Journal of Mathematical Imaging and Vision (2025) 67:38
https://doi.org/10.1007/s10851-025-01254-w

Vectorial Fractional-Order Regularizer-Based Diffeomorphic Image Registration Model and its Numerical Algorithm

Jin Zhang[1] · Xu Kong[1] · Jianping Zhang[2] · Fenlin Yang[3] · Ke Chen[4]

Received: 22 December 2024 / Accepted: 12 May 2025

Abstract
A diffeomorphic image registration model with a vectorial fractional-order regularizer is introduced to handle displacement fields with varying smoothness and to avoid mesh folding. Furthermore, we combine the damped Newton method with the Armijo line search and apply a multilevel strategy to solve the discretized version of the new model. Furthermore, both the existence of solutions to the model and the convergence of the algorithm have been established. Numerical experiments on synthetic and real images confirm the superiority of the proposed model and the effectiveness of the algorithm.

Keywords Diffeomorphic image registration · Mesh folding · Fractional-order regularizer

Mathematics Subject Classification 65F10 · 65M55 · 68U10

1 Introduction

Image registration plays a vital role in image processing and presents significant challenges in digital imaging technologies. The application of this technique spans several fields, including medical imaging, image analysis, remote sensing, and computer vision [1–3]. It has been proven that variational methods and PDE-based techniques are highly effective in numerous applications [4, 5].

Mathematically, variational image registration is defined as follows: For two images of the same object, the reference image F is kept fixed, while the template image M is deformed. The two images are treated as compactly supported functions within a bounded convex domain $\Omega \subset \mathbb{R}^d$, with d indicating the spatial dimension. This paper specifically addresses the case where $d = 2$.

The target of image registration is to identify an appropriate geometric transformation $\boldsymbol{y}(\boldsymbol{x}) : \Omega \to \Omega$ that ensures the deformed template image $M(\boldsymbol{y}(\boldsymbol{x}))$ closely matches the reference image $F(\boldsymbol{x})$. To enable visual evaluation of the displacement of points in the transformed template image from their original locations, the transformation $\boldsymbol{y}$ can be expressed as follows:

$$\boldsymbol{y}(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}),$$

here, $\boldsymbol{u} : \boldsymbol{x} \mapsto \boldsymbol{u}(\boldsymbol{x}) = (u_1(\boldsymbol{x}), u_2(\boldsymbol{x}))^\top$ represents the displacement field. Therefore, determining the transformation $\boldsymbol{y}$ and the displacement field $\boldsymbol{u}$ is equivalent. Common metrics for measuring the similarity between the transformed template image $M(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}))$ and the reference image $F(\boldsymbol{x})$ include normalized cross-correlation (NCC), mutual information (MI), and the sum of squared intensity differences (SSD). For further details, refer to [4, 5]. In this study, we focus on using SSD metrics for single-modality image registration, assuming that the density values of the images are comparable. Consequently, the image registration problem can be formulated as an optimization task:

$$\min_{\boldsymbol{u}} \{\mathcal{J}(\boldsymbol{u}) = \frac{1}{2} \int_\Omega (M(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})) - F(\boldsymbol{x}))^2 d\boldsymbol{x}\}, \tag{1}$$

This research work was supported by the Natural Science Foundation of China (NSFC) (No: 11801249, 11771369), and Natural Science Foundation of Shandong Province (No: ZR2024MF143), and the Education Bureau of Hunan Province, P.R.China (No: 22A0119).

✉ Ke Chen
k.chen@strath.ac.uk
https://www.strath.ac.uk/staff/chenkeprofessor/

1 School of Mathematical Sciences, Liaocheng University, Liaocheng 252059, China

2 School of Mathematics and Computational Science, Xiangtan University, Xiangtan 411105, China

3 College of Mathematics and Statistics, Jishou University, Jishou 416000, China

4 Department of Mathematics and Statistics, University of Strathclyde, Glasgow G1 1XQ, UK

as is well known, model (1) is considered an ill-posed problem. A common approach to regularizing this model (1) is to impose constraints on the displacement field $\boldsymbol{u}$ by adding an appropriate regularizer $\mathcal{S}(\boldsymbol{u})$ to eliminate unrealistic solutions. Therefore, finding the displacement field $\boldsymbol{u}$ requires solving the following optimization problem:

$$\min_{\boldsymbol{u}} \{\mathcal{J}(\boldsymbol{u}) = \mathcal{D}(\boldsymbol{u}) + \lambda \mathcal{R}(\boldsymbol{u})\}, \tag{2}$$

the regularization parameter $\lambda > 0$ controls the trade-off between the data fitting term and the smoothness of the displacement field $\boldsymbol{u}$.

It is widely acknowledged that the choice of regularizer significantly influences the resulting image registration models, affecting the solution and its properties [4, 5]. A variety of regularizers involving integer derivatives of the displacement field $\boldsymbol{u}$ have been proposed in the literature [6–10]. In the last twenty years, research has demonstrated that fractional-order derivatives, as opposed to integer-order derivatives, provide a more precise modeling approach for a range of scientific and engineering applications [11–19]. Inspired by these findings, this article explores the use of fractional-order derivatives in image registration. Several models for fractional-order image registration have been proposed [20–23]. However, these models have not taken mesh folding into consideration. In image registration, mesh folding can lead to the following several adverse effects: mesh folding may cause certain pixels to be covered or suppressed, leading to the loss of important information, which in turn affects the results of image analysis and processing; on the other hand, the folded grid may introduce artifacts or discontinuities in edge features, degrading image quality and resulting in unreliable analysis outcomes. Nowadays, several new image registration models [10, 24–35] have been proposed as far as the elimination of mesh folding is concerned. While certain models achieve satisfactory registration results without mesh folding, they tend to be more computationally intensive due to the complexity of their objective functions. Moreover, some models are only effective for smooth displacement fields, resulting in relatively poor performance when applied to discontinuous displacement fields.

Given the difficulty of directly assessing the smoothness or discontinuity of the displacement field $\boldsymbol{u}$ in real-world applications, it is impractical to assume smoothness a priori. To overcome the limitations of existing variational models, we introduce a diffeomorphic image registration model that employs a vectorial fractional-order regularizer. This model is designed to handle displacement fields with different levels of smoothness while effectively preventing mesh folding.

This paper is organized in the following way: Sect. 2 reviews the definitions related to fractional-order derivatives. In Sect. 3, we develop a differential diffeomorphic image registration model using a vectorial fractional-order regularizer and demonstrate the existence of solutions for this model. Section 4 presents an efficient numerical method to address the new registration model and analyzes its convergence. Section 5 showcases numerical experiments that illustrate the advantages of the proposed model and the effectiveness of the algorithm. Finally, Sect. 6 presents the conclusions and discusses future work.

2 Fractional-Order Derivatives

In this section, we present a brief overview of the definitions of the three most commonly used fractional derivatives.

1. **Grünwald-Letnikov (GL) definitions** [36]. Let $g(x)$ be a function defined on the interval $[a, b]$, where α is a positive real number, then the following two formulas

$$ {}^{\mathrm{GL}}D^{\alpha}_{[a,x]}g(x) = \lim_{h\to 0} \frac{1}{h^{\alpha}} \sum_{i=0}^{[\frac{x-a}{h}]} (-1)^i \binom{\alpha}{i} g(x-ih) \,,$$

$$ {}^{\mathrm{GL}}D^{\alpha}_{[x,b]}g(x) = \lim_{h\to 0} \frac{1}{h^{\alpha}} \sum_{i=0}^{[\frac{b-x}{h}]} (-1)^i \binom{\alpha}{i} g(x+ih) \,.$$

   are called α-order Grünwald-Letnikov left derivative and right derivative of $g(x)$, respectively, where $[.]$ represents a round-down operator. $\binom{\alpha}{i} = \frac{\alpha(\alpha-1)\cdots(\alpha-i+1)}{i!}$ represents the binomial coefficient.
2. **Riemann-Liouville (RL) definitions** [36]. Let n be a positive integer, where $n-1 \leq \alpha < n$ and $\Gamma(\cdot)$ represents Gamma function. The α-order Riemann-Liouville left derivative and right derivative of $g(x)$ are then defined as follows:

$$ D^{\alpha}_{[a,x]}g(x) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dx^n} \int_a^x \frac{g(\tau)}{(x-\tau)^{\alpha-n+1}} d\tau \,,$$

$$ D^{\alpha}_{[x,b]}g(x) = \frac{1}{\Gamma(n-\alpha)} (-1)^n \frac{d^n}{dx^n} \int_x^b \frac{g(\tau)}{(\tau-x)^{\alpha-n+1}} d\tau \,.$$

3. **Caputo (C) definitions** [36]. The α-order Caputo left derivative and right derivative of the function $g(x)$ are given by

$$ {}^{C}D^{\alpha}_{[a,x]} = \frac{1}{\Gamma(n-\alpha)} \int_a^x \frac{g^{(n)}(\tau)}{(x-\tau)^{\alpha-n+1}} d\tau \,,$$

$$ {}^{C}D^{\alpha}_{[x,b]} = \frac{(-1)^n}{\Gamma(n-\alpha)} \int_x^b \frac{g^{(n)}(\tau)}{(\tau-x)^{\alpha-n+1}} d\tau \,.$$

The three fractional derivatives described above are essentially equivalent, provided that the function $g(x)$ meets the

appropriate continuity and boundary conditions [20]. Therefore, we will omit the superscript of the fractional derivative symbol in the following sections. We will use D^α to denote the fractional-order derivative operator defined on the closed interval $[a, b]$. Similarly, $\nabla^\alpha = (\frac{\partial^\alpha}{\partial x_1^\alpha}, \frac{\partial^\alpha}{\partial x_2^\alpha}, \cdots, \frac{\partial^\alpha}{\partial x_d^\alpha})^\top$ and $\mathrm{div}^\alpha = \nabla^\alpha \cdot$ can be expressed as operators for gradient and divergence, respectively. Thus, the fractional α-order derivative $D^\alpha_{[a,b]} g(x)$ of the function $g(x)$ in the x-direction can be denoted by $\frac{\partial^\alpha g(x)}{\partial x_i^\alpha}$.

3 A New Variational Image Registration Model

The optimal regularizers ought to be competent in adapting to displacement fields featuring varying degrees of smoothness. Recent studies have shown that specific regularization methods are particularly effective for optical flow computation, image restoration, and image reconstruction [37]. The core idea behind these methods is to substitute $|\nabla \boldsymbol{u}|$ with a potential function $\varphi(|\nabla \boldsymbol{u}|)$. This approach allows for isotropic smoothing of $\boldsymbol{u}$ in homogeneous (or flat) regions while preserving discontinuities in inhomogeneous areas. Moreover, the literature [14–17, 20–23] clearly demonstrates that models using fractional-order regularizers achieve better results than those that utilize integer-order ones. Drawing inspiration from these findings and the concept of vectorization in color image denoising [38–40], we propose a new regularizer for image registration, defined as follows:

$$\mathcal{R}^{\mathrm{new}}(\boldsymbol{u}) = \int_\Omega \left(\sqrt{1 + |\nabla^\alpha \boldsymbol{u}|^2} - 1\right) d\boldsymbol{x} \,. \tag{3}$$

Thus, the variational model (2) is given by:

$$\min_{\boldsymbol{u}} \left\{ \mathcal{J}(\boldsymbol{u}) = \frac{1}{2} \int_\Omega (M(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})) - F(\boldsymbol{x}))^2 d\boldsymbol{x} + \lambda \int_\Omega (\sqrt{1 + |\nabla^\alpha \boldsymbol{u}|^2} - 1) d\boldsymbol{x} \right\}, \tag{4}$$

where $n - 1 < \alpha < n$, and $\nabla^\alpha \boldsymbol{u} = (\nabla^\alpha u_1, \nabla^\alpha u_2)^\top$, $|\nabla^\alpha \boldsymbol{u}| = \sqrt{|\nabla^\alpha u_1|^2 + |\nabla^\alpha u_2|^2}$, here and in what follows, $n = 2$. The Euler-Lagrange equation associated with the registration model (4) can be formulated as a system of two coupled partial differential equations:

$$\begin{cases} f_1(\boldsymbol{u}) - \lambda \nabla^\alpha \cdot \frac{\nabla^\alpha u_1}{\sqrt{1+|\nabla^\alpha \boldsymbol{u}|^2}} = 0 \\ f_2(\boldsymbol{u}) - \lambda \nabla^\alpha \cdot \frac{\nabla^\alpha u_2}{\sqrt{1+|\nabla^\alpha \boldsymbol{u}|^2}} = 0 \end{cases}, \tag{5}$$

subjects to $D^{\alpha-n+j} \boldsymbol{W}^i \cdot \boldsymbol{n} = 0$, $j = 0, 1$ on $\partial\Omega$, and $\boldsymbol{W}^i = \frac{\nabla^\alpha u_i}{\sqrt{1+|\nabla^\alpha \boldsymbol{u}|^2}}$, i=1,2; $\boldsymbol{n} = (n_1, n_2)$ represents the outer unit normal vector of the boundary $\partial\Omega$. Here $f_k(\boldsymbol{u}) = (M(\boldsymbol{u}) - F(\boldsymbol{x}))\partial_{u_k} M(\boldsymbol{u})$, $k = 1, 2$.

The use of $\mathcal{D}(\boldsymbol{u})$ as a similarity measure in model (2) leads to non-convex image registration models, making it difficult to guarantee a unique solution for model (2). This non-convexity may result in mesh folding within the displacement field $\boldsymbol{u}$, an occurrence that is not acceptable in practical applications. During image registration, the deformation field must retain physical meaning, avoiding excessive stretching, compression, or folding. The volume of discrete cells (in $3D$) or the area of pixels (in $2D$) directly measures local deformation magnitude and direction. Constraining the volume or area helps prevent grid structures from self-intersecting, folding, or experiencing extreme distortion. This is essential for ensuring the deformation remains physically plausible. To avoid the occurrence of mesh folding phenomenon, inspired by the model in literature [10, 24], we have constructed the following diffeomorphic image registration model based on the vectorized fractional-order regularizer:

$$\min_{\boldsymbol{u} \in \mathrm{BV}_2^\alpha} \left\{ \mathcal{J}(\boldsymbol{u}) = \frac{1}{2} \int_\Omega (M(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})) - F)^2 d\boldsymbol{x} + \lambda \mathcal{R}^{\mathrm{new}}(\boldsymbol{u}) + \mu \mathcal{R}_1(\boldsymbol{u}) \right\}, \tag{6}$$

where

$$\mathcal{R}_1(\boldsymbol{u}) = \int_\Omega \varphi(\det \nabla \boldsymbol{y}) d\boldsymbol{x},$$
$$\varphi(\det \nabla \boldsymbol{y}) = ((\det \nabla \boldsymbol{y} - 1)^2 / \det \nabla \boldsymbol{y})^2,$$
$$\nabla \boldsymbol{y} = I_2 + \nabla \boldsymbol{u},$$

and

$$\begin{aligned} \det \nabla \boldsymbol{y} &= \det(I_2 + \nabla \boldsymbol{u}) \\ &= \begin{vmatrix} 1 + \frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} \\ \frac{\partial u_2}{\partial x_1} & 1 + \frac{\partial u_2}{\partial x_2} \end{vmatrix} \\ &= (1 + \frac{\partial u_1}{\partial x_1})(1 + \frac{\partial u_2}{\partial x_2}) - \frac{\partial u_1}{\partial x_2} \frac{\partial u_2}{\partial x_1}, \end{aligned} \tag{7}$$

and BV_2^α represents the Banach space.

The advantages of our proposed new model are as follows. Firstly, the coupling between the main elements u_1 and u_2 of the displacement field $\boldsymbol{u}$ is ensured through the diffusion coefficient $D(\boldsymbol{u}) = \frac{1}{\sqrt{1+|\nabla^\alpha \boldsymbol{u}|^2}}$ which locally adjusts the level of the regularizer, thus further improving registration quality by observing the partial differential equations (5). Secondly, (a) : $\sqrt{1 + |\nabla^\alpha \boldsymbol{u}|^2} - 1 \approx \frac{1}{2}|\nabla^\alpha \boldsymbol{u}|^2$, as $|\nabla^\alpha \boldsymbol{u}| \to 0$; (b): $\sqrt{1 + |\nabla^\alpha \boldsymbol{u}|^2} - 1 \approx |\nabla^\alpha \boldsymbol{u}|$, as $|\nabla^\alpha \boldsymbol{u}| \to +\infty$, which means that the fractional-order diffusion-like regularizer is used in uniform (flat) regions, and the fractional-order TV-like regularizer is used in non-uniform regions. This suggests that our novel model is capable of producing satisfactory registration

results for problems with varying degrees of smoothness. Furthermore, our new model effectively prevents mesh folding. Concerning the existence of a solution for model (6), we propose the following arguments.

Definition 3.1 [41] Let $f : X \to \mathbb{R}$ be a measurable function (where X is a measure space); the essential supremum is then defined as:

$$\text{ess sup } f = \inf \{K \in \mathbb{R} \,|\, \gamma(\{x \in X \,|\, f(x) > K\}) = 0\}.$$

Lemma 3.1 *For* $\forall x \geq 0$, the function $f(x) = \sqrt{1+x^2} - 1 \leq x$.

Proof For $\forall x \geq 0$, it is obvious that we can obtain:

$$1 + x^2 \leq 1 + 2x + x^2 = (1+x)^2,$$

$$\therefore \sqrt{1+x^2} \leq 1 + x,$$

Furthermore, it can be obtained that:

$$\sqrt{1+x^2} - 1 \leq x.$$

Lemma 3.2 *If the function* $f(x) = \sqrt{1+x^2} - 1$ *is bounded, then* x *must be bounded.*

Proof It is known that the function $f(x) = \sqrt{1+x^2} - 1$ is bounded, which means that there exists a constant $M > 0$ such that for all x, we have

$$f(x) = |\sqrt{1+x^2} - 1| \leq M.$$

That is,

$$-M \leq \sqrt{1+x^2} - 1 \leq M,$$

furthermore, it can be obtained that:

$$1 - M \leq \sqrt{1+x^2} \leq 1 + M\ ,$$

squaring both sides and further simplifying, we obtain:

$$M^2 - 2M \leq x^2 \leq M^2 + 2M,$$

since $x^2 \geq 0$, it follows that:

$$|x| \leq \sqrt{M^2 + 2M}.$$

Theorem 3.1 If $\text{ess sup}_{\boldsymbol{x}\in\Omega} |M(\boldsymbol{x})| < K < +\infty$, $\text{ess sup}_{\boldsymbol{x}\in\Omega} |F(\boldsymbol{x})| < K < +\infty$, let $\triangle M \triangleq \{\boldsymbol{x} : M(\boldsymbol{x})$ is discontinuous at $\boldsymbol{x}\}$ be a zero measure set, then the solution to model *(6)* exists.

Proof Assume that $\{\boldsymbol{u}_n\}$ is a minimizing sequence for $\mathcal{J}(\boldsymbol{u})$, then

$$\mathcal{J}(\boldsymbol{u}_n) \leq \mathcal{J}(0) = \mathcal{D}(0) < +\infty,$$

in combination with the conclusions of Lemma 3.1 and Lemma 3.2, it can be known that there exists a constant $M_1 > 0$ for which the following expression holds:

$$\begin{aligned} &\|\sqrt{1 + |\nabla^\alpha \boldsymbol{u}_n|^2} - 1\|_{[L^2(\Omega)]^2} \leq \|\nabla^\alpha \boldsymbol{u}_n\|_{[L^2(\Omega)]^2} \\ &= \left(\int_\Omega |\nabla^\alpha \boldsymbol{u}_n|^2 d\boldsymbol{x}\right)^{\frac{1}{2}} \leq M_1. \end{aligned}$$

On the other hand, according to Lemma 3.3 in [22], it can be learned that for $\forall \boldsymbol{u}_n \in [H_0^\alpha(\Omega)]^2$, let $C_0 > 0$ be a constant, and the following inequality holds:

$$\|\boldsymbol{u}_n\|_{[L^2(\Omega)]^2} \leq C_0 \|\nabla^\alpha \boldsymbol{u}_n\|_{[L^2(\Omega)]^2}. \tag{8}$$

Therefore, for a certain constant M_2, the following expression holds:

$$\begin{aligned} \|\boldsymbol{u}_n\|_{[H_0^\alpha(\Omega)]^2} &= (\|\boldsymbol{u}_n\|_{[L^2(\Omega)]^2} + \|\nabla^\alpha \boldsymbol{u}_n\|_{[L^2(\Omega)]^2})^{\frac{1}{2}} \\ &\leq M_2 < +\infty. \end{aligned} \tag{9}$$

By formulas (8) and (9), it can be deduced that the semi-norm defined by $|\boldsymbol{u}|_{[H_0^\alpha(\Omega)]^2} = (\int_\Omega |\nabla^\alpha \boldsymbol{u}|^2 d\boldsymbol{x})^{\frac{1}{2}}$ is equivalent to the norm $\|\boldsymbol{u}\|_{[H_0^\alpha(\Omega)]^2}$ on space $[H_0^\alpha(\Omega)]^2$. i.e., $|\boldsymbol{u}|_{[H_0^\alpha(\Omega)]^2} \leq \|\boldsymbol{u}\|_{[H_0^\alpha(\Omega)]^2} \leq M_3 |\boldsymbol{u}|_{[H_0^\alpha(\Omega)]^2}$. This conclusion can be found in Proposition 1.5 in the literature [42].

According to Theorem 4.58 (see [43] for details), $H_0^\alpha(\Omega)$ is compactly embedded in $C^1(\Omega)$. Therefore, there exists a subsequence $\boldsymbol{u}_n$ still marked by n, and $\boldsymbol{u} \in [C^1(\Omega)]^2$, thereby making,

$$\lim_{n\to+\infty} \|\boldsymbol{u}_n - \boldsymbol{u}\|_{[C^1(\Omega)]^2} = 0 \tag{10}$$

Using this framework, we can now derive the weak lower continuity of the functional $\mathcal{J}(\boldsymbol{u})$.

Firstly, based on the lower semi-continuity of the norm (Refer to section D4 on page 639 in [44].) and (9), given that $|\cdot|_{[H_0^\alpha(\Omega)]^2}$ is the norm in the space $[H_0^\alpha(\Omega)]^2$, we then deduce that

$$\begin{aligned} &\int_\Omega |\sqrt{1 + |\nabla^\alpha \boldsymbol{u}|^2} - 1| d\boldsymbol{x} \\ &\leq \lim_{n\to+\infty} \inf \int_\Omega |\sqrt{1 + |\nabla^\alpha \boldsymbol{u}_n|^2} - 1| d\boldsymbol{x}. \end{aligned}$$

This fully indicates that

$$\mathcal{R}^{\text{new}}(\boldsymbol{u}) \leq \lim_{n\to+\infty} \inf \mathcal{R}^{\text{new}}(\boldsymbol{u}_n). \tag{11}$$

Additionally, according to (10), it follows that

$$
\begin{aligned}
&|\mathcal{D}(\boldsymbol{u}_n) - \mathcal{D}(\boldsymbol{u})| \\
&\quad \le K \int_{\Omega} |M(\boldsymbol{x} + \boldsymbol{u}_n(\boldsymbol{x})) - M(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}))| d\boldsymbol{x} \\
&\quad = K \int_{\Omega \backslash \triangle M} |M(\boldsymbol{x} + \boldsymbol{u}_n(\boldsymbol{x})) - M(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}))| d\boldsymbol{x}.
\end{aligned}
$$

Since $\triangle M \triangleq \{\boldsymbol{x} : M(\boldsymbol{x}) \text{ is discontinuous at } \boldsymbol{x}\}$ is a set of measure zero; therefore, $M(x)$ is continuous on $\Omega \backslash \triangle M$. Further, according to equation (10):

$$
\lim_{n \to +\infty} \|\boldsymbol{u}_n - \boldsymbol{u}\|_{[C^1(\Omega)]^2} = 0,
$$

thus, we can obtain:

$$
\lim_{n \to +\infty} \int_{\Omega \backslash \triangle M} |M(\boldsymbol{x} + \boldsymbol{u}_n(\boldsymbol{x})) - M(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}))| d\boldsymbol{x} = 0.
$$

So it can be deduced that:

$$
\lim_{n \to +\infty} \mathcal{D}(\boldsymbol{u}_n) = \mathcal{D}(\boldsymbol{u}). \tag{12}
$$

Furthermore, it can be inferred from (10) $\lim_{n \to +\infty} \frac{\partial (\boldsymbol{u}_n)_i}{\partial x_j} = \frac{\partial \boldsymbol{u}_i}{\partial x_j}, i, j = 1, 2$. This consequently leads to the conclusion that

$$
\lim_{n \to +\infty} \mathcal{R}_1(\boldsymbol{u}_n) = \mathcal{R}_1(\boldsymbol{u}). \tag{13}
$$

Based on (11), (12), and (13), it is evident that

$$
\lim_{n \to +\infty} \inf \mathcal{J}(\boldsymbol{u}_n) \ge \mathcal{J}(\boldsymbol{u}).
$$

This confirms the lower weak continuity of the functional $\mathcal{J}(\boldsymbol{u})$ and indicates the existence of a solution to (6)(refer to Section 2.1.2 in [37]). □

4 Numerical Solution for the Novel Model Proposed in (6)

Obtaining an analytical solution for model (6) is often challenging, necessitating appropriate discretization to derive a numerical solution. In this study, we first discretize Model (6) and then apply standard optimization techniques to solve the discretized system. We begin by introducing the discretization method, followed by a detailed presentation of the numerical algorithm used for the solution.

4.1 Finite Difference Discretization

We assume that the number of pixels in the discrete image is $m_1 \times m_2$ and that each pixel is a rectangle with side lengths h_1 and h_2. In this description, half-step indexing is allowed. Consistent with common practices in the field of image processing, we associate the pixels with grid points centered at $\boldsymbol{x}_{i_1+0.5, i_2+0.5}$. For the rectangle centered at $\boldsymbol{x}_{i_1+0.5, i_2+0.5}$, the indices of its four vertices are determined by the integer index $i_k, i_k + 1, k = 1, 2$. The displacement $\boldsymbol{u} = (u_1, u_2)^\top$ is discretized on the node grid (i.e., at the vertices of each rectangle; see Fig. 1a).

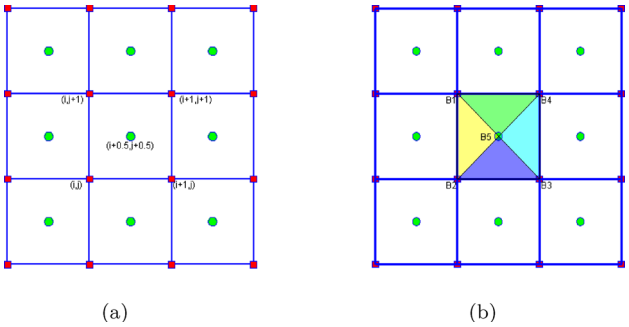


Fig. 1 **a** Nodal ■ and cell-centered •; **b** A pixel is divided into four triangles

We use a nodal grid to define a spatial partition $\Omega_h = \{\boldsymbol{x}_{i_1,i_2} \,|\, \boldsymbol{x}_{i_1,i_2} = (x_{1i_1},\ x_{2i_2})^\top = (i_1 h_1,\ i_2 h_2), i_1 = 1,\ 2,\ 3,\ \cdots, m_1 + 1; i_2 = 1,\ 2,\ 3,\ \cdots,\ m_2 + 1\}$, where $h_i = \frac{1}{m_i}$, and the discrete domain Ω_h consists of $m_1 \times m_2$ cells, each of size $h_1 \times h_2$.

4.1.1 Discretization of Displacement Field $\boldsymbol{u}$ and Proposed Regularizer $\mathcal{R}^{\text{New}}(\boldsymbol{u})$

The displacement field $\boldsymbol{u}$ is discretized on the nodal grid, represented in discrete form as $\boldsymbol{u}^h = (u_1^h, u_2^h)^\top$, where u_l^h $(l = 1,\ 2)$ denotes the grid functions defined on Ω_h. To simplify notation, we define $(u_l^h)_{i_1,i_2} \triangleq u_l^h(x_{1i_1}, x_{2i_2})$, $l = 1, 2$. To simplify calculations, the grid functions u_1^h and u_2^h defined on the discrete domain Ω_h will be converted into column vectors $\boldsymbol{u}_1^h$ and $\boldsymbol{u}_2^h$ in lexicographical order, respectively,

$$
\begin{aligned}
\boldsymbol{u}_1^h = \{&(u_1^h)_{1,1}, (u_1^h)_{2,1}, \cdots, (u_1^h)_{m_1+1,1}, (u_1^h)_{1,2}, \cdots, \\
&(u_1^h)_{m_1+1,2}, \cdots, (u_1^h)_{1,m_2}, \cdots, (u_1^h)_{m_1+1,m_2+1}\}^\top, \\
\boldsymbol{u}_2^h = \{&(u_2^h)_{1,1}, (u_2^h)_{2,1}, \cdots, (u_2^h)_{m_1+1,1}, (u_2^h)_{1,2}, \cdots, \\
&(u_2^h)_{m_1+1,2}, \cdots, (u_2^h)_{1,m_2}, \cdots, (u_2^h)_{m_1+1,m_2+1}\}^\top,
\end{aligned}
$$

here $\boldsymbol{u}_l^h \in \mathbb{R}^{N_1}$, $l = 1, 2$, $N_1 = (m_1 + 1) \times (m_2 + 1)$, and set $U^h = (\boldsymbol{u}_1^h, \boldsymbol{u}_2^h)^\top$.

In solving model (6), when addressing the first-order variation of the new regularizer $\mathcal{R}^{\text{New}}(\boldsymbol{u})$, which involves the fractional α-order gradient operator ∇^α and the fractional α-order divergence operator $\nabla^\alpha \cdot$, we introduce discrete forms

for both operators. Let

$$(\nabla^\alpha)^h(U^h)_{i_1,i_2} = ((\nabla^\alpha)^h(\boldsymbol{u}_1^h)_{i_1,i_2}, (\nabla^\alpha)^h(\boldsymbol{u}_2^h)_{i_1,i_2})^\top$$

denote the discrete from of the fractional α-order gradient operator ∇^α at every pixel (x_{1i_1}, x_{2i_2}), where

$$(\nabla^\alpha)^h(\boldsymbol{u}_l^h)_{i_1,i_2} = \left(\frac{(\partial^\alpha)^h(\boldsymbol{u}_l^h)_{i_1,i_2}}{\partial x_1^\alpha}, \frac{(\partial^\alpha)^h(\boldsymbol{u}_l^h)_{i_1,i_2}}{\partial x_2^\alpha}\right)^\top.$$

Let's first consider the discretization of $\alpha-$order fractional derivatives along the x_1-direction on Ω_h by using the shifted Grünwald approximation method [16, 45–47]

$$\begin{aligned} &D^\alpha_{[a,b]}g(x_{1i_1}, x_{2i_2}) \\ &= \frac{\delta_0^\alpha g(x_{1i_1}, x_{2i_2})}{h^\alpha} + o(h) \\ &= \frac{1}{2}\left(\frac{\delta_-^\alpha g(x_{1i_1}, x_{2i_2})}{h^\alpha} + \frac{\delta_+^\alpha g(x_{1i_1}, x_{2i_2})}{h^\alpha}\right) + o(h) \\ &= \frac{1}{2}(h^{-\alpha}\sum_{m=0}^{i_1+1}\rho_m^{(\alpha)} g_{i_1-m+1}^{i_2} \\ &\quad + h^{-\alpha}\sum_{m=0}^{m_1-i_1+2}\rho_m^{(\alpha)} g_{i_1+m-1}^{i_2}) + o(h), \end{aligned} \tag{14}$$

where $g_s^{i_2} := g_{s,i_2}$, $\rho_m^{(\alpha)} = (-1)^m\binom{\alpha}{m}$, $m = 1, 2, \cdots, m_1$, m_1+1 and $\rho_0^{(\alpha)} = 1$; $\rho_m^{(\alpha)} = (1-\frac{1+\alpha}{m})\rho_{m-1}^{(\alpha)}$. Here, the zero boundary value condition is applied to $\partial\Omega$ as in previous studies [20, 45]. Therefore, based on formula (14), the discrete form of the α-order fractional derivative at (x_{1i_1}, x_{2i_2}) with a fixed point x_{2i_2} along the x_1-direction can be expressed as follows:

$$\begin{bmatrix} \delta_0^\alpha g(x_{11}, x_{2i_2}) \\ \delta_0^\alpha g(x_{12}, x_{2i_2}) \\ \vdots \\ \delta_0^\alpha g(x_{1m_1-1}, x_{2i_2}) \\ \delta_0^\alpha g(x_{1m_1}, x_{2i_2}) \\ \delta_0^\alpha g(x_{1m_1+1}, x_{2i_2}) \end{bmatrix} = \frac{1}{2}h^{-\alpha}\begin{bmatrix} 2\rho_1^{(\alpha)} & \rho_0^{(\alpha)}+\rho_2^{(\alpha)} & \rho_3^{(\alpha)} & \cdots & \rho_{m_1+1}^{(\alpha)} \\ \rho_0^{(\alpha)}+\rho_2^{(\alpha)} & 2\rho_1^{(\alpha)} & \rho_0^{(\alpha)}+\rho_2^{(\alpha)} & \cdots & \rho_{m_1}^{(\alpha)} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \rho_{m_1}^{(\alpha)} & \rho_{m_1-1}^{(\alpha)} & \cdots & 2\rho_1^{(\alpha)} & \rho_0^{(\alpha)}+\rho_2^{(\alpha)} \\ \rho_{m_1+1}^{(\alpha)} & \rho_{m_1}^{(\alpha)} & \cdots & \rho_0^{(\alpha)}+\rho_2^{(\alpha)} & 2\rho_1^{(\alpha)} \end{bmatrix}\begin{bmatrix} g_1^{i_2} \\ g_2^{i_2} \\ \vdots \\ g_{m_1-1}^{i_2} \\ g_{m_1}^{i_2} \\ g_{m_1+1}^{i_2} \end{bmatrix}$$
$$\triangleq A_{m_1+1,\alpha}\boldsymbol{G}, \tag{15}$$

where $\boldsymbol{G} = [g_1^{i_2}, g_2^{i_2}, \cdots, g_{m_1-1}^{i_2}, g_{m_1}^{i_2}, g_{m_1+1}^{i_2}]^\top$.

In this way, the discrete form of the α-order fractional derivative $\frac{(\partial^\alpha)^h u_l^h}{\partial x_1^\alpha}$ along the x_1-direction at the nodes $u_{l\,i,j}^h$ $(i = 1, 2, \cdots, m_1+1;\ j = 1, 2, \cdots, m_2+1)$ can be expressed as the product of the matrix and vector as follows:

$$\frac{(\partial^\alpha)^h\boldsymbol{u}_l^h}{\partial x_1^\alpha} = (I_{m_2+1}\otimes A_{m_1+1,\alpha})\boldsymbol{u}_l^h \triangleq \boldsymbol{A}_{x_1}^{(\alpha)}\boldsymbol{u}_l^h;$$

where the symbol $\otimes$ represents the Kronecker product. Similarly, the discrete form of the α-order fractional derivative $\frac{(\partial^\alpha)^h\boldsymbol{u}_l^h}{\partial x_2^\alpha}$ along the x_2-direction can be written as follows:

$$\frac{(\partial^\alpha)^h\boldsymbol{u}_l^h}{\partial x_2^\alpha} = (A_{m_2+1,\alpha}\otimes I_{m_1+1})\boldsymbol{u}_l^h \triangleq \boldsymbol{A}_{x_2}^{(\alpha)}\boldsymbol{u}_l^h.$$

Thus, we can get

$$(\nabla^\alpha)^h\boldsymbol{u}_l^h = \begin{bmatrix}\frac{(\partial^\alpha)^h\boldsymbol{u}_l^h}{\partial x_1^\alpha} \\ \frac{(\partial^\alpha)^h\boldsymbol{u}_l^h}{\partial x_2^\alpha}\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}_{x_1}^{(\alpha)} \\ \boldsymbol{A}_{x_2}^{(\alpha)}\end{bmatrix}\boldsymbol{u}_l^h \triangleq \boldsymbol{B}\boldsymbol{u}_l^h.$$

Furthermore, we have

$$(\nabla^\alpha)^h U^h = \begin{bmatrix}(\nabla^\alpha)^h & 0 \\ 0 & (\nabla^\alpha)^h\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_1^h \\ \boldsymbol{u}_2^h\end{bmatrix} = \begin{bmatrix}\boldsymbol{B} & 0 \\ 0 & \boldsymbol{B}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_1^h \\ \boldsymbol{u}_2^h\end{bmatrix} \triangleq \boldsymbol{C}U^h$$

Next the proposed regularizer (3) is considered

$$\mathcal{R}^{\text{new}}(\boldsymbol{u}) = \int_\Omega \mathcal{R}[\boldsymbol{u}]d\boldsymbol{x}\ ,$$

where

$$\mathcal{R}[\boldsymbol{u}] = \sqrt{1+|\nabla^\alpha\boldsymbol{u}|^2} - 1\ .$$

According to the discrete form of the α-order fractional gradient operator ∇^α derived above, the discrete version of $\mathcal{R}[\boldsymbol{u}]$ can be written as:

$$\mathcal{R}^{\text{h}}[U^h] = \sqrt{1+(U^h)^\top\boldsymbol{C}^\top\boldsymbol{C}U^h} - 1\ .$$

Additionally, by applying the midpoint quadrature rule to approximate the integral, we can discretize the proposed reg-

ularizer as follows:

$$\mathcal{R}^{\text{hnew}}(U^h) = h_d(\sqrt{1 + (U^h)^\top \boldsymbol{C}^\top \boldsymbol{C} U^h} - 1)\,, \tag{16}$$

where $h_d = h_1 h_2$. We can further derive the discrete form $d\mathcal{R}^{\text{hnew}}(U^h)$ for the first-order variation of the regularizer $\mathcal{R}^{\text{new}}(\boldsymbol{u})$

$$d\mathcal{R}^{\text{hnew}}(U^h) = h_d \frac{\boldsymbol{C}^\top \boldsymbol{C} U^h}{\sqrt{1 + (U^h)^\top \boldsymbol{C}^\top \boldsymbol{C} U^h}}. \tag{17}$$

To fully leverage standard optimization methods, it is essential to compute the second-order variations of the proposed regularizer. To enhance computational efficiency, we employ the "frozen coefficients" method, which is commonly used in variational approaches involving TV operators [48–50]. The main idea is to linearize $d\mathcal{R}^{\text{hnew}}(U^h)$ using the known iteration value $U^{h(k)}$ from the previous step. Consequently, the second-order variation $d^2\mathcal{R}^{\text{hnew}}(U^h)$ of the proposed discrete regularizer $\mathcal{R}^{\text{hnew}}(U^h)$ can be expressed as follows:

$$d^2\mathcal{R}^{\text{hnew}}(U^h) = h_d \frac{\boldsymbol{C}^\top \boldsymbol{C}}{\sqrt{1 + (U^{h(k)})^\top \boldsymbol{C}^\top \boldsymbol{C} U^{h(k)}}}. \tag{18}$$

4.1.2 Discretizing Distance Measure $\mathcal{D}$

To find the gray value at any spatial location not on the grid points, image interpolation must be applied to the given discrete images. To effectively utilize the optimization method, cubic B-spline interpolation is employed during the registration process. The continuous smooth approximations of F and M are denoted as $\mathcal{F}$ and $\mathcal{M}$, respectively. Let

$$\begin{aligned}\boldsymbol{x}_1^h = \{&x_{11,1}, x_{12,1}, \cdots, x_{1m_1+1,1}, x_{11,2}, \cdots, x_{1m_1+1,2}, \cdots,\\ &x_{11,m_2+1}, \cdots, x_{1m_1+1,m_2+1}\}^\top,\\ \boldsymbol{x}_2^h = \{&x_{21,1}, x_{22,1}, \cdots, x_{2m_1+1,1}, x_{21,2}, \cdots, x_{2m_1+1,2}, \cdots,\\ &x_{21,m_2+1}, \cdots, x_{2m_1+1,m_2+1}\}^\top,\end{aligned}$$

and $X^h = [\boldsymbol{x}_1^h; \boldsymbol{x}_2^h] \in \mathbb{R}^{2N_1 \times 1}$. Using the cell-centered partition shown in Fig. 1a and the midpoint rule, we obtain:

$$\begin{aligned}\mathcal{D}(\boldsymbol{u}) &= \frac{1}{2}\int_\Omega (M(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})) - F(\boldsymbol{x}))^2 d\boldsymbol{x}\\ &= \frac{h_1 h_2}{2}\sum_{i_1=1}^{m_1}\sum_{i_2=1}^{m_2}(M(\boldsymbol{x}_{i_1+1/2,i_2+1/2} + \boldsymbol{u}(\boldsymbol{x}_{i_1+1/2,i_2+1/2}))\\ &\quad - F(\boldsymbol{x}_{i_1+1/2,i_2+1/2}))^2,\end{aligned}$$

Let $\vec{F} = \mathcal{F}(PX^h) \in \mathbb{R}^{N\times 1}$ represent the discretized reference image, and $\vec{M}(U^h) = \mathcal{M}(PX^h + PU^h) \in \mathbb{R}^{N\times 1}$ represents the discrete form of the transformed template. Here, $P \in \mathbb{R}^{2N\times 2N_1}$ as a matrix for averaging operations, is used to enable the conversion from the nodal grid to the cell-centered grid. Therefore, the discretized version of $\mathcal{D}(\boldsymbol{u})$ can be expressed as:

$$\mathcal{D}^h(U^h) = \frac{1}{2}h_1 h_2(\vec{M}(U^h) - \vec{F})^\top(\vec{M}(U^h) - \vec{F}), \tag{19}$$

Additionally, the first-order variation of $\mathcal{D}^h(U^h)$ regarding U^h can be written as:

$$d\mathcal{D}^h(U^h) = h_1 h_2 P^\top(\vec{M}_{U^h})^\top(\vec{M}(U^h) - \vec{F}), \tag{20}$$

where

$$\vec{M}_{U^h} = \frac{\partial \vec{M}(U^h)}{\partial U^h} = \frac{\partial \mathcal{M}(PX^h + PU^h)}{\partial(PX^h + PU^h)}.$$

Moreover, we can directly compute the second-order variation of $\mathcal{D}^h(U^h)$ as follows:

$$\begin{aligned}d^2\mathcal{D}^h(U^h) = h_1 h_2 P^\top[&(\vec{M}_{U^h})^\top \vec{M}_{U^h}\\ &+ \sum_{i=1}^{N} r_i(U^h)\nabla^2 r_i(U^h)]P\ ,\end{aligned} \tag{21}$$

where $r(U^h) = \vec{M}(U^h) - \vec{F} \in \mathbb{R}^N$. To enhance computational efficiency, we will disregard the high-order term in formula (21), and approximate the second-order variation $d^2\mathcal{D}^h(U^h)$ of $\mathcal{D}^h(U^h)$ as follows:

$$d^2\mathcal{D}^h(U^h) = h_1 h_2 P^\top(\vec{M}_{U^h})^\top \vec{M}_{U^h} P. \tag{22}$$

4.1.3 Discretization of the Constrained Regularizer $\mathcal{R}_1(\boldsymbol{u})$.

To preserve local details and ensure that registration results accurately reflect local deformations, we control the area change of the smallest measurable unit(i.e., a pixel) under the discrete transformation $\boldsymbol{y}$. As is well known, controlling pixel area changes is a challenging task. To ensure that the transformation $\boldsymbol{y}$ remains diffeomorphic, we employ a partitioning method similar to that described in references [10, 24]. After applying the transformation $\boldsymbol{y}$, the area of the set A can be expressed using the following formula:

$$\text{area}(\boldsymbol{y}(A)) = \int_{\boldsymbol{y}(A)} d\boldsymbol{x} = \int_A \det\nabla \boldsymbol{y}\, d\boldsymbol{x}.$$

It should be pointed out that the latter equation assumes that $\boldsymbol{y}$ possesses sufficient regularity, which is a crucial prerequisite. Thus, the discretization approach we use is directly based on

$\int_{\boldsymbol{y}(A)} d\boldsymbol{x}$ and measures the area covered through the transformed vertices. A rectangle formed by four nodes is called a pixel (see Fig. 1b). This pixel is subdivided into four triangles, and our discrete transformation model is a continuous vector field that remains linear on each triangle. In the yellow area of Fig. 1b, the nodes are labeled B_1, B_2, B_3, B_4, with the center point $B_5 := (B_1 + B_2 + B_3 + B_4)/4$. The four triangles are represented as follows: $F(1,2,5) := \triangle(B_1, B_2, B_5)$, $F(1,4,5) := \triangle(B_1, B_4, B_5)$, $F(3,4,5) := \triangle(B_3, B_4, B_5)$, $F(2,3,5) := \triangle(B_2, B_3, B_5)$. Using the triangle $F(1,2,5)$ from Fig. 1b as an example, we discuss the discretization of the area term. The computation for the other three triangles follows a similar method. To measure the area of the triangle $F(1,2,5)$, we use the projector F_k to extract the vertex positions, i.e., $F_k\boldsymbol{y} := \boldsymbol{y}(B_k)$, where $k \in \{1,2,5\}$, compute the difference vectors, and determine the area using the rule of Sarrus:

$$F_{(1,2,5)}\boldsymbol{y} = [\boldsymbol{p}^1, \boldsymbol{p}^2] = [F_1\boldsymbol{y} - F_5\boldsymbol{y}, F_2\boldsymbol{y} - F_5\boldsymbol{y}],$$

Let $\boldsymbol{p}^1 = (p_1^1, p_2^1)$, $\boldsymbol{p}^2 = (p_1^2, p_2^2)$, then

$$\begin{aligned} A \triangleq A(F_{(1,2,5)}\boldsymbol{y}) &= \frac{1}{2}\det([\boldsymbol{p}^1, \boldsymbol{p}^2]) \\ &= \frac{1}{2}\begin{vmatrix} p_1^1 & p_2^1 \\ p_1^2 & p_2^2 \end{vmatrix} = \frac{1}{2}(p_1^1 p_2^2 - p_2^1 p_1^2), \end{aligned}$$

$\varphi(x) = ((x-1)^2/x)^2$, therefore, the discrete form of the constrained regularizer $\mathcal{R}_1(\boldsymbol{u})$ on the triangle $F(1,2,5)$ is:

$$\mathcal{R}_{1,1}^{h,i}(\boldsymbol{u}) \triangleq \mathcal{R}_{1,1}^{h,i}(\boldsymbol{x}+\boldsymbol{u}) = \mathcal{R}_{1,F(1,2,5)}^{h,i}(\boldsymbol{y}) = \varphi(A(F_{(1,2,5)}\boldsymbol{y})).$$

Therefore, the discrete constrained regularization term can be expressed as:

$$\mathcal{R}_1^h(\boldsymbol{u}) = h_1 h_2 \sum_{i=1}^{N_1}\sum_{j=1}^{4} \mathcal{R}_{1,j}^{h,i}. \tag{23}$$

Furthermore, the first-order variation of the discrete constrained regularizer can be obtained:

$$d\mathcal{R}_1^h(\boldsymbol{u}) = h_1 h_2 \sum_{i=1}^{N_1}\sum_{j=1}^{4} d\mathcal{R}_{1,j}^{h,i}, \tag{24}$$

where

$$d\mathcal{R}_{1,1}^{h,i} = d\varphi(A)\, dA\, F_{(1,2,5)},$$

$$d\varphi(A) = 2(A+1)((A-1)/A)^3,$$

$$dA = \frac{1}{2}\begin{bmatrix} p_2^2 & -p_1^2 \\ -p_2^1 & p_1^1 \end{bmatrix}.$$

To improve computational efficiency, we approximate the Hessian matrix of the discrete constrained regularizer using the following method:

$$d^2\mathcal{R}_1^h(\boldsymbol{u}) = h_1 h_2 \sum_{i=1}^{N_1}\sum_{j=1}^{4} d^2\mathcal{R}_{1,j}^{h,i}, \tag{25}$$

where

$$d^2\mathcal{R}_{1,1}^{h,i} = F_{(1,2,5)}^{\top}\, dA\, d^2\varphi(A)\, dA\, F_{(1,2,5)},$$

and

$$d^2\varphi(A) = 2(A-1)^2(A^2+2A+3)/A^4.$$

4.2 Solving Discrete Registration Model

Based on the previously derived formula, the discrete form of the new model (6) is as follows:

$$\min_{U^h}\left\{\mathcal{J}(U^h) = \mathcal{D}^h(U^h) + \lambda\mathcal{R}^{\text{hnew}}(U^h) + \mu\mathcal{R}_1^h(U^h)\right\}. \tag{26}$$

To solve the discrete new model (26), we adopted a damped Newton method combined with the Armijo line search, which combines the fast convergence of Newton's method with the flexibility of line search, thus improving the stability and effectiveness of the optimization process. The basic idea of this method is to calculate the optimal search direction using the gradient and Hessian matrix of the objective function, in order to quickly approach a local optimal solution. To prevent instability caused by excessive step size in unfavorable conditions, a damping factor is added to adjust the step size and control the iterative process. Meanwhile, the step size is dynamically adjusted to meet specific descent conditions, ensuring that the objective function value is effectively reduced at each step. The following are the specific steps.

Starting with the previous iteration value $U^{h(k)}$ as the initial value, let

$$\begin{aligned} \mathcal{J}(U^{h(k)} + \delta_{U^h}) &\approx \tilde{\mathcal{J}}(\delta_{U^h}) \\ &= \mathcal{J}(U^{h(k)}) + (d\mathcal{J}(U^{h(k)}))^{\top}\delta_{U^h} \\ &\quad + \frac{1}{2}\delta_{U^h}{}^{\top} H(U^{h(k)})\delta_{U^h}, \end{aligned} \tag{27}$$

here, based on formulas (17), (18), (20), (22), (24) and (25), the Jacobian matrix $d\mathcal{J}(U^{h(k)})$ and the approximate Hessian matrix $H(U^{h(k)})$ of the objective function $\mathcal{J}(U^h)$ at the iteration value $U^{h(k)}$ are expressed as follows:

$$d\mathcal{J}(U^{h(k)}) = d\mathcal{D}^h(U^{h(k)}) + \lambda d\mathcal{R}^{\text{hnew}}(U^{h(k)})$$

$$+\mu d\mathcal{R}_1^h(U^{h(k)}), \tag{28}$$

$$H(U^{h(k)}) = d^2\mathcal{D}^h(U^{h(k)}) + \lambda d^2\mathcal{R}^{\text{hnew}}(U^{h(k)}) + \mu d^2\mathcal{R}_1^h(U^{h(k)}). \tag{29}$$

Assuming the first variation of the quadratic functional $\tilde{\mathcal{J}}(\delta_{U^h})$ is zero, we can obtain:

$$H(U^{h(k)})\delta_{U^h} = -d\mathcal{J}(U^{h(k)}). \tag{30}$$

Typically, the approximated Hessian matrix $H(U^{h(k)})$ is positive definite, and the quasi-Newton equation (30) can be solved using the preconditioned conjugate gradient method. Furthermore, the Newton direction δ_{U^h} can be obtained:

$$\delta_{U^h} = -H(U^{h(k)})^{-1} d\mathcal{J}(U^{h(k)}). \tag{31}$$

The search direction calculated from equation (31) may lead to an excessively large step size in some cases, causing the iteration point to overshoot the optimal solution or diverge. When far from the optimal point, Newton's method may exhibit instability or even fail to converge. Additionally, when the Hessian matrix is not positive definite, the Newton direction may not be a descent direction. In such cases, introducing a damping factor to control the step size can improve the global stability and robustness of Newton's method, enhancing its adaptability to complex optimization problems. To enhance the robustness and global convergence of the damped Newton method, and to ensure that each iteration effectively reduces the value of the objective function while avoiding divergence or oscillation, we introduced the Armijo line search method with dynamic step size control. This combination allows the algorithm to maintain convergence speed while effectively dealing with potential numerical instability issues when solving complex optimization problems. The specific steps of the algorithm are summarized in Algorithm 1:

The choice of the initial point has a significant impact on the convergence, convergence rate, and the likelihood of finding the global optimal solution in the Newton method. A well-chosen initial point can accelerate convergence and improve the reliability of the algorithm, while a poor choice may cause the algorithm to diverge or converge to a local minimum. To reduce the sensitivity of the Newton method to the initial point, we combine the multilevel method with Algorithm 1 to solve the discrete functional model (26). The multilevel method is a key technique in image registration, aimed at accelerating convergence and improving the stability of the registration process. By progressively aligning images at different resolutions or scales, this method reduces alignment errors step by step, avoids getting trapped in local minima, and ultimately enhances the accuracy of global registration. First, Algorithm 1 is applied at the coarsest level to solve the discrete functional (26); at each level, the solution obtained from the coarse grid serves as the initial value for the next finer grid using the interpolation operator I_h^H, and the procedure is repeated until the termination conditions are met. The specific procedures of the multilevel method are outlined in Algorithm 2.

Algorithm 1: Damped Newton Method with Armijo Line Search

1. **Initialization**:
   - Choose the initial point $\boldsymbol{u}_0$;
   - Set the damping factor σ_0(usually between (0, 1));
   - Determine the parameter c(e.g., $1e-4$);
   - Set the maximum number of iterations $k_{\max}$ and the convergence threshold ϵ;
2. **Iteration Process**:
   For each step $k = 0, 1, 2, \cdots, k_{\max}$:
   - Utilize equations (28) and (29) to compute the gradient $d\mathcal{J}(\boldsymbol{u}_k)$ and Hessian matrix $H(\boldsymbol{u}_k)$;
   - Calculate the Newton direction $\delta_{\boldsymbol{u}_k}$ using formula (31);
   - Set the initial step size: $\sigma = \sigma_0$;
3. **Armijo Line Search**:
   - Check the Armijo condition:
   $$\mathcal{J}(\boldsymbol{u}_k + \sigma_k\delta_{\boldsymbol{u}_k}) \leq \mathcal{J}(\boldsymbol{u}_k) + c\sigma_k(d\mathcal{J}(\boldsymbol{u}_k))^\top\delta_{\boldsymbol{u}_k};$$
   - If the condition is not satisfied, reduce σ (e.g., $\sigma_{k+1} = \gamma\sigma_k$, where $0 < \gamma < 1$), and retry until a suitable step size is found or the minimum step size is reached;
4. **Update**:
   - Update the iteration value:
   $$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \sigma\delta_{\boldsymbol{u}_k};$$
5. **Convergence Check**:
   - If $\|d\mathcal{J}(\boldsymbol{u}_{k+1})\| < \epsilon$, then stop;
   - If the maximum iteration count is reached, terminate the process;
6. **Output**:
   - Return the optimal solution $\boldsymbol{u}_{k+1}$ and its corresponding objective function value $\mathcal{J}(\boldsymbol{u}_{k+1})$.

4.3 Convergence Analysis of Algorithm 1

The global convergence proof for Algorithm 1 is presented below.

Theorem 4.1 Let $\{\boldsymbol{u}_k\}$ be the sequence generated by Algorithm *1*. Suppose that the function $\mathcal{J}(\boldsymbol{u})$ has a lower bound and that for $\forall \boldsymbol{u}_0 \in \mathbb{R}^n$, $\nabla\mathcal{J}(\boldsymbol{u})$ exists and is uniformly continuous on the following level set

$$L(\boldsymbol{u}_0) = \{\boldsymbol{u} \in \mathbb{R}^n | \mathcal{J}(\boldsymbol{u}) \leq \mathcal{J}(\boldsymbol{u}_0)\}.$$

Algorithm 2: The multilevel approach used for solving model (26).

Step 1. **Initialization**: Given the coarsest level $L_1 = 3$; the finest level $L_2 = \text{ceil}(\log 2(\min(m1, m2)))$; Construct the multiscale representations for both F and M;
Step 2. **Registration at the coarse level**: If $l = L_1$, given the initial value $\boldsymbol{u}_0$ at the coarsest level, solve model (26) using Algorithm 1 to obtain the displacement field $\boldsymbol{u}_k$;
Step 3. **Registration at a finer level**: At the finer level, the initial solution U_0 is obtained by applying an interpolation operator I_h^H to the solution from the previous level, specifically $U_0 = I_h^H \boldsymbol{u}_k$. Following this, Algorithm 1 is used to solve model (26) at this level;
Step 4: Update l to $l + 1$, and return to Step 2.

If the descent direction $\delta_{\boldsymbol{u}_k}$ satisfies the following condition:

$$0 \le \theta_k \le \frac{\pi}{2} - \mu, \quad \mu \in (0, \frac{\pi}{2}),$$

where θ_k represents the angle between the descent direction $\delta_{\boldsymbol{u}_k}$ and $-\nabla\mathcal{J}(\boldsymbol{u}_k)$, then when the Armijo rule is used to determine the step size σ_k, any accumulation point $\boldsymbol{u}^*$ of $\{\boldsymbol{u}_k\}$ satisfies $\nabla\mathcal{J}(\boldsymbol{u}^*) = 0$.

Proof: Using the method of contradiction, suppose that $\boldsymbol{u}^*$ is an accumulation point of the sequence $\{\boldsymbol{u}^k\}$ and that $\nabla\mathcal{J}(\boldsymbol{u}^*) \neq 0$. According to the conditions of the theorem, we obtain:

$$\lim_{k\to+\infty} \mathcal{J}(\boldsymbol{u}_k) = \mathcal{J}(\boldsymbol{u}^*), \quad \lim_{k\to+\infty} (\mathcal{J}(\boldsymbol{u}_k) - \mathcal{J}(\boldsymbol{u}_{k+1})) = 0.$$

Furthermore, according to the Armijo rule condition in Algorithm 1, we obtain: $\lim_{k\to+\infty} c(\nabla\mathcal{J}(\boldsymbol{u}_k))^\top s_k = 0$, which further implies

$$\lim_{k\to+\infty} (\nabla\mathcal{J}(\boldsymbol{u}_k))^\top s_k = 0, \tag{32}$$

where $s_k = \sigma_k \delta_{\boldsymbol{u}_k}$. If $\lim_{k\to+\infty} \nabla\mathcal{J}(\boldsymbol{u}_k) \neq 0$, then it follows from equation (32) that: $\lim_{k\to+\infty} \|s_k\| = 0$. Suppose the step size σ_k satisfies the Armijo rule condition in Algorithm 1. Then, when $\sigma_{k-1} = \frac{\sigma_k}{\gamma}$, the inequality in Algorithm 1 can be rewritten as:

$$\mathcal{J}(\boldsymbol{u}_k + \sigma_{k-1}\delta_{\boldsymbol{u}_k}) - \mathcal{J}(\boldsymbol{u}_k) > c\sigma_{k-1}(\nabla\mathcal{J}(\boldsymbol{u}_k))^\top \delta_{\boldsymbol{u}_k}. \tag{33}$$

Note that: $\sigma_{k-1}\delta_{\boldsymbol{u}_k} = \frac{\sigma_k}{\gamma}\delta_{\boldsymbol{u}_k} = \frac{s_k}{\gamma}$, the inequality (33) can thus be rewritten as:

$$\mathcal{J}(\boldsymbol{u}_k + \frac{1}{\gamma}s_k) - \mathcal{J}(\boldsymbol{u}_k) > c(\nabla\mathcal{J}(\boldsymbol{u}_k))^\top \frac{s_k}{\gamma}. \tag{34}$$

If we set $p_k = \frac{s_k}{\|s_k\|}$, then $\frac{s_k}{\gamma} = \frac{\|s_k\|}{\gamma} p_k$. Further, let $\omega_k = \frac{\|s_k\|}{\gamma}$, since $\lim_{k\to+\infty} \|s_k\| = 0$, we can obtain $\lim_{k\to+\infty} \omega_k = 0$ and the inequality (34) can be rewritten as:

$$\frac{\mathcal{J}(\boldsymbol{u}_k + \omega_k p_k) - \mathcal{J}(\boldsymbol{u}_k)}{\omega_k} > c(\nabla\mathcal{J}(\boldsymbol{u}_k))^\top p_k. \tag{35}$$

Since $\|p_k\| = 1$, the sequence $\{p_k\}$ is bounded. Therefore, there exists a convergent subsequence, which is still denoted as $\{p_k\}$. Let $\lim_{k\to+\infty} p_k = p^*$, where $\|p^*\| = 1$. Taking the limit on both sides of inequality (35) simultaneously, we can obtain the following result:

$$(\nabla\mathcal{J}(\boldsymbol{u}^*))^\top p^* \ge c(\nabla\mathcal{J}(\boldsymbol{u}^*))^\top p^*$$

Furthermore, it can be obtained that:

$$(1 - c)(\nabla\mathcal{J}(\boldsymbol{u}^*))^\top p^* \ge 0,$$

since $0 < c < 1$, combining the above inequality, it can be further deduced that:

$$(\nabla\mathcal{J}(\boldsymbol{u}^*))^\top p^* \ge 0, \tag{36}$$

On the other hand, noting that: $p_k = \frac{s_k}{\|s_k\|} = \frac{\delta_{\boldsymbol{u}_k}}{\|\delta_{\boldsymbol{u}_k}\|}$, $(\nabla\mathcal{J}(\boldsymbol{u}_k))^\top \delta_{\boldsymbol{u}_k} < 0$, it can be further deduced that:

$$(\nabla\mathcal{J}(\boldsymbol{u}_k))^\top p_k = (\nabla\mathcal{J}(\boldsymbol{u}_k))^\top \frac{s_k}{\|s_k\|} = (\nabla\mathcal{J}(\boldsymbol{u}_k))^\top \frac{\delta_{\boldsymbol{u}_k}}{\|\delta_{\boldsymbol{u}_k}\|} < 0,$$

therefore,

$$\begin{aligned} -(\nabla\mathcal{J}(\boldsymbol{u}_k))^\top p_k &= -(\nabla\mathcal{J}(\boldsymbol{u}_k))^\top \frac{\delta_{\boldsymbol{u}_k}}{\|\delta_{\boldsymbol{u}_k}\|} = \|(\nabla\mathcal{J}(\boldsymbol{u}_k))\|\cos\theta_k \\ &\ge \|(\nabla\mathcal{J}(\boldsymbol{u}_k))\|\sin\mu, \end{aligned}$$

Taking the limit on both sides of the above inequality simultaneously, we can obtain:

$$-(\nabla\mathcal{J}(\boldsymbol{u}^*))^\top p^* \ge \|(\nabla\mathcal{J}(\boldsymbol{u}^*))\|\sin\mu > 0,$$

that is,

$$(\nabla\mathcal{J}(\boldsymbol{u}^*))^\top p^* < 0,$$

which contradicts inequality (36). The assumption is wrong, therefore, $\nabla\mathcal{J}(\boldsymbol{u}^*) = 0$. This indicates that Algorithm 1 has convergence.

Regarding Theorem 4.1, we present twelve sets of data from numerical experiments. Figure 2 illustrates the trend of the energy functional $\tilde{\mathcal{J}}(\boldsymbol{u})$ with respect to the number of iterations. Figure 2 shows that as the number of iterations increases, the energy functional gradually decreases, indicating that Algorithm 1 is convergent.

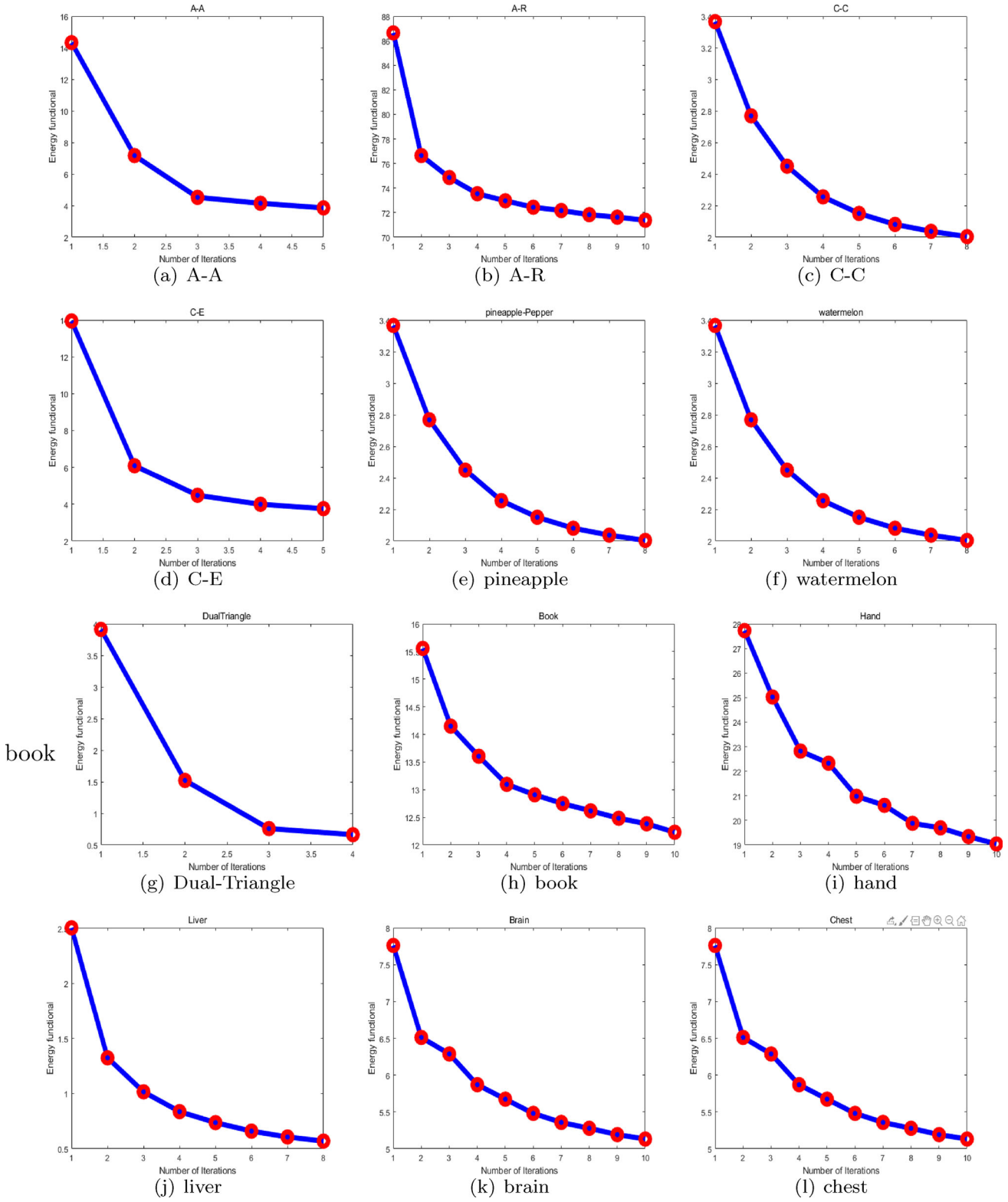
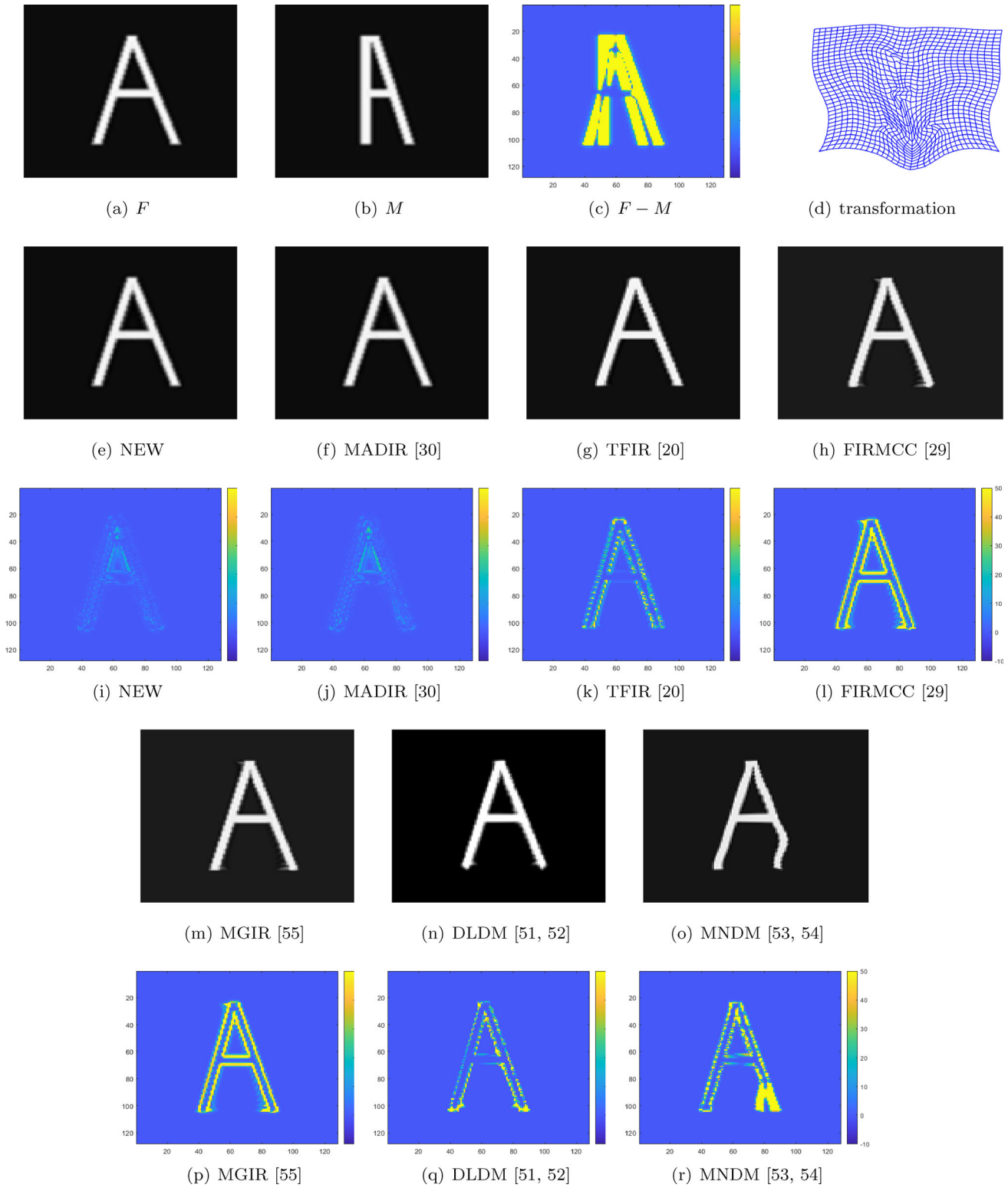


Fig. 2 The relationship between the energy functional and the number of iterations



(a) F (b) M (c) $F - M$ (d) transformation

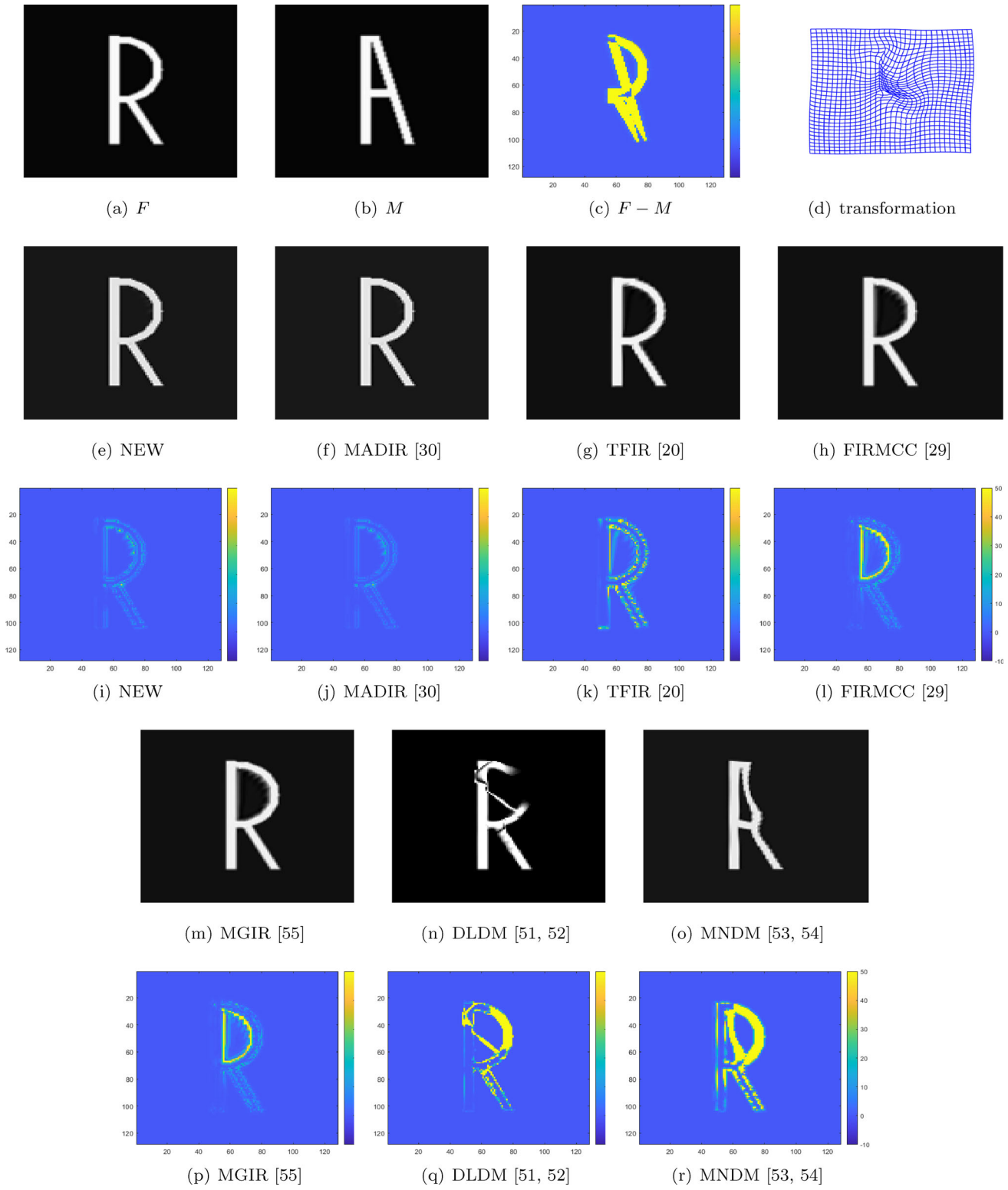
(e) NEW (f) MADIR [30] (g) TFIR [20] (h) FIRMCC [29]

(i) NEW (j) MADIR [30] (k) TFIR [20] (l) FIRMCC [29]

(m) MGIR [55] (n) DLDM [51, 52] (o) MNDM [53, 54]

(p) MGIR [55] (q) DLDM [51, 52] (r) MNDM [53, 54]

Fig. 3 The test image pairs, pre-registration differences, transformations from the new model, template images transformed using our new model and other competitive models, and post-registration differences



(a) F (b) M (c) $F - M$ (d) transformation

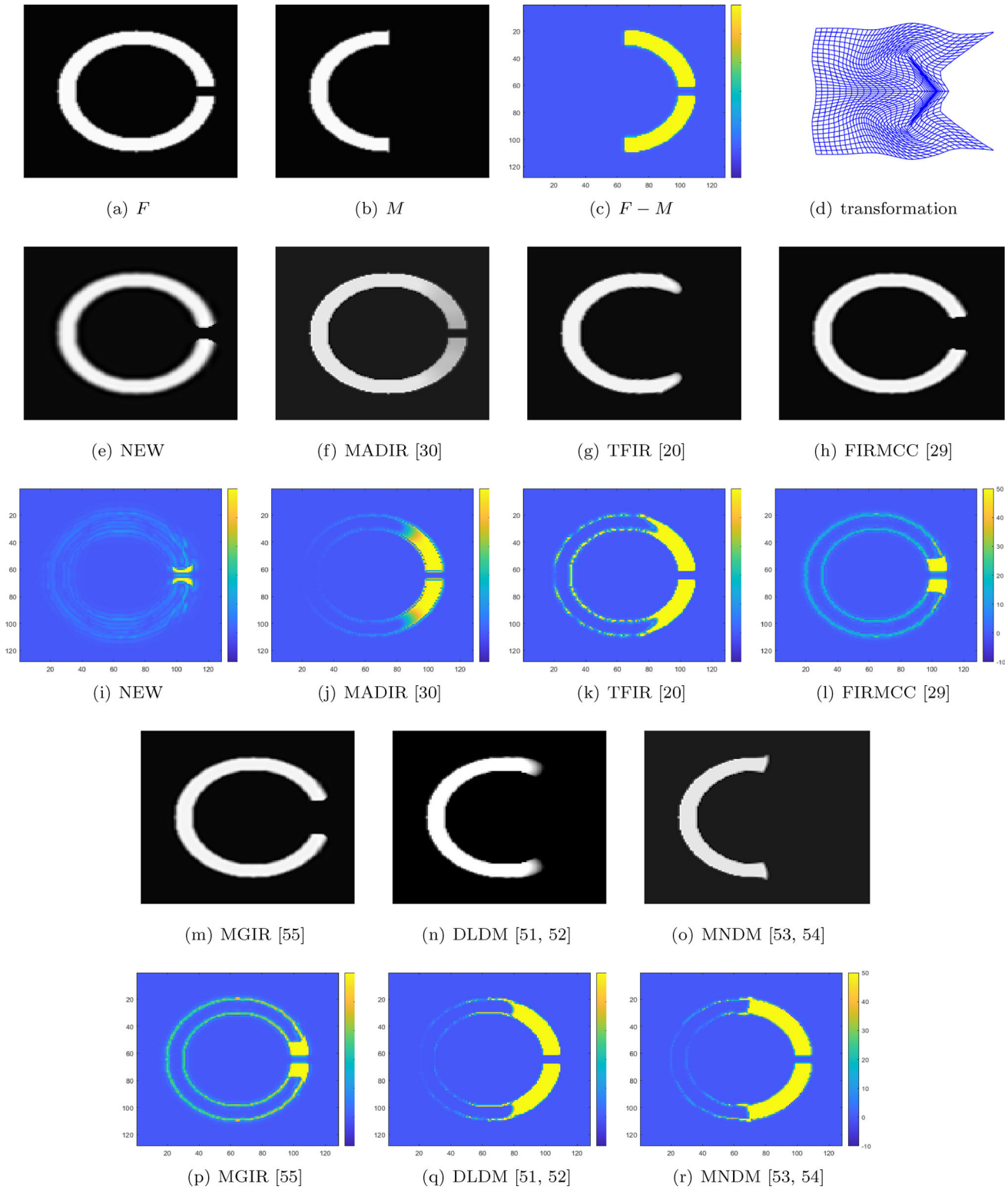
(e) NEW (f) MADIR [30] (g) TFIR [20] (h) FIRMCC [29]

(i) NEW (j) MADIR [30] (k) TFIR [20] (l) FIRMCC [29]

(m) MGIR [55] (n) DLDM [51, 52] (o) MNDM [53, 54]

(p) MGIR [55] (q) DLDM [51, 52] (r) MNDM [53, 54]

Fig. 4 The test image pairs, pre-registration differences, transformations from the new model, template images transformed using our new model and other competitive models, and post-registration differences



(a) F (b) M (c) $F - M$ (d) transformation

(e) NEW (f) MADIR [30] (g) TFIR [20] (h) FIRMCC [29]

(i) NEW (j) MADIR [30] (k) TFIR [20] (l) FIRMCC [29]

(m) MGIR [55] (n) DLDM [51, 52] (o) MNDM [53, 54]

(p) MGIR [55] (q) DLDM [51, 52] (r) MNDM [53, 54]

Fig. 5 The test image pairs, pre-registration differences, transformations from the new model, template images transformed using our new model and other competitive models, and post-registration differences

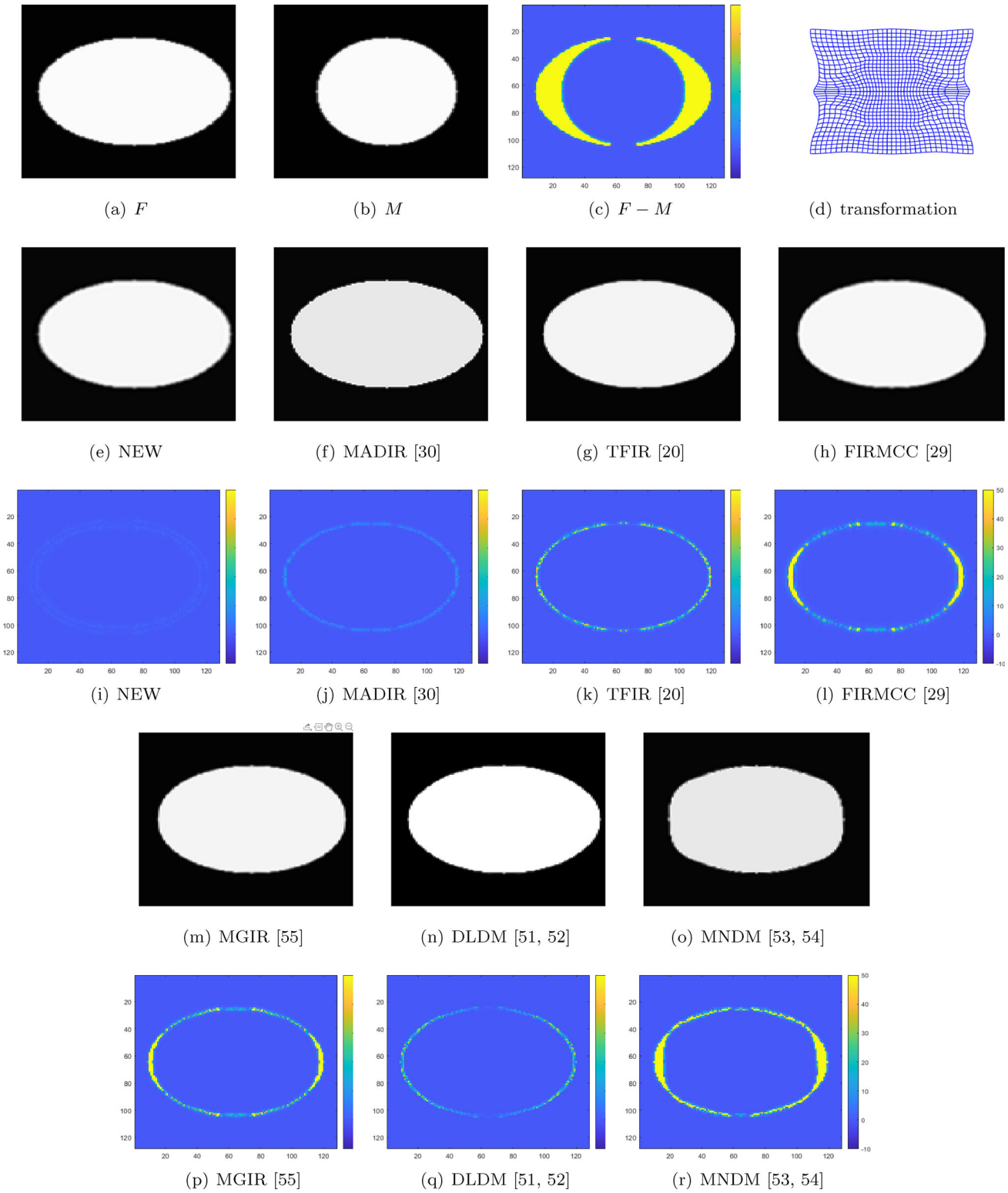


Fig. 6 The test image pairs, pre-registration differences, transformations from the new model, template images transformed using our new model and other competitive models, and post-registration differences

Table 1 Comparison of quantitative results for Example 1. $\mathcal{T}$ represents the total execution time in seconds, including the time for image generation

Model/Test images	A-A			A-R			C-C			C-E		
	$\varepsilon(\%)$	MFN	$\mathcal{T}(s)$	$\varepsilon(\%)$	MFN	$\mathcal{T}(s)$	$\varepsilon(\%)$	MFN	$\mathcal{T}(s)$	$\varepsilon(\%)$	MFN	$\mathcal{T}(s)$
NEW	0.0373	0	78	0.2512	0	75	1.1807	0	34.2	0.0020	0	33.1
MADIR [30]	0.058	0	681.3	0.091	0	587.6	3.12	0	863.3	0.0231	0	223.1
TFIR [20]	0.6530	0	19.6	1.1346	0	14.5	62.2716	0	7.2	0.3174	0	10.3
FIRMCC [29]	3.8	0	597.3	7.13	0	569.2	13.38	0	617.7	2.5	0	323.5
MGIR [55]	3.02	0	69.4	4.38	0	62.3	18.61	0	391.2	2.84	0	24.6
DLDM [51, 52]	1.4578	0	18.2	28.5086	0	11.5	67.5583	0	11.5	0.1623	0	18.4
MNDM [53, 54]	13.1615	0	42.8	55.2736	0	42.8	92.0491	0	41.3	16.8136	0	40.5

5 Numerical Experiments

We conducted three numerical tests on different image pair types-synthetic, natural, and medical images to demonstrate the performance of the proposed model. In these three tests, we compared the proposed model(NEW) with the following models: diffeomorphic log demons model (DLDM) [51, 52]; multimodality nonrigid demon model (MNDM) [53, 54]; a total fractional-order model(TFIR) [20]; fractional-order model with Cauchy–Riemann constraint(FIRMCC) [29]; multi-grid algorithm for 2D diffeomorphic model(MGIR) [55]; multiscale approach for 2D diffeomorphic model(MADIR) [30]. We evaluate the effectiveness of image registration using three common standards:

- From a visual perspective, it presents the registered template image and the differences between it and the reference image;
- The dissimilarity rate provided in references [4, 5, 20, 29] can be articulated as follows:

$$\varepsilon = \frac{\mathcal{D}(\boldsymbol{u}^*)}{\mathcal{D}(\boldsymbol{u}_0)} \times 100\%,$$

  In this context, $\mathcal{D}(\boldsymbol{u})$ is defined by equation (1), where $\boldsymbol{u}^*$ represents the optimal value determined by Algorithm 2, and $\boldsymbol{u}_0$ is set to 0.
- The mesh folding number MFN of the determinant $\det(J)$ of the Jacobian matrix J of the transformation $\varphi(\boldsymbol{x})$ described in references [20, 29, 30], where

$$J = d\varphi(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})) = \begin{bmatrix} 1 + (\boldsymbol{u}_1)_{x_1} & (\boldsymbol{u}_1)_{x_2} \\ (\boldsymbol{u}_2)_{x_1} & 1 + (\boldsymbol{u}_2)_{x_2} \end{bmatrix},$$

  and $\det(J) = (1 + (\boldsymbol{u}_1)_{x_1})(1 + (\boldsymbol{u}_2)_{x_2}) - (\boldsymbol{u}_1)_{x_2}(\boldsymbol{u}_2)_{x_1}$, MFN$= \#(\det(J) \leq 0)$.

5.1 Test Synthetic Images

In this numerical test, four synthetic image pairs with a resolution of 129×129, namely A-A, A-R, C-C, and C-E, are taken as Example 1. Figures 3, 4, 5 and 6, respectively, show the test image pairs, pre-registration image differences, the transformation generated by the new model, as well as the transformed template images and post-registration image differences produced by both the new model and other models. The quantitative results of our new model and other models for Example 1 are summarized in Table 1. For the image pair A-A, a careful examination of Fig. 3 reveals that, except for the MNDM [53, 54] model, our new model and several other models visually show relatively satisfactory registration results. Based on the differences in the registered images and the results recorded in Table 1, both our new model and the MADIR [30] model achieved better registration performance, but the MADIR [30] model took more time. For the image pair A-R with significant deformation, a close inspection of Fig. 4 reveals that, except for the DLDM [51, 52] model and MNDM [53, 54] model, both the new model and the other four models exhibit generally satisfactory registration results. From the differences in the registered images and the data in Table 1, it is observable that both proposed new model and the MADIR [30] model achieve superior registration performance without mesh folding. Although the MADIR [30] model delivers the best registration result, it takes the longest registration time. For the image pair C-C with greater deformation, a careful examination of Fig. 5, along with the registration results in Table 1, shows that our new model achieved the best registration performance in a shorter time without any mesh folding. Although the MADIR [30] model also produced relatively satisfactory registration results, it takes a longer time. For C-E image pairs, as shown in Fig. 6, except for the MNDM [53, 54] model, our new

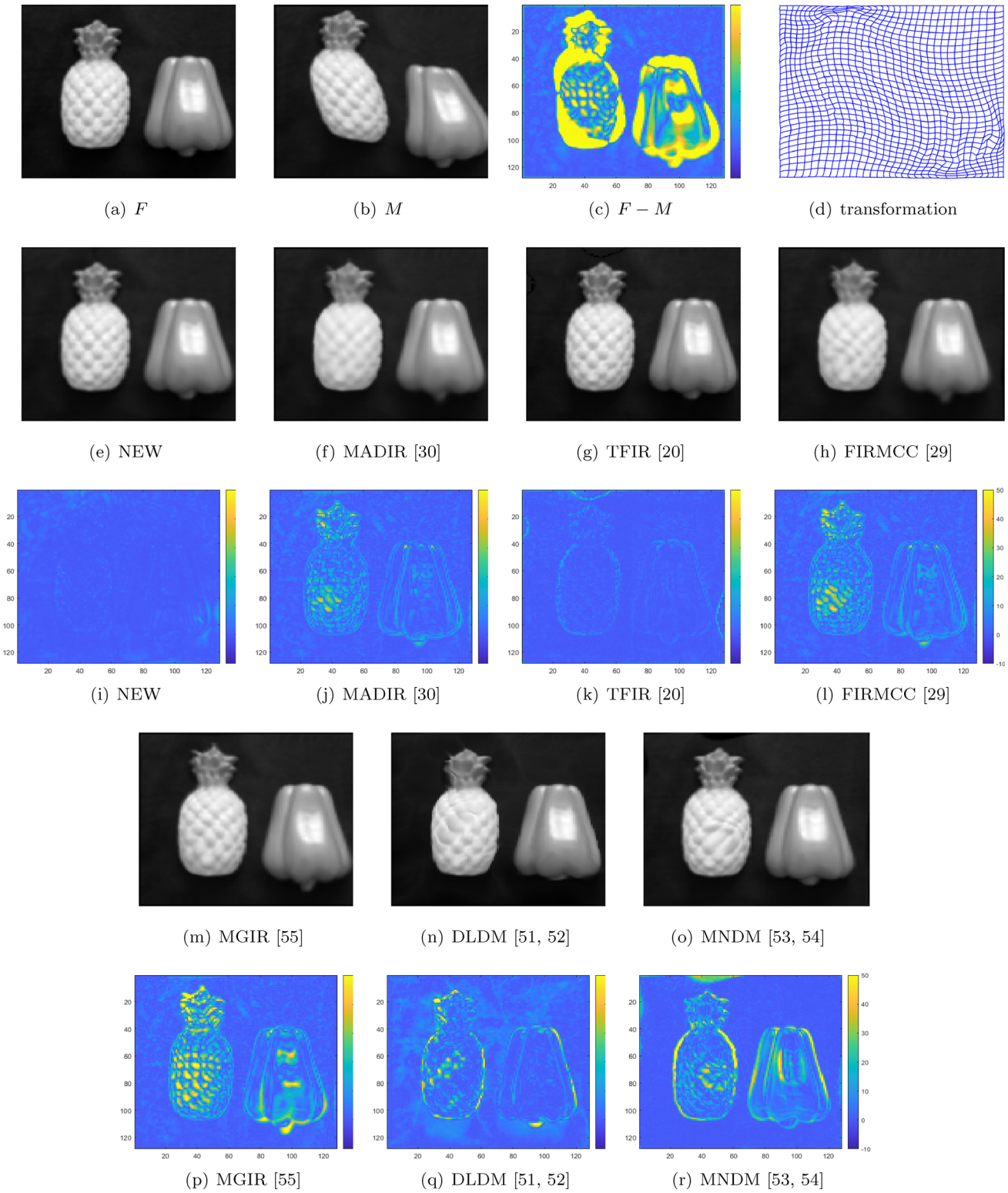
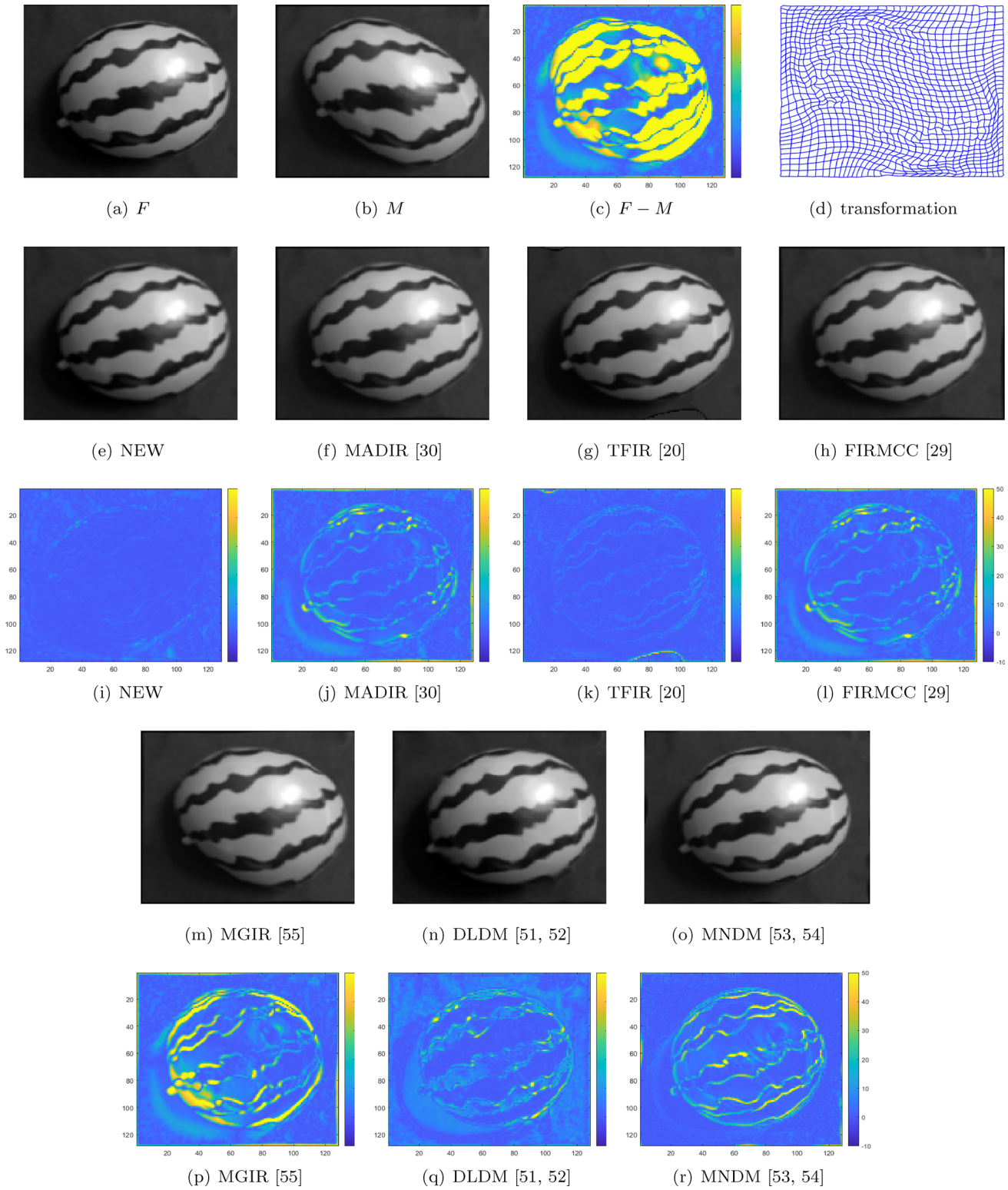


Fig. 7 The test image pairs, pre-registration differences, transformations from the new model, template images transformed using our new model and other competitive models, and post-registration differences



(a) F (b) M (c) $F - M$ (d) transformation

(e) NEW (f) MADIR [30] (g) TFIR [20] (h) FIRMCC [29]

(i) NEW (j) MADIR [30] (k) TFIR [20] (l) FIRMCC [29]

(m) MGIR [55] (n) DLDM [51, 52] (o) MNDM [53, 54]

(p) MGIR [55] (q) DLDM [51, 52] (r) MNDM [53, 54]

Fig. 8 The test image pairs, pre-registration differences, transformations from the new model, template images transformed using our new model and other competitive models, and post-registration differences

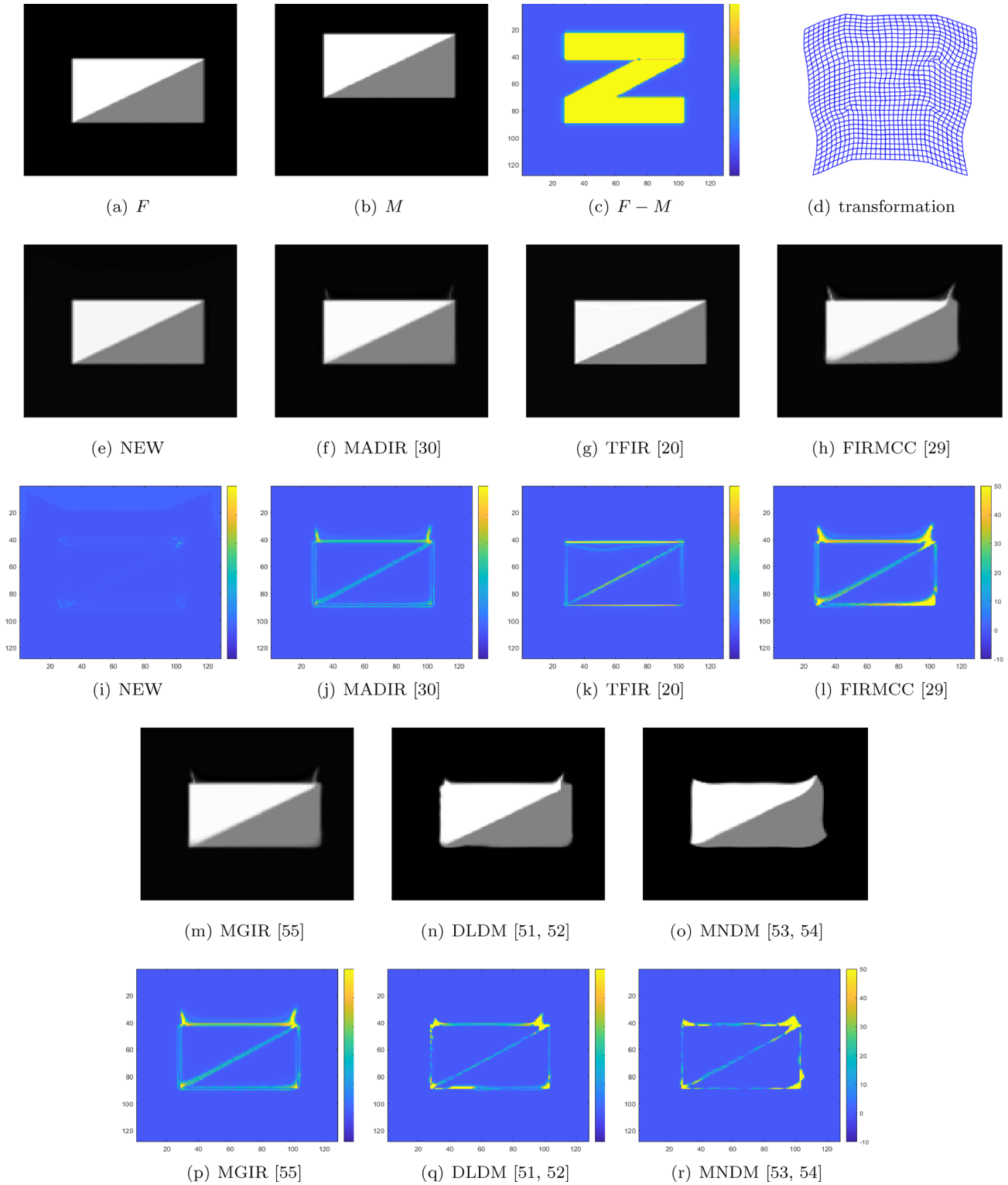
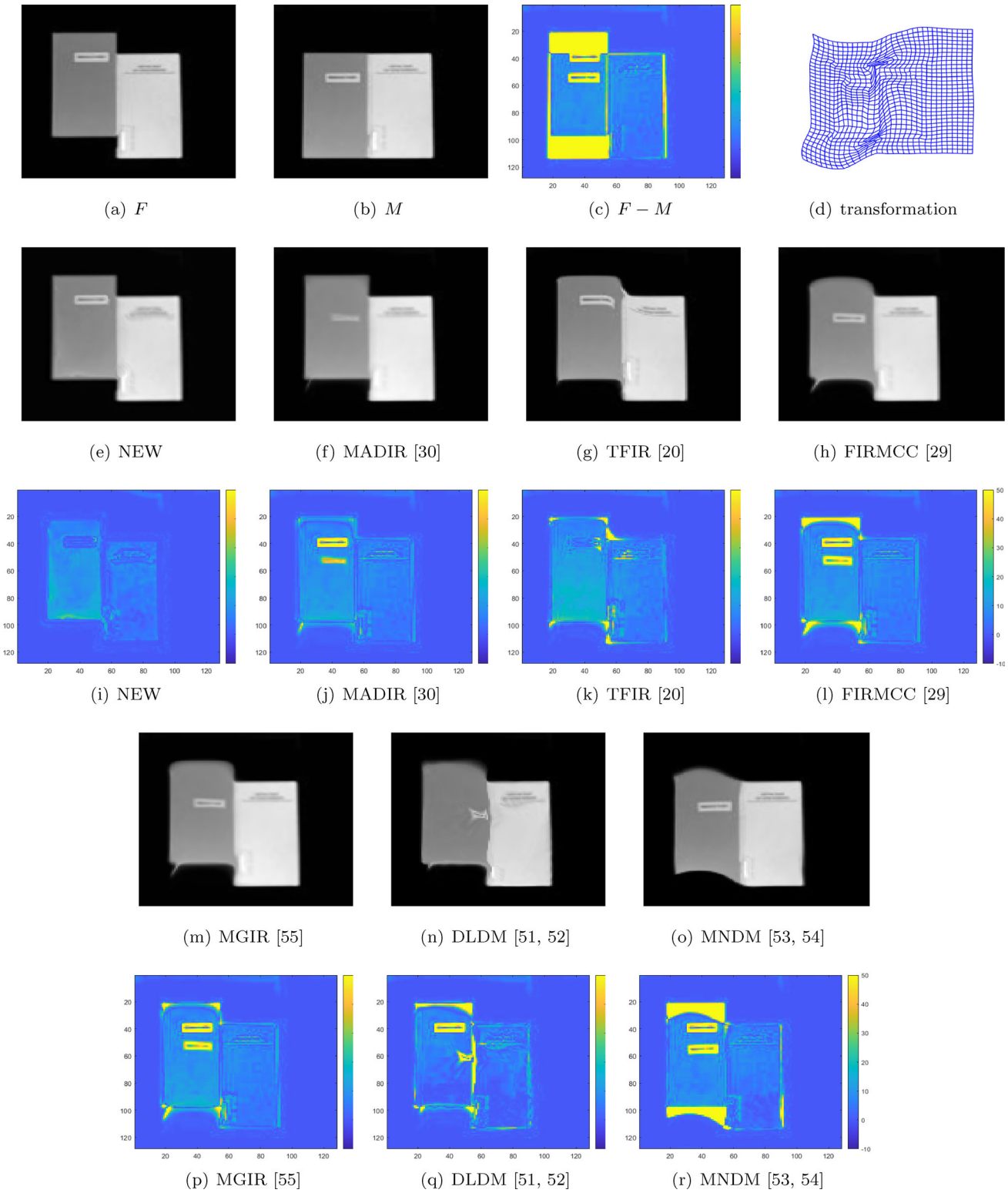


Fig. 9 The test image pairs, pre-registration differences, transformations from the new model, template images transformed using our new model and other competitive models, and post-registration differences



(a) F (b) M (c) $F - M$ (d) transformation

(e) NEW (f) MADIR [30] (g) TFIR [20] (h) FIRMCC [29]

(i) NEW (j) MADIR [30] (k) TFIR [20] (l) FIRMCC [29]

(m) MGIR [55] (n) DLDM [51, 52] (o) MNDM [53, 54]

(p) MGIR [55] (q) DLDM [51, 52] (r) MNDM [53, 54]

Fig. 10 The test image pairs, pre-registration differences, transformations from the new model, template images transformed using our new model and other competitive models, and post-registration differences

Table 2 Comparison of quantitative results for Example 2. $\mathcal{T}$ represents the total execution time in seconds, including the time for image generation

Model/Test images	Pineapple			Watermelon			Dual-Triangle			Book		
	$\varepsilon(\%)$	MFN	$\mathcal{T}(s)$	$\varepsilon(\%)$	MFN	$\mathcal{T}(s)$	$\varepsilon(\%)$	MFN	$\mathcal{T}(s)$	$\varepsilon(\%)$	MFN	$\mathcal{T}(s)$
NEW	0.1272	0	60.8	0.1123	0	54.8	0.0078	0	39.6	1.7112	0	77.6
MADIR [30]	2.26	0	363.8	3.41	0	365.1	0.1739	0	249.1	4.1233	0	371.25
TFIR [20]	0.2923	0	8.6	0.4314	0	7.9	0.3551	0	7.8	6.8922	0	11.1
FIRMCC [29]	2.64	0	80.5	3.58	0	81.3	1.5097	0	326.2	12.0538	0	469.8
MGIR [55]	6.99	0	38.3	12.33	0	38.1	0.4619	0	186.4	6.7169	0	139.9
DLDM [51, 52]	2.9409	0	13.9	2.9998	0	12.0	1.1921	0	30.5	7.1640	0	38.4
MNDM [53, 54]	5.9107	0	41.8	4.2063	0	44.6	2.0662	0	42.3	44.7230	0	40.4

model, as well as the other models, achieves favorable results. Without any folding, our new model delivers the best registration results in less time.

5.2 Test Natural Images

Four pairs of natural images with a resolution of 129×129, namely pineapple, watermelon, dual triangles, and book, are used as Example 2 in the experiment. The test image pairs, pre-registration image differences, transformations generated by the new model, template images produced by both the new model and other alternative models, as well as the post-transformation image differences, are shown in Figs. 7, 8, 9, and 10, respectively. The quantitative results for Example 2, obtained from both the new model and other models, are presented in Table 2. Among these four pairs of natural images, whether it is the pineapple and watermelon with smooth displacement fields or the dual triangles and book with non-smooth displacement fields, a detailed analysis of Figs. 7, 8, 9, and 10, and the data in Table 2 reveal that, compared to other models, our new model ensures satisfactory registration results consistently, successfully avoiding folding.

5.3 Test medical images

This test uses four pairs of medical images having a resolution of 129×129 Hand, Chest, Brain, and Liver as Example 4. The test image pairs, the differences of pre-registration images, the transformation produced by the new model and the resulting transformed template images produced by both the new and other models, as well as the differences of post-registration images are, respectively, presented in Figs. 11, 12, 13, and 14. Table 3 summarizes the quantitative results produced by the new model and other models. A detailed analysis of Figs. 11, 12, 13, and 14 reveals that, without any folding, the new model demonstrates excellent registration performance in terms of visual quality, post-registration image differences, and the data presented in Table 3.

6 Conclusion

This study introduces a diffeomorphic image registration model incorporating a vectorized fractional-order regularizer, along with its associated numerical approach. The model successfully avoids mesh folding and ensures visually satisfactory image registration, accommodating both smooth and non-smooth displacement fields. We have established the existence of solutions for the model and examined the algorithm's convergence. Comparative studies on registration results from 12 datasets with other competitive models indicate that our new model demonstrates superior performance. In future, we plan to expand this model for use in multimodal or $3D$ image registration applications.

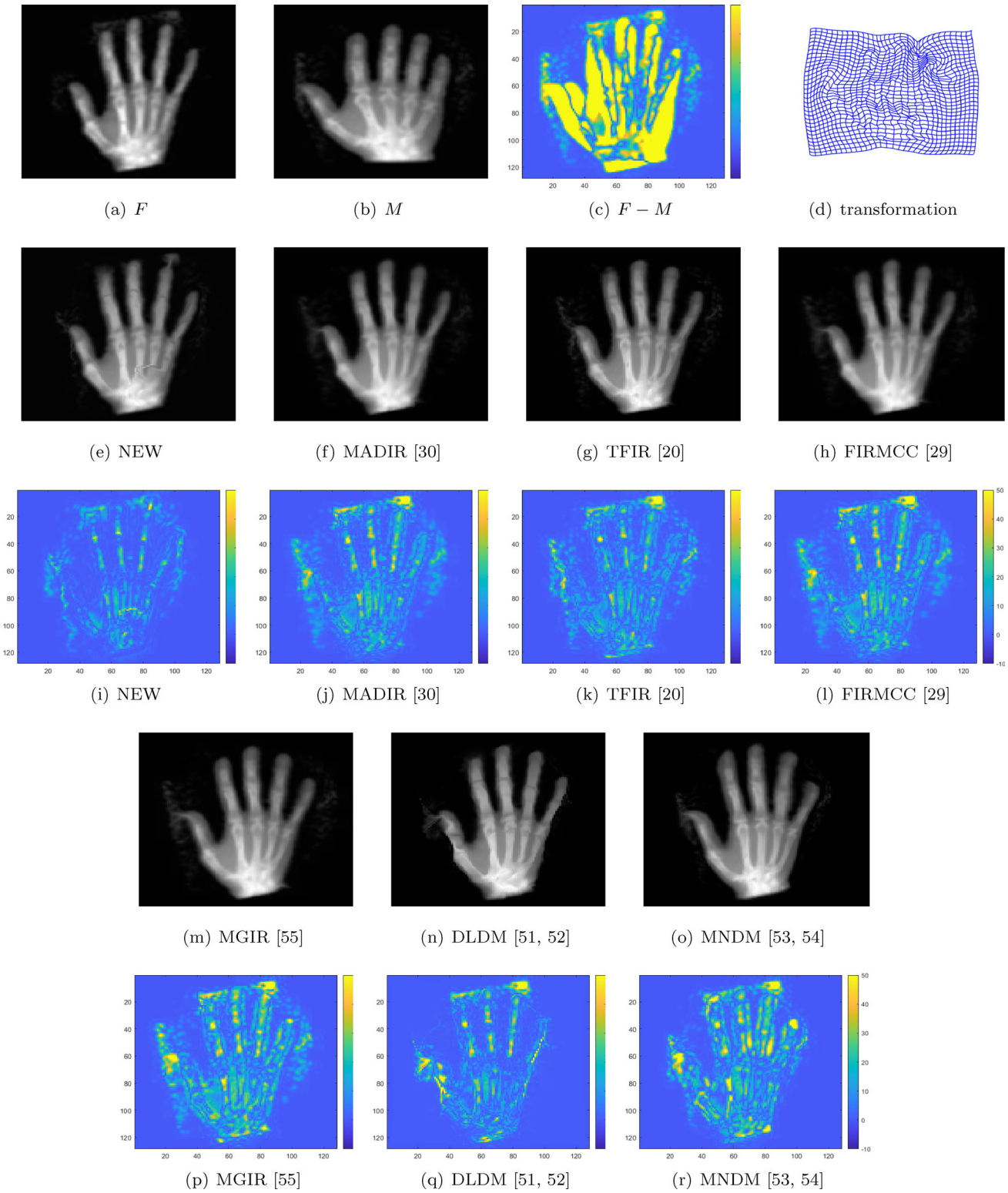
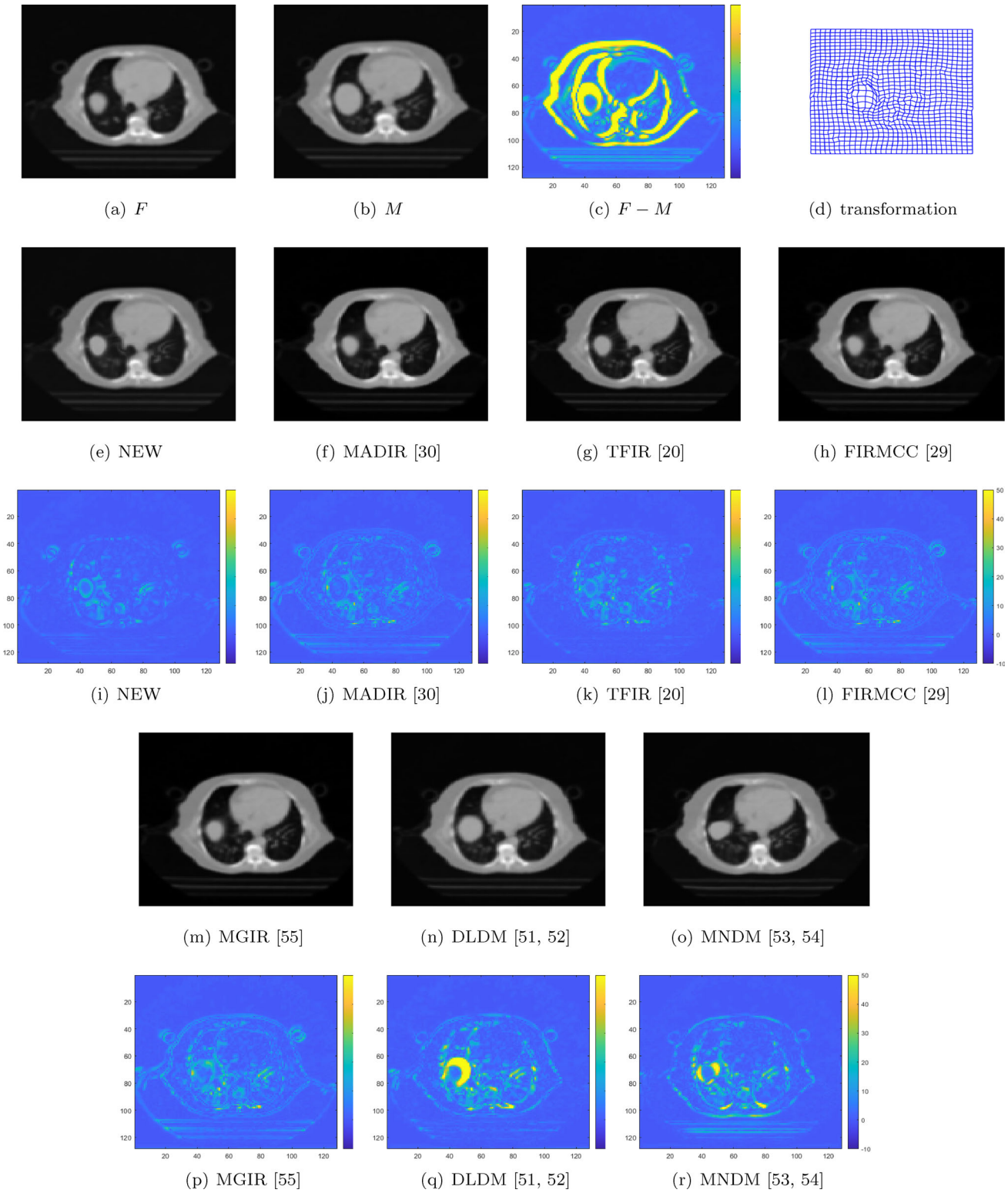


Fig. 11 The test image pairs, pre-registration differences, transformations from the new model, template images transformed using our new model and other competitive models, and post-registration differences



(a) F (b) M (c) $F - M$ (d) transformation

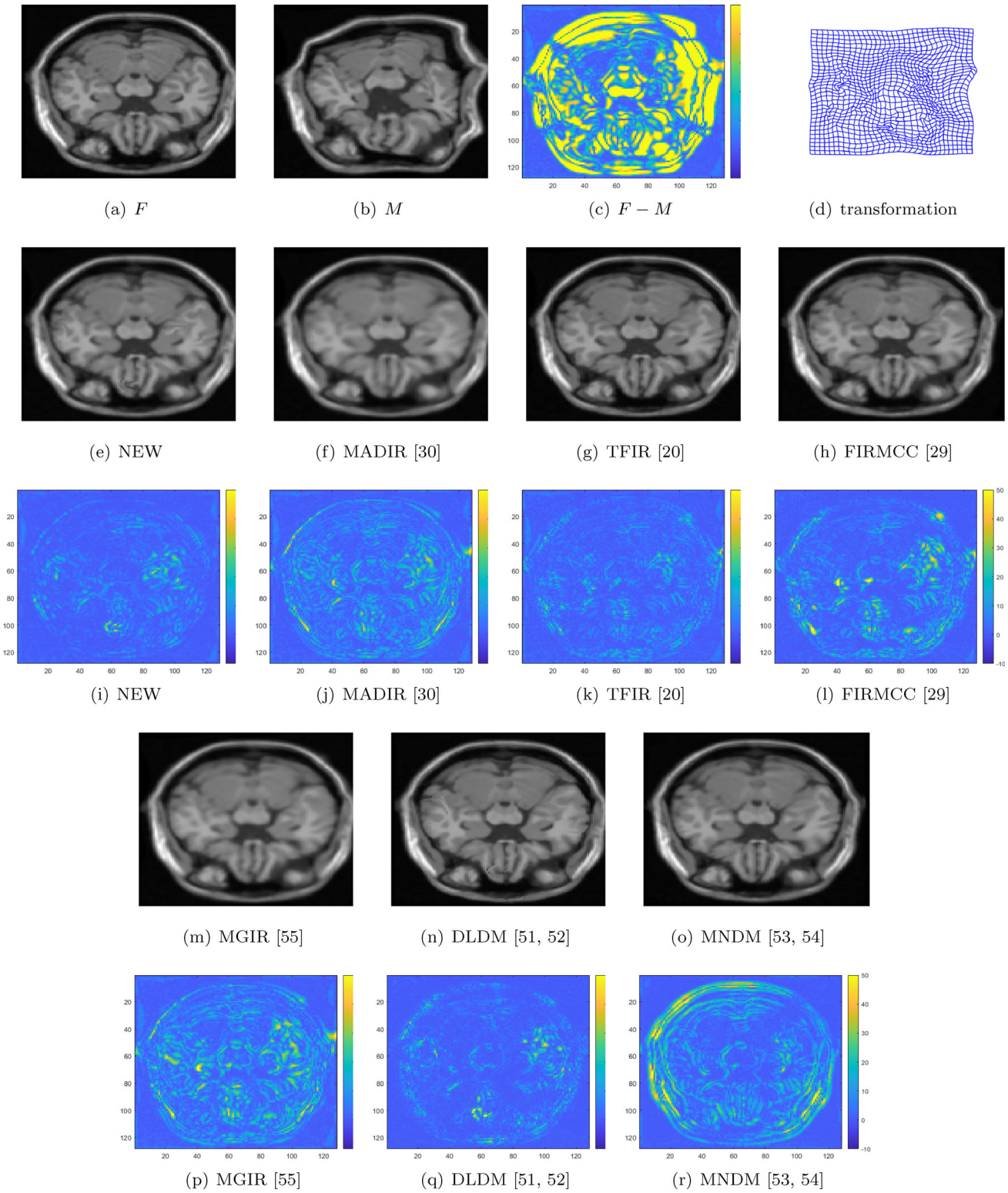
(e) NEW (f) MADIR [30] (g) TFIR [20] (h) FIRMCC [29]

(i) NEW (j) MADIR [30] (k) TFIR [20] (l) FIRMCC [29]

(m) MGIR [55] (n) DLDM [51, 52] (o) MNDM [53, 54]

(p) MGIR [55] (q) DLDM [51, 52] (r) MNDM [53, 54]

Fig. 12 The test image pairs, pre-registration differences, transformations from the new model, template images transformed using our new model and other competitive models, and post-registration differences



(a) F (b) M (c) $F - M$ (d) transformation

(e) NEW (f) MADIR [30] (g) TFIR [20] (h) FIRMCC [29]

(i) NEW (j) MADIR [30] (k) TFIR [20] (l) FIRMCC [29]

(m) MGIR [55] (n) DLDM [51, 52] (o) MNDM [53, 54]

(p) MGIR [55] (q) DLDM [51, 52] (r) MNDM [53, 54]

Fig. 13 The test image pairs, pre-registration differences, transformations from the new model, template images transformed using our new model and other competitive models, and post-registration differences

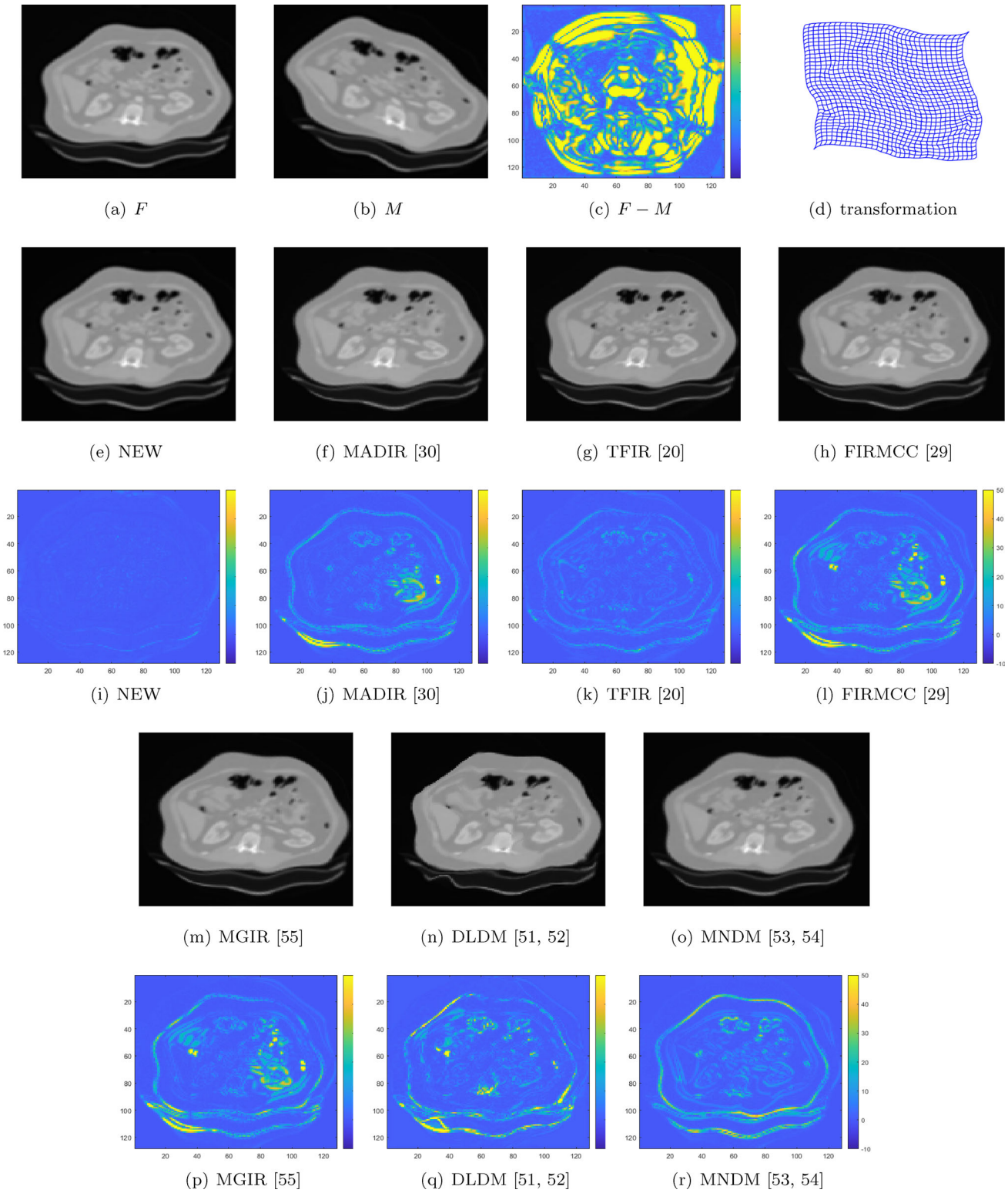


Fig. 14 The test image pairs, pre-registration differences, transformations from the new model, template images transformed using our new model and other competitive models, and post-registration differences

Table 3 Comparison of quantitative results for Example 3. $\mathcal{T}$ represents the total execution time in seconds, including the time for image generation

Model/Test images	Hand			Chest			Brain			Liver		
	$\varepsilon(\%)$	MFN	$\mathcal{T}$	$\varepsilon(\%)$	MFN	$\mathcal{T}$	$\varepsilon(\%)$	MFN	$\mathcal{T}$	$\varepsilon(\%)$	MFN	$\mathcal{T}$
NEW	1.8645	0	74.2	1.0685	0	70.8	1.1712	0	75.6	0.0305	0	58.1
MADIR [30]	3.47	0	241.1	1.44	0	172.1	2.87	0	225.1	2.44	0	311.3
TFIR [20]	4.2186	0	6.3	1.6606	0	9.4	1.3606	0	7.8	0.4186	0	8.2
FIRMCC [29]	5.42	0	75.3	2.01	0	180.3	3.45	0	113.1	3.70	0	264.8
MGIR [55]	7.9	0	24.8	3.2	0	18.2	5.39	0	25.1	4.1	0	61.1
DLDM [51, 52]	4.3156	0	12.4	9.8078	0	11.1	1.3360	0	23.4	3.2238	0	10.6
MNDM [53, 54]	8.1282	0	40.4	6.1336	0	40.2	6.6405	0	48.2	3.6457	0	42.1

References

1. Brown, L.G.: A survey of image registration techniques. ACM Comput. Surv. **24**(4), 325–376 (1992)
2. Goshtasby, A.A.: 2-D and 3-D Image Registration: for Medical, Remote Sensing, and Industrial Applications, New Jersey. Wiley Interscience, USA (2006)
3. Sotiras, A., Davatzikos, C., Paragios, N.: Deformable medical image registration: a survey. IEEE Trans. Med. Imaging **32**(7), 1153–1190 (2013)
4. Modersitzki, J.: Numerical Methods for Image Registration. Oxford University Press, New York (2004)
5. Modersitzki, J.: FAIR: Flexible Algorithms for Image Registration. SIAM, Philadelphia (2009)
6. Fischer, B., Modersitzki, J.: Fast diffusion registration. Contemp. Math. **313**, 117–129 (2002)
7. Fischer, B., Modersitzki, J.: Curvature based Image Registration. J. Math. Imaging Vision **18**, 81–85 (2003)
8. Chumchob, N., Chen, K., Brito, C.: A fourth order variational image registration model and its fast multigrid algorithm. SIAM J. Multiscale Model. Simul **9**(1), 89–128 (2011)
9. Mazlinda, I., Chen, K., Brito, C.: A novel variational model for image registration using Gaussian curvature. J. Geom. Imaging Comput. **1**(4), 417–446 (2014)
10. Burger, M., Modersitzki, J., Ruthotto, L.: A hyperelastic regularization energy for image registration. SIAM J. Scient. Comput. **35**(1), B132–B148 (2013)
11. Chaves, A.S.: A fractional diffusion equation to describe Lévy flights. Phys. Lett. A **239**(1), 13–16 (1998)
12. Metzler, R., Klafter, J.: The random walk's guide to anomalous diffusion: a fractional dynamics approach. Phys. Rep. **339**(1), 1–77 (2000)
13. Podlubny, I.: Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications. Elsevier Science, Math. Sci. Eng. (1999)
14. Chen, D., Chen, Y., Xue, D.: Three fractional-order $TV-l^2$ models for image denoising. J. Comput. Inform. Syst. **9**, 4773–4780 (2013)
15. Zhang, J., Wei, Z., Xiao, L.: Adaptive fractional-order multi-scale method for image denoising. J. Math. Imaging Vision **43**, 39–49 (2012)
16. Zhang, J.P., Chen, K.: A Total fractional-order variation model for image restoration with nonhomogeneous boundary conditions and its numerical solution. SIAM J. Imaging Sci. **8**(4), 2487–2518 (2015)
17. Wang, W., Xia, X.G., Zhang, S.L.: Vector total fractional-order variation and its applications for color image denoising and decomposition. Appl. Math. Model. **72**, 155–175 (2019)
18. Pang, Z.F., Zhang, H.L., Luo, S.S., Zeng, T.Y.: Image denoising based on the adaptive weighted TV^p regularization. Signal Process. **167**, 107325 (2020)
19. Pang, Z.F., Meng, G., Chen, H.L.K.: Image restoration via the adaptive TV^p regularization. Comput. Math. Appl. **80**, 569–587 (2020)
20. Zhang, J.P., Chen, K.: Variational image registration by a total fractional-order variation model. J. Comput. Phys. **293**, 442–461 (2015)
21. Han, H., Zhou, H.: A variational problem arising in registration of diffusion tensor image. Acta Math. Sci. Ser. B **37**(2), 539–554 (2017)
22. Han, H.: A variational model with fractional-order regularization term arising in registration of diffusion tensor image. Inverse Probl. Imaging **12**(6), 1263–1291 (2018)
23. Han, H., Wang, Z.P.: An alternating direction implicit scheme of a fractional-order diffusion tensor image registration model. Appl. Math. Comput. **356**, 105–118 (2019)
24. Haber, E., Modersitzki, J.: Image registration with guaranteed displacement regularity. Int. J. Comput. Vision **71**(3), 361–372 (2007)
25. Chun, K.C., Lui, L.M.: Landmark- and intensity-based registration with large deformations via quasi-conformal maps. SIAM J. Imaging Sci. **7**, 2364–2392 (2014)
26. Zhang, J., Chen, K., Yu, B.: A novel high-order functional based image registration model with inequality constraint. Comput. Math. Appl. **72**, 2887–2899 (2016)
27. Lam, K.C., Lui, L.M.: Optimized quasiconformal parameterization with user-defined area distortions. Commun. Math. Sci. **15**, 2027–2054 (2017)

28. Zhang, D.P., Chen, K.: A novel diffeomorphic model for image registration and its algorithm. J. Math. Imaging Vis. **60**, 1261–1283 (2018)
29. Han, H., Wang, Z.P.: A diffeomorphic image registration model with fractional-order regularization and Cauchy-Riemann constraint. SIAM J. Imaging Sci. **13**(3), 1240–1271 (2020)
30. Han, H., Wang, Z.P.: Multiscale approach for two-dimensional diffeomorphic image registration. SIAM J Multiscale Model. Simul. **19**(4), 1538–1572 (2021)
31. Zhang, J., Sun, Z.G., Kong, X., Zhang, J.P.: A vectorial minimized surface regularizer based image registration model and its numerical algorithm. Appl. Math. Model. **106**, 150–176 (2022)
32. Zhang, J.P., Li, Y.Y.: Diffeomorphic image registration with an optimal control relaxation and its implementation. SIAM J. Imaging Sci. **14**(4), 1890–1931 (2022)
33. Han, H., Wang, Z.P., Zhang, Y.M.: Multiscale approach for three-dimensional conformal image registration. SIAM J. Imaging Sci. **15**(3), 1431–1468 (2022)
34. Han, H., Jin, Q., Zhang, Y.M.: A 2*D* diffeomorphic image registration model with inequality constraint. Comput. Math. Appl. **132**, 135–144 (2023)
35. Ding, Z.J., Han, H., Wang, H.N.: Two-dimensional diffeomorphic model for multi-modality image registration. Comput. Appl. Math. **42**, 17 (2023)
36. Sun, Z.Z., Gao, G.H.: Finite Difference Methods for the Fractional Differential Equations. Science Press, Beijing (2015)
37. Aubert, G., Kornprobst, P.: Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations, 2nd ed. Springer (2006)
38. Blomgren, P., Chan, T.F.: Color TV: total variation methods for restoration of vector-valued images. IEEE Trans. Image Process. **7**, 304–309 (1998)
39. Bresson, X., Chan, T.F.: Fast dual minimization of the vectorial total variation norm and applications to color image proccessing. Inverse Probl. Imaging **2**, 455–484 (2008)
40. Brito-Loeza, C., Chen, K.: On high-order denoising models and fast algorithms for vector-valued images. IEEE Trans. Image Process. **19**, 1518–1527 (2010)
41. Brezis, H.: Functional Analysis. Springer, Sobolev Spaces and Partial Differential Equations (2010)
42. Conway, J.B.: A Course in Functional Analysis. Springer, New York (2007)
43. Demengel, F., Demengel, G., Erne, R.: Functional Spaces for the Theory of Elliptic Partial Differential Equations. Springer, London (2012)
44. Evance, L.: Partial Differential Equations. AMS, Providence, RI (2010)
45. Wang, H., Du, N.: Fast solution methods for space-fractional diffusion equations. J. Comput. Appl. Math. **255**, 376–383 (2014)
46. Hou, T., Tang, T., Yang, J.: Numerical analysis of fully discretized Crank-Nicolson scheme for fractional-in-space Allen-Cahn equations. J. Sci. Comput. **72**, 1214–1231 (2017)
47. Tian, W., Zhou, H., Deng, W.: A class of second order difference approximation for solving space fractional diffusion equations. Math. Comput. **84**, 1703–1727 (2015)
48. Vogel, C.R., Oman, M.E.: Iterative methods for total variation denoising. SIAM J. Sci. Comput. **17**(1), 227–238 (1996)
49. Chen, K., Tai, X.C.: A nonlinear multigrid method for total variation minimization from image restoration. J. Sci. Comput. **33**(2), 115–138 (2007)
50. Savage, J., Chen, K.: An improved and accelerated nonlinear multigrid method for total variation denoising. Int. J. Comput. Math. **82**(8), 1001–1015 (2005)
51. Vercauteren, T., Pennecb, X., Perchanta, A., Ayacheb, N.: Diffeomorphic demons: efficient non-parametric image registration. Neuroimage **45**, 61–72 (2009)
52. Lombaert, H.: Diffeomorphic Log Demons Image Registration, https://www.mathworks.com/ matlabcentral/fileexchange/39194-diffeomorphic-log-demons-image-registration (2012)
53. Kroon, D., Slump, C.: MRI modality transformation in demon registration. In Proceedings of the 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro (Boston, MA), IEEE, Washington, DC, pp. 963–966 (2009)
54. Kroon, D.: Multimodality nonrigid demon algorithm image registration. https://www.mathworks.com/matlabcentral/fileexchange/21451-multimodality-non-rigid-demon-algorithm-image-registration (2008)
55. Han, H., Wang, A.: A fast multi grid algorithm for 2D diffeomorphic image registration model. J. Comput. Appl. Math. **394**, 113576 (2021). https://doi.org/10.1016/j.cam.2021.113576