Biharmonic Volumetric Mapping Using Fundamental Solutions

Huanhuan Xu, Wuyi Yu, Shiyuan Gu, and Xin Li, *Member*, *IEEE Computer Society*

Abstract—We propose a biharmonic model for cross-object volumetric mapping. This new computational model aims to facilitate the mapping of solid models with complicated geometry or heterogeneous inner structures. In order to solve cross-shape mapping between such models through divide and conquer, solid models can be decomposed into subparts upon which mappings is computed individually. The biharmonic volumetric mapping can be performed in each subregion separately. Unlike the widely used harmonic mapping which only allows C^0 continuity along the segmentation boundary interfaces, this biharmonic model can provide C^1 smoothness. We demonstrate the efficacy of our mapping framework on various geometric models with complex geometry (which are decomposed into subparts with simpler and solvable geometry) or heterogeneous interior structures (whose different material layers can be segmented and processed separately).

Index Terms—Volumetric mapping, biharmonic mapping

1 Introduction

Computing lowly distorted volumetric mapping between two given objects is an important geometric problem in computer graphics and visualization. It serves as an important preprocessing step in many tasks in broad areas of computer-aided design and analysis, industrial manufacturing, medical image analysis, etc. With the advance of data acquisition techniques, massive volumetric models with many multiple attributes and complex geometry are collected and need to be processed. Surface mapping has been extensively studied and many effective algorithms have been developed to handle shells of 3D objects. However, many real-world objects are volumetric and have interior contents. Effective volumetric parameterization is critical to process such data that have not only boundary geometry but also interior material/structures. Due to the much more complicated topological and geometric structures of 3D manifolds, existing volumetric mapping techniques are still inadequate. A desirable volumetric mapping model should usually have the following properties:

1. *Generality*. Many real-world volumetric models have complicated geometry and interior structure (e.g., multiple material layers), therefore, the mapping algorithm needs to be general and can handle data with nontrivial topological types.
2. *Efficiency*. Volumetric models usually have much bigger sizes, compared with surface data, while computational efficiency is important in many engineering applications. An effective computation strategy to handle huge-size and geometrically complex volumetric models is through divide and conquer based on model decomposition. We can partition the complex model into solvable subdomains for individual processing. Such a partitioning is often desirable and sometime necessary. First, unlike surface parameterization, bijective (i.e., no degeneracy or flip-over) volumetric map may not exist globally between two solid models with complex geometry. Second, volumetric models are often so big that the direct computation is prohibitive and efficient computation conducted on smaller sub-domains is preferred.
3. *Smoothness*. A mapping indicates a transformation between two solid objects; it should be smooth, i.e., minimizing the stretch of the transformation and thus physically natural. Individually computed maps on subparts should compose a global map with smooth transition across the cutting interfaces.
4. *Feature preserving*. Volumetric models usually have different materials, rich interior structures and features. The mapping should be feature-aware. For example, corresponding features points/curves or layers of materials should be aligned correctly.
5. *Linear precision*. If the mapping (deformation) of the boundary surface follows a linear transformation, an induced volumetric mapping in the interior region that reproduces this transformation (see Section 5) is natural and therefore desirable.

Our goal is to develop a volumetric mapping model with these desirable properties.

The harmonic function is widely used in shape mapping and deformation, because it indicates the vanishing Laplacian inside the domain. A volumetric map is a harmonic map if all its three components are harmonic. It minimizes the stretching energy and therefore indicates a physically natural transformation between two solid domains. A discrete harmonic mapping can be computed

- *H. Xu, W. Yu, and X. Li are with the School of Electrical Engineering and Computer Science, Louisiana State University, Baton Rouge, LA 70803. E-mail: {hxu4, wuyiyu, xinli}@lsu.edu.*
- *S. Gu is with the Department of Mathematics, Louisiana State University, Baton Rouge, LA 70803. E-mail: gshy@math.lsu.edu.*

Manuscript received 18 Jan. 2012; revised 5 June 2012; accepted 10 Aug. 2012; published online 20 Aug. 2012.
Recommended for acceptance by W. Wang.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2012-01-0019.
Digital Object Identifier no. 10.1109/TVCG.2012.173.

efficiently by solving a linear system. However, harmonic volumetric mapping has its limitation in a divide-and-conquer computation framework. Individually computed harmonic maps could have undesirable discontinuity across the partitioning boundary interface. Typically, one is only able to enforce C^0 continuity across the mapping boundary. In other words, we can have only the positional constraints on the boundary points, but not their derivatives at the same time. Only C^0 continuity along the boundary interface introduces undesirable artifacts in parameterization and subsequent applications such as meshing, texture mapping, deformation, and physical simulations. Specifically, when volumetric parameterization is used for isogeometric analysis, higher continuity is often required.

Therefore, we propose to use a biharmonic model to construct the mapping on decomposed subdomains with C^1 continuity. The boundary condition in both positions and normal derivatives can be controlled; hence, we can obtain better smoothness across the cutting interface in a divide-and-conquer computation. Heterogeneous volumetric models with multiple materials can also be segmented and parameterized in a similar manner.

The **main contributions** of this paper include

1. We propose a biharmonic model to solve volumetric mappings for 3D heterogeneous data, following the surface mapping and boundary normal constraints. Compared with the harmonic map, it provides better boundary continuity control.
2. We develop a biharmonic volumetric mapping computation framework based on the method of fundamental solutions (MFS). Complex models can be decomposed and then parameterized effectively.
3. We demonstrate the effectiveness of our method on various models and show its applications in hex-remeshing and temporal data analysis.

We organize the remainder of this paper as follows: Section 2 reviews the related literatures. In Section 3, we formulate the problem and give an overview of our method. Algorithm details on the boundary positional constraint and normal derivative constraint configurations will be discussed in Sections 4.1 and 4.2, respectively. We discuss the properties of biharmonic volumetric mapping and some implementation details in Section 5, and show experiment results in Section 6. We apply our biharmonic mapping in hex-remeshing and temporal data parameterization in Section 7.

2 Related Work

2.1 Shape Modeling Using Biharmonic Functions

Biharmonic equation is a fourth-order partial differential equation used to minimize the thin-plate energy [1]. In recent years, it has been used to generate high-quality surface patch with the given positional and derivative boundary conditions [1], [2]. Relaxing each vertex of a discrete triangle mesh with harmonic weights or biharmonic weights can get a smooth surface [3]. Biharmonic equation is also used to do real-time mesh deformation [4], [5]. In [4], Helenbrook defines the boundary condition based on an explicit solution and shows the results on the 2D planar mesh. So, it is mainly applied in fluid flows. Jacobson et al. [5] set the solution of biharmonic equation as the linear bending weights for 2D/3D shape deformation and the implementation is based on the mixed finite element method. Lipman et al. [6] define the metric on 2D manifolds based on the biharmonic equation with the normal derivative equals to zeros along the boundary. Tankelevich et al. [7] compare surface reconstruction from points sets based on the solution of harmonic and biharmonic equations; they develop a special boundary condition for biharmonic equation such that it can be solved faster.

2.2 Volumetric Mapping

In recent years, volumetric mapping have gained great interest due to its rich applications in many fields such as computer-aided manufacturing [8], meshing [9], [10], shape registration [11], [12], [13], and trivariate spline construction [14], [15], [16]. Wang et al. [12] discretize the volumetric harmonic energy on the tetrahedral mesh using the finite element method, parameterized volumetric shapes over solid spheres by a variational algorithm. Xia et al. [10] and Han et al. [9] use this discrete harmonic volumetric map in polycube parameterization. Most closely related to this work, in [13], we compute the harmonic volumetric mapping between two solid objects using fundamental solution methods. Later, we incorporate feature alignment in this volumetric mapping framework [17]. In order to handle complicated shapes, the global harmonic mapping often cannot guarantee its bijectiveness. Decomposing complex shapes into simpler subparts is a potentially effective solution. But in such a divide-and-conquer scheme, using the harmonic model, we can at most guarantee a C^0 continuity along the partitioning boundary, which usually is not smooth enough, as an example shown in Fig. 4e.

Volumetric interpolation is a powerful tool for shape deformation. Ju et al. [18] generalize the mean value coordinates [19] from surfaces to volumes to get a smooth volumetric interpolation for cage-based deformation. Joshi et al. [20] present harmonic coordinates with nonnegative weights for volumetric interpolation and deformation in concave regions. Martin et al. [14] parameterize volumetric model with trivial topology to a cylinder using the finite element method, and later generalize the algorithm [15] to more complicated models with medial surfaces. Lipman et al. [21] develop Green's coordinates for volumetric deformation. Patanè et al. [22] use Radial Basis Function to approximate volumetric function along the volume data.

3 Algorithm Overview

3.1 Problem Definition

We consider the computation of the mapping $\Phi : \Omega \rightarrow M$, where volumetric domains $\Omega, M \subset \mathcal{R}^3$ are embedded in 3D space

$$\begin{cases} \nabla^4 \Phi = 0, & \text{in } \Omega, \\ \Phi = f, & \text{on } \partial\Omega, \\ \frac{\partial \Phi}{\partial n} = g, & \text{on } \partial\Omega, \end{cases} \tag{1}$$

where $\partial\Omega$ and ∂M denote the boundary surface of Ω and M, n is the surface normal on the domain boundary, $\partial\Phi/\partial n$

then denotes the outward normal derivative, f and g are prescribed functions: f is decided by a given surface mapping $f : \partial\Omega \to \partial M$, and $\frac{\partial\Phi}{\partial n}$ indicates the derivative along the normal direction on each boundary point. The mapping (vector function Φ) can be decomposed into three components ϕ^1, ϕ^2, ϕ^3 for x, y, z axis directions, each of which should be biharmonic.

Under reasonable (enough smoothness) assumption of f and g, the biharmonic model in (1) can be uniquely solved given a pair of boundary conditions f and g. A proof is sketched in the appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2012.173. In our experiments, we found that for most real-world volumetric data, our computed surface mapping f and derivative constraint g satisfy this smoothness assumption.

3.2 Solving Biharmonic Mapping Using MFS

Using the MFS [23], the solution (for simplicity, we use ϕ to denote each component ϕ^i) to (1) can be approximated by a linear combination of fundamental solutions of both the harmonic and biharmonic equations:

$$\phi(\mathbf{h}, \mathbf{b}, \mathbf{Q}, \mathbf{x}) = \sum_{j=1}^{N_s} h_j H(\mathbf{q}_j, \mathbf{x}) + \sum_{j=1}^{N_s} b_j B(\mathbf{q}_j, \mathbf{x}), \qquad (2)$$

where

- *Kernels*: $H(\mathbf{q}_j, \mathbf{x}) = 1/(4\pi|\mathbf{q}_j - \mathbf{x}|)$ is the fundamental solution of the harmonic term, and $B(\mathbf{q}_j, \mathbf{x}) = |\mathbf{q}_j - \mathbf{x}|/(8\pi)$ is the fundamental solution of the biharmonic term;
- $\mathbf{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_{N_s}\}$ is a $3N_s$-dimensional vector, representing the set of N_s singularity points, each $\mathbf{q}_j = [q_{3j-2}, q_{3j-1}, q_{3j}]^T \in \mathcal{R}^3$ denotes the position of a singularity point,
- $\mathbf{h} = [h_1, h_2, \ldots, h_{Ns}]^T$ and $\mathbf{b} = [b_1, b_2, \ldots, b_{Ns}]^T$ are vectors of harmonic and biharmonic coefficients associated with these N_s singularity points, i.e., the degree of the freedom to control the boundary fitting.

The vanishing bilaplacian operator on ϕ is enforced by the fundamental solutions H and B, we only need to ensure the function satisfy the boundary condition. This boundary fitting is performed on a set of N_c *collocation points* defined on the domain boundary $\partial\Omega$. The kernel function $H(\mathbf{q}_j, \mathbf{x})$ with respect to $\mathbf{q}_j$ (therefore, can also be directly denoted as $H_{\mathbf{q}_j}(\mathbf{x})$) is not defined on this singularity point $\mathbf{x} = \mathbf{q}_j$. Therefore, singularity points $\{\mathbf{q_j}\}$ need to be sampled outside the function domain, i.e., on a surface $\partial\Omega'$ outside Ω, $\Omega \subset \Omega'$. It has been shown [24] that an effective MFS system can be constructed by computing an offset surface outside $\partial\Omega$ then adaptively sampling $\{\mathbf{q_j}\}$.

To perform boundary fitting on each constraint point $\mathbf{x} \in \partial\Omega$, we evaluate $\phi(\mathbf{x})$ using (2). The boundary constraints are then $\phi(\mathbf{x}) = f(\mathbf{x})$ and $\frac{\partial\phi(\mathbf{x})}{\partial n(\mathbf{x})} = g(\mathbf{x})$. Enforcing these constraints on all collocation points reduces to a linear system $A\mathbf{w} = \mathbf{t}$, where the coefficient matrix A's dimension is $2N_C * 2N_S$, $\mathbf{w}$ is the unknown $2N_S$-dimensional coefficient vector $[\mathbf{h}, \mathbf{b}]^T$, and $\mathbf{t}$ is the $2N_C$-dimensional vector indicating the boundary condition evaluated on collocation points. When $N_C > N_S$, this system is an overdetermined system. With this condition, we can solve the system directly. Usually, the coefficient matrix $\mathbf{A}$ is dense and ill-conditioned; hence we can use truncated Singular Value Decomposition (SVD) to improve its numerical stability [24], [25]. In our work, we set $N_C < N_S$, and use a regularization term to avoid over fitting (see Section 5.5). We found this approach gives better numerical efficiency and stability in our mapping computation.

The biharmonic equation (1) can be solved using this above collocation method through a three-step pipeline: 1) place singularity and collocation points, 2) on collocation points, formulate boundary constraints evaluated by coefficients (to be solved) associated with these singularity points, 3) solve a linear system to get all the coefficients, enforcing the boundary constraints.

3.3 Mapping through Model Decomposition

When the volumetric regions Ω and M are simple, we can directly compute their mapping $\phi : \Omega \to M$ using the above method. However, when Ω and M have complex topology or geometry, mapping computation through a divide-and-conquer scheme based on model decomposition is desirable. Specifically, we consistently partition Ω and M into corresponding sets of subparts $\{\Omega_1, \ldots, \Omega_n\}$ and $\{M_1, \ldots, M_n\}$, such that the 1) topology of Ω_i and M_i are the same and 2) the dual graphs of their decompositions are isomorphic [26]. Then, on each corresponding pair Ω_i and M_i, we compute the volumetric mapping $\phi_i : \Omega_i \to M_i$ using the above method individually. The computation of this consistent decomposition and corresponding boundary constraints include the boundary positional constraint function f and the boundary normal derivative constraint function g is not the focus of this work, but will be briefly discussed in Section 4.

4 Boundary Condition Setting

This section elaborates the computation of boundary constraints f and g in (1).

The boundary positional constraint function f is decided by the boundary surface mapping between $\partial\Omega$ and ∂M. In this work, we construct the surface mapping f through model decomposition (Section 4.1).

Function g is usually unknown. In order to have smoothness across cutting boundaries, we derive g from f by using a *local to global* affine approximation technique, namely, we use local affine transformations to approximate a global mapping. It will ensure the unchanged normal derivative along the boundary interface (Section 4.2).

4.1 Positional Constraints by Surface Mapping

We will first discuss our positional constraints setting on a simple volumetric domain then talk about the decomposition and mapping for complex models in the divide-and-conquer scheme.

Surface mapping on a single volumetric domain. To compute the volumetric mapping $\phi : \Omega \to M$, we need to solve the surface map f between the boundaries $\partial\Omega$ and ∂M. Cross-surface parameterization methods such as [27], [28], [26], [29], [30] can be used for computing f. In this work, we use a harmonic intersurface map to serve as the

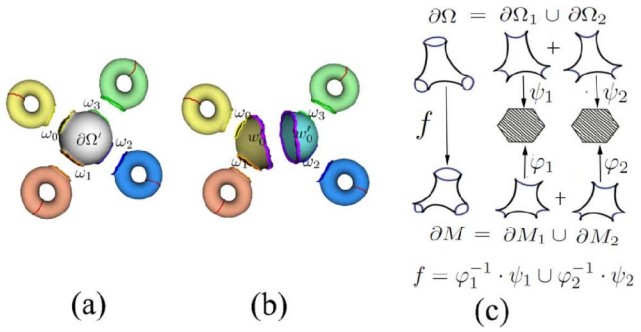


Fig. 1. Decomposition and mapping of the boundary surface of a 3D model. (a) and (b) show the computation of the *pants decomposition* [26]. (c) illustrate the parameterization of two corresponding pants patches, and the composed mapping.

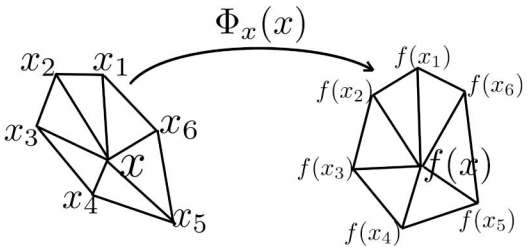


Fig. 2. Local linear approximation. $x_1, x_2, \ldots, x_6$ are the one-ring of the vertex x, under the surface mapping f they become $f(x_1)$, $f(x_2), \ldots, f(x_6)$. In this local region, f is approximated by the linear transformation $\Phi_x(x)$.

boundary positional constraint of our biharmonic volumetric map. We briefly recap our computation algorithm, which is based on [28], [26].

Given a surface $\partial\Omega$ with G genus and B boundaries, first, we remove the *topological handles*, and get a surface (base patch) $\partial\Omega'$ with $G+B$ boundaries and some handle patch. If $G+B=1$, $\partial\Omega'$ is a *topological disk*; if $G+B=2$, $\partial\Omega'$ is a *topological cylinder*; if $G+B=3$, $\partial\Omega'$ is a *topological pants patch* (a handle patch is also a topological pants patch). For $G+B>3$, $\partial\Omega'$ can be decomposed into several topological pants patches (See [26] for detailed constructive algorithm: iteratively remove a pants patch from $\partial\Omega'$ and reduce the boundary number of $\partial\Omega'$ by 1, until $\partial\Omega'$ itself becomes a pants patch). Figs. 1a and 1b show an example of the pants decomposition.

Then, any $\partial\Omega$, reduced to one of the above three topological types, can be parameterized using the harmonic map on the canonical disk, cylinder, and topological hexagons. The corresponding patch ∂M is parameterized likewise. We can then compose the parameterization and get the mapping $f : \partial\Omega \to \partial M$. Fig. 1c shows the process of parameterization of two corresponding pants patch over regular hexahedrons and the composed interpatch surface map.

Divide-and-conquer scheme. Complex Ω and M can be decomposed into subsolid parts $\{\Omega_i\}$ and $\{M_i\}$. We need to have a new surface mapping scheme that set up positional constraints between $\partial\Omega_i$ and ∂M_i separately.

Many shape decomposition techniques have been developed in computer graphics literature (see [31] and [32] for thorough surveys); most of these are for partitioning a single surface following its own geometry. Here, for the mapping purpose, the partitioning of Ω and M need to be *consistent* (i.e., each subpart Ω_i has same topology with its corresponding part M_i, and the dual graphs of these two decompositions are isomorphic). At the same time, decomposition of heterogeneous solid domains may need to follow the materials, semantics, or geometry of the objects. Volumetric decomposition is not the main focus of this paper. In this work, similar to [27], [29], we first get the consistent decomposition on the surface. Then, we use minimal surfaces to fill the topological disks along the interior boundary interface.

To obtain the consistent volumetric decomposition, we first get the consistent surface decomposition. Conventional consistent surface mesh segmentation is done interactively [29], [33], via tracing shortest paths connecting manually placed markers. It can also be computed using automatic manner using such as [26], [29].

In our experiments, we first compute the consistent surface decomposition and intersurface mapping using the algorithm of [26]. Then, it can be extended into the interior volumes. It is computed through a consistent region growing upon the tetrahedral meshes [34]. We start with the compatible skeletal graphs of the two given models, and simultaneously propagate towards boundary with guaranteed visibility. Such a growing can ensure the topologically coherent adjacency relationship among neighboring subregions and provide a consistent volumetric decomposition (the dual graphs of decompositions of the two models are isomorphic). More implementation details can be found in [34]. Through this way, we can decompose the Ω and M into n subregions that $\Omega = \{\Omega_1, \Omega_2, \ldots, \Omega_n\}$, $M = \{M_1, M_2, \ldots, M_n\}$. For each $\phi_i : \Omega_i \to M_i$, we can compute it using the above surface mapping algorithm for single volumetric domain.

4.2 Normal Derivative Constrains by Affine Approximation

In this section, we give an algorithm to compute the normal derivative function g by using affine approximation.

When we handle heterogeneous and decomposed volume data, we want to enforce higher continuity across the cutting interface. Since any directional derivative can be decomposed into the tangent derivative and normal derivative, so C^1 continuity consists of tangent derivative continuity and normal derivative continuity. From differential geometry, we know tangent derivative continuity can be ensured by the consistence of positional constraints. Then, we need to choose a function g as the normal derivative function such that the derivative of the mapping will not change when it goes across the subregion boundaries.

Here, we adopt an affine approximation method to set normal derivative boundary condition g for constraint points. We build an affine transformation matrix for each constraint point. Then, we define normal derivative function g on this point according to this local affine function. Given the surface mapping $f : \partial\Omega \to \partial M$. For a vertex $x \in \partial\Omega$, whose one-ring neighboring vertices are $x_1, x_2, \ldots, x_n$ (Fig. 2), we compute a local affine function $\Phi_x(x) = A_x \times x + D_x, x \in \partial\Omega$. The global mapping is $\Phi(x) = \{\Phi_x(x)\}$ for $x \in \partial\Omega$. So, the normal derivative boundary

condition of the biharmonic function Φ is given as $\frac{\partial \Phi(x)}{\partial n} = A_x^T \times n$ on point x.

The A_x, D_x on each point $x \in \partial\Omega$ can be computed from the surface mapping on the one-ring region in the following linear system:

$$\begin{cases} f(x_1) = A_x \times x_1 + D_x \\ f(x_2) = A_x \times x_2 + D_x \\ \quad\quad ... \\ f(x_n) = A_x \times x_n + D_x, \end{cases} \tag{3}$$

where A_x is 3×3 and D_x is 3×1. We solve this system using the least square method to get A_x, D_x. If the affine transformation is degenerated, e.g., a planar local region is transformed into another planar region, the rank of the coefficient matrix of the system (3) reduces to 3 and the linear system becomes under-determined. We still compute A_x, D_x that are the least squares solutions and have the smallest L_2 norm.

For a point x along the boundary interface, no matter which volumetric region it belongs to, it will be mapped to a same target point $f(x)$. According to our computation of $\Phi_x(x)$, the point from different volumetric regions will get the same value of A_x, D_x. That is, it will have the same normal derivative function g_x. So, we can see the point along the boundary interface, it not only keeps position consistence through boundary surface mapping, but also has normal derivative consistency. We also show in Section 5 that this biharmonic volumetric mapping keeps linear precision property.

5 Properties, Implementation, and Evaluations

In this section, we discuss properties, implementation details, and evaluations of our volumetric mapping.

5.1 Linear Precision Property

A function ϕ has the *linear precision* property if it can reproduce a linear function exactly: given a set of function values of $\phi(v_i) = r(v_i), v_i \in \partial\Omega$ for any linear function r, then $\phi(x) = r(x)$, $x \in \Omega$ [35].

Linear precision property is desirable in describing shape deformation [20]. For example, when the boundary surface transforms rigidly, it will be natural to see the interior region also transforms in the same rigid manner.

Linear precision of harmonic maps. Suppose ϕ is a harmonic function, i.e., it is a solution to Laplace's equation: $\nabla^2\phi(x) = 0, x \in \Omega$, and satisfies the boundary condition $\phi(v_i) = r(v_i), v_i \in \partial\Omega$, where $r(x)$ is a linear function. Then, since ϕ and r satisfy the same boundary conditions and are both solutions to Laplace's equation (the linear function r is also a harmonic function), by the uniqueness of solutions to Laplace's equation, they must be the same function $\phi \equiv r$. So the harmonic mapping ϕ keeps the linear precision property [36].

Linear precision of our biharmonic maps. Whether biharmonic mapping has the linear precision property is determined by the setting of its boundary condition. When its boundary positional constraints are decided by a linear function, using our normal derivative setting discussed in the last section, the computed biharmonic map has the linear precision property.

Suppose the surface mapping f is a global linear transformation $\phi(x) = r(x) = Ax + D, x \in \partial\Omega$. The local linear function $\phi_x(x) = A_x \times x + D_x$ where A_x and D_x are computed from its one-ring transformation, and we have $\phi_x(x) = \phi(x) = Ax + D$ (i.e., $A_x = A, D_x = D$). The normal derivative function

$$\frac{\partial \phi(x)}{\partial n} = g = \{g_x | x \in \partial\Omega\} = \left\{A_x^T \times n | x \in \partial\Omega\right\} = A^T \times n.$$

While $\frac{\partial r(x)}{\partial n} = A^T \times n$. So, we have $\frac{\partial \phi(x)}{\partial n} \equiv \frac{\partial r(x)}{\partial n}$ and $\phi(x) \equiv r(x), x \in \partial\Omega$. Also, ϕ, r are both solutions to biharmonic's equation ($\Delta^2 r(x) = 0$, since r is a linear function). According to the uniqueness solution of biharmonic's equation [37], we have $\phi \equiv r$. Therefore, this biharmonic mapping keeps the linear precision property.

5.2 Mapping Bijectivity

Bijectivity should usually be ensured in mapping computation. For general given shapes, finding volumetric mapping with guaranteed bijectivity is usually very difficult. To our best knowledge, mapping construction algorithms with guaranteed bijectivity are only known on simple shapes such as convex or star regions. Our biharmonic model, like the harmonic map, cannot guarantee the mapping bijectivity when the two given models are general shapes. However, our idea of processing complicated models through decomposition and divide and conquer is one potential direction to avoid the degeneracy/flipover caused by the model's complex topology/geometry. The shape will be decomposed into subparts with simple geometry, whose parameterization is more often bijective.

In practice, we can check the bijectivity of a parameterization by computing the Jacobian value on each tetrahedron or hexahedron (after remeshing). If all Jacobian values are larger than 0, then this piecewise linear mapping function is bijective. For regions with negative Jacobian, it is also possible to develop heuristic adaptive decomposition to further partition these regions. In our experiments, we have performed this Jacobian evaluation, and observed that following our proposed boundary condition, the resultant biharmonic mapping remains bijective. Figs. 8d and 8h list our computed Jacobian statistics.

5.3 Other Boundary Condition Setting Strategies

A unique biharmonic mapping can be specified by the positional constraint function f and normal derivative function g. The position constraints are decided by the surface mapping f between boundaries of the source Ω and the boundaries of target model M.

A harmonic intersurface map is a simple choice for f. We can analyze its continuity across the boundary of adjacent subparts. Any directional derivative can be expressed by the combination of the normal derivative $\frac{\partial}{\partial n}$ and two tangential derivatives ($\frac{\partial}{\partial \tau_1}$, $\frac{\partial}{\partial \tau_2}$) with the absolute value of coefficients less than 1:

$$\frac{\partial}{\partial d} = cos\ \alpha \frac{\partial}{\partial n} + cos\ \beta \frac{\partial}{\partial \tau_1} + cos\ \gamma \frac{\partial}{\partial \tau_2}, \tag{4}$$

where the direction cosine of d is $(cos\alpha, cos\beta, cos\gamma)$ along the direction n, τ_1, τ_2 and $cos\alpha^2 + cos\beta^2 + cos\gamma^2 = 1$.

Using harmonic surface mapping, the only discontinuity issue could appear on the segmentation curves on the boundary surface. On each such curve c shared by two adjacent regions, the harmonic surface mapping can guarantee C^1 continuity on the normal direction n and the tangential direction τ_1 along c. Therefore, the difference of the any directional derivative from the adjacent domains is bounded by the difference of the tangential derivatives $\frac{\partial}{\partial \tau_2}$ resulting from the separately computed surface harmonic maps. This term is usually quite small. In most of our experiments, $|\frac{\partial f_1}{\partial \tau_2} - \frac{\partial f_2}{\partial \tau_2}| < 0.02$.

Using biharmonic surface mapping with a carefully developed boundary condition will provide first-order smoothness along the cutting curve on the surface boundary. However, solving a fourth-order biharmonic equation on the surface is much more expensive. Harmonic mapping is also biharmonic and provides relatively good boundary condition, so we simply use it for boundary positional constraint.

After fixing the surface mapping f, then each given derivative function g will indicate a unique specific biharmonic mapping. Besides using our linear approximation method to decide function g, here we also explain two other ways to construct g.

Harmonic-based boundary condition. This boundary condition first needs to compute the harmonic volumetric mapping in each region. Then, it computes each biharmonic volumetric mapping with g equals to the average value of the normal derivative of its neighboring harmonic mappings.

We can see in this case if there is only one region, the resultant biharmonic mapping will be exactly the harmonic mapping. If there are more than one region, the intuition behind it is to both capture the harmonic mapping's good properties and improve the continuity along the boundary interface. We can verify that this boundary condition has C^1 continuity and also keeps linear precision property (See Appendix, available in the online supplemental material, for details).

The main disadvantage of this boundary condition setting is its increased computation complexity, since the computation of the g is based on the harmonic results. The total computational time doubles.

Clamped boundary condition. Another natural yet simple setting is to have $g = 0$. Setting the normal derivative function to zeros gives an implicit tangent boundary condition. This is different from the harmonic solution which has tangent discontinuity. We can see this boundary condition also satisfies $\frac{\partial \Phi_1(x)}{\partial n_1(x)} = \frac{\partial \Phi_2(x)}{\partial n_1(x)} = 0$, when $x \in \partial\Omega_{12}$. So, it has C^1 continuity along the boundary interface. But, it does not keep linear precision property, because $\frac{\partial \Phi_1(x)}{\partial n_1(x)} = 0$ which is usually inequivalent to $\frac{\partial r(x)}{\partial n} = A^T \times n$.

5.4 Measuring Mapping Distortion

Our computation algorithm does not depend on a tessellation of the volumetric region Ω and it has closed form. In order to measure the mapping distortion, we implement a metric on the tesselated tetrahedral mesh of Ω (without ambiguity, we also denote this tet-mesh as Ω). We evaluate the mapping $\phi(v_i)$ for each vertex $v_i \in \Omega$, while linearly interpolate the mapping inside each tetrahedron. Considering the Jacobian of the transformation defined on each tetrahedron, we can measure its condition number. As suggested in [38], such a condition number is an indicator of the Jacobian and is invariant to scale and rotation.

Suppose a tetrahedron T consists of four vertices $v_n, n = 0, 1, 2, 3$ with coordinates $X_n \in \mathcal{R}^3$. Define edge vectors $e_{k,n} = X_k - X_n$ with $k \neq n$ and $k = 0, 1, 2, 3$. Vertex v_n has three incident edge vectors, $e_{n+1,n}, e_{n+2,n}$ and $e_{n+3,n}$, where all the indices are modulo four. The Jacobian matrix at node n, denoted by $M_{T,n}$, is composed of the columns of these edge vectors, namely

$$M_{T,n} = (-1)^n (e_{n+1,n}, e_{n+2,n}, e_{n+3,n}).$$

Suppose under the mapping, the original tetrahedron deforms to a new tetrahedron T', whose corresponding matrix is $\mathbf{M_{T',n}}$. Consider the matrix $\mathbf{S_n}$ that transforms $\mathbf{M_{T,n}}$ to $\mathbf{M_{T',n}}$. Then, $\mathbf{S_n} = \mathbf{M_{T',n}} \cdot \mathbf{M_{T,n}^{-1}}$. We can verify that $\mathbf{S_n}$ is independent of n. Therefore, we write $\mathbf{S} = \mathbf{M_{T',n}} \cdot \mathbf{M_{T,n}^{-1}}$. Then, the condition number of a tetrahedron is

$$k(\mathbf{S}) = |\mathbf{S}||\mathbf{S^{-1}}|,$$

where $\mathbf{M_{T,n}}$ is invertible when T has the positive volume. $k(\mathbf{S})$ measures the condition number of the transformation between the original and mapped tetrahedron. We adopt the Euclidean norm of $\mathbf{S}$: $|\mathbf{S}| = [tr(\mathbf{S^T S})]^{1/2}$, then $k(\mathbf{S}) \geq 3$ where in the optimal case $k(\mathbf{I}) = |\mathbf{I}||\mathbf{I^{-1}}| = 3$. We use $k(\mathbf{S})$ to evaluate the volumetric mapping computed on tetrahedral meshes.

Fig. 3 shows a comparison on the polycube parameterization of Max-Planck model (also see Figs. 8a and 8b), using harmonic and biharmonic mappings. From the distributions of the tetrahedral condition numbers shown in Figs. 3 a and 3b, we can see that the biharmonic mapping and harmonic mapping have similar stretching distortion. But for the boundary elements, the biharmonic model introduces less stretching in their transformations. This is visualized in Figs. 3c and 3d, where the boundary tetrahedra deformed from polycube domain under biharmonic mapping are less stretched than the deformation guided by the harmonic map.

5.5 Improving Computation Efficiency

In Section 3.2, the boundary fitting for ϕ is formulated as solving a linear system $Aw = t$ and it is solved by using SVD. However, SVD decomposition is slow for large matrices. To handle complex volumetric data, we have to restrict the size of constraints points and singularity points N_C and N_S. This may significantly reduce the boundary fitting accuracy: either we lack enough particles (when N_S is small) for designing fine fields to fit the boundary condition, or we lack enough constraint points (when N_C is small) to sample and reflect the shape variance on the boundary. In [17], the truncated SVD is used to avoid overfitting and improve the numerical stability and efficiency. The computation of SVD is still very expensive for big solid models.

To improve the stability (overcome the singularity of the linear system and avoid overfitting) and improve the

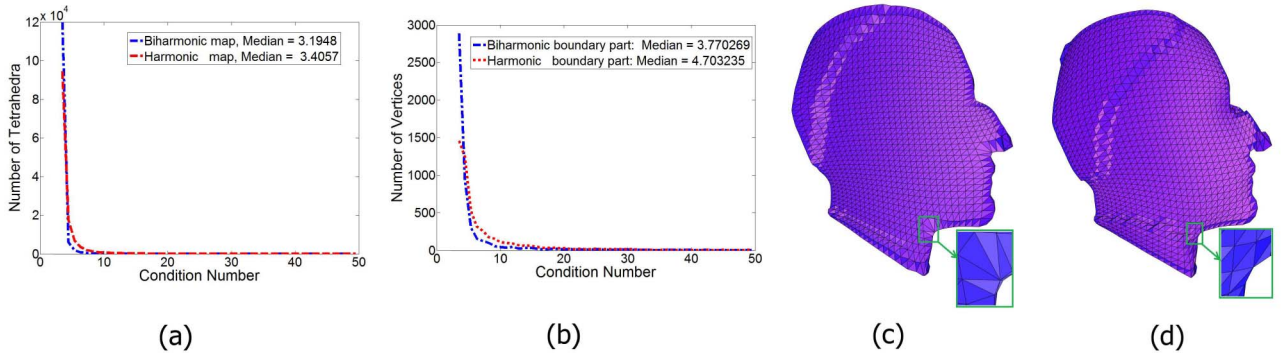


Fig. 3. (a) The tetrahedral condition number distributions of the harmonic and biharmonic maps. (b) The tetrahedral condition number on the boundary elements. (c, d) Polycube parameterization of the Max-Planck model using harmonic and biharmonic mappings, (c) shows the deformed tetrahedra mesh under the harmonic map, while (d) shows the deformed tetrahedra mesh under the biharmonic map.

efficiency of the boundary fitting, we apply an additional regularization term to impose penalty on the norm of coefficients of singularity points, which is also known as ridge regression [39],

$$E_3 = \sum_{j=1}^{N_S} h_j^2 + \sum_{j=1}^{N_S} b_j^2.$$

The new boundary fitting is formulated as minimizing

$$E = E_1 + E_2 + \alpha E_3, \tag{5}$$

where

$$E_1 = \sum_{i=1}^{N_C} \left| \sum_{j=1}^{N_S} \big(h_j H(\mathbf{q}_j, \mathbf{p}_i) + b_j B(\mathbf{q}_j, \mathbf{p}_i)\big) - f(\mathbf{p}_i) \right|^2,$$

$$E_2 = \sum_{i=1}^{N_C} \left| \sum_{j=1}^{N_S} (h_j \nabla H \cdot n(\mathbf{p}_i) + b_j \nabla B \cdot n(\mathbf{p}_i)) - g(\mathbf{p}_i) \right|^2,$$

and $\{\mathbf{p}_1, \ldots, \mathbf{p}_{N_C}\}$ are a set of constraint points sampled on the domain boundary $\mathbf{p}_i \in \partial\Omega$. $\alpha > 0$ is a parameter which controls the weight balance of boundary fitting and regularization. Note that (5) is still quadratic, which could be denoted as $E = \frac{1}{2}\mathbf{x}^{\mathrm{T}} A\mathbf{x} + \mathbf{b}^{\mathrm{T}}\mathbf{x} + c$, where the positive definite matrix A can be preassembled. With an appropriate α, the minimization of this quadratic energy E can be efficiently solved using LU-decomposition. Compared with SVD, this will significantly improve the computational efficiency and numerical stability. Therefore, this framework can now handle big volumetric data more efficiently.

To pick a suitable parameter α, we first show that α is related to the condition number of the coefficient matrix A. The numerical stability of the system depends on the condition number of the coefficient matrix. Smaller condition number indicates better numerical stability. Suppose we set a threshold value K for the condition number of A, namely, to make $cond(A) \le K$. Then, from (5), $E = E_1 + E_2 + \alpha E_3$, the coefficient matrix A also consists of three terms $A = A_1 + A_2 + \alpha A_3$ where the semidefinite matrix A_1 comes from E_1, semidefinite matrix A_2 comes from E_2, and diagonal matrix A_3 comes from E_3.

We can compute the singular value of $A_1 + A_2$. Let σ_{max} be the maximum singular value of $A_1 + A_2$ and its minimum singular value is 0 (since A_1 and A_2 are underdetermined). Also, the singular value of A_3 is 1. So, the condition number of the matrix A is $cond(A) = \frac{\alpha + \sigma_{max}}{\alpha}$. From $cond(A) \le K$, it has $\frac{\alpha+\sigma_{max}}{\alpha} \le K$ and $\frac{\sigma_{max}}{K-1} \le \alpha$.

Second, according to [40], if we solve the linear system $Ax = b$ by LU decomposition, and the elements of A and b are accurate up to s decimal places to the left of the decimal point ($s \approx 10^{-13}$ based on IEEE 754 float type) and $cond(A) \approx 10^t$, where $t \le s$, then the computed solution is accurate to about $s - t$ decimal places to the left of the decimal point.

So, we set the threshold value of condition number K based on the desired accuracy of the solution, then calculate α from such K. In our experiments, we take $K \approx O(10^6)$, then the computed solution is with $O(10^{-7})$ accuracy. For the spherical mapping of the Omotondo model, we have $\sigma_{max} = 36,825.09$ and set $K = 2.5 * 10^6$, then $\alpha \ge \frac{36,825.09}{2.5*10^6-1} \approx 0.015$. Similar computation can be applied in other models.

We test this on several models and show the side-by-side comparison on their running time and boundary fitting error, using same numbers/positions of source and constraint points. And the *boundary fitting error* is the average squared distances $|f(\mathbf{p}_i) - \Phi(\mathbf{p}_i)|^2, \mathbf{p}_i \in \partial\Omega$ between the target boundary points and the images of boundary points under the mapping. The results are shown in Table 1, which shows the improvement on both the computation efficiency and fitting accuracy.

6 Experimental Results

We implement our mapping computation in C++ and perform experiments on a 3 GHz Pentium-IV PC with 4 G RAM. Our experimental data include the heterogeneous data and decomposed volume data.

6.1 Mapping Heterogeneous Volume Data

Fig. 4 illustrates an example of multiple-layered volumetric model. We try to parameterize a head model (Fig. 4b) onto a cube domain Ω (Fig. 4a), with the brain region and the other layers being handled separately: we want the brain region $M_1 \subset M$ inside the head to be mapped to the smaller cube region Ω_1 inside Ω, and we denote the region outside the brain as $M_2 = M \backslash M_1$, whose corresponding parametric domain is $\Omega_2 = \Omega \backslash \Omega_1$. If we consider the volumetric mapping

TABLE 1
Statistics Comparison between the SVD Solver [17] and the New Solver

Model		LU		SVD	
Model Name	N_Ω	Time	Boundary Fitting Error	Time	Boundary Fitting Error
Omotondo/Sphere	3002	54.66	0.4308165e-3	555.31	0.13626636e-2
PCube/2-Torus	6622	388.32	0.2532e-3	3626.03	0.1e-2
Sphere/Max-Planck	9002	946.08	0.111371e-4	8568.29	0.1890862e-3
Sphere/Igea	15002	469	0.46195e-5	1180	0.8689456e-3

The constraint point ratio N_C/N_Ω and source point ratio N_s/N_Ω are 0.4 and 0.8, respectively, where N_Ω is the number of vertices on the source boundary surface $\partial\Omega$. The computation time is measured in seconds.

on both M_1 and M_2 : $\Phi_1 : \Omega_1 \rightarrow M_1$ and $\Phi_2 : \Omega_2 \rightarrow M_2$, naturally, along the brain cortex isosurface $\partial\Omega_1$, we want the mappings Φ_1 and Φ_2 to be smooth and obtain not only the positional continuity but also derivative consistency. If the harmonic volumetric mapping is used to handle both regions separately, then the derivative transition along the isosurface is not smooth, as indicated by the parametric coordinates in Fig. 4e. When we use the biharmonic volumetric mapping, we can guarantee the nice derivative smoothness, as shown in Fig. 4f. The derivative discontinuity $\delta_n = |\frac{\partial\Phi_1}{\partial n} - \frac{\partial\Phi_2}{\partial n}|, x \in \partial\Omega_1$ is computed and color-encoded in Figs. 4c and 4d to show numerically the mapping smoothness along the cutting boundary. Through the side-by-side comparison, biharmonic mapping demonstrates much better smoothness. We also use the color-encoded distance field to visualize the mapping result. When a map $\Phi : \Omega \rightarrow M$ is computed, the color-encoded (red indicates the maximum while blue indicates the minimum) distance field defined on one object can be transferred to the other, by plotting the color of a point $\mathbf{P} \in \Omega$ on its corresponding image $\phi(\mathbf{P}) \in M$ (or inversely, plotting the color of $\mathbf{P} \in M$ on $\phi(\mathbf{P}) \in \Omega$). We visualize the biharmonic volumetric mapping result in Figs. 4g, 4h, and 4i

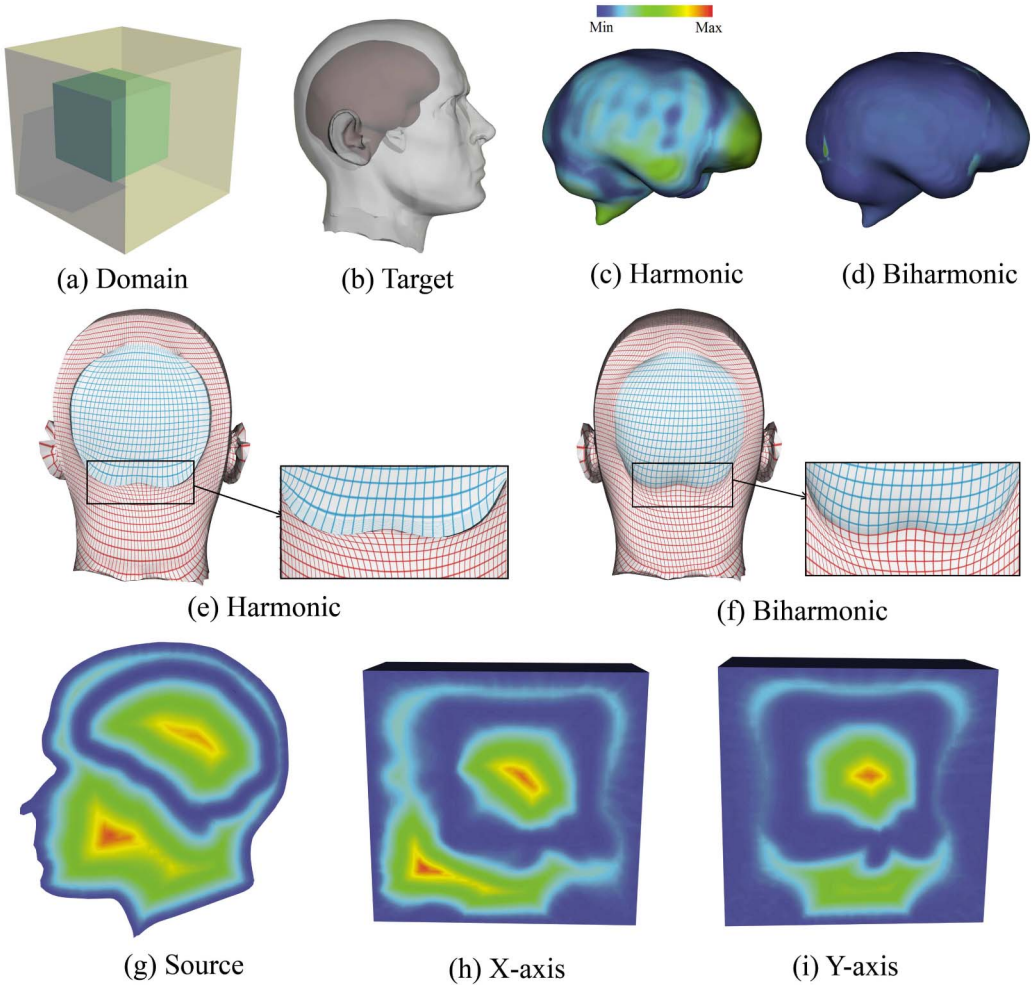


Fig. 4. Parameterization of an heterogeneous head model onto a cube domain (surface vertex number is 20k). (a) The cube domain, (b) the head model, with the brain region to be mapped onto the interior cube in (a). The derivative discontinuity ($\delta_n = |\frac{\partial\Phi_1}{\partial n} - \frac{\partial\Phi_2}{\partial n}|, x \in \partial\Omega_1$) of the harmonic mapping (c) versus biharmonic mapping (d). Mapping distortion and boundary smoothness are also illustrated through parametric coordinates in (e) and (f), shown from a cross section along Y-axis. (g-i) visualize the biharmonic mapping distribution using the transferred distance field: (g) the original distance field; (h,i) the transferred fields in x and y directions.

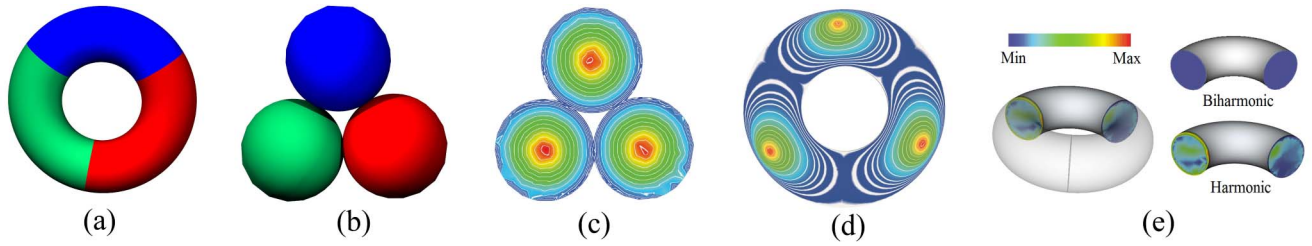


Fig. 5. Biharmonic mapping on decomposed models. (a) Decomposition of a solid torus model. (b) A corresponding parametric domain. (c) Color-encoded distance field on the parametric domain. (d) Transferred distance field under the mapping. (e) Boundary derivative errors δ_n under both harmonic and biharmonic mapping indicates the biharmonic mapping leads to smooth derivative transition.

by the transferred distance field from head/brain to the cube. Fig. 4g shows the distance field defined on the head/brain domain while the transferred distance field is shown in Figs. 4h and 4i from cross sections in x and y-directions.

6.2 Mapping Decomposed Volume Data

Figs. 5 and 6 show our biharmonic mapping computation applied on decomposed Torus and Rocker-arm models. The decomposition of the torus and Rocker-arm are illustrated using colors in (a). The corresponding target shape is shown in (b). We parameterize each subregion of torus onto a convex cell (as indicated by the corresponding color). Adjacent subregions share a surface interface and we enforce the positional consistency as well as the derivative constraint g with local linear approximation setting across the boundary. The color-encoded distance field computed on each convex shape (c) is transferred into the rocker-arm (d) (i.e., the color of each point $\mathbf{x}$ on Ω is rendered on $\Phi(\mathbf{x}) \in M$) to visualize the mapping result. Figs. 5e and 6e illustrate the normal derivative discontinuity δ_n along the cutting boundary. Compared with harmonic mapping, our biharmonic mapping with derivative boundary condition enforced along the cutting boundary brings a parameterization with smooth derivative transition across the boundary.

7 Applications

In this section, we apply our model in the hex-remeshing and 3D dynamic temporal data registration.

7.1 Hex-Remeshing

The hex-remeshing converts the tessellation of a volumetric model from a tetrahedral mesh to a hexahedral mesh. An effective hexahedral remeshing is desirable in many scientific and engineering tasks such as finite element simulation and isogeometric analysis, because a high-quality hexahedral mesh can significantly facilitate the computation and analysis in many of these tasks.

We apply our biharmonic volumetric mapping to hex-remeshing through the polycube domain. A polycube domain is a shape glued by a lot of small regular cubes [41]. The basic idea is to compute the mapping between a solid model M and a solid polycube domain Ω. Then, on Ω, we can sample points regularly and obtain a perfect hexahedral mesh, then with the mapping this hexahedral mesh can be transformed into the geometry of M. If the mapping ϕ has very small distortion, the generated hexahedral mesh for M has a very good quality.

In our experiment, we construct the polycube domain and obtain the surface mapping between the model and the polycube boundary using the method introduced in [33], and then compute the volumetric mapping use the algorithm proposed in this paper. We also compare our result with that generated through harmonic volumetric mapping suggested in [24].

Figs. 4e and 4f show an example of hex-remeshing of heterogeneous volumetric models. The side-by-side comparison shows that compared with the harmonic mapping, the biharmonic mapping generates a smoother and therefore more desirable hexahedral mesh.

Fig. 7 shows the parameterization of the Isis model M (b) to the polycube domain Ω (a). M is decomposed into two subparts: $M = \{M_1, M_2\}$, where M_1 is mapped to the upper cube and M_2 is mapped to the lower cube. The transferred hex-meshes from the polycube to Isis model are shown in (c) and (d), based on the harmonic and biharmonic volumetric mappings, respectively. We can see the biharmonic mapping provides significantly better smoothness along the cutting boundary.

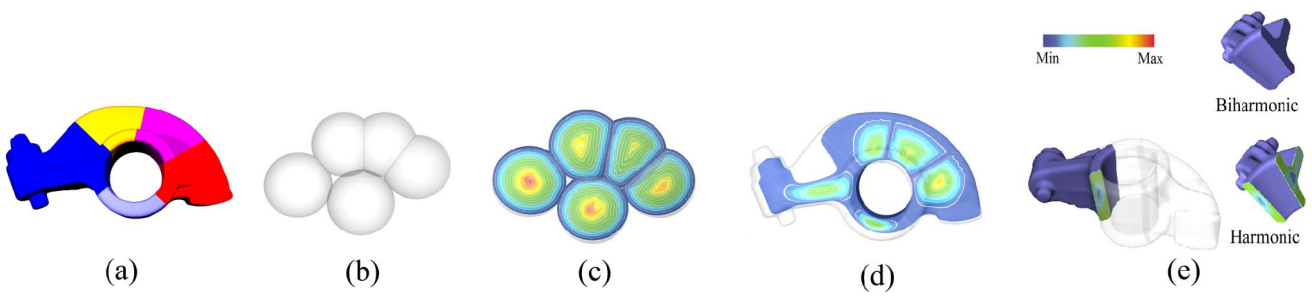


Fig. 6. Biharmonic mapping on decomposed models. (a) Decomposition of a solid rocker-arm model. (b) A corresponding parametric domain. (c) Color-encoded distance field on the parametric domain. (d) Transferred distance field under the mapping. (e) Boundary derivative errors δ_n under both harmonic and biharmonic mapping indicate that the biharmonic mapping leads to smooth derivative transition.

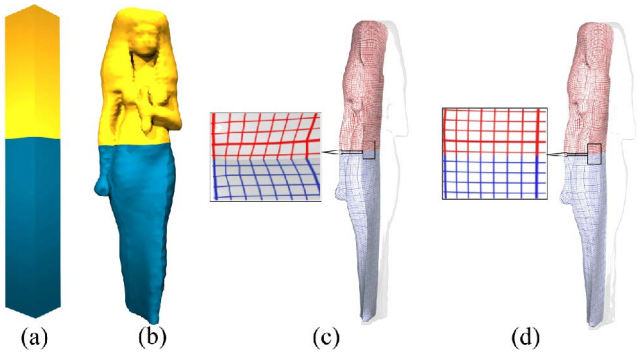


Fig. 7. Biharmonic mapping on Isis decomposed model (surface vertex number is 20k). (a) The polycube domain. (b) The corresponding Isis parametric domain. The regular hex-structure on the cubes transferred onto the Isis model using (c) harmonic mapping and (d) biharmonic mapping.

We show some more biharmonic polycube hex-remeshing results in Fig. 8. We parameterize a Max planck's head model (b) onto a unit cube (a), then the remeshed model is shown in (c). The hex-remeshing of a Chinese-horse model is shown in (e-g). We use the scaled Jacobian metric [42] which has a range $[-1, 1]$ with 1 being optimal to measure the quality of the generated mesh. The distribution of the Jacobian value in (d,h) and our results are comparable to the paper [43].

7.2 Consistent Parameterization of Temporal Data

Another application of our biharmonic volumetric mapping is registration. Registration has been a ubiquitous technique, which is widely used in many applications in computer vision, computational medicine, etc. An accurate registration indicates the natural differences between models in a quantitative way, and benefits the subsequent analysis tasks. Here, we use a medical example to demonstrate the usage of biharmonic volumetric mappings in 3D dynamic temporal registration.

Our experiment is conducted on the temporally scanned lung data collected during multiple respiratory cycles of some patient having the lung tumor. The goal is to accurately register the deforming volumetric region, so that the tumor's deformation (and its related motion) can be described, analyzed, and used to guide the subsequent radiation (in which the beam should correctly target the tumor without touching surrounding normal organs or tissues) [44].

Fig. 9. Vessels near the tumor.

Due to the natural heterogeneous property of the data we need to handle, the model decomposition is desirable. Also it is necessary to preserve the C^1 continuity across the boundary of lung and tumor domains. For example, for vessels that are surrounding organs and tissues, as shown in Fig. 9, when we segment the organ from the surrounding environment, these vessels are also cut apart across the partitioning boundary. Transformations (mappings) inside and outside the partitioning boundary are computed separately but their transitions shall naturally be smooth.

Fig. 10 shows some registration/matching results computed using biharmonic mapping. In (a), the volumetric models are extracted from three different time frames during one respiratory cycle. We parameterize all these data onto a common sphere domain for the consistent parameterization. To analyze the motion and deformation of the tumor region (red), it shall be mapped onto the red small sphere, while the left region of its surrounding lung tissue shall be mapped to the left outside region. Then, we compute our biharmonic model on these two subparts separately while preserving C^1 continuity along the isosurface. The mapping results are visualized using the transferred distance field from the canonical domain (b) to each model shown in (c). With the consistent parameterization over the canonical sphere, the registration between any pair of the models can be computed immediately, as shown

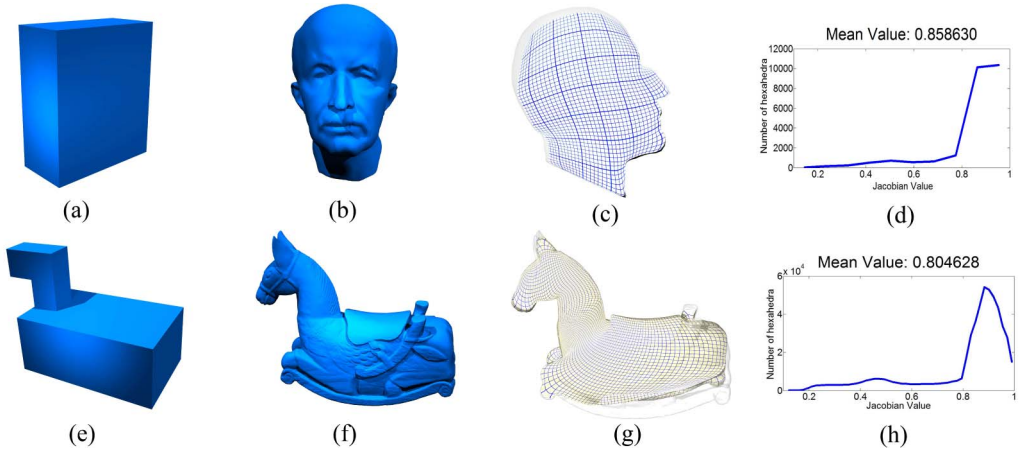


Fig. 8. Biharmonic polycube mapping for hex-remeshing. A cube model (surface vertex number: 12.5k) (a) is used to parameterize the Max Planck's head model (b), and the resultant hexahedral meshing of the Max Planck model is shown in (c). A polycube model (surface vertex number: 100k) (e) is used to parameterize the Chinese horse model (f), and the resultant hexahedral mesh is shown in (g). (d) and (h) show the distributions of Jacobian values of the deformed hexahedra.

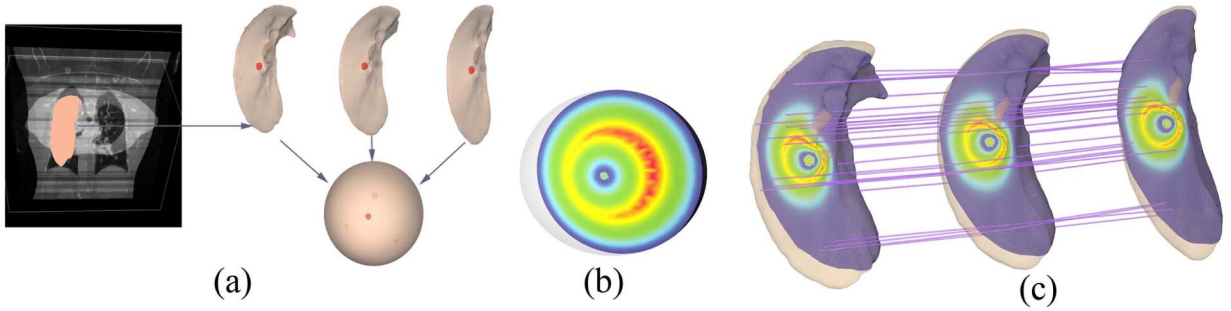


Fig. 10. (a) The deforming lung/tumor models and their maps to the sphere domain; (b) the transferred distance field from one lung model to the sphere domain; (c) the induced correspondence among deforming lung/tumor models.

in (c). The motion and deformation of the entire lung region can then be depicted on the sphere domain, and used to predict the tumor's trajectory.

8 Conclusions

We propose a biharmoinc volumetric mapping computation framework using the fundamental solution method. Compared with harmonic mapping, the biharmonic mapping allows better boundary control. In a divide-and-conquer computation framework for mapping huge, complex, or heterogeneous volumetric models, this biharmonic model is desirable because it can provide nice smoothness across the cutting boundary. Our proposed boundary derivative setting algorithm can ensure the linear precision property of our biharmonic transformation. Compared with harmonic mapping based on fundamental solutions [24], [17], our new solving scheme is also more accurate and efficient.

One limitation of our current volumetric mapping framework is that this mapping result depends on the boundary surface mapping. Intuitively, the volumetric mapping and its boundary surface mapping are closely related to each other. Smoother boundary surface mapping will give us better volumetric mapping. However, instead of computing surface mapping first, directly solving the volumetric mapping could be more natural in some scenarios. We will explore along this direction in the near future.

We will also explore the computation of biharmonic inter-surface mapping, and the numerical improvement of the computational efficiency and accuracy. Furthermore, we will explore other applications of biharmonic volumetric mapping.

Acknowledgments

This work was partially supported by Louisiana Board of Regents (RCS) LEQSF(2009-12)-RD-A-06 and LA-BOR PFund:NSF(2011)- PFund-236. Huanhuan Xu was supported in part by the Mark and Carolyn Guidry doctoral fellowship from Division of Electrical and Computer Engineering of LSU.

References

[1] J. Monterde and H. Ugail, "A General Fourth-Order PDE Method to Generate bézier Surfaces from the Boundary," *Computer Aided Geometry Design,* vol. 23, no. 2, pp. 208-225, 2006.

[2] A. Jacobson, E. Tosun, O. Sorkine, and D. Zorin, "Mixed Finite Elements for Variational Surface Modeling," *Computer Graphics Forum,* vol. 29, no. 5, pp. 1565-1574, 2010.

[3] L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel, "Interactive Multi-Resolution Modeling on Arbitrary Meshes," *SIG '98: Proc. 25th Conf. Computer Graphics and Interactive Techniques,* pp. 105-114, 1998.

[4] B.T. Helenbrook, "Mesh Deformation Using the Biharmonic Operator," *Int'l J. Numerical Methods in Eng.,* vol. 56, no. 7, pp. 1-11, 2003.

[5] A. Jacobson, I. Baran, J. Popović, and O. Sorkine, "Bounded Biharmonic Weights for Real-Time Deformation," *ACM Trans. Graphics,* vol. 30, no. 4, article 78, 2011.

[6] Y. Lipman, R.M. Rustamov, and T.A. Funkhouser, "Biharmonic Distance," *ACM Trans. Graphics,* vol. 29, no. 3, pp. 1-11, 2010.

[7] R. Tankelevich, G. Fairweather, and A. Karageorghis, "Three-Dimensional Image Reconstruction Using the PF/MFS Technique," *Eng. Analysis with Boundary Elements,* vol. 33, no. 12, pp. 1403-1410, 2009.

[8] C.C.L. Wang, K.-C. Hui, and K.-M. Tong, "Volume Parameterization for Design Automation of Customized Free-Form Products," *IEEE Trans. Automatic Science and Eng.,* vol. 4, no. 1, pp. 11-21, Jan. 2007.

[9] S. Han, J. Xia, and Y. He, "Hexahedral Shell mesh Construction via Volumetric Polycube Map," *Proc. ACM Symp. Solid and Physical Modeling,* pp. 127-136, 2010.

[10] J. Xia, Y. He, X. Yin, S. Han, and X. Gu, "Direct-Product Volumetric Parameterization of Handlebodies via Harmonic Fields," *Proc. Int'l Conf. Shape Modeling and Applications,* pp. 3-12, 2010.

[11] A.A. Joshi, D.W. Shattuck, P.M. Thompson, and R.M. Leahy, "Surface-Constrained Volumetric Brain Registration Using Harmonic Mappings," *IEEE Trans. Medical Imaging,* vol. 26, no. 12, pp. 1657-1669, Dec. 2007.

[12] Y. Wang, X. Gu, T.F. Chan, P.M. Thompson, and S.T. Yau, "Volumetric Harmonic Brain Mapping," *Proc. IEEE Int'l Symp. Biomedical Imaging: Macro to Nano,* pp. 1275-1278, 2004.

[13] X. Li, X. Guo, H. Wang, Y. He, X. Gu, and H. Qin, "Harmonic Volumetric Mapping for Solid Modeling Applications," *Proc. ACM Symp. Solid and Physical Modeling,* pp. 109-120, 2007.

[14] T. Martin, E. Cohen, and R.M. Kirby, "Volumetric Parameterization and Trivariate B-Spline Fitting Using Harmonic Functions," *Proc. ACM Symp. Solid and Physical Modeling,* pp. 269-280, 2008.

[15] T. Martin and E. Cohen, "Volumetric Parameterization of Complex Objects by Respecting Multiple Materials," *Computer Graphics,* vol. 34, no. 3, pp. 187-197, 2010.

[16] B. Li, X. Li, K. Wang, and H. Qin, "Generalized Polycube Trivariate Splines," *Proc. Int'l Conf. Shape Modeling and Applications,* pp. 261-265, 2010.

[17] X. Li, H. Xu, S. Wan, Z. Yin, and W. Yu, "Feature-Aligned Harmonic Volumetric Mapping Using MFS," *Computers and Graphics,* vol. 34, no. 3, pp. 242-251, 2010.

[18] T. Ju, S. Schaefer, and J.D. Warren, "Mean Value Coordinates for Closed Triangular Meshes," *Proc. ACM SIGGRAPH,* vol. 24, no. 3, pp. 561-566, 2005.

[19] M.S. Floater, "Mean Value Coordinates," *Computer Aided Geometric Design,* vol. 20, no. 1, pp. 19-27, 2003.

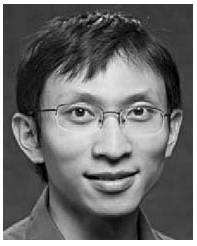
[20] P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki, "Harmonic Coordinates for Character Articulation," *Proc. ACM SIGGRAPH,* pp. 71-81, 2007.

[21] Y. Lipman, D. Levin, and D. Cohen-Or, "Green Coordinates," *ACM Trans. Graphics,* vol. 27, no. 3, pp. 1-10, 2008.
[22] G. Patanè, M. Spagnuolo, and B. Falcidieno, "Topology- and Error-Driven Extension of Scalar Functions from Surfaces to Volumes," *ACM Trans. Graphics,* vol. 29, pp. 4:1-4:20, 2009.
[23] A. Karageorghis and G. Fairweather, "The Method of Fundamental Solutions for the Numerical Solution of the Biharmonic Equation," *J. Computational Physics,* vol. 69, no. 2, pp. 434-459, 1987.
[24] X. Li, X. Guo, H. Wang, Y. He, X. Gu, and H. Qin, "Meshless Harmonic Volumetric Mapping Using Fundamental Solution Methods," *IEEE Trans. Automatic Science and Eng.,* vol. 6, no. 3, pp. 409-422, July 2009.
[25] B. Jin, "A Meshless Method for the Laplace and Biharmonic Equations Subjected to Noisy Boundary Data," *Computer Modeling in Eng. and Sciences,* vol. 6, no. 3, pp. 253-262, 2004.
[26] X. Li, X. Gu, and H. Qin, "Surface Mapping Using Consistent Pants Decomposition," *IEEE Trans. Visualization and Computer Graphics,* vol. 15, no. 4, pp. 558-571, July/Aug. 2009.
[27] T. KwoK, Y. Zhang, and C. Wang, "Efficient Optimization of Common Base Domains for Cross-Parameterization," *IEEE Trans. Visualization and Computer Graphics,* vol. 18, no. 10, pp. 1678-1692, Oct. 2012.
[28] X. Li, Y. Bao, X. Guo, M. Jin, X. Gu, and H. Qin, "Globally Optimal Surface Mapping for Surfaces with Arbitrary Topology," *IEEE Trans. Visualization and Computer Graphics,* vol. 14, no. 4, pp. 805-819, July/Aug. 2008.
[29] V. Kraevoy and A. Sheffer, "Cross-Parameterization and Compatible Remeshing of 3D Models," *Proc. ACM SIGGRAPH,* pp. 861-869, 2004.
[30] J. Schreiner, A. Asirvatham, E. Praun, and H. Hoppe, "Inter-Surface Mapping," *Proc. ACM SIGGRAPH,* pp. 870-877, 2004.
[31] A. Agathos, I. Pratikakis, S. Perantonis, N. Sapidis, and P. Azariadis, "3D Mesh Segmentation Methodologies for CAD Applications," *Computer-Aided Design and Applications,* vol. 4, nos. 1-6, pp. 827-841, 2007.
[32] A. Shamir, "A Survey on Mesh Segmentation Techniques," *Computer Graphics Forum,* vol. 27, no. 6, pp. 1539-1556, 2008.
[33] H. Wang, Y. He, X. Li, X. Gu, and H. Qin, "Polycube Splines," *Proc. ACM Symp. Solid and Physical Modeling,* pp. 241-251, 2007.
[34] W. Yu and X. Li, "Computing 3D Shape Guarding and Star Decomposition," *Computer Graphics Forum,* vol. 30, no. 7, pp. 2087-2096, 2011.
[35] L. Garcia-puente and F. Sottile, "Linear Precision for Parameteric Patches," *Advances Computational Math.,* vol. 33, pp. 191-214, 2010.
[36] P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki, "Harmonic Coordinates for Character Articulation," *ACM Trans. Graphics,* vol. 26, article 71, July 2007.
[37] L.C. Evans, *Partial Differential Equations.* Am. Math. Soc., 1998.
[38] L.A. Freitag and P.M. Knupp, "Tetrahedral Element Shape Optimization via the Jacobian Determinant and Condition Number," *Proc. Eighth Int'l Meshing Roundtable,* pp. 247-258, 1999.
[39] J. Wang, F. Wang, C. Zhang, H.C. Shen, and L. Quan, "Linear Neighborhood Propagation and Its Applications," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 9, pp. 1600-1615, Sept. 2009.
[40] D. Watkins, *Fundamentals of Matrix Computations.* Wiley Interscience, 2002.
[41] M. Tarini, K. Hormann, P. Cignoni, and C. Montani, "PolycubeMaps," *Proc. ACM SIGGRAPH,* pp. 853-860, 2004.
[42] P. Pebay, D. Thompson, J. Shepherd, P. Knupp, C. Lisle, V. Magnotta, and N. Grosland, "New Applications of the Verdict Library for Standardized Mesh Verification Pre, Post, and End-to-End Processing," *Proc. Int'l Meshing Roundtable,* pp. 535-552, 2007.
[43] J. Gregson, A. Sheffer, and E. Zhang, "All-Hex Mesh Generation via Volumetric Polycube Deformation," *Computer Graphics Forum,* vol. 30, no. 5, pp. 1407-1416, 2011.
[44] S.S. Iyengar, X. Li, H. Xu, S. Mukhopadhyay, N. Balakrishnan, A. Sawant, and P. Iyengar, "Toward More Precise Radiotherapy Treatment of Lung Tumors," *Computer,* vol. 45, no. 1, pp. 59-65, Jan. 2012.

Huanhuan Xu received the MSc degree from the University of Science Technology of China in 2006. She received the PhD degree from the School of Electrical Engineering and Computer Science, Louisiana State University. Her research interests include 3D computer graphics, geometric modeling, image processing, and pattern recognition.

Wuyi Yu received the BE degree in electrical engineering from Xiamen University, Xiamen, China. He is currently working toward the PhD degree at the School of Electrical Engineering and Computer Science, Louisiana State University. His research interests include geometric modeling, computer graphics, meshing, shape decomposition and analysis.

Shiyuan Gu received the BS degree from Nanjing University, China, the MS degree in computer science, and the PhD degree in mathematics from Louisiana State University. He is a postdoctoral researcher at the Argonne National Laboratory. His research interests include numerical analysis, image processing, computer graphics, scientific computing and visualization.



Xin Li received the BS degree in computer science from the University of Science and Technology of China, and the MS and PhD degrees in computer science from Stony Brook University (SUNY). He is an assistant professor at the School of Electrical Engineering and Computer Science, and Center for Computational and Technology, Louisiana State University. His research interests include geometric modeling and processing, and their applications in graphics, visualization, computational forensics, and computational medicine. He is a member of IEEE Computer Society. Visit http://www.ece.lsu.edu/xinli for more information.