

(c) Springer 2018 *Journal of Mathematical Imaging and Vision*

1 **A NOVEL DIFFEOMORPHIC MODEL FOR IMAGE REGISTRATION**
2 **AND ITS ALGORITHM***

3 DAOPING ZHANG[†] AND KE CHEN^{†‡}

4 **Abstract.** In this work, we investigate image registration by mapping one image to another in
5 a variational framework and focus on both model robustness and solver efficiency. We first propose
6 a new variational model with a special regularizer, based on the quasi-conformal theory, which can
7 guarantee that the registration map is diffeomorphic. It is well known that when the deformation is
8 large, many variational models including the popular diffusion model cannot ensure diffeomorphism.
9 One common observation is that the fidelity error appears small while the obtained transform is
10 incorrect by way of mesh folding. However direct reformulation from the Beltrami framework does
11 not lead to effective models; our new regularizer is constructed based on this framework and added
12 to the diffusion model to get a new model, which can achieve diffeomorphism. However the idea is
13 applicable to a wide class of models. We then propose an iterative method to solve the resulting
14 nonlinear optimization problem and prove the convergence of the method. Numerical experiments
15 can demonstrate that the new model can not only get a diffeomorphic registration even when the
16 deformation is large, but also possess the accuracy in comparing with the currently best models.

17 **Key words.** Image registration, diffeomorphic, Beltrami coefficient, optimization, Gauss-
18 Newton scheme.

19 **AMS subject classifications.** 65D, 65M, 65K, 68U, 68W

20 **1. Introduction.** Image registration is to find a transformation to map the cor-
21 responding image data, which are taken at different times, from different sensors, or
22 from different viewpoints, for the purpose of telling the difference or merging informa-
23 tion. Nowadays, image registration is widely used in many areas, such as computer
24 vision, biological imaging, remote sensing and medical imaging [6, 21, 26, 32, 36, 38,
25 40, 47, 57].

26 In reality, according to the specific application, image registration can be classified
27 into two categories: mono-modal registration and multi-modal registration. For multi-
28 modal registration, finding a suitable distance measure is the most essential step [22,
29 35, 36, 47, 57]. The idea of this paper will be applicable to multi-modal registration
30 framework, but we focus on the mono-modal registration in this work.

31 In dealing with the mono-modal registration, there are many choices of a data
32 fidelity term [33] and a common approach for computing this transformation is to use
33 the sum of squared differences (SSD) to measure the difference between the reference
34 image R and the deformed template image T [11]. However, minimization of SSD
35 alone in image registration is an ill-posed problem in the sense of Hadamard since
36 it may have many solutions. In order to overcome this difficulty, regularization is
37 indispensable [38, 52]. However, the choice of the regularization term, which needs
38 some prior information about physical properties and helps to avoid the local minima,
39 depends on the specific application.

40 All registration models are nonlinear but they can be classified into two main
41 categories according to the way deformation mapping is represented: linear registra-
42 tion and nonlinear registration. In linear registration, the deformation model is linear

*This work was partly funded by UK EPSRC grants EP/K036939/1 and EP/N014499/1. DPZ also acknowledges the studentship support of China Scholarship Council.

[†]EPSRC Liverpool Centre for Mathematics in Healthcare, Centre for Mathematical Imaging Techniques and Department of Mathematical Sciences, The University of Liverpool, Peach Street, Liverpool L69 7ZL, United Kingdom

[‡]Correspondence author. Web <http://www.liv.ac.uk/~cmchenke> Email: k.chen@liv.ac.uk

43 and global, including rotation, translation, shearing and scaling [11, 38]. Although
 44 the computation speed of a linear model is fast since it contains few variables, it is
 45 commonly used as the pre-registration for starting a more sophisticated model. This
 46 is mainly because linear models can not accommodate the local details (differences).
 47 In contrast, nonlinear registration models inspired by physical processes of trans-
 48 formations [47] such as the elastic model [5], fluid model [9], diffusion model [16],
 49 TV (total variation) model [19], MTV (modified TV) model [12], linear curvature
 50 model [17, 18], mean curvature model [14], Gaussian curvature model [27] and total
 51 fractional-order variation model [56] are proposed to account for localised variation
 52 in details, by allowing many degrees of freedom. The particular free-form deforma-
 53 tion models based on B-splines lying between the above two types possess simplicity,
 54 smoothness, efficiency and ability to describe local deformation with few degrees of
 55 freedom [44, 45, 47]. For relatively small deformation, all models can be effective,
 56 but for large deformation, not all models are effective and in particular few models
 57 can guarantee a one-to-one mapping unless one fine tunes the coupling parameters
 58 to reduce the deformation magnitude allowed (since the mapping quality is perfect if
 59 deformation is zero) which in turn loses the ability of modeling large deformation.

60 Over the last decade, more and more researchers have focused on diffeomorphic
 61 image registration where folding measured by the local invertibility quantity $\det(J_{\mathbf{y}})$
 62 is reduced or avoided. Here, \mathbf{y} denotes the transformation in the registration model
 63 and $\det(J_{\mathbf{y}})$ is the Jacobian determinant of \mathbf{y} . Under desired assumptions, obtaining
 64 a one-to-one mapping is a natural choice as reviewed in [47].

65 In 2004, Haber and Modersitzki [23] proposed an image registration model impos-
 66 ing volume preserving constraints, by ensuring $\det(J_{\mathbf{y}})$ is close to 1. Although
 67 volume preservation is very important in some applications where some underlying
 68 (e.g. anatomical) structure is known to be incompressible [47], it is not required or
 69 reasonable in others. In a later work, the same authors [25] relaxed the constraint to
 70 allow $\det(J_{\mathbf{y}})$ to lie in a specific interval. Yanovsky et al. [55] applied the symmetric
 71 Kullback-Leibler distance to quantify $\det(J_{\mathbf{y}})$ to achieve a diffeomorphic mapping.
 72 Burger et al. [7] designed a volume penalty term that ensured that shrinkage and
 73 growth had the same cost in their variational functional. The constrained hierar-
 74 chical parametric approach [41] ensures that the mapping is globally one-to-one and
 75 thus preserves topology in the deformed image. Sdika [46] introduced a regularizer to
 76 penalize the non-invertible transformation. In [51], Vercauteren et al. proposed an ef-
 77 ficient non-parametric diffeomorphic image registration algorithm based on Thirion's
 78 demons algorithm [49]. In addition, a framework called Large Deformation Diffeomor-
 79 phic Metric Mapping (LDDMM) can generate the diffeomorphic transformation for
 80 image registration [37, 3, 15, 50]. An entirely different framework proposed by Lam
 81 and Lui [30] obtains diffeomorphic registrations by constraining Beltrami coefficients
 82 of a quasi-conformal map $\mathbf{f} = y_1(\mathbf{x}) + iy_2(\mathbf{x})$, instead of controlling the map $\mathbf{y}(\mathbf{x})$
 83 directly.

84 In this paper, we aim to reformulate the Lam and Lui Beltrami measure as a
 85 direct regularizer for controlling $\det(J_{\mathbf{y}})$ and to assess the effectiveness of the resulting
 86 variational models; though the idea applies to any commonly used models, we apply
 87 it to the diffusion model as one simple example. Our contributions are two-fold:

- 88 • We propose a new Beltrami coefficient based regularizer that is explicitly
 89 expressed in terms of $\det(J_{\mathbf{y}})$. This establishes a link between the Beltrami
 90 coefficient of the transformation and the quantity $\det(J_{\mathbf{y}})$.
- 91 • An effective, iterative scheme is presented and numerical experimental results
 92 show that the new registration model has a good performance and produces

93 a diffeomorphic mapping while remaining competitive to the state-of-the-art
 94 models from non-Beltrami frameworks.

95 We remark that several interesting works that are concerned with reversible transfor-
 96 mations (such as [8, 54]) may also benefit from this study.

97 The rest of the paper is organized as follows. Section 2 briefly reviews the basic
 98 mathematical formulation of image registration modeling, several typical regulariza-
 99 tion terms and how to get a diffeomorphic transformation for image registration. In
 100 Section 3, we propose a new regularizer and a new registration model. The effective
 101 discretization and numerical scheme are discussed in Section 4. Numerical experiment
 102 results are shown in Section 5, and finally a summary is concluded in Section 6.

103 **2. Preliminaries, Regularization and Diffeomorphic Transformation.** In
 104 general, image registration aims to compare, in space \mathbb{R}^d , two or more images or
 105 image sequences in a video. In this work, we consider the case of a pair of images
 106 $T, R : \Omega \subset \mathbb{R}^d \rightarrow \mathbb{R}$ and $d = 2$. Here by convention, R is the Reference image and T
 107 is the (moving) Template image.

108 The aim of image registration is to find a transformation $\mathbf{y}(\mathbf{x})$ such that

109
$$T \circ \mathbf{y}(\mathbf{x}) = T(\mathbf{y}(\mathbf{x})) \approx R,$$

110 where $\mathbf{x} = (x_1, x_2)$ and $\mathbf{y}(\mathbf{x}) = (y_1(\mathbf{x}), y_2(\mathbf{x}))$. That is, the transformation $\mathbf{y}(\mathbf{x})$ moves
 111 T to match R . If we define $\mathbf{y}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x})$, then $\mathbf{u}(\mathbf{x}) = (u_1(\mathbf{x}), u_2(\mathbf{x}))$ indicates
 112 how much T moves i.e. $\mathbf{u}(\mathbf{x})$ is the displacement. Thus, the determination of the
 113 transformation $\mathbf{y}(\mathbf{x})$ is equivalent to the determination of the displacement field $\mathbf{u}(\mathbf{x})$.

114 **2.1. Data fidelity.** One way to ensure that $T(\mathbf{y})$ can approximate R is to min-
 115 imize the difference $T(\mathbf{y}) - R$. A commonly used difference measure is the sum of
 116 squared differences (SSD) defined by

117 (1)
$$\mathcal{D}[\mathbf{y}] = \frac{1}{2} \int_{\Omega} (T(\mathbf{y}) - R)^2 d\mathbf{x} = \frac{1}{2} \|T(\mathbf{y}) - R\|^2 = \frac{1}{2} \|T(\mathbf{x} + \mathbf{u}) - R\|^2 = \mathcal{D}[\mathbf{u}]$$

118 where $\|\cdot\|^2$ denotes the squared L_2 -norm. Of course, there are some other typical
 119 distance measures, including normalized cross correlation [38], mutual information
 120 [35, 38], normalized gradient fields [24, 39] and mass-preserving measure [7].

121 **2.2. Regularization.** Minimizing any of the above mentioned measures is inef-
 122 ficient to obtain a unique transformation \mathbf{y} for image registration, because $\min \mathcal{D}[\mathbf{y}]$
 123 is ill-posed [38, 39]. In order to overcome this problem, regularization is necessary.
 124 Combining distance measure and regularization gives the variational model for image
 125 registration:

126 (2)
$$\min_{\mathbf{u}} J(\mathbf{u}) = \mathcal{D}[\mathbf{u}] + \alpha S[\mathbf{u}],$$

127 where $\mathcal{D}[\mathbf{u}]$ is the distance measure from (1), $S[\mathbf{u}]$ is the regularizer to be discussed
 128 and α is a positive parameter to balance these two terms.

129 There exist many regularizers and we can classify them into three categories:

- 130 • First order regularizers involving $|\nabla \mathbf{u}|$ or $|\nabla \cdot \mathbf{u}|$. The diffusion regularizer
 131 [16] and the TV regularizer [19] are well-known first order regularizers. The
 132 former one aims to control smoothness of the displacement and the latter one
 133 can preserve the discontinuity.
- 134 • Fractional order regularizer $\nabla^\alpha \mathbf{u}$ with $\alpha \in (1, 2)$. In [56], a fractional or-
 135 der regularizer is used for image registration. Because the fractional order

136 regularizer is a global regularizer, its implementation must explore the struc-
 137 tured Toeplitz matrices. This regularizer can not only produce accurate and
 138 smooth solutions but also allow for a large rigid alignment [56].

- 139 • Second order regularizers involving $\nabla^2 \mathbf{u}$ or $\nabla \cdot (\nabla \mathbf{u} / |\nabla \mathbf{u}|)$. These include
 140 the linear curvature regularizer [17, 18], mean curvature regularizer [14] and
 141 Gaussian curvature regularizer [27].

142 The first two categories of models require an affine linear transformation in an initial
 143 pre-registration step while the latter category does not need a linear transformation
 144 in pre-registration.

145 Differing from the above three categories, an important class of fluid like models
 146 based on partial differential equations were developed to capture large deformations.
 147 Christensen et al. [10] proposed an effective viscous fluid model characterized by a
 148 spatial smoothing of the velocity field. For the viscous fluid model, the deformation
 149 is governed by the Navier-Stokes equation:

$$150 \quad (3) \quad \eta \nabla^2 \mathbf{v} + (\eta + \lambda) \nabla (\nabla \cdot \mathbf{v}) + \mathbf{F} = 0, \quad \mathbf{v} = \partial_t \mathbf{u} + \mathbf{v} \cdot \nabla \mathbf{u}.$$

151 Here, η and λ are the viscosity coefficients, the term $\nabla^2 \mathbf{v}$ constrains the velocity
 152 field to vary smoothly, the term $\nabla (\nabla \cdot \mathbf{v})$ allows structures in the template to change
 153 in mass and \mathbf{F} is the nonlinear deformation force field, which can be defined by
 154 $(T(\mathbf{x} + \mathbf{u}) - R) \nabla T$. The velocity field \mathbf{v} is initialized as $\mathbf{0}$ in implementation. In [10],
 155 the condition $|\det(J_{\mathbf{y}})| \geq 0.5$ is checked at each iteration and if not satisfied, restarting
 156 the numerical solver is initiated so that a diffeomorphic transform is obtained; see also
 157 [38]. Further in [55], the model is enhanced by incorporating a volume preservation
 158 idea relating to minimizing $|\det(J_{\mathbf{y}}) - 1|$ again to ensure diffeomorphism without
 159 restarting.

160 Next, we review the **Diffusion** model [16]

$$161 \quad (4) \quad \min_{\mathbf{u}} J(\mathbf{u}) = \mathcal{D}[\mathbf{u}] + \alpha S[\mathbf{u}] = \frac{1}{2} \int_{\Omega} (T(\mathbf{x} + \mathbf{u}) - R)^2 d\mathbf{x} + \frac{\alpha}{2} \int_{\Omega} \sum_{\ell=1}^2 |\nabla u_{\ell}|^2 d\mathbf{x}.$$

162 It leads to the Euler-Lagrange equation:

$$163 \quad (T(\mathbf{x} + \mathbf{u}) - R) \nabla_{\mathbf{u}} T(\mathbf{x} + \mathbf{u}) - \alpha \Delta \mathbf{u} = 0 \text{ i.e. } \begin{cases} (T(\mathbf{x} + \mathbf{u}) - R) \partial_{u_1} T(\mathbf{x} + \mathbf{u}) - \alpha \Delta u_1 = 0, \\ (T(\mathbf{x} + \mathbf{u}) - R) \partial_{u_2} T(\mathbf{x} + \mathbf{u}) - \alpha \Delta u_2 = 0, \end{cases}$$

164 subject to $\langle \nabla u_{\ell}, \mathbf{n} \rangle = 0$ on $\partial \Omega$ and $\ell = 1, 2$. Particularly, there exists a fast implemen-
 165 tation based on the so-called additive operator splitting (AOS) scheme [38, 53]. In
 166 [13], a fast solver was developed for this model.

167 However, as with other models reviewed in the three categories, the obtained
 168 solution \mathbf{u} or \mathbf{y} is mathematically correct but often incorrect physically. This is due
 169 to no guarantee of mesh non-folding which is measured by $\det(J_{\mathbf{y}}) > 0$ i.e. a positive
 170 determinant of the local Jacobian matrix $J_{\mathbf{y}}$ of the transform \mathbf{y} .

171 **2.3. Models of diffeomorphic transformation.** To achieve $\det(J_{\mathbf{y}}) > 0$, one
 172 can find several recent works that impose this constraint in some direct ways. We
 173 review a few of such models before we present our new constraint. In the form of (4),
 174 the idea is to choose $S_1[\cdot]$ in the following (note $\mathbf{y} = \mathbf{x} + \mathbf{u}$)

$$175 \quad (5) \quad \min_{\mathbf{u}} J(\mathbf{u}) = \mathcal{D}[\mathbf{u}] + \alpha S[\mathbf{u}] + \beta S_1[\mathbf{y}].$$

176 **Volume control.** In 2004, Haber and Modersitzki [23] used volume preserving
 177 constraint (area in 2D) for image registration, namely

178
$$\det(J_{\mathbf{y}}) = 1.$$

179 As a consequence, we can ensure that the transformation is diffeomorphic. However,
 180 volume preservation is not desirable when the anatomical structure is compressible in
 181 medical imaging.

182 **Slack constraint.** Improving on [25], the constraint $\det(J_{\mathbf{y}}) = 1$ is relaxed and
 183 a slack constraint is proposed

184
$$M_a \leq \det(J_{\mathbf{y}}) \leq M_b,$$

185 where a positive interval $[M_a, M_b]$ is provided by the user as prior information in the
 186 specific application e.g. $[M_a, M_b] = [0.1, 2]$.

187 **Unbiased transform.** In [55], according to the information theory, $\det(J_{\mathbf{y}})$ is
 188 controlled by the symmetric Kullback-Leibler distance

189
$$\int_{\Omega} |\det(J_{\mathbf{y}}) - 1| \log(|\det(J_{\mathbf{y}})|) dx.$$

190 It can help to get an unbiased diffeomorphic transformation. This idea was tested
 191 with the fluid regularizer (first order).

192 **Balance of shrinkage and growth.** Geometrically $\det(J_{\mathbf{y}}) = 1$ implies volume
 193 preservation. Similarly $\det(J_{\mathbf{y}}) < 1$ implies shrinkage while $\det(J_{\mathbf{y}}) > 1$ implies
 194 growth. A function that treats the cases of shrinkage and growth identically is $\phi(x) =$
 195 $((x - 1)^2/x)^2$ since $\phi(1/x) = \phi(x)$. A volume penalty

196 (6)
$$\int_{\Omega} \left(\frac{(\det(J_{\mathbf{y}}) - 1)^2}{\det(J_{\mathbf{y}})} \right)^2 dx$$

197 is used in the hyperelastic model [7], which ensures that shrinkage and growth have
 198 the same price.

199 **LDDMM Framework.** In LDDMM framework, the deformation is modeled by
 200 considering its velocity over time according to the transport equation. We can write
 201 its variational formulation as follows:

202
$$\begin{aligned} & \min_{\mathcal{T}, v} \mathcal{D}(\mathcal{T}(\cdot, 1), R) + \alpha \mathcal{S}(v) \\ & \text{s.t. } \partial_t \mathcal{T}(\mathbf{x}, t) + v(\mathbf{x}, t) \cdot \nabla \mathcal{T}(\mathbf{x}, t) = 0 \text{ and } \mathcal{T}(\mathbf{x}, 0) = T, \end{aligned}$$

203 where $v : \Omega \times [0, 1] \rightarrow \mathbb{R}^2$ is the velocity and $\mathcal{T} : \Omega \times [0, 1] \rightarrow \mathbb{R}$ is a series of images.
 204 For more details, please see [37, 3, 15, 47, 50]

205 **Beltrami indirect control.** In 2014, Lam and Lui [30] presented a novel ap-
 206 proach in a Beltrami framework to obtain diffeomorphic registrations with large defor-
 207 mations using landmark and intensity information via quasi-conformal maps. Before
 208 introducing this model, we first describe some basic theories about quasi-conformal
 209 map and Beltrami coefficient.

210 A complex map $z = x_1 + \mathbf{i}x_2 \mapsto f(z) = y_1(x_1, x_2) + \mathbf{i}y_2(x_1, x_2)$ from a domain
 211 in \mathbb{C} onto another domain is quasi-conformal if it has continuous partial derivatives
 212 and satisfies the following Beltrami equation:

213 (7)
$$\frac{\partial f}{\partial \bar{z}} = \mu(f) \frac{\partial f}{\partial z},$$

214 for some complex-valued Lebesgue measurable μ [4] satisfying $\|\mu\|_\infty < 1$. Here $\mu =$
 215 $\mu(\mathbf{y}) \equiv f_{\bar{z}}/f_z$ is called the Beltrami coefficient explicitly computable from \mathbf{y} since

$$216 \quad (8) \quad \begin{cases} f_z = \frac{\partial f}{\partial z} \equiv \frac{1}{2} \left(\frac{\partial f}{\partial x_1} - \mathbf{i} \frac{\partial f}{\partial x_2} \right) = \frac{(y_1)_{x_1} + (y_2)_{x_2}}{2} + \mathbf{i} \frac{(y_2)_{x_1} - (y_1)_{x_2}}{2}, \\ f_{\bar{z}} = \frac{\partial f}{\partial \bar{z}} \equiv \frac{1}{2} \left(\frac{\partial f}{\partial x_1} + \mathbf{i} \frac{\partial f}{\partial x_2} \right) = \frac{(y_1)_{x_1} - (y_2)_{x_2}}{2} + \mathbf{i} \frac{(y_2)_{x_1} + (y_1)_{x_2}}{2}, \end{cases}$$

217 where $(y_1)_{x_1} = \partial y_1 / \partial x_1$. Conversely $\mathbf{y} = \mathbf{y}^\mu$ can be computed for a given μ through
 218 solving $\mu(\mathbf{y}) = \mu$.

219 A quasi-conformal map is a homeomorphism (i.e. one-to-one) and its first-order
 220 approximation takes small circles to small ellipses of bounded eccentricity [20]. As a
 221 special case, $\mu = 0$ means that the map f is holomorphic and conformal, characterized
 222 by $f_{\bar{z}} = 0$ or y_1, y_2 satisfying the Cauchy-Riemann equations $(y_1)_{x_1} = (y_2)_{x_2}$, $(y_1)_{x_2} =$
 223 $-(y_2)_{x_1}$.

224 Thus in the context of image registration, enforcing $\|\mu\|_\infty < 1$ provides the con-
 225 trol for the transform f and ensures homeomorphism. The quasi-conformal hybrid
 226 registration model (QCHR) in [30] is

$$227 \quad (9) \quad \min_{\mathbf{y}} \int_{\Omega} |\nabla \mu|^2 + \alpha \int_{\Omega} |\mu|^p + \beta \int_{\Omega} (T(\mathbf{y}) - R)^2$$

228 subject to $\mathbf{y} = (y_1, y_2)$ satisfying

- 229 1). $\mu = \mu(\mathbf{y})$;
- 230 2). $\mathbf{y}(p_j) = q_j$ for $1 \leq j \leq m$ (Landmark constraints);
- 231 3). $\|\mu(\mathbf{y})\|_\infty < 1$ (bijectivity),

232 which indirectly controls $\det(J_{\mathbf{y}})$ via Beltrami coefficient, where $\mu(\mathbf{y})$ is the Beltrami
 233 coefficient of the transformation \mathbf{y} . The above model is solved by a penalty splitting
 234 method. It minimizes the following functional:

$$235 \quad (10) \quad \int_{\Omega} |\nabla \nu|^2 + \alpha \int_{\Omega} |\nu|^p + \sigma \int_{\Omega} |\nu - \mu|^2 + \beta \int_{\Omega} (T(\mathbf{y}^\mu) - R)^2$$

236 subject to the constraints that $\|\nu\|_\infty < 1$ and \mathbf{y}^μ be the quasi-conformal map with
 237 Beltrami coefficient μ satisfying $\mathbf{y}^\mu(p_j) = q_j$ for $1 \leq j \leq m$. Then in each iteration,
 238 it needs to solve the following two subproblems alternately:

$$239 \quad (11) \quad \begin{aligned} \mu_{n+1} &= \arg \min \sigma \int_{\Omega} |\mu - \nu_n|^2 + \beta \int_{\Omega} (T(\mathbf{y}^\mu) - R)^2 \\ \text{s.t. } \mathbf{y}^\mu(p_j) &= q_j \text{ for } 1 \leq j \leq m \end{aligned}$$

240 and

$$241 \quad (12) \quad \nu_{n+1} = \arg \min \int_{\Omega} |\nabla \nu|^2 + \alpha \int_{\Omega} |\nu|^p + \sigma \int_{\Omega} |\nu - \mu_{n+1}|^2.$$

242 In addition, it also solves the equation $\mu(\mathbf{y}) = \mu$ by the linear Beltrami solver (LBS)
 243 [34] to find \mathbf{y} and ensures that \mathbf{y} matches the landmark constraints.

244 Thus, instead of controlling the Jacobian determinant of the transformation di-
 245 rectly, controlling Beltrami coefficient is also a good alternative providing the same
 246 but indirect control. However, since their algorithm [30] has to deal with two main
 247 unknowns (the transformation \mathbf{y} and its Beltrami coefficient μ) and one auxiliary un-
 248 known (the coefficient ν) in a non-convex formulation, the increased cost, practical

249 implementation and convergence are real issues; for challenging problems, one cannot
 250 observe convergence and therefore the full capability of the model is not realized.

251 We are motivated to reduce the unknowns and simplify their algorithm. Our
 252 solution is to reformulate the problem in the space of the primary variable \mathbf{y} or \mathbf{u} ,
 253 not in the transformed space of variables μ, ν . We make use of the explicit formula
 254 of $\mu = \mu(\mathbf{y})$. Working with primal mapping \mathbf{y} enables us to introduce the advantages
 255 of minimizing a Beltrami coefficient to the above reviewed variational framework (2),
 256 effectively unifying the two frameworks.

257 Hence, we propose a new regularizer based Beltrami coefficient and, in the numer-
 258 ical part, we can find that it is easy to be implemented. Moreover the reformulated
 259 control regularizer can potentially be applied to a large class of variational models.

260 **3. The proposed image registration model.** In this section, we aim to
 261 present a new regularizer based on Beltrami coefficient, which can help to get a
 262 diffeomorphic transformation. Then combining the new regularizer with the diffusion
 263 model, we present a novel model. Of course, combining with other models may be
 264 studied as well since the idea is the same.

265 For $f(z) = y_1(x_1, x_2) + \mathbf{i}y_2(x_1, x_2)$, according to the Beltrami equation (7) and
 266 the definitions (8), we have

267 (13)
$$\mu(f) = \frac{\partial f}{\partial \bar{z}} / \frac{\partial f}{\partial z} = \frac{((y_1)_{x_1} - (y_2)_{x_2}) + \mathbf{i}((y_2)_{x_1} + (y_1)_{x_2})}{((y_1)_{x_1} + (y_2)_{x_2}) + \mathbf{i}((y_2)_{x_1} - (y_1)_{x_2})},$$

268

269 (14)
$$|\mu(f)|^2 = \frac{((y_1)_{x_1} - (y_2)_{x_2})^2 + ((y_2)_{x_1} + (y_1)_{x_2})^2}{((y_1)_{x_1} + (y_2)_{x_2})^2 + ((y_2)_{x_1} - (y_1)_{x_2})^2} = \frac{\|J_f\|_2^2 - 2 \det(J_f)}{\|J_f\|_2^2 + 2 \det(J_f)}.$$

270 Note $(y_1)_{x_1}(y_2)_{x_2} - (y_2)_{x_1}(y_1)_{x_2} = \det(J_f)$. So $\det(J_f)$ can be represented by the
 271 Beltrami coefficient $\mu(f)$

272 (15)
$$\det(J_f) = |f_z|^2(1 - |\mu(f)|^2)$$

273 Clearly $\det(\nabla f) > 0$ if $|\mu(f)| < 1$, and by the inverse function theorem, the map
 274 f is locally bijective. We conclude that f is diffeomorphism if we assume that Ω is
 275 bounded, simply connected.

276 For more details about quasi-conformal theory, the readers can refer to [1, 20, 31].

277 **3.1. New regularizer.** Our new regularizer based on $|\mu(f)| < 1$ to control the
 278 transformation to get a diffeomorphic mapping is

279 (16)
$$S_1[\mathbf{y}] = \int_{\Omega} \phi(|\mu|^2) d\mathbf{x}, \quad |\mu|^2 = \frac{\|J_{\mathbf{y}}\|_2^2 - 2 \det(J_{\mathbf{y}})}{\|J_{\mathbf{y}}\|_2^2 + 2 \det(J_{\mathbf{y}})}$$

280 which clearly involves the Jacobian determinant $\det(J_{\mathbf{y}})$ in a non-trivial way and we
 281 explore the choices of ϕ below.

282 **REMARK.** *Our new regularizer has two advantages: one is that the obtained trans-*
 283 *formation \mathbf{y} do not need to possess $\det(J_{\mathbf{y}}) \rightarrow 1$; the other one is that we only compute*
 284 *the transformation and do not need to compute its Beltrami coefficient and introduce*
 285 *another auxiliary unknown as [30]. In addition, from the numerical experiments,*
 286 *we can see that our new regularizer is easy to implement and obtains accurate and*
 287 *diffeomorphic transformations.*

288 **3.2. The proposed model.** The above regularizer (16) providing a constraint
 289 on \mathbf{y} is ready to be combined with an existing model. In the framework (5), using
 290 (16), the first version of our new model takes the form

$$291 \quad (17) \quad \min_{\mathbf{y}} \frac{1}{2} \|T(\mathbf{y}) - R\|_2^2 + \frac{\alpha}{2} \|\nabla \mathbf{u}\|_2^2 + \beta \int_{\Omega} \phi(|\mu|^2) d\mathbf{x}$$

292 where $\mathbf{u} = \mathbf{y}(\mathbf{x}) - \mathbf{x} = (y_1(\mathbf{x}), y_2(\mathbf{x})) - \mathbf{x}$ is the deformation field, $|\nabla \mathbf{u}|^2 = |\nabla u_1|^2 +$
 293 $|\nabla u_2|^2$ and $\mu = \mu(\mathbf{y})$. To promote $|\mu(f)| < 1$, our first and simple choice is $\phi(v) =$
 294 $\phi_1(v) = \frac{1}{(v-1)^2}$, which forces (17) and $\phi(v)$ to reduce v , at the initial guess $v = 0$
 295 when $\mathbf{u} = \mathbf{0}$, since $\phi_1(v) \rightarrow \infty$ when $v \rightarrow 1$.

296 **REMARK.** From (9) and (17), we see that the QCHR model focuses on obtaining a
 297 smooth Beltrami coefficient and our model focuses on the diffeomorphic transformation
 298 itself. There are major differences between the regularizer in QCHR model and our
 299 new regularizer: the former is characterized by the Beltrami coefficient μ directly and
 300 gradient of this Beltrami coefficient μ , while the latter is characterized by the Beltrami
 301 coefficient indirectly in terms of the transformation \mathbf{y} and the gradient of \mathbf{u} . Since
 302 $\mathbf{y} = \mathbf{x} + \mathbf{u}$ is our desired transformation, our direct regularizers such as $|\nabla \mathbf{u}|^2$ make
 303 more sense than indirect regularizers such as $|\nabla \mu|^2$.

304 However as long as $|\mu(f)| < 1$, we would not give a preference to forcing $|\mu(f)| \rightarrow$
 305 0. To put some control on bias, similarly to [7], we are led to 2 more choices of a less
 306 unbiased function to modify $S_1[\mathbf{y}]$

- 307 • $\phi(v) = \phi_2(v) = \frac{v}{(v-1)^2}$: balance $|\mu(f)|$ between 0 and 1 as $\phi_2(v) = \phi_2(1/v)$;
- 308 • $\phi(v) = \phi_3(v) = \frac{v^2}{(v-1)^2}$: encourage $|\mu(f)| \rightarrow 0$ and $|\mu(f)| \neq 1$;

309 Below, we list first order derivatives and second order derivatives for the above
 310 different $\phi(v)$:

- 311 • $\phi'_1(v) = \frac{2}{(v-1)^3}$ and $\phi''_1(v) = \frac{6}{(v-1)^4}$;
- 312 • $\phi'_2(v) = -\frac{v+1}{(v-1)^2}$ and $\phi''_2(v) = \frac{2v+4}{(v-1)^4}$;
- 313 • $\phi'_3(v) = -\frac{2v}{(v-1)^3}$ and $\phi''_3(v) = \frac{4v+2}{(v-1)^4}$

314 which will be used in subsequent solutions. With a general $\phi(v)$, the second version
 315 of our proposed model takes the form:

$$316 \quad (18) \quad \min_{\mathbf{u}} \frac{1}{2} \int_{\Omega} (T(\mathbf{x} + \mathbf{u}) - R)^2 d\mathbf{x} + \frac{\alpha}{2} \int_{\Omega} \sum_{\ell=1}^2 |\nabla u_{\ell}|^2 d\mathbf{x} + \beta \int_{\Omega} \phi(|\mu|^2) d\mathbf{x},$$

317 where $|\mu|^2 = \frac{(\partial_{x_1} u_1 - \partial_{x_2} u_2)^2 + (\partial_{x_1} u_2 + \partial_{x_2} u_1)^2}{(\partial_{x_1} u_1 + \partial_{x_2} u_2 + 2)^2 + (\partial_{x_1} u_2 - \partial_{x_2} u_1)^2}$ is written in component form ready for
 318 discretization, using $y_1 = x_1 + u_1(x_1, x_2)$, $y_2 = x_2 + u_2(x_1, x_2)$, and $\partial_{x_1} u_1 = \partial u_1 / \partial x_1$.

319 **REMARK.** For the existence or uniqueness of a solution of (18), this is out of the
 320 scope of the present work and will be considered in our forthcoming work.

321 **4. The numerical algorithm.** In this section, we will present a numerical al-
 322 gorithm to solve model (18). We choose the discretize - optimize approach. Directly
 323 discretizing this variational model gives rise to a finite dimensional optimization prob-
 324 lem. Then we use optimization methods to solve this resulting problem.

325 **4.1. Discretization.** We use finite differences to discretize model (18) on a unit
 326 square domain $\Omega = [0, 1]^2$. In implementation, we employ the nodal grid and define
 327 a spatial partition $\Omega_h = \{\mathbf{x}^{i,j} \in \Omega \mid \mathbf{x}^{i,j} = (x_1^i, x_2^j) = (ih, jh), 0 \leq i \leq n, 0 \leq j \leq n\}$,

328 where $h = \frac{1}{n}$ and the discrete domain consists of n^2 cells of size $h \times h$. We dis-
 329 cretize the displacement field \mathbf{u} on the nodal grid, namely $\mathbf{u}^{i,j} = (u_1^{i,j}, u_2^{i,j}) =$
 330 $(u_1(x_1^i, x_2^j), u_2(x_1^i, x_2^j))$. For ease presentation, according to the lexicographical or-
 331 dering, we reshape

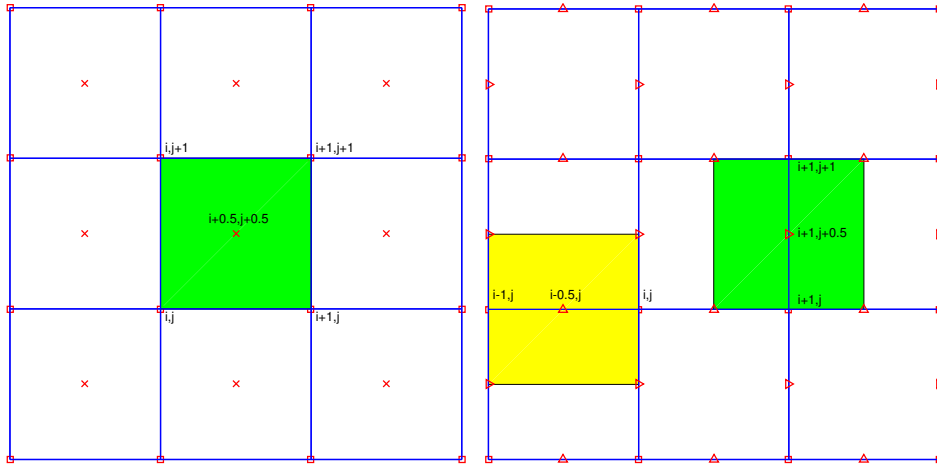
$$332 \quad X = (x_1^0, \dots, x_1^n, \dots, x_1^0, \dots, x_1^n, x_2^0, \dots, x_2^n, \dots, x_2^0, \dots, x_2^n)^T \in \mathbb{R}^{2(n+1)^2 \times 1},$$

333 and

$$334 \quad U = (u_1^{0,0}, \dots, u_1^{n,0}, \dots, u_1^{0,n}, \dots, u_1^{n,n}, u_2^{0,0}, \dots, u_2^{n,0}, \dots, u_2^{0,n}, \dots, u_2^{n,n})^T \in \mathbb{R}^{2(n+1)^2 \times 1}.$$

335 **4.1.1. Discretization of Term 1 in (18).** According to the cell-centered parti-
 336 tion in Figure 1(a) and mid-point rule, we get

$$337 \quad (19) \quad \begin{aligned} \mathcal{D}[\mathbf{u}] &:= \frac{1}{2} \int_{\Omega} (T(\mathbf{x} + \mathbf{u}(\mathbf{x})) - R(\mathbf{x}))^2 dx \\ &= \frac{h^2}{2} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (T(\mathbf{x}^{i+\frac{1}{2}, j+\frac{1}{2}} + \mathbf{u}(\mathbf{x}^{i+\frac{1}{2}, j+\frac{1}{2}})) - R(\mathbf{x}^{i+\frac{1}{2}, j+\frac{1}{2}}))^2. \end{aligned}$$



(a) Illustration of cell-centered partition: Green cell denoted by $\Omega_{i,j}$. Nodal Grid \square (b) Partition for ∂_x and ∂_y . The left yellow cell is $\Omega_{i,j}^{x_1}$ and the right green cell is $\Omega_{i,j}^{x_2}$.

FIG. 1. Partition of domain $\Omega = \cup_{i,j} \Omega_{i,j}$. Note that solutions u_1 and u_2 are defined at nodes.

338 Set $\vec{R} = R(PX) \in \mathbb{R}^{n^2 \times 1}$ as the discretized reference image and $\vec{T}(PX + PU) \in$
 339 $\mathbb{R}^{n^2 \times 1}$ as the discretized deformed template image, where $P \in \mathbb{R}^{2n^2 \times 2(n+1)^2}$ is an
 340 averaging matrix for the transfer from the nodal grid representation of U to the cell
 341 centered positions.

342 Consequently, for SSD, we obtain the following discretization:

$$343 \quad (20) \quad \mathcal{D}[\mathbf{u}] \approx \frac{h^2}{2} (\vec{T}(PX + PU) - \vec{R})^T (\vec{T}(PX + PU) - \vec{R}).$$

344 **4.1.2. Discretization of Term 2 in (18).** For the diffusion regularizer,

345 (21)
$$\mathcal{S}_{\text{diff}}[\mathbf{u}] := \frac{\alpha}{2} \int_{\Omega} \sum_{\ell=1}^2 |\nabla u_{\ell}|^2 d\mathbf{x},$$

346 according to the the partition in Figure 1(b) and mid-point rule, we have

347 (22)
$$\int_{\Omega_{i,j}^{x_1}} |\partial_{x_1} u_{\ell}|^2 d\mathbf{x} \approx h^2 (\partial_{x_1}^{i+\frac{1}{2},j} u_{\ell})^2 \quad 1 \leq j \leq n-1,$$

348 or at the boundary half-boxes

349 (23)
$$\int_{\Omega_{i,j}^{x_1}} |\partial_{x_1} u_{\ell}|^2 d\mathbf{x} \approx \frac{h^2}{2} (\partial_{x_1}^{i+\frac{1}{2},j} u_{\ell})^2 \quad j = 0, n.$$

350 And for $\int_{\Omega_{i,j}^{x_2}} |\partial_{x_2} u_{\ell}|^2 d\mathbf{x}$, $\ell = 1, 2$, we have similar results.

351 As designed, we use compact (short) difference schemes to compute the $\partial_{x_1} u_{\ell}$ and
352 $\partial_{x_2} u_{\ell}$, $\ell = 1, 2$:

353 (24)
$$\partial_{x_1}^{i+\frac{1}{2},j} u_{\ell} \approx \frac{u_{\ell}^{i+1,j} - u_{\ell}^{i,j}}{h}, \quad \partial_{x_2}^{i,j+\frac{1}{2}} u_{\ell} \approx \frac{u_{\ell}^{i,j+1} - u_{\ell}^{i,j}}{h}.$$

354 Then (21) can be rewritten in the following formulation:

355 (25)
$$\mathcal{S}_{\text{diff}}[\mathbf{u}] \approx \frac{\alpha h^2}{2} U^T A^T G A U.$$

356 See **Appendix A** for details on A and G .

357 *REMARK.* Note that here the matrix A is the discretized gradient matrix. So
358 $A^T G A$ is the discretized Laplace matrix.

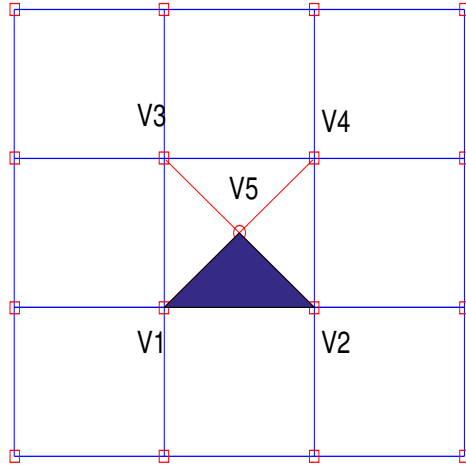


FIG. 2. Partition of a cell, nodal point \square and center point \circ . $\triangle V_1 V_2 V_5$ is $\Omega_{i,j,k}$.

359 **4.1.3. Discretization of Term 3 in (18).** For simplicity, denote $|\mu(\mathbf{y})| =$
360 $|\mu(\mathbf{x} + \mathbf{u})|$ by $|\mu(\mathbf{u})|$. From (18), note that $\phi(|\mu(\mathbf{u})|^2)$ involves only first order deriva-
361 tives and all $\mathbf{u}^{i,j}$ are available at vertex pixels. Thus it is convenient first to obtain

362 approximations at all cell centres (e.g. at V_5 in Figure 2) and second to use local
 363 linear elements to facilitate first order derivatives. We shall divide each cell (Figure
 364 2) into 4 triangles. In each triangle, we construct two linear interpolation functions to
 365 approximate the u_1 and u_2 . Consequently, all partial derivatives are locally constants
 366 or $\phi(|\mu(\mathbf{u})|^2)$ is constant in each triangle.

367 According to the partition in Figure 2, we get

$$368 \quad (26) \quad \mathcal{S}_{\text{Beltrami}}[\mathbf{u}] := \beta \int_{\Omega} \phi(|\mu(\mathbf{u})|^2) d\mathbf{x} = \beta \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^4 \int_{\Omega_{i,j,k}} \phi(|\mu(\mathbf{u})|^2) d\mathbf{x}.$$

369 Set $\mathbf{L}^{i,j,k}(\mathbf{x}) = (L_1^{i,j,k}(\mathbf{x}), L_2^{i,j,k}(\mathbf{x})) = (a_1^{i,j,k}x_1 + a_2^{i,j,k}x_2 + a_3^{i,j,k}, a_4^{i,j,k}x_1 + a_5^{i,j,k}x_2 +$
 370 $a_6^{i,j,k})$, which is the linear interpolation for \mathbf{u} in the $\Omega_{i,j,k}$. Note that $\partial_{x_1} L_1^{i,j,k} =$
 371 $a_1^{i,j,k}$, $\partial_{x_2} L_1^{i,j,k} = a_2^{i,j,k}$, $\partial_{x_1} L_2^{i,j,k} = a_4^{i,j,k}$ and $\partial_{x_2} L_2^{i,j,k} = a_5^{i,j,k}$. According to (18), the
 372 discretization of Beltrami regularizer can be written into following:

$$373 \quad (27) \quad \mathcal{S}_{\text{Beltrami}}[\mathbf{u}] \approx \frac{\beta h^2}{4} \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^4 \phi \left(\frac{(a_1^{i,j,k} - a_5^{i,j,k})^2 + (a_2^{i,j,k} + a_4^{i,j,k})^2}{(a_1^{i,j,k} + a_5^{i,j,k} + 2)^2 + (a_2^{i,j,k} - a_4^{i,j,k})^2} \right).$$

374 To simplify (27), define 3 vectors $\vec{\mathbf{r}}(U), \vec{\mathbf{r}}^1(U), \vec{\mathbf{r}}^2(U) \in \mathbb{R}^{4n^2}$ by $\vec{\mathbf{r}}(U)_\ell = \vec{\mathbf{r}}^1(U)_\ell \vec{\mathbf{r}}^2(U)_\ell$,
 375 $\vec{\mathbf{r}}^1(U)_\ell = (a_1^{i,j,k} - a_5^{i,j,k})^2 + (a_2^{i,j,k} + a_4^{i,j,k})^2$, $\vec{\mathbf{r}}^2(U)_\ell = 1/[(a_1^{i,j,k} + a_5^{i,j,k} + 2)^2 + (a_2^{i,j,k} -$
 376 $a_4^{i,j,k})^2]$ where $\ell = (k-1)n^2 + (j-1)n + i \in [1, 4n^2]$. Hence, (27) becomes

$$377 \quad (28) \quad \mathcal{S}_{\text{Beltrami}}[\mathbf{u}] \approx \frac{\beta h^2}{4} \phi(\vec{\mathbf{r}}(U)) e^T$$

378 where $\phi(\vec{\mathbf{r}}(U)) = (\phi(\vec{\mathbf{r}}(U)_1), \dots, \phi(\vec{\mathbf{r}}(U)_{4n^2}))$ denotes the pixel-wise discretization of
 379 u_1, u_2 at all cell centers, and $e = (1, \dots, 1) \in \mathbb{R}^{4n^2}$. Here, $\vec{\mathbf{r}}(U)$ is the square of the
 380 discretized Beltrami coefficient; we rewrite it in a compact form in **Appendix B**.

381 Finally, combining the above three parts (20), (25) and (28), we get the discretiza-
 382 tion formulation for model (18):

$$383 \quad (29) \quad \min_U J(U) := \frac{h^2}{2} (\vec{T}(PX + PU) - \vec{R})^T (\vec{T}(PX + PU) - \vec{R}) + \frac{\alpha h^2}{2} U^T A^T G A U \\
 + \frac{\beta h^2}{4} \phi(\vec{\mathbf{r}}(U)) e^T.$$

384 **REMARK.** According to the definition of ϕ and $\vec{\mathbf{r}}(U)_\ell \geq 0$, each component of
 385 $\phi(\vec{\mathbf{r}}(U))$ is non-negative and differentiable.

386 **4.2. Optimization method for the discretized problem (29).** In the nu-
 387 merical implementation, we choose line search method to solve the resulting uncon-
 388 strained optimization problem (29). In order to guarantee the search direction is a
 389 descent direction, we employ the Gauss-Newton direction as the standard direction
 390 involving non-definite Hessians does not generate a descent direction. Otherwise, us-
 391 ing a Gauss-Newton approach presents two advantages: one is that we do not need
 392 to compute the second order term and it can save computation time; the other one
 393 is that this Gauss-Newton matrix is more important than the second term, either
 394 because of small second order derivatives or because of small residuals [42].

395 Let $J(U) : \mathbb{R}^{2(n+1)^2} \rightarrow \mathbb{R}$ be twice continuously differentiable, $U^k \in \mathbb{R}^{2(n+1)^2}$ and
 396 the approximated Hessian $H(U^k)$ positive definite. We model J at the current point

397 U^k by the quadratic approximation $q^k(s)$,

$$398 \quad (30) \quad J(U^k + s) \approx q^k(s) = J(U^k) + d_J(U^k)^T s + \frac{1}{2} s^T H(U^k)^T s,$$

399 where $s = U - U^k$ and $d_J(U^k) = \nabla J(U^k)$. Minimizing $q^k(s)$ yields

$$400 \quad (31) \quad U^{k+1} = U^k - [H(U^k)]^{-1} d_J(U^k).$$

401 In order to guarantee the global convergence of the Gauss-Newton method, we
402 employ the line search and its iteration is as follows:

$$403 \quad (32) \quad U^{k+1} = U^k - \theta_k [H(U^k)]^{-1} d_J(U^k).$$

404 where θ_k is a step length.

405 Next, we will investigate the details about the approximated Hessian $H(U^k)$, step
406 length θ_k , stopping criteria and multilevel strategy.

407 **4.2.1. Approximated Hessian H .** We consider each of the three terms in
408 $J(U)$ from (29) separately.

409 Firstly, we consider the discretized SSD

$$410 \quad (33) \quad \frac{h^2}{2} (\vec{T}(PX + PU) - \vec{R})^T (\vec{T}(PX + PU) - \vec{R}).$$

411 Its gradient and Hessian are respectively

$$412 \quad (34) \quad \begin{cases} d_1 &= h^2 P^T \vec{T}_{\tilde{U}}^T (\vec{T}(\tilde{U}) - \vec{R}) \in \mathbb{R}^{2(n+1)^2 \times 1}, \\ H_1 &= h^2 P^T (\vec{T}_{\tilde{U}}^T \vec{T}_{\tilde{U}} + \sum_{\ell=1}^{n^2} (\vec{T}(\tilde{U}) - \vec{R})_{\ell} \nabla^2 (\vec{T}(\tilde{U}) - \vec{R})_{\ell}) P \end{cases}$$

413 where $\tilde{U} = PX + PU$ and $\vec{T}_{\tilde{U}} = \frac{\partial \vec{T}(\tilde{U})}{\partial \tilde{U}}$ as the Jacobian of \vec{T} with respect to \tilde{U} .

414 For H_1 , we cannot ensure that it is positive semi-definite. If it is not positive
415 definite, we may not get a descent direction. So we omit the second order term of H_1
416 to obtain the approximated Hessian of (33):

$$417 \quad (35) \quad \hat{H}_1 = h^2 P^T (\vec{T}_{\tilde{U}}^T \vec{T}_{\tilde{U}}) P.$$

418 **REMARK.** *Evaluation of the deformed template image T must involve interpola-*
419 *tion because \tilde{U} do not in general correspond to pixel points; in our implementation,*
420 *as with [39], we use B-splines interpolation to get $\vec{T}(\tilde{U})$.*

421 Secondly, for the discretized diffusion regularizer $\frac{\alpha h^2}{2} U^T A^T GAU$, its gradient and
422 Hessian are the following:

$$423 \quad (36) \quad \begin{cases} d_2 &= \alpha h^2 A^T GAU \in \mathbb{R}^{2(n+1)^2 \times 1}, \\ H_2 &= \alpha h^2 A^T GA \in \mathbb{R}^{2(n+1)^2 \times 2(n+1)^2}. \end{cases}$$

424 Since H_2 is positive definite when U is applied with Dirichet boundary conditions, we
425 do not approximate it.

426 Finally, for the discretized Beltrami term

$$427 \quad (37) \quad \frac{\beta h^2}{4} \phi(\vec{r}(U)) e^T,$$

428 the gradient and the Hessian are as follows:

$$429 \quad (38) \quad \begin{cases} d_3 &= \frac{\beta h^2}{4} d\bar{\mathbf{r}}^T d\phi(\bar{\mathbf{r}}) \in \mathbb{R}^{2(n+1)^2 \times 1}, \\ H_3 &= \frac{\beta h^2}{4} (d\bar{\mathbf{r}}^T d^2\phi(\bar{\mathbf{r}}) d\bar{\mathbf{r}} + \sum_{\ell=1}^{4n^2} [d\phi(\bar{\mathbf{r}})]_{\ell} \nabla^2 \bar{\mathbf{r}}_{\ell}) \in \mathbb{R}^{2(n+1)^2 \times 2(n+1)^2} \end{cases}$$

430 where $d\phi(\bar{\mathbf{r}}) = (\phi'(\bar{\mathbf{r}}_1), \dots, \phi'(\bar{\mathbf{r}}_{4n^2}))^T$ is the vector of derivatives of ϕ at all cell centers,

$$431 \quad (39) \quad \begin{cases} d\bar{\mathbf{r}} &= \text{diag}(\bar{\mathbf{r}}^1) d\bar{\mathbf{r}}^2 + \text{diag}(\bar{\mathbf{r}}^2) d\bar{\mathbf{r}}^1, \\ d\bar{\mathbf{r}}^1 &= 2 \text{diag}(A_1 U) A_1 + 2 \text{diag}(A_2 U) A_2, \\ d\bar{\mathbf{r}}^2 &= -\text{diag}(\bar{\mathbf{r}}^2 \odot \bar{\mathbf{r}}^2) [2 \text{diag}(A_3 U + 2) A_3 + 2 \text{diag}(A_4 U) A_4], \end{cases}$$

432 \odot denotes a Hadamard product, $d\bar{\mathbf{r}}, d\bar{\mathbf{r}}^1, d\bar{\mathbf{r}}^2$ are the Jacobian of $\bar{\mathbf{r}}, \bar{\mathbf{r}}^1, \bar{\mathbf{r}}^2$ with respect
 433 to U respectively, $[d\phi(\bar{\mathbf{r}})]_{\ell}$ is the ℓ th component of $d\phi(\bar{\mathbf{r}})$ and $d^2\phi(\bar{\mathbf{r}})$ is the Hessian
 434 of ϕ with respect to $\bar{\mathbf{r}}$, which is a diagonal matrix whose i th diagonal element is
 435 $\phi''(\bar{\mathbf{r}}_i)$, $1 \leq i \leq 4n^2$. Here $\text{diag}(v)$ is a diagonal matrix with v on its main diagonal.
 436 More details about $\bar{\mathbf{r}}^1, \bar{\mathbf{r}}^2, A_1, A_2, A_3$ and A_4 are shown in **Appendix B** and some
 437 illustration of our notation is given in **Appendix C**.

438 To extract a positive semi-definite part out of (38), we omit the second order
 439 term and obtain the approximated Hessian as

$$440 \quad (40) \quad \hat{H}_3 = \frac{\beta h^2}{4} d\bar{\mathbf{r}}^T d^2\phi(\bar{\mathbf{r}}) d\bar{\mathbf{r}}.$$

441 Therefore for functional $J(U)$ in (29) with any choice of ϕ , we obtain its gradient

$$442 \quad (41) \quad d_J = d_1 + d_2 + d_3$$

443 and approximated Hessian:

$$444 \quad (42) \quad H = \hat{H}_1 + H_2 + \hat{H}_3.$$

445 **4.2.2. Search Direction.** At each iteration, using (41) and (42), we need to
 446 solve the Gauss-Newton system to find the search direction of (29):

$$447 \quad (43) \quad H\delta U = -d_J,$$

448 where δU is the search direction. In our implementation, we use MINRES with
 449 diagonal preconditioning to solve this linear system [2, 43].

450 **4.2.3. Step Length.** We use the standard Armijo strategy with backtracking
 451 to find a suitable step length θ . In the implementation, we also need to check that
 452 $\bar{\mathbf{r}}(U)$ (54) is smaller than 1. Recall that $\bar{\mathbf{r}}(U)$ is the norm square of the discretized
 453 Beltrami term. As a safe guard, we choose $T0 = 10^{-8}$ and $Tol = 10^{-12}$ as the lower
 454 bound of the step length θ and $\theta \|\delta U\|$ [7, 28, 42, 48]. The algorithm is summarized
 455 in Algorithm 1.

456 **4.2.4. Stopping Criteria.** Here, we adopt the stopping criteria as in [39]:

- 457 (1.a) $\|J(U^{i+1}) - J(U^i)\| \leq \tau_J (1 + \|J(U^0)\|)$,
- 458 (1.b) $\|\mathbf{y}^{i+1} - \mathbf{y}^i\| \leq \tau_W (1 + \|\mathbf{y}^0\|)$,
- 459 (1.c) $\|d_J\| \leq \tau_G (1 + \|J(U^0)\|)$,
- 460 (2) $\|d_J\| \leq \text{eps}$,
- 461 (3) $i \geq \text{MaxIter}$.

Algorithm 1 Armijo Line Search: $(U, \text{ID}) \leftarrow \text{ALS}(U, \delta U)$

Step 1: Initialisation. Set $\text{ID} = 0$, $\theta = 1$, $\text{Tol} = 10^{-12}$, $\text{T0} = 10^{-8}$ and $\eta = 10^{-4}$.
 Compute $J(U)$ and d_J .

Step 2: Feasibility checking.
while $\theta \|\delta U\| \geq \text{Tol}$ **do**
 $U^{new} = U + \theta \delta U$;
 if $\|\bar{\mathbf{r}}(U^{new})\| \leq 1$ **then**
 If $\theta \geq \text{T0}$, exit this while loop and go to Step 3, else if $\theta < \text{T0}$, go to Step 4.
 end if
 Reduce the factor θ by $\theta = \theta/2$;
end while

Step 3: Line Search.
while $\theta \|\delta U\| \geq \text{Tol}$ **do**
 Compute $J(U^{new})$;
 if $J(U^{new}) < J(U) + \theta \eta d_J^T \delta U$ **then**
 If $\theta \geq \text{T0}$, exit this algorithm with $U = U^{new}$, else if $\theta < \text{T0}$, go to Step 4.
 end if
 Reduce the factor θ by $\theta = \theta/2$;
 $U^{new} = U + \theta \delta U$;
end while

Step 4: Set $\text{ID} = 1$ and $U = U^{new}$.

462 Here, eps is the machine precision and MaxIter is the maximal number of outer
 463 iterations. We set $\tau_J = 10^{-3}$, $\tau_W = 10^{-2}$, $\tau_G = 10^{-2}$ and $\text{MaxIter} = 500$. If any one
 464 of (1) (2) and (3) is satisfied, the iterations are terminated. Hence, a Gauss-Newton
 465 numerical scheme with Armijo line search can be developed. The resulting Gauss-
 466 Newton numerical scheme by using Armijo line search is summarized in Algorithm
 467 2.

Algorithm 2 Gauss-Newton scheme by using Armijo line search for Image Registra-
 tion: $(U, \text{ID}) \leftarrow \text{GNAIRA}(\alpha, \beta, U^0, T, R)$

Step 1: Set $i = 0$ at the solution point $U^i = U^0$.

Step 2: For (29), compute the energy functional $J(U^i)$, its gradient d_J^i and
 the approximated Hessian H^i by (42).

while “none of the listed 3 stopping criteria are satisfied” **do**
 — Solve the Gauss-Newton equation: $H^i \delta U^i = -d_J^i$;
 — $(U^{i+1}, \text{ID}) \leftarrow \text{ALS}(U^i, \delta U^i)$ by Algorithm 1;
if $\text{ID} = 1$ **then**
 Exit this algorithm.
else
 $i = i + 1$;
 Compute $J(U^i)$, d_J^i and H^i ;
end if
end while

468 Next, we discuss the global convergence result of Algorithm 2 for our reformulated
 469 problem (29). Firstly, we review some relevant theorem.

470 THEOREM 1 ([28]). *For the unconstrained optimization problem*

471
$$\min_U J(U)$$

472 *let an iterative sequence be defined by $U^{i+1} = U^i + \theta \delta U^i$, where $\delta U^i = -(H^i)^{-1} d_J(U^i)$*
 473 *and θ is obtained by Algorithm 1. Assume that three conditions are met: (i). d_J be*
 474 *Lipschitz continuous; (ii). the matrices H^i are SPD (iii). there exist constants $\bar{\kappa}$ and*
 475 *λ such that the condition number $\kappa(H^i) \leq \bar{\kappa}$ and the norm $\|H^i\| \leq \lambda$ for all i . Then*
 476 *either $J(U^i)$ is unbounded from below or*

477 (44)
$$\lim_{i \rightarrow \infty} d_J(U^i) = 0$$

478 *and hence any limit point of the sequence of iterates is a stationary point.*

479 REMARK. *In the above discretization leading to (29), we do not need to introduce*
 480 *the boundary condition. However for theory purpose, in the following, we will prove*
 481 *our convergence result under the Dirichlet boundary condition (namely, the boundary*
 482 *is fixed) and this condition is needed to prove the symmetric positive definite (SPD)*
 483 *property of the approximated Hessians. In practical implementation, such a condition*
 484 *is not required as confirmed by experiments.*

485 In addition, define an important set $\mathcal{X} := \{U \mid \bar{\mathbf{r}}(U)_\ell \leq 1 - \epsilon, 1 \leq \ell \leq 4n^2\}$
 486 for small ϵ . So $U \in \mathcal{X}$ means that the transformation is diffeomorphic. Under the
 487 suitable β , we assume that each U^i generated by Algorithm 2 is in the \mathcal{X} .

488 Secondly we stage a simple lemma that is needed shortly for studying H^i .

489 LEMMA 2. *Let a matrix be comprised of 3 submatrices $H = H_1 + H_2 + H_3$. If*
 490 *H_1 and H_2 are symmetric positive semi-definite and H_3 is SPD, then H is SPD with*
 491 *$\lambda_{h_3} \leq \lambda_h$, where λ_{h_3} and λ_h are the minimum eigenvalues of H_3 and H separately.*

492 *Proof.* According to Rayleigh quotient, we can find a vector v such that

493 (45)
$$\lambda_h = \frac{v^T H v}{v^T v}.$$

494 Then we have

495 (46)
$$\lambda_{h_3} \leq \frac{v^T H_1 v}{v^T v} + \frac{v^T H_2 v}{v^T v} + \frac{v^T H_3 v}{v^T v} = \frac{v^T H v}{v^T v} = \lambda_h,$$

496 which completes the proof. □

497 THEOREM 3. *Assume that T and R are twice continuously differentiable. For*
 498 *(29), when $\phi = \phi_1, \phi_2$ or ϕ_3 , by using Algorithm 2, we obtain*

499 (47)
$$\lim_{i \rightarrow \infty} d_J(U^i) = 0$$

500 *and hence any limit point of the sequence of iterates produced by Algorithm 2 is a*
 501 *stationary point.*

502 *Proof.* It suffices to show that Algorithm 2 satisfies the requirements of Theorem
 503 1. Recall $\bar{\mathbf{r}}(U)$ and we can see that it is continuous. Here, we use the Dirichlet bound-
 504 ary condition and we can assume that $\|U\|$ is bounded. Then $\bar{\mathbf{r}}(U)$ is a continuous
 505 mapping from a compact set to $\mathbb{R}^{4n^2 \times 1}$ and $\bar{\mathbf{r}}(U)$ is proper. So for some small $\epsilon > 0$,
 506 \mathcal{X} is compact.

507 Firstly, we show that in \mathcal{X} , d_J of (29) is Lipschitz continuous. When $\phi = \phi_1, \phi_2$
 508 or ϕ_3 , the term $\phi(\vec{\mathbf{r}}(U))e^T$ in the (29) is twice continuously differentiable with respect
 509 to $U \in \mathcal{X}$. In addition, T and R are twice continuously differentiable. So (29) is twice
 510 continuously differentiable with respect to $U \in \mathcal{X}$ and d_J is Lipschitz continuous.

511 Secondly, we show that in \mathcal{X} , $H^i = \hat{H}_1^i + H_2^i + \hat{H}_3^i$ is SPD. By the construction
 512 of \hat{H}_1^i and \hat{H}_3^i , they are symmetric positive semi-definite. H_2^i is symmetric positive
 513 definite under the Dirichlet boundary condition. Consequently H^i is SPD.

514 Thirdly, we show that both $\kappa(H^i)$ and $\|H^i\|$ are bounded. We notice that
 515 in each iteration, $H_2^i = \alpha h^2 A^T G A$ is constant and we can set $\|H_2^i\| = M_2$. For
 516 $\hat{H}_1^i = h^2 P^T (\vec{T}_U^T \vec{T}_U) P$, we get its upper bound M_1 because T is twice continuously
 517 differentiable and \mathcal{X} is compact. For $\phi = \phi_1, \phi_2$ or ϕ_3 , ϕ is twice continuously differ-
 518 entiable with respect to $U \in \mathcal{X}$, then we have $\|\hat{H}_3^i\| \leq \frac{\beta h^2}{4} \|\mathbf{d}\vec{\mathbf{r}}^T\| \|\mathbf{d}^2\phi(\vec{\mathbf{r}})\| \|\mathbf{d}\vec{\mathbf{r}}\| \leq M_3$.
 519 Hence, we have

$$520 \quad (48) \quad \|H^i\| \leq \|\hat{H}_1^i\| + \|H_2^i\| + \|\hat{H}_3^i\| \leq M_1 + M_2 + M_3.$$

521 So set $M = M_1 + M_2 + M_3$ and $\|H^i\| \leq M$. Set σ as the minimum eigenvalue of H_2^i .
 522 According to Lemma 2, the smallest eigenvalue λ_{min} of H^i should be larger than σ .
 523 The largest eigenvalue λ_{max} of H^i should be smaller than M due to $\lambda_{max} \leq \|H^i\|$.
 524 So the conditional number of H^i is smaller than $\frac{M}{\sigma}$.

525 Finally, we can find that (29) has lower bound 0. So by applying Theorem 1, we
 526 finish the proof. \square

527 **4.3. Multi-Level Strategy.** In practice, we employ the multilevel strategy. We
 528 firstly coarsen the template T and the reference R by L levels. Here, we set $T_L = T$
 529 and $R_L = R$ in the finest level and T_1 and R_1 in the coarsest level. Then we can obtain
 530 U_1 by solving our model (18) on the coarsest level. In order to give a good initial
 531 guess for the finer level, we adopt an interpolation operator on U_1 to obtain U_2^0 as the
 532 initial guess for the next level. We repeat this process and get the final registration on
 533 the finest level. A multi-level strategy has several advantages: in the coarse level, only
 534 important patterns can be considered and it is a standard technique used in order to
 535 avoid getting trapped in a meaningless local minimum; the computational speed is
 536 very fast because of less variables than on the fine level; the solution on the coarse
 537 level can be a good initial guess for the fine level.

538 The multilevel scheme representing our main algorithm is summarized below
 539 where I_H^h is an interpolation operator based on bi-linear interpolation techniques
 540 and I_h^H is a restriction operator for transferring information to a coarser level.

541 **5. Numerical Results.** In this section, we will give some numerical results to
 542 illustrate the performance of our model (18). We hope to achieve 3 aims:

- 543 1). Which choice of ϕ is the best for our model (18)?
- 544 2). We wish to compare with the current state-of-the-art methods (with codes listed
 545 for readers' benefit) in the literature for good diffeomorphic mapping:
 - 546 (a) Hyperelastic Model [7]: code from <http://www.siam.org/books/fa06/>
 - 547 (b) LDDMM [37]: code from
 548 <https://github.com/C4IR/FAIR.m/tree/master/add-ons/LagLDDMM>
 - 549 (c) Diffeomorphic Demons (DDemons) [51]: code from
 550 <http://www.insight-journal.org/browse/publication/154>
 - 551 (d) QCHR [30]; code provided by the author Dr. Kam Chu Lam.

552 All of the tests are performed on a PC with 3.40 GHz Intel(R) Core(TM) i7-4770
 553 microprocessor, and with installed memory (RAM) of 32 GB.

Algorithm 3 Multilevel Image Registration: $U \leftarrow \text{MLIR}(\alpha, \beta, U^0, T, R)$

Step 1: Compute the largest possible number of levels based on size of T, R :

$L = \text{Maxlevel}$; Define the coarsest level as level 1.

Work out the multilevel representation of given images R and T :

$R_L = R, T_L = T$;

$R_{L-1} = I_h^H R_L$,

$T_{L-1} = I_h^H T_L$; ... ;

$R_1 = I_h^H R_2, T_1 = I_h^H T_2$.

Step 2: Set the initial guess on the coarsest level:

$U_L = U^0, \quad U_j^0 = I_h^H U_{j+1}^0, \quad j = L - 1, \dots, 1.$

Step 3: Apply Algorithm 2 on the coarsest level $i = 1$ with U_1^0 :

$(U_1, \text{ID}) \leftarrow \text{GNAIRA}(\alpha, \beta, U_1^0, T_1, R_1)$;

if ID = 1 **then**

Exit this algorithm;

end if

for level $j = 2 : L$ **do**

Interpolate the solution from a coarser level $U_j^0 = I_h^H U_{j-1}$;

Apply Algorithm 2 on level j : $(U_j, \text{ID}) \leftarrow \text{GNAIRA}(\alpha, \beta, U_j^0, T_j, R_j)$;

if ID = 1 **then**

Exit this algorithm;

end if

end for

554 **3).** Most importantly, we like to test and highlight the advantages of our new model.

555 Let \mathbf{y} be the final transform obtained by a particular model for registering two
 556 given images T, R . We use the following three measures to quantify the performance
 557 of this model and use them for later comparisons:

558 (i). *Re_SSD* (the relative Sum of Squared Differences) which is given by

559 (49)
$$\text{Re_SSD} = \frac{\|T(\mathbf{y}) - R\|^2}{\|T - R\|^2};$$

560 (ii). $\min \det(J_{\mathbf{y}})$ and $\max \det(J_{\mathbf{y}})$ that are the minimum and the maximum of the
 561 Jacobian determinant of this transformation;

562 (iii). Jaccard similarity coefficient (*JSC*) as defined by

563 (50)
$$\text{JSC} = \frac{|DT_r \cap R_r|}{|DT_r \cup R_r|},$$

564 where DT_r and R_r represent respectively the segmented regions of interest
 565 (e.g. certain image feature such as an organ) in the deformed template (after
 566 registration) and the reference. Hence, *JSC* is the ratio of the intersection
 567 of DT_r and R_r to the union of DT_r and R_r [29]. *JSC* = 1 shows that a
 568 perfect alignment of the segmentation boundary and *JSC* = 0 indicates that
 569 the segmented regions have no overlap after registration.

570 Before computing *JSC*, in the first three examples below, we have employed a
 571 segmentation algorithm to segment the main features in both T and R but for
 572 the 4th example, the segmentation was manually done for both T and R .

573 In practice, we scale the intensity value of T and R to $[0, 255]$. Here, we state a strategy
 574 for choosing the parameters. For our model (18), α should be related to energy $\mathcal{D}[\mathbf{u}_0]$

575 where \mathbf{u}_0 is the initial guess for the displacement, and β should be related to α .
 576 Empirically, we set $\alpha \in [\alpha_1, \alpha_2]$, where $\alpha_1 = 0.5\mathcal{D}[\mathbf{u}_0]10^{-2}$ and $\alpha_2 = 2\mathcal{D}[\mathbf{u}_0]10^{-2}$.
 577 Respectively for $\phi = \phi_1, \phi_2, \phi_3$, we set $\beta \in [3\alpha, 5\alpha], [0.5\alpha, 2\alpha]$ and $[\alpha, 5\alpha]$. For
 578 simplicity, we denote by New 1, New 2 and New 3 the model (18) with ϕ_1, ϕ_2 and ϕ_3
 579 respectively.

580 It should be noted that a good registration result should produce a small Re_SSD ,
 581 be diffeomorphic and yield a large JSC value for a region of interest.

582 **5.1. Example 1 — Improvement over the diffusion model.** In this ex-
 583 ample, we test a pair of real medical images, X-ray Hands of resolution 128×128 .
 584 Figure 3 (a-b) show the template and the reference. We compare our model with
 585 the diffusion model and study the improvement over it. In implementation, for both
 586 models, we use a five-step multilevel strategy.

587 We conduct two experiments using different parameters:

588 **i). Fixed parameters.** Our first choice uses fixed parameters. For New 1-3, we
 589 set $\beta = 7, \beta = 1$ and $\beta = 9$ respectively, and fix $\alpha = 2$. To be fair, we also choose
 590 $\alpha = 2$ for the diffusion model. In this case, Figure 3 shows the deformed templates
 591 $T(\mathbf{y})$ from 4 models. From it, we can see that all four models can produce visually
 592 satisfactory results. To differentiate them, we have to check the quantitative measures
 593 from Table 1. We can notice that the transformation obtained by the diffusion model is
 594 non-diffeomorphic due to $\min \det(J_{\mathbf{y}}) < 0$ (i.e. mesh folded, though visually pleasing
 595 and the Re_SSD is small). Figure 4 illustrates the transform $\mathbf{y} = \mathbf{x} + \mathbf{u}$ locally at its
 596 folding point. In contrast, our New 1-3 can generate diffeomorphic transformations.

597 **ii). Optimized parameters.** The second choice uses the fine tuned parameters
 598 for the diffusion model. We tested $\alpha \in [1, 500]$ and found the smallest $\alpha = 430$ with
 599 which the diffusion model generates a diffeomorphic transformation. Then for our
 600 model, we also set $\alpha = 430$ (which is not optimized in order to favour the former)
 601 and set $\beta = 5$ for New 1-3 (to test the robustness of our model). Table 1 shows the
 602 detailed results for this second test. From it, we can see that the Re_SSD and JSC
 603 of our model are similar to the diffusion model. And the transformations obtained
 604 by New 1-3 are all diffeomorphic while the diffusion model is only diffeomorphic with
 605 the help of an optimized α . This shows that our model possesses the robustness (in
 606 the sense of not requiring optimized α) with the help of a positive β .

607 Hence, this example demonstrates that our New 1-3 are robust and can all help
 608 to get an accurate and diffeomorphic transformation.

TABLE 1

Test example 1 – Comparison of the new model (New 1-3) with the diffusion model based a fixed α and an optimized α for the latter. Clearly the latter model can produce an incorrect result if not tuned while New 1-3 are less sensitive to α with the help of β .

First Test $\alpha = 2$	Resolution	Re_SSD	$\min \det(J_{\mathbf{y}})$	$\max \det(J_{\mathbf{y}})$	JSC	time (s)
New 1	128×128	1.84%	0.0032	20.1582	99.35%	38.34
New 2	128×128	1.25%	0.0003	33.2404	99.54%	30.66
New 3	128×128	1.63%	0.0014	28.1372	99.26%	21.86
Diffusion Model	128×128	0.90%	-36.7964	72.2924	98.41%	13.42
Second Test $\alpha = 430$						
New 1	128×128	7.83%	0.1337	4.8247	98.28%	3.16
New 2	128×128	7.80%	0.1300	4.8364	98.28%	3.24
New 3	128×128	7.78%	0.1260	4.8472	98.36%	3.03
Diffusion Model	128×128	7.75%	0.0066	4.8278	98.30%	1.08

609 **5.2. Example 2 – Test of large deformation and comparison of models.**
 610 As known, if the underlying deformation is small, it is generally believed that most

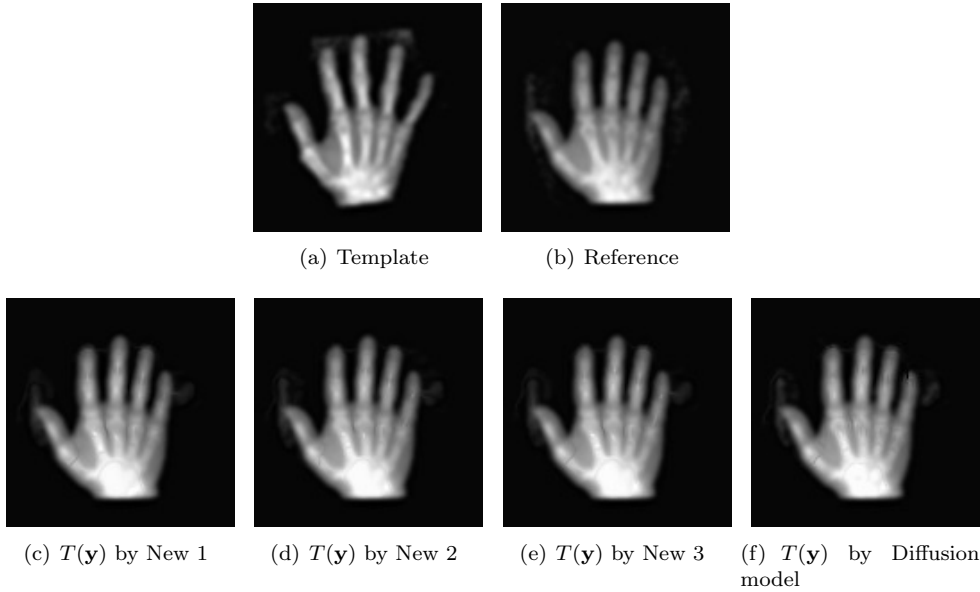


FIG. 3. Test example 1 results of Hand to Hand registration ($\alpha = 2$): in the top row, there are the template and reference. In the second row, there are the deformed templates obtained by model (18) and the diffusion model separately. Though the last column is visually fine, the transform is not correct – see Table 1.

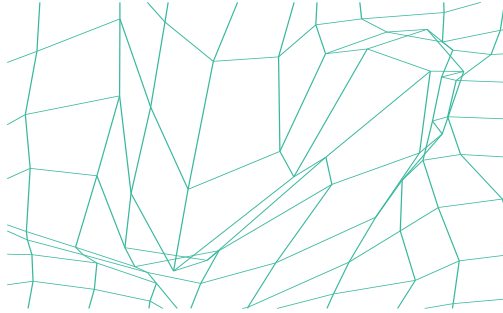


FIG. 4. Zooming in the transformation (obtained by the diffusion model) where there is folding.

611 models can deliver diffeomorphic transformations. This belief is true if one keeps
 612 increasing α , which in turn compromises the registration quality by resulting in an
 613 increase in Re_SSD (as seen in 2 tests of α in Example 1 where the larger $\alpha = 430$
 614 achieves diffeomorphism for diffusion with a worse Re_SSD value).

615 Therefore to test the capability of a registration model, we need to take an exam-
 616 ple that requires large deformation. To this end, we consider Example 2 – a classic
 617 synthetic example consisting of a **Disc** and a **C shape** of resolution 128×128 as
 618 shown in Figure 5 (a-b). We compare our 3 models (New 1-3) with 5 other mod-
 619 els: the hyperelastic model, LDDMM, DDemons, QCHR and the diffusion model in
 620 registration quality and performance. For this example, we use a five-step multilevel
 621 strategy for our model, the hyperelastic model and the diffusion model. For LDDMM
 622 and QCHR, we use a three-step multilevel strategy. We use a one-step multilevel

TABLE 2
Test example 2 – Comparison of the new model (New 1-3) with 5 other models.

	Resolution	Re_SSD	$\min \det(J_{\mathbf{y}})$	$\max \det(J_{\mathbf{y}})$	JSC	time (s)
New 1	128×128	0.06%	0.0042	22.4	95.57%	7.00
New 2	128×128	0.07%	0.0012	19.5	95.84%	10.10
New 3	128×128	0.06%	0.0034	22.6	95.37%	3.93
Hyperelastic Model	128×128	0.81%	0.2426	5.9	94.84%	1.84
LDDMM	128×128	0.06%	0.1175	12.0	96.00%	9.16
DDemons	128×128	1.71%	1.3×10^{-7}	8.2	92.69%	57.27
QCHR Model	128×128	7.69%	0.0255	57.4	85.36%	141.86
Diffusion Model	128×128	1.25%	-10.1612	162.5	94.21%	0.31

623 strategy for DDemons as we found that multilevel does not improve the results.

624 Following our stated strategy for choosing the parameter for our model, we set
 625 $\beta = 80, 120, 100$ for New 1-3 respectively and fix $\alpha = 70$. To be consistent, we also set
 626 $\alpha = 70$ for the diffusion model. For the hyperelastic model, LDDMM and QCHR, we
 627 set respectively $\{\alpha_l = 100, \alpha_s = 0, \alpha_v = 18\}$, $\alpha = 400$ and $\{\alpha = 0.1, \beta = 1\}$ as used in
 628 the literature [7, 37, 30] for the same example. For the parameters of DDemons, we
 629 tried to optimize the parameters $\{\sigma_s, \sigma_g\}$ in the domain $[0.5, 5] \times [0.5, 5]$ and took the
 630 optimal choice $\{\sigma_s = 1.5, \sigma_g = 3.5\}$.

631 We now present the comparative results. Figure 5 (c-j) show that except for
 632 the diffusion model, all the other models can produce the accepted registered results.
 633 Especially, our model and LDDMM are slightly better than the hyperelastic model,
 634 DDemons and QCHR. It is pleasing to see that the new model produces equally
 635 good results for this challenging example. From Table 2, we see that our New 1-3,
 636 hyperelastic model, LDDMM, DDemons and QCHR produce $\min \det(J_{\mathbf{y}}) > 0$ i.e.
 637 the transformations obtained by these five models are diffeomorphic but the diffusion
 638 model fails again with $\min \det(J_{\mathbf{y}}) < 0$.

639 Because New 1-3 are motivated by the QCHR model, we now discuss the results
 640 about these two types of models. On the one hand, according to Table 2, we can
 641 find that our model takes less time. This is because, as we have mentioned, the
 642 algorithm for QCHR needs to solve alternatively two subproblems (including several
 643 linear systems) in each iteration. Its convergence cannot be guaranteed. However,
 644 our model only needs to solve one linear system in each iteration. In addition, we
 645 employ the Gauss-Newton method which can be superlinearly convergent under the
 646 appropriate conditions. As we have also remarked, the QCHR algorithm can have
 647 convergence problems. This is now illustrated in Figure 6 where we plot the relative
 648 residual of our model (New 3) and the relative residual of QCHR. We observe that
 649 New 3 decreases to below 10^{-2} though not monotonically, but the relative residual of
 650 QCHR does not decrease and is over 0.1.

651 On the other hand, we can compare the obtained solutions' quality by checking
 652 the energy functionals. Using the same QCHR functional, the QCHR solution for
 653 Example 2 has the value 1042 while the transformation obtained by New 3 gives the
 654 value 147 which is much smaller. This indicates that the result obtained by the QCHR
 655 algorithm is not accurate. This is consistent with the fact that the Re_SSD and JSC
 656 of New 3 are also better than QCHR. Both discussions reach the same conclusion:
 657 the QCHR algorithm cannot obtain the minimizer of the original QCHR functional.

658 **5.3. Example 3 – Comparison of models for a challenging test.** Here,
 659 we illustrate the fact that area preservation between images can become unnecessary
 660 and trying to enforce it (as in the hyperelastic model) can fail to register an image.
 661 We choose the particular template and reference images, as shown in Figure 7 (a-b),

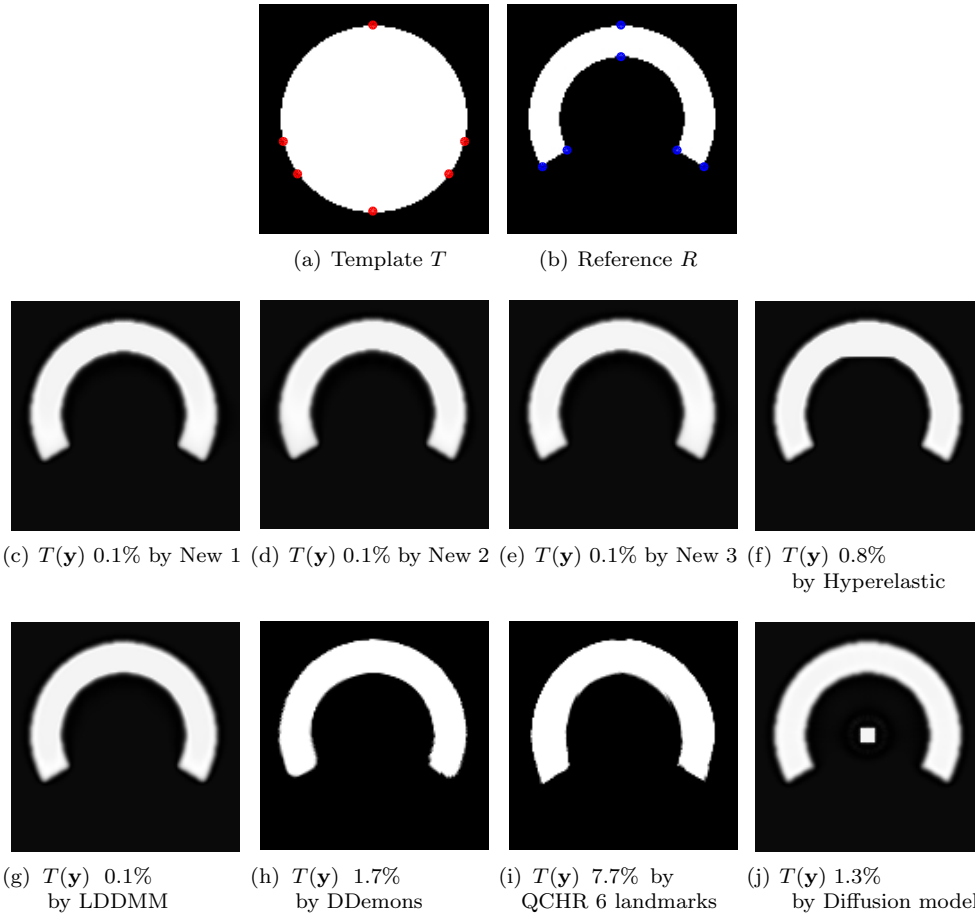


FIG. 5. Test example 2 results of Disc to C. The percentage value shows Re_SSD error. In the top row, there are the template and the reference. In the second and third row, there are the deformed templates obtained by New 1-3 and 5 other models separately. The landmarks in the template and reference are only used for QCHR and the last result (j) by Diffusion is evidently not correct.

662 having significantly different areas in their main features – here the area of ‘Disc’
 663 is much larger than ‘C’. The resolution of the images is 512×512 . We test the
 664 performance of New 1-3 and other models. In this example, we use a seven-step
 665 multilevel strategy for New 1-3, the hyperelastic model and the diffusion model. For
 666 LDDMM and QCHR, we use a five-step multilevel strategy. We use a single level for
 667 DDemons (since multilevels do not help).

668 In choosing the parameters for all the models to register this example, we first
 669 follow our strategy to set $\beta = 250, 50, 100$ for New 1-3 respectively and fix $\alpha = 50$.
 670 To be consistent, we also set $\alpha = 50$ for the diffusion model. For the hyperelastic
 671 model, we also set $\alpha_l = 50$ because it contains the diffusion term, and take $\alpha_s = 0$.
 672 For the third parameter α_v in the hyperelastic model, we test it in the range $[55, 150]$
 673 and choose its optimal value $\alpha_v = 75$. For LDDMM and QCHR, we set the default
 674 value $\alpha = 400$ and $\{\alpha = 0.1, \beta = 1\}$ as the previous example. For the parameters of
 675 DDemons, we test the parameters $\{\sigma_s, \sigma_g\}$ in the domain $[0.5, 5] \times [0.5, 5]$ and choose

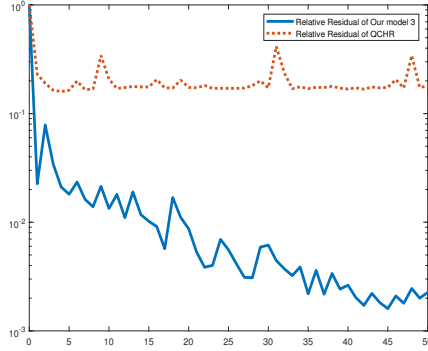


FIG. 6. *Example 2 Relative Residual of New 3 and QCHR: The solid line indicates the relative residual of New 3. And the dot line shows the relative residual of the second subproblem in QCHR. Here, we can find that in the same 50 iterations, the relative residual of New 3 is decreasing to below 10^{-2} , however the relative residual of QCHR is not decreasing and over 0.1. Hence, the convergence of the algorithm for QCHR can not be guaranteed.*

676 its optimal choice $\{\sigma_s = 2, \sigma_g = 5\}$. Hence we would expect the hyperelastic model
 677 and DDemons to perform well.

678 The test results for Example 3 are presented in Table 3 and Figure 7. Although
 679 all models except for the diffusion model produce diffeomorphic transformations, we
 680 can see visually that only 3 models (our New 2-3 and LDDMM) produce acceptable
 681 results, also confirmed by the table:

- 682 • The badly deformed template generated by our New 1 shows that the model
 683 lacks robustness;
- 684 • The hyperelastic model, though producing a diffeomorphic transform, fails
 685 (despite using an optimized parameter) because this model including a regu-
 686 larization term $(\det(J_{\mathbf{y}}) - 1)^4 / (\det(J_{\mathbf{y}}))^2$ tends to preserve area. If we do
 687 not optimize parameters for the hyperelastic model, our tests show that its
 688 results are even worse.
- 689 • In the previous example, we have pointed out that QCHR needs more compu-
 690 ting time and, from Table 3, we see that the time for QCHR is about 20
 691 times as long as our New 3;
- 692 • The DDemons is trapped in a local minimum and its cpu time is also excessive
 693 (> 5000 seconds). We also try to apply a multilevel strategy to DDemons,
 694 but for this example the result is not satisfied. The *Re_SSD*, *JSC* and cpu
 695 time of our New 3 are all slightly better than the second best LDDMM;
- 696 • Both Tables 2 and 3 show that the diffusion model produces solutions having
 697 a negative Jacobian (folding) which might be viewed non-physical; this model
 698 is included only for reference.

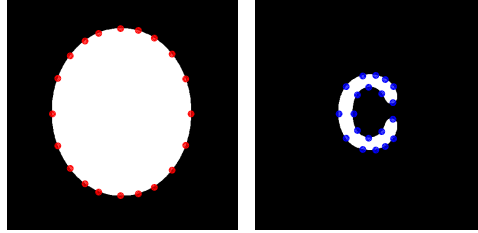
699 Hence, our model has advantages over other models for large deformation registrations
 700 not requiring preserving area.

701 We now give 2 remarks on comparing New 3 (or New 2) and QCHR. As remarked,
 702 QCHR regularizes the Beltrami coefficient only and the landmarks supplied to QCHR
 703 can severely affect the results while our model regularizes the deformation rather than
 704 Beltrami coefficient. Both points can be further tested below.

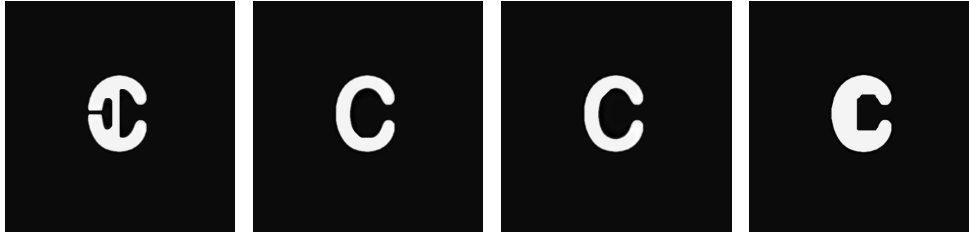
705 (i). On the first point, regularizing the Beltrami coefficient only leads to smooth
 706 Beltrami coefficient. To compare smoothness of solutions by New 3 and QCHR, we
 707 compute three smoothness measures $\|\nabla \mathbf{u}\|_{L^2}$, $\|\mu(\mathbf{y})\|_{L^2}$, $\|\nabla \mu(\mathbf{y})\|_{L^2}$ and present them

TABLE 3
Example 3 – Comparison of the new model (New 1-3) with 5 other models.

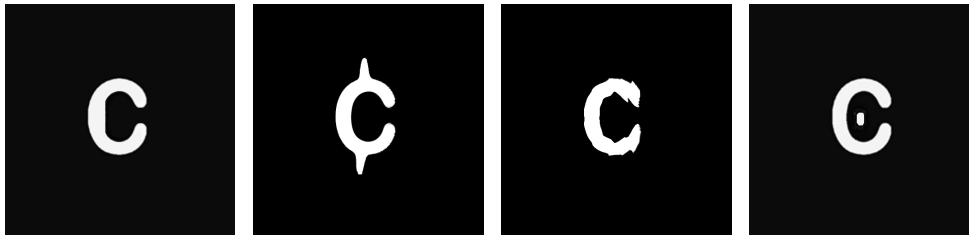
	Resolution	Re_SSD	$\min \det(J_{\mathbf{y}})$	$\max \det(J_{\mathbf{y}})$	JSC	time (s)
New 1	512×512	3.06%	0.0328	38.2272	78.93%	402.87
New 2	512×512	0.08%	0.0035	64.4950	97.84%	281.95
New 3	512×512	0.07%	0.0064	60.1743	97.82%	202.17
Hyperelastic Model	512×512	3.85%	0.4895	7.0781	75.49%	46.16
LDDMM	512×512	0.41%	0.0184	40.2544	95.05%	218.32
DDemons	512×512	2.83%	9.6×10^{-6}	34.8529	80.56%	> 5000
QCHR Model	512×512	2.03%	0.0207	4.4744	84.24%	4716.7
Diffusion Model	512×512	0.52%	-38.8337	286.3411	94.68%	5.52



(a) Template T (b) Reference R



(c) $T(\mathbf{y})$ by our model 1 (d) $T(\mathbf{y})$ by our model 2 (e) $T(\mathbf{y})$ by our model 3 (f) $T(\mathbf{y})$ by Hyperelastic model



(g) $T(\mathbf{y})$ by LDDMM (h) $T(\mathbf{y})$ by DDemons (i) $T(\mathbf{y})$ by QCHR with 20 pairs of landmarks (j) $T(\mathbf{y})$ by Diffusion model

FIG. 7. *Example 3 results of a large Disc to small letter C : in the top row, there are the template and reference. In the second and third row, there are the deformed templates obtained by model (18) and other models separately. The landmarks in the template and reference are only used for QCHR.*

708 in Table 4. Clearly the table indicates that QCHR does generate a smoother Beltrami
 709 coefficient than our model New 3 for both Examples 2-3, not a smoother deformation
 710 field. Hence, the model which only regularizes the Beltrami coefficient rather than
 711 the deformation is not sufficient to produce an accurate deformed template.

712 (ii). On the second point, we now illustrate the importance of landmarks for

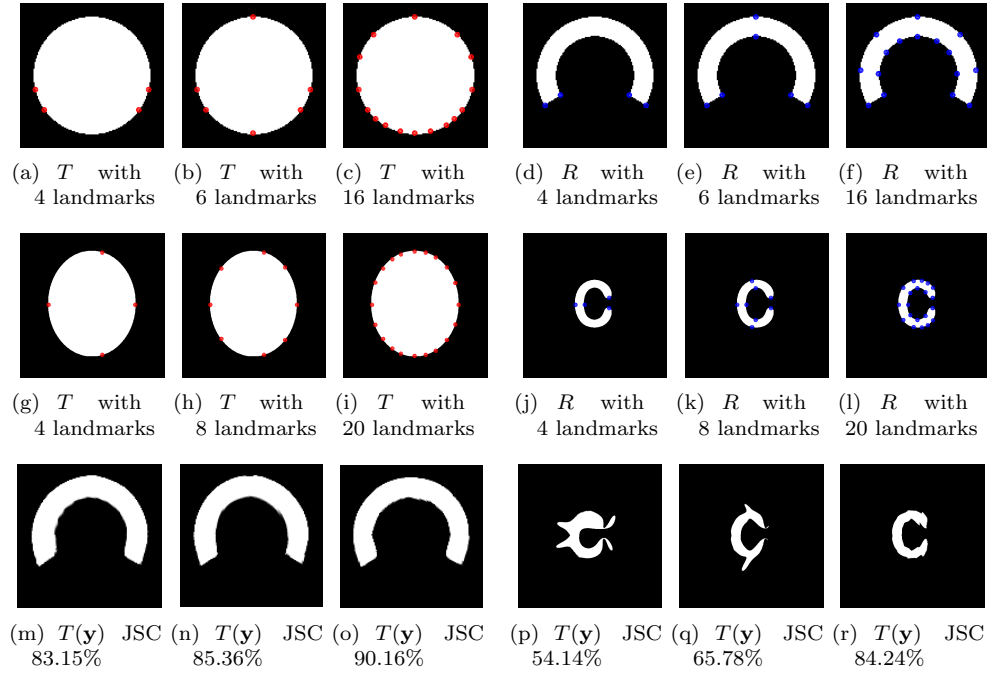


FIG. 8. Tests of QCHR with different landmarks: Example 2 (row 1) and Example 3 (row 2). On the left 3 columns of row 3, we show the registered templates for row 1. On the right 3 columns of row 3, we show the registered templates for row 2. Here, we can see that the accuracy of QCHR improves with the increase of landmarks.

TABLE 4

Comparison of smoothness measures for solutions obtained by New 3 and QCHR. The Beltrami coefficient μ obtained by QCHR is smoother than New 3 and the displacement \mathbf{u} obtained by New 3 is smoother than QCHR.

	$\ \nabla \mathbf{u}\ _{L^2}$	$\ \mu(\mathbf{y})\ _{L^2}$	$\ \nabla \mu(\mathbf{y})\ _{L^2}$	Re-SSD
Example 2				
QCHR with 16 pairs of landmarks	2.1099	0.6930	0.2782	4.90%
New 3	1.6155	0.5024	0.2800	0.06%
Example 3				
QCHR with 20 pairs of landmarks	1.5366	0.5853	0.0868	2.03%
New 3	1.3913	0.3352	0.1090	0.07%

713 QCHR although for other problems the model can yield good results without any
 714 landmarks. Fig. 8 shows three sets of increasing number of landmarks for Examples
 715 2-3. We observe that more landmarks lead to better results in terms of *JSC* values.

716 As a final comparison of New 3 with LDDMM and QCHR, Figure 9 plots the
 717 magnitudes of the Jacobian determinants of their transformations. It can be seen
 718 that New 3 and LDDMM give a similar pattern but both are different from QCHR.

719 **5.4. Example 4— Comparison of the new model with other models.** In
 720 the final test, we test a pair of anonymized CT images in resolution 512×512 from
 721 the Royal Liverpool University Hospital. Figure 10 (a-b) show the template and the
 722 reference. The template was taken in September 2016 and the reference was taken in
 723 May 2016. We want to compare the changes of our interested regions of abdominal
 724 aortic aneurysm with stents inserted inside them (with cross sections shown as two

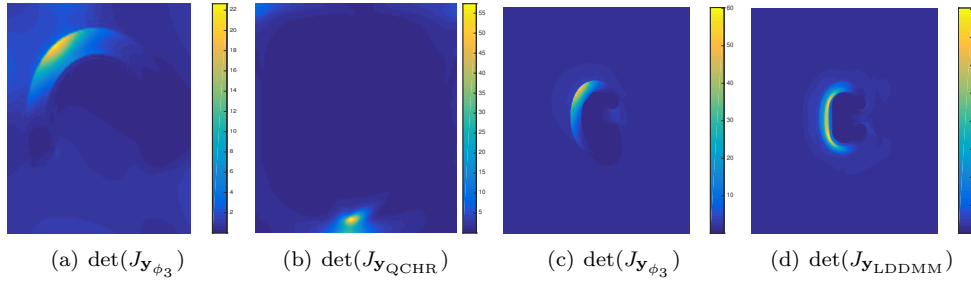


FIG. 9. Example 3 Illustration of Jacobian determinants of the transformations obtained by our New 3, QCHR and LDDMM for Example 2 (left two plots) and Example 3 (right two plots). Note all values are positive (since all models are diffeomorphic) and New 3 has similar distributions to LDDMM, different from QCHR.

TABLE 5
Example 4 – Comparison of New 1-3 with 5 other models

	Resolution	Re_SSD	$\min \det(J_{\mathbf{y}})$	$\max \det(J_{\mathbf{y}})$	JSC
New 1	512×512	4.75%	0.0124	52.6802	94.19%
New 2	512×512	3.49%	0.0068	46.6383	94.39%
New 3	512×512	3.47%	0.0051	49.9309	95.34%
Hyperelastic Model	512×512	4.44%	0.4181	3.6192	93.51%
LDDMM	512×512	5.18%	0.0319	20.8164	93.79%
DDemons	512×512	18.89%	0.1846	2.6309	87.40%
QCHR Model	512×512	26.71%	0.0481	16.2555	85.68%
Diffusion Model	512×512	10.02%	0.0342	7.3450	93.65%

725 while ‘circles’ in images in Figure 10 (a-b)) during these 4 months. In addition, the
 726 interested region is used to compute JSC . The small white region on top of the
 727 images helps us to identify the correct slice to compare.

728 Here, following the previous example, we use the same multilevel strategy: a
 729 seven-step multilevel strategy for our model, the hyperelastic model and the diffu-
 730 sion model, a five-step multilevel strategy for LDDMM and QCHR and a one-step
 731 multilevel strategy for DDemons.

732 Following our strategy for choosing the parameter of our model, we set $\alpha = 20$ and
 733 set $\beta = 100, 40, 75$ with New 1-3 respectively. For the diffusion model and LDDMM,
 734 we test α from $[100, 2000]$ and set the optimal value 1300 and 500 respectively. For
 735 the hyperelastic model, we set $\{\alpha_l = 20, \alpha_s = 0, \alpha_v = 50\}$. We use the default value
 736 $\{\alpha = 0.1, \beta = 1\}$ for QCHR. For the parameters of DDemons, we test the parameters
 737 $\{\sigma_s, \sigma_g\}$ in the domain $[0.5, 5] \times [0.5, 5]$ and choose $\{\sigma_s = 4, \sigma_g = 4.5\}$.

738 With the optimized parameters, all the models in this example generate diffeo-
 739 morphic transformations as seen from Table 5. DDemons and QCHR for this example
 740 are not as good as other models because they give worse Re_SSD and JSC . A worse
 741 JSC means the interested regions obtained by these two methods have significant
 742 differences from the reference (Figure 10 (h-i)). The diffusion model obtains a good
 743 JSC , however its deformed template is a bit far (overall) from the reference (since
 744 $Re_SSD = 10.02\%$). The other 2 models (Hyperelastic, LDDMM) generate good
 745 Re_SSD and JSC . However, our models produce the lowest Re_SSD and the best
 746 JSC . Hence, for this example of real images, our model is competitive to the state-
 747 of-the-art methods. Though there is broad agreement between Re_SSD and JSC ,
 748 one has to combine with segmentation models to ensure the strict agreement.

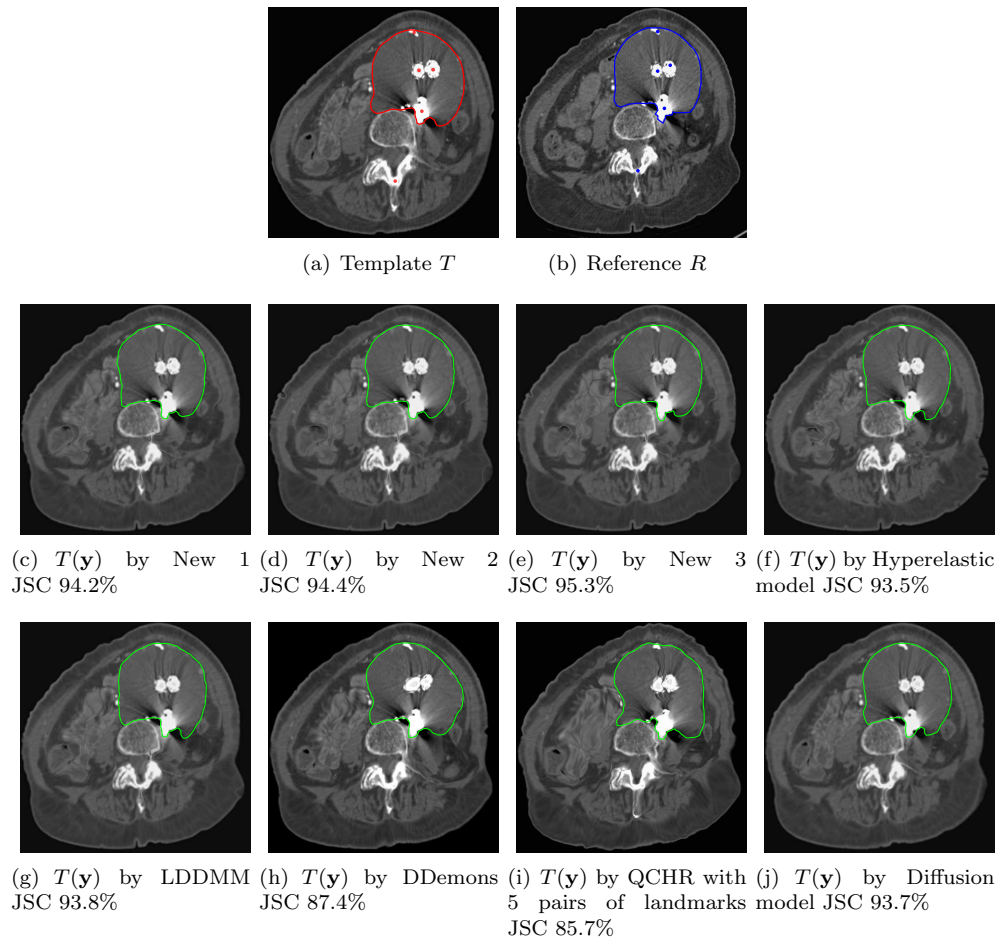


FIG. 10. *Example 4 – Registration results of a pair of CT images: the template T and the reference R in the top row. The contours show the regions of interest. In the second and third rows, we show the deformed templates obtained by 8 models. The 5 landmarks in the template and the reference are only used by QCHR.*

749 **REMARK.** *According to the above four examples, our New 1 is not robust while*
 750 *New 2-3 can both generate accurate and diffeomorphic transformations. However, we*
 751 *recommend New 3 as the first choice because of the least computing time and the best*
 752 *quality, and New 2 as the second choice.*

753 *We also test these four examples with the Dirichlet boundary condition. Similar*
 754 *results for Examples 1 and 4 are obtained. However, for Examples 2 and 3, the trans-*
 755 *formations would be different since the boundary is better modeled by the Neumann's*
 756 *condition.*

757 **6. Conclusions.** Controlling mesh folding is a key issue in image registration
 758 models to ensure local invertibility. Many existing models either do not impose any
 759 further controls on the underlying transformation beyond smoothness (so potentially
 760 generating unrealistic or non-physical transforms or mapping) or impose a *direct* (often
 761 strongly biased e.g. towards area or volume preservation) control on some explicit

762 function of the measure $\det(J_{\mathbf{y}})$. This paper introduces a novel, unbiased and robust
 763 regularizer which is reformulated from Beltrami coefficient framework to ensure a
 764 diffeomorphic transformation. Moreover we find that a direct approach (our New 1)
 765 from this Beltrami reformulation provides an alternative but less competitive method
 766 but further refinements (especially our New 3) of this new regularizer can give rise
 767 to more robust models than the existing methods. We highly recommend our model
 768 New 3 i.e. (18) with $\phi = \phi_3$.

769 In designing optimization methods for solving the resulting highly nonlinear vari-
 770 ational model, we give a suitable approximation of the exact Hessian matrix which is
 771 necessary to derive a convergent iterative method. Our test results can show that the
 772 new model (New 1-3, especially New 3) is competitive with the state-of-the-art mod-
 773 els. The main advantage lies in robustness. Our future work will include extensions
 774 to 3D problems, multi-modality models and development of faster iterative solvers.

775 **Appendix A. Computation of matrices A and G in §4.1.2.** Set $B =$
 776 $I_2 \otimes I_{n+1} \otimes \partial_n^{1,h} \in \mathbb{R}^{2n(n+1) \times 2(n+1)^2}$, $C = I_2 \otimes \partial_n^{1,h} \otimes I_{n+1} \in \mathbb{R}^{2n(n+1) \times 2(n+1)^2}$,

$$777 \quad \partial_n^{1,h} = \frac{1}{h^2} \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & \dots & \dots & \dots & \\ & & & -1 & 1 \\ & & & & -1 & 1 \end{bmatrix} \in \mathbb{R}^{n,n+1}, \quad A = \begin{bmatrix} B \\ C \end{bmatrix} \in \mathbb{R}^{4n(n+1) \times 2(n+1)^2},$$

778 where \otimes denotes a Kronecker product. To represent the difference between interior
 779 and boundary pixels, we need to introduce a diagonal matrix

$$780 \quad (51) \quad G = \begin{bmatrix} G_1 & 0 & 0 & 0 \\ 0 & G_2 & 0 & 0 \\ 0 & 0 & G_1 & 0 \\ 0 & 0 & 0 & G_2 \end{bmatrix} \in \mathbb{R}^{4n(n+1) \times 4n(n+1)},$$

781 where G_1 and G_2 are diagonal matrices. For G_1 , $G_{1_{i+1+jn, i+1+jn}} = 1$ if $0 \leq i \leq n -$
 782 $1, 1 \leq j \leq n - 1$ or $\frac{1}{2}$ if $0 \leq i \leq n - 1, j = 0, n$. Similarly, for G_2 , $G_{2_{i+1+j(n+1), i+1+j(n+1)}} =$
 783 1 if $1 \leq i \leq n - 1, 0 \leq j \leq n - 1$ or $\frac{1}{2}$ if $i = 0, n, 0 \leq j \leq n - 1$.

784 **Appendix B. Computation of the vector $\vec{r}(U)$ in §4.1.3.** We demonstrate
 785 how to build the linear interpolation \mathbf{L} in $\Delta V_1 V_2 V_5$, in Figure 2.

786 First of all, denote the 3 vertices of this triangle by $V_1 = \mathbf{x}^{1,1}$, $V_2 = \mathbf{x}^{2,1}$ and
 787 $V_5 = \mathbf{x}^{1.5,1.5}$. Set $\mathbf{L}(V_1) = (u_1^{1,1}, u_2^{1,1})$, $\mathbf{L}(V_2) = (u_1^{2,1}, u_2^{2,1})$ at the vertex pixels, and
 788 $\mathbf{L}(V_5) = (u_1^{1.5,1.5}, u_2^{1.5,1.5})$ at the cell centre (approximated values). Here the linear
 789 approximations are $\mathbf{L}(x_1, x_2) = (a_1 x_1 + a_2 x_2 + a_3, a_4 x_1 + a_5 x_2 + a_6)$.

790 After substituting V_1, V_2 and V_5 into \mathbf{L} , we get

$$791 \quad \begin{pmatrix} x_1^1 - x_1^{1.5} & x_2^1 - x_2^{1.5} \\ x_1^2 - x_1^{1.5} & x_2^2 - x_2^{1.5} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} u_1^{1,1} - u_1^{1.5,1.5} \\ u_2^{2,1} - u_2^{1.5,1.5} \end{pmatrix},$$

$$792 \quad \begin{pmatrix} x_1^1 - x_1^{1.5} & x_2^1 - x_2^{1.5} \\ x_1^2 - x_1^{1.5} & x_2^2 - x_2^{1.5} \end{pmatrix} \begin{pmatrix} a_4 \\ a_5 \end{pmatrix} = \begin{pmatrix} u_2^{1,1} - u_2^{1.5,1.5} \\ u_1^{2,1} - u_1^{1.5,1.5} \end{pmatrix}.$$

794 Then

$$795 \quad (52) \quad \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \frac{1}{\det} \begin{pmatrix} x_2^1 - x_2^{1.5} & -x_2^1 + x_2^{1.5} \\ -x_1^2 + x_1^{1.5} & x_1^1 - x_1^{1.5} \end{pmatrix} \begin{pmatrix} u_1^{1,1} - u_1^{1.5,1.5} \\ u_2^{2,1} - u_2^{1.5,1.5} \end{pmatrix},$$

796

$$(53) \quad \begin{pmatrix} a_4 \\ a_5 \end{pmatrix} = \frac{1}{\det} \begin{pmatrix} x_2^1 - x_2^{1.5} & -x_2^1 + x_2^{1.5} \\ -x_1^2 + x_1^{1.5} & x_1^1 - x_1^{1.5} \end{pmatrix} \begin{pmatrix} u_2^{1,1} - u_2^{1.5,1.5} \\ u_2^{2,1} - u_2^{1.5,1.5} \end{pmatrix},$$

$$\text{where } \det = \begin{vmatrix} x_1^1 - x_1^{1.5} & x_2^1 - x_2^{1.5} \\ x_1^2 - x_1^{1.5} & x_2^1 - x_2^{1.5} \end{vmatrix}.$$

According to (52) and (53), we can formulate two matrices $D1 \in \mathbb{R}^{4n^2 \times (n+1)^2}$ and $D2 \in \mathbb{R}^{4n^2 \times (n+1)^2}$ such that

$\mathbf{a}_1 - \mathbf{a}_5 = [D1, -D2]U = A_1U \in \mathbb{R}^{4n^2 \times 1}$, $\mathbf{a}_4 + \mathbf{a}_2 = [D2, D1]U = A_2U \in \mathbb{R}^{4n^2 \times 1}$, and

$\mathbf{a}_1 + \mathbf{a}_5 = [D1, D2]U = A_3U \in \mathbb{R}^{4n^2 \times 1}$, $\mathbf{a}_4 - \mathbf{a}_2 = [D2, -D1]U = A_4U \in \mathbb{R}^{4n^2 \times 1}$.

Here, $\mathbf{a}_\theta = (a_\theta^1, \dots, a_\theta^{4n^2})^T$, $\theta = 1, 2, 4, 5$, where $a_\theta^l = a_\theta^{i,j,k}$ and $l = (k-1)n^2 + (j-1)n + i$.

Next using the Hadamard product \odot , we get a compact form for

$$(54) \quad \begin{cases} \bar{\mathbf{r}}^1(U) &= A_1U \odot A_1U + A_2U \odot A_2U, \\ \bar{\mathbf{r}}^2(U) &= 1/((A_3U + 2) \odot (A_3U + 2) + A_4U \odot A_4U), \\ \bar{\mathbf{r}}(U) &= \bar{\mathbf{r}}^1 \odot \bar{\mathbf{r}}^2 \in \mathbb{R}^{4n^2 \times 1}. \end{cases}$$

Appendix C. Computing the gradient and approximated Hessian of the term (37). Here, as an example, we set $n = 2$ and $\phi = \phi_1$ to compute the gradient and approximated Hessian of the discretized Beltrami term (37).

Because of $n = 2$, we have

$$U = (u_1^{0,0}, \dots, u_1^{2,0}, \dots, u_1^{0,2}, \dots, u_1^{2,2}, u_2^{0,0}, \dots, u_2^{2,0}, \dots, u_2^{0,2}, \dots, u_2^{2,2})^T \in \mathbb{R}^{18 \times 1}.$$

From (52)-(53), we can formulate two matrices $D1, D2 \in \mathbb{R}^{16 \times 9}$ respectively by:

$$D1 = \begin{bmatrix} -2 & 2 & & & & & & & \\ & -2 & 2 & & & & & & \\ & & & -2 & 2 & & & & \\ & & & & -2 & 2 & & & \\ -1 & 1 & -1 & 1 & & & & & \\ & -1 & 1 & -1 & 1 & & & & \\ & & -1 & 1 & -1 & 1 & & & \\ & & & -1 & 1 & -1 & 1 & & \\ & & & & -2 & 2 & & & \\ & & & & & -2 & 2 & & \\ & & & & & & -2 & 2 & \\ -1 & 1 & -1 & 1 & & & & & \\ & -1 & 1 & -1 & 1 & & & & \\ & & -1 & 1 & -1 & 1 & & & \\ & & & -1 & 1 & -1 & 1 & & \end{bmatrix}, \quad D2 = \begin{bmatrix} -1 & -1 & 1 & 1 & & & & & \\ & -1 & -1 & 1 & 1 & & & & \\ & & & -1 & -1 & 1 & 1 & & \\ & & & & -1 & -1 & 1 & 1 & \\ & & -2 & & 2 & & & & \\ & & & -2 & & 2 & & & \\ & & & & -2 & & 2 & & \\ -1 & -1 & 1 & 1 & & & & & \\ & -1 & -1 & 1 & 1 & & & & \\ & & & -1 & -1 & 1 & 1 & & \\ -2 & & & -1 & -1 & 1 & 1 & & \\ & -2 & & & 2 & & & & \\ & & -2 & & 2 & & & & \\ & & & -2 & -2 & 2 & & & \\ & & & & -2 & & 2 & & \end{bmatrix}.$$

Then we can build A_1, A_2, A_3 and A_4 and compute $\bar{\mathbf{r}}^1, \bar{\mathbf{r}}^2$ and $\bar{\mathbf{r}}$ by (54). According to (39), we have $d\bar{\mathbf{r}} \in \mathbb{R}^{16 \times 18}$.

When $\phi(v) = \phi_1(v)$, we have $\phi'_1(v) = \frac{2}{(v-1)^3}$, $\phi''_1(v) = \frac{6}{(v-1)^4}$ and so $d\phi(\bar{\mathbf{r}}) = (\frac{2}{(\bar{\mathbf{r}}_1-1)^3}, \dots, \frac{2}{(\bar{\mathbf{r}}_{16}-1)^3})^T$ in (38). In (40) the i th diagonal element $[d^2\phi(\bar{\mathbf{r}})]_{ii} = \frac{6}{(\bar{\mathbf{r}}_i-1)^4}$, $1 \leq i \leq 16$. Similarly when $\phi(v) = \phi_2$, $d\phi(\bar{\mathbf{r}}) = (\frac{-\bar{\mathbf{r}}_1-1}{(\bar{\mathbf{r}}_1-1)^2}, \dots, \frac{-\bar{\mathbf{r}}_{16}-1}{(\bar{\mathbf{r}}_{16}-1)^2})^T$ and $[d^2\phi(\bar{\mathbf{r}})]_{ii} = \frac{2\bar{\mathbf{r}}_i+4}{(\bar{\mathbf{r}}_i-1)^4}$. When $\phi(v) = \phi_3$, $d\phi(\bar{\mathbf{r}}) = (\frac{-2\bar{\mathbf{r}}_1}{(\bar{\mathbf{r}}_1-1)^3}, \dots, \frac{-2\bar{\mathbf{r}}_{16}}{(\bar{\mathbf{r}}_{16}-1)^3})^T$ and $[d^2\phi(\bar{\mathbf{r}})]_{ii} = \frac{4\bar{\mathbf{r}}_i+2}{(\bar{\mathbf{r}}_i-1)^4}$.

Hence, we can get d_3 in (38) and \hat{H}_3 in (40).

- 821 [1] L. V. AHLFORS AND C. J. EARLE, *Lectures on Quasiconformal Mappings*, van Nostrand Prince-
 822 ton, 1966.
- 823 [2] R. BARRETT, M. W. BERRY, T. F. CHAN, J. DEMMEL, J. DONATO, J. DONGARRA, V. ELJKHOUT,
 824 R. POZO, C. ROMINE, AND H. VAN DER VORST, *Templates for the solution of linear systems:
 825 building blocks for iterative methods*, vol. 43, SIAM, 1994.
- 826 [3] M. F. BEG, M. I. MILLER, A. TROUVÉ, AND L. YOUNES, *Computing large deformation metric
 827 mappings via geodesic flows of diffeomorphisms*, International journal of computer vision,
 828 61 (2005), pp. 139–157.
- 829 [4] L. BERS, *Quasiconformal mappings, with applications to differential equations, function theory
 830 and topology*, Bulletin of the American Mathematical Society, 83 (1977), pp. 1083–1100.
- 831 [5] C. BROIT, *Optimal Registration of Deformed Images*, PhD thesis, University of Pennsylvania,
 832 USA, 1981.
- 833 [6] L. G. BROWN, *A survey of image registration techniques*, ACM Computing Surveys, 24 (1992),
 834 pp. 325–376.
- 835 [7] M. BURGER, J. MODERSITZKI, AND L. RUTHOTTO, *A hyperelastic regularization energy for
 836 image registration*, SIAM Journal on Scientific Computing, 35 (2013), pp. 132–148.
- 837 [8] Y. CHEN AND X. YE, *Inverse consistent deformable image registration*, in The Legacy of Alladi
 838 Ramakrishnan in the Mathematical Sciences, Springer, 2010, pp. 419–440.
- 839 [9] G. E. CHRISTENSEN, *Deformable shape models for anatomy*, PhD thesis, Washington University
 840 Saint Louis, USA, 1994.
- 841 [10] G. E. CHRISTENSEN, R. D. RABBITT, AND M. I. MILLER, *Deformable templates using large
 842 deformation kinematics*, IEEE transactions on image processing, 5 (1996), pp. 1435–1447.
- 843 [11] N. CHUMCHOB AND K. CHEN, *A robust affine image registration method*, International Journal
 844 of Numerical Analysis and Modeling, 6 (2009), pp. 311–334.
- 845 [12] ———, *A variational approach for discontinuity-preserving image registration*, East-West Jour-
 846 nal of Mathematics, (2010), pp. 266–282.
- 847 [13] N. CHUMCHOB AND K. CHEN, *A robust multigrid approach for variational image registration
 848 models*, Journal of Computational and Applied Mathematics, 236 (2011), pp. 653–674.
- 849 [14] N. CHUMCHOB, K. CHEN, AND C. BRITO, *A fourth-order variational image registration model
 850 and its fast multigrid algorithm*, Multiscale Modeling & Simulation, 9 (2011), pp. 89–128.
- 851 [15] P. DUPUIS, U. GRENANDER, AND M. I. MILLER, *Variational problems on flows of diffeomor-
 852 phisms for image matching*, Quarterly of applied mathematics, (1998), pp. 587–600.
- 853 [16] B. FISCHER AND J. MODERSITZKI, *Fast diffusion registration*, Contemporary Mathematics, 313
 854 (2002), pp. 117–128.
- 855 [17] ———, *Curvature based image registration*, Journal of Mathematical Imaging and Vision, 18
 856 (2003), pp. 81–85.
- 857 [18] ———, *A unified approach to fast image registration and a new curvature based registration
 858 technique*, Linear Algebra and Its Applications, 380 (2004), pp. 107–124.
- 859 [19] C. FROHN-SCHAUF, S. HENN, AND K. WITSCH, *Multigrid based total variation image registra-
 860 tion*, Computing and Visualization in Science, 11 (2008), pp. 101–113.
- 861 [20] F. P. GARDINER AND N. LAKIC, *Quasiconformal Teichmüller Theory*, no. 76, American Math-
 862 ematical Soc., 2000.
- 863 [21] A. A. GOSHTASBY, *2-D and 3-D Image Registration: for Medical, Remote Sensing, and Indus-
 864 trial Applications*, John Wiley & Sons, 2005.
- 865 [22] ———, *Image Registration: Principles, Tools and Methods*, Springer, 2012.
- 866 [23] E. HABER AND J. MODERSITZKI, *Numerical methods for volume preserving image registration*,
 867 Inverse Problems, 20 (2004), p. 1621.
- 868 [24] ———, *Intensity gradient based registration and fusion of multi-modal images*, in Medical Image
 869 Computing and Computer-Assisted Intervention—MICCAI 2006, Springer, 2006, pp. 726–
 870 733.
- 871 [25] ———, *Image registration with guaranteed displacement regularity*, International Journal of
 872 Computer Vision, 71 (2007), pp. 361–372.
- 873 [26] D. L. G. HILL, P. G. BATCHELOR, M. HOLDEN, AND D. J. HAWKES, *Medical image registration*,
 874 Physics in Medicine and Biology, 46 (2001), pp. R1–45.
- 875 [27] M. IBRAHIM, K. CHEN, AND C. BRITO-LOEZA, *A novel variational model for image registration
 876 using gaussian curvature*, Geometry, Imaging and Computing, 1 (2014), pp. 417–446.
- 877 [28] C. T. KELLEY, *Iterative methods for optimization*, SIAM, 1999.
- 878 [29] A. KLEIN, J. ANDERSSON, B. A. ARDEKANI, J. ASHBURNER, B. AVANTS, M.-C. CHIANG, G. E.
 879 CHRISTENSEN, D. L. COLLINS, J. GEE, P. HELLIER, ET AL., *Evaluation of 14 nonlinear
 880 deformation algorithms applied to human brain mri registration*, Neuroimage, 46 (2009),

- 881 pp. 786–802.
- 882 [30] K. C. LAM AND L. M. LUI, *Landmark-and intensity-based registration with large deformations*
883 *via quasi-conformal maps*, SIAM Journal on Imaging Sciences, 7 (2014), pp. 2364–2392.
- 884 [31] O. LEHTO AND K. I. VIRTANEN, *Quasiconformal mappings in the plane*, vol. 126, Springer New
885 York, 1973.
- 886 [32] H. LESTER AND S. R. ARRIDGE, *A survey of hierarchical non-linear medical image registration*,
887 *Pattern Recognition*, 32 (1999), pp. 129–149.
- 888 [33] M. Z. LIU, *Total Bregman divergence, a robust divergence measure, and its applications*, PhD
889 thesis, University of Florida, USA. ISBN: 978-1-267-37783-8, 2011.
- 890 [34] L. M. LUI, K. C. LAM, T. W. WONG, AND X. GU, *Texture map and video compression using*
891 *beltrami representation*, SIAM Journal on Imaging Sciences, 6 (2013), pp. 1880–1902.
- 892 [35] F. MAES, A. COLLIGNON, D. VANDERMEULEN, G. MARCHAL, AND P. SUETENS, *Multimodality*
893 *image registration by maximization of mutual information*, IEEE Transactions on Medical
894 Imaging, 16 (1997), pp. 187–198.
- 895 [36] J. A. MAINTZ AND M. A. VIERGEVER, *A survey of medical image registration*, *Medical Image*
896 *Analysis*, 2 (1998), pp. 1–36.
- 897 [37] A. MANG AND L. RUTHOTTO, *A Lagrangian Gauss–Newton–Krylov solver for mass- and*
898 *intensity-preserving diffeomorphic image registration*, SIAM Journal on Scientific Com-
899 puting, 39 (2017), pp. B860–B885.
- 900 [38] J. MODERSITZKI, *Numerical Methods For Image Registration*, Oxford University Press, 2004.
- 901 [39] ———, *FAIR: Flexible Algorithms for Image Registration*, SIAM, 2009.
- 902 [40] A. MOHAMED, E. I. ZACHARAKI, D. SHEN, AND C. DAVATZIKOS, *Deformable registration of*
903 *brain tumor images via a statistical model of tumor-induced deformation*, *Medical Image*
904 *Analysis*, 10 (2006), pp. 752–763.
- 905 [41] O. MUSSE, F. HEITZ, AND J. P. ARMSPACH, *Topology preserving deformable image matching*
906 *using constrained hierarchical parametric models*, IEEE Transactions on Image Processing,
907 10 (2001), pp. 1081–1093.
- 908 [42] J. NOCEDAL AND S. WRIGHT, *Numerical Optimization*, Springer, 2006.
- 909 [43] C. C. PAIGE AND M. A. SAUNDERS, *Solution of sparse indefinite systems of linear equations*,
910 *SIAM Journal on Numerical Analysis*, 12 (1975), pp. 617–629.
- 911 [44] T. ROHLFING, C. R. MAURER JR, D. A. BLUEMKE, AND M. A. JACOBS, *Volume-preserving non-*
912 *rigid registration of mr breast images using free-form deformation with an incompressibility*
913 *constraint*, IEEE Transactions on Medical Imaging, 22 (2003), pp. 730–741.
- 914 [45] D. RUECKERT, L. I. SONODA, C. HAYES, D. L. HILL, M. O. LEACH, AND D. J. HAWKES, *Non-*
915 *rigid registration using free-form deformations: application to breast MR images*, IEEE
916 *Transactions on Medical Imaging*, 18 (1999), pp. 712–721.
- 917 [46] M. SDIKA, *A fast nonrigid image registration with constraints on the Jacobian using large scale*
918 *constrained optimization*, IEEE Transactions on Medical Imaging, 27 (2008), pp. 271–281.
- 919 [47] A. SOTIRAS, C. DAVATZIKOS, AND N. PARAGIOS, *Deformable medical image registration: A*
920 *survey*, IEEE Transactions on Medical Imaging, 32 (2013), pp. 1153–1190.
- 921 [48] W. SUN AND Y.-X. YUAN, *Optimization theory and methods: nonlinear programming*, vol. 1,
922 Springer, 2006.
- 923 [49] J.-P. THIRION, *Image matching as a diffusion process: an analogy with Maxwell’s demons*,
924 *Medical Image Analysis*, 2 (1998), pp. 243–260.
- 925 [50] A. TROUVÉ, *Diffeomorphisms groups and pattern matching in image analysis*, *International*
926 *journal of computer vision*, 28 (1998), pp. 213–221.
- 927 [51] T. VERCAUTEREN, X. PENNEC, A. PERCHANT, AND N. AYACHE, *Diffeomorphic demons: Effi-*
928 *cient non-parametric image registration*, *NeuroImage*, 45 (2009), pp. S61–S72.
- 929 [52] C. R. VOGEL, *Computational Methods for Inverse Problems*, SIAM, 2002.
- 930 [53] J. WEICKERT, B. ROMENY, AND M. VIERGEVER, *Efficient and reliable schemes for nonlinear*
931 *diffusion filtering*, IEEE Transactions on Image Processing, 7 (1998), pp. 398–410.
- 932 [54] X. YANG, J. H. PEI, AND J. L. SHI, *Inverse consistent non-rigid image registration based on*
933 *robust point set matching*, *BioMedical Engineering OnLine*, 13 (2014).
- 934 [55] I. YANOVSKY, P. THOMPSON, S. OSHER, AND A. LEOW, *Large deformation unbiased diffeomor-*
935 *phic nonlinear image registration: Theory and implementation*, IEEE conference CVPR’
936 2007 (see also UCLA CAM Report 06-71), 71 (2007).
- 937 [56] J. ZHANG AND K. CHEN, *Variational image registration by a total fractional-order variation*
938 *model*, *Journal of Computational Physics*, 293 (2015), pp. 442–461.
- 939 [57] B. ZITOVA AND J. FLUSSER, *Image registration methods: a survey*, *Image and Vision Comput-*
940 *ing*, 21 (2003), pp. 977–1000.