

Volumetric Parameterization and Trivariate B-spline Fitting using Harmonic Functions

Tobias Martin*
School of Computing
University of Utah

Elaine Cohen†
School of Computing
University of Utah

Robert M. Kirby‡
School of Computing
University of Utah

Abstract

We present a methodology based on discrete volumetric harmonic functions to parameterize a volumetric model in a way that it can be used to fit a single trivariate B-spline to data so that simulation attributes can also be modeled. The resulting model representation is suitable for isogeometric analysis [Hughes T.J. 2005]. Input data consists of both a closed triangle mesh representing the exterior geometric shape of the object and interior triangle meshes that can represent material attributes or other interior features. The trivariate B-spline geometric and attribute representations are generated from the resulting parameterization, creating trivariate B-spline material property representations over the same parameterization in a way that is related to [Martin and Cohen 2001] but is suitable for application to a much larger family of shapes and attributes. The technique constructs a B-spline representation with guaranteed quality of approximation to the original data. Then we focus attention on a model of simulation interest, a femur, consisting of hard outer cortical bone and inner trabecular bone. The femur is a reasonably complex object to model with a single trivariate B-spline since the shape overhangs make it impossible to model by sweeping planar slices. The representation is used in an elastostatic isogeometric analysis, demonstrating its ability to suitably represent objects for isogeometric analysis.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations G.1.2 [Mathematics of Computing]: Approximation—Spline and piecewise polynomial approximation

Keywords: trivariate b-spline modeling and generation, volumetric parameterization, model acquisition for simulation

1 Introduction

A frequently occurring problem is to convert 3D data, for instance image data acquired through a CT-scan, to a representation on which physical simulation can be applied as well as for shape representation. Grids or meshes, based on primitives like triangles, tetrahedra, quadrilaterals and hexahedra are frequently used representations for both geometry and analysis purposes.

Mesh generation software like [Si 2005] generates an unstructured tetrahedral mesh from given input triangle meshes. Unstructured grids modeling techniques [Hua et al. 2004] improve the model-

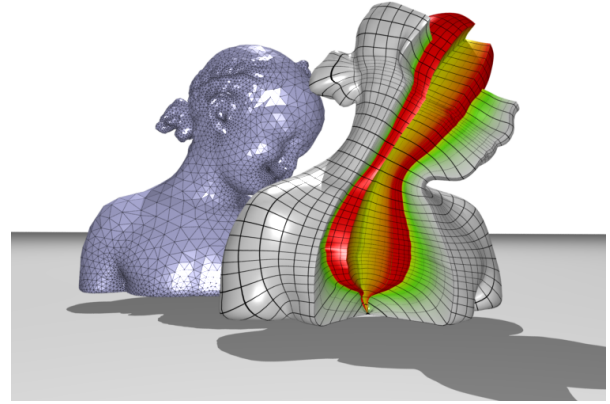


Figure 1: (a) triangle mesh of Bimba statue; (b) corresponding trivariate B-spline, where interior represents material information used in simulation.

ing and rendering of multi-dimensional, physical attributes of volumetric objects. However, unstructured grid techniques have drawbacks and certain types of simulation solvers have a preference for structured grids. Creating a structured quadrilateral surface representation and an integrated structured hexahedral internal volume representation from unstructured data is a problem that has undergone significant research. Though topologically limited, structured grids have advantages- especially with growing mesh sizes. For instance, simulation like linear elasticity, multiresolution algorithms like wavelet decomposition or multiresolution editing can be efficiently applied to them. Such structured hexahedral meshes are highly prized in many types of finite element simulations, and generally still require significant manual interaction.

For smoothly modeling geometry, attributes, and simulations simultaneously, trivariate tensor product B-splines have been proposed in *isogeometric analysis* [Hughes T.J. 2005; Zhou and Lu 2005], that applies the physical analysis directly to the geometry of a B-spline model representation that includes specified attribute data [Martin and Cohen 2001] (such as Lamé parameters used in linear elasticity). The user gets feedback directly as attributes of the B-spline model analysis representation, avoiding both the need to generate a finite element mesh and the need to reverse engineer from the finite element mesh. However, it is necessary to have a representation of the B-spline model suitable for this analysis. Generating a structured hexahedral grid, parameterizing the volume, and generating a suitable trivariate B-spline model from unstructured geometry and attributes is the main focus in this paper.

Our contributions in this work include

1. a framework to model a single trivariate B-spline representation from an exterior boundary, and possibly interior boundaries that have the same genus as the exterior boundary. The boundaries are triangle surfaces, representing geometry or material information, possibly generated from image data.

*e-mail: martin@cs.utah.edu

†e-mail: cohen@cs.utah.edu

‡e-mail: kirby@sci.utah.edu

2. a technique to create a trivariate B-spline that has a consistent parameterization across given isosurfaces.
3. demonstration of our framework on real unstructured data, a femur obtained through a CT-scan and apply stress simulation to it (see Figure 2). A femur consists of a cortical bone, with high densities, and an interior part consisting of a porous, trabecular bone. The transition between cortical and trabecular part is smooth, making trivariate B-splines a candidate to model such a scenario.

After discussing related literature, we review trivariate B-splines and harmonic functions in Section 3. A framework overview is given in Section 4. On a global view, modeling the exterior (Section 5) is the first stage in our framework, followed by modeling the interior in Section 6. A trivariate B-spline is fitted against an intermediate hexahedral mesh in Section 7. In the last sections (8, 9), we discuss extensions to our framework and experiments, respectively.

2 Previous Work

Parameterization is a hard problem for surfaces and even more so for volumes. In addition to use in modeling and remeshing, surface parameterization techniques have a wide variety of applications including texture mapping, detail transfer, fitting and morphing. For a more detailed description, please refer to the surveys [Sheffer et al. 2006; Floater and Hormann 2005]. Surface parameterizing techniques such as [Loop 1994; Grimm and Hughes 1995; Tong et al. 2006] deal with surface related issues and are not designed to be extended to model volumes. For instance, the authors in [Alliez et al. 2003] motivate anisotropic remeshing and align mesh elements using the principal direction of curvature of the respective triangle mesh. Their approach yields a high quality quadrilateral mesh that has no relationship to the interior. If one were to offset the mesh in the normal direction, one would quickly get self intersections among the elements. Even if this can be avoided, eventually the hexahedral elements have degeneracies and eventually touch each other without proper alignment. Requiring parameterization of the interior makes this problem even more difficult since it is prone to self intersecting offsets and has to deal with skewed and twisted parameterizations.

Usually, the above-mentioned techniques involve “patch gluing” where a certain level of smoothness along the patch boundaries is desired. In [Loop 1994], quadratic B-splines are generalized to fit arbitrary meshes creating hybrid triangular and rectangular surface patches. The lack of structure in the irregularities makes it clear that volumetric extensions do not immediately follow. Similarly, manifold splines [Grimm and Hughes 1995] extend B-splines to surfaces of arbitrary topology, by modeling the domain of the surfaces with a manifold whose topology matches that of the polyhedral mesh, then it embeds this domain into 2-space using a basis-function/control-point formulation. The domain of this technique is more complicated than the domain of a standard tensor product surface. As in [Loop 1994], this approach also generates spline patches and glues them together by overlapping them, to get a “match” in the parameterization. The “glue” consists of mathematical operations such as control point constraints. In the case of a volume, patch boundaries are surfaces. Establishing smoothness and continuity between them is very difficult. Converting a triangle mesh into a single trivariate B-spline has two advantages: First, smoothness is preserved throughout the object, which simplifies analysis. Second, modeling is simplified, avoiding any gluing.

The nature of B-spline surfaces or volumes does not naturally lend itself to bifurcations which may exist in the data. Also, objects with higher genus are difficult to model. However, in many applications,

like for instance, in the case of the femur, the object has a “topology” of a cylinder but shows local concavities and overhangs, that are not bifurcations, but cause representational complexity. Solutions to handle these concavities have been proposed for the surface case [Dong et al. 2005] but not for the trivariate case appropriate for tensor product B-spline volumes. Our approach enables the generation of a consistent parameterization and B-spline volume representation for these kinds of geometric objects.

The decomposition of 3D objects into simpler volumetric parts and the description of parts and the relationships between them is a good way of representation. [Binford 1971] proposed the generalized cylinder-based (GC) shape representation which was extended by [Chuang et al. 2004]. The solid of a CG is obtained by sweeping a planar cross-section according to a scaling function along a space curve. Similarly, [Jaillet et al. 1997] outlines a technique to generate B-spline surfaces from a set of planar cross-sections acquired from image data. They allow branches, solve the correspondence problem and skin the frame with B-spline surfaces. The overhang regions using [Jaillet et al. 1997]’s method require gluing of patches, a problem we do not have to deal with in our method. Our technique, however, currently is not suitable for models with branches, but since we deal with nonplanar cross-sections we do not require additional patch gluing for overhangs.

While being able to reconstruct some types of real world objects, the GC approaches and the approach by [Jaillet et al. 1997] are limited, because they require planar cross-sections. Note, representations that rely on planar cross-sections fail for objects with overhangs such as the femur in Figure 2. GC is a subclass of our method since we are able to model objects with overhangs as the femur in Figure 2. Furthermore, skinning introduces oscillations on the B-spline, which is amplified when dealing with volumes, as in our case. And, in our framework, once a certain stage of the process is reached, the rest proceeds automatically without further user input.

Harmonic volumetric mappings between two solid objects with the same topology have been used in a variety of instances. Using related techniques, [Wang et al. 2004] and [Li et al. 2007] are performing a 3D time-variant harmonic deformation from one volume to another volume with the same topology. In a diffeomorphic way, [Wang et al. 2004] applies this method to brain data.

In [Verroust and Lazarus 2000] a method is proposed, to construct skeleton curves from an unorganized collection of scattered points lying on a surface which can have a “tree-like” structure. They calculate a geodesic graph over the point set. Using that graph, they extract level sets, closed and piecewise linear. The centroids of all the level sets form the skeleton. When level sets are not convex the centroid may lie outside the objects. Furthermore, the skeleton may have loops if the centroid of a given level set a lies above the centroid of the level set b lying above a . We require the skeleton to have no loops and to be inside the geometric object. In our method we guarantee that. Similarly, [Lazarus and Verroust 1999] explicitly establishes a scalar function, similar to a harmonic function, over a triangle mesh. By choosing a source vertex, for every vertex on the triangle mesh, shortest distances are calculated which establish a parameterization in one parameter. The skeleton is calculated as in [Verroust and Lazarus 2000] and therefore cannot guarantee whether it lies within the triangle mesh.

3 Preliminaries and Notation

In this work we define volumetric harmonic functions over an input triangular boundary or tetrahedral mesh and generate a volumetric parameterization of the model. Then, a trivariate B-spline is fit to the data with parameters that measure error. The following sections

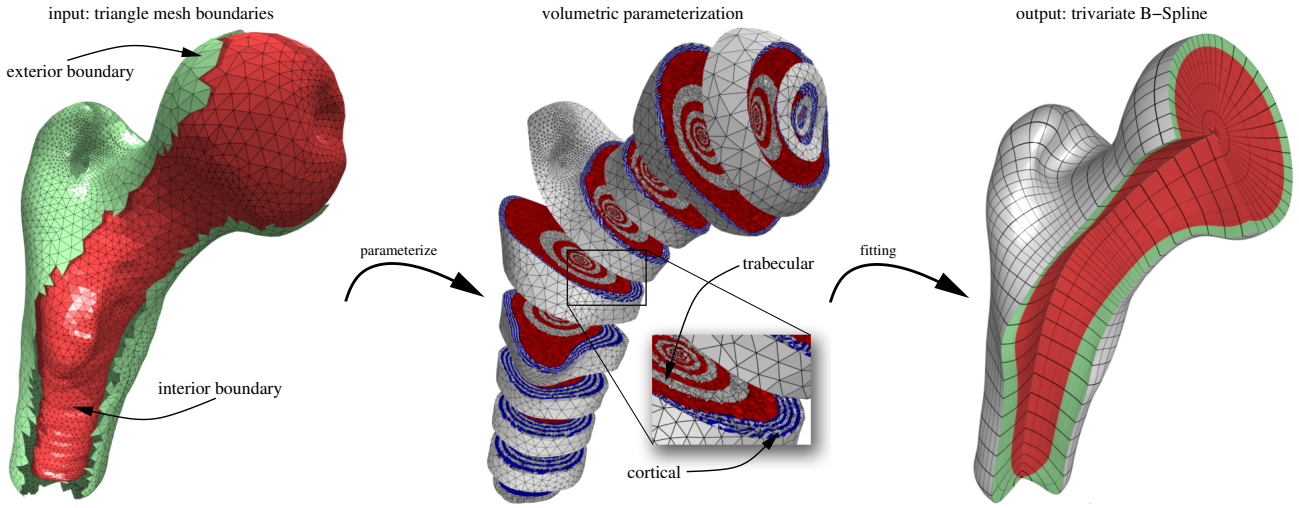


Figure 2: A femur consists of two materials: The outer solid part, or cortical bone, represented by the volume between the input triangle meshes; and the inner soft part, or trabecular bone, represented by the volume of interior triangle mesh (red). These volumes are parameterized (middle) and a single trivariate tensor product B-spline is fitted against it (right), respecting the input triangle meshes in its parameterization. This makes it easier to specify respective material properties. Black isolines represent knotlines in the trivariate parameterization.

briefly recall B-spline definitions and properties of harmonic functions and ways to solve them over a triangle and tetrahedral mesh.

3.1 Tensor-Product B-splines

A B-spline volume, or a trivariate tensor-product B-spline volume is a mapping $V : [0, 1]^3 \rightarrow \mathbb{P}^3$ that can be formulated as

$$V(u, v, w) = \sum_{i_0, i_1, i_2=0}^{n_0, n_1, n_2} P_{i_0, i_1, i_2} B_{i_0, p_0}(u) B_{i_1, p_1}(v) B_{i_2, p_2}(w).$$

where the $P_{i_0, i_1, i_2} \in \mathbb{R}^3$ are the control points of the $(n_0 + 1) \times (n_1 + 1) \times (n_2 + 1)$ control mesh, having basis functions B_{i_j, p_j} (defined in [Cohen et al. 2001]) of degree p_j with knot vectors $T^j = \{t_i^j\}_{i=0}^{n_j + p_j}$ for $j = 0, 1, 2$.

3.2 Discrete Harmonic Functions

Given a domain $\Omega \in \mathbb{R}^n$, where in our case $n = 2$ and $n = 3$, a harmonic function is a function $\mathbf{u} \in C^2(\Omega)$, $\mathbf{u} : \Omega \rightarrow \mathbb{R}$, satisfying Laplace’s equation, that is

$$\nabla^2 \mathbf{u} = 0, \quad (1)$$

$$\text{where } \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}.$$

Harmonic functions satisfy the maximum principle, namely they have no local minima/maxima and can therefore be used as Morse functions [Milnor 1963; Ni et al. 2004]. Also, this property makes them suitable to create a tensor-product style parameterization, as done in [Tong et al. 2006] for surfaces. In this paper harmonic functions are utilized in order to fit a trivariate tensor product B-spline to a tetrahedral mesh generated from a set of triangulated isosurfaces.

We describe a tetrahedral mesh by the tuple $(\mathcal{H}, \mathcal{T}, \mathcal{V}, \mathcal{C})$ over the domain Ω . \mathcal{H} is the set of tetrahedra and \mathcal{T} is the set of faces of the tetrahedra in \mathcal{H} . \mathcal{V} is the set of vertices, $\nu = (x_\nu, y_\nu, z_\nu) \in \mathcal{V} \subset \mathbb{R}^3$ of the tetrahedra in \mathcal{H} , and \mathcal{C} specifies the connectivity

of the mesh (the adjacency of vertices, edges, triangular faces and tetrahedra). Furthermore, \mathcal{T}_B is the subset of \mathcal{T} whose elements are faces of exactly one tetrahedron. The elements of \mathcal{T}_B form the original exterior triangle mesh for the object. $\mathcal{V}_B \subset \mathcal{V}$ is the set of vertices defining the triangles in \mathcal{T}_B .

Solving equations for any but the simplest geometries requires a numerical approximation. We use mean-value coordinates [Floater 2003] to solve Equation 1 on \mathcal{T}_B . Refer to [Ni et al. 2004] which discusses in more detail how to set up the appropriate linear system. The Finite Element Method (FEM) [Hughes 2000] is used to solve Equation 1 on \mathcal{H} . The set \mathcal{V} is decomposed into two disjoint sets, \mathcal{V}_C and \mathcal{V}_I , representing vertices that lie on the Dirichlet boundary (and hence denote positions at which the potential \mathbf{u} is known) and vertices for which the solution is sought, respectively.

Then, in the case of finite elements, solutions are of the form:

$$\mathbf{u}(x, y, z) = \sum_{\nu_k \in \mathcal{V}_I} \hat{\mathbf{u}}_k \phi_k(x, y, z) + \sum_{\nu_k \in \mathcal{V}_C} \hat{\mathbf{u}}_k \phi_k(x, y, z),$$

where the sums denote the weighted degrees of freedom of the unknown vertices, and the Dirichlet boundary condition of the solution, respectively. $\phi_i(x, y, z)$ are the linear hat functions, which are 1 at ν_i and 0 at ν_i ’s adjacent vertices. Using the weak Galerkin formulation [Hughes 2000] yields a linear system of the form $\mathbf{S}\hat{\mathbf{u}} = \vec{f}$, consisting of stiffness matrix \mathbf{S} and a right-hand-side function \vec{f} . Because the stiffness matrix is positive definite [Hughes 2000], the solution of the linear system is amenable to iterative methods such as the preconditioned conjugate gradient method [Axelsson 1994].

Every point inside the tetrahedral mesh volume either lies on the boundary or inside a tetrahedron and the point’s “ $\hat{\mathbf{u}}$ -value” is a linear combination of the vertices of the tetrahedron in which it lies. Given a tetrahedron defined by four vertices ν_{j_i} , $i = 1, 2, 3, 4$ and the corresponding basis functions ϕ_{j_i} , the $\hat{\mathbf{u}}$ -value of a point ν inside a tetrahedron, is the linear combination $\hat{\mathbf{u}}(\nu) = \sum_{i=1}^4 \hat{\mathbf{u}}_{j_i} \phi_{j_i}(\nu)$, where the $\hat{\mathbf{u}}_i$ ’s are the respective harmonic coefficients of the tetrahedron’s defining vertices.

The gradient $\nabla \hat{\mathbf{u}}$ over a tetrahedron is the linear combination

$$\nabla \hat{\mathbf{u}}(x, y, z) = \sum_{i=1}^4 \hat{\mathbf{u}}_{j_i} \nabla \phi_{j_i}(x, y, z),$$

$$\text{where } \nabla \phi_{j_i}(x, y, z) = \left(\frac{\partial \phi_{j_i}(x, y, z)}{\partial \nu_x}, \frac{\partial \phi_{j_i}(x, y, z)}{\partial \nu_y}, \frac{\partial \phi_{j_i}(x, y, z)}{\partial \nu_z} \right).$$

Note, that $\nabla \hat{\mathbf{u}}$ is constant over a tetrahedron and its boundary so it changes piecewise constantly over the tetrahedral mesh. In the following, \mathbf{u}_Ω means that the harmonic function \mathbf{u} is defined over domain Ω , where Ω is \mathcal{H} or \mathcal{T}_B .

4 Framework Overview

This section gives a high level overview of our proposed modeling framework. Our framework takes as input a tetrahedral mesh \mathcal{H} containing, if given, interior triangle meshes such as the trabecular bone triangle mesh illustrated in Figure 2. Given that, the following framework steps describe the generation of the trivariate B-spline.

Step 1 The user makes an initial choice of two critical points. These are used to establish a surface parameterization in two variables defined by orthogonal harmonic functions $\mathbf{u}_{\mathcal{T}_B}$ and $\mathbf{v}_{\mathcal{T}_B}$ (Section 5).

Step 2 Generate a structured quadrilateral mesh using the surface parameterization calculated in the previous step (Section 5.1).

Step 3 In this phase we move to working with the complete tetrahedral mesh. Two harmonic functions are calculated over \mathcal{H} (Section 6):

- $\mathbf{u}_{\mathcal{H}}$ is determined by solving Equation 1 with $\mathbf{u}_{\mathcal{T}_B}$ as the Dirichlet boundary condition.
- \mathbf{w} is a harmonic function orthogonal to $\mathbf{u}_{\mathcal{H}}$, having a harmonic value of 0 on \mathcal{T}_B and 1 on an interior skeleton generated using $\nabla \mathbf{u}_{\mathcal{H}}$. Interior boundaries have a value between 0 and +1.

Step 4 Isoparametric paths with constant \mathbf{u} -parameter value are extracted using $\nabla \mathbf{w}$. They start at vertices defining the quadrilateral mesh from step 2 and end at the skeleton. These paths are used to generate a structured hexahedral mesh which is a remesh of \mathcal{H} (Sections 6.1, 6.2 and 6.3).

Step 5 The trivariate B-spline is generated from the hexahedral mesh generated in step 4, by using an iterative fitting approach which avoids surface undulations in the resulting B-spline (Section 7).

The intermediate structured meshes are constructed so that they faithfully approximate the input data. The resulting B-spline can therefore have a high resolution. Additional post-processing steps include data reduction techniques to reduce complexity and to generate B-spline volumes of different resolutions.

5 Modeling the Exterior

In this section a parameterization \mathfrak{X}_2 in two variables \mathbf{u} and \mathbf{v} defined over \mathcal{T}_B is established. The choice of \mathfrak{X}_2 requires the user to choose two appropriate vertices ν_{min} and ν_{max} from the set of vertices in \mathcal{V}_B . Then, $\nabla^2 \mathbf{u}_{\mathcal{T}_B} = 0$ is solved with $\mathcal{V}_C = \{\nu_{min}, \nu_{max}\}$ as the Dirichlet boundary, where we set $\mathbf{u}_{\mathcal{T}_B}(\nu_{min}) = 0$ and $\mathbf{u}_{\mathcal{T}_B}(\nu_{max}) = 1$.

The choice of these two critical vertices depends on the model and on the simulation. As pointed out by [Dong et al. 2005], critical

vertices affect the quality of the parameterization which in our case also affects the trivariate B-spline we are fitting. Since the user might be aware of which regions require higher fidelity and lower distortion in later simulations, the user can select a pair of critical vertices to yield an appropriate parameterization. Since $\mathbf{u}_{\mathcal{T}_B}$ is defined only on \mathcal{T}_B it can be computed rapidly which allows the user to modify it if unsatisfied with the result.

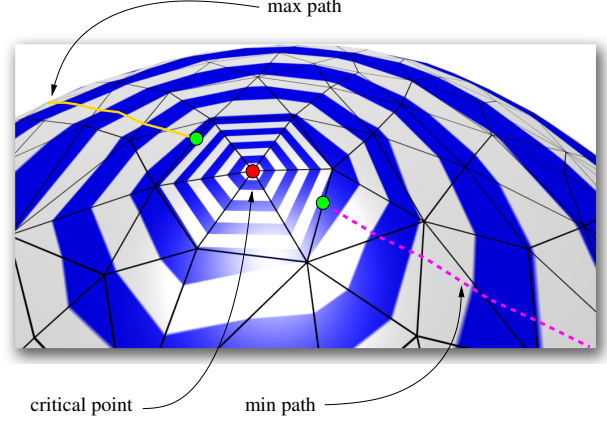


Figure 3: Critical paths end at the edge of a triangle, where one of its vertices is ν_{min} or ν_{max} .

Once the user is satisfied with $\mathbf{u}_{\mathcal{T}_B}$, the harmonic function $\mathbf{v}_{\mathcal{T}_B}$ is computed so that $\nabla \mathbf{u}_{\mathcal{T}_B}$ and $\nabla \mathbf{v}_{\mathcal{T}_B}$ are nearly orthogonal. In order to calculate $\mathbf{v}_{\mathcal{T}_B}$, two seed points s_0 and s_1 on \mathcal{T}_B are chosen. The first seed s_0 can be chosen arbitrarily. Given s_0 , $\nabla \mathbf{u}_{\mathcal{T}_B}$ is used to extract an isoparametric path as in [Dong et al. 2005]. The path is circular, i.e. it starts and ends at s_0 , and it has length l . s_1 is chosen on that path, so the path length between s_0 and s_1 is $l/2$. This 50:50-heuristic has proven to be successful. Note, that $\mathbf{u}_{\mathcal{T}_B}$ and $\mathbf{v}_{\mathcal{T}_B}$ are holomorphic 1-forms as defined in [Arbarello et al. 1938] and used in [Gu and Yau 2003] to compute global conformal parameterizations.

Starting from s_0 two paths are created p_0^+ and p_0^- by following $\nabla \mathbf{u}_{\mathcal{T}_B}$ and $-\nabla \mathbf{u}_{\mathcal{T}_B}$, respectively. They end at the edges of triangles that has ν_{max}/ν_{min} , respectively as one of its vertices (as shown in Figure 3). Merging p_0^+ and p_0^- yields p_0 . Vertices are inserted into the mesh where p_0 intersects edges. Call \mathcal{V}_{min} the set of these vertices. The same procedure is applied to determine p_1 passing through s_1 . Vertices are inserted into the mesh where p_1 intersects edges. These vertices define \mathcal{V}_{max} . Note that $\mathcal{V}_{min} \cap \mathcal{V}_{max} = \emptyset$, and since, as a property of harmonic functions, if there exists only one minimum (ν_{min}) and one maximum (ν_{max}), no saddle points can exist [Ni et al. 2004]. Then the mesh is retriangulated with the new vertex set.

Next, $\nabla^2 \mathbf{v}_{\mathcal{T}_B} = 0$ is solved with $\mathcal{V}_C = \mathcal{V}_{min} \cup \mathcal{V}_{max}$ as the Dirichlet boundary, where we set $\mathbf{v}_{\mathcal{T}_B}(\nu)_{\forall \nu \in \mathcal{V}_{min}} = 0$ and $\mathbf{v}_{\mathcal{T}_B}(\nu)_{\forall \nu \in \mathcal{V}_{max}} = 1$. Since the critical paths p_0 and p_1 do not reach the extremal points ν_{min} and ν_{max} (see Figure 3), $\mathbf{u}_{\mathcal{T}_B}$ and $\mathbf{v}_{\mathcal{T}_B}$ are not appropriately defined inside the ring of triangles around ν_{min} and ν_{max} . Let \mathbf{u}_{start} be the largest \mathbf{u} -value of the vertices defining the ring of ν_{max} , and let \mathbf{u}_{end} be the smallest \mathbf{u} -value of the vertices defining the ring of ν_{min} . Now, given $\mathbf{u}_{\mathcal{T}_B}^{-1}$ and $\mathbf{v}_{\mathcal{T}_B}^{-1}$, the inverse harmonic function \mathfrak{X}_2 is constructed which maps a parametric value in the domain $[\mathbf{u}_{start}, \mathbf{u}_{end}] \times [0, 1]$ onto \mathcal{T}_B , i.e. $\mathfrak{X}_2 : [\mathbf{u}_{start}, \mathbf{u}_{end}] \times [0, 1] \rightarrow \mathcal{T}_B$.

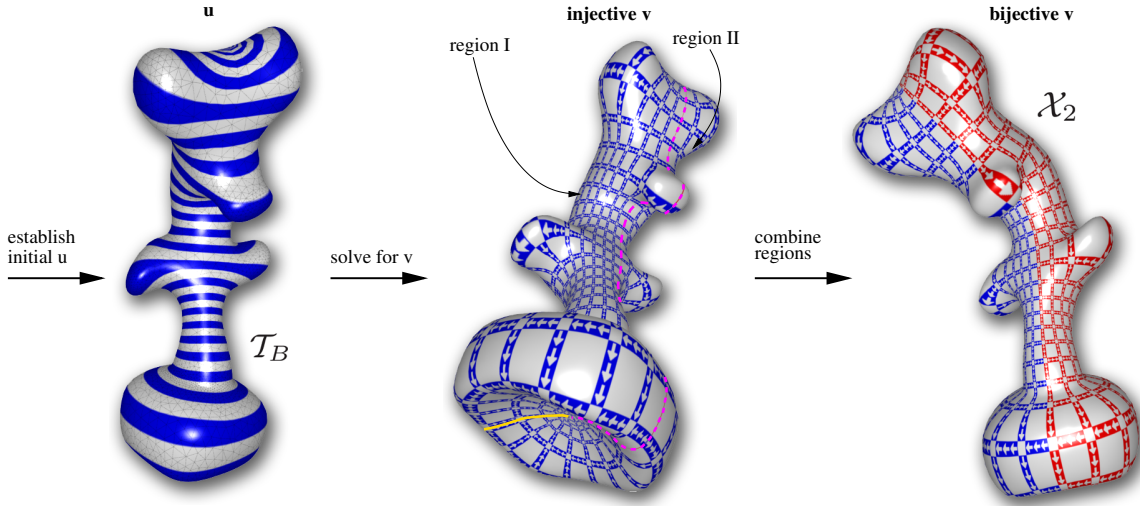


Figure 4: On the left: the harmonic function \mathbf{u}_{T_B} defined by two critical points is established over T_B ; middle: Based on \mathbf{u}_{T_B} , the orthogonal harmonic function \mathbf{v}_{T_B} is calculated. At this stage \mathbf{u}_{T_B} and \mathbf{v}_{T_B} define an injective transformation; on the right: Scaling and translation yields the parameterization \mathcal{X}_2 .

\mathcal{X}_2 is not bijective yet as the figure on the right illustrates. It shows a closed isoparametric line in \mathbf{u}_{T_B} , i.e. a closed piecewise polyline where each of its vertices has the same \mathbf{u} -value. The paths p_0 and p_1 divide the exterior surface into two regions I and II . Let $\alpha(\nu)$ be the part of the harmonic \mathbf{v} -mapping which maps a vertex ν in region I onto $[0, 1]$. The corresponding function for region II is called $\beta(\nu)$. In order to make \mathcal{X}_2 bijective we define a single harmonic \mathbf{v} -mapping

$$\gamma(\nu) = \begin{cases} \alpha(\nu)/2 & , \nu \in I \\ 1 - \beta(\nu)/2 & , \nu \in II \end{cases}$$

Figure 4 illustrates these transformations. At this stage, every $\nu \in \mathcal{V}_B$ has a \mathbf{u} - and \mathbf{v} -parameter value. Note that \mathbf{v} is periodic so $0 \equiv 1$.

A region whose corners consists of right-angles can be parameterized so that the resulting gradient fields are orthogonal [Tong et al. 2006]. However, in our case, \mathbf{u}_{T_B} degenerates to points (ν_{min} and ν_{max}), implying that $\nabla \mathbf{u}_{T_B}$ and $\nabla \mathbf{v}_{T_B}$ are not orthogonal near ν_{min} and ν_{max} . This means that a quadrangulation in this area is of poorer quality. Note, that ν_{min} and ν_{max} were chosen in areas which are not important in the proposed simulation. Furthermore, in case of the femur, 98% of the gradient vectors in $\nabla \mathbf{u}_{T_B}$ and $\nabla \mathbf{v}_{T_B}$ have an angle α between each other, where $\pi/2 - 0.17 < \alpha < \pi/2 + 0.13$.

5.1 \mathbf{u} - and \mathbf{v} -section Extraction

Similar to [Hormann and Greiner 2000], our goal is to extract a set of \mathbf{u} - and \mathbf{v} -parameter values so that the corresponding isoparametric curves on the model define a structured quadrilateral mesh which represents the exterior of the tetrahedral mesh faithfully.

Let $\widehat{T_B}$ be the exterior triangle mesh inversely mapped into the parameter space as illustrated in Figure 5 (left). We seek to find a set $U = \{\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{n_0}\}$ of \mathbf{u} -values and a set $V = \{\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_{n_1}\}$ of \mathbf{v} -values so that the collection of images of the grid form an error bounded grid to the model. The isocurve at

a fixed $\mathbf{u}_i \in U$ corresponds to the line $l_i(\mathbf{v}) = (\mathbf{u}_i, \mathbf{v})$ in parameter space, where $\mathbf{v} \in [0, 1]$, and, the isocurve at a fixed $\mathbf{v}_j \in V$ corresponds to the line $\widehat{l}_j(\mathbf{u}) = (\mathbf{u}, \mathbf{v}_j)$ in parameter space, where $\mathbf{u} \in [0, 1]$. $l_i(\mathbf{v})$ and $\widehat{l}_j(\mathbf{u})$ are orthogonal. Note, that \mathcal{X}_2 maps $l_i(\mathbf{v})$ and $\widehat{l}_j(\mathbf{u})$ to isocurves on T_B . The intersections of the lines $l_i(\mathbf{v})$ and $\widehat{l}_j(\mathbf{u})$ i.e. the parameter pairs $(\mathbf{u}_i, \mathbf{v}_j)$ define a structured grid with rectangular grid cells over the parametric domain and hence a quadrilateral mesh over T_B . This quadrilateral mesh is a remesh of T_B .

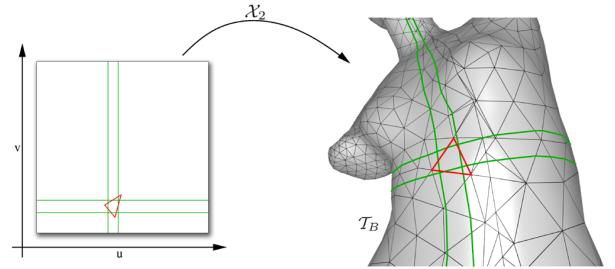


Figure 5: \mathcal{X} maps a vertical line at \mathbf{u}_0 in parameter space onto a closed isoparametric line on T_B . Accordingly, \mathcal{X} maps a horizontal line at \mathbf{v}_0 onto an isoparametric which starts at ν_{min} and ends at ν_{max} .

Let \mathcal{E} be the set of edges defining the triangles in $\widehat{T_B}$. U and V are chosen so that every edge in \mathcal{E} is intersected by at least one $l_i(\mathbf{v})$ and one $\widehat{l}_j(\mathbf{u})$, as shown in Figure 5 for one triangle. U and V are calculated independently from each other. An edge $e \in \mathcal{E}$ is defined by two points in parametric space $(\mathbf{u}_e, \mathbf{v}_e)$ and $(\mathbf{u}'_e, \mathbf{v}'_e)$. Based on \mathcal{E} , we define S_u to be the set of intervals defined as the collection of intervals $(\mathbf{u}_e, \mathbf{u}'_e)$ such that $(\mathbf{u}_e, \mathbf{v}_e)$ and $(\mathbf{u}'_e, \mathbf{v}'_e)$ are the endpoints defining an edge $e \in \mathcal{E}$. We employ the interval structure for stabbing queries [Edelsbrunner 1980], that takes a set of intervals (in our case S_u) and constructs an interval tree \mathcal{I}_u in $O(n \log n)$, where n is the number of intervals in S_u . Every node in \mathcal{I}_u includes an interval location $\mathbf{u} \in [0, 1]$. \mathcal{I}_u covers every interval \rightarrow edge \rightarrow triangle in T_B . The \mathbf{u} -values of the nodes in the tree define the set U and the vertical stabbing lines $l_i(\mathbf{v})$.

V is defined analogously, with the difference that S_v consists of intervals defined by the segments (v_e, v'_e) for which (u_e, v_e) and (u'_e, v'_e) are the endpoints of an edge $e \in \mathcal{E}$. Then, V consists of the v -values defining the nodes in \mathcal{I}_v and the horizontal stabbing lines $\hat{l}_j(u)$.

The algorithm to determine U and V guarantees that in a rectangle defined by the points $p_0 = (u_i, v_j)$, $p_1 = (u_{i+1}, v_j)$, $p_2 = (u_{i+1}, v_{j+1})$ and $p_3 = (u_i, v_{j+1})$, where $u_i, u_{i+1} \in U$ and $v_j, v_{j+1} \in V$, there is either the preimage of at most one vertex of a triangle (Case 1) or none (Case 2). These two cases are illustrated in Figure 6.

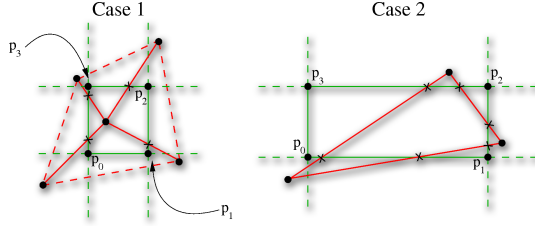


Figure 6: Either there is one vertex in the rectangle defined by the points p_0, p_1, p_2 and p_3 , or none. Crosses mark edge intersections.

We show this is true by contradiction. That is, assume that there are two vertices in the same rectangle. Since we require that the input mesh is a 2-manifold, there has to be a path defined by triangle edges from one vertex to the other. However, due to the interval tree property that every interval is cut by at least one stabbing line, at least one isoline with fixed u -value and one isoline with fixed v -value intersects an edge. Therefore, the two vertices must be separated.

Now we want to ensure that the quadrilateral grid that we are deriving is within error tolerance. Let us consider the rectangle $\hat{R}_{i,j}$ defined by the points (u_i, v_j) and (u_{i+1}, v_{j+1}) (as in Figure 6). The vertices of its corresponding bilinear surface $R_{i,j}$ on \mathcal{T}_B are $\mathcal{X}_2(u_i, v_j)$, $\mathcal{X}_2(u_{i+1}, v_j)$, $\mathcal{X}_2(u_{i+1}, v_{j+1})$ and $\mathcal{X}_2(u_i, v_{j+1})$. We measure how far $R_{i,j}$ is away from the triangle mesh. We look at this measurement for the two above cases separately.

For case 1, let (u^*, v^*) be the parameter value of the vertex lying in $\hat{R}_{i,j}$. Consider one of the triangles associated with that point, each edge of $\hat{R}_{i,j}$ maybe intersected by either zero, one, or two of the triangle's edges. If intersections exist, we transform them with \mathcal{X}_2 onto \mathcal{T}_B and measure how far they are away from $R_{i,j}$. Furthermore, the distance between $\mathcal{X}_2(u^*, v^*)$ and $R_{i,j}$ is determined. Given a user-defined ϵ , if the maximum of all these distances is smaller than $\epsilon/2$, we have sufficient accuracy, if not, then we insert a new u -slice between u_i and u_{i+1} and a new v -slice between v_j and v_{j+1} and reexamine the newly created rectangles. Case 2 is handled similarly to Case 1 without the projection of the interior point.

Depending on the resolution of \mathcal{T}_B , the sets U and V may have more parameter values than necessary. For instance, if \mathcal{T}_B is a densely triangulated cylinder, most of the parameter values in U are not necessary. To some extent, more isolines are needed around features. On the other hand, isolines might also be needed in areas on which force due to boundary conditions is applied. These regions could have no shape features at all. After the B-spline volume is modeled using our framework, refinement and data reduction techniques are applied to yield trivariate approximations with different resolutions. However, it still can be helpful to remesh the input triangle meshes with a feature aware triangulator such as Afront

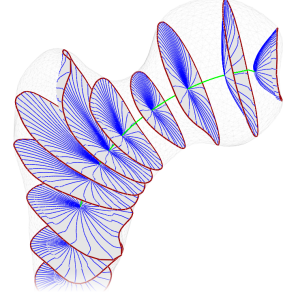
[Schreiner et al. 2006] which generates meshes having more triangles in regions with higher curvature and fewer triangles in regions with very low curvature.

Given an input triangle mesh, an upper bound on the error can be determined. Since there is a guarantee that every edge is intersected by at least one isoline with fixed u - and one isoline with fixed v -value, the maximum error can be computed in the following way: Given \mathcal{T}_B , we consider the ring of a vertex $\nu \in \mathcal{V}_B$, where the ring is the set of all adjacent vertices of ν being elements in \mathcal{V}_B . We construct a bounding box where one of its axes are coincident with the normal of ν . The height of the bounding box side coincident with the normal of ν is the error for that ring. We compute such a bounding box for every vertex on the exterior. The maximum height will be the maximum error.

6 Modeling the Interior

Once the exterior parameterization is determined, the tetrahedral mesh (\mathcal{H}) is parameterized. Using FEM, $\nabla^2 u_{\mathcal{H}} = 0$ is solved, where \mathcal{V}_B with its respective u -values is used as the Dirichlet boundary condition. Now, all elements in \mathcal{V} have a u -parameter value. In the surface case, fixing a u -value gives a line in parameter space and a closed isocurve on the surface. In the volume case, fixing a u -value gives a plane in parameter space and a surface called an isosheet in the volume. The boundary of an isosheet for a fixed u_0 is the isocurve on the surface at u_0 .

Now, for each boundary slice at u_{i_0} , it is necessary to extract its corresponding isosheet. First however, a skeleton is created to serve as isosheet center for all isosheets. Then, a third function w is created whose gradient field ∇w points to the skeleton. ∇w is used to trace a path starting at $p_{i_0,j} = \mathcal{X}_2(u_{i_0}, v_j)$ and ending at the skeleton on the sheet, for $j = 0, \dots, n_1$ (see right). ∇w is constructed to be tangent to the isosheet at a given point, so $\nabla u_{\mathcal{H}}$ and ∇w are orthogonal. This guarantees that every point on the extracted w -path has the same u -value.



The skeleton is created by tracing two paths which start at a user-specified seed using $+\nabla u_{\mathcal{H}}$ and $-\nabla u_{\mathcal{H}}$ and end at ν_{min} and ν_{max} , respectively. Merging these two paths yield the skeleton. By the definition of $\nabla u_{\mathcal{H}}$, the skeleton can have no loops. The skeleton has the properties of a Reeb Graph [Shinagawa et al. 1991], in that its end vertices correspond to ν_{min} and ν_{max} . While the Reeb graphs in [Shinagawa et al. 1991] are defined over a surface, our Reeb graph, i.e. the skeleton, is defined over the volume. Because of the way ∇w is built, a sheet is orthogonal to the skeleton, which is also a property of GC. The orthogonal property of the skeleton and ∇w is also a desirable property to attain a good B-spline fit. The skeleton can be computed in interactive time, and the user has flexibility in choosing the seed. In general, the seed should be placed such that the resulting skeleton lies in the “center” of the innermost isosurface, like the axis of a cylinder.

Just solving $\nabla^2 w = 0$ with the respective boundary conditions does not guarantee orthogonality of $\nabla u_{\mathcal{H}}$ and ∇w , and if $\nabla u_{\mathcal{H}}$ and ∇w are not orthogonal, there is no reason that a path will have the same u -value throughout. This implies the w -parameter will need further adjustment to guarantee a well behaved parameterization and so adjacent isosheets do not overlap. In order to enforce orthogonality ∇w is constructed in the following two steps:

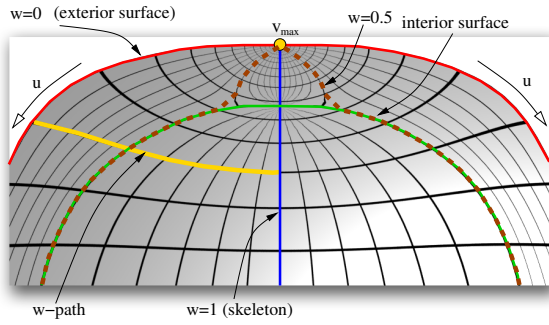


Figure 7: A cross section of an object with an exterior boundary and an interior isosurface representing geometry or attribute data. The skeleton and boundaries were used to establish $\nabla \mathbf{w}$. Isolines visualize the $\mathbf{u}\mathbf{w}$ -scalar field used to trace \mathbf{w} -paths from the exterior to the skeleton.

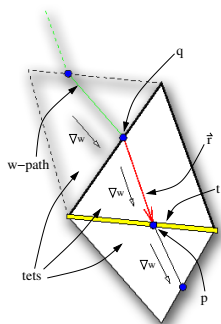
(1) The points defining the skeleton are inserted into the tetrahedral mesh and a new mesh is formed. Then, $\nabla^2 \hat{\mathbf{w}} = 0$ is solved over the tetrahedral mesh, subject to Dirichlet boundary conditions defined by the set $\mathcal{V}_C = \mathcal{V}_B \cup \mathcal{V}_{T_1} \cup \dots \cup \mathcal{V}_{T_k} \cup \mathcal{V}_S$. \mathcal{V}_B consists of the boundary vertices where $\hat{\mathbf{w}}(\nu)_{\forall \nu \in \mathcal{V}_B} = 0$, \mathcal{V}_S consists of the vertices defining the skeleton where $\hat{\mathbf{w}}(\nu)_{\forall \nu \in \mathcal{V}_S} = 1$, \mathcal{V}_{T_i} is the set of vertices defining the i th of k isosurfaces where $\hat{\mathbf{w}}(\nu)_{\forall \nu \in \mathcal{V}_{T_i}} = i/(k+1)$. In the case of the femur and in Figure 7, there is one isosurface, namely the surface separating the trabecular and cortical bone. In this case $\mathcal{V}_C = \mathcal{V}_B \cup \mathcal{V}_{T_1} \cup \mathcal{V}_S$, where $\hat{\mathbf{w}}(\nu)_{\forall \nu \in \mathcal{V}_{T_1}} = 1/2$. Then in step (2), for every tetrahedron, we project its $\nabla \hat{\mathbf{w}}$ gradient vector onto the plane whose normal is the corresponding $\nabla \mathbf{u}_{\mathcal{H}}$, to form $\nabla \mathbf{w}$.

6.1 Tracing w-paths

Flow line extraction over a closed surface triangle mesh is described in [Dong et al. 2005]. In our case, we extract flow lines throughout a volume. A flow line, or a \mathbf{w} -path will start on \mathcal{T}_B , where $\mathbf{w} = 0$ and traverses through \mathcal{H} until it reaches the skeleton on which $\mathbf{w} = 1$. The resulting \mathbf{w} -path is a piecewise linear curve where every segment belongs in a tetrahedron. The two ends of the segment lie on faces of the respective tetrahedron and is coincident with $\nabla \mathbf{w}$. Since $\nabla \mathbf{u}_{\mathcal{H}}$ and $\nabla \mathbf{w}$ are orthogonal, every point on such a segment has a constant \mathbf{u} -value, and therefore, the \mathbf{w} -path has a constant \mathbf{u} -value.

During \mathbf{w} -path traversal, in the regular case, the endpoint q of the \mathbf{w} -path will lie on a face of a tetrahedron. The next traversal point is determined by constructing a ray \vec{r} with origin at q with $\nabla \mathbf{w}_{\mathcal{H}}$ of the adjacent tetrahedron as direction. \vec{r} is then intersected with the faces of the adjacent tetrahedron, except the triangle on which q lies, to find the next q . Let p be the intersection between \vec{r} and triangle t . t is a face of two tetrahedra, the current and the next tetrahedron. The line segment \vec{qp} is added to the current \mathbf{w} -path, and p becomes q .

During the \mathbf{w} -path traversal, several pathological cases can arise. One is when the intersection point p lies on an edge e of the current tetrahedron. Since the edge is part of two triangles, an ambiguity exists as to which face should be chosen. Instead we consider all tetrahedra that have e as an edge. For each of these tetrahedra we



construct a ray having its origin at p with $\nabla \mathbf{w}$ of the tetrahedron as its direction. If there is an intersection between a tetrahedron's ray and one of its faces, then we choose that face of the respective tetrahedron as the next triangle. Analogously, at the other degeneracy, when p lies at a vertex of the tetrahedron, we examine every tetrahedron that coincides with this vertex. We choose the tetrahedron in which we can move furthest in $\nabla \mathbf{w}$ direction.

Another degenerate case arises when \vec{r} does not intersect with any triangle, edge or vertex of the current tetrahedron. This implies that \vec{r} points outward from the tetrahedron. When this occurs, we construct a plane through q orthogonal to $\nabla \mathbf{u}_{\mathcal{H}}$ of the current tetrahedron. Every point on that plane in the tetrahedron has the same \mathbf{u} -value. We intersect the plane with the edges of the triangle in which q is located. In general position, there are two intersections. We choose that intersection which has a bigger \mathbf{w} -value as next point on the \mathbf{w} -path, because it lies closer to the skeleton. Since the intersection point is a point on an edge or a vertex, the first special case is applied.

6.2 w-path Extraction

In Section 5.1, we discussed how the Cartesian product of the sets U and V spans over the $\mathbf{u}\mathbf{v}$ -domain. \mathcal{X}_2 maps the grid point $(\mathbf{u}_i, \mathbf{v}_j)$ to the point $\mathbf{p}_{i,j}$ in \mathcal{T}_B . The points $\mathbf{p}_{i,j}$ are used as starting points to trace \mathbf{w} -paths, as described above in Section 6.1. Now, $\mathcal{X}_3 : [\mathbf{u}_{start}, \mathbf{u}_{end}] \times [0, 1] \times [0, 1] \rightarrow \mathcal{H}$ is a parameterization in three variables \mathbf{u} , \mathbf{v} and \mathbf{w} , where $\mathcal{X}_3(\mathbf{u}, \mathbf{v}, 0) \equiv \mathcal{X}_2(\mathbf{u}, \mathbf{v})$, $\mathcal{X}_3(\mathbf{u}, \mathbf{v}, 1)$ defines the skeleton, and $\mathcal{X}_3(\mathbf{u}, \mathbf{v}, i/(k+1))$ defines the i th isosurface.

In this section, we want to find a set $W = \{\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_{n_2}\}$ where $\mathbf{w}_0 = 0$ and $\mathbf{w}_{n_2} = 1$, which contains n_2 parameter values. The Cartesian product $U \times V \times W$ defines a structured grid on $[\mathbf{u}_{start}, \mathbf{u}_{end}] \times [0, 1] \times [0, 1]$ and a structured hexahedral mesh with points $\mathbf{p}_{i,j,k} = \mathcal{X}_3(\mathbf{u}_i, \mathbf{v}_j, \mathbf{w}_k)$ in \mathcal{H} with degeneracies only along the skeletal axis. Note, that $\mathbf{p}_{i,j,k}$ refers to the k th point on the j th \mathbf{w} -path on isosheet i , i.e. by fixing \mathbf{u}_{i_0} , the points $\mathcal{X}_3(\mathbf{u}_{i_0}, \mathbf{v}_j, \mathbf{w}_k)$ lie and approximate isosheet i_0 and connect to a structured quadrilateral mesh called \mathcal{S}_{i_0} .

Let $\mathcal{h}_{i_0, j_0} : [0, 1] \rightarrow \mathcal{H}$ be the j_0 th \mathbf{w} -path on \mathcal{S}_{i_0} , defined by the points $\mathbf{p}_{i_0, j_0, k}$, where $k = 0, \dots, n_2$. Depending on the choice of \mathbf{w} -values in W , points on \mathcal{h}_{i_0, j_0} may have different \mathbf{u} -values. This leads to a modification of \mathbf{u} on the interior parameterization. This is allowed as long the \mathbf{u} -value of these points is smaller (bigger) than the \mathbf{u} -value of the upper (lower) adjacent isosheet. Otherwise, isosheets might intersect.

Originally, when a \mathbf{w} -path is extracted as discussed above, all parameter values in $[0, 1]$ map to points whose \mathbf{u} -values are the same. The reason for that is, that the line segments defining the initial \mathbf{w} -path all lie in tetrahedra and coincide with $\nabla \mathbf{w}_{\mathcal{H}}$ of the respective tetrahedron. Furthermore, every extracted \mathbf{w} -path is defined initially by different sets of \mathbf{w} -values. In order to determine W , we have to make sure that the quadrilateral sheets \mathcal{S}_i do not intersect.

Let $W_{i,j} = \{\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_{n_{2i,j}}\}$, where $\mathbf{w}_0 = 0 < \mathbf{w}_1 < \dots < \mathbf{w}_{n_{2i,j}} = 1$, be the sorted set of \mathbf{w} -parameter values for \mathcal{h}_{i_0, j_0} , consisting of $n_{2i,j} + 1$ points. $n_{2i,j}$ depends on the number of tetrahedra the path is travelling through. If the path is close to ν_{min} or ν_{max} , the path is probably shorter than a path towards to the middle of the object. A valid and common W could be found by calculating the union of all $W_{i,j}$, where then W would contain an unnecessarily large number of \mathbf{w} -values. Therefore as a first step we simplify every $W_{i,j}$ by removing unnecessary \mathbf{w} -values from it. Let \mathbf{u}_i be the \mathbf{u} -value of the current slice. We scan $W_{i,j}$ and remove an element \mathbf{w}_k when the \mathbf{u} -value of the point $(P_{k-1} + P_{k+1})/2$

is in the range $[(\mathbf{u}_i - \mathbf{u}_{i-1})/2, (\mathbf{u}_i + \mathbf{u}_{i+1})/2]$, where P_k is the position in the tetrahedral mesh where \mathbf{w}_k lies. This implies that sheet i does not intersect with one of its adjacent sheets. The P_k 's leading to the smallest differences are removed first. This is applied iteratively until no further points can be removed from the path.

Then, we merge the simplified sets $W_{i,j}$ to get W . After merging, elements in W may be very close to each other. We therefore remove elements in W , such that every pair of parameter values in W has at least a distance (in parameter space) of ϵ between each other. Furthermore, the parametric \mathbf{w} -values for the isosurfaces are added to W , too, such as $0.5 \in W$, where $\mathcal{X}_3(\mathbf{u}, \mathbf{v}, 0.5)$ represents the inner boundary in Figure 2.

6.3 w-path Smoothing

Since there is only an exterior \mathbf{v} -parameterization, points lying on a \mathbf{w} -path have a constant \mathbf{u} -value but not a constant \mathbf{v} -value. This results in path wiggling as shown in Figure 8. Path wiggling means that parts of a given path may lie closer to its adjacent path on the left than to its adjacent path on the right.

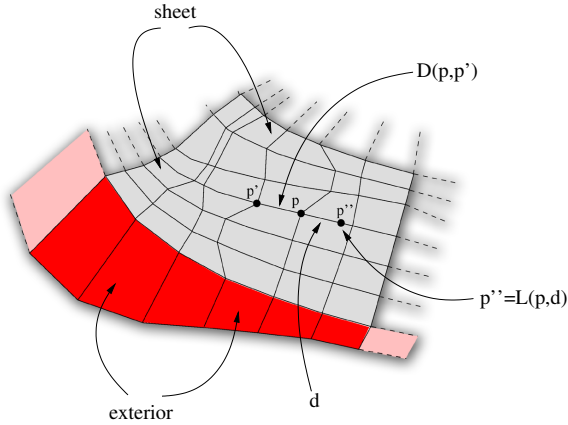


Figure 8: Due to the linear property of $\nabla \mathbf{u}$ and $\nabla \mathbf{v}$ and special cases during \mathbf{w} -path extraction as discussed in section 6.1, adjacent paths may collapse.

Laplacian smoothing [Freitag and Plassmann 1997] is an efficient way to smooth a mesh and remove irregularities. As pointed out in [Freitag and Plassmann 1997], applying it to a hexahedral mesh can lead to inconsistencies, like “tangling” of hexahedra. This especially happens in regions with overhangs, where in our case, Laplacian smoothing would change the \mathbf{u} - and \mathbf{w} -value of a point, leading to overlapping sheets. However, Laplacian smoothing is computationally efficient and we adapt it for our case in the following way.

Let $\nabla \mathbf{v}_{\mathcal{H}}$ be the cross product field between $\nabla \mathbf{u}_{\mathcal{H}}$ and $\nabla \mathbf{w}$, i.e. $\nabla \mathbf{v}_{\mathcal{H}} = \nabla \mathbf{u}_{\mathcal{H}} \times \nabla \mathbf{w}_{\mathcal{H}}$. Since \mathbf{v}_j is not constant along the \mathbf{w} -path, during mesh smoothing, we restrict the location $\mathbf{p}_{i,j,k}$ to change only along $\nabla \mathbf{v}_{\mathcal{H}}$. Let $L(\mathbf{p}, d)$ be a function defined over \mathcal{H} which determines a point \mathbf{p}' along $\nabla \mathbf{v}_{\mathcal{H}}$ with distance d from \mathbf{p} . \mathbf{p} and \mathbf{p}' both lie on a piecewise linear \mathbf{v} -path, and the \mathbf{v} -path section defined by \mathbf{p} and \mathbf{p}' has a length of d . Since $\nabla \mathbf{u}_{\mathcal{H}}$, $\nabla \mathbf{v}_{\mathcal{H}}$ and $\nabla \mathbf{w}$ are orthogonal vector fields, \mathbf{p} and \mathbf{p}' have the same \mathbf{u} - and \mathbf{w} -value. Furthermore, let $D(\mathbf{p}, \mathbf{q})$ be a function that computes the length of the \mathbf{v} -path section defined by \mathbf{p} and \mathbf{q} , requiring that \mathbf{p} and \mathbf{q} have the same \mathbf{u} - and \mathbf{w} -value. Now, the position of the mesh point $\mathbf{p}_{i,j,k}$ is updated by

$$\mathbf{p}'_{i,j,k} = L(\mathbf{p}_{i,j,k}, \frac{1}{2}(D(\mathbf{p}_{i,j,k}, \mathbf{p}_{i,j-1,k}) + D(\mathbf{p}_{i,j,k}, \mathbf{p}_{i,j+1,k}))).$$

After this procedure is applied to every $\mathbf{p}_{i,j,k}$ where $i > 0$ and $i < n_0$, the old positions $\mathbf{p}_{i,j,k}$ are overwritten with $\mathbf{p}'_{i,j,k}$. By repeating this procedure the mesh vertices move so that for a given $\mathbf{p}_{i,j,k}$, the ratio $D(\mathbf{p}_{i,j,k}, \mathbf{p}_{i,j+1,k})/D(\mathbf{p}_{i,j,k}, \mathbf{p}_{i,j-1,k})$ moves closer to 1, by maintaining a constant \mathbf{u} - and \mathbf{w} -value. Therefore, this approach avoids isosheet intersection. The procedure terminates when $\max |D(\mathbf{p}_{i,j,k}, \mathbf{p}_{i,j+1,k})| < \epsilon$, where ϵ is user-defined.

7 B-spline volume fitting

In the first stages of our framework we created a structured $(n_0 + 1) \times (n_1 + 1) \times (n_2 + 1)$ hexahedral mesh with vertices $\mathbf{p}_{i,j,k}$, from a set of unstructured triangle meshes. The hexahedral mesh has the same tensor-product nature as a trivariate B-spline. In this section we want to fit a trivariate B-spline to this grid. One of the first decisions to make is to choose between an interpolation or an approximation scheme. Our criteria include generating a consistent mesh, where adjacent sheets do not overlap, and minimizing oscillations in the B-spline volume.

The first aspect would imply an interpolation scheme: Since the points of the hexahedral mesh lie on the resulting B-spline, the error on these points is zero. However, interpolation can cause oscillations and there are no guarantees that the B-spline is consistent. Since the initial hexahedral mesh can have a very high resolution, solving a global interpolation problem requires additional extensive computation time. Furthermore, the input triangle meshes were eventually acquired through segmentation of volumetric image data, they approximate the original data already, especially after a triangle remesh. Interpolation of such an approximation would not necessarily make sense. Therefore, the second aspect implies an approximation scheme that also avoids wrinkles in the final mesh. The question is here, what approximation error should be chosen. This depends on the hexahedral mesh. Sheets which are bent need an adequate number of control points so that the intersection among adjacent sheets is avoided. The choice of an appropriate number of control points is difficult to determine.

We therefore adopt an approach which is a mix of both, maximizing its advantages and minimizing its drawbacks. We allow the user to control how close the B-spline is to the approximating points of the hexahedral mesh. A consistent B-spline with as few oscillations as possible is desirable. Our solution is to develop an approximation iteratively. The hexahedral mesh is chosen as the initial control mesh. This guarantees that the B-Spline volume lies inside the control volume and that no further features are introduced to the B-spline volume. Furthermore, we set degrees in the three directions $p_0 = 3$, $p_1 = 3$ and $p_2 = 1$, and use a uniform open knot vector in \mathbf{u} and \mathbf{w} , and a uniform periodic knot vector in \mathbf{v} .

For a fixed k_0 , let P_{i,j,k_0}^c be the c th control mesh in an iterative relaxation procedure, where $S_{k_0}^c(u, v)$ is the B-spline surface at iteration c with control mesh P_{i,j,k_0}^c . At $c = 0$, set $P_{i,j,k_0}^0 := \mathbf{p}_{i,j,k_0}$. In the c th iteration, where $c > 0$, we update

$$P_{i,j,k_0}^{c+1} = P_{i,j,k_0}^c + \lambda \Delta^{[c]}, \quad (2)$$

where $\Delta^{[c]}$ is a direction vector and is chosen such that $S_{k_0}^c(u, v)$ grows towards \mathbf{p}_{i,j,k_0} . $\lambda \in (0, 1)$ is a user-defined scalar, in our case $\lambda = 0.5$.

$\Delta^{[c]}$ is defined in terms of \mathbf{p}_{i,j,k_0} and $S_{k_0}^c(u^*, v^*)$ corresponding to the control point P_{i,j,k_0}^c . u^* and v^* can be determined by projecting the control point P_{i,j,k_0}^c onto $S_{k_0}^c$. A first-order approximation to this projection is to evaluate $S_{k_0}^c$ at the appropriate node [Cohen et al. 2001], i.e. $u_i^* = \sum_{\mu=1}^3 t_{i+\mu}^0/3$ and $v_j^* = \sum_{\mu=1}^3 t_{j+\mu}^1/3$. Since T^1 is uniform periodic, $v_j^* = t_{j+p-1}^1$,

where v^* in that case is also exact and corresponds to the j th control point. This is not true for the uniform open knot vectors T^0 . Either $t_{i+2}^u \leq u_i^* \leq t_{i+3}^u$ (Case 1) or $t_{i+1}^u \leq u_i^* \leq t_{i+2}^u$ (Case 2), therefore $S_{k_0}^c(u_i^*, v_j^*)$ lies only near P_{i,j,k_0}^c . If Case 1 applies, then let $i_0 = i - 1$, otherwise for Case 2, let $i_0 = i - 2$. Then, $S_{k_0}^c(u_i^*, v_j^*) = \sum_{k=0}^p (B_{i_0+k,p_0}(u_i^*) C_{i_0+k,j})$, where $C_{k,j} = (P_{k,j-1,k_0}^c + P_{k,j+1,k_0}^c)/6 + (2P_{k,j,k_0}^c)/3$. Note that, $B_{j-1,q}(v_j^*) = B_{j+1,q}(v_j^*) = 1/6$ and $B_{j,q}(v_j^*) = 2/3$.

In order to define $\Delta^{[c]}$, we ask how to change the current control point P_{i,j,k_0}^c such that $S_{k_0}^{c+1}(u_i^*, v_j^*)$ moves closer to \mathbf{p}_{i,j,k_0} . To answer this, we set

$$\overline{S_{k_0}^c(u_i^*, v_j^*)} = S_{k_0}^c(u_i^*, v_j^*) - \frac{2B_{i,p}(u_i^*)}{3} P_{i,j,k_0}^c,$$

and rewrite

$$\begin{aligned} \mathbf{p}_{i,j,k_0} &= \overline{S_{k_0}^c(u_i^*, v_j^*)} + \frac{2B_{i,p}(u_i^*)}{3} (P_{i,j,k_0}^c + \Delta^{[c]}) \quad (3) \\ \Delta^{[c]} &= \frac{3}{2B_{i,p}(u_i^*)} (\mathbf{p}_{i,j,k_0} - S_{k_0}^c(u_i^*, v_j^*)). \quad (4) \end{aligned}$$

The iteration stops when $\epsilon_{max} < \epsilon$, where $\epsilon_{max} = \max \|\mathbf{p}_{i,j,k} - S_{k_0}^c(u_i^*, v_j^*)\|_2$ for every $S_{k_0}^c$. $\|\cdot\|_2$ is the second vector norm and ϵ is user-defined. The resulting surfaces $S_{k_0}^c$ define the final trivariate B-spline control mesh.

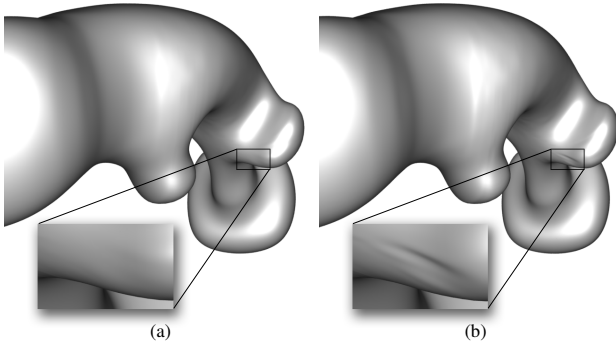


Figure 9: Different choices of λ achieve different qualities of approximations. For (a) $\lambda = 0.1$ and for (b) $\lambda = 0.8$ was used. In both cases $\epsilon = 0.01$.

The user choice of λ affects quality and running time of our proposed approximation method. Choosing a λ closer to one reduces the number of iterations but lowers the quality of the final solution. A λ closer to zero requires more iterations but results in a higher quality mesh. Please refer to figure 9 which shows the results for $\lambda = 0.1$ and $\lambda = 0.8$. For $\lambda = 0.1$, 12 iterations were required; for $\lambda = 0.8$ the algorithm terminated after three iterations. In both cases, $\epsilon = 0.01$. For $\lambda = 0.8$, it can be observed that the resulting mesh contains unpleasant wrinkles, as they typically appear in interpolation schemes.

7.1 Convergence

The proposed method converges, when at every step the magnitude of $\Delta_i^{[c]}$ gets smaller, and in the limit

$$\lim_{c \rightarrow \infty} \|\Delta_i^{[c]}\|_2 = 0.$$

Let us consider the 2D case with a uniform periodic knot vector T . The points \mathbf{p}_i , where $i = 0, \dots, n-1$, define a closed polyline. As above, the initial vertices which define the control polygon are

$P_i^{[0]} := \mathbf{p}_i$. We want to iteratively move the B-spline curve defined by $\{P_i^{[c]}\}$ and T closer to the initial data points $\{\mathbf{p}_i\}$, where

$$P_i^{[c+1]} = P_i^{[c]} + \lambda \Delta_i^{[c]}.$$

Due to the periodic and uniform knot vector T ,

$$\begin{aligned} \mathbf{p}_i &= \frac{1}{6} (P_{i-1}^{[c]} + P_{i+1}^{[c]}) + \frac{2}{3} (P_i^{[c+1]}) \\ &= \frac{1}{6} (P_{i-1}^{[c]} + P_{i+1}^{[c]}) + \frac{2}{3} (P_i^{[c]} + \Delta_i^{[c]}). \end{aligned}$$

Solving for $\Delta_i^{[c]}$ yields,

$$\Delta_i^{[c]} = \frac{3}{2} \left(\mathbf{p}_i - \left(\frac{1}{6} (P_{i-1}^{[c]} + P_{i+1}^{[c]}) + \frac{2}{3} P_i^{[c]} \right) \right). \quad (5)$$

In matrix notation, Equation 5 can be rewritten as

$$\vec{\Delta}^{[c]} = \frac{3}{2} (\mathbf{p} - \mathbf{C} \cdot \vec{P}^{[c]}), \quad (6)$$

where “ \cdot ” is the matrix-vector product. $\vec{\Delta}^{[c]}$ and $\vec{P}^{[c]}$ are vectors with n components, where the i th component is $\Delta_i^{[c]}$ and $P_i^{[c]}$, respectively, and \mathbf{C} is a $n \times n$ circulant matrix [Davis 1979], where row i is a circular shift of i components of the n -component row vector $[\frac{2}{3}, \frac{1}{6}, 0, \dots, 0, \frac{1}{6}]$, in short

$$\mathbf{C} = \text{circ} \left(\frac{2}{3}, \frac{1}{6}, 0, \dots, 0, \frac{1}{6} \right).$$

Note, that if $c = 0$, then

$$\vec{\Delta}^{[0]} = \frac{3}{2} (\vec{p} - \mathbf{C} \cdot \vec{P}^{[0]}) = \frac{3}{2} (\vec{p} - \mathbf{C} \cdot \vec{p}) = \frac{3}{2} (\mathbf{I} - \mathbf{C}) \cdot \vec{p},$$

where \mathbf{I} is the identity matrix.

If $c = 1$, then

$$\vec{\Delta}^{[1]} = \frac{3}{2} (\vec{p} - \mathbf{C} \cdot \vec{P}^{[1]}) = \left(\mathbf{I} - \frac{3}{2} \lambda \mathbf{C} \right) \cdot \vec{\Delta}^{[0]}.$$

From that, induction is used to show that

$$\vec{\Delta}^{[c+1]} = \left(\mathbf{I} - \frac{3}{2} \lambda \mathbf{C} \right) \cdot \vec{\Delta}^{[c]} = \mathbf{A}^{c+1} \cdot \frac{3}{2} (\mathbf{I} - \mathbf{C}) \vec{p}, \quad (7)$$

where

$$\mathbf{A} = \mathbf{I} - \frac{3}{2} \lambda \mathbf{C} = \text{circ} \left(1 - \lambda, -\frac{\lambda}{4}, 0, \dots, 0, -\frac{\lambda}{4} \right).$$

The magnitude of $\Delta^{[c]}$ converges against 0, implying that our fitting procedure converges, if

$$\lim_{c \rightarrow \infty} \mathbf{A}^c = \mathbf{Z},$$

where \mathbf{Z} is the zero matrix. This implies, according to [Davis 1979], that the eigenvalues of \mathbf{A} are, $|\lambda_r| < 1$, $r = 0, 1, \dots, n-1$. Since \mathbf{A} is a circulant matrix, it is diagonalizable by $\mathbf{A} = \mathbf{F}^* \cdot \mathbf{\Lambda} \cdot \mathbf{F}$, where $\mathbf{\Lambda}$ is a diagonal matrix, whose diagonal elements are the eigenvalues of \mathbf{A} , and \mathbf{F}^* is the Fourier matrix with entries given by

$$\mathbf{F}_{jk}^* = \frac{1}{\sqrt{n}} \cdot e^{-\frac{2\pi i j k}{n}}.$$

\mathbf{F}^* is the conjugate transpose of \mathbf{F} . Due to the fact, that \mathbf{A} is a circulant matrix, its eigenvalue vector \vec{v} can be computed by $\sqrt{n} \cdot \mathbf{F}^* \cdot \vec{v}_A = \vec{v}$, where

$$\vec{v}_A = [1 - \lambda, -\lambda/4, 0, \dots, 0, -\lambda/4].$$

Applying that to our case, we get

$$\lambda_r = (1 - \lambda) - \frac{\lambda}{4} \left(\cos r \frac{2\pi}{n} + \cos r(n-1) \frac{2\pi}{n} \right),$$

so $1 - \frac{3}{2}\lambda \leq \lambda_r \leq 1 - \frac{\lambda}{2}$. Hence, in every step the magnitude of $\Delta_i^{[c]}$ decreases which results in the convergence of our proposed data fitting approach for uniform periodic B-spline curves.

In the case when T is uniform open, equation 7 can be rewritten by $\Delta_{[c+1]} = \mathbf{A}^{c+1} \cdot \mathbf{S} \cdot (\mathbf{I} - \mathbf{C}) \cdot \vec{p}$, where $\mathbf{A} = (\mathbf{I} - \lambda \mathbf{S} \cdot \mathbf{C})$. \mathbf{S} is a diagonal matrix where the diagonal elements are defined by

$$S_{ii} = \frac{1}{B_{i,p}(u_i^*)}$$

and $C_{ij} = B_{j,p}(u_i^*)$ which is not circulant. Therefore, the bound on the eigenvalues given above does not apply, due to the end conditions. However, we conducted experiments with different values for λ , and the maximum eigenvalue is always less than one and stays the same independent of the problem size n , indicating convergence. For $\lambda = 1/2$, the eigenvalues of \mathbf{A} range from 0.15 to 0.85.

In the surface case, the two curve methods are interleaved as is done for tensor product nodal interpolation. It is guaranteed to converge given the curve method properties.

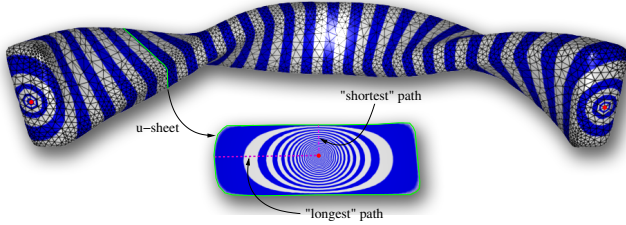


Figure 10: w -parameterizations using min/max points.

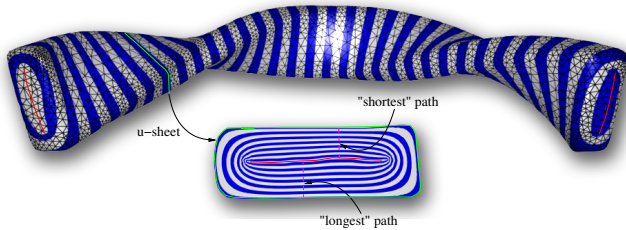


Figure 11: (b) w -parameterizations using min/max paths.

7.2 Simplification

The resulting trivariate B-spline tends to have a high resolution. Therefore, as a post-processing step we apply a data reduction algorithm [Lyche and Morken 1988] to the B-spline representation and iteratively decide how to reduce complexity on the surface or on the attribute data in the interior, by minimizing error.

8 Framework Extension

So far we have assumed that the user chooses two min/max points as the first step in our modeling framework. As discussed above, those two critical points are the end points of a skeleton line through the model. This works well when the lengths of the w -paths of a given slice are similar in length. If isosheets are circular and the skeleton goes through the center, the lengths of the w -paths is equal to the radius of the isosheet. By fixing \mathbf{u}_0 and \mathbf{w}_0 the quadrilateral q_i defined by the points $\mathfrak{X}_3(\mathbf{u}_0, \mathbf{v}_0, \mathbf{w}_0)$ and $\mathfrak{X}_3(\mathbf{u}_0, \mathbf{v}_i + \epsilon, \mathbf{w}_0 + \epsilon)$ for a given small ϵ has the same area for any \mathbf{v}_i .

However, input models exist, where w -path lengths of a given isosheet are different. Refer to Figure 10 for a simple example, where the user chose ν_{min} and ν_{max} . For a constant \mathbf{u} -value we extracted the corresponding isosheet. Isosheets in that model have a rectangular shape. For such a shape a skeletal line is not appropriate: Isoparametric lines in \mathbf{v} towards the exterior are rectangular, but change into circles when approaching the skeleton. This results in distortion: For a given isosheet we can find the shortest and the longest w -path. The quadrilaterals q_i do not have the same area, they are bigger around the longest path, compared to its areas around the shortest path.

Numerical applications such as finite elements desire more uniform element sizes. By modifying the choice of the skeleton the resulting B-spline can be improved. Instead of choosing a single vertex as a critical point, the user chooses a critical path as in Figure 11. The resulting skeleton is a surface. In that case, critical paths suit the rectangular shape better than critical points. The w -paths of a sheet have about the same length, resulting in less stretching and therefore more uniform quadrilaterals.

For a given input mesh, its medial axis and our choice of the skeleton are related. Selecting the medial axis as the skeleton leads to difficulties since it may have small branches which would require splitting the object up into parts which have to be glued together. This would make modeling very difficult. The skeleton we pick is a simplified version of the medial axis. It uses this representation's ability to deal with overhangs and localized features in that simplification. The skeleton we choose can be regarded as approximation of the medial axis. For instance, in Figure 11, our choice of min/max paths yields a skeleton which is a surface similar to the medial axis of the object.

9 Experiments

Except for initial user-required choice of the critical points ν_{min} , ν_{max} and the initial seed to determine the skeleton, our modeling framework runs fully automatically. \mathbf{u}_{TB} takes a few seconds to compute, allowing the user (being aware of simulation parameters) to try out different initial parameterizations. After that first step \mathbf{u}_H and \mathbf{w} are computed, paths are extracted and the trivariate B-spline is generated from that. We implemented the proposed framework on a 16 node cluster, reducing the modeling time for the femur in Figure 2 to about 30 minutes.

We applied isogeometric analysis [Hughes T.J. 2005] in form of Linear elasticity [Hughes 2000] (see Figure 12) to a data reduced version of the resulting B-spline volume. For a more detailed discussion on simulation convergence and error measure, we refer the reader to [Martin et al. 2008].

Isocurves of a harmonic function defined on a smooth surface converge to circles when approaching a critical vertex. In case that the harmonic values are linearly interpolated across a triangle mesh, isocurves are nonplanar " n -gons", where n depends on the number of triangles the respective isocurve crosses. When approaching a

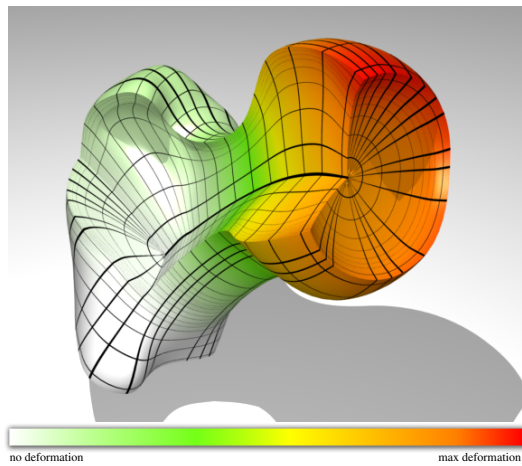


Figure 12: Isogeometric Analysis: Elastostatics applied to the data reduced trivariate B-spline representation of the femur. Load is applied to the head of the femur.

critical point, isocurves are defined only by a few vertices as can be seen in figure 3. In order to improve the parameterization in the regions near the critical points, additional vertices are inserted in the respective regions and retriangulation is applied in these areas. Similarly, inserting additional vertices in the region around the skeleton can help to improve the volumetric parameterization and therefore the quality of the initial structured hexahedral mesh.

10 Conclusion

In this paper we proposed a framework to model a single trivariate B-spline from input genus-0 triangle meshes. The final B-spline was computed using a novel iterative approximation approach, avoiding oscillations observed in B-spline interpolation. We guarantee that the slices defining the B-spline do not overlap and only have degeneracies only along the skeleton. Linear elasticity was applied to the resulting B-spline to demonstrate its practical use. Harmonic functions in three parameters are used to establish an initial parameterization suitable for tensor-product B-splines. This allows to model objects with overhangs such as the femur as in figure 2. We generalized our method to be able to work on simplified medial axis' which extends it to even more complex models. However, modeling B-splines from triangle meshes with a higher genus or bifurcations require an extension of our framework. This is subject to future research, where harmonic functions over the tetrahedral mesh will be used as a Morse function to decompose it into patches where each is approximated with a B-spline. Furthermore, especially in simulations, certain scenarios require higher resolutions in certain regions of the object. Due to the tensor-product nature of B-splines, refining means to increase the resolution in areas where additional control points are not necessary. Therefore, another path of investigation is to convert our resulting B-spline into a T-Spline [Sederberg T.W. 2003] representation, which allows local refinement.

Acknowledgements

This work was supported in part by NSF (CCR0310705) and NSF (CCF0541402). All opinions, findings, conclusions or recommendations expressed in this document are those of the author and do not necessarily reflect the views of the sponsoring agencies.

References

- ALLIEZ, P., COHEN-STEINER, D., DEVILLERS, O., LÉVY, B., AND DESBRUN, M. 2003. Anisotropic polygonal remeshing. vol. 22, 485–493.
- ARBARELLO, E., CORNALBA, M., GRIFFITHS, P., AND HARRIS, J. 1938. Topics in the theory of algebraic curves.
- AXELSSON, O. 1994. *Iterative Solution Methods*. Cambridge University Press, Cambridge.
- BINFORD, T. O. 1971. Visual perception by computer. In *Proceedings of the IEEE Conference on Systems and Controls*.
- CHUANG, J.-H., AHUJA, N., LIN, C.-C., TSAI, C.-H., AND CHEN, C.-H. 2004. A potential-based generalized cylinder representation. *Computers&Graphics* 28, 6 (December), 907–918.
- COHEN, E., RIESENFELD, R. F., AND ELBER, G. 2001. *Geometric modeling with splines: an introduction*. A. K. Peters, Ltd., Natick, MA, USA.
- DAVIS, P. J. 1979. *Circulant Matrices*. John Wiley & Sons, Inc.
- DONG, S., KIRCHER, S., AND GARLAND, M. 2005. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. In *Computer Aided Geometric Design*.
- EDELSBRUNNER, H. 1980. Dynamic data structures for orthogonal intersection queries. Technical Report F59, Inst. Informationsverarbeitung, Tech. Univ. Graz.
- FLOATER, M. S., AND HORMANN, K. 2005. Surface parameterization: a tutorial and survey. In *Advances in multiresolution for geometric modelling*, N. A. Dodgson, M. S. Floater, and M. A. Sabin, Eds. Springer Verlag, 157–186.
- FLOATER, M. 2003. Mean value coordinates. *Computer Aided Design* 20, 1, 19–27.
- FREITAG, L., AND PLASSMANN, P. 1997. Local optimization-based simplicial mesh untangling and improvement. Technical report. Mathematics and Computer Science Division, Argonne National Laboratory.
- GRIMM, C. M., AND HUGHES, J. F. 1995. Modeling surfaces of arbitrary topology using manifolds. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 359–368.
- GU, X., AND YAU, S.-T. 2003. Global conformal surface parameterization. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 127–137.
- HORMANN, K., AND GREINER, G. 2000. Quadrilateral remeshing. In *Proceedings of Vision, Modeling, and Visualization 2000*, infix, Saarbrücken, Germany, B. Girod, G. Greiner, H. Niemann, and H.-P. Seidel, Eds., 153–162.
- HUA, J., HE, Y., AND QIN, H. 2004. Multiresolution heterogeneous solid modeling and visualization using trivariate simplex splines. In *SM '04: Proceedings of the ninth ACM symposium on Solid modeling and applications*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 47–58.
- HUGHES T.J., COTTRELL J.A., B. Y. 2005. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 194, 4135–4195.

- HUGHES, T. J. R. 2000. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover.
- JAILLET, F., SHARIAT, B., AND VANDORPE, D. 1997. Periodic b-spline surface skinning of anatomic shapes. In *9th Canadian Conference in Computational Geometry*.
- LAZARUS, F., AND VERROUST, A. 1999. Level set diagrams of polyhedral objects. In *SMA '99: Proceedings of the fifth ACM symposium on Solid modeling and applications*, ACM Press, New York, NY, USA, 130–140.
- LI, X., GUO, X., WANG, H., HE, Y., GU, X., AND QIN, H. 2007. Harmonic volumetric mapping for solid modeling applications. In *Symposium on Solid and Physical Modeling*, 109–120.
- LOOP, C. 1994. Smooth spline surfaces over irregular meshes. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 303–310.
- LYCHE, T., AND MORKEN, K. 1988. A data reduction strategy for splines with applications to the approximation of functions and data. *IMA J. of Numerical Analysis* 8, 2, 185–208.
- MARTIN, W., AND COHEN, E. 2001. Representation and extraction of volumetric attributes using trivariate splines. In *Symposium on Solid and Physical Modeling*, 234–240.
- MARTIN, T., COHEN, E., AND KIRBY, M. 2008. A comparison between isogeometric analysis versus fem applied to a femur. *to be submitted*.
- MILNOR, J. 1963. Morse theory. In *Annual of Mathematics Studies*, Princeton University Press, Princeton, NJ, vol. 51.
- NI, X., GARLAND, M., AND HART, J. C. 2004. Fair morse functions for extracting the topological structure of a surface mesh. In *Proc. SIGGRAPH*.
- SCHREINER, J., SCHEIDEGGER, C., FLEISHMAN, S., AND SILVA, C. 2006. Direct (re)meshing for efficient surface processing. *Computer Graphics Forum (Proceedings of Eurographics 2006)* 25, 3, 527–536.
- SEDERBERG T.W., ZHENG J., N. A. 2003. T-splines and t-nurccs. In *Proc. SIGGRAPH*.
- SHEFFER, A., PRAUN, E., AND ROSE, K. 2006. Mesh parameterization methods and their applications. *Foundations and Trends in Computer Graphics and Vision* 2, 2.
- SHINAGAWA, Y., KUNII, T. L., AND KERGOSIEN, Y. L. 1991. Surface coding based on morse theory. *IEEE Comput. Graph. Appl.* 11, 5, 66–78.
- SI, H. 2005. Tetgen: A quality tetrahedral mesh generator and three-dimensional delaunay triangulator.
- TONG, Y., ALLIEZ, P., COHEN-STEINER, D., AND DESBRUN, M. 2006. Designing quadrangulations with discrete harmonic forms. In *ACM/EG Symposium on Geometry Processing*.
- VERROUST, A., AND LAZARUS, F. 2000. Extracting skeletal curves from 3d scattered data. *The Visual Computer* 16, 1, 15–25.
- WANG, Y., GU, X., THOMPSON, P. M., AND YAU, S.-T. 2004. 3d harmonic mapping and tetrahedral meshing of brain imaging data. In *Proc. Medical Imaging Computing and Computer Assisted Intervention (MICCAI), St. Malo, France, Sept. 26-30 2004*.
- ZHOU, X., AND LU, J. 2005. Nurbs-based galerkin method and application to skeletal muscle modeling. In *SPM '05: Proceedings of the 2005 ACM symposium on Solid and physical modeling*, ACM Press, New York, NY, USA, 71–78.