https://doi.org/10.1016/j.camwa.2016.10.018

Computers & Mathematics with Applications
Volume 72, Issue 12, December 2016, Pages 2887-2899

A novel high-order functional based image registration model with inequality constraint

Jin Zhang* Ke Chen† Bo Yu*

2016

Abstract

In this paper, a novel variational image registration model using a second-order functional as regularizer is presented. The main motivation for the new model stems from the LLT model (see [1]). In order to avoid mesh folding, inequality constraint on the determinant of the Jacobian matrix J of the transformation is also proposed. Furthermore, a fast solver is provided for numerical implementation of registration model with inequality constraints. Numerical experiments are illustrated to show the good performance of our new model according to the registration quality.

Keywords. Image registration, Regularization, Multilevel, Second-order functional.

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

1 Introduction

Image registration which is also called image matching is one of the most useful and fundamental tasks in imaging processing domain. It is often encountered in many fields such as astronomy, art, biology, chemistry, medical imaging and remote sensing and so on. For an overview of image registration methodology see ([2, 3, 4, 5]). Here we focus on deformable image registration in a variational framework.

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. Without loss of generality, here we focus on $d = 2$ throughout this paper, but it is readily extendable to $d = 3$ with some additional modifications. Let $\boldsymbol{x} = (x, y)^\top$, then $d_\Omega = d_x d_y$. The purpose of registration is to look for a transformation φ defined by

$$\varphi : \mathbb{R}^2 \to \mathbb{R}^2,$$

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 $\boldsymbol{u}$, $\boldsymbol{u} : \mathbb{R}^2 \to \mathbb{R}^2$, $\quad \boldsymbol{u} : \boldsymbol{x} \mapsto \boldsymbol{u}(\boldsymbol{x}) = (u_1(\boldsymbol{x}), u_2(\boldsymbol{x}))^\top$, that is to say

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

*School of Mathematical Sciences, Dalian University of Technology, Liaoning 116024, P R China.
†Centre for Mathematical Imaging Techniques and Department of Mathematical Sciences, University of Liverpool, United Kingdom (k.chen@liv.ac.uk).

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 $T(\boldsymbol{u})$. The image intensities of R and T are assumed to be comparable (i.e. in a monomodal registration) throughout this paper. In summary, the desired displacement $\boldsymbol{u}$ is a minimizer of the following joint energy functional

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

where

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

represents similarity measure which quantifies distance or similarity of transformed template image $T(\boldsymbol{u})$ and reference R and other choice is discussed in [3], $\mathcal{S}(\boldsymbol{u})$ 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 [3]. At present, a great number of regularization functionals have been proposed, such as first order derivatives-based on total variation-, diffusion- and elastic regularizer registration models and higher order derivatives-based on linear curvature, mean curvature and Gaussian curvature ones, we can refer to [3, 6, 7, 8, 9, 10, 11, 12]. As is well known, it is easy to implement for low order regularizations while they are less effective than high order ones in producing smooth displacement fields which are important in some applications including medical imaging. Although some of them high order regularizations generate more satisfactory registration results, more computational time is required owing to complexity of their regularization functional. In addition, mesh folding has not been taken into account. Searching for a model suitable for large and smooth deformation field with low computing time and no mesh folding is still a challenge. In this paper, a novel variational image registration model with inequality constraint is proposed.

The outline of the paper is organized as follows. In Section 2, we propose a new second-order functional based image registration model with inequality constraint then discuss its numerical method using a combination of the multiplier method and Gauss-Newton scheme with Armijos Line Search for solving the new model and further to combine with a multilevel method to achieve fast convergence in Section 3. Some experimental results including comparisons are illustrated in Section 4. Finally, conclusions and future work are summarized in Section 5.

2 The proposed new image registration model

In [1], Lysaker, Lundervold and Tai (LLT) proposed a second-order regularizer which has proved to be rather robust in image denoising, however, it hasn't been studied thoroughly yet for the registration problem (1). In addition, motivated by the fact that TV regularizer is much weaker than diffusion one in producing smooth displacement fields in image registration, we propose a new regularizer functional given by

$$\mathcal{S}^{\text{new}}(\boldsymbol{u}) = \frac{1}{2}\sum_{l=1}^{2}\int_\Omega |D^2(u_l)|^2 d_\Omega \tag{3}$$

where $|D^2(u_l)| = \sqrt{((u_l)_{xx})^2 + ((u_l)_{xy})^2 + ((u_l)_{yx})^2 + ((u_l)_{yy})^2} = \sqrt{\triangledown(u_l)_x \cdot \triangledown(u_l)_x + \triangledown(u_l)_y \cdot \triangledown(u_l)_y}$ is a convex functional, here symbol $\cdot$ denotes the inner product of the vectors, then equation (1) takes the following form

$$\min_{\boldsymbol{u}}\Big\{\mathcal{J}_\alpha[\boldsymbol{u}] = \frac{1}{2}\int_\Omega \big(T(\boldsymbol{u})-R\big)^2 d\Omega + \frac{\alpha}{2}\sum_{l=1}^{2}\int_\Omega (\triangledown u_{lx}\cdot\triangledown u_{lx} + \triangledown u_{ly}\cdot\triangledown u_{ly})d\Omega\Big\}. \tag{4}$$

In order to avoid mesh folding, an inequality constraint on the determinant of the Jacobian matrix J of the transformation φ is imposed on the objective function (4). Thus, the new registration model has the following form:

$$\begin{array}{ll} \min\limits_{\boldsymbol{u}} & \{\mathcal{J}_\alpha[\boldsymbol{u}] = \frac{1}{2}\int_\Omega \big(T(\boldsymbol{u})-R\big)^2 d\Omega + \frac{\alpha}{2}\sum\limits_{l=1}^{2}\int_\Omega (\triangledown(u_l)_x\cdot\triangledown(u_l)_x + \triangledown(u_l)_y\cdot\triangledown(u_l)_y)d\Omega\}, \\ \text{s.t.} & \mathcal{F}(\boldsymbol{u}) > 0, \end{array} \tag{5}$$

where

$$\begin{aligned} \mathcal{F}(\boldsymbol{u}) &= \det(J(\varphi(\boldsymbol{x})))] \\ &= \begin{vmatrix} 1+(u_1)_x & (u_1)_y \\ (u_2)_x & 1+(u_2)_y \end{vmatrix} \\ &= (1+(u_1)_x)(1+(u_2)_y) - (u_1)_y(u_2)_x \end{aligned} \tag{6}$$

Our proposed new model has the following advantages. Firstly, the new regularizer is rotational invariant. Secondly, the new registration model with regularizer $\mathcal{S}^{\text{new}}(\boldsymbol{u})$ doesn't require additional affine linear pre-registration step. Thirdly, a visually pleasing registration result can be obtained by using our new model with low computing time. Finally, there is no mesh folding for the deformed grids. Next we give numerical solution of new registration model (5).

3 Numerical solution of image registration model (5)

In general, the optimization problem (5) cannot be solved analytically, thus numerical schemes and appropriate discretizations are required. In this paper, we adopt the discretize-optimize method which aims to discretize the joint functional (5) and then solve the discrete minimization problem with inequality constraint by standard optimization methods. Although our work is related to previous work [13], they are totally different on their regularizer techniques and discrete method. Elastic regularizer with first order derivative was considered in [13], however, our new regularizer is second-order functional. Below we shall first briefly introduce the discretization we use and then specifically 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

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

3.1.1 Discretizing displacement field $\boldsymbol{u}$ and new regularizer $\mathcal{S}^{\text{new}}(\boldsymbol{u})$

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 $(u_l^h)_{i,j} = u_l^h(x_i, y_j),\ i = 1, 2, \cdots, n_1;\ j = 1, 2, \cdots, n_2$ and $l = 1, 2$. Below we 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

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

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

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

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

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

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

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

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

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 $\boldsymbol{u}_l^h(l = 1, 2)$:

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

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

$$\nabla^h \boldsymbol{u}_1^h = \begin{bmatrix} A_x^\top \\ A_y^\top \end{bmatrix} \boldsymbol{u}_1^h \triangleq B\boldsymbol{u}_1^h, \quad \nabla^h \boldsymbol{u}_2^h = \begin{bmatrix} A_x^\top \\ A_y^\top \end{bmatrix} \boldsymbol{u}_2^h \triangleq B\boldsymbol{u}_2^h \ . \tag{7}$$

Let $B_1 = BA_x^\top$; $B_2 = BA_y^\top$; $C = B_1^\top B_1 + B_2^\top B_2$, $\mathbb{C} = \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix}$ and

$$\mathcal{B}[\boldsymbol{u}] = \triangledown(u_l)_x \cdot \triangledown(u_l)_x + \triangledown(u_l)_y \cdot \triangledown(u_l)_y \tag{8}$$

Thus, we can get the discretization of (8) as following

$$\begin{aligned}
\mathbb{B}^h[\boldsymbol{U}^h] &= (BA_x^\top \boldsymbol{u}_1^h)^\top (BA_x^\top \boldsymbol{u}_1^h) + (BA_y^\top \boldsymbol{u}_1^h)^\top (BA_y^\top \boldsymbol{u}_1^h) \\
&+ (BA_x^\top \boldsymbol{u}_2^h)^\top (BA_x^\top \boldsymbol{u}_2^h) + (BA_y^\top \boldsymbol{u}_2^h)^\top (BA_y^\top \boldsymbol{u}_2) \\
&= (B_1 \boldsymbol{u}_1^h)^\top (B_1 \boldsymbol{u}_1^h) + (B_2 \boldsymbol{u}_1^h)^\top (B_2 \boldsymbol{u}_1^h) \\
&+ (B_1 \boldsymbol{u}_2^h)^\top (B_1 \boldsymbol{u}_2^h) + (B_2 \boldsymbol{u}_2^h)^\top (B_2 \boldsymbol{u}_2^h) \\
&= (\boldsymbol{u}_1^h)^\top (B_1{}^\top B_1) \boldsymbol{u}_1^h + (\boldsymbol{u}_1^h)^\top (B_2{}^\top B_2) \boldsymbol{u}_1^h \\
&+ (\boldsymbol{u}_2^h)^\top (B_1{}^\top B_1) \boldsymbol{u}_2^h + (\boldsymbol{u}_2^h)^\top (B_2{}^\top B_2) \boldsymbol{u}_2^h \\
&= (\boldsymbol{u}_1^h)^\top C \boldsymbol{u}_1^h + (\boldsymbol{u}_2^h)^\top C \boldsymbol{u}_2^h \\
&= ((\boldsymbol{u}_1^h)^\top, (\boldsymbol{u}_2^h)^\top) \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1^h \\ \boldsymbol{u}_2^h \end{bmatrix} \\
&= (\boldsymbol{U}^h)^\top \mathbb{C} \boldsymbol{U}^h
\end{aligned}$$

Thus by a midpoint quadrature rule, the new regularizer $\mathcal{S}^{\text{new}}(\boldsymbol{u}) = \frac{1}{2}\int_\Omega \mathcal{B}[\boldsymbol{u}] d_\Omega$ is descretized as

$$\mathcal{S}^h(\boldsymbol{U}^h) = \frac{1}{2} h_d (\boldsymbol{U}^h)^\top \mathbb{C} \boldsymbol{U}^h \tag{9}$$

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 [14]. The continuous smooth approximations for template T and reference R are denoted by $\boldsymbol{\mathcal{T}}$ and $\boldsymbol{\mathcal{R}}$, respectively. Next we derive discrete analogues for the particular building blocks . Let

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

$$\boldsymbol{y}_c = [y_{1,1},\ y_{2,1},\ \cdots,\ y_{n_1,1},\ y_{1,2},\ y_{2,2},\ \cdots,\ y_{n_1,2},\ \cdots,\ y_{1,n_2},\ y_{2,n_2},\ \cdots,\ y_{n_1,n_2}]^\top,$$

and $\boldsymbol{X}_c^h = [\boldsymbol{x}_c; \boldsymbol{y}_c]$.
We can get discrete reference image

$$\vec{R} = \boldsymbol{\mathcal{R}}(\boldsymbol{X}_c^h) \tag{10}$$

and discrete transformed template image

$$\vec{T}(\boldsymbol{U}^h) = \boldsymbol{\mathcal{T}}(\boldsymbol{X}_c^h + \boldsymbol{U}^h), \tag{11}$$

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

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

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 (10) and (11) our discretization of distance measure $\mathcal{D}$ (2) is straightforward:

$$\mathcal{D}^h(\boldsymbol{U}^h) = \frac{1}{2}h_1h_2(\vec{T}(\boldsymbol{U}^h) - \vec{R})^\top(\vec{T}(\boldsymbol{U}^h) - \vec{R})$$

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

$$d\mathcal{D}^h(\boldsymbol{U}^h) = h_1h_2(\vec{T}_{\boldsymbol{U}^h})^\top(\vec{T}(\boldsymbol{U}^h) - \vec{R}) \quad .$$

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

$$d^2\mathcal{D}^h(\boldsymbol{U}^h) = h_1h_2(\vec{T}_{\boldsymbol{U}^h})^\top\vec{T}_{\boldsymbol{U}^h} + h_1h_2\sum_{i=1}^{N} d_i(\boldsymbol{U}^h)\nabla^2 d_i(\boldsymbol{U}^h) \quad ,$$

where $d(\boldsymbol{U}^h) = \vec{T}(\boldsymbol{U}^h) - \vec{R} \in \mathbb{R}^N$. 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 ([3],[15]), we approximate $d^2\mathcal{D}^h(\boldsymbol{U}^h)$ by the following form

$$d^2\mathcal{D}^h(\boldsymbol{U}^h) = h_1h_2(\vec{T}_{\boldsymbol{U}^h})^\top\vec{T}_{\boldsymbol{U}^h} \quad . \tag{12}$$

3.1.4 Discretizing inequality constraint functional $\mathcal{F}(\boldsymbol{u})$

Because the discrete gradient operator ∇^h can be expressed as the product of the matrix and the vector, based on the above analysis, the discrete form of the partial derivative of the continuous displacement field component u_l can be expressed as the following form:

$$(\boldsymbol{u}_l^h)_x = A_x^\top \boldsymbol{u}_l^h \triangleq m_l \ ; \qquad (\boldsymbol{u}_l^h)_y = A_y^\top \boldsymbol{u}_l^h \triangleq w_l \quad , \quad l = 1, 2 \ .$$

Obviously, $m_l \in \mathbb{R}^N$, $w_l \in \mathbb{R}^N$, where $N = n_1 \times n_2$. Let

$$e = (1\ , 1\ , \cdots\ , \ 1)^\top \in \mathbb{R}^N$$

and

$$f = (e + m_1) \circledast (e + w_2) - w_1 \circledast m_2$$

where symbol $\circledast$ denotes element-wise multiplications of vectors. Therefore, the discrete form of the continuous inequality constraint function $\mathcal{F}(\boldsymbol{u})$ can be represented by

$$F^h(\boldsymbol{U}^h) = (f_1, f_2, \cdots, f_N)^\top \ . \tag{13}$$

Because the first order variational of the continuous inequality constraint function $\mathcal{F}(\boldsymbol{u})$ with respect of continuous displacement field $\boldsymbol{u}$ has the following form

$$d\mathcal{F}(\boldsymbol{u}) = ((u_2)_{xy} - (u_2)_{yx}\ , \ (u_1)_{yx} - (u_1)_{xy})^\top \ ,$$

we can get the discrete form of first order variational of $\mathcal{F}(\boldsymbol{u})$ is

$$dF^h(\boldsymbol{U}^h) = \begin{bmatrix} \mathbf{0} & A_y^\top A_x^\top - A_x^\top A_y^\top \\ A_x^\top A_y^\top - A_y^\top A_x^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1^h \\ \boldsymbol{u}_2^h \end{bmatrix} \triangleq \mathbb{A}\boldsymbol{U}^h \ , \tag{14}$$

obviously , $dF^h(\boldsymbol{U}^h) \in \mathbb{R}^{2N}$, $\mathbf{0} \in \mathbb{R}^{N\times N}$, $\mathbb{A} \in \mathbb{R}^{2N\times 2N}$.

3.2 Solving the discrete optimization problem

The discretized inequality constrained optimization (5) reads as follows:

$$\begin{aligned} &\min_{\boldsymbol{U}^h} \ \{\mathcal{J}_\alpha(\boldsymbol{U}^h) = \mathcal{D}^h(\boldsymbol{U}^h) + \alpha \mathcal{S}^h(\boldsymbol{U}^h)\}. \\ &\text{s.t.} \quad F^h(\boldsymbol{U}^h) > 0, \end{aligned} \tag{15}$$

To solve the above inequality constrained optimization problem (15) numerically, multiplier scheme which solves the constrained minimization problem by solving a sequence of unconstrained problem while estimating the Lagrange multipliers is used. For multiplier scheme solving inequality constrained optimization problems, more details see [16, 17, 18]. Before solving equation (15), we give the following two Lemmas and one theorem.

Lemma 1. *Let* $x \in \mathbb{R}$*, then function* $f(x) = x|x|$ *is continuously differentiable.*

Proof. $f(x) = \begin{cases} x^2, & x \geq 0 \\ -x^2, & x < 0 \end{cases}$,
when $x > 0$, $f'(x) = 2x$; when $x < 0$, $f'(x) = -2x$. Obviously, $\lim_{x\to 0+} f'(x) = \lim_{x \to 0-} f'(x) = 0$, so $f(x)$ is also differentiable at $x = 0$. We can draw the conclusion function $f(x)$ is continuously differentiable in $\mathbb{R}$. □

Lemma 2. *Let* $g(x) = f(x)|f(x)|$*, where* $x \in \mathbb{R}$*, if* $f(x)$ *is continuously differentiable, then* $g(x)$ *is also continuously differentiable.*

In fact, the proof of lemma 2 is similar to the one of lemma 1.

Theorem 1. Let *h(x)* = min{0, *f(x)*}, $x \in \mathbb{R}$, if $f(x)$ is continuously differentiable, then $[h(x)]^2 = [\min\{0, f(x)\}]^2$ is also continuously differentiable.

Proof. $h(x)$ can be written in the following form

$$h(x) = \min\{0, f(x)\} = \frac{f(x) - |f(x)|}{2} \ ,$$

so

$$[h(x)]^2 = \frac{[f(x)]^2 - f(x)|f(x)|}{2} \ ,$$

because $f(x)$ is continuously differentiable, according to lemma 2, we know that Theorem 1 is correct. □

Next, we construct the multiplier method for solving (15). Now, it's easy to derive the Augmented Lagrangian function of (15) :

$$\psi(\boldsymbol{U}^h, \lambda\ , \sigma) = \mathcal{J}_\alpha(\boldsymbol{U}^h) + \frac{1}{2\sigma} \sum_{i=1}^{N} ([\min\{0, \quad \sigma F_i^h(U^h) - \lambda_i\}]^2 - \lambda_i^2) \tag{16}$$

The formula for multiplier iteration is the following form

$$(\lambda_{k+1})_i = \max\{0, (\lambda_k)_i - \sigma F_i^h(U^{h^{(k)}})\} \tag{17}$$

Let

$$\beta_k = (\sum_{i=1}^{N} [\min\{F_i^h(U^{h^{(k)}}), \frac{(\lambda_k)_i}{\sigma}\}]^2)^{\frac{1}{2}} \tag{18}$$

Then the stopping criterion is

$$\beta_k \leq \varepsilon.$$

Algorithm 1: Multiplier scheme: $[\boldsymbol{u},\ \boldsymbol{\lambda}] \leftarrow \text{multiplier}(\mathcal{J}_\alpha(\boldsymbol{u}), d\mathcal{J}_\alpha(\boldsymbol{u}), \mathcal{F}(\boldsymbol{u}), d\mathcal{F}(\boldsymbol{u}), \boldsymbol{u}_0)$

Input : initial value $\boldsymbol{u}_0 \in \mathbb{R}^N$; the objective function $\mathcal{J}_\alpha(\boldsymbol{u})$ and its gradient $d\mathcal{J}_\alpha(\boldsymbol{u})$; the inequality constraint vector $\mathcal{F}(\boldsymbol{u})$ and its transpose of matrix Jacobian $d\mathcal{F}(\boldsymbol{u})$;
Set maxk $\leftarrow 10$, $\sigma_1 \leftarrow 1$, $\varepsilon \leftarrow 10^{-5}$, $\vartheta \leftarrow 0.3$, $\eta \leftarrow 0.2$ and $k \leftarrow 0$;
Set $\boldsymbol{\lambda}_1 \leftarrow (10^{-3}, 10^{-3}, \cdots, 10^{-3})^\top \in \mathbb{R}^N$ and $\beta_k \leftarrow 10$;
while $\beta_k > \varepsilon$ and $k <$ maxk **do**
  Solving unconstrained subproblem (16) by using Gauss-Newton scheme with Armijo line search ;
  $[\boldsymbol{u}] \leftarrow \text{GNIRArmijo}(\alpha, \boldsymbol{u}_0, \mathcal{J}_\alpha(\boldsymbol{u}), \mathcal{F}(\boldsymbol{u}), d\mathcal{J}_\alpha(\boldsymbol{u}), d\mathcal{F}(\boldsymbol{u}), \boldsymbol{\lambda}_1, \sigma)$;
  Computing β_k defined by (18);
  if $\beta_k > \varepsilon$;
  break **then**
  end
  Otherwise;
  if $k \geq 2$ and $\beta_\text{k} \geq \vartheta\beta_{\text{k}-1}$;
  Set $\sigma \leftarrow \eta\sigma$ **then**
  end
  Updating the multiplier vectors. Computing $\boldsymbol{\lambda}_{k+1}$ by using (17);
  Set $k \leftarrow k+1$;
  Set $\beta_{k-1} \leftarrow \beta_k$
end
.

Note that despite including min function in equation (16), by Theorem 1, we know that the augmented Lagrangian function is still continuously differentiable. The above detailed steps of the multiplier scheme is summarized in Algorithm 1.

In Algorithm 1, to solve the above unconstrained subproblem (16), standard optimization technique Gauss-Newton scheme is used. The main idea is to linearize ψ which is replaced by a quadratic $\hat{\psi}$ near the previous iterative value $\boldsymbol{U}^{h^{(k)}}$ by the Taylor expansion given by

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

where $d\psi(\boldsymbol{U}^{h^{(k)}})$, $\boldsymbol{H}$ are the Jacobian and the approximation of the Hessian of ψ at $\boldsymbol{U}^{h^{(k)}}$. For $d^2\mathcal{D}^h(\boldsymbol{U}^{h^{(k)}})$, $\mathbb{C}$ and $(M(\boldsymbol{U}^{h^{(k)}}))^\top M(\boldsymbol{U}^{h^{(k)}})$ are both positive semi-define, we know that $\boldsymbol{H}$ is also positive semi-definite. Hence, $\hat{\psi}$ is convex. see [18] for an extended description. Next we describe the specific steps.

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

$$d\psi(\boldsymbol{U}^{h^{(k)}}) = d\mathcal{D}^h(\boldsymbol{U}^{h^{(k)}}) + \alpha h_d\ \mathbb{C}\ \boldsymbol{U}^{h^{(k)}} + (M(\boldsymbol{U}^{h^{(k)}}))\circledast(\sigma F^h(\boldsymbol{U}^{h^{(k)}}) - \boldsymbol{\lambda}) \tag{19}$$

and

$$\boldsymbol{H} = d^2\mathcal{D}^h(\boldsymbol{U}^{h^{(k)}}) + \alpha h_d\mathbb{C} + (M(\boldsymbol{U}^{h^{(k)}}))(M(\boldsymbol{U}^{h^{(k)}}))^\top, \tag{20}$$

where $M(\boldsymbol{U}^{h^{(k)}}) = dF^h(\boldsymbol{U}^h) \in \mathbb{R}^{2N}$. Then perturbation $\delta_{\boldsymbol{U}^h}$ can be obtained by solving linear equation

$$\boldsymbol{H}\delta_{\boldsymbol{U}^h} = -d\psi(\boldsymbol{U}^{h^{(k)}})\ . \tag{21}$$

Usually, H is positive definite, thus we can solve (21) using a preconditioned conjugate gradient method. To guarantee the reduction of the objective function $\psi(\boldsymbol{U}^h)$, a standard Armijo line search scheme is used , details see [18]. Detailed algorithm is summarized in Algorithm 2. The above Gauss-Newton scheme is

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

Compute $\psi(\boldsymbol{u})$ and $d\psi(\boldsymbol{u})$ using (16) and (19), respectively;
Set $t \leftarrow 1$, MaxIter $\leftarrow 10$, and $\eta \leftarrow 10^{-4}$;
for $k = 1 : \text{MaxIter}$ **do**
  Set $\boldsymbol{u}_t \leftarrow \boldsymbol{u} + t\delta_{\boldsymbol{u}}$;
  Compute $\psi(\boldsymbol{u}_t)$ using (16);
  if $\psi(\boldsymbol{u}_t) < \psi(\boldsymbol{u}) + t\eta(d\psi(\boldsymbol{u}))^\top \delta_{\boldsymbol{u}}$;
  break **then**
  end
  Set $t \leftarrow \frac{t}{2}$;
end
Set $\boldsymbol{u} \leftarrow \boldsymbol{u}_t$.

Algorithm 3: Gauss-Newton scheme with Armijo Line Search for image registration: $\boldsymbol{u} \leftarrow \text{GNIRArmijo}(\alpha, \boldsymbol{u}, \mathcal{J}_\alpha(\boldsymbol{u}), \mathcal{F}(\boldsymbol{u}), d\mathcal{J}_\alpha(\boldsymbol{u}), d\mathcal{F}(\boldsymbol{u}), \boldsymbol{\lambda}, \sigma)$

Set $k \leftarrow 0$, maxIter$\leftarrow$10;
while true **do**
  Compute $\psi(\boldsymbol{u})$, $d\psi(\boldsymbol{u})$ and $\boldsymbol{H}$ using (16), (19) and (20), respectively;
  Update iteration count: $k \leftarrow k + 1$;
  Check the stopping rules: $k >$ maxIter;
  Solve quasi-Newton's equation: $\boldsymbol{H}\delta_{\boldsymbol{u}} = -d\psi(\boldsymbol{u})$ by using a preconditioned conjugate gradient method;
  if $\|\delta_{\boldsymbol{u}}\| < \text{tol}$;
  break **then**
  end
  Perform Armijo Line Search: $\boldsymbol{u}_t \leftarrow \text{Armijo}(\alpha, \delta_{\boldsymbol{u}}, \boldsymbol{u})$;
  if line search fail;
  break **then**
  end
  Update current values: $\boldsymbol{u} \leftarrow \boldsymbol{u}_t$;
end

summarized in Algorithm 3.

In order to save computational work and to speed up convergence, we combine Gauss-Newton method with multilevel scheme to solve (16). First, on the coarsest level we solve (16) by using Gauss-Newton method with Armijo Linear Search with initial value $\boldsymbol{U}^{h^{(0)}} = 0$. 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 in the trap of unwanted minimizers is highly 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. Next the multilevel scheme is summarized in Algorithm 4. Where bi-linear interpolation operator is denoted by I_H^h.

Algorithm 4: Multilevel Image Registration: $\boldsymbol{u} \leftarrow$ MLIR(MLData)

```
Maxlevel← ceil(log2(min(m1, m2))),      % The finest level;
Minlevel← 3,        % The coarsest level;
MLData,   % Multilevel representation of given images R and T ;
for l = Minlevel:Maxlevel do
    if l = Minlevel;
    u0 = 0;
    else;
    u0 ← I_H^h(u) then
    end
    u ← GNIRArmijo(α, u0, J_α(u), F(u), dJ_α(u), dF(u), λ, σ) ;
end
```

4 Numerical experiments

To illustrate the good performance of our new model, we compare it with three representative higher models based on linear curvature [10], mean curvature[11] and Gaussian Curvature [12] using two numerical examples. We use the relative reduction of the dissimilarity which is given by [3, 11]

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

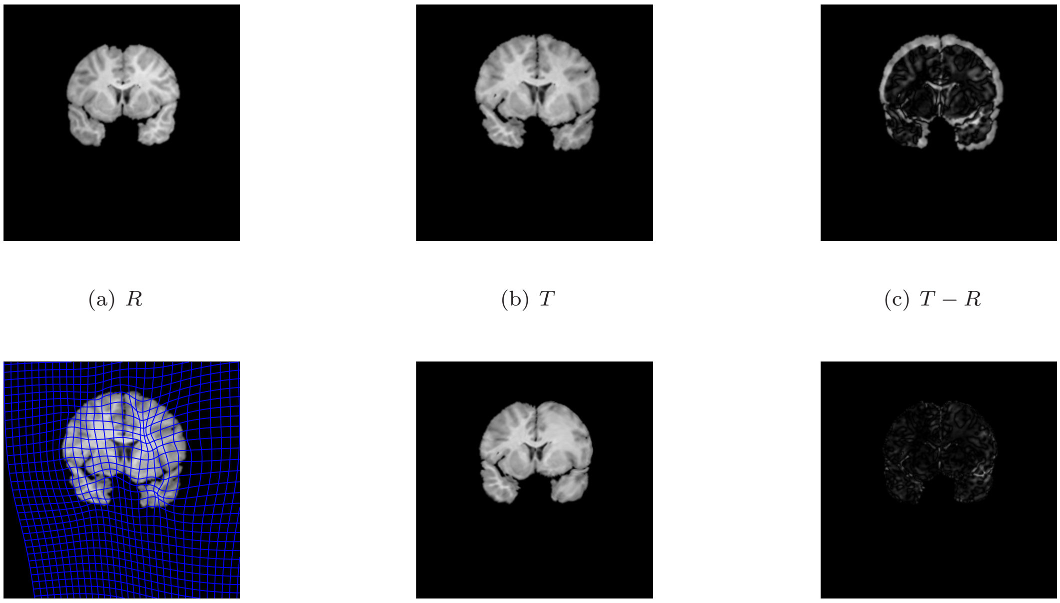
and the minimum value $\mathcal{M}$ of the determinant of the Jacobian matrix J of the transformation φ used in [12]

$$J = \begin{bmatrix} 1 + u_{1x} & u_{1y} \\ u_{2x} & 1 + u_{2y} \end{bmatrix}, \quad \mathcal{M} = \min(\det(J))$$

to measure the quality of registered images.

4.1 Test 1: A Pair of Brain MR Images

A pair of real medical images of size 256×256 are used in the first experiment. The test images and registered results using our new model are shown in Figure 1. The transformed template images for other three representative high models are shown in Figure 2. In Table 1, we record the values of the quantitative measurements for Example 1 using our new model and other three high order models. Although these four



(a) R (b) T (c) $T-R$

(d) T and $\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})$ (e) $T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x}))$ (f) $T(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x}))-R$

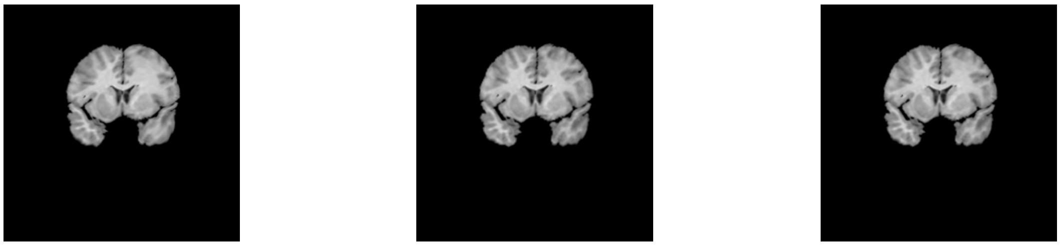
Figure 1: Registration results for a pair of Brain MR Images using our new model. (a) reference image, (b) template image, (c) difference before registration, (d) template and transformation $\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})$, (e) the transformed template image using our new model, (f) difference after registration.

high order models can produce satisfactory registration results, mean curvature-, and Gaussian curvature-based image registration models require more computational time due to complexity of their regularization functional. In table 1, we can see that our new model gives more registration quality with less time.

To assess how our new model is affected when varying regularize parameter α, Algorithm 4 was tested on Example 1 (See figure 1 (a) and (b)) with the results shown in Table 2. Here α is varied from 0.16 to 1000. In Table 1, we find that the transformation become poor when α become large, while α is greater than or equal to 0.16, the corresponding transformation $\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x})$ is one to one. The selection of appropriate α is a separate but important matter for it is generally unknown a priori and it appreciably affects on the qualities of registered images and the Algorithm 4 performance. Nevertheless, for the range of tested α in Table 2, our proposed new model still deliver better registration results in a appropriate range of α, thus for this example, the exact value of α isn't required as any α between 0.16 and 0.5 can give satisfactory results.

Model	Linear Curvature	Mean Curvature	Gaussian Curvature	New Model
α	0.4	0.0001	0.0001	0.16
Time (Seconds)	29.7	830.2	1053.7	53
ε (%)	11.95	19.98	10.62	7.42
$\mathcal{M}$	0.0184	0.8240	0.0138	0.0345

Table 1: Quantitative measurements for all models for processing Examples 1 shown in Figure 1 (a) and (b). $\mathcal{F}>0$ indicates that the deformation doesn't consist of folding and cracking of the deformed grid.



(a) Linear Curvature [10] (b) Mean Curvature [11] (c) Gaussian Curvature [12]

Figure 2: Comparison of registered results of three representative higher models .

α	0.16	0.17	0.18	0.19	0.2	0.4	0.5	1	10	100	1000
ε (%)	7.42	7.56	7.68	7.86	7.90	9.50	10.08	12.20	20.06	28.83	37.73
$\mathcal{M}$	0.0345	0.0443	0.0496	0.0579	0.0657	0.1462	0.1863	0.4111	0.8287	0.9959	1
MFN	0	0	0	0	0	0	0	0	0	0	0

Table 2: Comparisons for the regularizer parameter α-dependence using Example 1 (See figure 1 (a) and (b)). MFN denotes the mesh folding number of transformation $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$.

4.2 Test 2: A Pair of Synthetic Images

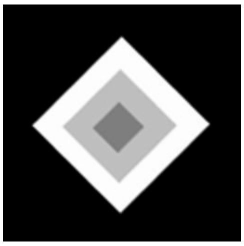
A pair of synthetic images of size 256×256 with piecewise constant for Test 2 need to be aligned. Figure 3 show the effects of using our new model and Figure 4 represent comparisons of transformations from several high-order regularizers. Table 3 records the results for Test 2. In Table 3, for a much smaller regularization parameter α, we can see all four models work fine in producing satisfactory registration results, although the registered result by our new model has the best value of ε. However, other three competitive high order models have mesh folding. In addition, we can also see the non-physical folding of meshes in Figure 4. For this example, an accurate regularizer parameter α is also unneeded. In Table 4, we find our proposed new model produce acceptable registration results for any α between 2.3×10^{-4} and 0.1.

Model	Linear Curvature	Mean Curvature	Gaussian Curvature	New Model
$\alpha(e^{-4})$	2.3	2.3	2.3	2.3
ε (%)	0.011	0.009	0.214	0.007
MFN	486	168	597	0
$\mathcal{M}$	-1.8381	-0.6160	-0.1952	0.0465

Table 3: Quantitative measurements for all models for processing Examples 2 shown in Figure 3 (a) and (b). $\mathcal{M} > 0$ indicates that the deformation doesn't consist of folding and cracking of the deformed grid. MFN denotes the mesh folding number of transformation $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$.

$\alpha = e^{-4}$	2.3	3	5	8	10	50	80	100	10^3	10^4	10^5
ε (%)	0.0068	0.0074	0.0086	0.0101	0.0109	0.0189	0.0225	0.0248	0.0672	0.2854	1.2175
$\mathcal{M}$	0.0465	0.1304	0.2090	0.2266	0.2656	0.2222	0.4380	0.3903	0.5554	0.5765	0.5494

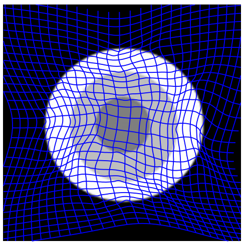
Table 4: Comparisons for the regularizer parameter α-dependence using Example 2 (See figure 3 (a) and (b)). $\mathcal{M} > 0$ indicates that the deformation doesn't consist of folding and cracking of the deformed grid.



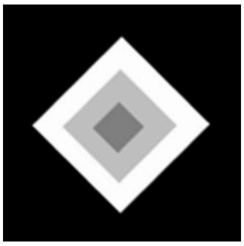
(a) R

(b) T

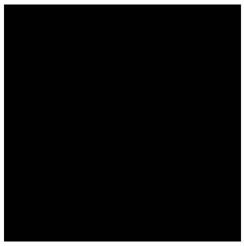
(c) $T - R$



(d) T and $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$

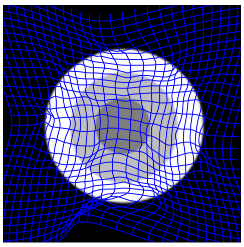


(e) $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}))$

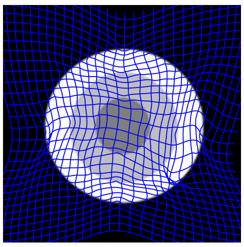


(f) $T(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})) - R$

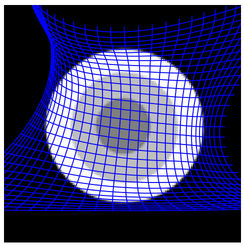
Figure 3: Registration results for a pair of synthetic images using our new model. (a) reference image, (b) template image, (c) difference before registration, (d) template and transformation $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$, (e) the transformed template image using our new model, (f) difference after registration.



(a) Linear Curvature [10]



(b) Mean Curvature [11]



(c) Gaussian Curvature [12]

Figure 4: Comparison of transformations of three representative higher models .

5 Conclusions

Motivated by the LLT model(see [1]), we proposed a new second-order functional based image registration model. The discretize-optimize method combining with multilevel scheme is used to solve the new model. For the ease of comparison, three representative higher models based on linear curvature [10], mean curvature[11] and Gaussian Curvature [12] are used for mono-modality images. Numerical experiments confirm that our new model is more effective and flexible than the competing models.

References

[1] M. Lysaker, Arvid Lundervold and Xue-Cheng Tai, *Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time*,IEEE Transactions on Image Processing, 12(12): 1579–1590, 2003.

[2] A. Sotiras, C. Davatzikos, and N. Paragios, *Deformable medical image registration: A survey*,IEEE Trans. Med. Imaging, 32(7): 1153–1190, 2013.

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

[4] B. Zitov and J. Flusser, *Image registration methods: a survey*, Image Vision Comput., 21(11): 977–1000, 2003.

[5] J. P. Pluim, J. B. Maintz, and M. A. Viergever, *Mutual information based registration of medical images: a survey.*, IEEE Trans. Med. Imaging, 22(8): 986–1004, 2003.

[6] 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, Springer-Verlag, 305–323, 2006.

[7] W.R. Hu, Y. Xie, L. Li and W.S. Zhang, *A* $TV - l_1$ *based nonrigid image registration by coupling parametric and non-parametric transformation*, International Journal of Automation and Computing, 12(5): 467–481, 2015.

[8] B. Fischer and J. Modersitzki, *Fast diffusion registration*, Contemporary Mathematics, 313: 117–129, 2002.

[9] Chaim Broit , *Optimal registration of deformed images*, University of Pennsylvania Philadelphia, PA, USA, 1981.

[10] B. Fischer, and J. Modersitzki, *Curvature based Image Registration*, Journal of Mathematical Imaging and Vision, 18: 81–85, 2003.

[11] N. Chumchob, K. Chen and C. Brito, *A fourth order variational image registration model and its fast multigrid algorithm*, SIAM J. Multiscale Modeling & Simulation, 9(1): 89–128, 2011.

[12] I. Mazlinda, K. Chen and C. Brito, *A novel variational model for image registration using Gaussian curvature*,Journal of Geometry, Imaging and Computing, 1(4): 417–446, 2014.

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

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

[15] 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: 201–218, 2008.

[16] A.R.Conn, N.Gould, and P.L. Toint, *A globally convergent lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds*, Math. Comput.,66(217): 261–288, 1997.

[17] J.F.Bonnans,J.C.Gilbert,C.Lemaréchal, and C.Sagastizábal, *Numerical optimization–Theoretical and Practical Aspects*, 2nd ed.Berlin,Germany: Springer-Verlag, 2006.

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

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

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