

# An augmented Lagrangian method for solving total variation (TV)-based image registration model

Journal of Algorithms &  
Computational Technology  
Volume 14: 1–16  
© The Author(s) 2020  
Article reuse guidelines:  
sagepub.com/journals-permissions  
DOI: 10.1177/1748302620973534  
journals.sagepub.com/home/act



Noppadol Chumchob<sup>1</sup>  and Ke Chen<sup>2</sup>

## Abstract

Variational methods for image registration basically involve a regularizer to ensure that the resulting well-posed problem admits a solution. Different choices of regularizers lead to different deformations. On one hand, the conventional regularizers, such as the elastic, diffusion and curvature regularizers, are able to generate globally smooth deformations and generally useful for many applications. On the other hand, these regularizers become poor in some applications where discontinuities or steep gradients in the deformations are required. As is well-known, the total (TV) variation regularizer is more appropriate to preserve discontinuities of the deformations. However, it is difficult in developing an efficient numerical method to ensure that numerical solutions satisfy this requirement because of the non-differentiability and non-linearity of the TV regularizer. In this work we focus on computational challenges arising in approximately solving TV-based image registration model. Motivated by many efficient numerical algorithms in image restoration, we propose to use augmented Lagrangian method (ALM). At each iteration, the computation of our ALM requires to solve two subproblems. On one hand for the first subproblem, it is impossible to obtain exact solution. On the other hand for the second subproblem, it has a closed-form solution. To this end, we propose an efficient nonlinear multigrid (NMG) method to obtain an approximate solution to the first subproblem. Numerical results on real medical images not only confirm that our proposed ALM is more computationally efficient than some existing methods, but also that the proposed ALM delivers the accurate registration results with the desired property of the constructed deformations in a reasonable number of iterations.

## Keywords

Image registration, augmented Lagrangian method, nonlinear multigrid method, total variation regularization

Received 4 December 2019; revised 22 July 2020; accepted 26 October 2020

## Introduction

One of important tasks in image analysis is image registration. Generally speaking, it is the process of matching two or more images of the same or similar object obtained from different times, perspectives, and/or imaging sensors. This process can be done mathematically by computing a spatial geometric transformation of deformations (also known as displacement fields) that maps each point in one image onto a corresponding point in the other image with an *optimal* or *meaningful* manner. Image registration has played an important role in several areas of applications. Especially in medical applications, for example, it has been used routinely in medical diagnosis, treatment guidance and monitoring for providing complementary

information. A good survey of the medical applications can be seen in literature<sup>1–4</sup> and the references therein.

Variational methods for image registration have been actively and extensively studied and applied in the field of image analysis. The main idea is to find spatial correspondences between two given images, a

<sup>1</sup>Department of Mathematics, Faculty of Science, Silpakorn University, Nakhon Pathom, Thailand

<sup>2</sup>Centre for Mathematical Imaging Techniques and Department of Mathematical Sciences, University of Liverpool, Liverpool, UK

### Corresponding author:

Ke Chen, Centre for Mathematical Imaging Techniques and Department of Mathematical Sciences, University of Liverpool, Liverpool, UK.  
Email: K.Chen@liverpool.ac.uk



so-called reference image  $R : \Omega \subset \mathbb{R}^2 \rightarrow V \subset \mathbb{R}$  and a so-called template image  $T : \Omega \subset \mathbb{R}^2 \rightarrow V \subset \mathbb{R}$ , where the image domain  $\Omega$  is assumed to be a rectangle. This is usually done by finding an optimal deformation  $\mathbf{u} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ ,  $\mathbf{u} : \mathbf{x} \mapsto \mathbf{u}(\mathbf{x}) = (u_1(\mathbf{x}), u_2(\mathbf{x}))^\top$ , such that the transformed version of the template image  $T_{\mathbf{u}}(\mathbf{x}) = T(\mathbf{x} + \mathbf{u}(\mathbf{x}))$  and the reference image  $R$  are spatially matched, according to an image-to-image dissimilarity measure,  $\mathcal{D}(\mathbf{u})$ . Without loss of generality, this work assumes that  $\Omega = [0, 1]^2 \subset \mathbb{R}^2$  and  $V = [0, 1]$  for 2D gray intensity images. Note that the choice of  $\Omega$  is somewhat arbitrary. The unit square  $\Omega = [0, 1]^2$  is adopted and used in all numerical sections throughout this work in order to make the performance comparison of the proposed numerical method with that of literature.<sup>5</sup>

If the image intensities of the given images  $R$  and  $T$  are comparable, one may have various choices for  $\mathcal{D}$ . Probably the most popular choice for the dissimilarity measure is provided by the so-called *sum of squared differences* (SSD)

$$\mathcal{D}(\mathbf{u}) = \frac{1}{2} \int_{\Omega} (T_{\mathbf{u}}(\mathbf{x}) - R(\mathbf{x}))^2 d\mathbf{x}$$

The registration task is then to solve the minimization problem

$$\min_{\mathbf{u}} \{\mathcal{D}(\mathbf{u})\} \quad (1)$$

where  $\mathbf{u}$  is searched over a set of admissible functions  $\mathcal{U}$  minimizing  $\mathcal{D}$ . As is well-known, the minimization of  $\mathcal{D}$  does not have a unique minimizer and it becomes necessary to impose a constraint on the solution  $\mathbf{u}$  via a regularizer  $\mathcal{R}$ . By modifying (1), the desired deformation  $\mathbf{u}$  is a minimizer of the variational problem

$$\min_{\mathbf{u}} \{\mathcal{J}_{\alpha}(\mathbf{u}) = \mathcal{D}(\mathbf{u}) + \alpha \mathcal{R}(\mathbf{u})\} \quad (2)$$

where  $\alpha > 0$  is a positive constant that compromises the quality of the similarity between  $T_{\mathbf{u}}$  and  $R$ , measured by  $\mathcal{D}$ , and the level of penalty for unwanted deformations, measured by  $\mathcal{R}$ .

In this work, we focus on the regularizer of the form

$$\mathcal{R}(\mathbf{u}) = \sum_{l=1}^2 \mathcal{R}^{\text{TV}}(u_l) \quad (3)$$

where

$$\mathcal{R}^{\text{TV}}(u_l) = \int_{\Omega} |\nabla u_l| d\mathbf{x} = \int_{\Omega} \sqrt{u_{l,x}^2 + u_{l,y}^2} d\mathbf{x}$$

with  $u_{l,x} = \frac{\partial u_l}{\partial x}$  and  $u_{l,y} = \frac{\partial u_l}{\partial y}$  to ensure that the constructed deformation  $\mathbf{u}$  is unique and preserves discontinuities of the deformations. This term is known as the total variation (TV) regularizer.<sup>5-7</sup> In contrast to other traditional regularizers generating globally smooth deformations, e.g. elastic<sup>8</sup> diffusion<sup>9</sup> and curvature regularizers<sup>10,11</sup> the TV regularizer is able to produce locally non-smooth deformations, which are required in matching several moved objects or partially occluded objects in medical applications, particularly at organ boundaries during the breathing induced organ motion.

In order to get a numerical solution of the variational problem (2), the standard gradient descent method can be applied. We first embed the associated Euler-Lagrange (EL) equation of (2) into a dynamic equation and drive it to a steady state. This yields the explicit scheme as given by

$$\begin{cases} \frac{u_1^{[k+1]} - u_1^{[k]}}{\tau} = \alpha \nabla \cdot \left( \frac{\nabla u_1^{[k]}}{\sqrt{(u_{1,x}^{[k]})^2 + (u_{1,y}^{[k]})^2 + \beta}} \right) - f_1(\mathbf{u}^{[k]}), \\ \frac{u_2^{[k+1]} - u_2^{[k]}}{\tau} = \alpha \nabla \cdot \left( \frac{\nabla u_2^{[k]}}{\sqrt{(u_{2,x}^{[k]})^2 + (u_{2,y}^{[k]})^2 + \beta}} \right) - f_2(\mathbf{u}^{[k]}), \end{cases} \quad (4)$$

where  $f_l(\mathbf{u}^{[k]}) = (T_{\mathbf{u}^{[k]}} - R) \partial_{u_l} T_{\mathbf{u}^{[k]}}$ ,  $k = 0, 1, 2, \dots$  is the time step,  $\tau > 0$  is the step size, and  $\beta > 0$  is a small real parameter to avoid zero division; see more details in literature.<sup>12-14</sup> This numerical scheme is easy to implement, but very slow to converge due to the constraint of stability conditions in the step size  $\tau$ . In order to speed up the convergence of (4), we may linearize nonlinear ‘coefficients’ in the associated system and define the iteration step as follows:

$$\begin{cases} \frac{u_1^{[k+1]} - u_1^{[k]}}{\tau} - \alpha \nabla \cdot \left( \frac{\nabla u_1^{[k+1]}}{\sqrt{(u_{1,x}^{[k]})^2 + (u_{1,y}^{[k]})^2 + \beta}} \right) \\ = -f_1(\mathbf{u}^{[k]}), \\ \frac{u_2^{[k+1]} - u_2^{[k]}}{\tau} - \alpha \nabla \cdot \left( \frac{\nabla u_2^{[k+1]}}{\sqrt{(u_{2,x}^{[k]})^2 + (u_{2,y}^{[k]})^2 + \beta}} \right) \\ = -f_2(\mathbf{u}^{[k]}) \end{cases} \quad (5)$$

The idea of the linearized gradient descent (LGD) method in (5) is to linearize and solve the EL equation via a fixed-point (FP) iteration in a similar way to the so-called *Lagged-diffusivity* method<sup>15</sup> or Quasi-Newton scheme.<sup>16,17</sup> For each iteration, a linear system needs to be solved. As can be seen, both gradient descent

methods in (4) and (5) share two drawbacks. At first, these gradient descent methods provide only the approximate solutions of the original problem (2), since the TV regularizer  $\mathcal{R}(\mathbf{u})$  in (3) is replaced with

$$\mathcal{R}_\beta(\mathbf{u}) = \sum_{l=1}^2 \mathcal{R}_\beta^{\text{TV}}(u_l) \quad (6)$$

where

$$\mathcal{R}_\beta^{\text{TV}}(u_l) = \int_{\Omega} |\nabla u_l|_\beta \, \mathbf{d}\mathbf{x} = \int_{\Omega} \sqrt{u_{l,x}^2 + u_{l,y}^2} + \beta \, \mathbf{d}\mathbf{x}$$

is used to avoid non-differentiability of  $\mathcal{R}^{\text{TV}}(u_l)$  and thus approximated to get (4) and (5). On the second, the choice of  $\beta$  will effect on the computational efficiency of the numerical methods and the smoothness of the constructed deformations. Larger the  $\beta$ , more efficient the methods are, whereas more smooth the constructed deformations will be. Therefore the registration performance in smoothing deformations by these gradient descent methods is very sensitive to the parameter  $\beta$ . Indeed, there are a few different methods existing in the literature for solving the associated EL equation of (2). For example, the use of the nonlinear multigrid (NMG) methods can be found in literature.<sup>5-7,18</sup> However, these existing NMG methods also suffer from difficulties related to the non-differentiability and non-linearity of  $\mathcal{R}^{\text{TV}}(u_l)$ . To develop efficient numerical methods for TV-based image registration is still a challenging task and has been an active research area so far.

Recently, the variable-splitting methods are the well-known techniques in the field of image restoration for solving variational models, which require the minimization of nonlinear and non-differentiable functionals. To the best of the author's knowledge, the variable-splitting methods for TV-based image registration model in (2) are still missing in the literature. In this paper, the proposed method is based on the so-called augmented Lagrangian method (ALM). We compare the performance of the proposed ALM with the LGD method in (5) and the NMG method developed by literature.<sup>5</sup> Numerical results show that our proposed ALM is more computationally efficient than these existing methods

The rest of this paper is organized as follows. In the second section, we derive our ALM algorithm to solve the variational problem (2), following its numerical implementation in the third section. In the fourth section, numerical comparisons are carried out to confirm the effectiveness of the proposed ALM. Finally, some concluding remarks are made in the last section.

## The proposed ALM

Before we derive the proposed ALM, we first rewrite (2) as

$$\min_{\mathbf{u}} \left\{ \mathcal{J}(\mathbf{u}) = \mathcal{D}(\mathbf{u}) + \alpha \sum_{l=1}^2 \int_{\Omega} |\nabla u_l| \, \mathbf{d}\mathbf{x} \right\} \quad (7)$$

Next, we introduce an auxiliary variable  $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2)$  such that  $\mathbf{w}_l = \nabla u_l$  and reformulate the minimization problem (7) to the the following constrained minimization problem

$$\min_{\mathbf{u}, \mathbf{w}} \mathcal{J}(\mathbf{u}, \mathbf{w}) \quad (8)$$

$$\mathcal{J}(\mathbf{u}, \mathbf{w}) = \mathcal{D}(\mathbf{u}) + \alpha \sum_{l=1}^2 \int_{\Omega} |\mathbf{w}_l| \, \mathbf{d}\mathbf{x}$$

$$\text{s.t. } \mathbf{w}_l = \nabla u_l \text{ for } l = 1, 2$$

For the minimization of (8), this work proposes to use the ALM and rewrite the constrained minimization problem (8) into an unconstrained minimization problem. We define the augmented Lagrangian functional for the above constrained minimization as follows

$$\begin{aligned} \mathcal{L}(\mathbf{u}, \mathbf{w}; \lambda_1, \lambda_2) &= \mathcal{J}(\mathbf{u}, \mathbf{w}) + \sum_{l=1}^2 \frac{\theta_l}{2} \int_{\Omega} |\mathbf{w}_l - \nabla u_l|^2 \, \mathbf{d}\mathbf{x} \\ &\quad + \sum_{l=1}^2 \int_{\Omega} \lambda_l \cdot (\mathbf{w}_l - \nabla u_l) \, \mathbf{d}\mathbf{x} \end{aligned} \quad (9)$$

where  $\theta_1, \theta_2$  are the positive penalty parameters and  $\lambda_1, \lambda_2$  are the Lagrange multipliers.

---

**Algorithm 1.** ALM for TV-based image registration

1) Initialization: set  $m=0$ , choose  $\theta_1, \theta_2 > 0$  and  $\lambda_1^{(0)}, \lambda_2^{(0)}$ .

2) Repeat

2.1) Compute  $(\mathbf{u}^{(m)}, \mathbf{w}^{(m)})$  as an (approximate) minimizer of the augmented Lagrangian functional with the Lagrange multipliers  $\lambda_1^{(m-1)}, \lambda_2^{(m-1)}$ , i.e.,

$$(\mathbf{u}^{(m)}, \mathbf{w}^{(m)}) \approx \underset{\mathbf{u}, \mathbf{w}}{\operatorname{argmin}} \mathcal{L}(\mathbf{u}, \mathbf{w}; \lambda_1^{(m-1)}, \lambda_2^{(m-1)}) \quad (10)$$

2.2) Update Lagrange multipliers

$$\lambda_1^{(m)} \leftarrow \lambda_1^{(m-1)} + \theta_1 (\mathbf{w}_1^{(m)} - \nabla u_1^{(m)}) \quad (11)$$

$$\lambda_2^{(m)} \leftarrow \lambda_2^{(m-1)} + \theta_2(\mathbf{w}_2^{(m)} - \nabla u_2^{(m)}) \quad (12)$$

until a stopping rule for ALM method is satisfied.

We propose an iterative algorithm to solve the minimization of  $\mathcal{L}(\mathbf{u}, \mathbf{w}; \lambda_1, \lambda_2)$ ; see Algorithm 1. Since  $\mathbf{u}$  and  $\mathbf{w}$  are coupled together in the minimization problem (9), it is very difficult to solve all variables simultaneously. We separate the minimization problem into two sub-problems and develop an alternating minimization procedure to approximate the solution. This process is repeated until one of the following stopping rules is satisfied:

$$\text{RelSSD}^{(m)} = \frac{\mathcal{D}(\mathbf{u}^{(m)})}{\mathcal{D}(\mathbf{u}^{(0)})} < \epsilon_1, \quad (13)$$

$$\max \left\{ \frac{\|u_1^{(m)} - u_1^{(m-1)}\|}{\|u_1^{(m)}\|}, \frac{\|u_2^{(m)} - u_2^{(m-1)}\|}{\|u_2^{(m)}\|} \right\} < \epsilon_2, \quad (14)$$

$$\max \left\{ \frac{\|\mathbf{w}_1^{(m)} - \nabla u_1^{(m)}\|}{|\Omega|}, \frac{\|\mathbf{w}_2^{(m)} - \nabla u_2^{(m)}\|}{|\Omega|} \right\} < \epsilon_3, \quad (15)$$

$$\max \left\{ \frac{\|\lambda_1^{(m)} - \lambda_1^{(m-1)}\|}{\|\lambda_1^{(m)}\|}, \frac{\|\lambda_2^{(m)} - \lambda_2^{(m-1)}\|}{\|\lambda_2^{(m)}\|} \right\} < \epsilon_4, \quad (16)$$

$$m \geq \epsilon_5, \quad (17)$$

where  $\epsilon_1, \epsilon_2, \epsilon_3$  and  $\epsilon_4$  denote the predefined small positive numbers, and  $\epsilon_5$  is the maximum iteration. Here  $m$  denotes the index of the current iteration.

The two-subproblems are as follows:

**u-subproblem.** Given  $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2)$ ,  $\lambda_1$  and  $\lambda_2$ , we search a minimizer  $\mathbf{u} = (u_1, u_2)^\top$  of the minimization problem:

$$\min_{\mathbf{u}} \mathcal{J}(\mathbf{u}) \quad (18)$$

$$\mathcal{J}(\mathbf{u}) = \mathcal{D}(\mathbf{u}) + \sum_{l=1}^2 \frac{\theta_l}{2} \int_{\Omega} |\mathbf{w}_l - \nabla u_l|^2 dx - \sum_{l=1}^2 \int_{\Omega} \nabla u_l \cdot \lambda_l dx \quad (19)$$

According to the calculus of variation, the solution of  $\mathbf{u}$ -subproblem is determined by the associated EL equation

$$\begin{cases} -\theta_1 \Delta u_1 + f_1(\mathbf{u}) = g_1, \\ -\theta_2 \Delta u_2 + f_2(\mathbf{u}) = g_2, \end{cases} \quad (20)$$

subject to the homogeneous Neumann boundary conditions  $\frac{\partial u_i}{\partial \mathbf{n}} = 0$ , where  $\mathbf{n}$  denotes the unit outward

normal vector on the image boundary  $\partial\Omega$ ,  $f_l(\mathbf{u}) = (T_u - R)\partial_{u_l} T_u$ ,  $g_l = -\theta_l \nabla \cdot \mathbf{w}_l - \nabla \cdot \lambda_l$ , and  $l=1, 2$ .  
**w-subproblem.** Given  $\mathbf{u}$ ,  $\lambda_1$  and  $\lambda_2$ , the following minimization problem

$$\min_{\mathbf{w}} \mathcal{J}(\mathbf{w}) \quad (21)$$

$$\begin{aligned} \mathcal{J}(\mathbf{w}) = & \alpha \sum_{l=1}^2 \int_{\Omega} |\mathbf{w}_l| dx + \sum_{l=1}^2 \frac{\theta_l}{2} \int_{\Omega} |\mathbf{w}_l - \nabla u_l|^2 dx \\ & + \sum_{l=1}^2 \int_{\Omega} \mathbf{w}_l \cdot \lambda_l dx, \end{aligned} \quad (22)$$

is solved for  $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2)$ . Despite the fact that the variables  $\mathbf{w}_1$  and  $\mathbf{w}_2$  do not decouple we can still explicitly solve this minimization problem for  $\mathbf{w}$  using the generalized shrinkage formula as used in:

$$\mathbf{w}_l = \max \left( |\mathcal{S}_l| - \frac{\alpha}{\theta_l}, 0 \right) \frac{\mathcal{S}_l}{|\mathcal{S}_l|} \quad (23)$$

with the convection ( $\frac{0}{|0|} = 0$ ), where  $\mathcal{S}_l = \nabla u_l - \frac{\lambda_l}{\theta_l}$  and  $l=1, 2$ . Note that several numerical methods, such as iterative reweighted least squares and interior-point method, have been applied to the minimization of subproblems as represented in (21), but these methods cannot usually converge to satisfactory solutions, especially for large-scale problems. To guarantee the convergence of minimization of subproblem (21), as well as to reduce the costs of numerical computation, generalized shrinkage/thresholding functions as given in (23) are employed as the solvers; see literature.<sup>19-21</sup>

## Numerical implementation

In this section, we present the details of how to solve the equation (20) and update the variables  $\mathbf{w}_1, \mathbf{w}_2, \lambda_1$ , and  $\lambda_2$  for each iteration.

### Finite difference discretization

In order to discretize the EL equation (20), let

$$\left( u_l^h \right)_{ij} = u_l^h(x_{1_i}, x_{2_j}), \quad \left( \mathbf{u}^h \right)_{ij} = \left( (u_1^h)_{ij}, (u_2^h)_{ij} \right)^\top$$

$$\left( w_{l,1}^h \right)_{ij} = w_{l,1}^h(x_{1_i}, x_{2_j}), \quad \left( w_{l,2}^h \right)_{ij} = w_{l,2}^h(x_{1_i}, x_{2_j}),$$

$$\left( \mathbf{w}_l^h \right)_{ij} = \left( (w_{l,1}^h)_{ij}, (w_{l,2}^h)_{ij} \right), \quad \left( g_l^h \right)_{ij} = g_l^h(x_{1_i}, x_{2_j}),$$

denote the grid functions for  $l=1, 2$  with the grid spacing  $h = (h_1, h_2) = (1/n_1, 1/n_2)$ , where the integers  $n_1 = 1/h_1$  and  $n_2 = 1/h_2$  are the number of uniform intervals in the  $x_1$  and  $x_2$  coordinate directions. Each grid point  $\mathbf{x}$  in the discretized domain  $\Omega_h \subset \Omega$  is given by  $\mathbf{x} = (x_{1_i}, x_{2_j})^\top = \left(\frac{(2i-1)h_1}{2}, \frac{(2j-1)h_2}{2}\right)^\top$  for  $1 \leq i \leq n_1$  and  $1 \leq j \leq n_2$ .

Applying the finite difference approximations with (20), the discrete EL equation at a grid point  $(x_{1_i}, x_{2_j})$  over the discrete domain  $\Omega_h$  is given by

$$\begin{cases} \underbrace{-\theta_1 \Delta^h(u_1^h)_{i,j} + f_1^h(\mathbf{u}^h)_{i,j}}_{\mathcal{N}_1^h(\mathbf{u}^h)_{i,j}} = (g_1^h)_{i,j}, \\ \underbrace{-\theta_2 \Delta^h(u_2^h)_{i,j} + f_2^h(\mathbf{u}^h)_{i,j}}_{\mathcal{N}_2^h(\mathbf{u}^h)_{i,j}} = (g_2^h)_{i,j}, \end{cases} \quad (24)$$

with the following notation

$$\begin{aligned} \delta_{x_1}^-(u_l^h)_{i,j} &= ((u_l^h)_{i,j} - (u_l^h)_{i-1,j})/h_1, \\ \delta_{x_1}^+(u_l^h)_{i,j} &= ((u_l^h)_{i+1,j} - (u_l^h)_{i,j})/h_1, \\ \delta_{x_2}^-(u_l^h)_{i,j} &= ((u_l^h)_{i,j} - (u_l^h)_{i,j-1})/h_2, \\ \delta_{x_2}^+(u_l^h)_{i,j} &= ((u_l^h)_{i,j+1} - (u_l^h)_{i,j})/h_2, \\ \nabla^h(u_l^h)_{i,j} &= (\delta_{x_1}^+(u_l^h)_{i,j}, \delta_{x_2}^+(u_l^h)_{i,j})^\top, \\ \nabla^h \cdot (\mathbf{w}_l^h)_{i,j} &= \delta_{x_1}^-(w_{l,1}^h)_{i,j} + \delta_{x_2}^-(w_{l,2}^h)_{i,j}, \\ -\Delta^h(u_l^h)_{i,j} &= -\nabla^h \cdot (\nabla^h(u_l^h)_{i,j}), \\ &= \frac{1}{h_1^2} ((\Sigma)_{i,j}^h(u_l^h)_{i,j} - (\bar{\Sigma})_{i,j}^h(u_l^h)_{i,j}), \\ (\Sigma)_{i,j}^h &= 2(1 + \gamma^2), \quad \gamma = h_1/h_2, \end{aligned}$$

$$\begin{aligned} (\bar{\Sigma})_{i,j}^h(u_l^h)_{i,j} &= (u_l^h)_{i+1,j} + (u_l^h)_{i-1,j} + \gamma^2(u_l^h)_{i,j+1} \\ &\quad + \gamma^2(u_l^h)_{i,j-1}, \end{aligned}$$

$$\begin{aligned} f_1^h(\mathbf{u}^h)_{i,j} &= f_1^h(u_1^h, u_2^h)_{i,j} \\ &= (T_{i,j}^{h^*} - R_{i,j}^h) \frac{T_{i+1,j}^{h^*} - T_{i-1,j}^{h^*}}{2h_1}, \\ f_2^h(\mathbf{u}^h)_{i,j} &= f_2^h(u_1^h, u_2^h)_{i,j} \end{aligned}$$

$$= (T_{i,j}^{h^*} - R_{i,j}^h) \frac{(T_{i,j+1}^{h^*} - T_{i,j-1}^{h^*})}{2h_2},$$

$$T_{i,j}^{h^*} = T^h(i + (u_1^h)_{i,j}, j + (u_2^h)_{i,j}),$$

$$\begin{aligned} (g_l^h)_{i,j} &= -\delta_{x_1}^-(\theta_l(w_{l,1}^h)_{i,j} + (\lambda_{l,1}^h)_{i,j}) \\ &\quad - \delta_{x_2}^-(\theta_l(w_{l,2}^h)_{i,j} + (\lambda_{l,2}^h)_{i,j}). \end{aligned}$$

We note that all finite difference approximations need to be adjusted at the image boundary  $\partial\Omega_h$  using the approximations of the boundary conditions

$$(u_l^h)_{i,1} = (u_l^h)_{i,2}, \quad (u_l^h)_{i,n_2} = (u_l^h)_{i,n_2-1},$$

$$(u_l^h)_{1,j} = (u_l^h)_{2,j}, \quad (u_l^h)_{n_1,j} = (u_l^h)_{n_1-1,j}.$$

In the following subsections the symbols ‘ $h$ ’ and ‘ $(\cdot)_{i,j}^h$ ’ will sometimes drop for simplicity.

### Nonlinear multigrid method for solving $u$ -subproblem

The most difficult part of the proposed ALM is the solution of the EL equation for  $u$ -subproblem in (20). This subproblem is nonlinear and impossible to obtain a closed-form solution. It therefore requires one to approximately solve the corresponding nonlinear discrete system via some iterative method (e.g. a FP method and a Newton-type method). As is well-known, the computational costs of iterative methods are too expensive and we are motivated to develop an efficient numerical method for reducing the computational work.

In this section, we propose a nonlinear multigrid (NMG) method due to literature.<sup>22</sup> The basic idea is to accurate the convergence of some basic iterative method on the finest grid by relying on the complementary interplay of smoothing and coarse-grid correction principles. For a more comprehensive treatment of MG methods in the area of image registration, we refer literature<sup>5,11,18,23–28</sup> and references therein.

Let us denote the nonlinear discrete system in (24) using the following notation:

$$\begin{cases} \mathcal{N}_1^h(\mathbf{u}^h) = g_1^h, \\ \mathcal{N}_2^h(\mathbf{u}^h) = g_2^h \end{cases} \quad (25)$$

where  $\mathbf{u}^h$  denotes the exact solution. Let  $\bar{\mathbf{u}}^h = (\bar{u}_1^h, \bar{u}_2^h)^\top$  be an approximate solution of  $\mathbf{u}^h$  resulting from applying a few iterations of the smoother on a fine-grid  $\Omega_h$  (*pre-smoothing step*). Define the algebraic error as

$e^h = \mathbf{u}^h - \bar{\mathbf{u}}^h$ . Therefore, the residual equation is given by

$$\mathcal{N}_l^h(\bar{\mathbf{u}}^h + \mathbf{e}^h) - \mathcal{N}_l^h(\bar{\mathbf{u}}^h) = \mathbf{g}_l^h - \mathcal{N}_l^h(\bar{\mathbf{u}}^h) = \mathbf{r}_l^h \quad (l = 1, 2).$$

To correct  $\bar{\mathbf{u}}^h$  numerically on  $\Omega_h$ ,  $\mathbf{e}^h$  is required to compute, but this computation is considerably and prohibitively expensive on  $\Omega_h$ . Since the smoother in the pre-smoothing step removes the high frequency component of the error, one can represent the nonlinear system to the coarse grid  $\Omega_H$  as given by

$$\begin{aligned} \underbrace{\mathcal{N}_l^h(\bar{\mathbf{u}}^h + \mathbf{e}^h)}_{\mathcal{N}_l^h(\mathbf{u}^h)} &= \underbrace{\mathbf{r}_l^h + \mathcal{N}_l^h(\bar{\mathbf{u}}^h)}_{\mathbf{g}_l^h} \\ &\downarrow \\ \underbrace{\mathcal{N}_l^H(\bar{\mathbf{u}}^H + \mathbf{e}^H)}_{\mathcal{N}_l^H(\mathbf{u}^H)} &= \underbrace{\mathbf{r}_l^H + \mathcal{N}_l^H(\bar{\mathbf{u}}^H)}_{\mathbf{g}_l^H}, \end{aligned} \quad (26)$$

where  $H$  is the index for the new cell size  $H_1 \times H_2$  with  $H_1 \geq h_1$  and  $H_2 \geq h_2$ . Next, the nonlinear residual equation (27) on  $\Omega_H$  has to be solved by an efficient method (*the coarsest grid solver*). The coarse-grid correction  $\mathbf{e}^H = \mathbf{u}^H - \bar{\mathbf{u}}^H$  on  $\Omega_H$  is then interpolated back to that on the fine grid  $\mathbf{e}^h$ , which can now be used to update  $\bar{\mathbf{u}}^h$  on  $\Omega_h$  by using  $\bar{\mathbf{u}}_{new}^h = \bar{\mathbf{u}}^h + \mathbf{e}^h$  (*coarse-grid correction step*). Finally, the smoother is again performed with a few iterations on  $\Omega_h$  to remove high frequency parts of the interpolated error (*post-smoothing step*). This procedure is known as a two-grid cycle, and it can be extended to a MG method with recursive application.

Below, we provide the technical details for our MG components.

1. The *standard coarsening method* is used in  $\Omega_H$  by doubling the grid spacing in each space direction — i.e.  $h_l \rightarrow 2h_l = H_l$  for  $l = 1, 2$ .
2. The *restriction operator*  $I_h^H$  and *interpolation operators*  $I_H^h$  are used as the intergrid transfer operators and determined by the averaging and bilinear interpolation techniques — cf. literature<sup>29–33</sup> for more details.
3. The *discretization coarse grid approximation method* is used to compute the coarse-grid operators.
4. The *coarsest grid solver* is the LGD method.
5. The *MG cycle* is  $V(\nu_1; \nu_2)$ -cycle, where  $\nu_1$  and  $\nu_2$  denote respectively the numbers of pre- and post-smoothing steps.
6. The *MG smoother* to be discussed shortly in the next section is obtained from a coupled outer-inner iteration method using a relaxation parameter  $\omega$ .

The implementation of the proposed NMG method can be summarized as follows:  $\bar{\mathbf{u}}^h \leftarrow \text{NMGCYC}(\bar{\mathbf{u}}^h, \mathbf{g}_1^h, \mathbf{g}_2^h, \mathbf{R}^h, \mathbf{T}^h, \nu_1, \nu_2, \theta_1, \theta_2, \omega, \text{Siter})$

1) If  $\Omega_h = \text{coarsest grid}$  ( $|\Omega_h| = 4 \times 4$ ), solve (25) using the LGD method and then stop.

Else continue with following step.

2) Pre-smoothing:

For  $k = 1$  to  $\nu_1$ ,

$[\bar{\mathbf{u}}^h] \leftarrow \text{Smoother}(\bar{\mathbf{u}}^h, \mathbf{g}_1^h, \mathbf{g}_2^h, \mathbf{R}^h, \mathbf{T}^h, \theta_1, \theta_2, \omega, \text{Siter})$

(Siter represents the maximum number of the inner iterations)

3) Restriction to the coarse grid:

$$\bar{\mathbf{u}}_1^H \leftarrow I_h^H \bar{\mathbf{u}}_1^h, \bar{\mathbf{u}}_2^H \leftarrow I_h^H \bar{\mathbf{u}}_2^h, \mathbf{R}^H \leftarrow I_h^H \mathbf{R}^h, \mathbf{T}^H \leftarrow I_h^H \mathbf{T}^h$$

4) Set the initial solution for the coarse-grid problem:  $[\tilde{\mathbf{u}}_1^H, \tilde{\mathbf{u}}_2^H] \leftarrow [\bar{\mathbf{u}}_1^H, \bar{\mathbf{u}}_2^H]$

5) Compute the new right-hand side for the coarse-grid problem:

$$\mathbf{g}_1^H \leftarrow I_h^H(\mathbf{g}_1^h - \mathcal{N}_1^h(\bar{\mathbf{u}}^h)) + \mathcal{N}_1^H(\bar{\mathbf{u}}^H),$$

$$\mathbf{g}_2^H \leftarrow I_h^H(\mathbf{g}_2^h - \mathcal{N}_2^h(\bar{\mathbf{u}}^h)) + \mathcal{N}_2^H(\bar{\mathbf{u}}^H)$$

6) Implement the NMG method on the coarse-grid problem:

$$\bar{\mathbf{u}}^H \leftarrow \text{NMGCYC}(\bar{\mathbf{u}}^H, \mathbf{g}_1^H, \mathbf{g}_2^H, \mathbf{R}^H, \mathbf{T}^H, \nu_1, \nu_2, \theta_1, \theta_2, \omega, \text{Siter})$$

7) Add the coarse-grid corrections:

$$\bar{\mathbf{u}}_1^h \leftarrow \bar{\mathbf{u}}_1^h + I_H^h(\bar{\mathbf{u}}_1^H - \tilde{\mathbf{u}}_1^H), \bar{\mathbf{u}}_2^h \leftarrow \bar{\mathbf{u}}_2^h + I_H^h(\bar{\mathbf{u}}_2^H - \tilde{\mathbf{u}}_2^H)$$

8) Post-smoothing:

For  $k = 1$  to  $\nu_2$ ,

$[\bar{\mathbf{u}}^h] \leftarrow \text{Smoother}(\bar{\mathbf{u}}^h, \mathbf{g}_1^h, \mathbf{g}_2^h, \mathbf{R}^h, \mathbf{T}^h, \theta_1, \theta_2, \omega, \text{Siter})$

To solve  $\mathbf{u}$ -subproblem using (25) numerically, the proposed NMG method is stopped if the maximum number of the MG cycles  $\varepsilon_1$  is reached or the mean of the relative residuals obtained from the discrete EL equation (25) is smaller than a small prescribed number  $\varepsilon_2 > 0$ .

Finally, the pseudo-code implementation of the proposed NMG method can be given in Algorithm 2.

---

**Algorithm 2** The NMG algorithm for  $\mathbf{u}$ -subproblem

$$\bar{\mathbf{u}}^h \leftarrow \text{NMG}(\bar{\mathbf{u}}^h, \mathbf{g}_1^h, \mathbf{g}_2^h, \mathbf{R}^h, \mathbf{T}^h, \nu_1, \nu_2, \theta_1, \theta_2, \omega, \text{Siter}, \vec{\varepsilon})$$

1) Initialization: select  $\vec{\varepsilon} = (\varepsilon_1, \varepsilon_2)^\top$ , set  $K = 0$ ,  $[\bar{\mathbf{u}}^h]^{[K]} = \bar{\mathbf{u}}^h$  and  $\text{RelRes} = \varepsilon_2 + 1$ .

2) Repeat

2.1)  $[\bar{\mathbf{u}}^h]^{[K+1]} \leftarrow \text{NMGCYC}([\bar{\mathbf{u}}^h]^{[K]}, \mathbf{g}_1^h, \mathbf{g}_2^h, \mathbf{R}^h, \mathbf{T}^h, \nu_1, \nu_2, \theta_1, \theta_2, \omega, \text{Siter})$ .

2.2) Compute

$$\text{RelRes} = \text{mean} \left\{ \frac{\|\mathbf{g}_l^h - \mathcal{N}_l^h([\bar{\mathbf{u}}^h]^{[K+1]})\|}{\|\mathbf{g}_l^h - \mathcal{N}_l^h([\bar{\mathbf{u}}^h]^{[0]})\|} \mid l = 1, \dots, 2 \right\}.$$

2.3) Set  $K = K + 1$ .

**until** a stopping rule for NMG method is met.

---

### The MG smoother

Following literature<sup>5,11,18,23–27</sup> the local Fourier analysis can be used to guarantee that there exists an efficient point-wise smoother within a MG method for solving the discrete nonlinear system (25). To obtain a high-potential point-wise smoother, this work proposes a coupled outer-inner iteration method in a FP framework.

Let  $\nu$  denote the index for the outer step. We start the outer iteration by introducing the iterative scheme as follows

$$\begin{cases} -\theta_1 \Delta u_1^{[\nu+1]} + f_1(\mathbf{u}^{[\nu+1]}) = g_1, \\ -\theta_2 \Delta u_2^{[\nu+1]} + f_2(\mathbf{u}^{[\nu+1]}) = g_2. \end{cases}$$

Since this iterative scheme is fully implicit, a linearization procedure of the nonlinear term  $f_l(\mathbf{u}^{[\nu+1]})$  is required. As pointed out in literature<sup>5,11,23,26–28</sup> it is appropriate to apply a global linearization technique with the nonlinear terms  $f_l(\mathbf{u}^{[\nu+1]})$ . Thus, the resulting approximation is given by

$$\begin{aligned} f_l(\mathbf{u}^{[\nu+1]}) &= f_l(u_1^{[\nu+1]}, u_2^{[\nu+1]}) \\ &\approx f_l(u_1^{[\nu]}, u_2^{[\nu]}) + \sigma_{l1}^{[\nu]} \delta u_1^{[\nu]} + \sigma_{l2}^{[\nu]} \delta u_2^{[\nu]} \end{aligned} \quad (27)$$

where

$$\begin{aligned} \sigma_{l1}(\mathbf{u}^{[\nu]}) &= \partial_{u_1} f_l(u_1^{[\nu]}, u_2^{[\nu]}) \\ &= (\partial_{u_1} T(\mathbf{u}^{[\nu]})) (\partial_{u_1} T(\mathbf{u}^{[\nu]})) \\ &+ (T(\mathbf{u}^{[\nu]}) - R) (\partial_{u_1 u_1} T(\mathbf{u}^{[\nu]})), \\ \sigma_{l2}(\mathbf{u}^{[\nu]}) &= \partial_{u_2} f_l(u_1^{[\nu]}, u_2^{[\nu]}) \\ &= (\partial_{u_2} T(\mathbf{u}^{[\nu]})) (\partial_{u_2} T(\mathbf{u}^{[\nu]})) \\ &+ (T(\mathbf{u}^{[\nu]}) - R) (\partial_{u_2 u_2} T(\mathbf{u}^{[\nu]})) \end{aligned}$$

To obtain a simple and fast iterative scheme, we use the approximations for  $\sigma_{l1}$  and  $\sigma_{l2}$  as introduced by literature.<sup>11,34–36</sup> Therefore, we have

$$\begin{aligned} \sigma_{11}(\mathbf{u}^{[\nu]}) &= (\partial_{u_1} T_{\mathbf{u}^{[\nu]}}) (\partial_{u_1} T_{\mathbf{u}^{[\nu]}}) \\ &+ (T_{\mathbf{u}^{[\nu]}} - R) (\partial_{u_1 u_1} T_{\mathbf{u}^{[\nu]}}) \\ &\approx (\partial_{u_1} T_{\mathbf{u}^{[\nu]}}) (\partial_{u_1} T_{\mathbf{u}^{[\nu]}}), \end{aligned}$$

$$\begin{aligned} \sigma_{22}(\mathbf{u}^{[\nu]}) &= (\partial_{u_2} T_{\mathbf{u}^{[\nu]}}) (\partial_{u_2} T_{\mathbf{u}^{[\nu]}}) \\ &+ (T_{\mathbf{u}^{[\nu]}} - R) (\partial_{u_2 u_2} T_{\mathbf{u}^{[\nu]}}) \\ &\approx (\partial_{u_2} T_{\mathbf{u}^{[\nu]}}) (\partial_{u_2} T_{\mathbf{u}^{[\nu]}}), \\ \sigma_{12}(\mathbf{u}^{[\nu]}) &= \sigma_{21}(\mathbf{u}^{[\nu]}) = 0. \end{aligned}$$

These approximations leads to the following linearized system

$$\begin{cases} -\theta_1 \Delta u_1^{[\nu+1]} + \sigma_{11}(\mathbf{u}^{[\nu]}) u_1^{[\nu+1]} = \\ g_1 - f_1(\mathbf{u}^{[\nu]}) + \sigma_{11}(\mathbf{u}^{[\nu]}) u_1^{[\nu]}, \\ -\theta_2 \Delta u_2^{[\nu+1]} + \sigma_{22}(\mathbf{u}^{[\nu]}) u_2^{[\nu+1]} = \\ g_2 - f_2(\mathbf{u}^{[\nu]}) + \sigma_{22}(\mathbf{u}^{[\nu]}) u_2^{[\nu]} \end{cases} \quad (28)$$

Next, we apply the finite difference discretization as discussed in Section 3.1 with (28) and solve the associated linear system for each outer step  $\nu$  by the Gauss-Seidel (GS) method as the inner iteration. The  $k$ th step of the GS method defined at a grid point  $(x_1, x_2)$  is then given by

$$(\mathbf{u}^{[\nu+1]})_{ij}^{[k+1]} = (\mathbf{v})_{ij}^{[k+1]} \quad (29)$$

where

$$\begin{aligned} (\mathbf{v})_{ij}^{[k+1]} &= \begin{pmatrix} F_1^{[\nu]} \\ F_2^{[\nu]} \end{pmatrix}, \\ F_1^{[\nu]} &= \left( \frac{1}{(\theta_1/h_1^2)(\bar{\Sigma})_{ij} + (\sigma_{11}(\mathbf{u}^{[\nu]}))_{ij}} \right) \\ &\times \left[ (g_1)_{ij} - f_1(\mathbf{u}^{[\nu]})_{ij} + (\sigma_{11}(\mathbf{u}^{[\nu]}))_{ij} (u_1^{[\nu]})_{ij} \right. \\ &\left. + (\theta_1/h_1^2)(\bar{\Sigma})_{ij} (u_1^{[\nu+1]})_{ij}^{[k+1/2]} \right] \\ F_2^{[\nu]} &= \left( \frac{1}{(\theta_2/h_2^2)(\bar{\Sigma})_{ij} + (\sigma_{22}(\mathbf{u}^{[\nu]}))_{ij}} \right) \times \\ &= \left[ (g_2)_{ij} - f_2(\mathbf{u}^{[\nu]})_{ij} + (\sigma_{22}(\mathbf{u}^{[\nu]}))_{ij} (u_2^{[\nu]})_{ij} \right. \\ &\left. + (\theta_2/h_2^2)(\bar{\Sigma})_{ij} (u_2^{[\nu+1]})_{ij}^{[k+1/2]} \right] \end{aligned}$$

and

$$(\bar{\Sigma})_{ij} (u_i^{[\nu+1]})_{ij}^{[k+1/2]} = (u_i^{[\nu+1]})_{i+1,j}^{[k]} + (u_i^{[\nu+1]})_{i-1,j}^{[k+1]}$$

$$\begin{aligned}
& + \gamma^2 (u_l^{[\nu+1]})_{i,j+1}^{[k]} \\
& + \gamma^2 (u_l^{[\nu+1]})_{i,j-1}^{[k+1]}
\end{aligned}$$

Here the superscripts  $k$ ,  $k + 1/2$ , and  $k + 1$  denote the current, intermediate and new approximations computed by the GS method.

In order to obtain more efficiency, one introduces a relaxation parameter  $\omega \in (0, 2)$  and iterates the successive over relaxation (SOR) steps by

$$(u^{[\nu+1]})_{i,j}^{[k+1]} = (1 - \omega)(u^{[\nu+1]})_{i,j}^{[k]} + \omega(v)_{i,j}^{[k+1]} \quad (30)$$

Finally, the implementation of the proposed MG smoother (30) on  $\Omega_h$  is summarized in Algorithm 3.

**Algorithm 3** The MG smoother for the proposed NMG algorithm (Algorithm 2)

$$\bar{u}^h \leftarrow \text{Smoother}(\bar{u}^h, g_1^h, g_2^h, R^h, T^h, \theta_1, \theta_2, \omega, \text{Siter})$$

- 1)] Use input parameters to compute  $(v)_{i,j}$  for all  $1 \leq i \leq n_1$  and  $1 \leq j \leq n_2$ .
- 2)] Perform SOR steps for  $k = 0, 1, 2, \dots, \text{Siter}$ 
  - 2.1)] Update  $(\bar{u}^h)_{i,j}^{[k+1]}$  by (30) for all  $1 \leq i \leq n_1$  and  $1 \leq j \leq n_2$ .

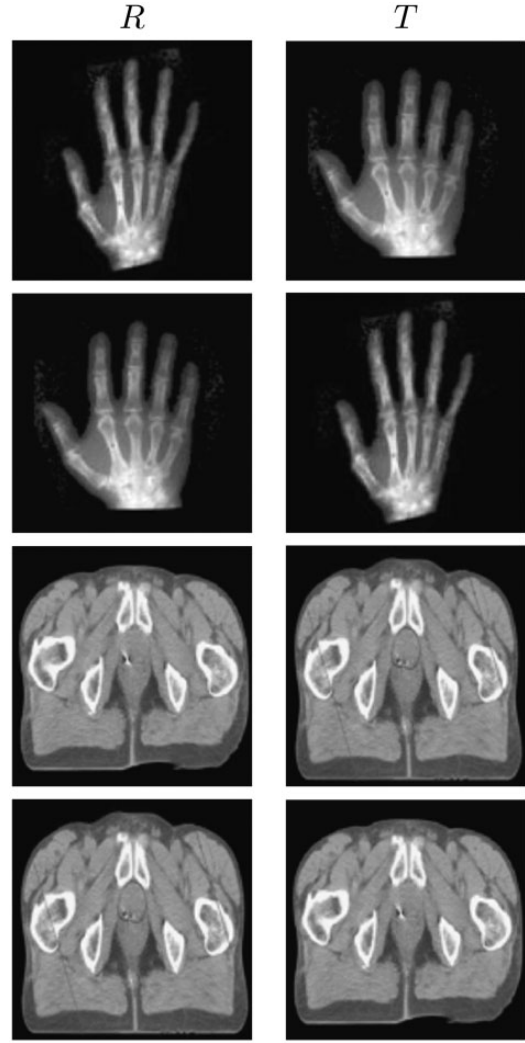
Note that we applied the so-called *local Fourier analysis* (LFA) to test the smoothing performance of the proposed MG smoother with different values of  $\omega$  (the relaxation parameter) and *Siter* (the maximum number of inner iterations) for the four representative registration problems given by Problems 1–4 (as shown respectively in Figure 1). Our results shows that the proposed FP smoother with  $\omega = 1.85$  and *Siter* = 2 is recommended; see its performance in leading the proposed ALM method to fast convergence from Tables 3 and 4. The theoretical details for applying LFA to analyze the proposed smoother are straightforward and similar to our previous work in (11)

**The closed form expressions for  $w_1$ ,  $w_2$ ,  $\lambda_1$ , and  $\lambda_2$**

Based on the formulation (23), we can get the closed form expressions for  $w_1$  and  $w_2$  as given by

$$(w_l)_{i,j} = \max\left(\left|(S_l)_{i,j}\right| - \frac{\alpha}{\theta_l}, 0\right) \frac{(S_l)_{i,j}}{\left|(S_l)_{i,j}\right|}, \quad (31)$$

where  $S_l = \left(\delta_{x_1}^+(u_l)_{i,j} - \frac{(\lambda_{l1})_{i,j}}{\theta_l}, \delta_{x_2}^+(u_l)_{i,j} - \frac{(\lambda_{l2})_{i,j}}{\theta_l}\right)^\top$  for  $l = 1, 2$ .



**Figure 1.** Four registration problems of the real medical images. (left column) Reference images; (right column) Template images; (top row) Problem 1; (2nd row) Problem 2; (3rd row) Problem 3; (bottom row) Problem 4.

Similarly, based on the formulations in (11) and (12), we may update all the Lagrangian multipliers by

$$(\lambda_1^{(k+1)})_{i,j} = (\lambda_1^{(k)})_{i,j} + \theta_1((w_1)_{i,j}^{(k+1)} - \nabla^+(u_1)_{i,j}^{(k+1)}) \quad (32)$$

$$(\lambda_2^{(k+1)})_{i,j} = (\lambda_2^{(k)})_{i,j} + \theta_2((w_2)_{i,j}^{(k+1)} - \nabla^+(u_2)_{i,j}^{(k+1)}) \quad (33)$$

## Numerical experiments

In this section, we present a number of numerical experiments to

- compare the overall performance of the proposed ALM with two related numerical solutions, which



are the LGD method (5) and the NMG method by Chumchob (5)

- assess the accuracy and efficiency of the proposed ALM with regard to parameter changes.

We note first that four registration problems consisting of four real medical images to be denoted as Problems 1–4 were selected for the experiments, as shown respectively in Figure 1. Second, we used  $n_1 = n_2 = n$  (i.e. we assume the grid spacing  $h_1 = h_2 = h = 1/n$ ) in all numerical tests. Third, the bilinear interpolation was employed to compute the transformed template image  $T_u$  once the deformations are found in all cases. We also note that all numerical algorithms for the proposed ALM, LGD and NMG methods were started with  $\mathbf{u}^{[0]} = 0$  and implemented under MATLAB R2018a and run on a machine configured with Intel(R) Core(TM) i7 Quad-core 4.2 GHz and 32GB of RAM.

### Performance comparison with the other two methods

In this test, the performance of three different methods for TV-based image registration model is compared. We apply the relative SSD (a qualitative measure in the accuracy), time per iteration (in seconds), total CPU time (in seconds), and total iterations to perform our evaluation on the four registration problems shown in Figure 1.

For the LGD method, we chose  $\tau = 10^{-5}$  and  $\beta = 10^{-6}$  and implemented a direct solver using MATLAB's \operator operator (also known as mldivide operator) for solving each linear system of equations arising from applying the FP method with the nonlinear discrete EL system.

For the NMG method, we implemented with the FAS-NMG method of Chumchob (5) The details of our numerical implementation can be summarized as follows. We solved the nonlinear discrete EL system with  $\beta = 10^{-6}$  in a V-cycle framework, where the coarsest grid is  $4 \times 4$ . The LGD method was used to solve the coarsest problem with  $\tau = 10^{-3}$ . We used the standard coarsening method and standard interpolation technique for transferring data between grids. We applied 15 pre- and 15 post-smoothing steps with a so-called FP-SOR smoother.

For the proposed ALM, we chose the penalty parameters  $\theta_1 = \theta_2 = 1/10$ . We used the predefined numbers  $\epsilon_1 = 10^{-2}$ ,  $\epsilon_2 = 10^{-4}$ ,  $\epsilon_3 = \epsilon_4 = 10^{-8}$  and  $\epsilon_5 = 200$  with Algorithm 1. The predefined numbers  $\epsilon_1 = 20$  and  $\epsilon_2 = 10^{-6}$  were applied by the proposed NMG method in Algorithm 2 to solve  $\mathbf{u}$ -subproblem. In all registration problems, the proposed NMG method with the MG smoother in Algorithm 3 was performed with the MG parameters,  $\nu_1 = \nu_2 = 5$  (5 pre- and 5

post-smoothing steps),  $\omega = 1.85$  (the relaxation parameter), and Siter = 2 (2 iterations of SOR steps).

All numerical methods in this test used the same regularization parameter  $\alpha = 1/50$  with the grid spacing  $h = 1/256$ . Each method was stopped when the

relative solution,  $\max\left\{\frac{\|u_1^{(m)} - u_1^{(m-1)}\|}{\|u_1^{(m)}\|}, \frac{\|u_2^{(m)} - u_2^{(m-1)}\|}{\|u_2^{(m)}\|}\right\}$ , was

brought below  $10^{-4}$  or the number of iterations increases beyond a predefined number. Here we used the predefined numbers  $m_{\max}^{\text{LGD}} = 200$  for the LGD method,  $m_{\max}^{\text{NMG}} = 50$  for the NMG method, and  $m_{\max}^{\text{ALM}} = 200$  for the proposed ALM.

From Table 1, one can see that only the proposed ALM converges, whereas the LGD and NMG methods are unable to converge in a reasonable number of iterations. As expected, we have found in this case that the small value of  $\beta$  has a significant effect on the accuracy of the registered images and the convergence of the LGD and NMG method. Moreover, Figure 2 shows that for each registration problem the proposed ALM yields the best value of the relative SSD. It is important to note that the smaller the value of the relative SSD is, the accurate registered image is achieved. This evidence ensures that the registration results by the proposed ALM are more reliable than the LGD and NMG methods. Particularly, Figure 3 shows that the proposed ALM is computationally efficient than the other two methods in delivering the high quality of the registered images. The proposed ALM takes only a few iterations (say 5 iterations) to drop the relative SSD below 0.20, which means that the dissimilarities between the reference and registered images have been reduced more than 80% within the first 5 iterations. This is a remarkable result to conclude that the computational performance of the proposed ALM in solving TV-based image registration is much more efficient than those of the other two methods.

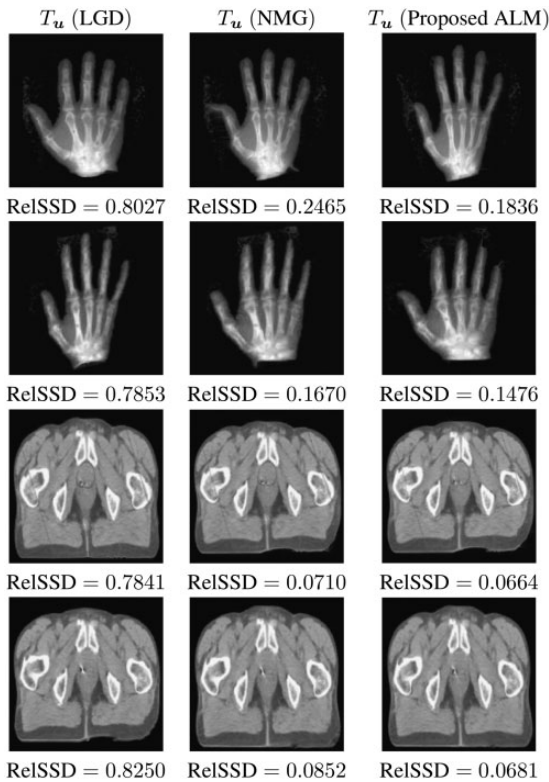
In terms of computation times per iteration, we can see that the proposed ALM requires to solve two sub-problems for  $\mathbf{u}$  and  $\mathbf{w}$  in each step. Thus, in accordance with our measurements, we expect that more computation time per iteration is required for the proposed ALM than those of the other two methods. As can be seen from Table 1, our numerical experiments reflect this expectation.

In terms of total CPU times, Table 1 illustrates the LGD method is the slowest method, while the NMG method is slightly faster than the proposed ALM. However, both LGD and NMG method are unable to converge. Thus we can conclude that the LGD and NMG methods are less computationally efficient than the proposed ALM.

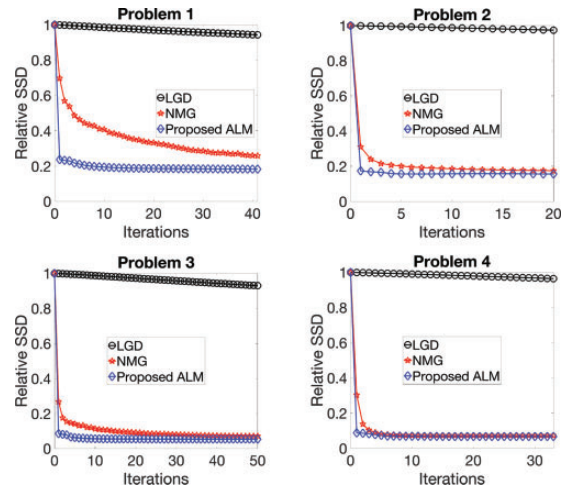
In Figure 4 we present the constructed deformations by the proposed ALM for all registration problems.

**Table 1.** Comparison of the relative SSD, time per iteration, total CPU time (s), and total iterations by three different numerical methods on the four registration problems in Figure 1 with  $\alpha = \frac{1}{50}$  and  $h = \frac{1}{256}$ . \* indicates that maximum number of iterations reached without convergence.

Method	Registration problem			
	1	2	3	4
<b>LGD</b>				
Relative SSD	0.8027	0.7853	0.7841	0.8250
Time per iteration (s)	0.2467	0.2458	0.2457	0.2463
Total CPU time (s)	49.3423	49.1558	49.1301	49.2524
Total iterations	200*	200*	200*	200*
<b>NMG</b>				
Relative SSD	0.2465	0.1670	0.0710	0.0852
Time per iteration (s)	0.4941	0.4930	0.4814	0.4914
Total CPU time (s)	24.7045	24.6477	24.0701	24.5724
Total iterations	50*	50*	50*	50*
<b>Proposed ALM</b>				
Relative SSD	0.1836	0.1476	0.0664	0.0681
Time per iteration (s)	0.5970	0.6893	0.7105	0.7138
Total CPU time (s)	28.4776	40.6674	37.6567	38.5466
Total iterations	41	59	53	54

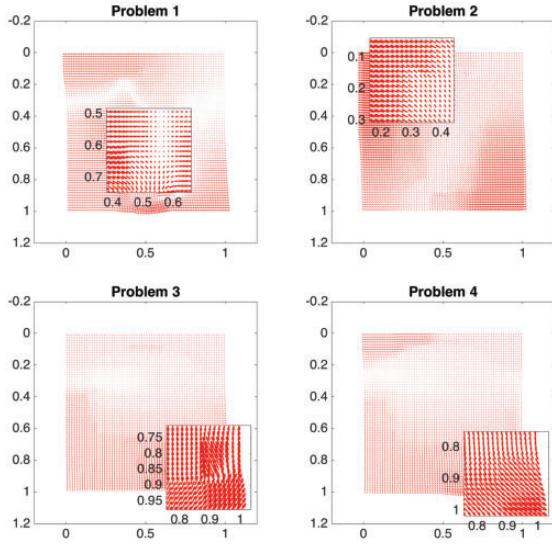


**Figure 2.** Qualitative comparison of registered images by three different methods on the four registration problems in Figure 1. (left) Registered images by LGD; (middle) registered images by NMG; (right) registered images by the proposed ALM. 'RelSSD' means the relative SSD defined in (13) to represent the dissimilarities between  $R$  and  $T_u$ . Note that the small the value of RelSSD is, the accurate registered image is obtained.



**Figure 3.** History of the relative SSD by three different methods in solving the four registration problems in Figure 1.

In Figure 5, we show the constructed surfaces of both components of the deformations in Figure 4. In Figure 6, we present the middle slices of the constructed surfaces of the components of the deformations in Figure 5. As expected from the use of TV regularizer, the visual inspection in Figure 4 shows that the proposed ALM delivers the visually appealing results in preserving the discontinuities of the constructed deformation as shown in the corresponding close-up regions. We can also see in Figure 4 that the constructed deformation yields multiple motions in the image to be registered and the motion discontinuities can be observed at the boundaries of the local



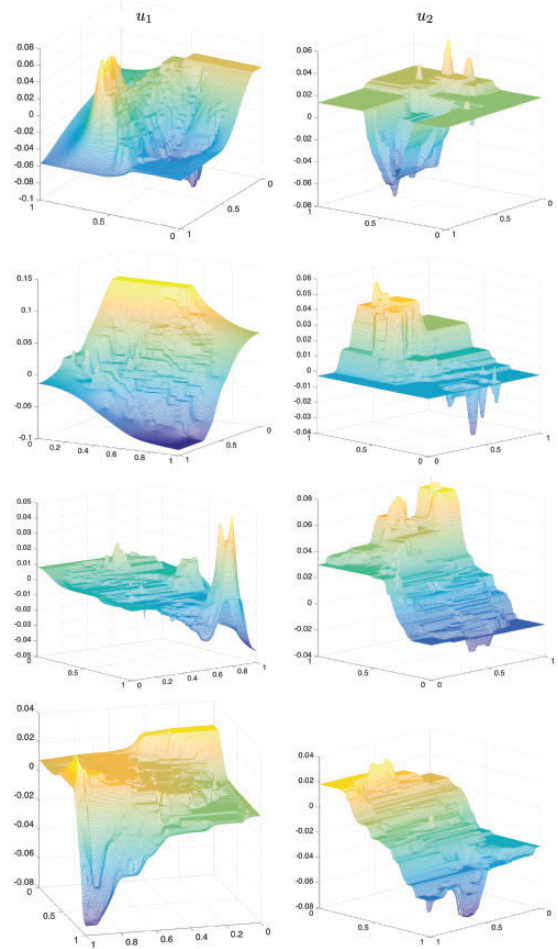
**Figure 4.** Constructed deformations by the proposed ALM for the four registration problems in Figure 1. Note that multiple motions in the corresponding image and motion discontinuities at the boundaries of the local regions determined by the constructed deformation can be observed in the corresponding close-up regions.

regions. We see further that the corresponding deformation in each registration problem is neither smooth nor reflecting a homogeneous motion. For each local region, the motion tends to change smoothly, and the gradients of the constructed deformation in this area are small. At the boundaries of the local regions, the gradients of the constructed deformation are large, and the discontinuities of the constructed deformation can be observed. Moreover, we can see from Figures 5 and 6 that the constructed surfaces of both components of the deformations are non-smooth and their middle slices are almost piecewise constant for all registration problems.

To summarize we have successfully developed the efficient and effective numerical method for TV-based image registration model. Our registration results on the real medical applications shown in Figure 1 demonstrate that the proposed ALM is more computationally efficient and effective than the other two methods. The most attractive features of our proposed method is that it is able to produce the high quality of the registered images in a reasonable number of iterations while it satisfies the requirement in constructing the deformations by the TV regularizer.

### Performance tests with the regard to parameter changes

We now present numerical results from several test cases, to assess the accuracy and efficiency of our proposed numerical techniques with the regard to parameter changes.



**Figure 5.** Constructed surfaces of both components of the constructed deformations by the proposed ALM for the four registration problems in Figure 1. (left) the first component  $u_1$ ; (right) the second component  $u_2$ ; (top row) Problem 1; (2nd row) Problem 2; (3rd row) Problem 3; (bottom row) Problem 4.

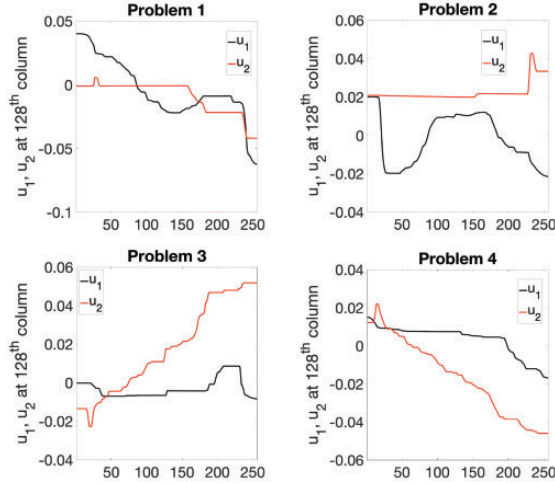
*h-Independence test.* One of the key properties of MG techniques is that their convergence does not depend on the number of grid points. Thus, in this test we designed our numerical experiments to investigate this property with the proposed ALM in Algorithm 1, and to back up our proposed NMG method in Algorithms 2.

We implemented the proposed ALM with the same regularization parameter  $\alpha = \frac{1}{250}$  and the penalty parameters  $\theta_1$  and  $\theta_2$  as shown in Table 2. In order to stop our proposed ALM, we used the same predefined numbers used in Section 4.1. We also used the same MG parameters to solve  $\mathbf{u}$ -subproblem as given in the previous section.

In the numerical results shown in Table 2, one can see five quantities: the relative SSD, the total iterations, the total CPU times (in seconds), the ratio of the total CPU times in increasing both image dimensions by a

factor of 2, and the number of average MG steps used to solve  $\mathbf{u}$ -subproblem by Algorithm 2 with different values of grid spacing  $h$ .

As expected from a numerical technique using a MG framework, Table 2 shows that Algorithm 1 not only



**Figure 6.** The middle slices of the corresponding surfaces of both components of the constructed deformations shown in Figure 5 by the proposed ALM.

converges within a few iterations, but it also provides the accurate registration results because the dissimilarities between the reference and registered images given have been reduced more than 90% for Problem 1, 89% for Problem 2, and 96% for Problems 3 and 4. Moreover, the NMG method in Algorithm 2 can reduce the mean of the relative residuals to  $\varepsilon_2 = 10^{-6}$  within a few MG steps for different values of  $h$ . In addition, we see an increase in the total CPU time by a factor of approximately 4.5 when both image dimensions is increased by a factor of 2. This shows that the proposed ALM has an optimal efficiency resulting from the NMG method proposed in Algorithm 2, which should be of order  $\mathcal{O}(N \log(N))$  where  $N = n^2$ . Then we can conclude that the proposed ALM has no effect on the convergence for different values of  $h$ .

**$\alpha$ -Dependence test.** Next we evaluate to show how our proposed ALM in Algorithm 1 is affected with varying the regularization parameter  $\alpha$ .

To this end, we performed the proposed ALM on the four registration problems in Figure 1 with the same grid spacing  $h = \frac{1}{256}$  and the penalty parameters  $\theta_1$  and  $\theta_2$  as given in Table 2. Our proposed ALM in

**Table 2.** Registration results for Problems 1–4 in Figure 1 by the proposed ALM in Algorithm 1 with  $\alpha = \frac{1}{250}$  and different values of grid spacing  $h$ . Note that ‘Ratio’ means the ratio of the total CPU times in increasing both image dimensions by a factor of 2.

Registration problem	Grid spacing ( $h$ )			
	$\frac{1}{256}$	$\frac{1}{512}$	$\frac{1}{1024}$	$\frac{1}{2048}$
1 ( $\theta_1 = \theta_2 = 1$ )				
Relative SSD	0.0968	0.0958	0.0951	0.0951
Total iterations	22	22	21	21
Total CPU time (s)	22.4489	74.1501	308.4895	1,308.3476
Ratio	–	3.3031	4.1603	4.2411
Average MG cycles	6	6	6	6
2 ( $\theta_1 = \theta_2 = 1$ )				
Relative SSD	0.1028	0.1027	0.1025	0.1002
Total iterations	16	16	15	15
Total CPU time (s)	16.1900	58.8500	257.0531	1,117.0938
Ratio	–	3.6350	4.3679	4.4294
Average MG cycles	7	7	7	7
3 ( $\theta_1 = \theta_2 = \frac{1}{4}$ )				
Relative SSD	0.0359	0.0359	0.0358	0.0357
Total iterations	36	34	33	33
Total CPU time (s)	38.8623	137.9971	577.3338	2,578.2640
Ratio	–	3.5509	4.1837	4.4658
Average MG cycles	7	7	7	7
4 ( $\theta_1 = \theta_2 = \frac{1}{4}$ )				
Relative SSD	0.0377	0.0375	0.0374	0.0373
Total iterations	31	31	30	30
Total CPU time (s)	30.9928	116.8559	500.9036	2,238.7849
Ratio	–	3.7704	4.2865	4.4695
Average MG cycles	6	6	6	6

**Table 3.** Registration results for Problems 1–4 in Figure 1 by the proposed ALM in Algorithm 1 with  $h = \frac{1}{256}$  and different values of the regularization parameter  $\alpha$ .

Registration problem	Regularization parameter ( $\alpha$ )					
	$\frac{1}{250}$	$\frac{1}{200}$	$\frac{1}{150}$	$\frac{1}{100}$	$\frac{1}{50}$	$\frac{1}{25}$
1 ( $\theta_1 = \theta_2 = 1$ )						
Relative SSD	0.0968	0.1015	0.1080	0.1313	0.2099	0.3466
Total iterations	20	21	25	30	33	49
Average MG cycles	6	6	6	6	6	6
2 ( $\theta_1 = \theta_2 = 1$ )						
Relative SSD	0.1028	0.1028	0.1069	0.1174	0.1675	0.2104
Total iterations	16	18	20	27	37	40
Average MG cycles	7	6	6	6	6	6
3 ( $\theta_1 = \theta_2 = \frac{1}{4}$ )						
Relative SSD	0.0359	0.0385	0.0436	0.0536	0.0958	0.2251
Total iterations	36	44	49	65	90	125
Average MG cycles	7	7	6	6	5	5
4 ( $\theta_1 = \theta_2 = \frac{1}{4}$ )						
Relative SSD	0.0377	0.0406	0.0449	0.0548	0.0863	0.1909
Total iterations	31	34	49	57	102	139
Average MG cycles	6	6	5	5	5	5

**Table 4.** Registration results for Problems 1–4 in Figure 1 by the proposed ALM in Algorithm 1 with  $h = \frac{1}{256}$ ,  $\alpha = \frac{1}{250}$  and different values of the penalty parameter  $\theta$ . Recall that \* indicates that maximum number of MG cycles reached without convergence.

Penalty parameter	Registration problem			
	1	2	3	4
$\theta = 10$				
Relative SSD	0.8128	0.8128	0.4923	0.5056
Total iterations	3	3	3	3
Average MG cycles	20*	15	18	18
$\theta = 8$				
Relative SSD	0.7721	0.7774	0.4271	0.4401
Total iterations	3	3	3	3
Average MG cycles	20*	19	15	15
$\theta = 6$				
Relative SSD	0.7098	0.7138	0.3476	0.3595
Total iterations	3	3	3	3
Average MG cycles	20*	20*	14	14
$\theta = 4$				
Relative SSD	0.6033	0.6029	0.2615	0.2710
Total iterations	3	3	3	3
Average MG cycles	20*	20*	19	19
$\theta = 2$				
Relative SSD	0.1935	0.2048	0.2187	0.2467
Total iterations	8	7	10	9
Average MG cycles	12	14	20*	20*
$\theta = 1$				
Relative SSD	0.0968	0.1028	0.2057	0.2443
Total iterations	20	16	17	19
Average MG cycles	6	7	20*	20*
$\theta = \frac{1}{2}$				
Relative SSD	0.0693	0.0696	0.0670	0.0441
Total iterations	39	31	25	19
Average MG cycles	5	5	20*	20*

(continued)

Table 4. Continued.

Penalty parameter	Registration problem			
	1	2	3	4
$\theta = \frac{1}{4}$				
Relative SSD	0.0617	0.0586	0.0359	0.0377
Total iterations	60	44	36	31
Average MG cycles	4	5	7	6
$\theta = \frac{1}{6}$				
Relative SSD	0.0600	0.0549	0.0338	0.0348
Total iterations	75	60	46	52
Average MG cycles	4	4	5	5
$\theta = \frac{1}{8}$				
Relative SSD	0.0584	0.0536	0.0328	0.0340
Total iterations	89	100	63	59
Average MG cycles	4	4	5	4
$\theta = \frac{1}{10}$				
Relative SSD	0.0579	0.0535	0.0326	0.0336
Total iterations	108	191	75	62
Average MG cycles	4	4	4	4

Algorithm 1 was stopped with the same predefined numbers used in Section 4.1. We also applied the same predefined number and MG parameters as shown in Section 4.1 with Algorithms 2 and 3 to solve  $\mathbf{u}$ -subproblem.

Table 3 presents the numerical results by the proposed ALM in Algorithm 1 with different values of the regularization parameter  $\alpha$ . It contains three quantities: the relative SSD, the total iterations, and the number of average MG steps used to solve  $\mathbf{u}$ -subproblem by Algorithm 2.

As presented in Table 3, decreasing the values of  $\alpha$  has significant effects on the accuracy of the registered images and the convergence of the proposed ALM in Algorithm 1, whereas the value of  $\alpha$  has only a small effect on the convergence of the proposed NMG method in Algorithm 2. We can see that large  $\alpha$  is not needed as small ones give better registration results, typically  $\alpha = \frac{1}{250}, \frac{1}{200}, \frac{1}{150},$  and  $\frac{1}{100}$ . It is important to note that the process to select the optimal value of  $\alpha$  is a separate but important issue because it is in general unknown a priori and it significantly affects on the accuracy of registered images and the ALM performance.

In order to find a suitable  $\alpha$  automatically, the ‘cooling’ (‘continuation’) process suggested in literature<sup>11,35,37–39</sup> is recommended for real applications. The basic idea is to start with a high initial value of  $\alpha$  and then slowly reduce  $\alpha$  such that the obtained solution can be used to be an excellent starting point for the next in order to decrease  $\mathcal{J}$  in (8).

( $\theta_1, \theta_2$ )-dependence test. We now present numerical results from several test cases to evaluate the

registration performance of our proposed ALM with different values of  $\theta_1$  and  $\theta_2$ .

We performed the proposed ALM in Algorithm 1 on the four registration problems shown in Figure 1 using  $h = \frac{1}{256}$  and  $\alpha = \frac{1}{250}$ . The predefined numbers and the MG parameters for Algorithms 1–3 are the same values used in Section 4.1.

We note that we take  $\theta_1 = \theta_2 = \theta$  in this numerical test to evaluate the performance of the proposed ALM. Table 4 presents the registration results for the four registration problems by the proposed ALM in Algorithm 1 with different values of  $\theta$ . Three qualities for the performance tests are the relative SSD, the total iterations, and the number of average MG steps used to solve  $\mathbf{u}$ -subproblem by Algorithm 2.

One sees from Table 4 that as  $\theta$  is decreased from 10 to  $\frac{1}{10}$  the dissimilarities between the given images decrease (the accuracy of the registered image increase), whereas the cost of the proposed ALM increases. We observe that the smaller  $\theta$  is the better the proposed NMG method performs and an increase in the number of smoothing steps is required to achieve fast MG convergence for large  $\theta$ . Next, we see that decreasing the value of  $\theta$  leads to the best registration result at  $\theta = \frac{1}{10}$ . We see further that large values of  $\theta$  are not required as small ones deliver better registration performance. Typically,  $\theta$  should be between  $\frac{1}{10}$  and  $\frac{1}{2}$  to give the accurate registration results in a reasonable number of iterations.

## Concluding remarks

In this paper, we first explained how standard methods solve the TV-based image registration model. Next, we

discussed how we are needed to develop a new method. In order to efficiently solve the model, we therefore proposed to use augmented Lagrangian method. We separated the associated minimization problem into two subproblems. As a result, the first subproblem is a nonlinear problem and impossible to obtain exact solution, whereas the other one has a closed-form solution. Next, we developed an efficient NMG method to solve the associated discrete nonlinear system. In order to assess the efficiency and effectiveness of our new method, we tested, using four registration problems of real medical images, how our new method performs. We found in our first numerical test that the registered images by different methods are not identical. Therefore, it can be concluded that different numerical algorithms for TV-based image registration model have a significant effect on the accuracy of registered images. We also found that our new method outperforms the existing methods. Moreover we found by the performance comparison in this test that our new method is able to deliver the accurate registration results with the desired properties of the constructed deformations in a reasonable number of iterations. Next, we found in the second test that there are no effect on the convergence of the new method for different numbers of grid points. Moreover, we observed from the third and last tests that the choice of the regularization and penalty parameters is important for the quality of the registered images and the computational performance of the new method. The outlines in selecting these two parameters to obtain the accurate registration results were discussed. Future work will extend the proposed method to high-order variational models for image registration.

### Acknowledgements

The authors would like to thank the anonymous reviewers for a number of useful suggestions to improve the quality of the paper


### Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

### Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The first author work was partially supported by the Thailand Research Fund Grant #MRG6080169.

### ORCID iD

Noppadol Chumchob  <https://orcid.org/0000-0002-4892-1314>

### References

1. Maintz J and Viergever M. A survey of medical image registration. *Med Image Anal* 1998; 2: 1–36.
2. Hajnal J, Hill D and Hawkes D. *Medical image registration*. The Biomedical Engineering Series. Boca Raton, FL: CRC Press, 2001.
3. Mani V and Arivazhagan S. Survey of medical image registration. *J Biomed Eng Technol* 2013; 1: 8–25.
4. Sotiras A, Davatzikos C and Paragios N. Deformable medical image registration: a survey. *IEEE Trans Med Imaging* 2013; 32: 1153–1190.
5. Chumchob N. Vectorial total variation-based regularization for variational image registration. *IEEE Trans Image Process* 2013; 22: 4551–4559.
6. F-Schauf C, Henn S, Hömke L, et al. 2006. Total variation based image registration. In: Tai X-C, Lie K-A, Chan TF, et al. (eds) *Proceedings of the international conference on PDE-Based image processing and related inverse problems series: Mathematics and visualization*. Berlin: Springer Verlag, pp.305–323.
7. F-Schauf C, Henn S and Witsch K. Multigrid based total variation image registration. *Comput Visual Sci* 2008; 11: 101–113.
8. Broit C. *Optimal registration of deformed images*. PhD thesis, University of Pennsylvania, USA, 1981.
9. Fischer B and Modersitzki J. Fast diffusion registration. *Contemp Math* 2002; 313: 117–129.
10. Fischer B and Modersitzki J. Curvature based image registration. *J Math Imag Vision* 2003; 18: 81–85.
11. Chumchob N and Chen K. A robust multigrid approach for variational image registration models. *J Comput Appl Math* 2011; 236: 653–674.
12. Acar R and Vogel C. Analysis of bounded variation penalty method. *Inverse Probl* 1994; 10: 1217–1229.
13. Chan T and Chen K. On a nonlinear multigrid algorithm with primal relaxation for the image total variation minimization. *Numer Algor* 2006; 41: 387–411.
14. Rudin L, Osher S and Fatemi E. Nonlinear total variation based noise removal algorithms. *Physica D* 1992; 60: 259–268.
15. Chan T and Mulet P. On the convergence of the lagged diffusivity fixed point method in total variation image restoration. *SIAM J Numer Anal* 1999; 36: 354–367.
16. Vogel C. *Computational methods for inverse problems*. Philadelphia: SIAM Publications, 2002.
17. Vogel C and Oman M. Iterative methods for total variation denoising. *SIAM J Sci Comput* 1996; 17: 227–238.
18. Chumchob N and Chen K. A variational approach for discontinuity-preserving image registration. *East-West J Math Special* 2010; 2010: 266–282.
19. Goldstein T and Osher S. The split bregman method for 11-regularized problems. *SIAM J Imag Sci* 2009; 2: 323–343.
20. Wang Y, Yang J, Yin W, et al. A new alternating minimization algorithm for total variation image reconstruction. *SIAM J Imaging Sci* 2008; 1: 248–272.
21. Wenqi L, Jinming D, Zhaowen Q, et al. Implementation of high-order variational models made easy for image processing. *Math Methods Appl Sci* 2016; 39: 4208–4233.

22. Brandt A. Multi-level adaptive solutions to BVPs. *Math Comp* 1977; 31: 333–390.
23. Chumchob N, Chen K and Brito C. A fourth order variational image registration model and its fast multigrid algorithm. *Multiscale Model Simul* 2011; 9: 89–128.
24. Chumchob N and Chen K. An improved variational image registration model and a fast algorithm for its numerical approximation. *Numer Meth Partial Differential Eq* 2012; 28: 1966–1995.
25. Chumchob N, Chen K and Brito C. A new variational model for removal of combined additive and multiplicative noise and a fast algorithm for its numerical approximation. *Int J Comput Math* 2013; 90: 140–161.
26. Jewprasert S, Chumchob N and Chantrapornchai C. A fourth-order compact finite difference scheme for higher-order PDE-based image registration. *East Asian J Appl Math* 2016; 5: 361–386.
27. Chumchob N, Jewprasert S and Chantrapornchai C. Multigrid solution of the nonlinear PDEs arising in elastic image registration with application to a group of monomodal images. *ScienceAsia* 2016; 42: 415–422.
28. Thompson T and Chen K. An effective diffeomorphic model and its fast multigrid algorithm for registration of lung CT images. *Comput Meth Appl Math* 2020; 20: 141–168.
29. Briggs W, Henson V and McCormick S. *A multigrid tutorial*. 2nd ed. Philadelphia, USA: SIAM Publications, 2000.
30. Hackbusch W. *Multi-grid methods and applications*. Berlin, Heidelberg, New York Springer-Verlag, 1985.
31. Trottenberg U, Oosterlee C and Schuller A. *Multigrid*. Cambridge: Academic Press, 2001.
32. Wesseling P. *Multigrid methods*. Philadelphia, PA Edwards, 2004.
33. Wienands R and Joppich W. *Practical Fourier analysis for multigrid method*. USA: Chapman & Hall/CRC, 2005.
34. Modersitzki J. *Numerical methods for image registration*. Oxford, 2004.
35. Haber E and Modersitzki J. A multilevel method for image registration. *SIAM J Sci Comput* 2006; 27: 1594–1607.
36. Köstler H, Ruhnau K and Wienands R. Multigrid solution of the optical flow system using a combined diffusion- and curvature-based regularizer. *Numer Linear Algebra Appl* 2008; 15: 201–218.
37. Henn S and Witsch K. Image registration based on multi-scale energy information. *Multiscale Model Simul* 2005; 4: 584–609.
38. Lelièvre P and Oldenburg D. Magnetic forward modeling and inversion for high susceptibility. *Geophys J Inter* 2006; 166: 76–90.
39. Newman GA and Hoversten GM. Solution strategies for two- and three-dimensional electromagnetic inverse problems. *Inverse Problems* 2000; 16: 1357–1375.5