

# Fitting Polynomial Volumes to Surface Meshes with Voronoi Squared Distance Minimization

Gilles-Philippe Paillé<sup>1</sup> Pierre Poulin<sup>1</sup> Bruno Lévy<sup>2</sup>

<sup>1</sup> LIGUM, Dept. I.R.O., Université de Montréal

<sup>2</sup> INRIA Nancy - Grand Est

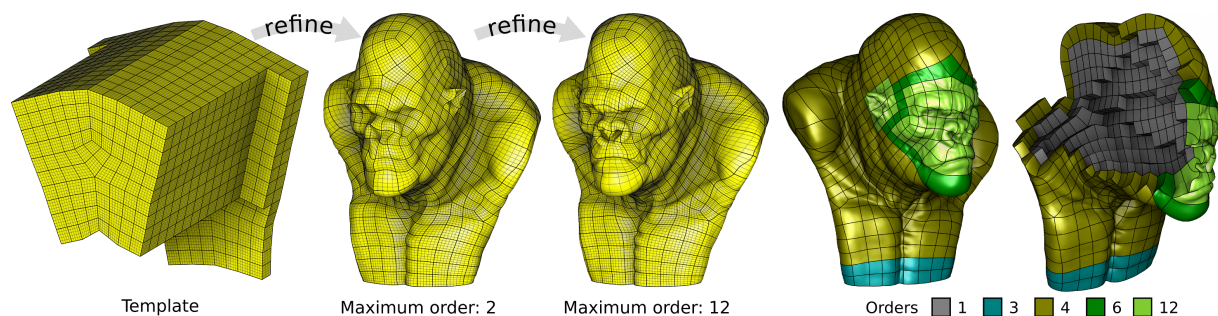


Figure 1: Our method fits an initial volumetric template mesh to a surface mesh using higher-order finite elements. This figure shows the evolution of the initial shape while adaptively increasing the order of each element. This adaptive mesh only stores 6% of all nodes that would be required by a full order-12 finite element mesh.

## Abstract

We propose a method for mapping polynomial volumes. Given a closed surface and an initial template volume grid, our method deforms the template grid by fitting its boundary to the input surface while minimizing a volume distortion criterion. The result is a point-to-point map distorting linear cells into curved ones. Our method is based on several extensions of Voronoi Squared Distance Minimization (VSDM) combined with a higher-order finite element formulation of the deformation energy. This allows us to globally optimize the mapping without prior parameterization. The anisotropic VSDM formulation allows for sharp and semi-sharp features to be implicitly preserved without tagging. We use a hierarchical finite element function basis that selectively adapts to the geometric details. This makes both the method more efficient and the representation more compact. We apply our method to geometric modeling applications in computer-aided design and computer graphics, including mixed-element meshing, mesh optimization, subdivision volume fitting, and shell meshing.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling

## 1. Introduction

Surface design is generally the first step in geometric modeling for computer graphics and computer-aided design. For some applications, such as physics simulation, volume geometry is required. Although a closed surface defines a volume, the surface geometry is often too complex to easily represent volumetric information. Volume meshing is thus required to achieve this goal.

One family of volume meshing techniques is based on creating an “easy-to-mesh” volumetric abstraction of the shape using a topologically equivalent simpler geometry that is fitted to the original surface to infer a volume mesh. This family of techniques (see Section 2) generally requires a parameterization beforehand to guide the fitting process. Current volume parameterization techniques first map the surface, then optimize the volume given the locked surface map.

While existing approaches can *independently* optimize each step of the process, it does not lead to *globally* optimal maps.

We propose to perform the fitting process without a parameterization step, and we allow the boundary of the volume to “flow” on the prescribed surface. This additional degree of freedom makes it possible to further optimize the quality of the mesh elements. Given a target surface mesh and its volumetric abstraction, we solve the variational problem of reproducing the target surface while preserving the shape of its interior, as described in Section 3. The result is a point-to-point map distorting linear cells into curved ones.

The surface fitting is accomplished with Voronoï Squared Distance Minimization (VSDM) [NYL12]. This formulation has the advantage of being implicitly feature-sensitive when using normal anisotropy, and being able to work with both polygon soups and noisy meshes with degenerated elements (see Section 4).

The interior shape is preserved using a barrier metric inspired by large deformations mechanics that enforces bijectivity of the map.

We discretize the problem using a hierarchical higher-order finite element method that offers better approximation properties than linear elements and offers easy control over the order of individual elements, thus saving memory and computation time where geometric precision is not needed. We also introduce integrated Fourier series as a hierarchical basis for higher-order elements and present some of its advantages over traditional bases used in the finite element community.

## 2. Previous Work

We review the previous work on surface fitting (Section 2.1), volume fitting (Section 2.2), and volume mesh optimization (Section 2.3). Since a complete review of the previous work would be well beyond the scope of this paper, we focus on the methods that share some concepts with the one presented herein.

### 2.1. Surface Fitting

Shrink wrapping [KVL99] is a two-step procedure where a template mesh is successively projected on the surface and relaxed using a regularization term. This method was improved using Laplacian energy and the dual mesh to minimize overlaps [YLSL11].

Manifold splines can be fitted to a mesh by first computing an initial parameterization, and then optimizing the location of the control points so that the surface approximates the target mesh while optimizing a regularization term to avoid wiggles in the solution. The template control mesh can be a base complex [EH96], a periodic global parameterization [LRL06], or a polycube-map [WHL\*07].

To avoid the parameterization step, distance to the surface can be approximated without requiring an initial point-to-point correspondence. This procedure can be applied to splines [PL03] and to subdivision surfaces [CWQ\*04, CWQ\*07].

VSDM [NYL12] uses a Voronoï diagram of a sampling of the optimized surface to partition the target mesh to minimize a functional problem over the patches of the partition. A careful choice of the sampling distribution and of the functional energy leads to an implicitly feature-sensitive method, which we target in our work.

### 2.2. Volume Fitting

Volume fitting borrows concepts from surface fitting and extends them to volumes.

As with surfaces, some techniques use an initial parameterization and then optimize the control points of a template spline to fit their corresponding target mesh points defined by the parameterization. The template control mesh can be cylindrical [MCK09] or a generalized polycube-map [LLWQ10, WLL\*12]. The quality of fitting results are highly dependent on this volume parameterization pre-processing. Existing volume parameterization methods [WGY03, LGW\*09, MC10, XHY\*10, PP12a] are highly constrained by the surface parameterization, which has to be precomputed, hence letting few degrees of freedom to optimize the volume. Our method avoids computing a parameterization by optimizing the shape of elements during the fitting stage to obtain globally optimal shape preservation.

When the target is a tetrahedral mesh, one can avoid the parameterization stage by directly considering the tetrahedra as the control mesh [HHQ05]. This corresponds to a change of interpolation function where the piecewise linear function extending the function in the volume is replaced by a spline function without altering the topology. This constraint restricts its application to arbitrary volume fitting.

Volume mesh fitting can also be achieved by computing a signed distance field to the target mesh, and moving surface vertices along the gradient of the distance field while optimizing the shape of the elements using a linear [CS09] or nonlinear [NS10] elasticity formulation. However, the elasticity models used in these methods are not barrier metrics, which means that elements can be inverted if the deformation is too large. Moreover, the surface fitting can get stuck in a local minimum if all vertices lie on a subset of the target surface. We avoid these two problems by using a barrier metric that enforces bijectivity of the map and a symmetric formulation of surface-to-surface distance.

### 2.3. Volume Mesh Optimization

Volume mesh optimization plays an important role in finite element analysis since poor meshing can lead to erroneous

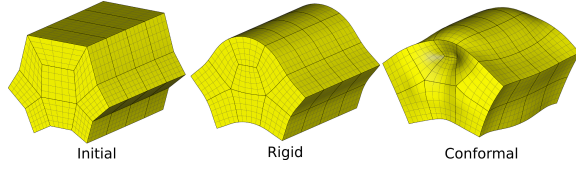


Figure 2: A initial embedding (left) is freely optimized (i.e., no surface constraints) using a rigid metric (center) and a conformal metric (right) using higher-order elements. The conformal metric tends to shrink or expand the volume around singularities.

results and instabilities. In this section, we concentrate on vertex relocation techniques where the topology is preserved during the optimization.

To optimize the shape of the elements, these methods rely on metrics that measure a distance between the deformed element and an ideal shape. This notion of distance is dependent on the metric used. We refer the reader to Knupp [Knu00] for a review of the different metrics used in the meshing community.

We highlight two of the most popular metrics, namely the condition number [FK99] and the mean ratio [TRBS98]. They are barrier metrics, which means that a valid mesh will stay valid during the optimization thanks to the infinite energy of an element passing from valid state to degenerate one. These metrics optimize conformality of the transformation but tend to create large and counterintuitive deformations in the vicinity of singularities as shown in Figure 2.

Some shape matching techniques [MHTG05, BDS\*12] use only vertex differences to compute the distance to the ideal shape. A similar approach is the combinatorial Laplacian discretization, which tries to relocate each vertex in the center of its one-ring. These techniques have the advantage of being easy to implement and fast to compute, but they can generate inverted elements. We conjecture that every metric that has a finite energy for degenerate element configurations is not guaranteed to give valid meshes in output.

The metric presented in this paper is a barrier metric inspired from large deformation mechanics that optimizes rigid transformation, and that behaves well in the vicinity of singularities.

### 3. Volume Shape Fitting

Volume shape fitting aims at deforming a shape in such a way that its boundary matches the boundary of a target shape while preserving the smoothness of the mapping. To achieve this, we define an energy functional from these two goals and solve it with a variational finite element method.

Given a parameterization domain  $\Omega$ , a piecewise differentiable embedding  $f : \Omega \rightarrow \mathbb{R}^3$ , and a target surface  $\mathcal{T}$ , we define the fitting energy functional as

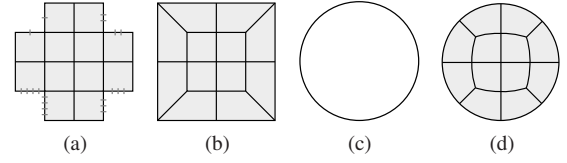


Figure 3: A 2D example of (a) a parameterization domain  $\Omega$ , (b) an initial embedding  $f_0(\Omega)$ , (c) the target surface, and (d) the final map  $f(\Omega)$ .

$$\mathcal{E}(f, \mathcal{T}) = \mathcal{E}_S(f, \mathcal{T}) + \lambda \mathcal{E}_\Omega(f) \quad (1)$$

where  $\mathcal{S} = f(\partial\Omega)$  is the surface of the volume to be fitted,  $\mathcal{E}_S$  is the surface energy functional (see Section 3.3),  $\mathcal{E}_\Omega$  is the volume energy functional (see Section 3.1), and  $\lambda$  is a regularization factor.

The parameterization domain is never computed and is only used as a reference for the definition of  $f$ . An initial embedding is always given as a starting point of our optimization (see Figure 3).

#### 3.1. Volume Energy Functional

The purpose of the volume energy functional is to keep  $f$  smooth and bijective without requiring any knowledge of the target surface. In other words, this functional measures, at each point of  $\Omega$ , how far the Jacobian matrix  $\mathbf{J}_f$  is from an ideal transformation.

Let  $|\Omega|$  be the volume of  $\Omega$  and  $\epsilon(\mathbf{J}_f)$  a potential energy density that measures the distortion of the mapping. The potential energy  $\epsilon(\mathbf{J}_f)$  will be explicated later. We define the volume energy functional as

$$\mathcal{E}_\Omega(f) = \frac{1}{|\Omega|} \int_\Omega \epsilon(\mathbf{J}_f) d\Omega. \quad (2)$$

The potential energy density should penalize degenerate Jacobian matrices such that it is impossible to create inversions in the process. Using a physical analogy, this corresponds to requiring an infinite amount of energy to compress an object such that its volume vanishes. We take this analogy literally by borrowing the concept of *Hencky* strain, or *true* strain, for large deformations in continuum mechanics [XC02]. The *true* strain matrix  $\mathbf{H}$  is defined as

$$\mathbf{H} = \frac{1}{2} \ln(\mathbf{J}_f^T \mathbf{J}_f)$$

where the logarithm of a symmetric matrix  $\mathbf{S}$  is computed using its diagonalization  $\ln(\mathbf{S}) = \ln(\mathbf{Q}\mathbf{D}\mathbf{Q}^T) = \mathbf{Q}\ln(\mathbf{D})\mathbf{Q}^T$ , and where the logarithm of a diagonal matrix is computed component-wise.

Given this strain measure, we can infer a potential energy density

$$\varepsilon(\mathbf{J}_f) = \frac{1}{2} \|\mathbf{H}\|^2.$$

The geometrical interpretation of this potential energy density can be made using the Lie group of symmetric positive-definite (SPD) matrices [Moa05]. From the point of view of Lie groups, the norm of the logarithm of a SPD matrix is the geodesic distance to the identity matrix. In our context, the norm represents the geodesic distance to the nearest rigid transformation.

### Volume Energy First Variation

A local minimum of a functional is found by looking at its first variation, which is the variational equivalent of the first derivative in differential calculus. The first variation is computed by adding a test function  $g$  in the manner of a directional derivative [EPT\*07]. If the first variation vanishes in all directions, i.e., for all  $g$ , then  $f$  is a minimum of the energy functional. It is noted  $\delta_g \mathcal{E}_\Omega(f) = 0$  and computed as

$$\begin{aligned} \delta_g \mathcal{E}_\Omega(f) &= \frac{d}{d\tau} \mathcal{E}_\Omega(f + \tau g)|_{\tau=0} \\ &= \frac{1}{|\Omega|} \int_\Omega \frac{d}{d\tau} \varepsilon(\mathbf{J}_f + \tau \mathbf{J}_g)|_{\tau=0} d\Omega \\ &= \frac{1}{|\Omega|} \int_\Omega \left\langle \frac{d\varepsilon}{d\mathbf{J}_f}, \mathbf{J}_g \right\rangle d\Omega \end{aligned} \quad (3)$$

where we use the matrix chain rule [PP12b] with the notation  $\langle \mathbf{A}, \mathbf{B} \rangle = \text{tr}(\mathbf{A}\mathbf{B}^T)$  and the potential energy density gradient is

$$\frac{d\varepsilon}{d\mathbf{J}_f} = \mathbf{J}_f^{-T} \mathbf{H}.$$

Details of these derivations can be found in Appendix A.

We do not need the second variation since we use a Quasi-Newton method for the optimization, that only needs first order derivatives (see Section 3.4). Saddle points and local maxima are avoided by enforcing Wolfe conditions [LN89, NW06].

### 3.2. Finite Element Formulation

We formulate the solution of this variational problem with a hierarchical higher-order element method. The domain is decomposed in a set of vertices  $\mathcal{V}$ , edges  $\mathcal{E}$ , facets  $\mathcal{F}$ , and cells  $\mathcal{C}$ , while the mapping  $f$  is defined with a set of nodes  $\mathcal{N}$  that are the degrees of freedom of  $f$ . On linear elements, nodes are located at vertices and represent the value of  $f$  sampled at each vertex. On higher-order elements, more nodes are added on edges, faces, and inside the volume, and these

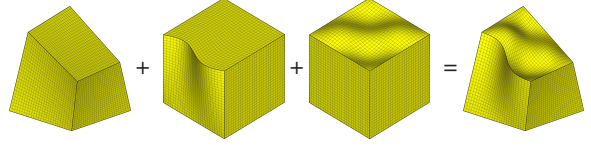


Figure 4: A high-order mapping  $f$  of a hexahedron using a linear map defined by the vertex nodes and some additional displacements defined by edge and facet nodes.

nodes represent additional displacements as illustrated in Figure 4.

Each type of element is described by a canonical geometry that facilitates computations, a reference geometry that represents the ideal shape of the element, an interpolation function that extends the node function values to the interior of the canonical volume, and some quadrature rule to accurately evaluate the integral on the domain of the element.

The canonical geometry of the hexahedron, the prism, and the tetrahedron are respectively defined by  $[0, 1]^3$ ,  $\{(u, v, w) \in [0, 1]^3 | u + v \leq 1\}$ , and  $\{(u, v, w) \in [0, 1]^3 | u + v + w \leq 1\}$ . The transformation from the canonical element to the reference element is expressed by a matrix  $\mathbf{A}$  defined by

$$\begin{aligned} \mathbf{A}_{hex} &= \mathbf{I} \\ \mathbf{A}_{prism} &= \begin{pmatrix} 1 & \frac{1}{2} & 0 \\ 0 & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ \mathbf{A}_{tet} &= \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{6} \\ 0 & 0 & \frac{\sqrt{2}}{\sqrt{3}} \end{pmatrix}. \end{aligned} \quad (4)$$

An interpolation function  $h(\mathbf{u})$  is defined for each element in its canonical space using the values of its nodes  $\{\mathbf{n}_i\}$ . For a hexahedron of order  $p$ , we decompose  $h$  into a linear interpolation using vertex nodes  $\{\mathbf{v}_i\}$ , extended with nodes on edges  $\{\mathbf{e}_{ij}\}$ , facets  $\{\mathbf{f}_{ijk}\}$ , and cells  $\{\mathbf{c}_{ijkl}\}$

$$\begin{aligned} h(\mathbf{u}) &= \sum_{i=1}^{(p+1)^3} \mathbf{n}_i N_i(\mathbf{u}) \\ h(\mathbf{u}) &= \sum_{i=1}^8 \mathbf{v}_i V_i(\mathbf{u}) + \sum_{i=1}^{12} \sum_{j=1}^{p-1} \mathbf{e}_{ij} E_{ij}(\mathbf{u}) \\ &\quad + \sum_{i=1}^6 \sum_{j=1}^{p-1} \sum_{k=1}^{p-1} \mathbf{f}_{ijk} F_{ijk}(\mathbf{u}) + \sum_{i=1}^{p-1} \sum_{j=1}^{p-1} \sum_{k=1}^{p-1} \mathbf{c}_{ijk} C_{ijk}(\mathbf{u}) \end{aligned} \quad (5)$$

where  $\{\mathbf{n}_i\} = \{\mathbf{v}_i\} \cup \{\mathbf{e}_{ij}\} \cup \{\mathbf{f}_{ijk}\} \cup \{\mathbf{c}_{ijkl}\}$ ,  $V_i$  is the usual linear bases, and  $E_{ij}$ ,  $F_{ijk}$ , and  $C_{ijk}$  are higher-order nodes. Edge, facet, and cell bases vanish on borders of the sub-element on which they are defined. For example, we have



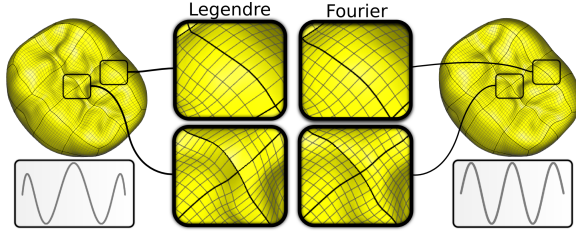


Figure 5: Effects of non-uniformity of integrated Legendre polynomials compared to integrated Fourier polynomials for the tooth model using elements of order 6, along with the plot of their respective sixth order polynomial. Note the irregularity in the mesh near edges (top) and vertices (bottom) for the integrated Legendre basis (left).

$$\begin{aligned} E_{1j}(u, v, w) &= (1-u)(1-v)\phi_j(w) \\ F_{1jk}(u, v, w) &= (1-u)\phi_j(v)\phi_k(w) \\ C_{ijk}(u, v, w) &= \phi_i(u)\phi_j(v)\phi_k(w) \end{aligned}$$

where  $\{\phi_i\}$  is a hierarchical functional basis. At this point, we choose  $i$  as the representative index for  $\{\phi_*\}$ . In this paper, we use the modified Fourier series [IN08] instead of the classical Fourier series, because they avoid the Gibbs phenomenon at the endpoints and exhibit better convergence properties for nonperiodic functions as demonstrated in [IN08]. Over the range  $[0, 1]$ , the basis is defined as  $\{1, \sqrt{2}\cos(i\pi x)\}$  for  $i \geq 1$ . We integrate this set in order to obtain a linear function along with higher-order modes vanishing at  $x = 0$  and  $x = 1$ ,

$$\phi_i(x) = \frac{\sqrt{2}\sin(i\pi x)}{i\pi}.$$

We could have also used the integrated Legendre polynomials, which are often used by the finite element community [SB11]. However, we found that the nonuniformity of the oscillations of these polynomials can sometimes be observed in the final geometry (see Figure 5).

Note that nodes are shared between adjacent elements. For this reason, one needs to carefully (i.e., coherently) define the orientation of each sub-element such that interpolation functions are conformal across elements.

Let  $\mathbf{x}$  be a point in the reference element. We define the volumetric deformation  $f$  as a function of the element mapping matrix  $\mathbf{A}$  (Equation (4)) and the nodal interpolation function  $h$  (Equation (5)) as

$$f(\mathbf{x}) = h(\mathbf{A}^{-1}\mathbf{x})$$

and its Jacobian matrix  $\mathbf{J}_f$  is then given by

$$\mathbf{J}_f = \mathbf{J}_h \mathbf{A}^{-T}.$$

### Volume Energy Integral Approximation

The finite element formulation permits to evaluate the energy integral on a per-element basis. We approximate the integral over an element using a quadrature rule that samples the integration domain given a set of position/weight pairs  $\{\mathbf{p}_i, w_i\}$ . Each type of element of each order has its own quadrature rule. Over a domain  $D$ , an integral is approximated by

$$\int_D f(\mathbf{x}) d\mathbf{x} \approx \sum_i f(\mathbf{p}_i) w_i. \quad (6)$$

Choosing a quadrature rule also depends on the functional basis used. For trigonometric polynomials, the trapezoidal rule is a simple choice and can be easily computed for every type of element. We use  $2p - 1$  samples per dimension for order  $p > 1$ , and 2 samples per dimension for linear elements. In Figure 5 (left), where integrated Legendre polynomials are used, a Gauss-Lobatto quadrature [SB11] is a more appropriate choice.

We now have all the ingredients that we need to discretize Equation (3). For an element  $t \in \mathcal{C}$ , we have

$$\begin{aligned} \mathcal{E}_{\Omega,t} &= \int_t \varepsilon(\mathbf{J}_f) d\mathbf{x} \\ &\approx \sum_i \varepsilon(\mathbf{J}_h|_{\mathbf{p}_i} \mathbf{A}^{-T}) w_i |\mathbf{A}| \end{aligned} \quad (7)$$

where position/weight pairs  $\{\mathbf{p}_i, w_i\}$  are relative to the canonical element.

Inside an element  $t \in \mathcal{C}$ , the first variation is computed by differentiating by the  $k$ th component of each node  $f_j$  of element  $t$

$$\frac{\partial \mathcal{E}_{\Omega,t}}{\partial f_{jk}} \approx \sum_i \left\langle \frac{d\varepsilon}{d\mathbf{J}_f}, \frac{\partial \mathbf{J}_h}{\partial f_{jk}} \mathbf{A}^{-T} \right\rangle w_i |\mathbf{A}|. \quad (8)$$

### 3.3. Surface Energy Functional

The surface energy functional measures how close the volume surface  $\mathcal{S}$  is from the target surface  $\mathcal{T}$ , regardless of the smoothness of  $f$ . We usually measure this distance by integrating the square of the normal distance of every point from  $\mathcal{S}$  to the nearest point  $\Pi_{\mathcal{T}}(\mathbf{x})$  on  $\mathcal{T}$

$$\mathcal{F}_{\mathcal{T} \rightarrow \mathcal{S}} = \int_{\mathcal{S}} \|\mathbf{n}_{\mathcal{T}}(\mathbf{x}) \cdot (\mathbf{x} - \Pi_{\mathcal{T}}(\mathbf{x}))\|^2 d\mathbf{x}$$

where  $\mathbf{n}_{\mathcal{T}}(\mathbf{x})$  is the normal at  $\Pi_{\mathcal{T}}(\mathbf{x})$ .

This formulation has the disadvantage of vanishing whenever  $\mathcal{S}$  matches a subset of  $\mathcal{T}$ . To correct this situation,

we use a symmetrized version as proposed by Nivoliens et al. [NYL12]. We also add a normalization factor to render the functional dimensionless

$$\mathcal{E}_S(f, \mathcal{T}) = \frac{1}{AL^2} (\mathcal{F}_{S \rightarrow \mathcal{T}} + \mathcal{F}_{\mathcal{T} \rightarrow S}) \quad (9)$$

where  $A$  and  $V$  are respectively the area of  $\mathcal{T}$  and the volume enclosed by  $\mathcal{T}$ , and  $L = V/A$  is an approximation of the average local feature size of  $\mathcal{T}$ .

The differences between methods minimizing this functional is the way it is discretized. For  $\mathcal{F}_{\mathcal{T} \rightarrow S}$ , VSDM does sample  $S$  and partition  $\mathcal{T}$  using the restricted Voronoï diagram [ES94] induced by this sampling. For each restricted cell, we integrate the squared distance from the surface to the sample associated to the cell. This ensures that the minimization is performed over the entire surface.  $\mathcal{F}_{S \rightarrow \mathcal{T}}$  is discretized the same way, although the sampling pattern and the gradient computation differ. We refer the reader to Nivoliens et al. [NYL12] for implementation details.

While no specific sampling prescription is given in the original article about VSDM, we found that best results are obtained when placing a sample at the barycenter of each facet of  $\mathcal{T}$ , and using the sampling of  $S$  naturally induced by the quadrature rule of Equation (6) when restricted to the surface.

### 3.4. Optimization

The fitting process works as follows. The user gives a target surface, represented by a triangle mesh, and an initial embedding of the parameterization domain, represented by a mixed finite element mesh containing hexahedra, prisms and tetrahedra. Given a regularization factor, the total energy functional of Equation (2) is minimized using a quasi-Newton method, namely L-BFGS [LN89, NW06]. The regularization factor plays an important role in the conditioning of the system. We use  $\lambda = 50$  for an initial embedding far from the target geometry, and  $\lambda = 0.05$  for the final close fitting. Smaller incremental steps can be taken when the template is not close to the target shape (e.g., for the King Kong and Hanger models). For higher-order fitting, the shape is first roughly fitted using linear elements only, and gradually enhanced with higher-order nodes (see Algorithm 1).

---

#### Algorithm 1 Incremental fitting

---

**Input:** 1) A closed surface  $\mathcal{T}$   
 2) An initial template volumetric mesh  $\mathcal{M}$   
**Output:**  $\mathcal{M}$  fitted to  $S$   
 Fit  $\mathcal{T}$  to  $\mathcal{M}$  with  $\lambda = 50$  (see Algo. 2)  $\triangleright$  Far-fitting phase  
**while** true **do**  $\triangleright$  Close-fitting phase  
   Increase finite element order of  $\mathcal{M}$   
   Fit  $\mathcal{T}$  to  $\mathcal{M}$  with  $\lambda = 0.05$  (see Algo. 2)

---

Model	Order	$ \mathcal{C} $	Time	SJ
Pensatore	16	7	76.03	0.955
Sculpture	4	30	0.44	0.990
Block	4	50	0.39	0.980
Tooth	8	68	11.21	0.976
Bone	4	79	0.66	0.959
Double torus	4	113	1.23	0.967
Hanger	2	212	0.72	0.983
Hand	4	230	1.26	0.957
Joint	2	302	0.52	0.993
Mudface	12	643	67.72	0.980
King Kong	12	2492	187.54	0.963

Table 1: Results for some models shown in Figures 6 and 7. The table shows the element order, the number of elements, optimization time in minutes, and the statistics of the volume mapping with the scaled Jacobian (SJ).

For each iteration of the minimization process, the functional of Equation (2) and its gradient need to be evaluated. Integrating the surface part of the function is described in Nivoliens et al. [NYL12]. The volumetric part is computed using Equations (7) and (8). The optimization stops when the norm of the gradient is below a threshold. We use  $10^{-8}$  in our implementation. For stability purpose, we define the energy to be infinite and its gradient to be 0 when  $|\mathbf{J}_f| < 0$ . This prevents L-BFGS from overshooting vertices. The process is illustrated in Algorithm 2.

---

#### Algorithm 2 Fitting process

---

**Input:** 1) A closed surface  $\mathcal{T}$   
 2) An initial template volumetric mesh  $\mathcal{M}$   
 3) A regularization factor  $\lambda$   
**Output:**  $\mathcal{M}$  fitted to  $S$   
**for** each iterate  $x^{(k)}$  of L-BFGS **do**  
   // Compute  $\mathcal{E} = \mathcal{E}_S + \lambda \mathcal{E}_\Omega$  and  $\nabla \mathcal{E}$  (Eq. (1))  
   Compute  $\mathcal{E}_\Omega(f)(x^{(k)})$  and  $\nabla \mathcal{E}_\Omega$  (Eqs. (7), (8))  
   Compute  $\mathcal{E}_S(f, \mathcal{T})(x^{(k)})$  and  $\nabla \mathcal{E}_S$  (Eq. (9), [NYL12])  
   Assemble  $\mathcal{E}$  and  $\nabla \mathcal{E}$  and return them to L-BFGS

---

The method is implemented in C++ in the *Graphite* software [Gra13] and all tests were conducted on a Intel® Core™ i7 930 processor with 12GB of memory and without any GPU acceleration. We took advantage of the multi-threading capacity on the multi-core architecture using OpenMP. The optimization time depends on the complexity of the target mesh and the parameterization domain, and on the order the finite element formulation. See Table 1 for more details.

### 4. Results

We now present results of some polynomial volume fitting applications in geometry processing.

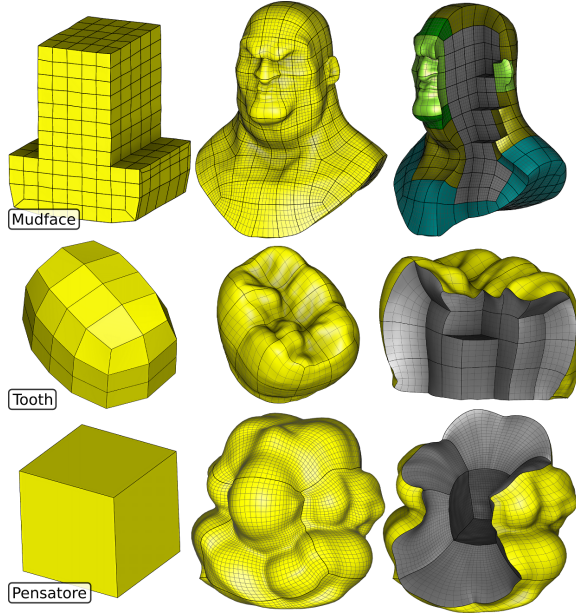


Figure 6: Higher-order finite element meshes (center) generated by our method, along with their initial templates (left) and a cutaway view of each mesh (right). The Mudface model (top) uses the same color scheme as the King Kong model in Figure 1. See Table 1 for statistics.

#### 4.1. Mixed-element Meshing

Mixed-element meshing of the volume enclosed by the target is a natural consequence of the fitting process (see Figures 6 and 7). All-hex meshes are created by restricting the base mesh to contain hexahedral elements only. Experimental evidence of energy convergence is shown in Figure 8 for two representative models.

VSDM is implicitly feature-sensitive, thanks to the normal distance minimization, which prevents the user from having to explicitly tag features. Our method can also fit volume meshes to surface meshes with noisy data, disconnected triangles, and degenerated triangles (see Figure 9). It is also possible to mesh a point cloud (see Figure 10) using inverse fitting only, taking the point cloud as a sampling of the target surface.

While the bijectivity of the map is enforced by the barrier metric that we use, it is not guaranteed with the higher-order function basis and quadratures that we use: while the Jacobian is guaranteed to be positive at the quadrature samples, it is still an open problem to prove that it is positive everywhere. Johnen et al. [JRG13] show that it is possible to define bounds on the Jacobian for standard polynomial elements, changing the element basis to Bézier interpolation, but this technique does not directly apply to trigonometric polynomials. Nevertheless, we did not find any inverted element, i.e., with negative Jacobian, in our empirical results

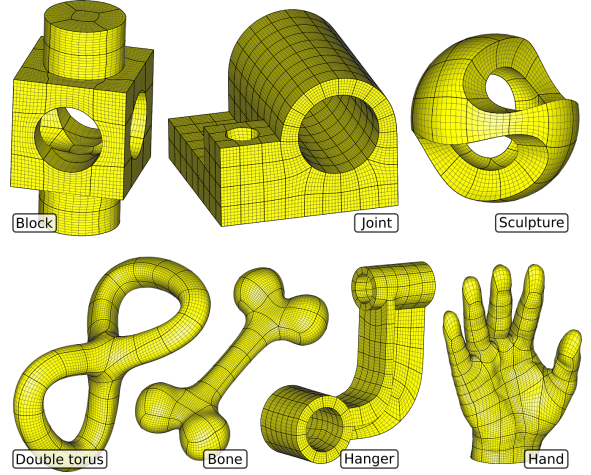


Figure 7: More results from Table 1. Most of the meshes benefit from the implicit feature-preservation property of the anisotropic VSDM formulation.

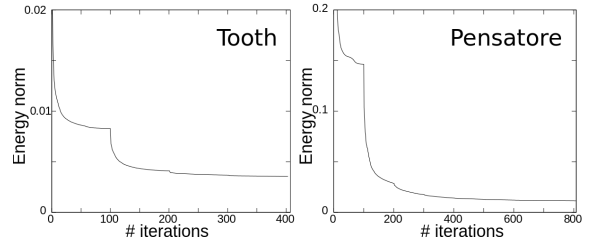


Figure 8: Energy evolution of the close-fitting phase for two models. Discontinuities in the plots are caused by hierarchical function refinement.

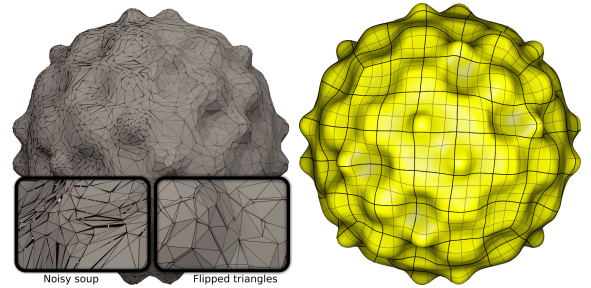


Figure 9: Our method can handle degenerated meshes issued from noisy data, polygon soups, and flipped triangles. We show the target mesh (left) and the fitted volumetric mesh (right) using elements of order 8. Note the smoothness of the result despite the asymmetric degeneracies.



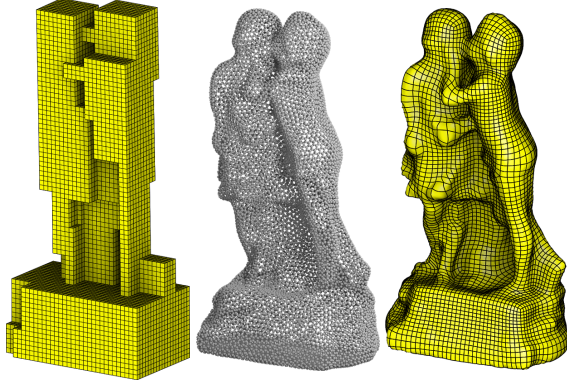


Figure 10: The abstraction (left), taken from Gregson et al. [GSZ11], fits a point cloud (middle) using elements of order 2 (right) with inverse fitting.

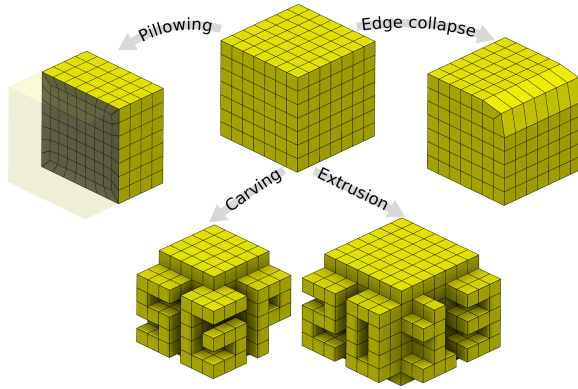


Figure 11: Template creation from a uniform grid using simple mesh interactive operations.

shown in Figures 6 and 7 for a dense set of randomly chosen samples inside each element.

Template meshes were created manually using a combination of simple mesh editing operations: uniform grid creation, pillowing, edge collapse, extrusion, and carving (see Figure 11). Mesh pillowing and edge collapsing are especially useful for pushing singularities inside the volume when approximating curved geometry. Extrusion can also be performed from a surface mesh. For example, the template of the Block model (see Figure 7) is created by extruding a surface mesh followed by carving. Automatic polycube creation techniques could also be used for this task [LJFW08, WJH\*08, HWFQ09, GSZ11].

#### 4.2. Mesh Optimization

The fitting process can also be used to optimize existing volume mesh (see Figure 12). We first extract the boundary of the volume mesh, then perform the fitting procedure of the

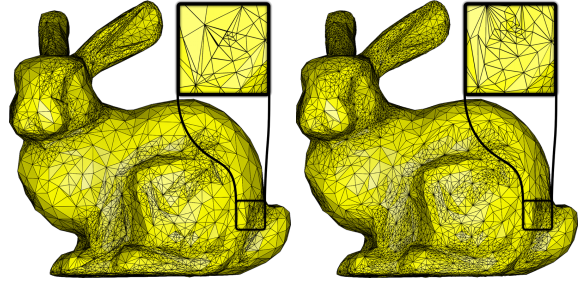


Figure 12: The original tetrahedral mesh of the bunny (left) contains inverted and degenerated elements. After optimization (right), the mesh does not contain any degeneracy.

volume over its own boundary. This will allow the volume energy functional to optimize elements while the surface energy functional preserves the overall boundary shape.

The input mesh can be any valid mesh, and to some extent, even an invalid mesh. Experiments have shown that inverted elements can be rectified by the L-BFGS line search when an element is near validity (see Figure 12).

#### 4.3. Subdivision Volumes

Fitting subdivision volumes is also a natural extension of our method. We first note that vertices  $\mathcal{V}_c$  of the control mesh are related to the vertices  $\mathcal{V}_s$  of the subdivided mesh by a subdivision matrix  $\mathbf{S}$  of dimensions  $|\mathcal{V}_s| \times |\mathcal{V}_c|$

$$\mathcal{V}_s = \mathbf{S}\mathcal{V}_c. \quad (10)$$

The optimization process works as follows. The vertices of the control mesh are given to L-BFGS as the degrees of freedom. For each iteration, we recover the subdivided vertex positions by applying Equation (10). We then compute the total energy functional and its gradient  $\mathbf{G}_s$  using the subdivided mesh. Finally, we recover the control mesh gradient  $\mathbf{G}_c$  that is given back to L-BFGS with

$$\mathbf{G}_c = \mathbf{S}^T \mathbf{G}_s.$$

In our implementation, we use the multi-linear averaging approximation scheme [BSWX02]. This scheme preserves features during the subdivision process by modifying  $\mathbf{S}$  to apply averaging using only vertices that are on the sub-manifold of a feature.

For the example (see Figure 13 (left)), we started with a control mesh with tagged sharp edges (in blue), and applied the optimization on this control mesh using the first subdivision as the finite element mesh. We then recursively applied the same procedure to the resulting subdivided mesh, until we achieved desired accuracy (4th subdivision for this example).



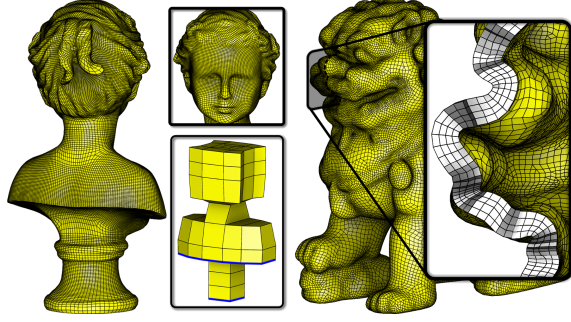


Figure 13: Some extensions of our technique for fitting subdivision volumes (left) and creating shell meshes (right). Note that the apparent jaggedness in the planar cut view of the shell mesh (white zone) is a visual artifact caused by sudden changes in rows of cells being displayed.

#### 4.4. Shell Meshes

We can use our volume energy optimization to create a shell mesh of a given surface mesh.

Given a number of layers  $l$ , we first create a tiny shell around the surface by simple extrusion about the normal of each vertex, avoiding self-intersection. Then, given an offset distance  $d$ , we optimize each element such that its thickness along its surface normal is equal to  $d/l$  and its tangential shape is equal to the polygon that it belongs to. This optimal shape is specified using the reference geometry matrix  $\mathbf{A}$ .

Figure 13 (right) shows an example of a shell mesh created by extruding the surface both inside and outside, each using two layers of elements.

#### 5. Discussion and Conclusion

We presented a method for fitting higher-order finite element meshes given a target surface mesh. Sharp and semi-sharp features are implicitly respected given the VSDM anisotropic formulation. Our logarithmic metric enforces bijectivity of the resulting elements. Integrated Fourier series for hierarchical finite element basis offer good convergence properties while maintaining uniformity of the map.

Some limitations are subject to future work. The nonlinear behavior of the higher-order elements makes it very hard to ensure bijectivity. While we have guarantees for a discrete set of samples, it remains an open problem to create bijective elements for arbitrary functional bases. Our method is subject to the same failure cases than surface VSDM (see [NYL12]). Features that are too close to each other can lead to errors in the surface fitting when the sampling is not dense enough. These errors can prevent global bijectivity of the surface map, which in turn prevent volume global bijectivity.

The given volumetric abstraction is currently taken as is,

without questioning its geometrical and topological validity. For abstractions very different from the actual surface geometry, large distortions can appear in the result. Incompatible topologies are also problematic since the mapping is theoretically impossible without topology modifications. Future research could alleviate these restrictions by dynamically creating, removing, or modifying elements of the abstraction.

Our implementation does not support nonconformal meshes. It could be interesting to support T-meshes using a Mortar element method [MMP89]. It could also be interesting to test the finite element basis and the logarithmic metric in the context of nonphysically-based computer animations.

#### Acknowledgements

Gilles-Philippe Paillé acknowledges financial support from FQRNT, and Pierre Poulin, from NSERC and GRAND. Bruno Lévy is partly supported by the European Research Council (GOODSHAPE ERC-StG-205693) and the ANR (MORPHO and BECASIM).

#### Appendix A: Logarithmic energy gradient

Let  $\epsilon = \frac{1}{2} \|\mathbf{H}\|^2$ ,  $\mathbf{H} = \frac{1}{2} \ln(\mathbf{U})$ , and  $\mathbf{U} = \mathbf{J}^T \mathbf{J}$ . Using trace identities [PP12b] and the symmetry of  $\mathbf{H}$ , we have

$$\partial \epsilon = \text{tr}(\mathbf{H} \partial \mathbf{H}) = \text{tr} \left( \frac{1}{2} \mathbf{H} \mathbf{U}^{-1} \partial \mathbf{U} \right) = \text{tr} \left( \mathbf{H} \mathbf{U}^{-1} \mathbf{J}^T \partial \mathbf{J} \right)$$

$$\frac{\partial \epsilon}{\partial \mathbf{J}} = \left( \mathbf{H} (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \right)^T = \mathbf{J}^{-T} \mathbf{H}.$$

#### References

- [BDS\*12] BOUAZIZ S., DEUSS M., SCHWARTZBURG Y., WEISE T., PAULY M.: Shape-up: Shaping discrete geometry with projections. *Computer Graphics Forum* 31, 5 (2012), 1657–1667. 3
- [BSWX02] BAJAJ C., SCHAEFER S., WARREN J., XU G.: A subdivision scheme for hexahedral meshes. *The Visual Computer* 18 (2002), 343–356. 8
- [CS09] CHOI J., SZYMCAK A.: Fitting solid meshes to animated surfaces using linear elasticity. *ACM Transactions on Graphics* 28, 1 (2009), 6:1–6:10. 2
- [CWQ\*04] CHENG K.-S. D., WANG W., QIN H., WONG K.-Y. K., YANG H., LIU Y.: Fitting subdivision surfaces to unorganized point data using SDM. In *Proceedings of the Computer Graphics and Applications, 12th Pacific Conference* (2004), PG '04, IEEE Computer Society, pp. 16–24. 2
- [CWQ\*07] CHENG K.-S. D., WANG W., QIN H., WONG K.-Y. K., YANG H., LIU Y.: Design and analysis of optimization methods for subdivision surface fitting. *IEEE Transactions on Visualization and Computer Graphics* 13, 5 (2007), 878–890. 2
- [EH96] ECK M., HOPPE H.: Automatic reconstruction of B-spline surfaces of arbitrary topological type. In *Proceedings of SIGGRAPH* (1996), SIGGRAPH '96, ACM, pp. 325–334. 2
- [EPT\*07] ECKSTEIN I., PONS J.-P., TONG Y., KUO C.-C. J., DESBRUN M.: Generalized surface flows for mesh processing. In

- Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (2007), SGP '07, Eurographics Association, pp. 183–192. 4
- [ES94] EDELSBRUNNER H., SHAH N. R.: Triangulating topological spaces. In *Proceedings of the tenth Annual Symposium on Computational Geometry* (1994), SCG '94, ACM, pp. 285–292. 6
- [FK99] FREITAG L. A., KNUPP P. M.: Tetrahedral element shape optimization via the jacobian determinant and condition number. In *Proceedings of the 8th International Meshing Roundtable* (1999), pp. 247–258. 3
- [Gra13] GRAPHITE, Feb. 2013. <http://alice.loria.fr>. 6
- [GSZ11] GREGSON J., SHEFFER A., ZHANG E.: All-hex mesh generation via volumetric polycube deformation. *Computer Graphics Forum* 30, 5 (2011), 1407–1416. 8
- [HHQ05] HUA J., HE Y., QIN H.: Trivariate simplex splines for inhomogeneous solid modeling in engineering design. *ASME Transactions: Journal of Computing and Information Science in Engineering* 5, 2 (2005), 149–157. 2
- [HWFQ09] HE Y., WANG H., FU C.-W., QIN H.: A divide-and-conquer approach for automatic polycube map construction. *Computers & Graphics* 33, 3 (2009), 369–380. 8
- [IN08] ISERLES A., NØRSETT S. P.: From high oscillation to rapid approximation I: Modified Fourier expansions. *IMA Journal of Numerical Analysis* 28, 4 (2008), 862–887. 5
- [JRG13] JOHNNEN A., REMACLE J.-F., GEUZAIN C.: Geometrical validity of curvilinear finite elements. *Journal of Computational Physics* 233 (2013), 359–372. 7
- [Knu00] KNUPP P. M.: Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part II - A framework for volume mesh optimization and the condition number of the Jacobian matrix. *International Journal for Numerical Methods in Engineering* 38 (2000), 1165–1185. 3
- [KVL99] KOBELT L. P., VORSATZ J., LABSIK U.: A shrink wrapping approach to remeshing polygonal surfaces. *Computer Graphics Forum* 18, 3 (1999), 119–130. 2
- [LGW\*09] LI X., GUO X., WANG H., HE Y., GU X. D., QIN H.: Meshless harmonic volumetric mapping using fundamental solution methods. *IEEE Transactions on Automation Science and Engineering* 6, 3 (2009), 409–422. 2
- [LJFW08] LIN J., JIN X., FAN Z., WANG C. C.: Automatic polycube-maps. In *Advances in Geometric Modeling and Processing*, Chen F., Jüttler B., (Eds.), vol. 4975 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2008, pp. 3–16. 8
- [LLWQ10] LI B., LI X., WANG K., QIN H.: Generalized polycube trivariate splines. In *Proceedings of the 2010 Shape Modeling International Conference* (2010), SMI '10, IEEE Computer Society, pp. 261–265. 2
- [LN89] LIU D. C., NOCEDAL J.: On the limited memory BFGS method for large scale optimization. *Mathematical Programming: Series A and B* 45, 3 (1989), 503–528. 4, 6
- [LRL06] LI W.-C., RAY N., LÉVY B.: Automatic and interactive mesh to T-spline conversion. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing* (2006), SGP '06, Eurographics Association, pp. 191–200. 2
- [MC10] MARTIN T., COHEN E.: Volumetric parameterization of complex objects by respecting multiple materials. *Computers & Graphics* 34, 3 (2010), 187–197. 2
- [MCK09] MARTIN T., COHEN E., KIRBY R. M.: Volumetric parameterization and trivariate B-spline fitting using harmonic functions. *Computer Aided Geometric Design* 26, 6 (2009), 648–664. 2
- [MHTG05] MÜLLER M., HEIDELBERGER B., TESCHNER M., GROSS M.: Meshless deformations based on shape matching. *ACM Transactions on Graphics* 24, 3 (2005), 471–478. 3
- [MMP89] MADAY Y., MAVRIPLIS C., PATERA A. T.: Nonconforming Mortar element methods: Application to spectral discretizations. In *Domain Decomposition Methods* (1989), Périaux, Widlund O. B., (Eds.), SIAM, pp. 392–418. 9
- [Moa05] MOAKHER M.: A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications* 26, 3 (2005), 735–747. 4
- [NS10] NOE K., SØRENSEN T.: Solid mesh registration for radiotherapy treatment planning. In *Biomedical Simulation*, Bello F., Cotin S., (Eds.), vol. 5958 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2010, pp. 59–70. 2
- [NW06] NOCEDAL J., WRIGHT S. J.: *Numerical Optimization*. Springer, 2006. 4, 6
- [NYL12] NIVOLIER V., YAN D.-M., LÉVY B.: Fitting polynomial surfaces to triangular meshes with Voronoi squared distance minimization. *Engineering with Computers* (2012), 1–12. 2, 6, 9
- [PL03] POTTSMANN H., LEOPOLDSEDER S.: A concept for parametric surface fitting which avoids the parametrization problem. *Computer Aided Geometric Design* 20, 6 (2003), 343–362. 2
- [PP12a] PAILLÉ G.-P., POULIN P.: As-conformal-as-possible discrete volumetric mapping. *Computers & Graphics* 36, 5 (2012), 427–433. 2
- [PP12b] PETERSEN K. B., PEDERSEN M. S.: *The Matrix Cookbook*. Technical University of Denmark, Nov 2012. URL: <http://www2.imm.dtu.dk/pubdb/p.php?3274>. 4, 9
- [SB11] SZABÓ B., BABUŠKA I.: *Introduction to Finite Element Analysis: Formulation, Verification and Validation*. John Wiley & Sons, April 2011. 5
- [TRBS98] TINOCO-RUIZ J., BARRERA-SÁNCHEZ P.: Smooth and convex grid generation over general plane regions. *Mathematics and Computers in Simulation* 46, 2 (1998), 87–102. 3
- [WGY03] WANG Y., GU X. D., YAU S.-T.: Volumetric harmonic map. *Communications in Information and Systems* 3, 3 (2003), 192–202. 2
- [WHL\*07] WANG H., HE Y., LI X., GU X., QIN H.: Polycube splines. In *Proceedings of the 2007 ACM Symposium on Solid and Physical Modeling* (2007), SPM '07, ACM, pp. 241–251. 2
- [WJH\*08] WANG H., JIN M., HE Y., GU X., QIN H.: User-controllable polycube map for manifold spline construction. In *Proceedings of the 2008 ACM symposium on Solid and physical modeling* (2008), SPM '08, pp. 397–404. 8
- [WLL\*12] WANG K., LI X., LI B., XU H., QIN H.: Restricted trivariate polycube splines for volumetric data modeling. *IEEE Transactions on Visualization and Computer Graphics* 18, 5 (2012), 703–716. 2
- [XC02] XIAO H., CHEN L.: Hencky's elasticity model and linear stress-strain relations in isotropic finite hyperelasticity. *Acta Mechanica* 157, 1 (2002), 51–60. 3
- [XHY\*10] XIA J., HE Y., YIN X., HAN S., GU X. D.: Direct-product volumetric parameterization of handlebodies via harmonic fields. In *Proceedings of the 2010 Shape Modeling International Conference* (2010), SMI '10, pp. 3–12. 2
- [YLSL11] YEH I.-C., LIN C.-H., SORKINE O., LEE T.-Y.: Template-based 3D model fitting using dual-domain relaxation. *IEEE Transactions on Visualization and Computer Graphics* 17, 8 (2011), 1178–1190. 2