FAST AND ROBUST ALGORITHMS FOR HARMONIC ENERGY MINIMIZATION ON GENUS-0 SURFACES

RONGJIE LAI*, ZAIWEN WEN†, WOTAO YIN‡, XIANFENG GU§, AND LOK MING LUI¶

Abstract. Surface harmonic map between genus-0 surfaces plays an important role in applied mathematics and engineering, with applications in medical imaging and computer graphics. Previous work [1] introduces a variational approach for computing surface harmonic maps. It obtains global conformal parameterizations of genus-0 surfaces through minimizing the harmonic energy, with two weaknesses: its gradient descent iteration is slow, and its solutions contain undesired parameterization foldings when the underlying surface has long sharp features. This paper addresses these weaknesses by proposing an algorithm that significantly accelerates the harmonic map computation and a method that iteratively removes foldings. They are achieved, respectively, by applying recent results of optimization on manifolds and taking advantages of the weighted Laplace-Beltrami eigen-projection. Experimental results show that the proposed approaches compute genus-0 surface harmonic maps much faster than the existing algorithm in [1] and the new results contain no foldings.

Key words. Harmonic Energy, Conformal Map, Optimization with Orthogonality Constraints, Weighted Laplace-Beltrami Eigenfunction.

AMS subject classifications. 15A15, 15A09, 15A23

1. Introduction.

Surface parameterization is the process of mapping a surface onto a simple domain, such as a unit sphere or 2D rectangle. It allows operations on the surface to be carried out on the simple parameter domain. A special type of parameterization is called the conformal parameterization. Under the conformal parameterization, angles and thus local geometry are well preserved. Surface conformal parameterization has been widely used in different areas such as in medical imaging and computer graphics. For example, in medical imaging, human brains are often conformally mapped to a unit sphere. Computations and analysis of the brain surface can then be carried on the simple sphere, rather than on the complicated brain cortical surface [2, 3]. In computer graphics, surface conformal parameterization is applied for texture mapping and solving PDEs on surfaces [4].

Several conformal parameterization algorithms have been proposed by different research groups. Levy et al. [5] compute a conformal parameterization of topological disks by approximating the Cauchy-Riemann equation using the least squares method. Eck et al. [6] introduce the discrete harmonic map, which approximates the continuous harmonic map [7] by minimizing a metric dispersion criterion. In [8], Desbrun et al use conformal maps to define geometry maps, where they compute the conformal maps from a topological disk to the complex plane. Kanai et al. use a harmonic map for geometric metamorphosis in [9]. These works mainly deal with the local conformal parameterizations of surface patches, which are homeomorphic to the topological disk.

In many situations, a *global* conformal parameterization that maps a surface onto one global parameter domain is desirable. The global nature avoids the needs of introducing cuts on the surface and partitioning the surface into several patches. Computations can then be performed on one simple parameter domain. Spherical conformal parameterization has been widely used in many different areas such as human brain mapping, computer vision and computer graphics, just to name a few. Several global conformal parameteri-

*Department of Mathematics, University of Southern California, Los Angeles, USA. (rongjiel@usc.edu).

†Department of Mathematics and Institute of Natural Sciences, Shanghai Jiaotong University, Shanghai, China. (zw2109@sjtu.edu.cn).

‡Department of Computational and Applied Mathematics, Rice University, Houston, USA. (wotao.yin@rice.edu).

§Department of Computer Science, State University of New York at Stony Brook, New York, USA (gu@cs.sunysb.edu).

¶Department of Mathematics, The Chinese University of Hong Kong, Hong Kong, (lmlui@math.cuhk.edu.hk).

zation methods have been proposed. Hurdal et al. propose a circle packing approach for spherical conformal parameterization [10]. This method is based on the mean value property of the harmonic map and does not consider the specific metric. Haker et al. [11] use a method to compute a global conformal mapping from a genus-0 surface to a sphere by representing the Laplace-Beltrami operator as a linear system. In their method, a face on the mesh is punctured to change the topology of the surface and is then mapped to the complex plane. Stereographic projection is involved to map the surface onto the sphere. This leads to algorithm instability. More recently, Gu and Yau in [1, 2] introduce a nonlinear optimization method to compute global conformal parameterizations for genus-0 surfaces. The optimization is done in the tangent spaces of the sphere using gradient descent. This method avoids the stereo-graphic projection and is more stable and accurate. Later, Lui et al. [3] propose an algorithm to extend this algorithm to compute the optimized spherical conformal parameterization with landmark-matching. A shape-based landmark-matching optimized conformal parameterized is further proposed in [12] to drive a landmark curve into shape correspondence. As for the global conformal parameterization of higher-genus surfaces, Gu et al. [13, 14] propose to compute the conformal parameterization using the holomorphic 1-form, which is based on techniques in algebraic geometry. Curvature flow methods for conformal parameterization of high-genus surfaces, which deform the background Riemannian metric into a uniformization metric, are proposed by Gu et al. in [15, 16].

This paper aims at efficiently computing global spherical conformal parameterizations of genus-0 closed surfaces. We address two weaknesses of the algorithm in [1, 2]. First, it often converges slowly. As an example, the conformal parameterization of a surface mesh with 16k vertices takes more than 20 minutes to compute. This hinders the application of this algorithm. Second, undesired foldings occur in the parameterization of long sharp features of the surface. This leads to inaccurate global parameterizations. The robustness of the algorithm cannot be guaranteed especially for surfaces with complicated geometries. To address these issues, we propose in this paper new methods with improved efficiency and robustness for surfaces with long sharp features, in the following steps. The global harmonic energy minimization of genus-0 surfaces is formulated as a minimization problem with spherical constraints. Then, a fast algorithm for the optimization on genus-0 manifolds is proposed by applying optimization techniques for orthogonality constraints [17]. To overcome the folding issues, the weighted Laplace-Beltrami eigen-projection is exploited, and a conformal correction method which fixes foldings is developed. Experimental results show that the proposed algorithm can compute genus-0 surface harmonic maps much faster than the algorithm in [1]. The results of the proposed algorithm are free of foldings.

In summary, the contributions of this paper include a faster algorithm for genus-0 surfaces' global spherical conformal parameterization and a novel method for avoiding the parameterization folding.

2. Mathematical Background. In this section, we give a brief review of the conformal parameterization and harmonic map. For details, we refer the reader to [18, 7, 19].

All Riemann surfaces are locally Euclidean. Given two Riemann surfaces $\mathcal{M}$ and $\mathcal{N}$, we can represent them locally as $\phi_{\mathcal{M}}(x^1, x^2) : \mathbb{R}^2 \to \mathcal{M} \subseteq \mathbb{R}^3$ and $\phi_{\mathcal{N}}(y^1, y^2) : \mathbb{R}^2 \to \mathcal{N} \subseteq \mathbb{R}^3$, where (x^1, x^2) and (y^1, y^2) are their coordinates, respectively. The inner product of the tangent vectors at each point of the surface can be represented by its first fundamental form. The first fundamental form on $\mathcal{M}$ can be written as $ds^2_{\mathcal{M}} = \sum_{i,j} g_{ij} dx^i dx^j$, where $g_{ij} = \frac{\partial \phi_{\mathcal{M}}}{\partial x^i} \cdot \frac{\partial \phi_{\mathcal{M}}}{\partial x^j}$ and $i, j = 1, 2$. Similarly, the first fundamental form on $\mathcal{N}$ can be written as $ds^2_{\mathcal{N}} = \sum_{i,j} \widetilde{g}_{ij} dy^i dy^j$ where $\widetilde{g}_{ij} = \frac{\partial \phi_{\mathcal{N}}}{\partial y^i} \cdot \frac{\partial \phi_{\mathcal{N}}}{\partial y^j}$ and $i, j = 1, 2$. Given a map $f : \mathcal{M} \to \mathcal{N}$ between the $\mathcal{M}$ and $\mathcal{N}$, with the local parameterization, f can be represented locally by its coordinates as $f : \mathbb{R}^2 \to \mathbb{R}^2$, $f(x^1, x^2) = (f_1(x^1, x^2), f_2(x^1, x^2))$. Every tangent vector $\vec{v}$ on $\mathcal{M}$ can be mapped (*pushed forward*) by f to a tangent vectors $f_*(\vec{v})$ on $\mathcal{N}$. The inner product of the vectors $f_*(\vec{v_1})$

and $f_*(\vec{v_2})$), where $\vec{v_1} = v_1^1\partial_{x_1} + v_1^2\partial_{x_2}$ and $\vec{v_2} = v_2^1\partial_{x_1} + v_2^2\partial_{x_2}$ are tangent vectors on $\mathcal{M}$, is:

$$\begin{aligned} f^*(ds^2_{\mathcal{N}})(\vec{v_1},\vec{v_2}) :=\ & \langle f_*(\vec{v_1}), f_*(\vec{v_2})\rangle_{\widetilde{g}} \\ &= \langle \sum_{i,m=1}^{2} v_1^i \frac{\partial f_m}{\partial x^i}\partial_{y_m}, \sum_{j,n=1}^{2} v_2^j \frac{\partial f_n}{\partial x^j}\partial_{y_n}\rangle_{\widetilde{g}} \\ &= \sum_{i,j}\left((\sum_{m,n} \widetilde{g}_{mn}\frac{\partial f_m}{\partial x^i}\frac{\partial f_n}{\partial x^j}) v_1^i v_2^j \right). \end{aligned} \tag{2.1}$$

Therefore, a new Riemannian metric $f^*(ds^2_{\mathcal{N}})$ on M is induced by f and $ds^2_{\mathcal{N}}$, called the *pull back metric*. We say that the map f is **conformal** if

$$f^*(ds^2_{\mathcal{N}}) = e^{2u} ds^2_{\mathcal{M}} \tag{2.2}$$

with a smooth function $u : \mathcal{M} \to \mathbb{R}$ on $\mathcal{M}$.

A parameterization $\varphi : R^2 \to \mathcal{M}$ is a **conformal parameterization** if φ is a conformal map. Intuitively, a map is conformal if it preserves the inner product of the tangent vectors up to a scaling factor, called the conformal factor e^{2u}. An immediate consequence is that every conformal map preserves angles.

For genus-0 surfaces, conformal maps are closely related to harmonic maps. Suppose now $(\mathcal{M}, g)$ and $(\mathcal{N}, \widetilde{g})$ are two genus-0 Riemannian surfaces with a diffeomorphism f. We define its energy density as:

$$e(f) = \|\mathrm{d}f\|^2 = \sum_{i,j=1,2} g^{ij}\langle f_*\partial_{x^i}, f_*\partial_{x^j}\rangle_{\widetilde{g}} \tag{2.3}$$

where (g^{ij}) is the inverse of (g_{ij}) and $f_*\partial_{x^i}$ is the push-forward map. The Harmonic energy of the map f is defined by

$$\mathcal{E}(f) = \frac{1}{2}\int_{\mathcal{M}} e(f)\mathrm{d}\mathcal{M}. \tag{2.4}$$

$\mathcal{E}$ defined an energy function on the set of all diffeomorphisms from $(\mathcal{M}, g)$ to $(\mathcal{N}, \widetilde{g})$. f is called a **harmonic map** if it is a critical point of the energy functional $\mathcal{E}$.

In practice, we consider (M, g) and $(\mathcal{N}, \widetilde{g})$ as embedding surfaces in $\mathbb{R}^3$ with the induced metric. In other words, we have:

$$\begin{aligned} f : \mathcal{M} &\longrightarrow \mathcal{N} \hookrightarrow \mathbb{R}^3 \\ x &\longmapsto y \longmapsto \vec{F}(x) = (f_1(x), f_2(x), f_3(x)). \end{aligned} \tag{2.5}$$

Since $(\mathcal{N}, \widetilde{g})$ is with the induced metric from $\mathbb{R}^3$, we have:

$$\langle f_*\partial_{x^i}, f_*\partial_{x^j}\rangle_{\widetilde{g}} = \langle \frac{\partial \vec{F}}{\partial x^i}, \frac{\partial \vec{F}}{\partial x^j}\rangle = \sum_{\alpha=1}^{3} \frac{\partial f_\alpha}{\partial x^i}\frac{\partial f_\alpha}{\partial x^j}, \qquad i,j = 1,2. \tag{2.6}$$

On the other hand, for C^1 function f_1, f_2, f_3 on $\mathcal{M}$, we have

$$\nabla_{\mathcal{M}} f_\alpha = \sum_{i,j=1}^{2} g^{ij} \frac{\partial f_\alpha}{\partial x^i} \partial_{x^j} \tag{2.7}$$

$$\|\nabla_{\mathcal{M}} f_\alpha\|^2 = \langle \nabla_{\mathcal{M}} f_\alpha, \nabla_{\mathcal{M}} f_\alpha \rangle_g = \sum_{i,j=1,2} g^{ij} \frac{\partial f_\alpha}{\partial x^i} \frac{\partial f_\alpha}{\partial x^j}, \quad \alpha = 1, 2, 3. \tag{2.8}$$

For a map $f : \mathcal{M} \to \mathcal{N}$ between two genus-0 surfaces $\mathcal{M}$ and $\mathcal{N}$, f is conformal if and only if it is a harmonic map [18]. Therefore, computing a conformal map between two genus-0 surfaces is equivalent to computing a harmonic map between them, which can be obtained by finding a critical point of the energy functional $\mathcal{E}$ in (2.4).

3. Optimization over Homeomorphisms between Genus-0 Surfaces. We focus on computing a harmonic map from a given genus-0 surface to the unit sphere. In other words, we consider $(\mathcal{N}, \widetilde{g})$ as the unit sphere (S^2, g_0) in $\mathbb{R}^3$ with induced metric g_0. By combining (2.3), (2.4), (2.6) and (2.8), computing the harmonic map between $\mathcal{M}$ and S^2 is equivalent to solving the following optimization problem with spherical constraints:

$$\begin{aligned} \min_{\vec{F}=(f_1,f_2,f_3)} \quad & \mathcal{E}(\vec{F}) = \frac{1}{2} \int_{\mathcal{M}} \|\nabla_{\mathcal{M}} f_1\|^2 + \|\nabla_{\mathcal{M}} f_2\|^2 + \|\nabla_{\mathcal{M}} f_3\|^2 \mathrm{d}\mathcal{M} \\ s.t. \quad & \|\vec{F}(x)\| = \sqrt{f_1^2 + f_2^2 + f_3^2} = 1, \quad \forall x \in \mathcal{M}. \end{aligned} \tag{3.1}$$

Hence a conformal map $\vec{F}$ from $\mathcal{M}$ to (S^2, g_0) can be found as the solution to the optimization problem (3.1). Due to its non-convexity, problem (3.1) can have multiple local minimizers. This coincides with the fact that conformal maps from surface $\mathcal{M}$ to the unit sphere are non-unique. However, any two different conformal maps of $\mathcal{M}$ only differ by a Mobius transformation of the unit sphere. In addition, the Harmonic energies of all conformal maps are identical [19]. Correspondingly, all local minimizers of problem (3.1) have the same Harmonic energy, and any one of them gives a conformal map from $\mathcal{M}$ to the unit sphere.

Unlike optimization in $\mathbb{R}^n$ where it is straightforward to decrease the objective along a straight search line (e.g., along the negative gradient direction), it is not as easy to do so in a curved manifold. A natural choice is the geodesic, which is the analog of straight line and has the shortest length between two different points. Another choice is iterative projection: descent along straight lines and project points back to the manifold. There are various optimization methods for optimization on manifold such as [20, 21, 22] and references therein, which are mostly based on either geodesics or projections. Considering the fact that at each $x \in \mathcal{M}$, $\|\vec{F}(x)\| = 1$ defines a unit sphere, we choose to develop a sphere-geodesic descent method based on the recent work [17]. It is numerically efficient and lets us apply state-of-the-art acceleration techniques such as Barzilai-Borwein steps and non-monotone line search with global convergence guarantees.

3.1. Constraint Preserving Update. The Lagrangian of problem (3.1) is

$$\mathcal{L}(\vec{F}, \lambda) := \mathcal{E}(\vec{F}) - \frac{1}{2} \int_{\mathcal{M}} \lambda(x) \left(\|\vec{F}(x)\|^2 - 1 \right) \mathrm{d}\mathcal{M},$$

where λ is the Lagrange multiplier. The first-order optimality conditions of (3.1) are (assuming they are well-defined)

$$\operatorname{grad}_{\vec{F}} \mathcal{L}(\vec{F}, \lambda) := H - \lambda \vec{F} = 0, \tag{3.2}$$

$$\|\vec{F}(x)\| = 1, \quad \forall x \in \mathcal{M}, \tag{3.3}$$

where $H = \operatorname{grad} \mathcal{E}(\vec{F}) = -(\triangle_{\mathcal{M}} f_1, \triangle_{\mathcal{M}} f_2, \triangle_{\mathcal{M}} f_3)$ is the Fréchet derivative of $\mathcal{E}(\vec{F})$ with respect to $\vec{F}$. For concise notation, we let A^*F denote the function $(A^*F)(x) := \langle A(x), \vec{F}(x) \rangle$. Applying the linear operator $\vec{F}^*$ to both sides of (3.2) and using the fact $(\vec{F}^* \vec{F})(x) = \|\vec{F}(x)\| = 1$, $\forall x \in \mathcal{M}$, we obtain $\lambda = \vec{F}^* H = H^* \vec{F}$. Plugging λ back to (3.2) gives $0 = H - (\vec{F}^* H)\vec{F} = H(\vec{F}^* \vec{F}) - \vec{F}(H^* \vec{F})$ or, equivalently

$$A\vec{F} = 0 \text{ with } A := H\vec{F}^* - \vec{F}H^*.$$

By definition, $A(x)$ is skew-symmetric at every $x \in \mathcal{M}$. Following [17], we use A and its skew-symmetry to define a search path maintaining $\|\vec{F}\| = 1$.

Observe that $A(x)\vec{F}(x)$ is the gradient of $\mathcal{E}$ at x projected to S^2. In $\mathbb{R}^3$, the steepest descent path is $\vec{Y}(x) := \vec{F}(x) - \tau A(x)\vec{F}(x)$, where τ is a scalar representing the step size. However, this $\vec{Y}(x)$ does not generally have a unit norm. If we instead apply the implicit update

$$\vec{Y}(x) = \vec{F}(x) - \frac{\tau}{2} A(x)(\vec{F}(x) + \vec{Y}(x))$$

and obtain

$$\vec{Y}(x) = \left(I + \frac{\tau}{2} A(x)\right)^{-1} \left(I - \frac{\tau}{2} A(x)\right) \vec{F}(x), \tag{3.4}$$

then the fact that $\left(I + \frac{\tau}{2} A(x)\right)^{-1} \left(I - \frac{\tau}{2} A(x)\right)$ is orthogonal gives us $\|\vec{Y}(x)\| = \|\vec{F}(x)\| = 1$. Hence, we define the update path $\vec{Y}[\tau]$ by

$$\vec{Y}[\tau] := \vec{F} - \frac{\tau}{2} A(\vec{F} + \vec{Y}[\tau]), \tag{3.5}$$

which has the following theorem.

THEOREM 3.1. *For every τ, $Y[\tau]$ of (3.5) satisfies $\|\vec{Y}[\tau]\| = \|\vec{F}\|$ point-wise. In addition, it is given in the closed-form*

$$\vec{Y}[\tau] = \left(I + \frac{\tau}{2} A\right)^{-1} \left(I - \frac{\tau}{2} A\right) \vec{F},$$

which can be computed as

$$\vec{Y}[\tau] = \alpha[\tau]\vec{F} + \beta[\tau]H,$$

where

$$\alpha[\tau] = \frac{\left(1+\frac{\tau}{2}\vec{F}^*H\right)^2 - \left(\frac{\tau}{2}\right)^2\|\vec{F}\|^2\|H\|^2}{1-\left(\frac{\tau}{2}\right)^2(\vec{F}^*H)^2+\left(\frac{\tau}{2}\right)^2\|\vec{F}\|^2\|H\|^2},$$
$$\beta[\tau] = \frac{-\tau\|\vec{F}\|^2}{1-\left(\frac{\tau}{2}\right)^2(\vec{F}^*H)^2+\left(\frac{\tau}{2}\right)^2\|\vec{F}\|^2\|H\|^2}.$$

The result of theorem is visualized in Figure 3.1.

According to Theorem 3.1, the cost of computing $\vec{Y}[\tau]$ is dominated by the computation of $\|H\|^2$ and $\vec{F}^*H$. It is also worth noting that τ sometimes needs updates, which incurs updates to $\vec{Y}[\tau]$, but the cost is relatively small.

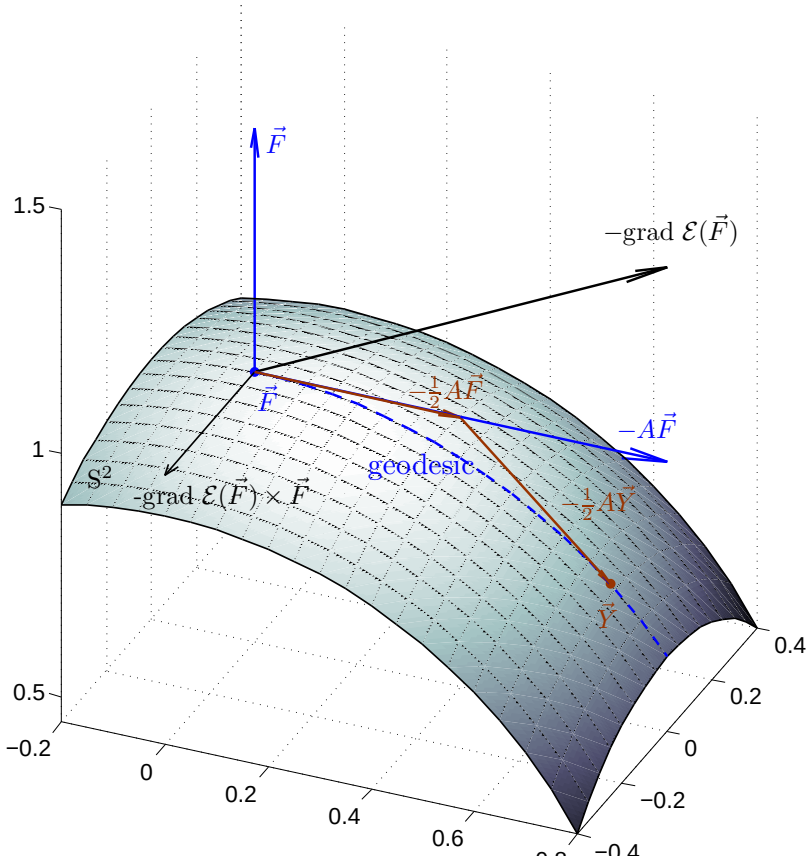


FIG. 3.1. *An illustration of constraints preserving update. Given x, let $S^2 = \{\vec{F}(x) \in \mathbb{R}^3 : \|\vec{F}(x)\|_2 = 1\}$. The point $\vec{F}(x) - \tau A(x)\vec{F}(x)$ is not feasible, but the point $\vec{Y}(x)$ given by (3.4) is feasible and on the geodesic along the negative projected gradient direction.*

3.2. Algorithm and Initial Map. To make the maximal use of the computed $\|H\|^2$ and $\vec{F}^*H$ at each iteration k, we determine a step size τ_k that makes significant descent while still guarantees the convergence of the overall iterations. To this end, instead of the classical Armijo-Wolfe based monotone line search, we apply nonmonotone curvilinear[1] search with an initial step size determined by the Barzilai-Borwein formula, which we have found more efficient for our problem. They were developed originally for $\mathbb{R}^n$ in [23] and [24],

[1] As our search path is a curve rather than a straight line

respectively. At iteration k, the step size is computed as

$$\tau_{k,1} = \frac{\int_{\mathcal{M}} \|D_{k-1}(x)\|^2 \mathrm{d}\mathcal{M}}{|\int_{\mathcal{M}} D^*_{k-1}(x) W_{k-1}(\mathrm{x}) \mathrm{d}\mathcal{M}|} \text{ or } \tau_{k,2} = \frac{|\int_{\mathcal{M}} D^*_{k-1}(x) W_{k-1}(x) \mathrm{d}\mathcal{M}|}{\int_{\mathcal{M}} \|W_{k-1}(x)\|^2 \mathrm{d}\mathcal{M}}, \tag{3.6}$$

where $D_{k-1} := \vec{F}_k - \vec{F}_{k-1}$ and $W_{k-1} = A_k \vec{F}_k - A_{k-1}\vec{F}_{k-1}$. The final value for τ_k is a fraction (up to 1, inclusive) of $\tau_{k,1}$ or $\tau_{k,2}$ determined by the nonmonotone search in Algorithm 1, Lines 3 and 5, which enforce a trend of descent in the objective value but do not require strict descent at each iteration. At the first iteration where $\vec{F}_{k-1}$ and A_{k-1} are not available, one can set a unit initial step size. The convergence of this algorithm can be obtained, and we refer the proof to [17].

Assembling the above parts, we arrive at Algorithm 1, in which ϵ is a stopping parameter, and ρ, δ, and ξ are curvilinear search parameters, which can be set to typical values as 10^{-4}, 0.1 and 0.85, respectively.

Algorithm 1: A Fast Algorithm

1 Given $\vec{F}_0$, pick $\rho,\ \delta,\ \xi,\ \epsilon \in (0,1)$. $k \leftarrow 0$.
2 **while** $\|\nabla \mathcal{E}(\vec{F}_k)\| > \epsilon$ **do**
3 Compute $\tau_k \leftarrow \tau_{k,1}\delta^h$ or $\tau_k \leftarrow \tau_{k,2}\delta^h$, where h is the smallest nonnegative integer satisfying $\mathcal{E}(\vec{Y}_k(\tau_k)) \le C_k + \rho\tau_k \mathcal{E}'(\vec{Y}_k(0))$.
4 $\vec{F}_{k+1} \leftarrow \vec{Y}_k(\tau_k)$.
5 $Q_{k+1} \leftarrow \xi Q_k + 1$ and $C_{k+1} \leftarrow \frac{\xi Q_k C_k + \mathcal{E}(\vec{F}_{k+1})}{Q_{k+1}}$.
6 $k \leftarrow k+1$.

Generally speaking, it is difficult to construct one-to-one and onto smooth maps from a given genus-0 surface to the unit sphere. In practice, we choose the Gauss map as the initial map $\vec{F}_0$, which is defined as follows:

DEFINITION 3.2 (Gauss map). *$\mathcal{G} : \mathcal{M} \to S^2, \mathcal{G}(p) = \vec{n}_p$, where $\vec{n}_p$ is the unit normal vector at $p \in \mathcal{M}$.*

3.3. Implementation and Simulations. In the implementation of Algorithm 1, we approximate $\mathcal{M}$ by a triangulated surface $\mathcal{M} = \{V = \{p_i\}_{i=1}^N, T = \{T_l\}_{l=1}^L\}$, where $p_i \in \mathbb{R}^3$ is the i-th vertex and T_l is the l-th triangle. For a function $h = (h(p_1), \cdots, h(p_N))$ defined on the triangle mesh, we approximate the Laplace-Beltrami operator and numerical integral on surface $\mathcal{M}$ by [25, 26, 27]:

$$\begin{aligned} \triangle_{\mathcal{M}} h(p_i) &\approx \frac{3}{\sum_{p_i \in T_l} \mathrm{Area}(T_l)} \sum_{j \in N_i} \omega_{ij}(p_i) \left(h(p_j) - h(p_i)\right), \\ \int_{\mathcal{M}} h \mathrm{d}_{\mathcal{M}}(x) &\approx \sum_{i=1}^N h(p_i) \cdot A_i \end{aligned} \tag{3.7}$$

where $\omega_{ij}(p_i) = \frac{cot\alpha_{ij}(p_i) + cot\beta_{ij}(p_i)}{2}$, α_{ij} and β_{ij} are the two angles opposite to the edge $\overline{p_i p_j}$, N_i is the first ring neighborhood of the vertex p_i, and $A_i = \frac{1}{3}\sum_{p_i \in T_l} \mathrm{Area}(T_l)$.

Moreover, given a surface map $\vec{F} : \mathcal{M} \to S^2, p \mapsto (f_1(p), f_2(p), f_3(p))$, the "conformal factor" with respect to the map $\vec{F}$ can be approximated according to the definition given in the section 2:

$$e^{2u(p_i)} = \frac{\sum_{p_i \in T_l} \mathrm{Area}(\vec{F}(T_l))}{\sum_{p_i \in T_l} \mathrm{Area}(T_l)} \tag{3.8}$$

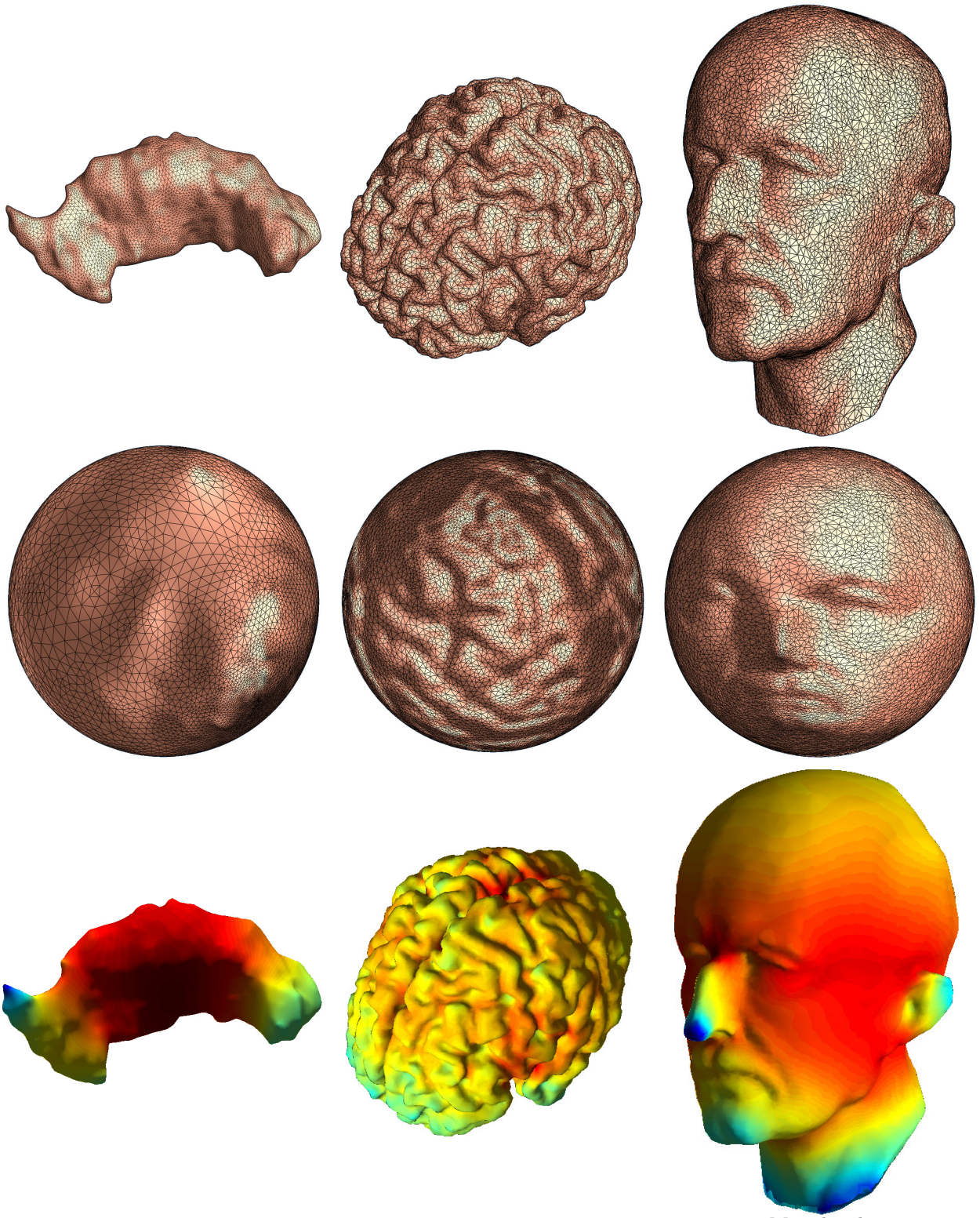


FIG. 3.2. *First row: three input surfaces; Second row: conformal maps obtained by Algorithm 1; Third row: surfaces color-coded by the corresponding u in the conformal factors.*

Table 3.1 and Figure 3.2 give the results of Algorithm 1 on three different examples: a Putamen surface, a brain surface and a Maxplanck surface. Their conformal maps are obtained by Algorithm 1 with a fixed $\epsilon = 10^{-10}$ and initialized by the Gauss map. The surface sizes, numbers of iterations and computation times are given in Table 3.1. The top row of Figure 3.3 shows how the energies decrease over the iterations. To illustrate the quality of the result maps, we compute the angle differences between triangles on the input surfaces and the corresponding triangles on the obtained maps. As shown in the histograms on the bottom row of Figure 3.3, most of the angle differences are close to zero, so the obtained maps do preserve angles and thus nearly satisfy the main property of the conformal map. In general, Algorithm 1 can efficiently compute conformal maps of a large class of surfaces as long as they do not contain extremely long and sharp patterns.

surface	# of vertices	# of iterations	time(s)	ϵ
Putamen	10000	1414	4.80	1e-10
Brain	15002	1329	17.80	1e-10
Maxplanck	12556	892	14.19	1e-10

TABLE 3.1
Iterations and computation times for surfaces in Figure.3.2

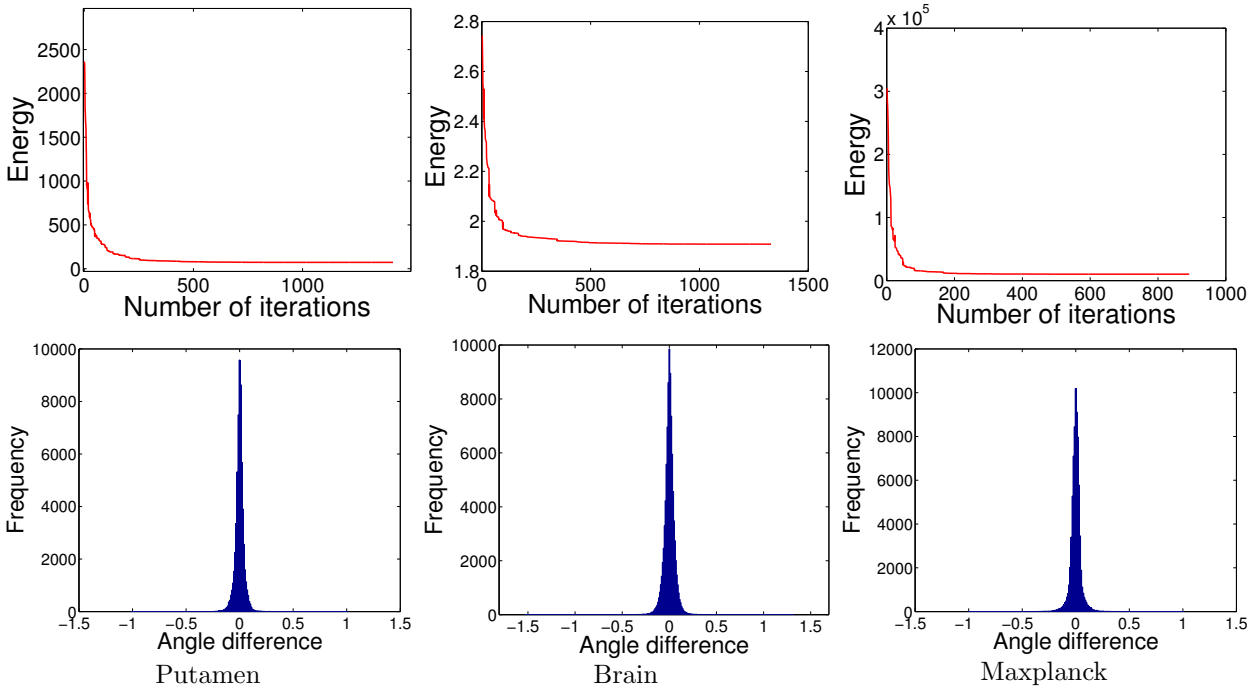


FIG. 3.3. *Top row: harmonic energy (3.1) vs. iteration number; Bottom row: angle difference histograms.*

3.4. Artificial Folding Issue. For arbitrary given surface $\mathcal{M}$, it is not always true that the Gauss map is a diffeomorphism from $\mathcal{M}$ to S^2. Therefore, the initial Gauss map can introduce artificial foldings for surfaces with complicated geometries, especially for those with long sharp features. The foldings often cause trouble for algorithms to return acceptable results. Such phenomenon has also been observed in the well-known algorithm [1, 2]. Generally speaking, it is challenging to construct a one-to-one and onto initial map from an arbitrary genus-0 surface to sphere.

Figure 3.4 depicts the artificial foldings introduced by the Gauss map near the long neck, tail and four legs of a dinosaur surface. To tackle this difficulty, we introduce a conformal correction method based on the weighted Laplace-Beltrami eigen-projection in the next section.

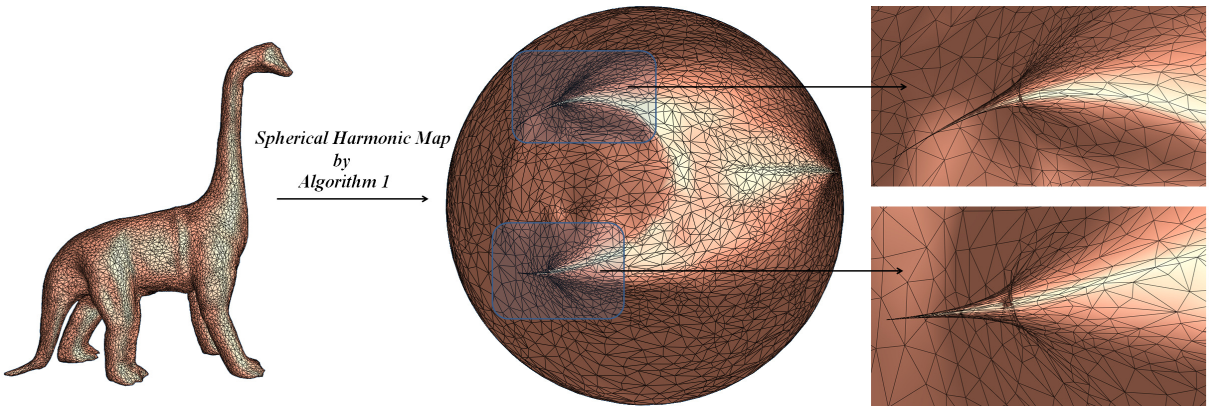


FIG. 3.4. *The artificial foldings in the dinosaur surface*

4. Conformal Correction by Weighted Laplace-Beltrami Eigen-projection. To remove the possible artificial foldings introduced by the Gauss map, we first introduce the concepts of star-shaped surface and Star map:

DEFINITION 4.1 (Star-shaped surface). *Let $\mathcal{M}$ be a closed surface embedded in $\mathbb{R}^3$ and $\mathcal{D} \subset \mathbb{R}^3$ be the domain bounded by $\mathcal{M}$. We call $\mathcal{M}$ a star-shaped surface if there exists a point $c \in \mathcal{D}$ such that for each point p in $\mathcal{D}$, the segment $\overline{cp}$ lies entirely in $\mathcal{D}$. The set of all such point c is called the kernel of $\mathcal{M}$. Each radial line initiated from a point in the kernel of $\mathcal{M}$ has only one intersection with $\mathcal{M}$.*

DEFINITION 4.2 (Star map). *Let $\vec{r}(p) = (r_1(p), r_2(p), r_3(p))$ be three coordinate functions of $\mathcal{M} \subset \mathbb{R}^3$. We write $\vec{c} = \frac{\int_{\mathcal{M}} \vec{r} \mathrm{d}\mathcal{M}}{\int_{\mathcal{M}} \mathrm{d}\mathcal{M}}$ as the center of $\mathcal{M}$ in $\mathbb{R}^3$. Then the Star map is defined as*

$$\mathfrak{S} : \mathcal{M} \to S^2, \mathfrak{S}(p) = \frac{\vec{c} - \vec{r}(p)}{\|\vec{c} - \vec{r}(p)\|}. \tag{4.1}$$

In particular, if $\mathcal{M}$ is a convex surface in $\mathbb{R}^3$, then $\mathcal{M}$ is also a star-shaped surface. Then the Star map defined as above is a one-to-one and onto map from $\mathcal{M}$ to S^2. In this case, we can also choose the Star map as an initial map for Algorithm 1. However, for general genus-0 surfaces, the Star map may not be able to construct a one-to-one and onto map to the unit sphere, so it may also introduce artificial foldings as the Gauss map does. Toward this challenge, we propose to use the weighted Laplace-Beltrami eigenfunctions to construct special Star maps and iteratively remove foldings.

Given a closed Riemannian surface $(\mathcal{M}, g)$, the Laplace-Beltrami (LB) operator is defined as

$$\triangle_g \phi = \frac{1}{\sqrt{G}} \sum_{i=1}^{2} \frac{\partial}{\partial x_i} (\sqrt{G} \sum_{j=1}^{2} g^{ij} \frac{\partial \phi}{\partial x_j}) \tag{4.2}$$

where (g^{ij}) is the inverse matrix of $g = (g_{ij})$ and $G = \det(g_{ij})$.

The LB operator is self adjoint and elliptic, so its spectrum is discrete. We let the eigenvalues of $\triangle_g$ be denoted as $0 = \lambda_0 < \lambda_1 < \lambda_2 < \cdots$ and the corresponding eigenfunctions as $\phi_0, \phi_1, \phi_2, \cdots$ such that

$$\triangle_g \phi_n = -\lambda_n \phi_n, \quad \int_{\mathcal{M}} \phi_n^2 \mathrm{d}\mathcal{M} = 1, \quad n = 0, 1, 2, \cdots. \tag{4.3}$$

Then $\{\phi_n \mid n = 1, 2, \cdots\}$ forms an orthonormal basis of the smooth function space on $\mathcal{M}$ [28]. A well-known example of the LB eigen-problems is the LB eigen-problem of the unit sphere (S^2, g_0), namely, $\triangle_{g_0}\phi_n = -\lambda_n\phi_n,\ n = 0, 1, 2, \cdots$, whose solutions are spherical harmonic functions. More specifically, if we use the standard spherical coordinate (θ, ξ) for the unit sphere in $\mathbb{R}^3$:

$$\begin{cases} x = \sin\theta\cos\xi, \\ y = \sin\theta\sin\xi, \\ z = \cos\theta, \end{cases} \qquad \theta \in [0, \pi],\ \xi \in [0, 2\pi), \tag{4.4}$$

the spherical harmonic functions can be written as smooth functions in θ and ξ. We are especially interested in the first three nontrivial spherical harmonic functions:

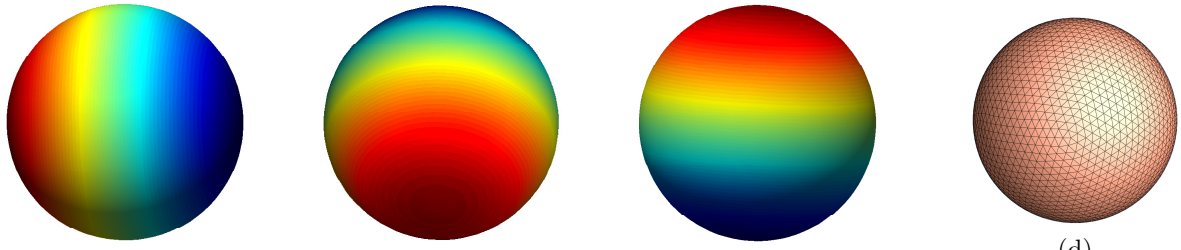
$$\begin{cases} \phi_1 = \frac{1}{2}\sqrt{\frac{3}{\pi}}\sin\theta\cos\xi, \\ \phi_2 = \frac{1}{2}\sqrt{\frac{3}{\pi}}\sin\theta\sin\xi, \\ \phi_3 = \frac{1}{2}\sqrt{\frac{3}{\pi}}\cos\theta. \end{cases} \tag{4.5}$$

An interesting observation is that the first three nontrivial spherical harmonic functions ϕ_1, ϕ_2, ϕ_3 provide us with left-right, up-down and forward-backward structures of the given sphere (see figure 4.1). More precisely, $\phi_1^2 + \phi_2^2 + \phi_3^2 = \frac{3}{4\pi}$, which can be utilized as good coordinates to construct the Star map. In other words, we define:

$$\mathfrak{S}_\Phi : S^2 \to S^2, \quad \mathfrak{S}_\Phi(p) = \frac{\vec{c} - \vec{\Phi}(p)}{\|\vec{c} - \vec{\Phi}(p)\|}, \tag{4.6}$$

where $\vec{\Phi} = (\phi_1, \phi_2, \phi_3)$ and $\vec{c} = \frac{\int_{\mathcal{M}} \vec{\Phi} \mathrm{d}\mathcal{M}}{\int_{\mathcal{M}} \mathrm{d}\mathcal{M}}$.

An advantage of using $\vec{\Phi} = \{\phi_1, \phi_2, \phi_3\}$ as coordinates to construct the Star map is that they are all intrinsically defined on the unit sphere. Thus the Star map construction in this way does not depend on the chosen Euclidean coordinate representation of the sphere. Moreover, due to the analytic form of ϕ_1, ϕ_2, ϕ_3 in (4.5), the Star map obtained from them is a smooth one-to-one and onto map from S^2 to S^2.



(a) (b) (c) (d)

FIG. 4.1. *(a), (b), (c) The first three nontrivial LB eigenfunctions ϕ_1, ϕ_2 and ϕ_3 color-coded on the unit sphere. (d) The surface reconstructed by ϕ_1, ϕ_2 and ϕ_3.*

A general given genus-0 closed surface $\mathcal{M}$ can be viewed as a topological manifold S^2 with a given metric g. Since there always exits a conformal map between $\mathcal{M} = (S^2, g)$ and (S^2, g_0) [7], we can let the corresponding conformal factor of (S^2, g) be e^{2u}. Namely, we have $e^{2u}g = g_0$. Consider the Laplace-Beltrami

eigen-problems on $\mathcal{M}$ with respect to the metric g_0:

$$\triangle_{g_0}\phi_n = \triangle_{e^{2u}g}\phi_n = -\lambda_n\phi_n,\ n = 0, 1, 2, \cdots . \tag{4.7}$$

Therefore, the solutions of (4.7) on $\mathcal{M}$ are spherical harmonic functions. On the other hand, by the definition of the Laplace-Beltrami operator in (4.2), we can have:

$$\triangle_{e^{2u}g}\phi = \frac{1}{e^{2u}\sqrt{G}}\sum_{i=1}^{2}\frac{\partial}{\partial x_i}(e^{2u}\sqrt{G}\sum_{j=1}^{2}e^{-2u}g^{ij}\frac{\partial\phi}{\partial x_j}) = e^{-2u}\triangle_g\phi, \tag{4.8}$$

where (g^{ij}) is the inverse matrix of $g = (g_{ij})$ and $G = \det(g_{ij})$. By combining with (4.7) and (4.8), we have the weighted Laplace-Beltrami eigensystem:

$$\triangle_g\phi_n^u = -\lambda e^{2u}\phi_n^u, \quad n = 0, 1, 2, \cdots \tag{4.9}$$

whose solutions are spherical harmonic functions. Therefore, from the properties of the first three spherical harmonic functions in (4.5), $\phi_1^u, \phi_2^u, \phi_3^u$ should satisfy $(\phi_1^u)^2 + (\phi_2^u)^2 + (\phi_3^u)^2 = \frac{3}{4\pi}$. In other words, if e^{2u} is the conformal factor of $(\mathcal{M}, g)$ to (S^2, g_0), $\phi_1^u, \phi_2^u, \phi_3^u$ will provide three good coordinates of $\mathcal{M}$ for the construction of the Star map, which in turn provide us with a one-to-one and onto map from $\mathcal{M}$ to the unit sphere. In summary, we obtain the following theorem:

THEOREM 4.3. *Given a genus-0 surface $(\mathcal{M}, g)$ with conformal factor e^{2u}, let $\vec{\Phi} = (\phi_1^u, \phi_2^u, \phi_3^u)$ be the first three nontrivial eigenfunctions of the weighted Laplace-Beltrami eigensystem:*

$$\triangle_g\phi_n^u = -\lambda e^{2u}\phi_n^u, \quad n = 0, 1, 2, \cdots \tag{4.10}$$

Then the Star map $\mathfrak{S}_{\vec{\Phi}}$ defined from Φ is a smooth one-to-one and onto map from $\mathcal{M}$ to S^2.

For a surface $\mathcal{M}$ with long and sharp parts like the dinosaur neck, tail and legs in Figure 3.4, the result mapping $\vec{F}_0 : \mathcal{M} \to S^2$ from Algorithm 1 contains artificial foldings introduced by the initial Gauss map. In this case, we can view $\vec{F}_0$ as an approximation of the conformal map which can give us the first approximation of the "conformal factor", e^{2u_0}, with respect to $\vec{F}_0$. However, e^{2u_0} might be inaccurate in the long and sharp feature parts. Due to the global properties of weighted LB eigen-system, the Star map $\mathfrak{S}_{\vec{\Phi}}$ obtained from the weight LB eigen-system helps us gradually correct the foldings. This motives us to propose the following second algorithm for computing a conformal map with conformal correction by weighted LB eigen-projection, which is also illustrated in Figure 4.2.

Algorithm 2: Conformal Correction by weighted LB Eigen-projection

1. Compute harmonic map $\vec{F}_0 : \mathcal{M} \to S^2$ using Algorithm 1.
   Compute the corresponding e^{2u_0} using the approximation formula (3.8).

Iterate the following steps starting from $k = 1$.

2. Given the conformal map $\vec{F}_{k-1}$ and conformal factor $e^{2u_{k-1}}$, solve (4.10).
3. Construct a Star map using $\{\phi_1^{u_{k-1}}, \phi_2^{u_{k-1}}, \phi_3^{u_{k-1}}\}$.
4. Start Algorithm 1 for the Star map and obtain $\vec{F}_k$ and u_k.

We would like to point out that it is challenge to quantitatively characterise foldings, thus it is hard to theoretically predict the step number of the above conformal correction. Generally speaking, the step number of the conformal correction is dependent on the complexity of the input surface. However, according

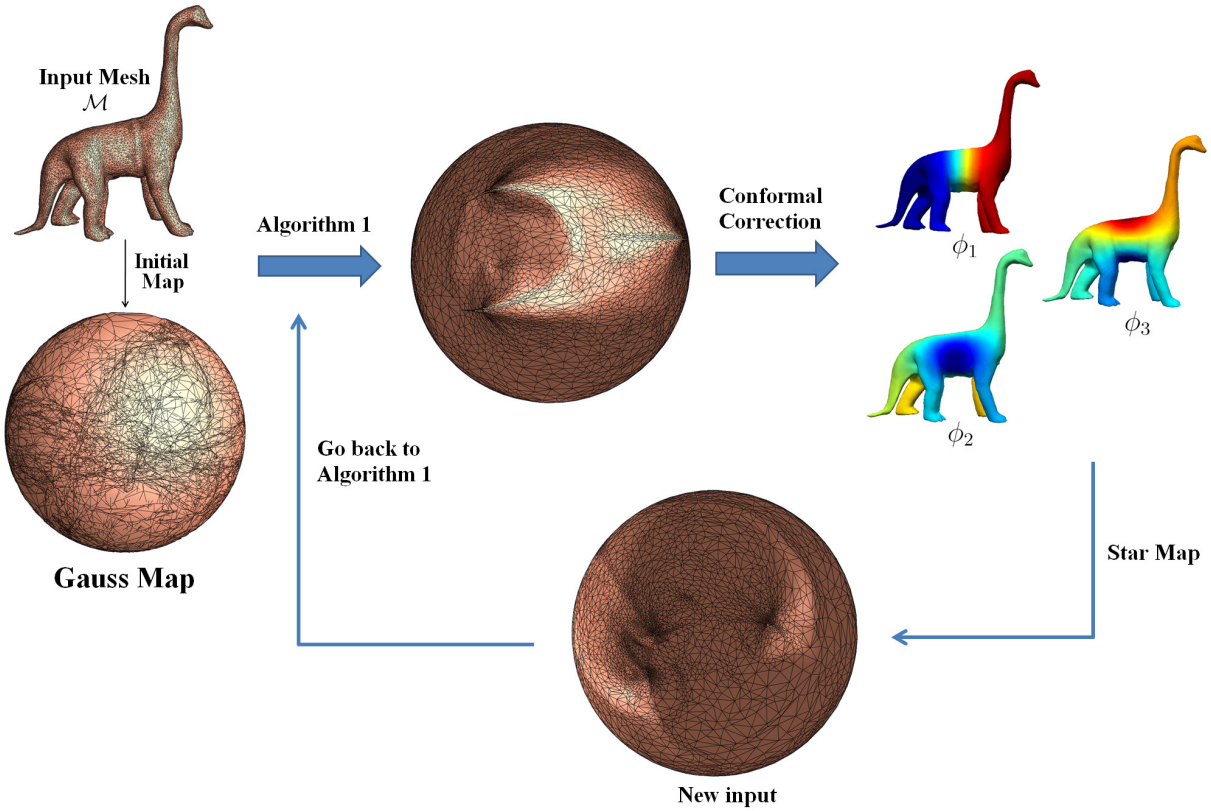


FIG. 4.2. *An illustration of Algorithm 2*

to our experiments in Section 5 to surfaces with long and sharp features as complex as the shapes of dinosaur, armadillo and bird, only two steps of conformal correction will provide us folding free conformal maps.

Similar to using the finite element method (FEM) for computing the LB eigen-system (4.9) on triangulated surfaces [29, 30, 31, 32], we use FEM to solve the above weighted LB eigen-system in the rest of this section. For any given surface $\mathcal{M}$ in $\mathbb{R}^3$, we represent $\mathcal{M}$ as a triangular mesh $\{V = \{p_i\}_{i=1}^N, T = \{T_l\}_{l=1}^L\}$, where $p_i \in \mathbb{R}^3$ is the ith vertex and T_l is the l-th triangle. One can choose linear elements $\{e_i\}_{i=1}^N$, which satisfy $e_i(p_j) = \delta_{i,j}$ in the Kronecker delta notion, and write $\mathbb{E} = Span_{\mathbb{R}}\{e_i\}_{i=1}^N$. Then the discrete version of the weak form of the continuous problem is to find a $\phi \in \mathbb{E}$ such that

$$\sum_l \int_{T_l} \nabla_{\mathcal{M}}\phi \nabla_{\mathcal{M}}\eta = \lambda \sum_l \int_{T_l} e^{2u}\phi\eta, \quad \forall \eta \in \mathbb{E}. \tag{4.11}$$

If we write

$$\begin{cases} \phi = \sum_i^N x_i e_i \\ A = (a_{ij})_{N\times N}, \quad a_{ij} = \sum_l \int_{T_l} \nabla_{\mathcal{M}} e_i \nabla_{\mathcal{M}} e_j \\ B_u = (b_{ij})_{N\times N}, \quad b_{ij} = \sum_l \int_{T_l} e^{2u} e_i e_j, \end{cases} \tag{4.12}$$

where the stiffness matrix A is symmetric and the mass matrix B_u is symmetric and positive definite, and

the discrete variational problem is equivalent to the generalized matrix eigen-problem:

$$\begin{cases} Ax = \lambda B_u x, \text{where } x = (x_1, \cdots, x_N)^T \\ \phi = \sum_i^N x_i e_i. \end{cases} \tag{4.13}$$

Note that both A and B_u are $N \times N$ sparse matrices. The problem can be efficiently solved by a variety of numerical packages. For instance, a standard function "eigs" in Matlab can be used to solve the above generalized eigenvalue problem.

5. Experimental Results. In this section, we demonstrate the efficiency and robustness of the proposed algorithms for computing harmonic maps of genus-0 surfaces to the unit sphere. On efficiency, we compare the proposed Algorithm 1 with the existing algorithm in [1]. On robustness against foldings, we extend the comparison to include the proposed Algorithm 2 on surfaces with long and sharp features. All experiments are performed on a PC with a 2.66GHz CPU. It is worth noting that the algorithm [1] was written in C++, our Algorithms 1 was implemented in MATLAB (Release 7.9.0) and the computation of weighted Laplace-Beltrami eigen-system was coded in C++. Since C++ usually tends to be efficient than MATLAB, the programming language difference does not introduce any biases toward our algorithms.

In our first experiment, we compare the speeds of Algorithm 1 and the algorithm [1] on several different surfaces listed in Figure 5.1. The number of iterations and computation times of both algorithms are given in Table 5.1. It is clear that Algorithm 1 is much more efficient on the given surfaces.

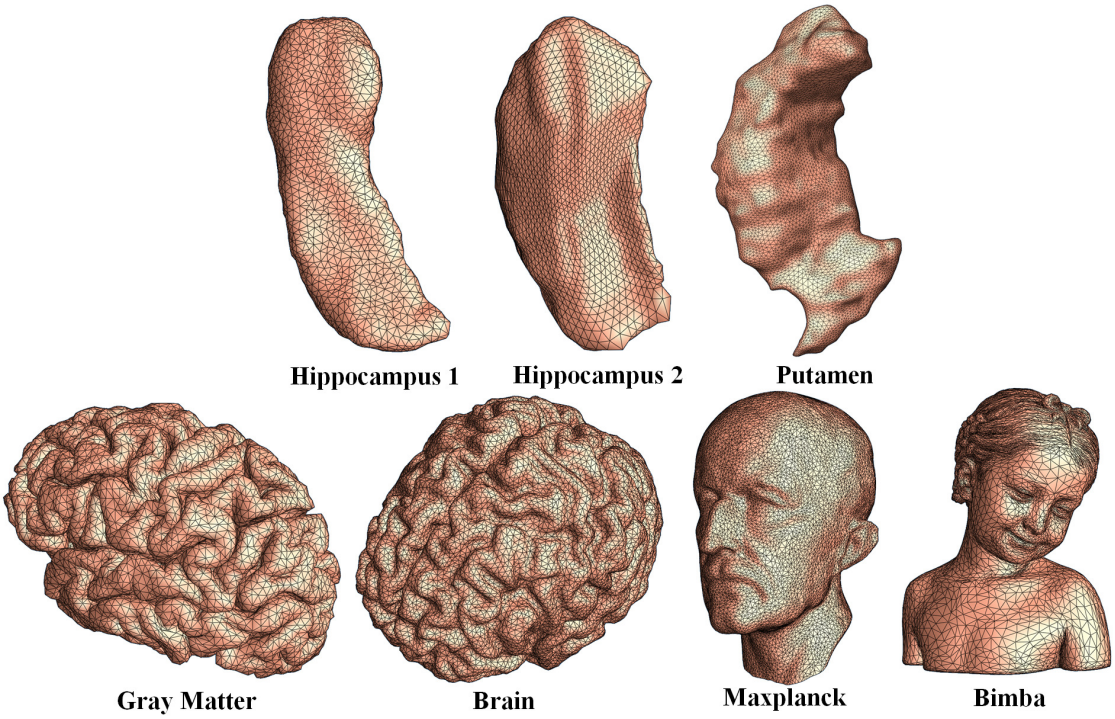


FIG. 5.1. *Surfaces for comparisons listed in table 5.1*

Our second experiment demonstrates folding-free results of Algorithm 2. The first row of Figure 5.2 depicts the results of Algorithm 1 with obvious foldings and inaccurate conformal factors, which are highlighted by rounded boxes. Foldings are also observed in the results of the algorithm [1] given in the second

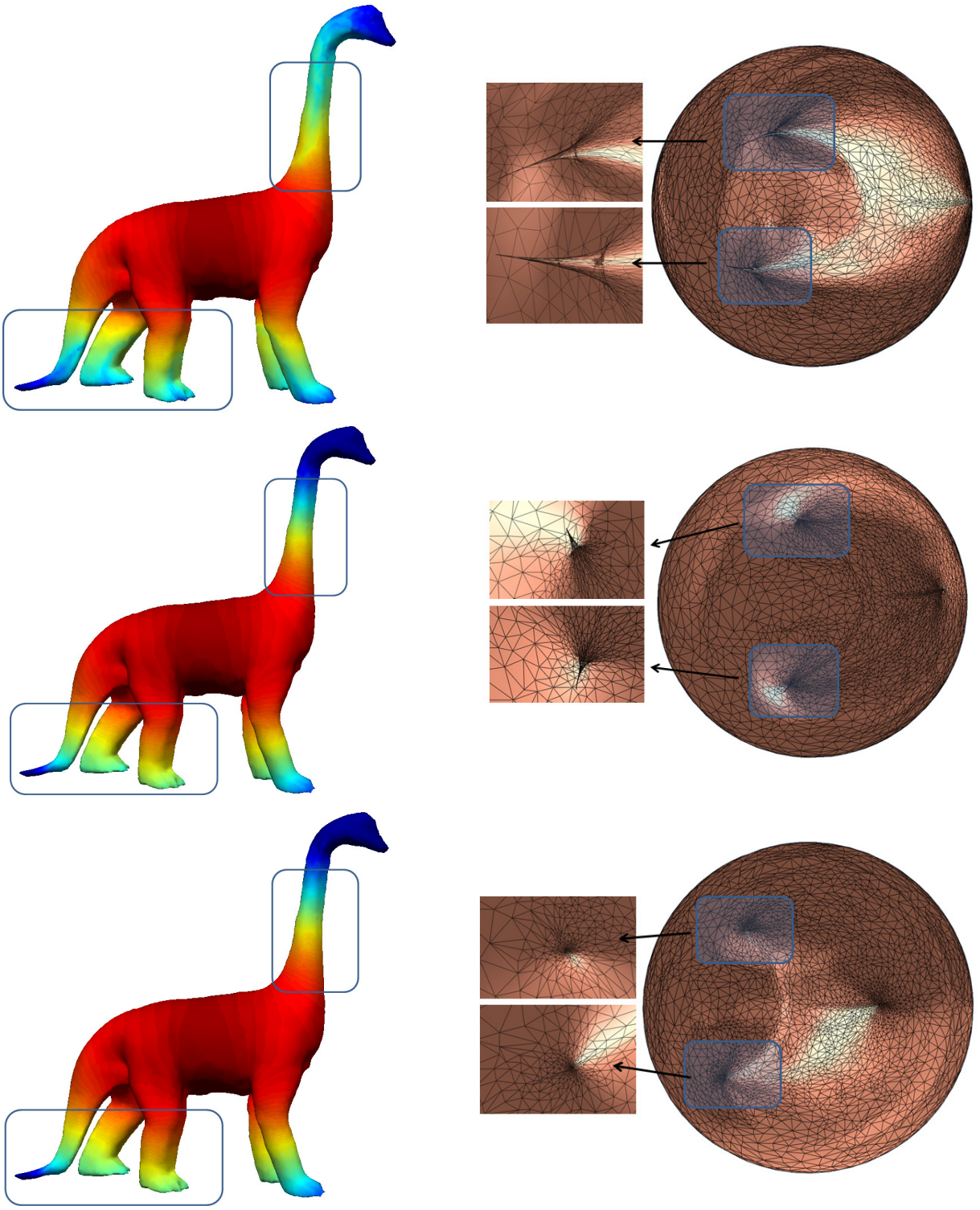


FIG. 5.2. *The first row: the mapping results and the corresponding conformal factors before conformal correction; The second and third rows: the mapping results and the corresponding conformal factors after one and two conformal-correction iterations, respectively.*

surface	# of vertices	ϵ	The proposed Algorithm 1		Algorithm in [1]	
			# of iterations	time(s)	# of iterations	time(s)
Hippocampus 1	2000	1e-10	210	0.19	790	42.63
Hippocampus 2	2562	1e-10	4225	5.83	1594	108.36
Putamen	10002	1e-10	1414	4.80	2684	760.33
Gray matter	10000	1e-10	1598	5.79	2794	822.90
Brain	15002	1e-10	1329	17.80	1560	664.97
Maxplanck	12556	1e-10	892	14.19	1990	704.01
Bimba	15002	1e-10	1432	17.40	3197	1358.72

TABLE 5.1
Comparison between the proposed Algorithm 1 and the algorithm in [1]

column of Figure 5.3. The second and third rows of Figure 5.2 show the results after one and two conformal correction iterations in Algorithm 2, respectively. Foldings are completely removed in the second result, which gives an accurate conformal map.

surface	# of vertices	ϵ	The proposed Algorithm 2 (Foldings Removed)				Algorithm in [1] (Foldings Remained)	
			# of iterations	# of iterations	# of iterations	time(s)	# of iterations	time(s)
Dino	5524	1e-10	4480	1518	3652	91.11	2854	835.53
Dilo	9731	1e-10	3610	5000	4184	73.94	3106	513.56
Bird 1	950	1e-10	1054	4444	1038	2.64	716	19.29
Bird 2	926	1e-10	592	1674	5000	14.44	599	14.09
Armadillo	16519	1e-10	4164	294	496	70.68	3355	1648.30

TABLE 5.2
Comparison between the proposed Algorithm 2 and the algorithm in [1].

To further demonstrate the robustness of Algorithm 2, we compute the conformal maps obtained by Algorithm 2 and the algorithm [1] for a Dino surface, a Dilo surface, two bird surfaces and an armadillo surface. With only two steps of conformal corrections, the results of Algorithm 2 are free of foldings. The corresponding conformal maps are illustrated in the third column of Figure 5.3. In comparison, the results of the algorithm [1] are given on the second column of Figure 5.3, and they contain foldings introduced by initial maps. In Table 5.2, we list the iteration numbers of each correction step and the total computing times of Algorithm 2, as well as those of the Algorithm [1]. In summary, the proposed Algorithm 2 can efficiently generate the folding-free conformal maps.

6. Conclusions. This paper introduces an efficient algorithm for minimizing the harmonic energy problem, which quickly computes conformal maps for genus-0 surfaces. To avoid foldings introduced by initial solutions which arise on surfaces with long and sharp features, a conformal correction iteration based on the weighted Laplace-Beltrami projection is proposed. Numerical comparisons to the existing method [1] on several different surfaces demonstrate the efficiency and accuracy of the proposed algorithms.

Acknowledgements. Rongjie Lai's work is supported by Zumberge Individual Award from USC's James H. Zumberge Faculty Research and Innovation Fund. Zaiwen Wen's work is supported in part by an internal research fund granted by Shanghai Jiaotong University. Wotao Yin's work is supported in part by NSF grant DMS-0748839 and ONR grant N00014-08-1-1101. Xianfeng Gu's work is is supported in part

of NSF Nets 1016286, NSF IIS 0916286, NSF CCF 1081424 and ONR N000140910228. Lok Ming Lui's work is supported in part of the CUHK Direct Grant (Project ID: 2060413) and HKRGC GRF (Project ID: 2130271).

REFERENCES

[1] X. Gu and S. Yau. Computing conformal structures of surfaces. *Communications in Information and Systems*, 2(2):121–146, 2002.

[2] X. Gu, Y. Wang, T. F. Chan, P. Thompson, and S. T. Yau. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Transaction on Medical Imaging*, 23:949–958, 2004.

[3] LM Lui, Y Wang, P.M. Thompson, and T.F. Chan. Landmark constrained genus zero surface conformal mapping and its application to brain mapping research. *Applied Numerical Mathematics*, 57:847–858, 2007.

[4] L.M. Lui, X. Gu, T.F. Chan, and S.-T. Yau. Variational method on riemann surfaces using conformal parameterization and its applications to image processing. *Methods and Applications of Analysis*, 15(4):513–538, 2008.

[5] B. Levy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. *Proceeding of ACM SIGGRAPH*, 2002.

[6] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. *Proceeding of ACM SIGGRAPH*, 1995.

[7] R. Schoen and S.-T. Yau. *Lectures on Differential Geometry*. International Press, 1994.

[8] P. Alliez, M. Meyer, and M. Desbrun. Interactive geometry remeshing. *Proceeding of ACM SIGGRAPH*, 2002.

[9] T. Kanai, H. Suzuki, and F. Kimura. Three-dimensional geometric metamorphosis based on harmonic maps. *The Visual Computer*, 14(4):166–176, 1998.

[10] M. K. Hurdal, K. Stephenson, P.L. Bowers, D.W.L. Sumners, and D.A. Rottenberg. Coordinate systems for conformal cerebellar flat maps. *NeuroImage*, S467, 2000.

[11] S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, G. Sapiro, and M. Halle. Conformal surface parameterization for texture mapping. *IEEE Trans. Visualization and Computer Graphics*, 6(2):181—189, 2000.

[12] LM Lui, S Thiruvenkadam, Y Wang, P.M. Thompson, and T.F. Chan. Optimized conformal surface registration with shape-based landmark matching. *SIAM Journal of Imaging Sciences*, 3(1):52–78, 2010.

[13] X. Gu and S.T. Yau. Computing conformal structures of surfaces. *Communications in Information and Systems*, 2(2):121–146, 2002.

[14] M Jin, Y Wang, Yau S.T., and Gu X. Optimal global conformal surface parameterization. *IEEE Visualization, Austin, TX*, pages 267–274, 2004.

[15] M. M.Jin, J. Kim, F. Luo, and X. Gu. Discrete surface ricci flow. *IEEE Transacation on Visualization and Computer Graphics*, 14(5):1030–1043, 2008.

[16] Y. Yang, J. Kim, F. Luo, S. Hu, and X. Gu. Optimal surface parameterization using inverse curvature map. *IEEE Transacation on Visualization and Computer Graphics*, 14(5):1054–1066, 2008.

[17] Z. Wen and W. Yin. A feasible method for optimization with orthogonality constraints. *UCLA CAM 10-77*, 2010.

[18] R. Schoen and S.-T. Yau. *Lectures on Harmonic Maps*. International Press, 1994.

[19] J. Jost. Riemannian geometry and geometric analysis. *Springer, 3rd edition*, 2001.

[20] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, Princeton, NJ, 2008.

[21] Uwe Helmke and John B. Moore. *Optimization and dynamical systems*. Communications and Control Engineering Series. Springer-Verlag London Ltd., London, 1994. With a foreword by R. Brockett.

[22] Constantin Udrişte. *Convex functions and optimization methods on Riemannian manifolds*, volume 297 of *Mathematics and its Applications*. Kluwer Academic Publishers Group, Dordrecht, 1994.

[23] Jonathan Barzilai and Jonathan M. Borwein. Two-point step size gradient methods. *IMA J. Numer. Anal.*, 8(1):141–148, 1988.

[24] Hongchao Zhang and William W. Hager. A nonmonotone line search technique and its application to unconstrained optimization. *SIAM J. Optim.*, 14(4):1043–1056, 2004.

[25] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. *Visualization and Mathematics III. (H.C. Hege and K. Polthier, Ed.) Springer Verlag*, pages 35–57, 2003.

[26] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Discrete differential geometry operators in nd. *In Proc. VisMath'02 Berlin Germany*, 2002.

[27] Guoliang Xu. Convergent discrete laplace-beltrami operator over triangular surfaces. *Proceedings of Geometric Modelling and Processing*, pages 195–204, 2004.

[28] I. Chavel. *Eigenvalues in Riemannian Geometry*. Academic press. INC, 1984.

[29] M. Reuter, F.E. Wolter, and N. Peinecke. Laplace-Beltrami spectra as Shape-DNA of surfaces and solids. *Computer-Aided Design*, 38:342–366, 2006.

[30] Y. Shi, R. Lai, S. Krishna, N. Sicotte, I. Dinov, and A. W. Toga. Anisotropic Laplace-Beltrami eigenmaps: Bridging Reeb graphs and skeletons. In *Proc. MMBIA*, 2008.

[31] R. Lai, Y. Shi, K. Scheibel, S. Fears, R. Woods, A. W. Toga, and T. F. Chan. Metric-induced optimal embedding for intrinsic 3D shape analysis. *CVPR*, 2010.

[32] Y. Shi, R. Lai, R. Gill, D. Pelletier, D. Mohr, N. Sicotte, and A. W. Toga. Conformal metric optimization on surface (cmos) for deformation and mapping in laplace-beltrami embedding space. *MICCAI*, 2011.

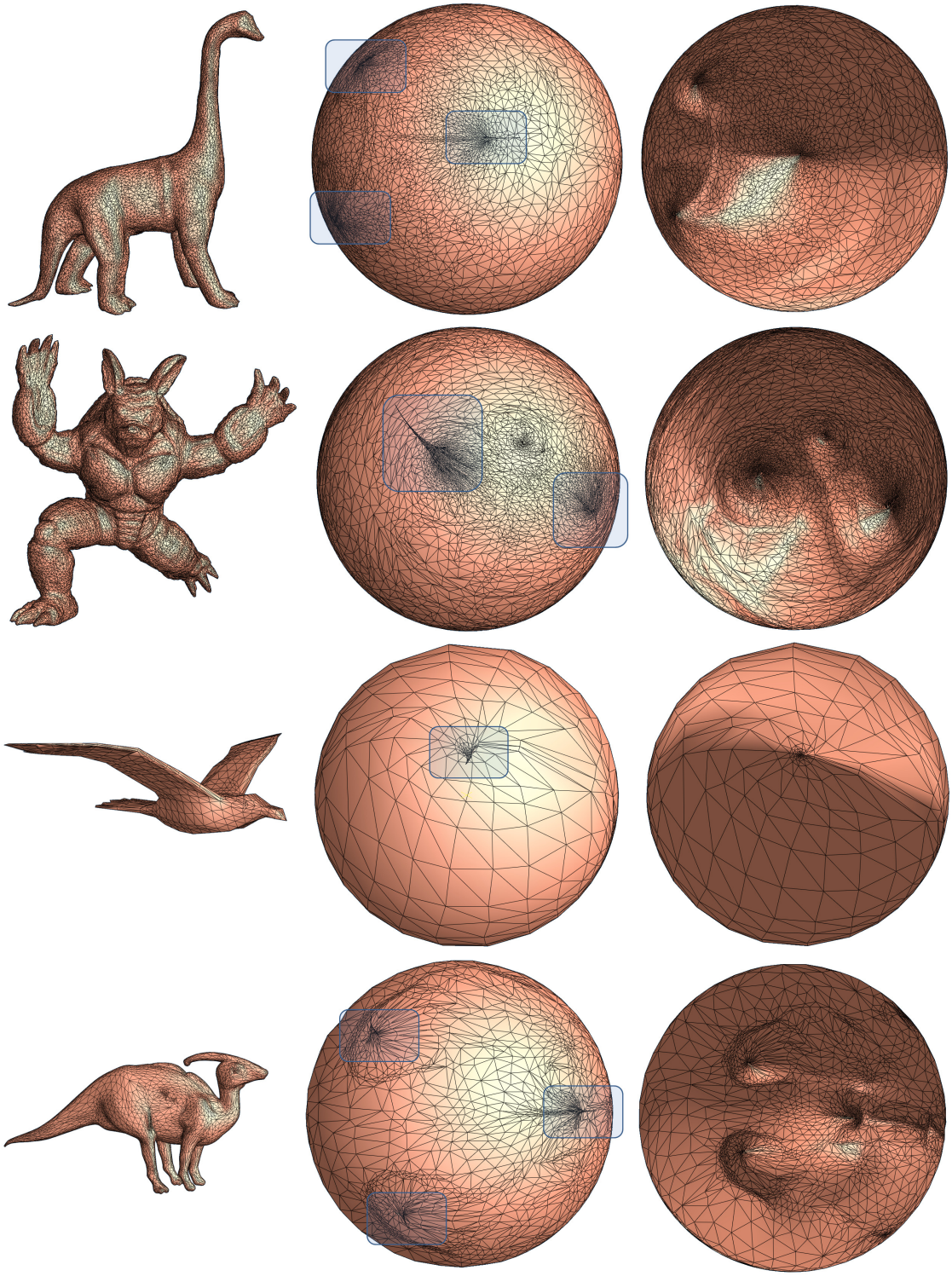


FIG. 5.3. *Comparison of the results on Dino, Armadillo, Bird and Dilo surfaces obtained by Algorithm 2 and the algorithm in [1]. First column: the input surfaces; Second column: the results of the algorithm in [1] with foldings; Third column: the results of the proposed Algorithm 2 with no foldings.*