East Asian Journal on Applied Mathematics
https://doi.org/10.4208/eajam.200816.031216a

An Efficient Numerical Method for Mean Curvature-based Image Registration Model*

Jin Zhang[†] Ke Chen[‡] Fang Chen[§] Bo Yu

2017

Abstract

Mean curvature-based image registration model firstly proposed by Chumchob-Chen-Brito (2011) offered a better regularizer technique for both smooth and non-smooth deformation fields. However, it is extremely challenging to solve efficiently this model and the existing methods are slow or become efficient only with strong assumptions on the smoothing parameter β. In this paper, we take a different solution approach. Firstly, we discretize the joint energy functional, following an idea of relaxed fixed point is implemented and combine with Gauss-Newton scheme with Armijo's Linear Search for solving the discretized mean curvature 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, Mean Curvature.

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. [4, 5, 6, 1, 2, 3].

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

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

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

*This work was supported by the UK EPSRC grant (number EP/K036939/1), the National Natural Science Foundation of China (11571061, 11501038), and Beijing Outstanding Talents Foundation (2014000020124G108).

[†]School of Mathematical Sciences, Dalian University of Technology, Liaoning 116024, P R China (zhangjinsunny321@163.com).

[‡]Centre for Mathematical Imaging Techniques and Department of Mathematical Sciences, University of Liverpool, United Kingdom (k.chen@liv.ac.uk).

[§]School of Applied Science, Beijing Information Science and Technology University, Beijing 100192, P R China .

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

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

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 [5]. At present, the common regularizer techniques such as diffusion-, elastic-, or linear curvature-based image registration can generate globally smooth displacement, more details see [7, 8, 9, 12, 11, 10, 5, 13] 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 [14, 15, 16]. Nevertheless, the TV model may not give satisfactory registration results for smooth displacement. In this paper, we consider mean curvature regularizer which is able to solve both smooth and non-smooth registration problems as introduced by Chumchob-Chen-Brito [23]:

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

here $\kappa(u_l)=\nabla\cdot\frac{\nabla u_l}{|\nabla u_l|_\beta}$, and β is a very small positive parameter to avoide non-differentiable at zero, more details see [23, 14, 15, 16]. Thus the original joint energy functional (1) becomes

$$\min_{\boldsymbol{u}}\Big\{\mathcal{J}_\alpha[\boldsymbol{u}]=\frac{1}{2}\int_\Omega\big(T(\boldsymbol{u})-R(\boldsymbol{x})\big)^2d\boldsymbol{x}+\alpha\cdot\mathcal{R}^{\mathrm{CCB}}(\boldsymbol{u})\Big\},\tag{4}$$

and the corresponding Euler-Lagrange (EL) equation for (4) is the following

$$\begin{cases}(T(\boldsymbol{u})-R)\partial_{u_1}T(\boldsymbol{u})+\alpha\nabla\cdot(\frac{1}{|\nabla u_1|_\beta}\nabla\kappa(u_1)-\frac{\nabla u_1\cdot\nabla\kappa(u_1)}{(|\nabla u_1|_\beta)^3}\nabla u_1)=0\\(T(\boldsymbol{u})-R)\partial_{u_2}T(\boldsymbol{u})+\alpha\nabla\cdot(\frac{1}{|\nabla u_2|_\beta}\nabla\kappa(u_2)-\frac{\nabla u_2\cdot\nabla\kappa(u_2)}{(|\nabla u_2|_\beta)^3}\nabla u_2)=0,\end{cases}\tag{5}$$

with boundary conditions $\langle\nabla u_l,\boldsymbol{\nu}\rangle_{\mathbb{R}^2}=\langle\nabla\kappa(u_l),\boldsymbol{\nu}\rangle_{\mathbb{R}^2}=0$ on $\partial\Omega$, $l=1,2$ and $\boldsymbol{\nu}$ is the unit outward normal vector. It is very difficult to solve efficiently equation (5) due to its high nonlinearity. Some possible numerical methods such as fixed point methods [20, 21, 38] and Newton method do not work for (5). Next we briefly review the existing numerical algorithms.

1) Time marching method. Time marching method [23] is applied to solve the nonlinear parabolic system of (5) instead of the nonlinear elliptic system of (5) by introducing time variable t:

$$\begin{cases}\partial_t u_1+(T(\boldsymbol{u})-R)\partial_{u_1}T(\boldsymbol{u})+\alpha\nabla\cdot(\frac{1}{|\nabla u_1|_\beta}\nabla\kappa(u_1)-\frac{\nabla u_1\cdot\nabla\kappa(u_1)}{(|\nabla u_1|_\beta)^3}\nabla u_1)=0\\\partial_t u_2+(T(\boldsymbol{u})-R)\partial_{u_2}T(\boldsymbol{u})+\alpha\nabla\cdot(\frac{1}{|\nabla u_2|_\beta}\nabla\kappa(u_2)-\frac{\nabla u_2\cdot\nabla\kappa(u_2)}{(|\nabla u_2|_\beta)^3}\nabla u_2)=0,\end{cases}$$

although this scheme is very easy to implement, it is very slow to converge because the length of the time-step is required to be very small for stability.

2)Stabilized fixed point (SFP) method. The general fixed point schemes don't work for (5) due to its high nonlinearity. In [23], the authors proposed a stabilized fixed point method by adding suitable stabilizing terms. Its main idea is to split the EL equation (5) into the convex part which is treated implicitly and the non-convex part which is treated explicitly. The corresponding stabilized fixed point equation takes the following form:

$$\begin{cases} -\gamma_1 \nabla \cdot \frac{\nabla u_1^{(k+1)}}{\left|\nabla u_1^{(k)}\right|_\beta} - \alpha \nabla \cdot \left(\frac{\nabla u_1^{(k)} \cdot \nabla \kappa(u_1^{(k)})}{\left|\nabla u_1^{(k)}\right|_\beta^3} \nabla u_1^{(k+1)}\right) + \sigma_{11}^{(k)} u_1^{(k+1)} + \sigma_{12}^{(k)} u_2^{(k+1)} \\ \quad = -\gamma_1 \nabla \cdot \frac{\nabla u_1^{(k)}}{\left|\nabla u_1^{(k)}\right|_\beta} + \sigma_{11}^{(k)} u_1^{(k)} + \sigma_{12}^{(k)} u_2^{(k)} - f_1(\boldsymbol{u}^{(k)}) - \alpha \nabla \cdot \left(\frac{\nabla \kappa(u_1^{(k)})}{\left|\nabla u_1^{(k)}\right|_\beta}\right) \\ -\gamma_2 \nabla \cdot \frac{\nabla u_2^{(k+1)}}{\left|\nabla u_2^{(k)}\right|_\beta} - \alpha \nabla \cdot \left(\frac{\nabla u_2^{(k)} \cdot \nabla \kappa(u_2^{(k)})}{\left|\nabla u_2^{(k)}\right|_\beta^3} \nabla u_2^{(k+1)}\right) + \sigma_{22}^{(k)} u_2^{(k+1)} + \sigma_{21}^{(k)} u_1^{(k+1)} \\ \quad = -\gamma_2 \nabla \cdot \frac{\nabla u_2^{(k)}}{\left|\nabla u_2^{(k)}\right|_\beta} + \sigma_{21}^{(k)} u_1^{(k)} + \sigma_{22}^{(k)} u_2^{(k)} - f_2(\boldsymbol{u}^{(k)}) - \alpha \nabla \cdot \left(\frac{\nabla \kappa(u_2^{(k)})}{\left|\nabla u_2^{(k)}\right|_\beta}\right) \end{cases} \tag{6}$$

where

$$f_l(\boldsymbol{u}^{(k)}) = (T(\boldsymbol{u}^{(k)}) - R)\partial_{u_l} T(\boldsymbol{u}^{(k)}),$$

$$\sigma_{l1}^{(k)} = (\partial_{u_l} T(\boldsymbol{u}^{(k)}))(\partial_{u_1} T(\boldsymbol{u}^{(k)})),$$

$$\sigma_{l2}^{(k)} = (\partial_{u_l} T(\boldsymbol{u}^{(k)}))(\partial_{u_2} T(\boldsymbol{u}^{(k)})), \quad l = 1, 2.$$

The stabilized fixed point method is convergent providing that the smoothing parameter β in (6) is not too small, otherwise, convergence is very slow.

3) Primal-dual fixed point method. We note that above SFP method tackles the nonlinearity in some direct way. The authors [23] also proposed primal-dual fixed point method which treat the nonlinearity in an indirect way. The main idea is to reduce high-order derivatives in (5) by introducing suitable intermediate variables

$$\nu_1 = -\kappa(u_1) = -\nabla \cdot \frac{\nabla u_1}{|\nabla u_1|_\beta}$$

and

$$\nu_2 = -\kappa(u_2) = -\nabla \cdot \frac{\nabla u_2}{|\nabla u_2|_\beta},$$

the corresponding equivalent system of EL equation (5) is given by

$$\begin{cases} -\nabla \cdot \frac{\nabla u_1}{|\nabla u_1|_\beta} - \nu_1 = 0 \\ -\nabla \cdot \frac{\nabla u_2}{|\nabla u_2|_\beta} - \nu_2 = 0 \\ f_1(\boldsymbol{u}) - \alpha \nabla \cdot \left(\frac{\nabla \nu_1}{|\nabla u_1|_\beta} + \frac{\nabla u_1 \cdot (-\nabla \nu_1)}{|\nabla u_1|_\beta^3} \nabla u_1\right) = 0 \\ f_2(\boldsymbol{u}) - \alpha \nabla \cdot \left(\frac{\nabla \nu_2}{|\nabla u_2|_\beta} + \frac{\nabla u_2 \cdot (-\nabla \nu_2)}{|\nabla u_2|_\beta^3} \nabla u_2\right) = 0 \end{cases} \tag{7}$$

with the boundary conditions transferred into $\nabla u_l = 0$ and $\nabla \nu_l = 0$ for $l = 1, 2$. They adopted pointwise collective Gauss-Seidel (PCGS) relaxation method to solve (7), we name this method as PDFP-1. To be more efficient, they introduced a relaxation parameter $\omega \in (0, 1)$ and iterate the $\omega-$PCGS steps, we name this method as PDFP-2. The PDFP method has been proven to be very efficient as a smoother for a nonlinear multi-grid by local Fourier analysis providing that the smoothing parameter is large enough (for example: $\beta \geq 5 \times 10^{-3}$).

As a matter of fact, the smoothing parameter β is smaller, and the corresponding nonlinearity is stronger, thus the convergence of many numerical methods can be slowed down. Small β does offer better residual, so we want to develop a new algorithm that converges even for very small β.

The rest of the paper is organized as follows. Section 2 proposes an efficient numerical scheme which doesn't impose a strong assumption on smoothing parameter β to solve (4). Section 3 illustrates the experimental results from syntectic and real images. Finally, conclusions and future work are summarized in Section 4.

2 A new numerical method for mean curvature-based registration model (4)

Over the past decades, there are 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 EL equations in the continuous domain and then solve its discrete forms on the corresponding discrete domain by appropriate methods, see [23, 5, 7, 8, 13, 9, 12, 16]. 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. [11, 10, 28, 27, 26]. In this paper, we prefer the second method. Although our work is related to previous work [11], they are totally different on their regularizer techniques and equations. Elastic regularizer with first order derivative was considered in [11], and it is convex. Mean curvature regularizer with high-order derivative is considered in this paper, and it is non-convex. If we use directly the scheme proposed in [11], it is very difficult to solve efficiently for (4). However, motivated by the idea of [11], we can change high-order regularizer $\mathcal{R}^{\mathrm{CCB}}(\boldsymbol{u})$ into convex by introducing a lagging into the denominator of $\mathcal{R}^{\mathrm{CCB}}(\boldsymbol{u})$ by using a previous and known iterate value, then solve the discrete energy functional using optimization methods. Next we shall first briefly introduce the discretization we use and then specifically describe the details of numerical algorithms.

2.1 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 $n_1 \times n_2$ cells has width $h_i = 1/n_i, i = 1, 2$. Let the discrete domain be denoted by

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

2.1.1 Discretizing displacement field $\boldsymbol{u}$ and the mean curvature-based regularizer $\mathcal{R}^{\mathrm{CCB}}(\boldsymbol{u})$

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 mean curvature regularizers $\mathcal{R}^{\mathrm{CCB}}(\boldsymbol{u})$ 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

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

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

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

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:

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

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

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

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

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 $N = n_1 n_2$. Furthermore, the kth component of the vectorized discrete mesh function $\boldsymbol{u}_l^h$can be denoted by $(\boldsymbol{u}_l^h)_k$, here $k = (j-1)\times n_1 + i$, $i = 1, 2, \cdots, n_1$; $j = 1, 2, \cdots, n_2$. The discrete gradient $(\nabla^h u_l^h)_{i,j}$ can also be represented by the product of the matrix $A_k^\top \in \mathbb{R}^{2\times N}$ 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

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

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

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

Let

$$\mathcal{B}[\boldsymbol{u}] = (\nabla \cdot \frac{\nabla u_1}{|\nabla u_1|_\beta})^2 + (\nabla \cdot \frac{\nabla u_2}{|\nabla u_2|_\beta})^2 \quad , \tag{8}$$

and

$$D = \begin{bmatrix} \frac{B}{|B\boldsymbol{u}_1^h|_\beta} & 0 \\ 0 & \frac{B}{|B\boldsymbol{u}_2^h|_\beta} \end{bmatrix} \quad .$$

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

$$\begin{aligned}
\mathbb{B}^h[\boldsymbol{U}^h] &= |\frac{-B^\top B\boldsymbol{u}_1^h}{|B\boldsymbol{u}_1^h|_\beta}|^2 + |\frac{-B^\top B\boldsymbol{u}_2^h}{|B\boldsymbol{u}_2^h|_\beta}|^2 \\
&= \frac{(\boldsymbol{u}_1^h)^\top B^\top B B^\top B\boldsymbol{u}_1^h}{|B\boldsymbol{u}_1^h|_\beta^2} + \frac{(\boldsymbol{u}_2^h)^\top B^\top B B^\top B\boldsymbol{u}_2^h}{|B\boldsymbol{u}_2^h|_\beta^2} \\
&= ((\boldsymbol{u}_1^h)^\top, (\boldsymbol{u}_2^h)^\top) \begin{bmatrix} \frac{B^\top B B^\top B}{|B\boldsymbol{u}_1^h|_\beta^2} & 0 \\ 0 & \frac{B^\top B B^\top B}{|B\boldsymbol{u}_2^h|_\beta^2} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1^h \\ \boldsymbol{u}_2^h \end{bmatrix} \\
&= (\boldsymbol{U}^h)^\top (\begin{bmatrix} \frac{B}{|B\boldsymbol{u}_1^h|_\beta} & 0 \\ 0 & \frac{B}{|B\boldsymbol{u}_2^h|_\beta} \end{bmatrix}^\top \begin{bmatrix} \frac{B}{|B\boldsymbol{u}_1^h|_\beta} & 0 \\ 0 & \frac{B}{|B\boldsymbol{u}_2^h|_\beta} \end{bmatrix})(\begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix}^\top \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix})\boldsymbol{U}^h \\
&= (\boldsymbol{U}^h)^\top D^\top D C^\top C \boldsymbol{U}^h.
\end{aligned}$$

Thus by a midpoint quadrature rule, the mean curvature regularizer $\mathcal{R}(\boldsymbol{u}) = \frac{1}{2}\int_\Omega \mathcal{B}[\boldsymbol{u}] d\boldsymbol{x}$ is descretized as

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

where $h_d = h_1 h_2$.

2.1.2 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 [36]. 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

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

and discrete transformed template image

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

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

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

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.

2.1.3 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:

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

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

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

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

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

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 ([5],[40]), we approximate $d^2\mathcal{D}^h(\boldsymbol{U}^h)$ by the following form

$$d^2\mathcal{D}^h(\boldsymbol{U}^h) = h_1 h_2 (\vec{T}_{\boldsymbol{U}^h})^\top \cdot \vec{T}_{\boldsymbol{U}^h} \quad . \tag{12}$$

2.2 Solving the discrete optimization problem

The discretized joint energy functional (4) reads as follows:

$$\min_{\boldsymbol{U}^h} \{\mathcal{J}_\alpha(\boldsymbol{U}^h) = \mathcal{D}^h(\boldsymbol{U}^h) + \alpha \cdot \mathcal{R}^h(\boldsymbol{U}^h)\}. \tag{13}$$

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 mean curvature regularizer $\mathcal{R}^h(\boldsymbol{U}^h)$. The lagged quantity in (13) uses a previous and known iterate $\boldsymbol{U}^{h^{(k)}} = (\boldsymbol{u}_1^{h^{(k)}}, \boldsymbol{u}_2^{h^{(k)}})^\top$. We note that the lagging method by 'frozen coefficients' is well known for variational approaches related to total variation (TV) operator (see e.g. [37, 35, 22, 19]). Thus we obtain the following form

$$\min_{\boldsymbol{U}^h} \{\mathcal{J}_\alpha(\boldsymbol{U}^h) = \mathcal{D}^h(\boldsymbol{U}^h) + \frac{1}{2}\alpha \cdot h_d \cdot (\boldsymbol{U}^h)^\top (D^{(k)})^\top D^{(k)} C^\top C \boldsymbol{U}^h\} \quad , \tag{14}$$

where

$$D^{(k)} = \begin{bmatrix} \frac{B}{|B\boldsymbol{u}_1^{h^{(k)}}|_\beta} & 0 \\ 0 & \frac{B}{|B\boldsymbol{u}_2^{h^{(k)}}|_\beta} \end{bmatrix} \quad .$$

To solve the above problem (14) 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

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

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 $(D^{(k)})^\top D^{(k)} C^\top C$ are both positive semi-define, we know that $\boldsymbol{H}$ is also positive semi-definite. Hence, $\hat{\mathcal{J}}_\alpha$ is convex. see [30] 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

$$d\mathcal{J}_\alpha(\boldsymbol{U}^{h^{(k)}}) = d\mathcal{D}^h(\boldsymbol{U}^{h^{(k)}}) + \alpha \cdot h_d \cdot (\boldsymbol{U}^{h^{(k)}})^\top (D^{(k)})^\top D^{(k)} C^\top C \tag{15}$$

and

$$\boldsymbol{H} = d^2\mathcal{D}^h(\boldsymbol{U}^{h^{(k)}}) + \alpha \cdot h_d \cdot (D^{(k)})^\top D^{(k)} C^\top C. \tag{16}$$

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

$$\boldsymbol{H}\delta_{\boldsymbol{U}^h} = -d\mathcal{J}_\alpha(\boldsymbol{U}^{h^{(k)}}). \tag{17}$$

Usually, $\boldsymbol{H}$ is positive definite, thus we can use a preconditioned conjugate gradient method to solve the quasi-Newtons equation (17), on the preconditioning techniques, we can refer to [31, 32, 33, 34]. 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 [30]. The procedure will be terminated when stopping rules are met. In this section we use following common stoping rules for the above Gauss-Newton scheme; see also [29, 24].

1. $\text{Stop}(1) = \text{abs}(\mathcal{J}_{\text{old}} - \mathcal{J}_{\text{c}}) \leq 10^{-3} * (1 + \text{abs}(\mathcal{J}_{\text{stop}}))$;
2. $\text{Stop}(2) = \text{norm}(\boldsymbol{u}_{\text{c}} - \boldsymbol{u}_{\text{old}}) \leq 10^{-2} * (1 + \boldsymbol{u}0)$;
3. $\text{Stop}(3) = \text{norm}(d\mathcal{J}_c) \leq 10^{-2} * (1 + \text{abc}(\mathcal{J}_{\text{stop}}))$;
4. $\text{Stop}(4) = \text{norm}(d\mathcal{J}_c) \leq \text{eps}$;
5. $\text{Stop}(5) = (\text{iter} \geq \text{maxIter})$;

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

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 (14), (15) and (16), respectively;
while true **do**
  Update iteration count: iter←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 (14), (15) and (16), respectively
end

value. eps denotes the machine precision and maxIter is an a priori chosen number. The numerical scheme is summarized in Algorithm 1.

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 (14) and (15), 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 (14);
  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 (14). First, we provide an initial value by multilevel affine linear preregistration on the coarsest level, then solve (14) 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 (14) 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 (13) 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 3. Where bi-linear interpolation operator is denoted by I_H^h.

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

```
Maxlevel← ceil(log2(min(n_1, n_2))),      % 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;
    Providing initial guess u_0 by using multilevel affine linear preregistration then
    end
    if l = Minlevel;
    u0 ← u_0;
    else;
    u0 ← I_H^h(u) then
    end
    u ← GNIRArmijo(α, u0) ;
end
```

3 Numerical experiments

In this section, our primary aim is to illustrate the effectiveness of our new Algorithm 3 and show it is more robust among the existing implementations for the mean curvature-based image registration model. From the experiment results in [23], we can see that primary-dual fixed point (PDFP-2) method as a smoother is much better than other fixed point methods for nonlinear multigrid. For ease of comparison, we shall denote by NMG for nonlinear multigrid method with smoother PDFP-2 and by A3 for our proposed new Algorithm 3.

To be fair on the measure of the quality of the registered images, the relative reduction of the dissimilarity rel.SSD proposed by Chumchob-Chen-Brito [23] is used, and it is defined as follows

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

Where $\boldsymbol{u}$ is the current optimal value and $\mathcal{D}_{\mathrm{stop}}$ is the value of $\mathcal{D}(\boldsymbol{u})$ at $\boldsymbol{u} = 0$.

We select three representative data sets shown respectively in Figure 1 (Two non-smooth registration problems and a smooth registration problem to be denoted respectively as Example 1 , Example 2 and Example 3) for the experiments.

In the first experiment, we first focus on the capabilities of our new Algorithm 3 for registration of the three test Examples $1-3$ in resolution 32×32, 512×512. The registered images by our new Algorithm 3 are shown in Figure 1 (right column). For three tests, smoothing parameter β is taken 10^{-6}. Clearly, registered images from our new Algorithm 3 is very satisfying. Below we mainly focus on the further gains from our new Algorithm 3.

3.1 Comparisons with previous methods for model (4)

No fast solvers existed for image registration model (4) before the work of [23], i.e. nonlinear multigrid method with smoother PDFP-2 which is denoted by NMG. To further show that our new Algorithm 3 is efficient and robust, next we compare it with NMG. Three specific comparisons are implemented with respect to parameters α, β in the model and the mesh parameter h. As the same model is solved, it's natural that we use the same parameters for the same example for fair comparison.

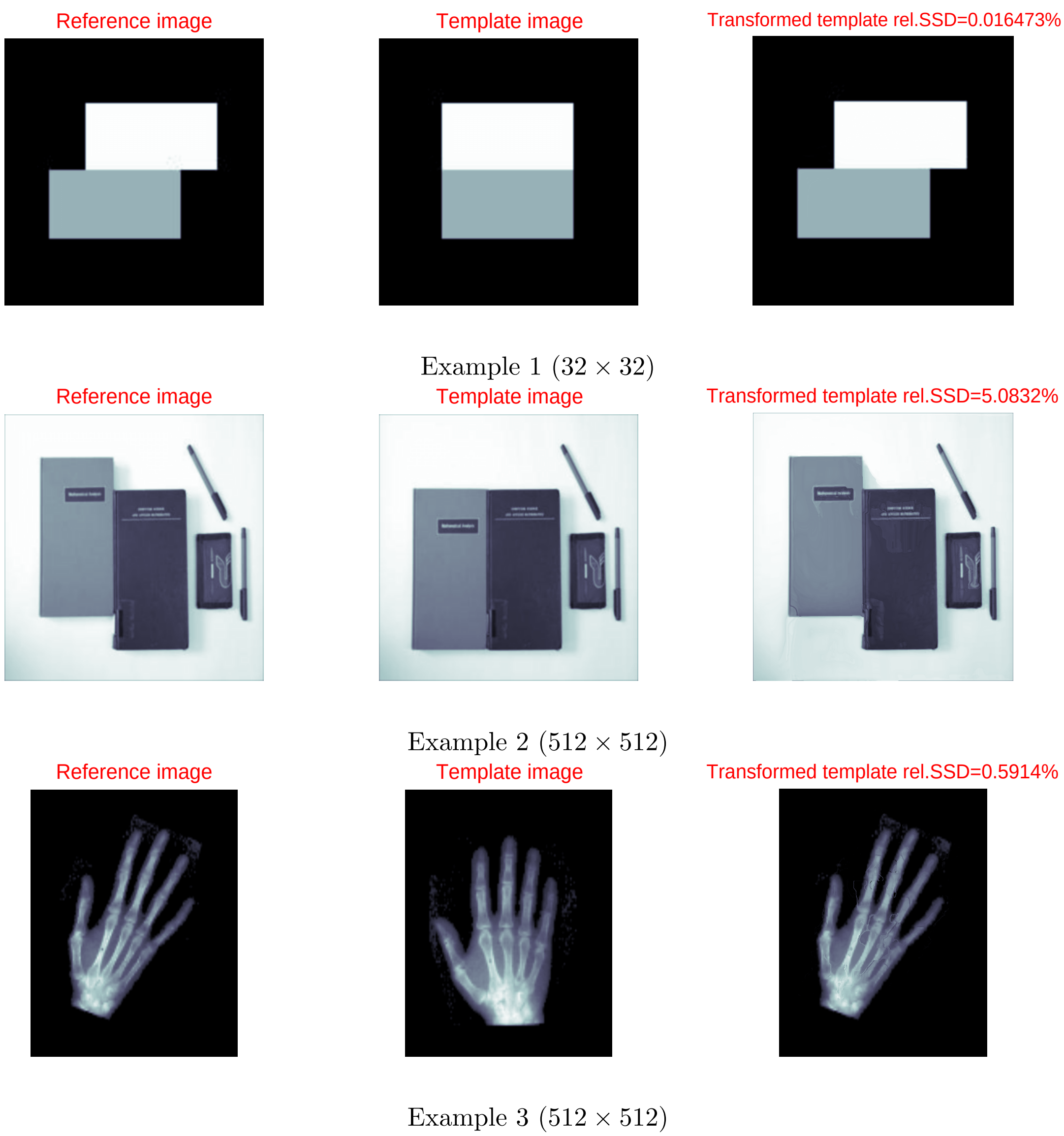

Figure 1: Registration results for three representative data sets(Example $1-2$ (non-smooth registration problems) and Example 3 (smooth registration problem)) using our new Algorithm 3 . Left column: reference image R, center column: template image T. right column: the deformed template image $T(\boldsymbol{u})$ obtained from Algorithm 3.

3.1.1 *h*-independent convergence test

We shall resolve the same Example $2-3$ as above using an increasing sequence of resolutions (or a decreasing mesh parameter) and show the results from A3 and NMG in Table 1. The required parameters in the experiments are taken: $\alpha = 0.75 \times 10^{-4}$, $\beta = 5 \times 10^{-3}$ for Example 2 and $\alpha = 10^{-4}$, $\beta = 1$ for Example 3. In Table 1, we compare the registration quality via *rel*.SSD and efficiency via CPU. The numerical experiments prove that two registration Algorithms are both converge and they are also accurate because the dissimilarities between the reference and transformed images have been reduced more than 92% for Example 2 and 96% for Example 3. For overall performance the experimental results suggest that our new Algorithm 3 is more efficient and would be preferred for practical applications because this method can find a highly accurate solution in a relatively short time and produce excellent image registration results in term of image quality.

		A3		NMG	
Example	h	*rel*.SSD	CPU(S)	*rel*.SSD	CPU(S)
2	1/128	4.49%	7	6.98%	213
	1/256	5.03%	15	7.01%	240
	1/512	5.08%	47	7.12%	267
Example	h	*rel*.SSD	CPU(S)	*rel*.SSD	CPU(S)
3	1/128	0.72%	5	2.58%	133
	1/256	0.61%	9	3.86%	160
	1/512	0.59%	26	3.79%	160

Table 1: Registration results of A3 and NMG for processing Examples 2 and 3 shown respectively in Figure 1. A3 means our new Algorithm 3; NMG means nonlinear multigrid with smoother PDFP-2 [23]. CPU means the total runtimes including Image output and pre-registration. For Example 2, parameters $\alpha = 0.75 \times 10^{-4}$, $\beta = 5 \times 10^{-3}$; for Example 3, $\alpha = 10^{-4}$, $\beta = 1$.

3.1.2 α-dependence test

Here we compare the sensitivity of A3 and NMG with respect to varying the regularization parameter α. To this end, two methods were tested on Example 3 (see Figure 1 last row) with the results shown in Table 2. Here the following parameters are used: $\beta = 1$ and $h = 1/512$ for all experiments and α is varied from 10^{-4} to 10^{-1}. In table 2, we can see a clear process of the changes of *rel*.SSD using our new Algorithm 3 and nonlinear multigrid with smoother PDFP-2. Although both of them improve the registration quality as α decrease, we can see that the performance of our new Algorithm 3 is more consistently behaved.

α	method	*rel*.SSD
10^{-4}	A3	0.59%
	NMG	3.79%
10^{-3}	A3	0.60%
	NMG	15.28%
10^{-2}	A3	0.64%
	NMG	30.19%
10^{-1}	A3	0.71%
	NMG	47.09%

Table 2: α-sensitivity comparison using Example 3 (see Figure 1 last row) with varying α and other fixed parameters.

3.1.3 β-dependence test

As is well known, the quantities of results and the performances of some numerical schemes in solving the nonlinear system related to the total variation (TV) regularization technique are affected significantly by the value of β. Theoretically β should be selected to be as small as possible, thus the solution of (4) converges to the solution of original problem (1), more details see [18]. Here we analyze how β affects the performance of our new Algorithm 3 (A3) and nonlinear multigrid method with smoother PDFP-2 (NMG). To this end, two methods were tested on Example 2 (see Figure 1 middle row) with the results shown in Table 3, where $*$ means no convergence. Here the following parameters are taken: $\alpha = 0.75 \times 10^{-4}$, and $h = 1/512$ for all experiments and β is varied from 10^{-16} to 1. For this example, on one hand we can see our Algorithm is still convergent when β is very small; On the other hand, we can also observe the quality of registered image by Algorithm 3 is not sensitive as β reduces.

β	method	rel.SSD
10^{-16}	A3	4.68%
	NMG	$*$
10^{-12}	A3	4.68%
	NMG	$*$
10^{-6}	A3	4.75%
	NMG	$*$
10^{-4}	A3	5.12%
	NMG	$*$
5×10^{-3}	A3	5.14%
	NMG	7.01%
10^{-2}	A3	5.25%
	NMG	8.93%
10^{-1}	A3	5.30%
	NMG	23.24%
10^{-0}	A3	5.32%
	NMG	45.57%

Table 3: β-sensitivity comparison using Example 2 (see Figure 1 middle row) with varying β and other fixed parameters.

4 Conclusions

The mean curvature-based image registration model is known to be effective to deliver better registration results for both smooth and non-smooth deformation fields. However, it is difficult to solve efficiently this model. Although Chumchob-Chen-Brito [23] developed a convergent multigrid method using primary-dual fixed-point method as a smoother to solve this model providing that the smoothing parameter β is larger enough (e.g.$\geq 5 \times 10^{-3}$). We are interested in obtaining a numerical algorithm that converges even for very small β. In this paper, we adopt discretize-optimize method, follow an idea of relaxed fixed point and combine with Gauss-Newton scheme with Armijo's Linear Search for solving the discretized mean curvature 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. In our future work, we plan to use homotopy method which has become a useful tool for solving nonlinear problems to solve discrete registration model (13).

References

[1] J. Maintz and M. Viergever, *A survey of medical image registration* ,Medical Image Analysis, 2 (1), pp. 1–36, 1998.

[2] 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.

[3] 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.

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

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

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

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

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

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

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

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

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

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

[14] 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.

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

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

[17] T.F.Chan, G.H.Golub, and P.MULET, *A nonlinear primal-dual method for total variation-based image restoration*, SIAM J.Sci.Comput., 20(1999), pp. 1964-1997.

[18] R.Acar and C.R.Vogel, and C.Fatemi, *Analysis of bounded variation penalty methods for ill-posed problems*, Inverse Problems, Vol. 10, pp. 1217-1229, 1994.

[19] 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.

[20] T.F. Chan and J.H. Shen, *Image Processing and Analysis - Variational, PDE, Wavelet, and Stochastic Methods*, SIAM Publications, Philadelphia, USA, 2005.

[21] K. Chen, *Matrix Preconditioning Techniques and Applications*,Cambridge University Press, UK, 2005.

[22] 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.

[23] 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.

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

[25] G. Golub, M. Heath, and G. Wahba, *Generalized cross-validation as a method for choosing a good ridge parameter*, Technometrics 21 , pp. 215–223, 1979.

[26] 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.

[27] 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.

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

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

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

[31] Z. Z. Bai, Y. M. Huang, M. K. Ng, *On Preconditioned Iterative Methods for Burgers Equations*, Siam Journal on Scientific Computing, 29(1):415-439, 2007.

[32] Z. Z. Bai, Y. M. Huang, M. K. Ng, *On preconditioned iterative methods for certain time-dependent partial differential equations*, Siam Journal on Numerical Analysis, 47(2), 1019-1037., 2009.

[33] Z. Z. Bai and M. K. Ng, *On Inexact Preconditioners for Nonsymmetric Matrices*, Siam Journal on Scientific Computing, 26(5), 1710-1724, 2005.

[34] Z. Z. Bai, M. K. Ng and Z. Q. Wang, *Constraint Preconditioners for Symmetric Indefinite Matrices*, Siam Journal on Matrix Analysis & Applications, 31(2), 410-433, 2010.

[35] 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.

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

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

[38] C.R.Vogel, *Computational Methods for Inverse Problems*, SIAM Publications, Philadelphia, USA, 2002.

[39] G. Wahba, *Spline models for observational data*, SIAM, Philadelphia, 1990.

[40] H.Köstler, K. Ruhnau, and R.Wienands, *Multigrid solution of the optical flow system using a combined diffusion- and curvature-based regularizer*, Numer. Linear Algebra Appl., 15 , pp. 201–218, 2008.