COMPUTATION OF QUASICONFORMAL SURFACE MAPS USING DISCRETE BELTRAMI FLOW

Abstract. The manipulation of surface homeomorphisms is an important aspect in 3D modeling and surface processing. Every homeomorphic surface map can be considered as a quasiconformal map, with its local non-conformal distortion given by its Beltrami differential. As a generalization of conformal maps, quasiconformal maps are of great interest in mathematical study and real applications. Efficient and accurate computational construction of desirable quasiconformal maps between general surfaces is crucial. However, in the literature we have reviewed, all existing computational works on construction of quasiconformal maps to or from a compact domain require global parametrization onto the plane, and have difficulty to be directly applied to maps between arbitrary surfaces. This work fills up the gap by proposing to compute quasiconformal homeomorphisms between arbitrary Riemann surfaces using discrete Beltrami flow, which is a vector field corresponding to the adjustment to the intrinsic Beltrami differential of the map. The vector field is defined by a partial differential equation (PDE) in a local conformal coordinate. Based on this formulation and a composition formula, we can compute the Beltrami flow of any homeomorphism adjustment as a vector field on the target domain defined from the source domain, with appropriate boundary conditions and correspondences. Numerical tests show that our method provides a robust and efficient way of adjusting surface homeomorphisms. It is also insensitive to surface representation and has no limitation to the classes of surfaces that can be processed. Extensive numerical examples will be shown.

Key words. quasiconformal maps, Beltrami differential, homeomorphism adjustments, least squares

AMS subject classifications. 30C62, 30C65, 53A05, 53A30

1. Introduction. Surface processing is an important task in many problems such as 3D modeling and medical image processing. One important aspect of surface processing is how to compute or construct maps between surfaces with some desired properties and correspondences such as in shape deformation, surface registration, shape classification, texture mapping, mesh editing, and etc.

One common and important problem is to define and compute a good map between two surfaces in different applications. One natural choice is to look for a map which minimizes angular distortions, hence preserving local surface geometry. This led to the widely studied conformal maps both mathematically and computationally. While conformal maps preserve local geometry well by minimizing angular distortions, the conformal factor of the map can vary greatly throughout the surface, likely causing significant expansions and compression of the map. Also, in image registration, one needs to enforce the correspondence of landmarks and/or feature curves between two surfaces. As a result, conformal mapping can rarely be achieved.

In order to consider more general surface maps, a generalization called quasiconformal maps, or QC maps can be a more useful tool in many applications. Unlike conformal maps, which aim at eliminating angular distortions, quasiconformal maps take into account that conformal mapping is not always possible. By describing local non-conformal distortions using intrinsic Beltrami differential, one can represent any surface map with bounded non-conformal distortions. Therefore, in principle, any surface homeomorphism that occurs in practice can be represented by a quasiconformal map. In another word, any surface homeomorphism can be characterized by the Beltrami differential. Moreover, many desired properties for surface maps can be expressed in terms of Beltrami differential. Beside the generality of quasiconformal maps, the study of quasiconformal maps with minimal distortions is of interest in its own right. Given two arbitrary surfaces, they may not have the same conformal struc-

tures. In other words, any homeomorphism between them must be quasiconformal. The deformation between different Riemann surfaces has been studied extensively in Teichmüller quasiconformal geometry.

Computing and constructing quasiconformal maps numerically is an important issue in practice. In reviewing the literature, which will be summarized in more details in the next section, we find that earlier works mainly studied the computation of quasiconformal maps between simply or multiply connected planar domains. The computation of quasiconformal maps from a parametrized domain onto a surface in $\mathbb{R}^3$ has also been studied. However, no work was found to compute and adjust maps between arbitrary surfaces in an intrinsic way. This motivates us to create such intrinsic algorithms.

In this work, we propose an approach that can compute quasiconformal homeomorphisms between arbitrary Riemann surfaces using discrete Beltrami flow. The key ingredients in our approach are: (1) Define a vector field that adjust the mapping according to the perturbation of the intrinsic Beltrami differential. In another word, a flow in term of Beltrami differential is translated into a flow in the mapping. Moreover, the vector field is defined by an intrinsic partial differential equation (PDE) in a local conformal coordinate. (2) Based on a composition formula, the Beltrami flow is induced on the target or source domain, with appropriate boundary conditions and correspondences. The resulting flow is solved on a compact domain. With this technique, we are able to directly adjust surface maps between arbitrary compact Riemann surfaces.

The organization of this paper is as follows. In Section 2, we review previous works on computing quasiconformal maps. In Section 3, we first present the theoretical background of Beltrami differential. Then we introduce the Beltrami flow and define the vector field that adjusts the mapping according to the perturbation of the intrinsic Beltrami differential. In Section 4, we give detailed algorithms for our method surface map adjustments in this paper. In Section 5, we show computational results of our algorithms and analyze the accuracy, robustness and efficiency of our approach. We summarize our work in Section 6 and suggest further work in the future.

2. Previous Work. The study of quasiconformal maps is closely related to that of conformal maps. Computational conformal geometry quickly developed after Thurston et al. [16] introduced the notion of circle packing for discretizing smooth surfaces. Using this approach, Collins et al. [3] developed a fast algorithm for finding circle packing metrics, which is a method of finding discrete conformal maps. Later, Yin et al. [20] introduced a method for finding conformal maps called discrete curvature flow, which is a discrete version of the classical Yamabe flow. As numerical methods developed, different applications emerged. Lui et al. [10] proposed to compute conformal mapping of cortical surfaces preserving landmarks for further medical analysis. Lévy et al. [14] proposed to generate texture atlas using least square error of conformality. Conformal mapping of surfaces is closely related to certain geometric problems. For example, one can compute the geodesics on surfaces by mapping them conformally onto the plane and solve a weighted Eikonal equation. To provide some theoretical guarantee on this method, Aflalo et al. [1] proposed an algorithm to compute conformal mappings with conformal factors as uniform as possible, and provided theoretical bounds. Besides, since the conformal structure of a surface carries information of its geometry, conformal maps are also being used in shape analysis. Kim et al. [8] proposed to compute intrinsic maps with surface correspondences by combining different conformal maps.

The computation of quasiconformal maps has also been studied for quite long and more extensively recently. One of the earliest work is by Mastin et al. [15] using finite difference method to compute quasiconformal maps with arbitrary Beltrami coefficient from a multiply connected planar domain onto a rectangular domain with slits removed. Later work by He et al. [7] solved the Beltrami equation using a discrete circle packing approach. The work can be applied on planar domains or any compact Riemann surface whose boundary is a finite union of Jordan curves. However, it mainly served as a constructive proof for the existence of quasiconformal maps, but not a practical algorithm. Daripa et al. [4] proposed a fast algorithm to compute quasiconformal maps by Fourier transform of an integral kernel for efficient integration. However, their method is also restricted to planar domains. Later, Daripa et al. [5] proposed a similar approach of evaluating certain singular integrals for computing quasiconformal mappings of doubly connected domains with unknown conformal modules. Recently, Zeng et al. [21] proposed solving Beltrami equations using discrete holomorphic differential forms. The main idea is to deform the conformal structure of the original surface by the Beltrami coefficient so that the desired map becomes a conformal map. Then techniques for computing conformal maps using discrete holomorphic differential forms can be used. Lately, Zeng et al. [22] used a similar metric deforming idea on their discrete Yamabe flow method to compute quasiconformal maps on Riemann surfaces. In the last two works, the final conformal structure of the target domain cannot be determined at the beginning. Their method can be more appropriately described as finding a quasiconformal parametrization with a given Beltrami coefficient onto a domain with conformal structure to be determined. To the best of our knowledge, no algorithm has been proposed for the computation of homeomorphisms between general domains or surfaces with a given Beltrami coefficient.

Besides the need for computing quasiconformal maps, the applications of quasiconformal maps is also of great interest in applications such as shape deformation, surface registration, texture mapping, brain mapping, mesh editing, and etc. The goal in these applications is to find mappings with desired properties that can be expressed in terms of Beltrami differential or other quantities. For example, one may want to find conformal maps between surfaces in surface mapping applications, because they preserve local geometry well. Lipman et al. [9] used quasiconformal maps to generate plane deformations with less distortion. Lui et al. [13] proposed to optimize surface registrations using an integral flow. Wong et al. [18] used the integral flow method for the inpainting and refinement of diffeomorphisms by computing new diffeomorphisms with inpainted or interpolated Beltrami coefficients. The inpainted maps are smoothly reconstructed in the missing regions and the refined maps are smoothly interpolated. However, for simply connected closed surfaces, the integral flow method has to be computed on an unbounded domain, which is computationally inefficient. Moreover, for simply connected closed surfaces, the method requires a global parametrization through a conformal map onto the plane. As a result, the method is only applicable for computing quasiconformal maps on open or closed simply connected surfaces. Besides applications in texture mapping, quasiconformal maps are also widely used in medical imaging. Lui et al. [11] proposed compressing surface registrations by storing the Fourier coefficients of the Beltrami coefficients. Lui et al. [12] also proposed using geometric-matching quasiconformal maps for hippocampal surface registration. In general quasiconformal maps are very useful tools in many applications, but there is no efficient computational algorithms available that work in general situations. This is the goal of our work.

3. Theoretical Background. In this section, we introduce all the theoretical background of our algorithms. First we define what quasiconformal maps are and show some related properties describing them, such as Beltrami coefficients and Beltrami differentials. Next, we define the Beltrami flow and discuss how it is related to the adjustment of quasiconformal maps. We give an example of the Beltrami flow for the case of Riemann sphere, where the Beltrami flow can be explicitly written as an integral formula. After that, we generalize the Beltrami flow through an intrinsically defined vector field which can be characterized by a PDE in local conformal coordinate. Then we derive a composition formula for quasiconformal maps which allows one to compute Beltrami flow of any homeomorphism adjustment as a vector field on the target domain defined from the source domain. Finally, we introduce the Tutte embedding, which will be useful for producing an initial map for simply connected open and closed surfaces.

3.1. Quasiconformal Maps and Beltrami Coefficients. Quasiconformal maps are generalizations of conformal maps. In conformal maps, the local metric between two surfaces are scaled by a conformal factor. In quasiconformal maps, a bounded non-conformal distortion is allowed. As a motivating example, consider a differentiable function $f\colon \mathbb{C} \to \mathbb{C}$. Write $f(z) = u(x,y) + \sqrt{-1} \cdot v(x,y)$, where u and v are the real and imaginary parts of f represented as real-valued functions on $\mathbb{R}^2$, and $z = x + \sqrt{-1} \cdot y$. f is said to be quasiconformal if it satisfies the Beltrami equation

$$f_{\bar{z}} = \mu f_z, \tag{3.1}$$

where $f_{\bar{z}}$ and f_z are complex derivatives defined by

$$\frac{\partial f}{\partial \bar{z}} = \frac{1}{2}(u_x - v_y) + \frac{\sqrt{-1}}{2}(v_x + u_y) \tag{3.2}$$

and

$$\frac{\partial f}{\partial z} = \frac{1}{2}(u_x + v_y) + \frac{\sqrt{-1}}{2}(v_x - u_y) \tag{3.3}$$

In (3.1), $\mu\colon \mathbb{C} \to \mathbb{C}$ is called the Beltrami coefficient. When $\mu \equiv 0$, (3.1) becomes the well-known Cauchy-Riemann equation and f is conformal. In general μ measures the local non-conformal distortion of f. Locally, f maps a disk onto an ellipse. The direction of maximal expansion is given by $\arg(\mu)/2$, with a factor of $|f_z| \cdot (1 + |\mu|)$. The direction of maximal shrinkage is given by $\arg(\mu)/2 + \pi/2$, with a factor of $|f_z| \cdot (1 - |\mu|)$. The dilation of the ellipse is given by $K = \frac{1+|\mu|}{1-|\mu|}$. An illustration of how μ affects the quasiconformal map $w(z) = z + \mu\bar{z}$ is shown in Figure 3.1.

On maps between Riemann surfaces, Beltrami coefficients are generalized as Beltrami differentials and has the form $\mu(z)\frac{\overline{dz}}{dz}$. Precisely, it means that if $f\colon M \to N$ is a map between two Riemann surfaces, and $w = g(z)$ is the representation of f in conformal local coordinates z of M and w of N, then $\mu(z)$ is the Beltrami coefficient of g. If z_1 is another conformal local coordinate of M, it can be easily derived by the chain rule that the Beltrami differential transforms as $\mu_1(z_1)\frac{\overline{dz_1}}{dz_1}$, where

$$\mu_1(z_1) = \mu(z)\frac{\overline{(dz/dz_1)}}{(dz/dz_1)}. \tag{3.4}$$

Hence, the Beltrami differential can be defined intrinsically on the surface. From now on, we may use Beltrami differential and Beltrami coefficient interchangeably when

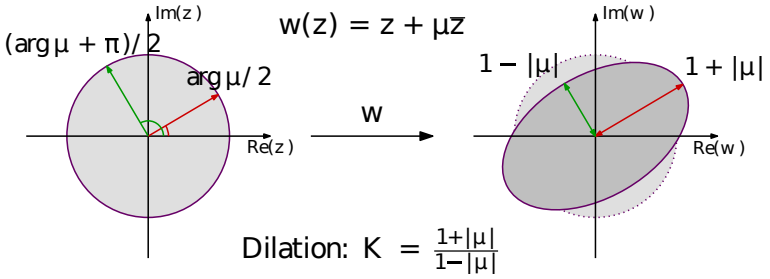


FIG. 3.1. *How the Beltrami coefficient μ affects the quasiconformal map $w(z) = z + \mu\bar{z}$. A disk from the source domain on the left (z-plane) is mapped onto an ellipse in the target domain on the right (w-plane) with a dilation given by K.*

there is no confusion, because representing the Beltrami differential often requires the specification of a conformal local coordinate. For the ease of understanding, one may simply assume that all maps are between complex planes, where the definition of Beltrami coefficient is clear. In fact, in our algorithm, we treat each face of the triangular mesh we consider as a coordinate patch and store the Beltrami differential as the Beltrami coefficient with respect to such coordinate system on each face.

In order to compute the proper adjustment needed to obtain a new surface map corresponding to a perturbed Beltrami coefficient, we introduce the composition formula of Beltrami coefficients. We denote as f^μ the function satisfying the Beltrami equation with Beltrami coefficient μ. Let μ, σ and τ be the Beltrami coefficients of quasiconformal maps f^μ, f^σ and f^τ, with $f^\tau = f^\sigma \circ (f^\mu)^{-1}$. The composition formula for τ is given by

$$\tau = \left(\frac{\sigma - \mu}{1 - \overline{\mu}\sigma} \frac{1}{\theta} \right) \circ (f^\mu)^{-1}, \tag{3.5}$$

where $\theta = \frac{\overline{p}}{p}$ and $p = \frac{\partial}{\partial z} f^\mu(z)$. A proof of this formula can be found in [2]. With this formula, we can compute the Beltrami coefficient τ required so that after composing with f^μ, we obtain a map with a Beltrami coefficient σ.

3.2. Finding the Surface Map with a Given Beltrami Differential. In this subsection, we illustrate how we can find the surface map with a given Beltrami differential. We first look at how this problem can be solved for a sphere, which can be parametrized by the extended complex plane $\overline{\mathbb{C}}$, defined as the complex plane $\mathbb{C}$ plus a point at infinity. Motivated by this example, we define what we called the Beltrami flow for adjusting maps between arbitrary Riemann surfaces.

Consider solving the Beltrami equation on the extended complex plane $\overline{\mathbb{C}}$. It is known as the generalized Riemann mapping theorem that given a Beltrami coefficient $\mu \colon \mathbb{C} \to \mathbb{C}$ with $\|\mu\|_\infty < 1$, there exist a unique homeomorphism f of $\overline{\mathbb{C}}$ with $f(0) = 0$, $f(1) = 1$ and $f(\infty) = \infty$ satisfying the Beltrami equation. Theoretically, the solution can be computed using the following perturbation formula.

THEOREM 3.1. *Let μ_t be the Beltrami coefficient $\mu_t(z) = \mu(z) + t\nu(z)$ and $f = f^\mu$. Then $f^{\mu_t}(w) = f(w) + t\dot{f}(w) + O(t^2)$, where*

$$\dot{f}(w) = -\frac{1}{\pi} \iint \nu(z) R(f(z), f(w)) (f_z(z))^2 \, dx \, dy, \tag{3.6}$$

with

$$R(z,w) := \frac{1}{z-w} - \frac{w}{z-1} + \frac{w-1}{z}. \tag{3.7}$$

We refer interested reader to [2] for a proof of the theorem. Using this formula, one may start from the identity map **Id** with $\mu \equiv 0$ and perturb its Beltrami coefficient in a small step at a time until the map with the required Beltrami coefficient is obtained.

Motivated by this theorem, it is observed that a flow $\dot{f}(w)$ can be defined and computed to adjust the Beltrami coefficient of a map between two surfaces, in this case, the Riemann sphere. If f is the identity map, we can define the flow V as the following function:

$$V(z) := P\nu(z) := -\frac{1}{\pi} \iint \nu(\zeta) \left(\frac{1}{\zeta - z} + \frac{z-1}{\zeta} - \frac{z}{\zeta - 1} \right) d\xi \, d\eta. \tag{3.8}$$

In practice, computing the integral for V over the whole complex plane is infeasible because it is not a compact domain. However, in [6], it is shown that the key property of V is that $\frac{\partial}{\partial \bar{z}} V(z) = \nu(z)$, which is equivalent to saying that the operator P inverts the $\overline{\partial}$-derivative. Therefore, if we can compute the flow properly by solving this equation on a compact domain, we will be able to adjust the map between arbitrary compact Riemann surfaces. We justify this claim for the complex plane $\mathbb{C}$ using the following theorem. Similar result follows for arbitrary surface maps. Note that unless otherwise specified, in all discussions from now on, all Beltrami coefficients μ and adjustments ν are only required to be measurable, and all PDEs are only required to be satisfied weakly. In practice, one may assume all Beltrami coefficients μ and adjustments ν to be piecewise constant on each face of triangular meshes, and all surface maps to be piecewise linear.

THEOREM 3.2. *Let* $\mathbf{id}\colon \mathbb{C} \to \mathbb{C}$ *be the identity map of the complex plane, with Beltrami coefficient* $\mu \equiv 0$*. If* V *is a complex-valued function on* $\mathbb{C}$ *such that*

$$\frac{\partial}{\partial \bar{z}} V(z) = \nu(z) \tag{3.9}$$

for a complex-valued function ν*. Let* $f_t(z) = \mathbf{id}(z) + tV(z)$ *and* μ_t *be the Beltrami coefficient of* f_t*. Then for every* $z \in \mathbb{C}$ *and small* $t > 0$*,* μ_t *satisfies*

$$\mu_t(z) = t\nu(z) + O(t^2). \tag{3.10}$$

Proof. It suffices to show that for all $z \in \mathbb{C}$, the Beltrami coefficient of f_t at z has a derivative equal to $\nu(z)$. For our application, it is sufficient to assume that V is C^1-continuous and that $\frac{\partial}{\partial z} V$ exists and is finite near z. Then we have

$$\frac{\partial f_t}{\partial \bar{z}} = \frac{\partial f}{\partial \bar{z}} + t \frac{\partial V}{\partial \bar{z}} = t\nu \tag{3.11}$$

and

$$\frac{\partial f_t}{\partial z} = \frac{\partial f}{\partial z} + t \frac{\partial V}{\partial z} = 1 + tV_z. \tag{3.12}$$

Then μ_t, the Beltrami coefficient of f_t, is given by

$$\mu_t = t\nu/(1 + tV_z). \tag{3.13}$$

Differentiating μ_t with respect to t and taking its value at $t = 0$, we have

$$\frac{\partial \mu_t}{\partial t}\Big|_{t=0} = \frac{(1 + tV_z)\nu - t\nu V_z}{(1 + tV_z)^2}\Big|_{t=0} = \nu. \tag{3.14}$$

Therefore μ_t satisfies (3.10) as required. □

To formulate (3.9) on an arbitrary Riemann surface, we generalize the definition of V as a vector field on the surface and call it a Beltrami flow.

DEFINITION 3.3. *Let $\nu(z)\frac{\overline{dz}}{dz}$ be a Beltrami differential on a surface M. We define a* Beltrami flow *of M with respect to ν to be the vector field $V\frac{\partial}{\partial z}$ on M if under conformal local coordinates, we have*

$$\frac{\partial}{\partial \bar{z}} V(z) = \nu(z). \tag{3.15}$$

With the result of Theorem 3.1, an example of Beltrami flow can be given for the extended complex plane $\overline{\mathbb{C}}$, which is homeomorphic to the Riemann sphere. Without loss of generality, by changing the 3 points fixed by the homeomorphism (0, 1 and ∞ in Theorem 3.1) to 0, 1 and -1, for any Beltrami coefficient $\mu(z)$, there is a unique homeomorphism $f\colon \overline{\mathbb{C}} \to \overline{\mathbb{C}}$ with Beltrami coefficient $\mu(z)$ and satisfies $f(0) = 0$, $f(1) = 1$ and $f(-1) = -1$. As such, the Beltrami flow $V(z)$ with respect to any Beltrami coefficient adjustment $\nu(z)$ satisfies (3.15) and $V(0) = V(1) = V(-1) = 0$ (since 0, 1 and -1 are fixed). Since $V(z)$ is precisely given by (3.8), we have the following example of a Beltrami flow:

EXAMPLE 3.4. *Define $V(z)$ as in (3.8). Then $V(z)$ is the unique Beltrami flow on the unit sphere with respect to the Beltrami differential $\nu(z)\frac{\overline{dz}}{dz}$ satisfyig $V(0) = 0$, $V(1) = 0$ and $V(-1) = 0$, where the whole sphere is parametrized by a single conformal coordinate patch using the standard stereographic projection.*

From this example, it can be seen that the Beltrami flow or the quasiconformal map with respect to a certain Beltrami differential may not be unique unless one or more points are fixed. It is because there can be conformal diffeomorphism of the surface that can be flowed from the identity map, such as the Möbius transformation for a sphere. In other words, there can be flows from the identity map preserving the Beltrami differential, therefore causing the non-uniqueness. As we will explain, using our least square approach, we can easily overcome this problem.

Another key idea in our approach is to define a proper vector field and flow intrinsically on the source or the target surfaces using a composition formula. In this way, we avoid the use of any global parametrization and allow us to compute maps for any compact domains directly. The idea is to compose a surface map $f\colon M \to N$ between Riemann surfaces M and N with a homeomorphism of N induced by a Beltrami flow with respect to some Beltrami differential ν, so that the Beltrami differential of f is adjusted in a specific way with the composition. The appropriate ν can be easily computed from the composition formula (3.5) to achieve the map with a desired Beltrami differential. To simplify the discussion, we assume that there is a global parametrization on every Riemann surface, although derivation of the formula can be carried out in any local conformal coordinate in the same way. To adjust the Beltrami coefficient of a surface map $f\colon M \to N$ between two Riemann surfaces, our approach is to look for a homeomorphism $g\colon N \to N$ such that $f \circ g$ has the desired Beltrami coefficient. Suppose that $\mu(z)$ is the Beltrami coefficient of f and our target Beltrami coefficient is $\sigma(z, t) = \mu(z) + \nu(z)t$, where $\nu(z)$ is some adjustment to μ.

Using the composition formula (3.5), the required Beltrami coefficient $\tau(z,t)$ of g is given by

$$\tau(z,t) = \left(\frac{\nu t}{1-\overline{\mu}(\mu+\nu t)}\frac{1}{\theta}\right) \circ (f^{\mu})^{-1}. \tag{3.16}$$

To compute the flow we need to construct g, we differentiate $\tau(z,t)$ with respect to t and obtain

$$\frac{\partial \tau}{\partial t}(z,t)|_{t=0} = \left(\frac{\nu}{1-|\mu|^2}\frac{1}{\theta}\right) \circ (f^{\mu})^{-1}. \tag{3.17}$$

Define ρ to be the derivative above. The problem of adjusting the Beltrami differential of f becomes the problem of computing a Beltrami flow of N with respect to ρ. We concretely express this into the following key theorem:

THEOREM 3.5. *Suppose $f\colon M \to N$ is a surface map between two Riemann surfaces (parametrizable with one coordinate patch) with Beltrami coefficient $\mu(z)$. Define ρ to be the derivative in (3.17). Suppose V is a Beltrami flow on N with respect to ρ, and $g\colon N \to N$ is a homeomorphism induced by V, i.e.,*

$$g(z,t) := z + V(z)t. \tag{3.18}$$

Then for small $t > 0$, $g(f(z),t)$ is a surface map with Beltrami coefficient

$$\sigma(z,t) = \mu(z) + \nu(z)t + O(t^2). \tag{3.19}$$

Note that ideally, one wants the $O(t^2)$ to be small such that $\sigma(z,1)$ is close to $\mu(z)+\nu(z)$, the required Beltrami coefficient. In our algorithms, we choose $t \in (0,1]$ such that it is as close to 1 as possible and makes $\sigma(z,t) - \mu(z) - \nu(z)t$ smaller in L^2 sense. As we approaches the desired Beltrami coefficient, $O(t^2)$ becomes negligible as $\nu(z)$ becomes closer and closer to 0. Therefore, our algorithms converges despite choosing $t = 1$ whenever possible.

4. Numerical Algorithms. In this section, we explain our numerical algorithms in detail for the computation of arbitrary surface maps. Our main focus is on the computation of discrete Beltrami flows on surfaces represented by triangular meshes, where the vector field corresponding to a Beltrami flow is computed using a least square approach. We first give a brief review of Tutte embedding, which can be used to create an initial map for simply connected surfaces with or without boundary. Then we discuss the computation of Beltrami flows for simply connected surface with boundary, which can be parametrized by a single coordinate patch. Finally, we generalize to other cases where more than one coordinate patches are required, which include simply connected surfaces and surfaces of higher genus.

4.1. Initialization by Tutte Embedding for Simply Connected Surfaces. To compute a surface map $f\colon M \to N$ between arbitrary compact Riemann surfaces with a desired Beltrami differential, our approach is to first start with an arbitrary initial map $f_0\colon M \to N$, and then adjust the Beltrami differential of f_0 by composing it with a map $g\colon N \to N$, where g is induced by a Beltrami flow with respect to a correct Beltrami differential given by the composition formula (3.5). For the ease of discussion, we may assume that f_0 is a homeomorphism, or that it is onto and

non-overlapping.[1] Our algorithms also converge faster when it is initialized using a homeomorphism.

For simply connected surfaces with boundary, we use the well-known Tutte embedding [17], which is widely used for mesh embedding and initialization problems. Consider the triangular mesh of the surface represented by a graph $G = \langle V, E, F\rangle$, and its boundary ∂G to be embedded in the plane as a (not necessarily strictly) convex polygon, the x and y coordinates of the embedding can be obtained by solving the following systems of linear equations:

$$\sum_{v_j \in N(v_i)} \frac{1}{|N(v_i)|} x_j = x_i, \quad i = 1, \ldots, |V| - |\partial G|, \tag{4.1}$$

$$\sum_{v_j \in N(v_i)} \frac{1}{|N(v_i)|} y_j = y_i, \quad i = 1, \ldots, |V| - |\partial G|, \tag{4.2}$$

where $\{v_1, \ldots, v_{|V|-|\partial G|}\}$ are all the interior points of G, $N(v_i)$ is the set of neighbors of v_i, and $|N(v_i)|$ is the number of neighbors of v_i. Under Tutte embedding, each point of a graph is mapped onto the plane such that it is the barycenter of its neighbors. The mapped graph is always planar using this method and the mapping is easily solved by the above linear systems. For simply connected closed surfaces, one may first remove a triangle on the triangular mesh and apply Tutte embedding to get a planar graph. Then the graph can be easily mapped onto a sphere by stereographic projection from a plane, with the triangle put back to fill the hole. For other types of surfaces, one may first create an arbitrary map. As the results show, our algorithms could work even when the initial map is overlapping.

4.2. Computing Beltrami Flows Using a Least Square Method. The key step in computing Beltrami flows is to solve the vector field $V(z)$. We use least square method to solve the first order PDE system (3.9) for planar domains or (3.15) on general Riemann surfaces rather than solving a second order PDE for the real or imaginary component separately. Advantages of using this approach include (1) numerical approximation of lower order differentials, (2) no subtle boundary conditions coupling the real and imaginary part when computing quasiconformal maps for domains with boundary, which will be discussed in more details later, (3) providing a good approximation when the exact solution may not exist, e.g. when there are too many constraints, or difficult to approximate discretely in practice. In this subsection, we consider the problem of computing a Beltrami flow to adjust a map between planner domains. Once we establish the problem for planar domains, it can be directly generalized to arbitrary surface maps.

In order to compute a Beltrami flow for a Riemann surface with respect to some Beltrami differential ν, consider solving the problem on a Riemann surface S parametrized by a single conformal coordinate patch $D \subset \mathbb{C}$, and ν is the adjustment to the Beltrami coefficient on D. Let $\psi\colon S \to D$ be the parametrization. Then the Beltrami flow can be defined as a complex-valued function on D that satisfies $\frac{\partial}{\partial \bar{z}} V(z) = \nu(z)$. Suppose $\nu(z) = a(z) + \sqrt{-1} \cdot b(z)$ and $V(z) = u(z) + \sqrt{-1} \cdot v(z)$, where a, b and u, v are real-valued functions representing the real and imaginary parts

[1] Otherwise the Beltrami flow inducing g will be a vector field on N with source domain M, which can be handled similarly but makes the notation more cumbersome.

of ν and V respectively, then u, v satisfies the system

$$\begin{aligned} u_x - v_y &= 2a, \\ v_x + u_y &= 2b. \end{aligned} \tag{4.3}$$

Using a least square approach, we minimize a least square integral of $\frac{\partial}{\partial \bar{z}} V(z) - \nu(z)$ weighted according to the area of the original patch S. Since the real and imaginary parts are given by the above system, the energy to minimize becomes

$$\int_S \left\{ \left[(u_x - v_y - 2a)^2 + (v_x + u_y - 2b)^2 \right] \circ \psi \right\} \, d\mathcal{A}. \tag{4.4}$$

The original problem of finding V becomes a standard least square problem, which can be solved easily. When V is computed numerically on a triangular mesh, we call the discrete flow computed using the above method a discrete Beltrami flow.

When the surface S cannot be parametrized using a single patch, we can write the least square using a number of patches. Suppose M is the union of of $N_1, N_2, \ldots, N_k \subset M$ with disjoint interior and each can be parametrized by $\psi_1, \psi_2, \ldots, \psi_k$ onto closed simply connected domains $D_1, D_2, \ldots, D_k \subset \mathbb{C}$ respectively, and $\nu_1, \nu_2, \ldots, \nu_k$ are the adjustments to the Beltrami coefficients on $D_1, D_2, \ldots, D_k$ respectively. Suppose for each $i = 1, 2, \ldots, k$, $\nu_i(z) = a_i(z) + \sqrt{-1} \cdot b_i(z)$ and $V_i(z) = u_i(z) + \sqrt{-1} \cdot v_i(z)$, where a_i's, b_i's and u_i's, v_i's are real-valued functions representing the real and imaginary parts of ν_i's and V_i's respectively. The least square energy (4.4) generalizes as

$$\sum_{i=1}^{k} \int_{N_i} \left\{ \left[(u_{ix} - v_{iy} - 2a_i)^2 + (v_{ix} + u_{iy} - 2b_i)^2 \right] \circ \psi_i \right\} \, d\mathcal{A}. \tag{4.5}$$

If ∂D_i and ∂D_j intersect, then for all $p \in \partial D_i \cap \partial D_i$, V_i and V_j have to satisfy a compatibility condition given by

$$V_j(\psi_j(p)) = J_{ij}(p) V_i(\psi_i(p)), \tag{4.6}$$

where $J_{ij}(p)$ is the Jacobian matrix of $\psi_j \circ \psi_i^{-1}$, and V_i, V_j are considered as vectors in $\mathbb{R}^2$. With these compatibility conditions, (4.5) becomes a constrained least square problem. This is helpful when one already has a parametrization for the surface. In general, all least square problems above can be formulated as an intrinsic integral on the surface, as we show in the later subsections.

When the surface considered has boundaries, one restricts the direction of flow to be along the tangential direction of the boundary. In this case, the above problems also become constrained least square problems. Sometimes it is also necessary to fix the flow to be 0 at a few points. For example, as shown in Example 3.4, 3 points have to be fixed for the case of a sphere. It is also possible to enforce more constraints as required, making the least square approach more flexible.

4.3. Discretization of Beltrami Differentials. On a general Riemann surface, a Beltrami differential is a $(1, -1)$-tensor specifying the direction and magnitude of the distortion. The Beltrami differential can be written as a Beltrami coefficient only if a local coordinate is given. As our algorithms work on discrete triangular meshes, in this subsection, we illustrate how we can represent Beltrami differentials as Beltrami coefficient on each face of a triangular mesh.

On each face of a triangular mesh determined by 3 points p_1, p_2 and p_3, a natural way to specify a local coordinate system is to identify $\overrightarrow{p_1 p_2}$ as the positive real direction

and the direction perpendicular to it as the positive imaginary direction. For example, if p_1, p_2 and p_3 are mapped to q_1, q_2 and q_3 respectively, we may map p_1, p_2, p_3 conformally to $(0,0)$, $(a_1,0)$, (a_2,a_3) and q_1, q_2, q_3 conformally to $(0,0)$, $(b_1,0)$, (b_2,b_3). Denote this linear map from $\mathbb{R}^2$ to $\mathbb{R}^2$ as $f(z) = u(x,y) + \sqrt{-1}\cdot v(x,y)$, where $z = x + \sqrt{-1}\cdot y$. Then we have

$$\begin{pmatrix} u_x & u_y \\ v_x & v_y \end{pmatrix} \begin{pmatrix} a_1 & a_2 \\ 0 & a_3 \end{pmatrix} = \begin{pmatrix} b_1 & b_2 \\ 0 & b_3 \end{pmatrix}. \tag{4.7}$$

Therefore, the constants u_x, u_y, v_x and v_y can be solved as

$$\begin{pmatrix} u_x & u_y \\ v_x & v_y \end{pmatrix} = \begin{pmatrix} b_1 & b_2 \\ 0 & b_3 \end{pmatrix} \begin{pmatrix} a_1 & a_2 \\ 0 & a_3 \end{pmatrix}^{-1}. \tag{4.8}$$

The Beltrami coefficient of the face can then be computed using the formula

$$\mu = \frac{u_x - v_y + \sqrt{-1}\cdot(v_x + u_y)}{u_x + v_y + \sqrt{-1}\cdot(v_x - u_y)}. \tag{4.9}$$

In our algorithms, all Beltrami differentials are represented as a Beltrami coefficient on each face of a triangular mesh in the form described above.

4.4. Computing Discrete Beltrami Flows on Simply Connected Surfaces with Boundary. In the simplest case, the surface M to be considered is a simply connected open surface, where only a single patch is necessary. We may map M conformally onto a domain $D \subset \mathbb{R}^2$. We may also assume that the required Beltrami coefficient ν is specified on D, i.e., ν is a complex-valued function on D. Then the problem reduces to computing the discrete Beltrami flow V on D such that

$$\frac{\partial}{\partial \bar{z}} V(z) = \nu(z). \tag{4.10}$$

As shown in Subsection 4.2, if we write $V(z) = u(z) + \sqrt{-1}\cdot v(z)$, and $\nu(z) = a(z) + \sqrt{-1}\cdot b(z)$, then the above PDE can be solved by minimizing the energy functional (4.4)

$$\int_S \left\{ \left[(u_x - v_y - 2a)^2 + (v_x + u_y - 2b)^2 \right] \circ \psi \right\} d\mathcal{A}. \tag{4.11}$$

In the Appendix, a detailed description is given on how to formulate a least square problem on a triangular mesh for solving for u and v. As discussed in Subsection 4.2, along the boundary, u and v is restricted by a linear relation to keep the flow along the tangential direction of the boundary. Once the discrete Beltrami flow V is solved, the perturbed map

$$f(z,t) = z + tV(z) \tag{4.12}$$

has the desired Beltrami coefficient $\mu_t(z) = t\nu(z)$ for small $t > 0$. Then we choose $t \in (0,1]$ such that it is as close to 1 as possible and decreases $\sigma(z,t) - \mu(z) - \nu(z)t$ in L^2 sense and adjust f according to the flow V and time step t. After the adjustment, we project the boundary points back onto the boundary and interpolate the projection movement to adjust the interior points. This completes one step of adjustment using discrete Beltrami flow.

4.5. Computing Discrete Beltrami Flow on General Surfaces. For general Riemann surfaces, since there may not be a global conformal parametrization as above, we have to model V as a vector field lying on the tangent plane at each point of the surface. To compute the discrete Beltrami flow on general surfaces, we assume that we are given a triangular mesh $\{P, T\}$ with the desired discrete Beltrami differential μ of the quasiconformal map ϕ to be computed onto a surface S. Suppose $P = \{p_1, \ldots, p_n\} \subset \mathbb{R}^3$ and $T = \{[p_{11}, p_{12}, p_{13}], \ldots, [p_{m1}, p_{m2}, p_{m3}]\}$. Assume that initially ϕ maps p_i to $q_i := \phi(p_i) \in S$. We need to adjust ϕ by a vector field V on S with P as its domain so that $\phi(p_i)$ can be perturbed to $\phi(p_i) + \alpha V(p_i)$, where α is the step size. Since there is no standard parametrization of tangent spaces in S, we compute V discretely by specifying the displacement u_i, v_i for each i with respect to a positively oriented basis of the tangent space at q_i. Denote the outward normal at q_i by $\mathbf{n}_i$ for $i = 1, 2, \ldots, n$. For each i, we also compute $\mathbf{u}_i$, $\mathbf{v}_i$ such that $\{\mathbf{n}_i, \mathbf{u}_i, \mathbf{v}_i\}$ is a positively oriented orthonormal basis. Then the required vector field V becomes

$$V(p_i) = u_i \mathbf{u}_i + v_i \mathbf{v}_i, \tag{4.13}$$

for $i = 1, 2, \ldots, n$.

Note that for each point p_i in P, $V(p_i)$ lies on the tangent plane at $\phi(p_i)$. To generalize the least square energy (4.4), one integrates on each face of the triangular mesh $\{\phi(P), T\}$ instead of on the plane. For each triangle $T_i = [p_{i1}, p_{i2}, p_{i3}]$, we map the image $\phi(T_i) = [\phi(p_{i1}), \phi(p_{i2}), \phi(p_{i3})]$ of it isometrically onto an (r, s)-coordinate plane local to the triangle using a conformal map ψ_i. With this coordinate system, we immediately have a basis $\{\mathbf{r_i}, \mathbf{s_i}\} \subset \mathbb{R}^3$ spanning the plane of $\phi(T_i)$. Then we project V onto the face $\phi(T_i)$ to obtain the corresponding value of V in terms of $\mathbf{r_i}$ and $\mathbf{s_i}$, i.e.,

$$V(p_{ij}) = r_{ij} \mathbf{r}_i + s_{ij} \mathbf{s}_i, \tag{4.14}$$

for $i = 1, 2, \ldots, n$ and $j = 1, 2, 3$. On triangle T_i, by writing a point as an (x, y)-coordinate and its image point as an (r, s)-coordinate, V may be regarded as a linear map in $\mathbb{R}^2$. Write $V(x, y) = r(x, y) + \sqrt{-1} \cdot s(x, y)$. Then the integrand in (4.4) can be generalized as

$$\sum_{i=1}^{m} \iint_{\psi_i(T_i)} (r_x - s_y - 2a)^2 + (r_x + s_y - 2b)^2, \tag{4.15}$$

where the desired Beltrami differential satisfied by V is written as a Beltrami coefficient on triangle $\psi_i(T_i)$ as $a + \sqrt{-1} \cdot b \in \mathbb{C}$.

It is obvious that on every image $\psi(T_i)$, the derivatives r_x, r_y, s_x and s_y are constant and can be expressed as a linear expression of u_{i1}, u_{i2}, u_{i3} and v_{i1}, v_{i2}, v_{i3}. Therefore the energy functional for the least square problem is quadratic in terms u_i's and v_i's. A detailed description is given in the Appendix for the precise least square problem to solve. For boundary points p_i, we may choose $\mathbf{u}_i$ to point in the tangential direction of the boundary and constrain v_i to be 0 in the quadratic minimization problem. Then the resulting flow is automatically a valid flow for adjusting homeomorphisms on S. We choose the time step t in a way similar to that of simply connected surfaces with boundary discussed in the previous subsection. After adjusting the surface map according to V, we project the resulting points back onto S and the resulting boundary points back on ∂S. This completes one step of adjustment using discrete Beltrami flow.

Level	Number of Faces	Number of Steps	Error (sup-norm)	Error (L^2-norm)
0	4	4	0.2000	0.0800
1	16	12	0.2795	0.0813
2	64	12	0.2251	0.0458
3	256	13	0.1767	0.0176
4	1024	12	0.1060	0.0052
5	4096	10	0.0582	0.0013
6	16384	9	0.0306	0.341×10^{-4}
7	65536	9	0.0157	0.855×10^{-5}

TABLE 5.1
Results for computing a diffeomorphism of the unit disk with $\mu(z) = 0.6z$ with tolerance 10^{-8}.

5. Numerical Tests. In this section, we present various numerical tests to demonstrate efficiency, accuracy and robustness of our proposed method. Our computations are all implemented in MATLAB using vectorized scripts on a quad-core mobile system with 2.6GHz CPU (boosts up to 3.6GHz) and 8GB of 1600GHz memory.

5.1. Validation Using Synthetic Examples. In this subsection, we validate our algorithm by computing the solution of a given Beltrami equation and show convergence as the mesh size gets refined.

We first test our algorithm by constructing diffeomorphisms with various Beltrami coefficients on the unit disk $D = \{x + \sqrt{-1} \cdot y | x^2 + y^2 <= 1\}$. It is known that if we fix 0 and 1, i.e. enforcing the constraint that $f(0) = 0$ and $f(1) = 1$, then the solution of the Beltrami coefficient for an arbitrary μ with $\|\mu\|_\infty < 1$ is unique. To apply discrete Beltrami flow algorithm for simply connected surfaces with boundary, we initialize the map with the identity map of the disk. In each step, we compute the discrete Beltrami flow with the required Beltrami coefficient fixing 0 and 1, where the flow is constrained in tangential direction along the boundary. The time step t is chosen in $[0, 1]$ using a method analogous to bisection such that it is as close to 1 as possible without causing overlapping. Then the map is updated by the flow to approximate the map with the target Beltrami coefficient. The convergence criterion was determined by if the sup-norm of the adjustment of the surface map in $\mathbb{R}^2$ is smaller than 10^{-8}.

We test the accuracy of the algorithm by computing the diffeomorphism on the unit disk with $\mu(z) = 0.6z$. The complexity of the triangular meshes for the disk varies from level 0 (4 faces) to level 7 ($4^{7+1} = 65536$ faces), where each higher complexity level is obtained by subdividing the triangular mesh of the previous level. The number of steps used and the error in sup-norm and L^2-norm is shown in Table 5.1. We observe that except at level 0, the number of steps required tends to decrease mildly as the mesh complexity grows. This could be attributed to the more accurate computed flow as the mesh becomes more complex, decreasing the number of steps required.

A plot of the logarithm of the sup-norm and L^2-norm errors against the logarithm of the number of faces is shown in Figure 5.1. As can be seen, both errors decrease as the mesh is refined. Since the map in some regions is highly distorted, the sup-norm will be larger in those regions. An adaptive mesh will produce better results. However, the numerical result shows that the convergence in L^2-norm is second order. A plot of the resulting diffeomorphism for 65536 faces is shown in Figure 5.2.

Similar results can be observed for computing diffeomorphisms of the unit sphere. In this test, we compute a diffeomorphism with $\mu(z) \equiv 0.3 + \sqrt{-1} \cdot 0.3$, where z

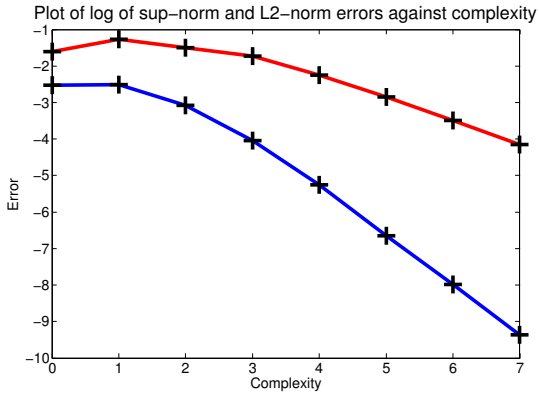
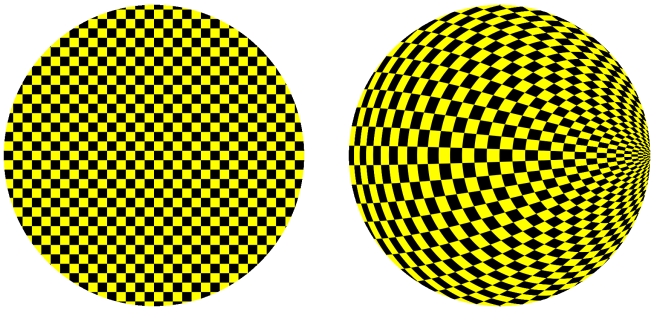


FIG. 5.1. *The sup-norm (red) and L^2-norm (blue) error of the discrete Beltrami flow method on the unit disk.*



(a) (b)

FIG. 5.2. *The result of our algorithm for finding a diffeomorphism of the unit disk with $\mu(z) = 0.6z$ and keeping 0 and 1 fixed. (a) shows a unit disk with a checkerboard pattern. (b) shows how the original checkerboard pattern is mapped onto the unit disk under the computed diffeomorphism.*

is the parametrization by stereographic projection of the sphere onto $\mathbb{C}$. We show convergence for triangular meshes on a sphere with complexity from level 0 (8 faces) to level 6 ($8 \cdot 4^6 = 32768$ faces), where each higher complexity level is obtained by mesh refinement of the previous level. The convergence criterion was determined by if the sup-norm of the adjustment of the surface map in $\mathbb{R}^2$ is smaller than 10^{-5}. The number of steps required, and the error in L^2-norm is shown in Table 5.2. Again, we observe that except at level 0, the number of steps required tends to decrease mildly as the mesh complexity grows. This could also be attributed to the more accurate computed flow as the mesh becomes more complex, decreasing the number of steps required.

A plot of the logarithm of the L^2-norm error against the logarithm of the number of faces is shown in Figure 5.3, which also demonstrates a second order convergence in L^2-norm. A plot of the resulting diffeomorphism for 32768 faces is shown in Figure 5.4.

5.2. Simply Connected General Surfaces. Since using Beltrami flow needs to start from an initial map, we demonstrate that our numerical algorithm is quite robust with respect to the initial map. In Figure 5.5 and Figure 5.6, we computed the conformal map of a fandisk and a hippocampus mesh by following the Beltrami flow

Level	Number of Faces	Number of Steps	Error (L^2-norm)
0	8	9	0.1246
1	32	13	0.3892
2	128	14	0.2692
3	512	14	0.1145
4	2048	12	0.0412
5	8192	11	0.0138
6	32768	10	0.0044

TABLE 5.2

Results for computing a diffeomorphism of the unit sphere with $\mu(z) \equiv 0.3 + \sqrt{-1} \cdot 0.3$ *with tolerance* 10^{-5}.

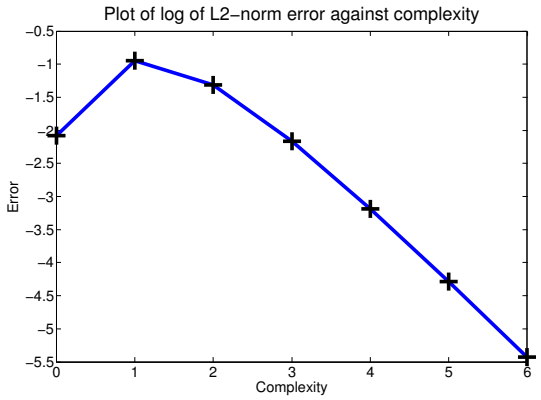
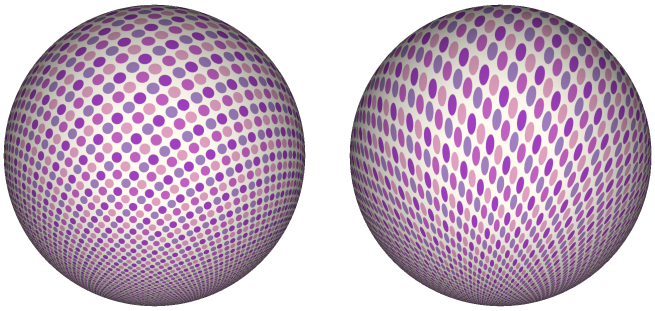


FIG. 5.3. *The* L^2*-norm error of the discrete Beltrami flow algorithm on the unit sphere. The error decreases steadily as complexity increases, justifying the* $O(h^2)$ *error bound of our algorithm.*

that minimizes $\int |\mu|^2$. To get the initial map, we simply projecting them onto a unit sphere centered at an interior point. For both cases, the magnitude of initial Beltrami coefficients are very close to 1 and even bigger than 1 at some places, indicating strong distortion and even overlapping of the initial maps. Using our algorithm, both of these meshes converge to the desired conformal maps.

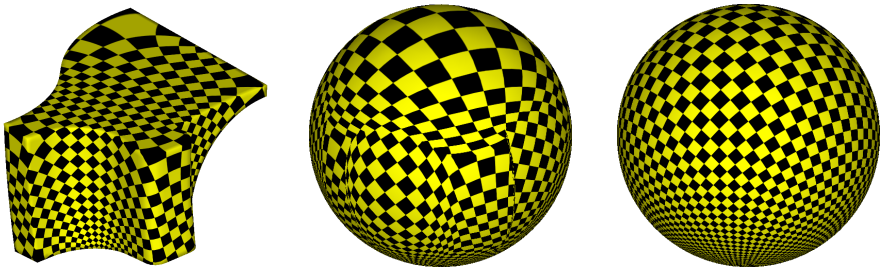
In general, convergence for the Beltrami flow may not be always possible due to the highly non-convex nature of the least square energy in terms of the map. However, in our tests, our algorithm always converges if the initial map from the mesh onto the target domain is not overlapping. Fortunately, Tutte embedding [17] can be used to produce a one-to-one map from any triangulated surface to the complex plane, which can be used as the initial map when necessary. To demonstrate the effectiveness of this initialization, we show the examples of a fish and a cortical surface in Figure 5.7 and Figure 5.8 respectively, where convergence was not achieved if the initial maps were simple projections.

5.3. Multiply Connected Domains and Surface Maps with Hard Constraints. Our algorithms also work on multiply connected surfaces with more than one boundary. Just like the Beltrami flow needs to lie on the tangent plane of the target surface, when we compute a Beltrami flow for a multiply connected surface, we restrict the direction of the flow to the direction of the tangent line at a point of the boundary. As such, the least square problem becomes a constrained energy minimization problem, whose linear system is still sparse and easy to solve. In Figure 5.9(b), a homemorphism of an annulus with inner radius 0.4 and outer radius 1, centered at



(a) (b)

FIG. 5.4. *The result of our algorithm for finding a diffeomorphism of the unit sphere with* $\mu(z) = 0.3 + \sqrt{-1} \cdot 0.3$ *and keeping* 0, 1 *and* ∞ *(as in stereographic projection) fixed. (a) shows a unit sphere with a color disk pattern. (b) shows how the original color disk pattern is mapped onto the unit sphere under the computed diffeomorphism.*



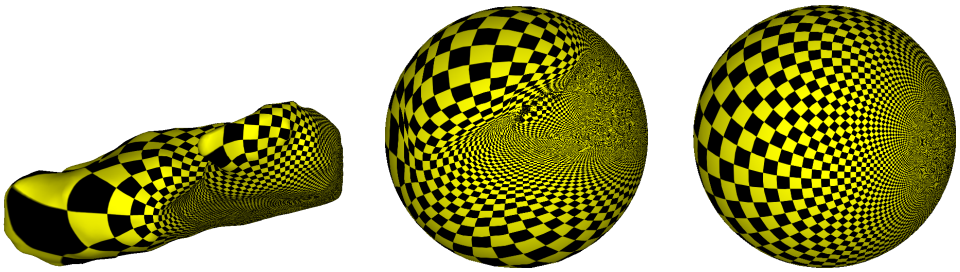
(a) (b) (c)

FIG. 5.5. *Computing a conformal map from a fandisk onto the unit sphere. (a) shows the original fandisk mesh. (b) shows the overlapping initial map constructed by projecting vertices of the mesh onto a unit disk centered at an interior point of the mesh. (c) shows the final map after the Beltrami flow algorithm is used to minimize the Beltrami coefficient in least square sense. All meshes are textured consistently with correspondence from a stereographic projection of the unit sphere in (c).*

0, is computed with Beltrami coefficient $\mu(z) = 0.6z^3$ and keeping the point $z = 1$ fixed. As can be observed, our algorithm is able to keep its boundaries fixed and find a homemorphism with the required distortion.

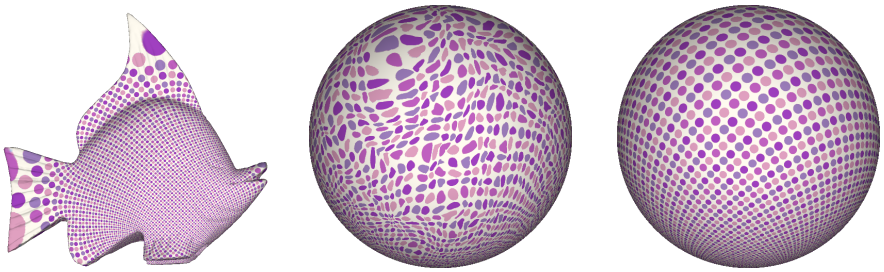
Using the same technique, our algorithm also works on surface maps with hard constraints. For example, we may force some curves on the source surface to be mapped onto some fixed curves on the target surface and adjust the map fixing these constraints. This is demonstrated in Figure 5.10 by mapping the Stanford Bunny onto a sphere with a single curve constraint. To get an initial map for this example, observe that if we cut open the Stanford Bunny along the curve, the surface becomes a simply connected open surface and can be easily mapped onto a disk by Tutte embedding. After that, it can be further mapped onto a sphere with the 2 sides of the cut identified along a fixed curve. We test our algorithm by computing a map with a minimized L^2-norm of μ while keeping a single curve fixed. The resulting map is smooth and shows very low angular distortions. The curve on Bunny is also mapped onto the desired target curve.

We can also compute a mapping with more than one curve constraint. In medical



(a) (b) (c)

FIG. 5.6. *Computing a conformal map from a hippocampus onto the unit sphere. (a) shows the original hippocampal mesh. (b) shows the overlapping initial map constructed by projecting vertices of the mesh onto a unit disk centered at an interior point of the mesh. (c) shows the final map after the Beltrami flow algorithm is used to minimize the Beltrami coefficient in the least square sense. All meshes are textured consistently with correspondence from a stereographic projection of the unit sphere in (c).*

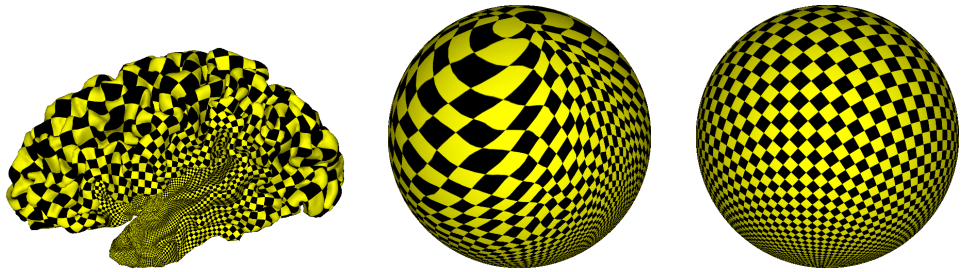


(a) (b) (c)

FIG. 5.7. *Computing a conformal map from a fish onto the unit sphere using discrete Beltrami flow with Tutte embedding. (a) shows the original fish mesh. (b) shows the initialization by Tutte embedding. (c) shows the final map after the Beltrami flow algorithm is used to minimize the Beltrami coefficient in least square sense. All meshes are textured consistently with correspondence from a stereographic projection of the unit sphere in (c).*

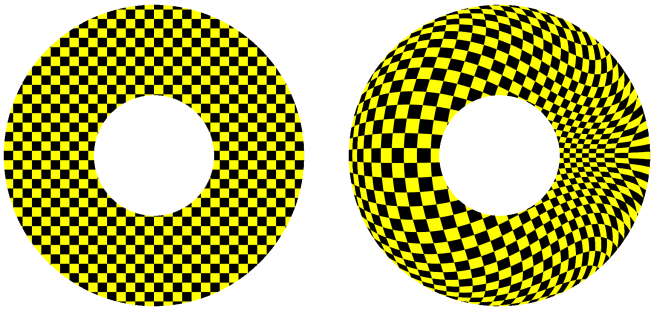
imaging, it is often necessary to register the same class of surfaces (e.g. cortical surfaces) onto a canonical domain, where feature curves are mapped onto consistent locations. In this example, we label 3 feature curves on a cortical surface, as shown in Figure 5.11(a), which are to be mapped onto 3 geodesics on the sphere. Initially, the cortical surface is mapped onto the sphere using Tutte's embedding. As seen in Figure 5.11(b), the feature curves are not mapped onto geodesics. Using our algorithm, we compute a flow that minimizes the L^2-norm of the Beltrami differential. At the same time, we enforce the vector field along the feature curves to point towards the target geodesics. When our algorithm converges, the feature curves are all mapped onto geodesics, as can be seen in Figure 5.11(c), with the distortion spread evenly over the whole sphere. This shows that our algorithm can easily handle more complicated constraints without any modifications.

5.4. Results for General Meshes with Non-Trivial Topology. In this subsection, we test our algorithm on general meshes with non-trivial topology to demonstrate the extra procedures needed using genus one examples. The same procedures can be applied to surfaces with higher genus.



(a) (b) (c)

FIG. 5.8. *Computing a conformal map from a cortical surface onto the unit sphere using discrete Beltrami flow with Tutte embedding. (a) shows the original cortical surface mesh. (b) shows the initialization by Tutte embedding. (c) shows the final map after the Beltrami flow algorithm is used to minimize the Beltrami coefficient in the least square sense. All meshes are textured consistently with correspondence from a stereographic projection of the unit sphere in (c).*



(a) (b)

FIG. 5.9. *The result of our algorithm for finding a diffeomorphism of an annulus with* $\mu(z) = 0.6z^3$ *and keeping the point* $z = 1$ *fixed. (a) shows an annulus with inner radius* 0.4*, outer radius* 1*, centered at* 0 *and textured with a checkerboard pattern. (b) shows how the original checkerboard pattern is mapped onto the annulus under the computed diffeomorphism.*

The main procedure involved for two triangulated surfaces is the computation of an initial bijective map from the source mesh onto the target mesh, based on Tutte embedding, and later evolved bijective maps, based on discrete Beltrami flow, from the target mesh onto itself. In the general discrete setting, vertices are most likely mapped inside triangles of the mesh. To construct an initial bijective map for genus one surfaces, we can construct 2 loops across a particular vertex to cut the mesh into an open surface. Then Tutte embedding can be used to map each surface onto $[0, 1]^2$. Once the embeddings for both surfaces are computed, a correspondence between the source and target meshes can be established and the intermediate mapping can be discarded. For two triangulated surfaces with genus $g > 1$, one can construct an initial map by mapping each surface onto a fundamental polygon in the standard hyperbolic disc after cutting the surface open along $2g$ homotopically different loops at a vertex following the procedure proposed in [19]. To explicitly represent the correspondence of the two meshes, we need to find the mapped locations of the vertices of the source mesh, e.g., inside which triangle of the target mesh. For the initial map, after each triangulated surface is mapped to $[0, 1]^2$, we first perform a search using a kd-tree to determine a set of closest neighboring vertices of the target mesh for each source

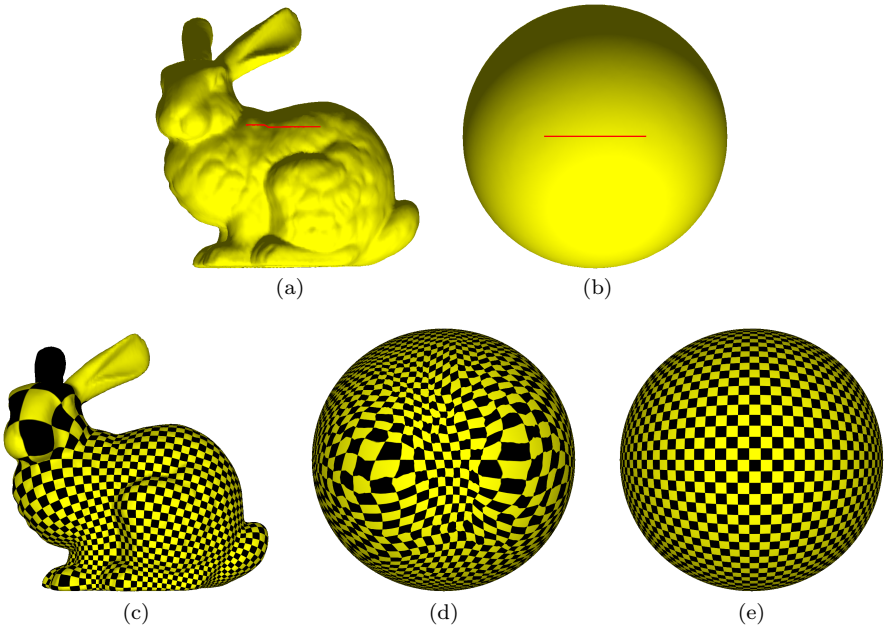


FIG. 5.10. *Computing a map with a minimized L^2-norm of μ from the Stanford Bunny onto a sphere, with a single curve constraint. (a) shows the Stanford Bunny with a curve marked. (b) shows a sphere with the target curve constraint. (c) shows the Stanford Bunny textured according to the final map. (d) shows the initial map constructed by a modified Tutte embedding onto the sphere. (e) shows the final map after the Beltrami flow algorithm is used to minimize the Beltrami coefficient in the least square sense. All meshes are textured consistently with correspondence from a stereographic projection of the unit sphere in (e).*

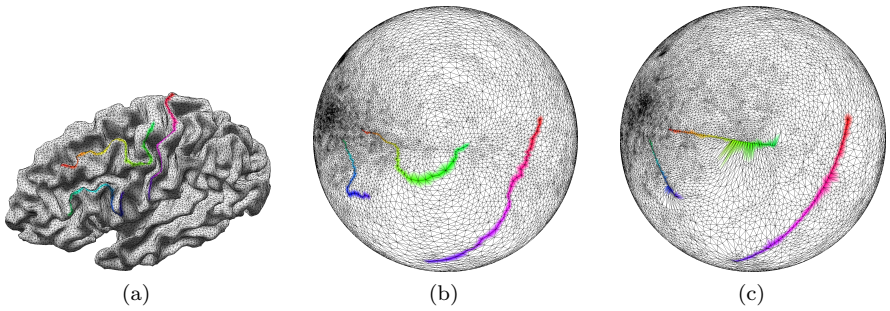
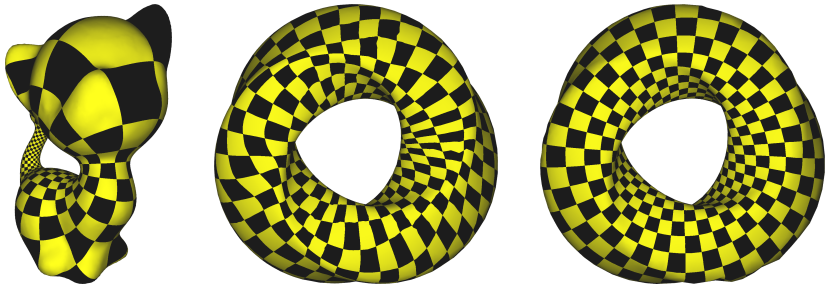


FIG. 5.11. *Computing a map with a minimized L^2-norm of μ from a cortical surface onto a sphere, with 3 curve constraints. (a) shows the cortical surface with 3 feature curves marked. (b) shows an initial map constructed by Tutte embedding. (c) shows the final map after the Beltrami flow algorithm is used to minimize the Beltrami coefficient in the least square sense. The feature curves are now mapped onto geodesics by enforcing the direction of their flows.*

vertex, then perform a linear search on the set of their neighboring triangles of the target surface to locate the triangle containing the source vertex. Then the location of the source vertex is represented by its barycentric coordinates in the triangle. The tangent and normal vectors at the mapped source vertex are then computed from the triangle. After one step of Beltrami flow, each mapped vertex of the source mesh has evolved to a new location in the tangent plane of the target mesh, we then project the



(a) (b) (c)

FIG. 5.12. *Computing a map minimizing the L^3-norm of the Beltrami differential between two triangulated surfaces: the kitten and a modified torus. (a) shows the kitten surface with a conformal texture. (b) shows the initial map of the kitten onto the modified torus. (c) shows the optimized map of the kitten onto the modified torus. Conformality is well restored by the discrete Beltrami flow algorithm.*

new location to the original target mesh. This can be done using the same procedure as before by projecting all the neighboring triangles onto the tangent plane and find the triangle which contains the new point.

We illustrate our algorithms with a few examples. First we consider mapping the kitten mesh, which is a genus one triangular mesh, onto a modified torus parametrized by $(x, y, z) = \phi_1(u, v)$ for $(u, v) \in [0, 1]^2$, where

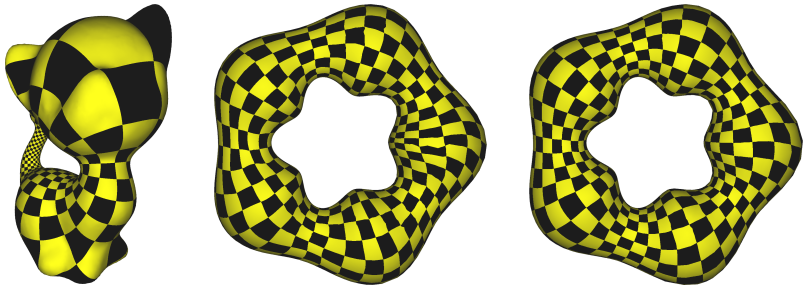
$$\begin{aligned} x &= \cos 2\pi u \cdot [r_1 + r_2 \cos 2\pi v + r_3 \cos 6\pi(u + v)] \\ y &= \sin 2\pi u \cdot [r_1 + r_2 \cos 2\pi v + r_3 \cos 6\pi(u + v)] \\ z &= r_2 \sin 2\pi v, \end{aligned}$$

where we have chosen $r_1 = 4$, $r_2 = 1.7$ and $r_3 = 0.3$. To show the flexibility of our method, we use Beltrami flow to minimize the L^3-norm (instead of L^2-norm) of the Beltrami differential of this mapping. Initially, the kitten surface is cut open to form a simply connected open surface and mapped onto the parametrization domain $[0, 1]^2$. Using a kd-tree, we compute the barycentric coordinates for each mapped vertex of the kitten surface into some triangle of the modified torus . The discrete Beltrami flow is carried out and the new location of each mapped source vertex after adjustment is projected onto the modified torus using the procedure we described above until numerical convergence is achieved. Both initial and final mapping between the two triangulated meshes are illustrated in Figure 5.12.

For the second example, we compute the mapping optimizing the L^4-norm of the Beltrami differential of the mapping from the kitten surface onto another modified torus parametrized by $(x, y, z) = \phi_2(u, v)$ for $(u, v) \in [0, 1]^2$, where

$$\begin{aligned} x &= \cos 2\pi u \cdot [r_1 + \cos 2\pi v \cdot (r_2 + r_3 \cos 10\pi u)] \\ y &= \sin 2\pi u \cdot [r_1 + \cos 2\pi v \cdot (r_2 + r_3 \cos 10\pi u)] \\ z &= \sin 2\pi v \cdot (r_2 + r_3 \cos 10\pi u), \end{aligned}$$

where we have chosen $r_1 = 4$, $r_2 = 1.5$ and $r_3 = 0.4$. We apply Beltrami flow to minimize the L^4-norm of the Beltrami differential of this mapping. Using the same algorithm, the discrete Beltrami flow successfully converges to the final mapping numerically. Both initial and final mapping between the two triangulated meshes are



(a) (b) (c)

FIG. 5.13. *Computing a map minimizing the L^4-norm of the Beltrami differential between two triangulated surfaces: the kitten and a modified torus. (a) shows the kitten surface with a conformal texture. (b) shows the initial map of the kitten onto the modified torus. (c) shows the optimized map of the kitten onto the modified torus. Conformality is well restored by the discrete Beltrami flow algorithm.*

Perturbation formula			
# of vertices	# of iterations	Time (seconds)	Error (L^2-norm)
33^2	26	23.8087	3.0506×10^{-3}
65^2	14	182.7729	1.0382×10^{-2}
129^2	12	2351.7488	5.4955×10^{-3}
257^2	Impractical to compute		
Beltrami flow			
# of vertices	# of iterations	Time (seconds)	Error (L^2-norm)
33^2	9	0.459429	8.6697×10^{-10}
65^2	7	1.1837	2.5620×10^{-8}
129^2	7	4.0143	1.3582×10^{-6}
257^2	7	19.1423	6.5365×10^{-7}

TABLE 5.3

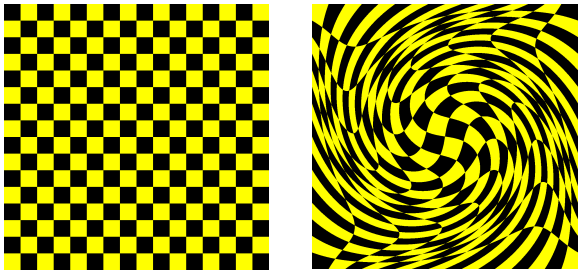
Comparison for the reconstruction of a diffeomorphism of the square $[-1,1]^2$ using the method proposed in [13] and our method, both with convergence tolerance of 10^{-3}.

illustrated in Figure 5.13. In both examples, the distortions in the initial mappings are minimized by our algorithm although different energy functional is used. This shows that our algorithm is able to work for general meshes, and the user is flexible to choose different energy to minimize to achieve desirable results.

5.5. Efficiency and Comparison with the Perturbation Approach. Our algorithm is also very efficient because the sparse linear system obtained from the least square problem can be solved easily. The discrete Beltrami flows on triangular meshes with 100K faces can be easily computed in less than 3 seconds per step. Further improvement can be achieved by using more efficient solvers for least square problems.

We also compare our algorithm with another approach of computing the flow using the purturbation formula (3.6). The method was proposed in [13] and applied to surface registration and compression of surface diffeomorphisms in [12] and [11]. The limitation of the algorithm is that the target domain has to be globally parametrized and simply connected, such as the plane and the sphere. We apply both algorithms to the reconstruction of a highly distorted diffeomorphism on a square $[-1,1]^2$, shown in Figure 5.14. The results of both algorithms are shown in Table 5.3.

As we can see, our proposed algorithm is much more efficient than the algorithm



(a) (b)

FIG. 5.14. *A highly distorted diffeomorphism of a square. (a) shows a square textured with a checkerboard pattern. (b) shows how the checkerboard pattern is mapped by the diffeomorphism onto the target square. This highly distorted diffeomorphism is used to compare the performance of our algorithm with the algorithm proposed in [13].*

using the integration formula, both in terms of the computational speed, number of iterations required, and the accuracy of the converged mapping. Moreover, our algorithm also works on general surfaces with arbitrary topology. This demonstrates several advantages of our algorithm for surface mapping.

6. Conclusion. In this paper, we propose a computational framework for intrinsic Beltrami flow that can effectively adjust the Beltrami coefficient of diffeomorphisms between arbitrary Riemann surfaces without the need of a global parametrization. By using a least square approach, we developed a straight forward algorithm that can be implemented easily and efficiently. We demonstrate the accuracy, robustness and flexibility of our method using extensive tests.

Appendix. We illustrate the detailed computation involved to compute the least square energy for discrete Beltrami flow on both a plane and in arbitrary Riemann surfaces with or without boundaries.

Computing the Least Square Beltrami Flow Energy on a Plane. Given a triangular mesh on the plane, we compute its Beltrami flow energy function and express it as a quadratic function of the coordinates. Assuming a single patch, we want to minimize

$$E(\mathbf{u},\mathbf{v}) = \sum_{T_i \in \mathcal{T}} \mathcal{A}_{T_i} \cdot ((u_x - v_y - 2a_i)^2 + (v_x + u_y - 2b_i)^2), \tag{6.1}$$

where $\mathbf{u}$ and $\mathbf{v}$ are vectors representing the real and imaginary parts of the flow at each vertex, $\mathcal{A}_{T_i}$ is the area of triangle T_i, and the derivatives u_x, u_y, v_x and v_y are taken on T_i. The first step in the discrete algorithm is to computed u_x, u_y, v_x and v_y on each face from the position and the images of the vertices. Let $T_i = [p_{i1}, p_{i2}, p_{i3}]$, and $u_{ij} = u(p_{ij})$ for $j = 1, 2, 3$, we may express u_x and u_y on T_i in terms of u_{i1}, u_{i2} and u_{i3} as

$$\begin{pmatrix} u_x \\ u_y \end{pmatrix} = N_i \begin{pmatrix} u_{i1} \\ u_{i2} \\ u_{i3} \end{pmatrix}, \tag{6.2}$$

where N_i is a 3 by 1 matrix. Similarly, let $v_{ij} = v(p_{ij})$ for $j = 1, 2, 3$, we have

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = N_i \begin{pmatrix} v_{i1} \\ v_{i2} \\ v_{i3} \end{pmatrix}. \tag{6.3}$$

Let $p_{i1} = (x_{i1}, y_{i1})$, $p_{i2} = (x_{i2}, y_{i2})$, $p_{i3} = (x_{i3}, y_{i3})$. It is easy to see that $u_{i2} - u_{i1} = (x_{i2} - x_{i1})u_x + (y_{i2} - y_{i1})u_y$ and $u_{i3} - u_{i2} = (x_{i3} - x_{i2})u_x + (y_{i3} - y_{i2})u_y$. Thus we have the system

$$\begin{pmatrix} x_{i2} - x_{i1} & y_{i2} - y_{i1} \\ x_{i3} - x_{i1} & y_{i2} - y_{i1} \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} = \begin{pmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_{i1} \\ u_{i2} \\ u_{i3} \end{pmatrix}. \tag{6.4}$$

Therefore N_i is given by

$$N_i = \begin{pmatrix} x_{i2} - x_{i1} & y_{i2} - y_{i1} \\ x_{i3} - x_{i1} & y_{i2} - y_{i1} \end{pmatrix}^{-1} \begin{pmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}. \tag{6.5}$$

Write N_i as $\begin{pmatrix} N_{ix} \\ N_{iy} \end{pmatrix}$. Then (6.1) can be written as

$$E(\mathbf{u}, \mathbf{v}) = \sum_{T_i \in \mathcal{T}} \mathcal{A}_{T_i} \cdot ((N_{ix}\mathbf{u}_i - N_{iy}\mathbf{v}_i - 2a_i)^2 + (N_{ix}\mathbf{v}_i + N_{iy}\mathbf{u}_i - 2b_i)^2), \tag{6.6}$$

where $\mathbf{u}_i = \begin{pmatrix} u_{i1} \\ u_{i2} \\ u_{i3} \end{pmatrix}$ and $\mathbf{v}_i = \begin{pmatrix} v_{i1} \\ v_{i2} \\ v_{i3} \end{pmatrix}$. The gradient of the least square energy can then be taken to get a linear system for solving $\mathbf{u}$ and $\mathbf{v}$.

Computing the Least Square Beltrami Flow Energy on Arbitrary Riemann Surfaces. For the least square energy on arbitrary Riemann surfaces, the situation is similar, except that now $\mathbf{u}$ and $\mathbf{v}$ represents the direction of the flow as the coefficients in terms of the basis $\mathbf{u}_i$ and $\mathbf{v}_i$ of each tangent plane at vertex p_i. Mathematically, for each point p_i mapped to $\phi(p_i)$ on the target surface S, we can compute an orthonormal basis $\{\mathbf{n}_i, \mathbf{u}_i, \mathbf{v}_i\}$ at $\phi(p_i)$ such that $\mathbf{n}_i$ is the positive or outward normal. Since V_i is written in terms of $\mathbf{u}_i$ and $\mathbf{V}_i$ for each i but the least square energy is an integral computed on each face $\phi(T_i)$, one needs to express V in terms of the local isometric coordinates (r, s) on each $\phi(T_i)$ such that the integral can be computed. We discuss the detail of this procedure in this section.

Since the least square energy functional is a sum of the integrals on all faces, we may consider the case for a face T_i. Assume that we have mapped it onto $\mathbb{R}^2$ isometrically and let $\{\mathbf{r}_i, \mathbf{s}_i\}$ be a positively ordered orthonormal basis on $\psi_i(T_i)$. Then it can be seen that

$$r_{ij} = u_{ij}\mathbf{u}_i \cdot \mathbf{r}_i + v_{ij}\mathbf{v}_i \cdot \mathbf{r}_i \tag{6.7}$$

and

$$s_{ij} = u_{ij}\mathbf{u}_i \cdot \mathbf{s}_i + v_{ij}\mathbf{v}_i \cdot \mathbf{s}_i \tag{6.8}$$

for $j = 1, 2, 3$.

Again we can use N_i defined in the previous section to compute r_x, r_y, s_x and s_y. Therefore we have

$$\begin{aligned}&\iint_{\psi_i(T_i)} (r_x - s_y - 2a_i)^2 + (r_x + s_y - 2b_i)^2 \\ &\quad = \mathcal{A}_{\psi_i(T_i)} \cdot ((N_{ix}\mathbf{r}_i - N_{iy}\mathbf{s}_i - 2a_i)^2 + (N_{ix}\mathbf{s}_i + N_{iy}\mathbf{r}_i - 2b_i)^2). \qquad (6.9)\end{aligned}$$

Summing up the above expression for each face $\psi_i(T_i)$, we obtain the required least square energy function quadratic in terms of coordinates of $\phi(p_{i1})$, $\phi(p_{i2})$ and $\phi(p_{i3})$.

REFERENCES

[1] Yonathan Aflalo, Ron Kimmel, and Michael Zibulevsky, *Conformal mapping with as uniform as possible conformal factor*, SIAM J. Imaging Sciences, (2013), pp. 78–101.

[2] Lars V. Ahlfors, *Lectures on Quasiconformal Mapping*, vol. 38 of University Lecture Series, American Mathematical Society, 2006.

[3] Charles R. Collins and Kenneth Stephenson, *A circle packing algorithm*, Comput. Geom. Theory Appl., 25 (2003), pp. 233–256.

[4] Prabir Daripa, *A fast algorithm to solve the beltrami equation with applications to quasiconformal mappings.*, Journal of Computational Physics, 106 (1993), pp. 355–365.

[5] Prabir Daripa and Daoud Mashat, *An efficient and novel numerical method for quasiconformal mappings of doubly connected domains.*, Numer. Algorithms, 18 (1998), pp. 159–175.

[6] Frederick P. Gardiner and Nikola Lakic, *Quasiconformal Teichmller Theory*, vol. 76 of Mathematical Surveys and Monographs, American Mathematical Society, 1999.

[7] Zheng-Xu He, *Solving beltrami equations by circle packing.*, Trans. Am. Math. Soc., 322 (1990), pp. 657–670.

[8] Vladimir Kim, Yaron Lipman, and Thomas Funkhouser, *Blended intrinsic maps*, ACM Transactions on Graphics (Proc. SIGGRAPH), 30 (2011).

[9] Yaron Lipman, Vladimir G. Kim, and Thomas A. Funkhouser, *Simple formulas for quasiconformal plane deformations.*, ACM Trans. Graph., 28 (2009).

[10] Lok Ming Lui, Yalin Wang, Tony F. Chan, and Paul Thompson, *Landmark constrained genus zero surface conformal mapping and its application to brain mapping research*, Appl. Numer. Math., 57 (2007), pp. 847–858.

[11] Lok Ming Lui, Tsz Wai Wong, Paul M. Thompson, Tony F. Chan, Xianfeng Gu, and Shing-Tung Yau, *Compression of surface registrations using beltrami coefficients*, in The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010, IEEE, 2010, pp. 2839–2846.

[12] ———, *Shape-based diffeomorphic registration on hippocampal surfaces using beltrami holomorphic flow*, in Medical Image Computing and Computer-Assisted Intervention - MICCAI 2010, 13th International Conference, Beijing, China, September 20-24, 2010, Proceedings, Part II, Tianzi Jiang, Nassir Navab, Josien P. W. Pluim, and Max A. Viergever, eds., vol. 6362 of Lecture Notes in Computer Science, Springer, 2010, pp. 323–330.

[13] Lok Ming Lui, Tsz Wai Wong, Wei Zeng, Xianfeng Gu, Paul M. Thompson, Tony F. Chan, and Shing-Tung Yau, *Optimization of surface registrations using beltrami holomorphic flow*, J. Sci. Comput., 50 (2012), pp. 557–585.

[14] Bruno Lvy, Sylvain Petitjean, Nicolas Ray, and Jrme Maillot, *Least squares conformal maps for automatic texture atlas generation*, in SIGGRAPH 2002, 2002, pp. 362–371.

[15] C. Wayne Mastin and Joe F. Thompson, *Discrete quasiconformal mappings*, Zeitschrift Fur Angewandte Mathematik Und Physik, 29 (1978), pp. 1–11.

[16] W.P. Thurston, *The geometry and topology of three-manifolds*, Princeton University, 1988.

[17] W. T. Tutte, *How to draw a graph*, Proc Lond Math Soc, 13 (1963), pp. 743–767.

[18] T.W. Wong, X.F. Gu, T.F. Chan, and L.M. Lui, *Parallelizable inpainting and refinement of diffeomorphisms using beltrami holomorphic flow*, in ICCV11, 2011, pp. 2383–2390.

[19] Tsz Wai Wong and Hongkai Zhao, *Computing surface uniformization using discrete beltrami flow*, to be appeared in SIAM Journal of Scientific Computing, (2014).

[20] Xiaotian Yin, Miao Jin, Feng Luo, and Xianfeng David Gu, *Emerging trends in visual computing*, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 38–74.

[21] Wei Zeng, Feng Luo 0002, Shing-Tung Yau, and Xianfeng David Gu, *Surface quasiconformal mapping by solving beltrami equations*, in Mathematics of Surfaces XIII, 13th IMA International Conference, York, UK, September 7-9, 2009, Proceedings, Edwin R.

Hancock, Ralph R. Martin, and Malcolm A. Sabin, eds., vol. 5654 of Lecture Notes in Computer Science, Springer, 2009, pp. 391–408.

[22] Wei Zeng, Lok Ming Lui, Feng Luo, Tony Fan-Cheong Chan, Shing-Tung Yau, and David Xianfeng Gu, *Computing quasiconformal maps using an auxiliary metric and discrete curvature flow*, Numer. Math., 121 (2012), pp. 671–703.