

## NUMERICAL METHOD FOR IMAGE REGISTRATION MODEL BASED ON OPTIMAL MASS TRANSPORT

YANGANG CHEN

Department of Applied Mathematics  
University of Waterloo, 200 University Avenue West  
Waterloo, ON, N2L 3G1, Canada

JUSTIN W. L. WAN

David R. Cheriton School of Computer Science  
University of Waterloo, 200 University Avenue West  
Waterloo, ON, N2L 3G1, Canada

(Communicated by Xuecheng Tai)

**ABSTRACT.** This paper proposes a numerical method for solving a non-rigid image registration model based on optimal mass transport. The main contribution of this paper is to address two issues. One is that we impose a proper periodic boundary condition, such that when the reference and template images are related by translation, or a combination of translation and non-rigid deformation, the numerical solution gives the underlying transformation. The other is that we design a numerical scheme that converges to the optimal transformation between the two images. As an additional benefit, our approach can decompose the transformation into translation and non-rigid deformation. Our numerical results show that the numerical solutions yield good-quality transformations for non-rigid image registration problems.

**1. Introduction.** In many applications, one has to compare two images  $T$  (template) and  $R$  (reference) which display the same object, but the object inside the images is not spatially aligned, or the devices that record the two images are different. The image registration problem is to find a coordinate transformation  $\phi$  which transforms the image  $T$  to another image  $T_\phi$ , such that  $T_\phi$  is similar and thus comparable to the image  $R$ .

One important application of image registration is to compare medical images of the same patient, such as CT (computed tomography) and MRI (magnetic resonance imaging) images of a damaged brain, which gives guidance for diagnosis and surgery [35, 29]. Image registration can also be used for image fusion [30]. Multiple images of the same object are taken, registered and then merged together, such that the integrated image is clearer than the original ones. We refer readers to [26] for more discussion on applications.

---

2010 *Mathematics Subject Classification.* Primary: 65N06, 65N22; Secondary: 35J96.

*Key words and phrases.* Image registration, mass transport, Monge-Ampère equation, periodic boundary condition, viscosity solution, Hamilton-Jacobi-Bellman (HJB) equation, monotonicity, finite difference.

The second author is supported by Natural Sciences and Engineering Research Council of Canada (NSERC).

Different approaches have been developed for image registration problems, including parametrized transformation [7, 51], landmark-based registration [43], elastic registration [6], fluid registration [14], diffusion registration [21], demon's registration [47], flow of diffeomorphism [18, 50], etc. A substantial discussion of existing methods can be found in [37, 45].

This paper considers a non-rigid image registration method based on Monge-Kantorovich mass transport [27, 28, 23, 44, 12, 39]. The approach was first proposed in [27, 28]. This image registration model treats two images  $R$  and  $T$  as two mass densities. The goal is to find a mapping which transforms one mass density  $T$  to the other  $R$  with mass conservation. Such transformation is non-unique. By defining a transformation-dependent cost function and minimizing it, we can obtain a unique optimal transformation. This optimal transformation has desirable properties. For instance, it is usually diffeomorphic and does not introduce foldings and crossings.

The primary advantage of this image registration model is that, unlike many other non-rigid methods that are only applicable for images with small deformations, this model can be applied to images with large deformations. See Figures 1-2 in [27] for an example of images with large deformations. Indeed, given *any*  $R$  and  $T$ , the transformed image  $T_\phi$  under the mass transport formulation can be equal to  $R$  [39].

Numerical methods have been developed for solving the image registration model based on optimal mass transport. In [27] and [28], the authors construct an initial mass-preserving mapping  $\phi_0$  by solving a nonlinear partial differential equation (PDE), and obtain a second mass-preserving mapping  $\phi_s$  by solving another nonlinear PDE system, such that  $\phi_0 \circ \phi_s$  is the optimal transformation. The entire process involves many intermediate steps. Also, in general, a nonlinear PDE (or PDE system) has multiple solutions. An immediate challenge is that the nonlinear PDE system in [27] and [28] can give multiple transformations between  $R$  and  $T$ , which may not be the optimal transformation<sup>1</sup>. To the best of our knowledge, it is unclear whether the numerical scheme in [27] and [28] gives the optimal transformation.

An alternative approach for solving this image registration model is to solve an equivalent Monge-Ampère equation, which is also a nonlinear PDE and thus has multiple solutions. Among the multiple solutions, there exists a unique globally convex solution. *The gradient of this solution* corresponds to the optimal transformation between  $R$  and  $T$  [27, 31]. The convex solution itself is usually called a scalar potential that generates the optimal transformation.

Some literature has investigated numerical schemes for the Monge-Ampère equation arising from the image registration model [23, 44, 12]. In [23], the author proposes a finite difference scheme for the Monge-Ampère equation, and proves that the resulting transformation between  $R$  and  $T$  is optimal. However, in order to guarantee the optimality, the computational cost *per pixel* must increase to infinity as the image size increases [19]. Hence the method is not practical for large images. Other numerical schemes for the image registration model via the Monge-Ampère equation can be found in [44] and [12]. We remark that although the methodology in [12] is gradient descent, it ends up being equivalent to solving the Monge-Ampère equation using explicit pseudo-timestepping. Whether the transformations given in [44] and [12] are optimal remains an open question.

---

<sup>1</sup>Although the optimality of “optimal transformation” may have various meanings, in this paper, unless specified, optimality refers to the minimization of the cost function in the mass transport.

Another issue in the existing literature is boundary conditions. We note that the transformation on the boundary of  $R$  and  $T$  cannot be determined from the mass transport model and thus needs to be specified by model users. Since the transformation on the boundary influences the transformation inside the images, it is crucial to specify a proper boundary condition, such that the quality of the resulting transformation is good. Here the transformation of good quality means that the resulting transformation should reflect the physical movement of the object inside  $R$  and  $T$ . For instance, if the object inside  $R$  and  $T$  is related by translation, then the resulting transformation should be able to recover the underlying translation.

A common boundary condition considered in the literature, such as in [23], is Neumann boundary condition, where the pixels on the boundary of one image stay on the boundary of the other under the transformation. Alternatively, assuming that  $R$  and  $T$  are periodic images, the authors in [44] use a periodic boundary condition on the solution of the Monge-Ampère equation (namely, on the scalar potential). However, under either of these boundary conditions, when  $R$  and  $T$  are related by translation, or by a combination of translation and non-rigid deformation, the resulting transformation may not be the underlying transformation. To the best of our knowledge, no existing literature using the mass transport model has registered translated images.

In this paper, we develop a numerical scheme for the image registration model based on optimal mass transport by solving the equivalent Monge-Ampère equation. We resolve the two major issues in the existing literature - imposing a proper boundary condition, and ensuring that the resulting transformation by a numerical method is the optimal transformation.

More specifically, we propose to use another periodic boundary condition, where we assume that *the transformation (or the displacement of the pixels)* on the boundary of the images is periodic. Thus, for the Monge-Ampère equation, it means that *the gradient of the solution* is periodic. This is different from [44] where periodicity is imposed on the solution itself. We will show that, in contrast to the commonly used Neumann boundary condition, our periodic boundary condition will recover the underlying transformation for the images that are related by translation or by a combination of translation and non-rigid deformation.

In order to ensure that our numerical scheme yields the optimal transformation between  $R$  and  $T$ , we adopt the theory of viscosity solution [16, 15]. Viscosity solution of the Monge-Ampère equation is the solution that corresponds to the optimal transformation between  $R$  and  $T$  [23, 24]. In this paper, we design a finite difference scheme based on our previous work [13] and the theoretical framework in [1]. Notably, we prove that our numerical solution converges to the viscosity solution of the Monge-Ampère equation. Consequentially, the resulting transformation computed by our numerical scheme is the optimal one.

In addition to these two contributions, our numerical scheme can automatically decompose the transformation between  $R$  and  $T$  into translation and non-rigid deformation components. This extra information is useful in visualizing and understanding the underlying transformation.

This paper is organized as follows. In Section 2, we describe the image registration model based on optimal mass transport. In particular, we have a detailed discussion on boundary conditions. Section 3 describes a finite difference discretization for the Monge-Ampère equation arising from the mass transport image registration model. In Section 4, we propose a modified Levenberg-Marquardt algorithm

to solve the discretized system. In Section 5, we prove that our numerical scheme converges to the optimal transformation between  $R$  and  $T$ . Experimental results in Section 6 show that our numerical scheme gives the optimal transformations, and the results under our periodic boundary condition outperform those under the commonly used Neumann boundary condition. Section 7 concludes the paper.

## 2. Image registration model based on optimal mass transport.

2.1. **Image registration.** In this paper, we use the following notation:

$T$	Template image.
$R$	Reference image.
$\rho^T$	Intensity of template image $T$ on domain $\Omega^T \subset \mathbb{R}^2$ .
$\rho^R$	Intensity of reference image $R$ on domain $\Omega^R \subset \mathbb{R}^2$ .
$\phi$	Coordinate transformation $\phi : \Omega^R \rightarrow \Omega^T$ . $\phi \in \mathbb{R}^{2 \times 1}$ .
$\phi^*$	Optimal coordinate transformation.
$T_\phi$	Transformed image under the transformation $\phi$ .
$\rho^{T_\phi}$	Intensity of transformed image $T_\phi$ on domain $\Omega^R$ .
$u$	Convex scalar potential. $u \in C(\Omega^R)$ .

The objective of image registration problem is to find a coordinate transformation  $\phi$  that minimizes the difference between  $\rho^{T_\phi}$  and  $\rho^R$ , which is usually measured by some function such as sum of squared differences

$$(1) \quad \mathcal{D}(\rho^{T_\phi}, \rho^R) \equiv \|\rho^{T_\phi} - \rho^R\|_{L_2(\Omega^R)}.$$

For simplicity, we restrict our discussion on two square images of the same size, or more specifically,  $\Omega^T = \Omega^R = [0, 1] \times [0, 1]$ . In this paper, unless specified, we will drop the superscripts in  $\Omega^T$  and  $\Omega^R$  and denote them as  $\Omega$ .

2.2. **Mass transport formulation.** The mass transport problem considers transforming one pile of soils  $T$  to another pile  $R$  with mass preservation. Mathematically, let  $\rho^T$  and  $\rho^R$  be two bounded positive density functions on  $\Omega^T$  and  $\Omega^R$  respectively. Suppose  $\rho^T$  and  $\rho^R$  have the same total mass

$$(2) \quad \int_{\Omega^T} \rho^T(\hat{\mathbf{x}}) d^2 \hat{\mathbf{x}} = \int_{\Omega^R} \rho^R(\mathbf{x}) d^2 \mathbf{x}.$$

The goal is to find a coordinate transformation  $\phi : \Omega^R \rightarrow \Omega^T$ , or  $\phi(\mathbf{x}) = \hat{\mathbf{x}}$ , such that  $\rho^T$  is transformed to  $\rho^R$  while the total mass is preserved:

$$\int_{\mathbf{x} \in \Omega^R} \rho^T(\phi(\mathbf{x})) d^2 \phi(\mathbf{x}) = \int_{\Omega^R} \rho^R(\mathbf{x}) d^2 \mathbf{x},$$

or equivalently,

$$(3) \quad \rho^T(\phi(\mathbf{x})) \det[D\phi(\mathbf{x})] = \rho^R(\mathbf{x}),$$

where “det” is the determinant and  $D\phi(\mathbf{x}) \in \mathbb{R}^{2 \times 2}$  is the Jacobian of the transformation  $\phi(\mathbf{x})$ .

Back to image registration, if we view the two images  $T$  and  $R$  as two piles of soils with the densities  $\rho^T$  and  $\rho^R$ , then the image registration problem can be interpreted as a mass transport problem. This idea motivates people to formulate the image registration problem as follows [27, 28, 39]: solve the mass transport problem (3) for the transformation  $\phi(\mathbf{x})$ , such that the template image  $\rho^T(\mathbf{x})$  is transformed to a new image, defined by

$$(4) \quad \rho^{T_\phi}(\mathbf{x}) \equiv \rho^T(\phi(\mathbf{x})) \det[D\phi(\mathbf{x})],$$

which equals to the reference image  $\rho^R(\mathbf{x})$ .

There are two remarks regarding the mass transport registration model. One is that  $\rho^{T_\phi}$  and  $\rho^R$  are always equal for any  $T$  and  $R$ . In other words, the error measure, such as sum of squared differences (1), is zero. As a result, this method can transform any template image  $T$  to any reference image  $R$  [39].

The other remark is that, according to (4), the image registration based on mass transport consists of two components. One component is the movement of pixels from  $\mathbf{x}$  to  $\phi(\mathbf{x})$ , which transforms the image to  $\rho^T(\phi(\mathbf{x}))$ . The other component, called morphing effect, changes the intensity at each moved pixel  $\phi(\mathbf{x})$  by a factor  $\det[D\phi(\mathbf{x})]$ , or more specifically, changes the intensity from  $\rho^T(\phi(\mathbf{x}))$  to  $\det[D\phi(\mathbf{x})]\rho^T(\phi(\mathbf{x}))$ . We note that morphing effect is an essential part of the mass transport model for two reasons. One is that if pixels can be treated as masses, then after a movement of masses, the accumulation/dissipation of masses in certain regions will inevitably cause an increase/decrease of the intensity. The other reason is that, although the movement of pixels alone registers  $T$  and  $R$ , it is the morphing effect that further corrects the registration error (e.g. sum of squared differences) to zero and makes  $T_\phi$  equals to  $R$ .

**2.3. Optimal mass transport.** The mass transport registration in Section 2.2 is ill-posed. More specifically, there exist multiple transformations that move the soil  $\rho^T$  to the same  $\rho^{T_\phi}$ . Among all possible transformations, one of them requires “least cost”, which is desirable. Following [27, 28, 2], we aim at finding the optimal transformation  $\phi^*(\mathbf{x})$  that minimizes the following cost function

$$(5) \quad \phi^*(\mathbf{x}) \equiv \arg \min_{\phi(\mathbf{x})} \int_{\mathbb{R}^2} \|\mathbf{x} - \phi(\mathbf{x})\|^2 \rho^R(\mathbf{x}) d^2\mathbf{x}.$$

In other words, the optimal transformation  $\phi^*(\mathbf{x})$  minimizes the total squared displacements weighted by the masses, where the transport cost of heavier soil particles is higher. In essence, (5) regularizes the mass transport registration and makes the transformation between  $\rho^T$  and  $\rho^{T_\phi}$  unique.

**2.4. Monge-Ampère equation with convexity constraint.** It has been proved in [31] that the optimal transformation that minimizes the cost function (5) can be written as

$$(6) \quad \phi^* = \nabla u.$$

Here  $u \in C(\Omega^R)$  is a convex scalar potential field, where its gradient  $\nabla u$  generates the optimal transformation  $\phi^*$ . Substituting (6) into (3), we have

$$(7) \quad \det[D^2u(\mathbf{x})] = \frac{\rho^R(\mathbf{x})}{\rho^T(\nabla u(\mathbf{x}))},$$

$$(8) \quad u \text{ is convex,}$$

where  $D^2u(\mathbf{x}) \in \mathbb{R}^{2 \times 2}$  is the Hessian matrix of  $u(\mathbf{x})$ . Equation (7) is called Monge-Ampère equation.

We note that since the Monge-Ampère equation (7) is nonlinear, the equation itself without the convexity constraint (8) can have multiple solutions [19, 3]. However, the solution of (7) that satisfies the convexity constraint (8) is unique [23], which we will denote as  $u^*$  whenever we need to distinguish it from the other solutions. We emphasize that the convexity of  $u^*$  is equivalent to the optimality of the transformation  $\phi^* = \nabla u^*$  [27, 23].

The convexity of  $u^*$  implies the following desirable properties for the optimal transformation  $\phi^*$ :

(i)  $\phi^*$  does not introduce foldings or invert the order of pixels. For simplicity, we consider one-dimensional images. The solution  $u^*$  being convex means  $\phi_x^* = u_{xx}^* > 0$  on the entire computational domain. Then for any  $x_1 < x_2$ , their corresponding new coordinates satisfy  $\phi^*(x_1) < \phi^*(x_2)$ . This implies no foldings or inversion of pixel order. We note that this property is a necessary condition for  $\phi^*$  to be bijective.

(ii)  $\phi^*$  is diffeomorphic under the assumption that  $\rho^T, \rho^R$  are continuous and  $\phi^*$  is bijective. Since  $u^*$  is convex (or more precisely, strictly convex), the Jacobian of the transformation  $D\phi^* = D^2u^*$  is positive definite and thus non-singular on the entire computational domain. Hence,  $\phi^*$  is invertible. Meanwhile, if  $\rho^T, \rho^R$  are  $\alpha$ -Hölder continuous, then  $u^* \in C^2(\Omega)$  [23, 10], which implies that  $\phi^* = \nabla u^*$  is differentiable. As a result,  $\phi^*$  is diffeomorphic. Diffeomorphism is observed in all of our experiments.

**2.5. Boundary conditions.** Up to this point, the image registration model is still incomplete. The transformation on the boundary cannot be derived from the mass transport formulation, and it needs to be specified by model users. In terms of the Monge-Ampère equation (7), it means that a boundary condition for the PDE is yet to be specified.

A natural attempt is to specify the value of the solution  $u$  on the boundary, called Dirichlet boundary condition. However, in the image registration context,  $u$  is a scalar potential, and it is not clear what value of scalar potential to specify on the boundary.

Alternatively, one can specify the value of the transformation  $\phi^*$  on the boundary. For instance, [23] assumes that

$$\text{for any } \mathbf{x} \in \partial\Omega^R, \text{ the new coordinate } \phi^*(\mathbf{x}) \in \partial\Omega^T,$$

where  $\partial\Omega^R$  and  $\partial\Omega^T$  are the boundaries of  $\Omega^R$  and  $\Omega^T$ , respectively. This means that any pixels that are on the boundary of  $R$  must stay on the boundary of  $T$  under the mapping. Since  $\phi^* = \nabla u$ , and we have assumed that  $\Omega^R = \Omega^T = [0, 1] \times [0, 1]$ , this boundary condition can be rewritten as

$$(9) \quad u_x|_{x=0} = 0, \quad u_x|_{x=1} = 1, \quad u_y|_{y=0} = 0, \quad u_y|_{y=1} = 1.$$

Equation (9) is called Neumann boundary condition.

However, under the Neumann boundary condition (9), the quality of the resulting transformation  $\phi^*$  may not be good. More specifically, assuming that  $R$  is a translation of  $T$ , we expect the resulting  $\phi^*$  to be the underlying translation, but in numerical experiments, the resulting  $\phi^*$  is not a translation; see Figure 1. The reason is that under a translation, the boundary of  $\Omega^R = [0, 1] \times [0, 1]$  cannot be the boundary of  $\Omega^T = [0, 1] \times [0, 1]$ , which violates the assumption of the Neumann boundary condition (9). More generally, when  $R$  and  $T$  are related by a combination of translation and non-rigid deformation, the resulting  $\phi^*$  may not reflect the underlying transformation, which will be shown in Section 6.

It seems that specifying non-zero values in (9) might recover the underlying transformation between  $T$  and  $R$ . However, feeding correct values for (9) requires knowing in advance what the underlying transformation is, and normally we do not know the answer (this is exactly what image registration tries to find), especially when the two images are related by both translation and non-rigid deformation.

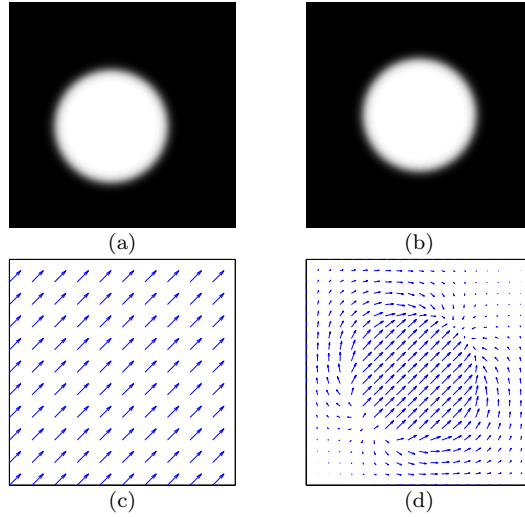


FIGURE 1. An example of image registration using the Neumann boundary condition. (a) Template image  $T$ . (b) Reference image  $R$ . (c) Underlying transformation between  $T$  and  $R$ , which is a pure translation. (d) Transformation given by the Neumann boundary condition.

We remark that [44] considers another boundary condition “ $u(\mathbf{x}) - \frac{1}{2}|\mathbf{x}|^2$  is periodic on  $\partial\Omega$ ”. Similar to (9), this boundary condition does not work for the registration problems that are mentioned in the previous paragraphs.

Our goal is to impose a proper boundary condition that will recover the underlying transformation when  $R$  and  $T$  are related by translation or by a combination of translation and non-rigid deformation. We note that the backgrounds of many images, especially medical images, display only one single color (black or white). Such images can be viewed as periodic density functions, if the domain of the images are extended to  $\mathbb{R}^2$ . Motivated by this fact and a very brief discussion in [27] and [28], we assume that

the displacement of a pixel,  $\phi^*(\mathbf{x}) - \mathbf{x}$ , is periodic on  $\partial\Omega$ .

To take one step further, we substitute  $\phi^* = \nabla u$  and obtain

$$(10) \quad \begin{aligned} (u_x - x)_{x=0} &= (u_x - x)_{x=1}, & (u_x - x)_{y=0} &= (u_x - x)_{y=1}, \\ (u_y - y)_{x=0} &= (u_y - y)_{x=1}, & (u_y - y)_{y=0} &= (u_y - y)_{y=1}. \end{aligned}$$

Equation (10) is a periodic boundary condition with respect to *the gradient* of  $u$ . To the best of our knowledge, this paper is the first application of the gradient-like periodic boundary condition (10) in the context of solving the Monge-Ampère equation (7) for the image registration model.

**3. Finite difference discretization.** Next, we develop a numerical scheme for the Monge-Ampère equation (7) under the convexity constraint (8), which yields the optimal transformation  $\phi^* = \nabla u$ . The convex solution of (7) is indeed a type of solution that has been studied extensively, called viscosity solution [23, 24]. The theory of viscosity solution can be found in [16, 15]. Our goal is to develop a

numerical scheme that converges to the viscosity solution of the Monge-Ampère equation (7).

Different numerical approaches have been developed to solve Monge-Ampère equations  $\det(D^2u) = f$ . Some methods, such as [3, 41, 40, 24, 4, 5, 17, 20], consider the case where  $f = f(x)$  does not depend on the gradient  $\nabla u$ . Other methods, including [8, 9, 23, 44], consider the case where  $f = f(\nabla u)$ . However, it is not obvious whether many of these methods can converge to the viscosity solution.

Very little work has been done for developing numerical schemes that are convergent to the viscosity solution of the Monge-Ampère equation. A finite difference method that can converge to the viscosity solution is proposed in [40, 24, 23]. However, the computational cost per grid point must increase to infinity as the grid size increases in order to achieve convergence[19].

In our previous work [13], we propose a numerical scheme that converges to the viscosity solution of the Monge-Ampère equation  $\det(D^2u) = f$ , where  $f = f(x)$  does not depend on  $\nabla u$  and the boundary condition is Dirichlet. In this paper, we plan to extend our numerical method in [13] to the Monge-Ampère equation (7), where  $f = f(\nabla u)$  and the boundary condition is periodic on  $\nabla u$ . The basic idea is to

- (i) convert the Monge-Ampère equation (7) with the convexity constraint (8) into an equivalent Hamilton-Jacobi-Bellman (HJB) equation;
- (ii) perform a mixed standard 7-point stencil and wide stencil finite difference discretization on the equivalent HJB equation; and
- (iii) solve the discretized system.

Steps (i) and (ii) follow our previous work [13]. However, Step (iii) requires a significant change, which will be discussed separately in Section 4.

**3.1. HJB formulation of the Monge-Ampère equation.** Step (i) is to convert the Monge-Ampère equation (7) with the constraint (8) into an equivalent HJB equation[32, 34]:

**Theorem 3.1.** *Let  $u \in C^2(\Omega^R)$  be convex, and let  $\rho^T \in C(\Omega^T)$  and  $\rho^R \in C(\Omega^R)$  be two positive functions. Then the Monge-Ampère equation (7) with the convexity constraint (8) is equivalent to the following HJB equation*

$$(11) \quad \max_{(a(\mathbf{x}), \theta(\mathbf{x})) \in \Gamma} \left\{ \mathcal{L}_{a(\mathbf{x}), \theta(\mathbf{x})} u(\mathbf{x}) + 2 \sqrt{a(\mathbf{x})(1-a(\mathbf{x})) \frac{\rho^R(\mathbf{x})}{\rho^T(\nabla u(\mathbf{x}))}} \right\} = 0.$$

Here  $(a(\mathbf{x}), \theta(\mathbf{x}))$  is a pair of control variables at point  $\mathbf{x} \in \Omega^R$ ,  $\Gamma \equiv [0, 1] \times [-\frac{\pi}{4}, \frac{\pi}{4}]$  is a set of admissible controls, and

$$(12) \quad \mathcal{L}_{a, \theta} u \equiv -\alpha_{11}(a, \theta) u_{xx} - 2\alpha_{12}(a, \theta) u_{xy} - \alpha_{22}(a, \theta) u_{yy}$$

is a second order differential operator with the coefficients

$$(13) \quad \begin{aligned} \alpha_{11}(a, \theta) &= \frac{1}{2}[1 - (1 - 2a) \cos 2\theta], \\ \alpha_{22}(a, \theta) &= \frac{1}{2}[1 + (1 - 2a) \cos 2\theta], \\ \alpha_{12}(a, \theta) &= \frac{1}{2}(1 - 2a) \sin 2\theta. \end{aligned}$$

*Proof.* We refer interested readers to [13, 34]. □

The HJB equation (11)-(12) turns out to be more manageable for numerical computation than the Monge-Ampère equation (7). The reason is that if we assume that the control pair  $(a, \theta)$  is given, then the differential operator (12) becomes



linear, which is less difficult than the nonlinear operator  $\det(D^2u)$  in (7) in terms of finite difference discretization.

**3.2. Standard 7-point stencil and wide stencil discretizations.** Next, we discretize the HJB equation (11)-(12) using finite difference. Interested readers can find details in our previous work [13]. To sketch the idea, we denote the  $N \times N$  grid points by  $\{\mathbf{x}_{i,j} \equiv (ih, jh), 1 \leq i, j \leq N\}$ , where  $h = 1/N$  is the mesh size. We also denote the finite difference approximations of  $u(\mathbf{x}_{i,j})$ ,  $a(\mathbf{x}_{i,j})$ ,  $\theta(\mathbf{x}_{i,j})$ ,  $\alpha_{11}(a(\mathbf{x}_{i,j}), \theta(\mathbf{x}_{i,j}))$ ,  $\alpha_{12}(a(\mathbf{x}_{i,j}), \theta(\mathbf{x}_{i,j}))$  and  $\alpha_{22}(a(\mathbf{x}_{i,j}), \theta(\mathbf{x}_{i,j}))$  as  $U_{i,j}$ ,  $a_{i,j}$ ,  $\theta_{i,j}$ ,  $(\alpha_{11})_{i,j}$ ,  $(\alpha_{12})_{i,j}$  and  $(\alpha_{22})_{i,j}$ , respectively. In addition, we introduce the forward/backward difference operators:

$$\begin{aligned} D_x^+ U_{i,j} &\equiv \frac{U_{i+1,j} - U_{i,j}}{h}, & D_x^- U_{i,j} &\equiv \frac{U_{i,j} - U_{i-1,j}}{h}, \\ D_y^+ U_{i,j} &\equiv \frac{U_{i,j+1} - U_{i,j}}{h}, & D_y^- U_{i,j} &\equiv \frac{U_{i,j} - U_{i,j-1}}{h}. \end{aligned}$$

According to the theory in [1], in order to converge to the viscosity solution, we intend to design a finite difference scheme that is monotone [1, 48]. We approximate  $u_{xx}(\mathbf{x}_{i,j})$  and  $u_{yy}(\mathbf{x}_{i,j})$  using standard central differencing  $D_x^+ D_x^- U_{i,j}$  and  $D_y^+ D_y^- U_{i,j}$ . Regarding the monotone discretization of the cross derivative  $u_{xy}(\mathbf{x}_{i,j})$ , one can verify the following:

(i) If the following condition is satisfied:

$$(14) \quad (\alpha_{11})_{i,j} \geq |(\alpha_{12})_{i,j}|, \quad (\alpha_{22})_{i,j} \geq |(\alpha_{12})_{i,j}|, \quad (\alpha_{12})_{i,j} \geq 0,$$

then  $u_{xy}(\mathbf{x}_{i,j})$  can be discretized monotonically by the standard 7-point stencil scheme  $\frac{1}{2}(D_x^+ D_y^+ + D_x^- D_y^-)U_{i,j}$ . Then the discretization of  $\mathcal{L}_{a(\mathbf{x}_{i,j}), \theta(\mathbf{x}_{i,j})} u(\mathbf{x}_{i,j})$ , namely the differential operator (12) at  $\mathbf{x}_{i,j}$ , reads

$$(15) \quad -(\alpha_{11})_{i,j} D_x^+ D_x^- U_{i,j} - (\alpha_{22})_{i,j} D_y^+ D_y^- U_{i,j} - (\alpha_{12})_{i,j} (D_x^+ D_y^+ + D_x^- D_y^-) U_{i,j}.$$

The stencil points of (15) are shown in Figure 2(a).

(ii) If the following condition is satisfied:

$$(16) \quad (\alpha_{11})_{i,j} \geq |(\alpha_{12})_{i,j}|, \quad (\alpha_{22})_{i,j} \geq |(\alpha_{12})_{i,j}|, \quad (\alpha_{12})_{i,j} \leq 0,$$

then  $u_{xy}(\mathbf{x}_{i,j})$  can be discretized monotonically by another standard 7-point stencil scheme  $\frac{1}{2}(D_x^+ D_y^- + D_x^- D_y^+)U_{i,j}$ . The discretization of  $\mathcal{L}_{a(\mathbf{x}_{i,j}), \theta(\mathbf{x}_{i,j})} u(\mathbf{x}_{i,j})$  becomes

$$(17) \quad -(\alpha_{11})_{i,j} D_x^+ D_x^- U_{i,j} - (\alpha_{22})_{i,j} D_y^+ D_y^- U_{i,j} - (\alpha_{12})_{i,j} (D_x^+ D_y^- + D_x^- D_y^+) U_{i,j}.$$

The stencil points of (17) are shown in Figure 2(b).

In general, however, neither Condition (14) nor (16) is satisfied. In order to obtain a monotone scheme, we consider performing a local coordinate rotation on  $\mathcal{L}_{a(\mathbf{x}_{i,j}), \theta(\mathbf{x}_{i,j})} u(\mathbf{x}_{i,j})$  to eliminate the cross derivative. As shown in Figure 2(c), we apply a clock-wise local coordinate rotation by the angle  $\theta_{i,j}$ , which rotates the local axes  $((\mathbf{e}_x)_{i,j}, (\mathbf{e}_y)_{i,j})$  to new axes  $((\mathbf{e}_z)_{i,j}, (\mathbf{e}_w)_{i,j})$ . Then the differential operator (12) is transformed into

$$\mathcal{L}_{a(\mathbf{x}_{i,j}), \theta(\mathbf{x}_{i,j})} u(\mathbf{x}_{i,j}) = -a_{i,j} u_{zz}(\mathbf{x}_{i,j}) - (1 - a_{i,j}) u_{ww}(\mathbf{x}_{i,j}),$$

where  $u_{zz}$  and  $u_{ww}$  are the directional derivatives along  $((\mathbf{e}_z)_{i,j}, (\mathbf{e}_w)_{i,j})$  axes. Then  $u_{zz}(\mathbf{x}_{i,j})$  can be approximated by central differencing:

$$D_z^+ D_z^- U_{i,j} \equiv \frac{1}{h} \left( \mathcal{I}_h U|_{\mathbf{x}_{i,j} + \sqrt{h}(\mathbf{e}_z)_{i,j}} - 2U_{i,j} + \mathcal{I}_h U|_{\mathbf{x}_{i,j} - \sqrt{h}(\mathbf{e}_z)_{i,j}} \right).$$

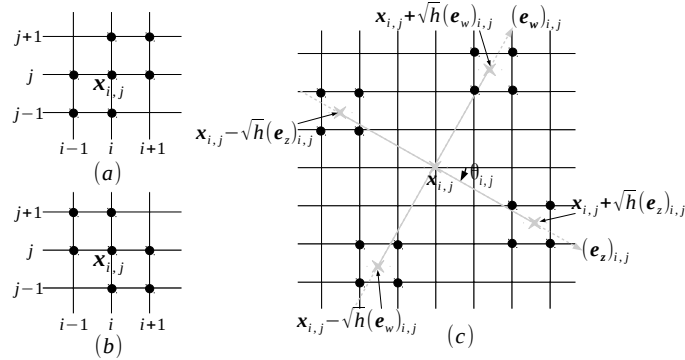


FIGURE 2. Standard 7-point stencil and wide stencil discretizations. (a) The stencil points of the discretization (15). (b) The stencil points of the discretization (17). (c) Wide stencil discretization: We apply a local coordinate rotation at the grid point  $\mathbf{x}_{i,j}$  by the angle  $\theta_{i,j}$ . The grey dashed lines are the orthogonal axes  $\{(\mathbf{e}_z)_{i,j}, (\mathbf{e}_w)_{i,j}\}$ . The stencil length is  $\sqrt{h}$  ( $\sqrt{h} > h$ ). The grey stars are the stencil points  $\mathbf{x}_{i,j} \pm \sqrt{h}(\mathbf{e}_z)_{i,j}$  and  $\mathbf{x}_{i,j} \pm \sqrt{h}(\mathbf{e}_w)_{i,j}$ . The unknowns at these stencil points are approximated by the bilinear interpolation from the neighboring points (black dots).

Here the stencil length is chosen as  $\sqrt{h}$  to maintain consistency, which will be proved in Section 5. The stencil points  $\mathbf{x}_{i,j} \pm \sqrt{h}(\mathbf{e}_z)_{i,j}$  may not coincide with any grid point, so we approximate the unknown values at these stencil points using bilinear interpolation, denoted as  $\mathcal{I}_h U|_{\mathbf{x}_{i,j} \pm \sqrt{h}(\mathbf{e}_z)_{i,j}}$ ; see Figure 2(c) for details. We note that  $u_{ww}(\mathbf{x}_{i,j})$  can be discretized in a similar fashion as  $D_w^+ D_w^- U_{i,j}$ . As a result,  $\mathcal{L}_{a(\mathbf{x}_{i,j}), \theta(\mathbf{x}_{i,j})} u(\mathbf{x}_{i,j})$  can be discretized monotonically by

$$(18) \quad -a_{i,j} D_z^+ D_z^- U_{i,j} - (1 - a_{i,j}) D_w^+ D_w^- U_{i,j}.$$

Since the stencil lengths of  $D_z^+ D_z^- U_{i,j}$  and  $D_w^+ D_w^- U_{i,j}$  are both  $\sqrt{h}$ , which is greater than  $h$ , we call (18) wide stencil discretization.

The advantage of the wide stencil discretization (18) is that it is unconditionally monotone. One may apply the wide stencil discretization at every grid point. However, we will prove that (18) is only first order accurate, while the standard 7-point stencil discretizations (15) and (17) are second order accurate. Considering this, we will only apply the wide stencil discretization (18) at the grid points where neither (14) nor (16) is satisfied. For the other grid points where (14) and (16) are fulfilled, we will apply the standard 7-point stencil discretizations (15) and (17), such that monotonicity still holds and the numerical scheme is as accurate as possible.

**3.3. The complete discretized system.** A complete finite difference discretization of (11)-(12) based on (15), (17) and (18) gives rise to a discrete system. Define a vector of the unknowns  $U \equiv \{U_{i,j} \mid 1 \leq i, j \leq N\} \in \mathbb{R}^{N^2 \times 1}$ , and correspondingly, vectors of the controls  $a \in \mathbb{R}^{N^2 \times 1}$ ,  $\theta \in \mathbb{R}^{N^2 \times 1}$ . Similar to [13], we can write the discretized system into the following matrix form:

$$(19) \quad \max_{(a, \theta) \in \Gamma} \{\mathbf{A}[a, \theta]U - F(a, \theta; U)\} = 0.$$

Here  $\mathbf{A}[a, \theta] \in \mathbb{R}^{N^2 \times N^2}$  is the matrix that assembles the coefficients of  $\{U_{i,j}\}$  in (15), (17) and (18), which depends on the controls  $(a, \theta)$ .  $F(a, \theta; U) \in \mathbb{R}^{N^2 \times 1}$  is the vector coming from the discretization of  $2\sqrt{a(\mathbf{x}_{i,j})(1-a(\mathbf{x}_{i,j}))} \frac{\rho^R(\mathbf{x}_{i,j})}{\rho^T(\nabla u(\mathbf{x}_{i,j}))}$  plus the boundary terms in (15), (17) and (18).  $F$  depends on both the controls  $(a, \theta)$  and the unknown  $U$ .

**4. Solving the discretized system.** In this section, we solve the nonlinear discretized system (19) using iterative methods. Suppose the approximation of the solution  $U$  at the  $k$ -th iteration is  $U^{(k)}$ , and the corresponding optimal controls are

$$(20) \quad (a^{(k)}, \theta^{(k)}) = \arg \max_{(a, \theta) \in \Gamma} \left\{ \mathbf{A}[a, \theta]U^{(k)} - F(a, \theta; U^{(k)}) \right\}.$$

This is a nonlinear optimization problem. Interested readers can refer to [13, 22] for how to solve this problem. We define the residual of (19) as

$$(21) \quad \begin{aligned} R(U^{(k)}) &\equiv \max_{(a, \theta) \in \Gamma} \left\{ \mathbf{A}[a, \theta]U^{(k)} - F(a, \theta; U^{(k)}) \right\} \\ &= \mathbf{A}[a^{(k)}, \theta^{(k)}]U^{(k)} - F(a^{(k)}, \theta^{(k)}; U^{(k)}). \end{aligned}$$

We also define the Jacobian matrix of (19) as

$$\mathbf{J}[U^{(k)}] \equiv \frac{dR^{(k)}}{dU^{(k)}} = \frac{\partial R^{(k)}}{\partial U^{(k)}} + \frac{\partial R^{(k)}}{\partial a^{(k)}} \frac{\partial a^{(k)}}{\partial U^{(k)}} + \frac{\partial R^{(k)}}{\partial \theta^{(k)}} \frac{\partial \theta^{(k)}}{\partial U^{(k)}},$$

where  $R^{(k)} \equiv R(U^{(k)})$ . Since  $(a^{(k)}, \theta^{(k)})$  are the optimal controls, which implies that  $\frac{\partial R^{(k)}}{\partial a^{(k)}} = 0$ ,  $\frac{\partial R^{(k)}}{\partial \theta^{(k)}} = 0$ , the Jacobian can be simplified as

$$(22) \quad \mathbf{J}[U^{(k)}] = \frac{\partial R^{(k)}}{\partial U^{(k)}} = \mathbf{A}[a^{(k)}, \theta^{(k)}] - \delta F^{a^{(k)}, \theta^{(k)}}[U^{(k)}],$$

where  $\delta F^{a^{(k)}, \theta^{(k)}}[U^{(k)}]$  represents the Fréchet derivative of  $F$  with respect to the vector  $U^{(k)}$ , given the fixed controls  $(a^{(k)}, \theta^{(k)})$ .

A standard approach of solving (19) is to apply Newton's method, which reads

$$\begin{aligned} E^{(k)} &\equiv -(\mathbf{J}^{(k)})^{-1}R^{(k)}, \\ U^{(k+1)} &= U^{(k)} + E^{(k)}, \end{aligned}$$

where  $\mathbf{J}^{(k)} \equiv \mathbf{J}[U^{(k)}]$ . It turns out that Newton's method fails to converge, since the Jacobian is singular.

To explain the reason behind singular Jacobian, we note that the differential operator  $\mathcal{L}_{a, \theta} u$  under the periodic boundary condition (10) has two linearly-independent zero kernels. One is the function  $v_1(x, y) \equiv x$ , which satisfies  $\mathcal{L}_{a, \theta} v_1 = 0$ . The other is the function  $v_2(x, y) \equiv y$ , which satisfies  $\mathcal{L}_{a, \theta} v_2 = 0$ .  $v_1$  and  $v_2$  correspond to the translations along  $x$  and  $y$  directions respectively<sup>2</sup>. These two translation kernels are desirable, since they allow pure translation to be a solution of the image registration problem where  $R$  and  $T$  are related by translation. Furthermore, since the Jacobian  $\mathbf{J}[U]$  is related to the discretization of  $\mathcal{L}_{a, \theta} u$ , it has discrete translation kernels<sup>3</sup>  $V_1 \in \mathbb{R}^{N^2 \times 1}$  and  $V_2 \in \mathbb{R}^{N^2 \times 1}$ , where

$$(23) \quad (V_1)_{i,j} \equiv x_i, \quad (V_2)_{i,j} \equiv y_j.$$

<sup>2</sup>Or more explicitly,  $\nabla v_1 = (1, 0)^T$  and  $\nabla v_2 = (0, 1)^T$  are the unit translations along  $x$  and  $y$  directions respectively.

<sup>3</sup>To be precise, the zero kernels of the Jacobian are similar to but not equal to  $V_1$  and  $V_2$ , since its zero kernels are perturbed by the Fréchet derivative of  $F$  in (19).

This results in a singular Jacobian.

In the event of a singular Jacobian, one may consider replacing the inverse of the Jacobian by its pseudo-inverse. Then the iteration becomes

$$\begin{aligned} \text{Solve } (\mathbf{J}^{(k)})^T \mathbf{J}^{(k)} E^{(k)} &= -(\mathbf{J}^{(k)})^T R^{(k)} \text{ for } E^{(k)}, \\ U^{(k+1)} &= U^{(k)} + E^{(k)}. \end{aligned}$$

More generally, we may consider introducing a regularizer:

$$(24) \quad \begin{aligned} \text{Solve } [\lambda I + (\mathbf{J}^{(k)})^T \mathbf{J}^{(k)}] E^{(k)} &= -(\mathbf{J}^{(k)})^T R^{(k)} \text{ for } E^{(k)}, \\ U^{(k+1)} &= U^{(k)} + E^{(k)}. \end{aligned}$$

This indeed leads to a known algorithm, called ‘‘Levenberg-Marquardt algorithm’’ [33, 36].

One advantage of using Levenberg-Marquardt algorithm (24) is that convergence of this iterative solver has been proved [36]. The parameter  $\lambda$  can be changed dynamically to make the algorithm not only convergent, but also more efficient. More specifically, when the approximated solution  $U^{(k)}$  is not close to the solution  $U$ ,  $\lambda$  can be increased, such that the algorithm behaves like a gradient descent and is less likely to diverge. Conversely, when  $U^{(k)}$  is close to  $U$ ,  $\lambda$  can be decreased, such that the algorithm behaves like Newton’s method and can converge faster [36]. In particular, Levenberg-Marquardt algorithm is able to converge even when the Jacobian becomes singular.

Despite the fact that Levenberg-Marquardt algorithm (24) converges, it may not converge to the solution of the discretized system (19). To explain this, we note that the algorithm is indeed designed for solving nonlinear least square problems. In the context of solving (19), the algorithm attempts to find a solution that minimizes the residual, which means that it attempts to solve  $U = \arg \min_{\hat{U}} \|R(\hat{U})\|^2$ . Its global minimum, which satisfies  $R(U) = 0$ , is the solution of (19). It is known that Levenberg-Marquardt algorithm may converge to a local minimum rather than the global minimum, depending on the initial guess  $U^{(0)}$ . In terms of image registration, this issue is observed when the two images are related by translation. The algorithm may be stuck in a local minimum solution that does not correspond to translation.

The local minimum issue using Levenberg-Marquardt algorithm (24) can be traced back to the two translation kernels  $V_1$  and  $V_2$ . If we define the ‘‘translation components’’ of a vector  $U$  as the projections of  $U$  to the translation kernels, then an initial guess  $U^{(0)}$  and the solution  $U$  usually have different translation components. Levenberg-Marquardt algorithm turns out to be inefficient in correcting the translation components.

In order to fix this problem, we add an additional step before each Levenberg-Marquardt iteration (24). In this additional step, we explicitly correct the translation components of the approximated solution  $U^{(k)}$ . The amount of correction is added such that the corrected solution, denoted as  $U^{(k+\frac{1}{2})}$ , minimizes the residual  $R(U)$ . This gives rise to our final algorithm for solving the discretized system (19); see Table 1 for the algorithm.

We remark that in Line 5 of the algorithm,  $(c_1^{(k)}, c_2^{(k)})$  is the amount of corrected translation components along the  $x$  and  $y$  directions. In order to find  $(c_1^{(k)}, c_2^{(k)})$ , we can use simple search, which means to discretize  $\Pi$  into a discrete set and then search for the minimum on the discrete set. Also, in practice, if  $(c_1^{(k)}, c_2^{(k)}) = (0, 0)$  or sufficiently small at the  $k$ -th iteration, which means that correction for the

<p>1: Start with an initial guess <math>U^{(0)} = \frac{1}{2}(x^2 + y^2)</math>.</p> <p>2: Set <math>(c_1^{(-1)}, c_2^{(-1)}) = (\infty, \infty)</math>, <math>\Pi \equiv [-\frac{1}{2}, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]</math>.</p> <p>3: <b>for</b> <math>k = 0, 1, \dots</math> until convergence <b>do</b></p> <p>4:   <b>if</b> <math>(c_1^{(k-1)}, c_2^{(k-1)}) \neq (0, 0)</math> <b>then</b></p> <p>5:     <math>(c_1^{(k)}, c_2^{(k)}) = \arg \min_{(c_1, c_2) \in \Pi} \ R(U^{(k)} + c_1 V_1 + c_2 V_2)\ </math>.</p> <p>6:     <math>U^{(k+\frac{1}{2})} = U^{(k)} + c_1^{(k)} V_1 + c_2^{(k)} V_2</math>.</p> <p>7:   <b>end if</b></p> <p>8:   Compute <math>(a^{(k+\frac{1}{2})}, \theta^{(k+\frac{1}{2})})</math> by (20).</p> <p>9:   Compute <math>R^{(k+\frac{1}{2})} \equiv R(U^{(k+\frac{1}{2})})</math> by (21).</p> <p>10:   Compute <math>\mathbf{J}^{(k+\frac{1}{2})} \equiv \mathbf{J}[U^{(k+\frac{1}{2})}]</math> by (22).</p> <p>11:   Solve</p> $\begin{aligned} [\lambda I + (\mathbf{J}^{(k+\frac{1}{2})})^T \mathbf{J}^{(k+\frac{1}{2})}] E^{(k+\frac{1}{2})} \\ = -(\mathbf{J}^{(k+\frac{1}{2})})^T R^{(k+\frac{1}{2})}. \end{aligned}$ <p>      for <math>E^{(k+\frac{1}{2})}</math>.</p> <p>12:   <math>U^{(k+1)} = U^{(k+\frac{1}{2})} + E^{(k+\frac{1}{2})}</math>.</p> <p>13: <b>end for</b></p>
---

TABLE 1. Modified Levenberg-Marquardt algorithm.

translation components of the approximated solution is completed, then Line 4-7 can be skipped for future iterations. It turns out that Line 4-7 only needs to be implemented for very few steps (typically, around 1-3 steps in our experiments).

An additional benefit of this algorithm is that it decomposes the resulting transformation  $\phi^* = \nabla u$  into a pure translation component and a non-rigid deformation component. The pure translation component is given by the accumulation of the corrections of the translation kernels, or more precisely, the accumulation of  $(c_1^{(k)}, c_2^{(k)})$ :

$$(25) \quad \phi^{tran} \equiv \sum_{k \geq 0} (c_1^{(k)}, c_2^{(k)})^T .$$

Subtracting  $\phi^{tran}$  from the resulting transformation yields the remaining non-rigid deformation component.

Finally, to link the theory with the experiments, we summarize the complete implementation of our numerical scheme as follows:

- 1: Normalize  $\rho^R$  and  $\rho^T$  in order to satisfy (2).
- 2: Discretize the Monge-Ampère equation (7) as described in Section 3, yielding the discrete system (19).
- 3: Solve (19) using the modified Levenberg-Marquardt algorithm (Table 1), yielding the convex scalar potential  $u$ .
- 4: Compute the transformation  $\phi^* = (u_x, u_y)^T$ .
- 5: Decompose the transformation into translation component  $\phi_{tran}$  and the non-rigid deformation component by (25).
- 6: Compute the transformed image (4).

**5. Convergence to the optimal transformation.** In this section, we prove that the transformation between  $R$  and  $T$ , given by our numerical scheme, is the optimal transformation that minimizes the cost function (5).

**Lemma 5.1** (Consistency). *The numerical scheme (19) is locally consistent. In other words, when  $h \rightarrow 0$ , the local truncation error of (19) goes to 0.*

**Lemma 5.2** (Monotonicity). *The numerical scheme (19) is a monotone scheme. More specifically, the left hand side of (19) at a grid point  $\mathbf{x}_{i,j}$  is an increasing function of  $U_{i,j}$  and a non-increasing function of  $U_{p,q}$  for any  $(p,q) \neq (i,j)$ .*

Interested readers are referred to the Appendix for the proofs of Lemma 5.1 and 5.2. The key point of introducing these two lemmas is that it leads to our main theoretical result:

**Theorem 5.3** (Optimal transformation). *Suppose  $u$  is given by the numerical solution of (19) arising from the finite difference scheme in Section 3, and assume that  $u$  is bounded in the max norm<sup>4</sup>. Then the transformation  $\phi^* = \nabla u$  is the optimal transformation that minimizes the cost function (5).*

*Proof.* Since the numerical scheme (19) satisfies consistency (Lemma 5.1), monotonicity (Lemma 5.2) and the boundedness of the solution, by the theorem in [1], the solution is guaranteed to converge to the viscosity solution of the Monge-Ampère equation (7). We refer readers to [1] for a complete proof. Then by [23, 24], the viscosity solution  $u$  satisfies the convexity constraint (8). Hence,  $\phi^* = \nabla u$  is the optimal transformation.  $\square$

**6. Experimental results.** This section includes several experimental results of image registration.

**6.1. Evaluation criteria.** One basic error measure for registered images is sum of squared differences (1). For mass transport registration model, (1) is always zero. The reason is that  $\rho^R$  equals to  $\rho^{T_{\phi^*}}$  by (3) and (4). In other words, the transformed image  $T_{\phi^*}$  and the reference  $R$  are always the same. This can be seen in Examples 2-7 in the following sections, where the mass transport registration under the periodic boundary condition is applied.

Table 2 reports the sum of the squared differences for the images before registration  $\|\rho^T - \rho^R\|_{L_2(\Omega)}$ , and after registration  $\|\rho^{T_{\phi^*}} - \rho^R\|_{L_2(\Omega)}$ . From Table 2, we notice that the error after the registration is significantly smaller than before the registration. Also, the error after the registration is not exactly but close to zero in practice, although it should be zero in theory. The reason is that Levenberg-Marquardt algorithm is implemented to solve the Monge-Ampère equation, which requires choosing a stop tolerance for convergence<sup>5</sup>.

Other than sum of squared differences, another criteria of evaluating the quality of the registration is the quality of the resulting transformation  $\phi^*$ . We expect  $\phi^*$  to reflect the underlying transformation between the two images, especially when  $T$  and  $R$  are related by translation, or by a combination of translation and non-rigid deformation. We will evaluate the resulting transformations qualitatively in Sections 6.2, 6.3 and 6.4, and also discuss the quantitative evaluation separately in Section 6.5.

<sup>4</sup>This is fulfilled in all our numerical experiments.

<sup>5</sup>We choose the tolerance to be  $10^{-4}$ .

Examples	Example 2	Example 3	Example 4
$\ \rho^T - \rho^R\ _{L_2(\Omega)}$	0.47	0.52	0.61
$\ \rho^{T_{\phi^*}} - \rho^R\ _{L_2(\Omega)}$	0	$2 \times 10^{-5}$	$1 \times 10^{-3}$
Examples	Example 5	Example 6	Example 7
$\ \rho^T - \rho^R\ _{L_2(\Omega)}$	0.64	0.69	0.59
$\ \rho^{T_{\phi^*}} - \rho^R\ _{L_2(\Omega)}$	$7 \times 10^{-4}$	$9 \times 10^{-4}$	$8 \times 10^{-3}$

TABLE 2. Sum of the squared differences before registration  $\|\rho^T - \rho^R\|_{L_2(\Omega)}$ , and after registration  $\|\rho^{T_{\phi^*}} - \rho^R\|_{L_2(\Omega)}$ . The values are computed for Examples 2-7 in Sections 6.3, 6.4 and 6.6. For each example,  $\|\rho^R\|$  has been normalized to 1. The approach is the mass transport registration under the periodic boundary condition.

net flow of mass/pixels	area change of a square element	change of mass/pixels intensity	morphing magnitude $\mu$	color of a square element
zero	invariance	invariance	$\mu = 0$	white
inflow	compressed	increase	$\mu > 0$	red
outflow	expanded	decrease	$\mu < 0$	blue

TABLE 3. A summary of morphing effect at a point (or an infinitesimal element).

An additional criteria to consider is morphing effect, which is the change of intensity at each moved pixel. We refer readers to the end of Section 2.2 for a description of morphing effect. We emphasize again that morphing effect is an essential component of the mass transport registration. However, in some image registration applications where the physical object inside the two images is incompressible and cannot be treated as masses, morphing effect may misinterpret the physics of the deformation, and is thus undesirable. Considering this, it is good to suppress morphing effect under the framework of mass transport. We will see that this can be achieved by imposing our periodic boundary condition (10).

To quantify the morphing effect, we define *morphing magnitude*:

$$\mu \equiv \log_{10} \det[D\phi^*(\mathbf{x})],$$

which is the (logarithmic) ratio of the intensity before and after morphing at a moved pixel  $\mathbf{x} \rightarrow \phi^*(\mathbf{x})$ . We visualize the morphing magnitude  $\mu$  using color scale in Figure 4–10 in the following sections. We note that  $\mu = 0$ , or white color, means no morphing effect; the larger  $|\mu|$  is, or the more intense the red/blue color is, the more severe the morphing effect is. Table 3 summarizes some basic features of morphing effect, which can be derived from the theory in Section 2.2.

### 6.2. Optimal versus non-optimal transformations.

**Example 1** (Figure 3). In this simple example, we assume that both the template image  $T$  and the reference image  $R$  are the same constant image, or,  $\rho^T = \rho^R = 1$  on the entire  $\Omega = [0, 1] \times [0, 1]$ . The transformation given by our monotone numerical scheme is the optimal transformation, which is the identity mapping  $\phi^*(\mathbf{x}) = \mathbf{x}$  and maps a square grid to itself. However, a non-monotone finite difference scheme in [3] gives a non-optimal transformation, which severely deforms a square grid.

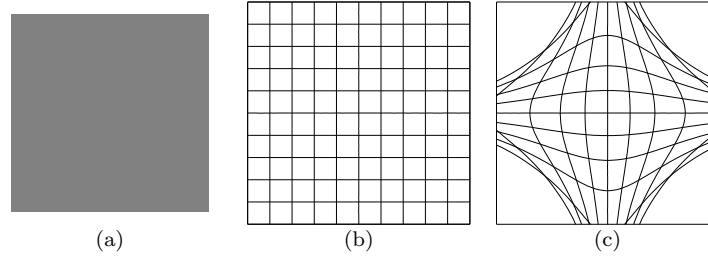


FIGURE 3. Optimal versus non-optimal transformations. **(a)** Constant images  $R$  and  $T$ , where  $\rho^T = \rho^R = 1$ . **(b)** The optimal transformation  $\phi^*$  obtained by our monotone numerical scheme. It is an identity mapping. The figure shows the deformed image of a square grid under  $\phi^*$ , which is the square grid itself. **(c)** The non-optimal transformation  $\phi$  obtained by a non-monotone finite difference scheme in [3]. The figure shows that a square grid is severely deformed under  $\phi$ .

### 6.3. Periodic versus Neumann boundary conditions.

**Example 2** (Figure 4). We revisit the case where  $R$  is a translation of  $T$ . Here we let the true underlying translation  $\phi_{true}^*$  be<sup>6</sup>

$$(\phi_{true}^*)^{-1}(\mathbf{x}) \equiv \mathbf{x} + (0.05, 0.05)^T, \mathbf{x} \in \Omega^T.$$

There is a clear distinction between the transformation  $\phi^*$  returned by the two boundary conditions. The periodic boundary condition yields a constant translation on the entire image, as expected; see Figure 4(d). Figure 4(e) shows that a square grid is translated under  $(\phi^*)^{-1}$  and remains a square grid. In particular, we observe the translation of the boundary of  $[0, 1] \times [0, 1]$ , highlighted by the thick black lines. Also, the entire Figure 4(e) is white, which indicates that no morphing effect occurs. This is a natural consequence of  $\phi^*$  being a pure translation. On the contrary, under the Neumann boundary condition, the resulting transformation in Figure 4(f) is not a constant translation. In Figure 4(g), we observe that in some regions, red/blue color is intense and square elements are severely compressed/expanded. This indicates that the registration under the Neumann boundary condition relies heavily on morphing effect.

**Example 3** (Figure 5). We consider two images  $R$  and  $T$  that are related by a combination of translation and dilation. We specify a true underlying transformation  $\phi_{true}^*$  that satisfies

$$(\phi_{true}^*)^{-1}(\mathbf{x}) \equiv \lambda \left[ \mathbf{x} - \begin{pmatrix} 0.45 \\ 0.45 \end{pmatrix} \right] + \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}, \mathbf{x} \in \Omega^T,$$

with the scaling (or dilating) factor

$$\lambda \equiv \begin{cases} 1.12, & \text{if } \|(\phi_{true}^*)^{-1}(\mathbf{x}) - (0.5, 0.5)^T\|^2 \leq 0.33^2, \\ 1, & \text{otherwise.} \end{cases}$$

The resulting transformation under the periodic boundary condition is shown in Figure 5(d1). To better understand this resulting transformation, we use (25) and decompose it into translation and non-rigid deformation components, represented by green and red arrows respectively in Figure 5(d2). We observe that the non-rigid

<sup>6</sup>The reason why we define  $\phi_{true}^*$  in terms of  $(\phi_{true}^*)^{-1}$  is that it is more intuitive to express the mapping from  $\Omega^T$  to  $\Omega^R$ , whereas  $\phi : \Omega^R \rightarrow \Omega^T$ .



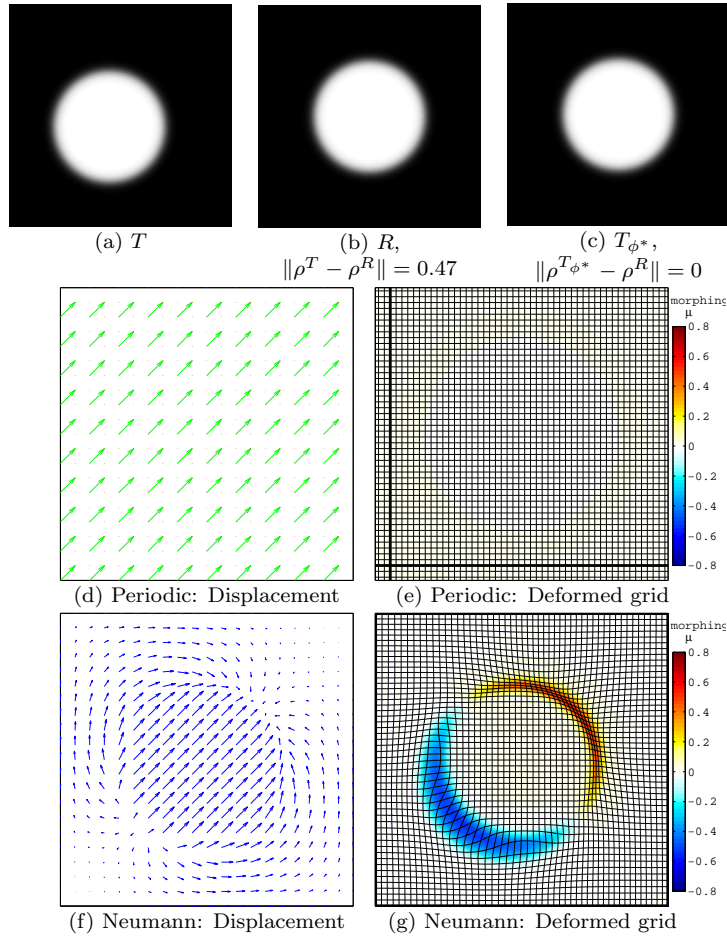


FIGURE 4. Image registration using the periodic and Neumann boundary conditions, where  $T$  and  $R$  are related by translation. (a) Template image  $T$ . (b) Reference image  $R$ . (c) Transformed image  $T_{\phi^*}$  under the *periodic boundary condition*. (d) Displacement of pixels from  $T$  to  $T_{\phi^*}$  under the *periodic boundary condition*, which is a pure translation. (e) A deformed grid obtained by applying the transformation  $(\phi^*)^{-1}$  on a square grid.  $(\phi^*)^{-1}$  is computed under the *periodic boundary condition*. The thick black lines show where the boundary of  $\Omega = [0, 1] \times [0, 1]$  is moved to under  $(\phi^*)^{-1}$ . The color bar is the morphing magnitude  $\mu$ . The intensity of the color shows the degree of morphing effect under  $(\phi^*)^{-1}$ . (f) Displacement of pixels from  $T$  to  $T_{\phi^*}$  under the *Neumann boundary condition*. (g) A deformed grid obtained by applying the transformation  $(\phi^*)^{-1}$  on a square grid.  $(\phi^*)^{-1}$  is computed under the *Neumann boundary condition*.

deformation component (red arrows) is clearly a dilation. Figure 5(e) shows once again that the boundary of  $\Omega = [0, 1] \times [0, 1]$  (the two thick black lines) is translated under  $(\phi^*)^{-1}$ . Also, the deformed grid is a symmetric and isotropic dilation with respect to the center of the circle. However, if we use the Neumann boundary condition, then we cannot identify the underlying combination of translation and

dilation in Figures 5(f) and 5(g). In addition, by a comparison between Figures 5(e) and 5(g), we observe that morphing effect under the periodic boundary condition is mild compared to the Neumann boundary condition.

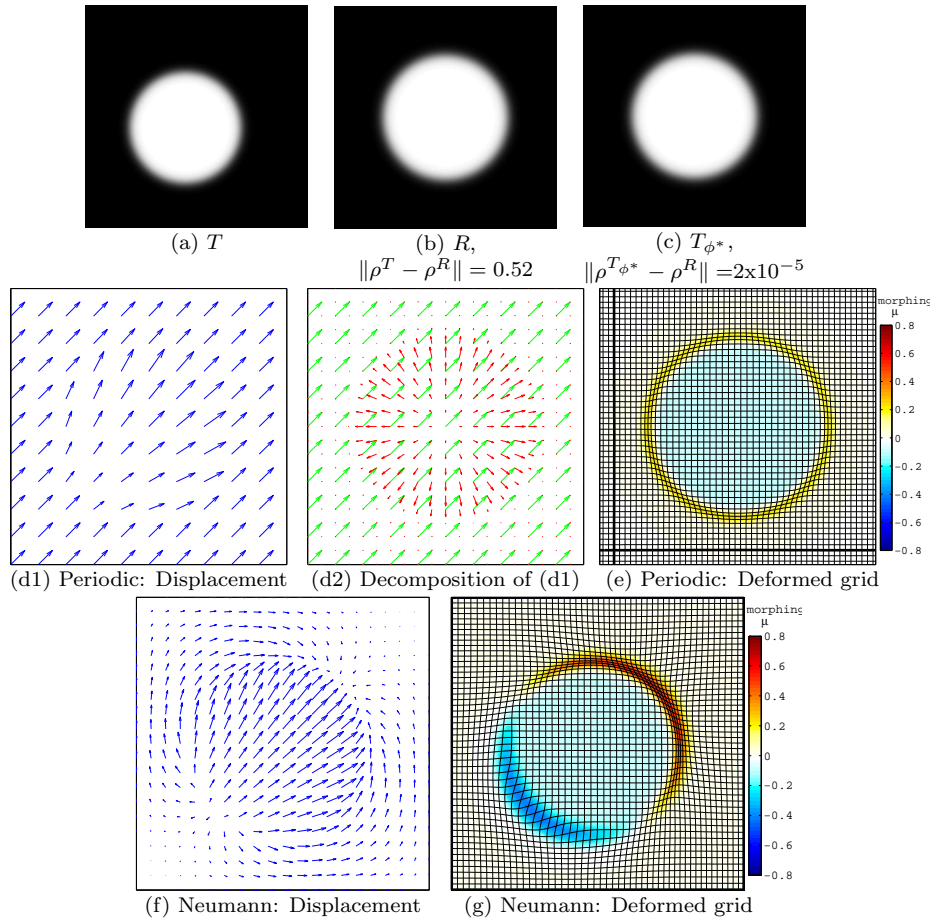


FIGURE 5. Image registration using the periodic and Neumann boundary conditions, where  $T$  and  $R$  are related by a combination of translation and dilation. (a) Template image  $T$ . (b) Reference image  $R$ . (c) Transformed image  $T_{\phi^*}$  under the *periodic boundary condition*. (d1) Displacement of pixels from  $T$  to  $T_{\phi^*}$  under the *periodic boundary condition*. (d2) Decomposition of the displacement into a combination of translation component (green) and dilation component (red). (e) A deformed grid obtained by applying the transformation  $(\phi^*)^{-1}$  on a square grid.  $(\phi^*)^{-1}$  is computed under the *periodic boundary condition*. The thick black lines show where the boundary of  $\Omega = [0, 1] \times [0, 1]$  is moved to under  $(\phi^*)^{-1}$ . The color bar is the morphing magnitude  $\mu$ . The intensity of the color shows the degree of morphing effect under  $(\phi^*)^{-1}$ . (f) Displacement of pixels from  $T$  to  $T_{\phi^*}$  under the *Neumann boundary condition*. (g) A deformed grid obtained by applying the transformation  $(\phi^*)^{-1}$  on a square grid.  $(\phi^*)^{-1}$  is computed under the *Neumann boundary condition*.

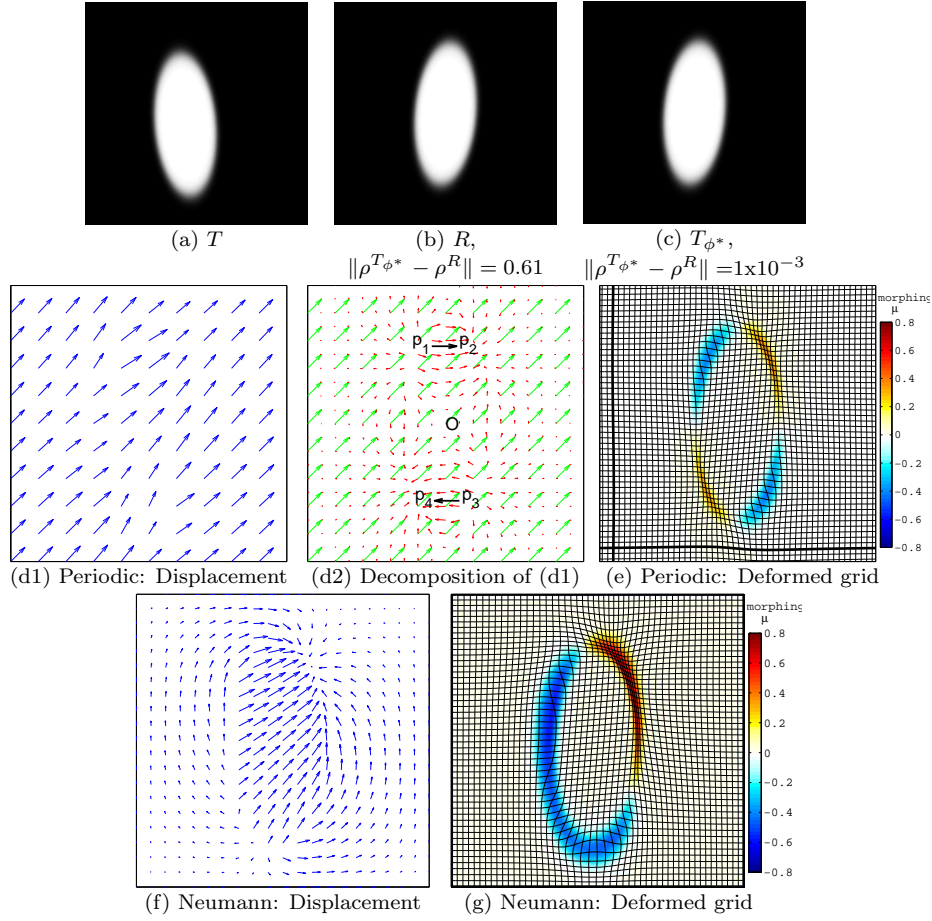


FIGURE 6. Image registration using the periodic and Neumann boundary conditions, where  $T$  and  $R$  are related by a combination of translation and rotation. (a) Template image  $T$ . (b) Reference image  $R$ . (c) Transformed image  $T_{\phi^*}$  under the *periodic boundary condition*. (d1) Displacement of pixels from  $T$  to  $T_{\phi^*}$  under the *periodic boundary condition*. (d2) Decomposition of the displacement into a combination of translation component (green) and local rotation component (red). (e) A deformed grid obtained by applying the transformation  $(\phi^*)^{-1}$  on a square grid.  $(\phi^*)^{-1}$  is computed under the *periodic boundary condition*. The thick black lines show where the boundary of  $\Omega = [0, 1] \times [0, 1]$  is moved to under  $(\phi^*)^{-1}$ . The color bar is the morphing magnitude  $\mu$ . The intensity of the color shows the degree of morphing effect under  $(\phi^*)^{-1}$ . (f) Displacement of pixels from  $T$  to  $T_{\phi^*}$  under the *Neumann boundary condition*. (g) A deformed grid obtained by applying the transformation  $(\phi^*)^{-1}$  on a square grid.  $(\phi^*)^{-1}$  is computed under the *Neumann boundary condition*.

**Example 4** (Figure 6). We consider two images  $R$  and  $T$  where the true underlying transformation

$$(\phi_{true}^*)^{-1}(\mathbf{x}) \equiv \begin{pmatrix} \cos \frac{\pi}{20} & \sin \frac{\pi}{20} \\ -\sin \frac{\pi}{20} & \cos \frac{\pi}{20} \end{pmatrix} \left[ \mathbf{x} - \begin{pmatrix} 0.45 \\ 0.45 \end{pmatrix} \right] + \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$$

is a combination of translation and rotation. Similar to the explanation in Example 3, Figure 6(d2) shows the translation and non-rigid deformation components under the periodic boundary condition. The non-rigid deformation (red arrows) is a *local rotation*. More explicitly, these red arrows rotate the pixels from  $p_1$  to  $p_2$  and from  $p_3$  to  $p_4$  around the center of rotation  $O$ . We note that the resulting transformation is not a *global rotation*. In other words, rotation does not occur in the black background area. Since the mass accumulates at  $p_2$  and  $p_4$ , the intensity will further increase after the pixels are moved toward them. Conversely, the intensity at  $p_1$  and  $p_3$  will further decrease after the pixels are moved away from them. Figure 6(e) illustrates such morphing effect. As a comparison, for the Neumann boundary condition, Figures 6(f) and 6(g) do not reflect the underlying combination of translation and rotation. Once again, morphing effect under the periodic boundary condition is much less severe compared to the Neumann boundary condition.

One may consider incorporating global rotation into the mass transport registration model. However, since the optimal transformation field  $\phi^* = \nabla u$  is curl-free, it would require modifying the mass transport model, which is beyond the scope of this paper.

#### 6.4. Mass transport registration versus empirical two-step registration.

In this section, we apply mass transport image registration on medical images and compare it with the empirical two-step registration, which consists of a pre-registration (rigid transformation) followed by an elastic registration.

**Example 5** (Figure 7). To evaluate the performance of our approach, we perform the following test: We first specify a certain underlying transformation  $\phi_{true}^*$ ; see Figure 7(e). Then we apply this underlying transformation to an image  $T$ , which gives the reference image  $R$ ; see Figure 7(a)-(b). Once obtaining  $T$  and  $R$ , we register these two images using mass transport registration with the periodic boundary condition. We expect that the numerical scheme gives a transformed image  $T_{\phi^*}$  that is the same as  $R$  and recovers the pre-specified underlying transformation. As expected, the resulting transformed image  $T_{\phi^*}$  is the same as  $R$ , where the magnitude of the error is  $10^{-4}$ ; see Figure 7(c)-(d). Moreover, the resulting transformation given by our numerical scheme, as shown in Figure 7(f), is a good approximation of the pre-specified underlying transformation.

We also register  $T$  and  $R$  using the empirical two-step registration, which consists of a pre-registration (rigid transformation) followed by an elastic registration. It is implemented by FAIR package [38]. The results are shown in Figure 8. The error between the transformed image  $T_{\phi}$  and the reference image  $R$  by the empirical approach is not as small as mass transport model. The reason is that mass transport registration applies an additional morphing, which further corrects the registration error to (nearly) zero. The resulting transformation in Figure 8(c) also recovers the underlying transformation in Figure 7(e), although the numerical transformation in the black background area is not the same as the underlying one.

Compared to the more empirical approach, our approach does not require two-step process. In addition, our approach makes the non-rigid model more universal (capable of handling both translation and non-rigid deformation) than the individual method employed in each step of the empirical approach.

**6.5. Quantitative evaluation for the transformations.** To quantitatively measure the accuracy of the resulting transformations, we compute

$$(26) \quad \|\phi^*(\mathbf{x}) - \phi_{true}^*(\mathbf{x})\|_{L_2(\Omega)},$$

which is the difference between the resulting transformation  $\phi^*$  and the true underlying transformation  $\phi_{true}^*$  in  $L_2$  norm. We note that the evaluation of (26) requires knowing in advance the true underlying transformation, which is usually unavailable in practice. However, in Examples 2-5, we pre-specify the underlying transformations between  $T$  and  $R$ , which allows us to perform such evaluation. Table 4 reports the errors of the transformations. In each example, the error given by

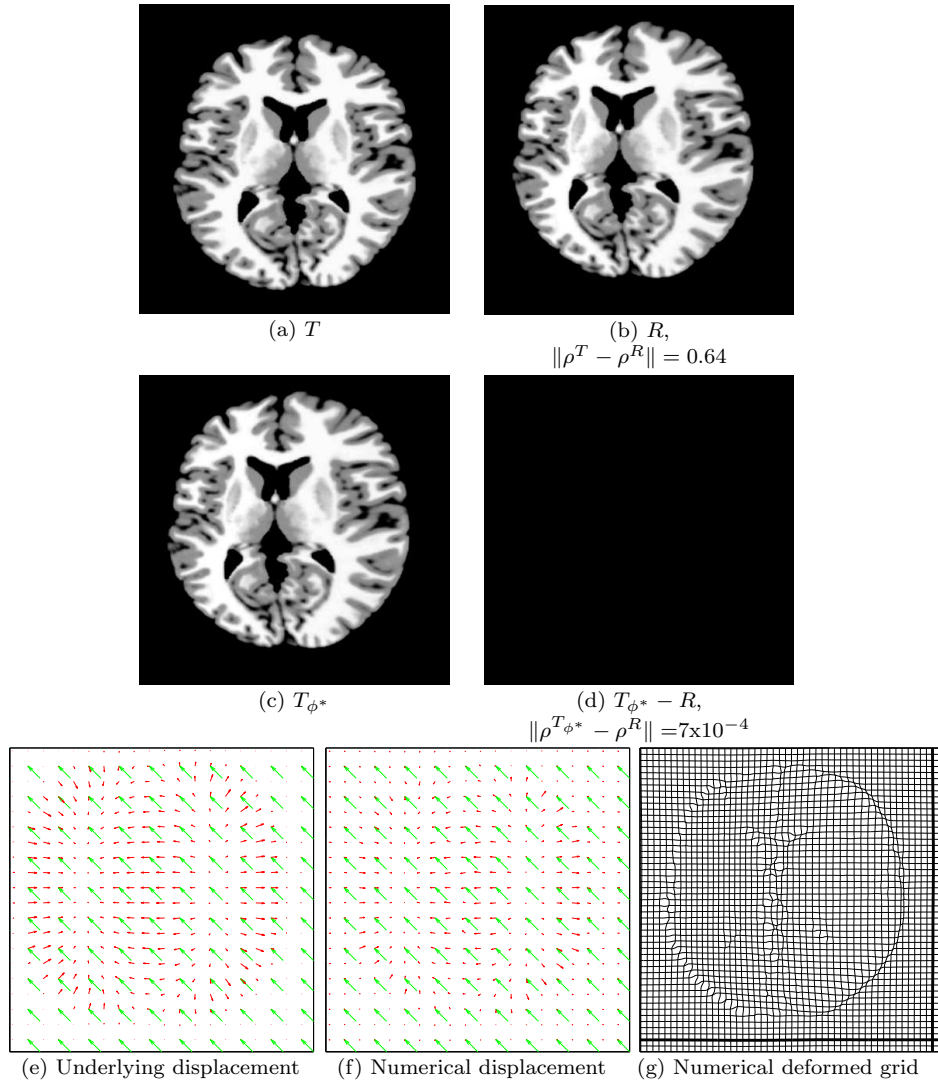


FIGURE 7. Mass transport registration under *periodic boundary condition*. (a) Template image  $T$ . (b) Reference image  $R$ . (c) Transformed image  $T_{\phi^*}$ . (d) Difference between transformed image  $T_{\phi^*}$  and  $R$ . (e) Pre-specified underlying transformation between  $T$  and  $R$ . (f) Transformation given by the numerical scheme, which is a good approximation to the pre-specified underlying transformation in (e). (g) A deformed grid obtained by applying the transformation  $(\phi^*)^{-1}$  on a square grid.

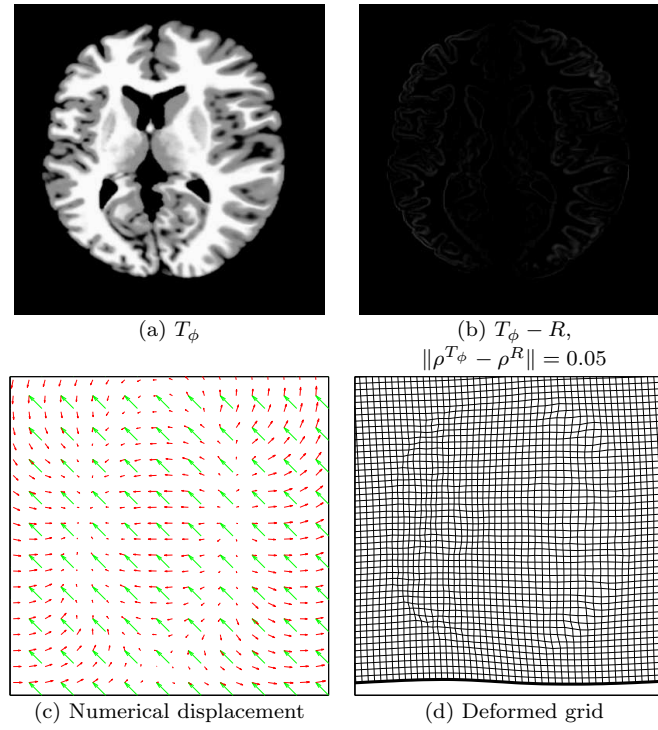


FIGURE 8. Empirical two-step registration, where  $T$  and  $R$  are the same as Figure 7(a)-(b). The registration is implemented by FAIR package [38]. (a) Transformed image  $T_\phi$ . (b) Difference between transformed image  $T_\phi$  and  $R$ . (c) Transformation given by the empirical approach, consisting of a rigid pre-registration (green arrows) and a non-rigid elastic deformation (red arrows). (d) A deformed grid obtained by applying the transformation  $\phi^{-1}$  on a square grid.

the mass transport registration with the periodic boundary condition is compared against either Neumann boundary condition or two-step empirical registration. Table 4 shows that the errors of the motion fields by the mass transport registration with the periodic boundary condition are the smallest. We note that the error for Example 4 is relatively large even under periodic boundary condition, since the mass transport model does not recover global rotation.

**6.6. More examples.** This section includes three more image registration examples that use the mass transport registration under the periodic boundary condition. In these examples, the true underlying transformations are unknown so we focus on the qualitative evaluation only.

**Example 6** (Figure 9). We register two images taken from the brain of a patient. The transformed image  $T_{\phi^*}$  (Figure 9(c)) is the same as the reference image  $R$  (Figure 9(b)). Regarding the resulting transformation, the numerical scheme can automatically translate  $T$  toward  $R$ , followed by a non-rigid deformation; see Figures 9(d)-(e). The two figures also indicate that in the segments where  $T$  and  $R$  look different, the non-rigid deformation and the morphing effect are relatively big. The big morphing effect in these distinct segments can be explained as follows: Moving

Examples	Example 2	Example 3
$\ \phi^*(\mathbf{x}) - \phi_{true}^*(\mathbf{x})\ _{L_2(\Omega)}$	<b>Periodic:</b> 0	<b>Periodic:</b> 0.0053
	Neumann: 0.056	Neumann: 0.056

Examples	Example 4	Example 5
$\ \phi^*(\mathbf{x}) - \phi_{true}^*(\mathbf{x})\ _{L_2(\Omega)}$	<b>Periodic:</b> 0.066	<b>Mass transport, periodic:</b> 0.0055
	Neumann: 0.088	Two-step empirical: 0.011

TABLE 4. The errors of the motion fields (26). The values are computed for Examples 2-5 in Sections 6.3 and 6.4. In each example, the mass transport registration under the periodic boundary condition is compared against either Neumann boundary condition or elastic registration.

the pixels alone may not be able to make these segments in  $T$  look the same as their counterparts in  $R$ . The mass transport model let pixels flow into/out of these segments, resulting in an increase/decrease of intensity, such that the corresponding segments of  $T$  and  $R$  can look the same.

**Example 7** (Figure 10). In this example, the two medical images are very different. Consequentially, the registration between them requires big non-rigid deformation and big morphing effect. This can be seen in Figures 10(d)-(e).

A noticeable fact is that under the transformation, the boundary of  $\Omega = [0, 1] \times [0, 1]$ , marked by the thick black lines in Figure 10(e), is not only translated, but also curved. Such curvature shows that the deformation from the actual object (dark background excluded) in  $T$  to its counterpart in  $R$  is smoothly propagated into the dark background. In other words, the transformation between the actual objects inside  $R$  and  $T$  is not influenced by the events in the dark background. This is different from the Neumann boundary condition, where the transformation between the actual objects is affected by the transformation imposed artificially on the boundary.

**Example 8** (Figure 11). This example is from [27]. The objective is to register the template image, Lena, to the reference image, Tiffany. Figure 11 shows that our proposed algorithm is able to translate Lena’s face towards Tiffany’s face before performing non-rigid deformation. The resulting transformed image of Lena is the same as the reference image, Tiffany, even if the original images are distinct. Reference [27] also achieves this. We note that to the transformed and reference images look very similar due to the combined effect of displacement of pixels and morphing effect, as described in Section 2.2.

**6.7. Unmorphed images.** In some registration problems, morphing effect may be prohibited. In this case, we may want a transformed image without morphing correction, namely,  $\rho_{\phi^*}^{T_{unmorph}}(\mathbf{x}) \equiv \rho^T(\phi^*(\mathbf{x}))$ . When the morphing magnitude of the underlying non-rigid deformation is small (e.g.  $|\mu| < 0.2$ ), the unmorphed transformed image  $T_{\phi^*}^{unmorph}$  may still look similar to the reference image  $R$ , and the difference between  $T_{\phi^*}^{unmorph}$  and  $R$  is visually negligible. Figure 12 shows that the unmorphed image  $T_{\phi^*}^{unmorph}$  by the mass transport model under the periodic boundary condition has the smallest error, compared to Neumann boundary condition or rigid registration only.

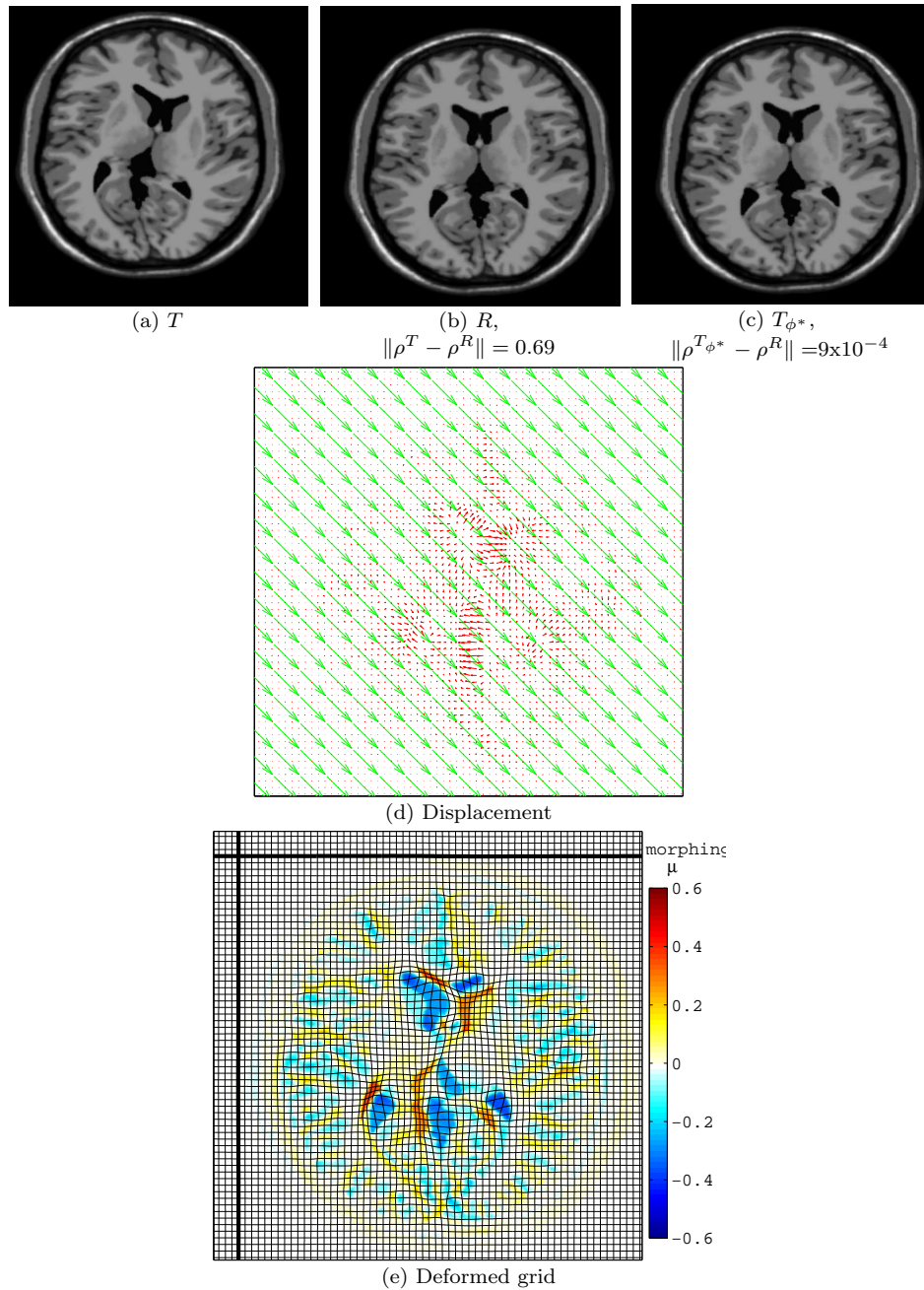


FIGURE 9. Medical image registration using the periodic boundary condition. (a) Template image  $T$ . (b) Reference image  $R$ . (c) Transformed image  $T_{\phi^*}$ . (d) Decomposition of the displacement into a combination of translation component (green) and non-rigid deformation component (red). (e) A deformed grid obtained by applying the transformation  $(\phi^*)^{-1}$  on a square grid.  $(\phi^*)^{-1}$  is computed under *periodic boundary condition*.



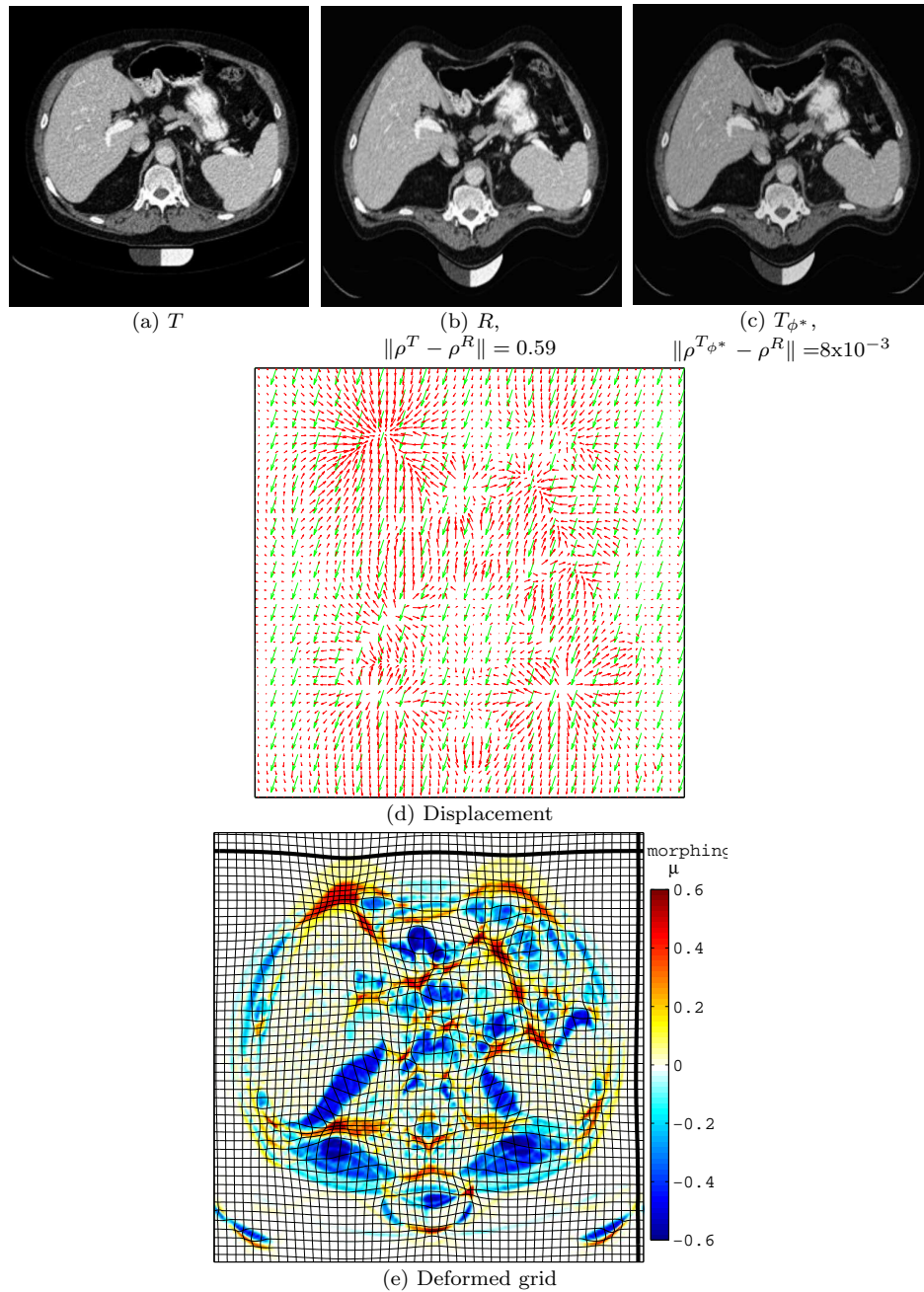


FIGURE 10. Medical image registration using the periodic boundary condition. (a) Template image  $T$ . (b) Reference image  $R$ . (c) Transformed image  $T_{\phi^*}$ . (d) Decomposition of the displacement into a combination of translation component (green) and non-rigid deformation component (red). (e) A deformed grid obtained by applying the transformation  $(\phi^*)^{-1}$  on a square grid.  $(\phi^*)^{-1}$  is computed under *periodic boundary condition*.

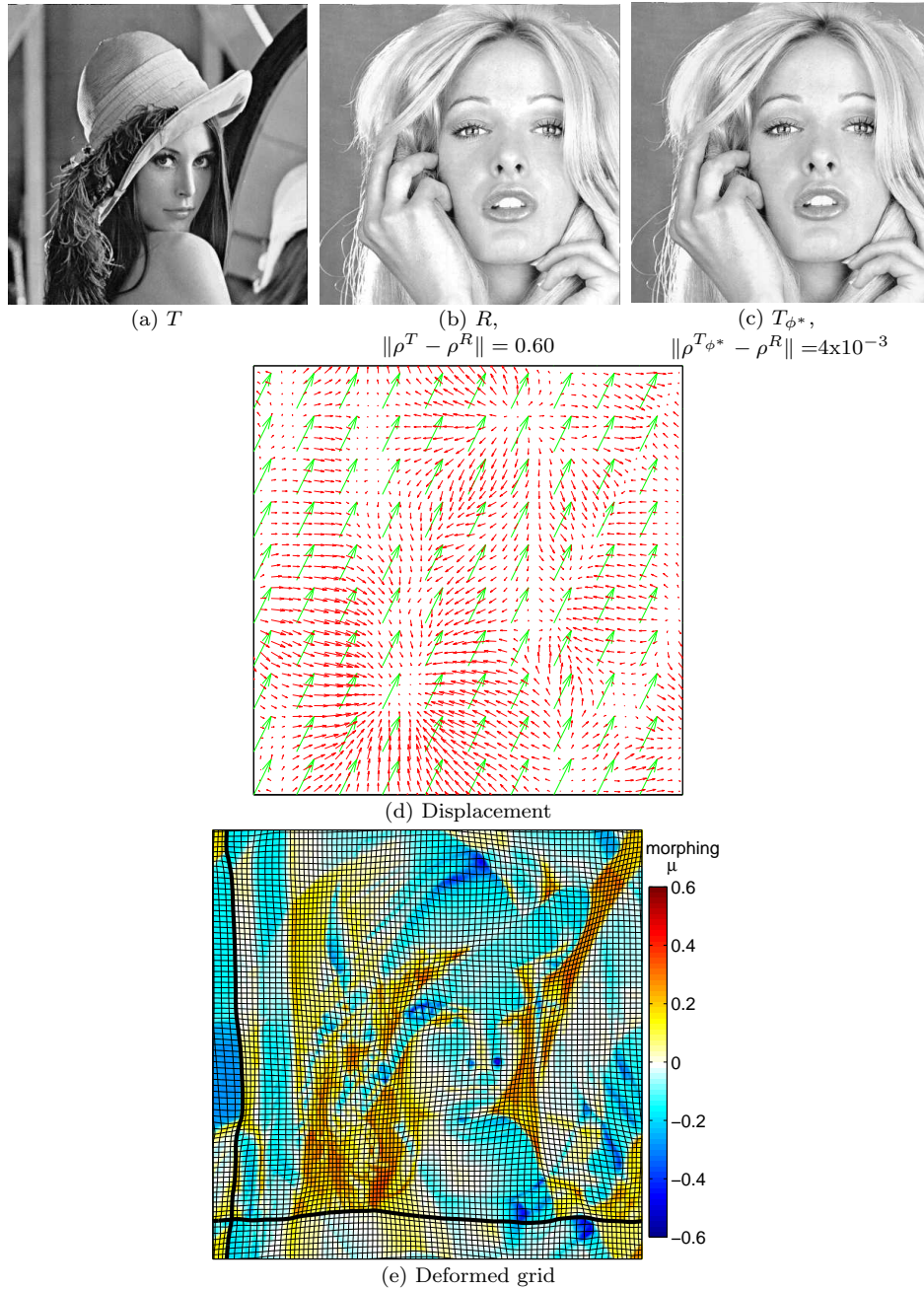


FIGURE 11. Image registration from Lena to Tiffany using the periodic boundary condition. (a) Template image  $T$ . (b) Reference image  $R$ . (c) Transformed image  $T_{\phi^*}$ . (d) Decomposition of the displacement into a combination of translation component (green) and non-rigid deformation component (red). (e) A deformed grid obtained by applying the transformation  $(\phi^*)^{-1}$  on a square grid.  $(\phi^*)^{-1}$  is computed under *periodic boundary condition*.

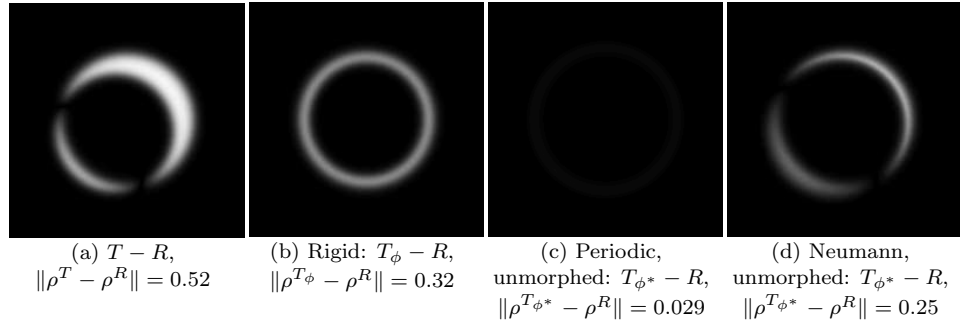


FIGURE 12. Image registration using the periodic and Neumann boundary conditions, where  $T$  and  $R$  are given in Figure 5(a) and Figure 5(b). (a) Difference between  $T$  and  $R$ . (b) Difference between  $T_\phi$  and  $R$ , where  $T_\phi$  is a transformed image under rigid registration only (or,  $T_\phi$  is a translation of  $T$  to align with  $R$ ). (c) Difference between *unmorphed* transformed image  $T_{\phi^*}^{unmorph}$  and  $R$  under the *periodic boundary condition*. (d) Difference between *unmorphed* transformed image  $T_{\phi^*}^{unmorph}$  and  $R$  under the *Neumann boundary condition*.

A modification of the mass transport model, which aims at leaving out the morphing effect, is proposed in [28] and [39]. However, modifying the mass transport formulation for the purpose of dropping morphing effect is beyond the scope of this work.

**6.8. Computational complexity.** We first test the computational complexity of the modified Levenberg-Marquardt algorithm (Table 1) for Example 3 with image sizes of  $100 \times 100$ ,  $200 \times 200$ ,  $400 \times 400$  and  $800 \times 800$  using MATLAB. The number of steps for convergence is roughly a constant 3 as the image size increases. We record the CPU time of (i) corrections of translation kernels (Line 4-7 of Table 1), and (ii) the primary nonlinear solver (Line 8-12 of Table 1) separately. Table 5 shows that as the image size (number of pixels) increases by 4, the CPU time for (i) and (ii) increases by a factor of 8 and (approximately) 16 respectively. The total CPU time for  $800 \times 800$  image is around 20 minutes.

We also test the computational time for different images of the same size  $600 \times 600$ . Since the translation components across different images are different, which is difficult to compare fairly, we skip the comparison of the CPU time for (i). Table 6 shows that for the images where the non-rigid deformation components are relatively larger, the number of steps for convergence and the CPU time for the primary nonlinear solver are also larger.

In addition, we report the convergence of residual versus the number of iteration  $k$ ; see Table 7.

**7. Conclusion.** The objective of this paper is to propose a numerical scheme together with a boundary condition for the mass transport registration model. The main contributions of this paper include that

- our numerical scheme gives an optimality-guaranteed transformation for the mass transport image registration; and
- our periodic boundary condition incorporates both translation and non-rigid deformation into the mass transport model.

Example	Example 3			
Image size	100x100	200x200	400x400	800x800
Number of steps for convergence	5	3	3	3
CPU time for corrections of translation kernels (sec)	1.0	4.6	30	259
CPU time for the primary nonlinear solver (sec)	3.1	7.3	58	1083
Total CPU time (sec)	4.1	11.9	88	1342

TABLE 5. Number of steps for convergence (residual tolerance  $10^{-4}$ ), and CPU time for Example 3 with different image sizes. Here (i) corrections of translation kernels refer to Line 4-7 of Table 1, and (ii) the primary nonlinear solver refers to Line 8-12 of Table 1. The experiments are run in MATLAB.

Examples	Example 3	Example 4	Example 5
Image size	600x600		
Number of steps for convergence	3	3	10
CPU time for the primary nonlinear solver (sec)	147	152	668

Examples	Example 6	Example 7
Image size	600x600	
Number of steps for convergence	10	19
CPU time for the primary nonlinear solver (sec)	627	1613

TABLE 6. Number of steps for convergence (residual tolerance  $10^{-4}$ ), and CPU time for nonlinear solver only for Examples 3-7 with the same image sizes. The experiments are run in MATLAB.

The number of iteration $k$	1	2	3	4	5
Residual	1382	195	2.32	1.71	0.131
The number of iteration $k$	6	7	8	9	10
Residual	0.0236	0.00492	$9.50 \times 10^{-4}$	$3.42 \times 10^{-4}$	$9.41 \times 10^{-5}$

TABLE 7. Residual versus the number of iteration  $k$  for Example 5.

We note that the standard mass transport model using the conventional boundary conditions can only do non-rigid registration. Our computation method can also provide translation component as a by-product.

In the course of achieving these two contributions, we design a mixed standard 7-point stencil and wide stencil finite difference discretization, and prove mathematically that the solution  $u$  is guaranteed to produce the optimal transformation. Also, due to the periodic boundary condition, Jacobian for the discretized system becomes singular. To address this, we propose a modified Levenberg-Marquardt algorithm.

In the experimental results, we first show that our numerical scheme yields the optimal transformation, whereas a non-monotone finite difference scheme gives a non-optimal transformation. In addition, the deformed grid images (Figure 4(e), 5(e), 6(e), 7(g), 9(e) and 10(e)) are smooth and do not contain foldings or crossings. Since these properties are possessed by the optimal transformations, they provide evidences that the resulting transformations are optimal.

We also show in the results that the periodic boundary condition outperforms the Neumann boundary condition. Under the periodic boundary condition, the transformed images  $T_{\phi^*}$  are the same as the reference images  $R$ . Moreover,  $\phi^*$  recovers the underlying combination of translation and non-rigid deformation. In addition, the morphing effect is mild compared to the Neumann boundary condition.

There are two major limitations of the mass transport registration model. One is that morphing effect may misinterpret the physics of the deformation in some applications. The other is that the mass transport model does not recover rotation globally, although it recovers rotation locally. Fixing these limitations is beyond our objective of developing numerical scheme and choosing boundary condition. It would require modifying the mass transport model itself. Nevertheless, model modification, and furthermore, a numerical scheme for the modified model, could be future works. For instance, one possible approach to incorporate global rotation into the mass transport model would be to introduce Killing energy minimization, as investigated by the coauthor of this paper in [11].

Regarding the numerical scheme, the computation could be sped up by multilevel approaches [46, 38] or multigrid algorithms [49], which is another potential future work.

### Appendix A. Proofs of Lemma 5.1 and 5.2.

*Proof of Lemma 5.1.* The proof follows our previous work in [13]. For the standard 7-point stencil discretizations (15) and (17), the truncation errors are both  $O(h^2)$ , since the truncation errors for  $D_x^+ D_x^- U_{i,j}$ ,  $D_y^+ D_y^- U_{i,j}$ ,  $\frac{1}{2}(D_x^+ D_y^+ + D_x^- D_y^-) U_{i,j}$  and  $\frac{1}{2}(D_x^+ D_y^- + D_x^- D_y^+) U_{i,j}$  are all  $O(h^2)$ .

For the wide stencil discretization (18), the truncation error is  $O(h)$ . To prove it, we introduce a smooth test function  $\varphi(\mathbf{x}) \in C^\infty(\Omega)$ . Taylor expansion shows that

$$\varphi_{zz}(\mathbf{x}_{i,j}) - \frac{1}{h} \left[ \mathcal{I}_h \varphi|_{\mathbf{x}_{i,j} + \sqrt{h}(\mathbf{e}_z)_{i,j}} - 2\varphi(\mathbf{x}_{i,j}) + \mathcal{I}_h \varphi|_{\mathbf{x}_{i,j} - \sqrt{h}(\mathbf{e}_z)_{i,j}} \right] = O(h),$$

since the bilinear interpolation  $\mathcal{I}_h \varphi|_{\mathbf{x}_{i,j} \pm \sqrt{h}(\mathbf{e}_z)_{i,j}}$  alone introduces truncation error  $O(h^2)$ , and the central difference with the stencil length  $\sqrt{h}$  is overall  $O((\sqrt{h})^2) = O(h)$  accurate. This leads to  $O(h)$  truncation error for (18).

We will also prove that the discretization of the periodic boundary condition (10) does not introduce any truncation error. Without loss of generality, we only check the discretization of  $(u_x - x)_{x=0} = (u_x - x)_{x=1}$ , which is  $D_x^+ U_{0,j} = D_x^+ U_{N,j} - 1$ . The associated truncation error is then

$$\begin{aligned} & [(\varphi_x(\mathbf{x}_{N,j}) - 1) - \varphi_x(\mathbf{x}_{0,j})] - [(D_x^+ \varphi_{N,j} - 1) - D_x^+ \varphi_{0,j}] \\ &= \frac{1}{2} [\varphi_{xx}(\mathbf{x}_{0,j}) - \varphi_{xx}(\mathbf{x}_{N,j})] h + \frac{1}{6} [\varphi_{xxx}(\mathbf{x}_{0,j}) - \varphi_{xxx}(\mathbf{x}_{N,j})] h^2 + \dots = 0, \end{aligned}$$

Notice that the periodic boundary condition (10) already implies that  $\varphi_{xx}(\mathbf{x}_{0,j}) = \varphi_{xx}(\mathbf{x}_{N,j})$ ,  $\varphi_{xxx}(\mathbf{x}_{0,j}) = \varphi_{xxx}(\mathbf{x}_{N,j})$ , etc. In fact, such equalities hold for any derivatives equal to or higher than second order. As a result, the entire expression

equals to 0, or, the discretization of the periodic boundary condition (10) induces zero truncation error.

Based on these truncation error analysis, we can follow the proof in [13, 22] and conclude that the local truncation error of the numerical scheme (19) goes to 0 when  $h \rightarrow 0$ .  $\square$

*Proof of Lemma 5.2.* Following the procedure in [13], one can verify the following: for the discretized differential operators (15), (17) and (18) under any admissible controls  $(a_{i,j}, \theta_{i,j}) \in \Gamma$ , the coefficient of  $U_{i,j}$  is positive, while the coefficients of  $\{U_{p,q|(p,q) \neq (i,j)}\}$  are non-positive. Hence, the discretized differential operators (15), (17) and (18) are monotonically increasing in  $U_{i,j}$ , and monotonically decreasing in  $\{U_{p,q|(p,q) \neq (i,j)}\}$ .

The gradient term  $\rho^T(u_x(\mathbf{x}_{i,j}), u_y(\mathbf{x}_{i,j}))$  can be discretized by standard upwinding scheme (see [48]), or more generally, Godunov scheme (see [25, 48, 42]). It is well-known that these two schemes are monotone [48].

Eventually, we can follow [13, 22] to conclude that the complete discretized system (19) is monotone.  $\square$

## REFERENCES

- [1] G. Barles and P. E. Souganidis, Convergence of approximation schemes for fully nonlinear second order equations, *Asymptotic Anal.*, **4** (1991), 271–283.
- [2] J.-D. Benamou, Y. Brenier and K. Guittet, [The Monge-Kantorovich mass transfer and its computational fluid mechanics formulation](#), *Internat. J. Numer. Methods Fluids*, **40** (2002), 21–30, ICFD Conference on Numerical Methods for Fluid Dynamics (Oxford, 2001).
- [3] J.-D. Benamou, B. D. Froese and A. M. Oberman, [Two numerical methods for the elliptic Monge-Ampère equation](#), *M2AN Math. Model. Numer. Anal.*, **44** (2010), 737–758.
- [4] K. Böhmer, [On finite element methods for fully nonlinear elliptic equations of second order](#), *SIAM J. Numer. Anal.*, **46** (2008), 1212–1249.
- [5] S. C. Brenner, T. Gudi, M. Neilan and L.-y. Sung,  [\$C^0\$  penalty methods for the fully nonlinear Monge-Ampère equation](#), *Math. Comp.*, **80** (2011), 1979–1995.
- [6] C. Broit, Optimal registration of deformed images.
- [7] L. G. Brown, [A survey of image registration techniques](#), *ACM computing surveys (CSUR)*, **24** (1992), 325–376.
- [8] P. A. Browne, C. J. Budd, C. Piccolo and M. Cullen, [Fast three dimensional r-adaptive mesh redistribution](#), *J. Comput. Phys.*, **275** (2014), 174–196.
- [9] C. J. Budd and J. F. Williams, [Moving mesh generation using the parabolic Monge-Ampère equation](#), *SIAM J. Sci. Comput.*, **31** (2009), 3438–3465.
- [10] L. A. Caffarelli, [Boundary regularity of maps with convex potentials. II](#), *Ann. of Math. (2)*, **144** (1996), 453–496.
- [11] K. Y. Chan and J. W. Wan, [Reconstruction of missing cells by a killing energy minimizing nonrigid image registration](#), in *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, IEEE, 2013, 3000–3003.
- [12] R. Chartrand, B. Wohlberg, K. R. Vixie and E. M. Bollt, A gradient descent solution to the Monge-Kantorovich problem, *Appl. Math. Sci. (Ruse)*, **3** (2009), 1071–1080.
- [13] Y. Chen and J. W. Wan, Monotone mixed narrow/wide stencil finite difference scheme for Monge-Ampère equation, arXiv preprint, [arXiv:1608.00644](#).
- [14] G. E. Christensen, *Deformable Shape Models for Anatomy*, PhD thesis, Washington University Saint Louis, Mississippi, 1994.
- [15] M. G. Crandall, H. Ishii and P.-L. Lions, [User’s guide to viscosity solutions of second order partial differential equations](#), *Bull. Amer. Math. Soc. (N.S.)*, **27** (1992), 1–67.
- [16] M. G. Crandall and P.-L. Lions, [Viscosity solutions of Hamilton-Jacobi equations](#), *Trans. Amer. Math. Soc.*, **277** (1983), 1–42.
- [17] E. J. Dean and R. Glowinski, [Numerical methods for fully nonlinear elliptic equations of the Monge-Ampère type](#), *Comput. Methods Appl. Mech. Engrg.*, **195** (2006), 1344–1386.

- [18] P. Dupuis, U. Grenander and M. I. Miller, [Variational problems on flows of diffeomorphisms for image matching](#), *Quarterly of applied mathematics*, **56** (1998), 587–600.
- [19] X. Feng, R. Glowinski and M. Neilan, [Recent developments in numerical methods for fully nonlinear second order partial differential equations](#), *SIAM Rev.*, **55** (2013), 205–267.
- [20] X. Feng and M. Neilan, [Vanishing moment method and moment solutions for fully nonlinear second order partial differential equations](#), *J. Sci. Comput.*, **38** (2009), 74–98.
- [21] B. Fischer and J. Modersitzki, [Fast inversion of matrices arising in image processing](#), *Numerical Algorithms*, **22** (1999), 1–11.
- [22] P. A. Forsyth and G. Labahn, [Numerical methods for controlled Hamilton-Jacobi-Bellman PDEs in finance](#), *Journal of Computational Finance*, **11** (2007), 1pp.
- [23] B. D. Froese, [A numerical method for the elliptic Monge-Ampère equation with transport boundary conditions](#), *SIAM J. Sci. Comput.*, **34** (2012), A1432–A1459.
- [24] B. D. Froese and A. M. Oberman, [Convergent finite difference solvers for viscosity solutions of the elliptic Monge-Ampère equation in dimensions two and higher](#), *SIAM J. Numer. Anal.*, **49** (2011), 1692–1714.
- [25] S. K. Godunov, [A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics](#), *Mat. Sb. (N.S.)*, **47** (1959), 271–306.
- [26] A. A. Goshtasby, *2-D and 3-D Image Registration: For Medical, Remote Sensing, and Industrial Applications*, John Wiley & Sons, 2005.
- [27] S. Haker and A. Tannenbaum, [Optimal mass transport and image registration](#), in *Variational and Level Set Methods in Computer Vision, 2001. Proceedings. IEEE Workshop on*, IEEE, 2001, 29–36.
- [28] S. Haker, L. Zhu, A. Tannenbaum and S. Angenent, [Optimal mass transport for registration and warping](#), *International Journal of computer vision*, **60** (2004), 225–240.
- [29] D. L. Hill, P. G. Batchelor, M. Holden and D. J. Hawkes, [Medical image registration](#), *Physics in medicine and biology*, **46** (2001), R1.
- [30] M. Irani and S. Peleg, [Improving resolution by image registration](#), *CVGIP: Graphical models and image processing*, **53** (1991), 231–239.
- [31] M. Knott and C. S. Smith, [On the optimal mapping of distributions](#), *Journal of Optimization Theory and Applications*, **43** (1984), 39–49.
- [32] N. V. Krylov, [The control of the solution of a stochastic integral equation](#), *Teor. Veroyatnost. i Primenen.*, **17** (1972), 111–128.
- [33] K. Levenberg, [A method for the solution of certain non-linear problems in least squares](#), *Quart. Appl. Math.*, **2** (1944), 164–168.
- [34] P.-L. Lions, [Hamilton-Jacobi-Bellman equations and the optimal control of stochastic systems](#), in *Proceedings of the International Congress of Mathematicians, Vol. 1, 2 (Warsaw, 1983)*, PWN, Warsaw, 1984, 1403–1417.
- [35] J. A. Maintz and M. A. Viergever, [A survey of medical image registration](#), *Medical image analysis*, **2** (1998), 1–36.
- [36] D. W. Marquardt, [An algorithm for least-squares estimation of nonlinear parameters](#), *J. Soc. Indust. Appl. Math.*, **11** (1963), 431–441.
- [37] J. Modersitzki, *Numerical Methods for Image Registration*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2004, Oxford Science Publications.
- [38] J. Modersitzki, *FAIR: Flexible Algorithms for Image Registration*, vol. 6 of Fundamentals of Algorithms, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2009.
- [39] O. Museyko, M. Stiglmayr, K. Klamroth and G. Leugering, [On the application of the Monge-Kantorovich problem to image registration](#), *SIAM J. Imaging Sci.*, **2** (2009), 1068–1097.
- [40] A. M. Oberman, [Wide stencil finite difference schemes for the elliptic Monge-Ampère equation and functions of the eigenvalues of the Hessian](#), *Discrete Contin. Dyn. Syst. Ser. B*, **10** (2008), 221–238.
- [41] V. I. Oliker and L. D. Prussner, [On the numerical solution of the equation  \$\(\partial^2 z / \partial x^2\)\(\partial^2 z / \partial y^2\) - \(\(\partial^2 z / \partial x \partial y\)\)^2 = f\$  and its discretizations. I](#), *Numer. Math.*, **54** (1988), 271–293.
- [42] S. Osher and C.-W. Shu, [High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations](#), *SIAM J. Numer. Anal.*, **28** (1991), 907–922.
- [43] K. Rohr, *Landmark-based Image Analysis: Using Geometric and Intensity Models*, vol. 21, Springer Science & Business Media, 2001.
- [44] L.-P. Saumier, M. Agueh and B. Khouider, [An efficient numerical algorithm for the  \$L^2\$  optimal transport problem with periodic densities](#), *IMA J. Appl. Math.*, **80** (2015), 135–157.

- [45] A. Sotiras, C. Davatzikos and N. Paragios, [Deformable medical image registration: A survey](#), *IEEE transactions on medical imaging*, **32** (2013), 1153–1190.
- [46] P. Thevenaz, U. E. Ruttimann and M. Unser, [A pyramid approach to subpixel registration based on intensity](#), *IEEE transactions on image processing*, **7** (1998), 27–41.
- [47] J.-P. Thirion, [Image matching as a diffusion process: An analogy with maxwell’s demons](#), *Medical image analysis*, **2** (1998), 243–260.
- [48] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*, Third edition. Springer-Verlag, Berlin, 2009.
- [49] U. Trottenberg, C. W. Oosterlee and A. Schüller, *Multigrid*, Academic Press, Inc., San Diego, CA, 2001, With contributions by A. Brandt, P. Oswald and K. Stüben.
- [50] A. Trounev, Diffeomorphisms groups and pattern matching in image analysis, *International Journal of Computer Vision*, **28** (1998), 213–221.
- [51] P. Viola and W. M. Wells III, Alignment by maximization of mutual information, *International journal of computer vision*, **24** (1997), 137–154.

Received May 2017; revised September 2017.

*E-mail address:* y493chen@uwaterloo.ca

*E-mail address:* justin.wan@uwaterloo.ca