Preprint

A Splitting Method for Diffeomorphism Optimization Problem using Beltrami Coefficients

Tsz Ching Ng and Lok Ming Lui

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 [1]. 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 make 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

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 $f^* : S_1 \to S_2$ such that:

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

L.M. Lui is from the Department of Mathematics, The Chinese University of Hong Kong.

where $\mathfrak{Diff}$ is the collection of all orientation-preserving diffeomorphisms between S_1 and S_2 and $E : \mathfrak{Diff} \to \mathbb{R}^+$ 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 $\mathfrak{B} = \{\mu : S_1 \to \mathbb{C} : ||\mu||_\infty < 1\}$. It can be shown that there is a bijection $\Phi : \mathfrak{B} \to \mathfrak{Diff}$ between $\mathfrak{B}$ and $\mathfrak{Diff}$ [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 1-1 or onto. The only constraint is that its supreme norm has to be strictly less than 1. The original DOP (1) can be reformulated over the space of BC $\mathfrak{B}$ as follows:

$$\nu^* = \mathbf{argmin}_{\mu \in \mathfrak{B}} E_B(\mu) \tag{2}$$

subject to the constraint that (i) $\nu^* = \mu(f^*) =$ BC of some $f^* \in \mathfrak{Diff}$ and (ii) $||\nu^*||_\infty < 1$. 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 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 on 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:

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

with the constraint that $\nu^* = \mu(f^*) =$ BC of some $f^* \in \mathfrak{Diff}$, where f^μ is the quasi-conformal map associated to μ. The first term minimize 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 $i = 1, 2, ..., n$.

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 augmented Lagrangian method of 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][9][10][11][15][27][50]. Conformal maps are usually computed using variational approaches to minimize some energy functionals, such as the harmonic energy [8] and the least-squares energy based on the Cauchy-Riemann equation [27]. 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. [22][51] 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. [43] 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][14] 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. [23] 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. [18] 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. [19] used a level-set representation to match curves embedded in surfaces, using a diffeomorphic flow parametrized using 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. [40] 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. [28] also 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. [29] 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. [44] computed surface registrations that maximize the mutual information between mean curvature and conformal factor maps across subjects. Lord et al. [20] matched surfaces by minimizing the deviation from isometry. Quasi-conformal surface registrations, which minimize geometric mismatching, have also been studied [1,24–26]. For example, Lui et al. [24] proposed to compute quasi-conformal registration between hippocampal surfaces, which matches geometric quantities (such as curvatures) and minimizes the conformality distortion [24]. 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 [25,26,47].

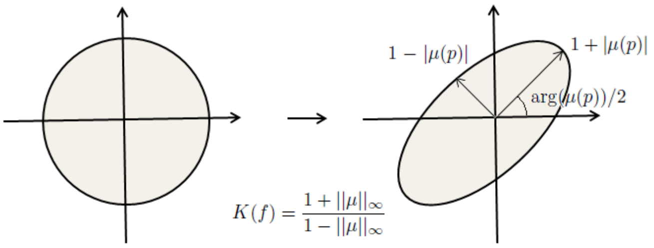


Fig. 1 Illustration of how the Beltrami coefficient determines the conformality distortion.

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 [16,17].

A surface S with a conformal structure is called a *Riemann surface*. Given two Riemann surfaces M and N, a map $f : M \to N$ 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 [17]. Mathematically, $f : \mathbb{C} \to \mathbb{C}$ is quasi-conformal provided that it satisfies the Beltrami equation:

$$\frac{\partial f}{\partial \overline{z}} = \mu(z)\frac{\partial f}{\partial z}. \tag{4}$$

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:

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

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 $S(z) = z + \mu(p)\overline{z}$, 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 Figure 1).

The maximal dilation of f is given by:

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

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 $\mu(z)\frac{\overline{dz}}{dz}$ 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

$$\mu_\alpha(z_\alpha)\frac{d\overline{z_\alpha}}{dz_\alpha} = \mu_\beta(z_\beta)\frac{d\overline{z_\beta}}{dz_\beta}, \tag{7}$$

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 $f : S_1 \to S_2$ 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 $f_{\alpha\beta} := \psi_\beta \circ f \circ f_\alpha^{-1}$ is quasi-conformal associated with $\mu_\alpha(z_\alpha)\frac{d\overline{z_\alpha}}{dz_\alpha}$.

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 $\mu := \mu(t, z) : \mathbb{C} \times S_1 \to \mathbb{C}$ such that $\mu(t, \cdot) \in \mathfrak{B}$ 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 $f(t, \cdot) : S_1 \to S_2$. $f(t, \cdot)$ is called the *Beltrami holomorphic flow (BHF)* of quasi-conformal maps associated with $\mu(t, \cdot)$. It can be shown that for any fixed $p \in S_1$, $f(\cdot, p) : \mathbb{C} \to \mathbb{C}$ 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:

$$\mu(t_0, z) = \mu(z) + (t - t_0)\nu(z) + (t - t_0)\epsilon(t - t_0)(z) \tag{8}$$

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

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

local uniformally as $t \to t_0$, where

$$\mathbf{V}(f^\mu, \nu)(p) = -\frac{f^\mu(f^\mu(p) - 1)}{\pi} \int_{\mathbb{C}} \frac{\nu(z)((f^\mu)_z(z))^2}{f^\mu(z)(f^\mu(z) - 1)(f^\mu(z) - f^\mu(p))} dxdy \tag{10}$$

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

$$\begin{aligned}\mathbf{V}(f^\mu, \nu)(p) = -\frac{f^\mu(f^\mu(p) - 1)}{\pi} \Big(&\int_{\mathbb{D}} \frac{\nu(z)((f^\mu)_z(z))^2}{f^\mu(z)(f^\mu(z) - 1)(f^\mu(z) - f^\mu(p))} dxdy \\ &+ \int_{\mathbb{D}} \frac{\overline{\nu(z)}(\overline{(f^\mu)_z(z)})^2}{\overline{f^\mu(z)}(1 - \overline{f^\mu(z)})(1 - \overline{f^\mu(z)}f^\mu(p))} dxdy \Big).\end{aligned} \tag{11}$$

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 Equation (1). 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 (1) 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. [26][48] 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. [52] proposed to compute the quasi-conformal map using the holomorphic differential 1-form under the auxiliary metric given by the BC. Zeng et al. [25] 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. [47] proposed to approximate the small perturbation of the quasi-conformal map from the identity map under a small variation $t\nu$ 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. [49] proposed to iteratively compute the quasi-conformal map f^ν with Beltrami coefficient ν by solving the Beltrami's equation in each iteration. The Beltrami's equation is solved by a L^p-minimization of the Beltrami energy:

$$\widetilde{f} = \mathbf{argmin}_{f \in \mathfrak{Diff}}\{||\frac{\partial f}{\partial \bar{z}} - \nu\frac{\partial f}{\partial z}||_p^p\} \tag{12}$$

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* ν, which has also been studied in [46][53]. In this paper, we will apply the method in [49] to cope with a subproblem in the alternating minimization algorithm for solving the DOP.

3.3 Alternating direction method with multipliers (ADMM)

Alternating direction method with multipliers (ADMM) is an optimization algorithm that solves the following type of problems:

$$\textbf{Minimize } \{E_1(x) + E_2(Ax)\} \tag{13}$$

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:

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

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

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

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

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

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

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:

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

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 [37–39].

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 $f^* : S_1 \to S_2$ that solves:

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

where $\mathfrak{Diff}$ is the collection of all surface diffeomorphisms between S_1 and S_2 and $E : \mathfrak{Diff} \to \mathbb{R}$ 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 $\mu(f) : S_1 \to \mathbb{C}$. 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 function in order to control the geometric distortion of the mapping. In other words, our optimization problem can generally be written as finding $f^* : S_1 \to S_2$ that solves:

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

subject to $||\mu(f^*)||_\infty := ||\frac{\partial f^*}{\partial \bar{z}} / \frac{\partial f^*}{\partial z}||_\infty < 1$, where $E_1 : \mathfrak{Diff} \to \mathbb{R}^+$ and $E_2 : \mathfrak{B} \to \mathbb{R}^+$ 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 $f^* : S_1 \to S_2$ and $\nu^* : S_1 \to \mathbb{C}$ such that:

$$(f^*, \nu^*) = \mathbf{argmin}_{f \in \mathfrak{Diff}, \mu \in \mathfrak{B}} \{E_1(f) + E_2(\mu)\} \tag{21}$$

subject to (i) $\nu^* = \mu(f^*)$ and (ii) $||\nu^*||_\infty < 1$, where $E_1 : \mathfrak{Diff} \to \mathbb{R}^+$ and $E_2 : \mathfrak{B} \to \mathbb{R}^+$ are energy functionals defined on $\mathfrak{Diff}$ and $\mathfrak{B}$ respectively.

Condition (i) guarantees that ν^* 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 ν^* and $f^* : S_1 \to S_2$ 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 ν^*. ν^* is given by:

$$\nu^* = \frac{\partial f^*}{\partial \bar{z}} / \frac{\partial f^*}{\partial z} \tag{22}$$

where

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

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

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

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

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

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 strategry is to apply an ADMM-like alternating optimization scheme. We consider the augmented Lagrangian of the problem (21), which is given by:

$$L(f, \nu, \lambda_{\mathbf{Re}}, \lambda_{\mathbf{Im}}, \rho) = E_1(f) + E_2(\nu) + < \lambda_{\mathbf{Re}}, \mathbf{Re}(\nu - \mu(f)) > + < \lambda_{\mathbf{Im}}, \mathbf{Im}(\nu - \mu(f)) > + \frac{\rho}{2} ||\nu - \mu(f)||_2^2 \tag{25}$$

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

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

$$f^{k+1} = \mathbf{argmin}_f \{L(f, \nu_k, \lambda^k_{\mathbf{Re}}, \lambda^k_{\mathbf{Im}}, \rho^k)\} \tag{26}$$

$$\nu^{k+1} = \mathbf{argmin}_\nu \{L(f_{k+1}, \nu, \lambda^k_{\mathbf{Re}}, \lambda^k_{\mathbf{Im}}, \rho^k)\} \tag{27}$$

$\lambda^k_{\mathbf{Re}}, \lambda^k_{\mathbf{Im}}$ and ρ^k are updated as follows.
If $||\nu_{k+1} - \mu(f_{k+1})||_2 < \eta_k$, update:

$$\begin{aligned} \lambda^{k+1}_{\mathbf{Re}} &= \lambda^k_{\mathbf{Re}} + \rho_k \mathbf{Re}(\nu_{k+1} - \mu(f_{k+1});) \\ \lambda^{k+1}_{\mathbf{Im}} &= \lambda^k_{\mathbf{Im}} + \rho_k \mathbf{Im}(\nu_{k+1} - \mu(f_{k+1})); \\ \rho_{k+1} &= \rho_k. \end{aligned} \tag{28}$$

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

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

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

4.1 Minimization of subproblem (27) involving ν

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:

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

In the discrete case, it becomes a sparse linear sytem 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:

$$\mu^{new} = \mu^{old} + dt d\mu, \text{ where} \tag{31}$$

$$d\mu = \Delta\mu - p(\mathbf{Re}(\mu))^{p-1} - ip(\mathbf{Im}(\mu))^{p-1} - \lambda_{\mathbf{Re}}^{k}\mathbf{Re}(\nu - \mu(f_{k+1})) - i\lambda_{\mathbf{Im}}^{k}\mathbf{Im}(\nu - \mu(f_{k+1})) - \rho_k(\nu - \mu(f_{k+1})) \tag{32}$$

The stopping criteria is chosen to be $\nabla_\nu L < \epsilon_k$

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:

$$f_{k+1} = \mathbf{argmin}_f\{E_1(f) + < \lambda_{\mathbf{Re}}, \mathbf{Re}(\nu - \mu(f)) > + < \lambda_{\mathbf{Im}}, \mathbf{Im}(\nu - \mu(f)) > + \frac{\rho}{2}||\nu - \mu(f)||_2^2\}. \tag{33}$$

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 $E_1(f)$. f is then updated by: $\overline{f} = f_k + \mathbf{V}_1$. 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: $\mathbf{V}_1 = 2(I_1 - I_2(f))\nabla f$.

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

$$\partial\widetilde{\nu} = -\lambda_{\mathbf{Re}} - i\lambda_{\mathbf{Im}} + \rho(\nu - \mu(f)). \tag{34}$$

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

$$\frac{\partial\widetilde{f}}{\partial\overline{z}} = \widetilde{\nu}\frac{\partial\widetilde{f}}{\partial z} \tag{35}$$

subject to the boundary constraints that $\widetilde{f}|_{\partial S_1} = \partial S_2$. 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 [26,48,25,49,47,52]. In this paper, we approximate the quasi-conformal map $\widetilde{f}$ with a Beltrami coefficient $\widetilde{\nu}$ by directy solving the Beltrami's equation (35) as in [49]. The details of the computation of the quasi-conformal map will be explained in Section 4.3

Once $\widetilde{f}$ is obtained, we get another descent direction $\mathbf{V}_2$ 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):

$$\frac{df^k}{dt} = \mathbf{V}_1 + \mathbf{V}_2 \tag{36}$$

We set the stopping criteria for the gradient descent algorithm as $\nabla_f L < \epsilon_k$ 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. More specifically, given a map f^μ with Beltrami coefficient μ, we need to deform f^μ to a new f^ν of Beltrami coefficient 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 $f^\mu : \Omega_1 \to \Omega_2$ be an initial quasi-conformal map, whose Beltrami coefficient is $\mu : \Omega_1 \to \mathbb{C}$. Assume μ changes to ν, and assume its associated quasiconformal map is denoted by f^ν. As in [49], 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 Beltrami coefficients 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:

$$\begin{tikzcd}
\mu_0 := \mu \arrow[r] \arrow[d, leftrightarrow] & \mu_1 \arrow[r] \arrow[d, leftrightarrow] & \ldots \arrow[r] & \mu_n \arrow[r] \arrow[d, leftrightarrow] & \ldots \arrow[r] & \mu_\infty = \nu \arrow[d, leftrightarrow] \\
f^{\mu_0} := f^\mu \arrow[r] & f^{\mu_1} \arrow[r] & \ldots \arrow[r] & f^{\mu_n} \arrow[r] & \ldots \arrow[r] & f^{\mu_\infty} = f^\nu
\end{tikzcd} \tag{37}$$

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 Beltrami coefficient μ_1 is close to $\nu_1 := (1-\epsilon)\mu_0 + \epsilon\nu$ $(\epsilon > 0)$. To find f_1, we need to solve:

$$f^{\mu_1} = \mathbf{argmin}_f \{ || \frac{\partial f}{\partial \bar{z}} / \frac{\partial f}{\partial z} - \nu_1 ||_\infty \} \tag{38}$$

For a small variation $\nu_1 - \mu_0$, the above problem can be solved by a L^p-minimization of the following Beltrmai energy (for sufficiently large p):

$$f^{\mu_1} = \mathbf{argmin}_{f \in \mathfrak{Diff}} \{ || \frac{\partial f}{\partial \bar{z}} - \nu_1 \frac{\partial f}{\partial z} ||_p^p \} \tag{39}$$

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

$$g_1 = \mathbf{argmin}_{g:\Omega_1 \to \mathbb{R}^2} \{ || \frac{\partial (f^{\mu_0} + g)}{\partial \bar{z}} - \nu_1 \frac{\partial (f^{\mu_0} + g)}{\partial z} ||_p^p \}. \tag{40}$$

subject to the boundary constraints.

Equivalently, we find g_1 that solves

$$g_1 = \mathbf{argmin}_{g:\Omega_1 \to \mathbb{R}^2} \{ || \mathcal{A}(\nu_1) g + \mathcal{A}(\nu_1) f^{\mu_0} ||_p^p \}, \tag{41}$$

subject to the boundary constraints, where $\mathcal{A}(\nu_1) := \frac{\partial}{\partial \bar{z}} - \nu_1 \frac{\partial}{\partial z}$.

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 Beltrami coefficient is denoted by μ_1.

Suppose at the n^{th} iteration, we have the quasi-conformal map f^{μ_n} with Beltrami coefficient μ_n. We then flow f^{μ_n} to $f^{\mu_{n+1}}$ whose Beltrami coefficient μ_{n+1} is close to $\nu_n := (1-\epsilon)\mu_n + \epsilon\nu$ $(\epsilon > 0)$. 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:

$$g_{n+1} = \mathbf{argmin}_{g:\Omega_1 \to \mathbb{R}^2} \{ || \mathcal{A}(\nu_{n+1}) g + \mathcal{A}(\nu_{n+1}) f^{\mu_0} ||_p^p \}, \tag{42}$$

subject to the boundary constraints, where $\mathcal{A}(\nu_{n+1}) := \frac{\partial}{\partial \bar{z}} - \nu_{n+1} \frac{\partial}{\partial z}$. $f^{\mu_{n+1}}$ is then obtained, whose Beltrami coefficient is denoted by μ_{n+1}. Note that in each step, ϵ can be chosen so that $||\mu_{n+1} - \nu||_\infty$ 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 Beltrami coefficients converge to ν. We call such a process to deform f^μ to f^ν iteratively the *Beltrami holomorphic flow (BHF)* from μ to ν, and denote it by: $\mathbf{BHF}(\mu \to \nu)$.

The Beltrami holomorphic flow can be summarized as follows:

Algorithm 1 : *(Beltrami holomorphic flow)*
Input : $f^\mu : \Omega_1 \to \Omega_2$ *with Beltrami coefficient* μ*, target Beltrami coefficient* ν*, 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* $\nu_n := (1-\epsilon)\mu_n + \epsilon\nu$*; 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* $E(f,\nu) = E_1(f) + E_2(\nu)$
Output : *Optimal quasi-conformal map* f^* *with Beltrami coefficient* ν^*

1. *Set* $\nu_0 = 0$ *and* $f_0 =$ *harmonic map between* Ω_1 *and* Ω_2*;*
2. *Given* f_n, ν_n, λ_n *and* ρ_n, *compute* f_{n+1} *by:*
$$f_{n+1} = \mathbf{argmin}_f L(f, \nu_n, \lambda_n, \rho_n);$$
3. *Compute* ν_{n+1} *by:*
$$\nu_{n+1} = \mathbf{argmin}_\nu L(f_{n+1}, \nu, \lambda_n, \rho_n);$$
4. *Compute* λ_{n+1} *and* ρ_{n+1}*;*
5. *If* $|E(f_{n+1}, \nu_{n+1}) - E(f_n, \nu_n)| \geq \epsilon$, *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* $E(f,\nu) = E_1(f) + E_2(\nu)$
Output : *Optimal quasi-conformal map* f^* *with Beltrami coefficient* ν^*

1. *Set* $\nu_0 = 0$ *and* $f_0 =$ *harmonic map between* Ω_1 *and* Ω_2*;*
2. *Given* f_n, ν_n, λ_n *and* ρ_n, *put* $\widetilde{f} = f_n$ *and* $\widetilde{\nu} = \nu_n$. *Compute the gradient descent directions* $\mathbf{V_1}$ *and* $\mathbf{V_2}$ *to minimize* $L(f, \nu_n, \lambda_n, \rho_n)$*; Update* $\widetilde{f} := \widetilde{f} + (\mathbf{V_1} + \mathbf{V_2})dt$*;*
3. *Compute the gradient descent direction* $\partial\nu$ *that minimizes* $L(f_{n+1}, \nu, \lambda_n, \rho_n)$*; Update* $\widetilde{\nu} := \widetilde{\nu} + (\partial\nu)dt$*;*
4. If $|\nabla_f L + \nabla_\nu L| < \epsilon_k$, put $f_{n+1} = \widetilde{f}$ and $\nu_{n+1} = \widetilde{\nu}$; Go to step 5. Otherwise, put $f_n = \widetilde{f}$ and $\nu_n = \widetilde{\nu}$. Go to step 2.
5. *Compute* λ_{n+1} *and* ρ_{n+1}*;*
6. *If* $|E(f_{n+1}, \nu_{n+1}) - E(f_n, \nu_n)| \geq \epsilon$, *go to step 2. Otherwise, stop the iteration.*

5 Numerical implementation details

In this section, we will explain in detail the numerical implementation of the algorithms proposed in section 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 equation (42). Let $f = (u + \sqrt{-1}v) : K_1 \to K_2$. 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:

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

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:

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

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,

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

where A is the area of T and

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

where $\mathbf{n}$ is the unit normal of T. Let $\nu(T)$ 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:

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

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:

$$\begin{aligned} \sum_{T_j} | \frac{1}{4\text{Area}(T_j)} &(1 - \nu(T_j), \sqrt{-1} + \sqrt{-1} \nu(T_j)) \sum_{i=1}^{3} (P_{T_j(i)} + \sqrt{-1} Q_{T_j(i)}) \mathbf{s}_i(T_j) \\ + \frac{1}{4\text{Area}(T_j)} &(1 - \nu(T_j), \sqrt{-1} + \sqrt{-1} \nu(T_j)) \sum_{i=1}^{3} (u_{T_j(i)} + \sqrt{-1} v_{T_j(i)}) \mathbf{s}_i(T_j) |^p, \end{aligned} \tag{48}$$

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

Secondly, the boundary constraint can be approximated by a linear constraint, so that the least square method can be applied to solve the problem (48). 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

$$b_i P_i - a_i Q_i = 0, \tag{49}$$

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 $||\nu_n - \mu_n||_\infty$ 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.

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

5.2 Implementation details of Algorithm 2 and Algorithm 3

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

In the discrete setting, ν_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 real-valued function defined on each triangular faces. 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

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

where ∇_T is given by (45).

The augmented Lagrangian 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 $L(f_n,\nu_n,\lambda_n,\rho_n)$, where $\mathbf{V_1}$ is the descent direction for $E_1(f)$ and $\mathbf{V_2}$ is the descent direction for $<\lambda_{\mathbf{Re}},\mathbf{Re}(\nu-\mu(f))>+<\lambda_{\mathbf{Im}},\mathbf{Im}(\nu-\mu(f))>+\frac{\rho}{2}||\nu-\mu(f)||_2^2$.

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 form 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. 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

$$\partial\mu(T)=\lambda_n(T)+\rho_n(\nu_n(T)-\mu(f_n)(T)) \tag{52}$$

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 number. Then we set $\mathbf{V_2}=(f'-f_n)/d\tau$.

After computing $\mathbf{V_1}$ and $\mathbf{V_2}$, we set $\tilde{f}=f_n+(\mathbf{V_1}+\mathbf{V_2})dt$ and project the boundary points to the corresponding boundaries. For Algorithm 2, we simply set $f_{n+1}=\tilde{f}$. For Algorithm 3, 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 $L(f_{n+1},\nu,\lambda_n,\rho_n)$ with respect to ν:

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

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

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

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:

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

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} andβ_{ij}, we follow the idea of [45]. 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

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

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

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

Therefore we have

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

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

In the situation when $E_2(\nu)$ is not easy to minimize, we can use example the gradient descent to compute the minimizer. As an example, consider $E_2(\nu)=||\nabla\nu||_2^2+||\nu||_4^4$. The descent direction for $\nu_n(T)$ is given by

$$\partial\nu=-\mathcal{I}_v^f\Delta\mathcal{I}_f^v\nu_n-p|\nu_n|^{p-2}\nu_n-\lambda^n_{\mathbf{Re}}\mathbf{Re}(\nu-\mu(f_{n+1}))-i\lambda^n_{\mathbf{Im}}\mathbf{Im}(\nu-\mu(f_{n+1}))-\rho_n(\nu-\mu(f_{n+1})). \tag{58}$$

On each face T, we set $\tilde{\nu}(T) = \nu_n(T) + \partial\nu(T)d\tau$. After that, we repeat the procedure as above but replacing ν_n by $\tilde{\nu}$. When the computation reach convergence, we set $\nu_{n+1} = \tilde{\nu}$.

Finally, in Step 4 of both algorithms, λ_{n+1} and ρ_{n+1} are updated as follows. If the gradient $||\nabla_f L + \nabla_\nu L|| > \delta$ for some predefined parameter δ, then we set $\lambda_{n+1} = \lambda_n$ and $\rho_{n+1} = \rho_n$. Otherwise we will check the magnitude of the residue $||\nu_{n+1} - \mu(f_{n+1})||_2$. If $||\nu_{n+1} - \mu(f_{n+1})||_2 < \eta$ for some parameter η, then we set

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

and $\rho_{n+1} = \rho_n$. Otherwise, we will set $\lambda_{n+1} = \lambda_n$ and set $\rho_{n+1} = \rho_n^{1+\delta}$, where δ 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 $\mathbf{D} = \{(x,y) \in \mathbb{R}^2 : (x-0.5)^2 + (y-0.5)^2 = 1\}$ matching the intensity functions.

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

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

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

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

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

Using the proposed splitting algorithm, we solve the above DOP with stopping criteria as $||f_{k+1} - f_k||_1 < 10^{-4}$ and $||\mu_{k+1} - \mu_k||_1 < 10^{-5}$. Figure 2(A) shows the input mesh of $\mathbf{D}$. (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 Figure 3(A) (which is visualized as an output mesh obtained by deforming the input mesh using the optimal diffeomorphism). Figure 3(B) shows the intensity function $G \circ f(x,y)$, which closely resemble to $F(x,y)$. It means the registration matches the intensity functions well. Figure 4(A), (B) and (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 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 disortion increases as iterations increase, since the initial map is the identity map with no conformality distortion. Conformality distortion is iteratively induces in order to tolerate for more intensity matching.

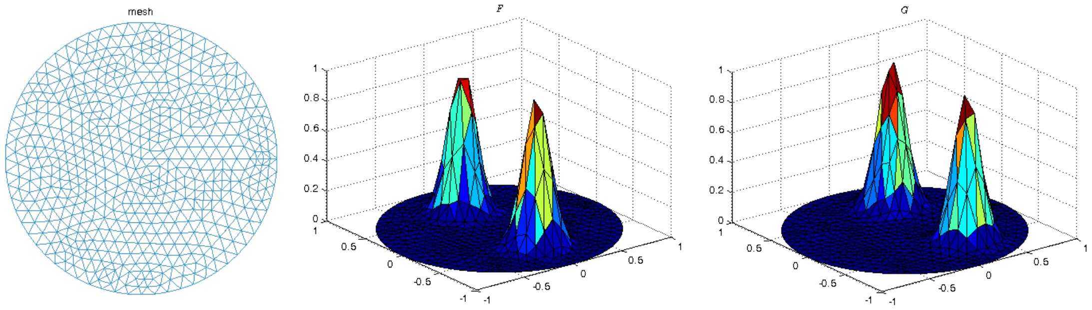
We also compare our proposed method with the algorithm in [1]. The result is shown in Figure 5. The optimal diffeomorphism is shown in Figure 5(A), which looks similar to the one we obtain (Figure 2(A)). However, the intensity mismatching error at the optimal state is higher than that of our proposed 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 of $\mathbf{D}$. Figure 6(A) 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:

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

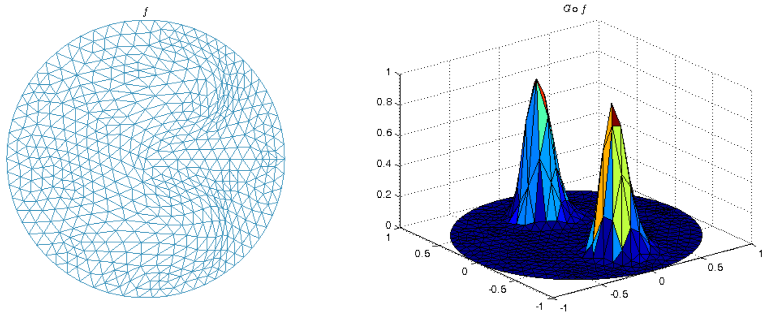
(with $\alpha = 100$ and $\beta = 10$).

Using the splitting method, we solve the above DOP and obtain the diffeomorphism f^*, which is shown in Figure 6(B). Landmark point is matched consistently. We also optimize the DOP using the algorithm in [1]. The obtained diffeomorphism is shown in Figure 6(C). Note that the landmark point cannot be matched consistently. Figure 7(A) and (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



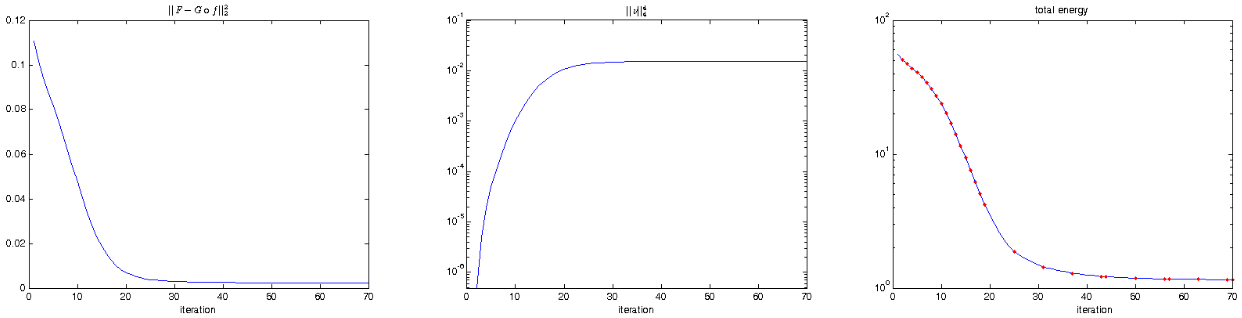
(A) Input mesh **(B) $F(x,y)$** **(C) $G(x,y)$**

Fig. 2 Example 1: (A) shows the input mesh. (B) shows the intensity $F(x, y)$ defined on $\mathbf{D}$. (C) shows the intensity $G(x, y)$ defined on $\mathbf{D}$.



(A) Output mesh **(B) $G(f)$**

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 $\mathbf{D}$, which closely resembles to $F(x, y)$.

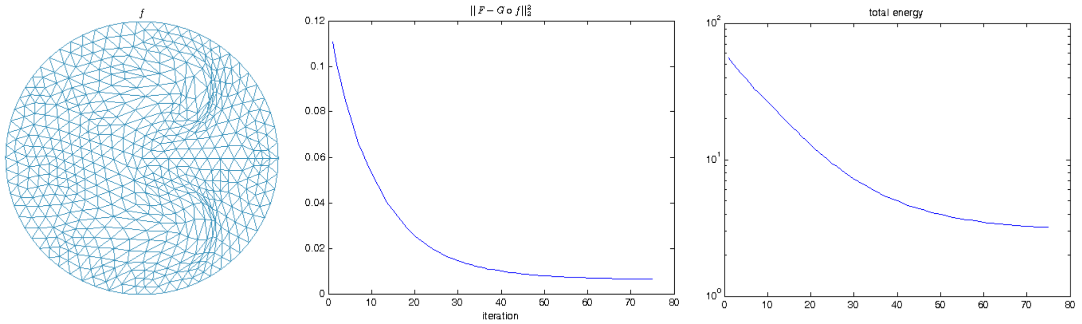


(A) Intensity mismatching **(B) Conformality distortion** **(C) Total energy**

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.

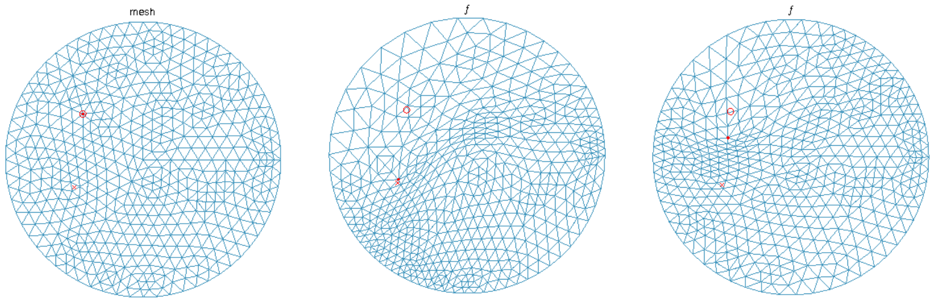
the total energy versus iterations. The red dots indicate the actual ADMM iterations. Figure 7(C), (D) and (E) show the landmark mismatching error, conformality distortion and total energy versus iterations using the integral method proposed in [1]. The landmark mismatching error converges at about 0.06. The algorithm get stuck at a local minimum with higher total energy value than our proposed splitting method.

Example 3: We have also compared our proposed splitting method to solve the DOP with the method proposed in [1] on more examples. Table 6.1 records the computational times to solve the DOP using the two different methods on 5 different examples. Two experiments to compute intensity matching diffeomorphisms (similar to Example 1) and three experiments to compute landmark matching diffeomorphisms (similar to Example 2) have been carried out. 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.



(A) Output mesh **(B) Intensity mismatching** **(C) Total energy**

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.



(A) Input mesh **(B) Output mesh (Splitting method)** **(C) Output mesh (Integral method)**

Fig. 6 Example 2: (A) shows the input mesh, with the initial landmark point and target landmark point labeled by ∘ and × 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 ∘ is moved to · 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].

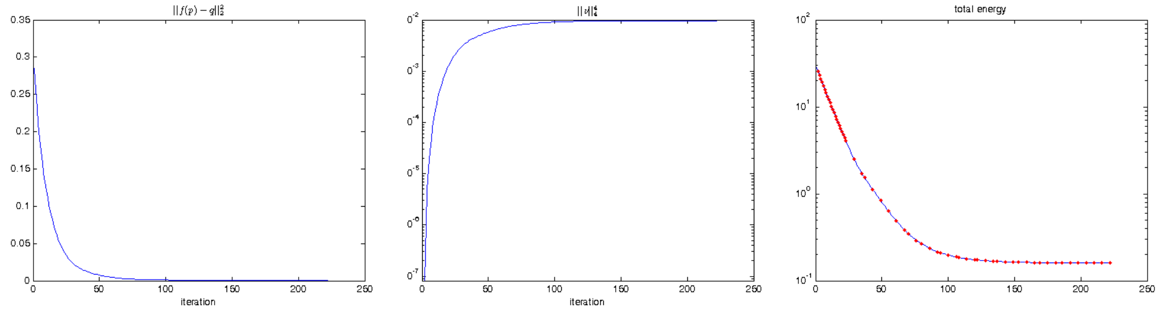
Examples	Splitting method		Integral method [1]	
	Time Taken	Final Energy	Time Taken	Final Energy
Intensity matching 1	$19s$	1.46×10^{-3}	$1680s$	4.03×10^{-3}
Intensity matching 2	$23s$	8.88×10^{-3}	$1650s$	2.27×10^{-2}
Landmark matching 1	$116s$	2.03×10^{-4}	$1731s$	1.50×10^{-2}
Landmark matching 2	$78s$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

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

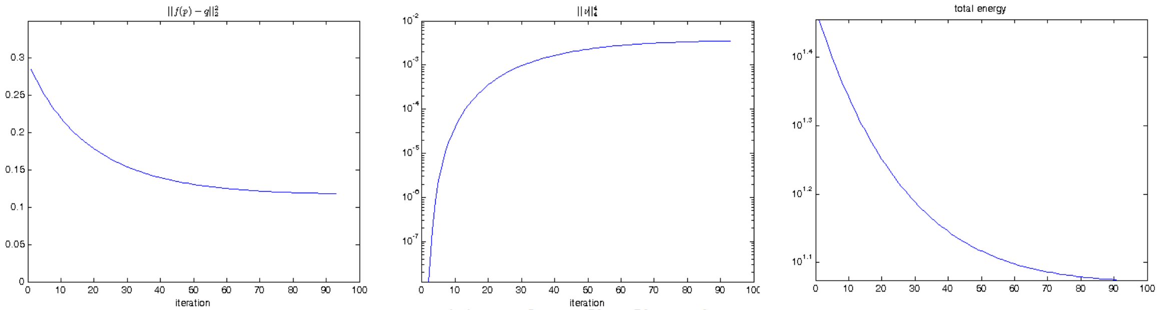
Example 4: We test our algorithm on a rectangular domain $R = [-0.5, 1.5] \times [-0.5, 1.5]$. We look for an optimal diffeomorphism $f^* : R \to R$ that matches two intensity functions $F : R \to \mathbb{R}$ and $G : R \to \mathbb{R}$ defined by:

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

We minimize: $E(f) = 500||F - G(f)||_2^2 + 5||\mu(f)||_4^4$. Figure 8(A) 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 Figure 9(A). (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.

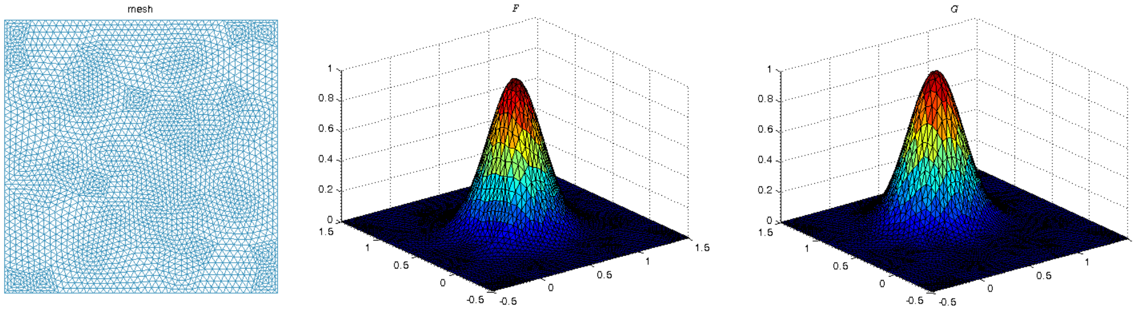


(A) Landmark mismatching (Splitting method) **(B) Conformality distortion (Splitting method)** **(C) Total energy (Splitting method)**



(D) Landmark mismatching (Integral method) **(E) Conformality distortion (Integral method)** **(F) Total energy (Integral method)**

Fig. 7 Example 2: (A), (B) and (C) show the landmark mismatching error, conformality distortion and total energy vesus iterations respectively using the splitting method. (D), (E) and (F) show the landmark mismatching error, conformality distortion and total energy vesus iterations respectively using the integral method in [1].



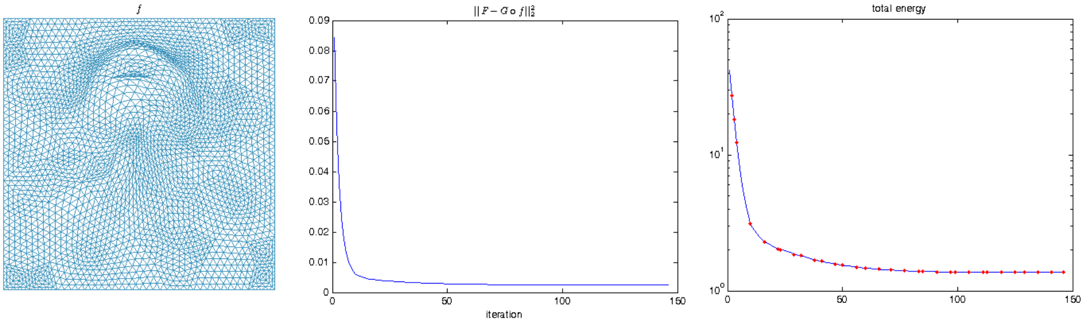
(A) Input mesh **(B) *F(x,y)*** **(C) *G(x,y)***

Fig. 8 Example 3: (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.

Example 5: In this example, we solve the DOP to find a quasi-conformal map $f : \mathbb{D} \to \mathbb{D}$ of the unit disk $\mathbb{D}$ with BC $= \nu$ that minimizes: $E(f) = ||\mu(f) - \nu||_4^4$. We set $\nu = 0.3$. Figure 10(A) 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.

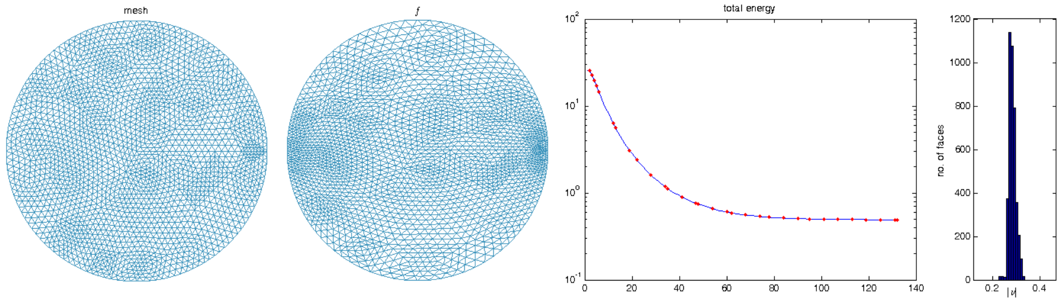
6.2 Solving DOPs on multiply-connected domains

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 $f : \Omega \to \Omega$ with BC $= \nu := 0.3$ that minimizes: $E(f) = ||\mu(f) - \nu||_4^4$. Figure 11(A) 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



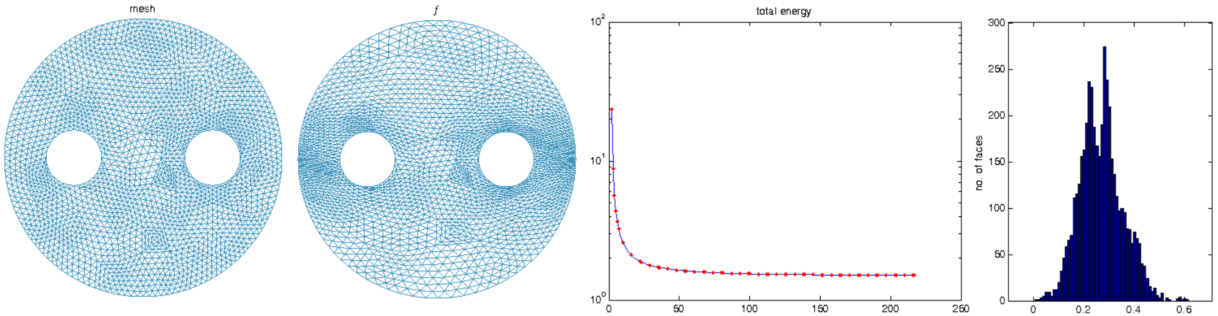
(A) Output mesh **(B) Intensity mismatching** **(C) Total energy**

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.



(A) Input mesh **(B) Output mesh** **(C) Total energy** **(D) Histogram of BC norm**

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.



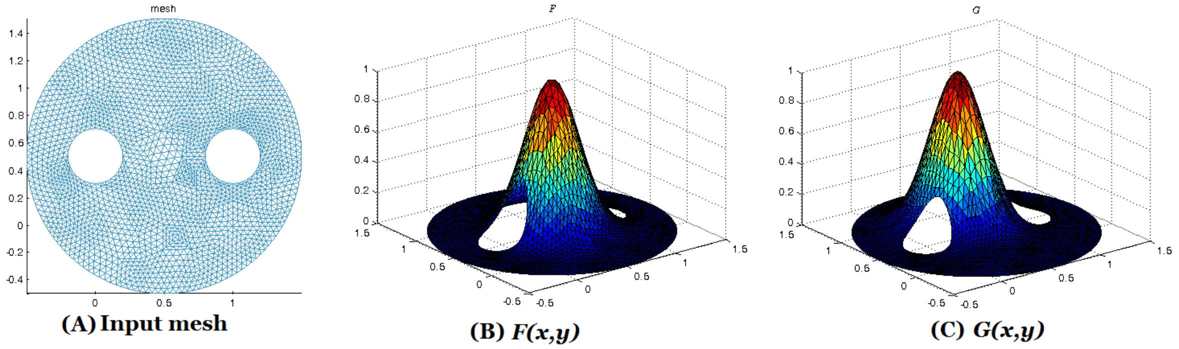
(A) Input mesh **(B) Output mesh** **(C) Total energy** **(D) Histogram of BC norm**

Fig. 11 Example 6: (A) shows the input mesh of a triply-connected domain. (B) shows the output mesh deformed from the inital 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.

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 ν may not exist. By minimizing E, one can find a diffeomorphism whose BC is closest to the given BC ν 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 12(A) shows an input mesh of a triply-connected domain Ω. Define $F : \Omega \to \mathbb{R}$ and $G : \Omega \to \mathbb{R}$ by:

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



(A) Input mesh (B) F(x,y) (C) G(x,y)

Fig. 12 Example 7: (A) shows the input mesh. (B) shows the intensity $F(x,y)$. (C) shows the intensity $G(x,y)$.

which is shown in Figure 12(B) and (C).

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

$$E(f)=\alpha\sum_{i=1}^{2}|f(p_i)-q_i|^2+\gamma_1||\mu(f)||_4^4+\gamma_2||\nabla\mu(f)||_2^2 \tag{65}$$

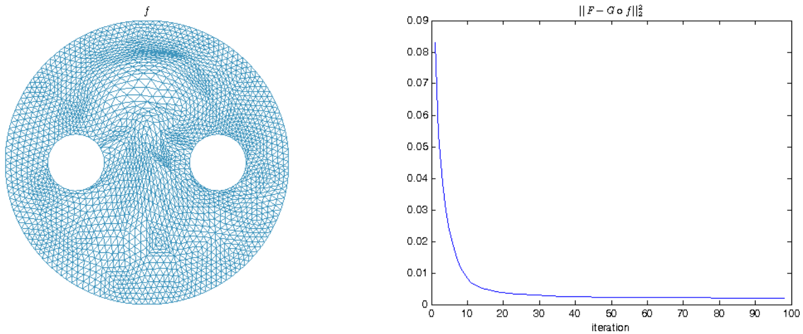
(where $\alpha=10^4$, $\gamma_1=10$ and $\gamma_2=0.5\times10^{-6}$). Using the splitting method, we obtain the optimal diffeomorphism f^* as shown in Figure 13(A). Figure 13(B), (C) and (D) shows the intensity mismatching error, conformality distortion and total energy versus iterations. Next, we take $\alpha=50$ and $\beta=5$. In other words, we like to obtain more conformality. Figure 14(A) shows the obtained diffeomorphism. Note that the diffeomorphism has less squeezing than that in Figure 12(A). Figure 14(B) and (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, more intensity mismatching error at the optimal state is higher than the previous results. Note that the algorithm in [1] cannot apply to multiply-connected domains. Thus, this problem cannot be solved by the method in [1].

Example 8: In this example, we test the algorithm to compute the landmark matching diffeomorphism of a triply-connected domain Ω. Figure 15(A) shows an input mesh of Ω. We look for a diffeomorphism $f^*:\Omega\to\Omega$ 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 Figure 15(B), which matches landmark consistently. Figure 16(B) and (C) shows the landmark mismatching error and the conformality distortion versus iterations respectively.

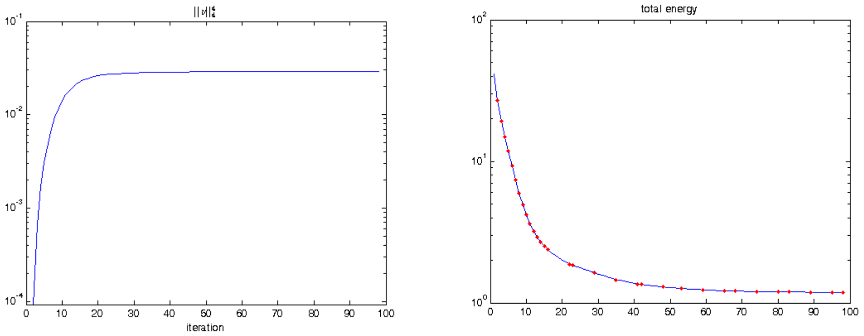
Example 9: In this example, we test the algorithm to compute a diffeomorphism $f^*:\Omega\to\Omega$ that matches both intensity functions and landmark constraints on a triply-connected domain Ω. Figure 17(A) shows the input mesh of Ω. The initial landmarks $\{p_1,p_2\}$ are labeled by $\circ$ and the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

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

(where $\alpha=10^2$, $\beta=10^2$ and $\gamma=10^{-2}$). The obtained diffeomorphism is shown in Figure 18(A). 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 19(A), (B), (C) and (D) show the conformality distortion, intensity mismatching error, landmark mismatching error and total energy versus iterations respectively. The landmark mismatching error converges to 0 at the optimal state.

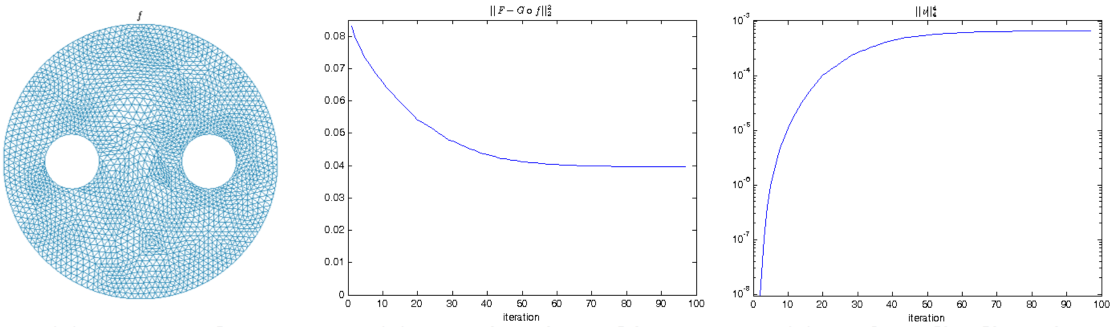


(A) Output mesh **(B) Intensity mismatching**



(C) Conformality distortion **(D) Total energy**

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.



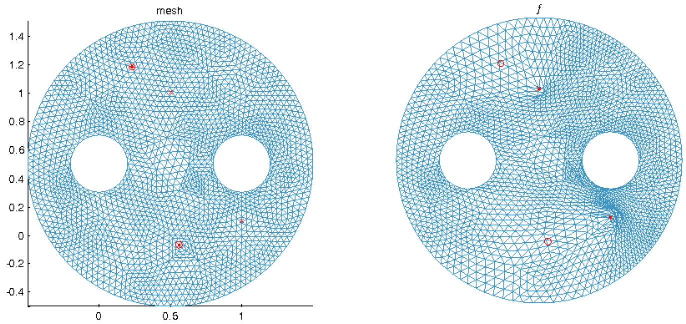
(A) Output mesh **(B) Intensity mismatching** **(C) Conformality distortion**

Fig. 14 Results of Example 7 with larger parameter for the conformality distortion energy (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 total energy versus iterations. The red dots indicate the actual ADMM iterations.

6.3 Solving DOPs on Riemann surfaces

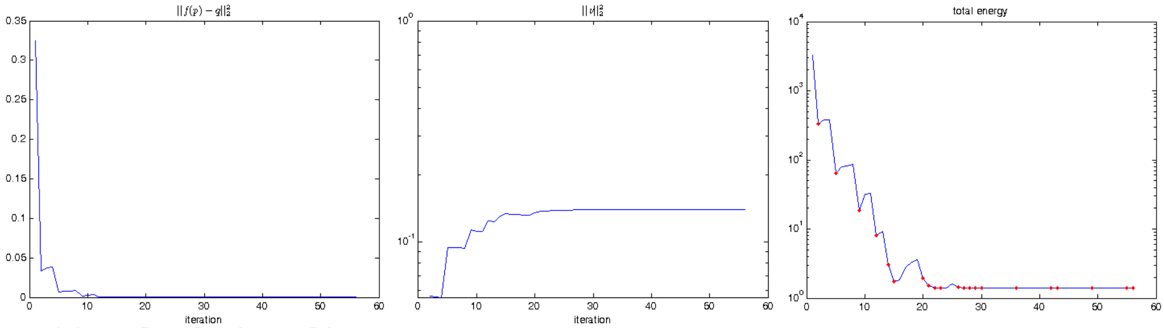
Example 10: Our proposed method can be easily applied to solving DOPs on Riemann surfaces through conformal parameterization. Every connected Riemann surfaces can be parameterized conformally onto the unit disk $\mathbb{D}$ or multiply-connected punctured disk. Solving the DOP on Riemann surfaces is 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 faces that matches both landmarks and curvatures. Figure 20(A) and (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:

$$E(f) = \alpha \sum_{i=1}^{2} |f(p_i) - q_i|^2 + \beta ||H_2(f) - H_1||_2^2 + \gamma_1 ||\mu(f)||_4^4 \tag{67}$$



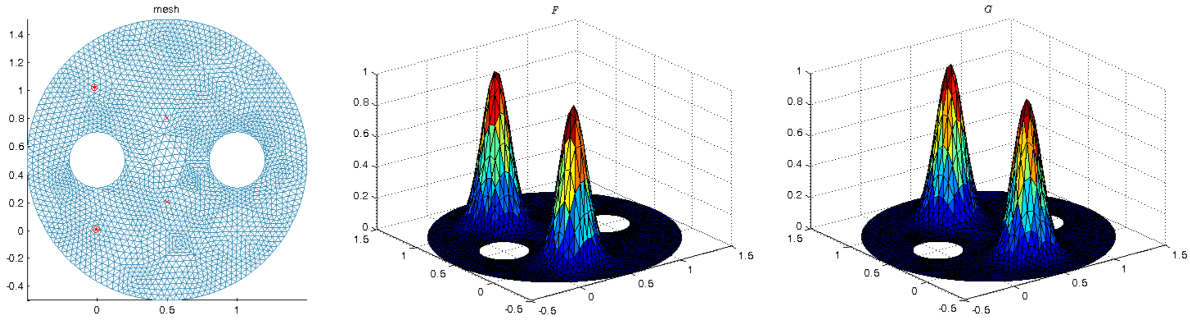
(A) Input mesh **(B) Output mesh**

Fig. 15 Example 8: (A) shows the input mesh. The initial landmark point and target landmark point are labeled by ○ and × respectively. (B) shows the output mesh deformed by the optimal diffeomorphism. The initial landmark points (labeled by ○) are moved to ·, which are close to the target landmark points (labeled by ·).



(A) Landmark mismatching **(B) Conformality distortion** **(C) Total energy**

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.

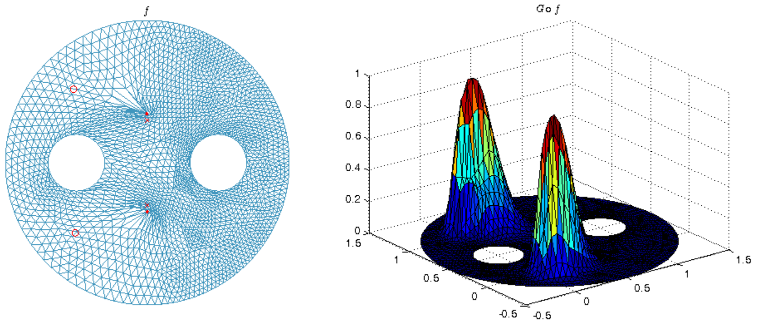


(A) Input mesh **(B) *F(x,y)*** **(C) *G(x,y)***

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 ○ and × respectively. (B) shows the intensity $F(x,y)$. (C) shows the intensity $G(x,y)$.

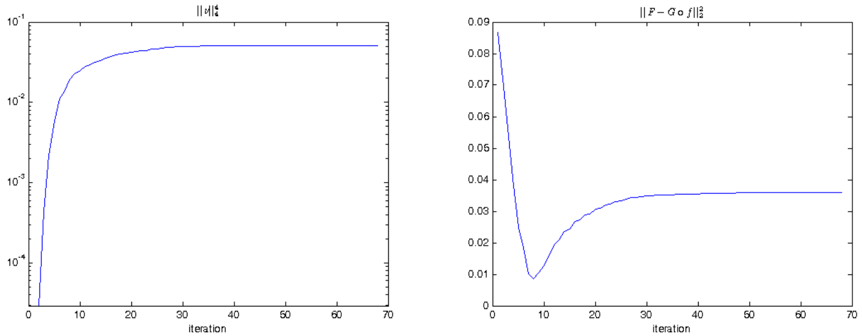
where $\alpha = 10^4$, $\beta = 10^2$, $\gamma_1 = 10^2$, $\gamma_2 = 10^2$ and H_1 and H_2 are the mean curvature functions on Face 1 and Face 2 respectively.

Figure 21(A) 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 quite consistently. (B) shows registration result on the conformal parameter domain. The energy plots of intenisity mismatching error, landmark mismatching error and the total energy versus iterations are shown in Figure 22(A), (B) and (C) respectively. The red dots indicate the energy values at the actual ADMM iterations.

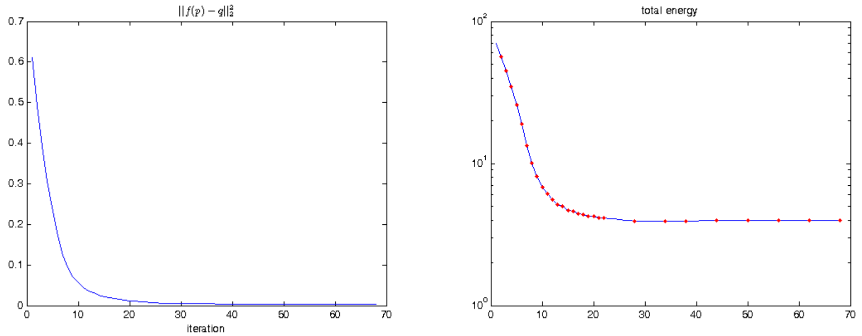


(A) Output mesh **(B) G(f)**

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.



(A) Conformality distortion **(B) Intensity mismatching**



(C) Landmark mismatching **(D) Total energy**

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.

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 an 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 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. It also extends the previous BHF algorithm to Riemann surfaces of arbitrary topologies.

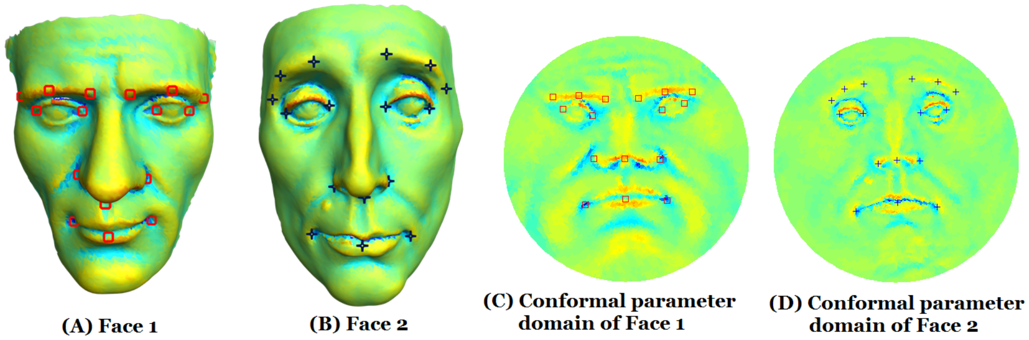

(A) Face 1 **(B) Face 2** **(C) Conformal parameter domain of Face 1** **(D) Conformal parameter domain of Face 2**

Fig. 20 Example 10: Solving DOP on Riemann surfaces. (A) and (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.

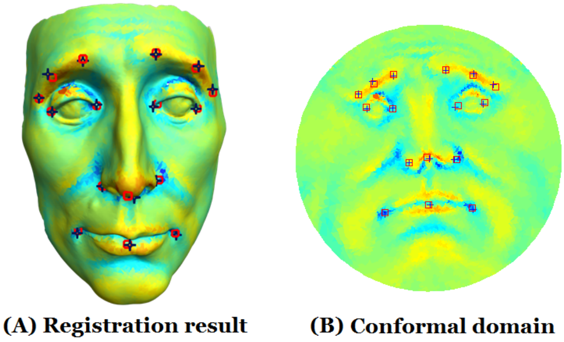
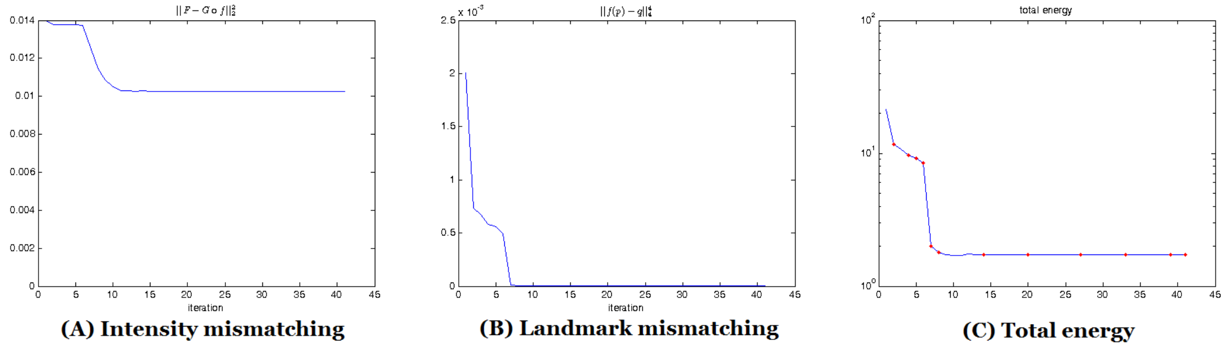

(A) Registration result **(B) Conformal domain**

Fig. 21 Example 10: (A) shows the registration result. The colormap and landmarks on Face 1 is 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 confomral parameter domain.



(A) Intensity mismatching **(B) Landmark mismatching** **(C) Total energy**

Fig. 22 Example 10: (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.

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.

References

1. L.M. Lui, T.W. Wong, W. Zeng, X.F. Gu, P.M. Thompson, T.F. Chan and S.T. Yau *Optimization of Surface Registrations using Beltrami Holomorphic Flow.* Journal of Scientific Computing, 50(3), 557-585 (2012)
2. S. Angenent, S. Haker, A. Tannenbaum, and R. Kikinis. On the Laplace-Beltrami operator and brain surface flattening. *IEEE Transaction of Medical Imaging*, 18(8):700–711, 1999.
3. S. Durrleman, X. Pennec, A. Trouve, P. Thompson, and N. Ayache. Measuring brain variability via sulcal lines registration: A diffeomorphic approach. *Medical Image Computing and Computer-Assisted Intervention (MICCAI 2007) Lecture Notes in Comput. Sci. 4791, Springer-Verlag, Berlin, Heidelberg*, page 675682, 2007.

4. S. Durrleman, X. Pennec, A. Trouve, P. Thompson, and N. Ayache. Inferring brain variability from diffeomorphic deformations of currents: An integrative approach. *Medical Image Analysis*, 12:626637, 2008.
5. B. Fischl, M. I. Sereno, R. B. Tootell, and A. M. Dale. High-resolution intersubject averaging and a coordinate system for the cortical surface. *Human Brain Mapping*, 8:272–284, 1999.
6. F. Gardiner. *Quasiconformal Teichmüller Theory.* American Mathematics Society, 2000.
7. J. Glaunès, M. Vaillant, and M. I. Miller. Landmark matching via large deformation diffeomorphisms on the sphere. *Journal of Mathematical Imaging and Vision*, 20:179–200, 2004.
8. X. Gu, Y. Wang, T. Chan, P. Thompson, and S.-T. Yau. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Transactions on Medical Imaging*, 23(8):949–958, 2004.
9. S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, G. Sapiro, and M. Halle. Conformal surface parameterization for texture mapping. *IEEE Transaction of Vision and Computer Graphics*, 6(8):181189, 2000.
10. M. Hurdal and K. Stephenson. Cortical cartography using the discrete conformal approach of circle packings. *NeuroImage*, 23:S119S128, 2004.
11. M. K. Hurdal and K. Stephenson. Discrete conformal methods for cortical brain flattening. *Neuroimage*, 45:86–98, 2009.
12. Z. T. J.H. Morra, L. Apostolova, A. Green, A. Toga, and P. Thompson. Comparison of adaboost and support vector machines for detecting alzheimer's disease through automated hippocampal segmentation. *IEEE Transactions on Medical Imaging*, 29(1):30–43, 2010.
13. M. Jin, J. Kim, F. Luo, and X. Gu. Discrete surface Ricci flow. *IEEE Transactions on Visualization and Computer Graphics*, 14(5):1030–1043, 2008.
14. S. Joshi and M. Miller. Landmark matching via large deformation diffeomorphisms. *IEEE Transactions on Image Processing*, 9(8):13571370, 2000.
15. L. Ju, J. Stern, K. Rehm, K. Schaper, M. Hurdal, and D. Rottenberg. Cortical surface flattening using least square conformal mapping with minimal metric distortion. *IEEE International Symposium on Biomedical Imaging*, pages 77–80, 2004.
16. O. Lehto and K. Virtanen. *Quasiconformal Conformal Mappings in the Plane.* Springer-Verlag New York, 1973.
17. F. Gardiner and N. Lakic. *Quasiconformal Teichmuller Theory.* American Mathematics Society, 2000.
18. A. Leow, C. Yu, S. Lee, S. Huang, H. Protas, R. Nicolson, K. Hayashi, A. Toga, and P. Thompson. Brain structural mapping using a novel hybrid implicit/explicit framework based on the level-set method. *NeuroImage*, 24(3):910–927, 2005.
19. N. Lepore and P. T. A.D. Leow. Landmark matching on the sphere using distance functions. *IEEE International Symposium on Biomedical Imaging (ISBI2006), April 6-9 2006.*
20. N. A. Lord, J. Ho, B. Vemuri, and S. Eisenschenk. Simultaneous registration and parcellation of bilateral hippocampal surface pairs for local asymmetry quantification. *IEEE Transactions on Medical Imaging*, 26(4):471–478, 2007.
21. L. Lui, Y. Wang, T. Chan, and P. Thompson. Brain anatomical feature detection by solving partial differential equations on general manifolds. *Discrete and Continuous Dynamical Systems B*, 7(3):605–618, 2007.
22. L. Lui, Y. Wang, T. Chan, and P. Thompson. Landmark constrained genus zero surface conformal mapping and its application to brain mapping research. *Applied Numerical Mathematics*, 5:847–858, 2007.
23. L. M. Lui, S. Thiruvenkadam, Y. Wang, T. Chan, and P. Thompson. Optimized conformal parameterization of cortical surfaces using shape based matching of landmark curves. *SIAM Journal of Imaging Science*, 3(1):52–78, 2010.
24. L.M. Lui, T.W. Wong, X.F. Gu, P.M. Thompson, T.F. Chan and S.T. Yau. *Hippocampal Shape Registration using Beltrami Holomorphic flow*, Medical Image Computing and Computer Assisted Intervention(MICCAI), Part II, LNCS 6362, 323-330 (2010)
25. W. Zeng, L.M. Lui, F. Luo, T.F. Chan, S.T. Yau, X.F. Gu *Computing quasiconformal maps using an auxiliary metric and discrete curvature flow.* Numerische Mathematik, 121(4), 671-703, 2012
26. L.M. Lui, K.C. Lam, T.W. Wong, X.F. Gu *Texture Map and Video Compression Using Beltrami Representation. SIAM Journal on Imaging Science*, 6(4), 1880-1902, 2013
27. B. Lvy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics (TOG)*, 21(3):362 – 371, 2002.
28. W. Zeng and X. Gu. *Registration for 3D Surfaces with Large Deformations Using Quasi-Conformal Curvature Flow.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR11), Jun 20-25, 2011, Colorado Springs, Colorado, USA.
29. O. Lyttelton, M. Bouchera, S. Robbinsa, and A. Evans. An unbiased iterative group registration template for cortical surface analysis. *NeuroImage*, 34:1535–1544, 2007.
30. J. Morra and et al. Automated mapping of hippocampal atrophy in 1-year repeat mri data in 490 subjects with alzheimer's disease, mild cognitive impairment, and elderly controls. *Neuroimage*, 45(1):S3–15, 2009.
31. J. Morra, Z. Tu, L. Apostolova, A. Green, C. Avedissian, S. Madsen, N. Parikshak, A. Toga, C. Jack, N. Schuff, M. Weiner, and P. Thompson. Automated 3d mapping of hippocampal atrophy and its clinical correlates in 400 subjects with alzheimer's disease, mild cognitive impairment and elderly controls. *NeuroImage, Special Issue on Mathematics in Brain Imaging, published online, PMID: 19041724*, 11 2008.
32. J. Morra, Z. Tu, L. Apostolova, A. Green, C. Avedissian, S. Madsen, N. Parikshak, A. Toga, C. Jack, N. Schuff, M. Weiner, and P. Thompson. Automated mapping of hippocampal atrophy in 1-year repeat mri data in 490 subjects with alzheimer's disease, mild cognitive impairment and elderly controls. *NeuroImage, Special Issue on Mathematics in Brain Imaging, published online, PMID: 19041724*, 11 2008.
33. J. Morra, Z. Tu, L. Apostolova, A. Green, C. Avedissian, S. Madsen, N. Parikshak, A. Toga, C. Jack, N. Schuff, M. Weiner, and P. Thompson. Validation of a fully automated 3d hippocampal segmentation method using subjects with alzheimer's disease, mild cognitive impairment and elderly controls. *NeuroImage*, 43(1):59–68, 10 2008.
34. J. Morra, Z. Tu, L. Apostolova, A. Green, C. Avedissian, S. Madsen, N. Parikshak, A. Toga, C. Jack, N. Schuff, M. Weiner, and P. Thompson. Automated mapping of hippocampal atrophy in 1-year repeat mri data in 490 subjects with alzheimer's disease, mild cognitive impairment and elderly controls. *Human Brain Mapping*, 30(9):2766–88, 9 2009.

35. J. Ogren, A. Bragin, C. Wilson, G. Hoftman, J. Lin, R. Dutton, T. Fields, A. Toga, P. Thompson, J. Engel, and R. Staba. 3D hippocampal atrophy maps distinguish two common temporal lobe onset seizure patterns. *Epilepsia*, 50(6):1361–1370, 6 2009.
36. J. Ogren, C. Wilson, A. Bragin, J. Lin, N. Salamon, R. Dutton, E. Luders, T. Fields, I. Fried, A. Toga, P. Thompson, J. Engel, and R. Staba. 3d surface maps link local atrophy and fast ripples in human epileptic hippocampus. *Ann Neurol*, 66(6):783–91, 2009.
37. R. Glowinski and P. Le Tallee, Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics SIAM, 1989.
38. S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning, 3(1), 2010
39. R. Glowinski and A. Marrocco, Sur lapproximation par elements Lnis dordre un, et la resolution par penalisation-dualite dune classe de problemes de dirichlet nonlineaires. Rev. Francaise dAut. Inf. Rech. Oper., R-2:41-76, 1975.
40. Y. Shi, P. M. Thompson, I. Dinov, S. Osher, and A. W. Toga. Direct cortical mapping via solving partial differential equations on implicit surfaces. *Medical Image Analysis*, 11(3):207–223, 2007.
41. P. Thompson, K. Hayashi, E. Sowell, N. Gogtay, J. Giedd, J. Rapoport, G. de Zubicaray, A. Janke, S. Rose, J. Semple, D. Doddrell, Y. Wang, T. C. T.G.M. van Erp, and A. Toga. Mapping cortical change in alzheimer's disease, brain development, and schizophrenia. *NeuroImage, Special Issue on Mathematics in Brain Imaging, Suppl 1:S2-18*, 23, 9 2004.
42. P. M. Thompson, K. M. Hayashi, G. I. de Zubicaray, A. L. Janke, S. E. Rose, J. Semple, M. S. Hong, D. H. Herman, D. Gravano, D. M. Doddrell, and A. W. Toga. Mapping hippocampal and ventricular change in alzheimer's disease. *NeuroImage*, 22(4):1754–66, 2004.
43. D. Tosun, M. Rettmann, and J. Prince. Mapping techniques for aligning sulci across multiple brains. *Medical Image Analysis*, 8:295309, 2004.
44. Y. Wang, M.-C. Chiang, and P. M. Thompson. Automated surface matching using mutual information applied to Riemann surface structures. *Proceeding in Medical Image Computing and Computer-Assisted Intervention MICCAI 2005*, 2:666–674, 2005.
45. A. Jacobson, O. Sorkine, newblock*A Cotangent Laplacian for Images as Surfaces* ACM Trans. Graph, 25(3), 646–653, 2012
46. O. Weber, A. Myles, D. Zorin. *Computing Extremal Quasiconformal Maps.* Computer Graphics Forum, 31(5), 16791689 (2012)
47. Tsz Wai Wong, Hongkai Zhao *Computation of quasiconformal surface maps using discrete Beltrami flow.* UCLA CAM report 12-85, 2013
48. L.M. Lui, K.C. Lam, S.T. Yau, X.F. Gu *Teichmüller extremal mapping and its applications to landmark matching registration.* arXiv:1211.2569(http://arxiv.org/abs/1210.8025), accepted, SIAM Journal on Imaging
49. T.C. Ng, X.F. Gu, L.M. Lui *Computing Teichmüller extremal map of multiply-connected domains using Beltrami holomorphic flow.* DOI 10.1007/s10915-013-9791-z, online first, Journal of Scientic Computing, 2013
50. Y. Wang, L. Lui, X. Gu, K. Hayashi, T. Chan, A. Toga, P. Thompson, and S. Yau. Brain surface conformal parameterization using riemann surface structure. *IEEE Transactions on Medical Imaging*, 26(6):853–865, 2007.
51. Y. Wang, L. M. Lui, T. F. Chan, and P. M. Thompson. Optimization of brain conformal mapping with landmarks. *Proceeding in Medical Image Computing and Computer-Assisted Intervention MICCAI 2005*, pages 675–683, 2005.
52. W. Zeng, F. Luo, S.-T. Yau and X. Gu, Surface Quasi-Conformal Mapping by Solving Beltrami Equations. *IMA Conference on the Mathematics of Surfaces XIII*, 2009.
53. X.F. Gu and S.T. Yau, Computational Conformal Geometry. *International Press*, 2007.