

A Splitting Method for Diffeomorphism Optimization Problem Using Beltrami Coefficients

Lok Ming Lui · Tsz Ching Ng

Received: 12 January 2014 / Revised: 12 August 2014 / Accepted: 13 August 2014 /

Published online: 3 September 2014

© Springer Science+Business Media New York 2014

Abstract Finding a meaningful 1–1 correspondence between different data, such as images or surface data, has important applications in various fields. It involves the optimization of certain energy functionals over the space of all diffeomorphisms. This type of optimization problems (called the *diffeomorphism optimization problems*, *DOPs*) is especially challenging, since the bijectivity of the mapping has to be ensured. Recently, a method, called the *Beltrami holomorphic flow* (*BHF*), has been proposed to solve the *DOP* using quasi-conformal theories (Lui et al. in J Sci Comput 50(3):557–585, 2012). The optimization problem is formulated over the space of Beltrami coefficients (*BCs*), instead of the space of all diffeomorphisms. *BHF* iteratively finds a sequence of *BCs* associated with a sequence of diffeomorphisms, using the gradient descent method, to minimize the energy functional. The use of *BCs* effectively controls the smoothness and bijectivity of the mapping, and hence makes it easier to handle the constrained optimization problem. However, the algorithm is computationally expensive. In this paper, we propose an efficient splitting algorithm, based on the classical alternating direction method of multiplier (*ADMM*), to solve the *DOP*. The basic idea is to split the energy functional into two energy terms: one involves the *BC* whereas the other involves the quasi-conformal map. Alternating minimization scheme can then be applied to minimize the energy functional. The proposed method significantly speeds up the previous *BHF* approach. It also extends the previous *BHF* algorithm to Riemann surfaces of arbitrary topologies, such as multiply-connected shapes. Experiments have been carried out on synthetic together with real surface data, which demonstrate the efficiency and efficacy of the proposed algorithm to solve the *DOP*.

Keywords Beltrami holomorphic flow · Diffeomorphism optimization problem · Beltrami coefficient · Quasi-conformal theories · Alternating direction method of multiplier

L. M. Lui (✉) · T. C. Ng

The Department of Mathematics, The Chinese University of Hong Kong, Hong Kong, China
e-mail: lmlui@math.cuhk.edu.hk

1 Introduction

Registration, which aims to find a meaningful one-to-one pointwise mapping between two corresponding data, is important in various fields, such as medical imaging, computer visions and computer graphics. For example, in medical imaging, finding accurate 1–1 correspondences between brain cortical surfaces is crucial for medical shape analysis. While in computer graphics, finding a 1–1 correspondence (called the texture mapping) between a surface mesh and a two dimensional image is necessary for generating a textured surface. Developing an effective way to obtain the registration becomes an important research field.

A meaningful registration often satisfies certain important properties. For example, for image registration, a good registration is determined by how well it matches the image intensities between the two images. While for geometric matching surface registration, it is often desirable to look for registrations that match surface curvatures as much as possible. Mathematically, this kind of problems can be formulated as an optimization problem of certain energy functionals over the space of all diffeomorphisms. More specifically, suppose S_1 and S_2 are two corresponding data to be registered. Our goal is to look for an orientation-preserving diffeomorphism $f^* : S_1 \rightarrow S_2$ such that:

$$f^* = \operatorname{argmin}_{f \in \mathfrak{Diff}} E(f) \quad (1)$$

where \mathfrak{Diff} is the collection of all orientation-preserving diffeomorphisms between S_1 and S_2 and $E : \mathfrak{Diff} \rightarrow \mathbb{R}^+$ is an energy functional defined on \mathfrak{Diff} . This kind of optimization problems, which optimize an energy functional over the space of all diffeomorphisms, is called a *diffeomorphism optimization problem (DOP)*.

Solving a DOP is generally challenging, since the bijectivity of the mapping can be easily lost during the optimization process. To tackle with this problem, finding a suitable representation for \mathfrak{Diff} that facilitates the optimization process is necessary. In [1], the Beltrami coefficient (BC) was proposed to represent an orientation-preserving diffeomorphism. A BC is a complex-valued function defined on S_1 with supreme norm strictly less than 1. We usually denote the set of all BCs by $\mathfrak{B} = \{\mu : S_1 \rightarrow \mathbb{C} : \|\mu\|_\infty < 1\}$. It can be shown that there is an injection $\Phi : \mathfrak{Diff} \rightarrow \mathfrak{B}$ from \mathfrak{Diff} to \mathfrak{B} under suitable boundary or landmark constraints [1]. In other words, every orientation-preserving diffeomorphism g is associated with a unique BC μ_g .

The BC is more suitable for solving the DOP as it has the least amount of constraints. For instance, the representation of a diffeomorphism using its coordinate functions has to satisfy the 1–1 and onto constraints. It can be reduced to a constraint on the Jacobian, which is a nonlinear partial differential inequality. This constraint adds extra difficulty when solving the DOP. On the contrary, the BC has the least amount of constraints. It does not need to be 1–1 or onto. The only constraint is that its supreme norm has to be strictly less than 1. The original DOP (1) can then be reformulated over the space of BC \mathfrak{B} as follows:

$$v^* = \operatorname{argmin}_{\mu \in \mathfrak{B}} E_B(\mu) \quad (2)$$

subject to the constraint that (i) $v^* = \mu(f^*) = \text{BC}$ of some $f^* \in \mathfrak{Diff}$ and (ii) $\|v^*\|_\infty < 1$. Formulating the DOP over \mathfrak{B} makes the optimization problem more manageable, since the bijectivity can be easily controlled.

It is noteworthy to mention that formulating the DOP over \mathfrak{B} does not only make the optimization problem easier, the BC also captures local geometric distortions of the diffeomorphism. Therefore, by incorporating the BC into the energy functional, one can enforce desired properties on the geometric distortions of the map. For example, suppose S_1 and

S_2 are two surfaces, say two human faces. Let H_1 and H_2 are the mean curvatures of S_1 and S_2 respectively. To obtain a geometric matching registration, one might need to find a diffeomorphism that matches the surface curvatures while minimizing the local geometric (conformality) distortion. This problem can be formulated as a DOP as follows:

$$\nu^* = \operatorname{argmin}_{\mu \in \mathfrak{B}} \left\{ \int_{S_1} |\mu|^p + \alpha \int_{S_1} |H_1 - H_2(f^\mu)|^2 \right\} \quad (3)$$

with the constraint that $\nu^* = \mu(f^*) = \text{BC}$ of some $f^* \in \text{Diff}$, where f^μ is the quasi-conformal map associated to μ . The first term minimizes the L^p norm of the conformality distortion measured by the BC of the quasi-conformal map. The second term aims to minimize the curvature mismatching error. Sometimes, it might be required that the registration matches corresponding landmarks $\{p_i\}_{i=1}^n$ and $\{q_i\}_{i=1}^n$ on S_1 and S_2 respectively. In this case, we add an extra constraint to the DOP (3) that $f^*(p_i) = q_i$ for $i = 1, 2, \dots, n$.

To solve the DOPs (2) or (3), a gradient descent based method was proposed in [1]. The quasi-conformal map associated to a given perturbed BC, which is needed in deriving the descent direction [such as the second term in (3)], was computed using an integral formula. Although the optimization problem can be effectively solved, the computation is quite inefficient. Also, this gradient descent based algorithm converges slowly. In this paper, we propose an efficient splitting method, based on the classical alternating direction method with multipliers (ADMM), to solve the DOP over the space of BCs. The basic idea is to split the energy functional into two energy terms: one involves the BC whereas the other involves the quasi-conformal map. Alternating minimization scheme can then be applied to minimize the the energy functional. The proposed method significantly speed up the previous BHF approach in [1]. It also extends the previous BHF algorithm to Riemann surfaces of arbitrary topologies. Experiments have been carried out on synthetic together with real medical data. Results show that the proposed algorithm solves the DOP efficiently.

In summary, the main contribution of this paper is to apply the alternating direction method of multipliers (ADMM) together with the quasi-conformal theories to solve the DOP. Using ADMM, a splitting method can be used to minimize the energy functional in the DOP alternatively over the quasi-conformal maps and BCs. The algorithm speeds up the previous method proposed in [1] and extends it to solve DOPs on Riemann surfaces of arbitrary topologies. Experimental results show that the newly proposed method often yields better registration results.

2 Related Work

Finding meaningful mapping or registration between corresponding data that optimizes certain kinds of energy functionals has been extensively studied. In this section, we briefly describe some related methods commonly used.

Conformal maps have been widely studied to obtain smooth 1–1 correspondence between surfaces that minimize angular distortions [2, 8–11, 13, 25, 39]. Conformal maps are usually computed using variational approaches that minimize some energy functionals, such as the harmonic energy [8] and the least-squares energy based on the Cauchy–Riemann equation [25]. A 1–1 correspondence between surfaces can be obtained in the optimal state. However, the above registration cannot map anatomical features, such as sulcal landmarks, consistently from subject to subject.

To obtain a surface registration that matches important landmark features, landmark-based diffeomorphisms are often used. Optimization of surface diffeomorphisms by landmark

matching has been extensively studied. Gu et al. [8] improved a conformal parameterization by composing an optimal Möbius transformation so that it minimizes a landmark mismatch energy. The resulting parameterization remains conformal, although features cannot be perfectly matched. Wang et al. [19,20,40] proposed a variational framework to compute an optimized conformal registration that aligns landmarks as well as possible. However, landmarks are not matched exactly and diffeomorphisms cannot be guaranteed when there is a large number of landmark features. Durrleman et al. [3,4] developed a framework using currents, a concept from differential geometry, to match landmarks within surfaces across subjects, for the purpose of inferring the variability of brain structure in an image database. Landmark curves are not perfectly matched. Tosun et al. [32] proposed a more automated mapping technique that attempts to align cortical sulci across subjects by combining parametric relaxation, iterative closest point registration, and inverse stereographic projection. Glaunès et al. [7,12] proposed to generate large deformation diffeomorphisms of a sphere onto itself, given the displacements of a finite set of template landmarks. The diffeomorphism obtained can better match landmark features.

Lui et al. [21] proposed to compute shape-based landmark matching registrations between brain surfaces using the integral flow method. The one-parameter subgroup within the set of all diffeomorphisms was considered and represented by smooth vector fields. Landmarks can be perfectly matched and the correspondence between landmark curves is based on shape information. Leow et al. [16] proposed a level-set-based approach for matching different types of features, including points, 2D and 3D curves represented as implicit functions. These matching features in the parameter domain were then pulled back onto surfaces to compute correspondence fields. In related work, Lepore et al. [17] used a level-set representation to match curves embedded in surfaces, using a diffeomorphic flow parametrized by velocity fields on the sphere. In this work, features within the landmark curves were not matched, and the landmark curves were matched as level sets. Later, Shi et al. [31] computed a direct harmonic mapping between two surfaces by embedding both surfaces as the level-set of an implicit function, and representing the mapping energy as a Dirichlet functional in 3D volume domains. Although such an approach can incorporate landmark constraints, it has not been proven to yield diffeomorphic mappings. Quasi-conformal mapping that matches landmarks consistently has also been proposed. Wei et al. [26] proposed to compute quasi-conformal mappings for feature matching face registration. The Beltrami coefficient associated to a landmark-matching parameterization is approximated. However, either exact landmark matching or the bijectivity of the mapping cannot be guaranteed, especially when very large deformations occur.

Since there may not be well-defined landmarks on surfaces, some authors proposed driving features into correspondence based on shape information or scalar fields defined on the surfaces. Lyttelton et al. [27] computed surface parameterizations that match surface curvature. Fischl et al. [5] improved the alignment of cortical folding patterns by minimizing the mean squared difference between the average convexity across a set of subjects and that of the individual. Wang et al. [33] computed surface registrations that maximize the mutual information between mean curvature and conformal factor maps across subjects. Lord et al. [18] matched surfaces by minimizing the deviation from isometry. Quasi-conformal surface registrations, which minimize geometric mismatching, have also been studied [1,22–24]. For example, Lui et al. [22] proposed to compute quasi-conformal registration between hippocampal surfaces, which matches geometric quantities (such as curvatures) and minimizes the conformality distortion [22]. In most situations, one has to pay extra attention to ensure the optimal map computed is diffeomorphic. In [1], a method, called the *Beltrami holomorphic flow*, has been proposed to optimize the energy functional defined over the space of quasi-conformal maps.

The algorithm effectively computes a diffeomorphic quasi-conformal map that optimizes the given energy functional, although it is computationally expensive. Furthermore, various techniques for computing the quasi-conformal map of a given BC have also been proposed [23, 24, 36].

3 Background

3.1 Quasi-conformal Geometry

In this subsection, we give a brief description on quasi-conformal theories. For details, we refer the readers to [6, 14, 15].

A surface S with a conformal structure is called a *Riemann surface*. Given two Riemann surfaces M and N , a map $f : M \rightarrow N$ is *conformal* if it preserves the surface metric up to a multiplicative factor called the *conformal factor*. An immediate consequence is that every conformal map preserves angles. With the angle-preserving property, a conformal map effectively preserves the local geometry of the surface structure. A generalization of conformal maps is the *quasi-conformal* maps, which are orientation preserving homeomorphisms between Riemann surfaces with bounded conformality distortion, in the sense that their first order approximations takes small circles to small ellipses of bounded eccentricity [15]. Mathematically, $f : \mathbb{C} \rightarrow \mathbb{C}$ is quasi-conformal provided that it satisfies the Beltrami equation:

$$\frac{\partial f}{\partial \bar{z}} = \mu(z) \frac{\partial f}{\partial z}. \quad (4)$$

for some complex-valued function μ satisfying $\|\mu\|_\infty < 1$. μ is called the *Beltrami coefficient*, which is a measure of non-conformality. It measures how far the map at each point is deviated from a conformal map. In particular, the map f is conformal around a small neighborhood of p when $\mu(p) = 0$. Infinitesimally, around a point p , f may be expressed with respect to its local parameter as follows:

$$\begin{aligned} f(z) &= f(p) + f_z(p)z + f_{\bar{z}}(p)\bar{z} \\ &= f(p) + f_z(p)(z + \mu(p)\bar{z}). \end{aligned} \quad (5)$$

Obviously, f is not conformal if and only if $\mu(p) \neq 0$. Inside the local parameter domain, f may be considered as a map composed of a translation to $f(p)$ together with a stretch map $S(z) = z + \mu(p)\bar{z}$, which is postcomposed by a multiplication of $f_z(p)$, which is conformal. All the conformal distortion of $S(z)$ is caused by $\mu(p)$. $S(z)$ is the map that causes f to map a small circle to a small ellipse. From $\mu(p)$, we can determine the angles of the directions of maximal magnification and shrinking and the amount of them as well. Specifically, the angle of maximal magnification is $\arg(\mu(p))/2$ with magnifying factor $1 + |\mu(p)|$; The angle of maximal shrinking is the orthogonal angle $(\arg(\mu(p)) - \pi)/2$ with shrinking factor $1 - |\mu(p)|$. Thus, the Beltrami coefficient μ gives us all the information about the properties of the map (see Fig. 1).

The maximal dilation of f is given by:

$$K(f) = \frac{1 + \|\mu\|_\infty}{1 - \|\mu\|_\infty}. \quad (6)$$

Quasiconformal mapping between two Riemann surfaces S_1 and S_2 can also be defined. Instead of the Beltrami coefficient, the *Beltrami differential* is used. A Beltrami differential

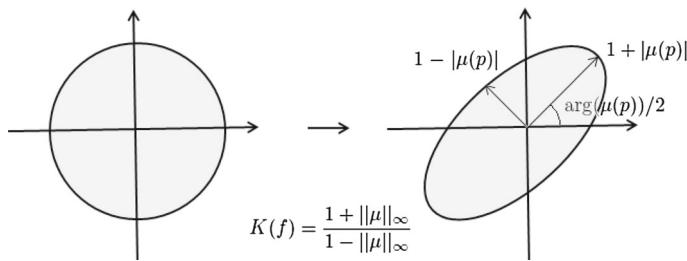


Fig. 1 Illustration of how the Beltrami coefficient determines the conformality distortion

$\mu(z)\frac{d\bar{z}}{dz}$ on a Riemann surface S is an assignment to each chart (U_α, ϕ_α) of an L_∞ complex-valued function μ_α , defined on local parameter z_α such that

$$\mu_\alpha(z_\alpha)\frac{d\bar{z}_\alpha}{dz_\alpha} = \mu_\beta(z_\beta)\frac{d\bar{z}_\beta}{dz_\beta}, \quad (7)$$

on the domain which is also covered by another chart (U_β, ϕ_β) . Here, $\frac{dz_\beta}{dz_\alpha} = \frac{d}{dz_\alpha}\phi_{\alpha\beta}$ and $\phi_{\alpha\beta} = \phi_\beta \circ \phi_\alpha^{-1}$.

An orientation preserving diffeomorphism $f : S_1 \rightarrow S_2$ is called quasi-conformal associated with $\mu(z)\frac{d\bar{z}}{dz}$ if for any chart (U_α, ϕ_α) on S_1 and any chart (V_β, ψ_β) on S_2 , the mapping $f_{\alpha\beta} := \psi_\beta \circ f \circ \phi_\alpha^{-1}$ is quasi-conformal associated with $\mu_\alpha(z_\alpha)$.

3.2 Beltrami Holomorphic Flow

The Beltrami holomorphic flow (BHF) refers to a flow of quasi-conformal maps over $t \in \mathbb{C}$, which is associated with a flow of Beltrami coefficients. Mathematically, suppose $\mu(t, z) : \mathbb{C} \times S_1 \rightarrow \mathbb{C}$ such that $\mu(t, \cdot) \in \mathcal{B}$ for all $t \in \mathbb{C}$. Assume that $\mu(\cdot, p)$ is holomorphic for each fixed $p \in S_1$. Then, $\mu = \mu(t, \cdot)$ is called a *holomorphic flow of the Beltrami coefficients*. For simply-connected S_1 and S_2 , each $\mu(t, \cdot)$ is associated with a quasi-conformal map $f(t, \cdot) : S_1 \rightarrow S_2$. $f(t, \cdot)$ is called the *Beltrami holomorphic flow (BHF)* of quasi-conformal maps associated with $\mu(t, \cdot)$. It can be shown that for any fixed $p \in S_1$, $f(\cdot, p) : \mathbb{C} \rightarrow \mathbb{C}$ is holomorphic. More specifically, let $\{\mu(t, \cdot)\}$ be a family of BCs depending on a complex parameter $t \in \mathbb{C}$. Suppose $\mu(t, \cdot)$ can be written in the form:

$$\mu(t, z) = \mu(z) + (t - t_0)v(z) + (t - t_0)\epsilon(t - t_0)(z) \quad (8)$$

for $z \in \mathbb{C}$, with a suitable $\mu(z)$ in the unit ball of $C^\infty(\mathbb{C})$, $v, \epsilon \in L^\infty(\mathbb{C})$ such that $\|\epsilon(t - t_0)\|_\infty \rightarrow 0$ as $t \rightarrow t_0$. Then for all $p \in S_1$,

$$f^{\mu(t, \cdot)}(p) = f^\mu(p) + (t - t_0)\mathbf{V}(f^\mu, v) + o(|t - t_0|) \quad (9)$$

local uniformly as $t \rightarrow t_0$, where

$$\mathbf{V}(f^\mu, v)(p) = -\frac{f^\mu(f^\mu(p) - 1)}{\pi} \int_{\mathbb{C}} \frac{v(z)((f^\mu)_z(z))^2}{f^\mu(z)(f^\mu(z) - 1)(f^\mu(z) - f^\mu(p))} dx dy \quad (10)$$

if S_1 and S_2 are genus-0 closed surfaces. Here, we have identified S_1 and S_2 with $\mathbb{S}^2 \cong \mathbb{C}$ through spherical conformal parameterizations. And if S_1 and S_2 are simply-connected open surfaces, we have

$$\begin{aligned} \mathbf{V}(f^\mu, v)(p) = & -\frac{f^\mu(f^\mu(p)-1)}{\pi} \left(\int_{\mathbb{D}} \frac{v(z)((f^\mu)_z(z))^2}{f^\mu(z)(f^\mu(z)-1)(f^\mu(z)-f^\mu(p))} dx dy \right. \\ & \left. + \int_{\mathbb{D}} \frac{\overline{v(z)}((f^\mu)_z(z))^2}{\overline{f^\mu(z)}(1-\overline{f^\mu(z)})(1-\overline{f^\mu(z)}f^\mu(p))} dx dy \right). \end{aligned} \quad (11)$$

Here, we have identified S_1 and S_2 with \mathbb{D} through conformal parameterizations.

BHF has been applied to solve the DOP [1]. In particular, as the Beltrami coefficient is updated during the gradient descent based iterative minimization process, the associated quasi-conformal map is also updated using Eqs. (10) or (11). This method has been successfully applied for surface registration in the field of medical imaging and computer graphics. However, the computation of the quasi-conformal map of a given BC using the integral formula (10) or (11) is quite time-consuming in practice. Recently, various algorithms have been proposed to compute the quasi-conformal map from a prescribed BC efficiently. For example, Lui et al. [24, 37] proposed to compute quasi-conformal maps by solving a generalized Laplace's equation. The algorithm, which is called the Linear Beltrami Solver (LBS), can be discretized into a sparse symmetric positive definite linear system. The linear system is then solved by the conjugate gradient method. Gu et al. [41] proposed to compute the quasi-conformal map using the holomorphic differential 1-form under the auxiliary metric given by the BC. Zeng et al. [23] proposed a curvature flow method to compute the quasi-conformal map through calculating a conformal map under the auxiliary metric given by the prescribed BC. The algorithm converges exponentially. Wong et al. [36] proposed to approximate the small perturbation of the quasi-conformal map from the identity map under a small variation $t v$ of the BC from $\mu = 0$ by solving a PDE. This gives a first-order approximation of the perturbation when t is small. The quasi-conformal map of a given target BC can then be obtained using a composition formula of quasi-conformal maps. Ng et al. [38] proposed to iteratively compute the quasi-conformal map f^v with Beltrami coefficient v by solving the Beltrami's equation in each iteration. The Beltrami's equation is solved by a L^p -minimization of the Beltrami energy:

$$\tilde{f} = \underset{f \in \mathcal{D}_{\text{iff}}}{\text{argmin}} \left\{ \left\| \frac{\partial f}{\partial \bar{z}} - v \frac{\partial f}{\partial z} \right\|_p^p \right\} \quad (12)$$

This is applied to solve a subproblem for computing the Teichmüller map between multiply-connected domains.

Note that when $p = 2$, the minimizer of the least square problem (12) is called the *least-square quasi-conformal map (LSQC) associated to v* , which has also been studied in [35, 42]. In this paper, we will apply the method in [38] to cope with a subproblem in the alternating minimization algorithm for solving the DOP.

3.3 Alternating Direction Method with Multipliers (ADMM)

ADMM is an optimization algorithm that solves the following type of problems:

$$\text{Minimize } \{E_1(x) + E_2(Ax)\} \quad (13)$$

where $A \in M_{m \times n}(\mathbb{R})$ has full column rank. E_1 and E_2 are often assumed to be convex functionals.

The problem (13) can be reformulated as a constrained optimization problem:

$$\text{Minimize } \{E_1(x) + E_2(y)\} \text{ subject to } Ax = y \quad (14)$$

The augmented Lagrangian associated to the problem (15) is given by

$$L(x, y, \lambda, \mu) = E_1(x) + E_2(y) + \lambda^T(Ax - y) + \frac{\mu}{2} \|Ax - y\|^2 \quad (15)$$

The classical augmented Lagrangian method iteratively solves the problem (15) as follow:

$$(x^{k+1}, y^{k+1}) = \operatorname{argmin}\{L(x, y, \lambda^k, \mu^k)\} \quad (16)$$

$$\lambda^{k+1} = \lambda^k + \mu_k(Ax^{k+1} - y^{k+1}) \quad (17)$$

where $\{\lambda^k\}$ is the sequence approximating the Lagrange multiplier of the constraint $Ax = y$ and $\{\mu_k\}$ is a sequence of positive real number, called the *penalty parameters*. A variants of the choices of $\{\lambda^k\}$ and $\{\mu_k\}$ have been proposed.

However, solving the subproblem (16) is sometimes non-trivial, since E_1 and E_2 are strongly coupled with each others under the constraint $Ax = y$. To simplify the problem, the alternating direction method with multiplier (ADMM) has been proposed to decouple the minimization process as follows:

$$\begin{aligned} x^{k+1} &= \operatorname{argmin}\{L(x, y^k, \lambda^k, \mu^k)\} \\ y^{k+1} &= \operatorname{argmin}\{L(x^{k+1}, y, \lambda^k, \mu^k)\} \\ \lambda^{k+1} &= \lambda^k + \mu_k(Ax^{k+1} - y^{k+1}) \end{aligned} \quad (18)$$

More specifically, ADMM first solves for x^{k+1} by fixing $y = y^k$, and then solves for y^{k+1} by fixing $x = x^{k+1}$. This simple decoupling leads to efficient and parallelizable optimization algorithms for the subproblem (16).

ADMM dates back to 1975 but regains lots of attention recently due to its simple implementation and extensive applications to image processing and compressive sensing. For details, we refer the readers to [28–30].

4 Proposed Algorithm

We restate our mathematical problem as follows. Let S_1 and S_2 be two corresponding domains, which can either be 2D domains or Riemann surfaces embedded in \mathbb{R}^3 . Our goal is to look for a diffeomorphism $f^* : S_1 \rightarrow S_2$ that solves:

$$f^* = \operatorname{argmin}_{f \in \mathfrak{Diff}} E(f) \quad (19)$$

where \mathfrak{Diff} is the collection of all surface diffeomorphisms between S_1 and S_2 and $E : \mathfrak{Diff} \rightarrow \mathbb{R}$ is a functional on \mathfrak{Diff} .

Solving the above DOP is generally challenging, since the bijectivity has to be ensured. It is especially difficult when S_1 and S_2 has complicated geometry. In [1], we introduced the idea of using the Beltrami coefficients to represent \mathfrak{Diff} . Every diffeomorphism $f \in \mathfrak{Diff}$ is associated with a unique Beltrami coefficient $\mu(f) : S_1 \rightarrow \mathbb{C}$. The Beltrami coefficient $\mu(f)$ also measures the local geometric (conformality) distortion of the mapping f . Thus, it is sometimes desirable to incorporate $\mu(f)$ in the energy functional in order to control the geometric distortion of the mapping. In other words, our optimization problem can generally be written as finding $f^* : S_1 \rightarrow S_2$ that solves:

$$f^* = \operatorname{argmin}_{f \in \mathfrak{Diff}} \{E_1(f) + E_2(\mu(f))\} \quad (20)$$

subject to $\|\mu(f^*)\|_\infty := \|\frac{\partial f^*}{\partial \bar{z}} / \frac{\partial f^*}{\partial z}\|_\infty < 1$, where $E_1 : \mathfrak{Diff} \rightarrow \mathbb{R}^+$ and $E_2 : \mathfrak{B} \rightarrow \mathbb{R}^+$ are energy functionals defined on \mathfrak{Diff} and \mathfrak{B} respectively. E_1 drives the mapping to satisfy

some desired properties, such as matching landmarks or curvatures. E_2 controls the local geometric distortions of the mapping.

The above optimization can be further formulated as finding $f^* : S_1 \rightarrow S_2$ and $v^* : S_1 \rightarrow \mathbb{C}$ such that:

$$(f^*, v^*) = \operatorname{argmin}_{f \in \mathfrak{D}, v \in \mathfrak{B}} \{E_1(f) + E_2(v)\} \quad (21)$$

subject to (i) $v^* = \mu(f^*)$ and (ii) $\|v^*\|_\infty < 1$. Condition (i) guarantees that v^* is an admissible Beltrami coefficient, which is associated to a quasi-conformal map f^* . Condition (ii) ensures that f^* is diffeomorphic. It can be understood by the following theorem.

Proposition 1 *If v^* and $f^* : S_1 \rightarrow S_2$ satisfies the constraints (i) and (ii), then f^* is a diffeomorphism.*

Proof Suppose $f^* = u + iv$ under some local coordinates. According to condition (i), the Beltrami coefficient of f^* is v^* . v^* is given by:

$$v^* = \frac{\partial f^*}{\partial \bar{z}} / \frac{\partial f^*}{\partial z} \quad (22)$$

where

$$\frac{\partial f^*}{\partial \bar{z}} = (u_x - v_y) + i(u_y + v_x); \quad \frac{\partial f^*}{\partial z} = (u_x + v_y) + i(v_x + u_y); \quad (23)$$

Now, the Jacobian of f^* , J_{f^*} , is given by:

$$\begin{aligned} J_{f^*} &= u_x v_y - u_y v_x \\ &= \frac{(u_x + v_y)^2 + (v_x + u_y)^2 - (u_x - v_y)^2 - (u_y + v_x)^2}{4} \\ &= |\frac{\partial f^*}{\partial z}|^2 - |\frac{\partial f^*}{\partial \bar{z}}|^2 = |\frac{\partial f^*}{\partial z}|^2(1 - |v^*|^2) \end{aligned} \quad (24)$$

Since $\|v^*\|_\infty < 1$, $|\frac{\partial f^*}{\partial z}|^2 \neq 0$. Also, $(1 - |v^*|^2) > 0$. Hence, $J_{f^*} > 0$ everywhere.

Since the Jacobian is positive everywhere, by the inverse function theorem, the mapping f^* is locally invertible everywhere. In other words, f^* is a diffeomorphism. \square

The incorporation of Beltrami coefficient into the optimization problem makes it easier to control the bijectivity of the mapping.

Now, in order to solve the DOP (21), our strategy is to apply an ADMM-like alternating optimization scheme. We consider the augmented Lagrangian of the problem (21), which is given by:

$$\begin{aligned} L(f, v, \lambda_{\mathbf{Re}}, \lambda_{\mathbf{Im}}, \rho) &= E_1(f) + E_2(v) + \langle \lambda_{\mathbf{Re}}, \mathbf{Re}(v - \mu(f)) \rangle \\ &\quad + \langle \lambda_{\mathbf{Im}}, \mathbf{Im}(v - \mu(f)) \rangle + \frac{\rho}{2} \|v - \mu(f)\|_2^2 \end{aligned} \quad (25)$$

where $\langle \alpha, \beta \rangle := \int_{S_1} \alpha \beta$ and $\|\alpha\| := (\int_{S_1} |\alpha|^2)^{1/2}$.

Following the minimization procedure of ADMM, we iteratively solve (21) as follows. Given f_k , v_k , $\lambda_{\mathbf{Re}}^k$, $\lambda_{\mathbf{Im}}^k$ and ρ_k at the k-th iteration, we compute

$$f^{k+1} = \operatorname{argmin}_f \left\{ L(f, v_k, \lambda_{\mathbf{Re}}^k, \lambda_{\mathbf{Im}}^k, \rho^k) \right\} \quad (26)$$

$$v^{k+1} = \operatorname{argmin}_v \left\{ L(f_{k+1}, v, \lambda_{\mathbf{Re}}^k, \lambda_{\mathbf{Im}}^k, \rho^k) \right\} \quad (27)$$

$\lambda_{\mathbf{Re}}^k$, $\lambda_{\mathbf{Im}}^k$ and ρ^k are updated as follows.

If $\|v_{k+1} - \mu(f_{k+1})\|_2 < \eta_k$, update:

$$\begin{aligned}\lambda_{\mathbf{Re}}^{k+1} &= \lambda_{\mathbf{Re}}^k + \rho_k \mathbf{Re}(v_{k+1} - \mu(f_{k+1})); \\ \lambda_{\mathbf{Im}}^{k+1} &= \lambda_{\mathbf{Im}}^k + \rho_k \mathbf{Im}(v_{k+1} - \mu(f_{k+1})); \\ \rho_{k+1} &= \rho_k.\end{aligned}\quad (28)$$

If $\|v_{k+1} - \mu(f_{k+1})\|_2 \geq \eta_k$, update:

$$\begin{aligned}\lambda_{\mathbf{Re}}^{k+1} &= \lambda_{\mathbf{Re}}^k; \\ \lambda_{\mathbf{Im}}^{k+1} &= \lambda_{\mathbf{Im}}^k; \\ \rho_{k+1} &= \rho_k(1 + \gamma_k).\end{aligned}\quad (29)$$

We proceed to discuss how we can solve the subproblems (26) and (27).

4.1 Minimization of Subproblem (27) Involving v

Subproblem (27) is often relatively easy to solve. In many situations, the Euler-Lagrange equation of $E_2(\mu)$ is an elliptic PDE. For example, if $E_2(\mu) = \int_{S_1} |\nabla \mu|^2 + |\mu|^2$, then the Euler-Lagrange equation for the subproblem (27) can be written as:

$$\Delta v - 2v - \lambda_{\mathbf{Re}}^k \mathbf{Re}(v - \mu(f_{k+1})) - i\lambda_{\mathbf{Im}}^k \mathbf{Im}(v - \mu(f_{k+1})) - \rho_k(v - \mu(f_{k+1})) = 0 \quad (30)$$

In the discrete case, it becomes a sparse linear system and can be solved efficiently.

In the situation that $E_2(\mu)$ is more difficult to optimize, we use the gradient descent method. For example, if $E_2(\mu) = \int_{S_1} |\nabla \mu|^2 + |\mu|^p$, then we optimize (27) iteratively by:

$$v^{new} = v^{old} + dt dv, \text{ where} \quad (31)$$

$$\begin{aligned}dv = \Delta v - p(\mathbf{Re}(v))^{p-1} - ip(\mathbf{Im}(v))^{p-1} - \lambda_{\mathbf{Re}}^k \mathbf{Re}(v - \mu(f_{k+1})) \\ - i\lambda_{\mathbf{Im}}^k \mathbf{Im}(v - \mu(f_{k+1})) - \rho_k(v - \mu(f_{k+1}))\end{aligned}\quad (32)$$

The stopping criteria is chosen to be $\nabla_v L(f_{k+1}, v, \lambda_{\mathbf{Re}}^k, \lambda_{\mathbf{Im}}^k, \rho_k) < \epsilon_k$.

4.2 Minimization of Subproblem (26) Involving f

Solving the subproblem (26) involving the quasi-conformal map f is comparatively more challenging. Subproblem (26) can be written as:

$$\begin{aligned}f_{k+1} = \operatorname{argmin}_f \left\{ E_1(f) + < \lambda_{\mathbf{Re}}^k, \mathbf{Re}(v_k - \mu(f)) > \right. \\ \left. + < \lambda_{\mathbf{Im}}^k, \mathbf{Im}(v_k - \mu(f)) > + \frac{\rho_k}{2} \|v_k - \mu(f)\|_2^2 \right\}.\end{aligned}\quad (33)$$

By looking for a descent direction in each step, we iteratively minimize the above problem.

To minimize the first term, we compute the descent direction \mathbf{V}_1 for minimizing $E_1(f)$. For instance, suppose E_1 is defined as the intensity mismatching error: $E_1(f) = \|I_1 - I_2(f)\|_2^2$, where I_1 and I_2 are the intensity functions defined on S_1 and S_2 respectively. Then, the descent direction \mathbf{V}_1 is given by: $\mathbf{V}_1 = 2(I_1 - I_2(f))\nabla I_2(f)$.

The last three terms can be minimized as follows. Note that the gradient descent direction $\partial \tilde{\mu}$ in term of the Beltrami coefficient for minimizing the last three terms is given by:

$$\partial \tilde{\mu} = \lambda_{\mathbf{Re}}^k + i\lambda_{\mathbf{Im}}^k + \rho_k(v_k - \mu(f)). \quad (34)$$

Doing gradient descent by several steps, we obtain a new Beltrami coefficient, $\tilde{\mu}$, that reduces the value of the last three energy terms. We can then look for a quasi-conformal map \tilde{f} with BC $\tilde{\mu}$. It is equivalent to looking for \tilde{f} satisfying:

$$\frac{\partial \tilde{f}}{\partial \bar{z}} = \tilde{\mu} \frac{\partial \tilde{f}}{\partial z} \quad (35)$$

subject to the boundary constraints that $\tilde{f}|_{\partial S_1} = \partial S_2$. In [1], the quasi-conformal map is obtained using the integration formula (10) or (11). The computation is therefore quite time-consuming. Various efficient algorithms have been introduced to compute the quasi-conformal map [23, 24, 36–38, 41]. In this paper, we approximate the quasi-conformal map \tilde{f} with a Beltrami coefficient $\tilde{\mu}$ by directly solving the Beltrami's equation (35) as in [38]. The details of the computation of the quasi-conformal map will be explained in Sect. 4.3.

Once \tilde{f} is obtained, we get another descent direction $\mathbf{V}_2 := \tilde{f} - f_k$ for the quasi-conformal map that minimizes the last three terms of the energy functional. The gradient descent method can then be applied to solve the subproblem (26):

$$\frac{df}{dt} = \mathbf{V}_1 + \mathbf{V}_2 \quad (36)$$

We set the stopping criteria for the gradient descent algorithm as $\nabla_f L(f, v_k, \lambda_{\mathbf{Re}}^k, \lambda_{\mathbf{Im}}^k, \rho_k) < \epsilon_k$ to obtain a updated f_{k+1} .

4.3 Computation of Quasi-conformal Map of a Given Beltrami Coefficient

To solve the subproblem (26), a crucial step is to find a quasi-conformal map of a given Beltrami coefficient (BC). More specifically, given a map f^μ with BC μ , we need to deform f^μ to a new f^ν with BC ν . Without loss of generality, we may assume the quasi-conformal maps are diffeomorphisms between Ω_1 and Ω_2 in \mathbb{R}^2 . If not, we can always parameterize the Riemann surfaces conformally onto the 2D parameter domains.

Let $f^\mu : \Omega_1 \rightarrow \Omega_2$ be an initial quasi-conformal map, whose BC is $\mu : \Omega_1 \rightarrow \mathbb{C}$. Assume μ changes to ν , and assume its associated quasiconformal map is denoted by f^ν . As in [38], our goal is to obtain a sequence of quasi-conformal maps $\{f^{\mu_n}\}_{n=1}^\infty$ such that $f^{\mu_0} = f^\mu$ and $f^{\mu_\infty} = f^\nu$. To do this, the basic idea is to flow μ to ν iteratively to obtain a sequence of BCs converging to ν . Their associated quasi-conformal maps $\{f^{\mu_n}\}_{n=1}^\infty$ converges to $f^{\mu_\infty} = f^\nu$. This procedure can be illustrated in more details as follows:

$$\begin{array}{ccccccc} \mu_0 := \mu & \longrightarrow & \mu_1 & \longrightarrow & \cdots & \longrightarrow & \mu_n & \longrightarrow & \cdots & \longrightarrow & \mu_\infty = \nu \\ \downarrow & & \downarrow & & & & \downarrow & & & & \downarrow \\ f^{\mu_0} := f^\mu & \longrightarrow & f^{\mu_1} & \longrightarrow & \cdots & \longrightarrow & f^{\mu_n} & \longrightarrow & \cdots & \longrightarrow & f^{\mu_\infty} = f^\nu \end{array} \quad (37)$$

More specifically, we first set $f^{\mu_0} = f^\mu$ and $\mu_0 = \mu$. We then flow f^{μ_0} to f^{μ_1} whose BC μ_1 is close to $v_1 := (1 - \epsilon)\mu_0 + \epsilon\nu$ ($\epsilon > 0$). To find f^{μ_1} , we need to solve:

$$f^{\mu_1} = \operatorname{argmin}_f \left\{ \left| \left| \frac{\partial f}{\partial \bar{z}} / \frac{\partial f}{\partial z} - v_1 \right| \right|_\infty \right\} \quad (38)$$

For a small variation $v_1 - \mu_0$, the above problem can be solved by a L^p -minimization of the following Beltrami energy (for sufficiently large p):

$$f^{\mu_1} = \operatorname{argmin}_{f \in \mathcal{D}_{\text{diff}}} \left\{ \left| \left| \frac{\partial f}{\partial \bar{z}} - v_1 \frac{\partial f}{\partial z} \right| \right|_p^p \right\} \quad (39)$$

subject to the boundary constraint that $f^{\mu_1}|_{\partial\Omega_1} = \partial\Omega_2$. This problem can be reformulated as finding $g_1 : \Omega_1 \rightarrow \mathbb{R}^2$ such that:

$$g_1 = \operatorname{argmin}_{g:\Omega_1 \rightarrow \mathbb{R}^2} \left\{ \left| \left| \frac{\partial(f^{\mu_0} + g)}{\partial\bar{z}} - v_1 \frac{\partial(f^{\mu_0} + g)}{\partial z} \right| \right|_p^p \right\}. \quad (40)$$

subject to the boundary constraints.

Equivalently, we find g_1 that solves

$$g_1 = \operatorname{argmin}_{g:\Omega_1 \rightarrow \mathbb{R}^2} \left\{ \left| \left| \mathcal{A}(v_1)g + \mathcal{A}(v_1)f^{\mu_0} \right| \right|_p^p \right\}, \quad (41)$$

subject to the boundary constraints, where $\mathcal{A}(v_1) := \frac{\partial}{\partial\bar{z}} - v_1 \frac{\partial}{\partial z}$.

To simplify the computation, we set $p = 2$ in our actual implementation. It is found that the performance is already satisfactory. As a result, we get a new quasi-conformal map $f^{\mu_1} := f^{\mu_0} + g_1$ whose BC is denoted by μ_1 .

Suppose at the n^{th} iteration, we have the quasi-conformal map f^{μ_n} with BC μ_n . We then flow f^{μ_n} to $f^{\mu_{n+1}}$ whose BC μ_{n+1} is close to $v_n := (1 - \epsilon)\mu_n + \epsilon v$ ($\epsilon > 0$). This is again done by a L^p -minimization of the Beltrami energy (39) subject to the boundary constraint. More explicitly, we find g_{n+1} that solves:

$$g_{n+1} = \operatorname{argmin}_{g:\Omega_1 \rightarrow \mathbb{R}^2} \left\{ \left| \left| \mathcal{A}(v_{n+1})g + \mathcal{A}(v_{n+1})f^{\mu_n} \right| \right|_p^p \right\}, \quad (42)$$

subject to the boundary constraints, where $\mathcal{A}(v_{n+1}) := \frac{\partial}{\partial\bar{z}} - v_{n+1} \frac{\partial}{\partial z}$. $f^{\mu_{n+1}}$ is then obtained, whose BC is denoted by μ_{n+1} . Note that in each step, ϵ can be chosen so that $\|\mu_{n+1} - v\|_\infty$ is minimized. In practice, we choose $\epsilon = 1$ and it works well for all our numerical experiments. A sequence of quasi-conformal maps $\{f_n\}_{n=1}^\infty$ is obtained, whose BCs converge to v . We call such a process to deform f^μ to f^v iteratively the *Beltrami holomorphic flow (BHF)* from μ to v , and denote it by: **BHF**($\mu \rightarrow v$).

The Beltrami holomorphic flow can be summarized as follows:

Algorithm 1 : (*Beltrami holomorphic flow*)

Input : $f^\mu : \Omega_1 \rightarrow \Omega_2$ with BC μ , target BC v , threshold ϵ'

Output : Sequence of quasi-conformal maps $\{f^{\mu_n}\}_{n=1}^\infty$

1. Set $f^{\mu_0} = f^\mu$. Solve Equation (42) to obtain g_1 ;
2. Given f^{μ_n} , compute $\mu_n := \mu(f_n)$ and $v_n := (1 - \epsilon)\mu_n + \epsilon v$; solve Equation (42) to obtain g_{n+1} ; Set $f_{n+1} := f_n + g_{n+1}$;
3. If $\|\mu_{n+1} - \mu_n\| \geq \epsilon'$, repeat step 2. Otherwise, stop the iteration.

4.4 Summary of the Splitting Method

The proposed splitting method using the ADMM method can be summarized as follows:

Algorithm 2 : (*Splitting method for DOP, version 1*)

Input : Energy functional $E(f, v) = E_1(f) + E_2(v)$, threshold ϵ

Output : Optimal quasi-conformal map f^* with Beltrami coefficient v^*

1. Set $v_0 = 0$ and $f_0 = \text{harmonic map between } \Omega_1 \text{ and } \Omega_2$;
2. Given f_n , v_n , λ_n and ρ_n , compute f_{n+1} by:

$$f_{n+1} = \operatorname{argmin}_f L(f, v_n, \lambda_n, \rho_n);$$

3. Compute v_{n+1} by:

$$v_{n+1} = \operatorname{argmin}_v L(f_{n+1}, v, \lambda_n, \rho_n);$$

4. Compute λ_{n+1} and ρ_{n+1} ;
5. If $|E(f_{n+1}, v_{n+1}) - E(f_n, v_n)| \geq \epsilon$, repeat step 2. Otherwise, stop the iteration.

In the situation when both the subproblem (26) and subproblem (27) are difficult to solve, the following algorithm can be used.

Algorithm 3 : (*Splitting method for DOP, version 2*)

Input : Energy functional $E(f, v) = E_1(f) + E_2(v)$, thresholds ϵ_k and ϵ

Output : Optimal quasi-conformal map f^* with Beltrami coefficient v^*

1. Set $v_0 = 0$ and $f_0 = \text{harmonic map between } \Omega_1 \text{ and } \Omega_2$;
2. Given f_n, v_n, λ_n and ρ_n , put $\tilde{f} = f_n$ and $\tilde{v} = v_n$. Compute the gradient descent directions \mathbf{V}_1 and \mathbf{V}_2 to minimize $L(f, v_n, \lambda_n, \rho_n)$; Update $\tilde{f} := \tilde{f} + (\mathbf{V}_1 + \mathbf{V}_2)dt$;
3. Compute the gradient descent direction ∂v that minimizes $L(f_{n+1}, v, \lambda_n, \rho_n)$; Update $\tilde{v} := \tilde{v} + (\partial v)dt$;
4. If $|\nabla_f L + \nabla_v L| < \epsilon_k$, put $f_{n+1} = \tilde{f}$ and $v_{n+1} = \tilde{v}$; Go to step 5. Otherwise, put $f_n = \tilde{f}$ and $v_n = \tilde{v}$. Go to step 2.
5. Compute λ_{n+1} and ρ_{n+1} ;
6. If $|E(f_{n+1}, v_{n+1}) - E(f_n, v_n)| \geq \epsilon$, go to step 2. Otherwise, stop the iteration.

5 Numerical Implementation Details

In this section, we will explain in details the numerical implementation of the algorithms proposed in Sect. 4.

In practice, 2D domains or surfaces in \mathbb{R}^3 are usually represented discretely by triangular meshes. Suppose K_1 and K_2 are two meshes with the same topology representing S_1 and S_2 . We define the set of vertices on K_1 and K_2 by $V^1 = \{\mathbf{v}_i^1\}_{i=1}^n$ and $V^2 = \{\mathbf{v}_i^2\}_{i=1}^n$ respectively. Similarly, we define the set of triangular faces on K_1 and K_2 by $F^1 = \{T_j^1\}_{j=1}^m$ and $F^2 = \{T_j^2\}_{j=1}^m$.

5.1 Implementation Details of Algorithm 1

The major step in computing the Beltrami holomorphic flow as described in Algorithm 1 is to solve equation (42). We first discretize the operator \mathcal{A} in Eq. (42). Let $f = (u + \sqrt{-1}v) : K_1 \rightarrow K_2$. To compute \mathcal{A} , we simply need to approximate the partial derivatives at each face T . We denote them by $D_x f = D_x u + \sqrt{-1}D_x v$ and $D_y f = D_y u + \sqrt{-1}D_y v$ respectively. Note that f is piecewise linear. The restriction of f on each triangular face T can be written as:

$$f|_T(x, y) = \begin{pmatrix} a_T x + b_T y + r_T \\ c_T x + d_T y + s_T \end{pmatrix} \quad (43)$$

Clearly, $D_x u(T) = a_T$, $D_y u(T) = b_T$, $D_x v(T) = c_T$ and $D_y v(T) = d_T$. Now, the gradient $\nabla_T f := (D_x f(T), D_y f(T))^t$ on each face T can be computed by solving the linear system:

$$\begin{pmatrix} \mathbf{v}_1 - \mathbf{v}_0 \\ \mathbf{v}_2 - \mathbf{v}_0 \end{pmatrix} \nabla_T f = \begin{pmatrix} \frac{f(\mathbf{v}_1) - f(\mathbf{v}_0)}{|\mathbf{v}_1 - \mathbf{v}_0|} \\ \frac{f(\mathbf{v}_2) - f(\mathbf{v}_0)}{|\mathbf{v}_2 - \mathbf{v}_0|} \end{pmatrix}, \quad (44)$$

where $[\mathbf{v}_0, \mathbf{v}_1]$ and $[\mathbf{v}_0, \mathbf{v}_2]$ are two edges on T . By solving equation (44), a_T, b_T, c_T and d_T can be obtained. Hence on each face T ,

$$\nabla_T f = \frac{1}{2A} \sum_{j=1}^3 f(\mathbf{v}_j) \mathbf{s}_j, \quad (45)$$

where A is the area of T and

$$\begin{aligned} \mathbf{s}_1(T) &= \mathbf{n} \times (\mathbf{v}_3 - \mathbf{v}_2) \\ \mathbf{s}_2(T) &= \mathbf{n} \times (\mathbf{v}_1 - \mathbf{v}_3) \\ \mathbf{s}_3(T) &= \mathbf{n} \times (\mathbf{v}_2 - \mathbf{v}_1), \end{aligned} \quad (46)$$

where \mathbf{n} is the unit normal of T . Let $v(T)$ be a constant over the face T . Using the relations $\frac{\partial}{\partial z} = (D_x - \sqrt{-1}D_y)/2$ and $\frac{\partial}{\partial \bar{z}} = (D_x + \sqrt{-1}D_y)/2$, the operator \mathcal{A} can be discretized on each face T as follows:

$$\mathcal{A}f(T) = \frac{1}{4A} (1 - v(T), \sqrt{-1} + \sqrt{-1}v(T)) \sum_{j=1}^3 f(\mathbf{v}_j) \mathbf{s}_j. \quad (47)$$

Note that the right hand side of the above equation is linear in every $u(\mathbf{v}_j)$ and $v(\mathbf{v}_j)$, $j = 1, 2, 3$. Let $g(\mathbf{v}_i^1) = (P_i, Q_i)^t$ and $f^\mu(\mathbf{v}_i^1) = u_i + \sqrt{-1}v_i$, the optimization problem (42) can be discretized as minimizing:

$$\begin{aligned} \sum_{T_j^1 \in F^1} \frac{1}{4\text{Area}(T_j^1)} &\left| (1 - v(T_j^1), \sqrt{-1} + \sqrt{-1}v(T_j^1)) \sum_{i=1}^3 (P_{T_j^1(i)} + \sqrt{-1}Q_{T_j^1(i)}) \mathbf{s}_i(T_j^1) \right. \\ &\left. + (1 - v(T_j^1), \sqrt{-1} + \sqrt{-1}v(T_j^1)) \sum_{i=1}^3 (u_{T_j^1(i)} + \sqrt{-1}v_{T_j^1(i)}) \mathbf{s}_i(T_j^1) \right|^p, \end{aligned} \quad (48)$$

where $T_j^1(i)$ are the indices of the vertices of T_j^1 , i.e. $T_j^1 = [\mathbf{v}_{T_j^1(1)}^1, \mathbf{v}_{T_j^1(2)}^1, \mathbf{v}_{T_j^1(3)}^1]$.

Secondly, the boundary constraint can be approximated by a linear constraint. For each boundary vertex $\mathbf{v}_i^1 \in \gamma_j$, we only require $\mathbf{V}(\mathbf{v}_i^1)$ to be tangential to γ'_j at $f^\mu(\mathbf{v}_i)$. That is, if $g(\mathbf{v}_i^1) = (P_i, Q_i)^t$ and $(a_i, b_i)^t$ is the direction of the tangent, then

$$b_i P_i - a_i Q_i = 0, \quad (49)$$

which is a linear constraint. By putting $p = 2$, the optimization problem (48) together with the constraint (49) becomes a least square problem. For each iteration of Algorithm 1, $g_n(\mathbf{v}_i^1)$ is solved as above. Set $\tilde{f}_{n+1}(\mathbf{v}_i^1) := f_n(\mathbf{v}_i^1) + g_n(\mathbf{v}_i^1)$. For each boundary vertex $\mathbf{v}_i^1 \in \gamma_j$, it is not necessary that $\tilde{f}_{n+1}(\mathbf{v}_i^1) \in \gamma'_j$ because the boundary constraints are approximated. Nevertheless, when $\|\mathbf{v}_n - \mu_n\|_\infty$ is sufficiently small, $\tilde{f}_{n+1}(\mathbf{v}_i^1)$ shall not be far away from γ'_j . Hence we can project $\tilde{f}_{n+1}(\mathbf{v}_i^1)$ onto γ'_j and obtain the solution $f_{n+1}(\mathbf{v}_i^1)$ such that $f_{n+1}(\gamma_j) = \gamma'_j$, i.e.

$$f_{n+1}(\mathbf{v}_i^1) := \underset{\mathbf{z} \in \gamma'_j}{\text{argmin}} \|\tilde{f}_{n+1}(\mathbf{v}_i^1) - \mathbf{z}\|_2. \quad (50)$$

5.2 Implementation Details of Algorithm 2 and Algorithm 3

In this subsection, we give the numerical implementation details for Algorithms 2 and 3.

In the discrete setting, v_n , $\mu(f_n)$ and λ_n are all complex-valued functions defined on each triangular faces. More precisely, they are functions from F^1 to \mathbb{C} . ρ_n is a positive constant. Given a piecewise linear homeomorphism f_n between K_1 and K_2 , the value of $\mu(f_n)$ on each face T is given by

$$\mu(f_n)(T) = \frac{(1, i)\nabla_T f_n}{(1, -i)\nabla_T f_n}, \quad (51)$$

where ∇_T is given by (45).

The augmented Lagrangian L in (2) can then be discretized.

Step 2 of Algorithm 2 and Algorithm 3 requires us to compute the descent direction $\mathbf{V}_1 + \mathbf{V}_2$ that minimize $L(f_n, v_n, \lambda_n, \rho_n)$, where \mathbf{V}_1 is the descent direction for $E_1(f)$ and \mathbf{V}_2 is the descent direction for $<\lambda_{\mathbf{Re}}, \mathbf{Re}(v - \mu(f))> + <\lambda_{\mathbf{Im}}, \mathbf{Im}(v - \mu(f))> + \frac{\rho}{2}\|v - \mu(f)\|_2^2$.

The computation of \mathbf{V}_1 depends on the form of the energy $E_1(f)$. For example, if $E_1(f)$ is defined as the intensity mismatching error $\|I_1 - I_2(f)\|_2^2$, then on each vertex v_i , the value of \mathbf{V}_1 is given by $\mathbf{V}_1(v_i) = 2(I_1(v_i) - I_2(f_n(v_i)))\nabla I_2(f_n(v_i))$. Here we assume that we have the explicit forms of I_1 , I_2 and ∇I_2 so that we can evaluate their values on arbitrary points. Otherwise, if the values of I_1 and I_2 are only known on each vertices, we can use the method introduced in the last subsection to compute ∇I_2 on each face. Interpolations can be used to evaluate the values $\nabla I_2(f_n(v_i))$ and $I_2(f_n(v_i))$.

To compute \mathbf{V}_2 , we will use Algorithm 1 as follows. On each face T , the decent direction for $\mu(f_n)$ is given by

$$\partial\mu(T) = \text{Area}(T)[\lambda_n(T) + \rho_n(v_n(T) - \mu(f_n)(T))] \quad (52)$$

Use Algorithm 1, we obtain a quasiconformal map f' such that its Beltrami coefficient $\mu(f')$ is approximately equal to $\mu(f_n) + \partial\mu d\tau$, where $d\tau$ is a small positive number. Then we set $\mathbf{V}_2 = (f' - f_n)/d\tau$.

After computing \mathbf{V}_1 and \mathbf{V}_2 , we set $\tilde{f} = f_n + (\mathbf{V}_1 + \mathbf{V}_2)d\tau$ and project the boundary points to the corresponding boundaries. For Algorithm 2, we repeat the above computations but replacing f_n by \tilde{f} at each time. When the computations converge, we set $f_{n+1} = \tilde{f}$.

Now, step 3 of both Algorithm 2 and Algorithm 3 require us to minimize $L(f_{n+1}, v, \lambda_n, \rho_n)$ with respect to v :

$$E_2(v) + <\lambda_{\mathbf{Re}}^n, \mathbf{Re}(v - \mu(f_{n+1}))> + <\lambda_{\mathbf{Im}}^n, \mathbf{Im}(v - \mu(f_{n+1}))> + \frac{\rho_n}{2}\|v - \mu(f_{n+1})\|_2^2.$$

Depend on the form of $E_2(v)$, the minimizer can either be found by solving a linear system directly, or by the gradient descent method. For example, if $E_2(v) = \|\nabla v\|_2^2 + \|v\|_2^2$, then on each face, the minimizer \tilde{v} satisfies the linear equation

$$\mathcal{I}_v^f \Delta \mathcal{I}_f^v v - 2v - \lambda_{\mathbf{Re}}^n \mathbf{Re}(v - \mu(f_{n+1})) - i\lambda_{\mathbf{Im}}^n \mathbf{Im}(v - \mu(f_{n+1})) - \rho_n(v - \mu(f_{n+1})) = 0. \quad (53)$$

In the above, \mathcal{I}_f^v is the interpolating matrix that converts functions defined on faces into functions on vertices and \mathcal{I}_v^f is the interpolating matrix from vertices to faces. Δ is the Laplace operator constructed by the cotangent formula. Let $T_1 = [v_i, v_j, v_k]$ and $T_2 = [v_i, v_j, v_l]$. The mesh Laplacian is defined as:

$$\Delta(f(v_i)) = \sum_{T \in N_i} \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{2} (f(v_j) - f(v_i)) \quad (54)$$

where α_{ij} and β_{ij} are the two interior angles of T_1 and T_2 which are opposite to the edge $[v_i, v_j]$. To find α_{ij} and β_{ij} , we follow the idea of [34]. Let l_{ij} be the length of the edge $[v_i, v_j]$. By law of cosines: $l_{ij}^2 = l_{jk}^2 + l_{ki}^2 - 2l_{jk}l_{ki} \cos \alpha_{ij}$,

we have

$$\cos \alpha_{ij} = \frac{-l_{ij}^2 + l_{jk}^2 + l_{ki}^2}{2l_{jk}l_{ki}}. \quad (55)$$

Similar, by the law of sines: $\text{Area}(T_1) = \frac{1}{2}l_{jk}l_{ki} \sin \alpha_{ij}$, we have

$$\sin \alpha_{ij} = \frac{2\text{Area}(T_1)}{l_{jk}l_{ki}}. \quad (56)$$

Therefore we have

$$\cot \alpha_{ij} = \frac{-l_{ij}^2 + l_{jk}^2 + l_{ki}^2}{4\text{Area}(T_1)} \quad (57)$$

and the discrete Laplacian operator can then be constructed. Similarly, β_{ij} can be computed. The minimizer \tilde{v} can be computed by solving the above linear system (53), and we set $v_{n+1} = \tilde{v}$.

In the situation when $E_2(v)$ is not easy to minimize, we can use example the gradient descent to compute the minimizer. As an example, consider $E_2(v) = \|\nabla v\|_2^2 + \|v\|_p^p$. The descent direction for $v_n(T)$ is given by

$$\begin{aligned} \partial v = \text{Area}(T) & \left[\mathcal{I}_v^f \Delta \mathcal{I}_v^f v_n - p|v_n|^{p-2}v_n - \lambda_{\text{Re}}^n \mathbf{Re}(v_n - \mu(f_{n+1})) \right. \\ & \left. - i\lambda_{\text{Im}}^n \mathbf{Im}(v_n - \mu(f_{n+1})) - \rho_n(v_n - \mu(f_{n+1})) \right]. \end{aligned} \quad (58)$$

On each face T , we set $\tilde{v}(T) = v_n(T) + \partial v(T)d\tau$. For Algorithm 2, we repeat the procedure as above but replacing v_n by \tilde{v} . When the computation reach convergence, we set $v_{n+1} = \tilde{v}$.

Finally, λ_{n+1} and ρ_{n+1} are updated as follows. If the gradient $\|\nabla_f L + \nabla_v L\| > \epsilon_n$ for some predefined parameter ϵ_n , then we set $\lambda_{n+1} = \lambda_n$ and $\rho_{n+1} = \rho_n$. Otherwise we will check the magnitude of the residue $\|v_{n+1} - \mu(f_{n+1})\|_2$. If $\|v_{n+1} - \mu(f_{n+1})\|_2 < \eta_n$ for some parameter η_n , then we set

$$\lambda_{n+1} = \lambda_n + \rho_n(v_{n+1} - \mu(f_{n+1})) \quad (59)$$

and $\rho_{n+1} = \rho_n$. Otherwise, we will set $\lambda_{n+1} = \lambda_n$ and set $\rho_{n+1} = \rho_n(1 + \gamma_n)$, where γ_n is some positive number.

6 Experimental Results

6.1 Solving DOPs on Simply-Connected Domains

Example 1 We first test our proposed algorithm to compute a diffeomorphism of the unit disk $\mathbb{D} = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 = 1\}$ matching the intensity functions.

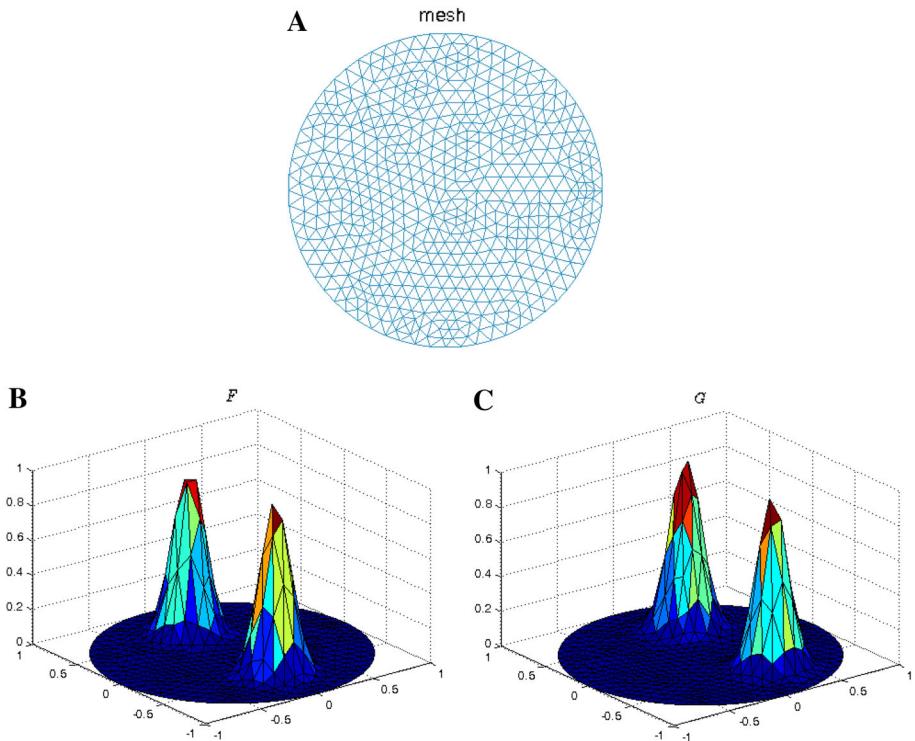


Fig. 2 Example 1: **a** shows the input mesh. **b** Shows the intensity $F(x, y)$ defined on \mathbb{D} . **c** Shows the intensity $G(x, y)$ defined on \mathbb{D}

Define $F : \mathbb{D} \rightarrow \mathbb{R}$ and $G : \mathbb{D} \rightarrow \mathbb{R}$ by:

$$\begin{aligned} F(x, y) &= \exp(-40x^2 - 40(y - 0.5)^2) + \exp(-40x^2 - 40(y + 0.5)^2) \\ G(x, y) &= \exp(-40(x - 0.25)^2 - 40(y - 0.5)^2) + \exp(-40(x - 0.25)^2 - 40(y + 0.5)^2) \end{aligned} \quad (60)$$

We proceed to look for a diffeomorphism $f^* : \mathbb{D} \rightarrow \mathbb{D}$ that matches the intensities by minimizing:

$$E(f) = \alpha \|F - G(f)\|_2^2 + \beta \|\mu(f)\|_4^4 \quad (61)$$

(with $\alpha = 50$ and $\beta = 0.0005$).

Using the proposed splitting algorithm, we solve the above DOP with stopping criteria as $\|f_{k+1} - f_k\|_1 < 10^{-5}$ and $\|\mu_{k+1} - \mu_k\|_1 < 10^{-5}$. Figure 2a shows the input mesh of \mathbb{D} . (b) and (c) shows the intensity $F(x, y)$ and $G(x, y)$ respectively. The optimal diffeomorphism f^* obtained from our proposed algorithm is shown in Fig. 3a (which is visualized as an output mesh obtained by deforming the input mesh using the optimal diffeomorphism). Figure 3b shows the intensity function $G \circ f^*(x, y)$, which closely resembles to $F(x, y)$. It means the registration matches the intensity functions well. Figure 4a–c shows the intensity mismatching energy, conformality distortion and total energy versus iterations respectively. Note that the number of iterations to minimize the subproblems might differ in each ADMM

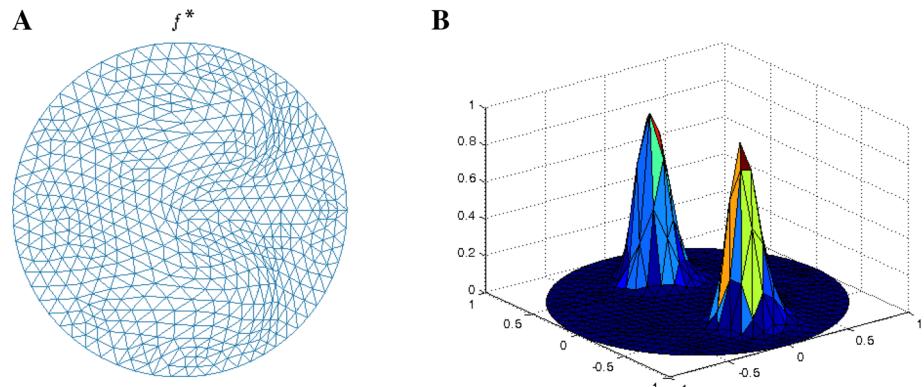


Fig. 3 Example 1: **a** shows the output mesh, obtained by deforming the input mesh by the optimal diffeomorphism f^* . **b** Shows the intensity function $G \circ f^*(x, y)$ defined on \mathbb{D} , which closely resembles to $F(x, y)$

iterations. Hence, in our plots, the energy values at each sub-iterations are also shown. The red dots in (c) indicate the energy values at each actual ADMM iterations. With that, the number of sub-iterations in each ADMM iterations can be demonstrated. From the energy plots, we observe that the intensity mismatching error reduces as iterations increase. The conformality distortion increases as iterations increase, since the initial map is the identity map with no conformality distortion. Conformality distortion is iteratively induced in order to tolerate for more intensity matching.

We also compare our proposed method with the algorithm in [1]. The result is shown in Fig. 5. The optimal diffeomorphism is shown in Fig. 5a, which looks similar to the one we obtain (Fig. 2a). However, the intensity mismatching error at the optimal state is higher than that of our proposed algorithm, as shown in (b). In fact, the total energy at the optimal state is higher than that of our proposed method, as shown in (c). It indicates that the proposed splitting method can optimize the map to a smaller energy value.

Example 2 In this example, we test our algorithm to compute the landmark-matching diffeomorphism f^* of \mathbb{D} . Figure 6a shows the input mesh. We look for a diffeomorphism that moves the initial landmark point p (labeled by \circ) to the target landmark point q (labeled by \times). We compute the diffeomorphism by minimizing:

$$E(f) = \alpha|f(p) - q|^2 + \beta\|\mu(f)\|_2^2 \quad (62)$$

(with $\alpha = 100$ and $\beta = 10$).

Using the splitting method, we solve the above DOP and obtain the optimal diffeomorphism f^* , which is shown in Fig. 6b. Landmark point is matched consistently. We also optimize the DOP using the algorithm in [1]. The obtained diffeomorphism is shown in Fig. 6c. Note that the landmark point cannot be matched consistently. Figure 7a, b show the landmark mismatching error and conformality distortion versus iterations using the splitting method. Note that the landmark mismatching error converges to 0. (c) shows the total energy versus iterations. The red dots indicate the actual ADMM iterations. Figure 7d–f show the landmark mismatching error, conformality distortion and total energy versus iterations using the integral method proposed in [1]. The landmark mismatching error converges at about 0.12. The algorithm get stuck at a local minimum with higher total energy value than our proposed splitting method.

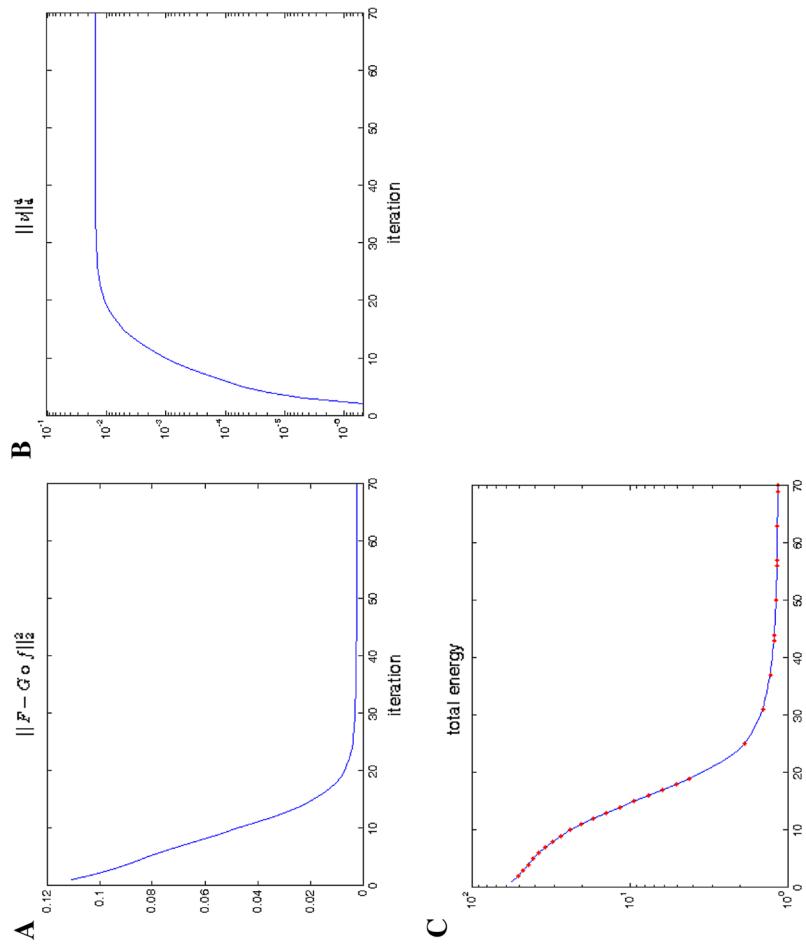


Fig. 4 Example 1: **a** shows the intensity mismatching error versus iterations. **b** Shows the conformality distortion versus iterations. **c** Shows the total energy versus iterations. The red dots indicate each actual ADMM iterations (Color figure online)

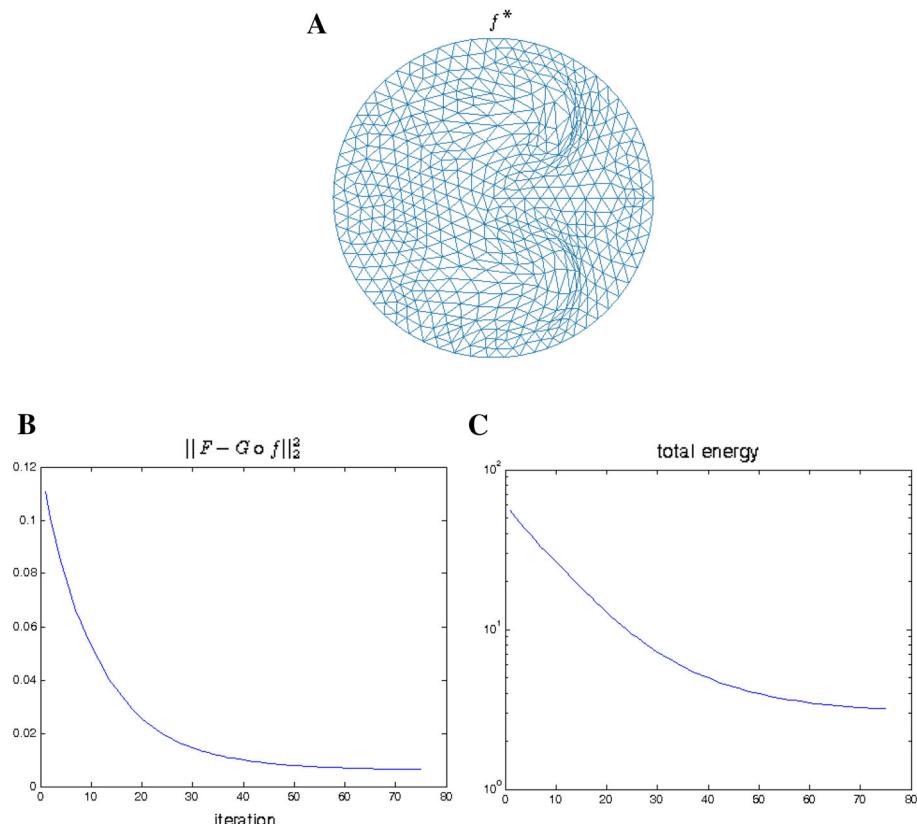


Fig. 5 Example 1: **a** shows the output mesh by deforming the input mesh by the optimal diffeomorphism obtained using the method in [1]. **b** shows the intensity mismatching error versus iterations. **c** shows the total energy versus iterations

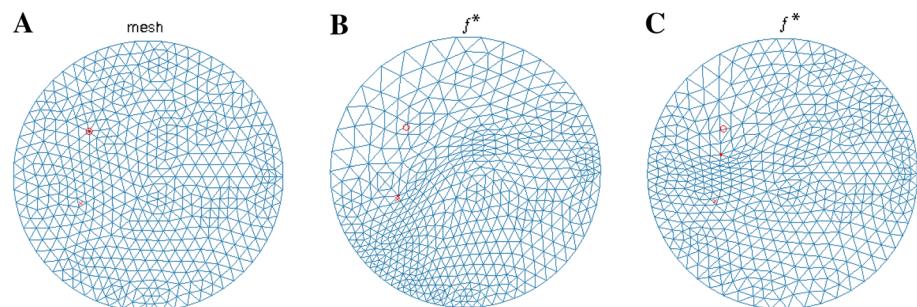


Fig. 6 Example 2: **a** shows the input mesh, with the initial landmark point and target landmark point labeled by \circ and \times respectively. **b** shows the output mesh obtained by deforming the initial mesh with the landmark-matching diffeomorphism using the proposed splitting method. The point labeled by \circ is moved to \cdot under the diffeomorphism. **c** shows the output mesh obtained by deforming the initial mesh with the landmark-matching diffeomorphism using the integral method in [1]

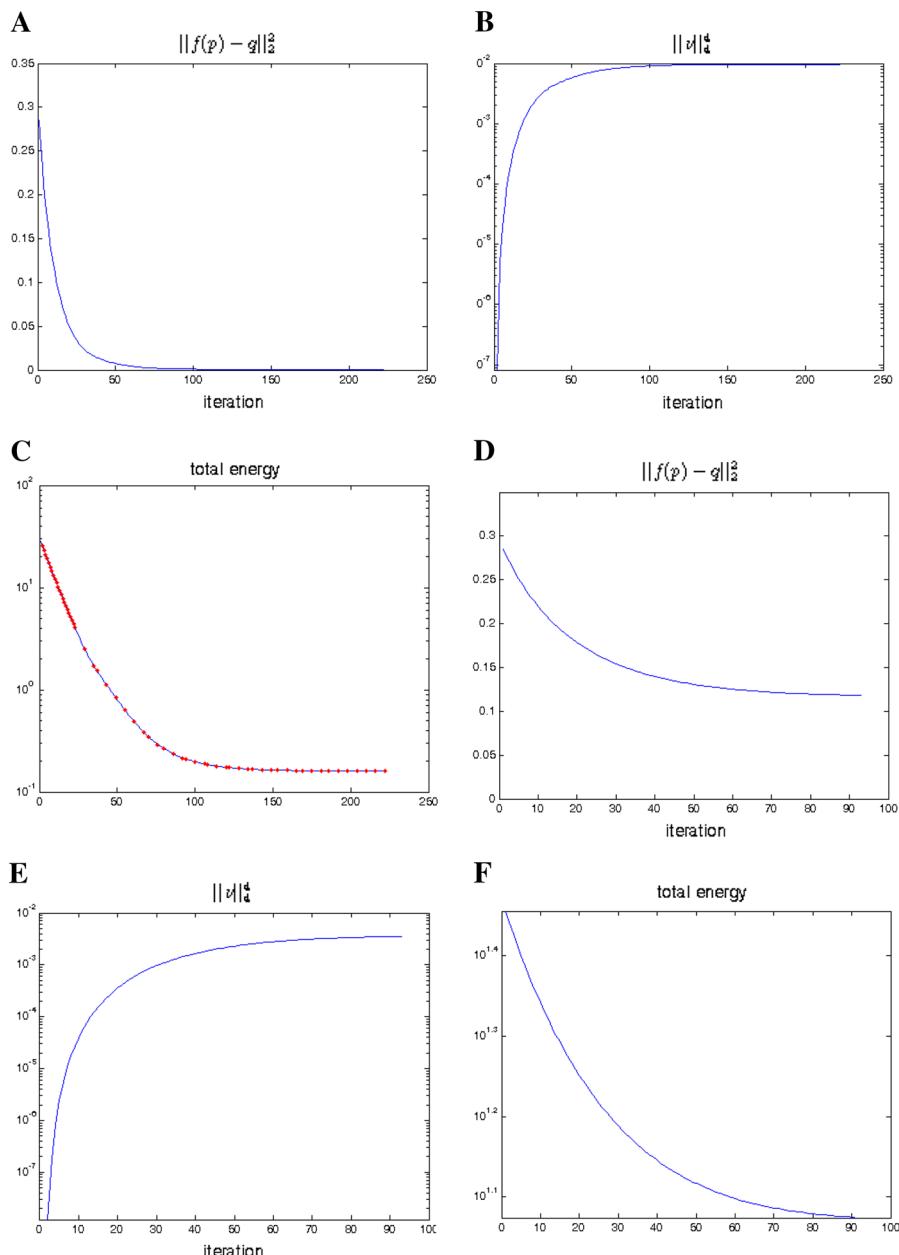
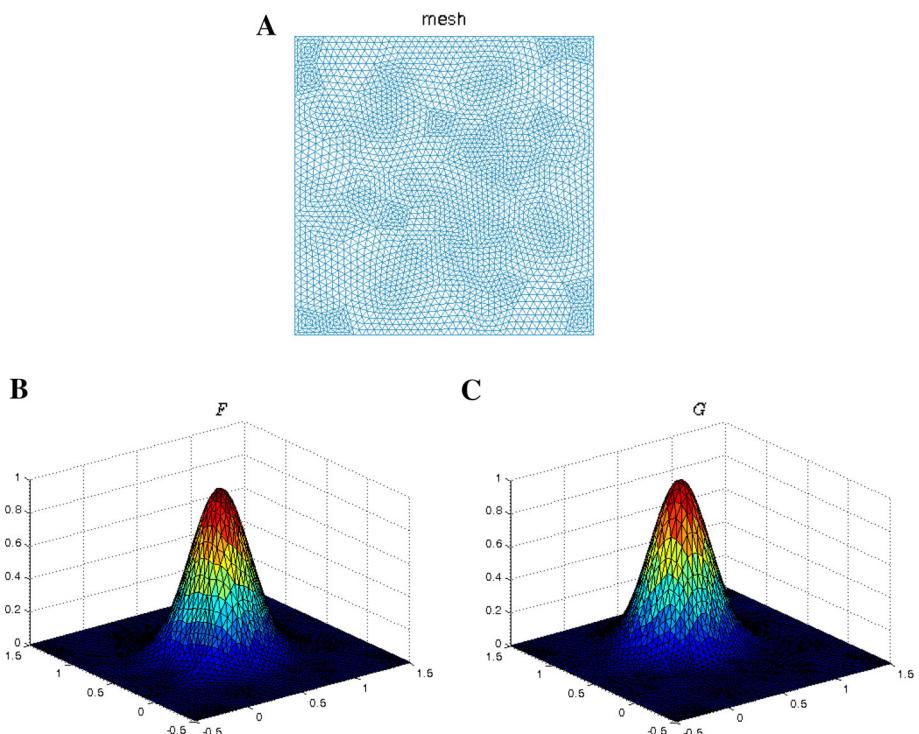


Fig. 7 Example 2: **a–c** show the landmark mismatching error, conformality distortion and total energy versus iterations respectively using the splitting method. **d–f** show the landmark mismatching error, conformality distortion and total energy versus iterations respectively using the integral method in [1]

Example 3 We have also compared our proposed splitting method to solve the DOP with the method proposed in [1] on more examples. Two experiments to compute intensity matching diffeomorphisms (similar to Example 1) and three experiments to compute landmark

Table 1 Comparisons between the proposed splitting method and the previous method in [1]

Examples	Splitting method		Integral method [1]	
	Time taken (s)	Final energy	Time taken (s)	Final energy
Intensity matching 1	19	1.46×10^{-3}	1680	4.03×10^{-3}
Intensity matching 2	23	8.88×10^{-3}	1650	2.27×10^{-2}
Landmark matching 1	116	2.03×10^{-4}	1731	1.50×10^{-2}
Landmark matching 2	78	4.18×10^{-4}	2163	2.99×10^{-2}
Landmark matching 3	77	2.91×10^{-4}	2356	2.13×10^{-1}

**Fig. 8** Example 4: **a** shows the input mesh. **b** Shows the intensity $F(x, y)$ defined on R . **c** Shows the intensity $G(x, y)$ defined on R

matching diffeomorphisms (similar to Example 2) have been carried out. Table 1 records the computational times to solve the DOP using the two different methods on five different examples. Results show that our proposed splitting method is much more efficient than the previous approach. Also, the optimal solutions we obtained using our proposed splitting method have much smaller final energy values.

Example 4 We test our algorithm on a rectangular domain $R = [-0.5, 1.5] \times [-0.5, 1.5]$. We look for an optimal diffeomorphism $f^* : R \rightarrow R$ that matches two intensity functions $F : R \rightarrow \mathbb{R}$ and $G : R \rightarrow \mathbb{R}$ defined by:

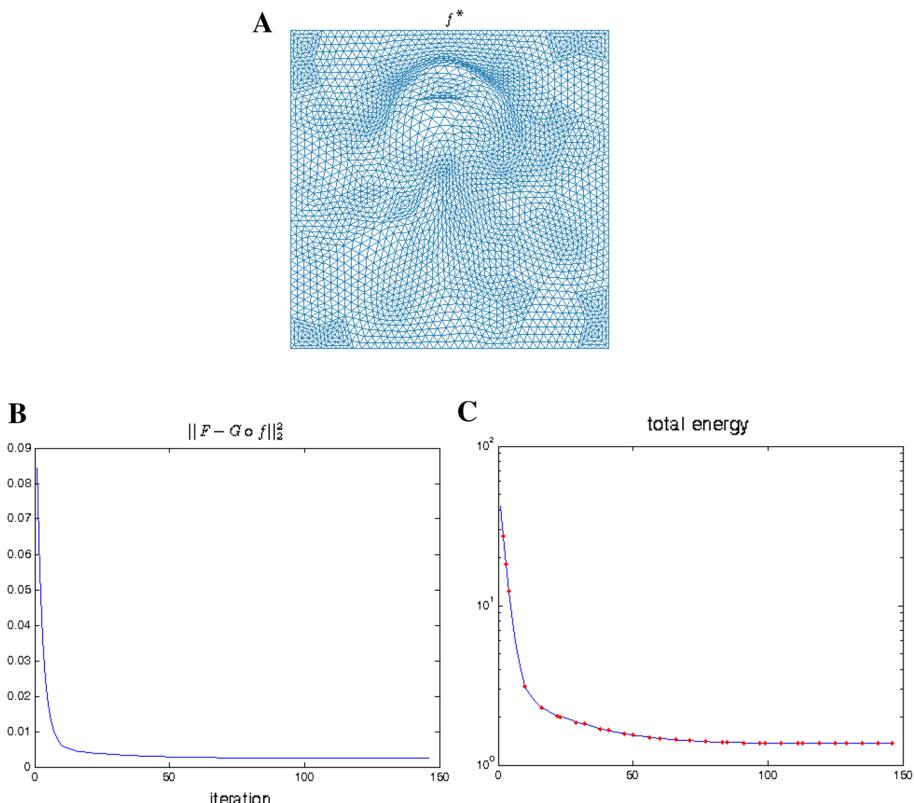


Fig. 9 Example 4: **a** shows the output mesh deformed from the initial mesh with the intensity matching diffeomorphism. **b** Shows the intensity mismatching error versus iterations. **c** Shows the total energy versus iterations. The red dots indicate the actual ADMM iterations (Color figure online)

$$\begin{aligned} F(x, y) &= \exp(-10(x - 0.5)^2 - 10(y - 0.5)^2) \\ G(x, y) &= \exp(-10(x - 0.5)^2 - 10(y - 0.75)^2) \end{aligned} \quad (63)$$

We minimize: $E(f) = 500\|F - G(f)\|_2^2 + 5\|\mu(f)\|_4^4$. Figure 8a shows the input mesh of R . (b) and (c) shows the intensity functions $F(x, y)$ and $G(x, y)$ respectively. Using the splitting method, we solve the above DOP and obtain the diffeomorphism f^* , which is shown in Fig. 9a. (b) shows the intensity mismatching error versus iterations. (c) shows the total energy versus iterations. Again, the red dots indicate the energy values at the actual ADMM iterations.

Example 5 In this example, we solve the DOP to find a quasi-conformal map $f^* : \mathbb{D} \rightarrow \mathbb{D}$ of the unit disk \mathbb{D} with BC = v that minimizes: $E(f) = \|\mu(f) - v\|_4^4$. We set $v = 0.3$. Figure 10a shows the input mesh. By solving the DOP, we find the optimal diffeomorphism f^* minimizing E and deform the input mesh through f^* . The deformed mesh is shown in (b). The total energy versus iterations is shown in (c). The red dots indicate the actual ADMM iterations. (d) shows the histogram of the norm of the BC, which accumulates at 0.3 as desired.

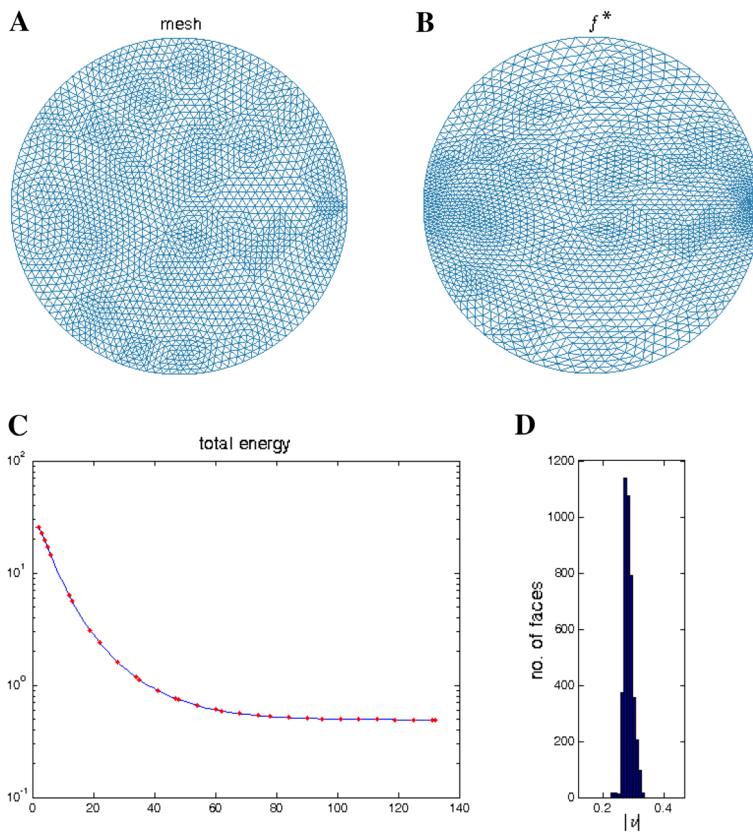


Fig. 10 Example 5: **a** shows the input mesh. **b** Shows the output mesh deformed by the optimal diffeomorphism. **c** Shows the total energy versus iterations. **d** shows the histogram of the norm of the BC, which accumulates at 0.3 as desired

6.2 Solving DOPs on Multiply-Connected Domains

Example 6 We test our proposed algorithm on a multiply-connected domain Ω . In this example, we solve the DOP to find a quasi-conformal map $f^* : \Omega \rightarrow \Omega$ such that its BC $\mu(f)$ minimizes $E(f) = ||\mu(f) - v||_4^4$, where $v = 0.3$. Figure 11a shows the input mesh. By solving the DOP, we find the optimal diffeomorphism f^* minimizing E and deform the input mesh through f^* . The deformed mesh is shown in (b). The total energy versus iterations is shown in (c). The red dots indicate the actual ADMM iterations. Note that a quasi-conformal map of a multiply-connected domain with a given Beltrami coefficient v may not exist. By minimizing E , one can find a diffeomorphism whose BC is closest to the given BC v in the L^4 -sense. (d) shows the histogram of the norm of the BC of the optimal diffeomorphism.

Example 7 In this example, we test our proposed algorithm to find an intensity matching diffeomorphism of a multiply-connected domain. Figure 12a shows an input mesh of a triply-connected domain Ω . Define $F : \Omega \rightarrow \mathbb{R}$ and $G : \Omega \rightarrow \mathbb{R}$ by:

$$\begin{aligned} F(x, y) &= \exp(-10(x - 0.5)^2 - 10(y - 0.5)^2) \\ G(x, y) &= \exp(-10(x - 0.5)^2 - 10(y - 0.75)^2), \end{aligned} \quad (64)$$

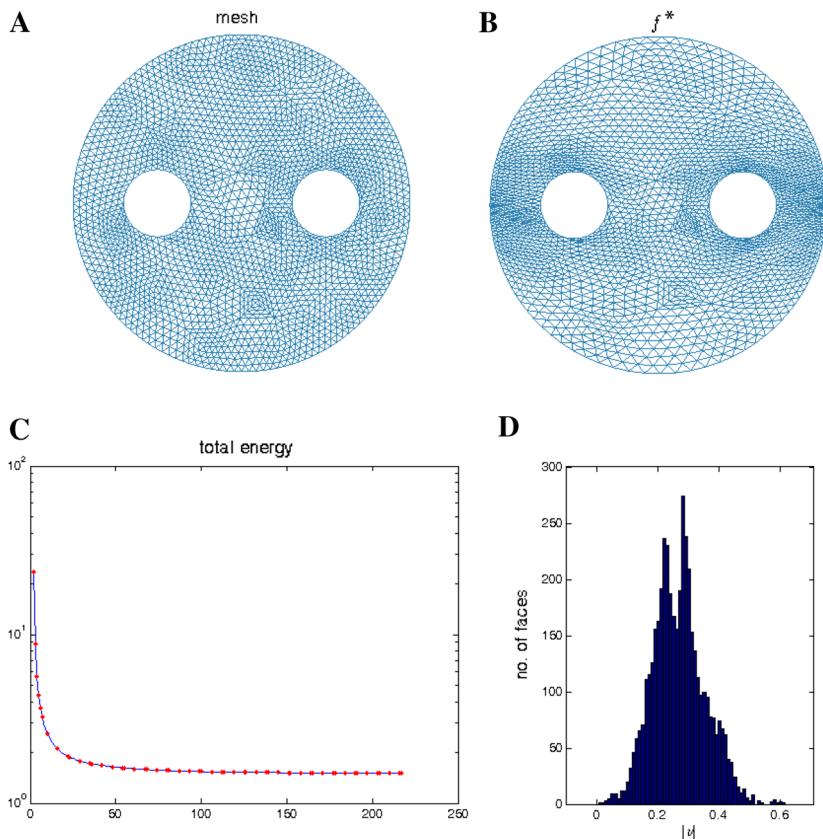


Fig. 11 Example 6: **a** shows the input mesh of a triply-connected domain. **b** Shows the output mesh deformed from the initial mesh using the optimal diffeomorphism. **c** Shows the total energy versus iterations. **d** shows the histogram of the norm of the BC of the optimal diffeomorphism

which are shown in Fig. 12b, c.

We proceed to look for a diffeomorphism $f^* : \Omega \rightarrow \Omega$ that matches the intensities by minimizing:

$$E(f) = \alpha ||G(f) - F||_2^2 + \beta ||\mu(f)||_4^4 \quad (65)$$

(where $\alpha = 500$, $\beta = 5$). Using the splitting method, we obtain the optimal diffeomorphism f^* as shown in Fig. 13a. Figure 13b–d shows the intensity mismatching error, conformality distortion and total energy versus iterations. Next, we take $\alpha = 50$ and $\beta = 5$. In other words, we like to obtain more conformality. Figure 14a shows the obtained diffeomorphism. Note that the diffeomorphism has less squeezing than that in Fig. 12a. Figure 14b, c shows the intensity mismatching error and conformality distortion versus iterations. As shown in the energy plots, the conformality distortion at the optimal state is much less than the previous results. However, the intensity mismatching error at the optimal state is higher than the previous result. Note that the algorithm in [1] cannot be applied to multiply-connected domains. Thus, this problem cannot be solved by the method in [1].

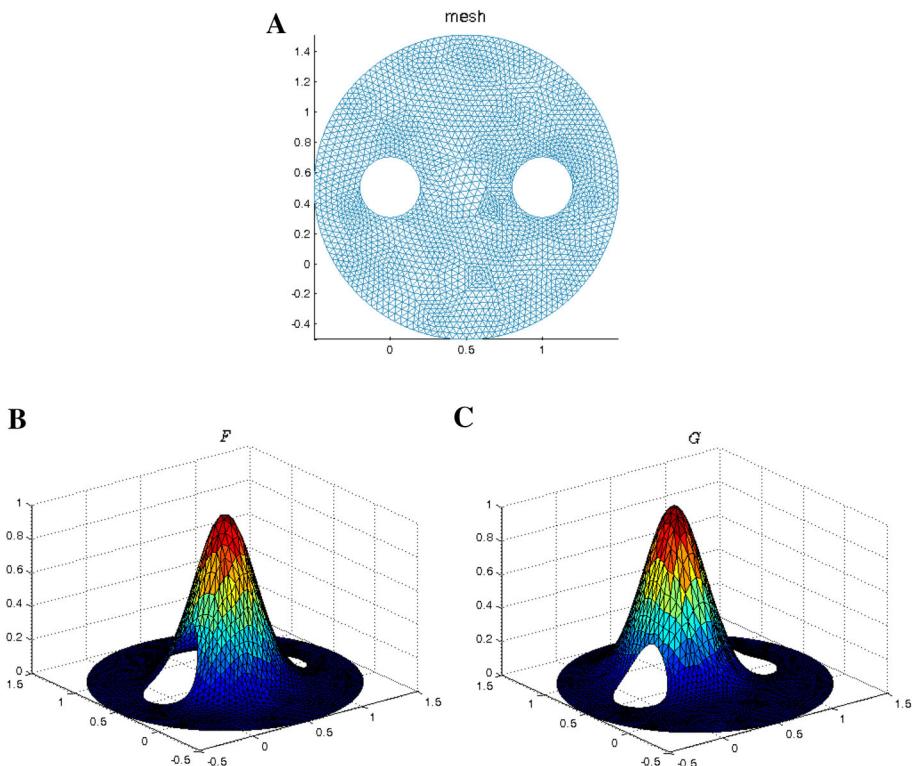


Fig. 12 Example 7: **a** shows the input mesh. **b** shows the intensity $F(x, y)$. **c** shows the intensity $G(x, y)$

Example 8 In this example, we test the algorithm to compute the landmark matching diffeomorphism of a triply-connected domain Ω . Figure 15a shows an input mesh of Ω . We look for a diffeomorphism $f^* : \Omega \rightarrow \Omega$ that matches the initial landmark points $\{p_1, p_2\}$ (labeled by \circ) to the target landmark points $\{q_1, q_2\}$ (labeled by \times). The obtained diffeomorphism using the splitting method is shown in Fig. 15b, which matches landmark consistently. Figure 16a, b shows the landmark mismatching error and the conformality distortion versus iterations respectively.

Example 9 In this example, we test the algorithm to compute a diffeomorphism $f^* : \Omega \rightarrow \Omega$ that matches both intensity functions and landmark constraints on a triply-connected domain Ω . Figure 17a shows the input mesh of Ω . The initial landmarks $\{p_1, p_2\}$ are labeled by \circ and the target landmarks $\{q_1, q_2\}$ are labeled by \times . The intensity functions $F : \Omega \rightarrow \mathbb{R}$ and $G : \Omega \rightarrow \mathbb{R}$, which are shown in Fig. 17b, c, are defined as in Eq. (63). Using the splitting method, we compute f^* by minimizing the following energy functional:

$$E(f) = \alpha \sum_{i=1}^2 |f(p_i) - q_i|^2 + \beta \|G(f) - F\|_2^2 + \gamma \|\mu(f)\|_4^4 \quad (66)$$

(where $\alpha = 10^2$, $\beta = 10^2$ and $\gamma = 10^{-2}$). The obtained diffeomorphism is shown in Fig. 18a. Landmarks are matched consistently. (b) shows the intensity function $G \circ f^*(x, y)$, which closely resembles to $F(x, y)$. It demonstrates the optimal diffeomorphism matches the intensity functions. Figure 19a–d show the conformality distortion, intensity mismatching error,

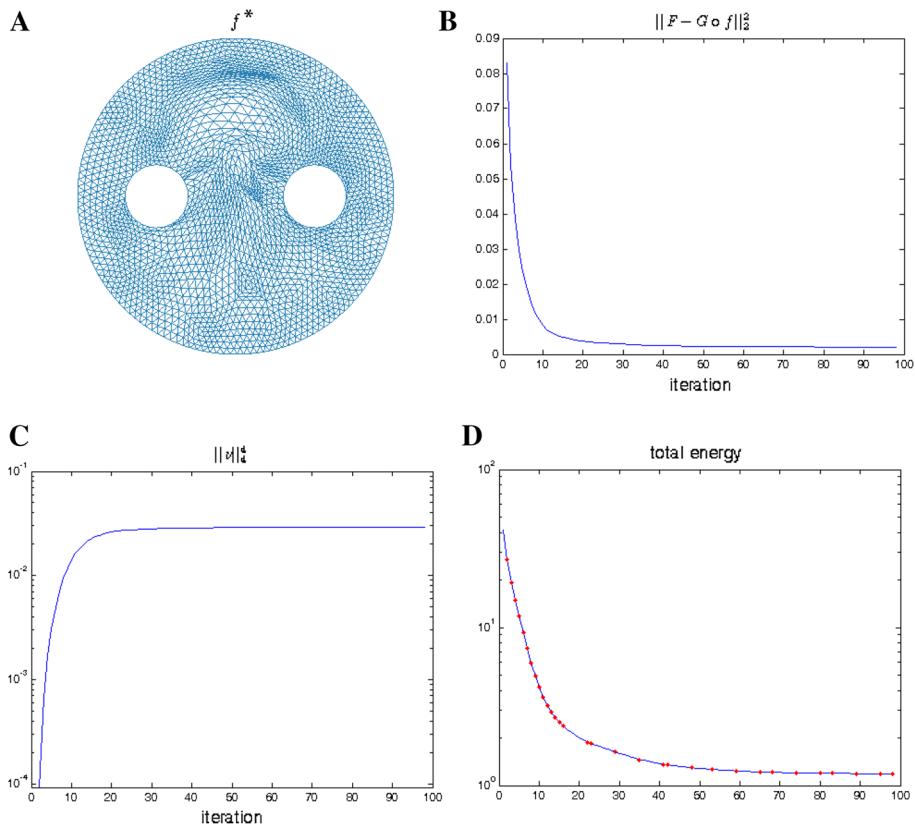


Fig. 13 Results of Example 7: **a** shows the output mesh deformed from the initial mesh using the intensity matching diffeomorphism. **b** Shows the intensity mismatching error versus iterations. **c** Shows the conformality distortion versus iterations. **d** shows the total energy versus iterations. The red dots indicate the actual ADMM iterations (Color figure online)

landmark mismatching error and total energy versus iterations respectively. The landmark mismatching error converges to 0 at the optimal state.

Example 10 This example tests our proposed algorithm to find an intensity matching diffeomorphism between two triply-connected domains Ω_1 and Ω_2 with different geometries. Figure 20a shows the input mesh representing Ω_1 . Figure 20b shows the target domain Ω_2 and the initial map, which is chosen to be the Teichmüller map [38]. Define $F : \Omega \rightarrow \mathbb{R}$ and $G : \Omega \rightarrow \mathbb{R}$ by:

$$\begin{aligned} F(x, y) &= \exp(-20x^2 - 20y^2) \\ G(x, y) &= \exp(-5x^2 - 5y^2), \end{aligned} \quad (67)$$

which are shown in Fig. 20c, d. We then look for a diffeomorphism $f^* : \Omega_1 \rightarrow \Omega_2$ that minimizes:

$$E(f) = \alpha \|G(f) - F\|_2^2 + \beta \|\mu(f)\|_6^6 \quad (68)$$

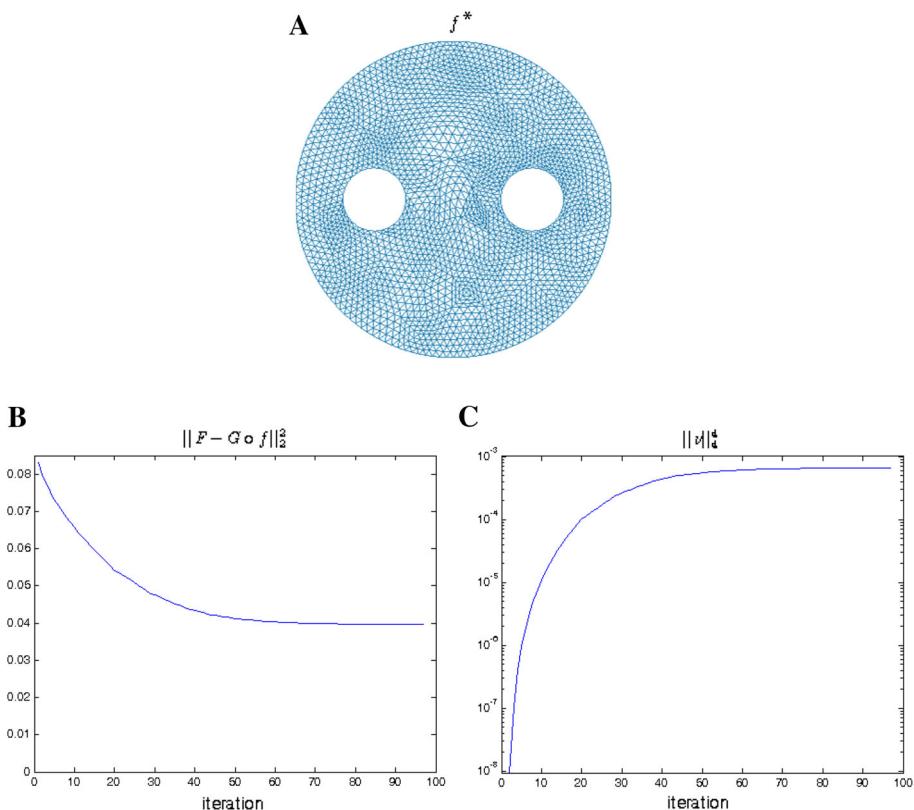


Fig. 14 Results of Example 7 with different parameters (as to preserve more conformality). **a** Shows the output mesh deformed from the initial mesh using the optimal diffeomorphism. The output mesh has less squeezing. **b** Shows the intensity mismatching error versus iterations. **c** Shows the conformality distortion versus iterations

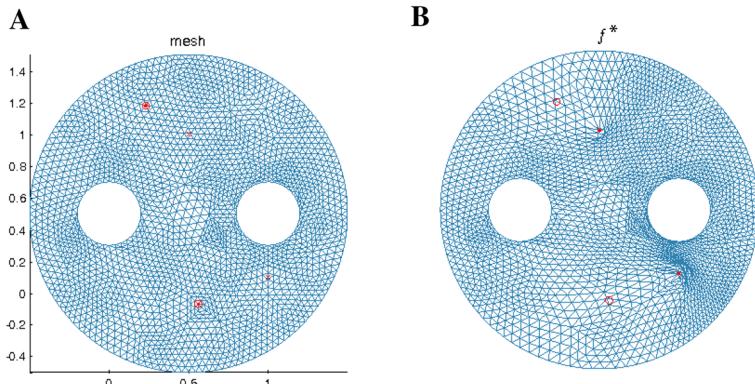


Fig. 15 Example 8: **a** shows the input mesh. The initial landmark point and target landmark point are labeled by open circle and multiplication sign respectively. **b** Shows the output mesh deformed by the optimal diffeomorphism. The initial landmark points (labeled by open circle) are moved to dot, which are close to the target landmark points (labeled by dot)

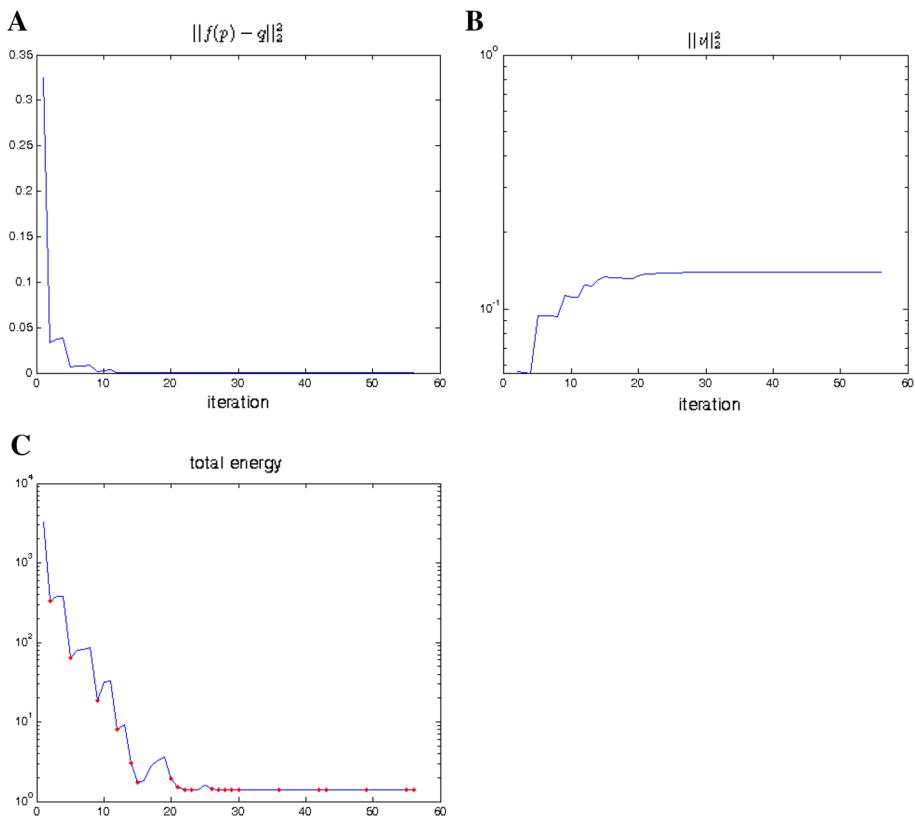


Fig. 16 Example 8: **a** shows the landmark mismatching error versus iterations. **b** Shows the conformality distortion versus iterations. **c** Shows the total energy versus iterations. The red dots indicate the actual ADMM iterations (Color figure online)

(where $\alpha = 5$, $\beta = 5$). Using the splitting method, we obtain the optimal diffeomorphism f^* as shown in Fig. 21A. Figure 21b,c show the intensity mismatching error and total energy versus iterations.

Example 11 In this example, we test the algorithm to compute the landmark matching diffeomorphism between two multiply-connected domains Ω_1 and Ω_2 with different geometries. Both Ω_1 and Ω_2 have three inner holes but with different sizes and at different locations. Figure 22a shows an input mesh of Ω_1 . Figure 22b shows the target domain Ω_2 and the initial map, which is again chosen to be the Teichmüller map [38]. We look for a diffeomorphism $f^* : \Omega_1 \rightarrow \Omega_2$ that matches the initial landmark points $\{p_1, p_2\}$ (labeled by \circ) to the target landmark points $\{q_1, q_2\}$ (labeled by \times). The obtained diffeomorphism using the splitting method is shown in Fig. 22c, which matches landmark consistently. Figure 23a–c shows the landmark mismatching error, conformality distortion and the total energy versus iterations respectively.

6.3 Solving DOPs on Riemann Surfaces

Example 12 Our proposed method can be easily applied to solving DOPs on Riemann surfaces through conformal parameterizations. Every connected Riemann surfaces can be para-

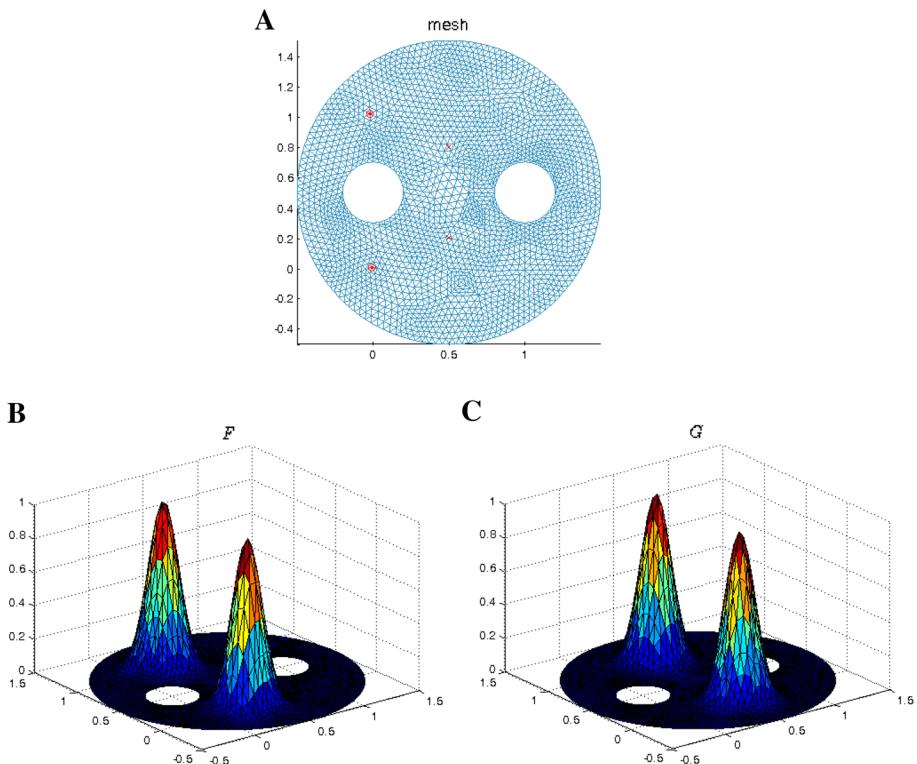


Fig. 17 Example 9: Landmark and intensity matching diffeomorphism. **a** Shows the input mesh. The initial landmark points and target landmark points are labeled by *open circle* and *multiplication sign* respectively. **b** shows the intensity $F(x, y)$. **c** shows the intensity $G(x, y)$

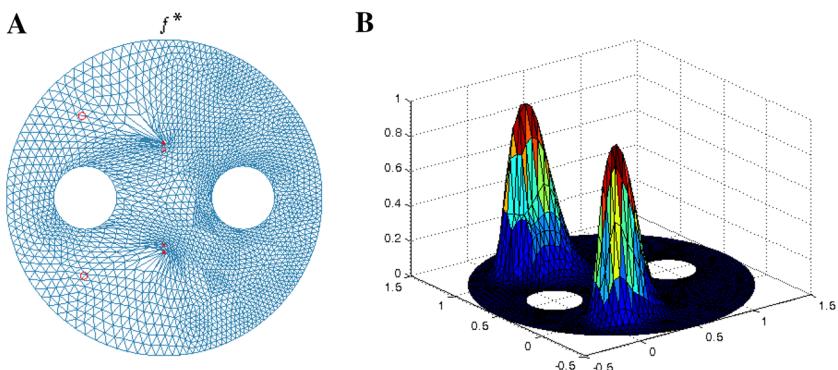


Fig. 18 Example 9: **a** shows the output mesh obtained by deforming the initial mesh using the optimal diffeomorphism. **b** Shows the intensity function $G \circ f^*(x, y)$, which closely resembles to $F(x, y)$. It demonstrates the optimal diffeomorphism matches the intensity functions

meterized conformally onto the unit disk \mathbb{D} or multiply-connected punctured disk. Solving the DOP on Riemann surfaces is then equivalent to solving a DOP on the 2D domains. To illustrate the idea, we test our method to find an optimized registration between two human

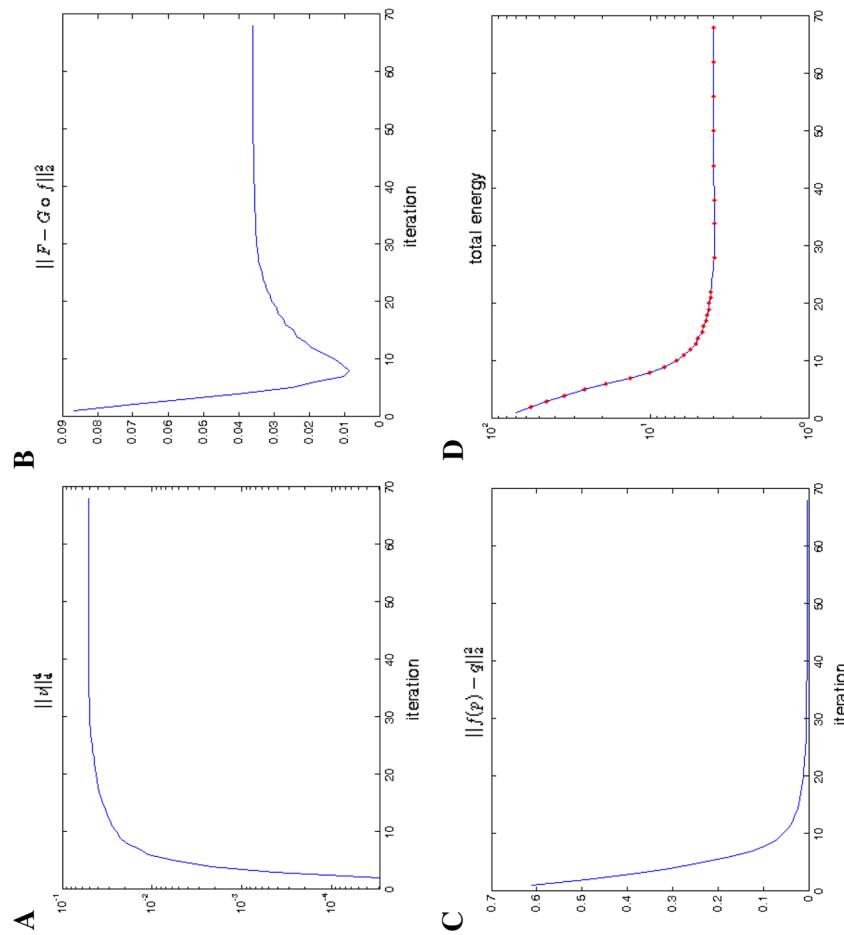


Fig. 19 Example 9. **a** shows the conformality distortion versus iterations. **b** Shows the intensity mismatching error versus iterations. **c** Shows the landmark mismatching error versus iterations. **d** Shows the total energy versus iterations. The red dots indicate the actual ADMM iterations (Color figure online)

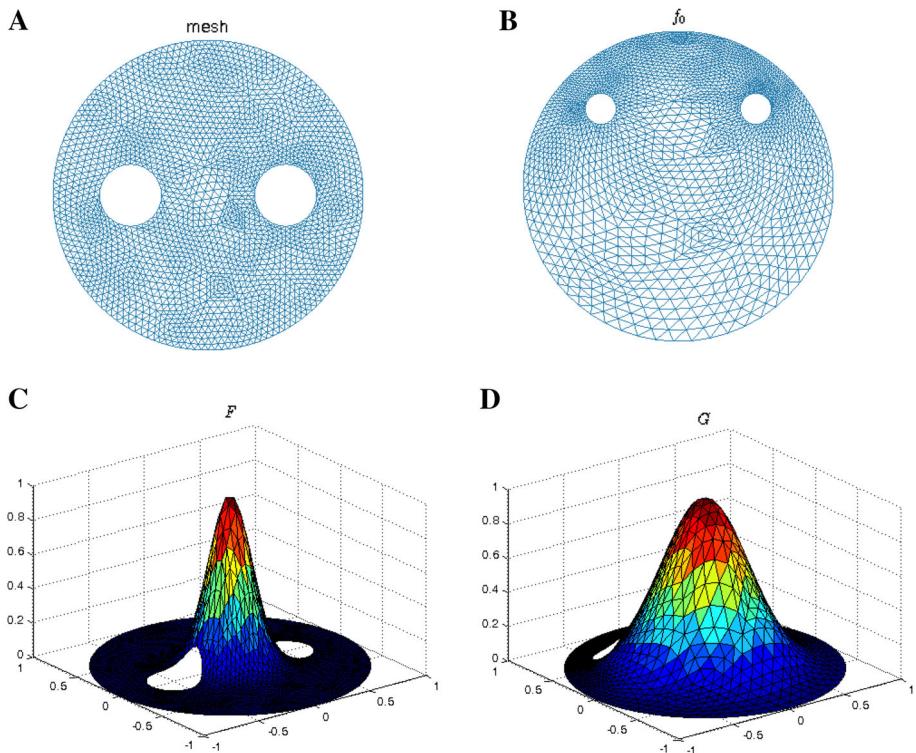


Fig. 20 Example 10: **a** shows the input mesh. **b** Shows the target domain and the initial map. **c** Shows the intensity $F(x, y)$. **d** Shows the intensity $G(x, y)$

faces that matches both landmarks and curvatures. Figure 24a, b show two human faces, with corresponding landmarks labeled on each of them. The colormaps on each surfaces are given by their mean curvatures. Their conformal parameterizations are shown in (c) and (d) respectively. We find a diffeomorphism that minimizes the following energy functional:

$$E(f) = \alpha \sum_{i=1}^2 |f(p_i) - q_i|^2 + \beta ||H_1(f) - H_2||_2^2 + \gamma ||\mu(f)||_4^4 \quad (69)$$

where $\alpha = 10^4$, $\beta = 10^2$, $\gamma = 10^2$ and H_1 and H_2 are the mean curvature functions on Face 1 and Face 2 respectively. Note that H_1 and H_2 are not smoothly defined in closed forms.

Figure 25a shows the registration result. The colormap (mean curvature) on Face 1 is mapped to Face 2 using the computed registration. Note that the high curvature regions (red colored regions) on Face 1 are mapped to corresponding high curvature regions on Face 2. Landmarks on Face 1 are also mapped to Face 2 using the obtained registration. Note that landmarks are matched consistently. (b) shows registration result on the conformal parameter domain. (c) shows the registration result visualized by the texture map. The energy plots of intensity mismatching error, landmark mismatching error and the total energy versus iterations are shown in Fig. 26a–c respectively. The red dots indicate the energy values at the actual ADMM iterations.

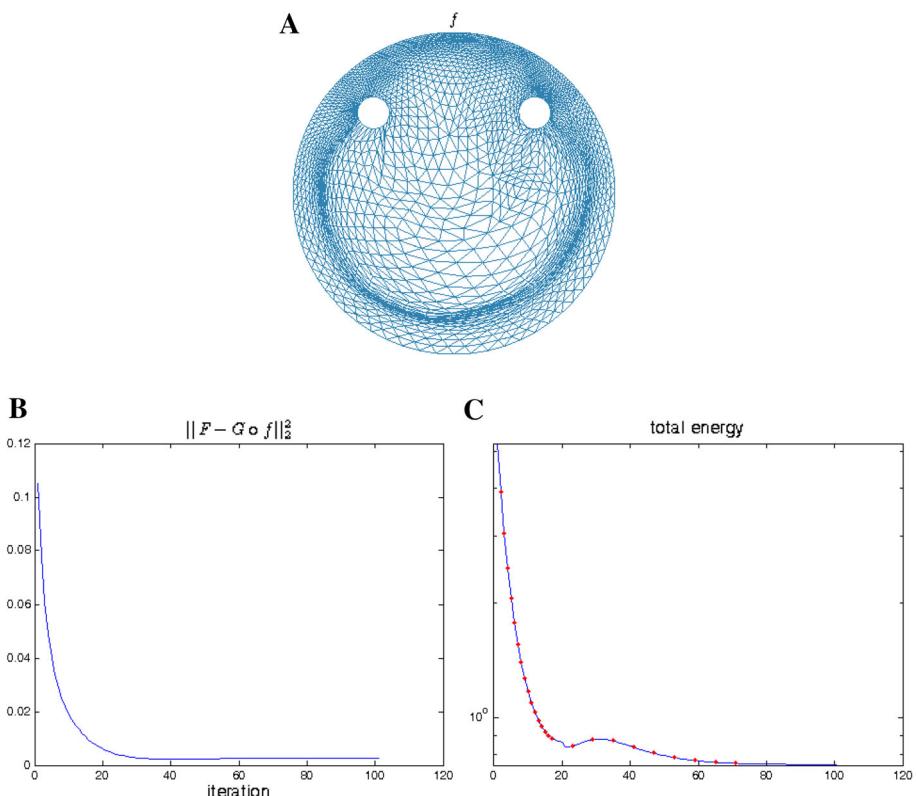


Fig. 21 Results of Example 10: **a** shows the output mesh deformed from the initial mesh using the intensity matching diffeomorphism. **b** Shows the intensity mismatching error versus iterations. **c** Shows the total energy versus iterations. The red dots indicate the actual ADMM iterations (Color figure online)

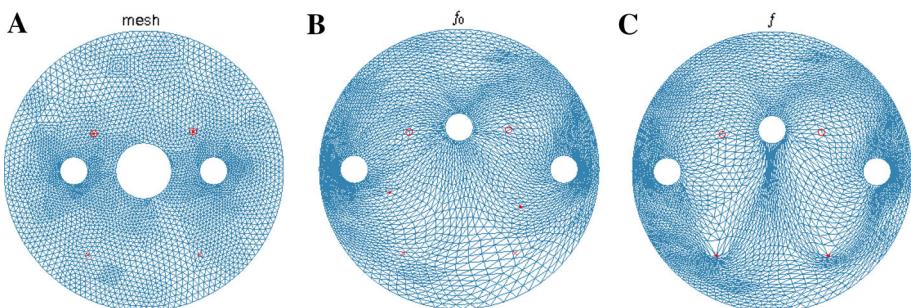


Fig. 22 Example 11: **a** shows the input mesh. **b** Shows the target domain and the initial map. The initial landmark point and target landmark point are labeled by open circle and multiplication sign respectively. **c** Shows the output mesh deformed by the optimal diffeomorphism. The initial landmark points (labeled by circle) are moved to \cdot , which are close to the target landmark points (labeled by \times)

Example 13 In this example, we test the proposed algorithm to solve the DOP on two multiply-connected real human faces for face registration. We compute the surface registration that matches both curvature and landmarks. Instead of parameterizing the surfaces

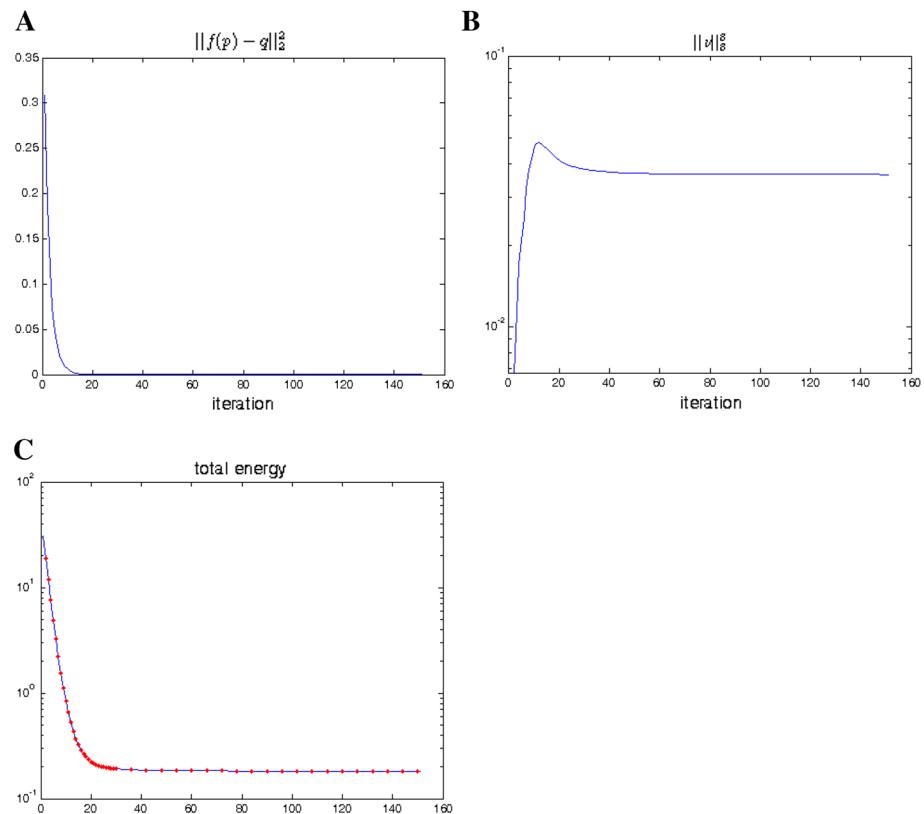


Fig. 23 Example 11: **a** shows the landmark mismatching error versus iterations. **b** Shows the conformality distortion versus iterations. **c** Shows the total energy versus iterations. The red dots indicate the actual ADMM iterations (Color figure online)

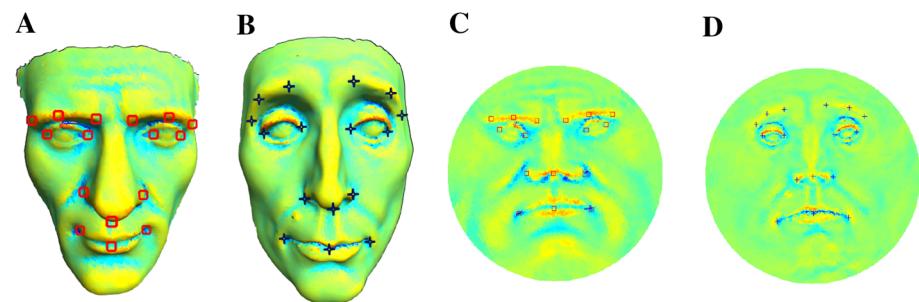


Fig. 24 Example 12: Solving DOP on Riemann surfaces. **a, b** show the meshes of two human faces. Corresponding feature landmarks on the human faces are shown. **c** Shows the conformal parameter domain of Face 1. **d** Shows the conformal parameter domain of Face 2. The colormaps are given by the mean curvatures of the human faces

onto a punctured disk with circular boundaries, we conformally parameterize the surfaces onto multiply-connected domains with boundaries of arbitrary shapes. Our method is able to solve the DOP on domains with arbitrary geometries. Figure 27a, b show two multiply-

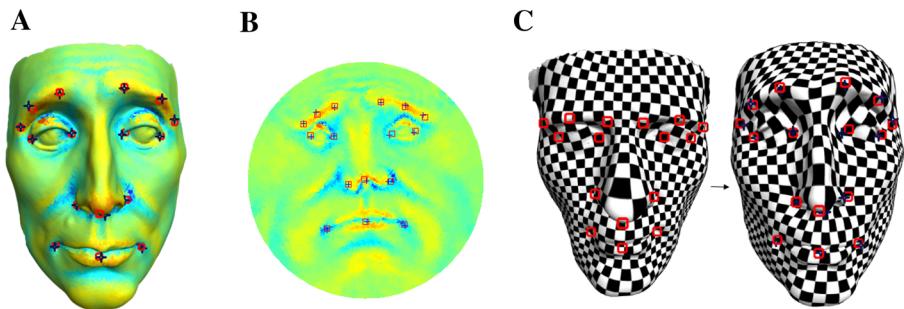


Fig. 25 Example 12: **a** shows the registration result. The colormap and landmarks on Face 1 are mapped to Face 2 using the obtained registration. Note that the high curvature regions are mapped to the corresponding high curvature regions. Landmarks are also consistently matched. **b** Shows the registration result on the conformal parameter domain

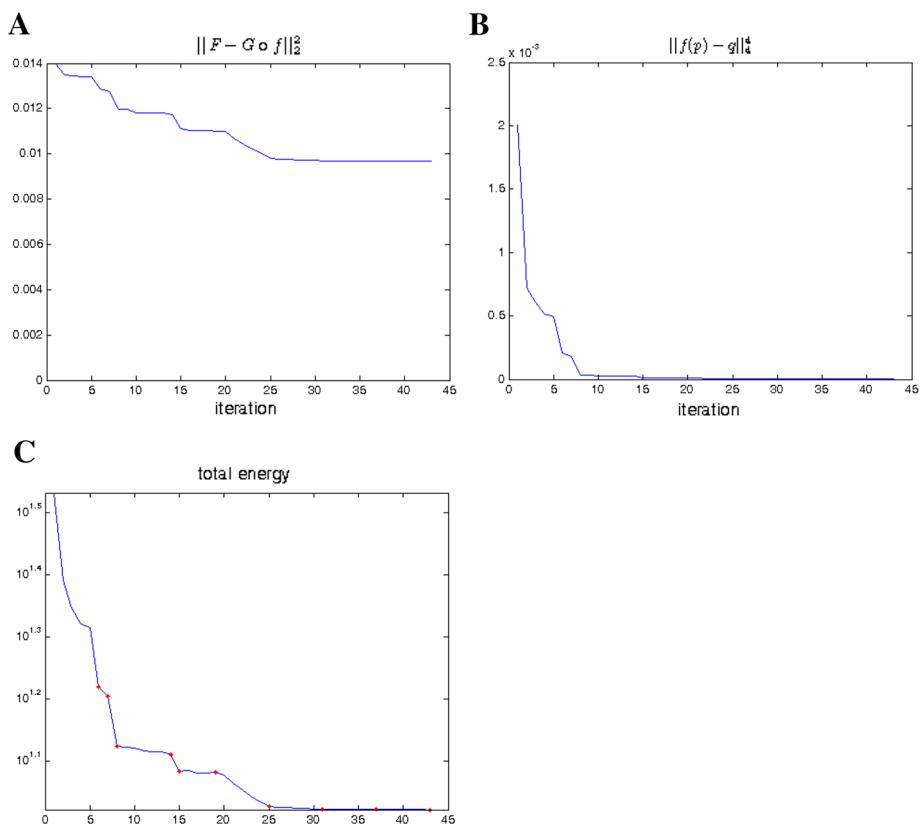


Fig. 26 Example 12: **a** shows the intensity mismatching error versus iterations. **b** Shows the landmark mismatching error versus iterations. **c** Shows the total energy versus iterations. The red dots indicate the actual ADMM iterations (Color figure online)

connected real human faces, with corresponding landmarks labeled on each of them. The colormaps on each surfaces are given by their mean curvatures. Their conformal parameter domains are shown in (c) and (d) respectively. We find a diffeomorphism that matches corre-

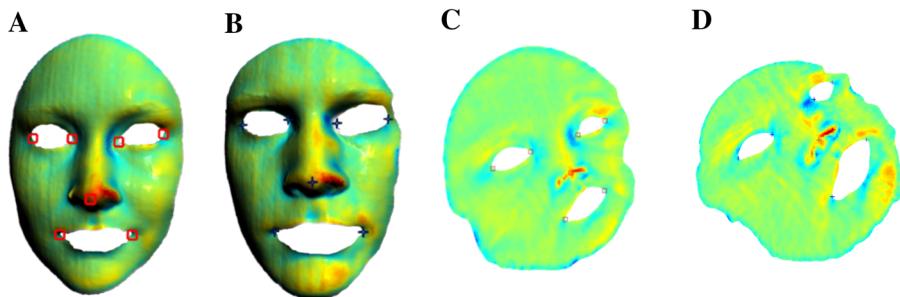


Fig. 27 Example 13: Solving DOP on multiply-connected real human faces. **a, b** Show the meshes of two multiply-connected human faces. Corresponding feature landmarks on the human faces are shown. **c** Shows the conformal parameter domain of Face 1. **d** Shows the conformal parameter domain of Face 2. The colormaps are given by the mean curvatures of the human faces

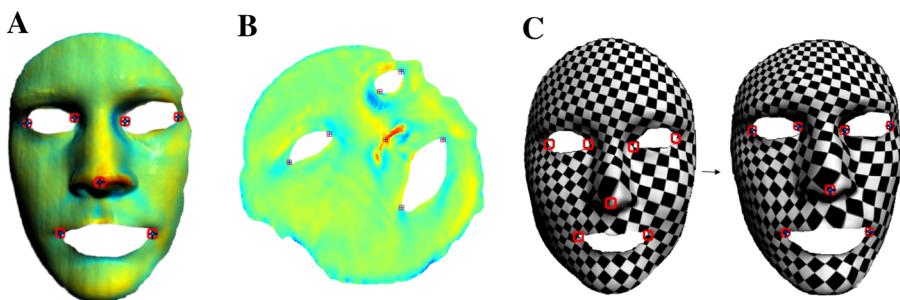


Fig. 28 Example 13: **a** shows the registration result. The colormap and landmarks on Face 1 are mapped to Face 2 using the obtained registration. Note that the high curvature regions are mapped to the corresponding high curvature regions. Landmarks are also consistently matched. **b** Shows the registration result on the conformal parameter domain

sponding landmarks and curvatures. The initial map is chosen to be the landmark matching Teichmüller map [38].

Figure 28 shows the registration result. The colormap (mean curvature) on Face 1 is mapped to Face 2 using the computed registration. Note that the high curvature regions (red colored regions) on Face 1 are mapped to corresponding high curvature regions on Face 2. Landmarks on Face 1 are also mapped to Face 2 using the obtained registration. Note that corresponding landmarks are matched consistently. (b) shows registration result on the conformal parameter domain. (c) shows the registration result visualized by the texture map. The energy plots of intensity mismatching error, conformality distortion and the total energy versus iterations are shown in Fig. 29a–c respectively. The red dots indicate the energy values at the actual ADMM iterations.

7 Conclusion

This paper introduces an efficient algorithm to solve the diffeomorphism optimization problem (DOP), using quasi-conformal theories. DOP is a type of optimization problems which minimizes energy functionals defined over the space of diffeomorphisms. In [1], the method of Beltrami holomorphic flow (BHF) was proposed to solve the DOP by representing the

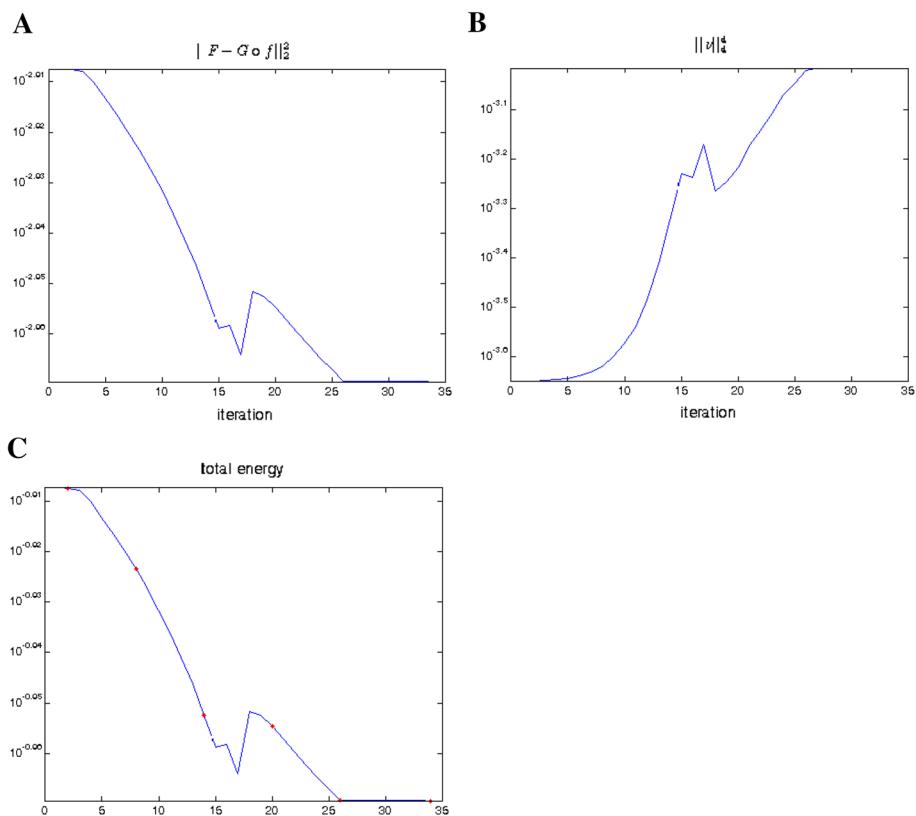


Fig. 29 Example 13: **a** shows the intensity mismatching error versus iterations. **b** Shows the conformality distortion versus iterations. **c** Shows the total energy versus iterations. The red dots indicate the actual ADMM iterations (Color figure online)

diffeomorphisms using the Beltrami coefficients. The optimal Beltrami coefficient associated to the diffeomorphism minimizing the energy functional can then be effectively found. However, the algorithm is computational expensive. In this work, we propose an efficient splitting algorithm, based on the classical alternating direction method of multiplier (ADMM), to solve the DOP. The basic idea is to split the energy functional into two energy terms: one involves the BC whereas the other involves the quasi-conformal map. Alternating minimization scheme can then be applied to minimize the energy functional. The proposed method significantly speeds up the previous BHF approach [1]. It also extends the previous BHF algorithm to Riemann surfaces of arbitrary topologies. Experiments have been carried out on synthetic together with real surface data, which demonstrate the efficiency and efficacy of the proposed algorithm to solve the DOP.

Acknowledgments Lok Ming Lui's research was supported in part by HKRGC GRF grant (CUHK project ID: 404612) and CUHK FIS grant 1902036.

References

1. Lui, L.M., Wong, T.W., Zeng, W., Gu, X.F., Thompson, P.M., Chan, T.F., Yau, S.T.: Optimization of surface registrations using Beltrami holomorphic flow. *J. Sci. Comput.* **50**(3), 557–585 (2012)

2. Angenent, S., Haker, S., Tannenbaum, A., Kikinis, R.: On the Laplace–Beltrami operator and brain surface flattening. *IEEE Trans. Med. Imaging* **18**(8), 700–711 (1999)
3. Durrleman, S., Pennec, X., Trouve, A., Thompson, P., Ayache, N.: Measuring brain variability via sulcal lines registration: a diffeomorphic approach. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI 2007) Lecture Notes in Computer Science 4791, pp. 675–682 Springer, Berlin, Heidelberg (2007)
4. Durrleman, S., Pennec, X., Trouve, A., Thompson, P., Ayache, N.: Inferring brain variability from diffeomorphic deformations of currents: an integrative approach. *Med. Image Anal.* **12**, 626–637 (2008)
5. Fischl, B., Sereno, M.I., Tootell, R.B., Dale, A.M.: High-resolution intersubject averaging and a coordinate system for the cortical surface. *Hum. Brain Mapp.* **8**, 272–284 (1999)
6. Gardiner, F.: Quasiconformal Teichmüller Theory. American Mathematical Society, Providence (2000)
7. Glauñès, J., Vaillant, M., Miller, M.I.: Landmark matching via large deformation diffeomorphisms on the sphere. *J. Math. Imaging Vis.* **20**, 179–200 (2004)
8. Gu, X., Wang, Y., Chan, T., Thompson, P., Yau, S.-T.: Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Trans. Med. Imaging* **23**(8), 949–958 (2004)
9. Haker, S., Angenent, S., Tannenbaum, A., Kikinis, R., Sapiro, G., Halle, M.: Conformal surface parameterization for texture mapping. *IEEE Trans. Vis. Comput. Graph.* **6**(8), 181189 (2000)
10. Hurdal, M., Stephenson, K.: Cortical cartography using the discrete conformal approach of circle packings. *NeuroImage* **23**, S119S128 (2004)
11. Hurdal, M.K., Stephenson, K.: Discrete conformal methods for cortical brain flattening. *Neuroimage* **45**, 86–98 (2009)
12. Joshi, S., Miller, M.: Landmark matching via large deformation diffeomorphisms. *IEEE Trans. Image Process.* **9**(8), 13571370 (2000)
13. Ju, L., Stern, J., Rehm, K., Schaper, K., Hurdal, M., Rottenberg, D.: Cortical surface flattening using least square conformal mapping with minimal metric distortion. In: IEEE International Symposium on Biomedical Imaging pp. 77–80 (2004)
14. Lehto, O., Virtanen, K.: Quasiconformal Conformal Mappings in the Plane. Springer, New York (1973)
15. Gardiner, F., Lakic, N.: Quasiconformal Teichmuller Theory. American Mathematical Society, Providence (2000)
16. Leow, A., Yu, C., Lee, S., Huang, S., Protas, H., Nicolson, R., Hayashi, K., Toga, A., Thompson, P.: Brain structural mapping using a novel hybrid implicit/explicit framework based on the level-set method. *NeuroImage* **24**(3), 910–927 (2005)
17. Lepore, N., Leow, P.T.A.D. : Landmark matching on the sphere using distance functions. In: IEEE International Symposium on Biomedical Imaging (ISBI2006), April 6–9 2006.
18. Lord, N.A., Ho, J., Vemuri, B., Eisenschenk, S.: Simultaneous registration and parcellation of bilateral hippocampal surface pairs for local asymmetry quantification. *IEEE Trans. Med. Imaging* **26**(4), 471–478 (2007)
19. Lui, L., Wang, Y., Chan, T., Thompson, P.: Brain anatomical feature detection by solving partial differential equations on general manifolds. *Discrete Contin. Dyn. Syst. B* **7**(3), 605–618 (2007)
20. Lui, L., Wang, Y., Chan, T., Thompson, P.: Landmark constrained genus zero surface conformal mapping and its application to brain mapping research. *Appl. Numer. Math.* **5**, 847–858 (2007)
21. Lui, L.M., Thiruvenkadam, S., Wang, Y., Chan, T., Thompson, P.: Optimized conformal parameterization of cortical surfaces using shape based matching of landmark curves. *SIAM J. Imaging Sci.* **3**(1), 52–78 (2010)
22. Lui, L.M., Wong, T.W., Gu, X.F., Thompson, P.M., Chan, T.F., Yau, S.T.: Hippocampal Shape Registration using Beltrami Holomorphic flow. In: Medical Image Computing and Computer Assisted Intervention (MICCAI). Part II, LNCS, vol. 6362, pp. 323–330 (2010)
23. Zeng, W., Lui, L.M., Luo, F., Chan, T.F., Yau, S.T., Gu, X.F.: Computing quasiconformal maps using an auxiliary metric and discrete curvature flow. *Numer. Math.* **121**(4), 671–703 (2012)
24. Lui, L.M., Lam, K.C., Wong, T.W., Gu, X.F.: Texture map and video compression using Beltrami representation. *SIAM J. Imaging Sci.* **6**(4), 1880–1902 (2013)
25. Lv, B., Petitjean, S., Ray, N., Maillot, J.: Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph. (TOG)* **21**(3), 362–371 (2002)
26. Zeng, W., Gu, X.: Registration for 3D surfaces with large deformations using quasi-conformal curvature flow. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR11), Jun 20–25, Colorado Springs, Colorado, USA (2011)
27. Lyttelton, O., Bouchera, M., Robbinsa, S., Evans, A.: An unbiased iterative group registration template for cortical surface analysis. *NeuroImage* **34**, 1535–1544 (2007)

28. Glowinski, R., Le Tallec, P.: Augmented Lagrangian and Operator-splitting Methods in Nonlinear Mechanics. Studies in Applied and Numerical Mathematics. Society for Industrial and Applied Mathematics (1989)
29. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **3**(1), 1–122 (2010)
30. Glowinski, R., Marrocco, A.: Sur l'approximation par éléments finis d'ordre un, et la résolution par penalisation-dualité d'une classe de problèmes de dirichlet non linéaires. *Rev. Française d'Aut. Inf. Rech. Oper.* **R-2**, 41–76 (1975)
31. Shi, Y., Thompson, P.M., Dinov, I., Osher, S., Toga, A.W.: Direct cortical mapping via solving partial differential equations on implicit surfaces. *Med. Image Anal.* **11**(3), 207–223 (2007)
32. Tosun, D., Rettmann, M., Prince, J.: Mapping techniques for aligning sulci across multiple brains. *Med. Image Anal.* **8**, 295309 (2004)
33. Wang, Y., Chiang, M.-C., Thompson, P.M.: Automated surface matching using mutual information applied to Riemann surface structures. *Proc. Med. Image Comput. Comput. Assist. Interv. MICCAI* **2005**(2), 666–674 (2005)
34. Jacobson, A., Sorkine, O.: A cotangent Laplacian for images as surfaces. *ACM Trans. Graph.* **25**(3), 646–653 (2012)
35. Weber, O., Myles, A., Zorin, D.: Computing extremal quasiconformal maps. *Comput. Graph. Forum* **31**(5), 16791689 (2012)
36. Wong, T.W., Zhao, H.: Computation of quasiconformal surface maps using discrete Beltrami flow. UCLA CAM report 12-85 (2013)
37. Lui, L.M., Lam, K.C., Yau, S.T., Gu, X.F.: Teichmüller extremal mapping and its applications to landmark matching registration. *SIAM J. Imaging Sci.* (2013). [arXiv:1211.2569](https://arxiv.org/abs/1211.2569) ([arXiv:1210.8025](https://arxiv.org/abs/1210.8025))
38. Ng, T.C., Gu, X.F., Lui, L.M.: Computing Teichmüller extremal map of multiply-connected domains using Beltrami holomorphic flow. *J. Sci. Comput.* (2013). doi:[10.1007/s10915-013-9791-z](https://doi.org/10.1007/s10915-013-9791-z)
39. Wang, Y., Lui, L., Gu, X., Hayashi, K., Chan, T., Toga, A., Thompson, P., Yau, S.: Brain surface conformal parameterization using riemann surface structure. *IEEE Trans. Med. Imaging* **26**(6), 853–865 (2007)
40. Wang, Y., Lui, L.M., Chan, T.F., Thompson, P.M.: Optimization of brain conformal mapping with landmarks. *Proc. Med. Image Comput. Comput. Assist. Interv. MICCAI* **2005**, 675–683 (2005)
41. Zeng, W., Luo, F., Yau, S.-T., Gu, X.: Surface quasi-conformal mapping by solving Beltrami equations. In: IMA Conference on the Mathematics of Surfaces XIII (2009)
42. Gu, X.F., Yau, S.T.: Computational Conformal Geometry. International Press, Boston (2007)