

MULTISCALE APPROACH FOR TWO-DIMENSIONAL DIFFEOMORPHIC IMAGE REGISTRATION*

HUAN HAN[†], ZHENGPING WANG[‡], AND YIMIN ZHANG[§]

Abstract. In a beautiful paper, Modin, Nachman, and Rondi [*Adv. Math.*, 346 (2019), pp. 1009–1066] introduced a hierarchical image registration model based on the large deformation diffeomorphic metric mapping (LDDMM) framework. Unfortunately, no numerical tests are performed to show the efficiency of this multiscale approach. The LDDMM image registration framework is essentially a variational problem with differential equation constraints and the structure of the cost functional is very complex. Therefore, it's necessary and meaningful to introduce some other analogous multiscale approaches with a much simpler cost functional. Motivated by the work of Modin, Nachman, and Rondi, we construct a multiscale image registration approach for the two-dimensional diffeomorphic image registration model in [H. Han and Z. Wang, *SIAM J. Imaging Sci.*, 13 (2020), pp. 1240–1271]. This approach achieves a smooth minimizer for the cost functional without regularization. This result is completely different from most published models which only achieve minimizers of the cost functional with some regularization. The existence of solutions for the multiscale approach and the convergence of the multiscale approach are proved. In addition, a multigrid based multiscale diffeomorphic image registration algorithm is presented. Moreover, numerical tests are also performed to show that the proposed multiscale approach achieves a satisfactory image registration result without mesh folding.

Key words. multiscale, diffeomorphism, multigrid, image registration, algorithm

AMS subject classifications. 65M12, 68U10, 94A08, 65K10

DOI. 10.1137/20M1383987

1. Introduction. Let Ω be an open bounded domain on \mathbb{R}^2 with Lipschitz boundary. $T, D : \Omega \rightarrow \mathbb{R}$ are two images defined on Ω , where T and D are called the floating image and the target image, respectively. The goal of diffeomorphic image registration is to seek an optimal diffeomorphism $\mathbf{h} : \Omega \rightarrow \Omega$ between T and D such that $T \circ \mathbf{h}(\cdot)$ looks like $D(\cdot)$ as much as possible. There are many metrics to quantitatively characterize the similarity between $T \circ \mathbf{h}(\cdot)$ and $D(\cdot)$, for example, mutual information [18] and sum of squared distance (SSD) [22]. In monomodality image registration, the most commonly used similarity is SSD defined by

$$(1.1) \quad \int_{\Omega} [T(\mathbf{h}(\mathbf{x})) - D(\mathbf{x})]^2 d\mathbf{x}.$$

On the other hand, $\mathbf{h}(\mathbf{x})$ is divided into identity part \mathbf{x} and displacement $\mathbf{u}(\mathbf{x})$. That is, $\mathbf{h}(\mathbf{x}) \triangleq \mathbf{x} + \mathbf{u}(\mathbf{x})$. Based on this division, the image registration problem is formulated by the following variational framework:

*Received by the editors December 2, 2020; accepted for publication (in revised form) August 2, 2021; published electronically October 18, 2021.

<https://doi.org/10.1137/20M1383987>

Funding: This research was supported by the National Natural Science Foundation of China (11771127, 11871386, 11871387, 11931012, and 11901443), the National Key Research and Development Program of China (2020Y-FA0714200), and Fundamental Research Funds for the Central Universities (WUT: 2020IVB033, WUT: 2020IB011).

[†]Department of Mathematics, Wuhan University of Technology, Wuhan 430070, China (hanhuan11@whut.edu.cn).

[‡]Center for Mathematical Sciences and Department of Mathematics, Wuhan University of Technology, Wuhan 430070, China (zpwang@whut.edu.cn).

[§]Corresponding author. Center for Mathematical Sciences and Department of Mathematics, Wuhan University of Technology, Wuhan 430070, China (zhangyimin@whut.edu.cn).

$$(1.2) \quad \mathbf{u} = \arg \min_{\mathbf{u} \in \mathcal{K}} S(\mathbf{u}),$$

where here and in what follows, $S(\mathbf{u}) \triangleq \int_{\Omega} [T(\mathbf{x} + \mathbf{u}(\mathbf{x})) - D(\mathbf{x})]^2 d\mathbf{x}$, and \mathcal{K} is some proper function space.

(1.2) is an ill-posed inverse problem. One classical way to make this inverse problem well-posed is adding some regularization $R(\mathbf{u})$ [1] on functional $S(\mathbf{u})$. That is,

$$(1.3) \quad \mathbf{u} = \arg \min_{\mathbf{u} \in \mathcal{K}} F(\mathbf{u}),$$

where $F(\mathbf{u}) = S(\mathbf{u}) + R(\mathbf{u})$.

By giving different $R(\mathbf{u})$, many variational image registration models [13, 30, 36, 47, 48] are proposed. However, in most of these models, the physical mesh folding phenomenon is not taken into consideration. In the physical view, mesh folding produces negative volume, which contradicts physical law. Therefore, it's necessary to eliminate mesh folding in image registration. For this purpose, quasi conformal/conformal theory is introduced in image registration and surface registration [16, 21, 22, 25, 26, 39, 40, 44, 45]. Following the pioneering work in [8], quasi conformal/conformal theory is introduced in many fields of image processing such as surface registration [6, 19, 21, 31, 32, 43], image registration [16, 44, 45], image segmentation [46], and image restoration [20]. Based on these works, there are also some optimization algorithms for computing quasi conformal/conformal diffeomorphism [24]. Moreover, there are also some related works to improve the quasi conformal/conformal image registration model. One can refer to [16, 44, 45] for details. These models are essentially introduced based on framework (1.3). The solutions produced by (1.3) are of good properties. However, (1.3) only seeks a $\mathbf{u} \in \mathcal{K}$ with proper smoothness to minimize the variational functional $F(\mathbf{u})$, which is not the ultimate goal of image registration. This raises a problem of whether one can find a global minimizer of $S(\mathbf{u})$ with proper smoothness. In [29], Modin, Nachman, and Rondi gave a beautiful answer to this question under the large deformation diffeomorphic metric mapping (LDDMM) framework [2, 3, 4, 5, 12, 15, 17, 22, 23, 27, 28, 38, 41]. Unfortunately, there are no numerical results to validate the efficiency of their theoretical results. In fact, LDDMM is a differential equation constrained variational problem and the structure of the cost functional is very complex. Therefore, it's necessary to construct an analogous multiscale image registration approach for some other diffeomorphic models (i.e., [16]) with a much simpler cost functional. Motivated by this problem, we aim to give a multiscale image registration approach for the two-dimensional (2D) diffeomorphic models in [16]. In [16], Han and Wang proposed a 2D diffeomorphic image registration model. This model produces 2D diffeomorphic deformation $\mathbf{h}(\cdot)$ without mesh folding (see [16] for details). One can refer to related papers [14, 16] for details. For the purpose of introducing a multiscale approach, we rewrite the 2D diffeomorphic image registration model in [16] as follows:

$$(1.4) \quad \mathbf{u} = \arg \min_{\mathbf{u} \in \mathcal{A} \setminus \mathcal{B}_\varepsilon} \lambda S(\mathbf{u}) + R(\mathbf{u}),$$

where here and in what follows, $\Omega \triangleq (a_1, b_1) \times (a_1, b_1)$, $R(\mathbf{u}) \triangleq \mu \int_{\Omega} |\nabla^\alpha \mathbf{u}|^2 d\mathbf{x}$, $\mu, \lambda > 0$, $\mathcal{A} \triangleq \{\mathbf{u} = (u_1, u_2)^T \in [H_0^\alpha(\Omega)]^2 : \frac{\partial u_1}{\partial x_1} = \frac{\partial u_2}{\partial x_2}, \frac{\partial u_1}{\partial x_2} = -\frac{\partial u_2}{\partial x_1}\}$, $\mathcal{B}_\varepsilon \triangleq \{\mathbf{u} = (u_1, u_2)^T \in \mathcal{A} : \det(\nabla(\mathbf{x} + \mathbf{u}(\mathbf{x}))) < \varepsilon\}$, $\alpha > 2$, $\varepsilon > 0$ is small enough. For $\mathbf{x} = (x_1, x_2) \in \Omega$, $i = 1, 2$, and function $g : \Omega \rightarrow \mathbb{R}$, the left and right fractional order differential derivatives are defined by

$$(1.5) \quad \frac{\partial^\alpha g(\mathbf{x})}{\partial x_i^\alpha} \triangleq \frac{1}{\Gamma([\alpha] + 1 - \alpha)} \left(\frac{\partial}{\partial x_i} \right)^{[\alpha]+1} \int_{a_1}^{x_i} \frac{g^{(i)}(\mathbf{x}, t)}{(x_i - t)^{\alpha - [\alpha]}} dt,$$

$$(1.6) \quad \frac{\partial^{\alpha*} g(\mathbf{x})}{\partial x_i^{\alpha*}} \triangleq \frac{1}{\Gamma([\alpha] + 1 - \alpha)} \left(-\frac{\partial}{\partial x_i} \right)^{[\alpha]+1} \int_{x_i}^{b_1} \frac{g^{(i)}(\mathbf{x}, t)}{(t - x_i)^{\alpha - [\alpha]}} dt,$$

respectively. Note that here $\nabla^\alpha \mathbf{u} = (\frac{\partial^\alpha u_i}{\partial x_j^\alpha})_{2 \times 2}$, $\Gamma(s) = \int_0^{+\infty} t^{s-1} e^{-t} dt$, $[\cdot]$ is a round down function, and $g^{(1)}(\mathbf{x}, t) = g(t, x_2)$, $g^{(2)}(\mathbf{x}, t) = g(x_1, t)$. Concerning the details of these two fractional order derivatives, one can refer to [12, 16, 47].

Remark 1.1. The original model in [16] is formulated by

$$(1.7) \quad \mathbf{u} = \arg \min_{\mathbf{u} \in \mathcal{A}} S(\mathbf{u}) + R(\mathbf{u}).$$

Here we add a new parameter λ in (1.4) for the purpose of introducing the multiscale approach in section 2. By ignoring the set \mathcal{B}_ε , (1.7) and (1.4) are equivalent. Moreover, by Remark 1.1 in [16],

$$(1.8) \quad \det(\nabla \mathbf{h}(\mathbf{x})) = \left(1 + \frac{\partial u_1}{\partial x_1} \right)^2 + \left(\frac{\partial u_1}{\partial x_2} \right)^2 \geq 0 \quad \forall \mathbf{x} \in \Omega.$$

To ensure the existence of inverse mapping for \mathbf{h} , deformations with $\det(\nabla \mathbf{h}(\mathbf{x})) = (1 + \frac{\partial u_1}{\partial x_1})^2 + (\frac{\partial u_1}{\partial x_2})^2 = 0$ for some $\mathbf{x} \in \Omega$ should be ruled out. Here \mathcal{B}_ε is additionally added for this purpose.

In this paper, motivated by [29, 37], we construct a multiscale approach for a 2D diffeomorphic image registration model (1.4). The proposed multiscale approach contains a series of function composition process and we show that it achieves a convergent decomposition of an optimal minimizer for variational functional $S(\mathbf{u})$. In addition, numerical tests are performed to show that the proposed multiscale approach can achieve an accurate image registration result. Compared with the original model (1.4), the advantage of the proposed multiscale approach contains the following two aspects:

- The proposed multiscale approach can efficiently deal with large deformations.
- The proposed multiscale approach is robust and essentially has nothing to do with parameters.

The rest of this paper is organized as follows. In section 2, a multiscale image registration approach is proposed. The existence of solutions for the multiscale approach and the convergence of the multiscale approach are proved. In section 3, a relaxed approach for the proposed model in section 2 is given and a related numerical algorithm is also presented. In section 4, several numerical tests are performed to validate the theoretical results in sections 2 and 3. At the end of this paper, we conclude our results and list some problems for future research.

2. Proposed multiscale approach and its theoretical analysis. There are many multiscale models for image registration. We refer the reader to [33, 34] for details. These models are essentially the combination of some different registration techniques. The proposed multiscale approach for model (1.4) is based on a variational framework and is completely different from the models in [33, 34]. This approach is formulated as follows:

- At the beginning, we seek the solution of the following variational problem:

$$(2.1) \quad \mathbf{u}^0 = \arg \min_{\mathbf{u} \in \mathcal{A} \setminus \mathcal{B}_{\varepsilon_0}} \lambda_0 \|T \circ \mathbf{h}(\cdot) - D(\cdot)\|_{L^2(\Omega)}^2 + R(\mathbf{u}),$$

where here and in what follows, $\mathbf{h}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$ and $\lambda_0 > 0, \varepsilon_0 > 0$.
Based on (2.1), we define $\tilde{\mathbf{h}}_0(\mathbf{x}) = \mathbf{x} + \mathbf{u}^0(\mathbf{x})$.

- In the second step, we seek the solution of the following variational problem:

$$(2.2) \quad \mathbf{u}^1 = \arg \min_{\mathbf{u} \in \mathcal{A} \setminus \mathcal{B}_{\varepsilon_1}} \lambda_1 \int_{\Omega} |T \circ \tilde{\mathbf{h}}_0(\mathbf{x} + \mathbf{u}(\mathbf{x})) - D(\mathbf{x})|^2 d\mathbf{x} + R(\mathbf{u})$$

for some $\lambda_1 > 0, \varepsilon_1 > 0$.

Based on (2.2), we define $\mathbf{h}_1(\mathbf{x}) = \mathbf{x} + \mathbf{u}^1(\mathbf{x})$ and $\tilde{\mathbf{h}}_1(\mathbf{x}) = \tilde{\mathbf{h}}_0 \circ \mathbf{h}_1(\mathbf{x})$.

⋮

- By induction, we seek a $\mathbf{u}^n (n \geq 1)$ such that

$$(2.3) \quad \mathbf{u}^n = \arg \min_{\mathbf{u} \in \mathcal{A} \setminus \mathcal{B}_{\varepsilon_n}} F_n(\mathbf{u}),$$

where $T_n(\mathbf{x}) = T \circ \tilde{\mathbf{h}}_{n-1}(\mathbf{x}), F_n(\mathbf{u}) = S_n(\mathbf{u}) + R(\mathbf{u}), S_n(\mathbf{u}) = \lambda_n \int_{\Omega} [T_n(\mathbf{x} + \mathbf{u}(\mathbf{x})) - D(\mathbf{x})]^2 d\mathbf{x}$, and $\lambda_n > 0, \varepsilon_n > 0$.

Here, $\mathbf{h}_n(\mathbf{x}) = \mathbf{x} + \mathbf{u}^n(\mathbf{x})$ and $\tilde{\mathbf{h}}_n(\mathbf{x}) = \tilde{\mathbf{h}}_{n-1} \circ \mathbf{h}_n(\mathbf{x})$.

Remark 2.1. The multiscale approach (2.1)–(2.3) can effectively deal with large deformation image registration, since $\tilde{\mathbf{h}}_n(\mathbf{x}) = \tilde{\mathbf{h}}_0 \circ \mathbf{h}_1 \circ \dots \circ \mathbf{h}_n(\mathbf{x})$ can produce large deformation even if $\mathbf{h}_k (k = 1, 2, \dots, n)$ are small deformations. This is an advantage of multiscale approach (2.1)–(2.3). One can see the details of large deformation image registration image pair $C - C$ in Test 1 of section 4.

(2.1)–(2.3) is an n scale approach for 2D diffeomorphic image registration.

(2.1)–(2.3) is uniformly formulated by

$$(2.4) \quad \mathbf{u}^n = \arg \min_{\mathbf{u} \in \mathcal{A} \setminus \mathcal{B}_{\varepsilon_n}} F_n(\mathbf{u}) \quad \forall n \in \mathbb{N}.$$

Concerning the existence of solutions for variational model (2.4), we have the following results.

THEOREM 2.2. *Assume $\Delta_T \triangleq \{\mathbf{x} : T(\mathbf{x}) \text{ is discontinuous at } \mathbf{x}\}$ is a zero measure set and $\text{ess sup}_{\mathbf{x} \in \Omega \setminus \Delta_T} |T(\mathbf{x})| < M_0 < +\infty, \text{ess sup}_{\mathbf{x} \in \Omega \setminus \Delta_T} |D(\mathbf{x})| < M_0 < +\infty$; then there admits a solution of (2.4) for each $n \in \mathbb{N}$.*

Proof. The proof of Theorem 2.2 is a standard process in calculus of variations. One can use the similar idea of Theorem 2.2 in [16] to complete the proof. Here we omit it. □

By (2.3), we know that for any $n \in \mathbb{N}^+$, there holds $F_n(\mathbf{u}^n) \leq F_n(\mathbf{0})$. That is,

$$(2.5) \quad F_n(\mathbf{u}^n) = \lambda_n \|T \circ \tilde{\mathbf{h}}^n(\cdot) - D(\cdot)\|_{L^2(\Omega)}^2 + R(\mathbf{u}^n) \leq F_n(\mathbf{0}) = \lambda_n \|T \circ \tilde{\mathbf{h}}^{n-1}(\cdot) - D(\cdot)\|_{L^2(\Omega)}^2.$$

Therefore,

$$(2.6) \quad \|T \circ \tilde{\mathbf{h}}^n(\cdot) - D(\cdot)\|_{L^2(\Omega)}^2 \leq \|T \circ \tilde{\mathbf{h}}^{n-1}(\cdot) - D(\cdot)\|_{L^2(\Omega)}^2 \quad \forall n \in \mathbb{N}^+.$$

This implies, $\{\|T \circ \tilde{\mathbf{h}}^n(\cdot) - D(\cdot)\|_{L^2(\Omega)}^2\}$ is a decreasing sequence with lower bound. Define

$$(2.7) \quad \phi_0 = \lim_{n \rightarrow +\infty} \|T \circ \tilde{\mathbf{h}}^n(\cdot) - D(\cdot)\|_{L^2(\Omega)}^2$$

and

$$(2.8) \quad \delta_0 = \inf \left\{ \int_{\Omega} |T(\mathbf{x} + \mathbf{u}(\mathbf{x})) - D(\mathbf{x})|^2 d\mathbf{x} : \mathbf{u} \in \mathcal{A} \right\},$$

where \mathbf{h}_n and $\tilde{\mathbf{h}}_n$ are deformations induced by the multiscale approach (2.1)–(2.3).

Concerning the problem of whether one can find a global minimizer of $S(\mathbf{u})$ on \mathcal{A} , we have the following result.

THEOREM 2.3. *Let \mathbf{h}_n and $\tilde{\mathbf{h}}_n$ be deformations induced by the multiscale approach (2.1)–(2.3), and assume $B = B(\Omega)$, M , and λ_n are three large numbers with $\lim_{n \rightarrow +\infty} \frac{B^{4n-3}M^{4^n}}{\lambda_n} = 0$, $\lim_{n \rightarrow +\infty} \varepsilon_n = 0$. Then there holds $\phi_0 = \delta_0$.*

Proof. The proof of Theorem 2.3 will be given at the end of this section. \square

Remark 2.4. There are two remarks on Theorem 2.3:

(i) Theorem 2.3 shows the convergence of multiscale approach (2.1)–(2.3). Theorem 2.3 implies that one can find an optimal solution for the 2D diffeomorphic image registration model by using the multiscale approach (2.1)–(2.3) as n is large enough. By Theorem 2.3, we know that the multiscale approach (2.1)–(2.3) is essentially equivalent to the variational problem,

$$(2.9) \quad \mathbf{u} = \arg \min_{\mathbf{u} \in \mathcal{A}} S(\mathbf{u}),$$

while the model (1.4) is equivalent to

$$(2.10) \quad \mathbf{u} = \arg \min_{\mathbf{u} \in \mathcal{A} \setminus \mathcal{B}_\varepsilon} S(\mathbf{u}) + \frac{1}{\lambda} R(\mathbf{u}).$$

In this view, (2.1)–(2.3) is much more accurate than model (1.4). Moreover, one can also notice from the comparison between (2.9) and (2.10) that the multiscale approach (2.1)–(2.3) essentially has nothing to do with parameters while the solution of (1.4) is affected by parameters such as λ , μ , and α . This makes the multiscale approach (2.1)–(2.3) much more robust. This is also the main motivation for us to introduce the multi scale approach (2.1)–(2.3).

(ii) One can notice from the proof of Theorem 2.3 that $B = B(\Omega)$ is a number depending on Ω , and M is a number depending on some diffeomorphism $\tilde{\mathbf{h}}$. Intuitively, by choosing B and M to be some large numbers, the conditions in Theorem 2.3 are naturally satisfied. In practice, for the domain $\Omega = (0, 128) \times (0, 128)$, we notice that $B = 10$, $M = 5$ is enough to satisfy the conditions in Theorem 2.3.

Before giving the proof of Theorem 2.3, let's introduce some notation and lemmas which are necessary in our proof.

For $\mathbf{f} : \Omega \rightarrow \Omega$, define

$$(2.11) \quad \mathcal{W}(\mathbf{f}) = \mathbf{f} - \mathbf{I},$$

where \mathbf{I} is identity mapping.

Based on (2.11), we have that

$$(2.12) \quad \mathcal{W}(\mathbf{f})(\mathbf{x}) = \mathbf{f}(\mathbf{x}) - \mathbf{x}.$$

Remark 2.5. For any deformation $\mathbf{f}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$, $\mathcal{W}(\mathbf{f})(\mathbf{x}) = \mathbf{u}(\mathbf{x})$.

The multiscale approach (2.1)–(2.3) contains a series of function composition. This raises a problem of whether $\mathbf{f} \circ \mathbf{g}$ is still diffeomorphic if \mathbf{f}, \mathbf{g} are diffeomorphisms. Concerning this problem, we have the following result.

LEMMA 2.6. *Assume $\mathbf{f}, \mathbf{g} : \Omega \rightarrow \Omega$ and $\mathcal{W}(\mathbf{f}) \in \mathcal{A} \setminus \mathcal{B}_{\varepsilon_1}, \mathcal{W}(\mathbf{g}) \in \mathcal{A} \setminus \mathcal{B}_{\varepsilon_2}$ for some $\varepsilon_1, \varepsilon_2 > 0$; then $\mathcal{W}(\mathbf{f} \circ \mathbf{g}) \in \mathcal{A} \setminus \mathcal{B}_{\varepsilon_1 \varepsilon_2}$.*

Proof. $\mathcal{W}(\mathbf{f}) \in \mathcal{A} \setminus \mathcal{B}_{\varepsilon_1}, \mathcal{W}(\mathbf{g}) \in \mathcal{A} \setminus \mathcal{B}_{\varepsilon_2}$ imply that for any $\mathbf{x} \in \Omega$, there hold

$$(2.13) \quad \frac{\partial f_1(\mathbf{x})}{\partial x_1} = \frac{\partial f_2(\mathbf{x})}{\partial x_2}, \frac{\partial f_1(\mathbf{x})}{\partial x_2} = -\frac{\partial f_2(\mathbf{x})}{\partial x_1}, \frac{\partial g_1(\mathbf{x})}{\partial x_1} = \frac{\partial g_2(\mathbf{x})}{\partial x_2}, \frac{\partial g_1(\mathbf{x})}{\partial x_2} = -\frac{\partial g_2(\mathbf{x})}{\partial x_1}$$

and

$$(2.14) \quad \det(\nabla \mathbf{f}(\mathbf{x})) \geq \varepsilon_1, \quad \det(\nabla \mathbf{g}(\mathbf{x})) \geq \varepsilon_2 \quad \text{for any } \mathbf{x} \in \Omega.$$

By the chain rule, we obtain that

$$(2.15) \quad \frac{\partial f_1(\mathbf{g}(\mathbf{x}))}{\partial x_1} = \frac{\partial f_1(\mathbf{g}(\mathbf{x}))}{\partial g_1} \frac{\partial g_1(\mathbf{x})}{\partial x_1} + \frac{\partial f_1(\mathbf{g}(\mathbf{x}))}{\partial g_2} \frac{\partial g_2(\mathbf{x})}{\partial x_1}$$

and

$$(2.16) \quad \frac{\partial f_2(\mathbf{g}(\mathbf{x}))}{\partial x_2} = \frac{\partial f_2(\mathbf{g}(\mathbf{x}))}{\partial g_2} \frac{\partial g_2(\mathbf{x})}{\partial x_2} + \frac{\partial f_2(\mathbf{g}(\mathbf{x}))}{\partial g_1} \frac{\partial g_1(\mathbf{x})}{\partial x_2}.$$

By (2.13), (2.15), and (2.16), we deduce that

$$(2.17) \quad \frac{\partial f_1(\mathbf{g}(\mathbf{x}))}{\partial x_1} = \frac{\partial f_2(\mathbf{g}(\mathbf{x}))}{\partial x_2}.$$

Similarly, we have

$$(2.18) \quad \frac{\partial f_1(\mathbf{g}(\mathbf{x}))}{\partial x_2} = -\frac{\partial f_2(\mathbf{g}(\mathbf{x}))}{\partial x_1}.$$

Moreover, by a simple derivation based on (2.17) and (2.18), we obtain that

$$(2.19) \quad \begin{aligned} \det(\nabla_{\mathbf{x}} \mathbf{f}(\mathbf{g}(\mathbf{x}))) &= \left(\frac{\partial f_1(\mathbf{g}(\mathbf{x}))}{\partial x_1} \right)^2 + \left(\frac{\partial f_1(\mathbf{g}(\mathbf{x}))}{\partial x_2} \right)^2 \\ &= \det(\nabla_{\mathbf{g}} \mathbf{f}(\mathbf{g})) \det(\nabla_{\mathbf{x}} \mathbf{g}(\mathbf{x})) \geq \varepsilon_1 \varepsilon_2. \end{aligned}$$

It follows from (2.17), (2.18), and (2.19) that $\mathcal{W}(\mathbf{f} \circ \mathbf{g}) \in \mathcal{A} \setminus \mathcal{B}_{\varepsilon_1 \varepsilon_2}$. □

Remark 2.7. Lemma 2.6 implies that the multiscale approach (2.1)–(2.3) produces diffeomorphic deformations $\mathbf{h} : \Omega \rightarrow \Omega$ without mesh folding.

LEMMA 2.8. *Assume $\mathbf{A} = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}$, then $\|\mathbf{A}\|^2 = 2 \det(\mathbf{A})$. Here and in what follows, $\|\mathbf{A}\|^2 = \sum_{i,j=1}^2 (a_{i,j})^2$ for any matrix $\mathbf{A} = (a_{i,j})_{n \times n}$.*

Proof. The proof is only a simple calculation; here we omit it. □

LEMMA 2.9. *Assume $\mathbf{u} \in \mathcal{A}$, then $\|\nabla \mathbf{u}\|^2 = 2 \det(\nabla \mathbf{u})$.*

Proof. The proof naturally comes from Lemma 2.8 by letting $\mathbf{A} = \nabla \mathbf{u}$. □

Since \mathbf{h}_n induced by multiscale approach (2.1)–(2.3) is diffeomorphic, this raises a problem of whether its inverse mapping is still diffeomorphic. Concerning this problem, we have the following results.

LEMMA 2.10. *If $\mathbf{h}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$ and $\mathbf{u} \in \mathcal{A} \setminus \mathcal{B}_\varepsilon$, then there exists $\mathbf{g}(\mathbf{x}) = \mathbf{h}^{-1}(\mathbf{x}) = \mathbf{x} + \mathbf{v}(\mathbf{x})$ and $\mathbf{v} \in \mathcal{A}$. Moreover, there hold*

- (i) $\mathbf{u}(\mathbf{x}) = -\mathbf{v}(\mathbf{x} + \mathbf{u}(\mathbf{x}))$, $\mathbf{v}(\mathbf{x}) = -\mathbf{u}(\mathbf{x} + \mathbf{v}(\mathbf{x}))$;
- (ii) $\frac{1}{\varepsilon} \geq \det(\nabla \mathbf{g}(\mathbf{x})) > 0$.

Proof. Since $\mathbf{u} \in \mathcal{A} \setminus \mathcal{B}_\varepsilon$, there hold $\det(\nabla \mathbf{h}(\mathbf{x})) \geq \varepsilon$ for any $\mathbf{x} \in \Omega$. By the inverse mapping theorem [10], there exists an inverse mapping $\mathbf{g} : \Omega \rightarrow \Omega$ such that $\mathbf{g}(\mathbf{x}) = \mathbf{h}^{-1}(\mathbf{x}) = \mathbf{x} + \mathbf{v}(\mathbf{x})$.

(i) By definition of inverse mapping, there holds

$$(2.20) \quad \mathbf{x} = \mathbf{g}(\mathbf{h}(\mathbf{x})) = \mathbf{h}(\mathbf{x}) + \mathbf{v}(\mathbf{h}(\mathbf{x})) = \mathbf{x} + \mathbf{u}(\mathbf{x}) + \mathbf{v}(\mathbf{x} + \mathbf{u}(\mathbf{x})).$$

This implies $\mathbf{u}(\mathbf{x}) = -\mathbf{v}(\mathbf{x} + \mathbf{u}(\mathbf{x}))$.

Similarly,

$$(2.21) \quad \mathbf{x} = \mathbf{h}(\mathbf{g}(\mathbf{x})) = \mathbf{g}(\mathbf{x}) + \mathbf{u}(\mathbf{g}(\mathbf{x})) = \mathbf{x} + \mathbf{v}(\mathbf{x}) + \mathbf{u}(\mathbf{x} + \mathbf{v}(\mathbf{x})).$$

That is, $\mathbf{v}(\mathbf{x}) = -\mathbf{u}(\mathbf{x} + \mathbf{v}(\mathbf{x}))$.

(ii) By (i), we know that $\mathbf{u}(\mathbf{x}) = -\mathbf{v}(\mathbf{x} + \mathbf{u}(\mathbf{x}))$, $\mathbf{v}(\mathbf{x}) = -\mathbf{u}(\mathbf{x} + \mathbf{v}(\mathbf{x}))$. Moreover, there hold $\frac{\partial u_1}{\partial g_1}(\mathbf{g}) = \frac{\partial u_2}{\partial g_2}(\mathbf{g})$, $\frac{\partial u_1}{\partial g_2}(\mathbf{g}) = -\frac{\partial u_2}{\partial g_1}(\mathbf{g})$.

Therefore, we deduce that

$$(2.22) \quad \frac{\partial v_1(\mathbf{x})}{\partial x_1} = -\frac{\partial u_1(\mathbf{x} + \mathbf{v}(\mathbf{x}))}{\partial x_1} = -\frac{\partial u_1}{\partial g_1}(\mathbf{g}) \left(1 + \frac{\partial v_1(\mathbf{x})}{\partial x_1}\right) - \frac{\partial u_1}{\partial g_2}(\mathbf{g}) \frac{\partial v_2(\mathbf{x})}{\partial x_1},$$

$$(2.23) \quad \frac{\partial v_2(\mathbf{x})}{\partial x_2} = -\frac{\partial u_2(\mathbf{x} + \mathbf{v}(\mathbf{x}))}{\partial x_2} = -\frac{\partial u_2}{\partial g_1}(\mathbf{g}) \frac{\partial v_1(\mathbf{x})}{\partial x_2} - \frac{\partial u_2}{\partial g_2}(\mathbf{g}) \left(1 + \frac{\partial v_2(\mathbf{x})}{\partial x_2}\right).$$

By (2.22) and (2.23), we have that

$$(2.24) \quad \left(1 + \frac{\partial u_1}{\partial g_1}(\mathbf{g})\right) \left(\frac{\partial v_1(\mathbf{x})}{\partial x_1} - \frac{\partial v_2(\mathbf{x})}{\partial x_2}\right) + \frac{\partial u_1}{\partial g_2}(\mathbf{g}) \left(\frac{\partial v_2(\mathbf{x})}{\partial x_1} + \frac{\partial v_1(\mathbf{x})}{\partial x_2}\right) = 0.$$

Similarly, we get that

$$(2.25) \quad \frac{\partial v_1(\mathbf{x})}{\partial x_2} = -\frac{\partial u_1(\mathbf{x} + \mathbf{v}(\mathbf{x}))}{\partial x_2} = -\frac{\partial u_1}{\partial g_1}(\mathbf{g}) \frac{\partial v_1(\mathbf{x})}{\partial x_2} - \frac{\partial u_1}{\partial g_2}(\mathbf{g}) \left(1 + \frac{\partial v_2(\mathbf{x})}{\partial x_2}\right),$$

$$(2.26) \quad \frac{\partial v_2(\mathbf{x})}{\partial x_1} = -\frac{\partial u_2(\mathbf{x} + \mathbf{v}(\mathbf{x}))}{\partial x_1} = -\frac{\partial u_2}{\partial g_1}(\mathbf{g}) \left(1 + \frac{\partial v_1(\mathbf{x})}{\partial x_1}\right) - \frac{\partial u_2}{\partial g_2}(\mathbf{g}) \frac{\partial v_2(\mathbf{x})}{\partial x_1}.$$

By (2.25) and (2.26), we deduce that

$$(2.27) \quad -\frac{\partial u_1}{\partial g_2}(\mathbf{g}) \left(\frac{\partial v_1(\mathbf{x})}{\partial x_1} - \frac{\partial v_2(\mathbf{x})}{\partial x_2}\right) + \left(1 + \frac{\partial u_1}{\partial g_1}(\mathbf{g})\right) \left(\frac{\partial v_2(\mathbf{x})}{\partial x_1} + \frac{\partial v_1(\mathbf{x})}{\partial x_2}\right) = 0.$$

(2.24) and (2.27) are linear equations on $\frac{\partial v_1(\mathbf{x})}{\partial x_1} - \frac{\partial v_2(\mathbf{x})}{\partial x_2}$ and $\frac{\partial v_2(\mathbf{x})}{\partial x_1} + \frac{\partial v_1(\mathbf{x})}{\partial x_2}$.

Moreover, the determinant of coefficient is

$$(2.28) \quad \begin{vmatrix} 1 + \frac{\partial u_1}{\partial g_1}(\mathbf{g}) & \frac{\partial u_1}{\partial g_2}(\mathbf{g}) \\ -\frac{\partial u_1}{\partial g_2}(\mathbf{g}) & 1 + \frac{\partial u_1}{\partial g_1}(\mathbf{g}) \end{vmatrix} = \left(1 + \frac{\partial u_1}{\partial g_1}(\mathbf{g})\right)^2 + \left(\frac{\partial u_1}{\partial g_2}(\mathbf{g})\right)^2 \geq \varepsilon > 0.$$

By Cramer’s rule [35], we obtain that $\frac{\partial v_1(\mathbf{x})}{\partial x_1} - \frac{\partial v_2(\mathbf{x})}{\partial x_2} = 0$ and $\frac{\partial v_2(\mathbf{x})}{\partial x_1} + \frac{\partial v_1(\mathbf{x})}{\partial x_2} = 0$. That is, $\mathbf{v} \in \mathcal{A}$. Moreover, by (2.19) in Lemma 2.6, we know that

$$(2.29) \quad 1 = \det(\nabla \mathbf{x}) = \det(\nabla(\mathbf{f}(\mathbf{g}(\mathbf{x})))) = \det(\nabla \mathbf{f}(\mathbf{g})) \det(\nabla \mathbf{g}(\mathbf{x})).$$

This implies $\frac{1}{\varepsilon} \geq \det(\nabla \mathbf{g}(\mathbf{x})) > 0$ for any $\mathbf{x} \in \Omega$. □

Next, we introduce three important lemmas which will be used later in our proof.

LEMMA 2.11. Assume $\mathcal{W}(\mathbf{g}) \in \mathcal{A} \setminus \mathcal{B}_\varepsilon$, then $\int_\Omega \mathbf{f}(\mathbf{g}(\mathbf{x}))d\mathbf{x} \leq CR(\mathbf{g}^{-1}) \int_\Omega \mathbf{f}(\mathbf{y})d\mathbf{y}$.

Proof. Let $\mathbf{y} = \mathbf{g}(\mathbf{x})$; then by Lemma 2.10, there exists an inverse mapping $\mathbf{g}^{-1} : \Omega \rightarrow \Omega$ such that $\mathbf{x} = \mathbf{g}^{-1}(\mathbf{y})$. Therefore,

$$(2.30) \quad \int_\Omega \mathbf{f}(\mathbf{g}(\mathbf{x}))d\mathbf{x} = \int_\Omega \mathbf{f}(\mathbf{y}) \det(\nabla_{\mathbf{y}} \mathbf{g}^{-1}(\mathbf{y}))d\mathbf{y}.$$

By Lemma 2.10, we obtain that $\mathbf{g}^{-1}(\mathbf{y}) \in \mathcal{A}$. Furthermore, by Lemma 2.9, we know that

$$(2.31) \quad \det(\nabla_{\mathbf{y}} \mathbf{g}^{-1}(\mathbf{y})) = \frac{1}{2} \|\nabla_{\mathbf{y}} \mathbf{g}^{-1}(\mathbf{y})\|^2.$$

By the Sobolev embedding theorem ($H_0^\alpha(\Omega) \hookrightarrow C^1(\Omega)$) [9, 12], we deduce that

$$(2.32) \quad \max_{\mathbf{y} \in \Omega} \|\nabla_{\mathbf{y}} \mathbf{g}^{-1}(\mathbf{y})\|^2 \leq \|\nabla_{\mathbf{y}} \mathbf{g}^{-1}(\mathbf{y})\|_{C^1(\Omega)}^2 \leq C \|\nabla^\alpha \mathbf{g}^{-1}(\mathbf{y})\|_{L^2(\Omega)}^2 = CR(\mathbf{g}^{-1}).$$

By (2.30), (2.31), and (2.32), we get that

$$(2.33) \quad \int_\Omega \mathbf{f}(\mathbf{g}(\mathbf{x}))d\mathbf{x} \leq \frac{1}{2} \max_{\mathbf{y} \in \Omega} \|\nabla_{\mathbf{y}} \mathbf{g}^{-1}(\mathbf{y})\|^2 \int_\Omega \mathbf{f}(\mathbf{y})d\mathbf{y} \leq CR(\mathbf{g}^{-1}) \int_\Omega \mathbf{f}(\mathbf{y})d\mathbf{y}. \quad \square$$

LEMMA 2.12. Assume $\mathbf{p}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$ and $\mathcal{W}(\mathbf{q}) \in \mathcal{A} \setminus \mathcal{B}_\varepsilon$, then $R(\mathcal{W}(\mathbf{p} \circ \mathbf{q})) = 2[R(\mathcal{W}(\mathbf{q})) + CR(\mathbf{q}^{-1})R(\mathcal{W}(\mathbf{p}))]$.

Proof. By the inequality $|a + b|^2 \leq 2|a|^2 + 2|b|^2$, we deduce that

$$(2.34) \quad R(\mathcal{W}(\mathbf{p} \circ \mathbf{q})) = R(\mathcal{W}(\mathbf{q}) + \mathbf{u}(\mathbf{q})) \leq 2R(\mathcal{W}(\mathbf{q})) + 2\mu \int_\Omega |\nabla^\alpha \mathbf{u}(\mathbf{q}(\mathbf{x}))|^2 d\mathbf{x}.$$

Note that here we use the equality $\mathbf{p} \circ \mathbf{q} = \mathbf{q}(\mathbf{x}) + \mathbf{u}(\mathbf{q}) = \mathbf{x} + \mathcal{W}(\mathbf{q}) + \mathbf{u}(\mathbf{q})$.

Let $f(\mathbf{x}) = |\nabla^\alpha \mathbf{u}(\mathbf{x})|^2$; then it follows from Lemma 2.11 that

$$(2.35) \quad \mu \int_\Omega |\nabla^\alpha \mathbf{u}(\mathbf{q}(\mathbf{x}))|^2 d\mathbf{x} \leq CR(\mathbf{q}^{-1})R(\mathcal{W}(\mathbf{p})).$$

By (2.34) and (2.35), we know that

$$(2.36) \quad R(\mathcal{W}(\mathbf{p} \circ \mathbf{q})) = 2[R(\mathcal{W}(\mathbf{q})) + CR(\mathbf{q}^{-1})R(\mathcal{W}(\mathbf{p}))]. \quad \square$$

LEMMA 2.13. Assume $\mathbf{h}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$, $\mathbf{g}(\mathbf{x}) = \mathbf{h}^{-1}(\mathbf{x}) = \mathbf{x} + \mathbf{v}(\mathbf{x})$, and $\mathbf{u}, \mathbf{v} \in \mathcal{A} \setminus \mathcal{B}_\varepsilon$; then $R(\mathbf{u}) = \int_\Omega \|\nabla^\alpha \mathbf{u}(\mathbf{x})\|^2 d\mathbf{x} \leq CR(\mathbf{g})R(\mathcal{W}(\mathbf{g}))$.

Proof. By Lemma 2.10, $\mathbf{u}(\mathbf{x}) = -\mathbf{v}(\mathbf{x} + \mathbf{u}(\mathbf{x}))$. Therefore

$$(2.37) \quad R(\mathbf{u}) = \int_\Omega \|\nabla^\alpha \mathbf{u}(\mathbf{x})\|^2 d\mathbf{x} = \int_\Omega \|\nabla^\alpha \mathbf{v}(\mathbf{x} + \mathbf{u}(\mathbf{x}))\|^2 d\mathbf{x}.$$

Let $f(\mathbf{x}) = |\nabla^\alpha \mathbf{v}(\mathbf{x})|^2$; then by Lemma 2.11, there hold

$$(2.38) \quad R(\mathbf{u}) = \int_\Omega f(\mathbf{h}(\mathbf{x}))d\mathbf{x} \leq CR(\mathbf{h}^{-1}) \int_\Omega \|\nabla^\alpha \mathbf{v}(\mathbf{x})\|^2 d\mathbf{x} = CR(\mathbf{g})R(\mathcal{W}(\mathbf{g})). \quad \square$$

At the end of this section, we give a proof of Theorem 2.3.

First, we introduce a function which is necessary in our proof. For any $\xi \geq 0$, define

$$(2.39) \quad \mathcal{M}(\xi) = \begin{cases} 1, & 0 \leq \xi \leq 1, \\ \xi, & \xi > 1. \end{cases}$$

Remark 2.14. In proof of Theorem 2.3, some inequalities between ξ^n and ξ are necessary, i.e., formula (2.50). Since $\xi^n \leq \xi (\xi \leq 1)$ and $\xi^n \geq \xi (\xi \geq 1)$, the discussion is necessary. To simplify this kind of discussion, we introduce the function $\mathcal{M}(\xi)$.

Concerning the properties of function \mathcal{M} , we have the following results.

LEMMA 2.15. For $\xi, \xi_1, \xi_2 \geq 0$, there hold

- (i) $\mathcal{M}(\xi) \geq \xi$;
- (ii) $\mathcal{M}^n(\xi) = \mathcal{M}(\xi)$ and $\mathcal{M}(\xi^n) = [\mathcal{M}(\xi)]^n (n \in \mathbb{N}^+)$;
- (iii) $\mathcal{M}(c\xi) \leq c\mathcal{M}(\xi)$ for $c \geq 1$;
- (iv) if $\xi_1 \leq \xi_2$, then $\mathcal{M}(\xi_1) \leq \mathcal{M}(\xi_2)$.

Proof. One can get the proof by direct computation; here we omit it. □

Proof of Theorem 2.3. It's clear that $\phi_0 \geq \delta_0$. We only need to prove $\phi_0 \leq \delta_0$.

By contradiction, assume $\delta_0 < \phi_0$; then there exists $0 < C_1 < 1$ such that $\delta_0 < C_1\phi_0 < \phi_0$.

By definition of δ_0 , there exists an $\bar{\mathbf{h}}(\mathbf{x}) = \mathbf{x} + \bar{\mathbf{u}}(\mathbf{x}) \in \mathcal{A}$ such that

$$(2.40) \quad \|T(\bar{\mathbf{h}}(\cdot)) - D(\cdot)\|_{L^2(\Omega)}^2 < C_1\phi_0 \quad \text{and} \quad R(\mathcal{W}(\bar{\mathbf{h}})) < +\infty.$$

Choosing $\mathbf{h} = \tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}}$, then by Lemmas 2.6 and 2.10, we know that $\tilde{\mathbf{h}}_{n-1}^{-1} \in \mathcal{A}$ and $\mathbf{h} \in \mathcal{A}$. Therefore, we deduce that

$$(2.41) \quad \begin{aligned} & \lambda_n \|T \circ \tilde{\mathbf{h}}_n(\cdot) - D(\cdot)\|_{L^2(\Omega)}^2 + R(\mathcal{W}(\mathbf{h}_n)) \\ & \leq \lambda_n \|T \circ \bar{\mathbf{h}}(\cdot) - D(\cdot)\|_{L^2(\Omega)}^2 + R(\mathcal{W}(\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})) \\ & \leq \lambda_n C_1\phi_0 + R(\mathcal{W}(\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})). \end{aligned}$$

That is,

$$(2.42) \quad \lambda_n(1 - C_1)\phi_0 + R(\mathcal{W}(\mathbf{h}_n)) \leq R(\mathcal{W}(\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})).$$

This implies

$$(2.43) \quad R(\mathcal{W}(\mathbf{h}_n)) \leq R(\mathcal{W}(\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})).$$

On the other hand, $\mathbf{h}_n, \tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}} \in \mathcal{A}$. By Lemmas 2.11, 2.12, and 2.13, we have that

$$(2.44) \quad R(\mathcal{W}(\tilde{\mathbf{h}}_n^{-1} \circ \bar{\mathbf{h}})) = R(\mathcal{W}(\mathbf{h}_n^{-1} \circ \tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})) \leq 2[R(\mathcal{W}(\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})) + CR((\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})^{-1})R(\mathcal{W}(\mathbf{h}_n^{-1}))].$$

Note that here $C = C(\Omega)$ is a number depending on Ω .

What's more, by (2.43) and Lemma 2.13, we deduce that

$$(2.45) \quad R(\mathcal{W}(\mathbf{h}_n^{-1})) \leq CR(\mathbf{h}_n)R(\mathcal{W}(\mathbf{h}_n)) \leq CR(\mathbf{h}_n)R(\mathcal{W}(\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})).$$

Moreover,

$$(2.46) \quad R(\mathbf{h}_n) = R(\mathcal{W}(\mathbf{h}_n) + \mathbf{x}) \leq 2R(\mathcal{W}(\mathbf{h}_n)) + \bar{C};$$

here and in what follows, $\bar{C} = 2R(\mathbf{x}) = 2\mu \int_{\Omega} |\nabla^\alpha \mathbf{x}|^2 d\mathbf{x}$ is a constant.

It follows from (2.43) and (2.45) that

$$(2.47) \quad R(\mathcal{W}(\mathbf{h}_n^{-1})) \leq CR(\mathcal{W}(\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}}))[2R(\mathcal{W}(\mathbf{h}_n)) + \bar{C}].$$

Noticing that $R(\mathcal{W}(\mathbf{h}_n)) \leq R(\mathcal{W}(\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}}))$ in (2.46), (2.47) is reformulated to

$$(2.48) \quad R(\mathcal{W}(\mathbf{h}_n^{-1})) \leq 2CR^2(\mathcal{W}(\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})) + \bar{C}R(\mathcal{W}(\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})).$$

In addition, by Lemma 2.13, we know that

$$(2.49) \quad \begin{aligned} R((\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})^{-1}) &= R(\mathbf{x} + \mathcal{W}((\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})^{-1})) \leq 2R(\mathbf{x}) + 2R(\mathcal{W}((\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})^{-1})) \\ &\leq 2R(\mathcal{W}((\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})^{-1})) + \bar{C} \\ &\leq CR(\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})R(\mathcal{W}(\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})) + \bar{C} \\ &\leq CR^2(\mathcal{W}(\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})) + \bar{C}R(\mathcal{W}(\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})) + \hat{C}. \end{aligned}$$

Note that here we use the relationship $R(\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}}) = R(\mathbf{x} + \mathcal{W}(\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})) \leq 2[R(\mathbf{x}) + R(\mathcal{W}(\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}}))]$.

By (2.44), (2.48), and (2.49), we get that

$$(2.50) \quad R(\mathcal{W}(\tilde{\mathbf{h}}_n^{-1} \circ \bar{\mathbf{h}})) \leq \sum_{k=0}^4 B_k R^k(\mathcal{W}(\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})) \leq B[\mathcal{M}(R(\mathcal{W}(\tilde{\mathbf{h}}_{n-1}^{-1} \circ \bar{\mathbf{h}})))]^4,$$

where $B_k (k = 0, 1, 2, 3, 4)$ are some given real numbers (functions of $C(\Omega)$, \bar{C} , and \hat{C}) and $B = B(\Omega) = \max\{\sum_{k=0}^4 |B_k|, 1\}$.

Furthermore, by Lemma 2.15 and (2.50), we have that

$$(2.51) \quad \begin{aligned} R(\mathcal{W}(\tilde{\mathbf{h}}_n^{-1} \circ \bar{\mathbf{h}})) &\leq B^5[\mathcal{M}(R(\mathcal{W}(\tilde{\mathbf{h}}_{n-2}^{-1} \circ \bar{\mathbf{h}})))]^4 \leq \dots \leq B^{4n-3}[\mathcal{M}(R(\mathcal{W}(\tilde{\mathbf{h}}_0^{-1} \circ \bar{\mathbf{h}})))]^{4^n} \\ &= B^{4n-3}M^{4^n}, \end{aligned}$$

where $M \triangleq \mathcal{M}(R(\mathcal{W}(\tilde{\mathbf{h}}_0^{-1} \circ \bar{\mathbf{h}})))$.

By (2.42) and (2.51), we deduce that

$$(2.52) \quad \lambda_n \phi_0(1 - C_1) + R(\mathbf{u}_n) \leq B^{4n-3}M^{4^n}.$$

This contradicts the fact that $\lim_{n \rightarrow +\infty} \frac{B^{4n-3}M^{4^n}}{\lambda_n} = 0$.

Therefore, $\phi_0 = \delta_0$. □

3. Relaxed multiscale approach and its numerical implementation. In this section, we discuss the problem of numerical implementation of the multiscale approach (2.4).

(2.4) is a partial differential equation (PDE) constrained problem. The constraints make too much trouble in numerical implementation and theoretical analysis. Motivated by [16], we transform model (2.4) into the following unconstrained variational problem approximately:

$$(3.1) \quad \mathbf{u}^n = \arg \min_{\mathbf{u} \in [H_0^\alpha(\Omega)]^2} F_n(\mathbf{u}) + \Theta_n R_1(\mathbf{u});$$

here and in what follows, $R_1(\mathbf{u}) = \int_{\Omega} (\frac{\partial u_1}{\partial x_1} - \frac{\partial u_2}{\partial x_2})^2 + (\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1})^2 d\mathbf{x}$, and Θ_n is large enough.

Remark 3.1. There are two remarks on (3.1):

(i) Some smoothness can already be achieved by minimizing $S_n(\mathbf{u}) + \Theta_n R_1(\mathbf{u})$. To make the algorithm much faster, one can also set $\mu = 0$, which is enough to produce smooth deformation \mathbf{h} . In extreme situations (such as a very large deformation), setting a positive μ can help to achieve much smoother registration results.

(ii) (3.1) is a relaxed form of (2.4). The cost functional of (3.1) is reformulated as $S_n(\mathbf{u}) + \bar{R}(\mathbf{u})$, where $\bar{R}(\mathbf{u}) = R(\mathbf{u}) + \Theta_n R_1(\mathbf{u})$. By replacing $R(\mathbf{u})$ with $\bar{R}(\mathbf{u})$ in (2.4), one can use the similar way of Theorem 2.3 to prove the convergence of the relaxed multiscale approach in this section. Here we do not repeat it. Based on this convergence result, one can notice from Remark 2.4 that the relaxed multiscale approach in this section is also equivalent to (2.9). That is, the relaxed multiscale approach has nothing to do with parameters. This is the main reason why no sensitivity tests on parameters like Θ_n , α , μ are listed in section 4. Moreover, we point out that the function of Θ_n is to approximately control \mathbf{u} into the set \mathcal{A} . For each scale, one should choose Θ_n to be a large number to eliminate mesh folding. If this condition is satisfied, Θ_n has no effect on final registration results.

Concerning the existence of solutions for the relaxed model (3.1), one can use the similar idea of Theorem 2.2 in [16] to give a proof. Moreover, one can also use the similar way of Theorem 2.1 in [14] to obtain the following Euler–Lagrange equation of (3.1):

$$(3.2) \quad -\Theta_n \Delta \mathbf{u}^n + \mu \operatorname{div}^{\alpha*}(\nabla^\alpha \mathbf{u}^n) + \mathbf{f}_n(\mathbf{u}^n) = 0,$$

where $\mathbf{f}_n(\mathbf{u}^n) = \lambda_n [T_n(\mathbf{x} + \mathbf{u}^n(\mathbf{x})) - D(\mathbf{x})] \nabla_{\mathbf{u}^n} T_n(\mathbf{x} + \mathbf{u}^n(\mathbf{x}))$ and the definition of $\operatorname{div}^{\alpha*}(\nabla^\alpha)$ can refer to [16].

Remark 3.2. $\mathbf{u} \in [H_0^\alpha(\Omega)]^2$ implies the homogeneous boundary conditions: $\frac{\partial^k u_i(x)}{\partial x_j^k} |_{x \in \partial\Omega} = 0$, ($k = 0, 1, 2, \dots, [\alpha]$; $i, j = 1, 2$) (see [9, 10] for details). By giving homogeneous boundary conditions, the following two properties of fractional derivatives hold [47]:

(i) Riemann–Liouville derivatives, Grunwald–Letnikov derivatives, and Caputo derivatives are equivalent.

(ii) There holds an integration by parts formula

$$(3.3) \quad \int_{a_1}^{b_1} \xi(\mathbf{x}) \cdot \frac{\partial^\alpha f(\mathbf{x})}{\partial x_i^\alpha} dx_i = \int_{a_1}^{b_1} \frac{\partial^{\alpha*} \xi(\mathbf{x})}{\partial x_i^{\alpha*}} \cdot f(\mathbf{x}) dx_i \quad i = 1, 2.$$

Note that these two properties are used for deriving (3.2).

By introducing a time variable t and defining $\mathbf{u}^n(\mathbf{x}) \triangleq \mathbf{u}^n(\mathbf{x}, t)$, we use the following process to update $\mathbf{u}^n(\mathbf{x}, t)$ to the steady state:

$$(3.4) \quad \frac{\partial \mathbf{u}^n}{\partial t} = \Theta_n \Delta \mathbf{u}^n - \mu \operatorname{div}^{\alpha*}(\nabla^\alpha \mathbf{u}^n) - \mathbf{f}_n(\mathbf{u}^n).$$

Remark 3.3. There are two remarks on (3.4):

(i) If $\mathbf{f}_n(\mathbf{u}^n)$ is a linear term on \mathbf{u}^n (i.e., some image denoising problems), one can solve the steady state equation (3.2) directly. Note that here $\mathbf{f}_n(\mathbf{u}^n)$ is a nonlinear term on \mathbf{u}^n with very complex structure. Directly solving the nonlinear steady state equation (3.2) by using the Newton’s method makes the algorithm very sensitive to the initial guess. This makes the algorithm not robust. Therefore, we choose to solve the gradient flow equation (3.4), which ensures that the cost functional $F_n(\mathbf{u}) + \Theta R_1(\mathbf{u})$ decreases with respect to t [14].

(ii) (3.4) is proposed based on a gradient flow approach. Theoretical results show that (3.2) is the steady state of (3.4). One can refer to [14] for details.

Now, we discuss the numerical implementation of (3.4) by using the multigrid method [7, 14].

One can notice that Ω is not necessary to be square in image registration. For the convenience of description for the multigrid method and without loss of generality, we assume that $\Omega \triangleq (a_1, b_1) \times (a_1, b_1)$ throughout this paper.

To numerically solve PDE (3.4), Ω is discretized in the following way. By giving some $N_1, N \in \mathbb{N}^+$, we define $h = \frac{b_1 - a_1}{N_1}$, $\tau = \frac{T}{N}$, $(x_1)_i = a_1 + ih$, $(x_2)_i = a_1 + ih$, $t_m = m\tau$ for $i = 0, 1, 2, \dots, N_1$ and $m = 0, 1, 2, \dots, N$.

Next, we discretize PDE (3.4). Using the finite difference method, $\frac{\partial \mathbf{u}^n}{\partial t}$, $\Delta \mathbf{u}^n$, and $\mathbf{f}_n(\mathbf{u}^n)$ are discretized by the following three formulas:

$$(3.5) \quad \left(\frac{\partial u_p^n}{\partial t}\right)_{i,j}^m \approx \frac{(u_p^n)_{i,j}^m - (u_p^n)_{i,j}^{m-1}}{\tau},$$

$$(3.6) \quad (\Delta u_p^n)_{i,j}^m \approx \frac{(u_p^n)_{i+1,j}^m + (u_p^n)_{i-1,j}^m + (u_p^n)_{i,j+1}^m + (u_p^n)_{i,j-1}^m - 4(u_p^n)_{i,j}^m}{h^2},$$

$$(3.7) \quad \mathbf{f}_n((\mathbf{u}^n)_{i,j}^{m-1}) = \lambda_n [T_{(\mathbf{u}^n)_{i,j}^{m-1}} - D_{i,j}] \partial_{(\mathbf{u}^n)_{i,j}^{m-1}} T_{(\mathbf{u}^n)_{i,j}^{m-1}},$$

where $T_{(\mathbf{u}^n)_{i,j}^{m-1}} \triangleq T(\mathbf{x}_{i,j} + (\mathbf{u}^n)_{i,j}^{m-1})$, $\mathbf{x}_{i,j} = (x_{1,i}, x_{2,j})$, and $p = 1, 2$.

At last, we focus on the numerical discretization of $\operatorname{div}^{\alpha*}(\nabla^\alpha \mathbf{u}^n)$. Before we list the approximation formulas, let’s recall the content on discretization of fractional order derivatives. By using Grunwald approximation [47], the fractional order derivatives $\frac{\partial^\alpha g(\mathbf{x})}{\partial x_i^\alpha}$, $\frac{\partial^{\alpha*} g(\mathbf{x})}{\partial x_i^{\alpha*}}$ ($i = 1, 2$) are discretized by the following two formulas:

$$(3.8) \quad \frac{\partial^\alpha g(\mathbf{x}_{p,q})}{\partial x_1^\alpha} = h^{-\alpha} \sum_{j=0}^{p+1} \rho_j^{(\alpha)} g_{p-j+1,q} + O(h)$$

and

$$(3.9) \quad \frac{\partial^{\alpha*} g(\mathbf{x}_{p,q})}{\partial x_1^{\alpha*}} = h^{-\alpha} \sum_{j=0}^{N_1-p+2} \rho_j^{(\alpha)} g_{p+j-1,q} + O(h),$$

where $\rho_0^{(\alpha)} = 1$, $\rho_j^{(\alpha)} = (1 - \frac{1+\alpha}{j})\rho_{j-1}^{(\alpha)}$ for $j = 1, 2, \dots, N_1$.

Define $U_q = (g_{1,q}, g_{2,q}, \dots, g_{N_1,q})^T$; then by (3.8) and (3.9), we obtain that

$$(3.10) \quad \frac{\partial^\alpha U_q}{\partial x_1^\alpha} \approx B_\alpha U_q, \quad \frac{\partial^{\alpha^*} U_q}{\partial x_1^{\alpha^*}} \approx B_\alpha^T U_q,$$

where

$$B_\alpha = \frac{1}{h^\alpha} \begin{pmatrix} \rho_1^{(\alpha)} & \rho_0^{(\alpha)} & \dots & 0 & 0 \\ \rho_2^{(\alpha)} & \rho_1^{(\alpha)} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho_{N_1-1}^{(\alpha)} & \rho_{N_1-2}^{(\alpha)} & \dots & \rho_1^{(\alpha)} & \rho_0^{(\alpha)} \\ \rho_{N_1}^{(\alpha)} & \rho_{N_1-1}^{(\alpha)} & \dots & \rho_2^{(\alpha)} & \rho_1^{(\alpha)} \end{pmatrix}_{N_1 \times N_1},$$

$$\frac{\partial^\alpha U_q}{\partial x_1^\alpha} = \left(\frac{\partial^\alpha g_{1,q}}{\partial x_1^\alpha}, \frac{\partial^\alpha g_{2,q}}{\partial x_1^\alpha}, \dots, \frac{\partial^\alpha g_{N_1,q}}{\partial x_1^\alpha} \right)^T, \text{ and } \frac{\partial^{\alpha^*} U_q}{\partial x_1^{\alpha^*}} = \left(\frac{\partial^{\alpha^*} g_{1,q}}{\partial x_1^{\alpha^*}}, \frac{\partial^{\alpha^*} g_{2,q}}{\partial x_1^{\alpha^*}}, \dots, \frac{\partial^{\alpha^*} g_{N_1,q}}{\partial x_1^{\alpha^*}} \right)^T.$$

Based on (3.10), we obtain that

$$(3.11) \quad \frac{\partial^{\alpha^*}}{\partial x_1^{\alpha^*}} \left(\frac{\partial^\alpha U_q}{\partial x_1^\alpha} \right) = B_\alpha^T B_\alpha U_q \triangleq A_\alpha U_q.$$

Similarly, we have that

$$(3.12) \quad \frac{\partial^{\alpha^*}}{\partial x_2^{\alpha^*}} \left(\frac{\partial^\alpha V_p}{\partial x_2^\alpha} \right) = B_\alpha^T B_\alpha V_p \triangleq A_\alpha V_p,$$

where $V_p = (g_{p,1}, g_{p,2}, \dots, g_{p,N_1})^T$.

Based on (3.11) and (3.12), we use the following two formulas to discretize the fractional order derivatives $\frac{\partial^{\alpha^*}}{\partial x_p^{\alpha^*}} \left(\frac{\partial^\alpha u_q^n}{\partial x_p^\alpha} \right)$ for $p, q = 1, 2$:

$$(3.13) \quad \frac{\partial^{\alpha^*}}{\partial x_1^{\alpha^*}} \left(\frac{\partial^\alpha u_q^n}{\partial x_1^\alpha} \right)_{i,j} = \sum_{l=1}^{N_1} a_\alpha(i, l) (u_q^n)_{l,j}^m,$$

$$(3.14) \quad \frac{\partial^{\alpha^*}}{\partial x_2^{\alpha^*}} \left(\frac{\partial^\alpha u_q^n}{\partial x_2^\alpha} \right)_{i,j} = \sum_{l=1}^{N_1} a_\alpha(j, l) (u_q^n)_{i,l}^m,$$

where here and in what follows, $a_\alpha(i, j)$ denotes the element on i_{th} row and j_{th} column of matrix A_α .

Combining the discretization results in (3.5)–(3.7) and (3.13)–(3.14), PDE (3.4) is discretized by the following algebraic equations:

$$(3.15) \quad \frac{(u_p^n)_{i,j}^m - (u_p^n)_{i,j}^{m-1}}{\tau} = \Theta_n \frac{(u_p^n)_{i+1,j}^m + (u_p^n)_{i-1,j}^m + (u_p^n)_{i,j+1}^m + (u_p^n)_{i,j-1}^m - 4(u_p^n)_{i,j}^m}{h^2} - \mu \sum_{l=1}^{N_1} a_\alpha(i, l) (u_p^n)_{l,j}^m - \mu \sum_{l=1}^{N_1} a_\alpha(j, l) (u_p^n)_{i,l}^m - f_p((\mathbf{u}^n)_{i,j}^{m-1}).$$

Similarly to [14], we use the following alternating direction implicit (ADI) scheme [15] as 1D solvers to solve (3.15):

$$(3.16) \quad \begin{cases} -\gamma_n (u_p^n)_{i+1,j}^{m-\frac{1}{2}} + [1 + 2\gamma_n + \mu\tau a_\alpha(i,j)] (u_p^n)_{i,j}^{m-\frac{1}{2}} - \gamma_n (u_p^n)_{i-1,j}^{m-\frac{1}{2}} \\ + \mu\tau \sum_{k=1, k \neq i}^{N_1} a_\alpha(i,k) (u_p^n)_{k,j}^{m-\frac{1}{2}} = (b_1^m)_{i,j}^{m-1}, \\ -\gamma_n (u_p^n)_{i,j+1}^m + [1 + 2\gamma_n + \mu\tau a_\alpha(i,j)] (u_p^n)_{i,j}^m - \gamma_n (u_p^n)_{i,j-1}^m \\ + \mu\tau \sum_{k=1, k \neq j}^{N_1} a_\alpha(j,k) (u_p^n)_{i,k}^m = (b_2^m)_{i,j}^{m-\frac{1}{2}}, \end{cases}$$

where $\gamma_n = \frac{\Theta_n \tau}{h^2}$, $(b_1^n)_{i,j}^{m-1} = (u_p^n)_{i,j}^{m-1} + \gamma_n [(u_p^n)_{i,j+1}^{m-1} - 2(u_p^n)_{i,j}^{m-1} + (u_p^n)_{i,j-1}^{m-1}] - \tau f_p((\mathbf{u}^n)_{i,j}^{m-1}) - \mu\tau \sum_{k=1}^{N_1} a_\alpha(j,k) (u_p^n)_{i,k}^{m-1}$, $(b_2^n)_{i,j}^{m-\frac{1}{2}} = (u_p^n)_{i,j}^{m-\frac{1}{2}} + \gamma_n [(u_p^n)_{i+1,j}^{m-\frac{1}{2}} - 2(u_p^n)_{i,j}^{m-\frac{1}{2}} + (u_p^n)_{i-1,j}^{m-\frac{1}{2}}] - \tau f_p((\mathbf{u}^n)_{i,j}^{m-\frac{1}{2}}) - \mu\tau \sum_{k=1}^{N_1} a_\alpha(i,k) (u_p^n)_{k,j}^{m-\frac{1}{2}}$.

At the end of this section, we introduce the numerical implementation of the multigrid technique for solving linear equations (3.16).

For each fixed j , the V-cycle of the first equation of (3.16) contains the following four steps:

Step 1: Smoothing. Here and in what follows, $H = 2h$, and Ω^h, Ω^H denote the fine grid and the coarse grid, respectively. By giving some initial guess on Ω^h and using the iterative formula

$$(3.17) \quad (u_p^n)_{i,j}^{(l+1)} = \frac{(b_1^n)_{i,j}^{m-1} + \gamma_n [(u_p^n)_{i+1,j}^{(l)} + (u_p^n)_{i-1,j}^{(l)}] - \mu\tau \sum_{k=1, k \neq i}^{N_1} a_\alpha(i,k) (u_p^n)_{k,j}^{(l)}}{1 + 2\gamma_n + \mu\tau a_\alpha(i,j)}$$

to relax ν times ($l = 0, 1, 2, \dots, \nu - 1$), we obtain a smooth approximation $\bar{\mathbf{u}}_j^{n,h}$ on Ω^h .

After obtaining the smooth approximation $\bar{\mathbf{u}}_j^{n,h}$, we compute the residual error $\mathbf{r}^{n,h}$ on Ω^h by the following formula:

$$(3.18) \quad (r_p^n)_{i,j}^h = (b_1^n)_{i,j}^{m-1} + \gamma_n [(\bar{u}_p^n)_{i+1,j} + (\bar{u}_p^n)_{i-1,j}] - \mu\tau \sum_{k=1, k \neq i}^{N_1} a_\alpha(i,k) (\bar{u}_p^n)_{k,j} - [1 + 2\gamma_n + \mu\tau a_\alpha(i,j)] (\bar{u}_p^n)_{i,j}.$$

Step 2: Restriction. In order to restrict the residual error $\mathbf{r}^{n,h}$ to the coarse grid Ω^H , we define the 1D restriction operator \mathcal{R}_h^H by

$$(3.19) \quad \mathbf{v}_j^H = \mathcal{R}_h^H \mathbf{v}_j^h,$$

where $v_{i,j}^H = \frac{1}{4}[v_{2i-1,j}^h + 2v_{2i,j}^h + v_{2i+1,j}^h]$.

Computing the residual error $(\mathbf{r}^n)_j^H = \mathcal{R}_h^H(\mathbf{r}^n)_j^h$ on Ω^H , then we relax the 1D solvers

$$(3.20) \quad (u_p^H)_{i,j}^{(l+1)} = \frac{(r_p^n)_{i,j}^H + \gamma_{n,H} [(u_p^{n,H})_{i+1,j}^{(l)} + (u_p^{n,H})_{i-1,j}^{(l)}] - \mu\tau \sum_{k=1, k \neq i}^{N_1} a_\alpha(i,k) (u_p^{n,H})_{k,j}^{(l)}}{1 + 2\gamma_{n,H} + \mu\tau a_\alpha(i,j)}$$

ν times with initial guess $(\mathbf{u}^n)_j^H = \mathbf{0}$ to compute $\bar{\mathbf{u}}_j^{n,H}$ on coarse grid Ω^H . Note that here and in what follows $\gamma_{n,H} = \frac{\Theta_n \tau}{H^2}$.

Based on the restriction and smoothing results, we update the residual error $\mathbf{r}^{n,H}$ on Ω^H by using the following formula:

$$(3.21) \quad (r_p^n)_{i,j}^H = (r_p^n)_{i,j}^H + \gamma_{n,H} \left[(\bar{u}_p^{n,H})_{i+1,j} + (\bar{u}_p^{n,H})_{i-1,j} \right] - \mu\tau \sum_{k=1, k \neq i}^{N_1} a_\alpha(i, k) (\bar{u}_p^{n,H})_{k,j} - [1 + 2\gamma_{n,H} + \mu\tau a_\alpha(i, j)] (\bar{u}_p^{n,H})_{i,j}.$$

Step 3: Coarsest grid solution. On the coarsest grid Ω^H , we accurately solve the linear algebraic system

$$(3.22) \quad -\gamma_{n,H} (u_p^{n,H})_{i+1,j} + [1 + 2\gamma_n + \mu\tau a_\alpha(i, j)] (u_p^{n,H})_{i,j} - \gamma_{n,H} (u_p^{n,H})_{i-1,j} + \mu\tau \sum_{k=1, k \neq i}^{N_M} a_\alpha(i, k) (u_p^{n,H})_{k,j} = (r_p^n)_{i,j}^H,$$

where here and in what follows, N_M denotes the total number of nodes on x_1 direction on the coarsest grid Ω^H .

By solving the linear system (3.22), we obtain an accurate solution $\mathbf{u}^{n,H}$ on the coarsest grid.

Step 4: Interpolation. In this step, the first task is to use $\mathbf{u}^{n,H}$ to correct the former approximations on the finer grid Ω^h . For this purpose, we interpolate the corrections to Ω^h by using the following 1D interpolation:

$$(3.23) \quad \mathbf{u}_j^{n,h} = \mathcal{I}_H^h \mathbf{u}_j^{n,H},$$

where $u_{2i,j}^{n,h} = u_{i,j}^{n,H}$, $u_{2i+1,j}^{n,h} = \frac{1}{2}(u_{i,j}^{n,H} + u_{i+1,j}^{n,H})$.

After interpolating the corrections to the next fine grid Ω^h , we use them to update the current approximations on Ω^h via $\mathbf{u}_j^{n,h} = \bar{\mathbf{u}}_j^{n,h} + \mathbf{u}_j^{n,h}$. Note that here $\bar{\mathbf{u}}_j^{n,h}$ is an approximation obtained by former restriction steps. Followed by this modification, by using the updated $\mathbf{u}_j^{n,h}$ as an initial guess, we relax (3.20) on Ω^h and repeat the interpolation, correction, and smoothing process until the V-cycle reaches the finest grid Ω^h . Finally, relax (3.17) with initial guess $\mathbf{u}_j^{n,h}$ to obtain the final solution for this round of the V-cycle. This completes one round of the V-cycle.

The process for computing $(\mathbf{u}^n)_j^{m-\frac{1}{2}}$ is essentially a series of V-cycles. This process is listed in Algorithm 3.1.

Next, we give the numerical implementation for solving the second equation of (3.16). By using $(\mathbf{u}^n)^{m-\frac{1}{2}}$ as an initial guess, we use the following four similar steps to solve the second equation of (3.16) for each fixed i .

Step 1: Smoothing. By giving some initial guess on Ω^h and using the iterative formula

$$(3.24) \quad (u_p^n)_{i,j}^{(l+1)} = \frac{(b_2^n)_{i,j}^{m-\frac{1}{2}} + \gamma_n \left[(u_p^n)_{i,j+1}^{(l)} + (u_p^n)_{i,j-1}^{(l)} \right] - \mu\tau \sum_{k=1, k \neq j}^{N_1} a_\alpha(j, k) (u_p^n)_{i,k}^{(l)}}{1 + 2\gamma_n + \mu\tau a_\alpha(i, j)}$$

to relax ν times ($l = 0, 1, 2, \dots, \nu - 1$), we obtain a smooth approximation $\bar{\mathbf{u}}_i^{n,h}$ on Ω^h .

Algorithm 3.1. ADI algorithm for system (3.16) on x_1 direction

Initialization: $\mathbf{u}^{n,h} = (\mathbf{u}^n)^{m-1}$, $\mathbf{u}_0^{n,h} = (\mathbf{u}^n)^{m-1} + \Phi$, $\lambda_n, \mu > 0$, $k = 0$, maximum iteration times K and total layer number L for V-cycle. Note that here and in what follows, we give a small distortion Φ to start the algorithm.

while $\|\mathbf{u}^{n,h} - \mathbf{u}_0^{n,h}\| \geq \|\Phi\|$ and $k \leq K$

$\mathbf{u}_0^{n,h} = \mathbf{u}^{n,h}$.

Step 1. By taking $\mathbf{u}_j^{n,h}$ as initial guess, we relax (3.17) to obtain a smooth approximation $\bar{\mathbf{u}}_j^{n,h}$ and compute the residual error $\mathbf{r}_j^{n,h}$ by (3.18); At the end of Step 1, we set $level = L$;

Step 2. Restricting residual to Ω^H by $\mathbf{r}_j^{n,H} = \mathcal{R}_h^H \mathbf{r}_j^{n,h}$. Set $level = level - 1$, relax (3.20) on coarse grid Ω^H with initial guess $\mathbf{u}_j^{n,H} = \mathbf{0}$ to obtain the smooth approximations $\bar{\mathbf{u}}_j^{n,H}$ and update $\mathbf{r}_j^{n,H}$ on Ω^H by (3.21).

Step 3. **If** $level = 1$,

do: accurately solving the linear algebraic system (3.22) to obtain the accurate solution $\mathbf{u}_j^{n,H}$ on coarsest grid Ω^H ;

else

do: repeat **Step 2** until $level = 1$.

endif.

Step 4. **If** $level = L$,

do: relax (3.17) to obtain the final solution $\mathbf{u}_j^{n,h}$ for this round of V-cycle and set $k = k + 1$;

else

repeat: (i) interpolate the correction to next finer grid by using $\mathbf{u}_{t,j}^{n,h} = \mathcal{I}_H^h \mathbf{u}_j^{n,H}$;

(ii) update current grid approximations via correction $\hat{\mathbf{u}}_j^{n,h} = \mathbf{u}_{t,j}^{n,h} + \bar{\mathbf{u}}_j^{n,h}$;

(iii) relax (3.20) with initial guess $\hat{\mathbf{u}}_j^{n,h}$ on fine grid Ω^h to obtain approximations $\mathbf{u}_j^{n,h}$ and set $level = level + 1$.

Repeat interpolation, correction, and smoothing process (i)–(iii) until $level = L$.

endif.

endwhile

Output: $(\mathbf{u}^n)_j^{m-\frac{1}{2}} = \mathbf{u}_j^{n,h}$.

After obtaining the smooth approximation $\bar{\mathbf{u}}_i^{n,h}$, we compute the residual error $\mathbf{r}_i^{n,h}$ on Ω^h by the following formula:

$$(3.25) \quad (r_p^n)_{i,j}^h = (b_2^n)_{i,j}^{m-\frac{1}{2}} + \gamma_n \left[(\bar{u}_p^n)_{i,j+1} + (\bar{u}_p^n)_{i,j-1} \right] - \mu\tau \sum_{k=1, k \neq j}^{N_1} a_\alpha(i, k) (\bar{u}_p^n)_{i,k} - [1 + 2\gamma_n + \mu\tau a_\alpha(i, j)] (\bar{u}_p^n)_{i,j}.$$

Step 2: Restriction. In order to restrict the residual error $\mathbf{r}^{n,h}$ to coarse grid Ω^H , we define the 1D restriction operator \mathfrak{R}_h^H by

$$(3.26) \quad \mathbf{v}_i^H = \mathfrak{R}_h^H \mathbf{v}_i^h,$$

where $v_{i,j}^H = \frac{1}{4}[v_{i,2j-1}^h + 2v_{i,2j}^h + v_{i,2j+1}^h]$.

Using 1D restriction operator \mathfrak{R}_h^H and computing the residual error $\mathbf{r}_i^{n,H} = \mathfrak{R}_h^H \mathbf{r}_i^{n,h}$ on Ω^H , then we relax the 1D solvers

(3.27)

$$(u_m^H)^{(l+1)}_{i,j} = \frac{(r_p^n)^H_{i,j} + \gamma_n \left[(u_p^{n,H})^{(l)}_{i,j+1} + (u_p^{n,H})^{(l)}_{i,j-1} \right] - \mu\tau \sum_{k=1, k \neq j}^{N_1} a_\alpha(j, k) (u_p^{n,H})^{(l)}_{i,k}}{1 + 2\gamma_{n,H} + \mu\tau a_\alpha(i, j)}$$

ν times with initial guess $\mathbf{u}_i^{n,H} = \mathbf{0}$ to compute $\bar{\mathbf{u}}_i^{n,H}$ on coarse grid Ω^H .

Based on the restriction and smoothing results, we update the residual error $\mathbf{r}^{n,H}$ on Ω^H by the following formulas:

(3.28)

$$(r_p^n)^H_{i,j} = (r_p)^H_{i,j} + \gamma_{n,H} \left[(\bar{u}_p^{n,H})_{i,j+1} + (\bar{u}_p^{n,H})_{i,j-1} \right] - \mu\tau \sum_{k=1, k \neq j}^{N_1} a_\alpha(j, k) (\bar{u}_p^{n,H})_{i,k} \\ - [1 + 2\gamma_{n,H} + \mu\tau a_\alpha(i, j)] (\bar{u}_p^{n,H})_{i,j}.$$

Step 3: Coarsest grid solution. On coarsest grid Ω^H , we accurately solve the following linear algebraic system:

$$(3.29) \quad -\gamma_{n,H} (u_p^{n,H})_{i,j+1} + [1 + 2\gamma_n + \mu\tau a_\alpha(i, j)] (u_p^{n,H})_{i,j} - \gamma_{n,H} (u_p^{n,H})_{i,j-1} \\ + \mu\tau \sum_{k=1, k \neq j}^{N_M} a_\alpha(j, k) (u_p^{n,H})_{i,k} = (r_p^n)^H_{i,j}.$$

By solving the linear systems (3.29), we obtain an accurate solution $\mathbf{u}^{n,H}$ on the coarsest grid.

Step 4: Interpolation. In this step, the first task is to use $\mathbf{u}_i^{n,H}$ to correct the former approximations on the finer grid Ω^h . For this purpose, we interpolate the corrections by using the following 1D interpolation:

$$(3.30) \quad \mathbf{u}_i^{n,h} = \mathcal{J}_H^h \mathbf{u}_i^{n,H},$$

where $u_{i,2j}^{n,h} = u_{i,j}^{n,H}$, $u_{i,2j+1}^{n,h} = \frac{1}{2}(u_{i,j}^{n,H} + u_{i,j+1}^{n,H})$.

After interpolating the correction to the next fine grid Ω^h , we use them to update the current approximations on Ω^h via $\mathbf{u}_i^{n,h} = \bar{\mathbf{u}}_i^{n,h} + \mathbf{u}_i^{n,h}$. Note that here $\bar{\mathbf{u}}_i^{n,h}$ is an approximation obtained by former restriction steps. Followed by this modification, by using the updated $\mathbf{u}_i^{n,h}$ as an initial guess, we relax (3.27) ν times and repeat the interpolation, correction, and smoothing process until the V-cycle reaches the finest grid Ω^h . Finally, we relax (3.24) with initial guess $\mathbf{u}_i^{n,h}$ to obtain the final solution for this round of the V-cycle.

Based on the above discussion, the algorithm for computing $(\mathbf{u}^n)_i^m$ is summarized as Algorithm 3.2.

Furthermore, the 1D smoother fast fractional-order image registration algorithm is summarized as Algorithm 3.3.

Concerning the convergence of Algorithm 3.3, one can use a similar idea of [14] to give some analysis.

By multiscale approach (2.1)–(2.3), the numerical implementation of multiscale approach (2.1)–(2.3) is summarized as the multiscale 2D fractional-order diffeomorphic image registration algorithm. One can see Algorithm 3.4 for details.

Algorithm 3.2. ADI algorithm for system (3.16) on x_2 direction

Initialization: $\mathbf{u}^{n,h} = \mathbf{u}^{m-\frac{1}{2}}$, $\mathbf{u}_0^{n,h} = \mathbf{u}^{m-\frac{1}{2}} + \Phi$, $\lambda_n, \mu > 0$, $k = 0$, maximum iteration times K , and total layer number L for V-cycle.

while $\|\mathbf{u}^{n,h} - \mathbf{u}_0^{n,h}\| \geq \|\Phi\|$ and $k \leq K$

$\mathbf{u}_0^{n,h} = \mathbf{u}^{n,h}$.

Step 1. By taking $\mathbf{u}_i^{n,h}$ as an initial guess, we relax (3.24) to obtain a smooth approximation $\tilde{\mathbf{u}}_i^{n,h}$ and compute the residual error $\mathbf{r}_i^{n,h}$ by (3.25); At the end of Step 1, we set $level = L$;

Step 2. Restricting residual to Ω^H by $\mathbf{r}_i^{n,H} = \mathfrak{R}_h^H \mathbf{r}_i^{n,h}$; Set $level = level - 1$, relax (3.27) on coarse grid with initial guess $\mathbf{u}_i^{n,H} = \mathbf{0}$ to obtain approximations $\tilde{\mathbf{u}}_i^{n,H}$ and update $(\mathbf{r}^n)_i^H$ on Ω^H by (3.28).

Step 3. **If** $level = 1$,

do: accurately solving the linear algebraic system (3.29) to obtain the accurate solution $\mathbf{u}_i^{n,H}$ on coarsest grid Ω^H ;

else

do: repeat **Step 2** until $level = 1$.

endif

Step 4. **If** $level = L$,

do: relax (3.24) to obtain the final solution $\mathbf{u}_i^{n,h}$ for this round of V-cycle and let $k = k + 1$;

else

do(repeat): (i) interpolate the correction to next finer grid by using $\mathbf{u}_{t,i}^{n,h} = \mathfrak{I}_H^h \mathbf{u}_i^{n,H}$;

(ii) update current grid approximations using correction $\hat{\mathbf{u}}_i^{n,h} = \mathbf{u}_{t,i}^{n,h} + \tilde{\mathbf{u}}_i^{n,h}$;

(iii) relax (3.27) with initial guess $\hat{\mathbf{u}}_i^{n,h}$ on fine grid to obtain approximations $\mathbf{u}_i^{n,h}$ and set $level = level + 1$.

Repeat interpolation, correction, and smoothing process (i)-(iii) until $level = L$.

endif

endwhile

Output: $(\mathbf{u}^n)_i^m = \mathbf{u}_i^{n,h}$.

Algorithm 3.3. 1D smoother fast fractional-order diffeomorphic image registration algorithm

Initialization: Given accuracy, initial error $E = 1$, $m = 1$, $\lambda_n, \varepsilon_n, \mu$, and maximum iteration times K .

while $E > \text{accuracy}$ and $m \leq K$

1. Use **Algorithm 3.1** to obtain $(\mathbf{u}^n)_j^{m-\frac{1}{2}}$ for $j = 1, 2, \dots, N_1 - 1$;

2. Based on $(\mathbf{u}^n)^{m-\frac{1}{2}}$ obtained by **Algorithm 3.1**, we compute $(\mathbf{u}^n)_i^m$ for $i = 1, 2, \dots, N_1 - 1$ via **Algorithm 3.2**;

3. Compute $T(\mathbf{x} + (\mathbf{u}^n)^m(\mathbf{x}))$, $\mathbf{f}((\mathbf{u}^n)^m)$, registration error $E = \frac{\|T(\mathbf{x}+(\mathbf{u}^n)^m) - D(\mathbf{x})\|_{L^2(\Omega)}^2}{\|T(\mathbf{x}) - D(\mathbf{x})\|_{L^2(\Omega)}^2}$ and set $m = m + 1$;

endwhile

Output: $T(\mathbf{x} + (\mathbf{u}^n)^m(\mathbf{x}))$ for some $m \leq K$.

Algorithm 3.4. Multiscale 2D fractional-order diffeomorphic image registration algorithm

Initialization: Given maximum scale number K_M and μ .

for scale = 1 : K_M

1. Let $\mathbf{u} = \mathbf{0}$ and update λ_n, ε_n to satisfy the conditions in Theorem 2.3;

2. Use **Algorithm 3.3** to obtain $\mathbf{u}_{i,j}^{\text{scale}}$ for $i, j = 1, 2, \dots, N_1 - 1$;

3. Compute $T_{\text{scale}}(\cdot) = T \circ \tilde{\mathbf{h}}^{\text{scale}}(\cdot)$, where $\mathbf{h}_{i,j}^{\text{scale}} = \mathbf{x}_{i,j} + \mathbf{u}_{i,j}^{\text{scale}}$ and $\tilde{\mathbf{h}}^{\text{scale}} = \tilde{\mathbf{h}}^{\text{scale}-1} \circ \mathbf{h}^{\text{scale}}$.

endfor

Output: $T \circ \tilde{\mathbf{h}}^{K_M}(\cdot)$.

4. Numerical tests. In this section, we perform several numerical tests to show the efficiency of Algorithm 3.4. The used data sets contain synthetic images, natural images, underwater distorted images, and medical images. For quantitative evaluation, we use the following two indexes:

- relative sum of squared difference ($Re - SSD$ for short) which is defined by

$$(4.1) \quad Re - SSD(T, D, \mathbf{u}) = \frac{SSD(T(\mathbf{x} + \mathbf{u}), D)}{SSD(T, D)},$$

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

- mesh folding number (MFN) which is defined by

$$(4.2) \quad MFN(\mathbf{u}) = \#(\det \bar{J}(\mathbf{u}) \leq 0),$$

where $\det \bar{J}(\mathbf{u}) = (1 + \frac{\partial u_1}{\partial x_1})(1 + \frac{\partial u_2}{\partial x_2}) - \frac{\partial u_1}{\partial x_2} \frac{\partial u_2}{\partial x_1}$ and for any set A , $\#(A)$ denotes the number of elements in A .

4.1. Test 1. In this test, we use four different synthetic image pairs ($A - A$, $A - R$, $C - E$, $C - C$) to perform the numerical test. This test contains two parts. In part 1, we use Algorithm 3.4 to perform the registration on these four synthetic image pairs, respectively. The simulation results are shown as Figures 1 through 8. It follows from Figures 1–8 that all $S(\mathbf{u})$ for four data pairs decrease with respect to scale number. Especially, for the former three data sets, $Re - SSD$ finally achieves 0.0584%, 0.0918%, and 0.0231%, respectively. This implies $T \circ \tilde{\mathbf{h}}^{K_M}(\cdot) \approx D(\cdot)$, which achieves the ultimate goal for diffeomorphic image registration. This numerical result validates the theoretical results in section 2 and shows the efficiency of Algorithm 3.4. In addition, for the $C - C$ image pair with large deformation, though the CPU consumption is much larger, the proposed Algorithm 3.4 also achieves a very good registration result (see Figures 7–8 for details). This validates the statement in Remark 2.1 that the proposed multiapproach can deal with large deformation registration. In addition, for the $C - C$ image pair, one can notice from Table 1 that model (1.4) based algorithm DFIRA [16] achieves only a local minimum while the multiscale approach based Algorithm 3.4 achieves a very good registration result. This validates the theoretical comparison in remark 2.4. In part 2, a quantitative comparison is performed by using five different image registration algorithms: the diffeomorphic log demons image registration (DLDIR) algorithm [11, 38], the A4 algorithm [14], DFIRA [16], the LDDMM-demons algorithm in [22], and Algorithm 3.4, where the open code of the DLDIR algorithm can be downloaded from [11] and one can see the detail of the LDDMM-demons algorithm in Algorithm 1 of [22]. The quantitative comparison result is shown in Table 1. It follows from Table 1 that Algorithm 3.4 is competitive to another four algorithms. This shows the advantage of Algorithm 3.4.

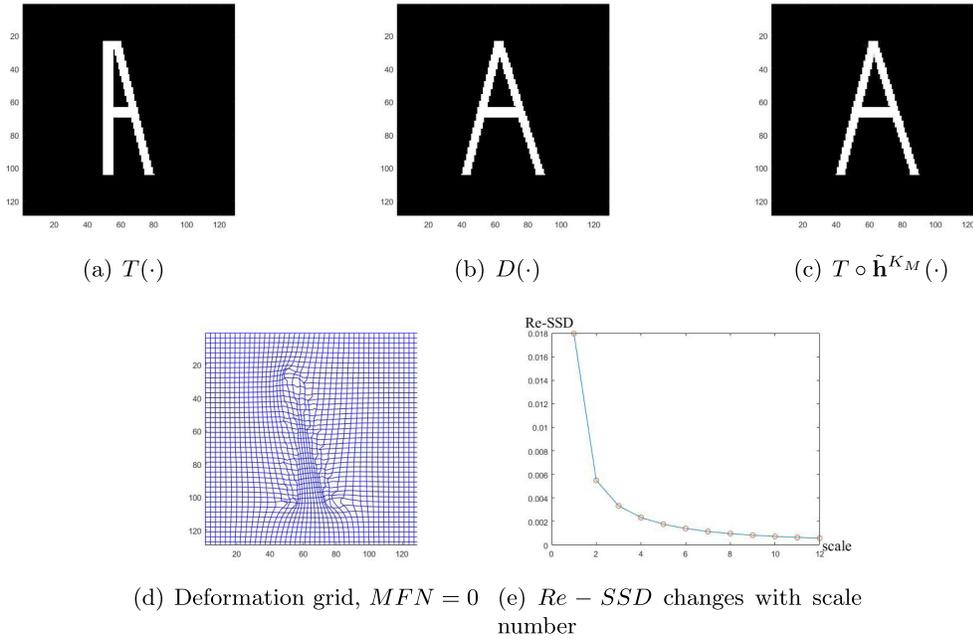


FIG. 1. Registration result of Algorithm 3.4 on image pairs $A - A$.

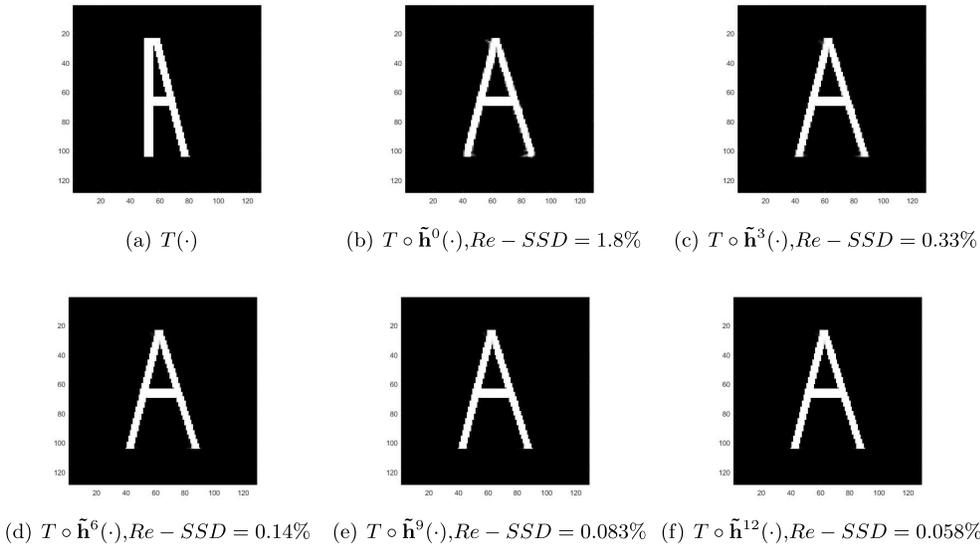
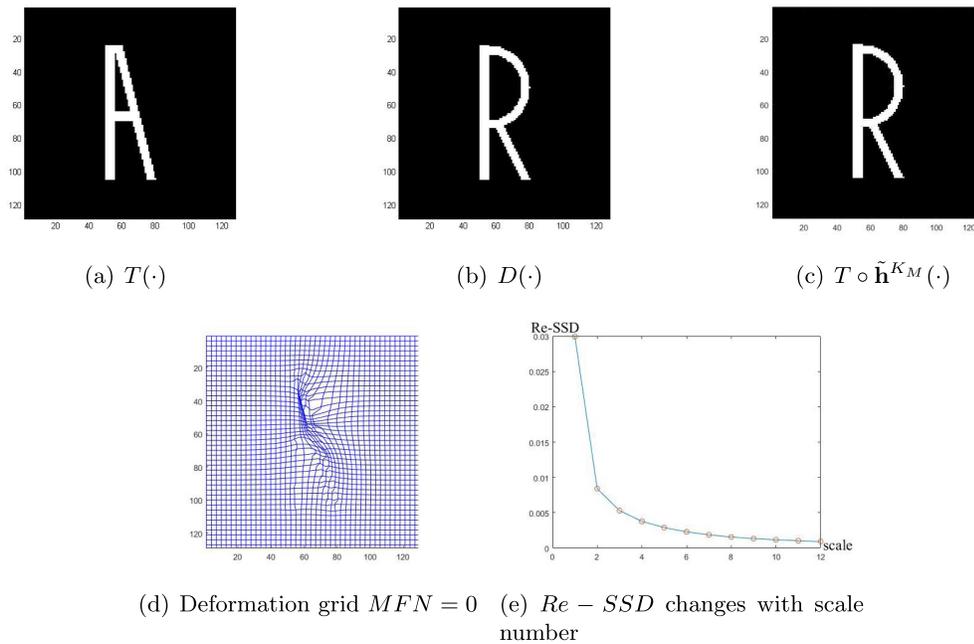
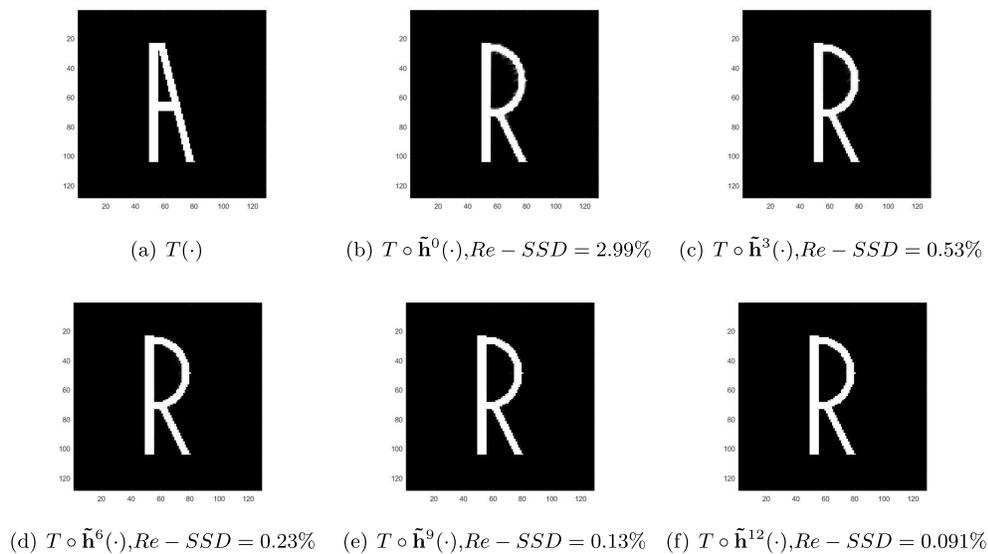


FIG. 2. Registration result of different scale on image pairs $A - A$.

4.2. Test 2. This test contains two parts. In the first part of this test, we use Algorithm 3.4 to perform image registration on two natural image pairs (Watermelon and Pineapple-Pepper) and two underwater distorted image pairs (Square and Fonts). Note that these two underwater image pairs can be downloaded from [42]. Quantitative evaluation and final registration results are listed in Figures 9 through 16.

FIG. 3. Registration result of Algorithm 3.4 on image pairs $A - R$.FIG. 4. Registration result of different scale on image pairs $A - R$.

In addition, we also plot the relationship between $Re - SSD$ and scale number. Moreover, the final deformation grid for Algorithm 3.4 is also given in Figures 9–16. In the second part, we use these four image pairs to perform a quantitative comparison between five different algorithms: Algorithm 3.4, the DDIR algorithm [11, 38],

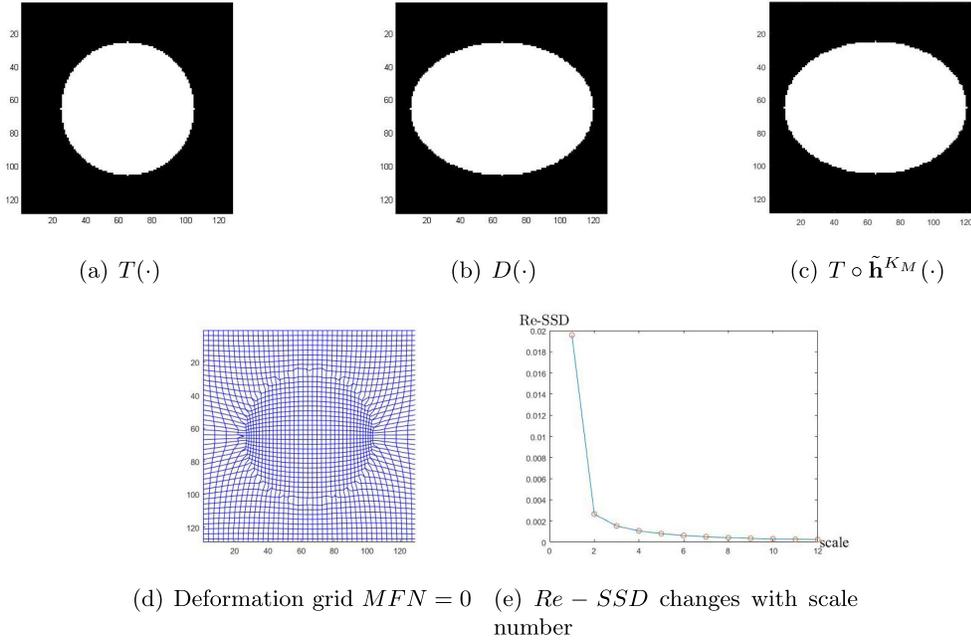


FIG. 5. Registration result of Algorithm 3.4 on image pairs $C - E$.

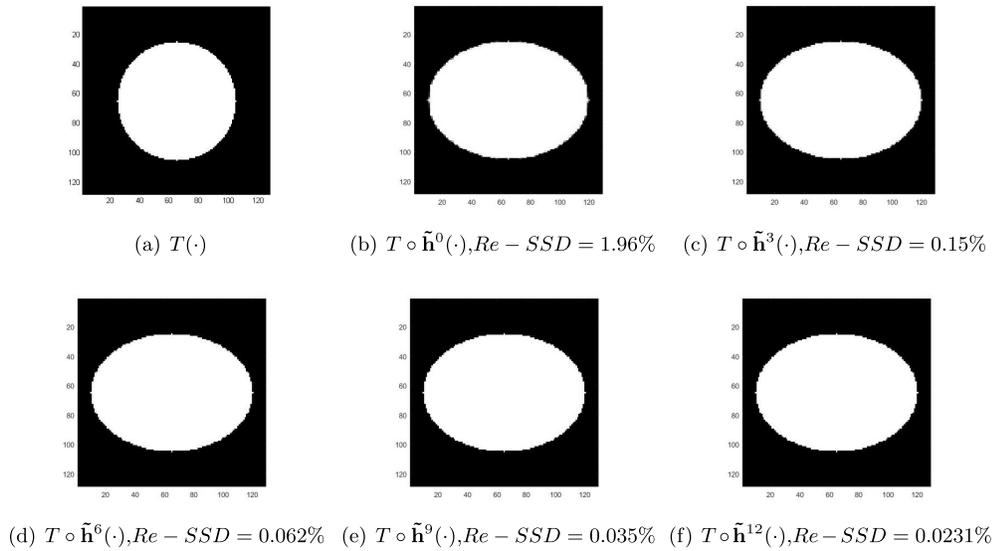


FIG. 6. Registration result of different scale on image pairs $C - E$.

the A4 algorithm [14], the LDDMM-demons algorithm [22], and DFIRA [16]. The comparison results are shown in Table 2. By Table 2, we know that Algorithm 3.4 is competitive to another four algorithms. This shows that Algorithm 3.4 is effective for natural images and underwater images.

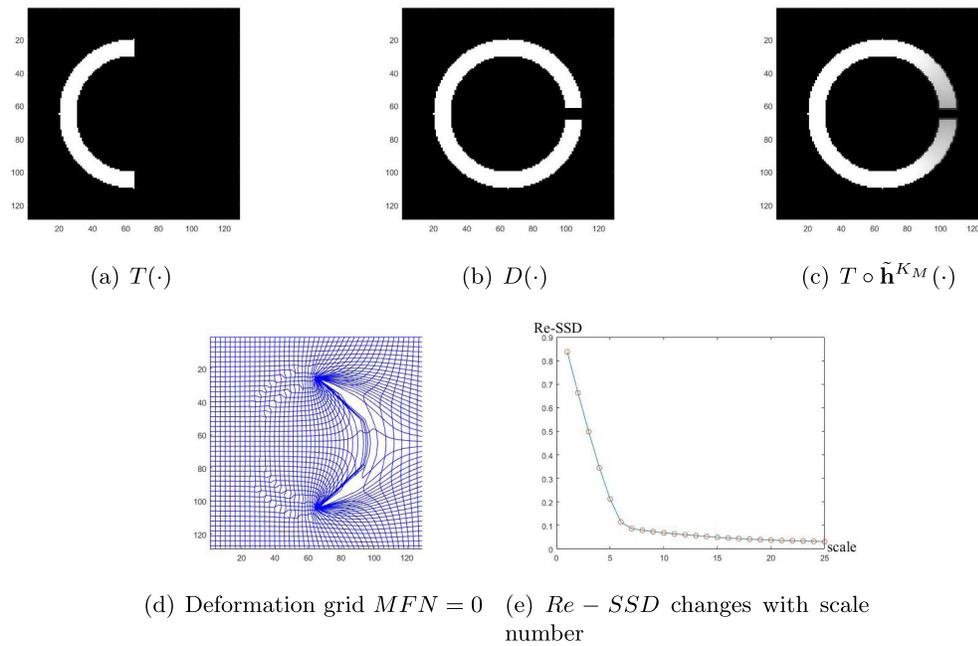


FIG. 7. Registration result of Algorithm 3.4 on image pairs $C - C$.

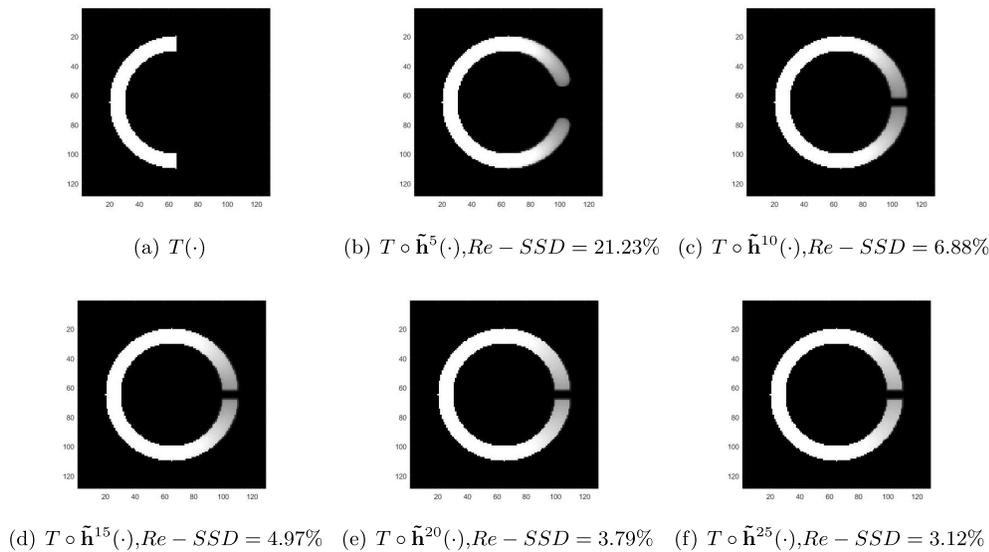


FIG. 8. Registration result of different scale on image pairs $C - C$.

4.3. Test 3. In this test, the registration is performed on four different medical image pairs: liver, brain, chest, and hand. In the first part of this test, we use Algorithm 3.4 to perform image registration on these four medical image pairs. Quantitative evaluation and final registration results are listed in Figures 17 through 24. In the

TABLE 1

Quantitative comparison between registration results of five different algorithms (synthetic images).

Data	Algorithm	Re-SSD(%)	MFN	CPU/s
A-A	Proposed	0.058	0	681.3
	DFIRA	3.80	0	597.3
	DLDIR	54.47	0	3.9
	A4	3.02	0	69.4
	LDDMM	7.48	0	100.45
A-R	Proposed	0.091	0	587.6
	DFIRA	7.13	0	569.2
	DLDIR	78.3	0	3.3
	A4	4.38	0	62.3
	LDDMM	11.23	0	100.17
C-E	Proposed	0.023	0	223.1
	DFIRA	2.5	0	323.5
	DLDIR	45.77	0	3.2
	A4	2.84	0	24.6
	LDDMM	3.15	0	101.8
C-C	Proposed	3.12	0	863.3
	DFIRA	13.38	0	617.7
	DLDIR	98.83	0	2.3
	A4	18.61	0	391.2
	LDDMM	9.33	0	50.1

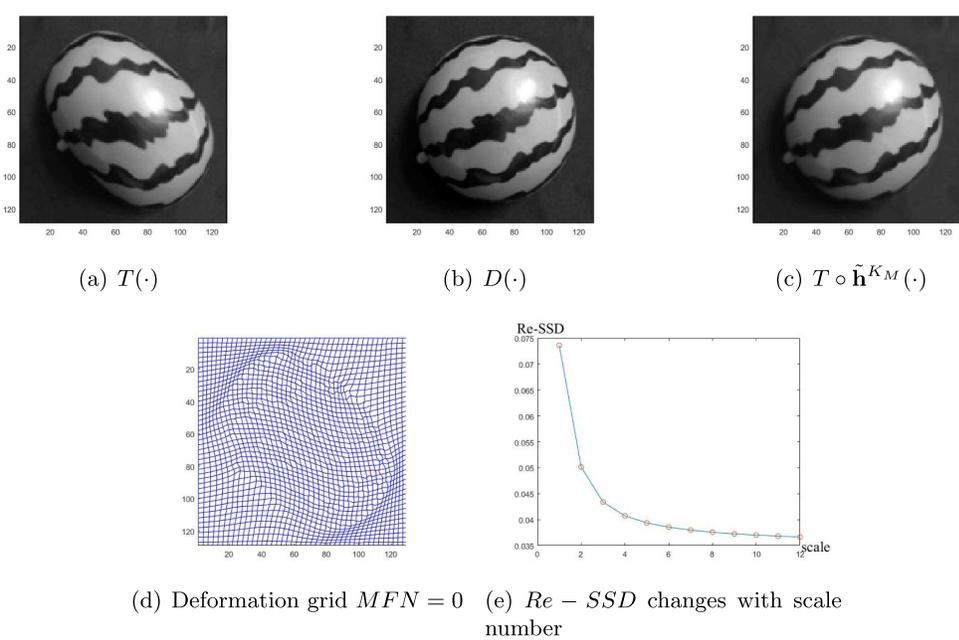


FIG. 9. Registration result of Algorithm 3.4 on image pairs Watermelon.

second part, we use these four medical image pairs to perform a quantitative comparison between five different algorithms: Algorithm 3.4, the DLDIR algorithm [11, 38], the A4 algorithm [14], the LDDMM-demons algorithm [22], and DFIRA [16]. The comparison results are shown in Table 3. By Table 3, we know that Algorithm 3.4 is competitive to another four algorithms. This shows the efficiency of Algorithm 3.4.

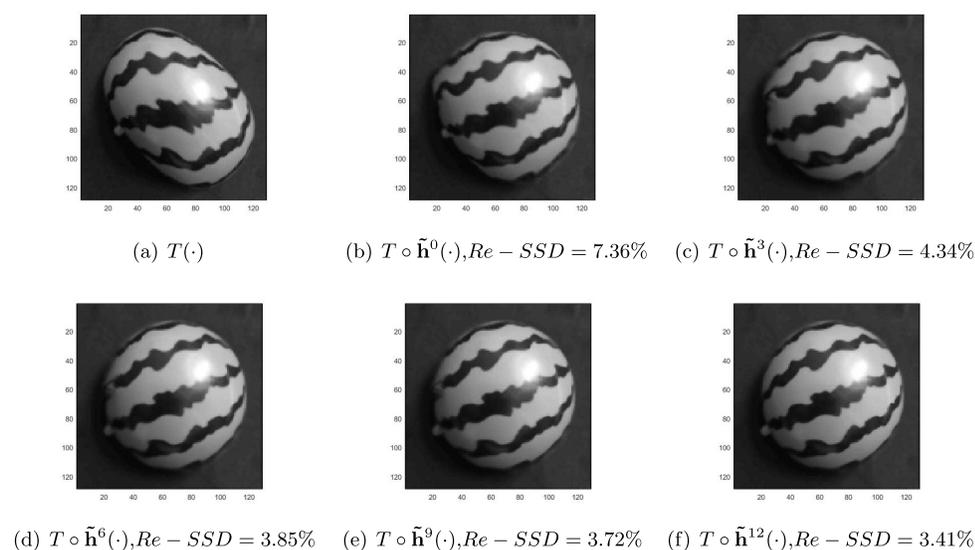


FIG. 10. Registration result of different scale on image pairs Watermelon.

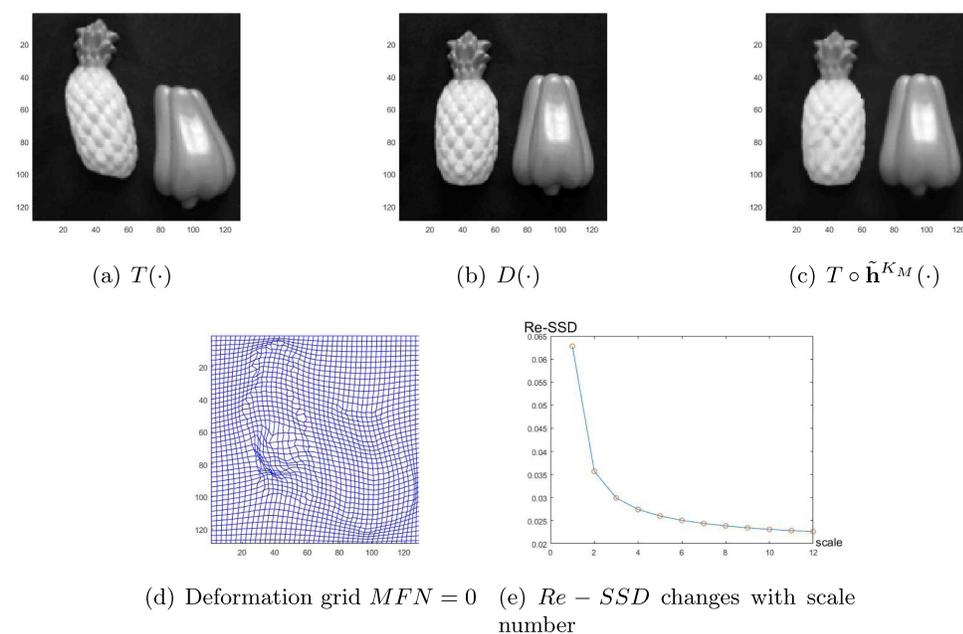


FIG. 11. Registration result of Algorithm 3.4 on image pairs Pineapple-Pepper.

5. Conclusion. In this paper, we propose a multiscale image registration approach for 2D diffeomorphic image registration. This approach achieves an optimal solution of 2D diffeomorphic image registration. Numerical tests show that the proposed model achieves a satisfactory image registration result and effectively eliminates mesh folding. For future research, we may focus on extending this multiscale approach into 3D diffeomorphic image registration.

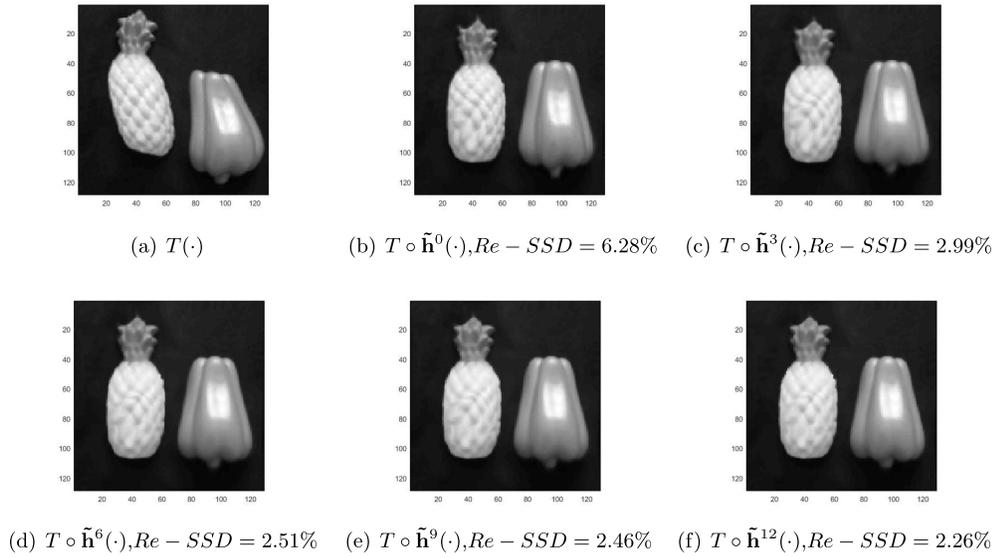


FIG. 12. Registration result of different scale on image pairs Pineapple-Pepper.

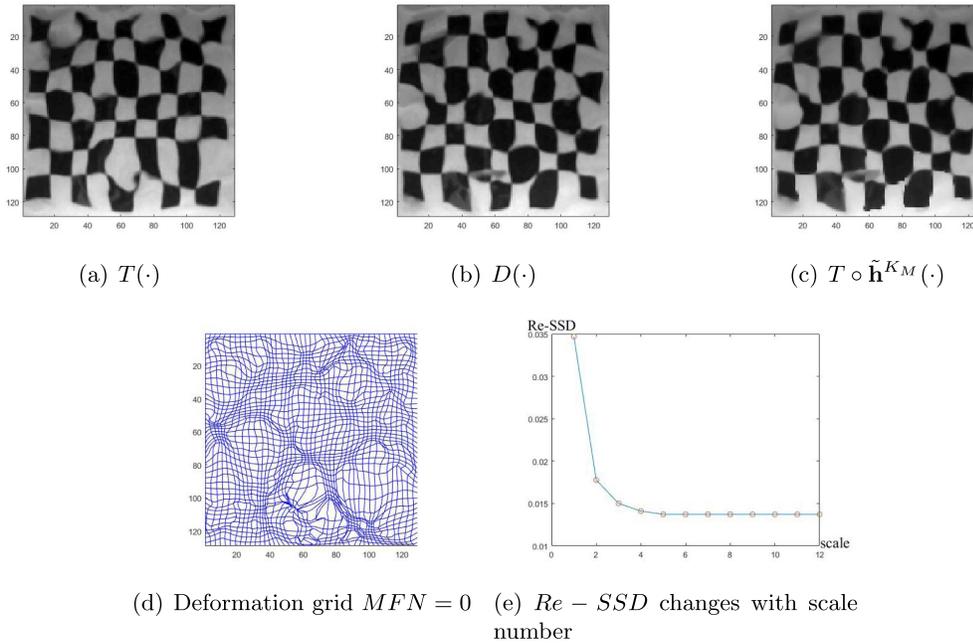


FIG. 13. Registration result of Algorithm 3.4 on Underwater square pairs.

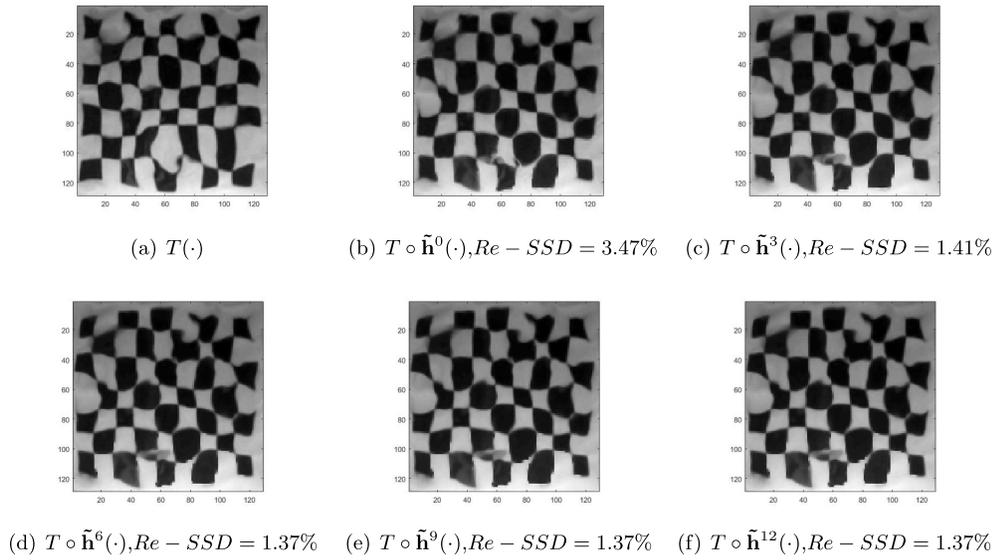


FIG. 14. Registration result of different scale on Underwater square pairs.

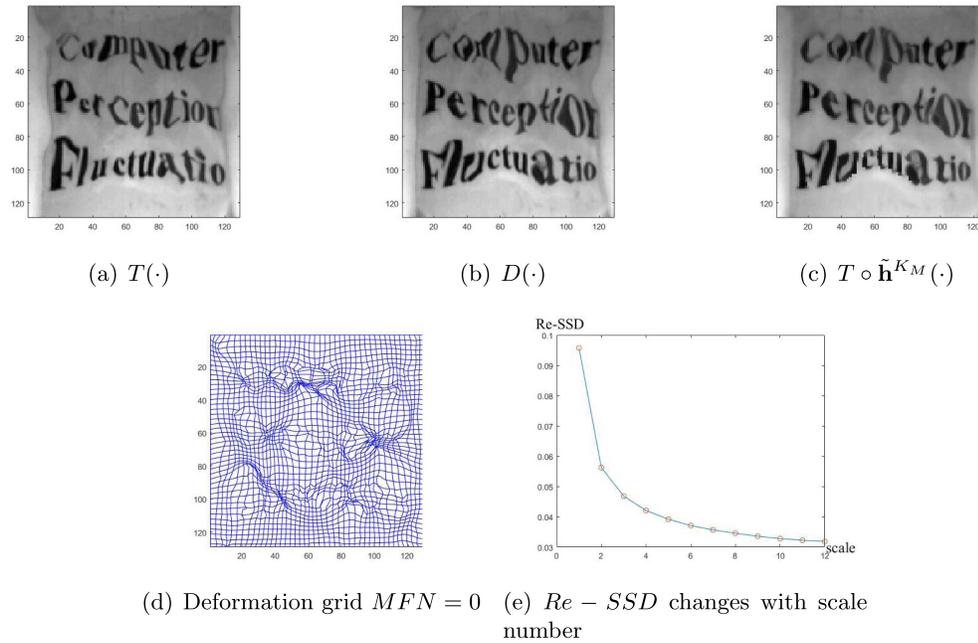


FIG. 15. Registration result of Algorithm 3.4 on Underwater font pairs.

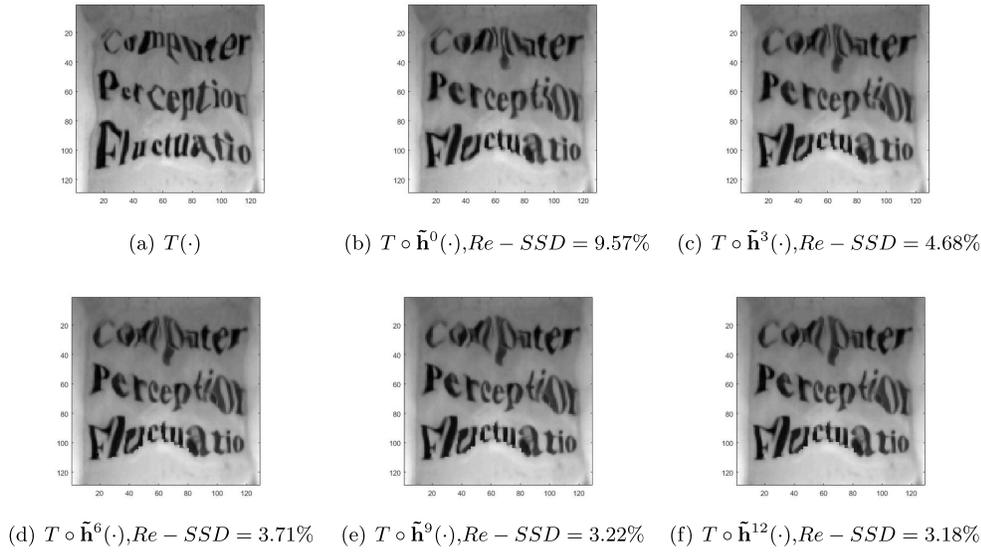


FIG. 16. Registration result of different scale on Underwater square pairs.

TABLE 2

Quantitative comparison between registration results of five different algorithms (natural images).

Data	Algorithm	$Re - SSD(\%)$	MFN	CPU/s
Watermelon	Proposed	3.41	0	365.1
	DLDIR	3.58	0	81.3
	DIDRT	8.57	0	31.7
	A4	12.33	0	38.1
	LDDMM	9.55	0	120.5
Pineapple-Pepper	Proposed	2.26	0	363.8
	DFIRA	2.64	0	80.5
	DLDIR	9.55	0	21.2
	A4	6.99	0	38.3
	LDDMM	8.38	0	120.8
Square	Proposed	1.37	0	285.9
	DFIRA	3.00	0	613.9
	DLDIR	43.27	0	12.6
	A4	2.62	0	24.9
	LDDMM	8.86	0	20.6
Fonts	Proposed	3.18	0	295.3
	DFIRA	8.67	0	651.7
	DLDIR	75.44	0	3.6
	A4	9.57	0	24.6
	LDDMM	13.33	0	79.1

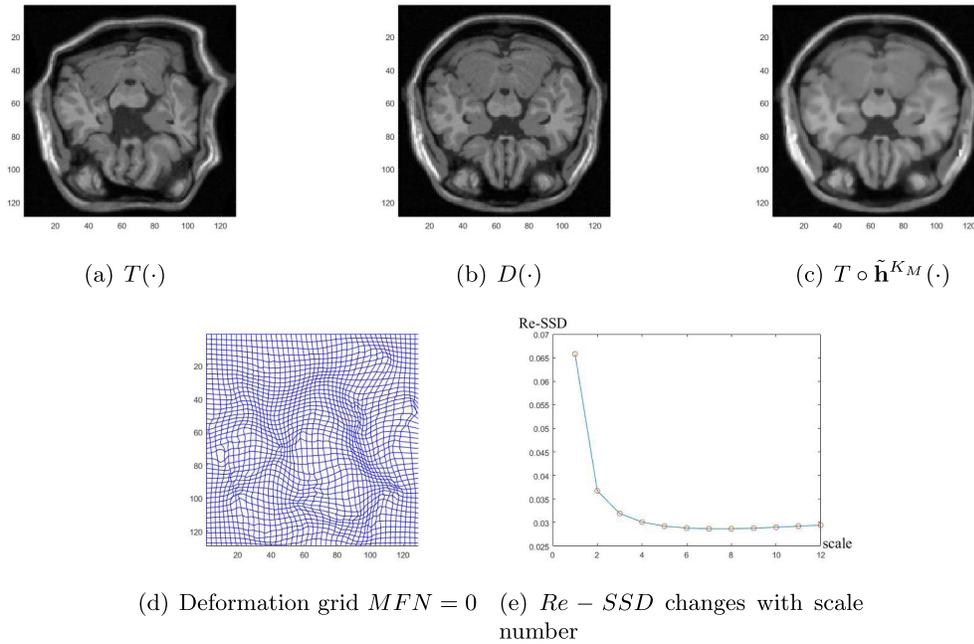


FIG. 17. Registration result of Algorithm 3.4 on image pairs Brain.

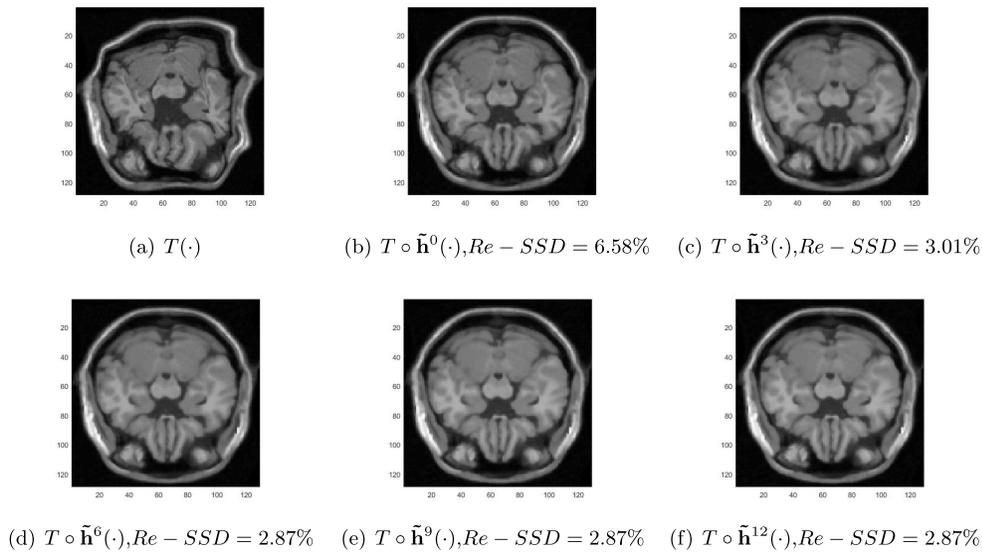


FIG. 18. Registration result of different scale on image pairs Brain.

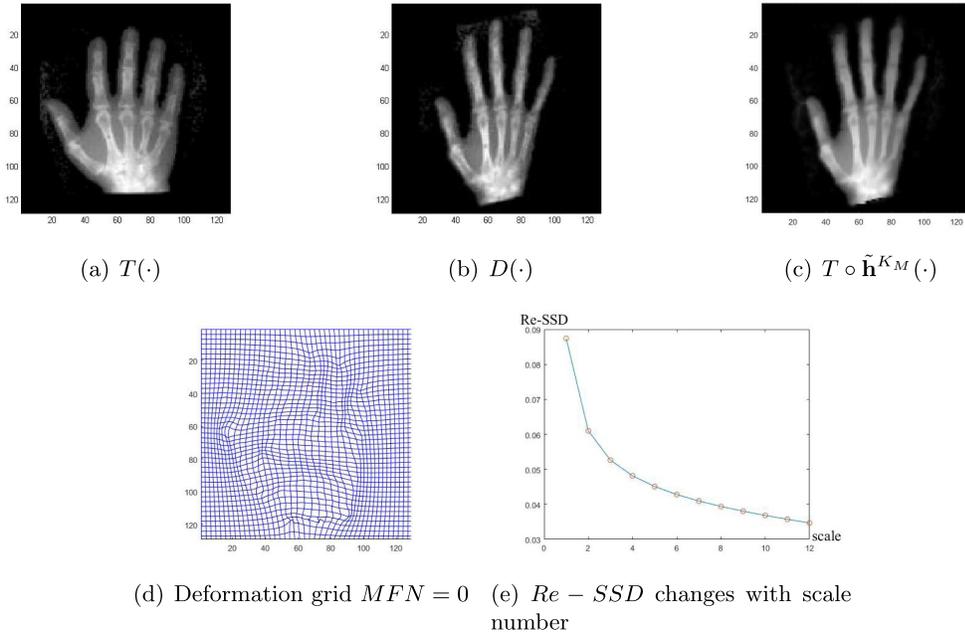


FIG. 19. Registration result of Algorithm 3.4 on image pairs Hand.

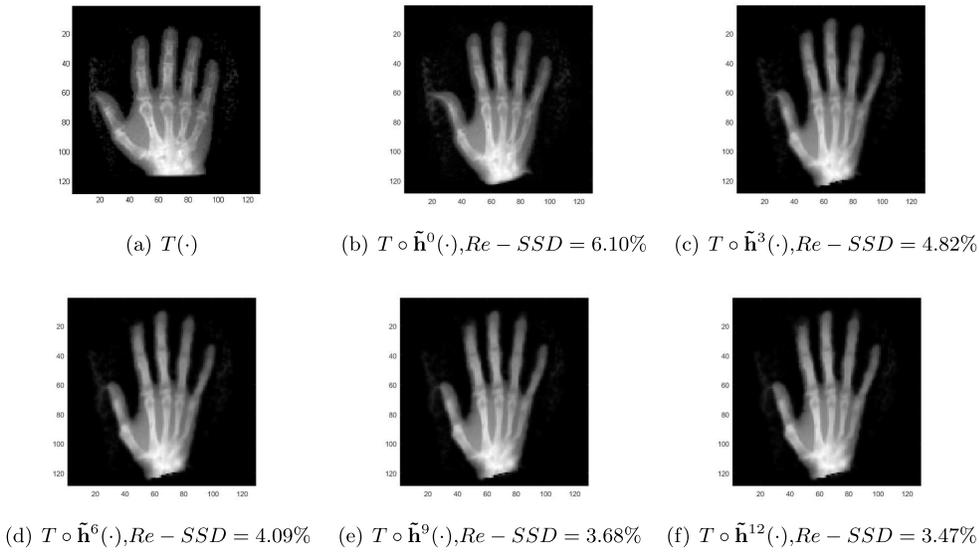


FIG. 20. Registration result of different scale on image pairs Hand.

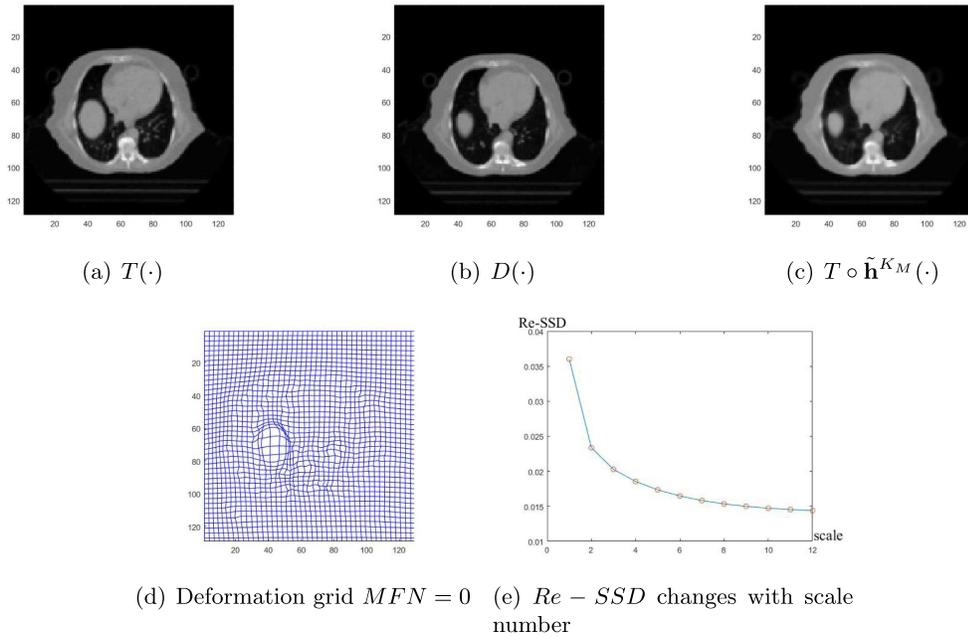


FIG. 21. Registration result of Algorithm 3.4 on image pairs Chest.

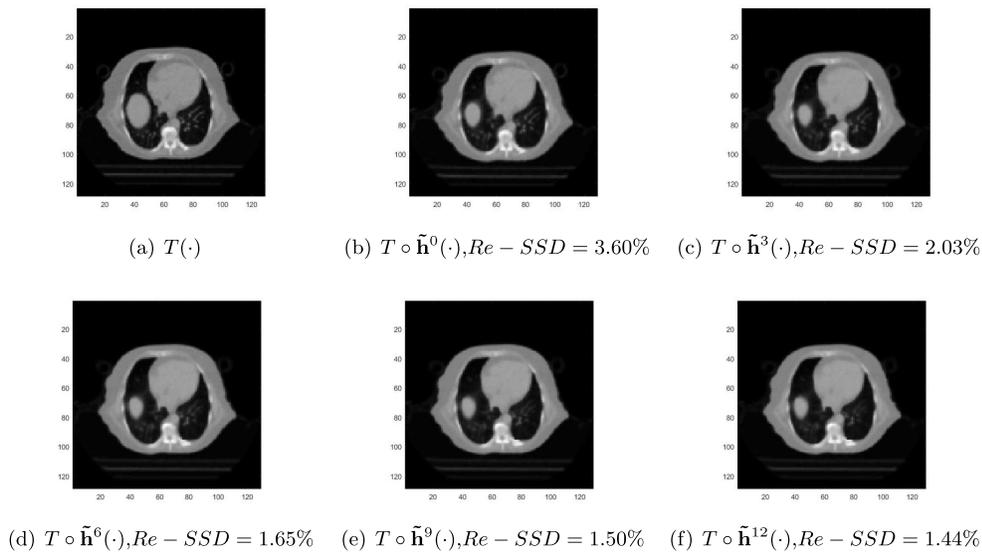


FIG. 22. Registration result of different scale on image pairs Chest.

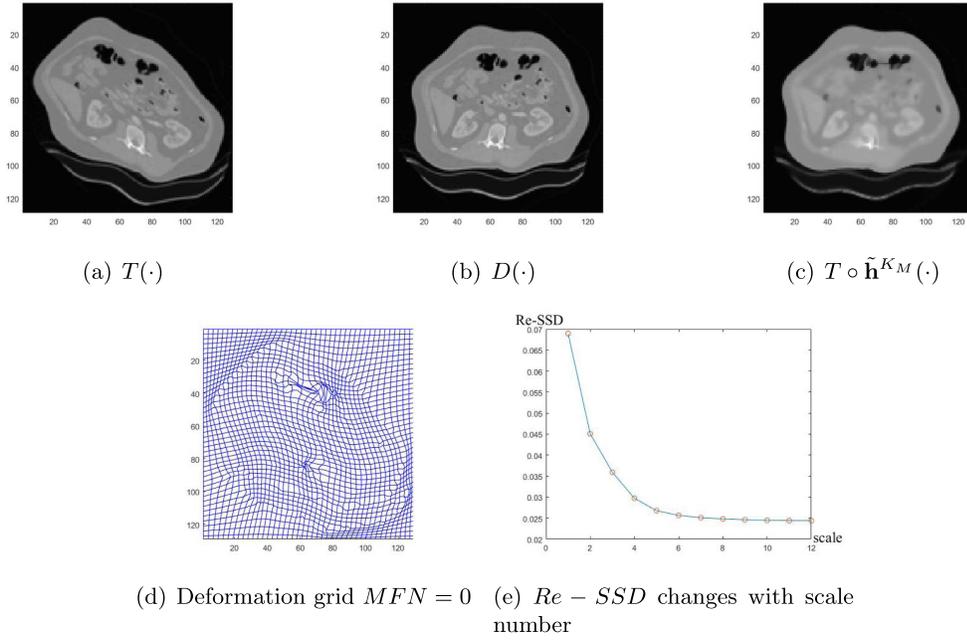


FIG. 23. Registration result of Algorithm 3.4 on image pairs Liver.

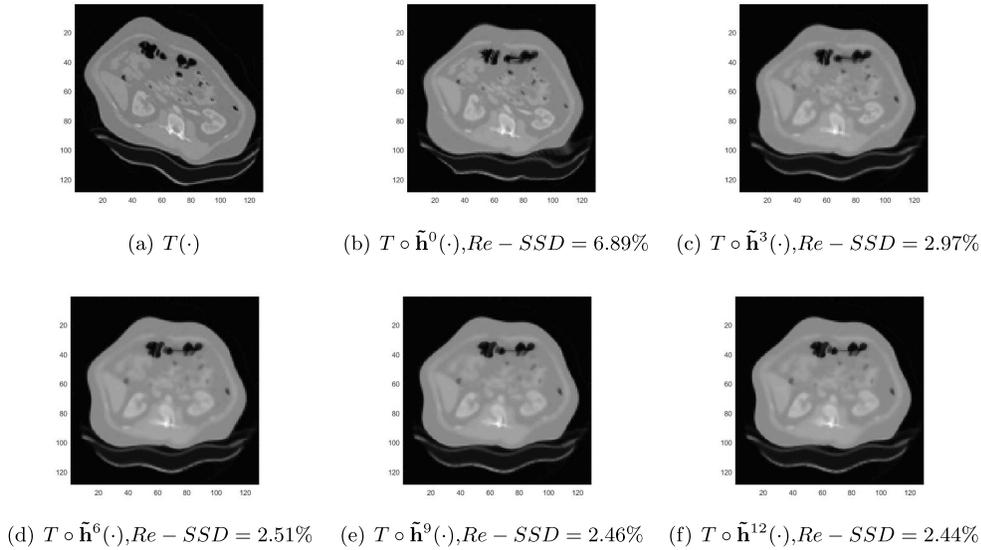


FIG. 24. Registration result of different scale on image pairs Liver.

TABLE 3

Quantitative comparison between registration results of five different algorithms (medical images).

Data	Algorithm	Re – SSD(%)	MFN	CPU/s
Brain	Proposed	2.87	0	225.1
	DFIRA	3.45	0	113.3
	DLDIR	10.8	0	18.1
	A4	5.39	0	25.1
	LDDMM	7.54	0	122.6
Hand	Proposed	3.47	0	241.1
	DFIRA	5.42	0	75.3
	DLDIR	16.95	0	13.0
	A4	7.9	0	24.8
	LDDMM	5.99	0	114.9
Chest	Proposed	1.44	0	172.1
	DFIRA	2.01	0	180.3
	DLDIR	10.93	0	5.88
	A4	3.2	0	18.2
	LDDMM	5.55	0	123.2
Liver	Proposed	2.44	0	311.3
	DFIRA	3.7	0	264.8
	DLDIR	6.43	0	19.9
	A4	4.1	0	61.1
	LDDMM	15.14	0	118.5

Acknowledgments. The authors of this paper would like to thank Professor Huan-Song Zhou at the Center for Mathematical Sciences, Wuhan University of Technology, for his support and help on this work. Thanks also to the referee for her/his very helpful remarks.

REFERENCES

- [1] G. AUBERT AND P. KORNPBST, *Mathematical Problems in Image Processing*, Springer, New York, 2006.
- [2] M. F. BEG, M. I. MILLER, A. TROUVÉ, AND L. YOUNES, *Computing large deformation metric mappings via geodesic flows of diffeomorphisms*, Int. J. Comput. Vis., 61 (2005), pp. 139–157, <https://doi.org/10.1023/B:VISI.0000043755.93987.aa>.
- [3] M. BRUVERIS, F. GAY-BALMAZ, D. D. HOLM, AND T. S. RATIU, *The momentum map representation of images*, J. Nonlinear Sci., 21 (2011), pp. 115–150, <https://doi.org/10.1007/s00332-010-9079-5>.
- [4] A. BUDHIRAJA, P. DUPUIS, AND V. MAROULAS, *Large deviations for stochastic flows of diffeomorphisms*, Bernoulli, 16 (2010), pp. 234–257, <https://doi.org/projecteuclid.org/euclid.bj/1265984710>.
- [5] C. CHEN AND O. OKTEM, *Indirect image registration with large diffeomorphic deformations*, SIAM J. Imaging Sci., 11 (2018), pp. 575–617, <https://doi.org/10.1137/17M1134627>.
- [6] G. CHOI, D. QIU, AND L. M. LUI, *Shape analysis via inconsistent surface registration*, Proc. A, 476 (2020), pp. 1–21, <https://doi.org/10.1098/rspa.2020.0147>.
- [7] N. CHUMCHOB AND K. CHEN, *A robust multigrid approach for variational image registration models*, J. Comput. Appl. Math., 236 (2011), pp. 653–674, <https://doi.org/10.1016/j.cam.2011.06.026>.
- [8] L. K. CHUN AND L. M. LUI, *Landmark and intensity based registration with large deformations via quasi-conformal maps*, SIAM J. Imaging Sci., 7(4) (2013), pp. 2364–2392, <https://doi.org/10.1137/130943406>.
- [9] F. DEMENGEL, G. DEMENGEL, AND R. ERNE, *Functional Spaces for the Theory of Elliptic Partial Differential Equations*, Springer, New York, 2012.
- [10] L. EVANS, *Partial Differential Equations*, AMS, Providence, RI, 2010.
- [11] H. LOMBAERT, *Diffeomorphic Log Demons Image Registration*, https://ww2.mathworks.cn/matlabcentral/fileexchange/39194-diffeomorphic-log-demons-image-registration?s_tid=FX_rc2-behav, 2012.

- [12] H. HAN, *A variational model with fractional-order regularization term arising in registration of diffusion tensor image*, Inverse Probl. Imaging, 12 (2018), pp. 1263–1291, <https://doi.org/aimsciences.org/article/doi/10.3934/ipi.2018053>.
- [13] H. HAN, *A fractional-order decomposition model of image registration and its numerical algorithm*, Comput. Appl. Math., 39 (2020), 45, <https://doi.org/10.1007/s40314-020-1066-3>.
- [14] H. HAN AND A. WANG, *A fast multi grid algorithm for 2D diffeomorphic image registration model*, J. Comput. Appl. Math., 394 (2021), 113576, <https://doi.org/10.1016/j.cam.2021.113576>.
- [15] H. HAN AND Z. WANG, *An alternating direction implicit scheme of a fractional-order diffusion tensor image registration model*, Appl. Math. Comput., 256 (2019), pp. 105–118, <https://doi.org/10.1016/j.amc.2019.03.024>.
- [16] H. HAN AND Z. WANG, *A diffeomorphic image registration model with fractional-order regularization and Cauchy-Riemann constraint*, SIAM J. Imaging Sci., 13 (2020), pp. 1240–1271, <https://doi.org/10.1137/19M1260621>.
- [17] H. HAN AND H. ZHOU, *A variational problem arising in registration of diffusion tensor images*, Acta Math. Sci. Ser. B, 37 (2017), pp. 539–554, [https://doi.org/10.1016/S0252-9602\(17\)30020-6](https://doi.org/10.1016/S0252-9602(17)30020-6).
- [18] W. V. HECKE AND A. LEEMANS, *Nonrigid coregistration of diffusion tensor images using a viscous fluid model and mutual information*, IEEE Trans. Med. Imaging, 26 (2007), pp. 1598–1612, <https://doi.org/10.1109/TMI.2007.906786>.
- [19] K. C. LAM AND L. M. LUI, *Optimized quasiconformal parameterization with user-defined area distortions*, Commun. Math. Sci., 15 (2017), pp. 2027–2054, <https://doi.org/10.4310/CMS.2017.v15.n7.a11>.
- [20] C. P. LAU, Y. LAI, AND L. M. LUI, *Restoration of atmospheric turbulence distorted images via rpca and quasiconformal maps*, Inverse Problems, 35 (2019), 074002, <https://doi.org/10.1088/1361-6420/ab0e4b>.
- [21] 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), pp. 926–954, <https://doi.org/10.1007/s10915-015-0113-5>.
- [22] 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. Imaging, 33 (2014), pp. 1005–1022, <https://doi.org/10.1109/TMI.2013.2274051>.
- [23] 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), pp. 254–271, <https://doi.org/10.1007/s11263-013-0681-5>.
- [24] L. M. LUI AND T. C. NG, *A splitting method for diffeomorphism optimization problem using Beltrami coefficients*, J. Sci. Comput., 63 (2015), pp. 573–611, <https://doi.org/10.1007/s10915-014-9903-4>.
- [25] L. M. LUI AND C. WEN, *Geometric registration of high-genus surfaces*, SIAM J. Imaging Sci., 7 (2014), pp. 337–365, <https://doi.org/10.1137/130932053>.
- [26] 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), pp. 557–585, <https://doi.org/10.1007/s10915-011-9506-2>.
- [27] A. MANG AND G. BIROS, *An inexact newton-krylov algorithm for constrained diffeomorphic image registration*, SIAM J. Imaging Sci., 8 (2015), pp. 1030–1069, <https://doi.org/10.1137/140984002>.
- [28] A. MANG AND G. BIROS, *Constrained h^1 -regularization schemes for diffeomorphic image registration*, SIAM J. Imaging Sci., 9 (2016), pp. 1154–1194, <https://doi.org/10.1137/15M1010919>.
- [29] K. MODIN, A. NACHMAN, AND L. RONDI, *A multiscale theory for image registration and nonlinear inverse problems*, Adv. Math., 346 (2019), pp. 1009–1066, <https://doi.org/10.1016/j.aim.2019.02.014>.
- [30] Z. NIE AND X. YANG, *Deformable image registration using functions of bounded deformation*, IEEE Trans. Med. Imaging, 38 (2019), pp. 1488–1500, <https://doi.org/10.1109/TMI.2019.2896170>.
- [31] D. QIU, K. C. LAM, AND L. M. LUI, *Computing quasiconformal folds*, SIAM J. Imaging Sci., 12 (2019), pp. 1392–1424, <https://doi.org/10.1137/18M1220042>.
- [32] D. QIU AND L. M. LUI, *Inconsistent surface registration via optimization of mapping distortions*, J. Sci. Comput., 83 (2020), 64, <https://doi.org/10.1007/s10915-020-01246-5>.
- [33] L. RISSER, F. VIALARD, R. WOLZ, M. MURGASOVA, D. D. HOLM, AND D. RUECKERT, *Simultaneous multi-scale registration using large deformation diffeomorphic metric mapping*, IEEE Trans. Med. Imaging, 30 (2011), pp. 1746–1759, <https://doi.org/10.1109/TMI.2011.2146787>.

- [34] S. SOMMER, F. LAUZE, M. NIELSEN, AND X. PENNEC, *Sparse multi-scale diffeomorphic registration: The kernel bundle framework*, J. Math. Imaging Vision, 46 (2013), pp. 292–308, <https://doi.org/10.1007/s10851-012-0409-0>.
- [35] G. STRANG, *Introduction to Linear Algebra*, Wellesley Cambridge Press, Wellesley, MA, 2016.
- [36] T. POCK, M. URSCHLER, AND C. ZACH, *A duality based algorithm for TV- L^1 -optical-flow image registration*, in Proceedings of the International Conference on Medical Image Computing and Computer-assisted Intervention, 1986, pp. 442–461.
- [37] E. TADMOR AND S. NEZZAR, *A multiscale image representation using hierarchical (BV, L^2) decompositions*, SIAM J. Multiscale Model. Simul., 2 (2004), pp. 554–579, <https://doi.org/10.1137/030600448>.
- [38] T. VERCAUTEREN, X. PENNEC, A. PERCHANT, AND N. AYACHE, *Diffeomorphic demons: Efficient non-parametric image registration*, Neuroimage, 45 (2009), pp. 61–72, <https://doi.org/10.1016/j.neuroimage.2008.10.040>.
- [39] Y. WANG, L. M. LUI, T. F. CHAN, AND P. M. THOMPSON, *Optimization of brain conformal mapping with landmarks*, in Medical Image Computing and Computer Assisted Intervention, Lecture Notes in Comput. Sci. 3750, Springer, New York, 2005, pp. 675–683, https://doi.org/10.1007/11566489_83.
- [40] 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. Imaging, 26 (2007), pp. 853–865, <https://doi.org/10.1109/TMI.2007.895464>.
- [41] J. WU AND X. TANG, *A large deformation diffeomorphic framework for fast brain image registration via parallel computing and optimization*, Neuroinformatics, 18 (2020), pp. 251–266, <https://doi.org/10.1007/s12021-019-09438-7>.
- [42] Y. TANG, *Reconstruction of Distorted Underwater Images Based on Edge Priority*, <https://github.com/tangyugui1998/Reconstruction-of-distorted-underwater-images>, 2020.
- [43] C. P. YUNG, G. CHOI, K. CHEN, AND L. M. LUI, *Efficient feature-based image registration by mapping sparsified surfaces*, J. Vis. Commun. Image Representation, 55 (2018), pp. 561–571, <https://doi.org/10.1016/j.jvcir.2018.07.005>.
- [44] D. ZHANG AND K. CHEN, *A novel diffeomorphic model for image registration and its algorithm*, J. Math. Imaging Vision, 60 (2018), pp. 1261–1283, <https://doi.org/10.1007/s10851-018-0811-3>.
- [45] D. ZHANG AND K. CHEN, *3D orientation-preserving variational models for accurate image registration*, SIAM J. Imaging Sci., 13 (2020), pp. 1653–1691, <https://doi.org/10.1137/20M1320006>.
- [46] D. ZHANG AND L. M. LUI, *Topology-preserving 3D image segmentation based on hyperelastic regularization*, J. Sci. Comput., 87 (2021), 74, <https://doi.org/10.1007/s10915-021-01433-y>.
- [47] J. ZHANG AND K. CHEN, *Variational image registration by a total fractional-order variation model*, J. Comput. Phys., 293 (2015), pp. 442–461, <https://doi.org/10.1016/j.jcp.2015.02.021>.
- [48] J. ZHANG, K. CHEN, AND B. YU, *A novel high-order functional based image registration model with inequality constraint*, Comput. Math. Appl., 72 (2016), pp. 2887–2899, <https://doi.org/10.1016/j.camwa.2016.10.018>.