J Inf Process Syst, Vol.10, No.1, pp. 23~35, March 2014
http://dx.doi.org/10.3745/JIPS.2014.10.1.023
pISSN 1976-913X
eISSN 2092-805X

Trivariate B-spline Approximation of Spherical Solid Objects

Junho Kim*, Seung-Hyun Yoon**, and Yunjin Lee***

Abstract—Recently, novel application areas in digital geometry processing, such as simulation, dynamics, and medical surgery simulations, have necessitated the representation of not only the surface data but also the interior volume data of a given 3D object. In this paper, we present an efficient framework for the shape approximations of spherical solid objects based on trivariate B-splines. To do this, we first constructed a smooth correspondence between a given object and a unit solid cube by computing their harmonic mapping. We set the unit solid cube as a rectilinear parametric domain for trivariate B-splines and utilized the mapping to approximate the given object with B-splines in a coarse-to-fine manner. Specifically, our framework provides user-controllability of shape approximations, based on the control of the boundary condition of the harmonic parameterization and the level of B-spline fitting. Experimental results showed that our method is efficient enough to compute trivariate B-splines for several models, each of whose topology is identical to a solid sphere.

Keywords—Trivariate B-spline Approximation, Volume Mesh Parameterization, Topological Sphere Model, Harmonic Mapping

1. INTRODUCTION

B-spline surfaces are a species of algebraic representations of the surfaces of 3D objects. With B-spline surface representations, the surface of an object is approximated with combinations of B-spline basis functions on planar parametric domains. The main advantage of B-spline surfaces is that the shapes can be intuitively manipulated with the control points induced from B-spline basis functions. Moreover, it is easy to perform any mathematical analysis on the surface, as B-spline basis functions are analytic in the parametric domain [1].

Due to the usefulness of B-spline surfaces, there have been a large number of modeling tools that provide functionalities with which a user can model the surface of an object by adjusting the control points of B-spline surfaces. Unfortunately, however, it is challenging for the user to model a highly detailed shape with B-spline surfaces using conventional modeling tools, since it is tedious work to adjust the control points of B-spline surfaces manually until the B-spline surfaces properly approximate the detailed shape.

※ This research is supported in part by the Ministry of Education, Science, and Technology(NRF-2009-0077344, NRF-2013R1A1A2010619, NRF-2012R1A1A1008908) and by the research program of Kookmin University in Korea.
Manuscript received June 21, 2005; accepted August 15, 2005.
Corresponding author: Yunjin Lee (yunjin@ajou.ac.kr)
* School of Computer Science, Kookmin University, Seoul, 136-702, Korea (junho@kookmin.ac.kr)
** Department of Multimedia Engineering, Dongguk University, Seoul,100-715, Korea (shyun@dongguk.edu)
*** Division of Digital Media, Ajou University, Gyeonggi-do,443-749, Korea (yunjin@ajou.ac.kr)

Recent advances in 3D scanning technologies provide another method of modeling, in such a way that a 3D scanner can capture the depth images of a real-world object from different views, where the shape of the object is obtainable from the registrations of the depth images. With a 3D scanner, the raw modeling data is a polygonal soup or a two-manifold polygonal mesh whose resolution is proportional to the resolution of the 3D scanner. As a result, one of the main advantages of modeling with the 3D scanner is that we can easily model a highly detailed object in the real world [2]. However, the drawback in this case is that performing a mathematical analysis or numerical computation on the surface of the object represented with a polygonal mesh can be challenging.

In recent decades, much research has been done on digital geometry processing to construct a bridge between polygonal meshes and algebraic surfaces [3]. In particular, surface mesh parameterization plays a key role in digital geometry processing, which formulates a smooth mapping between a polygonal mesh and a parametric domain [4,5]. With this type of mapping, we can embed a highly detailed polygonal mesh into a parametric domain and construct a B-spline surface that fits the geometric details captured with the mesh in this domain. This approach has been successfully applied to rapid prototyping in which only the surface data of a given 3D object is required. The recent demand for digital geometry processing comes from novel application areas, such as simulation, dynamics, and medical surgery simulations, where a representation of the interior volume data of a given 3D object is crucial. Moreover, in these applications, straightforward mathematical analyses or numerical computations of the volumetric interior parts of an object are necessary. For these reasons, it is quite necessary to handle a 3D volumetric object with the appropriate parameterization and trivariate B-spline approximation of the volume data. Although much research has been done on surface data [3,4], research on digital geometric processing of the interior volume data of an object is still in its initial state [6,7,8,9].

In this paper, we present an efficient framework for the shape approximation of a spherical solid object based on trivariate B-splines. To do this, we first constructed a one-to-one correspondence between a given object and a unit solid cube by computing their harmonic mapping. We set the unit solid cube as a rectilinear parametric domain for trivariate B-splines and utilized mapping to approximate the given object with B-splines in a coarse-to-fine manner. Specifically, our framework provides user-controllability of the shape approximations based on controls of the boundary condition of the harmonic parameterization and the level of B-spline fitting.

2. RELATED WORK

2.1 Mesh Parameterization

Mesh parameterization transforms a given mesh into a simple parametric domain to find a one-to-one correspondence between the original shape and the transformed shape in the parametric domain [4]. Because it is possible to define a piecewise smooth function on the entire shape via the parametric domain, mesh parameterization has been successfully utilized on various shape processing areas, such as texture mapping, shape deformation, remeshing, compression, and other areas [3,4].

In this section, we recapitulate the major research results on surface mesh parameterization methods and review recent research on volume parameterization. For much broader reviews of mesh parameterization, we refer the reader to an excellent survey paper [4].

Pinkall and Polthier [10] provided a discretized Laplace-Beltrami operator for surface meshes as the fundamental operator to diffuse a scalar field over the given surface mesh. Floater [11] presented a surface mesh parameterization method based on the convex combinations in which the parameterization is guaranteed to be one-to-one. Lévy et al. [12] and Desbrun et al. [13] proposed free-boundary parameterization methods, in which it is necessary to fix a specific region to the parametric domain using the Cauchy-Riemann equation or the Dirichlet energy minimization techniques.

Gu and Yau [14] proposed a novel surface parameterization method for topologically complicated objects. They found harmonic mappings for each homology basis induced from the handles in a high-genus object and calculated a global parameterization result through a linear combination of harmonic mappings. Recent methods based on curvature flows [15,16,17] provide global parameterization with singularity controls. In [15,16,17], the user can freely design the places of singularities in the parametric domain, as local metrics on a surface can be changed with respect to the curvature flows. However, these methods are highly non-linear, and it is still an open problem to extend them to volumetric meshes.

In contrast to surface mesh parameterization, volume mesh parameterization was recently highlighted because efficient handling of the insides of an object can be utilized in several novel application areas, including in medical simulations and dynamics.

Li et al. [6] calculated harmonic volume mapping with Green's function. Although they provided a fundamental solution for volume parameterization, the linear system for calculating the volume mapping is quite complicated on account of the global properties of Green's function. Xia et al. [8] showed that there is a unique critical point for Green's function on a star-shaped volume. They also provided a direct product volumetric parameterization method for arbitrary topological objects [9]. In theory, the cubic volume parameterization presented in this paper is a special case of [9] and is the direct-product of three line segments.

2.2 Volume Spline Approximation

Splines provide a compact representation of the geometry of an object with combinations of spline basis functions. As spline representation offers several advantages, including local controllability of the geometry and continuity of the shape approximation, it is widely used in industrial design areas, such as CAD, CAM, and CAE [1].

Volume splines have been utilized with simulations, such as the aerodynamic optimization of an industrial design [18]. Most previous approaches are limited to industrial products that were previously modeled with the B-splines, such as aircraft and cars. Recently, a few studies in the computer graphics community have focused on the approximations of arbitrary scanned 3D geometries with volumetric splines.

Martin et al. [7] proposed a method for approximating spherical solid objects using trivariate B-spline and latitude-like harmonic fields. They computed solid spherical parameterization based on the latitudes induced from the harmonic field generated by two user-specified poles on the surface part of a given spherical object. Unfortunately, the process of B-spline

approximation can be numerically unstable, as the parameter domain of the B-splines is not rectilinear. This is because their approach uses a latitude-like circular parametric domain, which causes the singular points in the center of cross-sections.

Recent work on volume parameterization can be utilized for volumetric spline approximations. However, it is difficult to directly apply the results of [6,8,9] to the problem of volumetric spline approximation, as several issues remain in the construction of B-splines for complicated boundary structures. Notice that it is still a non-trivial and challenging problem to consider the smooth continuities of trivariate B-splines where several volumetric cubes meet. Compared to [7], our work utilizes a unit solid cube as the simplest rectilinear parameter domain for spherical objects and it provides a stable means of B-spline approximation with user-controllability.

3. OVERVIEW

In this paper, we present a simple and effective representation of a spherical solid object with trivariate B-splines. Our algorithm consists of the two major steps of cubic volume parameterization (Sec. 4) and scattered data interpolation in the cubic parametric domain (Sec. 5), as shown in Fig. 1.

The goal of cubic volume parameterization is to obtain a smooth harmonic function ϕ: $S \rightarrow C$ between the input tetrahedral mesh S and the unit solid cubic domain C. Once we obtain the parameterization ϕ, we can treat any set of samplings about the object S as scattered data represented on the unit cubic parametric domain C , in such a way that any point p in the object S has its parametric representation $\phi(p)$ in C. This property makes it possible to convert the problem of the trivariate B-spline approximation of a 3D object into one of 3D scattered data interpolation in the unit solid cubic domain. Although we can adopt any B-spline approximation method into our framework, as our parametric domain C is rectilinear, we utilized the multilevel B-spline method [19] in order to approximate a 3D object in a coarse-to-fine manner.

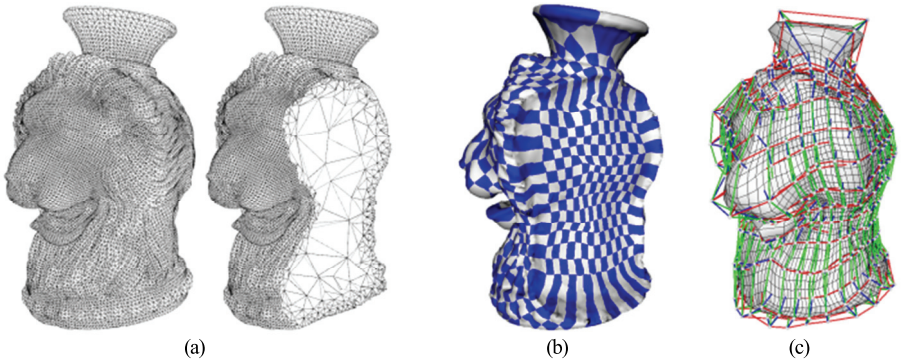


Fig. 1. Overall process: (a) input volume mesh, (b) cubic volume parameterization, (c) B-spline approximation of (a) using (b)

4. CUBIC VOLUME PARAMETERIZATION

We assume that the input solid object is represented with a tetrahedral volume mesh whose topology is identical to a solid sphere. If the user provides a spherical surface mesh with our framework, then we simply convert the surface mesh into a volume mesh with Delaunay tetrahedralization [20,21].

There are infinitely many mappings from volume mesh S to the cubic solid domain C. To obtain a plausible mapping ϕ: $S \rightarrow C$ from all possible mappings, we computed the harmonic volume parameterization with user-specified boundary constraints. The harmonic parameterization guarantees smoothness for the mapping ϕ, and the boundary conditions provide the user with an interactive design method for the mapping ϕ.

Harmonic mapping is a critical point of the harmonic energy that satisfies the following linear equation for each vertex v_i in a mesh [10,22]:

$$\Delta\phi(v_i) = \sum_{v_j \in ngb(v_i)} w_{ij}\left(\phi(v_j) - \phi(v_i)\right) = 0. \tag{1}$$

Here, Δ is known as the Laplace-Beltrami operator, $ngb(v_i)$ is a set of neighbor vertices of v_i, and w_{ij} is the coefficient of an edge e_{ij} reflecting the intrinsic property between two vertices v_i and v_j.

The user-specified constraints of the harmonic map ϕ can be easily included in Eq. (1) by fixing the parametric values of the constraint vertices. The distortion of a harmonic map ϕ depends on its boundary condition [4]. In practice, it is convenient that the system supports the functionality for the user to control the mapping with a few interactions.

In our system, we have provided a technique for the design of the entire harmonic mapping ϕ with eight user-specified feature points on the surface part of S, as follows: first, to obtain the boundary condition of ϕ, we computed a smooth mapping $\partial\phi : \partial S \rightarrow \partial C$, between the surface part of S (hereafter, denoted as ∂S) and the surface part of C (hereafter, denoted as ∂C) using surface harmonic mapping with the user-specified point constraints. Next, all other parts of the map ϕ are computed by volume harmonic mapping, while trying to preserving the boundary condition $\partial\phi$.

4.1 Interactive Design of the Boundary Condition

Based on the user constraints, we enforce the map $\partial\phi$ to partition ∂S smoothly into six rectangular regions, in such a way that the connections of the regions are topologically identical to those of six rectangles on ∂C, as shown in Fig. 2. Initially, the user specifies the eight feature points on ∂S as the constraints, each of which corresponds to a corner on ∂C (see Fig. 2(a)). Next, we partition ∂S by computing the twelve shortest paths between the pairs of the feature points, each of which corresponds to an edge of ∂C (see Fig. 2(b)). To obtain a smooth boundary condition, we adopt patch-wise transitions of the surface harmonic map for the globally smooth surface parameterization process [23,24] (see Fig. 2(c)). Here, the influence of the mapping in one patch can smoothly transit to its adjacent patches via a transition function on the border of two patches. Intuitively, based on the transition function, (1) can be considered as a vector equation, which computes the net flux among the outgoing vectors of a vertex starting from the vertex and ending with its one-ring neighbor vertices. To compute $\partial\phi$ with surface harmonic mapping, we used a surface Laplacian-Beltrami operator [10] whose edge weight is defined as

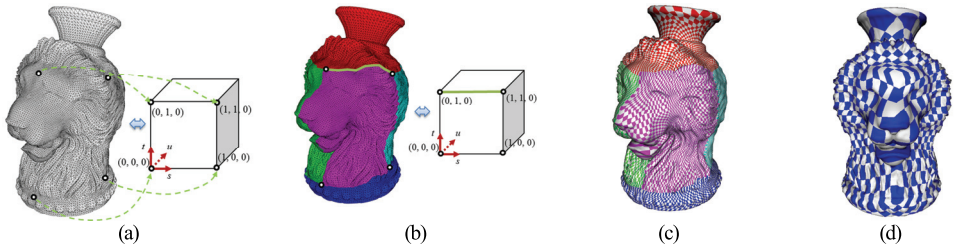


Fig. 2. Computing a smooth harmonic map $\partial\phi : \partial S \rightarrow \partial C$: (a) eight user-specified feature points, (b) initial partition, (c) patch-wise surface harmonic maps based on the initial partition, and (d) globally smooth harmonic map $\partial\phi$

$w_{ij} = \sum_{t \in N(i,j)} \cot \alpha_{ij}^{t}$, where $N(i, j)$ represents a set of triangles on ∂S sharing the edge e_{ij} and α^{t}_{ij} denotes the angle of the opposite corner against the edge e_{ij} in the triangle t. Given that the concept of the net flux vector applies regardless of whether the vertices involved in Eq. (1) exist in the same patch, we were able to obtain a smooth boundary condition with only the eight user-specified feature points.

Fig. 3 shows examples of the interactive design of the boundary conditions. As shown in Fig. 3, the user can simply design the boundary conditions for the cubic volume parameterizations by changing the positions of the eight feature points on the given objects.

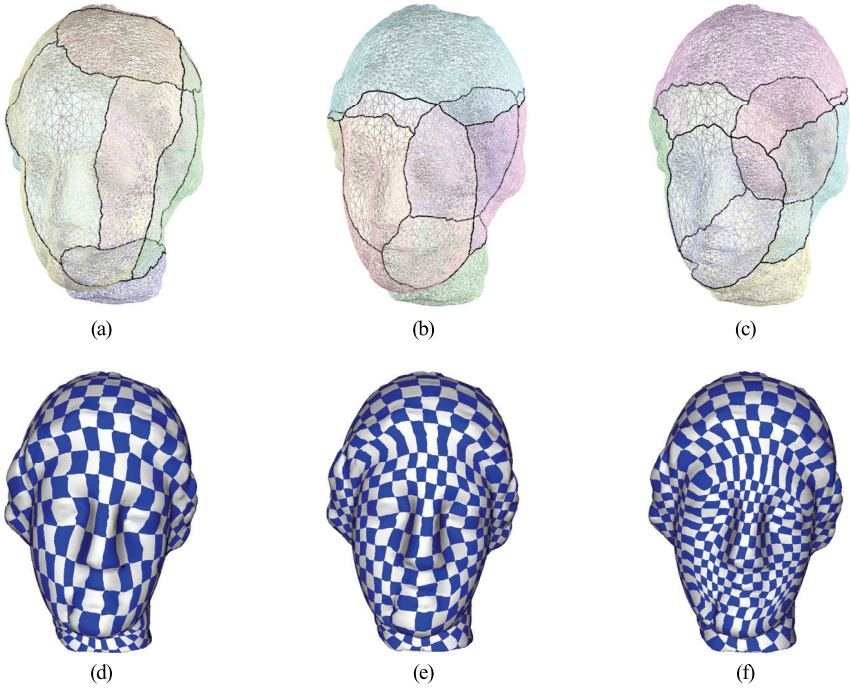


Fig. 3. Interactive design of the boundary condition: for the same objects, different boundary conditions of ϕ can be interactively computed based on different user-specified feature points

4.2 Cubic Volume Parameterization with Harmonic Mapping

Based on the boundary condition $\partial\phi$, we are ready to compute the cubic volume parameterization ϕ through which we can obtain the parameter values of the inside vertices with volumetric harmonic mapping. Specifically, each vertex inside the volume mesh must satisfy Eq. (1) with the volumetric Laplace-Beltrami operator [22], whose edge weight is $w_{ij} = \sum_{t \in N(i,j)} l_{ij} \cot \alpha_{ij}^t$, where t is a tetrahedron sharing the edge e_{ij}, l_{ij} is the length of the edge e_{ij}, and α_{ij}^t is the angle of the opposite wedge against the edge e_{ij} in the tetrahedron t.

Fig. 4 shows the cubic volume parameterization results of various volume meshes. The left side of Fig. 4 visualizes the boundary condition with texture mapping for each 3D model. The right side of Fig. 4 shows part of a volume mesh corresponding to a certain area of the parametric domain by mapping a tile texture to achieve effective visualization.

Fig. 4. Cubic volume parameterization results: in (a) and (b), the figures on the left show the boundary condition $\partial\phi$ while those on the right depict the result of cubic volume parameterization ϕ by visualizing the part corresponding to $\{(s, t, u) \mid 0 \leq t, u \leq 1 \text{ and } 0 \leq s \leq 0.8\}$

5. TRIVARIATE B-SPLINE APPROXIMATION

We reconstructed a trivariate B-spline approximation of a 3D model by using the parameterization result of a volume mesh S and the 3D positions of the vertices in S. At this point, we considered each parameter value $\phi(v_i)$ of a vertex v_i as a sampling point in the cubic solid domain C and the 3D position of v_i in S as the data defined in the sampling point. Then, we were able to consider the 3D positions of all vertices in S as the scattered data defined in C. As trivariate B-splines can be naturally defined in the cubic solid domain C, we were able to approximate the solid shape of a 3D model S by computing scattered data interpolations with B-spline functions for each coordinate of the 3D positions.

Let (x_i, y_i, z_i) and (s_i, t_i, u_i) be the 3D position of a vertex v_i and its parameter value $\phi(v_i)$, respectively. Without a loss of generality, we presented the approximations of the coordinate values x_i with scattered data interpolations. The other coordinate values y_i and z_i are utilized as the scattered data in the same manner. Using a set of 3D scattered points (s_i, t_i, u_i) in C over R^3 with scalar values x_i as data, where $0 \leq s_i, t_i, u_i \leq 1$, we constructed a $(l_k + 3) \times (m_k + 3) \times (n_k + 3)$ control lattice over the domain depicted in Fig. 5(a), where l_k, m_k, n_k represents the number of

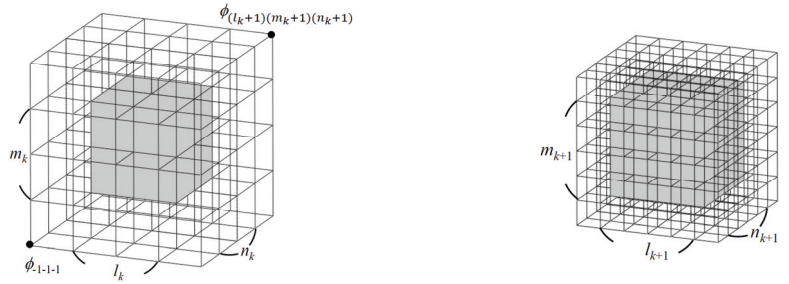


Fig. 5. Refinement of the control lattice: (a) 5×5×5 control lattice (l_k= m_k = n_k =2), (b) 7×7×7 control lattice (l_{k+1}= m_{k+1} = n_{k+1} =4)

knot spans for each axis at level k, respectively. Let $\phi_{\alpha\beta\gamma}$ be the value of the αβγ-th control point on the lattice located at (α, β, γ). The approximation function $f(s, t, u)$ is formulated by a uniform trivariate cubic B-spline function over R^3 as (2).

$$f(s,t,u)=\sum_{p=0}^{3}\sum_{q=0}^{3}\sum_{r=0}^{3}B_p(\hat{s})B_q(\hat{t})B_r(\hat{u})\phi_{\alpha\beta\gamma}, \tag{2}$$

where $\alpha=\lfloor sl\rfloor-1+p$, $\beta=\lfloor tm\rfloor-1+q$, $\gamma=\lfloor un\rfloor-1+r$, $\hat{s}=sl-\lfloor sl\rfloor$, $\hat{t}=tm-\lfloor tm\rfloor$, and $\hat{u}=un-\lfloor un\rfloor$. B_p, B_q, and B_r are the uniform cubic B-spline basis functions defined as $B_0(\lambda)=(1-\lambda)^3/6$, $B_1(\lambda)=(3\lambda^3-6\lambda^2+4)/6$, $B_2(\lambda)=(-3\lambda^3+3\lambda^2+3\lambda+1)/6$, and $B_3(\lambda)=\lambda^3/6$, where $0\le\lambda\le1$.

The resolution of a control lattice strongly influences the approximation quality of S with B-splines. As depicted in Fig. 6(a)-(c), a sparse control lattice generates an under-fitted shape that captures the overall characteristics of S well, while a dense one generates an over-fitted shape that likewise captures the details of S. In order to capture the overall shape, as well as the details of S, we used the multilevel B-spline approximation method [19] in which a coarse-to-fine hierarchy of control lattices is used to generate a desired approximation function.

From the control lattice at the coarsest level, the control lattice at level $(k + 1)$ is generated by the refinement of the control lattice at level k, as shown in Fig. 5(b). At each level k, we

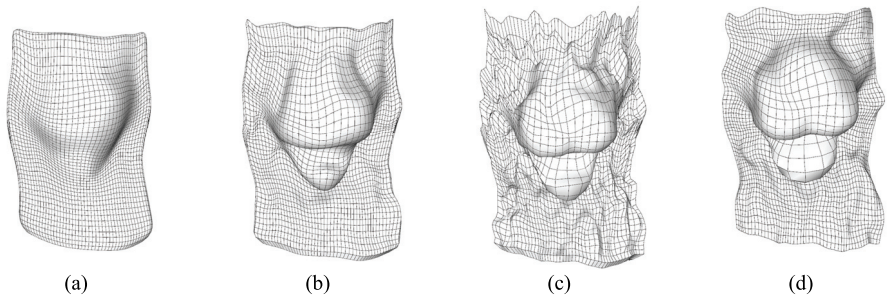


Fig. 6. Approximation results using different resolutions of control lattices: (a) l = m = n = 8, (b) l = m = n = 16, (c) l = m = n = 32; (d) the coarsest resolution is l = m = n = 1 and the finest resolution is l = m = n = 32

computed the residuals $e^k{}_i$ between the approximation values from the k-th level function $f^k(s_i, t_i, u_i)$ and the scattered data x_i. At the next level, the B-spline function with refined control lattice approximated the scattered data points (s_i, t_i, u_i) with the residuals $e^k{}_i$. Note that the refinement process is usually repeated until $e^k{}_i$ is smaller than a user-defined threshold or until k reaches the user-defined finest level. The desired approximation function was obtained by adding up the sequence of B-spline functions generated from different resolutions of control lattices. Fig. 6(d) shows the approximated shape with multilevel B-splines that captures the overall shape, as well as the details of a given object.

6. EXPERIMENTAL RESULTS

Fig. 8 shows several results of a trivariate B-spline reconstruction of a 3D object for the volume meshes in Fig. 7. In the experiments that we conducted, we used a resolution of 2×2×2 for the coarsest control lattice and a resolution of 64×64×64 for the finest control lattice. As shown in Fig. 8, our method smoothly approximates the surface part, as well as the interior volume of a given 3D model, with the B-spline functions.

Fig. 9 shows an example of hierarchical volume approximation with our method. As shown in Fig. 9, we can control the smoothness of the volume approximation simply by changing the number of levels used in the multilevel B-spline approximation process.

Fig. 10 shows an example of volume deformation. Once a 3D object is approximated with the trivariate B-spline functions using our method, we can simply deform the volume of the object by moving several control points of the B-splines. Unlike FFD [25], whose control points are

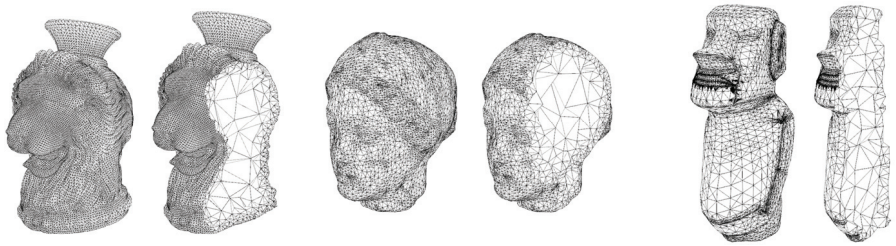


Fig. 7. Input volume meshes: (a) Lion vase model, (b) Igea model, (c) Easter Island model

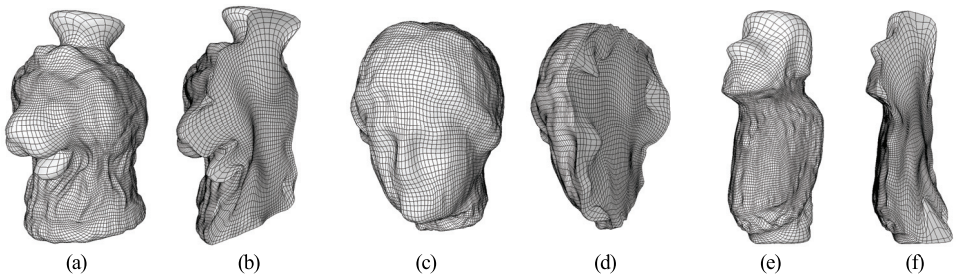


Fig. 8. Reconstruction by trivariate B-spline approximation: the coarsest resolution is $l = m = n = 2$ and the finest resolution is $l = m = n = 64$

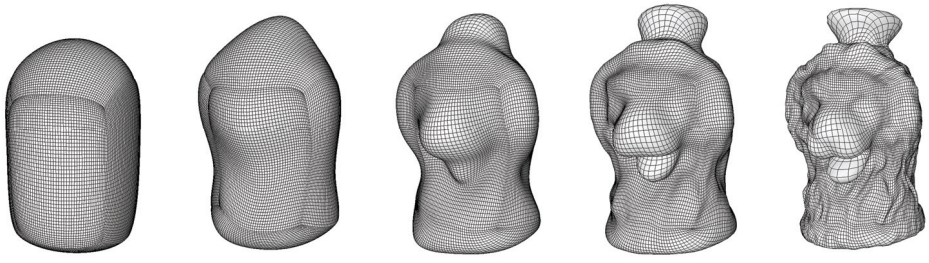


Fig. 9. Level-of-detail control using trivariate B-spline approximation

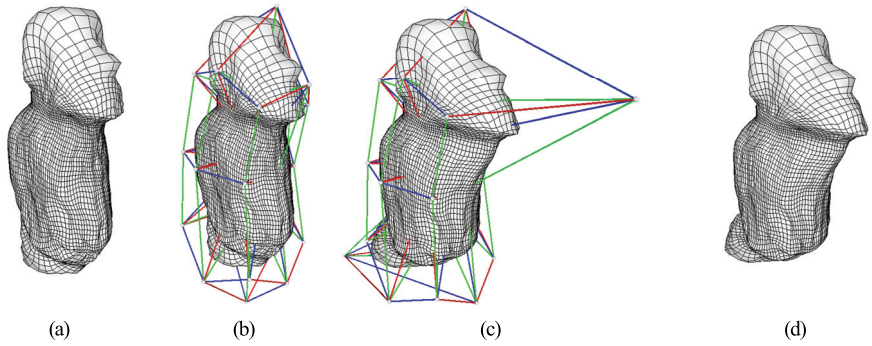


Fig. 10. Volume deformation with an adjustment of the B-spline control points

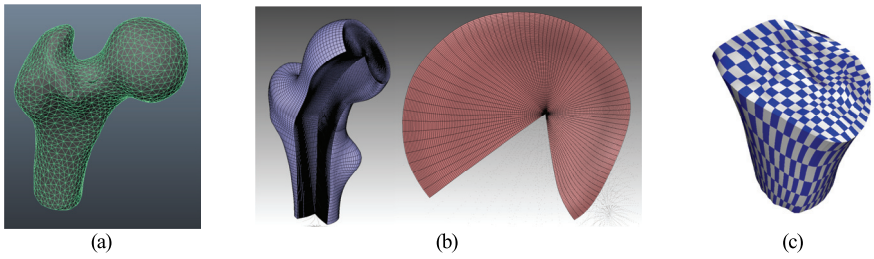


Fig. 11. Femur model and its cross-section of trivariate B-spline approximations: (a) Input model, (b) Martin et al. [7], (c) Ours

rectilinearly defined in the 3D spatial space, our control points are rectilinearly defined in the 3D parametric space, where a 3D volumetric object is mapped. As a result, the layout of our control points resembles the object, and the deformation is quite intuitive, as shown in Fig. 10.

In Fig. 11, we compared the quality of B-spline approximations of a femur model between Martin et al.'s [7] and ours, in terms of volume degeneracies and volume distortions. Since a B-spline function f is a continuous representation, we discretized f with a large number of finite elements in order to measure the degeneracies and the distortions of f, as explained below.

1. We sampled the parametric domain $C = \{(s, t, u) \mid 0 \le s, t, u \le 1\}$ in a uniform sampling rate (e.g., 20^3 or 30^3) and obtained a hexahedral volume mesh, each of whose cell is a cube.
2. Each cube was subdivided into 24 tetrahedra with a mid-point algorithm in a way where we inserted the midpoints at the center of a cell, edges, and faces.
3. We used the tetrahedra as the finite elements for measuring the degeneracies and the distortions of the B-spline approximation f.

Table 1 is a report on the comparison statistics about the number of degenerate finite elements, volumetric distortions, and L_∞ stretches. For the volume distortion, we measured the ratio of the volume changes between the finite elements t in C and their corresponding finite elements $f(t)$ in R^3 with (3). For the L_∞ stretches, we took the maximum value among all of the stretch values, each of which is defined from the L_∞ stretch of the affine mapping between t and $f(t)$.

$$\sum_t \log\left(\max\left(\frac{vol(t)}{vol(f(t))}, \frac{vol(f(t))}{vol(t)}\right)\right), \tag{3}$$

where $vol(\cdot)$ is a normalized volume of a finite element in C or R^3. Notice that $\max(\cdot,\cdot)$ measures the distortion, due to volumetric shrinking or expansion.

Experiments show that our approach is two times better than [7], in terms of the volume distortion and L_∞ stretch. Moreover, our B-splines have no degeneracy, in contrast to [7], which has large numbers of degenerated volumes. This is because [7] uses a latitude-like circular parametric domain, which causes the singular points in the center of cross-sections (see Fig. 11(b)). Notice that the volume degeneracy causes critical problems in the applications, such as numerical simulations.

Table 1. Comparison statistics with Femur model in Fig. 11

	Sampling rate.	Martin et al. [7]	Ours
# degenerated finite elements	20×20×20	3,650	0
	30×30×30	9,141	0
volume distortion	20×20×20	242,886	142,985
	30×30×30	750,032	487,064
L∞ stretches	20×20×20	71.45	38.13
	30×30×30	77.13	38.49

7. CONCLUSION

In this paper, we proposed a novel volumetric shape approximation framework with B-splines. Our framework provides an interactive method to parameterize an input volume mesh smoothly into a cubic solid domain where the trivariate B-splines are naturally defined. With harmonic mapping between the input volume mesh and the cubic solid domain, we converted the problem of the trivariate B-spline approximation of a 3D object into an issue of scattered data interpolation with multilevel B-splines. From the trivariate B-spline approximation of an object,

we showed that several useful functionalities can be simply defined for the volumetric shape of the object, such as hierarchical volumetric shape approximation and deformation.

The limitation of our method is that we can only handle a topologically spherical solid object. Clearly, necessary future work would involve extending our framework so that it can handle arbitrarily topological solid objects. In addition, it has been proven that harmonic mapping is not guaranteed to be objective in the 3D volumetric parameterization [8]. This may cause folder-overs in our trivariate B-spline functions. Finally, minimizing the amount of distortion by partitioning the input volume into several volume areas by eigen analysis or Morse-Smale complex computation is a potential research topic.

REFERENCES

[1] G. Farin, Curves and Surfaces for CAGD, 5 ed., Academic Press, 2002.

[2] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk, "The digital Michelangelo project: 3d scanning of large statues," Proc. ACM SIGGRAPH 2000, pp.131-144, 2000.

[3] P. Schröder and W. Sweldens, "Digital geometry processing," SIGGRAPH 2001 Course Notes, 2001.

[4] M.S. Floater and K. Hormann, "Surface parameterization: a tutorial and survey," Proc. of Advances in Multiresolution for Geometric Modelling, pp.157-186, Springer, 2005.

[5] A. Sheffer, E. Praun, and K. Rose, "Mesh parameterization methods and their applications," Foundations and Trends in Computer Graphics and Vision, vol.2, no.2, 2006.

[6] X. Li, X. Guo, H. Wang, Y. He, X. Gu, and H. Qin, "Harmonic volumetric mapping for solid modeling applications," Proc. of ACM Symposium on Solid and Physical Modeling, pp.109-120, 2007.

[7] T. Martin, E. Cohen, and M. Kirby, "Volumetric parameterization and trivariate B-spline fitting using harmonic functions," Proc. of ACM Symposium on Solid and Physical Modeling, pp.269-280, 2008.

[8] J. Xia, Y. He, S. Han, C.W. Fu, and X. Gu, "Parameterization of star-shaped volumes using green's functions," Proc. Geometric Modeling and Processing, pp.219-235, 2010.

[9] J. Xia, Y. He, X. Yin, S. Han, and X. Gu, "Direct-product volumetric parameterization of handlebodies via harmonic fields," Proc. Shape Modeling International, pp.3-12, 2010.

[10] U. Pinkall and K. Polthier, "Computing discrete minimal surfaces and their conjugates," Experimental Mathematics, vol.2, pp.15-36, 1993.

[11] M.S. Floater, "Parametrization and smooth approximation of surface triangulations," Computer Aided Geometric Design, vol.14, no.3, pp.231-250, 1997.

[12] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, "Least squares conformal maps for automatic texture atlas generation," ACM Trans. Graphics (Proc. ACM SIGGRAPH 2002), pp.362-371, 2002.

[13] M. Desbrun, M. Meyer, and P. Alliez, "Intrinsic parameterizations of surface meshes," Computer Graphics Forum, vol.21, no.3, pp.209-218, 2002.

[14] X. Gu and S.T. Yau, "Global conformal parameterization," Proc. Symposium on Geometry Processing, pp.127-137, 2003.

[15] L. Kharevych, B. Springborn, and P. Schröder, "Discrete conformal mappings via circle patterns," ACM Trans. Graphics, vol.25, pp.412-438, 2006.

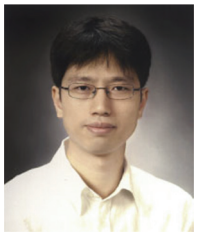
[16] M. Jin, J. Kim, F. Luo, and X. Gu, "Discrete surface Ricci flow," IEEE Trans. Visualization and Computer Graphics, vol.14, no.5, pp.1030-1043, 2008.

[17] B. Springborn, P. Schröder, and U. Pinkall, "Conformal equivalence of triangle meshes," ACM Trans. Graphics (Proc. ACM SIGGRAPH 2008), pp.77:1-77:11, 2008.

[18] J.E. Hicken and D.W. Zingg, "Aerodynamic optimization algorithm with integrated geometry parameterization and mesh movement," AIAA Journal, vol.48, pp.400-413, 2010.

[19] S. Lee, G. Wolberg, and S.Y. Shin, "Scattered data interpolation with multilevel B-splines," IEEE Trans. Visualization and Computer Graphics, vol.3, no.3, pp.228-244, 1997.

[20] H. Si, "TetGen: A Quality Tetrahedral Mesh Generator and a 3D Delaunay Triangulator," http://tetgen.org /
[21] P. Alliez, D. Cohen-Steiner, M. Yvinec, and M. Desbrun, "Variational tetrahedral meshing," ACM Trans. Graphics (Proc. ACMSIGGRAPH 2005), pp.617-625, 2005.
[22] Y. Wang, X. Gu, and S.T. Yau, "Volumetric harmonic map," Communications in Information and Systems, vol.3, no.3, pp.191-202, 2004.
[23] A. Khodakovsky, N. Litke, and P. Schröder, "Globally smooth parameterizations with low distortion," ACM Trans. Graphics (Proc. ACM SIGGRAPH 2003), pp.350-357, 2003.
[24] S. Dong, P.T. Bremer, M. Garland, V. Pascucci, and J.C. Hart, "Spectral surface quadrangulation," ACM Trans. Graphics (Proc. ACM SIGGRAPH 2006), pp.1057-1066, 2006.
[25] T.W. Sederberg, "Free-form deformation of solid geometric models," Computer Graphics (Proc. ACM SIGGRAPH 1986), pp.151-160, 1986.



Junho Kim

He received the BS, MS, and PhD degrees in computer science and engineering from the Pohang University of Science and Technology (POSTECH) in 1998, 2000, and 2005, respectively. He is currently an assistant professor of the School of Computer Science in Kookmin University. His research interests include computer graphics, computer vision, geometric modeling, and mixed reality.

Seung-Hyun Yoon

He received the BS degree in mathematics from Hanyang University in 2001 and the PhD degree in computer science and engineering from Seoul National University in 2007. He is currently an associate professor of the Department of Multimedia Engineering, Dongguk University. His research interests are in computer graphics and geometric modeling.

Yunjin Lee

She received her BS degree in 1999 and her PhD degree in 2005, all in Computer Science and Engineering from the Pohang University of Science and Technology (POSTECH) in Korea. She is currently an associate professor in the Division of Digital Media at Ajou University. Her research interests include non-photorealistic rendering, 3D mesh processing, and data compression.