

# Harmonic Volumetric Mapping for Solid Modeling Applications

Xin Li\*  
SBU

Xiaohu Guo †  
UT Dallas

Hongyu Wang ‡  
SBU

Ying He §  
NTU

Xianfeng Gu ¶  
SBU

Hong Qin ||  
SBU

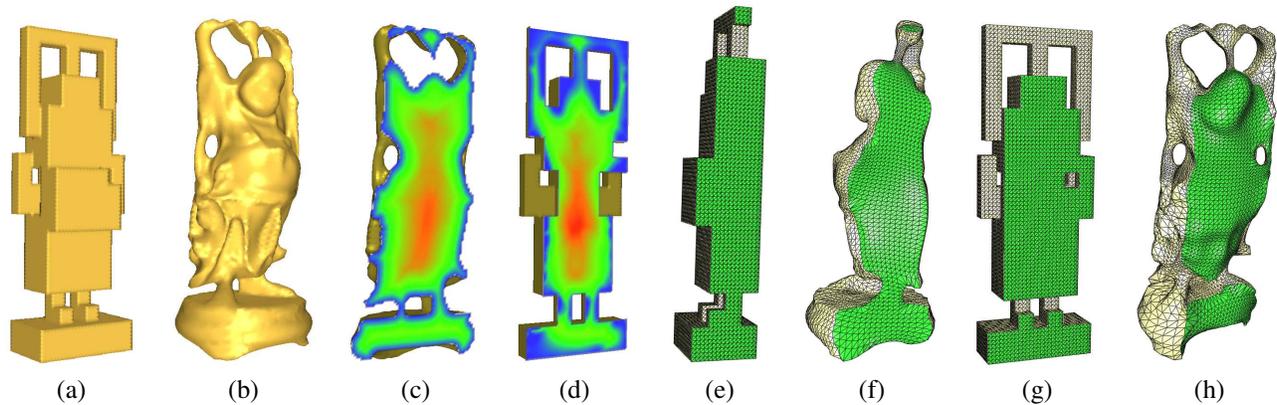


Figure 1: Harmonic volumetric mapping from a solid polycube model(a) to the solid Buddha model(b). (c) is the color-coded distance field of the Buddha interior. This color-coded distance field is transferred from the Buddha to the polycube model as shown in (d). (e) and (g) show the tetrahedral mesh of the polycube model with two different cross-sections. It is utilized to *remesh* the solid Buddha model; and the results are visualized with corresponding cross-sections in (f) and (h), respectively.

## Abstract

Harmonic volumetric mapping for two solid objects establishes a one-to-one smooth correspondence between them. It finds its applications in shape registration and analysis, shape retrieval, information reuse, and material/texture transplant. In sharp contrast to harmonic surface mapping techniques, little research has been conducted for designing volumetric mapping algorithms due to its technical challenges. In this paper, we develop an automatic and effective algorithm for computing harmonic volumetric mapping between two models of the same topology. Given a boundary mapping between two models, the volumetric (interior) mapping is derived by solving a linear system constructed from a boundary method called the *fundamental solution method*. The mapping is represented as a set of points with different weights in the vicinity of the solid boundary. In a nutshell, our algorithm is a true meshless method (with no need of specific connectivity) and the behavior of the interior region is directly determined by the boundary. These two properties help improve the computational efficiency and robustness. 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 shape registration, information reuse, deformation sequence analysis, tetrahedral remeshing and solid texture synthesis.

**Keywords:** Volumetric Mapping, Harmonic Mapping, Shape Registration and Analysis, Information Reuse, Solid Texture Synthesis

## 1 Introduction

With the rapid development of modern 3D digital acquisition techniques, more and more volumetric objects are routinely obtained and stored in shape repositories. How to effectively analyze, compare, and search these objects and reuse their information becomes a technical challenge in the fields of graphics, vision, and computer aided geometric design. The difficulties stem from the fact that (1) shapes of the modeled objects vary significantly, (2) they are acquired from different viewing positions, and (3) oftentimes shapes are deformable with time-varying behaviors. One viable approach for the 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 volumetric objects. Fundamentally, building the correspondence between objects is equivalent to seeking a mapping from one domain to another, which becomes one of the key issues nowadays in graphics and solid modeling fields. Two dimensional surface mappings and three dimensional volumetric mappings are most relevant and deserve extensive research investigation.

Computing correspondence between two surfaces has been widely studied, usually for surface deformation or morphing purpose. Its variation, parameterizing a surface onto planar domains also becomes a central research topic in the graphic and modeling area. This arose from applications such as texture mapping; and it aids in many scientific computations such as providing domains for continuous surface spline construction and physically-based simula-

\*SBU: Stony Brook Univ.,USA. e-mail: xinli@cs.sunysb.edu

†email: xguo@utdallas.edu

‡e-mail: wanghy@cs.sunysb.edu

§NTU: Nanyang Tech. Univ., Singapore. e-mail: yhe@ntu.edu.sg

¶e-mail: gu@cs.sunysb.edu

||e-mail: qin@cs.sunysb.edu

tion 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). Not only the thin-shell of the object but the whole 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 a better 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 fundamental solution method.

Harmonicity of the maps 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 mapping that leads to the physically-natural final 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 the two given solid objects. Then we seek a smooth interior region mapping by enforcing 3D 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 used 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 3D 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 accel-

eration. Third, our method can be used for a solid texture synthesis, which creates solid texture from the object's boundary surface texture mapping.

Our specific contributions are threefold:

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 fundamental solution method into the graphic modeling community. The technique is an efficient meshless boundary method with great potential. Earlier work provides some theoretical analysis from the mathematics and mechanical engineering point of view, 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 of using this method in the modeling area.
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 a tool for future graphics and modeling research.

We will briefly review the related literature in Section 2. Then we introduce the theory and algorithm of our method in Section 3, followed by Section 4 with some implementation details and property discussions. Finally, we demonstrate our experimental results with some applications in Section 5 and conclude our work in Section 6.

## 2 Related Work and Background Knowledge

Harmonic maps have been extensively studied in the literature of surface parameterization. It is addressed from the point of view of minimizing Dirichlet Energy. The discrete version of harmonic energy was first proposed by Pinkall and Polthier [Pinkall and Polthier 1993] and later introduced to computer graphics field in work of Eck et al. [Eck et al. 1995]. By discretizing the energy defined in [Pinkall and Polthier 1993], Desbrun et al. [Desbrun et al. 2002] constructed free-boundary harmonic maps. More harmonic and conformal maps are studied and surveyed by Floater and Hormann [Floater and Hormann 2005]. The harmonic maps between surfaces arose from shape blending [Kanai et al. 1998] and are widely used in shape morphing application later [Lee et al. 1999; Michikawa et al. 2001; Praun et al. 2001; Kraevoy and Sheffer 2004; Schreiner et al. 2004].

Harmonicity in volumetric sense is similarly defined as the vanishing Laplacian, representing the smoothness of the mapping function. Wang et al. [Wang et al. 2004b] studied the formula of harmonic energy defined on tetrahedral mesh and computed the discrete volumetric harmonic maps via a variational procedure. Ju et al. [Ju et al. 2005] generalized the mean value coordinates [Floater 2003] from surfaces to volumes and built a smooth volumetric interpolation based on this.

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 [Banerjee 1994], and was introduced into computer graphics for the simulation of deformable objects in [James and Pai 1999]. 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 [Belytschko et al. 1996; Guo et al. 2006]. A comprehensive review of the MFS and kernel functions for solving many elliptic PDE problems was documented in [Fairweather and Karageorghis 1998].

### 3 Theory and Algorithm

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

Our problem setting is 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 components for three axes as  $f_1, f_2$ , and  $f_3$ . In each direction,  $f_i$  maps the point  $\mathbf{p}$  onto a component  $q_i$  of  $\mathbf{q}$ . This problem is reduced to the computations of three separate  $f_i$ , with boundary mapping constraints in each corresponding axis.

#### 3.1 An Intuitive Explanation of Our Ideas

We first introduce our ideas in an intuitive way from the electrostatics point of view. Our target is the smooth mapping in each direction:  $f_i$ . It is just like building up a smooth scalar field defined in the region  $M_1$  with given boundary conditions. This smoothness can be simulated using an electric field. Many electric particles are placed outside of  $M_1$ ; each of them has some amount of charges. If the electric field of these particles simulates the boundary condition (or in other words, we make the electric potential on the boundary surface  $\partial M_1$  to have the  $f_i$ 's boundary constraint value); then we can use this electric particle system to simulate our mapping, which is rigorously written as a partial differential equation with boundary conditions in Section 3.2. The electric field provides a correct simulation for our problem because the electric field is harmonic, and it satisfies the vanishing Laplacian everywhere in  $\mathbb{R}^3$ , except for the positions where these particles are placed. Therefore, we also call these particles *singularity points* or *source points*, and we place them outside of  $M_1$ .

Since the electric field guarantees the vanishing Laplacian everywhere, we only need to enforce the boundary conditions with these particles. If we fix their positions, how many charges each particle should carry are the freedoms we try to solve in order to enforce the boundary constraint. When all these freedoms are settled down, the value of  $f_i$  on any interior point is the electric potential in that position. In Section 3.2, we will show the kernel function of the fundamental solution method to 3D harmonic problem has the same formula of how we compute potential in the electric field.

Intuitively, if we have dense enough particles placed outside of  $M_1$ , a boundary condition is always able to be well approximated, except in some highly discontinuous boundary regions. For more details we refer readers to [Garabedian 1998].

#### 3.2 Problem Formulations

We now formulate the volumetric mapping problem as follows:

Given a mapping  $\vec{f}'$  between the boundary surfaces  $\partial M_1$  and  $\partial M_2$ :  $\vec{f}'(\mathbf{p}) = \mathbf{q}$ ,  $\mathbf{p} \in \partial M_1$ ,  $\mathbf{q} \in \partial M_2$ , our goal is to compute a mapping  $\vec{f}: M_1 \rightarrow M_2$  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  $\Delta$  is defined continuously in 3D 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  $i = 1, 2, 3$ .

Since  $\Delta$  is a linear self-adjoint differential operator, we can compute its Green Function: given  $\Delta f_i(\mathbf{x}) = g_i(\mathbf{x})$ , we denote  $\Delta^{-1}$  as the inverse of the operator  $\Delta$ ,  $\Delta \Delta^{-1} = I$  where  $I$  is the identity operator; and we can write the solution as  $f_i(\mathbf{x}) = \Delta^{-1} g_i(\mathbf{x})$ .

Note that  $\Delta^{-1} g_i(\mathbf{x}) = \int K(\mathbf{x}, \mathbf{x}') g_i(\mathbf{x}') d\mathbf{x}'$ , here the kernel  $K(\mathbf{x}, \mathbf{x}')$  is the Green's function associated with  $\Delta$ , and we want that it satisfies  $\Delta K(\mathbf{x}, \mathbf{x}') = \delta(\mathbf{x} - \mathbf{x}')$  where  $\delta(\mathbf{x} - \mathbf{x}')$  is the Dirac function.

The solution to the above equation can thus be written in terms of the Kernel function as

$$f_i(\mathbf{x}) = \int K(\mathbf{x}, \mathbf{x}') g_i(\mathbf{x}') d\mathbf{x}'.$$

The Kernel function of 3D Laplacian operator is known to be  $K(\mathbf{x}, \mathbf{x}') = \frac{1}{4\pi |\mathbf{x} - \mathbf{x}'|}$ , where  $|\mathbf{x} - \mathbf{x}'|$  denotes the distance between the points  $\mathbf{x}$  and  $\mathbf{x}'$ .

Since  $f_i$  in the interior region is determined by the boundary values, we solve it using *Fundamental Solution Method* (MFS) [Fairweather and Karageorghis 1998] 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 is now represented in an exact manner; we only need to enforce a given boundary condition function  $\vec{f}'$ , or  $f_i$  for each  $f_i$ . The approximation equation we use to evaluate  $f_i$  on an interior or boundary point  $\mathbf{p}$  is

$$f_i(\vec{w}, \vec{\mathbf{Q}}; \mathbf{P}) = \sum_{n=1}^{N_s} w_n \cdot k(\mathbf{P}, \mathbf{Q}_n), \mathbf{P} \in M_1. \quad (1)$$

In the above equation,  $\vec{w} = (w_1, w_2, \dots, w_{N_s})^T$  is the vector of unknown coefficients we want to solve, representing the charges carried by each particle, where  $N_s$  is the number of source points. In addition,  $\vec{\mathbf{Q}}$  is a  $3N_s$ -dimensional vector containing positions of all  $N_s$  three-dimensional source points. As we introduced above,  $\mathbf{Q}_n \in \mathbb{R}^3$ ,  $n = 1, \dots, N_s$  lies outside of  $M_1$ , in other words, they are placed on the boundary  $\partial \widetilde{M}_1$  of a region  $\widetilde{M}_1$  containing  $M_1$  (i.e.  $M_1 \subset \widetilde{M}_1$ ).

Note that, with arbitrary  $\vec{w}$ ,  $f_i$  satisfies the Laplacian operator in an exact manner in the interior region but usually violates the boundary conditions. Hence we solve the set of expansion constraints  $\vec{w}$  to fit the boundary conditions. This fitting process reduces to a linear system:

$$\mathbf{A} \vec{w} = \vec{b}_i.$$

Matrix  $\mathbf{A}$  has its element  $A_{r,s} = k(\mathbf{P}_r, \mathbf{Q}_s)$  with  $\mathbf{P}_r$  being the *constraint point* (also called *collocation point*) on the boundary  $\partial M_1$ , and  $\mathbf{Q}_s$  being the *source point* on  $\partial \widetilde{M}_1$ .  $\vec{b}_i$  is the vector of images of these collocation points under the boundary condition, i.e.  $\vec{b}_i = \{b_{i_1}, b_{i_2}, \dots, b_{i_{N_c}}\}^T = \{f'_i(\mathbf{P}_1), f'_i(\mathbf{P}_2), \dots, f'_i(\mathbf{P}_{N_c})\}^T$ , where  $N_c$  is the number of collocation points.

### 3.3 Algorithm

The algorithmic flow is as follows:

In: Two solid objects  $M_1$  and  $M_2$ , and their boundary surface mapping  $\vec{f} : \partial M_1 \rightarrow \partial M_2$ .

Out: 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:  $\frac{\partial^2 \vec{f}}{\partial x^2} + \frac{\partial^2 \vec{f}}{\partial y^2} + \frac{\partial^2 \vec{f}}{\partial z^2} = 0$ .

- (1) Place the source points and the collocation points. (Section 4.1 and Section 4.5).
- (2) Compute the coefficient matrix using the method of fundamental solutions with respect to source points  $\mathbf{Q}_s$  and collocation points  $\mathbf{P}_r$  according to Equation 1.
- (3) Decompose the coefficient matrix using Singular Value Decomposition. (Section 4.2 and Section 4.6).
- (4) Solve this linear system with input boundary mapping constraints  $\vec{f}'$  using the decomposition result from Step (3), and get the harmonic volumetric mapping  $\vec{f}$  (represented by Equation 1).

The resultant volumetric mapping is harmonic, guaranteed by the kernel function we used. It minimizes the harmonic energy, which will be discussed in Section 4.3.1. We assume the initial boundary surface mapping is given as an input, and in Section 4.4, we briefly discuss how to obtain this surface mapping with existing techniques.

## 4 Implementation and Discussion

### 4.1 Placing Source Points and Collocation Points

We place the source points uniformly on an offset surface near the boundary surface of the solid model, as shown in Figure 2(b), the following procedure is a robust way to create such a sampling. The reason that we conduct the source points placement in this way will be discussed in Section 4.5. The algorithm makes use of the implicit form of the surface boundary of the solid model:

- (1) Compute the distance field in  $\mathbb{R}^3$  of the given object boundary surface  $\partial M_1$  using technique introduced in [Larsen et al. 2000]. We get a distance evaluation function  $dist(\mathbf{p})$  that the signed distance from any point  $\mathbf{p} \in \mathbb{R}^3$  to  $\partial M_1$  can be computed efficiently.
- (2) Build an offset surface  $\partial \widetilde{M}_1$  using Bloomenthal’s polygonization method [Bloomenthal 1994]. It 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  $\partial M_1$ , we set such input function  $dist2(\mathbf{p}) = dist(\mathbf{p}) + d$ .
- (3) Uniformly sample  $n$  points on mesh  $\partial \widetilde{M}_1$ .

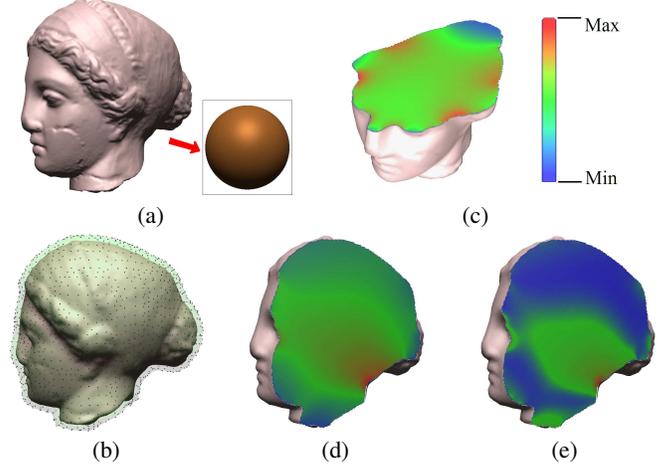


Figure 2: Volumetric harmonic mapping from the solid Igea model to a solid sphere model. (a) The source and target solid objects are shown. (b) Source points are placed on an offset surface. (c) and (d) The harmonic energy distribution of the mapping is color-coded and illustrated on two different cross-sections. (e) The deformation energy distribution is illustrated over one cross-section.

The collocation points are chosen from the vertices  $\mathbf{p}$  on the boundary surface of source solid model ( $\mathbf{p} \in \partial M_1$ ).

### 4.2 Solving the Linear System with the Dense Coefficient Matrix

As discussed in Section 3.2, we want to solve the linear system  $\mathbf{A}\vec{w} = \vec{b}$ . 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 [Ramachandran 2002], in which case, regular linear system solvers such as Gaussian elimination or LU decompositions often fail to produce a meaningful solution. Singular value decomposition (SVD) is a stable and robust approach for accurate results even when the coefficient matrix is ill-conditioned. Another advantage is that once we have decomposed the matrix, we can use the result for rapidly recomputing new mappings whenever boundary conditions change. This efficiency also arises from the boundary method, detailed discussion about this aspect will be given in Section 4.6. More advantages of using SVD in MFS are discussed in [Ramachandran 2002].

### 4.3 Energy of Volumetric Mapping

#### 4.3.1 Harmonic Energy

Harmonic Energy measures the smoothness of the mapping. It is measured by the integration of the square of the gradient over the interior region domain. 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

$$\int_{M_1} \langle \nabla \vec{f}, \nabla \vec{f} \rangle dx. \quad (2)$$

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

the energy:

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

where  $vol(\mathbf{p}_{u,v,w})$  is the interior volume on grid point  $\mathbf{p}_{u,v,w}$ , and  $vol(\mathbf{p}_{u,v,w})$  equals to the volume of the intersection of  $M_1$  and the small grid cube  $Cube_{\mathbf{p}_{u,v,w}}$  centered at  $\mathbf{p}_{u,v,w}$ . Here, the edge length of  $Cube_{\mathbf{p}_{u,v,w}}$  is the distance between two adjacent grids. We can use the volume of  $Cube_{\mathbf{p}_{u,v,w}}$  to approximate  $vol(\mathbf{p}_{u,v,w})$ . With the grid sampling density increasing, the value of Equation (3) is asymptotic to Equation (2). We use the simple volume grid data structure because it is easy to implement and efficient in tracing values of its neighboring grid points: the gradient of the harmonic mapping on each grid point can be represented by three vectors:  $\nabla \vec{f} = (\nabla f_1, \nabla f_2, \nabla f_3)$ , so  $\|\nabla \vec{f}(\mathbf{p}_{u,v,w})\|^2 = \|\nabla f_1\|^2 + \|\nabla f_2\|^2 + \|\nabla f_3\|^2$ . In the example of Figure 2, 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 as shown in the bar in (c): red represents maximum while blue represents minimum.

### 4.3.2 Deformation Energy

Once a correspondence between two solid objects is created, the interior voxel point's deformation can be estimated easily. This provides us a formal mechanism to compute the energy required to deform one object to another. Later we will use this energy to measure the distance between shapes in deformation sequences.

We compute the deformation energy starting from the classical strain and stress tensor analysis. Green's strain tensor  $\epsilon$  is used to quantify the local strain undergoing a 3-dimensional deformation. If a point  $\mathbf{p}$  is mapped to  $\mathbf{q}$ , then the  $3 \times 3$  tensor  $\epsilon$  has its elements  $\epsilon_{ij}$  represented as

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

where  $1 \leq i, j \leq 3$  are indices in axis directions,  $\delta_{ij}$  is the Kronecker delta:

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

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

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

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

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

Figure 2(e) color-codes the deformation energy distribution of the volumetric mapping from solid Igea model to the solid sphere.

## 4.4 Initial Boundary Surface Mapping

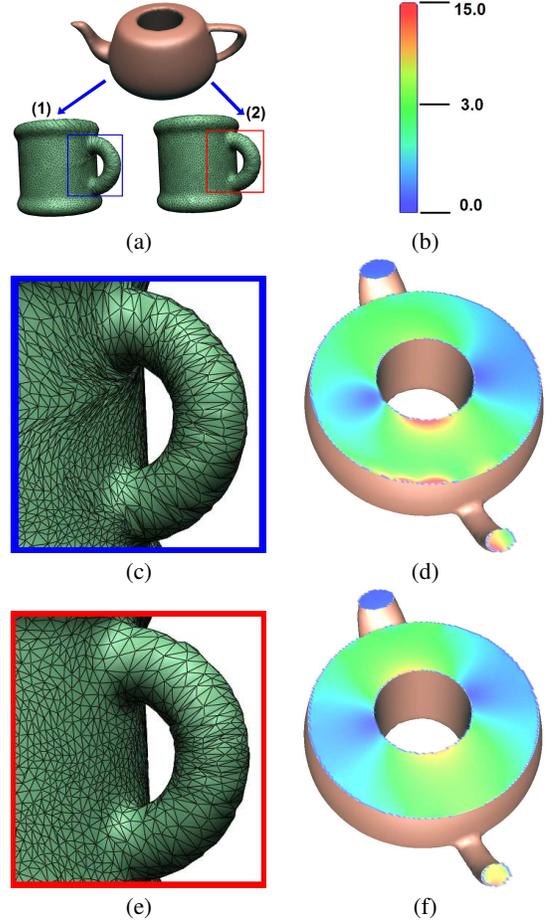


Figure 3: Different boundary conditions lead to different volumetric mapping results even for the same target object. Volumetric mappings from the solid Teapot model to a solid Cup model (a) under two different boundary mapping conditions (see (c) and (e)) have different harmonicity as shown in (d) and (f) (energy distributions are depicted on the Teapot model respectively). (c) and (e) highlight different surface mappings with magnified views.

Our proposed algorithm is simple and fully automatic once the initial boundary surface mapping is provided. Surface mapping techniques [Kanai et al. 1998; Lee et al. 1999; Michikawa et al. 2001; Praun et al. 2001; Kraevoy and Sheffer 2004; Schreiner et al. 2004; Zayer et al. 2005] can be used to build up the initial boundary mapping. We hope this surface mapping creation can be simple and automatic as well. On the other hand, the harmonic volumetric mapping depends on the initial surface mapping. How to generate such a mapping and how the quality of this mapping affects the volume mapping are worthy of more comprehensive research studies for this topic. Since the detailed discussion goes beyond the focus of this paper, we will only briefly discuss how we get a suitable 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 zero surfaces, the conformal surface mapping suffices. This map-

ping can be computed/combined through the conformal mapping between the surface and a sphere [Gu et al. 2004]. (2) If the surfaces are of higher genus, we prefer a globally smooth mapping and compute their quasiconformal mapping as the initial boundary condition using techniques introduced in [Li et al. 2007].

Volumetric mappings between two objects differ under different surface boundary mapping conditions. Our experiment demonstrates this in Fig 3. 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 boundary condition differences ((c),(e)). The second boundary mapping condition is smoother (e) 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 using a same cross-section. The color-coding scheme used in (d),(f) is as the bar depicts in (b).

In the near future, we will 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.

#### 4.5 Discussion on Source and Collocation Point Placement

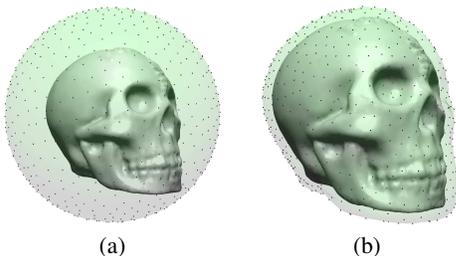


Figure 4: Source Points Placement. Source points are sampled either on a bounding sphere (a) or on an offset surface of the given model (b).

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 are now trying to refine our control of the electric field behavior using electric particles, the more particles (i.e., source points) we have, naturally the more refined result we should be able to get with the 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. The positions of these source points actually determine the behavior of the coefficient matrix  $\mathbf{A}$ , which can be highly ill-conditioned [Kitagawa 1988]. 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 [Golberg and Chen 1999]. 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 usually suggested placing source points uniformly on a sphere within three times the diameter of  $M_1$  [Golberg and Chen 1999][Bogomolny 1985], another type of suggestions [Tanklevich et al. ] is to place them on a

similar offset surface of  $M_1$ . 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 Figure 5. In this figure, Chart (a) plots the boundary constraint error when source points are placed on a sphere (see Figure 4(a)), while Chart (b) shows the case when source points are on an offset surface (see Figure 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. 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  $\partial M_1$ ; 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 that: (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 too closed 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 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 non-linear.

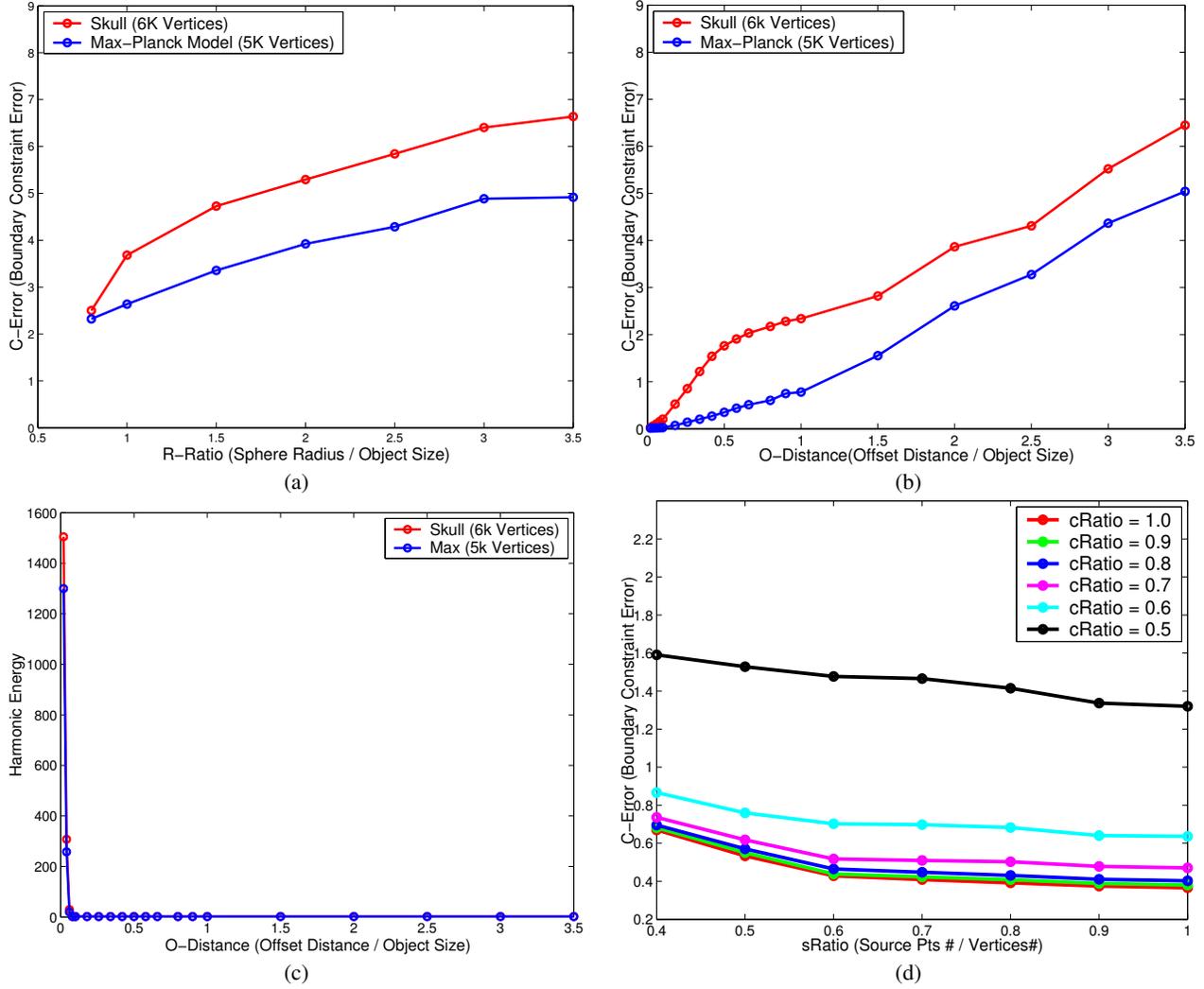


Figure 5: Volumetric mappings under different source/collocation point configurations. (a) shows the boundary constraint error(C-Error) under different R-Ratio when source points are placed on spheres. (b) and (c) plot the constraint error(b) and harmonic energy(c) respectively under different O-Distance 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.

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.

#### 4.6 Accelerating Volumetric Mappings for Time Series

We decompose the coefficient matrix  $A$  from the MFS using Singular Value Decomposition (see Section 4.2), which provides an efficient way to recompute the volumetric mapping with different boundary conditions. Under a new boundary configuration  $\vec{b}'$ , the corresponding  $\vec{w}'$  for the new volumetric mapping can be computed directly from  $A^{-1} \times \vec{b}'$ . With the decomposition results,  $A^{-1} = V \times W^{-1} \times U^T$ , where the matrix  $W^{-1}$  is a diagonal matrix, can be computed directly. Therefore, the decomposition matrix results can be reused under different boundary conditions. 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 5, 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.

#### 4.7 Comparison with Previous Work

We compare our mapping results with the method introduced in [Wang et al. 2004b]. 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 Equation (1). We tetrahedralize our volume data (in our work we produce the tetrahedralization using [Si 2006]), then compare our results with Wang et al.'s work in [Wang et al. 2004b]. As shown in Figure 6, the volumetric mapping from

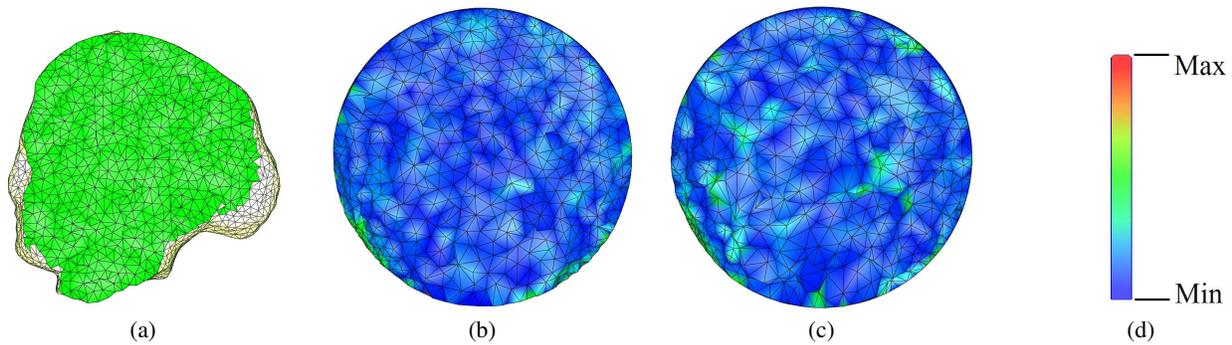


Figure 6: Comparison with previous work. The initial tetrahedralization of solid Igea model is shown in (a). In (b), the tetrahedral mesh on the solid sphere is transferred from the solid Igea model using our volumetric mapping algorithm. The result computed using Wang et al.’s variational technique is shown in (c). The harmonic energy distributions of two volumetric mappings are color-coded on resultant tetrahedra using a uniform color-coding scheme as shown in (d).

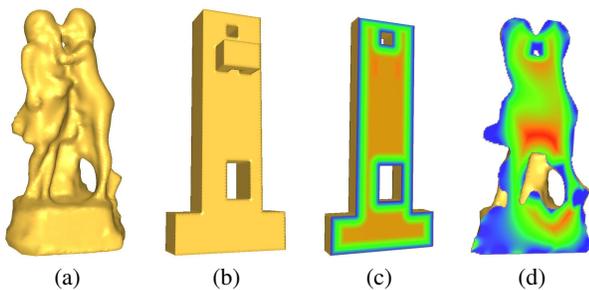


Figure 8: Harmonic volumetric mapping from the solid Sculpture model(a) to the polycube model(b). (c) color-codes the distance field of the solid polycube interior. In (d), the color-coding of the transferred distance field is visualized on the solid Sculpture model.

the solid Igea model (a) to a solid sphere can be visualized by transferring the tetrahedron mesh of the Igea to the solid sphere. Our resultant tetrahedron mesh on sphere (b) appears to be smoother than the mesh produced in [Wang et al. 2004b] (c). This smoothness can also be visualized from the distributions of the discretized harmonic energy of volumetric mappings, which are color-coded in (b) and (c) using a uniform scheme (d).

Another important advantage is the meshless property of our algorithm. The discretization accuracy and the computation time of [Wang et al. 2004b] depends heavily on the tetrahedralization quality of the source object: dense tetrahedralization necessarily results in high computational complexity; and irregular tetrahedralization leads to 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.

## 5 Experimental Results and Applications

Now we show some experimental results of harmonic volumetric mappings in Figure 7, Figure 1 and Figure 8. In Figure 7, a solid Pierrrot 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 Figure 2(c). Here in Figure 7(d), each volume point  $p$  in solid Pierrrot model (a) is mapped to an interior point  $q$  in solid sphere model (b). We transfer the color of  $q$  to the position of  $p$ . 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*. One more mapping example from a solid Max-Planck model (e) to the solid sphere (b) is computed and visualized in the same way as shown in (f). Another method to visualize volumetric maps is by the tetrahedral mesh. In (g), a tetrahedralization of solid Max-Planck model (e) 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. For tetrahedra shown in the cross-section in (g), their vertices are mapped to new positions as shown in (h).

In Figure 1, we visualize the volumetric mapping from a solid polycube model (a) to a solid Buddha model (b). (c) color-codes the distance field of the interior region of Buddha; and (d) shows the *transferred color-coded distance field*. Figure 8 shows another 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*.

### 5.1 Information Reuse

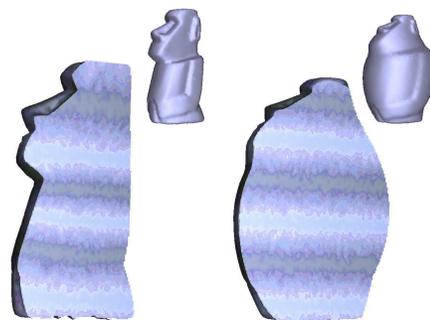


Figure 9: Volumetric harmonic mapping for information reuse. The material on the solid Moai model is preserved when it deforms during the animation.

Once the correspondence between two volume models has been established, the target object can easily reuse the information the source volume carries. The above *Color-coded Distance Field Transfer* method already demonstrates this. Figure 9 shows another example. When the Moai model deforms, the material information on the original model is preserved by the deformed model during the deformation. We can see from this example that the material

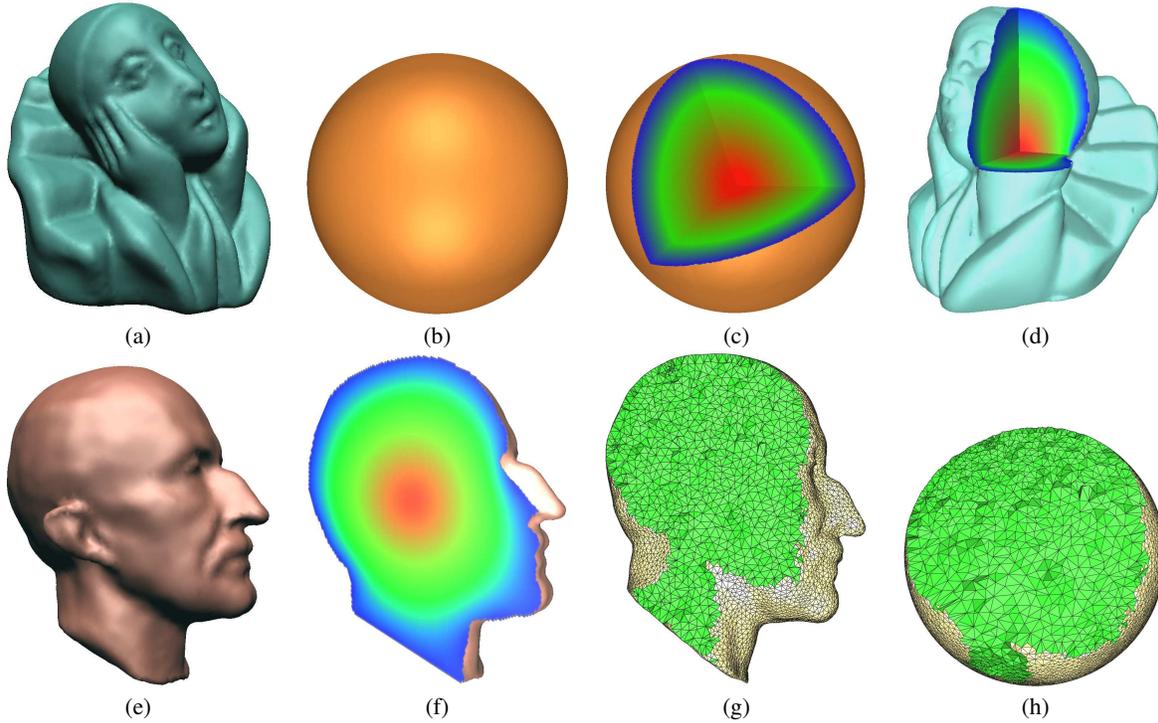


Figure 7: Volumetric mappings between solid objects and the solid sphere. The source Pierrot model is depicted in (a); the target model is a solid sphere (b). (c) shows the color-coded distance field in the solid sphere. (d) visualizes the volumetric mapping: each point  $p$  in the original model of (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)). Another example on the volumetric mapping from the Max-Planck model (e) to the solid sphere (b) is visualized in (f) by the same *color-coded distance field transfer* method. Tetrahedral meshes on the Max-Planck model (g) are mapped into the solid sphere (h). Their corresponding cross-sections are visualized in (g) and (h).

information can be transferred and reused with the help of our volumetric mapping. Information reuse and transfer have more potential applications in a larger scope, not just limited to material or solid texture, but applicable for all kinds of volumetric functions.

The direct application with the already-computed correspondence is registration, with which we can measure the difference between two objects in a quantitative way and perform both qualitative and quantitative analysis based on this matching. We will discuss this in the following sub-section.

## 5.2 Shape Matching and Analysis

We use an example to demonstrate the application 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 initial boundary surface mapping to compute the volumetric harmonic mappings from the referenced initial object to all of its sequentially deformed objects. Then we compute their deformation energies based on these mappings. This energy naturally measures the distance between the deformed shape and the original model. Since mappings from object to object have been created, not only a numerical distance value between objects, but also the error distribution can be illustrated on the solid model showing the stretching and bending of the deformation. Note that this procedure can be measured efficiently by reusing the decomposition results directly (see Section 4.6).

The deformation energies of the horse gallop sequence are shown in the Figure 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 can be color-coded and illustrated. Given a sampled model in the deformation sequence, which regions have high deformation energy concentration can be easily visualized from the color-coded distribution of the deformation energy, as we depicted on the original model with cross-sections.

## 5.3 Tetrahedral Remeshing

The tetrahedralization of an object can be transferred to another object under the volumetric mapping. We call it *tetrahedral remeshing*. As shown in Figure 1(e)-(h), we use the regular tetrahedral mesh of a solid polycube model to remesh the solid Buddha model. (e) and (g) show the tetrahedral mesh on the polycube; (f) and (h) show the remeshed solid Buddha model. Tetrahedralization for regular shapes like polycubes can be easily provided. Such highly regular connections offer great efficiency for geometry processing and computation for physically-based deformations or simulations [Bridson et al. 2005]. In addition, in order to take advantage of graphics hardware acceleration, such as modern GPUs, regular representation structure is always highly desirable.

## 5.4 Volume Texture Synthesis

Our method can also be used for volumetric texture synthesis. As shown in Figure 11, given a 2D 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

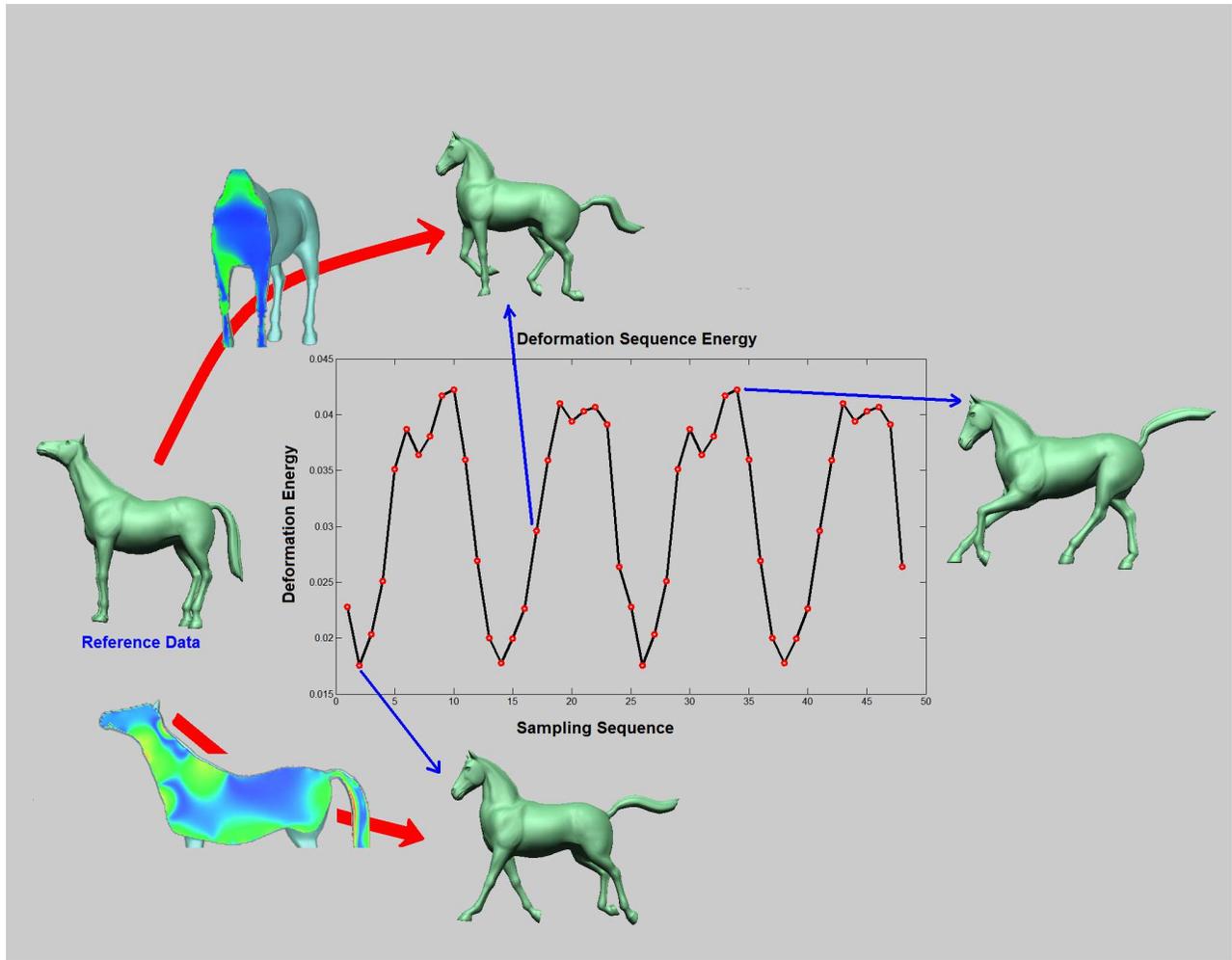


Figure 10: Energy analysis of deformation sequences. The horse model is sequentially deformed. The deformation energies are calculated (red circles). The distribution of the deformation energy required for each sequential model can be illustrated on the reference model.

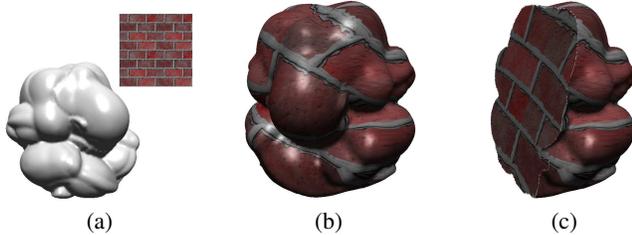


Figure 11: Solid texture synthesis. The solid Pensatore model and the image texture are shown in (a). The surface texture is firstly mapped to the Pensatore model as illustrated in (b). We synthesize the interior solid texture and illustrate a cross-section view in (c).

the boundary condition; instead of using the target boundary points positions, we use the texture  $(u, v)$  coordinates. Figure 11(a) shows an solid Pensatore model; and we map a 2D image texture onto its surface as shown in (b). This texture is smoothly extrapolated into the interior region using our method. (c) illustrates the synthesized solid texture that can be used for decorating the solid interior.

## 6 Conclusion

We have presented a simple, robust, and fully automatic method to compute harmonic volumetric maps based on a true meshless boundary method called *the fundamental solution method* (MFS). This appears to be the first attempt to bring this method into our graphics and modeling community. We conduct experiments to evaluate the performance of the fundamental solution method on the harmonic volumetric mapping problem in this paper; also 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 reuse, deformation sequence analysis, tetrahedral remeshing and solid texture synthesis, all of which in turn show the strong potential of harmonic volumetric mapping in graphics and solid modeling fields.

Building correspondence between solid models and canoni-

cal/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 4.4, our current volumetric harmonic map depends on the initial boundary surface mapping. In Figure 3, 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 smoother. In the near future, we will 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.

## Acknowledgements

The authors would like to acknowledge people who were involved in discussions related to this paper's topic, especially Yunfan Bao, Huayi Zeng, Ning Hao, and Yue Wang for their valuable technical opinions. 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.

## References

- ALLIEZ, P., COHEN-STEINER, D., YVINEC, M., AND DESBRUN, M. 2005. Variational tetrahedral meshing. *ACM Trans. Graph.* 24, 3, 617–625.
- BANERJEE, P. 1994. *The Boundary Element Methods in Engineering*. McGraw-Hill, New York.
- BELYTSCHKO, T., LU, Y. Y., AND GU, L. 1994. Element free galerkin methods. *International Journal for Numerical Methods in Engineering* 37, 229–256.
- BELYTSCHKO, T., KRONGAUZ, Y., ORGAN, D., FLEMING, M., AND KRYSL, P. 1996. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering* 139, 3–47.
- BLOOMENTHAL, J. 1994. An implicit surface polygonizer. *Graphics Gems IV*, 324–349.
- BOGOMOLNY, A. 1985. Fundamental solutions method for elliptic boundary value problems. *SIAM Journal on Numerical Analysis* 22, 644–669.
- BRIDSON, R., TERAN, J., MOLINO, N., AND FEDKIW, R. 2005. Adaptive physics based tetrahedral mesh generation using level sets. *Engineering with Computers* 21, 2–18.
- CHENG, H., GREENGARD, L., AND ROKHLIN, V. 1999. A fast adaptive multipole algorithm in three dimensions. *Journal of Computational Physics* 155, 468–498.
- DESBRUN, M., AND CANI, M. P. 1995. Animating soft substances with implicit surfaces. In *Proc. SIGGRAPH '95*, 287–290.
- DESBRUN, M., AND CANI, M. P. 1996. Smoothed particles: A new paradigm for animating highly deformable bodies. In *Proc. EG Workshop on Animation and Simulation*, 61–76.
- DESBRUN, M., MEYER, M., AND ALLIEZ, P. 2002. Intrinsic parameterizations of surface meshes. *Comput. Graph. Forum* 21, 3, 209–218.
- ECK, M., DE ROSE, T., DUCHAMP, T., HOPPE, H., LOUNSBERY, M., AND STUETZLE, W. 1995. Multiresolution analysis of arbitrary meshes. In *SIGGRAPH*, 173–182.
- FAIRWEATHER, G., AND KARAGEORGHIS, A. 1998. The method of fundamental solutions for elliptic boundary value problems. *Advances in Computational Mathematics* 9, 1-2 (September), 69–95.
- FLOATER, M. S., AND HORMANN, K. 2005. Surface parameterization: a tutorial and survey. In *Advances in Multiresolution for Geometric Modelling*, Mathematics and Visualization. Springer, Berlin, Heidelberg, 157–186.
- FLOATER, M. S. 2003. Mean value coordinates. *Computer Aided Geometric Design* 20, 1, 19–27.
- FRISKEN, S., AND PERRY, R. 2003. Simple and efficient traversal methods for quadtrees and octrees.
- GARABEDIAN, P. 1998. *Partial Differential Equations*. American Mathematical Society.
- GOLBERG, M. A., AND CHEN, C. S. 1999. The method of fundamental solutions for potential, helmholtz and diffusion problems. *Boundary integral methods: Numerical and mathematical aspects (A99-30801 07-64)* 1, 103–176.
- GU, X., WANG, Y., CHAN, T., THOMPSON, P., AND YAU, S. T. 2004. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Trans. Med. Imaging* 23, 8, 949–958.
- GUO, X., LI, X., BAO, Y., GU, X., AND QIN, H. 2006. Meshless thin-shell simulation based on global conformal parameterization. *IEEE Transactions on Visualization and Computer Graphics* 12, 3, 375–385.
- JAGNOW, R., DORSEY, J., AND RUSHMEIER, H. 2004. Stereological techniques for solid textures. *ACM Trans. Graph.* 23, 3, 329–335.
- JAMES, D. L., AND PAI, D. K. 1999. Artdefo: accurate real time deformable objects. In *Proc. SIGGRAPH '99*, 65–72.
- JU, T., SCHAEFER, S., AND WARREN, J. D. 2005. Mean value coordinates for closed triangular meshes. *SIGGRAPH* 24, 3, 561–566.
- KANAI, T., SUZUKI, H., AND KIMURA, F. 1998. Three-dimensional geometric metamorphosis based on harmonic maps. *The Visual Comput.* 14, 4, 166–176.
- KITAGAWA, T. 1988. On the numerical stability of the method of fundamental solutions applied to the dirichlet problem. *Japan Journal of Applied Mathematics* 35, 507–518.
- KRAEVOY, V., AND SHEFFER, A. 2004. Cross-parameterization and compatible remeshing of 3d models. *ACM Trans. Graph.* 23, 3, 861–869.
- LARSEN, E., GOTTSCHALK, S., LIN, M. C., AND MANOCHA, D. 2000. Fast distance queries with rectangular swept sphere volumes. In *Proc. IEEE International Conference on Robotics and Automation*, 3719–3726.
- LEE, A., DOBKIN, D., SWELDENS, W., AND SCHRÖDER, P. 1999. Multiresolution mesh morphing. In *Proc. SIGGRAPH*, 343–350.

- LI, X., BAO, Y., GUO, X., GU, X., AND QIN, H. 2007. Extremal quasiconformal surface mapping. *Manuscript submitted for publication*.
- LIU, Y., NISHIMURA, N., AND YAO, Z. 2005. A fast multipole accelerated method of fundamental solutions for potential problems. *Engineering Analysis with Boundary Elements* 29, 1016–1024.
- MICHIKAWA, T., KANAI, T., FUJITA, M., AND CHIYOKURA, H. 2001. Multiresolution interpolation meshes. In *Proc. Pacific Graphics*, 60–69.
- MONAGHAN, J. 1988. An introduction to sph. *Computer Physics Communications* 48, 89–96.
- MÜLLER, M., KEISER, R., NEALEN, A., PAULY, M., GROSS, M., AND ALEXA, M. 2004. Point based animation of elastic, plastic and melting objects. In *Proc. ACM SIGGRAPH/EG Symp. Computer Animation*, 141–151.
- MÜLLER, M., HEIDELBERGER, B., TESCHNER, M., AND GROSS, M. 2005. Meshless deformations based on shape matching. *ACM Trans. Graph.* 24, 3, 471–478.
- PAULY, M., KEISER, R., ADAMS, B., DUTRE, P., GROSS, M., AND GUIBAS, L. J. 2005. Meshless animation of fracturing solids. *ACM Trans. Graph.* 24, 3, 957–964.
- PEACHEY, D. R. 1985. Solid texturing of complex surfaces. In *SIGGRAPH*, ACM Press, New York, NY, USA, 279–286.
- PERLIN, K. 1985. An image synthesizer. In *SIGGRAPH*, ACM Press, New York, NY, USA, 287–296.
- PINKALL, U., AND POLTHIER, K. 1993. Computing discrete minimal surfaces and their conjugate. In *Experimental Mathematics*, vol. 2, 15–36.
- PORUMBESCU, S. D., BUDGE, B., FENG, L., AND JOY, K. I. 2005. Shell maps. *ACM Trans. Graph.* 24, 3, 626–633.
- PRAUN, E., SWELDENS, W., AND SCHRÖDER, P. 2001. Consistent mesh parameterizations. In *Proc. SIGGRAPH*, 179–184.
- RAMACHANDRAN, P. A. 2002. Method of fundamental solutions: singular value decomposition analysis. *Communications in Numerical Methods in Engineering* 18, 11, 789–801.
- SCHREINER, J., ASIRVATHAM, A., PRAUN, E., AND HOPPE, H. 2004. Inter-surface mapping. *SIGGRAPH*. 23, 3, 870–877.
- SI, H. 2006. <http://tetgen.berlios.de/> version 1.41.
- SUMNER, R. W., AND POPOVIC, J. Mesh data from deformation transfer for triangle meshes. <http://people.csail.mit.edu/sumner/research/deftransfer/data.html>.
- SUMNER, R. W., AND POPOVIC, J. 2004. Deformation transfer for triangle meshes. *ACM Trans. Graph.* 23, 3, 399–405.
- TANKLEVICH, R., FAIRWEATHER, G., KARAGEORGHIS, A., AND SMYRLIS, Y. S. Potential field based geometric modeling using the method of fundamental solutions. *Technical Report*.
- VORUGANTI, H. K., DASGUPTA, B., AND HOMMEL, G. 2006. A novel potential field based domain mapping method. In *Proceedings of 10th WSEAS Conference on Computers*.
- WANG, Y., GU, X., CHAN, T. F., THOMPSON, P. M., AND YAU, S. T. 2004. Volumetric harmonic brain mapping. In *ISBI '04: IEEE International Symposium on Biomedical Imaging: Macro to Nano.*, 1275–1278.
- WANG, Y., GU, X., AND YAU, S. T. 2004. Volumetric harmonic map. *Communications in Information and Systems* 3, 3, 191–202.
- YU, Y., ZHOU, K., XU, D., SHI, X., BAO, H., GUO, B., AND SHUM, H.-Y. 2004. Mesh editing with poisson-based gradient field manipulation. *ACM Trans. Graph.* 23, 3, 644–651.
- ZAYER, R., RÖSSL, C., KARNI, Z., AND SEIDEL, H.-P. 2005. Harmonic guidance for surface deformation. *Computer Graphics Forum* 24, 3, 601–609.