

A FOURTH ORDER VARIATIONAL IMAGE REGISTRATION MODEL AND ITS FAST MULTIGRID ALGORITHM

N. CHUMCHOB[†]¶, K. CHEN[‡]¶, AND C. BRITO-LOEZA[¶]

Abstract. Several partial differential equations (PDEs) based variational methods can be used for deformable image registration, mainly differing in how regularization for deformation fields is imposed [44]. On one hand for smooth problems, models of elastic-, diffusion-, and fluid-image registration are known to generate globally smooth and satisfactory deformation fields. On the other hand for non-smooth problems, models based on the total variation (TV) regularization are better for preserving discontinuities of the deformation fields. It is a challenge to design a deformation model suitable for both smooth and non-smooth deformation problems.

One promising model that is based on a curvature type regularizer and appears to deliver excellent results for both problems is proposed and studied in this paper. A related work due to Fischer and Modersitzki [21] and then refined by Henn and Witsch [37] used an approximation of the mean curvature and obtained improved results over previous models. However, this paper investigates the full curvature model and finds that the new model is more robust than approximated curvature models and leads to further improvement.

Associated with the new model is the apparent difficulty in developing a fast algorithm as the system of two coupled PDEs is highly nonlinear and of fourth order so standard application of multigrid methods does not work. In this paper, we first propose several fixed-point type smoothers. Then we use both local Fourier analysis and experiments to select the most effective smoother which turns out to be a primal-dual based method. Finally we use the recommended smoother to propose a nonlinear multigrid algorithm for the new model. Numerical tests using both synthetic and realistic images not only confirm that the proposed curvature model is more robust in registration quality for a wide range of applications than previous work [44, 37], but also that the proposed algorithm is fast and accurate in delivering visually-pleasing registration results.

Key words. Deformable image registration, Nonlinear multigrid method, Variational 4th order PDEs, Regularization, Curvature.

AMS subject classifications.

1. Introduction. One of the major problems of current research and applications in image processing is *image registration*. 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, 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 [43, 31, 44], and references therein. This work focuses on an improved deformable image registration model in a variational formulation.

The variational framework. Variational PDEs-based image registration models have been successfully proven to be very valuable tools in several applications,

[†]Department of Mathematics, Faculty of Science, Silpakorn University, Nakorn Pathom 73000, Thailand and Centre of Excellence in Mathematics, CHE, Si Ayutthaya Rd., Bangkok 10400, Thailand. Email: cnoppado@su.ac.th

[‡]Correspondence author. Email: k.chen@liv.ac.uk

[¶]All authors are with the Centre for Mathematical Imaging Techniques, Department of Mathematical Sciences, The University of Liverpool, Peach Street, Liverpool L69 7ZL, United Kingdom. Web: <http://www.liv.ac.uk/~cmchenke/cmit/>

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

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

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

$$\mathbf{u} : \mathbb{R}^d \rightarrow \mathbb{R}^d, \quad \mathbf{u} : \mathbf{x} \mapsto \mathbf{u}(\mathbf{x}) = (u_1(\mathbf{x}), u_2(\mathbf{x}), \dots, u_d(\mathbf{x}))^\top.$$

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 [44]. It is worth noticing that the displacement \mathbf{u} is more intuitive than the transformation φ because it can measure 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 $d = 2$ (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

$$(1.1) \quad \min_{\mathbf{u}} \left\{ \mathcal{D}(\mathbf{u}) = \frac{1}{2} \int_{\Omega} (T(\mathbf{x} + \mathbf{u}(\mathbf{x})) - R(\mathbf{x}))^2 d\mathbf{x} \right\}.$$

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. We shall adopt the Tikhonov regularization. As a consequence, the image registration problem can be posed as a minimization problem of the *joint* energy functional given by

$$(1.2) \quad \min_{\mathbf{u}} \{ \mathcal{J}_{\alpha}[\mathbf{u}] = \mathcal{D}(\mathbf{u}) + \alpha \mathcal{R}(\mathbf{u}) \},$$

where the regularizer \mathcal{R} will be discussed shortly. Here $\alpha > 0$ is the regularization parameter that compromises similarity and regularity, and \mathbf{u} is searched over a set \mathcal{U} of admissible functions minimizing \mathcal{J}_{α} . The set \mathcal{U} is generally assumed to be a linear subspace of a Hilbert space \mathcal{H} equipped with its usual scalar product $\langle \mathbf{u}, \boldsymbol{\eta} \rangle_{\mathcal{H}} = \int_{\Omega} \mathbf{u}(\mathbf{x}) \cdot \boldsymbol{\eta}(\mathbf{x}) d\mathbf{x} = \int_{\Omega} \langle \mathbf{u}(\mathbf{x}), \boldsymbol{\eta}(\mathbf{x}) \rangle_{\mathbb{R}^2} d\mathbf{x}$ where $\int_{\Omega} \langle \cdot, \cdot \rangle_{\mathbb{R}^2}$ denotes the Euclidean scalar product. The first variation of \mathcal{D} is given by

$$(1.3) \quad \delta \mathcal{D}(\mathbf{u}; \boldsymbol{\eta}) = \langle \mathbf{f}(\mathbf{u}), \boldsymbol{\eta} \rangle_{\mathcal{H}}$$

for all variational directions $\boldsymbol{\eta} \in \mathcal{H}$ where

$$(1.4) \quad \mathbf{f}(\mathbf{u}) = (f_1(\mathbf{u}), f_2(\mathbf{u}))^\top = (T(\mathbf{u}) - R) \nabla_{\mathbf{u}} T(\mathbf{u}),$$

is a nonlinear function that will be much used later. Here we denote by $\nabla_{\mathbf{u}} F = (\partial F / \partial u_1, \partial F / \partial u_2)^\top$ the gradient of a functional F with respect to $\mathbf{u}(\mathbf{x})$ to distinguish from the usual gradient $\nabla F = (\partial F / \partial x_1, \partial F / \partial x_2)^\top$. In what follows, we also use the notation $\partial_{x_\ell} F = \frac{\partial F}{\partial x_\ell}$ and $\partial_{x_1 x_2} F = \frac{\partial^2 F}{\partial x_1 \partial x_2}$.

Review of five PDE-based image registration models. Non-surprisingly, the choice of the deformation regularizer \mathcal{R} is very crucial for effective registration. Different choices of \mathcal{R} lead not only to different deformation fields but also to different Euler-Lagrange systems of coupled nonlinear PDEs. Below we review the specific choice of \mathcal{R} and the subsequent system in five commonly used variational models.

- 1) *Elastic image registration*: Choosing \mathcal{R} in (1.2) based on the linearized elastic potential of the deformation field as given by

$$(1.5) \quad \mathcal{R}^{\text{elas}}(\mathbf{u}) = \int_{\Omega} ((\mu/4) \sum_{l,m=1}^2 (\partial_{x_l} u_m + \partial_{x_m} u_l)^2 + (\lambda/2)(\nabla \cdot \mathbf{u})^2) d\mathbf{x},$$

leads to the Euler-Lagrange system of two second-order nonlinear PDEs:

$$(1.6) \quad \begin{cases} f_1(\mathbf{u}) - \alpha((\lambda + 2\mu)\partial_{x_1 x_1} u_1 + \mu\partial_{x_2 x_2} u_1 + (\lambda + \mu)\partial_{x_1 x_2} u_2) = 0 \\ f_2(\mathbf{u}) - \alpha((\lambda + \mu)\partial_{x_1 x_2} u_1 + \mu\partial_{x_1 x_1} u_2 + (\lambda + 2\mu)\partial_{x_2 x_2} u_2) = 0 \end{cases}$$

(elastic model)

subject to $\langle \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^\top) + \lambda \text{diag}(\nabla \cdot \mathbf{u}), \mathbf{n} \rangle_{\mathbb{R}^2} = 0$ on $\partial\Omega$. Here $\mu > 0$ and $\lambda \geq 0$ are the so-called Lamé constants which reflect material properties and $\mathbf{n} = (n_{x_1}, n_{x_2})^\top$ is the outward unit normal to the image boundary $\partial\Omega$. This model assumes that the displacement field \mathbf{u} is relatively small; see more details in [44] and references therein.

- 2) *Diffusion image registration* [17, 20, 40, 41, 44, 49]: Choosing \mathcal{R} in (1.2) based on the L^2 norm of ∇u_l as given by

$$(1.7) \quad \mathcal{R}^{\text{diff}}(\mathbf{u}) = \frac{1}{2} \sum_{l=1}^2 \int_{\Omega} |\nabla u_l|^2 d\mathbf{x},$$

leads to the Euler-Lagrange system of two second-order nonlinear PDEs also:

$$(1.8) \quad \begin{cases} f_1(\mathbf{u}) - \alpha \Delta u_1 = 0 \\ f_2(\mathbf{u}) - \alpha \Delta u_2 = 0 \end{cases} \quad (\text{diffusion model})$$

subject to $\langle \nabla u_l, \mathbf{n} \rangle_{\mathbb{R}^2} = 0$ on $\partial\Omega$. We note that this registration model can be viewed as a special case of the elastic model when non-physical parameters, $\mu = 1$ and $\lambda = -1$, are taken (as the physical $\lambda \geq 0$). Practically this model can be solved effectively as the operators are simple.

- 3) *Fischer-Modersitzki's curvature image registration* [21, 22, 23, 40, 41, 44]: Choosing \mathcal{R} in (1.2) based on an approximation of the mean curvature of the surface of u_l given by

$$(1.9) \quad \mathcal{R}^{\text{FMcurv}}(\mathbf{u}) = \frac{1}{2} \sum_{l=1}^2 \int_{\Omega} (\hat{\kappa}_M(u_l))^2 d\mathbf{x} = \frac{1}{2} \sum_{l=1}^2 \int_{\Omega} (\Delta u_l)^2 d\mathbf{x},$$

leads to the Euler-Lagrange system of two fourth-order nonlinear PDEs:

$$(1.10) \quad \begin{cases} f_1(\mathbf{u}) + \alpha \Delta^2 u_1 = 0 \\ f_2(\mathbf{u}) + \alpha \Delta^2 u_2 = 0 \end{cases} \quad (\text{Fischer-Modersitzki's curvature model})$$

subject to the special boundary conditions $\nabla u_l = 0$, $\nabla \Delta u_l = 0$ on $\partial\Omega$, for $l = 1, 2$. 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

$$(1.11) \quad \begin{aligned} \kappa_M(u_l) &= \nabla \cdot \frac{\nabla u_l}{\sqrt{1+|\nabla u_l|^2}} \\ &= \frac{(1+u_{l_{x_1}}^2)u_{l_{x_1x_1}} - 2u_{l_{x_1}}u_{l_{x_2}}u_{l_{x_1x_2}} + (1+u_{l_{x_2}}^2)u_{l_{x_2x_2}}}{(1+u_{l_{x_1}}^2+u_{l_{x_2}}^2)^{3/2}}. \end{aligned}$$

Observe that assuming that $|\nabla u_l| \approx 0$ yields $\kappa_M(u_l) \approx \widehat{\kappa}_M(u_l) = \Delta u_l$. Thus the above model uses an approximation of the surface curvature in its regularizer \mathcal{R} .

- 4) *Henn-Witsch's curvature image registration* [38, 37, 33, 35, 34]. Choosing \mathcal{R} based on an approximation of the sum of the squared principal curvatures $\kappa_{P_1}(u_l)$ and $\kappa_{P_2}(u_l)$ of the surface of u_l as well as an approximation of the Gaussian curvature $K_G(u_l)$ given by

$$(1.12) \quad \begin{aligned} \mathcal{R}^{\text{HWcurv}}(\mathbf{u}) &= \frac{1}{2} \sum_{l=1}^2 \int_{\Omega} ((\widehat{\kappa}_M(u_l))^2 - 2\widehat{K}_G(u_l)) d\mathbf{x} \\ &= \frac{1}{2} \sum_{l=1}^2 \int_{\Omega} (\Delta u_l)^2 - 2(u_{l_{x_1x_1}}u_{l_{x_2x_2}} - u_{l_{x_1x_2}}^2) d\mathbf{x}, \end{aligned}$$

leads to the Euler-Lagrange system of two modified fourth-order nonlinear PDEs:

$$(1.13) \quad \begin{cases} f_1(\mathbf{u}) + \alpha \Delta^2 u_1 = 0 \\ f_2(\mathbf{u}) + \alpha \Delta^2 u_2 = 0 \end{cases} \quad (\text{Henn-Witsch's curvature model})$$

subject to $B_l(u_l) = 0$ on $\partial\Omega$ with

$$B_1(u_l) = -\frac{\partial}{\partial \mathbf{n}} \Delta u_l - \frac{\partial}{\partial \mathbf{s}} \left[\frac{\partial^2 u_l}{\partial x_1 \partial x_2} (n_1^2 - n_2^2) + \left(\frac{\partial^2 u_l}{\partial^2 x_2} - \frac{\partial^2 u_l}{\partial^2 x_1} \right) n_{x_1} n_{x_2} \right],$$

and $B_2(u_l) = \frac{\partial^2 u_l}{\partial \mathbf{n}^2}$, where \mathbf{s} denotes the unit tangential vector (orthogonal to \mathbf{n}). To see how the curvatures are approximated in (1.12), we see that $|\nabla u_l| \approx 0$ leading to

$$\begin{aligned} \kappa_{P_1}^2(u_l) + \kappa_{P_2}^2(u_l) &= (\kappa_{P_1}(u_l) + \kappa_{P_2}(u_l))^2 - 2\kappa_{P_1}(u_l)\kappa_{P_2}(u_l) \\ &= (\kappa_M(u_l))^2 - 2K_G(u_l) \\ &\equiv \left(\nabla \cdot \left(\frac{\nabla u_l}{\sqrt{1+|\nabla u_l|^2}} \right) \right)^2 - 2 \left(\frac{u_{l_{x_1x_1}}u_{l_{x_2x_2}} - u_{l_{x_1x_2}}^2}{(1+|\nabla u_l|^2)^2} \right) \\ &\approx (\widehat{\kappa}_M(u_l))^2 - 2\widehat{K}_G(u_l) \\ &\equiv (\Delta u_l)^2 - 2(u_{l_{x_1x_1}}u_{l_{x_2x_2}} - u_{l_{x_1x_2}}^2). \end{aligned}$$

- 5) *Total variation (TV) image registration* [24, 25, 54]: Choosing \mathcal{R} based on the TV semi-norm of ∇u_l given by

$$(1.14) \quad \mathcal{R}^{\beta\text{TV}}(\mathbf{u}) = \sum_{l=1}^2 \int_{\Omega} |\nabla u_l|_{\beta} d\mathbf{x} = \sum_{l=1}^2 \int_{\Omega} \sqrt{u_{l_{x_1}}^2 + u_{l_{x_2}}^2} + \beta d\mathbf{x},$$

leads to the Euler-Lagrange system of two second-order nonlinear PDEs:

$$(1.15) \quad \begin{cases} f_1(\mathbf{u}) - \alpha \nabla \cdot \left(\frac{\nabla u_1}{|\nabla u_1|_{\beta}} \right) = 0 \\ f_2(\mathbf{u}) - \alpha \nabla \cdot \left(\frac{\nabla u_2}{|\nabla u_2|_{\beta}} \right) = 0 \end{cases} \quad (\text{the TV model})$$

subject to $\langle \nabla u_l, \mathbf{n} \rangle_{\mathbb{R}^2} = 0$ on $\partial\Omega$. Here $\beta > 0$ is a small real parameter for avoiding singularities when $|\nabla u_l| = 0$; see more details in [24, 25, 46]. Note that each displacement variable u_1 and u_2 is regularized separately in (1.14) and therefore the decoupling of the nonlinear diffusion processes can be clearly seen from (1.15). We also note that the rotational invariance of (1.14) is missing because it is the non-vectorial regularization. Hence non-smooth registration problems with non-axis-aligned discontinuities are difficult to solve with this model; see Figures 1.1–1.2 for an axis-aligned problem and 1.7–1.8 for a non-axis-aligned problem. Occasionally a pre-aligning step may be used to improve results for this type of non-smooth registration problems.

Below we shall use the notation $\mathcal{R}^{\text{elas}}$, $\mathcal{R}^{\text{diff}}$, $\mathcal{R}^{\text{FMcurv}}$, $\mathcal{R}^{\text{HWcurv}}$ and $\mathcal{R}^{\beta\text{TV}}$ to mean respectively the underlying models as shown. Here the first four models are quite different from the fifth one.

Firstly, $\mathcal{R}^{\text{elas}}$, $\mathcal{R}^{\text{diff}}$, $\mathcal{R}^{\text{FMcurv}}$, and $\mathcal{R}^{\text{HWcurv}}$ produce globally smooth deformation fields, although the latter two models are better than the former two. While they are useful for several applications, they become poor if discontinuities or steep gradients in the deformation fields are expected (e.g. resulting from matching several moved objects or partially occluded objects). See Figures 1.1–1.2 for a particular registration problem where these regularization techniques yield oversmooth deformation fields.

Secondly, $\mathcal{R}^{\beta\text{TV}}$ helps to preserve discontinuities of the deformation field in clear contrast to the first four models; see Figures 1.3–1.4 for example, in particular the piecewise smoothness shown in Figure 1.4 (c) at the top region. However, $\mathcal{R}^{\beta\text{TV}}$ may not be suitable for smooth registration problems, which are modeled better with the first four methods.

In addition to these 5 models, the optical flow model [1, 2] is also widely used which works the best if features have minor changes from R to T e.g. in matching sequential frames in a video.

Review of numerical techniques for deformable image registration. Efficient solution of the coupled nonlinear PDEs resulting from a variational registration model is an important task. Various numerical techniques have been proposed and tested, which may be broadly divided into two main categories: the so-called *parabolic* and *elliptic* approaches. A parabolic approach (gradient descent or time marching) introduces an artificial time variable and then determines the steady state solution of the system of time-dependent linear PDEs (see e.g. [20, 21, 33, 41, 44, 49]), often making use of the convenient explicit time marching schemes. Unfortunately such an approach is often quite slow. The elliptic approach solves the PDEs directly, with the common choice being the fixed-point (FP) iteration, Newton-type methods, multigrid (MG) methods; see e.g. [17, 25, 26, 28, 27, 32, 36, 39, 40, 42, 55].

The structure of the systems of PDEs obtained from the first four regularizers presented before is fundamentally different from that of the TV regularizer or the

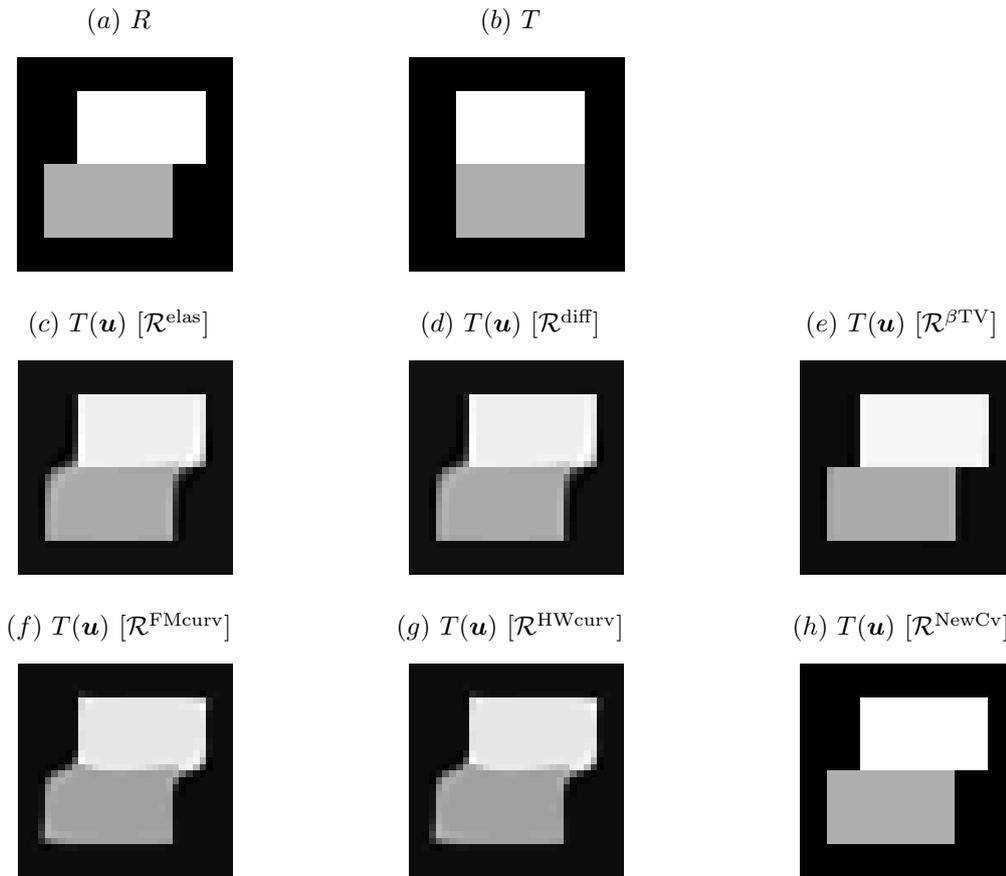


FIG. 1.1. Registered images for two rectangular blocks shown in (a) R and (b) T of size 32×32 (Example 1: results by (c) \mathcal{R}^{elas} with $(\mu, \lambda) = (1, 1)$, (d) \mathcal{R}^{diff} , (e) $\mathcal{R}^{\beta TV}$ with $\beta = 0.01$, (f) \mathcal{R}^{FMcurv} , (g) \mathcal{R}^{HWcurv} , (h) \mathcal{R}^{NewCv} with $\beta = 0.01$). A non-smooth deformation example to show that our registration model \mathcal{R}^{NewCv} gives the satisfactory registration results as good as those from $\mathcal{R}^{\beta TV}$, which is known to be suitable. Here the regularization parameter α was well-selected for all registration models.

full curvature model; see Section 2 later. While the differential operators in the first case are linear by construction, the latter ones are nonlinear and thus more difficult to handle. In this sense, the higher-order regularizer to be proposed in this paper shares this difficulty with the TV regularizer. Although all PDE techniques yield the system of nonlinear PDEs anyway due to the similarity term, this difference has a decisive fact on the design of a MG method. As is known, MG techniques (whenever they converge) are usually much faster than all other methods including the gradient descent approaches, commonly used in image processing applications. However the convergence of MG techniques is not always possible for a general problem. Therefore, developing convergent MGs for a nonlinear problem is of general interest. As far as our registration problems are concerned, the second order models can be solved efficiently using either a linear multigrid method within a fixed-point framework or a full approximation scheme [6, 8, 50, 52, 53] based nonlinear multigrid (FAS-NMG)

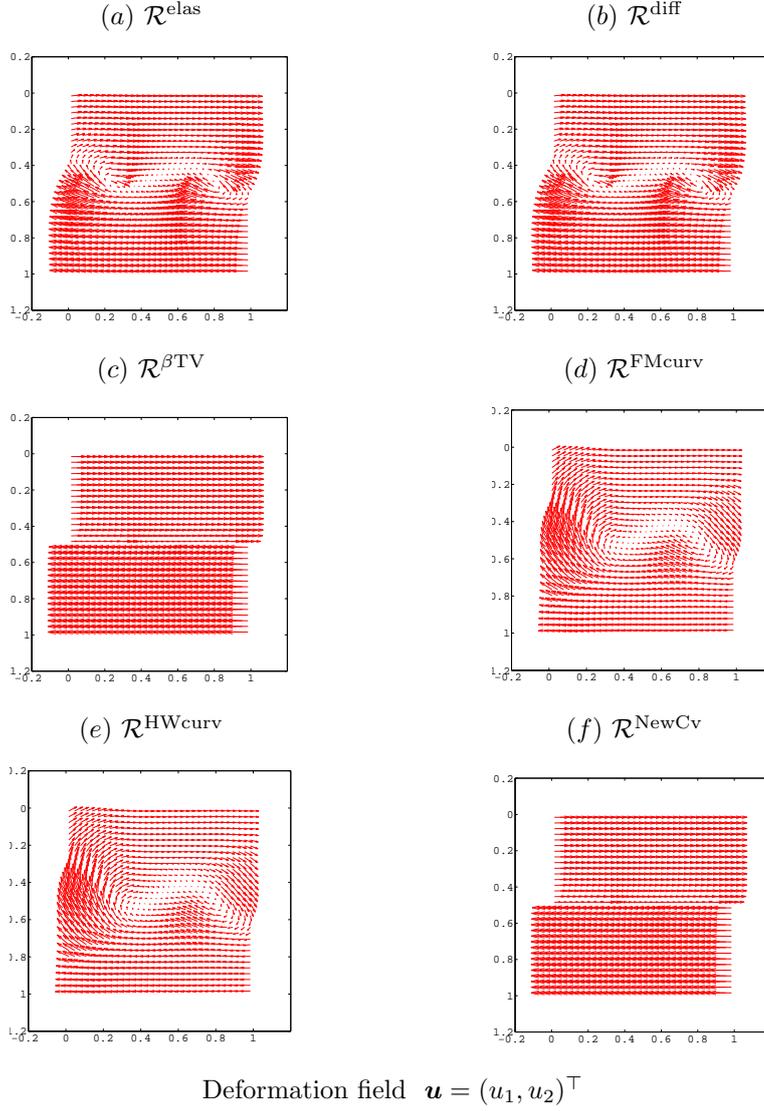


FIG. 1.2. Deformation fields for the non-smooth registration problem shown in Figure 1.1 (a) – (b) (Example 1): results by (a) $\mathcal{R}^{\text{elas}}$ with $(\mu, \lambda) = (1, 1)$, (b) $\mathcal{R}^{\text{diff}}$, (c) $\mathcal{R}^{\beta\text{TV}}$ with $\beta = 0.01$, (d) $\mathcal{R}^{\text{FMcurv}}$, (e) $\mathcal{R}^{\text{HWcurv}}$, and (f) $\mathcal{R}^{\text{NewCv}}$ with $\beta = 0.01$. The exact deformation field is given by a shift of the upper rectangular to the right and a shift of the lower rectangular to the left; c.f. Figure 1.1 (a) – (b).

method. Refer to [17, 25, 26, 28, 27, 32, 36, 33, 39, 40, 49, 55] and references therein. However, the effective solution of fourth order registration problems is much more challenging. Nevertheless a linear multigrid method within a fixed-point framework can be developed for the particular fourth order systems (1.10) and (1.13); see [28, 33, 40]. As we shall demonstrate, the new fourth order model to be proposed here cannot be solved by existing methods. We shall present our new algorithms shortly.

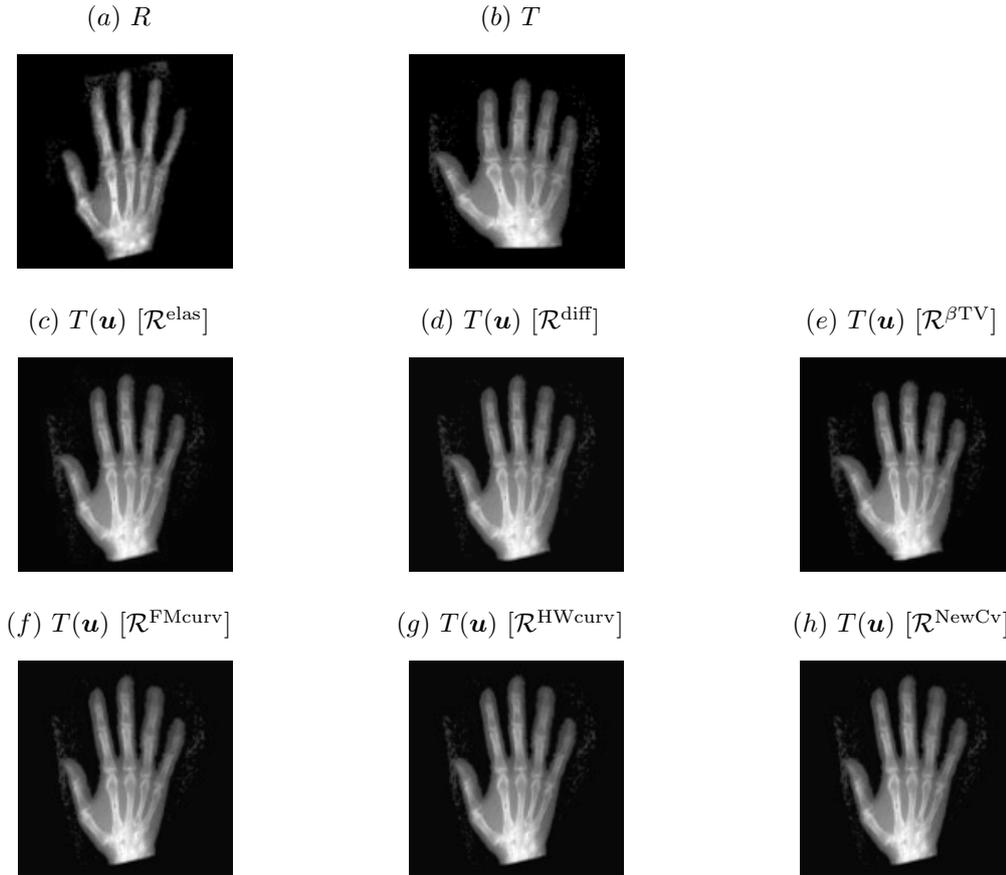


FIG. 1.3. Registered images for X-ray images shown in (a) R and (b) T of size 128×128 (Example 2): results by (c) $\mathcal{R}^{\text{elas}}$ with $(\mu, \lambda) = (1, 1)$, (d) $\mathcal{R}^{\text{diff}}$, (e) $\mathcal{R}^{\beta\text{TV}}$ with $\beta = 0.01$, (f) $\mathcal{R}^{\text{FMcurv}}$, (g) $\mathcal{R}^{\text{HWcurv}}$, (h) $\mathcal{R}^{\text{NewCv}}$. A smooth deformation example to show that our registration model $\mathcal{R}^{\text{NewCv}}$ gives the satisfactory registration results as good as those from $\mathcal{R}^{\text{FMcurv}}$ and $\mathcal{R}^{\text{HWcurv}}$, which are known to be suitable. Here the regularization parameter α was well-selected for all registration models.

The rest of the paper is organized as follows. Section 2 first presents a new PDE-based image registration model based on a curvature regularizer suitable for both smooth and non-smooth deformation problems and then discusses unilevel iterative numerical methods for it in Section 3. Section 4 presents a fast multigrid approach after first analyzing some iterative solvers as potential smoothers. Experimental results from real images illustrating the improved results from the new model and the efficiency from FAS-NMG are shown in Section 5 before conclusions in Section 6.

2. A new PDE-based image registration model. Motivated by the attractive properties of the Fischer–Modersitzki’s curvature registration model (1.9) improving on previous first order models (1.5) and (1.7), we consider an alternative formulation that uses the full curvature without approximations and hope to achieve further improvements in terms of registration quality.

Our main aim is to design a registration model or regularization energy, which

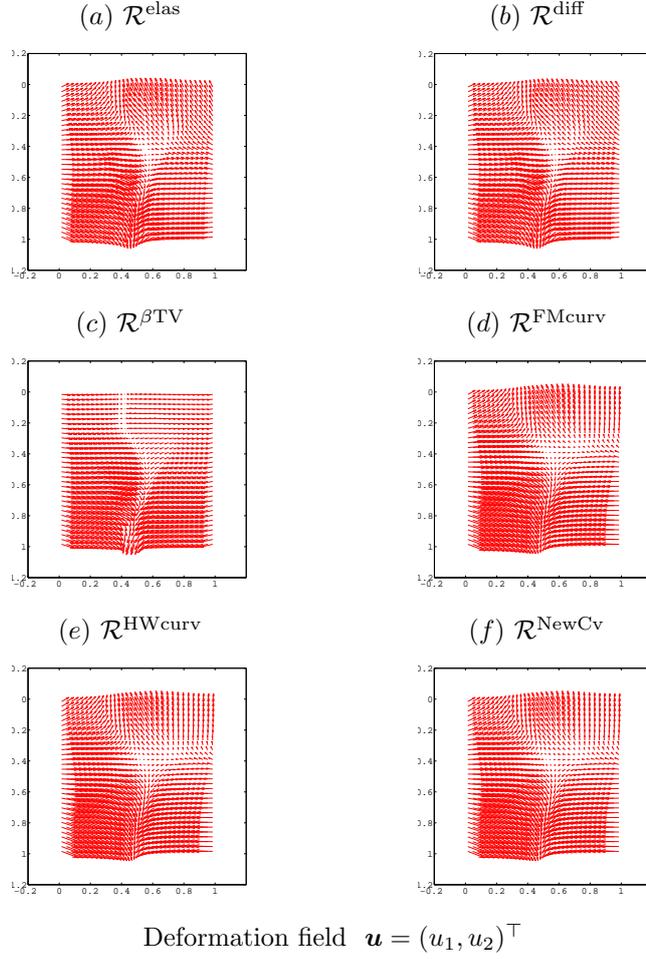


FIG. 1.4. Deformation fields for the smooth registration problem shown in Figure 1.3 (a) – (b) (Example 2): results by (a) \mathcal{R}^{elas} with $(\mu, \lambda) = (1, 1)$, (b) \mathcal{R}^{diff} , (c) $\mathcal{R}^{\beta TV}$ with $\beta = 0.01$, (d) \mathcal{R}^{FMcurv} , (e) \mathcal{R}^{HWcurv} , and (f) \mathcal{R}^{NewCv} . (c) shows the piecewise constant smoothness at the top region by $\mathcal{R}^{\beta TV}$.

is able to solve not only smooth and non-smooth registration problems, but also to allow affine-linear transformations. To this end, instead of using $\kappa_M(u_l)$ from (1.11), we consider the curvature of the level lines:

$$(2.1) \quad \kappa(u_l) = \nabla \cdot \frac{\nabla u_l}{|\nabla u_l|_\beta} = \frac{(\beta + u_{l_{x_1}}^2)u_{l_{x_1x_1}} - 2u_{l_{x_1}}u_{l_{x_2}}u_{l_{x_1x_2}} + (\beta + u_{l_{x_2}}^2)u_{l_{x_2x_2}}}{(\beta + u_{l_{x_1}}^2 + u_{l_{x_2}}^2)^{3/2}}$$

and propose the following regularization functional:

$$(2.2) \quad \mathcal{R}^{NewCv}(\mathbf{u}) = \sum_{l=1}^2 \int_{\Omega} \Phi(\kappa(u_l)) dx.$$

where $\Phi(s) = \frac{1}{2}s^2$ is mainly considered for developing a novel regularization energy

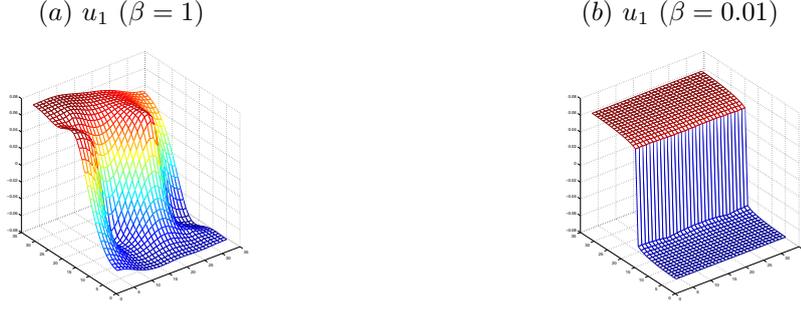


FIG. 1.5. Surface plots of u_1 for the non-smooth registration problem shown in Figure 1.1 (a) – (b) (Example 1): results by $\mathcal{R}^{\text{NewCv}}$ with (a) $\beta = 1$ and (b) $\beta = 0.01$. (a) and (b) show smoothing effects on the surface of u_1 at two different values of β .

in this work and the general notation in (2.2) allows for other choices, e.g.

$$\Phi(s) = \begin{cases} |s|, & s > 1 \\ s^2, & s \leq 1 \end{cases}$$

leading to similar registration qualities for either smooth or non-smoother registration problems. The registration model (1.2) with the regularization energy (2.2) yields to the Euler-Lagrange system of two coupled nonlinear PDEs:

$$(2.3) \quad \begin{cases} f_1(\mathbf{u}) + \alpha \nabla \cdot \underbrace{\left(\frac{1}{|\nabla u_1|_\beta} \nabla \kappa(u_1) - \frac{\nabla u_1 \cdot \nabla \kappa(u_1)}{(|\nabla u_1|_\beta)^3} \nabla u_1 \right)}_{\mathbf{V}_1} = 0 \\ f_2(\mathbf{u}) + \alpha \nabla \cdot \underbrace{\left(\frac{1}{|\nabla u_2|_\beta} \nabla \kappa(u_2) - \frac{\nabla u_2 \cdot \nabla \kappa(u_2)}{(|\nabla u_2|_\beta)^3} \nabla u_2 \right)}_{\mathbf{V}_2} = 0 \end{cases} \quad \text{(new curvature-type model)}$$

or in a compact notation

$$(2.4) \quad \begin{cases} f_1(\mathbf{u}) + \alpha \nabla \cdot \mathbf{V}_1 = 0 \\ f_2(\mathbf{u}) + \alpha \nabla \cdot \mathbf{V}_2 = 0 \end{cases} \quad \text{(new curvature-type model)}$$

where the nonlinear fitting terms f_l are defined in (1.4). Similar to those in [21], we chose the following boundary conditions

$$(2.5) \quad \langle \nabla u_l, \mathbf{n} \rangle_{\mathbb{R}^2} = \langle \nabla \kappa(u_l), \mathbf{n} \rangle_{\mathbb{R}^2} = 0 \text{ on } \partial\Omega$$

in this paper.

Our particular choice of regularizer in (2.2) has several advantages. Firstly the kernel of the proposed regularization energy (2.2) consists only of affine-linear transformations. That is, $\mathcal{R}^{\text{NewCv}}(\mathbf{A}\mathbf{x} + \mathbf{b}) = 0$ for $\mathbf{A} \in \mathbb{R}^{2 \times 2}$ and $\mathbf{b} \in \mathbb{R}^2$. Hence the proposed energy $\mathcal{R}^{\text{NewCv}}$ is invariant under planar rotation and translation. In other words, it shares this attractive property with the Fischer–Modersitzki’s and Henn–Witsch’s curvature approach. However, due to the adopted boundary conditions (2.5),

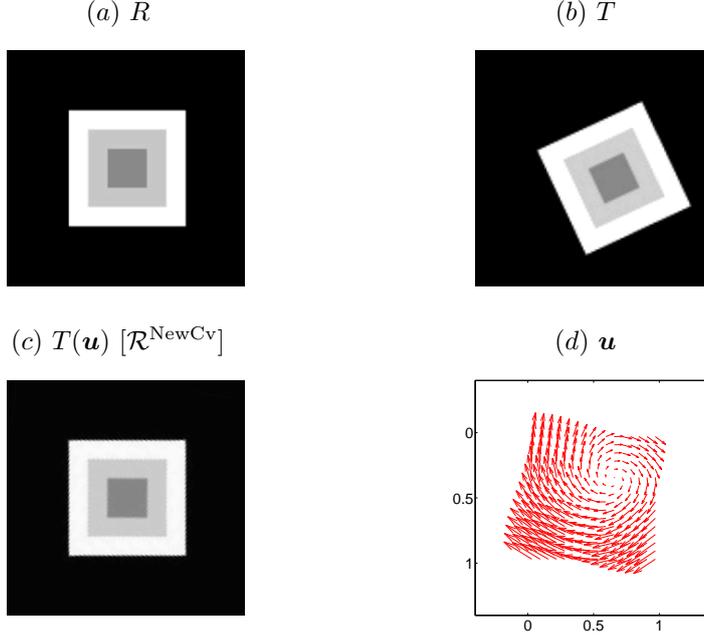


FIG. 1.6. Results by $\mathcal{R}^{\text{NewCv}}$ for the synthetic example given by [37]. Top to bottom from left to right: reference (a), template (b), registered image (c), associated deformation field (d). The registration problem to consider the effect of the underlying boundary conditions. Clearly, the affine-linear displacement is not penalized in the interior of the image domain.

the solution $\mathbf{u} = (u_1, u_2)^\top$ of the Euler-Lagrange system (2.3) is restricted to a particular subset of $H^2(\Omega) \times H^2(\Omega)$. In this case,

$$\left\{ u_l \in H^2(\Omega), \frac{\partial u_l}{\partial \mathbf{n}} = \frac{\partial \kappa(u_l)}{\partial \mathbf{n}} = 0, l = 1, 2 \right\} \subset H^2(\Omega) \times H^2(\Omega).$$

Consequently the affine-linear transformations are penalized by the underlying boundary conditions (zero Neumann boundary conditions) and only constant transformations belong to the kernel of (2.3). In order to analyze or study the effect of the boundary conditions in the new PDE model (2.3), the synthetic example given by [37] was considered. As shown in Figure 1.6, the template T is a rotated and translated version of the reference R . Therefore, the exact solution to produce a well-matched image is not a constant transformation. The result presented in Figure 1.6 (c) shows that the new PDE model (2.3) succeeds in completely matching the given images. Moreover, it is shown by Figure 1.6 (d) that the effect of the undesired boundary conditions is much less noticeable, in particular in the interior of the image domain. Note that we also consider the effect of the boundary conditions for the case where the required geometric transformation is very complex; see Example 3 and its results in Section 5.1 later.

Secondly the new PDE model (2.3) preserves discontinuities of \mathbf{u} because the diffusion coefficients $\frac{1}{|\nabla u_l|_\beta} \rightarrow 0$ and $\frac{\nabla u_l \cdot \nabla \kappa(u_l)}{|\nabla u_l|_\beta^3} \rightarrow 0$ when $|\nabla u_l|_\beta \rightarrow \infty$. In other words,

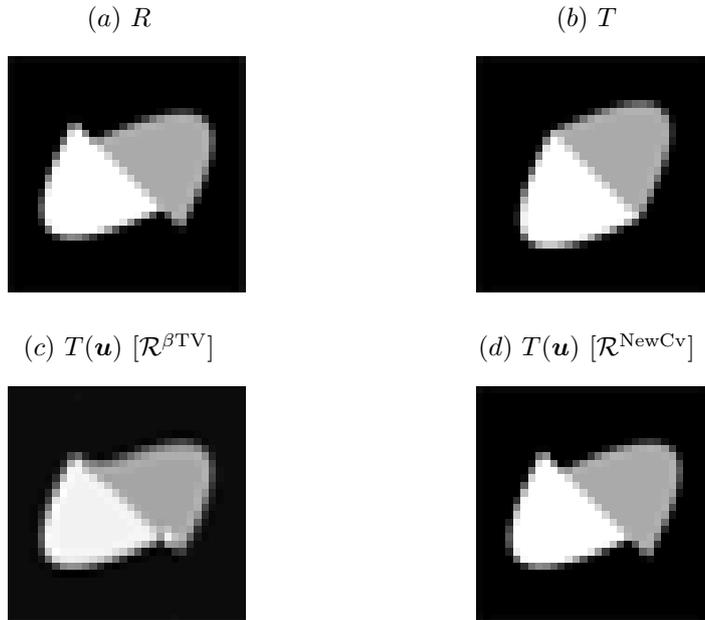


FIG. 1.7. Registered images for two objects shown in (a) R and (b) T of size 32×32 : results by (c) $\mathcal{R}^{\beta TV}$ with $\beta = 0.01$, (d) $\mathcal{R}^{\text{NewCv}}$ with $\beta = 0.01$. The non-smooth registration problem with non-axis-aligned discontinuities to show that the TV model (1.15) and the new PDE model (2.3) are not rotationally invariant.

for non-smooth registration problems the new PDE model preserves discontinuities of \mathbf{u} by reducing or stopping the diffusion (smoothing) process in inhomogeneous regions presenting large gradients. Note also that the proposed regularization energy $\mathcal{R}^{\text{NewCv}}$ reduces to $\mathcal{R}^{\text{FMcuv}}$ in (1.9) if $|\nabla u_i| \approx 0$ and we take $\beta = 1$.

Thirdly the role of the lifting parameter β for $\mathcal{R}^{\beta TV}$ and $\mathcal{R}^{\text{NewCv}}$ is not only to avoid division by zero in the numerical implementation, but can also be used to adjust the ellipticness of the PDE models (1.15) and (2.3). For instance, $\beta = 1$ results in a more elliptic PDE and therefore the recovered deformation fields are more smooth than those of a small β ; see Figure 1.9 for the results by $\mathcal{R}^{\beta TV}$ and Figure 1.5 by $\mathcal{R}^{\text{NewCv}}$ with different values of β . Although this can be used to approximate first-order homogeneous diffusion processes in (1.15) for the TV model, a higher-order PDE model like the one we are proposing in (2.3) is still needed to obtain correct solutions for difficult registration problems as we will show through Example 3 in Section 5.1; see Figures 5.1–5.3. However, $\beta \ll 1$ is required for non-smooth registration problems.

Finally, although any linear combination of a smooth and a non-smooth regularization energy can solve both smooth and non-smooth registration problems, a regularization energy based on higher-order derivatives like (2.2) is required for registration problems where an affine-linear pre-registration step is unavoidable.

REMARK 1. *Similar to the TV model (1.15) and most other variational models for registration [44], the new PDE model (2.3) is not rotationally invariant. This*

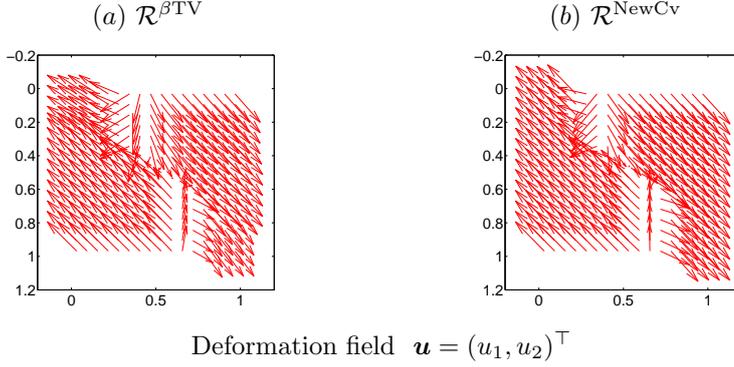


FIG. 1.8. Deformation fields for the non-smooth registration problem shown in Figure 1.7 (a) – (b): results by (a) $\mathcal{R}^{\beta TV}$ with $\beta = 0.01$, (b) \mathcal{R}^{NewCv} with $\beta = 0.01$. The exact deformation field is given by a shift of the upper object to the bottom-right corner and a shift of the lower object to the top-left corner. Clearly, the TV model (1.15) and the new PDE model (2.3) are not rotationally invariant, c.f. Figures 1.2.

is because the nonlinear diffusion processes resulting from the Gâteaux derivative of \mathcal{R}^{NewCv} do not enforce coupling between the displacement variables u_1 and u_2 . As already mentioned in Section 1, it may prevent to obtain a good registration in some situations, e.g non-smooth registration problems with non-axis-aligned discontinuities. Two registration problems and their results as shown in Figures 1.1–1.2 and 1.7–1.8 confirm this fact. A possible way to solve this problem that we will explore in the short future is to use vectorial regularizations similar to those in [5, 7, 11] for vector-valued image denoising.

From now on we shall use the notation \mathcal{R}^{NewCv} to mean the full curvature model (2.2) and the numerical solutions of (2.4) will be discussed next.

3. Numerical solution of the PDE system (2.4). While variational models have already made useful contributions to high-resolution image processing, a remaining major challenge is to implement fast and stable numerical algorithms for solving various associated Euler-Lagrange systems (of nonlinear PDEs). In this section we briefly review possible numerical methods that have been studied for other models and could be considered for solving (2.4). To proceed, denote the discrete domain consisting of $N = n^2$ cells of size $h \times h$ by

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

throughout this section where $h = 1/n$ denotes the grid space.

Finite difference discretization. We shall use a cell-centered finite difference approximation for the underlying PDEs. For simplicity, let $(u_l^h)_{i,j} = u_l^h(x_{1_i}, x_{2_j})$ denote the grid functions for $l = 1, 2$. After discretizing (2.4), the grid system at

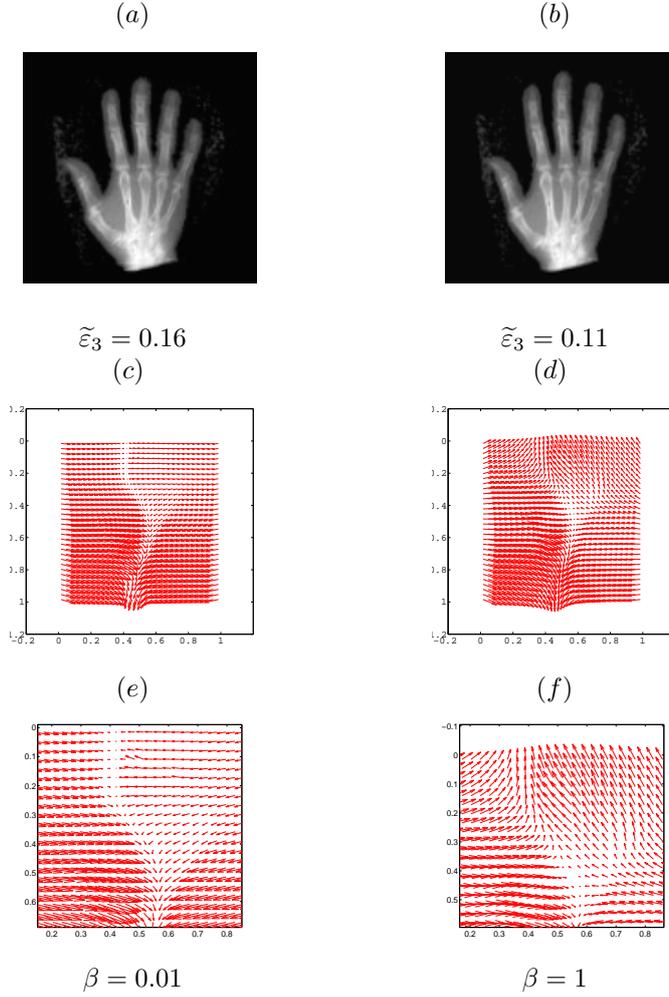


FIG. 1.9. Registered images and associated deformation fields for the registration problem (the smooth case) shown in Figure 1.3 (a) – (b) by $\mathcal{R}^{\beta TV}$ with different values of β . Top: registered images by (a) $\beta = 0.01$ and (b) $\beta = 1$. Middle: associated deformation fields at (c) $\beta = 0.01$ and (d) $\beta = 1$. Bottom: associated deformation fields at top regions of (c) and (d). Clearly, a suitably selected β can be used to approximate first-order homogeneous diffusion processes in (1.15) and improve results of $\mathcal{R}^{\beta TV}$. The associated deformation fields determined by $\beta = 1$ is more smoother than those of $\beta = 0.01$. Recall that $\tilde{\varepsilon}_3$ means the relative reduction of the dissimilarity defined in Algorithm 2.

$(i, j) \in \Omega_h$ is given by

$$(3.1) \quad \begin{cases} \underbrace{f_1^h(u_1^h, u_2^h)_{i,j} + \alpha \nabla \cdot (\mathbf{V}_1^h)_{i,j}}_{\overline{\mathcal{N}}_1^h(\mathbf{u}^h)_{i,j}} = 0 \\ \underbrace{f_2^h(u_1^h, u_2^h)_{i,j} + \alpha \nabla \cdot (\mathbf{V}_2^h)_{i,j}}_{\overline{\mathcal{N}}_2^h(\mathbf{u}^h)_{i,j}} = 0 \end{cases} \quad \text{i.e.} \quad \begin{cases} \overline{\mathcal{N}}_1^h(\mathbf{u}^h)_{i,j} = 0, \\ \overline{\mathcal{N}}_2^h(\mathbf{u}^h)_{i,j} = 0, \end{cases}$$

subject to the discrete boundary conditions,

$$(3.2) \quad \begin{cases} (u_l^h)_{i,1} = (u_l^h)_{i,2}, & (u_l^h)_{i,n} = (u_l^h)_{i,n-1}, \\ (u_l^h)_{1,j} = (u_l^h)_{2,j}, & (u_l^h)_{n,j} = (u_l^h)_{n-1,j}, \\ \kappa(u_l^h)_{i,1} = \kappa(u_l^h)_{i,2}, & \kappa(u_l^h)_{i,n} = \kappa(u_l^h)_{i,n-1}, \\ \kappa(u_l^h)_{1,j} = \kappa(u_l^h)_{2,j}, & \kappa(u_l^h)_{n,j} = \kappa(u_l^h)_{n-1,j}, \end{cases}$$

with the following notation for the fitting terms f_l from (1.4)

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

Here we approximate the term $\nabla \cdot (\mathbf{V}_l^h)_{i,j}$ as follows:

$$(3.3) \quad \left(\frac{\partial V_l^1}{\partial x_1}\right)_{i,j} + \left(\frac{\partial V_l^2}{\partial x_2}\right)_{i,j} = \frac{(V_l^1)_{i+1,j} - (V_l^1)_{i,j}}{h} + \frac{(V_l^2)_{i,j+1} - (V_l^2)_{i,j}}{h}.$$

Therefore, we need to calculate V_l^1 at the grid points $(i+1, j)$ and (i, j) and V_l^2 at the grid points $(i, j+1)$ and (i, j) . Below we list the approximation used in our numerical realizations for estimating V_l^1 at the grid point (i, j) :

$$\begin{aligned} \kappa(u_l^h)_{i,j} &= \left[\frac{\delta_{x_1}^-}{h} \left(\frac{\delta_{x_1}^+(u_l^h)_{i,j}/h}{\sqrt{\beta + (\delta_{x_1}^+(u_l^h)_{i,j}/h)^2 + (\delta_{x_2}^+(u_l^h)_{i,j}/h)^2}} \right) \right. \\ &\quad \left. + \frac{\delta_{x_2}^-}{h} \left(\frac{\delta_{x_2}^+(u_l^h)_{i,j}/h}{\sqrt{\beta + (\delta_{x_1}^+(u_l^h)_{i,j}/h)^2 + (\delta_{x_2}^+(u_l^h)_{i,j}/h)^2}} \right) \right], \\ (u_{l_{x_1}}^h)_{i,j} &= \delta_{x_1}^+(u_l^h)_{i,j}/h, \\ (u_{l_{x_2}}^h)_{i,j} &= \delta_{x_2}^+(u_l^h)_{i,j}/h, \\ \delta_{x_1}^\pm(u_l^h)_{i,j} &= \pm((u_l^h)_{i\pm 1,j} - (u_l^h)_{i,j}), \\ \delta_{x_2}^\pm(u_l^h)_{i,j} &= \pm((u_l^h)_{i,j\pm 1} - (u_l^h)_{i,j}), \\ |\nabla u_l|_\beta &= \sqrt{\beta + (\delta_{x_1}^+(u_l^h)_{i,j}/h)^2 + (\delta_{x_2}^+(u_l^h)_{i,j}/h)^2}, \\ ((\kappa(u_l))_{x_1})_{i,j} &= [\kappa(u_l)_{i+1,j} - \kappa(u_l)_{i,j}]/h, \\ ((\kappa(u_l))_{x_2})_{i,j} &= [\kappa(u_l)_{i,j+1} - \kappa(u_l)_{i,j}]/h. \end{aligned}$$

Discretization for V_l^1 at the grid point $(i+1, j)$ and V_l^2 at the grid points $(i, j+1)$ and (i, j) can be given similarly.

3.1. Method 1 – a semi-implicit time marching (SITM) method. As discussed in Section 1, the main idea of time marching approaches is to introduce an artificial time variable t and compute the steady-state solution of the system of time-dependent PDEs of the form:

$$\begin{cases} \partial_t u_1(\mathbf{x}; t) + \overline{\mathcal{N}}_1(\mathbf{u}(\mathbf{x}; t)) = 0 \\ \partial_t u_2(\mathbf{x}; t) + \overline{\mathcal{N}}_2(\mathbf{u}(\mathbf{x}; t)) = 0 \end{cases}$$

where

$$\begin{aligned} \bar{\mathcal{N}}_l(\mathbf{u}(\mathbf{x}; t)) &= f_l(\mathbf{u}(\mathbf{x}; t)) + \alpha \nabla \cdot \left(\frac{1}{|\nabla u_l(\mathbf{x}; t)|_\beta} \nabla \kappa(u_l(\mathbf{x}; t)) \right) \\ &\quad - \frac{\nabla u_l(\mathbf{x}; t) \cdot \nabla \kappa(u_l(\mathbf{x}; t))}{|\nabla u_l(\mathbf{x}; t)|_\beta^3} \nabla u_l(\mathbf{x}; t). \end{aligned}$$

In order to overcome the nonlinearity of $\bar{\mathcal{N}}_l$, the so-called *explicit scheme* can be conveniently applied, and the iteration is then given by

$$\begin{cases} \partial_t u_1(\mathbf{x}, t_{k+1}) = -\bar{\mathcal{N}}_1(\mathbf{u}(\mathbf{x}, t_k)) \\ \partial_t u_2(\mathbf{x}, t_{k+1}) = -\bar{\mathcal{N}}_2(\mathbf{u}(\mathbf{x}, t_k)) \end{cases} \quad k = 0, 1, 2, 3, \dots$$

where $\mathbf{u}(\mathbf{x}, t_0)$ is some initial displacement field, typically $\mathbf{u}(\mathbf{x}, t_0) = 0$.

For the time discretization we introduce a time-step $\tau > 0$, and update \mathbf{u} at the $(k+1)$ th time-step by

$$\begin{cases} u_1(\mathbf{x}, t_{k+1}) = u_1(\mathbf{x}, t_k) - \tau \bar{\mathcal{N}}_1(\mathbf{u}(\mathbf{x}, t_k)) \\ u_2(\mathbf{x}, t_{k+1}) = u_2(\mathbf{x}, t_k) - \tau \bar{\mathcal{N}}_2(\mathbf{u}(\mathbf{x}, t_k)) \end{cases}$$

which we simply denote by

$$(3.4) \quad \begin{cases} (u_1^{(k+1)})_{i,j} = (u_1^{(k)})_{i,j} - \tau \bar{\mathcal{N}}_1(\mathbf{u}^{(k)})_{i,j} \\ (u_2^{(k+1)})_{i,j} = (u_2^{(k)})_{i,j} - \tau \bar{\mathcal{N}}_2(\mathbf{u}^{(k)})_{i,j} \end{cases},$$

where the symbol h used in the previous section is dropped for simplicity. We note that the above time-marching based fourth-order numerical scheme for (2.4) is easy to implement, but very slow to converge because the time-step τ is required to be proportional to the fourth power of the grid spacing for stability reasons.

In order to speed up the convergence of (3.4), we may increase implicitness via linearizing nonlinear ‘coefficients’ in the associated system and define the iteration step as follows:

$$\begin{cases} (u_1^{(k+1)})_{i,j} = (u_1^{(k)})_{i,j} - \tau \bar{\mathcal{N}}_1^{\text{lin}}(\mathbf{u}^{(k+1)})_{i,j} \\ (u_2^{(k+1)})_{i,j} = (u_2^{(k)})_{i,j} - \tau \bar{\mathcal{N}}_2^{\text{lin}}(\mathbf{u}^{(k+1)})_{i,j} \end{cases} \quad \text{i.e.} \quad \begin{cases} \mathbf{A}_1(u_1^{(k)})(u_1^{(k+1)})_{i,j} = \mathbf{B}_1(\mathbf{u}^{(k)})_{i,j}, \\ \mathbf{A}_2(u_2^{(k)})(u_2^{(k+1)})_{i,j} = \mathbf{B}_2(\mathbf{u}^{(k)})_{i,j}, \end{cases}$$

or in full details

$$(3.5) \quad \left\{ \begin{aligned} & \underbrace{-\alpha \tau \nabla \cdot \left(\frac{\nabla u_1^{(k)} \cdot \nabla \kappa(u_1^{(k)})}{|\nabla u_1^{(k)}|_\beta^3} \nabla u_1^{(k+1)} \right)_{i,j} + (u_1^{(k+1)})_{i,j}}_{\mathbf{A}_1(u_1^{(k)})(u_1^{(k+1)})_{i,j}} \\ & \underbrace{(u_1^{(k)})_{i,j} - \tau [f_1(\mathbf{u}^{(k)})_{i,j} + \alpha \nabla \cdot \left(\frac{1}{|\nabla (u_1^{(k)})_{i,j}|_\beta} \nabla \kappa(u_1^{(k)})_{i,j} \right)]}_{\mathbf{B}_1(\mathbf{u}^{(k)})_{i,j}} \\ & \underbrace{-\alpha \tau \nabla \cdot \left(\frac{\nabla u_2^{(k)} \cdot \nabla \kappa(u_2^{(k)})}{|\nabla u_2^{(k)}|_\beta^3} \nabla u_2^{(k+1)} \right)_{i,j} + (u_2^{(k+1)})_{i,j}}_{\mathbf{A}_2(u_2^{(k)})(u_2^{(k+1)})_{i,j}} \\ & \underbrace{(u_2^{(k)})_{i,j} - \tau [f_2(\mathbf{u}^{(k)})_{i,j} + \alpha \nabla \cdot \left(\frac{1}{|\nabla (u_2^{(k)})_{i,j}|_\beta} \nabla \kappa(u_2^{(k)})_{i,j} \right)]}_{\mathbf{B}_2(\mathbf{u}^{(k)})_{i,j}} \end{aligned} \right.$$

which is a semi-implicit time marching scheme for (2.4). Here we denote by

$$(3.6) \quad \begin{aligned} \bar{\mathcal{N}}_l^{\text{lin}}(\mathbf{u}^{(k+1)})_{i,j} &= f_l(\mathbf{u}^{(k)})_{i,j} + \alpha \nabla \cdot \left(\frac{1}{|\nabla u_l^{(k)}|_\beta} \nabla \kappa(u_l^{(k)}) \right) \\ &\quad - \frac{\nabla u_l^{(k)} \cdot \nabla \kappa(u_l^{(k)})}{|\nabla u_l^{(k)}|_\beta^3} \nabla u_l^{(k+1)}_{i,j}. \end{aligned}$$

the frozen operator, linearized at a grid point (i, j) . We note that this frozen operator allows us to solve (2.4) as the system of two second-order linearized PDEs at each time step k because the coefficients from the higher-order derivatives are frozen in the associated discrete system.

3.2. Method 2 – a stabilized semi-implicit time marching (SSITM) method. Although this above idea of linearization via semi-implicitness seems reasonable, we found experimentally that the numerical scheme (3.5) (though a bit better than the explicit scheme (3.4)) is only stable when τ is small and small τ will lead to slow convergence in the overall registration process. The reason for this stability problem is that the discrete system has a highly nonlinear coefficient $\frac{\nabla u_l^{(k)} \cdot \nabla \kappa(u_l^{(k)})}{|\nabla u_l^{(k)}|_\beta^3}$ that can easily change its sign for large τ so neither positive-definiteness nor diagonal dominance can be guaranteed for numerical schemes of the underlying system (a matrix form of (3.5))

$$\begin{bmatrix} \mathbf{A}_1(u_1^{(k)}) & 0 \\ 0 & \mathbf{A}_2(u_1^{(k)}) \end{bmatrix} \begin{pmatrix} u_1^{(k+1)} \\ u_2^{(k+1)} \end{pmatrix} = \begin{pmatrix} \mathbf{B}_1(\mathbf{u}^{(k)}) \\ \mathbf{B}_2(\mathbf{u}^{(k)}) \end{pmatrix}.$$

In order to improve stability, the stabilizing terms based on the so-called *convexity-splitting* technique developed in [9, 10, 18, 19] may be added as follows:

$$(3.7) \quad \begin{cases} \gamma_1 \tau \mathcal{F}(u_1^{(k+1)})_{i,j} - \alpha \tau \nabla \cdot \left(\frac{\nabla u_1^{(k)} \cdot \nabla \kappa(u_1^{(k)})}{|\nabla u_1^{(k)}|_\beta^3} \nabla u_1^{(k+1)} \right)_{i,j} + (u_1^{(k+1)})_{i,j} \\ = \gamma_1 \tau \mathcal{F}(u_1^{(k)})_{i,j} + (u_1^{(k)})_{i,j} - \tau [f_1(\mathbf{u}^{(k)})_{i,j} + \alpha \nabla \cdot \left(\frac{1}{|\nabla u_1^{(k)}|_\beta} \nabla \kappa(u_1^{(k)}) \right)_{i,j}] \\ \gamma_2 \tau \mathcal{F}(u_2^{(k+1)})_{i,j} - \alpha \tau \nabla \cdot \left(\frac{\nabla u_2^{(k)} \cdot \nabla \kappa(u_2^{(k)})}{|\nabla u_2^{(k)}|_\beta^3} \nabla u_2^{(k+1)} \right)_{i,j} + (u_2^{(k+1)})_{i,j} \\ = \gamma_2 \tau \mathcal{F}(u_2^{(k)})_{i,j} + (u_2^{(k)})_{i,j} - \tau [f_2(\mathbf{u}^{(k)})_{i,j} + \alpha \nabla \cdot \left(\frac{1}{|\nabla u_2^{(k)}|_\beta} \nabla \kappa(u_2^{(k)}) \right)_{i,j}] \end{cases}$$

where $\gamma_l > 0$ and the stabilizing term $\mathcal{F}(u_l)$ is some appropriate partial differential operator arising from the minimization of a convex functional, such as $\int_\Omega |\nabla u_l| d\mathbf{x}$ or $\int_\Omega |\nabla u_l|^2 d\mathbf{x}$.

Note that $\frac{1}{|\nabla u_l^{(k)}|_\beta} \rightarrow 1$ as $|\nabla u_l^{(k)}| \rightarrow 0$ for smooth problems ($\beta = 1$) and $\frac{1}{|\nabla u_l^{(k)}|_\beta} \rightarrow 0$ as $|\nabla u_l^{(k)}| \rightarrow \infty$ for non-smooth problems ($\beta < 1$). Therefore, $\mathcal{F}(u_l^{(k+1)})_{i,j} = -\nabla \cdot \left(\frac{\nabla u_l^{(k+1)}}{|\nabla u_l^{(k+1)}|_\beta} \right)_{i,j}$ resulting from $\int_\Omega |\nabla u_l| d\mathbf{x}$ smooths \mathbf{u} isotropically inside homogeneous regions corresponding to weak gradients and preserves discontinuities of \mathbf{u} in inhomogeneous regions representing large gradients by reducing or stopping diffusion process. As a consequence, $\mathcal{F}(u_l^{(k+1)})_{i,j} = -\nabla \cdot \left(\frac{\nabla u_l^{(k+1)}}{|\nabla u_l^{(k+1)}|_\beta} \right)_{i,j}$ appears to be an appropriate choice for both smooth and non-smooth registration problems, while $\mathcal{F}(u_l^{(k+1)})_{i,j} = -\Delta(u_l^{(k+1)})_{i,j}$ resulting from $\int_\Omega |\nabla u_l|^2 d\mathbf{x}$ is only suitable for smooth cases.

3.3. Method 3 – fixed-point methods. As is well-known [51, 15, 16], fixed-point (FP) methods are usually faster than time marching approaches when appropriate FP schemes are applied. To try this idea, we use a similar linearization to the above (3.7) without γ_l, τ plus a linearized version of $f_l(u_1^{[\nu+1]}, u_2^{[\nu+1]})$ via Taylor's expansion as follows

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

where

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

and

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

Then a FP scheme for (2.4) can be given by (for $\nu = 0, 1, 2, 3, \dots$)

$$(3.9) \quad \left\{ \begin{array}{l}
-\alpha \nabla \cdot \left(\frac{\nabla u_1^{[\nu]} \cdot \nabla \kappa(u_1^{[\nu]})}{|\nabla u_1^{[\nu]}|_\beta^3} \nabla u_1^{[\nu+1]} \right)_{i,j} + (\sigma_{11}^{[\nu]})_{i,j} (u_1^{[\nu+1]})_{i,j} \\
+ (\sigma_{12}^{[\nu]})_{i,j} (u_2^{[\nu+1]})_{i,j} = (\sigma_{11}^{[\nu]})_{i,j} (u_1^{[\nu]})_{i,j} + (\sigma_{12}^{[\nu]})_{i,j} (u_2^{[\nu]})_{i,j} \\
-f_1(\mathbf{u}^{[\nu]})_{i,j} - \alpha \nabla \cdot \left(\frac{1}{|\nabla u_1^{[\nu]}|_\beta} \nabla \kappa(u_1^{[\nu]}) \right)_{i,j}, \\
-\alpha \nabla \cdot \left(\frac{\nabla u_2^{[\nu]} \cdot \nabla \kappa(u_2^{[\nu]})}{|\nabla u_2^{[\nu]}|_\beta^3} \nabla u_2^{[\nu+1]} \right)_{i,j} + (\sigma_{22}^{[\nu]})_{i,j} (u_2^{[\nu+1]})_{i,j} \\
+ (\sigma_{21}^{[\nu]})_{i,j} (u_1^{[\nu+1]})_{i,j} = (\sigma_{21}^{[\nu]})_{i,j} (u_1^{[\nu]})_{i,j} + (\sigma_{22}^{[\nu]})_{i,j} (u_2^{[\nu]})_{i,j} \\
-f_2(\mathbf{u}^{[\nu]})_{i,j} - \alpha \nabla \cdot \left(\frac{1}{|\nabla u_2^{[\nu]}|_\beta} \nabla \kappa(u_2^{[\nu]}) \right)_{i,j}.
\end{array} \right.$$

Here $\sigma_{l1}^{[\nu]}, \sigma_{l2}^{[\nu]}$ are refined as follows. We first see that $\sigma_{21}^{[\nu]} = \sigma_{12}^{[\nu]}$. In order to have a simple and stable numerical scheme as noted in several works in different contexts; see e.g. [28], [40] and [44, p.56-79], we approximate $\sigma_{lm}^{[\nu]}$ by $\sigma_{lm}^{[\nu]} = (\partial_{u_l} T(\mathbf{u}^{[\nu]})) (\partial_{u_m} T(\mathbf{u}^{[\nu]}))$ for $m = 1, 2$ since the image difference $T(\mathbf{u}^{[\nu]}) - R$ becomes small for well registered images and so the second-order derivatives need not be evaluated.

Unfortunately, we found experimentally that the above FP scheme is neither *stable* nor *convergent*. This difficulty arises from the unbalanced terms present in the resulting discrete system. For example, fixing $\beta = 10^{-2}$ in the flat regions where $|\nabla u_l| = 0$ reduces the diffusion coefficient $\frac{\nabla u_l \cdot \nabla \kappa(u_l)}{|\nabla u_l|_\beta^3} \approx O(10^6)$ compared with that of only $\frac{1}{|\nabla u_l|_\beta} \approx O(10^2)$ for the TV restoration case [46, 51].

3.4. Method 4 – a stabilized fixed-point (SFP) method. In order to improve the FP scheme (3.9), the convexity-splitting idea [9, 18, 19] is again considered

by adding stabilized terms as follows:

$$(3.10) \quad \left\{ \begin{array}{l} \gamma_1 \mathcal{F}(u_1^{[\nu+1]})_{i,j} - \alpha \nabla \cdot \left(\frac{\nabla u_1^{[\nu]} \cdot \nabla \kappa(u_1^{[\nu]})}{|\nabla u_1^{[\nu]}|_\beta^3} \nabla u_1^{[\nu+1]} \right)_{i,j} + (\sigma_{11}^{[\nu]})_{i,j} (u_1^{[\nu+1]})_{i,j} \\ \quad + (\sigma_{12}^{[\nu]})_{i,j} (u_2^{[\nu+1]})_{i,j} = \gamma_1 \mathcal{F}(u_1^{[\nu]})_{i,j} + (\sigma_{11}^{[\nu]})_{i,j} (u_1^{[\nu]})_{i,j} \\ \quad + (\sigma_{12}^{[\nu]})_{i,j} (u_2^{[\nu]})_{i,j} - f_1(\mathbf{u}^{[\nu]})_{i,j}, -\alpha \nabla \cdot \left(\frac{1}{|\nabla u_1^{[\nu]}|_\beta} \nabla \kappa(u_1^{[\nu]}) \right)_{i,j} \\ \gamma_2 \mathcal{F}(u_2^{[\nu+1]})_{i,j} - \alpha \nabla \cdot \left(\frac{\nabla u_2^{[\nu]} \cdot \nabla \kappa(u_2^{[\nu]})}{|\nabla u_2^{[\nu]}|_\beta^3} \nabla u_2^{[\nu+1]} \right)_{i,j} + (\sigma_{22}^{[\nu]})_{i,j} (u_2^{[\nu+1]})_{i,j} \\ \quad + (\sigma_{21}^{[\nu]})_{i,j} (u_1^{[\nu+1]})_{i,j} = \gamma_2 \mathcal{F}(u_2^{[\nu]})_{i,j} + (\sigma_{21}^{[\nu]})_{i,j} (u_1^{[\nu]})_{i,j} \\ \quad + (\sigma_{22}^{[\nu]})_{i,j} (u_2^{[\nu]})_{i,j} - f_2(\mathbf{u}^{[\nu]})_{i,j} - \alpha \nabla \cdot \left(\frac{1}{|\nabla u_2^{[\nu]}|_\beta} \nabla \kappa(u_2^{[\nu]}) \right)_{i,j} \end{array} \right.$$

and we shall name this resulting FP scheme as the *stabilized fixed-point* (SFP) method.

As mentioned in Method 2, we also found that $\mathcal{F}(u_i^{[\nu+1]}) = -\nabla \cdot \left(\frac{\nabla u_i^{[\nu+1]}}{|\nabla u_i^{[\nu]}|_\beta} \right)$ is a suitable choice for both smooth and non-smooth registration problems. Therefore, our SFP method can be explicitly expressed as follows:

$$(3.11) \quad \mathbf{N}^{\text{SFP}}[\mathbf{u}^{[\nu]}] \mathbf{u}^{[\nu+1]} = \mathbf{G}^{\text{SFP}}[\mathbf{u}^{[\nu]}]$$

where

$$\begin{aligned} \mathbf{N}^{\text{SFP}}[\mathbf{u}^{[\nu]}] &= \begin{bmatrix} -\alpha \mathcal{L}_1^{\text{SFP}}[u_1^{[\nu]}]_{i,j} & (\sigma_{12}^{[\nu]})_{i,j} \\ (\sigma_{21}^{[\nu]})_{i,j} & -\alpha \mathcal{L}_2^{\text{SFP}}[u_2^{[\nu]}]_{i,j} \end{bmatrix}, \\ \mathbf{G}^{\text{SFP}}[\mathbf{u}^{[\nu]}] &= \begin{pmatrix} (\hat{g}_1)_{i,j}^{[\nu]} \\ (\hat{g}_2)_{i,j}^{[\nu]} \end{pmatrix} \\ (\hat{g}_i)_{i,j}^{[\nu]} &= -\gamma_i \nabla \cdot \left(\frac{\nabla u_i^{[\nu]}}{|\nabla u_i^{[\nu]}|_\beta} \right)_{i,j} + (\sigma_{i1}^{[\nu]})_{i,j} (u_1^{[\nu]})_{i,j} + (\sigma_{i2}^{[\nu]})_{i,j} (u_2^{[\nu]})_{i,j} - f_i(\mathbf{u}^{[\nu]})_{i,j} \\ &\quad - \alpha \nabla \cdot \left(\frac{1}{|\nabla u_i^{[\nu]}|_\beta} \nabla \kappa(u_i^{[\nu]}) \right)_{i,j} \end{aligned}$$

and

$$\begin{aligned} -\alpha \mathcal{L}_i^{\text{SFP}}[u_i^{[\nu]}]_{i,j} (u_i^{[\nu+1]})_{i,j} &= -\alpha \nabla \cdot \left(\overbrace{\left(\frac{\gamma_i / \alpha}{|\nabla u_i^{[\nu]}|_\beta} + \frac{\nabla u_i^{[\nu]} \cdot \nabla \kappa(u_i^{[\nu]})}{|\nabla u_i^{[\nu]}|_\beta^3} \right)}^{D_i(u_i^{[\nu]})} \nabla u_i^{[\nu+1]} \right)_{i,j} \\ &\quad + (\sigma_{ii}^{[\nu]})_{i,j} (u_i^{[\nu+1]})_{i,j}. \end{aligned}$$

In each SFP outer iteration ν , a *pointwise collective Gauss-Seidel* (PCGS) relaxation method is used as the inner solver in our numerical scheme to solve approximately the associated linear system. Here the k th PCGS step is given by

$$(3.12) \quad (\mathbf{u}^{[\nu+1]})_{i,j}^{[k+1]} = \left(\mathbf{N}^{\text{SFP}}[\mathbf{u}^{[\nu]}]_{i,j} \right)^{-1} (\mathbf{G}^{\text{SFP}}[\mathbf{u}^{[\nu]}]_{i,j})^{[k+1/2]},$$

where

$$\mathbf{N}^{\text{SFP}}[\mathbf{u}^{[\nu]}]_{i,j} = \begin{bmatrix} \alpha \frac{(\Sigma_1^{[\nu]})_{i,j}}{h^2} + (\sigma_{11}^{[\nu]})_{i,j} & (\sigma_{12}^{[\nu]})_{i,j} \\ (\sigma_{21}^{[\nu]})_{i,j} & \alpha \frac{(\Sigma_2^{[\nu]})_{i,j}}{h^2} + (\sigma_{22}^{[\nu]})_{i,j} \end{bmatrix},$$

$$\begin{aligned} (\mathbf{G}^{\text{SFP}}[\mathbf{u}^{[\nu]}])_{i,j}^{[k+1/2]} &= \begin{pmatrix} (\widehat{g}_1)_{i,j}^{[\nu]} + \alpha(\overline{\Sigma}_1^{[\nu]})_{i,j}(u_1^{[\nu+1]})_{i,j}^{[k+1/2]} \\ (\widehat{g}_2)_{i,j}^{[\nu]} + \alpha(\overline{\Sigma}_2^{[\nu]})_{i,j}(u_2^{[\nu+1]})_{i,j}^{[k+1/2]} \end{pmatrix}, \\ (\Sigma_l^{[\nu]})_{i,j} &= (2D_{l3}(u_l^{[\nu]})_{i,j} + D_{l1}(u_l^{[\nu]})_{i,j} + D_{l2}(u_l^{[\nu]})_{i,j}), \end{aligned}$$

$$\begin{aligned} (\overline{\Sigma}_l^{[\nu]})_{i,j}(u_l^{[\nu+1]})_{i,j}^{[k+1/2]} &= (1/h^2) \left((D_{l3}(u_l^{[\nu]})_{i,j})(u_l^{[\nu+1]})_{i+1,j}^{[k]} + (D_{l1}(u_l^{[\nu]})_{i,j})(u_l^{[\nu+1]})_{i-1,j}^{[k+1]} \right. \\ &\quad \left. + (D_{l3}(u_l^{[\nu]})_{i,j})(u_l^{[\nu+1]})_{i,j+1}^{[k]} + (D_{l2}(u_l^{[\nu]})_{i,j})(u_l^{[\nu+1]})_{i,j-1}^{[k+1]} \right) \end{aligned}$$

$$D_{l1}(u_l^{[\nu]})_{i,j} = D_l(u_l^{[\nu]})_{i-1,j}, \quad D_{l2}(u_l^{[\nu]})_{i,j} = D_l(u_l^{[\nu]})_{i,j-1}, \quad D_{l3}(u_l^{[\nu]})_{i,j} = D_l(u_l^{[\nu]})_{i,j}.$$

We remark that other iterative techniques such as the line relaxation techniques or the preconditioned conjugate gradient method may also be used as inner solvers. However, the PCGS relaxation method appears a cheaper option. Finally, we note that the stabilizing terms $-\gamma_l \nabla \cdot \left(\frac{\nabla u_l^{[\nu+1]}}{|\nabla u_l^{[\nu]}|_\beta} \right)$ and $\sigma_{ll}^{[\nu]}$ lead the system (3.11) to be diagonally dominant and therefore using GS iterations is appropriate [45, 47, 16].

3.5. Method 5 – a primal-dual fixed-point (PDFP) method. In designing alternative methods for (2.4), we note that the previous four methods tackle the nonlinearity in some direct way. Below we consider an indirect way of treating the nonlinearity by reducing the high-order derivatives. In fact, high-order PDEs (in the context of mixed finite elements or in the denoising model [14]) as well as high-order ordinary differential equations are often reduced to low orders before numerical solution.

In order to apply this idea to (2.4), our first step is to introduce suitable intermediate variables (which we shall call *dual variables*)

$$v_1 = -\kappa(u_1) = -\nabla \cdot \frac{\nabla u_1}{|\nabla u_1|_\beta} \quad \text{and} \quad v_2 = -\kappa(u_2) = -\nabla \cdot \frac{\nabla u_2}{|\nabla u_2|_\beta},$$

leading to the equivalent system of four second-order nonlinear PDEs given by

$$(3.13) \quad \begin{cases} -\nabla \cdot \frac{\nabla u_1}{|\nabla u_1|_\beta} - v_1 = g_1 \\ -\nabla \cdot \frac{\nabla u_2}{|\nabla u_2|_\beta} - v_2 = g_2 \\ f_1(\mathbf{u}) - \alpha \nabla \cdot \left(\frac{\nabla v_1}{|\nabla u_1|_\beta} + \frac{\nabla u_1 \cdot (-\nabla v_1)}{|\nabla u_1|_\beta^3} \nabla u_1 \right) = g_3 \\ f_2(\mathbf{u}) - \alpha \nabla \cdot \left(\frac{\nabla v_2}{|\nabla u_2|_\beta} + \frac{\nabla u_2 \cdot (-\nabla v_2)}{|\nabla u_2|_\beta^3} \nabla u_2 \right) = g_4 \end{cases}$$

subject to the boundary conditions transferred into $\nabla u_l = 0$ and $\nabla v_l = 0$ for $l = 1, 2$ where $g_{\widehat{l}} = 0$ ($\widehat{l} = 1, \dots, 4$). The next step is to linearize (3.13) by a FP scheme as follows:

$$(3.14) \quad \mathbf{N}^{\text{PDFP}}[\mathbf{z}^{[\nu]}] \mathbf{z}^{[\nu+1]} = \mathbf{G}^{\text{PDFP}}[\mathbf{z}^{[\nu]}]$$

where linearization for $f_l(u_1^{[\nu+1]}, u_2^{[\nu+1]})$ is as in (3.8), $\mathbf{z}^{[\nu+1]} = (z_1^{[\nu+1]}, z_2^{[\nu+1]}, z_3^{[\nu+1]}, z_4^{[\nu+1]})^\top = (u_1^{[\nu+1]}, u_2^{[\nu+1]}, v_1^{[\nu+1]}, v_2^{[\nu+1]})^\top$,

$$\mathbf{N}^{\text{PDFP}}[\mathbf{z}^{[\nu]}] = \begin{bmatrix} -\widetilde{\mathcal{L}}_1[u_1^{[\nu]}] & 0 & -1 & 0 \\ 0 & -\widetilde{\mathcal{L}}_2[u_2^{[\nu]}] & 0 & -1 \\ \sigma_{11}^{[\nu]} & \sigma_{12}^{[\nu]} & -\alpha \widetilde{\mathcal{L}}_1[u_1^{[\nu]}] & 0 \\ \sigma_{21}^{[\nu]} & \sigma_{22}^{[\nu]} & 0 & -\alpha \widetilde{\mathcal{L}}_2[u_2^{[\nu]}] \end{bmatrix},$$

$$\mathbf{G}^{\text{PDFP}}[\mathbf{z}^{[\nu]}] = (g_1, g_2, \widehat{g}_3^{[\nu]}, \widehat{g}_4^{[\nu]})^\top,$$

$$\widetilde{\mathcal{L}}_l[u_i^{[\nu]}]z_i^{[\nu+1]} = \nabla \cdot \left(\frac{\widetilde{D}_l(u_i^{[\nu]})}{|\nabla u_i^{[\nu]}|_\beta} \nabla z_i^{[\nu+1]} \right) \quad (z_i^{[\nu+1]} = u_i^{[\nu+1]} \text{ or } v_i^{[\nu+1]}),$$

$$\widehat{g}_3^{[\nu]} = g_3 - f_1(u_1^{[\nu]}, u_2^{[\nu]}) + \sigma_{11}^{[\nu]}u_1^{[\nu]} + \sigma_{12}^{[\nu]}u_2^{[\nu]} + \alpha \nabla \cdot \left(\frac{\nabla u_1^{[\nu]} \cdot (-\nabla v_1^{[\nu]})}{|\nabla u_1^{[\nu]}|_\beta^3} \nabla u_1^{[\nu]} \right),$$

$$\widehat{g}_4^{[\nu]} = g_4 - f_2(u_1^{[\nu]}, u_2^{[\nu]}) + \sigma_{22}^{[\nu]}u_2^{[\nu]} + \sigma_{21}^{[\nu]}u_1^{[\nu]} + \alpha \nabla \cdot \left(\frac{\nabla u_2^{[\nu]} \cdot (-\nabla v_2^{[\nu]})}{|\nabla u_2^{[\nu]}|_\beta^3} \nabla u_2^{[\nu]} \right).$$

Here discretization of (3.14) is done as in Section 3. We shall call this numerical scheme a *primal-dual fixed-point*(PDFP) method because it includes the primal variables u_1, u_2 and the dual variables v_1, v_2 in a FP scheme. We remark that other choices of selecting the dual variables for (2.4) were also tested, but did not work well. For example, introducing the new variables

$$\vec{v}_1 = \frac{1}{|\nabla u_1|_\beta} \nabla \kappa(u_1) - \frac{\nabla u_1 \cdot \nabla \kappa(u_1)}{|\nabla u_1|_\beta^3} \nabla u_1$$

and

$$\vec{v}_2 = \frac{1}{|\nabla u_2|_\beta} \nabla \kappa(u_2) - \frac{\nabla u_2 \cdot \nabla \kappa(u_2)}{|\nabla u_2|_\beta^3} \nabla u_2$$

can only reduce the resulting PDEs to the third-order systems. We note further that in our numerical scheme each PDFP outer step is solved using a PCGS relaxation method as the inner linear solver (as with Method 4 of Section 3.4). Here, such an inner solution step is given by

$$(3.15) \quad (\mathbf{z}^{[\nu+1]})_{i,j}^{[k+1]} = (\mathbf{N}^{\text{PDFP}}[\mathbf{z}^{[\nu]}]_{i,j})^{-1} (\mathbf{G}^{\text{PDFP}}[\mathbf{z}^{[\nu]}]_{i,j})^{[k+1/2]},$$

where

$$(3.16) \quad \mathbf{N}^{\text{PDFP}}[\mathbf{z}^{[\nu]}]_{i,j} = \begin{bmatrix} \frac{(\widetilde{\Sigma}_1^{[\nu]})_{i,j}}{h^2} & 0 & -1 & 0 \\ 0 & \frac{(\widetilde{\Sigma}_2^{[\nu]})_{i,j}}{h^2} & 0 & -1 \\ (\sigma_{11}^{[\nu]})_{i,j} & (\sigma_{12}^{[\nu]})_{i,j} & \alpha \frac{(\widetilde{\Sigma}_1^{[\nu]})_{i,j}}{h^2} & 0 \\ (\sigma_{21}^{[\nu]})_{i,j} & (\sigma_{22}^{[\nu]})_{i,j} & 0 & \alpha \frac{(\widetilde{\Sigma}_2^{[\nu]})_{i,j}}{h^2} \end{bmatrix},$$

$$(3.17) \quad (\mathbf{G}^{\text{PDFP}}[\mathbf{z}^{[\nu]}]_{i,j})^{[k+1/2]} = \begin{pmatrix} (g_1)_{i,j} + (\widetilde{\Sigma}_1^{[\nu]})_{i,j} (u_1^{[\nu+1]})_{i,j}^{[k+1/2]} \\ (g_2)_{i,j} + (\widetilde{\Sigma}_2^{[\nu]})_{i,j} (u_2^{[\nu+1]})_{i,j}^{[k+1/2]} \\ (\widehat{g}_3)_{i,j}^{[\nu]} + \alpha (\widetilde{\Sigma}_1^{[\nu]})_{i,j} (v_1^{[\nu+1]})_{i,j}^{[k+1/2]} \\ (\widehat{g}_4)_{i,j}^{[\nu]} + \alpha (\widetilde{\Sigma}_2^{[\nu]})_{i,j} (v_2^{[\nu+1]})_{i,j}^{[k+1/2]} \end{pmatrix},$$

$$(3.18) \quad (\widetilde{\Sigma}_l^{[\nu]})_{i,j} = (2\widetilde{D}_{l3}(u_l^{[\nu]})_{i,j} + \widetilde{D}_{l1}(u_l^{[\nu]})_{i,j} + \widetilde{D}_{l2}(u_l^{[\nu]})_{i,j}),$$

$$(3.19) \quad (\widetilde{\Sigma}_l^{[\nu]})_{i,j} (z_i^{[\nu+1]})_{i,j}^{[k+1/2]} = \frac{1}{h^2} \left((\widetilde{D}_{l3}(u_l^{[\nu]})_{i,j}) (z_i^{[\nu+1]})_{i+1,j}^{[k]} \right. \\ \left. + (\widetilde{D}_{l1}(u_l^{[\nu]})_{i-1,j}) (z_i^{[\nu+1]})_{i-1,j}^{[k+1]} \right. \\ \left. + (\widetilde{D}_{l3}(u_l^{[\nu]})_{i,j}) (z_i^{[\nu+1]})_{i,j+1}^{[k]} \right. \\ \left. + (\widetilde{D}_{l2}(u_l^{[\nu]})_{i,j-1}) (z_i^{[\nu+1]})_{i,j-1}^{[k+1]} \right),$$

and

$$(3.20) \quad \begin{aligned} \tilde{D}_{l1}(u_l^{[\nu]})_{i,j} &= \tilde{D}_l(u_l^{[\nu]})_{i-1,j}, \\ \tilde{D}_{l2}(u_l^{[\nu]})_{i,j} &= \tilde{D}_l(u_l^{[\nu]})_{i,j-1}, \quad \tilde{D}_{l3}(u_l^{[\nu]})_{i,j} = \tilde{D}_l(u_l^{[\nu]})_{i,j}. \end{aligned}$$

Here the approximations in (3.15)–(3.20) need to be adjusted at the image boundary $\partial\Omega_h$ using the homogeneous Neumann boundary conditions, i.e.

$$(3.21) \quad (z_l^h)_{i,1} = (z_l^h)_{i,2}, \quad (z_l^h)_{i,n} = (z_l^h)_{i,n-1}, \quad (z_l^h)_{1,j} = (z_l^h)_{2,j}, \quad (z_l^h)_{n,j} = (z_l^h)_{n-1,j}.$$

Note that the above matrix \mathbf{N}^{PDFP} is invertible due to

$$\begin{aligned} \det(\mathbf{N}^{\text{PDFP}}) &= \alpha \frac{(\tilde{\Sigma}_1^{[\nu]})_{i,j}^2}{h^4} (\sigma_{22}^{[\nu]})_{i,j} + \alpha \frac{(\tilde{\Sigma}_2^{[\nu]})_{i,j}^2}{h^4} (\sigma_{11}^{[\nu]})_{i,j} + (\sigma_{11}^{[\nu]})_{i,j} (\sigma_{22}^{[\nu]})_{i,j} \\ &\quad + \left[\alpha^2 \frac{(\tilde{\Sigma}_1^{[\nu]})_{i,j}^2 (\tilde{\Sigma}_2^{[\nu]})_{i,j}^2}{h^8} - (\sigma_{12}^{[\nu]})_{i,j}^2 \right] > 0, \end{aligned}$$

since σ_{12} is relatively small and σ_{11}, σ_{22} are non-negative, so the last term is positive.

	Method	Rel. residual	Iterations	Run times (seconds)
Example 1 (32 × 32)	SSITM	4.1×10^{-2}	500	15.69
	SFP	5.8×10^{-2}	500	15.20
	PDFP	9.9×10^{-9}	334	13.57
Example 2 (32 × 32)	SSITM	4.2×10^{-3}	500	15.16
	SFP	7.2×10^{-3}	500	16.06
	PDFP	9.9×10^{-9}	182	7.53

TABLE 3.1

Run-time comparison (in seconds) of Method 2 (SSITM), Method 4 (SFP), and Method 5 (PDFP) for Example 1 (in 32 × 32 as shown in Figure 1.1 (a) – (b)) and Example 2 (as shown in Figure 1.3 (a) – (b)). Obviously, Method 5 (PDFP) is the fastest way in dropping the relative residual to 10^{-8} , while others are not successful. Here the maximum iteration for each method is 500.

We have so far presented five numerical methods for solving (2.4) where Method 2 is superseded by Method 4 and Method 3 is not recommended. So it remains to test the overall performances of the three numerical schemes (i.e. Methods 2, 4, 5). We tested them for both the non-smooth (Example 1) and the smooth (Example 2) registration problems as respectively shown in Figure 1.1 (a) – (b) and Figure 1.3 (a) – (b). We shall compare the relative residual (“Rel. residual”) and the relative SSD error (“Rel. SSD”) respectively defined by

$$(3.22) \quad \tilde{\varepsilon}_2 = \text{mean} \left\{ \frac{\|g_l^h - \mathcal{N}_l^h(\bar{\mathbf{z}}^h)\|_2}{\|g_l^h - \mathcal{N}_l^h(\bar{\mathbf{z}}_{initial}^h)\|_2} \mid l = 1, \dots, 4 \right\}, \quad \tilde{\varepsilon}_3 = \frac{\mathcal{D}^h(R^h, T^h(\bar{\mathbf{u}}^h))}{\mathcal{D}^h(R^h, T^h)}.$$

The test results represented in Figure 3.1 (a) – (d) and Table 3.1 show that the new Method 5 performs much better than the others, especially run times used for convergence to the minimizers of the minimization problem (1.2). Moreover, as shown

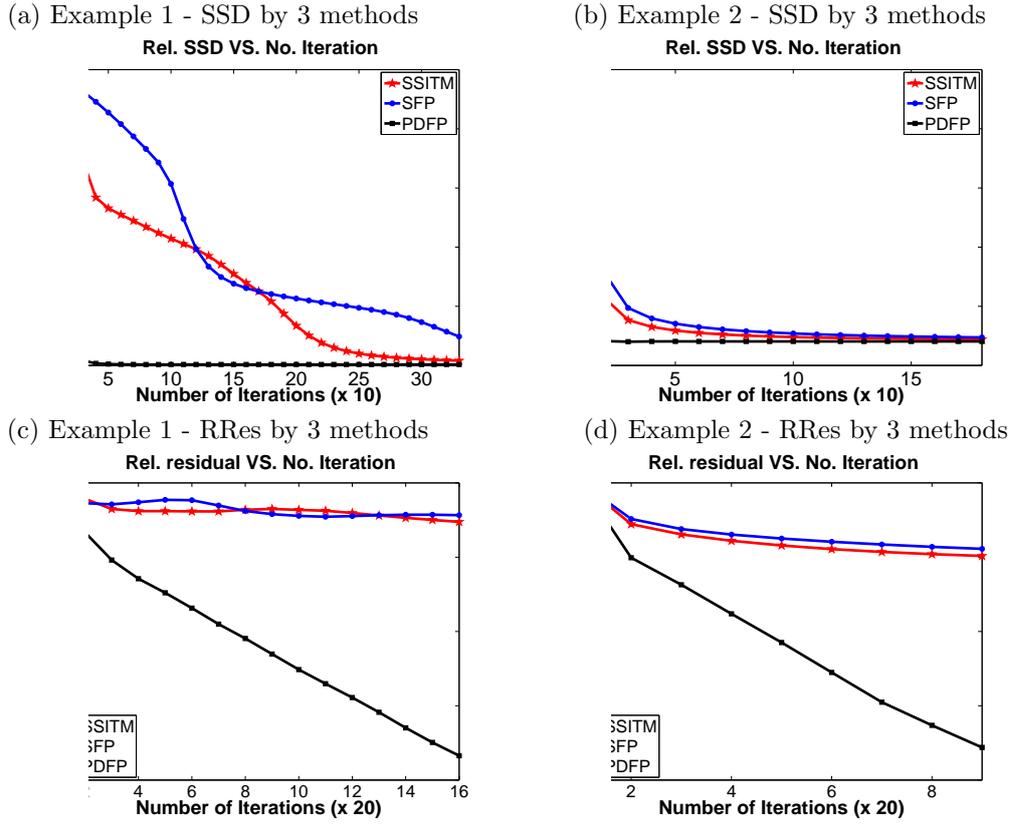


FIG. 3.1. Numerical results by Method 2 (SSITM (3.7)), Method 4 (SFP with $\gamma_l = 1/\sqrt{\beta}$), and Method 5 (PDFP) for Example 1 (in 32×32 as shown in Figure 1.1 (a) – (b)) and Example 2 (as shown in Figure 1.3 (a) – (b)). The top plots show the relative errors in SSD and the bottom plots show the relative residuals versus iterations. Clearly, Method 5 (PDFP) performs much better than the other two methods.

in Figure 3.2, Method 5 as expected produces a comparable residual for both the original system (2.4) and the equivalent system.

Although the above tests show that Method 5 is recommended as a unilevel method, our next task is to select a suitable smoother from these methods for designing a convergent MG method for (2.4). To proceed, we shall use a local Fourier analysis to decide which method (4 or 5) is better suited for our purpose. As it turns out, Method 5 is indeed the better method but, even so, modification is still needed for it to be an effective smoother.

4. A nonlinear multigrid method. Multigrid techniques [8, 30, 50, 52, 53] have been proved to be very useful in the context of deformable image registration for solving large systems of linear or nonlinear equations arising from high-resolution digital images and real-life applications. The basic idea of a multigrid method is to smooth high frequency components of the error of the solution on a fine grid by performing a few steps with a smoother (an iterative relaxation technique) such that a smooth error term can be well represented and approximated on a coarser grid. After a 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.

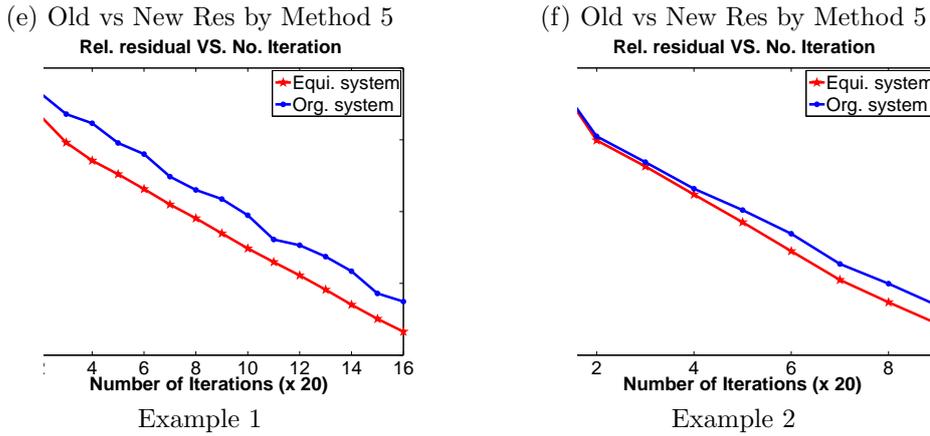


FIG. 3.2. Comparison of the relative residuals by Method 5 using both the original system (2.4) and the equivalent system (3.13).

Finally, the smoother is performed again in order to remove some new high frequency components of the error introduced by the interpolation. This is known as a two-grid cycle, and with recursive application it can be extended to a multigrid method.

A working MG has 3 main components: (i) Smoothing via an iterative method; (ii) Restriction from a fine grid to a coarse grid; (iii) interpolation from a coarse grid to a fine one. On the coarsest grid, an effective unilevel solver is used for accurate solution; here we shall use Method 5. Without reducing the importance of the restriction and interpolation operators, the efficiency of every MG method strongly relies on the efficiency of the *smoother* used at each level. We shall first discuss the choice of our smoothers before presenting an overall algorithm.

4.1. 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 versions of (2.4) and (3.13). 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 [50]. This approach has proved to be very useful in the understanding of MG methods when solving nonlinear problems; see for instance [3, 4, 9, 10, 12, 13, 28, 27, 32, 40, 48] for interesting examples and discussions. Note that although LFA does not consider boundary conditions, the boundary conditions used here in our model do not affect the analysis.

For linear problems, iterative methods such as damped Jacobi or Gauss-Seidel (GS) methods are usually enough to rapidly reduce high frequencies of the underlying error. However for nonlinear problems, *non-standard smoothers* are often required and their efficiency in *smoothing* is the decisive factor in determining whether a given MG is convergent or not. For nonlinear and anisotropic problems such as (2.4) and (3.13), developing such an effective smoother is by no means a trivial task. 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 [50, 53]. Below we shall first analyze Methods 4, 5 before considering improvements.

4.1.1. Analysis of Method 4 (Smoother 1). Here we will compute the smoothing factor of Method 4 iterations (as our Smoother 1 shortly) applied to the linearized system $\mathbf{N}_h^{\text{SFP}}[\bar{\mathbf{u}}^h]\mathbf{u}^h = \mathbf{G}_h^{\text{SFP}}[\bar{\mathbf{u}}^h]$ obtained by freezing coefficients in (3.11) at some outer step. Here \mathbf{u}^h and $\bar{\mathbf{u}}^h$ denote the exact solution and the current approximation and $\mathbf{N}_h^{\text{SFP}}[\bar{\mathbf{u}}^h]$ and $\mathbf{G}_h^{\text{SFP}}[\bar{\mathbf{u}}^h]$ the resulting discrete operators from the linearization at $\bar{\mathbf{u}}^h$. The analysis is carried out over the infinite grid

$$(4.1) \quad \Omega_h^\infty = \{\mathbf{x} \in \Omega \mid \mathbf{x} = (x_{1_i}, x_{2_j})^\top = ((2i-1)h/2, (2j-1)h/2)^\top, i, j \in \mathbb{Z}^2\}.$$

Let $\varphi_h(\boldsymbol{\theta}, \mathbf{x}) = \exp(\mathbf{i}\boldsymbol{\theta}\mathbf{x}/h) \cdot \hat{\mathbf{\Gamma}}$ be grid functions, where $\hat{\mathbf{\Gamma}} = (1, 1)^\top$, $\boldsymbol{\theta} = (\theta_1, \theta_2)^\top \in \Theta = [-\pi, \pi]^2$, $\mathbf{x} \in \Omega_h^\infty$, 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., μ is matrix with its (i, j) entry representing the smoothing factor for grid point $\xi = (i, j)$. Hence we define $\mu_{\text{loc}} = \mu(\xi)$ as the local smoothing factor and $\bar{\mu}_{\text{loc}}$ as the worst possible value of μ_{loc} over Ω_h . Thus for Method 4 from (3.12)

$$(4.2) \quad \bar{\mu}_{\text{loc}}^{\text{SFP}} = \max_{\xi \in \Omega_h} \mu_{\text{loc}}^{\text{SFP}}.$$

To determine $\mu_{\text{loc}}^{\text{SFP}}$ we consider the local discrete system $\mathbf{N}_h^{\text{SFP}}(\xi)\mathbf{u}^h(\xi) = \mathbf{G}_h^{\text{SFP}}(\xi)$ centered and defined only within a small neighborhood of ξ and $\mathbf{u}^h(\xi) = [u_1^h(\xi), u_2^h(\xi)]$. By using the splitting $\mathbf{N}_h^{\text{SFP}}(\xi) = \mathbf{N}_h^{\text{SFP}+}(\xi) + \mathbf{N}_h^{\text{SFP}-}(\xi)$, it is possible to write the local inner iterations of Method 4 as

$$(4.3) \quad \mathbf{N}_h^{\text{SFP}+}(\xi)\bar{\mathbf{u}}_{\text{new}}^h(\xi) + \mathbf{N}_h^{\text{SFP}-}(\xi)\bar{\mathbf{u}}_{\text{old}}^h(\xi) = \mathbf{G}_h^{\text{SFP}}(\xi)$$

where $\bar{\mathbf{u}}_{\text{old}}^h(\xi)$ and $\bar{\mathbf{u}}_{\text{new}}^h(\xi)$ stand for the approximations to $\mathbf{u}^h(\xi)$ before and after the inner smoothing step, respectively. Here

$$\begin{aligned} \mathbf{N}_h^{\text{SFP}+}(\xi) &= \begin{bmatrix} -\alpha\mathcal{L}_1^{h[+]}(\xi) & \sigma_{12}(\xi) \\ \sigma_{21}(\xi) & -\alpha\mathcal{L}_2^{h[+]}(\xi) \end{bmatrix}, \\ \mathbf{N}_h^{\text{SFP}-}(\xi) &= \begin{bmatrix} -\alpha\mathcal{L}_1^{h[-]}(\xi) & 0 \\ 0 & -\alpha\mathcal{L}_2^{h[-]}(\xi) \end{bmatrix}, \\ -\mathcal{L}_l^{h[+]}(\xi) &= \frac{1}{h^2} \begin{bmatrix} 0 & 0 & 0 \\ -D_{l2}(\bar{u}_l(\xi)) & \Sigma_l(\xi) + (h^2/\alpha)\sigma_{ll}(\xi) & 0 \\ 0 & -D_{l1}(\bar{u}_l(\xi)) & 0 \end{bmatrix}, \end{aligned}$$

and

$$-\mathcal{L}_l^{h[-]}(\xi) = \frac{1}{h^2} \begin{bmatrix} 0 & -D_{l3}(\bar{u}_l(\xi)) & 0 \\ 0 & 0 & -D_{l3}(\bar{u}_l(\xi)) \\ 0 & 0 & 0 \end{bmatrix}.$$

By subtracting (4.3) from $\mathbf{N}_h^{\text{SFP}}(\xi)\mathbf{u}^h(\xi) = \mathbf{G}_h^{\text{SFP}}(\xi)$ and defining $\bar{\mathbf{e}}_{\text{new}}^h(\xi) = \mathbf{u}^h(\xi) - \bar{\mathbf{u}}_{\text{new}}^h(\xi)$ and $\bar{\mathbf{e}}_{\text{old}}^h(\xi) = \mathbf{u}^h(\xi) - \bar{\mathbf{u}}_{\text{old}}^h(\xi)$ we obtain the local system of error equations

$$(4.4) \quad \mathbf{N}_h^{\text{SFP}+}(\xi)\bar{\mathbf{e}}_{\text{new}}^h(\xi) + \mathbf{N}_h^{\text{SFP}-}(\xi)\bar{\mathbf{e}}_{\text{old}}^h(\xi) = 0$$

or

$$(4.5) \quad \bar{\mathbf{e}}_{new}^h(\xi) = -[\mathbf{N}_h^{\text{SFP}+}(\xi)]^{-1} [\mathbf{N}_h^{\text{SFP}-}(\xi)] \bar{\mathbf{e}}_{old}^h(\xi) = \mathbf{S}_h^{\text{SFP}}(\xi) \bar{\mathbf{e}}_{old}^h(\xi)$$

where $\mathbf{S}_h^{\text{SFP}}(\xi)$ is the amplification factor. The effect of $\mathbf{S}_h^{\text{SFP}}(\xi)$ on the grid functions $\varphi_h(\boldsymbol{\theta}, \mathbf{x})$ within $\boldsymbol{\Theta}_{\text{high}} = \boldsymbol{\Theta} \setminus [-\pi/2, \pi/2]^2$ will determine the smoothing properties of Method 4. Thus, provided PCGS is used as the inner solver, $-\mathcal{L}_l^{h[+]}\!(\xi, \boldsymbol{\theta})$ and $-\mathcal{L}_l^{h[-]}\!(\xi, \boldsymbol{\theta})$ are defined by

$$-\mathcal{L}_l^{h[+]}\!(\xi, \boldsymbol{\theta}) = \frac{1}{h^2} (\Sigma_l(\xi) + (h^2/\alpha)\sigma_{ll}(\xi) - D_{l1}(\xi) \exp(-\mathbf{i}\theta_1) - D_{l2}(\xi) \exp(-\mathbf{i}\theta_2))$$

and

$$-\mathcal{L}_l^{h[-]}\!(\xi, \boldsymbol{\theta}) = -\frac{1}{h^2} (D_{l3}(\xi) (\exp(\mathbf{i}\theta_1) + \exp(\mathbf{i}\theta_2))).$$

and the local smoothing factor is

$$(4.6) \quad \mu_{\text{loc}}^{\text{SFP}} = \sup\{|\rho(\mathbf{S}_h^{\text{SFP}}(\xi, \boldsymbol{\theta}))| : \boldsymbol{\theta} \in \boldsymbol{\Theta}_{\text{high}}\}$$

where ρ indicates the spectral radius of $\mathbf{S}_h^{\text{SFP}}(\xi, \boldsymbol{\theta})$.

On a discrete grid of $\boldsymbol{\Theta}_{\text{high}}$, we shall be able to estimate the above factor shortly.

4.1.2. Analysis of Method 5 (Smoother 2). Now we consider the smoothing factor of Method 5 from (3.14). To this end $\mathbf{N}_h^{\text{PDFP}}[\bar{\mathbf{z}}^h] \mathbf{z}^h = \mathbf{G}_h^{\text{PDFP}}[\bar{\mathbf{z}}^h]$ will denote the linearized system with \mathbf{z}^h and $\bar{\mathbf{z}}^h$ the exact solution and current approximation. Here the grid function is defined by $\varphi_h(\boldsymbol{\theta}, \mathbf{x}) = \exp(\mathbf{i}\boldsymbol{\theta}\mathbf{x}/h) \cdot \hat{\mathbf{I}}$, where $\hat{\mathbf{I}} = (1, 1, 1, 1)^\top$. The local inner iterations for the PDFP algorithm can therefore be written as

$$(4.7) \quad \mathbf{N}_h^{\text{PDFP}+}(\xi) \bar{\mathbf{z}}_{new}^h(\xi) + \mathbf{N}_h^{\text{PDFP}-}(\xi) \bar{\mathbf{z}}_{old}^h(\xi) = \mathbf{G}_h^{\text{PDFP}}(\xi)$$

where

$$\mathbf{N}_h^{\text{PDFP}+}(\xi) = \begin{bmatrix} -\tilde{\mathcal{L}}_1^{h[+]}\!(\xi) & 0 & -1 & 0 \\ 0 & -\tilde{\mathcal{L}}_2^{h[+]}\!(\xi) & 0 & -1 \\ \sigma_{11}(\xi) & \sigma_{12}(\xi) & -\alpha\tilde{\mathcal{L}}_1^{h[+]}\!(\xi) & 0 \\ \sigma_{21}(\xi) & \sigma_{22}(\xi) & 0 & -\alpha\tilde{\mathcal{L}}_2^{h[+]}\!(\xi) \end{bmatrix},$$

$$\mathbf{N}_h^{\text{PDFP}-}(\xi) = \begin{bmatrix} -\tilde{\mathcal{L}}_1^{h[-]}\!(\xi) & 0 & 0 & 0 \\ 0 & -\tilde{\mathcal{L}}_2^{h[-]}\!(\xi) & 0 & 0 \\ 0 & 0 & -\alpha\tilde{\mathcal{L}}_1^{h[-]}\!(\xi) & 0 \\ 0 & 0 & 0 & -\alpha\tilde{\mathcal{L}}_2^{h[-]}\!(\xi) \end{bmatrix},$$

$$-\tilde{\mathcal{L}}_l^{h[+]}\!(\xi) = \frac{1}{h^2} \begin{bmatrix} 0 & 0 & 0 \\ -\tilde{D}_{l2}(\bar{u}_l(\xi)) & \tilde{\Sigma}_l(\xi) & 0 \\ 0 & -\tilde{D}_{l1}(\bar{u}_l(\xi)) & 0 \end{bmatrix},$$

and

$$-\tilde{\mathcal{L}}_l^{h[-]}\!(\xi) = \frac{1}{h^2} \begin{bmatrix} 0 & -\tilde{D}_{l3}(\bar{u}_l(\xi)) & 0 \\ 0 & 0 & -\tilde{D}_{l3}(\bar{u}_l(\xi)) \\ 0 & 0 & 0 \end{bmatrix}.$$

Following the similar process of subtracting (4.7) from $\mathbf{N}_h^{\text{PDFP}}(\xi)\mathbf{z}^h(\xi) = \mathbf{G}_h^{\text{PDFP}}(\xi)$ one obtains the system of local error equations

$$\mathbf{N}_h^{\text{PDFP}^+}(\xi)\bar{\mathbf{e}}_{new}^h(\xi) + \mathbf{N}_h^{\text{PDFP}^-}(\xi)\bar{\mathbf{e}}_{old}^h(\xi) = 0$$

or $\bar{\mathbf{e}}_{new}^h(\xi) = \mathbf{S}_h^{\text{PDFP}}(\xi)\bar{\mathbf{e}}_{old}^h(\xi)$ where $\bar{\mathbf{e}}_{old}^h(\xi) = \mathbf{z}^h(\xi) - \bar{\mathbf{z}}_{old}^h(\xi)$ and $\bar{\mathbf{e}}_{new}^h(\xi) = \mathbf{z}^h(\xi) - \bar{\mathbf{z}}_{new}^h(\xi)$ are the error functions and $\mathbf{S}_h^{\text{PDFP}}(\xi) = -[\mathbf{N}_h^{\text{PDFP}^+}(\xi)]^{-1}[\mathbf{N}_h^{\text{PDFP}^-}(\xi)]$. Hence, by considering the grid functions $\varphi_h(\boldsymbol{\theta}, \mathbf{x})$, we can represent $\tilde{\mathcal{L}}_l^{h[+]}\!(\xi, \boldsymbol{\theta})$ and $\tilde{\mathcal{L}}_l^{h[-]}\!(\xi, \boldsymbol{\theta})$ by

$$-\tilde{\mathcal{L}}_l^{h[+]}\!(\xi, \boldsymbol{\theta}) = \frac{1}{h^2}(\tilde{\Sigma}_l(\xi) - \tilde{D}_{l1}(\xi)\exp(-\mathbf{i}\theta_1) - \tilde{D}_{l2}(\xi)\exp(-\mathbf{i}\theta_2))$$

and

$$-\tilde{\mathcal{L}}_l^{h[-]}\!(\xi, \boldsymbol{\theta}) = -\frac{1}{h^2}(\tilde{D}_{l3}(\xi)(\exp(\mathbf{i}\theta_1) + \exp(\mathbf{i}\theta_2))).$$

From here, the PDFP local smoothing factor is defined by

$$(4.8) \quad \mu_{loc}^{\text{PDFP}} = \sup\{|\rho(\mathbf{S}_h^{\text{PDFP}}(\xi, \boldsymbol{\theta}))| : \boldsymbol{\theta} \in \boldsymbol{\Theta}_{\text{high}}\}.$$

The effectiveness of the above 2 smoothers (i.e. Methods 4, 5) is now tested by computing their smooth rates for Examples 1–2. The following Table 4.1 summarizes the smoothing factors of Smoother 1 (SFP) and Smoother 2 (PDFP) for Examples 1–2. Clearly for the smooth Example 2, both Smoothers 1 and 2 are effective and in

Smoother	Example 1 (non-smooth)	Example 2 (smooth)
1	0.9410	0.6825
2	0.9412	0.5212

TABLE 4.1

Smoothing factors $\bar{\mu}_{loc}$ after 10 outer iterations with $PCGSiter = 10$ by the SFP- and PDFP-type smoothers for the smooth and non-smooth registration problems in Examples 1–2 as shown respectively in Figures 1.1 (a) – (b) and 1.3 (a) – (b).

particular Smoother 2 is better than Smoother 1. But for the non-smooth Example 1, they are much less efficient. Next we consider a method to improve the smoothers and primarily to improve Smoother 2.

4.2. A new smoother and its analysis (Smoother 2*). Recall that μ is a matrix of amplification factors, whose maximum defines the smoothing factor as in (4.6) and (4.8). It turns out that the largest entries of μ coincide with locations where we observe strong jumps of the diffusion coefficients $\tilde{D}_l(\xi)$. Therefore, our idea of modifying the smoothers is to seek alternative ways to update the solutions where the diffusion coefficients have large jumps. Denote by set W all those pixels with such large coefficients jumps. The whole domain $\Omega_h = W \cup (\Omega_h \setminus W)$ admits two different iterative solvers.

We consider an under-relaxation idea for the sub-domain W (representing the jumps of $\tilde{D}_l(\xi)$) by updating all these odd points by

$$(4.9) \quad (\mathbf{z}^{[\nu+1]})_{i,j}^{[k+1]} = (1 - \omega)(\mathbf{z}^{[\nu+1]})_{i,j}^{[k]} + \omega \underbrace{(\mathbf{N}^{\text{PDFP}}[\mathbf{z}^{[\nu]}]_{i,j})^{-1}(\mathbf{G}^{\text{PDFP}}[\mathbf{z}^{[\nu]}]_{i,j})^{[k+1/2]}}_{\text{Standard PCGS step}}$$

where ω is to be chosen next. As with the previous section, we can analyze the smoothing factor for the ω -PCGS relaxation method in (4.9) by the LFA in the similar way to (3.15). Here

$$-\tilde{\mathcal{L}}_l^{h[+]}(\xi, \boldsymbol{\theta}) \quad \text{and} \quad -\tilde{\mathcal{L}}_l^{h[-]}(\xi, \boldsymbol{\theta})$$

are given by

$$(4.10) \quad -\tilde{\mathcal{L}}_l^{h[+]}(\xi, \boldsymbol{\theta}) = \frac{1}{h^2} (\tilde{\Sigma}_l(\xi) - \omega \tilde{D}_{l1}(\xi) \exp(-\mathbf{i}\theta_1) - \omega \tilde{D}_{l2}(\xi) \exp(-\mathbf{i}\theta_2))$$

and

$$(4.11) \quad -\tilde{\mathcal{L}}_l^{h[-]}(\xi, \boldsymbol{\theta}) = \frac{1}{h^2} ((1 - \omega) \tilde{\Sigma}_l(\xi) - \omega \tilde{D}_{l3}(\xi) (\exp(\mathbf{i}\theta_1) + \exp(\mathbf{i}\theta_2))).$$

Further with the updated formulae for $\tilde{\mathcal{L}}_l^{h[+]}$, $\tilde{\mathcal{L}}_l^{h[-]}$ at set W , the amplification factor $\rho(\mathbf{S}_h^{\text{PDFP II}})$ is similarly defined using the updated formulae for $\mathbf{S}_h^{\text{PDFP II}}(\xi) = -[\mathbf{N}_h^{\text{PDFP}+}(\xi)]^{-1} [\mathbf{N}_h^{\text{PDFP}-}(\xi)]$. Finally the overall smoothing factor is

$$\mu_{\text{loc}}^{\text{PDFP II}} = \max \left\{ \sup_{\xi \in W} \{ |\rho(\mathbf{S}_h^{\text{PDFP II}}(\xi, \boldsymbol{\theta}))| : \boldsymbol{\theta} \in \boldsymbol{\Theta}_{\text{high}} \}, \right. \\ \left. \sup_{\xi \in \Omega_h \setminus W} \{ |\rho(\mathbf{S}_h^{\text{PDFP}}(\xi, \boldsymbol{\theta}))| : \boldsymbol{\theta} \in \boldsymbol{\Theta}_{\text{high}} \} \right\}.$$

For completeness, we also applied this idea of introducing ω in W for Smoother 1 (SFP from (3.12)) and did a similar LFA analysis. For the same test examples as with Table 4.1, we now show the improved smoothing rates computed for the modified smoothers in Table 4.2 where we name the modified Smoother 2 (i.e. Method 5, PDFP II) as Smoother 2* and the modified Smoother 1 (i.e. from Method 4, SFP II) as Smoother 1*. Clearly we see that the above under-relaxation idea does improve Smoothers 1, 2; since more improvement can be observed in Smoother 2* (PDFP II) over Smoother 2 (PDFP), we shall take Smoother 2* as our recommended smoother.

Smoother	Example 1 (non-smooth)	Example 2 (smooth)
1* ($\omega = 0.7$)	0.8324	0.6711
2* ($\omega = 0.7$)	0.7613	0.5210

TABLE 4.2

Improved smoothing factors $\bar{\mu}_{\text{loc}}$ after using ω under-relaxation idea in sub-domain W Examples 1 – 2.

4.3. Nonlinear Multigrid Algorithm. Full approximation scheme based nonlinear multigrid method (FAS-NMG) has become an efficient approach for solving nonlinear problems, in particular image processing applications. Here instead of a scalar PDE we have a coupled system of 4 nonlinear PDEs from (3.13):

$$\mathcal{N}(\mathbf{z}^h) = \mathbf{g}^h, \quad i.e. \quad \begin{cases} \mathcal{N}_1^h(\mathbf{z}^h) = g_1^h \\ \vdots \\ \mathcal{N}_4^h(\mathbf{z}^h) = g_4^h \end{cases}$$

involving the nonlinear partial differential operator $\mathcal{N}_l^h(\mathbf{z}^h)$ given by the left-hand side of (3.13), where $g_l = 0$ on the finest grid, for $\hat{l} = 1, \dots, 4$.

Let $\bar{\mathbf{z}}^h = (\bar{z}_1^h, \bar{z}_2^h, \bar{z}_3^h, \bar{z}_4^h)^\top$ be the approximation of \mathbf{z}^h after a few smoothing iterations in a **pre-smoothing step** on a fine-grid problem. Then, the algebraic error \mathbf{e}^h of the solution is given by $\mathbf{e}^h = \mathbf{z}^h - \bar{\mathbf{z}}^h$ where we denote by \mathbf{z}^h the exact solution of (3.13). The residual equation system for the \hat{l} th equation is given by

$$\mathcal{N}_l^h(\bar{\mathbf{z}}^h + \mathbf{e}^h) - \mathcal{N}_l^h(\bar{\mathbf{z}}^h) = g_l^h - \mathcal{N}_l^h(\bar{\mathbf{z}}^h) = r_l^h.$$

In order to correct the approximated solution $\bar{\mathbf{z}}^h$ on the fine grid, one needs to compute the error \mathbf{e}^h . The computation of \mathbf{e}^h is prohibitively expensive and cannot be computed directly on the fine grid. However, 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:

$$(4.12) \quad \underbrace{\mathcal{N}_l^h(\bar{\mathbf{z}}^h + \mathbf{e}^h)}_{\mathcal{N}_l^h(\mathbf{z}^h)} = \underbrace{r_l^h + \mathcal{N}_l^h(\bar{\mathbf{z}}^h)}_{g_l^h} \rightarrow \underbrace{\mathcal{N}_l^H(\bar{\mathbf{z}}^H + \mathbf{e}^H)}_{\mathcal{N}_l^H(\mathbf{z}^H)} = \underbrace{r_l^H + \mathcal{N}_l^H(\bar{\mathbf{z}}^H)}_{g_l^H}$$

where $H = 2h$ is the new cell size $H \times H$ with $H \geq h$ and $g_l^H \neq 0$ on the coarse grid. After the nonlinear residual equation (4.12) on the coarse grid have been solved with a method of our choice, the coarse-grid correction $\mathbf{e}^H = \mathbf{z}^H - \bar{\mathbf{z}}^H$ is then interpolated back to the fine grid \mathbf{e}^h that can now be used for updating the approximated solution $\bar{\mathbf{z}}^h$ of the original system on the fine grid $\bar{\mathbf{z}}_{new}^h = \bar{\mathbf{z}}^h + \mathbf{e}^h$ (**coarse-grid correction step**). The last step for a FAS-NMG method is to perform the smoother again to remove high frequency parts of the interpolated error (**post-smoothing step**).

We now define our multigrid components for solving (2.4) via (3.13). Firstly, standard coarsening is used in computing the coarse-grid domain Ω_H by doubling the grid size in each space direction, i.e. $h \rightarrow 2h = H$. Secondly for intergrid transfer operators between Ω_h and Ω_H , the averaging and bi-linear interpolation techniques are used for the restriction and interpolation operators denoted respectively by I_h^H and I_H^h ; see the details in [8, 30, 50, 52, 53]. In order to compute the coarse-grid operator of $\mathcal{N}_l^h(\mathbf{z}^h)$ consisting of two main parts: $f_l^h(z_1^h, z_2^h)$ and $\tilde{\mathcal{L}}_l^h(z_l^h)$, a so-called *discretization coarse grid approximation* (DCA) is performed [8, 50, 53]. The idea is to re-discretize the Euler-Lagrange system directly. In the case of $f_l^H(z_1^H, z_2^H)$, we first use the restriction operator for both components of the deformation field \mathbf{z}^h , i.e. z_1^h and z_2^h , and the given images, R^h and T^h , and then compute the corresponding coarse-grid part of $f_l^H(z_1^H, z_2^H)$. For $\tilde{\mathcal{L}}_l^H(z_l^H)$, the corresponding coarse-grid part of $\tilde{\mathcal{L}}_l^h(z_l^h)$ is obtained using z_l^H and a DCA.

We now present our recommended Smoother 2* (modified Method 5) as an algorithm before presentation of the overall algorithm for solving (2.4).

ALGORITHM 1 (Recommended Smoother 2* – PDFP II).

Denote by

α	the regularization parameter
ω	the relaxation parameter
$K > 0$	a tolerance (typically $K = 10$)
<i>PCGSiter</i>	the maximum number of PCGS iterations

$$[\bar{\mathbf{z}}^h] \leftarrow \text{Smoother}(\bar{\mathbf{z}}^h, g_1^h, g_2^h, g_3^h, g_4^h, R^h, T^h, \alpha, \omega, K, \text{PCGSiter})$$

-
- Use input parameters to compute $(\sigma_{lm})_{i,j}$, $(\mathbf{G}_h^{\text{PDFP}}[\bar{\mathbf{z}}^h])_{i,j}$,
and $(\mathbf{N}_h^{\text{PDFP}}[\bar{\mathbf{z}}^h]_{i,j})^{-1}$ for $l, m = 1, 2$ and $1 \leq i, j \leq n$
 - Perform PCGS steps
 - for $k = 1 : \text{PCGSiter}$
 - for $i = 1 : n$
 - for $j = 1 : n$
 - if $\tilde{D}_l(\bar{u}_l)_{i,j} \geq K \cdot \text{mean}\{\tilde{D}_{l1}(\bar{u}_l)_{i,j}, \tilde{D}_{l2}(\bar{u}_l)_{i,j}, \tilde{D}_3(\bar{u}_l)_{i,j}\}$
for $l = 1$ or 2
 - Set $\omega = 0.7$
 - else
 - Set $\omega = 0$
 - end
 - Compute $(\tilde{\mathbf{z}}^h)_{i,j}^{[k+1]}$ using (3.15) and set
 $(\bar{\mathbf{z}})_{i,j}^{[k+1]} = (1 - \omega)(\bar{\mathbf{z}})_{i,j}^{[k]} + \omega(\tilde{\mathbf{z}}^h)_{i,j}^{[k+1]}$
-

To solve (3.13) numerically, our FAS-NMG method with the proposed MG smoother given by Algorithm 1 is applied recursively down to the coarsest grid consisting of a small number of grid points, typically 8×8 . A pseudo-code implementation of our FAS multigrid method is then summarized in the following algorithm:

ALGORITHM 2 (FAS-NMG Algorithm).

Denote FAS-NMG parameters as follows:

- ν_1 the number of pre-smoothing steps on each level
- ν_2 the number of post-smoothing steps on each level
- μ the number of multigrid cycles on each level ($\mu = 1$ for V-cycling and $\mu = 2$ for W-cycling).
- [Here we present the V-cycle with $\mu = 1$.]
- α the regularization parameter
- ω the relaxation parameter
- $K > 0$ tolerance
- PCGSiter* the maximum number of iterations using a smoother

$$\bar{\mathbf{z}}^h \leftarrow \text{FAS-NMG}(\bar{\mathbf{z}}^h, \alpha, \vec{\varepsilon})$$

-
- Select $\alpha, \vec{\varepsilon} = (\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4)$ and
initial guess solutions $\bar{\mathbf{z}}_{\text{initial}}^h = (\bar{z}_1^h, \bar{z}_2^h, \bar{z}_3^h, \bar{z}_4^h)^\top$ on the finest grid
 - Set $K = 0$, $[\bar{\mathbf{z}}^h]^K = \bar{\mathbf{z}}_{\text{initial}}^h$, $\tilde{\varepsilon}_2 = \varepsilon_2 + 1$, $\tilde{\varepsilon}_3 = \varepsilon_3 + 1$, and $\tilde{\varepsilon}_4 = \varepsilon_4 + 1$
 - While ($K < \varepsilon_1$ AND $\tilde{\varepsilon}_2 > \varepsilon_2$ AND $\tilde{\varepsilon}_3 > \varepsilon_3$ AND $\tilde{\varepsilon}_4 > \varepsilon_4$)
 - $[\bar{\mathbf{z}}^h]^{K+1} \leftarrow \text{FASCYC}(\bar{\mathbf{z}}^h, g_1^h, g_2^h, g_3^h, g_4^h, R^h, T^h, \nu_1, \nu_2, \alpha, \omega, \text{PCGSiter})$
 - Compute $\tilde{\varepsilon}_2$ and $\tilde{\varepsilon}_3$ at step $K + 1$ using (3.22),
[Recall that $\mathcal{D}^h(R^h, T^h(\cdot)) \sim \frac{h^2}{2} \|R^h, T^h(\cdot)\|_2^2$]
 - Compute the difference $\tilde{\varepsilon}_4 = |\mathcal{D}^h(R^h, T^h((\bar{\mathbf{u}}^h)^{K+1})) - \mathcal{D}^h(R^h, T^h((\bar{\mathbf{u}}^h)^K))|$
 - Set $K = K + 1$
 - end
-

where

$$[\bar{\mathbf{z}}^h] \leftarrow \text{FASCYC}(\bar{\mathbf{z}}^h, g_1^h, g_2^h, g_3^h, g_4^h, R^h, T^h, \nu_1, \nu_2, \alpha, \omega, K, \text{PCGSiter})$$

-
- If $\Omega_h = \text{coarset grid}$ ($|\Omega_h| = 8 \times 8$), solve (3.13) using Algorithm 1 and then stop. Else continue with following step.
 - Pre-smoothing:
For $k = 1$ to ν_1 , $[\bar{\mathbf{z}}^h] \leftarrow \text{Smoother}(\bar{\mathbf{z}}^h, g_1^h, g_2^h, g_3^h, g_4^h, R^h, T^h, \alpha, \omega, K, \text{PCGSiter})$
 - Restriction to the coarse grid:
 $\bar{\mathbf{z}}_1^H \leftarrow I_h^H \bar{\mathbf{z}}_1^h, \bar{\mathbf{z}}_2^H \leftarrow I_h^H \bar{\mathbf{z}}_2^h, \bar{\mathbf{z}}_3^H \leftarrow I_h^H \bar{\mathbf{z}}_3^h, \bar{\mathbf{z}}_4^H \leftarrow I_h^H \bar{\mathbf{z}}_4^h, R^H \leftarrow I_h^H R^h, T^H \leftarrow I_h^H T^h$
 - Set the initial solution for the coarse-grid problem:
 $[\bar{\mathbf{z}}_1^H, \bar{\mathbf{z}}_2^H, \bar{\mathbf{z}}_3^H, \bar{\mathbf{z}}_4^H] \leftarrow [\bar{\mathbf{z}}_1^H, \bar{\mathbf{z}}_2^H, \bar{\mathbf{z}}_3^H, \bar{\mathbf{z}}_4^H]$
 - Compute the new right-hand side for the coarse-grid problem:
 $g_1^H \leftarrow I_h^H (g_1^h - \mathcal{N}_1^h(\bar{\mathbf{z}}^h)) + \mathcal{N}_1^H(\bar{\mathbf{z}}^H), \quad g_2^H \leftarrow I_h^H (g_2^h - \mathcal{N}_2^h(\bar{\mathbf{z}}^h)) + \mathcal{N}_2^H(\bar{\mathbf{z}}^H),$
 $g_3^H \leftarrow I_h^H (g_3^h - \mathcal{N}_3^h(\bar{\mathbf{z}}^h)) + \mathcal{N}_3^H(\bar{\mathbf{z}}^H), \quad g_4^H \leftarrow I_h^H (g_4^h - \mathcal{N}_4^h(\bar{\mathbf{z}}^h)) + \mathcal{N}_4^H(\bar{\mathbf{z}}^H)$
 - Implement the FAS multigrid on the coarse-grid problem:
For $k = 1$ to μ ,
 $[\bar{\mathbf{z}}^H] \leftarrow \text{FASCYC}(\bar{\mathbf{z}}^H, g_1^H, g_2^H, g_3^H, g_4^H, R^H, T^H, \nu_1, \nu_2, \alpha, \omega, K, \text{PCGSiter})$
 - Add the coarse-grid corrections:
 $\bar{\mathbf{z}}_1^h \leftarrow \bar{\mathbf{z}}_1^h + I_h^h (\bar{\mathbf{z}}_1^H - \bar{\mathbf{z}}_1^H), \quad \bar{\mathbf{z}}_2^h \leftarrow \bar{\mathbf{z}}_2^h + I_h^h (\bar{\mathbf{z}}_2^H - \bar{\mathbf{z}}_2^H)$
 $\bar{\mathbf{z}}_3^h \leftarrow \bar{\mathbf{z}}_3^h + I_h^h (\bar{\mathbf{z}}_3^H - \bar{\mathbf{z}}_3^H), \quad \bar{\mathbf{z}}_4^h \leftarrow \bar{\mathbf{z}}_4^h + I_h^h (\bar{\mathbf{z}}_4^H - \bar{\mathbf{z}}_4^H)$
 - Post-smoothing:
For $k = 1$ to ν_2 , $[\bar{\mathbf{z}}^h] \leftarrow \text{Smoother}(\bar{\mathbf{z}}^h, g_1^h, g_2^h, g_3^h, g_4^h, R^h, T^h, \alpha, \omega, K, \text{PCGSiter})$
-

For practical applications our FAS-NMG method is stopped if the maximum number ε_1 of V- or W-cycles is reached (usually $\varepsilon_1 = 10$), the mean of the relative residuals obtained from the Euler-Lagrange equations (3.13) is smaller than a small prescribed number $\varepsilon_2 > 0$ (typically $\varepsilon_2 = 10^{-3}$), the relative reduction of the dissimilarity $\tilde{\varepsilon}_3$ is smaller than some $\varepsilon_3 > 0$ (we usually assign $\varepsilon_3 = 0.3$ meaning that the relative reduction of the dissimilarity would decrease about 70%), or the change in two consecutive steps of the data/fitting term \mathcal{D} is smaller than a small number $\varepsilon_4 > 0$ (typically $\varepsilon_4 = 10^{-6}$).

5. Further numerical experiments. In this section we present some experiments to

- (i) compare the modeling results of our new curvature model $\mathcal{R}^{\text{NewCv}}$ with two related approximation models $\mathcal{R}^{\text{FMcurv}}$ and $\mathcal{R}^{\text{HWcurv}}$ as well as $\mathcal{R}^{\beta\text{TV}}$;
- (ii) demonstrate the performance of our new Algorithm 2 for $\mathcal{R}^{\text{NewCv}}$ with regard to parameter changes.

Two representative data sets (a smooth registration problem and a non-smooth registration problem to be denoted respectively as Example 3¹ and Example 4) were selected for the experiments, as shown respectively in Figure 5.1 (a) – (d). Improvements of $\mathcal{R}^{\text{FMcurv}}$ and $\mathcal{R}^{\text{HWcurv}}$ over non-curvature models can also be found from [21, 22, 38, 37, 33, 35, 34]. In all cases, we use the bilinear interpolation to compute the transformed template image $T(\mathbf{u})$ once the displacement field is found. Below we mainly highlight the further gains from using $\mathcal{R}^{\text{NewCv}}$.

Note that our FAS-NMG algorithm also works for the model of $\mathcal{R}^{\text{FMcurv}}$. This can be done by slightly adapting our FAS-NMG method and the main part is to design a suitable smoother. According to the associated Euler-Lagrange system (1.10), the PDFP idea can be used to obtain a PDFP-type smoother by replacing the nonlinear

¹<http://www.math.mu-luebeck.de/safir/>

operator $\tilde{\mathcal{L}}_l$ in the main diagonal of \mathbf{N}^{PDFP} defined below (3.14) by the Laplacian and discarding the last terms of $\hat{g}_3^{[\nu]}, \hat{g}_4^{[\nu]}$. Of course, the PCGS relaxation method is an appropriate choice for solving the linearized system.

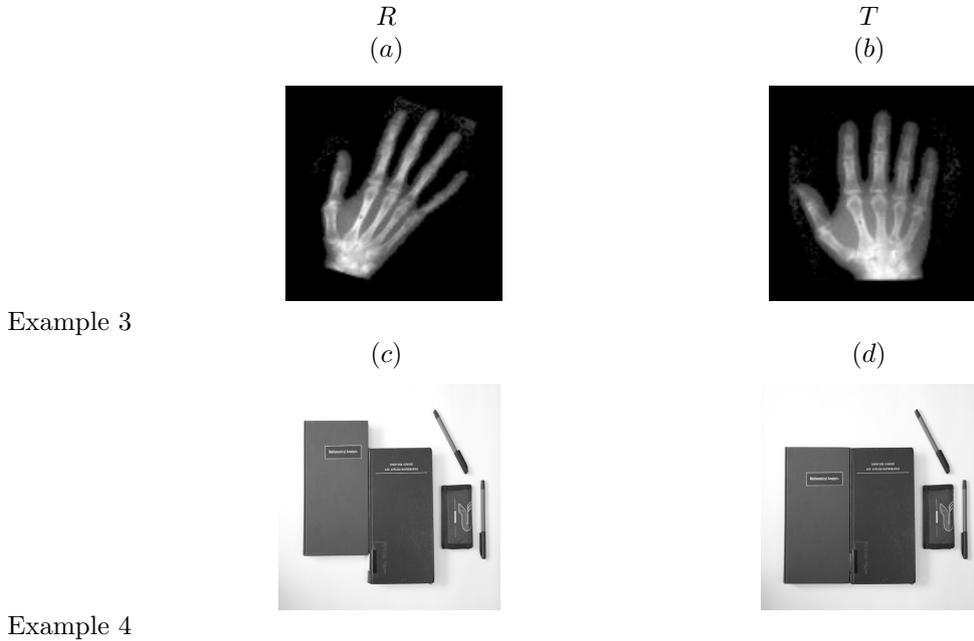


FIG. 5.1. The second set of 2 registration problems. Left to right: reference R and template T . Top to bottom: Example 3 (a smooth registration problem) and Example 4 (a non-smooth registration problem).

5.1. Comparison with other PDE-based image registration models. In the first experiment, our aim is to investigate capabilities of $\mathcal{R}^{\text{FMcurv}}$, $\mathcal{R}^{\text{HWcurv}}$, $\mathcal{R}^{\text{NewCv}}$, and $\mathcal{R}^{\beta\text{TV}}$ for registration of the two test Examples 3 – 4 in resolution 512×512 .

The registered results by the four models are shown in Figures 5.2 (a) – (d) and 5.4 (a) – (d) with the deformation results shown in Figures 5.3 (a) – (d) and 5.5 (a) – (d). For the smooth registration problem (Example 3), one can observe that all three curvature methods work fine in producing an acceptable registration although the registered result by the new model $\mathcal{R}^{\text{NewCv}}$ has the best value of $\tilde{\varepsilon}_3$. Moreover, we can clearly see that it is always the case that $\mathcal{R}^{\beta\text{TV}}$ even with the suitably selected β for estimating homogeneous diffusion processes in (1.15) fails and the regularizations based on higher-order derivatives like $\mathcal{R}^{\text{FMcurv}}$, $\mathcal{R}^{\text{HWcurv}}$, and $\mathcal{R}^{\text{NewCv}}$ are successful.

However, for the non-smooth registration problem (Example 4), one can clearly see that $\mathcal{R}^{\text{FMcurv}}$ and $\mathcal{R}^{\text{HWcurv}}$ failed to deliver a good registration (note other models from [44] cannot register this hard example either), from Figures 5.4 (a) – (b) and 5.5 (a) – (b). But our new model $\mathcal{R}^{\text{NewCv}}$ evidently produced visually pleasing results as good as those of $\mathcal{R}^{\beta\text{TV}}$. The main reason is that the exact deformation field should have a non-smooth shift for the left book to the top; c.f. Figures 5.5 (c)–(d). Precisely, this field is piecewise constant and substantially discontinuous at regions close to the

interface of the books. Consequently, $\mathcal{R}^{\text{FMcurv}}$ and $\mathcal{R}^{\text{HWcurv}}$ must fail because they smooth the field at those regions; see over smoothing results of the field in Figure 5.5 (a) – (b).

Both examples confirm that our new model $\mathcal{R}^{\text{NewCv}}$ is better and more flexible than $\mathcal{R}^{\beta\text{TV}}$, $\mathcal{R}^{\text{HWcurv}}$ and $\mathcal{R}^{\text{FMcurv}}$ [21, 22, 38, 37, 33, 35, 34] which are in turn better than a class of other registration models.

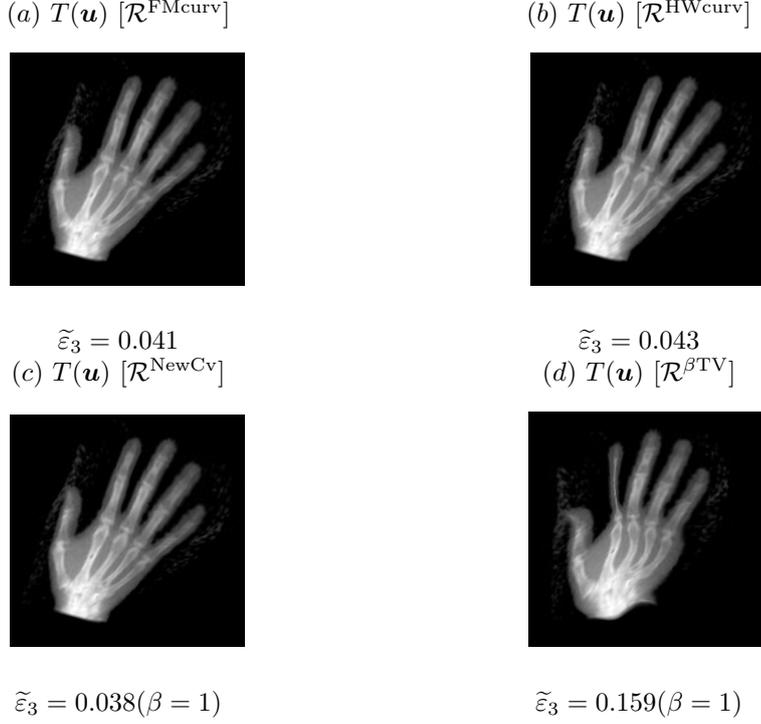


FIG. 5.2. Registered images for Example 3 (the smooth registration problem) shown in Figure 5.1 (a) – (b). Top to bottom: results by (a) $\mathcal{R}^{\text{FMcurv}}$, (b) $\mathcal{R}^{\text{HWcurv}}$, (c) $\mathcal{R}^{\text{NewCv}}$, and (d) $\mathcal{R}^{\beta\text{TV}}$. Clearly, $\mathcal{R}^{\beta\text{TV}}$ failed and $\mathcal{R}^{\text{NewCv}}$ gives the satisfactory registration results as good as those from $\mathcal{R}^{\text{FMcurv}}$ and $\mathcal{R}^{\text{HWcurv}}$, which are known to be suitable for smooth registration problems. Recall that $\tilde{\varepsilon}_3$ means the relative reduction of the dissimilarity defined in Algorithm 2.

Further, in Figures 5.2 (c) and 5.3 (c) we illustrate that our model is able to solve problems involving affine transformations. As we had already mentioned in Section 2, the use of the underlying boundary conditions may present some difficulties since affine transformations are constrained to be constant transformations. However far from boundaries in the interior of the domain this constraining effect is clearly negligible.

5.2. Tests of our new FAS-NMG algorithm. In the previous section we have used the LFA to inform our theoretical choice of suitable smoothers for our new FAS-NMG Algorithm 2. Here by experiments, we hope to first verify the reliability of this choice and then to further test the convergence issues of it with regard to

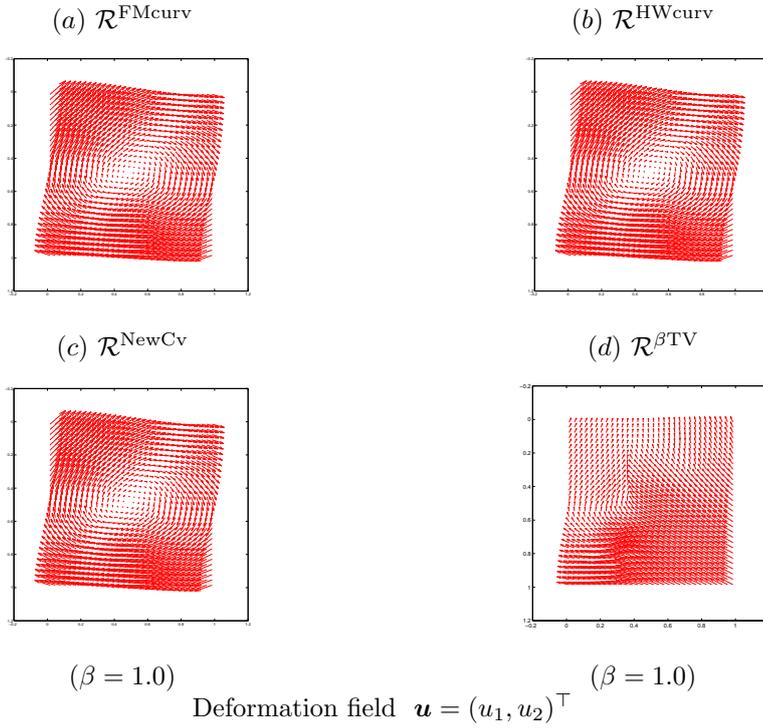


FIG. 5.3. Recovered deformation fields for Example 3 (the smooth registration problem) shown in Figure 5.1 (a) – (b). Top to bottom: results by (a) \mathcal{R}^{FMcurv} , (b) \mathcal{R}^{HWcurv} , (c) \mathcal{R}^{NewCv} , and (d) $\mathcal{R}^{\beta TV}$.

parameters α, β in the model and the mesh parameter h .

1) Comparison of smoothers and h -independent convergence tests. We shall re-solve the same Examples 3 – 4 as above using an increasing sequence of resolutions (or a decreasing mesh parameter h) and show the results in Table 5.1. Algorithm 2 is run using 3 separate smoothers (1 by Method 4 - SFP, 2 by Method 5 - PDFP I and 2* by a modified Method 5 - PDFP II respectively). In each case the algorithm is stopped when the mean of the relative residual below 10^{-6} with ‘M’ the recorded number multigrid cycles required. Then to get an measure of speed without using the machines-dependent CPUs, we work out the work units (WUs) for each case. We also use the relative reduction of dissimilarity $\tilde{\varepsilon}_3$ to indicate the quality of registration obtained at cycle ‘M’.

Here we define a work unit used in measured computational work as the work of performing a smoother or relaxation step on the finest grid defined as follows:

$$1 \text{ WU} = (\text{cost of discretizing and constructing the linearized system per grid point} \\ + \text{cost of PCGS updating per grid point})N \text{ (if } N \text{ is the number of grid points)}$$

For example, a work unit in performing one step of the PDFP I smoother can be estimated by

$$1 \text{ WU} = (150 + 123(PCGSiter))N$$

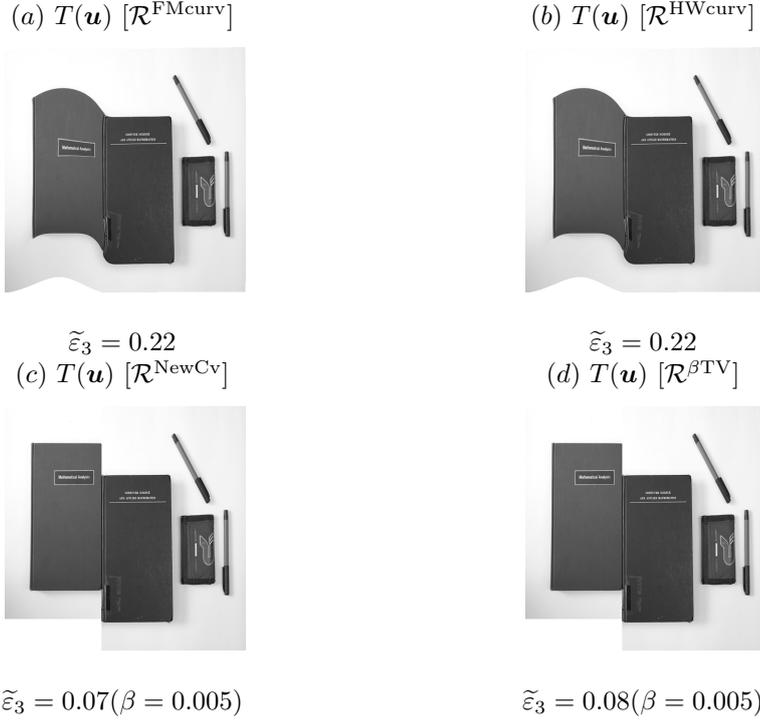


FIG. 5.4. Registered images for Example 4 (the non-smooth registration problem) shown in Figure 5.1 (c) – (d). Top to bottom: results by (a) \mathcal{R}^{FMcurv} , (b) \mathcal{R}^{HWcurv} , (c) \mathcal{R}^{NewCv} , and (d) $\mathcal{R}^{\beta TV}$. Clearly, \mathcal{R}^{FMcurv} and \mathcal{R}^{HWcurv} failed and \mathcal{R}^{NewCv} gives the satisfactory registration results as good as those from $\mathcal{R}^{\beta TV}$, which is known to be suitable for non-smooth registration problems.

where the number 150 is estimated from computing all nonlinear coefficients and the number 123 comes from each grid point in the linearized system (4×4) given in (3.15) when solved by the Gaussian elimination method, which have the cost of $\frac{(4)^3}{3} + \frac{(4)^2}{2} - \frac{5(4)}{6}$ additions and $\frac{(4)^3}{3} + (4)^2 - \frac{(4)}{3}$ multiplications. Therefore, the total costs of one V-cycle used L coarse grids can be estimated as follows:

$$\text{V-cycle cost} = (\nu_1 + \nu_2)(150 + 123(PCGSiter))N \sum_{k=0}^L (1/4)^k < \frac{4}{3}(\nu_1 + \nu_2) \text{ WUs.}$$

Here we have ignored the cost of interpolation and restriction procedures as well as the cost of residual correction procedure because they are relative small compared with that of smoothing procedures. Recall that ν_1 , ν_2 , and $PCGSiter$ denote respectively the number of pre- and post-smoothing and PCGS steps.

In the numerical results shown in Table 5.1, one can see six quantities: the numbers of pre- and post-smoothing and PCGS steps ν_1, ν_2 ; the multigrid cycles ‘M’; the relative reduction of dissimilarity $D = \tilde{\varepsilon}_3$ and WUs.

As expected from the LFA results in the last section, our numerical results confirm that Smoothers 2, 2* (as PDFP I and II) are much better than Smoother 1 (SFP) for

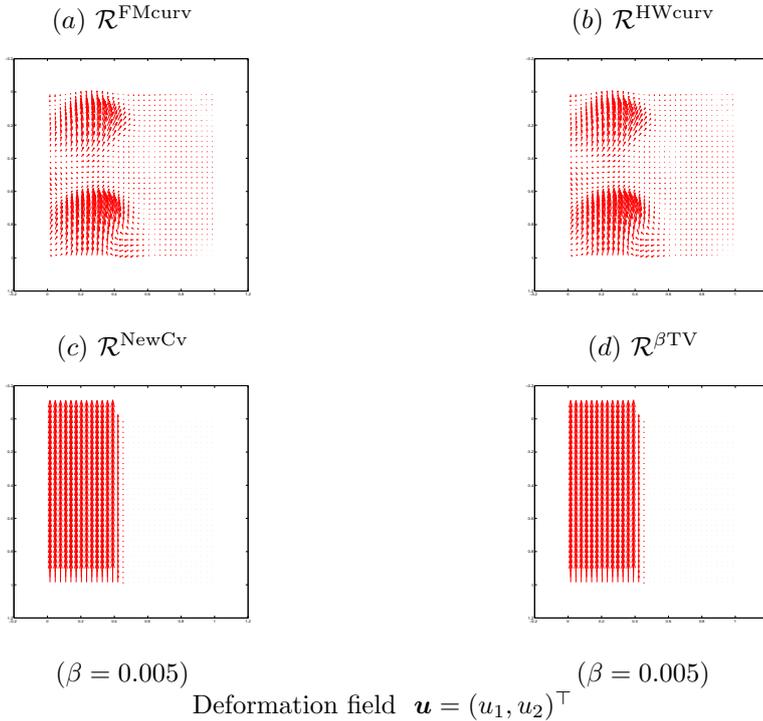


FIG. 5.5. Recovered deformation fields for Example 4 (the non-smooth registration problem) shown in Figure 5.1 (c) – (d). Top to bottom: results by (a) \mathcal{R}^{FMcurv} , (b) \mathcal{R}^{HWcurv} , (c) \mathcal{R}^{NewCv} , and (d) $\mathcal{R}^{\beta TV}$.

our FAS-NMG algorithm, because they not only lead to the convergence within a few MG cycles as expected of a multigrid technique, but also to the accurate results. The dissimilarities between the reference and registered images have been reduced more than 90% for both examples.

Overall, as LFA predicts, the above experimental results suggest that Smoother 2* (PDFP II) would be preferred for practical applications. In other tests, we note the Smoother 1 and other methods described in Section 3 can lead to the MG convergence for both registration problems when the number of pre- and post smoothing steps ν_1 and ν_2 are further increased.

2) α -dependence tests. Next we assess how our MG algorithm is affected when varying α . To this end, the MG algorithm based on Smoother 2* was tested on Example 3 (see Figure 5.1 (a) – (b)) with the results shown in Table 5.2. Here the following parameters are used: $\beta = 1$, $\nu_1 = \nu_2 = PCGSiter = 10$, and $h = 1/256$ for all experiments and α is varied from $1/10000$ to $1/10$. For this example, large α is not needed as small ones give better results. However, the selection of suitable α is a separate but important issue because it is in general unknown a priori and it significantly affects on the qualities of registered images as well as the MG performance. In order to estimate a reasonable α automatically, we may adapt our MG algorithm and follow the ‘cooling’ process suggested in [17, 29, 28] which resembles

$h = 1/n$ for image of $n \times n$	MG with Smoother 1 (SFP) $\nu_1/\nu_2/PCGSiter/M/D/WUs$	MG with Smoother 2 (PDFP I) $\nu_1/\nu_2/PCGSiter/M/D/ M/D/WUs$	MG with Smoother 2* (PDFP II) $\nu_1/\nu_2/PCGSiter/M/D/WUs$
Ex. 3	$\alpha = 10^{-4}, \gamma = \beta^{-\frac{1}{2}}$		
$n = 128$	10/10/10/18/0.03/480	10/10/10/6/0.0264/160	10/10/10/5/0.0258/133
$n = 256$	10/10/10/*/*/*	10/10/10/7/0.0388/187	10/10/10/6/0.0386/160
$n = 512$	10/10/10/*/*/*	10/10/10/7/0.0379/187	10/10/10/6/0.0379/160
$n = 1024$	10/10/10/*/*/*	10/10/10/8/0.0412/213	10/10/10/7/0.0398/187
Ex. 4	$\alpha = 0.75/10000$		
$n = 128$	10/10/15/*/*/*	10/10/15/11/0.0713/293	10/10/15/8/0.0698/213
$n = 256$	10/10/15/*/*/*	10/10/15/12/0.0739/320	10/10/15/9/0.0701/240
$n = 512$	10/10/15/*/*/*	10/10/15/12/0.0761/320	10/10/15/10/0.0712/267
$n = 1024$	10/10/15/*/*/*	10/10/15/13/0.0793/347	10/10/15/10/0.0753/267

TABLE 5.1

Registration results of Algorithms 2 with the proposed smoothers for processing Examples 3 – 4 shown respectively in Figure 5.1 (a) – (d). The letters ‘M’, ‘D’, and ‘WUs’ mean the number of multigrid cycles, the relative reduction of dissimilarity ($\tilde{\varepsilon}_3$), the work units, respectively. ‘*’ indicates failure in dropping the mean of the relative residual to 10^{-6} within 20 MG-cycles. Recall that γ is the SFP parameter.

the L-curve method in other inverse problems. Nevertheless, for the range of α tested in Table 5.2, our FAS-NMG still obtains the solution in a reasonable number of iterations.

α	β	M	D
10^{-4}	1	6	0.0379
10^{-3}	1	7	0.1528
10^{-2}	1	7	0.3019
10^{-1}	1	15	0.4709

TABLE 5.2

Results for α –dependence tests of Algorithms 2 with the PDFP II smoother for Example 3 shown in Figure 5.1 (a) – (b). The letters ‘M’ and ‘D’ mean the number of multigrid steps and the relative reduction of dissimilarity ($\tilde{\varepsilon}_3$).

3) β –dependence tests. As is well known, the quantities of results and the performances of the MG techniques in solving the nonlinear system related to the TV regularization technique are affected significantly by the values of β . As already discussed in Section 2, for registration purposes $\beta = 1$ is suitable for smooth registration problems because the diffusion coefficients (\tilde{D}_l) are almost isotropic in all regions and then it leads to the smooth deformation fields. On the other hand $\beta \ll 1$ is appropriate for non-smooth registration problems because the diffusion coefficients are zero in regions representing large gradients of the fields and then it allows discontinuities at those regions. Here our aim is to see how our MG algorithm is affected when varying the values of β .

To this end, the MG algorithm based on Smoother 2* was tested on the non-smooth Example 4 as from Figure 5.1 (c) – (d). Here the following parameters are taken: $\alpha = 0.75/10000$, $\nu_1 = \nu_2 = 10$, $PCGSiter = 15$, and $h = 1/256$ for all experiments and β is varied from 0.005 to 1. Table 5.3 shows that our MG algorithm converges in a few steps. Theoretically β should be selected to be as small as possible.

However, in practice, small β is not necessary and not recommendable. As shown in our experiments, $\beta = 1 \times 10^{-2}$ or 5×10^{-3} is enough to solve the non-smooth registration problem with the accurate results in a few MG steps.

α	β	M	D
0.75×10^{-4}	5×10^{-3}	9	0.0701
0.75×10^{-4}	1×10^{-2}	8	0.0893
0.75×10^{-4}	1×10^{-1}	7	0.2324
0.75×10^{-4}	1×10^{-0}	6	0.4557

TABLE 5.3

Results for β -dependence tests of Algorithm 2 with Smoother 2* for Example 4 shown respectively in Figure 5.1 (c) – (d). The letters ‘M’ and ‘D’ mean the number of multigrid steps and the relative reduction of dissimilarity ($\tilde{\varepsilon}_3$).

6. Conclusions. The majority of deformable registration models in the variational framework use the gradient information (first order derivatives) in their regularizers. For problems requiring less smooth deformation fields, such models become ineffective and the curvature like information (second order derivatives) used in regularization can improve the registration results, as shown in the recent works of [21, 22, 38, 37, 33, 35, 34] where high-order and essentially *linear PDEs* are solved.

Different from approximate curvature models of [21, 22, 38, 37, 33, 35, 34], the full curvature model considered in this paper does not make assumptions on the deformation fields. Consequently our results improve over previous approximate curvature models for both smooth and non-smooth registration problems in quality and robustness of image registration. Moreover the underlying high order and *nonlinear PDEs* stemmed from our full curvature model appears intractable for effective numerical solutions as commonly used iterative solvers do not converge. We introduced nontrivial fixed-point iterative methods and used them as smoothers for a nonlinear multigrid method. The fast convergence of the multigrid algorithm is assured by a local Fourier analysis. Numerical experiments demonstrated firstly the advantages of our new model and then the fast convergence of our recommended multigrid algorithm.

REFERENCES

- [1] G. AUBERT, R. DERICHE, AND P. KORNPBOST, *Computing optical flow via variational techniques*, SIAM J. Appl. Math., 60 (1999), pp. 156–182.
- [2] G. AUBERT AND P. KORNPBOST, *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations (2nd Edition)*, Springer, 2006.
- [3] N. BADSHAH AND K. CHEN, *Multigrid method for the Chan-Vese model in variational segmentation*, Communications in Computational Physics, 4 (2008), pp. 294–316.
- [4] ———, *On two multigrid algorithms for modelling variational multiphase image segmentation*, IEEE Transactions on Image Processing, 18 (2009), pp. 1097–1106.
- [5] P. BLOMGREN AND T.F. CHAN, *Color TV: total variation methods for restoration of vector-valued images*, IEEE Transactions on Image Processing, 7 (1998), pp. 304–309.
- [6] A. BRANDT, *Multi-level adaptive solutions to BVPs*, Math. Comp., 31 (1977), pp. 333–390.
- [7] X. BRESSON AND T.F. CHAN, *Fast dual minimization of the vectorial total variation norm and applications to color image processing*, Inverse Problems and Imaging, 2 (2008), pp. 455–484.
- [8] W.L. BRIGGS, V.E. HENSON, AND S.F. MCCORMICK, *A Multigrid Tutorial (2nd Edition)*, SIAM Publications, Philadelphia, USA, 2000.
- [9] C. BRITO-LOEZA AND K. CHEN, *Fast numerical algorithms for Euler’s Elastica digital inpainting model*, International Journal of Modern Mathematics, 5 (2010), pp. 157–182.

- [10] ———, *Multigrid algorithm for high order denoising*, SIAM Journal on Imaging Sciences, 3 (2010), pp. 363–389.
- [11] ———, *On high-order denoising models and fast algorithms for vector-valued images*, IEEE Transactions on Image Processing, 19 (2010), pp. 1518–1527.
- [12] T.F. CHAN AND K. CHEN, *On a nonlinear multigrid algorithm with primal relaxation for the image total variation minimization*, Journal of Numerical Algorithms, 41 (2006), pp. 387–411.
- [13] ———, *An optimization-based multilevel algorithm for total variation image denoising*, Multiscale Mod. Simu., 5 (2006), pp. 615–645.
- [14] T.F. CHAN, G.H. GOLUB, AND P. MULET, *A nonlinear primal-dual method for total variation-based image restoration*, SIAM J. Sci. Comput., 20 (1999), pp. 1964–1997.
- [15] T.F. CHAN AND J.H. SHEN, *Image Processing and Analysis - Variational, PDE, Wavelet, and Stochastic Methods*, SIAM Publications, Philadelphia, USA, 2005.
- [16] K. CHEN, *Matrix Preconditioning Techniques and Applications*, Cambridge University Press, UK, 2005.
- [17] N. CHUMCHOB AND K. CHEN, *A robust multigrid approach for variational image registration models*, Submitted, (2010).
- [18] D.J. EYRE, *Unconditionally gradient stable time marching the Cahn-Hilliard equation*, in Computational and Mathematical models of Microstructural Evolution (1998), edited by J.W. Bullard et al. Material Research Society, Warrendale, PA, 39-46, 1998.
- [19] ———, *An unconditionally stable one-step scheme for gradient systems*, Unpublished, (1998).
- [20] B. FISCHER AND J. MODERSITZKI, *Fast diffusion registration*, Contemporary Mathematics, 313 (2002), pp. 117–129.
- [21] ———, *Curvature based image registration*, Journal of Mathematical Imaging and Vision, 18 (2003), pp. 81–85.
- [22] ———, *A unified approach to fast image registration and a new curvature based registration technique*, Linear Algebra and Its Applications, 380 (2004), pp. 107–124.
- [23] ———, *Ill-posed medicine—an introduction to image registration*, Inverse Problems, 24 (2008), pp. (1–19).
- [24] C. FROHN-SCHAUF, S. HENN, L. HÖMKE, AND K. WITSCH, *Total variation based image registration*, in Proceedings of the International Conference on PDE-Based Image Processing and Related Inverse Problems Series: Mathematics and Visualization, edited by X.-C. Tai, K.-A. Lie, T.F. Chan and S. Osher, Springer Verlag, 305-323, 2006.
- [25] C. FROHN-SCHAUF, S. HENN, AND K. WITSCH, *Multigrid based total variation image registration*, Comput. Visual. Sci., 11 (2008), pp. 101–113.
- [26] S. GAO, L. ZHANG, H. WANG, R. DE CREVOISIER, D.D. KUBAN, R. MOHAN, AND L. DONG, *A deformable image registration method to handle distended rectums in prostate cancer radiotherapy*, Med. Phys., 33 (2006), pp. 3304–3312.
- [27] E. HABER, R. HORESH, AND J. MODERSITZKI, *Numerical optimization for constrained image registration*, To appear in Numerical Linear Algebra with Applications, DOI: 10.1002/nla.715 (2010).
- [28] E. HABER AND J. MODERSITZKI, *A multilevel method for image registration*, SIAM J. Sci. Comput., 27 (2006), pp. 1594–1607.
- [29] E. HABER AND D.W. OLDENBURG, *A GCV based method for nonlinear ill-posed problems*, Computational Geosciences, 4 (2000), pp. 41–63.
- [30] W. HACKBUSCH, *Multi-grid Methods and Applications*, Springer-Verlag, Berlin, Heidelberg, New York, 1985.
- [31] J.V. HAJNAL, D.L.G. HILL, AND D. HAWKES, *Medical Image Registration*, The Biomedical Engineering Series, CRC Press, 2001.
- [32] S. HAMILTON, M. BENZI, AND E. HABER, *New multigrid smoothers for the oseen problem*, To appear in Numerical Linear Algebra with Applications, (2009).
- [33] S. HENN, *A multigrid method for a fourth-order diffusion equation with application to image processing*, SIAM J. Sci. Comput., 27 (2005), pp. 831–849.
- [34] ———, *A full curvature based algorithm for image registration*, J Math Imaging Vis, 24 (2006), pp. 195–208.
- [35] ———, *A translation and rotation invariant Gauss-Newton like scheme for image registration*, BIT Numerical Mathematics, 46 (2006), pp. 325–344.
- [36] S. HENN AND K. WITSCH, *Iterative multigrid regularization techniques for image matching*, SIAM J. Sci. Comput., 23 (2001), pp. 1077–1093.
- [37] ———, *Image registration based on multiscale energy information*, Multiscale Modeling and Simulation, 4 (2005), pp. 584–609.
- [38] ———, *A variational image registration approach based on curvature scale space*, LNCS, 3459

- (2005), pp. 143–154.
- [39] L. HÖMKE, *A multigrid method for anisotropic PDEs in elastic image registration*, Numer. Linear Algebra Appl., 13 (2006), pp. 215–229.
 - [40] H. KÖSTLER, K. RUHNAU, AND R. WIENANDS, *Multigrid solution of the optical flow system using a combined diffusion- and curvature-based regularizer*, Numer. Linear Algebra Appl., 15 (2008), pp. 201–218.
 - [41] J. LARREY-RUIZ, R. VERDÚ-MONEDERO, AND J. MORALES-SÁNCHEZ, *A fourier domain framework for variational image registration*, Journal of Mathematical Imaging and Vision, 32 (2008), pp. 57–72.
 - [42] W. LU, M. L. CHEN, G. H. OLIVERA, K. J. RUCHALA, AND T. R. MACKIE, *Fast free-form deformable registration via calculus of variations*, Phys. Med. Biol., 49 (2004), pp. 3067–3087.
 - [43] J.B.A. MAINTZ AND M.A. VIERGEVER, *A survey of medical image registration*, Med. Image Anal., 2 (1998), pp. 1–36.
 - [44] J. MODERSITZKI, *Numerical Methods for Image Registration*, Oxford, 2004.
 - [45] W.C. RHEINOLDT, *Methods for Solving Systems of Nonlinear Equations (2nd Edition)*, SIAM Publications, Philadelphia, USA, 1998.
 - [46] L. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Phys.D, 60 (1992), pp. 259–268.
 - [47] Y. SAAD, *Iterative Methods for Sparse Linear Systems (2nd Edition)*, SIAM Publications, Philadelphia, USA, 2003.
 - [48] B. SEYNAEVE, E. ROSSEEL, B. NICOLAÏ, AND S. VANDEWALLE, *Fourier mode analysis of multigrid methods for partial differential equations with random coefficients*, Journal of Computational Physics, 224 (2007), pp. 132–149.
 - [49] M. STÜRMER, H. KÖSTLER, AND U. RÜDE, *A fast full multigrid solver for applications in image processing*, Numer. Linear Algebra with Appl., 15 (2008), pp. 187–200.
 - [50] U. TROTTEMBERG, C. OOSTERLEE, AND A. SCHULLER, *Multigrid*, Academic Press, 2001.
 - [51] C.R. VOGEL, *Computational Methods for Inverse Problems*, SIAM Publications, Philadelphia, USA, 2002.
 - [52] P. WESSELING, *Multigrid Methods*, Edwards: Philadelphus, PA, U.S.A., 2004.
 - [53] R. WIENANDS AND W. JOPPICH, *Practical Fourier Analysis for Multigrid Method*, Chapman & Hall/CRC, U.S.A., 2005.
 - [54] C. ZACH, T. POCK, AND H. BISCHOF, *A Duality Based Approach for Realtime TV- L^1 Optical Flow*, in Proceedings of the 29th DAGM conference on Pattern recognition, edited by F.A. Hamprecht, C. Schnörr, and B. Jähne, Springer Verlag, 214–223, 2007.
 - [55] D. ZIKIC, W. WEIN, AND A. KHAMENE, *Fast deformable registration of 3D-ultrasound data using a variational approach*, LNCS, 4190 (2006), pp. 915–923.