Technical Section

Feature-aligned harmonic volumetric mapping using MFS

Xin Li\textsuperscript{a,b,*}, Huanhuan Xu\textsuperscript{a,b}, Shenghua Wan\textsuperscript{a,b}, Zhao Yin\textsuperscript{c}, Wuyi Yu\textsuperscript{c}

\textsuperscript{a} Department of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA 70803, USA
\textsuperscript{b} Center for Computation and Technology, Louisiana State University, Baton Rouge, LA 70803, USA
\textsuperscript{c} Department of Automation, Xiamen University, Xiamen, Fujian 361005, China

\begin{abstract}

We present an efficient adaptive method to compute the harmonic volumetric mapping, which establishes a smooth correspondence between two given solid objects of the same topology. We solve a sequence of charge systems based on the harmonic function theory and the method of fundamental solutions (MFS) for designing the map with boundary and feature constraints. Compared to the previous harmonic volumetric mapping computation using MFS, this new scheme is more efficient and accurate, and can support feature alignment and adaptive refinement. Our harmonic volumetric mapping paradigm is therefore more effective for practical shape modeling applications and can handle heterogeneous volumetric data. We demonstrate the efficacy of this new framework on handling volumetric data with heterogeneous structure and nontrivial topological types.

\end{abstract}

1. Introduction

The rapid advancement of 3D scanning techniques makes it easier to acquire massive 3D data nowadays. When datasets can be acquired in an explosive rate, computational techniques only evolve modestly. As a result, 3D data matching, analyzing, and searching become bottleneck for their efficient processing. Compared with 2D images, 3D shapes have many distinctions including larger sets of degrees of freedom and spatial variations in terms of geometry, topology, feature, and material. A viable approach for the effective shape matching and analyzing is to establish the correspondence between objects of interest, which can be computed by either solving a non-rigid bijective registration between given objects or composing two parameterizations from both objects onto one common domain. The key is to compute a mapping from one domain to another. When it is possible to purely consider boundary surfaces of the 3D data, one can focus on mapping 2d-manifolds (surfaces). Surface mapping seeks a bijection between two 2-manifolds with similar topology, aiming for least distortion (using length-, angle-, or area-preserving as the criterion) which dictates its effects in applications. Surface parameterization and inter-surface mapping have been extensively studied, playing important roles in computer graphics, and serving as ubiquitous tools for many valuable applications. For example, in computer graphics, it has been used for texture mapping, texture transfer, and morphing animation. In geometric modeling, it has been used for detail transfer, surface editing, mesh simplification. In CAGD, it has been used to construct the parametric domain for continuous representations such as splines. In visualization, complicated geometric structures may be better visualized and analyzed by mapping surfaces and their properties to a simpler domain. In vision and medical imaging, it has been used for surface matching, data completion, and so on. Surveys of surface mapping and their applications are given in [13,42].

Solid volumetric data have richer contents than those of the boundary surface. When the data processing or analysis are related to material, intensity, or any other structural information defined over the whole 3D region of the object (instead of on just its boundary shell), we need to consider the shape as a 3-manifold and study the volumetric mapping. Therefore, volumetric mapping can also benefit aforementioned applications. Because of its importance, volumetric mapping and parameterization has gained greater interest in recent years, and a few related research work has been conducted towards various applications such as shape registration [47,29,30], volumetric deformation [21,20,32,5], and trivariate spline construction [33], and so on. Although many valuable concepts and demos have been presented, all indicating the importance of this technique, its study has just started and is far from adequate. Several key limitations of existing algorithms prevent them from being applied into real applications with complex scenarios.

\textit{Generality}: It is desirable that the mapping is general and can handle 3D shapes with variant topological types. Volumetric data from real scenarios usually have nontrivial topology, and most existing parameterization techniques [47,33] focus on topological solid-sphere shapes. Refs. [29,30] used the fundamental solution

\textsuperscript{*} Corresponding author at: Department of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA 70803, USA. Tel: +1 225 578 0289; fax: +1 225 578 4831.
E-mail address: xinli@lsu.edu (X. Li).
methods to compute harmonic volumetric mapping between 3D objects with general topology.

Efficiency: Solving the discretized vector field over a 3D voxelized domain or over a tetrahedral mesh usually is much slower than the surface mapping computation. The fundamental solution method of [29] is a boundary method. It reduces the volumetric mapping computation from the whole 3D domain to the degree of freedom with the boundary size, to be solved by a linear system of equations. However, it is still very time consuming to solve because the coefficient matrix is dense and ill-conditioned.

Heterogeneity: Most existing methods consider the volumetric mapping from homogeneous viewpoints and only compute the mapping purely based on geometry, without taking into account the interior structure and features. It is desirable to develop the capability of the mapping algorithm that can accommodate heterogeneous structures and integrate domain expertise in geometric modeling and processing.

In order to tackle these aforementioned limitations, this paper improves the algorithm of fundamental solution methods in mapping computation [30], and seeks a general and effective mapping computation algorithm with better efficiency, accuracy, and heterogeneity. We compute harmonic volumetric mapping by improving the fundamental solution methods of [30], and the side-by-side comparison shows that our new approach is more efficient and accurate. Furthermore, it supports feature alignment, which is important for many practical volumetric data processing tasks.

The main contributions of this work include:

- We use multiple fundamental solution systems and an adaptive refinement scheme for the computation of harmonic volumetric mapping. Compared to [30], this computation efficiency is greatly improved, so that large and complex data can be parameterized in the new framework. In the mean time, with an adaptive sampling scheme, the new computation also converges to a better boundary fitting result in salient manners.
- Our feature alignment scheme supports the computation of volumetric mapping composed by constrained harmonic functions that allow the alignment between various types of features including 0-manifolds (feature points), 1-manifolds (feature lines, such as skeletons), 2-manifolds (iso-surfaces).

The paper is organized as follows. Section 2 briefly reviews related literature. Then we introduce the theory and algorithms of our methods in Section 3, and address important implementation issues in Section 4. In Section 5, we demonstrate some experimental results, discuss and compare our algorithms with existing volumetric mapping methods, especially [30], and show the large efficiency/accuracy improvement over the current method. We also show a direct application on hex meshing. Finally, we conclude our work in Section 6.

2. Related work

Harmonic maps and surface parameterization: Surface mapping computes a one-to-one continuous map between a 2-manifold and a target domain with low distortions. It plays a critical role in various applications of graphics, CAGD, visualization, vision, medical imaging, and physical simulation. 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 [38] and later introduced to computer graphics field in work of Eck et al. [10]. By discretizing the energy defined in [38], Desbrun et al. [8] constructed free-boundary harmonic maps. Harmonic maps are directly used for shape blending [22] and in later shape morphing applications [26,39,24,41]. A lot of effective surface manipulation techniques and parameterization paradigms might be generalized onto 3-manifolds. A thorough survey on surface parameterization techniques is beyond the scope of this work, and we refer readers to nice survey reports of [13,42,18] for details.

Volumetric mapping: Solid geometry and volumetric data have richer contents than that of surfaces, and volumetric parameterization applies well to the aforementioned problems when volumetric data are of most significance. Furthermore, in scientific computation, physics-based simulation, fluid dynamics, and material modeling, we are primarily handling volumetric data, so volumetric parameterization is an enabling technology with great application potentials. The previously mentioned problems will certainly benefit from an effective volumetric shape representation/modeling paradigm. Because of its importance, volumetric parameterization has been gaining greater interest in recent years, a few related research work has been conducted towards various applications such as shape registration [47,29], volume deformation [21,20], and trivariate spline construction [33]. Wang et al. [47] parameterized solid shapes over solid spheres by a variational algorithm that iteratively reduces the discrete harmonic energy defined over tetrahedral meshes, the harmonic energy is rigorously deduced but the optimization is prone to getting stuck on local minima and it only focuses on topological solid sphere shapes such as human brain datasets. Ju et al. generalized the mean value coordinates [12] from surfaces to volumes for a smooth volumetric interpolation, Joshi et al. [20] presented harmonic coordinates for volumetric interpolation and deformation purposes, their method guaranteed the non-negative weights and therefore led to a more pleasing interpolation result in concave regions compared with that in [21]. Li et al. [29,30] used Green’s functions and fundamental solution methods to map solid shape onto general target domains. Martin et al. [33] parameterized genus-zero tetrahedral meshes onto a cylinder, and used it for trivariate spline construction. Lipman et al. [32] and Ben-Chen et al. [5] also used Green’s functions to generate harmonic functions to guide space deformation.

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 [3], and was introduced into computer graphics for the simulation of deformable objects in [19]. 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 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 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 [4,15]. A comprehensive review of the MFS and kernel functions for solving many elliptic PDE problems was documented in [11].

3. Theory and algorithm

A volumetric map \( \tilde{f} \) between two 3-manifolds embedding in \( \mathbb{R}^3 \) is a bijective mapping \( \tilde{f} : M_1 \rightarrow M_2, M_1, M_2 \subset \mathbb{R}^3 \). The
boundary constraint is a surface mapping $f'$ from the boundary surface of the first solid object $M_1$, denoted as $\partial M_1$, to the boundary surface of $M_2$, denoted as $\partial M_2$. The mapping $f(p) = \{q \in M_1, q \in M_2\}$ is composed of three real functions in three axes directions, i.e., $f = (f^1, f^2, f^3)$. Each real function $f^i$ ($i=1,2,3$) maps the point $p$ to $\{q_1, q_2, q_3\}$’s corresponding component $q_i$. This problem is then reduced to the computation of real functions $f^i$ ($i=1,2,3$), with the given boundary surface mapping constraints $f^i = (f^{i1}, f^{i2}, f^{i3})$. We want the volumetric mapping to follow the boundary constraints and minimize a specific metric distortion. In this work, our object is to minimize the harmonic energy under the Dirichlet boundary condition discussed above, defined by the boundary surface mapping.

### 3.1. Harmonic volumetric mapping

Harmonicity of a mapping characterizes the smoothness of the transformation, which is a natural phenomenon that depicts the minimized physical energy that arises from the difference between two shapes. In the surface case, a harmonic map (with boundary loop mapping predetermined) finds the functions with the vanishing Laplacian everywhere, and it minimizes the Dirichlet energy and leads to a minimal surface [38,10]. Intuitively speaking, finding a harmonic map between two surfaces with fixed boundary correspondence is like computing the physical deforming, finding a harmonic map between two surfaces with fixed boundary constraint is a surface mapping $f_0$ from the boundary surface of the first solid object $M_1$, ... denotes the distance between the point $p$ and this particle $Q_s$.

On the other hand, the linear nature of Laplacian equations indicates that the boundary-based methods are most suitable since the interior is now determined in an exact manner. In other words, according to the maximum principle of harmonic functions, the value of a harmonic function never reaches maximal or minimal values in the interior region of the domain, and values in these interior regions are fully determined by the boundary condition. The method of fundamental solution (MFS), based on Green’s theory is a natural boundary mesh-free method to solve this problem. MFS can be viewed as a modified Trefftz method, and the basic idea is to approximate the solution by a linear combination of fundamental solutions with sources located outside the problem domain. Refs. [29,30] applied MFS in the computation of harmonic volumetric mapping, where three linear systems with one single coefficient matrix are solved to get the harmonic volumetric mapping between two 3D objects.

Compared to mesh-based variational methods such as [47], (1) the MFS method is a meshless boundary method, which is more efficient than this conventional mesh based FEM method, and with both time complexity and storage complexity greatly reduced; (2) MFS is more general and can flexibly handle volumetric datasets with complicated topologies, including topological noise; (3) the new MFS framework can also handle heterogeneous materials instead of just homogeneous shapes.

### 3.2. Method of fundamental solutions

We briefly review the idea of MFS in solving harmonic volumetric maps and define the notations that are used in our algorithms.

**MFS in harmonic volumetric mapping:** We seek three harmonic functions $(f^1, f^2, f^3) : M_1 \rightarrow M_2$, with $\Delta f^i = 0$. Since $\Delta$ is a linear self-adjoint differential operator and $M_1$ is a bounded domain in $\mathbb{R}^3$, we can compute its Green function. A fundamental solution of this differential equation is a function $K(x, x')$ such that

$$\Delta K(x, x') = \delta(x, x'), \quad x, x' \in \mathbb{R}^3,$$

where $\delta(x, x')$ is the Dirac delta function, the kernel $K$ is defined everywhere except the singularity point at $x = x'$.

Then we have

$$f^i(x) = \int K(x, x') g^i(x') \, dx'.$$

Such a kernel function $K$ is known as Green's function associated with the 3D Laplacian operator $\Delta$, and has the formula: $K(x, x') = \frac{1}{(4\pi)|x-x'|}$, where $|x-x'|$ denotes the distance between the points $x$ and $x'$. Following this scheme, solving the aforementioned harmonic mapping $f$ is like designing electric fields. For each harmonic function $f^i$ (for each axis direction), we compute a particle system. The outcome electric potential field of any particle system is always a real harmonic function (guaranteed by the Kernel function), and we only need to find the particle system that fits the boundary condition, indicating the boundary surface mapping, coupled with three axis components. This process of solving the best particle system simulates the computation of $f^i$.

Suppose we have a particle system, and consider an electronic particle $Q_i$ (called a singularity point, or a source point) outside the domain $M_1$, the corresponding fundamental solution for 3D Laplacian equation (i.e. its potential) on a point $p$ can be formulated as

$$K(p, Q_i) = \frac{1}{4\pi|p-Q_i|},$$

where $|p-Q_i|$ denotes the distance between the point $p$ and this particle $Q_i$. 

0 ¼ ðp, QsÞ.
Therefore, considering the entire particle system \{Q_s\} with a set of source points, the MFS equation to evaluate \( f' \) on an interior or boundary point \( p \) is

\[
f'(\mathbf{w}', \mathbf{Q}; p) = \sum_{n=1}^{n_s} w'_{\mathbf{n}} \cdot K(p, \mathbf{Q}_n), \quad p \in M_1,
\]

where suppose we have \( n_s \) source points in the exterior of \( M_1 \), \( \mathbf{Q} \) is the \( 3n_s \)-dimensional vector concatenating positions of all \( n_s \) 3D source points, and \( \mathbf{w}' = (w'_1, w'_2, \ldots, w'_{n_s}) \) is the \( n_s \)-dimensional vector to be determined, which indicates charge amount distribution on these source points.

**Boundary fitting:** When every source point \( \mathbf{Q}_n \in \mathbb{R}^3 \), \( n = 1, \ldots, n_s \) is outside of \( M_0 \), any charge distribution guarantees the vanishing Laplacian \( \Delta f(p) = 0, \forall p \in M_1 \), only that \( f'_i \) might violate the boundary conditions. Source points \( \{\mathbf{Q}_n\} \) should lie outside of \( M_0 \), namely, locate on the boundary surface \( \partial M_0 \) of a region \( \Omega_1 \) that contains \( M_1 \) (i.e. \( M_1 \subset \Omega_1 \subset \mathbb{R}^3 \)). In [30], an offset surface \( \partial M_1 \) of \( M_1 \) is created (by first computing implicit distance field \( d(\partial M) \) with respect to \( \partial M_1 \) [25] and then generating the polygonization [6]) on the implicit surface \( d(\partial M_1) + \delta = 0 \), and a set of source points are uniformly sampled on this \( \partial M_1 \). Then it solves the charge amount on each particle such that the potential field approximates the boundary condition. The boundary condition is the surface mapping, whose fitting is conducted over a set of evaluation points (also called the collocation points or constraint points) \( \{p_i \in M_1\} \). Three particle systems with their charge distribution solved in this fitting process compose the volumetric map \( f = (f^1, f^2, f^3) \).

**Limitations of the aforementioned routine:** The above algorithm of [30] has following two key limitations:

- **Computation efficiency and fitting accuracy:** Three dense linear systems need to be solved. Suppose we have \( n_c \) collocation (evaluation) points and \( n_s \) source points, we have a \( n_c \times n_s \) system where the dimension of the coefficient matrix \( A \) is \( n_c \times n_s \). \( A \) is dense since every source point contributes to every constraint point. Furthermore, \( A \) is ill-conditioned. As suggested in [40,29,30], singular value decomposition (SVD), due to its stableness against the ill-conditioned system, is chosen to solve this system. However, SVD decomposition is slow for large matrices. For example, the solver of [30] needs more than one day when both \( n_c \) and \( n_s \) exceed 20k vertices. Therefore, when handling complex volumetric data, we have to restrict \( n_s \) and \( n_c \). This causes the salient decrease on the boundary fitting accuracy. Because now we either (1) lack enough particles for designing fine potential fields to well fit the boundary condition (when \( n_s \) is picked to be small), or (2) lack enough evaluation points to sample the shape variance on the boundary (when \( n_c \) is picked to be small).

- **Feature alignment:** Like other volumetric mapping methods [47,21,30] focuses on homogenous volumetric regions where boundary surface map is the only constraint in the mapping computation. In real scenarios, volumetric data usually contain different materials and densities, or have salient structure inside its interior region (see Fig. 1(a) for example). These information or structures are usually meaningful and should be considered. Therefore, a scheme that can properly handle heterogeneous structure is worthwhile, so that we will be able to align or match similar material/interest when necessary.

**New computation scheme:** To tackle these two limitations, we improve [30] as follows.

- Instead of one function \( f'_i : M_1 \to M_2 \), we compute a set of harmonic functions \( f'_i \) such that their sum approximates \( f \). The computation for each \( f'_i \) is more efficient and numerically more stable. Each \( f'_i \) is harmonic, and therefore their sum \( \sum_i f'_i \) is also harmonic. Each subsequent \( f'_i \) aims to refine the existing map \( \sum_{k=0}^{i-1} f'_k \) towards the exact boundary condition \( f \) (which could need a too big system [29] to solve within only one shot). Now we use less constraint points and source points to compute each \( f'_i \), and therefore the solving is much faster, the boundary condition for each \( f'_i \) is \( \delta f'_i = f - \sum_{k=0}^{i-1} f'_k \). This greatly improves the speed of the mapping computation, and makes the MFS practical for large volumetric datasets.
- As we will demonstrate in our experiments, with only a few \( f'_i \), the fitting accuracy can usually beat the algorithm of [29], and we can keep refining it using more \( f'_i \) when necessary. Also, unlike the [30] that conducts uniform sampling over the offset surface \( \partial M_1 \), we conduct the sampling adaptively following the geometry of the given shapes. Together with the multi-MFS scheme, our scheme intuitively allows a flexible placement of source points. It is well known that in MFS, the location of source points constitutes a key issue and it has large impact on numerical stability of the MFS computation. The locations of source points are either preassigned or determined along with the coefficients of the linear combination. Most papers place source points on a surface outside of \( M_1 \) uniformly and solve least square linear systems, but this not always guarantees that the computed solution converges to the exact solution as the number of source points increases. The MFS with moving source points has been considered by several authors (e.g. [34,11]). This leads to a slow nonlinear optimization, and still, it has been reported that the initial placement of the sources is usually very important in the convergence of these algorithms, as they converge to the first local minima encountered.

![Fig. 1.](image-url)

(a) The extracted and cleaned volumetric shape has three salient iso-surfaces: head, skull, and brain. A target domain (d) is generated to test the efficacy of our mapping with iso-surface constraints. (b) and (c) show the 30% and 60% morphing from (a) to (d), generated by linear interpolation.
In this paper, we also follow the preassign-approach that leads to linear systems. Unlike the placement scheme in [30], we allow the removal and adaptive adding-in of new source points according to the result in the previous round, and this mimics the adjustment of the source points during the MFS solving. We will show that our new scheme also improves the accuracy of the mapping computation using MFS.

- Feature alignment: We allow the setting of constraint points during the mapping. In real applications, three types of constraints are very useful: 0-manifolds (feature points), 1-manifolds (feature curves or skeletons), and 2-manifolds (iso-surfaces). These constraints are treated as a part of boundary fitting. We apply an adaptive scheme to balance the feature constraint and boundary constraint.

3.3. Algorithm pipeline

Our algorithm pipeline is as follows. The input is two given solid objects $M_1$ and $M_2$ and their boundary surface mapping $\tilde{f} = (f^1, f^2, f^3) : \partial M_1 \to \partial M_2$.

The output is a harmonic volumetric mapping composed of a set of harmonic functions:

$$\tilde{f} : M_1 \to M_2 = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \left( f_i^j, f_i^j, f_i^j \right),$$

such that on the boundary surface $p \in \partial M_1$, $\tilde{f}(p) = f(p)$ and in the interior region: $\nabla^2 \tilde{f} = 0$.

Each harmonic real function $f_i^j, i = 1, \ldots, n_1, j = 1, 2, 3$ is solved by one linear system $A[w_i] = b_i$. In the following algorithm, we omit the indices $i$ and $j$ (e.g. using the notation $A$ instead of $A[i]$) for simplicity, assuming this will not cause any ambiguity:

1. Place source points and collocation points (Section 4.1).
2. Compute the coefficient matrix $A$, whose $(u, v, w)$-th element $A_{uvw} = A(P_u, Q_v, Q_w)$ (Eq. (1)) for the collocation point $P_u$ and source point $Q_v, Q_w$.
3. Decompose $A$ using singular value decomposition $A = U \Sigma V^*$. The decomposed results $U, \Sigma, V^*$ are used to solve the fitting system.
4. Set the boundary condition $b$ at the right hand side of $A[w] = b$, and $b = (b_1)$, where $b_1$ is the boundary constraint evaluated on each collocation point.

For each $f_i^j$, this algorithm solves $A[w_i] = b_i$. When $i = 1$, the boundary condition is set to be a low-resolution surface mapping from $M_1$ to $M_2$. For $i > 1$, we use a higher-resolution surface mapping, and also apply the refined boundary fitting $\tilde{f}^i = f^i - \sum_{j=1}^{n_1} f_j^i$.

Note that our algorithm takes the boundary surface mapping $\tilde{f}$ as an input. We briefly discuss how to obtain such a surface mapping in Section 4.3.

4. Implementation and discussion

4.1. Source points and collocation points placement

In order to set up the coefficient matrix for boundary fitting, first we need to place source points and collocation points. The $n_1$ source points $Q = (Q_1, Q_2, \ldots, Q_{n_1})$ are particles in the exterior of $M_1$ and $n_2$ collocation points $P = (P_1, P_2, \ldots, P_{n_2})$ are evaluation points on the boundary $\partial M_1$. We solve the weights (charge amount) distribution $w_i, i = 1, \ldots, n_2$ on all source points $Q_i$ so that $f(P_i)$ satisfies the boundary condition approximately.

The distribution of source and collocation points greatly affects the numerical stability and therefore the mapping efficiency and quality. The boundary error is sensitive to the collocation and source points, so appropriately sampling $Q$ and $P$ is critical. In 2D cases, theoretic studies have been conducted for analytical and simply connected domains, for example, when $M_1 \subset \mathbb{R}^2$ is a planar disk [34], uniformly sampling both collocation and source points is ideal and leads to exponential decreasing on boundary fitting errors. When $M_1 \subset \mathbb{R}^3$ is analytic, there is also discussion on the existence of optimal placement [23], and one suggestion is to take a conformal mapping $\Psi$ from the unit disk $D$ to $M_1$, and place $Q$ and $P$ on $\Psi$’s images of the evenly sampled points on the parametric circle. However, for more complicated domain shapes in 2D, or in our 3D case that $M_1 \subset \mathbb{R}^3$, the optimal placement is still unknown. Ref. [30] shows that placing source points on a nearby offset surface produces more accurate mapping result. We also adopt the offset surfaces scheme but add in adaptivity, following both the geometry and sequential fitting errors.

4.1.1. Sampling collocation points

In 2D analytical boundary scenarios [11], uniformly sampled $\tilde{P}$ usually leads to a stable system and good approximation, therefore is suggested as the strategy for preassigning collocation points. Ref. [30] follows this strategy, and uses the uniform sampling scheme of [36] to generate evenly distributed collocation points on $\partial M_1$. The total number of collocation points (source points) is controlled by an aspect ratio $\kappaRatio = n_1/n_2$ ($\kappaRatio = n_1/n_2$), where $n_2$ is the total vertex number of $\partial M_1$.

However, our multi-level MFS solving shows that for most 3D piecewise-linear domains, adaptively sampling these collocation points following local geometry could lead to better convergence on boundary fitting errors. Intuitively, more evaluation points shall be placed on highly detailed regions for better sampling the boundary variance. Our geometry adaptive sampling algorithm is as follows:

1. Tesselate the boundary surface where we need to do the sampling; it is the domain boundary $\partial M_1$ for collocation points, and offset surface $\partial M_2$ for source points.
2. Refine $M_1$ (for collocation points sampling) or $M_2$ (for source points sampling) by subdivision and get a dense mesh $\partial M'$. The vertex number of $\partial M'$ is determined by our sampling budget.
3. Conduct surface simplification on $\partial M'$ using the quadric error metric [14], which efficiently produces a good-quality approximated simplified mesh $\partial M'$. The vertex number of $\partial M'$ is determined by our sampling budget.
4. Vertices of mesh $\partial M'$ are used as sampling points.

Table 1 illustrates our experiments conducted on the spherical mapping of a vase-lichen model with 40k vertices, and the mapping amount.

<table>
<thead>
<tr>
<th>Source and collocation points placement using geometry-adaptive (GA) sampling and uniform (UN) sampling.</th>
<th>Boundary fitting error</th>
<th>Collocation error</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S-Pts</strong></td>
<td><strong>C-Pts</strong></td>
<td><strong>UN</strong></td>
</tr>
<tr>
<td><strong>Source and collocation points placement</strong></td>
<td></td>
<td>0.0569371760</td>
</tr>
<tr>
<td><strong>UN</strong></td>
<td><strong>GA</strong></td>
<td>0.0546609675</td>
</tr>
<tr>
<td><strong>GA</strong></td>
<td><strong>UN</strong></td>
<td>0.0544202844</td>
</tr>
<tr>
<td><strong>GA</strong></td>
<td><strong>UN</strong></td>
<td>0.0484185840</td>
</tr>
</tbody>
</table>

The experiment is conducted on mapping vase-lichen model with 40k vertices to a solid sphere, $\kappaRatio = 0.05$, $\kappaRatio = 0.05$, and the offset surface is 0.15 times object size distant. The boundary fitting error indicates the boundary mapping quality. Using geometry-adaptive sampling leads to less boundary error.
is for \( c_{\text{Ratio}}=0.05 \), \( s_{\text{Ratio}}=0.05 \), and the offset distance is 0.15 times object size. It clearly shows the advantage of geometry-adaptive sampling over uniform sampling in placement of both collocation points and source points.

### 4.1.2. Sampling source points

We place source points following three aspects.

- **Geometry-adaptive sampling**: In the coarsest level \( f_{0} \) computation, we conduct geometry-adaptive sampling on source points to determine their locations.
- **Even partitioning**: In finer levels, we partition sampled source points \( Q_{s} \) into several subsets evenly. Each time we only use a subset of charge points for efficient boundary fitting.
- **Adaptive deletion/insertion**: Meanwhile, in each step, we remove redundant source points by analyzing the diagonal matrix from the SVD decomposition (see Section 4.2 for details), and adaptively add in extra source points near the regions with large fitting errors by projecting badly fitted boundary points onto the offset surface.

### 4.2. Solving MFS linear systems by SVD

The boundary fitting is reduced to solving linear systems \( A\tilde{w} = b \). \( A \) can be dense and ill-conditioned [40], so regular linear system solvers such as Gaussian elimination, LU, and QR decompositions usually fail to produce a stable solution. As suggested in [40,30], we use singular value decomposition (SVD) to decompose \( A \). There are three reasons.

1. It generates accurate and stable results when the coefficient matrix is highly ill-conditioned.
2. It flexibly gets to the least-square solution for over constrained boundary conditions (which always happen in our multi-round MFS solving).
3. Furthermore, in our approach, we also use the diagonal matrix \( \Sigma \) to adaptively remove redundant singularity points. When the singular value is small (in all our experiment, we set the threshold to \( 10^{-5} \)), the corresponding source point does not contribute much to the potential field, and therefore we remove them in the source point set from subsequent linear system solving and MFS evaluations.

### 4.3. Surface mapping as boundary condition

The boundary condition of our harmonic volumetric mapping is a surface mapping between \( \partial M_{1} \) and \( \partial M_{2} \). Existing inter-surface mapping techniques [22,26,35,39,24,41,48] can be used for creating the boundary surface mapping. Although surface mapping is not the focus of this paper, as discussed in [30], it is desirable to have a low distorted surface mapping, and [30] illustrates an example that larger angular distortion oftentimes leads to worse volumetric mapping result. However, it is still unknown that how exactly quality of boundary mapping and interior mapping relate. Intuitively, area preserving is also important (salient shrinkage on boundary mapping shall lead to large volume distortion). Ref. [30] uses [27] for boundary surface mapping computation which leads to least angular distortion, in our work, we use [28] to generate boundary surface mapping, which could (1) better balance area stretch and angle distortion, and (2) allow surface feature points and curves alignment which better fits our current framework.

4.4. Feature alignment

Feature and structure constraints are important issues in processing many real volumetric data. Specifically, three types of features are commonly considered: 0-manifolds (feature points), 1-manifolds (feature lines), and 2-manifolds (iso-surfaces).

**Feature point alignment**: Since we are using a meshless paradigm, all the feature alignment shall naturally be handled in terms of points. We can simply add the feature constraint defined on each point into the linear system as a new boundary condition. So each new pair of feature points for alignment corresponds to an additional row in the coefficient matrix \( A \) and the boundary condition vector \( b \).

**Feature line alignment**: For feature lines (for instance, skeletons) matching, we similarly sample and put in point-by-point constraints to \( A \). Feature curves such as skeletons are usually represented as a piecewise graph. Fig. 3(a) shows an example, in which one wants the volumetric mapping follows skeleton structure and is guided by the movement or deformation of the skeleton. Existing skeleton extraction algorithms [9,37,2,17] usually do not guarantee that the extracted skeletons from different objects are isomorphic even when these shapes are very similar. However, most skeletonization algorithms can preserve the homotopic structure of the shape and therefore their skeletal graphs are topologically equivalent. Detail discussion on feature lines extraction, their topology, and their matching is far beyond the scope of this work, and we refer readers to the survey paper [7]. In our experiment (Fig. 3), we use [9] to extract the skeleton and apply 1-manifold restamping algorithm of [1] to get dense point-by-point correspondence between skeletons of the cyberware Male and Female models. Fig. 3(b) visualizes the volumetric map result using the color-encoded distance field. The distance of each interior point to the boundary surface is computed and color-encoded, such a color is transferred to its corresponding point under the mapping. This visualizes the mapping behavior. (c) illustrates the skeleton fitting; in the interior of the target Female model, the green curve is its skeleton \( SK_{2} \). Sample points from the skeleton of the source Male model should match \( SK_{2} \), and the red points shows their images under the mapping. The rooted mean square fitting error is 0.5%.

**Feature surface alignment**: With two feature surfaces to match, we need to compute inter-surface maps between them. The inter-surface parameterization methods, which we used to generate boundary surface correspondence, can be applied to get such a map. Then we simply include corresponding point pairs as the boundary conditions. Figs. 1 and 2 show an example of a volumetric mapping over the heterogeneous data head–skull–brain model, which has three salient iso-surfaces: the outer boundary is a genus-0 (head) surface, and the interior skull iso-surface is genus-2, within which there is a genus-0 brain surface. We generate a parametric domain \( d \) to test the efficacy of our mapping on heterogeneous 3D data with iso-surface constraints. The outer head boundary surface is mapped onto a sphere boundary, the skull iso-surface is constrained on the polycube skull, while the brain iso-surface is mapped to a small cube inside. (b,c) show the 30% and 60% morphing from (a) to (d), generated by linear interpolation. (e,g) show two cross-sections on the polycube-sphere domain, and (f,h) show their corresponding cross-sections on the head–skull–brain model. The point clouds in (e)–(f) show sample points on the iso-surface (e, g), and their images after the volumetric mapping (f–i). Locations of these feature points in (g–i) demonstrate that the iso-surface constraints are precisely fitted, and the volumetric mapping align the feature surface very well (Fig. 2(j)).

**Feature alignment as soft constraints and weights**: As discussed above, features are aligned in a least square sense together with
the boundary fitting process, so they are treated as soft constraints. Compared with the massive point number on the boundary, if feature points (or samplings on feature lines) are considered as ordinary collocation points, they might be overshadowed by boundary collocation points during the fitting. We balance this by assigning each sample feature point an extra weight \( w \). This is equivalent to enforcing this feature \( w \) times. In all our experiments, we take \( w = 20 \) for feature points. This effectively leads to more precise feature alignment.

Unlike the traditional FEM-based methods that simply fixes feature vertices to enforce the constraints, in this section we discuss our method that blends several harmonic functions to get the feature-aligned map. In each iterative refinement, we use a harmonic function, so the resultant map, i.e. the summation of these functions is still globally harmonic, and there is no obvious flip-over or discontinuity around the feature regions. In the mean time, while the feature constraints is precisely enforced, the boundary fitting accuracy could decrease a little bit (i.e. the RMSE increases slightly on the boundary).

5. Experimental results and applications

We conduct a few volumetric mapping experiments over various volumetric data, with different sizes, topology and geometry complexities. We illustrate some of these mapping results in Figs. 4 and 5. We use the color-encoded distance field to visualize the mapping result. When a map \( f : M_1 \rightarrow M_2 \) is computed, the color-encoded (red indicates the maximum while blue indicates the minimum, see Fig. 5(h)) distance field defined on one region can be transferred to another region, by plotting the color of a point \( P \in M_1 \) on its corresponding image \( f(P) \in M_2 \) (or inversely, plotting the color of \( P \in M_2 \) on \( f(P) \in M_1 \)). This visualization shows the effect of the map. For example, when we transfer the distance field defined on the Max-Planck model (Fig. 4(c)) to the sphere, we can see a color-encoded head-shaped level-set in (d), while the original distance field of a sphere is concentric as shown in (f).

We also conduct thorough comparison between our method and the algorithm of [30]. Table 2 illustrates the side-by-side statistics. Using MFS [30], the \( c \)Ratio (and \( s \)Ratio), indicating the ratio of the number of collocation points (and source points) over the number of boundary points are listed in the CR (and SR)
corresponding CR (in second), and the ping:  

\[ f: M_1 \rightarrow M_2 \]

\( n_f \) (number of harmonic maps we solved), computation time, and RMSE.

<table>
<thead>
<tr>
<th>Models (vertex #)</th>
<th>CR</th>
<th>SR</th>
<th>Time (s)</th>
<th>RMSE</th>
<th>CR*</th>
<th>SR*</th>
<th>n_f</th>
<th>Time* (s)</th>
<th>RMSE*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Omotondo/Sphere (3k)</td>
<td>0.4</td>
<td>0.8</td>
<td>220.41</td>
<td>0.01649</td>
<td>0.4</td>
<td>0.8</td>
<td>6</td>
<td>42.41</td>
<td>0.001514</td>
</tr>
<tr>
<td>PCube/2-Torus (6.6k)</td>
<td>0.4</td>
<td>0.8</td>
<td>2393.10</td>
<td>0.00490</td>
<td>0.4</td>
<td>0.8</td>
<td>6</td>
<td>426.23</td>
<td>0.00475</td>
</tr>
<tr>
<td>Male/Female (6.3k)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.2</td>
<td>2</td>
<td>10</td>
<td>1000.68</td>
<td>0.00485</td>
</tr>
<tr>
<td>PCube/Skull (29k)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.03</td>
<td>0.03</td>
<td>7</td>
<td>14.42</td>
<td>0.01764</td>
</tr>
<tr>
<td>Vaselion/Sphere (40k)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.1</td>
<td>0.2</td>
<td>6</td>
<td>1440.39</td>
<td>0.0230</td>
</tr>
<tr>
<td>PCube/Kitten (80k)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.05</td>
<td>0.05</td>
<td>6</td>
<td>666.48</td>
<td>0.0139</td>
</tr>
<tr>
<td>PCube/Horse (100k)</td>
<td>0.01</td>
<td>0.02</td>
<td>6</td>
<td>23.82</td>
<td>0.01449</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A lot of 3D solid models have been tested, from data with small vertex size to large size. Using MFS [30]: CR and SR (\( \epsilon \) and \( \delta \) of the mapping), computation time (in second), and the RMSE (rooted mean square error of the boundary fitting) are listed; using the new computation framework, in the same row we list the statistics of the corresponding CR*, SR*, \( n_f \) (number of harmonic maps we solved), computation time, and RMSE*.

5.1. Hex-remeshing

A direct application for volumetric mapping is hex-mesh generation. Regular mesh structure is highly desirable for finite element analysis and physically based deformations/simulations, because regular meshes provide great efficiency for geometry processing and physically based computation [44]. Given a 3D solid data \( M \), we first generate a solid polycube model \( P \), then we compute the surface mapping \( f: \partial M \rightarrow \partial P \) and volumetric mapping \( f: M \rightarrow P \). With \( f \) we can transfer the regular structure on \( P \) to \( M \). On a solid polycube, a regular hexahedral structure can be easily generated, and since [43] introduced the concept of surface polycube map, several techniques [45,46,31,16] have been proposed to (automatically) construct polycube, and the surface-polycube mapping. In all of our experiments, we construct our polycubes using the algorithm of [45]. Fig. 7 illustrates an example of using a unit solid cube to remesh the solid David head. The original model \( M_2 \) is shown in (a), and the hex mesh of the parametric cube \( M_1 \) is shown in (b). We compute the volumetric map \( f: M_1 \rightarrow M_2 \) from the cube to David head. Then \( f(M_1) \) is a solid with the hex connectivity of \( M_1 \) and the head shape of \( M_2 \), and it is the remeshed David head, as illustrate in (c) and (d). Fig. 8 shows a few more examples. A hex-remeshed two-hole torus is shown in (a). The hex-mesh structure of the polycube (b) is used to remesh the kitten, shown in (c,d). Polycube (e) is used to remesh the Chinese horse model (f–h), (f, g) visualize the result hex-mesh in its interior regions from two different cross-sections.

6. Conclusion

We present a feature-aligned volumetric harmonic mapping computation algorithm using methods of fundamental solutions. The map \( f \) is composed of a set of harmonic functions \( \{ f_i \} \) which can be efficiently solved. Also, our adaptive source/collocation points placement improves the numerical issue of MFS solving. Therefore, our algorithm largely improves the existing harmonic
volumetric mapping computation algorithm using MFS [30]. The new algorithm has better efficiency and accuracy, and it supports feature points, curves, or surfaces alignment, which is important for integrating/matching heterogeneous volumetric data that have intrinsic interior structure. We demonstrate that harmonic volumetric mapping can be conducted on large data, heterogeneous data, and data with feature to match, which can not be handled properly in [30].

6.1. Future work

We plan to explore the application of our volumetric mapping framework in realistic scenarios such as heterogeneous volumetric medical data registration. And we would like to further explore and seek to improve the MFS method in solving harmonic volumetric mapping. Several directions might be interesting.

- **Bijective volumetric mapping via domain decomposition**: The bijectiveness of harmonic volumetric parameterization has been proved to exist in several special types of shape domains. We plan to explore effective mapping computation frameworks through domain decomposition methods.

- **Free boundary volumetric map**: Our current harmonic volumetric map depends on the boundary surface mapping. As discussed in Section 4.3, the effect of volumetric map is closely related to the boundary surface mapping. In the future work, we plan to explore free-boundary volumetric mapping, either allowing boundary vertices flowing over the boundary surface in a variational manner or including them as unknown variables in the MFS solving. A free-boundary volumetric mapping could have significantly less distortion.

- **Source and collocation points placement**: We have demonstrated our adaptive placement of source and collocation points leads...
to better convergence of boundary fitting errors in compared with generally suggested uniform sampling scheme (and also the algorithm in [30]). It is worthwhile to further explore this problem since it directly dictates the numerical effectiveness of the mapping computation. Ref. [23] discussed 2D Dirichlet problems over simply connect domains, and suggested a scheme based on planar conformal mapping: when we have a conformal map $\psi$ from $M_1$ to a regular exterior shape like a circle $M_2$, an ideal placement of collocation points is on the images of the inverse conformal map $\psi^{-1}$, where $\{p\}$ are evenly sampled points on $M_1$. We plan to explore this scheme in the future in the $\mathbb{R}^3$ case via surface conformal mapping.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments and suggestions. Discussions with Hong Qin and Warren Waggenspack inspired this work. The head of David model is from Stanford Michelangelo project, male and female models are from Cyberware. Other models are from AIM@SHAPE Shape Repository. This work is supported by Louisiana Board of Regents Research Competitiveness Subprogram (RCS) LEQSF(2009-12)-RD-A-06 and Pfund: NSF(2009)-PFUND-133. Huanhuan Xu is supported in part by the Mark and Carolyn Guidry doctoral fellowship from Electrical and Computer Engineering Department of LSU.

References