Meshless Harmonic Volumetric Mapping Using Fundamental Solution Methods

Xin Li, *Member, IEEE*, Xiaohu Guo, *Member, IEEE*, Hongyu Wang, Ying He, Xianfeng Gu, *Life Member, IEEE*, and Hong Qin, *Member, IEEE*

***Abstract*—Harmonic volumetric mapping aims to establish a smooth bijective correspondence between two solid shapes with the same topology. In this paper, we develop an automatic meshless method for creating such a mapping between two given objects. With the shell surface mapping as the boundary condition, we first solve a linear system constructed by a boundary method called the *method of fundamental solution*, and then represent the mapping using a set of points with different weights in the vicinity of the shell of the given model. Our algorithm is a true meshless method (without the need of any specific meshing structure within the solid interior) and the behavior of the interior region is directly determined by the boundary, which can improve the computational efficiency and robustness significantly. Therefore, our algorithm can be applied to massive volume data sets with various geometric primitives and topological types. We demonstrate the utility and efficacy of our algorithm in information transfer, shape registration, deformation sequence analysis, tetrahedral remeshing, and solid texture synthesis.**

***Note to Practitioners*—Building a one-to-one smooth correspondence between two solid geometric models is a fundamental problem. It directly applies to shape registration: when we have two solid shapes matched, we can easily compute and visualize for analysis purpose where and how much do they differ. Another important application is generating regular connectivity for arbitrary solid objects. This is performed by transferring regular structure from a canonical object to our target object once the correspondence (mapping) is created. Given two solid models, we can automatically compute a physically natural mapping between them. This "physically natural" property is characterized by so called harmonicity, which measures the smoothness of the mapping function, and is represented by the harmonic energy. A map with minimized harmonic energy means it is smoothest and natural; and our algorithm gets such a mapping. We show the desirable property of our map using applications such as shape comparison, tetrahedral mesh regularization, and solid texture synthesis. These applications in turn demonstrate the strong potential of our volumetric mappings. Our method is so far the first "meshless" attempt in computer graphics and modeling field for computing harmonic volumetric map, and it outperforms other existing techniques in smoothness, stability, and computational efficiency. The computational complexity of our algorithm is only solving a dense linear system. Furthermore, once the coefficient matrix is decomposed, it can be reused very efficiently whenever the boundary condition changes—only a matrix-vector multiplication is necessary for the new mapping computation. In our current work, the behavior of the volumetric mapping in the interior region is determined by the shell boundary correspondence, and we will pursue mapping with free boundary constraints in the future.**

***Index Terms*—Computational geometry and object modeling, computer graphics, computing methodologies, geometric algorithms.**

I. Introduction

RECENT advances in modern 3-D scanning and acquisition techniques have led to the rapid growth in terms of the number of digital models and their complexity. More and more volumetric objects are routinely obtained and stored in shape repositories. Shapes in databases usually vary significantly based on their topological types and geometric complexity; and they could be acquired from different viewing positions, in different resolutions; furthermore, shapes are oftentimes deformable with time-varying behaviors. All of these give rise to the difficulty in effectively analyzing, comparing, and searching shapes. One viable approach for the shape matching and analysis purpose is to establish the correspondence between objects of interest. Towards this goal, we need either a registration process between objects or a parameterization technique from objects onto certain canonical domains, both of which are very difficult due to the geometric and topological complexity of the underlying solid objects. In principle, building the correspondence between objects is equivalent to seeking a mapping from one domain to another, which remains to be one of the key issues in graphics and solid modeling fields in order to facilitate many applications. Two-dimensional surface mappings and 3-D volumetric mappings are most relevant and deserve extensive research investigation.

Computing correspondence between two surfaces has been widely studied in computer graphics, usually for surface deformation or morphing purpose. Its variation, parameterizing a surface onto planar domains, has been a central research topic in graphics and modeling areas in recent years. It arose from applications such as texture mapping; and it aids in many scientific computations such as providing domains for continuous

Manuscript received August 13, 2007; revised March 26, 2008. First published May 05, 2009; current version published July 01, 2009. This paper was recommended for publication by Associate Editor S. Hu and Editor M. Wang upon evaluation of the reviewers' comments. This work was supported in part by NSF funding CCF-0448399, DMS-0528363, DMS-0626223, IIS-0713145, IIS-0710819, and CCF-0727098, and in part by Louisiana Board of Regents PFund: NSF(2009)-PFUND-133, and Singapore National Research Foundation Interactive Digital Media R&D Program under Research Grant NRF2008IDM-IDM004-006.

X. Li is with the Department of Electrical and Computer Engineering, and the Center for Computation and Technology, Louisiana State University, Baton Rouge, LA 70803 USA (e-mail: xinli@lsu.edu).

H. Wang, X. Gu, and H. Qin are with the Computer Science Department, State University of New York (SUNY), Stony Brook, NY 11794 USA (e-mail: wanghy@cs.sunysb.edu; gu@cs.sunysb.edu; qin@cs.sunysb.edu).

X. Guo is with the Department of Computer Science, University of Texas at Dallas, Richardson, TX 75083 USA (e-mail: xguo@utdallas.edu).

Y. He is with the School of Computer Engineering, Nanyang Technology University, Singapore 639798 (e-mail: yhe@ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2009.2014735

surface spline construction and physically based simulation or deformation. In reality, despite the necessity of surface mapping techniques, interior volume data carries abundant information including material, density, texture, etc. (beyond pure geometric information). Therefore, not only the thin-shell of the object but the entire solid model should be taken into account in many cases of solid modeling, shape analysis, and physically based computation. For example, most of the physically based deformation techniques are volume-driven. Volumetric mapping between objects instead of surface mapping, serves as an enabling and more accurate tool for this task. In spite of this strong need, due to its technical challenges and computational complexity, much less work has been actually carried out in volumetric mapping compared with the surface case. In this paper, we aim to pursue a robust, efficient, and accurate algorithm to compute the harmonic volumetric mapping between two solid objects. We make use of the boundary method, in which the behavior of the interior region of the volume data is determined only by its surface boundary, thus naturally reduces this volume problem to that of its boundary surface scale. The harmonicity of the mapping is guaranteed by the method of fundamental solution.

Harmonicity of a mapping characterizes smoothness, which is a natural phenomenon that depicts the minimized physical energy configuration that arises from the difference between two shapes. In the surface case, harmonic mapping tries to achieve this by vanishing on the source surface the Laplace-Beltrami operator. Intuitively speaking, finding a harmonic mapping between two surfaces with fixed boundary correspondence is like computing the final deformation of a rubber membrane. The membrane has the source surface as its relaxed shape configuration, and is wrapped onto the target shape with certain fixed boundary constraints. The final mapping that leads to the physically natural deformation should minimize the harmonic energy and is what the algorithm aims to achieve. Similarly, for harmonic volumetric mapping, we fix the boundary mapping, which is now a surface mapping between shells of the two given solid objects. Then we seek a smooth interior region mapping by enforcing 3-D Laplacian everywhere to be zero. This is equivalent to arriving at the final stable configuration of a solid rubber subject to its boundary shape constraint.

Unlike the surface case, the variational procedure that minimizes a predefined energy needs to adjust a much larger number of points, which usually results in an intolerable computation complexity. According to the maximum principle of harmonic functions, interior value of a smooth field is determined by its boundary setting. Therefore, we can use a boundary method called *method of fundamental solution* (MFS) to solve this problem, hence reducing the volumetric solid problem to just the boundary surface scale. To our best knowledge, this is the first work in the computer graphics area that MFS is employed to solve the volumetric mapping problem.

Several applications are employed to demonstrate the efficacy of our mapping technique. These applications also show the importance of the volumetric harmonic mapping. The first and natural application is to use this correspondence to establish a registration between two solid models. Information on one model can be transferred to another; thus material, texture, and disparate functions defined on a volume domain can be transplanted and reused easily. With the registration being established, we can also measure the distance between shapes naturally by the energy required to deform one solid object to the other. We are able to visualize the deformation energy distribution, which aids in shape and deformation sequence analysis. Second, a solid parameterization can be computed once we have the mapping between a solid object and a canonical 3-D domain. We transplant the tetrahedralization of standard regular shapes such as polycubes onto other objects. Such a remeshing mechanism provides a highly regular tetrahedron structure for complex solid objects, which makes the geometry operations and computations more efficient, and suitable for graphics hardware acceleration. Third, our method can be used for a solid texture synthesis, which generates solid texture from the object's boundary surface texture mapping.

Our specific contributions in this paper are three-fold.

1) We develop a simple and efficient algorithm that can robustly and automatically compute the harmonic volumetric mapping from one volumetric object to another.
2) To the best of our knowledge, this is the first attempt to bring the method of fundamental solution into the graphics and solid modeling community. The technique is an efficient meshless boundary method with great potential. Earlier work provides some theoretical analysis from the point of view of mathematics and mechanical engineering, but it lacks experimental validation. We conduct experiments on the problem of computing harmonic volumetric mapping using this method; and we provide some valuable suggestions for using this method in the modeling field.
3) We demonstrate our harmonic volumetric mapping with several applications, which not only illustrate our mapping results, but also show the strong potential of volumetric mapping as an enable tool for future graphics and modeling research.

We will briefly review the related literature in Section II. Then we introduce the theory and algorithm of our method in Section III, followed by Section IV with some implementation details and property discussions. Finally, we demonstrate our experimental results with some applications in Section V and conclude our work in Section VI.

II. Related Work

Harmonic Maps: Having been extensively studied in the literature of surface parameterization, harmonic maps are usually addressed from the point of view of minimizing Dirichlet Energy. Its discrete version was first proposed by Pinkall and Polthier [1] and later introduced to computer graphics field in work of Eck *et al.* [2]. By discretizing the energy defined in [1], Desbrun *et al.* [3] constructed free-boundary harmonic maps. More harmonic and conformal maps are studied and surveyed by Floater and Hormann [4]. The harmonic maps between surfaces arose from shape blending [5] and are widely used in shape morphing applications later [6]–[10].

Harmonicity in the volumetric sense is similarly defined via the vanishing Laplacian, representing the smoothness of the mapping function. Wang *et al.* [11] studied the formula of harmonic energy defined on tetrahedral mesh and computed the discrete volumetric harmonic maps by a variational procedure. Ju *et al.* [12] generalized the mean value coordinates [13] from surfaces to volumes and built a smooth volumetric interpolation. Later, Joshi *et al.* [14] presented Harmonic Coordinates for volumetric interpolation and deformation purpose, their method guaranteed to provide nonnegative weights and therefore in concave regions, it led to a more pleasing interpolation result, compared to [12]. However, the technique introduced in [14] lacked the closed form expression for a given interior point, which increases computational burden and reduces numerical accuracy. In contrast to [11] and [12], our method is meshless and has the closed form representation. Therefore, our method has much better flexibility and efficiency when boundary condition change (for example, when mapping a sequence of different data) or mapping resolution change (for example, when adaptive or hierarchical structure is used) is necessary.

Boundary Method and MFS: We construct the mapping through a meshless procedure by using a boundary method called *method of fundamental solution* (MFS). Notable work among boundary methods for solving elliptic partial differential equations (PDEs) includes the classical boundary integral equation and boundary element method (BIE/BEM), which has been widely used in many engineering applications [15], and was introduced into computer graphics for the simulation of deformable objects in [16]. One of the major advantages of the BIE/BEM over the traditional finite element method (FEM) and finite difference method (FDM) is that only boundary discretization is usually required rather than the entire domain discretization needed for solving the PDEs numerically. Compared with the BIE/BEM approach, the MFS uses only the fundamental solution in the construction of the solution of a problem, without using any integrals over boundary elements. Furthermore, the MFS is a true meshless method, since only boundary nodes are necessary for all the computation. "Meshless" has the advantage of simplicity that neither domain nor mesh connectivity is required in storage and computation; so it becomes very attractive in scientific computing and modeling [17], [?]. A comprehensive review of the MFS and kernel functions for solving many elliptic PDE problems was documented in [19].

III. Theory and Algorithm

We pursue a volumetric map $\vec{f}$ from a given solid object M_1 to another object M_2, this is equivalent to building up a smooth one-to-one correspondence between M_1 and M_2. The boundary constraint is a surface mapping $\vec{f'}$ from the boundary surface of M_1, denoted as ∂M_1, to the boundary surface of M_2, ∂M_2.

We focus on objects embedded in $\mathbb{R}^3$. So the mapping $\vec{f}(\mathbf{p}) = \mathbf{q}\,(\mathbf{p} \in M_1$, and $\mathbf{q} \in M_2)$ can be decomposed into three separate components for three axes directions, i.e., $\vec{f} = (f^0, f^1, f^2)$. In each direction, f^i maps the point $\mathbf{p}$ to $\mathbf{q}$'s corresponding component q_i. This problem is then reduced to the computation of three separate $f^i (i = 0, 1, 2)$, with the given boundary mapping constraints $\vec{f'} = (f'^0, f'^1, f'^2)$.

A. An Intuitive Explanation of Our Idea

We first introduce our idea in an intuitive way through the electrostatics point of view. In each direction, our target is the harmonic function f^i. The harmonicity, or smoothness, can be simulated using an electric field. Suppose we have lots of electronic particles, if we can place them in $\mathbb{R}^3$ as we like, and set arbitrary charge amount on each of them, then we can flexibly control the electric field we get. An important fact is that the potential of electric fields is guaranteed to be harmonic. Therefore, once the electric field generated by this particle system simulates the boundary condition, in other words, the potential on each boundary point $\mathbf{p} \in \partial M_1$ satisfies the boundary constraint $f'^i(\mathbf{p})$, then we can use the potential of this particle system to simulate the mapping component f^i in the interior region.

Electric fields provide a correct simulation for harmonicity because its potential satisfies the vanishing Laplacian operator everywhere, except for the positions of those particles, where the potential is infinite. Therefore, we call these particles *singularity points* or *source points*; and in order to make the potential valid everywhere in the interior region of M_1, we should place them outside of M_1.

Since we know the harmonicity is guaranteed, we are only left to enforce the boundary conditions f'^i. We fix positions of all particles, and the charge amount carried on each particle provides the freedom we have for enforcing the boundary constraints. This fitting process, as we will show in the coming Section III-B, leads to a linear system. We place many estimation points on the boundary ∂M_1, and compute the charge amount distribution which can result in the desired potential given by f'^i. By solving a linear system, we obtain the best fit charge amount distribution, and get the simulation of f^i.

Intuitively, if we have dense enough particles placed outside of M_1, the smooth boundary condition can always be well approximated, only with exceptions in some extremely discontinuous boundary regions. For more details, we refer readers to [20].

This section gives an intuitive explanation and pipeline of our idea. In the coming section, we will rigorously formulate our algorithm, and address its theoretical foundation.

B. Formulations

We rigorously formulate our volumetric mapping problem as follows:

Given a one-to-one mapping $\vec{f'}$ between boundary surfaces ∂M_1 and ∂M_2: $\vec{f'}(\mathbf{p}) = \mathbf{q}, \mathbf{p} \in \partial M_1, \mathbf{q} \in \partial M_2$, our goal is to compute a mapping $\vec{f} : M_1 \to M_2$ such that

$$\begin{cases} \Delta \vec{f}(\mathbf{p}) = 0 & \mathbf{p} \in M_1, \\ \vec{f}(\mathbf{p}) = \vec{f'}(\mathbf{p}) & \mathbf{p} \in \partial M_1. \end{cases}$$

where the operator Δ is defined continuously in 3-D as

$$\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2},$$

and $\Delta \vec{f} = 0$ for $\vec{f} = (f^0, f^1, f^2)$ is equivalent to $\Delta f^i = 0$ for all i = 0,1,2.

Harmonicity and Kernel Function: Since Δ is a linear self-adjoint differential operator, we can compute its Green Function. We denote $g^i(\mathbf{x}) = \Delta f^i(\mathbf{x})$, and denote Δ^{-1} as the inverse of the operator Δ, so that $\Delta\Delta^{-1} = I$ where I is the identity operator; then we can write the solution as $f^i(\mathbf{x}) = \Delta^{-1} g^i(\mathbf{x})$.

Note that $f^i(\mathbf{x}) = \Delta^{-1} g^i(\mathbf{x}) = \Delta^{-1} \int \delta(\mathbf{x} - \mathbf{x}') g^i(\mathbf{x}') d\mathbf{x}'$, where δ is the Dirac function. If we make a *kernel function* $K(\mathbf{x}, \mathbf{x}')$ that satisfies $\Delta K(\mathbf{x}, \mathbf{x}') = \delta(\mathbf{x} - \mathbf{x}')$. We can rewrite the solution to f^i following the above equation in terms of the Kernel function as

$$f^i(\mathbf{x}) = \int K(\mathbf{x}, \mathbf{x}')g^i(\mathbf{x}')d\mathbf{x}'.$$

Such a kernel function K is known as the Green's function associated with the 3-D Laplacian operator Δ, and has the formula: $K(\mathbf{x}, \mathbf{x}') = (1/4\pi)(1/|\mathbf{x} - \mathbf{x}'|)$, where $|\mathbf{x} - \mathbf{x}'|$ denotes the distance between the points $\mathbf{x}$ and $\mathbf{x}'$.

Since f^i in the interior region is fully determined by the boundary values, we discretely solve it using *Method of Fundamental Solution* (MFS) [19] with the above kernel. The linear nature of Laplacian operator indicates that the boundary-based method such as MFS is most suitable since the interior harmonicity is now represented in an exact manner; we only need to enforce the given boundary condition function $\vec{f'}$, i.e., f_i' for each f^i. The MFS approximation equation to evaluate f^i on an interior or boundary point $\mathbf{p}$ is

$$f^i(\vec{w^i}, \vec{\mathbf{Q}}; \mathbf{p}) = \sum_{n=1}^{N_s} w_n^i \cdot K(\mathbf{p}, \mathbf{Q_n}), \mathbf{p} \in M_1. \qquad (1)$$

In this above equation, $\vec{\mathbf{Q}}$ is a $3N_s$-dimensional vector concatenating positions of all N_s 3-D source points. $\vec{w^i} = (w_1^i, w_2^i, \cdots, w_{N_s}^i)^T$ is the N_s-dimensional vector representing the charge amount distribution on these source points. It is firstly unknown and is what we want to solve.

Note that source points $\mathbf{Q_n} \in \mathbb{R}^3$, $n = 1, \cdots, N_s$ should lie outside of M_1, in other words, they are located on the boundary $\partial\widetilde{M_1}$ of a region $\widetilde{M_1}$ containing M_1 (i.e., $M_1 \subset \widetilde{M_1}$). Once no source points are inside M_1, an arbitrary charge distribution $\vec{w}$ can guarantee in an exact manner the vanishing Laplacian operator on f^i in the interior region of M_1, only violating the boundary conditions.

Boundary Fitting: To enforce the boundary conditions, we pursue a special set of charge amounts $\vec{w}$, such that the images of points on ∂M_1 satisfy the boundary map $\vec{f'}$. This boundary fitting process is performed as follows, we sample a set of estimation points on the M_1's boundary surface ∂M_1. These points are called *constraint points* or *collocation points*. Unlike the aforementioned source points which are electronic particles and will be used for estimating electric potentials in interior region of the solid object, collocation points are different types of points; the configuration of source points does not affect the configuration of collocation points and vice versa.

Boundary conditions are represented by enforcing collocation points to be mapped to the boundary of the target model ∂M_2. Their images are given with $\vec{f'}$. For example, for N_c number of collocation points $\mathbf{P_1}, \mathbf{P_2}, \cdots, \mathbf{P_{N_c}}$: in each axis direction, we denote their image vector as $\vec{b^i} = \{b_1^i, b_2^i, \cdots, b_{N_c}^i\}^T = \{f'^i(\mathbf{P_1}), f'^i(\mathbf{P_2}), \cdots, f'^i(\mathbf{P_{N_c}})\}^T$. According to (1), this vector $\vec{b^i}$ can be represented by $\mathbf{A}\vec{w^i}$, where $\mathbf{A}$ is called the *coefficient matrix*, whose element $A_{uv} = K(\mathbf{P_u}, \mathbf{Q_v})$ (P_u is a collocation point while Q_v is a source point). These images should satisfy the boundary condition, i.e., equal to $\vec{b^i}$. Therefore, the above fitting process reduces to a linear system on each axis direction

$$\mathbf{A}\vec{w^i} = \vec{b^i}.$$

We solve three linear systems in three different axis directions separately. The resultant vectors (w^0, w^1, w^2) corresponding to three different charge distributions, are used to compose our final harmonic volumetric mapping.

C. Algorithm

With the discussion above, we can formulate our algorithm. The input are two given solid objects M_1, M_2 and their boundary surface mapping $\vec{f'} : \partial M_1 \rightarrow \partial M_2$. The output is a harmonic volumetric mapping $\vec{f} : M_1 \rightarrow M_2$ s.t. on boundary $\vec{f}(\mathbf{p}) = \vec{f'}(\mathbf{p}), \mathbf{p} \in \partial M_1$ and in the interior region: $(\partial^2\vec{f}/\partial x^2) + (\partial^2\vec{f}/\partial y^2) + (\partial^2\vec{f}/\partial z^2) = 0$. $\vec{f}$ is decomposed to (f^0, f^1, f^2), we solve each f^i separately as follows.

In: M_1, M_2, $\vec{f'}$.

Out: f^i for i = 0,1,2.

1) Place the source points and the collocation points. (Section IV-A and Section IV-G).
2) Compute the coefficient matrix $\mathbf{A}$. Its element $\mathbf{A}_{uv}$ takes the value of the kernel function K on the collocation point $\mathbf{P_u}$ and the source point $\mathbf{Q_v}$.
3) Decompose the coefficient matrix using singular value decomposition (SVD). (Section IV-B and Section IV-F).
4) Solve this linear system under the given boundary mapping constraints f'^i using the decomposition result from Step 3), and get f^i represented by $\vec{w^i}$ in the form of (1).

The resultant volumetric mapping is harmonic, guaranteed by the kernel function. It minimizes the harmonic energy, which will be discussed in Section IV-C. We assume the boundary surface mapping $\vec{f'}$ is given as an input, and in Section IV-D, we briefly discuss how to obtain this surface mapping with existing techniques.

IV. Implementation and Discussion

A. Source Points and Collocation Points Placement

In the first step of our algorithm, we place source points uniformly on an offset surface $\partial\widetilde{M_1}$ outside the boundary surface ∂M_1, as shown in Fig. 1(b). The following procedure is a robust way to create such a sampling.

In: M_1

Out: A uniform sampling on the offset surface $\partial\widetilde{M_1}$.

1) Compute the implicit distance field in $\mathbb{R}^3$ with respect to the given object boundary surface ∂M_1 using technique introduced in [21]. We get a distance evaluation function $dist(\mathbf{p})$ that the signed distance from any point $\mathbf{p} \in \mathbb{R}^3$ to ∂M_1 can be computed efficiently.
2) Build an offset surface $\partial\widetilde{M_1}$ using Bloomenthal's polygonization method [22]. The Bloomenthal's polygonization method takes an implicit distance evaluation function $dist2(\mathbf{p})$ defined in $\mathbb{R}^3$ as the input. Therefore, to build the offset surface $\partial\widetilde{M_1}$ with distance d to ∂M_1, we set such input function $dist2(\mathbf{p}) = dist(\mathbf{p}) + d$.

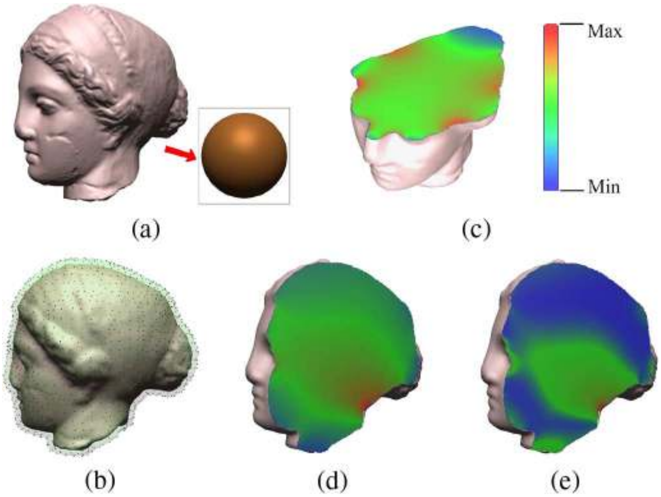


Fig. 1. Volumetric harmonic mapping from the solid Igea model to the solid sphere model. (a) shows the source and target objects. As shown in (b), source points are placed on an offset surface. The harmonic energy distribution of the mapping is color-coded and illustrated on two different cross sections [(c) and (d)], and the deformation energy distribution is illustrated over a cross section [(e)]. (For the references to colors in our figures, we refer the reader to the web version of this paper.)

3) Uniformly sample n points on mesh $\partial\widetilde{M_1}$. We use the uniform sampling technique introduced in [23].

We place collocation points by sampling boundary points $\mathbf{p} \in \partial M_1$ uniformly. The reason that we conduct the source and collocation points placement in this way will be discussed in Section IV-G.

B. Solving the Linear System

As discussed in Section III-B, we want to solve the linear system $\mathbf{A}\vec{w^i} = \vec{b^i}$. Element in the coefficient matrix $\mathbf{A}$ is the value of the kernel function on each collocation point, which is almost never zero, making the matrix quite dense. The matrix may be ill-conditioned [24], in which case, regular linear system solvers such as Gaussian elimination or LU decompositions usually fail to produce a stable solution. We use SVD because it approaches accurate and stable results even when the coefficient matrix is highly ill-conditioned. Another advantage of using SVD is that once we have decomposed the matrix, we can reuse the result for the rapid re-computation of new mappings whenever boundary conditions change. This efficiency also arises from the boundary method, and detailed discussion about this aspect will be given in Section IV-F. More advantages of using SVD in MFS are discussed in [24].

C. Energy of Volumetric Mapping

Harmonic Energy: Harmonic Energy measures the smoothness of the mapping. It is measured by the integration of the square of the gradient over the entire interior region domain M_1. Both the source and the target models should be normalized to unit size in volume before computing the harmonic energy. The total harmonic energy of this mapping is

$$\int_{M_1} \langle \nabla\vec{f}, \nabla\vec{f} \rangle d\mathbf{x}. \tag{2}$$

We built a volumetric grid structure and compute the gradient of the mapping on each grid point, then use the following formula to approximate the energy:

$$\sum_{\mathbf{p}_{\mathbf{u},\mathbf{v},\mathbf{w}}\in M_1} \left\|\nabla\vec{f}(\mathbf{p}_{\mathbf{u},\mathbf{v},\mathbf{w}})\right\|^2 \cdot vol(\mathbf{p}_{\mathbf{u},\mathbf{v},\mathbf{w}}), \tag{3}$$

where $vol(\mathbf{p}_{\mathbf{u},\mathbf{v},\mathbf{w}})$ is the interior volume on grid point $\mathbf{p}_{\mathbf{u},\mathbf{v},\mathbf{w}}$, and $vol(\mathbf{p}_{\mathbf{u},\mathbf{v},\mathbf{w}})$ equals to the volume of the intersection of M_1 and the small grid cube $Cube_{\mathbf{p}_{\mathbf{u},\mathbf{v},\mathbf{w}}}$ centered at $\mathbf{p}_{\mathbf{u},\mathbf{v},\mathbf{w}}$. Here, the edge length of $Cube_{\mathbf{p}_{\mathbf{u},\mathbf{v},\mathbf{w}}}$ is the distance between two adjacent grids. We can use the volume of $Cube_{\mathbf{p}_{\mathbf{u},\mathbf{v},\mathbf{w}}}$ to approximate $vol(\mathbf{p}_{\mathbf{u},\mathbf{v},\mathbf{w}})$. With the increase of the grid sampling density, the value of (3) is asymptotic to (2). We use the simple volume grid data structure because it is easy to implement and efficient in tracing function values on neighboring grid points: the gradient of the harmonic mapping on each grid point can be represented by three vectors: $\nabla\vec{f} = (\nabla f^0, \nabla f^1, \nabla f^2)$, so $\|\nabla\vec{f}(\mathbf{p}_{\mathbf{u},\mathbf{v},\mathbf{w}})\|^2 = \|\nabla f^0\|^2 + \|\nabla f^1\|^2 + \|\nabla f^2\|^2$. In the example of Fig. 1, the harmonic energy distribution of the volumetric mapping is colorized in (c) and (d) over two different cross sections. The color-coding scheme in our paper is shown in the bar in (c): red represents the maximum while blue represents the minimum.

Deformation Energy: Once a correspondence between two solid objects is created, deformation around each interior voxel point can be estimated easily. This provides us a formal mechanism to compute the energy required to deform one object to another. In Section V we will use this energy to measure the difference between two shapes.

To compute the deformation energy, we start from the classical strain and stress tensor analysis. Green's strain tensor ϵ is used to quantify the local strain undergoing a 3-dimensional deformation. If an interior point $\mathbf{p}$ is mapped to $\mathbf{q}$, then the 3 × 3 tensor ϵ has its elements ϵ_{ij} represented by

$$\epsilon_{ij} = \frac{\partial\mathbf{q}}{\partial p_i} \cdot \frac{\partial\mathbf{q}}{\partial p_j} - \delta_{ij},$$

where $0 \le i,j \le 2$ are indexes in axis directions, δ_{ij} is the Kronecker delta

$$\delta_{ij} = \begin{cases} 1: & i = j \\ 0: & i \ne j. \end{cases}$$

According to differential geometry, this strain tensor is invariant under rigid transformation and vanishes under identity mapping. The stress tensor represents the information of the internal forces under the deformation. A simplified linear form of elastic stress with the assumption of isotropy is defined as

$$\sigma_{ij} = \sum_{k=1}^{3} \lambda\epsilon_{kk}\delta_{ij} + 2\mu\epsilon_{ij},$$

where λ and μ are two Lamé constants of material, respectively, representing rigidity and resistance to volume dilation change. For example, in most of our experiments, we make use of the parameter of rubber, i.e., $\lambda = 0.0335$, $\mu = 0.0224$. Finally, the elastic potential density η on this point $\mathbf{p}$ is measured by

$$\eta(\mathbf{p}) = \frac{1}{2}\sum_{i=1}^{3}\sum_{i=1}^{3} \sigma_{ij}\epsilon_{ij},$$

representing the internal elastic energy under the shape change. Similar to (3), the total deformation energy of this volumetric mapping is computed by

$$\sum_{\mathbf{p}_{\mathbf{u},\mathbf{v},\mathbf{w}} \in M_1} \eta(\mathbf{p}_{\mathbf{u},\mathbf{v},\mathbf{w}}) \cdot vol(\mathbf{p}_{\mathbf{u},\mathbf{v},\mathbf{w}}). \qquad (4)$$

Fig. 1(e) color-codes the deformation energy distribution of the volumetric mapping from the solid Igea model to the solid sphere.

D. Boundary Conditions

The boundary condition of our harmonic volumetric mapping is a surface mapping between ∂M_1 and ∂M_2. We assume that it is provided as an input. Existing surface mapping techniques [5]–[10], [25] can be used for creating the boundary surface mapping. On one hand, the efficiency of the surface mapping is important. Given the boundary map, our following algorithm is fast and fully automatic; therefore, to make the entire volumetric mapping pipeline efficient and automatic, we hope this surface mapping creation is simple and automatic as well. On the other hand, surface mapping with relatively low distortion is preferred. As we know, quality of a harmonic mapping depends on the boundary condition. How to generate a suitable boundary surface mapping and how the quality of this mapping affects the volume mapping are worthy of more comprehensive research studies. Since the detailed discussion goes beyond the scope of this paper, we will only briefly explain how we get a relatively good initial surface mapping. We consider mapping between solid objects with the same topology, that is, the objects have pairs of corresponding boundary surfaces. Consider the mapping between each surface pair: 1) If the boundary surfaces are closed genus-0 surfaces, the conformal surface mapping suffices. This mapping can be computed/combined through a common sphere domain, similar to techniques in [5]. 2) If the surfaces are of higher genus, we prefer a globally smooth mapping. In this work, we compute their quasi-conformal mapping as the initial boundary condition using techniques introduced in [26].

To demonstrate the different volumetric mappings results from different boundary surface mappings, we perform experiments and show results in Fig. 2. The harmonic volumetric mappings from the solid Teapot model to the solid Cup model have different harmonicity under two different boundary surface mappings [as shown in (a)]. We render the mesh connectivity for the points on the target boundary surfaces to better visualize the differences [(c) and (e)]. The second boundary mapping (e) is smoother than the first one (c); it leads to a volumetric mapping (f) with smaller harmonic energy. The harmonic energy distributions for two volumetric mappings are visualized on the Teapot model from the same cross section. The color-coding scheme for (d) and (f) is depicted in (b).

In the future work, we plan to examine the technical issue of how the surface mapping and the volumetric mapping are related in a quantitative way, and how one mapping guides the computation of the other. By adjusting the surface boundary mapping condition accordingly, we will pursue the free boundary volumetric mapping with minimum harmonic energy.

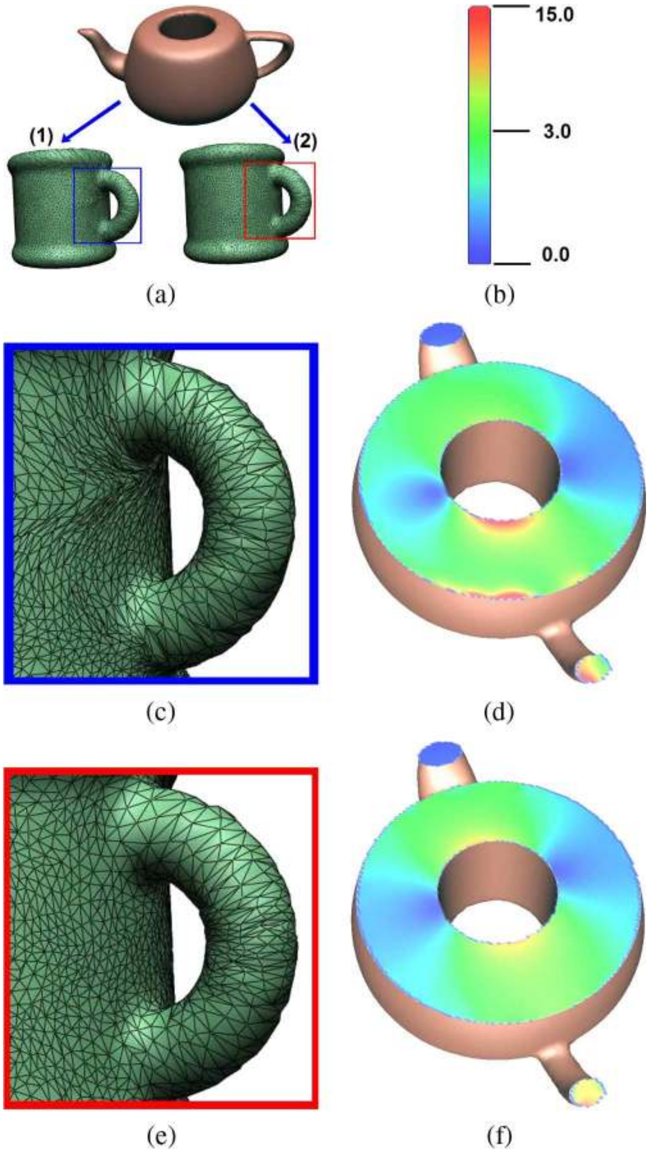


Fig. 2. Different boundary conditions lead to different volumetric mapping results even for the same target object. Volumetric mappings from the solid Teapot model to the solid Cup model (a) under two different boundary conditions [(c) and (e)] have different harmonicity as shown in (d) and (f) (energy distributions depicted on the Teapot model, respectively). (c) and (e) highlight different surface mappings with magnified views.

E. Comparison With Previous Work

We compare our mapping results with the method introduced in [11]. In their work, the discretized harmonic energy is defined on the tetrahedral mesh to guide their variational procedure. Once we compute our mapping, we can evaluate the mapping on any interior point using (1). We tetrahedralize our volume data (in our work we produce the tetrahedralization using [27]), then compare our results with Wang *et al.*'s work in [11]. As shown in Fig. 3, the volumetric mapping from the solid Igea model (a) to a solid sphere can be visualized by transferring the tetrahedral mesh of the Igea to the solid sphere. Our resultant tetrahedral mesh on sphere (b) appears to be smoother than the mesh produced in [11] (c). This smoothness is visualized from the distributions of the discretized harmonic energy [11] of volumetric mappings, which are color-coded in (b) and (c) using a uniform scheme (d).

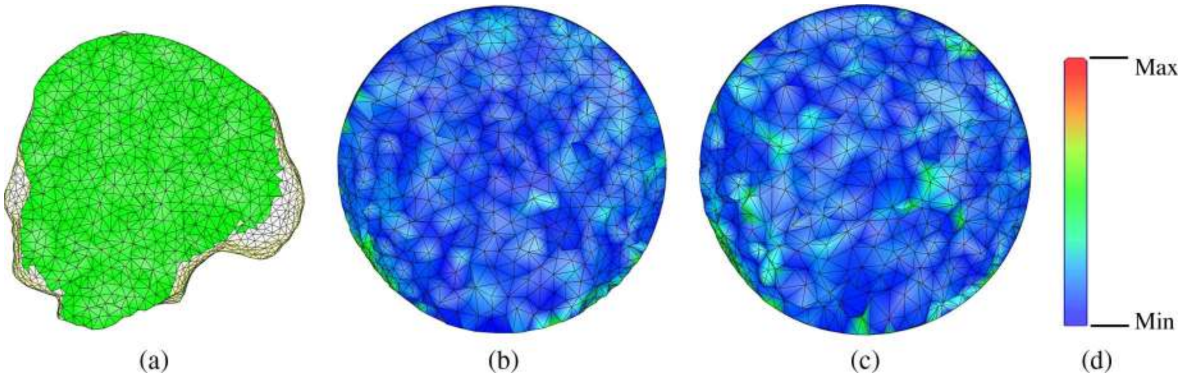


Fig. 3. Comparison with previous work. The initial tetrahedralization of the solid Igea model is shown in (a). In (b), the tetrahedral mesh on the solid sphere is transferred from the Igea model using our volumetric mapping algorithm. The result computed using Wang et al.'s variational technique is shown in (c). The harmonic energy distributions of two volumetric mappings are color-coded on resultant tetrahedral mesh using a uniform color-coding scheme as shown in (d).

Another important advantage of our algorithm is the meshless property. The discretization accuracy and the computational cost of [11] depend heavily on the tetrahedralization quality of the source object: dense tetrahedralization necessarily results in high computational complexity; and irregular tetrahedralization leads to large numerical error in approximating discretized harmonic energy. In contrast, our algorithm is independent of the connectivity, and thus is more flexible and can be adaptive to any volumetric data sets with spatial-varying resolution.

F. Computational Efficiency

The computation cost of our algorithm is equivalent to solving three linear systems. More importantly, since we decompose the coefficient matrix A from the MFS using SVD (Section IV-B), only one decomposition is necessary. Furthermore, it is very efficient to recompute the volumetric mapping under different boundary conditions.

Given a new boundary configuration $\vec{b'}$, the corresponding $\vec{w'}$ for the new volumetric mapping can be computed directly from $A^{-1}\vec{b'}$. With the decomposition results, $A^{-1} = VW^{-1}U^T$, where the matrix W^{-1} is a diagonal matrix that can be computed directly from W. Therefore, under a new boundary condition, the decomposition matrix results can be reused, only a multiplication operation between A^{-1} and $\vec{b'}$ is required.

This shows one more advantage of the boundary method over variational methods which apply iterations on the entire volume whenever the boundary condition is given. Under a new boundary condition, variational methods can not avoid a time-consuming recomputation. In our applications shown later in Section V, we take full advantage of our computational efficiency to compute a large number of sequential volumetric mappings in a temporal deformation sequence, by decomposing A only once.

G. Source, Collocation Points, and Mapping Efficiency

One important issue that we have to address is how many source points we need to use and where to place them. Using our electric field model, imagine that we want to refine our control of the electric field behavior, the more particles (i.e., source points) we have, naturally the more refined result we should be able to get with increasing computational complexity. On the other hand, numerically, their positions also matter. If all source points are placed in one position, there would be no way that we can achieve more satisfactory results with more source points.

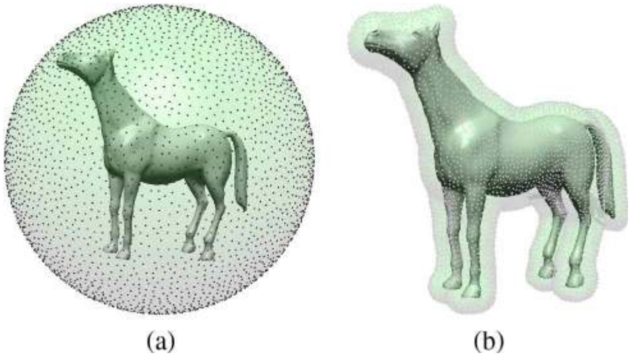


Fig. 4. Placement of Source Points. Source points are sampled either on a bounding sphere (a) or on an offset surface of the given model (b).

The positions of these source points actually determine the behavior of the coefficient matrix $\mathbf{A}$, which can be highly ill-conditioned [28]. The condition number of the matrix generally increases as the distance from $\widetilde{M_1}$ to M_1 increases, though the accuracy of the MFS approximation increases under this situation [29]. That is to say, distant source points give a smoother approximation, but unavoidably introduce larger numerical error. Theoretically optimal results of source positions are unknown at present; current literatures either suggest placing source points uniformly on a sphere within three times the diameter of M_1 [29], [30] or on an offset surface of M_1 [31]. The real-world computations in mechanical engineering field usually choose the source and collocation points in a trial-and-error manner or with the help of human experiences. Inspired by the above pioneering work, we use experimental results to find a suitable setting rule for our mapping problem, and guide the source and collocation points' placement in order to bridge the gap between theoretical results and practical common senses.

We conduct experiments in the following three aspects to find a suitable configuration for our volumetric mapping problem:

1) the shape of the surface $\widetilde{M_1}$ (source points are sampled on an offset surface or a sphere);
2) the distance from M_1 to $\widetilde{M_1}$;
3) the number of the source points and collocation points.

The experimental results are shown in charts plotted in Fig. 5. In this figure, Chart (a) plots the boundary constraint error when source points are placed on a sphere [see Fig. 4(a)], while Chart (b) shows the case when source points are on an offset surface

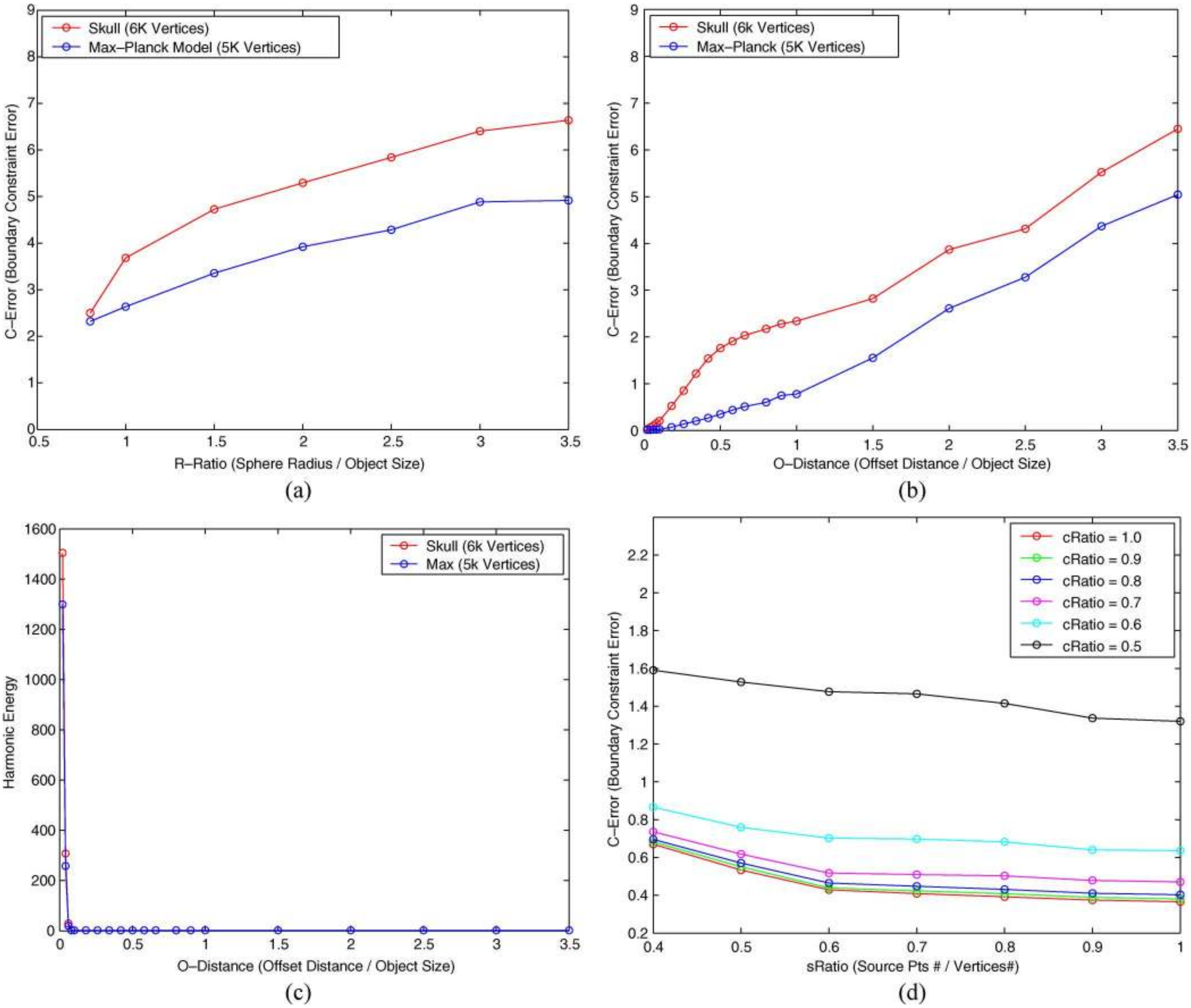


Fig. 5. Volumetric mappings under different source/collocation point configurations. (a) shows the boundary constraint error (*C-Error*) under different *R-Ratio* when source points are placed on spheres. (b) and (c) plot the constraint error (b) and harmonic energy (c) respectively under different *O-Distance* when source points are placed on offset surfaces. In (d), constraint error under different numbers of source points and collocation points are compared. The x axis is the *sRatio*. The y axis shows the constraint error. Different curves show the cases under different *cRatio*.

[see Fig. 4(b)]. In Chart (a), the x axis is the radius of the sphere, denoted as *R-Ratio*, represented by the ratio of the sphere radius over the object size, and y axis shows the boundary constraint error, denoted as *C-Error*. C-Error is computed using: $\sum_{\mathbf{p}} ||f'(\mathbf{p}) - f(\mathbf{p})||^2$ for all collocation points $\mathbf{p}$. C-Error measures the total fitting error of our volumetric mapping to the given boundary constraints. Therefore, we use its value to measure the quality of our mapping. Chart (b) shows the case that source points are placed on the offset surface; the x axis is the distance from $\partial\widetilde{M_1}$ to ∂M_1; its value, denoted as *O-Distance*, is the ratio of the distance over the source model size. Their corresponding C-Errors are plotted in y axis. Chart (c) shows the harmonic energy values (y axis) under the different offset surface settings (x axis).

Our statistical data demonstrates the following. 1) The closer to the model boundary source points are placed, the smaller the boundary constraint error can be achieved. 2) Placing source points on the sphere is not as good as on an offset surface. Because we require the object is totally inside the interior of the sphere, the radius of the sphere needs to be large enough and the average distance will be much larger compared with the offset surface placement. 3) If source points are placed on an offset surface that is too close to the model, the approximation for the fundamental solution is becoming unstable, which is shown from the values of their harmonic energies [Chart (c)]. Therefore, in our experiments and applications, we usually place source points on an offset surface with 0.1 O-Distance.

Chart (d) further shows how the numbers of source points and constraint points affect the boundary constraint errors. We define two ratios *cRatio* and *sRatio*, respectively. The cRatio is defined as the number of collocation points over the number of boundary points. The sRatio is defined as the number of source points over the number of boundary points. The x axis is the sRatio, and the y axis shows the boundary constraint error. Different curves show the cases using different cRatio. We can clearly see from this chart: the larger these ratios are, the smaller

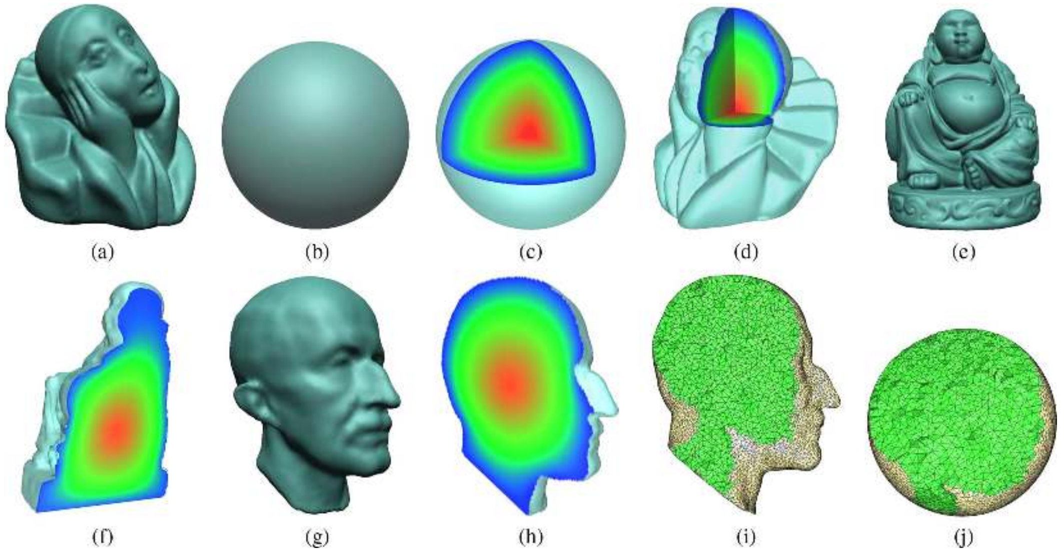


Fig. 6. Volumetric mappings to the canonical sphere. The Pierrot model (a) is mapped to the solid sphere (b); (c) shows the color-coded distance field in the sphere. (d) visualizes the volumetric mapping: each point p in the original model of (a) is mapped to a point q inside the solid sphere; the target position q's color (as shown in (c)) is transferred and depicted on the corresponding p position (as shown in (d)). Similar examples of mapping from genus-0 Buddha model (e) and the Max-Planck model (g) to the sphere (b) are visualized in (f) and (h) by this same *color-coded distance field transfer* method. The tetrahedral mesh on the Max-Planck model (i) is mapped onto the solid sphere (j). Their corresponding cross sections are visualized.

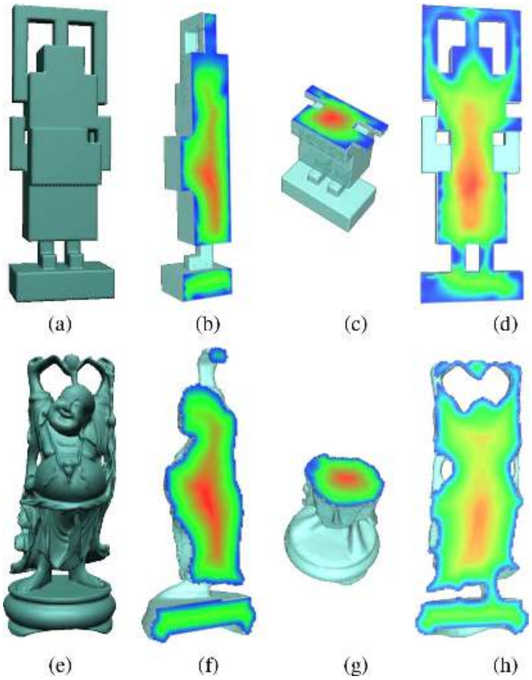


Fig. 7. Harmonic volumetric mapping from a solid polycube model (a) to the solid Buddha model (e). (f), (g) and (h) show the color-coded distance field of the Buddha, from three cross sections, respectively. This color-coded distance field is transferred from the Buddha to the PolyCube model as shown in (b), (c) and (d) correspondingly.

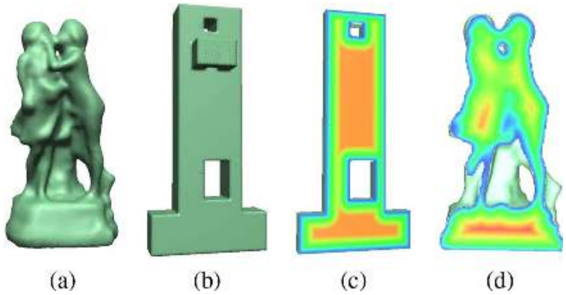


Fig. 8. Harmonic volumetric mapping from the solid Sculpture model (a) to a solid polycube model (b). (c) color-codes the distance field of the polycube. In (d), the transferred color-coded distance field is visualized on the Sculpture model.

boundary constraint error will be achieved. On the other hand, fewer source points create an over-constraint system which will be solved in a much shorter time. In our experiments, we usually set cRatio larger than 0.8 but sRatio around 0.6 for an efficient but well-fitted results for large models.

Unlike fixing the source/collocation points as discussed above, the positions of source points and collocation points can also be considered as unknowns in an optimization procedure, in which case they have to be computed along with the unknown weights during the optimization procedure. This necessarily complicates the entire solver and makes the computation procedure highly nonlinear.

Near a boundary region whose target shape is seriously wrinkled, the harmonic mapping may map interior points to the outside of the target object if the source/collocation points nearby are not dense enough. Such situation can be effectively remedied by increasing the density of source/collocation points around this region adaptively.

V. Experimental Results and Applications

We first show some experimental results of harmonic volumetric mappings in Figs. 6–8. In Fig. 6, a solid Pierrot model (a) is mapped to a solid sphere (b). The mapping result can be visualized using (c) and (d). In (c), the distance field of the interior region of the solid sphere is color-coded using the scheme shown in Fig. 1(c). Here in Fig. 6(d), each volume point $\mathbf{p}$ in solid Pierrot model (a) is mapped to an interior point $\mathbf{q}$ in

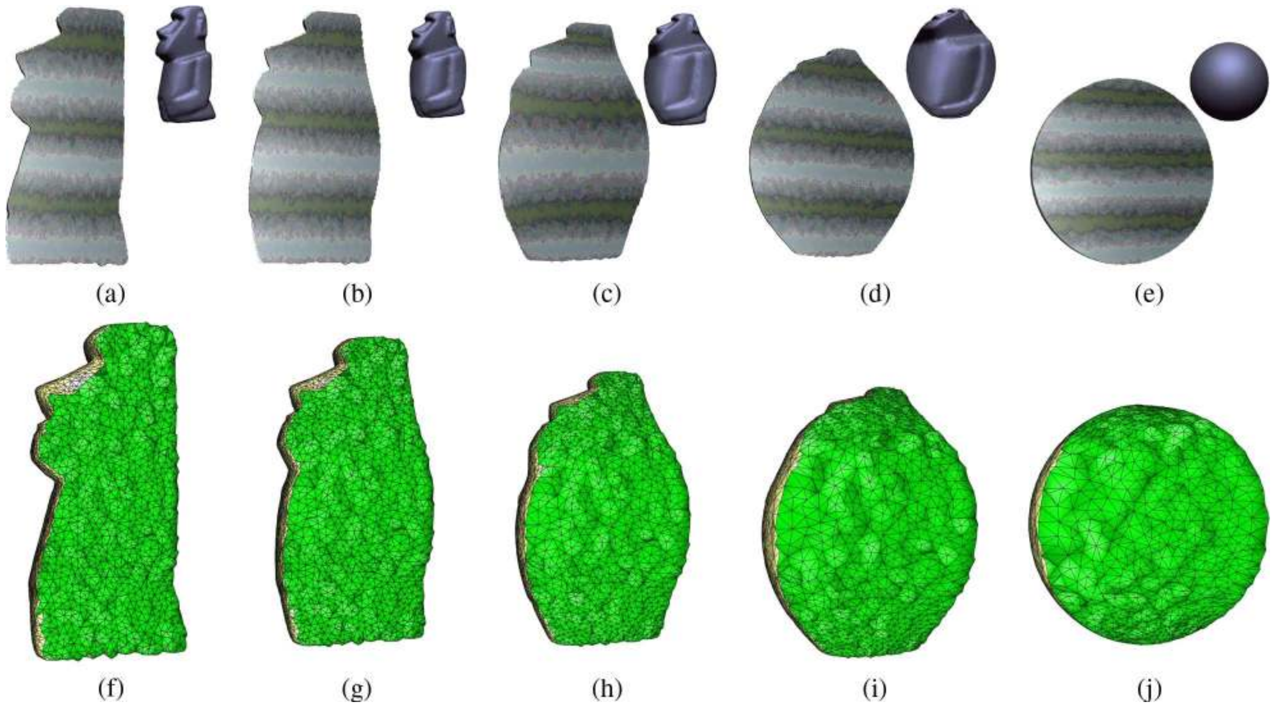


Fig. 9. Volumetric harmonic mapping for information reuse. The material on the solid Moai model is preserved when it deforms during the animation [(a)–(e)]. (f)–(j) show this consistency (via one corresponding cross section) on the original tetrahedral mesh and the mapped meshes during its morphing procedure.

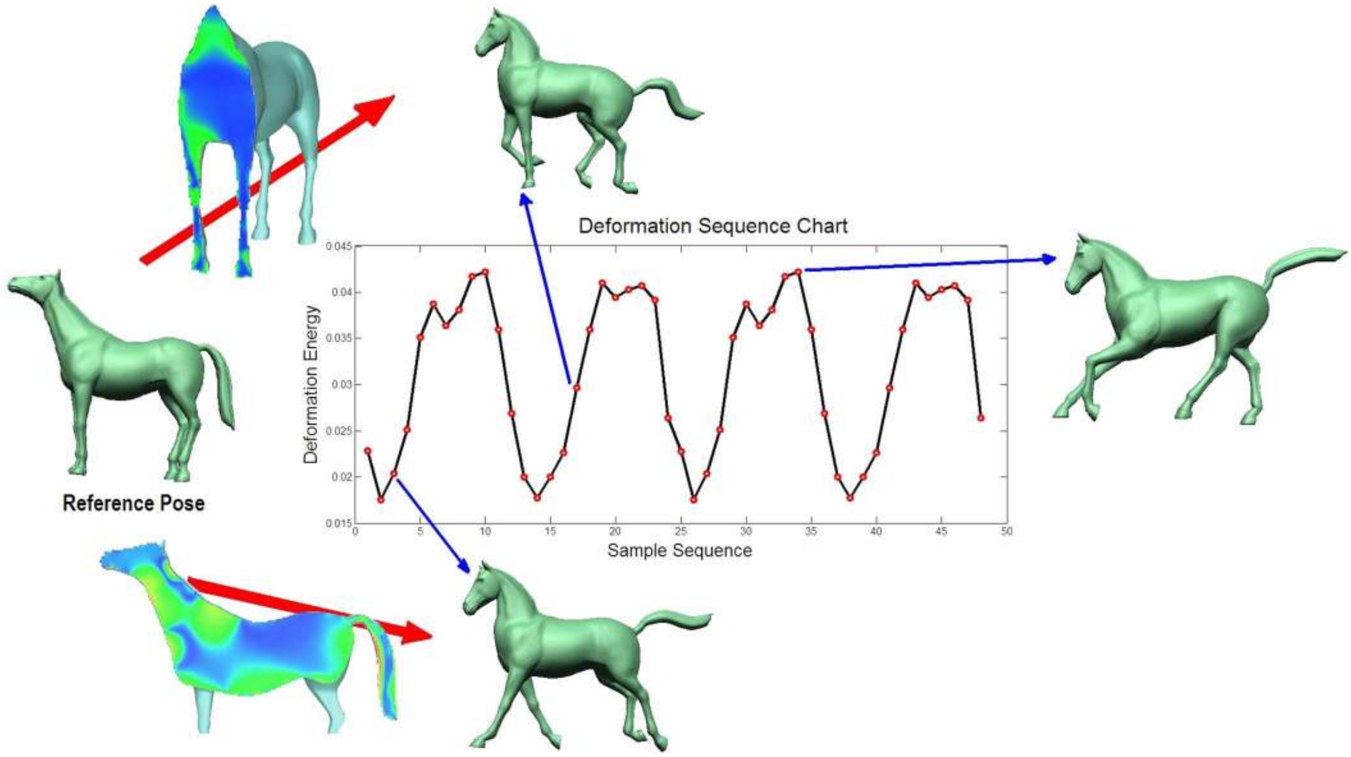


Fig. 10. Energy analysis of deformation sequences. The horse model is deformed in a sequence. The deformation energies are calculated (red circles). The distribution of the deformation energy required for each model in the sequence can be illustrated on the reference model.

solid sphere model (b). We transfer the color of $\mathbf{q}$ to the position of $\mathbf{p}$. This color-coded distance field on source model transferred from the target model provides an intuitive way to visualize the volumetric mapping result. We call this visualization method *Color-Coded Distance Field Transfer*. Another mapping example from the solid Buddha model (e) to the solid sphere (b) is computed and visualized in the same way as shown in (f). One more example, mapping from the solid Max-Planck model (g) to the solid sphere is visualized similarly in (h). We also use another method, tetrahedral mesh, to visualize volu-

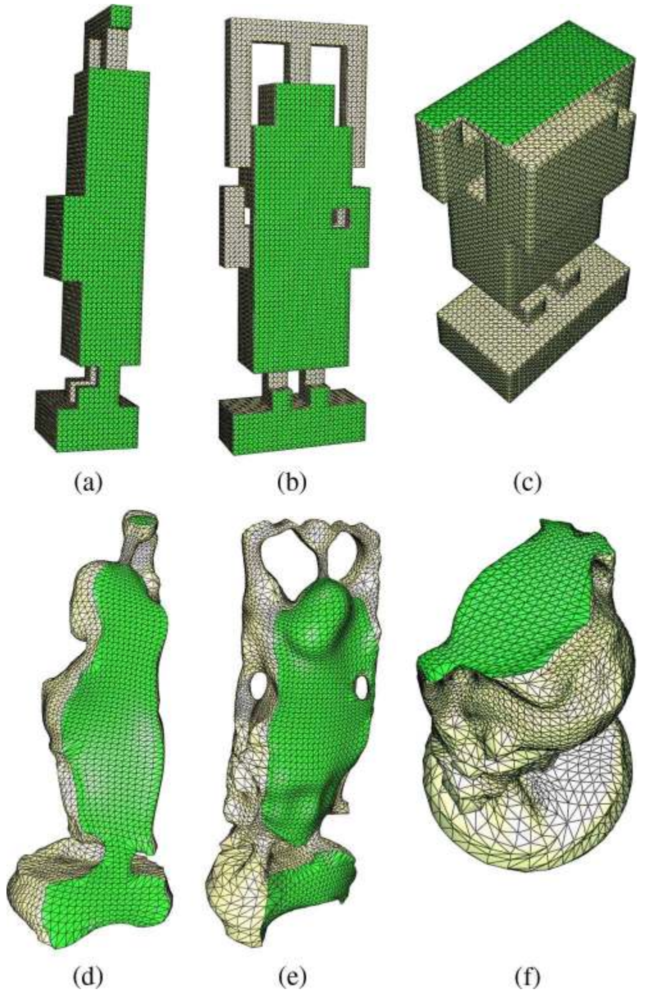


Fig. 11. Harmonic Volumetric Mapping for Tetrahedral Remeshing. (a), (b) and (c) show the tetrahedral mesh of the polycube model in Fig. 7(a) from three cross sections. It is utilized to *remesh* the solid Buddha model [Fig. 7(e)]; and the results are visualized with corresponding cross sections in (d), (e), and (f).

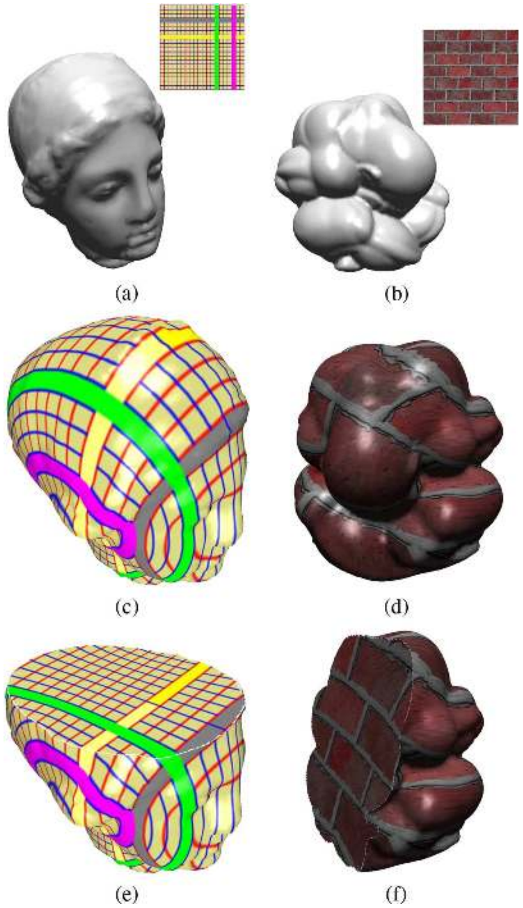


Fig. 12. Solid texture synthesis. In the first column, the solid Igea model and a corresponding 2-D image texture are shown in (a). The surface texture is firstly mapped to the Igea as illustrated in (c). We synthesize the interior solid texture and illustrate a cross section view in (e). Similarly, in the second column, the solid Pensatore model and its 2-D texture are shown in (b), the surface texture mapping and the synthesized solid textures are visualized in (d) and (f).

metric maps. In (i), a tetrahedralization of the solid Max-Planck model (g) is illustrated in one cross section. Under the volumetric mapping, each vertex of the tetrahedral mesh is mapped to a new position inside the solid sphere. Its corresponding tetrahedral mesh cross section is shown in (j).

In Fig. 7, we visualize the volumetric mapping from a solid polycube model (a) to a solid Buddha model (e). We color-code the distance field of the interior region of Buddha and show it from three cross sections in (b), (c) and (d); (f), (g) and (h) correspondingly show the *Transferred Color-Coded Distance Field*. Fig. 8 shows another high genus volumetric mapping example from a Sculpture model (a) to the polycube model (b). (c) color-codes the distance field of the polycube while (d) shows the *Transferred Color-Coded Distance Field*.

A. Information Transfer

Once the correspondence between two volume models has been established, we can easily transfer information from one object to the other. The previous *Color-Coded Distance Field Transfer* method already demonstrates this. The information being transferred can be all kinds of volumetric functions, no just limited to color; it can be material, solid texture, density, and even more complicated ones such as gradient or strain/stress tensors. Therefore, it has potential applications in a larger scope. Fig. 9 shows another example. When the Moai model deforms, the material information on the original model is transferred and preserved by the deformed model during the deformation. In the second row, we also show the corresponding tetrahedral mesh from the same corresponding cross section to visualize this transfer. We believe harmonic volumetric mapping will serve for automatic interior region registration on real temporal or deforming data in the future.

B. Shape Matching and Analysis

A direct application of mapping is registration. Based on a good registration, we can easily measure the difference between two objects in a quantitative way and perform both qualitative and quantitative analysis. We use an example to demonstrate the usage of volumetric mappings on shape matching and analysis. In this experiment, we analyze a horse-gallop deformation sequence. We use the vertex correspondence provided in the deformation sequence as the boundary surface mapping. Then we compute the volumetric harmonic mappings from the referenced, static-standing horse model to all the deformed poses in a sequence. With the mapping, we can compute their deformation energies. This energy naturally measures the distance from the deformed shape to the reference model. Since we have mappings between objects, we can get not only a numerical distance value, but also the precise error distribution between two shapes. This distribution can be clearly visualized under this deformation, where stretching and bending concentrate over the shape. Note that, as we discussed in Section IV-F, this procedure, with the computation of a large number of volumetric mappings, is performed efficiently by reusing one decomposition result.

The deformation energies of the horse gallop sequence are shown in Fig. 10. We can easily see from the energy chart that there are four running cycles in the data-set of the deformation sequence. And with the deformation energy, we naturally measure how different each model is from the reference model. The distributions of the deformation energy required from the reference model to the deformed model are color-coded and illustrated. Given a sampled model in the deformation sequence, which regions have high deformation energy concentration can be clearly visualized from the color-coded distribution of the deformation energy, as we depicted on the original model with cross sections. Without this correspondence, this kind of visualization and analysis is impossible.

C. Tetrahedral Remeshing

Regular tetrahedral mesh structure is highly desirable for finite element analysis and physically based deformations or simulations. This is because regular tetrahedralization provides great precision and efficiency for geometry processing and physically based computation [32]. With our volumetric mapping, we can easily transfer the tetrahedralization of an object to another object. We call this *tetrahedral remeshing*. As shown in Fig. 11, we use the regular tetrahedral mesh of a solid polycube model [Fig. 7(a)] to remesh the solid Buddha model [Fig. 7(e))]. (a), (b) and (c) show the tetrahedral mesh on the polycube from three cross sections; (d), (e) and (f) show the remeshed solid Buddha model. Tetrahedralization for regular shapes like polycubes can be easily created as shown in this example. So using our mapping, we can generate regular tetrahedral structure for complicated objects.

D. Volume Texture Synthesis

We can also synthesize volumetric texture using our method. As shown in Fig. 12, given a 2-D texture image, we get the surface texture mapping, then the texture applied on the surface can be smoothly propagated to the interior regions of solid objects. To synthesize the interior texture, we only need to make a change on the boundary condition; instead of using the target boundary points positions, we use the texture (u, v) coordinates. Fig. 12(a) shows an solid Igea model; and we map a 2-D image texture onto its surface as shown in (c). This texture is smoothly extrapolated into the interior region using our method. (e) illustrates the synthesized solid texture. (b), (d) and (f) show another example on the Pensatore model. From the given 2-D image, we can synthesize the volumetric texture to decorate the solid interior for graphics applications.

VI. Conclusion

Based on *the method of fundamental solution* (MFS), we design a simple, robust, and fully automatic meshless algorithm to compute harmonic volumetric maps. To the best of our knowledge, it is the first attempt to bring this method into graphics and geometric modeling community. We conduct experiments to evaluate the performance of the method of fundamental solution on the harmonic volumetric mapping problem in this paper; accordingly, we suggest the practical rules and develop the effective algorithm on the MFS settings. Then we demonstrate our mapping results in several applications, such as information transfer, deforming shape comparison and analysis, tetrahedral remeshing, and solid texture synthesis, all of which in turn show the strong potential of harmonic volumetric mapping.

Building correspondence between solid models and canonical/regular objects provides a natural mechanism to facilitate scientific computations and graphical simulations. If we exploit the regular structure of mapped volumetric domains (such as polycubes) and utilize graphics hardware acceleration, physically based simulations (such as simulating volumetric solid deformations or fluids in deformable bodies) can be efficiently performed.

As discussed in Section IV-D, our current harmonic volumetric map depends on the boundary surface mapping. In Fig. 2, we show that the volumetric mapping and its boundary surface mapping are closely related to each other. The harmonic energy of the volumetric map keeps decreasing with boundary surface mapping getting more smooth. In the future work, we plan to use the harmonic volumetric mapping to guide the variational process of surface mapping towards the global energy optimization (both for boundaries and solid interiors). Another possible extension is not to fix the positions of source points and collocation points. We can treat them as unknown variables in the MFS procedure. Although this results in a nonlinear optimization process, it may also lead to a free-boundary volumetric mapping procedure for better mapping results.

Acknowledgment

The deformation sequence data is courtesy of Sumner and Popovic's deformation transfer project. The Pensatore, Pierrot models are available in AIM@SHAPE Shape Repository. The authors would also like to acknowledge people who were involved in discussions related to this paper's topics, especially Y. Bao, H. Zeng, N. Hao, and Y. Wang for their valuable technical opinions.

References

[1] U. Pinkall and K. Polthier, "Computing discrete minimal surfaces and their conjugate," *Exper. Math.*, vol. 2, no. 1, pp. 15–36, 1993.

[2] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, "Multiresolution analysis of arbitrary meshes," in *Proc. SIGGRAPH*, 1995, pp. 173–182.

[3] M. Desbrun, M. Meyer, and P. Alliez, "Intrinsic parameterizations of surface meshes," *Comput. Graph. Forum*, vol. 21, no. 3, pp. 209–218, 2002.

[4] M. S. Floater and K. Hormann, "Surface parameterization: A tutorial and survey," *Adv. Multiresolution Geometric Model.*, pp. 157–186, 2005.

[5] T. Kanai, H. Suzuki, and F. Kimura, "Three-dimensional geometric metamorphosis based on harmonic maps," *Vis. Comput.*, vol. 14, no. 4, pp. 166–176, 1998.

[6] A. W. F. Lee, D. Dobkin, W. Sweldens, and P. Schröder, "Multiresolution mesh morphing," in *Proc. SIGGRAPH*, 1999, pp. 343–350.

[7] T. Michikawa, T. Kanai, M. Fujita, and H. Chiyokura, "Multiresolution interpolation meshes," in *Proc. Pacific Graphics*, 2001, pp. 60–69.

[8] E. Praun, W. Sweldens, and P. Schröder, "Consistent mesh parameterizations," in *Proc. SIGGRAPH*, 2001, pp. 179–184.

[9] V. Kraevoy and A. Sheffer, "Cross-parameterization and compatible remeshing of 3-D models," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 861–869, 2004.

[10] J. Schreiner, A. Asirvatham, E. Praun, and H. Hoppe, "Inter-surface mapping," in *Proc. SIGGRAPH*, 2004, vol. 23, no. 3, pp. 870–877.

[11] Y. Wang, X. Gu, and S. T. Yau, "Volumetric harmonic map," *Commun. Inf. Syst.*, vol. 3, no. 3, pp. 191–202, 2004.

[12] T. Ju, S. Schaefer, and J. D. Warren, "Mean value coordinates for closed triangular meshes," in *Proc. SIGGRAPH*, 2005, vol. 24, no. 3, pp. 561–566.

[13] M. S. Floater, "Mean value coordinates," *Comput. Aided Geomet.*, vol. 20, no. 1, pp. 19–27, 2003.

[14] P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki, "Harmonic coordinates for character articulation," in *Proc. SIGGRAPH'07*, 2007, pp. 71–81.

[15] P. K. Banerjee*, The Boundary Element Methods in Engineering*. New York: McGraw-Hill, 1994.

[16] D. L. James and D. K. Pai, "ArtDefo: Accurate real time deformable objects," in *Proc. SIGGRAPH '99*, 1999, pp. 65–72.

[17] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl, "Meshless methods: An overview and recent developments," *Comput. Meth. Appl. Mech. Eng.*, vol. 139, pp. 3–47, 1996.

[18] X. Guo, X. Li, Y. Bao, X. Gu, and H. Qin, "Meshless thin-shell simulation based on global conformal parameterization," *IEEE Trans. Visual. Comp. Graph.*, vol. 12, no. 3, pp. 375–385, 2006.

[19] G. Fairweather and A. Karageorghis, "The method of fundamental solution for elliptic boundary value problems," *Adv. Comput. Math.*, vol. 9, no. 1–2, pp. 69–95, 1998.

[20] P. R. Garabedian*, Partial Differential Equations*. New York: American Mathematical Society, 1998.

[21] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha, "Fast distance queries with rectangular swept sphere volumes," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2000, pp. 3719–3726.

[22] J. Bloomenthal, "An implicit surface polygonizer," in *Graphics Gems IV*, 1994, pp. 324–349.

[23] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Matching 3-D models with shape distributions," in *Proc. Int. Conf. Shape Model. Appl.*, 1994, p. 154.

[24] P. A. Ramachandran, "Method of fundamental solutions: Singular value decomposition analysis," *Comun. Numer. Meth. Eng.*, vol. 18, no. 11, pp. 789–801, 2002.

[25] R. Zayer, C. Rössl, Z. Karni, and H.-P. Seidel, "Harmonic guidance for surface deformation," *Comput. Graph. Forum*, vol. 24, no. 3, pp. 601–609, 2005.

[26] X. Li, Y. Bao, X. Guo, X. Gu, and H. Qin, "Global optimal surface mapping for shapes of arbitrary topology," *IEEE Trans. Visualiz. Comput. Graph.*, vol. 14, no. 4, pp. 805–819, Apr. 2009.

[27] H. Si [Online]. Available: http://tetgen.berlios.de/, ver. 1.41, 2006

[28] T. Kitagawa, "On the numerical stability of the method of fundamental solutions applied to the Dirichlet problem," *Jpn. J. Appl. Math.*, vol. 35, pp. 507–518, 1988.

[29] M. A. Golberg and C. S. Chen, "The method of fundamental solutions for potential, Helmholtz and diffusion problems," *Boundary Integral Methods: Numerical and Mathematical Aspects (A99-30801 07-64)*, vol. 1, pp. 103–176, 1999.

[30] A. Bogomolny, "Fundamental solutions method for elliptic boundary value problems," *SIAM J. Numer. Anal.*, vol. 22, pp. 644–669, 1985.

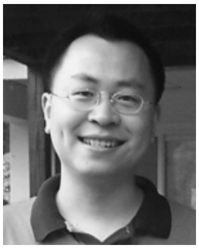
[31] R. Tanklevich, G. Fairweather, A. Karageorghis, and Y. S. Smyrlis, "Potential field based geometric modeling using the method of fundamental solutions,," *Int. J. Numer. Meth. Eng.*, vol. 68, no. 12, pp. 1257–1280, 2006R. Tanklevich, G. Fairweather, A. Karageorghis, and Y. S. Smyrlis, "Potential field based geometric modeling using the method of fundamental solutions," Tech. Rep..

[32] R. Bridson, J. Teran, N. Molino, and R. Fedkiw, "Adaptive physics based tetrahedral mesh generation using level sets," *Eng. Comput.*, vol. 21, pp. 2–18, 2005.

Xin Li (M'07) received the B.S. degree in 2003 from the University of Science and Technology of China, and the M.S. and Ph.D. degree sfrom State University of New York (SUNY), Stony Brook, both in computer science in 2005 and 2008, respectively.

He is an Assistant Professor in Department of Electrical and Computer Engineering, and Center for Computation and Technology, Louisiana State University. His research interests include visual and geometric computing, computer graphics, geometric modeling and processing, and visualization. His recent work includes curve analysis/comparison, general 3-D shape (surface and volumetric data) mapping, and their applications in graphics, vision, visualization, physical simulation, and CAGD..

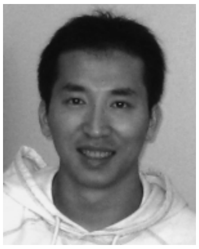
He is a member of ACM. For more information about Xin Li and shape mapping, please visit http://www.cs.sunysb.edu/~xinli.



Xiaohu Guo (M'06) received the B.S. degree in 2001 from the University of Science and Technology of China, and the M.S. and Ph.D. degrees in computer science from Stony Brook University (SUNY), Stony Brook, NY, in 2004 and 2006, respectively.

Currently, he is an Assistant Professor of computer science at the University of Texas at Dallas His research interests include computer graphics, animation and visualization, with an emphasis on geometric and physics-based modeling. His research is funded by the National Science Foundation, and he is currently the principal investigator for projects related both to physical simulation of deformable models and to geometric mapping of surface and volumetric models. For more information, please visit http://www.utdallas.edu/~xguo.

Dr. Guo is a member of the IEEE Computer Society.



Hongyu Wang received the B.S. degree in 1997 from Jilin University, Jilin, China, the M.S. degree in 2002 from the National University of Singapore, and the M.S. degree in 2006 from the State University of New York (SUNY), Stony Brook, all in computer science. He is currenty pursuing the Ph.D. degree in the Department of Computer Science at SUNY.

His research interests include computer graphics, geometric modeling, and computer aided geometric design. For further information, please visit http://www.cs.sunysb.edu/~wanghy.

Ying He received the Ph.D. degree in computer science from State University of New York (SUNY), Stony Brook, in 2006 and the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 2000 and 1997, respectively.

He is an Assistant Professor at School of Computer Engineering, Nanyang Technological University. His research interests include computer graphics, computer-aided design and visualization. For details, please visit http://www.ntu.edu.sg/home/yhe.

Xianfeng Gu (M'06) received the Ph.D. degree in computer science from Harvard University, Cambridge, MA, in 2003.

He is an Assistant Professor of computer science at the State University of New York (SUNY), Stony Brook. His research interests are computer graphics, computer vision, and medical imaging. His major works include geometry images, global conformal surface parameterization, manifold splines, and computational conformal geometry. For more information, see http://www.cs.sunysb.edu/~gu.

Dr. Gu won the U.S. National Science Foundation CAREER award in 2004. He is a member of the IEEE Computer Society.

Hong Qin received the B.S. degree in 1986 and the M.S. degree in 1989 in computer science from Peking University, Beijing, China. He received the Ph.D. degree in computer science from the University of Toronto, Toronto, ON, Canada, in 1995.

He is a Full Professor of Computer Science at the State University of New York (SUNY), Stony Brook.

Prof. Qin was awarded NSF CAREER Award from the National Science Foundation (NSF) in 1997. In December 2000, he received Honda Initiation Award. In February 2001, he was selected as an Alfred P. Sloan Research Fellow by the Sloan Foundation. In June 2005, he served as the general Co-Chair for Computer Graphics International 2005 (CGI'2005). Currently, he is an Associate Editor for IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS (IEEE TVCG), and he is also on the editorial board of *The Visual Computer (International Journal of Computer Graphics*) and *Journal of Computer Science and Technology* (JCST). He was the Conference Co-Chair for ACM Solid and Physical Modeling Symposium in 2007. In 2008, he co-chaired the ACM Symposium on Solid and Physical Modeling and IEEE International Conference on Shape Modeling and Applications. For details, please refer to http://www.cs.sunysb.edu/~qin.