http://www.sciencedirect.com/science/article/pii/S0307904X16304334
Applied Mathematical Modelling (c) 2016 --- Vol 40 (23/24), Pages 10740 – 10759

An Improved Discontinuity-Preserving Image Registration Model and Its Fast Algorithm*

Jin Zhang† Ke Chen‡ Bo Yu§

Abstract

Recently, Chumchob-Chen(2010) proposed a discontinuity-preserving image registration model which was more flexible than those common techniques such as the diffusion and total variation based regularization techniques. However, each component of displacement field is regularized separately in this model and then the nonlinear diffusion processes resulting from the first variation of the discontinuity-preserving regularization do not enforce coupling between the primary components of the displacement field. Thus the discontinuity-preserving model may prevent to obtain a good registration in some situations, for example non-smooth registration problems with non-axis-aligned discontinuities. To utilize interdependence between the primary components of the deformation field for smooth and non-smooth registration problems, we propose an improved discontinuity-preserving image registration model in this paper, second we propose an idea of relaxed fixed point combining with Gauss-Newton scheme with Armijo's line search for solving the new model and further to combine with a multilevel method to achieve fast convergence. Numerical experiments not only confirm that our proposed method is efficient and stable, but also it can give more satisfying registration results according to image quality.

Keywords. Deformable image registration, Regularization, Multilevel.

AMS Subject Classifications. 65F10, 65M55, 68U10

1 Introduction

Image registration which is also called image matching or image warping is one of the most useful and fundamental tasks in imaging processing domain. Its main idea is to find a reasonable spatial geometric transformation between given two images of the same object taken at different times or from different devices or perspectives, such that a transformed version of the first image is similar to the second one as much as possible. It is often encountered in many fields such as astronomy, art, biology, chemistry, medical imaging and remote sensing and so on. For a good overview about these applications, see e.g. [9, 30, 11, 29, 3, 18, 4].

Usually, a variational image registration model can be described by following form: given two images, one kept unchanged is called reference R and another kept transformed is called template image T. They can be viewed as compactly supported function, $R, T : \Omega \to V \subset \mathbb{R}_0^+$, where $\Omega \subset \mathbb{R}^d$ be a bounded convex domain and d denotes spatial dimension of the given images. The purpose of registration is to look for a transformation φ defined by

$$\varphi : \mathbb{R}^d \to \mathbb{R}^d,$$

*The work was supported by the UK EPSRC grant (number EP/K036939/1) and the National Natural Science Foundation of China (numbers 11171051, 91230103).

†School of Mathematical Sciences, Dalian University of Technology, Liaoning 116024,P R China (zhangjindlut@163.com).

‡Centre for Mathematical Imaging Techniques and Department of Mathematical Sciences, University of Liverpool, United Kingdom (K.Chen@liv.ac.uk).

§School of Mathematical Sciences, Dalian University of Technology, Liaoning 116024, P R China (yubo@dlut.edu.cn).

such that transformed template image $T_\varphi(\boldsymbol{x}) := T(\varphi(\boldsymbol{x}))$ is similar to R as much as possible. To be more intuitive to understand how a point in the transformed template $T(\varphi(\boldsymbol{x}))$ is moved away from its original position in T, we can split the transformation φ into two parts: the trivial identity part and displacement $\boldsymbol{u}$, $\boldsymbol{u} : \mathbb{R}^d \to \mathbb{R}^d$, $\quad \boldsymbol{u} : \boldsymbol{x} \to \boldsymbol{u}(\boldsymbol{x}) = (u_1(\boldsymbol{x}), u_2(\boldsymbol{x}), \cdots, u_d(\boldsymbol{x}))^\top$, that is to say

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

thus it is equivalent to find the transformation φ and the displacement $\boldsymbol{u}$. The transformed template image $T(\varphi(\boldsymbol{x})) = T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}))$ can be denoted $T(\boldsymbol{u})$. In summary, the desired displacement $\boldsymbol{u}$ is a minimizer of the following joint energy functional

$$\min_{\boldsymbol{u}} \{\mathcal{J}_\alpha[\boldsymbol{u}] = \mathcal{D}(\boldsymbol{u}) + \alpha\mathcal{R}(\boldsymbol{u})\}, \tag{1}$$

where

$$\mathcal{D}(\boldsymbol{u}) = \frac{1}{2}\int_\Omega (T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})) - R(\boldsymbol{x}))^2 d\boldsymbol{x} \tag{2}$$

represents similarity measure which quantifies distance or similarity of transformed template image $T(\boldsymbol{u})$ and reference R, $\mathcal{R}(\boldsymbol{u})$ is regularizer which rules out unreasonable solutions during registration process, and $\alpha > 0$ is a regularization parameter which balance similarity and regularity of displacement.

And non-surprisingly, different regularizer techniques can produce different registration model, and the choice of regularizer techniques is very crucial for the solution and its properties, more details see [30]. At present, the common regularizer techniques such as diffusion-, elastic-, or linear curvature-based image registration can generate globally smooth displacement, more details see [13, 16, 17, 27, 26, 25, 30, 36] and reference therein. However, these techniques become poor when displacement $\boldsymbol{u}$ is discontinuous. Total variation-based image registration is better for preserving discontinuities of the displacement, see [19, 20, 33]. Nevertheless, the TV model may not give satisfactory registration results for smooth displacement. Recently, Chumchob-Chen [14] proposed a discontinuity-preserving image registration model which can be interpreted as a half way model between diffusion and total variation registration. The discontinuity-preserving regularization technique which based on a modified total variation (MTV) regularization as given by the following form

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

Here $\phi(s) = \log(1 + s^2)$, and its diffusion coefficient $D(s) = \phi'(s)/s = 2/(1 + s^2)$. It is worth noticing that the diffusion coefficient (or the stopping function) $D(s)$ has the following basic properties: (1) $D(s) \to 0$, as $s \to \infty$. (2) $D(s) \to M$ $(0 < M < +\infty)$ as $s \to 0$. These mean that on one hand it preserves discontinuities of $\boldsymbol{u}$ by reducing or stopping the diffusion (smoothing) process in inhomogeneous regions, on the other hand it smooths $\boldsymbol{u}$ isotropically inside homogeneous regions. In other words, TV-like regularization is used in inhomogeneous regions and diffusion- or quadratic-like regularization is used in homogeneous regions. Moreover, it is more flexible than those common techniques such as the diffusion and total variation based regularization techniques. However, each displacement variable u_1 and u_2 is regularized separately in (3) and then the nonlinear diffusion processes resulting from the first variation of $\mathcal{R}^{\mathrm{MTV}}(\boldsymbol{u})$ do not enforce coupling between the primary components of the deformation field u_1 and u_2. Thus the $\mathcal{R}^{\mathrm{MTV}}(\boldsymbol{u})$ model may prevent to obtain a good registration result in some situations, for example non-smooth registration problems with non-axis-aligned discontinuities. Motivated by several regularization techniques that have been prove to be

very useful in vector-valued image denoising [5, 6, 8] and in optical flow computation [1, 2], we propose an improved discontinuity-preserving model for image registration in this paper. Moreover, other motivations can also be produced as follows: 1) these vectorial regularization techniques can preserve discontinuities of the displacement field for non-smooth registration problems, 2) they can enforce coupling between the primary components of the displacement field to improve the registration quality for both smooth and non-smooth registration problems.

It is usually difficult to solve analytically the optimization problem (1), thus numerical schemes and appropriate discretizations are nececcary. Developing an efficient numerical solution of the registration problem is an important task. Over the past decades, there two main types of numerical schemes to compute a numerical solution of minimization problem (1) for a given α. The first is optimize-discretize scheme, and its main idea is to let the first order variation of (1) vanish and obtain corresponding Euler-Lagrange equations in the continuous domain and then solve its discrete forms on the corresponding discrete domain by appropriate methods, see [15, 30, 13, 16, 36, 17, 27, 33]. The second is the discretize-optimize approach which aims to discretize the joint functional $\mathcal{J}_\alpha$ in (1) and then solve the discrete minimization problem by standard optimization methods; see, e.g. [26, 25, 24, 23, 22]. In this paper, we prefer the second method. Although our work is related to previous work [26], they are totally different on their regularizer techniques and equations. If we use directly the scheme proposed in [26], it is very difficult to solve efficiently for (9). However, motivated by the idea of [26], we can linearize the first order variation of regularizer (9) in order to take advantage of efficient optimization schemes by using a previous and known iterate value, then solve the discrete energy functional using optimization methods.

The rest of the paper is organized as follows. In Section 2, we first present a new discontinuity-preserving regularizer suitable for both smooth and non-smooth deformation problems, and then discusses an efficient numerical method to solve (9) in Section 3. Section 4 illustrates the experimental results from syntectic and real images. Finally, conclusions and future work are summarized in Section 5.

2 A new discontinuity-preserving regularizer based image registration model

Our ideal regulariser is expected to design a model suitable for both smooth problems and non-smooth problems. In this paper, we extend the modified total variation (TV) regularization $\mathcal{R}^{\mathrm{MTV}}(\boldsymbol{u})$ proposed by Chumchob-Chen [14] to the vectorial case using vectorization method for color image denoising proposed in [5, 6, 8]. At present, there are mainly three kinds of vectorization ideas in color image processing. Next, we briefly review the vectorial regularization techniques.

- Vectorial TV Model of Blomgren and Chan(TV1-BLC). The Blomgren and Chan model [5] is based on the Euclidean norm of the vector of component-wise scalar TV. Define $\mathrm{TV}(u_l) = \int_\Omega |\nabla u_l| d\boldsymbol{x}$ for channel l. Then the multidimensional TV norm was denoted by

$$\mathcal{R}^{\mathrm{VTV_1}}(\boldsymbol{u}) = \mathrm{TV}_m(\boldsymbol{u}) = \sqrt{\sum_{l=1}^{m} [\mathrm{TV}(u_l)]^2}$$

which leads to the corresponding Euler-Lagrange equations

$$-\alpha \frac{\mathrm{TV}(u_l)}{\mathrm{TV}_m(\boldsymbol{u})} \nabla \cdot \frac{\nabla u_l}{|\nabla u_l|} + u_l - u_l^0 = 0 \tag{4}$$

subject to $\langle \nabla u_l, \vec{\boldsymbol{v}}_l \rangle = 0$ on $\partial\Omega$, where $\vec{\boldsymbol{v}}_l$ is the normal unit vector on the boundary of the lth channel.

- Total Variation Model of Bresson and Chan. In [6], Bresson and Chan denote the vectorial regularizer by the finite positive measure

$$\mathcal{R}^{\mathrm{VTV}}(\boldsymbol{u}) = \int_\Omega |D\boldsymbol{u}| = \sup_{\boldsymbol{p}\in P} \{ \int_\Omega \langle \boldsymbol{u}, \nabla\cdot\boldsymbol{p}\rangle d\boldsymbol{x}\} \tag{5}$$

where $\boldsymbol{p} := (\boldsymbol{p}_1, \cdots, \boldsymbol{p}_m) : \Omega \to \mathbb{R}^{m\times n}$, $\boldsymbol{p}_i := (p_i{}^{x_1}, \cdots, p_i{}^{x_n}) : \Omega \to \mathbb{R}^n$, $\forall i \in [1,m]$, $\nabla\cdot$ is the divergence operator such that $\nabla\cdot\boldsymbol{q} := (\nabla\cdot\boldsymbol{q}_1, \cdots, \nabla\cdot\boldsymbol{q}_m) : \Omega \to \mathbb{R}^m$, $\forall \boldsymbol{q} : \Omega \to \mathbb{R}^{m\times n}$, $\nabla\cdot\boldsymbol{q}_i := \sum_{j=1}^{n} \partial_{x_j} q_i^{x_j} : \Omega \to \mathbb{R}$, $\forall i \in [1,m]$, the product $\langle \cdot\,, \cdot \rangle$ is the Euclidean scalar product defined as $\langle \boldsymbol{v}, \boldsymbol{w}\rangle := \sum_{i=1}^{m} \langle v_i, w_i\rangle$, $\forall (\boldsymbol{v}, \boldsymbol{w}) \in (\mathbb{R}^m)^2$, which implies that $\langle \boldsymbol{u}, \nabla\cdot\boldsymbol{p}\rangle = \sum_{i=1}^{m} \langle u_i, \nabla\cdot\boldsymbol{p}_i\rangle$ and the L^2 Euclidean norm $|\cdot|$ is naturally defined by $|\boldsymbol{v}| := \sqrt{\langle \boldsymbol{v}, \boldsymbol{v}\rangle} = \sqrt{\sum_{i=1}^{l} v_i^2}$, $\forall \boldsymbol{v} \in \mathbb{R}^l$. Depending on the set P of functions of the dual variable $\boldsymbol{p}$, the VTV norm (5) can be defined of different ways, Bresson and Chan [6] considered two cases:

$$P_1 := \{ \boldsymbol{p} \in C_c^1(\Omega\,; \mathbb{R}^{m\times n}) : \ |\boldsymbol{p}|_\infty = \max_{i=1,\cdots,m} |\boldsymbol{p}_i| \le 1 \}\ ;$$

$$P_2 := \{ \boldsymbol{p} \in C_c^1(\Omega\,; \mathbb{R}^{m\times n}) : |\boldsymbol{p}| = \sqrt{\sum_{i=1}^{m} \langle \boldsymbol{p}_i, \boldsymbol{p}_i\rangle} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} (p_i{}^{x_j})^2} \le 1 \}\ .$$

In [6] , it is shown that for smooth function $\boldsymbol{u}$ if $\boldsymbol{p} \in P_1$, then

$$\int_\Omega |D\boldsymbol{u}| = \sum_{i=1}^{m} \int_\Omega |\nabla u_i| d\boldsymbol{x} = \sum_{i=1}^{m} \mathrm{TV}(u_i) \triangleq \mathcal{R}^{\mathrm{VTV}_2}(\boldsymbol{u})\ . \tag{6}$$

Likewise selecting $\boldsymbol{p} \in P_2$ [6] yields

$$\int_\Omega |D\boldsymbol{u}| = \int_\Omega \sqrt{\sum_{i=1}^{m} |\nabla u_i|^2} d\boldsymbol{x} = \int_\Omega |\nabla \boldsymbol{u}| d\boldsymbol{x} \triangleq \mathcal{R}^{\mathrm{VTV}_3}(\boldsymbol{u})\ . \tag{7}$$

- Combining the above mentioned idea, Brito-Chen [8] proposed three vetorial high-order denoising models using channel coupling and high-order regularization. In addition, the vectorial regularization techniques are also widely used in the optical flow context proposed by [1, 2].

Note that among the above mentioned three vectorial regularization techniques, the vectorial $\mathcal{R}^{\mathrm{VTV}_2}$ based on P_1 is defined as the sum of the TV of each channel. This means that channels are considered as independent in the denoising process, which is not true on real-world images. In [5], although the proposed vectorial regularizer $\mathcal{R}^{\mathrm{VTV}_1}$ enforces coupling, the proposed vectorial scheme doesn't regularize the VTV to minimize it, details see [6]. The vectorial $\mathcal{R}^{\mathrm{VTV}_3}$ based on P_2 introduces a coupling between channels. Each channel uses information coming from other channels to improve the denoising model. In fact, the vector valued TV norm $\mathcal{R}^{\mathrm{VTV}_3}$ defined by the set P_2 is the most standard definition of the VTV norm as introduced in the book of Ambrosio, Fusco and Pallara [7]. In addition, the minimization algorithm based on $\mathcal{R}^{\mathrm{VTV}_3}$ using coupling information is fast, easy to code and well-posed.

Thus motivated by several regularization techniques that have been proved to be very useful in vector-valued image denoising [5, 6, 8] and in optical flow computation [1, 2], to make full use of the coupling

information, we propose an improved discontinuity-preserving model given by

$$\mathcal{R}^{\text{New}}(\boldsymbol{u}) = \int_{\Omega} \log(1+|\nabla \boldsymbol{u}|^2) d\boldsymbol{x} \tag{8}$$

then equation (1) takes the following form

$$\min_{\boldsymbol{u}} \Big\{ \mathcal{J}_\alpha(\boldsymbol{u}) = \frac{1}{2}\int_{\Omega} \big(T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})) - R(\boldsymbol{x})\big)^2 d\boldsymbol{x} + \alpha \int_{\Omega} \log(1+|\nabla \boldsymbol{u}|^2) d\boldsymbol{x} \Big\} \ . \tag{9}$$

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

$$\begin{cases} f_1(\boldsymbol{u}) - \alpha \nabla \cdot \left(\frac{2\nabla u_1}{1+|\nabla \boldsymbol{u}|^2}\right) = 0 \\ f_2(\boldsymbol{u}) - \alpha \nabla \cdot \left(\frac{2\nabla u_2}{1+|\nabla \boldsymbol{u}|^2}\right) = 0 \end{cases}, \tag{10}$$

subject to $\langle \nabla u_l, \boldsymbol{n} \rangle = 0$ on $\partial\Omega$, $l = 1, 2$. Here

$$f_l(\boldsymbol{u}) = (T(\boldsymbol{u}) - R)\ (\partial_{u_l} T(\boldsymbol{u})), \quad l = 1, 2 \ .$$

Our particular choice of regularizer in (8) has several advantages. Firstly, the new regularization $\mathcal{R}^{\text{New}}$ is rotational invariant. Secondly, observing the system of PDEs (10), the coupling between two primary components u_1 and u_2 of the deformation field $\boldsymbol{u}$ is through the diffusion coefficient $D(\boldsymbol{u}) = \|\nabla \boldsymbol{u}\|^{-1}$. Since $D(\boldsymbol{u})$ takes different values for all image pixels, the level of coupling varies from one region to another locally adjusting the level of regularization. Thirdly, it shares some attractive properties with the modified total variation $\mathcal{R}^{\text{MTV}}$ regularization; Fourthly a visually pleasing registration result can be obtained by using $\mathcal{R}^{\text{New}}$ for non-smooth registration problems with non-axis-aligned discontinuities. Fifthly, it incorporates the coupling information between two primary components of the deformation field to improve the registration quality over the $\mathcal{R}^{\text{MTV}}$ for both smooth and non-smooth registration problems. Finally, a fast numerical algorithm is straightforward to implement .

We remark that the Euler Lagrange equations (10) is nonlinear. To solve the resulting EL equation (10), one convenient way is time marching method (also known as gradient descent method). The main idea is to introduce an artificial time variable t and compute the steady-state solution of the following time-dependent PDEs :

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

where

$$\mathcal{N}_l(\boldsymbol{u}(\boldsymbol{x};t)) = f_l(\boldsymbol{u}(\boldsymbol{x};t)) - \alpha\nabla\cdot\left(\frac{2\nabla u_l(\boldsymbol{x};t)}{1+(|\nabla \boldsymbol{u}(\boldsymbol{x},t)|)^2}\right) \ , \quad l = 1, 2.$$

To overcome the nonlinearity of $\mathcal{N}_l$, an explicit scheme is used and corresponding iteration is give by

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

where $\boldsymbol{u}(\boldsymbol{x}, t_0)$ is some deformation fields, especially $\boldsymbol{u}(\boldsymbol{x}, t_0) = 0$.

A time-step τ>0 is introduced for the time discretization, and $\boldsymbol{u}$ is updated by the following form:

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

We find numerically it is easy to implement for the time marching method, however, it is very slow to converge since the time-step τ is required to be very small for stability reasons. Below we take a different solution approach for model (9).

3 Numerical solution of image registration model (9)

In this section, we shall first briefly discuss the discretization we use, then describe the details of numerical algorithms.

3.1 Finite difference discretization

Let given discrete images have $n_1 \times n_2$ pixels. For the sake of simplicity, we also assume further that image domain $\Omega = [0,1] \times [0,1] \subset \mathbb{R}^2$, then each side of these $n_1 \times n_2$ cell-centered has width $h_i = 1/n_i, i = 1,2$. Let the discrete domain be denoted by

$$\Omega_h = \{\boldsymbol{x} \in \Omega | \boldsymbol{x} = (x_{1_i}, x_{2_j})^\top = ((i-0.5)h_1, (j-0.5)h_2)^\top,\ i = 1,2,\cdots,n_1;\ j = 1,2,\cdots,n_2\}.$$

3.1.1 Discretizing displacement field $\boldsymbol{u}$ and new regularizer $\mathcal{R}^{\mathbf{New}}(\boldsymbol{u})$

Let the discrete form of the continuous displacement field $\boldsymbol{u} = (u_1, u_2)^\top$ be denoted by $\boldsymbol{u}^h = (u_1^h, u_2^h)^\top$, where u_1^h and u_2^h are denoted grid function and are discretized on the discrete domain Ω_h. For simplicity, let $(u_l^h)_{i,j} = u_l^h(x_{1_i}, x_{2_j})$, $i = 1,2,\cdots,n_1;\ j = 1,2,\cdots,n_2$ and $l = 1,2$. Since the first order variation of the new regularizer $\mathcal{R}^{\mathrm{New}}(\boldsymbol{u})$ is represented by the operators gradient ∇ and divergence $\nabla\cdot$, we first define discrete gradient operator ∇^h at each pixel (i,j) by

$$(\nabla^h \boldsymbol{u}^h)_{i,j} = ((\nabla^h u_1^h)_{i,j}, (\nabla^h u_2^h)_{i,j})^\top$$

with

$$(\nabla^h u_l^h)_{i,j} = ((\partial_1^h u_l^h)_{i,j}, (\partial_2^h u_l^h)_{i,j})^\top$$

$$(\partial_1^h u_l^h)_{ij} = \begin{cases} \frac{1}{h_1}((u_l^h)_{i+1,j} - (u_l^h)_{i,j}), & \text{if } i < n_1 \\ 0\quad , & \text{if } i = n_1 \end{cases}$$

$$(\partial_2^h u_l^h)_{ij} = \begin{cases} \frac{1}{h_2}((u_l^h)_{i,j+1} - (u_l^h)_{i,j}), & \text{if } j < n_2 \\ 0\quad , & \text{if } j = n_2. \end{cases}$$

Here homogeneous Neumann boundary conditions on $\boldsymbol{u}$ are assumed:

$$\frac{\partial u_l}{\partial \boldsymbol{\nu}} = 0, \quad l = 1,2 \quad \text{ on } \partial\Omega.$$

We know that the discrete divergence operator is the negative adjoint of the gradient operator by the analysis of the continuous setting, that is to say $\nabla\cdot = -\nabla^*$. Thus, we can define the divergence operator $\nabla\cdot$ by the following form:

$$(\nabla \cdot v_l)_{i,j} = \begin{cases} \frac{1}{h_1}((v_l^1)_{i,j} - (v_l^1)_{i-1,j}) \\ \frac{1}{h_1}((v_l^1)_{i,j}) \\ -\frac{1}{h_1}((v_l^1)_{i-1,j}) \end{cases} + \begin{cases} \frac{1}{h_2}((v_l^2)_{i,j} - (v_l^2)_{i,j-1}) & \text{if } \ 1 < i < n_1\ , 1 < j < n_2 \\ \frac{1}{h_2}((v_l^2)_{i,j}) & \text{if } \ i = j = 1 \\ -\frac{1}{h_2}((v_l^2)_{i,j-1}) & \text{if } \ i = n_1\ ,\ j = n_2. \end{cases}$$

For convenience, we change the grid functions u_1^h and u_2^h into the columns vectors $\boldsymbol{u}_1^h$ and $\boldsymbol{u}_2^h$ according to lexicographical ordering, respectively

$$\boldsymbol{u}_1^h = (u_{1_{1,1}}, u_{1_{2,1}}, \cdots, u_{1_{n_1,1}}, u_{1_{1,2}}, u_{1_{2,2}}, \cdots, u_{1_{n_1,2}}, \cdots, u_{1_{1,n_2}}, u_{1_{2,n_2}}, \cdots, u_{1_{n_1,n_2}})^\top,$$

$$\boldsymbol{u}_2^h = (u_{2_{1,1}}, u_{2_{2,1}}, \cdots, u_{2_{n_1,1}}, u_{2_{1,2}}, u_{1_{2,2}}, \cdots, u_{2_{n_1,2}}, \cdots, u_{2_{1,n_2}}, u_{2_{2,n_2}}, \cdots, u_{2_{n_1,n_2}})^\top,$$

then $\boldsymbol{u}_1^h \in \mathbb{R}^N$, $\boldsymbol{u}_2^h \in \mathbb{R}^N$ and $\boldsymbol{U}^h = (\boldsymbol{u}_1^h; \boldsymbol{u}_2^h) \in \mathbb{R}^{2N}$, where $N = n_1 n_2$. The discrete gradient $(\nabla^h u_l^h)_{i,j}$ can also be represented by the product of the matrix $A_k^\top \in \mathbb{R}^{2\times N}(k = 1, 2, \cdots, N)$ and the vector $\boldsymbol{u}_l^h (l = 1, 2)$:

$$A_k^\top \boldsymbol{u}_l^h = \begin{cases} (\frac{1}{h_1}((\boldsymbol{u}_l^h)_{k+1} - (\boldsymbol{u}_l^h)_k); \frac{1}{h_2}((\boldsymbol{u}_l^h)_{k+n_2} - (\boldsymbol{u}_l^h)_k)), & \text{if } k \bmod n_1 \neq 0 \text{ and } k + n_2 \leq N \\ (0; \frac{1}{h_2}((\boldsymbol{u}_l^h)_{k+n_2} - (\boldsymbol{u}_l^h)_k)), & \text{if } k \bmod n_1 = 0 \text{ and } k + n_2 \leq N \\ (\frac{1}{h_1}((\boldsymbol{u}_l^h)_{k+1} - (\boldsymbol{u}_l^h)_k); 0), & \text{if } k \bmod n_1 \neq 0 \text{ and } k + n_2 > N \\ (0; 0), & \text{if } k \bmod n_1 = 0 \text{ and } k + n_2 > N \end{cases}.$$

Let

$$A = (A_1, A_2, \cdots, A_N) = (A_{1,1}, A_{1,2}, \cdots, A_{N,1}, A_{N,2}) \in \mathbb{R}^{N\times 2N};$$

$$A_x = (A_{1,1}, A_{2,1}, \cdots, A_{N,1}) \in \mathbb{R}^{N\times N},$$

and

$$A_y = (A_{1,2}, A_{2,2}, \cdots, A_{N,2}) \in \mathbb{R}^{N\times N}.$$

In this notation, we can get

$$\nabla^h \boldsymbol{u}_1^h = \begin{bmatrix} A_x{}^\top \\ A_y{}^\top \end{bmatrix} \boldsymbol{u}_1^h \triangleq B\boldsymbol{u}_1^h, \quad \nabla^h \boldsymbol{u}_2^h = \begin{bmatrix} A_x{}^\top \\ A_y{}^\top \end{bmatrix} \boldsymbol{u}_2^h \triangleq B\boldsymbol{u}_2^h .$$

Thus, for discrete gradient operator ∇^h, we have

$$\begin{aligned} \nabla^h \boldsymbol{U}^h &= \begin{bmatrix} \nabla^h & 0 \\ 0 & \nabla^h \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1^h \\ \boldsymbol{u}_2^h \end{bmatrix} \\ &= \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1^h \\ \boldsymbol{u}_2^h \end{bmatrix} \\ &\triangleq \boldsymbol{A}\boldsymbol{U}^h. \end{aligned}$$

Let

$$\mathcal{B}[\boldsymbol{u}] = \log(1 + |\nabla \boldsymbol{u}|^2) \tag{11}$$

Hence, we can get the discretization of (11) as following

$$\mathbb{B}^h[\boldsymbol{U}^h] = \log(1 + |\nabla^h \boldsymbol{U}^h|^2) = \log(1 + (\boldsymbol{U}^h)^\top \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{U}^h)$$

According to (11), we have

$$\mathcal{R}^{\text{New}}(\boldsymbol{u}) = \int_\Omega \mathcal{B}[\boldsymbol{u}] d\boldsymbol{x}.$$

Using above those discrete analogues, and approximating the integral by a midpoint quadrature rule, the new regularizer (11) we proposed is descretized as

$$\mathcal{R}^{h\text{New}}(\boldsymbol{U}^h) = h_d \cdot \log(1 + (\boldsymbol{U}^h)^\top \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{U}^h), \tag{12}$$

where $h_d = h_1 h_2$.

3.1.2 Discretizing template image T and reference image R

For given discrete image, an image interpolation is needed to assign image intensity values for any spatial positions which are not necessarily grid points. Although linear interpolation is a reasonable tool in image registration due to its low computational costs, it isn't differentiable at grid points. In order to make full use of fast and efficient optimization method, a smooth interpolation is required. Thus a cubic B-spline approximation is used in our implementation. Further influence of higher or lower order B-spline interpolation to the quality of registration, see [37]. The continuous smooth approximations for template T and reference R are denoted by $\mathcal{T}$ and $\boldsymbol{\mathcal{R}}$, respectively.

Next we derive discrete analogues for the particular building blocks . Let

$$\boldsymbol{x}_{c,1} = [x_{1_{1,1}}, x_{1_{2,1}}, \cdots, x_{1_{n_1,1}}, x_{1_{1,2}}, x_{1_{2,2}}, \cdots, x_{1_{n_1,2}}, \cdots, x_{1_{1,n_2}}, x_{1_{2,n_2}}, \cdots, x_{1_{n_1,n_2}}]^\top,$$

$$\boldsymbol{x}_{c,2} = [x_{2_{1,1}}, x_{2_{2,1}}, \cdots, x_{2_{n_1,1}}, x_{2_{1,2}}, x_{2_{2,2}}, \cdots, x_{2_{n_1,2}}, \cdots, x_{2_{1,n_2}}, x_{2_{2,n_2}}, \cdots, x_{2_{n_1,n_2}}]^\top,$$

and $\boldsymbol{X}_c^h = [\boldsymbol{x}_{c,1}; \boldsymbol{x}_{c,2}]$.

We can get discrete reference image

$$\vec{R} = \boldsymbol{\mathcal{R}}(\boldsymbol{X}_c^h) \tag{13}$$

and discrete transformed template image

$$\vec{T}(\boldsymbol{U}^h) = \mathcal{T}(\boldsymbol{X}_c^h + \boldsymbol{U}^h) \tag{14}$$

here $\vec{T}(\boldsymbol{U}^h)$ is the discrete analogue of the transformed template image $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}))$ as a function of displacement $\boldsymbol{u}$. The Jacobian of $\vec{T}$ can be denoted by

$$\vec{T}_{\boldsymbol{U}^h} = \frac{\partial \vec{T}}{\partial U^h}(U^h) = \frac{\partial \mathcal{T}}{\partial \boldsymbol{U}_c^h}(\boldsymbol{U}_c^h)$$

where $\boldsymbol{U}_c^h = \boldsymbol{X}_c^h + \boldsymbol{U}^h$, and the Jacobian of $\vec{T}$ is a block matrix with diagonal blocks.

3.1.3 Discretizing distance measure $\mathcal{D}$

In the discrete analogue, the integral is approximated by a midpoint quadrature. According to (13) and (14) our discretization of distance measure $\mathcal{D}$ (2) is straightforward:

$$\mathcal{D}^h(\boldsymbol{U}^h) = \frac{1}{2} h_1 h_2 (\vec{T}(\boldsymbol{U}^h) - \vec{R})^\top \cdot (\vec{T}(\boldsymbol{U}^h) - \vec{R})$$

and the derivative of the discretized functional $\mathcal{D}^h(\boldsymbol{U}^h)$ with respect to $\boldsymbol{U}^h$ can still be computed

$$d\mathcal{D}^h(\boldsymbol{U}^h) = h_1 h_2 (\vec{T}(\boldsymbol{U}^h) - \vec{R})^\top \cdot \vec{T}_{\boldsymbol{U}^h}$$

In addition, the second derivative $d^2\mathcal{D}^h(\boldsymbol{U}^h)$ of the distance measure $\mathcal{D}^h$ can also be calculated straightforwardly,

$$d^2\mathcal{D}^h(\boldsymbol{U}^h) = h_1 h_2 (\vec{T}_{\boldsymbol{U}^h})^\top \cdot \vec{T}_{\boldsymbol{U}^h} + h_1 h_2 \sum_{i=1}^{n_1 n_2} d_i(\boldsymbol{U}^h) \nabla^2 d_i(\boldsymbol{U}^h) \quad ,$$

where $d(\boldsymbol{U}^h) = \vec{T}(\boldsymbol{U}^h) - \vec{R} \in \mathbb{R}^{n_1 n_2}$. On one hand, it is consuming and numerically unstable to compute higher order derivatives in registering two images for practical applications; On the other hand, the difference between $\vec{T}(\boldsymbol{U}^h)$ and $\vec{R}$ will become smaller if template image is well registered. To have an efficient and stable numerical scheme as proposed by several works [30, 28], we approximate $d^2\mathcal{D}^h(\boldsymbol{U}^h)$ by the following form

$$d^2\mathcal{D}^h(\boldsymbol{U}^h) = h_1 h_2 (\vec{T}_{\boldsymbol{U}^h})^\top \cdot \vec{T}_{\boldsymbol{U}^h} \quad . \tag{15}$$

3.2 Solving the discrete optimization problem

The discretized joint energy functional (9) reads as follows:

$$\min_{\boldsymbol{U}^h}\{\mathcal{J}_\alpha(\boldsymbol{U}^h)=\mathcal{D}^h(\boldsymbol{U}^h)+\alpha\cdot\mathcal{R}^{h\mathrm{New}}(\boldsymbol{U}^h)\}.$$

Thus we obtain the following form

$$\min_{\boldsymbol{U}^h}\{\mathcal{J}_\alpha(\boldsymbol{U}^h)=\mathcal{D}^h(\boldsymbol{U}^h)+\alpha\cdot h_d\cdot\log(1+(\boldsymbol{U}^h)^\top\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{U}^h)\}. \tag{16}$$

Obviously, the above functional in an algebraic form is nonlinear. In subsequent solutions, we need to differentiate it twice. To reduce nonlinearity, we shall introduce a lagging into the denominator of the first-order variation of the new regularizer $\mathcal{R}^{h\mathrm{New}}(\boldsymbol{U}^h)$. The lagged quantity in (17) uses a previous and known iterate $\boldsymbol{U}^{h^{(k)}}$. We note that the lagging method by 'frozen coefficients' is well known for variational approaches related to total variation (TV) operator (see e.g. [38, 34, 12, 10]).

To solve above problem (16) numerically, standard optimization technique Gauss-Newton scheme is used. The main idea is to linearize $\mathcal{J}_\alpha$ which is replaced by a quadratic $\hat{\mathcal{J}}_\alpha$ near the previous iterative value $\boldsymbol{U}^{h^{(k)}}$ by the Taylor expansion given by

$$\mathcal{J}_\alpha(\boldsymbol{U}^{h^{(k)}}+\delta_{\boldsymbol{U}^h})\approx\hat{\mathcal{J}}_\alpha(\boldsymbol{U}^{h^{(k)}}+\delta^h_{\boldsymbol{U}})=\mathcal{J}_\alpha(\boldsymbol{U}^{h^{(k)}})+d\mathcal{J}_\alpha(\boldsymbol{U}^{h^{(k)}})\cdot\delta_{\boldsymbol{U}^h}+\frac{1}{2}\delta^\top_{\boldsymbol{U}^h}\boldsymbol{H}\delta_{\boldsymbol{U}^h},$$

where $d\mathcal{J}_\alpha(\boldsymbol{U}^{h^{(k)}})$, $\boldsymbol{H}$ are the Jacobian and the approximation of the Hessian of $\mathcal{J}_\alpha$ at $\boldsymbol{U}^{h^{(k)}}$. For $d^2\mathcal{D}^h(\boldsymbol{U}^{h^{(k)}})$ and $\frac{\boldsymbol{A}^\top\boldsymbol{A}}{1+|\boldsymbol{A}\boldsymbol{U}^{h^{(k)}}|^2}$ are both positive semi-define, we know that $\boldsymbol{H}$ is also positive semi-definite. Hence, $\hat{\mathcal{J}}_\alpha$ is convex , see [32] for an extended description. Next we describe the specific steps.

Given initial value $\boldsymbol{U}^{h^{(k)}}$, we compute Jacobian $d\mathcal{J}_\alpha(\boldsymbol{U}^{h^{(k)}})$ and Hessian $\boldsymbol{H}$ at each outer iteration step by the following form, respectively

$$d\mathcal{J}_\alpha(\boldsymbol{U}^{h^{(k)}})=d\mathcal{D}^h(\boldsymbol{U}^{h^{(k)}})+2\alpha\cdot h_d\cdot\frac{\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{U}^{h^{(k)}}}{1+|\boldsymbol{A}\boldsymbol{U}^{h^{(k)}}|^2} \tag{17}$$

and

$$\boldsymbol{H}=d^2\mathcal{D}^h(\boldsymbol{U}^{h^{(k)}})+2\alpha\cdot h_d\cdot\frac{\boldsymbol{A}^\top\boldsymbol{A}}{1+|\boldsymbol{A}\boldsymbol{U}^{h^{(k)}}|^2}. \tag{18}$$

Then perturbation $\delta_{\boldsymbol{U}^h}$ can be obtained by solving linear equation

$$\boldsymbol{H}\delta_{\boldsymbol{U}^h}=-d\mathcal{J}_\alpha(\boldsymbol{U}^{h^{(k)}}). \tag{19}$$

Usually, $\boldsymbol{H}$ is positive definite, thus the quasi-Newton's equation (19) can be solved using a preconditioned conjugate gradient method, and corresponding stopping rule is $\mathrm{norm}(\boldsymbol{H}\delta_{\boldsymbol{U}^h}+d\mathcal{J}_\alpha(\boldsymbol{U}^{h^{(k)}}))/\mathrm{norm}(-d\mathcal{J}_\alpha(\boldsymbol{U}^{h^{(k)}}))\leq 10^{-6}$. In this paper, a standard Armijo line search scheme is used to guarantee the reduction of the objective function $\mathcal{J}_\alpha(\boldsymbol{U}^h)$, details see [32]. The procedure will be terminated when stopping rules are met. In this section we will use following common stopping rules for above Gauss-Newton scheme; see also [31, 21].

1. $\mathrm{Stop}(1)=\mathrm{abs}(\mathcal{J}_{\mathrm{old}}-\mathcal{J}_{\mathrm{c}})\leq 10^{-3}*(1+\mathrm{abs}(\mathcal{J}_{\mathrm{stop}}))$;
2. $\mathrm{Stop}(2)=\mathrm{norm}(\boldsymbol{u}_{\mathrm{c}}-\boldsymbol{u}_{\mathrm{old}})\leq 10^{-2}*(1+\boldsymbol{u}0)$;
3. $\mathrm{Stop}(3)=\mathrm{norm}(d\mathcal{J}_c)\leq 10^{-2}*(1+\mathrm{abc}(\mathcal{J}_{\mathrm{stop}}))$;

4. Stop(4) = norm($d\mathcal{J}_c$) $\leq$ eps;

5. Stop(5) = (iter $\geq$ maxIter);

If the first three of the above stopping criteria are met or the latter two are met at the same time, the iteration is terminated. Where $\mathcal{J}_{\text{old}}$ and $\mathcal{J}_{\text{c}}$ are previous iterative objective function value and current iterative one, respectively. $\mathcal{J}_{\text{stop}}$ is the value of original objective function at $\boldsymbol{u} = 0$. $\boldsymbol{u}_c$ is current iterative value and $\boldsymbol{u}_{\text{old}}$ is previous iterative one. $\boldsymbol{u}_0$ is initial iterative value. $d\mathcal{J}_c$ is the Jacobian of current objective function value. eps denotes the machine precision and maxIter is an a priori chosen number. The numerical scheme is summarized in Algorithm 1.

Algorithm 1: Gauss-Newton scheme with Armijo Line Search for image registration: $\boldsymbol{u} \leftarrow$ GNIRArmijo($\alpha, \boldsymbol{u}$)

Compute $\mathcal{J}_\alpha(\boldsymbol{u})$, $d\mathcal{J}_\alpha(\boldsymbol{u})$ and $\boldsymbol{H}$ using (16), (17) and (18), respectively;
while true **do**
  Update iteration count: iter$\leftarrow$iter + 1;
  Check the stopping rules;
  Solve quasi-Newton's equation: $\boldsymbol{H} \cdot \delta_{\boldsymbol{u}} = -d\mathcal{J}_\alpha(\boldsymbol{u})$ by using a preconditioned conjugate gradient method;
  Perform Armijo Line Search: $\boldsymbol{u}_t \leftarrow$ Armijo($\alpha, \delta_{\boldsymbol{u}}, \boldsymbol{u}$) ;
  if line search fail;
  break **then**
  end
  Update current values: $\boldsymbol{u} \leftarrow \boldsymbol{u}_t$;
  Compute $\mathcal{J}_\alpha(\boldsymbol{u})$, $d\mathcal{J}_\alpha(\boldsymbol{u})$ and $\boldsymbol{H}$ using (16), (17) and (18), respectively
end

In this section the Armijo Line Search can be briefly explained as follows. Starting with $t = 1$, the new iterate $\boldsymbol{U}^{h^{(k+1)}} = \boldsymbol{U}^{h^{(k)}} + t \cdot \delta_{\boldsymbol{U}^h}$ is used. Standard sufficient decrease condition can be written by the following form: $\mathcal{J}_\alpha(\boldsymbol{U}^{h^{(k+1)}}) < \mathcal{J}_\alpha(\boldsymbol{U}^{h^{(k)}}) + tol \cdot t \cdot ((d\mathcal{J}_\alpha(\boldsymbol{U}^{h^{(k)}}))^\top \cdot \boldsymbol{U}^{h^{(k)}})$, where let $tol = 10^{-4}$. If the above sufficient decrease condition couldn't be met, we set $t := \frac{1}{2}t$. To be safe, Armijo Linear Search would be terminated if an increment becomes relatively small. When this case occurs, optimization algorithm is concluded that it fails to converge. The algorithm is summarized in Algorithm 2.

Algorithm 2: Armijo Line Search: $\boldsymbol{u} \leftarrow$ Armijo($\alpha, \delta_{\boldsymbol{u}}, \boldsymbol{u}$)

Compute $\mathcal{J}_\alpha(\boldsymbol{u})$ and $d\mathcal{J}_\alpha(\boldsymbol{u})$ using (16) and (17), respectively;
Set $k \leftarrow 0$, $t \leftarrow 1$, MaxIter $\leftarrow 10$, and $\eta \leftarrow 10^{-4}$;
while true **do**
  Set $\boldsymbol{u}_t \leftarrow \boldsymbol{u} + t\delta_{\boldsymbol{u}}$;
  Compute $\mathcal{J}_\alpha(\boldsymbol{u}_t)$ using (16);
  if $\mathcal{J}_\alpha(\boldsymbol{u}_t) < \mathcal{J}_\alpha(\boldsymbol{u}) + t\eta(d\mathcal{J}_\alpha(\boldsymbol{u}))^\top \delta_{\boldsymbol{u}}$;
  break **then**
  end
  if $k >$ MaxIter;
  break **then**
  end
  Set $t \leftarrow \frac{t}{2}$ and $k \leftarrow k + 1$;
end
Set $\boldsymbol{u} \leftarrow \boldsymbol{u}_t$.

In order to save computational work and to speed up convergence, we combine Gauss-Newton method with multilevel scheme to solve (16). First, we provide an initial value by multilevel affine linear preregistration on the coarsest level, then solve (16) by using Gauss-Newton method with Armijo Linear Search. Second, we interpolate the coarse solution to next fine level as a initial value, then solve (16) on fine level by using the same scheme. Third, repeating the process, until the loop terminates. There are two major advantages in using multilevel scheme. Firstly, computing a minimizer need less iterations to solve optimization problems on the coarser levels. Secondly, the risk of getting trapped at unwanted minimizers is reduced. Note that every part of the discrete problem (16) is required to be continuously differentiable to make full use of efficient optimization techniques. Thus multilevel representation of given images is necessary. The objective of multilevel representation is to derive a family of continuous models for given images.

In addition, we know that affine transformation -based parametric image registration is applicable to a large class of non-rigid registration problems. In the two-dimensional case, the number of unknown parameters is 6 for an affine linear transformation image registration. The number of unknowns of deformable registration (e.g. variational models [30]) for a discrete image is proportional to the number of pixels. Thus we know that an affine method is always many orders of magnitude faster than a nonlinear method [[30]] due to much less unknowns involved. Motivated by several works (see [30] and Schmitt *et al.* [35]), we can use affine linear transformation-based parametric image registration as a pre-registration step for our new model by providing the good initial positions for the image to be registered. Next we summarize the multilevel scheme in Algorithm 3. In this Algorithm, bi-linear interpolation techniques are used for the interpolation operator denoted by I_H^h.

Algorithm 3: Multilevel Image Registration: $\boldsymbol{u} \leftarrow$ MLIR(MLData)

Maxlevel$\leftarrow$ ceil(log2(min(m_1, m_2))), % The finest level;
Minlevel$\leftarrow$ 3, % The coarsest level;
MLData, % Multilevel representation of given images R and T;
for l = Minlevel:Maxlevel **do**
  if l == Minlevel*;*
  Providing initial guess $\boldsymbol{u}_0$ by using affine linear preregistration **then**
  end
  if l == Minlevel*;*
  $\boldsymbol{u}0 \leftarrow \boldsymbol{u}_0$*;*
  else*;*
  $\boldsymbol{u}0 \leftarrow I_H^h(\boldsymbol{u})$ **then**
  end
  $\boldsymbol{u} \leftarrow$ GNIRArmijo($\alpha, \boldsymbol{u}0$) ;
end

4 Numerical experiments

In this section we present some experiments to

- compare the modeling results of our new model $\mathcal{R}^{\text{New}}$ with common variational models proposed in above Introduction;
- demonstrate the performance of our proposed Algorithm 3 for $\mathcal{R}^{\text{New}}$ with regard to parameter changes;

- show our proposed Algorithm 3 is more effective by comparing with Algorithm 1.

To measure the quality of the registered images, the relative reduction of the dissimilarity rel.SSD is used, and it is defined as follows

$$\text{rel} \cdot SSD = \frac{\mathcal{D}(\boldsymbol{u})}{\mathcal{D}_{\text{stop}}} \times 100\%$$

Where $\boldsymbol{u}$ is the current optimal value and $\mathcal{D}_{\text{stop}}$ is the value of $\mathcal{D}(\boldsymbol{u})$ at $\boldsymbol{u} = 0$. Four representative data sets (Two non-smooth registration problems and two smooth registration problems to be denoted respectively as Example 1 , Example 2 , Example 3 and Example 4)were selected for the experiments, as show respectively in Figure 1.

4.1 Comparison $\mathcal{R}^{\text{New}}$ with common regularizer techniques

In the first experiment, our aim is to investigate capabilities of $\mathcal{R}^{\text{New}}$ and common regularizer techniques for registration of the four test Examples $1-4$ in resolution 256×256 and 512×512. For the convenience of description, the common regularizer techniques such as diffusion-, elastic-, linear curvature and total variation-based image registration are denoted by $\mathcal{R}^{\text{diff}}$, $\mathcal{R}^{\text{elas}}$, $\mathcal{R}^{\text{Curv}}$ and $\mathcal{R}^{\beta\text{TV}}$, respectively. Below we mainly highlight the further gains from using $\mathcal{R}^{\text{New}}$. To be a fair comparison, we used the same Algorithm as explained in Section 3 for solving the discretized energy functional related to above regularizer techniques.

The registered results by these models are shown in Figure 2 until to Figure 9. On one hand, for the smooth registration problem (Example $3-4$), one can observe that our new model $\mathcal{R}^{\text{New}}$ works fine in producing acceptable registration results as well as do from $\mathcal{R}^{\text{Curv}}$, $\mathcal{R}^{\text{diff}}$, $\mathcal{R}^{\text{elas}}$ and $\mathcal{R}^{\text{MTV}}$ which are known to be suitable for smooth displacement; on the other hand, for the non-smooth registration problems (Example $1-2$), one can clearly see that our new model $\mathcal{R}^{\text{New}}$ produces more pleasing registration results than those from $\mathcal{R}^{\beta\text{TV}}$ and $\mathcal{R}^{\text{MTV}}$ which are known to be suitable for discontinuous displacement, especially on Example 1 with non-axis-aligned discontinuities. Numerical experiments show that our new model $\mathcal{R}^{\text{New}}$ are suitable for both smooth problems and non-smooth problems, especially for non-smooth registration problems with non-axis-aligned discontinuities. The main reason is that our new model utilize interdependence between the primary components of the deformation field for smooth and non-smooth registration problems.

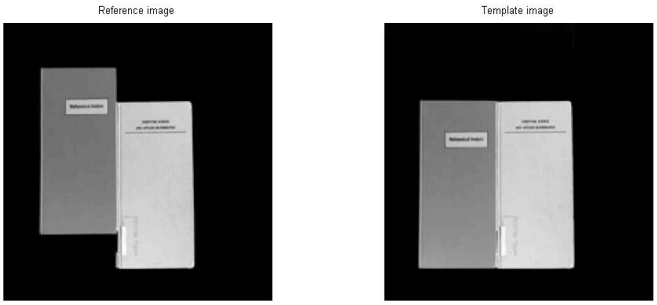
4.2 Tests of our new Algorithm 3

Here by experiments, we hope to test the convergence issues of it with regard to parameters α in the model and the mesh parameter h.

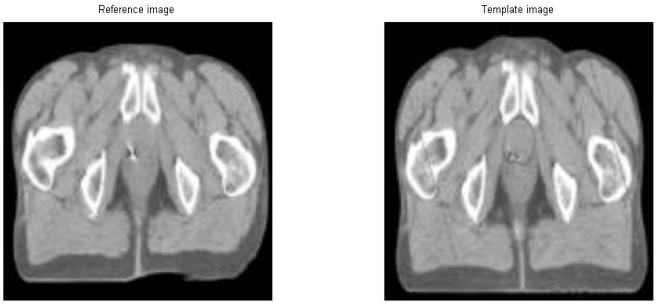
4.2.1 h-independent convergence tests

We shall resolve the same Example $2-3$ as above using an increasing sequence of resolutions (or a decreasing mesh parameter h) and show the results in Table 1. The results show that our new Algorithm 3 not only convergence within a very short time, but it is also accurate because the dissimilarities between the reference and registered images have been reduced more than 97%. For overall performance the experimental results suggest that our new Algorithm 3 would be preferred for practical applications.

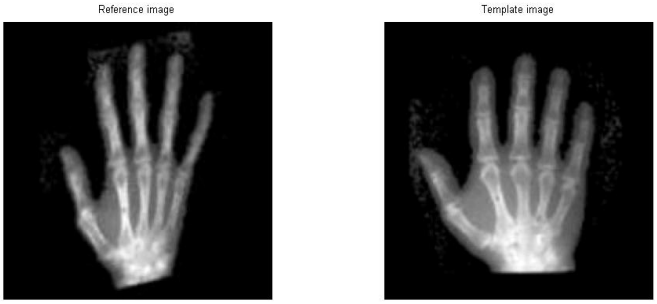
Example 1 (256×256)



Example 2 (256×256)



Example 3 (512×512)



Example 4 (512×512)

Figure 1: Four representative data sets of registration problems. Left column: reference image R, right column: template image T. Top to bottom: Example $1-2$ (non-smooth registration problems) and Example $3-4$ (smooth registration problem).

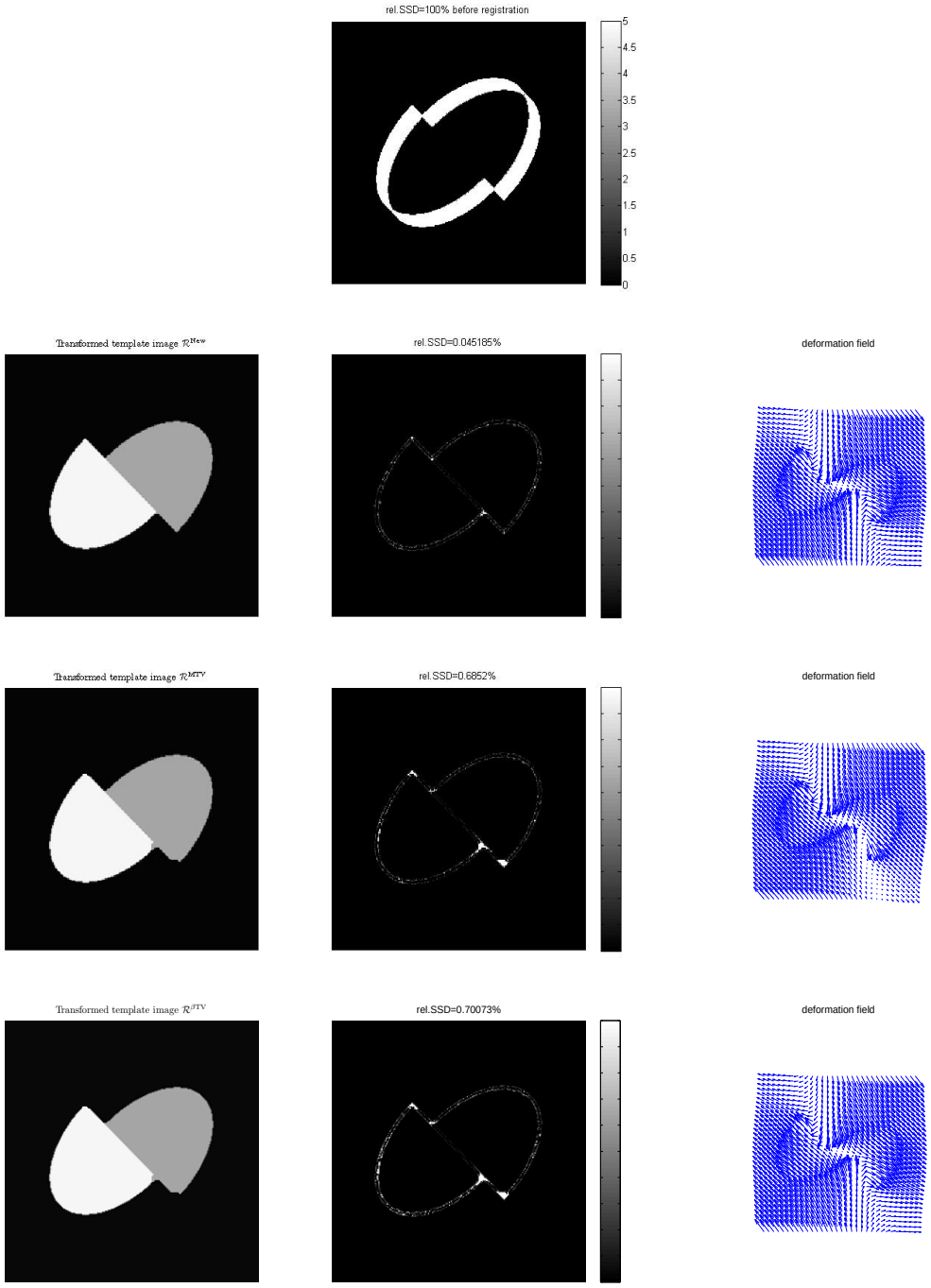


Figure 2: Registration results for two moving objects of size 256×256 (Example 1).Top row: Difference between reference image and template image before registration. The second row left: registered template image by $\mathcal{R}^{\mathrm{New}}$; The second row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\mathrm{New}}$; The second row right: deformation field from model $\mathcal{R}^{\mathrm{New}}$. The third row left: registered template image by $\mathcal{R}^{\mathrm{MTV}}$; The third row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\mathrm{MTV}}$; The third row right: c $\mathcal{R}^{\mathrm{MTV}}$. The last row left: registered template image by c; The last row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\beta\mathrm{TV}}$; The last row right: deformation field from model $\mathcal{R}^{\beta\mathrm{TV}}$. Here α were well-selected for all regularizer techniques.

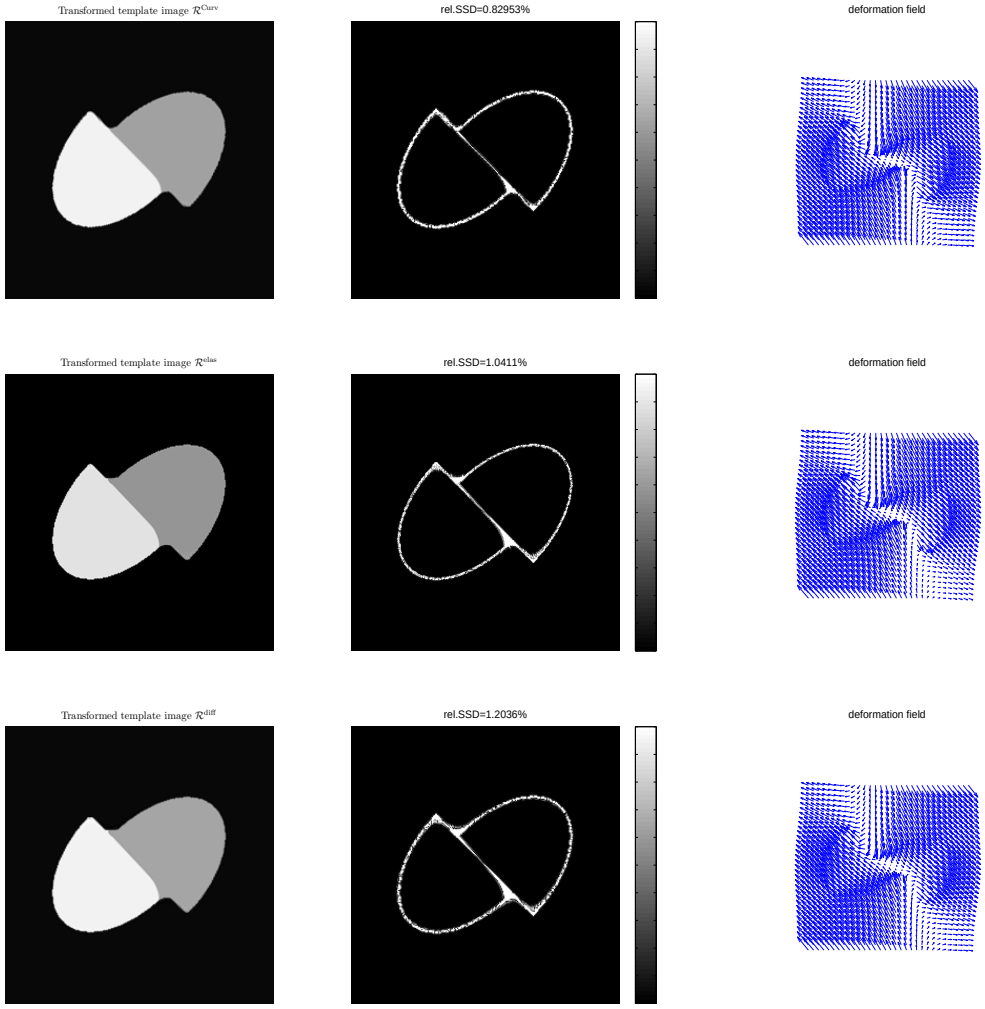


Figure 3: Registration results for two moving objects of size 256×256 (Example 1).Top row left: registered template image by $\mathcal{R}^{\text{Curv}}$; Top row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\text{Curv}}$; Top row right: deformation filed from model $\mathcal{R}^{\text{Curv}}$. The second row left: registered template image by $\mathcal{R}^{\text{elas}}$; The second row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\text{elas}}$; The second row right: deformation filed from model $\mathcal{R}^{\text{elas}}$. The last row left: registered template image by $\mathcal{R}^{\text{diff}}$; The last row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\text{diff}}$; The last row right: deformation filed from model $\mathcal{R}^{\text{diff}}$. Here α were well-selected for all regularizer techniques.

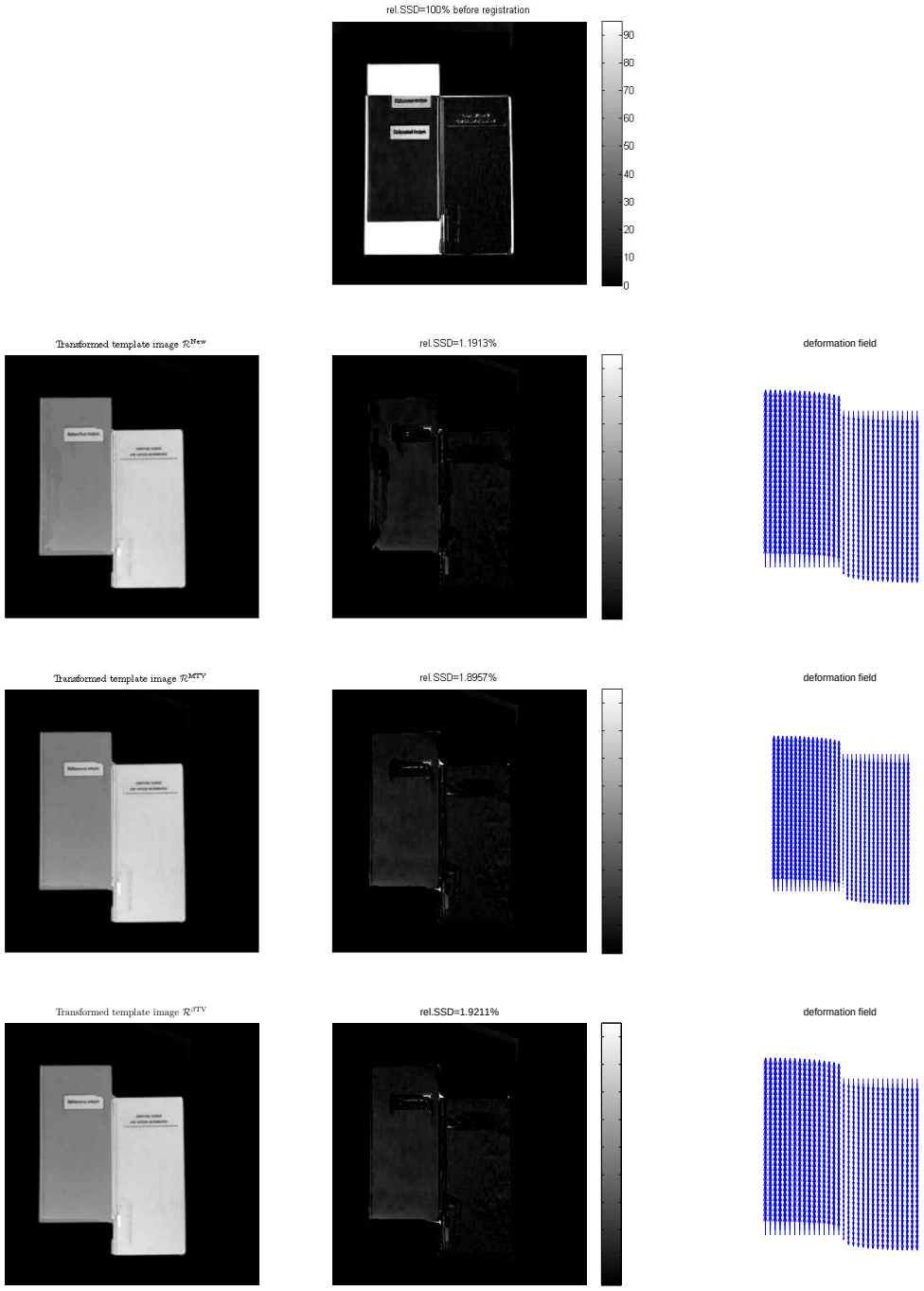


Figure 4: Registration results for two book images of size 256×256 (Example 2). Top row: Difference between reference image and template image before registration. The second row left: registered template image by $\mathcal{R}^{\mathrm{New}}$; The second row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\mathrm{New}}$; The second row right: deformation filed from model $\mathcal{R}^{\mathrm{New}}$. The third row left: registered template image by $\mathcal{R}^{\mathrm{MTV}}$; The third row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\mathrm{MTV}}$; The third row right: deformation filed from model $\mathcal{R}^{\mathrm{MTV}}$. The last row left: registered template image by $\mathcal{R}^{\beta\mathrm{TV}}$; The last row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\beta\mathrm{TV}}$; The last row right: deformation filed from model $\mathcal{R}^{\beta\mathrm{TV}}$. Here α were well-selected for all regularizer techniques.

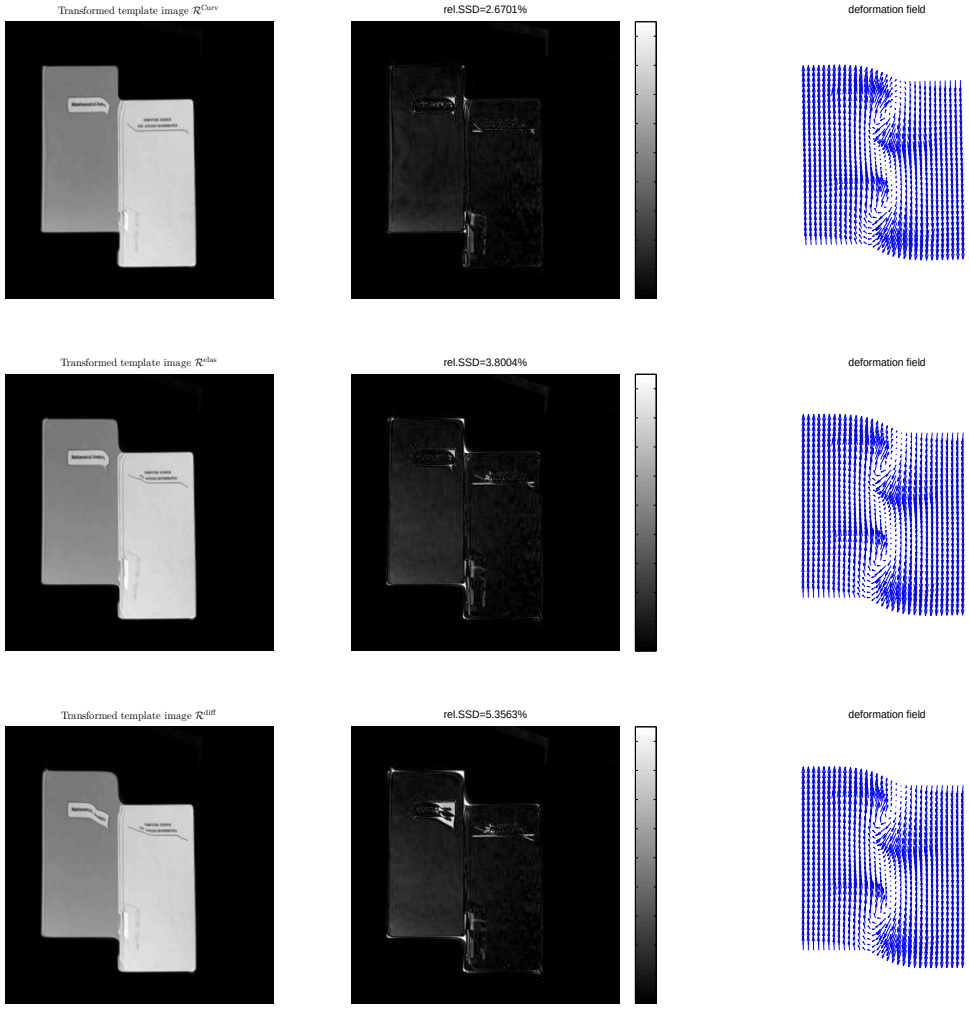


Figure 5: Registration results for two book images of size 256×256 (Example 2). Top row left: registered template image by $\mathcal{R}^{\text{Curv}}$; Top row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\text{Curv}}$; Top row right: deformation field from model $\mathcal{R}^{\text{Curv}}$. The second row left: registered template image by $\mathcal{R}^{\text{elas}}$; The second row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\text{elas}}$; The second row right: deformation field from model $\mathcal{R}^{\text{elas}}$. The last row left: registered template image by $\mathcal{R}^{\text{diff}}$; The last row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\text{diff}}$; The last row right: deformation field from model $\mathcal{R}^{\text{diff}}$. Here α were well-selected for all regularizer techniques.

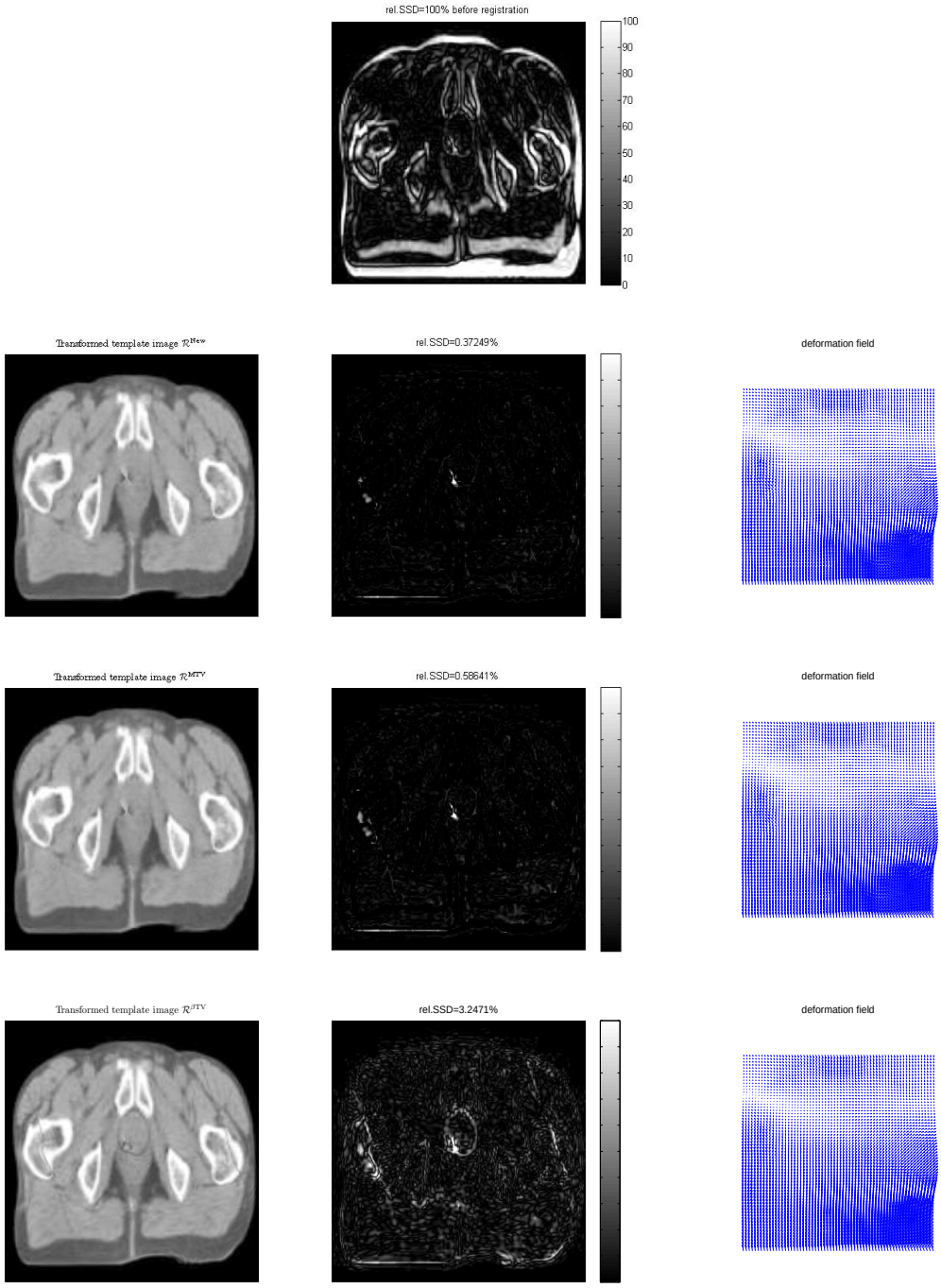


Figure 6: Registration results for MRI images of size 512×512 (Example 3). Top row: Difference between reference image and template image before registration. The second row left: registered template image by $\mathcal{R}^{\mathrm{New}}$; The second row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\mathrm{New}}$; The second row right: deformation field from model $\mathcal{R}^{\mathrm{New}}$. The third row left: registered template image by $\mathcal{R}^{\mathrm{MTV}}$; The third row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\mathrm{MTV}}$; The third row right: deformation field from model $\mathcal{R}^{\mathrm{MTV}}$. The last row left: registered template image by $\mathcal{R}^{\beta\mathrm{TV}}$; The last row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\beta\mathrm{TV}}$; The last row right: deformation field from model $\mathcal{R}^{\beta\mathrm{TV}}$. Here α were well-selected for all regularizer techniques.

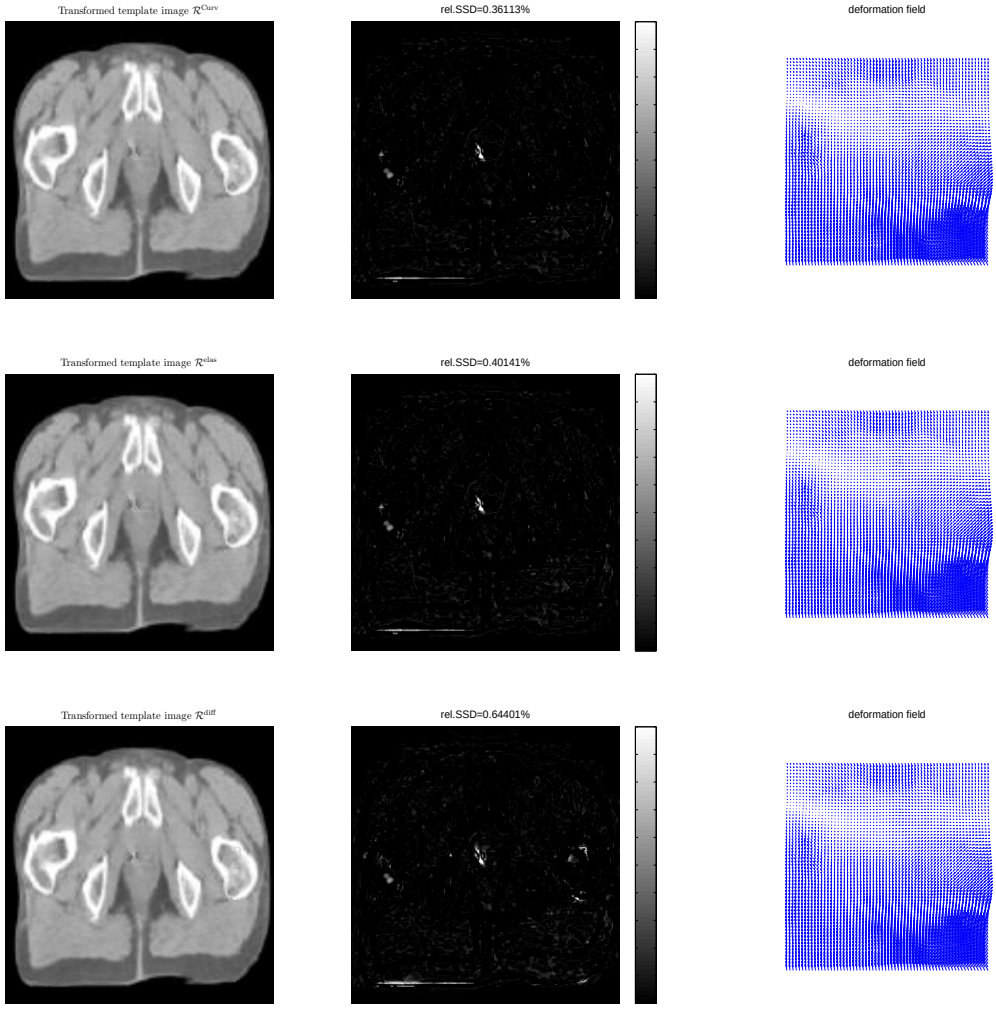


Figure 7: Registration results for MRI images of size 512×512 (Example 3). Top row left: registered template image by $\mathcal{R}^{\text{Curv}}$; Top row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\text{Curv}}$; Top row right: deformation field from model $\mathcal{R}^{\text{Curv}}$. The second row left: registered template image by $\mathcal{R}^{\text{elas}}$; The second row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\text{elas}}$; The second row right: deformation field from model $\mathcal{R}^{\text{elas}}$. The last row left: registered template image by $\mathcal{R}^{\text{diff}}$; The last row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\text{diff}}$; The last row right: deformation field from model $\mathcal{R}^{\text{diff}}$. Here α were well-selected for all regularizer techniques.

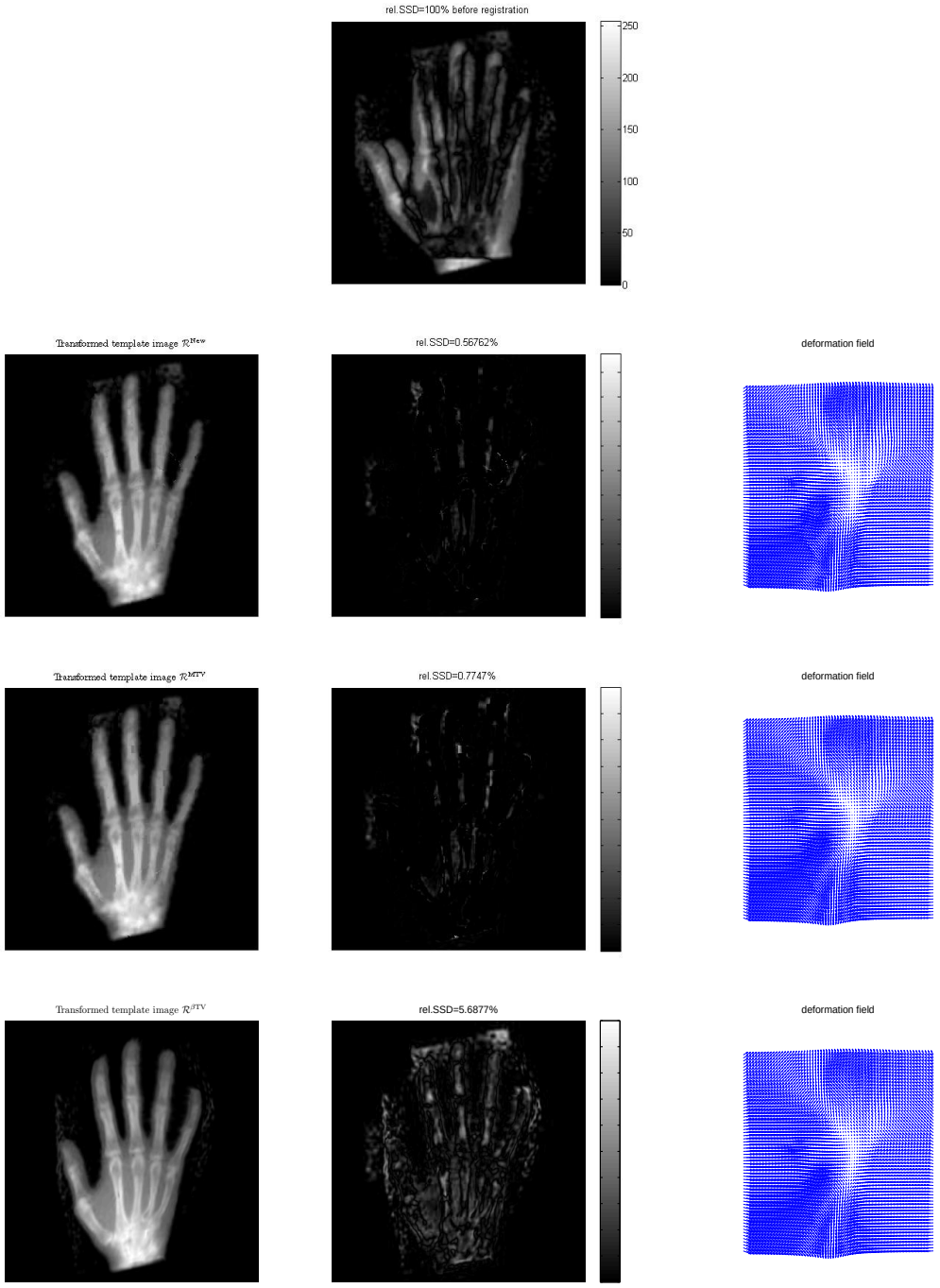


Figure 8: Registration results for **X**-ray images of size 512×512 (Example 4). Top row: Difference between reference image and template image before registration. The second row left: registered template image by $\mathcal{R}^{\mathrm{New}}$; The second row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\mathrm{New}}$; The second row right: deformation field from model $\mathcal{R}^{\mathrm{New}}$. The third row left: registered template image by $\mathcal{R}^{\mathrm{MTV}}$; The third row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\mathrm{MTV}}$; The third row right: deformation field from model $\mathcal{R}^{\mathrm{MTV}}$. The last row left: registered template image by $\mathcal{R}^{\beta\mathrm{TV}}$; The last row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\beta\mathrm{TV}}$; The last row right: deformation field from model $\mathcal{R}^{\beta\mathrm{TV}}$. Here α were well-selected for all regularizer techniques.

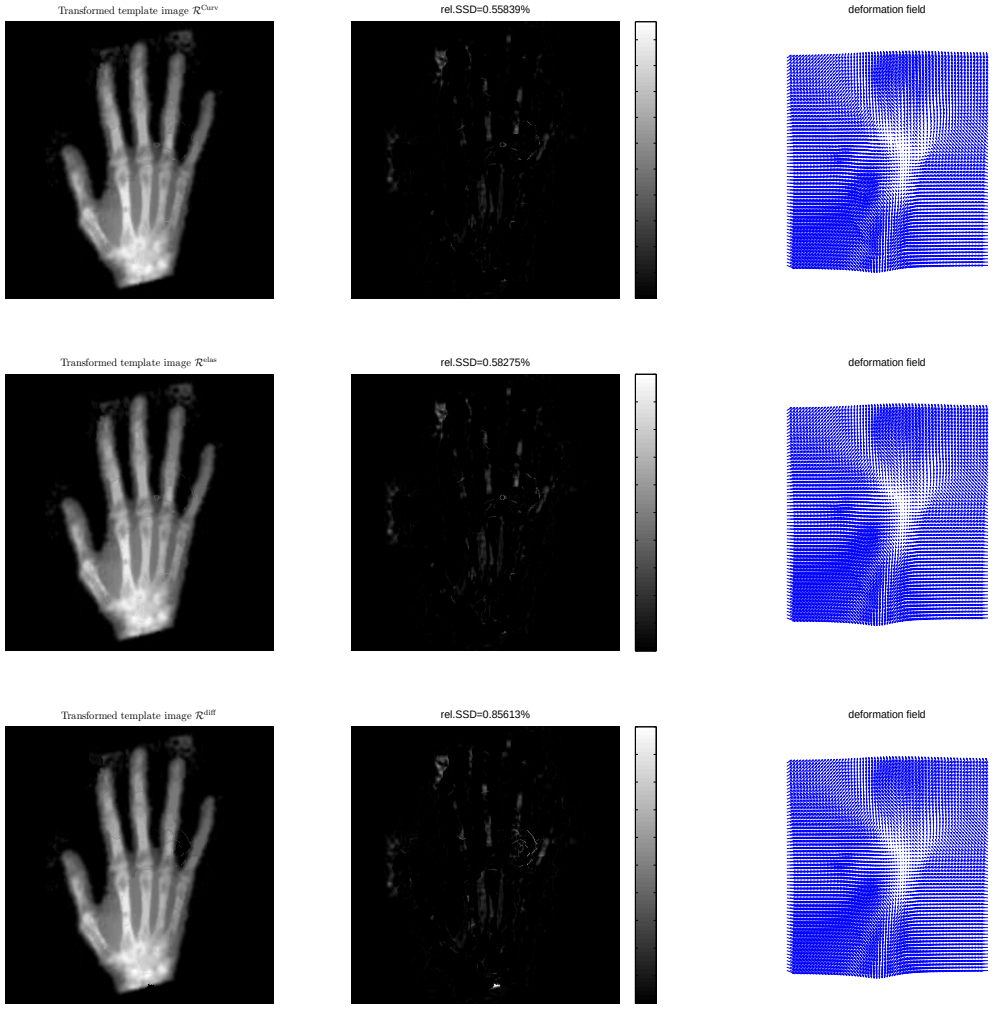


Figure 9: Registration results for **X**-ray images of size 512×512 (Example 4). Top row left: registered template image by $\mathcal{R}^{\text{Curv}}$; Top row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\text{Curv}}$; Top row right: deformation field from model $\mathcal{R}^{\text{Curv}}$. The second row left: registered template image by $\mathcal{R}^{\text{elas}}$; The second row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\text{elas}}$; The second row right: deformation field from model $\mathcal{R}^{\text{elas}}$. The last row left: registered template image by $\mathcal{R}^{\text{diff}}$; The last row middle: Difference between reference image and deformed template image after registration by $\mathcal{R}^{\text{diff}}$; The last row right: deformation field from model $\mathcal{R}^{\text{diff}}$. Here α were well-selected for all regularizer techniques.

$\mathcal{R}^{\text{New}}$ model				
Example	h	α	rel.SSD	CPU(second)
2	1/128	1.75e3	1.193%	8.9
	1/256	1.75e3	1.1918%	22.3
	1/512	1.75e3	1.1913%	76.9
Example	h	α	rel.SSD	CPU(second)
3	1/128	7.5e2	0.52252%	9.9
	1/256	7.5e2	0.42792%	26.6
	1/512	7.5e2	0.37249%	98.4

Table 1: Registration results of our Algorithm 3 for processing Example $2-3$ shown respectively in Figure 1. In the table, CPU means the total run-times including Image output and pre-registration.

4.2.2 α-dependence test

Here we analyze how sensitive the performance of our Algorithm 3 when varying α. To this end, our Algorithm 3 was tested on Example 4 (see Figure 1 last row) with the results shown in Table 2. Here the following parameters are used: $h = 1/256$ for all experiments and α is varied from 10^{-4} to 10^{5}. For this example, we can see that the performance of our Algorithm 3 is basically consistently behaved.

α	rel.SSD
10^{5}	2.3386%
10^{4}	0.90379%
10	0.64791%
1	0.65072%
10^{-1}	0.62244%
10^{-2}	0.63231%
10^{-4}	0.64264%

Table 2: Results for α-dependence tests of Algorithm 3 for Example 4 shown in Figure 1 last row .

4.2.3 Comparison Algorithm 3 with Algorithm 1

The main aim of the experiment is to show that our proposed Algorithm 3 is more effective than Algorithm 1 in achieving convergence. We took Example 4 to illustrate this point. Table 3 summarizes the registration results from Algorithm 3 and Algorithm 1 with different numbers of grid points. To be a fair comparison between them, we used the same regularizer parameter $\alpha = 10^{-1}$ and provide initial guess by using affine transformation-based pre-registration. As expected from the experiments, both methods are very accurate in registering the given images because the dissimilarities between the reference and registered images have been reduced more than 95%. However the proposed Algorithm 3 delivered more visually-pleasing registration results in terms of image quality in a very short time.

5 Conclusions

To make full use of interdependence between the primary components of the deformation for smooth and non-smooth registration problems, we propose an improved discontinuity-preserving image registration model in

	Algorithm 1 rel.SSD/CPU(Seconds)	Algorithm 3 rel.SSD/CPU(Seconds)
Example 4	$\alpha = 0.1000$	
h=1/128	2.6216%/815.8	0.7944%/9.3
h=1/256	3.2515%/2610.4	0.62244%/22.4
h=1/512	3.5241%/10766.1	0.56762%/69.8

Table 3: The registration results of the proposed numerical methods for processing Example 4 shown in Figure 1 (the last row). *rel*.SSD means the relative reduction of the dissimilarity. CPU means the total runtimes including Image output and pre-registration.

this paper. To solve the new model, we propose a method of frozen coefficients combine with Gauss-Newton scheme with Armijos Line Search and further to combine with a multilevel method to achieve fast convergence. Numerical experiments not only confirm that our proposed method is efficient and stable, but also it can give more satisfying registration results according to image quality. Future work will address our proposed Algorithm 3 for models of multimodal deformable image registration.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] G. Aubert, R. Deriche, and P. Kornprobst, *Computing optical flow via variational techniques*, SIAM J. Appl. Math., 60(1):156–182, 1999.

[2] G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, (2nd Edition). Springer, 2006.

[3] C. Sorzano, P. Thevenaz, and M. Unser, *Elastic registration of biological, images using vector-spline regularization* ,IEEE Transactions On Biomedical Engineering, 52 (4), pp. 652–663,2005.

[4] H. mei Chen, M. K. Arora, and P. K. Varshney, *Mutual information based image registration for remote sensing data*, International Journal of Remote Sensing, 24 (18), pp. 3701–3706, 2003.

[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] 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.

[7] L. Ambrosio, N. Fusco and D. Pallara, *Functions of Bounded Variation and Free Disconti- nuity Problems*, edition, Oxford University Press, 2000.

[8] C.Brito-Loeza and K. Chen, *On high-order denoising models and fast algorithms for vector-valued images*, IEEE Transactions on Image Processing, 19 (2010), pp. 1518–1527.

[9] L. G. Brown, *A survey of image registration techniques*, ACM Computing Surveys, 24 (1992), pp. 325–376.

[10] T.F. Chan, K. Chen, and J.L. Carter, *Iterative methods for solving the dual formulation arising from image restoration*, Electronic Transactions on Numerical Analysis, 26:299–311, 2007.

[11] J.V. Hajnal, D.L.G. Hill, and D. Hawkes, *Medical Image Registration*, The Biomedical Engineering Series, CRC Press, 2001.

[12] K. Chen and X.-C. Tai, *A nonlinear multigrid method for total variation minimization from image restoration*, Journal of Scientific Computing, 32(2):115–138, 2007.

[13] N. Chumchob and K. Chen , *A Robust Multigrid Approach for Variational Image Registration Models*, Journal of Computational and Applied Mathematics , Vol.236 (2011) 653–674.

[14] N. Chumchob, K. Chen, *A Variational Approach For Discontinuity-Preserving Image Registration*, East-West Journal of Mathematics, pp. 266–282, 2010.

[15] N. Chumchob, K. Chen and C. Brito-Loeza, *A Fourth Order Variational Image Registration Model And Its Fast Multigrid Algorithm*, SIAM J. Multiscale Modeling & Simulation, Vol 9(1), pp. 89–128, 2011.

[16] B. Fischer and J. Modersitzki, *Fast diffusion registration*, Contemporary Mathematics, 313 (2002), pp. 117–129.

[17] B. Fischer, and J. Modersitzki, *Curvature based Image Registration*, Journal of Mathematical Imaging and Vision, Vol.18 (2003), pp. 81–85.

[18] B. Fischer, and J. Modersitzki, *Ill-posed medicine – an introduction to image registration*, Inverse Problems, vol.24 (2008), pp. 1–19.

[19] 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, 2006, Springer-Verlag, pp. 305–323.

[20] C. Frohn-Schauf, S. Henn, and K. Witsch, *Multigrid based total variation image registration*, Comput. Vis. Sci., 11(2008), pp. 101–113.

[21] P.E.Gill, W. Murray and M. H. Wright, *Practical optimization*, Academic Press, London, 1981.

[22] E. Haber, S. Heldmann, and J. Modersitzki, *Adaptive mesh refinement for non parametric image registration*, SIAM J. Sci. Comput., 30(6): pp. 3012–3027, 2008.

[23] E. Haber, S. Heldmann, and J. Modersitzki, *A computational framework for image-based constrained registration*, Linear Algebra and its Application, vol. 431(3–4): pp. 459–470, 2009.

[24] E. Haber, R. Horesh, and J. Modersitzki, *Numerical optimization for constrained image registration*, Numer. Linear Algebra Appl. 17(2-3): pp. 343–359, 2010.

[25] E. Haber and J. Modersitzki, *Numerical solutions of volume preserving image registration*, Inverse Problems, 20 (2004), pp. 1621–1638.

[26] E. Haber and J. Modersitzki, *A Multilevel Method for Image Registration*, SIAM J.Sci.Comput., 27 (5), pp. 1594–1607, 2006.

[27] S. Henn, *A Multigrid Method for a Fourth-Order Diffusion Equation with Application to Image Processing*, SIAM J. Sci. Comput., Vol. 27 No.3(2005), pp. 831-849.

[28] 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.

[29] J.B.A. Maintz and M.A. Viergever, *A survey of medical image registration*, Med. Image. Anal., 2(1): 1–36, 1998.

[30] J. Modersitzki, *Numerical Methods for Image Registration*, Oxford University Press, New York, 2004.

[31] J. Modersitzki, *FAIR: Flexible Algorithms for Image Registration*, SIAM, Philadelphia, 2009.

[32] J.Nocedal and S.J.Wright, *Numerical optimization*, Springer-Verlag, New York, 1999.

[33] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys.D, Vol. 60(1992), pp. 259–268.

[34] J. Savage and K. Chen, *An improved and accelerated nonlinear multigrid method for total-variation denoising*, International Journal of Computer Mathematics, 82(8):1001–1015, 2005.

[35] O. Schmitt, J. Modersitzki, S. Heldmann, S.Wirtz and B. Fischer, *Image Registration of sectioned brains*, Int. J.Comp. Vision., Vol 73(1), pp. 5–39, 2007.

[36] M. Stürmer, H. Köstler, and U. Rüde, *A fast full mulitigrid solver for applications in image processing*, Numer. Linear Algebra Appl., 15 (2008), pp. 187–200.

[37] P. Thevenaz and M. Unser, *Optimization of mutual information for multiresolution image registration*, IEEE Trans. Image Process., 9 (2000), pp. 2083–2089.

[38] C.R.Vogel and M.E.Oman, *Iterative methods for total variation denoising*, SIAM J.Sci.Comput. 17 (1996)227–238.

[39] W. Zhu and T.F. Chan, *Image denoising using mean curvature*, SIAM J. Imaging Sci. 5 (1), pp. 1–32, 2012.