A class of second order difference approximation for solving space fractional diffusion equations. *Math. Comput.* **84** (2015) 1703–1727.

[24] Y. Wang, L.M. Lui, X. Gu, K.M. Hayashi, T.F. Chan, A.W. Toga, P.M. Thompson and S.T. Yau, Brain surface conformal parameterization using riemann surface structure. *IEEE Trans. Med. Imag.* **26** (2007) 853–865.

[25] L.M. Lui, T.W. Wong, W. Zeng, X. Gu, P.M. Thompson, T.F. Chan and S.T. Yau, Optimization of surface registrations using beltrami holomorphic flow. *J. Sci. Comput.* **50** (2012) 557–585.

[26] Y. Wang, L.M. Lui, T.F. Chan and P.M. Thompson, Optimization of brain conformal mapping with landmarks. *Med. Image Comput. Comput. Assisted Intervention (MICCAI)* **2** (2005) 675–683.

[27] P.T. Choi, K.C. Lam and L.M. Lui, FLASH: fast landmark aligned spherical harmonic parameterization for genus-0 closed brain surfaces. *SIAM J. Imaging Sci.* **8** (2015) 67–94.

[28] L.M. Lui and C. Wen, Geometric registration of High-Genus surfaces. *SIAM J. Imaging Sci.* **7** (2014) 337–365.

[29] K.C. Lam and L.M. Lui, Optimized quasiconformal parameterization with user-defined area distortions. *Commun. Math. Sci.* **15** (2017) 2027–2054.

[30] L.M. Lui, S. Thiruvenkadam, Y. Wang, P.M. Thompson and T.F. Chan, Optimized conformal surface registration with shape-based landmark matching. *SIAM J. Imaging Sci.* **3** (2010) 52–78.

[31] H. Han and Z. Wang, A diffeomorphic image registration model with fractional-order regularization and Cauchy–Riemann constraint. *SIAM J. Imaging Sci.* **13** (2020) 1240–1271.

[32] H. Han and A. Wang, A fast multi grid algorithm for 2D diffeomorphic image registration model. *J Comput. Appl. Math.* **394** (2021) 113576.

[33] H. Han, Z. Wang and Y. Zhang, Multiscale approach for Two-Dimensional diffeomorphic image registration. *SIAM J. Multiscale Model. Simul.* **19** (2021) 1538–1572.

[34] F.F. Wu and C.A. Desoer, Global inverse function theorem. *IEEE Trans. Circuit Theory* **19** (1972) 199–201.

[35] J. Zhang, K. Chen and B. Yu, A novel high-order functional based image registration model with inequality constraint. *Comput. Math. Appl.* **72** (2016) 2887–2899.

[36] Y.T. Lee, K.C. Lam and L.M. Lui, Landmark-matching transformation with large deformation via n-dimensional quasi-conformal maps. *J. Sci. Comput.* **67** (2016) 926–954.

[37] G.P. Paillé and P. Poulin, As-conformal-as-possible discrete volumetric mapping. *Comput. Graphics* **36** (2012) 427–433.

[38] J. Zhang and K. Chen, Variational image registration by a total fractional-order variation model. *J. Comput. Phys.* **293** (2015) 442–461.

[39] F. Demengel, G. Demengel and R. Erne, Functional Spaces for the Theory of Elliptic Partial Differential Equations. Springer, London (2012).

[40] J.B. Conway, A Course in Functional Analysis. Springer, New York (2007).

[41] C. Kanzow and Y. Shehu, Strong convergence of a double projection-type method for monotone variational inequalities in Hilbert spaces. *J. Fixed Point Theory Appl.* **20** (2018) 51–72.

[42] T.L. Friesz, Finite Dimensional Variational Inequalities and Nash Equilibria. Springer US, New York (2010).

[43] H. Han, A tensor voting based fractional-order image denoising model and its numerical algorithm. *Appl. Numer. Math.* **145** (2019) 133–144.

[44] B. He, X. Yuan and W. Zhang, A customized proximal point algorithm for convex minimization with linear constraints. *Comput. Optim. Appl.* **56** (2013) 559–572.

[45] G. Gu, B. He and X. Yuan, Customized proximal point algorithms for linearly constrained convex minimization and saddle-point problems: a unified approach. *Comput. Optim. Appl.* **59** (2014) 135–161.

[46] B. He and X. Yuan, Convergence analysis of primal-dual algorithms for a saddle point problem: from contraction perspective. *SIAM J. Imaging Sci.* **5** (2012) 119–149.

[47] A. Chambolle and T. Pock, A first-order primal-dual algorithms for convex problem with applications to imaging. *J. Math. Imaging Vison* **40** (2011) 120–145.
[48] N. Gould and P.L. Toint, Preprocessing for quadratic programming. *Math. Program.* **100** (2004) 95–132.
[49] BrainWeb: Simulated Brain Database, https://brainweb.bic.mni.mcgill.ca/brainweb/ (2013).
[50] 3D-IRCADb-02, https://www.ircad.fr/research/3d-ircadb-02/ (2019).
[51] Imlab-uiip/lung-segmentation-3d, https://github.com/imlab-uiip/lung-segmentation-3d/find/master (2019).
[52] H. Lombaert, Diffeomorphic Log Demons Image Registration (2012).
[53] Medical Image Registration Toolbox, http://www.bme.ogi.edu/~myron/matlab/MIRT/ (2016).
[54] H. Han, Z. Wang and Y. Zhang, Multiscale approach for Three-Dimensional conformal image registration. *SIAM J. Imaging Sci.* **15** (2022) 1431–1468.
[55] Z. Wang, A. Bovik, H. Sheikh and E. Simoncelli, Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13** (2004) 600–612.
[56] E. Agostino, F. Maes, D. Vandermeulen and P. Suetens, Viscous fluid model for multimodal non-rigid image registration using mutual information. *Med. Image Anal.* **7** (2003) 565–575.
[57] W. Bao, Y. Cai and X. Ruan, Groud states of Bose-Einstein condensates with higher order interaction. *Phys. D* **386, 387** (2019) 38–48.
[58] M. Burger, J. Modersitzki and L. Ruthotto, A hyperelastic regularization energy for image registration. *SIAM J. Sci. Comput.* **35** (2013) 132–148.
[59] M. Droske and M. Rumpf, A variational approach to nonrigid morphological image registration. *SIAM J. Appl. Math.* **64** (2004) 668–687.
[60] J. Zhang, K. Chen and B. Yu, An improved discontinuity-preserving image registration model and its fast algorithm. *Appl. Math. Model.* **40** (2016) 10740–10759.