Journal of Computational and Applied Mathematics 236 (2011) 653–674

Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A robust multigrid approach for variational image registration models

Noppadol Chumchob [a,b], Ke Chen [a,*]

[a] Centre for Mathematical Imaging Techniques, Department of Mathematical Sciences, The University of Liverpool, Peach Street, Liverpool L69 7ZL, United Kingdom

[b] Department of Mathematics, Faculty of Science, Silpakorn University, Nakorn Pathom 73000, Thailand

ARTICLE INFO

MSC:
65F10
65M55
68U10

Keywords:
Variational models
Deformable registration
Diffusion and curvature image models
Smoothing analysis
Nonlinear multigrid
Inverse problems

ABSTRACT

Variational registration models are non-rigid and deformable imaging techniques for accurate registration of two images. As with other models for inverse problems using the Tikhonov regularization, they must have a suitably chosen regularization term as well as a data fitting term. One distinct feature of registration models is that their fitting term is always highly nonlinear and this nonlinearity restricts the class of numerical methods that are applicable. This paper first reviews the current state-of-the-art numerical methods for such models and observes that the nonlinear fitting term is mostly 'avoided' in developing fast multigrid methods. It then proposes a unified approach for designing fixed point type smoothers for multigrid methods. The diffusion registration model (second-order equations) and a curvature model (fourth-order equations) are used to illustrate our robust methodology. Analysis of the proposed smoothers and comparisons to other methods are given. As expected of a multigrid method, being many orders of magnitude faster than the unilevel gradient descent approach, the proposed numerical approach delivers fast and accurate results for a range of synthetic and real test images.

1. Introduction

Image registration is one of the most useful and yet challenging tasks in image processing applications. It is the process of finding an *optimal geometric transformation* between *corresponding* images. It can also be seen as the process of overlaying two or more images of the same or similar scene taken at different times, from different perspectives, and/or by different imaging machineries. Therefore, this procedure is required whenever a series of corresponding images needs to be compared or integrated. Applications that require a registration step range from art, astronomy, biology, chemistry, criminology, physics and remote sensing. Particularly, in medical applications, non-invasive imaging is increasingly used in almost all stages of patient care: from disease detection to treatment guidance and monitoring. For an overview on registration methodology, we refer to [1–3], and the references therein. This paper focuses on a robust multigrid approach for deformable image registration models in a variational formulation.

Variational partial differential equations (PDEs) based image registration models have been successfully proven to be very valuable tools in several applications, although much improvement is still required. A general framework of the image registration can be formulated as follows: given two images of the same object, respectively referred to as *reference R* and *template T*, we search for a vector-valued transformation φ defined by

$$\varphi(\mathbf{u})(\cdot) : \mathbb{R}^d \to \mathbb{R}^d, \qquad \varphi(\mathbf{u})(\mathbf{x}) : \mathbf{x} \mapsto \mathbf{x} + \mathbf{u}(\mathbf{x})$$

* Corresponding author.
E-mail addresses: cnoppado@liv.ac.uk, cnoppado@su.ac.th (N. Chumchob), k.chen@liv.ac.uk, k.chen@liverpool.ac.uk (K. Chen).
URL: http://www.liv.ac.uk/~cmchenke/cmit/ (K. Chen).

0377-0427/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2011.06.026

that depends on the unknown *deformation* or *displacement field*

$$\mathbf{u} : \mathbb{R}^d \to \mathbb{R}^d, \qquad \mathbf{u} : \mathbf{x} \mapsto \mathbf{u}(\mathbf{x}) = (u_1(\mathbf{x}), u_2(\mathbf{x}), \ldots, u_d(\mathbf{x}))^\top$$

such that the transformed template $T \circ \varphi(\mathbf{u}(\mathbf{x})) = T(\mathbf{x} + \mathbf{u}(\mathbf{x})) = T(\mathbf{u})$ becomes similar to the reference R. Once the corresponding location $\varphi(\mathbf{u}(\mathbf{x})) = \mathbf{x} + \mathbf{u}(\mathbf{x})$ is calculated for each spatial location $\mathbf{x}$ in the image domain $\Omega \subset \mathbb{R}^d$, an image interpolation is required to assign the image intensity values for the transformed template $T(\mathbf{u})$ at non-grid locations within image boundaries. For locations outside the image boundaries, the image intensities are usually set to be a constant value, typically zero [3]. We see that the displacement $\mathbf{u}$ measures by how much a point in the transformed template $T(\mathbf{u})$ has moved away from its original position in T. Here we shall restrict ourselves to scalar or gray intensity images and model them as compactly supported functions mapping from the image domain $\Omega \subset \mathbb{R}^d$ into $V \subset \mathbb{R}_0^+$, where $d \in \mathbb{N}$ represents the spatial dimension of the images which is usually $d = 2$ (planar images) or $d = 3$ (volume data set) with boundary $\partial\Omega$. Without loss of generality we assume that the registration problem is described in the two-dimensional case ($d = 2$) throughout this work, but it is readily extendable to the three-dimensional case ($d = 3$). We also assume further that $\Omega = [0, 1]^2 \subset \mathbb{R}^2$ and $V = [0, 1]$ for 2D gray intensity images.

Assume the image intensities of R and T are comparable (i.e. in a monomodal registration scenario), the task is to solve the minimization problem of a similarity measure

$$\min_{\mathbf{u}} \left\{ \mathcal{D}(\mathbf{u}) = \frac{1}{2} \int_\Omega (T(\mathbf{x} + \mathbf{u}(\mathbf{x})) - R(\mathbf{x}))^2 \mathrm{d}\mathbf{x} \right\}, \tag{1}$$

whose functional is the sum of squared differences (SSD) [4–20,3,21,22] which is widely used. As is known, this problem is generally ill posed in the sense of Hadamard. Therefore, the minimization of $\mathcal{D}$ will not guarantee an unique solution. It becomes necessary to impose a constraint on the solution $\mathbf{u}$ via a deformation regularizer $\mathcal{R}$ for penalizing unwanted and irregular solutions using some priori knowledge. In the Tikhonov regularization framework, the image registration problem can be posed as a minimization problem of the *joint* energy functional given by

$$\min_{\mathbf{u}} \{ \mathcal{J}_\alpha[\mathbf{u}] = \mathcal{D}(\mathbf{u}) + \alpha \mathcal{R}(\mathbf{u}) \}, \tag{2}$$

where $\alpha > 0$ is the regularization parameter that compromises similarity and regularity, and $T(\mathbf{u}) = T(\mathbf{x}+\mathbf{u}(\mathbf{x}))$ represents the transformed template image. The regularizer $\mathcal{R}$ is designed to ensure that the deformation field $\mathbf{u}$ constructed is unique and close to the true solution. As the aim of this paper is to address the fast solutions issue, we consider two regularizers: firstly the *diffusion regularizer* as introduced in [4]:

$$\mathcal{R}(\mathbf{u}) = \frac{1}{2} \sum_{l=1}^{2} \int_\Omega |\nabla u_l|^2 \mathrm{d}\mathbf{x}, \tag{3}$$

and secondly the *curvature regularizer* in [5] as

$$\mathcal{R}^{\mathrm{Cv}}(\mathbf{u}) = \frac{1}{2} \sum_{l=1}^{2} \int_\Omega (\widehat{\kappa}_M(u_l))^2 \mathrm{d}\mathbf{x} = \frac{1}{2} \sum_{l=1}^{2} \int_\Omega (\Delta u_l)^2 \mathrm{d}\mathbf{x}, \tag{4}$$

as respective examples of second-order PDEs and fourth-order PDEs; our method will also be applicable to other models that lead to second-order PDEs e.g. the elastic and fluid models [3] and the total variation model [8,9], the combined regularization model [23] and other curvature models [24].

Although the multigrid techniques have been successfully used for numerical solutions for deformable image registration [9,10,12,14,16–18,21,22], none of the existing variants are optimal implementations. In these works, the nonlinear fitting term is not directly dealt with; we remark that other image restoration models [25–28] do not usually have nonlinearities resulting from a fitting term (unless a L_1 fitting or an expectation–maximization fitting is used). We may classify these existing variants into 2 categories:

(1) Linear multigrid framework. Haber and Modersitzki [12] used an inexact Gauss–Newton (GN) scheme combined with a linear multigrid method as an inner solver for solving the elastic image registration problem corresponding to (2). Henn [16] considered the curvature image registration problem and similarly used a coupled outer–inner iteration method with the inner solver provided by a linear multigrid method for solving the system of the fourth-order PDEs. Hömke [17] concentrated on the elastic image registration and introduced a numerical solution of the minimization problem given by (2) using a regularized GN method with a trust region approach in which one normal equation corresponding to a linear subproblem is solved iteratively with a linear multigrid method. Köstler et al. [18] introduced a combined diffusion- and curvature-based regularizer for optical flow and deformable image registration problems and solved the resulting minimization problem represented in terms of (2) with an inexact GN method where each GN perturbation is estimated by a linear multigrid approach. Stürmer et al. [21] considered the diffusion image registration and solved the system of nonlinear PDEs using a coupled outer–inner iteration method with the inner solver provided by a linear multigrid method commonly used for heat equations.

(2) Nonlinear multigrid framework. The use of the nonlinear multigrid (NMG) methods can be found in works of [9,10,14,22]. In particular, Frohn-Schauf et al. [9] considered the minimization problem of the data term $\mathcal{D}$ given by (1) only and then the total variation (TV) regularization at a GN step; further they solved the resulting nonlinear system by the full approximation scheme NMG (FAS-NMG) method due to Brandt [29] with an augmentation method and a line relaxation smoother. Gao et al. [10] used the FMG method for solving the system of nonlinear PDEs related to variational approach for the diffusion image registration, where they used the Newton–Gauss–Seidel smoother (i.e. global linearization by Newton's iteration and Gauss–Seidel (GS) iteration for the resulting linear systems [20,30]) and an adaptive procedure for determining large deformations. Finally Henn and Witsch [14] solved the elastic image registration problem using a nonlinear Jacobi smoother plus a line search procedure on the regularization parameter α, and Zikic et al. [22] used the full multigrid (FMG) method for a diffusion image registration problem and solved the system of nonlinear parabolic PDEs with a fixed point (FP) type of smoothers within the semi-implicit time-marching approach.

We also remark that the 2D optical flow formulation (that does not use SSD) suitable for registering closely related images (e.g. video sequences) can be solved by multigrid techniques [31,32] using time-marching smoothers. In this paper, our aim is to propose a different but robust smoother for the nonlinear multigrid framework that works for a range of models.

The rest of the paper is organized as follows. In Section 2, we consider our first model problem of diffusion registration, surveying and discussing its numerical treatments. A new fixed point smoother is proposed and analyzed in Section 3 for the FAS-NMG approach for the underlying nonlinear Euler–Lagrange system. In Section 4, we consider our first model problem of curvature registration and demonstrate how to use our proposed method. Experimental results from medical test images are illustrated in Section 5 in order to show the excellent performance of the proposed numerical scheme compared with other methods with conclusions summarized in Section 6.

2. The diffusion registration model and its numerical methods

We now introduce our first model and review briefly various solution methods paying particular attention to robustness of multigrid methods. The model itself is not particularly more important than other models from [3] but we use it to illustrate the fast solution issues of image registration. Below use the notation $\partial_{x_\ell} F = \frac{\partial F}{\partial x_\ell}$ and $\partial_{x_1 x_2} F = \frac{\partial^2 F}{\partial x_1 \partial x_2}$.

2.1. The diffusion model

The minimizer of the energy functional $\mathcal{J}_\alpha$ in (2), defined by (1) and (3), with respect to the minimizer $\mathbf{u} = (u_1, u_2)^\top$ satisfies the so-called *Euler–Lagrange* equations [3], given by the following system of two coupled, nonlinear, and elliptic partial differential equations (PDEs) for $\mathbf{u} = (u_1(x_1, x_2), u_2(x_1, x_2))^\top$:

$$\begin{cases} \mathcal{N}_1(\mathbf{u}) = -\alpha \Delta u_1 + \overbrace{(T(\mathbf{u}) - R)\partial_{u_1} T(\mathbf{u})}^{f_1(\mathbf{u})} = 0, \\ \mathcal{N}_2(\mathbf{u}) = -\alpha \Delta u_2 + \overbrace{(T(\mathbf{u}) - R)\partial_{u_2} T(\mathbf{u})}^{f_2(\mathbf{u})} = 0, \end{cases} \tag{5}$$

subject to the homogeneous Neumann's boundary conditions

$$\partial_{\boldsymbol{n}} u_1 = \partial_{\boldsymbol{n}} u_2 = 0 \quad \text{on } \partial\Omega. \tag{6}$$

Here the nonlinear functions $f_1(\mathbf{u}), f_2(\mathbf{u})$ are from the data fitting terms, which are (as remarked) features of registration models distinct from other imaging models [26]. Refer to [9,4,10,14,3,21,22]. In fact, the nonlinear coupling of the two PDEs is through the term $T(\mathbf{u})$. Hereby, Δ denotes the Laplace operator, and $\boldsymbol{n} = (n_1, n_2)^\top$ is the outward unit vector normal to the image boundary $\partial\Omega$. Note that the first and second terms in (5) are the first variations of the regularizer term $\mathcal{R}$ and the data term $\mathcal{D}$, respectively.

2.2. Discretization by a finite difference method

For sake of clarity, let $(u_l^h)_{i,j} = u_l^h(x_{1_i}, x_{2_j})$ denote the grid function for $l = 1, 2$ with grid spacing $h = (h_1, h_2) = (1/n_1, 1/n_2)$. Applying finite difference schemes based on the cell-centered grid points to discretize (5), the discrete Euler–Lagrange equations at a grid point (i, j) over the discrete domain,

$$\Omega_h = \{\mathbf{x} \in \Omega \,|\, \mathbf{x} = (x_{1_i}, x_{2_j})^\top = ((2i-1)h_1/2, (2j-1)h_2/2)^\top,\ 1 \le i \le n_1,\ 1 \le j \le n_2\} \tag{7}$$

are given by

$$\begin{cases} \mathcal{N}_1^h(\mathbf{u}^h)_{i,j} = -\alpha \mathcal{L}^h(u_1^h)_{i,j} + f_1^h(u_1^h, u_2^h)_{i,j} = g_{1_{i,j}}^h \\ \mathcal{N}_2^h(\mathbf{u}^h)_{i,j} = -\alpha \mathcal{L}^h(u_2^h)_{i,j} + f_2^h(u_1^h, u_2^h)_{i,j} = g_{2_{i,j}}^h \end{cases} \tag{8}$$

with the following notation

$$
\begin{aligned}
&-\mathcal{L}^h(u_l^h)_{i,j} = \frac{1}{h_1^2}((\Sigma)_{i,j}(u_l^h)_{i,j} - (\overline{\Sigma})_{i,j}(u_l^h)_{i,j}),\\
&(\Sigma)_{i,j} = 2(1+\gamma^2), \quad \gamma = h_1/h_2,\\
&(\overline{\Sigma})_{i,j}(u_l^h)_{i,j} = (u_l^h)_{i+1,j} + (u_l^h)_{i-1,j} + \gamma^2(u_l^h)_{i,j+1} + \gamma^2(u_l^h)_{i,j-1}\\
&f_1^h(u_1^h, u_2^h)_{i,j} = (T_{i,j}^{h^*} - R_{i,j}^h)((T_{i+1,j}^{h^*} - T_{i-1,j}^{h^*})/(2h_1)),\\
&f_2^h(u_1^h, u_2^h)_{i,j} = (T_{i,j}^{h^*} - R_{i,j}^h)((T_{i,j+1}^{h^*} - T_{i,j-1}^{h^*})/(2h_2)),\\
&T_{i,j}^{h^*} = T^h(i + (u_1^h)_{i,j}, j + (u_2^h)_{i,j}),\\
&(\mathbf{u}^h)_{i,j} = ((u_1^h)_{i,j}, (u_2^h)_{i,j})^\top.
\end{aligned}
$$

Here $g_{1_{i,j}}^h = g_{2_{i,j}}^h = 0$ on the finest grid in multigrid setting to be used shortly. We note that the approximations in (8) need to be adjusted at the image boundary $\partial\Omega_h$ using the homogeneous Neumann boundary conditions

$$
(u_l^h)_{i,1} = (u_l^h)_{i,2}, \qquad (u_l^h)_{i,n_2} = (u_l^h)_{i,n_2-1}, \qquad (u_l^h)_{1,j} = (u_l^h)_{2,j}, \qquad (u_l^h)_{n_1,j} = (u_l^h)_{n_1-1,j}. \tag{9}
$$

2.3. Review of non-multigrid numerical solvers

The **first** commonly used method is a gradient descent approach solving, instead of the nonlinear elliptic system (5), the nonlinear parabolic system

$$
\begin{cases}
\partial_t u_1 - \alpha\Delta u_1 = -(T(\mathbf{u}) - R)\partial_{u_1}T(\mathbf{u})\\
\partial_t u_2 - \alpha\Delta u_2 = -(T(\mathbf{u}) - R)\partial_{u_2}T(\mathbf{u}),
\end{cases} \tag{10}
$$

where $\mathbf{u} = \mathbf{u}(\mathbf{x}, t) = (u_1(\mathbf{x}, t), u_2(\mathbf{x}, t))^\top$ will converge to the solution of (5) when $t \to \infty$, with the initial solution $\mathbf{u}(\mathbf{x}, 0)$, typically $\mathbf{u}(\mathbf{x}, 0) = 0$. The advantage is that various time-marching schemes can be used to solve (10) in order to circumvent the nonlinearity on the right-hand side. For example, the semi-implicit scheme can be proceeded as follows (in matrix vector form obtained from the discretized version of Eq. (10))

$$
\begin{cases}
u_1^{(k+1)} = \left(\mathbf{I} - \alpha\tau\sum_{l=1}^{2}\mathbf{A}_l\right)^{-1}(u_1^{(k)} - \tau f_1(u_1^{(k)}, u_2^{(k)}))\\
u_2^{(k+1)} = \left(\mathbf{I} - \alpha\tau\sum_{l=1}^{2}\mathbf{A}_l\right)^{-1}(u_2^{(k)} - \tau f_2(u_1^{(k)}, u_2^{(k)})).
\end{cases} \tag{11}
$$

Here, $\mathbf{I}$ is the identity matrix, $f_l(u_1^k, u_2^k)$ is the discretized version of the second term in (5) using an appropriate finite difference approximation, usually the central finite difference, $\tau > 0$ is the time-step determined by a forward difference approximation of the time derivative $\partial_t u_l$, and $\mathbf{A}_l$ is the coefficient matrix from discretization of the Laplace operator Δ along the l-coordinate direction subject to Neumann boundary conditions. The DCT-based method from [3] or the FT-based method from [33,34] may be used to solve the above equation.

The **second** method by an additive operator splitting (AOS) scheme introduced in [4,35,36] is faster and more efficient than the standard semi-implicit scheme (11). The basic idea is to replace the inverse of the sum by a sum of inverses. The corresponding iterations are then defined by

$$
\begin{cases}
u_1^{(k+1)} = \frac{1}{2}\sum_{l=1}^{2}(\mathbf{I} - 2\alpha\tau\mathbf{A}_l)^{-1}(u_1^{(k)} - \tau f_1(u_1^{(k)}, u_2^{(k)}))\\
u_2^{(k+1)} = \frac{1}{2}\sum_{l=1}^{2}(\mathbf{I} - 2\alpha\tau\mathbf{A}_l)^{-1}(u_2^{(k)} - \tau f_2(u_1^{(k)}, u_2^{(k)})),
\end{cases} \tag{12}
$$

which is much cheaper than those obtained from (11) because the two tri-diagonal systems in each component are solved per iteration rather than the 5-band system.

The **third** method by the one-cycle multi-resolution (or a coarse-to-fine FMG technique) is proposed in [20]. The idea is to solve (5) first with the Newton–Gauss–Seidel relaxation method given by

$$
\begin{cases}
u_1^{(new)} = u_1^{(old)} - \dfrac{\mathcal{N}_1(\mathbf{u}^{(old)})}{\partial_{u_1}\mathcal{N}_1(\mathbf{u}^{(old)})}\\
u_2^{(new)} = u_2^{(old)} - \dfrac{\mathcal{N}_2(\mathbf{u}^{(old)})}{\partial_{u_2}\mathcal{N}_2(\mathbf{u}^{(old)})}
\end{cases} \tag{13}
$$

on the coarsest (lowest) resolution, and then interpolate the coarse solution with an appropriate method as a good initial guess solution to the next finer resolution. This process is repeated until (5) has been solved by (13) on the finest (highest) or original resolution. Although, this method provides well matched images in most of the cases and the basic idea is essentially the same as a FMG method as in [37,38,30,39,40], one should note that convergence cannot be proved even for much simpler equations [30]. This is due to the fact that the solution on the fine level depends strongly on the coarse one. As a consequence, errors introducing from the interpolation procedure may propagate in such a way that they spoil the overall results completely.

Although the above three methods are easy to implement on a unilevel and multilevel, they are not as efficient as a multigrid method.

2.4. Review of multigrid solvers and previous work

Multigrid techniques are widely used fast methods for various PDEs [37,38,41,30,39,40]. For deformable image registration models, we give a brief review here [9,10,12,14,16–18,21,22].

The basic idea of a multigrid method is to smooth high-frequency components of the error of the solution by performing a few steps with a smoother such that a smooth error term can be well represented and approximated on a coarser grid. After a linear or nonlinear residual equation has been solved on the coarse grid, a coarse-grid correction is interpolated back to the fine grid and used to correct the fine-grid approximation. Finally, the smoother is performed again in order to remove some new high-frequency components of the error introduced by the interpolation.

Below we shall first review the basic FAS-NMG algorithm before discussing previous work on solving (5). As it turns out, two approaches considered are both more efficient than an unilevel method but none are robust solvers.

2.4.1. The full approximation scheme (FAS)

For a nonlinear problem, the use of a full approximation scheme (the nonlinear multigrid method (NMG) in [29]) is natural. The FAS technique has been tried in various image processing applications; see e.g. [42,43,32,44,27,28,45,9,46,47]. We now denote our PDEs (5) by

$$\begin{cases} \mathcal{N}_1^h(\mathbf{u}^h) = g_1^h, \\ \mathcal{N}_2^h(\mathbf{u}^h) = g_2^h, \end{cases} \tag{14}$$

separating the linear operator $\mathcal{L}^h$ from the nonlinear operator f_l^h ($l = 1, 2$) on a general fine grid with step size $h = (h_1, h_2)$ on Ω_h.

Let $\boldsymbol{v}^h = (v_1^h, v_2^h)^\top$ be the result computed by performing a few steps with a smoother (**pre-smoothing step**) on the fine-grid problem. Then, the algebraic error $\mathbf{e}^h$ of the solution is given by $\mathbf{e}^h = \mathbf{u}^h - \boldsymbol{v}^h$. The residual equation system is given by

$$\begin{cases} \mathcal{N}_1^h(\boldsymbol{v}^h + \mathbf{e}^h) - \mathcal{N}_1^h(\boldsymbol{v}^h) = g_1^h - \mathcal{N}_1^h(\boldsymbol{v}^h) = r_1^h \\ \mathcal{N}_2^h(\boldsymbol{v}^h + \mathbf{e}^h) - \mathcal{N}_2^h(\boldsymbol{v}^h) = g_2^h - \mathcal{N}_2^h(\boldsymbol{v}^h) = r_2^h. \end{cases} \tag{15}$$

In order to correct the approximated solution $\boldsymbol{v}^h$ on the fine grid, one needs to compute the error $\mathbf{e}^h$. However, the error cannot be computed directly. Since high-frequency components of the error in pre-smoothing step have already been removed by the smoother, we can transfer the following nonlinear system to the coarse grid as follows

$$\begin{cases} \underbrace{\mathcal{N}_1^h(\boldsymbol{v}^h + \mathbf{e}^h)}_{\mathcal{N}_1^h(\mathbf{u}^h)} = \underbrace{r_1^h + \mathcal{N}_1^h(\boldsymbol{v}^h)}_{g_1^h} \\ \underbrace{\mathcal{N}_2^h(\boldsymbol{v}^h + \mathbf{e}^h)}_{\mathcal{N}_2^h(\mathbf{u}^h)} = \underbrace{r_2^h + \mathcal{N}_2^h(\boldsymbol{v}^h)}_{g_2^h} \end{cases} \rightarrow \begin{cases} \underbrace{\mathcal{N}_1^H(\boldsymbol{v}^H + \mathbf{e}^H)}_{\mathcal{N}_1^H(\mathbf{u}^H)} = \underbrace{r_1^H + \mathcal{N}_1^H(\boldsymbol{v}^H)}_{g_1^H} \\ \underbrace{\mathcal{N}_2^H(\boldsymbol{v}^H + \mathbf{e}^H)}_{\mathcal{N}_2^H(\mathbf{u}^H)} = \underbrace{r_2^H + \mathcal{N}_2^H(\boldsymbol{v}^H)}_{g_2^H} \end{cases} \tag{16}$$

where $H = 2h$ is the new cell size $H_1 \times H_2$ with $H_1 \geq h_1$ and $H_2 \geq h_2$ and $g_l^H \neq 0$ on the coarse grid. After the nonlinear residual equation system on the coarse grid (16) have been solved with a method of our choice, the coarse-grid correction $\mathbf{e}^H = \mathbf{u}^H - \boldsymbol{v}^H$ is then interpolated back to the fine grid $\mathbf{e}^h$ that can now be used for updating the approximated solution $\boldsymbol{v}^h$ on the fine grid, i.e. $\boldsymbol{v}_{new}^h = \boldsymbol{v}^h + \mathbf{e}^h$. The last step for a FAS multigrid is to perform the smoother again to remove high-frequency parts of the interpolated error (**post-smoothing step**).

In our FAS multigrid for diffusion image registration, standard coarsening is used in computing the coarse-grid domain Ω_H by doubling the grid size in each space direction, i.e. $h \rightarrow 2h = H$. For intergrid transfer operators between Ω_h and Ω_H, the averaging and bilinear interpolation techniques are used for the restriction and interpolation operators denoted respectively by I_h^H and I_H^h; see the details in [37,38,30,39,40]. In order to compute the coarse-grid operator of $\mathcal{N}_l^h(\mathbf{u}^h)$ consisting of two parts: $\mathcal{L}^h(u_l^h)$ and $f_l^h(u_1^h, u_2^h)$, a so-called *discretization coarse-grid approximation* (DCA) is performed [37,31,30,40]. The idea is to rediscretize the Euler–Lagrange directly. In the case of $\mathcal{L}^h(u_l^h)$, the corresponding coarse-grid

part $\mathcal{L}^H(u_l^H)$ is obtained by the restriction of u_l^h and a simple adaptation of the grid size to the discretized Laplacian. For $f_l^h(u_1^h, u_2^h)$, we first use the restriction operator with both components of the deformation field $\mathbf{u}^h$, u_1^h and u_2^h, and the given images, R^h and T^h, and then compute the corresponding coarse-grid part $f_l^H(u_1^H, u_2^H)$. To solve (14) numerically, our FAS multigrid is applied recursively down to the coarsest grid consisting of a small number of grid points, typically 4×4, and may be summarized as follows:

Algorithm 1 (*FAS Multigrid Algorithm*)**.** Denote FAS multigrid parameters as follows:

ν_1 pre-smoothing steps on each level

ν_2 post-smoothing steps on each level

l the number of multigrid cycles on each level ($l = 1$ for V-cycling and $l = 2$ for W-cycling). Here we present the V-cycle with $l = 1$.

α regularization parameter

ω relaxation parameter

GSiter the maximum number of iterations using a smoother

$[v_1^h, v_2^h] \leftarrow FASCYC(v_1^h, v_2^h, \mathcal{N}_1^h, \mathcal{N}_2^h, g_1^h, g_2^h, R^h, T^h, \nu_1, \nu_2, \alpha, \omega, GSiter)$.

- If $\Omega_h = coarset\ grid$ ($|\Omega_h| = 4 \times 4$), solve (14) using time-marching techniques such as semi-implicit or AOS scheme (Section 2.3) and then stop. Else continue with following step.
- Pre-smoothing:
  For $z = 1$ to ν_1, $[v_1^h, v_2^h] \leftarrow Smoother(v_1^h, v_2^h, g_1^h, g_2^h, R^h, T^h, \alpha, \omega, GSiter)$.
- Restriction to the coarse grid:
  $v_1^H \leftarrow I_h^H v_1^h,\ v_2^H \leftarrow I_h^H v_2^h,\ R^H \leftarrow I_h^H R^h,\ T^H \leftarrow I_h^H T^h$.
- Set the initial solution for the coarse-grid problem:
  $[\overline{v}_1^H, \overline{v}_2^H] \leftarrow [v_1^H, v_2^H]$.
- Compute the new right-hand side for the coarse-grid problem:
  $g_1^H \leftarrow I_h^H(g_1^h - \mathcal{N}_1^h(v_1^h, v_2^h)) + \mathcal{N}_1^H(v_1^H, v_2^H)$,
  $g_2^H \leftarrow I_h^H(g_2^h - \mathcal{N}_2^h(v_1^h, v_2^h)) + \mathcal{N}_2^H(v_1^H, v_2^H)$.
- Implement the FAS multigrid on the coarse-grid problem:
  $[v_1^H, v_2^H] \leftarrow FASCYC(v_1^H, v_2^H, \mathcal{N}_1^H, \mathcal{N}_2^H, g_1^H, g_2^H, R^H, T^H, \nu_1, \nu_2, \alpha, \omega, GSiter)$.
- Add the coarse-grid corrections:
  $v_1^h \leftarrow v_1^h + I_H^h(v_1^H - \overline{v}_1^H),\ v_2^h \leftarrow v_2^h + I_H^h(v_2^H - \overline{v}_2^H)$.
- Post-smoothing:
  For $z = 1$ to ν_2, $[v_1^h, v_2^h] \leftarrow Smoother(v_1^h, v_2^h, g_1^h, g_2^h, R^h, T^h, \alpha, \omega, GSiter)$.

For practical applications our FAS multigrid method is stopped if the maximum number of V- or W-cycles ε_1 is reached (usually $\varepsilon_1 = 20$), the mean of the relative residuals obtained from the Euler–Lagrange equations is smaller than a small number $\varepsilon_2 > 0$ (typically $\varepsilon_2 = 10^{-8}$ for a convergence test and only $\varepsilon_2 = 10^{-1}$ for a practical registration), the relative reduction of the dissimilarity is smaller than some $\varepsilon_3 > 0$ (we usually assign $\varepsilon_3 = 0.10$ meaning that the relative reduction of the dissimilarity would decrease about 90%), or the change in two consecutive steps of the data/fitting term $\mathcal{D}$ is smaller than a small number $\varepsilon_4 > 0$ (typically $\varepsilon_4 = 10^{-8}$). A pseudo-code implementation of our FAS multigrid method is then summarized in the following algorithm:

Algorithm 2 (*FAS Multigrid Method*)**.**

$\boldsymbol{v}^h \leftarrow FASMG(\boldsymbol{v}^h, \alpha, \overrightarrow{\varepsilon})$.

- Select α, $\overrightarrow{\varepsilon} = (\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4)$ and initial guess solutions $\boldsymbol{v}_{initial}^h = (v_1^h, v_2^h)^\top$ on the finest grid.
- Set $K = 0$, $(\boldsymbol{v}^h)^K = \boldsymbol{v}_{initial}^h$, $\widetilde{\varepsilon}_2 = \varepsilon_2 + 1$, $\widetilde{\varepsilon}_3 = \varepsilon_3 + 1$, and $\widetilde{\varepsilon}_4 = \varepsilon_4 + 1$.
- While ($K < \varepsilon_1$ AND $\widetilde{\varepsilon}_2 > \varepsilon_2$ AND $\widetilde{\varepsilon}_3 > \varepsilon_3$ AND $\widetilde{\varepsilon}_4 > \varepsilon_4$)
  - $(\boldsymbol{v}^h)^{K+1} = [v_1^h, v_2^h] \leftarrow$ $FASCYC(v_1^h, v_2^h, \mathcal{N}_1^h, \mathcal{N}_2^h, g_1^h, g_2^h, R^h, T^h, \nu_1, \nu_2, \alpha, \omega, GSiter)$
  - $\widetilde{\varepsilon}_2 = mean\{\frac{\|g_l^h - \mathcal{N}_l^h((\boldsymbol{v}^h)^{K+1})\|_{l_2}}{\|g_l^h - \mathcal{N}_l^h(\boldsymbol{v}_{initial}^h)\|_{l_2}} | l = 1, 2\}$
  - $\widetilde{\varepsilon}_3 = \frac{\overline{\mathcal{D}^h(R^h, T^h((\boldsymbol{v}^h)^{K+1}))}}{\mathcal{D}^h(R^h, T^h)}$, [Recall that $\mathcal{D}^h(R^h, T^h(\cdot)) \sim \frac{h_1 h_2}{2}\|R^h, T^h(\cdot)\|_{l_2}^2$]
  - $\widetilde{\varepsilon}_4 = |\mathcal{D}^h(R^h, T^h((\boldsymbol{v}^h)^{K+1})) - \mathcal{D}^h(R^h, T^h((\boldsymbol{v}^h)^K))|$
  - $K = K + 1$
- end

As is well known, in addition to restriction and interpolation operators, the above Algorithm 1 requires a suitable *smoother* based on some iterative relaxation method which is often the decisive factor in determining whether or not a multigrid algorithm converges. This issue is discussed next after we review the linear multigrid method.

2.4.2. The linear multigrid method for (5)

For a nonlinear problem, a linearization approach can lead to a coupled outer–inner iteration method with the inner solver provided by a linear multigrid method. For (5), the outer iteration is introduced either in a GN iteration [12,17,18] or in terms of the semi-implicit time-marching scheme [21] as follows

$$\begin{cases} \left(\mathbf{I} - \alpha\tau \sum_{l=1}^{2} \mathbf{A}_l\right) u_1^{(k+1)} = (u_1^{(k)} - \tau f_1(u_1^{(k)}, u_2^{(k)})) \\ \left(\mathbf{I} - \alpha\tau \sum_{l=1}^{2} \mathbf{A}_l\right) u_2^{(k+1)} = (u_2^{(k)} - \tau f_2(u_1^{(k)}, u_2^{(k)})), \end{cases} \tag{17}$$

which is a system of two linear elliptic PDEs. Finally each outer step k is solved iteratively by a linear multigrid. In order to reduce the number of outer steps, a scale space framework described in [15] is used for adapting automatically the registration parameters τ and α. Although, this numerical scheme is very accurate for providing visually pleasing registration results, we found experimentally that it is quite slow in fulfilling the necessary condition for being a minimizer of the variational problem represented by (2), i.e. in achieving convergence, because the linear system has to be solved many times with changing the right-hand side of (17). This is a convenient way of using a multigrid method but is not as optimal as a nonlinear multigrid method.

2.4.3. The nonlinear multigrid method for (5)

The above introduced FAS-NMG Algorithm 2 can be readily applied to (5). The choice of a suitable smoother is a key. Below we briefly review four types of smoothers that have been or can be used for diffusion image registration:

(1) The Newton–Gauss–Seidel relaxation smoother. This was used in [10] for (5). Although, there are no numerical results to demonstrate the convergence of the FMG technique in their work, we found with several tests that this kind of smoothers provides visually pleasing registration results within a few multigrid steps. However, it does not perform well as a good smoother in leading to the convergence of the FAS-NMG method. Note that this smoother can be derived directly from (13).

(2) The fixed point (FP) iteration based smoothers. The simple linearized iterations by the following

$$\begin{cases} -\alpha\Delta u_1^{[\nu+1]} = -(T(\mathbf{u}^{[\nu]}) - R)\partial_{u_1} T(\mathbf{u}^{[\nu]}) \\ -\alpha\Delta u_2^{[\nu+1]} = -(T(\mathbf{u}^{[\nu]}) - R)\partial_{u_2} T(\mathbf{u}^{[\nu]}) \end{cases} \tag{18}$$

as discussed in [3, p. 79] have been used by researchers; we shall call this method the standard FP (SFP) below. This SFP scheme encounters a singular system for all fixed point step ν due to Neumann's boundary conditions. Without any special treatment for overcoming the singularity, we found that simple iterative methods such as the Jacobi-, GS-, or successive over-relaxation (SOR) type methods usually fail to lead to convergence of the FAS-NMG technique; however we discuss ways of improving this idea below.

(3) The augmented system technique for the SFP scheme. This was introduced in [8,9]. The idea is to convert the singular system to a nonsingular one by augmenting extra equations. However, it may not lead to satisfaction of the necessary (first-order) condition of a minimizer of the variational problem represented by (2). In details, this smoother builds on the above SFP method (18) with an initial guess $\mathbf{u}^{[0]}$

$$\begin{cases} -\alpha\mathcal{L}(u_1^{[\nu+1]}) = g_1 - f_1(u_1^{[\nu]}, u_2^{[\nu]}) = G_1^{[\nu]} \\ -\alpha\mathcal{L}(u_2^{[\nu+1]}) = g_2 - f_2(u_1^{[\nu]}, u_2^{[\nu]}) = G_2^{[\nu]}. \end{cases} \quad \text{(SFP)} \tag{19}$$

Recall that $g_1 = g_2 = 0$ on the finest grid in multigrid setting and the symbol h and $(\cdot)_{i,j}$ in (8) are dropped for simplicity. As already mentioned above, the resulting system is singular in case of Neumann's boundary conditions approximated by (9). The reason for singularity is due to zero row sums at boundary points, i.e. the constant functions lie in the kernel of $\mathcal{L}$. In this case the discrete system has a solution if and only if the discrete compatibility condition

$$\sum_{i,j=1}^{n_1,n_2} (G_l^{[\nu]})_{i,j} = 0 \quad \text{for } l = 1, 2$$

is satisfied [37,8,9,30]. Obviously, this condition fails when the given images are substantially different. Recognizing the above difficulties, Frohn-Schauf et al. [8,9] proposed to solve a nearby problem created by a simple modification, which guarantees that discrete solutions exist for each fixed point problem from replacing $G_l^{[\nu]}$ ($l = 1, 2$) by

$$\widetilde{G}_l^{[\nu]} = G_l^{[\nu]} - \frac{\langle G_l^{[\nu]}, \mathbb{I}\rangle}{\langle \mathbb{I}, \mathbb{I}\rangle}\mathbb{I},$$

where $\mathbb{I}$ is the $n_1 \times n_2$-vector $(1, \ldots, 1)$. Note that if $\mathbf{u}^{[\nu+1]}$ solves the new discrete system

$$\begin{cases} -\alpha\mathcal{L}(u_1^{[\nu+1]}) = \widetilde{G}_1^{[\nu]} \\ -\alpha\mathcal{L}(u_2^{[\nu+1]}) = \widetilde{G}_2^{[\nu]}, \end{cases} \tag{20}$$

then $\mathbf{u}^{[\nu+1]} + \boldsymbol{c}$ also solves the same problem for any $\boldsymbol{c}$. This means that the solution is not unique. In order to determine the unique solution of the discrete system given by (20), they put a constraint on $\mathbf{u}^{[\nu+1]}$. This can be done by applying the zero-mean condition,

$$\sum_{i,j=1}^{n_1,n_2} (u_l^{[\nu+1]})_{i,j} = \langle u_l^{[\nu+1]}, \mathbb{I}\rangle = c_l = 0 \quad \text{for } l = 1, 2. \tag{21}$$

We shall denote the above method by MSFP-FS (a modified SFP due to Frohn-Schauf et al.).

(4) The modified standard FP smoothers. Following the same idea of overcoming singularities, below, consider 2 alternative ways of modifying the SFP method (18)

$$\begin{cases} -\alpha\mathcal{L}(u_1^{[\nu+1]}) + c_1 u_1^{[\nu+1]} = G_1^{[\nu]} + c_1 u_1^{[\nu]} \\ -\alpha\mathcal{L}(u_2^{[\nu+1]}) + c_2 u_2^{[\nu+1]} = G_2^{[\nu]} + c_2 u_2^{[\nu]}, \end{cases} \quad \text{(MSFP-1)} \tag{22}$$

$$\begin{cases} (-\alpha\mathcal{L} + \epsilon)u_1^{[\nu+1]} = G_1^{[\nu]} \\ (-\alpha\mathcal{L} + \epsilon)u_2^{[\nu+1]} = G_2^{[\nu]}, \end{cases} \quad \text{(MSFP-2)} \tag{23}$$

where c_l $(l = 1, 2)$ and ϵ are positive real numbers (very small number for ϵ). We note first that the modified SFP method of the first type (MSFP-1) given by (22) can be viewed as a semi-implicit time-marching scheme when $c_1 = c_2 = c > 0$ is interpreted to be the time-step, as used in [3, p. 80]. Second, the second type (MSFP-2) represented by (23) is the simplest way to stabilize the SFP method by adding the small number ϵ to all diagonal elements of $\mathcal{L}$, as used in [43] in a different context. Finally, we note that determining the optimal values for the fixed point parameters c_1, c_2 (or c), and ϵ in automatic procedures leading to convergence of the FP method (and a multigrid technique) is not straight forward for real-world applications because tuning is needed for different registration problems.

In other tests, we found experimentally that although the approximation solutions obtained from the MSFP-FS scheme are visually pleasing, they may not fulfill the necessary condition for being a minimizer of the variational problem (2). The reason is that this numerical scheme solves a nearby problem for each fixed point step ν by changing the right-hand side of (19) subject to the zero-mean condition (21).

These difficulties have motivated us to develop the new smoother in the next section.

3. The proposed algorithm with a new and robust smoother

To first design a better smoother for (5), we have to re-consider the nonlinear terms in (19) in a new FP scheme. Once this is done, a basic linear iterative solver such as the Jacobi, GS or SOR method for each corresponding system may be used. Then to improve the model robustness, we use a multi-resolution idea to choose the regularization parameter α.

3.1. The new FP smoother

Our idea of a new FP scheme is different from the SFP (18) and its variants, by aiming for full implicitness in both regularizer and data terms. This leads to

$$\begin{cases} -\alpha\mathcal{L}(u_1^{[\nu+1]}) + f_1(u_1^{[\nu+1]}, u_2^{[\nu+1]}) = g_1 \\ -\alpha\mathcal{L}(u_2^{[\nu+1]}) + f_2(u_1^{[\nu+1]}, u_2^{[\nu+1]}) = g_2 \end{cases} \tag{24}$$

before we introduce linearization. We linearize the data term $f_l(u_1^{[\nu+1]}, u_2^{[\nu+1]})$ via a first-order approximation of form (for $l = 1, 2$)

$$\begin{aligned} f_l(u_1^{[\nu+1]}, u_2^{[\nu+1]}) &\sim f_l(u_1^{[\nu]}, u_2^{[\nu]}) + \partial_{u_1} f_l(u_1^{[\nu]}, u_2^{[\nu]})\delta u_1^{[\nu]} + \partial_{u_2} f_l(u_1^{[\nu]}, u_2^{[\nu]})\delta u_2^{[\nu]}, \\ &= f_l(u_1^{[\nu]}, u_2^{[\nu]}) + \sigma_{l1}^{[\nu]}\delta u_1^{[\nu]} + \sigma_{l2}^{[\nu]}\delta u_2^{[\nu]}, \end{aligned} \tag{25}$$

where

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

and

$$\sigma_{l2}(\mathbf{u}^{[\nu]}) = \partial_{u_2} f_l(u_1^{[\nu]}, u_2^{[\nu]}) = (\partial_{u_l} T(\mathbf{u}^{[\nu]}))(\partial_{u_2} T(\mathbf{u}^{[\nu]})) + (T(\mathbf{u}^{[\nu]}) - R)(\partial_{u_2 u_l} T(\mathbf{u}^{[\nu]})).$$

By (25), it leads to the semi-implicitly fixed point iteration step in terms of a 2×2 matrix as follows

$$\mathbf{N}[\mathbf{u}^{[\nu]}]\mathbf{u}^{[\nu+1]} = \mathbf{G}[\mathbf{u}^{[\nu]}], \quad \text{(RFP)} \tag{26}$$

where RFP refers to robust FP (yet to be tested shortly), $\mathbf{u}^{[\nu+1]} = (u_1^{[\nu+1]}, u_2^{[\nu+1]})^\top$,

$$\mathbf{N}[\mathbf{u}^{[\nu]}] = \begin{bmatrix} -\alpha\mathcal{L} + \sigma_{11}(\mathbf{u}^{[\nu]}) & \sigma_{12}(\mathbf{u}^{[\nu]}) \\ \sigma_{21}(\mathbf{u}^{[\nu]}) & -\alpha\mathcal{L} + \sigma_{22}(\mathbf{u}^{[\nu]}) \end{bmatrix},$$
$$\mathbf{G}[\mathbf{u}^{[\nu]}] = \begin{pmatrix} G_1^{[\nu]} + \sigma_{11}(\mathbf{u}^{[\nu]})u_1^{[\nu]} + \sigma_{12}(\mathbf{u}^{[\nu]})u_2^{[\nu]} \\ G_2^{[\nu]} + \sigma_{21}(\mathbf{u}^{[\nu]})u_1^{[\nu]} + \sigma_{22}(\mathbf{u}^{[\nu]})u_2^{[\nu]} \end{pmatrix}.$$

To solve (26), we adopt the *block or pointwise collective Gauss–Seidel* (PCGS) relaxation method, i.e. all difference equations are updated simultaneously. A PCGS step is then given by

$$(\mathbf{u}^{[\nu+1]})_{i,j}^{[k+1]} = (\overline{\mathbf{N}}[\mathbf{u}^{[\nu]}]_{i,j})^{-1}(\mathbf{G}[\mathbf{u}^{[\nu]}])_{i,j}^{[k+1/2]}, \tag{27}$$

where using the same notation as (8)

$$\overline{\mathbf{N}}[\mathbf{u}^{[\nu]}]_{i,j} = \begin{bmatrix} (\alpha/h_1^2)(\Sigma)_{i,j} + (\sigma_{11}(\mathbf{u}^{[\nu]}))_{i,j} & (\sigma_{12}(\mathbf{u}^{[\nu]}))_{i,j} \\ (\sigma_{21}(\mathbf{u}^{[\nu]}))_{i,j} & (\alpha/h_1^2)(\Sigma)_{i,j} + (\sigma_{22}(\mathbf{u}^{[\nu]}))_{i,j} \end{bmatrix},$$
$$(\mathbf{G}[\mathbf{u}^{[\nu]}])_{i,j}^{[k+1/2]} = \begin{pmatrix} (G_1^{[\nu]})_{i,j} + (\sigma_{11}(\mathbf{u}^{[\nu]}))_{i,j}(u_1^{[\nu]})_{i,j} + (\sigma_{12}(\mathbf{u}^{[\nu]}))_{i,j}(u_2^{[\nu]})_{i,j} + (\alpha/h_1^2)(\overline{\Sigma})_{i,j}(u_1^{[\nu+1]})_{i,j}^{[k+1/2]} \\ (G_2^{[\nu]})_{i,j} + (\sigma_{21}(\mathbf{u}^{[\nu]}))_{i,j}(u_1^{[\nu]})_{i,j} + (\sigma_{22}(\mathbf{u}^{[\nu]}))_{i,j}(u_2^{[\nu]})_{i,j} + (\alpha/h_1^2)(\overline{\Sigma})_{i,j}(u_2^{[\nu+1]})_{i,j}^{[k+1/2]} \end{pmatrix},$$
$$(\overline{\Sigma})_{i,j}(u_l^{[\nu+1]})_{i,j}^{[k+1/2]} = (u_l^{[\nu+1]})_{i+1,j}^{[k]} + (u_l^{[\nu+1]})_{i-1,j}^{[k+1]} + \gamma^2(u_l^{[\nu+1]})_{i,j+1}^{[k]} + \gamma^2(u_l^{[\nu+1]})_{i,j-1}^{[k+1]}.$$

To gain more efficiency, one may introduce a relaxation parameter $\omega \in (0, 2)$ and iterate the ω-PCGS steps by

$$(\mathbf{u}^{[\nu+1]})_{i,j}^{[k+1]} = (1-\omega)(\mathbf{u}^{[\nu+1]})_{i,j}^{[k]} + \omega \underbrace{(\overline{\mathbf{N}}[\mathbf{u}^{[\nu]}]_{i,j})^{-1}(\mathbf{G}[\mathbf{u}^{[\nu]}])_{i,j}^{[k+1/2]}}_{\text{original PCGS result}}. \tag{28}$$

We note first that the proposed smoother given by (26) shows the interaction between the actual fixed point iteration that overcomes the nonlinearity of the operator $\mathcal{N}_l$ at each outer step ν and the ω-PCGS method that solves the resulting linear system of equations at each corresponding inner step k. Second, instead of solving the linear system of equations using the inner solver (28) with very high precision, it can perform only a *few iteration* to obtain an approximation solution at each outer step. Evidently, this procedure leads to a slight difference of convergence in the fixed point scheme when the proposed smoother is used as a stand-alone solver, whereas the computational costs significantly reduce; see Fig. 1(a). Moreover, the relaxation parameter ω also has a strong influence on the convergence speed. As the stand-alone solver of (26) we usually use $\omega > 1$, typically $\omega = 1.85$, because it results in speeding up the convergence compared with the PCGS approach ($\omega = 1$); see Fig. 1(b). For our multigrid algorithm, however, we use the local Fourier analysis and several experiments to select the optimal value of ω; see Section 3.2 later. Finally we remark that other iterative techniques such as the line relaxation techniques or the preconditioned conjugate gradient method may also be used as an inner solver. However, the ω-PCGS relaxation method appears a cheaper option for practical applications.

Implementation of our proposed smoother (26) based on the ω-PCGS method (28) on a fine grid can be summarized as follows:

Algorithm 3 (*Our Proposed Smoother*)**.** Denote by

α regularization parameter

ω relaxation parameter

GSiter the maximum number of ω-PCGS iterations

$[v_1^h, v_2^h] \leftarrow Smoother(v_1^h, v_2^h, g_1^h, g_2^h, R^h, T^h, \alpha, \omega, GSiter)$.

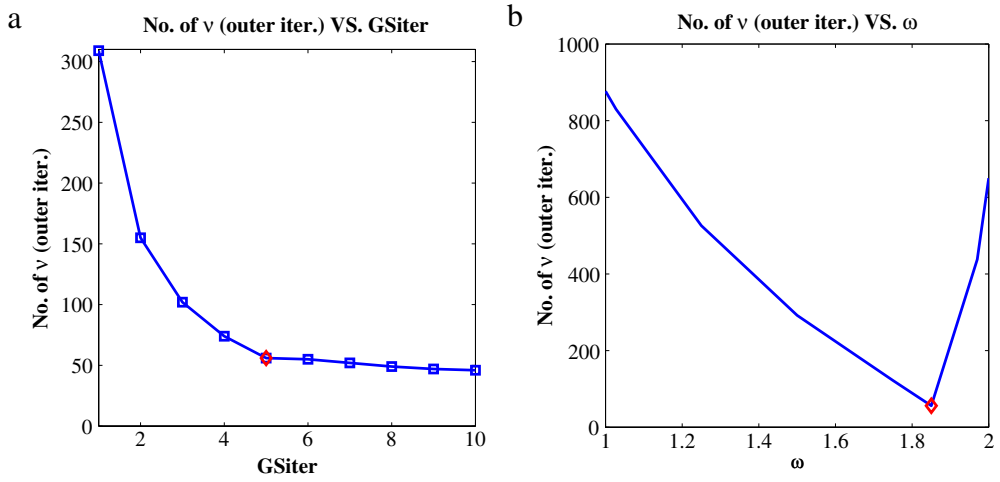

Fig. 1. Number of outer iterations ν in (26) used to drop the mean of relative residuals of (24) to 10^{-8} for different values of (a) *GSiter* and (b) ω at a fixed value of $\alpha = 0.1$ for processing the registration problem in Examples 1 as shown in Fig. 3(a)–(b) on a 32×32 grid. The red diamond indicates the optimal choice in each plot.

- Use input parameters to compute $(\sigma_{lm}[\boldsymbol{v}^h])_{i,j}$, $(\mathbf{G}[\boldsymbol{v}^h])_{i,j}$, and $(\overline{\mathbf{N}}[\boldsymbol{v}^h]_{i,j})^{-1}$
  for $l, m = 1, 2$, $1 \le i \le n_1$, and $1 \le j \le n_2$ (Here $(\boldsymbol{v}^h)_{i,j} = ((v_1^h)_{i,j}, (v_2^h)_{i,j})^\top$).
- Perform ω-PCGS steps
  - for $k = 1 : \mathit{GSiter}$
    - for $i = 1 : n_1$
      - for $j = 1 : n_2$
        - Compute $(\boldsymbol{v}^h)_{i,j}^{[k+1]} = ((v_1^h)_{i,j}^{[k+1]}, (v_2^h)_{i,j}^{[k+1]})^\top$ using (28)
      - end
    - end
  - end

We remark that in a different context [48,32] for a different technique, a first-order approximation of

$$\begin{aligned} T(\mathbf{u}^{[\nu+1]}) &= T(\mathbf{x} + \mathbf{u}^{[\nu+1]}) = T(\mathbf{x} + \mathbf{u}^{[\nu]} + \delta\mathbf{u}^{[\nu]}) \\ &\approx T(\mathbf{x} + \mathbf{u}^{[\nu]}) + \partial_{u_1} T(\mathbf{x} + \mathbf{u}^{[\nu]})\delta u_1^{[\nu]} + \partial_{u_2} T(\mathbf{x} + \mathbf{u}^{[\nu]})\delta u_2^{[\nu]}, \end{aligned} \tag{29}$$

similar to expanding $f_l(u_1^{[\nu+1]}, u_2^{[\nu+1]})$ in (25) is used to derive the FP schemes. We also wish to remark that the above quantities $\sigma_{l1}(\mathbf{u}^{[\nu]})$, $\sigma_{l2}(\mathbf{u}^{[\nu]})$ may be refined to derive a cheaper implementation than our FP method (26). We note first that $\sigma_{21}(\mathbf{u}^{[\nu]}) = \sigma_{12}(\mathbf{u}^{[\nu]})$. Second in order to have a simple and stable numerical scheme as pointed out by several works; see e.g. [18] and [3, p. 56/79], we approximate $\sigma_{lm}(\mathbf{u}^{[\nu]})$ by $\sigma_{lm}(\mathbf{u}^{[\nu]}) = (\partial_{u_l} T(\mathbf{u}^{[\nu]}))(\partial_{u_m} T(\mathbf{u}^{[\nu]}))$ for $m = 1, 2$ since the image difference $T(\mathbf{u}^{[\nu]}) - R$ becomes small for well registered images and the second-order derivatives of $T(\mathbf{u}^{[\nu]})$ $(\partial_{u_m u_l} T(\mathbf{u}^{[\nu]}))$, a problematic part of $\sigma_{lm}(\mathbf{u}^{[\nu]})$, are very sensitive to noise and are hard to estimate robustly. Finally, we note that if discrete image gradient $\partial_{u_l} T(\mathbf{u}^{[\nu]})$ does not vanish at one point, the system matrix of these linearized equations is strictly or irreducibly diagonally dominant. This guarantees the existence of a unique solution of each linearized system and global convergence of the Jacobi and GS iterations [49,50].

Below we analyze the smoothing property of our proposed smoother (26).

3.2. Local Fourier analysis (LFA)

LFA is a powerful tool to analyze the smoothing properties of iterative algorithms used in MG methods. Although LFA was originally developed for discrete linear operators with constant coefficients on infinite grids, it can also be applied to more general nonlinear equations with varying coefficients such as the discrete version of (5). To this end, first an infinite grid is assumed to eliminate the effect of boundary conditions and second it is also assumed that the discrete nonlinear operator can be linearized (by freezing coefficients) and replaced locally by a new operator with constant coefficients [30].

This approach has proved to be very useful in the understanding of MG methods when solving nonlinear problems; see for instance [42,51,43,44,52,12,53,54,18,55] for interesting examples and discussions.

3.2.1. Measure of h-ellipticity

It is well known that h-ellipticity is crucial for multigrid methods to be effective. It is often used to decide whether or not pointwise error smoothing procedures (e.g. our proposed smoother (26) based on (28)) can be constructed for the discrete operator under consideration. To this end, we shall show that the linearized system $\mathbf{N}_h[\overline{\mathbf{u}}^h]\mathbf{u}^h = \mathbf{G}_h[\overline{\mathbf{u}}^h]$ in (26) at some outer step provides a sufficient amount of h-ellipticity in a similar way shown in [12,18,30,40] for a discrete system of PDEs. Here $\mathbf{u}^h$ and $\overline{\mathbf{u}}^h$ denote the exact solution and the current approximation and $\mathbf{N}_h[\overline{\mathbf{u}}^h]$ and $\mathbf{G}_h[\overline{\mathbf{u}}^h]$ the resulting discrete operators from the linearization at $\overline{\mathbf{u}}^h$. For simplicity, our analysis is carried out over the infinite grid

$$\Omega_h^\infty = \{\mathbf{x} \in \Omega | \mathbf{x} = (x_{1_i}, x_{2_j})^\top = ((2i-1)h/2, (2j-1)h/2)^\top,\ i, j \in \mathbb{Z}^2\} \tag{30}$$

where $h = 1/n$ denotes the mesh parameter.

Let $\boldsymbol{\varphi}_h(\boldsymbol{\theta}, \mathbf{x}) = \exp(\mathbf{i}\boldsymbol{\theta}\mathbf{x}/h) \cdot \widehat{\mathbf{I}}$ be grid functions, where $\widehat{\mathbf{I}} = (1,1)^\top$, $\boldsymbol{\theta} = (\theta_1, \theta_2)^\top \in \boldsymbol{\Theta} = (-\pi, \pi]^2$, $\mathbf{x} \in \Omega_h^\infty$, and $\mathbf{i} = \sqrt{-1}$. It is important to remark that due to the locality nature of LFA, our analysis applies to each grid point separately, i.e. we consider the local discrete system $\mathbf{N}_h(\xi)\mathbf{u}^h = \mathbf{G}_h(\xi)$ centered and defined only within a small neighborhood of each grid point $\xi = (i, j)$ and $\mathbf{u}^h(\xi) = [u_1^h(\xi), u_2^h(\xi)]$. Applying the discrete operator $\mathbf{N}_h(\xi)$ to the grid functions $\boldsymbol{\varphi}_h(\boldsymbol{\theta}, \mathbf{x})$, i.e. $\mathbf{N}_h(\xi)\boldsymbol{\varphi}_h(\boldsymbol{\theta}, \mathbf{x}) = \widehat{\mathbf{N}}_h(\xi, \boldsymbol{\theta})\boldsymbol{\varphi}_h(\boldsymbol{\theta}, \mathbf{x})$, yields the Fourier symbol as follows:

$$\widehat{\mathbf{N}}_h(\xi, \boldsymbol{\theta}) = \begin{bmatrix} -\alpha\widehat{\mathcal{L}}^h(\boldsymbol{\theta}) + \sigma_{11}(\xi) & \sigma_{12}(\xi) \\ \sigma_{21}(\xi) & -\alpha\widehat{\mathcal{L}}^h(\boldsymbol{\theta}) + \sigma_{22}(\xi) \end{bmatrix} \tag{31}$$

(see details related to Fourier symbols of systems of PDEs in [30,40]). Here $\widehat{\mathcal{L}}^h(\boldsymbol{\theta})$ denotes the Fourier symbol of the discrete Laplacian operator $\widehat{\mathcal{L}}^h$. Following [30,40], the measure of h-ellipticity is defined via $\widehat{\mathbf{N}}_h(\boldsymbol{\xi}, \boldsymbol{\theta})$ as follows:

$$E_h(\mathbf{N}_h(\xi)) = \frac{\min\{|\det(\widehat{\mathbf{N}}_h(\xi, \boldsymbol{\theta}))| : \boldsymbol{\theta} \in \boldsymbol{\Theta}_{\text{high}}\}}{\max\{|\det(\widehat{\mathbf{N}}_h(\xi, \boldsymbol{\theta}))| : \boldsymbol{\theta} \in \boldsymbol{\Theta}\}}, \tag{32}$$

where $\boldsymbol{\Theta}_{\text{high}} = \boldsymbol{\Theta} \setminus (-\pi/2, \pi/2]^2$ denotes the range of high frequencies and

$$\det(\widehat{\mathbf{N}}_h(\xi, \boldsymbol{\theta})) = \alpha^2(\widehat{\mathcal{L}}^h(\boldsymbol{\theta}))^2 + \alpha c_1(\widehat{\mathcal{L}}^h(\boldsymbol{\theta})) + c_2 \tag{33}$$

represents the determinant of $\widehat{\mathbf{N}}_h(\xi, \boldsymbol{\theta})$ where

$$c_1 = -(\sigma_{11}(\xi) + \sigma_{22}(\xi)) \quad \text{and} \quad c_2 = \sigma_{11}(\xi)\sigma_{22}(\xi) - \sigma_{12}(\xi)\sigma_{21}(\xi).$$

For the Laplacian, we have

$$-\widehat{\mathcal{L}}^h(\boldsymbol{\theta}) = (2/h^2)(2 - (\cos\theta_1 + \cos\theta_2)), \qquad \min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}_{\text{high}}}(-\widehat{\mathcal{L}}^h(\boldsymbol{\theta})) = -\widehat{\mathcal{L}}^h(-\pi/2, 0) = 2/h^2$$

and $\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}(-\widehat{\mathcal{L}}^h(\boldsymbol{\theta})) = -\widehat{\mathcal{L}}^h(\pi, \pi) = 8/h^2$. Therefore,

$$E_h(\mathbf{N}_h(\xi)) = \frac{2\alpha + c_1h^2 + (c_2h^4/2\alpha)}{32\alpha + 4c_1h^2 + (c_2h^4/2\alpha)} \tag{34}$$

and

$$\lim_{h\to 0} E_h(\mathbf{N}_h(\xi)) = \frac{1}{16} \tag{35}$$

bounded away from 0 for all possible choices α, $h > 0$ and for all possible values of $\sigma_{11}(\xi)$, $\sigma_{12}(\xi)$, $\sigma_{21}(\xi)$, and $\sigma_{22}(\xi)$ (i.e. the results do not depend on the given images) over the whole discrete domain Ω_h.

As a result, the discrete system $\mathbf{N}_h[\overline{\mathbf{u}}^h]\mathbf{u}^h = \mathbf{G}_h[\overline{\mathbf{u}}^h]$ is appropriate to pointwise error smoothing procedures and so is our proposed smoother (26) combined with the ω-PCGS method (28).

3.2.2. Smoothing analysis for the proposed smoother

A robust and potential smoother has to take care of the high-frequency components of the error between the exact solution and the current approximation since the low-frequency components becomes the high-frequency components on coarser grids and they cannot be reduced on coarser grids by the coarse-grid correction procedure. A quantitative measure of the smoothing efficiency for a given algorithm is the *smoothing factor* denoted by μ from a LFA and numerically computed for test problems, which is defined as the worst asymptotic error reduction, by performing one smoother step, of all high-frequency error components [30,40]. Below we shall analyze the smoothing properties of the proposed smoother via (26) and (28).

As pointed out in many cases of nonlinear operators with varying coefficients in [42,51,43,44,52,12,53,54,18,55], the smoothing factor is **x**-dependent. Therefore, it is customary to look for the maximum over the local smoothing factors of the frozen operator $\mathbf{N}_h(\xi)$, i.e.

$$\mu^*_{\text{loc}} = \max_{\zeta\in\Omega_h} \mu_{\text{loc}}. \tag{36}$$

To determine μ_{loc} we consider again the local discrete system $\mathbf{N}_h(\xi)\mathbf{u}^h(\xi) = \mathbf{G}_h(\xi)$. By using the splitting $\mathbf{N}_h(\xi) = \mathbf{N}_h^{[+]}(\xi) + \mathbf{N}_h^{[0]}(\xi) + \mathbf{N}_h^{[-]}(\xi)$, it is possible to write the local inner iterations of (26) (for $\omega = 1$) as

$$\mathbf{N}_h^{[+]}(\xi)\overline{\mathbf{u}}^h_{new}(\xi) + \mathbf{N}_h^{[0]}(\xi)\overline{\mathbf{u}}^h_{new}(\xi) + \mathbf{N}_h^{[-]}(\xi)\overline{\mathbf{u}}^h_{old}(\xi) = \mathbf{G}_h(\xi) \tag{37}$$

where $\overline{\mathbf{u}}^h_{old}(\xi)$ and $\overline{\mathbf{u}}^h_{new}(\xi)$ stand for the approximations to $\mathbf{u}^h(\xi)$ before and after the inner smoothing step, respectively and

$$\mathbf{N}_h^{[+/0/-]}(\xi) = \begin{bmatrix} (\mathbf{N}_h^{[+/0/-]}(\xi))_{1,1} & (\mathbf{N}_h^{[+/0/-]}(\xi))_{1,2} \\ (\mathbf{N}_h^{[+/0/-]}(\xi))_{2,1} & (\mathbf{N}_h^{[+/0/-]}(\xi))_{2,2} \end{bmatrix}.$$

For a specification of this splitting, we use the stencil notation as follows:

$$\mathcal{L}^h_{[+]} = \frac{\alpha}{h^2}\begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}, \qquad \mathcal{L}^h_{[0]} = \frac{\alpha}{h^2}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad \mathcal{L}^h_{[-]} = \frac{\alpha}{h^2}\begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix},$$

$$(\mathbf{N}_h^{[+/-]}(\xi))_{l,m} = \begin{cases} \mathcal{L}^h_{[+/-]} & \text{for } l = m \\ 0, & \text{for } l \neq m, \end{cases} \qquad (\mathbf{N}_h^{[0]}(\xi))_{l,m} = \kappa_{l,m}\mathcal{L}^h_0 \quad (l, m = 1, 2),$$

and

$$\boldsymbol{\kappa} = \begin{bmatrix} \kappa_{1,1} & \kappa_{1,2} \\ \kappa_{2,1} & \kappa_{2,2} \end{bmatrix} = \begin{bmatrix} \Sigma + (h^2/\alpha)\sigma_{11}(\xi) & (h^2/\alpha)\sigma_{12}(\xi) \\ (h^2/\alpha)\sigma_{21}(\xi) & \Sigma + (h^2/\alpha)\sigma_{22}(\xi) \end{bmatrix}.$$

By subtracting (37) from $\mathbf{N}_h(\xi)\mathbf{u}^h(\xi) = \mathbf{G}_h(\xi)$ and defining $\overline{\mathbf{e}}^h_{new}(\xi) = \mathbf{u}^h(\xi) - \mathbf{u}^h_{new}(\xi)$ and $\overline{\mathbf{e}}^h_{old}(\xi) = \mathbf{u}^h(\xi) - \mathbf{u}^h_{old}(\xi)$ we obtain the local system of error equations

$$\mathbf{N}_h^{[+]}(\xi)\overline{\mathbf{e}}^h_{new}(\xi) + \mathbf{N}_h^{[0]}(\xi)\overline{\mathbf{e}}^h_{new}(\xi) + \mathbf{N}_h^{[-]}(\xi)\overline{\mathbf{e}}^h_{old}(\xi) = 0$$

or

$$\overline{\mathbf{e}}^h_{new}(\xi) = -[\mathbf{N}_h^{[0]}(\xi) + \mathbf{N}_h^{[+]}(\xi)]^{-1}[\mathbf{N}_h^{[-]}(\xi)]\overline{\mathbf{e}}^h_{old}(\xi) = \mathbf{S}_h(\xi)\overline{\mathbf{e}}^h_{old}(\xi) \tag{38}$$

where $\mathbf{S}_h(\xi)$ is the amplification factor. The effect of $\mathbf{S}_h(\xi)$ on the grid functions $\boldsymbol{\varphi}_h(\boldsymbol{\theta}, \mathbf{x})$ within $\boldsymbol{\Theta}_{\text{high}} = \boldsymbol{\Theta} \setminus [-\pi/2, \pi/2]^2$ will determine the smoothing properties of the PCGS method (27).

For the ω-PCGS approach (28), the amplification factor denoted by $\mathbf{S}_h(\xi, \omega)$ can be defined in a similar way as (38) and its Fourier symbol is given by

$$\widehat{\mathbf{S}}_h(\xi, \boldsymbol{\theta}, \omega) = [\widehat{\mathbf{N}}_h^0(\xi, \boldsymbol{\theta}) + \omega\widehat{\mathbf{N}}_h^+(\xi, \boldsymbol{\theta})]^{-1}[(1-\omega)\widehat{\mathbf{N}}_h^0(\xi, \boldsymbol{\theta}) - \omega\widehat{\mathbf{N}}_h^-(\xi, \boldsymbol{\theta})] \in \mathbb{C}^{2\times 2}. \tag{39}$$

Here the Fourier symbols of $\mathbf{N}_h^{[+/0/-]}(\xi)$ are

$$\widehat{\mathbf{N}}_h^{[+/0/-]}(\xi, \boldsymbol{\theta}) = \begin{bmatrix} (\widehat{\mathbf{N}}_h^{[+/0/-]}(\xi, \boldsymbol{\theta}))_{1,1} & (\widehat{\mathbf{N}}_h^{[+/0/-]}(\xi, \boldsymbol{\theta}))_{1,2} \\ (\widehat{\mathbf{N}}_h^{[+/0/-]}(\xi, \boldsymbol{\theta}))_{2,1} & (\widehat{\mathbf{N}}_h^{[+/0/-]}(\xi, \boldsymbol{\theta}))_{2,2} \end{bmatrix}. \tag{40}$$

Therefore, the local smoothing factor is

$$\mu_{\text{loc}} = \sup\{|\rho(\widehat{\mathbf{S}}_h(\xi, \boldsymbol{\theta}, \omega))| : \boldsymbol{\theta} \in \boldsymbol{\Theta}_{\text{high}}\} \tag{41}$$

where ρ indicates the spectral radius of $\widehat{\mathbf{S}}_h(\xi, \boldsymbol{\theta}, \omega)$. Recall that

$$(\mathbf{N}_h^{[+]}(\xi, \boldsymbol{\theta}))_{l,m} = \begin{cases} -\dfrac{\alpha}{h^2}(\exp(-\mathbf{i}\theta_1) + \exp(-\mathbf{i}\theta_2)) & \text{for } l = m \\ 0, & \text{for } l \neq m, \end{cases} \tag{42}$$

$$(\mathbf{N}_h^{[-]}(\xi, \boldsymbol{\theta}))_{l,m} = \begin{cases} -\dfrac{\alpha}{h^2}(\exp(\mathbf{i}\theta_1) + \exp(\mathbf{i}\theta_2)) & \text{for } l = m \\ 0, & \text{for } l \neq m \end{cases} \tag{43}$$

$$(\mathbf{N}_h^{[0]}(\xi, \boldsymbol{\theta}))_{l,m} = \frac{\alpha}{h^2}\kappa_{l,m} \tag{44}$$

will be used to compute (39).

To select the optimal value of ω and test our smoother we consider one set of medical images as shown respectively in Fig. 3(a)–(b) on a 32×32 grid. Fig. 2 shows the smoothing factors of the proposed smoother (26) based on the ω-PCGS approach (28) at different values of ω. It indicates that the optimal value ω providing $\mu^*_{\text{loc}} \approx 0.5$ is not exactly 1 but very close to 1, typically $\omega = 0.9725$.

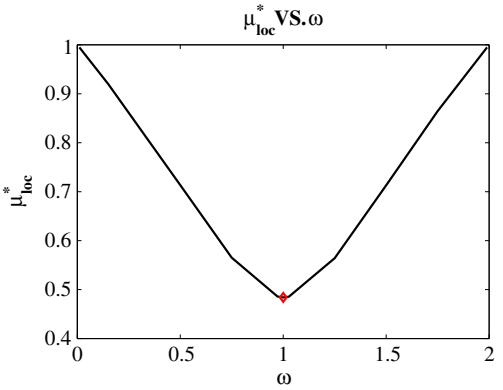

Fig. 2. Smoothing factors μ^*_{loc} at a fixed value of $\alpha = 0.1$ after 5 outer iterations with $GSiter = 5$ by the proposed smoother (26) based on the ω-PCGS approach (28) with different values of ω for the registration problem in Examples 1 as shown in Fig. 3(a)–(b) on a 32×32 grid. The red diamond indicates the optimal value of ω.

3.3. A nonlinear multigrid algorithm with automatic selection of α

As is typical of Tikhonov regularization, the energy functional $\mathcal{J}_\alpha$ in (2) has a regularization parameter α. To provide well matched images, we have to carefully select α because it is in general unknown a priori. In order to find a suitable α automatically, we follow the 'cooling' ('continuation') process suggested in [56–58,12,15,59,60]. The basic idea is to start with a high initial value of α and then slowly reduce α such that the obtained solution can be used to be an excellent starting point for the next in order to decrease $\mathcal{J}_\alpha$. An alternative approach can be the L-curve method.

Consider the discrete version of the minimization problem (2) with the same notation

$$\min \mathcal{J}_\alpha[\mathbf{u}] = \mathcal{D}(R, T(\mathbf{u})) + \alpha \mathcal{R}(\mathbf{u}). \tag{45}$$

Let α_1 be the initial value, which is sufficiently large. At the $(s+1)$th step we set $\alpha^{(s+1)} = \beta\alpha^{(s)} \in [\alpha_0, \alpha_1]$, where $\beta \in (0, 1)$ is a constant, usually chosen to be about 0.5, and α_0 is a small positive number, e.g. 5×10^{-5}. Subsequently, we apply $\alpha^{(s+1)}$ and the initial guess solution obtained by the previous iteration $\mathbf{u}^{(s+1)}_{initial} = \mathbf{u}^{(s)}$ with the associated inner loop to obtain the minimum $\mathbf{u}^{(s+1)}$ within some tolerance. As mentioned in [58], since the functional $\mathcal{J}_\alpha$ is changing at each outer loop iteration, the demand of decreasing the value of the same functional is not reasonable. Then, the solution $\mathbf{u}^{(s+1)}$ and parameter $\alpha^{(s+1)}$ are acceptable if they satisfy the so-called *consistent condition*:

$$\mathcal{J}_{\alpha^{(s+1)}}[\mathbf{u}^{(s+1)}] = \mathcal{D}[\mathbf{u}^{(s+1)}] + \alpha^{(s+1)}\mathcal{R}[\mathbf{u}^{(s+1)}] < \mathcal{J}_{\alpha^{(s+1)}}[\mathbf{u}^{(s)}] = \mathcal{D}[\mathbf{u}^{(s)}] + \alpha^{(s+1)}\mathcal{R}[\mathbf{u}^{(s)}].$$

However, if this condition is not satisfied, we increase β (usually to 0.9) and re-start the step. Our experience suggests that the stopping criterion given by

$$\frac{\|\mathbf{u}^{(s+1)} - \mathbf{u}^{(s)}\|_{l_2}}{\max\{\|\mathbf{u}^{(s+1)}\|_{l_2}, \|\mathbf{u}^{(s)}\|_{l_2}\}} < \delta \tag{46}$$

is suitable, where $\delta > 0$ is small (normally set to 10^{-3}).

Finally, we summarize this process as follows:

Algorithm 4 (*Multigrid Image Registration Through Cooling*)**.**

$[\boldsymbol{v}^*, \alpha^*] \leftarrow cooling(\boldsymbol{v}^{(0)}, \alpha^{(0)}, \overrightarrow{\varepsilon})$.

- Set $s = 1$, $\boldsymbol{v}^{(s)} = \boldsymbol{v}^{(0)}$, $\alpha^{(s)} = \alpha^{(0)}$, $\beta = 0.5$.
- Outer iteration: For $s = 1, 2, 3, \ldots$
  - 1. Set $\alpha^{(s+1)} = \beta\alpha^{(s)}$ in $[5 \times 10^{-5}, \alpha^{(s)}]$.
  - 2. Inner iteration: $\boldsymbol{v}_{new} \leftarrow FASMG(\boldsymbol{v}^{(s)}, \alpha^{(s+1)}, \overrightarrow{\varepsilon})$.
  - 3. If $\mathcal{J}_{\alpha^{(s+1)}}[\boldsymbol{v}_{new}] < \mathcal{J}_{\alpha^{(s+1)}}[\boldsymbol{v}^{(s)}]$.
    - 3.1 Set $\boldsymbol{v}^{(s+1)} = \boldsymbol{v}_{new}$, $\beta = 0.5$, $s = s + 1$, and go to 4.
      Else
    - 3.2 Set $\beta = 0.9$, and go to 4.
  - 4. Check for convergence using the criterion (46).
    If not satisfied, then return to 1, else, exit to the next step to stop.
- Set $\boldsymbol{v}^* = \boldsymbol{v}_{new}$ and $\alpha^* = \alpha^{(s)}$.

In order to save computational work for high-resolution digital images, the low-tolerance $\overrightarrow{\varepsilon}_{lo} = (2, 10^{-4}, 0.1, 10^{-4})$ is applied to reduce the accumulated costs in each minimization problem. Then our first algorithm, namely *a robust diffusion image registration* (RDR) *approach*, can be stated as follows:

Algorithm 5 (*The Basic RDR Method*)**.**

1. Input $\overrightarrow{\varepsilon}_{lo}$.Set $\alpha = 1$ (optional). Set $\overrightarrow{\varepsilon}_{hi} = (20, 10^{-8}, 0.10, 10^{-8})$ (high-tolerance).
2. Obtain the optimal regularization parameter α (through cooling) via Algorithm 4:
   – $[\boldsymbol{v}^{(0)}, \alpha] \leftarrow cooling(\boldsymbol{v}, \alpha, \overrightarrow{\varepsilon}_{lo})$.
3. Solve the discrete minimization problem (45) on the finest level using the found α:
   – $\boldsymbol{v} \leftarrow FASMG(\boldsymbol{v}^{(0)}, \alpha, \overrightarrow{\varepsilon}_{hi})$.

Although the above algorithm enables us to find a good α, the cost of resolving the same problem repeatedly is expensive. We propose to use a hierarchy of L grids (with level L the finest and level 1 the coarsest one) using a multi-resolution idea to gain efficiency while finding an effective α. Firstly we shall seek the optimal α on the coarsest level 1 with the grid size of 32×32 only (believed to be coarse enough) and secondly we use the multilevel continuation idea [12] to provide the initial guesses for the next finer level.

Algorithm 6 (*Multilevel Grid Continuation for Optimal α and Reliable Initial Solution*)**.**

$[\boldsymbol{v}^{[lev]}, \alpha^{[lev]}] \leftarrow RDR\_multiresolution(\boldsymbol{v}^{[lev]}, \alpha^{[lev]}, lev, \overrightarrow{\varepsilon})$.

- If $lev = 1$
  - $\boldsymbol{v}^{[lev]} = 0$
  - $\alpha^{[lev]} = C$ [$C > 0$ should be large enough e.g. $C = 100$]
  - $[\boldsymbol{v}^{[lev]}, \alpha^{[lev]}] \leftarrow cooling(\boldsymbol{v}^{[lev]}, \alpha^{[lev]}, \overrightarrow{\varepsilon})$
- Else
  - $\boldsymbol{v}^{[lev-1]} = (I_h^H v_1^{[lev]}, I_h^H v_2^{[lev]})^\top$
  - $[\boldsymbol{v}^{[lev-1]}, \alpha^{[lev-1]}] \leftarrow RDR\_multiresolution(\boldsymbol{v}^{[lev-1]}, \alpha^{[lev-1]}, lev - 1, \overrightarrow{\varepsilon})$
  - $\boldsymbol{v}^{[lev]} = (I_H^h v_1^{[lev-1]}, I_H^h v_2^{[lev-1]})^\top$
  - $\alpha^{[lev]} = 4\alpha^{[lev-1]}$ [Recall that $\alpha^{[lev]} = \overline{\alpha} n_{lev}^2$ and $n_{lel} = 2n_{lel-1}$]
  - $\boldsymbol{v}^{[lev]} \leftarrow FASMG(\boldsymbol{v}^{[lev]}, \alpha^{[lev]}, \overrightarrow{\varepsilon})$
- Endif

Finally the overall procedure of finding an optimal α and then starting a nonlinear multigrid method to solve (2) is summarized below as Algorithm 7:

Algorithm 7 (*The Refined RDR Multi-Resolution Method*)**.**

1. Input $\overrightarrow{\varepsilon}_{lo}$ and $\overrightarrow{\varepsilon}_{hi}$.
2. Obtain the optimal regularization parameter α (through cooling) and
   a good initial solution (through multi-resolution) $\mathbf{v}^{(0)}$ via Algorithm 6:
   – $[\boldsymbol{v}^{(0)}, \alpha] \leftarrow RDR\_multiresolution(\boldsymbol{v}^{[L]}, \alpha^{[L]}, L, \overrightarrow{\varepsilon}_{lo})$.
3. Solve the minimization problem (45) on the finest level $lev = L$ using the found α
   and the initial guess solution $\boldsymbol{v}^{(0)}$:
   – $\boldsymbol{v}^{[lev]} \leftarrow FASMG(\boldsymbol{v}^{(0)}, \alpha, \overrightarrow{\varepsilon}_{hi})$.

4. An application of Algorithm 7 to a curvature model

To test the robustness of our numerical algorithm for other registration models, in this section, we shall examine our second test model, namely, the *curvature image registration* model, as introduced in [5,6]; see also [7,18,33,3].

The curvature model. Based on an approximation of the mean curvature of the surface of u_l, Fischer–Modersitzki's curvature approach aims to find a reasonable deformation field $\mathbf{u}$ that minimizes the following functional [5,6]

$$\mathcal{J}_\alpha(\mathbf{u}) = \mathcal{D}(\mathbf{u}) + \alpha\mathcal{R}^{\mathrm{Cv}}(\mathbf{u})$$

where $\mathcal{R}^{\mathrm{Cv}}$ is as in (4). This leads to the Euler–Lagrange system of two fourth-order nonlinear PDEs:

$$\begin{cases} f_1(\mathbf{u}) + \alpha\Delta^2 u_1 = 0 \\ f_2(\mathbf{u}) + \alpha\Delta^2 u_2 = 0 \end{cases} \quad \text{(Fischer–Modersitzki's curvature model)} \tag{47}$$

subject to the special boundary conditions $\nabla u_l = 0,\ \nabla\Delta u_l = 0$ on $\partial\Omega$, for $l = 1, 2$. We remark that the use of second-order derivatives in the energy functional (4) not only provides smoother deformation fields $\mathbf{u}$ than those of (3), but also allows for an automatic rigid alignment. Here u_l is understood as a surface in $\mathbb{R}^3$ represented by $(x_1, x_2, u_l(x_1, x_2))$, where initially $u_l(x_1, x_2) = 0$, with the *mean curvature* of the surface of u_l is given by

$$\kappa_M(u_l) = \nabla\cdot\frac{\nabla u_l}{\sqrt{1+|\nabla u_l|^2}} = \frac{(1+u_{l_{x_1}}^2)u_{l_{x_1x_1}} - 2u_{l_{x_1}}u_{l_{x_2}}u_{l_{x_1x_2}} + (1+u_{l_{x_2}}^2)u_{l_{x_2x_2}}}{(1+u_{l_{x_1}}^2+u_{l_{x_2}}^2)^{3/2}}. \tag{48}$$

Observe that $|\nabla u_l| \approx 0$ yields $\kappa_M(u_l) \approx \widehat{\kappa}_M(u_l) = \Delta u_l$. See also [24].

Since the biharmonic operator which appears in (47) is known to lead to poor multigrid efficiency, therefore, it is common to split up the equation into a system of two Poisson-type equations; see [37,38,16,18,30,39,40]. Based on this splitting idea, (47) can be converted to the following system using additional unknown functions $v_1 = -\Delta u_1$ and $v_2 = -\Delta u_2$:

$$\begin{cases} -\Delta u_1 - v_1 = 0 \\ -\Delta u_2 - v_2 = 0 \\ f_1(\mathbf{u}) - \alpha\Delta v_1 = 0 \\ f_2(\mathbf{u}) - \alpha\Delta v_2 = 0 \end{cases} \tag{49}$$

subject to the boundary conditions transferred into $\nabla u_l = 0$ and $\nabla v_l = 0$ on $\partial\Omega$ where the data term $f_l(\mathbf{u})$ is given by (5).

To solve the above continuous system numerically, (49) is first discretized by the cell-centered finite difference scheme over the discrete domain

$$\Omega_h = \{\mathbf{x}\in\Omega\,|\mathbf{x} = (x_{1_i}, x_{2_j})^\top = ((2i-1)h/2, (2j-1)h/2),\ 1\le i,j\le n\}$$

consisting of $N = n^2$ cells of size $h\times h$ with grid spacing $h = 1/n$. Let $(z_{\widehat{l}}^h)_{i,j} = z_{\widehat{l}}^h(x_{1_i}, x_{2_j})$ denote the grid function for $\widehat{l} = 1,\ldots,4$, where $z_{\widehat{l}}^h = u_l^h$ or v_l^h for $l = 1, 2$. Then, the discrete system of (49) at a grid point (i, j) is given by

$$\begin{cases} \underbrace{-\mathcal{L}^h(u_1^h)_{i,j} - (v_1^h)_{i,j}}_{\mathcal{N}_1^h(\boldsymbol{z}^h)_{i,j}} = (g_1^h)_{i,j} \\ \underbrace{-\mathcal{L}^h(u_2^h)_{i,j} - (v_2^h)_{i,j}}_{\mathcal{N}_2^h(\boldsymbol{z}^h)_{i,j}} = (g_2^h)_{i,j} \\ \underbrace{f_1^h(u_1^h, u_2^h)_{i,j} - \alpha\mathcal{L}^h(v_1^h)_{i,j}}_{\mathcal{N}_3^h(\boldsymbol{z}^h)_{i,j}} = (g_3^h)_{i,j} \\ \underbrace{f_2^h(u_1^h, u_2^h)_{i,j} - \alpha\mathcal{L}^h(v_2^h)_{i,j}}_{\mathcal{N}_4^h(\boldsymbol{z}^h)_{i,j}} = (g_4^h)_{i,j}, \end{cases} \tag{50}$$

where $(\boldsymbol{z}^h)_{i,j} = ((z_1^h)_{i,j}, (z_2^h)_{i,j}, (z_3^h)_{i,j}, (z_4^h)_{i,j})^\top = ((u_1^h)_{i,j}, (u_2^h)_{i,j}, (v_1^h)_{i,j}, (v_2^h)_{i,j})^\top$ and $(g_{\widehat{l}}^h)_{i,j} = 0\ (\widehat{l} = 1,\ldots,4)$ on the finest grid in multigrid setting. Recall that the discrete versions of $\mathcal{L}^h$ and $f_l^h(u_1^h, u_2^h)_{i,j}$ are given in the same way as represented by Section 2.2 and the approximations used in (50) need to be modified at grid points near the image boundary $\partial\Omega_h$ using the homogeneous Neumann boundary conditions approximated by one-side differences for boundary derivatives:

$$(z_{\widehat{l}}^h)_{i,1} = (z_{\widehat{l}}^h)_{i,2}, \qquad (z_{\widehat{l}}^h)_{i,n} = (z_{\widehat{l}}^h)_{i,n-1}, \qquad (z_{\widehat{l}}^h)_{1,j} = (z_{\widehat{l}}^h)_{2,j}, \qquad (z_{\widehat{l}}^h)_{n,j} = (z_{\widehat{l}}^h)_{n-1,j}. \tag{51}$$

A robust smoother based on the linearization scheme (25). As mentioned above, our aim is to apply our linearization idea (25) in solving the equivalent system (49). This leads to the linearized system

$$\mathbf{N}^{\mathrm{Cv}}[\boldsymbol{z}^{[\nu]}]\boldsymbol{z}^{[\nu+1]} = \mathbf{G}^{\mathrm{Cv}}[\boldsymbol{z}^{[\nu]}], \tag{52}$$

where the symbols h and $(\cdot)_{i,j}$ in (50) are dropped for simplicity. Here

$$\mathbf{N}^{\mathrm{Cv}}[\boldsymbol{z}^{[\nu]}] = \begin{bmatrix} -\mathcal{L} & 0 & -1 & 0 \\ 0 & -\mathcal{L} & 0 & -1 \\ \sigma_{11}(\mathbf{u}^{[\nu]}) & \sigma_{12}(\mathbf{u}^{[\nu]}) & -\alpha\mathcal{L} & 0 \\ \sigma_{21}(\mathbf{u}^{[\nu]}) & \sigma_{22}(\mathbf{u}^{[\nu]}) & 0 & -\alpha\mathcal{L} \end{bmatrix} \tag{53}$$

and

$$\mathbf{G}^{\mathrm{Cv}}[\boldsymbol{z}^{[\nu]}] = \begin{pmatrix} g_1 \\ g_2 \\ g_3 - f_1(u_1^{[\nu]}, u_2^{[\nu]}) + \sigma_{11}(\mathbf{u}^{[\nu]})u_1^{[\nu]} + \sigma_{12}(\mathbf{u}^{[\nu]})u_2^{[\nu]} \\ g_4 - f_2(u_1^{[\nu]}, u_2^{[\nu]}) + \sigma_{21}(\mathbf{u}^{[\nu]})u_1^{[\nu]} + \sigma_{22}(\mathbf{u}^{[\nu]})u_2^{[\nu]} \end{pmatrix}. \tag{54}$$

As mentioned in Section 3.1, $\sigma_{21}^{[\nu]} = \sigma_{12}^{[\nu]}$ and $\sigma_{lm}^{[\nu]} = (\partial_{u_l} T(\mathbf{u}^{[\nu]}))(\partial_{u_m} T(\mathbf{u}^{[\nu]}))$ for $m = 1, 2$ are used to stabilize our numerical scheme. In order to solve (52), we again apply the ω-PCGS as the inner solver and its kth step is updated by

$$(\boldsymbol{z}^{[\nu+1]})_{i,j}^{[k+1]} = (1-\omega)(\boldsymbol{z}^{[\nu+1]})_{i,j}^{[k]} + \omega(\mathbf{N}^{\mathrm{Cv}}[\boldsymbol{z}^{[\nu]}]_{i,j})^{-1}(\mathbf{G}^{\mathrm{Cv}}[\boldsymbol{z}^{[\nu]}])_{i,j}^{[k+1/2]} \tag{55}$$

where

$$\mathbf{N}^{\mathrm{Cv}}[\boldsymbol{z}^{[\nu]}]_{i,j} = \begin{bmatrix} (\Sigma)_{i,j}/h^2 & 0 & -1 & 0 \\ 0 & (\Sigma)_{i,j}/h^2 & 0 & -1 \\ (\sigma_{11}(\mathbf{u}^{[\nu]}))_{i,j} & (\sigma_{12}(\mathbf{u}^{[\nu]}))_{i,j} & \alpha(\Sigma)_{i,j}/h^2 & 0 \\ (\sigma_{21}(\mathbf{u}^{[\nu]}))_{i,j} & (\sigma_{22}(\mathbf{u}^{[\nu]}))_{i,j} & 0 & \alpha(\Sigma)_{i,j}/h^2 \end{bmatrix} \tag{56}$$

$$(\Sigma)_{i,j} = \begin{cases} 2 \text{ at } (1,1),\ (n,1),\ (n,1), \text{ and } (n,n) \text{ (four corners)} \\ 3 \text{ for all } (1,j),\ (n,j),\ (i,1), \text{ and } (i,n) \text{ where } 2 \le i,j \le n-1 \text{ (boundary lines)} \\ 4 \text{ for all } (i,j) \text{ where } 2 \le i,j \le n-2 \text{ (interior points)}, \end{cases} \tag{57}$$

$$(\mathbf{G}^{\mathrm{Cv}}[\boldsymbol{z}^{[\nu]}])_{i,j}^{[k+1/2]} = \begin{pmatrix} (g_1)_{i,j} + (1/h^2)(\overline{\Sigma})_{i,j}(u_1^{[\nu+1]})_{i,j}^{[k+1/2]} \\ (g_2)_{i,j} + (1/h^2)(\overline{\Sigma})_{i,j}(u_2^{[\nu+1]})_{i,j}^{[k+1/2]} \\ (g_3)_{i,j} - f_1(u_1^{[\nu]}, u_2^{[\nu]})_{i,j} + (\sigma_{11}(\mathbf{u}^{[\nu]}))_{i,j}(u_1^{[\nu]})_{i,j} \\ + (\sigma_{12}(\mathbf{u}^{[\nu]}))_{i,j}(u_2^{[\nu]})_{i,j} + (\alpha/h^2)(\overline{\Sigma})_{i,j}(v_1^{[\nu+1]})_{i,j}^{[k+1/2]} \\ (g_4)_{i,j} - f_2(u_1^{[\nu]}, u_2^{[\nu]})_{i,j} + (\sigma_{22}(\mathbf{u}^{[\nu]}))_{i,j}(u_2^{[\nu]})_{i,j} \\ + (\sigma_{21}(\mathbf{u}^{[\nu]}))_{i,j}(u_1^{[\nu]})_{i,j} + (\alpha/h^2)(\overline{\Sigma})_{i,j}(v_2^{[\nu+1]})_{i,j}^{[k+1/2]} \end{pmatrix}, \tag{58}$$

and

$$(\overline{\Sigma})_{i,j}(z_{\widehat{l}}^{[\nu+1]})_{i,j}^{[k+1/2]} = (z_{\widehat{l}}^{[\nu+1]})_{i+1,j}^{[k]} + (z_{\widehat{l}}^{[\nu+1]})_{i-1,j}^{[k+1]} + (z_{\widehat{l}}^{[\nu+1]})_{i,j+1}^{[k]} + (z_{\widehat{l}}^{[\nu+1]})_{i,j-1}^{[k+1]}. \tag{59}$$

With the above smoother, the curvature model can be solved using our Algorithms 7–7.

Similarly to Section 3.2, we can show by the LFA that (1) $\lim_{h\to 0} E_h(\mathbf{N}_h^{\mathrm{Cv}}(\xi)) = \frac{1}{256}$, i.e. the linearized system (52) is h-elliptic; and (2) $\omega \approx 1$ (typically $\omega = 0.9725$) provides the good smoothing properties ($\mu_{\mathrm{loc}}^* \approx 0.5$), where the effects of ω on smoothing factors of (55) in processing the registration problem shown in Fig. 3(a)–(b) are almost identical with Fig. 2. We also conducted several numerical tests to confirm that (52) is a robust smoother for the FAS-NMG method to solve the curvature model (47); see Section 5.2.

5. Numerical experiments

The main aim of this section is to show that our new Algorithm 7 using (25) is effective and robust in leading to convergent multigrid methods in the FAS-NMG framework for both the diffusion and curvature models. In all experiments, the bilinear interpolation was used to compute the transformed template image $T(\mathbf{u})$.

5.1. The diffusion model tests

We first focus on the performance of Algorithm 7 for two sets of medical data, Examples[1] 1 and 2 shown respectively in Fig. 3(a)–(b) and (d)–(e). We first consider its convergence behavior with different resolutions, and then show comparisons with several other methods. In all experiments for this section $\nu_1 = 5$, $\nu_2 = 5$, *GSiter* $= 5$, and $\omega = 0.9725$ were used in this test.

[1] Source: http://www.math.mu-luebeck.de/safir/.

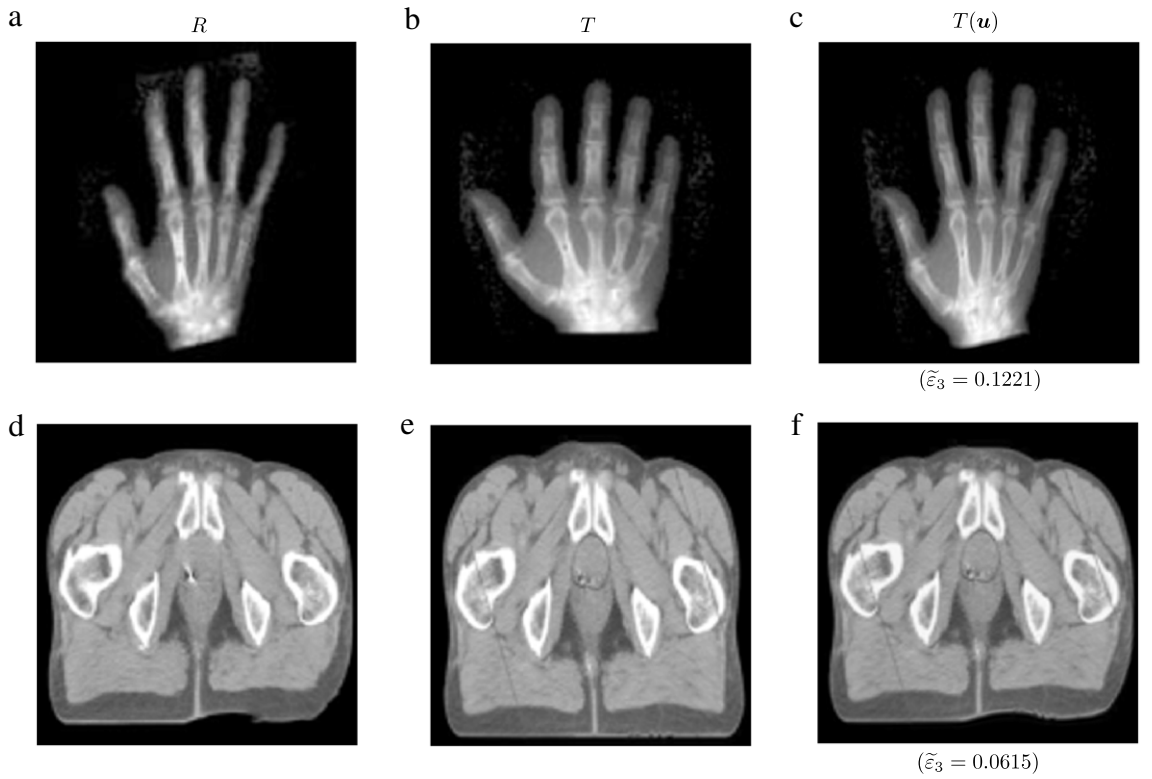

Fig. 3. Registration results for X-ray and MRI images using the RDR method with Algorithms 2, 5, and 7. Left column: reference R, center column: template T, right column: the deformed template image $T(\mathbf{u})$ obtained from Algorithm 7.

Table 1
Registration results of Algorithms 2, 5, and 7 for Examples 1 and 2 shown in Fig. 3(a)–(b) and (d)–(e). The letters 'M', 'R', 'D', 'C', and 'IC' mean the number of multigrid steps, the relative reduction of residual, the relative reduction of dissimilarity, the total runtimes, and the initial runtimes for determining the optimal α and initial guess $\mathbf{u}^{(0)}$, respectively.

	Algorithm 2 M/R/D/C	Algorithm 5 M/R/D/IC/C	Algorithm 7 M/R/D/IC/C
Example 1:	$\alpha = 0.1000$		
$h = 1/128$	$10/2.1 \times 10^{-9}/0.1082/4.3$	$6/1.8 \times 10^{-9}/0.1082/7.9/10.1$	$7/3.4 \times 10^{-9}/0.1082/2.1/6.2$
$h = 1/256$	$11/3.6 \times 10^{-9}/0.1161/22.5$	$6/3.1 \times 10^{-9}/0.1161/41.2/54.2$	$7/5.1 \times 10^{-9}/0.1161/3.9/24.2$
$h = 1/512$	$11/2.8 \times 10^{-9}/0.1221/106.4$	$6/2.5 \times 10^{-9}/0.1221/180.1/231.9$	$6/7.5 \times 10^{-9}/0.1221/7.3/65.1$
$h = 1/1024$	$11/8.6 \times 10^{-9}/0.1239/472.6$	$6/3.8 \times 10^{-9}/0.1239/798.1/1049.3$	$6/8.2 \times 10^{-9}/0.1239/26.1/256.4$
Example 2:	$\alpha = 0.1176$		
$h = 1/128$	$10/8.3 \times 10^{-9}/0.0522/4.2$	$5/1.5 \times 10^{-9}/0.0522/7.2/10.2$	$6/4.2 \times 10^{-9}/0.0552/2.9/4.9$
$h = 1/256$	$11/5.6 \times 10^{-9}/0.0582/22.9$	$6/2.6 \times 10^{-9}/0.0582/31.3/41.7$	$7/6.9 \times 10^{-9}/0.0582/3.0/17.9$
$h = 1/512$	$11/7.4 \times 10^{-9}/0.0615/108.3$	$7/5.4 \times 10^{-9}/0.0615/185.7/234.9$	$7/3.3 \times 10^{-9}/0.0615/12.5/79.1$
$h = 1/1024$	$11/3.1 \times 10^{-9}/0.0633/478.1$	$7/6.9 \times 10^{-9}/0.0633/550.1/943.1$	$7/5.0 \times 10^{-9}/0.0633/26.9/321.3$

5.1.1. h-independent convergence tests

One of the key properties of multigrid techniques is that their convergence does not depend on the number of grid points. Thus, in the first test we designed our experiments to investigate this property with Algorithms 2, 5, and 7, and to back up our theoretical results by LFA in Section 3.2. The number of multigrid steps (V-cycles) used to drop the mean of the relative residual below $\varepsilon_2 = 10^{-8}$, the relative reduction of dissimilarity, and runtimes (in seconds) are given in Table 1 with different sizes of grid points. The results show that all registration algorithms not only converge within a few multigrid steps, but they are also accurate because the dissimilarities between the reference and registered images have been reduced more than 87% for Example 1 and 94% for Example 2. For overall performance the experimental results suggest that Algorithm 7 would be preferred for practical applications because the multi-resolution idea used in the cooling process for α has been prove to be very useful for initialization.

5.1.2. Comparison with other multigrid methods

Methods in [4,12,18,21,22] are some existing unilevel or multigrid techniques used to solve the diffusion model.

Table 2
A comparison among different multigrid methods in [4,12,18,21,22] to solve the diffusion model in the first 20 iterations. The letters 'M', 'R', and 'D' mean the number of iterations in dropping the mean of the relative residuals resulting form (2) to 10^{-8}, the mean of the relative residuals, and the relative reduction of dissimilarity, respectively. Our proposed multigrid method in the last row is clearly the fastest in convergence.

Methods	M/R/D
Multi-resolution + AOS [4]	$20/7.1 \times 10^{-4}/0.1417$
Multi-resolution + FMG-V(2, 2) [21]	$20/5.9 \times 10^{-4}/0.1428$
Multi-resolution + Gauss–Newton + LMG-V(2, 2) [12,18]	$20/6.3 \times 10^{-7}/0.1344$
FAS-NMG-V(5, 5) + MSFP-FS smoother adapted from [8,9] (Section 2.4.3)	$20/2.6 \times 10^{-5}/0.1377$
FAS-NMG-V(5, 5) + MSFP-1 smoother adapted from [22,3] (Section 2.4.3)	$18/4.5 \times 10^{-9}/0.1351$
FAS-NMG-V(5, 5) + MSFP-2 smoother adapted from [43] (Section 2.4.3)	$18/7.6 \times 10^{-9}/0.1383$
FAS-NMG-V(5,5) + RFP smoother (26) (Section 3.1)	$\mathbf{11/2.0 \times 10^{-9}/0.1329}$

Table 3
Registration results of Algorithm 2 and AOS method [5] for Example 1 shown in Fig. 3(a)–(b). $*$ indicates either computation stopped after about 24 h or failure in dropping the relative residual to 10^{-8} in 10,000 iterations. Note that a much smaller residual tolerance of 10^{-1} is practically needed; here we test for convergence.

	Algorithm 2 M/R/D/C	AOS M/R/D/C
$h = 1/128$	$10/8.6 \times 10^{-9}/0.1248/4.2$ (0.07 min)	$10000/*/0.1248/934.1$ (>15 min)
$h = 1/256$	$11/2.0 \times 10^{-9}/0.1329/23.1$ (0.38 min)	$10000/*/0.1329/5500.8$ (>1.5 h)
$h = 1/512$	$11/9.4 \times 10^{-9}/0.1383/105.2$ (1.75 min)	$10000/*/0.1383/2498.3$ (>6.9 h)
$h = 1/1024$	$11/4.7 \times 10^{-9}/0.1404/427.6$ (7.96 min)	$*/*/*/*$ (>24 h)

In this section, we took Examples 1 to illustrate a comparison among our FAS-NMG method with Algorithm 2 and other six multigrid methods by starting with the fixed parameters $h = 1/256$, $\alpha = 1/8$ and $\mathbf{u}^{(0)} = 0$. Here we used $\tau = 10^{-2}$ and applied the so-called *multi-resolution* technique with the gradient descent methods of [4,21] for fairness. The FMG or LMG methods in [12,18,21] were performed using two pre-smoothing and two post-smoothing steps with the pointwise GS smoothers until the mean of relative residuals below a user-supplied threshold ($tol_{\text{LMG}} = 10^{-4}$).

Table 2 summarizes the results for all multigrid methods. As expected from the experiments, all methods are very fast and accurate in registering the given images because the dissimilarities between the reference and registered images have been reduced more than 80% within the first 20 iterations. However our method is not only the fastest way in solving the problem, but also in dropping the mean of the relative residuals below 10^{-8}.

5.1.3. Comparison with the AOS method [4]

The AOS method is one of the most widely used gradient descent techniques for diffusion image registration [4,19,21]. We take Example 1 to illustrate a comparison with our FAS-NMG method. Table 3 summarizes the results for Algorithm 2 and AOS methods with different numbers of grid points. We used Algorithm 2 with $\alpha = 1/8$ for fairness (as Algorithm 7 will be even better). That is, we started both methods with the same α and the same initial guess, $\mathbf{u}^{(0)} = 0$. For the AOS method, the time-step τ is required to be sufficiently small for each size of the problem. We used $\tau = 10^{-2}$ for $h = 1/128 - 1/1024$. As expected from the experiments, both methods are very accurate in registering the given images because the dissimilarities between the reference and registered images have been reduced more than 85%. However the AOS fails to drop the relative residual to 10^{-8} in a few time steps (even for $\tau > 10^{-2}$) and the runtimes used by the RDR method is much faster than those of the AOS technique in delivering the same level of the relative dissimilarity.

5.1.4. Comparison of multigrid methods with different smoothers

Our aim in this section is to show that we have proposed a robust smoother for the FAS-NMG technique in solving the discrete system represented in (14). To end this, we have conducted several experiments of our FAS multigrid with different kinds of smoothers:

(i) the proposed smoother based on the RFP method (26) represented by Algorithm 3 in Section 3
(ii) the GS smoother based on the SFP method given by (19) in Section 2.4.3 (the standard FP method defined in [3, p. 79] with the standard linear (inner) solver)
(iii) the GS smoother based on the MSFP-FS method given by (20) and (21) in Section 2.4.3 (a modified SFP method adapted from the numerical techniques of Frohn-Schauf et al. [8,9] with the standard linear solver)
(iv) the GS smoother based on MSFP-1 method given by (22) in Section 2.4.3 (a modified SFP method of the first type with the standard linear solver)
(v) the GS smoother based on MSFP-2 method given by (23) in Section 2.4.3 (a modified SFP method of the second type with the standard linear solver)

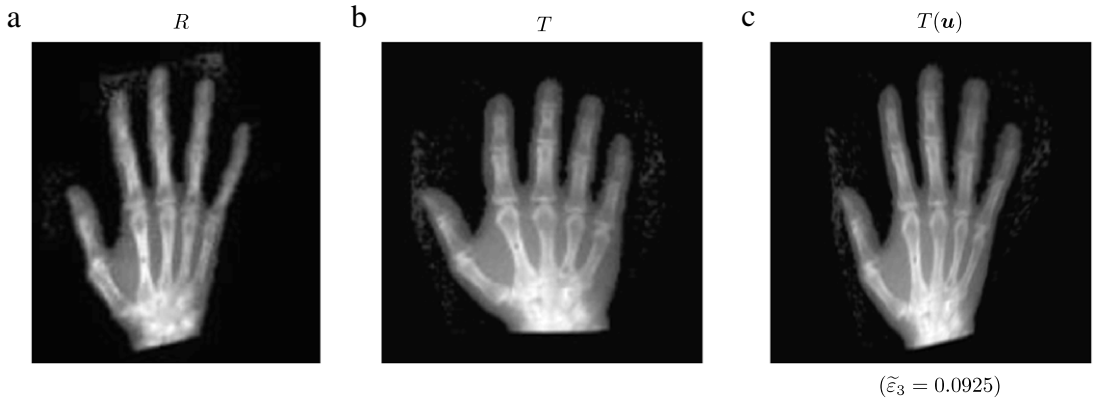

Fig. 4. Results from Example 1 as shown in (a)–(b) by the curvature model (47) using the FAS-NMG method with the smoother (52). Left to right: the reference R, the template T, and the registered image $T(\mathbf{u})$ by the curvature model (47).

(vi) line relaxation smoothers based on the RFP method in Section 3 (the RFP method given by (26) with another kind of iterative linear solvers)
(vii) the Newton–Gauss–Seidel smoother used in [10] and given explicitly in [20] in (13) (an alternative way of choice of smoothers used in solving the discrete Euler–Lagrange equations represented by (14)).

Omitting the computational results, we remark that these observations can be made:

(a) The smoother in (ii) requires more multigrid cycles than the proposed smoother and may not lead to the convergence of the FAS-NMG technique for small values of α.
(b) As expected, the results based on the smoother (iii) do not find a solution that is the necessary condition of the original variational problem (2), although the underlying MG performs better than with other smoothers (slightly less well than with our RFP smoother).
(c) The smoother in (iv)–(v) may take many multigrid cycles, not leading to the convergence of the FAS-NMG technique when the fixed point parameters c_1, c_2 (or c), and ϵ are not well selected (i.e. the NMG becomes sensitive to these parameters).
(d) As expected, line relaxation smoothers require less multigrid cycles but more computational costs than the proposed smoother.
(e) The Newton–Gauss–Seidel smoother provide well matched images within a few multigrid steps, but it may require more multigrid cycles than the proposed smoother in leading to the convergence of the FAS-NMG technique.

5.2. The curvature model

In this section, we aim to show that the FAS-NMG method with the smoother (52) based on our linearization idea (25) is effective and robust to solve the curvature model (47) within the multigrid framework similar to Algorithm 1.

To this end, we took only one data set of medical images, Examples 1, as shown in Fig. 3(a)–(b). We first investigate the convergence behavior and the registration accuracy of our FAS-NMG method with different sizes of image resolutions. Second we compare three numerical solution methods for solving the curvature model (47). In all experiments, $\Omega = [0, 1]^2$, $V = [0, 1]$, $\nu_1 = 10$, $\nu_2 = 10$, *PCGSiter* $= 10$, and $\omega = 0.9725$ were used.

5.2.1. h-independent convergence tests

In this test, we started the registration processes with $\alpha = 10^{-4}$, $\mathbf{u}^{(0)} = 0$, and $h = 1/128, \ldots, h = 1/1024$. The registered image is shown in Fig. 4(c). As expected from our LFA results, Fig. 5(a)–(b) shows that our FAS-NMG approach is h-independent. It takes only a few MG steps (almost the same number) to drop the mean of the relative residuals below 10^{-8}. Moreover, as shown in Fig. 5(b) only one MG step can reduce the dissimilarities between the reference and the registered image more than 90% for the given registration problem.

5.2.2. Comparison with two other methods

In this section we aim to investigate the performance of the FAS-NMG method with the smoother (52) and those of other two methods: the FP method (52) (the smoother used as a stand-alone solver) and the DCT-based method in [6]. We note that the FT-based method in [33,34] is faster than the DCT-based method with the ratio of 2.3 for solving each linear time-dependent problem resulting from (47). Thus, for this class of gradient descent techniques it is enough to use only the DCT-based method, which is one of the most widely used techniques for the curvature model.

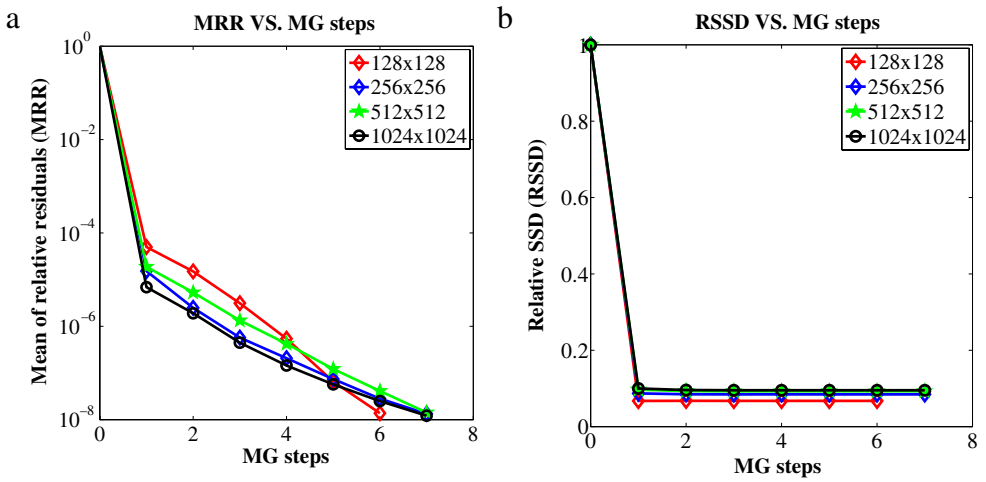

Fig. 5. Results from Example 1 as shown by Fig. 3(a)–(b) by the curvature model (47) using the FAS-NMG method with the smoother (52). Left to right: the histories of the mean of relative residuals (MRR) with respect to the MG steps and the histories of the relative SSD (RSSD) with respect to the MG steps.

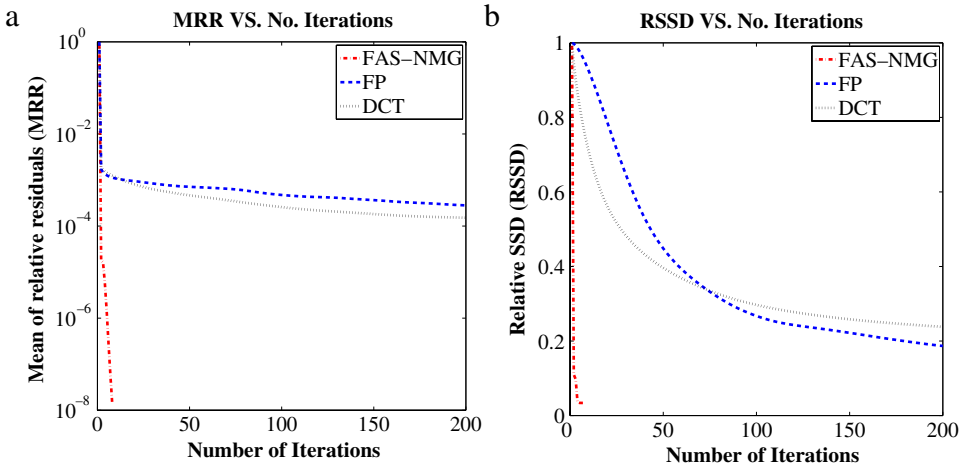

Fig. 6. Results from Example 1 as shown by Fig. 3(a)–(b) by the curvature model (47) using three numerical solution methods: the FAS-NMG method with the smoother (52), the FP method (52), and the DCT-based method in [6]. Left to right: (a) the histories of the mean of relative residuals (MRR) with respect to the iteration steps and (b) the histories of the relative SSD (RSSD) with respect to the iteration steps.

To this end, we started all methods with $h = 1/256$, $\alpha = 10^{-4}$ and $\mathbf{u}^{(0)} = 0$. For the DCT-based method, the time-step τ was selected to be $\tau = 10^{-2}$ (since the fixed parameters $1/h^4 = N^4 = 256^4$, $V = [0, 1]$ and $\alpha = 10^{-4}$ were used in the discrete system of (47), $\tau = 10^{-2}$ was a reasonable time-step).

Compared with the results by other two methods, Fig. 6(b) shows that our FAS-NMG method is very fast in reducing the dissimilarities between the reference and registered images. Moreover, as shown in Fig. 6(a) it takes only a few steps to drop the mean of the relative residuals (MRR) below 10^{-8}. This is a remarkable result to conclude that its performance in solving the curvature model (47) is much more efficient than those of the other two methods.

From both tests in Sections 5.1 and 5.2 they confirm that the smoothers based on our linearization idea by (25) are effective and robust in leading to convergent multigrid methods for both the diffusion and curvature models.

6. Conclusions

In this paper we first reviewed various iterative methods for solving image registration models and then addressed the problem of designing an optimal and efficient FAS-NMG technique. For the commonly used SSD model, we proposed a new, simple and yet effective fixed point smoothing scheme which is demonstrated to be more effective than a large class of other iterative methods for a diffusion model and a curvature model. A local Fourier analysis was done to confirm the effectiveness of the smoother. For an automatic way of selecting the regularization parameter α, we proposed a multi-resolution idea that seems to work well for a range of problems. Numerical results confirmed the expected robustness and excellent speed in

convergence compared with other competing methods. Future work will address multigrid techniques for models of multi-modal deformable image registration.

References

[1] J. Maintz, M. Viergever, A survey of medical image registration, Med. Image Analysis 2 (1) (1998) 1–36.
[2] J. Hajnal, D. Hill, D. Hawkes, Medical Image Registration, in: The Biomedical Engineering Series, CRC Press, 2001.
[3] J. Modersitzki, Numerical Methods for Image Registration, Oxford, 2004.
[4] B. Fischer, J. Modersitzki, Fast diffusion registration, Contemporary Mathematics 313 (2002) 117–129.
[5] B. Fischer, J. Modersitzki, Curvature based image registration, Journal of Mathematical Imaging and Vision 18 (2003) 81–85.
[6] B. Fischer, J. Modersitzki, A unified approach to fast image registration and a new curvature based registration technique, Linear Algebra and its Applications 380 (2004) 107–124.
[7] B. Fischer, J. Modersitzki, Ill-posed medicine—an introduction to image registration, Inverse Problems 24 (2008) 1–19.
[8] C.F. -Schauf, S. Henn, L. Hömke, K. Witsch, Total variation based image registration, in: X.-C. Tai, K.-A. Lie, T.F. Chan, S. Osher (Eds.), Proceedings of the International Conference on PDE-Based Image Processing and Related Inverse Problems Series: Mathematics and Visualization, Springer Verlag, 2006, pp. 305–323.
[9] C.F. -Schauf, S. Henn, K. Witsch, Multigrid based total variation image registration, Comput. Visual. Sci. 11 (2008) 101–113.
[10] S. Gao, L. Zhang, H. Wang, R. de Crevoisier, D. Kuban, R. Mohan, L. Dong, A deformable image registration method to handle distended rectums in prostate cancer radiotherapy, Medical Physics 33 (9) (2006) 3304–3312.
[11] E. Haber, J. Modersitzki, Numerical solutions for volumn preserving image registration, Inverse Problems 20 (2004) 1621–1638.
[12] E. Haber, J. Modersitzki, A multilevel method for image registration, SIAM Journal of Scientific Computing 27 (5) (2006) 1594–1607.
[13] E. Haber, S. Heldmann, J. Modersitzki, Adaptive mesh refinement for non parametric image registration, SIAM Journal of Scientific Computing 30 (6) (2008) 3012–3027.
[14] S. Henn, K. Witsch, Iterative multigrid regularization techniques for image matching, SIAM Journal of Scientific Computing 23 (4) (2001) 1077–1093.
[15] S. Henn, K. Witsch, Image registration based on multiscale energy information, Multiscale Modeling and Simulation 4 (2) (2005) 584–609.
[16] S. Henn, A multigrid method for a fourth-order diffusion equation with application to image processing, SIAM Journal of Scientific Computing 27 (3) (2005) 831–849.
[17] L. Hömke, A multigrid method for anisotropic PDEs in elastic image registration, Numerical Linear Algebra with Applications 13 (2006) 215–229.
[18] H. Köstler, K. Ruhnau, R. Wienands, Multigrid solution of the optical flow system using a combined diffusion- and curvature-based regularizer, Numerical Linear Algebra with Applications 15 (2008) 201–218.
[19] J.L. -Ruiz, J.M. -Sanchez, Optimal parameters selection for non-parametric image registration methods, LNCS 4179 (2006) 564–575.
[20] W. Lu, M.L. Chen, G.H. Olivera, K.J. Ruchala, T.R. Mackie, Fast free-form deformable registration via calculus of variations, Physics in Medicine and Biology 49 (2004) 3067–3087.
[21] M. Stürmer, H. Köstler, U. Rüde, A fast full multigrid solver for applications in image processing, Numerical Linear Algebra with Applications 15 (2008) 187–200.
[22] D. Zikic, W. Wein, A. Khamene, Fast deformable registration of 3D-ultrasound data using a variational approach, LNCS 4190 (2006) 915–923.
[23] N. Chumchob, K. Chen, A variational approach for discontinuity-preserving image registration, East–West Journal of Mathematics Special volume 2010 (2010) 266–282.
[24] N. Chumchob, K. Chen, C. Brito, A fourth order variational image registration model and its fast multigrid algorithm, SIAM Journal of Multiscale Modeling and Simulation 9 (2011) 89–128.
[25] L. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D 60 (1992) 259–268.
[26] T. Chan, J. Shen, Image Processing and Analysis — Variational, PDE, Wavelet, and Stochastic Methods, SIAM Publications, Philadelphia, USA, 2005.
[27] T. Chan, K. Chen, J. Carter, Iterative methods for solving the dual formulation arising from image restoration, Electronic Transactions on Numerical Analysis 26 (2007) 299–311.
[28] K. Chen, X.-C. Tai, A nonlinear multigrid method for total variation minimization from image restoration, Journal of Scientific Computing 32 (2) (2007) 115–138.
[29] A. Brandt, Multi-level adaptive solutions to BVPs, Mathematial Computing 31 (1977) 333–390.
[30] U. Trottenberg, C. Oosterlee, A. Schuller, Multigrid, Academic Press, 2001.
[31] A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, C. Schnörr, Variational optical flow computation in real time, IEEE Transactions on Image Processing 14 (5) (2005) 608–615.
[32] A. Bruhn, J. Weickert, Multigrid platform for real-time motion computation with discontinuity-preserving variational methods, International Journal of Computer Vision 70 (3) (2006) 257–277.
[33] J. Larrey-Ruiz, R. Verdú-Monedero, J. Morales-Sánchez, A fourier domain framework for variational image registration, Journal of Mathematical Imaging and Vision 32 (2008) 57–72.
[34] R. Verdú-Monedero, J. Larrey-Ruiz, Frequency implementation of the Euler–Lagrange equations for variational image registration, IEEE Signal Processing Letters 15 (2008) 321–324.
[35] T. Lu, P. Neittaanmäki, X.-C. Tai, A parallel splitting up method and its application to Navier–Stokes equations, Applied Mathematical Letters 4 (2) (1991) 25–29.
[36] J. Weickert, B. ter Haar Romeny, M. Viergever, Effcient and reliable schemes for nonlinear diffusion filtering, IEEE Transactions on Image Processing 7 (1998) 398–410.
[37] W. Briggs, V. Henson, S. McCormick, A Multigrid Tutorial, 2nd ed., SIAM Publications, Philadelphia, USA, 2000.
[38] W. Hackbusch, Multi-Grid Methods and Applications, Springer-Verlag, Berlin, Heidelberg, New York, 1985.
[39] P. Wesseling, Multigrid Methods, Edwards, Philadelphis, PA, USA, 2004.
[40] R. Wienands, W. Joppich, Practical Fourier Analysis for Multigrid Method, Chapman & Hall/CRC, USA, 2005.
[41] K. Chen, Matrix Preconditioning Techniques and Applications, Cambridge University Press, UK, 2005.
[42] N. Badshah, K. Chen, Multigrid method for the Chan–Vese model in variational segmentation, Communications in Computational Physics 4 (2) (2008) 294–316.
[43] C. Brito-Loeza, K. Chen, Multigrid method for a modified curvature driven diffusion model for image inpainting, Journal of Computational Mathematics 26 (6) (2008) 856–875.
[44] T. Chan, K. Chen, On a nonlinear multigrid algorithm with primal relaxation for the image total variation minimization, Journal of Numerical Algorithms 41 (2006) 387–411.
[45] C.F. -Schauf, S. Henn, K. Witsch, Nonlinear multigrid methods for total variation image denoising, Comput. Visual. Science 7 (2004) 199–206.
[46] J. Savage, K. Chen, An improved and accelerated nonlinear multigrid method for total-variation denoising, International Journal of Computer Mathematics 82 (8) (2005) 1001–1015.
[47] J. Savage, K. Chen, On multigrids for solving a class of improved total variation based staircasing reduction models, in: X.-C. Tai, K.-A. Lie, T.F. Chan, S. Osher (Eds.), Proceedings of the International Conference on PDE-Based Image Processing and Related Inverse Problems Series: Mathematics and Visualization, Springer Verlag, 2006, pp. 69–94.
[48] A. Bruhn, J. Weickert, Towards Ultimate Motion Estimation: Combining Highest Accuracy with Real-Time Performance, in: Proceeding of 10th International Conference on Computer Vision ICCV 2005, vol. 1, IEEE Computer Society Press, Beijing, People's Republic of China, 2005, pp. 749–755.

[49] W. Rheinboldt, Methods for Solving Systems of Nonlinear Equations, 2nd ed., SIAM Publications, Philadelphia, USA, 1998.
[50] Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd ed., SIAM Publications, Philadelphia, USA, 2003.
[51] N. Badshah, K. Chen, On two multigrid algorithms for modelling variational multiphase image segmentation, IEEE Transactions on Image Processing 18 (5) (2009) 1097–1106.
[52] T. Chan, K. Chen, An optimization-based multilevel algorithm for total variation image denoising, Multiscale Mod. Simu. 5 (2) (2006) 615–645.
[53] E. Haber, R. Horesh, J. Modersitzki, Numerical optimization for constrained image registration, Numerical Linear Algebra with Applications 17 (2010) 343–359.
[54] S. Hamilton, M. Benzi, E. Haber, New multigrid smoothers for the oseen problem, Numerical Linear Algebra with Applications 17 (2010) 556–576.
[55] B. Seynaeve, E. Rosseel, B. Nicolaï, S. Vandewalle, Fourier mode analysis of multigrid methods for partial differential equations with random coefficients, Journal of Computational Physics 224 (2007) 132–149.
[56] J. Chen, E. Haber, D. Oldenburg, Three-dimensional numerical modelling and inversion of magnetometric resistivity data, Geophysical Journal of Inter. 149 (3) (2002) 679–697.
[57] N. Chumchob, K. Chen, A robust affine image registration method, International Journal of Numerical Analysis and Modeling 6 (2) (2009) 311–334.
[58] E. Haber, D. Oldenburg, A GCV based method for nonlinear ill-posed problems, Computational Geosciences 4 (1) (2000) 41–63.
[59] P. Lelièvre, D. Oldenburg, Magnetic forward modelling and inversion for high susceptibility, Geophysical Journal of Inter. 166 (1) (2006) 76–90.
[60] G.A. Newman, G.M. Hoversten, Solution strategies for two- and three-dimensional electromagnetic inverse problems, Inverse Problems 16 (2000) 1357–1375.