This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2016.2605092, IEEE Transactions on Visualization and Computer Graphics

Knot Optimization for Biharmonic B-splines on Manifold Triangle Meshes

Fei Hou, Ying He, Hong Qin, and Aimin Hao

Abstract—Biharmonic B-splines, proposed by Feng and Warren, are an elegant generalization of univariate B-splines to planar and curved domains with fully irregular knot configuration. Despite the theoretic breakthrough, certain technical difficulties are imperative, including the necessity of Voronoi tessellation, the lack of analytical formulation of bases on general manifolds, expensive basis re-computation during knot refinement/removal, being applicable for simple domains only (e.g., such as Euclidean planes, spherical and cylindrical domains, and tori). To ameliorate, this paper articulates a new biharmonic B-spline computing paradigm with a simple formulation. We prove that biharmonic B-splines have an equivalent representation, which is solely based on a linear combination of Green's functions of the bi-Laplacian operator. Consequently, without explicitly computing their bases, biharmonic B-splines can bypass the Voronoi partitioning and the discretization of bi-Laplacian, enable the computational utilities on any compact 2-manifold. The new representation also facilitates optimization-driven knot selection for constructing biharmonic B-splines on manifold triangle meshes. We develop algorithms for spline evaluation, data interpolation and hierarchical data decomposition. Our results demonstrate that biharmonic B-splines, as a new type of spline functions with theoretic and application appeal, afford progressive update of fully irregular knots, free of singularity, without the need of explicit parameterization, making it ideal for a host of graphics tasks on manifolds.

Index Terms—Biharmonic B-splines, Green's functions, manifold triangle meshes, implicit representation, knot optimization

1 Introduction

Various types of spline functions, such as tensor-product B-splines, NURBS, and T-splines, have been broadly deployed in geometric design, data fitting, shape analysis, and many other graphics applications [1], because of many powerful and attractive properties. For example, their basis functions are localized, piecewise polynomials, and form a partition of unity; the resulting spline has certain order of smoothness and can model complicated geometry in a free-form manner. In principle, the above-mentioned splines are built on *planar* domains with *restricted* knot configuration, such as regular grids and T-junctions, so it is highly desirable to seek brand new spline schemes upon arbitrary *curved* domains with *fully irregular* knot sets.

Biharmonic B-splines, proposed by Feng and Warren [2], are one such novel advancement, that extends univariate B-splines to planar and curved domains. In [2], the key observation is that the discrete bi-Laplacian is a well-behaved analog of divided differences. Given a compact smooth manifold $\mathcal{M}$ and a set of knots $\{t_i\}_{i=1}^m$, $t_i \in \mathcal{M}$, the biharmonic B-spline basis function ψ_i, associated with knot t_i, is defined using Green's functions of the bi-Laplacian operator Δ^2 on the Voronoi cells around t_i. Feng and Warren showed that these basis functions are localized and form a partition of unity on domains without boundary. In sharp contrast to conventional B-splines, the knots of biharmonic B-splines are totally free without any additional constraint. They do not rely on any user-specified planar/volumetric parameterization and are free of singularity [2], [3]. Nevertheless, the flexibility of biharmonic B-splines comes with a high price: (1) The need of Voronoi tessellation is unavoidable, and computing such a Voronoi diagram on a curved domain is expensive; (2) The computation of the distance function is lacking an analytical formulation on general surfaces, indicating that the bases do not have a closed-form expression; (3) The bases must be re-evaluated wherever domain re-configuration is carried out during knot refinement and coarsening; and (4) Bi-Laplacian operators are well defined on simple domains (e.g., such as Euclidean planes, spherical and cylindrical domains, and tori), yet discretizing the bi-Laplacian operator on manifold is non-trivial.

To combat the aforementioned difficulties, the overarching goal of this paper is to expand the horizon of biharmonic B-splines at both theoretic and practical fronts. We promote the use of biharmonic B-splines through a novel, yet simpler, and equivalent formulation. Rather than the conventional means of computing their basis functions directly, and then formulating biharmonic B-spline as a linear combination of control points and accompanying basis functions, we advocate a fundamentally different approach in this paper. Through theoretic exploration, we are capable of *implicitly* formulating a biharmonic B-spline as a linear combination of Green's functions of bi-Laplacian operator *without explicitly expressing B-splines' bases*. Compared with the conventional *explicit* representation based on control points and basis functions, our implicit representation can completely bypass the Voronoi tessellation and the discretization of bi-Laplacian operator, and enable the construction of bihar-

- *F. Hou is with School of Computer Science and Engineering, Nanyang Technological University, Singapore and State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, China.*
- *Y. He (corresponding author) is with School of Computer Science and Engineering, Nanyang Technological University, Singapore.*
- *H. Qin is with Department of Computer Science, Stony Brook University, USA.*
- *A. Hao is with State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, China.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2016.2605092, IEEE Transactions on Visualization and Computer Graphics

monic B-spline on any compact smooth 2-manifold.

The proposed implicit representation naturally and elegantly brings forth two fundamental properties, which otherwise cannot be easily derived in the explicit representation. First, each Green's function is associated with a single knot, facilitating an optimization-driven knot selection for spline construction. Second, we prove that the linear combination of two biharmonic B-splines is still a biharmonic B-spline, whose knot set is simply the union of the knot sets of input splines. Such property is highly desirable to manifold data analysis and processing, where the user can easily combine multiple spline functions into a single representation without adding extra knots. It is worth noting that both knot optimization and the linear combination feature are not available to the conventional B-splines. For example, a degree-k B-spline basis function involves $k+1$ knots, making the knot optimization difficult. Also, adding two tensor-product B-splines with different knot vectors requires many additional knots to enforce the correct topology of knot pattern.

We develop a new computational framework for constructing biharmonic B-splines on triangular meshes. Our framework consists of algorithms for spline evaluation, optimal knot selection, data interpolation and hierarchical data decomposition. We compare biharmonic B-splines with least-square meshes [4][5], an elegant compact surface representation scheme that is also based on discrete basis functions. Least-square mesh can recover the geometry from the connectivity of the input mesh and a sparse set of control vertices with geometry, hence it is ideal for surface compression and progressive transmission. Thanks to its free-knot nature, biharmonic B-spline affords progressive update of fully irregular knots, is free of singularity, and can be constructed without the need of explicit parameterization, making it ideal for a host of signal processing tasks on manifolds.

The remaining of the paper is organized as follows: Section 2 reviews the related work on spline functions, discrete Laplacian and manifold data modeling techniques. Then Section 3 introduces the mathematical background of Feng and Warren's biharmonic B-splines [2]. Section 4 presents our implicit representation and the linear combination property, Section 5 details the computational framework of constructing biharmonic B-splines on triangle meshes. Section 6 introduces the optimization-driven knot selection for spline construction, followed by applications on data interpolation and hierarchical data decomposition in Section 7. Section 8 compares our method with existing work. Finally, Section 9 discusses the limitations and points out future work.

2 Related Work

There is a large body of literature on spline functions and their applications to data fitting and interpolation. Due to page limitation, this section reviews only the most related work. We refer readers to [6] for a comprehensive review.

Biharmonic B-splines generalize the univariate B-spline to curved domains by making use of the connection between divided difference and discrete Laplacian. Feng and Warren [2] constructed biharmonic B-spline bases from the divergence theorem and employed finite difference to approximate the boundary integral. Hou et al. [3] improved the numerical accuracy and stability of biharmonic B-splines by quadratic programming.

Laplacian and bi-Laplacian operators are widely-used differential operators in graphics with various applications, such as heat kernel signature [7], surface parameterization [8], volume parameterization [9], spectral graph [10], and biharmonic distance [11], etc. In geometry processing, the discrete Laplacian operator has been well studied in literature. Meyer et al. [12] proposed the cotangent-weighted discrete Laplacian operator for triangulated 2-manifold. Hildebrandt et al. [13] constructed strongly consistent discrete Laplacian-Beltrami operators on polyhedral surface. Wardetzky et al. [14] reviewed various properties of the discrete Laplacian operators. Many methods use iterated Laplacian for discrete bi-Laplacian [15][11]. Feng and Warren [2] continued to improve them to achieve cubic precision by making use of additional constraints. Rustamov [16] proposed multiscale biharmonic kernels, producing localized basis functions on different scales. In the limit of large scales, the multiscale biharmonic kernel converges to Green's function of the biharmonic equation. The kernels are also shape-aware, robust to noise, tessellation and partial object. However, the basis functions do not form partition of unity.

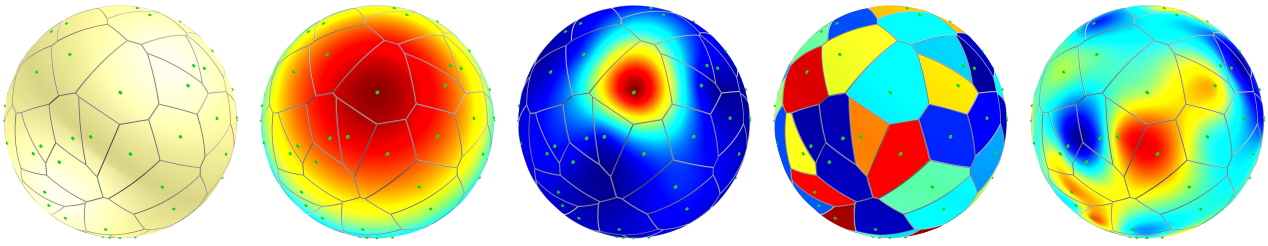
Green's functions are the fundamental solution of a linear differential operator. Observing that Green's function of the Laplacian operator has a unique critical point on star-shaped volume, Xia et al. [17] presented a volumetric parameterization for star shapes. Based on Green's functions of bi-Laplacian operator, Lipman et al. [11] presented biharmonic distance, which provides a nice trade-off between nearly geodesic distances for small distances and global shape-awareness for large distances. Recently, Wang et al. [18] developed a denoising algorithm which uses Green's functions of Laplacian operator as dictionary for feature and noise decoupling.

Knots optimization aims at finding the optimal number of knots and their locations in order to produce a spline satisfying certain accuracy constraints. It is a challenging problem to B-splines, since the basis functions are highly non-linear functions of knots. Miyata and Shen [19] presented adaptive free-knot splines that allow selection of knot numbers and replacement of knots at any location. Kang et al. [20] proposed an ℓ_1 optimization algorithm to determine the optimal number of knots and their locations for univariate B-splines. Recently, Brandt et al. [21] formulated the optimal spline approximation as an ℓ_0-regularized quadratic problem and solved it using a modified random coordinate-descent method. Although these algorithms work well for generating optimal spline curves, it is non-trivial to extend them to the surface case, in which knots are frequently constrained with certain patterns, such as regular grid. To our knowledge, there is no existing work to address the knot optimization problem for constructing spline surfaces.

3 Preliminary

This section briefly reviews the mathematical background of biharmonic B-splines. Refer to [2] for the technical details.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2016.2605092, IEEE Transactions on Visualization and Computer Graphics



(a) Knots $\mathcal{T}$ (b) Green's function $\phi_{t_i}(x)$ (c) Basis function $\psi_i(x)$ (d) Control points $\{\lambda_j\}_{j=1}^m$ (e) Spline $\sum_{j=1}^m \lambda_j \psi_j(x)$

Fig. 1: The illustration of biharmonic B-spline on sphere. (a) The knots (in green) and the Voronoi diagram. (b) Green's function of bi-Laplacian centered at knot t_j. (c) The basis function ψ_j for Voronoi cell $\mathcal{V}_j$ is localized. All basis functions form a partition of unity $\sum_j \psi_j(x) \equiv 1$ for all $x \in \mathbb{S}^2$. (d) The color in each Voronoi cell indicates the control point $\lambda_j \in [0, 1]$. (e) The biharmonic spline $f : \mathbb{S}^2 \to \mathbb{R}$ is a smooth function defined on the sphere.

3.1 Basis Functions

Let $\mathcal{M}$ be a compact smooth 2-manifold and $\mathcal{T} = \{t_i\}_{i=1}^m$, $t_i \in \mathcal{M}$, be a set of knots on $\mathcal{M}$. The Voronoi diagram $\mathcal{V} = \bigcup_{i=1}^m \mathcal{V}_i$ forms a partition of $\mathcal{M}$ and $\mathcal{V}_i$ is the Voronoi cell of knot t_i.

For arbitrary points $x, y \in \mathcal{M}$, Green's function of the bi-Laplacian operator Δ^2 is

$$\Delta^2 \phi_y(x) = \delta_y(x) - \frac{1}{area(\mathcal{M})}. \tag{1}$$

where $\delta_y(x)$ is the Dirac delta function centered at y and $area(\mathcal{M})$ is the area of the manifold $\mathcal{M}$. For compact smooth manifold, there always exists a symmetric solution (i.e. $\phi_y(x) = \phi_x(y)$) to Equation (1) [22].

Denote by $\partial\mathcal{V}_j$ the boundary of Voronoi cell $\mathcal{V}_j$. By the divergence theorem, one has

$$\int_{\mathcal{V}_j} \Delta^2 \phi_y(x)\, \mathrm{d}\sigma = \int_{\partial\mathcal{V}_j} \frac{\partial \Delta\phi_y(x)}{\partial \overrightarrow{n}}\, \mathrm{d}s, \tag{2}$$

where $\overrightarrow{n}$ is the outward unit normal to the boundary $\partial\mathcal{V}_j$, $\mathrm{d}\sigma$ and $\mathrm{d}s$ are the area and line integral elements, respectively. Let $c_j = \frac{area(\mathcal{V}_j)}{area(\mathcal{M})}$ denote the area ratio. When $y \in \mathcal{V}_j$, the above integral is $1 - c_j$, otherwise $-c_j$.

Then the basis function for Voronoi cell $\mathcal{V}_j$ is defined as

$$\psi_j(x) = c_j + \sum_i n_{ij} \phi_{t_i}(x), \tag{3}$$

where $\sum_i n_{ij}\phi_{t_i}(x)$ is a boundary sum that approximates the line integral on the right hand side of Equation (2).

Feng and Warren [2] showed that the basis functions are localized. Moreover, they form a partition of unity $\sum_j \psi_j(x) \equiv 1$ for any $x \in \mathcal{M}$, since Green's functions on common Voronoi edge of adjacent Voronoi cells cancel (i.e., $\sum_{i,j} n_{ij}\phi_{t_i}(x) = 0$) and $\sum_{j=1}^m c_j = 1$. Therefore, $\{\psi_j(x)\}_{j=1}^m$ are the analog of the basis functions of the conventional B-spline.

Remark. Note that the biharmonic B-spline's basis function ψ_j depends on the neighboring Voronoi cells of $\mathcal{V}_j$. Since the knots $\{t_i\}_{i=1}^m$ can be distributed on $\mathcal{M}$ in an arbitrary manner, there is no fixed relation between the basis function and knots. In sharp contrast, the basis functions of univariate B-splines are associated with fixed number of knots. For example, the basis function of a cubic B-spline always has 5 knots. Indeed, the loose connection between the biharmonic bases and the knots endows the biharmonic B-spline with the flexibility of modeling based on fully irregular knots.

3.2 Biharmonic B-splines

The biharmonic B-spline function $f : \mathcal{M} \to \mathbb{R}^d$ is given by

$$f(x) = \sum_{j=1}^m \lambda_j \psi_j(x), \tag{4}$$

where $\lambda_j \in \mathbb{R}^d$ is the control point. Using matrix representation, the biharmonic B-spline and its basis functions can be written as

$$f = \mathbf{\Lambda}^T \Psi, \tag{5}$$

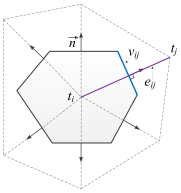
$$\Psi = \mathbf{c} + \mathbf{N}\mathbf{\Phi}, \tag{6}$$

where $\mathbf{c} = [c_1, \ldots, c_m]^T$, $\Psi(x) = [\psi_1(x), \ldots, \psi_m(x)]^T$, $\mathbf{\Phi}(x) = [\phi_1(x), \ldots, \phi_m(x)]^T$, $\mathbf{\Lambda} = [\lambda_1, \ldots, \lambda_m]^T$, and $\mathbf{N} = [n_{ij}]_{m\times m}$.

Similar to univariate B-splines, biharmonic B-splines also possess the knot insertion property. Feng and Warren [2] proved a refinement theorem for inserting new knots while keeping the spline function unchanged. Let $\mathbf{\Lambda}_0^T \Psi_0$ be a biharmonic B-spline defined on knot set $\mathcal{T}_0$. Given a refined knot set $\mathcal{T}_k \supset \mathcal{T}_0$, there exist the control points $\mathbf{\Lambda}_k^T$ such that $\mathbf{\Lambda}_0^T \Psi_0 = \mathbf{\Lambda}_k^T \Psi_k$.

3.3 Discretizing the Bi-Laplacian Operator

A key component in biharmonic B-splines is to discretize the bi-Laplacian operator Δ^2 on $\mathcal{M}$. We denote by v_{ij} the common Voronoi edge of $\mathcal{V}_i$ and $\mathcal{V}_j$ (i.e., the bisector of t_i and t_j), and e_{ij} the dual Delaunay edge. Let $\mathbf{A}$ be a diagonal matrix whose i-th entry is the area of $\mathcal{V}_i$ and $\mathbf{L}$ the discrete Laplacian matrix



$$\mathbf{L}_{ij} = \begin{cases} \sum_k l_{ik}, & \text{if } i = j \\ -l_{ij}, & \text{if } \mathcal{V}_i \bigcap \mathcal{V}_j \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

where $l_{ij} = \frac{|v_{ij}|}{|e_{ij}|}$. A common way to discretize the bi-Laplacian operator Δ^2 is $\mathbf{N} = \mathbf{L}\mathbf{A}^{-1}\mathbf{L}$. However, as pointed out by Feng and Warren [2], this discrete bi-Laplacian operator has only

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2016.2605092, IEEE Transactions on Visualization and Computer Graphics

linear precision on irregular grids, which may cause some numerical issues, such as non-localized basis functions. To fix this issue, Feng and Warren [2] proposed a simple method that approximates $\frac{\partial \Delta_\phi}{\partial \vec{n}}$ with cubic precision on the Voronoi edges. Hou et al. [3] proposed a two-stage linear least-square method to discretize Δ^2, where the first stage computes the optimal coefficient l_{ij} of the discrete Laplacian and the second stage optimizes the difference mask n_{ij}. In this paper, we discretize the basis functions using the technique in [3], which is numerically stable.

The discretized bi-Laplacian has the following properties: 1) The difference masks n_{ij} add up to zero, $\sum_{i=1}^{m} n_{ij} = 0$, since the sum of discrete Laplacian coefficients vanishes and the contributions of Green's functions on common edges of adjacent Voronoi cells cancel out. 2) The matrix $\mathbf{N}$ has a rank of $m-1$.

4 Implicit Representation

This section presents a simple and equivalent representation of biharmonic B-spline, which can significantly reduce the computational cost of spline construction and evaluation. Observing that the basis functions $\{\psi_j\}$ consist of Green's functions $\{\phi_{t_i}\}$ of bi-Laplacian, we investigate the relation between biharmonic B-splines and Green's functions and show that any biharmonic B-spline has an equivalent representation based on Green's functions.

Theorem 1 (Equivalent Representation). A biharmonic B-spline function $f(x) = \sum_j \lambda_j \psi_j(x)$ on a compact manifold $\mathcal{M}$ can be written as

$$f(x) = a + \sum_i w_i \phi_{t_i}(x), \tag{7}$$

where $\sum_i w_i = 0$ and $a = \sum_i \lambda_i c_i$. Conversely, each linear combination of Green's functions $a + \sum_i w_i \phi(t_i)$ with $\sum_i w_i = 0$ and an arbitrary a corresponds to a unique biharmonic B-spline.

Proof. The first statement is obvious, since

$$f(x) = \mathbf{\Lambda}^T \mathbf{\Psi} = \mathbf{\Lambda}^T \mathbf{c} + \mathbf{\Lambda}^T \mathbf{N} \mathbf{\Phi} = \mathbf{\Lambda}^T \mathbf{c} + (\mathbf{N}\mathbf{\Lambda})^T \mathbf{\Phi} = a + \mathbf{w}^T \boldsymbol{\phi},$$

where $a = \sum_{j=1}^{m} \lambda_j c_j$, $\mathbf{w} = [w_1, \ldots, w_m]^T$ and $w_i = \sum_{j=1}^{m} \lambda_j n_{ij}$. Fixing j, the sum $\sum_i n_{ij} = 0$. Therefore, we have $\sum_i w_i = \sum_j \lambda_j (\sum_i n_{ij}) = 0$, for any $1 \le i \le m$.

Next we prove the converse is also true. Define the augmented matrices $\widetilde{\mathbf{N}} \triangleq \begin{bmatrix} \mathbf{N} \\ \mathbf{c}^T \end{bmatrix}$ and $\widetilde{\mathbf{w}} = \begin{bmatrix} \mathbf{w} \\ a \end{bmatrix}$. Recall that $\sum_i n_{ij} = 0$ and $\mathrm{rank}(\mathbf{N}) = m-1$. Since the c_js add up to 1, the augmented matrix $\widetilde{\mathbf{N}}$ has a rank m. Also note that $\sum_i w_i = 0$, we have $\mathrm{rank}([\widetilde{\mathbf{N}}, \widetilde{\mathbf{w}}]) = \mathrm{rank}(\widetilde{\mathbf{N}}) = m$. Therefore, the system of linear equations

$$\widetilde{\mathbf{N}} \mathbf{\Lambda} = \widetilde{\mathbf{w}}, \tag{8}$$

has a unique solution.

As a result, the given linear combination of Green's functions can be written as

$$a + \sum_{i=1}^{m} w_i \phi_i(x) = \widetilde{\mathbf{w}}^T \begin{bmatrix} \mathbf{\Phi}(x) \\ 1 \end{bmatrix} = \left(\widetilde{\mathbf{N}}\mathbf{\Lambda}\right)^T \begin{bmatrix} \mathbf{\Phi}(x) \\ 1 \end{bmatrix}$$

$$= \mathbf{\Lambda}^T \left(\mathbf{N}^T \mathbf{\Phi}(x) + \mathbf{c}\right) = \mathbf{\Lambda}^T \mathbf{\Psi}(x),$$

where the last equation comes from the fact that $\mathbf{N}^T \mathbf{\Phi} = \mathbf{N}\mathbf{\Phi} = \mathbf{\Psi} - \mathbf{c}$. Therefore, the weighted sum of Green's functions corresponds to a unique biharmonic B-spline $\sum_j \lambda_j \psi_j$ for every $x \in \mathcal{M}$. ■

Throughout this paper, we call Equation (4) the *explicit* representation of the biharmonic B-spline, which explicitly reveals the relation between control points λ_j and basis functions ψ_j. In contrast, we call the Green's function based representation (Equation (7)) *implicit*, since the control points are "hidden" in the coefficients of Green's functions w_i.

Although the implicit representation seems to be unnatural at the first glance, it has three unique advantages over the explicit representation:

Firstly, unlike the basis function $\psi_j(x)$ which depends on the neighboring Voronoi cells of $\mathcal{V}_j$, each Green's function $\phi_i(x)$ is fully determined by t_i's location. Therefore, we can define a biharmonic B-spline without the Voronoi tessellation. Besides, local refinement is straightforward in the implicit representation, since we can add or delete a knot without changing the other Green's functions. In sharp contrast, the existing methods [2][3] have to re-tessellate the domain and re-compute the affected basis functions.

Secondly, the implicit representation reveals some remarkable results, which are quite vague in the explicit representation. For example, the knot insertion theorem in the explicit representation [2] has to re-compute the control points to keep the spline function unchanged. With our implicit representation, knot insertion becomes dramatically simple and intuitive: we can simply set the coefficient of the newly-inserted knots to 0 without changing the function $f(x)$. Also, the biharmonic B-spline bases are linear independent and the linear combination of two biharmonic B-spline (possibly with different knot configurations) is a biharmonic B-spline.

Theorem 2 (Linear Combination). Let $f_1 : \mathcal{M} \to \mathbb{R}^d$ (resp. f_2) be a biharmonic B-splines defined on knots $\mathcal{T}_1$ (resp. $\mathcal{T}_2$). Then the linear combination $\alpha_1 f_1 + \alpha_2 f_2$, $\alpha_1, \alpha_2 \in \mathbb{R}$, is also a biharmonic B-spline defined on knots $\mathcal{T}_1 \bigcup \mathcal{T}_2$.

Proof. Let $\mathcal{T}_1 = \{s_1, \ldots, s_m\}$ and $\mathcal{T}_2 = \{t_1, \ldots, t_n\}$.

By Theorem 1, $f_1(x) = a_1 + \sum_{i=1}^{m} w_{i,1} \phi_{s_i}(x)$ with $\sum_{i=1}^{m} w_{i,1} = 0$. $f_2(x)$ has a similar form. Since $\alpha_1 \sum_{i=1}^{m} w_{i,1} + \alpha_2 \sum_{j=1}^{n} w_{j,2} = 0$, $\alpha_1 f_1(x) + \alpha_2 f_2(x)$ is a weighted sum of Green's functions defined on knots $\mathcal{T}_1 \bigcup \mathcal{T}_2$ such that the coefficients add up to 0. ■

It is worth noting that the conventional tensor product B-splines do not have the linear combination property. Although the linear combination of two splines with different knot vectors can be written into a single spline, one has to insert many extra knots to ensure the correct topology of the knot configuration.

The linear combination property is not obvious at all in the explicit representation, since one must re-tessellate the Voronoi diagram for knots $\mathcal{T}_1 \bigcup \mathcal{T}_2$, then re-form the basis functions, and finally, re-compute the control points. In sharp contrast, our implicit representation allows reusing the existing Green's functions of f_1 and f_2, and the new coefficient can also be easily computed by linear combination.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2016.2605092, IEEE Transactions on Visualization and Computer Graphics

5 Biharmonic B-splines on Triangle Meshes

It is known that there exist symmetric Green's functions on compact and sufficiently smooth manifolds [22]. However, except for a few simple manifolds, neither the geodesic distance nor Green's function has an analytic expression on general manifold. Thus, one has to resort to numerical techniques for constructing biharmonic B-splines. In this section, we present a new computational framework for constructing biharmonic B-splines on triangle meshes, which are a dominant scheme for 3D representation.

Let $M = (V, E, F)$ be a closed manifold triangle mesh and V, E, and F are the sets of vertices, edges and faces, respectively. Denote $n = |V|$ the number of vertices of M and the set of knots $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$, $t_i \in M$. Since the knots t_i are arbitrary points on M, $\mathcal{T}$ does not necessarily coincide with the vertex set V. When a knot t_i falls in a triangular face, say $f_j \in F$, we can add t_i to V and re-tessellate f_j to three triangles. Thus, throughout the paper, we assume that the knot set $\mathcal{T}$ is the subset of the vertex set V, i.e., $\mathcal{T} \subseteq V$.

We define the discrete biharmonic B-spline $f : M \to \mathbb{R}^d$ as

$$f(x) = a + \sum_{i=1}^{m} w_i \phi_{t_i}(x), \qquad a, w_i \in \mathbb{R}^d,$$

where the coefficients add up to 0, $\sum_i w_i = 0$, and $\phi_{t_i}(x)$ are the discrete Green's function centered at knot t_i.

Define $g_{ij} = \phi_{t_i}(v_j)$ the Green's function centered at *knot* t_i evaluating at *vertex* v_j. We denote by $\mathbf{G} = (g_{ij})_{n\times m}$ the discrete Green's function on M. Lipman et al. [11] proved that $\mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{G} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T$. Denote by $\mathbf{M}_j$ the j-th column of matrix $\mathbf{M}$. They showed that $\mathbf{G}_j = \mathbf{x} - \frac{\mathbf{1}^T\mathbf{x}}{\mathbf{1}^T\mathbf{1}}\mathbf{1}$, where $\mathbf{x}$ is a particular solution to $\mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{x} = (\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T)_j$. To get the particular solution, replace the first row and the first column of $\mathbf{L}\mathbf{A}^{-1}\mathbf{L}$ by zeros and set the diagonal entry at their intersection to 1. Also replace the first row of $(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T)_j$ by 0. As pointed out in [11], this kind of linear system can be solved very efficiently by first performing Cholesky factorization of $\mathbf{L}\mathbf{A}^{-1}\mathbf{L}$ and then performing two backward substitutions for every given vector on the right hand side.

Given the coefficients $\mathbf{w}$ and a, we can evaluate the spline $f(x)$ at mesh vertices by first computing a particular solution of $(\mathbf{L}\mathbf{A}^{-1}\mathbf{L})\mathbf{x} = \mathbf{w}$ and then shifting its average to the constant a. See lines 2-3 in Algorithm 1.

Note that each Green's function $g_{ij} = \phi_{t_i}(v_j)$ is evaluated at mesh vertices v_j. Due to the lack of analytical formulation of Green's function on meshes, we have to seek numerical techniques to approximate $\phi_{t_i}(x)$ for a point $x \in M$ but $x \notin V$. We observe that the simple linear interpolation produces poor results, since Green's functions are highly nonlinear. In our implementation, rather than for an arbitrary point x, we approximate discrete Green's function for vertices of the planar subdivision[1] of M.

Let $M' = (V', E', F')$ be a planar subdivision (in one or a few rounds) of M, thus, $V \subset V'$. For each Green's function $\phi_{t_i}(x)$, we use biharmonic interpolation $\Delta^2\phi_{t_i}(x) = 0$ for any $x \in V'$. Since V' is the superset of V, Green's functions on some of its vertices have known values, which are used as the constraints.

Let $\mathbf{L}'$ be the Laplacian matrix for mesh M'. We first re-organize $\mathbf{L}'$ into $\mathbf{L}' = [\mathbf{L}_l, \mathbf{L}_r]$, so that each column of $\mathbf{L}_r$ corresponds to a vertex of the original mesh M, on which Green's function is already known. Then, we factorize the matrix $\mathbf{L}_l^T\mathbf{L}_l$ using Cholesky decomposition. Next, we compute Green's function for each vertex of V' in a column-by-column manner. See lines 5-12.

Algorithm 1: Spline Evaluation

Input : Coefficients $\mathbf{w}$ and a, and the planar subdivided mesh M'
Output: Function values $\mathbf{f} = [\ldots, f(v_k), \ldots]^T$, $v_k \in V'$

1 **if** $V' = V$ **then**
2 Compute a particular solution of $(\mathbf{L}\mathbf{A}^{-1}\mathbf{L})\mathbf{x} = \mathbf{w}$;
3 $\mathbf{f} \leftarrow a\mathbf{1} + \mathbf{x} - \frac{1}{n}\mathbf{1}\mathbf{1}^T\mathbf{x}$;
4 **else**
5 Construct the Laplacian matrix $\mathbf{L}'$ of the M';
6 Reorganize $\mathbf{L}' = [\mathbf{L}_l, \mathbf{L}_r]$;
7 Factor $\mathbf{L}_l^T\mathbf{L}_l$ using Cholesky decomposition;
8 $\mathbf{f} \leftarrow \mathbf{0}$;
9 **for** *each knot* $t_j \in \mathcal{T}$ **do**
10 Compute Green's function $\mathbf{G}_j$ centered at knot t_j;
11 Solve $\begin{bmatrix} \mathbf{L}_l^T\mathbf{L}_l & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n\times n} \end{bmatrix}\mathbf{x} = \begin{bmatrix} \mathbf{0} \\ \mathbf{G}_j \end{bmatrix}$ using backward substitutions;
12 $\mathbf{f} \leftarrow \mathbf{f} + w_j\mathbf{x}$;
13 $\mathbf{f} \leftarrow \mathbf{f} + a\mathbf{1}$;

Remark 1. Since the discrete Green's function $\mathbf{G}_{n\times m}$ is a dense matrix, our spline evaluation algorithm does not store it explicitly, which is very expensive. Instead, we compute each Green's function on the fly (see line 10).

Remark 2. Biharmonic B-spline is parameterization-free, making it different than the conventional B-splines. To evaluate the biharmonic B-spline $f(x)$ at a point x, we require that x is on the domain mesh M. Thus, in Algorithm 1, the mesh M' must be a *planar* subdivision of M, with exactly the same geometry. As Figure 2 shows, for a point $x \notin M$, the value $f(x)$ makes no sense.

6 Knot Optimization

As shown in Section 7.1, biharmonic B-spline can perfectly represent any vector-valued data defined on a triangle mesh by taking all vertices as knots. We can also fit a biharmonic B-spline to the given data within the user-controlled number of knots. Thanks to our implicit representation that associates each Green's function a single knot, we are able to develop a ℓ_1-based scheme towards better numerical robustness and accuracy.

Let $\mathbf{s} = [s_1, \ldots, s_n]^T$, $s_i \in \mathbb{R}^d$ be the input discrete signal defined on the mesh vertices. To ease the representation of our matrix oriented algorithm, we rewrite $\mathbf{s}$ in an n-by-d matrix $\mathbf{S} \in \mathbb{R}^{n\times d}$, where the i-th row of $\mathbf{S}$ is the d-dimensional data stored at vertex v_i.

Given a spline $f : M \to \mathbb{R}^d$ which is defined on the knot set $\mathcal{T}$ and has coefficients $\mathbf{w}$, we denote by $\mathbf{S}_f$ the

1. The planar subdivision is to split each edge at its middle point.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2016.2605092, IEEE Transactions on Visualization and Computer Graphics

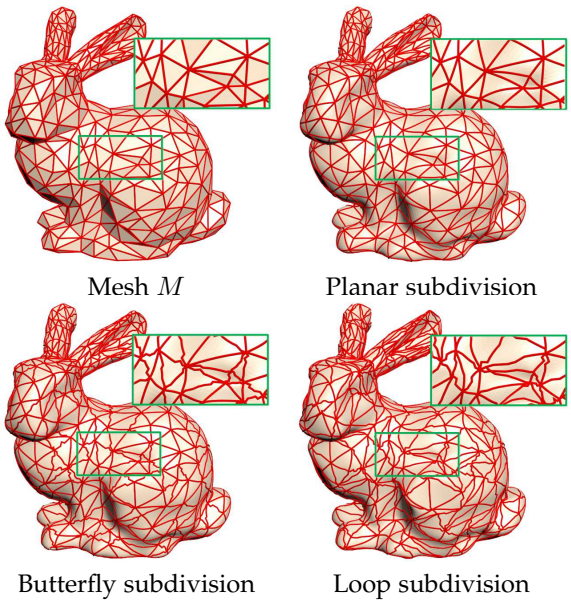


Fig. 2: Evaluating the splines using planar subdivision and other subdivision schemes. Only planar subdivision produces correct results. With Loop subdivision and butterfly subdivision, the subdivided mesh M' has different geometry than the domain mesh M. Thus, the value $f(x)$ makes no sense for a point $x \notin M$. The red curves are the images of the edges of M.

approximated data, i.e., the i-th row of $\mathbf{S}_f$ is $f(v_i) \in \mathbb{R}^d$. Our goal is to find the knot set $\mathcal{T}$ with as few as possible knots and the corresponding coefficients $\mathbf{w}$ so that $\|\mathbf{S} - \mathbf{S}_f\|_F^2 \leq \epsilon$, where ϵ is the user-specified error bound.

Our algorithm consists of two steps. We first determine the number of knots and their knot locations via the weighted ℓ_1 optimization. Then, fixing the knots, we compute the coefficients $\{w_i\}$ and a for the biharmonic spline $f(x) = a + \sum_i w_i \phi_i(x)$.

Step 1) Optimal Knot Selection. ℓ_1-norm minimization has proven highly effective to discover the sparsity in the optimization problem. A naïve way to determine $\mathcal{T}$ with the fewest knots is as follows: First, take a very large pool of candidate knots, e.g, $\mathcal{T} = V$. Then, minimize the following ℓ_1-norm of coefficients $\{w_i\}$:

$$\min_{\mathbf{w}} \|\mathbf{w}\|_1, \ \text{subject to} \ \|\mathbf{S} - \mathbf{S}_f\|_F^2 \leq \epsilon.$$

A small coefficient w_j means the contribution of $\phi_j(x)$ is not significant, as a result, we can ignore t_j without increasing the fitting error too much. This method is conceptually simple, however, it is not practical due to 2 reasons: 1) as $\mathcal{T} = V$, it is very expensive to store the discrete Green's function $\mathbf{G} \in \mathbb{R}^{|V| \times |V|}$, which is a dense matrix; and 2) even though one has sufficient memory to store $\mathbf{G}$, solving the quadratic programming on such dense matrix is extremely time consuming.

Observe that the spline coefficients $\mathbf{w} = \mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{S}$ for the interpolatory spline (see line 2 in Algorithm 2). Inspired by the work of [18], we minimize the fitting error $\|\mathbf{S} - \mathbf{S}_f^{(j)}\|_F^2$ under reweighted ℓ_1 constraint [23] for better approximation of ℓ_0 norm. In the j-th iteration, we minimize the following energy to obtain an optimal $\mathbf{S}_f^{(j)}$,

$$\begin{aligned} &\min_{\mathbf{S}_f^{(j)}} \|\mathbf{S} - \mathbf{S}_f^{(j)}\|_F^2, \\ \textit{subject to} \quad &\sum_i \|w_i^{(j)} (\mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{S}_f^{(j)})_i\|_1 < K, \ i \in \overline{Z} \\ &(\mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{S}_f^{(j)})_i = 0, \ i \in Z \end{aligned} \tag{9}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $(\cdot)_i$ denotes the i-th row of the matrix, and $K \in \mathbb{R}^+$ is the number of knots we want to use, which controls the sparsity of the coefficients $\mathbf{w}$ and the fitting error. Z is the set of knots whose coefficients are zero and $\overline{Z}$ is the complement of Z. $w_i^{(j)}$ is the weight of the i-th knot,

$$w_i^{(j+1)} = \frac{1}{\|(\mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{S}_f^{(j)})_i\|_1 + \epsilon}, \tag{10}$$

where $\epsilon = 1 \times 10^{-7}$ is to avoid numerical problems. We initialize the weights $w_i^{(1)}$ as $\frac{1}{\|(\mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{S})_i\|_1 + \epsilon}$ and label the knots, whose weights $\|(\mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{S})_i\|_1$ are among the smallest 40%, to set Z. Inspired by the feature-sign search algorithm [24] to predict the coefficient signs, we set the coefficients of elements in Z to zero instead of assigning them large weights as in [23] and [18]. Computational results show that this strategy can improve the performance significantly.

Remark. It is difficult to develop a knot selection scheme in the explicit representation [2][3], due to the complicated relationship between knots and the basis functions $\psi_j(x)$: the basis functions have various number of knots and conversely each knot is associated to various number of basis functions. With our implicit representation, each knot determines a unique Green's function and vice versa, making it the ideal representation for knot optimization.

Step 2) Spline Construction. Since the above ℓ_1-optimization mainly aims at finding a set of sparse knots $\mathcal{T} \subset V$, the resulting coefficients $\mathbf{w}$ are not optimal. Thus, fixing the knots $\mathcal{T}$, we further optimize $\mathbf{S}_f$ by the following linear constrained quadratic programming:

$$\min_{\mathbf{S}_f} \|\mathbf{S} - \mathbf{S}_f\|_2^2 \tag{11}$$

$$\text{subject to} \quad (\mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{S}_f)_j = 0, \ \forall v_j \in V \setminus \mathcal{T}, \tag{12}$$

where $(\mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{S}_f)_j$ is the j-th row of matrix $\mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{S}_f$. The constraints in Equation (12) apply to the *non-selected* vertices and make their corresponding coefficients w_i zero. We solve this linear constrained quadratic programming by Lagrange multiplier.

7 Applications and Results

The proposed implicit representation has several favorable properties for modeling and processing discrete vector-valued data on manifolds. For example, our data interpolation algorithm does not require any global solver, thus, it is efficient and numerically stable. Our data fitting algorithm

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2016.2605092, IEEE Transactions on Visualization and Computer Graphics

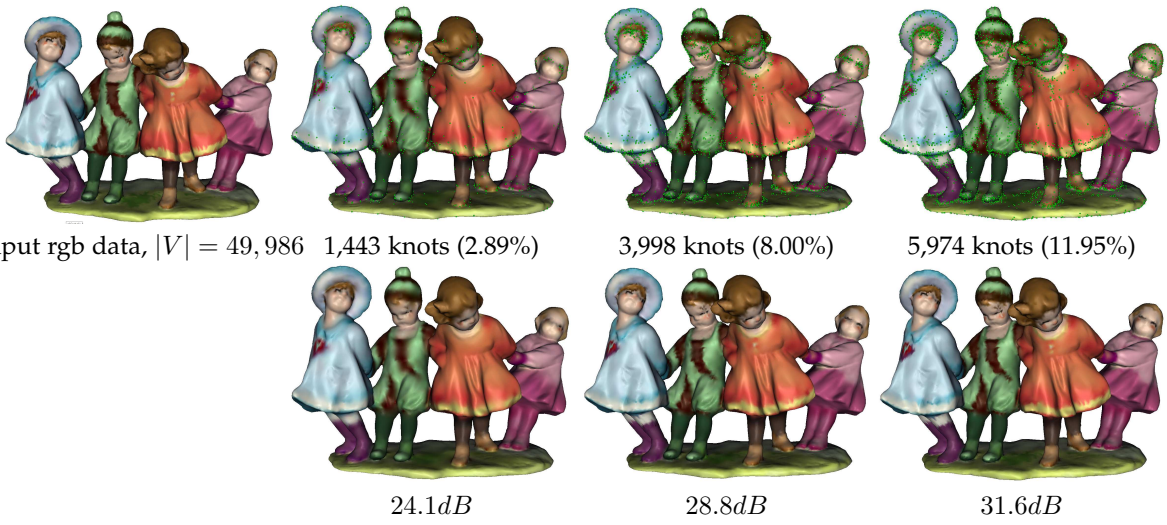


Fig. 3: Fitting vector-valued data (i.e., rgb color) defined on a genus-8 model. Top: the input data and the knots. The percentage shows the knot-vertex ratio, i.e., $\frac{|\mathcal{T}|}{|V|}$. Bottom: the reconstructed data.

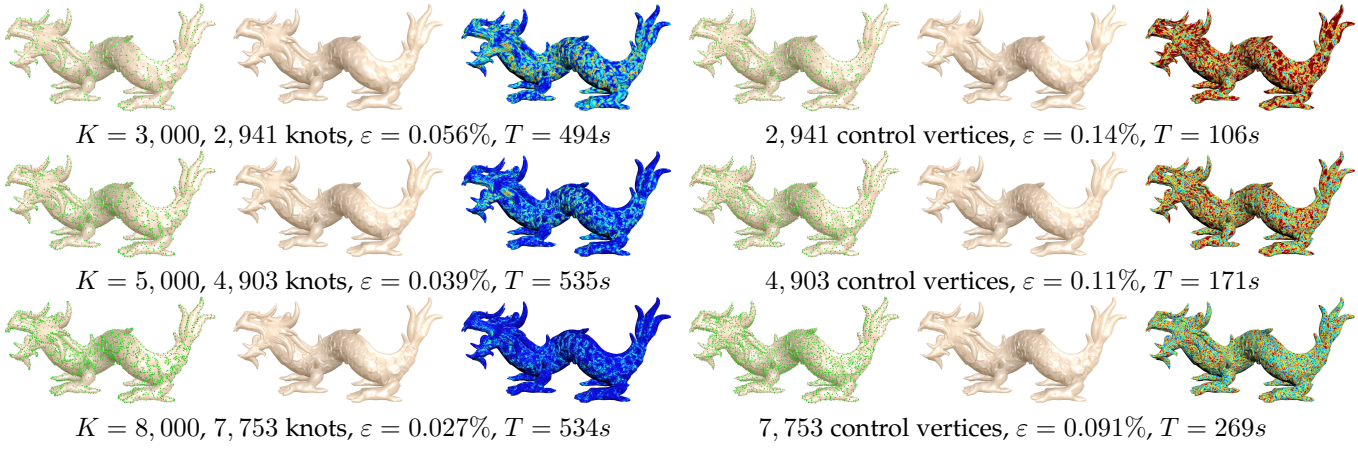


Fig. 4: Comparing biharmonic B-splines with LS-meshes in terms of approximation error and runtime performance. Biharmonic B-splines determine the knots by ℓ_1 optimization, whereas LS-meshes choose the control vertices by greedy selection combined with local error maxima. With the same number of knots and control vertices, biharmonic B-splines consistently outperform LS-meshes in terms of approximation error. However, LS-meshes are 2 to 5 times faster to construct than biharmonic B-splines. We visualize the fitting errors using colors, where warm colors indicate large error and cold colors small error.

allows knot optimization on a fully irregular knot configuration – such a property is not available to the conventional tensor-product B-splines and triangular B-splines. The implicit representation also enables linear combination of biharmonic B-splines defined on different knots, providing a natural way for hierarchical data decomposition.

7.1 Data Interpolation

Define $\mathbf{s} = [s_1, \ldots, s_n]^T$, where $s_i \in \mathbb{R}^d$ is the data defined at vertex v_i. Our goal is to find a biharmonic B-spline $f : M \to \mathbb{R}^d$ interpolating the given data, that is, $f(v_i) = s_i$, $i = 1, \ldots, n$. The following theorem states that such spline always exists and is unique if the knot set coincides with the vertex set $\mathcal{T} = V$.

Theorem 3 (Data Interpolation). Given a triangle mesh $M = (V, E, F)$ and any data $\mathbf{s} = [s_1, \ldots, s_n]^T$, $s_i \in \mathbb{R}^d$ defined on the vertices of M, there exists a unique biharmonic B-spline $f : M \to \mathbb{R}^d$ whose knot set $\mathcal{T}$ coincides with the vertex set $\mathcal{T} = V$ such that f interpolates the data exactly, i.e., $f(v_i) = s_i$, $i = 1, \ldots, n$.

Proof. We show that there exist unique coefficients $a \in \mathbb{R}$ and $\mathbf{w}$ such that $\mathbf{s} = a\mathbf{1} + \mathbf{G}\mathbf{w}$. Since $\mathcal{T} = V$ and $m = n$, we simply denote by $\mathbf{G}$ the discrete Green's function $\mathbf{G} = [\phi_{t_i}(t_j)]_{m \times m}$. Since $a\mathbf{1} + \mathbf{G}\mathbf{w} = [\mathbf{G}, \mathbf{1}][\mathbf{w}^T, a]^T$, it is equivalent to prove $\mathrm{rank}([\mathbf{G}, \mathbf{1}]) = \mathrm{rank}([\mathbf{G}, \mathbf{1}, \mathbf{s}])$.

Denote $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_m]$. Recall that $\mathbf{1}^T\mathbf{g}_j = 0$, $j = 1, \ldots, m$ and $\mathrm{rank}(\mathbf{G}) = m - 1$.

Let $\sum_{i=1}^{m-1} k_i\mathbf{g}_i + k_m\mathbf{1} = \mathbf{0}$. Multiply $\mathbf{1}^T$ on both sides of the equation. We derive $k_m\mathbf{1}^T\mathbf{1} = \mathbf{0}$, thus $k_m = 0$. Since $\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_{m-1}$ are linearly independent, $k_1 = k_2 \ldots = k_{m-1} = 0$. Therefore, $\mathrm{rank}([\mathbf{G}, \mathbf{1}]) = m$. Since $\mathrm{rank}([\mathbf{G}, \mathbf{1}]) \leqslant \mathrm{rank}([\mathbf{G}, \mathbf{1}, \mathbf{b}]) \leqslant m$, we have

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2016.2605092, IEEE Transactions on Visualization and Computer Graphics

Algorithm 2: Knot Optimization

Input : Triangle mesh $M = (V, E, F)$, input data $\mathbf{S} \in \mathbb{R}^{n\times d}$ defined on mesh vertices, the maximal number of knots K, the maximal number of iterations I_{max}

Output: The knot set $\mathcal{T}$ and the coefficients $\mathbf{w}$ and a of the biharmonic B-spline function $f(x) = a + \sum_i w_i\phi_i(x)$

$\mathcal{T} \leftarrow V$;
Initialize set Z and $\overline{Z}$;
Initialize the weights $w_i = \frac{1}{\|(\mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{S})_i\|_1}$;
$i \leftarrow 0$;
while $i < I_{max}$ *and* $|\mathcal{T}| \geq K$ **do**
    Solve the optimization in Equation (9);
    Update $\mathcal{T}$;
    Update weights by Equation (10);
    $i++$;
Fixing $\mathcal{T}$, solve the optimization in Equation (11);
Compute the coefficients $\mathbf{w} \leftarrow \mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{S}$ and $a \leftarrow \frac{1}{m}\mathbf{1}^T\mathbf{S}$;

$\mathrm{rank}([\mathbf{G}, \mathbf{1}]) = \mathrm{rank}([\mathbf{G}, \mathbf{1}, \mathbf{b}]) = m$, which completes the proof. ■

Algorithm 3: Data Interpolation with Linear Time Complexity

Input : Triangle mesh $M = (V, E, F)$ and data $\mathbf{s} = [s_1, \ldots, s_n]^T$ defined at mesh vertices V

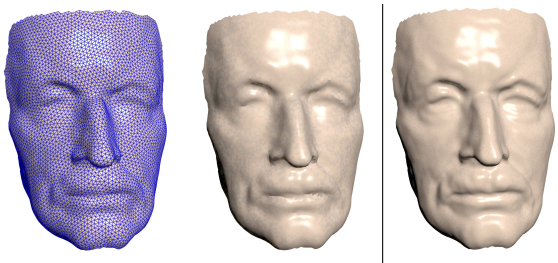
Output: The coefficients $\mathbf{w}$ and a of $f(x) = a + \sum_{i=1}^{m} w_i\phi_i$, $a, w_i \in \mathbb{R}^d$, such that $f(v_i) = s_i$, $i = 1, \ldots, n$

$\mathcal{T} \leftarrow V$ and $m \leftarrow n$;
$\mathbf{w} \leftarrow \mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{s}$;
$a \leftarrow \frac{1}{m}\mathbf{1}^T\mathbf{s}$;

Since the discrete Green's function $\mathbf{G}$ is a dense matrix, it is not space efficient to store the matrix when the mesh is of high resolution. In fact, we can compute the coefficients w_i and a without the discrete Green's function $\mathbf{G}$. By Theorem 3, there exist $\mathbf{w}$ and a such that $\mathbf{s} = a\mathbf{1} + \mathbf{G}\mathbf{w}$. Multiplying $\mathbf{L}\mathbf{A}^{-1}\mathbf{L}$ to both sides of this equation, we have $\mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{s} = a\mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{1} + \mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{G}\mathbf{w}$. Note that $\mathbf{L}\mathbf{1} = \mathbf{0}$ and $\mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{G} = \mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{G} = \mathbf{I} - \frac{1}{m}\mathbf{1}\mathbf{1}^T$, $\mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{s} = \mathbf{w} - \frac{1}{m}\mathbf{1}\mathbf{1}^T\mathbf{w}$, and $\mathbf{1}^T\mathbf{w} = 0$. We obtain $\mathbf{w} = \mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{s}$ and $a = \frac{1}{m}\mathbf{1}^T\mathbf{s}$, which implies that a is just the average of input data $\mathbf{s}$ and the coefficient w_i approximates the integral of bi-Laplacian $\Delta^2 s_i$. Since our data interpolation algorithm neither requires discrete Green's function nor solves any linear system, it is numerically stable and has a linear time complexity $O(n)$.

Figure 5 shows the biharmonic B-spline interpolating a Bunny model with 500 vertices. Although our algorithm is designed for closed triangle meshes, it can handle open meshes by double covering. Also note that the Max Planck model in Figure 6 has one boundary. Gluing two identical copies along their common boundary, we obtain a closed mesh, on which our biharmonic B-spline can be defined.

Fig. 5: Interpolatory spline on the Bunny model with $|V| = 500$ vertices. The biharmonic B-spline $f(x)$ interpolates all vertices, $f(v_i) = v_i$, $i = 1, \ldots, |V|$. The red curves are the images of the edges of M.



(a) (b)

Fig. 6: The Max Planck model is an open mesh. We first convert it to a closed mesh by double covering and then construct the interpolatory spline on the double-covered mesh. Therefore, the number of knots in the spline is almost twice of the number of vertices (excluding the boundary vertices). (a) Interpolated mesh $|V| = 8,745$ (double cover). (b) Spline $|V| = 34,974$.

7.2 Hierarchical Data Decomposition

The linear combination property allows us to decompose the input vector-valued data into a hierarchical structure. We adapt the knot optimization algorithm for data decomposition. Given a d-dimensional data $\mathbf{S} \in \mathbb{R}^{n\times d}$ defined on mesh vertices, our goal is to find a sequence of biharmonic B-splines $\mathbf{S}_1, \mathbf{S}_2, \cdots$, such that $\sum_i \mathbf{S}_i$ approximates $\mathbf{S}$ well.

Similar to data fitting, we adopt the ℓ_1 optimization to determine the optimal knot locations. For the i-th layer, we solve the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{S}_i} \|\mathbf{C}_i(\mathbf{S} - \sum_{j=1}^{i}\mathbf{S}_j)\|_F^2, \\ &s.t. \sum_k \|w_k(\mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{S}_i)_k\|_1 < K_i,\ k \in \overline{Z} \\ &(\mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{S}_i)_k = 0,\ k \in Z \end{aligned} \tag{13}$$

where the diagonal weight matrix $\mathbf{C}_i$ is the user-specified weight and K_i is the maximal number of knots for spline $\mathbf{S}_i$. The last layer can be simply set as the residue. Thanks to the linear combination property, users can easily modify the data via manipulating the coefficients, i.e., $\mathbf{S}_1 + \sum_{i\geq 2}\lambda_i\mathbf{S}_i$, where $\lambda_i \geq 0$. Figure 7 demonstrates hierarchical data decomposition and details enhancement and exaggeration on the Armadillo and Bunny models. We decompose the input models into 3 layers and represent the first two by biharmonic B-splines and the third layer as the residue, i.e., $\mathbf{S} = \mathbf{S}_1 + \mathbf{S}_2 + \mathbf{S}_3$. We can control the spline quality by specifying spatially-varying weights on 3D

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2016.2605092, IEEE Transactions on Visualization and Computer Graphics

models. For example, the region-of-interests (in purple) are given large weights in the first and second layers so that the resulting splines $\mathbf{S}_1 + \mathbf{S}_2$ reconstruct the details in these regions very well. The non-selected regions, receiving few knots, are smooth and lack details. Thanks to the linear combination property, we can enhance and exaggerate the details by manipulating the coefficient for the third layer $\mathbf{S}_3$, which encodes the high-frequency details.

Algorithm 4: Hierarchical data decomposition

Input : Triangle mesh $M = (V, E, F)$, input data $\mathbf{S} \in \mathbb{R}^{n\times d}$ defined at vertices, the number of decomposed layers m, and the weight $\mathbf{C}_i$ and the number of knots K_i for the i-th layer

Output: The decomposed data $\mathbf{S}_1, \ldots, \mathbf{S}_m$, where $\mathbf{S} = \mathbf{S}_1 + \cdots + \mathbf{S}_m$

$i \leftarrow 1$;
while $i < m$ **do**
  Initialize set Z and $\overline{Z}$;
  Initialize the weights $w_k = \frac{1}{\|(\mathbf{L}\mathbf{A}^{-1}\mathbf{L}\mathbf{S})_k\|_1},\ k \in \overline{Z}$;
  Solve the optimization in Equation (13);
  $\mathbf{S} \leftarrow \mathbf{S} - \mathbf{S}_i$;
  $i \leftarrow i + 1$;
$\mathbf{S}_m \leftarrow \mathbf{S}$;

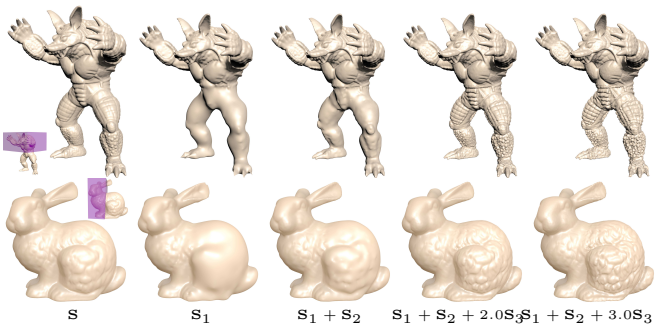


Fig. 7: Hierarchical data decomposition.

7.3 Results

We implemented our algorithm in C++ with trimesh2 and tested it on a laptop with an Intel i7 CPU2.40GHz. The models are rendered by 3DS Max with auto-smoothing preprocessing. It is worth noting that our data interpolation and data fitting algorithms can be applied to any vector-valued data defined on triangle meshes. For simplicity, in this paper, we demonstrate the efficacy of our algorithms on the coordinate functions for geometry modeling.

The data interpolation algorithm is highly efficient, since it directly computes the coefficients $\mathbf{w}$ and a, hereby has a linear time complexity. The data fitting algorithm consists of two steps, i.e., knot optimization and spline construction. The first step determines the optimal knots from a very large pool (i.e., the set of all vertices), thus, its performance highly depends on the mesh complexity. The second step is fairly efficient and takes only a few minutes on the test models. See Table 1 for the mesh complexity and runtime performance of the data fitting algorithm. The performance of the evaluation algorithm depends on both mesh complexity and the number of knots. It takes 162 seconds to compute 60K points on the 2,094-knot spline for the Heptoroid mesh with 15K vertices.

Model	$\lvert V\rvert$	$\lvert\mathcal{T}\rvert$	T_k (s)	$iter$	T_f (s)	ϵ
Dragon	99,999	2,941	442s	4	52s	0.056%
		4,903	489s	4	45s	0.039%
		7,753	485s	4	49s	0.027%
Armadillo	172,974	6,873	668s	1	164s	-
		24,182	604s	1	150s	-
Heptoroid	21,826	3,124	192s	4	33s	0.044%
Bunny	72,020	2880	271s	1	79s	-
		11,409	247s	1	75s	-
4-kid	49,986	1,443	263s	6	33s	$24.1dB$
		3,998	252	5	30s	$28.8dB$
		5,974	256s	5	27s	$31.6dB$
Gargoyle	28,632	264	72s	1	13s	-

TABLE 1: Runtime performance of knot optimization. The timings are broken down into knot optimization T_k and data fitting T_f. $|V|$: the number of vertices in the mesh; $|\mathcal{T}|$: the number of knots in the spline; ϵ: the normalized mean error; $iter$: the number of iterations in the knot optimization step.

Figure 4 visualizes the fitting results on the Dragon model. We can see that the parameter K (i.e., the maximal number of knots) directly controls the fitting quality.

Even though the Green's functions are globally supported, the basis functions of biharmonic B-spline $\psi(x)$ are localized and form a partition of unity. After deriving the formulation of $f(x) = a + \sum_i \omega_i \phi_{t_i}(x)$, we evaluate the basis functions $\psi(x)$ and convert the $f(x)$ to $f(x) = \sum_i \lambda_i \psi_i(x)$. The conversion algorithm is detailed in the proof of Theorem 1.

We evaluate the basis functions $\psi_i(x)$ in a manner similar to the method proposed by Hou et al. [3], but with more neighboring knots for robustness. Figure 8 demonstrates free-form deformation using biharmonic B-splines. Similar to the method in Section 7.2, we first decompose the model $\mathbf{S}$ into the sum of two frequency components $\mathbf{S}_1$, $\mathbf{S}_2$, with equal weights. Then we evaluate the basis functions $\psi(x)$ of the knots of $\mathbf{S}_1$, so, we can convert the formulation of $\mathbf{S}_1$ to Feng and Warren's formulation. We adjust the control points λ_i of certain knots to deform the shape of $\mathbf{S}_1$ locally and finally add $\mathbf{S}_1$, $\mathbf{S}_2$ together to derive the deformed shape. Rather than evaluating the spline using the explicit representation with updated λ_i, we compute the control coefficients ω_i by $f(x) = a + \sum_i \omega_i \phi_{t_i}(x)$ and then compute the new vertices by Algorithm 1 with $V' = V$. Since the matrix $\mathbf{L}\mathbf{A}^{-1}\mathbf{L}$ has been predecomposed and remains unchanged during deformation, we just need a backward substitution to solve the equation. Thanks to the localized basis functions, the splines are deformed locally.

It is worth noting that in [25], if the handles are changed, the basis functions must be recomputed, while our biharmonic B-spline basis functions remain unchanged if the knot configuration does not change.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2016.2605092, IEEE Transactions on Visualization and Computer Graphics

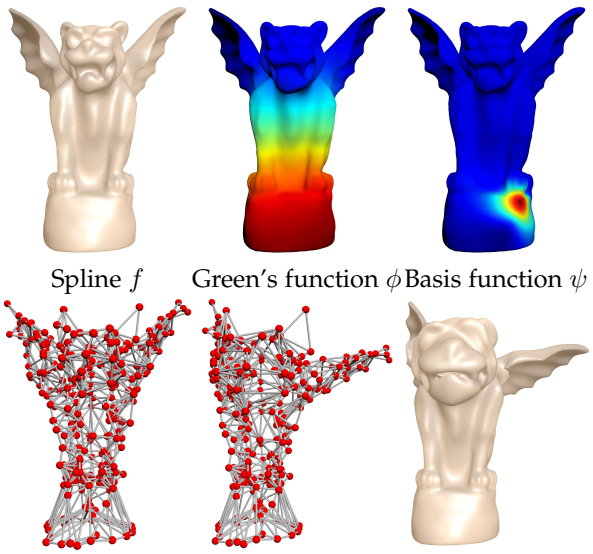


Fig. 8: Data decomposition and local control using the explicit representation, which is constructed using [2]. Although Green's functions ϕ are globally defined, the biharmonic basis functions ψ are localized. Therefore, one can directly manipulate the control points to edit the spline.

	Biharmonic B-splines	Tensor-product B-splines	T-splines	Triangular B-Splines
Localization	Yes	Yes	Yes	Yes
Partition of Unity	Yes	Yes	Yes	Yes
Knot configuration	Arbitrary	Regular grid	Regular grid with T-junctions	Triangulation
Parameterization	No	Yes	Yes	Yes
Singularity	No	Yes	Yes	Yes
Linear combination	Yes	No	No	No
Basis function	Analytical or discrete	(Rational) polynomial	Rational polynomial	Polynomial
Knot insertation	Yes	Yes	Yes	No
Knot optimization	Easy	Difficult	N.A.	N.A.
Evaluation	Expensive	Efficient	Efficient	Efficient
Derivative	Expensive	Efficient	Efficient	Efficient

TABLE 2: Comparison of various splines.

8 Comparisons & Discussions

8.1 Comparison with Least Square Meshes

Least square meshes (LS-meshes), proposed by Sorkine and Cohen-Or [4][5], are an elegant, compact and efficient representation of irregular triangle meshes. Let $G = (V, E)$ the graph associated to the triangle mesh $M = (V, E, F)$. The graph-Laplacian $\mathbf{L}$ of G is defined as follows:

$$L_{ij} = \begin{cases} 1, & \text{if } i = j \\ -\frac{1}{d_i}, & \text{if } (i,j) \in E \\ 0, & \text{otherwise.} \end{cases}$$

Let $C = \{v_{s_i} | v_{s_i} \in V, 1 \leq i \leq m, m \ll n\}$ be a *sparse* set of control vertices. LS-meshes approximate the geometry of M by solving the following linear least-squares problem

$$\min_{\mathbf{x}} \|\mathbf{L}\mathbf{x}\|^2 + \sum_{v_j \in C} \|x_j - v_j\|^2,$$

with the solution $\mathbf{x} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{b}$, where

$$\mathbf{M} = \begin{pmatrix} \mathbf{L} \\ \mathbf{F} \end{pmatrix}, F_{ij} = \begin{cases} 1 & j = s_i \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

$$\mathbf{b}_k = \begin{cases} 0 & k \leq n \\ v_{s_{k-n}} & n < k \leq n + m. \end{cases} \tag{15}$$

For matrix $(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T$, Sorkine and Cohen-Or showed that the columns corresponding to the control vertices are in fact basis functions, called LS basis, so that one can represent the geometry as a linear combination of them. The LS basis has several favorable properties for geometry modeling and processing. The basis functions are derived from both the mesh connectivity and limited geometrical information, and can be applied to meshes of arbitrary topology and connectivity. They tag specific "geometrically important" control vertices, hereby they are geometry-aware. Comparing with other popular basis, such as eigenvectors of Laplacian, the LS basis is fast to compute.

A critical step of LS-mesh construction is to choose control vertices to bring the LS-mesh close to the original mesh and minimize the approximation error. The greedy approach places control vertices one-by-one at vertices with highest reconstruction error. This method is able to achieve the best distribution of the control points and hence the smallest geometric error, but it is computationally expensive since it requires solving the least-squares system in each step. To improve the performance, Sorkine and Cohen-Or combined the greedy selection with local error maxima, which computes the LS-mesh every K steps and marks the vertex with maximal error as a control point, like in the greedy method. In the $K - 1$ steps in between, it selects control points by computing local maxima of the error. They observed that the combined approach is more practical in making the tradeoff between the approximation error and computation time.

LS-meshes and our method share four common features. First, both the LS basis and Green's function ϕ_i of bi-Laplacian are defined on the *entire* mesh. Thus, both methods represent the geometry as a linear combination of *global* functions. Second, both methods avoid *explicit* computation of the underlying basis in order to achieve good performance. Third, both methods can be defined on smooth manifolds and triangle meshes. Fourth, both the LS basis and biharmonic B-spline basis are *approximately* localized (i.e., the function values drop to 0 quickly, but they are not exactly 0) and they form partition of unity. However, there are fundamental differences between LS-meshes and biharmonic B-splines, which are listed as follows:

1) For biharmonic B-splines, the implicit representation and explicit representation are complementary to each other: Green's functions are globally defined and easy to compute, whereas the biharmonic B-spline basis functions are localized and form partition of unity. Therefore, users can choose the proper representation depending on their applications. Our implicit representation allows us to decouple the knots in Green's functions, hereby we can solve

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2016.2605092, IEEE Transactions on Visualization and Computer Graphics

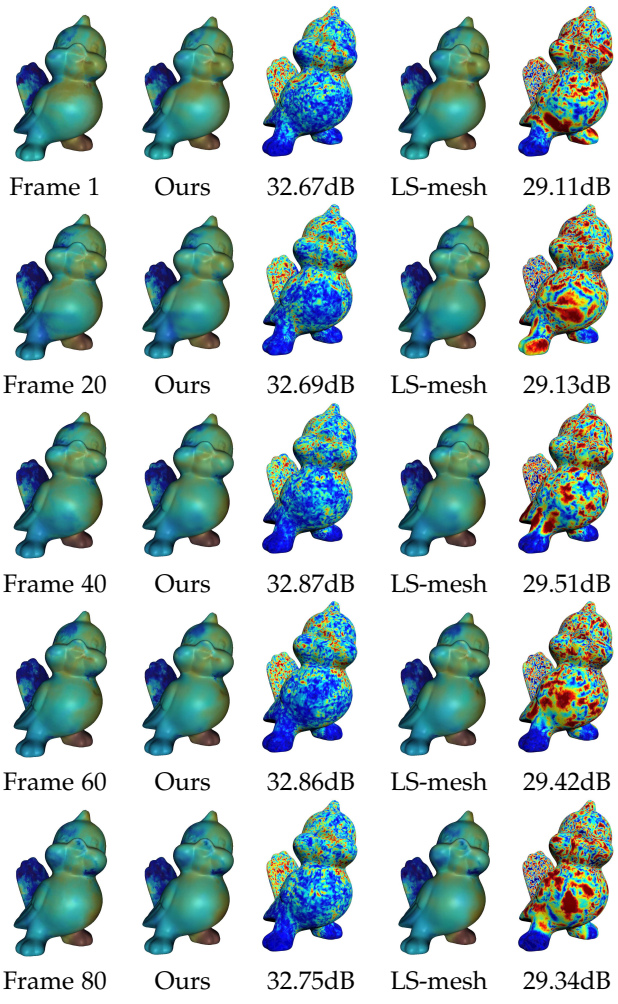
the problem of knot optimization, which is a central and challenging problem in spline theory. We note that each LS-basis function is also associated with several knots. However, it is not clear whether their basis functions have the knots-*decoupled* feature. Therefore, it is technically challenging to optimize the knots using LS-meshes. With the same number of knots in biharmonic B-splines and the number of control vertices in LS-meshes, we observe that biharmonic B-splines consistently outperform LS-meshes in terms of approximation error. However, LS-meshes are faster to compute than biharmonic B-splines. See Figure 4.

2) LS-meshes are a compact representation, which recovers the geometry from the connectivity of the input mesh and a sparse set of control vertices with geometry, therefore, they are ideal for geometry compression and progressive transmission. Biharmonic B-splines, in contrast, take more space than LS-meshes, since we have to store the entire mesh. However, biharmonic B-splines enable knot optimization and evaluation on any planar subdivision of the domain mesh, hence are promising for signal processing.

Figure 9 demonstrates biharmonic B-splines on fitting time-varying data on a 50K-vertex Bird model. For a fair comparison, both the biharmonic B-splines and the LS-meshes are given the same number of knots and control vertices (4,927 on average, roughly 10% of the number of mesh vertices). Biharmonic B-splines are superior in terms of PSNR and evaluation time, whereas LS-meshes are more efficient to construct. Given the same number of knots and control points, the biharmonic B-splines are 3.47dB more accurate than LS-meshes. To recover the geometry from a sparse set of control points, LS-meshes solve a linear system $\mathbf{Mx} = \mathbf{b}$ (see Equations (14) and (15)), where the matrix $\mathbf{M} \in \mathbb{R}^{(n+m)\times(n+m)}$ and the vector $\mathbf{b}$ are dependent of the control points. Since the vertices selected as the control points are varying frame by frame, the matrix $\mathbf{M}$ is not a constant, hence it is time consuming for LS-meshes to recover their geometries. In contrast, biharmonic B-splines represent each frame of the data using coefficients $\mathbf{w}$ and vector a. To recover the geometry, biharmonic B-splines solve a linear system $\mathbf{LA}^{-1}\mathbf{Lx} = \mathbf{w}$. Note that the left-hand side $\mathbf{LA}^{-1}\mathbf{L} \in \mathbb{R}^{n\times n}$ is *data independent*, hence can be pre-decomposed. As a result, given different $\mathbf{w}$, biharmonic B-splines simply perform a forward/backward substitution to solve the geometry $\mathbf{x}$. Computational results show that evaluating biharmonic B-splines is 40 times faster than that of LS-meshes, and construction of LS-meshes is 3 times more efficient than that of biharmonic B-splines. See the statistics in Figure 9.

Remark 1. In the continuous setting, the integral in Equation (2) falls to zero out of the Voronoi cell $\mathcal{V}_j$, hence the basis functions ψ_j are strictly localized. However, on triangle meshes, the integration is approximated by a weighted sum of discrete Green's functions (see Equation (3)), which are *not* strictly zeros along the boundary of $\mathcal{V}_j$. Therefore, the local property of biharmonic B-splines on meshes does not hold strictly. Nevertheless, we observe that such an approximate local property works pretty well in real-world examples. See the local control in shape deformation in Figure 8.

Remark 2. In [25], Botsch and Kobbelt developed a variant of LS-meshes for freeform modeling. Their basis functions rely on the size of the handle and support regions, which is user-specified and could be global. Also, they are mainly designed for efficient computation and lack of geometry meanings. Other similar basis functions, such as [26][27] for cage deformation, are defined in Euclidean spaces rather than on manifolds.



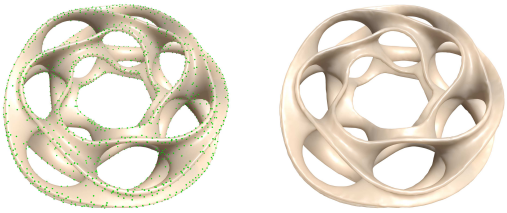
	Ours	LS-mesh
Average PSNR (dB)	32.74	29.27
Average evaluation time (ms)	28.54	1254.20
Average fitting time (s)	386.60	123.31

Fig. 9: Comparison of biharmonic B-splines and LS-meshes on representing time-varying data. Columns 1, 2 and 4 show the original data, and the reconstructed data by biharmonic B-splines and LS-meshes, respectively. Columns 3 and 5 visualize the normalized fitting error, where warm colors and cold colors indicate large and small errors respectively. This figure shows 5 frames of the 100-frame data. See also the accompanying video.

8.2 Comparison with Conventional Splines

In this section, we thoroughly compare biharmonic B-splines with the conventional splines, such as tensor-product B-splines [28], T-splines [29] and triangular B-splines [30], in terms of knots, basis functions, and domains. Table 2 summarizes the properties of various splines.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2016.2605092, IEEE Transactions on Visualization and Computer Graphics



3,124 (14.3%) knots $\epsilon = 0.044\%$

Fig. 10: Thanks to its parameterization-and-singularity-free property, biharmonic spline can be easily defined on 3D surfaces of complicated topology, which are quite difficult for conventional splines. The percentage shows the knot-vertex ratio.

The knots of the conventional splines must follow the required pattern, e.g., the knots of tensor-product B-splines form a regular grid. Most of the conventional splines support knot insertion and deletion, however, it is technically challenging to find the optimal location of knots. To our knowledge, knot optimization is only available to the univariate B-splines and tensor-product B-splines [20]. In sharp contrast, the knots of biharmonic splines are completely free. As a result, knot operations, such as insertion, deletion and optimization, are straightforward in biharmonic splines. Moreover, thanks to the free knot configuration, biharmonic B-spline has the unique linear combination property.

The basis functions of all types of splines are localized and they form a partition of unity, therefore, they offer intuitive modeling via manipulation of control points. The basis functions of the conventional splines are piecewise (rational) polynomials, which are efficient and numerically stable to compute. In contrast, except for a few simple domains (such as Euclidean planes, spheres, cylinders, and tori), there is lack of analytic solution of biharmonic equation on general manifold. Therefore, the basis functions of biharmonic B-splines do not have close-form formula and we have to seek numerical techniques to evaluate the spline, which is expensive compared with the conventional splines.

Since the conventional splines are all defined on planar domains, one needs the global parameterization for extending them to manifold domains. However, computing a high quality parameterization (i.e., with low shape distortion) is expensive, especially for models with complicated geometry and topology. More importantly, as pointed out in [31], singularities are unavoidable if the domain does not have an affine atlas. In sharp contrast, biharmonic B-spline does not require the parameterization at all and is free of singularities, making it ideal for a host of graphics tasks on manifold. Figure 10 demonstrates biharmonic B-splines on models with complicated geometry and topology, which pose great challenge to conventional splines.

8.3 Comparison with Coordinates

Coordinates are a powerful technique for many graphics applications, such as interpolation and shape editing. Popular examples include mean value coordinates [32], [33], Green coordinates [34], local barycentric coordinates [27], bounded biharmonic weights [35], [26], affine generalized barycentric coordinates [36], and many others. Coordinates have many favorable properties, such as localized, non-negative and partition of unity. Note that all the coordinates are defined in Euclidean spaces and extending them to curved surfaces is expensive due to the frequent computation of geodesic distances [37], [38]. In contrast, biharmonic B-splines are based on Green's functions of bi-Laplacian operator, which can be defined on manifolds. Moreover, it has analytic expression on simple domains, such as spheres. However, due to fewer degrees of freedom in the discrete case, the basis functions of biharmonic B-splines on triangle meshes are not strictly non-negative.

8.4 Relation to Polyharmonic Splines

Polyharmonic splines [39] are popular tools for scattered data interpolation in many dimensions. A polyharmonic spline in $\mathbb{R}^k$ is a function of the form

$$f(\mathbf{x}) = \sum_{j=1}^{N} \lambda_j \phi_k(\|\mathbf{x} - \mathbf{x}_j\|) + p(\mathbf{x}),$$

where x_j are centers that the interpolated spline shall pass, λ_j are the weights of the basis functions, and p is a low-degree polynomial satisfying the vanishing moment condition $\sum_{j=1}^{N} \lambda_j p(x_j) = 0$. The basis functions are radial basis functions of the form:

$$\phi_k(r) = \begin{cases} r^k & \text{for } k \text{ odd}, \\ r^k \ln(r) & \text{for } k \text{ even}. \end{cases}$$

Although the proposed implicit representation of biharmonic B-splines has a similar form as the polyharmonic splines, the two types of splines differ in two aspects: First, the basis functions (i.e., radial basis functions) of the polyharmonic B-splines are global, whereas the basis functions of biharmonic B-splines (i.e., ψ) are localized. Thus, biharmonic B-splines allow local editing using its explicit representation. Second, polyharmonic spline based data interpolation and fitting solves a dense linear system, which is computationally expensive for large scale datasets. In contrast, our interpolation algorithm runs in linear time and has also linear space complexity.

8.5 Explicit vs Implicit

Biharmonic B-splines have equivalent implicit and explicit representations. Each representation has its own merits and applications. As demonstrated above, the proposed implicit representation is superior for construction, evaluation and knot optimization. The explicit representation is desired in the applications which require direct manipulation of control points, such as spline editing shown in Figure 8. Note that the two representations are interchangeable. To recover the control points from the implicit representation, one just solves the linear system (see Equation (8)). As the matrix $\widetilde{N}$ can be pre-factored, solving the system takes only linear time. Therefore, the implicit representation complements to the explicit one and users can freely choose the desired representation for their applications.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2016.2605092, IEEE Transactions on Visualization and Computer Graphics

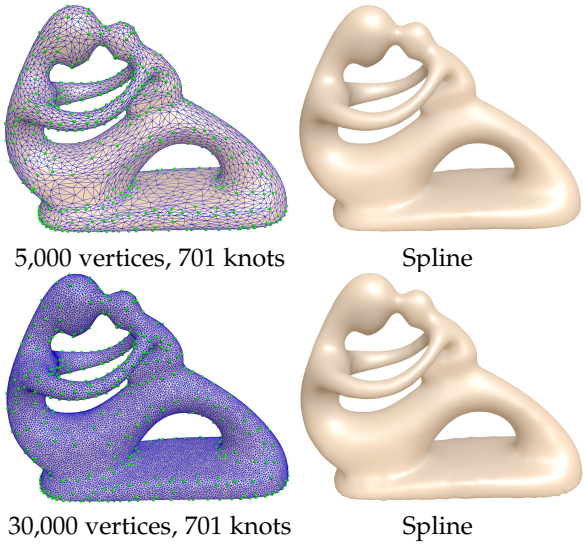


Fig. 11: As Green's functions are intrinsic to the geometry of domain M, the biharmonic B-spline is also intrinsic, hereby insensitive to the resolution and tessellation of M.

9 Conclusion

We presented the implicit representation of the biharmonic B-splines towards significant computational gain and immediate application benefits. This new representation completely avoids the Voronoi tessellation and the discretization of bi-Laplacian operator, and enables the computational utilities on any compact 2-manifold. We developed a new computational framework for constructing biharmonic B-splines on triangular meshes and advanced the new application frontiers of the biharmonic B-splines, including data interpolation, knot optimization, and hierarchical data decomposition. Our framework facilitates optimization-driven knot selection towards better numerical robustness and accuracy. Our results demonstrate that biharmonic B-splines, as a new type of spline functions with much more theoretical and application appeal, afford progressive update of fully irregular knots, are free of singularity, and void the need of explicit parameterization, making it ideal for a host of graphics tasks on manifolds.

Limitations and future work. The proposed biharmonic B-spline framework has several issues, which are worthy of further study.

- Due to the discrete nature of Green's functions on triangle meshes, it is expensive to evaluate biharmonic B-splines. We believe that the multi-scale approach or parallel computation can improve the performance of Algorithm 1.
- Our current implementation did not address the issue of sharp features, which is often required for modeling man-made objects. Sharp features can be introduced into biharmonic splines by restricting the region of the basis functions. For example, the knots on the left side of a feature cannot affect the region on its right side. We will address this problem in future work.
- Although our method can handle open meshes via double covering, the computational cost is also doubled. Directly defining biharmonic B-spline on open meshes is an interesting direction.
- We did not address the computation of derivatives of biharmonic B-spline, thus, our method is limited to applications involving the function value only, such as data interpolation, fitting and denoising, shown in this paper. It is desired to investigate along this direction in the future.

Acknowledgements: We thank the reviewers for their detailed and constructive comments, which help us to improve the quality of the paper significantly. This project was partially supported by MOE2013-T2-2-011, RG23/15, NSFC (Grant No. 61300068, 61190120, 61190121, 61190125, 61532002) and NSF (Grant No. IIS-0949467, IIS-1047715 and IIS-1049448).

References

[1] G. Farin, *Curves and Surfaces for CAGD: A Practical Guide*. Morgan-Kaufmann, 2002.
[2] P. Feng and J. Warren, "Discrete bi-Laplacians and biharmonic B-splines," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 115:1–115:11, 2012.
[3] F. Hou, H. Qin, and A. Hao, "Trivariate biharmonic B-splines," *Computer Graphics Forum*, 2015.
[4] O. Sorkine and D. Cohen-Or, "Least-squares meshes," in *Proceedings of the Shape Modeling International 2004*, ser. SMI '04, 2004, pp. 191–199.
[5] O. Sorkine, D. Cohen-Or, D. Irony, and S. Toledo, "Geometry-aware bases for shape approximation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 2, pp. 171–180, March 2005.
[6] H. Wendland, *Scattered Data Approximation*, ser. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2005.
[7] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *Comp. Graph. Forum*, vol. 28, no. 5, pp. 1383–1392, 2009.
[8] X. Gu and S. Yau, "Global conformal parameterization," in *First Eurographics Symposium on Geometry Processing*, 2003, pp. 127–137.
[9] H. Xu, W. Yu, S. Gu, and X. Li, "Biharmonic volumetric mapping using fundamental solutions," *IEEE Trans. on Visualization and Computer Graphics*, vol. 19, no. 5, pp. 787–798, 2013.
[10] F. R. Chung, *Spectral Graph Theory*. American Mathematical Soc., 1997.
[11] Y. Lipman, R. M. Rustamov, and T. A. Funkhouser, "Biharmonic distance," *ACM Trans. Graph.*, vol. 29, no. 3, pp. 27:1–27:11, 2010.
[12] M. Meyer, M. Desbrun, P. Schröder, and A. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds," in *Visualization and Mathematics III*, ser. Mathematics and Visualization. Springer, 2003, pp. 35–57.
[13] K. Hildebrandt and K. Polthier, "On approximation of the Laplace-Beltrami operator and the Willmore energy of surfaces," *Comput. Graph. Forum*, vol. 30, no. 5, pp. 1513–1520, 2011.
[14] M. Wardetzky, S. Mathur, F. Kälberer, and E. Grinspun, "Discrete Laplace operators: No free lunch," in *Proceedings of the Symposium on Geometry Processing*, 2007, pp. 33–37.
[15] A. Jacobson, E. Tosun, O. Sorkine, and D. Zorin, "Mixed finite elements for variational surface modeling," *Computer Graphics Forum*, vol. 29, no. 5, pp. 1565–1574, 2010.
[16] R. M. Rustamov, "Multiscale biharmonic kernels," *Computer Graphics Forum*, vol. 30, no. 5, pp. 1521–1531, 2011.
[17] J. Xia, Y. He, S. Han, C. Fu, F. Luo, and X. Gu, "Parameterization of star-shaped volumes using Green's functions," in *Proc. of Geometric Modeling and Processing*, 2010, pp. 219–235.
[18] R. Wang, Z. Yang, L. Liu, J. Deng, and F. Chen, "Decoupling noise and features via weighted ℓ_1-analysis compressed sensing," *ACM Trans. Graph.*, vol. 33, no. 2, pp. 18:1–18:12, 2014.
[19] S. Miyata and X. Shen, "Adaptive free-knot splines," *Journal of Computational and Graphical Statistics*, vol. 12, no. 1, pp. 197–213, 2003.
[20] H. Kang, F. Chen, Y. Li, J. Deng, and Z. Yang, "Knot calculation for spline fitting via sparse optimization," *Computer-Aided Design*, vol. 58, pp. 179–188, 2015.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2016.2605092, IEEE Transactions on Visualization and Computer Graphics

[21] C. Brandt, H.-P. Seidel, and K. Hildebrandt, "Optimal spline approximation via ℓ_0-minimization," *Computer Graphics Forum (Proc. EUROGRAPHICS)*, vol. 34, no. 2, pp. 617–626, 2015.

[22] F. John, *Partial Differential Equations*. Springer, 1982.

[23] E. Candès, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 877–905, 2008.

[24] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *In NIPS*. NIPS, 2007, pp. 801–808.

[25] M. Botsch and L. Kobbelt, "An intuitive framework for real-time freeform modeling," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 630–634, Aug. 2004.

[26] Y. Wang, A. Jacobson, J. Barbič, and L. Kavan, "Linear subspace design for real-time shape deformation," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 57:1–57:11, Jul. 2015.

[27] J. Zhang, B. Deng, Z. Liu, G. Patanè, S. Bouaziz, K. Hormann, and L. Liu, "Local barycentric coordinates," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 188:1–188:12, Nov. 2014.

[28] C. de Boor, *A Practical Guide to Spline*. Springer, 1978.

[29] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri, "T-splines and T-NURCCs," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 477–484, 2003.

[30] C. Dahmen, W. Micchelli, and H.-P. Seidel, "Blossoming begets B-spline bases built better by B-patches," *Mathematics of Computation*, vol. 59, pp. 97–115, 1992.

[31] X. Gu, Y. He, and H. Qin, "Manifold splines," *Graphical Models*, vol. 68, no. 3, pp. 237–254, 2006.

[32] K. Hormann and M. S. Floater, "Mean value coordinates for arbitrary planar polygons," *ACM Trans. Graph.*, vol. 25, no. 4, pp. 1424–1441, 2006.

[33] T. Ju, S. Schaefer, and J. D. Warren, "Mean value coordinates for closed triangular meshes," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 561–566, 2005.

[34] Y. Lipman, D. Levin, and D. Cohen-Or, "Green coordinates," *ACM Trans. Graph.*, vol. 27, no. 3, 2008.

[35] A. Jacobson, I. Baran, J. Popović, and O. Sorkine, "Bounded biharmonic weights for real-time deformation," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 78:1–78:8, Jul. 2011.

[36] S. Waldron, "Affine generalised barycentric coordinates," *Jaen Journal on Approximation*, vol. 3, no. 2, pp. 209–226, 2011.

[37] R. M. Rustamov, "Barycentric coordinates on surfaces," *Comput. Graph. Forum*, vol. 29, no. 5, pp. 1507–1516, 2010.

[38] D. Panozzo, I. Baran, O. Diamanti, and O. Sorkine-Hornung, "Weighted averages on surfaces," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 60:1–60:12, 2013.

[39] N. Aronszajn, T. Creese, and L. Lipkin, *Polyharmonic Functions*. Clarendon Press, Oxford, 1983.

Fei Hou received the Ph.D. degree in computer science from Beihang University, Beijing, China, in 2012. He is currently a Research Fellow at Nanyang Technological University, Singapore. His research interests include geometry processing, image based modeling, and data vectorization and medical image processing etc.

Ying He is currently an associate professor at School of Computer Science and Engineering, Nanyang Technological University, Singapore. He received the BS and MS degrees in electrical engineering from Tsinghua University, China, and the PhD degree in computer science from Stony Brook University, USA. His research interests fall into the general areas of visual computing and he is particularly interested in the problems which require geometric analysis and computation. For more information, visit http://www.ntu.edu.sg/home/yhe/

Hong Qin received the BS and MS degrees in computer science from Peking University, China. He received the PhD degree in computer science from the University of Toronto (UofT) in 1995. He is a full professor of computer science in the Department of Computer Science at State University of New York at Stony Brook (Stony Brook University). During his years at the University of Toronto, he received UofT Open Doctoral Fellowship. He was also a recipient of NSF CAREER Award from the US National Science Foundation (NSF), Honda Initiation Award, and Alfred P. Sloan Research Fellow by the Sloan Foundation. Currently, he serves as an associate editor for The Visual Computer, Graphical Models, and Journal of Computer Science and Technology. His research interests include geometric and solid modeling, graphics, physics-based modeling and simulation, computer aided geometric design, human computer interaction, visualization, and scientific computing. Detailed information about him can be found from his web site: http://www.cs.sunysb.edu/∼qin. He is a senior member of the IEEE and the IEEE Computer Society.

Aimin Hao Ph.D., born in 1968, is professor of Computer School, Beihang University and vice director of State Key Laboratory of Virtual Reality Technology and Systems. His research interests are virtual reality, database application and information system development.